Assignment: 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:

Python with pandas , numpy and scikit-learn (DecisionTreeClassifier)

1. Theory:

1.1 Methodology:

The implemented program utilizes several functions and libraries to preprocess the data, train the model, and evaluate its performance.

1.2 Advantages and Applications:

-Efficiency: Utilizing functions from popular libraries like pandas, scikit-learn, and matplotlib enhances code efficiency and readability.

- Modularity: Functions allow for modular code design, making it easier to understand, debug, and maintain.

- Reusability: Functions can be reused across different projects or scenarios, promoting code reuse and scalability.

1.3 Limitations:

- Dependency on External Libraries: The program relies on external libraries, which may introduce dependencies and compatibility issues.

- Potential Overhead: Utilizing functions from libraries may introduce overhead in terms of memory and computational resources.

2. Working/Algorithm:

The implemented program consists of several functions:

- `read_csv: Reads the CSV file containing the dataset using pandas.

- `describe`: Provides descriptive statistics of the dataset.

- `info`: Prints information about the dataset, including column names, data types, and non-null counts.

- `shape`: Returns the shape (number of rows and columns) of the dataset.

- `isnull().sum()`: Calculates and prints the number of missing values for each column in the dataset.

- `fillna(0, inplace=True)`: Fills missing values with zeros in the dataset.

- `LabelEncoder()`: Encodes categorical variables into numerical labels.

- `train_test_split`: Splits the dataset into training and testing sets.

- `DecisionTreeRegressor`: Initializes a Decision Tree Regressor model.

- `fit`: Fits the model to the training data.

- `predict`: Predicts the target variable using the trained model.

- `mean_squared_error`: Calculates the mean squared error between the predicted and actual target values.

- `mean_absolute_error`: Calculates the mean absolute error between the predicted and actual target values.

- `r2_score`: Calculates the R-squared score of the model.

3. Conclusion:

The program effectively utilizes functions from libraries like pandas, scikit-learn, and matplotlib to preprocess the data, train a Decision Tree Regressor model, and evaluate its performance. These functions enhance code efficiency, modularity, and reusability. However, it's essential to consider the limitations associated with external dependencies and potential overhead. Overall, the program provides a structured approach to building and evaluating machine learning models for regression tasks.