CST8502 - Lab 5
Python

Due Date: Check Brightspace for due dates.

Introduction

The goal of this lab is to perform classification using Decision Trees and kNN on Titanic dataset in Python. All tasks should be done in Python.

Set the random seed as 2025. Once this is set, it ensures consistent randomness across the script. You can do this by setting random_state in various methods and seed for random and numpy (after importing corresponding libraries, you can set random.seed(2025) and np.random.seed(2025))

Make sure to follow every step as given in this document. Otherwise, your answers will be different and autograder will mark it wrong.

Steps (all these steps should be done in Python):

1. Set random seed.
2. Load the **given** titanic.csv file and print number of instances, attributes and first 5 instances (this print should be with corresponding messages like "Number of instances: xxx etc).
3. Check for data quality issues and remove all irrelevant columns.
4. Create new columns for AgeGroup, Relatives and Fare (Free if 0, Low if less than 50, average if less than 100, high otherwise). (Refer to lab 3 for instructions for AgeGroup and Relatives)
5. Print the first 5 rows. You should have Passenger class, sex, age group, relatives, fare and embarked as attributes.
6. Scikit-learn decision tree will not accept categorical data, so, apply one-hot encoding to convert the attributes to binary.
7. Split data into train and test set. (Refer to https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)
8. Fit Decision Tree model in Scikit-learn Package for the train set. (Refer to https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier)
9. Predict the survival for the test set. Print accuracy and confusion matrix.
10. Visualize the decision tree (tree should be readable).
11. Write 5 rules in the answer document.
12. Now, prepare dataset for distance-based methods. Start with the original dataset and split into test and train sets. Make sure to have passenger class, sex, age, number of relatives, fare, and embarked as attributes.
13. Extract the passenger title (e.g., Mr., Mrs., Master, etc.) from each name. Next, calculate the average age for each title group using only passengers with valid age values. Finally, replace the missing age values with the corresponding group average based on the passenger's title
14. Numerical columns like Fare, Age etc. should be normalized. When you normalize, fit the scaler on train set and apply the scaler on the train and test set.
15. Categorical columns like PClass, Embarked etc. should be one-hot encoded.
16. Perform classification using kNN and print the results. (https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html)

17. **Bonus**: As a bonus activity, you can try other tree-based classifiers that takes nominal columns as is (without one-hot encoding) and creates tree model. If you try any other models that works with nominal columns, make sure to fit a tree on the train set and test it with the test set. Print confusion matrix, accuracy and then visualize the tree. This should be separate from the rest of the code and should be done as a separate section.

To get grades,

1. You should be ready with your Python code and results.
2. Submit your answer document AND colab notebook/jupyter notebook. Failure to submit any of these will end up in a grade of 0 for the lab.
3. **DO NOT ZIP**. Zipped files will not be graded.