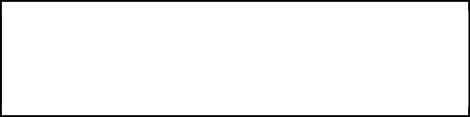
Bansilal Ramnath Agarwal Charitable Trust’s

**Vishwakarma Institute of Information Technology, Pune-48**

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Department of Computer Science and Engineering (Artificial Intelligence)



**Lab Assignment Submission**

**Machine Learning Lab**

**SY**

***Semester II Academic Year 2023-24***

***Guided by,***

Dr. Anuradha Yenkikar

***Student name: Neha Sawant***

***Batch:A3***

***Roll no: 281060***

***PRN:22210877***

**ASSIGNMENT NO: 1**

1.1] Problem Statement -

Perform the following operations using R/Python on suitable data sets:

a) read data from different formats (like csv, xls)

b) indexing and selecting data, sort data,

c) describe attributes of data, checking data types of each column,

d) counting unique values of data, format of each column, converting variable data type

(e.g. from long to short, vice versa),

e) identifying missing values and fill in the missing values

1.2] S/W Packages and Libraries used:

For the following assignment, the interpreter used was Google Collab and the Libraries used were

* Pandas - It is a powerful data manipulation library in Python, that provides data structures and functions for working with structured data.
* Numpy - NumPy is a fundamental package for scientific computing with Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

1.3] Theory-

**Introduction To pandas:**

1) Pandas is a powerful and widely-used open-source Python library for data manipulation and analysis.

2) It provides easy-to-use data structures and functions, making it an essential tool for working with structured data.

3) At the core of Pandas are two main data structures: Series and DataFrame.

4) A Series is a one-dimensional labelled array capable of holding any data type.

5) DataFrame is a two-dimensional labelled data structure with columns of potentially different types.

6) These data structures allow users to perform a wide range of operations on data, including loading data from various file formats (such as CSV, Excel, SQL databases), manipulating data (e.g., sorting, filtering, grouping), and performing statistical and analytical tasks.

Some basic functions that we used in program:

1. pd.read\_csv()**:** This function is used to read data from a CSV file into a DataFrame

2. head()**:** It is used to display the first few rows of the DataFrame, providing a quick overview of the data.

3. sort\_values()**:** This function sorts the DataFrame by the values of a specified column (in this case, 'Age'), allowing data to be arranged in ascending order.

4. describe()**:** It generates descriptive statistics for numerical columns in the DataFrame, such as count, mean, standard deviation, minimum, and maximum values.

5. head()**:** It returns the first n rows of a DataFrame, providing a quick way to preview the structure and content of the dataset.

6. describe()**:** This function generates descriptive statistics for numerical columns in the DataFrame, such as count, mean, standard deviation, minimum, and maximum values.

7. unique()**:** This function returns an array of unique values in a column of the DataFrame, useful for identifying distinct categories or groups in categorical data.

Methodology:

* For Reading Data from Different Formats:
  + Utilize libraries like Pandas to read data from various formats such as CSV, Excel (xls/xlsx), etc
  + Use of appropriate functions like read\_csv() or read\_excel() to load the data into Python data structures like DataFrames.
* For Indexing and Selecting Data, Sorting Data:
  + Pandas provides powerful indexing and selection mechanisms using methods like .loc[] and .iloc[].
  + Sorting data can be done using the sort\_values() function, specifying the column(s) to sort by.
* For Describing Attributes of Data, Checking Data Types:
  + Use the describe() method to generate descriptive statistics of the data.
  + Check data types of each column using the dtypes attribute of the DataFrame.
* For Counting Unique Values, Formatting Columns, Converting Variable Data Types:
  + Employ functions like value\_counts() to count unique values in a column.
  + Convert variable data types using functions like astype() to cast one data type to another.
* Identifying Missing Values and Filling Them:
  + Pandas provides functions like isna() or isnull() to identify missing values.
  + Use methods like fillna() or dropna() to handle missing data by filling them with appropriate values or dropping rows/columns with missing values.

1.4] Applications:

* Data Analysis: These operations are fundamental to exploratory data analysis (EDA), a crucial step in any data science or machine learning project.
* Data Cleaning: Handling missing values, formatting data, and converting data types are essential steps in data cleaning pipelines.

1.5] Limitations:

* Performance: While Pandas and NumPy are powerful libraries, they may not be optimal for very large datasets due to memory limitations.
* Learning Curve: These libraries have a steep learning curve, especially for beginners, which might hinder the quick implementation of complex operations.

1.6] Conclusion:

In conclusion, the assignment showcased the effective use of Python libraries like Pandas and NumPy for diverse data manipulation tasks, from reading various formats to handling missing values. These operations are vital for data analysis, cleaning, and visualization, offering essential insights into structured data. Despite their benefits, it's important to recognize the limitations of these libraries, such as performance constraints with large datasets. Nevertheless, with a solid understanding of their theoretical framework, Pandas and NumPy empower proficient data handling, supporting informed decision-making across domains.

**ASSIGNMENT NO: 2**

**Exploratory Data Analysis and Machine Learning Modeling**

2.1] Problem Statement -

Perform the following operations using Python on the data sets:

a) Compute and display summary statistics for each feature available in the dataset. (e.g minimum value, maximum value, mean, range, standard deviation, variance and percentiles

b) Data Visualization-Create a histogram for each feature in the dataset to illustrate the feature distributions.

c) Data cleaning, Data integration, Data transformation, Data model building (e.g. Classification)

2.2] S/W Packages and Libraries used:

For the following assignment, the interpreter used was Google Collab and the Primary Library used was-

* Matplotlib: Matplotlib is a versatile plotting library in Python, offering a wide range of visualization capabilities, including histograms for illustrating feature distributions.

2.3] Theory-

* For Computing Summary Statistics:
  + Utilize libraries like Pandas to load the dataset into a DataFrame.
  + Use Pandas' describe() method to generate summary statistics such as minimum, maximum, mean, standard deviation, variance, and percentiles for each feature.
* For Data Visualization - Histogram Creation:
  + Employ Matplotlib, a widely-used plotting library in Python, to create histograms for each feature.
  + Iterate through each feature in the dataset, plot its histogram using Matplotlib's hist() function, and customize the plot as necessary to illustrate feature distributions effectively.
* Data Cleaning:
  + Identify and handle missing values using techniques like imputation or removal.
  + Detect and handle outliers that may affect the integrity of the dataset.
* Data Integration:
  + Merge or concatenate multiple datasets if required for analysis.
  + Ensure consistency in data formats and representations across integrated datasets.

2.4] Applications:

Descriptive Analysis: Summary statistics provide insights into the central tendency, variability, and distribution of features, aiding in descriptive analysis.

Exploratory Data Analysis (EDA): Histograms visualize the distribution of individual features, revealing patterns, trends, and potential outliers.

2.5] Limitations:

Data Quality: Incomplete or inaccurate data can lead to biased summary statistics and misleading visualizations.

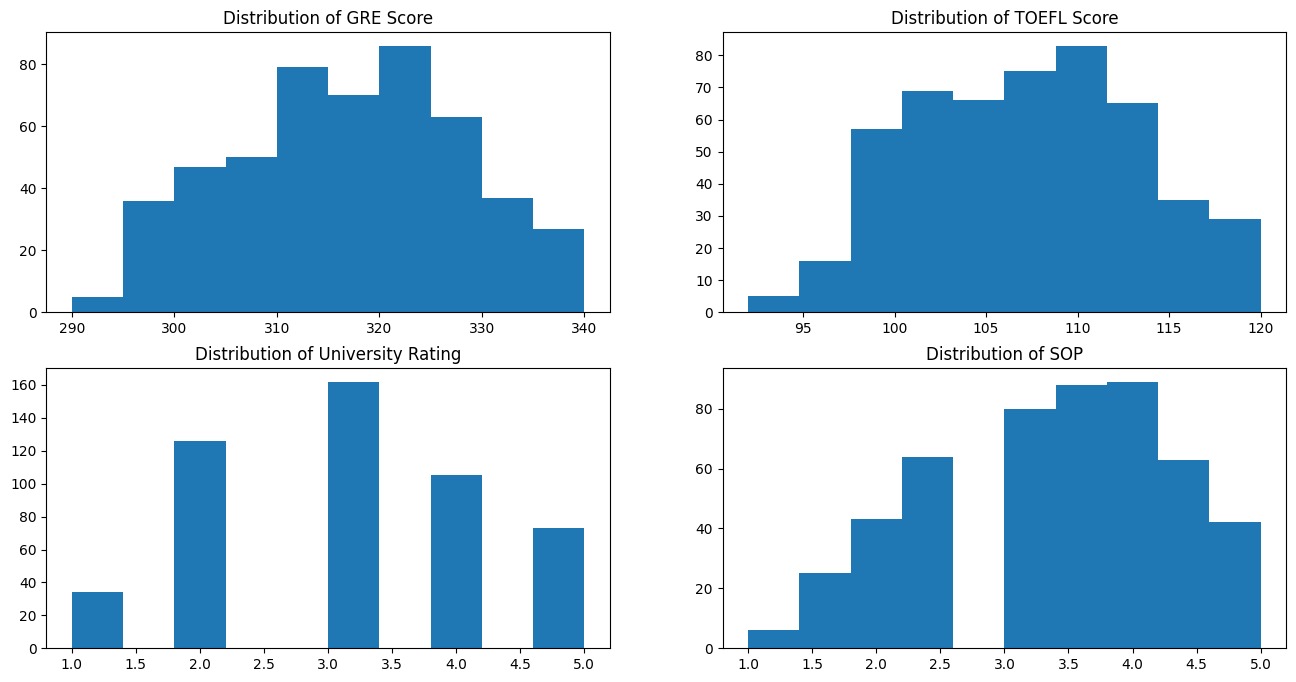
Model Performance: The success of classification models depends on factors like feature selection, model choice, and hyperparameter tuning, which may require iterative experimentation.

2.6] Working:

1. Load the dataset using Pandas.
2. Compute summary statistics using the **describe()** function.
3. Visualize data distributions using histograms with Matplotlib and Seaborn.
4. Perform data cleaning, integration, and transformation as necessary.
5. Build a machine learning classification model using Scikit-learn.
6. Evaluate the model's performance using appropriate metrics such as accuracy, precision, recall, etc.

**Diagram:**





2.7] Conclusion:

By computing summary statistics, visualizing feature distributions, and executing data cleaning, integration, transformation, and model-building processes, it addresses essential aspects of exploratory data analysis and classification tasks. While offering valuable insights and enabling the construction of predictive models, it's important to acknowledge potential limitations such as data quality issues and computational constraints. Nevertheless, leveraging these libraries empowers practitioners to navigate datasets efficiently, derive meaningful insights, and develop robust classification models for informed decision-making.

**ASSIGNMENT NO: 3**

3.1] Problem Statement -

Apply appropriate ML algorithm on a dataset collected in a cosmetics shop showing details of customers to predict customer response for special offers.

3.2] S/W Packages and Libraries used:

For the following assignment, the interpreter used was Google Collab and the Primary Library used was-

* Scikit-learn: Scikit-learn offers a wide range of machine learning algorithms and evaluation metrics, enabling the implementation and evaluation of the KNN classifier for predicting customer responses to special offers.

3.3] Theory-

* Data Preparation:
  + Load the dataset using Pandas to a DataFrame, ensuring it contains relevant customer details and responses to special offers.
  + Preprocess the data by handling missing values, encoding categorical variables, and splitting the dataset into features (input variables) and target variables (customer response).
* Feature Scaling:
  + Standardize the features using StandardScaler from Scikit-learn to ensure all features are on the same scale, preventing any particular feature from dominating the model training process.
* Train-Test Split:
  + Split the dataset into training and testing sets using train\_test\_split from Scikit-learn. This ensures the model is trained on one portion of the data and evaluated on another to assess its generalization performance.
* Model Selection and Training:
  + Apply the K-Nearest Neighbors (KNN) algorithm from Scikit-learn to predict customer responses to special offers. KNN is chosen for its simplicity and effectiveness in classification tasks.
  + Train the KNN classifier on the training data, tuning hyperparameters such as the number of neighbors (K) if necessary.

**Support Vector Machine (SVM):**

* Supervised Learning Algorithm: SVM is a supervised learning algorithm, meaning it requires labelled data for training. It learns to classify data into different categories based on the features provided.
* Classification and Regression: SVM can be used for both classification and regression tasks. In classification, it separates data points into different classes, while in regression, it predicts a continuous outcome.
* Hyperplane Separation: The primary goal of SVM is to find the hyperplane that best separates different classes in the feature space. This hyperplane has the maximum margin, which is the distance between the hyperplane and the nearest data points of each class.
* Effective in High-dimensional Spaces: SVM is effective in high-dimensional spaces, where the number of dimensions exceeds the number of samples. It can handle complex datasets with many features.
* Kernel Trick: SVM uses a kernel function to map the input data into a high-dimensional feature space. This allows it to find nonlinear decision boundaries in the original feature space**.**

3.4] Applications:

* Customer Relationship Management: Predicting customer responses to special offers can aid in personalized marketing strategies, enhancing customer satisfaction and loyalty.
* Sales Optimization: Understanding customer preferences enables the optimization of product offerings and promotional campaigns, leading to increased sales and revenue.

3.5] Limitations:

* Data Quality: The effectiveness of the predictive model heavily relies on the quality and representativeness of the dataset. Inaccurate or biased data may lead to unreliable predictions.
* Model Complexity: While KNN is simple and intuitive, it may not perform optimally with large datasets or in high-dimensional feature spaces. Other algorithms like Decision Trees or Random Forests could be explored for better performance in such scenarios.

3.6] Working:

* SVM maps the input data into a high-dimensional feature space using a kernel function.
* It then finds the hyperplane that best separates the data points of different classes.
* The hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points (support vectors) of each class.
* During prediction, SVM classifies new data points based on which side of the hyperplane they fall on.

3.7] Conclusion:

By following this methodology and understanding the applications and limitations of the assignment, practitioners can effectively apply machine learning techniques to predict customer responses in a cosmetics shop setting, leveraging the capabilities of libraries like Pandas, NumPy, and Scikit-learn.

**ASSIGNMENT NO: 4**

**K-Means Clustering**

4.1] Problem Statement -

a) Apply Data pre-processing (Label Encoding, Data Transformation) techniques if necessary.

b) Perform data preparation (Train-Test Split)

c) Apply Machine Learning Algorithm

d) Evaluate Model.

e) Apply Cross-Validation and Evaluate Model

4.2] S/W Packages and Libraries used:

For the following assignment, the interpreter used was Google Collab and the Primary Library used were Pandas, Numpy, Matplotlib, and sklearn which we have discussed earlier.

4.3] Theory-

* Data Pre-processing:
* If the dataset contains categorical variables, apply Label Encoding or One-Hot Encoding using libraries like scikit-learn's preprocessing module to convert them into numerical representations suitable for machine learning algorithms.
* Data Preparation (Train-Test Split):
* Split the pre-processed dataset into training and testing sets using train\_test\_split from the scikit-learn library. This ensures that the model's performance is evaluated on unseen data.
* Apply Machine Learning Algorithm:
* Choose an appropriate machine learning algorithm based on the nature of the problem (classification, regression, etc.) and the characteristics of the dataset. Common algorithms include Decision Trees, Random Forests, Support Vector Machines (SVM), or Gradient Boosting Machines (GBM).
* Evaluate Model:
* Evaluate the trained model's performance on the testing dataset using evaluation metrics such as accuracy, precision, recall, F1-score, or mean squared error (for regression).
* Visualize the model's performance using appropriate plots such as confusion matrix, ROC curve, or learning curve.
* Apply Cross-Validation and Evaluate Model:
* Apply cross-validation techniques such as k-fold cross-validation to assess the model's generalization performance and robustness.
* Compute cross-validated performance metrics to get a more reliable estimate of the model's performance.

4.4] Applications:

* Predictive Analytics: By applying machine learning algorithms to pre-processed data, businesses can make predictions and identify patterns in their datasets, leading to informed decision-making.
* Customer Segmentation: Machine learning models can help segment customers based on their behavior or characteristics, allowing businesses to tailor marketing strategies and offerings to specific customer groups.

4.5] Limitations:

* Data Quality: The effectiveness of machine learning models heavily relies on the quality and representativeness of the data. Poor-quality or biased data can lead to inaccurate predictions and unreliable model performance.
* Model Selection: Choosing the most suitable machine learning algorithm for a given problem requires domain knowledge and experimentation. There is no one-size-fits-all solution, and different algorithms may perform differently on different datasets.

4.6] Working:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassigning each datapoint to the new closest centroid of each cluster.

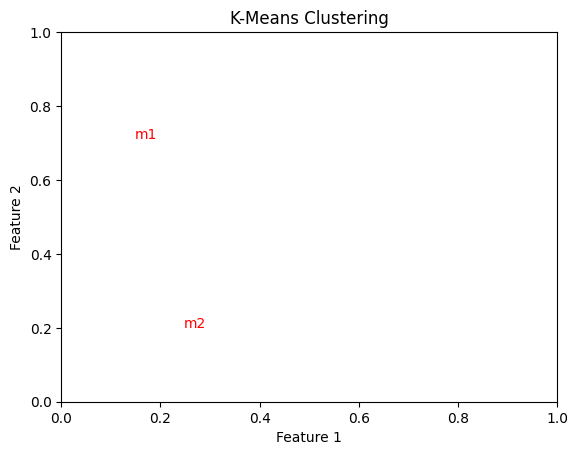
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready

**Diagram:**







4.7] Conclusion:

By adhering to this methodology and understanding the assignment's applications and limitations, practitioners can effectively apply machine learning techniques to preprocess data, prepare it for modeling, train machine learning models, evaluate their performance, and assess their generalization capabilities through cross-validation.

**ASSIGNMENT NO: 5**

**Data Visualization**

5.1] Problem Statement -

Visualize the data using Python by plotting the graphs for assignment no. 1 and 2. Consider a suitable data set.

a) Use Scatter plot, bar plot, Box plot and Histogram.

5.2] S/W Packages and Libraries used:

For the following assignment, the interpreter used was Google Collab and the Primary Library used was

* Seaborn: Seaborn is a statistical data visualization library in Python that provides a high-level interface for creating attractive and informative visualizations. It works well with Pandas DataFrames and offers a variety of plot types and customization options.

5.3] Theory-

* Data Selection:
  + Choose a suitable dataset that contains numerical and categorical variables, allowing for the creation of different types of plots such as scatter plots, bar plots, box plots, and histograms.
* Data Preprocessing:
  + Load the dataset into Python using Pandas.
  + Perform any necessary data cleaning and preprocessing steps, such as handling missing values or encoding categorical variables.
* Visualization:
* Utilize the Seaborn library for creating visualizations.
* Create scatter plots to visualize the relationship between two numerical variables.

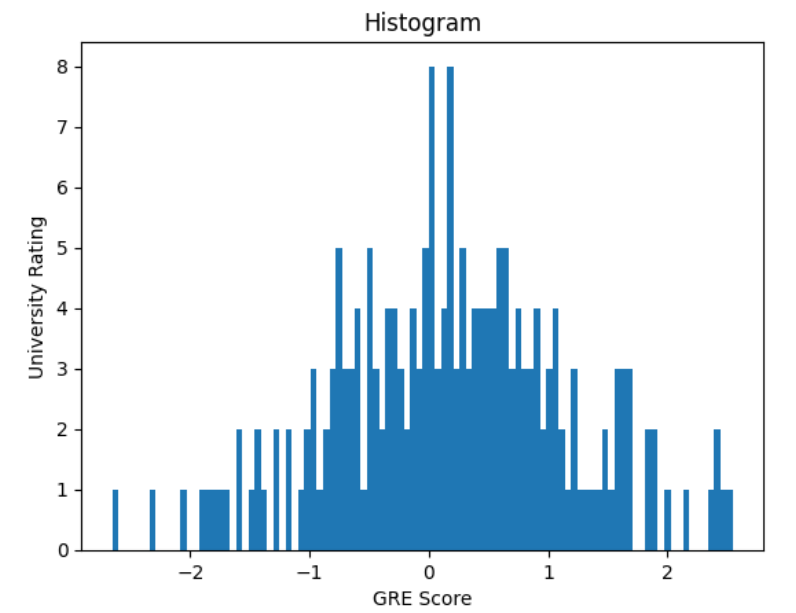
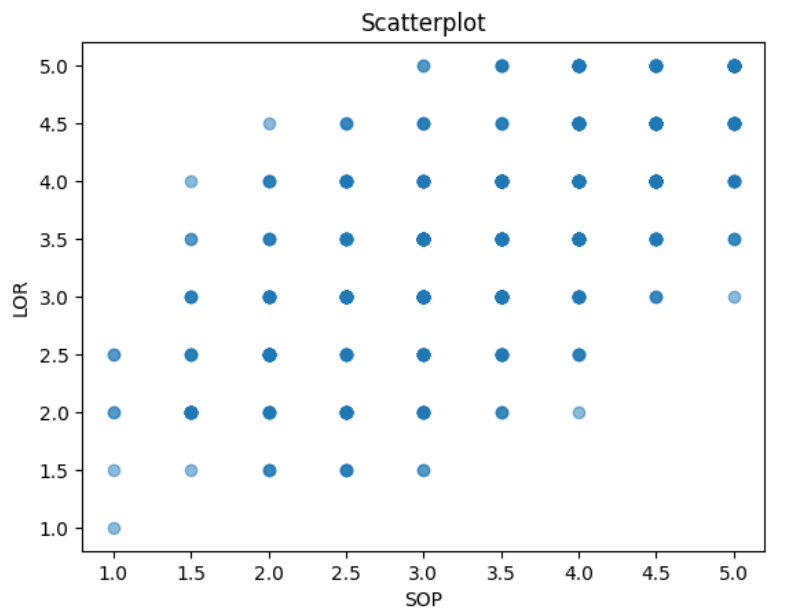
5.4] Applications:

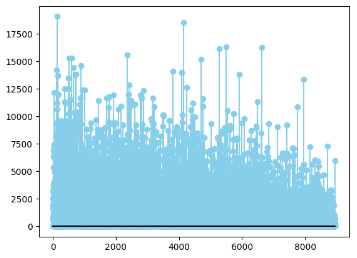
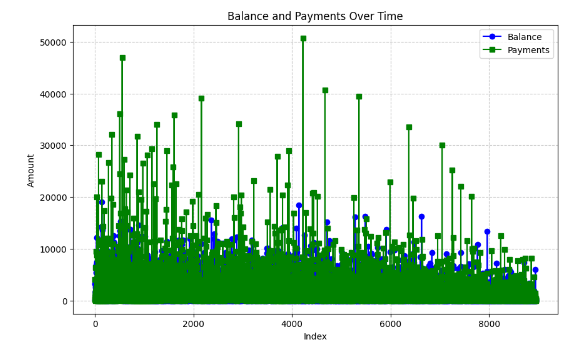
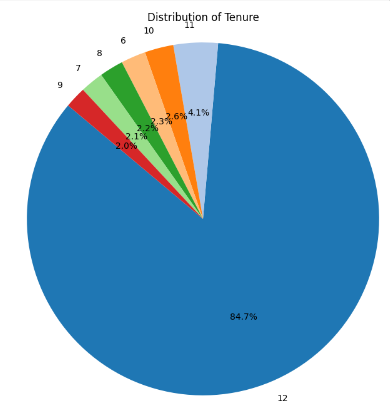
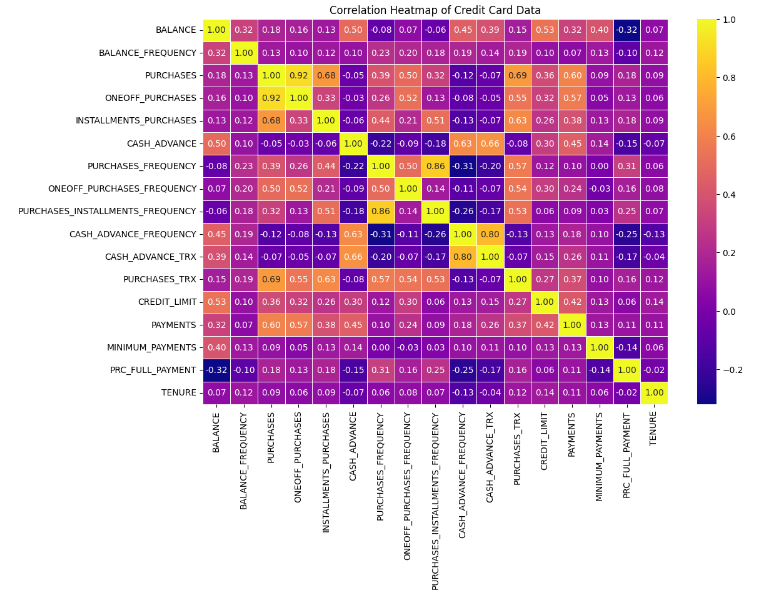
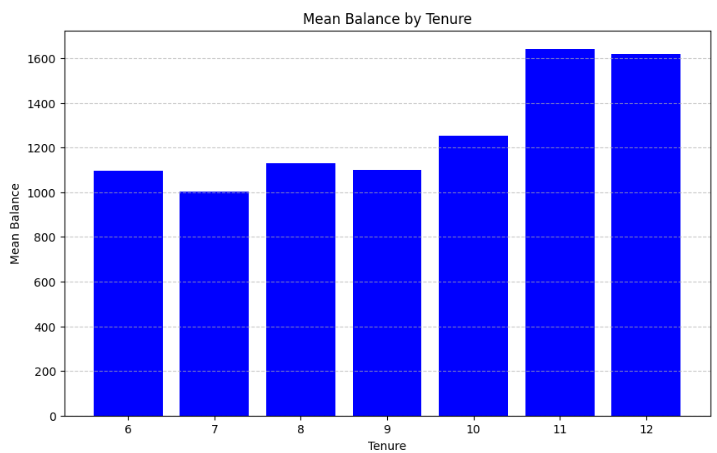
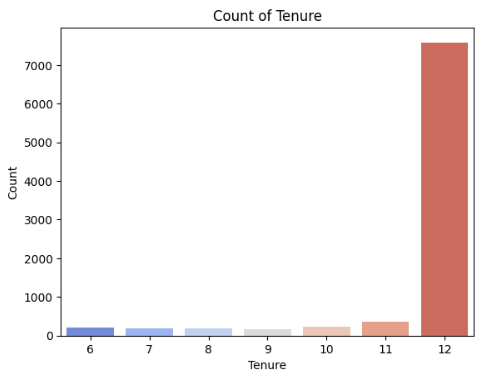
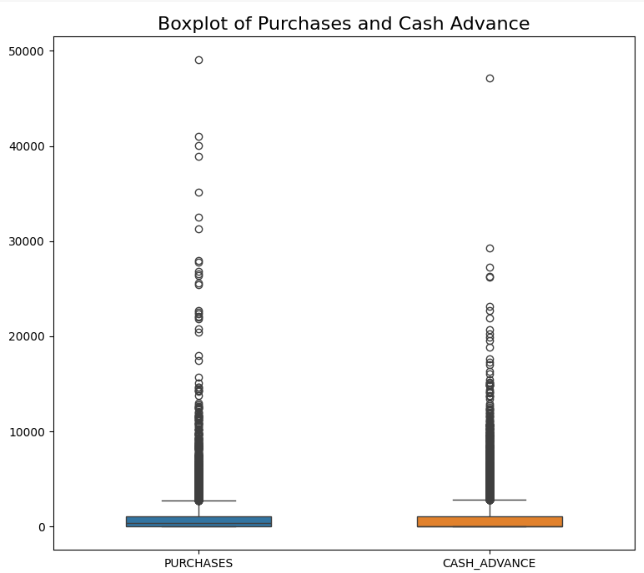
* Exploratory Data Analysis (EDA): Visualization techniques provide insights into the underlying structure and patterns within the data, aiding in understanding relationships and identifying trends.
* Data Communication: Visualizations serve as powerful tools for communicating findings and insights to stakeholders or decision-makers in a clear and concise manner.

5.5] Limitations:

* Data Quality: Visualizations are only as good as the quality of the underlying data. Inaccurate or incomplete data may lead to misleading visualizations and erroneous conclusions.
* Interpretation: While visualizations can highlight patterns and relationships in the data, they may not always imply causation. Care must be taken to interpret visualizations within the context of the data and domain knowledge.

5.6] Screenshots of output





5.7] Conclusion:

By following the outlined methodology and utilizing the Seaborn library, you can effectively visualize the selected dataset using scatter plots, bar plots, box plots, and histograms. These visualizations will help in exploring relationships, understanding distributions, and identifying patterns within the data, thereby facilitating further analysis and decision-making processes.

**ASSIGNMENT NO: 6**

**Regression technique**

6.1] Problem Statement-

Download the temperatures dataset from following link:

<https://www.kaggle.com/datasets/venky73/temperatures-of-india>.

This data consists of temperatures of INDIA averaging the temperatures of all places month wise. Temperatures values are recorded in CELSIUS.

a) Apply Linear Regression using suitable library function and predict the Month-wise

temperature.

b) Assess the performance of regression models

6.2] S/W Packages and Libraries used-

Software Package: Python

Libraries Used:

pandas: For data manipulation and analysis.

scikit-learn: For implementing machine learning algorithms, including Linear Regression.

matplotlib: For data visualization.

6.3] Theory-

Linear Regression

It's a statistical technique for forecasting analysis. Predictions are made using linear regression for continuous, real, or numerical variables like sales, earnings, age, and product price, among others.   
The term "linear regression" refers to a procedure that displays a linear relationship between one or more independent (y) variables and a dependent (y) variable.   
Given that linear regression displays a linear relationship, it can be used to determine how the value of the independent variable affects the value of the dependent variable.

Linear Regression Types:

Basic Linear Regression:   
A linear regression procedure is referred to as simple linear regression if it uses one independent variable to predict the value of a number of dependent variables.

Multiple Linear Regression: This type of linear regression method is employed when multiple independent variables are combined to predict the value of a numerical dependent variable.

Applications of Simple Linear Regression:

1. Student grades determined by the number of hours studied (ideally):In this case, exam scores are dependent on the number of hours studied, but the number of hours studied is independent.
2. Estimating agricultural yields using rainfall data: The measure of precipitation is an independent variable, and yield is a dependent variable.
3. Estimating an individual's salary based on years of experience: Experience is now the independent variable, and salary is the dependent variable.

Limitations of Simple Linear Regression:

1. Assumes linearity: Regression models, particularly linear regression, assume a straight-line relationship between the independent and dependent variables. If the underlying relationship is more complex (curved, exponential, etc.), the model may not accurately capture the true association.
2. Sensitive to outliers: Outliers (data points significantly different from the majority) can disproportionately influence the regression line, leading to misleading results.
3. Doesn't establish causation: Even if a strong correlation is found between variables, regression models cannot determine causality. There might be a third, unseen variable influencing both the independent and dependent variables, creating a false association.

* Data Preprocessing:
  + Load the temperature data.
  + Prepare the data by separating features (months) and target variable (monthly temperatures).
  + Split the data into training and testing sets.
* Model Training:
  + Initialize a Linear Regression model.
  + Fit the model on the training data.
* Model Evaluation:
  + Predict the month-wise temperatures using the trained model.
  + Evaluate the model's performance using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-Square metrics.

6.4] Application:

* The application of this regression technique is to predict the month-wise temperatures in India based on historical data.
* It can be used by meteorologists, climatologists, and environmental scientists for various purposes such as weather forecasting, climate trend analysis, and planning for agriculture and infrastructure.

6.5] Limitations:

* Linear Regression assumes a linear relationship between the independent and dependent variables, which may not always hold true in real-world scenarios.
* It may not capture complex patterns or non-linear relationships present in the data.
* The accuracy of predictions can be affected by outliers and noise in the data.
* The model's performance heavily depends on the quality and representativeness of the training data.

6.6] Working:

Linear regression is a fundamental technique in statistics and machine learning used to model the relationship between a dependent variable and one or more independent variables. The goal is to create a linear model that predicts the dependent variable based on the independent variables.

Here's how a linear regression model works for prediction:

* Importing necessary libraries and modules: They provide pre-built functionalities and extend your program's capabilities
* Data Collection: Collect data on the variables of interest. For example, in a simple linear regression, you would have one independent variable (for eg. Year here) and one dependent variable (for eg. Temperature here).
* Data Preprocessing and EDA: This step involves cleaning the data and analysing it intricately.
* Splitting the Data: Split the dataset into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate its performance.
* Model Training: Use the training data to fit a linear regression model. The model tries to find the best-fitting linear relationship between the independent and dependent variables. In simple linear regression, this relationship is represented by a line (y = mx + b), where m is the slope and b is the intercept.
* Making Predictions: Once the model is trained, use it to make predictions on the testing data. The model calculates the predicted values of the dependent variable based on the values of the independent variable(s).
* Evaluating the Model: Evaluate the model's performance using metrics such as mean squared error (MSE) or R-squared. These metrics measure how well the model's predictions match the actual values in the testing data.

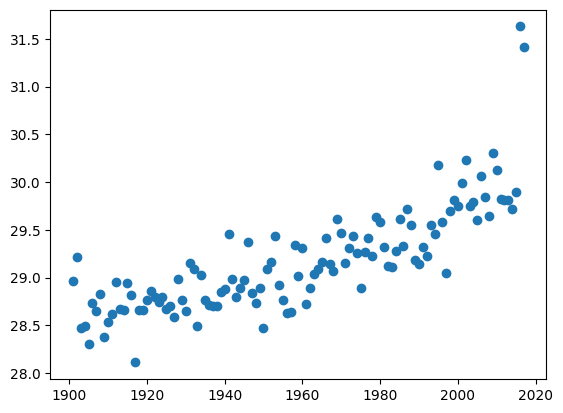
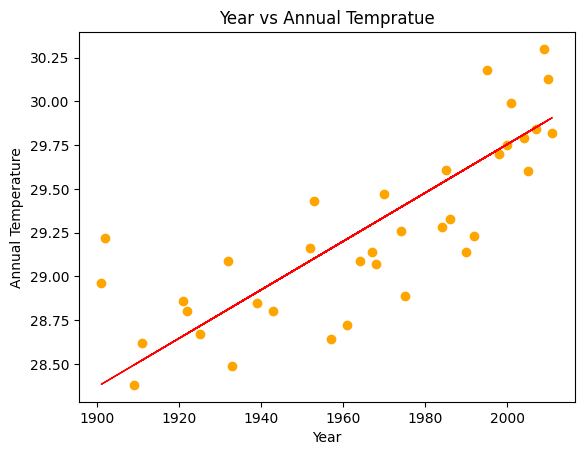


fig.Plot depicting output of the regression model.(year vs annual temperature)

fig.Scatter plot(visualization on dataset)

6.7] Conclusion:

* Linear Regression can be effectively used to predict month-wise temperatures in India.
* The model's performance can be assessed using metrics like MSE, MAE, and R-Square.
* Despite its limitations, Linear Regression provides a simple and interpretable approach for temperature prediction, which can be valuable for various applications in climate science and related fields.

**ASSIGNMENT NO: 7**

**Decision Tree**

7.1] Problem Statement-

To implement Classification techniques for following scenario:

Every year many students give the GRE exam to get admission in foreign Universities. The

data set contains GRE Scores (out of 340), TOEFL Scores (out of 120), University Rating

(out of 5), Statement of Purpose strength (out of 5), Letter of Recommendation strength (out

of 5), Undergraduate GPA (out of 10), Research Experience (0=no, 1=yes), Admitted (0=no,

1=yes). Admitted is the target variable.

The counselor of the firm is supposed to check whether the student will get an admission or

not based on his/her GRE score and Academic Score. So to help the counselor to take

appropriate decisions, build a machine learning model classifier using a Decision tree to

predict whether a student will get admission or not.

a) Apply Data pre-processing (Label Encoding, Data Transformation....) techniques if

necessary.

b) Perform data-preparation (Train-Test Split)

c) Apply Machine Learning Algorithm

d) Evaluate Model.

7.2] S/W Packages and Libraries used-

Software Package: Python

Libraries Used:

pandas: For data manipulation and analysis.

scikit-learn: For implementing machine learning algorithms, including Decision Tree Classifier.

matplotlib: For data visualization.

7.3] Theory-

**Classification**: Classification is the process of organizing a dataset into classes or categories. This can be applied to both structured and unstructured data, with the goal of predicting the class of given data points based on their features.

**Decision Tree:** A Decision Tree is a predictive model that utilizes a tree-like structure to represent decisions and their potential consequences. It starts with a root node and branches out into decision nodes, which are then further split into leaf nodes. Each node represents a decision based on a feature, and the leaves represent the final outcome or prediction.

**Entropy**: Entropy is a measure of the randomness or impurity in a dataset. In the context of Decision Trees, entropy is used to quantify the homogeneity of a sample. A sample with low entropy is more homogeneous, while a sample with high entropy is more diverse.

**Constructing a Decision Tree:**

1. Calculate the entropy of the target variable.

2. Split the dataset based on different attributes and calculate the entropy for each branch.

3. Calculate the information gain for each attribute and select the attribute with the highest information gain as the decision node.

4. Continue this process recursively until all data is classified, with branches either becoming leaf nodes or further split.

**Pruning:** Pruning is a technique used to prevent overfitting in Decision Trees by removing nodes or branches that are not significant. It helps improve the performance of the tree by eliminating unnecessary complexity. Pruning can be done during tree construction (pre-pruning) or after the tree is built (post-pruning).

we will utilize these principles of Decision Trees, including entropy calculation, information gain, tree construction, pruning, and transformation to decision rules, to develop a predictive model for determining student admissions to foreign universities based on GRE scores and academic performance.

* Data Preprocessing:
  + Load the data.
  + Perform any necessary data transformations (e.g., label encoding for categorical variables).
  + Check for missing values and handle them if necessary.
* Data Preparation (Train-Test Split):
  + Split the data into training and testing sets to train the model on one set and evaluate its performance on another.
* Model Training:
  + Initialize a Decision Tree Classifier model.
  + Fit the model on the training data.
* Model Evaluation:
  + Predict admission outcomes for the test data.
  + Evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score.
  + Visualize the results if necessary.

7.4] Applications:

* The application of this classification technique is to predict whether a student will get admission to a foreign university based on their GRE score and academic performance.
* It can be used by education consultants, university admission offices, and students themselves to assess their chances of admission and make informed decisions.

7.5] Limitations-

* Decision Tree models are prone to overfitting, especially on noisy data or data with a large number of features.
* They may not capture complex relationships between variables as effectively as other algorithms.
* Decision Trees are sensitive to small variations in the data, which can lead to different tree structures and results.
* The interpretability of the model might be limited if the tree becomes too large and complex.

7.6] Working-

Step 1: Initialization

* Select a decision tree algorithm.
* Instantiate the decision tree classifier with specified parameters.

Step 2: Model Training

* Train the decision tree classifier using the training dataset (x\_train, y\_train).
* The algorithm recursively partitions the feature space based on the target variable to create a tree structure.

Step 3: Prediction

* For each instance in the testing dataset (x\_test):
* Traverse the decision tree by following the learned rules.
* Determine the predicted class based on the final leaf node reached.

Step 4: Evaluation

* + Compare the predicted labels with the true labels from the testing dataset to assess model performance.
  + Calculate evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

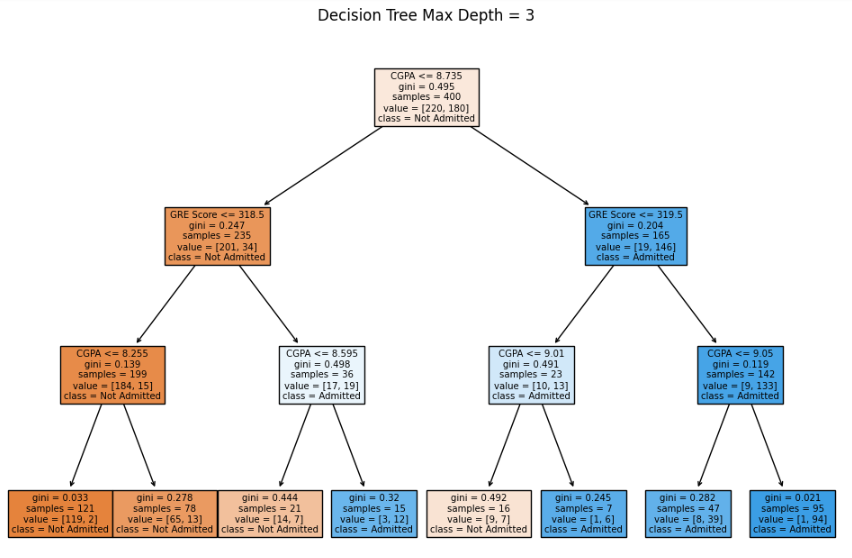
Step 5: Interpretation

* + Visualize the decision tree graphically to understand the rules learned by the model.
  + Analyze feature importance to identify the most influential features in decision-making.

Step 8: The model is ready.

**Diagram:**

****



7.7] Conclusion-

* The Decision Tree Classifier can be used effectively to predict admission outcomes based on GRE scores and academic performance.
* It provides a simple and interpretable model that can help counselors and students make informed decisions about university admissions.
* However, it's important to be aware of its limitations and consider other machine learning techniques for more accurate predictions, especially in complex scenarios with a large number of features.