CHAPTER 1
INTRODUCTION

1.1 OBJECTIVE:

Certainly! In a nutshell, with the increasing demand for loans, banks face challenges in efficiently screening and verifying applicants' eligibility, which can be both time-consuming and prone to errors. Additionally, banks need to minimize the risk of defaults to protect their capital. To address these challenges, a machine learning approach can be implemented.

Machine learning techniques, particularly classification algorithms, can analyze historical data of loan applicants to identify patterns and factors associated with successful loan outcomes. By training these models on past data, they can learn to predict the eligibility of new loan applicants more accurately.

This approach offers several benefits:

1. **Efficiency:** Machine learning models can automate and streamline the loan approval process, reducing the time and effort required for manual screening.

2. **Accuracy:** By analyzing large volumes of data, machine learning models can identify subtle patterns and factors that human analysts might overlook, leading to more accurate predictions of loan eligibility.

3. **Risk Management:** By predicting the likelihood of default, machine learning models help banks assess the risk associated with each loan applicant more effectively, enabling them to make informed decisions to minimize potential losses.

Overall, leveraging machine learning in the loan approval process can enhance efficiency, accuracy, and risk management for banks while improving the experience for loan applicants.

CHAPTER 2
SYSTEM ANALYSIS

2.1 INTRODUCTION

The foundation of banking institutions lies in the provision of loans, which constitute a significant source of revenue primarily generated through interest payments. Despite the rigorous verification and validation processes undertaken by loan corporations, there remains uncertainty regarding the borrower's ability to repay the loan without encountering financial difficulties. Loan prediction emerges as a valuable tool for both bank employees and loan applicants alike. The primary objective of this paper is to expedite the identification of deserving candidates for loans, thereby offering a swift and efficient solution. By leveraging a Loan Prediction System, banks can assign weights to various factors involved in the loan approval process and use these weights to evaluate new loan applications promptly. Additionally, the system establishes a threshold time within which loan approvals are determined, ensuring timely decisions. Furthermore, the system enables prioritized processing of applications, allowing for expedited review of critical cases. Overall, the Loan Prediction System aims to enhance the efficiency of loan processing, benefiting both banks and loan applicants by facilitating informed decision-making and timely responses.

Loans are at the core of banking operations, constituting a primary revenue source driven by interest earnings. Despite the rigorous verification and validation processes employed by loan corporations, there remains uncertainty regarding the borrower's ability to repay the loan without difficulty. In this context, Loan Prediction systems emerge as invaluable tools for both banking professionals and loan applicants alike. The primary objective of this paper is to introduce a rapid and efficient method for identifying deserving loan candidates, thereby providing a distinct advantage to banks. The proposed Loan Prediction System employs sophisticated algorithms to evaluate the significance of various factors in the loan approval process, facilitating swift application evaluation. By considering historical data alongside new information, the system enables timely decision-making on loan approvals. Furthermore, the system sets predefined thresholds, ensuring efficient processing while prioritizing critical cases. Overall, the implementation of a Loan Prediction System promises enhanced operational efficiency and improved outcomes for both banks and loan applicants.

2.2 EXISTING SYSTEM

Borrowers rely on loan applications to qualify for mortgages, and the evolving landscape of technology offers a promising solution. This model has the capability to anticipate outcomes and adapt swiftly to a diverse array of inputs. Notably, this approach not only streamlines processes but also saves the banking industry and its personnel a substantial amount of time.

Conventional Loan Approval Systems often hinge on rule-based frameworks, manually crafted by bank officials to determine loan eligibility based on predetermined criteria. However, these systems lack the sophistication and adaptability offered by modern data science algorithms and techniques. In the current paradigm, loan approval hinges on factors such as a high Cibil score, substantial assets, educational qualifications, and a stable income source. Yet, the absence of data science algorithms in existing systems limits their effectiveness.

Rule-based systems exhibit notable limitations, particularly in dynamic real-world scenarios. They can prove challenging to maintain and update, especially as new loan schemes constantly emerge. Consequently, the reliance on outdated rule-based approaches can hinder agility and responsiveness in the face of evolving borrower needs and market dynamics.

2.3 PROPOSED SYSTEM

The proposed Loan Approval system will utilize machine learning to build a model capable of accurately predicting loan approval decisions for loan seekers. This model will be trained on a dataset containing historical loan details data, encompassing various factors such as education level, Cibil score, annual income, additional income, requested loan amount, loan term, residential and commercial asset values, and employment status. Once trained, the model will have the ability to predict the probability of loan approval for individual loan seekers based on their characteristics and financial details. This prediction will align with the specific loan policies of the bank or lending institution. In summary, the proposed system aims to leverage machine learning to streamline the loan approval process by providing accurate predictions of loan approval outcomes based on historical data and borrower profiles.

- The proposed system is expected to be more accurate in detecting fraudulent transactions than rule-based systems, as it is able to learn from data and adapt to changing fraud patterns.
- The proposed system is expected to produce fewer false positives than rule-based systems, as it is able to consider a wider range of factors when evaluating a transaction.
- The proposed system is scalable to handle large volumes of transaction data.
- The proposed system can be customized to meet the specific needs of a financial institution or payment processor.

In the Loan Approval system project, we employ Machine Learning algorithms to train multiple models and predict the loan approval outcome. To assess the performance of these models, we evaluate them using various evaluation metrics such as:

1. **Test Accuracy:** Measures the proportion of correctly classified loan applications in the test dataset.

2. **False Positives and False Negatives:** Quantify the instances where the model incorrectly predicts loan approval or rejection.

3. **Bias and Variance:** Indicate the model's ability to generalize to unseen data (low bias and variance) or overfit to the training data (high variance).

4. **F1 Score:** Harmonic mean of precision and recall, providing a balance between the two metrics.

5. **Recall Score:** Measures the proportion of actual positives correctly identified by the model.

6. **Precision Value:** Measures the proportion of predicted positives that are actually positive.

7. **Cohen Kappa Score:** Evaluates the agreement between the model's predictions and the actual outcomes, accounting for chance agreement.

8. **ROC AUC Score:** Area under the Receiver Operating Characteristic curve, indicating the model's ability to distinguish between positive and negative classes across different thresholds.

These evaluation metrics collectively provide insights into the performance and reliability of the Loan Approval system's machine learning models, enabling stakeholders to make informed decisions and improvements as needed.

2.3.1 BENEFITS PROPOSED SYSTEM

The proposed system offers a number of benefits over existing systems, including:

- **Improved accuracy:** The proposed system is expected to be more accurate in detecting fraudulent transactions than rule-based systems, as it is able to learn from data and adapt to changing fraud patterns.

- **Reduced false positives:** The proposed system is expected to produce fewer false positives than rule-based systems, as it is able to consider a wider range of factors when evaluating a transaction.

- **Scalability:** The proposed system is scalable to handle large volumes of transaction data.

- **Flexibility:** The proposed system can be customized to meet the specific needs of a financial institution or payment processor.

2.4 FEASIBILITY STUDY

A feasibility study for the proposed Loan Approval system would involve assessing various aspects to determine the viability and practicality of implementing the system. Here's an overview of what the feasibility study might include:

1. Technical Feasibility: Evaluate whether the technology required to develop and implement the system is readily available and feasible. Consider factors such as the availability of data, compatibility with existing systems, and the expertise required to develop and maintain the system.

2. Financial Feasibility: Assess the financial implications of developing and implementing the system. This involves estimating the costs associated with software development, hardware requirements, training, maintenance, and potential savings or revenue generated by the system.

3. Operational Feasibility: Examine whether the proposed system aligns with the operational requirements and processes of the organization. Consider factors such as the ease of integration with existing workflows, potential disruptions to operations during implementation, and the system's usability for end-users.

4. Legal and Regulatory Feasibility: Evaluate the legal and regulatory requirements that may impact the implementation of the system. Ensure compliance with data privacy regulations, financial industry standards, and any other relevant laws or regulations.

5. Schedule Feasibility: Determine whether the proposed timeline for developing and implementing the system is realistic and achievable. Consider factors such as resource availability, project complexity, and potential dependencies on other projects or initiatives.

6. Risk Assessment: Identify potential risks and challenges associated with the implementation of the system. Develop strategies to mitigate these risks and ensure the successful execution of the project.

By conducting a comprehensive feasibility study, organizations can make informed decisions about whether to proceed with the development and implementation of the Loan Approval system. This study provides valuable insights into the technical, financial, operational, legal, and schedule-related aspects of the project, enabling stakeholders to assess the feasibility and potential impact of the proposed system accurately.

CHAPTER 3
SYSTEM SPECIFICATION

3.1 SOFTWARE REQUIREMENTS

Machine learning project, you'll need a solid setup that encompasses various components, including hardware, software, programming languages, frameworks, development tools, and libraries.

Operating System: Most machine learning frameworks are compatible with widely used operating systems such as Windows, macOS, and Linux. Therefore, you have the flexibility to choose the operating system that best suits your preferences and requirements.

Python: Python stands out as the predominant programming language in the field of machine learning. It offers simplicity, readability, and a rich ecosystem of libraries and frameworks tailored specifically for data science and machine learning tasks. Most machine learning frameworks are built using Python, making it the language of choice for developing and deploying machine learning models.

Machine Learning Frameworks: There is a plethora of machine learning frameworks available, each offering unique features and capabilities. Some of the most popular frameworks include TensorFlow, PyTorch, Keras, and scikit-learn. These frameworks provide comprehensive tools, APIs, and algorithms for building, training, and deploying machine learning models across a wide range of applications.

Development Tools: Various integrated development environments (IDEs) are commonly used for machine learning development, providing a conducive environment for coding, experimentation, and collaboration. Popular choices include PyCharm, Jupyter Notebook, and Visual Studio Code, each offering unique features and functionalities to streamline the development process.

Libraries: Libraries play a crucial role in data manipulation, analysis, and visualization, facilitating various tasks throughout the machine learning workflow. Fundamental libraries such as NumPy, Pandas, and Matplotlib are indispensable for handling numerical data, performing data analysis, and visualizing results effectively.

Hardware Requirements: To handle the computational demands of machine learning tasks, it's essential to have a computer with adequate hardware specifications. This typically includes a powerful CPU and GPU, a minimum of 8GB of RAM (though more is often beneficial), and sufficient storage capacity to accommodate datasets and model files.

In summary, getting started with a machine learning project entails setting up a robust environment that encompasses the right hardware, software, programming languages, frameworks, development tools, and libraries. By leveraging these components effectively, you can embark on your machine learning journey with confidence and efficiency.

3.2 HARDWARE REQUIREMENTS

To effectively train and deploy machine learning models, you'll need a robust hardware setup that includes:

1. **CPU:** A modern multicore processor, such as an Intel Core i5 or i7, is recommended. These processors provide the necessary computational power to handle the training and deployment of machine learning models efficiently.

2. **RAM:** A minimum of 8GB of RAM is recommended to ensure smooth operation, especially when working with medium-sized datasets. However, for larger datasets or more complex models, additional RAM may be required to handle the computational load effectively.

3. **GPU:** A powerful graphics processing unit (GPU) can significantly accelerate the training and inference of deep learning models, especially for computationally intensive tasks. Nvidia GPUs, such as those in the GeForce or Quadro series, are commonly used for machine learning tasks due to their high performance and extensive support for deep learning frameworks.

4. **Storage:** Machine learning projects often involve working with large datasets, so ample storage space is essential. Solid-state drives (SSDs) are recommended for fast access to data, which can speed up data processing and model training times.

By ensuring that your hardware setup meets these requirements, you can effectively tackle machine learning tasks and optimize the performance of your models.

CHAPTER 4
SOFTWARE DESCRIPTION

4.1 COMPONENTS

A software description for a machine learning project typically includes details about the programming languages, frameworks, libraries, and tools used in the development process. Here's an overview of the components commonly found in a software description for a machine learning project:

1. **Programming Languages:** Specify the primary programming language used for developing the machine learning models. Python is the most commonly used language due to its simplicity, extensive libraries, and wide adoption in the machine learning community.

2. **Machine Learning Frameworks:** Describe the machine learning frameworks utilized for building and training models. This may include popular frameworks such as TensorFlow, PyTorch, Keras, scikit-learn, or others, depending on the specific requirements of the project.

3. **Development Tools:** List the integrated development environments (IDEs) or code editors used for coding and experimentation. Common tools include PyCharm, Jupyter Notebook, Visual Studio Code, or others that provide features for code editing, debugging, and visualization.

4. **Libraries:** Specify the libraries and packages used for data manipulation, analysis, and visualization. This may include fundamental libraries such as NumPy, Pandas, Matplotlib, as well as domain-specific libraries for tasks like natural language processing (NLTK), computer vision (OpenCV), or time series analysis (statsmodels).

5. **Version Control:** Mention the version control system used for managing code changes and collaboration among team members. Git is a popular choice, often combined with platforms like GitHub or GitLab for hosting repositories and managing project workflows.

6. **Documentation Tools:** Describe the tools and platforms used for documenting the project, including code documentation, project specifications, and user guides. Common documentation tools include Sphinx, MkDocs, Read the Docs, or simply markdown files within the project repository.

7. **Testing and Evaluation:** Outline the methodologies and tools used for testing and evaluating the performance of machine learning models. This may include metrics such as accuracy, precision, recall, F1 score, ROC AUC score, as well as techniques for cross-validation and hyperparameter tuning.

8. **Deployment:** Discuss the strategies and tools used for deploying machine learning models into production environments. This may involve containerization technologies like Docker, cloud platforms such as AWS, Azure, or Google Cloud Platform, or specialized deployment frameworks like TensorFlow Serving or FastAPI..

CHAPTER 5
PROJECT DESCRIPTION

5.1 PROBLEM DEFINITION

The problem definition in a machine learning project outlines the specific task or objective that the project aims to address. It serves as the foundation for the entire project, guiding the development of machine learning models and determining the success criteria. Here's how you might define the problem in a machine learning project:

1. **Problem Statement:** Clearly state the problem that the machine learning project seeks to solve. This could be a classification task, regression task, clustering task, or another type of problem. For example, "Predicting whether a customer will churn or not" or "Identifying fraudulent transactions in financial data."

2. **Objective:** Define the project's primary objective in terms of the desired outcome. This could be maximizing accuracy, minimizing error, optimizing a specific metric, or achieving a certain level of performance. For example, "Maximize the accuracy of predicting customer churn to reduce customer attrition rates by 10%."

3. **Scope:** Describe the scope of the problem, including any limitations or constraints that may affect the project. This could include constraints related to data availability, computational resources, time, or budget. For example, "The model will only consider historical transaction data from the past 12 months."

4. **Data:** Specify the data sources and types of data that will be used to solve the problem. This includes both the input features (independent variables) and the target variable (dependent variable) that the model aims to predict. For example, "The dataset includes customer demographics, transaction history, and churn status."

5. **Evaluation Metric:** Define the evaluation metric that will be used to assess the performance of the machine learning models. This could be accuracy, precision, recall, F1 score, ROC AUC score, or another appropriate metric based on the nature of the problem. For example, "The model will be evaluated based on its F1 score, with a target score of 0.8."

6. **Success Criteria:** Establish the criteria that define the successful completion of the project. This could be achieving a certain level of performance on the evaluation metric, meeting specific business objectives, or surpassing the performance of baseline models or benchmarks. For example, "The project will be considered successful if the model achieves an F1 score of 0.8 or higher on the test dataset."

By clearly defining the problem, objectives, scope, data, evaluation metrics, and success criteria, stakeholders can align their expectations and ensure that the machine learning project addresses the intended goals effectively.

5.2 SOFTWARE DEVELOPMENT LIFE CYCLE

5.2.1 SDLC (Software Development Cycle):

The Software Development Life Cycle (SDLC) is a structured approach to building software that ensures the quality, reliability, and correctness of the final product. It encompasses a series of phases, from initial planning and requirements gathering to deployment and maintenance. The primary objectives of the SDLC process are to deliver high-quality software that meets customer expectations, adhere to predefined timelines, and stay within budget constraints. By following a systematic SDLC process, organizations can mitigate risks, improve project management, enhance collaboration among team members, and ultimately deliver successful software solutions that satisfy stakeholders' needs.

5.2.2 SDLC Phases:

The entire SDLC process is divided into the following stages

- Phase 1: Requirement collection and analysis
- Phase 2: Feasibility study
- Phase 3: Design
- Phase 4: Coding
- Phase 5: Testing
- Phase 6: Installation/Deployment
- Phase 7: Maintenance

5.3 PLATFORM KNOWLEDGE

5.3.1 What is data science?

Data science is the study of data to extract meaningful insights for business. It is a multidisciplinary approach that combines principles and practices from the fields of mathematics, statistics, artificial intelligence and computer engineering to analyze large amounts of data. This analysis helps data scientists to ask and answer questions like what happened, why it happened, what will happen, and what can be done with the results.

5.3.2 IMPORTANCE OF DATA SCIENCE

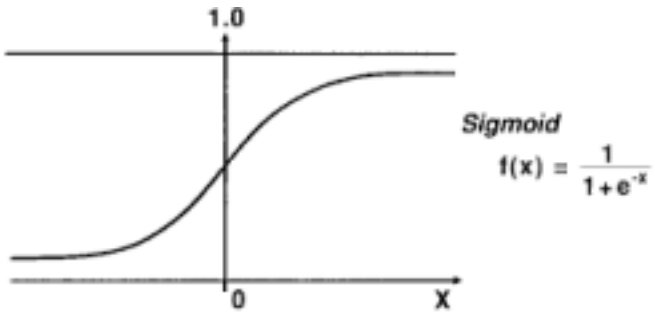
Solves real problem using data.Modern organizations have lots of data.Online systems and payment portals capture more data in the fields of e-commerce, medicine, finance, and every other aspect of human life. We can process the information and predict the results

5.4 MACHINE LEARNING

5.4.1 Machine Learning Algorithms

1. Logistic Regression:

It is one of the most simplest and popular machine learning model.



This model takes the input values as x and gives the output values as f(x) i.e. 0 or 1. If I need to built a machine learning model then each data point of independent variable will be x (i.e. sum of x1 * w1 + x2 * w2 . . . .so on) and this will give a value that is between 0 to 1. If I consider that 0.50 as deciding value or threshold. Then any result above 0.5 would be taken as 1 and below that as 0.

2. Decision Tree:

The first step is to split the labelled data into train and test data.The tree begins with the root node which consists of the entire training data.The best attribute is found using the Attribute selection measure.Entropy: (chances of being incorrect).Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. When the data is highly impure, or highly pure, the entropy is 0. Entropy can be calculated as:

Entropy(s)= -P(Yes) * log2 P(Yes) - P(No) * log2 P(No)

Entropy(S) = 1 when P(Yes) = P(No) = 0.5.

Entropy(S) = 0 when P(Yes) = 1 or 0

Information Gain:

Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute. It calculates how much information a feature provides us with a class.According to the value of information gain, we split the node and build the decision tree. A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

Information Gain = Entropy(S)- [(Weighted Avg) *Entropy (each feature)]

Select the feature with maximum information gain and divide the dataset. Recursively make new decision trees using the subsets of the dataset created in above step.Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

3. Random Forest:

A random forest algorithm is a supervised learning algorithm that can be used for both classification and regression tasks. It is an ensemble learning method, which means that it combines the predictions of multiple individual models to produce a more accurate prediction. Random forest algorithms work by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.
Random forest algorithms are more robust to overfitting than individual decision trees because they average the predictions of multiple trees. This is because each tree in the forest is trained on a different bootstrap sample of the training data and a different random subset of features.

5.5 LITERATURE SURVEY

Paper 1: Loan Credibility Prediction System using Data Mining Techniques Abstract: As we all know that now-a-days there's a rising in banking sector, ensuing several folks applying for bank loans. Looking for the mortal to whom the loan is approved could be a troublesome method. Data processing techniques are getting very popular today attributable to the wide handiness of giant amount and therefore the want for remodeling such data into knowledge. Techniques of data mining enforced in numerous domains like retail business, telecommunication business, biological information analysis, etc. During this paper, we had a tendency to plan a model that predicts loan approval/rejection of associate degree mortal by taking help of data processing techniques. This will be done by training the model with the info of the previous records of the folks applied for loan.

Paper 2: An Approach for Prediction of Loan Approval using Machine Learning Algorithm Abstract: Banks have several commodities to sell however, main supply of financial gain of any banks is on its credit line. So, they'll earn from interest of these loans that they credit. A bank's profit or a loss depends to an oversized extent on loans

i.e., whether or not the purchaser's area unit return the loan or defaulting. By predicting the loan defaulters, the bank will scale back its Non-Performing Assets.

This makes the study of this development important. Previous analysis during this era has shown that there are a large number of strategies to check the matter of dominant loan default. However, because the right prediction is important for the maximization of profits, it's essential to check the character of the various strategies and their comparison.

So, it becomes necessary in this predictive analytic to check the matter of predicting loan defaulters: The logistic regression model. The information is collected from Kaggle for learning and prediction. Logistic Regression models are performed and therefore the totally different measures of performances are computed. The models are compared on the idea of the performance measures like sensitivity and specificity.

The ultimate results have shown that the model turn out totally different results. Model is marginally higher as a result of it includes variables (personal attributes of client like age, purpose, credit history, credit quantity, credit length, etc.) aside from bank account info (which shows wealth of a customer) that ought to be taken under consideration to calculate the likelihood of getting loan properly.

Therefore, by employing a logistic regression approach, the proper customers to be targeted for granting loan is simply detected by evaluating their chances. The model concludes that a bank shouldn't solely target the main clients for granting loan however it ought to assess the opposite attributes of a customer similarly that play a really necessary half in credit granting choices and predicting the loan defaulters.

5.6 METHODOLOGY

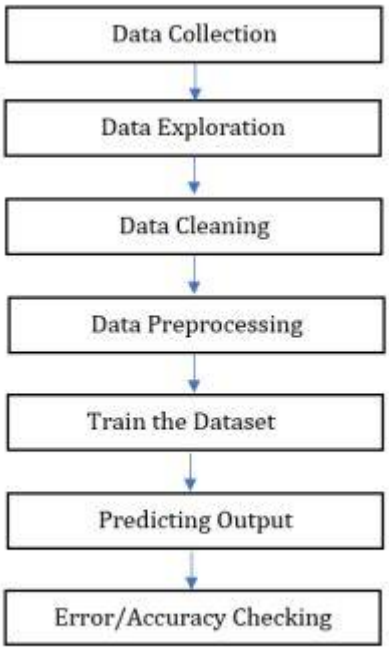


Fig -1: Proposed System

The given problem is a supervised classification problem as we are finding whether the person is reliable for a loan or not that is Yes or No. This can be solved with any of the algorithms listed below: I Decision tree ii. Logistic regression iii. Random Forest These algorithms are some of the few algorithms that can be used to solve the problem.

5.6.1 Collection of Data:

The input dataset contains information about customers who have applied for loan approval. It is stored as a CSV file, which can be easily read into a Python environment using the `read_csv ()'. method from the Pandas library. Therefore, it's necessary to import Pandas into the Python environment to handle and manipulate the dataset effectively.

Some of the features included in the dataset are:

1. Loan ID: Unique identifier for each loan application.

2. Married: Indicates whether the applicant is married or not.

3. Gender: Gender of the applicant.

4. Dependents: Number of dependents (e.g., children, family members) of the applicant.

5. Education: Level of education attained by the applicant.

6. Self-employed: Indicates whether the applicant is self-employed or not.

7. Applicant Income: Income of the applicant.

8. Loan Amount Term: Term or duration of the loan in months.

9. Loan Amount: Amount of loan requested by the applicant.

10. Credit History: Credit history of the applicant.

11. Loan Status: Indicates whether the loan was approved or not.

12. Property Area: Location or area where the property is located.

These features provide essential information about the applicants and their loan applications, which will be used for analysis, modeling, and prediction purposes. By leveraging this dataset, machine learning models can be trained to predict loan approval decisions based on various applicant attributes.

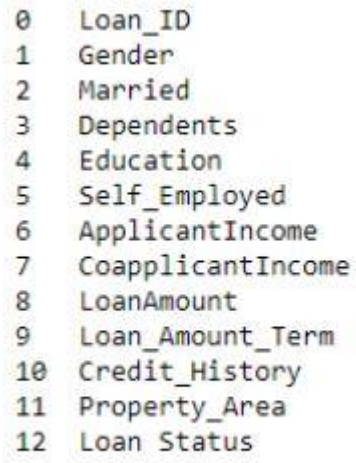


Fig -2: Attributes Of Dataset

5.6.2 Data Exploration:

Importing libraries and packages in Python is essential for performing various tasks, such as data exploration, manipulation, visualization, and modeling. Some commonly used libraries for data analysis and machine learning tasks include Pandas, NumPy, Matplotlib, and Seaborn.

Once the necessary libraries are imported, the next step typically involves loading the dataset into the Python environment. This can be achieved using functions provided by the Pandas library, such as `read_csv ()'. for reading CSV files.

After loading the dataset, it's common practice to inspect the data to get an initial understanding of its structure and contents. This can be done by using methods like `head()` or `sample()` in Pandas to view the top rows or a random sample of the dataset, respectively.

Furthermore, it's important to check for missing or null values in the dataset, as these can impact the quality of the analysis and modeling process. Pandas provides functions like `isnull()` and `info()` to identify null values and get a summary of the dataset, including information about the number of non-null entries in each column.

By importing the necessary libraries, loading the dataset, and checking for null values, data analysts and data scientists can start exploring and analyzing the data effectively to derive insights and make informed decisions.

5.6.3 Data Cleaning:

When dealing with null values in a dataset, it's essential to handle them appropriately to ensure the reliability and accuracy of any analysis or modeling performed on the data. One common approach is to remove or drop the rows or columns containing null values.

The `dropna()` function in Pandas is specifically designed for this purpose. It allows you to drop rows or columns from a DataFrame that contain null values. By using this function to remove null values from the dataset, you ensure that only complete and valid data remains for analysis.

Removing null values is important because they can adversely affect statistical analyses, machine learning algorithms, and visualizations. Null values can skew results, introduce biases, or cause errors in computations if not properly handled.

After dropping null values, the dataset becomes more reliable and suitable for further exploration, analysis, and modeling. However, it's important to consider the potential impact of removing data on the overall dataset size and representativeness, and to make informed decisions based on the specific context and objectives of the analysis.

```
df.isnull().sum()
```

```
Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
Loan_Status          0
dtype: int64
```

Fig -3: Data Cleaning

5.6.4 Data Preprocessing:

In machine learning, it's essential to convert categorical variables, such as those with article (text) types, into numerical format because most machine learning algorithms require numerical inputs. This process is called encoding or converting categorical variables to numerical format. One common approach is to use techniques like one-hot encoding or label encoding to transform categorical variables into numerical representations.

Standardizing the data is another important preprocessing step in machine learning. Standardization involves scaling the features to have a mean of 0 and a standard deviation of 1. This ensures that all features are on a similar scale, which can help improve the performance of machine learning models, particularly those sensitive to feature scales, such as linear models and distance-based algorithms.

By converting categorical variables to numerical format and standardizing the data, we ensure that the input features are suitable for use with machine learning algorithms, enabling them to effectively learn patterns and make predictions from the data.

5.6.5 Data Visualizations:

When dealing with categorical data in a machine learning project, converting it into numerical format is often necessary as many machine learning algorithms expect numerical inputs. This transformation allows the algorithms to process and analyze the data effectively.

There are a few common methods for converting categorical data into numerical data:

1. **Label Encoding:** This method assigns a unique integer to each category within a categorical variable. Each category is mapped to a numerical value, which allows the algorithm to understand and process the data. However, care should be taken as label encoding may introduce an ordinal relationship between categories that may not exist.

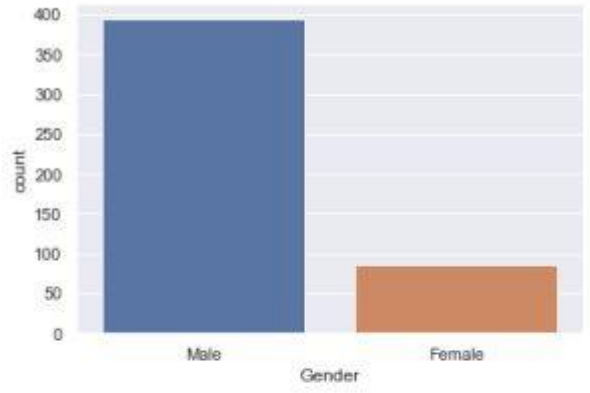
2. **One-Hot Encoding:** This technique creates binary columns for each category in the original variable. Each category is represented by a binary column where a value of 1 indicates the presence of the category and 0 indicates its absence. One-hot encoding is useful for categorical variables with no ordinal relationship among categories.

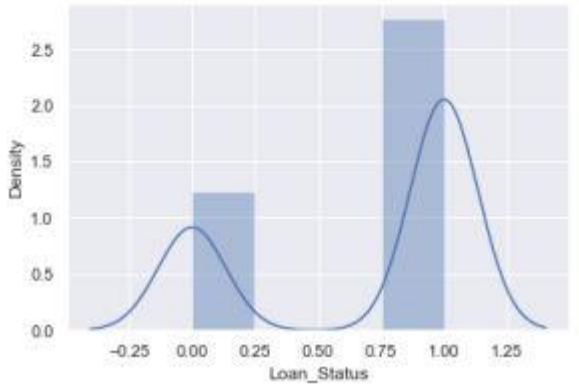
3. **Ordinal Encoding:** If the categorical variable has an inherent order or hierarchy, ordinal encoding can be used. In this method, each category is assigned a numerical value based on its order or rank.

4. **Hash Encoding:** This method hashes the categorical variables into numerical values. It's particularly useful when dealing with a large number of unique categories.

After converting categorical data into numerical format using one of these methods, the data becomes suitable for analysis and modeling with machine learning algorithms. This transformation allows the algorithms to effectively learn patterns and make predictions from the data.

```
sns.countplot(x='Gender',data=df)
<AxesSubplot:xlabel='Gender', ylabel='count'>
```





5.6.6 Data Modelling:

Logistic regression is a statistical method used for modeling categorical outcomes, especially when the dependent variable is binary. It is commonly employed in classification tasks where the outcome is either a "yes" or "no", "0" or "1", or some other distinct value. Instead of predicting the exact value of the outcome, logistic regression provides probabilistic values between 0 and 1, representing the likelihood of belonging to a certain category.

The logistic regression model employs an "S"-shaped logistic function, also known as the sigmoid function, to predict the probability of the outcome being in a particular category. This function allows the model to capture nonlinear relationships between the independent variables and the probability of the outcome.

By fitting the logistic function to the data, logistic regression can effectively classify observations into different categories based on their features. It is a valuable tool for understanding the relationship between independent variables and the probability of a particular outcome, making it useful in various fields such as healthcare, finance, and marketing.

Overall, logistic regression is a versatile and widely used technique for binary classification tasks, providing interpretable results and insights into the factors influencing the outcome.

$$\log(1/1-y) = b0+b1x1+b2x2+b3x3+.....+bnx$$

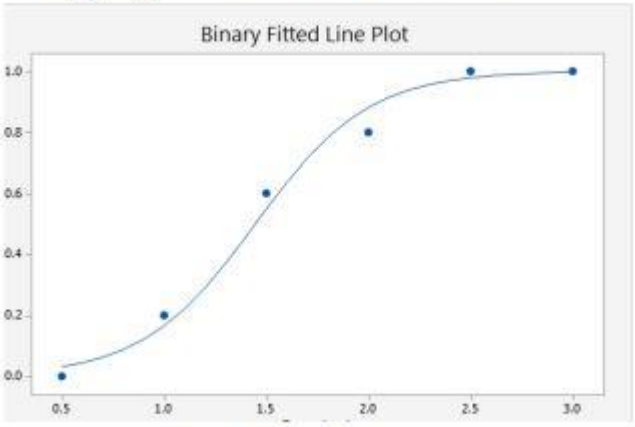


Chart -1: Logistic Function

The Logistic Regression was employed to fit the test data set and prediction result was displayed Successfully.

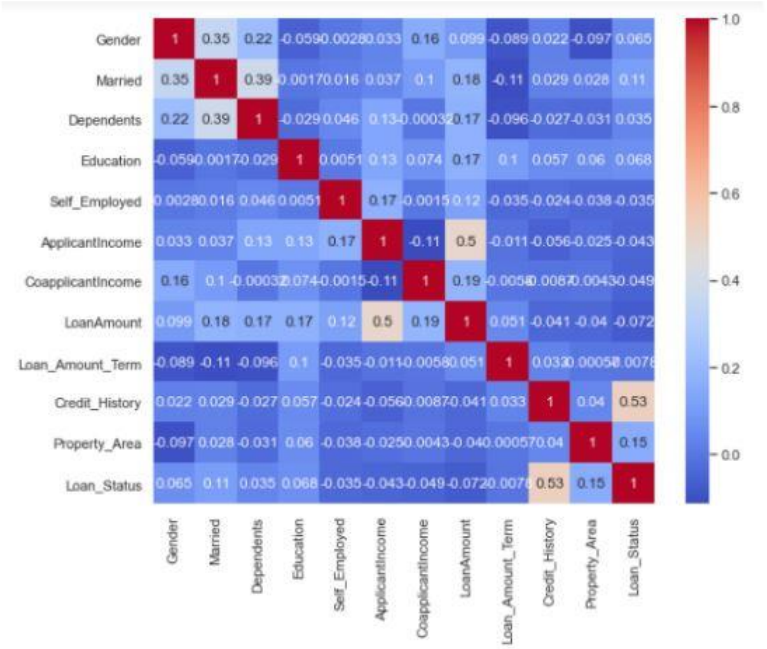
5.7 EXPERIMENTAL ANALYSIS

Label encoding and dummy encoding are techniques used to convert categorical variables into numerical format, which is necessary for many machine learning algorithms. Here's an explanation of both techniques:

1. **Label Encoding:** Label encoding is used when the categorical variable has an inherent ordinal relationship, meaning there is a natural order or hierarchy among the categories. In label encoding, each category is assigned a unique integer value based on its position or order in the variable's domain. For example, if the variable "Dependents" has categories like "0", "1", "2", and "3+", label encoding would assign them numerical values like 0, 1, 2, and 3, respectively. Label encoding is useful when there is a clear order among the categories, such as in the case of ordinal variables like "Dependents".

2. **Dummy Encoding (One-Hot Encoding):** Dummy encoding, also known as one-hot encoding, is used when the categorical variable does not have an ordinal relationship among its categories. In dummy encoding, each category is represented by a binary column, where a value of 1 indicates the presence of the category and 0 indicates its absence. For example, if the variable "Property_Area" has categories like "Urban", "Rural", and "Semiurban", dummy encoding would create three binary columns: "Property_Area_Urban", "Property_Area_Rural", and "Property_Area_Semiurban". Each column would have a value of 1 for observations belonging to the respective category and 0 for observations not belonging to that category. Dummy encoding is useful for categorical variables with no inherent order among the categories.

By applying label encoding to variables like "Dependents" and "Property_Area" and dummy encoding to the remaining categorical variables, we convert all categorical variables into numerical format, making them suitable for input into machine learning algorithms. This preprocessing step ensures that the model can effectively learn from the data and make accurate predictions regarding loan approval based on the applicant's information.



The train-test split is a fundamental step in supervised machine learning, where the dataset is divided into two subsets: the training dataset and the test dataset.

1. **Train Dataset:** The training dataset is used to train the machine learning model. It consists of a subset of the original dataset and contains both independent features and their corresponding dependent labels. The model learns patterns and relationships from this data to make predictions.

2. **Test Dataset:** The test dataset is used to evaluate the performance of the trained machine learning model. It is another subset of the original dataset that the model has not seen during training. The model makes predictions on this dataset, and its performance metrics are calculated based on how well these predictions match the actual labels.

Logistic Function:

The logistic function, also known as the sigmoid function, is used in logistic regression to model the probability that a given input belongs to a certain category. It takes any real-valued number and maps it to a value between 0 and 1, which can be interpreted as the probability of the input belonging to the positive class.

Performance Metrics:

- **Accuracy:** Accuracy measures the proportion of correctly predicted instances (both true positives and true negatives) out of the total number of instances. It provides an overall assessment of the model's correctness.

- **Precision:** Precision measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives). It indicates how precise the model is when it predicts positive instances.

- **Recall:** Recall measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives + false negatives). It indicates the model's ability to identify all relevant instances.

- **F1 Score:** The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall, giving equal weight to both metrics. It is particularly useful when dealing with imbalanced datasets.

These performance metrics help assess the effectiveness and reliability of the trained machine learning model in making predictions. They provide insights into different aspects of the model's performance, such as its accuracy, precision, and ability to correctly identify positive instances.

5.8 CONCLUSIONS

In our model, we utilized a logistic regression algorithm to predict whether a loan application would be approved or not based on various input variables. Logistic regression is a statistical technique commonly used for binary classification tasks, making it suitable for predicting outcomes such as loan approval (1) or rejection (0).

When the program takes input data, it processes it through the logistic regression model, which then produces an output in binary form, either 0 or 1. If the output is 1, it indicates that the loan is approved, and if the output is 0, it indicates that the loan is not approved.

The implementation of this loan credibility prediction system assists organizations in making informed decisions regarding whether to approve or reject loan requests from customers. By leveraging machine learning techniques like logistic regression, organizations can automate and streamline the loan approval process, leading to more efficient and effective decision-making.

While logistic regression is a commonly used and effective technique for binary classification tasks, there is always room for improvement. Future work could involve exploring and incorporating alternative techniques or algorithms that may outperform logistic regression in terms of predictive accuracy and performance. These techniques could be tested and evaluated to determine their suitability for the specific domain and dataset at hand, ultimately enhancing the capabilities of the loan prediction system.

CHAPTER 6
DATA SCIENCE

6.1 WHAT IS DATA SCIENCE

Data science has emerged as a vital field in today's data-driven world, playing a pivotal role in unlocking valuable insights from vast and complex datasets. It encompasses a diverse set of techniques, methodologies, and tools aimed at extracting actionable knowledge from data to inform decision-making and drive innovation across industries. At its core, data science involves the systematic analysis of data to uncover patterns, trends, and relationships that may not be readily apparent. Through the application of statistical methods, machine learning algorithms, and data visualization techniques, data scientists are able to derive meaningful insights that can lead to improved business strategies, product enhancements, and process optimizations.

Data science encompasses a wide range of applications, spanning various domains such as healthcare, finance, marketing, and beyond. In healthcare, for example, data science is used to analyze patient records, medical images, and genetic data to improve diagnostics, treatment outcomes, and personalized medicine. In finance, data science helps detect fraudulent transactions, predict market trends, and optimize investment strategies. Similarly, in marketing, data science is employed to segment customers, personalize marketing campaigns, and measure campaign effectiveness, driving higher conversion rates and customer engagement.

Data science also plays a crucial role in addressing societal challenges and driving positive social impact. For instance, it is used in disaster response and humanitarian efforts to analyze real-time data from social media, sensors, and satellite imagery to assess the extent of damage, coordinate rescue efforts, and allocate resources efficiently. Furthermore, data science is instrumental in tackling pressing issues such as climate

change, poverty alleviation, and public health by analyzing large-scale datasets to identify trends, inform policy decisions, and develop targeted interventions.

Data science is a dynamic and interdisciplinary field with widespread applications across industries and sectors. By harnessing the power of data, data scientists are able to uncover actionable insights that drive innovation, improve decision-making, and address complex challenges facing society. As the volume and complexity of data continue to grow, the role of data science will become increasingly vital in unlocking the full potential of data to create value and drive positive change.

6.1.1 Key Components of Data Science:

1. **Statistics and Mathematics**: Data scientists utilize statistical methods and mathematical algorithms to analyze data, identify patterns, and make predictions.

2. **Programming and Computer Science**:Proficiency in programming languages such as **Python, R, and SQL** is essential for data manipulation, analysis, and visualization. Additionally, knowledge of computer science principles facilitates the development of algorithms and the implementation of machine learning models.

3. **Machine Learning and Artificial Intelligence**: Data science encompasses machine learning techniques that enable computers to learn from data and make predictions or decisions without explicit programming. This includes supervised learning, unsupervised learning, and reinforcement learning algorithms.

4. **Data Visualization**: Data scientists use data visualization tools and techniques to present complex datasets in a visual format, making it easier to interpret and communicate insights to stakeholders.

5. **Domain Knowledge**: Understanding the domain or industry context is crucial for effective data analysis and interpretation. Data scientists often collaborate with domain experts to ensure that insights derived from data align with business objectives.

Overall, data science plays a pivotal role in extracting actionable insights from data, driving innovation, optimizing processes, and informing strategic decision-making in various fields such as **healthcare, finance, marketing, and beyond.**

6.2 WHY DATA SCIENCE

Data science plays a crucial role in today's data-driven world for several reasons:

1. **Unlocking Insights**: With the exponential growth of data in various forms, ranging from structured databases to unstructured text and multimedia, data science provides the tools and techniques to extract valuable insights from this wealth of information. By analyzing large and complex datasets, data science uncovers patterns, trends, and correlations that can inform decision-making and drive innovation.

2. **Informed Decision-Making**: In business and organizations across industries, data science enables evidence-based decision-making. By leveraging data analytics and predictive modeling, decision-makers can better understand customer behavior, market trends, and operational inefficiencies, leading to more informed strategies and improved outcomes.

3. **Optimizing Processes**: Data science helps optimize processes and operations by identifying inefficiencies, automating repetitive tasks, and streamlining workflows. By analyzing data from various sources, organizations can identify bottlenecks, streamline supply chains, and improve resource allocation, leading to cost savings and operational efficiencies.

4. **Predictive Capabilities**: One of the key strengths of data science is its ability to make predictions based on historical data. Predictive modeling techniques enable organizations to forecast future trends, anticipate customer needs, and mitigate risks. This predictive capability is invaluable in fields such as finance, healthcare, and marketing, where accurate forecasts can drive strategic planning and decision-making.

5. **Driving Innovation**: Data science fuels innovation by uncovering new insights and opportunities that may not be apparent through traditional analysis methods. By applying advanced analytics, machine learning, and artificial intelligence techniques, data scientists can develop innovative products, services, and solutions that address complex challenges and meet evolving customer needs.

6. **Competitive Advantage**: In today's competitive landscape, organizations that harness the power of data science gain a significant competitive advantage. By leveraging data-driven insights, organizations can differentiate themselves in the market, identify new revenue streams, and stay ahead of competitors by anticipating market trends and customer preferences.

Overall, data science is essential because it enables organizations to harness the power of data to drive strategic decision-making, optimize operations, foster innovation, and gain a competitive edge in today's data-driven economy.

Data science is instrumental in today's data-driven world due to its ability to extract valuable insights and knowledge from large and complex datasets. By leveraging statistical analysis, machine learning algorithms, and data visualization techniques, data science enables organizations to uncover patterns, trends, and correlations within their data that may not be immediately apparent. These insights provide a foundation for evidence-based decision-making, allowing businesses to understand customer behavior, optimize operations, and drive innovation.

Moreover, data science offers predictive capabilities that enable organizations to anticipate future trends and outcomes. Through predictive modeling and forecasting techniques, data scientists can analyze historical data to make informed predictions about future events, such as customer preferences, market trends, and business performance. By leveraging these predictive insights, organizations can proactively adapt their strategies, mitigate risks, and capitalize on emerging opportunities, ultimately gaining a competitive edge in their respective industries. Overall, the use of data science empowers organizations to harness the full potential of their data, driving strategic decision-making, fostering innovation, and achieving sustainable growth.

6.3 FIELDS THAT USES DATA SCIENCE

Data science finds applications across various fields and industries, revolutionizing the way organizations operate and make decisions. Some of the key fields that extensively utilize data science include:

1. **Healthcare**: Data science plays a crucial role in healthcare by analyzing patient records, medical imaging data, genomic data, and clinical trial results to improve diagnostics, treatment outcomes, and personalized medicine. It also aids in predictive analytics for disease outbreaks, patient readmissions, and healthcare resource allocation.

2. **Finance**: In the finance industry, data science is used for fraud detection, risk assessment, algorithmic trading, credit scoring, and customer segmentation. It helps financial institutions identify fraudulent transactions, predict market trends, optimize investment portfolios, and personalize financial services for customers.

3. **Marketing**: Data science drives marketing strategies by analyzing customer behavior, preferences, and purchasing patterns. It enables marketers to segment customers, personalize marketing campaigns, optimize pricing strategies, and measure the effectiveness of marketing initiatives through techniques such as customer churn prediction, recommendation systems, and sentiment analysis.

4. **Retail**: Retailers leverage data science for demand forecasting, inventory management, pricing optimization, and customer analytics. By analyzing sales data, foot traffic patterns, and customer demographics, retailers can enhance customer experiences, optimize product assortments, and maximize revenue through targeted promotions and discounts.

5. **Manufacturing**: Data science is employed in manufacturing for predictive maintenance, quality control, supply chain optimization, and process automation. By analyzing sensor data from machinery and equipment, manufacturers can predict equipment failures, optimize production schedules, minimize downtime, and improve overall operational efficiency.

6. **Telecommunications**: In the telecommunications industry, data science is used for network optimization, customer churn prediction, and fraud detection. It helps telecom companies analyze call detail records, network traffic data, and customer interactions to improve network performance, enhance customer retention, and prevent fraudulent activities.

7. **Energy and Utilities**: Data science enables energy and utility companies to optimize energy production, distribution, and consumption. It is used for predictive maintenance of infrastructure, demand forecasting, grid optimization, and renewable energy integration, contributing to sustainable energy practices and cost savings.

8. **Transportation and Logistics**: Data science is essential in transportation and logistics for route optimization, fleet management, demand forecasting, and supply chain visibility. By analyzing data from GPS systems, sensors, and logistics platforms, companies can reduce transportation costs, improve delivery times, and enhance overall logistics efficiency.

These are just a few examples of the diverse fields that rely on data science to drive innovation, improve decision-making, and achieve business objectives in today's data-driven world.

6.4 IMPORTANCE OF DATA SCIENCE

Data science has emerged as a critical component of modern business strategies, owing to its profound impact on decision-making processes, innovation, and competitive advantage. At its core, data science harnesses the power of data to extract actionable insights and drive informed decision-making across various industries. By leveraging advanced analytics techniques, such as machine learning and predictive modeling, data scientists can uncover hidden patterns, trends, and correlations within vast and complex datasets. These insights enable organizations to make strategic decisions that drive growth, optimize operations, and enhance customer experiences.

Moreover, data science facilitates predictive analytics, enabling organizations to anticipate future trends and outcomes based on historical data and statistical models. By forecasting demand, market dynamics, and consumer behavior, businesses can proactively adapt their strategies and capitalize on emerging opportunities, gaining a competitive edge in rapidly evolving markets. Predictive analytics also aids in risk management by identifying potential threats and vulnerabilities, allowing organizations to mitigate risks and protect against financial losses.

In addition to predictive capabilities, data science enables organizations to personalize products, services, and marketing campaigns to meet the unique needs and preferences of individual customers. By analyzing customer data and behavior, businesses can deliver targeted recommendations, personalized offers, and tailored experiences, fostering customer loyalty and satisfaction. This personalized approach not only enhances customer retention but also drives revenue growth by increasing cross-selling and upselling opportunities.

Furthermore, data science plays a crucial role in driving innovation and product development by providing insights that inform new product ideas, process improvements, and business transformation initiatives. By analyzing market trends, competitor strategies, and consumer feedback, organizations can identify unmet needs and develop innovative solutions that address evolving customer demands. This culture of innovation is essential for staying ahead of the competition and maintaining relevance in today's dynamic business landscape.

Data science also contributes to operational efficiency and cost reduction by identifying inefficiencies, automating repetitive tasks, and optimizing resource allocation. Through data-driven insights, organizations can streamline processes, reduce waste, and improve productivity, leading to significant cost savings and improved bottom-line performance. Additionally, data science plays a crucial role in risk management and fraud detection by analyzing patterns and anomalies in transaction data, healthcare records, and insurance claims. By detecting fraudulent activities early and mitigating risks, organizations can protect their assets, reputation, and stakeholder trust.

Data science is indispensable for organizations seeking to unlock the full potential of their data and drive business success in today's data-driven economy. By harnessing the power of data science, organizations can make smarter decisions, innovate faster, personalize customer experiences, optimize operations, and stay ahead of the competition in an increasingly competitive marketplace.

6.5 PROS AND CONS OF DATA SCIENCE

6.5.1 MERITS OF DATA SCIENCE:

1.**Informed Decision-Making**: Data science enables organizations to make data-driven decisions by analyzing and interpreting large volumes of data. By extracting valuable insights from datasets, decision-makers gain a deeper understanding of market trends, customer behavior, and operational performance, leading to more informed and strategic decision-making processes.

2. **Predictive Analytics**: Data science facilitates predictive analytics, allowing organizations to forecast future trends and outcomes based on historical data patterns. Predictive models enable businesses to anticipate customer preferences, market shifts, and demand fluctuations, enabling proactive decision-making and strategic planning.

3. **Personalization**: With data science, organizations can personalize products, services, and marketing campaigns to meet the unique needs and preferences of individual customers. By analyzing customer data, businesses can deliver targeted recommendations, personalized offers, and tailored experiences, enhancing customer satisfaction and loyalty.

4. **Innovation**: Data science drives innovation by uncovering new insights and opportunities for product development, process optimization, and business transformation. By analyzing market trends, competitor strategies, and consumer feedback, organizations can identify emerging trends and develop innovative solutions that address evolving customer demands.

5. **Operational Efficiency**: Data science helps organizations optimize processes, automate repetitive tasks, and streamline operations, leading to improved efficiency and productivity. By identifying inefficiencies and bottlenecks, businesses can optimize workflows, reduce costs, and allocate resources more effectively, ultimately enhancing overall operational performance.

6. **Risk Management**: Data science plays a crucial role in risk management by identifying potential risks, threats, and vulnerabilities through data analysis. By analyzing patterns and anomalies in data, organizations can detect fraudulent activities, mitigate risks, and protect against financial losses and reputational damage.

7. **Competitive Advantage**: Organizations that embrace data science gain a significant competitive advantage in their respective industries. By leveraging data-driven insights, businesses can differentiate themselves in the market, identify new opportunities, and stay ahead of competitors by anticipating market trends and customer preferences.

8. **Improved Customer Experiences**: Data science enables organizations to enhance customer experiences by delivering personalized products, services, and interactions. By analyzing customer data and feedback, businesses can tailor their offerings to meet individual needs, leading to increased customer satisfaction and loyalty.

6.5.2 DEMERITS OF DATA SCIENCE:

1. **Data Quality Issues**: One of the major challenges in data science is dealing with poor-quality or incomplete data. Data may contain errors, inconsistencies, or missing values, which can affect the accuracy and reliability of analytical results. Poor data quality can lead to biased insights and erroneous conclusions, undermining the effectiveness of data-driven decision-making.

2. **Privacy and Security Concerns**: Data science often involves the analysis of sensitive and confidential information, raising concerns about privacy and data security. Organizations must ensure compliance with data protection regulations and implement robust security measures to safeguard against unauthorized access, data breaches, and cyber threats. Failure to address privacy and security concerns can erode trust and damage reputation.

3. **Bias and Fairness Issues**: Data science models can be susceptible to bias, leading to unfair or discriminatory outcomes. Biases may arise from biased training data, algorithmic bias, or biased decision-making processes. Biased models can perpetuate inequality and injustice, particularly in areas such as hiring, lending, and criminal justice. Addressing bias and ensuring fairness in data science models is crucial for promoting ethical and responsible use of data.

4. **Over Reliance on Data**: Data science relies heavily on data to generate insights and make predictions. However, there may be instances where data alone may not provide a complete picture or adequately capture the complexities of real-world scenarios. Over Reliance on data without considering contextual factors or domain knowledge can lead to erroneous conclusions and misguided decision-making.

5. **Complexity and Interpretability**: Data science models, particularly those based on machine learning algorithms, can be complex and difficult to interpret. Black-box models, such as deep learning neural networks, may lack transparency, making it challenging to understand how predictions are generated. Lack of interpretability can hinder trust and acceptance of data science results, especially in critical domains where transparency is essential.

6. **Resource Intensive**: Implementing data science initiatives can be resource-intensive in terms of time, expertise, and infrastructure. Organizations require skilled data scientists, data engineers, and domain experts to design, develop, and deploy data science solutions. Moreover, data science projects often require significant computational resources and specialized software tools, adding to the overall cost and complexity of implementation.

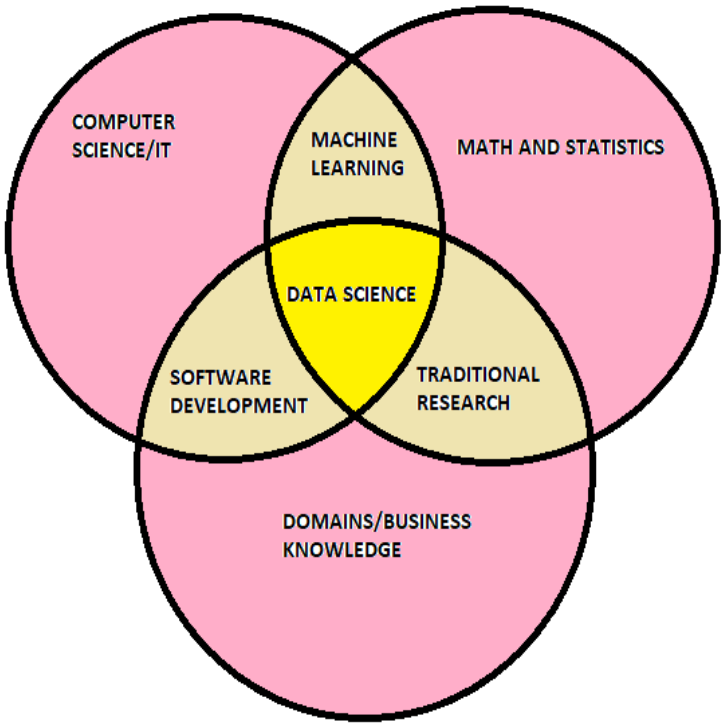
7. **Ethical Dilemmas**: Data science raises ethical dilemmas and moral considerations, particularly concerning privacy, consent, and data usage. Organizations must navigate ethical challenges related to data collection, storage, and sharing, ensuring that data science initiatives uphold principles of fairness, transparency, and accountability. Failure to address ethical concerns can lead to public backlash, regulatory scrutiny, and legal consequences.

8. **Limited Generalization**: Data science models may have limited generalization capabilities, particularly when applied to new or unseen data. Models trained on historical data may not necessarily perform well in different contexts or under changing conditions. Organizations must validate and test models rigorously to ensure their reliability and generalizability across diverse scenarios.

CHAPTER 7
MACHINE LEARNING WITH DATA SCIENCE

7.1 INTRODUCTION

Many people perceive data science as a broader field encompassing machine learning, and they're partially correct. Data science indeed involves handling vast amounts of data and applying machine learning algorithms, methods, and technologies to derive insights from this data. However, mastering data science requires proficiency not only in machine learning but also in mathematics, statistics, and subject expertise. Subject expertise refers to domain knowledge relevant to the data being analyzed, allowing one to abstract and interpret the data effectively. These three components—mathematics, statistics, and subject expertise—are considered the pillars of data science. Mastering all three areas is essential to becoming a top-tier data scientist. Hugh Conway's diagram illustrates this concept succinctly, showcasing the interconnectedness of these fundamental aspects in data science.



If you're considering a career in data science, there are a few key areas you'll need to focus on. Firstly, you'll need solid knowledge in three main domains: Analytics, Programming, and Domain Knowledge. However, mastering data science isn't just about having knowledge; it also requires honing critical skills. To become a proficient data scientist, you'll need to practice and develop certain skills:

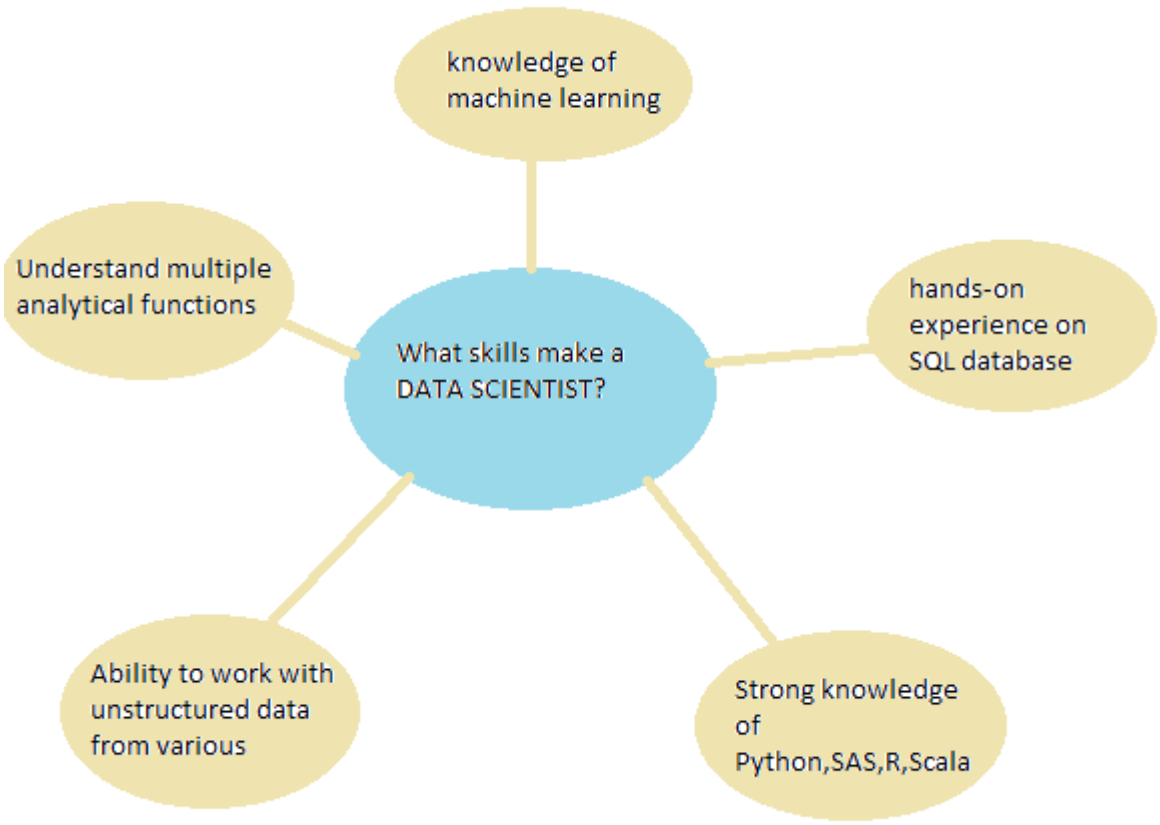
1. Expertise in Python, SAS, R, or SCALA programming languages.

2. Hands-on experience with SQL coding for data manipulation and analysis.

3. The ability to work with unstructured data effectively.

4. Understanding various analytical functions to extract insights from data.

5. Lastly, knowledge of machine learning algorithms and techniques is essential for predictive modeling and analysis.



7.2 HOW ML WORKS

Machine Learning is often seen as a subset of Data Science, but its definition goes beyond that. In simple terms, Machine Learning involves techniques to collect data, learn from it using various methods, and then use algorithms to predict future trends based on that data. Essentially, it revolves around spotting patterns and gaining insights from data.

To illustrate this definition, consider the example of Google. Google records your search history and uses machine learning algorithms to suggest similar searches in the future. Similarly, Amazon recommends products based on your previous searches, and Netflix suggests TV shows or movies based on your viewing history. These real-life examples demonstrate how machine learning works in practice.

As the domain of Machine Learning continues to grow exponentially, certain skills are essential for success in this field. These include:

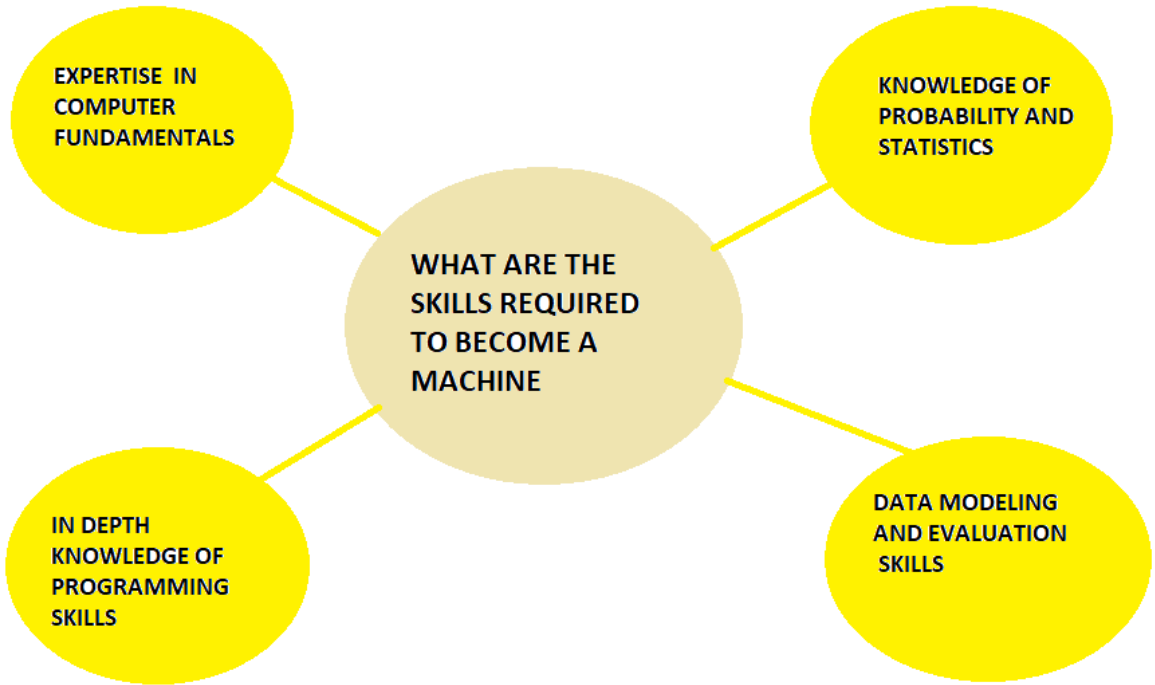
1. Strong computer fundamentals.

2. Proficiency in programming languages.

3. Knowledge of probability and statistics.

4. Data modeling skills.

By mastering these skills, you can excel in the field of Machine Learning and contribute to its ongoing advancement and innovation.



7.3 DATA SCIENCE VS MECHINE LEARNING

7.3.1 DATA SCIENCE:

Scope: Data science encompasses a wide range of activities, such as collecting, cleaning, analyzing, and visualizing data, along with developing data-driven solutions. Its focus is on extracting valuable insights from data to aid decision-making processes.

Goal: The primary objective of data science is to derive knowledge and insights from data, with a particular emphasis on solving complex real-world problems across different domains. While machine learning is often utilized, it's just one tool among many in the data scientist's toolkit.

Techniques: Data science utilizes various techniques, including statistical analysis, data visualization, exploratory data analysis (EDA), and machine learning. However, it extends beyond machine learning to include data engineering, integration, and domain-specific expertise.

Skills: Data scientists require a diverse skill set, encompassing tasks like data cleaning, statistical analysis, visualization, and domain knowledge. While they may possess expertise in machine learning, their focus is not limited to it alone.

Application: Data science finds applications in creating dashboards, generating reports, identifying trends, and building predictive models. Data scientists tackle a wide range of data-related tasks within organizations, contributing to informed decision-making and problem-solving efforts.

7.3.2 MACHINE LEARNING:

Scope: Machine learning is a specialized branch of artificial intelligence (AI) that focuses on developing models capable of extracting insights and making predictions from data autonomously, without explicit programming.

Goal: The primary objective of machine learning is to create models that can learn patterns from data and make predictions automatically. This technology is widely used for predictive analytics and automation across various industries.

Techniques: Machine learning employs specific techniques, including supervised, unsupervised, and reinforcement learning. These techniques involve training models on data and optimizing their performance to achieve desired outcomes.

Skills: Machine learning engineers and practitioners need deep expertise in machine learning algorithms, feature engineering, model selection, and hyperparameter tuning. Strong programming skills are essential, and professionals may specialize in model development and deployment.

Application: Machine learning finds applications in diverse areas such as image recognition, natural language processing (NLP), recommendation systems, fraud detection, and autonomous decision-making systems. Its versatility makes it a valuable tool for solving a wide range of complex problems.

7.4 PROS AND CONS OF MACHINE LEARNING

7.4.1 MERITS OF ML:

Machine learning, a subset of artificial intelligence, offers a myriad of advantages across various industries and applications, revolutionizing the way organizations operate and make decisions. Firstly, machine learning enables automation of repetitive and labor-intensive tasks, freeing up valuable time and resources for businesses. By leveraging algorithms that can learn from data, tasks such as data entry, processing, and analysis can be automated, leading to increased efficiency and productivity.

Secondly, machine learning facilitates predictive analytics by extracting insights from vast datasets and making accurate predictions about future trends and outcomes. This capability is invaluable for businesses in forecasting demand, identifying market opportunities, and mitigating risks. Whether it's predicting customer churn, stock prices, or equipment failure, machine learning algorithms can provide valuable insights to inform decision-making processes.

Moreover, machine learning empowers organizations to make data-driven decisions by analyzing complex datasets and identifying patterns and correlations that may not be apparent to human analysts. By leveraging algorithms that can process and interpret large volumes of data, businesses can gain deeper insights into customer behavior, market trends, and operational performance, enabling them to make informed decisions with confidence.

Additionally, machine learning enables personalized experiences for users by analyzing their preferences, behavior, and past interactions. From personalized product recommendations to customized marketing campaigns, machine learning algorithms can tailor experiences to individual users' needs and preferences, enhancing customer satisfaction and loyalty. This personalized approach not only improves customer engagement but also drives revenue growth for businesses.

Furthermore, machine learning facilitates continuous learning and improvement by enabling algorithms to adapt and evolve over time. By continuously learning from new data and feedback, machine learning models can refine their predictions and recommendations, staying relevant and accurate in dynamic environments. This adaptability makes machine learning particularly well-suited for tasks that require ongoing learning and adaptation, such as fraud detection, anomaly detection, and recommendation systems. Overall, machine learning offers a multitude of advantages, from automation and predictive analytics to improved decision-making and personalized experiences, empowering organizations to thrive in today's data-driven world.

7.4.2 DEMERITS OF ML:

While machine learning (ML) offers numerous benefits, it also presents several challenges and limitations that organizations must consider. Firstly, ML algorithms require substantial amounts of high-quality data for training, which can be both time-consuming and resource-intensive to collect and label. Moreover, biased or incomplete datasets can lead to biased models and inaccurate predictions, undermining the reliability and effectiveness of ML systems.

Secondly, ML models can be complex and difficult to interpret, particularly deep learning models, which operate as "black boxes" with little transparency into their decision-making processes. This lack of interpretability can hinder trust and understanding of ML systems, especially in critical applications such as healthcare and finance where transparency is essential.

Additionally, ML models are susceptible to overfitting, where they perform well on the training data but fail to generalize to unseen data. Overfitting can lead to poor performance and unreliable predictions, necessitating careful model selection, regularization techniques, and cross-validation to mitigate this risk.

Furthermore, ML systems may raise ethical and privacy concerns, particularly when dealing with sensitive or personal data. Issues such as data privacy, algorithmic bias, and fairness must be carefully addressed to ensure that ML applications do not inadvertently discriminate against certain groups or infringe upon individuals' rights to privacy and autonomy.

Moreover, implementing and maintaining ML systems requires specialized expertise and resources, including data scientists, machine learning engineers, and computational infrastructure. Small businesses and organizations with limited resources may struggle to leverage ML effectively, limiting their ability to compete in data-driven markets.

Lastly, ML systems are vulnerable to adversarial attacks, where malicious actors manipulate input data to deceive or compromise the performance of ML models. Adversarial attacks can have serious consequences, particularly in security-critical applications such as autonomous vehicles, medical diagnosis, and cybersecurity.

Overall, while machine learning offers significant opportunities for innovation and advancement, organizations must carefully consider the challenges and limitations associated with ML implementation to ensure the responsible and ethical use of this powerful technology.

CHAPTER 8
PYTHON

8.1 INTRODUCTION

Python is a high-level, interpreted programming language known for its simplicity and versatility. Created by Guido van Rossum in the late 1980s, Python has gained immense popularity due to its readability, ease of use, and extensive support for libraries and frameworks. Python's syntax emphasizes code readability and allows developers to express concepts in fewer lines of code compared to other programming languages.

8.2 PYTHON FEATURES

Python offers a wide range of features that make it suitable for various applications, from web development and data analysis to artificial intelligence and scientific computing. Some key features of Python include:

1. **Simple and Easy to Learn**: Python's straightforward syntax and readability make it accessible to beginners and experienced developers alike. Its simplicity allows developers to focus on solving problems rather than grappling with complex syntax.

2. **Extensive Standard Library**: Python comes with a rich standard library that provides ready-to-use modules and functions for performing common tasks, such as file I/O, networking, and data manipulation. This extensive library reduces development time and effort by eliminating the need to write code from scratch.

3. **Dynamic Typing and Memory Management**: Python is dynamically typed, meaning variable types are inferred at runtime, allowing for more flexible and expressive code. Additionally, Python features automatic memory management through garbage collection, relieving developers from manual memory management tasks.

4. **Platform Independence**: Python is platform-independent, meaning Python code written on one platform can run on various operating systems without modification. This cross-platform compatibility makes Python suitable for developing applications that need to run on different environments seamlessly.

8.3 PYTHON ECOSYSTEM

Python boasts a vibrant ecosystem of libraries, frameworks, and tools that enhance its capabilities and support various application domains. Some popular Python libraries and frameworks include:

1. **NumPy**: NumPy is a powerful library for numerical computing in Python, providing support for multidimensional arrays, mathematical functions, and linear algebra operations. NumPy is widely used in scientific computing, data analysis, and machine learning.

2. **Pandas**: Pandas is a versatile data manipulation library built on top of NumPy, offering data structures and functions for data analysis and manipulation. Pandas is particularly useful for working with structured data, such as tabular data and time series data.

3. **Django**: Django is a high-level web framework for building robust and scalable web applications in Python. Django follows the "batteries-included" philosophy, providing built-in features for authentication, routing, templating, and database management, among others.

4. **TensorFlow**: TensorFlow is an open-source machine learning framework developed by Google for building and training deep learning models. TensorFlow offers a flexible architecture for implementing various neural network architectures and supports distributed computing for training models at scale.

8.4 PYTHON APPLICATIONS

Python finds applications across a wide range of domains, owing to its versatility and ease of use. Some common applications of Python include:

1. **Web Development**: Python is widely used for web development, with frameworks like Django, Flask, and Pyramid enabling developers to build scalable and secure web applications quickly.

2. **Data Analysis and Visualization**: Python is a popular choice for data analysis and visualization tasks, thanks to libraries like Pandas, Matplotlib, and Seaborn. Python's simplicity and extensive support for data manipulation make it ideal for analyzing large datasets and generating insightful visualizations.

3. **Machine Learning and Artificial Intelligence**: Python's simplicity and rich ecosystem of machine learning libraries, such as TensorFlow, PyTorch, and Scikit-learn, make it a preferred language for developing machine learning and AI applications. Python's syntax facilitates rapid prototyping and experimentation, making it suitable for research and development in this domain.

4. **Automation and Scripting**: Python is commonly used for automation and scripting tasks, such as batch processing, file manipulation, and system administration. Python's readability and cross-platform compatibility make it well-suited for writing scripts that automate repetitive tasks and streamline workflows.

8.5 CONCLUSION

Python's simplicity, versatility, and rich ecosystem make it one of the most popular programming languages worldwide. Whether you're a beginner learning to code or an experienced developer working on complex projects, Python offers a wide range of tools and resources to meet your needs. From web development and data analysis to machine learning and automation, Python empowers developers to create innovative solutions and drive technological advancements across various industries. As Python continues to evolve and grow, its impact on the world of technology and software development is set to expand, making it an essential skill for aspiring programmers and seasoned professionals alike.

CHAPTER 9
ALGORITHM USED FOR ML

9.1 OVERVIEW

Machine learning (ML) is a branch of artificial intelligence (AI) that focuses on developing algorithms and models capable of learning patterns and making predictions from data. ML algorithms enable computers to perform tasks without being explicitly programmed, relying instead on patterns and inference derived from data. This document explores the core algorithms used in machine learning, providing an overview of their principles and applications.

9.2 SUPERVISED LEARNING ALGORITHMS

Supervised learning is a type of machine learning where algorithms learn from labeled data, consisting of input-output pairs, to make predictions on unseen data. Some common supervised learning algorithms include:

1. Linear Regression

2. Logistic Regression

3. Decision Trees

4. Random Forests

5. Support Vector Machines (SVM)

This section provides an overview of each algorithm, including its principles, advantages, and applications.

9.3 UNSUPERVISED LEARNING ALGORITHMS

Unsupervised learning is a type of machine learning where algorithms learn from unlabeled data to discover hidden patterns or structures. Some common unsupervised learning algorithms include:

1. K-Means Clustering

2. Hierarchical Clustering

3. Principal Component Analysis (PCA)

4. t-Distributed Stochastic Neighbor Embedding (t-SNE)

5. Generative Adversarial Networks (GANs)

This section discusses each algorithm's characteristics, use cases, and limitations.

9.4 SEMI-SUPERVISED LEARNING ALGORITHMS

Semi-supervised learning is a hybrid approach that combines labeled and unlabeled data to train models. It is particularly useful when labeled data is scarce or expensive to obtain. Some common semi-supervised learning algorithms include:

1. Label Propagation

2. Self-Training

3. Co-Training

4. Tri-Training

5. Graph-Based Methods

This section explores the principles and applications of semi-supervised learning algorithms, highlighting their benefits and challenges.

9.5 REINFORCEMENT LEARNING ALGORITHMS

Reinforcement learning is a type of machine learning where algorithms learn by interacting with an environment to maximize rewards. It is commonly used in sequential decision-making tasks, such as robotics and game playing. Some common reinforcement learning algorithms include:

1. Q-Learning

2. Deep Q-Networks (DQN)

3. Policy Gradient Methods

4. Actor-Critic Methods

5. Monte Carlo Tree Search (MCTS)

This section provides an overview of reinforcement learning algorithms, discussing their key concepts and applications.

9.6 ENSEMBLING LEARNING ALGORITHMS

Ensemble learning combines multiple base learners to improve predictive performance and robustness. It is a powerful technique commonly used in machine learning competitions and real-world applications. Some common ensemble learning algorithms include:

1. Bagging (Bootstrap Aggregating)

2. Boosting (AdaBoost, Gradient Boosting)

3. Stacking

4. Random Forests

5. XGBoost

This section explores the principles of ensemble learning and discusses various ensemble techniques and their applications.

CHAPTER 10
CODE PROCESS

10.1 PREPROCESS THE DATA

Preprocessing the data is a crucial step in preparing it for analysis and modeling. Here's a brief overview of the preprocessing steps you might take for a loan approval dataset:

1. **Data Cleaning:**

- Handling missing values: Identify and handle missing values in the dataset. This can involve imputation (replacing missing values with a statistical measure like mean or median) or deletion (removing rows or columns with missing values).

- Removing duplicates: Check for and remove any duplicate rows in the dataset.

- Outlier detection: Identify and handle outliers in the data. Outliers can skew analysis and modeling results.

2. **Feature Selection:**

- Identify relevant features: Determine which features are relevant for predicting loan approval. This may involve domain knowledge, exploratory data analysis, or feature importance techniques.

- Remove irrelevant features: Remove features that are not useful or redundant for the analysis.

3. **Feature Encoding:**

- Convert categorical variables: Encode categorical variables into numerical format using techniques like one-hot encoding or label encoding.

- Handle ordinal variables: If there are ordinal variables (variables with a natural order), encode them appropriately.

4. **Feature Scaling:**

- Scale numerical variables: Scale numerical variables to a similar range to prevent features with larger scales from dominating the model. Common scaling techniques include normalization (scaling to a [0, 1] range) or standardization (scaling to have mean 0 and standard deviation 1).

5. **Data Transformation:**

- Perform transformations: Apply transformations to variables to make their distributions more Gaussian-like, if necessary. This can include techniques like log transformation or Box-Cox transformation.

6. **Data Splitting:**

- Split the data into training and testing sets: Divide the dataset into two subsets – one for training the model and one for testing the model's performance.

7. **Handling Imbalanced Data:**

- If the dataset is imbalanced (e.g., significantly more approved loans than rejected loans), consider techniques such as oversampling, under sampling, or synthetic data generation to balance the classes.

8. **Feature Engineering:**

- Create new features: Generate new features from existing ones that may better capture patterns in the data.

These are some common preprocessing steps, but the specific steps you take may vary depending on the characteristics of your dataset and the requirements of your analysis or modeling task. Each step should be carefully executed to ensure the quality and integrity of the data for subsequent analysis.

10.2 EXPLORATORY DATA-ANALYSIS

Exploratory Data Analysis (EDA) is a critical phase in the data analysis process where you aim to understand the structure, relationships, and patterns within your dataset. Here's how you might approach EDA for a loan approval dataset:

1. **Summary Statistics:**

- Calculate basic summary statistics such as mean, median, standard deviation, minimum, and maximum for numerical variables.

- Count the frequency of different categories for categorical variables.

2. **Data Visualization:**

- Use visualizations like histograms, box plots, and density plots to understand the distribution of numerical variables.

- Create bar plots, pie charts, or count plots to visualize the distribution of categorical variables.

- Use scatter plots, pair plots, or heatmaps to explore relationships between pairs of variables.

3. **Identify Missing Values:**

- Determine the presence of missing values in the dataset.

- Visualize missing data patterns using heatmaps or bar plots to identify any systematic missingness.

4. **Explore Relationships:**

- Investigate the relationships between independent variables (features) and the target variable (loan approval status).

- Analyze correlations between numerical variables using correlation matrices or heatmaps.

- Explore how categorical variables influence loan approval rates through cross-tabulations or chi-square tests.

5. **Outlier Detection**:

- Identify outliers in the dataset using visualization techniques like box plots or scatter plots.

- Determine whether outliers are genuine data points or errors that need to be addressed.

6. **Feature Importance:**

- Assess the importance of different features in predicting loan approval using techniques like feature importance plots or statistical tests.

7. **Segmentation Analysis:**

- Explore how loan approval rates vary across different segments of the dataset (e.g., by gender, age group, income level, etc.).

- Conduct subgroup analysis to identify patterns and trends within specific segments.

8. **Time-Series Analysis (if applicable):**

- If the dataset includes time-series data (e.g., loan application dates), analyze trends, seasonality, and patterns over time.

9. **Dimensionality Reduction:**

- Apply dimensionality reduction techniques like PCA (Principal Component Analysis) or t-SNE (t-Distributed Stochastic Neighbor Embedding) to visualize high-dimensional data in lower dimensions.

10. **Data Quality Assessment**:

- Assess the overall quality and consistency of the data.

- Identify any potential data quality issues such as inconsistencies, errors, or biases.

By conducting thorough exploratory data analysis, you can gain valuable insights into the underlying structure of the data, which can inform subsequent modeling and decision-making processes.

10.3 FEATURE ENGINEERING

Transformation: All the categorical variables are stored in a new dataframe. One hot encoding:Dummy variables are created for all categorical variables using get_dummies function from pandas and all encoded columns are created. Taking the best estimators, training algorithm is performed on Test data.

The min-max scaling method is used for numerical features.

Min-max scaling is similar to z-score normalization in that it will replace every value in a column with a new value using a formula. In this case, that formula is:

$m = (x - xmin) / (xmax - xmin)$

Where:

- *m* is our new value
- *x* is the original cell value
- *xmin* is the minimum value of the column
- *xmax* is the maximum value of the column

Using this formula, we will see that the values of each column will now be between zero and one.

CHAPTER 11
CONCLUSION AND FUTURE WORKS

11.1 CONCLUSION

In conclusion, the exploratory data analysis (EDA) conducted on the loan approval dataset provided valuable insights into the characteristics and patterns within the data. Through summary statistics, data visualization, and relationship exploration, we gained a deeper understanding of the factors influencing loan approval decisions.

Key findings from the EDA include:

- Distribution of applicant income, loan amount, and loan term.

- Identification of missing values and potential outliers.

- Analysis of relationships between features and loan approval status.

- Exploration of correlations between numerical variables.

- Assessment of feature importance and segmentation analysis.

Overall, the EDA process helped us identify potential areas for further analysis and model development. By understanding the underlying structure of the data and the factors driving loan approval decisions, we are better equipped to build predictive models that can accurately forecast loan outcomes and support decision-making in the banking sector.

Moving forward, additional steps such as feature engineering, model selection, and evaluation will be necessary to develop robust loan approval models. However, the insights gained from EDA serve as a solid foundation for these subsequent analyses and contribute to the overall goal of improving loan approval processes and mitigating risk for financial institutions.

11.2 FUTURE ENHANCEMENT

In considering future enhancements to the loan approval system and analysis, several avenues can be explored to further improve its effectiveness and efficiency:

1. **Incorporation of Additional Data Sources:** Integration of alternative data sources such as social media activity, transaction history, or behavioral data could provide deeper insights into applicants' creditworthiness and improve predictive accuracy.

2. **Advanced Machine Learning Models:** Experimentation with more sophisticated machine learning algorithms, such as ensemble methods, gradient boosting, or neural networks, may enhance the predictive power of the model and capture complex relationships in the data more effectively.

3. **Feature Engineering:** Continued refinement of feature engineering techniques, including the creation of new features and transformation of existing ones, can help extract more meaningful information from the dataset and improve model performance.

4. **Real-time Data Processing:** Implementation of real-time data processing capabilities can enable the system to adapt to changing market conditions and make timely decisions on loan applications, leading to improved customer experience and operational efficiency.

5. **Explainable AI Techniques:** Adoption of explainable AI techniques can enhance model interpretability and transparency, enabling stakeholders to understand the rationale behind loan approval decisions and build trust in the system.

6. **Risk Stratification and Portfolio Management:** Development of risk stratification models and portfolio management tools can help financial institutions optimize their lending strategies, allocate resources more effectively, and minimize exposure to credit risk.

7. **Regulatory Compliance and Ethical Considerations:** Continual monitoring of regulatory requirements and adherence to ethical standards are essential aspects of system enhancement, ensuring compliance with industry regulations and safeguarding against potential biases or discriminatory practices.

8. **Customer-Centric Innovations:** Exploration of customer-centric innovations such as personalized loan products, flexible repayment options, and digital lending platforms can enhance the overall customer experience and satisfaction while maintaining risk management principles.

By focusing on these areas of future enhancement, the loan approval system can evolve into a more robust, agile, and customer-focused platform, capable of delivering greater value to both financial institutions and borrowers alike.

CHAPTER 12

SOURCE CODE

```
from google.colab import drive
drive.mount('/content/drive')
import pandas as pd
import numpy as np
from sklearn import preprocessing
df=pd.read_csv('/content/drive/MyDrive/Loan/loan_approval_dataset.csv')
df
df1=df.copy()
df.info()
df.isnull().sum()
df.describe()

df.Education.value_counts()
df.self_employed.value_counts()
df.loan_status.value_counts()
label_encoder = preprocessing.LabelEncoder()
df['Education']= label_encoder.fit_transform(df['Education'])
df['Education'].unique()
label_encoder = preprocessing.LabelEncoder()
df['self_employed']= label_encoder.fit_transform(df['self_employed'])
df['self_employed'].unique()
label_encoder = preprocessing.LabelEncoder()
df['loan_status']= label_encoder.fit_transform(df['loan_status'])
df['loan_status'].unique()
df
x=df.iloc[:,:12]
x
y=df.iloc[:,[-1]]
y
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =
train_test_split(x,y,random_state=84,test_size=0.30,shuffle=True)
x_train.sort_index(ascending=True,inplace=True)
x_train
y_train.sort_index(ascending=True,inplace=True)
y_train
x_test.sort_index(ascending=True,inplace=True)
x_test
y_test.sort_index(ascending=True,inplace=True)
y_test
# Using UNSCALED DATA for the prediction
#LOGISTIC REGRESSION
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,
precision_score, recall_score
from sklearn.metrics import classification_report
logreg = LogisticRegression(random_state=42)
logreg.fit(x_train,y_train)
y_pred=logreg.predict(x_test)
acc=accuracy_score(y_test,y_pred)
print(acc*100)
ps=precision_score(y_test,y_pred)
ps
rc=recall_score(y_test,y_pred)
rc
cm=confusion_matrix(y_test,y_pred)
cm
import matplotlib.pyplot as plt
import seaborn as sns
sns.heatmap(cm,annot=True,fmt='g')
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix', fontsize=17)
plt.show()
print(classification_report(y_test,y_pred,labels=[1]))
# Here, I used the raw (unscaled numerical data) numerical data for
prediction in LOGISTIC REGRESSION algorithm.
# Accuracy of the model is 61%.
# But the False Negative value is 2.

#RANDOM FOREST
from sklearn.ensemble import RandomForestClassifier
Classifier= RandomForestClassifier(n_estimators= 100, criterion=
"entropy")
Classifier.fit(x_train,y_train)
y_pred1=Classifier.predict(x_test)
acc=accuracy_score(y_test,y_pred1)
print(acc*100)
ps1=precision_score(y_test,y_pred1)
ps1
rc1=recall_score(y_test,y_pred1)
rc1
cm1=confusion_matrix(y_test,y_pred1)
cm1
sns.heatmap(cm1,annot=True,fmt='g')
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix', fontsize=17)
plt.show()
print(classification_report(y_test,y_pred1,labels=[1]))
```

```
# Using unscaled numerical data in RANDOM FOREST algorithm.
# The ACCURACY is 97%
# But False Negative value is 12.
#DECISION TREE ALGORITHM
from sklearn.tree import DecisionTreeClassifier
dtree=DecisionTreeClassifier(criterion='gini',max_depth=5,min_samples_s
plit=2,min_samples_leaf=1)
dtree.fit(x_train,y_train)
y_pred2=dtree.predict(x_test)
acc2=accuracy_score(y_test,y_pred2)
print(acc2*100)
ps2=precision_score(y_test,y_pred2)
ps2
rc2=recall_score(y_test,y_pred2)
rc2
cm2=confusion_matrix(y_test,y_pred2)
cm2
sns.heatmap(cm2,annot=True,fmt='g')
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix', fontsize=17)
plt.show()
print(classification_report(y_test,y_pred2,labels=[1]))
# Using unscaled numerical data in DECISION TREE algorithm.
# The ACCURACY is 97%.
# But the False Negative value is 30.
# UNSCALED DATA CONCLUSION
# Here i've used LOGISTIC REGRESSION, RANDOM FOREST AND DECISION TREE
algorithms

#    ALGORITHMS              ACC (%)      PS        RC      FN
#-------------------------------------------------------------
#1) Logistic regression        61        0.7      0.009     2
#2) Random forest              97       0.96       0.95    15
#3) Decision tree             96.8      0.94       0.98    30
#   Conclusion, The model is more effective when using Random forest
algorithm for prediciton
#   and worst prediction in Logistic regrssion.
# SCALING THE DATA
df1
df_num=df1.select_dtypes(include=[int,float])
df_num
df_cat=df1.select_dtypes(include=[object])
df_cat
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
# using MinMax Scaler
```

```
mm=MinMaxScaler()
n=mm.fit_transform(df_num)
df_mm=pd.DataFrame(n,columns=df_num.columns)
df_mm
sc=StandardScaler()
c=sc.fit_transform(df_num)
df_sc=pd.DataFrame(c,columns=df_num.columns)
df_sc
# using Label encoding
label_encoder = preprocessing.LabelEncoder()
df_cat['Education']= label_encoder.fit_transform(df_cat['Education'])
df_cat['Education'].unique()
label_encoder = preprocessing.LabelEncoder()
df_cat['self_employed']=
label_encoder.fit_transform(df_cat['self_employed'])
df_cat['self_employed'].unique()
label_encoder = preprocessing.LabelEncoder()
df_cat['loan_status']=
label_encoder.fit_transform(df_cat['loan_status'])
df_cat['loan_status'].unique()
df_cat
df_pre=pd.concat([df_mm,df_cat],axis=1)
df_pre
      x1=df_pre.iloc[:,:12]

x1
y1=df_pre.iloc[:,[-1]]
y1
x_trainn,x_testt,y_trainn,y_testt=train_test_split(x1,y1,random_state=9
4,test_size=0.30,shuffle=True)
x_trainn.sort_index(ascending=True,inplace=True)
x_trainn
y_trainn.sort_index(ascending=True,inplace=True)
y_trainn
x_testt.sort_index(ascending=True,inplace=True)
x_testt
y_testt.sort_index(ascending=True,inplace=True)
y_testt
#LOGISTIC REGRESSION
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,
precision_score, recall_score
from sklearn.metrics import classification_report
logregg = LogisticRegression(random_state=92)
logregg.fit(x_trainn,y_trainn)
y_predd=logregg.predict(x_testt)
ac=accuracy_score(y_testt,y_predd)
print(ac*100)
pss=precision_score(y_testt,y_predd)
```

```
pss
rcc=recall_score(y_testt,y_predd)
rcc
cmm=confusion_matrix(y_testt,y_predd)
cmm
sns.heatmap(cmm,annot=True,fmt='g')
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix', fontsize=17)
plt.show()
print(classification_report(y_testt,y_predd,labels=[1]))
# using scaled numerical data in LOGISTIC REGRESSION algorithm
# The ACCURACY is 91%.
# The False Negative value is 56.
#RANDOM FOREST
from sklearn.ensemble import RandomForestClassifier
Classifierr= RandomForestClassifier(n_estimators= 100, criterion=
"entropy")
Classifierr.fit(x_trainn,y_trainn)
y_predd1=Classifierr.predict(x_testt)
ac1=accuracy_score(y_testt,y_predd1)
print(ac1*100)
pcc1=precision_score(y_testt,y_predd1)
pcc1
rcc1=recall_score(y_testt,y_predd1)
rcc1
cmm1=confusion_matrix(y_testt,y_predd1)
cmm1
sns.heatmap(cmm1,annot=True,fmt='g')
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix', fontsize=17)
plt.show()
print(classification_report(y_testt,y_predd1,labels=[1]))
#DECISION TREE ALGORITHM
from sklearn.tree import DecisionTreeClassifier
dtreee=DecisionTreeClassifier(criterion='gini',max_depth=5,min_samples_
split=2,min_samples_leaf=1)
dtreee.fit(x_trainn,y_trainn)
y_predd2=dtreee.predict(x_testt)
ac2=accuracy_score(y_testt,y_predd2)
print(ac2*100)
pss2=precision_score(y_testt,y_predd2)
pss2
rcc2=recall_score(y_testt,y_predd2)
rcc2
cmm2=confusion_matrix(y_testt,y_predd2)
cmm2
```

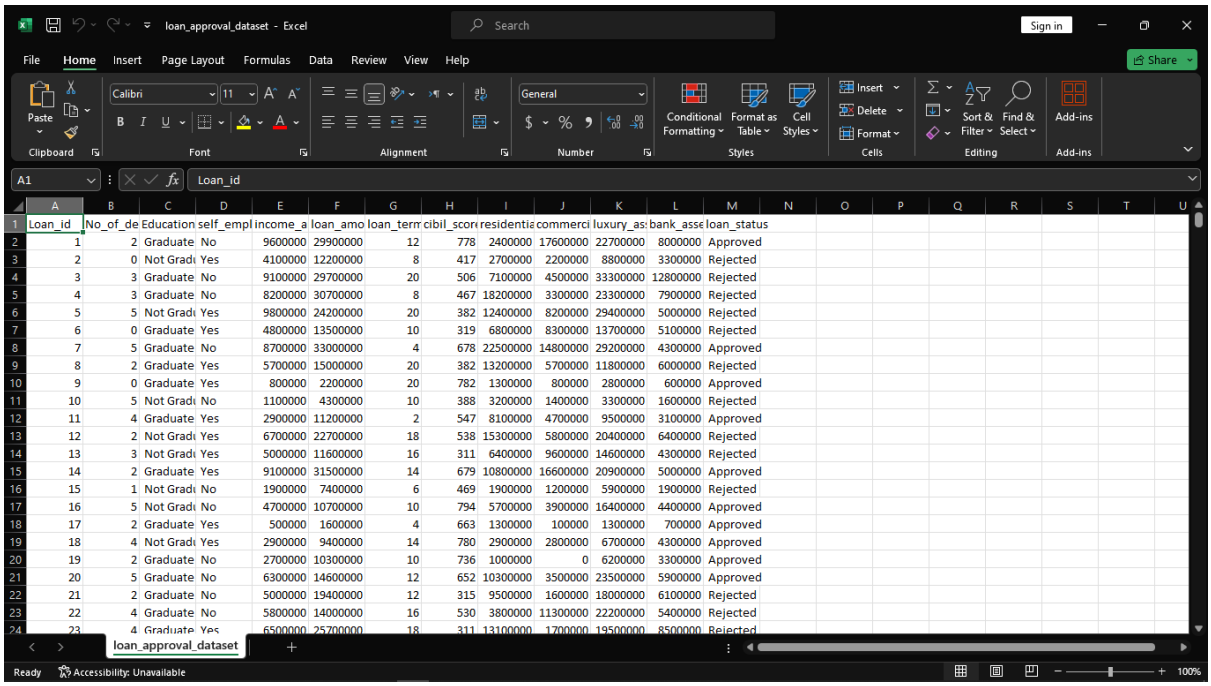
```
sns.heatmap(cmm2,annot=True,fmt='g')
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix', fontsize=17)
plt.show()
print(classification_report(y_testt,y_predd2,labels=[1]))
# using scaled numerical data in DECISION TREE algorithm
# The ACCURACY is 96.7%
# The False Negative value is 35
# SCALED DATA CONCLUSION
# Here i've used LOGISTIC REGRESSION, RANDOM FOREST AND DECISION TREE
algorithms

#    ALGORITHMS              ACC (%)     PS       RC     FN
#-------------------------------------------------------------
#1) Logistic regression        91       0.8      0.8     56
#2) Random forest             97.8      0.9      0.9     12
#3) Decision tree             96.7      0.9      0.98    35
#   Conclusion, The model is more effective when using Random forest
algorithm for prediciton
#   and worst prediction in Logistic regrssion.
print('Overall conclusion, The Random Forest algorithm is the best
ALGORITHM for this model. \nIt gives 97% accuracy for both scaled and
unscaled data and low False Negative vallues like 15 and 12.')
#Bar plot between annual income and loan amount
import matplotlib.pyplot as plt
crosstb = pd.crosstab(df1.loan_status,df1.self_employed)
barplot = crosstb.plot.bar(rot=0)
barplot.legend(title='self_employed', bbox_to_anchor=(1, 1.02),
loc='upper left')
#The Above Plot shows the relation between self employed & Loan Status
crosstb1 = pd.crosstab(df1.Education,df1.loan_status)
barplot1 = crosstb1.plot.bar(rot=0)
barplot1.legend(title='Loan status', bbox_to_anchor=(1, 1.02),
loc='upper left')
#This above plot shows the relation between education and loan status
#Scatter plot for cibil_score and loan_amount
plt.scatter(df1['cibil_score'], df1['loan_amount'], c ="black")
plt.show()
#Scatter plot for loan_status and cibil_score
plt.scatter(df1['loan_status'], df1['cibil_score'], c ="black")
plt.show()
#Count plot for annual income column
sns.countplot(x ='income_annum', data = df1)
plt.show()
#Count plot for loan status column
sns.countplot(x ='loan_status', data = df1)
plt.show()
```

```
#plot
#rocauc curverAUC ROC stands for “Area Under the Curve” of the
“Receiver Operating Characteristic” curve.
#The AUC ROC curve is basically a way of measuring the performance of
an ML model
# ROC AUC SCORE FOR UNSCALED DATA
from sklearn.metrics import roc_auc_score
score = roc_auc_score(y_test, y_pred)
print(f"ROC AUC: {score:.4f}")
#this is for Logistic regression
score = roc_auc_score(y_test, y_pred1)
print(f"ROC AUC: {score:.4f}")
#this is for RANDOM FOREST ALGORITHM
score = roc_auc_score(y_test, y_pred2)
print(f"ROC AUC: {score:.4f}")
#this is for DECISION TREE ALGORITHM
# ROC AUC SCORE FOR SCALED DATA
score = roc_auc_score(y_testt, y_predd)
print(f"ROC AUC: {score:.4f}")
#this is for Logistic regression
score = roc_auc_score(y_testt, y_predd1)
print(f"ROC AUC: {score:.4f}")
#this is for RANDOM FOREST ALGORITHM
score = roc_auc_score(y_testt, y_predd2)
print(f"ROC AUC: {score:.4f}")
#this is for DECISION TREE ALGORITHM
# CONCLUSION
# BASED ON THE ROC AND AUC CURVE, RANDOM FOREST ALGORITHM IS MORE
EFFECTIVE FOR THIS DATASET.
```

CHAPTER 13

SCREENSHOTS

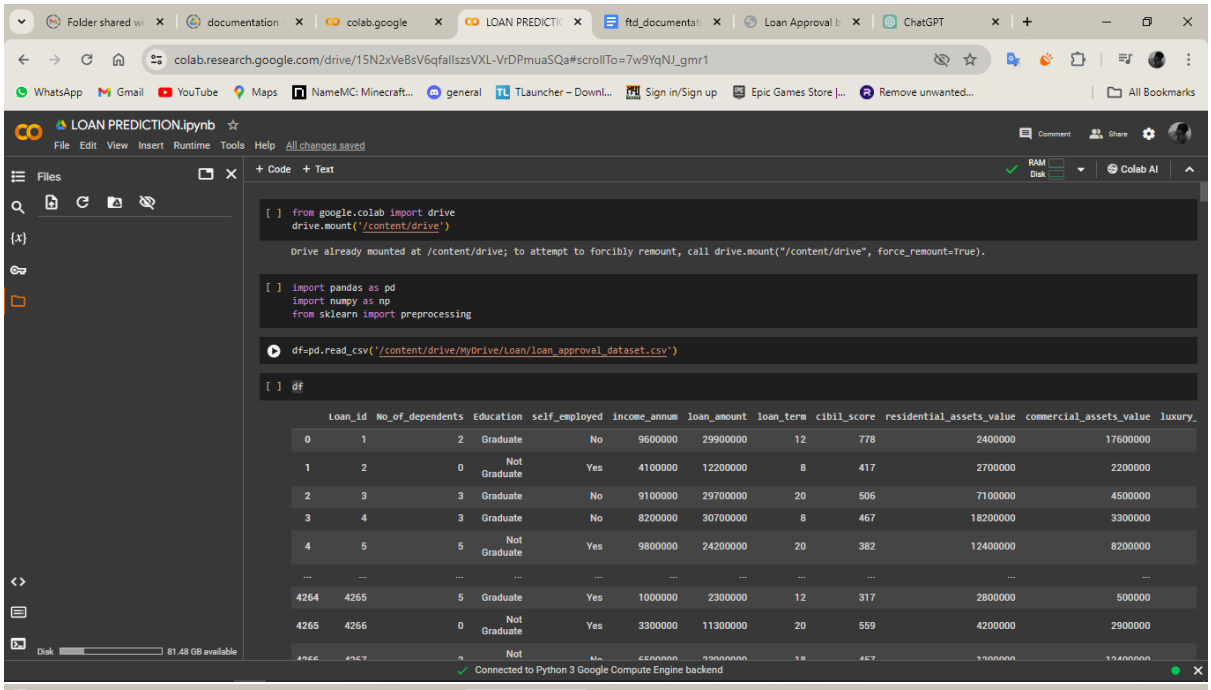
13.1 EXCEL SHEET

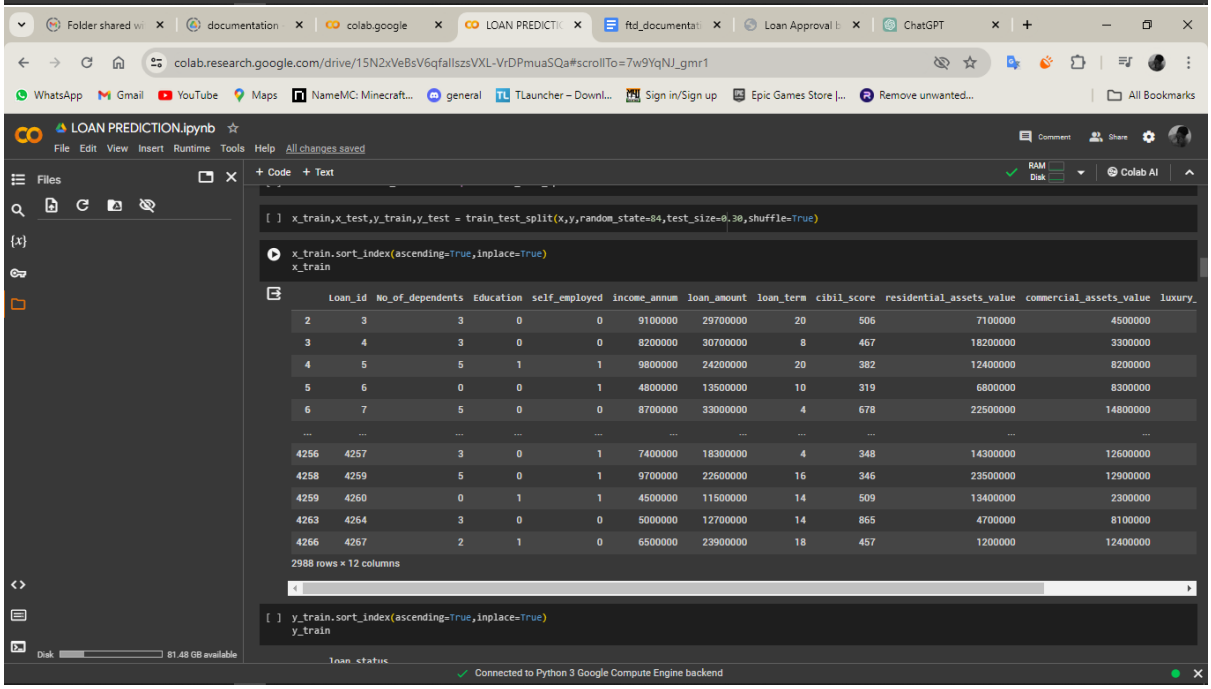


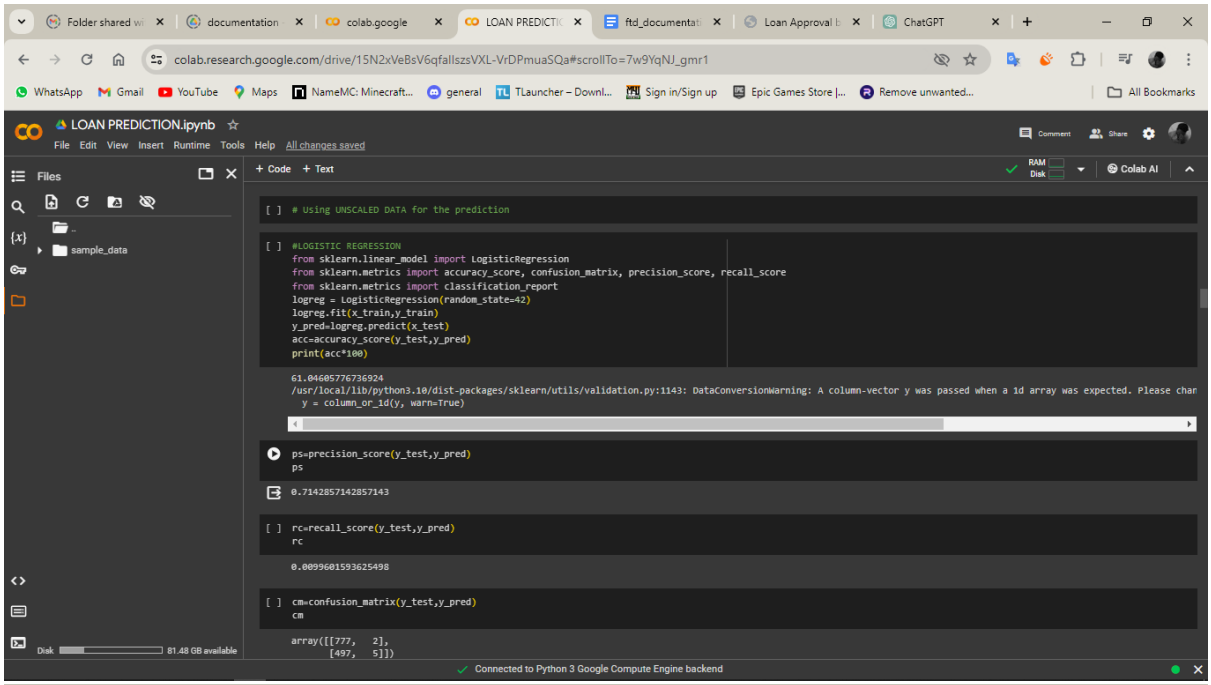
loan_approval_dataset - Excel

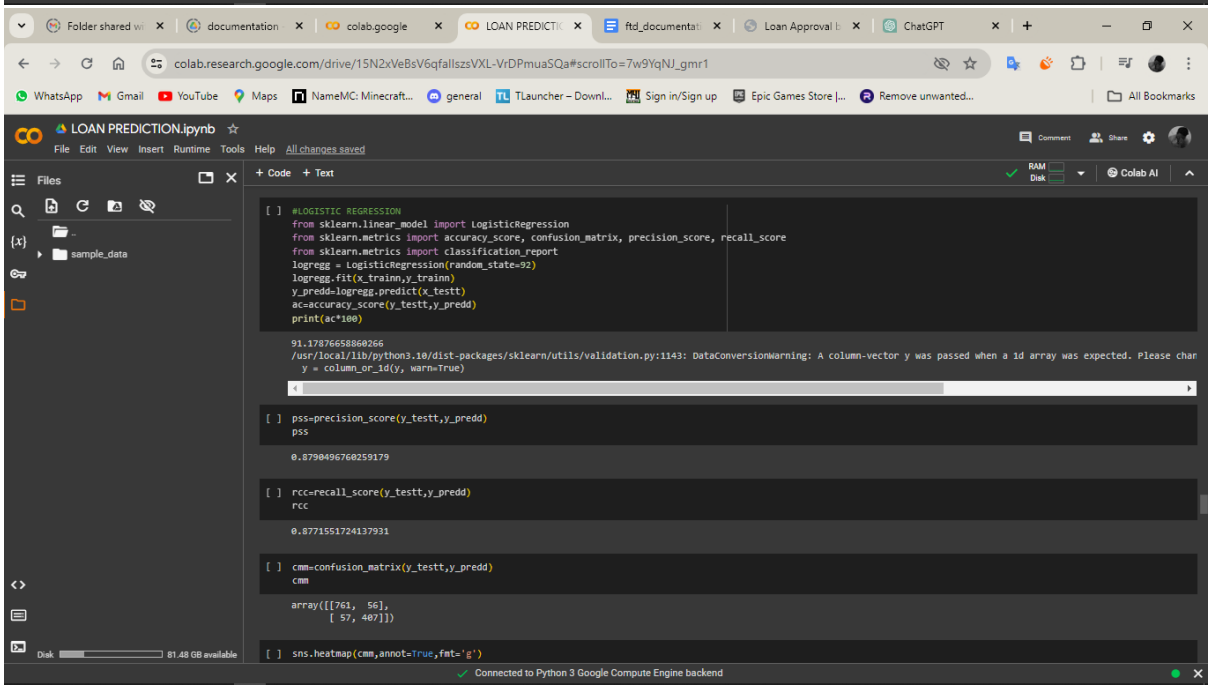
Loan_id	No_of_de	Education	self_empl	income_a	loan_amo	loan_term	cibil_scor	residentia	commerci	luxury_as	bank_asse	loan_status
1	2	Graduate	No	9600000	29900000	12	778	2400000	17600000	22700000	8000000	Approved
2	0	Not Grad	Yes	4100000	12200000	8	417	2700000	2200000	8800000	3300000	Rejected
3	3	Graduate	No	9100000	29700000	20	506	7100000	4500000	33300000	12800000	Rejected
4	3	Graduate	No	8200000	30700000	8	467	18200000	3300000	23300000	7900000	Rejected
5	5	Not Grad	Yes	9800000	24200000	20	382	12400000	8200000	29400000	5000000	Rejected
6	0	Graduate	Yes	4800000	13500000	10	319	6800000	8300000	13700000	5100000	Rejected
7	5	Graduate	No	8700000	33000000	4	678	22500000	14800000	29200000	4300000	Approved
8	2	Graduate	Yes	5700000	15000000	20	382	13200000	5700000	11800000	6000000	Rejected
9	0	Graduate	Yes	800000	2200000	20	782	1300000	800000	2800000	600000	Approved
10	5	Not Grad	No	1100000	4300000	10	388	3200000	1400000	3300000	1600000	Rejected
11	4	Graduate	Yes	2900000	11200000	2	547	8100000	4700000	9500000	3100000	Approved
12	2	Not Grad	Yes	6700000	22700000	18	538	15300000	5800000	20400000	6400000	Rejected
13	3	Not Grad	Yes	5000000	11600000	16	311	6400000	9600000	14600000	4300000	Rejected
14	2	Graduate	Yes	9100000	31500000	14	679	10800000	16600000	20900000	5000000	Approved
15	1	Not Grad	No	1900000	7400000	6	469	1900000	1200000	5900000	1900000	Rejected
16	5	Not Grad	No	4700000	10700000	10	794	5700000	3900000	16400000	4400000	Approved
17	2	Graduate	Yes	500000	1600000	4	663	1300000	100000	1300000	700000	Approved
18	4	Not Grad	Yes	2900000	9400000	14	780	2900000	2800000	6700000	4300000	Approved
19	2	Graduate	No	2700000	10300000	10	736	1000000	0	6200000	3300000	Approved
20	5	Graduate	No	6300000	14600000	12	652	10300000	3500000	23500000	5900000	Approved
21	2	Graduate	No	5000000	19400000	12	315	9500000	1600000	18000000	6100000	Rejected
22	4	Graduate	No	5800000	14000000	16	530	3800000	11300000	22200000	5400000	Rejected
23	4	Graduate	Yes	6500000	25700000	18	311	13100000	1700000	19500000	8500000	Rejected

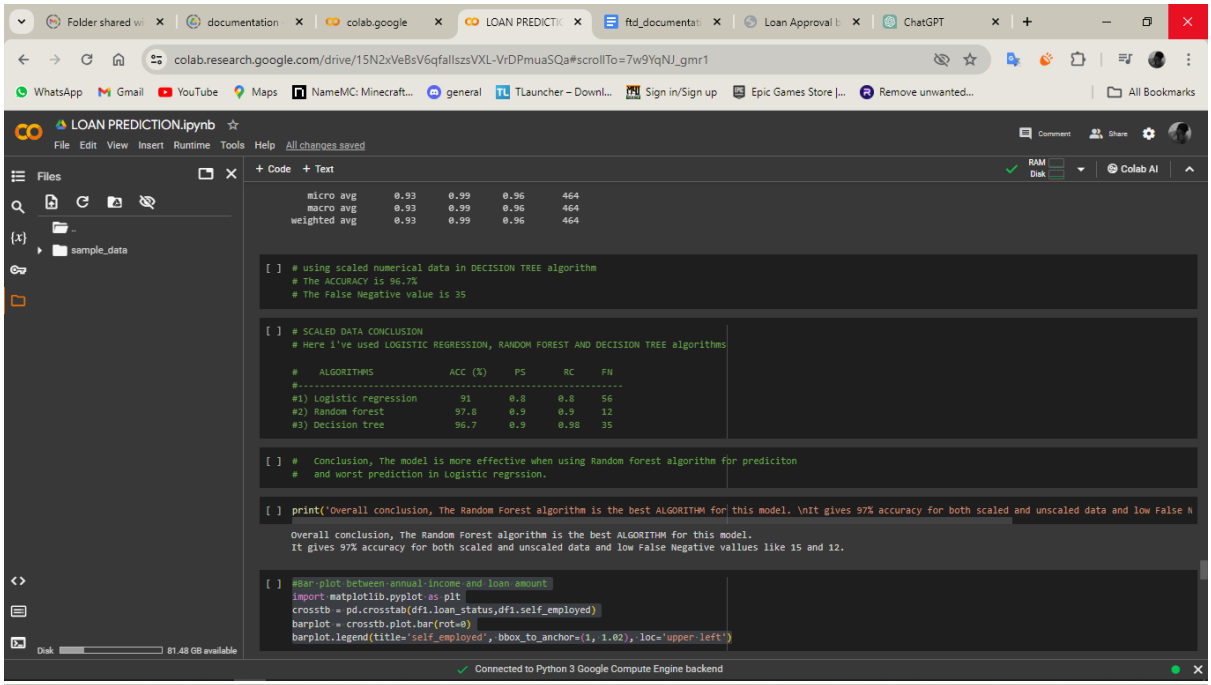
13.2 CODE SNAPS

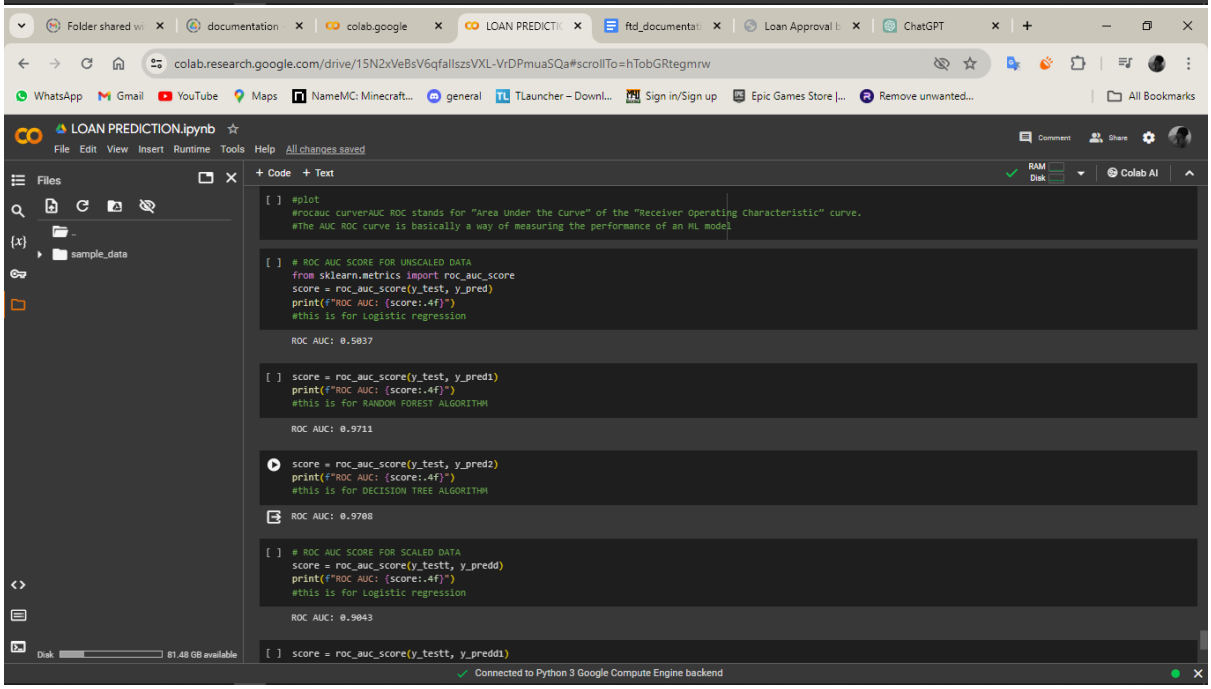





LOAN PREDICTION.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Files
sample_data
[ ] # Using UNSCALED DATA for the prediction
[ ] #LOGISTIC REGRESSION
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score
from sklearn.metrics import classification_report
logreg = LogisticRegression(random_state=42)
logreg.fit(x_train,y_train)
y_pred=logreg.predict(x_test)
acc=accuracy_score(y_test,y_pred)
print(acc*100)
61.04605776736924
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please char
y = column_or_1d(y, warn=True)
ps=precision_score(y_test,y_pred)
ps
0.7142857142857143
[ ] rc=recall_score(y_test,y_pred)
rc
0.009601593625498
[ ] cm=confusion_matrix(y_test,y_pred)
cm
array([[777, 2],
[497, 5]])
Disk 81.48 GB available
Connected to Python 3 Google Compute Engine backend


LOAN PREDICTION.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Files
sample_data
[ ] #LOGISTIC REGRESSION
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score
from sklearn.metrics import classification_report
logregg = LogisticRegression(random_state=92)
logregg.fit(x_trainn,y_trainn)
y_predd=logregg.predict(x_testt)
ac=accuracy_score(y_testt,y_predd)
print(ac*100)
91.17876658860266
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please char
y = column_or_1d(y, warn=True)
[ ] pss=precision_score(y_testt,y_predd)
pss
0.8790496760259179
[ ] rcc=recall_score(y_testt,y_predd)
rcc
0.877155172413793
[ ] cmm=confusion_matrix(y_testt,y_predd)
cmm
array([[761, 56],
[57, 407]])
[ ] sns.heatmap(cmm,annot=True,fmt='g')
Disk 81.48 GB available
Connected to Python 3 Google Compute Engine backend


LOAN PREDICTION.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
micro avg 0.93 0.99 0.96 464
macro avg 0.93 0.99 0.96 464
weighted avg 0.93 0.99 0.96 464
using scaled numerical data in DECISION TREE algorithm
The ACCURACY is 96.7%
The False Negative value is 35
SCALED DATA CONCLUSION
Here i've used LOGISTIC REGRESSION, RANDOM FOREST AND DECISION TREE algorithms
ALGORITHMS ACC (%) PS RC FN
#1) Logistic regression 91 0.8 0.8 56
#2) Random forest 97.8 0.9 0.9 12
#3) Decision tree 96.7 0.9 0.98 35
Conclusion, The model is more effective when using Random forest algorithm for prediciton
and worst prediction in Logistic regrssion.
print('Overall conclusion, The Random Forest algorithm is the best ALGORITHM for this model. \nIt gives 97% accuracy for both scaled and unscaled data and low False N
Overall conclusion, The Random Forest algorithm is the best ALGORITHM for this model.
It gives 97% accuracy for both scaled and unscaled data and low False Negative vallues like 15 and 12.
#Bar plot between annual income and loan amount
import matplotlib.pyplot as plt
crosstb = pd.crosstab(df1.loan_status,df1.self_employed)
barplot = crosstb.plot.bar(rot=0)
barplot.legend(title='self_employed', bbox_to_anchor=(1, 1.02), loc='upper left')
Files
sample_data
Disk 81.48 GB available
Connected to Python 3 Google Compute Engine backend


LOAN PREDICTION.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
#roc
#rocauc curverAUC ROC stands for "Area Under the Curve" of the "Receiver Operating characteristic" curve.
#The AUC ROC curve is basically a way of measuring the performance of an ML model
# ROC AUC SCORE FOR UNSCALED DATA
from sklearn.metrics import roc_auc_score
score = roc_auc_score(y_test, y_pred)
print(f"ROC AUC: {score:.4f}")
#this is for Logistic regression
ROC AUC: 0.5037
score = roc_auc_score(y_test, y_pred1)
print(f"ROC AUC: {score:.4f}")
#this is for RANDOM FOREST ALGORITHM
ROC AUC: 0.9711
score = roc_auc_score(y_test, y_pred2)
print(f"ROC AUC: {score:.4f}")
#this is for DECISION TREE ALGORITHM
ROC AUC: 0.9708
# ROC AUC SCORE FOR SCALED DATA
score = roc_auc_score(y_testt, y_predd)
print(f"ROC AUC: {score:.4f}")
#this is for Logistic regression
ROC AUC: 0.9043
score = roc_auc_score(y_testt, y_predd1)
Files
sample_data
Disk 81.48 GB available
Connected to Python 3 Google Compute Engine backend

CHAPTER 14
REFERENCE

[1]. Dileep B. Desai, Dr. R.V.Kulkarni “A Review: Application of Data Mining Tools in CRM for Selected Banks”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (2), 2013, 199 –201.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989. [2]. J.H. Aboobyda, and M.A. Tarig, “Developing Prediction Model of Loan Risk in BanksUsing Data Mining”, MachineLearning and Applications: An International Journal (MLAIJ), vol. 3, no.1, pp. 1–9, 2016. K. Elissa, "Title of paper if known," unpublished. [3]. A.B. Hussain, and F.K.E. Shorouq, “Credit risk assessment model for Jordanian commercial banks: Neuralscoring approach”, Review of Development Finance, Elsevier, vol. 4, pp. 20–28, 2014. JAC: A JOURNAL OF COMPOSITION THEORYVolume XIII, Issue V, MAY 2020ISSN: 0731-6755Page No: 324 [4]. T. Harris, “Quantitative credit risk assessment using support vector machines: Broad versus Narrow default definitions”, Expert Systems with Applications, vol. 40, pp. 4404–4413, 2013.