**Company Bankruptcy Prediction**

**Turab Ahmed**

**AlmaBetter, Bangalore.**

**1.Introduction:**

Estimating the risk of corporate bankruptcies is of large importance to creditors and investors. There are large indirect and direct costs associated with financial distress and Corporate bankruptcies can have serious effects both locally and globally on employees, investors, customers, suppliers and their financiers are all affected when a company disappears in some cases a corporate bankruptcy can cause an entire industry to suffer.

Machine learning models have successfully been used for many classification and regression problems and these models have often outperform traditional classification methods. The purpose of bankruptcy prediction is to predict future perspectives of a company. For a given period, this problem can be modelled as a two-class classification problem. Companies either survive the given time period or go bankrupt during it. The problem is to predict which of these two possible outcomes that is the more probable one.

**2.Data and Problem Statement:**

The data used in this study are Taiwan Economic Journal for the years 1999 to 2009. The goal of the Machine Learning Model is to predict whether the company will go for bankruptcy or not. The data set contains 6819 rows and 96 columns, column **Bankrupt?** Is the target variable here.

**3. Steps involved:**

* **Exploratory Data Analysis:**

After loading the dataset we performed Exploratory Data Analysis (EDA) on understanding the data. We see what distribution all the variables follow through EDA.

* **Class Imbalance treatment:**

Our dataset contains class imbalance which might tend to disturb our accuracy hence we applied SMOTE (**Synthetic Minority Over-sampling Technique**) from oversampling method to have balanced data on our target variable

* **Feature Selection**

In these steps we checked the multicollinearity relationship between the variables, if it exists then we will drop those variable and its values.

* **Standardization of features**

Our main motive through this step was to scale our data to equal units that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it. We have used **MinMaxScaler** for Standardization

* **Fitting different models**

For modelling we tried various classification algorithms like:

1. **Logistic Regression**
2. **Decision Tree Classifier**
3. **Random Forest Classifier**
4. **K Nearest Neighbor**
5. **XGBoost classifier**
6. **SVM Model**
7. **Artificial Neural Network**

* **Tuning the hyperparameters for better accuracy**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy by selecting the learning rate (C).

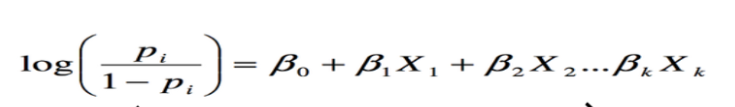
**4. Algorithms:**

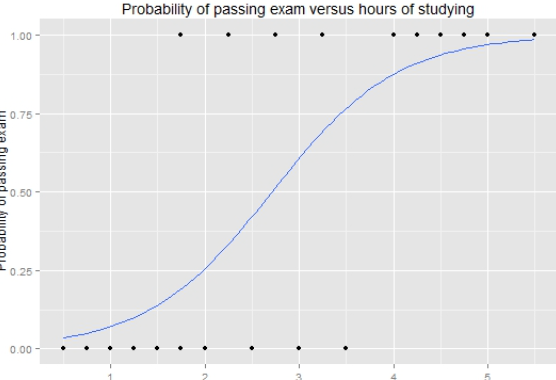
* 1. **Logistic Regression:**

Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a [data set](https://whatis.techtarget.com/definition/data-set). Logistic regression has become an important tool in the discipline of [machine learning](https://searchenterpriseai.techtarget.com/definition/machine-learning-ML). The approach allows an [algorithm](https://whatis.techtarget.com/definition/algorithm) being used in a machine learning application to classify

incoming data based on historical data. As more relevant data comes in, the algorithm should get better at predicting classifications within data sets.

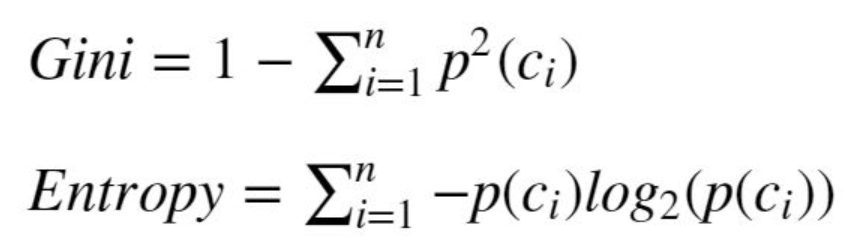
Logistic regression can also play a role in [data preparation](https://searchbusinessanalytics.techtarget.com/definition/data-preparation) activities by allowing data sets to be put into specifically predefined buckets during the extract, transform, load ([ETL](https://searchdatamanagement.techtarget.com/definition/Extract-Load-Transform-ELT)) process in order to stage the information for analysis.

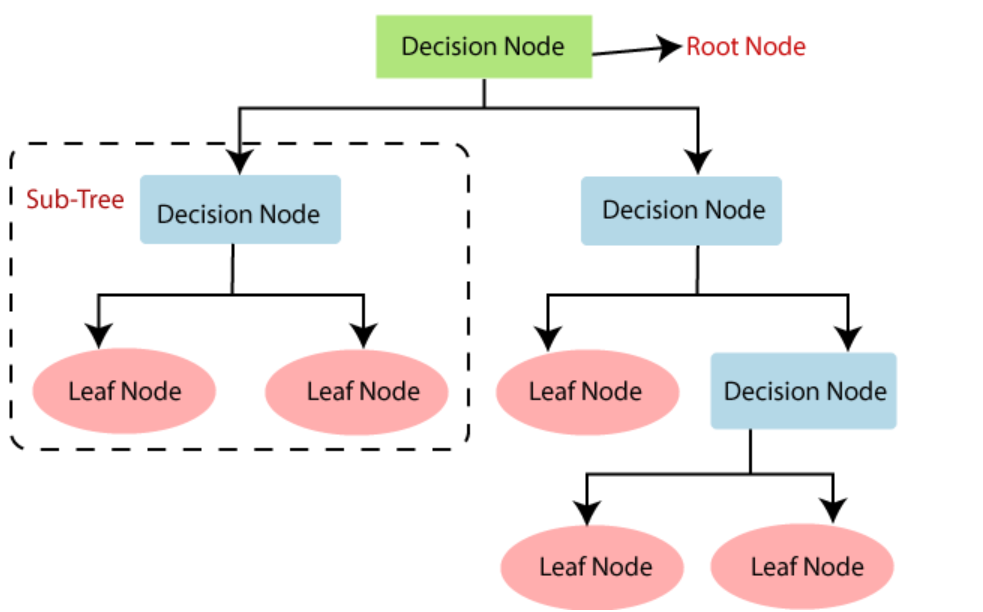




.**4.2 Decision Tree Classifier:**

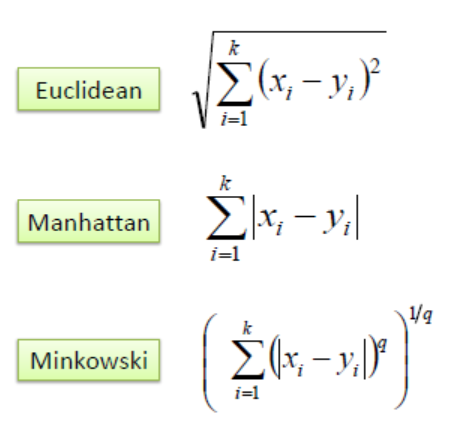
Decision Trees (DTs) are a **non-parametric supervised learning** method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

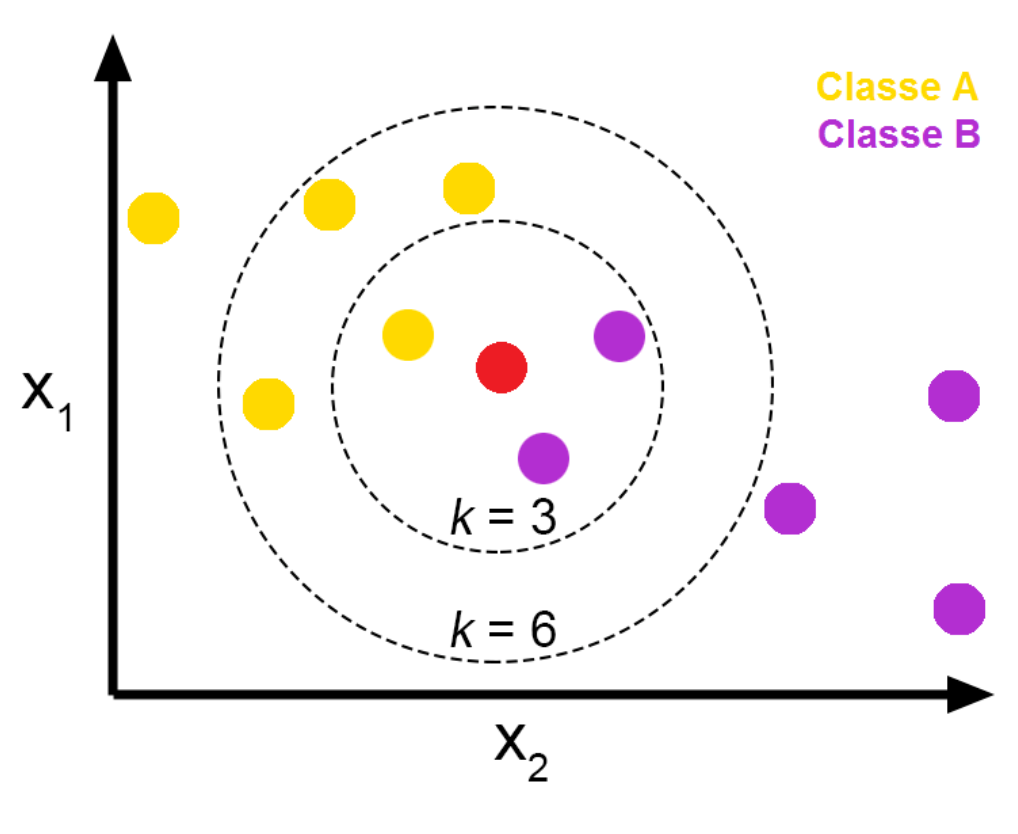




* 1. **K Nearest Neighbours:**

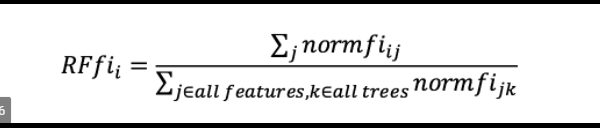
K-NN is a **non-parametric** algorithm, which means it does not make any assumption on underlying data.It is also called a **lazy learner** algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

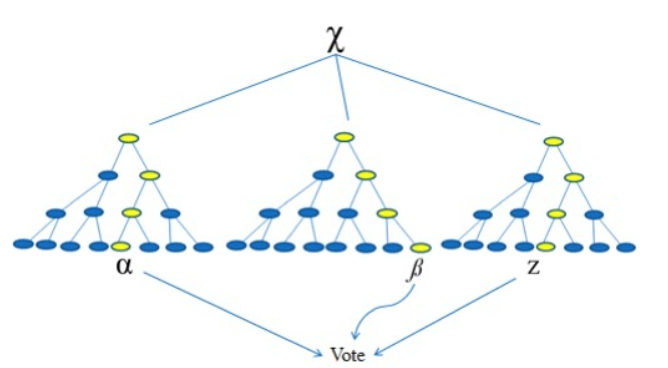




* 1. **Random Forest:**

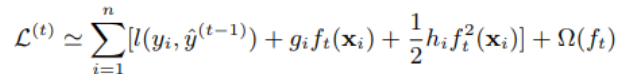
Random forest classifiers fall under the broad umbrella of ensemble-based learning methods. The algorithm uses **randomness** to build each individual tree to promote uncorrelated forests, which then uses the forest's predictive powers to make accurate decisions.

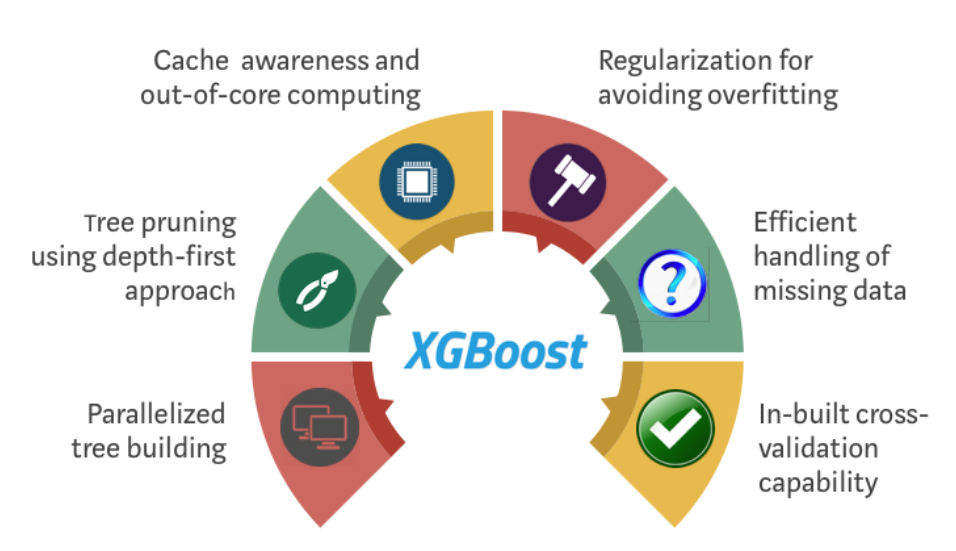




* 1. **XGBoost classifier:**

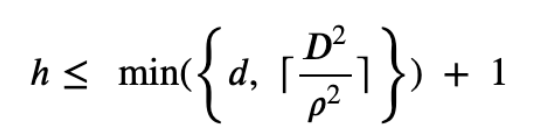
XGBoost is a gradient boosting package that implements a gradient boosting framework. The algorithm is scalable for parallel computing. XGBoost is an implementation of **gradient boosted decision trees** designed for speed and performance

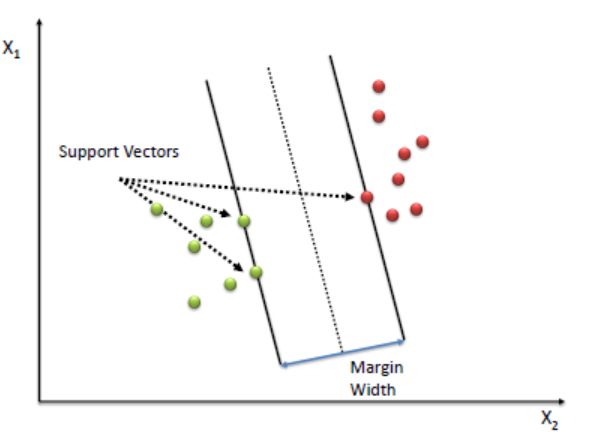




**4.5 SVM Model:**

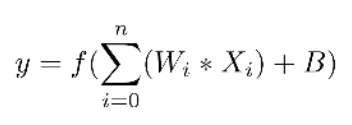
support-vector machines are [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) models with associated learning [algorithms](https://en.wikipedia.org/wiki/Algorithm) that analyze data for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis). SVMs are one of the most robust prediction methods, being based on statistical learning frameworks. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-[probabilistic](https://en.wikipedia.org/wiki/Probabilistic_classification) [binary](https://en.wikipedia.org/wiki/Binary_classifier) [linear classifier](https://en.wikipedia.org/wiki/Linear_classifier). SVM maps training examples to points in space so as to maximize the width of the gap between the two categories.

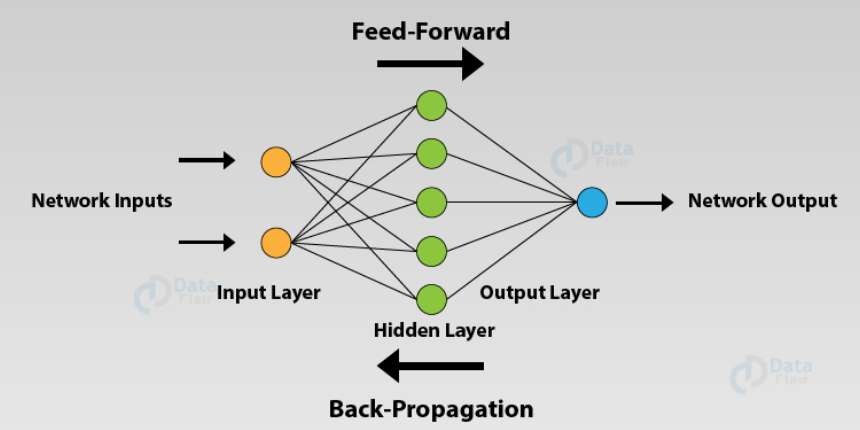




* 1. **ANN Model:**

An ANN is based on a collection of connected units or nodes called [artificial neurons](https://en.wikipedia.org/wiki/Artificial_neuron), which loosely model the [neurons](https://en.wikipedia.org/wiki/Neuron) in a biological brain. Each connection, like the [synapses](https://en.wikipedia.org/wiki/Synapse) in a biological brain, can transmit a signal to other neurons. An artificial neuron receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a [real number](https://en.wikipedia.org/wiki/Real_number), and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a [weight](https://en.wikipedia.org/wiki/Weight_(mathematics)) that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.





**5. Hyper Parameter Tuning:**

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

We used Grid Search CV for hyperparameter tuning. This also results in cross validation and in our case, we divided the dataset into different folds.

**5.1 Grid Search CV:**

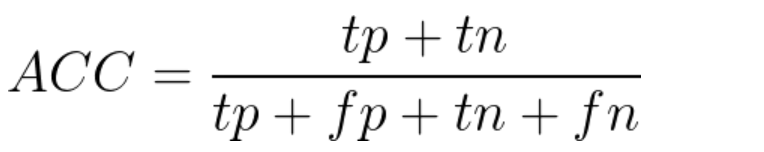
Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

**6. Model performance:**

The performance of the model can be evaluated using the following approaches

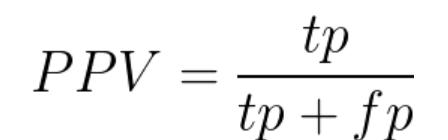
**6.1. Accuracy Score:**

It measures how many observations, both positive and negative, were correctly classified.



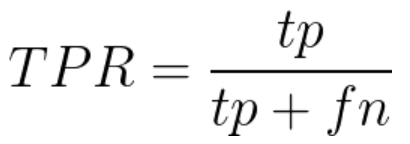
**6.2. Precision:**

It measures how many observations predicted as positive are in fact positive. Taking our fraud detection example, it tells us what the ratio of transactions is correctly classified as fraudulent.



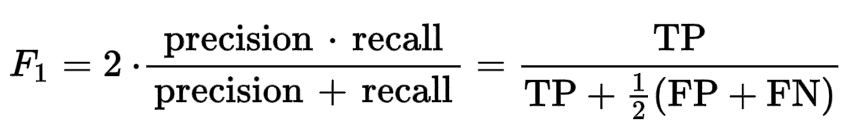
**6.3 Recall:**

It measures how many observations out of all positive observations have we classified as positive. It tells us how many fraudulent transactions we recalled from all fraudulent transactions.



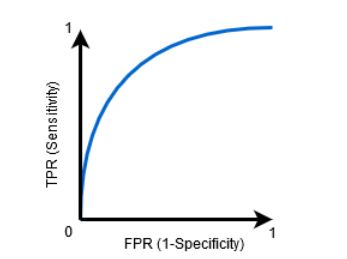
**6.4. F1 Score:**

It’s the harmonic mean between precision and recall.



**6.5. ROC-AUC Score/Curve:**

To get one number that tells us how good our curve is, we can calculate the Area Under the ROC Curve, or ROC AUC score. The more top-left your curve is the higher the area and hence higher ROC AUC score.



**7. Conclusion:**

That's it! We reached the end of our exercise. Starting with loading the data so far, we have done EDA, Class imbalance treatment, removing multicollinearity columns, feature selection and then model building. In all these models, our recall score revolves in the range of 27 to 89%.

Logistic regression model is a good model in our case with Recall score of 89%

**References-**

1. Unfold Data Science
2. Wikipedia
3. Analytics Vidhya