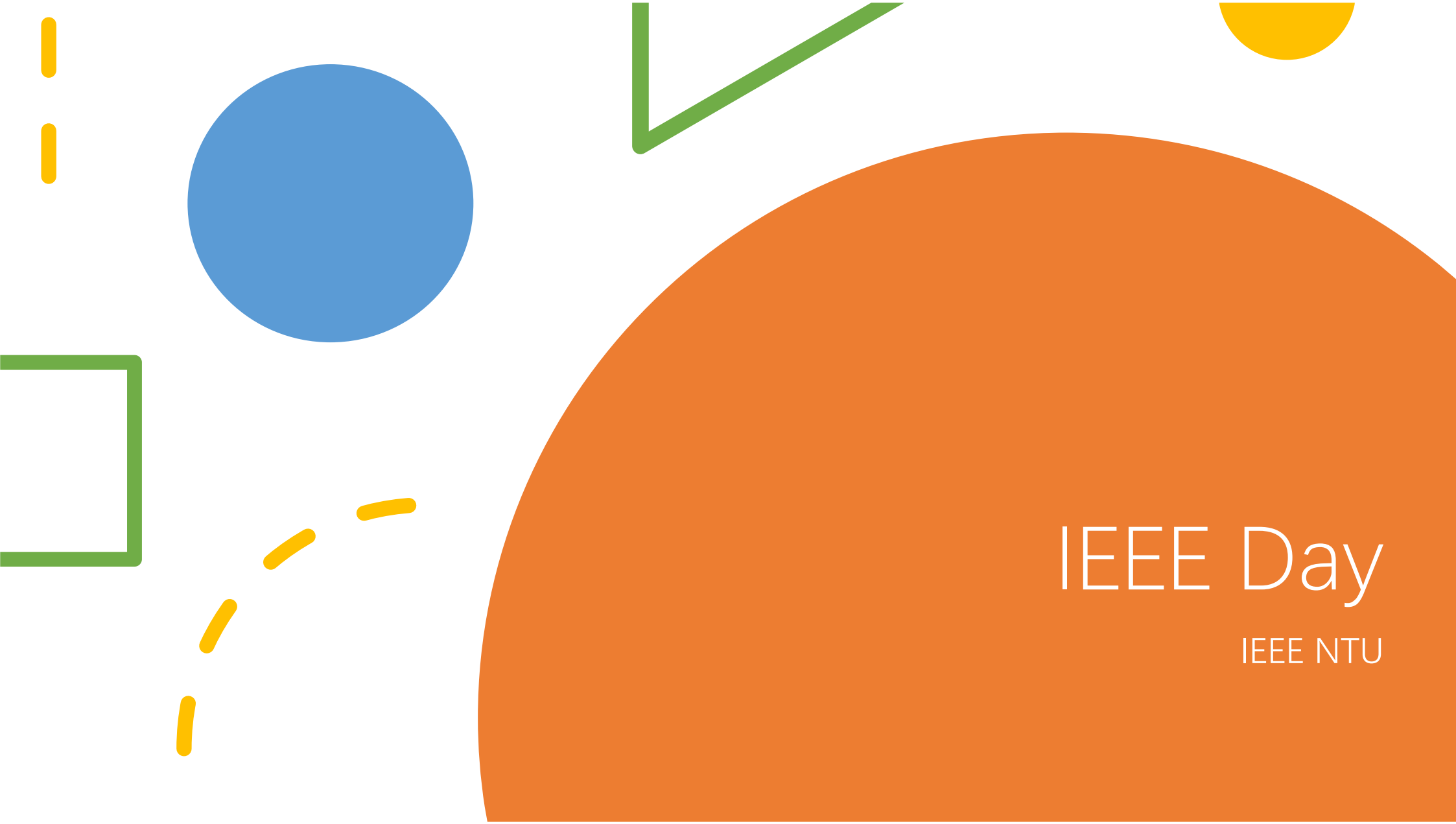

IEEE Day
IEEE NTU

Install
Anaconda

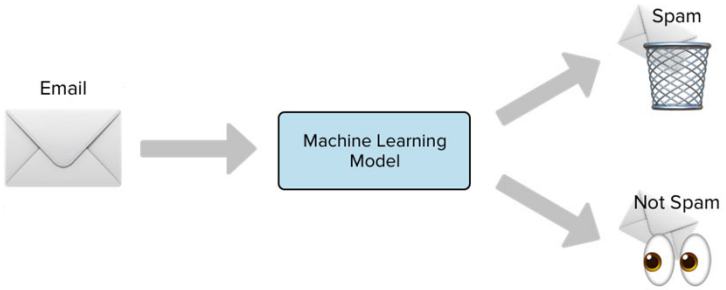
Index

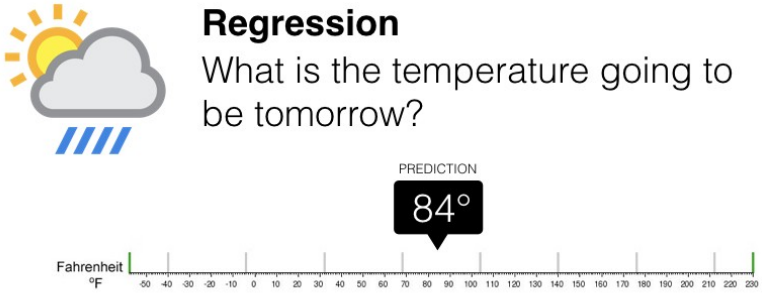
- What are common Machine Learning Tasks
- What are some classification Tasks
- Classification Toolsets: Random Forest, Logistic Regression,
- Why starting with these models?
- Math principle behind regression
- Dataset Management Tool: Pandas, DataFrame
- Hands on project

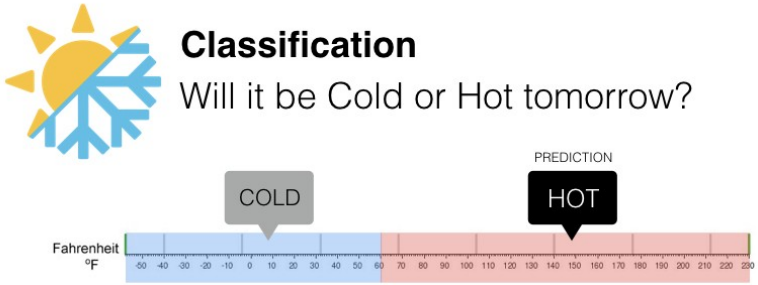
What are common ML Tasks

- Classification / Regression
- Object detection / Tracking
- NLP Tasks
- Reinforcement Learning Tasks
- Generative Adversarial Networks (GAN)
- New Frontiers

Common ML Tasks: Classification/Regression







Example

- Classification can be used to analyze whether an email is a spam or not spam.
  - The algorithm checks the keywords in an email and the sender's address is to find out the probability of the email being spam.
- Regression model can be used to predict temperature for the next day, we can use a classification algorithm to determine whether it will be cold or hot according to the given temperature values.

Understanding the difference between Regression and Classification

Both are supervised learning algorithms

Regression -> Output is a numerical value

Classification -> Output of the algorithm falls into one of various pre chosen categories

Common ML Tasks: Classification

- Why we need classification
- What are the inputs
- What are the outputs
- What are some common methods of classification

Common ML Tasks: Classification

- **Why** we need **classification**
  - Manufacturing: Detect pattern in manufacturing to respond accordingly
  - Access control: Face recognition
  - Agricultural: Crop classification, identification
  - Social network: Friend recommendation, ads
  - Business: Customer churn prediction based on behavior
  - Security: Malware classification

Common ML Tasks: Classification

- What are some **possible inputs**
  - Integer/ floating point data
  - Images
  - Time series data

Common ML Tasks: Classification

What are some **possible outputs**?

- Binary Output
- Continuous output
- Discrete classes output

Common ML Tasks: Classification / Regression

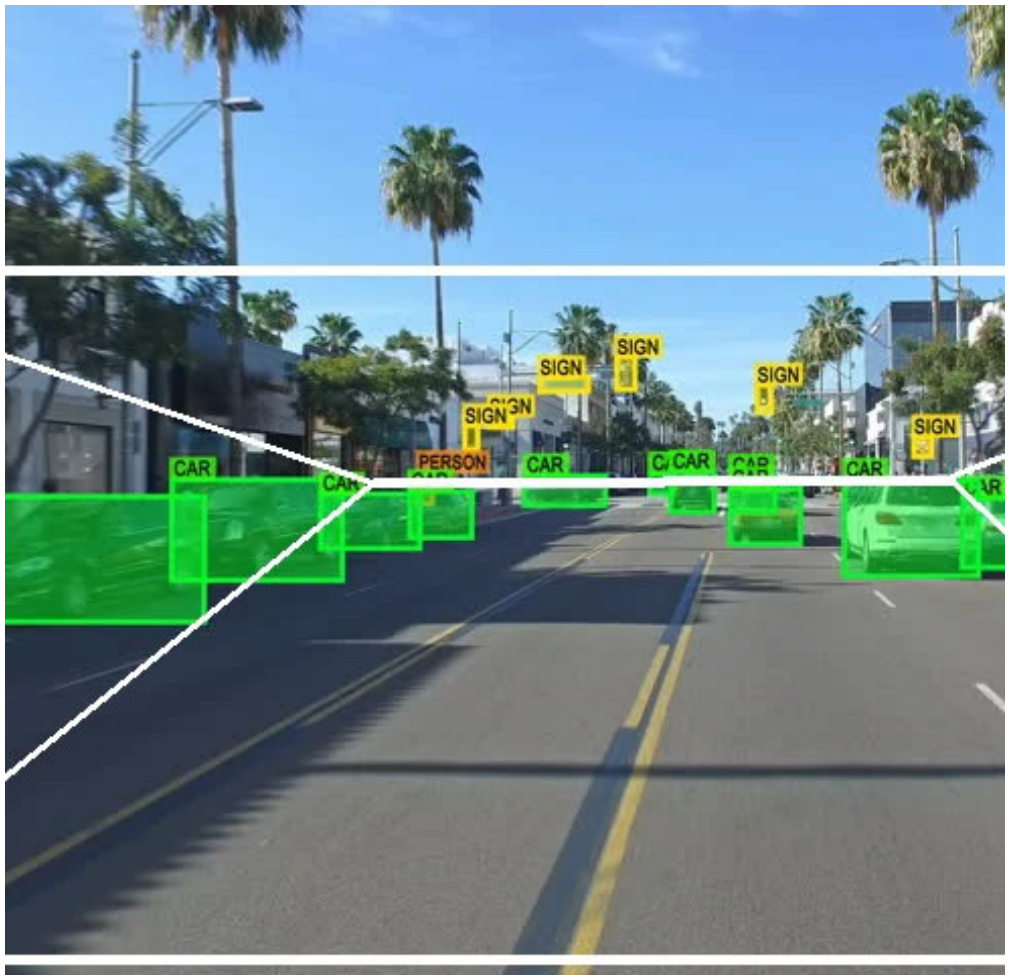
- What are some Common structure for classification
  - Linear classifier
  - MLP/CNN Models
  - Decision tree

Common ML Tasks: Object detection / tracking

- Why we need Object Detection
- What are the inputs (image)
- What are the outputs
- Common NN structure for object detection

Common ML Tasks: Object detection / tracking

- Why we need Object Detection
  - Access control, surveillance system: human detection, bounding box for face rec
  - Surveillance camera
  - Manufactory: detection and classification for objects
  - Self-driving car: traffic, traffic light, pedestrian detection



Common ML Tasks: Object detection / tracking

- What are the inputs (image)
  - Image from camera modules
  - Image are usually too large for NN, so usually compress it first

Common ML Tasks:
Object detection / tracking

- What are the outputs
  - Bounding boxes
  - Bounding boxes with class label

```
results.xyxy[0]  # img1 predictions (tensor)
results.pandas().xyxy[0]  # img1 predictions (pandas)
#      xmin    ymin    xmax   ymax  confidence  class    name
# 0  749.50   43.50  1148.0  704.5    0.874023      0  person
# 1  433.50  433.50   517.5  714.5    0.687988     27     tie
# 2  114.75  195.75  1095.0  708.0    0.624512      0  person
# 3  986.00  304.00  1028.0  420.0    0.286865     27     tie
```

Common ML Tasks: Object detection / tracking

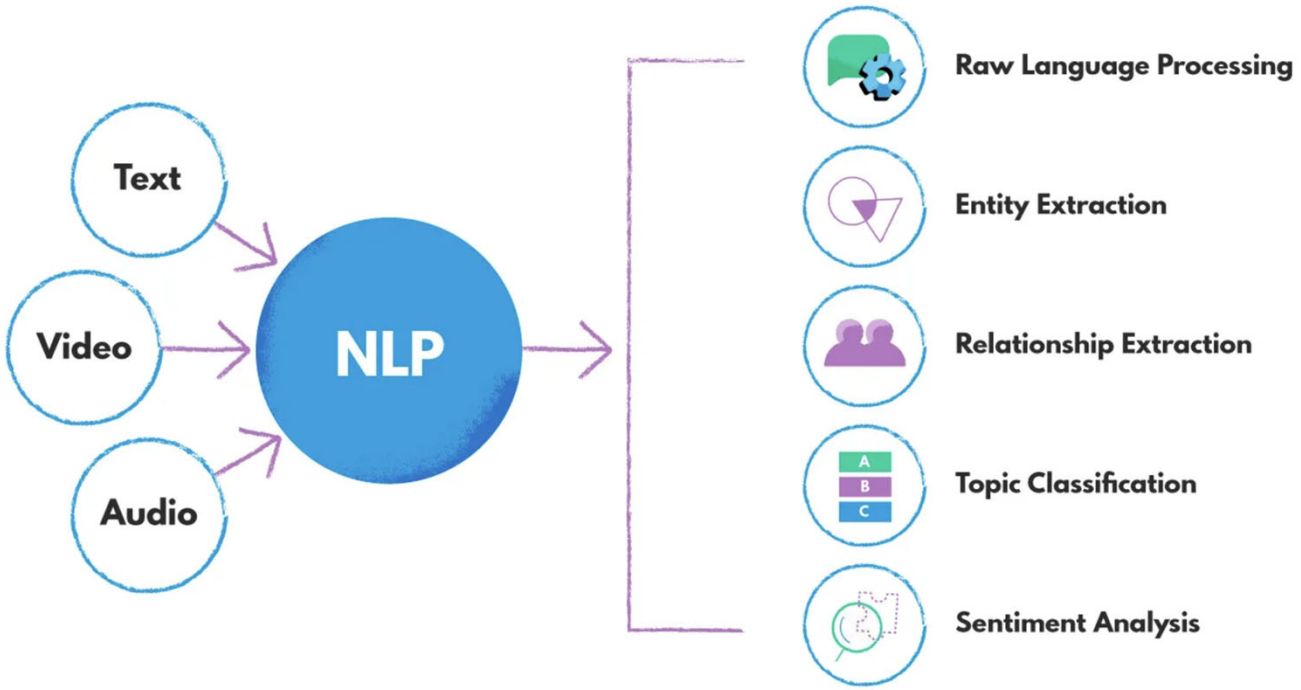
- Common NN structure for object detection
  - Fast R-CNN
  - YOLO
  - ViT (Vision Transformer)

Common ML Tasks: NLP

- What are some applications for NLP
- What are the inputs
- What are the outputs
- Attention is all you need

Common ML Tasks: NLP

- What are some applications for NLP
  - Information retrieve
  - Text classification
  - Translation
  - Question and answering
  - Semantic role labeling (For instance Navigating an agent in a house)

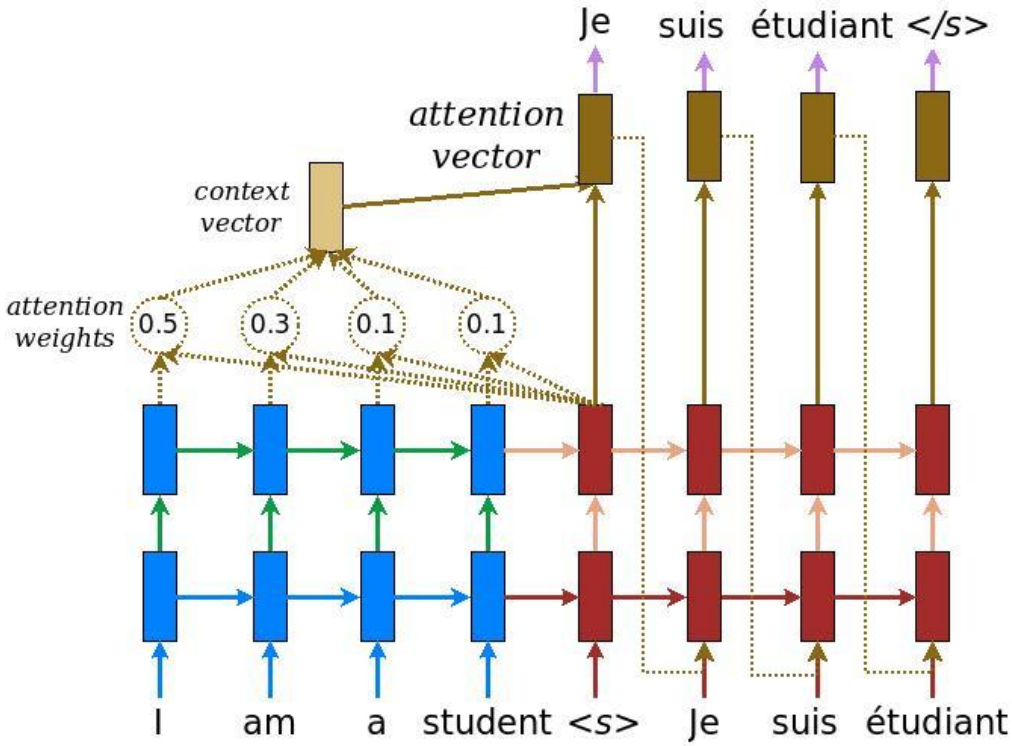


Common ML Tasks: NLP

- What are the inputs
  - Human sentence
  - “Go through the living room, and clean the floor of the kitchen”
  - Tokenization of the words

Common ML Tasks: NLP

- What are the outputs
  - In the case of text retrieve, Tokens
  - In the case of translation, token to token
  - In the case of navigating, photos from the agent as the input and navigating command as the output



Common ML Tasks: NLP

- Attention is all you need (Self-attention/ Cross-attention)

$$\begin{bmatrix}0.321\\0.577\\\vdots\end{bmatrix} \quad \begin{bmatrix}0.901\\\cdot\\\vdots\end{bmatrix} \quad \begin{bmatrix}1.144\\\cdot\\\vdots\end{bmatrix} \quad \cdots \quad \begin{bmatrix}0.314\\\cdot\\\vdots\end{bmatrix} \Big\} D^{(v)}$$

$$\mathbf{v}_1 \quad \mathbf{v}_2 \quad \mathbf{v}_3 \quad \mathbf{v}_N$$

$$\begin{bmatrix}\\\end{bmatrix} \quad \begin{bmatrix}\\\end{bmatrix} \quad \begin{bmatrix}\\\end{bmatrix} \quad \cdots \quad \begin{bmatrix}\\\end{bmatrix} \Big\} D^{(k)}$$

$$\mathbf{k}_1 \quad \mathbf{k}_2 \quad \mathbf{k}_3 \quad \mathbf{k}_N$$

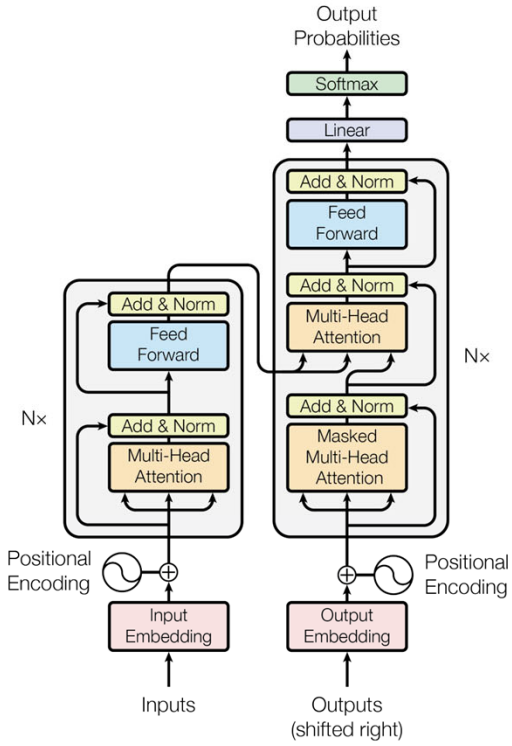
$$\mathbf{q} = \begin{bmatrix}\\\end{bmatrix}$$

$$\alpha_i = f(\mathbf{q}, \mathbf{k}_i)$$

$$\{\alpha_i\}_{i=1}^{N} = \text{softmax}\left(\{\tanh(\mathbf{q}^T\mathbf{k}_i)\}_{i=1}^{N}\right)$$

$$\mathbf{y} = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i$$

$$\sum_{i=1}^{N} \alpha_i = 1$$

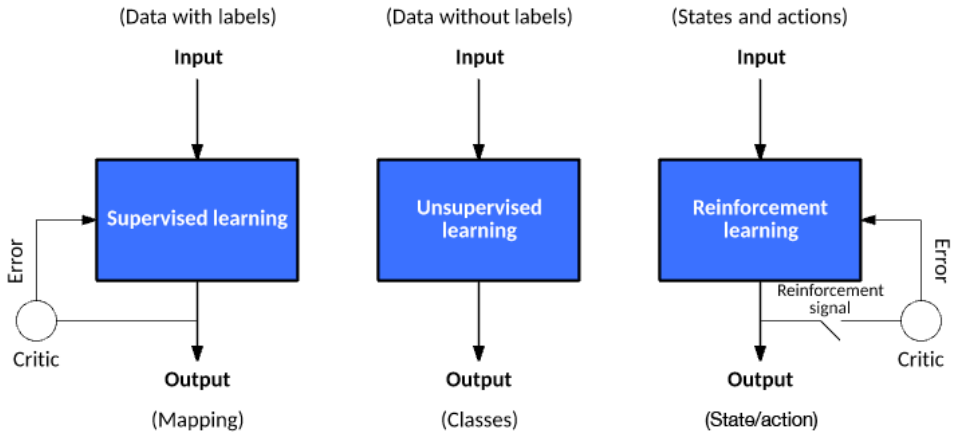


What are common ML Tasks: RL Learn Tasks

There are some cases where its very hard to define

What are common ML Tasks: RL Learn Tasks

- **Input:** The environment the agent is in, the possible next movements(action space).
- **Output:** The optimal next step
- **Training:** Based on input. Model will return a state and user will decide to reward or penalize model based on output. The model then continues to learn. Best solution decided based on maximum reward.



What are common ML Tasks: RL Learn Tasks

- Reinforcement learning allows us to take sutiable action to maximise reward in a particular situation (finding best possible path/behaviour).
- Differs from supervised learning as in RL instead of an answer key, the reinforcement agent decides what to do with task.
- Thus, in absence of training dataset, it is bound to learn from experience. Ie. It learns from its mistakes.

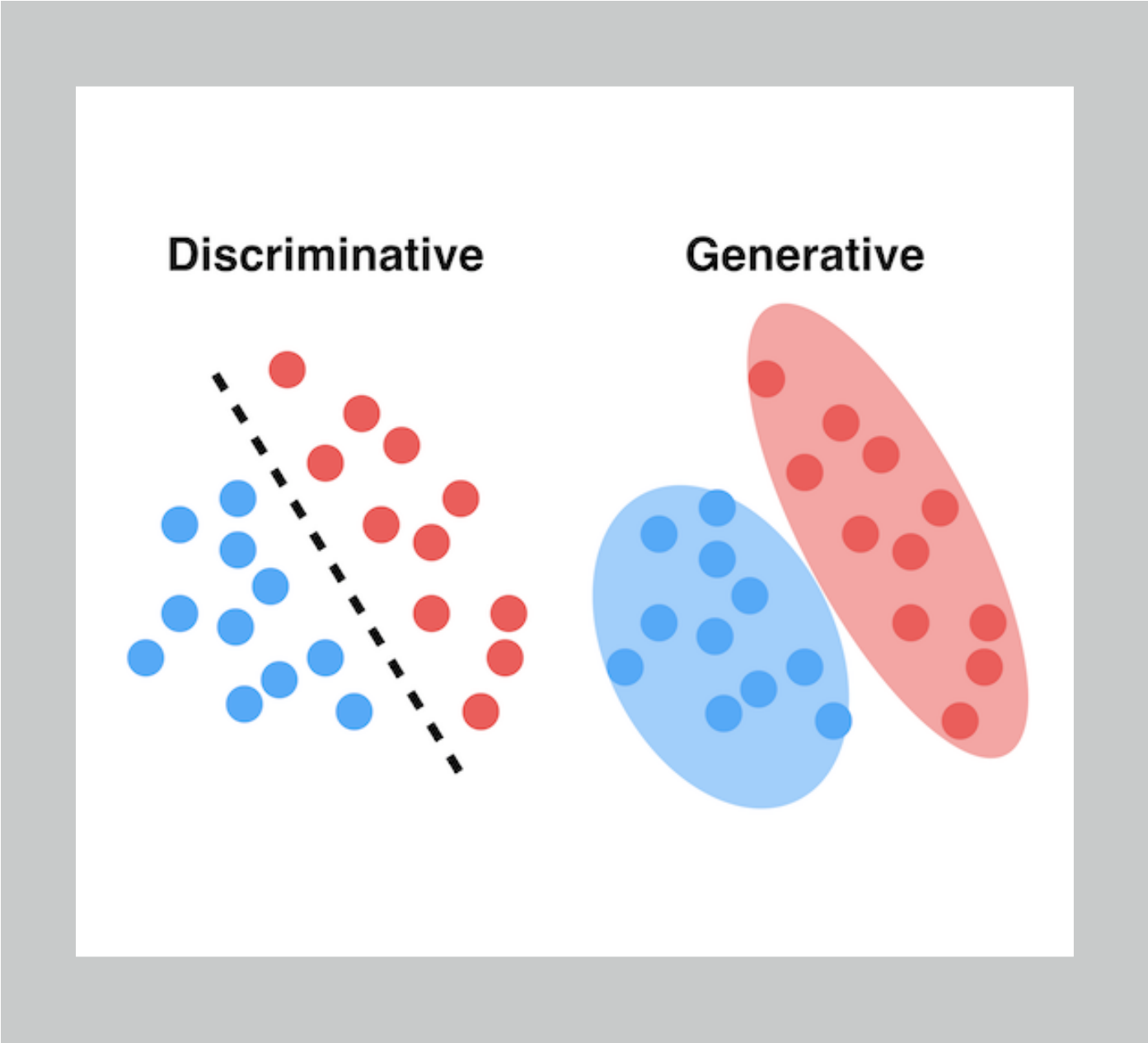
What are common ML Tasks: Generative adversarial networks

- Generative models that create new data instances that resemble your training data.
- It achieves this by pairing with -
    1. Generator: learns to produce the target output (becomes negative training examples for the discriminator)
    2. Discriminator: learns to distinguish true data from output of generator (penalizes generator for producing implausible results).

What are common ML Tasks: GAN

For a set of data instances, X and a set of labels, Y.

- **Generative** models:
  - Create new data instances.
  - Capture the joint probability p(X,Y).
- **Discriminative** models:
  - Discriminate between different kinds of data instances.
  - Capture the conditional probability p(Y | X).



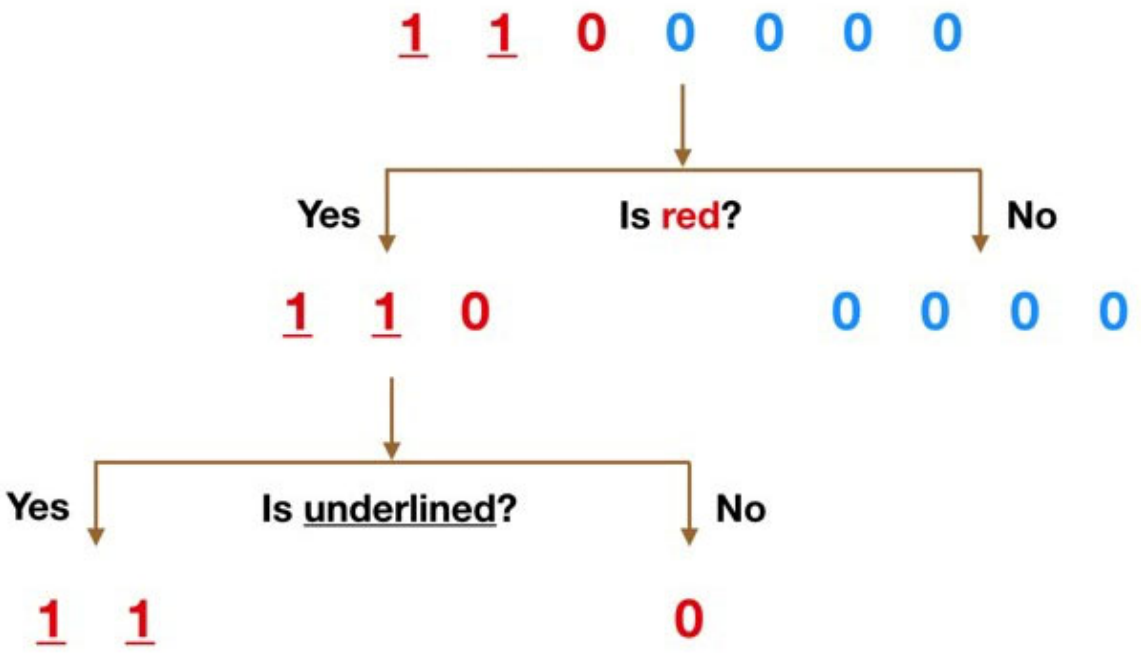
Classification Toolsets

- What are some Common models for classification
  - Linear classifier
  - MLP/CNN Models
  - **Decision tree**
  - **Logistic Regression**

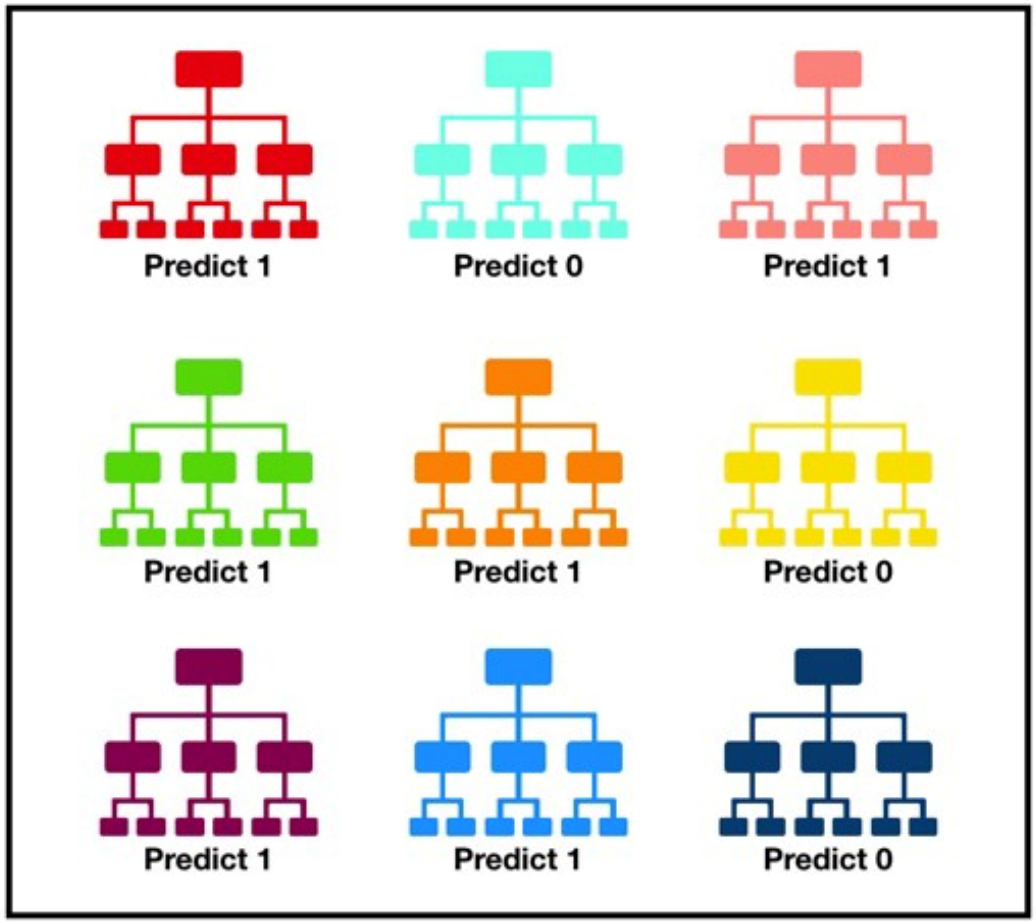
Random Forest

- The ability to precisely classify observations is extremely valuable for various business applications like predicting whether a particular user will buy a product or forecasting whether a given loan will default or not.
- Random forest Classifier is one such way to implement classification
- Comprises of multiple decision trees

What feature will allow me to split the observations at hand in a way that the resulting groups are as different from each other as possible (and the members of each resulting subgroup are as similar to each other as possible)?



Implementation of a random forest



Tally: Six 1s and Three 0s

Prediction: 1

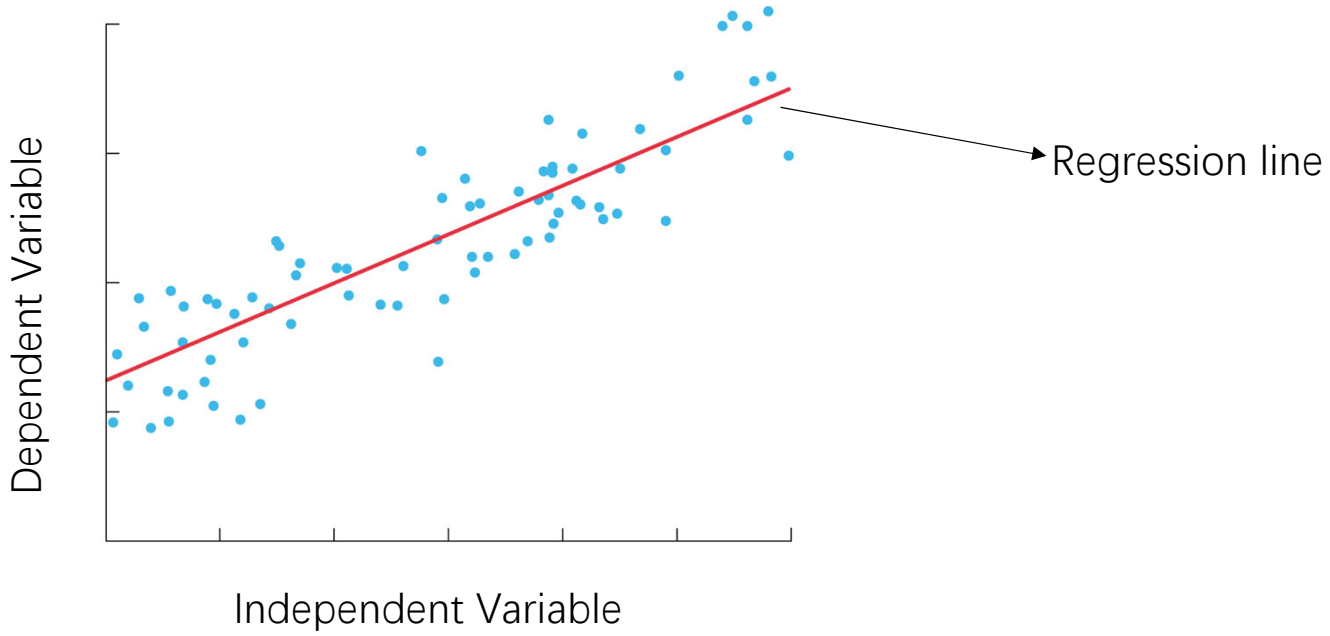
How does Regression work?

- Dependent vs. Independent Variables
- Understanding the line of best fit
  - R-Squared value
- Investigating the relationship between variables
  - Which variables matter most?
  - Which variables can we ignore?
  - How do those variables interact with each other?

Dependent vs. Independent Variables

- **Dependent** - the main factor that you're trying to understand or predict.
- Also known as the **response variable, left hand variable,** and placed on the **Y axis.**
- Dependent variables change with changes in some of the other variables in the model.
- **Independent** - the factors you suspect have an impact on your dependent variable.
- Also known as a **predictor, right hand variable,** and placed on the **X axis**.
- Independent variables aren't influenced by other variables in the model, rather they influence other variables.
- For example, we want to determine if COVID-19 infection rates are affected by vaccination status, the COVID-19 rates are the **DV,** and the vaccination status (categorical: vaccinated/un-vaccinated) is the **IV.**

Understanding the line of best fit



Regression line - best explanation of the relationship between the independent variable and dependent variable.
The line of best fit (red line) is the best estimate of a straight line that minimizes the distance between itself and the data points.

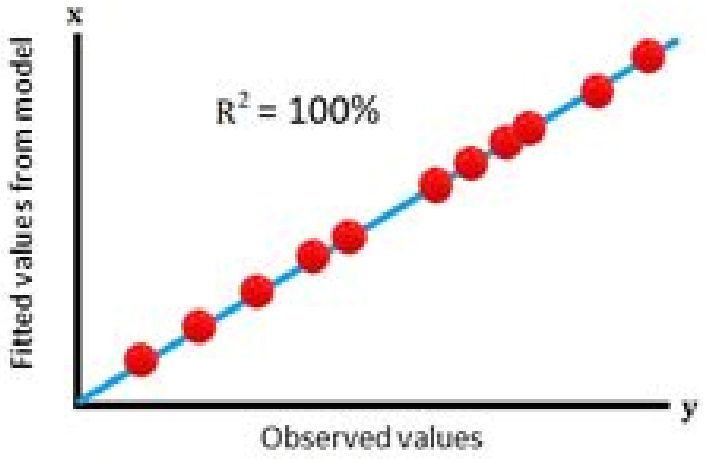
Investigating the relationship between variables

- After fitting a linear regression model, you need to determine how well the model fits the data. Does it do a good job of explaining changes in the dependent variable?
- R-squared is a goodness-of-fit measure for linear regression models. It indicates the percentage (1-100%) of the variance in the response/dependent variable that is explained(or caused) by the predictor/independent variable.
- Higher R-squared values represent smaller differences between the observed data and the fitted values. So, higher R-squared, stronger the relationship between the DV and the IV.

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$$

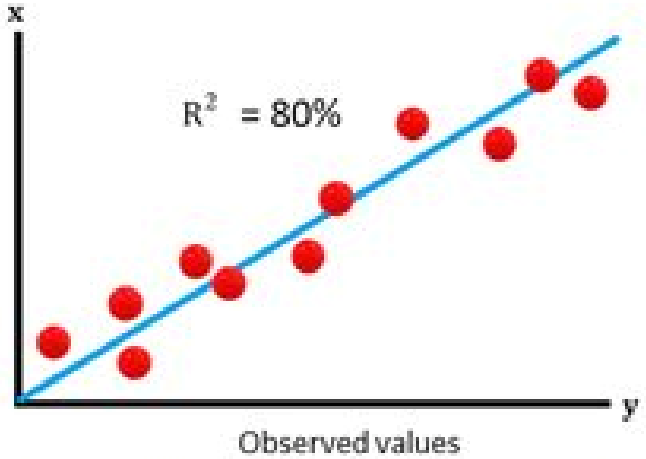
Comparison of R-Squared for Different Linear Models (Same Data Set)

Great



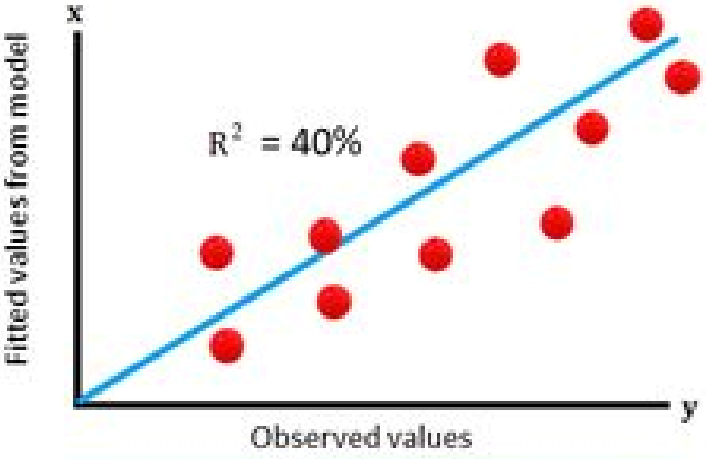
Fitted = observed: Model explains all variance

Good



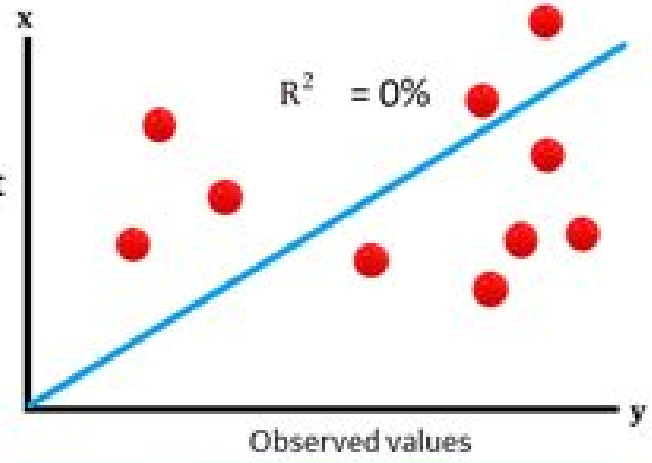
Model explains bulk of variance

OK



Model explains 40% of variance, so is reasonable.

Inconsistent



Model fails to explain any variance

Logistic Regression

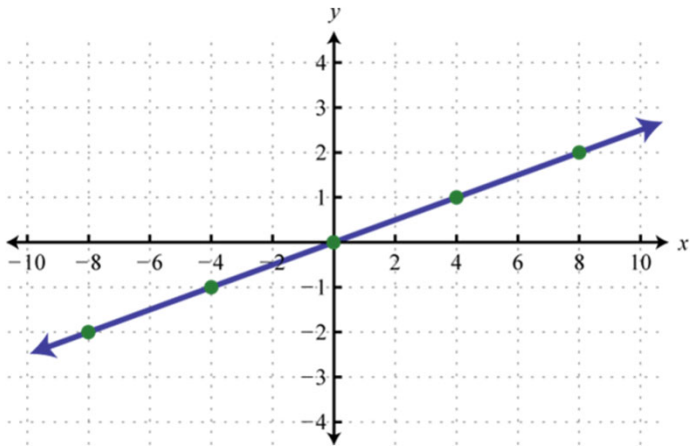
- Used in Binary classification problems

No. Of hours	Pass/Fail
0	0
3	0
3.5	1
2	0
6	1
1	1
4	1
2	0
0.5	0
1.5	0
7.5	1
3	0
4	1
2	0
6.5	1

- 0 = Fail
- 1 = Pass

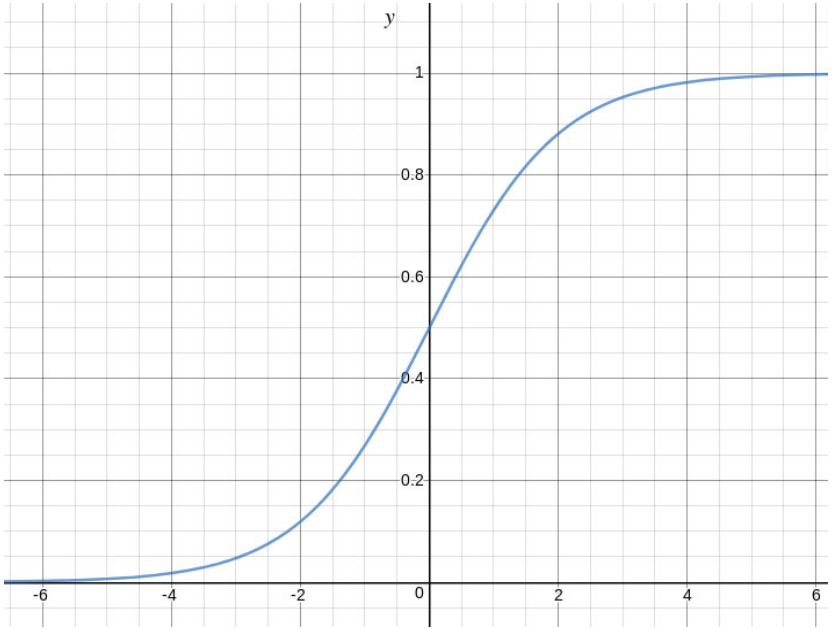
$$h(x_i) = \beta^T x_i$$

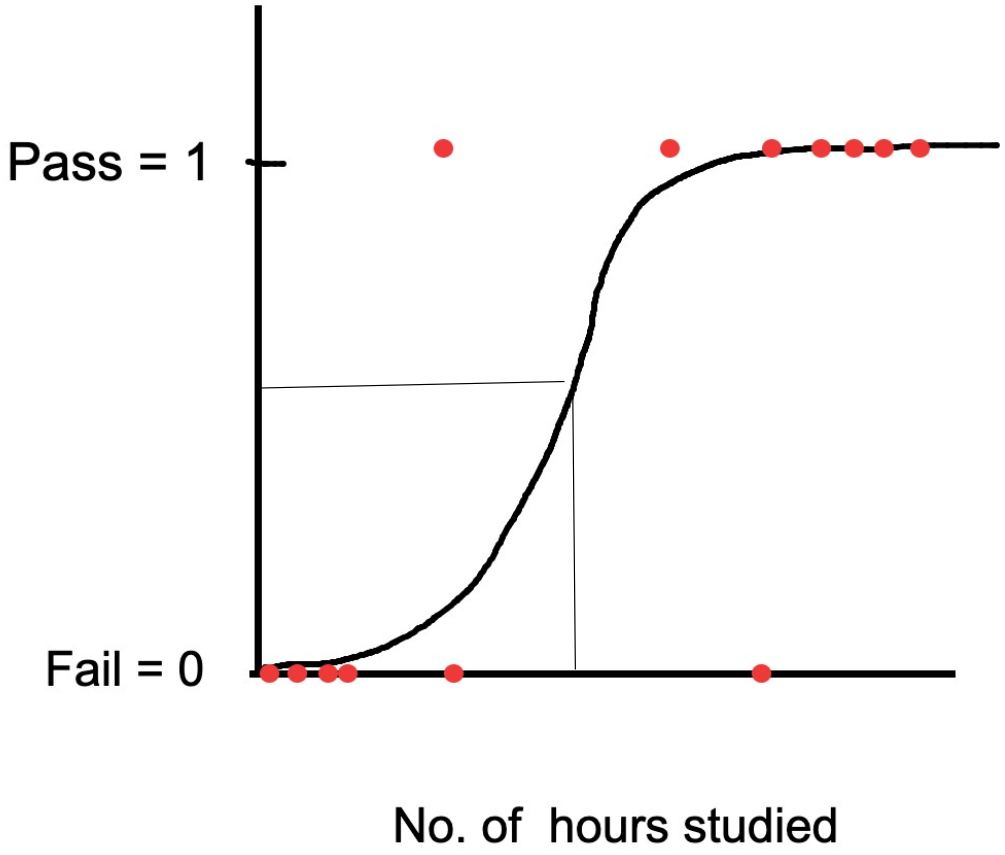
X i --> Independent variable
h(X I) --> Dependent variable
β --> Linear coefficient / Slope of the graph



$$h(x_i) = g(\beta^T x_i) = \frac{1}{1 + e^{-\beta^T x_i}}$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

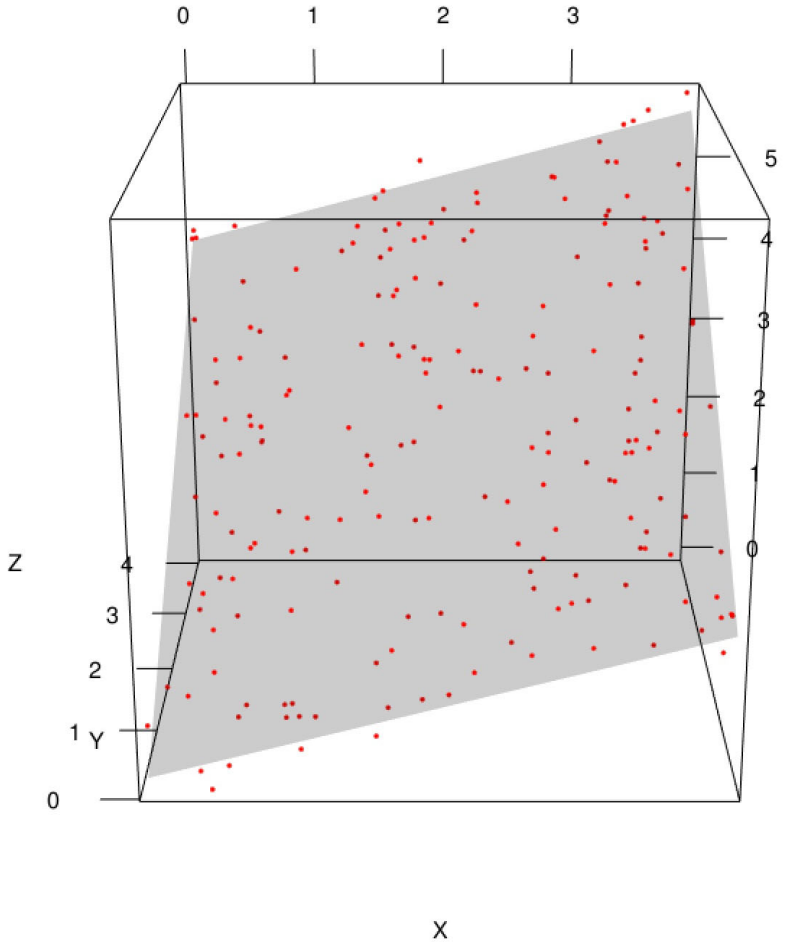




> 50% -- Classify as pass
<= 50% -- Classify as fail

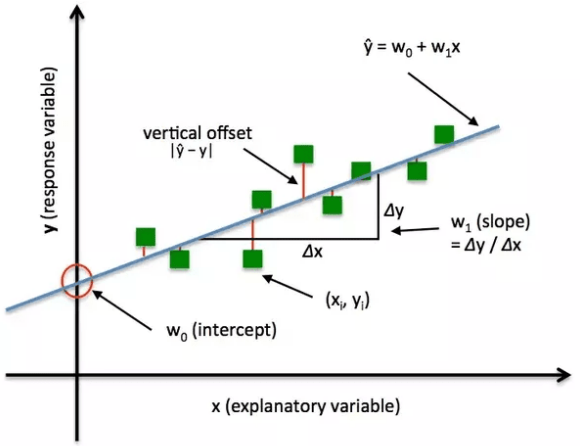
How to obtain parameters? Gradient Descent

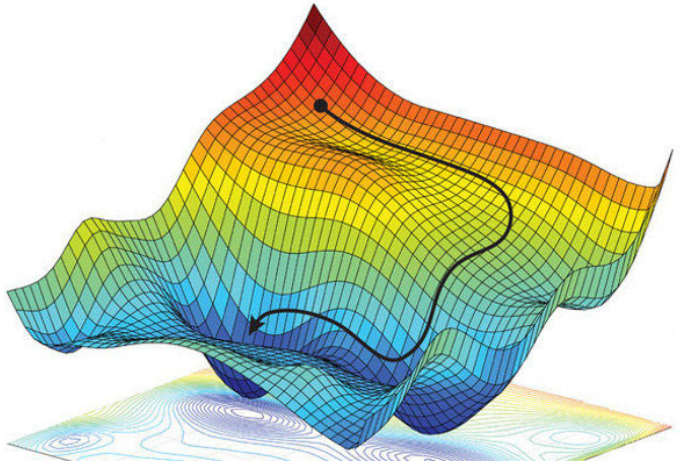
- For instance, given: $z = a + bx + cy$, how to calculate a, b, c?



How do we know if a set of parameter is good or not: Loss function

- Loss function is defined by how well the model is fitting the actual data
- Loss function is the rubric of how well the parameters are
- By finding the a, b, c that have the minimal loss function value, we will obtain the optimal a, b, c.





How to obtain parameters? Gradient Descent

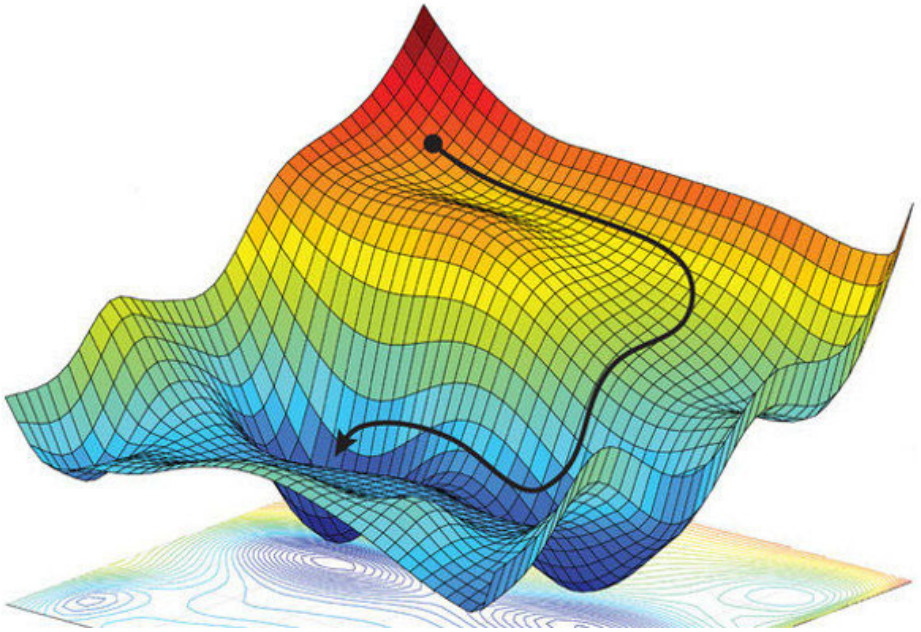
- iterative first-order optimization algorithm used to find a local minimum/maximum of a given function
- Used in ML and DL to minimize cost/loss function
- Function requirements: Differentiable and convex

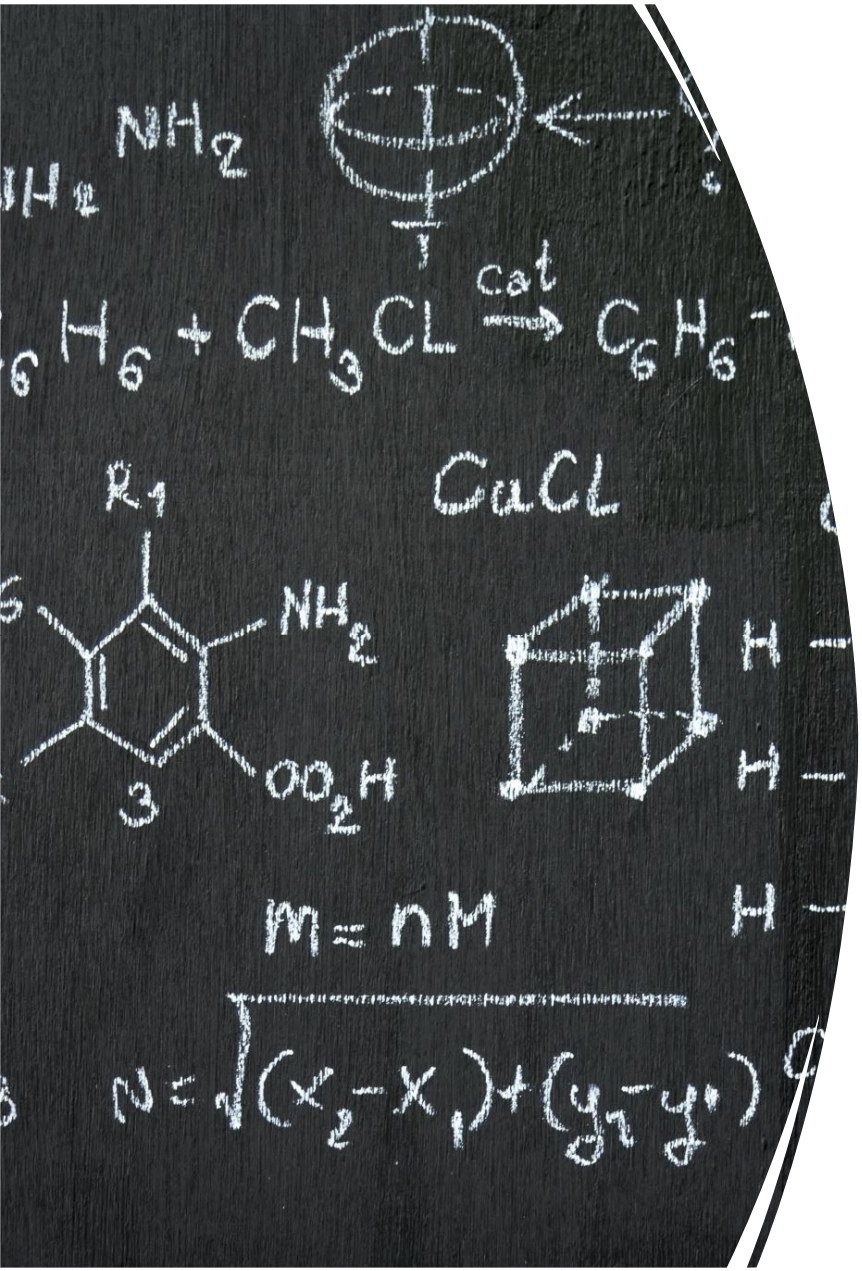
- **The Algorithm:**

- The gradient descent algorithm iteratively calculates the next point using gradient at the current position, scales it (by a learning rate) and subtracts obtained value from the current position (makes a step) to minimize the function.

$$p_{n+1} = p_n - \eta \nabla f(p_n)$$

- Eta = learning rate





Gradient Descent

Example:

f(x) = x^2 – 4x + 1

df(x)/dx = 2x – 4

Learning rate = 0.1

Starting point: x = 9

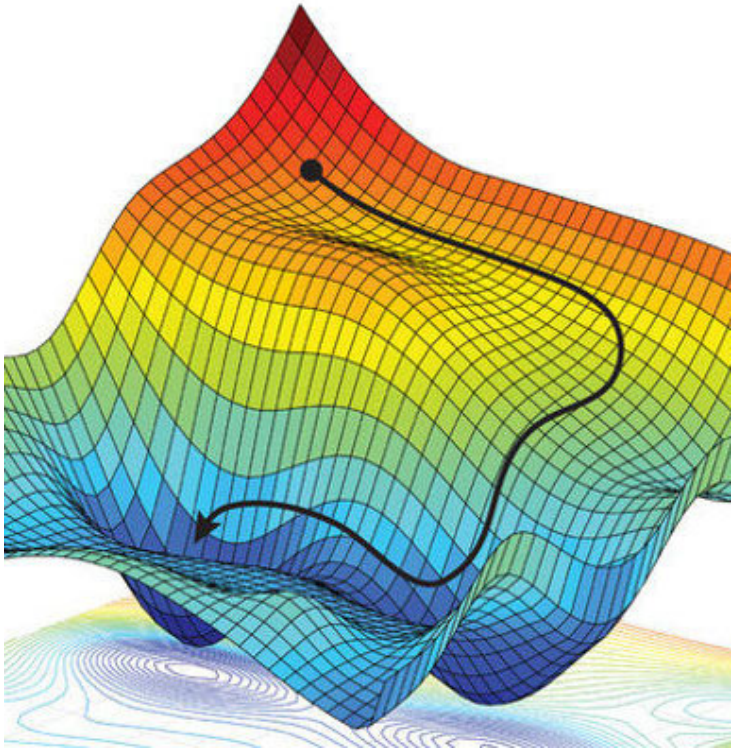
x1 = 9 – 0.1 (2 * 9 – 4) = 7.6

x2 = 7.6 – 0.1 (2 * 7.6 – 4) = 6.48 and so on

Gradient Descent

Challenges in selecting learning rate:

- The smaller the learning rate, the longer the time taken for gradient descent to converge. If the learning rate is small, we might reach the maximum number of iterations allowed before reaching optimum point.
- If the learning rate is too large, the algorithm may not converge to the optimal point (jump around) or even diverge completely.



Dataset Management Tools: Pandas

- What is Pandas
- How to setup it up
- How to import data from various sources
- How to do basic analysis with Pandas
- Basic data cleaning with pandas
- Plot Pandas data

Dataset Management Tools: Pandas

- High-level data manipulation tool
- It is built on the Numpy package and its key data structure is called the DataFrame.
- Used for the following tasks
    - Data cleansing
    - Data fill
    - Data normalization
    - Merges and joins
    - Data visualization
    - Statistical analysis
    - Data inspection
    - Loading and saving data
    - ...

Dataset Management Tools: Pandas

- How to set it up
  - Installation: pip install pandas
  - Using it in Python: import pandas as pd
  - Initialize new objects:
    - `df = pd.DataFrame({the python dict obj})`
    - `s = pd.Series(np.random.randn(5), index=["a", "b", "c", "d", "e"])`

Dataset Management Tools: Pandas

- Importing data using Pandas
  - Pandas supports data import from many widely used sources like HTML,CSV,JSON
  - `df = pd.read_csv('data.csv')`
  - `df = pd.read_json('data.json')`
  - `df = pd.read_html('data.html')`

- Finding Data
  - Numeric index based : `df.iloc[row,column]`
  - Find data in range : `df[start:end]`
  - Label based : `df.loc[label]`

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
5	25.29	4.71	Male	No	Sun	Dinner	4
6	8.77	2.00	Male	No	Sun	Dinner	2
7	26.88	3.12	Male	No	Sun	Dinner	4
8	15.04	1.96	Male	No	Sun	Dinner	2
9	14.78	3.23	Male	No	Sun	Dinner	2
10	10.27	1.71	Male	No	Sun	Dinner	2
11	35.26	5.00	Female	No	Sun	Dinner	4

Dataset Management Tools: Pandas

- How to do basic analysis with Pandas
  - After you have imported external data to DataFrame:
  - Get a glimpse of the data:
    - `df.head() – it read the first 5 line of data`
  - Get the datatype of each column
    - `df.info() – it printes: #Column and #Non-Null and datatype of each col`
  - Get min, max, avg, median, stdev basic description of data
    - `df.describe() – it shows the stats of a given dataframe`
  - Get min, max, avg, median, stdev of data
    - `df.mean() / df.max() / df.min() / …`

Dataset Management Tools: Pandas

- Basic data cleaning with pandas
  - Clean empty cells
    - `Drop the rows since we have enough data : df.dropna()`
    - `Replace with default values : df.fillna(130)`
  - Change the value to same format
    - `20200101 and 2020/01/01 are mixed : df['Date'] = pd.to_datetime(df['Date’])`
  - Clean wrong data
    - `Limit the range:`

```
for x in df.index:
  if df.loc[x, "Duration"] > 120:
  df.loc[x, "Duration"] = 120
```

Dataset Management Tools: Pandas (Cont.)

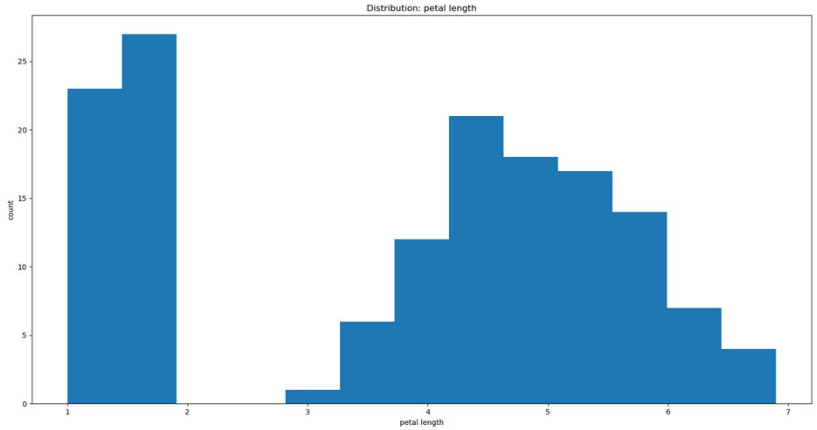
- Basic data cleaning with pandas
  - Clean wrong data
    - Limit the range:

```
for x in df.index:
  if df.loc[x, " SomeIndex "] > 120:
  df.loc[x, " SomeIndex "] = 120
```

    - Clean out of range data

```
for x in df.index:
  if df.loc[x, “SomeIndex"] > 120:
  df.drop(x)
```

  - Remove redundant data
    - `df.drop_duplicates(inplace = True)`



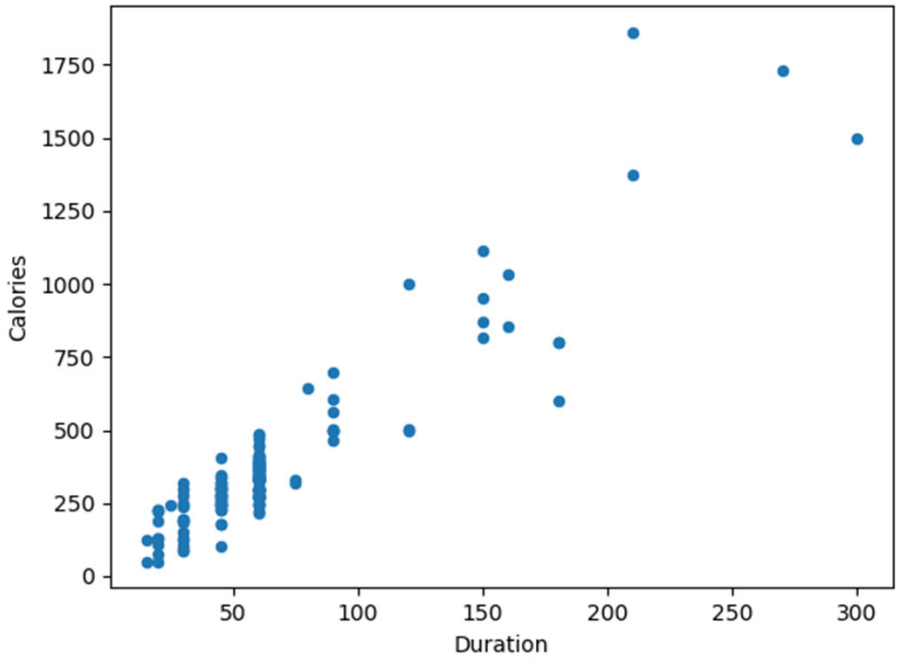
Dataset Management Tools: Pandas

- Visualizing Data
  - plot() method : uses the matplotlib.plt wrapper
  - 'kind' argument specifies type of plot. Eg: bar, hist, scatter.

- Finding Correlation
  - Corr() method
  - 'Method' argument specifies type : 'pearson','spearman','kendall

- Example

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('data.csv')
df.plot(kind ='scatter,x='XColName',y='YColName')
plt.show()
```



Hands-on project

Notebook

Dataset

- Create a Notebook exploring the dataset provided
- Look into the relationships between variables
- Create a ML model to come up with a solution to a problem statement

- Judged based on the accuracy of your results and the quality of your notebook (Data cleaning and exploration) as well as how well you formulate your problem statement and go about solving it

Competition

Competition Dataset:
http://sg.ericfan.win:8080

Check your accuracy by uploading your results from your model on test-set:
http://sg.ericfan.win:8081

Upload your Notebook file:
http://sg.ericfan.win:8082/index.php/s/EqW2wJoeaoqx343

What are common ML Tasks: AlphaFold

- AI Program developed by DeepMind, which predicts 3D models of protein structures
- The protein-folding problem: the challenge to find a method to reliably determine a protein's structure just from its sequence of amino acids
- Algorithm trained on over 170,000 proteins from a public repository of protein structures and sequences
- Uses a form of attention network

What are common ML Tasks: AlphaFold

haFold is a distance map predictor implemented as a very deep residual neural networks with 220 residual blocks processing a repres
ality 64×64×128 – corresponding to input features calculated from two 64 amino acid fragments. Each residual block has three layers i
onvolutional layer – the blocks cycle through dilation of values 1, 2, 4, and 8. In total the model has 21 million parameters. The networ
of 1D and 2D inputs, including evolutionary profiles from different sources and co-evolution features. Alongside a distance map in the f
ained histogram of distances, AlphaFold predicts Φ and Ψ angles for each residue which are used to create the initial predicted 3D str
d authors concluded that the depth of the model, its large crop size, the large training set of roughly 29,000 proteins, modern Deep Le
nd the richness of information from the predicted histogram of distances helped AlphaFold achieve a high contact map prediction prec

SVM

- Used in classification problems
- How does SVM work
  - Create a N-Dim space in the space and put input data into it
  - Draw hyper planes in the space
  - Adjust the parameters in the space to make the margin of hyper plane to the dots of input data as far as possible