Assignment #2: Adult Income Prediction

Supervised Learning Module

Sergio A. Rojas, PhD.

Aim:

This assignment aims to assess your proficiency in data analysis, feature engineering, classification algorithms, and model selection. You will work with the Adult Income Dataset to predict whether an individual earns over $50K per year.

Dataset:

Download the Adult Income Dataset from the UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/2/adult

This dataset contains demographic and socioeconomic information about individuals, along with their income level.

Tasks:

Q1: Data Exploration and Analysis (5 points)

- Load the dataset into a suitable data structure (e.g., Pandas DataFrame).
- Display the data's shape (number of rows and columns).
- Analyze data types for each column.
- Generate descriptive statistics for numerical features (mean, standard deviation, minimum, maximum).
- Use visualization techniques (e.g., histograms, boxplots) to explore the distribution of numerical features and identify potential outliers.
- Analyze the target variable ("income") and its distribution.

Q2: Data Cleaning and Preprocessing (5 points)

- Check for null values and report the count and percentage of missing data for each column.
- Check for duplicate rows and remove any duplicates if necessary.
- Handle missing values using appropriate strategies (e.g., imputation, deletion).
- Define the target variable as the "income" column.

Q3: Data Splitting (5 points)

- Split the data into training and testing sets (80/20 split).
- Check if the splits are balanced in terms of the target variable. If not, adjust the splitting strategy to ensure balance.
- Report the size of the training and testing sets.

Q4: Rule-Based Classifier and Feature Analysis (10 points)

- Compute and plot pairwise correlations between features to identify potential relationships.
- Create pairwise scatter plots to visualize the relationships between features and the target variable.
- Based on your analysis, select the three most promising discriminative variables and build a collection of rule-based classifiers using pairwise predictors.
- For each pair of predictors, implement a rule-based classifier and evaluate its performance using accuracy, specificity, sensitivity, and F1-score.
- Report the best-performing rule-based classifier and its performance metrics.

Q5: Decision Tree Classifier (10 points)

- Implement a decision tree classifier using a suitable library (e.g., scikit-learn).
- Train the decision tree model on the training data.
- Experiment with different hyperparameters (e.g., maximum depth, minimum samples split) using GridSearchCV or RandomizedSearchCV.
- Evaluate the decision tree model's performance on the testing set using the same metrics as Q5.
- Select the best-performing configuration based on metrics and report the chosen hyperparameters.
- Visualize the decision tree (optional) to understand the decision-making process.

Q6: k-Nearest Neighbors Classifier (10 points)

- Encode categorical variables using one-hot encoding.
- Implement a k-NN classifier using a suitable library (e.g., scikit-learn).
- Train the k-NN model on the training data.
- Experiment with different values of k and evaluate the model's performance on the testing set.
- Report the performance for different k values and identify the optimal k based on the metrics.
- Discuss the impact of k on the classifier's performance.

Q7: Comparative Analysis (5 points)

- Compare the performance of the rule-based classifier, decision tree, and k-NN classifier using the same evaluation metrics.
- Discuss the advantages and disadvantages of each approach in the context of this dataset and task.

Bonus: Majority-vote classifier (5 points)

- Build a majority-vote classifier that combines the predictions of the three best-performing models.
- Evaluate the performance of the majority-vote classifier and compare it to the individual models.
- Analyze the impact of different train-test splits (40/60, 60/40, 80/20, 90/10) on the performance and generalization of individual and majority-class models.

Submission:

Submit by **September 3th/2024, 5pm**, a PDF version of Jupyter Notebook (Colab) containing your code, explanations, comments, visualizations, and performance results onto the provided shared folder.

Additional Notes:

- For examples and inspiration, refer to the following suggested Kaggle reference:

  https://www.kaggle.com/code/mzohaibzeeshan/adult-income-prediction

  https://www.kaggle.com/code/codealiahmad/adult-income-set-data-cleaning-eda-ali-ahmed

- Provide explanations for your code steps and choices and discuss your findings in detail.
- Use visualizations to illustrate your results and insights.
- You may work in pairs for this assignment.