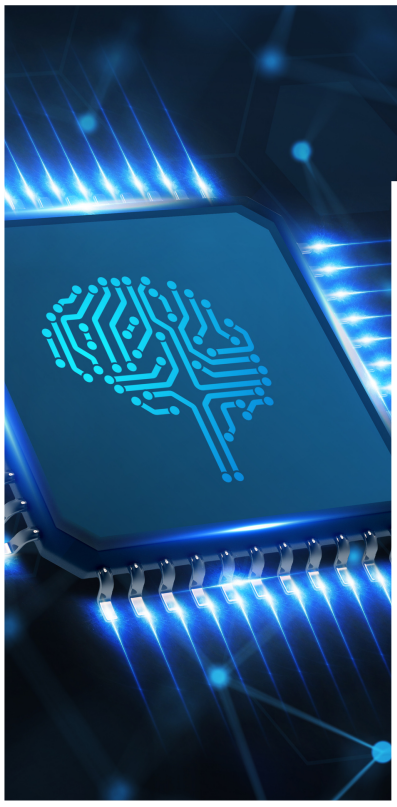


TOP 100

Machine Learning

Interview Questions

with explanatory youtube links

1) What is Machine Learning?

ML is the art of granting machines the ability to think. It is a field of computer science that uses statistical techniques to give machines the ability to learn without being explicitly programmed. Machine Learning deals with building algorithms that can receive input data, perform statistical analysis to predict output, and update the output as newer data become available.

2) What are some use cases of Machine Learning from our daily uses?

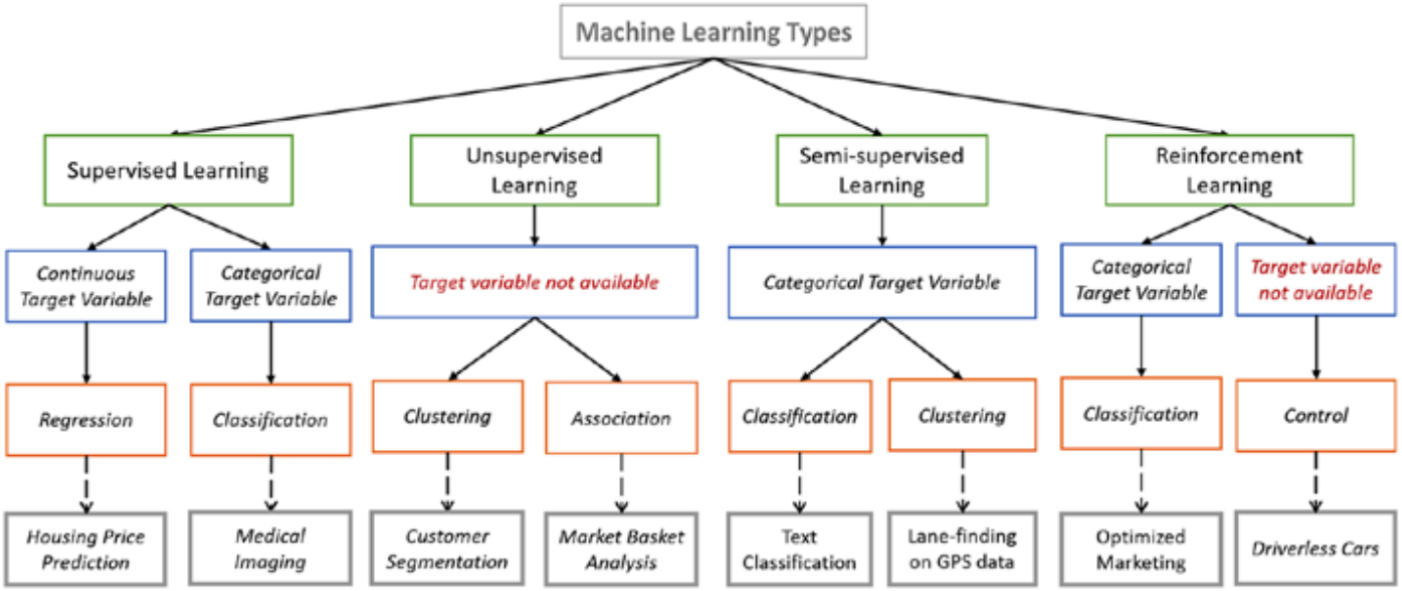
- YouTube/Netflix/Amazon Prime: Recommends us videos or movies or series based on our interest.
- Gmail: Classifies spam emails on your behalf
- Banks: Detect Fraudulent/Anomalous behavior and holds the transaction. It also helps in Fraud Detection.
- Voice Assistance: Alexa/Siri use Machine Learning to response to the questions asked.

There are Machine Learning use cases in almost every industry today.

3) What are different kinds of Machine Learning? What does that signify?

The three kinds of learning in Machine Learning are:

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning



4) What are different steps in Machine Learning?

We perform below steps in Machine Learning:

- Collect data
- Filter data
- Analyse data
- Train algorithms
- Test algorithms
- Use algorithms for future predictions

5) What is Bias and Variance? How can we have optimum of both?

- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

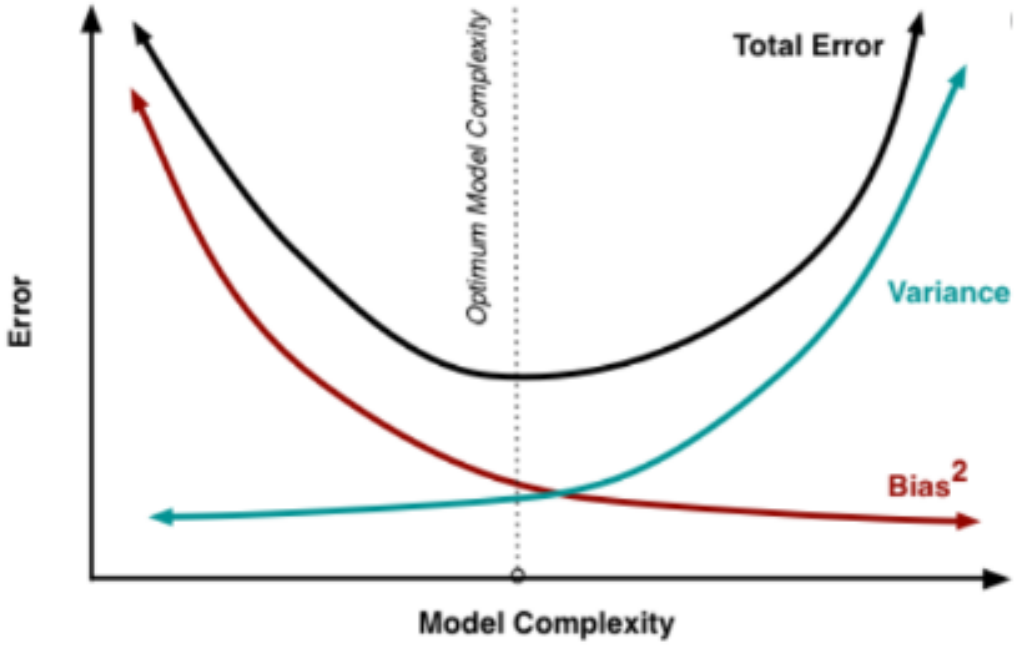
- Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.

In supervised learning, underfitting happens when a model unable to capture the underlying pattern of the data. These models usually have high bias and low variance. It happens when we have very less amount of data to build an accurate model or when we try to build a linear model with a nonlinear data.

Also, these kinds of models are very simple to capture the complex patterns in data like Linear and logistic regression.In supervised learning over fitting happens when our model captures the noise along with the underlying pattern in data.

It happens when we train our model a lot over noisy dataset. These models have low bias and high variance. These models are very complex like Decision trees which are prone to over fitting.

Bias Variance Trade-off:

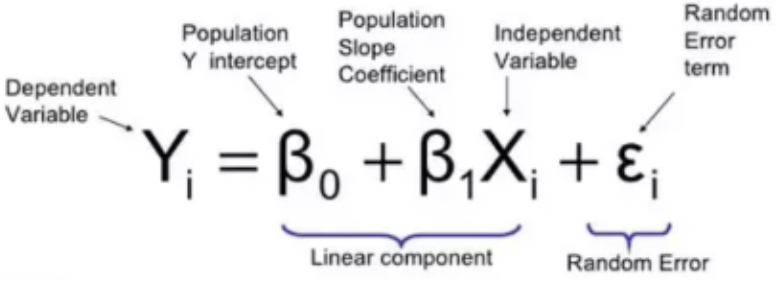


For any supervised algorithm, having a high bias error usually means it has low variance error and vice versa. To be more specific, parametric or linear ML algorithms often have a high bias but low variance. On the other hand, non-parametric or non-linear algorithms have vice versa.
The goal of any ML model is to obtain a low variance and a low bias state, which is often a task due to the parametrization of machine learning algorithms. Common ways to achieve optimum Bias and Variance are:

1. By minimizing total error
2. Using Bagging and resampling techniques
3. Adjusting minor values in Algorithms

6) What is Linear Regression?

Linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables or independent variables.
The case of one explanatory variable is called simple linear regression, for more than one explanatory variable, the process is called multiple linear regression.
A linear regression line has an equation of the form



7) What are assumptions of Linear Regression?

Short Trick: Assumptions can be abbreviated as LINE in order to remember.

L : Linearity (Relationship between x and y is linear)
I : Independence (Observations are independent of each other)
N : Normality (for any fix value of x, y is normally distributed)
E : Equal Variance (homoscedasticity)

Long Answer: There are three main assumptions in a linear regression model:

1. The assumption about the Relationship of dependent and independent variable: there should be a linear relationship between the dependent and independent variables. It is known as the 'linearity assumption'.
2. Assumptions about the **residuals**:

- Normality assumption: It is assumed that the error terms, ε(i), are normally distributed.
  (Explanation: If the residuals are not normally distributed, their randomness is lost, which implies that the model is not able to explain the relation in the data.)
- Zero mean assumption: It is assumed that the residuals have a mean value of zero.
  (Explanation: Zero conditional mean is there which says that there are both negative and positive errors which cancel out on an average. This helps us to estimate dependent variable precisely.)
- Constant variance assumption: It is assumed that the residual terms have the same (but unknown) variance, σ2 This assumption is also known as the assumption of homogeneity or **homoscedasticity**.
- Independent error assumption: It is assumed that the residual terms are independent of each other, i.e. their pair-wise covariance is zero.
  (Explanation: because the observations with larger errors will have more pull or influence on the fitted model.)

3. Assumptions about the **estimators**:

- The independent variables are measured without error.
- The independent variables are linearly independent of each other, i.e. there is no multicollinearity in the data.
  (Explanation: If the independent variables are not linearly independent of each other, the uniqueness of the least squares solution (or normal equation solution) is lost.)

8) What is Regularization? Explain different types of Regularizations.

Regularization is a technique which is used to solve the overfitting problem of the machine learning models.
The types of Regularization are as follows:

- The L1 regularization (also called Lasso)
- The L2 regularization (also called Ridge)T
- The L1/L2 regularization (also called Elastic net)

L1 Regularization: L1 Regularization or Lasso Regularization adds a penalty to the error function. The penalty is the sum of the **absolute** values of weights.
Where λ is the tuning parameter called regularization parameter which decides how much we want to penalize the model. The term for L1 regularization is highlighted below with red box.

L1 regularization on least squares:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

L2 Regularization: L2 Regularization or Ridge Regularization also adds a penalty to the error function. But the penalty here is the sum of the **squared** values of weights.
Where λ is the tuning parameter called regularization parameter which decides how much we want to penalize the model. The term for L2 regularization is highlighted below with red box.

L2 regularization on least squares:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

Elastic-net Regularization: Elastic-net is a mix of **both L1 and L2 regularizations**. A penalty is applied to the sum of the absolute values and to the sum c

$$\frac{\sum_{i=1}^{n} (y_i - x_i^J \hat{\beta})^2}{2n} + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^{m} \hat{\beta}_j^2 + \alpha \sum_{j=1}^{m} |\hat{\beta}_j| \right)$$

Lambda is a shared penalization parameter while alpha sets the ratio between L1 and L2 regularization in the Elastic Net Regularization. Hence, we expect a hybrid behaviour between L1 and L2 regularization.

9) How to choose the value of the regularisation parameter (λ)?

Selecting the regularisation parameter is a tricky business. If the value of λ is too high, it will lead to extremely small values of the regression coefficient β, which will lead to the model underfitting (high bias – low variance).

On the other hand, if the value of λ is 0 (very small), the model will tend to overfit the training data (low bias – high variance).

There is no proper way to select the value of λ. What you can do is have a sub-sample of data and run the algorithm multiple times on different sets. Here, the person has to decide how much variance can be tolerated. Once the user is satisfied with the variance, that value of λ can be chosen for the full dataset.

One thing to be noted is that the value of λ selected here was optimal for that subset, not for the entire training data.

10) Explain gradient descent? 

Definition: Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function (cost).

(Note: Cost function is the average of the loss functions for all the training examples.)

When it is used: Gradient descent is best used when the parameters cannot be calculated analytically (e.g. using linear algebra) and must be searched for by an optimization algorithm.Details: The goal of any Machine Learning Model to minimise the cost function. To get the minima of the cost function we use Gradient Descent Algorithm.

Without going too much mathematical, let's see the steps involved in Gradient Descent.

Step 1: Initializing the coefficient of the function. Initial value should either be 0 or very small value.

coefficient = 0.0

Step2: For the coefficient we have chosen in step 1, we will calculate the cost by putting it in function.

cost = f(coefficient)

Step 3: Find the derivative(delta) of the cost or the derivative of the function

delta = derivative(cost)

Step 4: Now that we know from the derivative which direction is downhill, we can now update the coefficient values. A Learning Rate (alpha) must be specified that controls how much the coefficients can change on each update.

coefficient = coefficient – (alpha * delta)

This process is repeated until the cost of the coefficients (cost) is 0.0 or close enough to zero to be good enough, or the number of epochs is reached.

11) What is Learning Rate? How to choose the value of the parameter learning rate (α)?

Learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect the loss gradient. The lower the value, the slower we travel along the downward slope.

Selecting the value of learning rate is a tricky business. If the value is too small, the gradient descent algorithm takes ages to converge to the optimal solution. On the other hand, if the value of the learning rate is high, the gradient descent will overshoot the optimal solution and most likely never converge to the optimal solution.

To overcome this problem, you can try different values of alpha over a range of values and plot the cost vs the number of iterations. Then, based on the graphs, the value corresponding to the graph showing the rapid decrease can be chosen.
If you see that the cost is increasing with the number of iterations, your learning rate parameter is high and it needs to be decreased.

12) How to carry out hypothesis testing in linear regression?

Hypothesis testing can be carried out in linear regression for the following purposes:

1. To check whether a predictor is significant for the prediction of the target variable. Two common methods for this are —

- **Using p-values:** If the p-value of a variable is greater than a certain limit (usually 0.05), the variable is insignificant in the prediction of the target variable.
- **By checking the values of the regression coefficient:** If the value of regression coefficient corresponding to a predictor is zero, that variable is insignificant in the prediction of the target variable and has no linear relationship with it.

2. To check whether the calculated regression coefficients are good estimators of the actual coefficients.

13) What is Variance Inflation Factor (VIF)? What is its significance?

Variance Inflation Factor (VIF) is used to detect the presence of multicollinearity. Variance inflation factors (VIF) measure how much the variance of the estimated regression coefficients is inflated as compared to when the predictor variables are not linearly related.
It is obtained by regressing each independent variable, say X on the remaining independent variables (say Y and Z) and checking how much of it (of X) is explained by these variables.
higher the VIF, higher the R-Squared which means the variable X is collinear with Y and Z variables. If all the variables are completely orthogonal, R-Square will be 0 resulting in VIF of 1.
Rule of thumb is to avoid any variable with VIF > 5.

14) How does Residual Plot and Q-Q plot help in linear regression model?

Residual plots and Q-Q plots are used to visually check that your data meets the homoscedasticity and normality assumptions of linear regression.
A residual plot lets you see if your data appears homoscedastic.
Homoscedasticity means that the residuals, the difference between the observed value and the predicted value, are equal across all values of your predictor variable.

If your data are homoscedastic then you will see the points randomly scattered around the x axis. If they are not (e.g. if they form a curve, bowtie, fan etc.) then it suggests that your data doesn't meet the assumption.

Q-Q plots let you check that the data meet the assumption of normality. They compare the distribution of your data to a normal distribution by plotting the quartiles of your data against the quartiles of a normal distribution. If your data are normally distributed, then they should form an approximately straight line.

15) What is difference between R-Squared and Adjusted R Squared?

- Short Answer: In case of adjusted R2 we include only the important and useful variables, whereas in R2 all the variables are included.

- Long Answer: R-squared or R2 explains the degree to which your input variables explain the variation of your output / predicted variable. So, if R-square is 0.8, it means 80% of the variation in the output variable is explained by the input variables. So, in simple terms, higher the R squared, the more variation is explained by your input variables and hence better is your model.
  However, the problem with R-squared is that it will either stay the same or increase with addition of more variables, even if they do not have any relationship with the output variables. This is where “Adjusted R square” comes to help. Adjusted R-square penalizes you for adding variables which do not improve your existing model. Adjusted R2 also explains the degree to which your input variables explain the variation of your output / predicted variable but adjusts for the number of terms in a model. If you add more and more useless variables to a model, adjusted r-squared will decrease. If you add more useful variables, adjusted r-squared will increase.Adjusted R2 will always be less than or equal to R2. So, basically in adjusted R2 we include only the important and useful variables, whereas in R2 all the variables are included.
  Hence, if you are building Linear regression on multiple variable, it is always suggested that you use Adjusted R-squared to judge goodness of model. In case you only have one input variable, R-square and Adjusted R squared would be exactly same.Typically, the more non-significant variables you add into the model, the gap in R-squared and Adjusted R-squared increases.

16) What are forecast KPI's or metrics? Explain Bias RMSE, MAE, MAPE.

Forecast KPI's are used to evaluate forecast accuracy. The several KPI's for it is as follows:

Bias: Bias represents the historical average error. Basically, will your forecasts be on average too high (i.e. you overshot the demand) or too low (i.e. you undershot the demand)? This will give you the overall direction of the error.

$$bias = \frac{1}{n}\sum_{n} e_t$$

Mean Absolute Percentage Error MAPE is the sum of the individual absolute errors divided by the demand (each period separately). Actually, it is the average of the percentage errors.

Issue with MAPE: MAPE divides each error individually by the demand, so it is skewed: high errors during low-demand periods will have a major impact on MAPE. Due to this, optimizing MAPE will result in a strange forecast that will most likely undershoot the demand. Just avoid it.

The Mean Absolute Error (MAE) is a very good KPI to measure forecast accuracy. As the name implies, it is the mean of the absolute error.

$$MAE = \frac{1}{n}\sum |e_t|$$

Issue with MAE: One of the first issues of this KPI is that it is not scaled to the average demand. If one tells you that MAE is 10 for a particular item, you cannot know if this is good or bad. If your average demand is 1000, it is of course astonishing, but if the average demand is 1, this is a very poor accuracy. To solve this, it is common to divide MAE by the average demand to get a %

$$MAE\% = \frac{\frac{1}{n}\sum |e_t|}{\frac{1}{n}\sum d_t} = \frac{\sum |e_t|}{\sum d_t}$$

RMSE: It is defined as the square root of the average squared error.

$$RMSE = \sqrt{\frac{1}{n}\sum e_t^2}$$

Issue with RMSE: Just as for MAE, RMSE is not scaled to the demand. We can then define RMSE% as such,

$$RMSE\% = \frac{\sqrt{\frac{1}{n}\sum e_t^2}}{\frac{\sum d}{n}}$$

(In all above equations, et is an error term)

17) What is Multicollinearity and what to do with it? How to remove multicollinearity?

In regression, "multicollinearity" refers to predictors that are correlated with other predictors. Multicollinearity occurs when your model includes multiple factors that are correlated not just to your response variable, but also to each other. In other words, it results when you have factors that are a bit redundant.

If multicollinearity is a problem in your model -- if the VIF for a factor is near or above 5 -- the solution may be relatively simple. Try one of these:

- Remove highly correlated predictors from the model. If you have two or more factors with a high VIF, remove one from the model. Because they supply redundant information, removing one of the correlated factors usually doesn't drastically reduce the R-squared. Consider using stepwise regression, best subsets regression, or specialized knowledge of the data set to remove these variables. Select the model that has the highest R-squared value.
- Use Principal Components Analysis, regression methods that cut the number of predictors to a smaller set of uncorrelated components.

Linear Regression Code Snippet.

Users > satyam > Desktop > ML_Algos > linear_regression.py

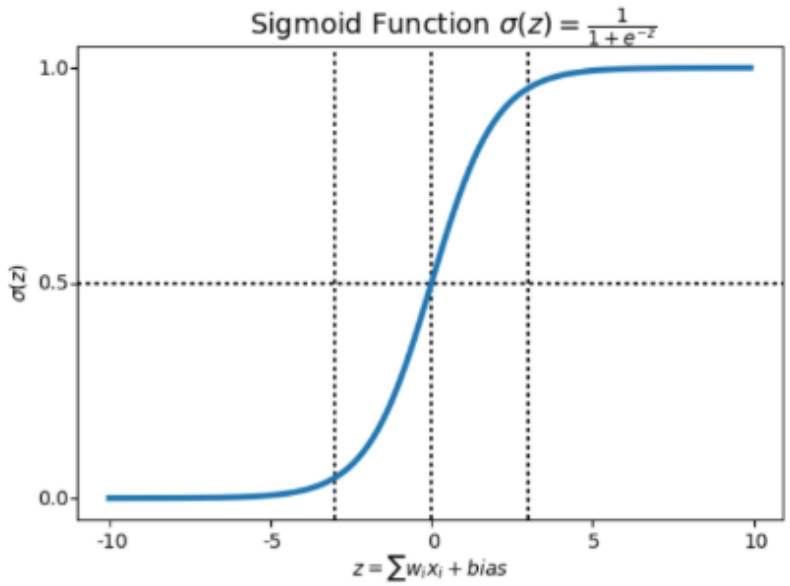
```python
#Author: Dhrub Satyam
#Linear Regression Example.
#Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as seabornInstance
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
%matplotlib inline

#Import Data
dataset = pd.read_csv('/path/to/data/file.csv')

#Dividing into Attribute and Labels
X = dataset['columnX'].values.reshape(-1,1)
y = dataset['columnY'].values.reshape(-1,1)

#Train Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

#Calling the regressor and training the algorithms
regressor = LinearRegression()
regressor.fit(X_train, y_train)

#To retrieve the intercept:
print(regressor.intercept_)
#For retrieving the slope:
print(regressor.coef_)

#Predicting the test data
y_pred = regressor.predict(X_test)

#Printing the Predicted Variable
df = pd.DataFrame({'Actual': y_test.flatten(), 'Predicted': y_pred.flatten()})
print(df)
```

Logistic Regression

18) What is Logistic Regression? What is Logistics Function or Sigmoid Function and What is the range of value of this function?

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The target or dependent variable can only have 2 possible classes (e.g.: Pass/Fail, 0/1, Spam/No-Spam etc.)

Logistics Function/Sigmoid Function: The Logistic Function also called the sigmoid function. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.



1 / (1 + e^-value)

The 'e' in the above equation represents the S-shaped curve that has values between 0 and 1. We write the equation for logistic regression as follows:

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

In the above equation, b0 and b1 are the two coefficients of the input x. We estimate these two coefficients using "maximum likelihood estimation". Below is the maximum log-likelihood cost function for Logistic regression.

$$J(\boldsymbol{w}) = -\sum_i y^{(i)} \log\left(\phi\left(z^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - \phi\left(z^{(i)}\right)\right)$$

To get a better grasp on this cost function, let's look at the cost that wecalculate for one single-sample instance:

$$J\left(\phi(z), y; \boldsymbol{w}\right) = -y \log\left(\phi(z)\right) - (1 - y) \log\left(1 - \phi(z)\right)$$

Looking at the preceding equation, we can see that the first term becomes zero if y = 0, and the second term becomes zero if y =1, respectively:

$$J\left(\phi(z), y; \boldsymbol{w}\right) = \begin{cases} -\log\left(\phi(z)\right) & \text{if } y = 1 \\ -\log\left(1 - \phi(z)\right) & \text{if } y = 0 \end{cases}$$

19) Why is logistics regression despite being a classification algorithm has "Regression" in its name?

Although the dependent variable in logistic regression is binary, the predicted log odds can be any real number between negative infinity and positive infinity. In other words, your predictions are continuous. In order to use logistic regression for classification, you need to convert the log odds to a probability (which is also continuous from 0 to 1) and then set a probability threshold. The choice of threshold is done outside the model.

20) What is Maximum Likelihood?

It is a method in statistics for estimating parameter(s) of a model for given data. The basic intuition behind MLE is that the estimate which explains the data best, will be the best estimator.
The main advantage of MLE is that it has asymptotic property. It means that when the size of the data increases, the estimate converges faster towards the population parameter. We use MLE for many techniques in statistics to estimate parameters. I have explained the general steps we follow to find an estimate for the parameter.

Step 1: *Make an assumption about the data generating function.*

Step 2: *Formulate the likelihood function for the data, using the data generating function.*
The likelihood function is nothing but the probability of observing this data given the parameters (P(D|θ)). The parameters depend on our assumptions and the data generating function.

Step 3: *Find an estimator for the parameter using optimization technique.*
This is done by finding the estimate which maximizes the likelihood function. This is the reason why we name the estimator calculated using MLE as M-estimator.

Example : We have tossed a coin n times and observed k heads. Note that we consider head is success and tail is failure.

Step 1: (Assumption): The coin follows Bernoulli distribution function.

Step 2: (Likelihood Function): The likelihood function is binomial distribution function P(D|θ)) in this case. We need to find out the best estimate for p (Probability of getting head) given that k of n tosses are Heads.

Step 3: (Estimation): M- estimator is

$$P = k/n$$

21) What is odds ratio?

If the probability of something happening is p, the odds-ratio is given by p/(1-p).
Example:
Suppose in a throw of a fair dice, A is the event that either 1 or 2 will surface.

So, p(A) = p = 1/3 and hence odds of A = (1/3)/(2/3) = 1/2

Odds of A is 50%, in this case, which can be explained as the likelihood of the event A is 50% of the likelihood of the complementary event of A.

22) Can we use Linear Regression in place of Logistics Regression for classification?

No. The reasons why linear regressions cannot be used in case of binary classification are as follows:

Distribution of error terms: The distribution of data in case of linear and logistic regression is different. Linear regression assumes that error terms are normally distributed. In case of binary classification, this assumption does not hold true.

Model output: In linear regression, the output is continuous. In case of binary classification, an output of a continuous value does not make sense. For binary classification problems, linear regression may predict values that can go beyond 0 and 1. If we want the output in the form of probabilities, which can be mapped to two different classes, then its range should be restricted to 0 and 1.
As the logistic regression model can output probabilities with logistic/sigmoid function, it is preferred over linear regression.

Variance of Residual errors: Linear regression assumes that the variance of random errors is constant. This assumption is also violated in case of logistic regression.

23) What are the assumptions of Logistics Regression?

1. The logistic regression assumes that there is minimal or no multicollinearity among the independent variables.
2. The Logistic regression assumes that the independent variables are linearly related to the log of odds.
3. The logistic regression usually requires a large sample size to predict properly.
4. The Logistic regression which has two classes assumes that the dependent variable is binary and ordered logistic regression requires the dependent variable to be ordered.
5. The Logistic regression assumes the observations to be independent of each other.

24) What is the decision boundary in case of Logistics Regression?

Decision boundary helps to differentiate probabilities into positive class and negative class.
For Logistic Regression, we only have Linear Decision Boundary. The Boundary or line that separates the two classes are chosen by setting a threshold probability.

Step1 : Sigmoid gave the output in the range 0 to 1.

Step2: Set a threshold probability (assume 0.5)

Step3: Classification will happen by separating the points with probability less than or greater than 0.5

(In the above example, 0.5 is a linear decision boundary).

25) Which cost function is used in Logistics Regression? Why cannot we use MSE (Mean Squared Error) as a cost function in Logistics Regression?

The cost function used in Logistics Regression is Sigmoid Function or Logistic Function.
The main reason not to use the MSE as the cost function for logistic regression is because you don't want your cost function to be non-convex in nature.
If the cost function is not convex then it is difficult for the function to optimally converge.

26) Can logistics regression handle categorical variables directly? If not, then how to use categorical value in Logistics Regression?

The inputs to a logistic regression model need to be numeric. The algorithm cannot handle categorical variables directly. So, they need to be converted into a format that is suitable for the algorithm to process.

The various levels of a categorical variable will be assigned a unique numeric value known as the dummy variable. These dummy variables are handled by the logistic regression model as any other numeric value.

27) How does Logistics Regression deals with Multiclass Classification? What is one-vs-all method?

The most famous method of dealing with multiclass classification using logistic regression is using the one-vs-all approach. Under this approach, a number of models are trained, which is equal to the number of classes. The models work in a specific way.

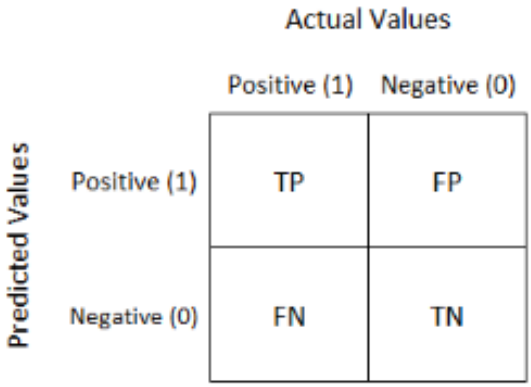
For example, the first model classifies the datapoint depending on whether it belongs to class 1 or some other class; the second model classifies the datapoint into class 2 or some other class. This way, each data point can be checked over all the classes.

28) What is the evaluation matrix for Classification Algorithms?

The different ways are as follows:

- Confusion matrix
- Accuracy
- Precision
- Recall
- Specificity
- F1 score
- Precision-Recall or PR curve
- ROC (Receiver Operating Characteristics) curve
- PR vs ROC curve

Confusion Matrix: it is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



Definition of the terms:

Positive (P): Observation is positive (for example: is an apple).
Negative (N): Observation is not positive (for example: is not an apple).
True Positive (TP): Observation is positive, and is predicted to be positive.
False Negative (FN): Observation is positive, but is predicted negative.
True Negative (TN): Observation is negative, and is predicted to be negative.
False Positive (FP): Observation is negative, but is predicted positive.

Classification Rate/Accuracy: Classification Rate or Accuracy is given by the relation:

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

However, there are problems with accuracy. It assumes equal costs for both kinds of errors. A 99% accuracy can be excellent, good, mediocre, poor or terrible depending upon the problem.

Recall/Sensitivity/True Positive Rate: Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN).

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

Precision: To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labelled as positive is indeed positive (a small number of FP).

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

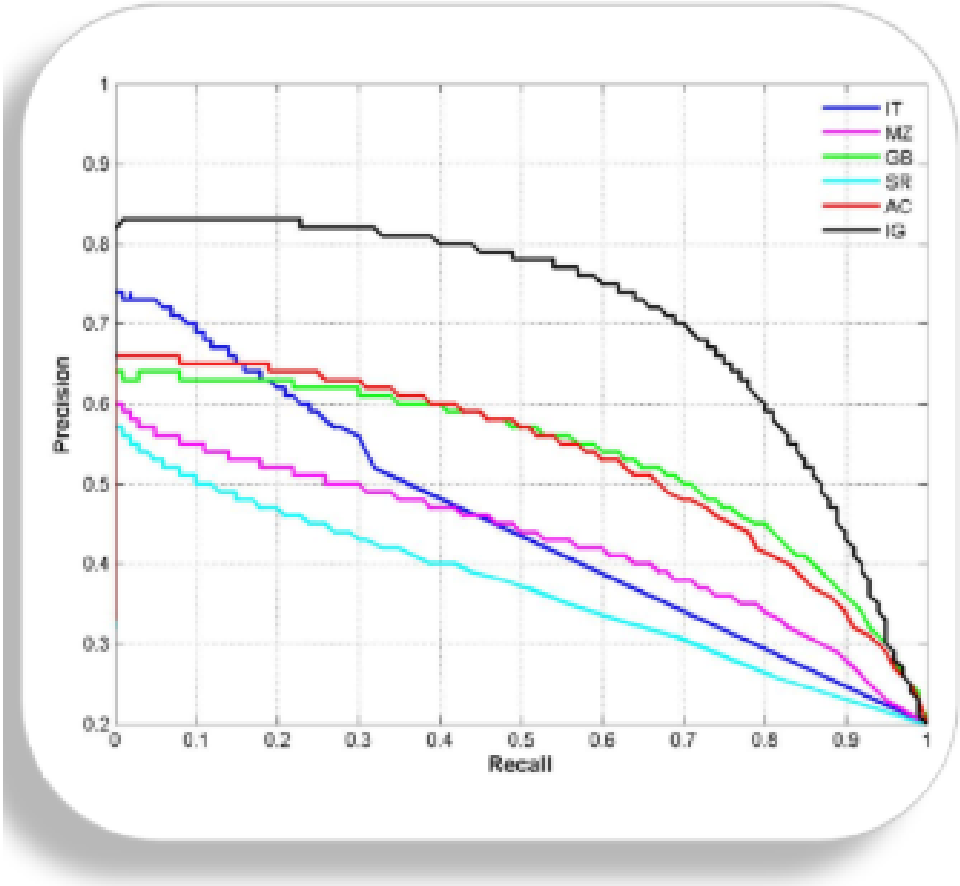
F-measure/F-stats/F1 Score: Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.The F-Measure will always be nearer to the smaller value of Precision or Recall.

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

Specificity: Percentage of negative instances out of the total actual negative instances. Therefore, denominator (TN + FP) here is the actual number of negative instances present in the dataset. It is similar to recall but the shift is on the negative instances. Like finding out how many healthy patients were not having cancer and were told they don't have cancer. Kind of a measure to see how separate the classes are.

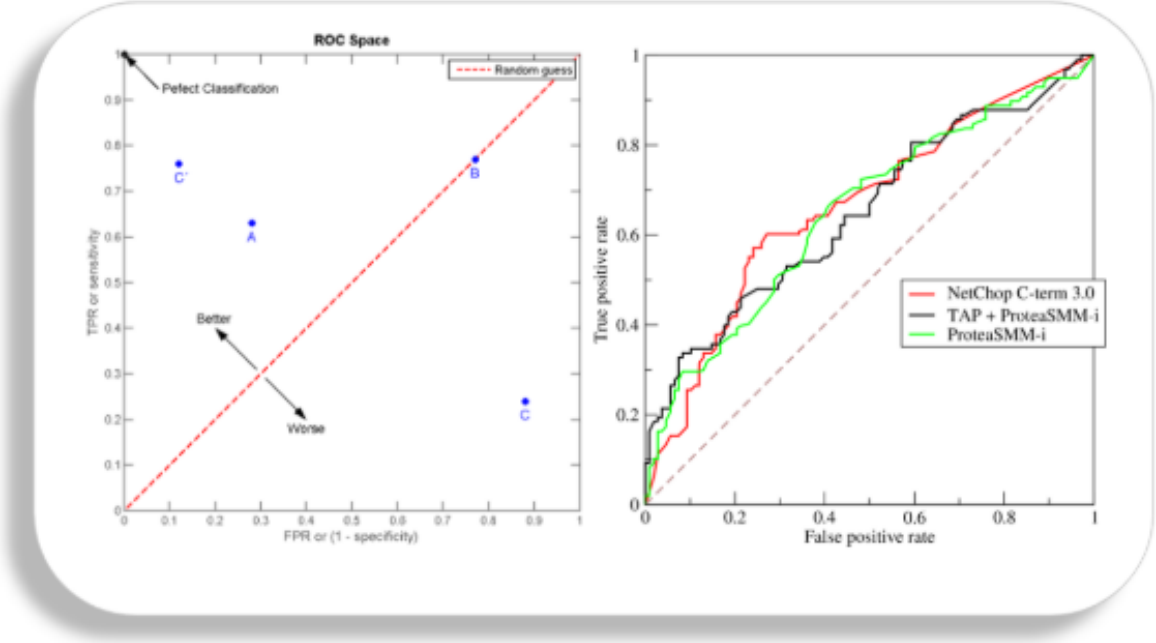
$$\frac{TN}{TN + FP}$$

PR Curve: It is the curve between precision and recall for various threshold values. In the figure below we have 6 predictors showing their respective precision-recall curve for various threshold values. The top right part of the graph is the ideal space where we get high precision and recall. Based on our application we can choose the predictor and the threshold value. PR AUC is just the area under the curve. The higher its numerical value the better.



ROC Curve: ROC stands for receiver operating characteristic and the graph is plotted against TPR and FPR for various threshold values. As TPR increases FPR also increases. As you can see in the first figure, we have four categories and we want the threshold value that leads us closer to the top left corner. Comparing different predictors (here 3) on a given dataset also becomes easy as you can see in figure 2, one can choose the threshold according to the application at hand. ROC AUC is just the area under the curve, the higher its numerical value the better.

PR vs ROC Curve:Both the metrics are widely used to judge a model's performance.



Which one to use PR or ROC?

The answer lies in TRUE NEGATIVES.Due to the absence of TN in the precision-recall equation, they are useful in imbalanced classes. In the case of class imbalance when there is a majority of the negative class. The metric doesn't take much into consideration the high number of TRUE NEGATIVES of the negative class, which is in majority, giving better resistance to the imbalance. This is important when the detection of the positive class is very important.

Like to detect cancer patients, which has a high-class imbalance because very few have it out of all the diagnosed. We certainly don't want to miss on a person having cancer and going undetected (recall) and be sure the detected one is having it (precision).

Due to the consideration of TN or the negative class in the ROC equation, it is useful when both the classes are important to us. Like the detection of cats and dog. The importance of true negatives makes sure that both the classes are given importance, like the output of a CNN model in determining the image is of a cat or a dog.

29) Logistic Regression Code Snippet:

Users > satyam > Desktop > ML_Algos > LogisticRegression.py

```python
#Author: Dhrub Satyam
#Logistic Regression Example
#import libraries
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.cross_validation import train_test_split
from sklearn import metrics

# load dataset
data = pd.read_csv("/path/to/data/file.csv")

#split dataset in features and target variable

X = data[feature_cols] # Features
y = data.label # Target variable

# split X and y into training and testing sets
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)


# instantiate the model (using the default parameters)
logreg = LogisticRegression()

# fit the model with data
logreg.fit(X_train,y_train)

#predicting the test data
y_pred=logreg.predict(X_test)

# printing the confusion matrix
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
cnf_matrix

#Printing Evaluation Matrix
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
```

30) What is Decision Tree Algorithm?

Decision tree is a supervised machine learning algorithm mainly used for the Regression and Classification. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. Decision tree can handle both categorical and numerical data.

31) What are the different algorithms used in Decision Tree? Discuss.

Algorithms used in Decision Trees:

1. CART (Classification and Regression Trees): Uses Gini as splitting criteria.
2. C4.5 (successor of ID3): Uses Entropy as splitting Criteria.
3. ID3 (Iterative Dichotomiser 3): Uses Entropy as Splitting Criteria.
4. CHAID (Chi-square automatic interaction detection): Uses Chi-Square as splitting criteria.

32) What is Pruning? What are different types of Pruning?

Pruning is a technique used in Decision Tree that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting.
There are generally two methods for pruning trees: pre-pruning and post-pruning.
Pre-pruning is going to involve techniques that perform early stopping (e.g. we stop the building of our tree before it is fully grown).
Post-pruning will involve fully growing the tree in its entirety, and then trimming the nodes of the tree in a bottom-up fashion.

33) What is bagging?

Bagging, also called Bootstrap aggregating, is a machine learning ensemble algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach.

Let’s assume we have a sample dataset of 1000 instances (x) and we are using the CART algorithm. Bagging of the CART algorithm would work as follows.

- Create many (e.g. 100) random sub-samples of our dataset with replacement.
- Train a CART model on each sample.
- Given a new dataset, calculate the average prediction from each model.

34) What are advantages and disadvantages of using Decision Trees?

Advantages

1. Easy to Understand: Decision tree output is very easy to understand even for people from non-analytical background. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.
2. Useful in Data exploration: Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. With the help of decision trees, we can create new variables / features that has better power to predict target variable. You can refer article (Trick to enhance power of regression model) for one such trick. It can also be used in data exploration stage. For example, we are working on a problem where we have information available in hundreds of variables, there decision tree will help to identify most significant variable.
3. Less data cleaning required: It requires less data cleaning compared to some other modelling techniques. It is not influenced by outliers and missing values to a fair degree.
4. Data type is not a constraint: It can handle both numerical and categorical variables.
5. Non-Parametric Method: Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

Disadvantages

1. Over fitting: Over fitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning (discussed in detailed below).
2. Not fit for continuous variables: While working with continuous numerical variables, decision tree loses information when it categorizes variables in different categories.
3. Decision Trees do not work well if you have smooth boundaries. i.e they work best when you have discontinuous piece wise constant model. If you truly have a linear target function decision trees are not the best.
4. Decision Tree's do not work best if you have a lot of un-correlated variables. Decision tree's work by finding the interactions between variables. if you have a situation where there are no interactions between variables linear approaches might be the best.

5. Data fragmentation: Each split in a tree leads to a reduced dataset under consideration. And, hence the model created at the split will potentially introduce bias.

35) What are the different splitting criteria used in Decision Tree? Discuss them in detail.

Algo / Split Criterion	Tree Type	Used For	Description	Formula	Splitting Rule
Gini Split / Gini Index	CART	Classification/Catagorical Variables	Favors larger partitions. Very simple to implement.	$1-\sum(P(x=k))^2$	Feature with lower gini index is chosen for split
Information Gain / Entropy	ID3 / C4.5	Classification/Catagorical Variables	Favors partitions that have small counts but many distinct values.	$\sum_{i=1}^{c} - p_i \log_2 p_i$	Choose attribute with the largest information gain as the decision node
Chi-Square	CHAID	Classification/Catagorical Variables	Algorithm to find out the statistical significance between the differences between sub nodes and parent node.	((Actual – Expected)^2 / Expected)^1/2	Feature with higher value of chi-square is chosen for split.
Reduction in Variance		Regression/Continuous variables	Algorithm used for continuous target variables (regression problems). This algorithm uses the standard formula of variance to choose the best split.	$\text{Variance} = \frac{\Sigma(X - \overline{X})^2}{n}$	The split with lower variance is selected as the criteria to split the population:

36) What is Ensemble Modelling? Why is it used?

Ensemble modelling is a process where multiple diverse models are created to predict an outcome, either by using many different modelling algorithms or using different training data sets. The ensemble model then aggregates the prediction of each base model and results in once final prediction for the unseen data. The motivation for using ensemble models is to reduce the generalization error of the prediction.

How to avoid Over-fitting in Decision Tree? Discuss the parameters/hyper parameters in detail.

37) Decision Tree Code Snippet:

Users > satyam > Desktop > ML_Algos > decisiontree.py

```python
#Author: Dhrub Satyam
#Decision Tree Example
#import libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

# load dataset
data = pd.read_csv("/path/to/data/file.csv")

#split dataset in features and target variable

X = data[feature_cols] # Features
y = data.label # Target variable

# split X and y into training and testing sets
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)


# Create Decision Tree classifer object
"""After you get the accuracy and want to improve the accuracy of the model
   you can try optimising model by specifying like below:"""
#clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)
clf = DecisionTreeClassifier()

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

#Printing Evaluation Matrix
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

```

38) What is Random Forest? Why is it preferred over Decision Trees?

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

39) What are the advantages and disadvantages of Random Forest?

Advantage

The following are the advantages of Random Forest algorithm −

- It overcomes the problem of overfitting by averaging or combining the results of different decision trees.
- Random forests work well for a large range of data items than a single decision tree does.
- Random forest has less variance then single decision tree.
- Random forests are very flexible and possess very high accuracy.
- Scaling of data does not require in random forest algorithm. It maintains good accuracy even after providing data without scaling.
- Random Forest algorithms maintains good accuracy even a large proportion of the data is missing.

Disadvantage

The following are the disadvantages of Random Forest algorithm −

- Complexity is the main disadvantage of Random forest algorithms.
- Construction of Random forests are much harder and time-consuming than decision trees.
- More computational resources are required to implement Random Forest algorithm.
- It is less intuitive in case when we have a large collection of decision trees.
- The prediction process using random forests is very time-consuming in comparison with other algorithms.

40) What are the different steps involved in Random Forest Modelling?

A random forest can be considered as an ensemble of decision trees. The idea behind ensemble learning is to combine weak learners to build a more robust model, a strong learner, that has a better generalization error and is less susceptible to overfitting.

The random forest algorithm can be summarized in four simple steps:

1. Draw a random bootstrap sample of size n (randomly choose n samples from the training set with replacement)
2. Grow a decision tree from the bootstrap sample. At each node:

- Randomly select d features without replacement.
- Split the node using the feature that provides the best split according to the objective function, for instance, by maximizing the information gain.

3. Repeat the steps 1 to 2 k times.

4. Aggregate the prediction by each tree to assign the class label by majority vote.
There is a slight modification in step 2 when we are training the individual decision trees: instead of evaluating all features to determine the best split at each node, we only consider a random subset of those.

41) What is cross validation? Why is it used?

The technique of cross validation (CV) is best explained by example using the most common method, K-Fold CV When we approach a machine learning problem, we make sure to split our data into a training and a testing set.

In K-Fold CV, we further split our training set into K number of subsets, called folds. We then iteratively fit the model K times, each time training the data on K-1 of the folds and evaluating on the Kth fold (called the validation data).

As an example, consider fitting a model with K = 5. The first iteration we train on the first four folds and evaluate on the fifth. The second time we train on the first, second, third, and fifth fold and evaluate on the fourth. We repeat this procedure 3 more times, each time evaluating on a different fold. At the very end of training, we average the performance on each of the folds to come up with final validation metrics for the model.

For hyper parameter tuning, we perform many iterations of the entire K-Fold CV process, each time using different model settings. We then compare all of the models, select the best one, train it on the full training set, and then evaluate on the testing set. If we have 10 sets of hyper parameters and are using 5-Fold CV, that represents 50 training loops.

Fortunately, as with most problems in machine learning, someone has solved our problem and model tuning with K-Fold CV can be automatically implemented in Scikit-Learn.

42) What is OOB (Out Of Bag) Error in Random Forest?

Out-of-bag (OOB) error, also called out-of-bag estimate, is a method of measuring the prediction error of random forests, boosted decision trees, and other machine learning models utilizing bootstrap aggregating (bagging) to sub-sample data samples used for training.

Sub sampling allows one to define an out-of-bag estimate of the prediction performance improvement by evaluating predictions on those observations which were not used in the building of the next base learner.

In Layman Term: When you train each tree in random forest, you will not use all the samples. So for each bag, those unused samples can be used to find the prediction error for that particular bag. The OOB error rate can then be obtained by averaging the prediction error from all the bags.

43) What are different Hyperparameter tuning involved in Random Forest?

The different hyperparameters involved in Random Forest are:

1. max_depth : The max_depth of a tree in Random Forest is defined as the longest path between the root node and the leaf node. Using the max_depth parameter, I can limit up to what depth I want every tree in my random forest to grow.
2. min_sample_split: a parameter that tells the decision tree in a random forest the minimum required number of observations in any given node in order to split it.
3. max_leaf_nodes: This hyperparameter sets a condition on the splitting of the nodes in the tree and hence restricts the growth of the tree. The condition is based on the maximum number of leaf nodes allowed.
4. min_samples_leaf : This Random Forest hyperparameter specifies the minimum number of samples that should be present in the leaf node after splitting a node.
5. n_estimators: This is the number of trees you want to build before taking the maximum voting or averages of predictions. Higher number of trees give you better performance but makes your code slower.

6. max_sample (bootstrap sample): The max_samples hyper parameter determines what fraction of the original dataset is given to any individual tree.
7. max_features : This resembles the number of maximum features provided to each tree in a random forest.

44) Random Forest Code Snippet:

Users > satyam > Desktop > ML_Algos > randomforest.py

```
#Author: Dhrub Satyam
#Random Forest Example
#import libraries
import pandas as pd
from sklearn.ensemble import RandomForestClassifier # Import Random Forest Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

# load dataset
data = pd.read_csv("/path/to/data/file.csv")

#split dataset in features and target variable

X = data[feature_cols] # Features
y = data.label # Target variable

# split X and y into training and testing sets
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)

#Create a Gaussian Classifier
clf=RandomForestClassifier(n_estimators=100)

#Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train,y_train)

#Predict
y_pred=clf.predict(X_test)

#Print Accuracy
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

K Nearest Neighbors

45) What is KNN Algorithm? How does it work?

The K Nearest Neighbours algorithm is a classification algorithm used in Data Science and Machine Learning.

The goal is to classify a new data point/observation into one of multiple existing categories. So, a number of neighbours 'k' is selected (usually k = 5), and the k closest data points are identified (either using Euclidean or Manhattan distances)

Of the k closest data points (or 'neighbours'), the category with the highest number of k-close neighbours is the category assigned to the new data point. Intuitively, this makes sense - a data point belongs in the category it's most similar to with respect to its features/properties. The most similar data points are the ones that are nearest to that data point, if you visualize it on a graph.

1. On a new data point, a kNN will calculate it's distance from every single data point in our dataset. The most popular distance metric used is the Euclidean Distance.
2. Once every single distance is calculated, the algorithm will pick the K nearest data points.For classification problems, it'll make a prediction based on the class of those k-nearest datapoints. In this context, it's a good idea to choose K as an odd number to avoid dies. The predicted class will be the most frequent occurring class within K data points.
3. For regression, it can predict based on the mean or median of those data points.

46) What is "K" in KNN Algorithm?

K = Number of nearest neighbours you want to select to predict the class of a given item

47) How to decide the value of K in KNN Algorithm? Why is odd value of K preferable while choosing?

Short Answer:

If K is small, then results might not be reliable because noise will have a higher influence on the result. If K is large, then there will be a lot of processing which may adversely impact the performance of the algorithm.
So, following is must be considered while choosing the value of K:

a. K should be the square root of n (number of data points in training dataset)

b. K should be odd so that there are no ties. If square root is even, then add or subtract 1 to it.

K should be odd so that there are no ties in the voting. If square root of number of data points is even, then add or subtract 1 to it to make it odd.

Long Answer:

There is no straightforward method to calculate the value of K in KNN. You have to play around with different values to choose the optimal value of K. Choosing a right value of K is a process called Hyper parameter Tuning.

The value of optimum K totally depends on the dataset that you are using. The best value of K for KNN is highly data-dependent. In different scenarios, the optimum K may vary. It is more or less hit and trail method.

You need to maintain a balance while choosing the value of K in KNN. K should not be too small or too large.
A small value of K means that noise will have a higher influence on the result.

Larger the value of K, higher is the accuracy. If K is too large, you are under-fitting your model. In this case, the error will go up again. So, at the same time you also need to prevent your model from under-fitting. Your model should retain generalization capabilities otherwise there are fair chances that your model may perform well in the training data but drastically fail in the real data. Larger K will also increase the computational expense of the algorithm.

There is no one proper method of estimation of K value in KNN. No method is the rule of thumb but you should try considering following suggestions:

1. Square Root Method: Take square root of the number of samples in the training dataset.

2. Cross Validation Method: We should also use cross validation to find out the optimal value of K in KNN. Start with K=1, run cross validation (5 to 10 fold), measure the accuracy and keep repeating till the results become consistent.

K=1, 2, 3... As K increases, the error usually goes down, then stabilizes, and then raises again. Pick the optimum K at the beginning of the stable zone. This is also called Elbow Method.

3. Domain Knowledge also plays a vital role while choosing the optimum value of K.

4. K should be an odd number.I would suggest to try a mix of all the above points to reach any conclusion.

48) Why is KNN Algorithm called a Lazy Learner? Can we use KNN for large datasets?

When it gets the training data, it does not learn and make a model, it just stores the data. It does not derive any discriminative function from the training data. It uses the training data when it actually needs to do some prediction. So, KNN does not immediately learn a model, but delays the learning, that is why it is called lazy learner.

KNN works well with smaller dataset because it is a lazy learner. It needs to store all the data and then makes decision only at run time. It needs to calculate the distance of a given point with all other points. So if dataset is large, there will be a lot of processing which may adversely impact the performance of the algorithm.

KNN is also very sensitive to noise in the dataset. If the dataset is large, there are chances of noise in the dataset which adversely affect the performance of KNN algorithm. For each new data point, the kNN classifier must:

1. Calculate the distances to all points in the training set and store them
2. Sort the calculated distances
3. Store the K nearest points
4. Calculate the proportions of each class
5. Assign the class with the highest proportion

Obviously, this is a very taxing process, both in terms of time and space complexity. The first operation is a quadratic time process, and the sorting a O(n log n) process. Together, one could say that the process is a $O(n^3 \log n)$ process; a monstrously long process indeed.Another problem is memory, since all pairwise distances must be stored and sorted in memory on a machine. With very large datasets, local machines will usually crash.

49) Discuss the advantages and disadvantages of KNN Algorithm.

Advantages of KNN

1. No Training Period: KNN is called Lazy Learner (Instance based learning). It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.

2. Since the KNN algorithm requires no training before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm.

3. KNN is very easy to implement. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)

Disadvantages of KNN

1. Does not work well with large dataset: In large datasets, the cost of calculating the distance between the new point and each existing points is huge which degrades the performance of the algorithm.

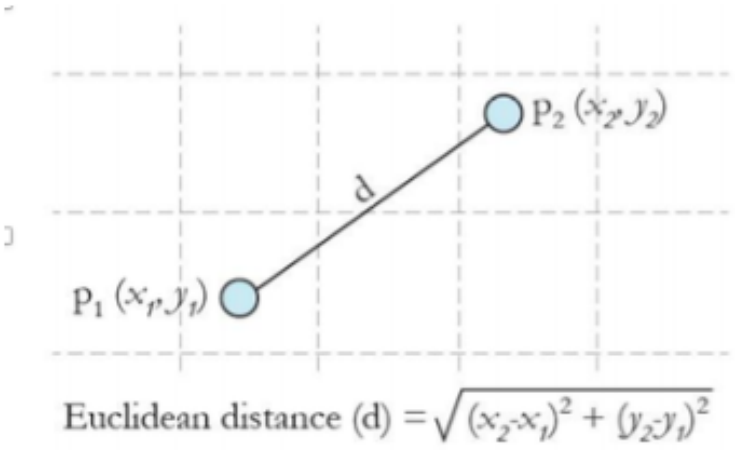
2. Does not work well with high dimensions: The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension.

3. Need feature scaling: We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions.

4. Sensitive to noisy data, missing values and outliers: KNN is sensitive to noise in the dataset. We need to manually impute missing values and remove outliers.

50) What are Euclidean, Manhattan and Chebyshev distance?

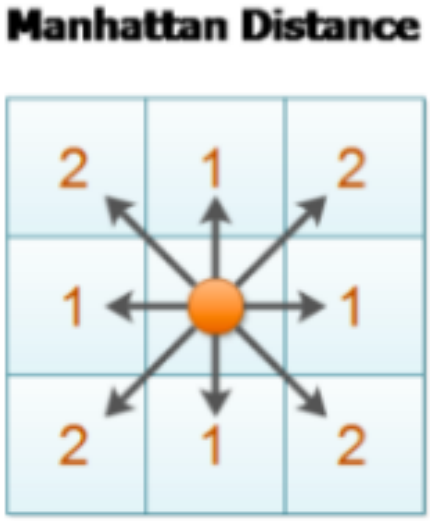
The Euclidean distance or Euclidean metric is the "ordinary" (i.e. straight-line) distance between two points in Euclidean space.



$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \mathrm{d}(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

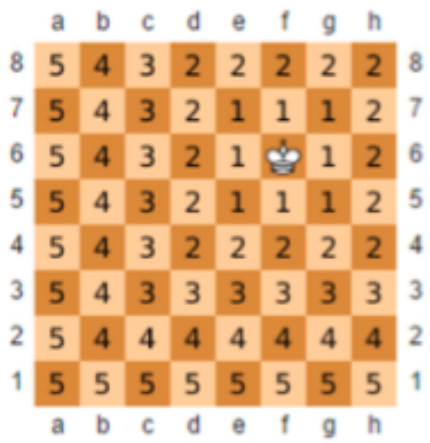
The Manhattan distance, also known as rectilinear distance, city block distance, taxicab metric is defined as the sum of the lengths of the projections of the line segment between the points onto the coordinate axes.In chess, the distance between squares on the chessboard for rooks is measured in Manhattan distance.

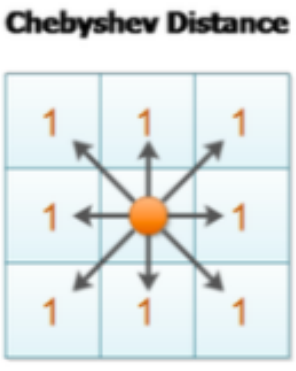


$$|x_1 - x_2| + |y_1 - y_2|$$

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

The Chebyshev distance between two vectors or points p and q, with standard coordinates and respectively, is: It is also known as chessboard distance, since in the game of chess the minimum number of moves needed by a king to go from one square on a chessboard to another equals the Chebyshev distance between the centres of the squares .





$$\max(|x_1 - x_2|, |y_1 - y_2|)$$

$$D_{\text{Chebyshev}}(p,q) := \max_i(|p_i - q_i|).$$

Euclidean distance	**Manhattan distance**
It is computed as the hypotenuse like in the Pythagorean theorem.	It is computed as the sum of two sides of the right triangle but not the hypotenuse.
The mathematical equation to calculate Euclidean distance is : $Euclidean_{distance} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$ Where (x_1, y_1) and (x_2, y_2) are coordinates of the two points between whom the distance is to be determined.	The mathematical equation to calculate Manhattan distance is : $Manhat\tan_{distance} = \lvert x_1 - x_2\rvert + \lvert y_1 - y_2\rvert$ Where (x_1, y_1) and (x_2, y_2) are coordinates of the two points between whom the distance is to be determined.
It the square roots of the sum of difference between the coordinates of the two points.	It the sum of absolute differences of their Cartesian coordinates.
If the points are very far then this will give wrong result because of the curvature of the earth.	It is more suitable natural settings and urban environment.
It always gives the shortest distance between the two points	It may give a longer distance between the two points

Click here for Q.51 to Q.100

To know more information about statistics and it's implementation in Data Science check out the links below

IBM Certified Data Science and AI Program for working professionals.

Eligibility: Minimum 1 yrs of exp.

Duration: 7.5 Months

PDF

IBM Certified Artificial Intelligence and ML Program for working professionals.

Eligibility: Minimum 4 yrs of exp.

Duration: 9 Months

PDF

IBM Certified Data Science and AI Program for Managers and Leaders.

Eligibility: Minimum 8 yrs of exp.

Duration: 11 Months

PDF

Talk to our Industrial Expert and start your journey towards Data Science and AI

Schedule