Unit – 1

1. Explain machine learning in brief. Discuss the application & limitations of machine learning.

- Machine Learning is a subset of AI (Artificial Intelligence) that uses statistical algorithms to build systems.
- A Machine learning system can learn and improve without being explicitly programmed.
- It can be classified into three types namely supervised learning, unsupervised learning, and reinforcement learning.
- Some Examples of Machine learning are the recommendation systems in music and video streaming services like Spotify, and YouTube and online shopping services like Amazon, Flipkart, etc.…

Applications of ML:

- **Image Recognition:** Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is the Automatic friend tagging suggestion: Facebook provides us with a feature of an auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with a name, and the technology behind this is machine learning's face detection and recognition algorithm. It is based on the Facebook project named "Deep Face," which is responsible for face recognition and person identification in the picture.
- **Speech Recognition:** While using Google, we get an option of "Search by voice," it comes under speech recognition, and it's a popular application of machine learning. Speech recognition is a

process of converting voice instructions into text, and it is also known as "Speech to text", or "Computer speech recognition." At present, machine learning algorithms are widely used in various applications of speech recognition. Google Assistant, Siri, Cortana, and Alexa are using speech recognition technology to follow voice instructions.

- **Traffic prediction:** If we want to visit a new place, we take the help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions. It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways: o Real Time location of the vehicle from the Google Map app and sensors and o Average time taken in past days at the same time. Everyone who is using Google Maps is helping this app to make it better. It takes information from the user and sends it back to its database to improve performance.
- **Product recommendations:** Machine learning is widely used by various e-commerce and entertainment companies such as Amazon, Netflix, etc., for product recommendations to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning. Google understands user interest using various machine learning algorithms and suggests the product as per customer interest. As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.
- **Self-driving cars:** One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car

manufacturing company is working on a self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

- **Email Spam and Malware Filtering:** Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Below are some spam filters used by Gmail:
    - ❖ Content Filter
    - ❖ Header filter
    - ❖ General blacklists filter
    - ❖ Rules-based filters
    - ❖ Permission filters

  Some machine learning algorithms such as Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier are used for email spam filtering and malware detection.

- **Virtual Personal Assistant:** We have various virtual personal assistants such as Google Assistant, Alexa, Cortana, and Siri. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, calling someone, opening an email, Scheduling an appointment, etc. These virtual assistants use machine learning algorithms as an important part. These assistants record our voice instructions, send them over to the server on a cloud, decode it using ML algorithms, and act accordingly.

Limitations of ML:

- **Possibility of High Error:** In ML, we can choose the algorithms based on accurate results. For that, we must run the results on every algorithm. The main problem occurs in the training and testing of data. The data is huge, so sometimes removing errors becomes nearly impossible. These errors can cause a headache to users. Since the data is huge, the errors take a lot of time to resolve.
- **Algorithm Selection:** The selection of an algorithm in Machine Learning is still a manual job. We must run and test our data in all the algorithms. After that only we can decide what algorithm, we want. We choose them based on result accuracy. The process is very much time-consuming.
- **Data Acquisition**: In ML, we constantly work on data. We take a huge amount of data for training and testing. This process can sometimes cause data inconsistency. The reason is some data constantly keep on updating. So, we must wait for the new data to arrive. If not, the old and new data might give different results. That is not a good sign for an algorithm.
- **Time and Space**: Many ML algorithms might take more time than you think. Even if it's the best algorithm it might sometimes surprise you. If your data is large and advanced, the system will take time. This may sometimes cause the consumption of more CPU power. Even with GPUs alongside, it sometimes becomes hectic. Also, the data might use more than the allotted space.

2. Compare Low dimensional & High dimensional Data about machine learning.

In machine learning, the dimensionality of data refers to the number of features or attributes that describe each instance or sample. Low-dimensional data typically refers to data with a small number of features, while high-dimensional data refers to data with many features. Here are some differences between low and high-dimensional data in the context of machine learning:

- Feature Selection: In low-dimensional data, it is easier to select important features or attributes that are relevant to the problem being solved. This can help improve the accuracy and performance of machine learning models. In contrast, high-dimensional data can make feature selection more challenging, as many of the features may be redundant, noisy, or irrelevant.
- Overfitting: High-dimensional data can lead to overfitting, which occurs when a model fits the training data too closely and fails to generalize well to new, unseen data. This is because high-dimensional data can lead to models that are too complex and have too many parameters, which can cause overfitting.
- Computation: High-dimensional data can also require more computational resources, both in terms of storage and processing power. This can make it more difficult to train machine learning models on large datasets with high-dimensional data.
- Interpretability: In some cases, low-dimensional data can be easier to interpret, as it is simpler to visualize and understand. In contrast, high-dimensional data can be more difficult to interpret, as there may be many interdependent features that are difficult to separate or analyze.

Overall, low-dimensional data can be easier to work with and lead to simpler, more interpretable models. However, high-dimensional data can provide more information and may be necessary for certain applications, such as image or speech recognition, where the input data can have many pixels or features. In these cases, techniques such as feature selection, dimensionality reduction, and regularization can help improve the performance of machine learning models on high-dimensional data.

3. Differentiate among AI, ML & DL.

- **AI:**
  - AI stands for Artificial Intelligence and is the study/process which enables machines to mimic human behavior through algorithms.
  - AI is the broader family consisting of ML and DL as its components.
  - AI is a computer algorithm that exhibits intelligence through decision-making.
  - Search Trees and much complex math are involved in AI.
  - The aim is to increase chances of success and not accuracy.
  - Three broad categories/types of AI are ANI (Artificial Narrow Intelligence), AGI(Artificial General Intelligence), and ASI(Artificial Super Intelligence).
  - The efficiency of AI is the efficiency provided by ML and DL respectively.
- ML:
  - ML stands for Machine Learning and is the study of statistical methods enabling machines to improve with experience.
  - ML is a subset of AI.
  - ML is an AI algorithm that allows the system to learn from data.
  - If you have an idea of logic (Math) you can visualize the complex functionalities like K-Mean, Support Vector Machines, etc.… then it defines the ML aspect.
  - The aim is to increase accuracy not caring about the success ratio.

  - ❖ Three categories in ML: Supervised Learning, Unsupervised Learning, and Reinforcement Learning.
  - ❖ Less efficient than DL as it cannot work on longer dimensions or a higher amount of data.

- DL:
  - ❖ DL stands for Deep Learning and is the study that makes use of neural Networks (like neurons present in the human brain) to imitate functionality just like the human brain.
  - ❖ DL is a subset of ML.
  - ❖ DL is an ML algorithm that uses neural networks to analyze data and provide output accordingly.
  - ❖ If you are clear about the math involved in it but don't have an idea about the features, you break the complex functionalities into linear/lower dimension features by adding more layers, then it defines the DL aspect.
  - ❖ It attains the highest rank in terms of accuracy when it is trained with a large amount of data.
  - ❖ DL can be considered as neural networks with many parameters layers lying in one of the four fundamental network architectures: Unsupervised Pre trained Networks, Convolutional Neural Networks, Recurrent Neural Networks, and Recursive Neural Networks
  - ❖ More powerful than ML as it easily works for larger sets of data.

4. Discuss various types of machine learning algorithms.

- **Supervised Learning Algorithm:**

  Supervised learning is a type of Machine learning in which the machine needs external supervision to learn. The supervised learning models are trained using the labeled dataset. Once the training and processing are done, the model is tested by providing sample test data to check whether it predicts the correct output. The goal of supervised learning is to map input data with output data. Supervised learning is based on supervision, and it is the same as when a student learns things under the teacher's supervision. An example of supervised learning is spam filtering. Supervised learning can be divided further into two categories of problems:

- ❖ Classification
- ❖ Regression

Examples of some popular supervised learning algorithms are Simple Linear regression, Decision Tree, Logistic Regression, KNN algorithm, etc.

Unsupervised Learning Algorithm:

It is a type of machine learning in which the machine does not need any external supervision to learn from the data, hence called unsupervised learning. The unsupervised models can be trained using the unlabeled dataset that is not classified, nor categorized, and the algorithm needs to act on that data without any supervision. In unsupervised learning, the model doesn't have a predefined output, and it tries to find useful insights from a huge amount of data. These are used to solve the Association and Clustering problems. Hence further, it can be classified into two types:

- ❖ Clustering
- ❖ Association

Examples of some Unsupervised learning algorithms are K-means Clustering, Apriori Algorithm, Eclat, etc.

Reinforcement Learning:

In Reinforcement learning, an agent interacts with its environment by producing actions and learning with the help of feedback. The feedback is given to the agent in the form of rewards, such as for each good action, he gets a positive reward, and for each bad action, he gets a negative reward. There is no supervision provided to the agent. The q-Learning algorithm is used in reinforcement learning.

5. Explain, why the knowledge of linear algebra, statistics & probability theory is beneficial in machine learning development.

Linear algebra, statistics, and probability theory are all essential components of machine learning, and they play a crucial role in the development of algorithms and models used in this field.

- Linear algebra is the study of vector spaces and linear transformations, and it provides a mathematical framework for representing and manipulating data in machine learning. Linear algebra is used extensively in the development of algorithms for tasks such as data preprocessing, feature extraction, and model training. For example, matrix multiplication, eigenvalue decomposition, and singular value decomposition are all linear algebra techniques that are widely used in machine learning.
- Statistics is the branch of mathematics that deals with the collection, analysis, interpretation, presentation, and organization of data. Statistics provides a foundation for understanding the underlying principles of machine learning, such as the assumptions and limitations of different algorithms and models. Statistical techniques such as hypothesis testing, regression analysis, and Bayesian inference are used in machine learning to estimate model parameters, evaluate model performance, and make predictions.
- Probability theory is the study of random phenomena, and it provides a mathematical framework for understanding uncertainty and randomness in machine learning. Probability theory is used extensively in the development of algorithms for tasks such as

decision-making, risk assessment, and uncertainty quantification. For example, probability distributions such as the Gaussian distribution and the Bernoulli distribution are used to model uncertainty in machine learning.

In summary, knowledge of linear algebra, statistics, and probability theory is essential for developing and implementing effective machine-learning algorithms and models. These mathematical concepts provide a foundation for understanding the underlying principles of machine learning, and they are used to develop and optimize algorithms that can analyze and interpret large datasets, make predictions, and automate decision-making processes.

UNIT2

1. *Explain what bias and variance and their effect on the accuracy of the ML model developed. Also, discuss the bias-variance trade-off in brief.*

Bias:

In general, a machine learning model analyses the data finds patterns in it, and makes predictions. While training, the model learns these patterns in the dataset and applies them to test data for prediction. While making predictions, a difference occurs between prediction values made by the model and actual values/expected values, and this difference is known as bias errors or Errors due to bias. It can be defined as an inability of machine learning algorithms such as Linear Regression to capture the true relationship between the data points. Each algorithm begins with some amount of bias because bias occurs from assumptions in the model, which makes the target function simple to learn.

A model has either:

- **Low Bias:**
  A low-bias model will make fewer assumptions about the form of the target function.
- **High Bias:**
  A model with a high bias makes more assumptions, and the model becomes unable to capture the important features of our dataset.

A high-bias model also cannot perform well on new data. Generally, a linear algorithm has a high bias, as it makes them learn fast. The simpler the algorithm, the higher the bias it has likely to be introduced. Whereas a nonlinear algorithm often has low bias. Some examples of machine learning algorithms with low bias are Decision Trees, k-Nearest Neighbors, and Support Vector Machines. At the same time, an algorithm with high bias is Linear Regression, Linear Discriminant Analysis, and Logistic Regression.

Ways to reduce High Bias:

High bias mainly occurs due to a much simpler model. Below are some ways to reduce the high bias:

- Increase the input features as the model is under-fitted.
- Decrease the regularization term.
- Use more complex models, such as including some polynomial features.

Variance:

The variance would specify the amount of variation in the prediction of the different training data used. In simple words, variance tells how much a random variable is different from its expected value. Ideally, a model should not vary too much from one training dataset to another, which means the algorithm should be good at understanding the hidden mapping between inputs and output variables.

Variance errors are either of low variance or high variance. Low variance means there is a small variation in the prediction of the target function with changes in the training data set. At the same time, High variance shows a large variation in the prediction of the target function with changes in the training dataset.

A model that shows high variance learns a lot and performs well with the training dataset and does not generalize well with the unseen dataset. As a result, such a model gives good results with the training dataset but shows high error rates on the test dataset. Since, with high variance, the model learns too much from the dataset, it leads to overfitting of the model.

A model with high variance has the below problems:

- A high variance model leads to overfitting.
- Increase model complexities.

Usually, nonlinear algorithms have a lot of flexibility to fit the model and have high variance. Some examples of machine learning algorithms with low variance are Linear Regression, Logistic Regression, and Linear discriminant analysis. At the same time, algorithms with high variance are decision trees, Support Vector Machines, and K-nearest neighbors.

Ways to Reduce High Variance:

- Reduce the input features or several parameters as a model is overfitted.
- Do not use a complex model.
- Increase the training data.
- Increase the Regularization term.

2. Differentiate between Overfitting & Underfitting in machine learning.

Underfitting:

A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data, i.e., it only performs well on training data but performs poorly on testing data. (It’s just like trying to fit undersized pants!)

Underfitting destroys the accuracy of our machine-learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough. It usually happens when we have fewer data to build an accurate model and when we try to build a linear model with fewer non-linear data.

In such cases, the rules of the machine learning model are too easy and flexible to be applied to such minimal data and therefore the model will probably make a lot of wrong predictions. Underfitting can be avoided by using more data and reducing the features by feature selection. In a nutshell, Underfitting refers to a model that can neither performs well on the training data nor generalize to new data.

Reasons for Underfitting:

- High bias and low variance
- The size of the training dataset used is not enough.
- The model is too simple.
- Training data is not cleaned and contains noise in it.

Techniques to reduce underfitting:

- Increase model complexity.
- Increase the number of features, performing feature engineering.
- Remove noise from the data.
- Increase the number of epochs or increase the duration of training to get better results.

Overfitting:

A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise.

The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore, they can build unrealistic models.

A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.

In a nutshell, Overfitting is a problem where the evaluation of machine learning algorithms on training data is different from unseen data.

The reasons for Overfitting:

- ❖ High variance and low bias
- ❖ The model is too complex.
- ❖ The size of the training data

Examples:

Techniques to reduce overfitting:

- ❖ Increase training data.
- ❖ Reduce model complexity.
- ❖ Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
- ❖ Ridge Regularization and Lasso Regularization
- ❖ Use dropout for neural networks to tackle overfitting.

3. Explain cross-validation (CV) in brief using a diagram and discuss different-different CV techniques used in machine learning.

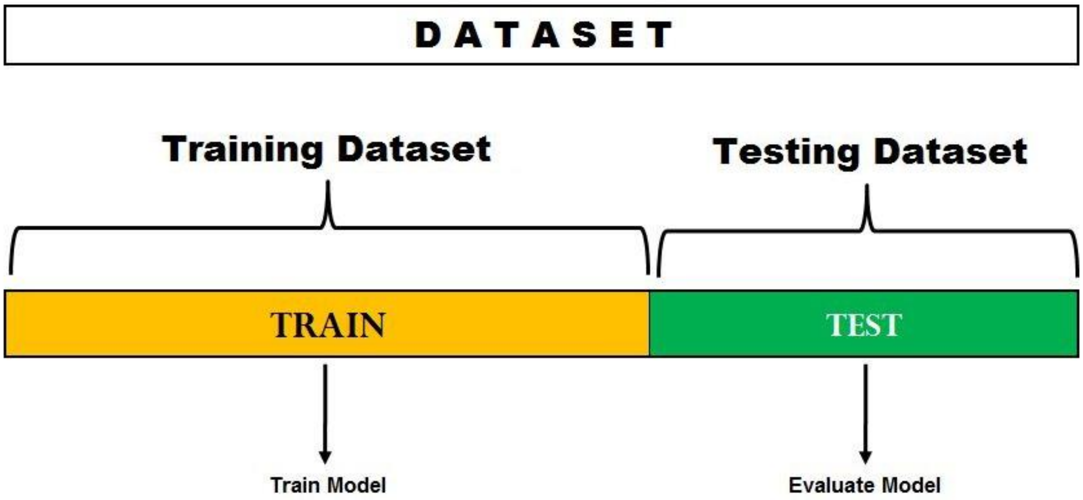
Cross-Validation:

Cross-validation is a technique used in machine learning to evaluate the performance of a model on unseen data. It involves dividing the available data into multiple folds or subsets, using one of these folds as a validation set, and training the model on the remaining folds. This process is repeated multiple times, each time using a different fold as the validation set. Finally, the results from each validation step are averaged to produce a more robust estimate of the model's performance.

The main purpose of cross-validation is to prevent overfitting, which occurs when a model is trained too well on the training data and performs poorly on new, unseen data. By evaluating the model on multiple validation sets, cross-validation provides a more realistic estimate of the model's generalization performance, i.e., its ability to perform well on new, unseen data.

Hold Out method:

This is the simplest evaluation method and is widely used in Machine Learning projects. Here the entire dataset(population) is divided into 2 sets – the train set and the test set. The data can be divided into 70-30 or 60-40, 75-25 or 80-20, or even 50-50 depending on the use case. As a rule, the proportion of training data must be larger than the test data.



The data split happens randomly, and we can’t be sure which data ends up in the train and test bucket during the split unless we specify a random state. This can lead to extremely high variance and every time, the split changes, the accuracy will also change.

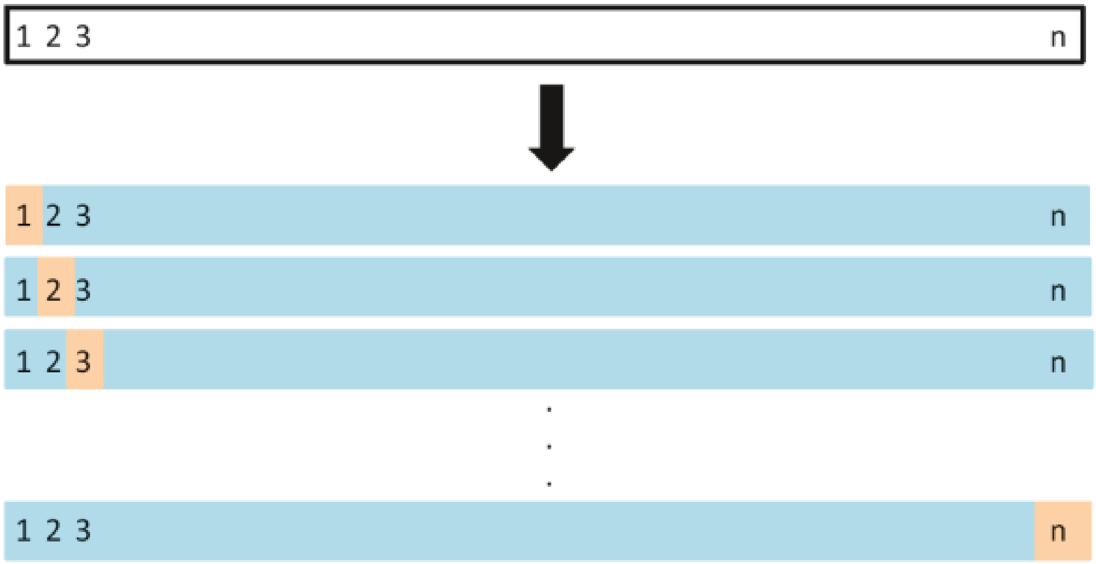
<u>Drawbacks:</u>

- ❖ In the Holdout method, the test error rates are highly variable (**high variance**), and it depends on which observations end up in the training set and test set
- ❖ Only a part of the data is used to train the model (**high bias**) which is not a very good idea when the data is not huge, and this will lead to an overestimation of test error.

One of the major advantages of this method is that it is computationally inexpensive compared to other cross-validation techniques.

Leave One Out Cross-Validation

In this method, we divide the data into train and test sets – but with a twist. Instead of dividing the data into 2 subsets, we select a single observation as test data, and everything else is labeled as training data and the model is trained. Now the 2nd observation is selected as test data and the model is trained on the remaining data.



This process continues ‘n’ times and the average of all these iterations is calculated and estimated as the test set error.

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i.$$

When it comes to test-error estimates, LOOCV gives unbiased estimates (**low bias**). But bias is not the only matter of concern in estimation problems. We should also consider variance.

LOOCV has an extremely **high variance** because we are averaging the output of n-models fitted on an almost identical set of observations, and their outputs are highly positively correlated.

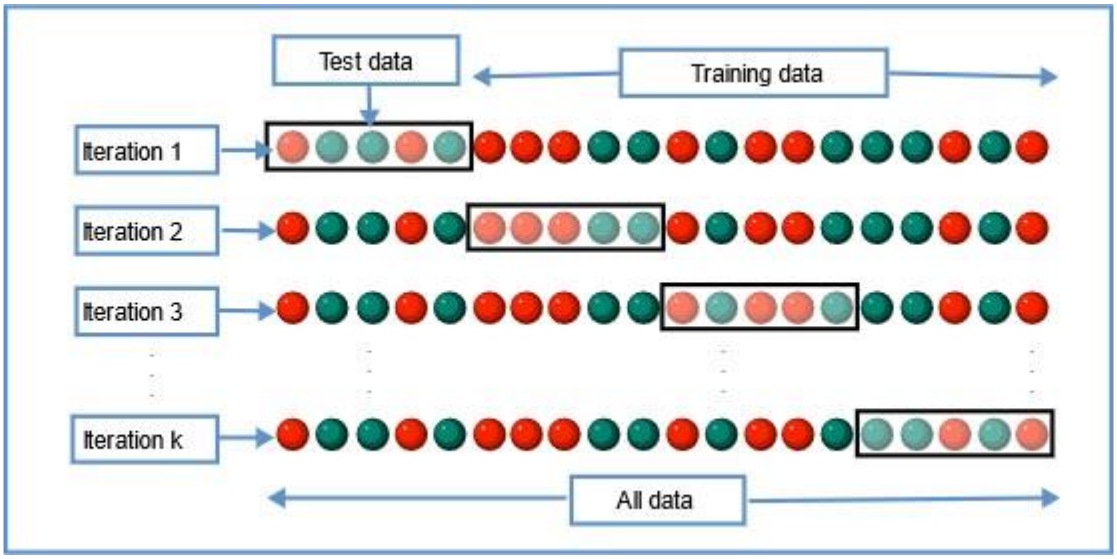
And you can see this is computationally expensive as the model is run ‘n’ times to test every observation in the data. Our next method will tackle this problem and give us a good balance between bias and variance.

This output clearly shows how LOOCV keeps one observation aside as test data and all the other observations go to train data.

K-Fold Cross-Validation

In this resampling technique, the whole data is divided into k sets of almost equal sizes. The first set is selected as the test set and the model is trained on the remaining k-1 sets. The test error rate is then calculated after fitting the model to the test data.

In the second iteration, the 2nd set is selected as a test set and the remaining k-1 sets are used to train the data and the error is calculated. This process continues for all the k sets.



The mean of errors from all the iterations is calculated as the CV test error estimate.

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i.$$

In K-Fold CV, the no of folds k is less than the number of observations in the data (k<n) and we are averaging the outputs of k-fitted models

that are somewhat less correlated with each other since the overlap between the training sets in each model is smaller. This leads to **low variance** than LOOCV.

The best part about this method is each data point gets to be in the test set exactly once and gets to be part of the training set k-1 times. As the number of folds k increases, the variance also decreases (low variance). This method leads to **intermediate bias** because each training set contains fewer observations (k-1)n/k than the Leave One Out method but more than the Hold Out method.

Typically, K-fold Cross Validation is performed using k=5 or k=10 as these values have been empirically shown to yield test error estimates that neither have high bias nor high variance.

The major disadvantage of this method is that the model has to be run from scratch k-times and is more computationally expensive than the Hold Out method but better than the Leave One Out method.

Stratified K-Fold Cross-Validation

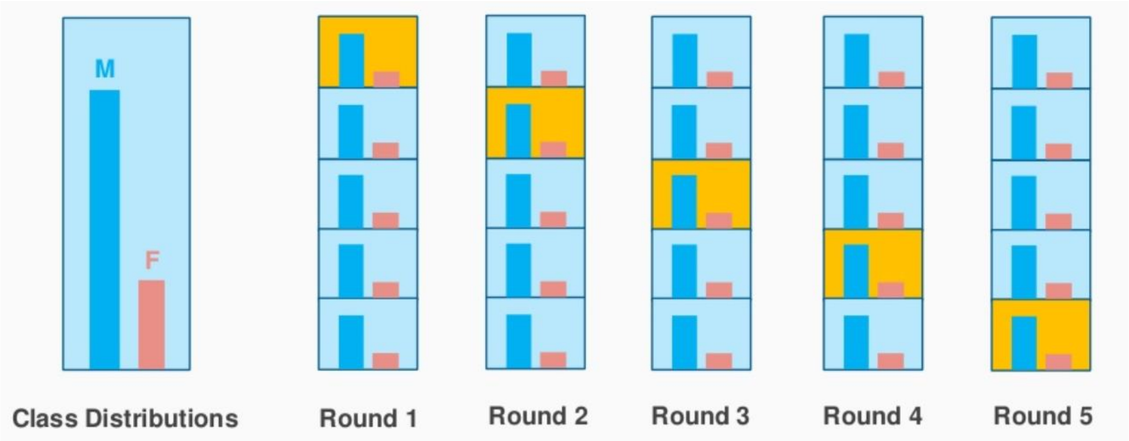
This is a slight variation from K-Fold Cross Validation, which uses **‘stratified sampling’** instead of ‘random sampling.’

Let’s quickly understand what stratified sampling is and how is it different from random sampling.

Suppose your data contains reviews for a cosmetic product used by both the male and female population. When we perform random sampling to split the data into train and test sets, there is a possibility that most of the data representing males is not represented in training data but might end up in test data. When we train the model on sample training data that is not a correct representation of the actual population, the model will not predict the test data with good accuracy.

This is where Stratified Sampling comes to the rescue. Here the data is split in such a way that it represents all the classes from the population.

Let’s consider the above example which has a cosmetic product review of 1000 customers out of which 60% is female and 40% is male. I want to split the data into train and test data in proportion (80:20). 80% of 1000 customers will be 800 which will be chosen in such a way that there are 480 reviews associated with the female population and 320 representing the male population. Similarly, 20% of 1000 customers will be chosen for the test data (with the same female and male representation).



This is exactly what stratified K-Fold CV does and it will create K-Folds by preserving the percentage of sample for each class. This solves the problem of random sampling associated with Hold out and K-Fold methods.

The output clearly shows the stratified split done based on the classes ‘0’ and ‘1’ in ‘y’.

4. What do you mean by performance metrics/KPI? List different-2 KPIs used to evaluate ML model performance.

Evaluating the performance of a Machine learning model is one of the important steps while building an effective ML model.

To evaluate the performance or quality of the model, different metrics are used, and these metrics are known as performance metrics or evaluation metrics.

These performance metrics help us understand how well our model has performed for the given data. In this way, we can improve the model's performance by tuning the hyperparameters.

Each ML model aims to generalize well on unseen/new data, and performance metrics help determine how well the model generalizes on the new dataset. n machine learning, each task or problem is divided into classification and Regression. Not all metrics can be used for all types of problems; hence, it is important to know and understand which metrics should be used.

Different evaluation metrics are used for both Regression and Classification tasks.

- **Performance Metrics for Classification**

In a classification problem, the category or classes of data is identified based on training data. The model learns from the given dataset and then classifies the new data or groups based on the training. It predicts class labels as the output, such as Yes or No, 0 or 1, Spam or Not Spam, etc.

To evaluate the performance of a classification model, different metrics are used, and some of them are as follows:

❖ Accuracy:

The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions.

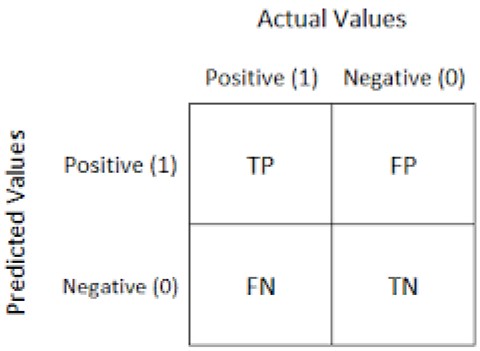
It can be formulated as:

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

Although it is simple to use and implement, it is suitable only for cases where an equal number of samples belong to each class.

❖ Confusion Matrix:

A confusion matrix is a tabular representation of prediction outcomes of any binary classifier, which is used to describe the performance of the classification model on a set of test data when true values are known. The confusion matrix is simple to implement, but the terminologies used in this matrix might be confusing for beginners. A typical confusion matrix for a binary classifier looks like the below image (However, it can be extended to use for classifiers with more than two classes).



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

We can determine the following from the above matrix:

● In the matrix, columns are for the prediction values, and rows specify the Actual values. Here Actual and prediction give two possible classes, Yes or No. So, if we are predicting the presence of a disease in a patient, the Prediction column with Yes means, Patient has the disease, and for NO, the Patient doesn't have the disease.

● In this example, the total number of predictions are 165, out of which 110 time predicted yes, whereas 55 times predicted No.

● However, 60 cases in which patients don't have the disease, whereas 105 cases in which patients have the disease.

In general, the table is divided into four terminologies, which are as follows:

➢ True Positive (TP): In this case, the prediction outcome is true, and it is true, also.

➢ True Negative (TN): in this case, the prediction outcome is false, and it is false, also.

➢ False Positive (FP): In this case, prediction outcomes are true, but they are false.

➢ False Negative (FN): In this case, predictions are false, and they are true.

❖ Precision

The precision metric is used to overcome the limitation of Accuracy. The precision determines the proportion of positive prediction that was correct. It can be calculated as the True Positive or predictions that are true to the total positive predictions (True Positive and False Positive).

$$P = \frac{TP}{TP+FP} = \frac{\text{Cancer patients correctly identified}}{\text{Cancer patients correctly identified+incorrectly labelled cancer patients as non-cancerous}}$$

❖ Recall or Sensitivity

It is also like the Precision metric; however, it aims to calculate the proportion of actual positive that was identified incorrectly. It can be calculated as a True Positive or a prediction that is true to the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (true Positive and false negative).

The formula for calculating Recall is given below:

$$R = \frac{TP}{TP+FN} = \frac{\text{Cancer patients correctly identified}}{\text{Cancer patients correctly identified+incorrectly labelled non-cancer patients as cancerous}}$$

When to use Precision and Recall?

From the above definitions of Precision and Recall, we can say that recall determines the performance of a classifier concerning a false negative, whereas precision gives information about the performance of a classifier with a concerning positive. So, if we want to minimize the false negative, then, Recall should be as near to 100%, and if we want to minimize the false positive, then precision should be close to 100% as possible. In simple words, if we maximize precision, it will minimize the FP errors, and if we maximize recall, it will minimize the FN error.

❖ F-Scores

F-score or F1 Score is a metric to evaluate a binary classification model based on predictions that are made for the positive class. It is calculated with the help of Precision and Recall. It is a type of single score that represents both Precision and Recall. So, the F1 Score can be calculated as the harmonic mean of both precision and Recall, assigning equal weight to each of them.

The formula for calculating the F1 score is given below:

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

When to use F-Score? As F-score makes use of both precision and recall, it should be used if both are important for evaluation, but one (precision or recall) is slightly more important to consider than the other. For example, when False negatives are comparatively more important than false positives, or vice versa.

❖ AUC-ROC

Sometimes we need to visualize the performance of the classification model on charts; then, we can use the AUC-ROC curve. It is one of the popular and important metrics for evaluating the performance of the classification model. Firstly, let's understand ROC (Receiver Operating Characteristic curve) curve. ROC represents a graph to show the performance of a classification model at different threshold levels.

The curve is plotted between two parameters, which are:

- True Positive Rate
- False Positive Rate

TPR or true Positive rate is a synonym for Recall, hence can be calculated as:

$$\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$$

FPR or False Positive Rate can be calculated as:

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}}$$

To calculate value at any point in a ROC curve, we can evaluate a logistic regression model multiple times with different classification thresholds, but this would not be much efficient. So, for this, one efficient method is used, which is known as AUC.

<u>AUC:</u>

Area Under the ROC curve AUC is known for Area Under the ROC curve. As its name suggests, AUC calculates the two-dimensional area under the entire ROC curve, as shown below image:

AUC calculates the performance across all the thresholds and provides an aggregate measure. The value of AUC ranges from 0 to 1.

It means a model with 100% wrong prediction will have an AUC of 0.0, whereas models with 100% correct predictions will have an AUC of 1.0. When to Use AUC should be used to measure how well the predictions are ranked rather than their absolute values.

Moreover, it measures the quality of predictions of the model without considering the classification threshold. When not to use AUC As AUC is scale-invariant, which is not always desirable, and we need calibrating probability outputs, then AUC is not preferable. Further, AUC is not a useful metric when there are wide disparities in the cost of false negatives vs. false positives, and it is difficult to minimize one type of classification error.

Performance Metrics for Regression

Regression is a supervised learning technique that aims to find the relationships between the dependent and independent variables. A predictive regression model predicts a numeric or discrete value. The metrics used for regression are different from the classification metrics. It means we cannot use the Accuracy metric (explained above) to evaluate a regression model; instead, the performance of a Regression model is reported as errors in the prediction.

Following are the popular metrics that are used to evaluate the performance of Regression models.

❖ Mean Absolute Error (MAE)

Mean Absolute Error or MAE is one of the simplest metrics, which measures the absolute difference between actual and predicted values, where absolute means taking a number as Positive.

To understand MAE, let's take an example of Linear Regression, where the model draws a best-fit line between dependent and independent variables.

To measure the MAE or error in prediction, we need to calculate the difference between actual values and predicted values. But to find the absolute error for the complete dataset, we need to find the mean absolute of the complete dataset.

The below formula is used to calculate MAE:

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_j - \check{y}_j|$$

Here, Y is the Actual outcome, Y' is the predicted outcome, and N is the total number of data points. MAE is much more robust for the outliers. One of the limitations of MAE is that it is not differentiable, so for this, we need to apply different optimizers such as Gradient Descent. However, to overcome this limitation, another metric can be used, which is Mean Squared Error or MSE.

❖ Mean Squared Error

Mean Squared error or MSE is one of the most suitable metrics for Regression evaluation. It measures the average of the Squared difference between predicted values and the actual value given by the model. Since in MSE, errors are squared, therefore it only assumes non-negative values, and it is usually positive and non-zero.

Moreover, due to squared differences, it penalizes small errors also, and hence it leads to over-estimation of how bad the model is. MSE is a much-preferred metric compared to other regression metrics as it is differentiable and hence optimized better.

The formula for calculating MSE is given below:

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

Here, Y is the Actual outcome, Y' is the predicted outcome, and N is the total number of data points.

❖ R Squared Score

R squared error is also known as the Coefficient of Determination, which is another popular metric used for Regression model evaluation. The R-squared metric enables us to compare our model with a constant baseline to determine the performance of the model. To select the constant baseline, we need to take the mean of the data and draw the line at the mean. The R-squared score will always be less than or equal to 1 without concern if the values are too large or small.

5. What do you understand by fine-tuning an ML model? Discuss Grid Search & Randomized search method in brief.

Fine-tuning a machine learning model to improve the accuracy of the results that are forecasted.

The steps involved in fine-tuning are:

- ✓ After getting to know more about machine learning and about tuning in detail, you should then determine the metric that you are going to use to record the accuracy of the model.
- ✓ Test the accuracy of the model after you set the required accuracy metric by using cross-validation methodologies.
- ✓ Once you are set with the accuracy, then determine the parameters that your machine learning model requires with the help of the validation curve.
- ✓ Afterward, do a grid search to enhance the parameter condition.
- ✓ If you aren’t satisfied with the accuracy, then keep on tuning it continuously.

Grid Search and Randomized Search are two popular methods used in machine learning to tune hyperparameters for a model. Hyperparameters are parameters that are not learned from the data during training but are set before training and can have a significant impact on the model's performance.

Grid Search:

Grid Search is a method of systematically trying out a range of values for each hyperparameter in a model, creating a grid of all possible combinations of hyperparameters, and testing each combination to find the optimal set of hyperparameters. The grid search algorithm exhaustively searches over all possible hyperparameter values, making it a computationally expensive method, but it ensures that the optimal hyperparameters are found.

For example, if we have two hyperparameters, C and gamma, with the following possible values:

C = [1, 10, 100] and gamma = [0.1, 1, 10],

then the grid search algorithm will test the model with all possible combinations of hyperparameters:

(C=1, gamma=0.1),(C=1, gamma=1),(C=1, gamma=10),(C=10, gamma=0.1),
(C=10,gamma=1),(C=10,gamma=10),(C=100,gamma=0.1),(C=100, gamma=1), and (C=100, gamma=10)

- **Randomized Search:**

Randomized Search is a method of randomly sampling hyperparameters from a specified range of values and testing each combination to find the optimal set of hyperparameters.

The main difference between Randomized Search and Grid Search is that Randomized Search doesn't search over all possible hyperparameters, but instead randomly selects a subset of hyperparameters to test. This makes it a more computationally efficient method than Grid Search, but there is no guarantee that the optimal set of hyperparameters will be found.

For example, if we have two hyperparameters, C and gamma, with the following possible values: C = [1, 10, 100] and gamma = [0.1, 1, 10], then the Randomized Search algorithm will randomly select hyperparameters to test from the specified ranges, such as (C=10, gamma=0.1), (C=100, gamma=1), (C=1, gamma=10), and so on. The number of iterations of random sampling can be specified beforehand.

In summary, Grid Search and Randomized Search are two popular methods used for hyperparameter tuning in machine learning.

Grid Search exhaustively searches over all possible hyperparameter combinations, while Randomized Search randomly samples hyperparameters from a specified range.

Grid Search ensures that the optimal hyperparameters are found, but can be computationally expensive, while Randomized Search is more computationally efficient, but there is no guarantee of finding the optimal set of hyperparameters.

6. What do you understand by ensemble learning? Also, discuss the concept of bagging & boosting in detail.

Ensemble Learning:

Ensemble learning is a machine learning technique that combines multiple models to improve predictive performance. Ensemble learning involves creating a group of models that individually produce predictions, and then combining these predictions in some way to produce a final prediction. Ensemble learning can be used with a wide variety of machine learning algorithms, including decision trees, neural networks, and support vector machines.

- Bagging (Bootstrap Aggregating) is an ensemble learning method that involves training multiple models on different subsets of the training data. These subsets are randomly selected with replacements from the original dataset, which means that some instances may be included in multiple subsets, and some may not be included at all. Each model is then trained on one of these subsets, and the final prediction is made by averaging the predictions of all models. Bagging reduces the variance of the model and decreases the chances of overfitting.

- Boosting is another ensemble learning method that involves creating multiple models, but in contrast to bagging, each subsequent model is trained on a modified version of the training data. Boosting focuses on minimizing the model's bias by emphasizing the misclassified data points. The algorithm identifies the most difficult-to-classify data points and assigns them higher weights, and the next model is trained on the data points weighted accordingly. This process is repeated for several models, and the final prediction is made by combining the predictions of all models. Boosting reduces the bias of the model, but it may increase variance and the chances of overfitting.

In summary, bagging and boosting are two popular ensemble learning methods that aim to improve the predictive performance of machine learning models. Bagging reduces the variance of the model and decreases the chances of overfitting while boosting reduces the bias of the model by focusing on the misclassified data points.

UNIT3:

1. Explain regression & classification problems using suitable examples.

REGRESSION:

A regression problem is when the output variable is a real or continuous value, such as “salary” or “weight”. Many different models can be used, the simplest is linear regression.

It tries to fit data with the best hyper-plane which goes through the points.

- PROBLEM:

  Which of the following is a regression task?

  - Predicting the age of a person
  - Predicting the nationality of a person
  - Predict whether the stock price of a company will increase tomorrow.
  - Predict whether a document is related to the sighting of UFOs?
- SOLUTION:

  Predicting the age of a person (because it is a real value, predicting nationality is categorical, whether the stock price will increase is a discrete yes/no answer, and predicting whether a document is related to UFO is again discrete- a yes/no answer).

  <u>Popular regression algorithms include:</u>

  - Simple Linear Regression

- Multiple Linear Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression

Ex: The price of a house. Let’s assume we must build a model to predict a house price by showing images to the ML model.

The dataset which we will use will have target output in numeric variables / unlabeled output i.e., 4000000, 4555545.67. So, we will provide input and output labels based on numeric data inserted in the dataset to the algorithm. Then using the below popular algorithm, we will build a predictive model.

CLASSIFICATION:

A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. A classification model attempts to draw some conclusions from observed values.

Given one or more inputs a classification model will try to predict the value of one or more outcomes. For example, when filtering emails “spam” or “not spam”, and when looking at transaction data, “fraudulent”, or “authorized”.

In short, Classification either predicts categorical class labels or classifies data construct a model) based on the training set and the values (class labels) in classifying attributes and uses it in classifying new data. There are several classification models.

Classification models include logistic regression, decision tree, random forest, gradient-boosted tree, multi-layer perceptron, one-vs-rest, and Naive Bayes.

- PROBLEM:

  Which of the following is/are classification problem(s)?

  - Predicting the gender of a person by his/her handwriting style Predicting house price based on area.
  - Predict whether monsoons will be normal next year.
  - Predict the number of copies a music album will be sold next month.
- SOLUTION:

  Predicting the gender of a person

  Predicting whether monsoons will be normal next year.

2. Explain Linear regression using a suitable diagram.

Regression:

Regression is a technique for investigating the relationship between independent variables or features and a dependent variable or outcome. It's used as a method for predictive modeling in machine learning, in which an algorithm is used to predict continuous outcomes.

- Linear Regression:
  - Linear regression is a statistical regression method that is used for predictive analysis.

- ❖ It is one of the very simple and easy algorithms which works on regression and shows the relationship between the continuous variables.
- ❖ It is used for solving the regression problem in machine learning.
- ❖ Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), hence called linear regression.
- ❖ If there is only one input variable (x), then such linear regression is called simple linear regression. And if there is more than one input variable, then such linear regression is called multiple linear regression.
- ❖ The relationship between variables in the linear regression model can be explained using the below image. Here we are predicting the salary of an employee based on the year of experience.
- ❖ Below is the mathematical equation for Linear regression:

$$Y= ax +b$$

Here, Y = dependent variables (target variables), X= Independent variables (predictor variables), and a and b are the linear coefficients

Some popular applications of linear regression are:

- ❖ Analyzing trends and sales estimates
- ❖ Salary forecasting
- ❖ Real estate prediction
- ❖ Arriving at ETAs in traffic.

3. Discuss the Gradient Descent Algorithm in detail.

Gradient Descent is a powerful optimization algorithm used to minimize the error or cost function of a mathematical model.

It is a widely used algorithm in machine learning and deep learning to train models by minimizing the error or cost function.

In Gradient Descent, the objective is to find the optimal values of the parameters that minimize the cost function. The cost function represents the difference between the predicted output of the model and the actual output.

The goal of Gradient Descent is to iteratively adjust the values of the parameters in the direction of the steepest descent of the cost function until the minimum of the cost function is reached.

The process of Gradient Descent can be explained as follows:

- ❖ **Initialization:** In the first step, the initial values of the parameters are randomly selected.
- ❖ **Forward Propagation:** The inputs are passed through the model to get the predicted output.
- ❖ **Calculation of the Cost Function:** The cost function is calculated using the predicted output and the actual output.
- ❖ **Backward Propagation:** The gradients of the cost function concerning each of the parameters are calculated using the chain rule of differentiation.
- ❖ **Parameter Update:** The values of the parameters are updated in the direction of the steepest descent of the cost function using the gradients calculated in the previous step.
- ❖ **Repeat:** Steps 2-5 are repeated until the cost function reaches a minimum or convergence is achieved.

There are two types of Gradient Descent algorithms:

- **Batch Gradient Descent:**

  In Batch Gradient Descent, the gradients of the cost function concerning each of the parameters are calculated using the entire dataset. This method is computationally expensive and is used when the dataset is small.

- **Stochastic Gradient Descent:**

  In Stochastic Gradient Descent, the gradients of the cost function concerning each of the parameters are calculated using a single sample from the dataset. This method is computationally efficient and is used when the dataset is large.

  There is also a hybrid approach called Mini-Batch Gradient Descent, which is a compromise between the above two approaches. In Mini-Batch Gradient Descent, the gradients of the cost function concerning each of the parameters are calculated using a small batch of samples from the dataset.

  There are several variations of the Gradient Descent algorithm, such as Momentum-based Gradient Descent, Adagrad, RMSProp, and Adam. These variations are designed to overcome the limitations of the standard Gradient Descent algorithm and converge faster and more efficiently.

  In conclusion, Gradient Descent is a powerful optimization algorithm used to minimize the error or cost function of a mathematical model. It is widely used in machine learning and deep learning to train models by minimizing the error or cost function. The algorithm is iterative and updates the parameters in the direction of the steepest descent of the cost function until the minimum of the cost function is reached.

4.Discuss Multiple linear regression (MLR) using examples.

Multiple Linear Regression (MLR) is a statistical technique used to analyze the relationship between two or more independent variables and a dependent variable. In other words, it is used to model the relationship between a dependent variable and multiple independent variables.

Let's take an example to understand MLR in detail:

Suppose we want to predict the house prices based on its size, number of bedrooms, and the age of the house. Here, the dependent variable is the price of the house, and the independent variables are the size, number of bedrooms, and the age of the house.

To perform MLR, we need a dataset with values for the dependent and independent variables. Let's assume that we have a dataset with the following information:

Size (sq ft)	Bedrooms	Age (years)	Price ($)
2000	3	10	200,000
3000	4	5	350,000
1500	2	20	150,000
2500	4	8	275,000
1800	3	15	200,000

To perform MLR on this dataset, we follow these steps:

- Choose the dependent variable and independent variables:

In this example, the dependent variable is the house price, and the independent variables are the size, number of bedrooms, and age of the house.

- Formulate the hypothesis:

We formulate the hypothesis that there is a linear relationship between the dependent variable and the independent variables.

- Estimate the coefficients: We use statistical methods to estimate the coefficients of the independent variables in the regression equation. The regression equation can be written as follows:

  $\text{Price (\$)} = \beta 0 + \beta 1 \times \text{Size (sq ft)} + \beta 2 \times \text{Bedrooms} + \beta 3 \times \text{Age (years)} + \varepsilon$

  where β0, β1, β2, and β3 are the coefficients of the independent variables, and ε is the error term.
- Evaluate the model:

  We evaluate the model by calculating the R-squared value, which measures how well the model fits the data.
- Make predictions:

  We use the regression equation to make predictions of house prices based on the values of the independent variables.

  For example,

  if we want to predict the price of a house with a size of 2800 sq ft, 3 bedrooms, and an age of 7 years, we can use the regression equation as follows:

  $\text{Price (\$)} = \beta 0 + \beta 1 \times \text{Size (sq ft)} + \beta 2 \times \text{Bedrooms} + \beta 3 \times \text{Age (years)} + \varepsilon = \beta 0 + \beta 1 \times 2800 + \beta 2 \times 3 + \beta 3 \times 7 + \varepsilon$

  We can substitute the estimated coefficients for β0, β1, β2, and β3 and get the predicted value of the house price.

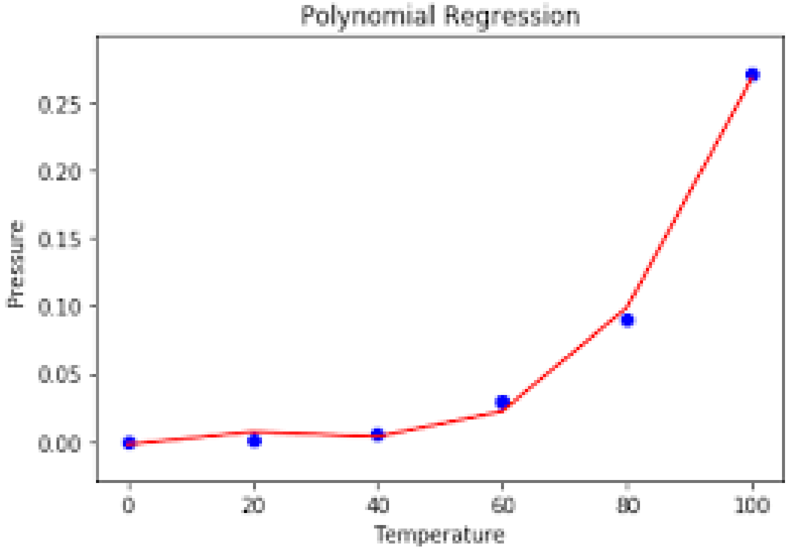
  In conclusion, Multiple Linear Regression (MLR) is a statistical technique used to analyze the relationship between a dependent variable and multiple independent variables. It is widely used in many fields, including finance, marketing, and healthcare. By using MLR, we can make predictions based on multiple independent variables, which can help us make better decisions.

5. Explain Polynomial Regression briefly and discuss how it differs from multiple linear regression.

Polynomial Regression is a regression analysis technique that models the relationship between a dependent variable and one or more independent variables by fitting a polynomial equation to the data. It is an extension of multiple linear regression and is used when the relationship between the dependent and independent variable(s) is not linear.

In Polynomial Regression, we use a polynomial equation of degree n to model the relationship between the dependent variable and independent variable(s). The degree of the polynomial determines the complexity of the model. For example, a polynomial equation of degree 2 is a quadratic equation, while a polynomial equation of degree 3 is a cubic equation.

Here's an example to illustrate Polynomial Regression:



Suppose we have a dataset of student scores in a math exam and the number of hours they studied. We want to predict the student's score based on the number of hours studied. We suspect that the relationship between the number of hours studied and the score is not linear. Therefore, we use Polynomial Regression to model the relationship.

To perform Polynomial Regression, we fit a polynomial equation to the data. The polynomial equation is of the form:

y = b0 + b1x + b2x^2 + b3x^3 + ... + bnx^n + ε

where y is the dependent variable (score), x is the independent variable (number of hours studied), b0, b1, b2, b3, ..., bn are the coefficients of the polynomial equation, ε is the error term, and n is the degree of the polynomial.

The goal of Polynomial Regression is to find the values of the coefficients that best fit the data. This is done by minimizing the sum of squared errors between the predicted values and the actual values of the dependent variable.

Now, let's discuss the difference between Polynomial Regression and Multiple Linear Regression.

Multiple Linear Regression is a regression analysis technique that models the relationship between a dependent variable and two or more independent variables. It assumes that the relationship between the dependent variable and the independent variable(s) is linear. Therefore, it cannot capture the nonlinear relationships between the variables.

On the other hand, Polynomial Regression is a regression analysis technique that models the relationship between a dependent variable and one or more independent variables using a polynomial equation. It can capture the nonlinear relationships between the variables.

In Multiple Linear Regression, the relationship between the dependent and independent variable(s) is modeled using a straight line.

In Polynomial Regression, the relationship between the dependent variable and the independent variable(s) is modeled using a curved line.

In summary, Polynomial Regression is a powerful regression analysis technique that can capture the nonlinear relationships between the dependent variable and independent variable(s). It is different from Multiple Linear Regression, which assumes a linear relationship between the dependent variable and independent variable(s).

UNIT - 4

Que1. Why is visualization used in machine learning? Discuss its importance in the data exploration phase of ML.

- ➢ Exploring and comprehending data with visualization is a key component of machine learning. It enables data scientists to understand more about the underlying structures, correlations, and patterns of the data, which can help them create machine learning models that work well.
- ➢ Finding significant characteristics and patterns that may be applied to the construction of predictive models is one of the main objectives of data exploration in machine learning. This procedure is made easier by visualization, which enables data scientists to visually check the data and spot any outliers, missing numbers, or other abnormalities that could have an impact on the data's quality.
- ➢ Understanding the distribution of the data is another benefit of visualization and is crucial for choosing the right machine learning algorithms. For instance, linear regression models may be effective for data that is regularly distributed, but non-linear techniques like decision trees or neural networks may be more suitable for data that is not linear.
- ➢ Additionally, feature selection—the process of deciding which features are most crucial to the model's ability to predict outcomes—can be aided by visualization. Data scientists can distinguish between characteristics that strongly correlate with

the target variable and those that do not by visualizing the correlations between various features.

- Last but not least, visualization may be used to explain machine learning model findings to stakeholders who may not be aware of the algorithmic specifics. Stakeholders can better comprehend the insights and forecasts produced by the model by presenting the results in an attractive and understandable style.
- In conclusion, visualization is an essential tool for machine learning's data exploration stage. It enables data scientists to understand the data, spot trends and connections, and pick the best machine learning algorithms. Data scientists may create more precise and efficient machine learning models that can lead to improved decision-making and better commercial results by utilizing the power of visualization.

Que 2. What do you understand by "Curse of Dimensionality"? Does proper Feature Selection help in avoiding this? Please discuss in brief.

- The phenomenon known as the "Curse of Dimensionality" describes how machine learning algorithms perform worse as the dataset's number of features (or dimensions) grows. The number of potential feature combinations exponentially increases in high-dimensional datasets, resulting in sparse data and overfitting. The machine learning algorithm may become less accurate and effective as a result.

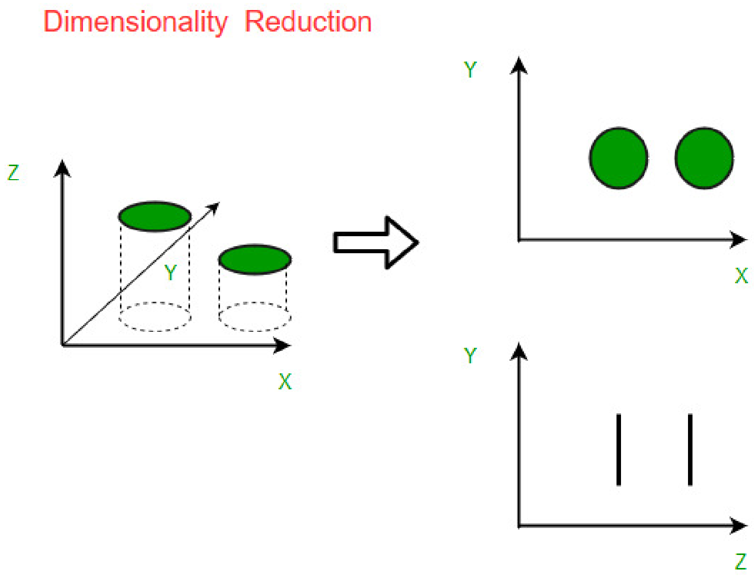
➢ By limiting the amount of characteristics to those that are most important, proper feature selection can assist in avoiding the curse of dimensionality. Techniques for feature selection can be used to find the dataset's most instructive characteristics and eliminate unnecessary or unimportant ones. This can enhance the machine learning algorithm's performance and lower the chance of overfitting.

➢ There are several methods for selecting features, such as filter methods, wrapper methods, and embedding methods. Utilizing statistical approaches, filter algorithms assess each feature's significance and choose the most useful ones. Wrapper approaches assess the usefulness of each feature and choose the most informative ones based on the performance of the machine learning algorithm. Embedded approaches carry out feature selection throughout the machine learning algorithm's training phase.

➢ Since fewer features need less computation and memory, careful feature selection can also aid in lowering the computational complexity of the machine learning method. As a result, the algorithm may train and test more quickly and adapt to bigger datasets.

➢ In conclusion, the curse of dimensionality, which occurs when the number of features rises, is a typical issue in machine learning. By selecting the most informative characteristics and lowering the chance of overfitting, appropriate feature selection strategies can assist in avoiding this issue. Feature selection can aid in increasing the effectiveness and scalability of the machine learning algorithm by lowering the amount of features.

Que 3. Explain the term Dimensionality Reduction in brief. Also discuss the working of principle.



- The practice of lowering the number of variables or features in a dataset while keeping the most important data is known as dimensionality reduction. To put it another way, it seeks to simplify a dataset's structure while minimizing the impact on its performance. This method is frequently utilized in several disciplines, including pattern recognition, data science, and machine learning, among others.
- The two major methods of feature selection and feature extraction make up the dimensionality reduction operating concept. According to certain criteria, such as their correlation with the target variable, their significance in explaining the

variation in the data, or their mutual information, the most pertinent features from a dataset are chosen through the process of feature selection. Contrarily, feature extraction entails projecting the original features onto a new set of coordinates that capture the most significant patterns or structures in the data in order to convert them into a lower-dimensional space. There are two components of dimensionality reduction:

- Feature selection: In order to acquire a smaller subset that may be utilized to model the problem, we aim to locate a subset of the original collection of variables, or features. Typically, there are three approaches:
    1. Filter
    2. Wrapper
    3. Embedded
- Feature extraction: This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser no. of dimensions.
- The various methods used for dimensionality reduction include:
- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Generalized Discriminant Analysis (GDA)

Que 4. Component Analysis Algorithm (PCA).

- A common statistical method used to lessen the dimensionality of a dataset is principal component analysis (PCA). By determining the most significant variables or features that contribute to the variance in the data, it is an unsupervised learning approach that aims to uncover the underlying structure of the data.

- The fundamental goal of PCA is to convert the original dataset into a new collection of variables, referred to as principal components, that encompass the broadest range of data variance. The linear combination of the initial variables that captures the greatest amount of data variation is referred to as the first principal component. Subject to the restriction that it be uncorrelated with all the preceding principle components, each succeeding principal component is defined as the linear combination of the initial variables that captures the greatest amount of variation in the data.
- The PCA algorithm consists of the following steps: Standardize the data: PCA requires that the data be standardized so that each variable has a mean of 0 and a standard deviation of 1.
    - Compute the covariance matrix: Compute the covariance matrix of the standardized data. The covariance matrix gives us information about how the variables are related to each other.
    - Compute the eigenvectors and eigenvalues of the covariance matrix: The eigenvectors and eigenvalues of the covariance matrix give us information about the directions and magnitudes of the principal components.
    - Select the principal components: Choose the principal components based on the eigenvalues, which represent

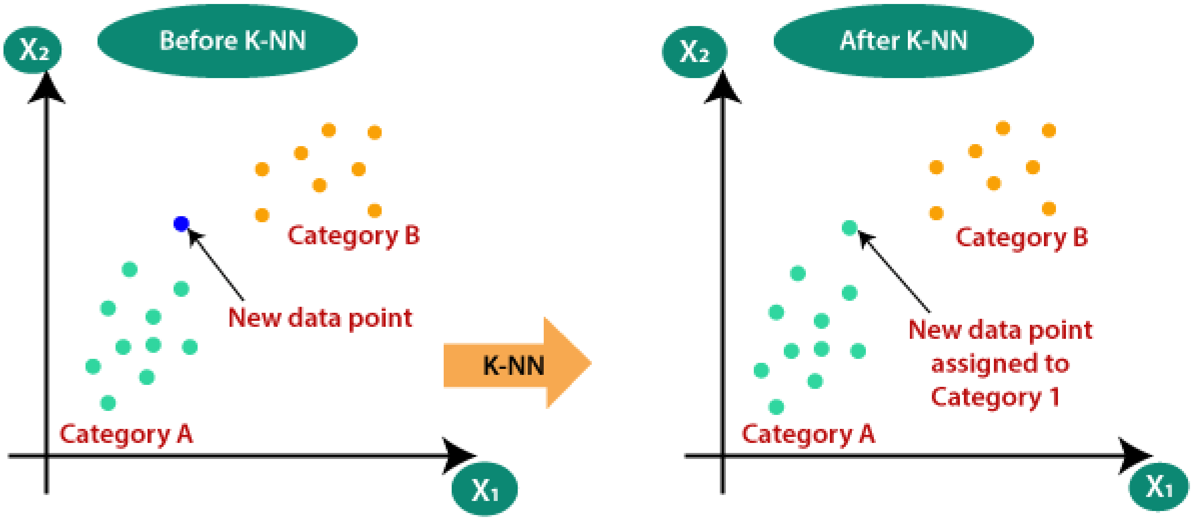
the amount of variance in the data that is captured by each principal component.

- Compute the transformed data: Transform the original data into the new coordinate system defined by the principal components.
- PCA is a powerful technique that can be used for a wide range of applications, including data compression, feature extraction, and data visualization. However, it is important to note that PCA assumes that the underlying data is linearly related, and may not be appropriate for datasets with complex nonlinear relationships. Additionally, the interpretability of the principal components can be challenging, and it is important to carefully consider the meaning of the results before drawing conclusions.

Que 5. Explain K Nearest Neighbours (KNN) and its process in detail using a suitable diagram.

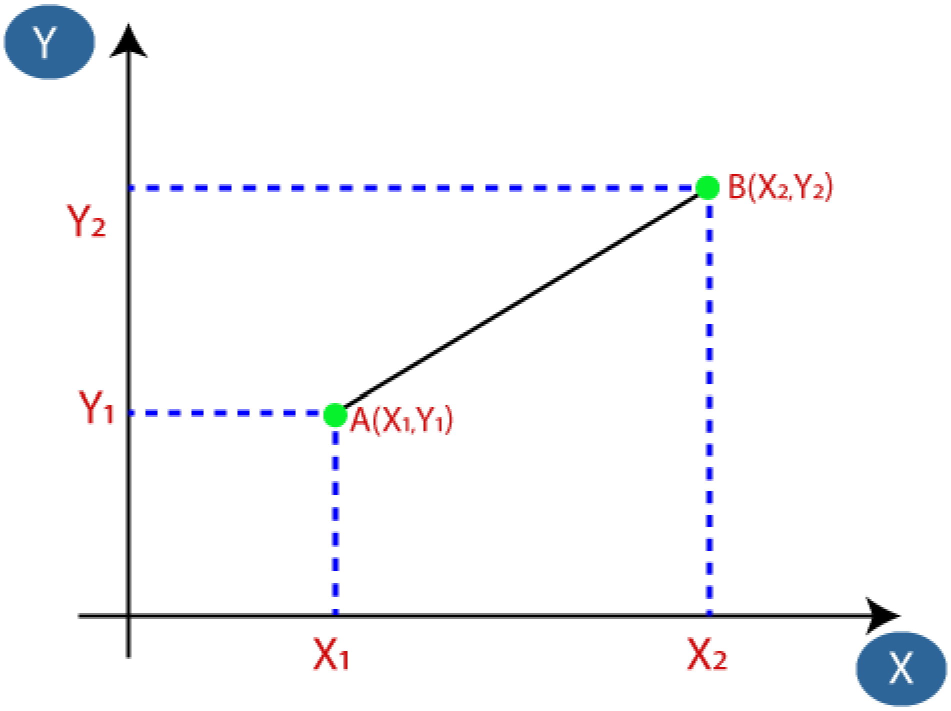
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- The K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.

- The K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- The KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.



- The K-NN working can be explained on the basis of the below algorithm:
    - Select the number K of the neighbors.
    - Calculate the Euclidean distance of K number of neighbors.

    - Take the nearest K neighbor as per the Euclidean distance.
    - Among these K neighbors, count the number of data points in each category.
    - Assign the new data points to that category for which the number of the neighbor is maximum.
    - Our model is ready.



Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

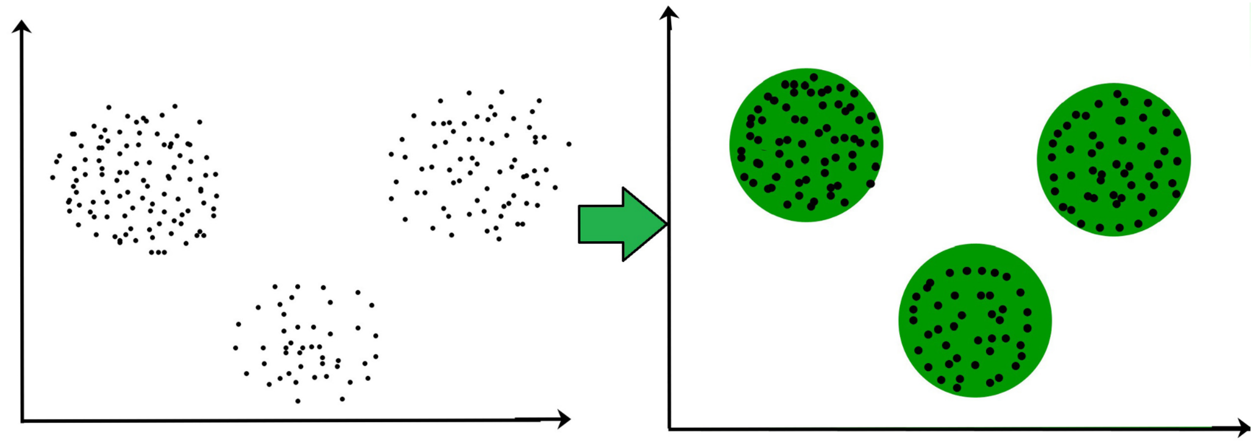
Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

UNIT - 5

Que 1. What do you understand by clustering in ML? Discuss circumstances, when it is applicable to use for ML model development.

- Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.
- For ex– The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.



Clustering Methods:

- **Density-Based Methods**: These methods consider the clusters as the dense region having some similarities and differences from the lower dense region of the space. These methods have good accuracy and the ability to merge two clusters. Example DBSCAN (Density-Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points to Identify Clustering Structure), etc.
- **Hierarchical Based Methods**: The clusters formed in this method form a tree-type structure based on the hierarchy. New clusters are formed using the previously formed one. It is divided into two category:
  - **Agglomerative** (bottom-up approach)
  - **Divisive** (top-down approach)
- **Partitioning Methods**: These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major

  parameter example K-means, CLARANS (Clustering Large Applications based upon Randomized Search), etc.
  - **Grid-based Methods**: In this method, the data space is formulated into a finite number of cells that form a grid-like structure. All the clustering operations done on these grids are fast and independent of the number of data objects, for example STING (Statistical Information Grid), wave cluster, CLIQUE (CLustering In Quest), etc.

Applications of clustering:

- Marketing: It can be used to characterize & discover customer segments for marketing purposes.
- Biology: It can be used for classification among different species of plants and animals.
- Libraries: It is used in clustering different books on the basis of topics and information.
- Insurance: It is used to acknowledge the customers, their policies and identify the frauds.
- City Planning: It is used to make groups of houses and to study their values based on their geographical locations and other factors present.
- Earthquake studies: By learning the earthquake-affected areas we can determine the dangerous zones.

Que 2. Explain working of K-Means using a suitable diagram.

- The goal of clustering is to divide the population or set of data points into a number of groups so that the data points within each group are more comparable to one another and different from the data points within the other groups. It is essentially a grouping of things based on how similar and different they are to one another.

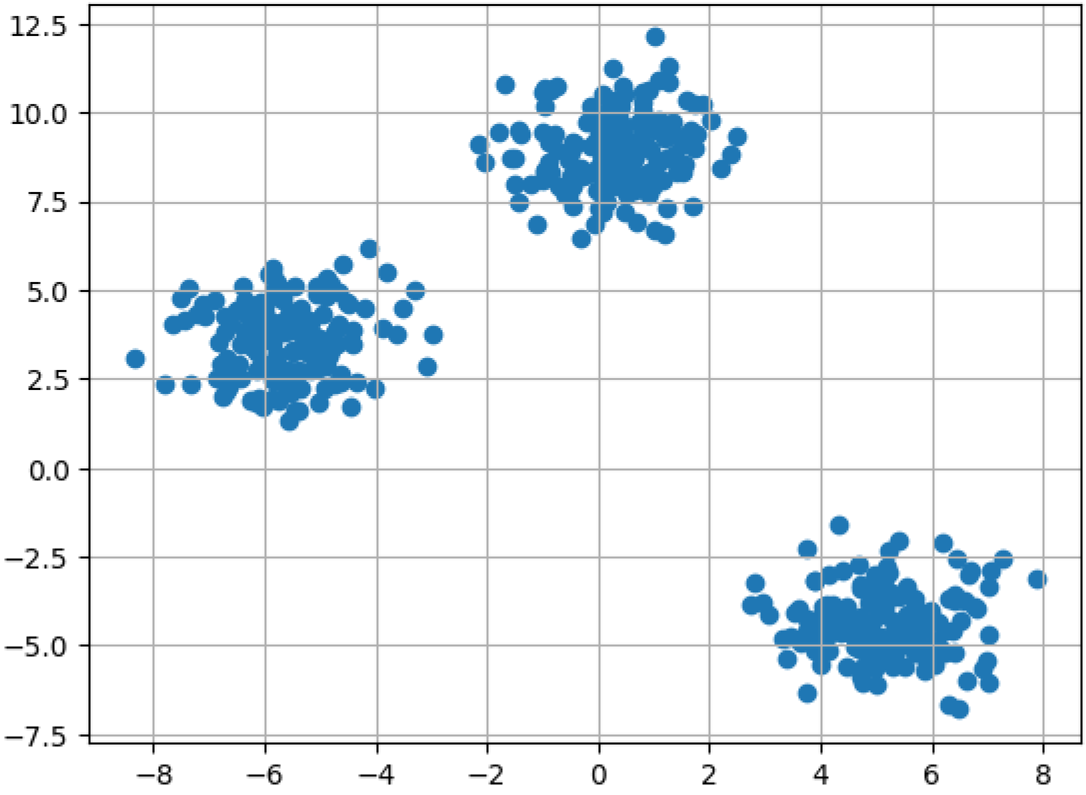
- We are given a data set of items, with certain features, and values for these features (like a vector). The task is to categorize those items into groups. To achieve this, we will use the K-means algorithm; an unsupervised learning algorithm. ‘K’ in the name of the algorithm represents the number of groups/clusters we want to classify our items into.
- The algorithm works as follows:
    - First, we randomly initialize k points, called means or cluster centroids.
    - We categorize each item to its closest mean and we update the mean’s coordinates, which are the averages of the items categorized in that cluster so far.
    - We repeat the process for a given number of iterations and at the end, we have our clusters.

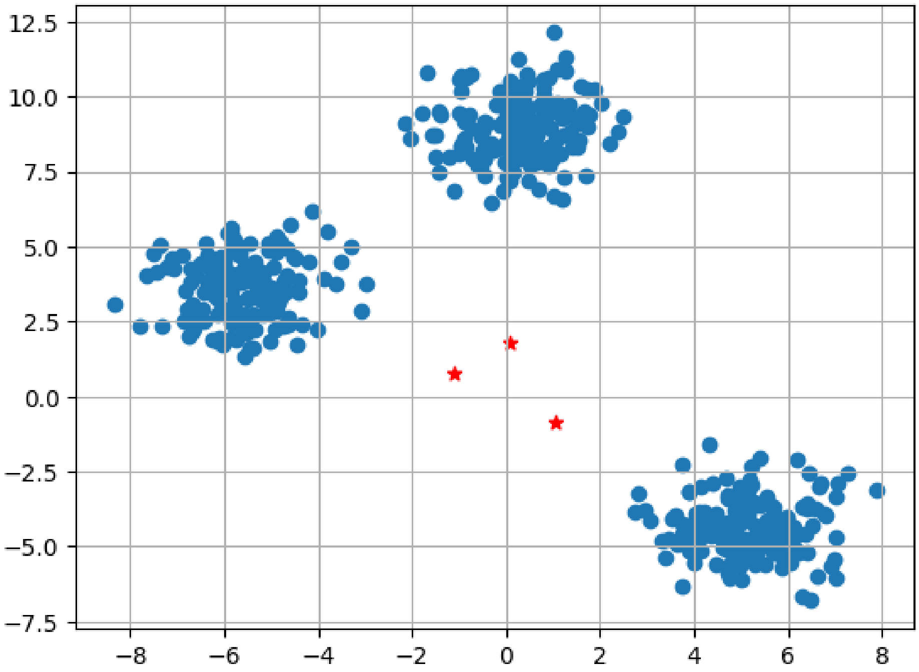
- Pseudocode:

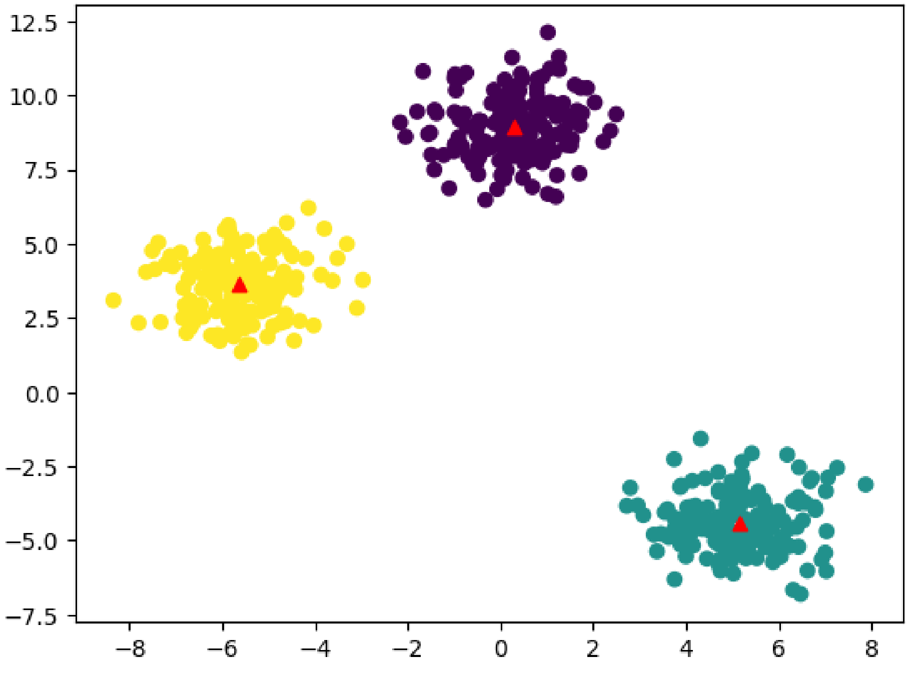
```
Initialize k means with random values

--> For a given number of iterations:

    --> Iterate through items:

        --> Find the mean closest to the item by calculating
        the euclidean distance of the item with each of the means
```

--> Assign item to mean

--> Update mean by shifting it to the average of the items in that cluster.

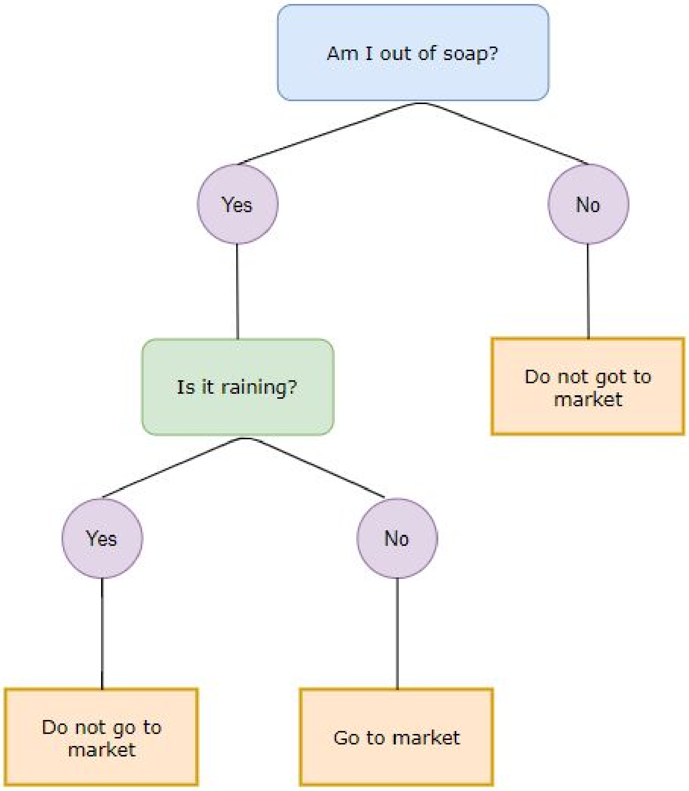



12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−8
−6
−4
−2
0
2
4
6
8


12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−8
−6
−4
−2
0
2
4
6
8

Que 3. Explain working of Decision Tree based machine learning algorithm using a suitable example.

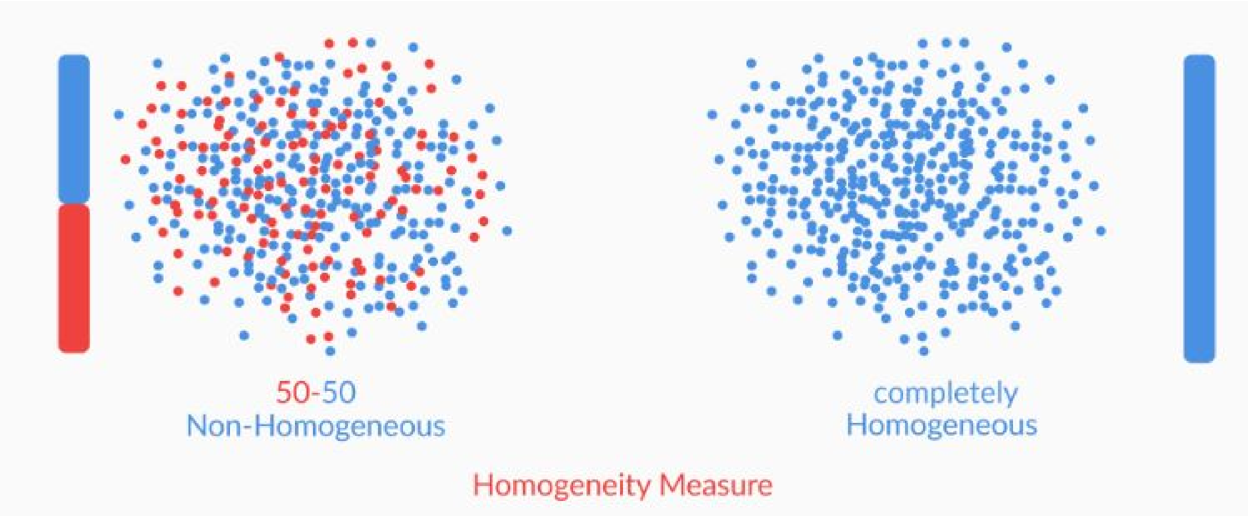
- A decision tree is a type of supervised learning algorithm that is commonly used in machine learning to model and predict outcomes based on input data. It is a tree-like structure where each internal node represents a decision or test on a specific feature or attribute, each branch represents the outcome of that decision, and each leaf node represents the final decision or prediction.
- Decision trees can be considered a set of if-then-else statements. In other words, the process of making decisions involves asking questions with two or more possible outcomes. Let’s understand this decision-making process with the help of a simple decision tree example shown in the image given below.



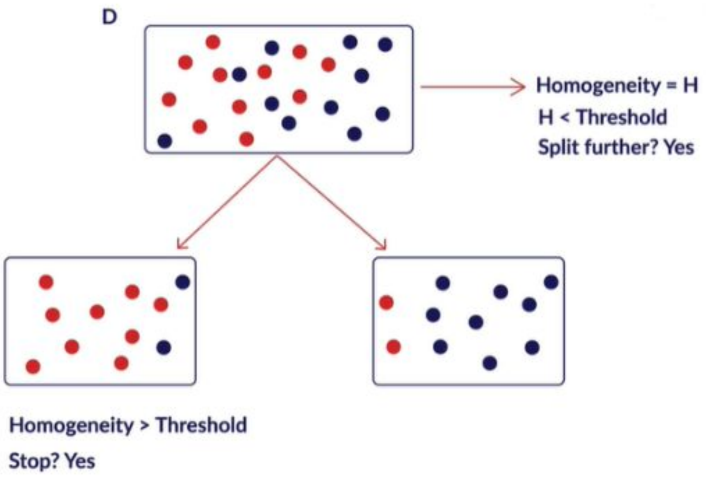
Constructing a decision tree involves following steps:

- Recursive binary splitting/partitioning the data into smaller subsets
- Selecting the best rule from a variable/ attribute for the split
- Applying the split based on the rules obtained from the attributes
- Repeating the process for the subsets obtained
- Continuing the process until the stopping criterion is reached
- Assigning the majority class/average value as the prediction

- In order to construct a decision tree, you must know how to select the node that will lead to the best possible solution. Homogeneity/Purity is one of the major factors while constructing a decision tree.



- Consider a data set 'D' with homogeneity 'H' and a defined threshold value. When homogeneity exceeds the threshold value, you need to stop splitting the node and assign the prediction to it. As this node does not need further splitting, it becomes the leaf node.



- Till the homogeneity 'H' is less than the threshold, you need to continue splitting the node. The process of splitting needs to be continued until homogeneity exceeds the threshold value and the majority data points in the node are of the same class.

Que 4. Discuss the process of Random Forest algorithm in detail.

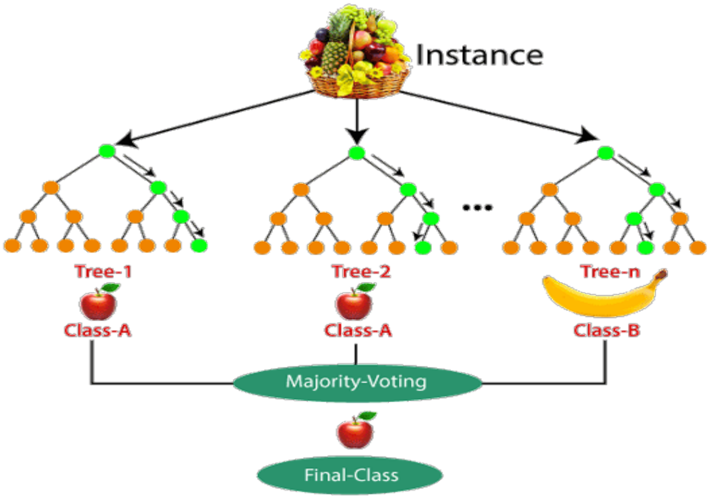
- Random Forest is one of the most popular and commonly used algorithms by Data Scientists. Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.
- One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and

categorical variables, as in the case of classification. It performs better for classification and regression tasks.

- Ensemble simply means combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.
- Ensemble uses two types of methods:
  - Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
  - Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XGBOOST.

Steps involved in Random Forest Algorithm:

1. In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.
2. Individual decision trees are constructed for each sample.
3. Each decision tree will generate an output.
4. Final output is considered based on Majority Voting or Averaging for Classification and regression, respectively.



Important features of Random Forest-

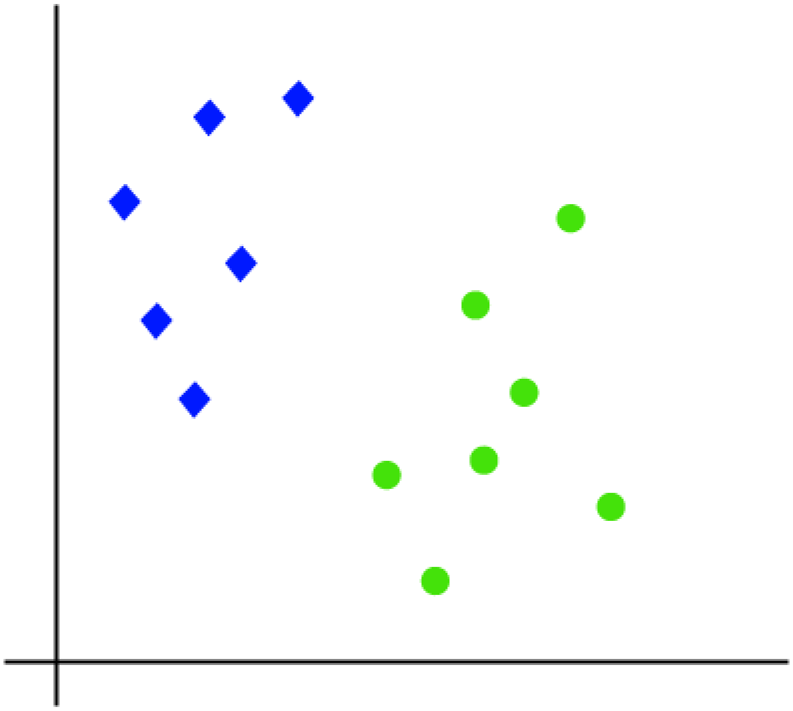
- Diversity: Not all attributes/variables/features are considered while making an individual tree; each tree is different.
- Immune to the curse of dimensionality: Since each tree does not consider all the features, the feature space is reduced.
- Parallelization: Each tree is created independently out of different data and attributes. This means we can fully use the CPU to build random forests.
- Train-Test split: In a random forest, we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- Stability: Stability arises because the result is based on majority voting/ averaging.

Que 5. Explain Support Vector Machine (SVM) algorithm with a suitable diagram.

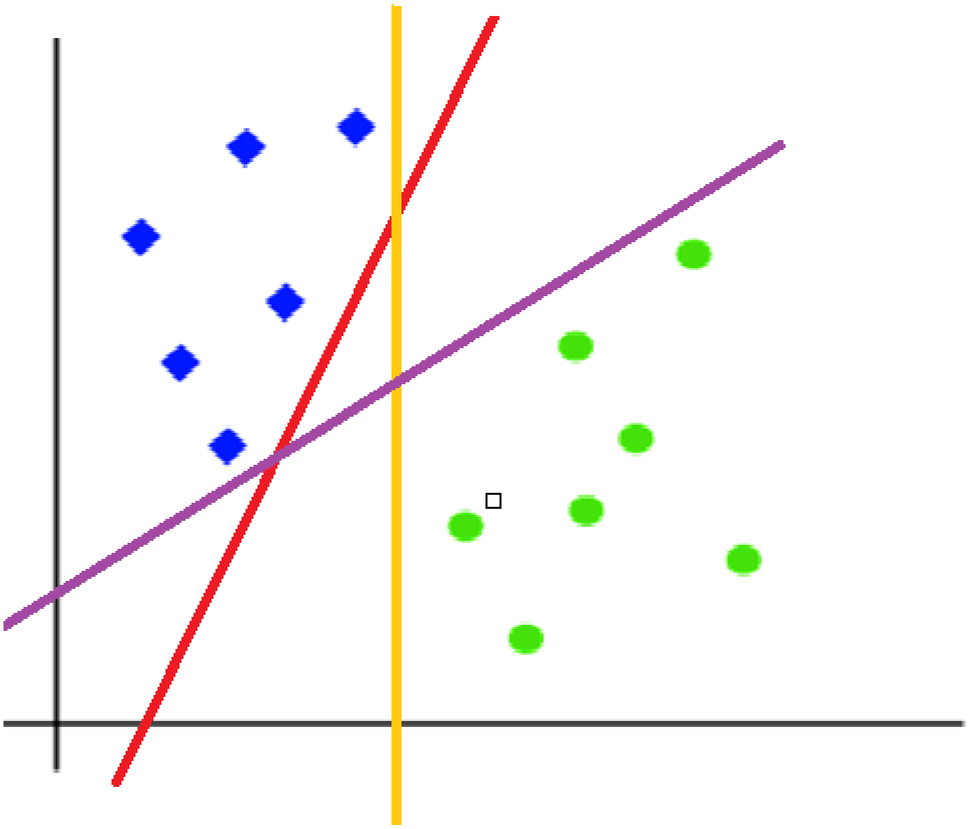
- Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it’s best suited for classification. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.
- The SVM kernel is a function that takes low-dimensional input space and transforms it into higher-dimensional space, i.e. it converts non separable problems to separable problems. It is mostly useful in non-linear separation problems. Simply put, the kernel does some extremely complex data transformations and then finds out the process to separate the data based on the labels or outputs defined.
- Types of SVM:
    - Linear SVM: When the data is perfectly linearly separable only then we can use Linear SVM. Perfectly linearly separable means that the data points can be classified into 2 classes by using a single straight line(if 2D).
    - Non-Linear SVM: When the data is not linearly separable then we can use Non-Linear SVM, which means when the data points cannot be separated into 2 classes by using a straight line (if 2D) then we use some advanced techniques like kernel tricks to classify them. In most

real-world applications we do not find linearly separable data points hence we use kernel trick to solve them.

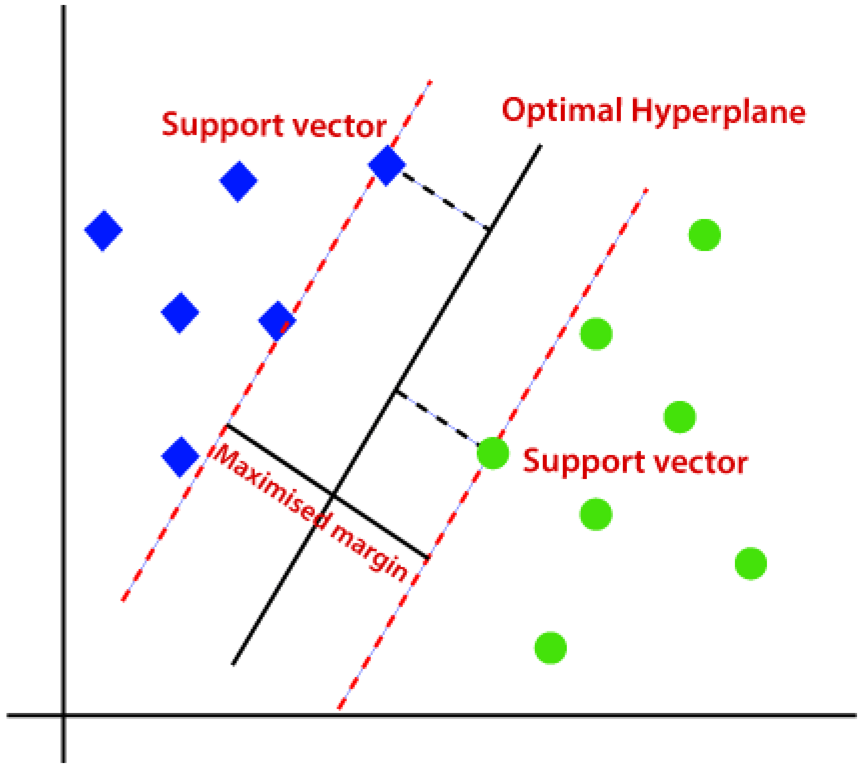
- SVM is defined such that it is defined in terms of the support vectors only, we don't have to worry about other observations since the margin is made using the points which are closest to the hyperplane (support vectors), whereas in logistic regression the classifier is defined over all the points. Hence SVM enjoys some natural speed-ups. Let's understand the working of SVM using an example. Suppose we have a dataset that has two classes (green and blue). We want to classify the new data point as either blue or green.



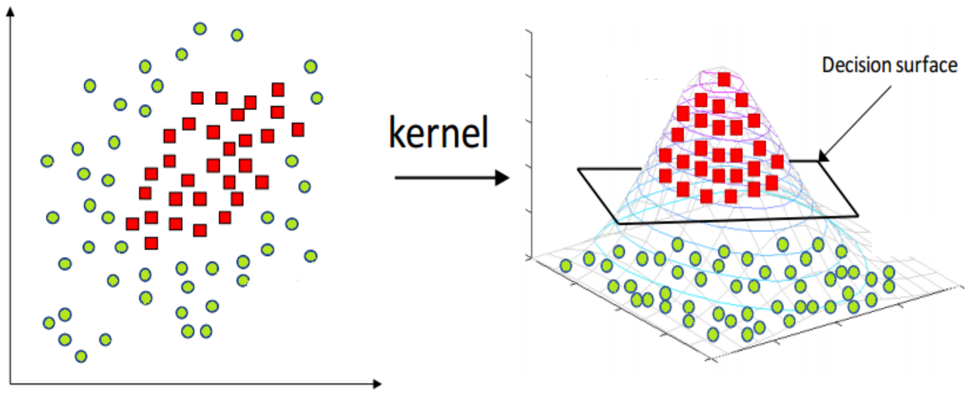
- To classify these points, we can have many decision boundaries, but the question is which is the best and how do we find it? NOTE: Since we are plotting the data points in a 2-dimensional graph we call this decision boundary a straight line but if we have more dimensions, we call this decision boundary a “hyperplane”.



- The best hyperplane is that plane that has the maximum distance from both the classes, and this is the main aim of SVM. This is done by finding different hyperplanes which classify the labels in the best way then it will choose the one which is farthest from the data points or the one which has a maximum margin.



- So what we do is try converting this lower dimension space to a higher dimension space using some quadratic functions which will allow us to find a decision boundary that clearly divides the data points. These functions which help us do this are called Kernels and which kernel to use is purely determined by hyperparameter tuning.


kernel
Decision surface