Artificial Intelligence(AI) & Machine Learning(ML)

- Artificial Intelligence(AI):

↳ AI is a broad field for the development of intelligent systems capable of performing tasks that typically require human intelligence.

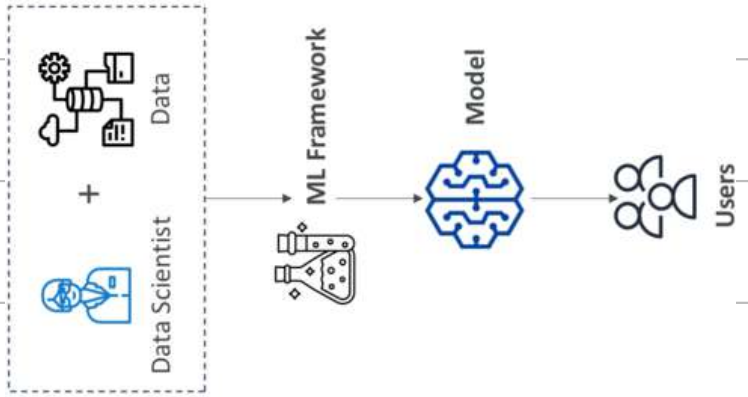
(i) Perception (ii) Reasoning (iii) Learning

(iv) Problem-Solving (v) Decision-maki

↳ Some of the usecases are Computer Vision, Facial Recognition, fraud Detection & IDP

↳ Components of AI include:

1. Data Layer
2. ML framework & Algorithm Layer
3. Model layer



4. Application Layer

- Machine Learning (ML):

↳ It's a type of AI for building methods that allow machines to learn.

↳ Data is leveraged to improve computer performance on a set of task.

↳ Make predictions based on data used to train the model.

- Deep Learning:

↳ Uses neurons and synapses to train our model.

↳ Process more complex patterns in data than ML.

↳ Deep learning because there's more than one layer of learning.

↳ Large amount of input data & requires GPU (Graphical Processing Unit)

Ex: Computer Vision, Natural Language Processing (NLP).

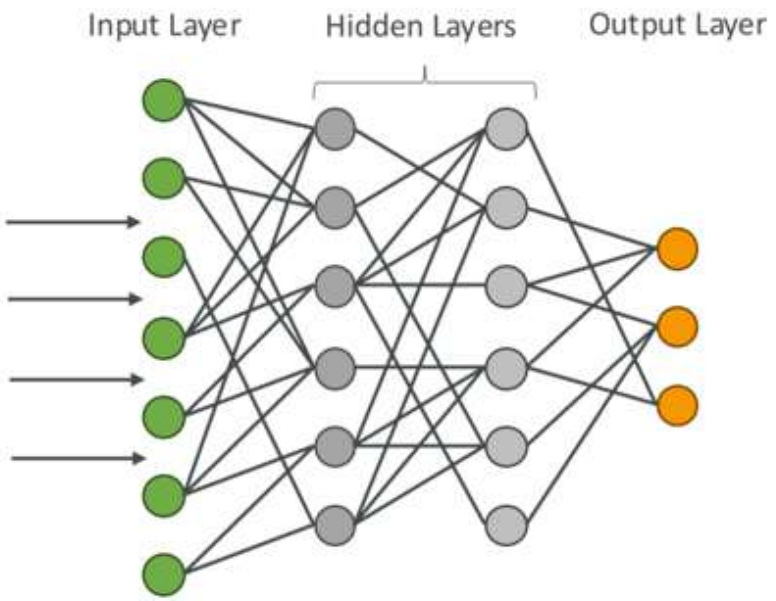
- Neural Networks:

↳ Nodes are tiny units that are connected together & organized in layers.

↳ When NN sees a lot of data, it identifies patterns and changes connections b/w nodes.

↳ Nodes are talking to each other by passing data to the next layer.

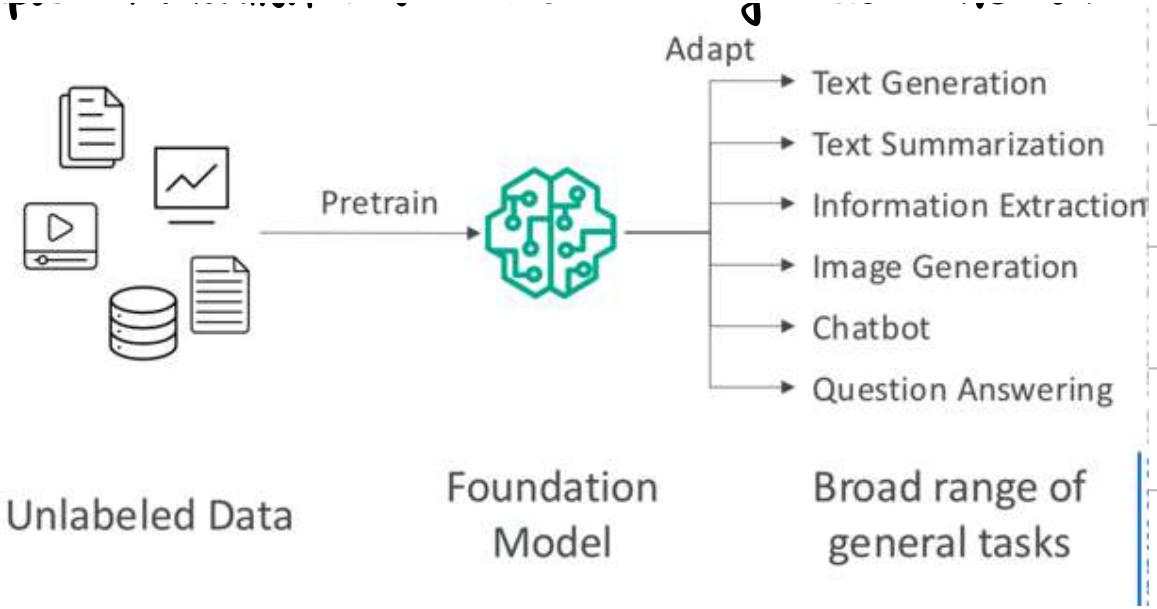
↳ NNs may have billions of nodes.



- Generative AI (Gen-AI):

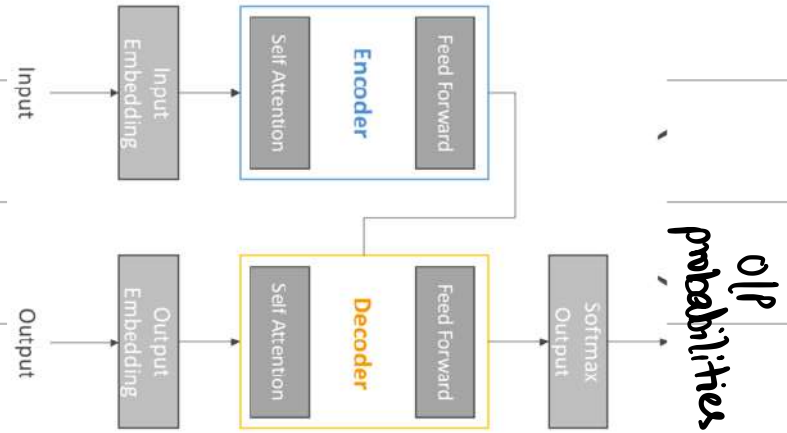
↳ Subset of deep learning

↳ Multi-purpose foundation models backened by neural networks.



↳ They can be fine-tuned if necessary to better fit our use-cases

- Tramformer Model (LLM):



o/p probabilities

↳ Able to process sentence as a whole instead of word by word.

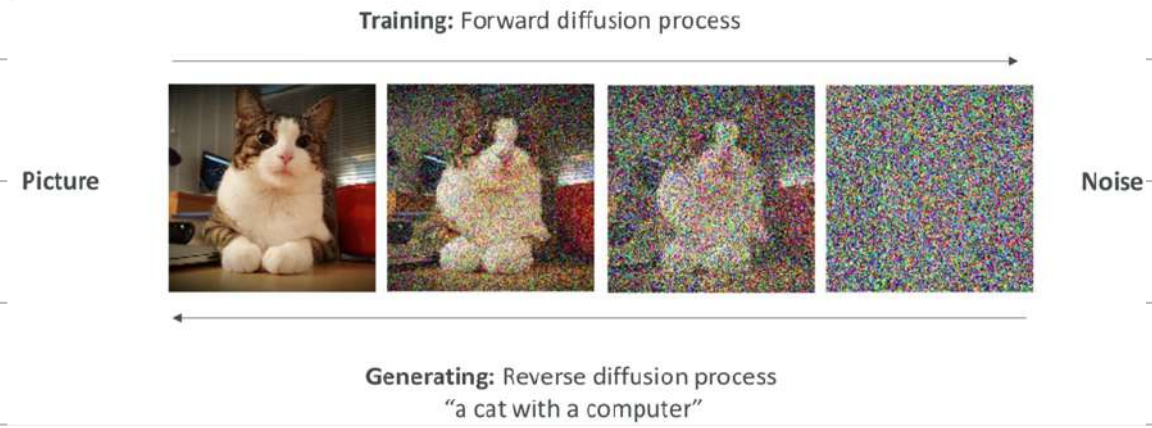
↳ Faster and more efficient text processing (less training time)

↳ It gives relative importance to specific words in a sentence.

↳ Transformer Based LLMs are powerful models that can generate human-like text and is trained on vast amount of data.

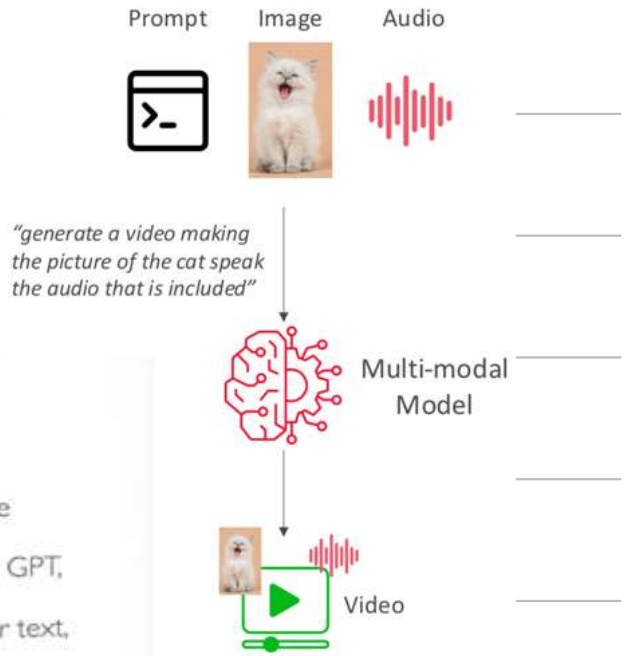
Ex: Google BERT, OpenAPI ChatGPT.

- Diffusion Models:



- Multi-modal models:

↳ different types of input and output:



ML Terms You May Encounter in the Exam

- GPT (Generative Pre-trained Transformer) – generate human text or computer code based on input prompts
- BERT (Bidirectional Encoder Representations from Transformers) – similar intent to GPT, but reads the text in two directions
- RNN (Recurrent Neural Network) – meant for sequential data such as time-series or text, useful in speech recognition, time-series prediction
- ResNet (Residual Network) – Deep Convolutional Neural Network (CNN) used for image recognition tasks, object detection, facial recognition
- SVM (Support Vector Machine) – ML algorithm for classification and regression
- WaveNet – model to generate raw audio waveform, used in Speech Synthesis
- GAN (Generative Adversarial Network) – models used to generate synthetic data such as images, videos or sounds that resemble the training data. Helpful for data augmentation
- XGBoost (Extreme Gradient Boosting) – an implementation of gradient boosting

- Training Data:

↳ We need to have a good data (Garbage in => Garbage out)

↳ Several options to model our data, which will impact types of algorithms we can use.

↳ Labelled Data: Includes both input features and output label (Supervised learning)

↳ Unlabelled Data: Contains only input features (Unsupervised learning)

↳ Structured data: data organized in a structured format, often in rows and columns. Two best examples are tabular data and time series data.

↳ Unstructured data: doesn't follow a specific structure & is text-heavy / multimedia content. It includes text data and image data.

- Supervised Learning:

↳ Learn a mapping function that can predict the output for new unseen input data

↳ It needs labelled data.

i. Regression:

↳ Used to predict numerical value based on I/P data.

↳ the output variable is continuous. Used when you want to predict quantity/real value.

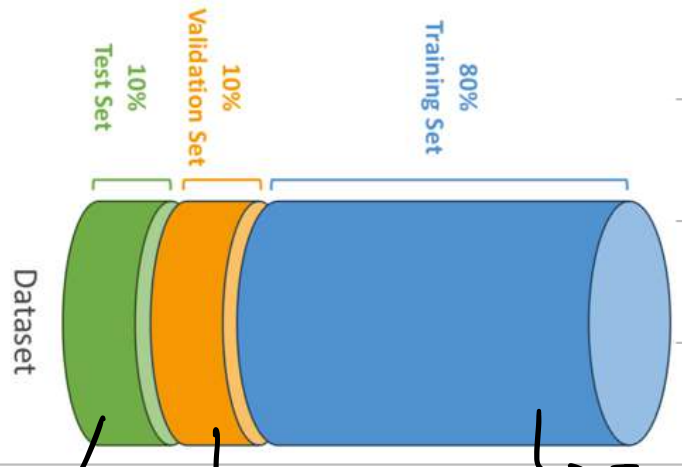
Ex: House Price Prediction, Stock Price Prediction, Weather forecasting.

ii, Classification:

↳ Used to predict the categorical label of input data.

↳ Op variable is discrete.

Ex: Spam Detection (Binary Classification), Classifying animals (Multiclass Classification), labelling a movie like comedy, action ..etc (Multi-label Classification).



↙ final model performance ↘ tune model parameters & validate performance. → Train the model

↳ Feature Engineering:

↳ process of using domain knowledge to select & transform raw data into useful features.

↳ helps enhancing performance of ml models.

↳ Techniques include: feature extraction, feature selection & feature transformation

- Unsupervised learning:

↳ The goal is to discover inherent patterns, structures or relationships within i/p data.

↳ The machine creates groups itself and we just have to label them.

↳ Usecases include customer segmentation, targeted marketing, recommender systems.

↳ feature engineering can help improve the quality of the training.

i. Clustering:

↳ used to group similar data points together in clusters based on their features.

↳ The best example is customer segmentation based on k-means clustering.

ii, Association Rule Learning Technique:

↳ The goal is to identify associations between products & optimize product placement.

↳ It is called Market Basket Analysis using the Apriori algorithm.

iii, Anamoly Detection Technique:

↳ It is used to identify transactions that deviate significantly from typical behavior.

↳ It is mostly used for fraud Detection using Isolation forest.

- Semi-supervised learning:

↳ Use a small amount of labelled data and a large amount of unlabelled data to train the systems.

↳ After that, the partially trained algorithm itself labels the unlabelled data.

↳ This is called psuedo-labeling.

↳ The model is then re-trained on the resulting data mix without being explicitly programmed.

- Reinforcement Learning:

↳ A type of ML where an agent learns to make decisions by performing actions in an environment to maximize cumulative rewards.

Agent : learner & decision-maker

Environment: the external system the agent interacts with.

Action: the choices made by the agent.

Reward: the feedback from the environment based on agent's action.

State: the current situation of the environment

Policy: the strategy the agent uses to determine actions based on the state.

↳ The learning process is as following:

1. The agent observes the current state of the environment.

2. It selects an action based on its policy.

3. The environment transitions to a new state and provides a Reward.

4. The agent updates its policy to improve future decisions.

↳ The goal is to maximize cumulative reward over time.

↳ Used in gaming, robotics, finance, healthcare and autonomous vehicles.

- Reinforcement learning from human feedback:

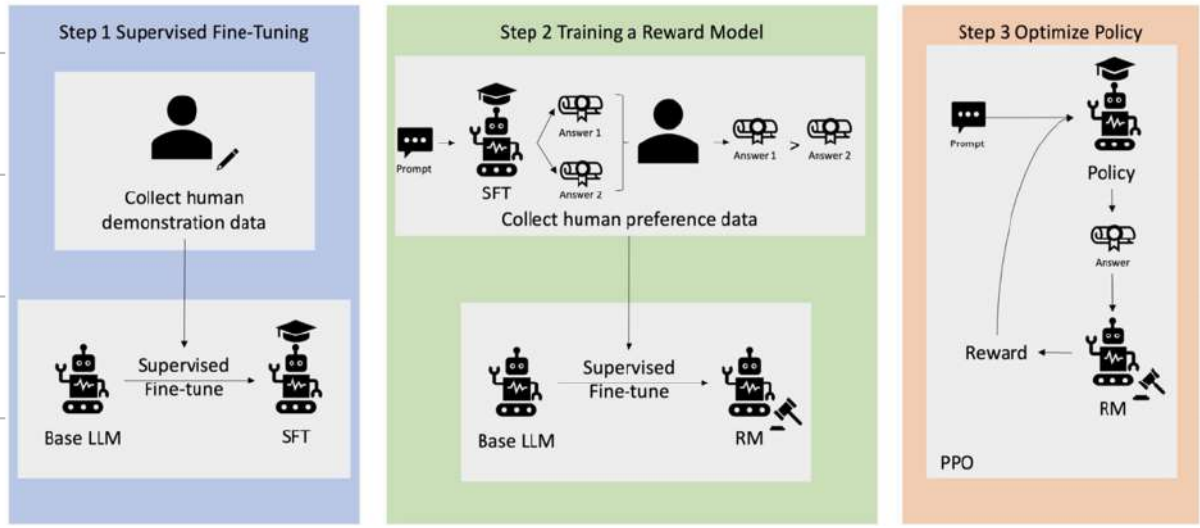
↳ Use human feedback to help ML models to self-learn more efficiently.

↳ RHFL incorporates human feedback in reward function, to be more aligned to our goals.

1. The model's responses are compared to human's responses.

2. A human will assess the quality of the model's responses.

↳ Used throughout GenAI applications like LLMs & significantly enhances model performance.



↳ The 4 main steps of RLHF are

1. Data Collection.

2. Supervised fine-tuning of a language model.

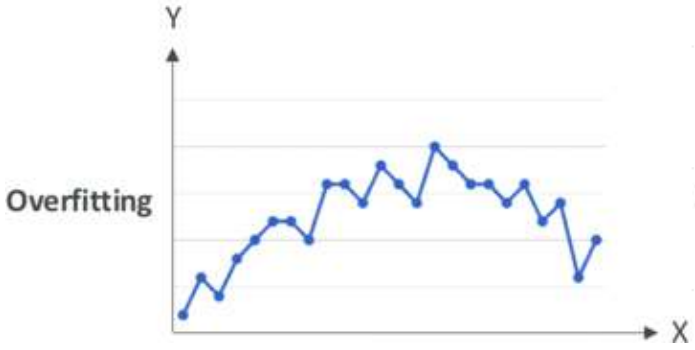
3. Build a seperate reward model.

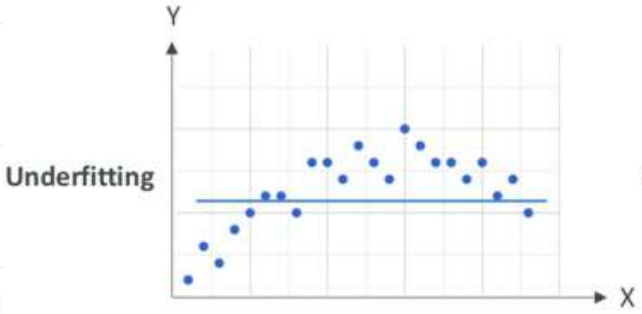
4. Optimize the language model with the reward-based model.

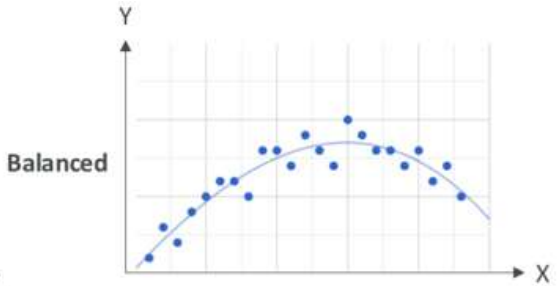
- Overfitting: Performs well on training data but doesn't work well on evaluation data.

- Underfitting: Performs poorly on training data. Model is too simple or poor data features.

- Balanced fit: Neither overfitting nor underfitting







- Bias:

↳ Difference or error between predicted and actual value.

↳ High Bias means model doesn't closely match the training data i.e underfitting.

↳ To reduce bias we either use more complex model or increase no. of features.

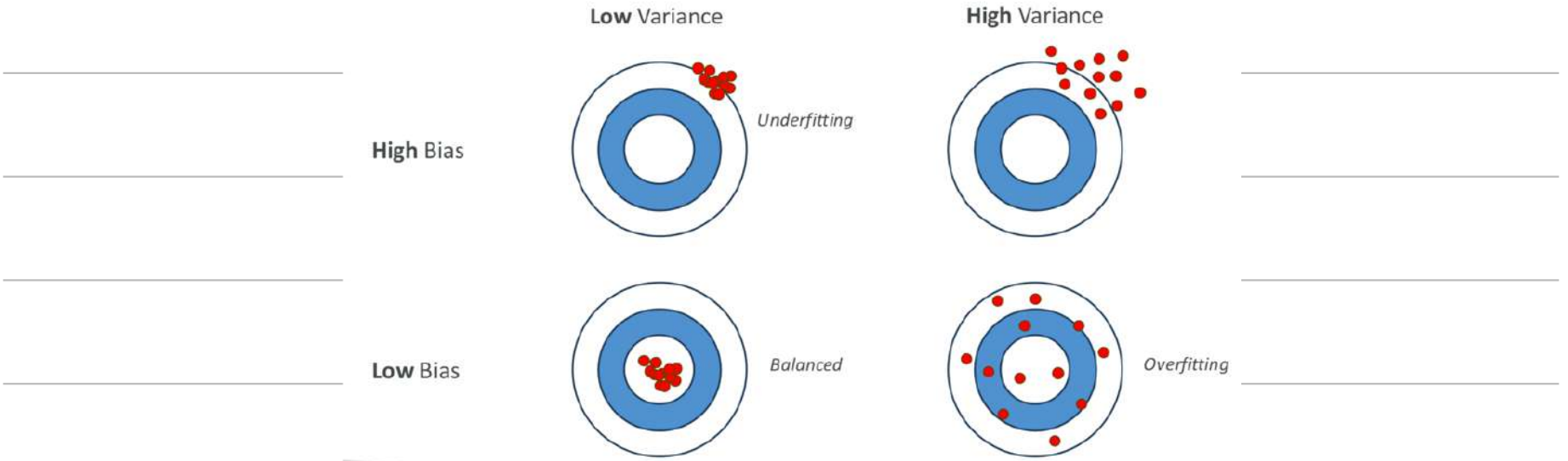
- Variance:

↳ how much the performance of a model changes if trained on a different dataset which has a similar distribution.

↳ high variance when model is very sensitive to changes in training data i.e overfitting.

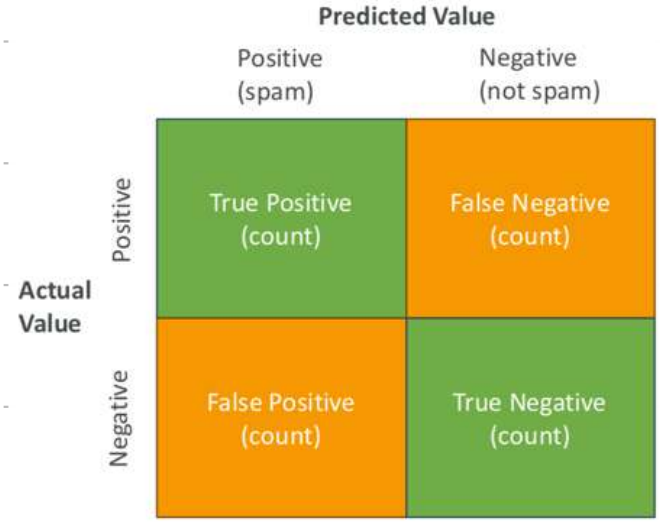
↳ To reduce variance we can use feature selection, split into train & test sets multiple times.

- We're trying to make a model with low bias and low variance (balanced fit)



- Model Evaluation Metrics:

1. Confusion Matrix:



TP - spam predicted as spam

FN - spam predicted as not spam

FP - not spam predicted as spam

TN - not spam predicted as not spam

↳ Confusion matrix can be multidimensional as well & is best to evaluate classification models.

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}$$

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$ (Rarely used)

Precision: best when FP are costly

Recall: best when FN are costly

F1 score: when you need balance b/w precision and recall, especially in imbalanced datasets

Accuracy: best for balanced datasets

AUC-ROC: (Area under the curve - receiver operator curve)

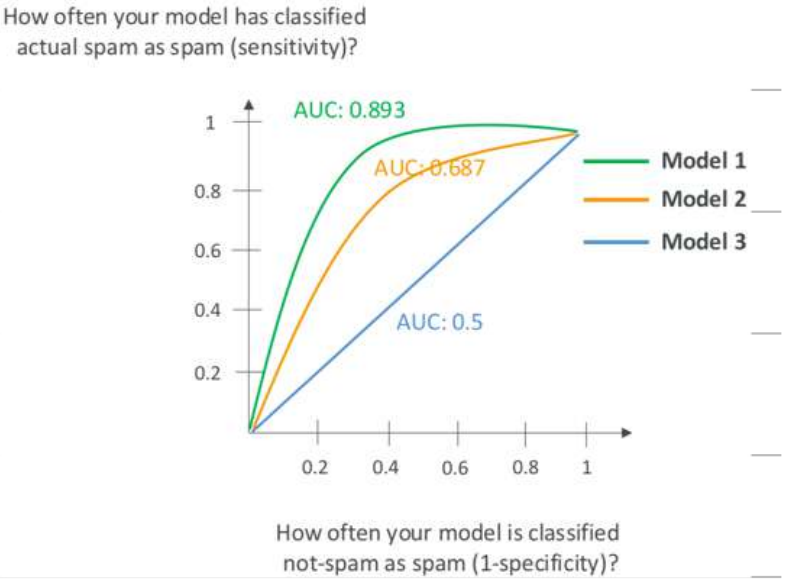
↳ Value from 0 to 1 (perfect model)

↳ Uses sensitivity (TP rate) and "1-specificity" (FP rate)

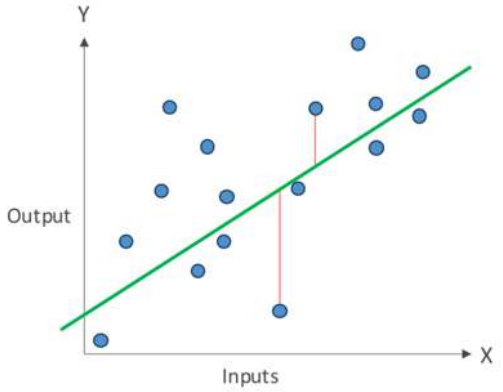
↳ AUC-ROC shows what the curve for true positive compared to FP looks like

at various thresholds, with multiple confusion matrixes.

↳ You compare them to one another to find out the threshold you need for your business use case.



Model Evaluation – Regressions Metrics:



MAE = Mean Absolute Error
between predicted and actual values

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAPE =
Mean Absolute Percentage Error

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{\hat{y}_i}\right|$$

RMSE =
Root mean squared error (RMSE)

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}}$$

R^2 (R Squared): explains variance in your model
R^2 close to 1 means predictions are good

↳ All of these metrics are used for evaluating models that predict continuous value.

↳ MAE, MAPE, RMSE measure the error i.e how "accurate" the model is.

- Inferencing:

↳ Inferencing is when a model is making prediction on new data.

i, Real Time:

↳ They need to make a decision quickly

↳ Speed > Accuracy.

ii, Batch:

↳ Large amount of data analyzed all at once.

↳ Often used for data analysis

↳ Accuracy > Speed.