## **Artificial Intelligence**

## **and Machine Learning**

Project Report

Semester-IV (Batch-2022)

Loan Prediction

A red and white sign

Description automatically generated with low confidence

**Submitted To:**  **Submitted By:**

Ms. Shagun Sharma Khushi Jamloki 2210990513 (G7)

Keshav Goel 2210990504 (G7)

Diya Raina 2210990304 (G7)

**Department of Computer Science and Engineering**

## **Chitkara University Institute of Engineering & Technology,**

## **Chitkara University, Punjab**

**TABLE OF CONTENTS**

**Abstract**.........................................................................................................................**3**

**1.Introduction**................................................................................................................**4-9**

1.1 Background............................................................................................

1.2 Objective................................................................................................

1.3 Significance...........................................................................................

1.4 Target Users..........................................................................................

**2. Problem Definition and Requirement**......................................................................**10**

2.1 Problem Statement................................................................................

2.2 Requirement..........................................................................................

**3. Materials and Methodologies**...................................................................................**11-13**

3.1 About Dataset.........................................................................................

3.2 Handling Missing values and Relabelling Categorical values...............

3.3 Feature Engineering...............................................................................

**4. Data Analysis and Methodologies**...........................................................................**14-19**

4.1 EDA.......................................................................................................

4.2 Data Distribution...................................................................................

**5. Algorithm and Methodologies followed**.................................................................**20-36**

5.1 Model Training.....................................................................................

5.2 Algorithm Used....................................................................................

**6. References**..................................................................................................................**37**

**7. Conclusion**..................................................................................................................**38**

**ABSTRACT**

The prediction of loan approval is a crucial task for financial institutions and has been a longstanding challenge in the industry. Historically, banks and other lenders relied on manual processes and subjective criteria to evaluate loan applications, which often led to inconsistent decisions and increased risk of loan defaults. With the rise of machine learning techniques, there is now an opportunity to develop more accurate and reliable predictive models that can help financial institutions make better lending decisions. This study proposes a comparative analysis of various machine learning algorithms for predicting loan approval. The explored algorithms include Random Forest Classifier, K-Nearest Neighbors Classifier, Support Vector Classifier, and Logistic Regression. The dataset is prepared by performing exploratory data analysis and feature engineering. Statistics like accuracy score and F1 score are used to judge the execution of each approach. The findings show that the Light GBM had the highest accuracy of 99.2%, followed by Random Forest (98.7%), XGB (98.3%), Decision Tree(97.7%),Support Vector Machine (94.3%) , K Nearest Neighbour (92.2%) and Logistic Regression(91.1%) These results highlight the potential of machine learning algorithms to improve the loan approval process and reduce the risk of loan defaults. Overall, this study provides insights into the effectiveness of different machine learning algorithms for loan approval prediction and can be useful for financial institutions in improving their decision-making process. The proposed approach can also be extended to other domains where classification is a critical task

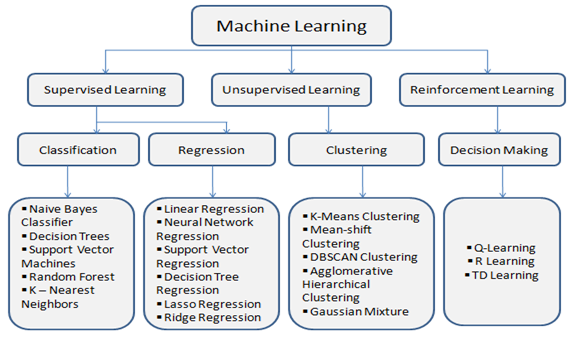
**1.Introduction**

Small loan is an important aspect of our everyday life: it allows aspiring entrepreneurs to get started on ideas that could be grown into business; it allows curious students to afford higher education that is otherwise unavailable without a stable income; more importantly, it allows ordinary people who have no friends or relatives for support to obtaining short-term financial assistance and get back on their feet. Nevertheless, with loan it comes with the possibility of default as well. Default is a financial term describing the failure of meeting the legal obligation of a loan - paying back the principal and interest. It’s a common problem in the financial industries and one of the major risks of offering loans. As of 2023, more than 20 million people (about the population of New York) in the US have active loans owing a collective debt of 178 billion dollars (about $550 per person in the US). Despite that, more than 20% of all applicants were denied loans. The loan approval or rejection has enormous ramifications for both the applicant and the bank, causing possible opportunity costs for both parties. Banks like Wells Fargo and Morgan Stanley have looked at the use of AI in determining lending risk and developing a loan prediction system in recent years to overcome human bias and delays in the application processing time. Traditional methods often rely heavily on manual assessment of factors like income, credit history, and other dynamic parameters to determine lending risk. However, machine learning approaches offer the potential to streamline this process and improve accuracy without requiring extensive manual intervention or domain expertise. For example, traditional credit scoring models typically use statistical techniques to analyze historical data and create risk models based on factors such as payment history, credit utilization, and length of credit history. While effective, these models can be limited in their ability to capture complex patterns and adapt to changing dynamics.



**1.1 Background:**

Machines have been an integral part of life in today's world, especially computers. The Association of Computing Machinery [ACM] definition of computing is any goal-oriented activity, which requires benefits from computers or creating computers. This definition is the accepted direction of machines and their computations. Machine Learning [ML] is a technique to analyze data that automates the building of the Analytical models. ML is born from the pattern recognition and with the idea that computers can learn from data, can identify patterns and make informed decisions by the theory stating, computers can learn anything without being explicitly being programmed to perform specific tasks. There are mainly two approaches in machine learning such as Supervised Learning [SL] and Unsupervised Learning [UL], but recently Semi-Supervised Learning [SSL] and Reinforcement learning have emerged. Machine learning algorithms can automatically learn patterns and relationships from large volumes of data, enabling more accurate and flexible credit scoring models. For instance, algorithms like Random Forest, Support Vector Machines (SVM), or Gradient Boosting Machines (GBM) can analyze various features derived from transactional data to predict creditworthiness. These machine learning-based approaches offer several advantages over traditional methods. They can handle a broader range of input features, including non-linear relationships and interactions between variables. Moreover, they can adapt more effectively to changes in the economic environment or customer behavior, enhancing the robustness of credit assessment models. By reducing the reliance on manual intervention and domain expertise, machine learning algorithms also have the potential to improve fairness and consistency in lending decisions. However, it's crucial to ensure that these models are carefully trained and validated to mitigate any potential biases or unintended consequences. Overall, integrating machine learning algorithms into the loan approval process represents a significant step forward in enhancing efficiency, accuracy, and fairness while reducing the manual effort required for risk assessment.



**1.2 Objective:**

**1.2.1 Data Acquisition and Preprocessing**: Gather and preprocess a comprehensive dataset for loan approval prediction, addressing challenges such as data quality, missing values, and potential biases. Employ robust preprocessing techniques to handle data inconsistencies and ensure data integrity.

**1.2.2 Accurate Loan Approval Prediction:** Develop a highly accurate machine learning model for predicting loan approval decisions based on various applicant features. The model should exhibit high precision and recall in identifying applicants likely to be approved or denied.

**1.2.3 Early Identification and Decision Support:** Enable early identification of loan approval risks, facilitating timely decision-making and support. Early detection of potential risks helps prevent defaults and minimizes financial losses for lending institutions.

**1.2.4 Personalized Risk Assessment**: Tailor the model to account for individual applicant characteristics, financial histories, and loan preferences. Provide personalized risk assessments to applicants, helping them understand their likelihood of loan approval and potential terms.

**1.2.5 User-friendly Interface:** Design an intuitive and user-friendly interface for the loan approval prediction system. The interface should be accessible to both loan officers and applicants, facilitating easy interaction and decision-making.

**1.2.6 Interpretability and Transparency:** Ensure the model's interpretability and transparency by providing explanations for loan approval decisions. This transparency builds trust with stakeholders and enables better understanding of the factors influencing loan decisions.

**1.2.7 Continuous Model Improvement:** Implement mechanisms for continuous learning and model adaptation to evolving lending practices and market dynamics. Regularly update the model with new data and feedback to enhance its accuracy and relevance over time.

**1.2.8 Comprehensive Evaluation and Benchmarking:** Conduct rigorous evaluation and benchmarking of the loan approval prediction model using appropriate performance metrics and validation techniques. Evaluate model performance on historical data, compare with existing methods, and validate across different loan types and applicant demographics.

**1.3 Significance:**

The development of a machine learning-based loan approval prediction system carries substantial significance across various sectors. Here are key points outlining the importance of such a system:

**1.3.1 Enhanced Financial Inclusion:** Access to credit is critical for individuals and businesses to pursue opportunities and achieve financial stability. By providing accurate and timely loan approval predictions, the system can help improve financial inclusion by enabling lenders to assess risks more effectively and extend credit to a wider range of applicants, including those with limited credit history or in underserved communities.

**1.3.2 Risk Mitigation for Lenders:** Loan approval prediction systems can assist lenders in making informed decisions by assessing the creditworthiness of applicants. By accurately identifying potential defaulters or high-risk applicants, the system helps mitigate financial losses for lending institutions, ultimately contributing to the stability and sustainability of the lending industry.

**1.3.3 Efficiency and Streamlining Processes:** Traditional loan approval processes often involve extensive manual review and decision-making, leading to delays and inefficiencies. Implementing a machine learning-based system automates and streamlines the process, reducing the time and resources required for loan assessment and approval. This increased efficiency benefits both lenders and applicants by expediting access to credit.

**1.3.4 Fairness and Avoidance of Bias:** Machine learning models can help minimize human biases inherent in traditional lending practices by basing decisions on objective data and algorithms. By ensuring fairness and impartiality in loan approval decisions, the system promotes equal opportunities for all applicants, regardless of demographic factors or personal characteristics.

**1.3.5 Economic Growth and Development:** Access to credit is essential for driving economic growth and development by enabling investment, entrepreneurship, and consumption. By facilitating more efficient and accurate loan approvals, the system contributes to fostering a conducive environment for business expansion, job creation, and overall economic prosperity.

**1.3.6 Adaptability and Scalability:** Machine learning algorithms can adapt to changing market dynamics and evolving lending practices, ensuring the system remains effective and relevant over time. Additionally, the scalability of the system allows for widespread adoption across different lending institutions and geographic regions, maximizing its impact on the broader financial ecosystem.

**1.3.7 Risk Management and Regulatory Compliance:** Implementing a robust loan approval prediction system helps lending institutions manage risks effectively and comply with regulatory requirements. By leveraging advanced analytics and predictive modeling, the system assists in identifying and addressing potential risks associated with lending activities, ensuring adherence to industry standards and regulations.

**1.4 Target Users:**

The loan approval system is designed to serve various categories of users, each with distinct needs and objectives. Here are the target user groups:

**1.4.1 Lending Institutions and Professionals:** The primary users of the loan approval system are lending institutions, including banks, credit unions, and financial organizations. Loan officers, credit analysts, and risk managers within these institutions are the primary stakeholders who utilize the system to assess loan applications, make informed decisions, and manage lending risks effectively.

**1.4.2. Individual Loan Applicants:** The secondary users of the system are individual loan applicants seeking financial assistance for personal, business, or educational purposes. These users rely on the system to determine their eligibility for loans, understand the factors influencing loan approval decisions, and navigate the application process efficiently.

**1.4.3 Financial Advisors and Consultants:** Financial advisors, consultants, and professionals in the finance industry represent another user group interested in the loan approval system. They may utilize the system to offer guidance and recommendations to clients regarding loan options, eligibility criteria, and financial planning strategies.

**1.4.4. Regulatory Authorities and Compliance Officers:** Regulatory authorities and compliance officers responsible for overseeing lending practices and ensuring adherence to industry regulations may also benefit from the loan approval system. They can use the system to monitor and assess compliance with lending standards, detect potential risks, and implement regulatory measures effectively.

**1.4.5. Software Developers and Technologists:** Information technology professionals, software developers, and technologists interested in financial technology (Fintech) and machine learning applications represent another user category. They may explore the system for learning purposes, technical insights into machine learning algorithms, and opportunities for innovation and optimization in the lending domain.

**1.4.6. Researchers and Academics:** Researchers, academics, and scholars in the fields of finance, economics, and data science may also find value in the loan approval system. They can utilize the system to study lending practices, analyze loan approval trends, and conduct research on topics related to credit risk assessment and financial decision-making.

By catering to the diverse needs of these user groups, the loan approval system aims to enhance efficiency, accuracy, and transparency in the loan application and approval process while promoting financial inclusion and responsible lending practices.

1. **Problem Definition and Requirements:**

**2.1 Problem Statement:**

To design and implement the system using machine learning and data mining to predict the probability of the user to get loan or not from bank to improve the accuracy and to minimize the frauds. Banks, Housing Finance Companies and some NBFC deal in various types of loans like housing loan, personal loan, business loan etc in all over the part of countries. These companies have existence in Rural, Semi Urban and Urban areas. After applying loan by customer these companies validate the eligibility of customers to get the loan or not. This paper provides a solution to automate this process by employing machine learning algorithm. So, the customer will fill an online loan application form. This form consists of details like Sex, Marital Status, Qualification, Details of Dependents, Annual Income, Amount of Loan, Credit History of Applicant and others. To automate this process by using machine learning algorithm, First the algorithm will identify those segments of the customers who are eligible to get loan amounts so bank can focus on these customers.

**2.2 Requirements:**

1. Python Programming Language: The project will be implemented using Python, which is a popular language for machine learning and data analysis tasks.
2. Python Integrated Development Environment (IDE):

* PyCharm: PyCharm is a powerful IDE developed by JetBrains, which provides excellent support for Python development, debugging, and integration with various libraries and frameworks.
* Google Colab Notebook: Google Colab Notebook, which is a web-based interactive computing environment that allows you to combine code, visualizations, and narrative text.

1. Python Libraries and Frameworks:

* NumPy: For numerical computing operations.
* Pandas: For data manipulation and analysis.
* Scikit-learn: A machine learning library for Python, providing a wide range of algorithms and tools for model building, evaluation, and deployment.
* Matplotlib and Seaborn: For data visualization and plotting.

1. Version Control System: Git: A distributed version control system widely used for tracking changes in source code and collaborating on projects.
2. **Materials and Methods:**

This section outlines the methodology employed in this study to detect depression using machine learning techniques. The methodology encompasses data collection, preprocessing, feature extraction, and the selection and training of machine learning algorithms.

**3.1 About Dataset:**

This data was collected from kaggle.com: [Loan-Approval-Prediction-Dataset (kaggle.com)](https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset/data)

The loan approval dataset is a collection of financial records and associated information used to determine the eligibility of individuals or organizations for obtaining loans from a lending institution. It includes various factors such as cibil score, income, employment status, loan term, loan amount, assets value, and loan status. This dataset is commonly used in machine learning and data analysis to develop models and algorithms that predict the likelihood of loan approval based on the given features.

loan\_id: Unique id’s for each applicant

no\_of\_dependents: Number of Dependents of the Applicant

education: Education of the Applicant (Graduate/Not Graduate)

self\_employed: Employment Status of the Applicant

income\_annum: Annual Income of the Applicant

loan\_amount: Loan Amount

loan\_term: Loan Term in Years

cibil\_score: Credit Score

residential\_assets\_value

commercial\_assets\_value

luxury\_assets\_value

bank\_asset\_value

loan\_status: Loan Approval Status (Approved/Rejected)

**3.2 Handling Missing Values and Relabelling Categorical Features:**

By the data cleaning scans, we have confirmed that there are no null values in this dataset.

no\_of\_dependents, education, self\_employed and loan\_status are categorical columns. There are a total 4269 rows in this dataset, with 13 columns (features). There are 2656 data with an approved loan\_status, which is about 62.2% compared to the "rejected" group. The dataset is slightly imbalanced, but it is acceptable, and we don't need to rebalance it. Other columns are numerical.

**For the Self-Employed feature:**

* The original categorical values representing "No" and "Yes" were converted into binary codes.
* "No" was encoded as 0 and "Yes" as 1, creating a binary representation of the feature.

**For the Loan Status feature:**

* The original categorical values representing "Not approved" and "Approved" were transformed into binary codes.
* "Not approved" was encoded as 0 and "Approved" as 1, creating a binary representation of the loan approval status.

**For the Education (Graduated) feature:**

* The initial categorical values representing "Not graduated" and "Graduated" were converted into binary codes.
* "Not graduated" was encoded as 0 and "Graduated" as 1, resulting in a binary representation of the education attainment status.

Converting categorical features into binary representations simplifies data analysis and modeling tasks, enabling the application of machine learning algorithms that require numerical inputs. This transformation facilitates the interpretation of the features and enhances the efficiency of analytical processes.

**3.3 Feature Engineering:**

Feature Engineering is the process of creating new features or transforming existing features to improve the performance of a machine-learning model. It involves selecting relevant information from raw data and transforming it into a format that can be easily understood by a model. The goal is to improve model accuracy by providing more meaningful and relevant information.

**3.3.1 For the income-to-loan score:**

Generated a new feature named 'income\_to\_loan\_score' to assess the relationship between income, loan term, loan amount, and credit score. The formula calculates the income-to-loan score by multiplying the product of the annual income, loan term, and credit score, then dividing it by the loan amount. The income-to-loan score serves as a comprehensive indicator of an individual's financial capacity relative to the loan terms and creditworthiness. This feature encapsulates crucial factors influencing loan repayment capability and risk assessment, providing valuable insights for machine learning models in predicting loan outcomes and optimizing lending decisions.

**3.3.2 Dropping Columns:**

The columns dropped include 'loan\_status', 'self\_employed', 'education', 'bank\_asset\_value', and 'commercial\_assets\_value'. The exclusion of these columns aims to streamline the feature set for analysis or modeling purposes, focusing on the most relevant and informative features while removing potentially redundant or less impactful variables. This process helps enhance the efficiency and effectiveness of machine learning algorithms by reducing dimensionality and minimizing noise in the dataset.

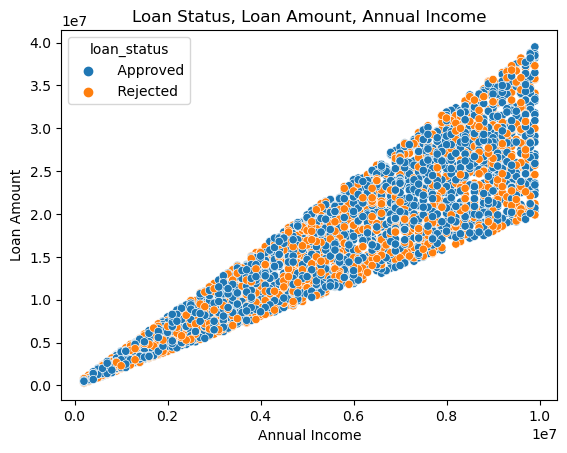
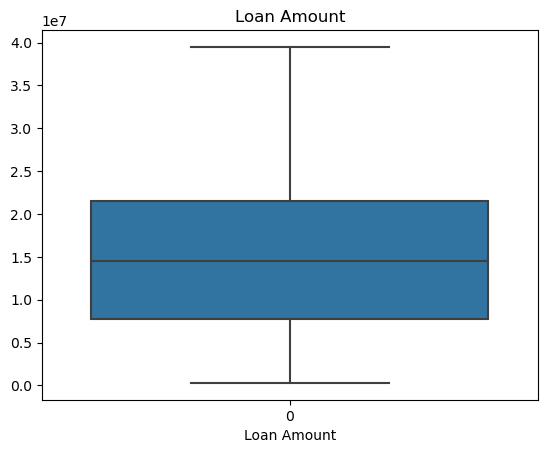
**4 Data Analysis & Methodology:**

**4.1 Exploratory Data Analysis:**

It involves analyzing and visualizing data to understand its key characteristics, uncover patterns, and identify relationships between variables refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables.

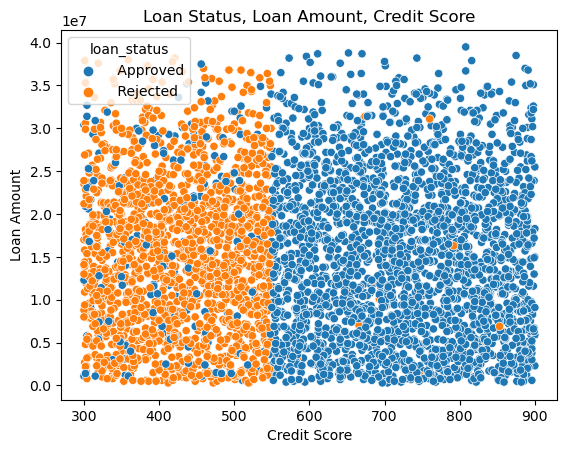
**4.2 Data Distributions:**

**4.2.1 Loan status and loan amount:**



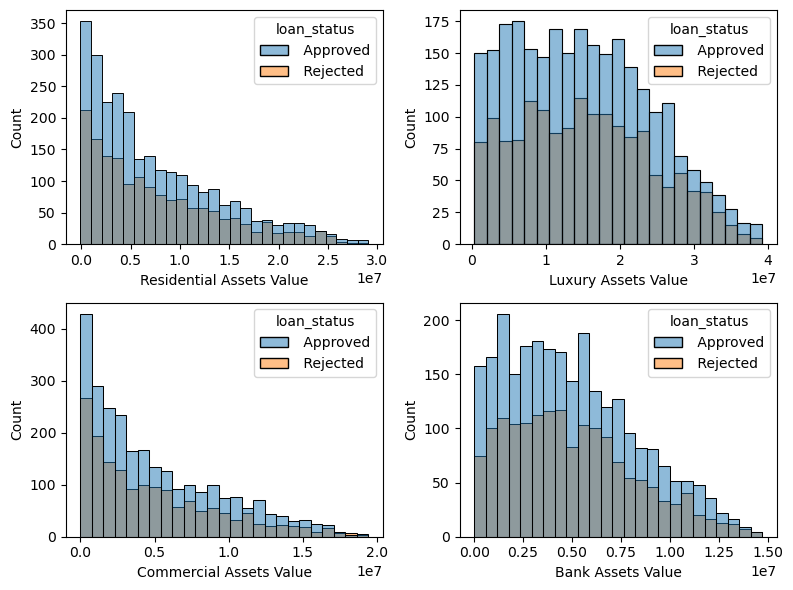
When annual income increases, the loan amount tends to increase. But the annual income doesn't show an obvious trend with the loan status. In this dataset, the applicants who have the lower annual income have a narrow range in loan amounts. Vise Versa, the applicants who have the higher lower annual income have a wider range in the loan amounts. Besides that the lenders will only accept the loan amount that aligns with the annual income in order to ensure the applicants have ability to pay the loan back, it is easy to imagine the applicants who have higher annual income have more flexibility on the amount of the loan, whatever for themselves or for the lenders. In this dataset, the applicants who has highest annual income has been approved when they apply for the highest loan amount, but at the same time, the applicants who have the highest annual income have chances of being rejected when they apply lower loan amount. It can be caused by different lenders and other conditions of the applicants. Let's use some code to take a closer look at it.

**4.2.2 Credit Score:**



The loan amount is not the main reason that causes their applications to be rejected. Taking a closer look at this subset, their credit scores (cibil\_score) are considered as "Poor".  
According to Equifax, the standard credit scores are: 300-579: Poor. 580-669: Fair. 670-739: Good. 740-799: Very good. The loan status is highly related to the credit score. It is also interesting to see the credit score that separates the loan status is not 579 which is the highest score of the "poor" credit score. In other words, poor credit scores which are above 540 - 550 still have a good chance of being approved by loan lenders. This could be attributed to lenders' flexibility or specific factors that impact approval decisions. However, we also notice a puzzling trend: some of the applicants with high credit scores (above 740) were still rejected.

**4.2.3 residential\_assets\_value, commercial\_assets\_value, luxury\_assets\_value, bank\_asset\_value:**

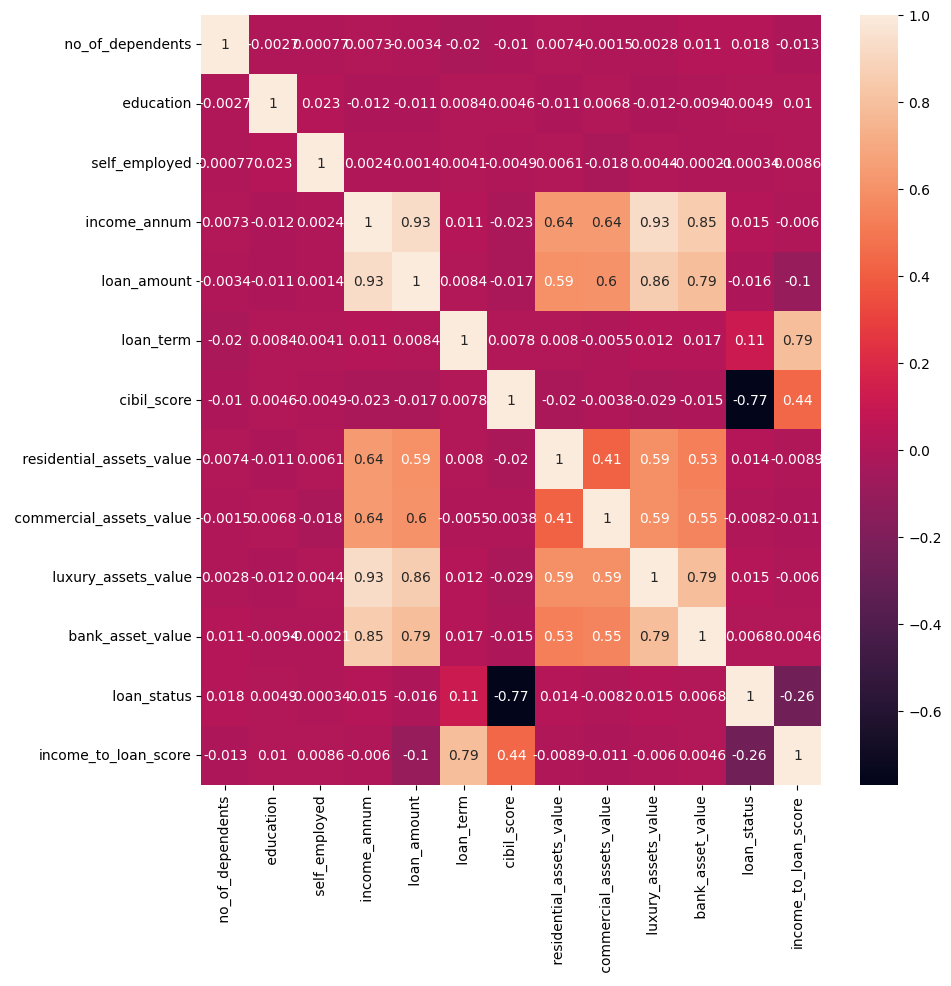


**Residential Assets Value:** This is likely a measure of the total value of residential properties or real estate assets owned by the individuals or organizations in the dataset.

**Commercial Assets Value:** This could represent the total value of commercial properties or business-related assets owned by the individuals or organizations in the dataset. Commercial properties might include office buildings, retail spaces, warehouses, and similar assets.

**Luxury Assets Value:** This might refer to the total value of high-end or luxury items owned by individuals or organizations. These could include items such as luxury vehicles, valuable artwork, jewelry, and other premium possessions.

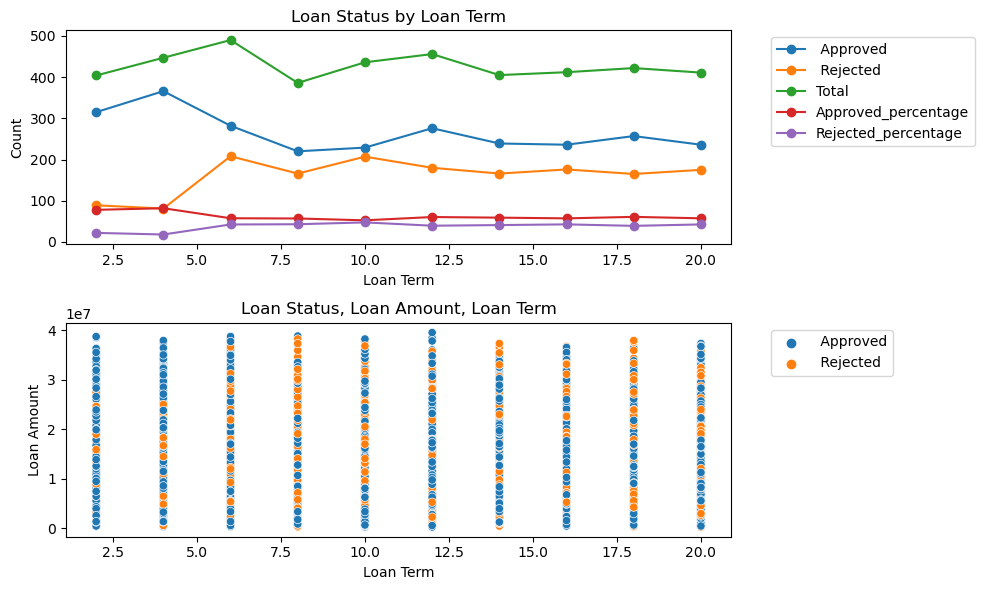
**Bank Asset Value:** This is possibly the total value of assets held by the bank or lending institution itself. It might include cash reserves, investments, and other financial assets."



All the asset values have moderate to strong positive linear relationships with the annual income. As the applicants who have more annual income tend to have more flexibility on purchasing the properties with higher asset values, especially the luxury assets value. The differences in correlation strengths might be due to various factors in the data and the context of the variables:

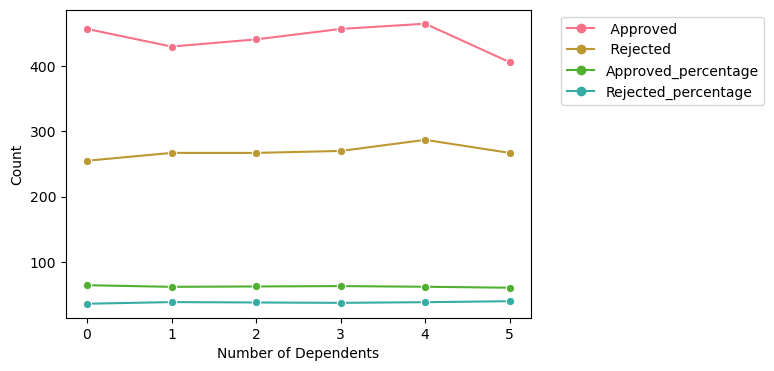
* **Nature of Assets:** Luxury assets and bank assets might have a stronger connection to an individual's income. People with higher incomes might be more likely to have luxury assets or maintain bank assets. Residential and commercial assets might be influenced by other factors such as location, real estate market trends, and investment strategies, which could result in a slightly weaker correlation with income.
* **Economic Status:** People with higher incomes could afford luxury items and have larger bank assets, resulting in a tighter correlation. Residential and commercial assets might be affected by broader economic trends and market conditions, leading to a less direct correlation with individual income.
* **Diverse Income Sources:** Some individuals might have diverse income sources beyond their primary job, impacting the relationship between assets and annual income.
* **Data Variability:** Natural variability in data could contribute to variations in correlation strength. A smaller dataset might result in less precise estimates of correlation.
* **Outliers and Extreme Values:** Extreme values or outliers in the data can influence correlation values. If a few individuals with extremely high income also have high asset values, it could strengthen the correlation.

**4.2.4 Loan Term:**



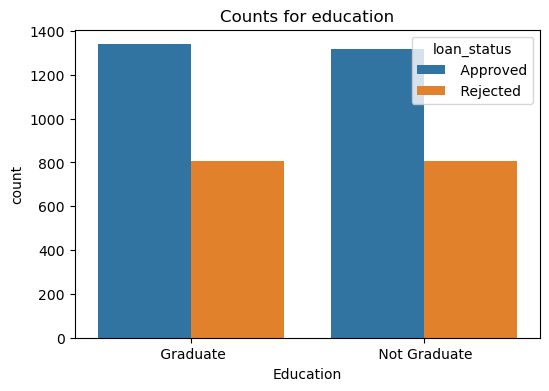
The total observations for each of the loan terms are very even, we appreciate the owner of this dataset, so we can analyze the data much easier without worrying about the balancing. The shortest loan term in this dataset, which is 2 years, gets one of the most chances for being approved by the lenders, compared to other loan terms, except the 4 years loan term which gets the most chances for being approved. When the loan term is more than 4 years, the chance of being rejected has significantly increased, vice versa, the chance of being approved has dropped. Until the loan term equals to 8, both chances of being approved and rejected tend to be normal. When the loan term reaches 10 years, it is a loan term whose chance of being approved and rejected are approximately the same. After 10 years, the trend is becoming more consistent and the chance of being approved is slightly higher than the chance of being rejected. In the group of 2 years of loan term, the applicants who apply for loans for more than $30,000,000 have all been approved. Similar situation as the applicants in the group of the 4 years of loan term, who apply for more than \$30,000,000 loan amount. Compared to the higher loan amount, the lower loan amount has more chance of being rejected, especially when the applicants want the 4 years of loan term. In my experience, short term loans with lower loan amounts should be even easier to pay back, compared to larger amounts. The low credit scores should be one of the most important reasons that the applications have been rejected, another reason can be the annual income since most of the applicants in this subset (group) have an annual income lower than the median. Other than those 2 reasons, I guess the number of dependents can also be a reason for rejection (in this subset, the number of dependents is all more than 2), especially when the applicants have lower annual income.

**4.2.5 Number Of dependents:**



By checking the line chart above which presents the percentage of the applicants being approved or rejected by the number of dependents, we see 2 percentage lines are very evenly, even though we can see a wave over the "approved" line, it might be because of the total number of the applicants. So, there is no such obvious trend between them.

**4.2.6 Education:**



The bar chart compares the count of approved and rejected loans between two educational categories: “Graduate” and “Not Graduate.” Graduates have a higher count of both approved (blue bar) and rejected (orange bar) loans compared to non-graduates. This suggests that educational attainment may influence loan approval rates.

**4.2.7 Self Employed:**

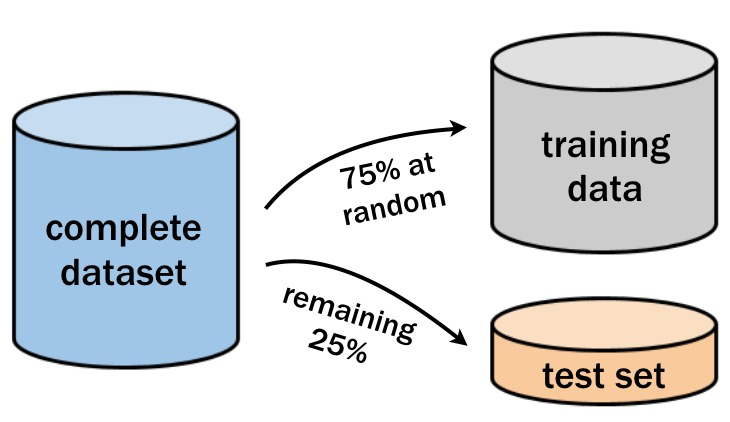
Same thing as the self\_employed - there are no significant differences between the self\_employed and other variables in this dataset.

**5. Algorithms and Methodologies followed:**

**5.1 Model Training**

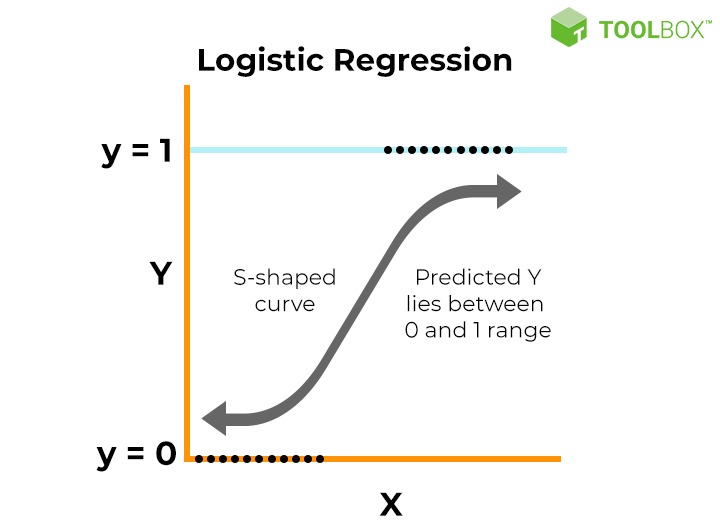
#### **5.1.1 Creating Train and Test Dataset**

Creating train and test datasets is a fundamental step in machine learning model development. The process involves splitting the available data into two subsets: one for training the model and the other for evaluating its performance. The training dataset is used to train the model by exposing it to examples with known input features and corresponding target labels. The model learns from these examples, adjusting its parameters to minimize the difference between its predictions and the actual target labels. The test dataset, on the other hand, is kept separate from the training process and is used to evaluate the model's performance. It consists of examples with known input features but with withheld target labels. The model's predictions on the test dataset are compared against the true target labels to assess its accuracy, precision, recall, and other performance metrics. By splitting the data into training and test sets, we can ensure that the model's performance is properly evaluated on unseen data, providing an indication of its ability to generalize to new, unseen examples. This process helps prevent overfitting, where the model performs well on the training data but poorly on unseen data, thus improving the model's reliability and effectiveness in real-world applications. Once trained, the model can then be evaluated on a separate validation dataset to assess its generalization performance and fine-tune hyperparameters if necessary. Finally, the trained model can be deployed to make predictions on new, unseen data, such as loan applications, to assist in decision-making processes.

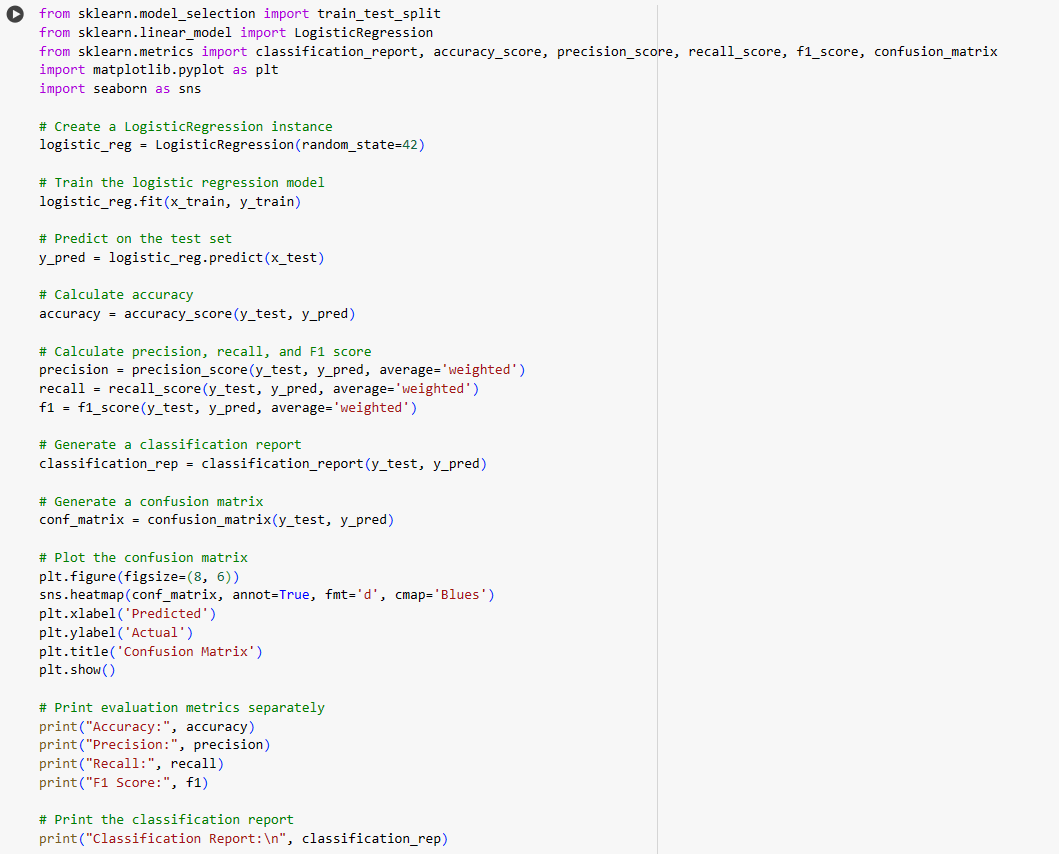


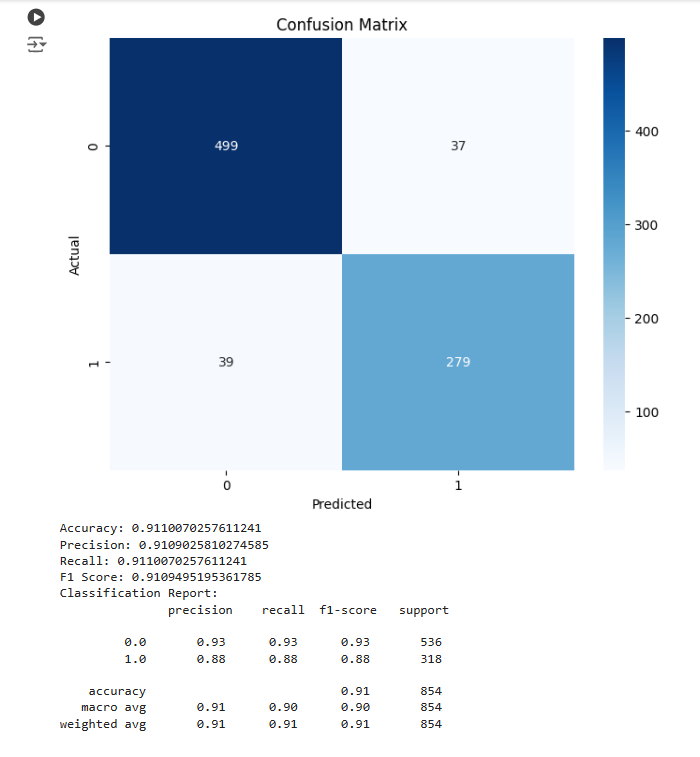
**5.2 Algorithms Used:**

**5.2.1 Logistic Regression**

Logistic regression is a foundational statistical technique used extensively in binary classification tasks, making it particularly relevant in scenarios such as predicting loan approval outcomes. In the context of loan approval prediction, logistic regression serves as a powerful tool for assessing the likelihood of a loan application being approved or denied based on various input features. At its core, logistic regression models the probability that a given example belongs to a particular class, typically represented as a binary outcome (e.g., approved or not approved). The logistic regression model achieves this by employing the logistic function, also known as the sigmoid function, to map the linear combination of input features to a probability between 0 and 1. The logistic function is defined as: *σ*(*z*)=1+*e*−*z*1 where *z* represents the linear combination of input features and their associated weights (coefficients), plus an intercept term. The logistic regression model learns the optimal values for these coefficients during the training process, adjusting them to maximize the likelihood of correctly classifying examples in the training dataset. In mathematical terms, the linear combination *z* is expressed as: ***z*=*β*0 +*β*1 *x*1 +*β*2 *x*2 +…+*βn* *xn*** Here, *β*0 is the intercept term, and *β*1 ,*β*2 ,…,*β**n* are the coefficients associated with each input feature *x*1 ,*x*2 ,…,*xn* , respectively. Once the logistic regression model has learned the optimal coefficients, it predicts the probability of the positive class (e.g., loan approval) using the logistic function. This predicted probability is then compared against a predefined threshold (typically 0.5), and if the probability exceeds the threshold, the example is classified as belonging to the positive class; otherwise, it is classified as belonging to the negative class. In summary, logistic regression offers a interpretable and effective approach for binary classification tasks like loan approval prediction, providing insights into the probability. 

**5.2.1.1 Applying Logistic Regression:**





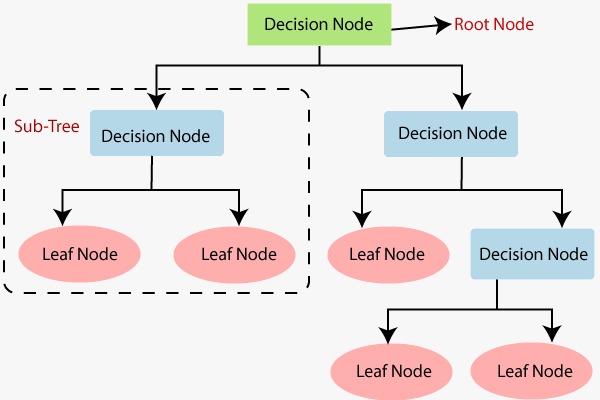
**5.2.2 Decision Tree**

Decision trees are versatile and intuitive machine learning models used for both classification and regression tasks. They operate by recursively partitioning the feature space into smaller regions, guided by a set of decision rules based on input features. At each node of the tree, a decision is made based on a feature value, leading to the creation of branches representing different outcomes. This process continues until a stopping criterion is met, such as reaching a maximum depth or having a minimum number of samples in each leaf node.

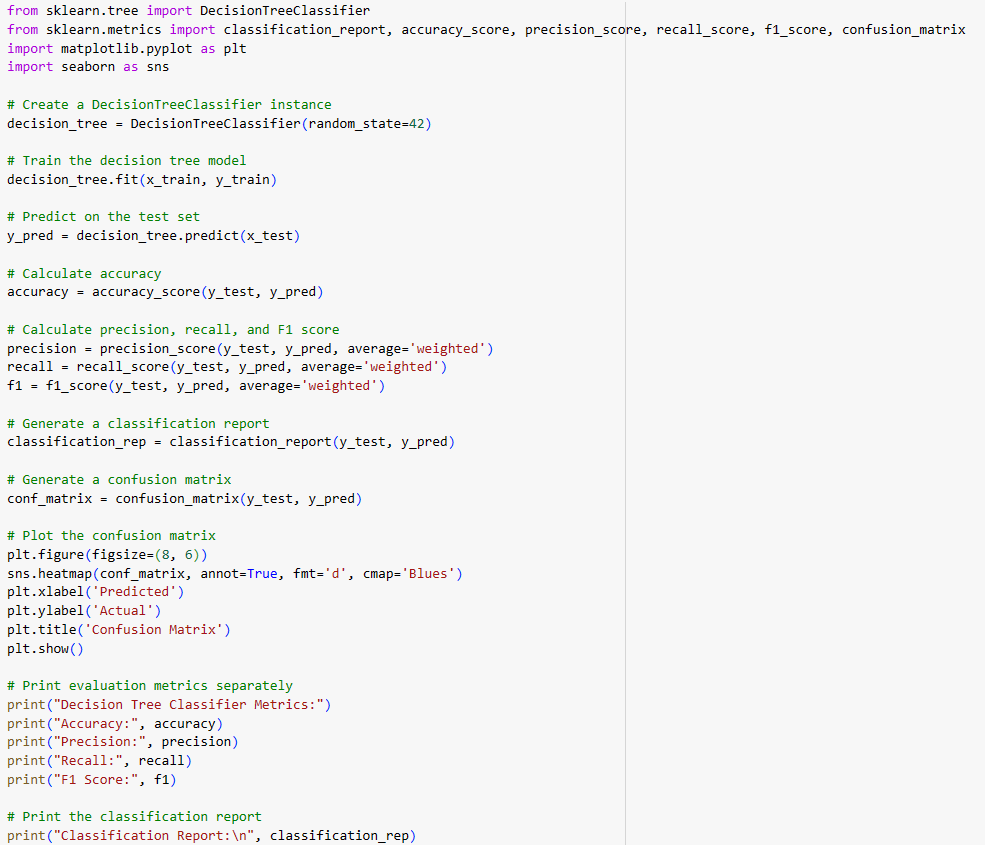
Formulas:

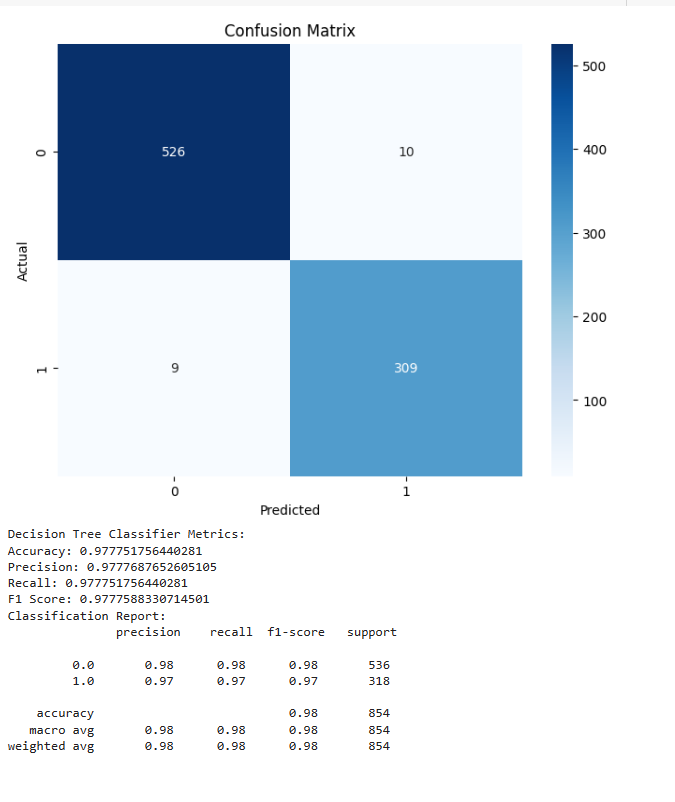
* Gini Impurity (for classification):

Decision trees are attractive due to their simplicity, interpretability, and ability to handle nonlinear relationships in the data. However, they can be prone to overfitting, especially when growing deep trees with complex decision boundaries. Regularization techniques such as pruning and setting minimum sample requirements help mitigate overfitting and improve generalization performance.



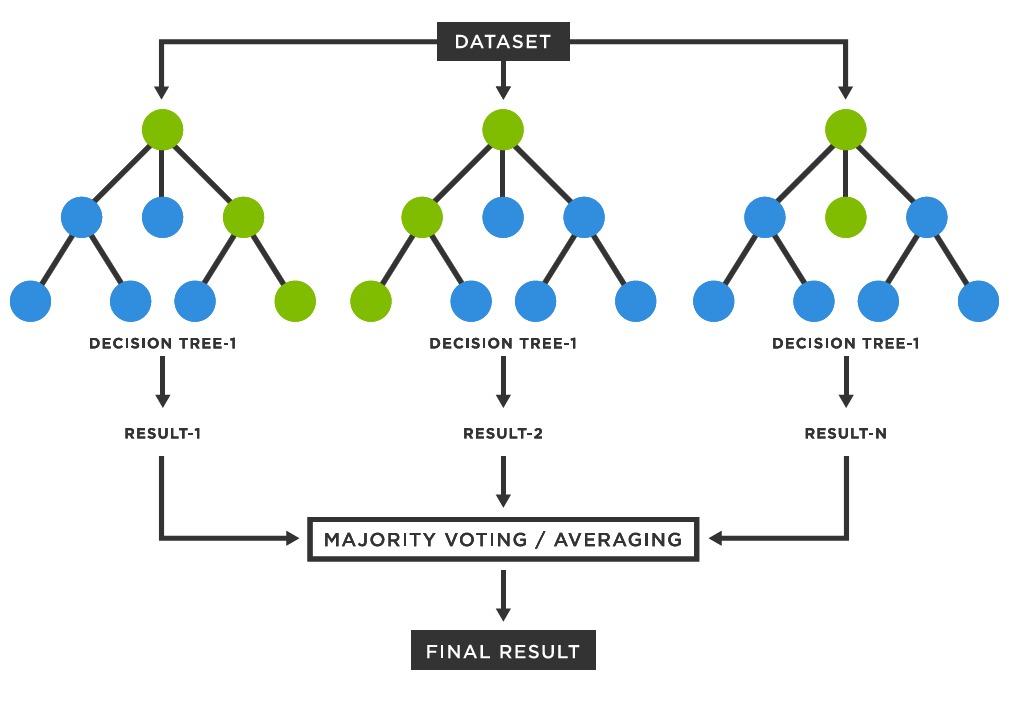
**5.2.2.1 Applying Decision Tree:**



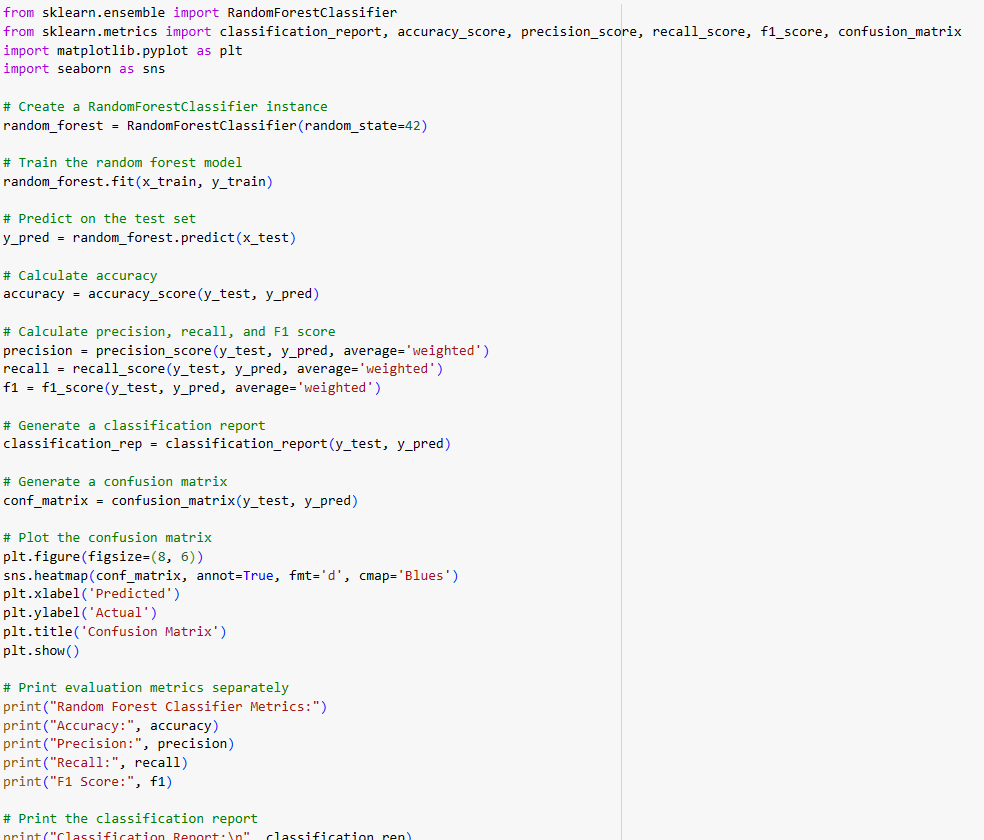


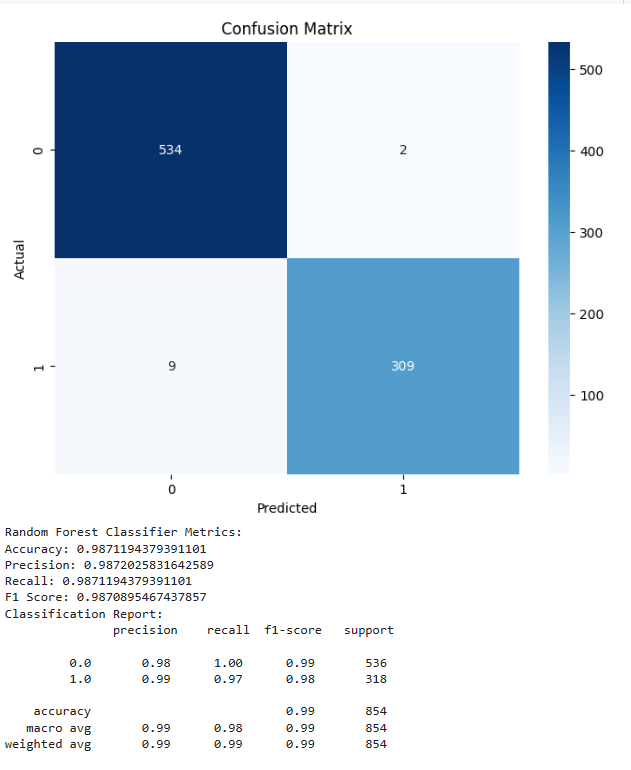
**5.2.3 Random Forest:**

Random Forest is a powerful ensemble learning algorithm widely used in the field of Artificial Intelligence and Machine Learning. Introduced by Leo Braiman in 2001, this algorithm leverages the strength of multiple decision trees to enhance prediction accuracy and robustness. The core concept of Random Forest involves creating a multitude of decision trees, each trained on a different subset of the data, and then combining the predictions of these individual trees to generate a final prediction. Initially, the algorithm employs random sampling to select subsets of data from the dataset. This ensures diversity in the training process and helps prevent the model from overfitting to specific patterns in the data. Subsequently, decision trees are constructed for each sample, where each tree learns to make predictions based on the features present in the data. This process involves recursively partitioning the feature space to create hierarchical decision rules. Once the decision trees are constructed, predictions are obtained from each tree independently. These predictions are then subjected to a voting mechanism, where each predicted result undergoes a voting process. Random Forests enhance generalization capabilities, making them well-suited for complex datasets with high-dimensional feature spaces. Additionally, the algorithm is known for its high accuracy and efficiency, making it suitable for processing large datasets efficiently. Key components of Random Forests include decision trees, which serve as the building blocks of the ensemble. Feature bagging is incorporated during training to prevent the dominance of any single feature, ensuring a balanced contribution from all features. The majority voting mechanism is used to determine the final prediction based on individual tree outputs, facilitating robust decision-making. Hyperparameters such as the number of trees, maximum number of features considered for splitting a nod, and minimum number of samples required to split an internal node can be tuned to optimize model performance Random Forests find widespread applications across various domains, including finance, healthcare, and image analysis. Their ability to handle complex datasets, high-dimensional feature spaces, and provide insights into feature importance makes them particularly favored in real-world scenarios. In conclusion, Random Forests represents a versatile and reliable algorithm in the realm of AI and ML, offering a robust solution for classification and regression tasks by harnessing the collective power of multiple decision trees.



**5.2.3.1 Applying Random Forest:**





**5.2.4 SVM**

Support Vector Machines (SVMs) are powerful supervised machine learning algorithms used for classification tasks by identifying an optimal line or hyperplane that maximizes the separation between different classes in an N-dimensional space. SVMs were developed in the 1990s by Vladimir N. Vanik and colleagues, with a key paper published in 1995. SVMs excel in distinguishing between two classes by finding the best hyperplane that maximizes the margin between the closest data points of opposite classes. The choice of kernel functions, such as linear, polynomial, radial basis function (RBF), or sigmoid kernels, is crucial when dealing with non-linear data to transform it into a higher-dimensional space for linear separation24.

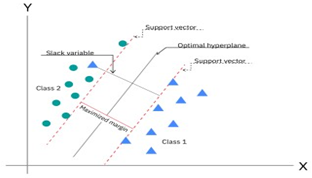
**Linear SVMs**

Linear SVMs are used with linearly separable data; this means that the data does not need to undergo any transformations to separate the data into different classes. The decision boundary and support vectors form the appearance of a street, and Professor Patrick Winston from MIT uses the analogy of "[fitting the widest possible street](https://ocw.mit.edu/courses/6-034-artificial-intelligence-fall-2010/resources/mit6_034f10_svm/)" (link resides outside ibm.com) to describe this quadratic optimization problem. Mathematically, this separating hyperplane can be represented as: wx + b = 0

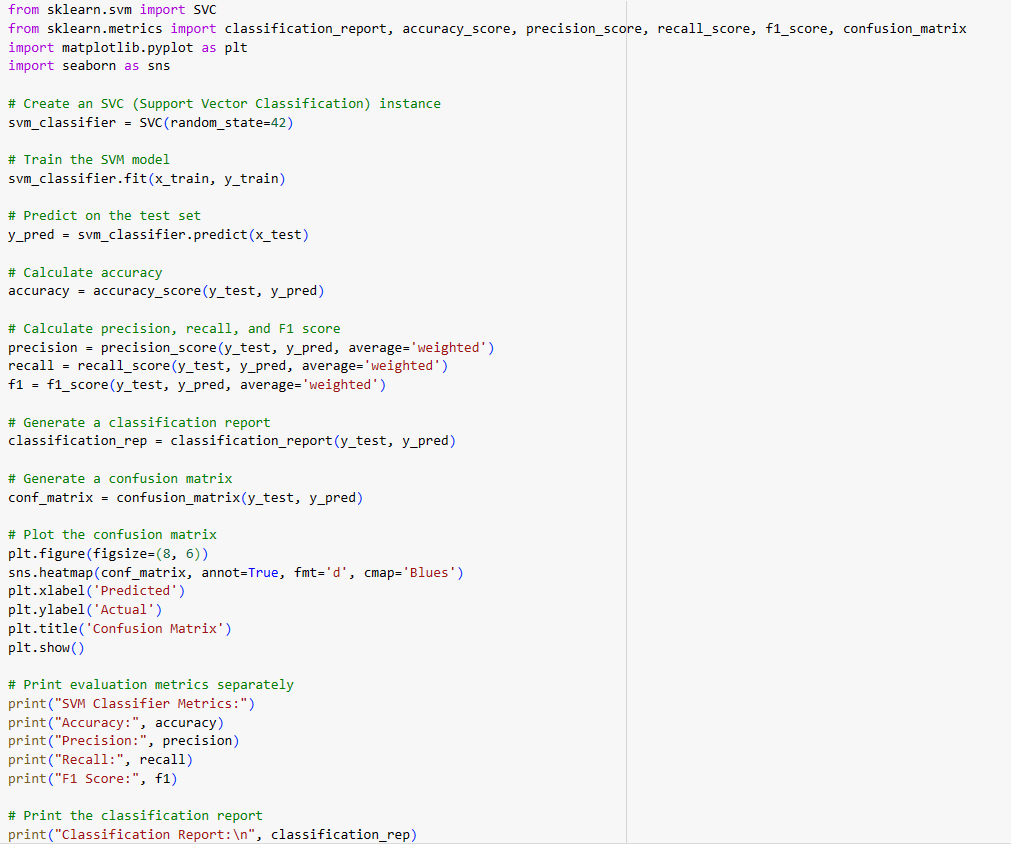
where w is the weight vector, x is the input vector, and b is the bias term.

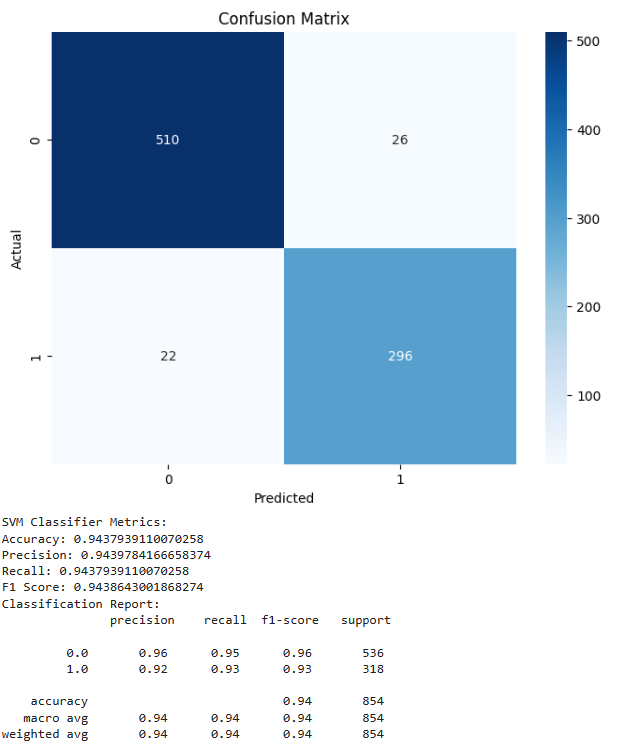
There are two approaches to calculating the margin, or the maximum distance between classes, which are hard-margin classification and soft-margin classification. If we use a hard-margin SVMs, the data points will be perfectly separated outside of the support vectors, or "off the street" to continue with Professor Hinton’s analogy. This is represented with the formula, (wxj + b) yj ≥ a, and then the margin is maximized, which is represented as:

 max ɣ= a / ||w||, where a is the margin projected onto w. Soft-margin classification is more flexible, allowing for some misclassification through the use of slack variables (`ξ`). The hyperparameter, C, adjusts the margin; a larger C value narrows the margin for minimal misclassification while a smaller C value widens it, allowing for more misclassified data.



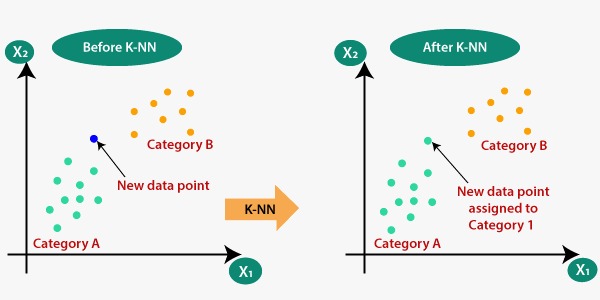
**5.2.4.1 Applying SVM**



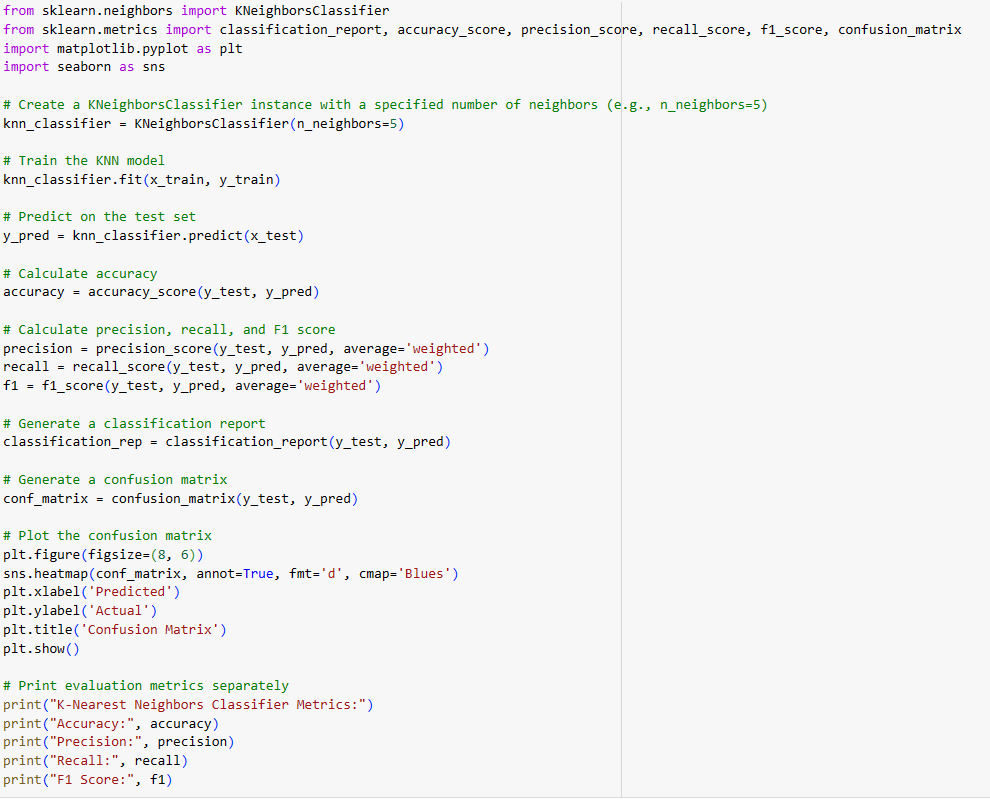


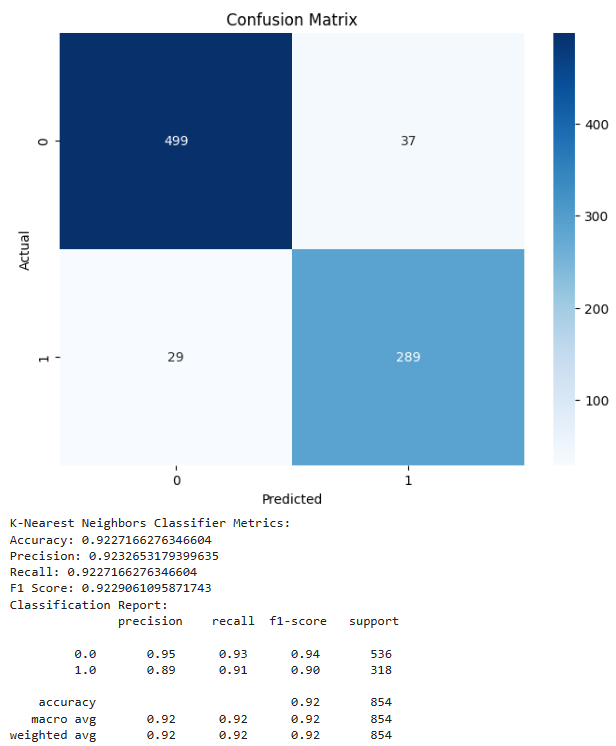
**5.2.5 K-Nearest Neighbors (KNN)**

The K-Nearest Neighbors (KNN) algorithm is a fundamental supervised learning technique in Artificial Intelligence (AI) and Machine Learning (ML) that has been widely used for both classification and regression problems. The core idea behind KNN is to find the k-nearest data points to a given test data point and use these nearest neighbors to make a prediction. The value of k is a hyperparameter that needs to be tuned, and it represents the number of neighbors to consider. Initially, data is loaded into memory, typically accomplished with libraries such as pandas or NumPy. Following this, the dataset is divided into training and test sets, with the former used to train the algorithm and the latter employed for evaluation. Normalization of the data is crucial to ensure that each feature contributes equally to the distance metric calculation. Once normalized, distances between the test data point and every data point in the training set are computed. The algorithm then identifies the k-nearest neighbors based on these distances. For classification tasks, KNN assigns the test data point to the class that is most frequent among its k-nearest neighbors. Conversely, for regression tasks, the algorithm assigns the test data point the average value of its k-nearest neighbors. Subsequently, the performance of the KNN algorithm is assessed using various metrics like accuracy, precision, recall, and F1-score. The choice of distance metric significantly impacts the algorithm's performance. Commonly used metrics include Euclidean distance, Manhattan distance, and Minkowski distance, with the latter being a generalized form allowing for the creation of other metrics. Weighted KNN is a variant that assigns greater importance to nearer neighbors by employing a weight calculated using a vein function, often the inverse function. Selecting the appropriate value for k, termed parameter tuning, is crucial for achieving higher accuracy. While there's no definitive method for determining k, it depends on the problem type and business context. A value of five is commonly preferred to avoid noise and overfitting. KNN offers simplicity and requires no training before prediction. However, it struggles with large datasets due to the computational cost of calculating distances and necessitates feature scaling to prevent erroneous predictions. Real-world applications of KNN span recommendation systems, finance (e.g., risk assessment in credit data), pattern recognition, data mining, financial market predictions, and intrusion detection. In conclusion, KNN remains a powerful technique in AI and ML, favored for its simplicity, adaptability, and effectiveness across various domains despite its limitations with large datasets and scaling requirements.



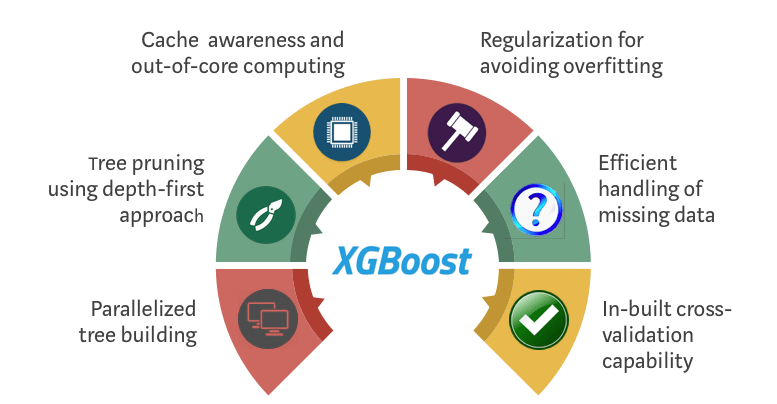
**5.2.5.1 Applying K-Nearest Neighbors (KNN)**



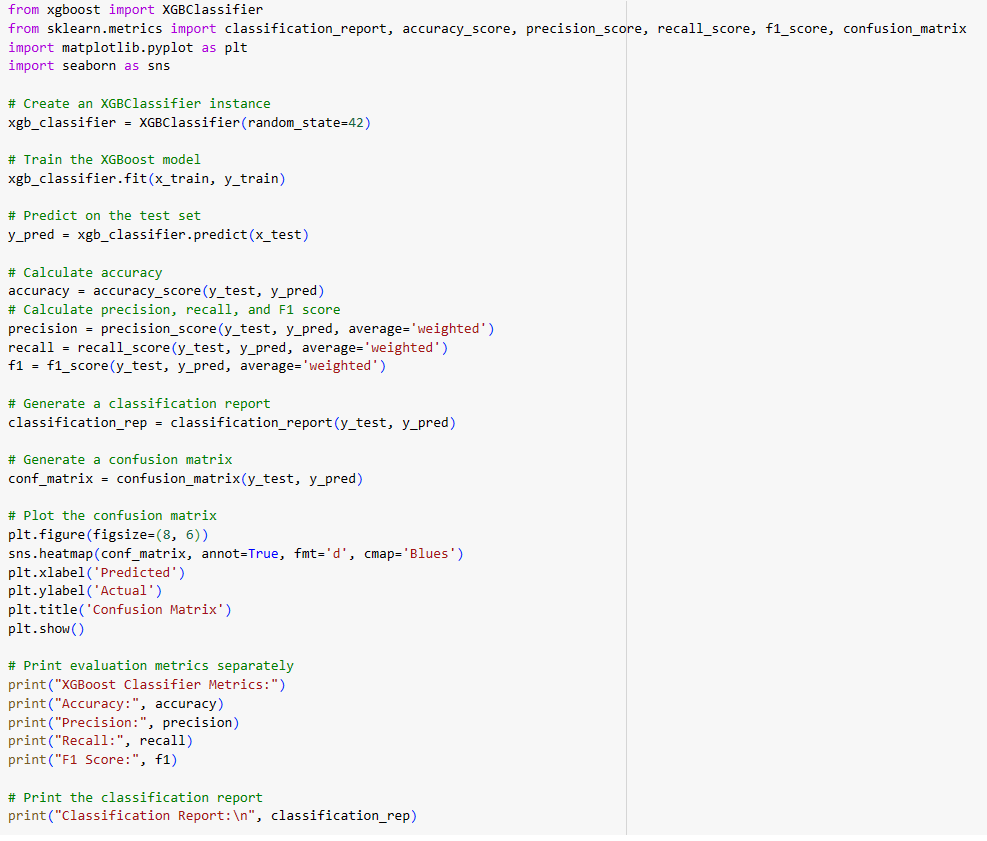


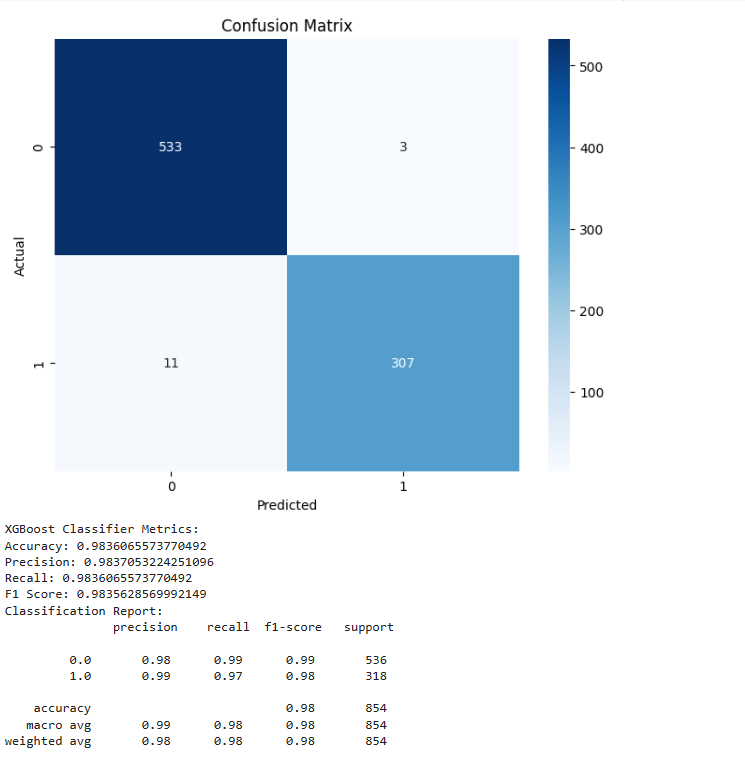
**5.2.6 XGBoost**

XGBoost, an abbreviation for Extreme Gradient Boosting, represents a cutting-edge advancement in the realm of Artificial Intelligence and Machine Learning (AI ML). Developed by Tianqi Chen, this algorithm has garnered immense popularity owing to its scalability, accuracy, and efficiency. Essentially, XGBoost serves as an open-source implementation of the Gradient Boosting Machine (GBM) framework but introduces several innovative features and optimizations to elevate model performance. One of the standout characteristics of XGBoost lies in its incorporation of regularization techniques such as L1 (Lasso) and L2 (Ridge). These techniques play a pivotal role in preventing overfitting by penalizing large model weights, thereby enhancing the generalization capability of the model. Additionally, XGBoost integrates tree pruning mechanisms to eliminate splits that contribute insignificantly to the overall performance of the tree. This pruning reduces complexity, leading to faster and more accurate predictions. Another notable feature is its ability to handle missing data seamlessly, ensuring robust predictions even in scenarios where certain features are absent. Furthermore, XGBoost offers built-in cross-validation capabilities, facilitating more accurate model evaluation and parameter tuning. Its evolution and popularity are underscored by its remarkable track record in Kaggle competitions, where it has contributed to numerous victories. The availability of implementations in multiple programming languages such as R, Python, Scala, Julia, and Perl further underscores its accessibility and versatility. Advantages of XGBoost encompass its scalability, accuracy, speed, and robustness to outliers and noise in data. These traits make it an indispensable tool across a myriad of domains including finance, healthcare, natural language processing, and computer vision. Its proficiency in handling both structured and unstructured data, coupled with its ability to generate highly accurate predictive models, positions XGBoost as a formidable asset in the AI ML landscape. In conclusion, XGBoost emerges as a sophisticated and efficient machine learning algorithm, offering a plethora of advanced features, optimizations, and advantages over traditional GBM models. Its widespread adoption, robustness, and versatility underscore its significance in empowering data scientists and machine learning practitioners to develop high-performance models for diverse AI ML applications.



**5.2.6.1 Applying XGBoost**

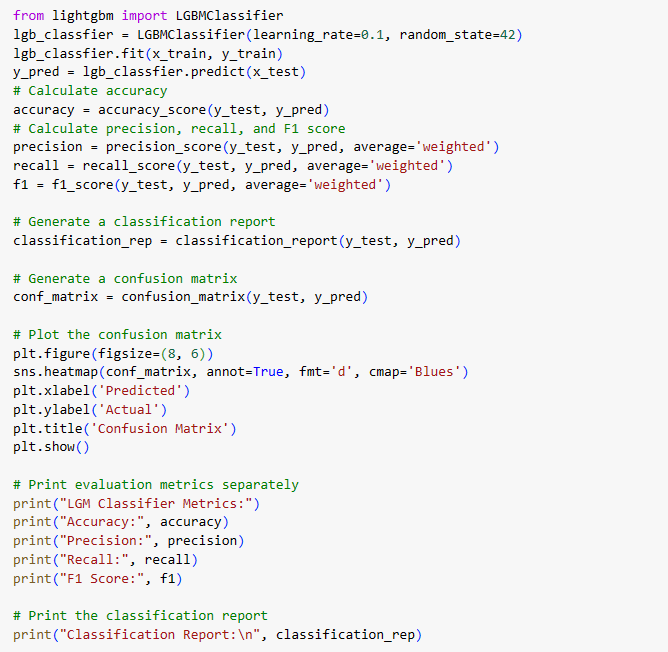


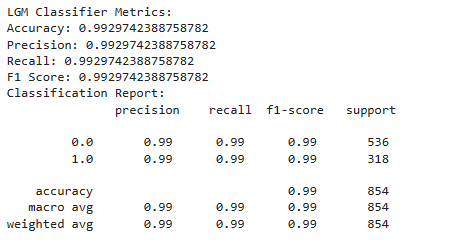


**5.2.7 LightGBM**

LightGBM, short for Light Gradient Boosting Machine, is a powerful gradient boosting framework that has gained significant popularity in the field of Artificial Intelligence and Machine Learning. Developed by Microsoft, LightGBM is an open-source, distributed framework designed to enhance efficiency, scalability, and accuracy in model training. It leverages tree-based learning algorithms to construct a robust ensemble learner by sequentially adding weak learners in a gradient descent manner. This approach optimizes memory usage and training time, making LightGBM a preferred choice for various machine learning tasks. One of the key strengths of LightGBM lies in its ability to handle large-scale data efficiently. It offers several advantages that set it apart from other boosting frameworks. These advantages include faster training speed, higher efficiency, lower memory usage, better accuracy, and support for parallel, distributed, and GPU learning. LightGBM achieves these benefits through innovative techniques like Gradient-based One-Side Sampling (GOSS) and histogram-based algorithms for efficient tree construction. LightGBM's core parameters play a crucial role in fine-tuning model performance. Parameters such as learning rate, number of leaves, maximum depth, regularization terms, and optimization strategies are essential for customizing the behavior and performance of LightGBM models during training. For instance, the learning rate determines the step size during gradient descent, while the number of leaves controls the complexity of the model. Tuning these parameters is vital for achieving optimal model performance across various machine learning tasks. Moreover, LightGBM supports various objectives such as regression, binary classification, and multiclass classification, making it versatile for different types of predictive modeling. The framework also offers features like feature fraction and bagging fraction to enhance model diversity and reduce overfitting. Additionally, regularization parameters like lambda\_l1 and lambda\_l2 help prevent overfitting by penalizing large coefficients. In terms of installation and usage, setting up LightGBM involves installing necessary dependencies, cloning the repository from GitHub, building the framework using CMake, and installing the Python package using pip. Proper installations ensure that users can leverage LightGBM's efficient algorithms and functionalities effectively for machine learning tasks. Overall, LightGBM stands out as a high-performance gradient boosting framework that excels in handling large datasets, optimizing memory usage, and achieving superior model accuracy. Its efficient training speed, lower memory footprint, and support for parallel processing make it a valuable tool for data scientists and machine learning practitioners seeking to build accurate and scalable models in AI and ML applications.

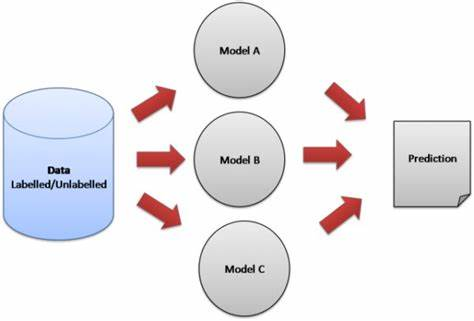
**5.2.7.1 Applying LightGBM**



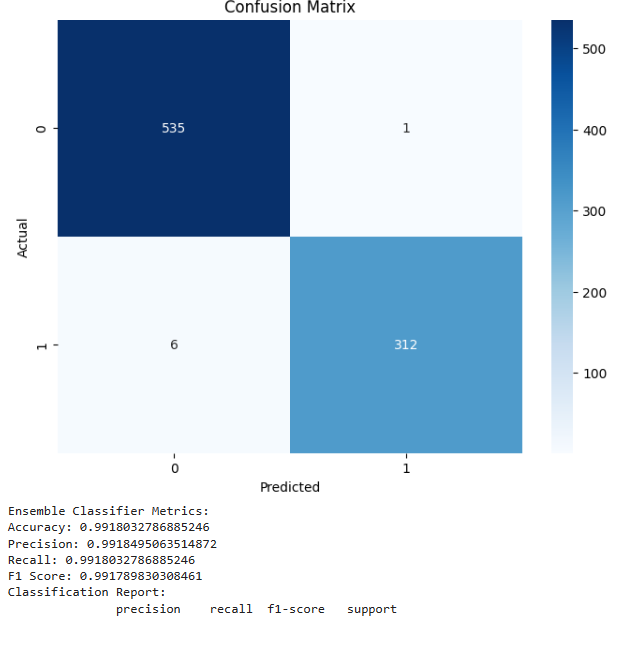


**5.2.8 Ensembling**

Ensembling stands as a formidable technique within the realm of machine learning, offering a robust approach to enhancing predictive accuracy and fortifying models against various data intricacies. At its core, ensembling revolves around the notion of amalgamating multiple individual models, thus harnessing the collective wisdom ingrained within each, to attain superior predictive outcomes. This approach effectively leverages the diverse strengths and perspectives inherent in different algorithms, consequently mitigating the limitations and biases that may arise from relying solely on a single model. Two prominent methodologies underpin ensembling: bagging and boosting. Bagging, exemplified by algorithms like Random Forests, entails training numerous models on distinct subsets of the dataset and subsequently aggregating their predictions, thereby fostering a collective intelligence that often outperforms individual models. Conversely, boosting iteratively refines weak learners by prioritizing the rectification of errors made by preceding models, culminating in a final prediction that synthesizes the insights gleaned from each successive iteration. Notable examples of boosting algorithms include Gradient Boosting Machines (GBM), XGBoost, and LightGBM, renowned for their efficacy in classification and regression tasks. Despite its computational demands, ensembling offers manifold advantages. Beyond its propensity for enhancing predictive accuracy, ensembling engenders heightened generalization capabilities, thereby enabling models to effectively navigate noisy or outlier-laden datasets. This resilience to data complexities positions ensembling as a versatile strategy, indispensable for tackling the multifaceted challenges encountered across diverse domains within the machine learning landscape. In essence, ensembling emerges as a cornerstone technique in the pursuit of refining model performance and addressing real-world complexities. Its capacity to seamlessly integrate multiple models into a unified predictive framework underscores its significance as a go-to strategy for augmenting the efficacy and reliability of machine learning solutions across a myriad of applications and industries



**5.2.8.1 Applying Ensemble learning**



**6. References**

* [Loan-Approval-Prediction-Dataset (kaggle.com)](https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset/code?datasetId=3523739&sortBy=voteCount)
* [Machine Learning Tutorial (geeksforgeeks.org)](https://www.geeksforgeeks.org/machine-learning/)
* [Machine Learning Tutorial (tutorialspoint.com)](https://www.tutorialspoint.com/machine_learning/index.htm)
* [Loan Information](https://www.investopedia.com/terms/l/loan.asp)
* [Cibil Score Information](https://www.equifax.com/personal/education/credit/score/what-is-a-credit-score/#:~:text=300%2D579%3A%20Poor,740%2D799%3A%20Very%20good)

**Conclusion**

In conclusion, this data analysis project aimed to explore the relationships between various variables in a dataset and the loan approval status. Through a comprehensive exploratory data analysis (EDA), we examined key features such as loan amount, annual income, loan term, asset values, and demographic factors. Our analysis revealed several important insights:

**Loan Amount and Annual Income**: We found a strong positive correlation between loan amount and annual income. As expected, individuals with higher annual incomes were more likely to secure larger loan amounts. This underscores the importance of a borrower's financial capacity in determining loan approval.

**Loan Term and Approval Rate**: Shorter loan terms, particularly 2 and 4 years, exhibited higher approval rates compared to longer terms. This suggests that borrowers opting for shorter loan durations are perceived as lower risk by lenders.

**Asset Values and Income**: Luxury asset values and bank asset values demonstrated stronger positive correlations with annual income compared to residential and commercial asset values. This suggests that higher-income individuals are more likely to possess luxury items and maintain larger bank assets.

**Categorical Variables**: Our ANOVA tests examined the relationships between categorical variables (no\_of\_dependents, education, self\_employed) and loan amount. While some variables showed minor effects on loan amount, none exhibited statistically significant relationships.

**Machine Learning model**: The explored algorithms include Random Forest Classifier, K-Nearest Neighbors Classifier, Support Vector Classifier, and Logistic Regression. The dataset is prepared by performing exploratory data analysis and feature engineering. Statistics like accuracy score and F1 score are used to judge the execution of each approach. The findings show that the Light GBM had the highest accuracy of 99.2%, followed by Random Forest (98.7%), XGB (98.3%), Decision Tree(97.7%),Support Vector Machine (94.3%) , K Nearest Neighbour (92.2%) and Logistic Regression(91.1%) These results highlight the potential of machine learning algorithms to improve the loan approval process and reduce the risk of loan defaults.

In light of these findings, it's clear that loan approval decisions are driven by a combination of financial factors, including income, loan amount, and loan term. While certain categorical variables appeared to have limited impact on the loan amount, the overall focus should remain on financial indicators. It's important to note that our analysis is subject to certain limitations. The dataset may not capture all relevant factors affecting loan approval, and outliers or data inaccuracies could influence results. Additionally, causation cannot be established through correlation alone.

In conclusion, this analysis sheds light on the intricate relationships between various factors and loan approval outcomes. The insights gained can guide data-driven decision-making in the lending industry.