Assignment No:- 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

Software Library Package:

We'll use Python with the pandas library for data manipulation and analysis.

Theory:

i) Methodology:

Pandas is a powerful library for data manipulation and analysis in Python. It provides data structures like DataFrame and Series which are ideal for handling structured data. Here's a brief overview of the operations:

a) Reading Data: Pandas provides functions like `read_csv()` and `read_excel()` to read data from CSV and Excel files respectively.

b) Indexing and Selecting Data: You can use indexing and selection techniques like `loc[]` and `iloc[]` to select specific rows and columns from the DataFrame.

c) Sorting Data: The `sort_values()` function can be used to sort the data based on one or more columns.

d) Describing Attributes and Data Types: Functions like `info()`, `describe()`, and `dtypes` provide information about the DataFrame's attributes and data types of columns.

e) Counting Unique Values and Converting Data Types: `value_counts()` can be used to count unique values, and `astype()` can be used to convert data types.

f) Identifying and Filling Missing Values: Functions like `isna()`, `fillna()`, and `dropna()` help in identifying and filling missing values in the DataFrame.

ii) Advantages and Applications:

- Advantages: Pandas simplifies data manipulation and analysis tasks with its intuitive syntax and powerful functions. It integrates well with other libraries in the Python ecosystem such as NumPy and Matplotlib.
- Applications: Pandas is widely used in data preprocessing, data cleaning, exploratory data analysis, and feature engineering tasks in data science projects.

Limitations/Example:

Pandas may struggle with large datasets due to its in-memory processing nature. For very large datasets, alternative libraries like Dask or PySpark may be more suitable.

Working/Algorithm:

a) Reading Data:

Assume we have data stored in a CSV file named "employee_data.csv" containing information about employees such as their ID, name, department, and salary. We can use the `read_csv()` function from pandas to read this data into a DataFrame.

b) Indexing and Selecting Data:

After reading the data, we might want to select specific rows based on certain conditions, for example, employees belonging to the "Sales" department. We can use boolean indexing or the `query()` function to achieve this.

c) Sorting Data:

To analyze employee salaries effectively, we may need to sort the data based on the "Salary" column to identify the highest-paid employees. We can use the `sort_values()` function for this task.

d) Describing Attributes and Data Types:

Once we have the DataFrame, we can use methods like `info()` to get an overview of the data, `describe()` to obtain descriptive statistics, and `dtypes` attribute to check the data types of each column.

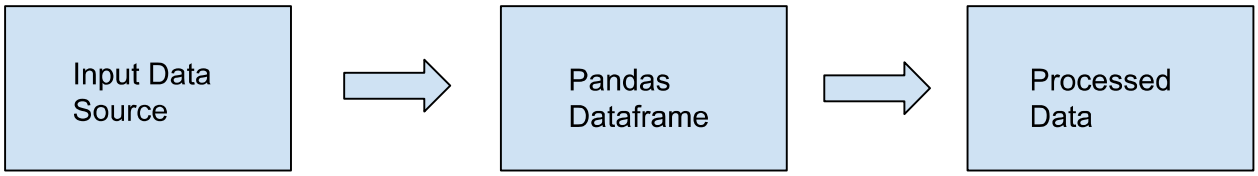
e) Counting Unique Values and Converting Data Types:

We may want to count the number of unique departments or employee IDs in the dataset. We can use the `value_counts()` function for this task. Additionally, if we need to convert data types, we can use the `astype()` method.

f) Identifying and Filling Missing Values:

During data analysis, we may encounter missing values in certain columns, such as "Salary" or "Department". We can use functions like `isna()` or `isnull()` to identify missing values, and then fill them using `fillna()` or `dropna()` as appropriate.

Diagram:



Conclusion:

By following these steps and utilizing pandas functionalities, we can effectively read, manipulate, and analyze data stored in various formats like Excel. Pandas provides a versatile toolkit for data handling in Python, making it a powerful choice for data scientists and analysts.

Assignment No:- 2

Problem Statement:

Perform the following operations using R/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)

Software Library Package:

For this task, we'll use the following Python libraries:

- `pandas` for data manipulation and analysis.
- `matplotlib` for data visualization.
- `scikit-learn` for data cleaning, transformation, and model building.

Theory:

i) Methodology:

a) Computing Summary Statistics: Summary statistics provide a concise summary of the dataset's characteristics, including measures like mean, median, standard deviation, variance, and percentiles.

b) Data Visualization (Histograms): Histograms visualize the distribution of data values for each feature, aiding in understanding the data's distribution and identifying patterns.

c) Data Cleaning, Integration, Transformation, and Model Building: These steps are essential in the data analysis pipeline. Data cleaning involves handling missing values and outliers. Integration combines multiple datasets if applicable. Transformation involves scaling or encoding features. Model building refers to creating predictive models for tasks like classification.

ii) Advantages and Applications and Limitations/Example:

- Advantages: Summary statistics and histograms provide quick insights into the dataset's characteristics and distribution. Data cleaning, integration,

transformation, and model building are crucial for accurate analysis and prediction.
- Applications: These techniques are widely used in various fields such as finance, healthcare, marketing, etc., for tasks like risk assessment, customer segmentation, etc.
- Limitations/Example: Summary statistics may not capture the full complexity of data. Histograms may not reveal patterns in sparse data. Data cleaning and integration can be time-consuming. Incorrect transformation or model selection can lead to inaccurate results.

Working/Algorithm:

a) Computing Summary Statistics:

To compute summary statistics for each feature in the dataset:

- Calculate the minimum value, maximum value, mean, range, standard deviation, variance, and percentiles for each feature.
- Use statistical functions provided by libraries like `pandas` or `numpy` to compute these statistics.

b) Data Visualization - Creating Histograms:

To create histograms for each feature in the dataset:

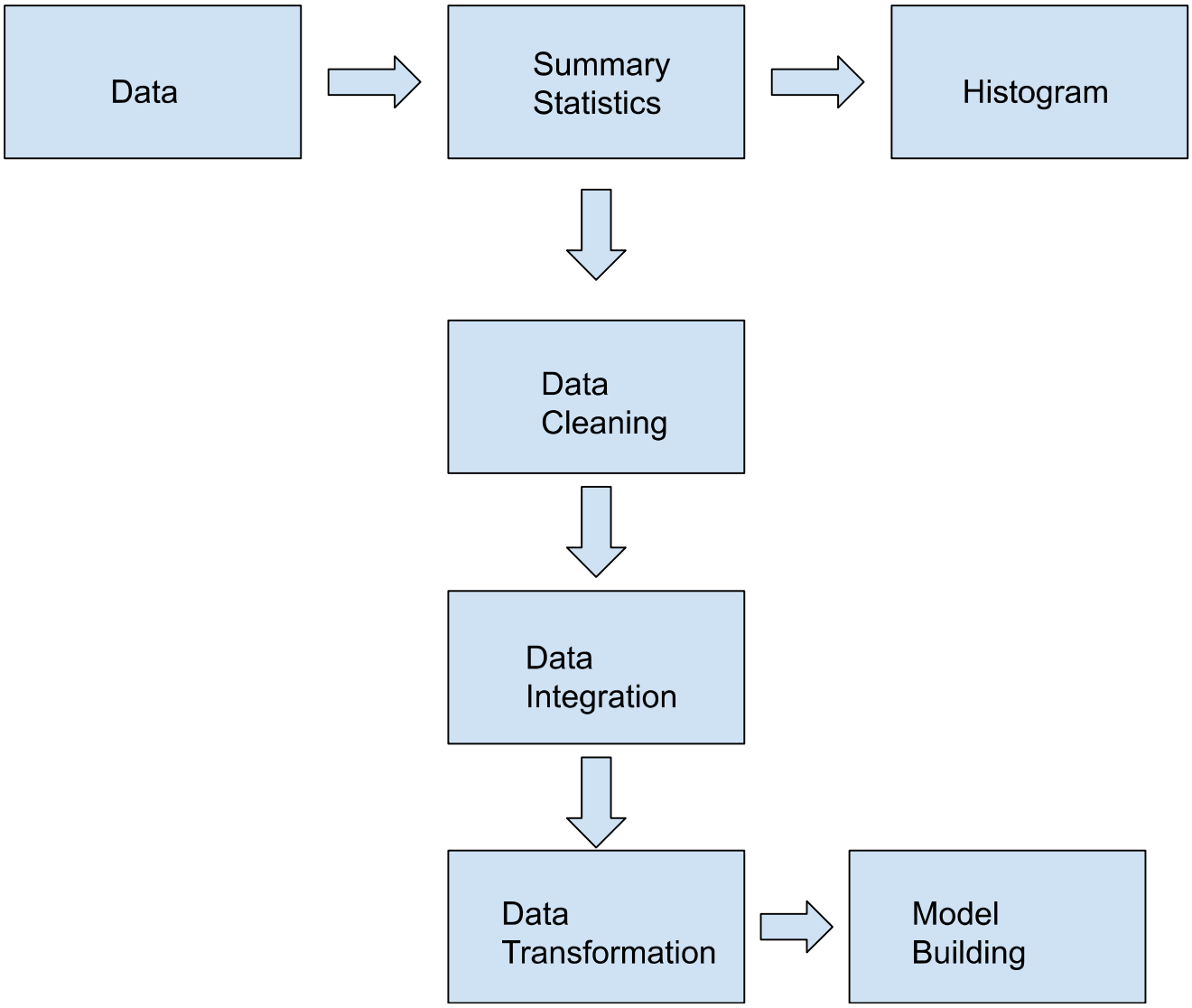
- Use a plotting library such as `matplotlib` to generate histograms.
- Plot the frequency distribution of each feature by dividing the data into bins and counting the number of occurrences in each bin.
- Customize the histograms by adjusting bin width, color, labels, etc.

c) Data Cleaning, Integration, Transformation, and Model Building:

For data cleaning, integration, transformation, and model building:

- Data cleaning involves handling missing values, duplicates, and outliers.
- Data integration combines multiple datasets if applicable.
- Data transformation may include scaling numerical features, encoding categorical variables, or creating new features.
- Model building involves selecting an appropriate machine learning algorithm, training the model on the data, and evaluating its performance.

Diagram:



Conclusion:

Performing operations such as computing summary statistics, creating histograms for data visualization, and conducting data cleaning, integration, transformation, and model building are essential steps in the data analysis process. These steps help in gaining insights from data, identifying patterns, and building predictive models for various applications. Choosing appropriate methodologies and tools ensures efficient and accurate data analysis and model building.

Assignment No:- 3

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.

Software Library Package:

For this task, we'll use the following Python libraries:

- `pandas` for data manipulation and analysis.
- `scikit-learn` for machine learning algorithms.
- `matplotlib` and `seaborn` for data visualization.

Theory:

i) Methodology:

For this problem, we can follow these steps:

Data Collection: Gather customer details such as age, gender, income, purchase history, etc.
Data Preprocessing: Clean the data by handling missing values, encoding categorical variables, and scaling numerical features if necessary.
Feature Selection: Identify relevant features that influence customer response to special offers.
Model Selection: Choose an appropriate machine learning algorithm such as logistic regression, decision tree, random forest, or gradient boosting for classification.
Model Training: Train the selected model on the training dataset.
Model Evaluation: Evaluate the model's performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.
Prediction: Make predictions on new data to predict customer response to special offers.

ii) Advantages and Applications and Limitations/Example:

- Advantages: Predicting customer response to special offers can help businesses optimize marketing strategies, improve customer satisfaction, and increase sales. Machine learning algorithms enable automated decision-making based on historical data.

- Applications: This approach can be applied in various industries such as retail, e-commerce, hospitality, etc., to personalize marketing campaigns and improve customer engagement.
- Limitations/Example: The accuracy of predictions depends on the quality and quantity of data collected. Overfitting can occur if the model is too complex or trained on insufficient data. For example, if the dataset lacks diversity or representative samples, the model's predictions may be biased or inaccurate.

Working/Algorithm:

a) Data Collection and Preprocessing:

- Data Collection: Gather customer details such as age, gender, income, purchase history, etc., from the cosmetics shop database.
- Data Preprocessing: Clean the data by handling missing values, encoding categorical variables, and scaling numerical features if necessary. This ensures the data is in a suitable format for machine learning algorithms.

b) Feature Selection:

Identify relevant features that may influence customer response to special offers. Features may include demographic information, purchase behavior, frequency of visits, etc. Feature selection helps in reducing dimensionality and improving model performance.

c) Model Selection and Training:

Choose an appropriate machine learning algorithm for classification tasks, such as logistic regression, decision tree, random forest, or gradient boosting. Train the selected model on the preprocessed dataset using appropriate training techniques like cross-validation.

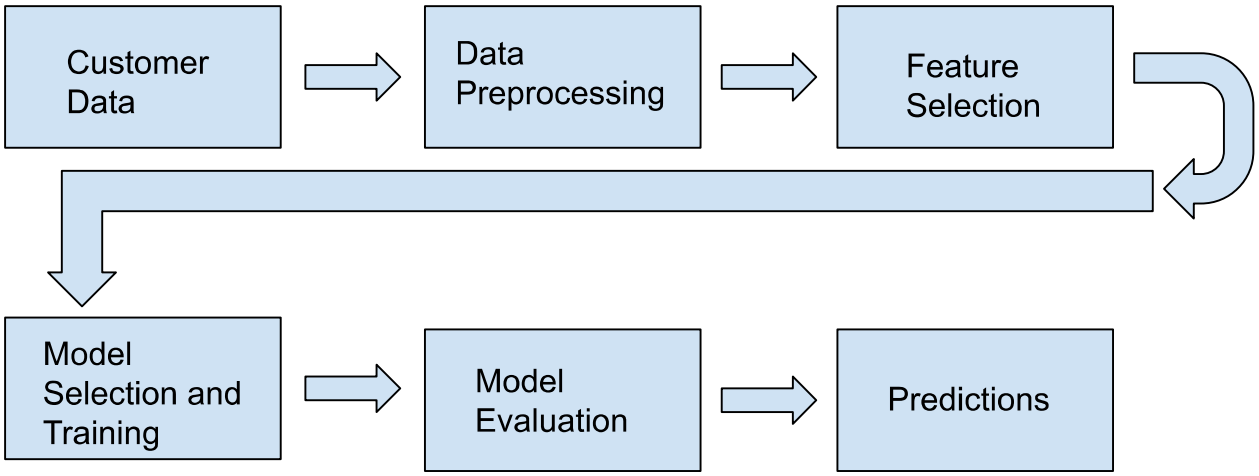
d) Model Evaluation:

Evaluate the trained model's performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC. This step helps in assessing how well the model predicts customer responses to special offers and identifying areas for improvement.

e) Prediction:

Make predictions on new data to predict customer response to special offers. Use the trained model to classify customers into different response categories, such as "likely to respond" or "unlikely to respond," based on their features.

Diagram:



Conclusion:

Predicting customer response to special offers using machine learning algorithms based on customer details collected can help cosmetics shops tailor their marketing strategies and increase customer engagement. By leveraging historical customer data, businesses can make informed decisions and improve the effectiveness of their marketing campaigns. It's essential to ensure data quality, model performance, and ethical considerations when implementing such predictive analytics solutions.

Assignment No:- 4

Problem Statement:

Write a program to do following:

We have given a collection of 8 points. P1=[0.1,0.6] P2=[0.15,0.71] P3=[0.08,0.9] P4=[0.16, 0.85] P5=[0.2,0.3] P6=[0.25,0.5] P7=[0.24,0.1] P8=[0.3,0.2]. Perform the k-mean clusterin with initial centroids as m1=P1=Cluster#1=C1 and m2=P8=cluster#2=C2.

Answer the following:

a) Which cluster does P6 belong to?

b) What is the population of a cluster around m2?

c) What is the updated value of m1 and m2?

Software Library Package:

For this task, we'll use the following Python library:

- `numpy` for numerical computations.
- `scikit-learn` for machine learning algorithms.
- `matplotlib` and `seaborn` for data visualization.

Theory:

i) Methodology:

a) K-means Clustering: K-means is a partitioning algorithm that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (centroid).

b) Algorithm Steps:

` Initialize k centroids randomly.
Assign each data point to the nearest centroid.
Update the centroids based on the mean of the points assigned to each cluster.
Repeat steps 2 and 3 until convergence.

c) Advantages: K-means is simple, easy to implement, and computationally efficient for large datasets.

d) Applications: K-means clustering is widely used in various fields such as customer segmentation, image segmentation, anomaly detection, etc.

e) Limitations/Example: K-means may converge to local optima and may not perform well on non-linear or irregularly shaped clusters.

Working/Algorithm:

Initialize centroids: m1 = P1 = [0.1, 0.6], m2 = P8 = [0.3, 0.2].
Assign points to clusters:

- Assign each point to the nearest centroid:
  - P1, P2, P3, P4, and P5 are closer to m1 (C1).
  - P6, P7, and P8 are closer to m2 (C2).

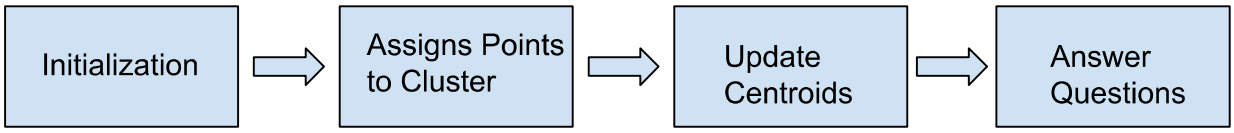
Update centroids:

- Calculate the mean of points in each cluster:
  - For C1: (0.1+0.15+0.08+0.16+0.2)/5 = 0.138, (0.6+0.71+0.9+0.85+0.3)/5 = 0.672.
  - For C2: (0.25+0.24+0.3)/3 = 0.263, (0.5+0.1+0.2)/3 = 0.267.
- Updated centroids: m1 = [0.138, 0.672], m2 = [0.263, 0.267].

Answering questions:

- a) P6 belongs to Cluster C2.
- b) The population of Cluster C2 around m2 is 3.
- c) Updated values of m1 and m2 are [0.138, 0.672] and [0.263, 0.267] respectively.

Diagram:



Conclusion:

K-means clustering is a powerful unsupervised learning algorithm used for clustering data into groups based on similarity. By iteratively updating centroids and assigning points to clusters, it efficiently partitions the data. However, the choice of initial centroids can significantly affect the clustering results. Understanding the algorithm's working principles and its advantages and limitations is essential for its effective application in various domains.

Assignment No:- 5

Problem Statement:

Visualize the data using R/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
OR
b) Perform the data visualization operations using Tableau for the given dataset.

Software Library Package:

For this task, we'll use the following Python libraries:

- `pandas` for data manipulation.
- `matplotlib` and `seaborn` for data visualization.

Theory:

i) Methodology:

- Scatter Plot: A scatter plot is used to visualize the relationship between two variables. Each point represents an observation in the dataset, with one variable on the x-axis and the other on the y-axis.
- Bar Plot: A bar plot is used to visualize the distribution of a categorical variable or the relationship between a categorical variable and a numerical variable. It represents the frequency or mean of each category.
- Box Plot: A box plot is used to visualize the distribution of a numerical variable and identify outliers. It shows the quartiles of the dataset, including the median, first quartile, third quartile, and outliers.
- Histogram: A histogram is used to visualize the distribution of a numerical variable. It divides the range of values into bins and shows the frequency of observations in each bin.

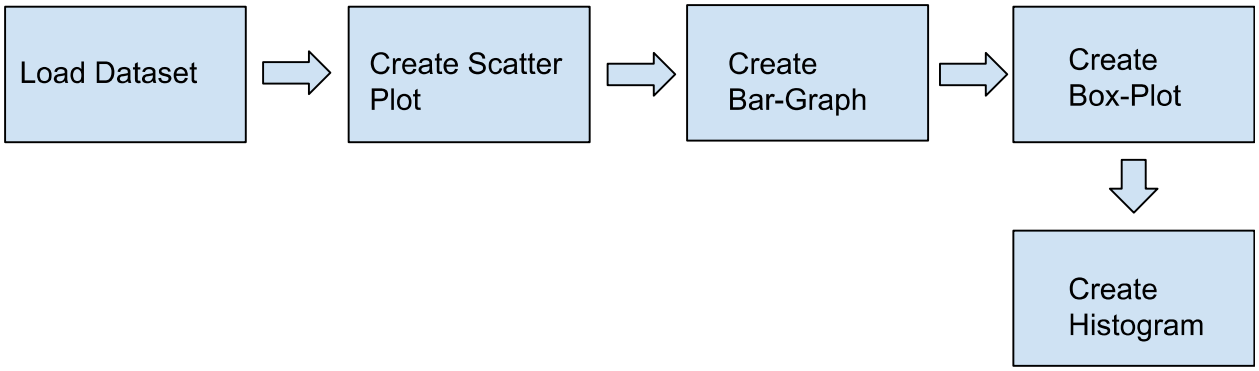
ii) Advantages and Applications and Limitations/Example:

- Advantages: Python libraries such as `matplotlib` and `seaborn` offer a wide range of customizable plots for effective data visualization. Visualization enhances data exploration, analysis, and communication of insights to stakeholders.
- Applications: Data visualization is widely used in various domains such as finance, healthcare, marketing, etc., for exploratory data analysis, presentation of results, and decision-making.
- Limitations/Example: Misleading visualizations or incorrect interpretation of plots can lead to incorrect conclusions. It's important to choose appropriate visualization techniques and ensure clarity in conveying insights.

Working/Algorithm:

- Load the dataset into Python using `pandas`.
- Use `matplotlib` and `seaborn` to create scatter plots, bar plots, box plots, and histograms based on the data variables.
- Customize the plots as needed to improve readability and convey insights effectively.

Diagram:



Conclusion:

Data visualization using Python libraries such as `matplotlib` and `seaborn` enables effective exploration and communication of insights from the dataset. By leveraging scatter plots, bar plots, box plots, and histograms, analysts can gain deeper insights into the data distribution, relationships between variables, and identify patterns and outliers. It's important to choose appropriate visualization techniques and customize plots to effectively convey insights to stakeholders.

Assignment No:- 6

Problem Statement:

Assignment on Regression technique.
Download temperature data from the link below.
https://www.kaggle.com/venky73/temperaturesof-india?select=temperatures.csv
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 a suitable library function and predict the Month-wise temperature.
b) Assess the performance of regression models using MSE, MAE and R-Square metrics
c) Visualize a simple regression model.

Software Library Package:

For this task, we'll use the following Python libraries:

- `pandas` for data manipulation.
- `scikit-learn` for linear regression modeling.
- `matplotlib` for data visualization.
- `numpy` for mathematical operations.

Theory:

i) Methodology:

- Linear Regression: Linear regression is a statistical method used to model the relationship between a dependent variable (target) and one or more independent variables (features). It assumes a linear relationship between the variables and estimates the coefficients of the linear equation that best fits the data.
- Advantages: Simple and interpretable model, easy to implement and understand, provides insights into the relationship between variables.
- Applications: Used in various fields such as finance, economics, healthcare, etc., for predicting outcomes based on input variables.
- Limitations/Example: Assumes linearity between variables, sensitive to outliers, may not capture complex relationships in the data.

ii) Advantages and Applications and Limitations/Example:

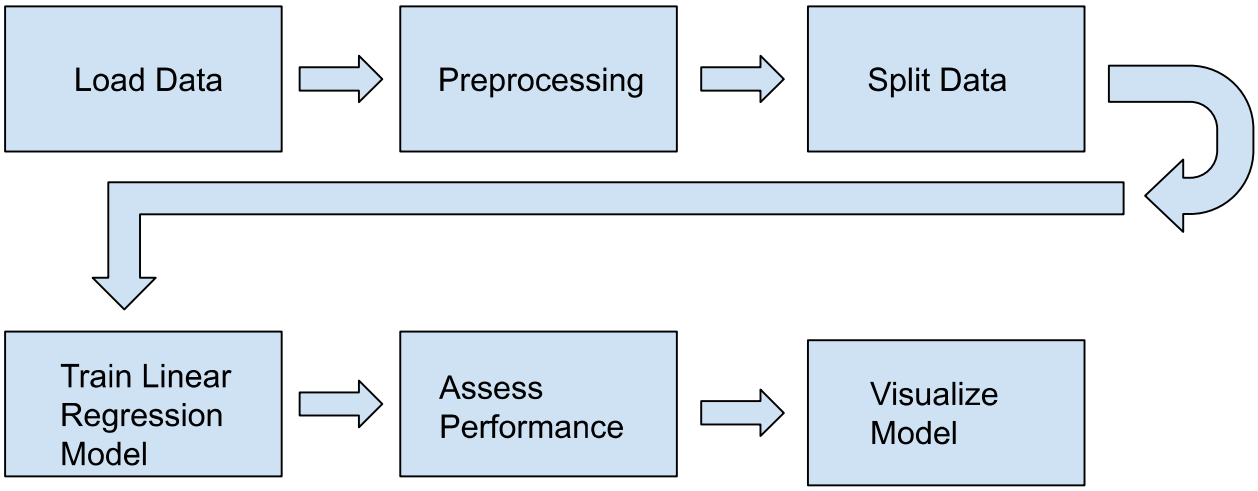
- Advantages: Linear regression is straightforward to implement, easy to interpret, and computationally efficient. It can provide insights into the relationship between the independent and dependent variables.
- Applications: Used in various domains such as finance, economics, healthcare, etc., for predicting outcomes based on input variables.

- Limitations/Example: Linear regression assumes a linear relationship between variables, which may not always hold true. It may not capture complex nonlinear relationships in the data. Additionally, it is sensitive to outliers and multicollinearity.

Working/Algorithm:

- Load the temperature data into Python using `pandas`.
- Preprocess the data if necessary, such as handling missing values or encoding categorical variables.
- Split the data into training and testing sets.
- Fit a linear regression model to the training data using `scikit-learn`.
- Assess the performance of the model using metrics such as mean squared error (MSE), mean absolute error (MAE), and R-squared.
- Visualize the regression model using `matplotlib`.

Diagram:



Conclusion:

Linear regression is a simple yet powerful technique for modeling the relationship between variables. By fitting a linear equation to the data, it allows us to make predictions and understand the influence of independent variables on the dependent variable. However, it's important to assess the model's performance using appropriate metrics and visualize the results to ensure the model's validity and interpretability.

Assignment No:- 7

Problem Statement:

Assignment on Classification technique
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.
Data Set: https://www.kaggle.com/mohansacharya/graduate-admissions
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.

Software Library Package:

For this task, we'll use the following Python libraries:

- `pandas` for data manipulation.
- `scikit-learn` for machine learning algorithms (specifically, `DecisionTreeClassifier`).
- `numpy` for mathematical operations.

Theory:

i) Methodology:

- Decision Tree Classifier: Decision trees are a popular supervised learning method used for classification tasks. They partition the feature space into regions and assign a class label to each region based on majority voting. Decision trees are constructed recursively by selecting the best feature to split the data at each node based on a criterion (e.g., Gini impurity or entropy).
- Advantages: Easy to interpret and understand, can handle both numerical and categorical data, requires little data preprocessing.
- Applications: Used in various domains such as finance, healthcare, marketing, etc., for classification tasks such as customer segmentation, fraud detection, etc.
- Limitations/Example: Prone to overfitting, sensitive to noisy data and outliers, may not capture complex relationships in the data. For example, a decision tree

may create overly complex decision boundaries if the data is highly dimensional or contains many features.

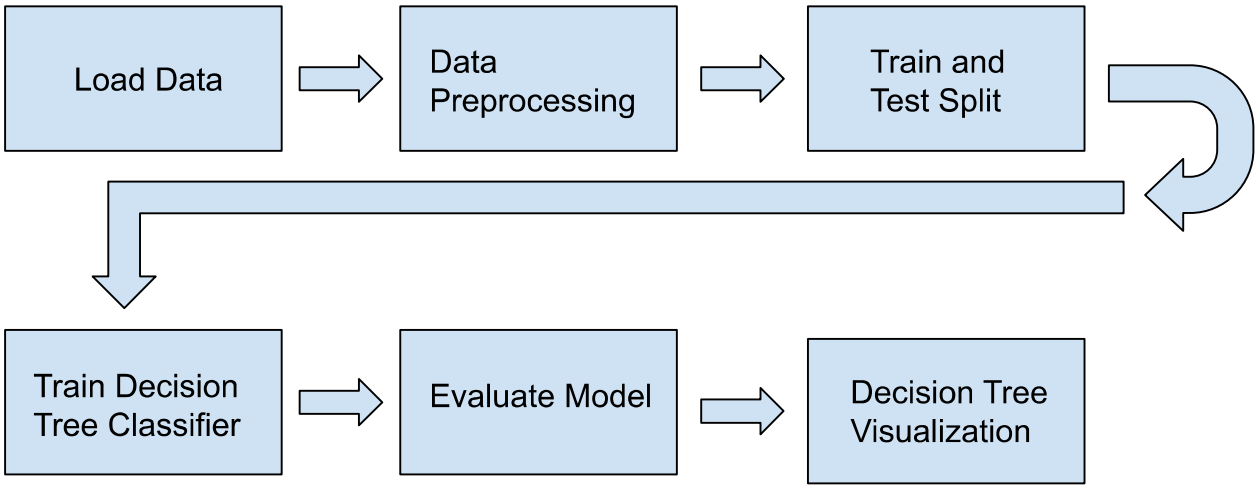
ii) Advantages and Applications and Limitations/Example:

- Advantages: Decision trees are easy to interpret and understand, handle both numerical and categorical data, and require little data preprocessing. They can capture non-linear relationships between features and the target variable.
- Applications: Used in various domains such as finance, healthcare, marketing, etc., for classification tasks such as customer segmentation, fraud detection, etc.
- Limitations/Example: Prone to overfitting, sensitive to noisy data and outliers, may not generalize well to unseen data. For example, a decision tree with deep branching may create overly complex decision boundaries, leading to poor generalization performance.

Working/Algorithm:

- Load the dataset into Python using `pandas`.
- Perform data preprocessing if necessary, such as label encoding for categorical variables.
- Split the data into training and testing sets using train-test split.
- Train a decision tree classifier model using the training data.
- Evaluate the model's performance using appropriate metrics such as accuracy, precision, recall, and F1-score.
- Visualize the decision tree if desired for better interpretation.

Diagram:



Conclusion:

Decision trees are a powerful and interpretable machine learning algorithm for classification tasks. By constructing a tree-like model of decisions based on features, they provide insights into the decision-making process. However, it's important to handle overfitting and interpretability issues carefully, especially with deep decision trees. Overall, decision trees can be valuable tools for predicting admission outcomes in foreign universities based on student scores.