IT & Computer Science Department

Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan

Project Part-2:

EXPLORATORY DATA ANALYSIS AND VISUALIZATION

Introduction:

In this assignment, your role is to conduct thorough exploratory data analysis (EDA) and visualization to uncover patterns and relationships within the dataset. The client expects you to provide clear visualizations and insights into the dataset's characteristics.

TASKS:

1. **Feature understanding and Visualization (15 marks):**

- **Utilize appropriate visualizations to show the distribution of features within the dataset. This includes histograms, density plots, boxplots, or other relevant graphs.**
- **Investigate and visualize relationships between pairs of features. Use scatter plots or correlation matrices to illustrate correlations.**
- **Identify and visually represent features that are important based on their distribution and relationship with the target variable.**

Answer:

- **Distribution of Features:**

  Visualizing feature distributions helps understand the spread and central tendency of numerical or categorical data. Histograms, density plots, and boxplots depict these distributions effectively.

- **Relationships Between Features:**

  Investigating relationships through scatter plots or correlation matrices reveals connections or dependencies between pairs of features. Correlation matrices quantify these relationships numerically.

- **Identification of Important Features:**

  Features impacting the target variable significantly are visually represented. For instance, using count plots or scatter plots against the target variable illustrates their importance.

Comparison with Python Code:

- Visualizations like histograms, density plots, and boxplots in Python (using Matplotlib or Seaborn) provide an actual representation of feature distributions.
- Scatter plots and correlation matrices generated using Python libraries visualize relationships between pairs of features numerically and visually.
- Count plots or other plots against the target variable highlight feature importance.

2. Data Wrangling (20 marks):

Perform essential data wrangling tasks, including handling missing values, duplicates, and outliers. Use appropriate techniques in Python (pandas) to ensure data cleanliness.

Answer:

Data wrangling ensures data cleanliness by handling missing values, duplicates, and outliers.

Handling Missing Values:

Detecting missing values and filling or dropping them using appropriate techniques (like mean/median imputation or dropping rows/columns) maintains data integrity.

Dealing with Duplicates:

Identifying and removing duplicates ensures data consistency, preventing biases in analysis or modeling.

Outlier Treatment:

Detecting and handling outliers using statistical methods (e.g., IQR, Z-score) maintains data quality for accurate analysis.

Comparison with Python Code:

- Pythons pandas library facilitates detecting missing values using functions like isnull() and handling them using fillna() or dropna().
- The code snippet demonstrates identifying duplicates with duplicated() and removing them with drop_duplicates().
- Handling outliers with statistical methods like IQR is implemented in Python using Pandas and NumPy library.

3. Feature Engineering (25 marks):

Engineer new features or transform existing ones to enhance the predictive power of the dataset, where it is possible/required. Consider techniques such as one-hot encoding, scaling, or creating interaction terms.

Answer:

Feature engineering involves creating new features or transforming existing ones to improve predictive power.

One-Hot Encoding:

Converting categorical variables into numerical form through one-hot encoding expands the feature space, aiding machine learning models.

Scaling Numerical Features:

Scaling numerical features to a similar range prevents dominance of certain features in algorithms sensitive to feature scales.

Creating Interaction Terms:

Generating new features by combining existing ones captures interactions, providing additional predictive power.

Comparison with Code:

- Python libraries like Scikit-learn offer tools for **one-hot encoding** and scaling **numerical features** using **MinMaxScaler.**
- Creating interaction terms or generating new features through mathematical operations is exemplified in the code snippet.

4. Visual Representation (10 marks):

Develop a visually appealing and informative representation of the dataset, such as a block diagram, highlighting key components and their interactions.

Answer:

The block diagram showcases dataset components, interactions, and data preparation steps, providing a clear visual representation of the dataset's structure and relationships.

Dataset Key Components:

User-related Features:

Includes user_name, user_location, user_description, user_created, user_followers, user_friends, user_favourites, user_verified.

Tweet-related Features:

Comprises date, text, hashtags, source.

Statistical Features:

Encompasses retweets, favorites, is_retweet.

Feature Interactions:

Arrows between components showcase potential interactions or relationships between various features. For instance, interactions between user-related features or between tweet-related features.

Data Preparation Steps:

Represents the steps taken for data cleaning, handling missing values, duplicates, outliers, and feature engineering etc.

Feature Engineering Impact:

Demonstrates the impact of engineered features, if any, on original features or their relationships.

Block Diagram:

The following block diagram is shown: