COMPARISON OF VARIOUS MACHINE LEARNING ALGORITHMS ACROSS DIFFERENT LIBRARIES BASED ON ACCURACY

**Dataset 1: Credit Score Classification Prediction**

Our project involves evaluating a dataset for credit score categorization using machine learning. Credit scoring is important in finance because it helps lenders measure the risk of granting credit to individuals. Our dataset covers a variety of variables such as biographical information, credit history, and financial activity. We intend to use machine learning algorithms such as Decision Tree Classifiers and Random Forest Classifiers to create models using two different libraries: Scikit-learn and XGBoost that accurately classify individuals into various credit risk categories, such as Poor, Standard, and Good. Our goal is to create a reliable credit scoring model that improves risk assessment accuracy and offers information about creditworthiness factors. Such a model can help financial organizations make more informed lending decisions, lower risks, and streamline loan approval processes. Our analysis aims to help advance credit scoring algorithms and promote responsible lending practices in the financial industry, ultimately benefiting both lenders and borrowers.

**Dataset Explanation:**

**ID:** Represents a unique identification of an entry.

**Customer\_ID:** Represents a unique identification of a person.

**Month:** Represents the month of the year.

**Name:** Represents the name of a person.

**Age:** Represents the age of the person.

**SSN:** Represents the social security number of a person.

**Occupation:** Represents the occupation of the person.

**Annual\_Income:** Represents the annual income of the person.

**Monthly\_Inhand\_Salary:** Represents the monthly base salary of a person.

**Num\_Bank\_Accounts:** Represents the number of bank accounts a person holds.

**Num\_Credit\_Card:** Represents the number of other credit cards held by a person.

**Interest\_Rate:** Represents the interest rate of a credit card.

**Num\_of\_Loan:** Represents the number of loans taken from the bank.

**Type\_of\_Loan:** Represents the types of loans taken by a person.

**Delay\_from\_due\_date:** Represents the average number of days delayed from the payment date.

**Num\_of\_Delayed\_Payment:** Represents the average number of payments delayed by a person.

**Changed\_Credit\_Limit:** Represents the percentage change in credit card limit.

**Num\_Credit\_Inquiries:** Represents the number of credit card inquiries.

**Credit\_Mix:** Represents the classification of the mix of credits.

**Outstanding\_Debt:** Represents the remaining debt to be paid (in USD).

**Credit\_Utilization\_Ratio:** Represents the utilization ratio of credit cards.

**Credit\_History\_Age:** Represents the age of credit history of the person.

**Payment\_of\_Min\_Amount:** Represents whether only the minimum amount was paid by the person.

**Total\_EMI\_per\_month:** Represents the monthly EMI payments (in USD).

**Amount\_invested\_monthly:** Represents the monthly amount invested by the customer (in USD).

**Payment\_Behaviour:** Represents the payment behavior of the customer (in USD).

**Monthly\_Balance:** Represents the monthly balance amount of the customer (in USD).

**Credit\_Score:** Represents the bracket of credit score (Poor, Standard, Good).

**Characteristics:**

|  |  |
| --- | --- |
| **Characteristics** | **Values** |
| Number of independent and dependent variables | Independent columns: 27  Dependent column: 1 |
| Number of instances and features | Instances: 1,00,000  Features: 28 |
| Data Types | Numerical, Categorical, and Textual |
| Summary of variables |  |
| Data Cleaning | The number of irrelevant variables: 6 ('ID', 'Customer\_ID', 'Month', 'Name', 'SSN', 'Type\_of\_Loan')  Number of duplicates removed: No duplicates in the dataset.  Dimensionality reduction technique used: Dimensionality reduction technique not used.  Number of Missing Values: Missing Values in 8 variables. Missing values are fixed using Mean and Mode.  The number of outliers Removed and technique: Outliers removed from 4 variables ('Age', 'Monthly\_Inhand\_Salary', 'Num\_Credit\_Card', 'Amount\_invested\_monthly'). |
| Data Normalization | Technique used: MinMaxScaler |
| Data balancing characteristics and splitting | Number of records in each class: Standard – 53174, Poor – 28998, Good – 17828  Number of records for training: 58458  Number of records for testing: 25054 |

**Dataset 2: Bank Marketing**

This machine learning research uses the Bank Marketing dataset to boost customer engagement and optimize marketing efforts. We intend to use decision trees and random forest models to identify customers who are likely to subscribe to term deposits, allowing for focused marketing campaigns. The major corporate goal is to reduce marketing expenditures by properly forecasting client behavior and targeting efforts to those who are most likely to respond positively. The project underlines the significance of proactive bank marketing and customer relationship management in establishing a strong brand image and meeting customer expectations. We try to improve client satisfaction, raise returns, and assure the success of banking businesses in today's competitive environment by utilizing new methodologies and data-driven insights.

**Dataset Explanation:**

**age:** Represents the age of the individual.

**job:** Describes the occupation or job of the person.

**marital:** Indicates the marital status of the person (e.g., married, single, divorced).

**education:** Represents the educational level of the person (e.g., primary, secondary, tertiary).

**default:** Indicates whether the person has credit in default ('yes', 'no', or 'unknown').

**housing:** Shows whether the person has a housing loan ('yes', 'no', or 'unknown').

**loan:** Indicates whether the person has a personal loan ('yes', 'no', or 'unknown').

**contact:** Describes the method of communication used to contact the person (e.g., 'cellular', 'telephone').

**day:** Indicates the day of the week of the last contact.

**month:** Represents the month of the last contact.

**Duration:** Represents the duration of the last contact in seconds.

**campaign:** Indicates the number of contacts made during this campaign.

**pdays:** Describes the number of days since the person was last contacted or -1 if they were not previously contacted.

**previous:** Represents the number of contacts made before this campaign.

**poutcome:** Indicates the outcome of the previous marketing campaign.

**deposit:** The target variable, indicating whether the person subscribed to a term deposit ('yes' or 'no').

**Characteristics:**

|  |  |
| --- | --- |
| **Characteristics** | **Values** |
| Number of independent and dependent variables | Independent columns: 16  Dependent column: 1 |
| Number of instances and features | Instances: 11,162  Features: 17 |
| Data Types | Numerical, and Categorical |
| Summary of variables |  |
| Data Cleaning | The number of irrelevant variables: 0  Number of duplicates removed: No duplicates in the dataset.  Dimensionality reduction technique used: Not used any dimensionality reduction techniques.  Number of Missing Values: No Missing Values.  The number of outliers Removed and technique: Outliers not removed. |
| Data Normalization | Technique used: MinMaxScaler |
| Data balancing characteristics and splitting | Number of records in each class: 0 – 5873 and 1 – 5289  Number of records for training: 7813  Number of records for testing: 3349 |

**Dataset 3: Bank customer churn prediction**

This study examines the "Bank Customer Churn" dataset to learn why consumers leave a bank's services. Customer churn, or when customers end their association with a company, is harmful to the company's success. We want to unearth insights that will help the bank improve its customer retention strategy by analyzing them using decision trees and random forest techniques. The dataset comprises customer demographics, credit scores, account balances, tenure, and other data, with the target variable "Exited" indicating if a client churned (1) or not (0). We hope to identify the variables causing customer turnover by studying this data, doing feature engineering, and utilizing predictive modeling approaches. Finally, this study will provide the bank with significant insights into how to proactively handle churn and promote long-term client relationships, thereby increasing profitability and market competitiveness.

**Dataset Explanation:**

**RowNumber:** corresponds to the record (row) number and has no effect on the output.

**CustomerId:** contains random values and has no effect on customers leaving the bank.

**Surname:** the surname of a customer has no impact on their decision to leave the bank.

**CreditScore:** can have an effect on customer churn, since a customer with a higher credit score is less likely to leave the bank.

**Geography:** a customer’s location can affect their decision to leave the bank.

**Gender:** it’s interesting to explore whether gender plays a role in a customer leaving the bank.

**Age:** this is certainly relevant since older customers are less likely to leave their bank than younger ones.

**Tenure:** refers to the number of years that the customer has been a client of the bank. Normally, older clients are more loyal and less likely to leave a bank.

**Balance:** also a very good indicator of customer churn, as people with a higher balance in their accounts are less likely to leave the bank compared to those with lower balances.

**NumOfProducts:** refers to the number of products that a customer has purchased through the bank.

**HasCrCard:** denotes whether a customer has a credit card or not. This column is also relevant since people with credit cards are less likely to leave the bank.

**IsActiveMember:** active customers are less likely to leave the bank.

**EstimatedSalary:** as with balance, people with lower salaries are more likely to leave the bank compared to those with higher salaries.

**Exited:** whether the customer left the bank or not (Target column).

**Complain:** whether the customer has a complaint or not.

**Satisfaction Score:** Score provided by the customer for their complaint resolution.

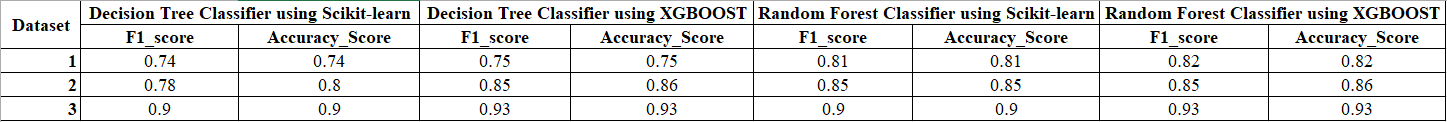
**Card Type:** type of card held by the customer.

**Points Earned:** the points earned by the customer for using a credit card.

**Characteristics:**

|  |  |
| --- | --- |
| **Characteristics** | **Values** |
| Number of independent and dependent variables | Independent columns: 17  Dependent column: 1 |
| Number of instances and features | Instances: 10,000  Features: 18 |
| Data Types | Numerical, Categorical, and Textual |
| Summary of variables |  |
| Data Cleaning | The number of irrelevant variables: 3 ('RowNumber','CustomerId','Surname').  Number of duplicates removed: No duplicates in the dataset.  Dimensionality reduction technique used: PCA (Principal Component Analysis).  Number of Missing Values: No Missing Values.  The number of outliers Removed and technique: Outliers not removed. |
| Data Normalization | Technique used: MinMaxScaler |
| Data balancing characteristics and splitting | Number of records in each class: 0 – 7962 and 1 – 7962 (After balancing the data using Over sampling technique).  Number of records for training: 11146  Number of records for testing: 4778 |

**Q1: Do the two implementations of identically named techniques perform differently or the same?**

****

By evaluating the table, the Decision Tree Classifier built with Scikit-learn and XGBoost, as well as the Random Forest Classifier, produces varying accuracies. This disparity highlights the fact that different approaches have distinct performance characteristics.

**Q2: If they are performing differently, then what could be the reason?**

By using different libraries like Scikit-learn and XGBoost offers implementation of Decision Tree and Random Forest, their internal working is different from each other. For example, variation mostly occurs in changing parameters. Some changing parameters of the model will show the variation in accuracy.

Comparison of decision tree classifier with two different libraries

A screenshot of a computer code

Description automatically generated

Fig. 1. Decision Tree Classifier model building using Scikit-learn

A screenshot of a computer screen

Description automatically generated

Fig. 2. Decision Tree Classifier model building using XGBoost

Figure 1 and 2 illustrates the implementation of different libraries in the Decision Tree Classifier.

In the Scikit-learn library, we are using parameters such as max\_depth and random state. The max\_depth means, the maximum depth of the tree and random state used to control the randomness of the tree.

In the XGBoost library, we are using parameters such as criterion, max\_depth and n\_estimators. By using criterion as entropy, it helps to measure the quality of a split. Max\_depth means the maximum depth of the tree and n\_estimators sets the number of trees (estimators) in the ensemble.

A screenshot of a computer program

Description automatically generated

Fig. 3. Random Forest Classifier model building using Scikit-learn

A screenshot of a computer code

Description automatically generated

Fig. 4. Random Forest Classifier model building using Scikit-learn

Figure 3 and 4 illustrates the implementation of different libraries in the Random Forest Classifier.

In the Scikit-learn library, we are using parameters such as max\_depth and random state. The max\_depth means, the maximum depth of the tree and random state used to control the randomness of the tree.

In the XGBoost library, we are using parameters such as n\_estimators, max\_depth and random state. n\_estimators sets the number of trees (estimators) in the ensemble. Max\_depth means the maximum depth of the tree and the random state used to control the randomness of the tree.

We can conclude that the default hyperparameters, as well as hyperparameters specified during model training, can have a considerable impact on model performance. Different libraries may use different default settings for hyperparameters or provide distinct sets of hyperparameters to modify resulting in differences in model behavior.