BOSSCODER ACADEMY

LEARN MACHINE LEARNING

through these

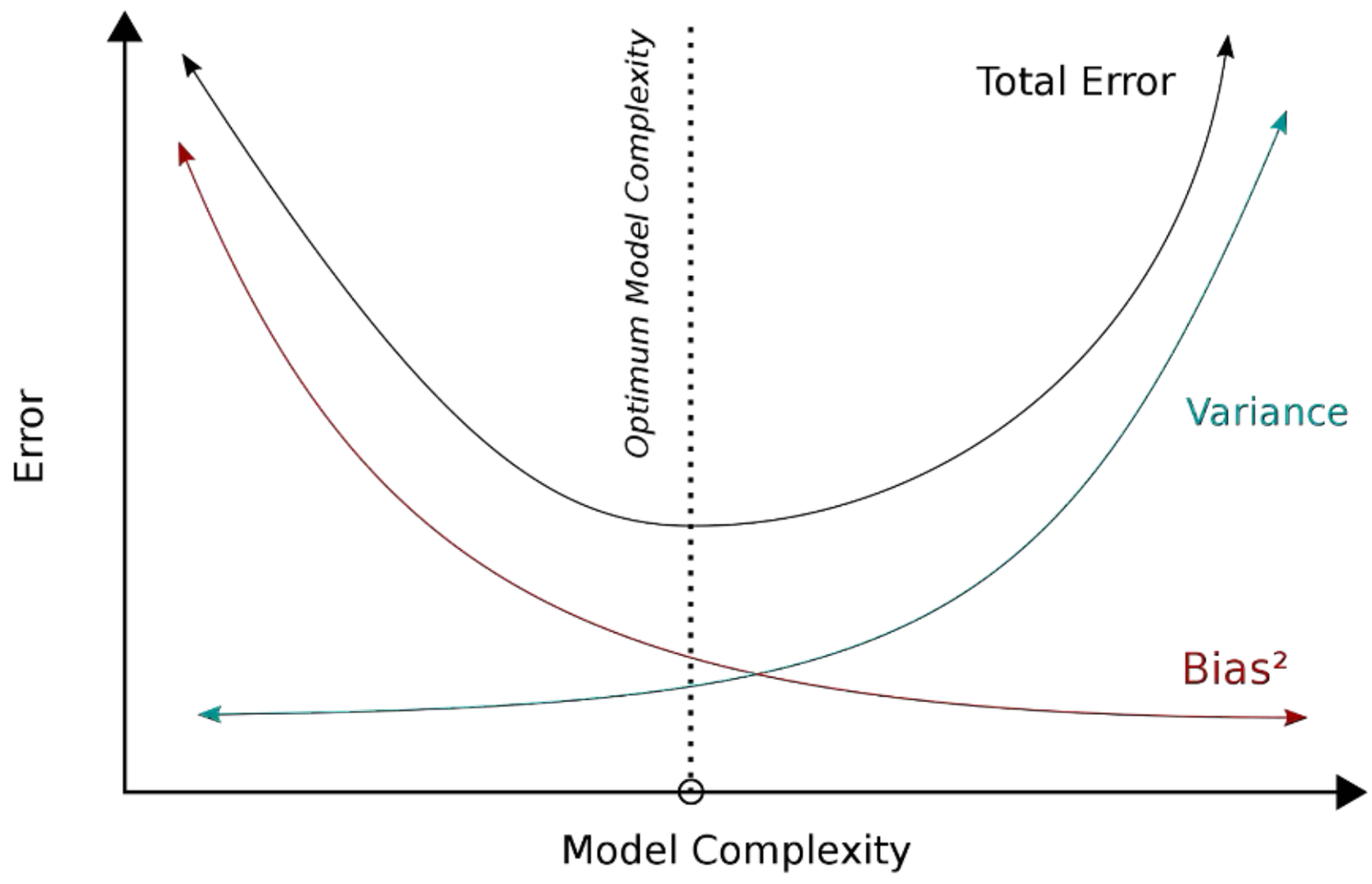
Most Asked Interview Questions

Q. 1

Explain Bias-Variance Tradeoff.

Ans. 1

The bias-variance tradeoff represents the balance between the model's ability to generalize across different datasets (bias) and its sensitivity to small fluctuations in the training set (variance). A high-bias model is too simple and underfits the data, missing the underlying trend. A high-variance model is too complex, overfitting the data and capturing noise as if it were a real pattern. The goal is to find a sweet spot that minimizes the total error.

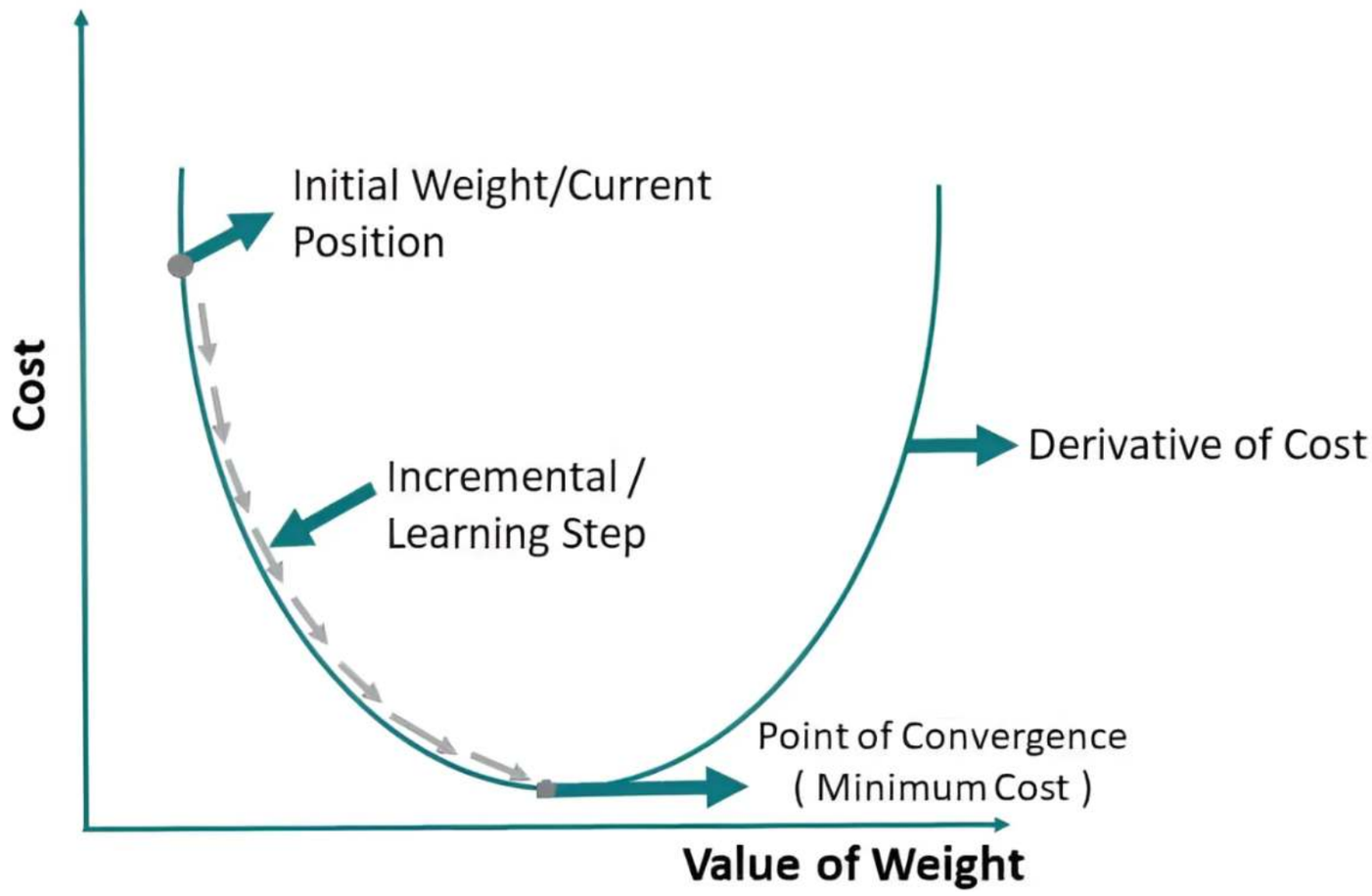


Q. 2

How does Gradient Descent Work?

Ans. 2

Gradient Descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of the steepest descent as defined by the negative of the gradient. In machine learning, it's used to find the parameters of a model that minimize the cost function. The learning rate determines the size of the steps taken to reach the minimum.

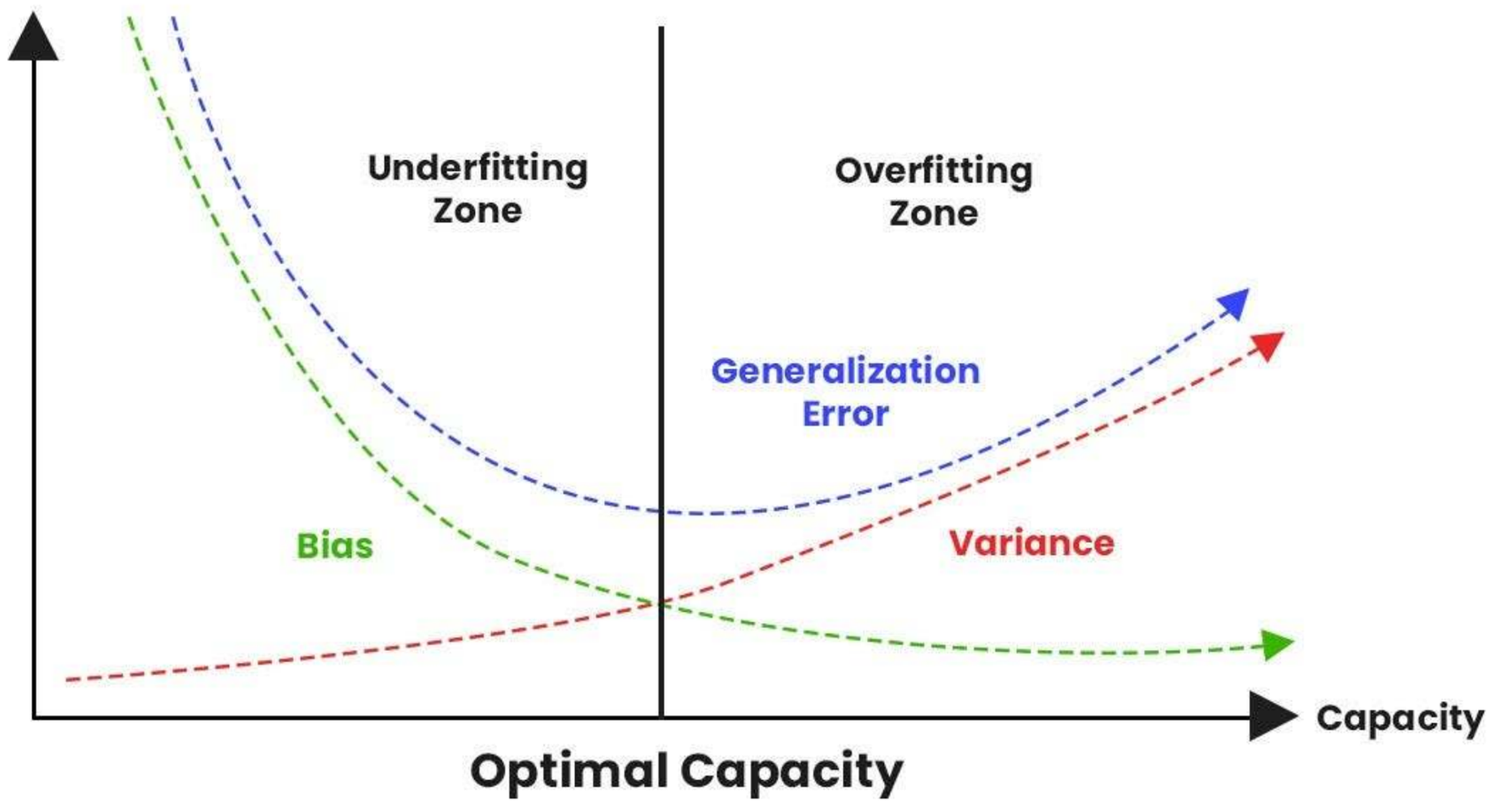


Q. 3

What is Regularization? Give Examples.

Ans. 3

Regularization is a technique used to prevent overfitting by adding a penalty on the size of the coefficients. The penalty term discourages complex models and thus reduces variance without substantially increasing bias. Examples include L1 regularization (Lasso), which adds the absolute value of the magnitude of coefficients as penalty, and L2 regularization (Ridge), which adds the square of the magnitude of coefficients.



Q. 4

Explain the Difference between Bagging and Boosting.

Ans. 4

Both Bagging and Boosting are ensemble techniques to improve model predictions, but they work differently.

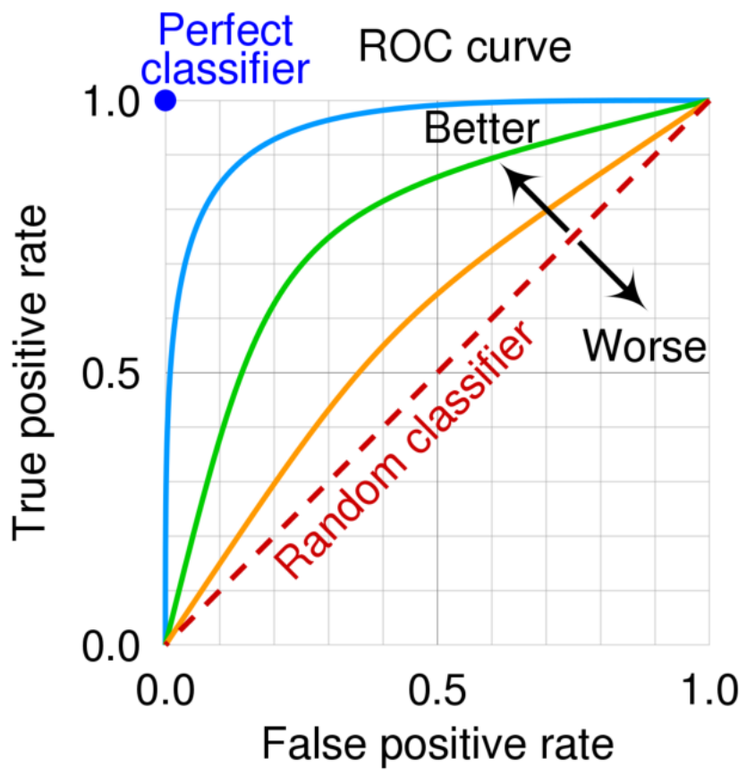
S.NO	Bagging	Boosting
1	The simplest way of combining predictions that belong to the same type.	A way of combining predictions that belong to the different types.
2	Aim to decrease variance, not bias.	Aim to decrease bias, not variance.
3	Each model receives equal weight.	Models are weighted according to their performance.
4	Each model is built independently.	New models are influenced by the performance of previously built models.
5	Different training data subsets are randomly drawn with replacement from the entire training dataset.	Every new subset contains the elements that were misclassified by previous models.
6	Bagging tries to solve the over-fitting problem.	Boosting tries to reduce bias.
7	If the classifier is unstable (high variance), then apply bagging.	If the classifier is stable and simple (high bias) the apply boosting.
8	Example: The Random Forest model uses Bagging.	Example: The AdaBoost uses Boosting techniques

Q. 5

Describe the ROC Curve and AUC.

Ans. 5

The ROC Curve (Receiver Operating Characteristic Curve) is a graph showing the performance of a classification model at all classification thresholds. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR). AUC (Area Under the ROC Curve) measures the entire two-dimensional area underneath the entire ROC curve and provides an aggregate measure of performance across all possible classification thresholds. An AUC of 1 represents a perfect model; an AUC of 0.5 represents a worthless model.

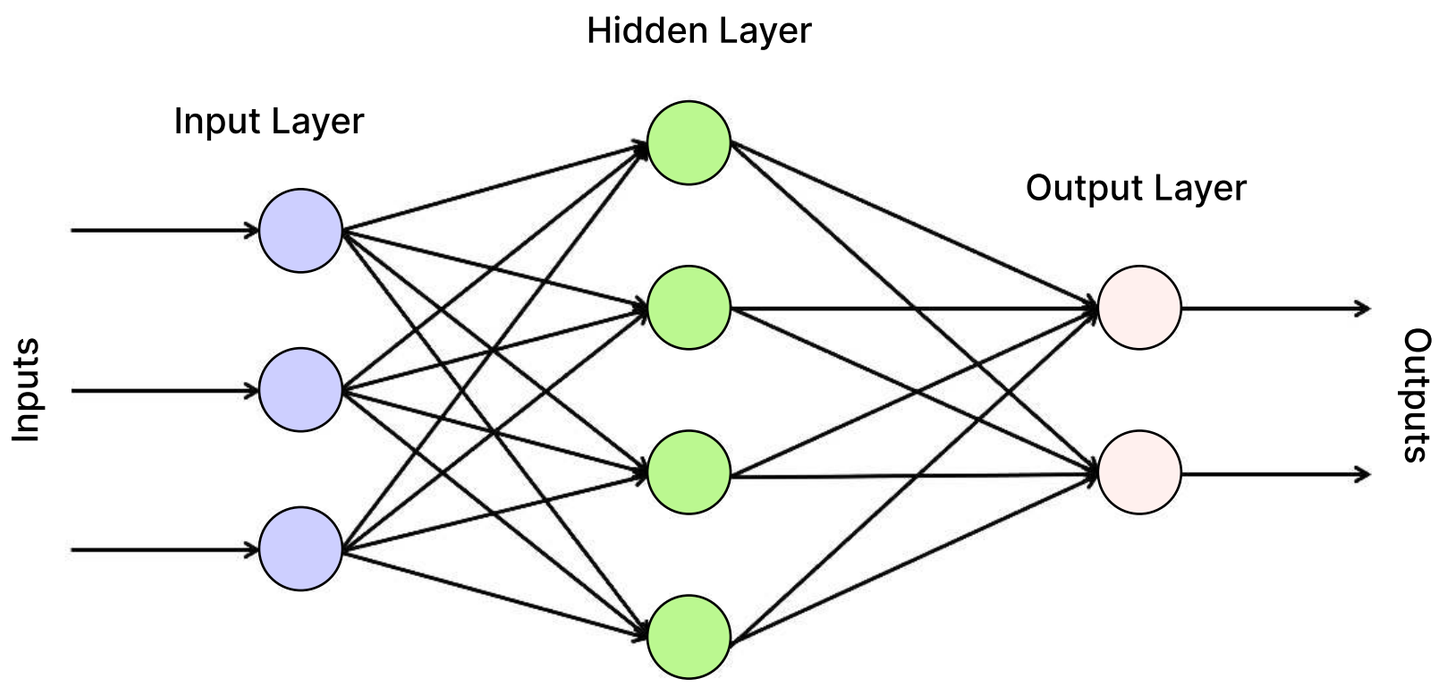


Q. 6

What are Convolutional Neural Networks (CNNs) and where are they used?

Ans. 6

CNNs are a class of deep neural networks, most commonly applied to analyzing visual imagery. They use a mathematical operation called convolution in at least one of their layers. A key feature of CNNs is their ability to automatically and adaptively learn spatial hierarchies of features from images. CNNs are widely used in image and video recognition, recommender systems, and natural language processing.

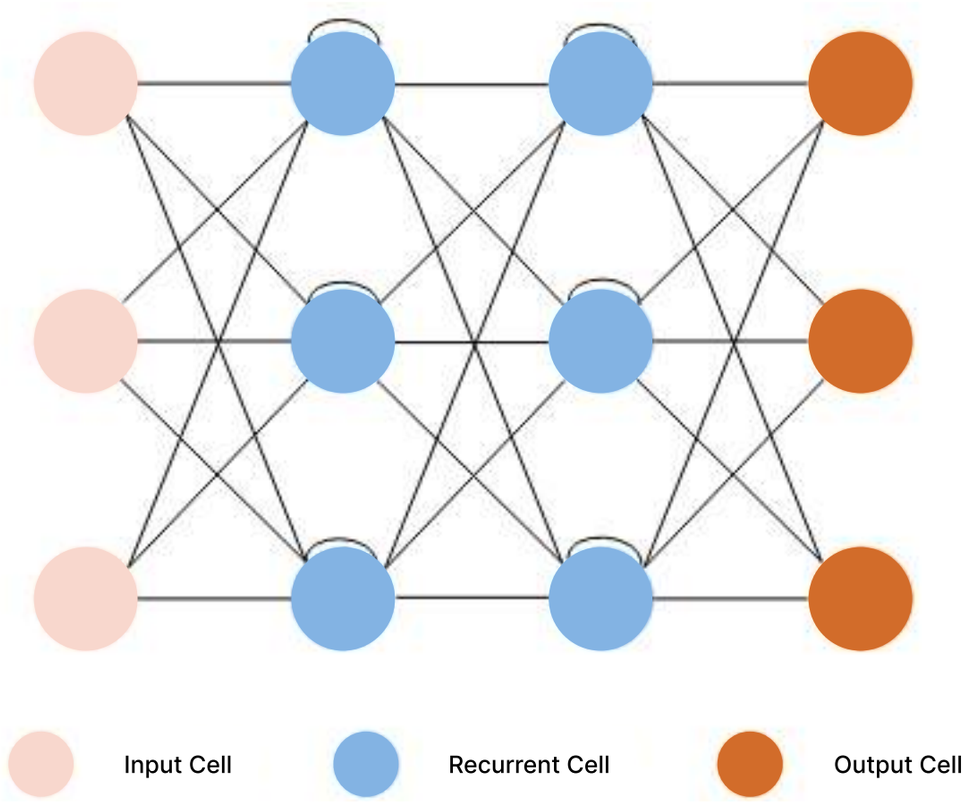


Q. 7

How do Recurrent Neural Networks (RNNs) differ from CNNs?

Ans. 7

While CNNs are primarily used for spatial data (like images), RNNs are designed to work with sequence data (like text or time series). RNNs have loops allowing information to persist, meaning they can keep track of information in a sequence, making them ideal for tasks like language modeling and text generation. Unlike CNNs, RNNs can handle inputs of varying lengths.

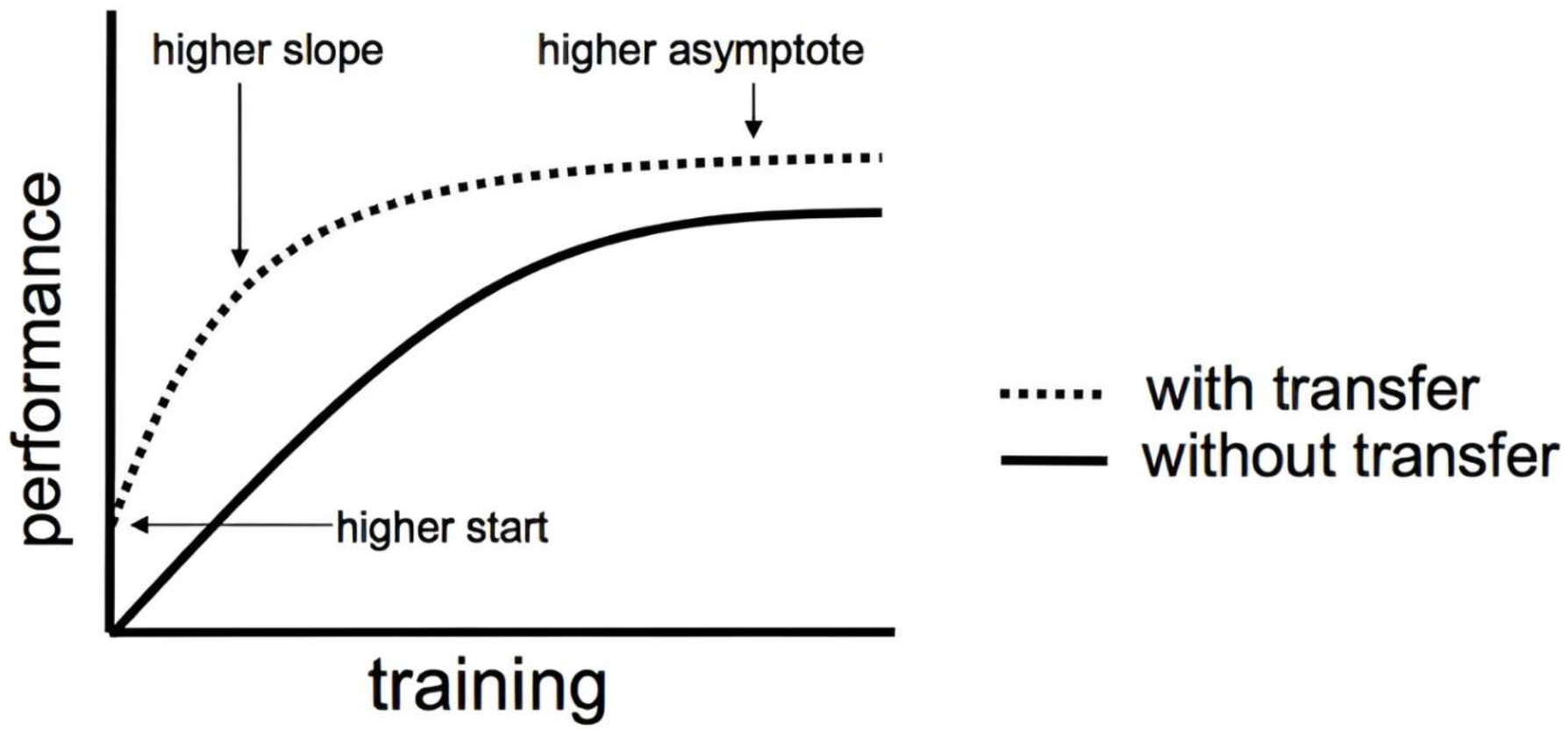


Q. 8

Explain the concept of Transfer Learning.

Ans. 8

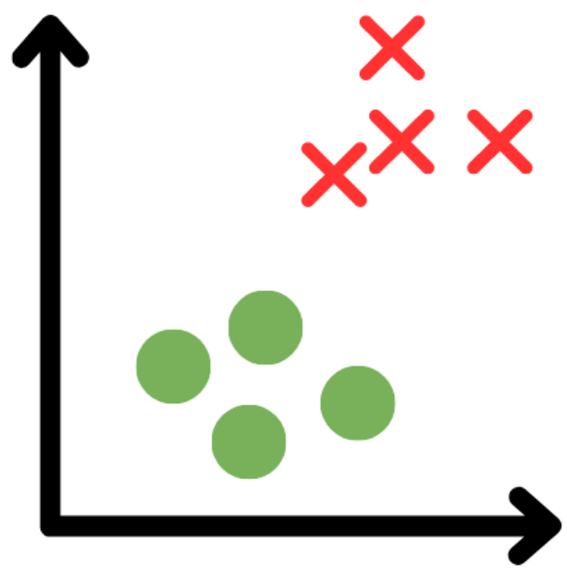
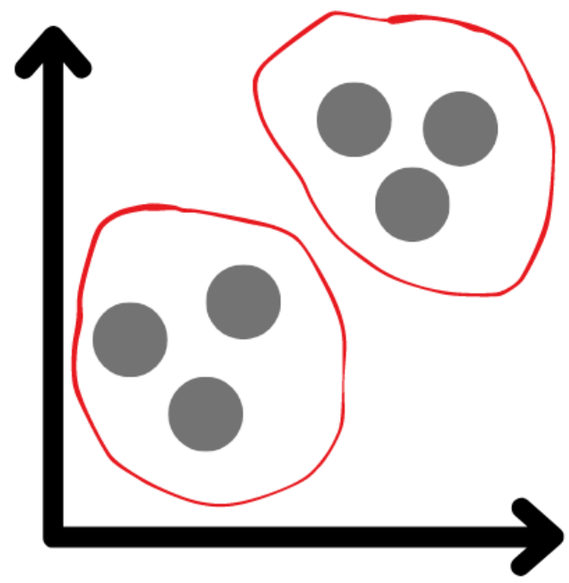
Transfer Learning involves taking a pre-trained model (trained on a large dataset) and fine-tuning it with a smaller dataset for a similar or different task. This approach allows leveraging learned feature maps without starting from scratch, saving time and computational resources. It's particularly useful in deep learning where large datasets and extensive training are usually required.



Q. 9

What is the difference between Supervised and Unsupervised learning?

Ans. 9

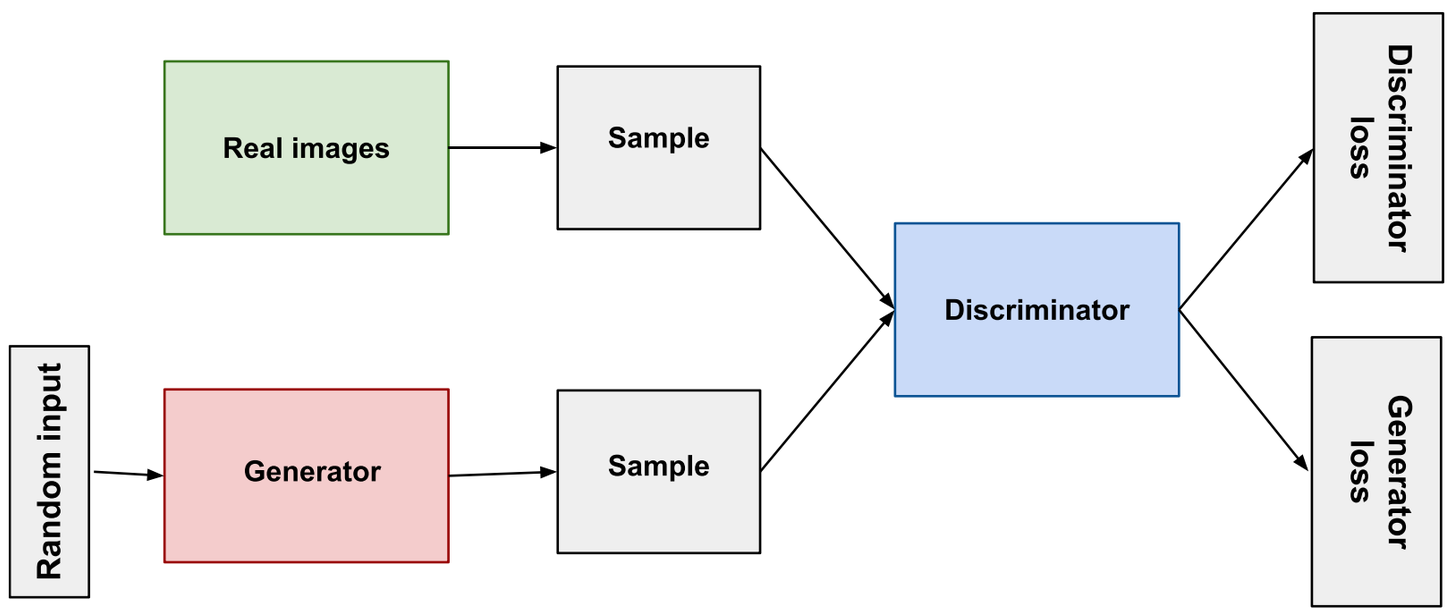
Supervised learning	Unsupervised learning
Input data is labeled	Input data is unlabeled
Has a feedback mechanism	Has no feedback mechanism
Data is classified based on the training dataset	Assigns properties of given data to classify it
Divided into Regression & Classification	Divided into Clustering & Association
Used for prediction	Used for analysis
Algorithms include: decision trees, logistic regressions, support vector machine	Algorithms include: k-means clustering, hierarchical clustering, apriori algorithm
A known number of classes	A unknown number of classes
	

Q. 10

What are GANs, and how do they work?

Ans. 10

Generative Adversarial Networks (GANs) consist of two models: a generative model that captures the data distribution, and a discriminative model that estimates the probability that a sample came from the training data rather than the generative model. The two models are trained simultaneously in a game; the generator tries to produce data that is indistinguishable from real data, while the discriminator tries to distinguish between real and fake data.



Case-Study Based Questions

How would you approach building a model to predict stock prices for the next day?

Approach

This problem requires analyzing time series data. One approach could be to use RNNs or LSTM (Long Short Term Memory) networks to capture temporal dependencies and trends in historical price data. Feature engineering will also be crucial, incorporating not just price but also volume, historical averages, and potentially external data like economic indicators.

Imagine you need to classify emails into spam and not-spam. How would you design this system?

Approach

This is a classic example of a supervised learning classification problem. One could use Naive Bayes, SVM, or deep learning models like CNNs for text classification. The key is to convert emails into a suitable format for these models, using techniques like TF-IDF or word embeddings. Performance can be improved by including more contextual features and fine-tuning the model.

You’re tasked with designing a recommendation system for a streaming service. What approach would you take?

Approach

A collaborative filtering approach could be initially adopted, leveraging user-item interactions. Matrix factorization techniques like SVD can be employed here.

For a more sophisticated solution, one could use deep learning-based models that can also incorporate content-based features (like genre or director of a movie) to make recommendations even for new users or items (the cold start problem).

Develop a strategy for a model that can translate spoken language in real-time.

Approach

This problem involves both speech recognition and machine translation. An end-to-end deep learning approach could be used, where an RNN model first transcribes speech to text, and then another model translates the text to the target language.

Attention mechanisms and Transformer models would be crucial for handling long sequences and improving translation accuracy.

How would you build a model to identify objects in a video in real-time?

Approach

This requires a combination of object detection and video processing techniques. CNNs, specifically models like YOLO (You Only Look Once) or SSD (Single Shot Detector), could be used for object detection. These models need to be optimized for real-time performance, considering the computational constraints. Additionally, incorporating temporal information using RNNs or 3D CNNs could improve detection accuracy by considering object movement over time.

Each of these case-study questions touches on different aspects of machine learning and data science, requiring a combination of theoretical knowledge and practical application. Approaching these problems involves understanding the data, choosing the right model, feature engineering, and iterative improvement based on performance metrics.

BOSSCODER ACADEMY

WHY BOSSCODER?

750+ Alumni placed at Top Product-based companies.

More than **136% hike** for every **2 out of 3** working professional.

Average package of **24LPA.**

The syllabus is most **up-to-date** and the list of problems provided covers **all important topics.**

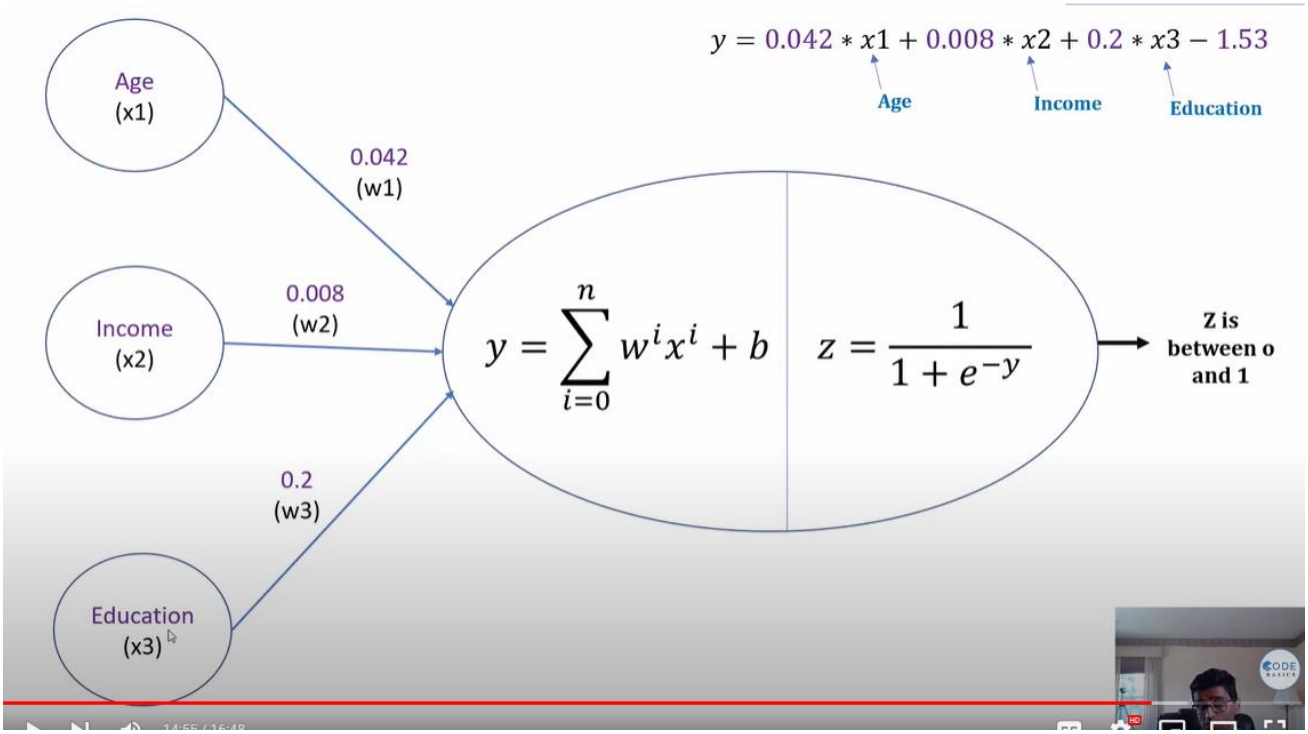
Lavanya
Meta

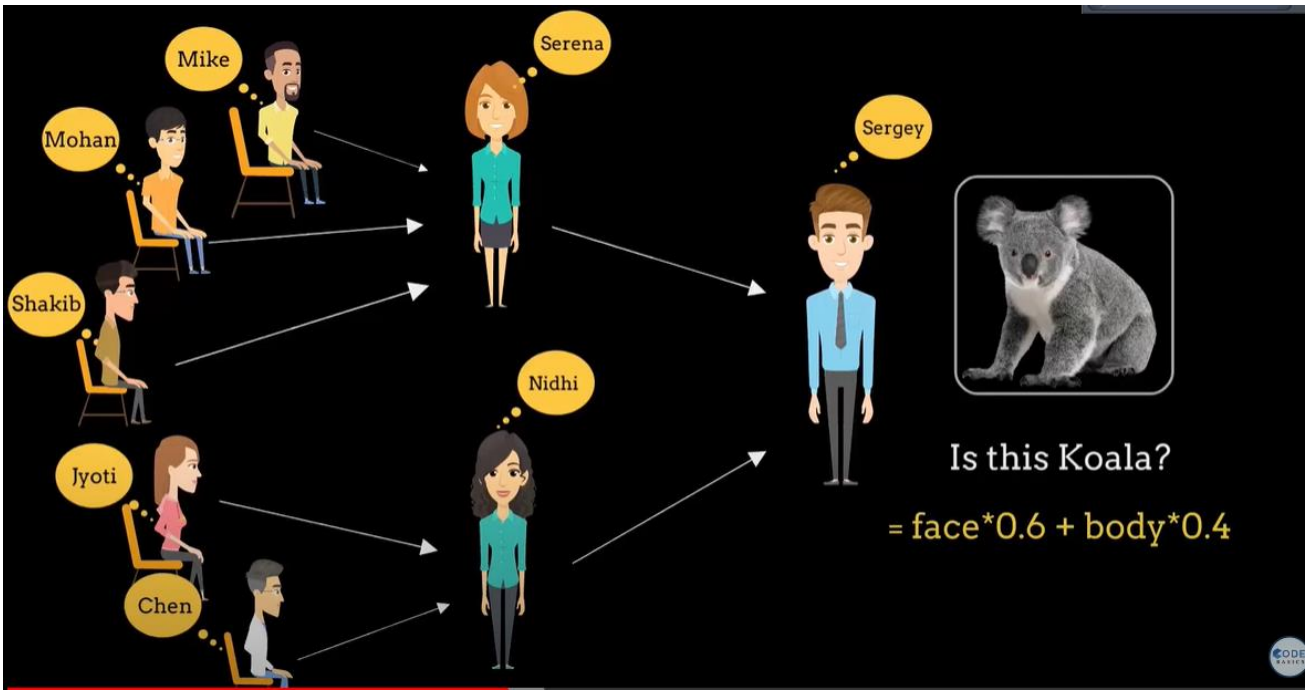
Course is very well **structured** and **streamlined** to crack any **MAANG** company

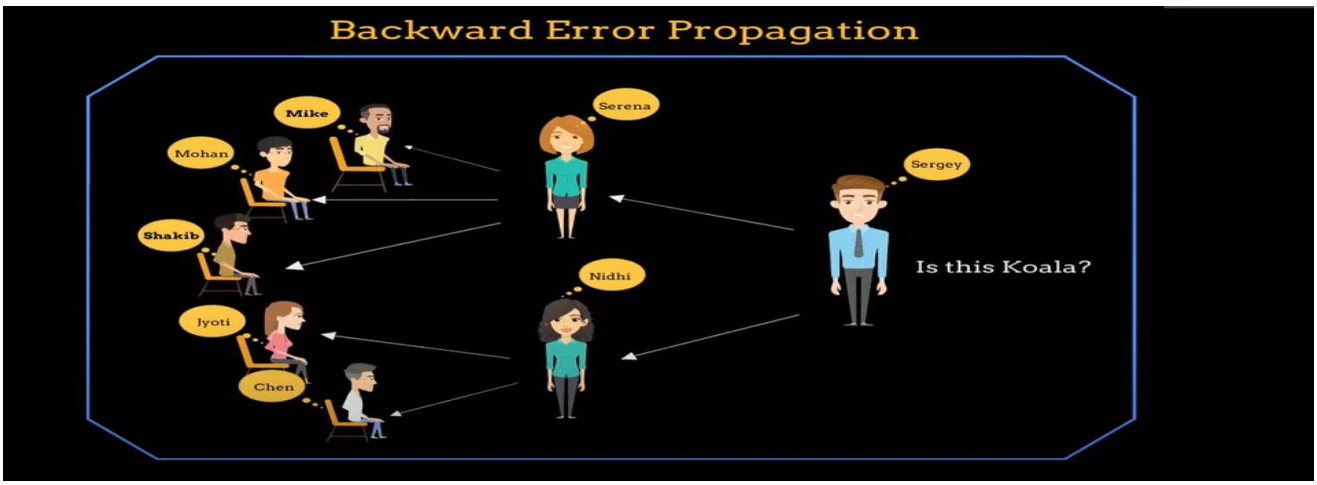
Rahul .
Google

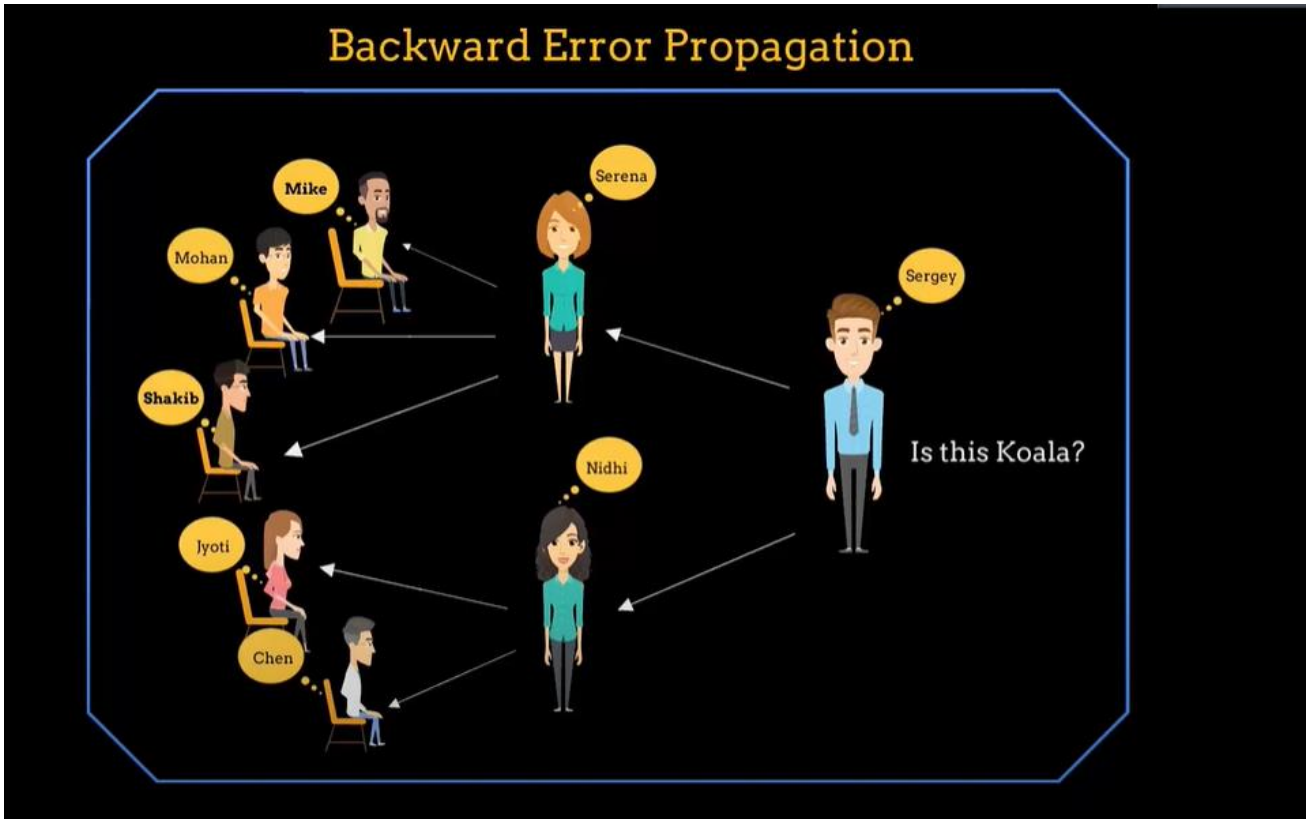
EXPLORE MORE

DEEP LEARNING BASICES

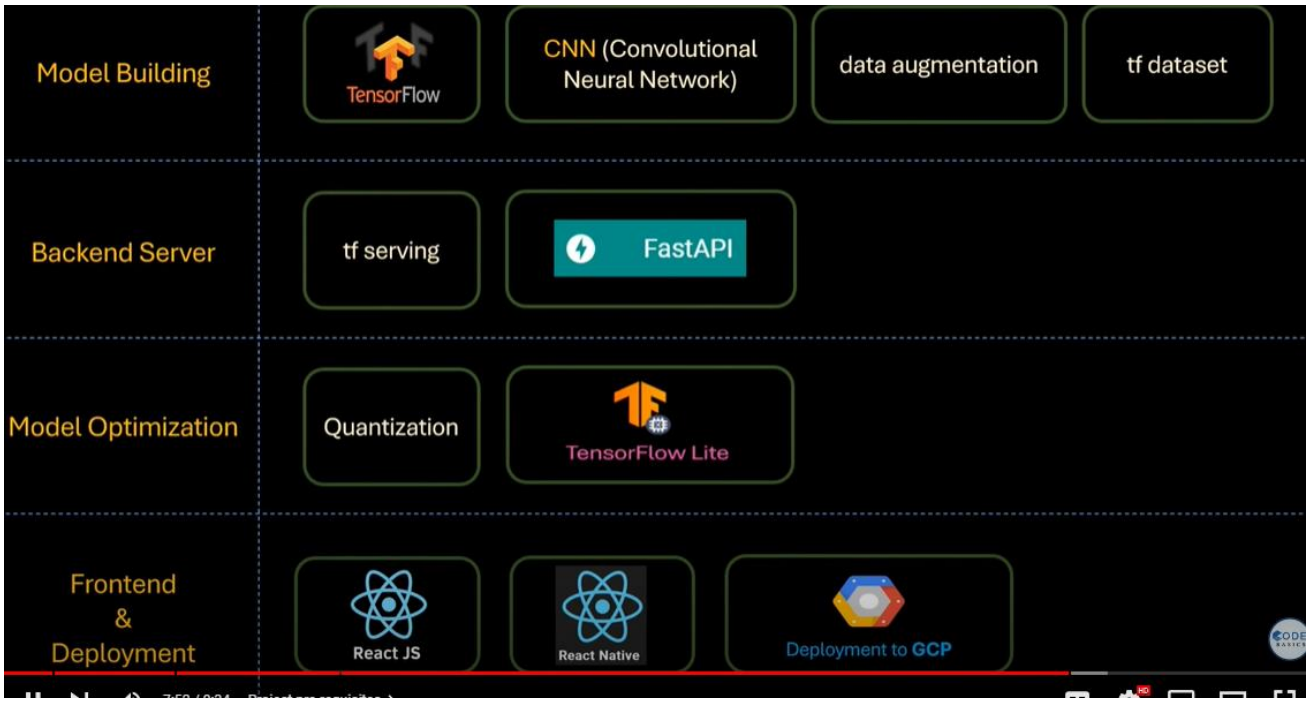


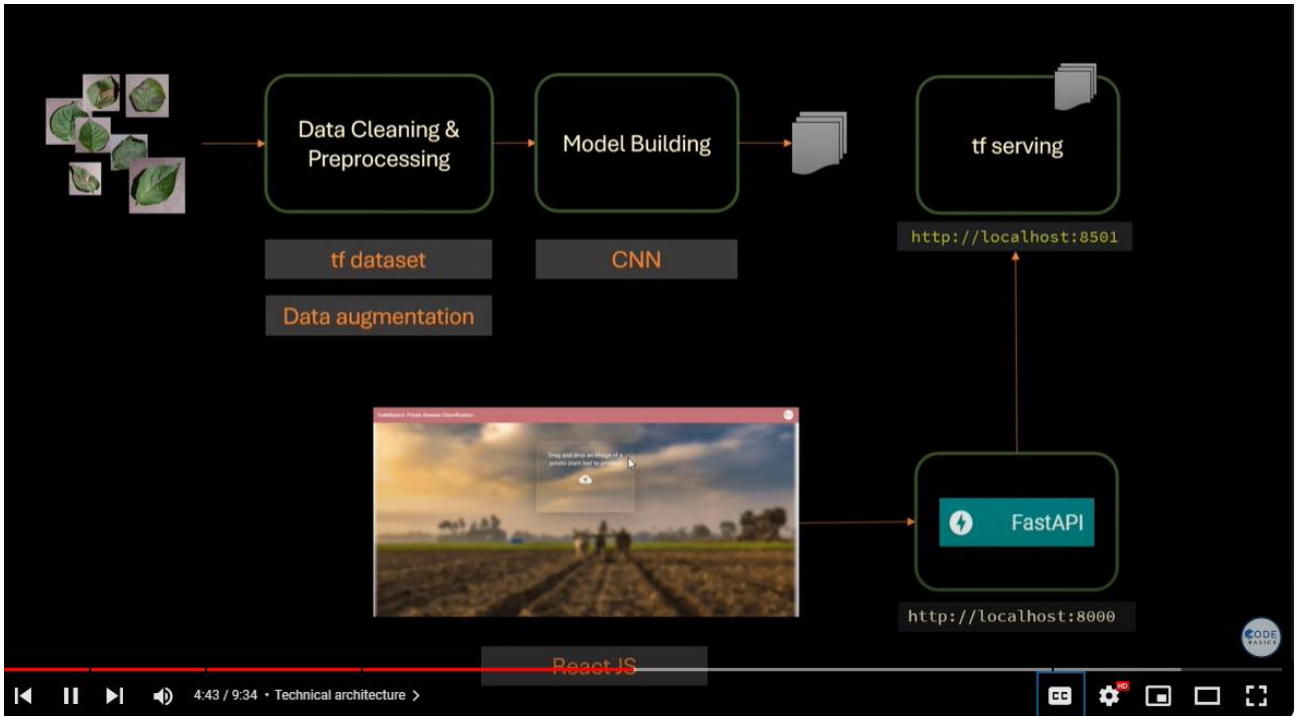


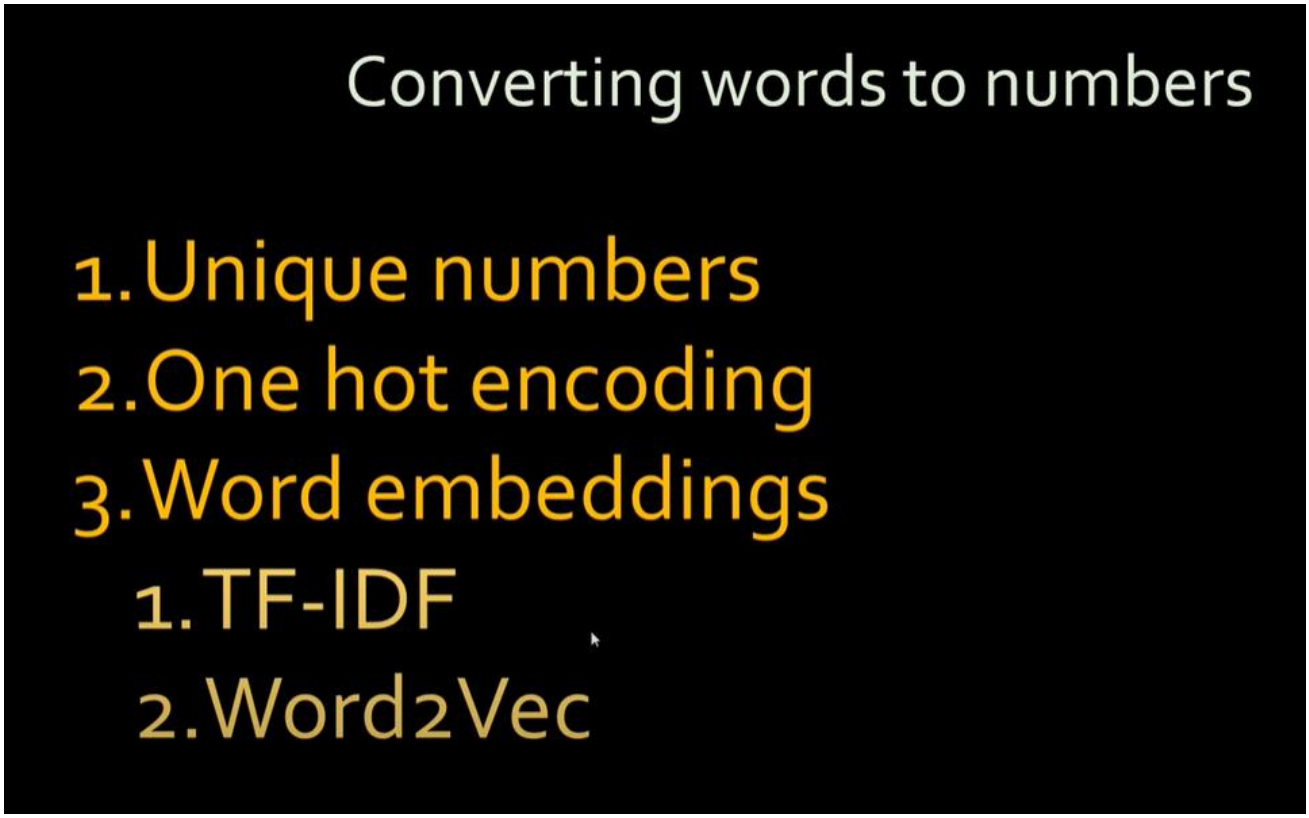


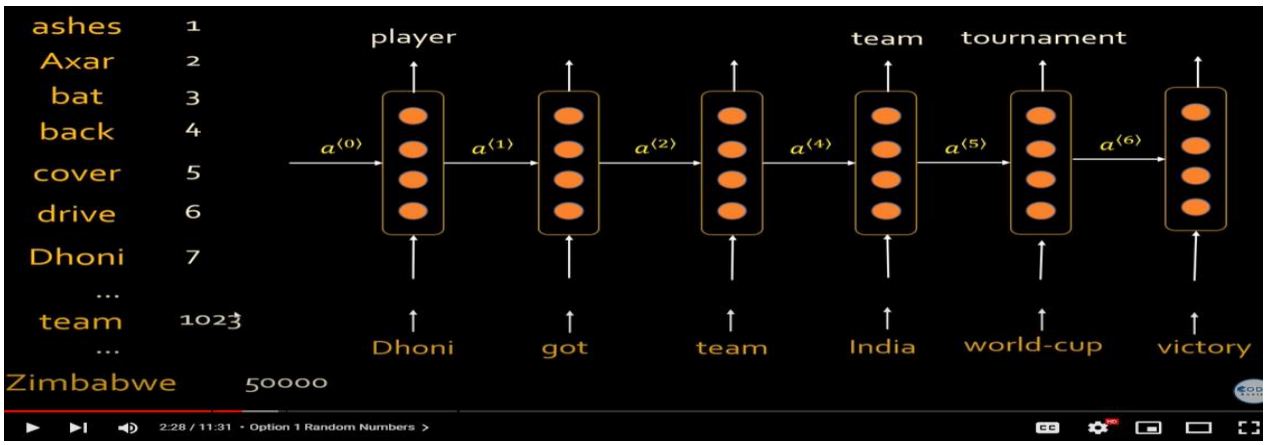


WHY NEED CNN : https://www.youtube.com/watch?v=zfiSAzpy9NM&t=1151s




Data Cleaning & Preprocessing
Model Building
tf serving
http://localhost:8501
tf dataset
CNN
Data augmentation
FastAPI
http://localhost:8000
React JS
4:43 / 9:34 • Technical architecture >


Converting words to numbers
1. Unique numbers
2. One hot encoding
3. Word embeddings
1. TF-IDF
2. Word2Vec


ashes 1
Axar 2
bat 3
back 4
cover 5
drive 6
Dhoni 7
...
team 1023
...
Zimbabwe 50000
player
team
tournament
Dhoni
got
team
India
world-cup
victory
2:28 / 11:31 • Option 1 Random Numbers >

Issue with option 1: unique numbers

Numbers are random. They don't capture relationship between words

	ashes	Axar	Bat	back	cover	drive	dhoni	..	Zimbabve
ashes	1	0	0	0	0	0	0	...	0
Axar	0	1	0	0	0	0	0	...	0
Bat	0	0	1	0	0	0	0	...	0
Back	0	0	0	1	0	0	0	...	0
...									
Zimbabwe	0	0	0	0	0	0	0	...	1

Issue with option 2: one hot encoding

1. Doesn't capture relationship between words
2. Computationally in-efficient

Dhoni

Person: 1
Healthy/fit: 0.9
Location: 0
Has two eyes: 1
Has government: 0

Cummins

Person: 1
Healthy/fit: 0.87
Location: 0
Has two eyes: 1
Has government: 0

Australia

Person: 0
Healthy/fit: 0.7
Location: 1
Has two eyes: 0
Has government: 1

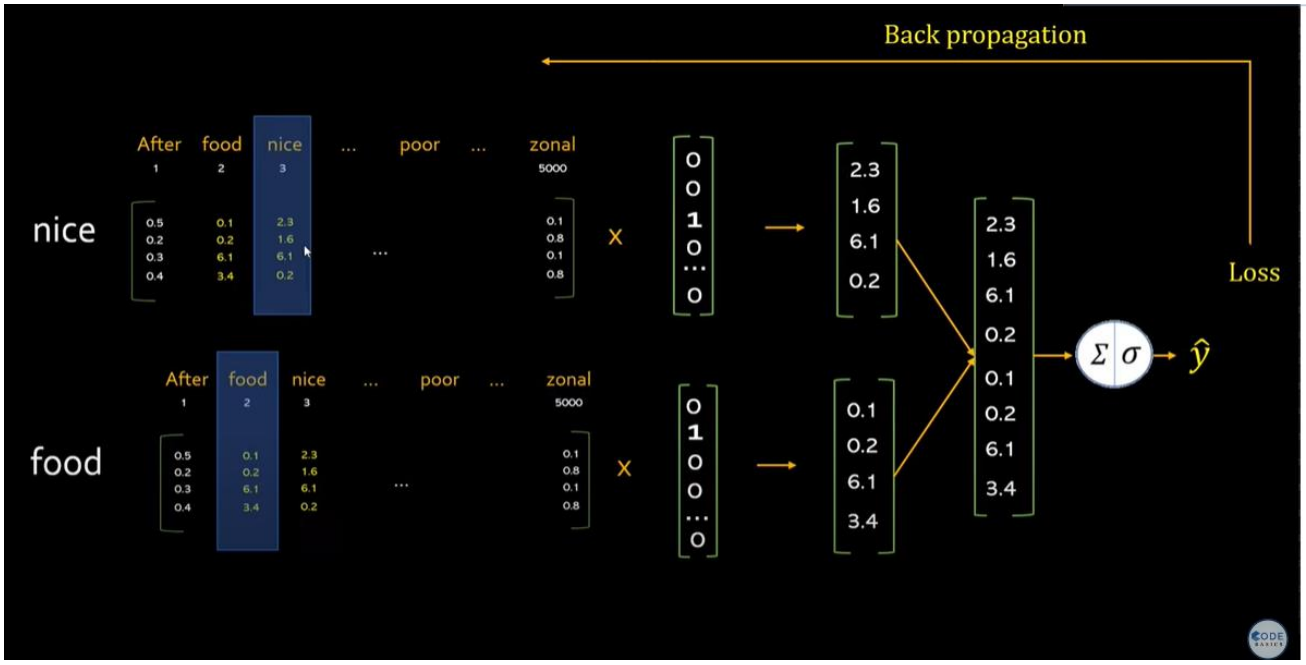
Dhoni

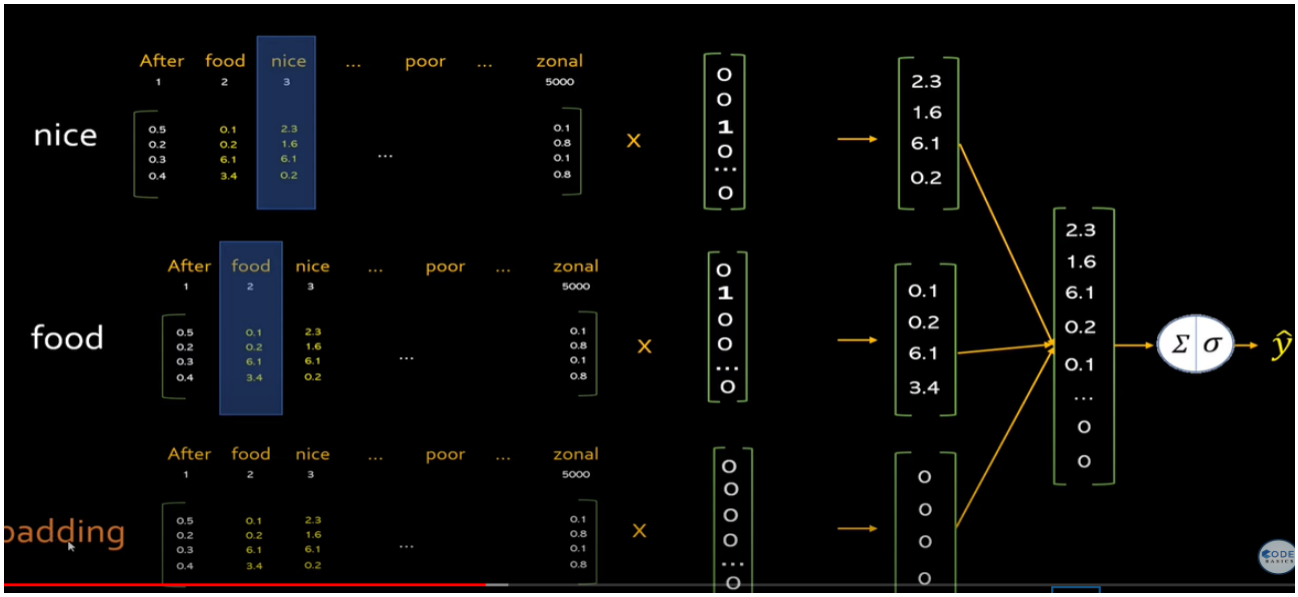
Cummins

Australia

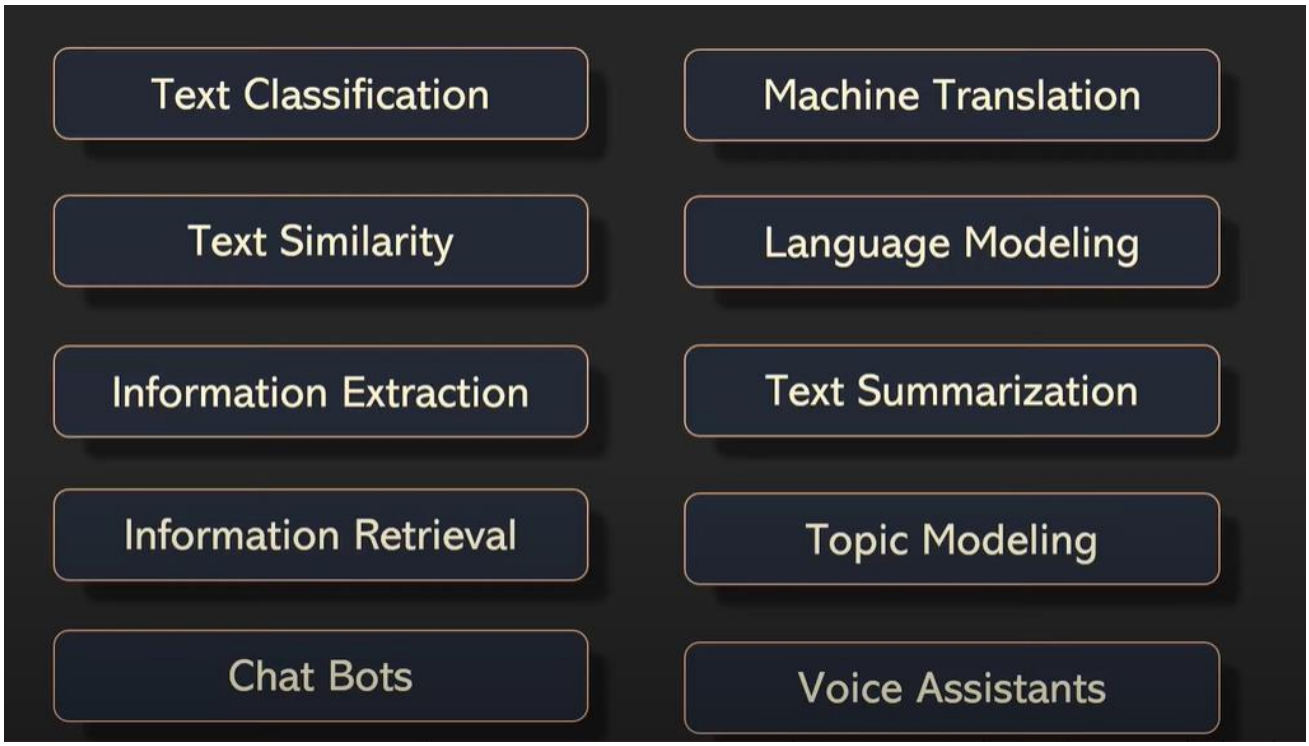
$$\begin{bmatrix} 1 \\ 0.9 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0.87 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0.7 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

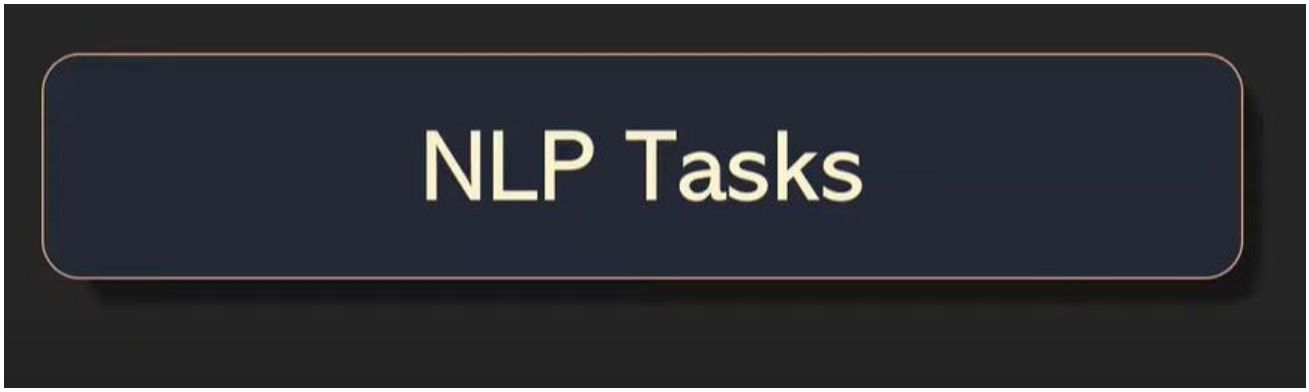
	ashes	Australia	Bat	Cummins	cover	Dhoni	World cup	..	Zimbabwe
Person	0	0.02	0.1	**0.95**	0.03	**0.96**	0.67	...	0.04
Country	0	**0.97**	0	0	0	0	0	...	**1**
Healthy & Fit	0	0	0.3	**0.87**	0	**0.9**	0	...	0
Event	**1**	0.1	0	0	0.4	0	**1**	...	0
gear	0	0	**1**	0	0	0	0	...	0

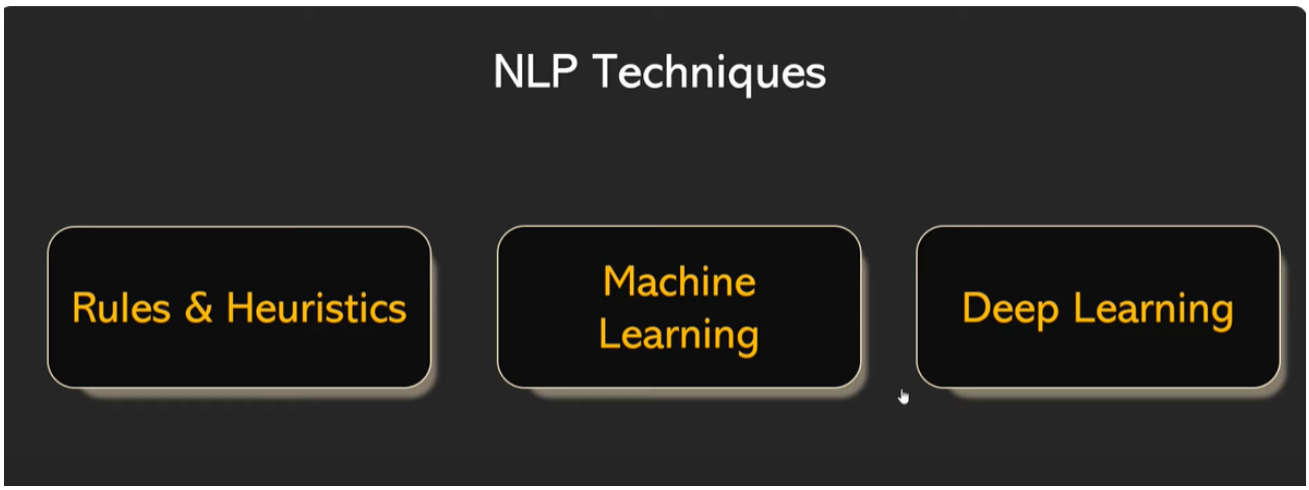



nice
food
padding
After food nice ... poor ... zonal
1 2 3 5000
x
2.3
1.6
6.1
0.2
0.1
0.2
6.1
3.4
Σ σ
ŷ

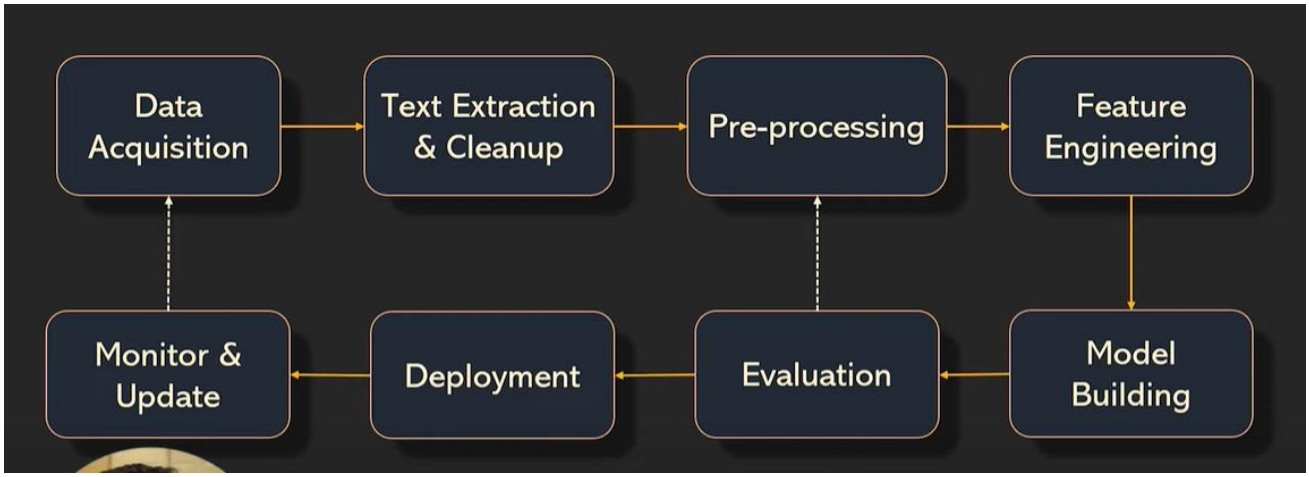
NLP BASICS

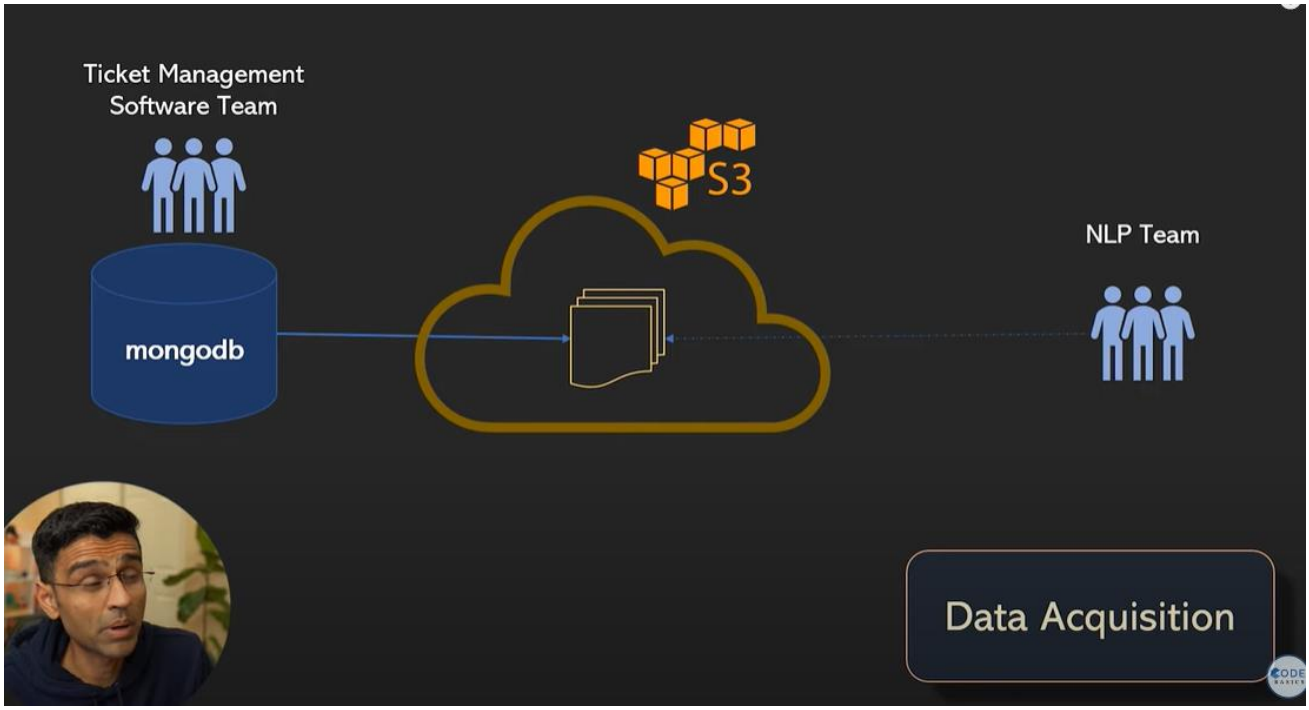


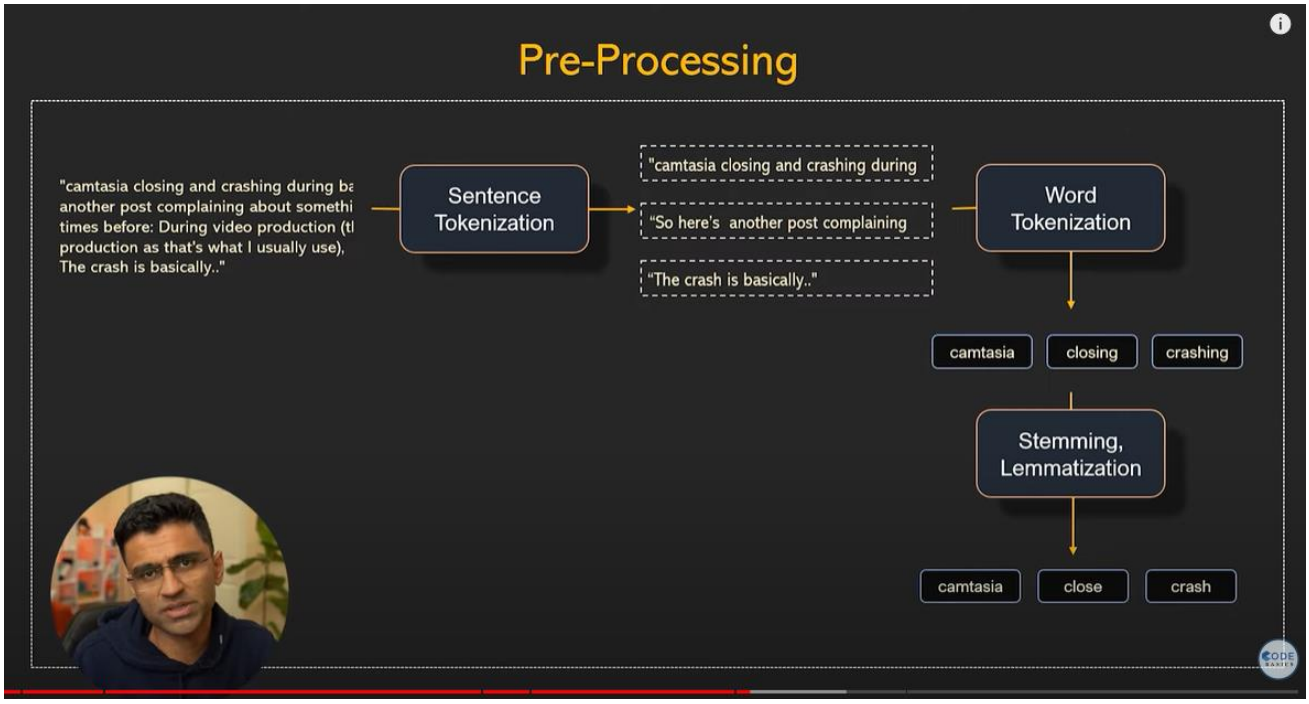


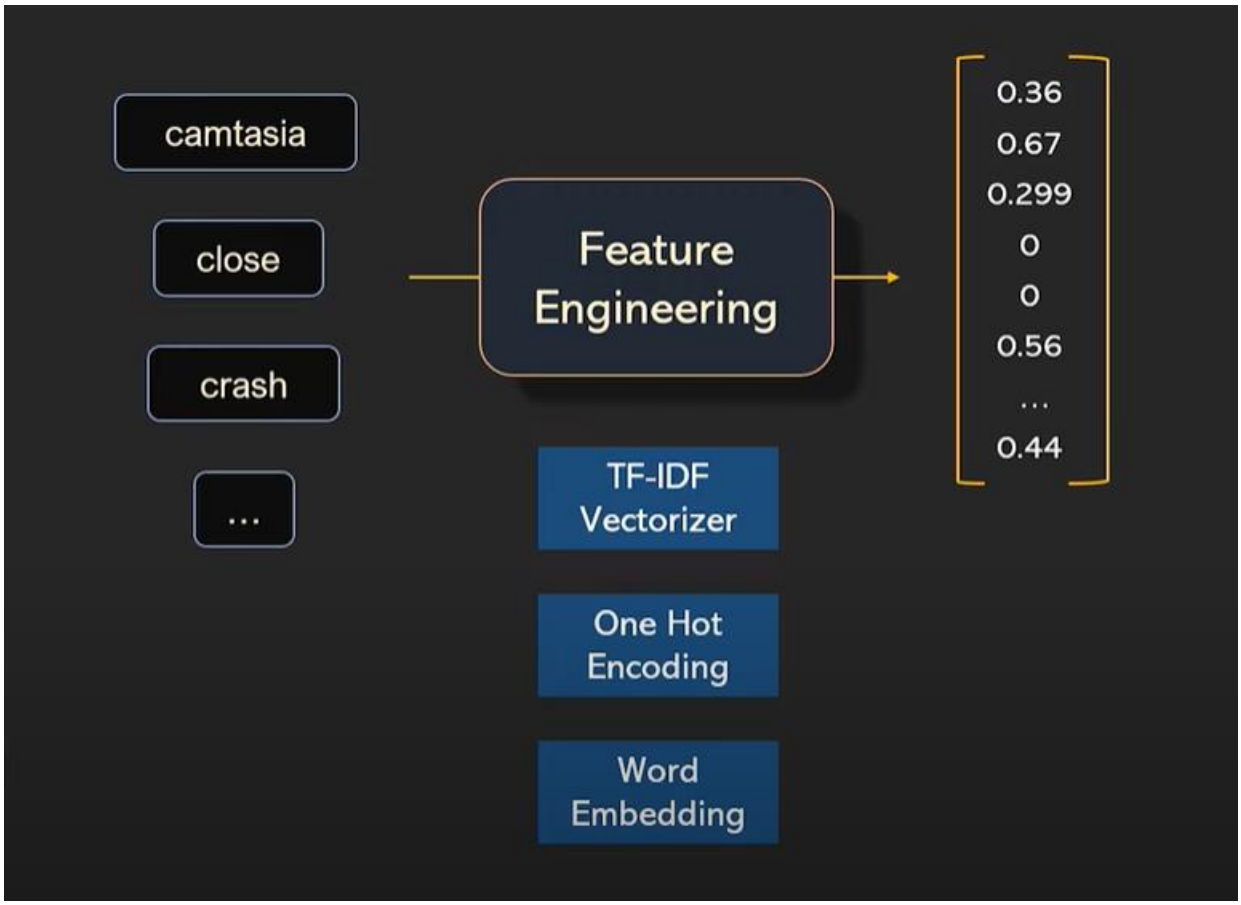


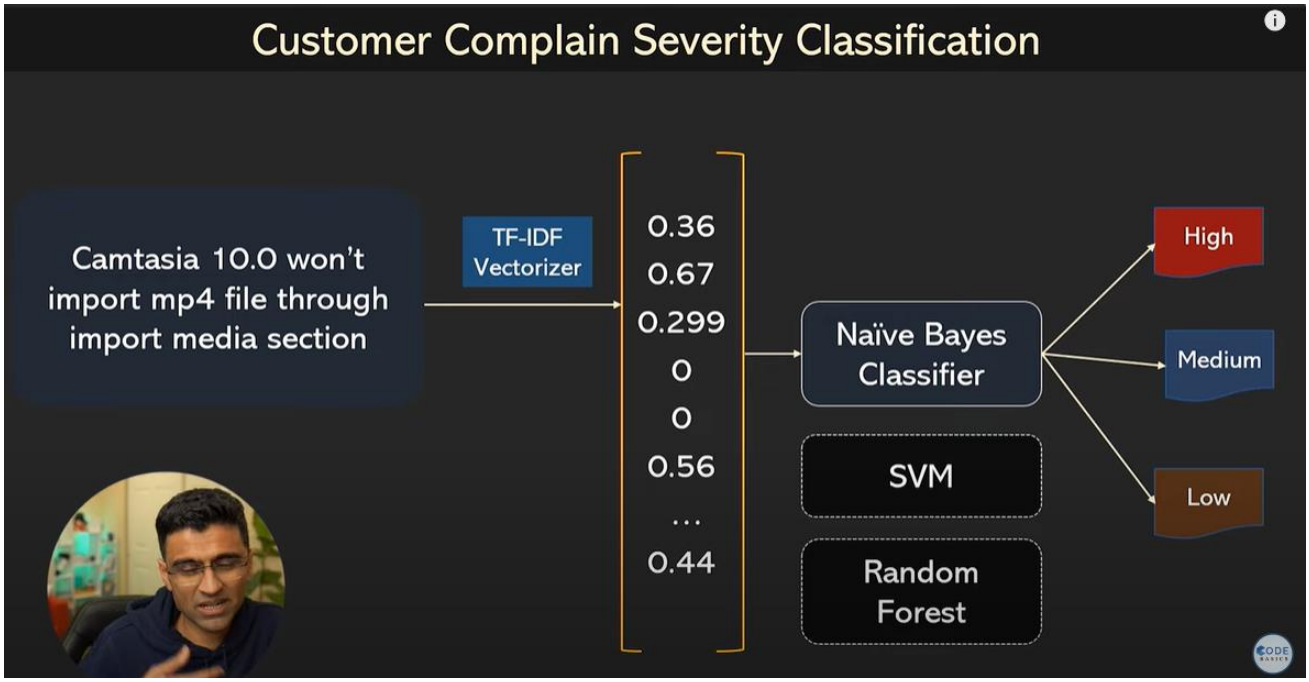
NLP PIPELINE COMPLTE OVERFLOW :

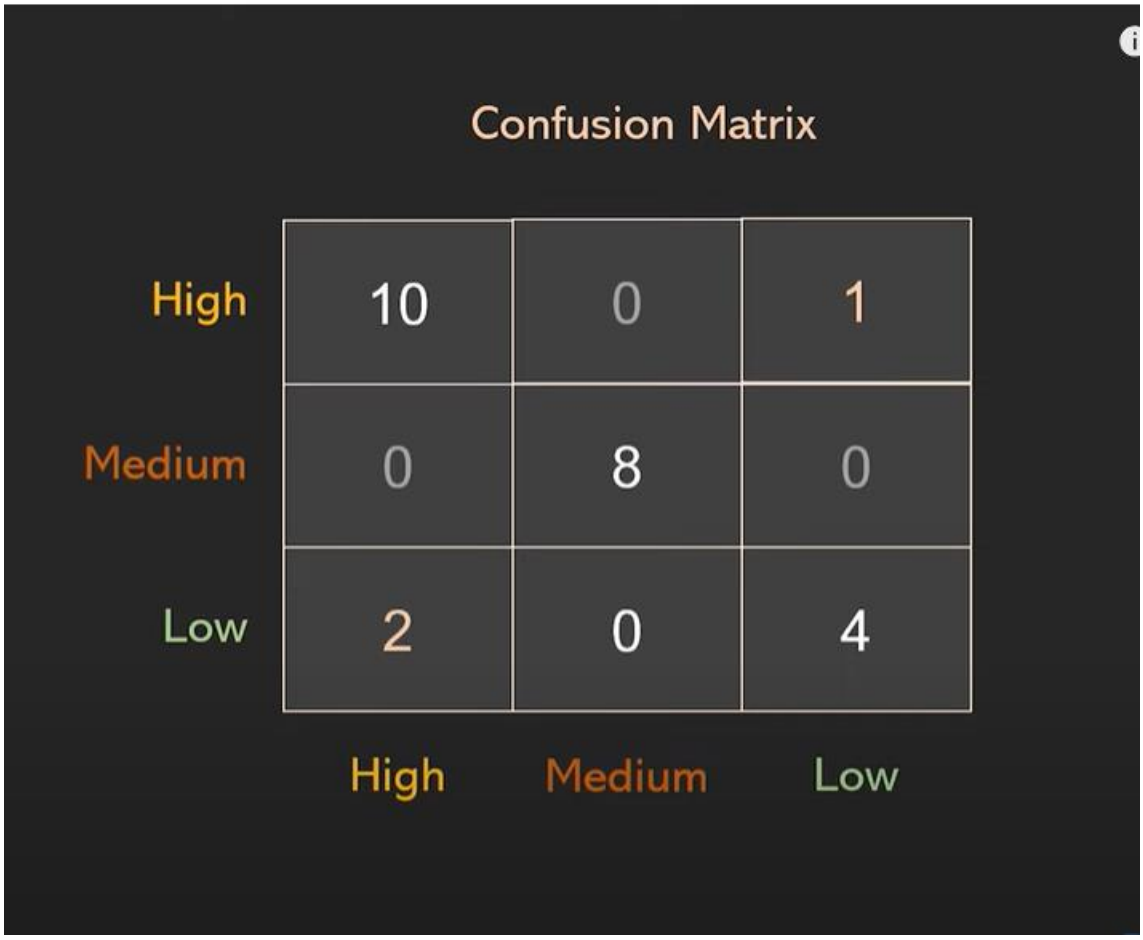

Data Acquisition
Text Extraction & Cleanup
Pre-processing
Feature Engineering
Monitor & Update
Deployment
Evaluation
Model Building


Ticket Management Software Team
S3
NLP Team
mongodb
Data Acquisition


Pre-Processing
"camtasia closing and crashing during ba
another post complaining about somethi
times before: During video production (tl
production as that's what I usually use),
The crash is basically.."
Sentence Tokenization
"camtasia closing and crashing during
"So here's another post complaining
"The crash is basically.."
Word Tokenization
camtasia
closing
crashing
Stemming, Lemmatization
camtasia
close
crash


camtasia
close
crash
...
Feature Engineering
0.36
0.67
0.299
0
0
0.56
...
0.44
TF-IDF Vectorizer
One Hot Encoding
Word Embedding

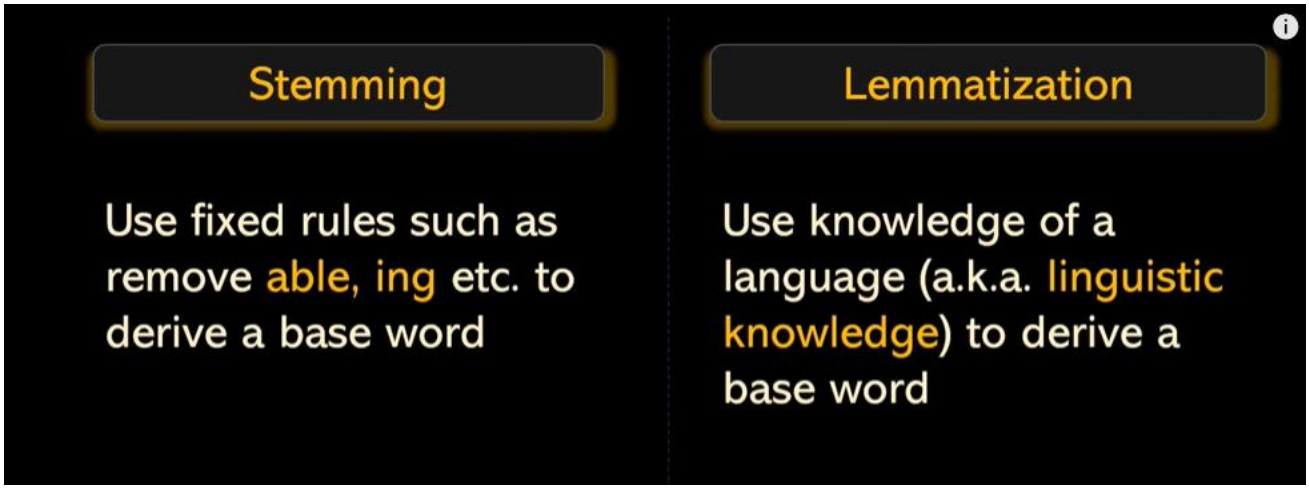

Customer Complain Severity Classification
Camtasia 10.0 won't import mp4 file through import media section
TF-IDF Vectorizer
0.36
0.67
0.299
0
0
0.56
...
0.44
Naïve Bayes Classifier
SVM
Random Forest
High
Medium
Low


Confusion Matrix
High
Medium
Low
10
0
1
0
8
0
2
0
4
High
Medium
Low

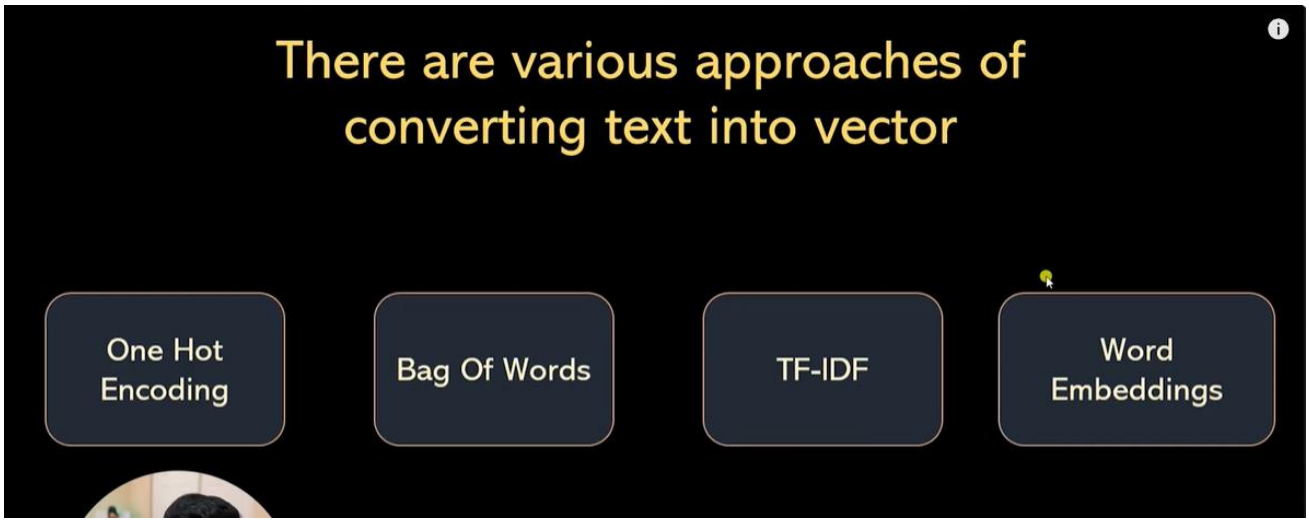
Spacy	NLTK
Spacy is Object Oriented	NLTK is mainly a string processing library
Spacy is user friendly	NLTK is also user friendly but probably less user friendly compared to Spacy
Provides most efficient NLP algorithm for a given task. Hence if you care about the end result, go with Spacy	Provides access to many algorithms. If you care about specific algo and customizations go with NLTK
Spacy is new library and has a very active user community	NLTK is old library. User community as active as Spacy

Spacy	NLTK
Provides most efficient NLP algorithm for a given task. Hence if you care about the end result, go with Spacy	Provides access to many algorithms. If you care about specific algo and customizations go with NLTK
Perfect for app developers	Perfect for researchers

Tokenization is a process of splitting text into meaningful segments



Text representation :



When should I not remove stop words?

Chat bot, Q&A system

Language Translation

Any case where valuable information is lost

to, for, a, over, from, the, had, …

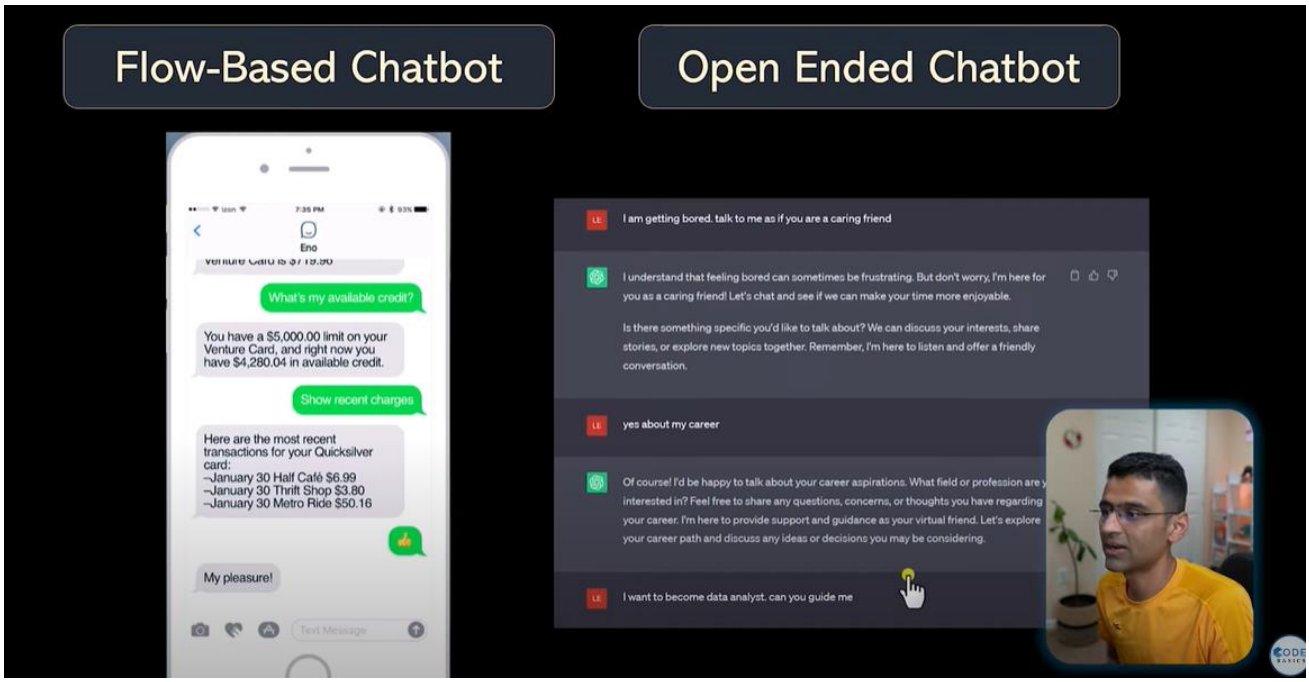
STOP WORDS

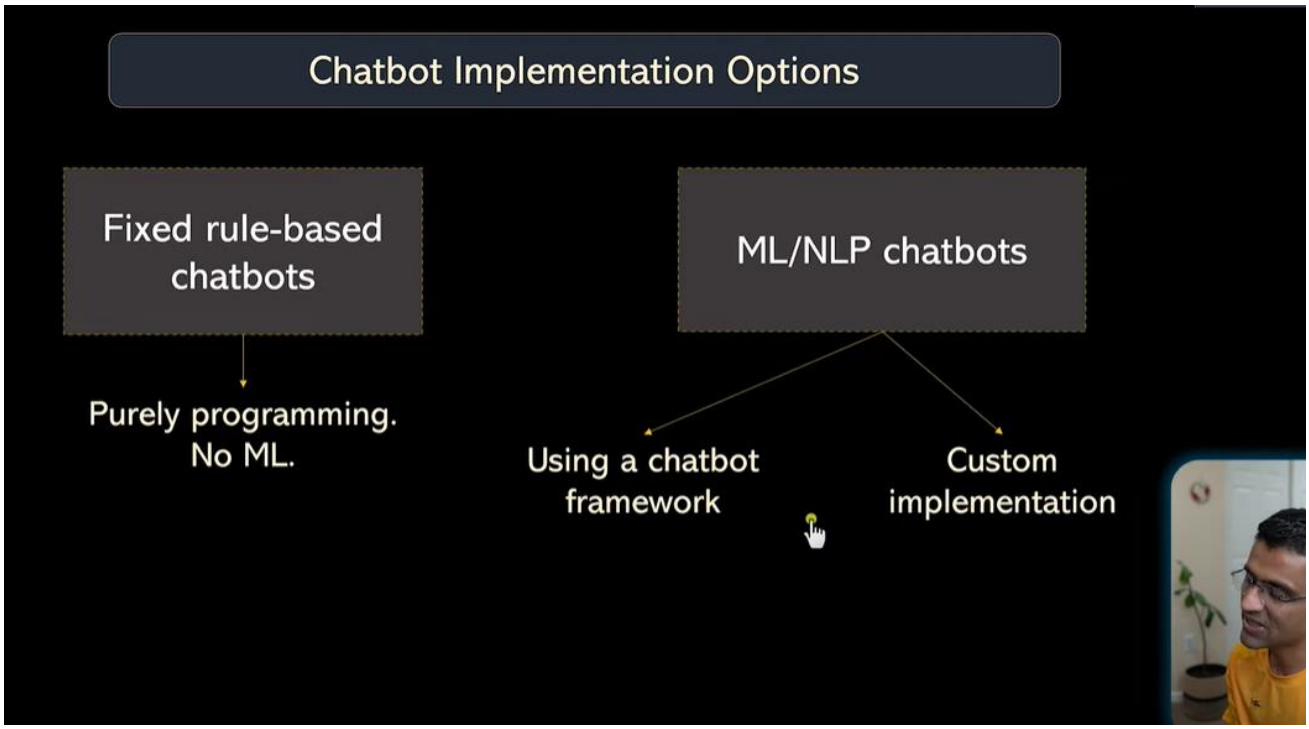
Word embedding techniques

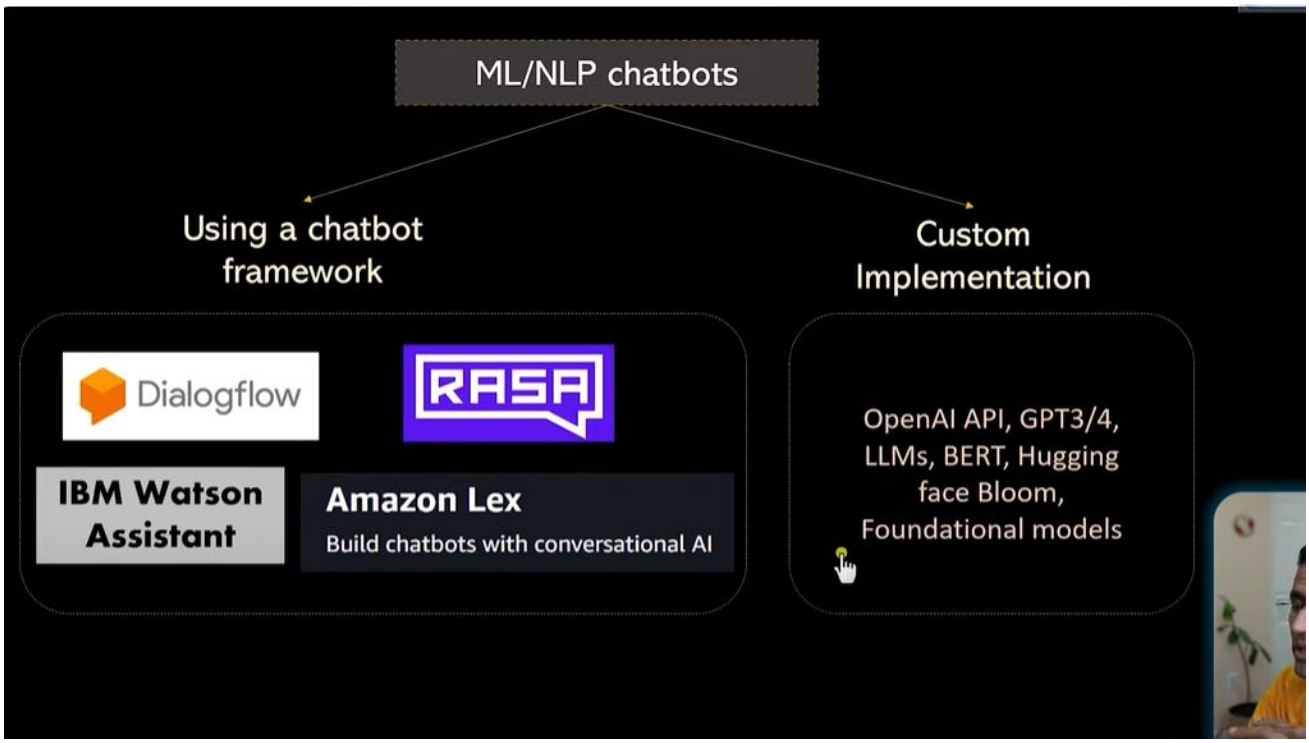
Word2vec

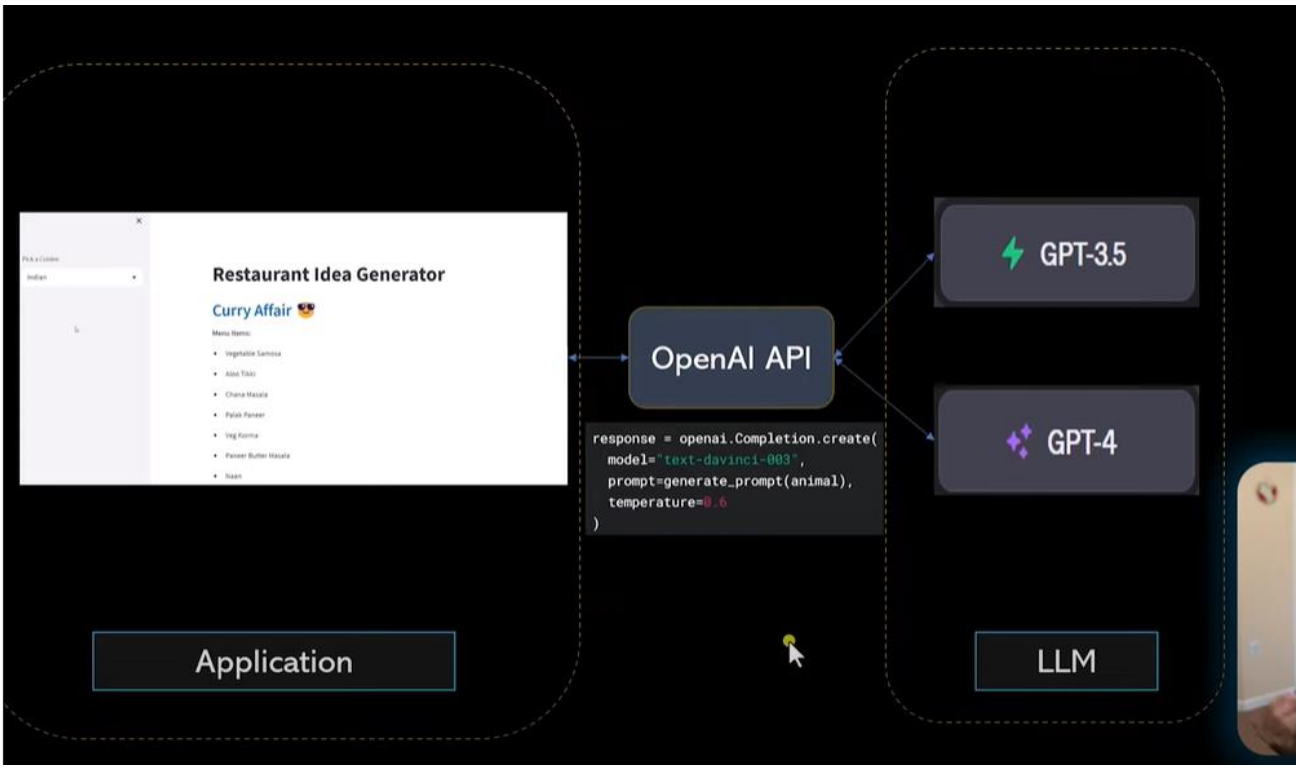
GloVe

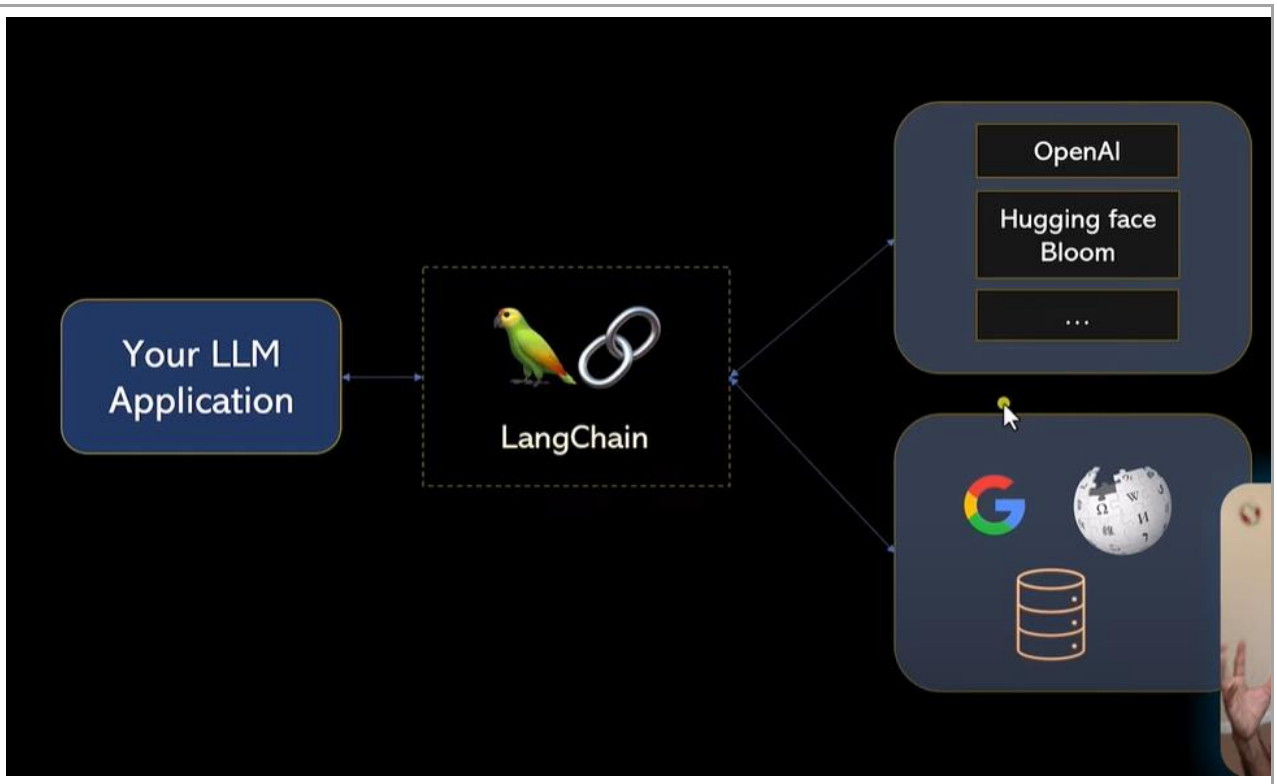
fastText


Flow-Based Chatbot
Open Ended Chatbot
Eno
What's my available credit?
You have a $5,000.00 limit on your Venture Card, and right now you have $4,280.04 in available credit.
Show recent charges
Here are the most recent transactions for your Quicksilver card:
–January 30 Half Café $6.99
–January 30 Thrift Shop $3.80
–January 30 Metro Ride $50.16
My pleasure!
Text Message
I am getting bored. talk to me as if you are a caring friend
I understand that feeling bored can sometimes be frustrating. But don't worry, I'm here for you as a caring friend! Let's chat and see if we can make your time more enjoyable.
Is there something specific you'd like to talk about? We can discuss your interests, share stories, or explore new topics together. Remember, I'm here to listen and offer a friendly conversation.
yes about my career
Of course! I'd be happy to talk about your career aspirations. What field or profession are y interested in? Feel free to share any questions, concerns, or thoughts you have regarding your career. I'm here to provide support and guidance as your virtual friend. Let's explore your career path and discuss any ideas or decisions you may be considering.
I want to become data analyst. can you guide me


Chatbot Implementation Options
Fixed rule-based chatbots
Purely programming. No ML.
ML/NLP chatbots
Using a chatbot framework
Custom implementation


ML/NLP chatbots
Using a chatbot framework
Custom Implementation
Dialogflow
RASA
IBM Watson Assistant
Amazon Lex
Build chatbots with conversational AI
OpenAI API, GPT3/4, LLMs, BERT, Hugging face Bloom, Foundational models


Restaurant Idea Generator
Curry Affair
OpenAI API
response = openai.Completion.create(
model="text-davinci-003",
prompt=generate_prompt(animal),
temperature=0.6
)
GPT-3.5
GPT-4
Application
LLM


OpenAI
Hugging face
Bloom
...
Your LLM
Application
LangChain

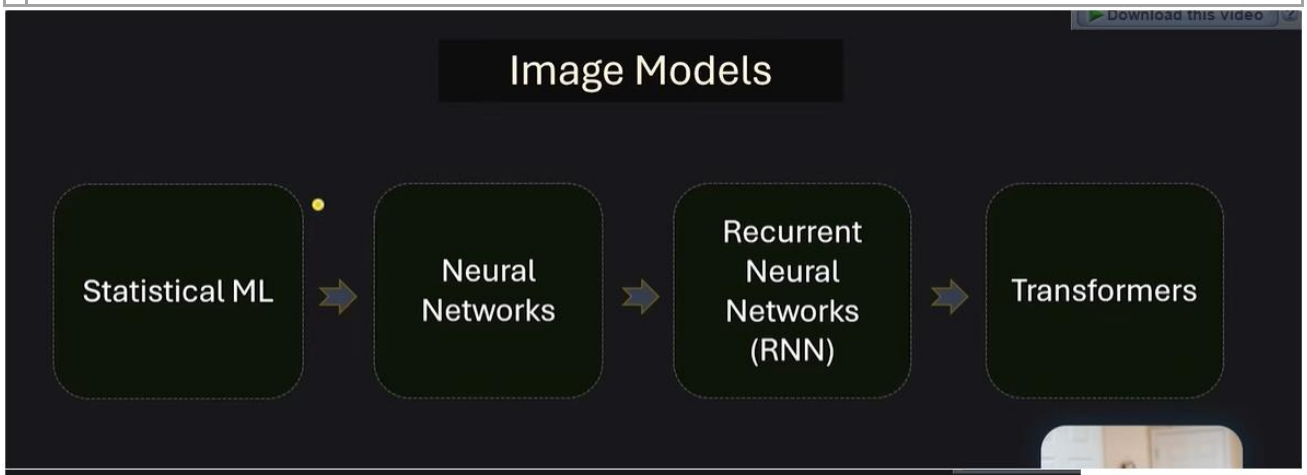
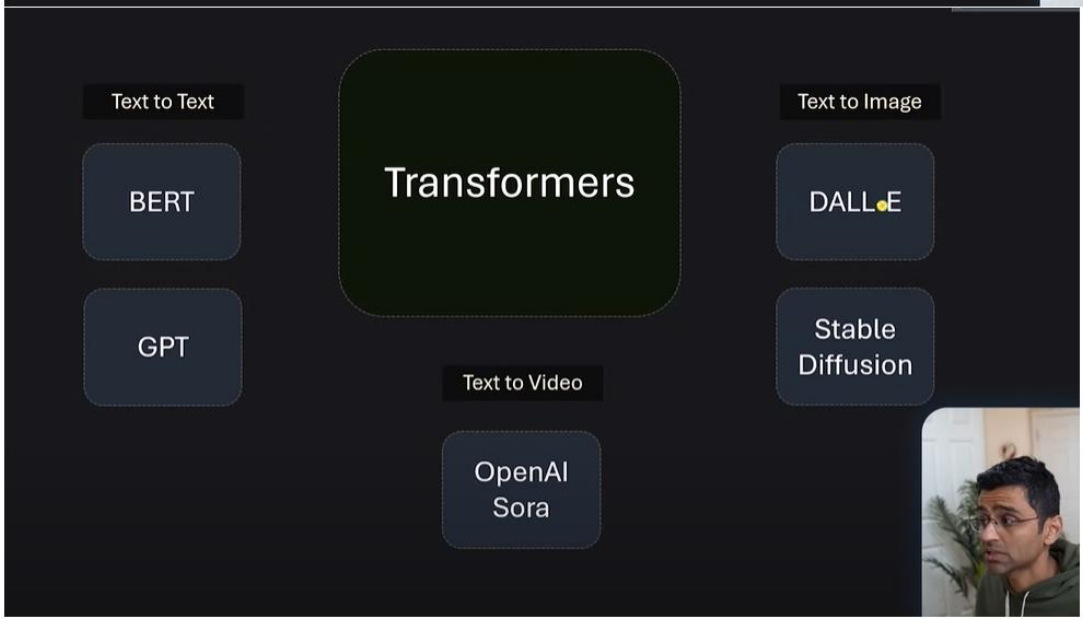


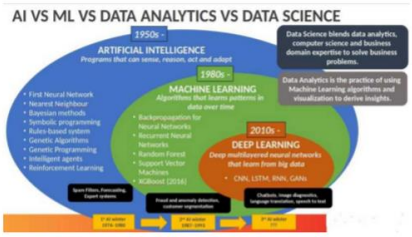
Image Models
Statistical ML
Neural
Networks
Recurrent
Neural
Networks
(RNN)
Transformers


Text to Text
BERT
GPT
Transformers
Text to Image
DALL·E
Stable
Diffusion
Text to Video
OpenAI
Sora

ML,DL,NLP

Q1. What is the difference between AI, Data Science, ML, and DL?

Ans 1 :



Artificial Intelligence: AI is purely math and scientific exercise, but when it became computational, it started to solve human problems formalized into a subset of computer science. Artificial intelligence has changed the original computational statistics paradigm to the modern idea that machines could mimic actual human capabilities, such as decision making and performing more "human" tasks. Modern AI into two categories

1. General AI - Planning, decision making, identifying objects, recognizing sounds, social & business transactions
2. Applied AI - driverless/ Autonomous car or machine smartly trade stocks

Machine Learning: Instead of engineers "teaching" or programming computers to have what they need to carry out tasks, that perhaps computers could teach themselves – learn something without being explicitly programmed to do so. ML is a form of AI where based on more data, and they can change actions and response, which will make more efficient, adaptable and scalable. e.g., navigation apps and recommendation engines. Classified into:-

1. Supervised
2. Unsupervised
3. Reinforcement learning

Data Science: Data science has many tools, techniques, and algorithms called from these fields, plus others –to handle big data

The goal of data science, somewhat similar to machine learning, is to make accurate predictions and to automate and perform transactions in real-time, such as purchasing internet traffic or automatically generating content.

Deep Learning: It is a technique for implementing ML.

ML provides the desired output from a given input, but DL reads the input and applies it to another data. In ML, we can easily classify the flower based upon the features. Suppose you want a machine to look at an image and determine what it represents to the human eye, whether a face, flower, landscape, truck, building, etc.

Machine learning is not sufficient for this task because machine learning can only produce an output from a data set – whether according to a known algorithm or based on the inherent structure of the data. You might be able to use machine learning to determine whether an image was of an "X" – a flower, say – and it would learn and get more accurate. But that output is binary (yes/no) and is dependent on the algorithm, not the data. In the image recognition case, the outcome is not binary and not dependent on the algorithm.

The neural network performs MICRO calculations with computational on many layers. Neural networks also support weighting data for 'confidence. These results in a probabilistic system, vs. deterministic, and can handle tasks that we think of as requiring more 'human-like' judgment.

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised learning

In a supervised learning model, the algorithm learns on a labeled dataset, to generate reasonable predictions for the response to new data. (Forecasting outcome of new data)

- Regression
- Classification

Unsupervised learning

An unsupervised model, in contrast, provides unlabelled data that the algorithm tries to make sense of by extracting features, co-occurrence and underlying patterns on its own. We use unsupervised learning for

- Clustering
- Anomaly detection
- Association
- Autoencoders

Reinforcement Learning

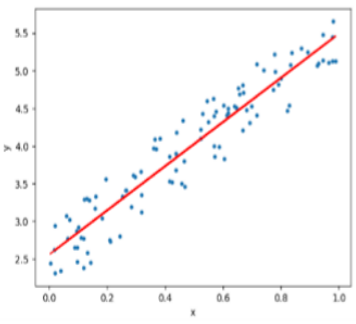
Reinforcement learning is less supervised and depends on the learning agent in determining the output solutions by arriving at different possible ways to achieve the best possible solution.

Q4. What is Linear Regression?

Ans 4:

Linear Regression tends to establish a relationship between a dependent variable(Y) and one or more independent variable(X) by finding the best fit of the straight line.

The equation for the Linear model is Y = mX+c, where m is the slope and c is the intercept



In the above diagram, the blue dots we see are the distribution of 'y' w.r.t 'x.' There is no straight line that runs through all the data points. So, the objective here is to fit the best fit of a straight line that will try to minimize the error between the expected and actual value.

Q6. What is L1 Regularization (L1 = lasso) ?

Ans 6:

The main objective of creating a model(training data) is making sure it fits the data properly and reduce the loss. Sometimes the model that is trained which will fit the data but it may fail and give a poor performance during analyzing of data (test data). This leads to overfitting. Regularization came to overcome overfitting.

Lasso Regression (**Least Absolute Shrinkage and Selection Operator**) adds "Absolute value of magnitude" of coefficient, as penalty term to the loss function.

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2}_{\text{Loss function}} + \underbrace{\lambda \sum_{j=0}^{M} W_j^2}_{\text{Regularization Term}}$$

Methods like Cross-validation, Stepwise Regression are there to handle overfitting and perform feature selection work well with a small set of features. These techniques are good when we are dealing with a large set of features.

Along with shrinking coefficients, the **lasso performs feature selection**, as well. (Remember the 'selection' in the lasso full-form?) Because some of the coefficients become exactly zero, which is equivalent to the particular feature being excluded from the model.

Q7. L2 Regularization(L2 = Ridge Regression)

Ans 7:

$$Cost\ function = Loss + \frac{\lambda}{2m} * \sum \|w\|^2$$

Overfitting happens when the model learns signal as well as noise in the training data and wouldn't perform well on new/unseen data on which model wasn't trained on.

To avoid overfitting your model on training data like **cross-validation sampling, reducing the number of features, pruning, regularization**, etc.

So to avoid overfitting, we perform Regularization.

Q8. What is R square(where to use and where not)?

Ans 8.

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

The definition of R-squared is the percentage of the response variable variation that is explained by a linear model.

R-squared = Explained variation / Total variation

R-squared is always between 0 and 100%.

0% indicates that the model explains none of the variability of the response data around its mean.

100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

Sum Squared Regression Error

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

Sum Squared Total Error

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Q2. Difference between logistic and linear regression?

Answer:

Linear and Logistic regression are the most basic form of regression which are commonly used The essential difference between these two is that Logistic regression is used when the dependent variable is binary In contrast Linear regression is used when the dependent variable is continuous and the nature of the regression line is linear

Key Differences between Linear and Logistic Regression

Linear regression models data using continuous numeric value As against logistic regression models the data in the binary values

Linear regression requires to establish the linear relationship among dependent and independent variables whereas it is not necessary for logistic regression

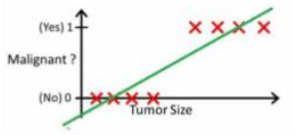
In linear regression the independent variable can be correlated with each other On the contrary in the logistic regression the variable must not be correlated with each other

Q3. Why we can't do a classification problem using Regression?

Answer:-

With linear regression you fit a polynomial through the data - say like on the example below we fit a straight line through ¤tumor size tumor type¤ sample set

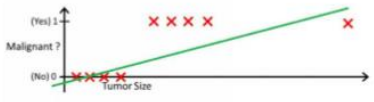
---



Above malignant tumors get and non-malignant ones get 0 and the green line is our hypothesis h x To make predictions we may say that for any given tumor size x if h x gets bigger than 0 5 we predict malignant tumors Otherwise we predict benignly

It looks like this way we could correctly predict every single training set sample but now let's change the task a bit

Intuitively it's clear that all tumors larger certain threshold are malignant So let's add another sample with huge tumor size and run linear regression again



Now our h x >0 5⟶malignant doesn't work anymore To keep making correct predictions we need to change it to h x >0 2 or something - but that not how the algorithm should work

We cannot change the hypothesis each time a new sample arrives Instead we should learn it off the training set data and then using the hypothesis we've learned make correct predictions for the data we haven't seen before

Linear regression is unbounded.

Q9. What is Variance and Bias tradeoff?

Answer:

In predicting models the prediction error is composed of two different errors

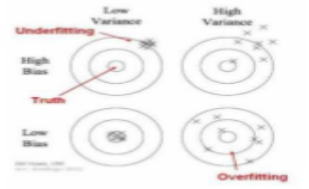
Bias

2 Variance

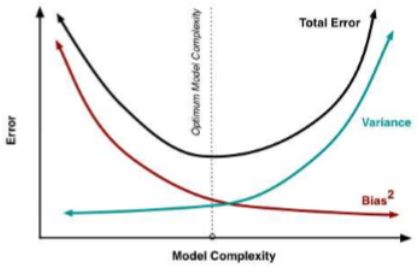
It is important to understand the variance and bias trade-off which tells about to minimize the Bias and Variance in the prediction and avoids overfitting & under fitting of the model

Bias It is the difference between the expected or average prediction of the model and the correct value which we are trying to predict Imagine if we are trying to build more than one model by collecting different data sets and later on evaluating the prediction we may end up by different prediction for all the models So bias is something which measures how far these model prediction from the correct prediction It always leads to a high error in training and test data

Variance Variability of a model prediction for a given data point We can build the model multiple times so the variance is how much the predictions for a given point vary between different realizations of the model



For example Voting Republican - 3 Voting Democratic - 6 Non-Respondent - 2 Total - 50
The probability of voting Republican is 3/ 3□ 6 or 44 8% We put out our press release that the Democrats are going to win by over 0 points; but when the election comes around it turns out they lose by 0 points That certainly reflects poorly on us Where did we go wrong in our model?
Bias scenario□s using a phonebook to select participants in our survey is one of our sources of bias By only surveying certain classes of people it skews the results in a way that will be consistent if we repeated the entire model building exercise Similarly not following up with respondents is another source of bias as it consistently changes the mixture of responses we get On our bulls-eye diagram these move us away from the center of the target but they would not result in an increased scatter of estimates
Variance scenarios the small sample size is a source of variance If we increased our sample size the results would be more consistent each time we repeated the survey and prediction The results still might be highly inaccurate due to our large sources of bias but the variance of predictions will be reduced



Bagging (Bootstrap Aggregation) is used when our goal is to reduce the variance of a decision tree Here the idea is to create several subsets of data from the training sample chosen randomly with replacement Now each collection of subset data is used to train their decision trees As a result we end up with an ensemble of different models Average of all the predictions from different trees are used which is more robust than a single decision tree

Boosting is another ensemble technique to create a collection of predictors In this technique learners are learned sequentially with early learners fitting simple models to the data and then analyzing data

Q10. What are Ensemble Methods?

Answer

Bagging and **Boosting**
Decision trees have been around for a long time and also known to suffer from bias and variance You will have a large bias with simple trees and a large variance with complex trees

Ensemble methods - which combines several decision trees to produce better predictive performance than utilizing a single decision tree The main principle behind the ensemble model is that a group of weak learners come together to form a strong learner

Two techniques to perform ensemble decision trees
Bagging
2 Boosting

for errors In other words we fit consecutive trees random sample and at every step the goal is to solve for net error from the prior tree
When a hypothesis misclassifies an input its weight is increased so that the next hypothesis is more likely to classify it correctly By combining the whole set at the end converts weak learners into a better performing model

The different types of boosting algorithms are
1□ **AdaBoost**
2□ □ **radient Boosting**
3 **X□ Boost**

Q12. What is the Confusion Matrix?

Answer:

A confusion matrix is a table that is often used to describe the performance of a classification model or classifier on a set of test data for which the true values are known It allows the visualization of the performance of an algorithm

---

A confusion matrix is a summary of prediction results on a classification problem The number of correct and incorrect predictions are summarized with count values and broken down by each class

This is the key to the confusion matrix
It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made

	Class 1 Predicted	*Class 2 Predicted*
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Here

- Class Positive
- Class 2 Negative

Definition of the Terms

1 **Positive (P)** Observation is positive for example is an apple
2 **Negative (N)** Observation is not positive for example is not an apple
3 **True Positive (TP)** Observation is positive and is predicted to be positive
4 **False Negative (FN)** Observation is positive but is predicted negative
5 **True Negative (TN)** Observation is negative and is predicted to be negative
6 **False Positive (FP)** Observation is negative but is predicted positive

Q13. What is Accuracy and Misclassification Rate?

Answer

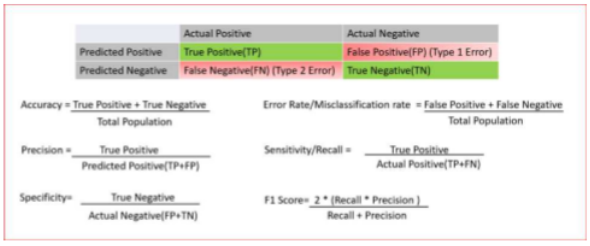
Accuracy
Accuracy is defined as the ratio of the sum of True Positive and True Negative by Total TP□TN□FP□FN

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

However□there are problems with accuracy□It assumes e□ual costs for both kinds of errors□A □□□ accuracy can be excellent□good□mediocre□poor□or terrible depending upon the problem□

Misclassification □ ate

Misclassification Rate is defined as the ratio of the sum of False Positive and False Negative by Total TP□TN□FP□FN
Misclassification Rate is also called Error Rate



Q16. What are F1 Score, precision and recall?

Recall -
Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples

1. High Recall indicates the class is correctly recognized small number of FN
2. Low Recall indicates the class is incorrectly recognized large number of FN

Recall is given by the relation

$$Recall = \frac{TP}{TP + FN}$$

Precision:
To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples

1. High Precision indicates an example labeled as positive is indeed positive a small number of FP
2. Low Precision indicates an example labeled as positive is indeed positive large number of FP

The relation gives precision

$$Precision = \frac{TP}{TP + FP}$$

emember -
High recall low precision This means that most of the positive examples are correctly recognized low FN but there are a lot of false positives
Low recall high precision This shows that we miss a lot of positive examples high FN but those we predict as positive are indeed positive low FP

Q14. True Positive Rate & True Negative Rate

Answer:

True Positive Rate:
Sensitivity (SN) is calculated as the number of correct positive predictions divided by the total number of positives. It is also called **Recall (REC)** or true positive rate (TPR). The best sensitivity is 1.0, whereas the worst is 0.0.

$$SN = \frac{TP}{TPFN} = \frac{TP}{P}$$

True Negative Rate

Specificity (SP) is calculated as the number of correct negative predictions divided by the total number of negatives. It is also called a true negative rate (TNR). The best specificity is 1.0, whereas the worst is 0.0.

$$SN = \frac{TP}{TPFN} = \frac{TP}{P}$$

Since we have two measures Precision and Recall it helps to have a measurement that represents both of them We calculate an **F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more**

The F-Measure will always be nearer to the smaller value of Precision or ecall

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall + Precision}$$

Q6. What is Pipeline in sklearn ?

A pipeline is what chains several steps together once the initial exploration is done For example some codes are meant to transform features normalize numerically or turn text into vectors or fill up missing data and they are transformers; other codes are meant to predict variables by fitting an algorithm

such as random forest or support vector machine they are estimators Pipeline chains all these together which can then be applied to training data in block
Example of a pipeline that imputes data with the most frequent value of each column and then fit a decision tree classifier

Q7. What is Principal Component Analysis(PCA), and why we do?

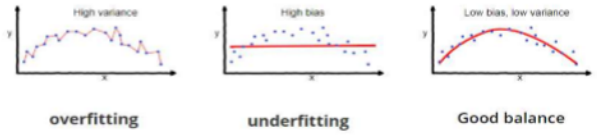
The main idea of principal component analysis PCA is to reduce the dimensionality of a data set consisting of many variables correlated with each other either heavily or lightly while retaining the variation present in the dataset up to the maximum extent The same is done by transforming the variables to a new set of variables which are known as the principal components or simply the PCs and are orthogonal ordered such that the retention of variation present in the original variables decreases as we move down in the order So in this way the st principal component retains maximum variation that was present in the original components The principal components are the eigenvectors of a covariance matrix and hence they are orthogonal

Main important points to be considered

1. Normalize the data
2. Calculate the covariance matrix
3. Calculate the eigenvalues and eigenvectors
4. Choosing components and forming a feature vector
5. Forming Principal Components

Bias: Bias means that the model favors one result more than the others Bias is the simplifying assumptions made by a model to make the target function easier to learn The model with high bias pays very little attention to the training data and oversimplifies the model It always leads to a high error in training and test data

Variance: Variance is the amount that the estimate of the target function will change if different training data was used The model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn t seen before As a result such models perform very well on training data but have high error rates on test data



So the end goal is to come up with a model that balances both Bias and Variance This is called *Bias Variance Trade-off* To build a good model we need to find a good balance between bias and variance such that it minimizes the total error

Q5: What is the difference between machine learning and deep learning?

Machine Learning | deep learning

Machine Learning is a technique to learn from that data and then apply wha t has been learnt to make an informed decision | The main difference betwe en deep and machine learning is, machine learning models become better progressively but the model still needs some guidance. If a machine-learning model returns an inaccurate prediction then the programmer need s to fix that problem explicitly but in the case of deep learning, the model do es it by himself.

>Machine Learning can perform well with small size data also | Deep Learn ing does not perform as good with smaller datasets.

>Machine learning can work on some low-end machines also | Deep Learning involves many matrix multiplication op erations which are better suited for GPUs

>Features need to be identified and extracted as per the domain before pu shing them to the algorithm | Deep learning algorithms try to learn high-level features from data.

>It is generally recommended to break the problem into smaller chunks, sol ve them and then combine the results | It generally focusses on solving the problem end to end

>Training time is comparatively less | Training time is comparatively more

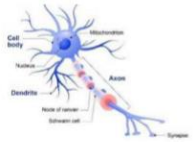
>Results are more interpretable | Results Maybe more accurate but less int erpretable

> No use of Neural networks | uses neural networks

> Solves comparatively less complex problems | Solves more complex prob lems.

Q6: What is perceptron and how it is related to human neurons?

If we focus on the structure of a biological neuron, it has dendrites, which are used to receive inputs. These inputs are summed in the cell body and using the Axon it is passed on to the next biological neuron as shown below.

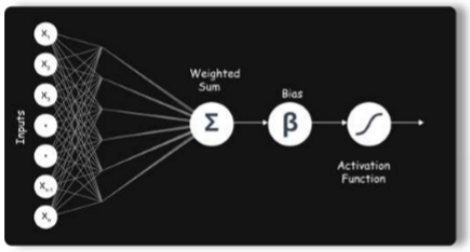


Dendrite: Receives signals from other neurons

Cell Body: Sums all the inputs

Axon: It is used to transmit signals to the other cells

Similarly, a perceptron receives multiple inputs, applies various transformations and functions and provides an output. A Perceptron is a linear model used for binary classification. It models a neuron, which has a set of inputs, each of which is given a specific weight. The neuron computes some function on these weighted inputs and gives the output.



Q7: Why deep learning is better than machine learning?

Though traditional ML algorithms solve a lot of our cases, they are not useful while working with high dimensional data that is where we have a large number of inputs and outputs. For example, in the case of handwriting recognition, we have a large amount of input where we will have different types of inputs associated with different types of handwriting.

Q8: What kind of problem can be solved by using deep learning?

Deep Learning is a branch of Machine Learning, which is used to solve problems in a way that mimics the human way of solving problems. Examples:

- Image recognition
- Object Detection
- Natural Language processing- Translation, Sentence formations, text to speech, speech to text
- understand the semantics of actions

Q9: List down all the activation function using mathematical Expression and example. What is the activation function?

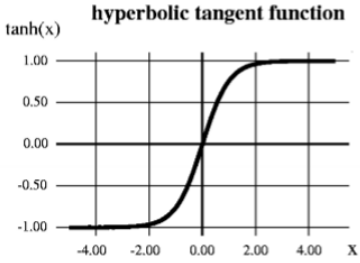
Activation functions are very important for an Artificial Neural Network to learn and make sense of something complicated and the Non-linear complex functional mappings between the inputs and response variable. They introduce non-linear properties to our Network. Their main purposes are to convert an input signal of a node in an A-NN to an output signal.

So why do we need Non-Linearities?

iNeuron

Non-linear functions are those, which have a degree more than one, and they have a curvature when we plot a Non-Linear function. Now we need a Neural Network Model to learn and represent almost anything and any arbitrary complex function, which maps inputs to outputs. Neural-Networks are considered Universal Function Approximations. It means that they can compute and learn any function at all.

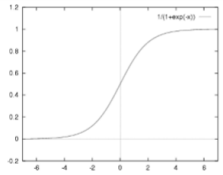
Most popular types of Activation functions -



Most popular types of Activation functions -

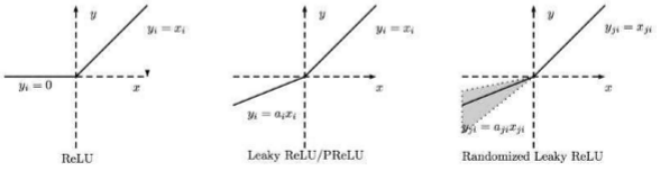
- Sigmoid or Logistic
- Tanh — Hyperbolic tangent
- ReLu -Rectified linear units

Sigmoid Activation function: It is a activation function of form f(x) = 1 / 1 + exp(-x) . Its Range is between 0 and 1. It is an S-shaped curve. It is easy to understand.



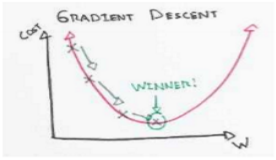
Hyperbolic Tangent function- Tanh : It's mathematical formula is f(x) = 1 — exp(-2x) / 1 + exp(-2x). Now it's the output is zero centred because its range in between -1 to 1 i.e. -1 < output < 1 . Hence optimisation is easier in this method; Hence in practice, it is always preferred over Sigmoid function.

ReLu- Rectified Linear units: It has become more popular in the past couple of years. It was recently proved that it has six times improvement in convergence from Tanh function. It's R(x) = max (0,x) i.e. if x < 0 , R(x) = 0 and if x >= 0 , R(x) = x. Hence as seen that mathematical form of this function, we can see that it is very simple and efficient. Many times in Machine learning and computer science we notice that most simple and consistent techniques and methods are only preferred and are the best. Hence, it avoids and rectifies the vanishing gradient problem. Almost all the deep learning Models use ReLu nowadays.



Q10: Detail explanation about gradient decent using example and Mathematical expression?

Gradient descent is an optimisation algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by negative of the gradient. In machine learning, we used gradient descent to update the parameters of our model. Parameters refer to coefficients in the Linear Regression and weights in neural networks.



The size of these steps called the learning rate. With the high learning rate, we can cover more ground each step, but we risk overshooting the lower point since the slope of the hill is constantly changing. With a very lower learning rate, we can confidently move in the direction of the negative gradient because we are recalculating it so frequently. The Lower learning rate is more precise, but calculating the gradient is time-consuming, so it will take a very large time to get to the bottom.

Math

Now let's run gradient descent using new cost function. There are two parameters in cost function we can control: m (weight) and b (bias). Since we need to consider that the impact each one has on the final prediction, we need to use partial derivatives. We calculate the partial derivative of the cost function concerning each parameter and store the results in a gradient.

Math

Given the cost function:

$$f(m,b) = \frac{1}{N}\sum_{i=1}^{n}(y_i - (mx_i + b))^2$$

The gradient can be calculated as:

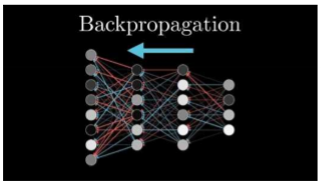
$$f'(m,b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum -2x_i(y_i - (mx_i + b)) \\ \frac{1}{N}\sum -2(y_i - (mx_i + b)) \end{bmatrix}$$

To solve for the gradient, we iterate by our data points using our new m and b values and compute the partial derivatives. This new gradient tells us about the slope of the cost function at our current position (current parameter values) and the directions we should move to update our parameters. The learning rate controls the size of our update.

Q11: What is backward propagation?

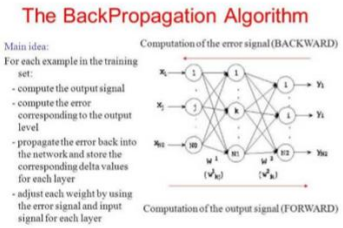
Back-propagation is the essence of the neural net training and this method of fine-tuning the weights of a neural net based on the errors rate obtained in the previous epoch. Proper tuning of the weights allows us to reduce error rates and to make the model reliable by increasing its generalisation.

Backpropagation is a short form of "backward propagation of errors." This is the standard method of training artificial neural networks. This helps to calculate the gradient of a loss function with respects to all the weights in the network.



Most prominent advantages of Backpropagation are:

- Backpropagation is the fast, simple and easy to program.
- It has no parameters to tune apart from the numbers of input.
- It is the flexible method as it does not require prior knowledge about the network
- It is the standard method that generally works well.
- It does not need any special mentions of the features of the function to be learned.



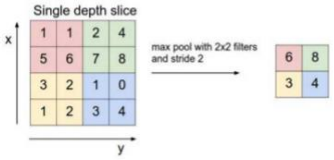
Q16: What is CNN?

This is the simple application of a filter to an input that results in inactivation. Repeated application of the same filter to input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in input, such as an image.

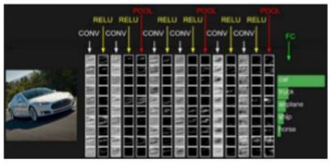
Convolutional layers are the major building blocks which are used in convolutional neural networks.

A covnets is the sequence of layers, and every layer transforms one volume to another through differentiable functions.
Different types of layers in CNN:
Let's take an example by running a covnets on of image of dimensions **32 x 32 x 3.**

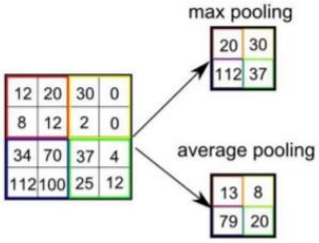
1. Input Layer: It holds the raw input of image with width 32, height 32 and depth 3.
2. Convolution Layer: It computes the output volume by computing dot products between all filters and image patches. Suppose we use a total of 12 filters for this layer we'll get output volume of dimension 32 x 32 x 12.
3. Activation Function Layer: This layer will apply the element-wise activation function to the output of the convolution layer. Some activation functions are RELU: max(0, x), Sigmoid: 1/(1+e^-x), Tanh, Leaky RELU, etc. So the volume remains unchanged. Hence output volume will have dimensions 32 x 32 x 12.
4. Pool Layer: This layer is periodically inserted within the covnets, and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.



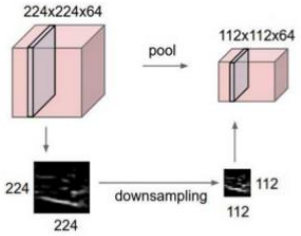
5. Fully-Connected Layer: This layer is a regular neural network layer that takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.



control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation.



The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 downsamples every depth slice in the input by two along both width and height, discarding 75% of the activations. Every MAX operation would, in this case, be taking a max over four numbers (little 2x2 region in some depth slice). The depth dimension remains unchanged.



Q1: What are Epochs?

One Epoch is an ENTIRE dataset is passed forwards and backwards through the neural network

Since one epoch is too large to feed to the computer at once we divide it into several smaller batches

We always use more than one Epoch because □□e *epoc*□ □*ed*□ *to* □□*derfitt*□□□

As the number of epochs increases several times the weight are changed in the neural network and the curve goes from underfitting up to optimal to overfitting curve

Q2. What is the batch size?

Batc□ Si□e

The total number of training and examples present in a single batch

Unlike the learning rate hyperparameter where its value doesn t affect computational time the batch sizes must be examined in conjunctions with the execution time of training The batch size is limited by hardware s memory while the learning rate is not Leslie recommends using a batch size that fits in hardware s memory and enables using larger learning rate

If our server has multiple GPUs the total batch size is the batch size on a GPU multiplied by the numbers of GPU If the architectures are small or your hardware permits very large batch sizes then you might compare the performance of different batch sizes Also recall that small batch sizes add regularization while large batch sizes add less so utilize this while balancing the proper amount of regularization It is often better to use large batch sizes so a larger learning rate can be used

Q3: What is dropout in Neural network?

Dropout refers to ignoring units during the training phase of a certain set of neurons which is chosen randomly These units are not considered during the particular forward or backward pass

More technically at each training stage individual nodes are either dropped out of the net with probability □*p* or kept with probability *p* so that a reduced network is left; incoming and outgoing edges to a dropped-out node are also removed

Q20: What is learning Rate?

Learning Rate

The learning rate controls how much we should adjust the weights concerning the loss gradient. Learning rates are randomly initialised.

Lower the values of the learning rate slower will be the convergence to global minima.

Higher values for the learning rate will not allow the gradient descent to converge

iNeuron

Since our goal is to minimise the function cost to find the optimised value for weights, we run multiples iteration with different weights and calculate the cost to arrive at a minimum cost

Q4: List down hyperparameter tuning in deep learning.

The process of setting the hyper-parameters requires expertise and extensive trial and error There are no simple and easy ways to set hyper-parameters specifically learning rate batch size momentum and weight decay

Approaches to searching for the best configuration:

- Grid Search
- Random Search

Q8: What is Transfer learning in deep learning ?

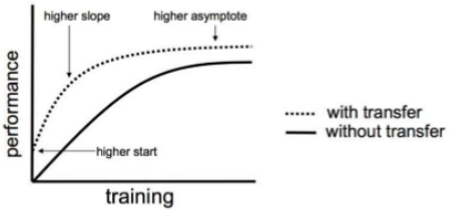
Transfer learning: It is a machine learning method where a model is developed for the task is again used as the starting point for a model on a second task

It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems

Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task

Transfer learning is an optimization that allows rapid progress or improved performance when modelling the second task

Transfer learning only works in deep learning if the model features learned from the first task are general



iNeuron

□ pproac□

1 Observe and understand the clues available during training by monitoring validation□test loss early in training tune your architecture and hyper-parameters with short runs of a few epochs

□□ Signs of □□*derfitt*□□ or □□*erfitt*□□ of the test or validation loss early in the training process are useful for tuning the hyper-parameters□

Q9. What is data augmentation? Can you give some examples?

Answer:

Data augmentation is a technique for synthesizing new data by modifying existing data in such a way that the target is not changed or it is changed in a known way

Computer vision is one of fields where data augmentation is very useful There are many modifications that we can do to images:

- Resize
- Horizontal or vertical flip
- Rotate
- Add noise
- Deform
- Modify colors

Each problem needs a customized data augmentation pipeline For example on OCR doing flips will change the text and won t be beneficial; however resizes and small rotations may help

iNeuron

Q4. Transfer learning is one of the most useful concepts today. Where can it be used?

Answer:

Pre-trained models are probably one of the most common use cases for transfer learning

For anyone who does not have access to huge computational power, training complex models is always a challenge Transfer learning aims to help by both improving the performance and speeding up your network

In layman terms, transfer learning is a technique in which a model that has already been trained to do one task is used for another without much change This type of learning is also called multi-task learning

Many models that are pre-trained are available online Any of these models can be used as a starting point in the creation of the new model required After just using the weights, the model must be refined and adapted on the required data by tuning the parameters of the model

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes