Classical Machine Learning

- Tejas Bankar

- Data Science is the process of extracting knowledge and insights from raw data by using scientific tools and methods.
- Scientific methods includes:
  Statistics
  Machine Learning
  Deep Learning
  Visualization
  Neural Network
  Natural Language Processing
  Time Series

- Machine Learning is a combination of statistics and computer science in which we are using a past data to answer questions.
- We are training different statistical model on past dataset by using computer science to find the pattern and rules by using which we can predict the answer for unseen data.

Supervised ML:

- When the model is getting trained on the label dataset then it is called as supervised learning.
- Supervised Learning categorized in two types:
  1. Regression : when output variable contains continuous data.
  2. Classification : when output variable contains categorical data.
- Types of supervised ML algorithms:
  1) Linear Regression
  2) Logistic Regression
  3) KNN Classifier, KNN Regressor
  4) Naive Bayes Classifier
  5) Decision Trees
  6) Support Vector Machine (Regression, Classification)
  7) Random Forest
  8) Ada Boost
  9) GDBoost
  10) XGBoost

Unsupervised ML :

- When the model is getting trained on unlabeled dataset then it is called as unsupervised learning.
- Unsupervised learning categorized in two categories:
  1. Clustering : when we want to discover inherent grouping in data.
  2. Association : when we want to discover rules that describe large portion of data.
- Types of Unsupervised ML algorithm:
  1) Hierarchical Clustering
  2) K-means clustering
  3) DBSCAN
  4) Principal Component Analysis

Statistical Concepts

Mean

- It is an average of all the values. It is sum of all the values divided by total number of values.

$$\mathrm{Mean} = \frac{\sum x_i}{n}$$

Median

- The median is the middle value in a sorted list of numbers. If the number of values is odd, the median is the middle number. If even, it is the average of the two middle numbers.

- For odd number of values

$$\mathrm{Median} = \frac{n+1}{2}$$

- For even numbers

$$\mathrm{Median} = \frac{\left(\frac{n}{2}\right)^{th} \text{value} + \left(\frac{n}{2}+1\right)^{th} \text{value}}{2}$$

Mode

- Mode is the most frequently appeared value in data.

Standard Deviation

- Standard deviation is the measure of variation or dispersion in set of values.

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{n}}$$

where, σ = standard deviation, μ = mean , n = number of datapoints.

- A low standard deviation indicates that values are close to the mean and higher standard deviation indicates that values are spread out over wide range.
- Standard deviation helps in assessing consistency in data.

Variance

- Variance is the average of squared difference from the mean.

  $$variance = \sigma^2 = \frac{\sum(x_i - \mu)^2}{n}$$

  where, σ = standard deviation, μ = mean , n = number of datapoints.

- Variance is also the measure of spread in data like standard deviation. It gives sense of overall spread in data.
- Difference in variance and standard deviation is that, variance is expressed in squared unit of original data. For example if data is in meters then variance will be in square meters. Standard deviation on the hand expressed in same unit as of data. It is more easier to interpret than variance.

Covariance

- Covariance is the measure of directional relationship or joint variability between two random variables.

  $$cov(X,Y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n}$$

  where, $x_i$and $y_i$are the values of two variables, $\bar{x}$ and $\bar{y}$ are the mean of x and y variables respectively, n = number of datapoints.

- Range of covariance is - infinity to + infinity.
- Positive covariance indicates that the variables tend to move in the same direction, while a negative covariance indicates that variables move in opposite directions.

Coefficient of Variation (CV)

- The coefficient of variation is a relative measure of variability that indicates the size of a standard deviation in relation to its mean.
- $CV = \frac{\sigma}{\mu}$
- It is standardized unitless measure which allows us to compare variability between different datasets, even if they have different units.
- Lower CV indicates more precise estimates, while higher CV indicates greater variability relative to the mean.

Coefficient of Correlation (R)

- Coefficient of Correlation is the measure of strength and direction of linear relationship between two variables.

  $$R = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

  $$R = \frac{cov(X,Y)}{\sigma_x \ \sigma_y}$$

- Coefficient of Correlation ranges from -1 to +1. If the correlation coefficient is close to +1 then it is strong positive linear relationship between two variables, if it is close to -1 then its strong negative linear relationship. If the correlation coefficient is in between -0.3 to +0.3 then its a week relationship. If the value is 0 then it means no linear relationship.
- It helps in understanding how one variable changes with respect to another variable. For example, a positive correlation indicates as one variable increases, other also tends to increase.

Normal Distribution

- Normal Distribution is a continuous probability distribution which is symmetrical around its mean and most of the observations are clustered around central peak and the probabilities for values further away from the mean tapered off equally in both sides.
- It is also known as Gaussian distribution.
- The shape of the curve is called as Bell shape curve.
- The empirical rule for normal distribution is 68 % of observations should fall within 1st std deviation range from the mean, 95 % observations should fall within 2nd std deviation range from mean and 99.7 % of observations should fall within 3rd std deviation range from mean.

Skewness :

- If there is a distortion or asymmetry in data, then it deviates from symmetrical bell shape curve. It is called as Skewness.
- There are two types of skewness:
  1. Positive skewness / right skewness :
     - When the tail of distribution is longer towards right hand side of the curve then it is positive skewness.
     - Example, if we take age of India's population then it will show positive skewed curve because younger population is more than senior.

  2. Negative skewness / left skewness :
     - When the tail of distribution is longer towards left hand side of the curve then it is negative skewness.
     - Example, distribution of age of deaths.

Standard Normal Distribution :

- The standard normal distribution is a special case of the normal distribution where the mean is zero and the standard deviation is 1. This distribution is also known as the Z-distribution.
- A value on the standard normal distribution is known as a standard score or a Z-score.
- Standard score or Z-score represents the number of std deviation away the datapoint is from the mean.

Central Limit Theorem

- The Central Limit Theorem states that the sampling distribution of the sample means approaches a normal distribution as the sample size gets larger — no matter what the shape of the population distribution.

- This fact holds especially true for sample sizes over 30.

Law of Large Numbers

- Law of large numbers states that if the same experiment or study is repeated independently for large number of times, then the average of the results of the trials must be close to the expected value.

Hypothesis Testing

- Hypothesis testing is a statistical tool which is use to confirm our observation or assumption or hypothesis about population using sample data.
- First we study the sample of data and we assume that observation on sample is also followed by population.
- So, using Hypothesis testing, we can determine whether we have enough statistical evidence to conclude if the hypothesis about population is true or not.
- There are two hypothesis statements:
  1. Null Hypothesis (H0) :
     - It states that there is no significant difference between sample and population variables.
     - It treats as everything is same or equal.

  2. Alternative Hypothesis (H1) :
     - It states that there is significant difference between sample and population variables.
     - It treats as everything is not same or equal.

- To decide whether to accept or reject null hypothesis, we perform hypothesis test.
- There are two types of tests:
  1. One-tailed Test :
     - Region of rejection is only on one side of the sampling distribution.
  2. Two-tailed Test :
     - Region of rejection is on both sides of the sampling distribution.

- From this tests we get p-value.
- p value is a probability for a null hypothesis to be true.
- Then we compare this p value against significance level,
- Significance level is like a threshold we set for p value, because 100 % accuracy is not possible in real world.
- It is usually 0.05, but it depends on business domain and complexity.
- If p value is more that significance level, then we accept the null hypothesis, because there are no significant evidence against null hypothesis.
- If p value is less than significance level, then we reject the null hypothesis and accept the alternative hypothesis, because there are significant evidence against null hypothesis.

- Type I Error :
  - If we reject the null hypothesis when it is actually true.
- Type II Error :
  - If we accept the null hypothesis when it is actually false.

- Types of Hypothesis Tests :
- For Normality : H0 : data is normally distributed
  - Shapiro test : from scipy.stats import shapiro
  - normality : from scipy.stats import normality
  - kstest : from scipy.stats import kstest

- chi-square test : To find relationship between 2 categorical variables.
  - H0 : Two variables are independent
  - H1 : Two variables are dependent
  - from sklearn.feature_selection import chi2

- T-Test : To compare mean of sample and population | H0 : mean of mean of two variables is same
  - It assumes that data is normally distributed.
  - It is use when mean and std are not known
  - It is use when samples are less than 30
  - from scipy.stats import ttest_1samp

- z-Test : To compare mean of sample and population | H0 : mean of mean of two variables is same
  - It is use when samples are more than 30
  - It is use when std is known.

Machine Learning Algorithms:

Linear Regression

- Linear Regression is a supervised learning algorithm which is use for regression problem. Linear regression which means predicting the continuous value from linear relationship between two or more variables.
- Linear regression gives us a best fit regression line in such a way that it passes through max data points and vertical distance between datapoints and regression line is minimum. The equation for regression line is y = mx + c if there is only one independent variable and if there are multiple independent variables then equation is y = m1x1 + m2x2 +... + c.
- The aim in a linear regression model is to get a best values of slope and intercept in such a way that MSE is minimum. To achieve this we uses Gradient Descent algorithm, which is a optimization iterative algorithm, it minimizes the cost function to get a best values of coefficients, for that it uses partial derivative of cost function with respect to coefficients and it finds the new value of coefficients for every iteration, the rate at which GD updates is called as learning rate, initially the learning rate is high and as it approaches towards global minima then learning rate decreases, finally it gives convex curve and at the global minima point we get the best values of coefficients. In linear regression coefficients are slope and intercept and cost function is MSE.

Assumptions of Linear regression are:

- Linearity : There should be linear relationship between dependent and independent variables. By using scatter plot we can check for linear relationship.
  If not : Linear regression gives best fit straight line, if relation between dependent and independent variables is not linear (curvilinear) then model will not able to study right pattern and error will be more because best fit regression line will not pass through max data points and residual will be more.

- No Multicollinearity : There should not be relationship between independent variable. By using scatter plot, correlation matrix and vif we can check this.
  If not : If multicollinearity is present then it is hard for model to estimate relationship between independent variable and dependent variable. It will affect the precision of coefficients, but generally it does not affect on predicting power of model.

- Residuals should follow a normal distribution. By using Q-Q plot, Histogram, kde plot, hypothesis testing which includes shapiro test, kstest, normality test we can check this assumption.
  If not : There may be problem with the stability, reliability of model on unseen data. Model will give random error, so to have a generalize performance of model, residual should be normally distributed.)

- There should be homoscedastic behaviour in model, means residuals should be constant along the values of dependent variable. By using scatter plot of residuals against dependent variable we can check for this assumption.
  If not : Model will not give constant error over the range of dependent variable. So to have a generalize performance of model, homoscedasticity should be present. Transformation can be done if homoscedasticity is not there.

Evaluation Metrics for Regression

- Evaluation parameters for linear regression are mean absolute error, mean squared error, RMSE and r2 score.

r2 score :

- r2 score is also called as coefficient of determination, it is a measure to indicate how close the datapoints are fitted to regression line. formula for r2 score is :
  $$\mathrm{r}^2 = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}}$$

SSE (Sum of Squared Error):

- It's a sum of squared error between actual value and predicted value
  $$SSE = \sum (Ya - Yp)^2$$

SSR (Sum of Squares due to residual / Residual sum square):

- It's a sum of square error between predicted value and mean of dependent variable
  $$SSR = \sum (Yp - \hat{y})^2$$

SST (Total sum square):

- It's a sum of square error between actual value and mean of dependent variable
  $$SST = \sum (Ya - \hat{y})^2$$

Drawbacks of r2 score :

- r2 value will never decrease. It always increases even if we add non correlated features.

Adjusted r2 :

- Adjusted r2 is a modified version of r2 that has been adjusted for number of predictors in model.
- Adjusted r2 increases only when correlated features are added.
- Adjusted r2 will not increase when non correlated features are added.
- Adjusted r2 $= 1 - \frac{(1 - r2)(N-1)}{(N - P - 1)}$

  P = number of independent variables.

VIF :

- VIF is variance inflation factor which is used to check multicollinearity.
- It gives strength of correlation between two independent variables.
- It's range from 1 to inf
- It's formula is :
  VIF = $\frac{1}{1 - \mathrm{r}^2}$

- Therefore when r2 is 0 then VIF is 1 which means there is no correlation between independent variables.
- If r2 is 0.8 then VIF is 5 and if r2 is 0.9 then VIF is 10, so whether to select VIF as 5 or VIF as 10 is depends on problem complexity. Generally we select 10.
- If we found VIF > 10 in two or more features then we have to drop one of the feature. We drop feature which having highest VIF , and it's a iterative process repeats until we have all features having VIF < 10.

What is Least Squares Regression?
• Least square means fitting the regression line by minimizing the sum of squares of the distance between the actual and predicted values.

What is Partial Least Squares Regression?
• Partial least square regression technique is used for prediction of dependent variables from large number of independent variables.
• Once the reduction to a smaller set is accomplished, least square regression is done on the fresh components. This technique is helpful when we have collinear predictors/ independent variables.

Logistic Regression

- It is a supervised learning algorithm use for a classification problem. It calculates probabilities of classes of target variable to make a prediction. It is a binary classifier.
- There are two types in logistic regression, first is binomial where target variable is having only two classes, and second is multinomial or multiclass where target variable is having multiple classes.
- Logistic Regression uses sigmoid function to create a sigmoid curve on which probabilities are mapped and from which we predict in which class it will be going to fall, for that we uses threshold value, by default it is 0.5.
- Sigmoid function is a transformation of linear regression function which estimates the probability of class.
- Sigmoid function is :

$$\text{p} = \frac{1}{1 + e\hat{} - (mx + c)}$$

- Aim of Logistic Regression algorithm is to get a best sigmoid curve, for that we uses gradient descent to minimize cost function, cost function in logistic regression is log loss function which is
  $Log\ Loss = -\frac{1}{N}\sum\big(Ya\ * \log(P) + (1 - Ya) * \log(1 - P)\big)$
- It takes partial derivative of cost function with respect to m and c, and it finds new values of coefficients for each iteration, and at global minima we get values of coefficients which gives best sigmoid curve.

- Multiclass Logistic Regression :
- There is a OVR (one vs rest) concept in a logistic regression use to deal with multiclass classification problem.
- Logistic regression is a binary classifier so it will create binary model at one time.
- suppose there are 3 different categories .. A, B and C.
- for every input sample it will create 3 models:
  For the 1st iteration it will create Model M1 of (A) vs (B,C) and finds the probability P1 for class A.
  For the 2nd iteration it will create Model M2 of (B) vs (A,C) and finds the probability P2 for class B.
  For the 3rd iteration it will create Model M3 of (C) vs (A,B) and finds the probability P3 for class C.
  And then it will check that which model has highest probability. Class with the highest probability is predicted class.
- Suppose here Model M2 has highest probability P2 then output class will be 'B'.
- This is how Logistic Regression deals with the multiclass classification problem using OVR concept.

Assumptions made in logistic regression :

1. There should be no multicollinearity between independent variable
2. There should be linear relationship between the log odds and independent variables(X).(if log odds are linearly related to independent variables, then the relation between X and P is nonlinear, and if it is nonlinear then only it will form S-shaped curve which we want.)

Advantages :

1. Easy to implement
2. Less likely to overfitting
3. Performs well on linearly separable dataset.
4. Performs well on simple dataset (less features)

Disadvantages :

1. When independent variables are highly correlated with each other, it may affect an performance of model.
2. If there is no linear relationship between independent variable and logit odd, it may affect the performance of model.
3. It is highly sensitive to outliers
4. Linearly separable dataset is rarely available in real world.

Evaluation Metrics for Classification

Confusion Matrix :

- Confusion matrix is an evaluation metrics of classifier.
- It has 4 elements :
  1. True Positive : It means actual value is positive and predicted value is also positive
  2. True Negative : It means actual value is negative and predicted value is also negative
  3. False Positive : It means actual value is negative but predicted value is positive
  4. False Negative : It means actual value is positive but predicted value is negative

TPR :

- It is Proportion of positive class got correctly classified by classifier
  e.g. Out of 100 COVID+ patients 80 are correctly classified as COVID+ by classifier.

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

  $TPR = \frac{80}{80 + 20} = 0.8$

TNR / Specificity :

- It is proportion of negative class got correctly classified by classifier
  e.g. Out of 100 non COVID persons 70 are correctly classified as non-COVID by classifier

- $TNR = \frac{TN}{TN + FP}$

  $TNR = \frac{70}{70 + 30} = 0.7$

FPR :

- It is Proportion of negative class got incorrectly classified by classifier
  e.g. Out of 100 COVID negative persons 20 are incorrectly classified as COVID +ve by classifier.

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

  $FPR = \frac{20}{20 + 80} = 0.2$

FNR :

- It is Proportion of positive class got incorrectly classified by classifier
  e.g. Out of 100 COVID +ve patients 20 are incorrectly classified as COVID -ve by classifier.
- $FNR = \frac{FN}{FN \ + TP}$

  $FNR \ = \frac{20}{20 \ + 80} = 0.2$

Accuracy Score :

- Accuracy score measures how correctly classifier is predicting.
- It's a ratio of correct predictions to total number of predictions.
- $accuracy\_score \ = \frac{TP \ + \ TN}{TP \ + \ TN \ + \ FP \ + \ FN}$
- When we get good accuracy_score, it doesn't mean that model is performing well.
- accuracy_score is good metric when we are dealing with balance data, but most of the times target classes in data is imbalance, and on such data accuracy_score is not good metric to use.
- e.g. if we are having 95 non-spam mails and 5 spam mails, and if we train model on this imbalanced data, then model will get biased on majority classes and on test data which is also imbalanced in same proportion, definitely we will get good accuracy_score because model will perform well on majority classes, but it will not magnify false predictions it made on minority classes. So we can't just rely on accuracy score and we need to use other metrics like precision, recall, f1 score to understand model performance well according to our use case.

Precision :

- It tells us Out of total predicted positive values how many are the actual positive values
- $precision \ = \frac{TP}{TP + FP}$
- Precision is important where False Positives are of a concern.
- e.g. In case of Spam Mail :
  detected spam mail as spam mail is TP
  detected non spam mail as spam mail is FP
- So, it is totally unacceptable if our important mail goes in spam mail, on the other hand we can afford some of the spam mails comes in our inbox.
- In this case minimizing the FP near to 0 will increase the precision, which we want.

Increasing the precision might decrease the recall, means decreasing the FP might increase the FN.

Recall :

- It is Proportion of positive class got correctly classified by classifier.
- It tells us out of total actual positive values how many are the predicted positive values.
- $recall = \frac{TP}{TP + FN}$
- Recall is important where False Negatives are of a concern.
- e.g. In case of medical field,
  detecting cancer patient as not having cancer is FN
  detecting cancer patient as having cancer is TP
- So, it is very dangerous that actual cancer patient is detected as not having cancer. To avoid this error we want FN as lowest as possible. On the other hand we can accept if some of non-cancer patient detected as cancer.
- So, we want recall as near to 1

f1 score :

- f1 score combines precision and recall into single metric.
- f1 score is a harmonic mean of precision and recall.
- $f1 \ = \frac{2 * P * R}{P + R}$
- regular mean treats all values equally, but harmonic mean gives much more weight to low values. As a result the classifier will only get high f1 score if both precision and recall are high.
- Hence f1 score is important and gives more clear evaluation when we want both precision and recall to be high.
- Suppose we are having precision = 0.6 and recall = 0.95, then simple mean is 0.775 but harmonic mean = 0.735, so harmonic mean represents importance of both precision and recall. It will be high only if both precision and recall are high.

Classification Report :

- Classification report consist of multiple parameters such as accuracy_score, precision, recall, f1 score. It is very useful report to evaluate model performance.

ROC AUC :

- ROC_AUC is evaluation metric use for binary classification.
- Receiver Operator Characteristic (ROC) is a probability curve that plots the TPR(True Positive Rate) against the FPR(False Positive Rate) at various threshold values.
- FPR is on x axis and TPR is on y axis
- The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes, it summarizes ROC graph by using single number. It ranges from 0 to 1. Greater the AUC better is the performance of model.
- The perfect classifier will have AUC = 1, which means high value of TPR and low value of FPR.
- When AUC = 0.5 then it means classifier is random, so generally in real life AUC >= 0.7 considered as good.

KNN

- KNN is a simple supervised ML algorithm which can be used for both classification and regression problem. It is distance based algorithm, and as name suggest it considers K number of nearest neighbours to predict the class of new datapoint in case of classification and continuous value in case of regression.
- To find K nearest neighbours there are two distance metrics,
  first is eulcidean distance, it is shortest distance between two datapoints and formula is : $d = \sqrt{(Y2 - Y1)^2 + (X2 - X1)^2}$
  and second is manhattan distance, it is sum of absolute diff between points across all the dimensions, and formula is : $d = |Y2 - Y1| + |X2 - X1|$
- For the classification problem it takes majority of neighbours class and that will be the predicted class.
- For regression problem it takes mean of values of neighbours and that will be the predicted value.
- To decide best value of K we can use Hyperparameter tuning and passing the range of values of K in GSCV or RSCV.
- Advantages of KNN is that :
- It is non parametric algorithm, so it does not assume anything
- It is easy to implement

Disadvantage of KNN is that :

- It is highly sensitive to outliers.
- Feature scaling is mandatory in KNN
- It does not work well with large dataset.
- Lazy Algorithm

Why KNN is lazy algorithm ?
KNN not learns any function or parameters during training, instead it stores all datapoints in n-dimensions every time, that's why it is lazy algorithm.

Decision Tree

- DT is a supervised learning algorithm which can be used for both classification as well as regression problem. Mostly prefer for classification problem.
- It gives tree like structure where there are 3 main parameters , first is root node or decision node, this represents the feature is going to split further. then 2nd is branches, this represents the decision rule on which basis the root node is going to split. and 3rd is leaf node, which represents the outcome.
- To make DT there are 2 methods first is ID3 and second is CART.

- In ID3 we finds Information Gain for all features, IG is measure of reduction in entropy, and Entropy is the measure of impurity in node.
  $entropy = - P(Y) \log_2 P(Y) - P(N) \log_2 P(N)$
  *IG = Total Entropy − summation of weighted avg * entropy of each feature*.
- Then whichever feature having highest IG will becomes decision node and that feature will split first.

- In CART we finds Gini Index for all features.
  $GI = 1 - (P(Y)^2 + P(N)^2)$
- then whichever feature is having lowest GI is going to split and it becomes Decision node.

- Difference between entropy and Gini is that time complexity is more when we use entropy as it uses log term while Gini uses simple mathematics. Entropy ranges from 0 to 1 while gini ranges from 0 to 0.5. For pure node both entropy and gini is 0 and for total impure node entropy is 1 and gini is 0.5.
- For Regression it takes best threshold value as a decision splitter. It splits the target variable in two regions and predicted value is mean of particular region.
- Initially it takes mean of 2 consecutive rows as a threshold value and splits the target variable in two regions then it finds the mse for both the region and sums it as a MSE for that threshold value. This step repeated till last row and then whichever threshold has lowest MSE will be our best threshold and it becomes decision node or splitter.
- Decision tree is prone to overfit so to reduce overfitting we can use hyperparameter tuning, there are multiple hyperparameters like criterion, splitter, max_depth, min_sample_split, min_sample_leaf.
- Another way to reduce overfitting of DT is pruning. Pruning is technique used to reduce the size of trees by removing unwanted sections of tree which are non-critical, It reduces the complexity of model and slightly increases the training error but drastically reduces the testing error.
- There are two types of pruning, first is pre pruning also called as forward pruning, it stops the non-significant branches from generating. Hyperparameter tuning is use to achieve this by using hyperparameters like max_depth, min_sample_split, min_sample_leaf. Second method of pruning is post pruning also called as backward pruning, in this process the tree is generated first and then non-significant branches are removed. In this we use cost complexity pruning technique also called ccp_alpha. higher is the value of ccp_alpha more nodes are pruned.

Random Forest

- Random Forest is an extension to bagging method use for both classification and regression problem. It uses bagging and random feature selection to create an uncorrelated Forest of decision trees.
- Bagging is an ensemble technique and it is also called as bootstrap aggregation, here bootstrapping means creating diff subsets of training dataset by randomly selecting samples with replacement, means individual datapoint can be selected for multiple times.
- Then this bootstrap samples are trained independently
- Aggregation means output or predictions of all the models are aggregated together to get best optimal prediction. In regression mean of all the outputs is taken, and in classification there are 2 methods, soft voting and hard voting, in Soft voting we take a mean of probabilities of Yes and No and whichever is having highest mean will be our predicted class, and in hard voting we take the majority of all the outputs.
- The 2 most important hyperparameters in RF are n_estimators which represents number of DT's to be trained, and second is max_features which represents maximum number of features to select for training the model.

Advantages :

- It is less likely to overfitting than DT.
- It works well without tuning.

Disadvantages :

- Time complexity is more
- Random Forest algorithm may change considerably by a small change in the data.
- sometimes it fails to determine significance of each feature.

AdaBoost

- AdaBoost is also called as Adaptive Boosting it is used as a boosting technique of ensemble method.
- Initially all the datapoints will be assigned with equal sample weights. which is $\frac{1}{n}$
- Then it will create decision stump for each feature and select the best decision stump as first weak learner.
- Then we calculate Total Error of decision stump or weak learner, formula for Total Error is number of total misclassified data points divide by total data points.
- Then we calculate performance of stump (α) which gives actual influence of model in classifying data correctly.
  $\alpha = \frac{1}{2} \ln\left(\frac{1 - TE}{TE}\right)$
  As the TE increases alpha decreases and vice versa.
- Now we have to create new model, for that we need to make new dataset, to create new dataset we need to reduce the weights for correctly classified samples and increase the weights for incorrectly classified samples. For that we have to update the sample weights which we had initially taken.
- The formula for new sample weights is old sample weights * e ^(+- alpha). for correctly classified samples alpha is -ve for incorrectly classified samples alpha is +ve.
- Now as the sum of updated weights is not equal to 1, so to make it equal to 1, the normalization is done by dividing all the weights by sum of updated weights.
- Now we will divide data points into buckets of sample weights. Buckets are created by taking cumulative freq. of updated sample weights.
- Now the algorithm takes random numbers between 0 to 1. and since bucket of incorrectly classified records have higher sample weights, so the probability of selecting those records is very high for next model. This way we get new dataset and all the previous steps are repeated on it again.
- We do multiple iterations of this until we minimizes the error, by default in sklearn the algorithm will iterate for 50 times.

Advantages :

- accuracy of weak classifier can be improved by using AdaBoost.

Disadvantages :

- sensitive to outliers, needs quality dataset

Gradient Boosting

- GB is supervised machine learning algorithm which is a boosting algorithm of ensemble technique. It is one of the most popular algorithm, we can use it for regression as well as classification problem.
- GB builds models sequentially and each new model tries to reduce residual of previous model.
- There are 3 main components of GB:
  1. Loss Function : It depends on problem we are solving, for regression there are MSE,RMSE etc. for classification there are LogLoss, Hinge Loss etc. The condition is that the loss function should be differentiable.
  2. Weak Learner : weak learners are the models or shallow Decision Trees which means highly pruned DT with max number of leavesin between 8 to 32.
  3. Additive Model : It is a sequential model and at each iteration our loss function should be reduced.

GB Regressor :

- Initially we create a base model which has constant output or predicted value(ŷ). we finds this constant predicted value in such a way that the loss is minimum, for that we takes first order derivative of loss function w.r.t ŷ (predicted value). In regression loss function is simply the residual square which is $\sum \frac{1}{2}(Y - \hat{y})^2$
- Now we have to find the pseudo residual for this previous model, for that the formula is derivative of loss function w.r.t predicted value(ŷ). ($r_{im} = \frac{dL(Yi,\hat{y})}{d(\hat{y})}$) , here Yi means actual value and ŷ means predicted value.).
- Now we have to create next model and for that output or dependent variable is residual of previous model. so we are predicting the residual of previous model. Now again it will find the constant predicted value for this model in such a way that the loss is minimum.
- This finding constant predicted value and residuals is repeated until the residuals are minimizes, so that we get less variance and good results. By default it will take 100 number of DT's , and then the final output will be , prediction of base model plus summation of lambda * residual for each model.
- In GB the models are distinct shallow DT's means highly pruned trees which are dissimilar for each model. so that the speed will increase.

GB Classifier :

- It is use when target variable is binary
- All the steps are similar to regression, only the loss function in classification is Log-likelihood function which is also called as log loss function which we use in logistic regression.
- To initialize the base model we have to find a constant value, for that we use log(odds) because when we differentiate loss function we get function of log(odds), so,
- After transforming the loss function the equation becomes, : L = - Y * log(odds) + log(1 + e ^ log(odds))
- log(odds) gives us the probability of class
- Now we have to find the value of log(odds) for which the loss function is minimum. For that we take the derivative of this loss function w.r.t to log(odds) and then put it equal to 0, so we will get a probability.
- Now we have to find the residual between actual probability and predicted probability, and similar to regression we will predict this residuals in next model. Now again it will find the constant predicted value of log(odds) for this model in such a way that the loss is minimum.
- This steps repeated until residual is minimizes.

Hyperparameters in GB are:

1. n_estimators
2. Learning Rate

Advantages of GB:

1. performance of GB is very good
2. Preprocessing is not required we can use directly with missing values and outliers as it is.
3. It is very flexible model, we can use different loss functions and provides several hyper parameter tuning options that make the function fit very flexible.

Disadvantages of GB:

1. small change in training dataset can create radical change in model.
2. Not easy to understand predictions
3. Gradient Boosting Models will continue improving to minimize all errors. This can overemphasize outliers and cause overfitting.
4. Computationally expensive - often require many trees (>1000) which can be time and memory exhaustive.

Support Vector Classifier

Support Vector Machine

- SVM is a supervised learning algorithm which can be used for both classification as well as regression problem. Mostly prefer for classification problem.
- In SVM we find the optimal hyperplane which separates the classes in n dimensional space. To make the hyperplane It selects the extreme datapoints of classes, means closest data points of two classes. This extreme datapoints is called as support vectors, that's why algorithm is known as support vector machine. Then it will creates marginal planes through this support vectors which are parallel to each other And the distance between this marginal planes is called margin, and aim of this algorithm is to find the hyperplane which having the maximum margin.
- There are two types of SVM, first is Linear SVM , where data is linearly separable, in this type we can separate the classes by straight line or plane. Second type is Non-linear SVM, where data is not linearly separable, so we need to add one more dimension to make the data separable.
- SVM Kernels:
- Kernel Function is a method used to take data as input and transform it into the required form of processing data.
- Kernel is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces. Basically, It returns the inner product between two points in a standard feature dimension.

Types of SVM Kernels:

- Gaussian kernel Radial Basis function
- Sigmoid kernel
- Polynomial kernel
- Linear kernel

Advantages :

- SVM works relatively well when there is a clear margin of separation between classes.
- SVM is more effective in high dimensional spaces.
- SVM is effective in cases where the number of dimensions is greater than the number of samples.
- SVM is relatively memory efficient

Disadvantages :

- SVM algorithm is not suitable for large data sets.
- SVM does not performs well when classes are overlapping

Naive Bayes

- Naive Bayes is a supervised learning algorithm which is based on Bayes theorem and used for solving classification problems.
- Naive means all the features are independent to each other and Bayes is for Bayes theorem.
- Bayes theorem is use to determine the probability of hypothesis with prior knowledge. It depends on conditional probability.
- Using Bayes theorem we can find posterior probability, which is probability of event 'A' happening given that event 'B' has occurred, 'A' is hypothesis and 'B' is evidence. Formula for posterior probability is likelihood probability * prior probability divide by marginal probability.
  $$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$
- There are 3 variants in Naive Bayes, 1st Gaussian NB which can be use when there are continuous variables, 2nd is Multinominal NB which can be used when there are discrete variables, 3rd is Bernoulli NB which can be used when there are binary variables

Applications :

- NB is mainly used in text classification that includes high dimensional training dataset.

Advantages :

- It is one of the most fast and effective algorithm use for classification.
- It performs well on high dimensional dataset.

Disadvantage :

- It assumes that all the variables are independent to each other.

Ensemble Techniques

- Ensemble method means combining multiple models to solve complex problem and to get better prediction.
- There are multiple methods in ensemble learning like Bagging, Boosting, Voting, Stacking.
- Bagging is a parallel approach and Boosting is a sequential approach.
- In bagging we have to create multiple bootstrap datasets on which we train the models independently and then aggregate output of all models together to get optimum prediction.
- In boosting we sequentially convert weak learner to strong learner by reducing the error in each iteration.
- In Voting Ensemble method we train multiple models on same data independently and then we aggregate output of all models together to get final output.
- In stacking we train multiple base models, & using output of these models we train meta model which gives us final output.

Voting Ensemble

- In Voting Ensemble method we train multiple models on same data independently and then we aggregate output of all models together to get final output.

Assumptions in Voting Ensemble :

- All the models should be independent of each other
- The minimum accuracy of all the models should be 51 %. Otherwise if any model has accuracy less than 50% then final accuracy of Voting Ensemble will be worse than that model.
- Mathematically we get more accuracy in Voting Ensemble than base models.

VotingClassifier :

- In classification there are two types for voting classifier.
  1. Hard Voting : In hard voting classifier we take a majority of classes

2. Soft Voting : In soft voting classifier we take mean of probabilities of classes & whichever class having highest mean that will be output class.

VotingRegressor :

- In regression we simply take a mean of all models output.
- Voting Ensemble is very imp concept for tuning and getting better results. Sometimes single model can gives better results than Voting Ensemble, but sometimes VotingEnsemble can perform very well by combining multiple independent models.
- It's all about doing experimentation.

Blending & Stacking

- In stacking we train multiple base models, & using output of these models we train meta model which gives us final output.
  STEP 1 : Train base models
  STEP 2 : Make Predictions using base models
  STEP 3 : Train meta model using predictions of base models as independent variables and original target variable as target variable.
- If we use same data for base model training-testing and meta model training-testing, then there may be problem of overfitting. To avoid this there are two methods.

Hold out method / Blending :

- When we perform stacking using hold out method then it is called as blending.
- In Blending we first split data into 2 parts (D_train, D_test). Then we again split D_train into 2 parts (D_train_base, D_test_base). Now we train our base models on D_train and make predictions on D_test_base to create new dataset for meta model. Then we train Meta model on this new dataset and make predictions on D_test for testing.

K-Fold method :

- In K-Fold method first we split data into 2 parts (train and test).
- Then we create K folds of train data.
- Then We train base model for K number of times on different combinations of K folds and make predictions accordingly. We will do this for multiple base models & we will use predictions of this models as a input data to meta model.
- Then we will train our Meta model
- Now we will forgot our previous K Folds and base models. & we will train our base models one whole train data.
- So here we first get meta model and then base models.
- Now we have base models and meta model. We will test this on test data.

Unsupervised ML

K-means clustering

- It is a unsupervised learning algorithm use to create clusters or groups of similar instances or data points.
- In K-means clustering, K number of clusters are created in a dataset and every data point is allocated to nearest cluster.
- Initially it will take K number of centroids at random positions which are used as beginning points for every cluster. Then it will create clusters by finding nearest datapoints by measuring Euclidean distance.
- Then it will finds mean of all data points of particular cluster and updates the centroid to that mean position.
- This step repeated until centroids are stabilized at one place and then we get good clusters in such a way that sum of distance between centroid and datapoints of clusters is minimum.
- To find the best number of cluster, we can use elbow method in which we finds the WCSS for range of K values. WCSS is Within Cluster Sum Square. it is summation of squared distance between each datapoints and its centroid for all clusters.
- Then we will plot a graph of WCSS and K values, and the point where last sharp bend found is consider as best value of K.
- Disadvantage :
- We have to predefined the number of clusters, and it always tries to create the clusters of the same size.

Hierarchical clustering

- Hierarchical clustering is a unsupervised learning algorithm which is use for clustering.
- In this algorithm we create the hierarchical series of clusters which looks like a tree and this is also called as dendrogram.
- To create this hierarchy of clusters there are two methods, agglomerative and divisive.
- Agglomerative Hierarchical Clustering is a bottom-up approach, in which the algorithm starts with taking all data points as separate clusters and then start combining the closest pair of clusters together until only one cluster is left.
- Divisive Hierarchical clustering is top-down approach, it is opposite to the Agglomerative Hierarchical Clustering. In this we starts with only one cluster which contains all datapoints and then at each iteration, we split the farthest point in the cluster and repeat this process until each cluster only contains a single data point.
- To measure the distance between two clusters there are various methods which are also called as linkage methods:
  - single linkage : shortest distance between the closest points of two clusters,
  - complete linkage : farthest distance between two points of two clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.
  - Average Linkage : It is the linkage method in which the distance between each pair of data points are added up and then divided by the total number of data points to calculate the average distance between two clusters. It is also one of the most popular linkage methods.
  - Centroid Linkage : It is distance between centroids of clusters.

- Now using this Dendrogram, to find optimal number of clusters for our model, we will find the maximum vertical distance that does not cut any horizontal bar. The number of clusters will be the number of vertical lines which are being intersected by the line drawn using the threshold.

Advantages :

- No need to predefined the number of clusters.

Disadvantage :

- It took more time than K-means clustering because it has to create hierarchy from all data points to create single cluster.

K-Means and Hierarchical Clustering both fail in creating clusters of arbitrary shapes. They are not able to form clusters based on varying densities. That’s why we need

DBSCAN clustering.

- DBSCAN (Density-based spatial clustering of applications with noise)
- DBSCAN is popular unsupervised learning algorithm which is use for clustering problem.
- DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise
- It groups 'densely grouped' data points into a single cluster.
- In DBSCAN there are 2 main parameters, epsilon and minPoints. Epsilon is the radius of the circle to be created around each data point to check the density and minPoints is the minimum number of data points required inside that circle
- DBSCAN creates a circle of epsilon radius around every data point and classifies them into Core point, Border point, and Noise.
- A data point is a Core point if the circle around it contains at least 'minPoints' number of data points which we have initially define as a parameter.
- If the number of points are less than minPoints, then it is classified as Border Point.
- And if there is no datapoint in circle of any data point within epsilon radius, then it is consider as noise, which is a outlier & it is seperated away from any cluster.
- For locating datapoints, it uses euclidean distance
- DBSCAN is very sensitive to the values of epsilon and minPoints, therefore it is very important to select right values of epsilon and minPoints, because even a small change in this parameters can significantly affect the result of DBSCAN.
- Value of minPoints should be 1 more than number of dimensions. It should be at least 3. Value of minPoints is generally twice the number of dimensions, but there is no fix rule, it also depends on domain and complexity of problem.
- Value of epsilon can be choose from k-distance graph.

Advantages :

- It is robust to outliers
- No need to predefined the number of clusters.
- It can identify clusters in large spatial datasets by looking at the local density of the data points.

silhouette score

- silhouette score is a metric used to calculate the goodness of a clusters. Its value ranges from -1 to 1.
- If its value is close to 1 then clusters are well apart from each other. If the value is close to -1 then clusters are assigned in a wrong way.
- $Silhouette\ Score\ = \dfrac{b-a}{\max(a,b)}$

  where,

  a = average intra-cluster distance i.e. the average distance between each point within a cluster.

  b = average inter-cluster distance i.e. the average distance between all clusters.

ML Concepts

ML Project Pipeline :

1. Understanding of problem statement and business requirements
2. gathering require quality data from relevant source.
3. understanding data, all features and its impact on business
4. cleaning and structuring data using relevant tools or feature engineering
5. Exploratory Data Analysis (EDA),
6. Feature Selection
7. Model training
8. Model Evaluation and Hyperparameter tuning
9. Automation of end to end pipeline
10. Deployment
11. Continuous maintenance

Outliers

- Outlier is a datapoint which is far away from the observation.

How outliers are introduce in data :

- Data entry error (ground truth)
- Measurement error (instrumental error)
- Natural error (exceptional cases - most of the outliers belong to this category)
- Intentional error (dummy error)
- Sampling error (mixing of data from wrong resources)

Impact of outliers :

- It will impact on the mean and std of data.
- Reduce the power of some algorithm (sensitive to some algorithms : distance base and gradient base)

How to detect outliers :

Box plot :

- Boxplot displays a distribution of data by with five numbers, which has upper tail (Q3 + 1.5*IQR), Q3(75%), median, Q1(25%), lower tail (Q1 - 1.5*IQR). IQR is inter quartile range which can be calculated as Q3 - Q1.
- Points beyond upper tail and points below lower tail are outliers.

z-score :

- z-score is also called as standard score, because its formula is same as a standard scaler : (X - Xmean) / std.
- It centres the data to its mean with unit std, so it tells us how many std away a data point is.
- if z-score is > 3 or < -3, then we consider it as outlier.

IQR :

- IQR is inter quartile range which can be calculated as Q3 - Q1. It is middle 50% values.
- Q1 is a first quartile, it is 25th percentile value of data
- Q3 is a third quartile, it is 75th percentile value of data

- we can find this value by either df.describe method, np.percentile(data, 1-100) method or np.quantile(data, 0.1-1) method
- Then we find the upper tail and lower tail
- upper tail = Q3 + 1.5*IQR
- lower tail = Q1 - 1.5*IQR
- Points beyond upper tail and points below lower tail are outliers.

How to handle the outliers :

- before handling outliers, it is good practice to make data normally distributed if not.

Flooring & Capping method:

- In this method first we calculate the 90th and 10th percentile value, then if any value is more than 90th percentile value then it is replace by that 90th percentile value which is called as capping, similarly if value is less than 10th percentile value then it is replace by 10th percentile value which is called as flooring.

Mean/Median replacement :

- In this method outlier is replaced by mean or median.
- If the data is continuous, then mean of data excluding outlier can be taken for replacement.
- If the data is discrete then median can be taken.

Dropping outliers:

- This should be last preference because data is the fuel for ML.

Missing Values

- To check missing values we can use df.isna().sum() function

How to handle missing values :

mean/median/mode imputation :

- We can impute missing values by mean in case of continuous data, median in case of discrete data and mode in case of categorical data. We can also customize this imputation method as per nature of data and business.
- Suppose our target variable contains binary classes 0 and 1. If there is independent feature which contains missing values and its mean or median or mode is different for data which having target class as 0 and data which having target class 1. In this case imputing all missing values with single mean/median/mode of all data is not good strategy, so we can take mean/median/mode of specific data w.r.t target classes. Similarly if target variable is continuous then we can separate independent variables w.r.t some specific range of target value as per business complexity.

forward fill & backward fill :

- In forward fill we impute the missing value by previous value. fillna(method = 'ffill')
- In backward fill we impute the missing value by next value. fillna(method = 'bfill')
- In cases where data is continuous and aranged in sequential manner, ffill and bfill is more appropriate.

IterativeImputer :

- It is a multivariate method in which 2 or more features are consider for imputing missing values in one feature.
- It is imported from sklearn.impute library
- It uses all features as an independent feature except feature in which we have to impute missing values and feature which contain missing values is a target variable.
- For all the rows where target variable is not having missing value sklearn IterativeImputer runs regression model and then make a predictions where values are missing. This predicted values will be imputed values.

KNNImputer :

- It uses KNN algorithm at backend and uses Euclidean Distance to find nearest neighbours.
- Suppose there are 3 features and 1 feature is having missing value, so it will finds closest values for other two features and then base on this neighbours it will impute the value for missing place.

Anomaly detection:

- In some cases like fraud detection detecting outlier is more important than regular datapoints. It is called anomaly detection

Encoding

- Encoding is a method of converting categorical feature into numeric, because machine only understands numbers.

Types of categorical variables:

- Nominal variable : In this different categories not having relationship with each other or there is no order in it.
  eg. Color : red, green, blue

- Ordinal variable : In this categories are having order it between them.
  eg. rank : first, second, third
  education : high school, college, degree, PG,phd

Types of encoding :

One Hot Encoding :

- It is use for nominal categorical variable.
- It splits the categories in different columns and puts 0 if category is not in the row and 1 if it is present.
- We can implement this method by using pd.get_dummies() or OneHotEncoder
- Disadvantage of this method is if there are lot of categories in variable then it will lead to curse of dimensionality, and it can affect the performance of model.

Label Encoding :

- This method is use for ordinal categorical data.
- It assigns labels to each category starting from 0.

Feature Scaling

- Feature Scaling is a process of rescaling all the features into same scale.

Why feature scaling is important ? In which types of models it is important ? How feature with different scale impacts on model performance ?

- In Gradient Descent based and Distance based algorithms Feature Scaling is required, because :

Gradient Descent Based Models:

- In Gradient Descent based models, if we have features with different scale, then step size while convergence in gradient descent will be different for different features, and we will not having smooth convergence. So for smooth convergence and to have same step size for all features, we need to scale all features into same scale.
- Linear regression, Logistic regression are the gradient descent base models.

Distance Based Models:

- In Distance based models, if we have features with different scale, then it will give higher weightage to feature which having higher magnitude. It will affect the performance of the model. Also we don't want this biasness in our model.
- KNN, SVM are the distance based models.

Tree Based Models:

- Tree Based algorithms are not sensitive to the scale of the feature, because decision tree is only splitting a node based on single feature only, this split on feature is not influence by other features.

Types of Feature Scaling :

Normalization :

- Normalization converts the values in the range of 0 to 1.
- It is also called as MinMax Scaler.
- It's formula is,
  X' = (X - Xmin) / (Xmax - Xmin)

Standardization :

- Standardization is another scaling technique where the values are centred around the mean with a unit standard deviation. This means that the mean of the data becomes zero and the resultant distribution has a unit standard deviation.
- It's formula is,
  X' = (X - u) / sigma

Normalization Vs Standardization :

- Normalization is good to use when data does not follow a Gaussian distribution. This can be useful in algorithms that do not assume any distribution of the data like K-Nearest Neighbours and Neural Networks.
- Standardization, on the other hand, can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range. So, even if you have outliers in your data, they will not be affected by standardization.
- At end of the day, the choice of using normalization or standardization will depend on your problem and the machine learning algorithm you are using. There is no hard and fast rule to tell you when to normalize or standardize your data. You can always start by fitting your model to raw, normalized and standardized data and compare the performance for best results.

Imbalance Data

- When the target class has uneven distribution of data then it is imbalance data. Imbalance data can impact our accuracy.
- Examples of Imbalance data :
  Fraud Detection
  Spam filteration
  disease screening

Problem with imbalance data :

- If the dataset is imbalance, we can get high accuracy for prediction of majority class, but it will fail to capture minority class.

Handling the imbalance data :

Resampling technique :

- Oversampling : Adding copies of minority class
- Undersampling : Removing samples from majority class
- Oversampling is widely use method, because in undersampling data is lost.
  By using imblearn library we can use different resampling methods.

Random Sampler :

- For oversampling it will randomly add copies of minority class.
- For undersampling it will randomly deletes or removes samples of majority class
- from imblearn.over_sampling import RandomOverSampler
- RandomOverSampler().fit_resample(x,y)
- Similar for RandomUnderSampler

SMOTE & its Variants :

SMOTE (Synthetic Minority Oversampling Technique) :

- This technique generates synthetic data for minority class.
- SMOTE picks datapoint from minority class and finds K number of nearest neighbours for this point. Then it will generates random synthetic datapoints between sample and its neighbours.
- from imblearn.over_sampling import SMOTE
  smote = SMOTE(k =)
  smote.fit_resample(x,y)

Problem with Vanilla SMOTE :

- SMOTE can generate noisy minority data for classes where borderlines between classes are not well defined.
- In real life data majority of the time we don't have clear borderlines between classes. There can be some isolated minority points deep inside majority class region, borderlines may overlapped, there can be different clusters of minority class. In this type of cases SMOTE can generate more noisy data.

- To deal with this issues there are some variants of SMOTE.

SMOTETomek :

- It uses oversampling and undersampling at same time. It does oversampling of data using SMOTE and undersampling using Tomek link.
- First it does oversampling using SMOTE and then undersampling process starts using TomekLinks.
- If suppose two datapoints are mutual closest neighbours and if they belongs to two different classes, then they form a pair of Tomek link. These pairs of samples are often at the ambiguous borderline between classes, where it is easily to have false classification.
- The idea behind SMOTETomek is to make the training dataset cleaner by removing this Tomek links at the borderlines.
-
  ```
  from imblearn.combine import SMOTETomek
  from imblearn.under_sampling import TomekLinks
  resample=SMOTETomek(tomek=TomekLinks(sampling_strategy='all'), random_state=1)
  X_rs, y_rs = resample.fit_resample(X, y)
  ```

SMOTEENN :

- It also does oversampling and undersampling like SMOTETomek.
- It uses Wilson's Edited Nearest Neighbour rules (ENN) in the under-sampling step to remove instances of the majority classes.
- For each sample it calculates K Nearest Neighbours (kNN)
- Then it calculates the ratio of majority class among the kNN which also called 'r'.
- If sample is from minority class and r is >0.5, then it removes majority class instances.
- If sample is from majority class and r i <0.5, then it removes sample.
- SMOTEENN removes more samples than SMOTETomek.
-
  ```
  from imblearn.combine import SMOTEENN
  resample=SMOTEENN(random_state=0)
  X_rs, y_rs = resample.fit_resample(X, y)
  ```

Similarly lot of other variants of SMOTE are there like BorderlineSMOTE, SVMSMOTE, KMeansSMOTE and so on...

Feature Selection

- It is a way of selecting a subset of most relevant features set by removing the redundant, irrelevant or noisy features.
- It helps to reduce overfitting, improves accuracy, reduce training and testing time, also it helps in avoiding curse of dimensionality.

Feature Selection methods :

Filter Method :

- In Filter method, features are selected on the basis of statistical measures. This methods does not depend on learning algorithm and chooses the features as a preprocessing step.
- Filter methods are computationally efficient.

1. 1) Correlation :
   1. Pearson correlation coeff (continuous vs continuous)
   2. Spearman correlation coeff (descrete vs descrete)
   3. Kendall correlation coeff (categorical vs continuous or cont vs cat)
2. Information Gain (cat vs cat)
3. Fishers Score
4. Missing Value Ratio
5. Variance threshold method :
   - It removes all the features which doesn't meet our threshold.
   - By default it removes features having 0.0 variance.
   -
     ```
     from sklearn.feature_selection import VarianceThreshold
     var_thresh = VarianceThreshold(threshold = 0.0)
     var_thresh.fit(df)
     var_thresh.get_support()
     ```

6. Chi-Square test (cat vs cat)
7. Anova test (Continuous vs Categorical or Cat vs cont)
8. Mean Absolute Difference

Wrapper Method :

- It uses ML algorithms to find best subset of features.
- In wrapper methods, we use subset of features to train model, based on the result of this model we decide to add or remove features from subset.
- This methods are computationally more expensive.

1. Forward Feature Selection :
   - It is an iterative method in which we start with having no feature in the model.
   - In each iteration, we keep adding the feature which improves the performance of our model, till an addition of new feature dœs not improves performance of model.

2. Backward Elimination :
   - It is opposite to Forward feature selection method.
   - In this we start with all features and removes the least significant feature at each iteration which improves performance ofour model. It repeats until no improvement is observe on removal of feature.

3. Exhaustive Feature Selection
4. Recursive Feature Selection
5. Embedded Method :
   a. Random Forest Importance
   b. AdaBoost Importance
   c. Regularization (L1 >> Lasso)

Dimensionality Reduction

Principle Component Analysis (PCA)

- PCA is a dimensionality reduction method used to reduce dimensions of large dataset, by using orthogonal transformation. This new transformed features are called

Principle Components.

- PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.
- PCA works by considering the variance of each feature, because high variance feature can shows the good split between classes, and hence it reduces the dimensionality.
- Some properties of these principal components are given below:
  - The principal component must be the linear combination of the original features.
  - These components are orthogonal, i.e., the correlation between a pair of variables is zero.
  - The importance of each component decreases when going from 1 to n, it means the 1st PC has the most importance, and nth PC will have the least importance.
- PCA is based on some mathematical and statistical concepts like covariance, matrix, eigen values and eigen vectors.
- First we find the mean of features, then
- Finds the covariance and create covariance matrix
- Then we will find the eigen values and eigen vectors of this covariance matrix
- This Eigen Vectors are nothing but Principle Components.
- Principal component analysis preserves the essence of original data while compressing it, just like the television which instead of having two dimensional screen, can represent a three dimensional data without information loss!