Python Workshop Day 1 Spring

By Ethan Doan and Mai Her

Today's Agenda

1. Installing python and jupyter notebooks through anaconda
2. Python basics, data structures and dataframes
3. Object Oriented Programming
4. Machine Learning
5. Neural Networks

What is python?

- a **high-level,** interpreted, programming language
- Language with a **focus on readability**
- **widely-used** for scientific computing, data analysis, machine learning, automation, etc. but is a general purpose language.
- **Open sourced** - support from a huge community of independent and institutional developers
- Multi-paradigm - supports booth Object oriented programming and functional.

Why do engineers need Python?

Versatility
Can be used in a wide range of applications

01

Ease of use
Simple and straightforward syntax

02

Large community
Lots of available resources and help

03

Job demand
Better job prospects and earning potential

04

This you, the engineer →

Install Python Environment

Python Environment

A way to keep track of all the versions of Python packages you have installed

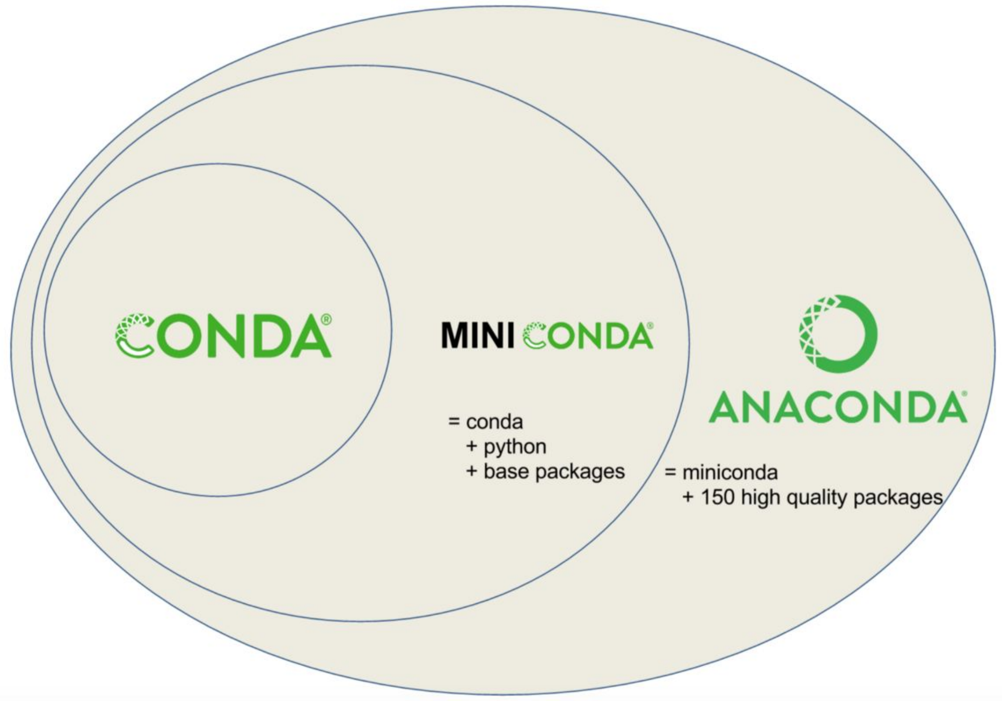
If your laptop specs are:

- Intel Core i3 7th gen and older
- Total storage less than 256 GB
- Total memory less than 12 GB

□ **Install Miniconda**
https://docs.conda.io/en/latest/miniconda.html

Otherwise,

https://www.anaconda.com/



Install Jupyter Notebook/Lab

Windows

Open Anaconda prompt

Mac and Linux

Open Terminal

Type in...

1. Create virtual environment

```
conda create --name NETS python=3.9
```

2. Activate virtual environment

```
conda activate NETS
```

3. Install Jupyter and other necessary libraries

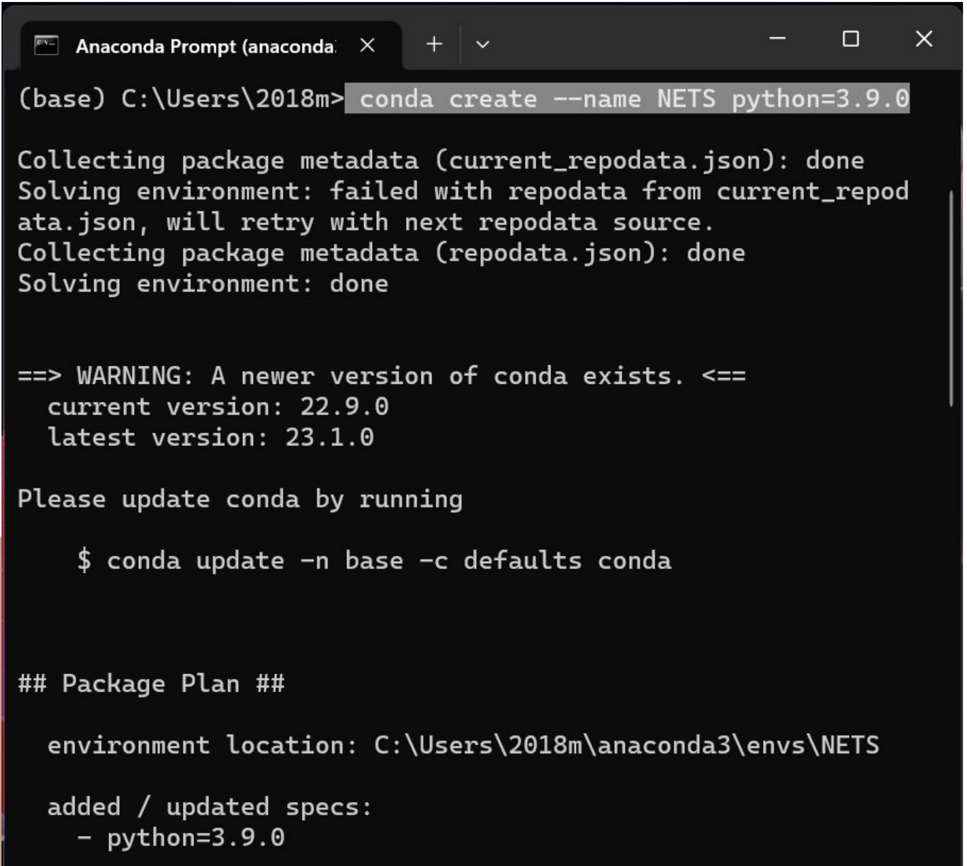
```
conda install --yes numpy matplotlib pandas jupyter seaborn scikit-learn
```

4. Open Jupyter Lab

```
jupyter lab
```

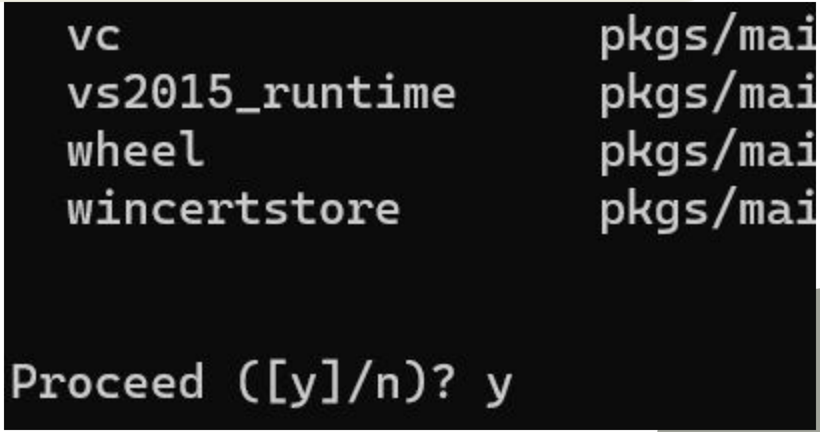
1. Create virtual environment

conda create --name NETS python =3.9



```
Anaconda Prompt (anaconda

(base) C:\Users\2018m> conda create --name NETS python=3.9.0

Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repod
ata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 22.9.0
  latest version: 23.1.0

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: C:\Users\2018m\anaconda3\envs\NETS

  added / updated specs:
    - python=3.9.0
```

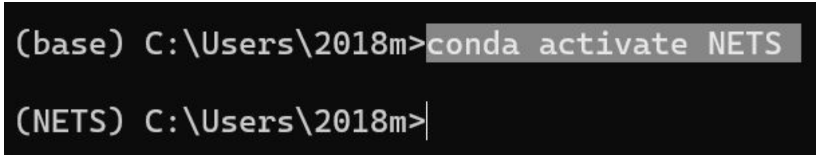
Proceed by typing ‘y’



```
  vc                 pkgs/mai
  vs2015_runtime     pkgs/mai
  wheel              pkgs/mai
  wincertstore       pkgs/mai


Proceed ([y]/n)? y
```

2. Activate virtual environment

conda activate NETS

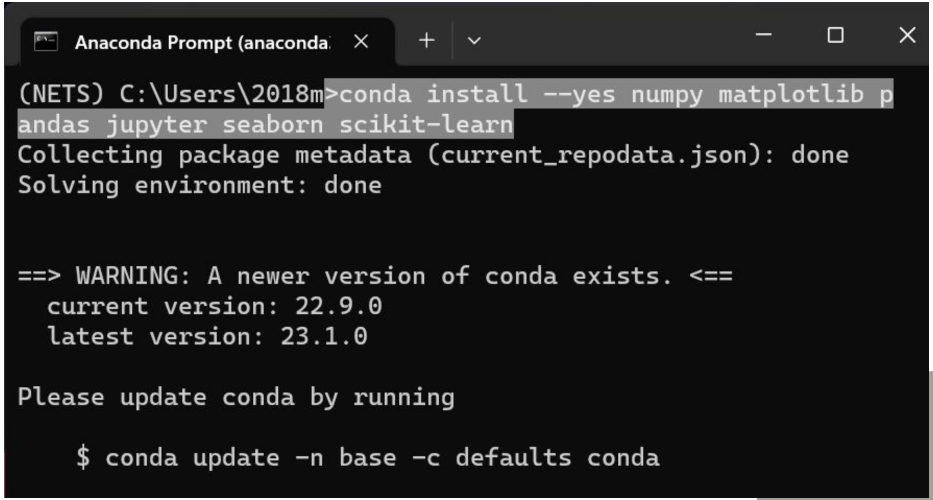


```
(base) C:\Users\2018m>conda activate NETS

(NETS) C:\Users\2018m>
```

3. Install Jupyter and other necessary libraries

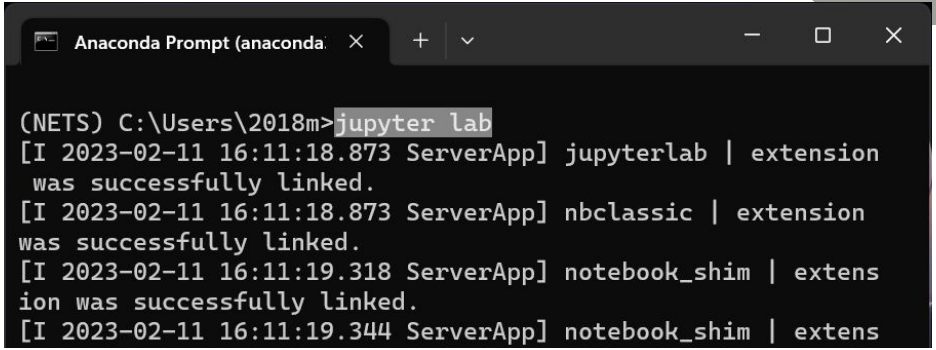
```
conda install --yes numpy matplotlib pandas jupyter seaborn scikit-learn
```

1. **Numpy**
2. **Pandas**
3. **Seaborn**
4. **Matplotlib**
5. **Math**
6. **Scikit-Learn**
7. SciPy
8. Keras
9. TensorFlow
10. PyTorch



4. Open Jupyter Lab

```
jupyter lab
```



Classes and Objects

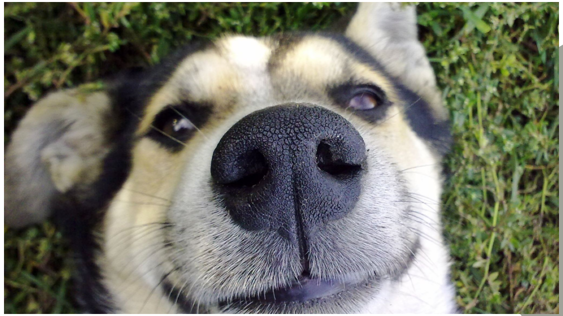
Classes are templates for an object

Objects are "instances" of those classes

Here is a bunch of objects

They are all dogs

We can define a general category for it called dog

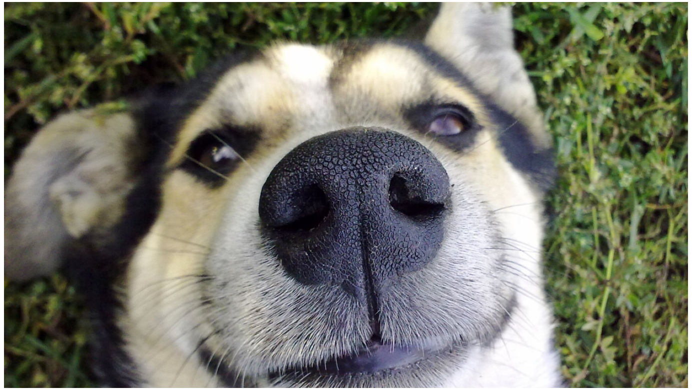


Here we have an "instance" of a single dog.

Objects have attributes

Our dog has many attributes like

Breed, fur color, age, health, name etc....



Objects have methods

A method is just an object specific function. Our dog can run, eat jump.

Making my own objects? Seems Hard.

Unless you go into hardcore software engineering, implementing objects into your code is not too important. The key takeaway is by understanding this, it will help you code better and make debugging easier.

The main takeaway from this should be understanding how to deal with custom data types, how to access attributes, and how to access methods.

New_object_variable = Object(arg1 = init1, arg2 = init2, etc…)

Variable = object.attribute

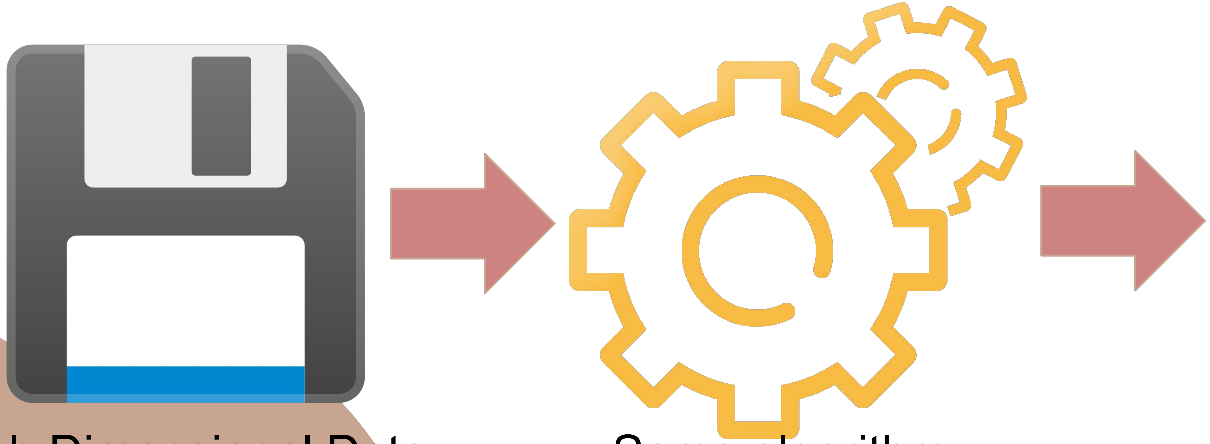
object.method()

10 minute break

Grab some food!!!

What is ML?

- Having a computer do a task without explicitly giving instructions to it.
- Instead, feeding data into an algorithm to learn how to do a task and improve doing that task gradually with more data.



High Dimensional Data

Some algorithm

Some model that predicts or classifies

Getting Data.

For a good model, you are going to need data. TONS of data.

Alone, it is hard to collect the amount of data needed for a good model.

So we mostly rely on databases which typically get data from collective groups of researchers.

Some materials research uses:

Materials Project, Aflow, OQMD

The Materials Project
materialsproject.org

AFLOW
Automatic - FLOW for Materials Discovery

OQMD
The Open Quantum Materials Database

You can have data like this....

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
2	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
3	7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968	3.2	0.68	9.8	5
4	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8	5
5	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998	3.16	0.58	9.8	6
6	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
7	7.4	0.66	0.0	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5
8	7.9	0.6	0.06	1.6	0.069	15.0	59.0	0.9964	3.3	0.46	9.4	5

(this is data on wine quality, there are 1600 entries) from
Prediction-of-Wine-Quality/winequality-red.csv at master · amberkakkar01/Prediction-of-Wine-Quality (github.com)

Or like this...

JESSICA LI · UPDATED 3 YEARS AGO

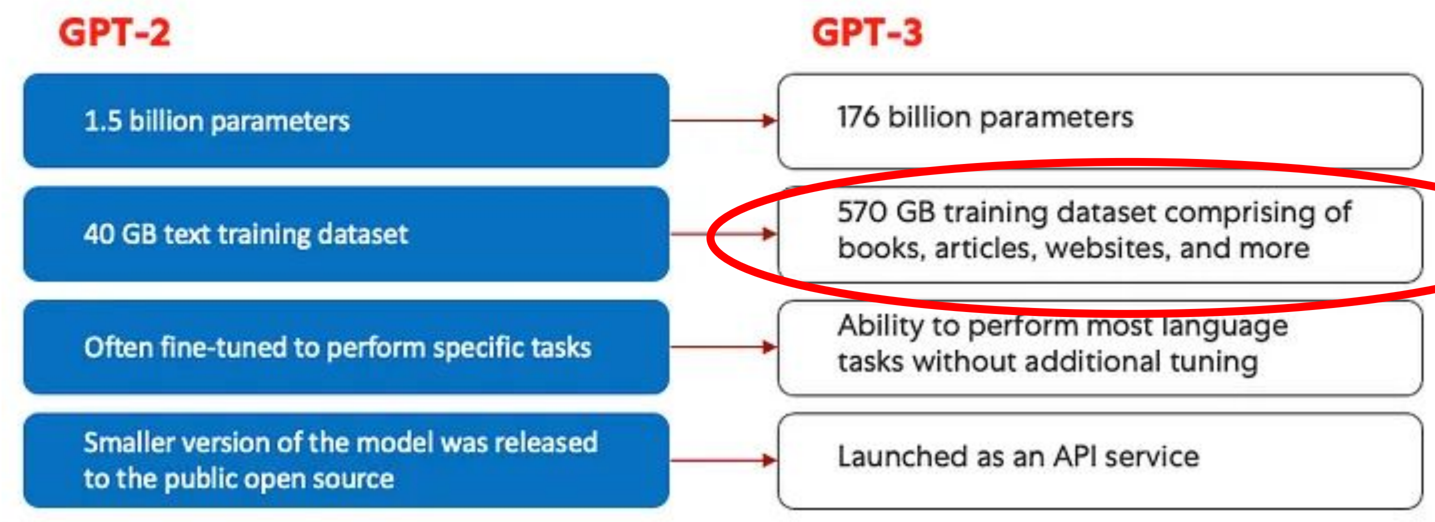
901 | New Notebook | Download (787 MB)

Stanford Dogs Dataset

Over 20,000 images of 120 dog breeds

Data Card | Code (215) | Discussion (5)

Or like this...


GPT-2
GPT-3
1.5 billion parameters
176 billion parameters
40 GB text training dataset
570 GB training dataset comprising of books, articles, websites, and more
Often fine-tuned to perform specific tasks
Ability to perform most language tasks without additional tuning
Smaller version of the model was released to the public open source
Launched as an API service

Machine Learning Targets

Think of the **target** as your **dependent variable** or **y variable.**

Furthermore, we can break things down into two different problems:

Prediction: predicting a future value, targets are **values**

Classification: Classifying objects, targets are **categories**

What animal is this?
Tiger? Dog? Lion? Cat?

Is this a dog?
TRUE or FALSE

Has this dog been a good boy?
On a scale of 1 – 10

What will be the price of this stock tomorrow?

What will be the concentration of citric acid in an orange given these parameters?

Dimensional Data

1	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
2	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
3	7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968	3.2	0.68	9.8	5
4	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8	5
5	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998	3.16	0.58	9.8	6
6	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
7	7.4	0.66	0.0	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5
8	7.9	0.6	0.06	1.6	0.069	15.0	59.0	0.9964	3.3	0.46	9.4	5

We are familiar with having **one independent** variable, however, we can make models that take in **several or many independent variables.**

A dataset has **multiple dimensions** if there are **multiple independent variables** or **features**
A dataset with a lot of features is known as **high dimensional data.**

Let's start by focusing on this.

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968	3.2	0.68	9.8	5
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8	5
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998	3.16	0.58	9.8	6
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
7.4	0.66	0.0	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5
7.9	0.6	0.06	1.6	0.069	15.0	59.0	0.9964	3.3	0.46	9.4	5

Learning is Supervised: we have a clear target: **quality (dependent variable)** and a bunch of **features (independent variable)** (fixed acidity, citric acid, density, pH etc…) with clear values.

Images do not have a clear featurization making them complicated, but information on there target is known.

The model ChatGPT runs on and how it is trained is **unsupervised**. Furthermore, there is no clear featurization of the text data.

Despite image recognition and ChatGPT (NLP) having no clear features, some models can still make accurate predictions/classifications!

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968	3.2	0.68	9.8
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998	3.16	0.58	9.8
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
7.4	0.66	0.0	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4
7.9	0.6	0.06	1.6	0.069	15.0	59.0	0.9964	3.3	0.46	9.4

Independent variables/features/X-values

quality
5
5
5
6
5
5
5

Dependent variable/
Target/y-value

Linear Regression

Linear Regression: Take data in and produce a model in the form of

$$Y = f(X_1, X_2, ..., X_p) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j$$

X_j can be:

- Quantitative inputs
- Transformations of quantitative inputs, e.g., log, exp, powers, etc. Basis expansions, e.g., $X_2 = X_1^2$, $X_3 = X_1^3$
- Interactions between variables, e.g., $X_1 X_2$
- Encoding of levels of inputs

Ridge and Lasso Regression

- Math gets more complicated, but the general idea is that ridge and lasso regressions **punish features that correlate with each other.**
- For a good linear regression, it assumes all features are **independent** from each other, but with your data, that may not be true. So these algorithms are here to help with that.
- Introduces **hyperparameter alpha**

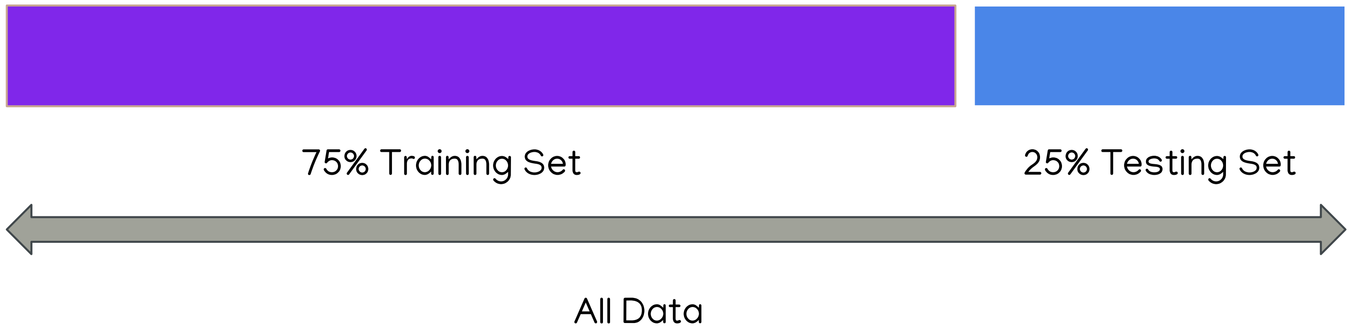
$$RSS_{ridge}(w,b) = \underbrace{\sum_{i=1}^{n}(y_i - (w_i x_i + b))^2}_{\text{Fit training data well}} + \alpha \overbrace{\underbrace{\sum_{j=1}^{p} w_j^2}_{\text{Keep parameters small}}}^{\text{L2 penalty / Penalty Term / Regularisation Term}}$$

A trade-off between fitting the training data well and keeping parameters small

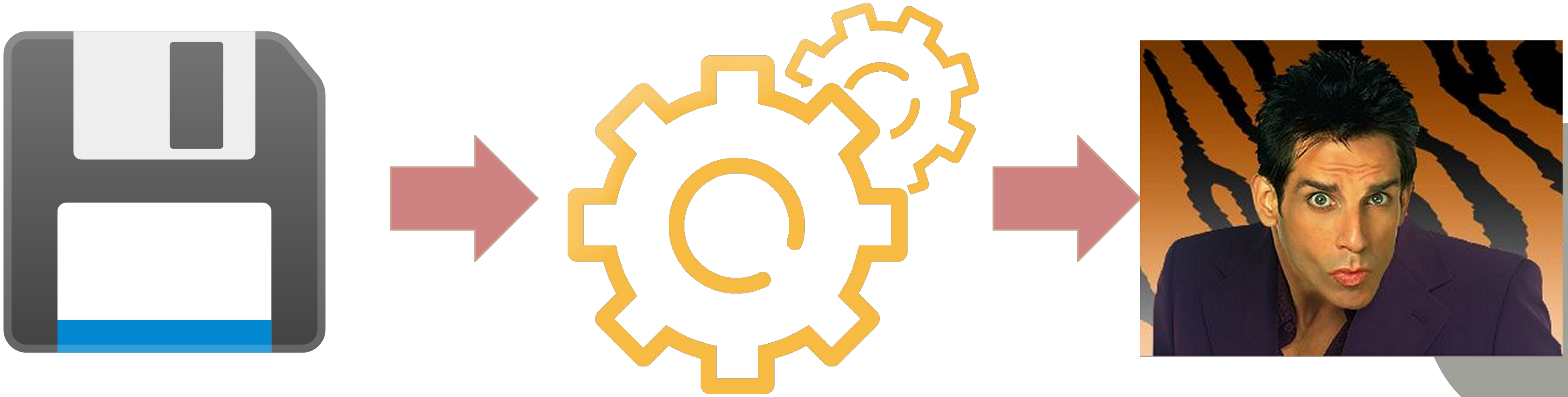
Cross validation

Now that you have a model, we need some measurement on how well it performs.

Let's split up our data into some arbitrary parts. We make a **training set** and a **test set**.



Cross validation



75% Training Set

Some algorithm

model

Cross validation

25% Test Set

Trained Model

Some prediction/ classification

Compare: Some prediction/ classification vs Actual Results → Metric of performance

Cross Validation

The model does <u>not</u> see the test set when training. After training it extrapolates the dependent variable and that dependent variable is tested against the actual values.

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	
7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968	3.2	0.68	9.8	
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8	
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998	3.16	0.58	9.8	?
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	
7.4	0.66	0.0	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	
7.9	0.6	0.06	1.6	0.069	15.0	59.0	0.9964	3.3	0.46	9.4	

Cross validation

Test set

pH	sulphates	alcohol	quality
3.51	0.56	9.4	
3.2	0.68	9.8	
3.26	0.65	9.8	
3.16	0.58	9.8	?
3.51	0.56	9.4	
3.51	0.56	9.4	
3.3	0.46	9.4	

Model after Training

extrapolation

quality
Maybe 5?
Maybe 6?
Maybe 10?
Maybe 9?
Maybe 6?
Maybe 7?

compare

Actual result

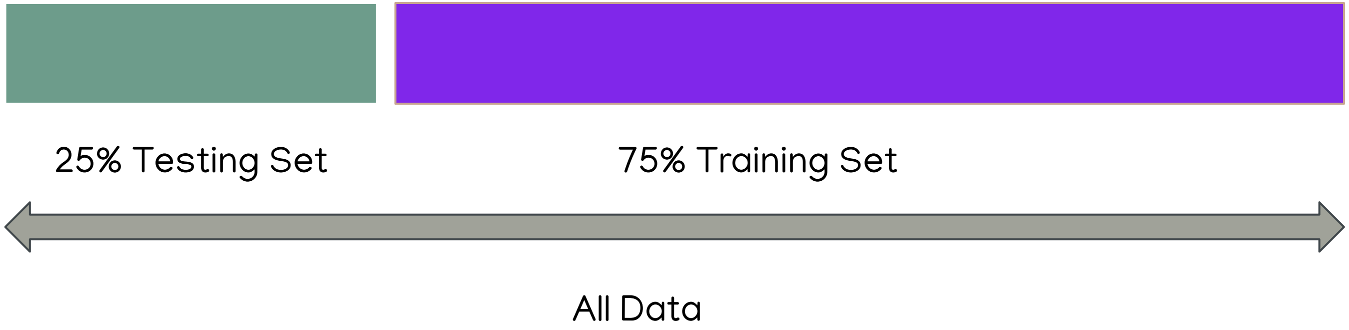
quality
5
5
5
6
5
5
5

Model is right 1 time out of 6 of the time.

Cross Validation

Repeat several times with a different splits of data!!!

Then take the mean validation score over several trials. Very sciency.



Cross validation metrics

For prediction problems: Mean absolute error, Mean square error, R^2 score

For classification problems: Accuracy, Precision, Recall, F1

Hyperparameter Tuning

Change input parameter until you yield a good validation score

For our Lasso and Ridge model it means changing alpha until our model achieves a good cv score.

Basically, make a bunch of models varying parameters (alpha) and graph it.

Find the minimum value. That alpha is the best score.

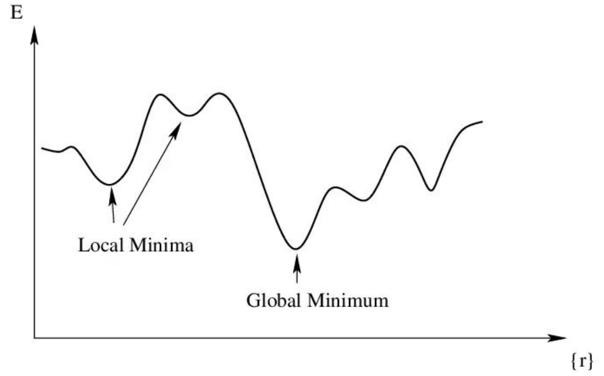
General Process For Developing Models

1. Get Data.
2. Clean up data. Removing duplicate entries, removing incomplete entries, removing outliers, scaling or transforming the data, etc.
3. Plug in data into an algorithm
4. See how that model performed.
5. Modify hyperparameters, optimizers, and or learning rate until you can find the best validation score
6. Export model to be used in applications.

Somethings to know

A big idea in machine learning or computations in general is finding the fastest way to the global minimum and getting out of a local minimum.

To achieve this, we often write code to go through possible hyperparameters, optimizers, learning rate, find best features and drop unnecessary features, etc.
This is to cut loss with the most efficient time.



Let's make a Neural Network.