CSC1028 Project Diary – Kaylem McShane

Developing machine learning models to solve problems is often described as more of an art than a science. This is another way of saying that it is very hard to guess how well a given technique will work on a problem. AI developers therefore spend a lot of their time searching over different machine learning architectures and hyper parameters trying to find ones that work well for their problem. This can lead to messy code and create a feeling like you are building on sand, as soon as you decide that one part of the model is working well changing another part may invalidate that choice. This has led to the field of AutoML which is a process of automatically searching for good AI architecture and hyper parameters. This approach is a good way to approach AI problems where instead of writing a single algorithm that you change you develop a python script that can generate different AI experiments with different architectures and parameters. Using this approach you can automatically run lots of different experiments and automatically search for the best approach. As you work on the project you are adding different alternative architectures to your python script so that you can automatically search over more approaches. In this way you never take a step back, every choice you add is at worst making your system try unhelpful approaches but the optimal design can be easily produced from your configuration.

The goal of this project is to take one or more of the AI algorithms described as part of the FastAI course (the leading course on Deep learning which includes a number of state of the art models for solving common machine learning problems). For each problem express the choices available in how the algorithm is implemented using a csv format (like a row in excel).

Make the resulting training algorithms capable of running on google colab and as a docker gpu container so that the machine learning algorithms can be easily trained on a cloud server.

Mulitple students can work on this project each focused on a different AI problem being configured e.g. Image recognition, time series prediction, object detection etc.

CSC1028 Project Diary – Kaylem McShane

Introduction

Note: As I am only beginning this diary on 19/01/2021 I am going to give a summary of the first fortnight of term, then moving forward I will log daily updates as necessary

On confirmation of my project choice, I was excited to begin working and preparing for the project. Previous to this course, I have had no experience in working with AI, but it was a concept that greatly interested me due to the plethora of applications and potential future developments that could be made with this technology.

Before getting into the FastAi course recommended by John, I began reading AI Basics by N.Gupta & R.Mangla. Whilst this text provided no insight to the practical implementations of machine learning, it introduced me to the theoretical areas of AI such as problem representation and AI components. However, the text was not particularly relevant to the current project as it branched into expert systems.

I did not carry out any further python training over the past 2 weeks as I have been using python since GCSE level and am proficient in the fundamental syntax and logic of python. Knowing I may be using it during this project, I did begin applying the skills I learned during CSC1027(OOP Programming) within python and developed 2 basic classes that modelled a bouncing ball and a ball during projectile motion. I used the matplot lib import to plot graphs to simulate the motion and visually represent the data as I felt these skills would be applicable across a range of projects.

From week 1 I began working through the FastAi course and the associated texts it recommended to begin to understand the practical and theoretical applications of the library. Whilst this is a practical module, I personally feel more comfortable and adept working on a problem with at least a fundamental background on the theory of the problem, therefore I began summarising key notes and familiarising myself with common terms such as the learning rate, metrics, loss function etc. Again whilst the mathematics is not the most essential feature of the course, I felt the optimisation algorithms such as SGD will be very useful whilst working on the project as a good implementation will ensure it runs well.

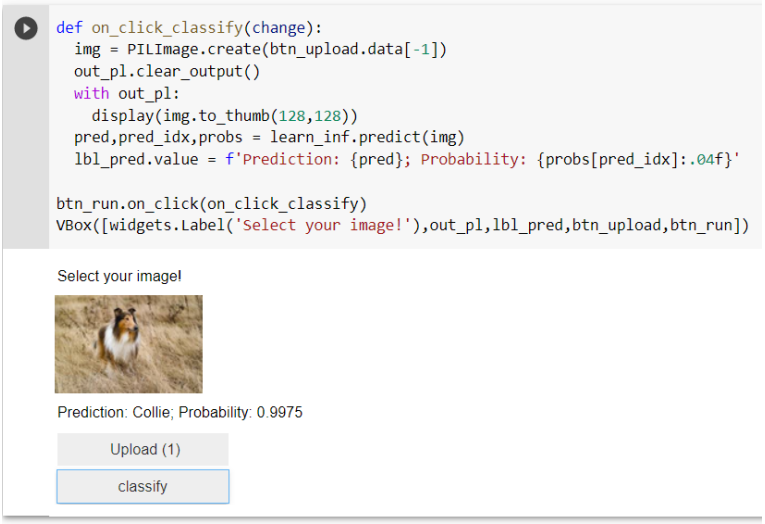
While researching the use of machine learning in Vision systems, I used ‘Visualizing and Understanding Convolutional Networks’ by Matthew D. Zeiler and Rob Fergus, to visualise how layers represent the images which

CSC1028 Project Diary – Kaylem McShane

improved my understanding of the application and enabled me to develop a basic app containing a model that recognises certain dog breeds.

```
def on_click_classify(change):
  img = PILImage.create(btn_upload.data[-1])
  out_pl.clear_output()
  with out_pl:
    display(img.to_thumb(128,128))
  pred,pred_idx,probs = learn_inf.predict(img)
  lbl_pred.value = f'Prediction: {pred}; Probability: {probs[pred_idx]:.04f}'

btn_run.on_click(on_click_classify)
VBox([widgets.Label('Select your image!'),out_pl,lbl_pred,btn_upload,btn_run])
```



For the data set of my app, I used the Bing search API contained within Microsoft Azure. Whilst this provided me with a large data set, it lead to several issues with regards to data integrity. My data set was irrelevant data items such as memes, adverts and other miscellaneous images. This was explored briefly in the fastai course through subjects such as “Clear skin vs acne prone skin”, as the bing API would tend to show altered or advertisement-based images thus creating a biased data set. This made me consider the data source I would use for my system in order to ensure validity and reliability throughout the project.

I used the course to explore the difference between single label and muti-categorical images which made me begin to consider the implementation of both cases within the project, truth be told I’m not entirely sure how to go about applying this knowledge but I know this is to be expected and once I have a reasonable understanding of the fundamentals I’m sure I will hit the ground running.

Over the past 2 days (18-19/01/21), I have been working on the collaborative filtering model and the fundamentals behind it’s implementation and the different applications available, such as Netflix movie recommendations and the optimisation of the model by identifying a suitable number of latent factors that provide a high accuracy whilst avoiding overfitting.
Before my meeting with John on Friday (22/01/21) it is my intention to have finished the fastAi course videos which will conclude by exploring the tabular modelling and NLP approaches and if I have time I plan to read certain extracts of a book I purchased over the holidays, “AI with Python” by Jim Smith

CSC1028 Project Diary – Kaylem McShane

Publishing to see if there are any methods or implementations worth adapting to operate with FastAi to improve the project.

20/01/21

Today I finished the tabular data model section of the fastai course and so far it has been the area that has most peaked my interest due to the range of predictive applications and various methods available to implement it. Whilst learning about the various pre-processing features to be implemented I began considering how these could be improved upon such as the automation of removing low-importance variables and redundant features by analysing the data that is used to create the visual graphs that enable the developer to manually optimise the model. I enjoyed learning about the ensembling process and how the neural net and decision tree models can be brought together to generate a more accurate result through averaging the results.

21/01/21

I moved on to the final section of the FastAi course which was the NLP deep dive. I honestly found this section of the course quite boring and found that it didn't quite peak my interest as much as other models. However, I still found it important to continue on and learn the fundamentals of the model such as the tokenisation and numericalisation processes, in order too ensure that I would be able to implement such a model if the project required.

Due to the requirements of other modules I did not get the chance to look further at "AI with Python" text but will try to explore it's contents over the weekend.

CSC1028 Project Diary – Kaylem McShane

22/01/21

Today was the first meeting with John with regards to our progress with the project. I was able to clarify the structure of the diary and what should be included so going forward I'll know what links and sources to use to explain how I improved my research. We each decided upon an area of ML to cover: I took Time Series, Mark took computer vision and Nathan took NLP. In preparation for this I am going to revisit the lesson on tabular data within the fastai course, I have created a Kaggle account in order to access a wide range of datasets and John also provided me with a series of links to use in order to find state of the art programs that can be adopted for later use.

https://paperswithcode.com/

https://modelzoo.co/
https://forums.fast.ai/t/language-model-zoo-gorilla/14623

https://www.microprediction.com/blog/popular-timeseries-packages

https://github.com/CMU-Perceptual-Computing-Lab/openpose

https://datasetsearch.research.google.com/

For the remainder of the day I intend to begin reading through some of the links and will continue to do so over the weekend so come Monday I can begin to formally research and document any options I think are suitable.

We also discussed some of the problems that could be tackled using time series prediction that didn't fit the common trend of such models such action recognition, cyber security and stock market research.

Finally, due to my progress being slightly ahead, John felt it would be best for me to centre my how to guide around time series prediction problems such as "How to recognise events within a Time Series System".

From the meeting, I went back to ensure the example from section 09 of the FastAi course was operational and encountered several issues that required addressing. Firstly, when accessing the Kaggle API, the .kaggle directory was not initialising on my device. Therefore I used the following code to download, install and move the relevant Json file to a new directory named .kaggle/. Source: kaggle.com/general/74235

CSC1028 Project Diary – Kaylem McShane

```
[5]  ! pip install -q kaggle
     from google.colab import files

     files.upload()

     ! mkdir ~/.kaggle
     ! cp kaggle.json ~/.kaggle/
     ! chmod 600 ~/.kaggle/kaggle.json
     ! kaggle datasets list
```

Secondly, when declaring the conditional and categorical variables the cont_cat_split() function, contained within the fastAi library was not running. Using gitmemory.com/issue/fastai/fastai/3156/760985175, I was directed the fastAi repository at https://github.com/fastai/fastai/pull/3157 which highlighted that this was an issue with the fastAi library not accepting Panda's data frames. There was a published temporary solution by a FastAi developer which provided me with the following function to hard code into the model.

```
[25] def cont_cat_split(df, max_card=20, dep_var=None):
         "Helper function that returns column names of cont and cat variables from given `df`."
         cont_names, cat_names = [], []
         for label in df:
             if label in L(dep_var): continue
             if (pd.api.types.is_integer_dtype(df[label].dtype) and
                 df[label].unique().shape[0] > max_card or
                 pd.api.types.is_float_dtype(df[label].dtype)):
                 cont_names.append(label)
             else: cat_names.append(label)
         return cont_names, cat_names
```

Thirdly, one of the provided sklearn imports were generating an invalid requirement error, I was then able to pull the upgraded install from stack overflow as followed.

```
[49] #does not work execute below code
     pip install –pre -f https://sklearn-nightly.scdn8.secure.raxcdn.com scikit-learn –U
```

```
  File "<ipython-input-49-af6a9d2d782e>", line 2
    pip install –pre -f https://sklearn-nightly.scdn8.secure.raxcdn.com scikit-learn –U
                ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
[50] pip install --pre --extra-index https://pypi.anaconda.org/scipy-wheels-nightly/simple scikit-learn
```

```
Looking in indexes: https://pypi.org/simple, https://pypi.anaconda.org/scipy-wheels-nightly/simple
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.6/dist-packages (0.22.2.post1)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.6/dist-packages (from scikit-learn) (1.0.0)
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.6/dist-packages (from scikit-learn) (1.19.5)
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.6/dist-packages (from scikit-learn) (1.4.1)
```

CSC1028 Project Diary – Kaylem McShane

Finally, when creating the TabularPanda Data Loader I exprianced an issue I had never encoutered before, nor found any clear soloutions on line or through the fastAi github. It stated that a value error had occurred and was unable to coerce to series, length must be 1, given 0. I contacted John for assistance who sign posted me towards https://forums.fast.ai/t/09-tabular-value-error-unable-to-coerce-to-series-length-must-be-1-given-0/80580, a fastai forum which taught me that this issue occurs when a list is empty. As the normalisaton process requires numeric inputs, or else the list is converted to an empty list, I had to cast the list to a float data type as opposed to an object.

```
df_nn_final.dtypes
df_nn_final = df_nn_final.astype({"saleElapsed": float})
df_nn_final.dtypes
```

```
[113] procs_nn = [Categorify, FillMissing, Normalize]
      to_nn = TabularPandas(df_nn_final, procs_nn, cat_nn, cont_nn,splits=splits, y_names=dep_var)
```

Beyond these issues, the code is now fully operational and I now have access to kaggle datasets for future use.

25/01/2021

Today I began researching pre-existing state of the art models that I could apply to the project. I am finding it quite difficult due to my limited knowledge in the area but John was able to assist with several links, giving me a better idea as to what I am trying to find. I managed to find a state of the art skeleton-based action recognition model that expands on Graph Convolutional Networks to focus more on joints and bones for accurate recognition. The model has a 99-96.1% accuracy, however, it was not developed using the fast ai library therefore I am unsure of the suitability for or project or what steps need to be taken to allow it to function correctly. (https://paperswithcode.com/task/skeleton-based-action-recognition)

Another model that John recommended was Human Activity Recogition using FastAi which was published on Kaggle. I read through the code implementation of this model and I do believe it could be adapted for a range of different implementations, given that a standardised data cleaning process is applied in

order for the model to cope with cotinuous and catagorical variables, due to the model only being designed around cotinuous independent varables and a catagorical dependent. (https://www.kaggle.com/abhisheksinghblr/human-activity-recognition-using-fast-ai/notebook). This example has given me a bit more reassurance and I am going to continue this line of study with pre-exisiting fastai models.

I haven't yet found a functioning model to support this but in theory I believe an audio event recognition system could be implemented using CNN and the librosa library to convert audio files into spectrograms that can be passed into the model(https://librosa.org/doc/latest/index.html).

Having read further into the DENet architecture recommended in one of the links John has passed on I belive it can be passed through the fastai Learner as it is classed as a recurrent CNN. I have began reading the paper on it and have identifed the ideal learning rate, optimiser and batch size, however, I am not yet sure how to pass the input data for training. I have found a Fastai audio tutorial I am going to read over and see if it is applicable and I may contact John too ensure this is a feasible challenge and the libraries are compatible. (file:///C:/Users/mcsha/Downloads/Greco2021_Article_DENetADeepArchitectureForAudio.pdf,

https://github.com/MiviaLab/DENet,

https://nbviewer.jupyter.org/github/mogwai/fastai_audio/blob/master/tutorials/01_Intro_to_Audio.ipynb,)

The following link leads to a site detailing how to derive hyperparameters from a dataset. This can be used to create a CSV file of the hyperparamters of an audio file which could then be passed to the tabular learner to be processed. I believe this does tie in with preparing the data for the DENet arhitecture,however, if it so happens that DENet architecture cannot be implemented within fastai I believe cleaning the data and forming a csv file will be an appropriate alternative. This method will also provide interoperability with the Human Activity recognition project, meaning only 1 type of learner is required as opposed too 2(Tabular & CNN). (https://www.kdnuggets.com/2020/02/audio-data-analysis-deep-learning-python-part-1.html)

CSC1028 Project Diary – Kaylem McShane

26/01/21

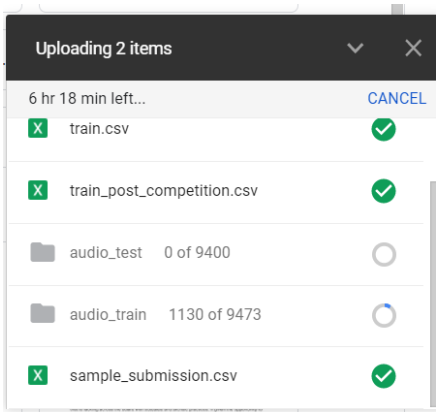
So far today's had a bit of give and take, I tried using the fast ai model linked above to try and adjust the human activity recognition model, however, as this model is still in development not all libraries are fully available thus restricting access to the dataset and certain libraries. While searching for a solution I was able to locate a pre-trained audio recognition model that was created within Fastai on the following github repo: (https://github.com/dzlab/deepprojects/blob/master/classification/Freesound_General_Purpose_Audio_Tagging.ipynb)

From this I learned I was mistaken in some of my assumptions yesterday, mainly that CNN is suitable for use in Time Series, and by using the librosa audio library to convert audio files into spectrographs, there is a clear method to train a visual model that represents audio. The dataset it accesses is stored within google drive so I am firstly going to link my personal google drive to use on it too ensure the model is operational, then I will consider how to alter the model to take any audio dataset.

Finally finding a model that functions in away that suits the requirements of this model is a big relief. Despite beginning to understand some of the fundamental implementations and theories around AI models and the fast ai library, I still feel unprepared to readjust and redesign models to suit particular needs and being able to find existing models that can be suitably implemented with minor adjustments would be more effective given my current level.

After implementing the audio model, I am going to focus my time into finding a functional anomaly detection model that is focused on text analysis, so if I am successful I will have 3 functional time series classification models.

Update: Have now discovered that google drive + large data sets = long waits:/. I'll begin looking into the anomaly prediction models as I await the count down.



CSC1028 Project Diary – Kaylem McShane

While waiting on the upload, I began working through a tutorial I found on fastai2 and the timeseries learner which greatly helped my understanding of the topic and made me think I had found a feasible solution to several of the problems I had been facing. However, when I tried to run the code I learned that despite the tutorial being live, the library is no longer supported and has become outdated meaning it cannot be ran, resulting in a lot of wasted time. On top of this, after the upload completed I realised that a significant proportion of the data set was not made available by the developer of the audio model and also that several libraries were no longer usable. At this stage I was frustrated with the whole process and decided it wasn't worth pursuing anymore so I decided to leave it for the night and contact John in the morning for advice in finding suitable models.

27/01/2021

Productive morning so far, after what felt like an endless run of Errors yesterday, I finally found a working time series library within fastAi that can be used to operate on raw data. I plan to continue reading up on this model to see if it has been reworked for image recognition and if so, apply it and the librosa library to audio data to create an accurate recognition model. (https://github.com/timeseriesAI/tsai)

I spoke to John after all the issues yesterday and he said this was actually a good outcome, which is reassuring, as it means very few state of the art models have been implemented within fastai. Therefore, if we run the state-of-the-art models normally, this can be expanded upon to integrate them into the fastAi Library.

So far, the tsai model is promising, I've worked through the tutorials published on Github and they're all executing without a problem. I've read and made notes on the data preparation section which detailed the shape and requirements of any data structures that are passed to the model. I also studied the image time series model and how images are processed into tensors. These functions all operated as required, therefore, if I can construct a generic function to take an audio based data set and convert it into spectrograph images I could theoretically apply the imaging principles to analyse the sounds. The only area I am concerned about is maintaining the labels but I will look into this.

CSC1028 Project Diary – Kaylem McShane

For the remainder of the day I want to refocus on non-fastai based state-of-the-art models which can be converted. As of now, I have looked into the DENet architecture but I have only been able to view the library and as of yet no applications of the model.

I went back to the previously referenced skeleton-based action recognition model and was able to find a pretrained model that may be useful. The github repo didn't state the necessary imports required and as I have to run it on my local device to test I will need to look into the research paper to see if it references the required imports. The DENet architecure seems slightly more complex as there is no availible pre-trained models. I feel my lack of knowledge is prohibiting me in this area so I think it's best to step away from it till my next supervision meeting and focus on another area- another model or even converting audio files too images from a dataset.

Between now and Friday I have to focus on an assignment within another module but I will expnd on this and try and find an other useful model before the meeting on Friday. I would say up to know the work I've carried out has been very productive, whilst I only have limited functioning code to show for the work, I now have a better understanding of fastai and what is not availible or supported as of now.

28/01/21

I didn't spend much time on the project today but I was able to make a good breakthrough on the audio front as I found and impelemented a conversion application from audio file to image. (https://www.kaggle.com/rftexas/converting-sounds-into-images-a-general-guide)

Using the free_sound_audio data set I was able to download the above code and implement it. These images can then be passed to the data prep module in the tsai git repositiory which will normalise and clean the images whilst maintaining 100% of the meta data.

This breakthrough was not perfect, however, and several alterations are required. Firstly, the module was not able to run on colab due to google colab not permitting local directory access except for individual files. To overcome this issue temporairily I ran the programme within pycharm.

CSC1028 Project Diary – Kaylem McShane

Secondly, the file name of the audio file and the name of the saved spectrograph do not match. This will create an issue with labelling the datasets using the provided csv as the order of the directory does not match that of the csv file. Therefore I will have to amend the labelling process.

Thirdly, the programme uses a for loop to take each individual file and perform individual transforms. This does not impact the performance of the output of the model, but it does result in a less efficient use of processing power, especially when converting a 20,000+ dataset.

Whilst a lot of work remains on this front, this was a massive step forward.

29/01/21

Second meeting with John today. Before the meeting I revisited the audio/image converter and was able to resolve the naming issue. It was due to the .split function being told to ignore the first 2 "/" instead of 3 – the issue was easily resolved and was a consequence of my directory construction.

```
def rename_file(img_name):
    img_name = img_name.split("/")[2]
    img_name = img_name[:-4]
    img_name += ".jpg"
    return img_name
```

```
def rename_file(img_name):
    img_name = img_name.split("/")[3]
    img_name = img_name[:-4]
    img_name += ".jpg"
    return img_name
```

I was also asked to share some of my google colab files to Mark and Nathan in order to help them with their projects and dealing with the uncertainties of fastai. This will benefit the project as a whole as if we are all using the same libraries, there will be compatibility between the projects allowing for future integration.

Post-meeting 29/01/21

Had my second meeting with John today, the progress I was making with regards to the overall understanding of the project was positive and was even able to assist the others in the project. We discussed the need to begin working towards the overall aim of the project therefore John suggested we all firstly revisit the project description to reaffirm our understanding of the goal. John also recommended the open pose and YOLO libraries for me to use for

analysing a stream of data. For the remainder of today I'm going to try and slow down and revisit the key information of the project and begin researching open pose and YOLO. Due to working commitments I won't revisit the project until Sunday but by then I hope to be in a position to implement some open pose and yolo libraries and if the progress is positive I may try to produce a working time series by the next meeting(better to be ambitious and make some progress rather than focusing on the small stuff).

The following site provided a good integration of open pose and fastai's vision module.

(https://towardsdatascience.com/openpose-pnasnet-5-for-pose-classification-competition-fastai-dc35709158d0)

31/01/21

Took some time today to look into existing action recognition models that used fast ai and libraries such as open pose. This is a topic of widespread discussion but minimal implementation therefore I may have to explore converting an existing model into fast ai. I did, however, find several models online using fastai and opencv. One that stood out and I am currently trying to fix is a real-time hand key-point detection model. Once the model can detect hands, it can be improved by then recognising particular gestures.
(https://towardsdatascience.com/from-zero-to-real-time-hand-keypoints-detection-in-five-months-with-opencv-tensorflow-and-fastai-f98e87221475)

(https://colab.research.google.com/drive/14-SR-31tV_RuRK2wM8C4VAmcSxH9Z-lR#scrollTo=Iya8qxka35Pz)

OpenCV does appear to be a more commonly used package with regards to fast ai and with its emphasis on real-time computer vision models this could be a good starting point to explore for creating an operational model.

This lead me to the following tutorial and git repository for a facial expression detection model. (https://towardsdatascience.com/video-facial-expression-detection-with-deep-learning-applying-fast-ai-d9dcfd5bcf10).

After having a quick read of the report, I believe this model will be a great start for creating a functioning time-series recognition model because:

1. It contains a git repository will all required datasets

2. It includes a demonstration of setting up the live video feed.
3. It shows how to use the trained model in conjunction with the video feed to make predictions and writes them to a csv file, enabling the system to log events.
4. Finally, it is constructed in google collab and will integrate with our projects aims.

01/02/21

The main focus of the next day or two is the construction of the how to guide. My guide will be showing how to operate colab and fix common bugs that are still live within the git repository. Whilst this is a change in direction from what I discussed with John, I felt my initial concept of discussing the classifier was not unique enough and would have very little impact on the community. This debug guide on the other hand, directly addresses a range of issues that have not been resolved and compile solutions from numerous sources into a singular document that can be used as a platform for future development.

Due to the issues I had personally with different sources not operating sufficiently, I believe this guide will be of excellent use to the wider community and I have uploaded my amended model to github and included a link to the repository in the guide.

02/02/21

Today I completed my how to Guide and implemented it within a html site. I feel it adequately covers the specified topic without needlessly repeating the source information. I have now refocused on my project and found a functional audio classifier, implemented within fastai with a complete data set. With the exception of a few imports I had to bring across the model is now fully functional. If I can generalise the datasets and export the trained model I am confident that this will be a suitable addition to the project as samples of a data stream can be passed through. (https://github.com/fastaudio/fastaudio).(72% accuracy).

After revisiting the tsai github cited earlier I have found an imaging time series notebook that I intend to study and get running so I can now have functional

audio and imaging models. (https://colab.research.google.com/drive/1RS17q-QPuucXA0JigxI-qK5tjGEl1mzq). As the model works with up to 86% accuracy, I believe the next step in my development should be applying the model to a data stream in order to have it acting with the appropriate time series behaviours. Based on John's recommendations last week on open cv and open pose I have been trying to find a suitable model that applies these libraries for action recognition. I found the following site and shared it with the other members of the group due to the range of open source projects available within it, I believe it will be a good starting point for looking into open pose and open cv more. (https://awesomeopensource.com/projects/time-series-classification)

How to Guide: Reasoning

For my how to guide I decided upon a tutorial to fix up a pre-existing notebook for educational use, including the setting up and initialisation of the notebook on the online google colab system. My reasoning for creating this guide is that I found due to the rapid development of ai technology, libraries are changing and are being updated constantly with backwards compatibility not being the key priority at this moment in time, therefore, individuals such as myself who are aiming to further their understanding of the subject area are being hindered by trying to repair and create functional models out of published examples. When operating example 9 from the fastai course, I experienced issues across all areas of the model: creating the dataset, preparing the data, and creating models in both tree and neural network form. Therefore, by giving examples of how to resolve these issues, it allows the wider community to begin using the example, but also to transfer the displayed solutions to other models for future reference. My guide is heavily textual with supporting images where required but I try to minimise the use of specialised terminology unless it is applicable to the problem at hand. This removes the need for a formal background in ai and makes the guide accessible to all users. I also included links to the key pages and forums I used to gather my information to highlight the starting point for solving any problems that the reader may face. When researching other guides and solutions, a common theme I discovered was the publishing of the github repository of the completed example. Therefore, I made my own adjusted copy available in line with the GNU General Public License. Similar to this guide I had discovered on Medium(https://medium.com/@pierre_guillou/fastai-how-to-start-663927d4db63), I did not intend to copy the work of Jeremy Howard and the

fastai team but instead summarised and highlighted important points of information whilst providing the links and access to the course for those who wish to learn more. I believe this guide will effectively direct those who are new to fastai to begin recognising problems and the sources that are trusted and have the potential to resolve the problem, developing their skills and furthering their work.

05/02/21(before meeting)

Truth be told I wasn't able to commit as much time as I would of preferred into researching and working on the main project as I dedicated a significant amount of time into the how to guide and also had commitments to another coursework piece that required work within a group. After the meeting with John, I'm planning to focus back on the projects tasks and begin working to the overall goal.

After Meeting

In the meeting with John, we spoke about moving beyond the initial research period of the assignment and beginning to home in and develop one of the pre-existing models that I discovered. It was recommended that I plan out the individual "building blocks" of the code and identify the functionality of each section and how it can be generalised for use for systems with a range of datasets. We discussed the need to prevent an over reliance on an individual dataset and the lack of standards with regards to data labelling and how this was a preventing factor in the project, pushing me to consider how to address this.

To do

- Break down existing models to identify key building blocks
- Consider the different ways in which models can be trained
- Consider the best way to treat datasets and define a standard labelling process for the project.

CSC1028 Project Diary – Kaylem McShane

08/02/2021

I took some time today to break down the audio classifier code into more general steps so it can be applied to the project. The dataset used contains a csv file of metadata which can be used for labelling. The 3 key attributes are 'filename', 'fold' and 'category'. I am aiming to take another unrelated dataset and create a csv file with the following attributes and train the model on this new dataset. If I am successful, then this will show the code has a general structure and can be repurposed for further use.

I also discovered a series of youtube videos on audio processing- more accurately around extraction and time domain audio features. This model is a basic classifier therefore it is important to ensure it is applicable in a time series context. (https://www.youtube.com/playlist?list=PL-wATfeyAMNqIee7cH3q1bh4QJFAaeNv0)

09/02/21

Today I dedicated a significant amount of time in altering the esc50 audio classifier to work with a different dataset and was finally successful. Initially I believed that by reworking the csv file to contain the 3 key attributes referenced in the initial model, this would be enough to replicate the conditions required to operate the defined splitter. This however, was incorrect. In order to over come the issue and replaced the cross validation splitter with a TrainTestSplitter that was used in the fastai course. This took a random sample of 20% of the dataset for testing. The labelling process remained the same and read the labels from a csv file. The dataset I initially began working on was the Freeset Audio Tagging 2019 dataset available on Kaggle. After some assistance from John, I was able to place the data from a pandas data frame, to a datablock and finally into a data loader. This, however resulted in an error due to the dataloader containing tensors of varying sizes. John advised me that this error was a result of the audio files being of varying lengths and sizes. From the fastaudio online guide(https://fastaudio.github.io/API_Reference/augment.signal/), I was able to find and employ the ResizeSignal function to carry out an item transform that converts every audiofile to the stated length. The code then executed without fault until it was time to train the learner. Unbeknownst to me the labels of the dataset I was using were quite sporadic and didn't use standard naming conventions i.e 'crowd,applause' & 'screaming,crowd' were both labels

within the same dataset. This resulted in issues where the training and testing sets did not contain each type of label and threw an error, leaving me unable to train the model. This left me at a cross roads, as the dataset contained >4500 audio files, manually labelling them was not an option. Therefore I had no choice but to use a new dataset. On Kaggle, I then found a dataset containing raw audio files of dogs and cats. (https://www.kaggle.com/mmoreaux/audio-cats-and-dogs)

For the most part this dataset integrated directly to the code, everything was pre-processed ok, however, the labelling convention for this dataset was to use the filename of the wav file. In order to test the characteristics of the model I had been working on, I repurposed the train_test_split csv file to contain a column of file names, and I then applied an excel formula to extract the labels and auto-populate the label column. I then manually uploaded the file and executed the model which trained successfully to approximately 76.7% on the initial run and was then improved to 98.2% by applying the learning rate finder and retraining.

Today was successful because I now know that the models can be repurposed with minor changes to the functional code within it. I know have a standard model that pre-processes and extracts the relevant information form csv files.

Going forward, my next step will be to apply more labelling methods as to increase the range of datasets that can be applied. A method I am considering is taking the label from a title and writing it and the file name to a csv file which can then be stored and used within the model.

12/02/2021

Due to my software design principals coursework, I haven't been able to dedicate much time the past 2 days to the project but after the meeting I'll be back into it.

After Meeting

We have just had our latest meeting with John. In terms of progress, the breakthrough with the audio classifier has pushed me ahead and allowed me an air of flexibility with regards to where to go next. John mentioned exporting the working model and creating a new colab notebook which could be used to

run predictions and analyse the functioning model without all the training code behind it. We also discussed the need to break down the code and present it as a generic guide so it can be easily adjusted and viewed as user friendly to further push the development of the overall project. When I asked about the labelling process, John explained the role our project plays in a larger project which reflected the need for understandable colab code for training and analysis. We also discussed the upcoming blog post assignment and the role it will play in bringing new comers to a project such as this up to speed in a limited amount of time, i.e a week. John also referred to the possibility of branching into speech recognition and I raised my own views of using the tabular learner to approach the data in another manner. The key take aways from the meeting are:

- Create a prediction application for exported models
- Evaluation and explanation of input data
- Begin working on the blog post
- Explore the tabular implementation of audio data
- If all the above is met, contact John about speech recognition.

13/02/2021

Between last night and this morning I set out to create the prediction algorithm for my exported models. I used the example from the fastai course to format the data and the learn.predict method to carry out the predictions on the file. Initially I used absolute file paths which were functional but did limit the scope of the applications functionality. I used the path.ls function, along with the file_exts parameter to generalise this meaning the code can be applied to a greater range of models. Firstly the program returns all files with the .pkl extension to find and load the model to the learner. Then, the system takes a user input to find the relevant file extension, this is verified by ensuring the '.' Is present and adding it if it is not. If the user enters a non-sensical extension, an empty tensor is returned thus making the code obsolete and maintaining integrity.

I shared the code with Nathan and Mark, I don't know if they'll find any use in it but I figured it might assist them in testing their own models, especially for Mark whose also using the fastai vision imports.

CSC1028 Project Diary – Kaylem McShane

15/02/2021

Over the weekend I cleaned up the prediction model I created and began researching the implementation of audio models within a tabular setting. What I discovered was that at this moment in time, any state of the art models are relying heavily on CNN's and melSpectrogram analysis. I found this quite surprising as with any data conversion i.e audio→image, there is always going to be a degree of loss. A tabular model on the other hand would extract all the relevant data from the files and therefore propose minimal loss. I've yet to find whether there is a practical reason for not using tabular analysis or if it is just due to a lack of research as it is a relatively new field with regards to fastai.

For the time being I am going to take the audio classifier I edited and modify the comments to improve usability by end users and sign post any adjustments that could be made to improve it's applications.

I'm currently trying to create a more flexible and automated learner. In order to this I intend to use the same methods that I employed in the prediction model i.e using file extensions to identify the required files. This can be used to pull the .csv meta data files and then also to place all .wav files in the same directory for access.

I have now got the model working on the cat and dog dataset. This was a good starting point as I knew this dataset was functional. The next test will be too take a new dataset and apply it to the program too ensure it is truly flexible.

I found another dataset that was similar enough to be applied to the model but still different enough that it can be tested on the model. So far I have been able to load the dataset, place it in the datablock but once it reaches the dataloader it crashes. I encountered a similar issue previously and found the issue to be related to the filepath, however, after checking this data I am not sure what the issue is. I will revisit it tomorrow and if needed contact John.

5 minutes later: so I fixed it. Problem was due to an unlabelled training set being included with the labelled training data, an error was thrown by the get_y function. After removing these files from the set and csv file the programme ran successfully. Initially, the accuracy was only 56%, however due to the use of the learning rate finder I was able to improve this to 80%. This is for a medical model that can distinguish between regular heart beats and

those with medical anomalies and considering the intricacies of this model against a simple cat vs dog classifier I believe this is a very positive outcome.

16/02/21

From the meeting on Friday, I have covered all the points that were raised with regards to researching and building on the generalised structure of the models. Therefore, I personally feel it would be beneficial, given my position to dedicate the next two days to begin planning and building my blog post as I have to dedicate Thursday to preparing my SDP coursework for submission. Then, by Friday I will have a substantial framework prepared for my blog post as well as enough progress on the project itself to move on to the next piece over the course of next week.

The first section of the blog I worked on was a general overview of the background of the project so readers can easily assess if this post is relevant to their own work. I then went on to discuss getting started within the project. I used elements of the how to guide from assignment 1 to support the blog post, i.e operating google colab, creating a Kaggle dataset. With regards to fastAi, I included all relevant sign posts to the information and course available to gain an in-depth knowledge of the library, however, I also created a brief explanation of the key characteristics and terminology of the library. I felt this was useful as it is not intended to devalue or override the fastAi course but compliment it and offer a brief introduction for those with a background in machine learning and fastAi, and also give a brief explanation for those whose primary focus is to simply use the project rather than focus on implementation.

I then went on to give an overview of datasets. Unlike the how to guide, I gave a brief description of the three main methods of obtaining a dataset and how to implement them within fastAi. Whilst in the latter stages of my work I have primarily worked with Kaggle datasets, which I make clear within the blog, I feel by showing the range of methods available, readers will take the work and expand upon it with these implementations.

17/02/21

Today I continued on with the blog post, I detailed what area of machine learning I was focusing on and why I felt it was a relevant topic to work on. I

then began to provide a brief overview of the structure of audio recognition and the best way to address it within fastAi through the librosa library. I then extended this to discuss the fastAudio module and how it compliments the fastAi library. Finally, I documented the stages and changes I made when adjusting the ESC50 identifier to work with another dataset and the importance of verifying that the models are flexible and not dependent on a single data set.

19/02/21

As well as my diary, for this meeting I am submitting the code I have built for running predictions and the code for generating the models. When using the model generator, it's important to use the set_a_adjusted.csv file, this is because the original from the dataset contains the unlabelled test data which throws an error within the data loader because there is no value within the get_y column.

From the meeting we discussed how a series of self contained programs may be more beneficial than an integrated model relying on tabular and vision based imports. For the remainder of the day I went about converting my vision model to a tabular model and have so far succeeded in the following:

- Converting audio datasets to a csv file
- Loading the labels and data into a dataframe
- Creating the data loader
- Defining the learner.

I have currently hit a stumbling block with regards training the model and have asked John for assistance. Hopefully by Monday the model will be fully functioning.

21/02/21

I was able to solve the initial problem with the notebook. The default batch size was 1024 from the sample. However, my total data set was significantly smaller therefore, an empty batch was being pushed through the learner thus generating a NaN error. However, whilst this issue has been resolved, the model is now stating one of the data fields contains a character value as opposed to an integer. After running the dtypes function to check each of the columns, they all appear to be the correct data type.

CSC1028 Project Diary – Kaylem McShane

I also spent some time today implementing some of the change that were suggested about the how to guide. I have:

- Fixed typos and included links to the colab site
- Elaborated on the imports within the model
- Created an introductory paragraph to highlight the audience which will find the guide helpful.
- Elaborated on what a random forest is and how it is created.

I spent some time trying to solve the issue with the learner. What I found is the loss function I was using, MSE, is only applicable to regression models. Classification models such as the one I am creating require the use of the flattened cross entropy loss. However, when trying to implement the cross entropy loss function a CUDA error is invoked.

22/02/21

Firstly today, I updated the how to guide in preparation for the assessment 2 submission. I fixed several typos, included more information on random forests for readers who do not have an understanding of the method, included explanations of the imports that are used and finally, I included a paragraph in the introduction to highlight the audience the guide is for. Within the feedback John mentioned the report like structure of my guide. Personally, I find this structure very useful as it is structured in such a way that a reader can jump straight in and find the section most relevant to them, or just read through the guide in its entirety. This tailors the guide for more experienced users looking a quick solution, as well as for newer users experiencing these issues for the first time with hopes of understanding and resolving them.

After updating the how to guide, I returned to the tabular model I was training. By adjusting the learner to its default state and only passing the data loader and metrics, I was able to get the model running on the heart beat dataset. It became clear quite quickly that this method wasn't as effective as the melSpectrogram approach. The initial training cycle had an accuracy of 16% which increased to 35% after using the learning rate finder. In comparison, the initial model had an accuracy of 80%.

I then went on to try and generalise the model further, this time using the cats and dogs dataset. This model was slightly more promising with an accuracy of 69%. In comparison to the equivalent cnn model, this is still poor as it achieved a 98% accuracy rating on the same dataset.

CSC1028 Project Diary – Kaylem McShane

I have found that the tabular model doesn't provide the same flexibility with regards to naming conventions within the dataset as the image based system. In order for the model to train correctly a csv file with the columns filename and label must be passed to ensure it goes through the system successfully. I have also encountered an issue where the first data item in the data frame is incomplete. This results in the data frame being unable to pass to the data loader and also in one of the attributes being stored as a string rather than a float. For now a simple cast function can be used but I will aim to remove the error fully to avoid relying on casting.

I am running the ESC50 dataset on the model. This is much larger than the other 2 therefore I hope that by increasing the batch size I can improve the overall system accuracy. As the dataset has over 2000 items it is taking a significant length of time to generate the csv file containing all the feature data. Unlike the audio to melSpectrogram function which can be done as a batch transform on the CPU, this transformation is linear on the CPU, meaning that the wait is significantly longer.

The model has just completed training and whilst it is a success with regards to functionality, it's a failure for usability as on this particular dataset I am achieving an accuracy of 3% at best. I may have to look into the effect of batch size but also investigate the usefulness of my chosen parameters.

After going through a selection of Kaggle datasets, I found that the parameters I am using are commonly used and have been implemented in existing datasets such as this: https://www.kaggle.com/harish24/music-genre-classification

On my model the accuracy was 13.5%, therefore I may need to adopt a new method for pre-processing the data or splitting the sets. I intend to go through existing examples using the above data set and compare the performance of these systems.

This evening I constructed a first draft of the blog post html files for next Friday's submission. I've given it to a friend on the Software engineering course to proofread and give is thoughts on how easy it is to follow and if it is accessible to those with no prior knowledge to artificial intelligence or fastAi.

CSC1028 Project Diary – Kaylem McShane

23/02/21

I contacted John today about next steps regarding the project and moving beyond the tabular learner. He suggested that I contact Jamie Gardner, to allow me to begin moving towards the projects goals of deliverable notebooks within an application format.

Jamie got in contact with me and suggested storing any notes/logs within a wiki or notion page to reproduce steps in the process. I have also gave him collaborative access to the github, to enable him to make notes within the repo wiki on the processes being employed.

I then spoke to John who outlined the overall aims of the project moving forward with my first step focusing on identifying and creating a 'snapshot' of the libraries being employed within the models, to ensure they remain functional years down the line.

26/02/21

After the meeting with John today it was decided to begin working on the deployment of the model. The first port of call was to download the instance of the libraries the code is currently running on. From this, a command line script can be used to initialise the device the app is running on, even if no internet access is available.

I asked John for some resources to build a foundation in JSON knowledge as I have no experience whatsoever I their use. They are very useful and easily implemented and for certain areas of the deployment I can see how they can be initialised but with regards to deploying the library 'snapshot', I'm struggling to find an application but I'm going to keep looking for now.

Been reading into JSON files more, still no further on with directories but I have come across several references of serialising audio data so that it may be stored within a JSON file. Furthermore, pandas have a built-in function that allows for the conversion of a panda's data frame to a JSON file. This means, in theory I can convert the datasets that have been used within my examples into JSON files for deployment.

CSC1028 Project Diary – Kaylem McShane

01/03/21

I spent a good deal of the weekend focusing on familiarising myself with JSON files and their structure by using the guides John sent me. I was also furthering the development of the 2nd assessment and have completed a significant portion of it which I will detail as part of the submission

I contacted John this morning as I was struggling on preparing the python libraries for deployment. He mentioned venv, virtual environments within python. This will enable me to install the required libraries locally on my device within a virtual environment which can then be exported as a requirements text file. This text file can then be passed to the system/notebook on initialisation to install the necessary libraries and initialise the environment for use. Currently, my system is throwing errors due to the SSL not being certified and Git not installing correctly, therefore for this week my main plan is to create a virtual environment for the project and to finalise assessment 2 for submission.

I was able to resolve the SSL issue by manually installing the latest SSL version onto my device which then enabled the command line to carry out pip installs.

02/03/21

Whilst the SSL issue was resolved locally, within the venv virtual environment it still persisted. I tried to get around the issue by using the Computer Science buildings virtual computers. However, due to administrative restrictions it would not let me directly download the relevant libraries to the virtual machines therefore I must work locally.

I have been able to generate a requirements txt file containing all the relevant libraries and their versions. This was done by using the pip freeze>requirements.txt function which I applied within one of my functioning google colab notebooks. By doing this I ensured that all version numbers are consistent with what I had been initially running.

An issue that arose within the library was that the cupy-cuda library was not available. When trying to run it, it was not available on my machine as I had a NVIDIA framework pre-downloaded as part of the drivers in an RTX2070. After uninstalling the framework and replacing it with Cuda 11.2, I found the system was still not taking to the library.

CSC1028 Project Diary – Kaylem McShane

Assessment 2

When constructing the blog post for the project, I broke it down into 4 key sections. This will enable readers to jump into whatever section they feel is beneficial at their stage of development as opposed to a single script of text which would take considerable time to traverse.

Section 1 is a simple about section. It introduces the reader to the overall concept of the project and the areas I had been working on. It goes onto to reference the google colab notebook editor and I took this opportunity to utilise the how to guide from assessment 1 and provided a link to it, allowing the reader to improve their knowledge of cola before proceeding any further.

Section 2 is the 'Getting Started' section. Firstly, I introduce the fastai library and encourage the reader to go through the fastai course before delving into the project itself. I do then go on to provide an overview of the key ML concepts relating to my project throughout the section. I felt it was important to provide access to the fastai course as it had been assembled by the developers of library itself, superseding any explanation I could provide with my limited expertise. However, I still provide an overview of key concepts to provide a brief refresher for those with experience in fastai and machine learning but who are just out of practice, and to those who wish to simply want an overview of the project. Whilst the guide can expand the readers knowledge and help them adjust the project and expand upon the created model, I ensure enough information is provided to ensure the blog makes sense. Moving on from fastai, I go on to briefly explain the format of datasets in the context of machine learning. I provide an overview of the three main methods using the bing search API, Kaggle and the untar method. Whilst for the project I focused mainly on Kaggle and the bing api is only applicable for image-based datasets, I felt it was important to cover all the key methods to provide readers with the flexibility to explore and expand on my approach in the future.

Section 3 then began talking about the implementation of the project itself. I begin with an overview of analysing audio data using melSpectrograms and a CNN learner. I then went onto discuss the deconstruction of an existing model to allow it to run on a different dataset and then how this was further generalised to work on a range of datasets. This was an important aspect of the blog post as it just did not show the end result of the project but also the

methodology, allowing the reader to see the skills that are deployed in the development of a system such as this. Using a similar format, I go on to detail the construction of a prediction model for the CNN and how I approached the task from a different point of view by using a tabular learner.

The final section provides an overview of the next steps for the project, recognising the limitations of the blog post as it is not yet completed, but providing enough of a framework for another developer to pick up the project and begin working.

I updated the how to guide in preparation for the assessment 2 submission. I fixed several typos, included more information on random forests for readers who do not understand the method, included explanations of the imports that are used and finally, I included a paragraph in the introduction to highlight the audience the guide is for. Within the feedback John mentioned the report like structure of my guide. Personally, I find this structure very useful as it is structured in such a way that a reader can jump straight in and find the section most relevant to them, or just read through the guide in its entirety. This tailors the guide for more experienced users looking a quick solution, as well as for newer users experiencing these issues for the first time with hopes of understanding and resolving them.

The final piece of the assessment is my source code which is available from the following git repository: https://github.com/kmcshane811/AutoML This contains tabular and CNN classifiers as well as the CNN prediction model. I also provide all the adjusted csv files for the datasets so that the user can run the code without the need for any complicated adjustments.

05/03/21

After the meeting with John today, we defined a specific outline for the deployment of the project and what the application will contain and consist off. Due to the recurring error within the venv, I decided it would be a practical use of my time to locally download the dependent libraries in order too ensure the installation was running correctly. Out of the 394 installs only 23 where incomplete due to the repository not being available to the pip installer. I am unsure as to what has caused this issue, but I intend to research it over the weekend and try and find a solution.

CSC1028 Project Diary – Kaylem McShane

08/03/21

Over the weekend I attempted to fix the issue in a number of ways:

- I updated my pip to the latest version
- I updated python to the latest version
- I tried to install the libraries directly from a wheel file
- I enlisted Jamie to download the requirements on a mac system.

This showed me that the issue is not related to the python or pip versions but more likely to the windows 10 OS as Jamie was able to download the requirements on macOS. When downloading the wheel file, the following error was produced: 'ERROR: ideep4py-2.0.0.post3-cp27-cp27m-manylinux1_x86_64.whl is not a supported wheel on this platform.'

After speaking with both Jamie and John, they both recommended I make use of the WSL(Windows Subsystem for Linux) as it allows for more flexibility when installing packages. Jamie sent me the installation guide which can be found here: https://docs.microsoft.com/en-us/windows/wsl/install-win10#manual-installation-steps , and after going through this tutorial I was able to generate a functioning ubuntu shell for running commands. The required packages then began to run as required, as did the venv, resolving the issue from the windows command line.

The next step of the project will be to run training code through the python shell using the venv.

09/03/21

This morning I reworked the CatDogAudioClassifier into a python shell document so it could be run locally from the command line within the venv. I decided that this code would be the most suitable for this test as I know for certain it works and it has a fixed file path for the datasets. When I tried to run the document however, an error occurred stating that the necessary NVIDIA drivers were not installed. I then went on to use the following guide to install cuda on ubuntu.

https://medium.com/@stephengregory_69986/installing-cuda-10-1-on-ubuntu-20-04-e562a5e724a0

Once the cuda drivers were installed, I resolved a few minor bugs in the code relating to file paths and this allowed the training model to run

successfully from the shell. I then spoke to John about next steps and it as agreed the best starting point would be to implement the discussed directories and also explore the development of use cases for non-experienced users in the form of exe files. I began looking into generating python executables and found pyinstaller to be the best library for this. After installing it and running the commands on ubuntu I experienced an error due to the shell not having the correct permissions to files.

10/03/21

I began today by focusing on the pyinstaller error. I found a range of attempted solutions such as manually changing access rights, using the chmod function and cloning directories however the solution itself was much more straightforward, when running Ubuntu run as administrator. My main aim now is to generate the exe files on the linux system and try make the code function correctly on windows.

I have been able to generate the venv in windows and run the code, however, an error is occurring due to an all zero input from one of the conversions. I find this confusing as the parameters and input for this is the same as the linux version.

I discovered the issue was caused by fastai using multiple 'workers' which is not supported in windows. In order to rectify this issue, I added the parameter num_workers = 0 to the dataloader.

12/03/21

Now I have both the windows and linux shells running fastai code, my next priority is too create .exe files that allow the code to be automatically compiled within the python shell.

So far, I have discovered my python environment didn't install correctly and is missing library folders necessary to support the pyinstaller package, I tried repairing the environment within the installation exe but the issue is still persisting.

I have been able to successfully compile and run the training file on Ubuntu and the linuxOS using pyinstaller, once I resolve the issues with the windows python libraries I believe it will be up and running not long after.

CSC1028 Project Diary – Kaylem McShane

After meeting with John, we discussed the end goals of the project in the context of the module. We discussed the need to fork the relevant repos from GitHub into the autoMl git for submission. This will ensure any changes from the fastai community will not impact the integrity of the project and ensure it is robust for future use. We discussed the usefulness of having packaged and executable python files for locally training python file and how to expand upon this to improve user friendliness in the form of use cases. We also commented on the importance of laying out a clear plan for the project, beyond the time frame of the module, as well as leaving the model in a state that can be easily picked up and modified by future developers in order to progress our work.

After trying to resolve the windows issue, I discovered that code compiled within ubuntu can still be executed within the windows environment and executed within windows. The next step will be to convert the generalised version of the notebook into an executable file. This will allow separate directories to be used and multiple datasets, providing further flexibility with the programme.

At the end of today I have now created a functioning model that can run with any dataset on a local python shell given that the dataset directory contains a csv of the filenames and the audio files.

15/03/21

Today I converted the other models I had made into compiled code. The project now contains a generalised audio classifier trained on a CNN and a tabular model, as well as a data conversion that generates a csv file of audio metadata and finally a prediction model that can pull an exported trained model from a directory and run predictions on audio files placed within the directory. By separating the data preparation for the tabular model, the tabular learner now has the flexibility to be adjusted for standard tabular classifications as well as audio classifications.

16/03/21

Today was incredibly beneficial in the overall deployment of the project. Firstly, I created a JSON path generator which, on execution will create a JSON file that finds and stores the absolute file paths of the key directories and the venv used in the execution of the project.

Secondly, I then went on to adjust each of the programs I had created so that they relied on the JSON as opposed to the package locations I had initially used. This allowed me to package the datasets into separate directories, making it easier for the user to prepare experiments. I was also able to store the exported .pkl file in the relevant directory, improving the operability of the system.

Thirdly, after some research into executing python scripts within venv's, I found the best practice was to create batch files that activate the venv with each program and then execute the program within the venv's python.exe. I automated this process by creating a python script that wrote the batch files and stored them within an experiment folder.

Finally, after creating all the necessary building blocks, I created a basic prototype within a tkinter GUI which enables the user to run the programs at the push of a button, within the venv. The only pre-requisites are to run the json generator and batch generator before the gui, but neither file are dependent on the venv, allowing them to be double clicked, thus reducing the complexity for the end user.

19/03/2021

Assessment 3

- Improve how to guide
- Improve blog post
- Test cases
- Research social media outlets
- Create social media post
- Detail how to expand upon the project
- Diary submission

Today's meeting was focused solely on the final submission of the project and outlining what is expected for the submission. Currently I have a basic prototype that can execute batch files stored within a separate folder. This doesn't fully meet the criteria outlined by John, but it is on the right track and I intend to take one final week to work towards the end product before focusing in on the final submission over the easter break.

For now, me, Mark and Nathan are planning to have a call where I'm going to demonstrate my deployment of the code locally within a virtual environment

in order for us to have a standardised output. I am currently uploading the project to drive for it to be shareable to them, to give access to functioning source code.

After the meeting with Mark and Nathan, I took the time to show them how my code runs, how to create, activate and install to a venv and how to create a python script that creates batch files for compiled python files. Nathan commented on how my system for inputting datasets and predictions is quite inefficient which I would definitely agree with. We all agreed it could be streamlined to assist the user in inputting the data but due to the time constraints of the project it would be unrealistic to suggest that this could be improved upon in this time frame. We all also agreed to complete the test cases of each other's projects in order to provide a minimum of two test reviews.

I began to consider possible sources to publish my social media post, the initial stand out was r/machine learning, a sub reddit dedicated solely to machine learning. Due to the heavy reliance on fastAi and fastAudio I also considered the fastAi forum. Finally, within the Kaggle discussion page there is a thread dedicated to autoML which is the direct topic of the project which coud provide valuable insight. Over the next few weeks I will keep reading through the forums and see what the best outlet would be.

I've made a copy of the code I have been working available on google drive and shared links with mark, Nathan, John and Jamie to both aid and get advice on how to implement the project and whether or not this is the best practice to follow.

21/03/21

Due to the size of the file, John suggested stripping out as many redundant files as possible to reduce the size of the file. I discovered the dist and build folders where incredibly bloated and were not accessed directly by the code, therefore I deleted them and found that the file size had reduced from approximately 15.5GB to 1.69GB, an 89% reduction with no discernible loss in quality.

CSC1028 Project Diary – Kaylem McShane

22/03/21

After minimising the file size, I shared it with John who made the following comments about what to change and amend in order to keep in line with the overall needs of the project:

1. Have experiments, datasets, training, trained all root level folders

2. You have 2 models so you should have two training folders

3. The trained models should be self-contained folders with the code to run the models (so you can choose to just download a trained model and it will work without the others being present)

4. Ideally trained models can be run either as a flask webserver with rest api or as a command line program that processes a folder of input data

5. Trained models should output json files as their results data

6. The experiment folder should have a json with a specification of hyper parameters and the paths to the training and test datasets

7. The trained model should also contain performance results from the experiment

8. Don't have any code outside of the folders except to provide a gui that makes it easier to run the experiments, alter/generate the experiment folder/json

9. Experiments should run with one click, there shouldn't be multiple stages to trigger training e.g. the audio/tabular case.

At a glance the easiest case to resolve is case 9, as all it requires is for 2 of the programmes to be integrated into a single file which was the first change I implemented, enabling me to delete the audio_tabular_prep file.

The next case which was feasibly fixed without much modification was the storage of results from a trained model within a JSON file in the deployment directory.

Due to the time limitations of the module and my lack of experience with flask webservers, I believe that case 4 is not fully feasible in the time constraints of

the module but as the files can be ran within the command line, I believe it is suitable for now.

23/03/21

After fixing some of the more relatively straightforward issues yesterday, I am planning to move onto the bigger upheaval of implementing the directories, json files and the relative file paths. For the ease of developing the program I have placed all the directories within a single parent folder, but it is designed in such a way that if all the directories are located within the same parent folder, they will be operational.

Firstly, I constructed the directories themselves and copied across the relevant source files from the developed prototype to begin the implementation. I also created a simple python script to generate the JSON files. This is not going to be published within the deployment but was only used for individual use as I find it easier to writ JSONs within python as opposed to notepad. The same premise is applied to a python script to generate the batch files.

Once the structure of the program was developed, I worked on implementing the trained model which can be used to run predictions. John requested that it can be ran either as a command line programme which takes a file location as input, runs the predictions, and returns them in the form of a json file within another user requested folder, or a flask web server which allows for individual file uploads for predictions. Due to my lack of experience with Flask and the time constraints with the project, my aim is to focus solely on command line to create a functioning model which can then be expanded upon for a flask model and docker image.

I then went on to convert the cnn training model into an acceptable format for this structure. Relatively speaking this was a straightforward process with the only real challenge being to convert the relative file path to an absolute file path in order to access the dataset and create the data frame.

I converted the tabular model to work with the relative file paths. This was slightly more complex due to having to read and write to a csv for the meta data files but beyond having to be more cautious of what directory I was operating within this was a straightforward process.

I reworked the GUI to run outside of the directories, ensuring each directory is self-contained within the system.

CSC1028 Project Diary – Kaylem McShane

24/03/21

Today I focused on recording the results of the experiment and storing them in a JSON accessible from trained folder, in order to do this I relied on the callbacks built into fast ai learners, a class used for recording values and dynamically changing learner parameters. In this instance I used the csv logger callback to generate a csv file containing all measured metrics, this was then converted into a pandas dataframe and appended to the json file.

25/03/21

John contacted me today about my program not running on his device, indicating an issue with deployment. He said it was due to me not running the python scripts through the venv python.exe therefore I changed the execution location within the batch file using:

..\auto-ml-env\Scripts\python.exe

This allowed the training modules to run with no issue. However, an error did occur within the predict_cnn file as the import ipywidgets could not be found, despite this being installed within the venv. After looking into the issue, I found that this import was not critical to the execution of the program and removed it, allowing the code to run without fault as before.

26/03/21

This morning I tried running my program on the EEECS virtual machines and encountered the following error:

No Python at 'C:\Users\mcsha\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.python.3.8…'

This was confusing as I had used no absolute file paths within any of my code. However, after some research into venv's, I found that they are not supported to be copy and pasted between devices due to absolute file paths being automatically added. The solution to this is to create a requirements.txt file and run a batch file that creates a venv on the user's local device. Too ensure the system remains robust I will use the forked git repositories as opposed to the published versions.

During the meeting with John he made a series of recommendations on possible solutions and also put me in contact with Ben, another student who has encountered similar issues in the past. Whilst I wait for Ben's reply, I'm currently using a pyinstaller based solution as I already have used his package.

CSC1028 Project Diary – Kaylem McShane

By using the following code:

pyinstaller --onefile --paths .\auto-ml-env\Lib\site-packages RUN_ME.py

I compile the python code whilst specifying which dependencies are required to run the file.

3 hours later: The aforementioned method didn't work so I took a different route. Within the RUN_ME.py file I included a script to check if the venv exists within the directory. If it does not, it creates the venv and installs all necessary dependencies within the venv and then allows the program to run as intended with the batch files and relative file paths. This is beneficial because it allows the project to be exported without the venv, minimising the file size thus making it more portable.

29/03/01

For the remainder of the module, I am focused on preparing my work for assessment 3 and creating a deployable piece of work that can be operated and expanded upon by future users. To work towards this goal, I implemented several validation and verification techniques within my code, most notably try, except methods too ensure that any errors are caught without crashing the program. I also added a null check on the prediction too ensure the user does not push a null path. I then went on to comment the code to give brief descriptions of what each library and import is designed to do, as well as any functions and declarations I have made which are not inherently clear.

After cleaning up the code and ensuring it is robust and self-contained, I began the compilation process using pyinstaller to generate the .pyc files which can be executed within the command line. I have added a source file to the main directory to provide an open-source model of my program and when releasing it with the blog post I will include a compiled version with no source code and an open-source model which can be adjusted as needed.

I began creating my test plan for the project today, consisting of 14 test cases which covers all the functionality and possible faults across the scope of the project too ensure the system is sustainable and suffers no critical failures. I am hoping to have 4 tests conducted by 4 students across EEECS, Nathan and Mark have agreed to help and 2 software engineering students I am friends with have also agreed to carry out the tests.

CSC1028 Project Diary – Kaylem McShane

30/03/21

Today I spent a considerable amount of time working on the blog post and the local deployment of the project. John contacted me about not being able to run the project on his device due to differences with python versions so from tomorrow I am planning to research the packaging and creation of exe files that store all required dependencies for the project, removing the need for python entirely.

31/03/21

The aim of today was to package the project as .exe files as opposed to .pyc files. The fundamental issue I faced was the compatibility issues between pyinstaller and matplotlib as changes within the library limited the ability of pyinstaller to package. This problem is quite unique, and I haven't found any definite solutions yet but have received support from John and contacted Ben who recommended manually passing the data files using the –add-data function.

I've been working on this problem over the past 2 days and feel as if I have gotten no where and that I am going in circles. Due to the upcoming assessment 3 deadline, I feel it is more suitable for me to dedicate more time to creating a deliverable project that can be ran in a limited capacity and expanded upon by others than to risk generating something that doesn't work at all, limiting the usefulness of the code. I will continue to work on this problem in the background and hopefully find a solution soon.

12/04/21

Due to the easter break I took some time away from the project to spend with family and today is my first day back. Before the break, I had continued the next section of my blog post, constructed a test plan and made the test programme available via google drive and also was left with a fully working prototype. Today I intend to take the time to read through some online forums to identify a suitable location to publish my social media post and make the project available and I have also organised a meeting with Jack and Adrian to discuss how they found the testing of the system.

As previously mentioned, I had published a post within the Kaggle autoMl forum and since that post I have had little to no response from the community, therefore I feel posting the guide there would have little to no impact.

CSC1028 Project Diary – Kaylem McShane

I turned my attention to reddit and began looking into r/MachineLearning, a subreddit dedicated to machine learning with 1.8m members. This subreddit is divided under projects, research, discussion, and news threads. Due to the publishing of other projects, I felt this subreddit would be suitable. However, after further inspection I found that it was best suited more for finished projects and that there was no demand for guides or project documentation, only the finished deliverable. After reading around the subreddit, I found the following listed within the rules:

"Beginner's tutorials and projects go elsewhere

Beginner's guides, tutorials, and projects should be posted to /r/LearnMachineLearning"

This then directed me to the beginner thread r/LearnMachineLearning which is subdivided into the following threads: Tutorial, Question, Help, Request, Project, Discussion.

As the project is not yet finished in its entirety, I feel it may be suitable to publish the blogpost as a tutorial alongside the source code of the prototype. This can then act as a guide for beginners to familiarise themselves with the building blocks of machine learning whilst also inviting the user to experiment with the code and expand the project beyond what I have achieved throughout the course of this module, I will read into several articles too ensure this is a valid option.

Within the tutorial thread of the subreddit, I found a range of useful guides and tutorials but the overarching theme was that they were short, succinct and quite niche in their content such as this guide focused on the core mathematics of machine learning: https://www.reddit.com/r/learnmachinelearning/comments/moty4d/math_you_need_to_succeed_in_ml_interviews/

At the time of writing this piece received 11 comments in 17 hours, some offered praise, some pointed out errors within the information provided thus giving the publisher a chance to update the guide and some also provided links to other resources that could be used in conjunction with the resource to further expand the knowledge of both the publisher and any recommended parties. I also came across some more advanced tutorials such as the application of machine learning within cyber security which I feel could be classified as a larger field such as Auto Machine Learning. Furthermore, with

many guides dedicated to learning the fundamentals of machine learning, the blog post offers a guide on the fundamentals of fastAi and how it can be incorporated into the auto machine learning process thus making this a useful guide for this particular thread.

Jack and Adrian both completed the test case, and the results were interesting. Jack runs a Nvidia RTX 2070 super GPU, same as me, whilst Adrian runs the older GTX1660TI GPU. Newer RTX models do not have cuda installed whereas the older GTX models do so the code was executed successfully by Adrian and unsuccessfully by Jack. This indicates that the end user must have cuda installed for the programme to run in its current state. This issue is one I faced and to resolve it I had to uninstall the current frameworks and install the older CUDA system. At this present moment I am unaware of a method to resolve this without the end user being required to make drastic changes.

13/04/21

I took some time to do to browse through reddit and read a series of tutorials and guides to help better my own how to guide. I found a range of sources that provided me with enough direction to make the relevant changes I feel will improve the overall appeal of my guide. These changes were detailed and will be put with my full assessment 3 submission hence the lack of detail here.

19/04/21

I returned to the project today after dedicating a significant amount of time to my other 2 modules. This has not impeded my process on assessment 3 at all as I have just completed my blog post, which similarly to the how to guide I will discuss in more detail in a formal diary entry for assessment 3. My aim for tomorrow now the blog is done is to construct a social media post to be posted on r/learnMachineLearning. Once I have met all the requirements for assessment 3 and I'm satisfied with my work I aim to return to the project and continue on the conversion to exe files.

I updated the github to contain the local deployment files and the commented code, making it publically available alongside the test plan

CSC1028 Project Diary – Kaylem McShane

Assessment 3 Diary

Blog Post

From Assessment 2, I have modified the blog post by expanding on the content of the original submission whilst maintaining the same structure.

Firstly, I removed the ‘Next Steps’ section as many of the mentioned aims had been achieved throughout my own development of the project and was no longer relevant to the blog, i.e., creating a self-contained python environment in the form of the venv’s and creating a local directory structure that enables the code to be ran flexibly on local machines whilst training multiple models.

Secondly, I replaced the ‘Future Development’ section with the ‘Deployment’ section of the blog. This is dedicated to showing the conversion from Colab notebooks to .py files, the creation of the virtual environments and the relevant local libraries and finally the compilation from .py files to .pyc files. This section is useful for the reader as it acts as a stand alone guide for converting fastai .ipynb files to local .py files with the relevant libraries that are required which is applicable across a range of systems and not just this particular project.

Thirdly, I added a next steps section to the blog which details the future development of the system and the next steps to be undertaken by anyone who wishes to further the project in their own time. I structured this in a cascading manner as each milestone should lead to the next. The first requirement is the conversion from .pyc to .exe files to remove the need for a python executable environment entirely. The next step is the deployment of the application as a docker GPU image which is deployable as a flask webserver, enabling all interested users to access the system regardless of hardware limitations. The final step is the deployment of the system through ONNX, making the system accessible to C++ applications and therefore improving the range of uses that are applicable to the system.

Finally, I added a link within the blog post to allow the reader to download the project, containing the test plan directly from the blog, removing the need to access the github repository directly thus improving the ease of access for the end user.

CSC1028 Project Diary – Kaylem McShane

How to guide improvements

After reading through a range of tutorials from r/LearnMachineLearning:

https://blog.paperspace.com/seq-to-seq-attention-mechanism-keras/

https://arcalan.medium.com/transfer-learning-in-deep-learning-using-tensorflow-2-0-f30af2162fd

https://blog.paperspace.com/neural-network-pruning-explained/

I found several differences between these will received guides and my own how to guide.

Firstly, the guide should be succinct and direct therefore they are contained within a single static html page and not several linked pages. This localises all the information in a single location and makes it easier for the reader to traverse between sections of the guide without having to change location.

Secondly, whilst the aim of the guide is to inform and teach the reader, it should not be a detailed summary of the entire topic. In the context of my how to guide, the aim is too teach the reader how to set-up a colab notebook and debug the tabular example from the fastai course. Therefore, it is reasonable to assume that the reader is aware of fastai and understands what it is and some of the core fundamentals.

A similarity I observed between my own work and the other guides was the use of imagery without to support and reinforce the guide, indicating that this is a good stimulus method to help engage the reader.

For the final submission I intend to compress the guide into a singular static HTML page as well as removing the FastAi section as I feel it is unnecessary for the readers that this guide is intended to assist. As a reminder, this guide has been designed for beginners of the FastA.i library to initialise and begin using the google colab notebook whilst detailing how to debug some common problems within the example notebook for tabular learners. I also intend to reduce the content of the introduction to be more focused on the issue it is trying to resolve and less focused around the module and my aims as a student. I also made a few adjustments to the physical structure to ensure blocks of text remained within a structured format and were not repositioned or truncated by the use of any images.

CSC1028 Project Diary – Kaylem McShane

Social Media Post

I had published a post within the Kaggle autoMl forum and since that post I have had little to no response from the community, therefore I feel posting the guide there would have little to no impact.

I turned my attention to reddit and began looking into r/MachineLearning, a subreddit dedicated to machine learning with 1.8m members. This subreddit is divided under projects, research, discussion, and news threads. Due to the publishing of other projects, I felt this subreddit would be suitable. However, after further inspection I found that it was best suited more for finished projects and that there was no demand for guides or project documentation, only the finished deliverable. After reading around the subreddit, I found the following listed within the rules:

"Beginner's tutorials and projects go elsewhere

Beginner's guides, tutorials, and projects should be posted to /r/LearnMachineLearning"

This then directed me to the beginner thread r/LearnMachineLearning which is subdivided into the following threads: Tutorial, Question, Help, Request, Project, Discussion.

As the project is not yet finished in its entirety, I feel it may be suitable to publish the blogpost as a tutorial alongside the source code of the prototype. This can then act as a guide for beginners to familiarise themselves with the building blocks of machine learning whilst also inviting the user to experiment with the code and expand the project beyond what I have achieved throughout the course of this module, I will read into several articles too ensure this is a valid option.

Within the tutorial thread of the subreddit, I found a range of useful guides and tutorials but the overarching theme was that they were short, succinct and quite niche in their content such as this guide focused on the core mathematics of machine learning: https://www.reddit.com/r/learnmachinelearning/comments/moty4d/math_you_need_to_succeed_in_ml_interviews/

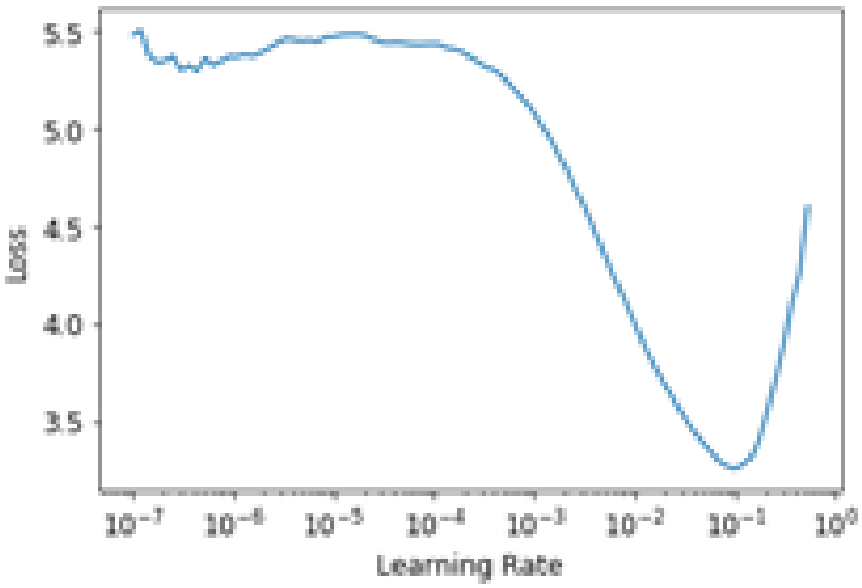
CSC1028 Project Diary – Kaylem McShane

At the time of writing this piece received 11 comments in 17 hours, some offered praise, some pointed out errors within the information provided thus giving the publisher a chance to update the guide and some also provided links to other resources that could be used in conjunction with the resource to further expand the knowledge of both the publisher and any recommended parties. I also came across some more advanced tutorials such as the application of machine learning within cyber security which I feel could be classified as a larger field such as Auto Machine Learning. Furthermore, with many guides dedicated to learning the fundamentals of machine learning, the blog post offers a guide on the fundamentals of fastAi and how it can be incorporated into the auto machine learning process thus making this a useful guide for this particular thread.

After deciding on the location of the post I then began to consider the content of the post itself. The general of structure of posts on the sub reddit seems to be a striking title followed by a brief description of the work and a link to a more in-depth description of the work.

‘Automating the Machine Learning Process

This autoMl project is designed to make the machine learning process open to individuals of all backgrounds by enabling the user to generate a range of audio classification models using the fastAi library.



CSC1028 Project Diary – Kaylem McShane

In its current form, the project can be deployed locally or within google colab with the relevant files and notepads available here:

https://github.com/kmcshane811/AutoML

The development and future for the project are detailed within the following blog post for those interested in the development of other autoMl projects and all feedback is appreciated!

https://kmcshane811.github.io

‘

I feel this post conveys the overall aim of the project to make machine learning accessible to programmers of all backgrounds and how to access the project directly for those interested in the final piece as well as the blog post for those interested in the overall development of the system.

I used the image of the learning rate curve as it shows the system’s ability to minimise loss by finding a suitable learning rate which supports the idea of having an automated system.

Link:
https://www.reddit.com/r/learnmachinelearning/comments/mur9f4/automating_the_machine_learning_process/

Test Plan

I developed a test plan consisting of 14 test cases which assessed the full scope and functionality of the project.

Test cases 1 and 2 were based around the initial boot-up of the project and how the system reacts when the venv is/isn’t present. Test case 1 demonstrates the construction of the venv within the python script and then its initialisation to run python scripts with dependencies such as fastai. Test case 2 shows how the system reacts when the venv is present. As downloading the dependencies requires an internet connection and takes some time to download everything, it would be incredibly wasteful of resources and inefficient to this each time therefore the venv should only be generated if one is not already present.

CSC1028 Project Diary – Kaylem McShane

Test cases 3-6 focus on the core functionality of the project such as training models, viewing results, and making predictions off a provided dataset. By using a provided dataset, I can verify it works before deployment and it means that any issues that occur are not within the dataset itself. It also allows for an accurate comparison of execution times between my personal device and the devices of those testing the code, allowing me to see the variance in time based on hardware specifications.

Test cases 7-11 are focused on destructive testing in which the tester tries to cause the system to crash. Due to my use of external python scripts and try except methods to catch any erroneous behaviour, I am confident that the system will be sustainable and not suffer any critical errors. However, as it is often hard to critique your own work, I am hoping that something will be highlighted that I have missed so that It can amended before being rolled out directly.

Finally, test cases 12-14 are a repeat of test cases 3-5 but with a key difference. The tester is given the requirements of a dataset and asked to fetch a new dataset from Kaggle which can be passed through the system to train a new model using both the cnn and tabular learner and predict using the CNN learner. In doing this, the system can be tested in a live scenario and can be assessed on its ability to work with a range of datasets, providing the flexibility associated with an autonomous system.

The test was carried out by Jack and Adrian and the results were varied and gave rise to a key issue that will need to be explored further. Due to the systems dependence on Cuda 10.1, any RTX GPU's will not have the required libraries preinstalled and will require the user to manually download the environment. Older GTX models on the other hand seem to run the system with no issues and allowed for all test cases to be completed.

CSC1028 Project Diary – Kaylem McShane

Thoughts on the Module

From starting programming during my GCSE's, this module was the most challenging and complex challenge I have ever undertaken. The field of Machine Learning always greatly interested me and being given the opportunity to explore it at such an early stage in my degree was a welcome challenge.

This module helped me to build upon my independent learning skills, using the guidance of the lecturer to build my knowledge as opposed to being rigidly taught from a set specification. This allowed me a greater freedom within the project to make mistakes, explore libraries I had not used before and to make informed decisions which effectively led to the developed project I have submitted.

Whilst I was not able to complete the project to the exact specification within the time constraints of the module, I'm proud of the progress I have made in such a relatively short period of time for a project of this scope and I look forward to working on the project in my spare time outside of the module.