ASSIGNMENT NO.2

Title: 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/Libraries Used:

- Python

- pandas

- matplotlib

- scikit-learn

Theory/Methodology:

We'll use Python along with pandas for data manipulation, matplotlib for data visualization, and scikit-learn for building a classification model. Here's how we'll perform each task:

a) Computing Summary Statistics:

- Using pandas' `describe()` function to compute summary statistics like minimum, maximum, mean, standard deviation, variance, and percentiles for each feature in the dataset.

b) Data Visualization with Histograms:

- Creating histograms for each feature using matplotlib to visualize the distributions of the features in the dataset.

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

- Data Cleaning: Handling missing values, outliers, and any inconsistencies in the dataset using pandas.

- Data Integration: Combining multiple datasets if necessary to enrich the dataset.

- Data Transformation: Scaling or normalizing features, encoding categorical variables, and any other necessary transformations using pandas or scikit-learn.

- Model Building: Utilizing scikit-learn to build a classification model, such as Logistic Regression, Decision Trees, or Random Forests, depending on the problem at hand.

Advantages/Application:

- Python offers a wide range of libraries suitable for various stages of the data analysis pipeline, from data manipulation to visualization and modeling.

- pandas provides intuitive and powerful tools for data manipulation, making it easy to clean, transform, and analyze datasets.

- matplotlib offers flexible and customizable plotting capabilities, allowing users to create informative visualizations to explore the data.

- scikit-learn provides efficient implementations of various machine learning algorithms, making it convenient to build and evaluate classification models.

Limitations/Examples:

- Depending on the size of the dataset, certain operations like computing summary statistics or building complex models may require significant computational resources.

- Data cleaning and preprocessing steps can be time-consuming, especially for large and messy datasets.

- The choice of classification model and its performance may vary depending on the specific characteristics of the dataset and the problem being addressed.

Working/Algorithm::

1. Load the dataset into a pandas DataFrame.
2. Compute summary statistics using the `describe()` function.
3. Create histograms for each feature using matplotlib.
4. Perform data cleaning, integration, and transformation as needed.
5. Split the dataset into training and testing sets.
6. Choose a classification algorithm and train the model using scikit-learn.
7. Evaluate the model's performance using appropriate metrics.
8. Iterate on the model by tuning hyperparameters or trying different algorithms if necessary.

Conclusion:

By leveraging the capabilities of Python libraries like pandas, matplotlib, and scikit-learn, we can efficiently perform exploratory data analysis, data visualization, data cleaning, and classification modeling tasks. This integrated approach enables us to gain insights from the data, build predictive models, and make data-driven decisions effectively. However, it's important to carefully preprocess the data and choose appropriate modeling techniques to ensure the reliability and effectiveness of the results.