Machine Learning

Wednesday, March 15, 2023 8:17 AM

1. How do you define Machine Learning?

Machine Learning is a field of study that enables machines to learn from data, without being explicitly programmed. In other words, it is a method of teaching computers to automatically improve their performance on a task by learning from data.

In practice, this involves training machine learning models on large datasets and using those models to make predictions or d ecisions about new data. Machine learning algorithms can be broadly classified into three categories: supervised learning, unsupervised lear ning, and reinforcement learning.

Supervised learning involves training a model on labeled data, where the desired output is already known. The model learns to make predictions by finding patterns in the input data that are associated with the correct output.

Unsupervised learning involves training a model on unlabeled data, where the desired output is not known. The model learns to identify patterns and structure in the data, which can be useful for tasks such as clustering and anomaly detection.

Reinforcement learning involves training a model to make decisions in an environment by receiving feedback in the form of rew ards or punishments. The model learns to maximize its reward over time by exploring different actions and observing their outcomes.

2. What do you understand by "labelled training dataset"?

Labelled training data refers to a dataset where each data point is associated with a label or a category that describes the desired output. For example, in an image classification task, each image would be associated with a label that identifies the object in the image (e.g. "cat", "dog", "car", etc.).

Labelled training data is used in supervised learning algorithms, where the goal is to train a model to predict the label or category of new, unseen data points based on their features. By providing the model with labelled training data, it can learn to recognize pat terns in the data that are associated with different labels, and use these patterns to make accurate predictions on new data.

Labelled training data is extremely useful in machine learning algorithm development because it allows the model to learn fro m examples of what it is trying to predict. Without labelled data, the model would not have any information about what it is trying to lear n, and it would not be able to make accurate predictions on new data.

In addition, labelled training data allows us to evaluate the performance of the machine learning model during development. B y comparing the predicted labels of the model to the true labels of the labelled data, we can measure the accuracy of the model and ident ify areas where it needs improvement. This allows us to iteratively refine the model until it achieves the desired level of accuracy on new, uns een data.

3. What are the two most common ML tasks you have performed so far? [update]

1. Data Visualization in EDA Phase
   I have worked with electric drive's time series data. The IoT device recorded at best second sampled data of operational signals like - current, speed, torque, temperatures and so on. Accounting to just the second and minute sampled data there were close to 40 features (signals)

   Visualization helped a lot in getting many insights about the signal's data
   - Distribution of the same signals across days/months and years. - Histograms / KDE Distribution plots (kernel smoothing for probability density estimation)
   - Signal patterns before failures and after component replacement - Trend plots
   - Relationship between signals, identifying correlation between signals or any underlying signal pattern - Scatter matrix
   - Outliers in a signal especially temperature signals - Boxplots

   The initial analysis was done using libraires like pyplot library of matplotlib and the seaborn library
   The later analysis for deeper understanding was done using advanced libraires like plotly

2. Dimensionality Reduction in Feature
   Since I had to deal with a lot of features , we opted for dimensionality reduction methods
   Before applying any DR methods, I took inputs from domain and findings from visualization for selecting features from all the available signals - a subset of the original features based on their relevance or importance to the target variable or the machine learning task

   After feature selection, we opted for feature extraction/feature transformation methods. - it is a process of transforming the original features into a new set of features that capture the essential information in the data.

   - PCA
   - t-SNE

4. What kind of ML algorithm would you use to walk a robot in various unknown area?

To walk a robot in various unknown areas, you would likely use a Reinforcement Learning (RL) algorithm. RL is a type of machine learning algorithm that learns through trial and error by interacting with an environment to maximize a reward signal.

In the case of walking a robot in unknown areas, the robot would need to learn how to move its legs and body to navigate the terrain and avoid obstacles. The RL algorithm would provide the robot with a reward signal based on its progress towards the goal (e.g. reaching a particular location) and penalize it for collisions or falls. The robot would then use this feedback to adjust its movements and learn how to walk more effectively.

RL is well-suited for tasks where the optimal solution is not known in advance, and where the agent (in this case, the robot) needs to learn through experience. However, RL algorithms can be computationally expensive and require a large amount of training data. Additionally, it may be necessary to use a simulation or a safe environment to train the robot before deploying it in the real world.

5. What kind of ML algorithm would you use to segment your user into multiple groups?

To segment users into multiple groups, you would likely use a Clustering algorithm. Clustering is an unsupervised learning technique that aims to partition a dataset into distinct groups or clusters based on their similarity or distance from each other.

The type of clustering algorithm to use depends on the nature of the data and the specific requirements of the segmentation task. Some common clustering algorithms include:

- K-means clustering: This is a popular clustering algorithm that partitions the dataset into K clusters based on the mean distance between data points.
- Hierarchical clustering: This algorithm builds a hierarchy of clusters by recursively merging or splitting clusters based on their similarity.
- Density-based clustering: This algorithm identifies clusters based on the density of data points in the dataset.

The output of a clustering algorithm is a set of cluster labels, where each label corresponds to a group of similar users. The clusters can then be analyzed and interpreted to gain insights into the characteristics and behavior of different user groups. For example, clustering can be used in customer segmentation to identify different customer segments based on their purchasing behavior or demographic information.

It is important to note that clustering is an unsupervised learning technique, and the quality of the segmentation depends on the quality of the data and the choice of clustering algorithm. It may be necessary to preprocess the data, remove outliers, and normalize the features before applying the clustering algorithm. Additionally, it may be necessary to validate the quality of the clustering results using metrics such as silhouette score or cluster purity.

6. What type of learning algorithm relies on similarity measure to make a prediction?

The type of learning algorithm that relies on similarity measure to make a prediction is Instance-based Learning.
Instance-based learning algorithms are also known as lazy learning algorithms. These algorithms work by storing the training data and making predictions based on the similarity between new instances (input data) and the stored instances. The algorithm does not learn an explicit model or representation of the data, but instead uses the stored instances to make predictions.

The most common instance-based learning algorithm is k-Nearest Neighbors (k-NN). k-NN works by finding the k-nearest instances in the training data to a given input instance and using their labels to make a prediction. The similarity between instances is typically measured using distance metrics such as Euclidean distance or cosine similarity.

Instance-based learning algorithms can be useful in situations where the data is noisy, the underlying distribution is unknown, or the relationship between the input and output variables is complex. However, they can be computationally expensive and may require a large amount of memory to store the training data. Additionally, the performance of the algorithm can be sensitive to the choice of distance metric and the value of k.

7. What is online learning system?

In the context of machine learning, an online learning system is a type of machine learning algorithm that learns from streaming data, rather than from a pre-defined, static dataset.

Online learning algorithms continuously update their model as new data becomes available. This means that the model can adapt to changing conditions and make predictions or decisions in real-time, without needing to retrain on the entire dataset.

Online learning systems are particularly useful in situations where data is rapidly changing, such as in financial forecasting, fraud detection, and recommendation systems. They can also be more computationally efficient than traditional batch learning algorithms, as they don't require all the data to be processed at once.

Some examples of online learning algorithms include stochastic gradient descent, online passive-aggressive algorithms, and online decision trees.

If I am able to train the model on continuous dataset.
We are doing batch learning.

8. What is out of core learning?

In the context of machine learning, out-of-core learning (also known as "out-of-memory" learning) refers to a technique used for training machine learning models on very large datasets that cannot fit into the computer's main memory (RAM).

Traditional machine learning algorithms typically assume that the entire dataset is available in memory, which can limit their scalability when working with large datasets. Out-of-core learning addresses this limitation by allowing algorithms to operate on smaller chunks of data, called "batches," that are read from disk one at a time.
During training, the model processes each batch of data sequentially, updating its parameters after each batch, until it has seen all the data. This process is repeated several times, or "epochs," to improve the accuracy of the model.

Out-of-core learning techniques can be used for a variety of machine learning algorithms, such as neural networks, decision trees, and support vector machines. They are particularly useful in applications like natural language processing, image recognition, and big data analytics, where the size of the dataset can be enormous.

9. Can you name a couple of ML challenges that you have faced? [update]

- All anomalies are not abnormalities
- Feature selection
- Curse of dimensionality - grouping using clustering
- Data labels missing so validation becomes difficult

10. Can you give 1 example of hyperparameter tuning w.r.t some classification algorithm?

Random Forest (w.r.t AD done previously with Arpit/Nikhil)
- n_estimators: The number of trees in the forest. A higher value of n_estimators can improve the performance of the model, but also increases computation time and memory requirements.
- max_depth: The maximum depth of each tree in the forest. A deeper tree can capture more complex relationships in the data, but can also lead to overfitting. Setting a lower max_depth can prevent overfitting, but may result in underfitting.
- Lr : learning rate in gradient descent
- Esp : epsilon in DBScan
- Min_samples in DBScan
- Threshold of p in Logistic Regression (0.5 to any other value)

11. What is out of bag evaluation ?

Out-of-bag (OOB) evaluation is a technique used in ensemble learning methods, such as Random Forests, to estimate the performance of a model on unseen data without the need for a separate validation set.

The OOB evaluation method works by randomly sampling data from the original dataset with replacement to create multiple decision trees in the Random Forest model. Some of the original data points are not used to build each of the decision trees, and these data points are known as out-of-bag (OOB) samples.
To evaluate the model's performance, the OOB samples are used as a validation set to estimate the model's accuracy. The accuracy of each individual decision tree is determined by comparing the predictions made on the OOB samples to their actual labels. The overall accuracy of the Random Forest model is then calculated by aggregating the individual accuracies of each decision tree.

This approach can be useful because it provides a way to estimate the model's performance without the need for a separate validation set. It also ensures that all of the data is used for training, which can lead to better performance compared to traditional cross-validation techniques.

12. What do you mean by hard and soft voting classifier ?

In machine learning, ensemble methods are used to combine multiple individual models to improve the overall performance of the prediction task. Voting classifier is one such ensemble method which is used to combine the predictions of multiple individual classifiers.

In a voting classifier, each individual classifier predicts the class label of an unseen instance, and the final prediction is made by taking the majority vote of all the individual classifiers. There are two types of voting classifier - hard voting and soft voting.

In hard voting, the predicted class label is the mode of the class labels predicted by the individual classifiers. That is, the class label with the highest number of votes is selected as the final prediction. This approach works well when the individual classifiers have different

decision boundaries and can accurately predict the class labels.
In soft voting, instead of considering only the predicted class label of each individual classifier, the class probabilities predicted by each classifier are taken into account. The final prediction is made by averaging the class probabilities predicted by all the individual classifiers and selecting the class with the highest average probability. This approach works well when the individual classifiers provide probability estimates for each class, and when these probabilities are well-calibrated.

Overall, soft voting is generally considered to be more effective than hard voting in most cases because it takes into account more information from the individual classifiers. However, the choice between the two approaches depends on the specific problem and the individual classifiers being used.

13. Let's suppose your ML algorithm is taking 5 min time to train, How will you bring down time to 5 second for training? (Hint: Distributed Computation)

Distributed computation can help to bring down the training time of a machine learning algorithm by dividing the computational workload among multiple machines or processors. This approach is often referred to as parallel processing.

In traditional single-machine environments, the amount of data that can be processed in a given amount of time is limited by the hardware capabilities of the machine, including the number of processors, the amount of RAM, and the available storage. By distributing the computation across multiple machines or processors, it is possible to overcome these limitations and perform computations on larger datasets and more complex algorithms.
Distributed computation can be achieved in several ways, such as:

- Data parallelism: In this approach, the same machine learning model is trained on different subsets of the data, and the results are combined at the end to create a final model. This approach is useful for algorithms that can be easily parallelized, such as decision trees and neural networks.
- Model parallelism: In this approach, different parts of the model are trained on different machines, and the results are combined to create a final model. This approach is useful for deep learning models that have a large number of layers.
- Task parallelism: In this approach, different parts of the algorithm are executed on different machines simultaneously, and the results are combined to create a final result. This approach is useful for algorithms that have multiple stages, such as feature extraction, model training, and evaluation.

Distributed computation can significantly reduce the training time of machine learning algorithms, enabling faster experimentation, model tuning, and deployment. However, it requires careful design and implementation to ensure that the distributed system is reliable, fault-tolerant, and scalable.

14. Let's Suppose I have trained 5 diff model with same training dataset & all of them have achieved 95% precision. Is there any chance that you can combine all these models to get better result? If yes, How? If no, Why?

Yes, it is possible to combine the predictions of multiple models to achieve better results than any individual model. This technique is called ensemble learning.
Ensemble learning combines the predictions of multiple models using various techniques such as:

- Voting: In this approach, the predictions of the individual models are combined by taking a majority vote. For example, if there are three models that predict the class labels for a given instance as A, B, and A, then the final prediction would be A.

- Weighted Voting: In this approach, each model's prediction is given a weight, and the final prediction is made by taking a weighted average of the predicted class probabilities. The weights are determined based on the performance of the individual models on a validation set.

- Stacking: In this approach, the predictions of the individual models are used as input to a meta-model, which learns to combine the predictions of the individual models to make the final prediction.

If all five models have achieved 95% precision, it is possible that by combining them using ensemble learning techniques, we can achieve better results than any individual model. However, the effectiveness of ensemble learning depends on several factors such as the diversity of the individual models, the correlation between their predictions, and the size and quality of the training and validation data.
In general, ensemble learning can be a powerful technique for improving the accuracy and robustness of machine learning models, and it is often used in real-world applications where high accuracy is critical.

15. What do you understand by Gradient decent? How will you explain Gradient decent to a kid?

Gradient descent is a technique used to train machine learning models to minimize errors in their predictions. Let me try to explain it in simple terms.
Imagine that you are walking down a hill trying to reach the bottom. You cannot see the entire hill at once, but you can only see a small portion of it that is in front of you. You want to take steps that lead you downhill so that you reach the bottom faster. In order to do that,

you need to know which direction is downhill.

Gradient descent works in a similar way. The machine learning algorithm starts at a random point in the model's parameter space and calculates the gradient of the loss function at that point. The gradient tells us which direction the loss function is decreasing the most quickly. The algorithm then takes a step in that direction and repeats the process, gradually moving closer to the point of minimum loss.

In other words, gradient descent is like a person walking down a hill, taking small steps in the direction that leads them downhill the fastest until they reach the bottom. The loss function is like the hill, and the gradient tells us which way is downhill. By following the gradient, we can minimize the loss function and train our machine learning models to make better predictions.

I hope this analogy helps to explain the concept of gradient descent to a kid.

16. Can you please explain diff between regression & classification?

Regression and classification are two different types of machine learning tasks used to analyze and make predictions about data. Regression is a type of supervised learning that involves predicting a continuous numerical value, such as the price of a house or the temperature of a city. The goal of regression is to find a function that can map the input data to a continuous output variable. Regression algorithms aim to minimize the difference between the predicted output and the actual output for a given set of input data.

Classification, on the other hand, is also a type of supervised learning but involves predicting a categorical or discrete variable. In classification, the goal is to classify input data into one of several predefined classes or categories. For example, classifying whether an email is spam or not, or identifying handwritten digits as a number from 0 to 9.

Classification algorithms aim to learn the relationship between input features and class labels to make accurate predictions on unseen data.
In summary, regression is used to predict continuous numerical values, while classification is used to classify data into different categories or classes.

17. Explain a clustering algorithm of your choice.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular unsupervised clustering algorithm used in machine learning to group together similar data points in a dataset. DBSCAN is particularly useful in scenarios where the number of clusters is unknown or the data is spread out unevenly.

The algorithm works by defining clusters based on the density of the data points. A data point is considered to be a core point if it has at least a specified minimum number of other points (MinPts) within a certain distance (Epsilon or ε) from it. Points that are not core points but are within the specified distance of core points are classified as border points. Any points that are not core or border points are classified as noise points.
The DBSCAN algorithm works as follows:

- Randomly select an unvisited data point and check its surrounding points to determine whether it is a core point or not.
- If the point is a core point, all points within the specified distance ε are added to the same cluster.
- If the point is not a core point, but is within the specified distance ε of a core point, it is added to the same cluster as that core point.
- Continue to visit the remaining unvisited data points until all data points have been visited.
- Any unvisited data points that are not within ε of any core points are classified as noise points.

DBSCAN has several advantages over other clustering algorithms, such as its ability to identify clusters of arbitrary shape and its robustness to noise and outliers in the data. However, the performance of the algorithm can be affected by the choice of ε and MinPts parameters, which need to be carefully selected based on the characteristics of the data.

18. How you can explain ML, DL, NLP, Computer vision & reinforcement learning with example in your own terms?

- Machine Learning (ML): Machine learning is a subset of artificial intelligence that involves training computers to learn patterns from data and make predictions or decisions without being explicitly programmed. An example of machine learning is a spam filter that learns to identify and filter out unwanted emails based on patterns it learns from analyzing thousands of emails.

- Deep Learning (DL): Deep learning is a subset of machine learning that uses neural networks with multiple layers to learn complex representations of data. An example of deep learning is a self-driving car that learns to identify and respond to traffic signals, pedestrians, and other objects on the road by analyzing vast amounts of sensor data.

- Natural Language Processing (NLP): Natural language processing is a field of artificial intelligence that involves teaching computers to understand and generate human language. An example of NLP is a virtual assistant like Siri or Alexa that can understand spoken commands and respond with appropriate actions.

- Computer Vision: Computer vision is a field of artificial intelligence that involves teaching computers to interpret and analyze visual data from the world around us. An example of computer vision is facial recognition technology used by security systems to identify people

from images or videos.

- Reinforcement Learning: Reinforcement learning is a type of machine learning that involves training computers to learn by trial and error. An example of reinforcement learning is a robot that learns to navigate a maze by receiving rewards for finding the right path and punishments for taking the wrong path.

19. How you can explain semi-supervised ML in your own way with example?

Semi-supervised learning is a type of machine learning that **combines both labeled and unlabeled data to train a model.** In traditional supervised learning, a machine learning model is trained on labeled data, where each example in the training set has a corresponding label or output value. In unsupervised learning, the model is trained on unlabeled data without any explicit feedback.

Semi-supervised learning, on the other hand, involves using both labeled and unlabeled data to train a model. The goal is to leverage the large amounts of unlabeled data that is often available in real-world scenarios to improve the performance of the model.

Semi-supervised learning algorithms **typically work by using the labeled data to train an initial model, and then use the unlabeled data to refine and improve the model.** This can be done by, for example, using the unlabeled data to generate additional training examples, or by using the unlabeled data to create a better representation of the underlying data distribution.

Semi-supervised learning is useful in situations where labeled data is scarce or expensive to obtain, but large amounts of unlabeled data are available. For example, in image recognition tasks, it may be difficult to obtain labeled examples for every possible object or scenario, but there may be a large amount of unlabeled data available that can be used to improve the model's performance.

Semi-supervised learning has been successfully applied in a variety of domains, including natural language processing, computer vision, and speech recognition.

20. What is difference between abstraction & generalization in your own word. (Oops concept)

In the context of computer science and programming, abstraction and generalization refer to two important concepts.
Abstraction is the process of removing unnecessary details from a problem or solution to simplify it and make it more manageable. It involves focusing on the essential aspects of a problem while ignoring the irrelevant or non-essential details. Abstraction helps in reducing the complexity of a problem and makes it easier to understand and solve.

On the other hand, generalization is the process of finding common patterns or characteristics among a set of specific instances or objects. It involves extracting common features from a set of examples and creating a more general model or concept that captures the essence of those examples. Generalization helps in creating a more comprehensive and flexible solution that can be applied to a wide range of scenarios.
In simpler terms, abstraction is about simplifying a complex problem by removing unnecessary details, while generalization is about finding commonalities among different specific instances to create a more general model or concept.

21. What are the steps that you have followed in your last project to prepare the dataset?

22. In your last project what steps were involved in model selection procedure?

23. If I give you 2 columns of any dataset, what will be the steps will be involved to check the relationship between those 2 columns?

If you have two columns of a dataset and you want to check the relationship between them, you can follow these steps:

- Visualize the data: Plot a scatter plot of the two columns to get an initial sense of their relationship. This will give you a visual representation of the data points and their distribution.
- Calculate correlation coefficient: Calculate the correlation coefficient between the two columns to determine the strength and direction of their linear relationship. The correlation coefficient can range from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no correlation.
- Conduct hypothesis testing: Depending on your research question, you may need to conduct hypothesis testing to determine if there is a statistically significant relationship between the two columns. This can be done using techniques such as t-tests or ANOVA.
- Check for outliers: Identify any outliers or extreme values that may be affecting the relationship between the two columns. Outliers can have a significant impact on the correlation coefficient and can skew the results.
- Consider other factors: Keep in mind that correlation does not necessarily imply causation. There may be other factors that are influencing the relationship between the two columns that are not accounted for in the analysis.

By following these steps, you can gain a better understanding of the relationship between the two columns and any underlying factors that may be affecting it.

24. Can you please explain 5 diff kind of strategies at least to handle missing values in dataset?

here are five different strategies to handle missing values in a dataset:

- Deletion: One way to handle missing values is to simply remove any rows or columns that contain missing values. This approach is straightforward and can be effective if the missing data is random and does not significantly impact the overall dataset. However, it can also lead to loss of valuable information and reduce the sample size.
- Imputation: Imputation involves replacing missing values with estimated values based on the rest of the data. There are several imputation techniques available, including mean imputation, median imputation, and regression imputation. Imputation can help preserve the sample size and reduce bias in the analysis, but the accuracy of the imputed values depends on the quality of the estimation technique used.
- Prediction: Another approach is to use machine learning algorithms to predict the missing values based on the patterns observed in the rest of the data. This can be done using techniques such as K-nearest neighbors (KNN) and decision trees. However, this approach can be computationally intensive and may require a large amount of data to accurately predict missing values.
- Extension: In some cases, missing data can be extrapolated from other related data sources. For example, missing temperature data can be estimated based on historical temperature data or data from nearby weather stations. This approach requires domain knowledge and expertise to identify appropriate sources of data.
- Domain-specific: Finally, some domains may have specific techniques or guidelines for handling missing data. For example, in clinical trials, missing data may be handled using imputation techniques based on the type of data and the stage of the trial. Domain-specific techniques can help ensure that missing data is handled in a way that is appropriate for the specific context and analysis.

Overall, the choice of missing data handling strategy depends on the specific dataset, the amount and pattern of missing data, and the research question at hand.

25. What kind of diff. issues you have faced w.r.t your raw data? At least mention 5 issues.

26. What is your strategy to handle categorical dataset? Explain with example.

Handling categorical data in a machine learning model is a crucial step as most machine learning algorithms only work with numerical data. Here are a few strategies to handle categorical data:

- Label Encoding: This technique is used to convert categorical data into numerical form. Each category is assigned a numerical label starting from 0 to n-1, where n is the number of categories. For example, if we have three categories "Red", "Green", and "Blue", we can encode them as 0, 1, and 2 respectively.

- One-Hot Encoding: This technique is used when there is no inherent order in the categories. In this technique, we create a new column for each category and encode it with a binary value of 0 or 1. For example, if we have three categories "Red", "Green", and "Blue", we can encode them as [1, 0, 0], [0, 1, 0], and [0, 0, 1] respectively.

- Binary Encoding: This technique is used when there are too many categories to one-hot encode. In this technique, we convert each category into a binary string and create new columns for each digit. For example, if we have three categories "Red", "Green", and "Blue", we can encode them as 00, 01, and 10 respectively.

- Frequency Encoding: In this technique, we encode each category with its frequency in the dataset. For example, if the category "Red" appears 5 times in the dataset, we can encode it as 5.

- Target Encoding: In this technique, we encode each category with the mean of the target variable for that category. For example, if we have a binary target variable and the category "Red" has a mean of 0.2, we can encode it as 0.2.

For example, suppose we have a categorical variable "Color" with categories "Red", "Green", and "Blue", and we want to predict the price of a product based on its color. We can use one-hot encoding to convert the categorical variable into numerical form. The resulting dataset would have three columns "Color_Red", "Color_Green", and "Color_Blue", with binary values of 0 or 1 indicating the color of the product. We can then use this dataset to train our machine learning model to predict the price of the product based on its color.

27. How do you define a model in terms of machine learning or in your own word?

In machine learning, a model is a mathematical or computational representation of a system or process that is used to make predictions or decisions based on data. A model is trained on a dataset using a learning algorithm, which involves finding the optimal values of model parameters that can minimize the error between the predicted outputs and actual outputs. The trained model can then be used to make predictions on new data that it has not seen before. The goal of a machine learning model is to generalize well on unseen data, and to avoid overfitting, which occurs when a model is too complex and performs well on the training data but poorly on the testing data.

28. What do you understand by k fold validation & in what situation you have used k fold cross validation?

K-fold cross-validation is a technique used to evaluate the performance of a machine learning model on a limited data sample. It involves splitting the data into k subsets, where one subset is used as the testing set and the other k-1 subsets are used as the training

set. This process is repeated k times, with each subset being used as the testing set once. The results are then averaged over the k iterations to get a final estimate of the model's performance.

K-fold cross-validation is useful when the size of the dataset is limited and we want to maximize the use of available data. It helps to reduce the variance of the model and provides a more accurate estimate of the model's performance. K-fold cross-validation can be used in any situation where we want to evaluate the performance of a machine learning model, such as classification or regression.

For example, let's say we have a dataset of 1000 samples and we want to train a logistic regression model to predict whether a customer will buy a product or not based on their age, gender, and income. We can use k-fold cross-validation to evaluate the performance of the model. We can split the data into 10 subsets, use 9 subsets as the training set, and one subset as the testing set. We can repeat this process 10 times, using each subset as the testing set once. We can then calculate the average accuracy over the 10 iterations to get a more accurate estimate of the model's performance.

29. What is meaning of bootstrap sampling? explain me in your own word.

Bootstrap sampling is a resampling technique used in statistics to estimate the population parameters. It involves randomly selecting a subset of the original data with replacement to create a new dataset. This new dataset is of the same size as the original data, but with some of the original data points appearing multiple times and some not at all.

The bootstrap sampling technique allows for the estimation of the distribution of the sample mean and other statistics of the population, even when the population distribution is unknown or the sample size is small. It is particularly useful in situations where it is difficult or impossible to obtain a large sample size, or when the original sample is biased or contains outliers.

In machine learning, bootstrap sampling is often used in combination with other techniques such as k-fold cross-validation to evaluate and improve the performance of predictive models. By randomly resampling the original dataset multiple times, we can obtain multiple estimates of the model's performance, which can be averaged to get a more reliable estimate.

30. What do you understand by underfitting & overfitting of model with example?

Underfitting and overfitting are the two most common problems that occur while training machine learning models.
Underfitting happens when a model is too simple to capture the underlying patterns in the data, resulting in poor performance on both the training and test data. In other words, the model is not able to learn the important features of the data and is too generalized to make accurate predictions.

For example, let's consider a dataset of housing prices based on the size of the house. If we use a linear model to predict the price of the house based only on the size of the house, it would be underfitting since the price of the house depends on several other factors like location, number of rooms, etc. Therefore, the model would perform poorly on both the training and test data.

On the other hand, overfitting happens when a model is too complex and captures the noise and random fluctuations in the training data, resulting in poor performance on the test data. In other words, the model is too specific to the training data and does not generalize well to new data.

For example, if we use a high degree polynomial model to predict the price of the house based on the size of the house, the model may fit the training data very well but may not perform well on the test data. This is because the model is too complex and captures the noise in the training data, resulting in poor performance on new data.

Therefore, it is important to strike a balance between underfitting and overfitting by choosing an appropriate model complexity and regularizing the model to avoid overfitting.

31. What is diff between cross validation and bootstrapping?

Both cross-validation and bootstrapping are resampling techniques used in machine learning.

Cross-validation is a technique used to estimate the performance of a machine learning model by splitting the data into k-folds and using each fold for testing and the remaining k-1 folds for training. This process is repeated k times, and the results are averaged to get an estimate of the model's performance. Cross-validation is generally used to tune hyperparameters of a model and to estimate the model's generalization performance.

Bootstrapping, on the other hand, is a resampling technique where multiple random samples are drawn with replacement from the original dataset to create new datasets. These new datasets are then used to estimate the uncertainty of a model. Bootstrapping is often used to estimate the confidence interval or standard error of a model's performance metrics or to obtain a robust estimate of the model's parameters.

In summary, cross-validation is used to estimate the performance of a model, while bootstrapping is used to estimate the uncertainty of a model.

32. What do you understand by silhouette coefficient?

The silhouette coefficient is a metric used to evaluate the quality of clusters in unsupervised machine learning algorithms such as K-Means, DBSCAN, and hierarchical clustering. The silhouette coefficient measures how well a data point fits into its assigned cluster compared to its fit in neighboring clusters.
The silhouette coefficient ranges from -1 to 1, with higher values indicating better clustering performance. A silhouette coefficient close to 1 indicates that the data point is well-clustered and is closer to its own cluster's centroid than to other clusters' centroids. A silhouette coefficient close to -1 indicates that the data point is not well-clustered and is closer to other clusters' centroids than to its own cluster's centroid. A silhouette coefficient close to 0 indicates that the data point is on or very close to the decision boundary between two clusters.

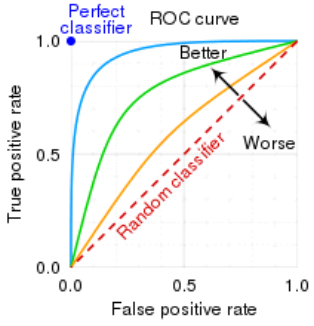
The silhouette coefficient is calculated as follows for each data point:

- Calculate the average distance between the data point and all other points in the same cluster. This is called the "intra-cluster distance".
- Calculate the average distance between the data point and all other points in the next closest cluster. This is called the "nearest-cluster distance".
- Calculate the silhouette coefficient for the data point as (nearest-cluster distance - intra-cluster distance) divided by the maximum of the two.
- Repeat steps 1-3 for all data points in the dataset and calculate the average silhouette coefficient across all data points to evaluate the overall clustering performance.

In general, a higher average silhouette coefficient indicates better clustering performance. However, the silhouette coefficient should be used in combination with other evaluation metrics and visual inspection to properly evaluate the quality of clusters.

33. What is the advantage of using ROC Score?

ROC (Receiver Operating Characteristic) score is a performance metric used in classification models to evaluate the performance of a model at various classification thresholds. The ROC score measures the tradeoff between the true positive rate (sensitivity) and false positive rate (1-specificity) at different thresholds.



The advantage of using ROC score is that it is more robust to class imbalance and uneven costs of false positives and false negatives. It provides a better measure of the model's ability to distinguish between the positive and negative classes and can be used to compare the performance of different classification models. The ROC curve can also be used to determine the optimal threshold for the model based on the desired tradeoff between sensitivity and specificity.

34. Explain me complete approach to evaluate your regression model

The complete approach to evaluate a regression model involves the following steps:

- Splitting the data: The first step is to split the data into two parts: training data and testing data. The training data is used to train the model, while the testing data is used to evaluate the performance of the model.
- Building the model: The second step is to choose a regression algorithm and train the model on the training data. The choice of the algorithm depends on the problem at hand and the characteristics of the data.
- Evaluating the model: Once the model is trained, it is time to evaluate its performance. There are several metrics that can be used to evaluate a regression model, including:
  - Mean squared error (MSE)
  - Root mean squared error (RMSE)
  - Mean absolute error (MAE)
  - R-squared (R^2)

  These metrics are used to compare the predicted values of the model with the actual values.
- Tuning the model: If the performance of the model is not satisfactory, it may be necessary to tune the model. This involves changing the hyperparameters of the model or trying a different algorithm to see if it can improve the performance.
- Final evaluation: Once the model has been tuned, it is evaluated again on the testing data to see if the performance has improved. If the performance is satisfactory, the model can be deployed for production use.

  Overall, the goal of the evaluation process is to build a model that is accurate and generalizes well to new data.

35. Give me example of lazy learner and eager learner algorithms example.

Lazy learner algorithms are those that do not build any generalization model during the training phase and wait until the time of prediction. One of the most common examples of a lazy learner algorithm is k-Nearest Neighbors (k-NN), which predicts the class of a test instance based on the majority class of the k nearest training instances.

Eager learner algorithms, on the other hand, build a generalization model during the training phase, which is then used to make predictions for new instances. Examples of eager learner algorithms include decision trees, random forests, and support vector machines. These algorithms build a model using the training data and then use it to classify new instances.

36. What do you understand by holdout method?

Holdout method is a technique in machine learning that involves splitting the dataset into two parts: a training set and a validation set. The training set is used to fit the model, while the validation set is used to evaluate the performance of the model.

In the holdout method, a random subset of the data is selected for training the model and the remaining subset is used for validation. Typically, the data is split into a 70-30 or 80-20 ratio for training and validation, respectively. The model is trained on the training set and then evaluated on the validation set to estimate its performance.

The holdout method is a simple and straightforward approach to model evaluation, but it has some drawbacks. It can lead to overfitting if the training set is too small, and it can be biased if the data is not randomly sampled. To overcome these issues, other techniques such as cross-validation are used.

37. What is diff between predictive modelling and descriptive modelling.

Predictive modeling and descriptive modeling are two different types of statistical analysis used in data science.
Predictive modeling is a type of modeling that involves creating a model that can predict the outcome of future events based on historical data. It uses statistical techniques and machine learning algorithms to identify patterns in the data and make predictions about future events. The goal of predictive modeling is to accurately predict the outcome of an event, based on the available data.

On the other hand, descriptive modeling is a type of modeling that is used to describe the characteristics of a given dataset. Descriptive modeling is typically used to summarize data and provide insights into the underlying patterns and relationships. The goal of descriptive modeling is to gain an understanding of the data, rather than making predictions about future events.
To summarize, predictive modeling is used to predict future events, while descriptive modeling is used to describe existing data.

38. How you have derived a feature for model building in your last project?

39. Explain 5 different encoding techniques.

There are several encoding techniques used in machine learning to convert categorical variables into numerical format. Here are five common techniques:

- One-Hot Encoding: In this technique, a new column is created for each category of the categorical variable, and a binary flag is set for each row that corresponds to that category. This is a commonly used encoding technique for nominal categorical variables.
- Label Encoding: In this technique, each category of the categorical variable is assigned a unique numerical value. This technique is used for ordinal categorical variables.
- Binary Encoding: This technique is similar to One-Hot Encoding, but it reduces the number of columns by half. Instead of creating a new column for each category, a binary value is assigned to each category, and these binary values are concatenated to form a single column.
- Count Encoding: In this technique, each category is replaced by the count of its occurrences in the dataset. This technique is useful when the number of categories is very high.
- Target Encoding: In this technique, each category is replaced by the mean of the target variable for that category. This technique is useful when the target variable is continuous and the number of categories is moderate.

These techniques are used to preprocess the categorical variables before feeding them to machine learning models, as most machine learning models require numerical input. The choice of encoding technique depends on the nature of the data and the machine learning model being used.

40. How do you define some features are not important for ML model? What strategy will you follow

To determine if some features are not important for a machine learning model, there are various strategies that can be followed, such as:

- Feature importance techniques: This involves using algorithms that provide information on feature importance. For example, decision trees can be used to calculate feature importance scores. Other algorithms such as Random Forest and Gradient Boosting also have built-in methods to calculate feature importance.

- Correlation analysis: This involves analyzing the correlation between each feature and the target variable. If the correlation coefficient is close to zero, it suggests that the feature is not important.
- Univariate feature selection: This involves selecting the features that are most relevant to the target variable using statistical tests such as chi-squared or ANOVA.
- Recursive Feature Elimination (RFE): This is an iterative approach that involves removing the least important feature at each step until the desired number of features is reached.
- Expert knowledge: Sometimes, domain knowledge or expert knowledge can be used to determine which features are important and which are not.
  Once the unimportant features are identified, they can be removed from the dataset. This can help improve the performance of the model by reducing overfitting, improving model accuracy, and reducing training time.

41. What is difference between Euclidian distance and Manhattan distance. Explain in simple words.

Euclidean distance and Manhattan distance are both methods of calculating the distance between two points in a multi-dimensional space.
Euclidean distance is the most common distance metric and is calculated as the square root of the sum of the squared differences between corresponding elements of the two vectors. It is like the straight-line distance between two points. For example, if you have two points (x1, y1) and (x2, y2), then the Euclidean distance between them can be calculated as follows:

distance = sqrt((x2 - x1)^2+ (y2 - y1)^2)

Manhattan distance, on the other hand, is the sum of the absolute differences between corresponding elements of the two vectors. It is like the distance a taxi would travel on a grid-like city block. For example, if you have two points (x1, y1) and (x2, y2), then the Manhattan distance between them can be calculated as follows:

distance = |x2 - x1| + |y2 - y1|

In general, Euclidean distance is more sensitive to outliers and is affected by the scale of the variables, while Manhattan distance is less sensitive to outliers and is not affected by the scale of the variables. The choice of distance metric depends on the specific problem and the nature of the data.

42. What do you understand by feature selection, transformation, engineering and EDA &What are the steps that you have performed in each of these in detail with example.

Feature selection : In this phase domain or business case in hands plays a major role. There are many ways to find relation between the features and their impact on the output (methods like correlation, visualization, statistical analysis) but the inputs from domain to select the features has a upper hand.

Feature transformation : The process to deriving new features from the existing features. Example: Feature decomposition techniques like PCA, SVD, mathematical combinations of features etc.

Feature engineering : The process of transforming the data to a format that is suitable as an input for the model like image data to arrays or tensors, text data to encoded numeric, logarithmic or exponential conversion of features etc.

EDA : Data visualization, statistical analysis to draw inferences about the data.

43. What is difference between single values decomposition (SVD) and PCA? (hint: SVD is one of the way to do PCA)

Issues with too many features

1. Model would not be able to generalize to the optimized value of weights (example: For a case with 100 features the model will be required to optimize 100m and biases in case of linear regression which is not possible)
2. Computationally very high
3. Time complexity

Both Single Value Decomposition (SVD) and Principal Component Analysis (PCA) are linear algebra techniques used for dimensionality reduction, but they have some fundamental differences:

- Objective: PCA is a statistical method that aims to identify the principal components that explain the maximum variance in the data, whereas SVD is a matrix factorization method that decomposes a matrix into its constituent parts to facilitate further analysis.
- Input Data: PCA operates on the covariance matrix of the input data, whereas SVD can operate on any rectangular matrix. SVD is more general and can be applied to a wider range of problems beyond data analysis.
- Output: The output of PCA is a set of orthogonal principal components, which are linear combinations of the original variables, while the output of SVD is a set of orthogonal left and right singular vectors and singular values.
- Dimensionality Reduction: In PCA, the number of principal components is usually chosen to capture a certain percentage of the

variance in the data, and the remaining components are discarded. In SVD, the number of singular values can also be used to reduce the dimensionality of the data.
- Applications: PCA is commonly used in exploratory data analysis, feature extraction, and image compression, while SVD has a wide range of applications in signal processing, image processing, and machine learning, including collaborative filtering and recommendation systems.

In summary, both SVD and PCA are powerful techniques for analyzing high-dimensional data and reducing its complexity, but they have different objectives, input data requirements, outputs, and applications.

Important Explanation :
SVD (Single Value Decomposition) is actually one way to perform PCA (Principal Component Analysis). PCA is a technique used for dimensionality reduction, where a set of correlated variables is transformed into a smaller set of uncorrelated variables, known as principal components. SVD is a matrix factorization method that decomposes a matrix into three matrices, including two orthogonal matrices and a diagonal matrix, and is commonly used for data compression and dimensionality reduction.
In fact, PCA can be computed using the SVD of the data matrix, where the singular vectors of the SVD correspond to the principal components of the data. This is because the singular vectors are orthonormal and capture the direction of maximum variance in the data. So, by taking the top k singular vectors of the SVD, we can construct a new matrix with fewer dimensions that captures most of the variance in the original data. Therefore, SVD is one of the most common methods used for computing PCA.

44. What kind of feature transformations you have done in your last project?

- Data Smoothening
- Data Aggregation
- Generalization
- Normalization

45. Have you taken any external feature in any of project from any 3rd party data? If yes, explain that scenario.

Not third party but Fault log information and failure dates from MOR can be considered as external features

46. If your model is overfitted, what you will do next?

If a model is overfitted, it means that it has performed well on the training data but not on the testing data. In such a situation, the following steps can be taken to address overfitting:

- Collect more data: If the dataset is small, collecting more data can help to generalize the model better.
- Feature selection: Removing irrelevant or redundant features can reduce overfitting.
- Regularization: This involves adding a penalty term to the cost function to reduce the impact of large weights in the model.
- Cross-validation: Cross-validation techniques such as k-fold cross-validation can be used to estimate the performance of the model and to prevent overfitting.
- Ensemble learning: Using ensemble methods such as bagging and boosting can help to reduce overfitting by combining multiple models.
- Early stopping: This involves stopping the training of the model when the performance on the validation set starts to degrade.
- Reduce model complexity: This involves reducing the complexity of the model by decreasing the number of layers, nodes, or parameters.

Ans by Sudhanshu:
A overfitted model is high bias and low variance. Solution - cross validation, pruning, hyper parameter tuning, regularization (L1, L2), early stopping, drop out (in case of NN), dimensionality reduction

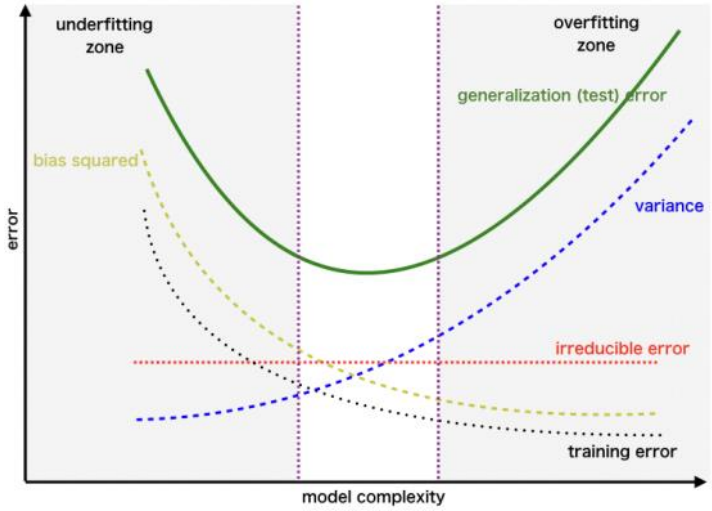
47. Explain me bias variance trade-off.

Bias-variance trade-off is a fundamental concept in machine learning that refers to the trade-off between the ability of a model to accurately capture the underlying patterns in the data and its ability to generalize to new, unseen data.

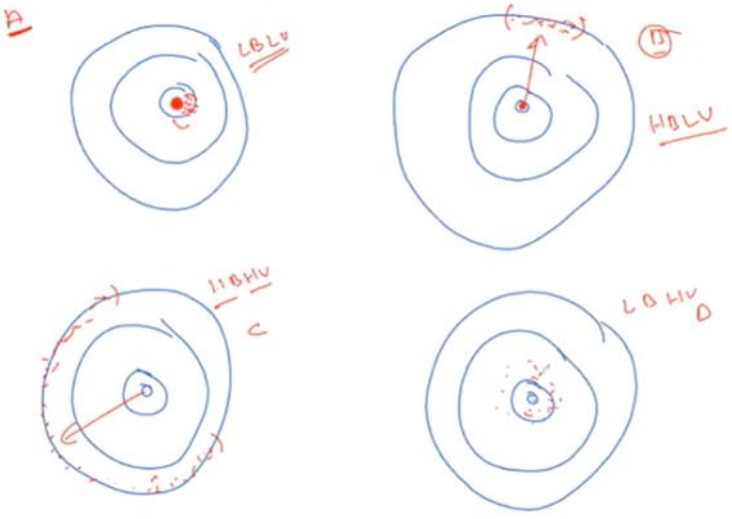
Bias refers to the error that is introduced by approximating a real-life problem with a simple model. High bias models are too simplistic and cannot capture the underlying patterns in the data. These models may result in underfitting, meaning they are unable to learn the patterns in the training data or generalize to new data.

Variance refers to the error that is introduced by the model's sensitivity to small fluctuations in the training data. High variance models are too complex and can overfit to the training data, meaning they capture noise in the data and cannot generalize to new data.

The goal of machine learning is to find a model that has low bias and low variance, which means that it can accurately capture the underlying patterns in the data while being able to generalize to new data. However, reducing one comes at the expense of the other, so it is important to find the right balance between bias and variance. This can be achieved by adjusting the model complexity or by using techniques such as regularization, cross-validation, and ensemble learning.



Bias : How far the predicted data is from the expected data/ actual data
Variance : The scatterness of the predicted data



48. What steps would you take to improve accuracy of your model? At-least mention 5approach. And justify why would you choose those approach

There are several steps one can take to improve the accuracy of a model. Here are five approaches that can be taken:

- Feature engineering: This involves analyzing the data and selecting relevant features for the model. It can involve creating new features, combining existing features or selecting a subset of the features. Feature engineering can help to reduce the noise in the data and improve the model's ability to generalize to new data.
- Hyperparameter tuning: The performance of many machine learning algorithms is dependent on the values of their hyperparameters. Hyperparameters are values set by the user that are used to control the behavior of the algorithm. By tuning these hyperparameters, it is possible to find a configuration that leads to better performance on the data.
- Cross-validation: Cross-validation is a technique used to assess the performance of a model. By dividing the data into multiple subsets and testing the model on each subset, it is possible to get a more accurate estimate of the model's performance. This can help to identify overfitting and other issues with the model.
- Ensemble methods: Ensemble methods involve combining the predictions of multiple models to improve accuracy. This can be done using techniques such as bagging, boosting or stacking. Ensemble methods can help to reduce the variance in the model and improve its ability to generalize to new data.
- Data cleaning: Data cleaning involves removing noise, missing values or outliers from the data. By cleaning the data, it is possible to improve the quality of the data and reduce the chances of the model overfitting. Data cleaning can involve techniques such as imputation, outlier removal or data normalization.

The choice of approach depends on the specific problem and the nature of the data. For example, if the data is noisy, data cleaning may be a good first step. If the model is overfitting, cross-validation and hyperparameter tuning may be more appropriate. Ultimately, the

goal is to find the approach or combination of approaches that leads to the best performance on the data.

Answer by Sudhanshu :

- Look into the data set, increase the volume of the dataset.
- Remove outliers
- Too many features : Dimensionality reduction
- Imbalance dataset : balance it out - up or down sampling
- Feature engineering, selection , transformation
- Multiple models - Stacking, ensemble

49. Explain process of feature engineering in context of text categorization.

Feature engineering is a crucial step in text categorization where we transform the raw text data into feature vectors that can be used as input for machine learning models. The process of feature engineering in text categorization involves the following steps:

- Text Preprocessing: This step involves cleaning the text data by removing stop words, punctuations, and other unnecessary elements. It also includes stemming, lemmatization, and tokenization to convert the text into a standard format.
- Feature Extraction: Once the text data is cleaned, the next step is to extract relevant features from the text. This can be done using various techniques such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), word embeddings, and n-grams.
- Feature Selection: After feature extraction, we need to select the most relevant features for our model. This can be done using techniques such as chi-square test, mutual information, and correlation-based feature selection.
- Feature Transformation: Once the relevant features are selected, we need to transform them to make them more suitable for machine learning models. This can be done using techniques such as normalization, scaling, and dimensionality reduction.
- Model Training: Finally, we train our machine learning models using the transformed features and evaluate their performance using appropriate metrics.

Choosing the right approach for feature engineering depends on the specific problem we are trying to solve and the characteristics of the text data. For example, if we have a large amount of text data, we may choose to use TF-IDF for feature extraction as it can handle large datasets efficiently. Similarly, if we have a lot of noisy data, we may choose to use feature selection techniques to remove irrelevant features and improve the performance of our model.

50. Explain vectorization and hamming distance.

Vectorization is the process of representing data as numerical vectors in order to perform mathematical operations and analysis on it. In the context of natural language processing, vectorization involves converting text into numerical vectors that can be used in machine learning algorithms. There are several techniques for vectorization, such as bag-of-words, word embeddings, and term frequency-inverse document frequency (TF-IDF).

Hamming distance is a measure of the difference between two binary strings of equal length. It is calculated by counting the number of positions where the corresponding bits in the two strings are different. For example, the Hamming distance between the binary strings "11001" and "10101" is 2, because they differ in the 2nd and 4th positions. Hamming distance is used in various applications such as error detection and correction, data compression, and cryptography.

51. Can you please explain chain rule and its use?

In calculus, the chain rule is a method used to calculate the derivative of composite functions. The chain rule states that if we have a function f(x) and g(x) such that the output of g(x) is the input to f(x), then the derivative of the composite function $(f \circ g)(x)$ is given by the product of the derivative of f with respect to its input and the derivative of g with respect to its input. Mathematically, it can be written as:

$d(f \circ g)/dx = (df/dg) * (dg/dx)$

where df/dg represents the derivative of f with respect to g, and dg/dx represents the derivative of g with respect to x.

The chain rule is an essential tool in calculus and is used in various fields such as physics, engineering, and economics to calculate the derivatives of composite functions. It allows us to break down complex functions into simpler ones and find their derivatives. For example, in neural networks, the chain rule is used to calculate the gradients of the loss function with respect to the weights of the network, which is necessary for training the network using backpropagation.

52. What is difference between correlation and covariance?

Correlation and covariance are two measures that describe the relationship between two variables. The main differences between correlation and covariance are as follows:

- Definition: Correlation is a normalized measure that describes the strength and direction of the linear relationship between two variables. Covariance, on the other hand, is a measure of the joint variability of two random variables.
- Scale: Correlation coefficients range between -1 and +1, where -1 indicates a perfect negative linear relationship, 0 indicates no linear relationship, and +1 indicates a perfect positive linear relationship. Covariance can take any value between negative infinity and positive infinity, where a positive value indicates a positive relationship and a negative value indicates a negative relationship.
- Units: Correlation is unitless, which means that it does not depend on the units of the variables being measured. Covariance, however, has units that are the product of the units of the two variables being measured.
- Interpretation: Correlation is a more intuitive measure of the strength of the relationship between two variables because it is normalized and ranges between -1 and +1. Covariance, however, is less intuitive because it is influenced by the scale of the variables being measured.
- Calculation: The correlation coefficient can be calculated directly from the covariance matrix, by dividing the covariance by the product of the standard deviations of the two variables. So, correlation is a normalized form of covariance.
- In summary, while both correlation and covariance measure the relationship between two variables, correlation is a more standardized measure that is unitless and ranges between -1 and +1, while covariance is a measure of the joint variability of two random variables and depends on the scale of the variables.

Answer by Sudhanshu :
Correlation tries to represent how a dataset is strongly related to each other, covariance tries to measure how a change in one impacts the other feature.

53. What are the sampling techniques you have used in your project?

- Random sampling
- Stratified sampling
- Stratified cross validation (when we have imbalanced data set)
- Cluster based sampling
- Snowball sampling

54. Have you ever used Hypothesis testing in your last project, if yes, explain How?

Null hypothesis : The given feature is going to impact the target
Alternate hypothesis : There is no impact of a given feature onto the target

On running statistical test we can conclude if a feature impacts the value of the target and if it should be considered or not

55. In which case you will use naïve Bayes classifier and decision tree separately?

Naive Bayes classifiers and decision trees are both popular machine learning algorithms that can be used for a variety of classification tasks. The choice of which algorithm to use depends on the specific characteristics of the data and the requirements of the application. Here are some general guidelines for when to use Naive Bayes classifiers and decision trees separately:

Use Naive Bayes classifiers when:

- The data has a large number of features compared to the number of instances.
- **The features are independent of each other** (or nearly independent) given the class label.
- The distribution of the features is not important, as long as they can be estimated from the data.
- The training data is limited or noisy, and a simple model is preferred to avoid overfitting.
  For example, Naive Bayes classifiers are commonly used in spam email detection, sentiment analysis, and document classification tasks.

Use decision trees when:

- The data has both categorical and numerical features.
- **The relationships between the features are complex and nonlinear.**
- The data contains missing values or noisy data that can be dealt with by the algorithm.
- Interpretability is important, and a tree-based model can provide insights into the decision-making process.
  For example, decision trees are commonly used in credit scoring, customer segmentation, and medical diagnosis tasks.

It's important to note that these are just general guidelines, and the choice of algorithm ultimately depends on the specific characteristics of the data and the requirements of the application. In some cases, it may be necessary to experiment with both algorithms and compare their performance to determine which one works best.

56. What is the adv & disadvantage of naïve Bayes classifier, explain

Naive Bayes Classifier is a probabilistic machine learning algorithm that is based on Bayes' theorem. It is widely used in various applications such as spam filtering, text classification, and sentiment analysis. The main advantages and disadvantages of the Naive Bayes Classifier are as follows:
Advantages:

- Naive Bayes Classifier is a simple and easy-to-understand algorithm. It is easy to implement and requires very little training data.
- It can handle **high-dimensional datasets with a large number of features**, making it suitable for text classification and natural language processing applications.
- Naive Bayes Classifier is a **fast algorithm** and can be trained and applied to new data quickly.
- It performs well in practice, even with the assumption of independence between features, making it a reliable and robust algorithm.
- It works well with both binary and categorical features.

- Disadvantages:
- **If categorical variable is not seen before, the algorithm while testing shows zero probability, hence algorithm fails to detect**
- Naive Bayes Classifier **assumes that all features are independent**, which is not always the case in real-world datasets. This can result in inaccurate predictions when there is strong correlation between features.
- It relies on **strong assumptions about the distribution (gaussian) of the data and can perform poorly when these assumptions are violated**.
- Naive Bayes Classifier tends to perform poorly when the test set contains categories that were not present in the training set.
- It is a probabilistic model and may not always provide the correct predictions with high certainty.
- It is sensitive to the quality of the input data and can be affected by outliers or missing values in the dataset.

In summary, Naive Bayes Classifier is a simple, fast and reliable algorithm that can handle high-dimensional datasets with categorical and binary features. However, it has some limitations such as the assumption of independence between features and the sensitivity to data quality.

57. In case of numerical data what is naïve Bayes classification equation you will use?

In the case of numerical data, the Naive Bayes classification equation typically involves **assuming a Gaussian distribution** for each feature given the class label. The equation for the Gaussian Naive Bayes classifier can be expressed as follows:

Equation of y will be as follows because x will be normally distributed in Naïve Bayes is used

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

P(y|x) = P(y) * P(x1|y) * P(x2|y) * ... * P(xn|y)

Where:
- y is the class label (i.e., the target variable)
- x is the set of input features (i.e., the independent variables)
- P(y|x) is the posterior probability of the class label given the input features
- P(y) is the prior probability of the class label
- P(xi|y) is the conditional probability of feature xi given the class label y, assuming a Gaussian distribution

The conditional probability of each feature xi given the class label y can be estimated by fitting a Gaussian distribution to the values of xi for all instances in the training set that belong to the class y. This involves calculating the mean and variance of xi for the instances in the class y, and using these parameters to define a Gaussian distribution. The prior probability of each class can be estimated by calculating the frequency of each class in the training set.

Once the conditional and prior probabilities have been estimated, the classifier can be used to predict the class label of new instances by calculating the posterior probability of each class given the input features, and selecting the class with the highest probability as the predicted class.

It's important to note that the Naive Bayes classifier makes the naive assumption that all features are independent of each other given the class label, which may not always hold true in practice. However, despite this limitation, Naive Bayes classifiers can still be effective in many real-world applications, particularly when dealing with high-dimensional data.

58. Give me scenario where I will be able to use a boosting classifier and regressor?

Boosting is a machine learning technique that combines several weak learners (i.e., models that perform only slightly better than random guessing) to create a strong learner that can make accurate predictions. Here are two scenarios where boosting can be used for classification and regression tasks:

- Boosting Classifier: Suppose we have a dataset with many features and a large number of classes, and we want to train a classifier to accurately classify each data point into the correct class. In this scenario, we can use a boosting classifier such as AdaBoost or Gradient Boosting to train a series of weak classifiers on the data, with each subsequent classifier focused on correcting the errors of the previous classifiers. Boosting can be particularly useful when dealing with imbalanced datasets, where some classes have very few examples, as it can help improve the classification accuracy of the minority classes.

- Boosting Regressor: Suppose we have a dataset with a large number of features and a continuous target variable that we want to predict accurately. In this scenario, we can use a boosting regressor such as AdaBoost or Gradient Boosting to train a series of weak regressors on the data, with each subsequent regressor focused on correcting the errors of the previous regressors. Boosting can be particularly useful when dealing with non-linear relationships between the features and the target variable, as it can help capture the non-linearities and improve the prediction accuracy.
  Overall, boosting can be a powerful technique for improving the performance of classifiers and regressors in scenarios where the data is complex, noisy, or high-dimensional. However, it can also be computationally expensive and may require careful tuning of hyperparameters to avoid overfitting.

59. In case of Bayesian classifier what exactly it tries to learn. Define its learning procedure.

Bayesian classifiers are a type of probabilistic classifier that use Bayes' theorem to classify data based on its probability of belonging to a certain class. In the context of Bayesian classification, the algorithm tries to learn the probability distribution of the input features and the class labels, given a training set of labeled examples.

The learning procedure of a Bayesian classifier involves two main steps: parameter estimation and classification.

- Parameter estimation: In this step, the algorithm estimates the probability distributions of the input features and the class labels using the training set. This involves calculating the prior probability of each class, as well as the conditional probability of each feature given each class. In other words, the algorithm tries to learn how each feature is related to each class.
- Classification: Once the probability distributions have been estimated, the algorithm can be used to classify new, unlabeled data points. To classify a new data point, the algorithm calculates the posterior probability of each class given the input features using Bayes' theorem, and selects the class with the highest probability as the predicted class.

The key advantage of Bayesian classifiers is that they can handle missing data and noisy data well, and can provide a measure of uncertainty in the classification results. However, they can be computationally expensive and may require a large amount of training data to accurately estimate the probability distributions.

- Always tried to find probability of occurrence of A w.r.t B

60. Give me a situation where I will be able to use SVM instead of Logistic regression.

SVM and logistic regression are both popular algorithms for binary classification tasks, but there are certain situations where SVM may be preferred over logistic regression. Here is an example:

Suppose we have a dataset with many features and a non-linear decision boundary between the classes. In this scenario, logistic regression may not be the best choice as it assumes a linear decision boundary between the classes. SVM, on the other hand, can handle non-linear decision boundaries using non-linear kernels such as the RBF kernel.

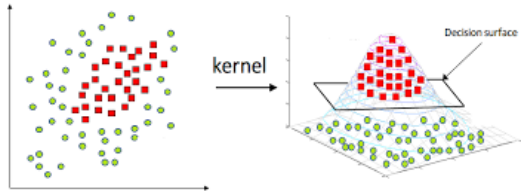
For example, consider a problem of classifying images of handwritten digits into different classes. Each image can be represented as a high-dimensional feature vector, where each pixel is a feature. In this scenario, there may not be a linear decision boundary between the classes, as the shapes of the digits can be complex and non-linear. In such cases, SVM with an RBF kernel may be a better choice than logistic regression, as it can capture the non-linear relationships between the features and the class labels.

Overall, SVM is preferred over logistic regression when dealing with complex and non-linear decision boundaries in high-dimensional feature spaces. However, it is important to note that SVM can be computationally expensive and may require more parameter tuning than logistic regression.

- Non linear -> SVM, Linear dataset - Logistic

61. What do you understand by rbf kernel in SVM?

What is kernel trick ?
Convert lower dimension data to higher dimension data



Converting 2D to 3D helps in separating the data , evident visually.
Kernel trick is used to find the third coordinate to easily separate the data.

This is what is done by RBF

The RBF (Radial Basis Function) kernel is a commonly used kernel in Support Vector Machines (SVM) for classification and regression tasks. It is a non-linear kernel that **allows SVM to model non-linear decision boundaries by transforming the input data into a higher-dimensional space.**
The RBF kernel is defined as:

K(x, xi) = exp(-gamma ||x - xi||^2) ; gamma = 1/2 sigma^2 where sigma = population standard deviation

where x and xi are data points, gamma is a hyperparameter that controls the shape of the kernel, and ||x - xi||^2 is the squared Euclidean distance between the two data points in the input space.

The RBF kernel maps the input data to an infinite-dimensional feature space, where it can be easier to find a linear boundary that separates the classes. The hyperparameter gamma controls the width of the kernel, which in turn determines the degree of non-linearity in the decision boundary. A higher value of gamma will result in a narrower kernel, leading to a more complex decision boundary that may overfit the training data, while a lower value of gamma will result in a wider kernel, leading to a simpler decision boundary that may underfit the data.

Overall, the RBF kernel is a powerful tool for SVM as it can model non-linear decision boundaries in high-dimensional feature spaces, making it suitable for a wide range of classification and regression problems.

62. Give me 2 scenarios where AI can be used to increase revenue of travel industry.

There are many scenarios where AI can be used to increase revenue in the travel industry, but here are two examples:

- Personalized Recommendations: AI can be used to analyze customer data and recommend personalized travel packages and experiences that align with their interests, preferences, and travel history. By leveraging machine learning algorithms and natural language processing, AI can analyze a vast amount of customer data and make targeted recommendations that can increase the likelihood of customers booking travel with a particular company.

- Dynamic Pricing: AI can also be used to optimize pricing strategies in real-time. By analyzing customer behavior, historical data, and market trends, AI can recommend pricing adjustments that can maximize revenue while also ensuring that customers are getting a fair price. For example, AI can recommend price adjustments for specific routes, travel times, or travel dates based on demand and supply, maximizing the revenue for the travel companies while also providing customers with competitive pricing. This dynamic pricing can be implemented in both the airline and hotel sectors to increase revenue.

63. What do you understand by leaf node in decision tree?

A leaf node in a decision tree is a node that has no child nodes. In other words, it is a terminal node that **represents a final decision or classification based on the values of the input features**. Once a leaf node is reached during the traversal of the decision tree, the corresponding prediction or decision is made.

For example, consider a decision tree that predicts whether or not a person will buy a product based on their age, income, and gender. The tree may have multiple internal nodes, each representing a decision based on one of the input features. However, once all the decisions have been made and the traversal reaches a leaf node, the final prediction is made (e.g., "buy product" or "do not buy product").

Leaf nodes are important in decision tree algorithms because they represent the end result of the decision-making process. By recursively splitting the data based on the selected attributes, the decision tree is grown until leaf nodes are reached, and the corresponding predictions or decisions are made. The accuracy of the decision tree model is determined by how well it predicts the correct outcomes at the leaf nodes.

64. What is information gain & Entropy in decision tree?

Information gain and entropy are both concepts used in decision trees to determine the best attribute to split the data at each node. Entropy is a measure of impurity in the data. The entropy of a set of examples S with respect to a binary classification problem (i.e., where each example can be classified as either positive or negative) is defined as:

Entropy(S) = -p_pos log2 p_pos - p_neg log2 p_neg

where p_pos and p_neg are the proportions of positive and negative examples in S. Entropy is highest (i.e., most impure) when p_pos = p_neg = 0.5, and lowest (i.e., most pure) when one class dominates (i.e., p_pos or p_neg is close to 1.0).

Information gain, on the other hand, is a measure of the reduction in entropy achieved by splitting the data at a particular attribute. The information gain of an attribute A with respect to a set of examples S is defined as:

InformationGain(A) = Entropy(S) - Sum(w_i * Entropy(S_i))

where w_i is the proportion of examples in S that belong to subset S_i after splitting on attribute A. The information gain of an attribute is highest when it leads to the greatest reduction in entropy (i.e., the most information is gained by splitting on that attribute).

In decision tree algorithms, the attribute with the highest information gain is chosen as the splitting criterion at each node, as it provides the most information about the classification problem. By recursively splitting the data based on the selected attributes, the decision tree is grown until a stopping criterion is met (e.g., a maximum depth is reached or the impurity of the remaining examples falls below a certain threshold).

65. Give disadvantages of using Decision tree

While decision trees are a popular and powerful machine learning technique, they also have some disadvantages. Here are some of the main disadvantages of using decision trees:

- Overfitting: Decision trees are prone to overfitting when they become too complex and capture noise in the data. Overfitting can be mitigated by pruning the tree or using ensemble methods like Random Forest.
- Instability: Small changes in the data can lead to different decision trees being generated, which can make the model unstable and difficult to interpret.
- Bias: Decision trees can be biased towards features with many levels or high cardinality, which can lead to suboptimal splits and lower accuracy.
- Difficulty with continuous variables: Decision trees can have difficulty handling continuous variables, which can lead to poor splits and lower accuracy.
- Lack of smoothness: Decision trees can produce jagged decision boundaries, which can lead to overfitting and poor generalization.
- Interpretability: While decision trees are often cited as being interpretable, this can be difficult when the tree becomes too large or complex.
- Computationally expensive
- Post pre pruning leads to loss of data

Can decision tree be used for high variance dataset?
Yes. It will give me clarity in terms of making clear separation. But it might end up overfitting.

Overall, decision trees can be a powerful machine learning technique when used correctly, but they do have some limitations that need to be taken into account. By carefully choosing hyperparameters and avoiding overfitting, decision trees can be a valuable tool for data analysis and prediction.

66. List some of the features of random forest.

Random Forest is a powerful ensemble learning algorithm that combines multiple decision trees to improve the performance and generalization of the model. Some of the key features of random forest include:

- Ensemble method: Random Forest combines multiple decision trees to create a more robust and accurate model. Each tree in the forest is trained on a randomly sampled subset of the training data, and the final prediction is made by averaging the predictions of all the trees.
- Bagging: Random Forest uses a technique called bagging (bootstrap aggregating) to create the random subsets of the training data for each tree. This helps to reduce overfitting by introducing randomness into the training process.
- Feature selection: Random Forest selects a random subset of features at each split point in the decision tree, which helps to reduce the correlation between the trees and improves the diversity of the forest.
- Non-parametric: Random Forest is a non-parametric algorithm, which means it does not make assumptions about the distribution of the data. This makes it suitable for a wide range of applications.
- High accuracy: Random Forest has been shown to achieve high accuracy on a wide range of classification and regression tasks.
- Robust to noisy data: Random Forest is robust to noisy and missing data, thanks to its ability to handle missing values and outliers.
- Easy to use: Random Forest is relatively easy to use, and requires minimal parameter tuning compared to other algorithms.
  Overall, Random Forest is a versatile and powerful algorithm that can be used in a wide range of applications, including classification, regression, and feature selection.

- It does not overfit like decision tree
- It has less parameter tuning
- Can handle both continuous and categorical variables

- It works with missing values
- Robust to outliers
- Reduces variance
- Highlights important features
- Better decision maker (bagging approach) so less bias

67. How can you avoid overfitting in decision tree?

Always first look into the data set

Overfitting in decision trees occurs when the model is too complex and fits the training data too closely, leading to poor generalization to new data. To avoid overfitting in decision trees, we can use several techniques, including:

- Pruning: This involves removing branches or nodes from the tree that do not improve the performance on a validation set. There are two types of pruning: pre-pruning and post-pruning. Pre-pruning involves stopping the tree growth early, based on a maximum depth or minimum number of samples per leaf, while post-pruning involves removing branches that do not improve the performance on a validation set.

- Setting minimum samples per leaf: This involves setting a minimum number of samples required to be at a leaf node of the tree. This reduces the complexity of the tree and prevents the model from fitting noise in the data.

- Setting maximum depth: This involves limiting the maximum depth of the tree to prevent the model from becoming too complex and overfitting the data.
- Using ensemble methods: This involves combining multiple decision trees to reduce overfitting. Bagging, Boosting, and Random Forest are some popular ensemble methods used in decision trees.

- Increasing the amount of data: Adding more training data can help the model generalize better to new data and reduce overfitting.

- K fold cross validation

- Overall, avoiding overfitting in decision trees requires balancing the complexity of the model with its ability to generalize to new data. By using the techniques mentioned above, we can improve the performance of the decision tree model on new, unseen data.

68. Explain polynomial regression in your own way.

Polynomial regression is a type of regression analysis that models the relationship between a dependent variable and one or more independent variables by fitting a polynomial function to the data.

Unlike linear regression, which assumes a linear relationship between the independent and dependent variables, polynomial regression models can capture more complex relationships that may not be linear. This is achieved by adding higher-order terms to the linear regression model, such as squared or cubed terms of the independent variable. The resulting function is a polynomial equation of degree n, where n is the highest order of the terms included in the model.

For example, if we want to model the relationship between the height of a plant (dependent variable) and the amount of fertilizer applied (independent variable), we might use a polynomial regression model with a quadratic term to capture any non-linearities in the relationship. The resulting equation might look something like:
height = b0 + b1*fertilizer* + *b2*fertilizer^2
where b0, b1, and b2 are the coefficients of the model that are estimated from the data.

To fit a polynomial regression model to the data, we can use the same techniques as for linear regression, such as ordinary least squares (OLS) or gradient descent. The goal is to find the values of the coefficients that minimize the difference between the predicted values of the dependent variable and the actual values in the data.

Once the model is trained, we can use it to make predictions on new data by plugging in the values of the independent variable(s) and calculating the corresponding predicted value of the dependent variable using the learned coefficients.
Overall, polynomial regression is a useful tool for modeling non-linear relationships between variables, and can be used in a wide range of applications, such as finance, economics, and biology.

69. Explain learning mechanism of linear regression.

The learning mechanism of linear regression involves finding the best linear relationship between a set of input variables (also known as features or predictors) and a continuous output variable (also known as the response or target variable).

The goal of linear regression is to find a line that best fits the data, where "best" is defined as minimizing the difference between the predicted values and the actual values of the output variable. This is achieved by minimizing a cost function, such as the mean squared error (MSE) or mean absolute error (MAE), which measures the difference between the predicted and actual values.

To find the best line, the linear regression algorithm uses an optimization algorithm, such as **gradient descent, to iteratively adjust the parameters of the line until the cost function is minimized**. The parameters of the line are the intercept (also known as the bias) and the slope (also known as the weight) for each input variable.

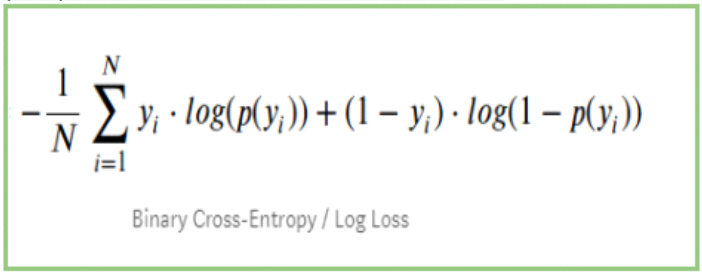
During training, the algorithm repeatedly calculates the predictions of the output variable based on the current parameter values, and then updates the parameters based on the gradient of the cost function with respect to the parameters. This process continues until the cost function is minimized or a stopping criterion is met, such as a maximum number of iterations or a minimum change in the cost function.
Once the model is trained, it can be used to make predictions on new data by simply plugging in the values of the input variables and calculating the corresponding output value using the learned parameters.

Overall, the learning mechanism of linear regression involves iteratively adjusting the parameters of a linear model to minimize a cost function, in order to find the best linear relationship between the input variables and the output variable.

3 pen approach by Sudhanshu ????

70. What is the cost function in logistic regression?

The cost function in logistic regression is used to measure the difference between the predicted probability of the model and the actual output for each data point. The goal of logistic regression is to find the optimal values for the coefficients that minimize this cost function and maximize the accuracy of the model. The most commonly used cost function in logistic regression is the Binary Cross-Entropy (BCE) loss function, which is defined as follows:



p(y) = predicted y

BCE = - (1/n) * sum(y * log(y_pred) + (1-y) * log(1-y_pred))
where:

- y is the actual binary output value (0 or 1)
- y_pred is the predicted probability of the model that the output is 1
- n is the number of data points in the dataset

The BCE cost function measures the difference between the predicted probability of the model and the actual output for each data point. It is commonly used in binary classification problems, where the goal is to predict a binary output based on a set of input variables. During the training process of logistic regression, the optimizer tries to minimize the BCE cost function by updating the coefficients of the model. This is typically done using gradient descent or other optimization algorithms. By minimizing the cost function, the logistic regression model is able to make better predictions on new, unseen data.

71. What is the error function in linear regression?

The error function in linear regression is also called the cost function or the objective function. The goal of linear regression is to fit a line to a set of data points in such a way that the line best represents the relationship between the independent variable(s) and the dependent variable. The error function measures the difference between the predicted output and the actual output for each data point and provides a way to evaluate how well the line fits the data.
In linear regression, the most commonly used error function is the Mean Squared Error (MSE), which is calculated as the average of the squared differences between the predicted output and the actual output for each data point:

MSE = (1/n) * sum((y - y_pred)^2)
where:

- y is the actual output value
- y_pred is the predicted output value
- n is the number of data points in the dataset

The goal of linear regression is to minimize the MSE by finding the optimal values for the slope and intercept of the line that best fits the data. This is typically achieved using an optimization algorithm such as gradient descent or normal equations.
Other error functions that can be used in linear regression include Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), although MSE is the most commonly used.

In regression analysis, there are several commonly used evaluation metrics to measure the performance of the model. These include

Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R2, and Adjusted R2. Here's an explanation of each of these metrics along with their formulas:

- Mean Squared Error (MSE):
  The mean squared error (MSE) is the average of the squared differences between the predicted output and the actual output for each data point.
  MSE = (1/n) * sum((y - y_pred)^2)
  where:
  - y is the actual output value
  - y_pred is the predicted output value
  - n is the number of data points in the dataset

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

- Root Mean Squared Error (RMSE):
  The root mean squared error (RMSE) is the square root of the MSE and represents the standard deviation of the residuals.
  RMSE = sqrt((1/n) * sum((y - y_pred)^2))

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\|y(i) - \hat{y}(i)\|^2}{N}},$$

- Mean Absolute Error (MAE):
  The mean absolute error (MAE) is the average of the absolute differences between the predicted output and the actual output for each data point.
  MAE = (1/n) * sum(|y - y_pred|)

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

- R2:
  The R2 metric, also known as the coefficient of determination, measures the proportion of variance in the dependent variable that is explained by the independent variables in the model.
  R2 = 1 - (sum((y - y_pred)^2) / sum((y - mean(y))^2))
  where:
  - y is the actual output value
  - y_pred is the predicted output value
    R2 ranges from 0 to 1, with higher values indicating a better fit of the model to the data.

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

Formula

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination
RSS = sum of squares of residuals
TSS = total sum of squares

- Adjusted R2:
  The adjusted R2 metric adjusts the R2 value for the number of independent variables in the model, to penalize overfitting.
  Adjusted R2 = 1 - ((1-R2)*(n-1)/(n-k-1))
  where:
  - n is the number of data points in the dataset
  - k is the number of independent variables in the model

  The adjusted R2 ranges from 0 to 1, with higher values indicating a better fit of the model to the data. It is typically used to compare the performance of models with different numbers of independent variables.

$$R^2_{adjusted} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

where
R^2 = sample R-square
p = Number of predictors
N = Total sample size.

72. What is the use of implementing OLS technique wrt dataset?

OLS stands for Ordinary Least Squares. It is a statistical method used to estimate the parameters of a linear regression model. In OLS, the goal is to find the line of best fit that minimizes the sum of the squared distances between the observed values and the predicted values. This is achieved by finding the values of the coefficients that minimize the residual sum of squares (RSS).

The OLS technique is useful in analyzing datasets where there is a linear relationship between the independent and dependent variables. It is commonly used in econometrics, finance, and other fields where linear regression models are frequently used.

By implementing OLS on a dataset, we can estimate the parameters of a linear regression model and make predictions based on the relationship between the independent and dependent variables. This can help us to understand the factors that affect a particular outcome and to make predictions about future observations.

Overall, OLS is a widely used and powerful technique for analyzing linear regression models and can provide valuable insights into relationships between variables in a dataset.

- Forward and backward elimination of the data using adjusted R2 and p values (explore)
- Statistical significance of features or labels

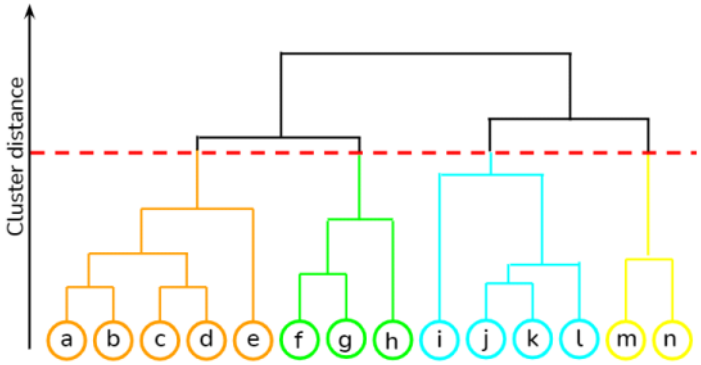
73. Explain dendrogram in your own way.

A dendrogram is a tree-like diagram that is commonly used in hierarchical clustering to represent the arrangement of clusters and their relationships with each other.

At the bottom of the dendrogram, each observation or data point is represented as a single point or dot. As we move up the dendrogram, these points start to cluster together and form larger groups. The height of the branches represents the distance or dissimilarity between these clusters. The longer the branch, the greater the distance between the clusters.

The dendrogram can help us understand the structure of the data and identify clusters that are similar or dissimilar to each other. By cutting the dendrogram at a certain level, we can form different numbers of clusters and compare their characteristics. We can also use the dendrogram to identify outliers or observations that don't fit well into any of the clusters.

Overall, the dendrogram is a useful tool for visualizing the results of hierarchical clustering and understanding the relationships between the different clusters in the data.



74. How do you measure quality of clusters in DBSCAN?

DBSCAN is a density-based clustering algorithm that groups data points based on their proximity to each other. The quality of the clusters in DBSCAN can be evaluated using a variety of metrics.

One common way to evaluate the quality of the clusters is to use the silhouette coefficient. The silhouette coefficient measures how well a data point fits into its assigned cluster compared to its fit in neighboring clusters. A high silhouette score indicates that the data point is well-clustered, while a low score indicates that the data point may be better suited to a different cluster.

Another way to evaluate the quality of the clusters is to use the Davies-Bouldin Index (DBI). The DBI measures the average similarity between each cluster and its most similar cluster, relative to the average distance between each cluster's data points. A lower DBI value indicates better clustering performance.
The Calinski-Harabasz Index (CHI) is another metric that can be used to measure the quality of clusters in DBSCAN. The CHI measures the ratio of between-cluster variance to within-cluster variance. A higher CHI value indicates better clustering performance.

In addition to these metrics, it's also important to visually inspect the clusters and ensure that they make intuitive sense based on the underlying data.

75. How do you evaluate DBSCAN algorithm?

There are several methods to evaluate the performance of DBSCAN algorithm:

- Visual Inspection: One of the simplest ways to evaluate DBSCAN is by visually inspecting the resulting clusters. This involves plotting the data and examining the clusters to see if they make intuitive sense based on the underlying data.
- Silhouette Score: The Silhouette score is a measure of how well each data point fits into its assigned cluster relative to other clusters. A high Silhouette score indicates that the point is well-clustered, while a low score suggests that the point may be better suited to a different cluster. The average Silhouette score across all points in the dataset can be used to evaluate the overall performance of the DBSCAN algorithm.
- Davies-Bouldin Index (DBI): The DBI is a measure of cluster separation and compactness. A lower DBI score indicates better clustering performance, with values closer to zero indicating more clearly separated and compact clusters.
- Calinski-Harabasz Index (CHI): The CHI measures the ratio of between-cluster variance to within-cluster variance. Higher CHI values indicate better clustering performance, with values closer to infinity indicating clearly separated and compact clusters.
- Adjusted Rand Index (ARI): The ARI measures the similarity between the true labels and the predicted cluster labels. A score of 1 indicates perfect clustering, while a score of 0 indicates that the clustering is no better than random.
- Fowlkes-Mallows Index (FMI): The FMI is similar to ARI but is less sensitive to noise in the data. A score of 1 indicates perfect clustering, while a score of 0 indicates that the clustering is no better than random.
- Jaccard similarity coefficient: The Jaccard similarity coefficient measures the similarity between the true labels and the predicted cluster labels. A score of 1 indicates perfect clustering, while a score of 0 indicates that the clustering is no better than random.

In general, a combination of multiple evaluation metrics and visual inspection is recommended to properly evaluate the performance of the DBSCAN algorithm.

76. What do you understand by market basket analysis?

Market basket analysis is a data mining technique used to analyze customer purchasing patterns and identify relationships between

different products or items. It is often used by retailers to understand customer behavior, improve product offerings, and increase sales. The basic idea behind market basket analysis is to analyze transaction data and identify which products are frequently purchased together. This can be done by calculating metrics such as support, confidence, and lift, which provide information about the frequency and strength of the relationships between different items.

Support measures the frequency of occurrence of a particular item or set of items in the transaction data. Confidence measures the likelihood that an item will be purchased given that another item has already been purchased. Lift measures the strength of the relationship between two items and indicates how much more likely an item is to be purchased when another item is also purchased.

By analyzing the results of market basket analysis, retailers can gain insights into customer preferences and behavior, which can help them to optimize their product offerings, promotions, and marketing strategies. For example, if customers frequently purchase coffee and cream together, a retailer might consider offering a discount on cream to increase sales of both items.

77. Explain centroid formation technique in K Means algorithm.

The K Means algorithm is a popular unsupervised machine learning algorithm used for clustering data points into K number of clusters. One of the key steps in the K Means algorithm is the formation of centroids, which represent the center of each cluster.
The centroid formation technique in K Means works as follows:

- Initialization: The algorithm first randomly selects K number of initial centroids, which serve as the center of the initial clusters.
- Assigning data points to clusters: Each data point in the dataset is then assigned to the closest centroid based on their distance using a distance metric such as Euclidean distance.
- Recalculating centroids: Once all data points are assigned to clusters, the centroid of each cluster is recalculated by taking the mean of all data points assigned to that cluster.
- Re-assigning data points: All data points are then re-assigned to their nearest centroid based on the updated centroid locations.
- Repeat: Steps 3-4 are repeated until the centroids converge and the assignments no longer change, or until a predefined number of iterations is reached.

The formation of centroids in K Means is an iterative process that aims to minimize the sum of squared distances between each data point and its assigned centroid. This technique can be used to group similar data points together into K number of clusters, with the resulting clusters being represented by the centroids.

78. Have you ever used SVM regression in any of your project, If yes, Why?

79.

GINI impurity is a measure of the impurity or uncertainty in a dataset, often used in decision trees and other machine learning algorithms. It measures the probability of incorrectly classifying a randomly chosen element from the dataset if it were randomly labeled based on the distribution of classes in the dataset.
The GINI impurity is calculated as the sum of the squared probabilities of each class in the dataset, subtracted from 1. This gives a measure of the probability of misclassifying a randomly chosen element, with a higher value indicating higher impurity or uncertainty.

The formula for GINI impurity can be written as:
$G = 1 - \sum(p\_i)^2$

where G is the GINI impurity, and p_i is the probability of an element belonging to class i.

In decision trees, the GINI impurity is used as a criterion for selecting the best split point for a given node. The split point with the lowest GINI impurity is chosen, as it will result in the greatest reduction in impurity in the resulting child nodes.

Overall, the concept of GINI impurity is an important tool for assessing the quality of splits in decision trees and other classification algorithms, and for understanding the uncertainty or impurity in a given dataset.

80. Let’s suppose I have given you dataset with 100 column how you will be able to control growth of decision tree?

When working with a dataset that has a large number of columns, it is important to control the growth of the decision tree in order to prevent overfitting and improve generalization performance. Here are some techniques that can be used to achieve this:

- Feature selection: Instead of using all 100 columns in the dataset, we can select a subset of the most informative features and use them to build the decision tree. This can be done using techniques such as correlation analysis, information gain, or other feature selection methods.

- Pre-pruning: This involves stopping the tree-building process before it reaches its maximum depth or when a certain number of samples are reached in a given node. This helps to prevent overfitting and ensures that the resulting tree is smaller and more interpretable.

- Post-pruning: This involves building a complete decision tree and then removing or merging nodes that do not contribute significantly to the overall performance of the tree. This technique can help to simplify the tree and make it more interpretable while still maintaining good predictive performance.

- Ensemble: Instead of building a single decision tree, we can build multiple trees and combine their predictions using techniques such as bagging or boosting. This can help to reduce the variance of the model and improve its generalization performance.

Overall, controlling the growth of the decision tree is important for improving its interpretability, reducing overfitting, and improving its generalization performance on new data. The specific techniques used will depend on the characteristics of the dataset and the goals of the analysis.

81. If you are using Ada-boost algorithm & if it is giving you underfitted result What is the hyperparameter tuning you will do?

If AdaBoost is giving an underfitted result, it means that the model is not complex enough to capture the patterns in the data. In this case, we can try to increase the complexity of the model by tuning the hyperparameters. Here are some possible hyperparameter tuning techniques that can be applied:

- Increase the number of estimators: AdaBoost builds an ensemble of weak learners, and increasing the number of estimators can improve the complexity of the model and its ability to capture the patterns in the data.
- Increase the learning rate: The learning rate controls the contribution of each estimator to the final model. A higher learning rate can make the model more aggressive in correcting the errors, which can improve its performance.
- Increase the depth of the base estimator: AdaBoost can use any algorithm as the base estimator, and increasing the depth of the base estimator can make it more capable of capturing the patterns in the data.
- Change the base estimator: AdaBoost can use any algorithm as the base estimator, and switching to a more complex algorithm can improve the ability of the model to capture the patterns in the data.
- Use feature selection: Removing unimportant features can simplify the problem and make it easier for the model to learn.
- Increase the sample size: Adding more samples to the training set can help the model to better capture the patterns in the data.

The specific hyperparameters that need to be tuned will depend on the characteristics of the data and the performance metrics being optimized.

82. Explain gradient boosting algorithm.

Gradient boosting is a machine learning algorithm that is used for both regression and classification problems. It is a boosting technique in which multiple weak models are combined to form a single strong model. The algorithm works by iteratively adding weak models to the ensemble and fitting them to the residual errors of the previous models.
Here is a step-by-step explanation of how gradient boosting works:

- Initialize the model: Start with a simple model that can be easily optimized, such as a decision tree with a depth of 1.
- Make predictions: Use the current model to make predictions on the training set.
- Compute the residuals: Compute the difference between the predicted values and the actual values in the training set.
- Train the next model: Train a new weak model on the residuals computed in step 3.
- Combine the models: Add the new model to the ensemble, and combine it with the previous models.
- Repeat: Repeat steps 2-5 until the desired performance is achieved, or until a stopping criterion is met.

The key idea behind gradient boosting is that each new model in the ensemble should be optimized to correct the errors of the previous models. This is achieved by training each model on the residuals of the previous models. The final prediction is obtained by aggregating the predictions of all the models in the ensemble.
Gradient boosting has several hyperparameters that can be tuned, such as the learning rate, the number of trees in the ensemble, and the maximum depth of the trees. By adjusting these hyperparameters, the performance of the model can be improved.

83. Can we use PCA to reduce dimensionality of highly non-linear data.

Though PCA is great, it does have some drawbacks. One of the major drawbacks of PCA is that it does not retain non-linear variance. This means PCA will not be able to get results for figures like this.

Trefoil Knot.

In simple terms, PCA works on retaining only global variance, and thus retaining local variance was the motivation behind t-SNE

t-SNE is a nonlinear dimensionality reduction technique that is well suited for embedding high dimension data into lower dimensional data (2D or 3D) for data visualization.
t-SNE stands for t-distributed Stochastic Neighbor Embedding, which tells the following :
Stochastic → not definite but random probability
Neighbor →concerned only about retaining the variance of neighbor points
Embedding → plotting data into lower dimensions

84. How do you evaluate performance of PCA.

PCA (Principal Component Analysis) is a popular technique used for dimensionality reduction, feature extraction, and data visualization. There are several ways to evaluate the performance of PCA, some of which are:

- Explained Variance: PCA aims to capture as much of the variance in the original data as possible. The amount of variance explained by each principal component can be calculated and used to evaluate the performance of PCA. A higher amount of explained variance indicates that PCA has successfully reduced the dimensionality of the data while retaining most of the information.
- Reconstruction Error: PCA also involves projecting the original data onto a lower-dimensional space. The performance of PCA can be evaluated by comparing the reconstructed data from the reduced dimensionality space with the original data. A lower reconstruction error indicates that the PCA has done a good job of preserving the structure of the original data.
- Clustering or Classification Performance: PCA can also be used as a pre-processing step for clustering or classification algorithms. The performance of these algorithms can be evaluated before and after applying PCA to see if there is any improvement in performance.
- Visualization: PCA can be used to reduce high-dimensional data into a 2D or 3D space for visualization purposes. The visual inspection of the data points in the reduced dimensional space can provide insights into the structure of the data and help in evaluating the performance of PCA.
- Overall, the choice of evaluation metric depends on the specific application and goal of using PCA.

85. Have you ever used multiple dimensionality techniques in any project? if yes, give reason. If no, where can we use it?

86. What do you understand by curse of dimensionality explain with help of example

The curse of dimensionality refers to the phenomenon where the performance of machine learning algorithms deteriorates as the number of dimensions or features in the dataset increases, making it harder to find patterns and relationships in the data. As the number of dimensions increases, the amount of data required to cover the space of possible values grows exponentially, making it harder to obtain enough data to accurately represent the distribution of the data.
For example, let's consider a dataset of 1000 points in a 2-dimensional space (x,y). If we increase the number of dimensions to 3 (x,y,z), we now need 1000^3 or 1 billion data points to cover the same space with the same density of data. As we increase the number of dimensions further, the number of data points required grows even more exponentially.

This can cause several problems for machine learning algorithms, such as increased computation time, overfitting, and poor generalization performance. In high-dimensional spaces, the data becomes more sparse and the distance between nearest neighbors increases, making it harder to accurately classify or cluster the data. Additionally, high-dimensional spaces tend to have more complex and irregular decision boundaries, which can make it harder for algorithms to find the optimal solution.

To mitigate the curse of dimensionality, it is often useful to reduce the number of dimensions by using techniques such as Principal Component Analysis (PCA) or feature selection. These techniques can help to identify the most important features and reduce the number of dimensions, allowing algorithms to better identify patterns and relationships in the data.

87. What is the difference between anomaly detection and novelty detection

Anomaly detection and novelty detection are both techniques used in machine learning to identify outliers or unusual data points, but they differ in their goals and the types of data they are designed to handle.

Anomaly detection is the task of identifying data points that are significantly different from the majority of the data, often indicating some kind of problem or error in the data. Anomaly detection algorithms are typically trained on a dataset of "normal" or expected data points, and then identify data points that deviate significantly from this expected distribution. Anomaly detection can be applied to a wide range of data types, including numerical, categorical, and text data, and is commonly used in fraud detection, network intrusion detection, and

fault detection in industrial processes.

Novelty detection, on the other hand, is the task of identifying new or unknown data points that do not match the expected distribution of the training data. Novelty detection algorithms are trained on a dataset of "normal" data points, and then identify any new data points that do not fit this expected distribution as potential novelties or outliers. Novelty detection is often used in scenarios where the goal is to detect unexpected or previously unknown events, such as in detecting new types of network attacks or identifying rare diseases in medical images.

In summary, while both anomaly detection and novelty detection are used to identify outliers or unusual data points, anomaly detection focuses on identifying significant deviations from expected data, while novelty detection focuses on identifying new or unknown data points that do not match the expected distribution of the training data.

ML part-2

Tuesday, March 28, 2023 12:27 PM

1.Tell me something about your project you have done in past?
2. What was your Dataset size for ML Project?
3. What is type of your dataset?
4. What was frequency of your dataset? (E.g. batch, streaming etc.)
5. What was source system for your dataset? (E.g. sensor, satellite Kafka, cloud, etc.)
6. What was kind of derived dataset that you have mentioned in project?
7. How you have done validation dataset?
8. Have you created any pipeline to validated this dataset or you were using any tool?
9. What do you understand by data lake?
10. What do you understand by data warehousing?
11. Can you please name some validations that you have done on top of your data?
12. How you have handled streaming dataset?
13. How many different types of environments were available in your project?
14. What was your delivery mechanism for particular project?
15. Have you used any OPS pipeline for this current project?
16. How you were doing model retraining?
How you have implemented model retraining in your project?
18. How frequently you have been doing model retraining and what was the strategy for model retraining?
19. What was kind of evaluation you were doing in production environment
20. What was no. of request (hits) your model was receiving on daily basis?
21. How you have implemented logging in project for any failure cases?
22. How you have integrated notification (or Alarm) system for your project?
23. How you have implemented model monitoring?
24. How you have derived final KPI (Key Performance Indicator) for your client?
25. How many dashboards were there in your project?
26. On which platform you have productionized your model?
27. What kind of API you have exposed to receive data for model?
28. What was size of your final production environment (system configuration)?
What an all Databases you have used in project?
30. What kind of optimization you have done in your project, till what depth & explain the example
31. Can you please talk about complete team structure and team size?
32. What was duration of your complete project?
33. What was your day to day responsibility in last 2 month?
34. What kind of change request you have been receiving after you productionized project
35. What kind of testing you have done in development, UAT, pre-pod and prod?
36. Have you used some of the predefined AI-OPS pipelines if yes explain
37. Who has implemented AI-OPS in your project?
38. What was OPS stack you have been using?
39. What do you understand by CI-CD & have you implemented those in your project. If yes, what was the tech stack you used for CI-CD pipeline?
40. What was biggest challenge you faced in project and how you have resolved it?
41. Give me one scenario where you worked as team player?
What was your overall learning from current project?
43. How do you keep yourself updated for new technology?

44. Have you designed an architecture for this project? If yes, define a strategy wrt to your current project.

Company specific

Tuesday, March 28, 2023 12:25 PM

Company: Google
Role: Data Scientist

1. Why do you use feature selection?
2. What is the effect on the coefficients of logistic regression if two 3. predictors are highly correlated?
3. What are the confidence intervals of the coefficients?
4. What's the difference between Gaussian Mixture Model and K-Means?
5. How do you pick k for K-Means?
6. How do you know when Gaussian Mixture Model is applicable?
7. Assuming a clustering model's labels are known, how do you evaluate the performance of the model?

Company: Uber
Role: Data Scientist

1. Pick any product or app that you really like and describe how you would improve it.
2. How would you find an anomaly in a distribution ?
3. How would you go about investigating if a certain trend in a distribution is due to an anomaly?
4. How would you estimate the impact Uber has on traffic and driving conditions?
5. What metrics would you consider using to track if Uber's paid advertising strategy to acquire new customers actually works? How would you then approach figuring out an ideal customer acquisition cost?

Company: TCS
Role: Data Scientist

1. Explain about Time series models you have used?
2. SQL Questions - Group by Top 2 Salaries for Employees - use Row num and Partition
3. Pandas find Numeric and Categorical Columns. For Numeric columns in Data frame, find the mean of the entire column and add that mean value to each row of those numeric columns.
4. What is Gradient Descent? What is Learning Rate and Why we need to reduce or increase? Why Global minimum is reached and Why it doesn't improve when increasing the LR after that point?
5. What is Log-Loss and ROC-AUC?
6. What is Multi-collinearity? How will you choose one features if there are 2 highly correlated features? Give Examples with the techniques used.
7. VIF – Variance Inflation Factor – Explain.
8. Do you know to use Amazon SageMaker for MLOPS?
9. Explain your Projects end to end (15-20mins).

Company: Capital One
Role: Data Scientist

1. How would you build a model to predict credit card fraud?
2. How do you handle missing or bad data?
3. How would you derive new features from features that already exist?
4. If you're attempting to predict a customer's gender, and you only have 100 data points, what problems could arise?
5. Suppose you were given two years of transaction history. What features would you use to predict credit risk?
6. Design an AI program for Tic-tac-toe
7. Explain overfitting and what steps you can take to prevent it.
8. Why does SVM need to maximize the margin between support vectors?

Company: Latentview Analytics
Role: Data Scientist
Experience: 2 years

1. What is mean and median
2. Difference between normal and gaussian distribution
3. What is central limit theorem
4. What is null hypothesis
5. What is confidence interval
6. What is covariance and correlation and how will u interpret it.
7. How will you find out the outliers in the dataset and is it always to remove outliers
8. Explain about Machine Learning
9. Explain the algorithm of your choice
10. Different methods of missing values imputation
11. Explain me your ml project
12. How did you handle imbalance dataset
13. What is stratified samplings

14. Difference between standard scalar and normal scalar
15. Different type of visualization in DI project
16. What architecture have you used
17. Why have u not used RNN in your nlp project
18. Why we don't prefer CNN in nlp based project
19. What is exploding gradient and vanishing gradient and how to rectify it
20. Difference between LSTM and GRU
21. What is precision and recall
22. What is auc metic
23. What if your precision and recall are same
24. What is Bias Variance Trade Off?

Company: Verizon
Role: Data Scientist

1. How many cars are there in Chennai? How do u structurally approach coming up with that number?
2. Multiple Linear Regression?
3. OLS vs MLE?
4. R2 vs Adjusted R2? During Model Development which one do we consider?
5. Lift chart, drift chart
6. Sigmoid Function in Logistic regression
7. ROC what is it? AUC and Differentiation?
8. Linear Regression from Multiple Linear Regression
9. P-Value what is it and its significance? What does P in P-Value stand for? What is Hypothesis Testing? Null hypothesis vs Alternate Hypothesis?
10. Bias Variance Trade off?
11. Over fitting vs Underfitting in Machine learning?
12. Estimation of Multiple Linear Regression
13. Forecasting vs Prediction difference? Regression vs Time Series?
14. p,d,q values in ARIMA models

Company: Fractal
Role: Data Scientist

1. Difference between array and list
2. Map function
3. Scenario, if coupon distributed randomly to customers of swiggy, how to check there buying behavior. Use segmenting customers .Compare customers who got coupon and who did not
4. Which is faster dictionary or list for look up
5. How to merge two arrays
6. How much time svm takes to complete if 1 iteration takes 10sec for 1st class. And there are 4 classes.
7. Kernals in svm, there difference

Company name: Infosys
Role: Data scientist

1. curse of dimensionality? How would you handle it?
2. How to find the multi collinearity in the data set
3. Explain the difference ways to treat multi collinearity!
4. How you decide which feature to keep and which feature to eliminate after performing multi collinearity test?
5. Explain logistic regression
6. we have sigmoid function which gives us the probability between 0-1 then what is the need of logloss in logistic regression?
7. P value and its significance in statistical testing?
8. How do you split the time series data and evaluation metrics for time series data
9. How did you deploy your model in production? How often do you retrain it?

Company: Wipro
Role: Data Scientist

1. Difference between WHERE and HAVING in SQL
2. Basics of Logistics Regression
3. How do you treat outliers ?
4. Explain confusion matrix ?
5. Explain PCA (Wanted me to explain the co-variance matrix and eigen vectors and values and the mathematical expression and mathematical derivation for co-variance matrix)
6. How do you cut a cake into 8 equal parts using only 3 straight cuts ?
7. Explain kmeans clustering
8. How is KNN different from k-means clustering?
9. What would be your strategy to handle a situation indicating an imbalanced dataset?
10. Stock market prediction: You would like to predict whether or not a certain company will declare bankruptcy within the next 7 days (by training on data of similar

companies that had previously been at risk of bankruptcy). Would you treat this as a classification or a regression problem?

Company: Accenture
Role: Data Scientist

1. What is difference between K-NN and K-Means clustering?
2. How to handle missing data? What imputation techniques can be used?
3. Explain topic modelling in NLP and various methods in performing topic modeling.
4. Explain how you would find and tackle an outlier in the dataset.
5. Follow up: What about inlier?
6. Explain back propagation in few words and its variants?
7. Is interpretability important for machine learning model? If so, ways to achieve interpretability for a machine learning models?
8. Is interpretability important for machine learning model? If so, ways to achieve interpretability for a machine learning models?
9. How would you design a data science pipeline?
10. Explain bias - variance trade off. How does this affect the model?
11. What does a statistical test do?
12. How to determine if a coin is biased? Hint: Hypothesis testing

Company: Tiger Analytics
Role: Senior Analyst

1. What is deep learning, and how does it contrast with other machine learning algorithms?
2. When should you use classification over regression?
3. Using Python how do you find Rank, linear and tensor equations for an given array of elements? Explain your approach.
4. What exactly do you know about Bias-Variance decomposition? 5.What is the best recommendation technique you have learnt and what type of recommendation technique helps to predict ratings? 6.How can you assess a good logistic model?
5. How to you read the text from an image? Explain?
6. What are all the options to convert speech to text? Explain and name few available tools to implement the same?

Company Name : Tata IQ
Role: Data Analyst

1. Why data science as a career?
2. Stats:
3. What is p value?
4. What is histograms?
5. What is confidence interval?
6. You are a Sr data analyst at a new Online Cab booking Startups
7. How you will do data collection and how you will leverage the data to give useful insights to the Company?
8. Guestimate: No Of cabs booking per day in Ranchi
9. You are product head manager(not remember exactly) at a NBFC which gives a Secured loans what factors will you consider giving loan to ?
10. Inventory Database based on that have to do basic pandas/sql query? Joins / merge to get avg sales, its chart?
11. You have a list of 3 numbers return the min diff. Can use any python/sql
12. What is Big Data?

Role: Junior Data Scientist

1. Explain the architecture of CNN
2. If we put a 3×3 filter over 6×6 image what will be the size of the output image
3. What will you do to reduce overfitting In deep learning models
4. Can you write a program for inverted star program in python
5. Write a program to create a dataframe and remove elements from it
6. I have 2 guns with 6 holes in each, and I load a single bullet In each gun, what is the probability that if I fire the guns simultaneously atleast 1 gun will fire (atleast means one or more than one)
7. There are 2 groups g1 and g2, g1 will ask g2 members to give them 1 member so thay they both will be equal in number, g2 will ask g1 members to give them 1 member so thay they will be double of g1, how many members are there in the groups (I'm not sure of this question as I tried to solve but didnt get correct answer)

Company: Mindtree
Role: Data Scientist

1. What is central tendency
2. Which central tendency method is used If there exists any outliers
3. Central limit theorem

4. Chi-Square test
5. A/B testing
6. Difference between Z and t distribution (Linked to A/B testing)
7. Outlier treatment method
8. ANOVA test
9. Cross validation
10. How will you work in a machine learning project if there is a huge imbalance in the data
11. Formula of sigmoid function
12. Can we use sigmoid function in case of multiple classification
13. What is Area under the curve
14. Which metric is used to split a node in Decision Tree
15. What is ensemble learning
16. 3 situation based questions

Company: Genpact
Role: Data Scientist

1. Why do we select validation data other than test data?
2. Difference between linear logistic regression?
3. Why do we take such a complex cost function for logistic?
4. Difference between random forest and decision tree?
5. How would you decide when to stop splitting the tree?
6. Measures of central tendency
7. What is the requirement of k means algorithm
8. Which clustering technique uses combining of clusters
9. Which is the oldest probability distribution
10. What all values does a random variable can take
11. Types of random variables
12. Normality of residuals

Company: Ford
Role: Data Scientist

1. How would you check if the model is suffering from multi Collinearity?
2. What is transfer learning? Steps you would take to perform transfer learning.
3. Why is CNN architecture suitable for image classification? Not an RNN?
4. What are the approaches for solving class imbalance problem?
5. When sampling what types of biases can be inflected? How to control the biases?
6. Explain concepts of epoch, batch, iteration in machine learning.
7. What type of performance metrics would you choose to evaluate the different classification models and why?
8. What are some of the types of activation functions and specifically when to use them?
9. What are the conditions that should be satisfied for a time series to be stationary?
10. What is the difference between Batch and Stochastic Gradient Descent?
11. What is difference between K-NN and K-Means clustering?

Company: Quantiphi
Role: Machine Learning Engineer

1. What happens when neural nets are too small? What happens when they are large enough?
2. Why do we need pooling layer in CNN? Common pooling methods?
3. Are ensemble models better than individual models? Why/why - not?
4. Use Case - Consider you are working for pen manufacturing company. How would you help sales team with leads using Data analysis?
5. Assume you were given access to a website google analytics data.
6. In order to increase conversions, how do you perform A/B testing to identify best page design.
7. How is random forest different from Gradient boosting algorithm, given both are tree-based algorithm?
8. Describe steps involved in creating a neural network?
9. In brief, how would you perform the task of sentiment analysis?

Company: TheMathCompany
Role: Analyst (Data Science)

1. Central limit theorem
2. Hypotheses testing
3. P value
4. T-test
5. Assumptions of linear regression.
6. Correlation and covariance.
7. How to identify & treat outliers and missing values.

8. Explain Box and whisker plot.
9. Explain any unsupervised learning algorithm.
10. Explain Random forest.
11. Business and technical questions related to your project.
12. Explain any scope of improvement in your project.
13. Questions based on case studies.
14. Write SQL query to find employee with highest salary in each department.
15. Write SQL query to find unique email domain name & their respective count
16. Solve question (17) using Python.

Rounds:

1. Technical Test (Python, SQL, Statistics) (Coding+MCQ) (90 min).
2. Telephonic interview (10 min).
3. Technical interview (45 min).
4. Fitment interview (25 min).
5. HR interview (30 min).

Company: Cognizant
Role: Data Scientist

1. SQL question on inner join and cross join
2. SQL question on group-by
3. Case study question on customer optimization of records for different marketing promotional offers
4. Tuple and list
5. Linear regression
6. Logistic regression steps and process
7. Tell me about your passion for data science? Or What brought you to this field?
8. What is the most common problems you face whilst working on data science projects?
9. Describe the steps to take to forecast quarterly sales trends. What specific models are most appropriate in this case?
10. What is the difference between gradient and slope, differentiation and integration?
11. When to use deep learning instead of machine learning. Advantages, Disadvantages of using deep learning?
12. What are vanishing and exploding gradients in neural networks?

Company: Husqvarna Group
Role: Data Scientist

1. Telecom Customer Churn Prediction. Explain the project end to end?
2. Data Pre-Processing Steps used.
3. Sales forecasting how is it done using Statistical vs DL models - Efficiency.
4. Logistic Regression - How much percent of Customer has churned and how much have not churned?
5. What are the Evaluation Metric parameters for testing Logistic Regression?
6. What packages in Python can be used for ML? Why do we prefer one over another?
7. Numpy vs Pandas basic difference.
8. Feature on which this Imputation was done, and which method did we use there?
9. Tuple vs Dictionary. Where do we use them?
10. What is NER - Named Entity Recognition?

Company: Deloitte
Role: Data Scientist

1. Conditional Probability
2. Can Linear Regression be used for Classification? If Yes, why if No why?
3. Hypothesis Testing. Null and Alternate hypothesis
4. Derivation of Formula for Linear and logistic Regression
5. Why use Decision Trees?
6. PCA Advantages and Disadvantages?
7. What is Naive Bayes Theorem? Multinomial, Bernoulli, Gaussian Naive Bayes.
8. Central Limit Theorem?
9. Scenario based question on when to use which ML model?
10. Over Sampling and Under Sampling
11. Over Fitting and Under Fitting
12. Core Concepts behind Each ML model mentioned in my Resume.
13. Genie Index Vs Entropy
14. how to deal with imbalance data in classification modelling?

Company: Wipro
Role: Data Scientist

1. What is a Python Package, and Have you created your own Python Package?
2. Explain about Time series models you have used?
3. SQL Questions - Group by Top 2 Salaries for Employees - use Row num and Partition
4. Pandas find Numeric and Categorical Columns. For Numeric columns in Data frame, find the mean of the entire column and add that mean value to each row of those numeric columns.
5. What is Gradient Descent? What is Learning Rate and Why we need to reduce or increase? Why Global minimum is reached and Why it doesn't improve when increasing the LR after that point?
6. Two Logistic Regression Models - Which one will you choose - One is trained on 70% and other on 80% data. Accuracy is almost same.
7. What is Log-Loss and ROC-AUC?
8. Do you know to use Amazon SageMaker for MLOPS?
9. Explain your Projects end to end (15-20mins).

Company: Infosys
Role: Data Scientist

1. Measures of central tendency
2. What is the requirement of k means algorithm
3. Which clustering technique uses combining of clusters
4. Which is the oldest probability distribution
5. What all values does a random variable can take
6. Types of random variables
7. Normality of residuals
8. Probability questions
9. Sensitivity and specificity etc.
10. Explain bias - variance trade off. How does this affect the model?
11. What is multi collinearity? How to identify and remove it.

Company: Tiger Analytics
Role: Data Scientist

1. What are the projects done by you.
2. Suppose there is a client who wants to know if giving discounts is beneficial or not. How would you approach this problem?
3. The same client want to know how much discount he should give in the next month for maximum profits.
4. Can you have a modeling approach to say in last year what mistakes client did in giving discounts. Meaning if they should have have a different discount and increased sales.
5. What feature engineering techniques you used in past projects.
6. What models you used and selected the final model.

Company: Genpact
Role: Data Scientist

1. What makes you feel that you would be suitable for this role, since you come from a different background?
2. What is an imbalanced data set??
3. What are the factors you will consider in order to predict the population of a city in the future?
4. Basic statistics questions?
5. What are the approaches for treating the missing values?
6. Evaluation metrics for Classification?
7. Bagging vs Boosting with examples
8. Handling of imbalanced datasets
9. What are your career aspirations?
10. What's the graph of y = |x|-2
11. Estimate on no. Of petrol cars in Delhi
12. Case study on opening a retail store
13. Order of execution of SQL

Company: Ericsson
Role: Data Scientist

1. Round No: 1st Round
2. How to reverse a linked list
3. Give a logistic regression model in production, how would you find out the coefficients of different input features.
4. What is the p- value in OLS regression
5. What's the reason for high bias or variance
6. Which models are generally high biased or high variance
7. Write code to find the 8 highest value in the DataFrame
8. What's difference between array and list

9. Whats the difference between Gradient boosting and Xgboost
10. Is XOR data linearly separable
11. How do we classify XOR data using logistic regression
12. Some questions from my previous projects
13. Given a sand timer of 4 and 7 mins how would you calculate 10 mins duration.
14. What's the angle between hour and minute hand in clock as 3:15

Company: FISERVE
Role: Data Scientist

1. How would you check if the model is suffering from multi Collinearity?
2. What is transfer learning? Steps you would take to perform transfer learning.
3. Why is CNN architecture suitable for image classification? Not an RNN?
4. What are the approaches for solving class imbalance problem?
5. When sampling what types of biases can be inflected? How to control the biases?
6. Explain concepts of epoch, batch, iteration in machine learning.
7. What type of performance metrics would you choose to evaluate the different classification models and why?
8. What are some of the types of activation functions and specifically when to use them?
9. What is the difference between Batch and Stochastic Gradient Descent?
10. What is difference between K-NN and K-Means clustering?
11. How to handle missing data? What imputation techniques can be used?

Company: Landmark group
Role: Data Scientist

1. Use Case - Consider you are working for pen manufacturing company. How would you help sales team with leads using Data analysis?
2. Interviewers ask about scenarios or use-case based questions to know interviewee thought process and problem-solving skills.
3. Assume you were given access to a website google analytics data.
4. In order to increase conversions, how do you perform A/B testing to identify best page design.
5. How is random forest different from Gradient boosting algorithm, given both are tree-based algorithm?
6. Describe steps involved in creating a neural network?
7. LSTM solves the vanishing gradient problem, that RNN primarily have. How?
8. In brief, how would you perform the task of sentiment analysis?

Company: Axtria

1. RNN, NN and CNN difference.
2. Supervised, unsupervised and reinforcement learning with there algo example.
3. Difference between ai, ml and dl
4. How u do dimentionality reduction.
5. What is Multicollinearity
6. Parameters of random forest
7. Parameters of deep learning algos
8. Different feature selection methods
9. Confusion matrix

Company: Bridgei2i
Role: Senior Analytics Consultant

1. What is the difference between Cluster and Systematic Sampling?
2. Differentiate between a multi-label classification problem and a multi-class classification problem.
3. How can you iterate over a list and also retrieve element indices at the same time?
4. What is Regularization and what kind of problems does regularization solve?
5. If the training loss of your model is high and almost equal to the validation loss, what does it mean? What should you do?
6. Explain evaluation protocols for testing your models? Compare hold-out vs k-fold cross validation vs iterated k-fold cross-validation methods of testing.
7. Can you cite some examples where a false positive is important than a false negative?
8. What is the advantage of performing dimensionality reduction before fitting an SVM?
9. How will you find the correlation between a categorical variable and a continuous variable ?
10. How will you calculate the accuracy of a model using a confusion matrix?
11. You are given a dataset with 1500 observations and 15 features. How many observations you will select in each decision tree in a random forest?
12. Given that you let the models run long enough, will all gradient descent algorithms lead to the same model when working with Logistic or Linear regression problems?
13. What do you understand by statistical power of sensitivity and how do you calculate it?
14. What is pruning, entropy and information gain in decision tree algorithm?
15. What are the types of biases that can occur during sampling?

Company: Prodapt Solutions
Role: Data Scientist

1. Telecom Customer Churn Prediction. Explain the project end to end?
2. Data Pre-Processing Steps used.
3. Sales forecasting how is it done using Statistical vs DL models - Efficiency.
4. Logistic Regression - How much percent of Customer has churned and how much have not churned?
5. What are the Evaluation Metric parameters for testing Logistic Regression?
6. What packages in Python can be used for ML? Why do we prefer one over another?
7. Numpy vs Pandas basic difference.
8. Feature on which this Imputation was done, and which method did we use there?
9. Tuple vs Dictionary. Where do we use them?
10. What is NER - Named Entity Recognition?

Company: Landmark group
Role: Data Scientist

1. SQL question on inner join and cross join
2. SQL question on group-by
3. Case study question on customer optimization of records for different marketing promotional offers
4. Tuple and list
5. Linear regression
6. Logistic regression steps and process
7. Tell me about your passion for data science? Or What brought you to this field?
8. What is the most common problems you face whilst working on data science projects?
9. Describe the steps to take to forecast quarterly sales trends. What specific models are most appropriate in this case?
10. What is the difference between gradient and slope, differentiation and integration?
11. When to use deep learning instead of machine learning. Advantages, Disadvantages of using deep learning?
12. What are vanishing and exploding gradients in neural networks?
13. What happens when neural nets are too small? What happens when they are large enough?
14. Why do we need pooling layer in CNN? Common pooling methods?
15. Are ensemble models better than individual models? Why/why - not?

Company: Mindtree
Role: Data Scientist

1. What is central tendency
2. Which central tendency method is used If there exists any outliers
3. Central limit theorem
4. Chi-Square test
5. A/B testing
6. Difference between Z and t distribution (Linked to A/B testing)
7. Outlier treatment method
8. ANOVA test
9. Cross validation
10. How will you work in a machine learning project if there is a huge imbalance in the data
11. Formula of sigmoid function
12. Can we use sigmoid function in case of multiple classification (I said no)
13. Then which function is used
14. What is Area under the curve
15. Which metric is used to split a node in Decision Tree
16. What is ensemble learning
17. 3 situation based questions

Company: CodeBase Solutions
Role: Data Scientist

1. What are the ML techniques you've used in projects?
2. Very first question was PCA? Why use PCA?
3. Types of Clustering techniques (Not algorithms)? Which Clustering techniques will you use in which Scenario - example with a Program?
4. OCR - What type of OCR did you use in your project - Graphical or Non - Graphical?
5. OCR - What is a Noise? What types of noise will you face when performing OCR? Handwritten can give more than 70% accuracy when I wrote in 2012 but you're saying 40%.
6. Logistic Regression vs Linear Regression with a real-life example - explain?
7. Is Decision tree Binary or multiple why use them?
8. Do you know Map Reduce and ETL concepts?
9. What is a Dictionary or Corpus in NLP and how do you build it?
10. How do you basically build a Dictionary, Semantic Engine, Processing Engine in a NLP project, where does all the Synonyms (Thesaurus words go).

11. What are the Types of Forecasting? What are the ML and DL models for forecasting (He said Fast-forwarding models as example) other than Statistical (ARIMA) models you've used in your projects?
12. What is a Neural Network? Types of Neural Networks you know?
13. Write a Decision Tree model with a Python Program.
14. How do you build an AZURE ML model? What are all the Azure products you've used? I said Azure ML Studio.
15. Cibil score is an example for Fuzzy model and not a Classification model.
16. What is an outlier give a real life example? how do you find them and eliminate them? I gave an example of calculating Average salary of an IT employee.

Company: Deloitte
Role: Data Scientist

1. G values, P values, T values
2. Conditional Probability
3. Central Values of Tendency
4. Can Linear Regression be used for Classification? If Yes, why if No why?
5. Hypothesis Testing. Null and Alternate hypothesis
6. Derivation of Formula for Linear and logistic Regression
7. Where to start a Decision Tree. Why use Decision Trees?
8. PCA Advantages and Disadvantages?
9. Why Bayes theorem? DB Bayes and Naïve Bayes Theorem?
10. Central Limit Theorem?
11. R packages in and out? For us it's Python Packages in and out.
12. Scenario based question on when to use which ML model?
13. Over Sampling and Under Sampling
14. Over Fitting and Under Fitting
15. Core Concepts behind Each ML model.
16. Genie Index Vs Entropy
17. how to deal with imbalance data in classification modelling? SMOTHE techniques

Company: Verizon
Role : Data Scientist

1. How many cars are there in Chennai? How do u structurally approach coming up with that number?
2. Multiple Linear Regression?
3. OLS vs MLE?
4. R2 vs Adjusted R2? During Model Development which one do we consider?
5. Lift chart, drift chart
6. Sigmoid Function in Logistic regression
7. ROC what is it? AUC and Differentiation?
8. Linear Regression from Multiple Linear Regression
9. P-Value what is it and its significance? What does P in P-Value stand for? What is Hypothesis Testing? Null hypothesis vs Alternate Hypothesis?
10. Bias Variance Trade off?
11. Over fitting vs Underfitting in Machine learning?
12. Estimation of Multiple Linear Regression
13. Forecasting vs Prediction difference? Regression vs Time Series?
14. p,d,q values in ARIMA models
15. What will happen if d=0
16. What is the meaning of p,d,q values?
17. Is your data for Forecasting Uni or multi-dimensional?
18. How to find the nose to start with in a Decision tree.
19. TYPES of Decision trees - CART vs C4.5 vs ID3
20. Genie index vs entropy
21. Linear vs Logistic Regression
22. Decision Trees vs Random Forests
23. Questions on liner regression, how it works and all
24. Asked to write some SQL queries
25. Asked about past work experience
26. Some questions on inferential statistics (hypothesis testing, sampling techniques)
27. Some questions on table (how to filter, how to add calculated fields etc)
28. Why do u use Licensed Platform when other Open source packages are available?
29. What certification Have u done?
30. What is a Confidence Interval?
31. What are Outliers? How to Detect Outliers?
32. How to Handle Outliers?

Company: L&T Financial Services
Role: Data Scientist

1. Explain your Projects
2. Assumptions in Multiple linear regression

3. Decision tree algorithm
4. Gini index
5. Entropy
6. Formulas of gini and entropy
7. Random forest algorithm
8. XGBoost Algorithm
9. Central Limit theorem
10. R2
11. Adj R2
12. VIF
13. Different Methods to measure Accuracy
14. Explain Bagging and Boosting
15. Difference Between Bagging and Boosting
16. Various Ensemble techniques
17. P value and it's significance
18. F1 Score
19. Type 1 and Type II error
20. Logical questions for Type 1 and Type II error
21. Logical questions for Null and alternate Hypothesis

Role: Data Scientist

1. Decorators in Python

- Live Example

2. Generators in Python

- Live Example

3. SQL Questions
   3.1 Group by Top 2 Salaries for Employees
   3.2 use Row num and Partition

4. Pandas find Numeric and Categorical Columns.
   4.1 For Numeric columns, find the mean of the entire column and add that value to each row of the column.

5. What is Gradient Descent?
   5.1 What is Learning Rate and Why is it reduce sometimes

6. Two Logistic Regression Models - Which one will you choose - One is trained on 70% and other on 80% data. Accuracy is almost same.
7. What is LogLoss ?

8. Explain your Projects end to end.(15-20mins)

Role: Data Science Intern

1. Tell me about your journey as a Data Science aspirant
2. What was the one challenging project or task that you did in this domain and why was it challenging?
3. What model did you use for that? I replied Random Forests
4. What is Random Forest and how is it used?
5. How are Random Forest different from Decision Trees and what problems do they solve that decision trees can't?
6. Multi class Classification and which metric is preferred for it
7. Given a banking scenario to predict Loan Defaulters, which metric will you use?
8. How will you handle the class imbalance in this case?

Data Science Interview Questions

1. Naive bayes assumptions
2. What are the approaches for solving class imbalance problem?
3. When sampling what types of biases can be inflected? How to control the biases?
4. GRU is faster compared to LSTM. Why?
5. What is difference between K-NN and K-Means clustering ?
6. How to determine if a coin is biased ? Hint: Hypothesis testing
7. How will u present the statistical inference of a particular numerical column?
8. How would you design a data science pipeline ?
9. Explain back propagation in few words and it's variants?
10. Explain topic modeling in NLP and various methods in performing topic modeling.

Company: Myntra
Role: Data Analyst

1. Introduce yourself.
2. One complex sql query- 2 table are there, Table1(cust_id,Name) Table2(cust_id,Transaction_amt)
3. Write a query to return the name of customers with 8th highest lifetime purchase.
4. Achieve the same using python.

ML questions:

1. What's the problem in having multi collinearity in data set.
2. If there is business requirement to keep two corelated features in model, what would you do.
3. How would you deal with feature of 4 categories and 20% null values

Company: Nira Finance
Role: Data Scientist

1. Asked to explain my project.
2. Have you not done any classification problem as your Resume only mentions regression tasks.
3. Explain the working of Gradient boosting.
4. Difference between boosting and bagging.
5. What would you do when output is imbalanced.
6. What is more preferred over sampling or under sampling.
7. In what case under sampling a non-harmful approach.
8. How would you measure the performance of models built on imbalanced dataset.
9. What is the meaning of precision and recall.
10. Tell me about a task you did and are very proud of.
11. Do you have any questions for me
12. Post interview I was given a ML assignment to solve and submit within next 2 weeks.

Company: Myntra
Role: Data Analyst
Round type: Use case Round

Problem Statement:

Given 2 teams of Myntra namely:

1. Finance Team: They focus to take decisions which are Money driven.

2. Customer Experience Team : They focus to improve the Customer Experience with Myntra.

Whenever Customer places a refund request Myntra can process it in 2 different ways:

1. Directly accept the Return request.

2. Put the request on hold and verify the product for damages or manhandling by customer. Only if the products are found to be in proper state, accept the return.

Now, there is a conflict of opinion between these two teams.
Finance Team likes the 2nd option as it minimize the chances of loss.

But Customer Experience teams likes the 1st option as their main aim is to improve Customer Experience.

Now suppose you are part of the Customer Experience team. How would you convince the Finance team to follow the 1st step.

What kind of Data you would be looking for solving this task.

Is there any need for model building for this use case.

Company: Ericsson
Role: Data Scientist

Round No: 1st Round

1. How to reverse a linked list
2. Give a logistic regression model in production, how would you find out the coefficients of different input features.
3. What is the p- value in OLS regression
4. What's the reason for high bias or variance
5. Which models are generally high biased or high variance
6. Write code to find the 8 highest value in the DataFrame

7. What's difference between array and list
8. Whats the difference between Gradient boosting and Xgboost
9. Is XOR data linearly separable
10. How do we classify XOR data using logistic regression
11. Some questions from my previous projects
12. Given a sand timer of 4 and 7 mins how would you calculate 10 mins duration.
13. What's the angle between hour and minute hand in clock as 3:15

Company: Legato Health Technologies
Role: MLOps Engineer
Experience: 2-3 years

1. Complete ML technical stack used in project?
2. Different activation function?
3. How do you handle imbalance data ?
4. Difference between sigmoid and softmax ?
5. Explain about optimizers ?
6. Precision-Recall Trade off ?
7. How do you handle False Positives ?
8. Explain LSTM architecture by taking example of 2 sentences and how it will be processed?
9. Decision Tree Parameters?
10. Bagging and boosting ?
11. Explain bagging internals
12. Write a program by taking an url and give a rough code approach how you will pass payload and make a post request?
13. Different modules used in python ?
14. Another coding problem of checking balanced parentheses?

Role: Junior Data Scientist

1. Explain the architecture of CNN
2. If we put a 3×3 filter over 6×6 image what will be the size of the output image
3. What will you do to reduce overfitting In deep learning models
4. Can you write a program for inverted star program in python
5. Write a program to create a dataframe and remove elements from it
6. I have 2 guns with 6 holes in each, and I load a single bullet In each gun, what is the probability that if I fire the guns simultaneously atleast 1 gun will fire (atleast means one or more than one)
7. There are 2 groups g1 and g2, g1 will ask g2 members to give them 1 member so thay they both will be equal in number, g2 will ask g1 members to give them 1 member so thay they will be double of g1, how many members are there in the groups (I'm not sure of this question as I tried to solve but didnt get correct answer)

Data Science Interview Questions:

1. How do check the Normality of a dataset?
2. Difference Between Sigmoid and Softmax functions?
3. Can logistic regression use for more than 2 classes?
4. What are Loss Function and Cost Functions? Explain the key Difference Between them?
5. What is F1 score? How would you use it?
6. In a neural network, what if all the weights are initialized with the same value?
7. Why should we use Batch Normalization?
8. In a CNN, if the input size 5 X 5 and the filter size is 7 X 7, then what would be the size of the output?
9. What do you mean by exploding and vanishing gradients?
10. What are the applications of transfer learning in Deep Learning?
11. Why does a Convolutional Neural Network (CNN) work better with image data?
12. What is the Central Limit Theorem and why is it important?
13. What is the difference between type I vs type II error?
14. Tell me the difference between an inner join, left join/right join, and union.
15. Explain the 80/20 rule, and tell me about its importance in model validation.
16. What is one way that you would handle an imbalanced data set that’s being used for prediction (i.e., vastly more negative classes than positive classes)?
17. Is it better to spend five days developing a 90-percent accurate solution or 10 days for 100-percent accuracy?
18. Most common characteristics used in descriptive statistics?
19. What do you mean by degree of freedom?
20. Why is the t-value same for 90% two tail and 95% one tail test?
21. What does it mean if a model is heteroscedastic? what about homoscedastic?
22. You roll a biased coin (p(head)=0.8) five times. What’s the probability of getting three or more heads?
23. What does interpolation and extrapolation mean? Which is generally more accurate?
24. What the aim of conducting A/B Testing?
25. Explain p-value.

26. Explain how a ROC curve works?
27. What is pruning in Decision Tree?
28. How will you define the number of clusters in a clustering algorithm?
29. When to use Precision and when to use Recall?
30. What are the assumptions required for linear regression? What if some of these assumptions are violated?
31. How are covariance and correlation different from one another?
32. How can we relate standard deviation and variance?
33. Explain the phrase "Curse of Dimensionality".
34. What does the term Variance Inflation Factor mean?
35. What is the significance of Gamma and Regularization in SVM?
36. How will you calculate the Sensitivity of machine learning models?
37. What do you mean by cluster sampling and systematic sampling?
38. Explain Eigenvectors and Eigenvalues.
39. Explain Gradient Descent.
40. How does Backpropagation work? Also, it states its various variants.
41. What do you know about Autoencoders?
42. What is Dropout in Neural Networks?
43. What is the difference between Batch and Stochastic Gradient Descent?
44. What are the different kinds of Ensemble learning?
45. What is entropy, information gain and gini index in decision tree classifier and regression?
46. What is central tendency
47. Which central tendency method is used If there exists any outliers
48. Central limit theorem
49. Chi-Square test
50. A/B testing
51. Difference between Z and t distribution (Linked to A/B testing)
52. Outlier treatment method
53. ANOVA test
54. Cross validation
55. How will you work in a machine learning project if there is a huge imbalance in the data
56. Formula of sigmoid function
57. Can we use sigmoid function in case of multiple classification (I said no)
58. Then which function is used
59. What is Area under the curve
60. Which metric is used to split a node in Decision Tree
61. What is ensemble learning
62. Three situation based questions

Company: Legato Health Technologies
Role: MLOps Engineer

1. Complete ML technical stack used in project?
2. Different activation function?
3. How do you handle imbalance data ?
4. Difference between sigmoid and softmax ?
5. Explain about optimisers ?
6. Precision-Recall Trade off ?
7. How do you handle False Positives ?
8. Explain LSTM architecture by taking example of 2 sentences and how it will be processed?
9. Decision Tree Parameters?
10. Bagging and boosting ?
11. Explain bagging internals
12. Write a program by taking an url and give a rough code approach how you will pass payload and make a post request?
13. Different modules used in python ?
14. Another coding problem of checking balanced parentheses?

Company: Cerence
Role: NLU Developer

Question1 :
Write a function that take two strings as inputs and return true if they are anagrams of each other and false otherwise
e.g.
(hello, hlleo) --> true
(hello, helo) --> false

Question 2 :
Write a function that take an array of strings "A" and an integer "n",
that return the list of all strings of length "n" from the array "A" that can be constructed

as the concatenation of two strings from the same array "A"
e.g.
A = [dog, tail, sky, or, hotdog, tailor, hot] and n=6
output should be "hotdog" and "tailor"

Question 3 :
Given an array "arr" of numbers and a starting number "x",
Find "x" such that the running sums of "x" and the elements of the array "arr" are never lower than 1.
e.g.
arr = [-2, 3, 1, -5].
The running sums will be x-2, x-2+3, x-2+3+1 and x-2+3+1-5.
So, the output should be 4.

Company: GEOTAB

Python :
1. Is python a language that follows pass by value, or pass by reference or pass by object reference
2. What are lambda functions and how to use them
3. Difference between mutable and immutable objects with example.
4. What are Python decorators? Why do we use them

SQL :
1. What is the difference between Inner join and left inner join ?
2. What are window functions ?
3. What is the use of groupby ?

SQL Round
3 tables given as below:

TRIPS
trip_id
vehicle_id
start_time
stop_time

VEHICLE_MAKE
vehicle_id
make_id

MAKES
make_id
make_name

There is a table which contains vehicle trips. Trips are not necessarily in order.

There is a table which contains vehicle makes. Makes are not necessarily known.

PROBLEM: Write SQL code that provides the number of trips that started on September 1st, 2020 for each vehicle with a KNOWN make.
Order the results by the trip count.

op
```
 vehicle_id | trip_count
    4 |    2
    1 |    1
    2 |    1
```

Role: MLOps Engineer

1st round:-
Introduction
Current NLP architecture used in my project
How will you identify Data Drift? Once identified how would you automate the handling of Data Drift
Data Pipeline used
Fasttext word embedding vs word2vec
When should we use Tf-IDF and when predictive based word embedding will be advantageous over Tf-IDF
Metrics used to validate our model
In MongoDB write a query to find employee names from a collection

In Python write a program to separate 0s and 1s from an array- (0,1,0,1,1,0,1,0)

Company: Latentview.

Initial they had asked for the explaining the project which I had done. I explained the Customer prediction case . Then I was asked with python questions by sharing my screen.

1. How do you handle the correlated variables without removing them
2. Explain the SMOTE, ADAYSN technique
3. What is stratified sampling technique
4. Explain the working of random forest and xgboost
5. How do you optimise the Recall of your output
6. What are chisquare and ANOVA test
7. In python they asked for LOC,ILOC, how do you remove duplicate,How to unique values in column,
8. In SQL they asked for the query for having matches between different teams

Company: Myntra
Role: Data Analyst

Introduce yourself.

One complex sql query- 2 table are there, Table1(cust_id,Name) Table2(cust_id,Transaction_amt)
Write a query to return the name of customers with 8th highest lifetime purchase.

Achieve the same using python.

ML questions:

What's the problem in having multi collinearity in data set.
If there is business requirement to keep two corelated features in model, what would you do.
How would you deal with feature of 4 categories and 20% null values.

Company: Enquero Global
Role: Data Scientist

1. Previous job role and responsibilities
2. Problem statement of your project and How do you overcome challenges
3. How do you handle feature which had many categories.
4. When to use precision and recall.
5. What are outliers & how do you handle them
6. Joins, self joins, said me to write sql queries on self joins
7. How good your with python
8. Logic for reverse string.
9. Data collection- how do you collect data and data preprocessing.
10. Focused on EDA part
11. Have you deveployed any project in cloud if you which cloud you had used and how do you do that.
12. How do you interact with domain expertise and business analytics people.
13. How do you replace Missing values for continuous variables.
14. What is your largest data set you have handled till now and tell me size of dataset.
15. Overall Mostly focused on SQL, DATA COLLECTION, EDA, feature engineering and selection.
16. A continuous variable is having missing values, so how will you decide that the missing values should be imputed by mean or median?
17. What is PCA and what each component means? Also, what is the maximum value for number of components?
18. What is test of independence? How do you calculate Chi-square value?
19. When precision is preferred over recall or vice-versa?
20. Advantages and disadvantages of Random forest over Decision Tree?
21. What is the c hyperparameter in SVM algorithm and how it affects bias variance tradeoff?
22. What are the assumptions of linear regression?
23. Difference between Stemming and Lemmatization?
24. Difference between Correlation and Regression?
25. What is p-value and confidence interval?
26. What is multicollinearity and how do you deal with multicollinearity? What is VIF?
27. What is the difference between apply, applymap and map function in python?

Deloitte Interview :

Role: Data Scientist
Candidate Name: Wanted to remain anonymous
ROUND 1 :

Introduction

1. Started with Classification particularly Imbalance , oversampling.
- Which class should i oversample etc.
- Telecom Churn Case Study Questions like Evaluation metric for imbalance data
- What threshold to choose to diving the classes (0.5 in case of balanced else sensitivity / specifivity etc.
- What if i don't use SMOTE() for handling imbalance how should i select the threshold now (messed up by me, roc , auc etc) Ans = Presion - Recall Curve

2. NLP Questions
- Sentiment analysis, preprocessing like (TFID, BOW), Embeddings, stemming, Lemmatization libraries in know : nltk, spacy

3. Regression Preprocessing
- Outlier, missing value imputation, Distribution, dummies, multicollinearity etc
- You have two highly co-related columns which one will you drop? : "Based on Business Problem i will see accordingly.",

4. Naive Bayes Explanation , Drawback of Naive Bayes(couldn't answer drawback of Naive Bayes, 'Assume all are independent', him)
5. Hand Gesture Recognition Techniques (End to End)
6. Resource Timesheet Forecasting . (What is it?? what you do on this?, " Explained with a story based on what i do in TCS".
7. Do you know any Boosting Algorithms : YES
- where have you used?? in Telecom Churn and Healthcare Analytics by AV
8. Gradient Descent (How it works)
9. KNN related. How do we choose value of K ??
10. Statistical Computing:
- Type 1 and Type 2 error
- Alternate name of Type 1 error (couldn't answer alternate name of Type 1 error, 'False +ive, him)
- What is p-Value (Explained with the example of Linear Regression from stats model)
11. Do you have exposure of Time Seires analysis : NO (didn't ask anything and seems fine with him)

Statistics

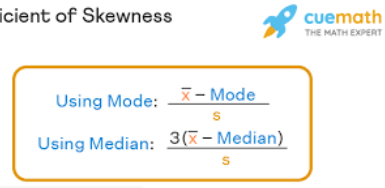
Wednesday, March 15, 2023 8:36 AM

1. Where you have used Hypothesis Testing in your Machine learning Solution.

   Finding correlation between features

2. What kind of statistical tests you have performed in your ML Application
3. What do you understand by P Value? And what is use of it in ML?
4. Which type of error is severe Error, Type 1 or Type 2? And why with example.
5. Where we can use chi square and have used this test anywhere in your application
6. Can we use Chi square with Numerical dataset? If yes, give example. If no, give reason?
7. What do you understand by ANOVA Testing?
8. Give me a scenario where you can use Z test and T test.
9. What do you understand by inferential Statistics?
10. When you are trying to calculate Std Deviation or Variance, why you used N-1 in Denominator? (Hint: Basel Correction)
11. What do you understand by right skewness, Give example?
12. What is difference between Normal distribution and Std Normal Distribution and Uniform Distribution?
13. What is different kind of Probabilistic distributions you heard of?
14. What do you understand by symmetric dataset?
15. In your last project, were you using symmetric data or Asymmetric Data, if it's a symmetric, what kind of EDA you have performed?
16. Can you please tell me formula for skewness?

    Coefficient of Skewness

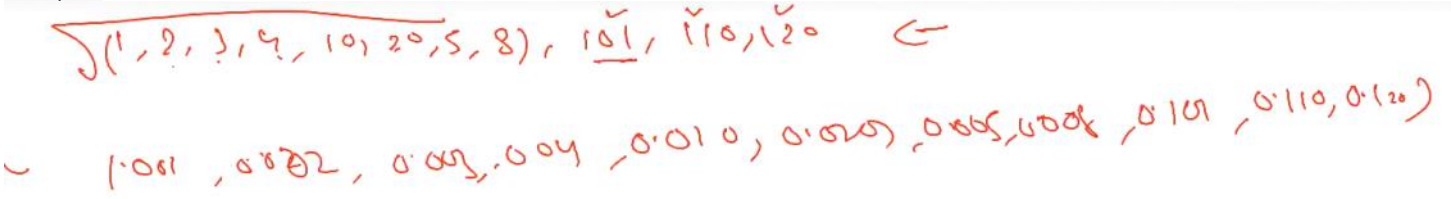
    

    S = standard deviation

    If data is screwed, we can do that following things to convert the data to normal distribution

    - Normalize the data
    - Logarithm scaling of the data
    - Square root of the data points which reduces the variance in the data and the data is scaled to a smaller/lower scale
    - Preferred method than IQR since won't have data loss with this method.

    Example :

    Here there are three point which are far away from the rest of the distribution.
    We could apply IQR and remove those points
    Alternatively, we can apply transformations like square root of the data points so that the scale becomes smaller and variance reduces
    And this way there won't be any data loss

17. Have you applied student T distribution Anywhere?

    - When sample size is smaller than 30, t distribution can be used
    - Use of t distribution : In hypothesis testing, if population is not known to me, but sample and sample size is known and we want to know of the sample falls in a specific critical distribution of the population, then we can use t distribution.

18. What do you understand by statistical analysis of data, Give me scenario where you have used statistical analysis in last projects?

    - Pandas describe function - min, max, etc
    - OLS Ordinary Least Square- method summary, r2, p value covariance etc
    - ROC, AUC
    - TP, FP, TN, FN

19. Can you please tell me criterion to apply binomial distribution, with example?

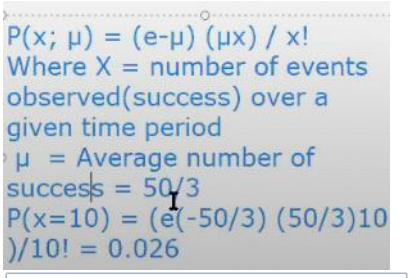
Binomial Distribution can be applied when

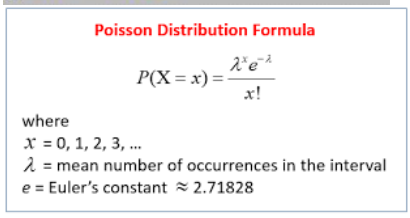
- Two events are independent to each other - coin toss, dice roll
- Number of outcomes is fixed

20. There are 100 people, who are taking this particular 30 days Data science interview preparation course, what is the probability that 10 people will be able to make transition in 1 week? If 50 people were able to make transition in 3 weeks? (Hint: Poisson Distribution)

It is a Poisson distribution because

- In a unit time interval some event is going to happen and we need to find the probability of the event occurring in some timeinterval.





21. Let us suppose I have appeared in 3 interviews, what is the probability that I am able to crack at least 1 interview? [justify]

Probability of success = 1/2 = 0.5
Probability of failure = 1/2 = 0.5

Crack at least 1 interview means I can crack 1, 2 or 3 interviews

Probability of cracking all three interviews = 1
Probability of cracking only 1 interview = 1/2

Probability of cracking at least 1 interview = 1 - (1/2)*(1/2)*(1/2) = 1 - 1/8 = 7/8=0.875

22. Explain Gaussian Distribution in your own way.

23. What do you understand by 1st ,2nd and 3rd Standard Deviation from Mean?

24. What do you understand by variance in data in simple words?

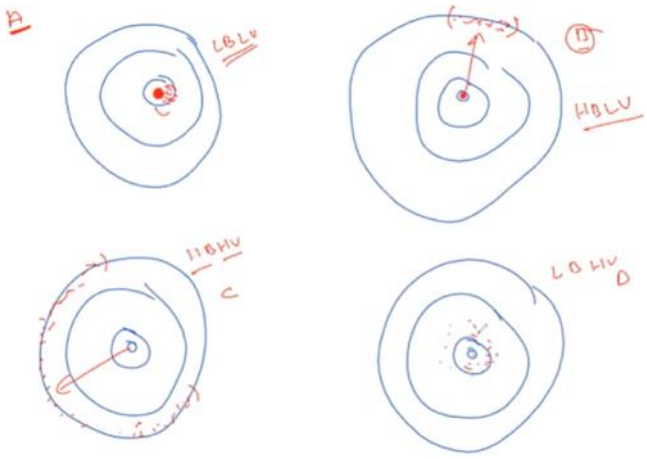
Range of values, give real life example
Life when we drive a car starting from zero till 100 the speed will take many values
This can be called the variance of the values of speed

25. If variance of dataset is too high, in that case How you will be able to handle it or decrease it?

Decrease the scale - standardization, log or square root

26. Explain the relationship between Variance and Bias.

Bias : How far the predicted data is from the expected data/ actual data
Variance : The scatterness of the predicted data



27. Tell me what kind of graph-based approach I will be able to apply to find out standardization of Dataset?
28. What do you understand by Z Value given in Z Table?

Z scores (Z value) is the number of standard deviations a score or a value (x) is away from the mean. In other words, the Z-score measures the dispersion of data. Technically, a Z-score tells you how many standard deviations value (x) is below or above the population mean (μ).

29. Do you know a Standard Normal Distribution Formula?

$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$\mu =$ Mean
$\sigma =$ Standard Deviation
$\pi \approx 3.14159\cdots$
$e \approx 2.71828\cdots$

If we have data on x axis and y axis is plotted with this formulas, the data will look like a bell curve

30. Can you please explain critical region in your way?
31. Have you used AB testing in your project So far? If yes, Explain. If no, Tell me about AB testing.
32. Can we use Alternate hypothesis as a null Hypothesis?
33. Can you please explain confusion matrix for more than 2 variables?
34. Give me an example of False Negative From this interview?
35. What do you understand by Precision, Recall and F1 Score with example?
36. What kind of questions do you ask to your client if they give you dataset?
37. Have you ever done F test on your dataset, if yes, give example. If No, then explain F distribution?
38. What is AUC & ROC Curve? Explain with uses.
39. Who decided in your last project, what will be the accuracy of your model & what was the criterion to make the decision.
40. What do you understand by 1 tail test & 2 tail test? give example.
41. What do you understand by power of a test?
42. How do you set level of significance for your dataset?
43. Have you ever used T table in any of your project so far? If No, then why statistic is important for data scientist? If yes, explain the scenario.
44. Can we productionize statistical model?
45. How frequently do you build the model and test it?
46. What are the testing techniques that you use for model testing, name some of those?
47. What do you understand by sensitivity in dataset? Give example.
48. Let's suppose you are trying to solve classification problem; how do you decide which algorithm to use? Give scenarios.
49. Can we use Logistic regression for classification if my no. of classes are 5?
50. Let's suppose there is company like OLA or UBER who provides service to many customers, then how will they make sure that car availability in particular region and what kind of dataset is required?
51. AI Solution for architecture -- Let's suppose there is agricultural field in diff areas in India, and we know soil & weather condition

is different over India, So I am trying to build system which helps me understanding what kind of treatments I will be able to apply on my crops, which crop I can grow in particular month so I can be able to maximize the benefit from the soil. Then what kind of algorithm you will use whether its ML, DL, Vision? What will be your approach and what kind of solution design you will provide?

52. I have a client, they are facing a problem in terms of maintaining the pipeline for water. So what kind of AI solution you will design to identify the leakage and maintenance?
53. Let's suppose I am building solution for blind people what kind of AI solution you will provide to help them to interact with the system, an Affordable solution?
54. What is difference between R2 and Adjusted R2?
55. Where do you apply Regularization and What kind of regularization you have applied and Why?
56. What do you understand by multicollinearity and homoscedasticity in Dataset?
57. Can you please explain 1 example of Polynomial Regression and how to build model for polynomial regression.
58. There is some client who are intercepting a call like 3,4 or 5 people talking in zoom call. Tell me approach so that we can able to separate the voices of each and every person.(Hint: Speech Diarization)
59. In case of multilinear regression model, let's suppose my number of features are 5. Can you explain me what kind of line it draws? Explain.
60. List no. of algorithms that you know from clustering.
61. Tell me what is evaluation techniques for clustering algorithms. List some of those.
62. Can you please explain random state in train & test split function.
63. Let us suppose client has provided me a data, how will you evaluate that the data is fit for model building?
64. Have you ever worked in your last project from scratch? Or you started working in middle. If you have started working from scratch then how what kind of work were doing. And if you have started from middle then what were your responsibility?
65. What do you understand by machine learning? how will you explain ML to Kids.
66. Let's suppose there is project which I am going to start for client (security & surveillance project). Client requirement is like this- they want to develop a system which can detect any kind of intrusion or unwanted or unclassified entity in region. 1. What kind of solution you will provide to solve this requirement. 2.And what kind of feature you will be able to provide? . Give complete proposal for this solution.

What are a Z test, Chi-Square test, F test, and T-test?

- **T-test:** Hypothesis testing for small sample size.
- **Z-test:** Hypothesis testing for Large sample.
- **Chi-square Test:** determine the difference observed and expected frequencies of certain observations.
- **F test:** Hypotheses of interest are about the differences between population means.

Checklist

Tuesday, March 28, 2023 12:34 PM

My Statistics Checklist before going for a Data Science Interview:

1. Inferential and descriptive Statistics
2. Sample
3. Population
4. Random variables
5. Probability Distribution Function
6. Probability Mass Function
7. Cumulative Distribution Function
8. Expectation and Variance
9. Binomial Distribution
10. Bernoulli Distribution
11. Normal Distribution
12. Z-score
13. Central Limit Theorem
14. Hypothesis Testing
15. Confidence Interval
16. Chi Square Test
17. Anova Test
18. F-Stats

My checklist before going for an SQL round of interview:

1. WHERE , AND, OR, NOT, IN
2. ORDER BY, ASC, DESC
3. IS NULL
4. LIMIT
5. MIN, MAX, COUNT, AVG, SUM
6. LIKE, WILDCARDS
7. IN BETWEEN
8. INNER JOIN
9. LEFT JOIN
10. Subqueries(most important)
11. UNION
12. GROUP BY
13. HAVING
14. LEFT, RIGHT, MID, CONCAT
15. PARTITION BY, OVER
16. LEAD,LAG

17. RANK, DENSE_RANK, PERCENT_RANK
18. ROW_NUMBER, CUME_DIST
19. FIRST_VALUE, LAST_VALUE
20. AS

Cheat Sheet for Formulas

Wednesday, March 29, 2023 1:50 PM

Gini Impurity

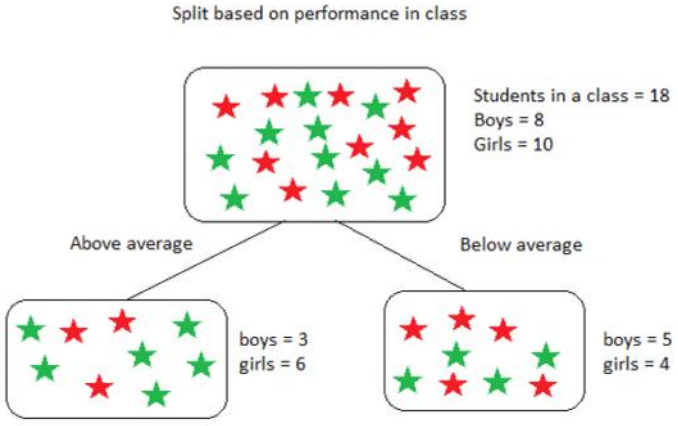
The Gini impurity measure is one of the methods used in decision tree algorithms to decide the optimal split from a root node, and subsequent splits. It is the most popular and the easiest way to split a decision tree and it works only with categorical targets as it only does binary splits.

Gini Impurity is calculated using the formula,

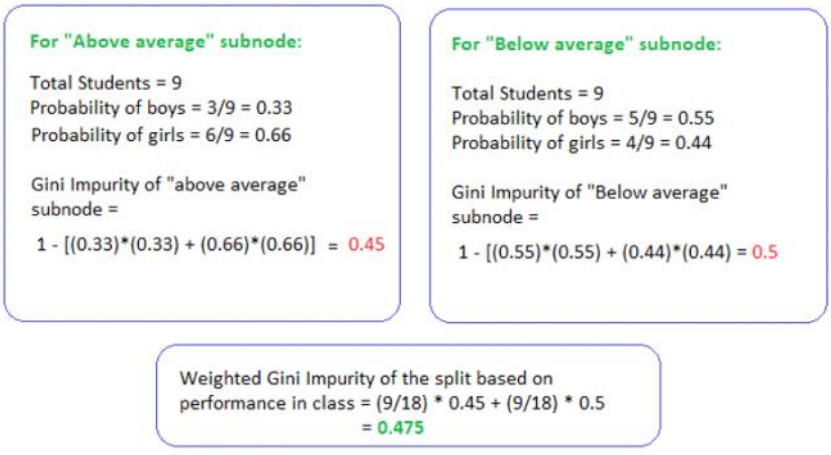
$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Lower the Gini Impurity, higher is the homogeneity of the node. **The Gini Impurity of a pure node(same class) is zero.**

To calculate Gini impurity, let's take an example of a dataset that contains 18 students with 8 boys and 10 girls and split them based on performance as shown below.



The calculation of Gini Impurity of the above would be as follows:



In the above calculation, to find the Weighted Gini Impurity of the split (root node), we have used the probability of students in the sub nodes, which is nothing but 9/18 for both "Above average" and "Below average" nodes as both the sub nodes have equal no of students even though the count of boys and girls in each node varies depending on their performance in class.

Following are the steps to split a decision tree using Gini Impurity:

1. Similar to what we did in entropy/Information gain. For each split, individually calculate the Gini Impurity of each child node
2. Calculate the Gini Impurity of each split as the weighted average Gini Impurity of child nodes
3. Select the split with the lowest value of Gini Impurity
4. Until you achieve homogeneous nodes, repeat steps 1-3

To summarize on Gini Impurity :

- It helps to find out the root node, intermediate nodes and leaf node to develop the decision tree
- It is used by the CART (classification and regression tree) algorithm for classification trees.
- It reaches its minimum (zero) when all cases in the node fall into a single target.

****

Entropy

Entropy is the measures of impurity, disorder or uncertainty in a bunch of examples. Entropy controls how a Decision Tree decides to split the data. The below image shows impurity level of each set.

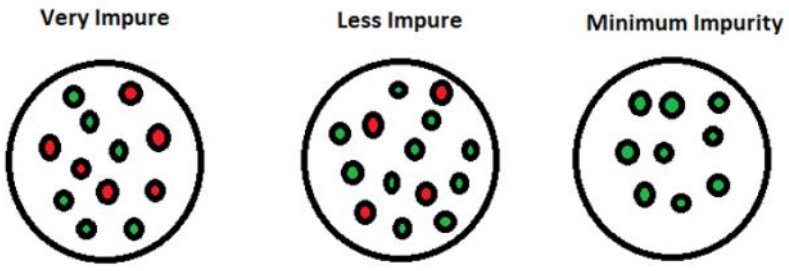


Image by Author

If we have a set of K different values , then we can calculate the entropy using this formula:

$$entropy = -\sum_{i=1}^{k} P(value_i) \cdot \log_2\left(P(value_i)\right)$$

where, P(value*i*) is the probability of getting the *i*th value when randomly selecting one from the set.

For eg, let's take the following image with green and red circles.

Image by Author

In this group, we have 14 circles, out of which 10 are green (10/14) and 4 are red (4/14). Let's find the entropy of this group.

Entropy(Green) = $\log_2(10/14) = -0.4854268272$

Entropy(Red) = $\log_2(4/14) = -1.807354922$

Entropy (Set) = $-(10/14)(-0.4854268272) - (4/14)(-1.807354922) =$ **0.86**

Note:

- The entropy of a group in which all examples belong to the same class will always be 0 as shown below:

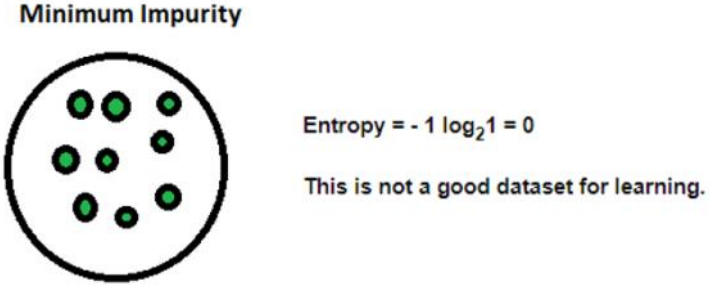
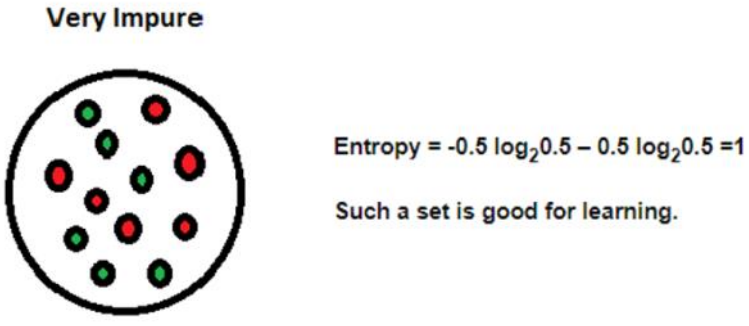


Image by Author

- The entropy of a group with 50% in either class will always be 1 as shown below:



****

Information Gain

Information gain (IG) measures how much "information" a feature gives us about the class.It tells us how important a given attribute of the feature vectors is. **Information gain (IG)** is used to decide the ordering of attributes in the nodes of a decision tree.

Information gain (IG)is calculated as follows:
Information Gain = entropy(parent) – [average entropy(children)]
Let's look at an example to demonstrate how to calculate Information Gain.
Let's say a set of 30 people both Male and female are split according to their age. Each person's age is compared to 30 and they are separated into 2 child groups as shown in the image and their corresponding node's entropy is calculated. The main node is called the Parent node and the 2 sub nodes are called child nodes.

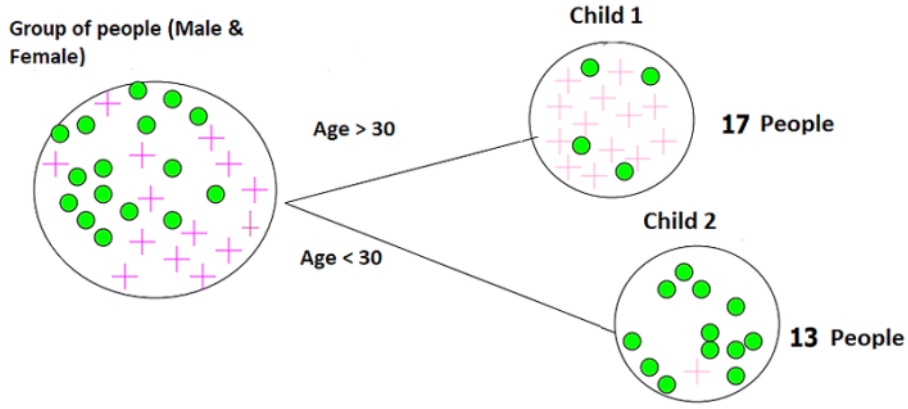


Image by Author
The entropies of parent and child nodes are calculated as shown below. The Information gain is then calculated using the entropy of individual nodes.

parent entropy $-\left(\frac{14}{30}\cdot\log\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log\frac{16}{30}\right)=$ **0.996**

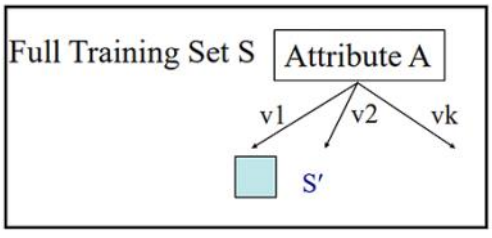
child 1 entropy $-\left(\frac{13}{17}\cdot\log\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log\frac{4}{17}\right)=$ **0.787**

child 2 entropy $-\left(\frac{1}{13}\cdot\log\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log\frac{12}{13}\right)=$ **0.391**

(Weighted) Average Entropy of children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain - 0.996 - 0.615 = 0.38 for this split

The steps that needs to be followed to construct a decision tree using Information gain is shown below:



Step 1:
Choose the attribute with highest Information gain from the set as root node.
Step 2:
Construct child nodes for each value of A
Step 3:
Repeat recursively until the whole tree is built

Image by Author

Entropy and Information Gain are two main concepts that are used when it comes to constructing a decision tree, to determine the nodes and the best way to split.

Variance Inflation Factor (VIF)

bias and variance tradeoff machine learning

ROC and AUC curve

Gradient Descent

Loss regularization

Distances - Manhattan, Euclidean

Interview

Tuesday, September 12, 2023 10:13 AM

1. Types of data you have worked with, domain and also structure wise
2. How to find the multi collinearity in the data set, start with explaining what is multi collinearity
3. Explain the difference ways to treat multi collinearity , why should we be treating them!
4. How to handle missing data? What imputation techniques can be used?
5. What are Outliers? Explain a simple method to Detect Outliers?
6. Consider the data is without labels , how will you approach the problem for anomaly detection?
7. In an AD classifier if you get an accuracy of 96% , why should you not be satisfied with the results?
8. EDA steps that are performance
9. Matrices used for model evaluation
10. Random forest model
11. Explain the concept of GINI Impurity.
12. What are the assumptions of linear regression?
13. Explain the concept of gradient descent
14. Why do we need ROC/ AUC curve for?
15. Sigmoid Function in Logistic regression

16. How is KNN different from k-means clustering?
17. Explain centroid formation technique in K Means algorithm
18. What does the term Variance Inflation Factor mean?
19. How are covariance and correlation different from one another?
20. Statistical analysis what can be done
21. What is low bias? What is high variance? Why do we need Bias and Variance tradeoff

Tools : Python, Spyder 6 GB data , No Viz No cloud

Situational
Care to give and overview of the type of datasets you have worked with and the models you have ventured

1. Consider you have a dataset with second sampling and another set of with minute sampling. And you want to use the whole set combined as features to your model., how would you approach the problem. ; to convert two set of signals with different sampling rate to a common dataset of features ? [Up sampling, down sampling, methods to do so]
2. If you want to remove the noise in the features before giving it to a model , what statistical method will you use. Other methods to discover outliers [Z score, Scatter plots] What are the five statistical measures represented in a boxplot?
3. Consider you have 40 features. What steps would you take before sending these features to the model and why ? [find correlation, feature transformations, explain a feature transformation method in detail]

- [ ] 4. Explain the working principle of PCA, including the steps involved / pre requisites to carry out a PCA. How would you finalize on the ideal number of PCAs you want for your features.
    1. Standardization is required or not ?
    2. How would you do Fit and transform in the training and testing set
    3. Why only transform for testing
- [ ] 5. Say after PCA you ended up having 10 features. This dataset has no labels i.e. no target variable. How would you approach model building if you are asked to find patterns in the signals/ features. [unsupervised model example,]
- [ ] 6. What will be your approach on validating the output of an unsupervised model? [internal - cohesion , separation ; external - with labels use confusion matrix, without labels use twin sample validation]
- [ ] 7. Say you built a random classifier with a 1000 trees. The training error is 0 and validation error is quite high. What went wrong here ? [Overfitting] What situation is it - high / low bias/variance. Please explain the need to strike the balance between bias and variance.
- [ ] 8. How would you ensure you are not overfitting the model in the future ? [cross validation, regularization techniques]
- [ ] 9. How to improve your model when recall is low and precision is high

Deep Learning questions

- [x] What is a full connected layer
- [x] What is vanishing and exploding gradient and how to tackle it
- [x] Which activation function can be used in hidden layer and which can be used in output layer, give examples. What is the role of an activation function
- [ ] What is feature extraction, what happens exactly , explain
- [x] What is embedding , can you explain few embedding techniques
- [x] What is the downfall of one hot encoding
- [x] What is type 1 and type 2 error
- [ ] What are you achievements as a data scientist

Q9. You came to know that your model is suffering from low bias and high variance. Which algorithm should you use to tackle it? Why?

Answer: Low bias occurs when the model's predicted values are near to actual values. In other words, the model becomes flexible enough to mimic the training data distribution. While it sounds like great achievement, but not to forget, a flexible model has no generalization capabilities. It means, when this model is tested on an unseen data, it gives disappointing results.

In such situations, we can use bagging algorithm (like random forest) to tackle high variance problem. Bagging algorithms divides a data set into subsets made with repeated randomized sampling. Then, these samples are used to generate a set of models using a single learning algorithm. Later, the model predictions are combined using voting (classification) or averaging (regression).

Also, to combat high variance, we can:

Use regularization technique, where higher model coefficients get penalized, hence lowering model complexity.

Use top n features from variable importance chart. May be, with all the variable in the data set, the algorithm is having difficulty in finding the meaningful signal.

From <https://www.kaggle.com/discussions/general/231361>

Company Que-Ans

Sunday, October 15, 2023 3:33 PM

1. What is feature selection and why do you use feature selection?

   Feature selection is the process of selecting a subset of relevant features for use in model construction Feature selection methods can be used to identify and remove unneeded, irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may in fact decrease the accuracy of the model.

   There are three general classes of feature selection algorithms:

   a. filter methods

      Filter feature selection methods apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either selected to be kept or removed from the dataset. The methods are often univariate and consider the feature independently, or with regard to the dependent variable.

      Some examples of some filter methods include the Chi squared test, information gain and correlation coefficient scores.

   b. wrapper methods and

      Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model is used to evaluate a combination of features and assign a score based on model accuracy.

      The search process may be methodical such as a best-first search, it may stochastic such as a random hill-climbing algorithm, or it may use heuristics, like forward and backward passes to add and remove features.

      An example if a wrapper method is the recursive feature elimination algorithm.

   c. embedded methods

      Embedded methods learn which features best contribute to the accuracy of the model while the model is being created. The most common type of embedded feature selection methods are regularization methods.

      Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm (such as a regression algorithm) that bias the model toward lower complexity (fewer coefficients).

      Examples of regularization algorithms are the LASSO, Elastic Net and Ridge Regression.

2. What is the effect on the coefficients of logistic regression if two 3. predictors are highly correlated?
3. What are the confidence intervals of the coefficients?
4. What's the difference between Gaussian Mixture Model and K-Means?
5. How do you pick k for K-Means?
6. How do you know when Gaussian Mixture Model is applicable?
7. Assuming a clustering model's labels are known, how do you evaluate the performance of the model?