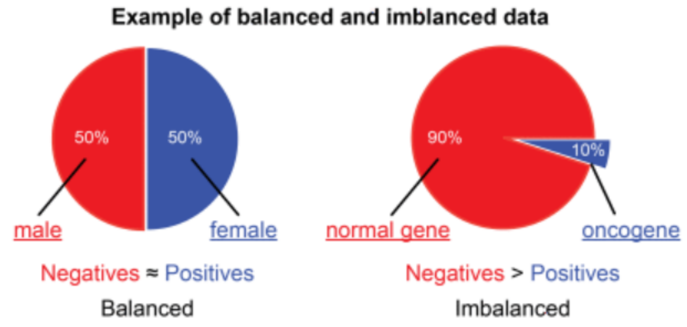
Classification algorithm in various situation

Balance Dataset -

If in our data set we have positive values which are approximately same as negative values. Then we can say our dataset in balance



Imbalance Dataset -

If there is the very high different between the positive values and negative values. Then we can say our dataset in Imbalance Dataset.

Problem with an Imbalanced Datasets -

Common problems are - Electricity Theft, Fraudulent Transactions in banks, Identification of rare Diseases etc.

Let say we are training our model on detecting the fraud detection. But here's the catch. The fraud transaction is relatively rare. So you start to training you model and get over 95% accuracy and get over 95% accuracy. When we give inputs to our model so our model is predicting "Not a Fraud Transaction" every time.

This is clearly a problem because many machine learning algorithms are designed to maximize overall accuracy.

Now what happen?? You get 95% accuracy but your model in predicting wrong every time??

Techniques to Convert Imbalanced Dataset into Balanced Dataset

Imbalanced data is not always a bad thing, and in real data sets, there is always some degree of imbalance. That said, there should not be any big impact on your model performance if the level of imbalance is relatively low.

1. Use the right evaluation metrics
2. Over-sampling (Up Sampling)
3. Under-sampling (Down Sampling)

4. Feature selection
5. Cost-Sensitive Learning Technique
6. Ensemble Learning Techniques

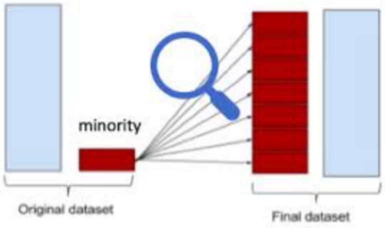
1. Use the right evaluation metrics-

- **Confusion Matrix-** A table showing correct predictions and types of incorrect predictions.
- **Precision-** The number of true positives divided by all positive predictions. Precision is also called Positive Predictive Value. It is a measure of a classifier's exactness. Low precision indicates a high number of false positives.
- **Recall-** The number of true positives divided by the number of positive values in the test data. Recall is also called Sensitivity or the True Positive Rate. It is a measure of a classifier's completeness. Low recall indicates a high number of false negatives.
- **F1-Score-** The weighted average of precision and recall.

2. Over-sampling (Up Sampling)-

Over-sampling increases the number of minority class members in the training set. The advantage of over-sampling is that no information from the original training set is lost, as all observations from the minority and majority classes are kept. On the other hand, it is **prone to over fitting.**

This technique is used to modify the unequal data classes to create balanced datasets. When the quantity of data is insufficient, the oversampling method tries to balance by incrementing the size of rare samples.



Advantages

- Unlike under sampling this method leads to no information loss.
- Outperforms under sampling

Disadvantages

- It increases the likelihood of overfitting since it replicates the minority class events.

2.1 Informed Over Sampling: Synthetic Minority Over-sampling Technique for imbalanced data (SMOTE) -

This technique is followed to avoid overfitting which occurs when exact replicas of minority instances are added to the main dataset. A subset of data is taken from the minority class as an example and then new synthetic similar instances are created. These synthetic instances are then added to the original dataset. The new dataset is used as a sample to train the classification models.

Eg - Total Observations = 1000
Fraudulent Observations = 20
Non Fraudulent Observations = 980
Event Rate = 2 %

A sample of 15 instances is taken from the minority class and similar synthetic instances are generated 20 times
Post generation of synthetic instances, the following data set is created

Minority Class (Fraudulent Observations) = 300
Majority Class (Non-Fraudulent Observations) = 980

Event rate= 300/1280 = 23.4 %

Advantages

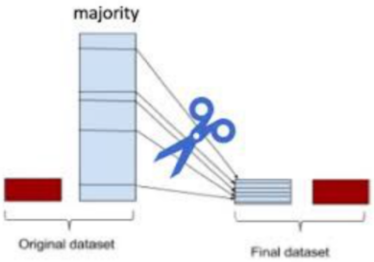
- Reduce the problem of overfitting caused by random oversampling as synthetic examples are generated rather than replication of instances
- No loss of useful information

Disadvantages

- While generating synthetic examples SMOTE does not take into consideration neighboring examples from other classes. This can result in increase in overlapping of classes and can introduce additional noise
- SMOTE is not very effective for high dimensional data

3. Under-sampling (Down Sampling)-

Under-sampling aims to reduce the number of majority samples to balance the class distribution. Since it is removing observations from the original data set, it might discard useful information.



This technique balances the imbalance dataset by reducing the size of the class which is in abundance. There are various methods for classification problems such as cluster centroids and Tomek links. The cluster centroid methods replace the cluster of samples by the cluster centroid of a K-means algorithm and the Tomek link method removes unwanted overlap between classes until all minimally distanced nearest neighbors are of the same class.

4. Feature selection -

In order to tackle the imbalance problem, we calculate the one-sided metric such as Correlation Coefficient (CC) and Odds Ratios (OR) or two-sided metric evaluation such as Information Gain (IG) and Chi-Square (CHI) on both the positive class and negative class. Based on the scores, we then identify the significant features from each class and take the union of these features to obtain the final set of features. Then, we use this data to classify the problem.

Identifying these features will help us generate a clear decision boundary with respect to each class. This helps the models to classify the data more accurately. This performs the function of intelligent subsampling and potentially helps reduce the imbalance problem.

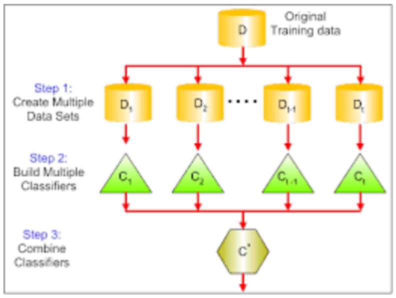
5. Cost-Sensitive Learning Technique -

The Cost-Sensitive Learning (CSL) takes the misclassification costs into consideration by minimising the total cost. The goal of this technique is mainly to pursue a high accuracy of classifying examples into a set of known classes. It is playing as one of the important roles in the machine learning algorithms including the real-world data mining applications.

6. Ensemble Learning Techniques -

It combines the result or performance of several classifiers to improve the performance of single classifier. This method modifies the generalisation ability of individual classifiers by assembling various classifiers.

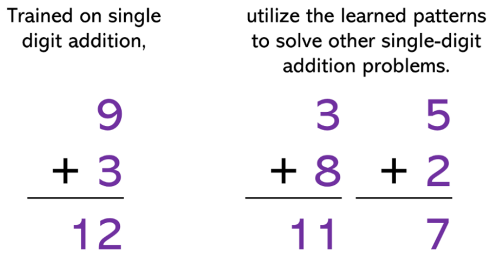
It mainly combines the outputs of multiple base learners. There are various approaches in ensemble learning such as Bagging, Boosting, etc.


D
Original Training data
Step 1: Create Multiple Data Sets
D_1
D_2
D_{t-1}
D_t
Step 2: Build Multiple Classifiers
C_1
C_2
C_{t-1}
C_t
Step 3: Combine Classifiers
C^*

Generalization vs Extrapolation -

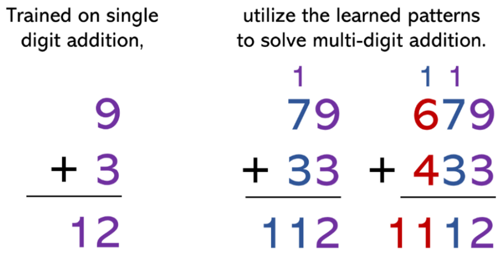
Generalization is the entire point of machine learning. Trained to solve one problem, the model attempts to utilize the patterns learned from that task to solve the same task, with slight variations.

Consider a child being taught how to perform single-digit addition. Generalization is the act of performing tasks of the same difficulty and nature. Its also called Interpolation.


Trained on single digit addition,
utilize the learned patterns to solve other single-digit addition problems.
9
+ 3
12
3
+ 8
11
5
+ 2
7

Extrapolation is when the model is able to obtain higher-dimensional insights from a lower-dimensional training.

Consider a first grader who is taught single digit addition, then presented with a multi-digit addition problem. The first grader thinks, “okay, so when the units digit adds to larger than ten, there is a tens component and a ones component.


Trained on single digit addition,
utilize the learned patterns to solve multi-digit addition.
9
+ 3
12
1
79
+ 33
112
1 1
679
+ 433
1112

What is the impact of Outliers on a dataset?

Outliers can drastically change the results of the data analysis and statistical modeling. There are numerous unfavourable impacts of outliers in the data set:

- It increases the error variance and reduces the power of statistical tests
- If the outliers are non-randomly distributed, they can decrease normality
- They can bias or influence estimates that may be of substantive interest
- They can also impact the basic assumption of Regression, ANOVA and other statistical model assumptions.

How to detect Outliers?

Most commonly used method to detect outliers is visualization. We use various visualization methods, like **Box-plot, Histogram, Scatter Plot.**

There are various thumb rules to detect outliers are -

- Any value, which is beyond the range of -1.5 x IQR to 1.5 x IQR
- Use capping methods. Any value which out of range of 5th and 95th percentile can be considered as outlier
- Data points, three or more standard deviation away from mean are considered outlier
- Outlier detection is merely a special case of the examination of data for influential data points and it also depends on the business understanding
- Bivariate and multivariate outliers are typically measured using either an index of influence or leverage, or distance. Popular indices such as Mahalanobis’ distance and Cook’s D are frequently used to detect outliers.

How to remove Outliers?

Most of the ways to deal with outliers are similar to the methods of missing values like deleting observations, transforming them, binning them, treat them as a separate group, imputing values

Deleting observations: We delete outlier values if it is due to data entry error, data processing error or outlier observations are very small in numbers. We can also use trimming at both ends to remove outliers.

Transforming and binning values: Transforming variables can also eliminate outliers. Natural log of a value reduces the variation caused by extreme values. Binning is also a form of variable transformation. Decision Tree algorithm allows to deal with outliers well due to binning of variable. We can also use the process of assigning weights to different observations.

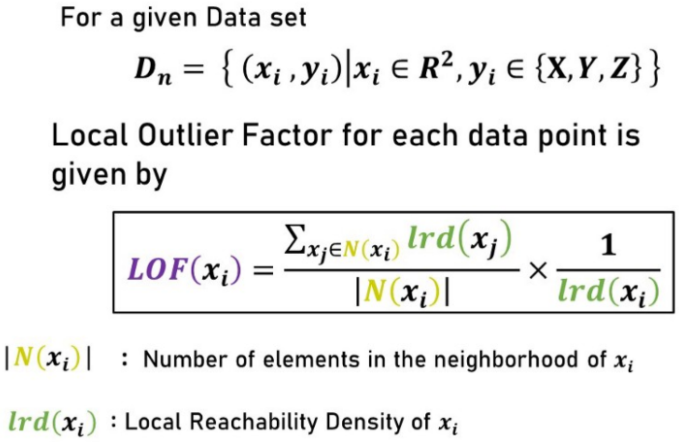
Imputing: We can use mean, median, mode imputation methods. Before imputing values, we should analyse if it is natural outlier or artificial. If it is artificial, we can go with imputing values. We can also use statistical model to predict values of outlier observation and after that we can impute it with predicted values.

Treat separately: If there are significant number of outliers, we should treat them separately in the statistical model. One of the approach is to treat both groups as two different groups and build individual model for both groups and then combine the output.

Local Outlier Factor -

Local Outlier Factor(LOF) is an algorithm used to detect outliers in any datasets.

Let us understand, in this algorithm, a score (scalar value) which is called as Local Outlier Factor (LOF) is the deciding factor. The mathematical expression of finding this factor is given below-



For a given Data set

$$\boldsymbol{D_n} = \{ (\boldsymbol{x_i}, \boldsymbol{y_i}) | \boldsymbol{x_i} \in \boldsymbol{R^2}, \boldsymbol{y_i} \in \{\mathbf{X}, \boldsymbol{Y}, \boldsymbol{Z}\} \}$$

Local Outlier Factor for each data point is given by

$$\boxed{\boldsymbol{LOF}(\boldsymbol{x_i}) = \frac{\sum_{x_j \in N(x_i)} \boldsymbol{lrd}(\boldsymbol{x_j})}{|\boldsymbol{N}(\boldsymbol{x_i})|} \times \frac{\mathbf{1}}{\boldsymbol{lrd}(\boldsymbol{x_i})}}$$

$|\boldsymbol{N}(\boldsymbol{x_i})|$: Number of elements in the neighborhood of x_i

$\boldsymbol{lrd}(\boldsymbol{x_i})$: Local Reachability Density of x_i

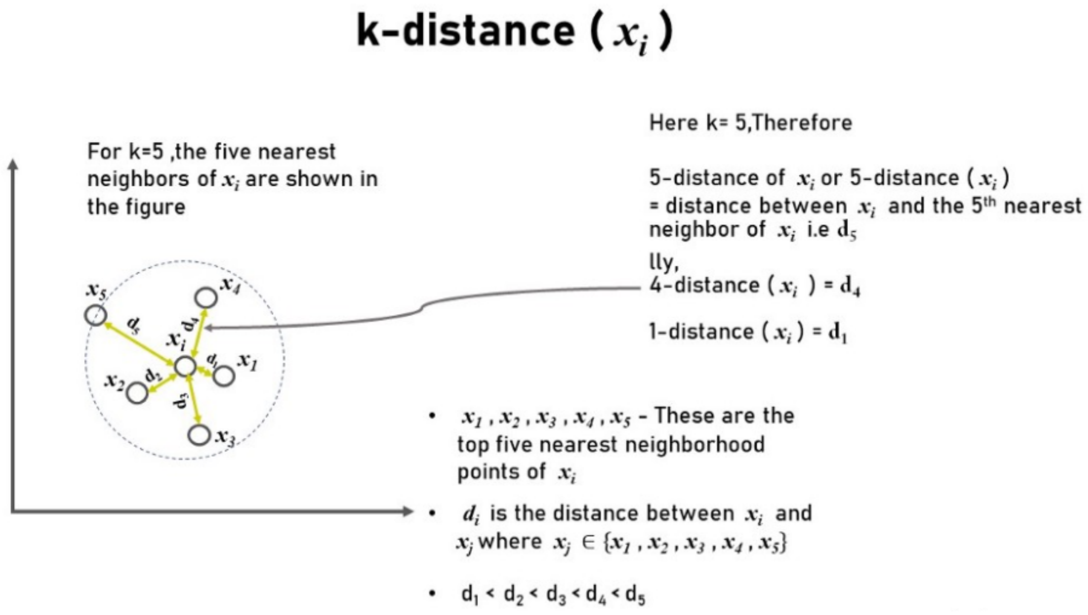
LOF is assigned to each data points, these assigned LOF scores of each data points are compared to find outliers. The more the value of LOF of any data point more the chance that it will be an outlier.

2.Important parameters used for calculating Local Outlier Factors(LOF)

1. k-distance (x_i)
2. Nearest Neighbor $N_k(x_i)$
3. Reachability Distance (x_i,x_j)
4. Local Reachability Density lrd(x_i)

1. k-distance (x_i)

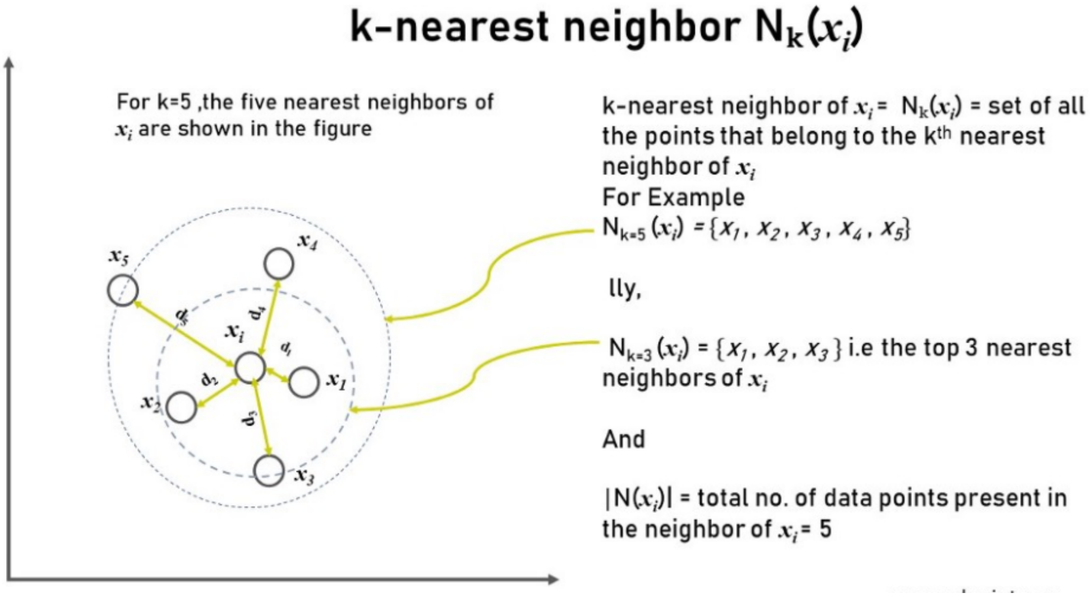
The k-distance of a data point x_i in a dataset is the distance of the kth nearest neighbour of x_i from x_i.



k-distance (x_i)
For k=5 ,the five nearest neighbors of x_i are shown in the figure
Here k= 5,Therefore
5-distance of x_i or 5-distance (x_i) = distance between x_i and the 5th nearest neighbor of x_i i.e d_5
lly,
4-distance (x_i) = d_4
1-distance (x_i) = d_1
x_1
x_2
x_3
x_4
x_5
x_i
d_1
d_2
d_3
d_4
d_5
• x_1 , x_2 , x_3 , x_4 , x_5 - These are the top five nearest neighborhood points of x_i
• d_i is the distance between x_i and x_j where $x_j \in \{x_1 , x_2 , x_3 , x_4 , x_5\}$
• $d_1 < d_2 < d_3 < d_4 < d_5$

2. Nearest Neighbor $N_k(x_i)$

The Nearest Neighbor of a data point x_i denoted by $N_k(x_i)$ is a set of all the points that belong to the kth nearest neighbour of x_i (i.e points in the neighborhood of x_i).



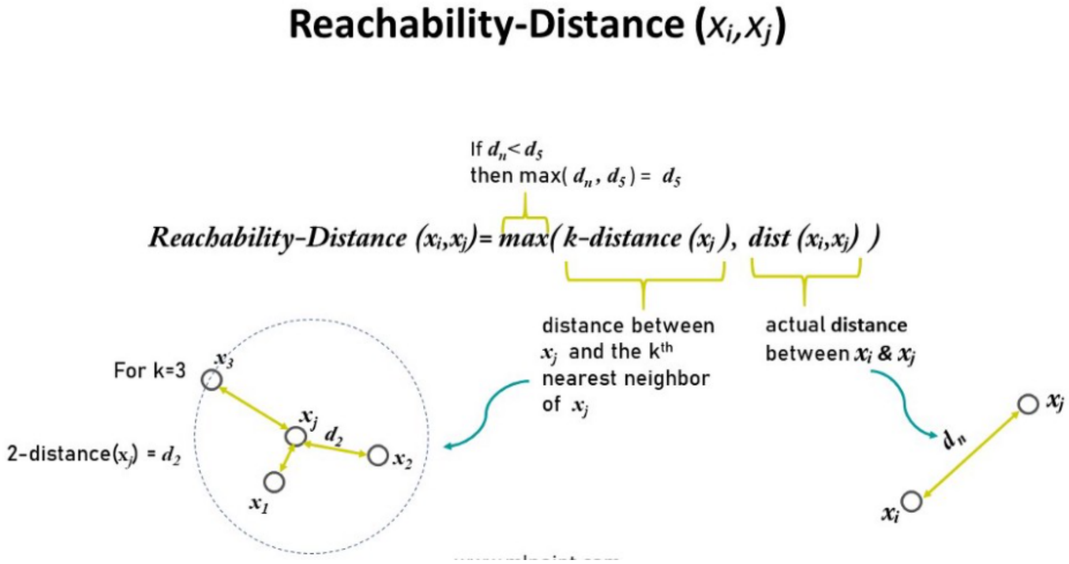
k-nearest neighbor $N_k(x_i)$
For k=5 ,the five nearest neighbors of x_i are shown in the figure
k-nearest neighbor of x_i = $N_k(x_i)$ = set of all the points that belong to the kth nearest neighbor of x_i
For Example
$N_{k=5}(x_i) = \{x_1, x_2, x_3, x_4, x_5\}$
lly,
$N_{k=3}(x_i) = \{x_1, x_2, x_3\}$ i.e the top 3 nearest neighbors of x_i
And
$|N(x_i)|$ = total no. of data points present in the neighbor of x_i = 5
x_1
x_2
x_3
x_4
x_5
x_i
d_1
d_2
d_3
d_4
d_5

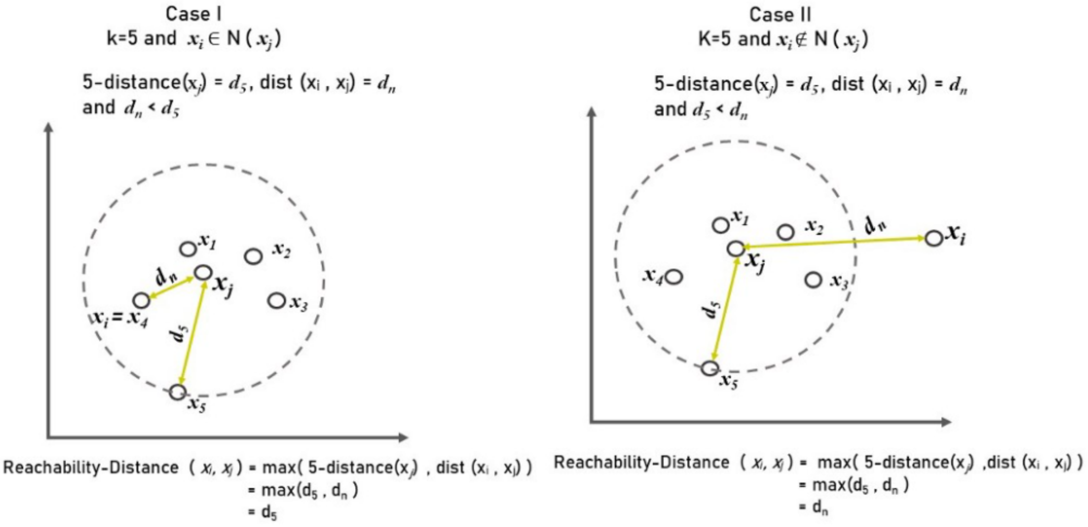
$|N(x_i)|$ denotes the total number of data points present in the neighborhood of x_i for the given k value.

3. Reachability-Distance (x_i,x_j)

The Reachability-Distance of a data point x_j from x_i is the maximum of k-distance of x_j and the actual distance between x_i & x_j. Mathematically,

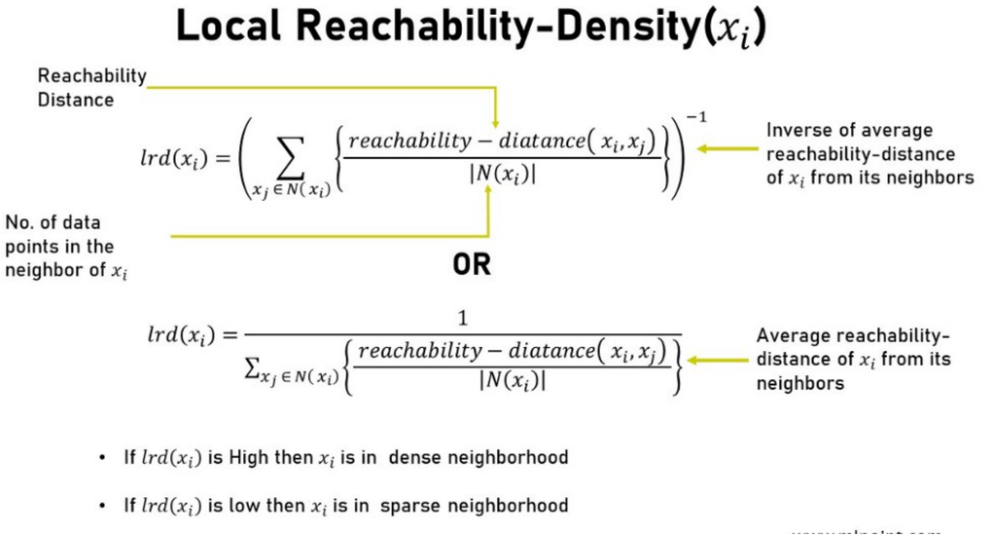
$$Reachability - Distance(x_i, x_j) = max(k - distance(x_j), dist(x_i, x_j))$$


Reachability-Distance (x_i,x_j)
If $d_n < d_5$
then max(d_n , d_5) = d_5
Reachability-Distance (x_i,x_j)= max(k-distance (x_j), dist (x_i,x_j))
distance between x_j and the kth nearest neighbor of x_j
actual distance between x_i & x_j
For k=3
x_3
x_j
d_2
x_2
x_1
2-distance(x_j) = d_2
x_j
d_n
x_i


Case I
k=5 and $x_i \in$ N (x_j)
5-distance(x_j) = d_5, dist (x_i , x_j) = d_n
and $d_n < d_5$
x_1
x_2
x_j
x_3
d_n
$x_i = x_4$
d_5
x_5
Reachability-Distance (x_i, x_j) = max(5-distance(x_j) , dist (x_i , x_j))
= max(d_5 , d_n)
= d_5
Case II
K=5 and $x_i \notin$ N (x_j)
5-distance(x_j) = d_5, dist (x_i , x_j) = d_n
and $d_5 < d_n$
x_1
x_2
d_n
x_i
x_4
x_j
x_3
d_5
x_5
Reachability-Distance (x_i, x_j) = max(5-distance(x_j) ,dist (x_i , x_j))
= max(d_5 , d_n)
= d_n

4. Local Reachability Density lrd(x_i)

The Local Reachability Density of a data point x_i is the inverse of the average reachability distance of x_i from its neighborhood. Basically , It measures how close the neighborhood data points of x_i from it. The lower the density, the farther x_i is from its neighbours.



Local Reachability-Density(x_i)

$$lrd(x_i) = \left(\sum_{x_j \in N(x_i)} \left\{ \frac{reachability - diatance(x_i, x_j)}{|N(x_i)|} \right\} \right)^{-1}$$

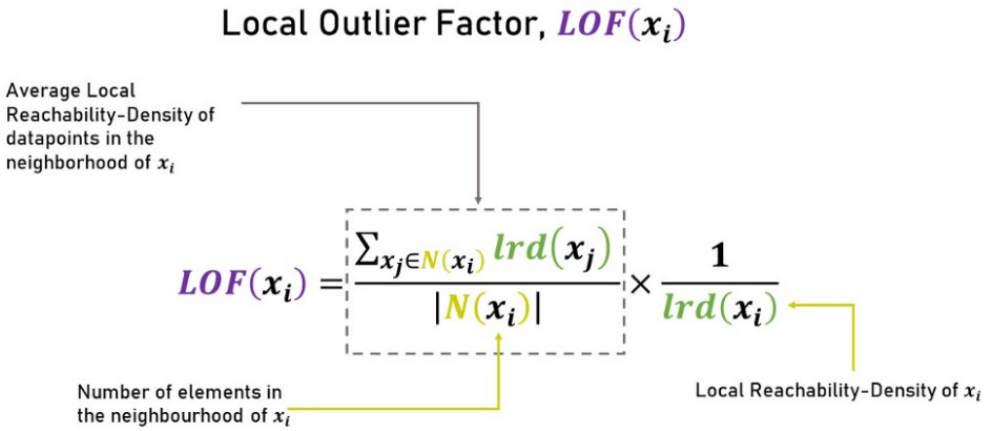
Reachability Distance
No. of data points in the neighbor of x_i
Inverse of average reachability-distance of x_i from its neighbors
OR

$$lrd(x_i) = \frac{1}{\sum_{x_j \in N(x_i)} \left\{ \frac{reachability - diatance(x_i, x_j)}{|N(x_i)|} \right\}}$$

Average reachability-distance of x_i from its neighbors

- If $lrd(x_i)$ is High then x_i is in dense neighborhood
- If $lrd(x_i)$ is low then x_i is in sparse neighborhood

3. Understanding Local LOF with important parameters

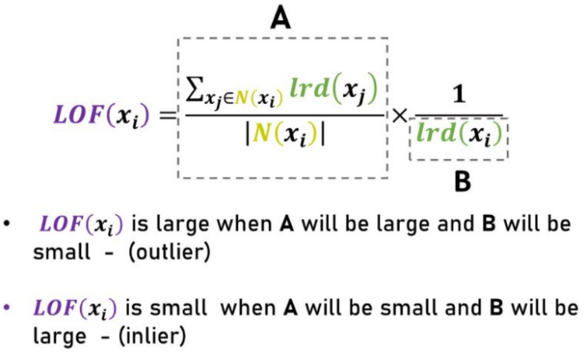


Local Outlier Factor, $LOF(x_i)$

$$LOF(x_i) = \frac{\sum_{x_j \in N(x_i)} lrd(x_j)}{|N(x_i)|} \times \frac{1}{lrd(x_i)}$$

Average Local Reachability-Density of datapoints in the neighborhood of x_i
Number of elements in the neighbourhood of x_i
Local Reachability-Density of x_i

The above shows the relation between Local Reachability Density lrd(x_i) and average local reachability- density of data points in the neighborhood of x_i.



$$LOF(x_i) = \underbrace{\frac{\sum_{x_j \in N(x_i)} lrd(x_j)}{|N(x_i)|}}_{A} \times \frac{1}{\underbrace{lrd(x_i)}_{B}}$$

- $LOF(x_i)$ is large when **A** will be large and **B** will be small - (outlier)
- $LOF(x_i)$ is small when **A** will be small and **B** will be large - (inlier)

LOF(x_i) ~ 1 means **Similar density as neighbors,**
LOF(x_i) < 1 means **Higher density than neighbors (Inlier),**
LOF(x_i) > 1 means **Lower density than neighbors (Outlier)**

LOF is also called as a density-based outlier detection method because it uses the relative density of data points against its neighbors to detect outliers. As the density around the outlier is significantly different from the density of its neighbors.

4. Implementation using Python -

Refer - https://medium.com/mlpoint/local-outlier-factor-a-way-to-detect-outliers-dde335d77e1a (https://medium.com/mlpoint/local-outlier-factor-a-way-to-detect-outliers-dde335d77e1a)

5.Conclusion

The LOF of a data point gives a score by comparing the density of that point to the density of its neighbors. If the density of a point is much smaller than the densities of its neighbors (LOF ≫1), the point is far from dense areas and, hence, an outlier.

Also, By varying k(n_neighbors) the outliers for the entire dataset or for the local small regions can be detected.

It is not the only method for anomaly detection, there are other methods such as DBSCAN, Isolation Forests, Elliptic Envelope e.t.c.

Feature Selection methods -

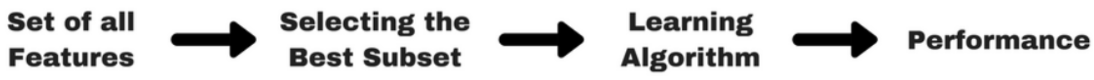
Table of Contents

1. Importance of Feature Selection -

We need not use every feature for creating an algorithm. We can assist our algorithm by feeding in only those features that are really important.

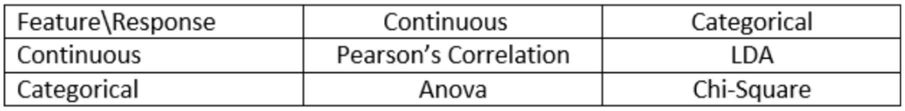
- It enables the machine learning algorithm to train faster.
- It reduces the complexity of a model and makes it easier to interpret.
- It improves the accuracy of a model if the right subset is chosen.
- It reduces overfitting.

2. Filter Methods -



Filter methods are generally used as a preprocessing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. The correlation is a subjective term here.

For basic guidance, we can refer to the following table for defining correlation co-efficients.



Feature\Response	Continuous	Categorical
Continuous	Pearson's Correlation	LDA
Categorical	Anova	Chi-Square

- **Pearson's Correlation-** It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1. Pearson's correlation is given as:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

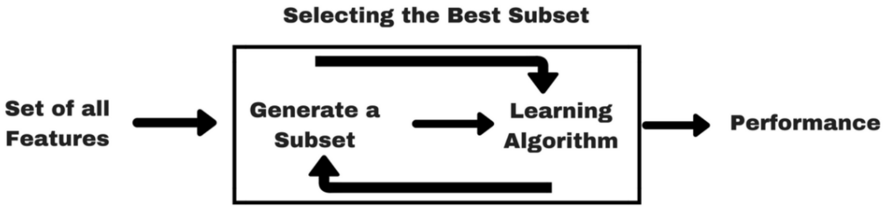
- **LDA-** Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.

- **ANOVA-** ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.

- **Chi-Square-** It is a is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.

One thing that should be kept in mind is that filter methods do not remove **multicollinearity**. So, you must deal with multicollinearity of features as well before training models for your data.

3. Wrapper Methods



In wrapper methods, we try to use a subset of features and train a model using them. Based on the inferences that we draw from the previous model, we decide to add or remove features from your subset. The problem is essentially reduced to a search problem. These methods are usually computationally very expensive.

Some common examples of wrapper methods are -

- **Forward Selection-** Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.

- **Backward Elimination-** In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.

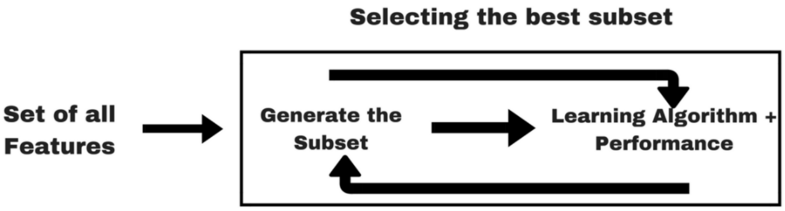
- **Recursive Feature elimination-** It is a greedy optimization algorithm which aims to find the best performing feature subset. It repeatedly creates models and keeps aside the best or the worst performing feature at each iteration. It constructs the next model with the left features until all the features are exhausted. It then ranks the features based on the order of their elimination.

One of the best ways for implementing feature selection with wrapper methods is to use Boruta package that finds the importance of a feature by creating shadow features.

It works in the following steps-

1. Firstly, it adds randomness to the given data set by creating shuffled copies of all features (which are called shadow features).
2. Then, it trains a random forest classifier on the extended data set and applies a feature importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important.
3. At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z-score than the maximum Z-score of its shadow features) and constantly removes features which are deemed highly unimportant.
4. Finally, the algorithm stops either when all features get confirmed or rejected or it reaches a specified limit of random forest runs.

4. Embedded Methods -



Embedded methods combine the qualities' of filter and wrapper methods. It's implemented by algorithms that have their own built-in feature selection methods.

Some of the most popular examples of these methods are LASSO and RIDGE regression which have inbuilt penalization functions to reduce overfitting.

- **Lasso regression** performs L1 regularization which adds penalty equivalent to absolute value of the magnitude of coefficients.
- **Ridge regression** performs L2 regularization which adds penalty equivalent to square of the magnitude of coefficients.

5. Difference between Filter and Wrapper methods -

- Filter methods measure the relevance of features by their correlation with dependent variable while wrapper methods measure the usefulness of a subset of feature by actually training a model on it.
- Filter methods are much faster compared to wrapper methods as they do not involve training the models. On the other hand, wrapper methods are computationally very expensive as well.
- Filter methods use statistical methods for evaluation of a subset of features while wrapper methods use cross validation.
- Filter methods might fail to find the best subset of features in many occasions but wrapper methods can always provide the best subset of features.

- Using the subset of features from the wrapper methods make the model more prone to overfitting as compared to using subset of features from the filter methods.

Feature Selection - https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/ (https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/)

Feature Importance

It means sort the feature which most important for classfication. It is useful in understanding a model better. It increases interpretability.

Eg Log-regression, Decision Tree.

In K-NN by default it does not have feature importance but we get by some modification.

Forward Feature Selection

It means selection of most important feature among a large no. of features. Eg from 1000 dim to 10 dim.

Process:-
Let we have features f= {f1,f2,f3...fd} and model F which is K-NN and its not trained yet, given the train and test dataset,, till now we dont have build the model.
Itr 1:- Now we take f1, take training and test data corresponding to that feature and train model f and we get accuracy(a1) on test data.
Itr 2:- Now we take f2, take training and test data corresponding to that feature and train model f and we get accuracy(a2) on test data.
.
.
Itr n:- Now we take fn, take training and test data corresponding to that feature and train model f and we get accuracy(an) on test data.

Now check which feature have highest accuracy let it is f10. Now rest we have feature f = {f1,f2...f9,f11..fd} note here we dont consider f10 a feature bcz its already sorted out. Sf={f10}

Now we repeat the above itration with (f1,f10), (f2,f10) ... (fd,f10). Now find the highest one let it is f6. So we get two imp feature which is f10,f6. Now rest we have feature f = {f1,f2,f4,f5,f7...f9,f11..fd} note here we dont consider f10,f6 features bcz its already sorted out. Sf={f10,f6}

Now we repeat the above itration with (f1,f10,f6), (f2,f10,f6) ... (fd,f10,f6). Now find the highest one let it is f6. So we get two imp feature which is f10,f6,f4. continue this till we dont reach desired no. of feature.

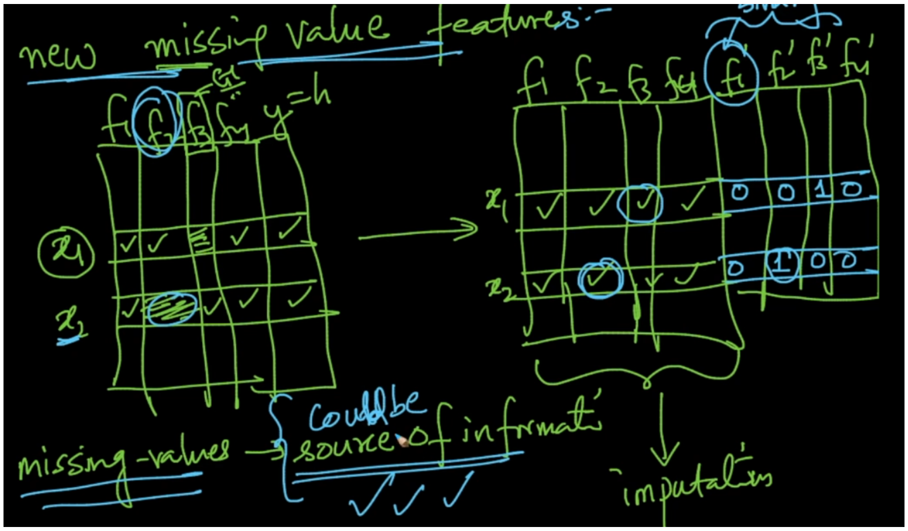
The above whole process is called Forward Feature Selection Process

Backword Feature Selection

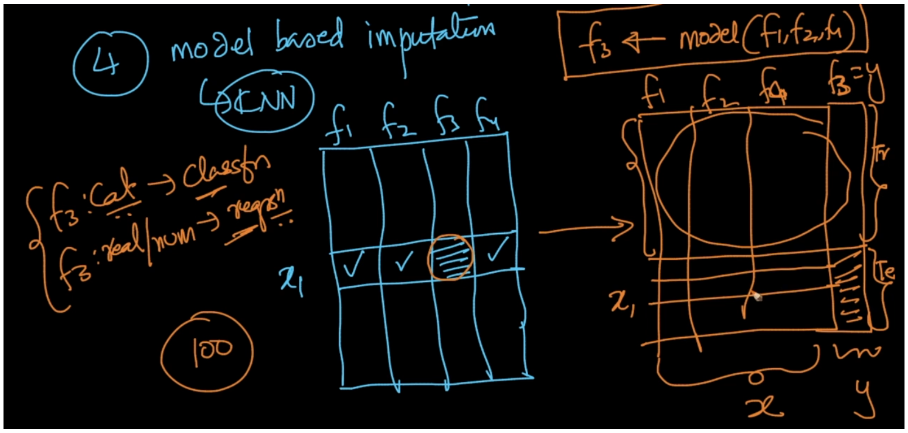
In this we first remove the feature which have lowest accuracy... same as above.

Handling Missing Value by Imputation -

1. We can replace non missing value to **mean, median and mode(for categorical label)** in dataset.
2. For classification task, we impute based on **class label**.
3. **New missing value feature(adding new set of feature) -**
   We will create new features(f1', f2'...fn') for all labels(f1, f2...fn). In new feature value is 1 for corresponding missing value and 0 for filled value.


new missing value features:-
f1 f2 f3 f4 y=h
x4
x2
f1 f2 f3 f4 f1' f2' f3' f4'
x1
x2
0 0 1 0
0 1 0 0
missing-values → could be source of information
imputation

4. Model based Imputation - We will use ML Model to imputed missing value, By splitting data into 2 half, first one having is used to train model and second one used to predict missing value. Mostly we use K-NN.



Curse of Dimensionality

As dimension increases the no. of datapoint to perform good classification for model increases exponenially and thus performance decreases

1.Hughes Phenomenon -

Perfomance decreases as dimensionality increses for finite no. of datapoint

2. Distance Function (Eucidean distance)

Let data is distributed uniformly and random K-NN does not work well in high dimension space. K-NN depends on Euclien distance, Euclien distance logically does not make sense in high dimension.

So solution is use K-NN with cosine-similarity, cos-sim also impacted high dim but less than Eucliean

Data having- high dim & dense - impact or curse of dimensionality is high high dim & sparse - impact of dimensionality is lower

3. Overfitting and Underfitting

When dim increases chances of Overfitting also increase.

For feature selection, we choose Forward feature selection over PCA, Tsne bcz

1. PCA and Tsne reduce random feature, dont use class label
2. PCA and Tsne is classification oriented

K-NN on Text-data-

- Prefer cos-sim over eculidean distance
- Perfer sparse(BoW) represn instead of dense(W2V) repres

Bias-Variance Tradeoff

For Naive-Bayes the Bais-Variance tradeoff are as follows-

- high-bias means Underfitting
- high-variance means Overfitting

$$GeneralizatioError = Bias^2 + Variance + IrreducibleError$$

Bias = Error due to simplfying assumption(high training error)

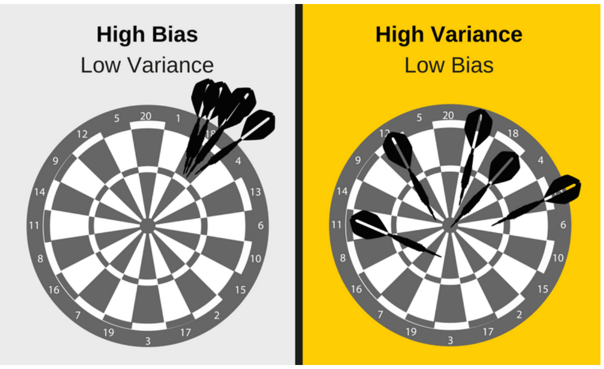
- Bias is the difference between your model's expected predictions and the true values.

Variance = A small change in training data result large change in model.

- High variance algorithms will produce drastically different models depending on the training set.
- How much a model(decision surface) changes as training data changes(**high test error**)
- Irreducible err = Noise or Error that we can not reduce further for a given model

Trade-off is tension between the error introduced by the bias and the variance.

- Algorithm cannot simultaneously be more complex and less complex.



Low variance (high bias) algorithms tend to be less complex, with simple or rigid underlying structure.

- They train models that are consistent, but inaccurate on average.
- These include linear or parametric algorithms such as regression and naive Bayes.

Low bias (high variance) algorithms tend to be more complex, with flexible underlying structure.

- They train models that are accurate on average, but inconsistent.
- These include non-linear or non-parametric algorithms such as decision trees and nearest neighbors.

High-Bias: Suggests more assumptions about the form of the target function. Eg Linear Regression, Linear Discriminant Analysis and Logistic Regression.

High Variance: Suggests large changes to the estimate of the target function with changes to the training dataset. a high-variance model is Overfit model Eg. Decision Trees, k-Nearest Neighbors and Support Vector Machines.

- alpha in Naive-bayes and K in K-NN are called Hyperparameter.
    - Correct Hyperparameter is decided using Cross-Validation

Refer-

1. Naive Bayes - https://elitedatascience.com/bias-variance-tradeoff (https://elitedatascience.com/bias-variance-tradeoff)
2. How to handle imbalance datset - https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/ (https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/)