# **Bankruptcy Prediction Using Machine Learning**

**Abstract**

Our project is geared towards employing machine learning methodologies to predict bankruptcy. By leveraging data analysis and predictive modeling, we aim to forecast the probability of a company encountering financial insolvency.

**Dataset Used**

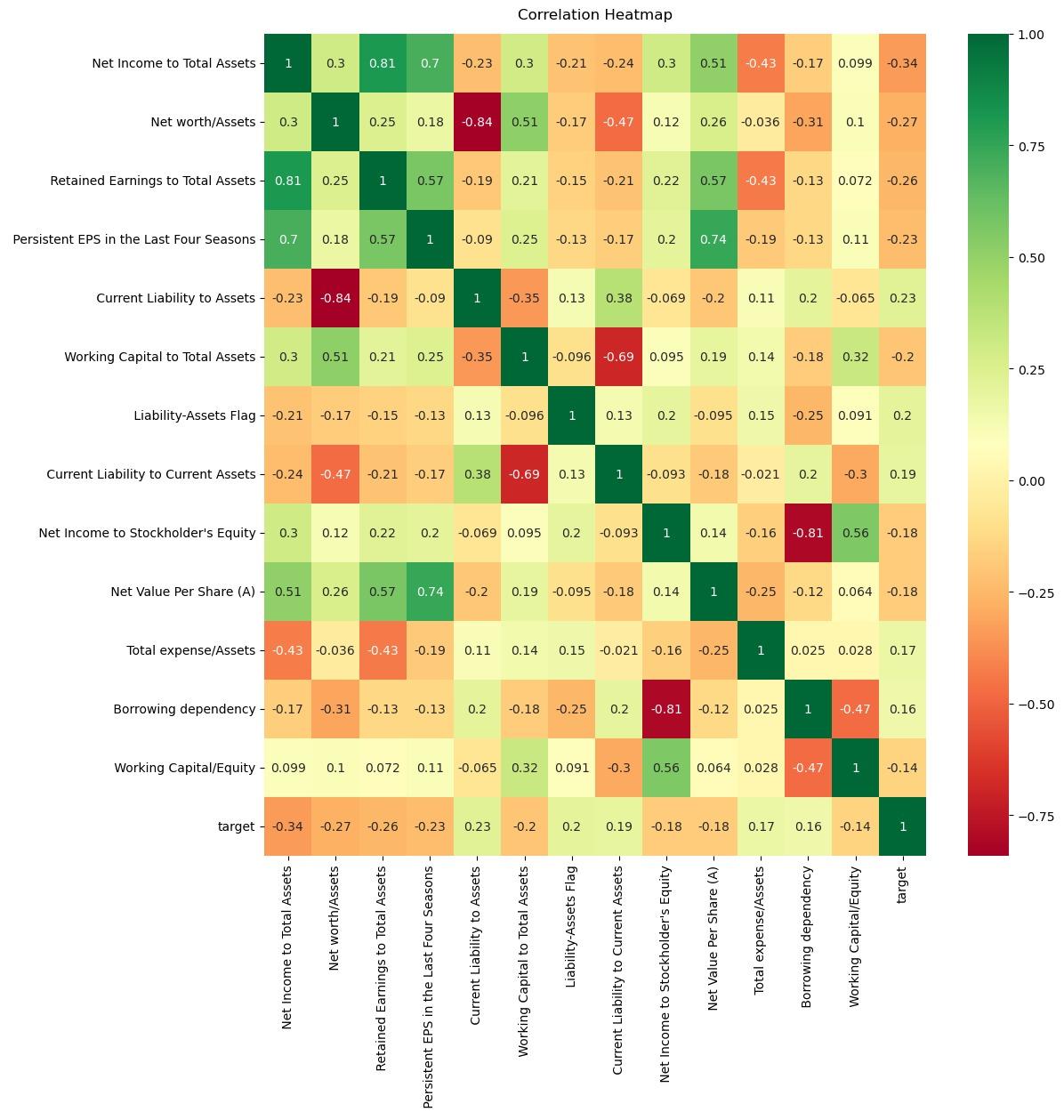
The data were collected from the Taiwan Economic Journal for the years 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange.

The columns contain 95 financial ratios of the company out of which best features are to be used for the optimum prediction.

Link: <https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction>

**Methodology**

* **Importing Libraries and Data:**
  + Essential libraries such as pandas, numpy, and scikit-learn are imported for data manipulation, analysis, and machine learning.
  + The dataset is loaded from a CSV file into a DataFrame (df), containing features and the target variable indicating bankruptcy status.
* **Checking Data Quality:**
  + Data quality is assessed by checking for missing values and duplicated rows using pandas' isna() and duplicated() functions, ensuring reliability for model training.
* **Feature Engineering and Selection:**
  + The target column is renamed to 'target' for clarity.
  + Relevant features for predicting bankruptcy are identified through feature selection, enhancing model efficiency and interpretability.
  + The SelectKBest method with the ANOVA F-value metric selects the top 25 best features based on their relationship with the target variable.
* **Feature Reduction via Correlation Analysis:**
  + A correlation heatmap is plotted for the top 25 selected features to identify those with a correlation above 0.85 as highly correlated.
  + Redundant features are removed based on lower 'feature scores', retaining essential predictors.
  + This leads to a final set of 13 features for training the bankruptcy prediction model.



* **Handling Class Imbalance:**
  + Class imbalance is addressed using oversampling (SMOTE) technique to balance the dataset, preventing biased training.
* **Model Building and Evaluation:**
  + Classification models including Decision Tree, Random Forest, Logistic Regression, Naive Bayes, SVM, and XGBoost are implemented.
  + Models are evaluated on balanced and unbalanced datasets, with hyperparameter tuning performed via Bayesian optimization (BayesSearchCV).
  + Performance metrics (accuracy, precision, recall, F1 score, AUC-ROC) and confusion matrices assess predictive capabilities.
* **Ensemble Modeling:**
  + A Voting Classifier ensemble model combines multiple base models using soft voting, leveraging their collective strengths for improved accuracy.
* **Output and Interpretation:**
  + Model performances are summarized in a DataFrame and saved as 'output.csv', with ensemble model metrics providing a comparative performance overview.

**FINANCIAL RATIOS USED**

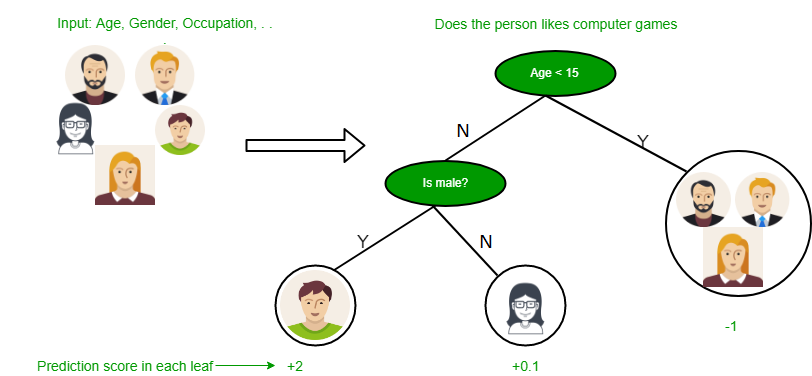
* **Net income to total assets**  - Net Income = Pre-Tax Income (EBT) – Taxes.
* **Net worth/assets** - Net worth is calculated by subtracting all liabilities from assets.
* **Retained Earnings/Total Assets (RE/TA)**  - This ratio measures the amount of reinvested earnings or losses, which reflects the extent of the company's leverage.
* **EPS** - Earnings per share (EPS) is calculated as a company's profit divided by the outstanding shares of its common stock. The resulting number serves as an indicator of a company's profitability.
* **Current Liabilities to Assets Ratio** - measures the proportion of a company's assets that are financed by its current liabilities, indicating short-term financial stability and the company's ability to meet its short-term obligations with its current assets.
* **The Working Capital to Total Assets ratio** - measures a company's ability to cover its short term financial obligations (Total Current Liabilities) by comparing its Total Current Assets to its Total Assets.
* **Current Ratio** - A firm's total current assets are divided by its total current liabilities. It shows the ability of a firm to meets its current liabilities with current assets.
* **Net income to stockholder’s Equity** - Return on equity (ROE) is the measure of a company's net income divided by its shareholders' equity.
* **The Net Asset Value Per Share (NAVPS**) - is a real estate metric that indicates the value of a mutual fund or an exchange-traded fund (ETF). The NAVPS is obtained by dividing the net asset value (total assets less liabilities) of a fund by the number of outstanding shares.
* **Liabilities - assets flags -** the term "liabilities - assets" refers to the calculation where you subtract the total assets of a company from its total liabilities. This calculation can result in what is commonly referred to as "net liabilities."
* **Total expense/asset** - total expense ratio is calculated by dividing total fund costs by total fund assets. The total fund's costs are all costs associated with managing the operations of the investment. The total fund's assets are the total investments of the company in stocks, bonds, and securities.
* **Borrowing dependency** - Borrowing dependency in companies refers to the extent to which a company relies on borrowed funds, such as loans or bonds, to finance its operations or growth, indicating its leverage and potential financial risk exposure.
* **Working cap/equity -** The Working Capital to Equity ratio measures the proportion of shareholders' equity that is tied up in the day-to-day funds necessary to keep a company operating, providing insight into the company's operational efficiency and financial leverage.

**Models Employed**

**Decision Trees**

A decision tree is a type of supervised learning algorithm that is commonly used in machine learning to model and predict outcomes based on input data. It is a tree-like structure where each internal node tests on attribute, each branch corresponds to attribute value and each leaf node represents the final decision or prediction.

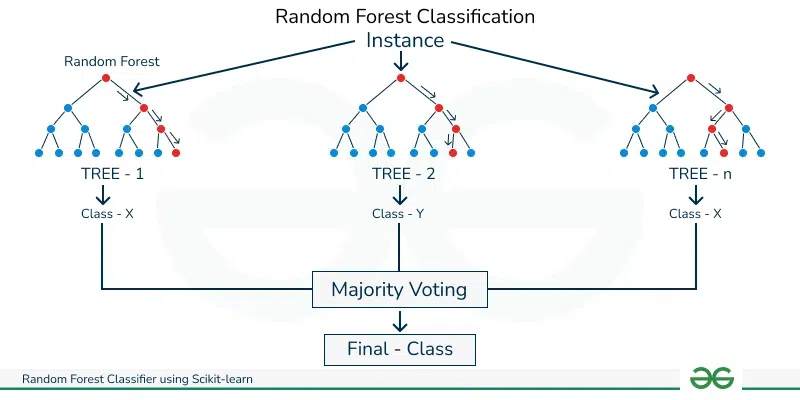
The process of forming a decision tree involves recursively partitioning the data based on the values of different attributes. The algorithm selects the best attribute to split the data at each internal node, based on certain criteria such as information gain or Gini impurity. This splitting process continues until a stopping criterion is met, such as reaching a maximum depth or having a minimum number of instances in a leaf node.



**Random Forest**

The Random forest or Random Decision Forest is a supervised Machine learning algorithm used for classification, regression, and other tasks using decision trees. Random Forests are particularly well-suited for handling large and complex datasets, dealing with high-dimensional feature spaces, and providing insights into feature importance.

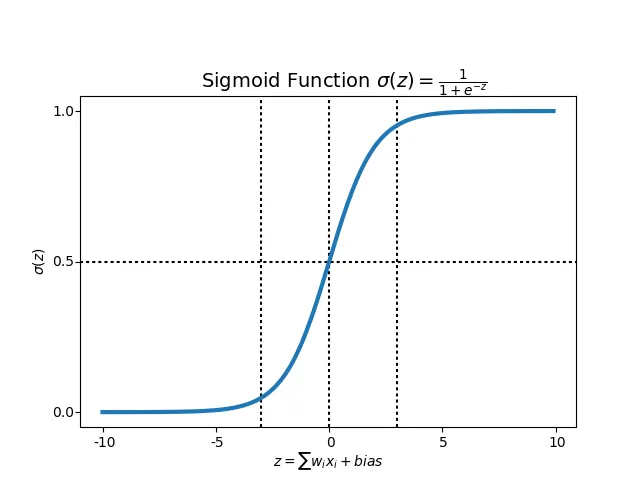
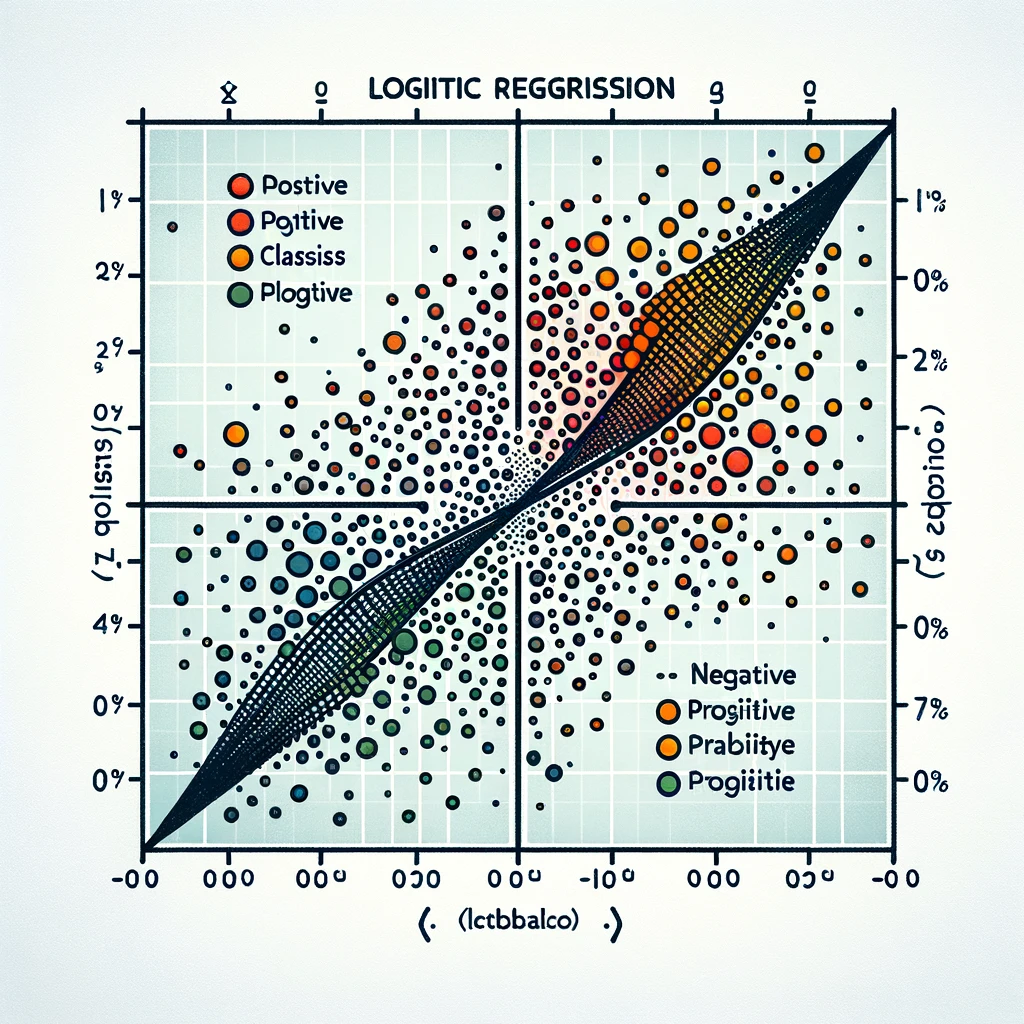
It is a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction. The process is known as feature bagging, helps prevent the dominance of any single feature and promotes a more robust model.



**Logistic Regression**

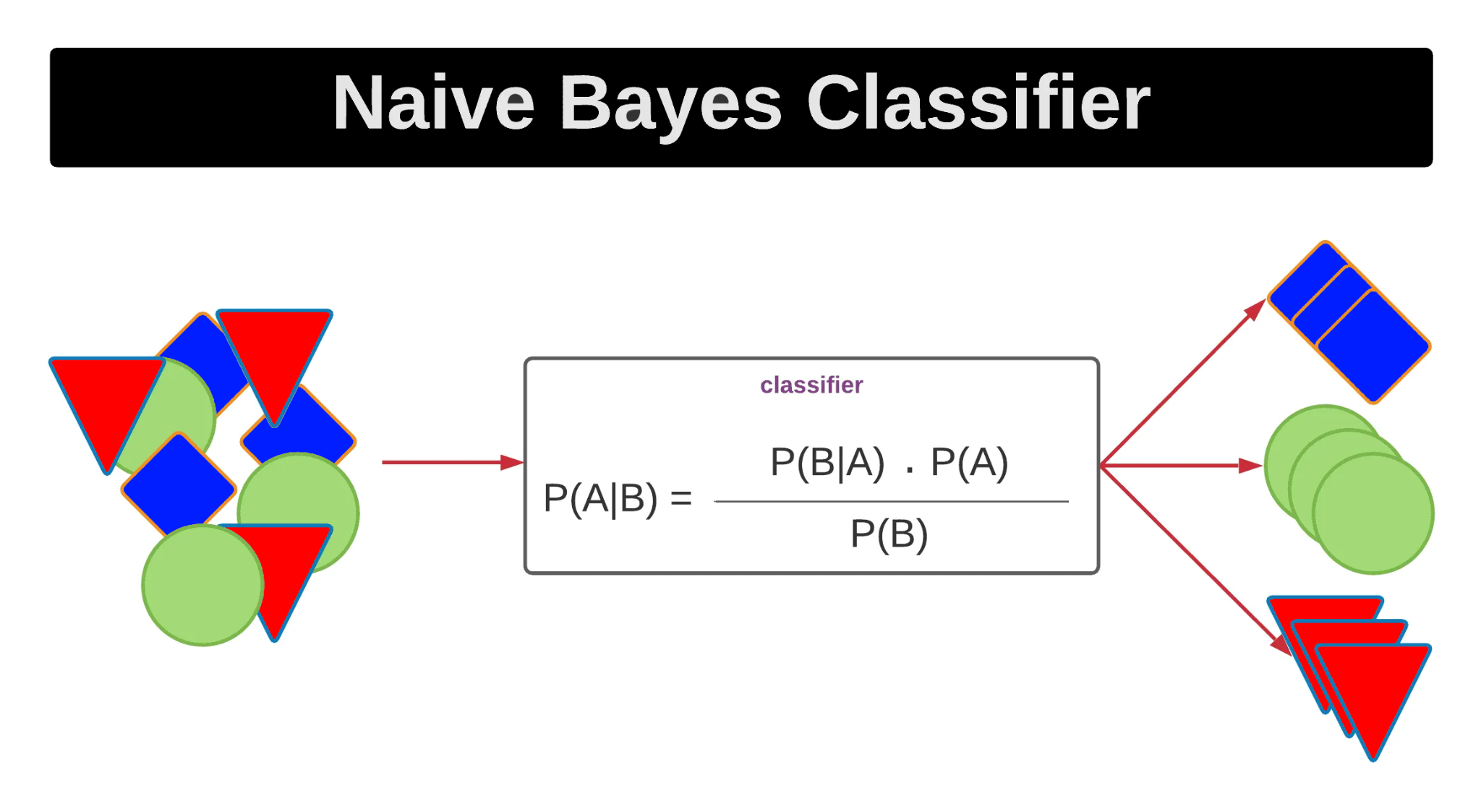
Logistic regression is a supervised machine learning algorithm used for binary classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyzes the relationship between two data factors.

In this we use sigmoid function, that takes input as independent variables and produces a probability value between 0 and 1.Here, instead of fitting a regression line, we fit an “S” shaped logistic function

.****

**Naive Bayes**

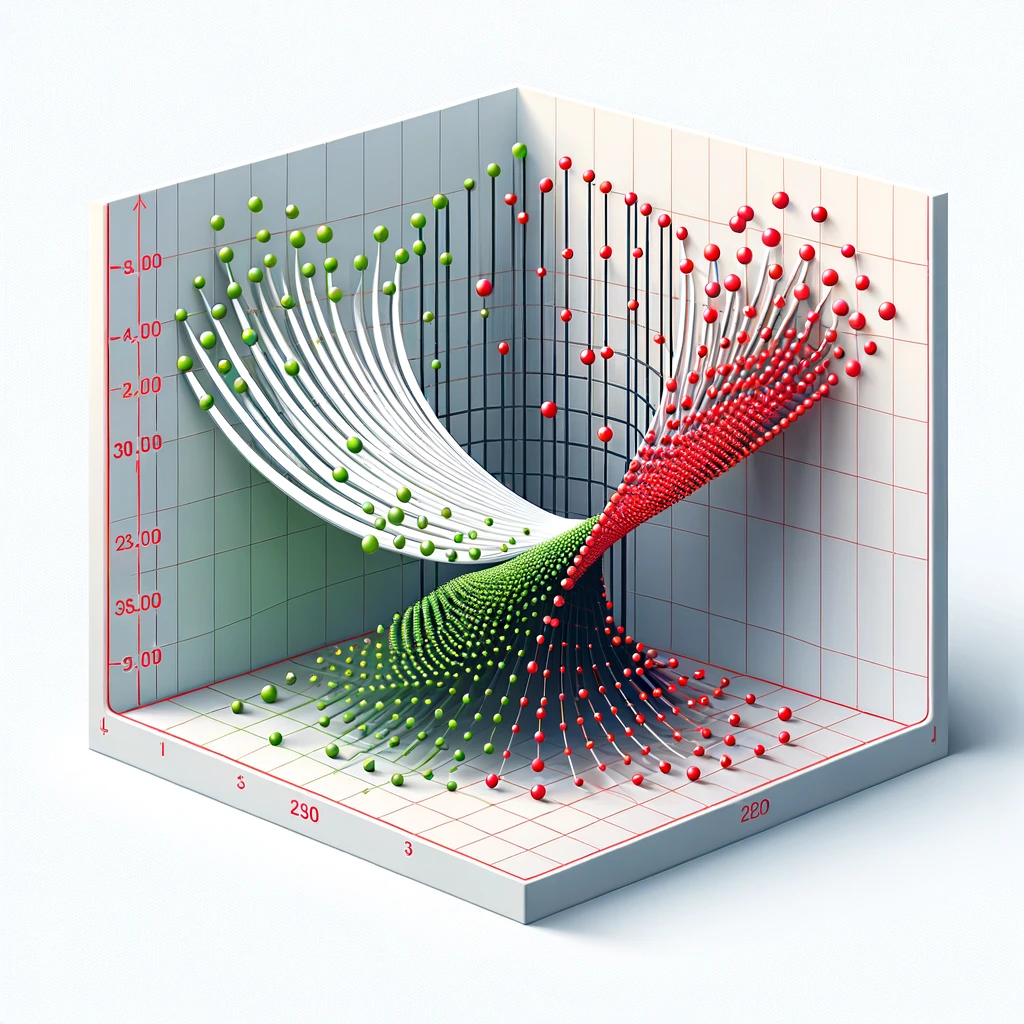
Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem with an assumption of independence between the features



**Support Vector Machine**

A support vector machine (SVM) is a machine learning algorithm that uses supervised learning models to solve complex classification problems by performing optimal data transformations.

SVM uses kernel methods to transform data features by employing kernel functions. Kernel functions rely on the process of mapping complex datasets to higher dimensions in a manner that makes data point separation easier.

****

**Key Components:**

Hyperplane: A decision boundary separating classes, a line in 2D or a plane/hyperplane in higher dimensions.

Support Vectors: Support vectors are the data points that are closest to the hyperplane and influence its position and orientation.

Margin: The margin is the distance between the hyperplane and the nearest data point from either class.

**XG Boost**

**XGBoost** is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction.

**Initialization:** XGBoost begins with an initial prediction, typically the mean for regression or log odds for classification. It proceeds with multiple boosting rounds, refining predictions iteratively.

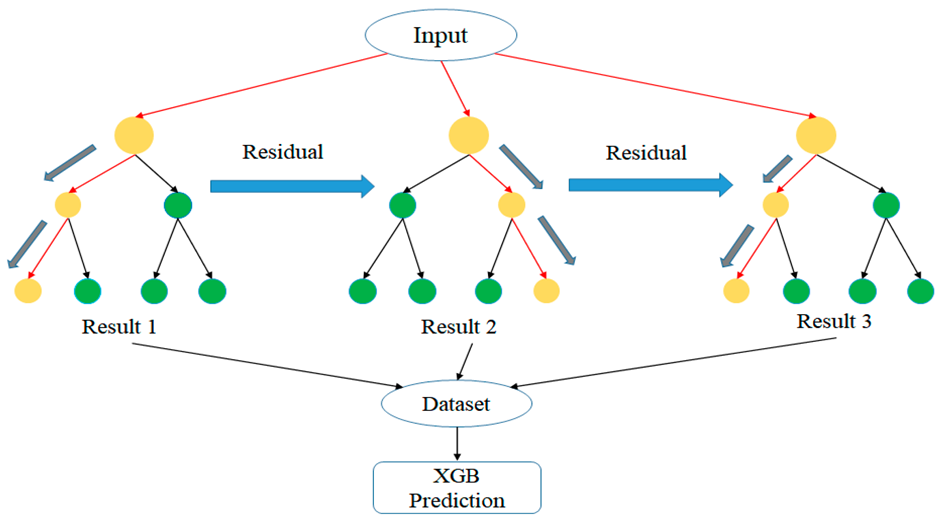
**Gradient and Hessian Computation:** XGBoost calculates gradients and hessians of a predefined loss function. These gradients and hessians guide the training of regression trees in subsequent steps.

**Tree Training:** Regression trees are trained to minimize the negative gradient divided by the hessian. This ensures that each tree corrects errors made by the existing ensemble.

**Model Update:** After training each tree, XGBoost updates the model by adding the scaled prediction of the new tree to the ensemble.

**Regularization:** XGBoost applies regularization techniques like tree complexity constraints and column subsampling to prevent overfitting.

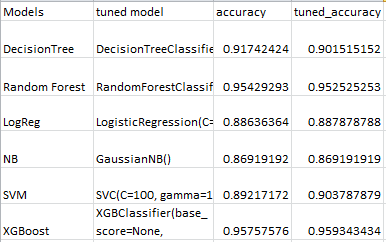
**Prediction Aggregation:** Predictions from all trees are aggregated to produce the final outcome, typically by weighted averaging.



**PERFORMANCE METRICS**

* **Accuracy**: Accuracy is the percentage of predictions that the model makes correctly. It is calculated by dividing the number of correct predictions by the total number of predictions.
* **Precision**: Precision is the percentage of positive predictions that the model makes that are actually correct. It is calculated by dividing the number of true positives by the total number of positive predictions.
* **Recall**: Recall is the percentage of all positive examples that the model correctly identifies. It is calculated by dividing the number of true positives by the total number of positive examples.
* **F1 score**: The F1 score is a weighted average of precision and recall. It is calculated by taking the harmonic mean of precision and recall.
* **Confusion matrix**: A confusion matrix is a table that shows the number of predictions for each class, along with the actual class labels. It can be used to visualize the performance of the model and identify areas where the model is struggling.

**RESULTS**

****

**(complete results in notebook)**

**ENSEMBLE MODEL RESULTS**

In our project, we employed an ensemble approach to classification by combining multiple machine-learning algorithms using a Voting Classifier. This technique aggregates the predictions of individual classifiers and yields improved performance compared to any single classifier alone.

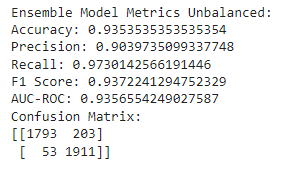
### **Ensemble Configuration**

We assembled the following classification algorithms into our Voting Classifier:

* Random Forest Classifier
* Support Vector Machines (SVM)
* XG Boost
* Decision Trees
* Naive Bayes
* Logistic Regression

### **Results Summary**

Upon training and evaluating the ensemble classifier on our dataset, we obtained the following performance metrics:



**Conclusion**

The ensemble classifier demonstrated robust performance across multiple evaluation metrics, indicating its effectiveness in accurately classifying instances in our dataset. By leveraging the diverse strengths of individual classifiers, the ensemble approach achieved superior predictive accuracy and generalization capability.