Project #3

CIS 2541 - Prof. John P. Baugh – Oakland Community College – OR

Objectives

- To apply data preparation, including data cleaning
- To practice and learn more about Decision Trees and ensemble techniques (e.g, Random Forests)
- To predict the survival of passengers on the Titanic using the Titanic dataset
- You may work in groups of up to **3 students**
- Make sure to indicate who you were in a group with, within the .ipynb or .py file markdown or comments

Instructions

Dataset

The dataset is available in multiple places, such as through *Data Science Dojo*:

https://github.com/datasciencedojo/datasets/blob/master/titanic.csv

The dataset contains information about the passengers on the *Titanic*, including whether they survived or not (the target). Key features include:

- PassengerId
- Survived (target variable)
- Pclass (ticket class)
- Name
- Sex
- Age
- SibSp (number of siblings/spouses aboard)
- Parch (number of parents/children aboard)
- Ticket
- Fare
- Cabin
- Embarked (port of embarkation)

Steps

1. **Data Collection**
    a. Obtain the Titanic dataset (link provided in previous section)
    b. For convenience: https://github.com/datasciencedojo/datasets/blob/master/titanic.csv

2. **Data Cleaning and Preparation**
    a. **Handle missing values**
        i. Identify missing values and decide how to handle them (e.g., mean/median imputation for numerical features, mode imputation for categorical features)
    b. **Drop irrelevant features**
        i. Remove features that are not useful for prediction (e.g., PassengerId, Name, Ticket, Cabin)
    c. **Convert categorical variables**
        i. Encode categorical variables (e.g., Sex, Embarked) using one-hot encoding
    d. **Feature engineering (optional)**
        i. Create new features if necessary

3. **Exploratory Data Analysis (EDA)**
    a. Analyze the distribution of features and their relationship with the target variable (Survived)
    b. Visualize data using histograms, bar plots, and correlation matrices

4. **Splitting the data**
    a. Split the data into training, validation, and testing: 60% training, 20% validation, 20% testing

5. **Model training**
    a. **Baseline model**
        i. Train a baseline Random Forest classifier with Default parameters
    b. **Model Evaluation**

i. Evaluate the baseline model using **ROC-AUC** (as was done in the Credit scoring example in the lecture)

6. **Hyperparameter Tuning**
    a. **Parameters to tune**
        i. Experiment with different values for max_depth, min_samples_leaf, and n_estimators

7. **Final Model Evaluation**
    a. **Retrain with best parameters**
        i. Retrain the RF model with the best hyperparameters found during tuning
    b. **Evaluate on Test Set**
        i. Evaluate the final model on the test set and compare with baseline results
        ii. Did it improve from the default RF model?

8. **Reporting**
    a. **Summary of Findings**
        i. Summarize the key findings, including model performance metrics
    b. **Visualizations**
        i. Include relevant visualizations such as the ROC-AUC curves
    c. **Challenges and improvements**
        i. In Markup within your Ipynb file (or comments in your .py file), briefly discuss any challenges faced during the project and potential improvements for future work

Hints

You might find the following hints / code segments helpful:

- `df['Age'].fillna(df['Age'].median(), inplace=True)`
- `df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)`
- `df.drop(…)`
- `X = df.drop('Survived', axis=1)`
- `y = df['Survived']`

Deliverables

- Turn in a zip including your source code and screenshots of the program functioning, as follows:
    - An **ipynb** file for Jupyter Notebooks
        - Alternatively, a **py** source file is acceptable as well
    - **Include screenshots of your program working,** placed inside the zip file that you turn in
        - This should include screenshots of the outputs including diagrams and printing of the shapes, evaluation metrics, etc.