
CREATING A NEURAL NETWORK THAT DETECTS COWS

Artefact by Uxue Galvin

Commentary

Saturday, August 31, 2024

7:58 PM

I chose to make a machine learning algorithm because it is a subject that intrigued me but I knew nothing about. Machine learning and computing also link well to my A-level subjects and degree choice (Physics) - I wanted to develop skills in computing since it will no doubt be essential in a Physics degree and possibly in a future career. However, it wasn't possible to choose the type of algorithm and its application, without any knowledge of machine learning and how difficult it would be. So I decided to delay my decision until after I had learnt more about AI and did some simple projects. Consequently, my project had two different phases: phase 1, where I developed the necessary skills in programming and machine learning; and phase 2, where I decided on my project and carried it out. There was planning involved in both phases, but I could only construct a more detailed timeline by phase 2 once I knew more about the shape that my project would take.

I understood that one of the necessary requirements for machine learning is computer programming abilities, so I began phase 1 by taking a course in Python. This was a book called Learn Python the Hard Way designed for absolute beginners like me. Completing it gave me enough proficiency in programming to follow step-by-step machine learning tutorials. For example, I made some regression and classification algorithms. These taught me about data manipulation, tools needed for machine learning, and how to evaluate results. As a result I took notes and added this to my research, while also contributing to my skills development.

Once I felt I had developed enough programming and AI skills, I began phase 2 by choosing my project. My inspiration for the application of the algorithm came from my grandad, who is a citizen scientist. He measures water quality of a river near him, and thought that it is affected when there are cows in fields nearby. This prompted me to add the idea of making a device that detects cows to my list of project ideas. After evaluating advantages and disadvantages of each branch of the mind map, this was the project I decided on. I later used an interview with him as primary research to record more about it, as he is the intended user. My plan for my project outcome was: a device containing a camera, a neural network that detects the presence of cows in an image, and a turbidity sensor. At set intervals, the camera would take photo and the turbidity would be recorded. The neural network would detect whether there were cows or no cows in the photo, and this result, along with the time and turbidity reading, would be added to a spreadsheet stored on the device.

Constructing the cow detecting algorithm was the main part of my project. My prototype neural network was made using a tutorial which teaches a computer to categorise images into those that contain cats and those that contain dogs. After I got this working to a reasonable accuracy, I trained the model using my own dataset of photos with cows and photos without. I then compressed the model to put it on the Raspberry Pi, a small computer used for hobby electronics, and connected the Arduino, which takes data from the turbidity sensor.

In this part of the project I put some skills I had developed into practice, such as programming and working with computers, as well as my AI skills. Designing the project and how long it would take required short-term and long-term planning, which I did using a week-by-week timeline and various to-do lists. I also developed some unexpected skills that I had not planned for: electronics, while making sure the turbidity sensor and Raspberry Pi camera worked.

There were a few problems that I encountered in the process. The biggest setback was when my computer broke, which meant working on another computer. I had backed up most of my files on OneDrive, but some coding files were not in a compatible format. Coincidentally, I had set up an account on GitHub to keep these files just the previous week. GitHub is an online platform used by developers to upload and share code as part of projects. This meant the most important files were on the cloud, so I could retrieve them when using my new computer; most of my work was not lost. However, while my old computer had 2 operating systems that I could switch between for different aspects of my project, I can only use one operating system on the newer computer. I tried to change this but in the end had to settle with using Ubuntu, which works well for data science but is less convenient for Microsoft apps.

Creating the dataset of cows and no cows also proved to be more difficult than expected. At first, I thought of using a dataset from the internet of images of cows, and a dataset with images of fields. Finding images of cows was not a problem but finding images without, that were varied and similar to cow images, had no simple solution. Instead I had to collect a lot of data myself by taking photos of real cows in fields, and photos of the same fields without cows. This meant photos were varied but was inefficient at producing the thousands of photos needed. Another more efficient method of primary data

collection was by taking videos and extracting photos from there. My final method was writing a piece of code that downloads all of the images in an image search page. These solutions helped me with my problem solving and computational proficiency.

I believe that my strengths lie in the project itself, where I independently learnt a lot in a short space of time to successfully create a functioning device. As I mentioned before, this also required problem solving and adapting to new constraints. My ability to maintain flexibility was also an advantage because I know that if I had made a rigid plan without understanding my project properly, this would have changed too drastically later on to be an effective use of my time. Once I understood my project, I broke it down into steps, and broke these steps down into smaller steps. This improved my efficiency, because I was able to see if the simple parts of the program worked first, before adding complexity. My time management was effective in many ways because I remained up to date most of the time, while also making sure to prioritise my A-levels when it was necessary.

Nonetheless I still believe that some aspects could have been improved on. After training the cow detecting model a few times, I achieved an accuracy of around 81%, which is the model that is used on the device. This accuracy is high, but there is room for improvement. To help with this I asked Fred Mannings, who co-owns an AI start-up, to give me feedback. He gave me useful suggestions that I implemented where I could. But each change I made took a long time for the effect to be seen, because it meant another hour to train the model. As a result I didn't have enough time to make as many improvements as I would have liked to the model.

Some of my time management still wasn't optimal; although I did well on this when working on the project itself, my documentation at times was neglected as a result. Getting behind in paperwork was not detrimental to the quality or quantity of it, but it did mean that I had to catch up on the backlog in one go. Managing my time more effectively would have meant doing the documentation incrementally instead, which would have been easier.

If I did the project again, I may have chosen a less ambitious goal, because I would have appreciated some time at the end to improve on my final outcome. But overall I'm really happy that I've created a device that works as I planned it to. The skills I've developed will be vital to my future, such as working with computers beyond the user interface, creating a project from beginning to end, and learning new skills independently. As well as skills, I've also gained a deeper understanding into what a project can be like, and insights into how machine learning and AI works.

Personal audit

06 December 2023 15:28

Personal Audit – 20 Qns about you & choosing a topic

Answer the questions and save this document – you will need to upload it to an EP folder when classes start. You may find discussing the questions with friends or family helps.

1. What do you do in your spare time?	Play piano and clarinet, bake, read, watch films and series, meet friends, play chess, listen to music, go outdoors
2. What do you watch on You-tube? Why?	Videos of classical music performances
3. Which Instagram accounts do you follow? Why?	N/A
4. What are your favourite things to watch on Netflix/ TV / IPlayer / Prime?	Good Omens, Stranger Things
5. Which real or imaginary character do you identify with most? Why is this?	Salieri from Amadeus
6. What is your bucket list of things to complete by the time you are 25?	Reach a chess rating of 2000, get a university degree, get a job, learn to code
7. Any good books you have read recently? Is there a genre you like?	Pride & Prejudice Classics
8. How do you feel about sport? Do you play one a lot? Do you enjoy sport?	Depends on the sport Running, walking, swimming, cycling
9. What sort of music do you like?	70s and classical music
10. Favourite song? Best lyrics? Best album cover? Best music video?	The Chain or Silver Springs (Fleetwood Mac) Gypsy (Fleetwood Mac) lyrics Led Zeppelin Mothership album cover
11. What would you go on a protest march in support of?	Anti-fascism
12. Are there any political events currently that have grabbed your attention?	Extreme right almost winning Spanish elections
13. How do you feel about artistic skills? Painting / drawing / photography/ sculpture/ clay?	Want to improve drawing Would like to learn more about photography Little to no experience in painting, sculptures, clay
14. How do you feel about crafts and creating things? Sewing/ woodwork/ leatherwork/ carving/ construction?	I'd like to try woodwork and sewing
15. How do you feel about working with digital media? Documentaries/ filming/ podcasts/ writing &/or recording music?	Not very interested
16. What item of clothing you possess best represents you? What is your favourite outfit?	Giraffe shirt A shirt with wide jeans
17. If you could learn anything what would that be?	Perfect pitch, overcoming irrational fears, coding, making my own clothes, singing
18. What subjects are you studying at A level? (If you are doing essay-based subjects this will make writing a good dissertation harder as you already have these skills)	Maths, Further Maths, Music, Physics
19. Is there anything you wish you could have taken at A level but are not?	Biology?
20. How do you relax?	Listen to music, go for walks or runs, watch TV

After EPQ launch homework

06-December-2023 15:33

Assignment following EPQ launch webinar.

Dear EPQ student.

You have signed up to watch a live GoTo webinar hosted by Southampton University, launching the EPQ and explaining what it is. This assignment follows on from that webinar, and you will need to submit it to provide evidence of completion in your EPQ classes. This work is designed to be completed in your EPQ IL sessions for the weeks of 18 Sept + 25 Sept. (3 x 1hr 5 mins, and the webinar is the 4th hour).

Note that students will have a library trip in the week of 2 October 2023 (B group classes) or week of 9 October 2023 (A group classes). This will be in the time slot you are timetabled for EPQ.

Task 1. Watch these short videos showcasing EPQs from our students and answer the questions that follow.

EPQ stories 1 https://web.microsoftstream.com/video/9f5608df-2f6b-4f36-94db-b4adb6031f9f	
Name:	Project:
How did she come up with an idea?	She loved textiles in y8 DT but couldn't do DT at GCSE or A-level, and she wanted to represent her being Kenyan, and she wanted to learn more about the world of fashion and textiles
What did she enjoy early on?	Making bucket hats
What will she take away from EP?	She can make a piece of clothing, has learnt about using sewing machines, and can evaluate different sources of information
What advice does she give?	Don't worry about choosing the project, choose it yourself, and ask if you need help
What were her final thoughts?	Your EPQ does not have to be related to what you want to do in the future
EPQ stories 2 https://web.microsoftstream.com/video/26caa0aa-304e-42b3-a08f-ac06f4482830	
Name:	Project:
How did you come up with the idea?	In an exercise writing what it would be fun to do as an EPQ project
What surprised Sam?	He enjoyed some jobs he didn't expect he would, such as admin and organising people
What does Sam think will happen if you choose something boring?	It will feel like a waste of time
How does he think you should pick your project?	Observe what you enjoy separate from studies
Tips for working with mentors:	Listen to what they have to say because they have lots of good life advice
Project management tips:	Start early and time management is important
What's the most fun Sam has ever had in education?	The EPQ project
EPQ stories 3 https://web.microsoftstream.com/video/61c76593-fc2a-48f8-8476-7b298557ccdd	
Name:	Project:
How did EPQ compare to A levels?	It was a fun hobby between A-levels
Secret to success?	You must love what you are doing
What does Lucy say she did when she was feeling uninspired?	Do the admin side of it and sit at the piano and eventually an idea would come to her
How did she stay motivated?	Make sure to do the actual project and not focus on the more uninteresting side too much

1 of 7

What is Lucy proud of?	What she created
What advice does she give?	Make sure you are passionate about it and do it little and often
EPQ stories 4 https://web.microsoftstream.com/video/3e16595b-61e8-44fa-91bd-59b50bb0bade	
Name: Aminah	Project: Spoken Word Poetry Video
Name: Michael	Project: Book of drawings and written ideas about Cambridge
Where did their ideas come from?	Aminah - always loved poetry Michael - wanted to be away from screens and writing so combined outdoors and drawing
How did Aminah and Michael manage their time?	Aminah - Got feedback from other people and made a plan Michael - Kept perspective and made sure to get feedback from other people
What advice do they give?	Do something you enjoy, manage time, don't be afraid to change it
What was a lightbulb moment?	Aminah got ideas from Michael and made a Marvel mind map
What did Aminah learn early on?	How to use Adobe
What were their words of wisdom	Aminah - do something you love and plan time effectively, small steps, get feedback, start early Michael - be honest with yourself, keep time for yourself
What did Aminah say about independence?	Tried to limit asking for help to a degree as she wanted to increase independence
What was Michael proud of?	His project and that he got to know the city well
Who published Michael's project?	Museum of Cambridge
What did Michael gain that he didn't have at secondary school?:	Confidence
EPQ stories 5 https://web.microsoftstream.com/video/71b012d2-a97e-4270-9e3a-93f058b0b0cd	
Name: Remy	Project: Prompt Based Play
Name: Jess	Project: Public Speaking Workshops for Primary School Pupils
What was it like working independently?	Remy - found it a challenge but it became more enjoyable as her time management improved Jess - got a bit dull but she was able to keep herself interested
How did they build ideas	Remy - liked GCSE Drama Jess - felt passionate about public speaking, wanted to gain skills
What advice do they give?	Remy - be kind to yourself as often the results are not what you originally planned Jess - be flexible with time, take breaks when you need to
What are they proud of?	Remy - skills like directing and time management Jess - networking skills and teaching people
EPQ stories 6 https://web.microsoftstream.com/video/55b8fe12-3cf1-4c34-bed4-bc711f199e05	
Name: Sophie	Project: Dissertation: Is Exposure Therapy The Best Way Of Treating A Specific Phobia?
How did she pick an idea?	Chose a dissertation because it would be useful for university
Top tip for tackling a 5000-word dissertation?	Took detailed notes of research, wrote down sources in a log
How did she edit?	Asked other people to read through
What does Sophie think gives you a head start for university?	Research skills, note taking, making a bibliography, reading academic sources

Where is Sophie headed after Sixth Form?	Psychology at Oxford
Tips for dissertations?	Work hard, do the paperwork
EPQ stories 7 https://web.microsoftstream.com/video/07ae7769-0753-49a4-85f8-c8b44b3db8ac	
Name: Rebecca	Project: A Quiet Book
How did she choose the project?	Had studied child development at GCSE, wanted to do something hands on
What did she enjoy early on?	Creating front cover and first page, she was proud and saw how it would come together
What did she like about lessons?	Seeing other people's projects
How did she evidence skill development?	Made 2 spider diagrams showing skills, one before and one after
What skills did she gain?	Research skills, evaluating, presenting her work to others and accepting criticism
What mark did she get?	Full marks
What is she taking to uni, skills wise?	Research, having faith that things will come together
What is her advice:	As long as all actions are evaluated by the end, it's okay. Choose a project that interests you
EPQ stories 8 https://web.microsoftstream.com/video/6148699b-82af-48d6-aa00-c961d4ce9482	
Name: Avigail	Project: Design and build and autowah guitar pedal
What did she enjoy?	It was worth it and she could make the pedal exactly the way she wanted
Why did she make a pedal?	Loves funk, pedals are expensive and cheap ones are poor quality, so she decided to make her own and customise the features she wanted
How did she manage the workload?	Broke it up and didn't spend too long at once. By doing something she enjoyed it became more manageable
How did she balance her EPQ with her subjects?	Used teachers' advice: one "power hour" per week, and one lesson
Advice for new students?	Explore range of ideas, but stick to it once you have decided
Biggest challenge?	Harder than expected; very intricate and detailed work
What was the best thing to come out of her EPQ?	The pedal; also skills like problem solving

Task 2. The links below go to examples of a good artifact and dissertation project. You can see their 7-minute video presentation, and their final submitted paperwork via the links below. Look at these projects.

They are found on the EPQ Moodle site: <https://moodle.hrsfc.ac.uk/> To access this you need your student login + the default password is Network123! (Capital 'N', ends in an exclamation mark)

Type of outcome	Artifact	Dissertation
Video presentation	https://web.microsoftstream.com/video/15874d75-25ad-426c-95eb-dc8c7802b9d8	https://web.microsoftstream.com/video/8828396b-d5fa-4015-ba16-32003b5d2282
Project paperwork.	https://moodle.hrsfc.ac.uk/pluginfile.php/14832/mod_folder/content/0/Artefact%20Full%20project%20paperwork.pdf?forcedownload=1	https://moodle.hrsfc.ac.uk/pluginfile.php/14832/mod_folder/content/0/Dissertation%20full%20project%20paperwork.pdf?forcedownload=1
What inspiration can you take from this project?		
What planning did the student use?	Mood board, mind map, Pinterest board to find style Essay and mood board for each character Annotated sheet music Found references for animations	Created mind maps which then made into lists Colour coded timeline to keep on track Logged diary and flagged problems Note taking (many methods), outlining, Cornell method, mind maps
How did the research move the project forwards?	Made a survey to decide to focus on which specific parts of play	Websites, books, academic research papers, articles Research log with evaluations to make sure data was trustworthy Interview with 2 people Made a survey sent to all students Primary research
How did they show skill progression across the project time?	Time management Did more research after an evaluation	Found valid data Developed research skills

Task 3. OneNote is a good way to keep your EPQ work organised. These videos show you how it works:

<https://support.microsoft.com/en-gb/office/onenote-video-training-1c983b65-42f6-42c1-ab61-235aae5d0115>. Watch the video and then set up an EPQ OneNote in the OneDrive on the college system. Add sections for Project choice, and Planning

Task 4. Think about what a good project may be to do. For inspiration look at the previous EPQ magazines <https://moodle.hrsfc.ac.uk/mod/folder/view.php?id=477>. Considering this and the [personal audit](#) you completed for summer work, construct a table like the one shown in the image below.

How did I choose photography?



Project idea I could be interested in	Skills I may be interested in gaining as a result of doing EPQ (pick 6 and fill them in)					
	1	2	3	4	5	6
Machine learning algorithm	Programming	Maths	Self-learning	Time management	Knowledge of AI	Self-discipline
Organise cycling expedition	Organisation	Fitness	Logistics	Navigation	Bike maintenance	Budgeting
Make an application to learn Basque	Programming	User interface design	Linguistics	Teaching	Software knowledge	Time management
Meal carbon footprint calculator	Research	Programming	Knowledge on environmental impact	Maths	Time management	Knowledge of food industry

Add your summer audit and this HW into the Project choice section of your OneNote by inserting as a print out.

WHAT I WANT TO DO FOR EPQ

- Develop a useful skill
- Relevant to subjects
- Fun
- Haven't done something like this before
- I am drawn to this
- Improve a skill

Idea						
Machine learning		✓	✓	✓	✓?	✓
Arduino				✓		✓
Piano performance	✓				✓	
Coding		✓	✓	✓		✓
Carpentry						✓
Sewing				✓	✓	✓
Russian	✓		✓		✓	
Bike maintenance			✓			✓
Carbon footprint of food		✓		✓		✓
Climbing	✓	✓			✓	
Organise cycling expedition	✓	✓	✓		✓	✓
Write a piece of music	✓				✓	
Make a speaker				✓		✓
Gardening		✓	✓		✓	✓
Basque	✓	✓				

Timetable

Wednesday, October 18, 2023 3:58 PM

	Mon	Tue	Wed	Thu	Fri
9:00	Maths	Physics	Chemistry	Further maths	EPQ
10:25	Maths	Physics	Chemistry	Further maths	
11:35	Physics	Tutor	Maths	Chemistry	
12:40				Wind orchestra	
13:45			EPQ	Maths	Further maths
14:55		Further maths	EPQ (with tutor)	Physics	Chemistry

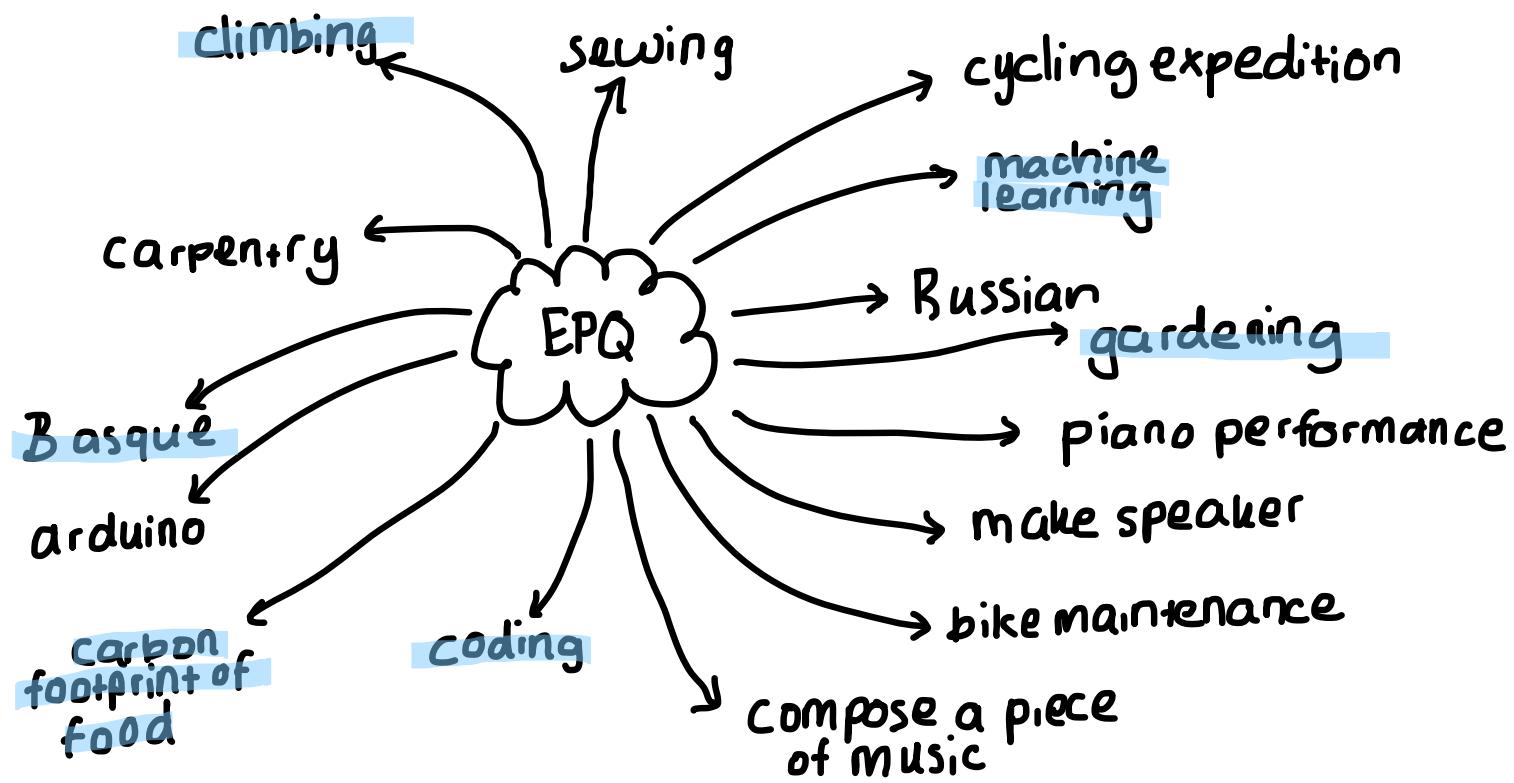
Idea generating table

28 February 2024 15:00

Idea	Dissertation	Artefact	Event
Basque	The history of the Basque language and people	Basque language learning app	Organise a festival of Basque culture and food
Carbon footprint	What everyone should be doing to decrease their carbon footprint	Carbon footprint calculator app	Fundraiser for the environment
Coding	Which programming language is the best?	A website	24 hour coding challenge

Ideas mind map

06 December 2023 21:53

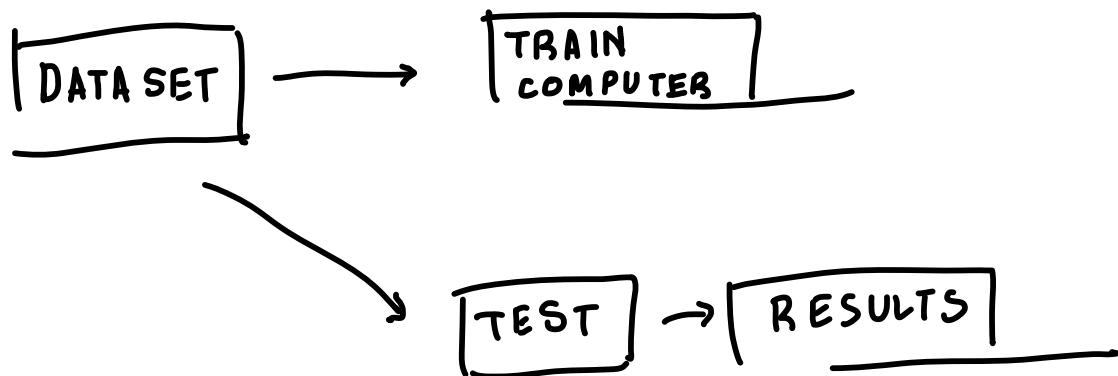


■ probably will choose one of these

Project outcome planning

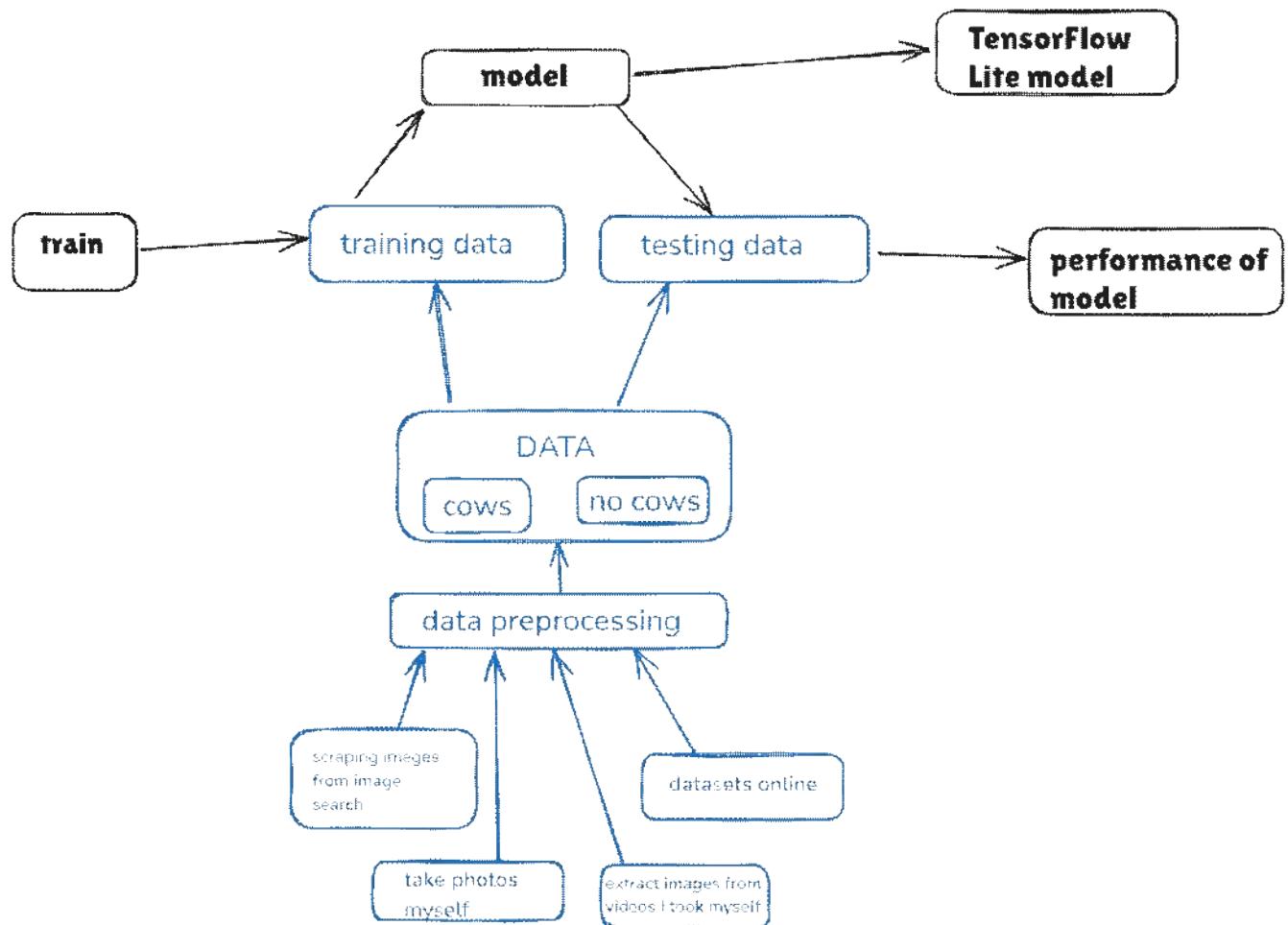
28 February 2024 15:41

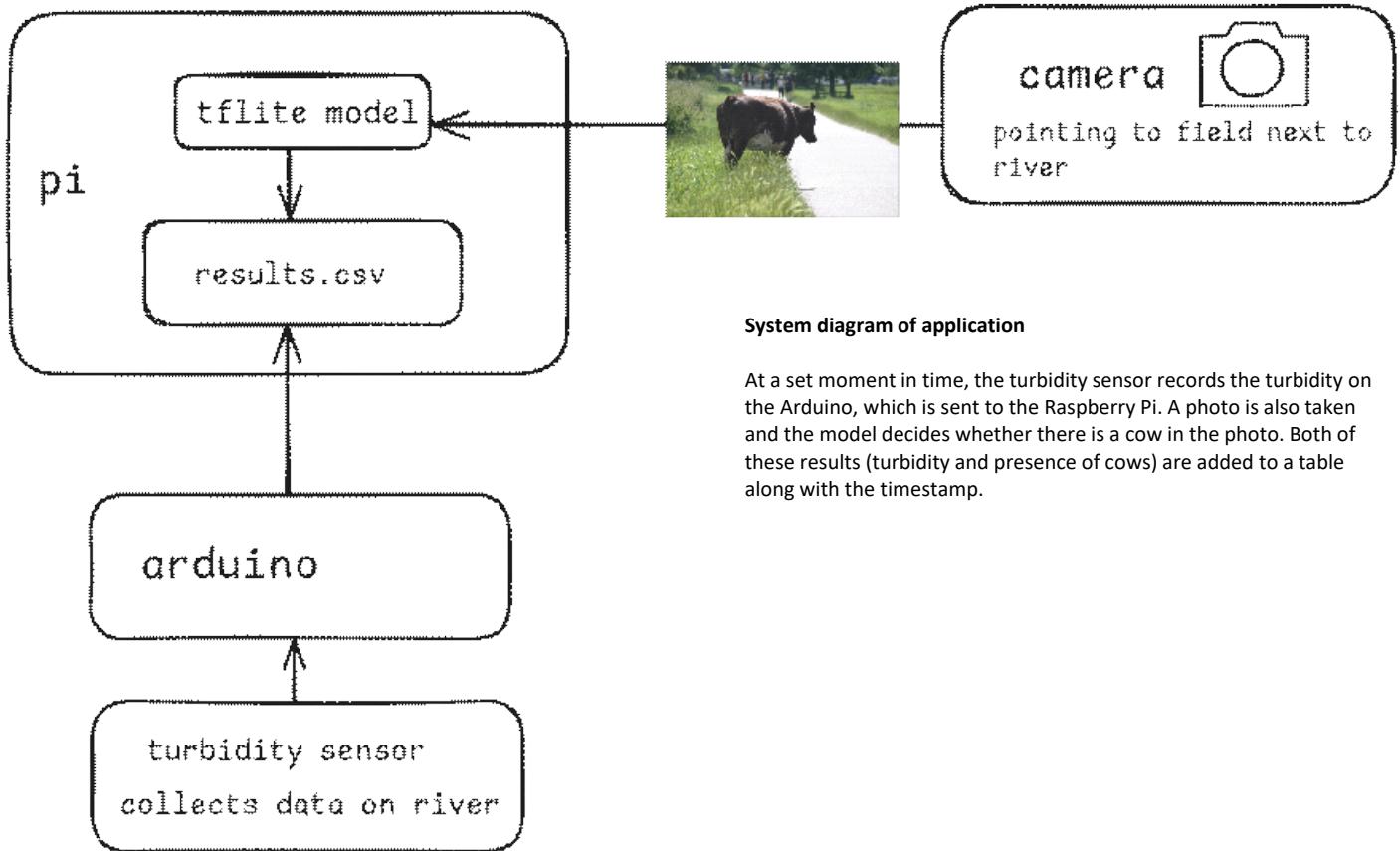
Rough plan (without having decided on project)



System diagram of training

This is the way I will make the model that detects cows





System diagram of application

At a set moment in time, the turbidity sensor records the turbidity on the Arduino, which is sent to the Raspberry Pi. A photo is also taken and the model decides whether there is a cow in the photo. Both of these results (turbidity and presence of cows) are added to a table along with the timestamp.

January checklist

06 December 2023 15:10

- OneNote organised
- Project choice
 - Personal audit
 - After-launch homework
 - Mind maps (project choice)
- Planning
 - Diary updated
 - PPR (on Moodle)
 - Project breakdown
 - Timeline
- Research
 - Primary research
 - Notes from independent learning
 - Source tracker
 - Evaluate with CRAAP
 - Inc from webinars about EPQ
 - 20-25 pieces of research
- Skills
 - [Skills tracker](#)
 - Log lpthw
- Evaluation
 - November progress review
- EPQ notes
 - Mark scheme webinar
 - AO1, AO2, AO3, AO4
 - Time & project management video

February to do list

14 February 2024 15:06

- Personal audit
- Idea generating tables
- Mind map
 - Improve
- Project breakdown
- Timeline
 - Improve
- Research**
 - 20 pieces
- PPR
 - Rationale
 - Aims & objectives
 - Initial planning
- Primary research
- Source tracker
 - CRAAP
- November self-review
- Skills overview
- Skills audit
- Skills log
- Project outcome

Easter to do list

27 March 2024 15:45

- Bibliography
- February self-review
- Skills log/tracker
- CRAAP testing of sources
- Finish mid-project of PPR
- Updated, detailed timeline
- Update diary

May to do list

08 May 2024 15:03

For after Social Action Week

Actual project:

- Finish Ipthw
- Make my own linear regression algorithm!
- Make my own classification algorithm!
- Multiple linear regression
- Decide on project
 - Mind map of ideas
 - Pros / cons
- Collect data
 - Primary?
- Refine data (not sure what this consists of)
- Fit model
- Evaluate performance of model

Add to OneNote:

- Skills
 - Comments on Ipthw
 - New updated skills chart
 - Skills log
- Update diary !!
- Planning
 - Timeline
 - Project breakdown
- CRAAP everything
- Evaluation
 - February self-review
 - PPR
 - Problems / solutions
- Research
 - what is linear regression
 - cost function
 - residual-sum-square
 - gradient descent
 - overfitting & underfitting
 - ordinary least squares
 - Classification
 - object oriented programming
 - Libraries
 - Shell
 - List comprehension

Final to do list

Wednesday, July 10, 2024 3:03 PM

CHECKLIST

- Webinar
- Diary week by week
- Timelines super-detailed week by week
- 25-30 sources of notes
 - Variety of note-taking
 - CRAAP
 - Bibliography (alphabetical)
- Skills audit
 - At least 8 skills
 - 3 beginning middle end
 - Clear colour code
- Skills log

WHAT I HAVEN'T DONE

- Cover page & contents
- PPR**
- Commentary**
- Mind maps
- Project breakdown
- Long-term planning
 - Timeline
 - Gantt chart
- Diary**
- Secondary research**
- Primary research
- Bibliography (update)
- Skills tracking
 - Skills developed section of diary, or separate?
- Skills audit evaluated
- Properly write skills development evidence**
- Final outcome**
- February self-review
- Summer peer-review
- Video presentation process**
 - Planning
 - Script
 - Evidence of making
- Presentation peer feedback
- Project feedback & evaluation
 - Where it could go

Resource repository:
[AO1- Manage](#)
[AO2- Use Resources](#)
[AO3- Develop & Realise](#)
[AO4- Review](#)



Note: You may have to ctrl + click
to open links from this document.

KEY



Icing on the cake.



We love to see it.

(Everyone should aim for)



Important.

(Everyone should have)



Essential.

(Bare minimum)

EPQ Checklist

Done?

Links to Mark Scheme 	Cover page	
	Contents	
	PPR <i>Required by OCR.</i>	
	Commentary Only if <u>not</u> dissertation. 1500 words.	
	Candidate Authentication Form Signed & uploaded on the SharePoint hand-in site.	
AO1 12 Marks 20%	Personal Audit Set as holiday work before Y12.	
	Any homework or notes from lessons. Could also be AO2. <i>E.g., notes from Library induction, notes from EPQ day webinar, etc.</i>	
	Mind maps + project breakdown (evaluated)	
	Long-term planning: Timelines (evaluated) <i>E.g., Gantt Chart</i>	
	Diary/ Log	
	Evidence of short-term plans <i>E.g., 'next steps' in diary, to-do lists, etc.</i>	

Links functional at the time of the making of this resource, academic year 22-23.

A02 12 Marks 20%	<p>Source tracker</p> <p>Source evaluation As a column in source tracking table, or alongside research notes. E.g., CRAAP test, or another method of evaluation.</p> <p>Secondary research notes- you access this source, but the information itself was collected by someone else (pre-existing). Just like the notes you would take in a lesson or a lecture. E.g., bullet points, summarise info, highlight + annotate, Cornell notes, etc.</p> <p>Primary research- information that you have collected yourself (did not exist before). Planning, execution, and analysis of results/conclusion. E.g., interview, survey, visit.</p> <p>Bibliography E.g., Harvard style, or another referencing style of your preference.</p>
A03 24 Marks 40%	<p>Skills tracking (evaluated) E.g., 'skills developed' column in diary, or table.</p> <p>Skills audit (evaluated) Beginning, middle, end; rating out of 10? E.g., skills wheel or table.</p> <p>Skills development evidence E.g., plans, drafts, prototypes, pictures or screenshots, step-by-step examples with explanation, edits, feedback & comments, annotated changes, etc.</p> <p>Employability Skills</p> <p>Your final outcome! E.g., the full dissertation (5000 words), pictures of the final artefact/event.</p>
A04 12 Marks	<ol style="list-style-type: none"> 1. November PR self-assessment sheet 2. February PR self-assessment sheet 1. Spring peer review

Links functional at the time of the making of this resource, academic year 22-23.

20%	<p>2. Summer peer review</p> <p>Video presentation process <i>E.g., planning, script, evidence of making it, etc.</i></p> <p>Video presentation peer feedback & evaluation <small>(NOTE: The linked resource is from 2019 and refers to a face-to-face presentation in the EPQ 'marketplace', which no longer exists now with the digitisation of EPQ hand-in; however, the guidance remains very similar).</small></p> <p>Overall project feedback & evaluation</p>
-----	--

A few ideas to further elevate your project- ***think of more if you can!***

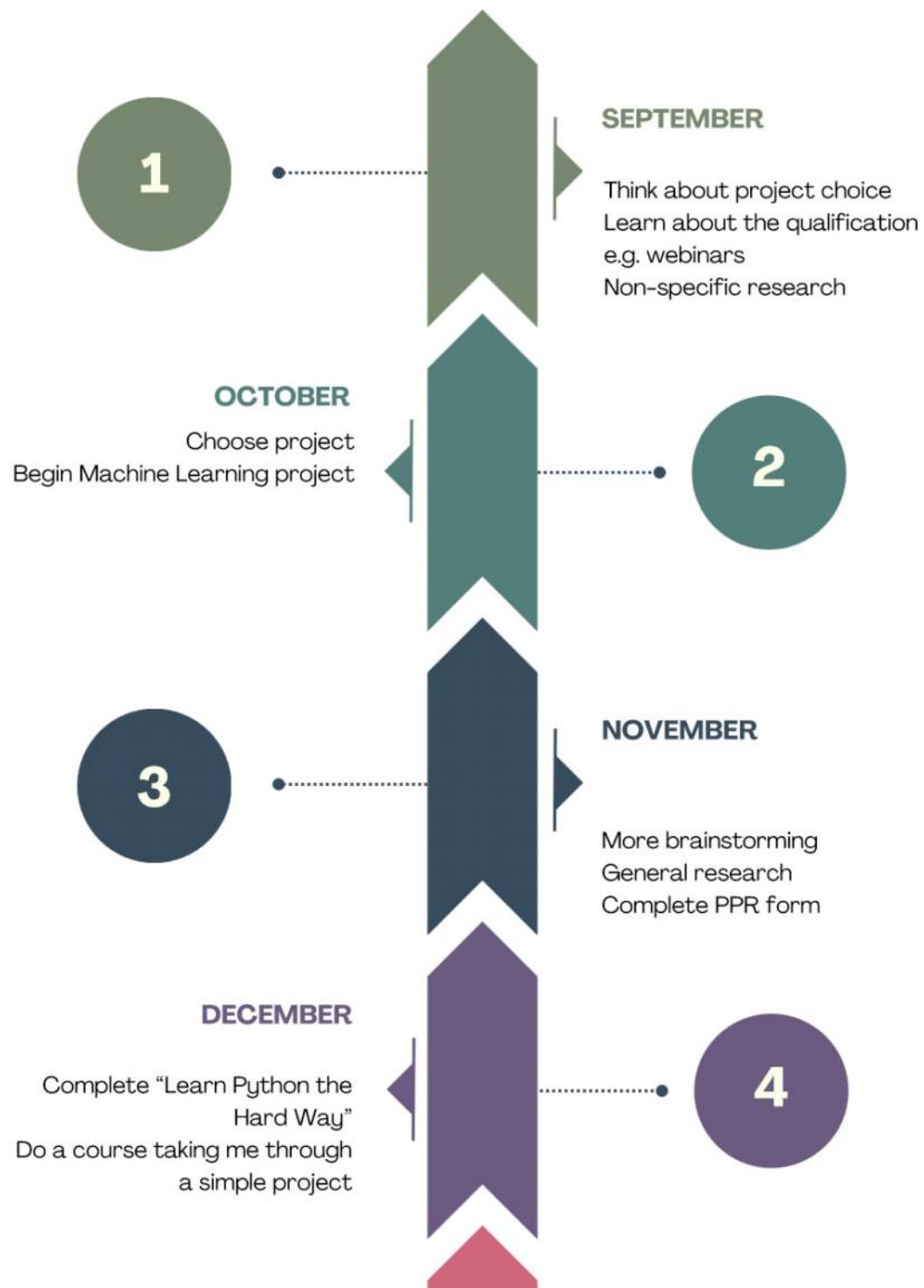
1. Colour coding where applicable
2. Pictures with every AO, not just AO3
3. Extra tables or 'timelines' of your own creation.
 E.g., a skills development table with pictures + explanation, combining step-by-step examples together with evaluation for each stage; mixing and matching different forms of 'tracking' to better suit your project's AO3.
 E.g., making another timeline just for 1 aspect of your project; like a short-term timeline showing just the process of your dissertation writing, stage by stage.
4. Evaluating everything from beginning to end
5. Peer feedback wherever possible
6. Have you demonstrated the full 'scope' of your project?
 E.g., looking into how you might professionally publish your artefact (let's say, if it's a cookbook); even if you don't actually **do** that, good to show you're considering how far it could go.

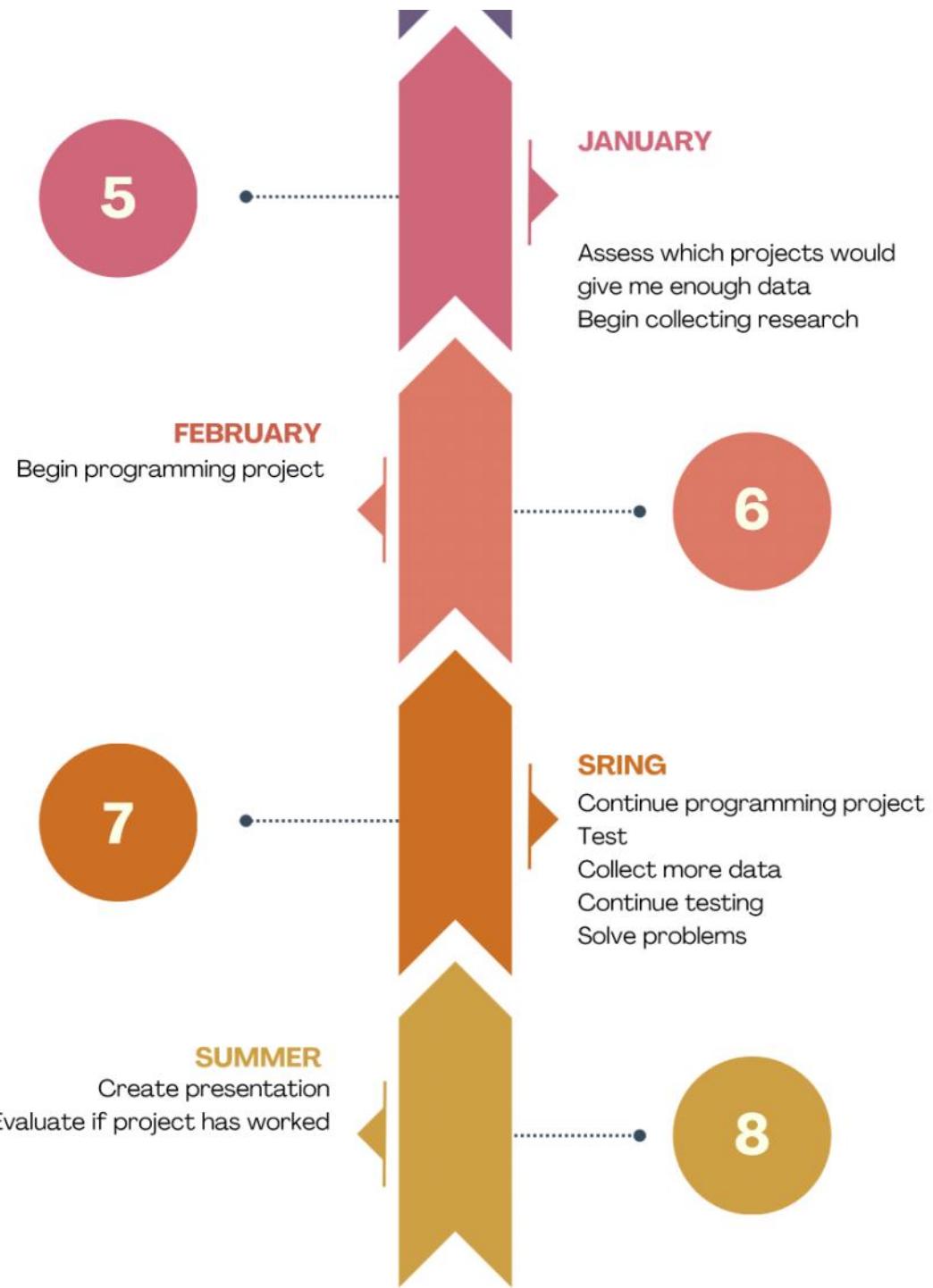
Links functional at the time of the making of this resource, academic year 22-23.

Timeline 1

26 October 2023 17:43

EPQ timeline





Timeline 2 (Gantt Chart)

Wednesday, August 14, 2024 5:32 PM



Timeline 3

Thursday, August 15, 2024 9:13 AM

Week beginning	Tasks
11/03/24	Research: neural network, convolution
18/03/24	Research
25/03/24	Prototype (a neural network that distinguishes between images of cats and images of dogs)
Easter holiday	Continue with prototype
15/04/24	Work out how to use turbidity sensor and connect to Arduino
22/04/24	Write code that sends the data from Arduino onto the Raspberry Pi and puts the data in a spreadsheet
29/04/24	<i>Revision break for end of year exams</i>
06/05/24	<i>Revision break for end of year exams</i>
13/05/24	Finish prototype (cats vs dogs) Start making dataset of cows and no cows
20/05/24	Collecting data
27/05/24 (half term)	<i>Away for half term</i>
03/06/24	Collecting data Data preprocessing – make sure each image is right size, same format (e.g. jpg), in 2 big "cows" and "no cows" folders
10/06/24	Train data and make code to test its performance with unseen data
17/06/24	Tensorflow Lite – research, convert model, test performance
24/06/24	Overdue tasks
01/07/24	Make code that takes photos every x seconds Make code that takes a photo every x seconds and detects presence of cows in each Test in the field
08/07/24	Test Tensorflow Lite model on Raspberry Pi
15/07/24 (summer)	Peer feedback
22/07/24	<i>On holiday</i>
29/07/24	<i>On holiday</i>
05/08/24	Make video presentation
12/08/24	Write commentary

November progress review

06 December 2023 15:41

This is a self review sheet: you will not get any progress review information from your EP mentor at this point.

STUDENT COPY Preparation for Nov Progress Review

Date: 15/11/23

Mentor: Diane Michelson

Student name:	Uxue Galvin
Project area / topic:	Machine Learning
Grade I am aiming for:	A
Attendance (see in Promonitor):	100%
Time spent on EP each week:	1h

Assessment of your progress and paperwork

Place an X in the box to indicate your assessment of your paperwork quality - this aligns with the mark scheme bands.

	Done	Limited	Competent	Good	Strong	In OneNote?
Summer work – Audit	Yes			X		Yes
Evidence of generating ideas (e.g. mind map)	Yes	X				Yes
PPR Rationale, aims + goals identified.	No					No
PPR initial plans + factors to consider.	No					No
Project Breakdown	Yes	X				Yes
Timeline of tasks, planning till Sept 2024.	Yes	X				Yes
Diary, with detail, regular entries, short term plans.	Yes				X	Yes
Method of tracking sources used + evaluation of quality.	Yes			X		Yes
Evidence of notes from sources researched.	Yes		X			Yes

Anything else you have done?

Number of sources looked at so far?

If part of a group project how are roles divided?

Who are you working with?

Began to look at tutorials that will help me develop the skills I need for my project. I have looked at 2 suources so far

I use an IL period for EP each week	Always	Often	Sometimes	Never
I work on EP outside of my class + IL period each week.	Always	Often	Sometimes	Never
I update my diary every time I work on my project.	Always	Often	Sometimes	Never
I record my sources every time I do some research	Always	Often	Sometimes	Never

I would judge my progress so far as (tick one):

Outstanding	
Making expected progress	
Needs to improve	X

If you ticked "needs to improve" highlight which of the categories below you feel you need to improve. Say **how** you will do this

- Attendance
- Punctuality

Record the **HOW** here:

Allocate and stick to an hour of

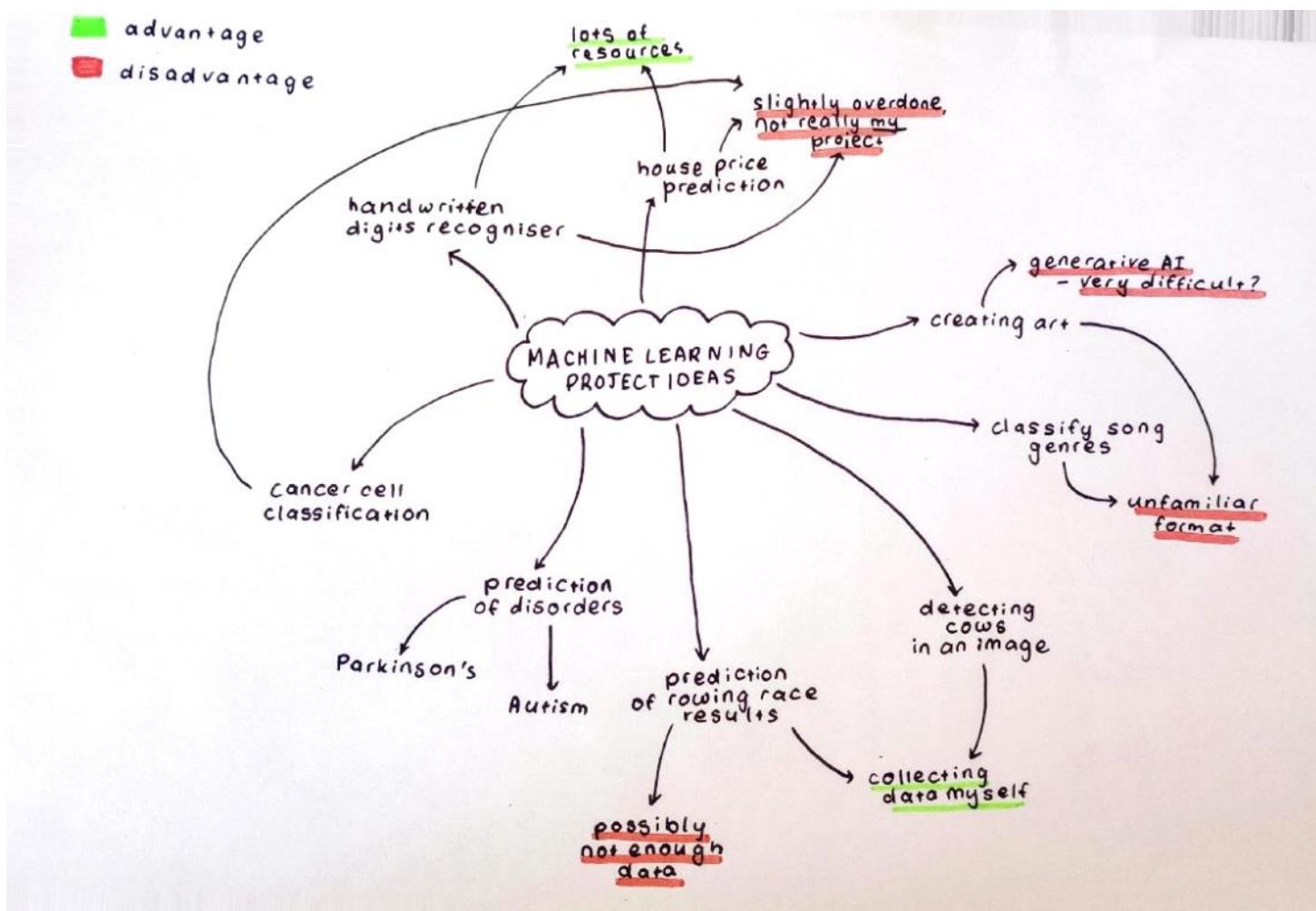
independent learning per week

Write and complete a to-do list of EPQ

<ul style="list-style-type: none"> • Communication • Organisation • Preparation • Engagement in lessons • Subject understanding • Independent learning 	related activities for each week
What are your next steps?	Create timeline Complete PPR Do more research Brainstorm project ideas Continue machine learning project

Machine learning project ideas

28 November 2023 21:17



With these advantages and disadvantages in mind, I have decided to make a project that detects cows in an image. This has plenty of possibilities of being developed as a project, because I could connect a camera and see if it detects the presence of cows in real time.

The inspiration for this project idea is from a relative, who collects data on the quality of the river next to his house. He believes that there is a correlation between how clean the water is and the presence of cows in the field nearby. A way to prove this would be to have a device which detects whether there are cows or no cows in the field and measures the turbidity, at given time intervals.

Project Progression Record

Level 3 Extended Project Qualification

This form should be completed by the learner. Expand the boxes as needed

Name: Uxue Galvin	Centre: HRSFC
Candidate no: 153880	Centre no: 22147

Current programme of study: all A Levels you currently take		
Qualification Type GCE	A Level	Subject <ul style="list-style-type: none">• Maths• Further maths• Physics• Chemistry

Before starting this form you should have already explored various themes and ideas for your project and proposed a suitable topic to your mentor/supervisor

Date Project started: 18/09/23
Project Topic: Machine Learning
Rationale: why you want to do this project and how it links to progression plans/current courses/personal interest. <u>Expand the box as needed</u>
Doing maths and physics provide a strong foundation in analytical thinking and problem-solving, which are important skills in machine learning. The project offers a practical application of the concepts I'm learning in class, bridging the gap between my studies and the real world.
Besides the academic aspect, my interest in machine learning is also inspired by a will to explore the subject and learn more about it. I'm fascinated by AI, especially as it becomes more and more prevalent. Nonetheless, the prominence of AI also has its dangers, which is why I believe it is important to educate oneself about it. Making a machine learning algorithm will teach me about the fundamentals of AI and increase my understanding.
Another reason why I want to do a machine learning project is because I see myself working in this field in the future. I recognize the potential of machine learning to make a big impact in many industries. In current global events and everyday life we can see the increasing importance of AI, such as in chatbots, educational platforms, data science, and computer vision. I know that in a job in computer science, employers often look for projects that a candidate may have done in the past. This project is a hands-on opportunity for me to understand the subject better and possibly prepare for my future career.

<p>Aims and Objectives: What outcome(s) do you hope to achieve? What skills are you aiming to develop?</p> <p>My hope is to have a working machine learning algorithm by the end. However, I understand that it may not necessarily work as planned. My main objective is to have learnt about how to develop a machine learning project. I want to develop skills including time management, programming (which my project will involve a lot of), problem solving, self-learning, evaluation and improve my knowledge of AI.</p>	<p>How will you develop these skills?</p> <ol style="list-style-type: none"> 1. I am currently doing a course (a book called "Learn Python the Hard Way") in Python, the programming language. Completing this, along with applying my new knowledge and learning more as I program my project, will help me develop my pogramming skills. 2. In order to improve time management skills, I am timing how long I spend on EPQ each week. This means I am more likely to get more done in the set time that I give myself. I'm also writing what I plan to do each week so that I have a clear idea of what I need to complete by the end of that week. 3. Completing this course, and others in machine learning, will allow me to develop skills independently. This is a new experience for me so I'm hoping to improve my self-learning. The courses will also teach me about AI and machine learning, as they will hopefully help me understand how to make a project. 4. Recording my progress in my diary enables me to keep a clear record of what I have done each week, thereby showing how well I have spent my time each week and making sure I am on track. I also evaluate whether each task I have done is helpful which develops my evaluation skills
---	---

Initial Planning

What range of resources might you use/what types of research might you complete.....e.g. specialist materials/workshop space. Books/academic journals etc. How will you ensure a good range? Expand the box as needed

A computer

A text editor and terminal, which I will be using to make my project and run it

A dataset to train and test on

Books will be a good source of research to help me understand some concepts

Articles and vidos can be another source of research to help inspire me with project ideas

Tutorials and courses in programming and machine learning will help me develop the fundamental skills needed to embark on such a project

What factors might you need to take into account when planning how to manage your project? How will you plan to bring your project in for the deadline? You should create a timeline.

Though I have an outline of what I plan to complete in the next few months, I understand that I will only know how long these tasks will take once I learn more about the project I am doing. This means I need to maintain a flexible approach to my timeline, because each point may take a longer or shorter time than I expected. Moving forward in the project I will have a clearer idea of what to do in the time ahead of me, so I plan to update my timeline as I progress.

2023

Summer

Thinking about project choice

September:

Learn about qualification

Do general research

October:

Choose project

Begin Machine Learning course

November:

More brainstorming

General research

Complete PPR form

Continue Machine Learning course

December:

Complete "Learn Python the Hard Way"

Do a course taking me through a simple project

2024

January:

Assess which projects would give me enough data

Begin collecting research

February:

Begin programming my project

Spring:

Test project

Collect more data

Continue testing

Signed UXUE GALVIN

Date 22/12/23

Mid Project Review

Date: 06/03/24

How much progress have you made in meeting your outcomes? Expand the boxes as needed

My aims for this point were to have finished the course in Python, my research on programming, tutorials on machine learning algorithms, and deciding what my machine learning algorithm will do. I also aimed to start making the project, doing more research into machine learning, and finding data to train and test my project. In some respects I am on track, because I feel that my computational abilities are ready to start a project, and doing the tutorials and research have

vastly increased my understanding of machine learning. However, I still haven't chosen a project, which is a crucial step that I need to take to bring the project forward.

How has your project changed or evolved?

I began with a very vague idea of my project. Further research and skills development has helped me understand what my project will look like. Although I still don't know what the algorithm will do, I have an idea of how long, and how much effort, it may take. This understanding also means I can plan ahead more thoroughly in the future.

What have been the strengths/weaknesses with your project management so far?

Once I have begun working on the project I find it exciting and easy to motivate myself, so I can be very productive. Maintaining a flexible timeline and plan from the beginning has allowed me to understand the best way to manage my time as I learn more about what I am doing.

I still haven't found a balance between working on the project itself, and the reflecting and documenting side of the qualification. When I am working on the project, my notes in the meantime are minimal because I plan to complete them later. However, this leads to a backlog of paperwork that is not yet completed. I also tend to work on my project in bursts, instead of in stages, which leads to long periods of lack of motivation.

How are you evidencing the research you have done and how you are managing your project?

For every piece of research I do, I take notes; I have grouped my research pages into a section of OneNote. Each piece of research references a source, which is then evaluated using the CRAAP method in my source tracker.

What do you still need to do?

My next steps will be to brainstorm what projects I could do, and then make a decision. After this I can gather data and start training the model. It will quite possibly be an iterative approach; I might edit the code and run the model then see how I can improve. After this I can evaluate the performance of my model, and ask others (with a machine learning background) to give me feedback.

Signed: UXUE GALVIN

Date: 06/03/24

Response to advice from Mentor (at Progress Review points and in 1-1 conversations)

Date: 21/06/24

At PR + Folder review points log your response to verbal advice or assessment from your mentor: what will you change as a result? Expand the box as needed.

My mentor emphasised the fact that although my project has a lot of potential, I have fallen behind on the paperwork. For example, I haven't properly logged my skills development, or evaluated many of my sources using the CRAAP method. I've also left my diary without updating it for a few weeks. As a result, I will try to catch up on these things by making a checklist of what I am missing. From now on I'll change my way of working so that each time I do some work on the

project, I update my OneNote properly so that there is no backlog, which leads me to demotivation.

End of Project

Expand the boxes as needed

Evaluation of video presentation after considering peer feedback

The feedback has been positive overall, and it has also told me what I could have improved. I think that the project outcome could have been better explained, and I should have shown more evidence of research and skills development. One of the specific things that people wanted more elaboration on was documentation of skills development. From this I can conclude that if I did the presentation again, I would add another slide showing the evidence of developing skills throughout the project, and talk a bit more about the final outcome. Nonetheless, I was approaching the time limit so I think my presentation has gone well, given the constraints.

Evaluation of the whole project

I am very satisfied with the outcome of the project. Every component of my device works and the accuracy of my cow detecting machine learning model is adequately high. For me an even more important result of this project is how much my computational abilities have improved: I now have a good knowledge base to learn more about computing, which I'm sure to use in the future. I've also developed many other essential skills in the past year as a result of my project, which I think is the aim of the EPQ.

Although everything has gone to plan overall, I think that I might have done some things differently in retrospect. For example, the application of the machine learning algorithm was more complex than I foresaw, so I had to leave a lot of time to design this. This left less time to improve on the final outcome. It would have also given me more time for testing the device in the field.

Throughout the project, I have also noted my struggles of keeping on top of paperwork. This has led to me occasionally getting behind in updating it. As a result, instead of recording some of my progress little and often, I did it less often and in larger chunks. I think this is an area of time management that I have gotten better at in the course of the project, but could still be improved.

Where is the evidence within your project that demonstrates:

Expand the text boxes as needed.

Outcome	Evidence – where can it be found? E.g. log book, folder etc.
Exploration of a range of ideas for your project	Idea generating table, diary, mind maps
How you have managed your time	Timetable, timeline, diary
Use of a wide range of research and resources	Source tracker, research notes, bibliography, primary sources
Evaluated research and selected appropriate research	Source tracker

How you made decisions and solved problems	Diary, project log, problems / solutions
How you used technology – where appropriate	Diary, project log
Presentation given to an audience	Presentation script, feedback
Making links to HE/Career	Diary, presentation script, commentary, rationale
How you have reflected on your project management skills and the quality of your outcome.	Commentary, what I would do differently, presentation script
On-going evaluation	Progress reviews, diary
Final Evaluation of project and presentation	Commentary, presentation script, what I would do differently, presentation feedback
Non dissertations– 1500 word supporting statement	Commentary

Date of Submission: 03/09/24

Student Signature: UXUE GALVIN

This form should be used to record the progress of each learner.

A copy of this form must accompany each learners work when it is submitted for Moderation.

Diary term 1

26 October 2023 17:03

Week beginning	Time spent	Progress	Evaluation (positives/negatives)	Skills development	Plans for next week
Summer	30m	Personal audit Looked at past students' projects Thought about possible projects	Audit allowed me to see what I enjoy doing and am passionate about Should have recorded ideas Unsure on which to pursue and the ideas felt too ambitious	Self reflection	Continue learning about the qualification Brainstorm more ideas for what to do
18/09/23	45m	EPQ launch webinar - took notes	Advice on how to approach as a whole Quite general and not specific to what I want	Planning	Complete homework about different projects
25/09/23	2h	Post-webinar task: Watched videos on others talk about their EPQ Brainstormed my own ideas and evaluated how useful, relevant, fun, new they are	Detail on how other people approached EP Advice from students and mistakes that they made I have more ideas on what to do that are more realistic	IT skills Brainstorming	Continue brainstorming ideas
02/10/23	1h	Library session - learnt about what websites to trust when doing research and where to look in the school library for research Now fairly sure that I want to do my EPQ in machine learning	Very useful recommendations for websites, eBooks, books for references Made a decision on the area of my project but I don't know about my specific project	Research skills	Continue brainstorming ideas Look for more websites to
09/10/23	1h	First EPQ session (independent as teacher was ill) Accessed Moodle Created EPQ notebook on OneNote	I have a space to keep all of my EPQ work Moodle has plenty of useful resources that can help me with tips on research, how to get a good grade, etc. My project choice isn't specific enough yet	IT skills Knowledge of qualification	Continue brainstorming ideas
16/10/23	1h	First face-to-face EPQ session Made diary	Have a place to record all the work I have done each week Learnt more about the qualification and what it comprises	Planning Knowledge of qualification	Complete PPR form Brainstorm specific project ideas
23/10/23 (half term)	1h	Completed Moodle tasks	Told me more about the qualification I have already been given quite a lot of advice so I think that time would have been better spent in trying to narrow down my project Haven't completed PPR form	Knowledge of qualification	Complete PPR form
30/10/23	1h	I have continued to research what project to do, and found a competition for an algorithm that recognises handwritten digits, which looks interesting Completed some pieces of research that told me more about machine learning Learned about Harvard referencing in class and made a bibliography Watched a video at the beginning of a machine learning course	I now have a possible idea of a specific project to do The research, especially the video introducing the machine learning course, really helped as it taught me more about machine learning and told me what to expect about a project in it. The articles also gave me ideas on a possible project choice I also know how to make a bibliography now and use references	Research skills Evaluating sources	Continue with machine learning course Make a final decision on what project I am going to do?
06/11/23	1h	I didn't complete much this week Started looking at the PPR form Began to complete November self review sheet	Haven't kept on schedule in EPQ independent learning sessions Still haven't narrowed down options for projects No progress		Complete PPR form Continue with Machine Learning course
13/11/23	2h	I began a course in Machine Learning from Coursera, and recorded what I learnt in the research section I also began a course to learn Python ("Learn Python the Hard Way") - I completed up to exercise 10	Still not on schedule I don't feel like I know enough about how my project will continue	Coding	Continue on both courses
20/11/23	1h	Continued Learn Python the Hard Way (exercise 20) Completed research on creating a timeline, time management, and note-taking	This research was definitely more relevant to my project so it left me feeling more prepared I didn't continue the machine learning course, but I think I will put this aside for the moment since I should get a grasp of the basics of programming first	Coding Time management Planning	Continue on both courses
27/11/23	1h	Completed some research about AI I have a good idea of what I am going to do for my project and my next steps. I have decided to create an algorithm that predicts results of rowing races based on the prior results of rowers Created a loose timeline that I can add to later	I don't think that this research is helping me as it is very general and from the perspective of users of these tools, and not the creators I will need more guidance on my timeline because I don't think it is detailed enough at the moment However, I am feeling more confident now that I have a feasible project idea and a much clearer idea on what I need to do next	Coding Note-taking Evaluating sources Research skills	Complete more pieces of research Attempt a course that gives me a project and takes me through how to do it
04/12/23	1h	Continued Learn Python the Hard Way - doing stuff with files on Python, learning about functions, and consolidation History of AI	I'm enjoying learning Python and it's nice to know that I'm making progress in this respect Learning about the history of AI will not directly help me in my project, but it provides a good background	Research skills Coding	Continue on Learn Python the Hard Way
11/12/23	2h	Continued Learn Python the Hard Way – learning about Boolean logic and if/else/elif statements Researched what libraries are	Researching about libraries proved difficult to understand, because many explanations included technical terms that I didn't understand However, summarising things in simple terms is a useful skill so I tried to do what I could	Research skills Coding & computational proficiency	Continue doing Learn Python the Hard Way Do more research into these concepts
18/12/23	2h	Continued Learn Python the Hard Way (exercise 35)	Made progress in Learn Python the Hard Way but no research	Coding	Finish Learn Python the Hard Way? More research PPR
Christmas holidays	2h	Completed first section of my PPR Continued Learn Python the Hard Way (exercise 40) Evaluated how the project has gone so far	Though slightly late, I am happy to have completed the PPR for this stage of my project. It has allowed me to evaluate my progress so far, and I understand that I can take an approach where I do a lot of important skills development before deciding my project choice	Evaluation Planning Coding	Make a new, updated timeline that is more specific Finish Learn Python the Hard Way Start a machine learning algorithm tutorial

Diary term 2

06 January 2024 11:08

Week beginning	Time spent	Progress	Evaluation (positives/negatives)	Skills development	Plans for next week
08/01/24	2h	Python the Hard Way (exercise 44) Learned how to create a virtual environment, installed Jupyter Notebook Research: virtual environments, Jupyter Notebook	I've learnt all I need to for now from the Python course, so I can start doing machine learning tutorials Learning about virtual environments and Jupyter Notebook has given me a digital setup where I can do my project and prototypes	Computational proficiency Research	Start a simple machine learning tutorial
15/01/24	2h	Followed a tutorial for a simple linear regression algorithm (using Jupyter Notebook) Research: linear regression, cost function	My first machine learning algorithm! This was simple and satisfying and gave me an idea of the steps I will have to go through for my project It also helped influence my choice of project because this algorithm definitely appeals to me Learning about cost function helped me gain insight into how a machine <i>learns</i> Doing the tutorial on Jupyter Notebook helped me gain familiarity with it	Understanding of Machine Learning Research	Do a tutorial on a classification algorithm
22/01/24	1h	Research: steps to create a machine learning algorithm, loss function Edited the code of the linear regression tutorial (using skills from the Python course)	No progress in looking at other types of ML algorithms However, it was useful to apply my new Python skills to improving the code from the tutorial	Research Understanding of ML Programming	Do a classification algorithm
29/01/24	1h	Learnt how to make csv file from Jupyter Notebook, which could be a way of storing data for my project Did a linear regression tutorial from scratch	Still haven't done a classification tutorial However, doing another linear regression tutorial, this time without relying on libraries, was valuable because it taught me how linear regression actually works	Understanding of ML Computational proficiency	Do a classification algorithm
05/02/24	1.5h	Started a tutorial on logistic regression (type of classification algorithm) Contains a lot of data handling since the dataset is very large and imperfect	This tutorial is more thorough so takes longer but closer to programming a real life algorithm, where the data isn't always perfect Taught me how dealing with data may be a lot more time-consuming than working on the model itself It also gives an explanation as to what a logistic regression algorithm is and does, so serves as both a piece of research and skills development	Data handling Understanding of ML Research	Finish the logistic regression algorithm
12/02/24	1h	Finished (most of) the logistic algorithm tutorial – I skipped the last part because it contained a lot of evaluation which I will research into later on if I have time The accuracy was 85%, which is what I aimed for	I'm satisfied that my model turned out to be as accurate as the tutorial said it would I think I've done enough tutorials to decide which one to choose as my project	Research Understanding of ML Data Handling	Choose project
19/02/24 (half term)		Away for holiday so no EPQ completed			
26/02/24	1.5h	Mid-project review Multiple linear regression tutorial	Doing the review helped me realise that I am missing a few things from my OneNote so I am going to try to keep more up-to-date from now on Really need to choose a project now!	Research ML skills Evaluation	Choose project
04/03/24	2h	Brainstorming – made a mind map listing the advantages and disadvantages I could think of for each project Decided on my project – a neural network that detects the presence of cows in an image. This can be applied to seeing if there is a correlation between the presence of cows in a field and the turbidity of a river nearby Made a precise timeline, with tasks for each week Made some of the project breakdown	Now I have a very clear plan of what I need to do each week which makes the project management side a lot more straightforward	Decision making Brainstorming Planning	Research on neural networks
11/03/24	2h	I watched a very good video on neural networks which I took notes on I also taught myself about convolutional neural networks, which is what I am going to use for my cows-vs-no-cows algorithm	The video explained neural networks well, but it seems like something very difficult to do from scratch Nonetheless, I am happy that I now understand neural networks because this is the theory behind a key part of my project	Research	More convolution algorithm research
18/03/24	1h	Made a convolution algorithm to help me understand more about what they do	Last week I don't feel I had properly taken in what a convolution algorithm does, but this tutorial helped me understand it in practice so now it is clear in my mind	Self-learning Research Programming	Start making prototype
25/03/24	1h	I started making my neural network prototype on Jupyter Notebook – one that sorts images into images of cats and images of dogs So far I have made a new virtual environment for this purpose, downloaded the dataset, and created folders to put testing data and training data	I can see clearly how I can change the code so that it detects cows instead of cats and dogs, so I think it will be more straightforward than I thought to make the neural network	Programming	Keep making the prototype
Easter holiday	30min	Made an Easter to-do list as advised by tutor I completed little EPQ this holiday because I was revising for exams	It showed me how little of the paperwork side I have <i>finished</i> (although most of the things I need to do are in note form, just not presentable); I have a lot to catch up on	Planning	Complete some of to-do list Keep making prototype

Diary term 3

Wednesday, August 14, 2024 2:18 PM

Week beginning	Time spent	Progress	Evaluation (positives/negatives)	Skills development	Plans for next week
15/04/24	2h	Encountered a bug with the cats vs dogs algorithm Diary catch-up and to-do list	The bug worries me (especially since time is tight) but I'm hoping that leaving it for a week will give me more ideas	Problem solving Planning	See if I can fix the bug and finish the prototype
22/04/24	2h	I've finished writing the code for training the cats vs dogs and it is working for first time! Made an "inference" script, which gives the model unseen data and shows the rough accuracy for "cat" images and "dog" images However the training takes a very long time so I am using my brother's gaming computer which is more powerful	I'm now ready to start making the neural network that recognises cows, instead of cats and dogs The gaming computer will be far more efficient	Computational proficiency ML skills Problem solving	Start the cow detection algorithm
29/04/24		<i>Break to prepare for end of year exams</i>			
06/05/24		<i>Break to prepare for end of year exams</i>			
13/05/24	3h	Making my own dataset: extracted images from videos I took myself of cows/no cows, and took individual photos Tried out some training	Using frames from images is an efficient way to get lots of images, however, I think I still need more data because the accuracy with unseen data is quite low	Programming ML skills Data handling	Get more data to try to increase the accuracy
20/05/24	2h	Making cows dataset: scraping images in bulk from image searches, and finding datasets from the web Learned about GitHub and uploaded files onto here in order to back them up	GitHub is used by many developers, and is a way of sharing a project, so it will be very useful for not just having somewhere to back up my files My dataset is also much bigger, but I still need to preprocess it because they are in many folders and I need one big folder instead	ML skills Data handling Computational proficiency	Get the dataset into a format that I can use in the code for training
27/05/24 (half term)		<i>Away for half term</i>			
03/06/24	2h	Processing the data so that all of the "cows" images are in one large folder and all of the "no cows" images are in another More training, and the accuracy is a lot better now Primary research: sent an email to Rob Reed, who gave me the idea of the project, with some questions which he answered	The neural network is now complete, which is an important milestone in the project which I am proud of The primary research helped remind me and inform me of the objectives of my project	ML skills Data handling Research	Write up the primary research Convert the model to TensorFlow Lite so it is possible to use it on the Raspberry Pi
10/06/24	1h	My computer broke, so I have a new computer This does not have my files on, but since I backed the files up onto GitHub, I downloaded them again	This is quite a setback, but I am glad that my files were backed up or I may have lost a great deal more Nonetheless I couldn't get any work down this week as a result so am behind schedule	Problem solving Computational proficiency	Convert the model to TensorFlow Lite
17/06/24	1h	Converted the model with highest accuracy to TensorFlow Lite, so it is ready to go on the Raspberry Pi 1-1 meeting with tutor, which told me that I am still behind in the paperwork, so I made another to-do list	The meeting emphasised my issue throughout the project which shown me that I am often too absorbed in the project itself and don't pay enough attention to documenting my progress	Evaluation Planning Programming ML skills	Go through to-do list Make the inference for the tflite model
24/06/24	2h	Made the code for the inference - testing with unseen data and checking accuracy - with a tflite model Updated my GitHub repository and wrote a readme (gives instructions of how to use, explains background and purpose of project)	Writing the readme showed me the big picture of my project as a whole, so was useful to tell me what I've done and what I still have to do	Programming ML skills Evaluation	Try the inference on the Raspberry Pi
01/07/24	2h	Learned how to take photos from the command line, so I made a program that takes a photo and model detects whether there is a cow in the photo or not Filmed a video demonstrating this at home with photos of cows & no cows in front of the camera Went out into the field with my laptop to try on real cows but it didn't detect any due to poor quality camera	I made a lot of progress this week with the project and can see that there isn't much left to do Disappointed that the program on my laptop didn't work in the field, but I have already tested it at home and it will be much more likely to work with the Raspberry Pi and camera	Programming ML skills	Try inference on Pi
08/07/24	1h	The Pi camera was not working, so I went to the Raspberry Pi shop. There was an issue with both the camera and the Pi, so they replaced both and now I have a working Pi and camera I trained the model one last time, to get a graph showing how the accuracy and loss change throughout	I am frustrated by the delay in getting the inference on the Pi but at least now I know that it should work	Programming Problem solving	Try inference on Pi and maybe code that takes photos continuously
15/07/24 (summer)	2h	Emailed Fred Mannings, my dad's colleague, who has a start-up that does machine learning; asked him for advice on my GitHub readme and how to improve accuracy of the neural network Wrote the script for inference on the Raspberry Pi, which works	It was very useful to hear from an expert, and he was also very encouraging and complimentary of the project I have some clear next steps to act on that he pointed out The inference on the Pi is the last part machine learning part of my project, so the only part of the project I have left is the turbidity sensor	Evaluation Programming Computational proficiency	Connect turbidity sensor to Raspberry Pi and record data in spreadsheet
22/07/24		<i>On holiday</i>			
29/07/24		<i>On holiday</i>			
05/08/24	1h	Worked out how to use turbidity sensor, and printed some data into a csv file onto the Raspberry Pi Acted on some of the points in the peer review	Now that all the pieces of my device are working, I can consider my project finished!	Skills with electronics Programming (Python & C) Computational proficiency	Write up skills development more thoroughly and final outcome Plan video presentation Start writing commentary Write my own evaluation of the project
12/08/24	2h	Filled in gaps in my OneNote Planned video – 2 drafts of script	My OneNote is now almost ready to be exported I'll film my video next week since it is ready	Evaluation Planning	Make the video
19/08/24	3h	Filmed and edited video Collected feedback	My feedback has been very positive so I'm happy with the video outcome	Video-editing Presentation skills	Complete PPR, commentary, cover page and export the pdf
26/08/24	2h	Final section of PPR and commentary	My project is now complete!	Evaluation	
02/09/24					

Bibliography

Wednesday, April 17, 2024 3:05 PM

- 0sparsh2 (2021). cats-vs-dogs-coursera-assignment/Cats_vs_Dogs_CourseraAssignment.ipynb. [online] GitHub. Available at: https://github.com/0sparsh2/cats-vs-dogs-coursera-assignment/blob/main/Cats_vs_Dogs_CourseraAssignment.ipynb [Accessed 17 Aug. 2024].
- Alake, R. (2023). Loss Functions in Machine Learning Explained. [online] Datacamp. Available at: https://www.datacamp.com/tutorial/loss-function-in-machine-learning?dc_referrer=https%3A%2F%2Fwww.google.com%2F [Accessed 11 Aug. 2024].
- Amos, D. (2023). Object-Oriented programming (OOP) in python 3 – real python. [online] realpython.com. Available at: <https://realpython.com/python3-object-oriented-programming/> [Accessed 24 Apr. 2024].
- Anyoha, R. (2017). The History of Artificial Intelligence. [online] Science in the News. Available at: <https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/> [Accessed 12 Jan. 2024].
- Banerjee, P. (2020). Logistic Regression Classifier Tutorial. [online] kaggle.com. Available at: <https://www.kaggle.com/code/prashant111/logistic-regression-classifier-tutorial#7.-Exploratory-data-analysis-> [Accessed 5 Jun. 2024].
- Breuss, M. (2023). Python Virtual Environments: A Primer – Real Python. [online] Real Python. Available at: <https://realpython.com/python-virtual-environments-a-primer/#why-do-you-need-virtual-environments> [Accessed 4 Mar. 2024].
- Bridges, J. (2017). How to Make a Timeline - Project Management Training. ProjectManager. Available at: <https://www.youtube.com/watch?v=JnttyHx-sTQ> [Accessed 28 Oct. 2023].
- Chris Stokel-Walker (2023). How to use AI to make your life simpler, cheaper and more productive. [online] New Scientist. Available at: <https://www.newscientist.com/article/2384092-how-to-use-ai-to-make-your-life-simpler-cheaper-and-more-productive/> [Accessed 5 Nov. 2023].
- Computer History Museum (2019). Ada Lovelace | Babbage Engine | Computer History Museum. [online] Computerhistory.org. Available at: <https://www.computerhistory.org/babbage/adalovelace/> [Accessed 14 Jan. 2024].
- Cone, M. (2023). Getting Started | Markdown Guide. [online] www.markdownguide.org. Available at: <https://www.markdownguide.org/getting-started/> [Accessed 7 Mar. 2024].
- Dfrobot.com. (2024). Non-contact Optical Turbidity Sensor/ Transmitter Arduino Wiki - DFRobot. [online] Available at: https://wiki.dfrobot.com/SKU_SEN0554_Turbidity_Sensor [Accessed 16 Aug. 2024].
- Dia, M. (n.d.). A Brief History of AI. [online] www.thats-ai.org. Available at: <https://www.thats-ai.org/en-GB/units/a-brief-history-of-ai> [Accessed 14 Jan. 2024].
- Donges, N. (2021). Gradient Descent: An Introduction to One of Machine Learning's Most Popular Algorithms. [online] Built In. Available at: <https://builtin.com/data-science/gradient-descent> [Accessed 14 Aug. 2024].
- Gadre, V. (2021). Simple Linear Regression Using Python. [online] Geek Culture. Available at: <https://medium.com/geekculture/simple-linear-regression-bd4348e1ee62> [Accessed 3 Mar. 2024].
- GeeksforGeeks (2023). Machine Learning Algorithms. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/machine-learning-algorithms/> [Accessed 15 Aug. 2024].
- Grant Sanderson (2017). But what is a Neural Network? | Deep learning, chapter 1. YouTube. Available at: <https://www.youtube.com/watch?v=aircArvnKk> [Accessed 13 Aug. 2024].
- How to Take Cornell Notes Properly (Video). (2015). Mometrix Test Preparation. Available at: <https://www.youtube.com/watch?v=ErSjc1PEGKE> [Accessed 27 Oct. 2023].
- <https://helpdeskgeek.com/author/bstockton> (2020). What Is An Executable File & How To Create One. [online] Help Desk Geek. Available at: <https://helpdeskgeek.com/how-to/what-is-an-executable-file-how-to-create-one/> [Accessed 6 Mar. 2024].
- IBM (2023). About Linear Regression | IBM. [online] www.ibm.com. Available at: <https://www.ibm.com/topics/linear-regression> [Accessed 13 Aug. 2024].

- Investopedia Team (2022). What is the Difference Between R-Squared and Adjusted R-Squared? [online] Investopedia. Available at: <https://www.investopedia.com/ask/answers/012615/what-difference-between-rsquared-and-adjusted-rsquared.asp> [Accessed 15 Aug. 2024].
- Janakiev, N. (2019). Using Virtual Environments in Jupyter Notebook and Python. [online] janakiev.com. Available at: <https://janakiev.com/blog/jupyter-virtual-envs/> [Accessed 4 Mar. 2024].
- Jason Brownlee (2016). Simple Linear Regression Tutorial for Machine Learning. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/simple-linear-regression-tutorial-for-machine-learning/> [Accessed 5 Jun. 2024].
- Jason Brownlee (2019). Your First Machine Learning Project in Python Step-By-Step. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/machine-learning-in-python-step-by-step/> [Accessed 16 Feb. 2024].
- Jupyter.org. (2019). Project Jupyter. [online] Available at: <https://jupyter.org/install>.
- Khan Academy. (n.d.). Divergence (article) | Khan Academy. [online] Available at: <https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/divergence-and-curl-articles/a/divergence> [Accessed 14 Aug. 2024].
- Machine Learning Works. (n.d.). Multiple Linear Regression. [online] Available at: <https://www.machinelearningworks.com/tutorials/multiple-linear-regression> [Accessed 15 Aug. 2024].
- mindtools.com. (n.d.). 10 Common Time-Management Mistakes. [online] Available at: <https://www.mindtools.com/awmnxfj/10-common-time-management-mistakes> [Accessed 25 Oct. 2023].
- Nadeem (2021). Cost Function & Loss Function. [online] Medium. Available at: <https://nadeemm.medium.com/cost-function-loss-function-c3cab1ddffa4> [Accessed 12 Aug. 2024].
- Oluseye Jeremiah (2023). Slicing and Indexing in Python – Explained with Examples. [online] freeCodeCamp.org. Available at: <https://www.freecodecamp.org/news/slicing-and-indexing-in-python/> [Accessed 12 Aug. 2024].
- Politi, M. (2022). Convolutions in One Dimension using Python. [online] Towards Data Science. Available at: <https://towardsdatascience.com/convolutions-in-one-dimension-using-python-54d743f18063> [Accessed 13 Aug. 2024].
- Prakash, A. (2020). What is a Package Manager in Linux? [online] It's FOSS. Available at: <https://itsfoss.com/package-manager/> [Accessed 4 Mar. 2024].
- Riptutorial.com. (2024). Python Language Tutorial => Read from serial port. [online] Available at: <https://riptutorial.com/python/example/20311/read-from-serial-port#example> [Accessed 17 Aug. 2024].
- Rouse, M. (2012). What is a Compile? - Definition from Techopedia. [online] Techopedia.com. Available at: <https://www.techopedia.com/definition/540/compile> [Accessed 24 Apr. 2024].
- Saha, S. (2018). A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way. [online] Towards Data Science. Available at: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> [Accessed 17 Aug. 2024].
- Sanderson, G. (2022). But what is a convolution? [online] [www.youtube.com](https://www.youtube.com/watch?v=KuXjwB4LzSA). Available at: <https://www.youtube.com/watch?v=KuXjwB4LzSA> [Accessed 12 Aug. 2024].
- Siddiqi (2020). What are Dependencies in Programming. [online] CodersLegacy. Available at: <https://coderslegacy.com/what-are-dependencies-in-programming/> [Accessed 6 Mar. 2024].
- Sparkes, M. (2023). The biggest scientific challenges that AI is already helping to crack. [online] New Scientist. Available at: <https://www.newscientist.com/article/2384085-the-biggest-scientific-challenges-that-ai-is-already-helping-to-crack/> [Accessed 5 Nov. 2023].
- Srinivasan, A. (2020). The first of its kind AI Model- Samuel's Checkers Playing Program. [online] IBM Data Science in Practice. Available at: <https://medium.com/ibm-data-ai/the-first-of-its-kind-ai-model-samuels-checkers-playing-program-1b712fa4ab96> [Accessed 14 Jan. 2024].
- Statistics How To. (2013). Adjusted R2 / Adjusted R-Squared: What is it used for? [online] Available at: <https://www.statisticshowto.com/adjusted-r2/> [Accessed 15 Aug. 2024].
- www.heavy.ai. (n.d.). What is an Open Source Library? Definition and FAQs | HEAVY.AI. [online] Available at: <https://www.heavy.ai/technical-glossary/open-source-library#:~:text=In%20computer%20science%2C%20a%20library> [Accessed 24 Apr. 2024].

[www.markdownguide.org](https://www.markdownguide.org/cheat-sheet/). (n.d.). Markdown Cheat Sheet | Markdown Guide. [online] Available at: <https://www.markdownguide.org/cheat-sheet/> [Accessed 7 Mar. 2024].

www.redhat.com. (2022). What is a configuration file? [online] Available at: <https://www.redhat.com/en/topics/management/what-are-configuration-files>.

[www.upyesp.org](https://www.upyesp.org/posts/makrdown-vscode-math-notation/). (2024). Cheat Sheet: Adding Math Notation To Markdown. [online] Available at: <https://www.upyesp.org/posts/makrdown-vscode-math-notation/> [Accessed 5 Jun. 2024].

[www.youtube.com](https://www.youtube.com/watch?v=DhLu8sI9uY4). (2022). How to Create Python Virtual Environment in Ubuntu Terminal 18.04/20.04 in 2022 || virtualenv (venv). [online] Available at: <https://www.youtube.com/watch?v=DhLu8sI9uY4> [Accessed 4 Mar. 2024].

Yuskaitis, V. (2023). EPQ Launch Webinar.

Zed Shaw (2017). Learn Python 3 the hard way : a very simple introduction to the terrifyingly beautiful world of computers and code. [online] Boston: Addison-Wesley. Available at: https://digitalideation.github.io/digcre_h2101/resources/LearnPython3theHardWay.pdf [Accessed 13 Nov. 2023].

Project bibliography

Saturday, August 31, 2024 7:30 PM

These are all the sources of code that I used for my project. I have also added the links to the sources where they are used in my skills development section. I haven't added these sources to my skills tracker because there is no evaluation necessary for these sources: the code either works for me, or it doesn't. If the code doesn't work, I have not included it.

0sparsh2 (2021). cats-vs-dogs-coursera-assignment/Cats_vs_Dogs_CourseraAssignment.ipynb at main · 0sparsh2/cats-vs-dogs-coursera-assignment. [online] GitHub. Available at: https://github.com/0sparsh2/cats-vs-dogs-coursera-assignment/blob/main/Cats_vs_Dogs_CourseraAssignment.ipynb [Accessed 2024].

Atlassian (2024). How to Save a Plot to a File Using Matplotlib | Atlassian. [online] Atlassian. Available at: <https://www.atlassian.com/data/notebook/how-to-save-a-plot-to-a-file-using-matplotlib> [Accessed 12 Jul. 2024].

Batch renaming files (2010). Batch renaming files. [online] Unix & Linux Stack Exchange. Available at: <https://unix.stackexchange.com/questions/1136/batch-renaming-files> [Accessed 6 Jun. 2024].

Brand, J. (2024). Using ffmpeg to change framerate. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/45462731/using-ffmpeg-to-change-framerate> [Accessed 18 May 2024].

Eric Cabrel Tiogo (2024). How to copy files over SSH - Eric Cabrel Tiogo - Medium. [online] Medium. Available at: <https://medium.com/@tericcabrel/how-to-copy-files-over-ssh-8aff6b7c560f> [Accessed 26 Apr. 2024].

Jevtic, G. (2023). How to Use SSH to Connect to a Remote Server in Linux or Windows. [online] Knowledge Base by phoenixNAP. Available at: https://phoenixnap.com/kb/ssh-to-connect-to-remote-server-linux-or-windows#ftoc_heading_7 [Accessed 25 Apr. 2024].

Riptutorial.com. (2024). Python Language. [online] Available at: <https://riptutorial.com/python/example/20311/read-from-serial-port#example> [Accessed 5 Aug. 2024].

Simone (2017). python - Keras - Plot training, validation and test set accuracy. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/41908379/keras-plot-training-validation-and-test-set-accuracy> [Accessed 13 Jul. 2024].

Singh, G.P. (2022). bing-image-downloader: Python library to download bulk images from Bing.com. [online] PyPI. Available at: <https://pypi.org/project/bing-image-downloader/> [Accessed 20 May 2024].

Take (2012). Take a picture from terminal. [online] Ask Ubuntu. Available at: <https://askubuntu.com/questions/106770/take-a-picture-from-terminal> [Accessed 6 Jul. 2024].

Tclha (2024). Loading TFLite model for Inference (Python). [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/73724481/loading-tflite-model-for-inference-python> [Accessed 18 Jun. 2024].

TensorFlow. (2024). Convert TensorFlow models. [online] Available at: https://www.tensorflow.org/lite/models/convert/convert_models#python_api [Accessed 6 2024].

user714852 (2024). Using PIL to turn a RGB image into a pure black and white image. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/9506841/using-pil-to-turn-a-rgb-image-into-a-pure-black-and-white-image> [Accessed 5 Aug. 2024].

user780756 (2024). Extract all video frames as images with FFMPEG. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/34786669/extract-all-video-frames-as-images-with-ffmpeg> [Accessed 18 May 2024].

Secondary source tracker

01 November 2023 15:48

No.	Source	What is it about, briefly?	Evaluation summary	Issues?
1	Ng, A (2012) Coursera, Machine Learning, available at https://www.coursera.org/learn/machine-learning-course/lecture/Ujm7v/what-is-machine-learning	A course that teaches Machine Learning fundamentals	The course is from Coursera, which is a popular platform which many people use to develop their career, so it is trusted by many. The course itself is from Stanford University. Both of these organisations' purpose is to educate people. Andrew Ng, the instructor, has authored many research papers in machine learning and related fields. I think that I can trust this source because the purpose of this platform is to teach and the instructor has quite a lot of credibility.	The course was released in 2012 and there have been developments in the field since. However, I don't believe that the developments since 2012 have greatly affected the core concepts of machine learning so I think this source will be sufficient to teach me them.
2	Sparkes, M. (2023). <i>The biggest scientific challenges that AI is already helping to crack</i> . [online] New Scientist. Available at: https://www.newscientist.com/article/2384085-the-biggest-scientific-challenges-that-ai-is-already-helping-to-crack/ [Accessed 5 Nov. 2023].	How AI has and is helping with scientific challenges in medicine and climate change	This source definitely has some credibility in that the New Scientist is very well-respected as a reliable source of information. All of the people quoted are researchers at universities. It also contains links to articles which go into more depth into some of the areas that it mentions. The article is also quite recent (4 months ago from the time of writing).	The article's purpose was to inform people of how AI is currently benefitting scientists, which means that it didn't go into much depth in the making of this AI and the projects are likely to be far above my level. Therefore I don't think that it is particularly relevant to my project as I think projects such as these would be far too advanced to do a project in.
3	Chris Stokel-Walker (2023). <i>How to use AI to make your life simpler, cheaper and more productive</i> . [online] New Scientist. Available at: https://www.newscientist.com/article/2384092-how-to-use-ai-to-make-your-life-simpler-cheaper-and-more-productive/ [Accessed 5 Nov. 2023].	Small jobs that AI can do to help us in our lives	See source 2 This source was more useful as it gave some more achievable ideas that I could take inspiration from to do as a project.	It didn't give much detail on how to do these projects as its purpose is to inform people of how they can use AI. As a result, it doesn't tell me about how these algorithms could be made.
4	Anyoha, R. (2017). <i>The History of Artificial Intelligence</i> . [online] Science in the News. Available at: https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/ [Accessed 12 Jan. 2024].	An overview of the development of AI from the mid c19th to 2000s	It provided a good outline of the main events of how AI has developed. This is a blog post from a website called Science in the News, made by Harvard graduates. Since Harvard is a very prestigious university, this gives the source credibility.	The post is from 2018 and there have definitely been significant developments in AI since then. Nonetheless, this has helped give me an idea of the timeline before 2010s. No sources are given, however the author, Rockwell Anyoha, works on machine learning methods according to the website.
5	GeeksforGeeks (2023). Machine Learning Algorithms. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/machine-learning-algorithms/ [Accessed 15 Aug. 2024].	Lists the different types of machine learning, the types within these types, and gives a brief explanation of each	I wanted to know a few examples of different types of ML within the categories of "supervised learning" and "unsupervised learning", which I can later research into. The website is a platform with informative articles and courses on many different areas of computer science, so I think it is a trustworthy resource. The article was published recently.	The author is not shown.
6	Computer History Museum (2019). <i>Ada Lovelace Babbage Engine Computer History Museum</i> . [online] Computerhistory.org. Available at: https://www.computerhistory.org/babbage/ada-lovelace/ [Accessed 14 Jan. 2024].	A brief biography of Ada Lovelace and her contribution to computer science	Since this source is from a museum, its purpose is to provide accurate information about the past. It specialises in computing history, of which Ada Lovelace and the Babbage Machine are an important milestone, so the article is likely to be factually correct. The content is also very relevant to the history of AI.	
7	Srinivasan, A. (2020). <i>The first of its kind AI Model- Samuel's Checkers Playing Program</i> . [online] IBM Data Science in Practice. Available at: https://medium.com/ibm-data-ai/the-first-of-its-kind-ai-model-samuels-checkers-playing-program-1b712fa4ab96 [Accessed 14 Jan. 2024].	An article about a program that played checkers, made in 1959	I believe this is a very forward-looking program in the context of the history of AI so it is important to mention in its timeline. Though posted on Medium, the article comes from IBM Data Science in Practice, which is written by data scientists, some of whom work in IBM. Thus I believe it has authority in the subject of AI.	
8	Zed Shaw (2017). Learn Python 3 the hard way : a very simple introduction to the terrifyingly beautiful world of computers and code. [online] Boston: Addison-Wesley. Available at: https://digitalitation.github.io/digcre_h21_01/resources/LearnPython3theHardWay.pdf [Accessed 13 Nov. 2023]	A book that takes a complete beginner in Python and coding through all the basics	This book is famous for being a great tutorial for Python, and has made Zed Shaw, a software developer, renowned as a result. The book is available online for free, so it was clearly made exclusively for the purpose of educating people. Though the original was made a few years ago, Shaw updated it for Python 3 so it is up to date, and should continue to be until Python is updated.	
9	Yuskaitis, V. (2023). EPQ Launch Webinar.	A webinar that we watched in school that explained about the EPQ	This was definitely relevant, because I watched it before I knew anything about the EPQ. As a result it gave me a good explanation and introduction as to what was happening next. It also gave tips and explained why the EPQ is useful. Victoria Yuskaitis's job is to help students with the EPQ, so she is an authoritative source.	Some of the feedback was quite general and non-specific. It was not all relevant to me because it contained information on doing a dissertation and an event. By this point I was fairly set on doing an artefact.
10	mindtools.com. (n.d.). 10 Common Time-Management Mistakes. [online] Available at: https://www.mindtools.com/awmnxfi/10-common-time-management-mistakes [Accessed 25 Oct. 2023].	An article with 10 common time-management mistakes and how to solve them	Mindtools.com specialises in courses to help with wellbeing and your career. This means it is a trustworthy source for advice on skills such as time-management, which are vital in the workplace. The purpose of the article is to educate. Time-management is also a crucial skill for this project so learning how to avoid common mistakes is relevant.	There is no date or author on the page, which makes it difficult to tell how up-to-date the article is, and whether the author is trustworthy.
11	How To Take Cornell Notes Properly (Video). (2015). Mometrix Test Preparation. Available at: https://www.youtube.com/watch?v=ErSc1PEGKE [Accessed 27 Oct. 2023].	A video explaining how to take Cornell notes	This method seems like a useful and logical way of taking notes, so it's definitely something I will consider using for note-taking. Mometrix is a channel that provides resources for preparing for tests and studying, and the video has been viewed many times, which suggests that it has been used by many as a credible resource. The purpose is to teach viewers how to take Cornell notes.	
12	Bridges, J. (2017). How to Make a Timeline - Project Management Training. ProjectManager. Available at: https://www.youtube.com/watch?v=jNttyHx-stQ [Accessed 28 Oct. 2023].	A video explaining how to make a timeline	The video taught me in a condensed way how to make a timeline, which is something I will have to do multiple times in my project. The video was made by Project Manager, which makes software that helps with project management for business. Thus, the company is an expert source of information on project management and a credible source. They probably earn little money from the YouTube channel so the purpose of the video is mainly to educate.	Some of the guidance was things I already knew so perhaps not a hugely efficient use of my time.
13	Jason Brownlee (2019). Your First Machine Learning Project in Python Step-By-Step. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/machine-learning-in-python-step-by-step/ [Accessed 16 Feb. 2024].	An article taking through the steps of a basic machine learning project	I thought I should be familiar with the basic structure of a machine learning project, so that I can plan ahead for my project and how long any tutorials for example projects might take. Therefore I think this is a relevant piece of research. Having later gone through various tutorials, the steps given in this tutorial match all of them, so I can fairly certainly verify the accuracy of the source. Jason Brownlee has made this website to teach developers about machine learning, and he has many tutorials and articles. Brownlee did a masters and PhD in AI so seems to be an authoritative source.	Since this is a basic structure to cover all ML projects, it is slightly vague/flexible. This means I will need to learn more in depth about different types of projects to understand more.
14	Gadre, V. (2021). Simple Linear Regression Using Python. [online] Geek Culture. Available at:	A tutorial on a simple linear regression algorithm	This was a very simple ML algorithm, so adequate for being my first. It explained each action well so gave me a good understanding of linear regression. All of the code worked as the tutorial promised, so it is an	Since the tutorial relies on libraries, it doesn't feel like I completely understand what the libraries are doing. This is why I am going to try to find a tutorial that does the

	https://medium.com/geekculture/simple-linear-regression-bd4348e1ee62 [Accessed 3 Mar. 2024]		accurate source. The author seems to be knowledgeable in this field since he is said to be a data scientist and ML engineer. The source is recent.	same, from scratch, without libraries.
15	www.youtube.com . (2022). How to Create Python Virtual Environment in Ubuntu Terminal 18.04/20.04 in 2022 virtualenv (venv). [online] Available at: https://www.youtube.com/watch?v=DhLusI9uY4 [Accessed 4 Mar. 2024].	Shows what commands I need to use to create and activate a virtual environment	Virtual environments are crucial in any programming project, not just ML, so this is not only relevant to my project but also for any programming that I do. The source was accurate because using the commands that it told me, I made a virtual environment. It was published recently.	
16	Jupyter.org. (2019). Project Jupyter. [online] Available at: https://jupyter.org/install .	How to open and install Jupyter Notebook	Since I am using Jupyter Notebook for tutorials and possibly my project, I need to know how to open it, for which this page has been useful. The Jupyter website is probably the most authoritative source to tell me this. Though published a few years ago, the source is still accurate because it worked when I tried to do what the web page told me. The purpose was to instruct. Jupyter is an open-source project so is made purely for the benefit of developers.	It didn't tell me how to use a virtual environment in Jupyter Notebook, but I have found another source that shows me this.
17	Janakiev, N. (2019). Using Virtual Environments in Jupyter Notebook and Python. [online] janakiev.com. Available at: https://janakiev.com/blog/jupyter-virtual-envs/ [Accessed 4 Mar. 2024].	A tutorial on how to use a virtual environment in Jupyter Notebook	Janakiev's website contains many blog posts on data science and machine learning, so he seems to be an expert in this field. Though not particularly recent, the article tells me how to use a virtual environment in Jupyter Notebook, and it worked for me so the source is accurate. I need to do this if I want to be within a virtual environment while using Jupyter Notebook, which is what I will be doing when doing my project.	
18	Breuss, M. (2023). Python Virtual Environments: A Primer – Real Python. [online] Real Python. Available at: https://realpython.com/python-virtual-environments-a-primer/#why-do-you-need-virtual-environments [Accessed 4 Mar. 2024].	An article showing (seemingly) everything you need to know about virtual environments: how to use, why, how it works, any alternatives	I wanted to understand a bit about what virtual environments are and why I need them (until now I just accepted that I need them for all projects). After reading this, I have grasped what they do and their importance. Real Python contains a host of tutorials, articles, chat rooms, etc so seems to be a valid source of information about Python-related topics. Thus I can infer that the purpose of the article is to inform. As well as the author, the article – like all others in Real Python – was made by a team of developers so it is likely to be accurate and fact-checked.	Some parts went too far in depth for my understanding and needs, so I had to assess which parts I needed and understand these as much as I could.
19	Prakash, A. (2020). What is a Package Manager in Linux? [online] It's FOSS. Available at: https://itsfoss.com/package-manager/ [Accessed 4 Mar. 2024].	Explains what a package manager is in simple terms	Package managers are a piece of jargon that I have seen a lot but I didn't fully understand, so this was a simple explanation that told me all I needed to know. It's FOSS contains blog posts and tutorials about Linux, of which a package manager is a feature, so it's within the realm of knowledge of this website. The author contains years of experience in IT. This was not posted recently, but I can be fairly sure that the basic concept of package managers has stayed the same since. The purpose is probably to inform.	The explanation still contained some jargon so I had to look up the definition of these from other sources.
20	www.redhat.com . (2022). What is a configuration file? [online] Available at: https://www.redhat.com/en/topics/management/what-are-configuration-files .	Explains what a configuration file is	Red Hat is a company providing open source software solutions, of which configuration files are an integral part of software. I needed only a simple explanation that I could understand of what a configuration file is, which this blog post provided. It was published recently. Therefore, I am confident in the accuracy of this source.	When I later clicked on the link, the web page was not found, so the post seems to have been deleted. This suggests there was a problem with the post.
21	Stockton, B. (2020). What Is An Executable File & How To Create One. [online] Help Desk Geek. Available at: https://helpdeskgeek.com/how-to/what-is-an-executable-file-how-to-create-one/ [Accessed 6 Mar. 2024].	Explains what an executable file is and does	Another piece of jargon I wasn't sure about, which appears a lot in software, so it was important to understand what it is. Help Desk Geek is a website that is dedicated to "explaining tech in an easy-to-understand way" which is what I needed, and tells me that the purpose of the article is to inform. The author has a background in computing and writing, so is qualified to explain it. Again, though the article isn't recent, this is a fundamental in computing software so it is recent enough to not affect the reliability of the source.	
22	Siddiqi (2020). What are Dependencies in Programming. [online] CodersLegacy. Available at: https://coderslegacy.com/what-are-dependencies-in-programming/ [Accessed 6 Mar. 2024].	What dependencies are in programming	This is a concept that is relevant to software and also the purpose of virtual environments, so it links strongly to my project. CodersLegacy is a coding site for new programmers (which fits my description) so it seems that the purpose of the articles on the site are to educate.	The author uses a pen-name and contains no description that I can find so I don't know whether they are a credible source of information.
23	Cone, M. (2023). Getting Started / Markdown Guide. [online] www.markdownguide.org . Available at: https://www.markdownguide.org/getting-started/ [Accessed 7 Mar. 2024].	An introduction to Markdown	Markdown is used for many purposes, but the ones which are relevant to my project are for blocks of text on Jupyter Notebook, and readme files on GitHub. Thus, learning about Markdown is relevant for me. The website is dedicated to resources on Markdown so I believe it is an accurate source for learning about Markdown.	
24	www.markdownguide.org . (n.d.). Markdown Cheat Sheet / Markdown Guide. [online] Available at: https://www.markdownguide.org/cheat-sheet [Accessed 7 Mar. 2024].	Basic syntax on Markdown	To use Markdown I need to know the syntax so this cheat sheet told me all of what I needed to know, which I have used for many of my Jupyter Notebook files. The source is accurate because I verified what it told me by testing the display on Jupyter.	
25	www.heavy.ai . (n.d.). What is an Open Source Library? Definition and FAQs / HEAVY.AI. [online] Available at: https://www.heavy.ai/technical-glossary/open-source-library#:~:text=In%20computer%20science%2C%20a%20library [Accessed 24 Apr. 2024].	Explains what programming libraries are and do	This gave me a brief explanation of a library, which I wanted to know since they are used in virtually every programming project. The source is a blog post on a website of a company that specialises in Big Data, which means they probably use this article to educate readers and then raise awareness about their company.	The author and date published is not shown.
26	Rouse, M. (2012). What is a Compile? - Definition from Techopedia. [online] Techopedia.com. Available at: https://www.techopedia.com/definition/540/compile [Accessed 24 Apr. 2024].	What compile means in a software context	This term was used in the definition of a library, so to fully understand this I had to understand what to compile is. The source explained this in a way that I understood. The author is an award-winning technical writer, which means she can explain things well. The purpose of Techopedia is to provide technical definitions factually, which shows that they care about accuracy.	It hasn't been updated since 2012, but I believe the definition hasn't changed since either.
27	Amos, D. (2023). Object-Oriented programming (OOP) in python 3 – real python. [online] realpython.com. Available at: https://realpython.com/python3-object-oriented-programming/ [Accessed 24 Apr. 2024].	What object-oriented programming is in Python, and how to use it	This subject was touched upon at the end of Learn Python the Hard Way, so I thought I should do more research into it. However, I have not used it in the making of my project. Real Python contains a host of tutorials, articles, chat rooms, etc so seems to be a valid source of information about Python-related topics. Thus I can infer that the purpose of the article is to inform. As well as the author, the article – like all others in Real Python – was made by a team of developers so it is likely to be accurate and fact-checked. The author is a programmer, writer and mathematician.	This article went quite in depth, which I didn't need, so I only used it to give a brief outline of the concept.
28	Jason Brownlee (2016). Simple Linear Regression Tutorial for Machine Learning. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/simple-linear-regression-tutorial-for-machine-learning/ [Accessed 5 Jun. 2024].	A linear regression tutorial without using libraries	This tutorial gave me a deeper understanding of what goes on in a linear regression algorithm. Jason Brownlee has made this website to teach developers about machine learning, and he has many tutorials and articles. Brownlee did a masters and PhD in AI so seems to be an authoritative source. The source seems accurate because all the operations that the tutorial instructed gave me the right answer.	
29	Banerjee, P. (2020). Logistic Regression Classifier Tutorial. [online] kaggle.com.	Tutorial on a logistic regression algorithm, a	This tutorial was long and did have a lot of data handling, which made it feel like there was too much of it and not enough of the actual machine learning	The tutorial could have been posted by anyone. Nonetheless, as I mentioned, it is very easy to verify the

	Available at: https://www.kaggle.com/code/prashant11/logistic-regression-classifier-tutorial#7-Exploratory-data-analysis [Accessed 5 Jun. 2024].	type of classification algorithm	part. But then I realised that the training of the model actually requires virtually no work, and the data handling is a very important part. So the tutorial was very instructive. I trust this source because it is very easy to validate whether the code in the tutorial is right, and the vast majority of it worked for me. The purpose of the source is to educate because it is a tutorial. Kaggle is a competition platform + online data science community, so I believe it is unlikely that it has a hidden agenda.	accuracy because the code worked.
30	Alake, R. (2023). Loss Functions in Machine Learning Explained. [online] Datacamp. Available at: https://www.datacamp.com/tutorial/loss-function-in-machine-learning?dc_refferrer=https%3A%2F%2Fwww.google.com%2F [Accessed 11 Aug. 2024].	What a loss function is, different types and how they are used	The source gave me a good overview of the loss function and was easy to understand, which is what I needed. Published less than a year ago, it appears recent and the website seems well maintained. The author is said to be a Machine Learning architect, so is probably an authoritative source of information on machine learning. The website is a well-known company that provides courses in data science, so its purpose is to educate. Thus, I believe I can trust this resource.	
31	Nadeem (2021). Cost Function & Loss Function. [online] Medium. Available at: https://nadeem.medium.com/cost-function-loss-function-c3cab1ddffa4 [Accessed 12 Aug. 2024].	Explanation of what a cost function is and what a loss function is	The author "Nadeem" is said to be a senior data scientist and AI researcher, which means they are probably a reliable source for data science and machine learning. It told me clearly the difference between loss and cost function, which is what I was researching. The article confirms what I have researched elsewhere so is probably accurate. Since it is a Medium post, the purpose is to inform and educate. It is also relatively recent and it is about fundamental concepts in machine learning which are unlikely to have changed in the past 2 years.	I must proceed with caution with Medium posts because they could be written by anyone. Additionally, only the author's first name is shown which means they are almost anonymous and takes away from their authority.
32	Sanderson, G. (2022). But what is a convolution? [online] www.youtube.com . Available at: https://www.youtube.com/watch?v=KuXjwB4LzSA [Accessed 12 Aug. 2024].	Explains what a convolution algorithm is and its applications, such as in image processing	3Blue1Brown is a famous YouTube channel with over 6 million subscribers. It contains videos on maths, physics, computer science and data science. The author, Grant Sanderson, has a maths degree from Stanford University, and taught for Khan Academy, another credible institution for education. The video was comprehensive and with good visuals which helped me understand fully what a convolution algorithm does. It was published recently, and with the purpose of educating a wide audience. In the description, Sanderson referenced other videos and also wrote minor corrections, which shows that he cares about the accuracy of the video.	YouTube videos are also made for entertainment so it's possible that, especially in one with such a large audience, some information is sacrificed for the sake of brevity of the video and / or because it wouldn't engage viewers. 3Blue1Brown is also sponsored by companies as well as individuals, so he may be influenced by them since he requires their support.
33	Oluseye Jeremiah (2023). Slicing and Indexing in Python – Explained with Examples. [online] freeCodeCamp.org. Available at: https://www.freecodecamp.org/news/slicing-and-indexing-in-python/ [Accessed 12 Aug. 2024].	Shows how to manipulate lists in Python	This article is very recent, and lists are a well-established concept in programming so are unlikely to have changed in Python recently. It told me exactly what I needed about lists, which are not only necessary in programming but also in data manipulation, which is a big part of machine learning. The author's description is Data Scientist and Machine Learning Engineer, which is also relevant and gives him authority. FreeCodeCamp is a charity that pledges to teach coding for free, so its purpose is to educate. Copying and running the code that the article used as examples, they all worked as it said they would, thus I think the source can be deemed trustworthy.	
34	Politi, M. (2022). Convolutions in One Dimension using Python. [online] Towards Data Science. Available at: https://towardsdatascience.com/convolutions-in-one-dimension-using-python-54d743f18063 [Accessed 13 Aug. 2024].	Explains what a convolution algorithm is and does, and shows you how to make one in Python	Towards Data Science is a group that posts articles on Medium about Data Science. They seem to be funded by donations and Medium, so probably have no hidden agenda, and say that they aim to well-written, informative articles on data science. There are many different authors that submit articles for them to publish, and Towards Data Science will select a few out of these. Therefore this article is probably accurate since it was deemed worthy of publication by Towards Data Science. The article gave me a practical understanding of convolutions, which I believe is important since they are vital for Convolutional Neural Networks (what I am working on).	
35	Grant Sanderson (2017). But what is a Neural Network? Deep learning, chapter 1. YouTube. Available at: https://www.youtube.com/watch?v=aircArувnKk [Accessed 13 Aug. 2024].	Explains the maths behind a neural network and how it works	3Blue1Brown is a famous YouTube channel with over 6 million subscribers. It contains videos on maths, physics, computer science and data science. The author, Grant Sanderson, has a maths degree from Stanford University, and taught for Khan Academy, another credible institution for education. The source provided me with a basic understanding of what a neural network is, which I believe is what I need to use one, because if I go too deep I think it might be too complex for my understanding.	YouTube videos are also made for entertainment so it's possible that, especially in one with such a large audience, some information is sacrificed for the sake of brevity of the video and / or because it wouldn't engage viewers. 3Blue1Brown is also sponsored by companies as well as individuals, so he may be influenced by them since he requires their support.
36	Donges, N. (2021). Gradient Descent: An Introduction to One of Machine Learning's Most Popular Algorithms. [online] Built In. Available at: https://builtin.com/data-science/gradient-descent [Accessed 14 Aug. 2024].	Explains what gradient descent is, graphically and mathematically, and how learning rate affects it	This article explained gradient descent, a vital part of the "learning" step in machine learning, in a comprehensive way, so this knowledge will be very useful to my project. The author specialises in machine learning and software so this is his field. Built In is a "community for online startup and tech companies", so the purpose of this article may have been to raise the author's credibility while teaching others. This means I can probably trust the source to be accurate.	
37	Khan Academy. (n.d.). Divergence (article) Khan Academy. [online] Available at: https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/divergence-and-curl/articles/a/divergence [Accessed 14 Aug. 2024].	Explains divergence, a concept in vector calculus	Khan Academy is a widely used not-for-profit education platform, and the articles are probably made for the purpose of educating. Furthermore, it has a good reputation at teaching calculus. This article told me what I need to know (which was not a lot) about the uses of divergence.	There are no sources and no authors.
38	Statistics How To. (2013). Adjusted R2 / Adjusted R-Squared: What is it used for? [online] Available at: https://www.statisticshowto.com/adjusted-r2/ [Accessed 15 Aug. 2024].	Explains what R-squared and adjusted R-squared are, and the difference between them	I wanted to understand what both of these are and the difference between them, since they are used in linear regression. This article helped me understand them and also gave the equation. Statistics How To is a website that teaches statistics, and it advertises a book that was probably written by one of the website authors. So, I believe that the website and this article were made to educate. The source seems accurate because it contains references.	
39	BM (2023). About Linear Regression IBM. [online] www.ibm.com . Available at: https://www.ibm.com/topics/linear-regression [Accessed 13 Aug. 2024].	Explains what linear regression is, its uses, assumptions it makes, and examples	IBM is a famous technology company which sells AI tools, amongst other things, so is knowledgeable in this field. The company most likely wishes to make money but it's not clear how they could do so directly from this article by educating people, so the purpose of the article is probably to give people a positive image of the company by trying to educate people. Thus, it probably is not trying to bias people. The content is relevant because this could be the algorithm with which I choose to do my project, so I need to learn more about linear regression.	
40	Machine Learning Works. (n.d.). Multiple Linear Regression. [online] Available at: https://www.machinelearningworks.com/tutorials/multiple-linear-regression [Accessed 15 Aug. 2024].	A tutorial on multiple linear regression	This website is dedicated to teaching about ML, which gives it some credibility. It took me through the steps, including code, of a multiple linear regression algorithm, and all of the code worked. This is what I was looking for, and the only requirement for it to be accurate was for the output in the article to match what my output was. Thus, despite perhaps lacking in credibility, the source was accurate.	There is no date and no authors. When I visited the site a few months later, it was for sale so can no longer be accessed.
41	Dfrobot.com. (2024). Non-contact Optical Turbidity Sensor/ Transmitter Arduino Wiki - DFRobot. [online] Available at: https://wiki.dfrobot.com/SKU_SEN0554	Instructions on how to use the turbidity sensor I purchased	I wanted to know a way of using a turbidity sensor and this told me the software I had to use (Arduino) and some sample code to use it. This is the instructions for the turbidity sensor I have (Non-Contact Optical Turbidity Sensor), so the code is likely to be reliable. The article was posted recently.	There are no authors, and the URL has "wiki" so it was probably made collaboratively by volunteers.

	Turbidity_Sensor [Accessed 16 Aug. 2024].		
42	Saha, S. (2018). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. [online] Towards Data Science. Available at: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 [Accessed 17 Aug. 2024].	How convolutional neural networks work	I already knew what a convolution algorithm works but I wanted to know how it was used in neural networks, which this article told me. Towards Data Science is a group that posts articles on Medium about Data Science. They seem to be funded by donations and Medium, so probably have no hidden agenda, and say that they aim to well-written, informative articles on data science. There are many different authors that submit articles for them to publish, and Towards Data Science will select a few out of these. Therefore this article is probably accurate since it was deemed worthy of publication by Towards Data Science.
43	Riptutorial.com. (2024). Python Language Tutorial => Read from serial port. [online] Available at: https://riptutorial.com/python/example/20311/read-from-serial-port#example [Accessed 17 Aug. 2024].	Python commands needed to read from a serial port	This tutorial told me all the commands I needed to write the code to read from the serial monitor. Since it is a tutorial, its purpose is to educate. It's difficult to tell if the website is reliable, however, my code worked when I used the commands that this tutorial told me to, so I've verified that it is accurate.
44	Osparsh2 (2021). cats-vs-dogs-coursera-assignment/Cats_vs_Dogs_CourseraAssignment.ipynb. [online] GitHub. Available at: https://github.com/Osparsh2/cats-vs-dogs-coursera-assignment/blob/main/Cats_vs_Dogs_CourseraAssignment.ipynb [Accessed 5 Aug. 2024].	Code from a ML course on coursera, posted on GitHub	Since the code comes from a ML course on Coursera, it should be relatively simple since it was made as an example exercise, and the code should work. But it was posted by an anonymous user on GitHub so there may be things missing. Nonetheless, this is the project I need because it is very similar to the neural network I want to construct.

Primary source: Rob Reed

01 November 2023 15:46

Rob has been the main inspiration for my choice of project, because he collects data on the water quality of a river he lives near, and he believes that there is a correlation between water quality and the presence of cows. So I spoke to him about why he thinks this, the river, and what he does.

I've highlighted the particularly relevant parts for me in yellow.

What is citizen science and what does it entail?

A member of the general public who collects and analyses data relating to the natural world, typically as part of a collaborative project with professional scientists.

In what way are you and/or have you been a citizen scientist?

I work with the Westcountry Rivers Trust (founding organisation The Rivers Trust). I record each month water quality data (temp, turbidity, conductivity (as a measure of total dissolved solids) phosphate. The trust monitors water quality and performance of Environment Agency (EA) and impact or otherwise of legislation.

Riverfly monitor, surveying the number and type of riverfly larvae as an indication of long term trends and short term pollution effects. Pollution events are reported to the EA.

I also record Devon moths. I note: numbers, species. This then is part of Devon Biodiversity Record Centre, used to assess changes in Biodiversity. Hence Devon Moths is in effect another Citizen Science organisation.

How do you think that you are making a positive impact?

As part of the greater assessment of water quality, yes. I've had no direct pollution events other than extensive dredging upstream which release huge quantities of sediment over a week or so. I reported this to Devon Council who took action.

In what ways can others get involved in citizen science?

Have a look at the web site www.ceh.ac.uk and search citizen science.

Why is the river important to you?

The pleasure of getting involved in the minutia of its wildlife, insect, birdlife and of course fishing. It was mainly fishing that fired my interest in insects.

When did you start monitoring the river?

Back in 2015.

What have you noticed from your data?

The water quality in the river is fairly stable. The locals claim less fish life in the river (a local guy who lived at Edge Mill, on the river (look at a map), remembered his father/uncle shooting salmon when the family were hungry).

- This could be due to farming/domestic inputs (over the past 10 years much of the sewage input into the river is now treated)
- Farming load from herbicides and concentration of cattle has increased over the past few decades
- Could just be that predation by otters and water fowl has increased as hunting is no longer practiced (we can't have it all ways!!)

What is the main problem with your method of collecting data at the moment?

Sampling has to be random to avoid denuding a sample site but does lead to difficulty in selecting representative/repeatable locations

Ammonia kits and some form of organic load monitoring (the latter would be expensive) would be handy to measure the instantaneous de-oxygenation capacity of cattle.

It should be noted that I have found no correlation between presence of cattle on water quality but long term measurement is not possible without decades of data.

What are the effects of the cows on the river?

- Release of sediment can stifle river insect life downstream due to smothering of algae (major food source for some larvae) and reducing oxygen diffusion into the river bed.
- Organic load removing oxygen.
- Faecal matter acting as a sediment with adverse impact noted above.

Programming research checklist

Monday, March 04, 2024 10:26 PM

- virtual environment
 - how to install
 - what is it
- jupyter notebook
 - how to install
- markdown
- package manager (pip)
- object oriented programming
- Libraries
- Shell
- args & kwargs
- List comprehension

Glossary

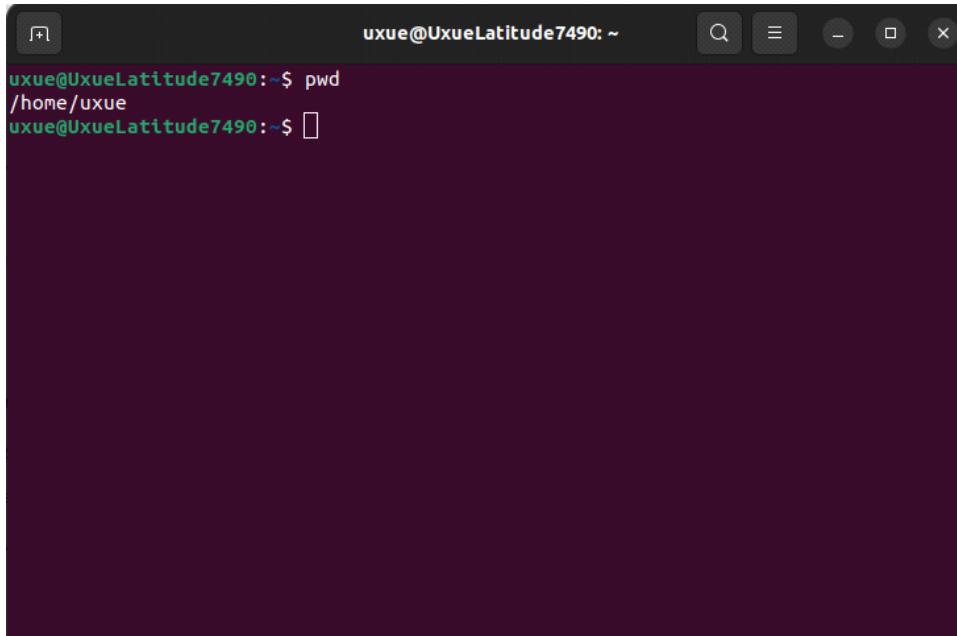
12 June 2024 14:58

Shell [8]

This interprets commands that I type into the computer. Some simple commands include:

pwd print working directory
hostname my computer's network name
mkdir make directory
cd change directory
ls list directory
rmdir remove directory
pushd push directory
popd pop directory
cp copy a file or directory
mv move a file or directory
less page through a file
cat print the whole file
xargs execute arguments
find find files
grep find things inside files
man read a manual page
apropos find which man page is appropriate
env look at your environment
echo print some arguments
export export/set a new environment variable
exit exit the shell
sudo become super user root

Throughout the project I use it to navigate through my computer's file system, install packages onto my computer, and more. The shell, or the terminal, looks like this:



```
uxue@UxueLatitude7490:~$ pwd
/home/uxue
uxue@UxueLatitude7490:~$
```

where **uxue@UxueLatitude7490:~\$** is the prompt, **pwd** is the command, and **/home/uxue** is the output. Sometimes a lot of the output is too much for me to understand, but I have highlighted the important parts in yellow where relevant.

Package manager

What is it?

A tool that allows users to install, remove, upgrade, configure and manage software packages.

What is a package?

A software package is an application, command line tool or software library. It is an archive file which contains the set of instructions to perform tasks [22], a file used to configure the initial settings [21], and sometimes information about dependencies (when one piece of software relies on another [23]).

Why do we need package managers?

Without package managers, the user has to refer to a file and see what software components the application needs, compile the software and handle its dependencies. Linux distributions created a packaging format to provide users with ready-to-use precompiled software. In other words, using packaging managers is like baking a cake vs buying a cake.

Virtual environments [19]

Why are they needed?

- Avoid system pollution
 - Can install packages for your project that mix with packages from the system - could affect system's behaviour
 - If operating system is updated, installed packages may get overwritten & lost
- Dependency conflicts
 - Different projects may need different library versions. With one place to install libraries, you can't use multiple versions
 - Virtual environments keep libraries separate for each project

- Minimise reproducibility issues
 - If others want to replicate your project, it is easy to see what dependencies your project has if the dependencies are contained in a virtual environment
- Overcome administrative constraints
 - Administrative privileges may be necessary to install some packages on a computer
 - A new installation location can be used in a virtual environment to work with external packages

What are they?

A virtual environment is a folder structure that gives you everything you need to run a lightweight but isolated Python environment. It reproduces the folder structure that a standard Python installation creates.

Jupyter notebook [17] [18]

To install:

```
pip install jupyter notebook
```

To open: [17]

```
jupyter notebook
```

To add virtual environment to Jupyter Notebook: [18]

```
python -m ipykernel install --user --name=myenv
```

Markdown [24] [25]

Markdown serves as a simple markup language enabling the addition of formatting elements to plain text documents.

Why is it used:

- Portable - files with Markdown-formatted text can be opened with almost any application
- Platform independent - can create on any device and any OS
- Future-proof - if the application used stops working, you can still read Markdown text in any other application
- Used very widely - e.g. sites such as Reddit and GitHub [24]

Each cell in Jupyter Notebook can be code, or Markdown, so it is useful for me to understand how to use Markdown if I want to write a block of text in Jupyter Notebook.

Cheat sheet for basic syntax: [25]

Element	Markdown Syntax
<u>Heading</u>	# H1 ## H2 ### H3
<u>Bold</u>	**bold text**
<u>Italic</u>	<i>*italicized text*</i>
<u>Blockquote</u>	> blockquote

Ordered List	1. First item 2. Second item 3. Third item
Unordered List	- First item - Second item - Third item
Code	`code`
Horizontal Rule	---
Link	[title](https://www.example.com)
Image	![alt text](image.jpg)

Libraries [26]

Libraries are a set of resources that are precompiled, so that other computer programmers can use them. These include reusable scripts, functions, and files.

- **Compiling** means when code that humans can read i.e. in programming languages such as Python and JavaScript, is translated into code that a machine can read [27]

Object-oriented programming [28]

Object-oriented programming is an approach to programming that structures a program into different objects, which each has a set of properties and behaviours. Each object contains data and behaviour. A class is like a blueprint for creating a project.

Python definitions [8]

Friday, February 16, 2024 11:09 AM

```
print() # repeats what is inside the brackets
# octothorpe/hash: to write comments. Text after the hash is ignored
+, -, /, *, >, <, >=, <=
= # makes the left of = sign (variable) equal the right
(f"") # allows you to write variables in the quotation marks, with {}
.format # another way of using f"" with a variable before the .
# string =
"\n" # new line
"""
""" # type anything (including the enter key) between
"\t" # indent
"\\" # one backslash
...
''' # like three quotation marks
input() # what you type into the program then press enter
from "?" import "?" # some new commands!
argv # argument variable: what is typed after "python script.py" - need to type
"from sys import argv" to use it
open() # opens the file inside the brackets
.read() # reads the contents of the file (preceded by open(filename))
open(filename, 'w') # opens file to write in
.truncate() # deletes all text in the file
.write() # writes what is in the brackets in the file
.close() # closes the file
len() # length in bytes of the file
exists() # whether the file exists (true or false) - need to type from "os.path"
import exists" first
def function(): # function: the commands below (with an indent) say what the
function does
f # in def function(f), f.seeks, etc. means file
f.seek() # seek moves the position of the File Handle (like a cursor) - the
number in the brackets is bytes. So f.seek(0) means to the start of the file
return # works out the answer e.g. in return a + b
```

KEYWORDS

```
assert = ensure something is true
    assert False, "Error!"
break = stop loop now
    while True: break
continue = don't process more of loop, do it again
    while True: continue
del = delete from dictionary
    del X(Y)
except = if exception, do this
    except ValueError, e: print(e)
exec = run string as Python
    exec 'print("hello")'
finally = regardless of exceptions, do this
    finally: pass
global = declare you want a global variable
is = like == to test equality
    l is l == True
lambda = create short anonymous function
    s = lambda y: y ** y; s(3)
pass = this block is empty
    def empty(): pass
raise = raise exception when things go wrong
    raise ValueError("No")
try = try this block; if exception, go to except
    try: pass
with = with an expression as a variable do
    with X as Y: pass
yield = pause here and return to caller
    def X() yield Y; X().next()
```

DATA TYPES

```
None = nothing or no value
bytes = stores bytes of text, PNG, file, etc.
    x = b"hello"
floats = stores decimals
dicts = stores a key=value mapping of things
    e = {'x': 1, 'y': 2}
```

STRING ESCAPE SEQUENCES

```
\a = Bell
\b = Backspace
\f = Formfeed
\r = Carriage
\v = Vertical tab
```

OLD STYLE STRING FORMATS

```
%d = decimal integers (not floating point)
    "%d" % 45 == '45'
%i = %d
%o = octal number
    "%o" % 1000 == '1750'
%u = unsigned number
    "%u" % -1000 == '-1000'
%x = hexadecimal lowercase
    "%x" % 1000 == 3e8
%X = hexadecimal uppercase
```

```
%e = exponential notation, lowercase e
    "%e" % 1000 == '1.000000e+03'
%E = exponential notation, uppercase E
%f = floating point real number
%F = %F
%g = whichever of %f or %e is shorter
%G = uppercase %F
%c = character format
    "%c" % 34 == ''
%r = repr format (debugging format)
    "%r" % int == "<type 'int'>"
%s = string format
    "%s there" % 'hi' == 'hi there'
%% = %
```

OPERATORS

```
** = power of
// = floor division
% = string interpolate or modulus
@ = at (decorators)
= = assign
```

```
class = make new type of thing
object = most basic type of thing
        = any instance of a thing
instance = what you get when you tell Python to create a class
self = inside class function, a variable for instance/object being tested
inheritance = one class inherits traits from another
composition = class is composed of other classes as parts
attribute = property of classes that are from composition, usually variables
is-a = phrase to say something inherits from another thing
has-a = phrase to say something is composed from another thing

class X(Y) "make class called X that is-a Y"
class X(object): def __init__(self, J) "class X has-a __init__ that takes self
and J parameters"
class X(object): def M(self, J)
```

Lists [34]

Monday, August 12, 2024 3:19 PM

How to Extract Substrings from a String

```
In [1]: sentence = "The quick brown fox jumps over a lazy dog"
first_3_letters = sentence[:3]
# remember that indexing of lists starts from 0,
# so 3 means 4th character!
# all characters BEFORE (not including) 4th character

print(first_3_letters)
```

The

```
In [2]: letters_five_to_nine = sentence[4:9]
# characters five to nine,
# or index 4 to 8 (not including tenth character / index 9)

print(letters_five_to_nine)
```

quick

How to Filter a List

```
In [3]: numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
every_other_number = numbers[::2]
# [::2] is the same as writing [0::2].
# Means starting from index 0, and moving by 2 each time

print(every_other_number)
```

[1, 3, 5, 7, 9]

```
In [4]: every_other_starting_from_two = numbers[1::2]

print(every_other_starting_from_two)
```

[2, 4, 6, 8]

How to Extract Columns from a 2D List

```
In [5]: two_dimensions = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
column = [row[1] for row in two_dimensions]
print(column)
```

[2, 5, 8]

In other words, two_dimensions list looks like

[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

and that's why column 1 (the second column) is [2, 5, 8]

```
In [6]: column_0 = [row[0] for row in two_dimensions]
column_2 = [row[2] for row in two_dimensions]
```

```
print("First column: ", column_0)
print("Third column: ", column_2)
```

First column: [1, 4, 7]
Third column: [3, 6, 9]

How to Modify Parts of a List

```
In [7]: numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
numbers[1:4] = [10, 20, 30]
# replacing 2nd, 3rd, 4th term

print(numbers)
[1, 10, 20, 30, 5, 6, 7, 8, 9]

In [8]: numbers[4:4] = [40, 50]
# adds 40 and 50 at 5th term

print(numbers)
```

[1, 10, 20, 30, 40, 50, 5, 6, 7, 8, 9]

Machine learning

Monday, March 04, 2024 10:32 PM

- basic steps of machine learning project
- what is linear regression
- cost function
- residual-sum-square
- gradient descent**
- mean square error
- Loss function
- List comprehension
- Neural network

Scientific breakthroughs by AI

Sunday, November 05, 2023 6:13 PM

SCIENTIFIC BREAKTHROUGHS [2]

- Future medicines
 - Protein structure
 - Prediction
 - Mapping protein dynamics
 - Design
 - Understanding effect of mutations
 - Drug development
- Climate change
 - Energy efficiency
 - Improving computing tasks
 - Matrix multiplication
 - Sorting algorithms
 - Process to make concrete
 - Aerial for satellite design
 - Nuclear fusion reactor

MAKING LIFE EASIER [3]

- Summarising media
 - YouTube and podcasts
 - Transcriptions of content
- Menu/recipe generator
 - Generate menu from leftovers
 - Dinner party menu
- Fitness apps
 - Creates an exercise routine
 - Compares fitness data to others and creates personalised training plan
- Education
 - Tutoring bots
 - Quiz generator
- Holiday planner

History of AI [4] [5]

10 January 2024 15:35

- Pre c19th
 - 1837: Charles Babbage designed Analytical Machine
 - 1843: Ada Lovelace published translation with notes of her own of an article about Analytical Machine [6]
- First half of c19th
 - Portrayals of AI in science fiction
 - Metropolis (1927) depicts humanoid robot that impersonates a human
 - The Tin Man in The Wizard of Oz (1939)
- 1950s
 - Alan Turing's 1950 paper: Computing Machinery and Intelligence
 - BUT computers were still primitive and expensive
 - Leasing a computer was \$200 000 per month in today's money
 - 1951: Theseus, a mouse that could navigate its way through a maze and remember its path
 - 1955: *Dartmouth Summer Research Project on Artificial Intelligence* (DSRPAI)
 - Allen Newell, Cliff Shaw, and Herbert Simon presented Logic Theorist
 - Failure to agree with standard methods in AI but agreed that AI is achievable
 - Term "artificial intelligence" coined by John McCarthy, and other terms that are still used today
 - 1957: Newell and Simon's General Problem Solver
 - 1959: Arthur Samuel created checkers-playing program [7]
 - Used concept known as "reinforcement learning". He also coined the term "machine learning"
- 1960s
 - 1964-1967: Joseph Weizenbaum's ELIZA
 - 1963: Defense Advanced Research Projects Agency (DARPA) fund AI at MIT
- 1970s
 - Funding dwindled as it was realised that there was a lack of computational power to do anything substantial
- 1980s
 - 1982: John Hopfield & David Rumelhart popularised "deep learning" techniques
 - Edward Feigenbaum developed "expert systems"
 - 1982-1990: Japanese government heavily funded AI-related endeavours in Fifth Generation Computer project
 - \$400 million dollars with the goals of revolutionizing computer processing, implementing logic programming, and improving artificial intelligence
 - However, most goals not met, funding ceased, and AI fell out of the limelight
- 1990s & 2000s
 - Despite no government funding and public hype, many benchmarks of AI achieved
 - 1997: Gary Kasparov beaten by IBM's Deep Blue
 - 1997: speech recognition software by Dragon Systems implemented on Windows
 - 2001: Cynthia Breazeal developed Kismet, a robot that could recognise and display emotions

About machine learning [1]

04 November 2023 11:41

Definition

- Arthur Samuel (1959) - field of study that gives computers the ability to learn without being explicitly programmed
- Tom Mitchell (1998): well-posed learning problem - computer program is said to learn from experience (E) with respect to some task (T) and performance measure (P), if performance on T measured by P improves with experience E
 - E.g. Playing checkers:
 - E = experience of playing games of checkers
 - T = task of playing checkers
 - P = probability that program will win next game

Types of machine learning algorithms

- Supervised learning - we teach computer
- Unsupervised learning - computer learns by itself
- Others: reinforcement learning, recommender systems

Supervised learning

E.g. House price prediction: we give computer a graph, it creates a line of best fit and can predict the price of houses in the interval given

= "right answers" given

i.e. We give computer correct answers and it produces more

Example of regression: predicts continuous valued output (price)

E.g. Breast cancer - malignant or benign tumour

Example of classification: discrete valued output - malignant (1), or benign (0)

Can also be done with more features. Eventually want an infinite number of features

Within the category of supervised learning there are two different types: [5]

- Regression
 - Simple linear regression
 - Multiple linear regression
- Classification
 - Logistic regression

Unsupervised learning

Computer is not told which values are correct or incorrect, so it has to categorise the values itself

E.g. Cocktail party = more than one person talking, at different distances from some microphones. Program must separate the different voices

This also has different types: [5]

- Clustering
- Dimensionality reduction

Model & cost function [1]

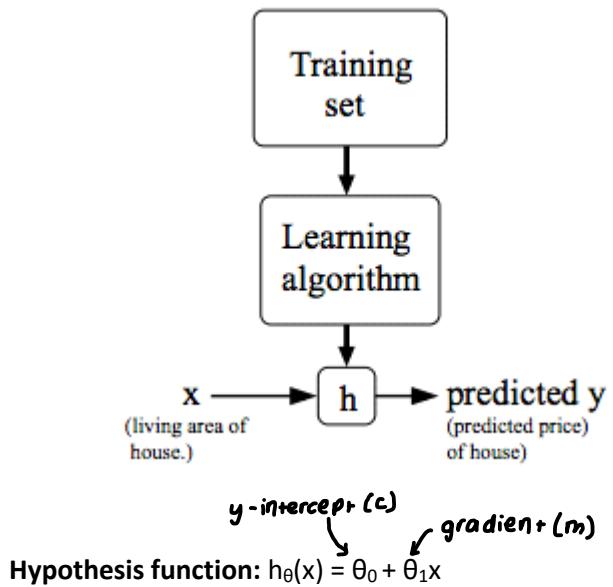
Friday, November 24, 2023 10:39 AM

Training set notation

m = no. of training examples

x = "input" variables / features

y = "output" variable / "target" variable



To calculate accuracy of hypothesis function, **cost function** is used

- Takes average difference of hypothesis results with inputs from x's and actual y output
- Calculated with this equation:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2$$

want to minimise *so that it isn't negative*
no. of values on graph

Objective is to minimise $J(\theta_1)$

What is the difference between cost function and loss function? [32]

Loss function: Used when we refer to the error for a single training example.

Cost function: Used to refer to an average of the loss functions over an entire training data.

Loss function [31]

Sunday, August 11, 2024 4:03 PM

A loss function (also known as an error function) is a fundamental element in machine learning used to quantify the difference between the predicted output of a model and the actual target values. It is used to measure the accuracy of the model's predictions. The model can then use the gradient of the loss function to adjust the model's parameters to reduce the loss and improve the model's performance.

Loss functions for regression

- **Mean Square Error (MSE)** calculates the average of the squared differences between predicted and actual values, heavily penalising larger errors. This makes MSE effective in scenarios where large errors must be avoided, such as in house price prediction.
- **Mean Absolute Error (MAE)** calculates the average absolute differences between predicted and actual values without squaring them, treating all errors equally. MAE is less sensitive to outliers compared to MSE, making it suitable for when the model should not be overly influenced by anomalies in the data. This means an example of its use could be in predicting delivery times for a service like UberEats, where rare delays due to traffic or weather should not disproportionately skew the model.
- **Huber Loss, or Smooth Mean Absolute Error**, is a combination of the two: it behaves like MSE for small errors and MAE for large errors. As a result, the sensitivity to outliers is reduced whilst still being accurate.

Loss functions for classification

- **Binary Cross-Entropy Loss** (also known as **Log Loss**) is often used for binary classification problems. It measures the difference between predicted probabilities and actual binary outcomes, penalising predictions that are far from the actual values. This loss function is important in tasks such as logistic regression and training neural networks for binary classification, where the model must output the probability of a sample belonging to a particular class.
- **Hinge Loss** is often used in classifiers which try to maximise the difference between data points and the decision boundary. Hinge Loss penalizes incorrect classifications and correct classifications that are too close to the decision boundary. This encourages the model to maximise the margin between classes, making it better at generalising and accurately classifying unseen data.

This table summarises the features of each loss function:

Loss function	Applicability to classification	Applicability to regression	Sensitivity to outliers
MSE	✓	✗	High
MAE	✓	✗	Low
Cross-entropy	✗	✓	Medium
Hinge Loss	✗	✓	Low
Huber Loss	✓	✗	Medium

Use

There are ways of implementing a loss function in Python, but it can also be done using libraries. Here is some example code using libraries to implement the Mean Square Error:

```
1 from sklearn.metrics import mean_squared_error
2
3 # actual values
4 y_true = [1, 2, 3, 4, 5]
5
6 # predicted values
7 y_pred = [1.1, 2.2, 2.9, 4.1, 4.9]
8
9 # Calculate MSE using scikit-learn
10 mse_value = mean_squared_error(y_true, y_pred)
11 print(mse_value)
12 # 0.016
```


Gradient descent [1] [37]

07 December 2023 21:04

Gradient descent = finds the values of a function's parameters that minimise the cost function

Gradient = change in weights with regard to change in error, or the slope of a function

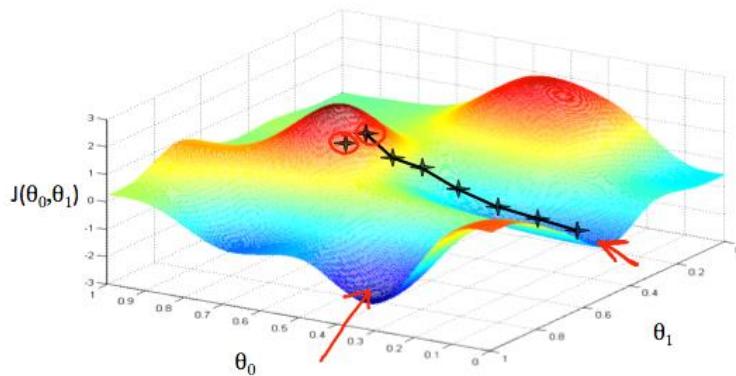
- Higher gradient = faster learning
- Slope of 0 = stops learning

How gradient descent works

We want to minimise the cost function $J(\theta_0, \theta_1)$ which has parameters θ_0 and θ_1 . At first, θ_0 and θ_1 are assigned some random numbers. Then, using the equation below, the cost function takes steps in the steepest downwards direction.

Visual representation

On 3D graph, at any point on curve, take one step in the direction that gets you downhill
Repeat until at minimum point or local minimum



Mathematical representation

Repeat until convergence:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

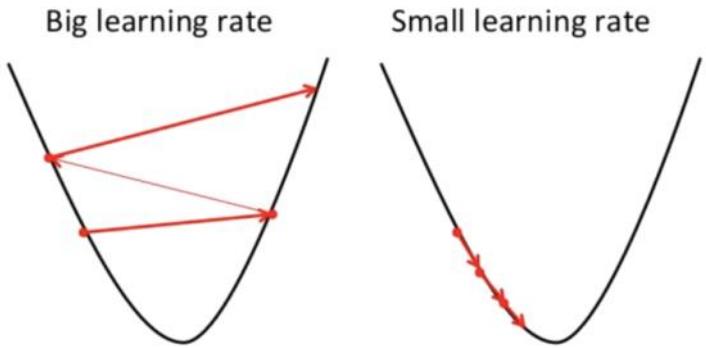
derivative term

- Assignment: e.g. $a := b$ means the value of a is overwritten and the value of a becomes the value of b
 - Different to $=$, which is truth assertion
- Learning rate = how big the step downhill is

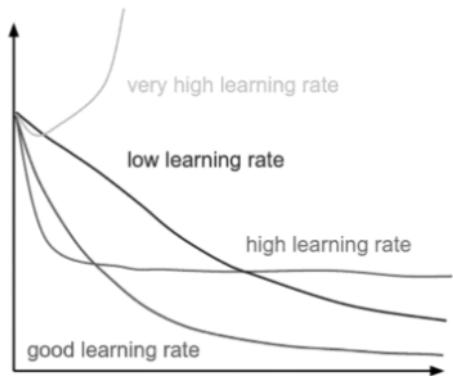
The derivative term can also be written as $\nabla f(a)$, which represents divergence. This means a derivative in the x, y, z direction, etc. Divergence has applications in fluid flow and Maxwell's equations so learning about it could be useful for when I do a physics degree. [38]

Learning rate

This means how big the steps are in the direction of the local minimum. The learning rate must be neither too high nor too low: too high could mean overstepping the minimum and bouncing between either side of it; too low would mean it would take a very long time to get to the minimum.



To find the right learning rate, and check the gradient descent algorithm works, you can plot a graph of cost function vs number of iterations.



How does gradient descent converge with a fixed step size α ?

- As minimum is approached, steps become smaller and smaller
- If gradient is steep, derivative term is high $\Rightarrow \alpha$ term is larger number
- As gradient decreases, derivative term decreases
 $\Rightarrow \alpha$ term decreases
 \Rightarrow gradient descent automatically takes smaller steps

Machine learning project steps [14]

14 February 2024 15:43

OVERVIEW

- Define problem
- Prepare data
- Evaluate algorithms
- Improve results
- Present results

1 Download, install and start Python SciPy

Some libraries must be installed. Libraries are collections of prewritten code that programmers use to optimise tasks, like the furniture and décor that goes inside a house. Most of the time, a library is a collection of objects and functions that can be used individually, and must be configured to work together. [13]

2 Load the data

Import all modules, functions and objects from the libraries

E.g.

```
1 # Load libraries
2 from pandas import read_csv
3 from pandas.plotting import scatter_matrix
4 from matplotlib import pyplot as plt
5 from sklearn.model_selection import train_test_split
6 from sklearn.model_selection import cross_val_score
7 from sklearn.model_selection import StratifiedKFold
8 from sklearn.metrics import classification_report
9 from sklearn.metrics import confusion_matrix
10 from sklearn.metrics import accuracy_score
11 from sklearn.linear_model import LogisticRegression
12 from sklearn.tree import DecisionTreeClassifier
13 from sklearn.neighbors import KNeighborsClassifier
14 from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
15 from sklearn.naive_bayes import GaussianNB
16 from sklearn.svm import SVC
17 ...
```

Load dataset

Name each column (line 4)

```
1 ...
2 # Load dataset
3 url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"
4 names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
5 dataset = read_csv(url, names=names)
```

3 Summarise data set

Find dimensions of data set

Preview using `print(dataset.head(20))` - first 20 lines

Statistical summary

Class distribution

4 Data visualisation

Plot graphs of each individual variable

- Histogram

Plot graphs of all pairs of attributes

5 Create models and evaluate accuracy

80% used to train

20% used to test accuracy

Linear regression using libraries [15]

Monday, March 04, 2024 7:43 PM

This comes from a step-by-step tutorial on how to create your own linear regression model. My input is shown in the grey boxes, and the output is shown outside and under the boxes. My comments, which describe what each line of code does, are shown in # blue

linearregressionwithlibraries

March 4, 2024

```
[37]: import numpy as np # numpy is a library that performs mathematical operations
import pandas as pd # pandas is used for data manipulation and analysis
import matplotlib.pyplot as plt # matplotlib (a component of numpy) is what plots the graphs
import statsmodels.api as sm # statsmodels explores data, estimates statistical models, and performs statistical tests

[38]: data = pd.read_csv('data.csv') # defining what the command "data" means - to read the csv file

[39]: data # we are shown what the raw data given to us is, which we will use to train and test the model

[39]:      SAT    GPA
0     1714  2.40
1     1664  2.52
2     1760  2.54
3     1685  2.74
4     1693  2.83
...
79    1936  3.71
80    1810  3.71
81    1987  3.73
82    1962  3.76
83    2050  3.81

[84 rows x 2 columns]

[40]: data.describe() # gives us descriptive statistics of the data

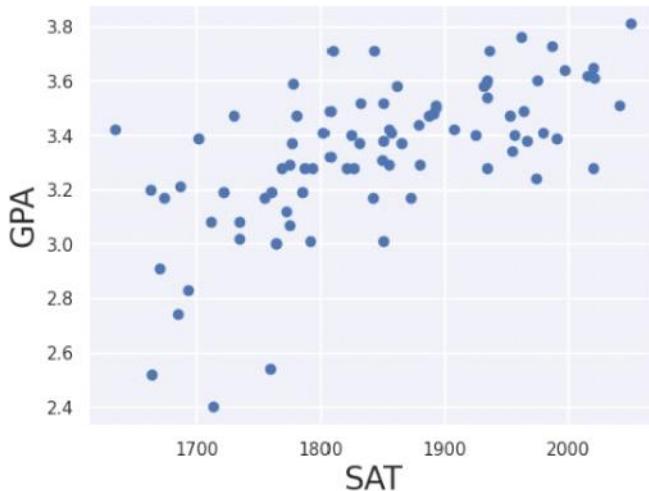
[40]:      SAT    GPA
count    84.000000  84.000000
mean    1845.273810  3.330238
std     104.530661  0.271617
min    1634.000000  2.400000
25%    1772.000000  3.190000
50%    1846.000000  3.380000
```

```
75%    1934.000000  3.502500
max     2050.000000  3.810000
```

```
[41]: y = data ['GPA'] # GPA is the dependent variable so is on the y-axis of the graph
x1 = data ['SAT'] # SAT is the independent variable so is on the x-axis
```

```
[42]: plt.scatter(x1,y) # plots a scatter plot
plt.xlabel('SAT', fontsize = 20) # the x-axis is labeled SAT
plt.ylabel('GPA', fontsize = 20) # the y-axis is labeled GPA
plt.show # shows the plot
```

```
[42]: <function matplotlib.pyplot.show(close=None, block=None)>
```



```
[43]: x = sm.add_constant(x1)
results = sm.OLS(y,x).fit()
results.summary()
```

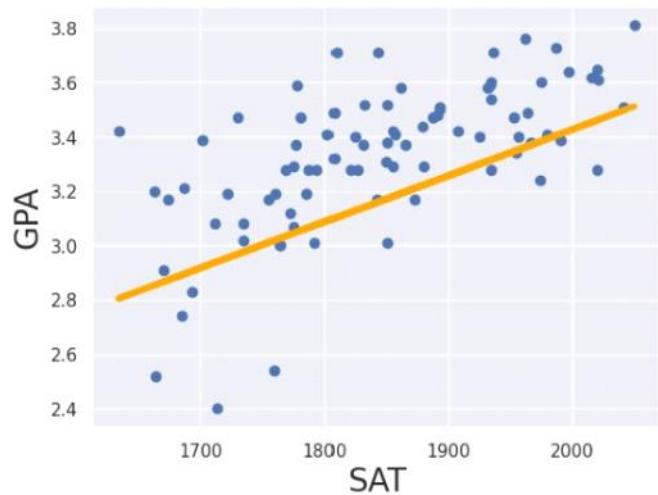
```
[43]:
```

Dep. Variable:	GPA	R-squared:	0.406			
Model:	OLS	Adj. R-squared:	0.399			
Method:	Least Squares	F-statistic:	56.05			
Date:	Sun, 03 Mar 2024	Prob (F-statistic):	7.20e-11			
Time:	11:34:45	Log-Likelihood:	12.672			
No. Observations:	84	AIC:	-21.34			
Df Residuals:	82	BIC:	-16.48			
Df Model:	1					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	0.2750	0.409	0.673	0.503	-0.538	1.088
SAT	0.0017	0.000	7.487	0.000	0.001	0.002
Omnibus:	12.839	Durbin-Watson:			0.950	
Prob(Omnibus):	0.002	Jarque-Bera (JB):			16.155	
Skew:	-0.722	Prob(JB):			0.000310	
Kurtosis:	4.590	Cond. No.			3.29e+04	

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 3.29e+04. This might indicate that there are strong multicollinearity or other numerical problems.

```
[45]: plt.scatter(x1,y) # plot a scatter plot
yhat = 0.0017*x1 + 0.0275
# define the regression equation, to plot it later. The 0.0017 and 0.0275 come from the "coef" column from the table above
fig = plt.plot(x1,yhat, lw=4, c='orange', label = 'regression line') # plot the regression line against the independent variable, SAT
plt.xlabel('SAT', fontsize = 20) # label the axes
plt.ylabel('GPA', fontsize = 20)
plt.show()
```



4

Linear regression [40]

Sunday, March 03, 2024 10:46 AM

A form of analysis in which the independent variable is used to predict the value of the dependent variable. There can be multiple independent variables. Linear regression estimates the coefficients of the linear equation, which forms a straight line or surface that minimises the difference between the predicted values of the independent variable and the real values.

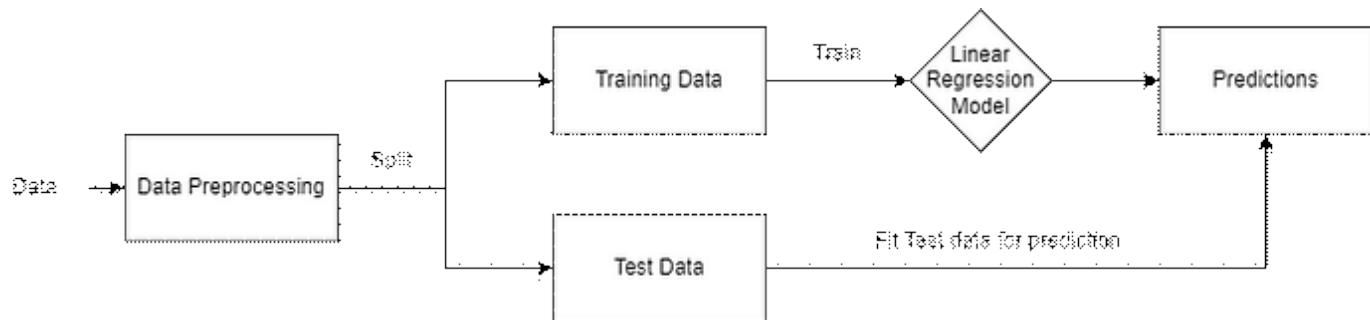
Some programs and environments make linear regression easier, such as:

- R linear regression.
- MATLAB linear regression.
- Sklearn linear regression.
- Linear regression Python.
- Excel linear regression.

In my linear regression project I have been using Sklearn.

Assumptions made:

- Variables are continuous
- There is a linear relationship between the variables - use a scatterplot to check this
- The observations must be independent
- No significant outliers in the data
- No homoscedasticity - variances along the line of best fit are similar throughout the line
- The errors follow a normal distribution



Simple linear regression from scratch [29]

Sunday, June 02, 2024 10:33 PM

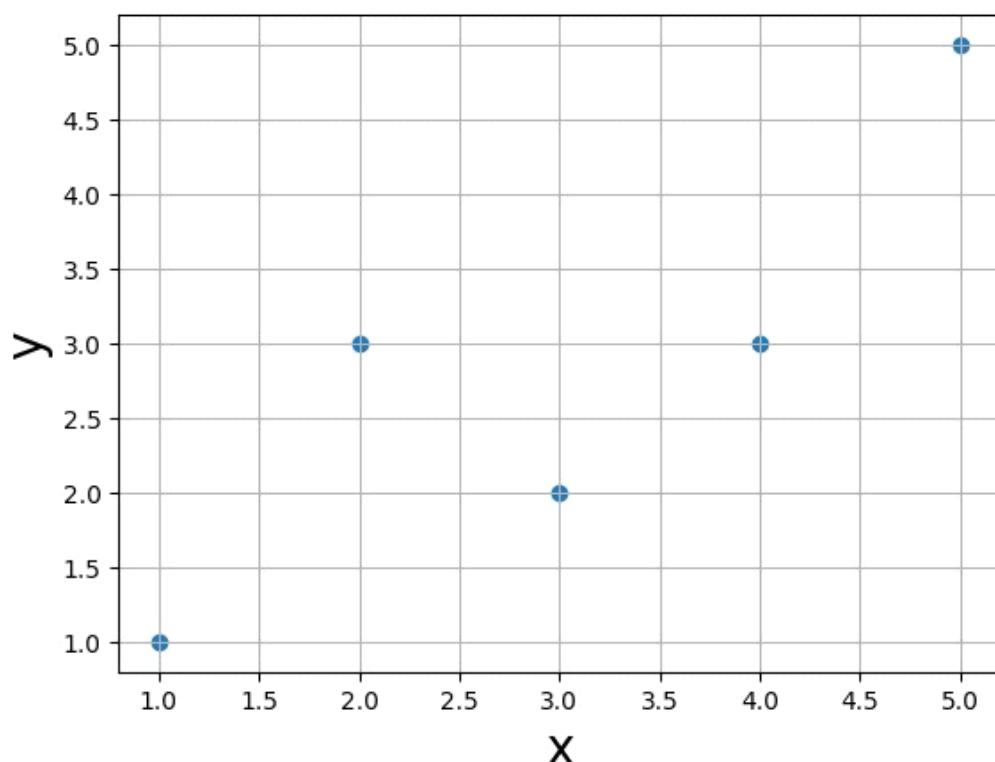
```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
```

```
In [2]: rawdata = {'x': [1, 2, 4, 3, 5], 'y': [1, 3, 3, 2, 5]}
df = pd.DataFrame(rawdata)
df
```

```
Out[2]:   x  y
0  1  1
1  2  3
2  4  3
3  3  2
4  5  5
```

```
In [3]: df.to_csv('rawdata.csv', index=False)
```

```
In [4]: data = pd.read_csv('rawdata.csv')
y = data ['y']
x1 = data ['x']
plt.scatter(x1,y)
plt.xlabel('x', fontsize = 20)
plt.ylabel('y', fontsize = 20)
plt.grid()
plt.show()
```



Simple linear regression equation

$$y = B_0 + B_1 \times x$$

y is the output variable we want to predict

x is the input variable we know

B₀ (intercept) is the bias

B₁ is the slope

To estimate B₁:

$$B_1 = \frac{\sum (x_i - \text{mean}(x)) \times (y_i - \text{mean}(y))}{\sum (x_i - \text{mean}(x))^2}$$

where x_i and y_i means using all different values of x and y where i means the i'th value of x and y

To estimate B₀:

$$B_0 = \text{mean}(y) - B_1 \times \text{mean}(x)$$

Estimating B₁

[23] - [43]

$$\text{mean} = \frac{1}{n} \times \sum x$$

This can be calculated using the mean command using pandas

```
In [5]: mean_x = df['x'].mean() # calculating mean(x)
print(mean_x)
```

3.0

```
In [6]: def xminusmeanx(row):
    return row['x'] - mean_x
# calculates x - mean(x) for every value of x

df['x-mean_x'] = df.apply(xminusmeanx, axis=1)
# adds a column (as seen by axis=1) to df
# with the values calculated by the function above
print(df)
```

	x	y	x-mean_x
0	1	1	-2.0
1	2	3	-1.0
2	4	3	1.0
3	3	2	0.0
4	5	5	2.0

```
In [7]: mean_y = df['y'].mean()
def yminusmeany(row):
    return row['y'] - mean_y

df['y-mean_y'] = df.apply(yminusmeany, axis=1)
print(df)
```

	x	y	x-mean_x	y-mean_y
0	1	1	-2.0	-1.8
1	2	3	-1.0	0.2
2	4	3	1.0	0.2
3	3	2	0.0	-0.8
4	5	5	2.0	2.2

```
In [8]: def multiply(row):
    return row['x-mean_x']*row['y-mean_y']

df['multiply'] = df.apply(multiply, axis=1)
print(df)

      x  y  x-mean_x  y-mean_y  multiply
0   1  1      -2.0     -1.8      3.6
1   2  3      -1.0      0.2     -0.2
2   4  3       1.0      0.2      0.2
3   3  2       0.0     -0.8     -0.0
4   5  5       2.0      2.2      4.4
```

```
In [9]: numerator = df['multiply'].sum()
print(f"numerator = {numerator}")

numerator = 8.0
```

```
In [10]: def squared(row):
    return (row['x-mean_x'])**2

df['(x - mean_x)^2'] = df.apply(squared, axis=1)
print(df)

      x  y  x-mean_x  y-mean_y  multiply  (x - mean_x)^2
0   1  1      -2.0     -1.8      3.6      4.0
1   2  3      -1.0      0.2     -0.2      1.0
2   4  3       1.0      0.2      0.2      1.0
3   3  2       0.0     -0.8     -0.0      0.0
4   5  5       2.0      2.2      4.4      4.0
```

```
In [11]: denominator = df['(x - mean_x)^2'].sum()
print(f"denominator = {denominator}")

denominator = 10.0

Thus  $B_1 = \frac{8}{10}$ 
```

```
In [12]: B1 = numerator/denominator
print(f"B1 = {B1}")

B1 = 0.8
```

Calculating B0

Reminder: $B_0 = \text{mean}(y) - B_1 \times \text{mean}(x)$

```
In [13]: B0 = mean_y-B1*mean_x
print(f"B0 = {B0}")

B0 = 0.3999999999999947
```

Floating point error! So must round B0 to 2SF

```
In [14]: B0 = round(B0, 2) # 2 is the number of sig fig
print(f"B0 = {B0}")
B0 = 0.4
```

Forming the equation

We now know that

$$y = B0 + B1 \times x$$

is equal to

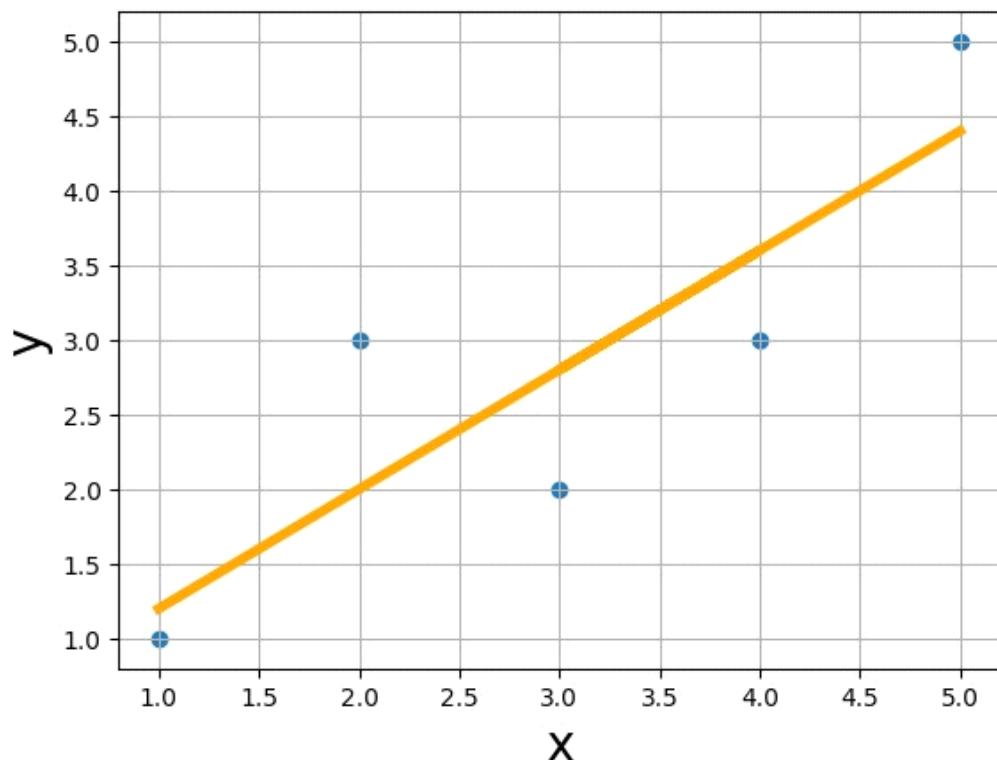
$$y = 0.4 + 0.8 \times x$$

```
In [15]: def predy(row):
    return 0.4+0.8*row['x']

df['predicted y'] = df.apply(predy, axis=1)
print(df)
```

x	y	x-mean_x	y-mean_y	multiply	(x - mean_x)^2	predicted y
0	1	1	-2.0	-1.8	3.6	4.0
1	2	3	-1.0	0.2	-0.2	1.0
2	4	3	1.0	0.2	0.2	1.0
3	3	2	0.0	-0.8	-0.0	0.0
4	5	5	2.0	2.2	4.4	4.0

```
In [16]: plt.scatter(x1, y)
yhat = B0 + B1*x1
fig = plt.plot(
    x1,
    yhat,
    lw=4,
    c='orange',
    label='regression line')
plt.xlabel('x', fontsize=20)
plt.ylabel('y', fontsize=20)
plt.grid()
plt.show()
```



Estimating error

Calculated using Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{\sum(p_i - y_i)^2}{n}}$$

where p is the predicted value and n is the number of predictions

```
In [17]: def p_minus_y_squared(row):
    return (
        row['predicted y']-row['y'])**2

df['(p - y)^2'] = df.apply(
    p_minus_y_squared,
    axis=1)
print(df)

      x  y  x-mean_x  y-mean_y  multiply  (x - mean_x)^2  predicted y  (p - y)^2
0   1  1       -2.0     -1.8      3.6          4.0         1.2      0.04
1   2  3       -1.0      0.2     -0.2          1.0         2.0      1.00
2   4  3        1.0      0.2      0.2          1.0         3.6      0.36
3   3  2        0.0     -0.8     -0.0          0.0         2.8      0.64
4   5  5        2.0      2.2      4.4          4.0         4.4      0.36

In [18]: rmse_numerator = df['(p - y)^2'].sum()
n = len(df)
rmse = math.sqrt(rmse_numerator/n)
```

```
rmse = round(rmse, 3)
print(rmse)
```

0.693

Thus each prediction is wrong by about 0.693 units

Multiple linear regression [41]

10 June 2024 14:26

Multiple linear regression

Linear regression with multiple independent variables

```
In [1]: import numpy as np  
import pandas as pd
```

```
In [2]: dataset = pd.read_csv('./Tutorials/50_Startups.csv')
```

```
In [3]: dataset.head()
```

```
Out[3]:   R&D Spend Administration Marketing Spend      State    Profit  
0       165349.20        136897.80        471784.10  New York  192261.83  
1       162597.70        151377.59        443898.53  California 191792.06  
2       153441.51        101145.55        407934.54  Florida   191050.39  
3       144372.41        118671.85        383199.62  New York  182901.99  
4       142107.34        91391.77        366168.42  Florida   166187.94
```

```
In [4]: # convert to numpy  
  
x = dataset.iloc[:, 0:4].to_numpy()  
y = dataset.iloc[:, -1].to_numpy()
```

```
In [5]: from sklearn.preprocessing import OneHotEncoder  
from sklearn.compose import ColumnTransformer  
  
encoder = OneHotEncoder(  
    drop='first',  
    dtype=int) # making into 1s and 0s  
ct = ColumnTransformer(  
    [('categorical_encoding',  
        encoder,  
        [3])],  
    remainder='passthrough')
```

```
In [6]: x = ct.fit_transform(x)
```

```
In [7]: from sklearn.model_selection import train_test_split  
  
x_train, x_test, y_train, y_test = train_test_split(  
    x,  
    y,  
    test_size=.25,  
    random_state=0)
```

```
In [8]: from sklearn.linear_model import LinearRegression  
  
regressor = LinearRegression()
```

```
In [9]: regressor.fit(x_train, y_train)
```

```
Out[9]: ▾ LinearRegression ⓘ ?  
LinearRegression()  
  
In [10]: predictions = regressor.predict(x_test)  
  
In [11]: from sklearn.metrics import r2_score  
  
r_squared = r2_score(  
    y_test,  
    predictions)  
  
In [12]: print(r_squared)  
0.931581519915206
```

$$R^2$$

A measure that tells us how well the regression line fits the actual data.

$R^2 = 1$ would mean all the variation in the y values is accounted for by the x values

$R^2 = 0$ would mean none of the variation in the y values is accounted for by the x values [39]

```
In [13]: N = len(x_test)  
print(f"N = {N}")  
  
N = 13
```

Now we need to find the number of predictors in x_test. Originally, there were 4 predictors in our dataset, but after categorically encoding we had 6 (the state column became three separate columns of 0's and 1's). However, we must remember that we dropped the first of these columns to avoid the dummy variable trap. So, there are 5 predictors in our dataset.

```
In [14]: k = 5  
  
In [15]: adj_r_squared = 1 - (((1 - (r_squared ** 2)) * (N - 1)) / (N - k - 1))  
# see equation for adjusted R squared below  
print(f'The adjusted R score of our model is {adj_r_squared}')
```

The adjusted R score of our model is 0.7734470769957578

Why is an adjusted R^2 necessary?

Adding more variables/predictors will increase the value of R^2 , which seems like it should be a good thing. However, it is actually because the model is overfitting. The adjusted R^2 will take the predictors into account and shows a more reliable correlation. [39]

Simple logistical regression algorithm

Wednesday, June 05, 2024 9:41 PM

Intro to Logistic Regression

A Logistic Regression algorithm works by implementing a linear equation with independent or explanatory variables to predict a response value. E.g. with just one explanatory variable (x_1)

$$z = \beta_0 + \beta_1 x_1$$

where z is the target variable / response variable, and β_1 & β_2 are the parameters of the model

If there are multiple explanatory variables, this is extended to

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

Sigmoid function

z is converted to a probability value between 0 and 1. The sigmoid function maps predicted values to probability values. It is given by the following formula:

$$\phi = \frac{1}{1+e^{-z}}$$

Decision boundary

The probability value is mapped to either 0 or 1 based on whether it is above or below a threshold value, or the decision boundary.

$$p \geq 0.5 \therefore \text{class} = 1$$

$$p < 0.5 \therefore \text{class} = 0$$

Assumptions

- Dependent variable is binary, multinomial or ordinal
- Observations are independent
- Little to no multicollinearity between independent variables
- Linearity of independent variables
- Success of model depends on sample size

Types

1. Binary logistical regression - target variable has two possible categories (e.g. yes or no, true or false, pass or fail)
2. Multinomial logistical regression - 3+ categories in no particular order
3. Ordinal logistical regression - 3+ categories with intrinsic order (e.g. poor, average, good and excellent)

Importing libraries and data

```
In [255...]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import os
```

```
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

In [256]: import warnings
         warnings.filterwarnings('ignore')

In [257]: data = '/home/uxue/projects/EPQ/Tutorials/weatherAUS.csv'
         df = pd.read_csv(data)
```

Preparing data

```
In [258]: df.shape # "shape" tells us no. of columns by no. of rows
Out[258]: (145460, 23)
```

```
In [259]: df.head() # "head" gives a preview of the dataframe
```

```
Out[259]:
   Date  Location  MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine  WindGustDir
0  2008-12-01    Albury     13.4      22.9       0.6        NaN        NaN
1  2008-12-02    Albury      7.4      25.1       0.0        NaN        NaN
2  2008-12-03    Albury     12.9      25.7       0.0        NaN        NaN
3  2008-12-04    Albury      9.2      28.0       0.0        NaN        NaN
4  2008-12-05    Albury     17.5      32.3       1.0        NaN        NaN
```

5 rows × 23 columns

```
In [260]: col_names = df.columns
          col_names
```

```
Out[260]: Index(['Date', 'Location', 'MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation',
                 'Sunshine', 'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm',
                 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
                 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am',
                 'Temp3pm', 'RainToday', 'RainTomorrow'],
                dtype='object')
```

```
In [261]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Date             145460 non-null   object  
 1   Location         145460 non-null   object  
 2   MinTemp          143975 non-null   float64 
 3   MaxTemp          144199 non-null   float64 
 4   Rainfall          142199 non-null   float64 
 5   Evaporation      82670 non-null   float64 
 6   Sunshine          75625 non-null   float64 
 7   WindGustDir       135134 non-null   object  
 8   WindGustSpeed     135197 non-null   float64 
 9   WindDir9am        134894 non-null   object  
 10  WindDir3pm        141232 non-null   object  
 11  WindSpeed9am     143693 non-null   float64 
 12  WindSpeed3pm     142398 non-null   float64 
 13  Humidity9am       142806 non-null   float64 
 14  Humidity3pm       140953 non-null   float64 
 15  Pressure9am       130395 non-null   float64 
 16  Pressure3pm       130432 non-null   float64 
 17  Cloud9am          89572 non-null   float64 
 18  Cloud3pm          86102 non-null   float64 
 19  Temp9am           143693 non-null   float64 
 20  Temp3pm           141851 non-null   float64 
 21  RainToday          142199 non-null   object  
 22  RainTomorrow       142193 non-null   object  
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

If Dtype is *object*, the variable is categorical.

If Dtype is *float64*, the variable is numerical.

```
In [262]: # find categorical variables

categorical = [var for var in df.columns if df[var].dtype=='O']

print('There are {} categorical variables\n'.format(
    len(categorical)))

print('The categorical variables are:', categorical)
```

There are 7 categorical variables

The categorical variables are: ['Date', 'Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday', 'RainTomorrow']

```
In [263]: # view categorical variables

df[categorical].head()
```

	Date	Location	WindGustDir	WindDir9am	WindDir3pm	RainToday	RainTomorrow
0	2008-12-01	Albury	W	W	WNW	No	
1	2008-12-02	Albury	WNW	NNW	WSW	No	
2	2008-12-03	Albury	WSW	W	WSW	No	
3	2008-12-04	Albury	NE	SE	E	No	
4	2008-12-05	Albury	W	ENE	NW	No	

- There are six categorical variables: location, windgustdir, winddir9am, winddir3pm, raintoday, raintomorrow
- There are two binary categorical variables: raintoday, raintomorrow
- Raintomorrow is the target variable

Problems within categorical variables

Missing values within categorical variables

```
In [264]: df[categorical].isnull().sum()
```

```
Out[264]: Date          0
Location        0
WindGustDir    10326
WindDir9am     10566
WindDir3pm      4228
RainToday       3261
RainTomorrow    3267
dtype: int64
```

```
In [265]: # print categorical variables containing missing values
cat1 = [var for var in categorical if df[var].isnull().sum() != 0]
print(df[cat1].isnull().sum())
WindGustDir    10326
WindDir9am     10566
WindDir3pm      4228
RainToday       3261
RainTomorrow    3267
dtype: int64
```

Frequency counts of categorical variables

```
In [266]: for var in categorical:
    print(df[var].value_counts())
```

```
Date
2013-11-12    49
2014-09-01    49
2014-08-23    49
2014-08-24    49
2014-08-25    49
..
2007-11-29    1
2007-11-28    1
2007-11-27    1
2007-11-26    1
2008-01-31    1
Name: count, Length: 3436, dtype: int64
Location
Canberra        3436
Sydney          3344
Darwin          3193
Melbourne       3193
Brisbane        3193
Adelaide        3193
Perth           3193
Hobart          3193
Albany          3040
MountGambier     3040
Ballarat         3040
Townsville       3040
GoldCoast        3040
Cairns          3040
Launceston      3040
AliceSprings     3040
Bendigo          3040
Albury           3040
MountGinini      3040
Wollongong       3040
Newcastle         3039
Tuggeranong      3039
Penrith          3039
Woomera          3009
Nuriootpa        3009
Cobar            3009
CoffsHarbour     3009
Moree            3009
Sale              3009
PerthAirport      3009
PearceRAAF        3009
Witchcliffe      3009
BadgerysCreek     3009
Mildura          3009
NorfolkIsland     3009
MelbourneAirport   3009
Richmond          3009
SydneyAirport      3009
WaggaWagga        3009
Williamtown       3009
Dartmoor          3009
Watsonia          3009
Portland          3009
Walpole           3006
NorahHead          3004
SalmonGums         3001
```

```
Katherine      1578
Nhil          1578
Uluru         1578
Name: count, dtype: int64
WindGustDir
W            9915
SE           9418
N             9313
SSE          9216
E             9181
S             9168
WSW          9069
SW            8967
SSW          8736
WNW          8252
NW            8122
ENE          8104
ESE          7372
NE            7133
NNW          6620
NNE          6548
Name: count, dtype: int64
WindDir9am
N            11758
SE           9287
E             9176
SSE          9112
NW            8749
S             8659
W             8459
SW            8423
NNE          8129
NNW          7980
ENE          7836
NE            7671
ESE          7630
SSW          7587
WNW          7414
WSW          7024
Name: count, dtype: int64
WindDir3pm
SE           10838
W            10110
S             9926
WSW          9518
SSE          9399
SW            9354
N             8890
WNW          8874
NW            8610
ESE          8505
E             8472
NE            8263
SSW          8156
NNW          7870
ENE          7857
NNE          6590
Name: count, dtype: int64
RainToday
No            110319
```

```
Yes      31880
Name: count, dtype: int64
RainTomorrow
No       110316
Yes      31877
Name: count, dtype: int64
```

```
In [267]: # view frequency distribution of categorical variables

for var in categorical:
    print(df[var].value_counts()/float(len(df)))
```

```
Date
2013-11-12    0.000337
2014-09-01    0.000337
2014-08-23    0.000337
2014-08-24    0.000337
2014-08-25    0.000337
...
2007-11-29    0.000007
2007-11-28    0.000007
2007-11-27    0.000007
2007-11-26    0.000007
2008-01-31    0.000007
Name: count, Length: 3436, dtype: float64
Location
Canberra      0.023622
Sydney        0.022989
Darwin         0.021951
Melbourne     0.021951
Brisbane       0.021951
Adelaide       0.021951
Perth          0.021951
Hobart         0.021951
Albany          0.020899
MountGambier   0.020899
Ballarat        0.020899
Townsville      0.020899
GoldCoast       0.020899
Cairns          0.020899
Launceston     0.020899
AliceSprings   0.020899
Bendigo         0.020899
Albury          0.020899
MountGinini     0.020899
Wollongong      0.020899
Newcastle        0.020892
Tuggeranong    0.020892
Penrith         0.020892
Woomera         0.020686
Nuriootpa       0.020686
Cobar           0.020686
CoffsHarbour   0.020686
Moree           0.020686
Sale             0.020686
PerthAirport    0.020686
PearceRAAF      0.020686
Witchcliffe    0.020686
BadgerysCreek   0.020686
Mildura          0.020686
NorfolkIsland   0.020686
MelbourneAirport 0.020686
Richmond         0.020686
SydneyAirport   0.020686
WaggaWagga      0.020686
Williamtown     0.020686
Dartmoor         0.020686
Watsonia         0.020686
Portland         0.020686
Walpole          0.020665
NorahHead        0.020652
SalmonGums       0.020631
```

```
Katherine      0.010848
Nhil          0.010848
Uluru         0.010848
Name: count, dtype: float64
WindGustDir
W            0.068163
SE           0.064746
N             0.064024
SSE          0.063358
E             0.063117
S              0.063028
WSW          0.062347
SW            0.061646
SSW          0.060058
WNW          0.056730
NW            0.055837
ENE          0.055713
ESE          0.050681
NE             0.049038
NNW          0.045511
NNE          0.045016
Name: count, dtype: float64
WindDir9am
N            0.080833
SE           0.063846
E             0.063083
SSE          0.062643
NW            0.060147
S              0.059528
W             0.058153
SW            0.057906
NNE          0.055885
NNW          0.054860
ENE          0.053870
NE             0.052736
ESE          0.052454
SSW          0.052159
WNW          0.050969
WSW          0.048288
Name: count, dtype: float64
WindDir3pm
SE           0.074508
W             0.069504
S              0.068239
WSW          0.065434
SSE          0.064616
SW            0.064306
N             0.061116
WNW          0.061006
NW            0.059192
ESE          0.058470
E             0.058243
NE             0.056806
SSW          0.056070
NNW          0.054104
ENE          0.054015
NNE          0.045305
Name: count, dtype: float64
RainToday
No            0.758415
```

```
Yes      0.219167
Name: count, dtype: float64
RainTomorrow
No       0.758394
Yes      0.219146
Name: count, dtype: float64

In [268]: # check for cardinality (no. of labels in a variable)
# we don't want this to be high

for var in categorical:
    print(var, 'contains', len(df[var].unique()), 'labels')

Date contains 3436 labels
Location contains 49 labels
WindGustDir contains 17 labels
WindDir9am contains 17 labels
WindDir3pm contains 17 labels
RainToday contains 3 labels
RainTomorrow contains 3 labels

In [269]: # preprocess Date as it has high cardinality

df['Date'].dtypes

Out[269]: dtype('O')

In [270]: # O means that it is an object

# make the dates (currently strings) into datetime format

df['Date'] = pd.to_datetime(df['Date'])

In [271]: # extract year from date

df['Year'] = df['Date'].dt.year

df['Year'].head()

Out[271]: 0    2008
1    2008
2    2008
3    2008
4    2008
Name: Year, dtype: int32

In [272]: # extract month from date

df['Month'] = df['Date'].dt.month

df['Month'].head()

Out[272]: 0    12
1    12
2    12
3    12
4    12
Name: Month, dtype: int32
```

```
In [273... # extract day from date
df['Day'] = df['Date'].dt.day
df['Day'].head()

Out[273... 0    1
1    2
2    3
3    4
4    5
Name: Day, dtype: int32

In [274... # summary of dataset again
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 26 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   Date              145460 non-null   datetime64[ns]
 1   Location          145460 non-null   object 
 2   MinTemp           143975 non-null   float64
 3   MaxTemp           144199 non-null   float64
 4   Rainfall          142199 non-null   float64
 5   Evaporation       82670 non-null    float64
 6   Sunshine          75625 non-null    float64
 7   WindGustDir       135134 non-null   object 
 8   WindGustSpeed     135197 non-null   float64
 9   WindDir9am        134894 non-null   object 
 10  WindDir3pm        141232 non-null   object 
 11  WindSpeed9am      143693 non-null   float64
 12  WindSpeed3pm      142398 non-null   float64
 13  Humidity9am       142806 non-null   float64
 14  Humidity3pm       140953 non-null   float64
 15  Pressure9am       130395 non-null   float64
 16  Pressure3pm       130432 non-null   float64
 17  Cloud9am          89572 non-null    float64
 18  Cloud3pm          86102 non-null    float64
 19  Temp9am           143693 non-null   float64
 20  Temp3pm           141851 non-null   float64
 21  RainToday          142199 non-null   object 
 22  RainTomorrow       142193 non-null   object 
 23  Year               145460 non-null   int32  
 24  Month              145460 non-null   int32  
 25  Day                145460 non-null   int32  
dtypes: datetime64[ns](1), float64(16), int32(3), object(6)
memory usage: 27.2+ MB
```

Three additional columns now - Year, Month, Day

```
In [275... # drop (ie delete) initial Date variable
df.drop('Date', axis=1, inplace = True)
```

```
In [276... # preview dataset again
      df.head()
```

```
Out[276... Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir Wind
      0 Albury    13.4     22.9      0.6        NaN       NaN       NaN        W
      1 Albury     7.4     25.1      0.0        NaN       NaN       NaN      WNW
      2 Albury    12.9     25.7      0.0        NaN       NaN       NaN      WSW
      3 Albury     9.2     28.0      0.0        NaN       NaN       NaN      NE
      4 Albury    17.5     32.3      1.0        NaN       NaN       NaN        W
```

5 rows × 25 columns

Explore categorical variables

```
In [277... # find categorical variables
      categorical = [var for var in df.columns if df[var].dtype=='O']
      print('There are {} categorical variables\n'.format(len(categorical)))
      print('The categorical variables are :', categorical)
      There are 6 categorical variables
      The categorical variables are : ['Location', 'WindGustDir', 'WindDir9am',
      'WindDir3pm', 'RainToday', 'RainTomorrow']
      Date has been removed so the number of categorical variables has decreased by 1
```

```
In [278... # check for missing values
      df[categorical].isnull().sum()
```

```
Out[278... Location          0
      WindGustDir     10326
      WindDir9am     10566
      WindDir3pm      4228
      RainToday       3261
      RainTomorrow     3267
      dtype: int64
```

Explore Location variable

```
In [279... # no. of labels in Location variable
      print('Location contains', len(df['Location'].unique()), 'labels')
      Location contains 49 labels
```

```
In [280... # check labels
      df.Location.unique()
```

```
Out[280]: array(['Albury', 'BadgerysCreek', 'Cobar', 'CoffsHarbour', 'Moree',
   'Newcastle', 'NorahHead', 'NorfolkIsland', 'Penrith', 'Richmond',
   'Sydney', 'SydneyAirport', 'WaggaWagga', 'Williamtown',
   'Wollongong', 'Canberra', 'Tuggeranong', 'MountGinini', 'Ballara
t',
   'Bendigo', 'Sale', 'MelbourneAirport', 'Melbourne', 'Mildura',
   'Nhil', 'Portland', 'Watsonia', 'Dartmoor', 'Brisbane', 'Cairns',
   'GoldCoast', 'Townsville', 'Adelaide', 'MountGambier', 'Nuriootp
a',
   'Woomera', 'Albany', 'Witchcliffe', 'PearceRAAF', 'PerthAirport',
   'Perth', 'SalmonGums', 'Walpole', 'Hobart', 'Launceston',
   'AliceSprings', 'Darwin', 'Katherine', 'Uluru'], dtype=object)
```

```
In [281]: # check frequency distribution of values
```

```
df.Location.value_counts()
```

```
Out[281]: Location
Canberra           3436
Sydney             3344
Darwin              3193
Melbourne           3193
Brisbane             3193
Adelaide             3193
Perth                3193
Hobart                3193
Albany                3040
MountGambier          3040
Ballarat              3040
Townsville             3040
GoldCoast              3040
Cairns                3040
Launceston             3040
AliceSprings          3040
Bendigo                3040
Albury                  3040
MountGinini             3040
Wollongong              3040
Newcastle               3039
Tuggeranong              3039
Penrith                 3039
Woomera                  3009
Nuriootpa                3009
Cobar                     3009
CoffsHarbour            3009
Moree                     3009
Sale                      3009
PerthAirport             3009
PearceRAAF                3009
Witchcliffe              3009
BadgerysCreek            3009
Mildura                  3009
NorfolkIsland            3009
MelbourneAirport          3009
Richmond                  3009
SydneyAirport             3009
WaggaWagga                3009
Williamtown              3009
Dartmoor                  3009
Watsonia                  3009
Portland                  3009
Walpole                     3006
NorahHead                  3004
SalmonGums                 3001
Katherine                  1578
Nhil                      1578
Uluru                      1578
Name: count, dtype: int64
```

```
In [282]: # make Location variable into 1s and 0s,
# by making all possible locations into columns
# and displaying '1' in that column if it is in that location

pd.get_dummies(df.Location, drop_first=True, dtype=int).head()
# default dtype is bool (T or F) so int makes it into 1 or 0
```

```
Out[282... Albany Albury AliceSprings BadgerysCreek Ballarat Bendigo Brisbane Cairns
      0     0     1       0       0     0     0     0     0     0
      1     0     1       0       0     0     0     0     0     0
      2     0     1       0       0     0     0     0     0     0
      3     0     1       0       0     0     0     0     0     0
      4     0     1       0       0     0     0     0     0     0
      5 rows × 48 columns
```

Explore WindGustDir variable

```
In [283... # no. of labels
print('WindGustDir contains', len(df['WindGustDir'].unique()), 'labels')
WindGustDir contains 17 labels

In [284... # check labels
df['WindGustDir'].unique()

Out[284... array(['W', 'WNW', 'WSW', 'NE', 'NNW', 'N', 'NNE', 'SW', nan, 'ENE',
       'SSE', 'S', 'NW', 'SE', 'ESE', 'E', 'SSW'], dtype=object)

In [285... # check frequency distribution
df.WindGustDir.value_counts()

Out[285... WindGustDir
      W    9915
      SE   9418
      N    9313
      SSE  9216
      E    9181
      S    9168
      WSW  9069
      SW   8967
      SSW  8736
      WNW  8252
      NW   8122
      ENE  8104
      ESE  7372
      NE   7133
      NNW  6620
      NNE  6548
      Name: count, dtype: int64

In [286... # encode this into dummy variables too (1s and 0s), inc extra dummy varia
pd.get_dummies(df.WindGustDir, drop_first=True, dummy_na=True, dtype=int)
```

```
Out[286... ENE ESE N NE NNE NNW NW S SE SSE SSW SW W WNW WSW NaN
      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0
      1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
      2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
      3   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
      4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0
```

```
In [287... # sum no. of 1s per variable over the rows, which tells us no. of observations
observations = pd.get_dummies(
    df.WindGustDir,
    drop_first=True,
    dummy_na=True).sum(axis=0)
observations
```

```
Out[287... ENE      8104
ESE      7372
N       9313
NE      7133
NNE     6548
NNW     6620
NW      8122
S       9168
SE      9418
SSE     9216
SSW     8736
SW      8967
W       9915
WNW     8252
WSW     9069
NaN     10326
dtype: int64
```

Thus there are 10326 missing values in WindGustDir variable

Explore WindDir9am variable

```
In [288... # print no. of labels
print('WindDir9am contains', len(
    df.WindDir9am.unique()), 'labels')
WindDir9am contains 17 labels
```

```
In [289... # check labels
df.WindDir9am.unique()
```

```
Out[289... array(['W', 'NNW', 'SE', 'ENE', 'SW', 'SSE', 'S', 'NE', nan, 'SSW', 'N',
    'WSW', 'ESE', 'E', 'NW', 'WNW', 'NNE'], dtype=object)
```

```
In [290... # frequency distribution
df.WindDir9am.value_counts()
```

```
Out[290... WindDir9am  
N      11758  
SE     9287  
E      9176  
SSE    9112  
NW     8749  
S      8659  
W      8459  
SW     8423  
NNE    8129  
NNW    7980  
ENE    7836  
NE     7671  
ESE    7630  
SSW    7587  
WNW    7414  
WSW    7024  
Name: count, dtype: int64
```

```
In [291... WindDir9am_dummies = pd.get_dummies(  
        df.WindDir9am,  
        drop_first=True,  
        dummy_na=True,  
        dtype=int)  
WindDir9am_dummies.head()
```

```
Out[291...   ENE  ESE   N   NE  NNE  NNW   NW   S   SE  SSE  SSW   SW   W   WNW   WSW   NaN  
0       0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0  
1       0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0  
2       0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0  
3       0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0  
4       1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
In [292... WindDir9am_dummies.sum(axis=0)
```

```
Out[292...   ENE    7836  
   ESE    7630  
      N    11758  
      NE    7671  
     NNE    8129  
     NNW    7980  
      NW    8749  
      S    8659  
      SE    9287  
     SSE    9112  
     SSW    7587  
      SW    8423  
      W    8459  
     WNW    7414  
     WSW    7024  
     NaN   10566  
dtype: int64
```

There are 10566 missing values from WindDir9am

Exploring WindDir3pm variable

```
In [293... # no. of labels

def len_labels(field):

    var = df[field]
    print(field,
          'contains',
          len(var.unique()),
          'labels')

len_labels('WindDir3pm') # <?>
```

WindDir3pm contains 17 labels

```
In [294... def labels(field):

    var = df[field]
    print(var.unique())

labels('WindDir3pm')
```

```
['WNW' 'WSW' 'E' 'NW' 'W' 'SSE' 'ESE' 'ENE' 'NNW' 'SSW' 'SW' 'SE' 'N' 'S'
 'NNE' 'nan' 'NE']
```

```
In [295... df.WindDir3pm.value_counts()

def freq_dist(field): # frequency distribution

    var = df[field]
    print(var.value_counts())

freq_dist('WindDir3pm')
```

```
WindDir3pm
SE      10838
W       10110
S       9926
WSW     9518
SSE     9399
SW      9354
N       8890
WNW     8874
NW      8610
ESE     8505
E       8472
NE      8263
SSW     8156
NNW     7870
ENE     7857
NNE     6590
Name: count, dtype: int64
```

```
In [296... def dummies(field):

    var = df[field]
    return pd.get_dummies(
        var,
        drop_first=True,
        dummy_na=True,
```

```
        dtype=int)

dummies('WindDir3pm').head()

Out[296...  ENE  ESE  N  NE  NNE  NNW  NW  S  SE  SSE  SSW  SW  W  WNW  WSW  NaN
0         0   0   0   0    0    0   0   0   0    0    0   0   0   0   1   0   0
1         0   0   0   0    0    0   0   0   0    0    0   0   0   0   0   1   0
2         0   0   0   0    0    0   0   0   0    0    0   0   0   0   0   1   0
3         0   0   0   0    0    0   0   0   0    0    0   0   0   0   0   0   0
4         0   0   0   0    0    0   0   1   0   0    0    0   0   0   0   0   0
```



```
In [297... dummies('WindDir3pm').sum(axis=0)

Out[297...  ENE      7857
            ESE      8505
            N       8890
            NE      8263
            NNE     6590
            NNW     7870
            NW      8610
            S       9926
            SE      10838
            SSE     9399
            SSW     8156
            SW      9354
            W       10110
            WNW    8874
            WSW    9518
            NaN     4228
            dtype: int64
```

Exploring RainToday variable

```
In [298... len_labels('RainToday')
RainToday contains 3 labels

In [299... labels('RainToday')
['No' 'Yes' 'nan']

In [300... df.RainToday.value_counts()

Out[300... RainToday
      No      110319
      Yes     31880
      Name: count, dtype: int64

In [301... dummies('RainToday').head()
```

```
Out[301...   Yes  NaN
0      0    0
1      0    0
2      0    0
3      0    0
4      0    0
```

```
In [302... dummies('RainToday').sum(axis=0)
```

```
Out[302...  Yes    31880
NaN     3261
dtype: int64
```

There are 3261 missing values in the RainToday variable

Numerical variables

```
In [303... # find numerical variables
numerical = [var for var in df.columns if df[var].dtype != 'O']
print('There are {} numerical variables'.format(
    len(numerical)))
print('The numerical variables are', numerical)
```

There are 19 numerical variables

The numerical variables are ['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am', 'Temp3pm', 'Year', 'Month', 'Day']

```
In [304... # view numerical variables
```

```
df[numerical].head()
```

```
Out[304...   MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine  WindGustSpeed  WindSpeed9
0      13.4      22.9       0.6        NaN         NaN          44.0           2
1       7.4      25.1       0.0        NaN         NaN          44.0
2      12.9      25.7       0.0        NaN         NaN          46.0           1
3       9.2      28.0       0.0        NaN         NaN          24.0           1
4      17.5      32.3       1.0        NaN         NaN          41.0
```

Problems within numerical variables

Missing values

```
In [305... df[numerical].isnull().sum()
```

```
Out[305... MinTemp      1485
       MaxTemp      1261
       Rainfall     3261
       Evaporation  62790
       Sunshine    69835
       WindGustSpeed 10263
       WindSpeed9am 1767
       WindSpeed3pm 3062
       Humidity9am  2654
       Humidity3pm  4507
       Pressure9am  15065
       Pressure3pm  15028
       Cloud9am     55888
       Cloud3pm     59358
       Temp9am      1767
       Temp3pm      3609
       Year          0
       Month         0
       Day           0
       dtype: int64
```

16 out of 19 have missing values

Outliers

```
In [306... # summary stats
print(round(df[numerical].describe(), 2))
```

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed
count	143975.00	144199.00	142199.00	82670.00	75625.00	135197.
mean	12.19	23.22	2.36	5.47	7.61	40.
std	6.40	7.12	8.48	4.19	3.79	13.
min	-8.50	-4.80	0.00	0.00	0.00	6.
25%	7.60	17.90	0.00	2.60	4.80	31.
50%	12.00	22.60	0.00	4.80	8.40	39.
75%	16.90	28.20	0.80	7.40	10.60	48.
max	33.90	48.10	371.00	145.00	14.50	135.
	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	
count	143693.00	142398.00	142806.00	140953.00	130395.00	
mean	14.04	18.66	68.88	51.54	1017.65	
std	8.92	8.81	19.03	20.80	7.11	
min	0.00	0.00	0.00	0.00	980.50	
25%	7.00	13.00	57.00	37.00	1012.90	
50%	13.00	19.00	70.00	52.00	1017.60	
75%	19.00	24.00	83.00	66.00	1022.40	
max	130.00	87.00	100.00	100.00	1041.00	
	Pressure3pm	Cloud9am	Cloud3pm	Temp9am	Temp3pm	Year
count	130432.00	89572.00	86102.00	143693.00	141851.00	145460.00
mean	1015.26	4.45	4.51	16.99	21.68	2012.77
std	7.04	2.89	2.72	6.49	6.94	2.54
min	977.10	0.00	0.00	-7.20	-5.40	2007.00
25%	1010.40	1.00	2.00	12.30	16.60	2011.00
50%	1015.20	5.00	5.00	16.70	21.10	2013.00
75%	1020.00	7.00	7.00	21.60	26.40	2015.00
max	1039.60	9.00	9.00	40.20	46.70	2017.00
	Month	Day				
count	145460.00	145460.00				
mean	6.40	15.71				
std	3.43	8.79				
min	1.00	1.00				
25%	3.00	8.00				
50%	6.00	16.00				
75%	9.00	23.00				
max	12.00	31.00				

Windspeed, rainfall and evaporation seem to have outliers - draw a box plot to visualise this

```
In [307]: plt.figure(figsize=(15,10))

def boxplot(n, field):

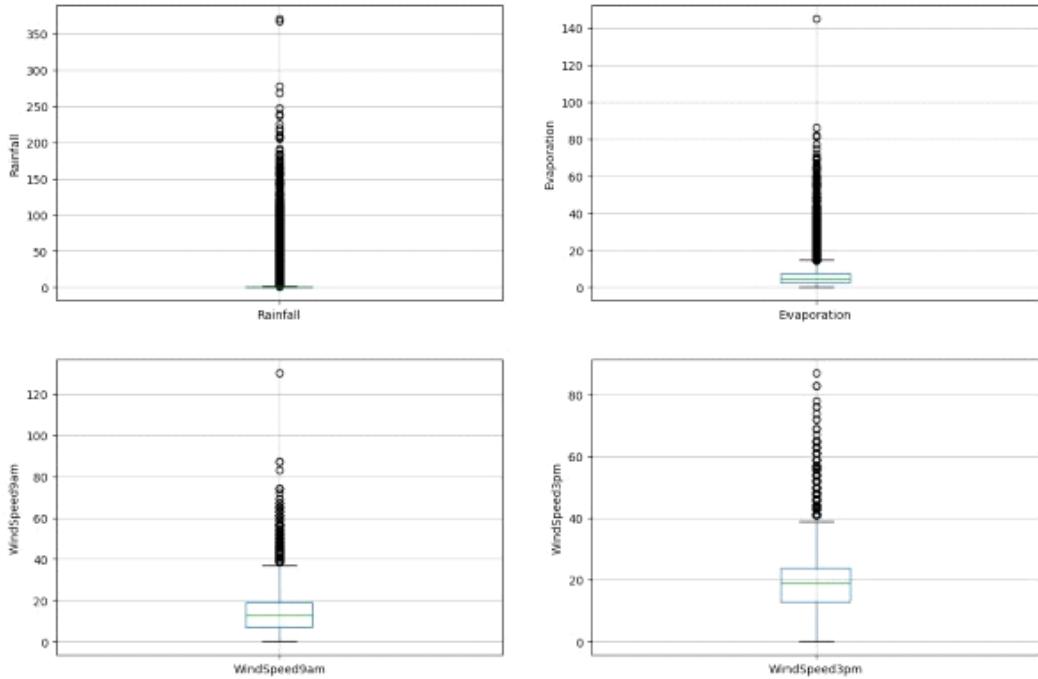
    var = df[field]
    plt.subplot(2, 2, n)
```

```

fig = df.boxplot(column=field)
fig.set_title('')
fig.set_ylabel(field)

boxplot(1, 'Rainfall')
boxplot(2, 'Evaporation')
boxplot(3, 'WindSpeed9am')
boxplot(4, 'WindSpeed3pm')

```



These plots confirm that there are a lot of outliers in these variables!

```

In [308]: # plot histogram to check distribution

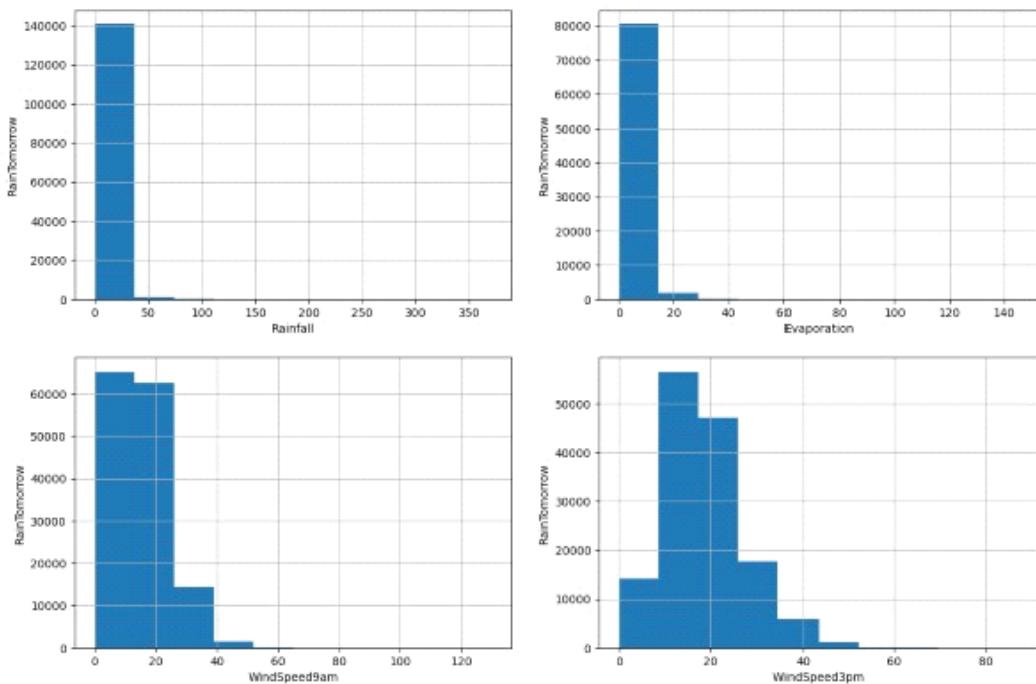
plt.figure(figsize=(15,10))

def histogram(n, field):

    var = df[field]
    plt.subplot(2, 2, n)
    fig = var.hist(bins=10)
    fig.set_xlabel(field)
    fig.set_ylabel('RainTomorrow')

histogram(1, 'Rainfall')
histogram(2, 'Evaporation')
histogram(3, 'WindSpeed9am')
histogram(4, 'WindSpeed3pm')

```



All four variables are skewed and not of normal distribution, so we will use the interquartile range to find outliers

```
In [309]: # find outliers for Rainfall
```

```
def iqr(field):
    var = df[field]
    IQR = var.quantile(0.75) - var.quantile(0.25)
    Lower_fence = var.quantile(0.25) - (IQR * 3)
    Upper_fence = var.quantile(0.75) + (IQR * 3)
    print(f'{field} outliers are values\'\n' f'< {Lower_fence} or > {Upper_fence}\')
```

```
iqr('Rainfall')
```

Rainfall outliers are values< -2.4000000000000004 or > 3.2

$0.0 \leq \text{Rainfall} \leq 3.72$ so outliers > 3.2

```
In [310]: iqr('Evaporation')
```

Evaporation outliers are values< -11.800000000000002 or > 21.800000000000004

$0.0 \leq \text{Evaporation} \leq 145.0$ so outliers > 21.8

```
In [311]: iqr('WindSpeed9am')
```

WindSpeed9am outliers are values< -29.0 or > 55.0

$0.0 \leq \text{WindSpeed9am} \leq 130.0$ so outliers > 55.0

```
In [312]: iqr('WindSpeed3pm')
```

WindSpeed3pm outliers are values< -20.0 or > 57.0

0.0 ≤ Rainfall ≤ 87.0 so outliers > 57.0

Declare vector and target variable

```
In [313...]: x = df.drop(['RainTomorrow'], axis=1)  
y = df['RainTomorrow']
```

Split data into training and test set

```
In [314...]: from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(  
    X,  
    y,  
    test_size = 0.2,  
    random_state = 0)  
  
In [315...]: X_train.shape, X_test.shape  
  
Out[315...]: ((116368, 24), (29092, 24))
```

Feature engineering

```
In [316...]: X_train.dtypes  
  
Out[316...]: Location          object  
MinTemp           float64  
MaxTemp           float64  
Rainfall           float64  
Evaporation       float64  
Sunshine           float64  
WindGustDir        object  
WindGustSpeed      float64  
WindDir9am         object  
WindDir3pm         object  
WindSpeed9am       float64  
WindSpeed3pm       float64  
Humidity9am        float64  
Humidity3pm        float64  
Pressure9am        float64  
Pressure3pm        float64  
Cloud9am            float64  
Cloud3pm            float64  
Temp9am             float64  
Temp3pm             float64  
RainToday           object  
Year                int32  
Month               int32  
Day                 int32  
dtype: object
```

```
In [317...]: categorical = [col for col in X_train.columns if X_train[col].dtypes == 'object']
```

categorical

```
Out[317... ['Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday']
```

```
In [318... numerical = [col for col in X_train.columns if X_train[col].dtypes != 'O']  
numerical
```

```
Out[318... ['MinTemp',  
           'MaxTemp',  
           'Rainfall',  
           'Evaporation',  
           'Sunshine',  
           'WindGustSpeed',  
           'WindSpeed9am',  
           'WindSpeed3pm',  
           'Humidity9am',  
           'Humidity3pm',  
           'Pressure9am',  
           'Pressure3pm',  
           'Cloud9am',  
           'Cloud3pm',  
           'Temp9am',  
           'Temp3pm',  
           'Year',  
           'Month',  
           'Day']
```

Engineering missing variables in numerical variables

```
In [319... X_train[numerical].isnull().sum()
```

```
Out[319... MinTemp      1183  
          MaxTemp      1019  
          Rainfall     2617  
          Evaporation  50355  
          Sunshine    55899  
          WindGustSpeed 8218  
          WindSpeed9am 1409  
          WindSpeed3pm 2456  
          Humidity9am  2147  
          Humidity3pm  3598  
          Pressure9am 12091  
          Pressure3pm 12064  
          Cloud9am     44796  
          Cloud3pm     47557  
          Temp9am      1415  
          Temp3pm      2865  
          Year          0  
          Month         0  
          Day           0  
          dtype: int64
```

```
In [320... X_test[numerical].isnull().sum()
```

```
Out[320]: MinTemp      302
          MaxTemp       242
          Rainfall      644
          Evaporation   12435
          Sunshine     13936
          WindGustSpeed 2045
          WindSpeed9am  358
          WindSpeed3pm  606
          Humidity9am   507
          Humidity3pm   909
          Pressure9am   2974
          Pressure3pm   2964
          Cloud9am      11092
          Cloud3pm      11801
          Temp9am       352
          Temp3pm       744
          Year           0
          Month          0
          Day            0
          dtype: int64
```

```
In [321]: # print percentage of missing values
# in numerical variables in training set

for col in numerical:

    if X_train[col].isnull().mean() > 0:

        print(col,
              round(X_train[col].isnull().mean(),
                    4))
```

```
MinTemp 0.0102
MaxTemp 0.0088
Rainfall 0.0225
Evaporation 0.4327
Sunshine 0.4804
WindGustSpeed 0.0706
WindSpeed9am 0.0121
WindSpeed3pm 0.0211
Humidity9am 0.0185
Humidity3pm 0.0309
Pressure9am 0.1039
Pressure3pm 0.1037
Cloud9am 0.385
Cloud3pm 0.4087
Temp9am 0.0122
Temp3pm 0.0246
```

```
In [322]: # to deal with missing values,
# fill the missing values with the median
# this is called median imputation

for df1 in [X_train, X_test]:

    for col in numerical:

        col_median = X_train[col].median()
        df1[col].fillna(col_median, inplace=True)
```

```
In [323... X_train[numerical].isnull().sum()

Out[323... MinTemp      0
          MaxTemp      0
          Rainfall     0
          Evaporation  0
          Sunshine    0
          WindGustSpeed 0
          WindSpeed9am  0
          WindSpeed3pm  0
          Humidity9am   0
          Humidity3pm   0
          Pressure9am   0
          Pressure3pm   0
          Cloud9am      0
          Cloud3pm      0
          Temp9am       0
          Temp3pm       0
          Year          0
          Month         0
          Day           0
          dtype: int64
```

Now there are no missing values in X_train

```
In [324... X_test[numerical].isnull().sum()

Out[324... MinTemp      0
          MaxTemp      0
          Rainfall     0
          Evaporation  0
          Sunshine    0
          WindGustSpeed 0
          WindSpeed9am  0
          WindSpeed3pm  0
          Humidity9am   0
          Humidity3pm   0
          Pressure9am   0
          Pressure3pm   0
          Cloud9am      0
          Cloud3pm      0
          Temp9am       0
          Temp3pm       0
          Year          0
          Month         0
          Day           0
          dtype: int64
```

There are also no missing values in X_test

Engineering missing values in categorical variables

```
In [325... # % of missing values in categorical vars in training set
          X_train[categorical].isnull().mean()
```

```
Out[325]: Location      0.000000
          WindGustDir   0.071068
          WindDir9am    0.072597
          WindDir3pm    0.028951
          RainToday     0.022489
          dtype: float64
```

```
In [326]: for col in categorical:
            if X_train[col].isnull().mean() > 0:
                print(col, (X_train[col].isnull().mean()))
WindGustDir 0.07106764746322013
WindDir9am 0.07259727760208992
WindDir3pm 0.028951258077822083
RainToday 0.02248900041248453
```

```
In [327]: # missing values are filled in with the most frequent value (mode)

for df2 in [X_train, X_test]:
    df2['WindGustDir'].fillna(
        X_train['WindGustDir'].mode()[0],
        inplace=True)
    df2['WindDir9am'].fillna(
        X_train['WindDir9am'].mode()[0],
        inplace=True)
    df2['WindDir3pm'].fillna(
        X_train['WindDir3pm'].mode()[0],
        inplace=True)
    df2['RainToday'].fillna(
        X_train['RainToday'].mode()[0],
        inplace=True)
```

```
In [328]: X_train[categorical].isnull().sum()
```

```
Out[328]: Location      0
          WindGustDir   0
          WindDir9am    0
          WindDir3pm    0
          RainToday     0
          dtype: int64
```

```
In [329]: X_test[categorical].isnull().sum()
```

```
Out[329]: Location      0
          WindGustDir   0
          WindDir9am    0
          WindDir3pm    0
          RainToday     0
          dtype: int64
```

```
In [330]: # checking for missing values
# in whole of training and testing sets

X_train.isnull().sum()
```

```
Out[330]: Location      0  
          MinTemp       0  
          MaxTemp       0  
          Rainfall       0  
          Evaporation    0  
          Sunshine        0  
          WindGustDir     0  
          WindGustSpeed    0  
          WindDir9am      0  
          WindDir3pm      0  
          WindSpeed9am    0  
          WindSpeed3pm    0  
          Humidity9am     0  
          Humidity3pm     0  
          Pressure9am     0  
          Pressure3pm     0  
          Cloud9am        0  
          Cloud3pm        0  
          Temp9am         0  
          Temp3pm         0  
          RainToday        0  
          Year             0  
          Month            0  
          Day              0  
          dtype: int64
```

```
In [331]: X_test.isnull().sum()
```

```
Out[331]: Location      0  
          MinTemp       0  
          MaxTemp       0  
          Rainfall       0  
          Evaporation    0  
          Sunshine        0  
          WindGustDir     0  
          WindGustSpeed    0  
          WindDir9am      0  
          WindDir3pm      0  
          WindSpeed9am    0  
          WindSpeed3pm    0  
          Humidity9am     0  
          Humidity3pm     0  
          Pressure9am     0  
          Pressure3pm     0  
          Cloud9am        0  
          Cloud3pm        0  
          Temp9am         0  
          Temp3pm         0  
          RainToday        0  
          Year             0  
          Month            0  
          Day              0  
          dtype: int64
```

Engineering outliers in numerical variables

Rainfall, Evaporation, WindSpeed9am and WindSpeed3am contain outliers. These will be removed by capping maximum values.

```
In [332... def max_value(df3, variable, top):  
    return np.where(df3[variable] > top, top, df3[variable])  
  
for df3 in [X_train, X_test]:  
    df3['Rainfall'] = max_value(  
        df3,  
        'Rainfall',  
        3.2)  
    df3['Evaporation'] = max_value(  
        df3,  
        'Evaporation',  
        21.8)  
    df3['WindSpeed9am'] = max_value(  
        df3,  
        'WindSpeed9am',  
        55)  
    df3['WindSpeed3pm'] = max_value(  
        df3,  
        'WindSpeed3pm',  
        57)
```

```
In [333... X_train.Rainfall.max(), X_test.Rainfall.max()  
Out[333... (3.2, 3.2)
```

```
In [334... X_train.Evaporation.max(), X_test.Evaporation.max()  
Out[334... (21.8, 21.8)
```

```
In [335... X_train.WindSpeed9am.max(), X_test.WindSpeed9am.max()  
Out[335... (55.0, 55.0)
```

```
In [336... X_train.WindSpeed3pm.max(), X_test.WindSpeed3pm.max()  
Out[336... (57.0, 57.0)
```

```
In [337... X_train[numerical].describe()
```

```
Out[337...  
      MinTemp      MaxTemp      Rainfall      Evaporation      Sunshine  
count  116368.000000  116368.000000  116368.000000  116368.000000  116368.000000  
mean   12.190189    23.203107    0.670800    5.093362    7.982476  
std    6.366893    7.085408    1.181512    2.800200    2.761639  
min   -8.500000   -4.800000   0.000000   0.000000   0.000000  
25%    7.700000   18.000000   0.000000   4.000000   8.200000  
50%   12.000000   22.600000   0.000000   4.700000   8.400000  
75%   16.800000   28.200000   0.600000   5.200000   8.600000  
max   31.900000   48.100000   3.200000  21.800000  14.500000
```

Can now see that outliers in Rainfall, Evaporation, WindSpeed9am and WindSpeed3pm are capped

Encode categorical variables

```
In [338... categorical
```

```
Out[338... ['Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday']
```

```
In [339... X_train[categorical].head()
```

	Location	WindGustDir	WindDir9am	WindDir3pm	RainToday
22926	NorfolkIsland	ESE	ESE	ESE	No
80735	Watsonia	NE	NNW	NNE	No
121764	Perth	SW	N	SW	Yes
139821	Darwin	ESE	ESE	E	No
1867	Albury	E	ESE	E	Yes

```
In [340... # encode RainToday
```

```
import category_encoders as ce
encoder = ce.BinaryEncoder(cols=['RainToday'])
X_train = encoder.fit_transform(X_train)
X_test = encoder.transform(X_test)
```

```
In [341... X_train.head()
```

	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGust
22926	NorfolkIsland	18.8	23.7	0.2	5.0	7.3	6.8
80735	Watsonia	9.3	24.0	0.2	1.6	10.9	10.9
121764	Perth	10.9	22.2	1.4	1.2	9.6	9.6
139821	Darwin	19.3	29.9	0.0	9.2	11.0	11.0
1867	Albury	15.7	17.6	3.2	4.7	8.4	8.4

5 rows × 25 columns

RainToday has been made into RainToday_0 and RainToday_1

```
In [342... X_train = pd.concat([
    X_train[numerical],
    X_train[['RainToday_0', 'RainToday_1']],
    pd.get_dummies(
        X_train.Location,
        dtype=int),
    pd.get_dummies(
        X_train.WindGustDir,
```

```
        dtype=int),
pd.get_dummies(
    X_train.WindDir9am,
    dtype=int),
pd.get_dummies(
    X_train.WindDir3pm,
    dtype=int)],
axis = 1)
```

In [343]: X_train.head()

Out[343]:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	WindS
22926	18.8	23.7	0.2	5.0	7.3	52.0	
80735	9.3	24.0	0.2	1.6	10.9	48.0	
121764	10.9	22.2	1.4	1.2	9.6	26.0	
139821	19.3	29.9	0.0	9.2	11.0	43.0	
1867	15.7	17.6	3.2	4.7	8.4	20.0	

5 rows × 118 columns

X_train dataset created!

In [344]: X_test = pd.concat(
 [X_test[numerical], X_test[['RainToday_0', 'RainToday_1']],
 pd.get_dummies(
 X_test.Location,
 dtype=int),
 pd.get_dummies(
 X_test.WindGustDir,
 dtype=int),
 pd.get_dummies(
 X_test.WindDir9am,
 dtype=int),
 pd.get_dummies(
 X_test.WindDir3pm,
 dtype=int)],
 axis = 1)

In [345]: X_test.head()

Out[345]:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	WindS
138175	21.9	39.4	1.6	11.2	11.5	57.0	
38638	20.5	37.5	0.0	9.2	8.4	59.0	
124058	5.1	17.2	0.2	4.7	8.4	50.0	
99214	11.9	16.8	1.0	4.7	8.4	28.0	
25097	7.5	21.3	0.0	4.7	8.4	15.0	

5 rows × 118 columns

X_test dataset created!

Now the datasets are ready for model building. First we need to map all of the feature variables onto the same scale - feature scaling.

Feature scaling

In [346... X_train.describe()

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGust
count	116368.000000	116368.000000	116368.000000	116368.000000	116368.000000	116368.000000
mean	12.190189	23.203107	0.670800	5.093362	7.982476	1.830000
std	6.366893	7.085408	1.181512	2.800200	2.761639	1.000000
min	-8.500000	-4.800000	0.000000	0.000000	0.000000	0.000000
25%	7.700000	18.000000	0.000000	4.000000	8.200000	0.000000
50%	12.000000	22.600000	0.000000	4.700000	8.400000	0.000000
75%	16.800000	28.200000	0.600000	5.200000	8.600000	0.000000
max	31.900000	48.100000	3.200000	21.800000	14.500000	1.000000

8 rows × 118 columns

In [347... cols = X_train.columns

```
In [348... from sklearn.preprocessing import MinMaxScaler  
scaler = MinMaxScaler()  
X_train = scaler.fit_transform(X_train)  
X_test = scaler.transform(X_test)
```

In [349... X_train = pd.DataFrame(X_train, columns=cols)

In [350... X_test = pd.DataFrame(X_test, columns=cols)

In [351... X_train.describe()

```
Out[351...      MinTemp      MaxTemp      Rainfall      Evaporation      Sunshine
count  116368.000000  116368.000000  116368.000000  116368.000000  116368.000000
mean    0.512133     0.529359     0.209625     0.233640     0.550516
std     0.157596     0.133940     0.369223     0.128450     0.190458
min    0.000000     0.000000     0.000000     0.000000     0.000000
25%    0.400990     0.431002     0.000000     0.183486     0.565517
50%    0.507426     0.517958     0.000000     0.215596     0.579310
75%    0.626238     0.623819     0.187500     0.238532     0.593103
max    1.000000     1.000000     1.000000     1.000000     1.000000
8 rows × 118 columns
```

X_train dataset is now ready

Model training!

Removing missing values in both X_train and y_train

```
In [352... X_train.shape
```

```
Out[352... (116368, 118)
```

```
In [353... y_train.head()
```

```
Out[353... 22926    Yes
          80735    No
          121764   No
          139821   No
          1867     Yes
Name: RainTomorrow, dtype: object
```

```
In [354... y_train = y_train.reset_index(drop=True)
# making sure numbers are in correct order for both
# so that they can be merged
X_train = X_train.reset_index(drop=True)
y_train = pd.get_dummies(y_train, dtype=int)
# making Yes & No into 1s & 0s

y_train = y_train.drop(['No'], axis=1)
# because for some reason get_dummies
# makes it into a Yes column and a No column
# which are the exact opposite of each other
y_train.head()
```

```
Out[354... Yes
      0   1
      1   0
      2   0
      3   0
      4   1
```

```
In [355... df3 = pd.concat([X_train, y_train], axis=1)
df3.head()
```

```
Out[355...  (MinTemp,) (MaxTemp,) (Rainfall,) (Evaporation,) (Sunshine,) (WindGustSpeed,) (
      0    0.675743    0.538752     0.0625    0.229358    0.503448    0.356589
      1    0.440594    0.544423     0.0625    0.073394    0.751724    0.325581
      2    0.480198    0.510397     0.4375    0.055046    0.662069    0.155039
      3    0.688119    0.655955     0.0000    0.422018    0.758621    0.286822
      4    0.599010    0.423440     1.0000    0.215596    0.579310    0.108527
)
5 rows × 119 columns
```

```
In [356... df3 = df3.dropna()
df3.head()
```

```
Out[356...  (MinTemp,) (MaxTemp,) (Rainfall,) (Evaporation,) (Sunshine,) (WindGustSpeed,) (
      0    0.675743    0.538752     0.0625    0.229358    0.503448    0.356589
      1    0.440594    0.544423     0.0625    0.073394    0.751724    0.325581
      2    0.480198    0.510397     0.4375    0.055046    0.662069    0.155039
      3    0.688119    0.655955     0.0000    0.422018    0.758621    0.286822
      4    0.599010    0.423440     1.0000    0.215596    0.579310    0.108527
)
5 rows × 119 columns
```

```
In [357... y_train = df3[['Yes']]
X_train = df3.drop(['Yes'], axis=1)
X_train.head()
```

```
Out[357... (MinTemp,) (MaxTemp,) (Rainfall,) (Evaporation,) (Sunshine,) (WindGustSpeed) (  
0 0.675743 0.538752 0.0625 0.229358 0.503448 0.356589  
1 0.440594 0.544423 0.0625 0.073394 0.751724 0.325581  
2 0.480198 0.510397 0.4375 0.055046 0.662069 0.155039  
3 0.688119 0.655955 0.0000 0.422018 0.758621 0.286822  
4 0.599010 0.423440 1.0000 0.215596 0.579310 0.108527  
5 rows × 118 columns
```

```
In [358... y_train = y_train.rename(  
    columns={'Yes':'RainTomorrow'})  
y_train.head()
```

```
Out[358... RainTomorrow  
0 1  
1 0  
2 0  
3 0  
4 1
```

Starting and fitting the model

```
In [359... from sklearn.linear_model import LogisticRegression  
  
# instantiate the model  
logreg = LogisticRegression(  
    solver='liblinear',  
    random_state=0)  
  
# fit the model  
logreg.fit(X_train, y_train)
```

```
Out[359... LogisticRegression  
LogisticRegression(random_state=0, solver='liblinear')
```

```
In [360... y_pred_test = logreg.predict(X_test)  
y_pred_test
```

```
Out[360... array([0, 0, 0, ..., 1, 0, 0])
```

`predict_proba` method

This method gives probability for the target variable. 0 is for probability of no rain and 1 is for probability of rain

```
In [361... logreg.predict_proba(X_test)[:,0]
Out[361... array([0.83215823, 0.74551076, 0.79860387, ..., 0.42025444, 0.6575465 ,
0.96954917])
In [362... logreg.predict_proba(X_test)[:,1]
Out[362... array([0.16784177, 0.25448924, 0.20139613, ..., 0.57974556, 0.3424535 ,
0.03045083])
```

Check accuracy score

```
In [363... y_test.head()
Out[363... 138175    Yes
38638     Yes
124058     No
99214     No
25097     No
Name: RainTomorrow, dtype: object
In [364... y_test = y_test.reset_index(drop=True)
y_test = np.array([v == 'Yes' for v in y_test])
In [365... y_test.astype(int)
Out[365... array([1, 1, 0, ..., 1, 0, 0])
In [366... print(y_pred_test)
[0 0 0 ... 1 0 0]
In [367... from sklearn.metrics import accuracy_score
print('Model accuracy score: {:.4f}'.format(
accuracy_score(y_test, y_pred_test)))
Model accuracy score: 0.8484
```

Compare train-set and test-set accuracy

to check for overfitting & underfitting

```
In [368... y_pred_train = logreg.predict(X_train)
y_pred_train
Out[368... array([0, 0, 0, ..., 0, 0, 0])
In [369... print('Training set accuracy score: {:.4f}'.format(
accuracy_score(y_train, y_pred_train)))
Training set accuracy score: 0.8488
```

Check for overfitting & underfitting

```
In [370... print('Training set score: {:.4f}'.format(
    logreg.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(
    logreg.score(X_test, y_test)))
```

```
Training set score: 0.8488
Test set score: 0.8484
```

Since these values are comparable there is no question of overfitting. However, the model performance on both training and test set is comparable so it is likely that the model is underfitting.

Logistic regression uses a default value of $C = 1.0$. Increasing C will mean that model trusts the training data a lot and will fit more closely with the training data, thus it is less likely that the model is underfitting.

```
In [371... # fit the model with C = 100
# instantiate model
logreg100 = LogisticRegression(C = 100,
                                solver = 'liblinear',
                                random_state = 0)

# fit the model
logreg100.fit(X_train, y_train)
```

Out[371... ▾ LogisticRegression ⓘ ?
LogisticRegression(C=100, random_state=0, solver='liblinear')

```
In [372... # print scores on training and test set
print('Training set score: {:.4f}'.format(
    logreg100.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(
    logreg100.score(X_test, y_test)))
```

```
Training set score: 0.8489
Test set score: 0.8491
```

This increases the training set score slightly and the test set score by a bit more, so a complex model (one with a higher value of C) should perform better.
What if we decreased C to 0.01?

```
In [373... logreg001 = LogisticRegression(C = 0.01,
                                       solver = 'liblinear',
                                       random_state = 0)

logreg001.fit(X_train, y_train)
```

Out[373... ▾ LogisticRegression ⓘ ?
LogisticRegression(C=0.01, random_state=0, solver='liblinear')

```
In [374]: print('Training set score: {:.4f}'.format(
    logreg001.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(
    logreg001.score(X_test, y_test)))
```

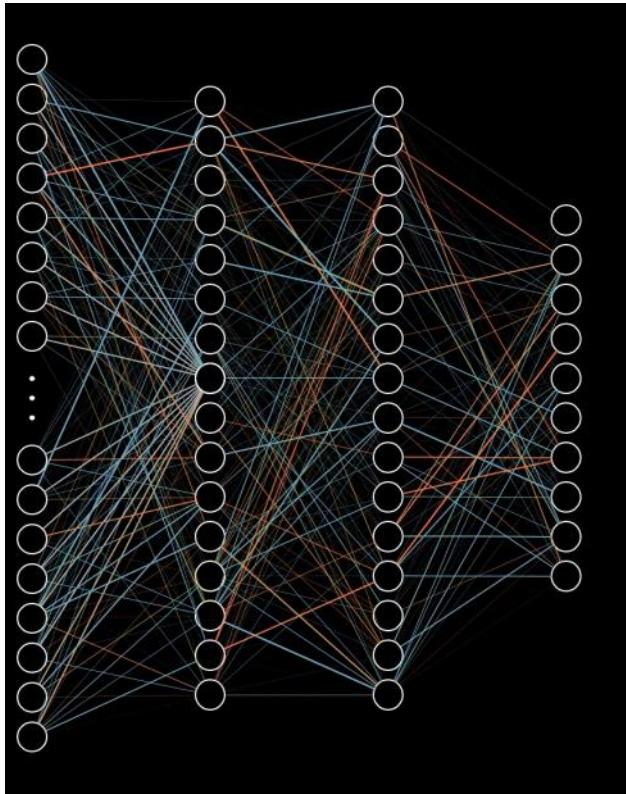
Training set score: 0.8427
Test set score: 0.8418

This gives a lower test set score. Therefore, C should be higher to get better results.

Neural network [36]

Tuesday, August 13, 2024 1:22 PM

In image processing



The circles represent neurons
and the lines are the connections
between them.

Neuron – thing that holds a number between 0 and 1

FIRST LAYER: Network starts with 1 neuron corresponding with a pixel in an image. The neuron holds a number that represents the greyscale value of the pixel. (0 for black – 1 for white).

- Number is called **activation**

HIDDEN LAYERS:

Activations in one layer determine the activations in the next layer(s)

LAST LAYER: Number of neurons = number of categories. The activation of each neuron represents how "sure" the system is that the image belongs to this category. The brightest neuron, or activation that is closest to 1, will be the output, or category that the system "thinks" the input is in.

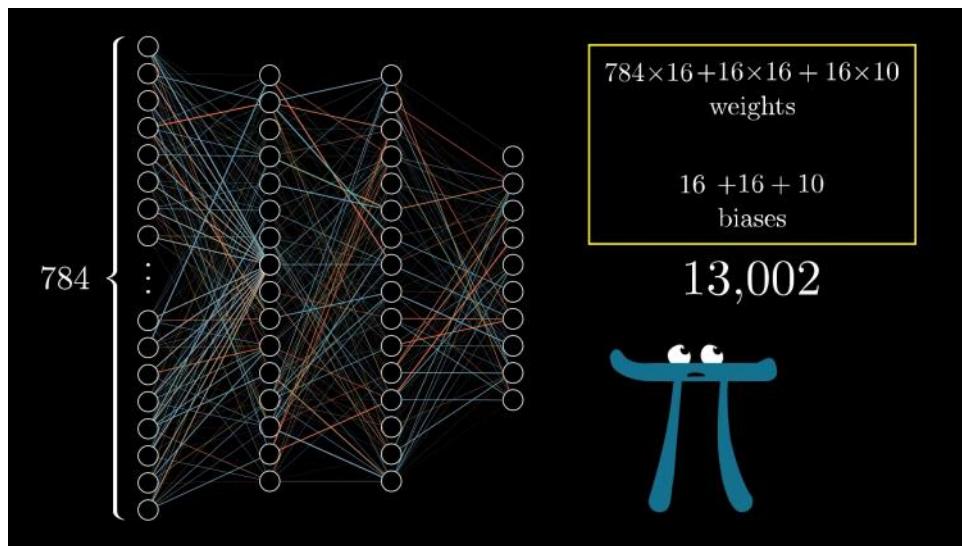
Why layers?

Imagine that you are training a neural network to recognise handwritten numbers. A way to illustrate the layers is that the penultimate layer is made up of a neuron representing a loop or a line in a particular place in the square of pixels. If the input is an 8, the neurons representing a loop at the top and a loop at the bottom will be activated, which causes the neuron in the last layer representing 8 to be activated.

So how does the system recognise the loops? In the previous layer, each neuron might represent a curve or line in a particular place in the square of pixels. In other words, the loop is broken down into edges. And so on. The goal is to have some mechanism that combines pixels into edges into patterns

into digits.

Weights



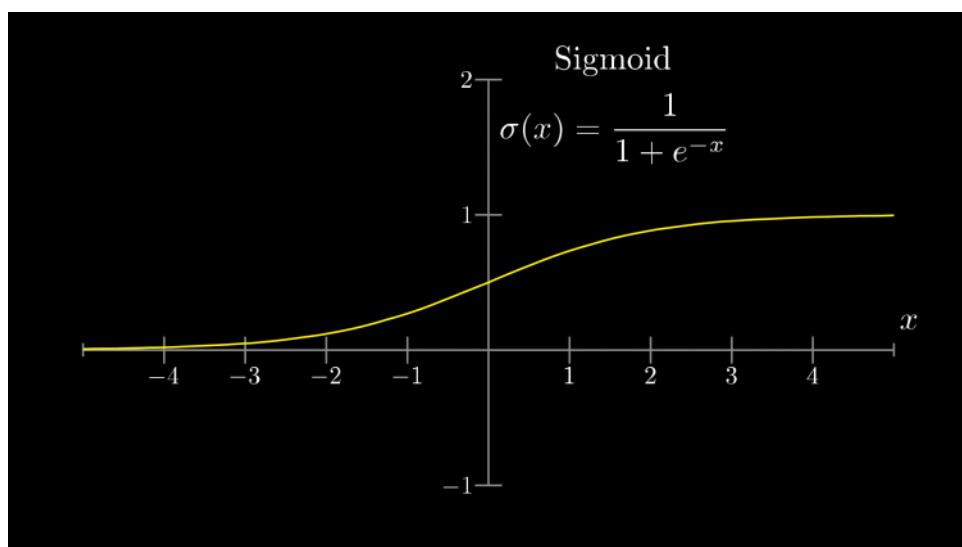
The weights are represented by the colour of the connections between neurons

Say we want to detect a set of black pixels in a specific region, what parameters should there be? A **weight** is assigned to each of the connections between neurons in different layers. The activation of each neuron is multiplied by its corresponding weight for a specific neuron in the next layer. These are all added together.

Perhaps the neuron must only activate meaningfully when the weighted sum is greater than a specific number - the **bias**. The bias is subtracted from this sum.

However, this sum may be any number, while we need it to be between 0 and 1. A common function used to solve this is the **sigmoid function**.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



As you can see in the graph, very negative values of x will be increasingly close to 0, and very positive values of x will be increasingly close to 1.

The output of the sigmoid function is the activation of the neuron in the next layer.

Equation:

$\sigma(w_1a_1 + w_2a_2 + w_3a_3 + \dots + w_na_n - 10)$, where:

- w is the bias
- a is the activation of the neurons in the previous layer
- 10 is the bias

However, the above equation is just for one neuron in the next layer. We can represent all the neurons in the next layer by using vectors:

$$\sigma(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b})$$
$$\sigma \left(\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \dots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \right)$$

where $w_{k,n}$ is the weight from the n-th neuron in the previous layer to the k-th neuron in the next layer.

Learning

In a neural network, "learning" means finding the right weights and biases (in this case, 13 000 of them) to categorise the image.

Neurons

Since the first layer represents all of the pixels in an image, all of the neurons will depend the input. So a neuron can be more accurately thought of as a **function**. The neural network also output a number based on the input, so it can also be thought of as a very complicated function.

Convolution algorithm [35]

Saturday, June 29, 2024 7:44 PM

Why is this relevant to neural networks? [42]

As I learnt when doing research into neural networks, the first layer of a neural network is the activation of each pixel in an image. However, if the image is any more complex than an extremely basic binary image, you can't just use the activation of each pixel, or the accuracy will be very low. First, you have to put the image through a convolutional algorithm. The object of the convolution operation is to extract high level features such as edges from the image.

Formula

$$y = x * w \rightarrow y[i] = \sum_{k=-\infty}^{\infty} x[i-k]w[k]$$

Why are the extremes ∞ ?

If the input vector's size is smaller than it should be, we put 0s either side to make it the right size - this is called **padding**.

Let x = input variable, with size n

Let w = filter, with size m

$n \leq m$

\therefore input with padding has size $n + 2p$ and formula is

$$y = x * w \rightarrow y[i] = \sum_{k=0}^{m-1} x^{p}[i+m-k]w[k]$$

In [18]: `import numpy as np`

In [65]: `def conv1D(x, w, p=0, s=1):`

```
    """
    x : input vector
    w : filter
    p : padding size
    s : stride
    """

    # stride is by how much the filter is shifted right

    assert len(w) <= len(x), "x should be bigger than w"
    assert p >= 0, "padding must not be negative"

    w_r = np.array(w[::-1]) # rotation of w
    x_padded = np.array(x)

    if p > 0 :
        zeros = np.zeros(shape = p)
        x_padded = np.concatenate([zeros, x_padded, zeros]) # add zeros around original vector so they are same

    out = []
    # iterate through the original array s cells per step
    for i in range(0, int((len(x_padded) - len(w_r))) + 1 , s): # start, length, step
        print(x_padded[i:i + len(w_r)]) # shifting w_r down x
        print(x_padded[i:i + len(w_r)] * w_r) # multiplication
        out.append(np.sum(x_padded[i:i + len(w_r)] * w_r)) # adding together the products and putting in a new

    print("\n")
    return np.array(out)
```

In [66]: `x = [3,6,8,2,1,4,7,9] # input vector`

`w = [4,8,6,3,2] # filter`

`conv1D(x,w,2,1)`

```
[0.  0.  3.  6.  8.]
[0.  0.  18.  0.  32.]
[0.  3.  6.  8.  2.]
[0.  9.  36.  0.  8.]
[3.  6.  8.  2.  1.]
[6.  18.  48.  0.  4.]
[6.  8.  2.  1.  4.]
[12. 24. 12.  0.  16.]
[8.  2.  1.  4.  7.]
[16.  6.  6.  0.  28.]
[2.  1.  4.  7.  9.]
[4.  3.  24.  0.  36.]
[1.  4.  7.  9.  0.]
[2.  12. 42.  0.  0.]
[4.  7.  9.  0.  0.]
[8.  21. 54.  0.  0.]
```

Out[66]: `array([50., 53., 76., 64., 56., 67., 56., 83.])`

In [67]: `np.convolve(x, w, mode = 'same') # using the built-in convolution function in Numpy gives the same answer`

Out[67]: `array([50, 53, 76, 64, 56, 67, 56, 83])`

Processing math: 100%

EPQ launch webinar notes [9]

Wednesday, December 13, 2023 3:03 PM

The EPQ tests our ability to:

- plan
- research
- develop
- critically evaluate

Tips for an artefact:

- think about the demographic
 - how does this impact the choices you make?
- define purpose
- impact
- what is the artefact communicating?
- why is it important / relevant?
- how do you test if you have achieved your goal?

Time management [10]

10 January 2024 15:46

Mistake	Why?	How to solve it
Not writing a to-do list		Write a to-do list Prioritise tasks on the list e.g. A-F coding system For large projects, make sure to break down projects into steps
Not setting goals	Goals give you something to focus on	Set goals
Not prioritising	End up wasting time on unimportant things	Set priorities Determine what tasks are high-yield and/or high priority, and low-value, "fill in" work
Getting distracted	Prevent you from being in a focused state, where you are completely engaged in a task	Identify what distracts you and minimise it
Procrastinating		Break projects down into manageable steps
Multitasking	Can take 20% - 40% longer and quality can be worse	Give tasks your full attention
Not taking breaks		Schedule breaks every 1-2h, take walks
Ineffective scheduling		Schedule high value tasks at times when you work best and low energy jobs in quieter times

Timeline [12]

10 January 2024 15:46

Features:

Events that have happened in the past or are projected to happen in the future

Start & end date

Dates & events / milestones marked

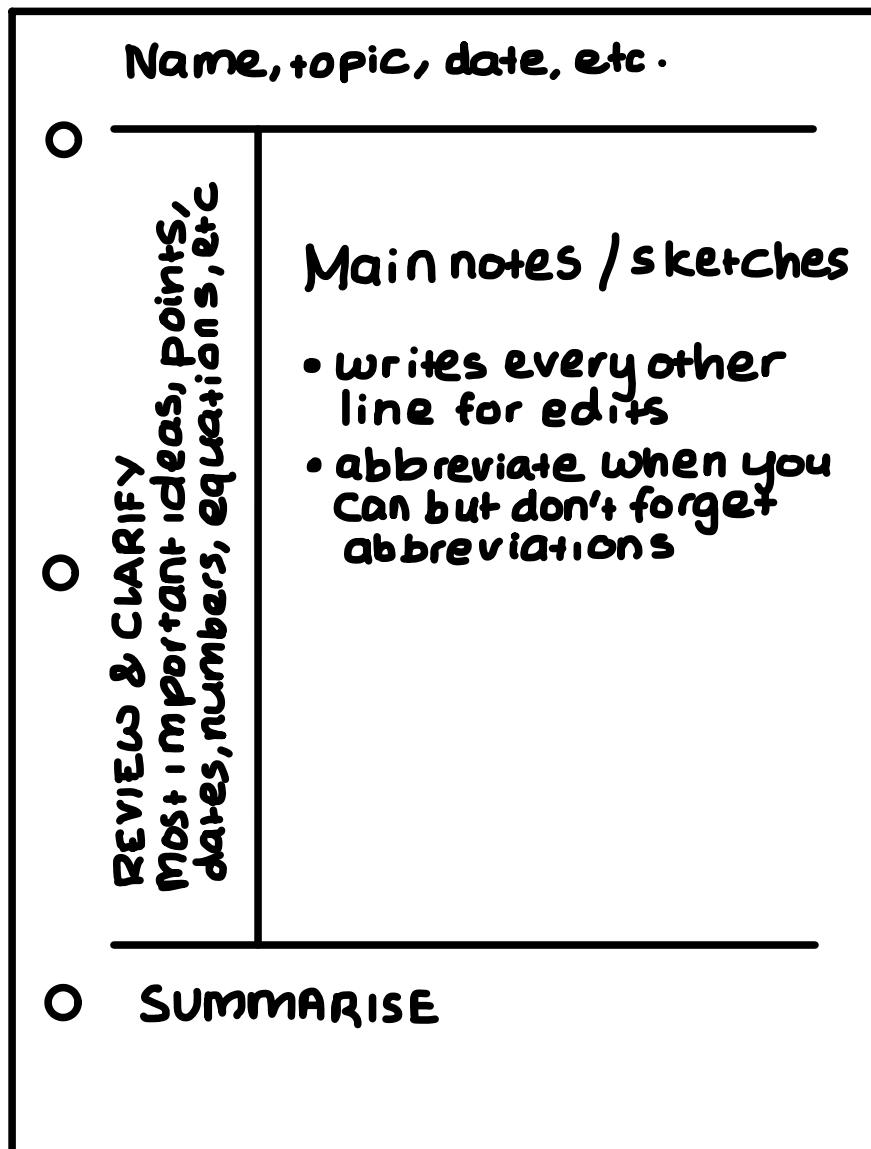
Shows what will happen when

How to make one

1. Tracking or comparing events?
2. List events chronologically
3. Choose time intervals
4. Segment timeline with equal intervals
5. Label start & end date
6. Plot important dates & events
7. Compare projected dates with actual dates

Cornell notes [11]

10 January 2024 15:46



Skills audit November

29 November 2023 15:47

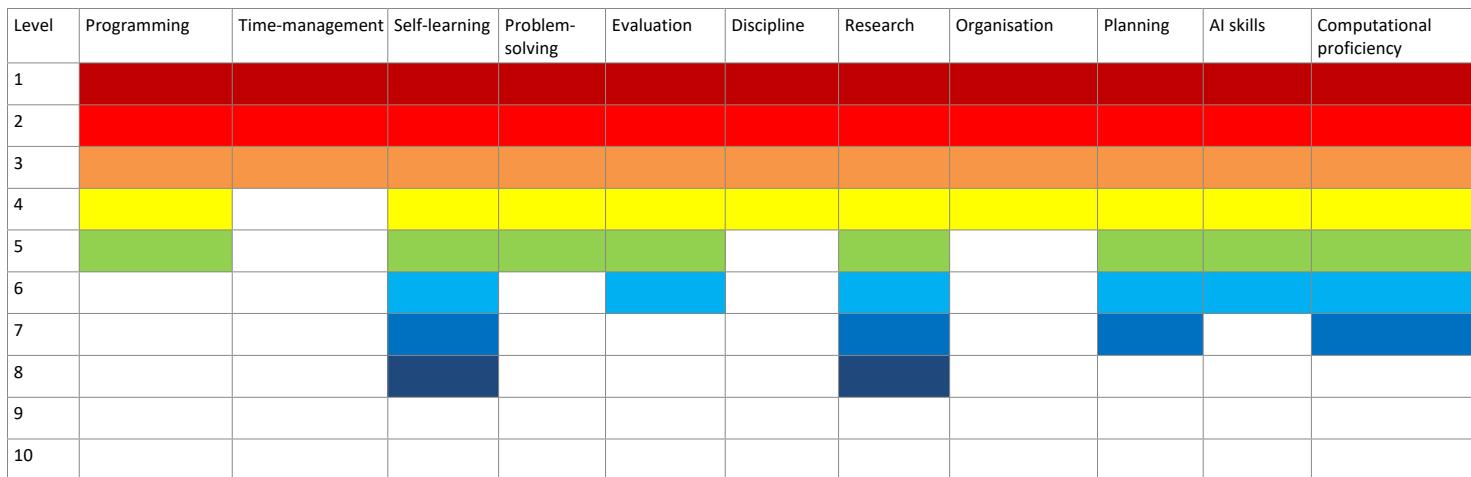
Level	Programming	Time-management	Self-learning	Problem-solving	Evaluation	Discipline	Research	Organisation	Planning	AI skills	Computational proficiency
1											
2											
3				Yellow					Yellow		
4											
5											
6											
7											
8											
9											
10											

Programming	For me this means I'd like to develop my skills in writing commands and scripts that will make the computer do what I want it to. It includes learning programming languages such as Python and using the command line. This is a vital skill in today's computer-orientated world and I've always been interested in learning it. It is something I have almost no experience in.
Time-management	Here I'm referring to using my time effectively and efficiently, and not procrastinating. Although I would say my time-management was good during exams, I think I need to improve on this, especially in a project such as EPQ where it is a very unstructured qualification.
Self-learning	I want to be able to teach myself things such as programming. I do have some experience with this (for example, I taught myself a lot about chess in lockdown, and during GCSEs I taught myself content that my teachers missed out), but I think this is a skill I could develop more.
Problem-solving	Problem-solving is a skill that is required in all of the subjects I do, as well as all of the subjects I am contemplating doing at university. It is of course also essential in everyday life.
Evaluation	This means judging how well my project, and specific aspects of it, are going. I have little experience in documenting my evaluation of a project so I'm hoping to get better at this by reviewing how the project is going regularly.
Discipline	In a project which feels very self-motivated, I need to be able to work hard on tasks without them being laid out for me as they would be in my A-levels. Although in some sense I am quite disciplined when studying, I think a different type of discipline is required for EPQ.
Computational proficiency	Again, something that I lack skill in but is necessary for everyday life. I plan to be able to use the computer from a developer's perspective and not just a user.
Research	I need to learn to use Harvard Referencing, as well as note-taking and evaluating whether a source is relevant, accurate and credible.
Organisation	Learning how to organise the whole project and my time effectively is new to me in this context. I do consider myself an organised person in daily life and schoolwork so I have some experience. Still, organising such a large project will put my skills to the test and hopefully develop them further.
Planning	Although this skill overlaps with organisation, planning is specifically about what I will do in the future, and making sure I complete the tasks that I wanted to do in the past. Again, I think that I have used this skill a lot in my daily life, but I still believe I can advance this ability.
AI skills	This is something very specific to my project. Constructing, training and testing a machine learning algorithm is different to running any script so I think this is separate from programming skills. I have absolutely no experience in this.

Level	Colour	Description
1	Red	No experience in this skill.
2	Red	A small bit of knowledge/experience.
3	Orange	Some experience in the past but not much applying the skills in this context.
4	Yellow	I've started developing this skill.
5	Light Green	I'm learning more about this.
6	Blue	Confident in some aspects but lacking in many.
7	Blue	I've got a lot of experience but there's plenty of room to develop the skills.
8	Dark Blue	I have a good basic base for this skill.
9	Purple	Quite confident in this skill.
10	Dark Purple	Very confident in this skill.

Skills audit March

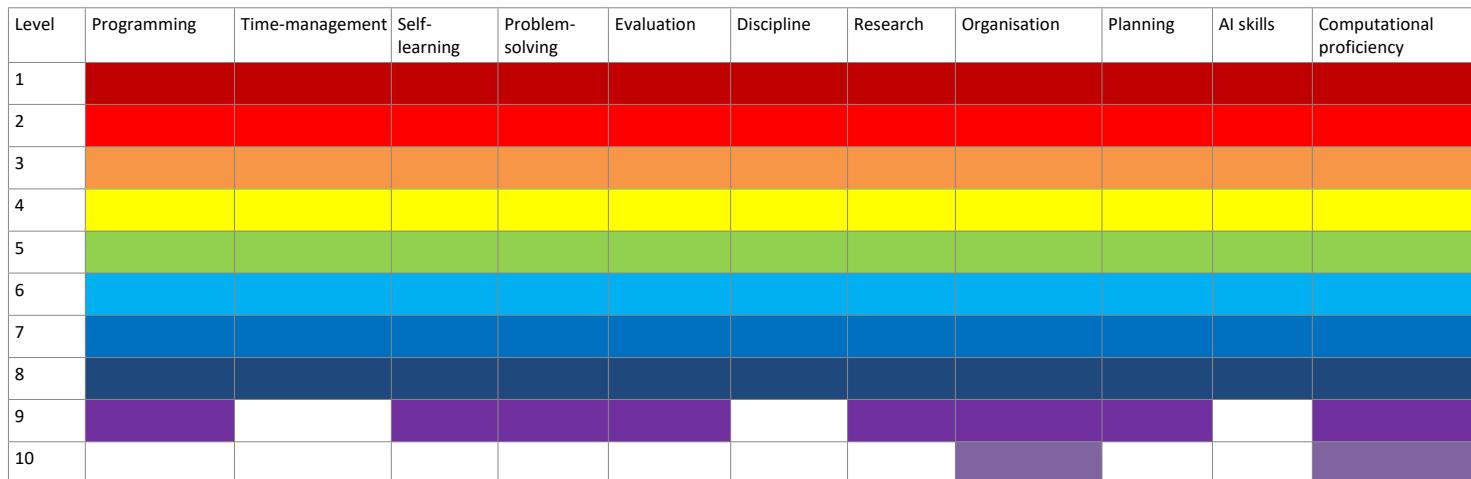
Wednesday, June 19, 2024 3:11 PM



Programming	Having completed the course in Python (Learn Python the Hard Way) I feel a lot more confident in programming. But I do feel that each skill that the course taught was used in a simple manner, which made it easy to learn, but I haven't learnt how to use them in more complex, real-life scenarios.
Time-management	My time-management has improved to an extent, but I still struggle with procrastination – I tend to end up doing schoolwork in the times I dedicate to EPQ.
Self-learning	Since I'm towards the end of the stage of my project where I am developing the necessary AI skills to start making my own algorithm, I've been doing a lot of self-learning. My ability to understand what I need to learn to develop a new skill, and do so independently, has improved a lot.
Problem-solving	I've encountered some minor problems so far and have dealt with them quite well; nonetheless, I think the significant setbacks are yet to come as I begin to write my own script.
Evaluation	Doing the self-reviews have helped me develop my evaluation skills to a degree, but I think I'll advance my skills further when I am evaluating my actual project.
Discipline	As I mentioned before, procrastination has been an issue at times, so there's a lot of room for improvement in disciplining myself.
Computational proficiency	Learning to program has improved my computational proficiency. Also, setting up a digital environment to do my projects in has shown me important know-hows in using a computer as a developer; this includes creating a virtual environment, and using Jupyter Notebook.
Research	I've done a lot of research into programming concepts and machine learning, so this has helped me with my note-taking, making citations, evaluating sources, and finding useful information in sources.
Organisation	It doesn't feel as though I have made as much progress in organisation skills, because I'm in the stage of my project where I am doing a lot of research and I don't know enough to organise the project itself. However, I'm hoping that the next stage will help me further this skill.
Planning	My planning has improved as I believe I have chosen the right approach so far for my project. For example, I've made a timeline and have been sticking to it more or less, so this is evidence of good planning.
AI skills	My AI skills have improved a lot because I've been completing tutorials about numerous different types of ML algorithms. I've also completed research on some important AI concepts. To develop the skill further, I'll apply this experience to making my own project, which will put my skills to the test since doing my own project will be harder than following a step-by-step tutorial.

Skills audit summer

Wednesday, June 19, 2024 3:11 PM



Programming	After completing a full project of my own, my competence in computer programming has improved vastly. I still have a lot to learn, but I now have a good understanding of programming, that puts me in a great position to increase my expertise in the future.
Time-management	Although my time management has gotten a lot better over the course of the project, I would say that this was one of my weaknesses throughout. As mentioned elsewhere, I spent a lot of time on the project, often without documenting what I did. This was less of a problem towards the end as I tried to change it, but I still think there is room for improvement.
Self-learning	While at the start of my project, I had no idea where to start to learn about AI, my independence has forced me to develop skills by myself. I think that the comparison in skill level with in November (especially in programming and AI skills) is evidence that my self-learning has developed a lot.
Problem-solving	As I have documented in my Problems / Solutions section, I encountered numerous problems in my project. I've learnt to take a step back and work on something else for a while, and this can help me understand what is wrong and/or find another approach in the meantime.
Evaluation	Assessing has come up in so many aspects of my project, and I've tried to write my thoughts where I can in these pages. As a result, evaluating has become easier and I've used it in every step of the way. Getting feedback for my project has also helped me with my evaluation of the project as a whole.
Discipline	Like with time-management, this skill is far from perfect. I do feel as though I could have disciplined myself better overall if I had done this again. My discipline has improved though, since I had no choice but to catch up on the tasks I didn't want to do; I just think that I should have done the tasks immediately.
Computational proficiency	Especially after the broken computer incident, working with computers and the problems that entails have become natural to me. I've had to use the command line very frequently, navigate the file system, overcome issues with the operating system, work on unfamiliar software, to name a few. This has increased my confidence with digital skills.
Research	Since a large, and vital, part of my research has been research, this skill has benefitted during the year. Since March, I've had to gather research from sources for my final outcome, which has given me another opportunity to further this skill.
Organisation	I'm very satisfied with my organisation of the project as a whole; the flexible approach I chose definitely worked to my benefit. I've also made sure to keep all files and evidence tidy.
Planning	My planning has improved throughout the course of the project: this can be seen in my more detailed timelines as the project progresses. My outcome also went as I planned it to, so I think that this is strong evidence that my planning skills have improved.
AI skills	Constructing a machine learning algorithm, gathering data, training, testing and implementing the model has helped me build my AI skills from having no knowledge at all. Nonetheless, I'm still hesitant to put a higher skill level than 8 since AI is such a vast and deep field; I feel like I'm only scratching the surface.

Skills development evidence

Sunday, July 07, 2024 12:34 PM

Learn Python the Hard Way [8]

06 January 2024 11:00

Learn Python the Hard Way is a course to learn Python for beginners in book format. Each exercise is a piece of code that I have to copy into a Python file. I then run the code in the Python file on the shell. It will show the output which are the result of the commands of the code in the file. In my case I have screenshots both the window showing the Python file (on a text editor, in a white background) and the shell (on the terminal, in a dark purple background).

The book is around 55 exercises, but I stopped at 44 because by this point I had been taught all the basic skills I needed. After this is more creative exercises that use your skills, and I thought that a better use of my time would be to apply my skills in machine learning tutorials.

I have summarised each exercise below.

Exercise 0: The Setup

Teaches you how to set up your Python environment. You learn how to install Python, a text editor, and use the terminal.

Exercise 1: A Good First Program

Introduces you to writing your first Python script. You learn to print simple text and get familiar with running Python scripts from the terminal.

Exercise 2: Comments and Pound Characters

Explains how to use comments in Python code. You learn about the # symbol to add non-executable text in your scripts.

Exercise 3: Numbers and Math

Covers basic arithmetic operations in Python. You practice using operators like +, -, *, /, and %.

Exercise 4: Variables and Names

Introduces variables and how to store data in them. You learn to assign values to variables and use them in your calculations.

Exercise 5: More Variables and Printing

Expands on variables by introducing formatted strings. You learn how to embed variables within strings using f-strings or the .format() method.

Exercise 6: Strings and Text

Teaches you more about strings, including concatenation and string formatting. You practice embedding multiple variables in strings.

Exercise 7: More Printing

Focuses on more complex string manipulations. You learn to use escape sequences and how to control output format using commas.

Exercise 8: Printing, Printing

Further practices string formatting with multiple variables. You deepen your understanding of string interpolation and f-strings.

Exercise 9: Printing, Printing, Printing

Explores multiline strings and using print to output them. You learn how to create blocks of text with triple quotes ("""" or """).

Exercise 10: What Was That?

Introduces escape sequences in strings. You learn how to include special characters in strings, such as newlines and tabs.

Exercise 11: Asking Questions

Teaches how to get user input using input() function. You practice combining user input with formatted strings.

Exercise 12: Prompting People

Enhances the use of input() with custom prompts. You learn to prompt users with more specific instructions when collecting input.

Exercise 13: Parameters, Unpacking, Variables

Introduces command-line arguments using argv. You learn how to pass and unpack multiple arguments to a script from the terminal.

Exercise 14: Prompting and Passing

Combines user input with command-line arguments. You practice writing scripts that take both user input and command-line arguments.

Exercise 15: Reading Files

Teaches how to read files in Python using the open() function. You learn to open, read, and print the contents of a file.

Exercise 16: Reading and Writing Files

Introduces writing to files. You learn how to create, write, and truncate files using open() with different modes ('w', 'r+').

Exercise 17: More Files

Covers file operations, including copying content from one file to another. You practice checking file existence and copying data using Python.

Exercise 18: Names, Variables, Code, Functions

Introduces functions and how to define them. You learn about function parameters, return values, and how to call functions.

Exercise 19: Functions and Variables

Explores the relationship between functions and variables. You learn how functions can use and modify variables.

Exercise 20: Functions and Files

Combines functions with file handling. You practice writing functions that read from and manipulate files.

Exercise 21: Functions Can Return Something

Focuses on returning values from functions. You learn how to use return statements and capture returned values in variables.

Exercise 22: What Do You Know So Far?

A review exercise to consolidate your learning. You revisit all previous exercises to ensure understanding.

Exercise 23: Read Some Code

Encourages you to read other people's code. You learn how to analyze and understand code written by others.

Exercise 24: More Practice

Combines several concepts learned so far, including strings, functions, and variables. You practice building more complex scripts.

Exercise 25: Even More Practice

Provides additional practice with functions and lists. You learn how to manipulate lists and return multiple values from functions.

Exercise 26: Congratulations, Take a Test!

A debugging exercise where you fix broken code. You practice identifying and correcting syntax and logical errors in a script.

Exercise 27: Memorizing Logic

Introduces Boolean logic and its operations. You learn about and, or, not, and how to apply them in decision-making.

Exercise 28: Boolean Practice

Provides practice with Boolean expressions. You solve various Boolean logic problems to solidify your understanding.

Exercise 29: What If

Introduces conditional statements with if. You learn how to create simple decision-making structures in your code.

Exercise 30: Else and If

Expands on if statements by introducing else and elif. You learn to create more complex decision-making structures.

Exercise 31: Making Decisions

Combines if, else, and elif in a practical scenario. You practice writing scripts that make decisions based on user input.

Exercise 32: Loops and Lists

Introduces for loops and lists. You learn to iterate over list elements and perform actions on them.

Exercise 33: While Loops

Teaches how to use while loops. You practice creating loops that run based on a condition.

Exercise 34: Accessing Elements of Lists

Focuses on list indexing. You learn how to access and manipulate elements in a list by their index.

Exercise 35: Branches and Functions

Combines functions, conditional statements, and loops in a more complex script. You practice writing functions that include branching logic.

Exercise 36: Designing and Debugging

Encourages you to design and debug your own programs. You learn to apply a structured approach to problem-solving in Python.

Exercise 37: Symbol Review

Reviews all Python symbols, operators, and keywords. You solidify your understanding of the syntax and grammar of Python.

Exercise 38: Doing Things to Lists

Teaches various list methods and operations. You learn how to modify, split, join, and extend lists.

Exercise 39: Dictionaries, Oh Lovely Dictionaries

Introduces dictionaries in Python. You learn how to create, access, and manipulate key-value pairs.

Exercise 40: Modules, Classes, and Objects

Introduces object-oriented programming by explaining modules, classes, and objects. You learn how to create and use classes and objects in Python.

Exercise 41: Learning to Speak Object-Oriented

Explores object-oriented terminology. You learn key OOP concepts like inheritance, composition,

and method overriding.

Exercise 42: Is-A, Has-A, Objects, and Classes

Focuses on relationships between objects using is-a and has-a. You practice defining classes with inheritance and composition.

Exercise 43: Basic Object-Oriented Analysis and Design

Guides you through designing a simple game using object-oriented principles. You learn how to structure and organize code using classes.

Exercise 44: Inheritance vs. Composition

Compares inheritance and composition. You learn when to use each technique for organizing code in an object-oriented way.

ex1

The screenshot shows a code editor with two tabs: 'ex1.py' and 'ex2.py'. The 'ex1.py' tab contains the following Python code:

```
1 print("Hello World!")
2 print("Hello Again")
3 print("I like typing this.")
4 print("This is fun.")
5 print('Yay! Printing.')
6 print("I'd much rather you 'not'.")
7 print('I "said" do not touch this.')
```

The terminal window below shows the execution of the script:

```
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex1.py
Hello World!
Hello Again
I like typing this.
This is fun.
Yay! Printing.
I'd much rather you 'not'.
I "said" do not touch this.
uxue@uxue-dell:~/School/EPQ/lpthw$ open ex1.py
uxue@uxue-dell:~/School/EPQ/lpthw$ S
```

ex2

The screenshot shows a code editor with three tabs: 'ex1.py', 'ex2.py', and 'ex1.py'. The 'ex2.py' tab contains the following Python code:

```
1 # A comment, this is so you can read your program later.
2 # Anything after the # is ignored by python.
3
4 print("I could have code like this.") # and the comment after is ignored
5
6 # You can also use a comment to "disable" or comment out code:
7 # print("This won't run.")
8
9 print("This will run.")
```

The terminal window below shows the execution of the script:

```
uxue@uxue-dell:~/School/EPQ/lpthw$ open ex2.py
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex2.py
I could have code like this.
This will run.
uxue@uxue-dell:~/School/EPQ/lpthw$ uxue@uxue-dell:~/School/EPQ/lpthw$ uxue@uxue-dell:~/School/EPQ/lpthw$ uxue@uxue-dell:~/School/EPQ/lpthw$ uxue@uxue-dell:~/School/EPQ/lpthw$ uxue@uxue-dell:~/School/EPQ/lpthw$
```

ex3

Open ex1.py ex2.py ex3.py

```

1 print("I will now count my chickens:") # prints the text in the brackets
2
3 print("Hens", 25 + 30 / 6) # prints "hens" then works out the question
4 print("Rosters", 100 - 25 * 3 % 4) # see above
5
6 print("Now I will count the eggs:") # see line 1
7
8 print(3 + 2 + 1 - 5 + 4 % 2 - 1 / 4 + 6) # prints the answer
9
10 print("Is it true that 3 + 2 < 5 - 7?")
11
12 print(3 + 2 < 5 - 7) # states whether this statement is "true" if the answer to the left is
   smaller than the answer to the right, or "false" if not
13
14 print("What is 3 + 2?", 3 + 2)
15 print("What is 5 - 7?", 5 - 7)
16
17 print("Oh, that's why it's False.")
18
19 print("How about some more.")
20
21 print("Is it greater?", 5 > -2)
22 print("Is it greater or equal?", 5 >= -2)
23 print("Is it less or equal?", 5 <= -2)

```

uxue@uxue-dell:~/School/EPQ/lpthw\$ python3 ex3.py

```

I will now count my chickens:
Hens 30.0
Rosters 97
Now I will count the eggs:
6.75
Is it true that 3 + 2 < 5 - 7?
False
What is 3 + 2? 5
What is 5 - 7? -2
Oh, that's why it's False.
How about some more.
Is it greater? True
Is it greater or equal? True
Is it less or equal? False
uxue@uxue-dell:~/School/EPQ/lpthw$
```

ex4

Open ex1.py ex2.py ex3.py ex4.py

```

1 cars = 100
2 space_in_a_car = 4
3 drivers = 30
4 passengers = 90
5 cars_not_driven = cars - drivers
6 cars_driven = drivers
7 carpool_capacity = cars_driven * space_in_a_car
8 average_passengers_per_car = passengers / cars_driven
9
10
11 print("There are", cars, "cars available.")
12 print("There are only", drivers, "drivers available.")
13 print("There will be", cars_not_driven, "empty cars today.")
14 print("We can transport", carpool_capacity, "people today.")
15 print("We have", passengers, "to carpool today.")
16 print("We need to put about", average_passengers_per_car, "in each car.")

```

uxue@uxue-dell:~/School/EPQ/lpthw\$ open ex4.py

```

uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex4.py
There are 100 cars available.
There are only 30 drivers available.
There will be 70 empty cars today.
We can transport 120 people today.
We have 90 to carpool today.
We need to put about 3.0 in each car.
uxue@uxue-dell:~/School/EPQ/lpthw$
```

ex5

```

1 name = 'Zed A. Shaw'
2 age = 35
3 height = 74
4 weight = 180
5 eyes = 'Blue'
6 teeth = 'White'
7 hair = 'Brown'
8 height_cm = height * 2.54
9 weight_kg = weight * 0.45359237
10
11 print(f"Let's talk about {name}.")
12 print(f"He's {height} inches tall; that's {height_cm} centimetres.")
13 print(f"He's {weight} pounds heavy; that's {weight_kg} kilograms.")
14 print("Actually that's not too heavy.")
15 print(f"He's got {eyes} eyes and {hair} hair.")
16 print(f"His teeth are usually {teeth} depending on the coffee.")
17
18 total = age + height + weight
19 print(f"If I add {age}, {height}, and {weight} I get {total}.")

```

```

uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex5.py
Let's talk about Zed A. Shaw.
He's 74 inches tall; that's 187.96 centimetres.
He's 180 pounds heavy; that's 81.6466266 kilograms.
Actually that's not too heavy.
He's got Blue eyes and Brown hair.
His teeth are usually White depending on the coffee.
If I add 35, 74, and 180 I get 289.
uxue@uxue-dell:~/School/EPQ/lpthw$ 

```

ex7

```

1 print("Mary had a little lamb.")
2 print("Its fleece was white as {}".format('snow'))
3 print("And everywhere that Mary went.")
4 print("." * 10) # what'd that do?
5
6 end1 = "C"
7 end2 = "h"
8 end3 = "e"
9 end4 = "e"
10 end5 = "s"
11 end6 = "e"
12 end7 = "B"
13 end8 = "u"
14 end9 = "r"
15 end10 = "g"
16 end11 = "e"
17 end12 = "r"
18
19 # watch that comma at the end. try removing it to see what happens
20 print(end1 + end2 + end3 + end4 + end5 + end6, end=' ')
21 print(end7 + end8 + end9 + end10 + end11 + end12)

```

```

uxue@uxue-dell:~/School/EPQ/lpthw$ command 'python3' from deb python3
command 'python' from deb python-is-python3
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex7.py
Mary had a little lamb.
Its fleece was white as snow.
And everywhere that Mary went.
.....
Cheese Burger
uxue@uxue-dell:~/School/EPQ/lpthw$ 

```

ex8

```
Open ▾  ex1.py  ex2.py  ex3.py  ex4.py  ex5.py  ex6.py  ex7.py  ex8.py  ex8.py
Save  ⌂  -  x

1 formatter = "{} {} {} {}"
2
3 print(formatter.format(1, 2, 3, 4))
4 print(formatter.format("one", "two", "three", "four"))
5 print(formatter.format(True, False, False, True))
6 print(formatter.format(formatter, formatter, formatter, formatter))
7 print(formatter.format(
8     "Try your",
9     "Own text here",
10    "Maybe a poem",
11    "Or a song about fear"
12 ))
```

```
xue@uxue-dell: ~/School/EPQ/lpthw
xue/School/EPQ/lpthw/ex8.py": No such file or directory
xue@uxue-dell:~/School/EPQ/lpthw$ open ex8.py
xue@uxue-dell:~/School/EPQ/lpthw$ python3 ex8.py
1 2 3 4
one two three four
True False False True
() () () () () () () () () () () ()
Try your Own text here Maybe a poem Or a song about fear
xue@uxue-dell:~/School/EPQ/lpthw$
```

ex9

The screenshot shows a terminal window with the following text:

```
uxue@uxue-dell:~/School/EPQ/lpthw$ open ex9.py
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex9.py
Here are the days: Mon Tue Wed Thu Fri Sat Sun
Here are the months: Jan
Feb
Mar
Apr
May
Jun
Jul
Aug

There's something going on here.
With the three double-quotes.
We'll be able to type as much as we like.
Even 4 lines if we want, or 5, or 6.

uxue@uxue-dell:~/School/EPQ/lpthw$
```

ex10

ex10

```

1 tabby_cat = "\tI'm tabbed in."
2 persian_cat = "I'm split\non a line."
3 backslash_cat = "I'm \\ a \\ cat."
4
5 fat_cat = """
6 I'll do a list:
7 \t* Cat food
8 \t* Fishies
9 \t* Catnip\n\t* Grass
10 ''' # so I think """ and ''' do exactly the same thing?
11
12 print(tabby_cat)
13 print(persian_cat)
14 print(backslash_cat)
15 print(fat_cat)

```

uxue@uxue-dell:~/School/EPQ/lpthw\$ open ex10.py
uxue@uxue-dell:~/School/EPQ/lpthw\$ python3 ex10.py

```

With the three double-quotes.
We'll be able to type as much as we like.
Even 4 lines if we want, or 5, or 6.

I'm tabbed in.
I'm split
on a line.
I'm \ a \ cat.

I'll do a list:
    * Cat food
    * Fishies
    * Catnip
    * Grass

```

uxue@uxue-dell:~/School/EPQ/lpthw\$

ex11

```

1 print("How old are you?", end=' ')
2 age = input()
3 print("How tall are you?", end=' ')
4 height = input()
5 print("How much do you weigh?", end=' ')
6 weight = input()
7
8 print(f"So, you're {age} years old, {height} tall and {weight} heavy.")

```

I'm tabbed in.
I'm split
on a line.
I'm \ a \ cat.

I'll do a list:
 * Cat food
 * Fishies
 * Catnip
 * Grass

uxue@uxue-dell:~/School/EPQ/lpthw\$ open ex11.py
uxue@uxue-dell:~/School/EPQ/lpthw\$ python3 ex11.py

```

How old are you? 16
How tall are you? 160cm
How much do you weigh? 55kg
So, you're 16 years old, 160cm tall and 55kg heavy.

```

uxue@uxue-dell:~/School/EPQ/lpthw\$

ex12

Open ex12.py Save

ex12.py
~/School/EPQ/lpthw

1 age = input("How old are you? ")
2 # "age" means what is typed in (input) of "how old are you?"
3 height = input("How tall are you? ")
4 weight = input("How much do you weigh? ")
5
6 print(f"So, you're {age} years old, {height} tall and {weight} heavy.")

uxue@uxue-dell: ~/School/EPQ/lpthw

```
* Cat food
* Fishies
* Catnip
* Grass

uxue@uxue-dell:~/School/EPQ/lpthw$ open ex11.py
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex11.py
How old are you? 16
How tall are you? 160cm
How much do you weigh? 55kg
So, you're 16 years old, 160cm tall and 55kg heavy.
uxue@uxue-dell:~/School/EPQ/lpthw$ open ex12.py
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex12.py
How old are you? 16
How tall are you? 610cm
How much do you weigh? 55kg
So, you're 16 years old, 610cm tall and 55kg heavy.
uxue@uxue-dell:~/School/EPQ/lpthw$
```

ex13

Open ex13.py Save

ex13.py
~/School/EPQ/lpthw

1 from sys import argv
2 # read the WYSS section for how to run this
3 script, first, second, third = argv
4
5 print("The script is called:", script)
6 print("Your first variable is:", first)
7 print("Your second variable is:", second)
8 print("Your third variable is:", third)

uxue@uxue-dell: ~/School/EPQ/lpthw

```
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex13.py argv1 argv2 argv3
The script is called: ex13.py
Your first variable is: argv1
Your second variable is: argv2
Your third variable is: argv3
uxue@uxue-dell:~/School/EPQ/lpthw$
```

ex14

The screenshot shows a terminal window titled "ex14.py" running on a Linux system. The command "python3 ex14.py Uxue" is entered, and the script runs as follows:

```
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex14.py Uxue
Hi Uxue, I'm the ex14.py script.
I'd like to ask you a few questions.
Do you like me, Uxue?
Type your answer here: Yes
Where do you live, Uxue?
Type your answer here: England
What kind of computer do you have?
Type your answer here: Dell

Alright, so you said 'Yes' about liking me.
You live in England. Not sure where that is.
And you have a Dell computer. Nice.

uxue@uxue-dell:~/School/EPQ/lpthw$
```

Ex15

The screenshot shows a terminal window with the following content:

```
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex15.py ex15_sample.txt
Here's your file ex15_sample.txt:
This is stuff I typed into a file.
It is really cool stuff.
Lots and lots of fun to have in here.

Type the filename again:
> ex15_sample.txt
This is stuff I typed into a file.
It is really cool stuff.
Lots and lots of fun to have in here.

uxue@uxue-dell:~/School/EPQ/lpthw$
```

ex16

ex16

```

1 from sys import argv
2
3 script, filename = argv
4
5 print(f"We're going to erase {filename}.")
6 print("If you don't want that, hit CTRL-C (^C).")
7 print("If you do want that, hit RETURN.")
8
9 input("?")
10
11 print("Opening the file...")
12 target = open(filename, 'w')
13
14 print("Truncating the file. Goodbye!")
15 target.truncate() # truncate = delete all text in the file (?)
16
17 print("Now I'm going to ask you for three lines.")
18
19 line1 = input("line 1: ")
20 line2 = input("line 2: ")
21 line3 = input("line 3: ")
22
23 print("I'm going to write these to the file.")
24
25 target.write(line1)
26 target.write("\n")
27 target.write(line2)
28 target.write("\n")
29 target.write(line3)
30 target.write("\n")
31
32 print("And finally, we close it.")
33 target.close()

```

uxue@uxue-dell: ~/School/EPQ/lpthw\$ python3 ex16.py test.txt
We're going to erase test.txt.
If you don't want that, hit CTRL-C (^C).
If you do want that, hit RETURN.
?
Opening the file...
Truncating the file. Goodbye!
Now I'm going to ask you for three lines.
line 1: This is the first line.
line 2: This is the second line.
line 3: This is the third line
I'm going to write these to the file.
And finally, we close it.

ex17

```

1 from sys import argv
2 from os.path import exists
3
4 script, from_file, to_file = argv
5
6 print(f"Copying from {from_file} to {to_file}")
7
8 # we could do these two on one line, how?
9 in_file = open(from_file)
10 indata = in_file.read() # so indata opens and reads the first file
11
12 out_file = open(to_file, 'w') # opens second file, to write in
13 out_file.write(indata) # writes first file's contents into second file
14
15 print("Alright, all done.")
16
17 out_file.close()
18 in_file.close() # closes both files

```

uxue@uxue-dell:~/School/EPQ/lpthw\$ python3 ex17.py test.txt new_file.txt
Copying from test.txt to new_file.txt
Alright, all done.

ex18

ex18.py

```

1 # this one is like your scripts with argv
2 def print_two(*args):
3     arg1, arg2 = args
4     print(f"arg1: {arg1}, arg2: {arg2}")
5
6 # that *args is actually pointless, we can just do this
7 def print_two_again(arg1, arg2):
8     print(f"arg1: {arg1}, arg2: {arg2}")
9
10 # this just takes one argument
11 def print_one(arg1):
12     print(f"arg1: {arg1}")
13
14 # this one takes no arguments
15 def print_none():
16     print("I got nothin'.")
17
18
19 print_two("Zed", "Shaw")
20 print_two_again("Zed", "Shaw")
21 print_one("First!")
22 print_none()

```

uxue@uxue-dell:~/School/EPQ/lpthw\$ open ex18.py
uxue@uxue-dell:~/School/EPQ/lpthw\$ python3 ex18.py
arg1: Zed, arg2: Shaw
arg1: Zed, arg2: Shaw
arg1: First!
I got nothin'.

ex19

ex19.py

```

1 def cheese_and_crackers(cheese_count, boxes_of_crackers):
2     print(f"You have {cheese_count} cheeses!")
3     print(f"You have {boxes_of_crackers} boxes of crackers!")
4     print("Man that's enough for a party!")
5     print("Get a blanket.\n")
6
7
8 print("We can just give the function numbers directly:")
9 cheese_and_crackers(20, 30)
10
11
12 print("OR, we can use variables from our script:")
13 amount_of_cheese = 10
14 amount_of_crackers = 50
15
16 cheese_and_crackers(amount_of_cheese, amount_of_crackers)
17
18
19 print("We can even do maths inside too:")
20 cheese_and_crackers(10 + 20, 5 + 6)
21
22
23 print("And we can combine the two, variables and maths:")
24 cheese_and_crackers(amount_of_cheese + 100, amount_of_crackers + 1000)

```

uxue@uxue-dell:~/School/EPQ/lpthw\$ python3 ex19.py
We can just give the function numbers directly:
You have 20 cheeses!
You have 30 boxes of crackers!
Man that's enough for a party!
Get a blanket.

OR, we can use variables from our script:
You have 10 cheeses!
You have 50 boxes of crackers!
Man that's enough for a party!
Get a blanket.

We can even do maths inside too:
You have 30 cheeses!
You have 11 boxes of crackers!
Man that's enough for a party!
Get a blanket.

And we can combine the two, variables and maths:
You have 110 cheeses!
You have 1050 boxes of crackers!
Man that's enough for a party!
Get a blanket.

ex20

Open ▾ Save

ex20.py

~/School/EPQ/lpthw

ex13.py ex14.py ex15.py *ex16.py ex17.py ex18.py ex19.py ex20.py

```

1 from sys import argv
2
3 script, input_file = argv
4
5 def print_all(f): # what is f? I think it specifies what file
6     print(f.read()) # print what is read in the file
7
8 def rewind(f):
9     f.seek(0) # seek moves the position of the File Handle (like a cursor) - the number is bytes
10
11 def print_a_line(line_count, f):
12     print(line_count, f.readline()) # print the line specified in the bracket?
13
14 current_file = open(input_file)
15
16 print("First let's print the whole file:\n")
17
18 print_all(current_file)
19
20 print("Now let's rewind, kind of like a tape.")
21
22 rewind(current_file)
23
24 print("Let's print three lines:\n")
25
26 current_line = 1
27 print_a_line(current_line, current_file)
28
29 current_line += 1 # go down one line so current line is line 2
30 print_a_line(current_line, current_file)
31
32 current_line += 1
33 print_a_line(current_line, current_file)

```

uxue@uxue-dell:~/School/EPQ/lpthw\$ python3 ex20.py test.txt

First let's print the whole file:

This is the first line.
This is the second line.
This is the third line

Now let's rewind, kind of like a tape.
Let's print three lines:

1 This is the first line.
2 This is the second line.
3 This is the third line

ex21

Open ▾ Save

ex21.py

~/School/EPQ/lpthw

ex14.py ex15.py *ex16.py ex17.py ex18.py ex19.py ex20.py ex21.py

```

1 def add(a, b):
2     print(f"ADDING {a} + {b}")
3     return a + b
4
5 def subtract(a, b):
6     print(f"SUBTRACTING {a} - {b}")
7     return a - b
8
9 def multiply(a, b):
10    print(f"MULTIPLYING {a} * {b}")
11    return a * b
12
13 def divide(a, b):
14    print(f"DIVIDING {a} / {b}")
15    return a / b
16
17
18 print("Let's do some maths with just functions!")
19
20 age = add(30, 5)
21 height = subtract(78, 4)
22 weight = multiply(90, 2)
23 iq = divide(100, 2)
24
25 print(f"Age: {age}, Height: {height}, Weight: {weight}, IQ: {iq}")
26
27
28 # A puzzle for the extra credit, type it in anyway.
29 print("Here is a puzzle.")
30
31 what = add(age, subtract(height, multiply(weight, divide(iq, 2))))
32
33 print(f"That becomes: {what}. Can you do it by hand?")
34
35 def return_test(a, b, c):

```

uxue@uxue-dell:~/School/EPQ/lpthw\$ python3 ex21.py

Let's do some maths with just functions!

ADDING 30 + 5
SUBTRACTING 78 - 4
MULTIPLYING 90 * 2
DIVIDING 100 / 2
Age: 35, Height: 74, Weight: 180, IQ: 50.0
Here is a puzzle.

DIVIDING 50.0 / 2
MULTIPLYING 180 * 25.0
SUBTRACTING 74 - 4500.0
ADDING 35 + -4426.0
That becomes: -4391.0 Can you do it by hand?

10 + 3 - 6
Answer: 7

ex22

```
1 print() # repeats what is inside the brackets
2 # octothorpe/hash: to write comments. Text after the hash is ignored
3 +, -, /, *, >, <, >=, <=
4 = # makes the left of = sign (variable) equal the right
5 f("") # allows you to write variables in the quotation marks, with {}
6 .format # another way of using f"" with a variable before the .
7 # string =
8 "\n" # new line
9 """
10 """ # type anything (including the enter key) between
11 "\t" # indent
12 "\\" # one backslash
13 """
14 """ # like three quotation marks
15 input() # what you type into the program then press enter
16 from "?" import "?" # some new commands!
17 argv # argument variable: what is typed after "python script.py" - need to type "from sys import argv" to use it
18 open() # opens the file inside the brackets
19 .read() # reads the contents of the file (preceded by open(filename))
20 open(filename, 'w') # opens file to write in
21 .truncate() # deletes all text in the file
22 .write() # writes what is in the brackets in the file
23 .close() # closes the file
24 len() # length in bytes of the file
25 exists() # whether the file exists (true or false) - need to type from "os.path import exists"
first
26 def function(): # function: the commands below (with an indent) say what the function does
27 f # in def function(f), f.seeks, etc. means file
28 f.seek() # seek moves the position of the File Handle (like a cursor) - the number in the
brackets is bytes. So f.seek(0) means to the start of the file
29 return # works out the answer e.g. in return a + b
```

ex23

```
1 import sys
2 script, encoding, error = sys.argv
3
4
5 def main(language_file, encoding, errors):
6     line = language_file.readline() # reads one line from the given file
7
8     if line: # tests whether the line has something in it
9         print_line(line, encoding, errors) # see 13-18
10        return main (language_file, encoding, errors) # jumps back to the top
11
12
13 def print_line(line, encoding, errors): # this function does the encoding
14    next_lang = line.strip() # .strip() removes white spaces at the beginning and end
15    raw_bytes = next_lang.encode(encoding, errors=errors) # .encode() makes a string into bytes;
inside brackets tells it encoding I want and how to handle errors
16    cooked_string = raw_bytes.decode(encoding, errors=errors) # .decode() makes raw bytes into a
string
17
18    print(raw_bytes, "<==>", cooked_string)
19 # Decode Bytes, Encode Strings (DBES)
20
21 languages = open("languages.txt", encoding="utf-8") # opens languages.txt file
22
23 main(languages, encoding, error)
```

Python 2 ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

ex24

ex25

```

*ex25.py
~/School/EPQ/lpthw

1 def break_words(stuff):
2     """This function will break up words for us.""" # documentation commentsd
3     words = stuff.split(' ') # what does .split do?
4     return words
5
6 def sort_words(words):
7     """Sorts the words."""
8     return sorted(words) # what does sorted do?
9
10 def print_first_word(words):
11     """Prints the first word after popping it off."""
12     word = words.pop(0) # what does .pop do?
13     print(word)
14
15 def print_last_word(words):
16     """Prints the last word after popping it off."""
17     word = words.pop(-1)
18     print(word)
19
20 def sort_sentence(sentence):
21     """Takes in a full sentence and returns the sorted words."""
22     words = break_words(sentence)
23     return sort_words(words)
24
25 def print_first_and_last(sentence):
26     """Prints the first and last words of the sentence."""
27     words = break_words(sentence)
28     print_first_word(words)
29     print_last_word(words)
30
31 def print_first_and_last_sorted(sentence):
32     """Sorts the words then prints the first and last one."""
33     words = sort_sentence(sentence)
34     print_first_word(words)
35     print_last_word(words)

```

Python 2 ▾ Tab Width: 8 ▾ In 1 Col 1 ▾ INC

ex26

```

ex26.py
~/School/EPQ/lpthw

1 from sys import argv
2
3 print("How old are you?", end=' ')
4 age = input()
5 print("How tall are you?", end=' ')
6 height = input()
7 print("How much do you weigh?", end=' ')
8 weight = input()
9
10 print(f"So, you're {age} old, {height} tall and {weight} heavy.")
11
12 script, filename = argv
13
14 txt = open(filename)
15
16 print(f"Here's your file {filename}:")
17 print(txt.read())
18
19 print("Type the filename again:")
20 file_again = input('>')
21
22 txt_again = open(file_again)
23
24 print(txt_again.read())
25
26
27 print("Let's practice everything.")
28 print('You\'d need to know \'bout escapes \n\twith \\ that do \n newlines and \t tabs.')
29
30 poem = """
31 \The lovely world
32 with logic so firmly planted
33 cannot discern \n the needs of love
34 nor comprehend passion from intuition
35 and requires an explanation
36 \n\twhere there is none.
37 """
38
39 print(poem)
40 print(open())
41 print(poem)

```

uxue@uxue-dell: ~/School/EPQ/lpthw\$ python3 ex26.py test.txt

```

bash: syntax error near unexpected token `import'.
uxue@uxue-dell: ~/School/EPQ/lpthw$ python3 ex26.py test.txt
How old are you? 17
How tall are you? 175cm
How much do you weigh? 71kg
So, you're 17 old, 175cm tall and 71kg heavy.
Here's your file [filename]:
This is the first line.
This is the second line.
This is the third line

Type the filename again:
> test.txt
This is the first line.
This is the second line.
This is the third line

Let's practice everything.
You'd need to know 'bout escapes
    with \ that do
newlines and    tabs.

The lovely world
with logic so firmly planted
cannot discern
the needs of love
nor comprehend passion from intuition
and requires an explanation

where there is none.

This would be five: 5
With a starting point of: 10000
We'd have 500000 beans, 500.0 jars, and 50.0 crates.
We can also do that this way:
We'd have 500000.0 beans, 500.0 jars, and 50.0 crates.
Too many cats! The world is doomed!
The world is full
People are greater than or equal to dogs.
People are less than or equal to dogs.
People are dogs.


```

uxue@uxue-dell: ~/School/EPQ/lpthw\$

ex28

```
1 1. True and True TRUE
2 2. False and True FALSE
3 3. 1 == 1 and 2 == 1 FALSE
4 4. "test" == "test" TRUE
5 5. 1 == 1 or 2 != 1 TRUE
6 6. True and 1 == 1 TRUE
7 7. False and 0 != 0 FALSE
8 8. True or 1 == 1 TRUE
9 9. "test" == "testing" FALSE
10 10. 1 != 0 and 2 == 1 FALSE
11 11. "test" != "testing" TRUE
12 12. "test" == 1 FALSE
13 13. not (True and False) TRUE
14 14. not (1 == 1 and 0 != 1) FALSE
15 15. not (10 == 1 or 1000 == 1000) FALSE
16 16. not (1 != 10 or 3 == 4) FALSE
17 17. not ("testing" == "testing" and "Zed" == "Cool Guy") TRUE
18 18. 1 == 1 and (not ("testing" == 1 or 1 == 0)) TRUE
19 19. "chunky" == "bacon" and (not (3 == 4 or 3 == 3)) FALSE
20 20. 3 == 3 and (not ("testing" == "testing" or "Python" == "Fun")) FALSE
```

ex29

```
1 people = 20
2 cats = 30
3 dogs = 15
4
5
6 if people < cats:
7     print("Too many cats! The world is doomed!")
8
9 if people > cats:
10    print("Not many cats! The world is saved!")
11
12 if people < dogs:
13    print("The world is drooled on!")
14
15 if people > dogs:
16    print("The world is dry!")
17
18
19 dogs += 5
20
21 if people >= dogs:
22    print("People are greater than or equal to dogs.")
23
24 if people <= dogs:
25    print("People are less than or equal to dogs.")
26
27
28 if people == dogs:
29    print("People are dogs.")
```

```
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex29.py
Too many cats! The world is doomed!
The world is dry!
People are greater than or equal to dogs.
People are less than or equal to dogs.
People are dogs.
uxue@uxue-dell:~/School/EPQ/lpthw$
```

ex30

Open New

ex30.py
~/School/EPQ/lpthw

Save ☰ - □ ×

ex24.py *ex25.py ex26.py ex28.txt ex29.py ex30.py

```

1 people = 50
2 cars = 30
3 trucks = 60
4
5
6 if cars > people:
7     print("We should take the cars.")
8 elif cars < people:
9     print("We should not take the cars.")
10 else:
11     print("We can't decide.")
12
13 if trucks > cars:
14     print("That's too many trucks.")
15 elif trucks < cars:
16     print("Maybe we could take the trucks.")
17 else:
18     print("We still can't decide.")
19
20 if people > trucks:
21     print("Alright, let's just take the trucks.")
22 else:
23     print("Fine, let's stay home then.")

```

```

uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex30.py
We should not take the cars.
That's too many trucks.
Fine, let's stay home then.
uxue@uxue-dell:~/School/EPQ/lpthw$ 

```

ex31

Open New

***ex31.py**
~/School/EPQ/lpthw

Save ☰ - □ ×

ex24.py *ex25.py ex26.py ex28.txt ex29.py ex30.py *ex31.py

```

1     print("""You enter a dark room with two doors.
2 Do you go through door #1 or door #2?""")
3
4 door = input("> ")
5
6 if door == "1":
7     print("There's a giant bear here eating a cheese cake.")
8     print("What do you do?")
9     print("1. Take the cake.")
10    print("2. Scream at the bear.")
11
12    bear = input("> ")
13
14    if bear == "1":
15        print("The bear eats your face off. Good job!")
16    elif bear == "2":
17        print("The bear eats your legs off. Good job!")
18    else:
19        print(f"Well, doing {bear} is probably better.")
20        print("Bear runs away.")
21
22 elif door == "2":
23     print("You stare into the endless abyss at Cthulhu's retina.")
24     print("1. Blueberries.")
25     print("2. Yellow jacket clothespins.")
26     print("3. Understanding revolvers yelling melodies.")
27
28     insanity = input("> ")
29
30     if insanity == "1" or insanity == "2":
31         print("Your body survives powered by a mind of jello.")
32         print("Good job!")
33     else:
34         print("The insanity rots your eyes into a pool of muck.")
35         print("Good job!")
36
37 else:
38     print("You stumble around and fall on a knife and die. Good job!")

```

```

uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex31.py
You enter a dark room with two doors.
Do you go through door #1 or door #2?
> 1
There's a giant bear here eating a cheese cake.
What do you do?
1. Take the cake.
2. Scream at the bear.
> 1
The bear eats your face off. Good job!
uxue@uxue-dell:~/School/EPQ/lpthw$ 

```

ex32

```

< ex25.py < ex26.py < ex28.txt < ex29.py < ex30.py >
1 the_count = [1, 2, 3, 4, 5]
2 fruits = ['apples', 'oranges', 'pears', 'apricots']
3 change = [1, 'pennies', 2, 'dimes', 3, 'quarters']
4
5 # this first kind of for-loop goes through a list
6 for number in the_count:
7     printf("This is count %d" % number)
8
9 # same as above
10 for fruit in fruits:
11     printf("A fruit of type: %s" % fruit)
12
13 # also we can go through mixed lists too
14 # notice we have to use () since we don't know what's in it
15 for i in change:
16     printf("I got %r" % i)
17
18 # we can also build lists, first start with an empty one
19 elements = []
20
21 # then use the range function to do 0 to 5 counts
22 for i in range(0, 6): # so 0 is included but 6 is not
23     printf("Adding %d to the list." % i)
24     # append is a function that lists understand
25     elements.append(i)
26
27 # now we can print them out too
28 for i in elements:
29     printf("Element was: %d" % i)

```

uxue@uxue-dell:~/School/EPQ/iphw\$ python3 ex32.py

```

This is count 1
This is count 2
This is count 3
This is count 4
This is count 5
A fruit of type: apples
A fruit of type: oranges
A fruit of type: pears
A fruit of type: apricots
I got
I got pennies
I got 2
I got dimes
I got 3
I got quarters
Adding 0 to the list.
Adding 1 to the list.
Adding 2 to the list.
Adding 3 to the list.
Adding 4 to the list.
Adding 5 to the list.
Element was: 0
Element was: 1
Element was: 2
Element was: 3
Element was: 4
Element was: 5

```

ex33

```

Open < ex26.py < ex28.txt < ex29.py < -/School/EPQ/iphw < ex30.py < ex31.py < ex32.py < ex33.py >
1 ex33.py
2 numbers = [] # new list that items will soon be added to
3
4 printf("Make your very own list!\\nSpecify the parameters of the list by answering these questions...")
5
6 printf("What will the highest number of the list be?")
7 i = int(input("..."))
8 printf("What is the interval between each number in the list?")
9 interval = int(input("> "))
10
11 def fill_list(i):
12     printf("Building list...")
13     while i < len(numbers):
14         numbers.append(i) # add i to the "numbers" list
15
16         i += interval
17     printf("Numbers now: ", numbers) # prints the "numbers" list
18     printf("I ls now (%d)" % i) # prints the new value of i
19     printf(".....")
20
21 full_list()
22 printf("Here are the numbers in order in your list: ")
23
24 for n in numbers:
25     printf(n) # prints each item of the "numbers" list
26
27 printf("All done!")

```

uxue@uxue-dell:~/School/EPQ/iphw\$ python3 ex33.py

```

Make your very own list!
Specify the parameters of the list by answering these questions.
What will the highest number of the list be?
> 5
What is the interval between each number in the list?
> 1
Building list...
Numbers now: []
Numbers now: [0]
Numbers now: [0, 1]
Numbers now: [0, 1, 2]
Numbers now: [0, 1, 2, 3]
Numbers now: [0, 1, 2, 3, 4]
Numbers now: [0, 1, 2, 3, 4, 5]
Numbers now: [0, 1, 2, 3, 4, 5]
Here are the numbers in order in your list:
0
1
2
3
4
5
All done!

```

ex35

```

Open < ex28.txt < ex29.py < ex30.py < *ex31.py < ex32.py < ex33.py < ex34.py < ex35.py >
1 from sys import exit
2
3 def gold_room():
4     printf("This room is full of gold. How much do you take?")
5
6     choice = input("> ")
7     if "1" or "2" or "3" or "4" or "5" or "6" or "7" or "8" or "9" or "0" in choice: #
8         how_to_write_if_integer_in_choice()
9         how_much = int(choice)
10    else:
11        dead("Man, learn to type a number.")
12
13    if how_much < 50:
14        printf("Nice, you're not greedy, you win!")
15        exit(0)
16    else:
17        dead("You greedy bastard!")
18
19 def bear_room():
20     printf("There is a bear here.")
21     printf("The bear has a bunch of honey.")
22     printf("The fat bear is in front of another door.")
23     printf("How are you going to move the bear?")
24     bear_moved = False
25
26     while True:
27         choice = input("> ")
28
29         if choice == "take honey":
30             dead("The bear looks at you then slaps your face off.")
31         elif choice == "taunt bear" and not bear_moved:
32             printf("The bear has moved from the door.")
33             printf("You can go through it now.")
34             bear_moved = True
35         elif choice == "taunt bear" and bear_moved:
36             dead("The bear gets pissed off and chews your leg off.")
37         elif choice == "open door" and bear_moved:
38             gold_room()
39         else:
40             printf("I got no idea what that means.")
41
42
43 def cthulhu_room():
44     printf("Here you see the great evil Cthulhu.")
45     printf("He, it, whatever stares at you and you go insane.")
46     printf("Do you flee for your life or eat your head?")
47
48     choice = input("> ")
49
50     if "flee" in choice:
51         start()

```

```

52     elif "head" in choice:
53         dead("Well that was tasty!")
54     else:
55         cthulhu_room()
56
57
58 def dead(why):
59     print(why, "You lost. Good job!")
60     exit(0)
61
62 def start():
63     print("You are in a dark room.")
64     print("There is a door to your right and left.")
65     print("Which one do you take?")
66
67     choice = input("> ")
68
69     if choice == "left":
70         bear_room()
71     elif choice == "right":
72         cthulhu_room()
73     else:
74         dead("You stumble around the room until you starve.")
75
76
77 start()
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex35.py
You are in a dark room.
There is a door to your right and left.
Which one do you take?
> left
There is a bear here.
The bear has a bunch of honey.
The fat bear is in front of another door.
How are you going to move the bear?
> take honey
The bear looks at you then slaps your face off. You lost. Good job!
uxue@uxue-dell:~/School/EPQ/lpthw$ 
```

ex36

```

1 from sys import argv
2 script, user_name = argv
3
4 hand = ['lamp', 'sword']
5
6 class always_possible:
7     def __init__(self, input):
8         self.input = input
9
10
11
12 class BaseInputs:
13
14     def __init__(self):
15         self.exit_strings = ["leave", "exit", "bye"]
16
17     def inputer(self, text):
18         user_input = input(">")
19         if self.exit_strings in user_input:
20             start()
21         return user_input
22
23 class DoorInput(BaseInputs):
24
25     def inputer(self, text):
26         user_input = super().inputer(self.exit_strings)
27         if "slam" in user_input:
28             print("slamming door")
29
30 door_input = DoorInput()
31 door_input.inputer("slam door")
32
33
34
35
36
37 def start():
38     print(f"Dear {user_name},\nYou need to find a way to leave this building.\nThere are
39     2 doors in front of you.\nDo you open the door on the left or the door on the right?")
40     choose_door()
41
42 def choose_door():
43     door_choice = input("> ")
44
45     if "left" in door_choice:
46         dark_room()
47     elif "right" in door_choice:
48         light_room()
49     else:
50         print("Type either left or right.")
51         choose_door() 
```

```

52
53
54 def dark_room():
55     if 'lamp' in hand:
56         dark_room2()
57     else:
58         dark_room1()
59
60 def dark_room1():
61     print(f"Dear {user_name},\nYou are now in the dark_room.\nAny number of hidden
62 dangers could be here.")
63     dark_action = input("> ")
64
65     if "leave" in dark_action:
66         start()
67     else:
68         die("The only way to survive is to leave. You have been attacked by the darkness
69 - death ensues.")
70
71
72 dark_room_list = ['monster', 'key']
73
74 def dark_room2():
75     print(f"Dear {user_name},\nIn this room you see:\n",dark_room_list)
76
77     if 'sword' in hand:
78         sword_dark_room()
79     else:
80         no_sword_dark_room()
81
82 def sword_dark_room():
83     sword_dark_action = input("> ")
84
85     if ("fight" in sword_dark_action) or ("kill" in sword_dark_action):
86         pprint("Well done, the monster is dead.")
87         monster_dead()
88     elif "leave" in sword_dark_action:
89         start()
90     else:
91         die("The only way to survive is to kill the monster. As you did not, the monster
92 kills you.")
93
94 def no_sword_dark_room():
95     no_sword_dark_action = input("> ")
96
97     if "leave" in no_sword_dark_action:
98         start()
99     elif "fight" or "kill" in no_sword_dark_action:
100        die("You need a sword to fight the monster. It kills you.")
101    else:
102        die("The only way to survive is to leave. As you did not, the monster kills
103 you.")
104
105 def monster_dead():
106     dark_room_list.remove('monster')
107
108
109 def extract_thing(user_command, action):
110     user_command.find(action)
111     pos = len(action)
112     thing = user_command[pos:]
113     return thing
114
115
116 #user_command = input("> ")
117
118 #if "hold " in user_command:
119 #    hold_thing = extract_thing(user_command, "hold ")
120 #    hand.append(hold_thing)
121 #    print("You are now holding:")
122 #    print(hand)
123
124 #if "drop " in user_command:
125 #    drop_thing = extract_thing(user_command, "drop ")
126 #    hand.remove(drop_thing)
127 #    print("You are now holding:")
128 #    print(hand)
129
130

```

```

131 def die(why):
132     print(why, "You lost; try again?")
133     exit_or_again()
134
135
136 def exit_or_again():
137     again = input("Type yes or no > ")
138
139     if "yes" in again:
140         start()
141     elif "no" in again:
142         exit(0)
143     else:
144         exit_or_again()
145 |
146
147 start()

```

ex37

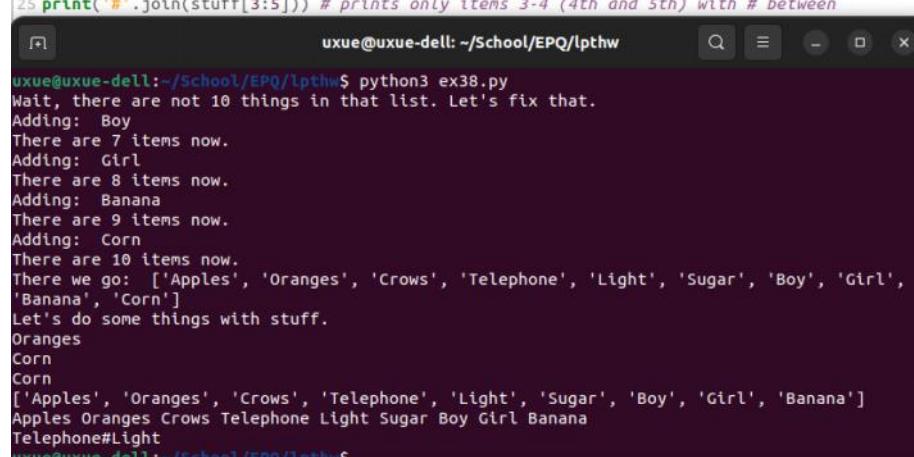
```

1 KEYWORDS
2
3 assert = ensure something is true
4     assert False, "Error!"
5 break = stop loop now
6     while True: break
7 continue = don't process more of loop, do it again
8     while True: continue
9 del = delete from dictionary
10    del X(Y)
11 except = if exception, do this
12     except ValueError, e: print(e)
13 exec = run string as Python
14     exec 'print("hello")'
15 finally = regardless of exceptions, do this
16     finally: pass
17 gobal = declare you want a global variable
18 is = like == to test equality
19     l is l == True
20 lambda = create short anonymous function
21     s = lambda y: y ** y; s(3)
22 pass = this block is empty
23     def empty(): pass
24 raise = raise exception when things go wrong
25     raise ValueError("No")
26 try = try this block; if exception, go to except
27     try: pass
28 with = with an expression as a variable do
29     with X as Y: pass
30 yield = pause here and return to caller
31     def X(): yield Y; X().next()
32
33
34 DATA TYPES
35
36 None = nothing or no value
37 bytes = stores bytes of text, PNG, file, etc.
38     x = b"hello"
39 floats = stores decimals
40 dicts = stores a key-value mapping of things
41     e = {'x': 1, 'y': 2}
42
43
44 STRING ESCAPE SEQUENCES
45
46 \a = Bell
47 \b = Backspace
48 \f = Formfeed
49 \r = Carriage
50 \v = Vertical tab
51
52
53 %i = %
54 %o = octal number
55 "%o" % 1000 == '1750'
56 %u = unsigned number
57 "%u" % -1000 == '-1000'
58 %x = hexadecimal lowercase
59 "%x" % 1000 == 3e8
60 %% = hexadecimal uppercase
61 %%e = exponential notation, lowercase e
62 "%e" % 1000 == '1.000000e+03'
63 %%E = exponential notation, uppercase E
64 %%f = floating point real number
65 %%F = %%F
66 %%g = whichever of %%f or %%e is shorter
67 %%G = uppercase %%F
68 %%c = character format
69 "%c" % 34 == ''
70 %%r = repr format (debugging format)
71 "%r" % int == "<type 'int'>"
72 %%s = string format
73 "%s there" % 'hi' == 'hi there'
74 %% = %
75
76
77
78
79
80
81 OPERATORS
82
83 ** = power of
84 // = floor division
85 % = string interpolate or modulus
86 @ = at (decorators)
87 = = assign

```

Ex38

```
1 ten_things = "Apples Oranges Crows Telephone Light Sugar"
2
3 print("Wait, there are not 10 things in that list. Let's fix that.")
4
5 stuff = ten_things.split(' ') # makes list where space means new item
6 more_stuff = ["Day", "Night", "Song", "Frisbee",
7                 "Corn", "Banana", "Girl", "Boy"]
8
9 while len(stuff) != 10:
10     next_one = more_stuff.pop() # takes last item in list
11     # more_stuff.pop() [call pop on more_stuff] = pop(more_stuff) [call pop with argument
12     more_stuff]
12     print("Adding: ", next_one)
13     stuff.append(next_one)
14     print(f"There are {len(stuff)} items now.")
15
16 print("There we go: ", stuff)
17
18 print("Let's do some things with stuff.")
19
20 print(stuff[1]) # print 2nd item
21 print(stuff[-1]) # prints last item of stuff
22 print(stuff.pop()) # takes last item of stuff
23 print(stuff)
24 print(' '.join(stuff)) # prints all items with space between
25 print('#'.join(stuff[3:5])) # prints only items 3-4 (4th and 5th) with # between
```



A terminal window titled 'uxue@uxue-dell: ~/School/EPQ/lpthw' showing the execution of the Python script 'ex38.py'. The output shows the program adding items to a list and then performing various string operations like printing specific items or joining them with different delimiters.

```
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex38.py
Wait, there are not 10 things in that list. Let's fix that.
Adding: Boy
There are 7 items now.
Adding: Girl
There are 8 items now.
Adding: Banana
There are 9 items now.
Adding: Corn
There are 10 items now.
There we go: ['Apples', 'Oranges', 'Crows', 'Telephone', 'Light', 'Sugar', 'Boy', 'Girl',
'Banana', 'Corn']
Let's do some things with stuff.
Oranges
Corn
Corn
['Apples', 'Oranges', 'Crows', 'Telephone', 'Light', 'Sugar', 'Boy', 'Girl', 'Banana']
Apples Oranges Crows Telephone Light Sugar Boy Girl Banana
Telephone#Light
uxue@uxue-dell: ~/School/EPQ/lpthw$
```

Ex39

```

1 # create a mapping of state to abbreviation
2 states = {
3     'Oregon': 'OR',
4     'Florida': 'FL',
5     'California': 'CA',
6     'New York': 'NY',
7     'Michigan': 'MI',
8 }
9
10 # create a basic set of states and some cities in them
11 cities = {
12     'CA': 'San Francisco',
13     'MI': 'Detroit',
14     'FL': 'Jacksonville',
15 }
16
17 # add some more cities
18 cities['NY'] = 'New York'
19 cities['OR'] = 'Portland'
20
21 # print out some cities
22 print("-" * 10)
23 print("NY State has: ", cities['NY'])
24 print("OR State has: ", cities['OR'])
25
26 # print some states
27 print("-" * 10)
28 print("Michigan's abbreviation is: ", states['Michigan'])
29 print("Florida's abbreviation is: ", states['Florida'])
30
31 # do it by using the state then cities dict
32 print("-" * 10)
33 print("Michigan has: ", cities[states['Michigan']])
34 print("Florida has: ", cities[states['Florida']])
35
36 # print every state abbreviation
37 print("-" * 10)
38 for state, abbrev in list(states.items()):
39     print(f"{state} is abbreviated {abbrev}")
40
41 # print every city in state
42 print("-" * 10)
43 for abbrev, city in list(cities.items()):
44     print(f"{abbrev} has the city {city}")
45
46 # now do both at the same time
47 print("-" * 10)
48 for state, abbrev in list(states.items()):
49     print(f"{state} state is abbreviated {abbrev}")
50     print(f"and has city {cities[abbrev]}")
51
52 print("-" * 10)
53 # safely get an abbreviation by state that might not be there
54 state = states.get('Texas')
55
56 if not state:
57     print("Sorry, no Texas")
58
59 # get a city with a default value
60 city = cities.get('TX', 'Does Not Exist')
61 print(f"The city for the state 'TX' is: {city}")

```

uxue@uxue-dell:~/School/EPQ/lpthw\$ python3 ex39.py

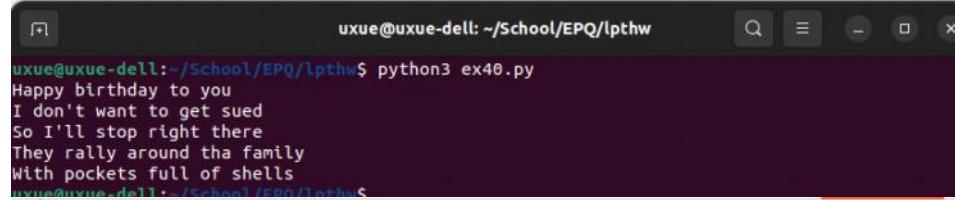
```

-----
NY State has: New York
OR State has: Portland
-----
Michigan's abbreviation is: MI
Florida's abbreviation is: FL
-----
Michigan has: Detroit
Florida has: Jacksonville
-----
Oregon is abbreviated OR
Florida is abbreviated FL
California is abbreviated CA
New York is abbreviated NY
Michigan is abbreviated MI
-----
CA has the city San Francisco
MI has the city Detroit
FL has the city Jacksonville
NY has the city New York
OR has the city Portland
-----
Oregon state is abbreviated OR
and has city Portland
Florida state is abbreviated FL
and has city Jacksonville
California state is abbreviated CA
and has city San Francisco
New York state is abbreviated NY
and has city New York
Michigan state is abbreviated MI
and has city Detroit
-----
Sorry, no Texas
The city for the state 'TX' is: Does Not Exist

```

Ex40

```
1 class Song(object):
2
3     def __init__(self, lyrics):
4         self.lyrics = lyrics
5
6     def sing_me_a_song(self):
7         for line in self.lyrics:
8             print(line)
9
10 happy_bday = Song(["Happy birthday to you",
11                     "I don't want to get sued",
12                     "So I'll stop right there"])
13
14 bulls_on_parade = Song(["They rally around tha family",
15                         "With pockets full of shells"])
16
17 happy_bday.sing_me_a_song()
18
19 bulls_on_parade.sing_me_a_song()
```



```
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex40.py
Happy birthday to you
I don't want to get sued
So I'll stop right there
They rally around tha family
With pockets full of shells
uxue@uxue-dell:~/School/EPQ/lpthw$
```

```
1 class = make new type of thing
2 object = most basic type of thing
3      = any instance of a thing
4 instance = what you get when you tell Python to create a class
5 self = inside class function, a variable for instance/object being tested
6 inheritance = one class inherits traits from another
7 composition = class is composed of other classes as parts
8 attribute = property of classes that are from composition, usually variables
9 is-a = phrase to say something inherits from another thing
10 has-a = phrase to say something is composed from another thing
11
12 class X(Y) "make class called X that is-a Y"
13 class X(object): def __init__(self, J) "class X has-a __init__ that takes self and J
parameters"
14 class X(object): def M(self, J)
```

Ex41

```
Make a class named Butto that is-a Cord.
> class Butto(Cord):
ANSWER: class Butto(Cord):

class Bead has-a function baseball that takes self and cause, copy params.
> class Bead(object):\n\ndef baseball(self, cause, copy):
ANSWER: class Bead(object):
    def baseball(self, cause, copy)

From apparatus get the dock attribute and set it to 'birthday'.
> apparatus.dock = 'birthday'
ANSWER: apparatus.dock = 'birthday'

class Arch has-a __init__ that takes self and change params.
> class Arch(object):
ANSWER: class Arch(object):
    def __init__(self, change)

From advice get the band function, call it with params self, crayon, copper, change.
> advice.band(self, crayon, copper, change)
ANSWER: advice.band(crayon, copper, change)

From cry get the cactus attribute and set it to 'drawer'.
> cry.cactus = 'drawer'
ANSWER: cry.cactus = 'drawer'

Set drain to an instance of class Desire.
> drain = Desire()
ANSWER: drain = Desire()

class Bird has-a function country that takes self and berry params.
> class Bird(object):\n\ndef country (self, berry):
ANSWER: class Bird(object):
    def country(self, berry)

From brother get the coach function, call it with params self, base, coach, copy.
> brother.coach(base, coach, copy)
ANSWER: brother.coach(base, coach, copy)
```

Ex42

```
1 ## Animal is-a object (yes, sort of confusing) look at the extra credit
2 class Animal(object):
3     pass
4
5 ## Dog is-a Animal, which is-a object
6 class Dog(Animal):
7
8     def __init__(self, name):
9         ## class Dog has-a __init__ that takes self and name parameters
10        self.name = name
11
12 ## Cat is-a Animal, which is-a object
13 class Cat(Animal):
14
15     def __init__(self, name):
16         ## class Cat has-a __init__ that takes self and name parameters
17         self.name = name
18
19 ## Person is-a object
20 class Person(object):
21
22     def __init__(self, name):
23         ## class Person has-a __init__ that takes self and name parameters
24         self.name = name
25
26         ## Person has-a pet of some kind
27         self.pet = None
28
29 ## Employee is-a Person, which is-a object
30 class Employee(Person):
31
32     def __init__(self, name, salary):
33         ## ?? hmm what is this strange magic?
34         super(Employee, self).__init__(name)
35         ## from self, get the salary attribute and call it salary
36         self.salary = salary
37
38 ## Fish is-a object
39 class Fish(object):
40     pass
41
42 ## Salmon is-a Fish, which is-a object
43 class Salmon(Fish):
44     pass
45
46 ## Halibut is-a Fish, which is-a object
47 class Halibut(Fish):
48     pass
49
50
51 ## rover is-a Dog
52 rover = Dog("Rover")
53
54 ## set Satan to instance "Satan" of class Cat
```

Ex44

```
1 class Parent(object):
2
3     def override(self):
4         print("PARENT override()")
5
6     def implicit(self):
7         print("PARENT implicit()")
8
9     def altered(self):
10        print("PARENT altered()")
11
12 class Child(Parent):
13
14     def override(self):
15         print("CHILD override()")
16
17     def altered(self):
18         print("CHILD, BEFORE PARENT altered()")
19         super(Child, self).altered()
20         print("CHILD, AFTER PARENT altered()")
21
22 dad = Parent()
23 son = Child()
24
25 dad.implicit() # prints PARENT implicit
26 son.implicit() # prints PARENT implicit
27
28 dad.override() # prints PARENT override
29 son.override() # prints CHILD override
30
31 dad.altered() # prints PARENT altered
32 son.altered() # prints CHILD, BEFORE PARENT altered; PARENT altered; CHILD, AFTER
33 # PARENT altered
```

```
uxue@uxue-dell:~/School/EPQ/lpthw$ python3 ex44.py
PARENT implicit()
PARENT implicit()
PARENT override()
CHILD override()
PARENT altered()
CHILD, BEFORE PARENT altered()
PARENT altered()
CHILD, AFTER PARENT altered()
```

Installing a virtual environment [16]

Monday, March 04, 2024 9:31 PM

Install the package that makes virtual environments (using pip, a package manager)

```
uxue@uxue-dell:~/School/EPQ/linear_regression$ pip install virtualenv
Defaulting to user installation because normal site-packages is not writeable
Collecting virtualenv
  Using cached virtualenv-20.25.1-py3-none-any.whl (3.8 MB)
Collecting filelock<4,>=3.12.2
  Using cached filelock-3.13.1-py3-none-any.whl (11 kB)
Collecting platformdirs<5,>=3.9.1
  Using cached platformdirs-4.2.0-py3-none-any.whl (17 kB)
Collecting distlib<1,>=0.3.7
  Using cached distlib-0.3.8-py2.py3-none-any.whl (468 kB)
Installing collected packages: distlib, platformdirs, filelock, virtualenv
WARNING: The script virtualenv is installed in '/home/uxue/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed distlib-0.3.8 filelock-3.13.1 platformdirs-4.2.0 virtualenv-20.25.1
uxue@uxue-dell:~/School/EPQ/linear_regression$ python3 -m venv Env_example
uxue@uxue-dell:~/School/EPQ/linear_regression$ source Env_example/bin/activate
(Env_example) uxue@uxue-dell:~/School/EPQ/linear_regression$ deactivate
uxue@uxue-dell:~/School/EPQ/linear_regression$
```

To activate virtual environment and deactivate:

```
uxue@uxue-dell:~/School/EPQ/linear_regression$ . venv/bin/activate
(venv) uxue@uxue-dell:~/School/EPQ/linear_regression$ deactivate
uxue@uxue-dell:~/School/EPQ/linear_regression$
```

Create a virtual environment called Env_example

Activate the virtual environment

Installing Jupyter Notebook [17] [18]

Monday, March 04, 2024 9:35 PM

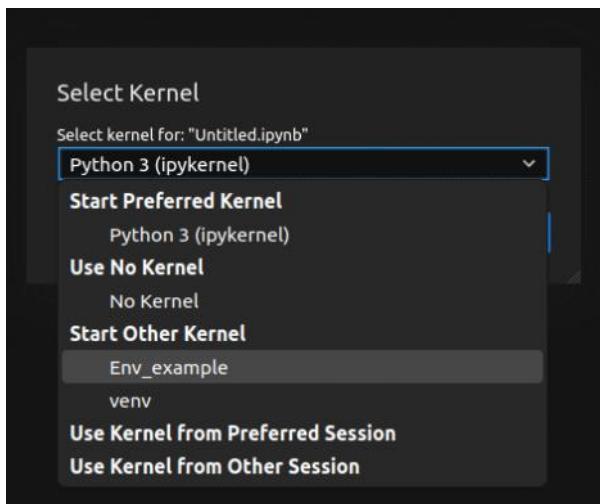
```
[Env_example] uxue@uxue-dell:~/School/EPQ/linear_regression$ pip install jupyter
Collecting notebook
  Using cached notebook-7.1.5-py3-none-any.whl (5.0 kB)
    ...
    Downloading jupyter_server-2.13.0-py3-none-any.whl (200 kB)
      ...
      Collecting jupyterlab-server==2.22.1
        ...
        Using cached jupyterlab_server-2.25.3-py3-none-any.whl (59 kB)
          ...
          Using cached tornado==6.2.0
            ...
            Collecting tornado-6.2.0-py3-none-any.whl (44 kB)
              ...
              Collecting nbformat-5.4.4-py3-none-any.whl (591 kB)
                ...
                Collecting nbconvert-6.2.0
                  ...
                  Using cached nbconvert-6.2.0-py3-none-any.whl (257 kB)
                    ...
                    Collecting nbformat-5.9.0-py3-none-any.whl (11.4 kB)
                      ...
                      Using cached nbformat-5.9.0-py3-none-any.whl (11.4 kB)
                        ...
                        Collecting packaging-23.2-py3-none-any.whl (53 kB)
                          ...
                          Collecting pygments-2.12-cp310-cp310-manylinux_2_28_x86_64.whl (5.1 kB)
                            ...
                            Collecting wsproto-client-1.7.0-py3-none-any.whl (58 kB)
                              ...
                              Collecting traitlets-5.1.1-py3-none-any.whl (85 kB)
                                ...
                                Collecting ipython-7.1.1-py3-none-any.whl (257 kB)
                                  ...
                                  Collecting nbconvert-6.2.0-py3-none-any.whl (257 kB)
                                    ...
                                    Collecting nbformat-5.9.0-py3-none-any.whl (77 kB)
                                      ...
                                      Collecting argon2-cffi
                                        ...
                                        Using cached cffi-23.1.0-py3-none-any.whl (15 kB)
                                          ...
                                          Collecting overrides
                                            ...
                                            Using cached overrides-7.7.0-py3-none-any.whl (17 kB)
                                              ...
                                              Collecting jupyter_client-8.0.0-py3-none-any.whl (105 kB)
                                                ...
                                                Collecting terminado-0.1.0
                                                  ...
                                                  Collecting nbconvert-6.2.0-py3-none-any.whl (14 kB)
                                                    ...
                                                    Collecting jupyter_server-terminals
                                                      ...
                                                      Using cached jupyter_server_terminals-0.5.2-py3-none-any.whl (13 kB)
                                                        ...
                                                        Collecting SendTrash-1.4.2-py3-none-any.whl (18 kB)
                                                          ...
                                                          Using cached jupyter-core-6.5.0+rc12
                                                            ...
                                                            Using cached jupyter_core-6.5.1-py3-none-any.whl (28 kB)
                                                              ...
                                                              Collecting anyio==3.1.0
                                                                ...
                                                                Collecting aiofiles-3.0.0
                      ...
                      Collecting Jinja2
                        ...
                        Using cached Jinja2-3.1.3-py3-none-any.whl (133 kB)
                          ...
                          Collecting ipython
                            ...
                            Using cached ipython-events-0.9.0-py3-none-any.whl (10 kB)
                              ...
                              Collecting pexpect
                                ...
                                Using cached pexpect-4.2.0-py3-none-any.whl (54 kB)
                                  ...
                                  Collecting https://27.0.0.1:5000
                                    ...
                                    Using cached https://27.0.0.1:5000-py3-none-any.whl (15 kB)
                                      ...
                                      Collecting async_lru-2.0.4-py3-none-any.whl (6.1 kB)
                                        ...
                                        Collecting toolz-1.0.0-py3-none-any.whl (32 kB)
                                          ...
                                          Collecting parso
                                            ...
                                            Collecting pandocfilters
                                              ...
                                              Collecting packaging
                                                ...
                                                Collecting overrides
                                                  ...
                                                  Collecting nest-asyncio
                                                    ...
                                                    Collecting mistune
                                                      ...
                                                      Collecting MarkupSafe
                                                        ...
                                                        Collecting jupyterlab-pygments
                                                          ...
                                                          Collecting jsonpointer
                                                            ...
                                                            Collecting json5
                                                              ...
                                                              Collecting idna
                                                                ...
                                                                Collecting httpcore
                                                                  ...
                                                                  Collecting certifi
                                                                    ...
                                                                    Collecting babel
                                                                      ...
                                                                      Collecting attrs
                                                                        ...
                                                                        Collecting terminado
                                                                          ...
                                                                          Collecting rfc3339-validator
                                                                            ...
                                                                            Collecting requests
                                                                              ...
                                                                              Collecting rfc3339
                                                                                ...
                                                                                Collecting re
                                                                                  ...
                                                                                  Collecting et-normalizer
                                                                                    ...
                                                                                    Collecting certifi
                                                                

```

Jupyter Notebook is web-based application where you can make documents with code, equations, text, and graphs. As a result, I am going to use it when doing machine learning tutorials, and perhaps for my project itself as well.

Open Jupyter Notebook (it then launches in the browser)

Use virtual environment (Env_example) in Jupyter Notebook



Select desired virtual environment when opening a new notebook in Jupyter

Linear regression using libraries

Monday, March 04, 2024 7:48 PM

This is inspired by the code in my [piece of research](#) on linear regression. I realised that the code in the tutorial was not very robust so I modified it to make it more resilient to different data

linearregressionwithlibraries-2

March 4, 2024

```
[89]: import numpy as np # numpy is a library that performs mathematical operations
import pandas as pd # pandas is used for data manipulation and analysis
import matplotlib.pyplot as plt # matplotlib (a component of numpy) is what
    ↪plots the graphs
import statsmodels.api as sm # statsmodels explores data, estimates statistic
    ↪models, and performs statistical tests
```

```
[90]: data = pd.read_csv('data.csv') # defining what the command "data" means - to
    ↪read the csv file
```

```
[91]: data # we are shown what the raw data given to us is, which we will use to
    ↪train and test the model
```

```
[91]:      SAT      GPA
0     1714    2.40
1     1664    2.52
2     1760    2.54
3     1685    2.74
4     1693    2.83
...
79    1936    3.71
80    1810    3.71
81    1987    3.73
82    1962    3.76
83    2050    3.81
```

[84 rows x 2 columns]

```
[92]: data.describe() # gives us descriptive statistics of the data
```

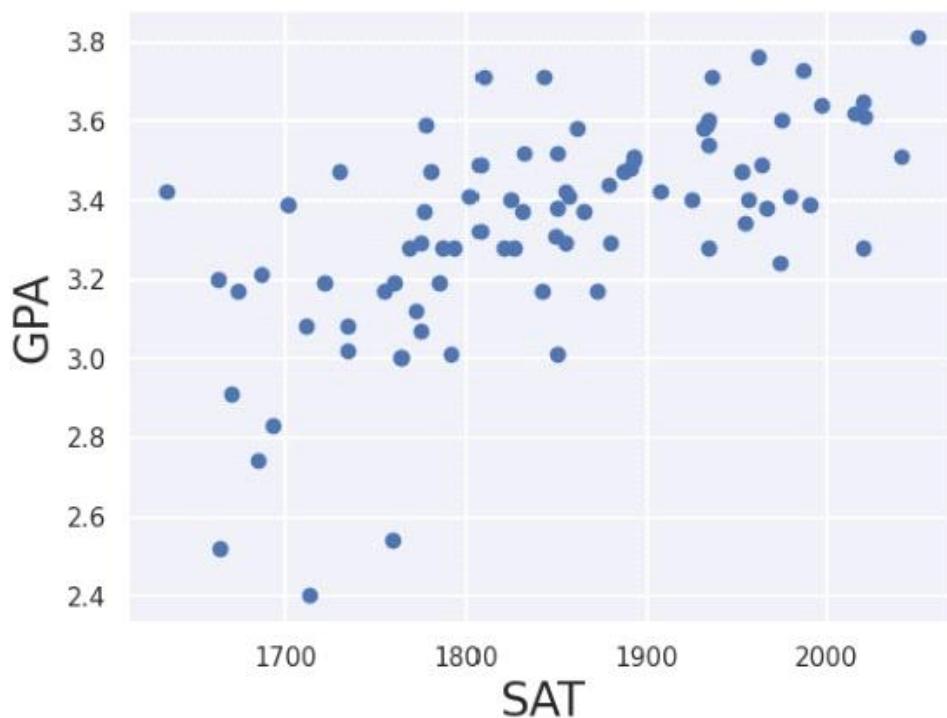
```
[92]:          SAT        GPA
count    84.000000  84.000000
mean    1845.273810  3.330238
std     104.530661  0.271617
min    1634.000000  2.400000
25%   1772.000000  3.190000
50%   1846.000000  3.380000
```

```
75%      1934.000000  3.502500
max      2050.000000  3.810000
```

```
[93]: y = data ['GPA'] # GPA is the dependent variable so is on the y-axis of the graph
x1 = data ['SAT'] # SAT is the independent variable so is on the x-axis
```

```
[94]: plt.scatter(x1,y) # plots a scatter plot
plt.xlabel('SAT', fontsize = 20) # the x-axis is labeled SAT
plt.ylabel('GPA', fontsize = 20) # the y-axis is labeled GPA
plt.show # shows the plot
```

```
[94]: <function matplotlib.pyplot.show(close=None, block=None)>
```



```
[95]: x = sm.add_constant(x1)
results = sm.OLS(y,x).fit()
summary = results.summary()
summary
```

```
[95]:
```

Dep. Variable:	GPA	R-squared:	0.406			
Model:	OLS	Adj. R-squared:	0.399			
Method:	Least Squares	F-statistic:	56.05			
Date:	Sun, 03 Mar 2024	Prob (F-statistic):	7.20e-11			
Time:	21:40:04	Log-Likelihood:	12.672			
No. Observations:	84	AIC:	-21.34			
Df Residuals:	82	BIC:	-16.48			
Df Model:	1					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	0.2750	0.409	0.673	0.503	-0.538	1.088
SAT	0.0017	0.000	7.487	0.000	0.001	0.002
Omnibus:	12.839	Durbin-Watson:	0.950			
Prob(Omnibus):	0.002	Jarque-Bera (JB):	16.155			
Skew:	-0.722	Prob(JB):	0.000310			
Kurtosis:	4.590	Cond. No.	3.29e+04			

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 3.29e+04. This might indicate that there are strong multicollinearity or other numerical problems.

```
[96]: summary.tables[1].data
# this gives us the second table (the one we are interested in) in a list form, so that we can take the part of the list we are interested # and use it for the line of best fit
```

```
[96]: [['', 'coef', 'std err', 't', 'P>|t|', '[0.025', '0.975'],
['const',
 ' 0.2750',
 ' 0.409',
 ' 0.673',
 ' 0.503',
 ' -0.538',
 ' 1.088'],
['SAT',
 ' 0.0017',
 ' 0.000',
 ' 7.487',
 ' 0.000',
 ' 0.001',
 ' 0.002']]
```

```
[97]: c = float(table2[1][1]) # we want 0.02750, which is the 1st term of the 1st list (numbering of lists starts from 0)
m = float(table2[2][1]) # we want 0.0017, which is the 1st term of the 2nd list
print(f"m = {m}")
```

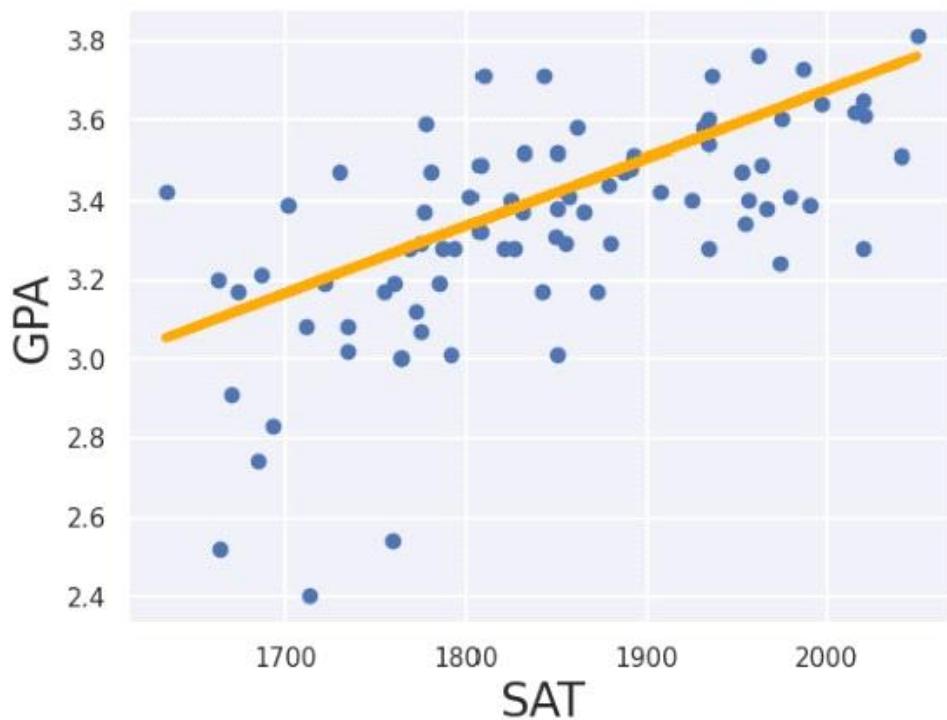
```

print(f"c = {c}")

m = 0.0017
c = 0.275

[98]: plt.scatter(x1,y) # plot a scatter plot
yhat = m*x1 + c
# define the regression equation, to plot it later
plt.plot(x1,yhat, lw=4, c='orange', label = 'regression line') # plot the
# regression line against the independent variable, SAT
plt.xlabel('SAT', fontsize = 20) # label the axes
plt.ylabel('GPA', fontsize = 20)
plt.show()

```



We have created a line of best fit for the data!

Creating a csv file in Python

Thursday, May 16, 2024 10:27 PM

I learnt how to make a dictionary in Learn Python the Hard Way, so a way of making a csv file is by converting the dictionary into a csv file. I can do this using Pandas library, using the `to_csv` command. Source: <https://www.datacamp.com/tutorial/save-as-csv-pandas-dataframe>

```
[1]: import pandas as pd
[10]: food = {'fruit': ['apple', 'pear', 'banana'], 'vegetable': ['carrot', 'pea', 'lettuce']} # making dictionary
[4]: df = pd.DataFrame(food) # making dictionary into a data frame
[5]: df # showing data frame
[5]:
   fruit  vegetable
0  apple      carrot
1    pear        pea
2  banana     lettuce
[11]: df.to_csv('fruit.csv', index=False) # making data frame into a csv file, without an extra column showing numbers
```

I now have a csv file called fruit.csv:

	fruit	vegetable
1	apple	carrot
2	pear	pea
3	banana	lettuce

GitHub

Sunday July 14, 2024 3:33 PM

GitHub is an online platform where developers can upload coding projects.

Different parts:

- README
 - As the name says, this is the part that the user reads to understand what the project is and does, the context and purpose, and how to use the code
- Requirements
 - This is the packages that you need to install if you want to do the project yourself
- Scripts
 - The important parts of the project that you can save and run yourself but it may be difficult to make sense of what to do with them without the instructions from the README file



I created a profile "adimena" on GitHub to post these projects in case I want to share them with anyone, and so that my files are backed up (I don't think OneDrive can store Python and Jupyter Notebook files).

I can files to a repository on the website, which is how I first made my "EPQ" repository.

The EPQ repository contains several Python files related to machine learning models. The README file includes a note about the repository being a collection of machine learning projects for my EPQ.

But, there is also a way of making and editing my repository from the command line.

```
(~/EPQ) user@user:~/Desktop$ cd EPQ/clean/
~/EPQ/clean$ git add .
~/EPQ/clean$ git commit -m "Initial Commit"
[adimena EPQ] Initial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
~/EPQ/clean$ git push -u origin master
Enumerating objects: 12, done.
Delta compression using up to 8 threads
Writing objects: 100% (12/12), 238 bytes | 238.00 KB/s, done.
Total 12 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Total 12 (delta 8), reused 0 (delta 0), pack-reused 0
remote: 
remote: Processing delta objects: 100% (8/8), done.
remote: 
remote: Writing objects: 100% (12/12), done.
remote: 
remote: To https://github.com/adimena/EPQ-vs-no-cows.git
   0a3f3d1..3e3a2c1  master >--> master
~/EPQ/clean$
```

Here I have cloned the GitHub repository on the cloud onto my device, into the folder "EPQ_clean".

Using the git clone command

```
[~/EPQ] user@user:~/Desktop$ git clone https://github.com/adimena/EPQ-vs-no-cows.git
Cloning into 'EPQ-vs-no-cows'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (12/12), done.
remote: Writing objects: 100% (12/12), done.
remote: Total 12 (delta 8), reused 0 (delta 0), pack-reused 0
remote: 
remote: Processing deltas: 100% (8/8), done.
remote: 
remote: Writing objects: 100% (3/3), done.
remote: 
remote: To https://github.com/adimena/EPQ-vs-no-cows.git
   0a3f3d1..3e3a2c1  master >--> master
~/EPQ
```

Shows the command I have all of the files in this directory, all of the files in the repository. I want to remove the folder "cowscat".

I use the normal tree command to remove a directory, but add ! at the front of the command.

```
[~/EPQ] user@user:~/Desktop$ rm -rf !cowscat/
~/EPQ$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git add ..." to stage)

The file has not yet been deleted yet. I have just expressed the intention to do so. I need to commit the change to my repository to confirm the change.
```

```
[~/EPQ] user@user:~/Desktop$ git commit -m "Delete cowscat Folder"
[adimena EPQ] Delete cowscat Folder
 1 file changed, 1 deletion(-)
 delete mode 100800 cowscat/photos/ttfile.py
~/EPQ$ git push
Enumerating objects: 3, done.
Delta compression using up to 8 threads
Writing objects: 100% (3/3), 238 bytes | 238.00 KB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: 
remote: Processing deltas: 100% (3/3), completed with 1 local object.
remote: 
remote: Writing objects: 100% (3/3), done.
remote: 
remote: To https://github.com/adimena/EPQ-vs-no-cows.git
   3e3a2c1..0a3f3d1  master >--> master
~/EPQ$
```

Finally, git push -u origin master change is made on the remote repository (the one visible to everyone on the internet) as well.

```
[~/EPQ] user@user:~/Desktop$ git log
[adimena EPQ] Delete cowscat Folder
 1 file changed, 1 deletion(-)
 delete mode 100800 cowscat/photos/ttfile.py
[adimena EPQ] Initial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
[adimena EPQ] Initial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
~/EPQ$
```

The highlighted shows the latest change made. So my change made from the command line has worked.

This is what the repository for my neural network looks like:





GitHub

04 September 2024 08:53

GitHub is an online platform where developers can upload coding projects.

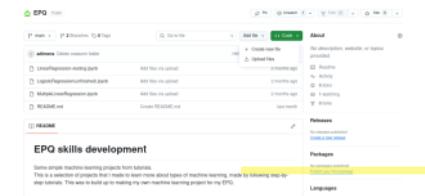
Different parts:

- README
 - As the name says, this is the part that the user reads to understand what the project is and its context and purpose, and how to use the code
- Requirements
 - This is the packages that you need to install if you want to do the project yourself
- Project
 - The important parts of the project that you can save and run yourself, but it may be difficult to make sense of what to do with them without the instructions from the README file



I created a profile "adimena" on GitHub to post these projects in case I want to share them with anyone, and so that my files are backed up (I don't think OneDrive can store Python and Jupyter Notebook files).

I can files to a repository on the website, which is how I first made my "EPQ" repository.



But, there is also a way of making and editing my repository from the command line.

```
(fenny) ~:~/Documents/EPQ$ cd EPQ_clean/
(fenny) ~:~/Documents/EPQ$ git clone git@github.com:adimena/Cows-vs-no-cows.git
Cloning into 'Cows-vs-no-cows'...
remote: Counting objects: 1000 (done)
remote: Compressing objects: 100% (129/129), done.
remote: Writing objects: 100% (129/129), pack-reduced 0
remote: Total 129 (delta 83), reused 0 (delta 0), pack-reduced 0
Resolving deltas: 100% (129/129), 12.47 MiB / 5.09 MiB/s, done.
(fenny) ~:~/Documents/EPQ$
```

Here I have cloned the GitHub repository on the cloud onto my device, into the folder "EPQ_clean", using the git clone command

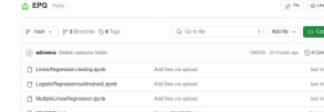
```
(fenny) ~:~/Documents/EPQ$ ls -l
total 0
(fenny) ~:~/Documents/EPQ$ cd EPQ_clean/
(fenny) ~:~/Documents/EPQ$ git rm -r .gitignore
```

I use the normal shell command to remove a directory, but put git in front of the command.

```
(fenny) ~:~/Documents/EPQ$ git status
# On branch main
# Your branch is up-to-date with 'origin/main'.
# Changes to be committed:
#   (use "git restore --staged <file>" to unstage)
#     (use "git add <file>" to include in what's tracked)
#       (use "git rm <file>" to untrack)
The file has not yet been deleted yet, I have just expressed the intention to do so. I need to commit the change to my repository to confirm
```

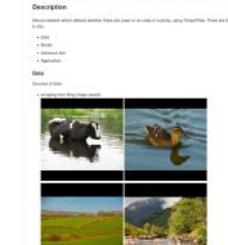
```
(fenny) ~:~/Documents/EPQ$ git commit
[master (root-commit) 1e0a54d] Remove .gitignore file
 1 file changed, 0 deletion(-)
 1 file removed
(fenny) ~:~/Documents/EPQ$ git push
Counting objects: 2, done.
Delta compression objects: 100% (1/1).
Writing objects: 100% (1/1), 12.47 MiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reduced 0
remote: warning: Cows-vs-no-cows: Your push to 'main' was rejected because it contained local objects.
To git@github.com:adimena/Cows-vs-no-cows.git
 ! [rejected]  (non-fast-forward) Cows-vs-no-cows
```

Finally, git push will mean this change is made on the remote repository (the one visible to everyone on the internet) as well.



The highlighted shows the latest change made. So my change made from the command line has worked.

This is what the repository for my neural network looks like:





Project log

Sunday, July 07, 2024 12:34 PM

Neural network – cats vs dogs [44]

Saturday, June 29, 2024 10:59 AM

This is a prototype for my neural network, which distinguishes between images of cats and images of dogs. I used a tutorial (see source 44).

Jupyter Notebook:

```
In [31]: import os
import zipfile
import random
import tensorflow as tf
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from shutil import copyfile

In [32]: local_zip = '/home/uxue/projects/EPO/kagglecatsanddogs.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/home/uxue/projects/EPO')
zip_ref.close()

In [33]: print(len(os.listdir('./PetImages/Cat/')))
print(len(os.listdir('./PetImages/Dog/')))

12501
12501

In [34]: to_create = [
    './cats-v-dogs',
    './cats-v-dogs/training',
    './cats-v-dogs/testing',
    './cats-v-dogs/training/cats',
    './cats-v-dogs/training/dogs',
    './cats-v-dogs/testing/cats',
    './cats-v-dogs/testing/dogs'
]

for directory in to_create:
    try:
        os.mkdir(directory)
        print(directory, 'created')
    except:
        print(directory, 'failed')

./cats-v-dogs failed
./cats-v-dogs/training failed
./cats-v-dogs/testing failed
./cats-v-dogs/training/cats failed
./cats-v-dogs/training/dogs failed
./cats-v-dogs/testing/cats failed
./cats-v-dogs/testing/dogs failed

In [35]: def split_data(SOURCE, TRAINING, TESTING, SPLIT_SIZE):
    all_files = []

    for file_name in os.listdir(SOURCE):
        file_path = SOURCE + file_name

        if os.path.getsize(file_path):
            all_files.append(file_name)
        else:
            print('{} is zero length, so ignoring'.format(file_name))

    n_files = len(all_files)
    split_point = int(n_files * SPLIT_SIZE)

    shuffled = random.sample(all_files, n_files)

    train_set = shuffled[:split_point]
    test_set = shuffled[split_point:]

    for file_name in train_set:
        copyfile(SOURCE + file_name, TRAINING + file_name)

    for file_name in test_set:
        copyfile(SOURCE + file_name, TESTING + file_name)

CAT_SOURCE_DIR = './PetImages/Cat/'
TRAINING_CATS_DIR = './cats-v-dogs/training/cats/'
TESTING_CATS_DIR = './cats-v-dogs/testing/cats/'
DOG_SOURCE_DIR = './PetImages/Dog/'
TRAINING_DOGS_DIR = './cats-v-dogs/training/dogs/'
TESTING_DOGS_DIR = './cats-v-dogs/testing/dogs/'

split_size = .9
split_data(CAT_SOURCE_DIR, TRAINING_CATS_DIR, TESTING_CATS_DIR, split_size)
split_data(DOG_SOURCE_DIR, TRAINING_DOGS_DIR, TESTING_DOGS_DIR, split_size)

666.jpg is zero length, so ignoring
11702.jpg is zero length, so ignoring

In [36]: print(len(os.listdir('./cats-v-dogs/training/cats/')))
print(len(os.listdir('./cats-v-dogs/training/dogs/')))
print(len(os.listdir('./cats-v-dogs/testing/cats/')))
print(len(os.listdir('./cats-v-dogs/testing/dogs/')))

12486
12492
3382
3390

In [44]: model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), input_shape=(150, 150, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(128, activation='relu'),
```

```

666.jpg is zero length, so ignoring
11702.jpg is zero length, so ignoring

In [36]: print(len(os.listdir("./cats-v-dogs/training/cats/")))
print(len(os.listdir("./cats-v-dogs/training/dogs/")))
print(len(os.listdir("./cats-v-dogs/testing/cats/")))
print(len(os.listdir("./cats-v-dogs/testing/dogs/")))

12486
12492
3382
3390

In [44]: model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), input_shape=(150, 150, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer=RMSprop(learning_rate=0.001), loss='binary_crossentropy', metrics=['acc'])

TRAINING_DIR = './cats-v-dogs/training'
train_datagen = ImageDataGenerator(
    rescale=1 / 255,
    rotation_range=40,
    width_shift_range=.2,
    height_shift_range=.2,
    shear_range=.2,
    zoom_range=.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
train_generator = train_datagen.flow_from_directory(
    TRAINING_DIR,
    batch_size=64,
    class_mode='binary',
    target_size=(150, 150)
)

VALIDATION_DIR = './cats-v-dogs/testing'
validation_datagen = ImageDataGenerator(
    rescale=1 / 255,
    rotation_range=40,
    width_shift_range=.2,
    height_shift_range=.2,
    shear_range=.2,
    zoom_range=.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
validation_generator = validation_datagen.flow_from_directory(
    VALIDATION_DIR,
    batch_size=64,
    class_mode='binary',
    target_size=(150, 150)
)

```

Found 24976 images belonging to 2 classes.
 Found 6772 images belonging to 2 classes.

```

In [48]: import warnings
from PIL import Image as pil_image

#warnings.filterwarnings('ignore')
#print('warnings ignored')

history = model.fit(train_generator,
                     epochs=15,
                     verbose=1,
                     validation_data=validation_generator)

```

```

-----
ImportError: Traceback (most recent call last)
Cell In[48], line 7
  2 from PIL import Image as pil_image
  4 #warnings.filterwarnings('ignore')
  5 #print('warnings ignored')
-> 7 history = model.fit(train_generator,
  8                     epochs=15,
  9                     verbose=1,
 10                     validation_data=validation_generator)

File ~/projects/EP0/tfenv/lib/python3.10/site-packages/keras/src/utils/traceback_utils.py:122, in filter_traceback
  119     filtered_tb = _process_traceback_frames(e._traceback)
  120     # To get the full stack trace, call:
  121     # `keras.config.disable.traceback_filtering()`
-> 122     raise e.with_traceback(filtered_tb) from None
 123 finally:
 124     del filtered_tb

File ~/projects/EP0/tfenv/lib/python3.10/site-packages/keras/src/utils/image_utils.py:227, in load_img(path, color_mode, target_size, interpolation, keep_aspect_ratio)
 195 """Loads an image into PIL format.
 196 
 197 Example:
 198     (1) A PIL Image instance.
 199     ...
 200     if pil_image is None:
-> 201         raise ImportError(
 202             "Could not import PIL.Image. The use of 'load_img' requires PIL."
 203         )
 204     if isinstance(path, io.BytesIO):
 205         img = pil_image.open(path)

ImportError: Could not import PIL.Image. The use of 'load_img' requires PIL.

```

I couldn't work out how to get past this error message, as I downloaded PIL as it asked to but it still had the same message. I decided to write the code on a text editor instead of Jupyter Notebook, and it

works when I run it on the terminal, so I am using the text editor from now on.

Using text editor and then running from command line:

The following splits the data into training and testing sets (about 10% of the data is usually used for testing).

```
1 import os
2 import zipfile
3 import random
4 import tensorflow as tf
5 from tensorflow.keras.optimizers import RMSprop
6 from tensorflow.keras.preprocessing.image import ImageDataGenerator
7 from shutil import copyfile
8
9 to_create = [
10     './cats-v-dogs',
11     './cats-v-dogs/training',
12     './cats-v-dogs/testing',
13     './cats-v-dogs/training/cats',
14     './cats-v-dogs/training/dogs',
15     './cats-v-dogs/testing/cats',
16     './cats-v-dogs/testing/dogs'
17 ]
18
19 for directory in to_create:
20     try:
21         os.mkdir(directory)
22         print(directory, 'created')
23     except:
24         print(directory, 'failed')
25
26 def split_data(SOURCE, TRAINING, TESTING, SPLIT_SIZE):
27     all_files = []
28
29     for file_name in os.listdir(SOURCE):
30         file_path = SOURCE + file_name
31
32         if os.path.getsize(file_path):
33             all_files.append(file_name)
34         else:
35             print('() is zero length, so ignoring'.format(file_name))
36
37     n_files = len(all_files)
38     split_point = int(n_files * SPLIT_SIZE)
39
40     shuffled = random.sample(all_files, n_files)
41
42     train_set = shuffled[:split_point]
43     test_set = shuffled[split_point:]
44
45     for file_name in train_set:
46         copyfile(SOURCE + file_name, TRAINING + file_name)
47
48     for file_name in test_set:
49         copyfile(SOURCE + file_name, TESTING + file_name)
50
51
52 CAT_SOURCE_DIR = "./PetImages/Cat/"
53 TRAINING_CATS_DIR = "./cats-v-dogs/training/cats"
54 TESTING_CATS_DIR = "./cats-v-dogs/testing/cats"
55 DOG_SOURCE_DIR = "./PetImages/Dog/"
56 TRAINING_DOGS_DIR = "./cats-v-dogs/training/dogs"
57 TESTING_DOGS_DIR = "./cats-v-dogs/testing/dogs"
58
59 split_size = .9
60 split_data(CAT_SOURCE_DIR, TRAINING_CATS_DIR, TESTING_CATS_DIR, split_size)
61 split_data(DOG_SOURCE_DIR, TRAINING_DOGS_DIR, TESTING_DOGS_DIR, split_size)
```

This script is the training script. It makes the neural network and trains it with the training data.

```
1 import tensorflow as tf
2 from tensorflow.keras.preprocessing.image import ImageDataGenerator
3
4 model = tf.keras.models.Sequential([
5     tf.keras.layers.Conv2D(16, (3,3), input_shape=(150, 150, 3), activation='relu'),
6     tf.keras.layers.MaxPooling2D(2,2),
7     tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
8     tf.keras.layers.MaxPooling2D(2,2),
9     tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
10    tf.keras.layers.MaxPooling2D(2,2),
11    tf.keras.layers.Flatten(),
12    tf.keras.layers.Dense(128, activation='relu'),
13    tf.keras.layers.Dense(128, activation='relu'),
14    tf.keras.layers.Dense(1, activation='sigmoid')
15 ])
16
17 model.compile(optimizer=tf.keras.optimizers.RMSprop(learning_rate=0.001), loss='binary_crossentropy', metrics=['acc'])
18
19
20 TRAINING_DIR = './projects/EPO/cats-v-dogs/training'
21 train_datagen = ImageDataGenerator(
22     rescale=1/255,
23     rotation_range=40,
24     width_shift_range=.2,
25     height_shift_range=.2,
26     shear_range=.2,
27     zoom_range=.2,
28     horizontal_flip=True,
29     fill_mode='nearest')
30
31 train_generator = train_datagen.flow_from_directory(
32     TRAINING_DIR,
33     target_size=(150, 150),
34     batch_size=128,
35     class_mode='binary',
36     target_size=(150, 150)
37
38 VALIDATION_DIR = './projects/EPO/cats-v-dogs/testing'
39 validation_datagen = ImageDataGenerator(
40     rescale=1/255,
41     rotation_range=40,
42     width_shift_range=.2,
43     height_shift_range=.2,
44     shear_range=.2,
45     zoom_range=.2,
46     horizontal_flip=True,
47     fill_mode='nearest')
48
49
50 validation_generator = validation_datagen.flow_from_directory(
51     VALIDATION_DIR,
52     target_size=(150, 150),
53     batch_size=32,
54     class_mode='binary',
55     target_size=(150, 150)
56 )
57
58 import warnings
59 from PIL import Image as pil_image
60
61 warnings.filterwarnings('ignore')
62 #print('Warnings ignored')
63
64 history = model.fit(train_generator,
65                      epochs=15,
66                      verbose=1,
67                      validation_data=validation_generator)
```

Here I am running it for the first time on the command line.

The above shows the model training. There are 15 epochs – so it goes through the whole testing and training data 15 times over. By the end, the accuracy when it goes through the testing set (validation accuracy) is 87%.

However, it takes a long time on my laptop – 7673 seconds according to this, 128 minutes, 2h 8min.

As a result, I am going to use a more powerful computer to train the model, by ssh-ing onto my brother's gaming computer.

SSH is a way of connecting to another computer and managing it, transferring files and executing

commands on that computer.

Command: ssh [username]@[host_ip_address]
(source: https://phoenixnap.com/kb/ssh-to-connect-to-remote-server-linux-or-windows#ftoc_)

```
uxue@UxueLatitude7490: ~ ssh aster@192.168.0.15
aster@192.168.0.15's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

201 updates can be applied immediately.
6 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

8 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com

Last login: Sat Jun 29 21:29:19 2024 from 192.168.0.11
```

As you can see from the command at the bottom, I am now on the shell on my brother's computer. I trained the model on here and copied the model (a file ending in .keras, which I made sure the training code would save at the end of training) onto my computer.

Source: <https://medium.com/@tericcabrel/how-to-copy-files-over-ssh-8aff6b7c560f>

```
uxue@UxueLatitude7490: ~$ scp asier@192.168.0.15:/home/asier/projects/EPQ/cnn3.keras ./projects/EPQ
asier@192.168.0.15's password:
          100%  146MB   9.6MB/s   00:15
```

Inference script

This is the script in which the model I saved categorises all unseen data in the cats folder and the dogs folder and calculates the percentage that are labelled "dogs". This means the number under "cats" must be as close as possible to 0, and the number under "dogs" must be as close as possible to 1.

```
1 ===
2 Inference for cats vs dogs
3 ===
4
5 import glob
6 import keras
7 import numpy as np
8
9 from keras.preprocessing import image
10
11 # load model
12
13 model = keras.models.load_model('./models/cnn3.keras')
14
15 model.compile(loss='binary_crossentropy',
16                 optimizer='rmsprop',
17                 metrics[accuracy])
18
19 # inference
20
21 for item in ['cats','dogs']:
22
23     print(item)
24     print(" - *")
25     data = []
26
27     for idx, name in enumerate(glob.glob("./catsdogsdataset/testing/[item]/*.jpg")):
28         # go through all images in "testing" folder
29
30         # prepare to use model on image
31         img = image.load_img(name, target_size=(150,150))
32         x = image.img_to_array(img)
33         x = np.expand_dims(x, axis=0)
34
35         # make list with all images in cats or dogs folder
36         data.append(x)
37
38         # stop after 1000 images
39         if idx > 1000:
40             break
41
42     images = np.vstack(data) # make into a grid
43     classes = model.predict(images) # find probability of being dog (since cats is 0 and dogs is 1)
44     results = []
45
46     for c in classes:
47         result = int(c[0] > 0.5) # c[0] is just c but without the square brackets
48         results.append(result)
49
50     print(f"images = {len(results)}")
51     print(f"how many times model chose dogs (correctly and incorrectly) = {sum(results)}")
52     print(f"{sum(results)}/{len(results)}")
53     print("\n")
```

This was inspired by the code from the tutorial:

```
import numpy as np
from google.colab import files
from keras.preprocessing import image

uploaded = files.upload()

for fn in uploaded.keys():

    # predicting images
    path = '/content/' + fn
    img = image.load_img(path, target_size=(150,150))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    images = np.vstack([x])
    classes = model.predict(images, batch_size=10)
    print(classes[0])
    if classes[0]>0.5:
        print(fn + " is a dog")
    else:
        print(fn + " is a cat")
```

But I changed it so that instead of going through one images at a time, it went through all of the testing images.

```
313/313 44s 139ms/step
0.6436712657468506
dogs
-----
/home/uxue/projects/EPQ/tfenv/lib/python3.10/
  warnings.warn(str(msg))
2024-07-02 12:24:48.692344: W external/local_
313/313 43s 136ms/step
0.98000399920016
```

The number above dogs corresponds to the proportion of cat images that have been predicted as dogs. This number is very high when it should be as close as possible to 0. The next decimal should be as close as possible to 1, but nonetheless feels too high to signify that the model is accurate. Thus the problem is that there is a bias towards predicting dogs.

```
(venv2) (base) asier@z390:~/projects/EPQ$ ls cats-v-dogs/testing/cats | wc -l
8978
(venv2) (base) asier@z390:~/projects/EPQ$ ls cats-v-dogs/testing/dogs | wc -l
8772
(venv2) (base) asier@z390:~/projects/EPQ$ ls cats-v-dogs/training/dogs | wc -l
12500
(venv2) (base) asier@z390:~/projects/EPQ$ ls cats-v-dogs/training/cats | wc -l
12500
(venv2) (base) asier@z390:~/projects/EPQ$ rm -rf cats-v-dogs/
```

Problem identified – strange numbers in dataset (here I am getting the computer to print the number of items in the specified folders). From the tutorial there should be 11250 in each training folder, not 12500. There should be 1250 in each testing folder.

The last command is deleting that whole directory so that I can create a new one which will have the correct number of items in each folder.

```
(venv2) (base) aster@z390:~/projects/EPQ$ python3 splitdata.py
2024-07-02 18:42:50.950974: E external/local_xla/xla/stream_executor/cuda/cuda_
tered
2024-07-02 18:42:50.971382: E external/local_xla/xla/stream_executor/cuda/cuda_
tered
2024-07-02 18:42:50.971419: E external/local_xla/xla/stream_executor/cuda/cuda_
egistered
2024-07-02 18:42:50.984154: I tensorflow/core/platform/cpu_feature_guard.cc:210
To enable the following instructions: AVX2 FMA, in other operations, rebuild Te
2024-07-02 18:42:51.687006: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc
./cats-v-dogs created
./cats-v-dogs/training created
./cats-v-dogs/testing created
./cats-v-dogs/training/cats created
./cats-v-dogs/training/dogs created
./cats-v-dogs/testing/cats created
./cats-v-dogs/testing/dogs created
666.jpg is zero length, so ignoring
11792.jpg is zero length, so ignoring
(venv2) (base) aster@z390:~/projects/EPQ$ ls
cats-v-dogs  catsvdogs.py  catsvdogs.py-  cnn4.keras  cnn.keras
(venv2) (base) aster@z390:~/projects/EPQ$ ls cats-v-dogs/
testing  training
(venv2) (base) aster@z390:~/projects/EPQ$ ls cats-v-dogs/testing
cats  dogs
(venv2) (base) aster@z390:~/projects/EPQ$ ls cats-v-dogs/training
cats  dogs
(venv2) (base) aster@z390:~/projects/EPQ$ ls cats-v-dogs/training | wc -l
2
(venv2) (base) aster@z390:~/projects/EPQ$ ls cats-v-dogs/training/cats | wc -l
11250
(venv2) (base) aster@z390:~/projects/EPQ$ ls cats-v-dogs/training/dogs | wc -l
11250
(venv2) (base) aster@z390:~/projects/EPQ$ ls cats-v-dogs/testing/dogs | wc -l
1250
(venv2) (base) aster@z390:~/projects/EPQ$ ls cats-v-dogs/testing/cats | wc -l
1250
```

I ran the script again that splits the data, and the directories all have the correct number now.

```
cats
-----
2024-07-02 22:37:10.655594: W external/local_tsl/tsl/
313/313 39s 125ms/step
0.3254349130173965
> /home/uxue/projects/EPQ/inference.py(36)<module>()
-> pass
(Pdb) c
dogs
-----
/home/uxue/projects/EPQ/tfenv/lib/python3.10/site-pac
  warnings.warn(str(msg))
2024-07-02 22:38:34.113488: W external/local_tsl/tsl/
313/313 40s 129ms/step
0.8371325734853029
```

This time, only around 30% of cats are categorised as dogs, so 70% of cats are categorised correctly; and over 80% of dogs are categorised correctly.

Thus accuracy shown here is around $(0.837 + (1 - 0.325)) / 2 = 0.756$.

The accuracy at epoch 15 in training is 0.8299.

```
Epoch 15/15
391/391 145s 365ms/step - acc: 0.8414 - loss: 0.3599 - val_acc: 0.8299 - val_loss: 0.3890
```

This is evidence of the model training slightly too well on the training data, then not doing well enough on unseen data.

Should ideally be higher than this, however, it is something I can work with.

Making my own dataset

Saturday, June 29, 2024 1:43 PM

There are no datasets on the internet of cows/no cows, so I need to create my own. I need a few thousand images of cows and no cows. My sources of data will be:

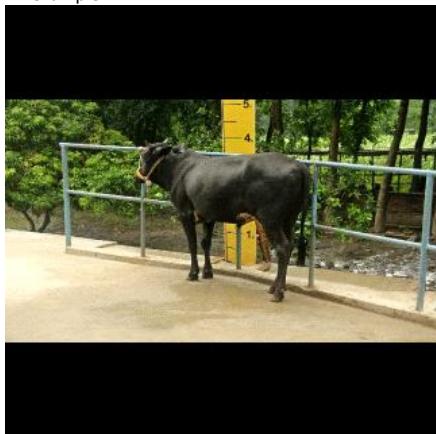
- Taking photos myself
- Taking videos of cows and no cows, and extracting images from these videos
 - Hopefully will have a similar effect to taking photos, but it means I can generate photos more quickly
- Using datasets from the internet of cows and fields
- "Scraping"
 - Making an algorithm that saves every photo on a particular webpage, saving me from copying each one manually
 - I can use Google Image Search with search terms such as "cows" and "fields"

Dataset online

I couldn't find any datasets to put in the "no cows" folder, but I found a dataset from Kaggle with a variety of cow breeds at different angles.

<https://www.kaggle.com/datasets/anandkumarsahu09/cattle-breeds-dataset?resource=download>

An example:



Taking videos & photos

I went out onto the fields to find cows and took some photos with cows and without cows.



I also took some videos of cows (making sure there was at least one cow in every single frame) and some videos without cows (making sure there were no cows in any frame).



Then, with help from here (<https://stackoverflow.com/questions/34786669/extract-all-video-frames-as-images-with-ffmpeg>) I extracted photos from one of the videos, using the command

ffmpeg -i video.mp4 photoname%04d.jpg -hide_banner

```
(tfenv) uxue@uxuelatitude7490:~/projects/EPIC/cows/nocows/v1photos$ ffmpeg -i nocow1.mp4 nocow1%04d.jpg -hide_banner
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'nocow1.mp4':
  Metadata:
    major_brand     : mp42
    minor_version  : 0
    compatible_brands: isommp42
  creation_time   : 2024-06-29T15:24:25.000000Z
  location        : +52.2162+000.1176/
  location-eng    : +52.2162+000.1176/
  con.android.version: 12
Duration: 00:00:07.09, start: 0.000000, bitrate: 20487 kb/s
Stream #0:0(eng): Video: h264 (High) (avc1 / 0x31637661), yuv420p(tv, bt709), 1920x1080, 19744 kb/s, 29.63 fps, 29.67 tbr, 90k tbn, 180k tbc (default)
  Metadata:
    rotate         : 90
    creation_time  : 2024-06-29T15:24:25.000000Z
    handler_name   : VideoHandle
    vendor_id      : [0][0][0][0]
  Side data:
    displaymatrix: rotation of -90.00 degrees
Stream #0:1(eng): Audio: aac (LC) (mp4a / 0x6134786D), 48000 Hz, stereo, fltp, 288 kb/s (default)
  Metadata:
    creation_time  : 2024-06-29T15:24:25.000000Z
    handler_name   : SoundHandle
    vendor_id      : [0][0][0][0]
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> mjpeg (native))
Press [q] to stop, [?] for help
[swscaler @ 0x61517e15980] deprecated pixel format used, make sure you did set range correctly
Output #0, image2, to 'nocow1%04d.jpg':
  Metadata:
    major_brand     : mp42
    minor_version  : 0
    compatible_brands: isommp42
    con.android.version: 12
    location        : +52.2162+000.1176/
    location-eng    : +52.2162+000.1176/
    encoder         : Lavf58.76.100
Stream #0:0(eng): Vldeo: mjpeg, yuvj420p(pc, bt709, progressive), 1080x1920, q=2-31, 200 kb/s, 29.67 fps, 29.67 tbn (default)
  Metadata:
    encoder        : Lavc58.134.100 mjpeg
    creation_time  : 2024-06-29T15:24:25.000000Z
    handler_name   : VideoHandle
    vendor_id      : [0][0][0][0]
  Side data:
    cpb: bitrate max/min/avg: 0/0/200000 buffer size: 0 vbv_delay: N/A
    displaymatrix: rotation of -90.00 degrees
frame= 210 fps= 66 q=24.8 lsize=N/A time=00:00:07.07 bitrate=N/A speed=2.24x
video:17025kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
(tfenv) uxue@uxuelatitude7490:~/projects/EPIC/cows/nocows/v1photos$ ls
001.bmp      nocow10036.jpg  nocow10080.jpg  nocow10124.jpg  nocow10168.jpg
002.bmp      nocow10037.jpg  nocow10081.jpg  nocow10125.jpg  nocow10169.jpg
003.bmp      nocow10038.jpg  nocow10082.jpg  nocow10126.jpg  nocow10170.jpg
```

Although this extracted a lot of images, it was too many for one short video because it meant that the images looked mostly the same. So I found a way to adjust the framerate.

```
(tfenv) uxue@xuelatitude7490:~/projects/EPO/cows/NOCOWS/videophotos$ ffmpeg -i nocow1.mp4 -vf fps=3 nocow1%04d.jpg -hide_banner
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'nocow1.mp4':
  Metadata:
    major_brand : mp4
    minor_version : 0
    compatible_brands: isommp42
    creation_time : 2024-06-29T15:24:25.000000Z
    location : +52.2162+000.1176/
    location-eng : +52.2162+000.1176/
    com.android.version: 12
Duration: 00:00:07.09, start: 0.000000, bitrate: 20487 kb/s
Stream #0:0(eng): Video: h264 (High) (avc1 / 0x31367661), yuv420p(tv, bt709), 1920x1080, 19744 kb/s, 29.63 fps, 29.67 tbr, 90k tbn, 180k tbc (default)
  Metadata:
    rotate : 90
    creation_time : 2024-06-29T15:24:25.000000Z
    handler_name : VideoHandle
    vendor_id : [0][0][0][0]
  Side data:
    displaymatrix: rotation of -90.00 degrees
Stream #0:1(eng): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 288 kb/s (default)
  Metadata:
    creation_time : 2024-06-29T15:24:25.000000Z
    handler_name : SoundHandle
    vendor_id : [0][0][0][0]
Stream mapping:
  Stream #0:0 (h264 (native)) -> mjpeg (native)
Press [q] to stop, [?] for help
[swscaler @ 0x5c1522899940] deprecated pixel format used, make sure you did set range correctly
Output #0, image2, to 'nocow1%04d.jpg':
  Metadata:
    major_brand : mp4
    minor_version : 0
    compatible_brands: isommp42
    com.android.version: 12
    location : +52.2162+000.1176/
    location-eng : +52.2162+000.1176/
    encoder : Lavf58.76.100
Stream #0:0(eng): Video: mjpeg, yuvj420p(pc, bt709, progressive), 1080x1920, q=2-31, 200 kb/s, 3 fps, 3 tbn (default)
  Metadata:
    encoder : Lavc58.134.100 mjpeg
    creation_time : 2024-06-29T15:24:25.000000Z
    handler_name : VideoHandle
    vendor_id : [0][0][0][0]
  Side data:
    cpb: bitrate max/min/avg: 0/0/200000 buffer size: 0 vbv_delay: N/A
    displaymatrix: rotation of -0.00 degrees
frame= 21 f裴e=0.0 q裴24.8 l裴tze=N/A time=0:07.00 bitrate=N/A speed=7.34x
video:2693K audio:0K subtitle:0K other streams:0KB global headers:0KB muxing overhead: unknown
(tfenv) uxue@xuelatitude7490:~/projects/EPO/cows/NOCOWS/videophotos$ ls
nocow10001.jpg  nocow10003.jpg  nocow10005.jpg  nocow10007.jpg  nocow10009.jpg  nocow10011.jpg  nocow10013.jpg  nocow10015.jpg  nocow10017.jpg  nocow10019.jpg  nocow10021.jpg
nocow10002.jpg  nocow10004.jpg  nocow10006.jpg  nocow10008.jpg  nocow10010.jpg  nocow10012.jpg  nocow10014.jpg  nocow10016.jpg  nocow10018.jpg  nocow10020.jpg  nocow1.mp4
ffmpeg -i video.mp4 -vf fps=3 photoname%04d.jpg -hide_banner
```

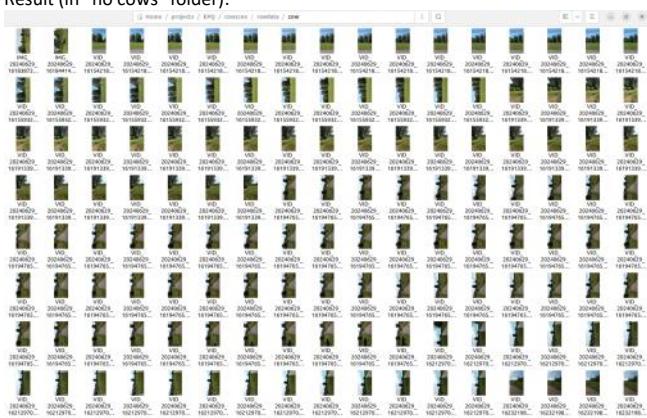
Where 3 is the framerate per second

(source: <https://stackoverflow.com/questions/45462731/using-ffmpeg-to-change-framerate>)

To do multiple videos at once, I made python file and then ran in terminal

```
1 import os
2 from glob import glob
3
4
5 for filename in glob("./*.mp4"): # enumerate?
6
7     cmd = f"ffmpeg -i {filename} -vf fps=3 {filename}_nocows%04d.jpg -hide_banner"
8     print(cmd)
9
10    os.system(cmd)
```

Result (in "no cows" folder):



Scraping

I tried using the command suggested here (<https://stackoverflow.com/questions/4602153/how-do-i-use-wget-to-download-all-images-into-a-single-folder-from-a-url>), but no images were saved.

```
(venv) tom@tom-XPS-15: ~ % wget -nd -r -P /save/location -A jpg,bmp,gif,png "https://www.google.com/search?client=ubuntu-snhs=f9f25705f4f01b7d&channel=fssxsrf=ADLYWIKhHLAIumbcZ2TX0s5KvU_L8Jvneg:1719658753065&q=dataset+of+english+countrieside+photos&fb=AfOa4sJWe7Rqy32pFwRj0UKwd8nb0Jfs8GB51Q006L3J_86uW0eqwdnVbya5F-x2j06tnu6LRjjgjArFfLolnByg2tMdxJFpG2a0dg6n0NSOUel_Zp-AnWJju5Bq_UoE1RqnxiqtwxQlddk0bEv4-fvXrL1Uxtmb7rYJIBkr79X849_zmlFz-2AGdJ47Zhwholsasa=X&ved=2ahUKEvlV3PG11CHAXJ2TEEAHxHceQtkLegjIEBA8Bblw-1858&bth=941&dpr=1"
The destination name is too long (454), reducing to 236
--2024-06-29 12:01:09-- https://www.google.com/search?client=ubuntu-snhs=f9f25705f4f01b7d&channel=fssxsrf=ADLYWIKhHLAIumbcZ2TX0s5KvU_L8Jvneg:1719658753065&q=dataset+of+english+countrieside+photos&fb=AfOa4sJWe7Rqy32pFwRj0UKwd8nb0Jfs8GB51Q006L3J_86uW0eqwdnVbya5F-x2j06tnu6LRjjgjArFfLolnByg2tMdxJFpG2a0dg6n0NSOUel_Zp-AnWJju5Bq_UoE1RqnxiqtwxQlddk0bEv4-fvXrL1Uxtmb7rYJIBkr79X849_zmlFz-2AGdJ47Zhwholsasa=X&ved=2ahUKEvlV3PG11CHAXJ2TEEAHxHceQtkLegjIEBA8Bblw-1858&bth=941&dpr=1&tbt=lsch [following]
Resolving www.google.com (www.google.com)... 216.58.264.88, 2400:1450:4089:827::2004
Connecting to www.google.com (www.google.com)|216.58.264.88|:2004... connected.
HTTP request sent, awaiting response... 200 OK
Location: https://www.google.com/search?client=ubuntu-snhs=f9f25705f4f01b7d&channel=fssxsrf=ADLYWIKhHLAIumbcZ2TX0s5KvU_L8Jvneg:1719658753065&q=dataset+of+english+countrieside+photos&fb=AfOa4sJWe7Rqy32pFwRj0UKwd8nb0Jfs8GB51Q006L3J_86uW0eqwdnVbya5F-x2j06tnu6LRjjgjArFfLolnByg2tMdxJFpG2a0dg6n0NSOUel_Zp-AnWJju5Bq_UoE1RqnxiqtwxQlddk0bEv4-fvXrL1Uxtmb7rYJIBkr79X849_zmlFz-2AGdJ47Zhwholsasa=X&ved=2ahUKEvlV3PG11CHAXJ2TEEAHxHceQtkLegjIEBA8Bblw-1858&bth=941&dpr=1&tbt=lsch
Reusing existing connection to www.google.com:443.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
/save/location: No such file or directory
/save/location/search?client=ubuntu-snhs=f9f25705f4f01b7d&channel=fssxsrf=ADLYWIKhHLAIumbcZ2TX0s5KvU_L8Jvneg:1719658753065&q=dataset+of+english+countrieside+photos&udn=2&fb=AfOa4sJWe7Rqy32pFwRj0UKwd8nb0Jfs8GB51Q006L3J_86uW0eqwdnVbya5F-tmp: Bad file descriptor
Cannot write to '/save/location/search?client=ubuntu-snhs=f9f25705f4f01b7d&channel=fssxsrf=ADLYWIKhHLAIumbcZ2TX0s5KvU_L8Jvneg:1719658753065&q=dataset+of+english+countrieside+photos&udn=2&fb=AfOa4sJWe7Rqy32pFwRj0UKwd8nb0Jfs8GB51Q006L3J_86uW0eqwdnVbya5F-tmp' (Success).
```

Since it didn't work with Google Images, I tried searching for a scraping algorithm with Bing. There is a handy library that I can use in Python which reduces the code to almost one line. (Source: <https://pypi.org/project/bing-image-downloader/>)

```
1 from bing_image_downloader import downloader
2
3 downloader.download('english fields', limit=10, output_dir='dataset', adult_filter_off=True, force_replace=False,
4 timeout=60, verbose=True)
5 # substitute 'english fields' with any term that you want to search for on Bing Images
```

The execution:

```
(searchenv) uxue@UXUELATITUDE7490:~/projects/EPQ/scraping$ python3 search.py
[!] Downloading Images to /home/uxue/projects/EPQ/scraping/dataset/english_fields.

[!] Indexing page: 1
[!] Indexed 10 Images on Page 1.

=====
[!] Downloading Image #1 from https://cdn.wallpapersafari.com/98/35/evz6Ru.jpg
[!] File Downloaded !

[!] Downloading Image #2 from https://web-cdn.wlngly.io/image-albums/qLUHfQv7P6yxCKGKOcdbjtpFSBj4KEph.jpg
[!] File Downloaded !

[!] Downloading Image #3 from http://l.rgbimg.com/cache1r4tWN/users/n/micromoth/680/na8rB1u.jpg
[!] File Downloaded !

[!] Downloading Image #4 from https://joolnn.com/images/english-countryside.jpg
[!] File Downloaded !

[!] Downloading Image #5 from https://cdn9.dissolve.com/p/D1230_19_657/D1230_19_657_1200.jpg
[!] File Downloaded !

[!] Downloading Image #6 from http://i.dailymail.co.uk/l/pix/2016/05/01/33093C0000000578-0-image-a-12_1462407427920.jpg
[Error] Invalid image, not saving http://i.dailymail.co.uk/l/pix/2016/05/01/33093C0000000578-0-image-a-12_1462407427920.jpg

[!] Issue getting: http://i.dailymail.co.uk/l/pix/2016/05/01/33093C0000000578-0-image-a-12_1462407427920.jpg
[!] Error: Invalid image, not saving http://i.dailymail.co.uk/l/pix/2016/05/01/33093C0000000578-0-image-a-12_1462407427920.jpg

[!] Downloading Image #6 from https://w.wallhaven.cc/full/qd/wallhaven-qdvsd5.jpg
[!] Issue getting: https://w.wallhaven.cc/full/qd/wallhaven-qdvsd5.jpg
[!] Error: HTTP Error 521:
[!] Downloading Image #6 from http://www.strangehistory.net/blog/wp-content/uploads/2016/01/english-fields.jpg
[Error] Invalid image, not saving http://www.strangehistory.net/blog/wp-content/uploads/2016/01/english-fields.jpg

[!] Downloading Image #6 from https://thumbs.dreamstime.com/b/english-countryside-fields-meadows-green-wheat-separated-hedge-lines-summer-cloudy-day-58757547.jpg
[!] File Downloaded !

[!] Downloading Image #7 from https://www.shutterstock.com/image-photo/english-fields-260nw-624693068.jpg
[Error] Invalid image, not saving https://www.shutterstock.com/image-photo/english-fields-260nw-624693068.jpg

[!] Issue getting: https://www.shutterstock.com/image-photo/english-fields-260nw-624693068.jpg
[!] Error: Invalid image, not saving https://www.shutterstock.com/image-photo/english-fields-260nw-624693068.jpg

[!] Indexing page: 2
[!] Indexed 35 Images on Page 2.

=====
[!] Downloading Image #7 from https://www.wallpaperbetter.com/wallpaper/126/656/846/countryside-england-britain-fields-road-grass-wind-1080P-wallpaper.jpg
[!] File Downloaded !

[!] Downloading Image #8 from https://c8.alamy.com/comp/BR14BY/typical-english-country-side-small-fields-hedges-and-woodland-on-the-BR14BY.jpg
[!] File Downloaded !

[!] Downloading Image #9 from https://i.pinimg.com/originals/13/54/c0/1354c03284ed745c4d8db1455419fde4.jpg
[!] File Downloaded !

[!] Downloading Image #10 from https://l1ve.staticflickr.com/5561/14632396138_c70a3e01e6_b.jpg
[!] File Downloaded !

[!] Done. Downloaded 10 Images.
(searchenv) uxue@UXUELATITUDE7490:~/projects/EPQ/scraping$ ls
(searchenv) uxue@UXUELATITUDE7490:~/projects/EPQ/scraping$ search.py
(searchenv) uxue@UXUELATITUDE7490:~/projects/EPQ/scraping$ ls dataset
'english_fields'
(searchenv) uxue@UXUELATITUDE7490:~/projects/EPQ/scraping$ ls dataset/'english fields'
Image_1.jpg Image_2.jpg Image_4.jpg Image_6.jpg Image_8.jpg
Image_3.jpg Image_5.jpg Image_7.jpg Image_9.jpg
(searchenv) uxue@UXUELATITUDE7490:~/projects/EPQ/scraping$
```

The result: a folder with images of 'english fields'



After doing this with a variety of different search terms, I have some folders each with some images in. Here is the COWS folder:



Here is the NEUTRAL folder (no cows):



This is required more thinking of what might be mistaken by a computer for a cow, or a feature of an image with a cow, such as empty fields, dogs, ducks, rivers, people, etc.

Putting all of the images into one large folder

So because all of the images are categorised into different folders, I am copying them into one big one to train.

```
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn$ cp cowsdata2/cows_dataset/NEUTRAL/*.*.jpg cowsdata2.1/nocows
cp: not executing 'cp' on 'cowsdata2.1/nocows/Image_100.jpg' with 'cowsdata2/cows_dataset/NEUTRAL/devon_cow_ntryside_not_cows/Image_100.jpg'
cp: will not overwrite just-created 'cowsdata2.1/nocows/Image_101.jpg' with 'cowsdata2/cows_dataset/NEUTRAL/devon_cow_ntryside_not_cows/Image_101.jpg'
cp: will not overwrite just-created 'cowsdata2.1/nocows/Image_102.jpg' with 'cowsdata2/cows_dataset/NEUTRAL/devon_cow_ntryside_not_cows/Image_102.jpg'
cp: will not overwrite just-created 'cowsdata2.1/nocows/Image_103.jpg' with 'cowsdata2/cows_dataset/NEUTRAL/devon_cow_ntryside_not_cows/Image_103.jpg'
cp: will not overwrite just-created 'cowsdata2.1/nocows/Image_104.jpg' with 'cowsdata2/cows_dataset/NEUTRAL/devon_cow_ntryside_not_cows/Image_104.jpg'
cp: will not overwrite just-created 'cowsdata2.1/nocows/Image_105.jpg' with 'cowsdata2/cows_dataset/NEUTRAL/devon_cow_ntryside_not_cows/Image_105.jpg'
cp: will not overwrite just-created 'cowsdata2.1/nocows/Image_106.jpg' with 'cowsdata2/cows_dataset/NEUTRAL/devon_cow_ntryside_not_cows/Image_106.jpg'
cp: will not overwrite just-created 'cowsdata2.1/nocows/Image_107.jpg' with 'cowsdata2/cows_dataset/NEUTRAL/devon_cow_ntryside_not_cows/Image_107.jpg'
... still not overwritten
```

BUT for some reason there are only 400 copied from 2500.

```
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn$ ls cowsdata2.1/nocows/ | wc -l
467
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn$ ls cowsdata2/cows_dataset/NEUTRAL/*.*.jpg | wc -l
2506
```

Since the images in most folders are named Image_1, Image_2 etc., if I try to copy them across like this, many will be deleted because they have the same name. So I am renaming all the images in each folder with a different number at the start of their name, so that every image is made sure to have a different name. Also making sure all are jpgs (not pngs) because I don't think my ML model can use pngs.

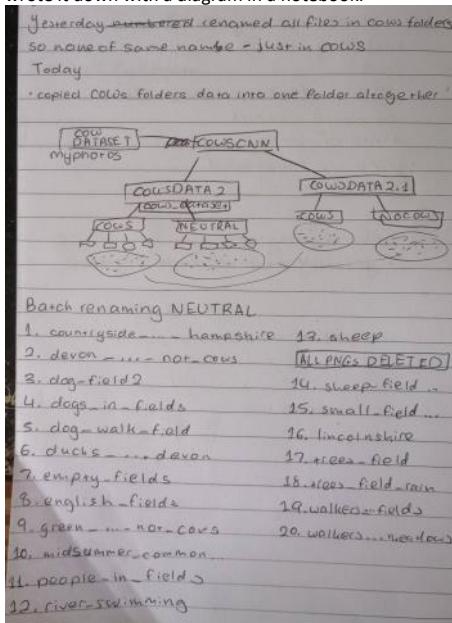
Batch renaming command:

<https://unix.stackexchange.com/questions/1136/batch-renaming-files>

```
for f in *.png; do
    mv -- "$f" "1$f";
done

(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn$ ls
countryside_walk_in_hampshire  dogs_in_fields  empty_fields  midsummer_common_cambridge_NOT_cow  sheep  stream_in_rural_lincs  walkers_english_fields
devon_countryside_not_cows  dogs_walk_field  english_fields  people_in_fields  sheep_field_not_cows  trees_field  walkers_english_meadows
dog_field  ducks_in_stream_devon  green_field_not_cow  river_swimming  small_field_in_rural_england  trees_field_rain
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn$ cd countryside_walk_in_hampshire/
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn$ for f in *.*; do mv -- "$f" "1$f"; done
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn$ ls
Image_100.jpg  Image_101.jpg  Image_102.jpg  Image_103.jpg  Image_104.jpg  Image_105.jpg  Image_106.jpg  Image_107.jpg  Image_108.jpg  Image_109.jpg  Image_110.jpg  Image_111.jpg  Image_112.jpg  Image_113.jpg  Image_114.jpg  Image_115.jpg  Image_116.jpg  Image_117.jpg  Image_118.jpg  Image_119.jpg  Image_120.jpg  Image_121.jpg  Image_122.jpg  Image_123.jpg  Image_124.jpg  Image_125.jpg  Image_126.jpg  Image_127.jpg  Image_128.jpg  Image_129.jpg  Image_130.jpg  Image_131.jpg  Image_132.jpg  Image_133.jpg  Image_134.jpg  Image_135.jpg  Image_136.jpg  Image_137.jpg  Image_138.jpg  Image_139.jpg  Image_140.jpg  Image_141.jpg  Image_142.jpg  Image_143.jpg  Image_144.jpg  Image_145.jpg  Image_146.jpg  Image_147.jpg  Image_148.jpg  Image_149.jpg  Image_150.jpg  Image_151.jpg  Image_152.jpg  Image_153.jpg  Image_154.jpg  Image_155.jpg  Image_156.jpg  Image_157.jpg  Image_158.jpg  Image_159.jpg  Image_160.jpg  Image_161.jpg  Image_162.jpg  Image_163.jpg  Image_164.jpg  Image_165.jpg  Image_166.jpg  Image_167.jpg  Image_168.jpg  Image_169.jpg  Image_170.jpg  Image_171.jpg  Image_172.jpg  Image_173.jpg  Image_174.jpg  Image_175.jpg  Image_176.jpg  Image_177.jpg  Image_178.jpg  Image_179.jpg  Image_180.jpg  Image_181.jpg  Image_182.jpg  Image_183.jpg  Image_184.jpg  Image_185.jpg  Image_186.jpg  Image_187.jpg  Image_188.jpg  Image_189.jpg  Image_190.jpg  Image_191.jpg  Image_192.jpg  Image_193.jpg  Image_194.jpg  Image_195.jpg  Image_196.jpg  Image_197.jpg  Image_198.jpg  Image_199.jpg  Image_200.jpg  Image_201.jpg  Image_202.jpg  Image_203.jpg  Image_204.jpg  Image_205.jpg  Image_206.jpg  Image_207.jpg  Image_208.jpg  Image_209.jpg  Image_210.jpg  Image_211.jpg  Image_212.jpg  Image_213.jpg  Image_214.jpg  Image_215.jpg  Image_216.jpg  Image_217.jpg  Image_218.jpg  Image_219.jpg  Image_220.jpg  Image_221.jpg  Image_222.jpg  Image_223.jpg  Image_224.jpg  Image_225.jpg  Image_226.jpg  Image_227.jpg  Image_228.jpg  Image_229.jpg  Image_230.jpg  Image_231.jpg  Image_232.jpg  Image_233.jpg  Image_234.jpg  Image_235.jpg  Image_236.jpg  Image_237.jpg  Image_238.jpg  Image_239.jpg  Image_240.jpg  Image_241.jpg  Image_242.jpg  Image_243.jpg  Image_244.jpg  Image_245.jpg  Image_246.jpg  Image_247.jpg  Image_248.jpg  Image_249.jpg  Image_250.jpg  Image_251.jpg  Image_252.jpg  Image_253.jpg  Image_254.jpg  Image_255.jpg  Image_256.jpg  Image_257.jpg  Image_258.jpg  Image_259.jpg  Image_260.jpg  Image_261.jpg  Image_262.jpg  Image_263.jpg  Image_264.jpg  Image_265.jpg  Image_266.jpg  Image_267.jpg  Image_268.jpg  Image_269.jpg  Image_270.jpg  Image_271.jpg  Image_272.jpg  Image_273.jpg  Image_274.jpg  Image_275.jpg  Image_276.jpg  Image_277.jpg  Image_278.jpg  Image_279.jpg  Image_280.jpg  Image_281.jpg  Image_282.jpg  Image_283.jpg  Image_284.jpg  Image_285.jpg  Image_286.jpg  Image_287.jpg  Image_288.jpg  Image_289.jpg  Image_290.jpg  Image_291.jpg  Image_292.jpg  Image_293.jpg  Image_294.jpg  Image_295.jpg  Image_296.jpg  Image_297.jpg  Image_298.jpg  Image_299.jpg  Image_300.jpg  Image_301.jpg  Image_302.jpg  Image_303.jpg  Image_304.jpg  Image_305.jpg  Image_306.jpg  Image_307.jpg  Image_308.jpg  Image_309.jpg  Image_310.jpg  Image_311.jpg  Image_312.jpg  Image_313.jpg  Image_314.jpg  Image_315.jpg  Image_316.jpg  Image_317.jpg  Image_318.jpg  Image_319.jpg  Image_320.jpg  Image_321.jpg  Image_322.jpg  Image_323.jpg  Image_324.jpg  Image_325.jpg  Image_326.jpg  Image_327.jpg  Image_328.jpg  Image_329.jpg  Image_330.jpg  Image_331.jpg  Image_332.jpg  Image_333.jpg  Image_334.jpg  Image_335.jpg  Image_336.jpg  Image_337.jpg  Image_338.jpg  Image_339.jpg  Image_340.jpg  Image_341.jpg  Image_342.jpg  Image_343.jpg  Image_344.jpg  Image_345.jpg  Image_346.jpg  Image_347.jpg  Image_348.jpg  Image_349.jpg  Image_350.jpg  Image_351.jpg  Image_352.jpg  Image_353.jpg  Image_354.jpg  Image_355.jpg  Image_356.jpg  Image_357.jpg  Image_358.jpg  Image_359.jpg  Image_360.jpg  Image_361.jpg  Image_362.jpg  Image_363.jpg  Image_364.jpg  Image_365.jpg  Image_366.jpg  Image_367.jpg  Image_368.jpg  Image_369.jpg  Image_370.jpg  Image_371.jpg  Image_372.jpg  Image_373.jpg  Image_374.jpg  Image_375.jpg  Image_376.jpg  Image_377.jpg  Image_378.jpg  Image_379.jpg  Image_380.jpg  Image_381.jpg  Image_382.jpg  Image_383.jpg  Image_384.jpg  Image_385.jpg  Image_386.jpg  Image_387.jpg  Image_388.jpg  Image_389.jpg  Image_390.jpg  Image_391.jpg  Image_392.jpg  Image_393.jpg  Image_394.jpg  Image_395.jpg  Image_396.jpg  Image_397.jpg  Image_398.jpg  Image_399.jpg  Image_400.jpg
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn$ rm *.png
```

It wasn't easy to keep track of what I had and hadn't copied into the new directory (cowsdata2.1) so I wrote it down with a diagram in a notebook.



After finally copying them all across, I checked how many items were in each directory.

```
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn/cowsdata2.1$ ls cows | wc -l
3464
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowscnn/cowsdata2.1$ ls nocows | wc -l
2616
```

3464 in "cows" and 2616 in "nocows". They need to be the same size so I deleted about 800 from "cows".

And thus the dataset is ready to be used to train and test the neural network!

Greyscale



This is the original image – I want it to be greyscale.

I found this code: <https://stackoverflow.com/questions/9506841/using-pil-to-turn-a-rgb-image-into-a-pure-black-and-white-image>

Using the part of the code that is for greyscale (not black and white):

```
from PIL import Image

img = Image.open('to_convert.jpg')

r = img.convert('1')
r.save('converted_greyscale.jpg')
```



This is the result that I wanted.

To do the same for many images, I added a loop and made the whole folder greyscale.

```
1 import os
2 from glob import glob
3
4 from PIL import Image
5
6 for filename in glob(f"./cowsdata2/cows_dataset/COWS/odds/*.jpg"):
7
8     img = Image.open(filename)
9     r = img.convert('1')
10    beginning, end = os.path.split(filename)
11    r.save(f"./cows_greyscale/{end}")
```

Neural network – cows vs no cows

Saturday, July 06, 2024 9:23 AM

To split the data and train the model, I used very similar code to the prototype.

Script that creates directories to put training and testing data:

```
1 import os
2
3
4
5 to_create = [
6     './cow-V-nocow2',
7     './cow-V-nocow2/training',
8     './cow-V-nocow2/testing',
9     './cow-V-nocow2/training/cow',
10    './cow-V-nocow2/training/nocow',
11    './cow-V-nocow2/testing/cow',
12    './cow-V-nocow2/testing/nocow'
13 ]
14
15 for directory in to_create:
16     try:
17         os.mkdir(directory)
18         print(directory, 'created')
19     except:
20         print(directory, 'failed')
```

Script that trains and saves the model:

```
1 import os
2
3 import random
4 from shutil import copyfile
5
6 import tensorflow as tf
7 from tensorflow.keras.preprocessing.image import ImageDataGenerator
8 from keras.layers import Dropout
9
10
11
12 if True:
13     def split_data(SOURCE, TRAINING, TESTING, SPLIT_SIZE):
14         all_files = []
15
16         for file_name in os.listdir(SOURCE):
17             file_path = SOURCE + file_name
18
19             if os.path.getsize(file_path):
20                 all_files.append(file_name)
21             else:
22                 print('[] is zero length, so ignoring'.format(file_name))
23
24         n_files = len(all_files)
25         split_point = int(n_files * SPLIT_SIZE)
26
27         shuffled = random.sample(all_files, n_files)
28
29         train_set = shuffled[:split_point]
30         test_set = shuffled[split_point:]
31
32         for file_name in train_set:
33             copyfile(SOURCE + file_name, TRAINING + file_name)
34
35         for file_name in test_set:
36             copyfile(SOURCE + file_name, TESTING + file_name)
37
38
39 COW_SOURCE_DIR = './cowsdata2.1/cows/'
40 TRAINING_COW_DIR = './cow-V-nocow2/training/cow/'
41 TESTING_COW_DIR = './cow-V-nocow2/testing/cow/'
42 NOCOW_SOURCE_DIR = './cowsdata2.1/nocows/'
43 TRAINING_NOCOW_DIR = './cow-V-nocow2/training/nocow/'
44 TESTING_NOCOW_DIR = './cow-V-nocow2/testing/nocow/'
45
46     split_size = .9
47     print(COW_SOURCE_DIR)
48     split_data(COW_SOURCE_DIR, TRAINING_COW_DIR, TESTING_COW_DIR, split_size)
49     split_data(NOCOW_SOURCE_DIR, TRAINING_NOCOW_DIR, TESTING_NOCOW_DIR, split_size)
50
51 model = tf.keras.models.Sequential([
52     tf.keras.layers.Conv2D(32, (3,3), input_shape=(150, 150, 3), activation='relu'),
```

```

53     tf.keras.layers.MaxPooling2D(2,2),
54 #     tf.keras.layers.Dropout(0.4),
55     tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
56     tf.keras.layers.MaxPooling2D(2,2),
57 #     tf.keras.layers.Dropout(0.4),
58     tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
59     tf.keras.layers.MaxPooling2D(2,2),
60     tf.keras.layers.Flatten(),
61 #     tf.keras.layers.Dropout(0.4),
62     tf.keras.layers.Dense(512, activation='relu'),
63     tf.keras.layers.Dense(128, activation='relu'),
64     tf.keras.layers.Dense(1, activation='sigmoid'),
65   ])
66
67 model.compile(optimizer=tf.keras.optimizers.RMSprop(learning_rate=0.001), loss='binary_crossentropy', metrics=['acc'])
68
69
70 TRAINING_DIR = './cow-v-nocow2/training/'
71 train_datagen = ImageDataGenerator(
72     rescale=1 / 255,
73     rotation_range=40,
74     width_shift_range=.2,
75     height_shift_range=.2,
76     shear_range=.2,
77     zoom_range=.2,
78     horizontal_flip=True,
79     fill_mode='nearest'
80   )
81 train_generator = train_datagen.flow_from_directory(
82     TRAINING_DIR,
83     batch_size=64,
84     class_mode='binary',
85     target_size=(150, 150)
86   )
87
88 VALIDATION_DIR = './cow-v-nocow2/testing/'
89 validation_datagen = ImageDataGenerator(
90     rescale=1 / 255,
91     rotation_range=40,
92     width_shift_range=.2,
93     height_shift_range=.2,
94     shear_range=.2,
95     zoom_range=.2,
96     horizontal_flip=True,
97     fill_mode='nearest'
98   )
99 validation_generator = validation_datagen.flow_from_directory(
100    VALIDATION_DIR,
101    batch_size=64,
102    class_mode='binary',
103    target_size=(150, 150)
104  )
105
106
108 import warnings
109 from PIL import Image as pil_image
110
111 history = model.fit(train_generator,
112     epochs=15,
113     verbose=1,
114     validation_data=validation_generator)
115
116 model.save('./cnn.keras')

```

Creating the directories

```

(venv2) (base) asier@z390:~/projects/EPQ$ python pre-cnn.py
./cow-v-nocow created
./cow-v-nocow/training created
./cow-v-nocow/testing created
./cow-v-nocow/training/cow created
./cow-v-nocow/training/nocow created
./cow-v-nocow/testing/cow created
./cow-v-nocow/testing/nocow created
(venv2) (base) asier@z390:~/projects/EPQ$ ls
cats-v-dogs      catsvdogs.py    cowdataset      cuda-keyring_1.1-1_all.deb  splitdata.py    venv
catsvdogs2.py    cnn4.keras     cows.py        PetImages                  splitdata.py~  venv2
catsvdogs2.py~   cnn5.keras     cows.py~       pre-cnn.py                tfenv
'#catsvdogs.py#'  cnn.keras     cow-v-nocow  pre-cnn.py~              uxue
(venv2) (base) asier@z390:~/projects/EPQ$ ls cow-v-nocow
testing  training

```

Inference after training the model for the first time (with a very small dataset):

```

cow
-----
8/8 ━━━━━━━━━━ 1s 109ms/step
0.02643171806167401
> /home/uxue/projects/EPQ/cowscnn/inference.py(36)<module>()
-> pass
(Pdb) c
nocow
-----
7/7 ━━━━━━━━━━ 1s 101ms/step
0.4619289340101523

```

This shows that:

3% of cow images were predicted to have no cows in, thus 97% were predicted to have cows.

46% of non-cow images were predicted to have no cows in, so 46% were predicted correctly.

These are quite bad scores with a clear bias towards cows, so I definitely need to improve this by

enlarging my dataset.

Inference script:

```

1 """
2 Inference for my model
3 """
4
5 import glob
6 import keras
7 import numpy as np
8
9 from keras.preprocessing import image
10
11 # load model
12
13 model = keras.models.load_model('./models/bestmodelyet.keras')
14
15 model.compile(loss='binary_crossentropy',
16                 optimizer='rmsprop',
17                 metrics=['accuracy'])
18
19 # inference
20
21 for item in ['cow', 'nocow']:
22
23     print(item)
24     print("."*10)
25     data = []
26
27     for idx, fname in enumerate(glob.glob(f'./camdata/{item}/*.JPG')): # go through all images in "camdata" folder
28
29         # prepare to use model on image
30         img = image.load_img(fname, target_size=(150,150))
31         x = image.img_to_array(img)
32         x = np.expand_dims(x, axis=0)
33
34         # make list with all images in cow or nocow folder
35         data.append(x)
36
37         # stop after 1000 images (but there are much less in this testing set)
38         if idx > 1000:
39             break
40
41     images = np.vstack(data) # make into a grid
42     classes = model.predict(images) # find probability of being cow
43     results = []
44
45     for c in classes:
46         result = int(c[0] < 0.5) # c[0] is just c but without the square brackets
47         results.append(result)
48
49     print(results)
50     print(f'images = {len(results)}')
51     print(f'how many times model detected cow (correctly and incorrectly) = {sum(results)}')
52     print(f'{sum(results)}/{len(results)}')
53     print('\n\n')

```

With new data from scraping + data from field, I did the inference again:

This time I printed the list of how the model has categorised the images in the testing set. Below "cow", the list should be full of 1s, which it mostly is, and thus the accuracy here is high (89%). Below "nocow", the list should be full of 0s, which it is less so. Nonetheless, since only 25% of the photos have been categorised incorrectly, the accuracy here is 75%. Finding a mean of these two gives an accuracy of 82%!

Greyscale

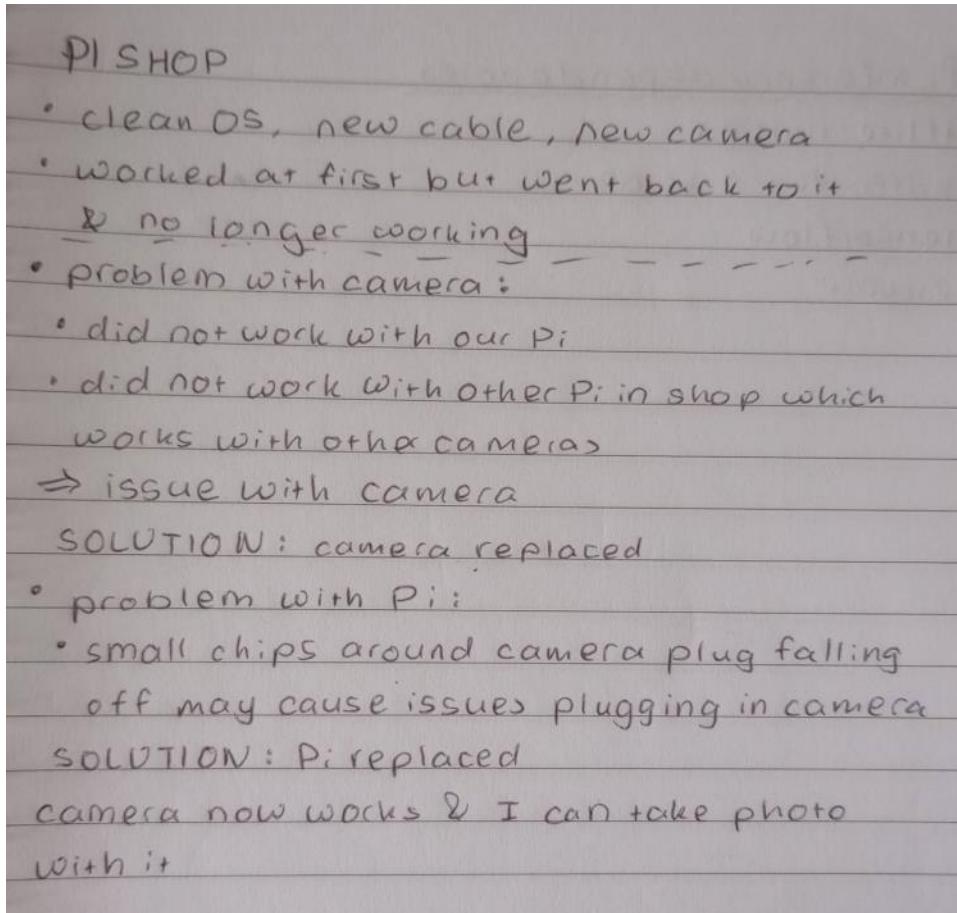
In Fred Manning's peer review, he said to try training the model with all of the images in greyscale (more details on the process of making the images greyscale in the "Making my own dataset" section). However, the validation accuracy and validation loss were worse in training than when using colour. So, I decided to use the model that has an average accuracy of 82%.

```
87/87 ━━━━━━━━ 120s 1s/step - acc: 0.5955 - loss: 1.2293 - val_acc: 0.7089 - val_loss: 0.5615
Epoch 2/15
87/87 ━━━━━━━━ 105s 1s/step - acc: 0.7090 - loss: 0.5499 - val_acc: 0.7418 - val_loss: 0.5353
Epoch 3/15
87/87 ━━━━━━━━ 104s 1s/step - acc: 0.7054 - loss: 0.5524 - val_acc: 0.7117 - val_loss: 0.5410
Epoch 4/15
87/87 ━━━━━━━━ 104s 1s/step - acc: 0.7147 - loss: 0.5319 - val_acc: 0.7343 - val_loss: 0.5113
Epoch 5/15
87/87 ━━━━━━━━ 104s 1s/step - acc: 0.7271 - loss: 0.5235 - val_acc: 0.7437 - val_loss: 0.5185
Epoch 6/15
87/87 ━━━━━━━━ 103s 1s/step - acc: 0.7167 - loss: 0.5260 - val_acc: 0.7446 - val_loss: 0.4987
Epoch 7/15
87/87 ━━━━━━━━ 103s 1s/step - acc: 0.7236 - loss: 0.5185 - val_acc: 0.7418 - val_loss: 0.5005
Epoch 8/15
87/87 ━━━━━━━━ 104s 1s/step - acc: 0.7292 - loss: 0.5101 - val_acc: 0.7258 - val_loss: 0.5058
Epoch 9/15
87/87 ━━━━━━━━ 103s 1s/step - acc: 0.7283 - loss: 0.5090 - val_acc: 0.7390 - val_loss: 0.5052
Epoch 10/15
87/87 ━━━━━━━━ 104s 1s/step - acc: 0.7412 - loss: 0.5037 - val_acc: 0.7465 - val_loss: 0.5002
Epoch 11/15
87/87 ━━━━━━━━ 103s 1s/step - acc: 0.7314 - loss: 0.5065 - val_acc: 0.7512 - val_loss: 0.4855
Epoch 12/15
87/87 ━━━━━━━━ 103s 1s/step - acc: 0.7359 - loss: 0.5061 - val_acc: 0.7352 - val_loss: 0.4876
Epoch 13/15
87/87 ━━━━━━━━ 103s 1s/step - acc: 0.7372 - loss: 0.4959 - val_acc: 0.7474 - val_loss: 0.4930
Epoch 14/15
87/87 ━━━━━━━━ 103s 1s/step - acc: 0.7410 - loss: 0.4990 - val_acc: 0.7455 - val_loss: 0.4849
Epoch 15/15
87/87 ━━━━━━━━ 104s 1s/step - acc: 0.7376 - loss: 0.5002 - val_acc: 0.7437 - val_loss: 0.5000
```

Model on Raspberry Pi

Sunday, August 18, 2024 4:08 PM

Getting Pi and camera working



I bought a new Raspberry Pi 5 with a working Pi camera. It worked at first but after leaving it aside few some time as I worked on other parts of the project, the camera no longer worked.

I took both the Pi and the camera to the Raspberry Pi shop in Cambridge. The people there tested the camera on a working Pi in the shop, and the camera still didn't work. Therefore the problem was the camera, and they replaced it.

Whilst looking at my Pi, they also noticed that the small chips around the camera plug were falling off, which isn't normal. This could cause problem plugging in the camera if the loose chips were lost. So they also replaced the Pi.

Converting model

Once I had overcome this hurdle, I wanted to put the model on the Raspberry Pi and try it. However, the model is too large for such a small computer so I needed to compress it into a TensorFlow Lite model.

```

1 import tensorflow as tf
2
3 import keras
4
5 model = keras.models.load_model('./bestmodelyet.keras')
6
7 # Export the keras model to a saved model format
8 model.export("saved_model")
9
10 # Convert the saved model to TensorFlow Lite
11 converter = tf.lite.TFLiteConverter.from_saved_model("saved_model")
12 tflite_model = converter.convert()
13
14 # Save the TensorFlow Lite model to a file
15 with open("modeltest.tflite", "wb") as f:
16     f.write(tflite_model)

```

I used this code: https://www.tensorflow.org/lite/models/convert/convert_models#python_api

TensorFlow Lite inference

To test the compressed model and work out how to load the TensorFlow Lite model, I made an inference script that uses the compressed model.

This code told me how to load the model:

<https://stackoverflow.com/questions/73724481/loading-tflite-model-for-inference-python>

```

1 """
2 Inference for tflite model
3 """
4
5 import glob
6 import numpy as np
7 import tensorflow as tf
8
9 from keras.preprocessing import image
10
11
12 interpreter = tf.lite.Interpreter('./model.tflite')
13 interpreter.allocate_tensors()
14
15
16 for item in ['cow', 'nocow']:
17     print(item)
18     print("-"*10)
19     data = []
20     for idx, fname in enumerate(glob.glob(f'./camdata/{item}/*.JPG')):
21         img = image.load_img(fname, target_size=(150,150))
22         x = image.img_to_array(img)
23         x = np.expand_dims(x, axis=0)
24         data.append(x)
25     interpreter.allocate_tensors()
26     input_details = interpreter.get_input_details()
27     output_details = interpreter.get_output_details()
28     interpreter.set_tensor(input_details[0]['index'], x)
29     interpreter.invoke()
30     output_data = interpreter.get_tensor(output_details[0]['index'])
31     output_index = np.argmax(output_data[0])
32     print(output_index < 0.5)

```

Inference on Pi

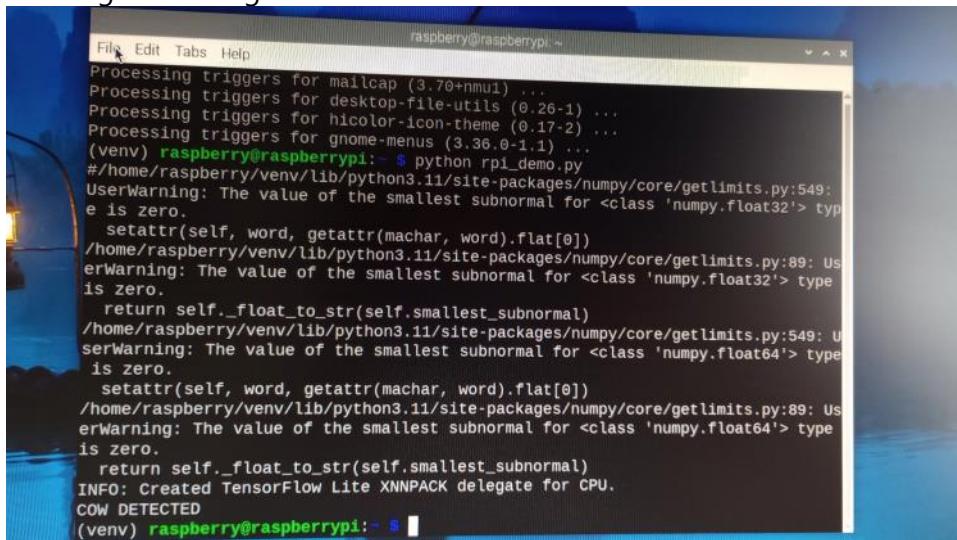
I used the code from the previous inference script to write the equivalent on the Raspberry Pi.

```

1 """
2 Demonstrate that the tflite model works on the Pi 5
3 Install libhdf5-dev for keras to work
4 sudo apt-get install libhdf5-dev
5 """
6
7 import tensorflow.lite_runtime.interpreter as tflite
8 import keras
9 import numpy as np
10
11 from keras.preprocessing import image
12
13
14 interpreter = tflite.Interpreter('./model.tflite')
15 interpreter.allocate_tensors()
16 FNAME = "cow.jpg" # test file
17
18 # load test file
19 img = image.load_img(FNAME, target_size=(150, 150))
20 x = image.img_to_array(img)
21 x = np.expand_dims(x, axis=0)
22
23 # load model
24 interpreter.allocate_tensors()
25 input_details = interpreter.get_input_details()
26 output_details = interpreter.get_output_details()
27 interpreter.set_tensor(input_details[0]['index'], x)
28 interpreter.invoke()
29 output_data = interpreter.get_tensor(output_details[0]['index'])
30 output_index = np.argmax(output_data[0])
31 result = output_data < 0.5
32
33 if result:
34     print('COW DETECTED')
35 else:
36     print('NO COW DETECTED')

```

Running the code gave me this result:



```

File Edit Tabs Help
raspberry@raspberrypi: ~
Processing triggers for mailcap (3.70+nmu1) ...
Processing triggers for desktop-file-utils (0.26-1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1.1) ...
(venv) raspberry@raspberrypi: ~ $ python rpi_demo.py
#home/raspberry/venv/lib/python3.11/site-packages/numpy/core/getlimits.py:549:
UserWarning: The value of the smallest subnormal for <class 'numpy.float32'> type
e is zero.
    setattr(self, word, getattr(machar, word).flat[0])
/home/raspberry/venv/lib/python3.11/site-packages/numpy/core/getlimits.py:89: Us
erWarning: The value of the smallest subnormal for <class 'numpy.float32'> type
is zero.
    return self._float_to_str(self.smallest_subnormal)
/home/raspberry/venv/lib/python3.11/site-packages/numpy/core/getlimits.py:549: U
serWarning: The value of the smallest subnormal for <class 'numpy.float64'> type
is zero.
    setattr(self, word, getattr(machar, word).flat[0])
/home/raspberry/venv/lib/python3.11/site-packages/numpy/core/getlimits.py:89: Us
erWarning: The value of the smallest subnormal for <class 'numpy.float64'> type
is zero.
    return self._float_to_str(self.smallest_subnormal)
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
COW DETECTED
(venv) raspberry@raspberrypi: ~

```

So it detected a cow correctly in this photo:



Live testing

Saturday, July 13, 2024 12:42 PM

Since the end result should be a device that takes a photo at set time intervals and then detects cows (or a lack thereof), I made a script that does this on my laptop.

To find out how to do this, I made tried to take a photo from the command line.

This source told me how: <https://askubuntu.com/questions/106770/take-a-picture-from-terminal>

This command works:

```
(tfenv) uxue@UxueLatitude7490:~/projects/EPQ/cowsCNN$ fswebcam -r 640x480 --jpeg 85 -D 1 web-cam-shot.jpg
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Delaying 1 seconds.
--- Capturing frame...
Captured frame in 0.00 seconds.
--- Processing captured image...
Setting output format to JPEG, quality 85
Writing JPEG image to 'web-cam-shot.jpg'.
```

So to make the script, I put it in a loop, and used the inference script to help me analyse the photo. I then stored the result, along with the time that the photo was taken, in a csv file.

```
1 import os
2 from datetime import datetime
3 import time
4 from colorama import Fore
5
6 import keras
7 import numpy as np
8 import pandas as pd
9
10 from keras.preprocessing import image
11
12
13 model = keras.models.load_model('./models/bestmodelyet.keras')
14
15 model.compile(loss='binary_crossentropy',
16                 optimizer='rmsprop',
17                 metrics=['accuracy'])
18
19 resultsdict = {'timestamp': [], 'cows': []}
20
21 for i in range(100):
22     timenow = datetime.now().isoformat()
23     resultsdict['timestamp'].append(timenow)
24     CMD = "fswebcam --device /dev/video0 -r 640x480 --jpeg 85 -D 1 photo.jpg"
25     os.system(CMD)
26     FNAME = "photo.jpg"
27
28     img = image.load_img(FNAME, target_size=(150,150))
29     x = image.img_to_array(img)
30     x = np.expand_dims(x, axis=0)
31
32     images = np.vstack([x])
33     classes = model.predict(images)
34     result = classes[0][0] < 0.5
35     resultsdict['cows'].append(result)
36
37     if result:
38         print(Fore.GREEN + 'COW DETECTED')
39     else:
40         print(Fore.RED + 'NO COW DETECTED')
41
42     df = pd.DataFrame(resultsdict)
43     df.to_csv('results.csv', index=False)
44
45     time.sleep(5)
```

Csv file output:

(where the sequence of numbers before the comma is the date and time, and the "true" or "false" is whether a cow was detected in that photo)

timestamp,cows

```
2024-07-13T16:44:45.351462,False
2024-07-13T16:44:57.297135,False
2024-07-13T16:45:09.198640,False
2024-07-13T16:45:21.077290,True
2024-07-13T16:45:32.977844,True
2024-07-13T16:45:44.854842,False
2024-07-13T16:45:56.735884,False
```

2024-07-13T16:46:08.633114,False
2024-07-13T16:46:20.535250,False
2024-07-13T16:46:32.429303,True
2024-07-13T16:46:44.309034,True
2024-07-13T16:46:56.183383,True
2024-07-13T16:47:08.057120,False
2024-07-13T16:47:19.939274,False
2024-07-13T16:47:31.822795,True
2024-07-13T16:47:43.709790,True
2024-07-13T16:47:55.609258,True
2024-07-13T16:48:07.485249,False
2024-07-13T16:48:19.389280,False
2024-07-13T16:48:31.284907,False
2024-07-13T16:48:43.145763,False
2024-07-13T16:48:55.017126,False
2024-07-13T16:49:06.881043,False
2024-07-13T16:49:18.759704,False
2024-07-13T16:49:30.640627,True
2024-07-13T16:49:42.521388,False
2024-07-13T16:49:54.397965,False
2024-07-13T16:50:06.303222,False
2024-07-13T16:50:18.178752,False
2024-07-13T16:50:30.068637,False

I tried this at home by putting photos of cows in front of the camera, which mostly worked. However, when I went into the field with my laptop (in the photo below), my laptop's camera quality was too poor, and the cows weren't close enough, for the program to work well enough.

Nonetheless, the device will be using the Raspberry Pi camera, which is better quality, so I'm going to focus on that side of the project next. I'm very happy with the outcome because the neural network is now finished!



Graph plotting

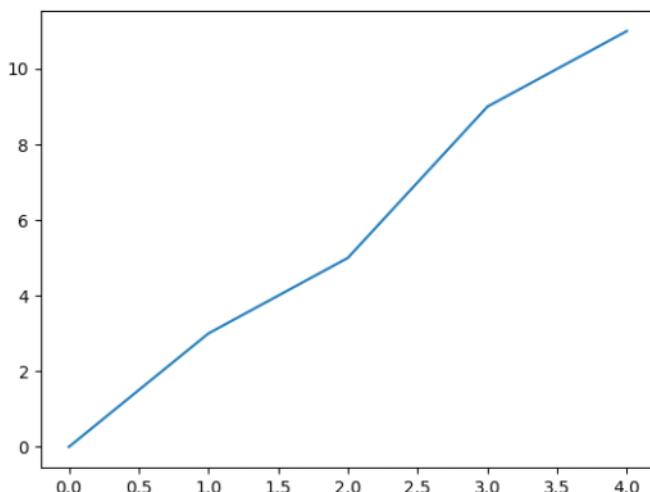
Sunday, July 14, 2024 5:39 PM

To do a rough analysis of the neural network's performance, I wanted to make a graph showing the loss and accuracy while training. Though I can plot graphs easily on Jupyter Notebook, the terminal can't display graphs, so I had to learn how to save a figure instead.

By researching (<https://www.atlassian.com/data/notebook/how-to-save-a-plot-to-a-file-using-matplotlib>) I made a script with a very simple graph to test if this works.

```
1 import matplotlib
2 import matplotlib.pyplot as plt
3
4 plt.plot([0, 1, 2, 3, 4], [0, 3, 5, 9, 11])
5
6 plt.savefig('./cowsnn/trash/graph.png')
```

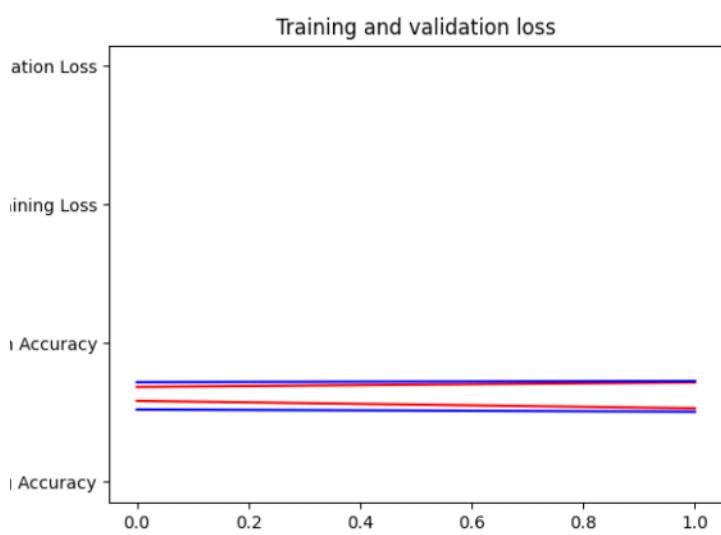
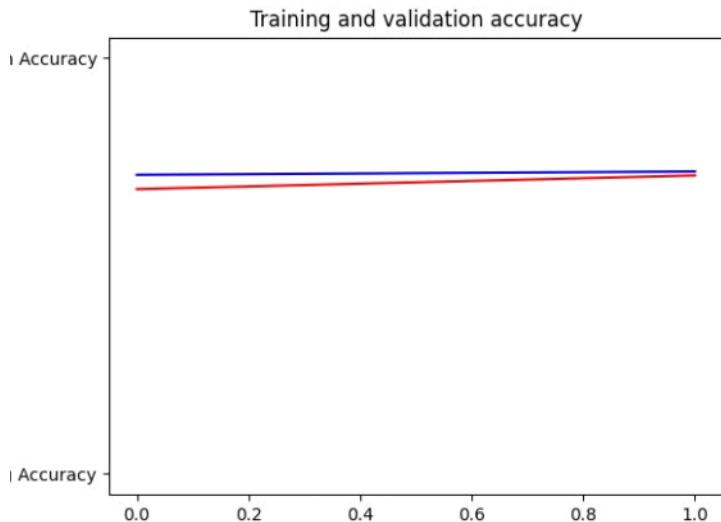
This code gives me this graph:



Now applying this to the model at the end of the training program (only for 2 epochs as opposed to 15, which would be very time consuming for a test)

(Using https://github.com/Osparsh2/cats-vs-dogs-coursera-assignment/blob/main/Cats_vs_Dogs_CourseraAssignment.ipynb, which shows how to plot graphs at the end)

```
135 # Plotting training and validation accuracy per epoch
136
137 plt.plot(epochs, acc, 'r', "Training Accuracy")
138 plt.plot(epochs, val_acc, 'b', "Validation Accuracy")
139 plt.title('Training and validation accuracy')
140 plt.figure()
141 plt.savefig('Training_and_validation_accuracy.jpg')
142
143
144 # Plotting training and validation loss per epoch
145
146 plt.plot(epochs, loss, 'r', "Training Loss")
147 plt.plot(epochs, val_loss, 'b', "Validation Loss")
148
149 plt.title('Training and validation loss')
150 plt.savefig('Training_and_validation_loss.png')
```

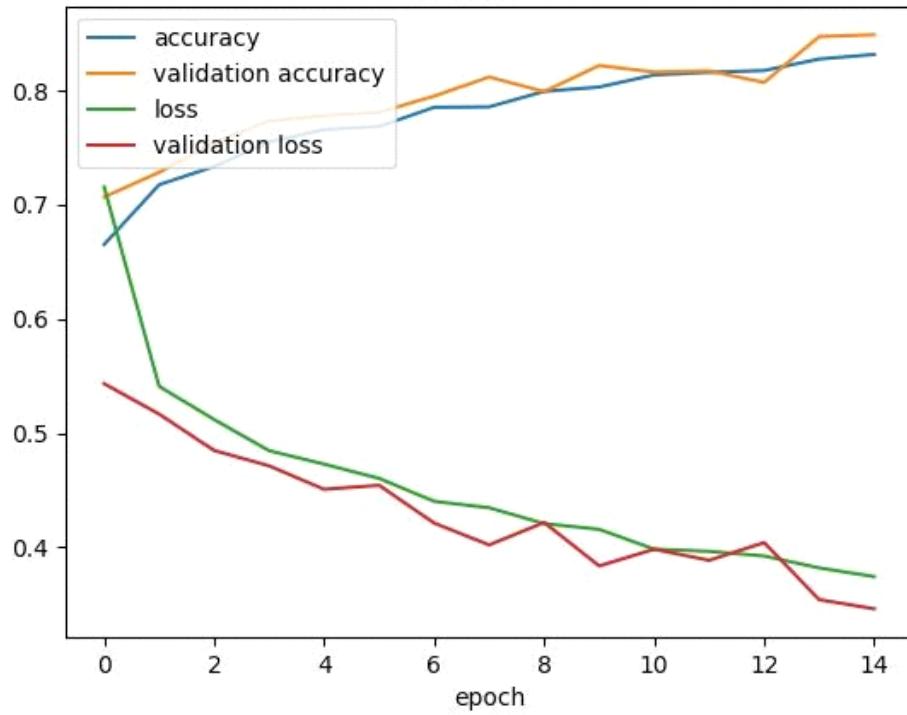


The y-axis labels above don't make sense and are cut off, so I tried some other code, using this code (from <https://stackoverflow.com/questions/41908379/keras-plot-training-validation-and-test-set-accuracy>)

```

132 epochs=range(len(acc)) # Get number of epochs
133 |
134 # plot both accuracy and loss graph
135 plt.plot(acc)
136 plt.plot(val_acc)
137 plt.plot(loss)
138 plt.plot(val_loss)
139 plt.title('model loss')
140 plt.ylabel('')
141 plt.xlabel('epoch')
142 plt.legend(['accuracy', 'validation accuracy', 'loss', 'validation loss'], loc='upper left')
143 plt.savefig('accuracy_and_loss_graph.jpg')

```



This graph shows both the loss and accuracy in one graph.
Since it worked first time I did it for all 15 epochs.

90/90	202s 2s/step - acc: 0.6173 - loss: 1.1245 - val_acc: 0.7067 - val_loss: 0.5433
Epoch 2/15	187s 2s/step - acc: 0.7189 - loss: 0.5496 - val_acc: 0.7283 - val_loss: 0.5166
Epoch 3/15	187s 2s/step - acc: 0.7209 - loss: 0.5214 - val_acc: 0.7543 - val_loss: 0.4848
Epoch 4/15	186s 2s/step - acc: 0.7486 - loss: 0.4978 - val_acc: 0.7732 - val_loss: 0.4713
Epoch 5/15	187s 2s/step - acc: 0.7563 - loss: 0.4932 - val_acc: 0.7780 - val_loss: 0.4508
Epoch 6/15	186s 2s/step - acc: 0.7652 - loss: 0.4662 - val_acc: 0.7808 - val_loss: 0.4543
Epoch 7/15	187s 2s/step - acc: 0.7798 - loss: 0.4434 - val_acc: 0.7951 - val_loss: 0.4212
Epoch 8/15	187s 2s/step - acc: 0.7885 - loss: 0.4294 - val_acc: 0.8117 - val_loss: 0.4021
Epoch 9/15	186s 2s/step - acc: 0.7995 - loss: 0.4200 - val_acc: 0.7990 - val_loss: 0.4217
Epoch 10/15	186s 2s/step - acc: 0.8027 - loss: 0.4192 - val_acc: 0.8218 - val_loss: 0.3837
Epoch 11/15	186s 2s/step - acc: 0.8119 - loss: 0.4013 - val_acc: 0.8161 - val_loss: 0.3983
Epoch 12/15	186s 2s/step - acc: 0.8193 - loss: 0.3859 - val_acc: 0.8171 - val_loss: 0.3886
Epoch 13/15	185s 2s/step - acc: 0.8068 - loss: 0.3998 - val_acc: 0.8070 - val_loss: 0.4041
Epoch 14/15	186s 2s/step - acc: 0.8230 - loss: 0.3830 - val_acc: 0.8472 - val_loss: 0.3542
Epoch 15/15	187s 2s/step - acc: 0.8389 - loss: 0.3660 - val_acc: 0.8488 - val_loss: 0.3462

(plotting this data)

Turbidity – csv file

Friday, August 16, 2024 6:43 PM

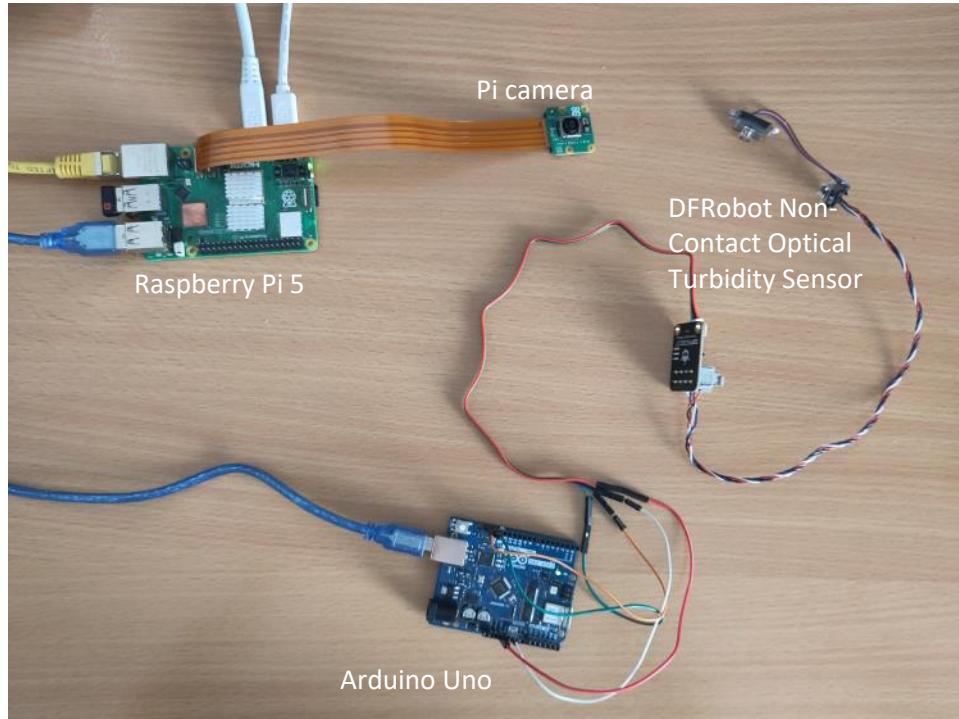
Hardware

DFRobot Non-Contact Optical Turbidity Sensor

Arduino Uno

Raspberry Pi& camera

Serial cable



Code for Arduino

This is connected to the turbidity sensor, which cannot function as a standalone but can store the data from the sensor.

From the instructions of the turbidity sensor [43] I found this code:

```

#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX
unsigned char str[5] = { }; //Serial receives data
unsigned char col;
unsigned int distance = 0;
unsigned char a[5] = {
    0x18,0x05, 0x00, 0x01 ,0x0D
};
void setup()
{
    Serial.begin(9600);
    mySerial.begin(9600);
}
void loop() {
    mySerial.write(a, 5);
    while (!mySerial.available());
    while (mySerial.available() > 0) //Detect if there is data on serial port
    {
        for (int i = 0; i < 5; i++)
        {
            str[i]=mySerial.read();
            delay(5);
        }
        Serial.println(str[3],DEC);
        mySerial.flush();
    }
    delay(500);
}

```

I changed the code slightly so that it would fit my purpose more and was shorter:

(Note that Arduino uses C, of which Python is similar but closer to natural language. So it took some trial and error since I was unfamiliar with this programming language.)

```

1 int list[] = {0, 1, 0, 2, 3, 100, 110, 99, 103, 4, 1, 0};
2 void setup()
3 {
4     Serial.begin(9600);
5 }
6 void loop()
7 {
8     for (byte i = 0; i < 12; i = i + 1) {
9         Serial.println(list[i]);
10        delay(3000);
11    }
12 }

```

Raspberry Pi code

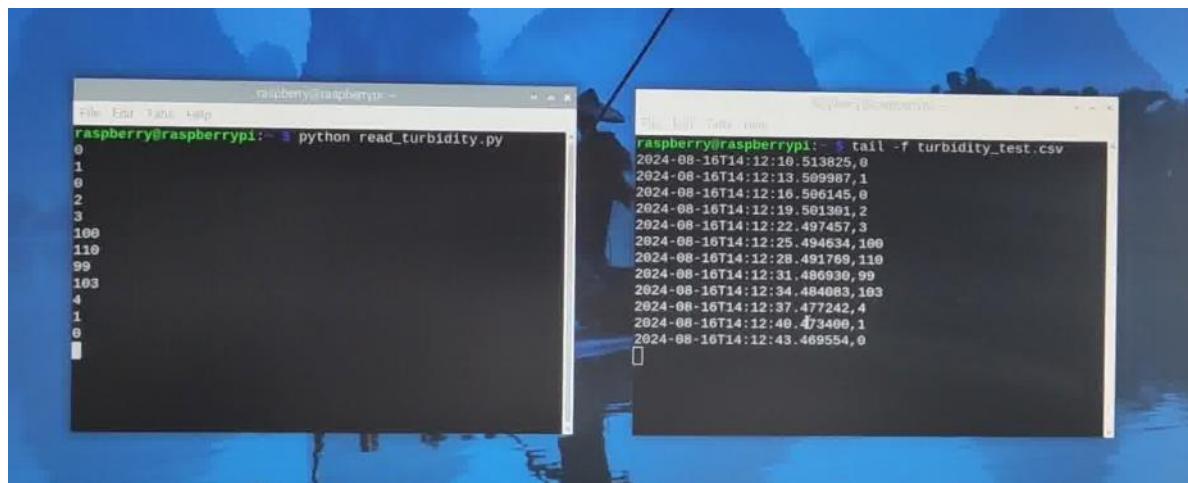
Using source [44]

```

1 """
2 Writes output of serial line + time to a csv file in real time
3 To be used for turbidity sensor for cows-v-no-cows project
4 """
5
6 import serial
7 import csv
8 from datetime import datetime
9
10 ser = serial.Serial('/dev/ttyACM0', 9600)
11
12 while True:
13     with open('turbidity_test.csv', 'a') as csvfile:
14         writer = csv.writer(csvfile)
15         data = ser.readline()
16         timenow = datetime.now()
17         decoded = data.decode("utf-8").strip()
18         print(decoded)
19         writer.writerow([timenow.isoformat(), decoded])
~~

```

The Raspberry Pi and the Arduino are connected by a cable (the serial line). As the docstring says, this code gets the data that was transferred across the serial cable and appends it into a csv file.



The screenshot shows two terminal windows side-by-side on a Raspberry Pi. The window on the left displays the command `python read_turbidity.py` followed by a series of numerical values: 0, 1, 0, 2, 3, 100, 110, 99, 103, 4, 1, 0. The window on the right displays the command `tail -f turbidity_test.csv` followed by the same sequence of values, each timestamped with the current date and time (e.g., 2024-08-16T14:12:10, 2024-08-16T14:12:13, etc.).

```
raspberry@raspberrypi: ~ python read_turbidity.py
0
1
0
2
3
100
110
99
103
4
1
0

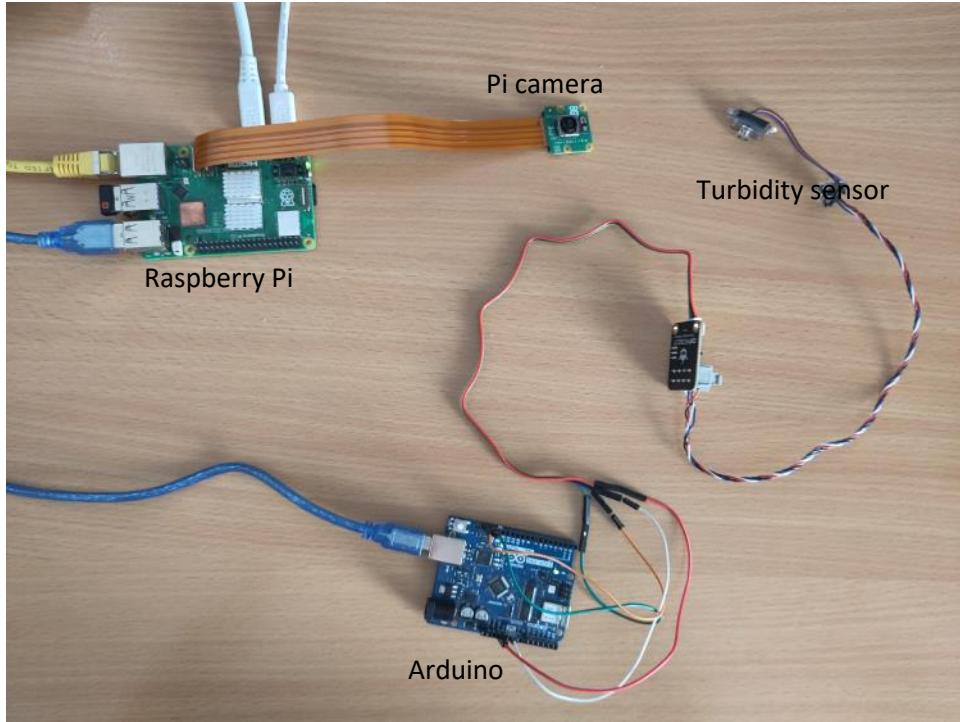
raspberry@raspberrypi: ~ tail -f turbidity_test.csv
2024-08-16T14:12:10,513825,0
2024-08-16T14:12:13,509987,1
2024-08-16T14:12:16,506145,0
2024-08-16T14:12:19,501301,2
2024-08-16T14:12:22,497457,3
2024-08-16T14:12:25,494634,100
2024-08-16T14:12:28,491789,110
2024-08-16T14:12:31,486930,99
2024-08-16T14:12:34,484083,103
2024-08-16T14:12:37,477242,4
2024-08-16T14:12:40,473400,1
2024-08-16T14:12:43,469554,0
```

This is a screenshot of the Raspberry Pi. The window on the left is the data being sent, and the window on the right is the data (with the time it was sent) being added to a csv file.

Final outcome

Sunday, August 18, 2024 10:48 AM

All of the evidence of my final outcome working as it should is assembled here.



Serial monitor (blue cable) printing turbidity on Raspberry Pi

Code on the Arduino:

```
1 int list[] = {0, 1, 0, 2, 3, 100, 110, 99, 103, 4, 1, 0};
2 void setup()
3 {
4 | Serial.begin(9600);
5 }
6 void loop()
7 {
8 for (byte i = 0; i < 12; i = i + 1) {
9 | Serial.println(list[i]);
10 | delay(3000);
11 }
12 }
```

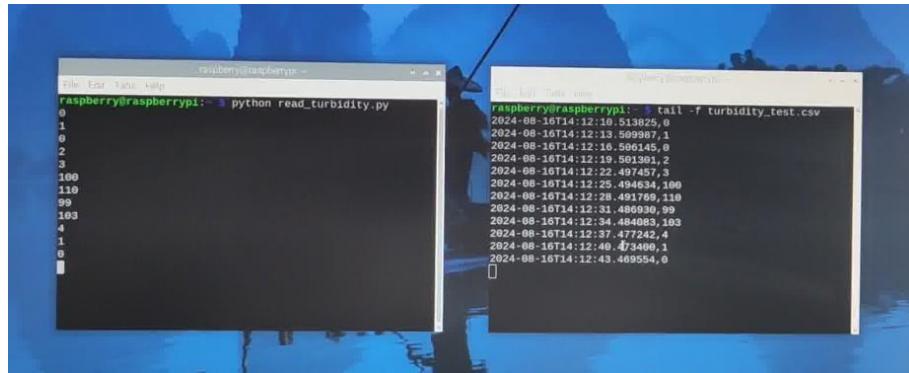
Code on the Raspberry Pi:

```

1 """
2 Writes output of serial line + time to a csv file in real time
3 To be used for turbidity sensor for cows-v-no-cows project
4 """
5
6 import serial
7 import csv
8 from datetime import datetime
9
10 ser = serial.Serial('/dev/ttyACM0', 9600)
11
12 while True:
13     with open('turbidity_test.csv', 'a') as csvfile:
14         writer = csv.writer(csvfile)
15         data = ser.readline()
16         timenow = datetime.now()
17         decoded = data.decode("utf-8").strip()
18         print(decoded)
19         writer.writerow([timenow.isoformat(), decoded])
```

```

Result:



The turbidity sensor reads the turbidity at set time intervals and sends it to the Arduino. Since the Arduino cannot function individually as a computer, this data is sent to the Raspberry Pi, via the serial monitor (blue cable). The photo is of the Raspberry Pi screen. The window on the left is the data being sent, and the window on the right is the data (with the time it was sent) being added to a csv file.

### Model predicting the presence of cows in a photo on the Raspberry Pi

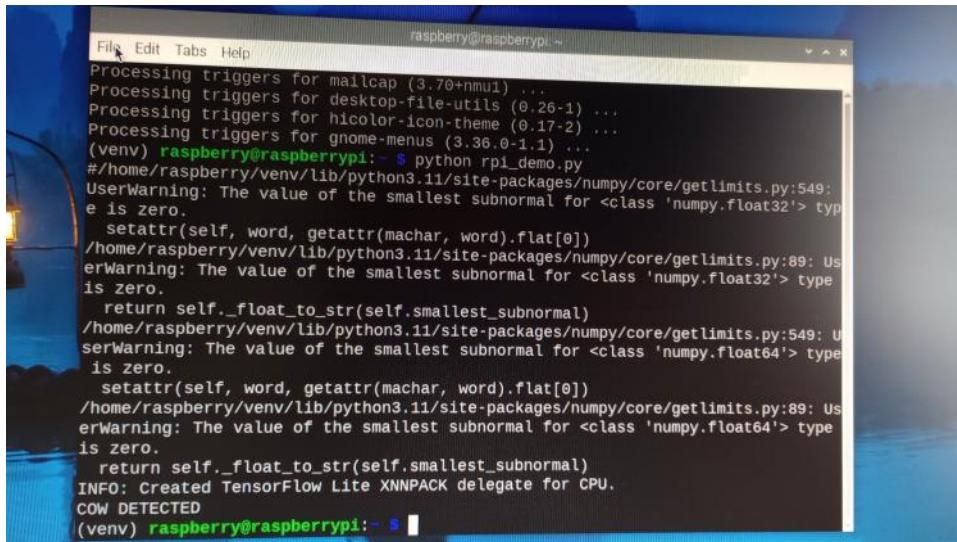
Code:

```

1 """
2 Demonstrate that the tflite model works on the Pi 5
3 Install libhdf5-dev for keras to work
4 sudo apt-get install libhdf5-dev
5 """
6
7 import tensorflow.lite_runtime.interpreter as tflite
8 import keras
9 import numpy as np
10
11 from keras.preprocessing import image
12
13
14 interpreter = tflite.Interpreter('./model.tflite')
15 interpreter.allocate_tensors()
16 FNAME = "cow.jpg" # test file
17
18 # load test file
19 img = image.load_img(FNAME, target_size=(150, 150))
20 x = image.img_to_array(img)
21 x = np.expand_dims(x, axis=0)
22
23 # load model
24 interpreter.allocate_tensors()
25 input_details = interpreter.get_input_details()
26 output_details = interpreter.get_output_details()
27 interpreter.set_tensor(input_details[0]['index'], x)
28 interpreter.invoke()
29 output_data = interpreter.get_tensor(output_details[0]['index'])
30 output_index = np.argmax(output_data[0])
31 result = output_data < 0.5
32
33 if result:
34 print('COW DETECTED')
35 else:
36 print('NO COW DETECTED')

```

### Result:



```

File Edit Tabs Help
raspberry@raspberrypi: ~
Processing triggers for mailcap (3.70+nmui) ...
Processing triggers for desktop-file-utils (0.26-1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1.1) ...
(venv) raspberry@raspberrypi: ~ $ python rpi_demo.py
#home/raspberry/venv/lib/python3.11/site-packages/numpy/core/getlimits.py:549:
UserWarning: The value of the smallest subnormal for <class 'numpy.float32'> type
e is zero.
 setattr(self, word, getattr(machar, word).flat[0])
/home/raspberry/venv/lib/python3.11/site-packages/numpy/core/getlimits.py:89: Us
erWarning: The value of the smallest subnormal for <class 'numpy.float32'> type
is zero.
 return self._float_to_str(self.smallest_subnormal)
#home/raspberry/venv/lib/python3.11/site-packages/numpy/core/getlimits.py:549: U
serWarning: The value of the smallest subnormal for <class 'numpy.float64'> type
is zero.
 setattr(self, word, getattr(machar, word).flat[0])
/home/raspberry/venv/lib/python3.11/site-packages/numpy/core/getlimits.py:89: Us
erWarning: The value of the smallest subnormal for <class 'numpy.float64'> type
is zero.
 return self._float_to_str(self.smallest_subnormal)
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
COW DETECTED
(venv) raspberry@raspberrypi: ~ $

```

The Raspberry Pi contains the compressed TensorFlow Lite model. In this program it correctly detects a cow in this photo:



It can then append this result to the same csv file.

## Accuracy of model

Testing the model on the computer shows us that the probability of classifying an image of cows correctly is **89%**, and the probability of classifying an image of no cows correctly is  $(1 - 0.254 = )$  **76%**. Taking the mean of these gives an overall accuracy of **82%**.

# November progress review

06 December 2023 22:02

This is a self review sheet: you will not get any progress review information from your EP mentor at this point.

## STUDENT COPY Preparation for Nov Progress Review

**Date:** 15/11/23

**Mentor:** Diane Michelson

|                                 |                  |
|---------------------------------|------------------|
| Student name:                   | Uxue Galvin      |
| Project area / topic:           | Machine Learning |
| Grade I am aiming for:          | A                |
| Attendance (see in Promonitor): | 100%             |
| Time spent on EP each week:     | 1h               |

### Assessment of your progress and paperwork

Place an X in the box to indicate your assessment of your paperwork quality - this aligns with the mark scheme bands.

|                                                          | Done | Limited | Competent | Good | Strong | In OneNote? |
|----------------------------------------------------------|------|---------|-----------|------|--------|-------------|
| Summer work – Audit                                      | Yes  |         |           | X    |        | Yes         |
| Evidence of generating ideas (e.g. mind map)             | Yes  | X       |           |      |        | Yes         |
| PPR Rationale, aims + goals identified.                  | No   |         |           |      |        | No          |
| PPR initial plans + factors to consider.                 | No   |         |           |      |        | No          |
| Project Breakdown                                        | Yes  | X       |           |      |        | Yes         |
| Timeline of tasks, planning till Sept 2024.              | Yes  | X       |           |      |        | Yes         |
| Diary, with detail, regular entries, short term plans.   | Yes  |         |           |      | X      | Yes         |
| Method of tracking sources used + evaluation of quality. | Yes  |         |           | X    |        | Yes         |
| Evidence of notes from sources researched.               | Yes  |         | X         |      |        | Yes         |

Anything else you have done?

Number of sources looked at so far?

If part of a group project how are roles divided?

Who are you working with?

Began to look at tutorials that will help me develop the skills I need for my project. I have looked at 2 sources so far

|                                                         |        |       |           |       |
|---------------------------------------------------------|--------|-------|-----------|-------|
| I use an IL period for EP each week                     | Always | Often | Sometimes | Never |
| I work on EP outside of my class + IL period each week. | Always | Often | Sometimes | Never |
| I update my diary every time I work on my project.      | Always | Often | Sometimes | Never |
| I record my sources every time I do some research       | Always | Often | Sometimes | Never |

I would judge my progress so far as (tick one):

|                          |   |
|--------------------------|---|
| Outstanding              |   |
| Making expected progress |   |
| Needs to improve         | X |

If you ticked "needs to improve" highlight which of the categories below you feel you need to improve. Say **how** you will do this

- Attendance
- Punctuality

Record the HOW here:

Allocate and stick to an hour of independent learning per week

Write and complete a to-do list of EPQ

|                                                                                                                                                                                                                    |                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Communication</li> <li>• Organisation</li> <li>• Preparation</li> <li>• Engagement in lessons</li> <li>• Subject understanding</li> <li>• Independent learning</li> </ul> | related activities for each week                                                                                     |
| What are your next steps?                                                                                                                                                                                          | Create timeline<br>Complete PPR<br>Do more research<br>Brainstorm project ideas<br>Continue machine learning project |

# February self review

01 May 2024 15:12

| Student name                                                                                                      | Uxue Galvin                                                                                             |                                       |           |           |             |                                 |                  |         |                      |
|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|---------------------------------------|-----------|-----------|-------------|---------------------------------|------------------|---------|----------------------|
| Project area / topic                                                                                              | Machine Learning Algorithm                                                                              |                                       |           |           |             |                                 |                  |         |                      |
| Grade I am aiming for                                                                                             | A                                                                                                       | Attendance                            |           | 100%      |             | Time spent on EP each week      | 1h               |         |                      |
| <b>Assessment of your progress and paperwork.</b> Place an X in the box to indicate assessment of your paperwork. |                                                                                                         |                                       |           |           |             |                                 |                  |         |                      |
|                                                                                                                   | Do I have this?                                                                                         | Quality of the document scheme bands) |           |           |             | (Note this aligns with the mark |                  |         | Uploaded to OneNote? |
|                                                                                                                   |                                                                                                         | Limited                               | Competent | Good      | Strong      |                                 |                  |         |                      |
| Evidence of generating ideas (e.g. mind map).                                                                     | Y                                                                                                       | X                                     |           |           |             |                                 |                  |         | Y                    |
| PPR Rationale, aims + goals identified.                                                                           | Y                                                                                                       |                                       |           |           |             |                                 | X                |         | Y                    |
| PPR initial plans + factors to consider.                                                                          | Y                                                                                                       |                                       |           |           |             |                                 | X                |         | Y                    |
| Timeline of tasks, planning till Sept 2024.                                                                       | Y                                                                                                       |                                       | X         |           |             |                                 |                  |         | Y                    |
| Diary, detail, regular entries, short term plans.                                                                 | Y                                                                                                       |                                       | X         |           |             |                                 |                  |         | Y                    |
| Method of tracking sources, evaluating quality.                                                                   | Y                                                                                                       |                                       | X         |           |             |                                 |                  |         | Y                    |
| Evidence of notes from sources researched.                                                                        | Y                                                                                                       |                                       |           |           |             |                                 | X                |         | Y                    |
| Primary research?                                                                                                 | N                                                                                                       |                                       |           |           |             |                                 |                  |         | N                    |
| Evidence of skill development?                                                                                    | Y                                                                                                       |                                       | X         |           |             |                                 |                  |         | Y                    |
| Evidence of how you have dealt with problems                                                                      | N                                                                                                       |                                       |           |           |             |                                 |                  |         | N                    |
| Evidence of reflection                                                                                            | Y                                                                                                       |                                       |           |           | X           |                                 |                  |         | Y                    |
| Range of sources used includes                                                                                    | Web X                                                                                                   | journal X                             | book X    | interview | newspaper X | podcast                         | Documentary / TV | YouTube | images               |
|                                                                                                                   | Other? (What are these?)<br>Course, webinar                                                             |                                       |           |           |             |                                 |                  |         |                      |
| What is your primary research going to be?                                                                        | Interview                                                                                               |                                       |           |           |             |                                 |                  |         |                      |
| How are you showing evidence of skill gain?                                                                       | "skills developed" section of diary, skills log, skills tracker, screenshots of how my project is going |                                       |           |           |             |                                 |                  |         |                      |

|                                                         |               |       |                  |       |
|---------------------------------------------------------|---------------|-------|------------------|-------|
| I use an IL period for EP each week                     | <b>Always</b> | Often | Sometimes        | Never |
| I work on EP outside of my class + IL period each week. | <b>Always</b> | Often | Sometimes        | Never |
| I update my diary every time I work on my project.      | Always        | Often | <b>Sometimes</b> | Never |
| I record my sources every time I do some research       | <b>Always</b> | Often | Sometimes        | Never |

I would judge my progress so far as (tick one):

|                          |   |
|--------------------------|---|
| Outstanding              |   |
| Making expected progress | X |
| Needs to improve         | X |

If you ticked "needs to improve" circle which of the categories below you feel you need to improve. Say how you will do this

- Attendance
- Punctuality
- Communication
- **Organisation** - balance doing the project with doing the

- Your mentor will discuss your progress with you in 1-2-1 sessions and will assess your progress on what is present in your OneNote. Note this information and your response in the PPR document.

What are your next steps?  
Make a decision on what my machine learning algorithm will do, collect data, run the programme, and evaluate its performance.  
Get OneNote up to date.

## Spring peer review

01 May 2024 15:13

### **Summer Peer review instructions:**

You will see that this word document allows you to record your name, and that of your reviewer. The owner of the folder can then pre-fill in the yellow sections indicating where they think they have shown evidence of particular skills. As this is later in the process it focusses on the skill development and evaluation A03 and A04 sections of the mark scheme.

If you are using this document to peer review a classmate's EP folder, add your name and feedback in the green sections. Often it is a good idea to look for what is missing as well as what is present. If you can circle sections of the mark scheme as well as give general feedback that will be helpful.

Finally, once you have got your folder peer review back, make sure you enter some notes and actions for yourself. If you can demonstrate taking and acting on advice it helps to show active project management.

## Peer Marking exercise

Date

Your name: Uxue Galvin

Reviewer's name: Mabel

Task: open up all your documents so that your reviewer can assess them.

Task: use the mark scheme to identify areas where there is no/poor/little evidence being provided. Offer suggestions on how to improve the evidence being provided to better show how the project is being managed.

## Marking Criteria for AO3: Develop and Realise

| AO3                      | Band 1 (1–6)                                                                                                                                                                                                      | Band 2 (7–12)                                                                                                                                                                                             | Band 3 (13–18)                                                                                                                                                                                                                                        | Band 4 (19–24)                                                                                                                                                                                                                    |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Skill selection</b>   | Range of skills identified is superficial and have limited relevance to the project in range.                                                                                                                     | Range of skills identified is adequate and have some relevance to the project.                                                                                                                            | Range of skills identified is good and have appropriate relevance to the project.                                                                                                                                                                     | Range of skills identified is excellent and are fully relevant to the project.                                                                                                                                                    |
| <b>Skill development</b> | Evidence of limited skill development demonstrated throughout the project. Limited use of skills and/or technologies to solve problems and limited critical decision making in order to achieve planned outcomes. | Evidence of some skill development demonstrated throughout the project. Some use of skills and/or technologies to solve problems, and some critical decision making in order to achieve planned outcomes. | Evidence of good skill development demonstrated throughout the project. Good use of skills and/or technologies to solve problems and good critical decision making in order to achieve planned outcomes although one or two opportunities are missed. | Evidence of excellent skill development demonstrated throughout the project. Sophisticated use of skills and/or technologies to solve problems and excellent critical decision making in order to fully achieve planned outcomes. |
| <b>Skill application</b> | Limited application of skills, with little or no demonstration of initiative, creativity or enterprise in order to achieve planned outcomes.                                                                      | Some application of skills, with some demonstration of initiative, creativity and/or enterprise in order to achieve planned outcomes.                                                                     | Good application of skills, with demonstration of initiative, creativity and enterprise in order to achieve planned outcomes although one or two opportunities are missed.                                                                            | Excellent application of skills, with full demonstration of initiative, creativity and enterprise in order to achieve planned outcomes.                                                                                           |
| <b>Project outcomes</b>  | Superficial exploration of the theme/topic results in a project with limited planned outcomes that meet only a few of the aims and objectives.                                                                    | Adequate exploration of the theme/topic results in a project with adequate planned outcomes that meet some of the aims and objectives.                                                                    | Good exploration of the theme/topic results in a project with appropriate planned outcomes that meet the aims and objectives although one or two opportunities are missed.                                                                            | Excellent exploration of the theme/topic results in a project with well-considered planned outcomes that fully meet the aims and objectives.                                                                                      |
|                          | 1 2 3 4 5 6                                                                                                                                                                                                       | 7 8 9 10 11 12                                                                                                                                                                                            | 13 14 15 16 17 18                                                                                                                                                                                                                                     | 19 20 21 22 23 24                                                                                                                                                                                                                 |

|                                       |
|---------------------------------------|
| How are you demonstrating challenge?  |
| How are you demonstrating skill gain? |
| How are you demonstrating Initiative? |
| How are you demonstrating enterprise? |
| What can you not see evidence of?     |

0 marks – No response or no response worthy of credit

Assessment here:

Range of skills identified is good and are fully relevance to project.

Great evidence of skills development.

Good use of skills to solve problems and good critical decision making.

Good application of skills with demonstration of initiative, creativity and enterprise.

To do:  
 Complete skills log and update skills level table (in the skills November section of your AO3 folder).  
 Add more skill application evidence.  
 Add more project outcome evidence

Marking Criteria for AO4: Review

| AO4                                                   | Band 1 (1–3)                                                                                                                                           | Band 2 (4–6)                                                                                                                                                | Band 3 (7–9)                                                                                                                                                                   | Band 4 (10–12)                                                                                                                                                       |
|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Communicating project outcomes and conclusions</b> | Limited range and use of media and communication skills to present project outcomes and conclusions that superficially engage the audience.            | Adequate range and use of media and communication skills to present project outcomes and conclusions that engage the audience.                              | Good range and use of media and communication skills to present project outcomes and conclusions that mostly engage the audience although one or two opportunities are missed. | Excellent range and use of media and communication skills to present project outcomes and conclusions that fully engage the audience.                                |
| <b>Iterative review</b>                               | Evaluation of the project is superficial and has been used in a limited way to meet few aims and objectives of the project.                            | Some on-going evaluation of the project but is not always critical or reflective to adapt, develop and meet some of the aims and objectives of the project. | Good, informed on-going evaluation of the project. Critical and reflective evaluation has been used to adapt, develop and meet most of the aims and objectives of the project. | Sophisticated on-going evaluation of the project. Critical and extensive evaluation has been used to adapt, develop and meet the aims and objectives of the project. |
| <b>Evaluation of own learning and performance</b>     | Limited attempt to evaluate their own work, learning and performance. Limited identification of issues, areas for improvement and transferable skills. | Some evaluation of their own work, learning and performance. Some identification of issues, areas for improvement and transferable skills.                  | Good, informed evaluation of their own work, learning and performance. Good identification of issues, areas for improvement and transferable skills.                           | Sophisticated evaluation of their own work, learning and performance. Excellent identification of issues, areas for improvement and transferable skills.             |
|                                                       | 1      2      3                                                                                                                                        | 4      5      6                                                                                                                                             | 7      8      9                                                                                                                                                                | 10     11     12                                                                                                                                                     |

0 marks – No response or no response worthy of credit

Assessment here:

Good, informed evaluation of project – could do with some more.

To do:

Update PPR.

Complete February self review.

Response to assessment here: what will you change?

# Problems / solutions

01 May 2024 15:01

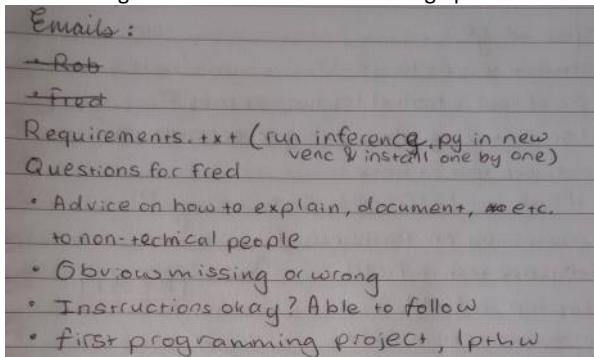
| Date     | Problem                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Solution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13/06/24 | The screen of my computer broke, and I couldn't find a replacement screen, so I had to get a new computer. The drive from my old computer was moved, but since I had 2 operating systems (Windows and Ubuntu) only the files etc. from Windows were kept. The files for EPQ which were not backed up (because they were the wrong format for OneDrive) were on Ubuntu, so most of these were lost.                                                                                                                             | Coincidentally, I put most of the important files on GitHub last week, so I was able to download them and recuperate most of what I'd lost. There might still be some missing that I didn't put on GitHub, but I will have to do without them.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 13/06/24 | Another problem due to my computer breaking: as I mentioned before, I had two operating systems on my old computer. This is because I find it easier to use Ubuntu for the coding side, and Windows for any Microsoft apps such as OneNote. The use of two operating systems on one computer is called "dual boot". On my new computer, I tried to set up dual boot but was unable to: when Ubuntu worked, Windows would show the blue screen of death; when Windows worked, it wouldn't even show the option of using Ubuntu. | It's difficult to find resources to fix problems with dual boot, because not many people use it, and Windows doesn't make it easy to set up or easy to diagnose the problem (they aren't very transparent). Trying to fix it is also time consuming, and I don't have much time left to finish the project. My work around is just using one operating system for the time being. I'm using Ubuntu, which doesn't have Microsoft Office, so I have had to use the web version. This works... most of the time.                                                                                                                                                                                                      |
| 23/04/24 | The training of the neural network takes over 2 hours on my laptop, which is inefficient since I want to do multiple iterations                                                                                                                                                                                                                                                                                                                                                                                                | I used my brother's gaming computer for training, which takes just over half an hour. I did this by ssh-ing into his computer and copying the training scripts and the data across (more on ssh-ing in the Cats vs Dogs algorithm section)                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 25/04/24 | The prototype neural network is biased towards categorising dogs                                                                                                                                                                                                                                                                                                                                                                                                                                                               | The dataset was faulty – I found this out because the number of images in each testing and training folder was definitely wrong (it's not possible to go through the photos in each because there are thousands). Deleting the training and testing folders, and running the script that splits data again, got rid of the bias and improved the accuracy.                                                                                                                                                                                                                                                                                                                                                          |
| 17/04/24 | The code on Jupyter Notebook for my prototype is giving me an error – I need to download PIL, but I have already downloaded PIL                                                                                                                                                                                                                                                                                                                                                                                                | I couldn't figure out the reason why this problem arose, but when I used a text editor and ran the code on the terminal instead, the code worked without an error message. Therefore, the solution is to run the code from the terminal from now on.                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 13/05/24 | There are no datasets of cows vs no cows on the Internet, and it's difficult to make one myself because I need thousands of images                                                                                                                                                                                                                                                                                                                                                                                             | I ended up getting images from a few different sources: taking photos myself, taking videos and extracting images from them (more efficient than taking photos but less varied), a dataset of cows that I found, and writing a script that downloads all the images from Bing image search (scraping).                                                                                                                                                                                                                                                                                                                                                                                                              |
| 26/05/24 | "Scraping" isn't downloading any images                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | This may have been a fault with the script itself, but also the fact that Google doesn't allow scraping because it messes with the search numbers. So I used Bing instead, which worked.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 15/08/24 | Converted pdf files from Jupyter Notebook don't upload                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | This was a surprisingly persistent problem that I couldn't work out for a while. Jupyter Notebook does have a way of exporting files as a pdf, but it doesn't work, so I exported it as HTML. Converting it on websites such as html2pdf.com sometimes didn't format it correctly, but when it did, the pdf wouldn't upload as a printout onto OneNote. Eventually I realised that I could open the HTML file on the browser, and print the page as a pdf. This would cut off some long lines of code, so I had to write the long lines in a different way which was shorter. Nonetheless doing so is good practice, and saving the pdf in this way meant it finally was able to upload as a printout onto OneNote. |
| 08/07/24 | Raspberry Pi camera isn't working                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | I went to the Pi shop, and the people there couldn't tell me for certain what the problem was, except that it was the camera, not the Pi. So I replaced the camera and now it is working.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 06/07/24 | Computer camera doesn't detect cows when I test the model with my laptop in the field                                                                                                                                                                                                                                                                                                                                                                                                                                          | My laptop camera is quite poor quality, and the cows actually ran away when I tried to get close to them. However, this was only a test of the model which I had already done at home. Also, the device would be using the Pi camera which is better quality, so this would solve the problem of bad quality photos.                                                                                                                                                                                                                                                                                                                                                                                                |

# Peer review: Fred Mannings

Thursday, July 11, 2024 8:14 PM

My dad's colleague, Fred Mannings, co-owns a machine learning start-up part-time (<https://predixus.com/>) so I thought he would be a credible source to ask for advice from.

I find writing emails difficult so wrote a rough plan before writing to him:



Here is some screenshots of our correspondence.

**Re: EPQ Cows Project**

From: UXUE <uxue.galvin@proton.me>  
To: Frederick Mannings

Dear Fred,

I'm sure you're really busy but hopefully this will be quick for you, if you don't mind helping me. I started to put my project on GitHub yesterday and it's a citizen science project that could be helpful for other people. Here is the link:

<https://github.com/adimena/Cows-vs-no-cows/tree/main>

I'm currently doing A-levels and wanted to get better at programming, so I learnt Python over the summer with Learn Python the Hard Way. This is my first programming project. One of the aspects of my project that I will be assessed on is other people peer-reviewing my work, so it would be great to get some feedback. This is of course quite a simple project and I've tried to make it easy to understand what I've done. I'd like to know if I've been successful in this. Could you think of anything obvious that is missing or wrong? Are the instructions easy to follow?

I really appreciate your advice and your time.

Best regards,

UXUE

From: UXUE <uxue.galvin@proton.me>  
To: Frederick Mannings <fred.mannings@predixus.com>

Good Morning @UXUE,

I've compiled my review below, and separated it into sections.

#### Project Coherence:

Overall, I have zero issues navigating through the project on Github and understanding the general logic. The Readme is well structured and contains the key info, and I can tell that you have made a concerted effort to separate out the concerns of your project into separate files.

However, I do have some suggestions / potential improvements:

1. Dataset Sizing - It's always good to tell the reader the size of dataset that you are working with. I appreciate that the project may not have concluded, but this is a good piece of information to include in the Readme in the final iteration. I have more comments on this point in the next section.
2. Project Structure - I see you have used python as a pure scripting language throughout this project. It is, after all, a scripting language. If you wish to abstract the core functionality of your project into a form that you can reuse later, I would suggest constructing simple functions from the logic, and where necessary using `if __name__ == "__main__"` constructs to run the files as scripts.
3. Comments - I am not an advocate of commenting every line of code. I don't actually believe this is useful to people. The code *should be* the documentation. That being said, readers, that are not as experienced as myself, may appreciate some light comments throughout the code. e.g, for [this line](#), something as simple as '# construct a binary classification CNN' would suffice
4. Typos - There also seems to be a typo in the README [here](#) and [here](#) where I believe 'pip' should be referenced instead of 'python'.

#### The Data:

As I am sure you know, "Garbage-in-Garbage-out" is *real* in machine learning. And so special considering is always needed when concerning the data.

For your use case, I can see a few potential gotchas that you might want to think about / make comments on in your final report:

- Each Cow / No-Cow data point may contain more contextual information than Cow information. In such cases, the model can often use the context as a proxy for the label.

The classic example that did the rounds a decade ago was when a model was trained to detect whether there are dogs, and it did great job! But it turned out that all the photos in the training set were of dogs, outside, on grass. What the model was actually doing was just detecting lots of green grass, rather than 'properly' extracting the notion of 'dog'.

So, the caution here, is to ensure your model is properly abstracting the *notion* of cow. From the 'Coming Soon' section of the readme, I believe you are on track to do this by investigating what is failing and what is performing well, and why.

You can also throw into the hold out set some red-herring images, to evaluate what happens when a model sees a horse, for example. This will give you an idea as to whether the model is abstracting the features of a cow, or just its surroundings. The level of the classification may also be of use in this scenario. A 0.8 inference level for a cow and a 0.55 inference level for a horse would both be classified as a cow (assuming 0.5 is the threshold), but clearly these two inferences are not the same.

The final thing to emphasise on that note, is: maybe all of that doesn't matter! Whether it does or not is dependent on the use case of the model. If images are fed to the model in a very controlled setting, where you know, that the image is either (definitely) a cow or not a cow, then who cares how it makes its judgement as long as it's correct? Of course, if the usage context of the model changes then special consideration on all of these points will need to be taken into account.

- How many data points are in your training, test and hold out sets? And how are you selecting them (do you shuffle, etc)? I note that you're drawing data from several sources (Kaggle, self-sourced, Bing) - is the data from each source spread over the datasets, or is it isolated in one or the other set? I can infer this from the code, but it's always nice to see these key points answered as a short statement in the README.

These things are important to think about because the biggest problem that I have seen in my professional career with ML is 'information bleeding'. This is the issue, where the model does not abstract the desired behaviour from the train set and instead overfits to data that is common in the train **and test set**. This will give the developer a false sense of performance, that won't be represented in the real world.

One way to mitigate this is to have 3<sup>rd</sup> hold out set - a *validation* set. The purpose of this dataset is to literally check that the model is doing what you want it to do, rather than checking that it does what it's doing well. And in your case, what you want is for the model to abstract the notion of cow, rather than overfitting to the kaggle or bing datasets. In this dataset you should store images that the model has never seen before, from a source, that the model has also never seen before. Only after going through the whole train and test verification phase do you test on the validation set. If you have enough data, I would put the images that you personally collected into the validation set, and train/test the model solely on the data from the other sources.

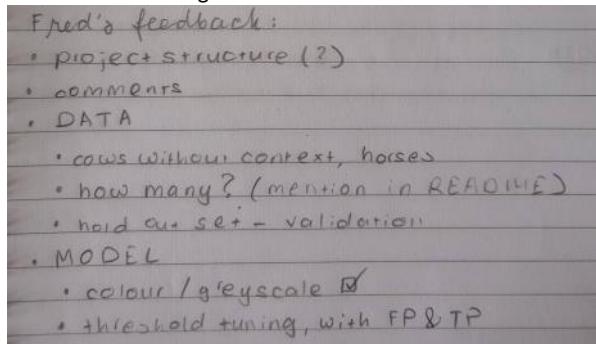
#### The Model

- I note by the shape of the model that you are factoring colour into the CNN (model input shape is 150x150x3 ?). Have you investigated what effect a single greyscale channel in place of colour has on the performance? Sometimes, especially in scenarios where colour patterns are not important to classification, this can improve many things. One major reason being, is that the model has 2 less dimensions to risk overfitting to, or even to hallucinate in.
- Threshold tuning - when evaluating the output of the model, 2 main things can be considered. The ability of the model to predict, and the position of the threshold to perform the classification. To inspect these two things independently, you can produce an ROC (Receiver Operating Characteristic) curve. In short, the ROC curve shows the trade-off between false positives and true positives *over* the classification threshold. Sometimes, extra performance can be squeezed out by tuning the classification threshold away from 0.5. See [this SKlearn tutorial](#) for more info.

Overall, I am impressed with the levels of performance that you are reaching with this model, and your ability to produce this having only started to learn Python recently.

Any questions you have to any of my comments, don't hesitate to reply to this email.

I wrote a list of things to act on after this email:



# What I would do differently

Saturday, August 31, 2024 6:35 PM

Though I'm proud of my outcome and my project, I acknowledge that parts of my project could have been better executed. It would have been difficult to know this at the beginning, when I had no experience of managing a project myself, nor the field that concerned my project. However, this is the advice I would give myself if I were in this position again.

My choice for the application of my machine learning algorithm, when I eventually came to it, was far from simple. As I learnt later on, making the different components of the device added a few steps to my project. For example, I had to compress the model, make sure the Raspberry Pi camera worked, and connect the Arduino and turbidity sensor and Raspberry Pi. This required unforeseen skills and had to be done in a short space of time, so there was little margin for error. It also pressed me for time, meaning I had to leave out improving the accuracy of my model.

This was another of my regrets – not being able to improve the model as much as I would have liked. The accuracy was over 80%, which I am very happy with, but I think that with more data refinement, I could have improved the accuracy further. I did follow Fred Mannings's advice on ways that I could improve my model. One of his recommendations was using greyscale on images, but this actually worsened the performance of the model. It's difficult to find out why and I know this is beyond my abilities for the moment. As I stated in the previous paragraph, the project left me relatively little time to investigate these things further.

Something that I definitely think was within my capabilities of changing was keeping on top of paperwork. I often wrote in my diary throughout the year that I hadn't kept documenting everything. Consequently, I sometimes did chunks of documenting in one go, instead of the more efficient alternative of incrementally adding to my evidence. So my piece of advice for my past self would be to try to remain up to date on my OneNote.

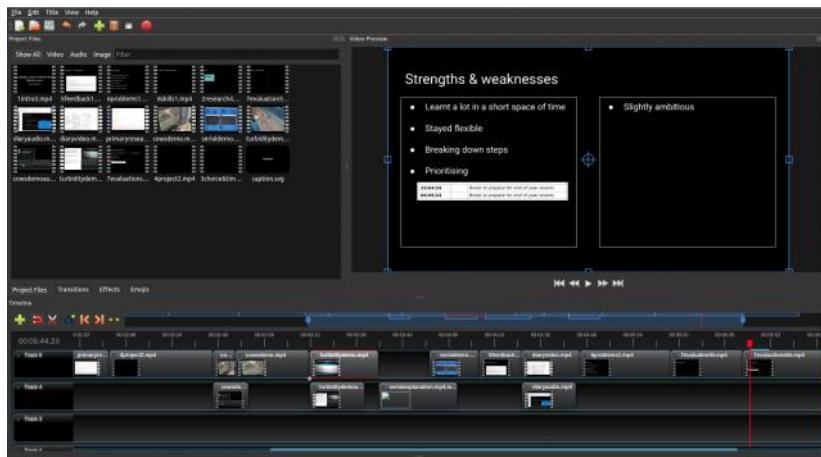
## How the project could go further

If I had more time, I would have liked to have the chance to do more field testing. Getting a long lasting source of power for the Raspberry Pi would have been another issue to resolve. I also want to leave the device collecting data for a prolonged period of time, so that I could see how much of a correlation there is between water turbidity and the presence of cows. I'm satisfied with my device because it is ready to test this hypothesis.

# Video presentation planning

Sunday, August 18, 2024 3:56 PM

## Software



I have chosen to use OpenShot as video editing software, since I already have some experience using this, and it is possible to use it on my computer. I am also going to use Google Slides for presenting because I am more familiar with this than Powerpoint.

For recording, I chose a screen recorder called Kazam.

## What I may want to include

- Why did you choose the topic and format that you did?
- Have you done as much as you expected?
- Did it take more time than expected or allowed? Why? What did you do about that?
- Did you meet your aims and objectives, including any amended ones? How do you know?
- How effective do you think you have been at managing this project? How do you know?
- What were the main decisions that you had to make during the project, and the main problems that you faced? How did you approach / solve these decisions / problems?
- Did anything in particular hold you back or cause problems?
- Did you find any particular source of information more helpful than others?
- What would you do differently if you were to do it again?
- What have been the main strengths and weaknesses of your project?
- What new skills did you learn/what new technologies did you use? Did you develop any skills that you already had?
- How did it differ to completing your A levels?
- In doing the Project, what did you learn about yourself and the way in which you learn?
- What sort of study habits do you think you possess? Anything about the way that you handle a project that surprised you?
- Has the Project helped you in terms of your future plans? (e.g. study at university; career choices; going into employment). Can you see yourself using any of the skills?

## Structure



## Initial plan

### Outlining what my project is, why I chose it

Mind maps  
Preliminary research  
Talking to Rob :)

### My project

Neural network  
Data  
Testing  
Raspberry Pi  
Turbidity sensor

### Planning

Timelines, to-do lists

### Research

Explain how my research has also been my skills development  
Did a lot of research before actually choosing my project topic!  
Primary research with Rob and surprisingly, TTP

### Decisions and problem solving

Realised that I wouldn't have time to make the actual device, but I've made every component of it and it all works  
Not everything can be researched  
Broken computer  
Accuracy isn't perfect... but to figure out why is far beyond my abilities within 9 months

### Helpful sources of information

Tutorials!! I had a lot to learn  
Fred Mannings  
3Blue1Brown helped teach me really important concepts like what is a neural network?

### What would I do differently?

Very difficult to say because I was starting from the beginning so I had no idea what I was doing  
Have a less ambitious project  
Do paperwork incrementally, instead of big chunks  
Most sources are blog posts and tutorials... but didn't want to sacrifice on quality of my research for the sake of ticking boxes

### Strengths & weaknesses

I learnt a lot, in quite a short space of time  
Keeping flexible  
Breaking down not just the project but every step of the project  
Getting behind on paperwork

### New skills & development

COMPUTERS because I knew nothing about them previously  
Problem-solving, and working around problems

Prioritising  
Project management  
Research skills

**Self-reflection**

I like the project more than the documentation and project management  
I like to-do lists  
I like things on paper

**How this is going to help me**

STEM-related career so computers important  
What a project can be like (if it's not well organised)  
Learning about ML & AI important in today's world  
Data science

# Video presentation script

Tuesday, August 20, 2024 6:40 PM

## FIRST DRAFT (after editing)

### Slide 3 - choosing the project

I decided after brainstorming to create a machine learning algorithm, but I wanted to find a real-life application for it. My granddad, a citizen scientist, collects data on water quality of a river he lives near. He wants to investigate the correlation between water turbidity and presence of cows. I realised that I could help with this by making an algorithm that detects the presence of cows.

### Slide 5 – research

Initially I didn't know anything about programming or machine learning. Before doing research on machine learning, I had to learn Python. I then used this new skill while doing machine learning tutorials, making simple regression and classification algorithms. Since I was both taking notes from sources, and learning to apply this, I considered much of my research as skills development as well.

As well as tutorials, my sources include blog posts, articles, books and videos. I kept track of my sources and evaluated them using the CRAAP method, on my skills tracker. I used around 50 secondary sources and two primary sources.

### Slide 2 – explaining project

After developing the knowledge I needed to begin, I built a neural network that classifies data into images with cows and images without. I did this by collecting images of cows and no cows from a variety of sources. After training the model a few times, I chose the most accurate model to use on the Raspberry Pi.

At set times, the device takes a photo and records the turbidity. The turbidity sensor is connected to the Arduino, which collects this piece of data and sends it to the Raspberry Pi. The Pi puts the data in a table along with if there are cows in the photo at that time.

### Slide 4 – planning & project management

I aimed for a fully functional device that I could use in the field. This goal was ambitious, but I came pretty close since all of the components of the device work individually. The project took longer than expected since I had to learn so much before I began. My initial timeline was basic while I was uncertain how long anything would take. Further along I made a monthly timeline and then a weekly timeline. I also made many to-do lists throughout.

When making the device, I realised that doing *all* of the research first then starting the project wasn't the right approach. Instead I researched while developing the project, because I couldn't anticipate what I *didn't* know.

### Slide 6 – problems

### Slide 7 – strengths & weaknesses

I am pleased with many aspects of the project. I learnt a lot about programming and machine learning in quite a short space of time. I also made sure not to be too rigid with planning and timelines early on, because a lot of plans evolved in the process. I broke down each part of the project into manageable steps which improved my efficiency. Throughout the year I made sure to work hard on the EPQ while ensuring it didn't interfere with my A-levels.

Nonetheless, in retrospect, choosing a less complex and ambitious application of my project could have led to a more complete outcome. I also often spent a long time on the actual project work without documenting my project. This meant I ended up doing the documentation in big chunks. If I did this again, I would do the paperwork incrementally.

I acknowledge that many of my sources are blog posts and tutorials; this could have been more varied. But I didn't want to sacrifice on the quality of my research as a result.

### Slide 8 – skills development

Some new skills I developed include working with computers. I've overcome my fear of the command line and I'm comfortable using Python. These skills will be essential when I do a physics degree, and for a STEM-related career. With the increasing prevalence of AI, I'm glad that this project has helped me gain a better understanding of it. More generally, I've learnt a lot about project management, research, and self-learning. These are skills I'll use no matter where I work.

### Slide 9 – self-reflection

Overall, I'm really happy with the amount I've learnt and achieved. I've also learnt some things about myself: I really enjoy working on projects, not paperwork – although I understand why both are essential. I like making to-do lists because timelines don't always go as planned. At the beginning, I had no idea whether I'd manage to complete my project, so this project has taught me about what I am capable of.

## **SECOND DRAFT**

### **Project management**

After brainstorming project ideas, I decided to create a machine learning algorithm. I didn't have any knowledge of machine learning, or much experience with working with computers. This meant I had to break the project into two different phases. Phase 1 was developing the skills I needed and researching the technology, and phase 2 was developing my final outcome.

### **Research & skills development**

In phase 1, I learnt some basic Python skills using Learn Python the Hard Way, because this was the language I programmed the machine learning algorithm in later on. I then used this new skill while doing machine learning tutorials, writing simple regression and classification algorithms. Since I was both taking notes from a source and learning to apply this knowledge, I considered it both research and skills development.

As well as tutorials, my sources include blog posts, articles, books and videos. I kept track of my sources and evaluated them using the CRAAP method, on my skills tracker. I used around 50 secondary sources and two primary sources.

### **Project choice & timeline**

Once I felt I had the abilities needed to make my own machine learning algorithm, I looked for a real-life application of it. My granddad, a citizen scientist, collects data on water quality of a river he lives near. He wants to investigate the correlation between water turbidity and presence of cows. I realised that I could help with this by making an algorithm that detects the presence of cows.

In this phase, I made a more detailed timeline to the ones I made previously and stuck closely to it, also making many to-do lists. I aimed for a fully functional device that I could use in the field. This goal was ambitious, but I came pretty close since all of the components of the device work individually.

### **Project**

I built a neural network that classifies data into images with cows and images without. I did this by collecting images of cows and no cows from a variety of sources. After training the model a few times, I chose the most accurate model to use on the Raspberry Pi. At set times, the device takes a photo and records the turbidity. The turbidity sensor is connected to the Arduino, which collects this piece of data and sends it to the Raspberry Pi. The Pi puts the data in a table along with if there are cows in the photo at that time.

### **Peer feedback**

Once I had finished constructing the neural network, I asked for feedback from Fred Mannings, who co-owns an AI start-up. He gave me feedback and suggestions about the way I had explained the project, the data I used, and the accuracy of the model.

### **Problems**

There were a few problems that I encountered while making my project. The biggest setback was when my computer broke, which meant working on another computer. Thankfully, most of my files were backed up. However, while my old computer had 2 operating systems that I could switch between for different aspects of my project, I can only use one operating system on the newer computer. I tried to change this but in the end had to settle with using Ubuntu, which works well for data science but is less convenient for Microsoft apps.

### **Evaluation; strengths & weaknesses**

I am pleased with many aspects of the project. I learnt a lot about programming and machine learning in quite a short space of time. I made sure not to be too rigid with planning and timelines early on, because a lot of plans evolved in the process. I broke down each part of the project into manageable steps which improved my efficiency. Throughout the year I made sure to work hard on the EPQ while ensuring it didn't interfere with my A-levels.

Nonetheless, in retrospect, choosing a less complex and ambitious application of my project could have led to a more complete outcome. I also often spent a long time on the actual project work without documenting my project. This meant I ended up doing the documentation in big chunks. If I did this again, I would do the paperwork incrementally.

I acknowledge that many of my sources are blog posts and tutorials; this could have been more varied. But I didn't want to sacrifice on the quality of my research as a result.

### **Skills development**

Overall, I'm really happy with the amount I've learnt and achieved. Some new skills I developed include working with computers. I've overcome my fear of the command line and I'm comfortable using Python. These skills will be essential when I do a physics

degree, and for a STEM-related career. With the increasing prevalence of AI, I'm glad that this project has helped me gain a better understanding of it. More generally, I've learnt a lot about project management, research, and self-learning. These are skills I'll use no matter where I work.

#### FINAL EDITS (on paper)

My name is Uxue and for my EPQ I did an artefact

##### Project management

- After brainstorming project ideas, I decided to create a machine learning algorithm. I didn't have any knowledge of machine learning, or much experience with working with computers. This meant I had to break the project into two different phases. Phase 1 was developing the skills I needed and researching the technology, and phase 2 was developing my final outcome.

##### → Research & skills development

- In phase 1, I learnt some basic Python skills using Learn Python the Hard Way, because this was the language I programmed the machine learning algorithm in later on. I then used this new skill while doing machine learning tutorials, writing simple regression and classification algorithms. Since I was both taking notes from a source and learning to apply this knowledge, I considered it both research and skills development.
- As well as tutorials, my sources include blog posts, articles, books and videos. I kept track of my sources and evaluated them using the CRAAP method, on my skills tracker. I used around 50 secondary sources and two primary sources.

##### → Project choice & timeline redo ✓

Once I felt I had the abilities needed to make my own machine learning algorithm, I looked for a real-life application of it. My granddad, a citizen scientist, collects data on water quality of a river he lives near. He wants to investigate the correlation between water turbidity and presence of cows. I realised that I could help with this by making an algorithm that detects the presence of cows.

- In this phase, I made a more detailed timeline to the ones I made previously and stuck closely to it, also making many to-do lists. I aimed for a fully functional device that I could use in the field. This goal was ambitious, but I came pretty close since all of the components of the device work individually.   
*believe I achieved it*

##### → Project redo

I built a neural network that classifies data into images with cows and images without. I did this by collecting images of cows and no cows from a variety of sources. After training the model a few times, I chose the most accurate model to use on the Raspberry Pi. At set times, the device takes a photo and records the turbidity. The turbidity sensor is connected to the Arduino, which collects this piece of data and sends it to the Raspberry Pi. The Pi puts the data in a table along with if there are cows in the photo at that time.

\*the Raspberry Pi is a small, cheap computer often used for coding projects like this one.

##### → Peer feedback

Once I had finished constructing the neural network, I asked for feedback from Fred Mannings, who co-owns an AI start-up. He gave me feedback and suggestions about the way I had explained the project, the data I used, and the accuracy of the model.

##### → Problems

There were a few problems that I encountered while making my project. The biggest setback was when my computer broke, which meant working on another computer. Thankfully, most of my files were backed up. However, while my old computer had 2 operating systems that I could switch between for different aspects of my project, I can only use one operating system on the newer computer. I tried to change this but in the

end had to settle with using Ubuntu, which works well for data science but is less convenient for Microsoft apps.

#### → Evaluation; strengths & weaknesses

I am pleased with many aspects of the project. I learnt a lot about programming and machine learning in quite a short space of time. I made sure not to be too rigid with planning and timelines early on, because a lot of plans evolved in the process. I broke down each part of the project into manageable steps which improved my efficiency. Throughout the year I made sure to work hard on the EPQ while ensuring it didn't interfere with my A-levels.

redo ✓

- Nonetheless, in retrospect, choosing a less complex and ambitious application of my project could have led more to a more complete outcome. I also often spent a long time on the actual project work without documenting my project. This meant I ended up doing the documentation in big chunks. If I did this again, I would do the paperwork incrementally.
- I acknowledge that many of my sources are blog posts and tutorials; this could have been more varied. But I didn't want to sacrifice on the quality of my research as a result.

could have given me  
time to have  
room for error

#### → Skills development

Overall, I'm really happy with the amount I've learnt and achieved. Some new skills I developed include

- working with computers. I've overcome my fear of the command line and I'm comfortable using Python. These skills will be essential when I do a physics degree, and for a STEM-related career. With the increasing prevalence of AI, I'm glad that this project has helped me gain a better understanding of it. More generally, I've learnt a lot about project management, research, and self-learning. These are skills I'll use no matter where I work.

- Diary
- Skills dev
- Problems: \* r-pi
- More ~~planning~~ about outcome
- Primary source

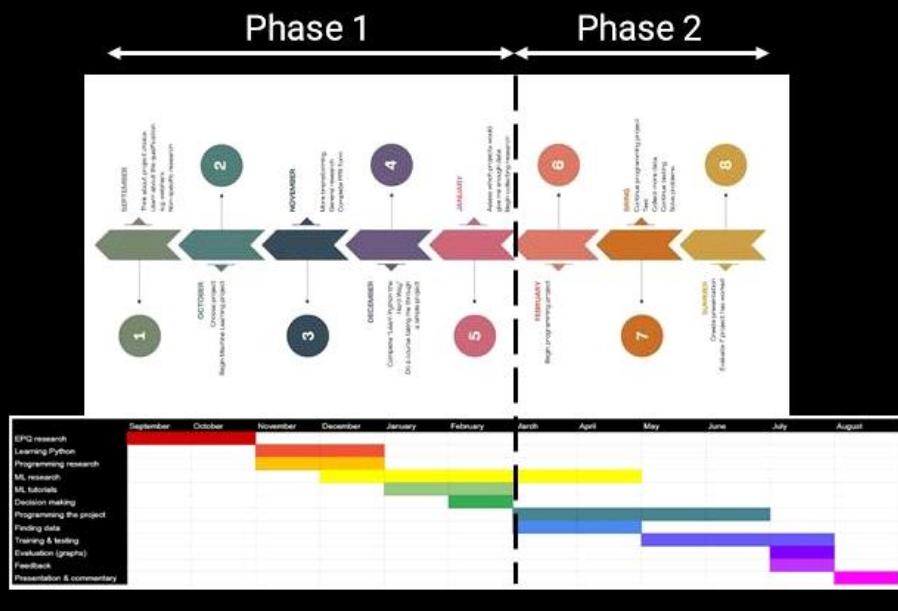
Throughout the project I updated my diary each week, with what I had completed, my evaluation of this week, the skills I had developed, and short term plans. The "evaluation" column helped me assess what was going well and what wasn't. It also helped to write down my general thoughts and how I felt the project was going.

SLIDES

# Building a neural network that detects cows

By Uxue Galvin

## Project management



# Research

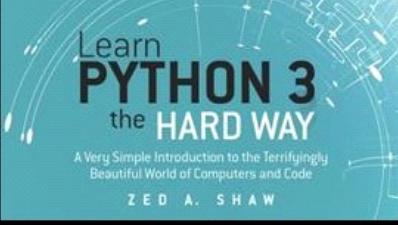
**Technology**

## The biggest scientific challenges that AI is already helping to crack

AI isn't just for chatbots – many companies are using it to tackle everything from protein folding and drug development to commercially viable nuclear fusion

By Matthew Sparkes

25 July 2023



**Learn Python 3 the Hard Way**  
A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code  
ZED A. SHAW

**Multiple linear regression**

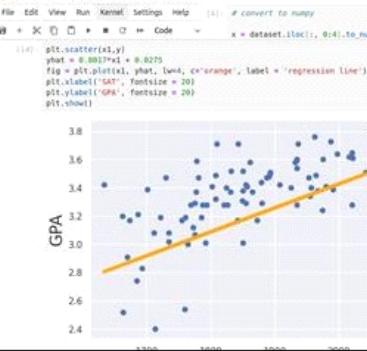
Linear regression with multiple independent variables

```
(1) import numpy as np
(2) import pandas as pd
(3)
(4) dataset = pd.read_csv('Tutorials/50_Startups.csv')
(5)
(6) dataset.head()
(7)
```

|   | R&D Spend | Administration | Marketing Spend | State      | Profit    |
|---|-----------|----------------|-----------------|------------|-----------|
| 0 | 165349.20 | 136897.80      | 471784.10       | New York   | 192261.83 |
| 1 | 162597.70 | 151377.59      | 443808.53       | California | 191793.06 |
| 2 | 133441.51 | 101145.55      | 407934.54       | Florida    | 191050.39 |
| 3 | 144372.41 | 118671.85      | 383199.42       | New York   | 182901.99 |
| 4 | 142107.34 | 91393.77       | 366168.42       | Florida    | 166187.94 |

**jupyter LinearRegression+testing.ipynb**

```
(1) # convert to numpy
(2) x = dataset.iloc[:, 0:4].to_numpy()
(3)
(4) # convert to numpy
(5) plt.scatter(x,y)
(6) yhat = 0.0037*x1 + 0.0229
(7) fig = plt.plot(x1, yhat, 'orange', label = 'regression line')
(8) plt.xlabel('GDP', fontsize = 20)
(9) plt.ylabel('GPA', fontsize = 20)
(10) plt.show()
```



**Intro to Logistic Regression**

A Logistic Regression algorithm works by implementing a linear equation with independent or explanatory variables ( $x_i$ )

$$z = \beta_0 + \beta_1x_1$$

$z$  is the linear target variable / response variable, and  $\beta_0, \beta_1$  are the parameters of the model

If there are multiple explanatory variables, this is extended to

$$z = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n$$

**Sigmoid Function**

$z$  is converted to a probability value between 0 and 1. The sigmoid function maps predicted values to probabilities

$$\hat{p} = \frac{1}{1 + e^{-z}}$$

**Decision boundary**

The probability value is mapped to either 0 or 1 based on whether it is above or below a threshold value, or threshold

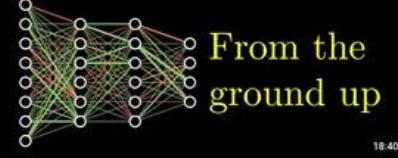
$$P \geq 0.5 ; \text{class} = 1$$

$$P < 0.5 ; \text{class} = 0$$

**Assumptions**

- Dependent variable is binary, multinomial or ordinal
- Observations are independent

**Neural Networks**



From the ground up



# Research

**Secondary sources**

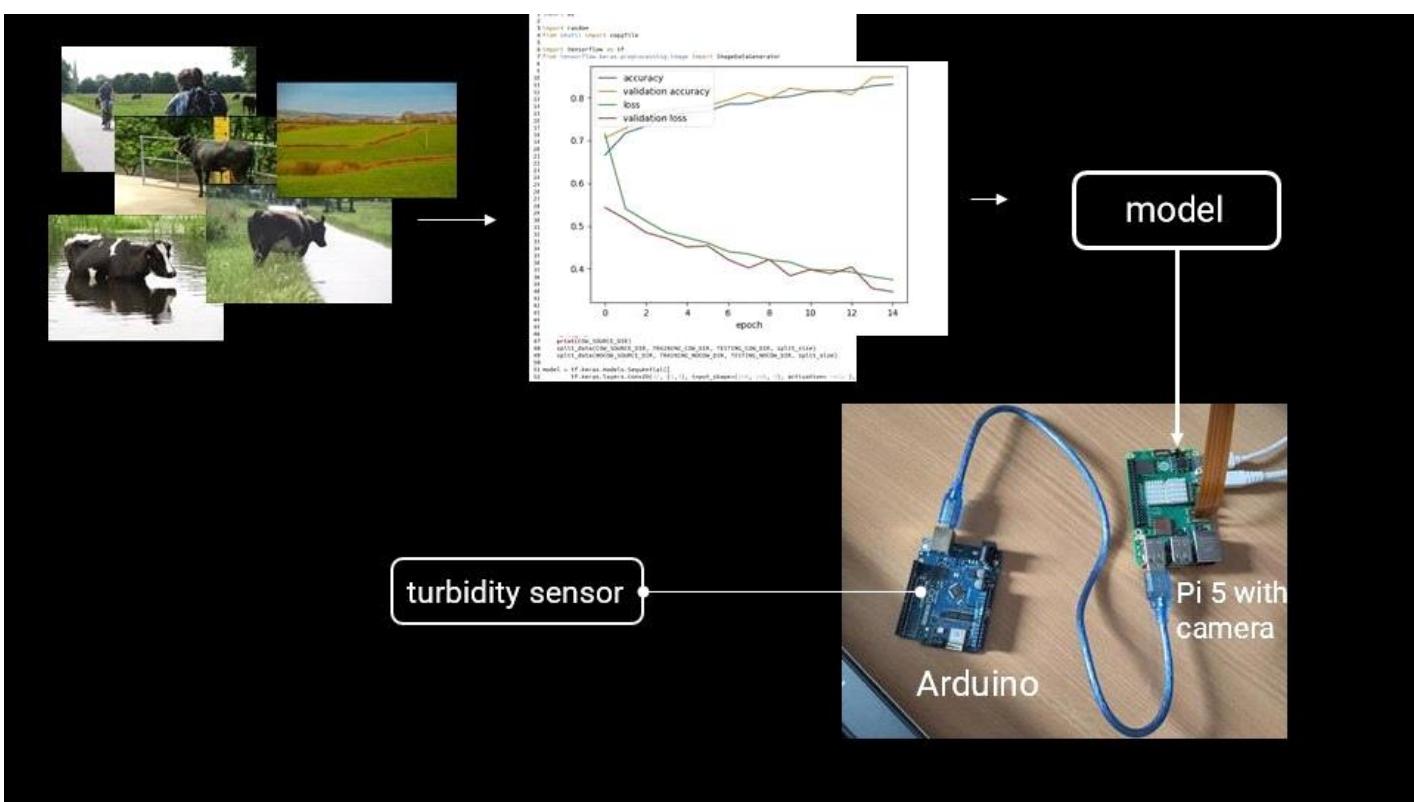
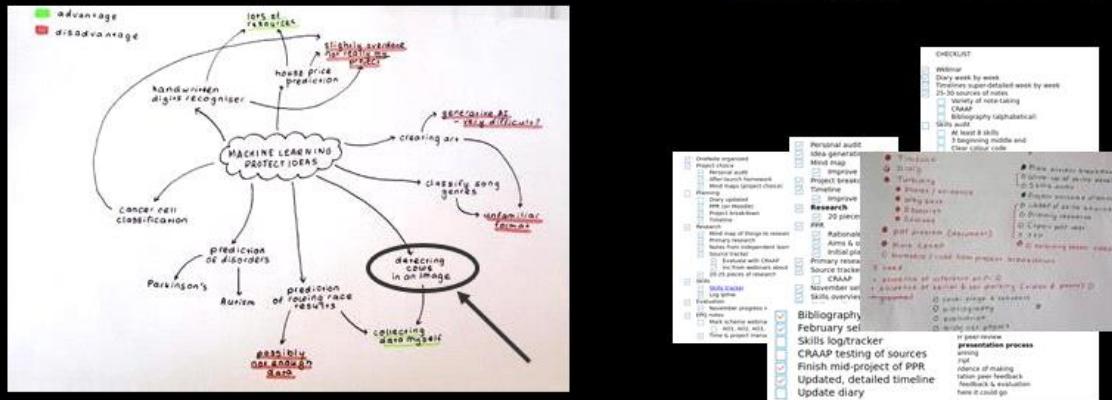
01 November 2023 15:48

| No. | Date accessed | Source                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | What is about?                                                               | Evaluation summary                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Issues?                                                                                                                                                                                                                                                                                                                                                                       |
|-----|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | 04/11/23      | <a href="https://www.coursera.org/learn/machine-learning-course/lecture/Ujm7v/what-is-machine-learning">Ng, A (2012) Coursera, Machine Learning, available at <a href="https://www.coursera.org/learn/machine-learning-course/lecture/Ujm7v/what-is-machine-learning">https://www.coursera.org/learn/machine-learning-course/lecture/Ujm7v/what-is-machine-learning</a></a>                                                                                                                                                                      | A course that teaches Machine Learning fundamentals                          | This course from Coursera, a popular platform which many people use to develop their career, seems trusted by many. The course itself is from Stanford University. Both of these organisations' purpose is to educate people. Andrew Ng, the instructor, has authored many books, papers, and online courses, reviewed files, etc. I think that it's just the first few weeks of the course that I'm looking at is the most useful, but quite a lot of credibility. | The course was released in 2012 and there have been developments in the field since. However, I don't believe that the developments since 2012 have greatly affected the core concepts of machine learning so I think this source will be sufficient to teach them.                                                                                                           |
| 2   | 05/11/23      | <a href="https://www.newscientist.com/article/2384085-the-biggest-scientific-challenges-that-ai-is-already-helping-to-crack/">Sparkes, M. (2023). The biggest scientific challenges that AI is already helping to crack. [online] New Scientist. Available at: <a href="https://www.newscientist.com/article/2384085-the-biggest-scientific-challenges-that-ai-is-already-helping-to-crack/">https://www.newscientist.com/article/2384085-the-biggest-scientific-challenges-that-ai-is-already-helping-to-crack/</a> [Accessed 5 Nov. 2023].</a> | Helping with scientific challenges in machine learning                       | This source definitely has some credibility in that the New Scientist is very well-respected as a reliable source of information. All of the people quoted are researchers at universities. It's a good article, but it's not very in-depth into the actual challenges that AI faces. I think it's relevant to my project as it's a general overview of what AI can do.                                                                                             | The article's purpose was to inform people of how AI is currently benefitting scientists, which means that it didn't go into much depth in the making of this AI and the projects are likely to be far above my level. Therefore I don't think that it is particularly relevant to my project as I think projects such as these would be far too advanced to do a project in. |
| 3   | 05/11/23      | <a href="https://www.newscientist.com/article/2384092-how-to-use-ai-to-make-your-life-simpler-cheaper-and-more-productive/">Stokel-Walker, C. (2023). How to use AI to make your life simpler, cheaper and more productive. [online] New Scientist. Available at: <a href="https://www.newscientist.com/article/2384092-how-to-use-ai-to-make-your-life-simpler-cheaper-and-more-productive/">https://www.newscientist.com/article/2384092-how-to-use-ai-to-make-your-life-simpler-cheaper-and-more-productive/</a> [Accessed 5 Nov. 2023].</a>  | Shows how AI can help us daily lives                                         | Seems like a good source as it's from a reliable source. It provides real-life examples of how AI can be used to make our lives easier. It also gives some inspiration on how to do projects.                                                                                                                                                                                                                                                                       | It didn't give much detail on how to do these projects as its purpose is to inform people of how they can use AI. As a result, it doesn't tell me about how these algorithms could be made.                                                                                                                                                                                   |
| 4   | 12/01/23      | <a href="https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/">Anyoha, R. (2017). The History of Artificial Intelligence. [online] Science in the News. Available at: <a href="https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/">https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/</a> [Accessed 1 Jan. 2023].</a>                                                                                                                                                            | Provides a good overview of how AI has developed from the mid 1950s to 2000s | Seems like a good source as it's from a reliable source. It provides a good overview of how AI has developed from the mid 1950s to 2000s. It's made by Harvard graduates. Since Harvard is a very prestigious university, this gives the source credibility.                                                                                                                                                                                                        | The post is from 2018 and there have definitely been significant developments in AI since then. Nonetheless, this has helped give me an idea of the timeline before 2010s. No sources are given, however the author, Rockwell Anyoha, works on machine learning methods according to this website.                                                                            |

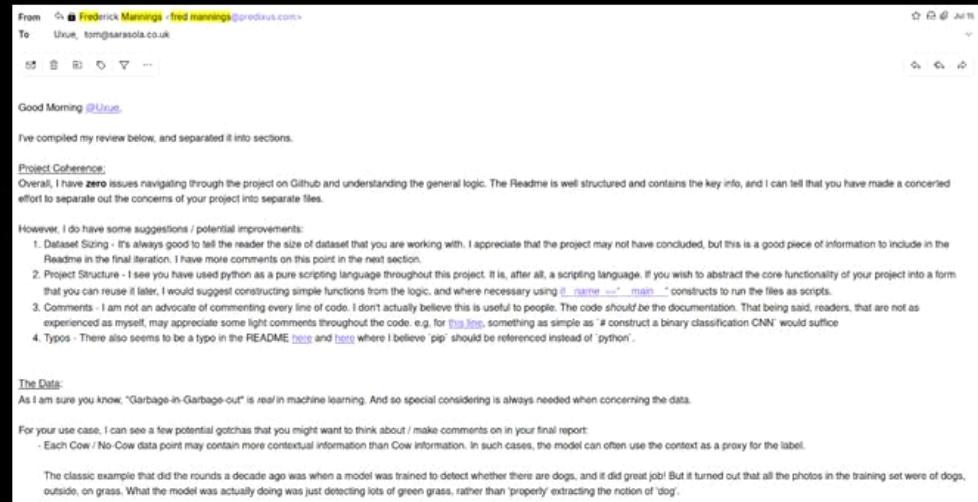
Currency  
 Relevance  
 Accuracy  
 Authority  
 Purpose

## Phase 2

| <b>Week beginning</b>   | <b>Tasks</b>                                                                                         |
|-------------------------|------------------------------------------------------------------------------------------------------|
| 11/03/24                | Research: neural network, convolution                                                                |
| 18/03/24                | Research                                                                                             |
| 25/03/24                | Prototype (a neural network that distinguishes between images of cats and images of dogs)            |
| Easter holiday          | Continue with prototype                                                                              |
| 15/04/24                | Work out how to use turbidity sensor and connect to Arduino                                          |
| 22/04/24                | Write code that sends the data from Arduino onto the Raspberry Pi and puts the data in a spreadsheet |
| 29/04/24                | Revision break for end of year exams                                                                 |
| 06/05/24                | Revision break for end of year exams                                                                 |
| 13/05/24                | Finish prototype (cats vs dogs)                                                                      |
| 20/05/24                | Start making dataset of cows and no cows                                                             |
| 27/05/24<br>(half term) | Collecting data<br><br>Away for half term                                                            |



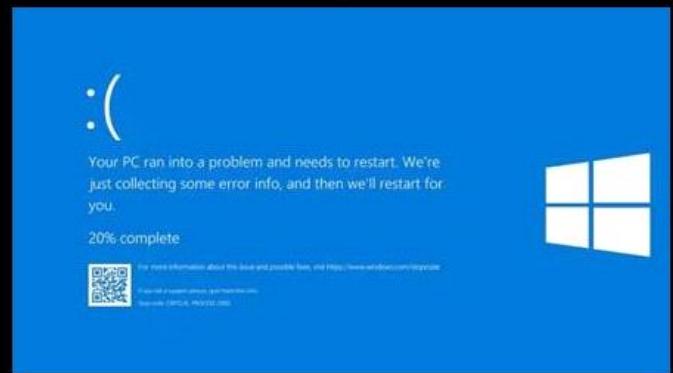
# Peer feedback – Fred Mannings



## Problem solving

- Training neural network takes too long
- Bias towards particular category
- Jupyter Notebook problem
- Can't find datasets
- pdf files not printing onto OneNote
- Raspberry Pi camera not working

Broken computer



## Strengths & weaknesses

- Learnt a lot in a short space of time
  - Stayed flexible
  - Breaking down steps
  - Prioritising

|                 |                                               |
|-----------------|-----------------------------------------------|
| <b>29/04/24</b> | <i>Break to prepare for end of year exams</i> |
| <b>06/05/24</b> | <i>Break to prepare for end of year exams</i> |

- Slightly ambitious
  - Sometimes got behind in paperwork
  - Types of sources

## New skills & skills development



ChatGPT

| writing JPEG Image to "web-can-shot.jpg".                                          |  | user@UserLatitude490:~/projects/EPQ/crosscan |  |
|------------------------------------------------------------------------------------|--|----------------------------------------------|--|
| [T]Item: exec@UserLatitude490:~/projects/EPQ/crosscan\$ open web-can-shot.jpg      |  | user@UserLatitude490:~/projects/EPQ/crosscan |  |
| ... Opening /dev/video0.                                                           |  |                                              |  |
| [T]Item: exec@UserLatitude490:~/projects/EPQ/crosscan\$ cat /dev/video0 opened.    |  |                                              |  |
| No input was specified, using the first.                                           |  |                                              |  |
| *** Capturing frame...                                                             |  |                                              |  |
| Captured Frame in 0.00 seconds,                                                    |  |                                              |  |
| Setting output format to MJPG, quality 80                                          |  |                                              |  |
| Writing MJPG Image to "web-can-shot.jpg".                                          |  |                                              |  |
| [T]Item: exec@UserLatitude490:~/projects/EPQ/crosscan\$ open web-can-shot.jpg      |  |                                              |  |
| [T]Item: exec@UserLatitude490:~/projects/EPQ/crosscan\$ cat results.csv            |  |                                              |  |
| [T]Item: exec@UserLatitude490:~/projects/EPQ/crosscan\$ No such file or directory. |  |                                              |  |
| [T]Item: exec@UserLatitude490:~/projects/EPQ/crosscan\$ gretsch                    |  |                                              |  |
| [T]Item: exec@UserLatitude490:~/projects/EPQ/crosscan\$ cat timestamp.csv          |  |                                              |  |
| 2024-07-13T05:44:05,351402,False                                                   |  |                                              |  |
| 2024-07-13T05:45:09,196849,False                                                   |  |                                              |  |
| 2024-07-13T05:45:21,077295,True                                                    |  |                                              |  |
| 2024-07-13T05:45:44,854874,False                                                   |  |                                              |  |
| 2024-07-13T05:45:56,735884,False                                                   |  |                                              |  |
| 2024-07-13T05:46:08,532325,True                                                    |  |                                              |  |
| 2024-07-13T05:46:32,420385,True                                                    |  |                                              |  |
| 2024-07-13T05:46:46,19,183985,True                                                 |  |                                              |  |
| 2024-07-13T05:47:08,051739,False                                                   |  |                                              |  |
| 2024-07-13T05:47:31,827795,True                                                    |  |                                              |  |
| 2024-07-13T05:47:43,799795,True                                                    |  |                                              |  |
| 2024-07-13T05:48:07,485249,False                                                   |  |                                              |  |
| 2024-07-13T05:48:19,2892895,True                                                   |  |                                              |  |
| 2024-07-13T05:48:43,2892895,True                                                   |  |                                              |  |
| 2024-07-13T05:50,883841,False                                                      |  |                                              |  |
| 2024-07-13T05:49:18,1730764,False                                                  |  |                                              |  |
| 2024-07-13T05:49:30,883841,True                                                    |  |                                              |  |
| 2024-07-13T05:49:52,523389,True                                                    |  |                                              |  |
| 2024-07-13T05:49:54,397965,False                                                   |  |                                              |  |
| 2024-07-13T05:50,1787732,False                                                     |  |                                              |  |
| 2024-07-13T05:50,39,860837,False                                                   |  |                                              |  |
| [T]Item: exec@UserLatitude490:~/projects/EPQ/crosscan\$ ^C                         |  |                                              |  |



# Presentation feedback

Saturday, August 24, 2024 7:18 PM

## **BLANK FEEDBACK FORM**

# Uxue's EPQ presentation feedback

\* Required

1. How well do you understand what I did for my project? \*

- 1 - I only understand a little bit of it
- 2 - I got a vague idea
- 3 - I have a lot of questions
- 4 - some parts could be elaborated on more
- 5 - I understand all of it

2. Which of these aspects need more evidence and/or explanation?

- Time management
- Research
- Skills development (inc. problem solving)
- Evaluation & reflection

## 3. Rate my presentation skills e.g. display, spoken communication

\*



## 4. Tick any that you feel could be improved or elaborated on in the video

- Detailed planning, mind maps, diary, lists, targets, timeline
- Clear evidence of research and how it was used to further project
- Mix of primary and secondary sources, wide range, large number
- Clear skill stretch, developed new skills, out of comfort zone
- Documentation of skill development clear, detailed
- Overcoming problems and making decisions
- My strengths & weaknesses
- Project's strengths & weaknesses
- Thoughtful, realistic and detailed evaluation of outcome
- What I would do differently if I did it again
- Excellent knowledge of content of project
- Creative and useful feedback collected
- Project represents 9 months of work
- Other

5. Rate the presentation overall \*

|   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|----|

6. Anything missing?

7. Any additional feedback?

## RESPONSES

## Uxue's EPQ presentation feedback

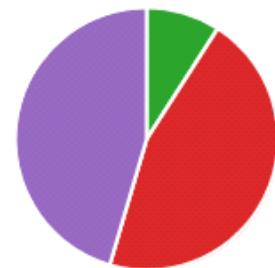
11 Responses

01:22 Average time to complete

Active Status

1. How well do you understand what I did for my project?

- 1 - I only understand a little bit ... 0
- 2 - I got a vague idea 0
- 3 - I have a lot of questions 1
- 4 - some parts could be elaborated 5
- 5 - I understand all of it 5



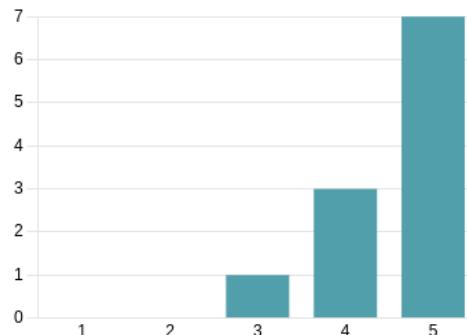
2. Which of these aspects need more evidence and/or explanation?

- Time management 2
- Research 3
- Skills development (inc. problem solving) 4
- Evaluation & reflection 1



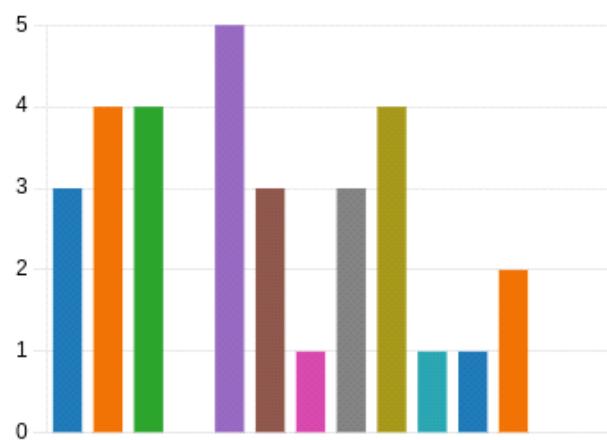
3. Rate my presentation skills e.g. display, spoken communication

**4.55**  
Average Rating



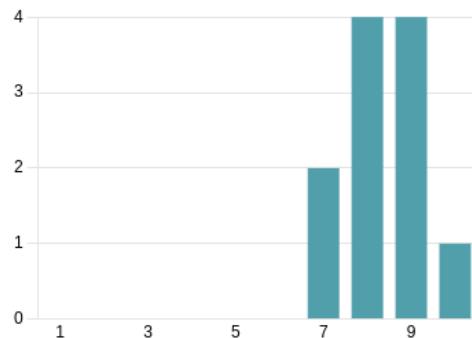
4. Tick any that you feel could be improved or elaborated on in the video

- Detailed planning, mind maps, ... 3
- Clear evidence of research and ... 4
- Mix of primary and secondary s... 4
- Clear skill stretch, developed ne... 0
- Documentation of skill develop... 5
- Overcoming problems and maki... 3
- My strengths & weaknesses 1
- Project's strengths & weaknesses 3
- Thoughtful, realistic and detaile... 4
- What I would do differently if I ... 1
- Excellent knowledge of content ... 1
- Creative and useful feedback col... 2
- Project represents 9 months of ... 0
- Other 0



## 5. Rate the presentation overall

**8.36**  
Average Rating



The feedback has been positive overall, and it has also told me what I could have improved. I think that the project outcome could have been better explained, and I should have shown more evidence of research and skills development. One of the specific things that people wanted more elaboration on was documentation of skills development. From this I can conclude that if I did the presentation again, I would add another slide showing the evidence of developing skills throughout the project, and talk a bit more about the final outcome. Nonetheless, I was approaching the time limit so I think my presentation has gone well, given the constraints.