CS304 - Homework 1 – Classification Using Titanic Dataset

In this homework, you will work on the Titanic dataset given in the .csv files: titanictrain.csv, titanictest.csv. This is an example of **structured dataset** where each feature is given in a column.

Titanic dataset contains a number of features to predict whether the passenger survived or not [1].

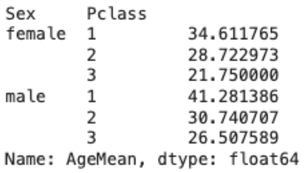
In this homework you will: i) to load and analyze the data, ii) prepare the data for training, iii) train two different machine learning models and compare their performances.

Step 1: Load and analyze the data

a. Load the two datasets using Pandas `.read_csv()` method
b. Display the shapes (number of rows and columns) of training and test datasets and compare.
c. Display the first five rows of training and test sets using the `.head()` method
d. Inspect the feature names. Further information about the features can be found in [1] and [2].
e. Determine which column exists in the training set but is missing in the test dataset
f. Observe which columns contain numerical and which columns contain text-based data. Inspect the training and test data using the `.info()` method. Inspect the statistical propoerties of numerical attributes using `.describe()` method.
g. Determine the number and percentage of passengers who survived. Is this a balanced dataset?

Step 2: Prepare the data for classification

a. From the training dataset, extract the target labels given in column “Survived” to a new array named **y_train**.
b. We will only use the features “Pclass”, “Sex”, “Age”, and “Fare”. This is because other features (name or the ticket number etc.) does not have an effect on the survival of the passenger. The passenger’s gender affects his/survival since females are prioritized when in danger. Create a new DataFrame named **x_train** and copy these columns from the training dataset.
c. Add another feature named “FamilySize” which is the sum of columns “Sibsp” and “Parch” . Limit the sum so that it does not exceed 4.
d. Transform the categorical data given in the “Sex” (two values: Male and Female) to numerical data (i.e. 0 or 1)
e. Desing data **imputation methods** to fill in the missing values in the “Fare” and “Age” columns. You can try several of the following options:
    i. Set the missing values to some value: zero, the mean, the median of the existing values.
    ii. (optional) Calculate the mean Age and mean Fare for each of the groups of data based on “Sex” and “Pclass” using the Pandas `.groupby()` method [3], [4]. For example, you can find the average age for each group:



```
Sex     Pclass
female  1         34.611765
        2         28.722973
        3         21.750000
male    1         41.281386
        2         30.740707
        3         26.507589
Name: AgeMean, dtype: float64
```

Using the code:

```
age_mean = train_df.groupby(['Sex', 'Pclass']).Age.mean()
```

```
age_mean.name = 'AgeMean'
```

Then, fill in the missing values using the mean the group that they belong regarding “Sex” and “Pclass”. You can use the Pandas `.merge()` and `fillna` methods to achieve this.

  iii. (optional) You can also use more sophisticated imputation methods (`KNNImputer`, `IterativeImputer`)

f. Scale the numerical features “Fare” and “Age” using `MinMaxScaler()` or `StandardScaler()` to see if scaling the features will help.

g. (Optional) You can use a *pipeline* to handle categorical data, missing data and feature scaling given in steps d, e, f above.

Step 3: Train two different ML models and compare their performances.

a. Split your training dataset into two parts to be used in training (and testing) the models. (Since the original test set is missing the target labels). Use the `train_test_split()` function, and a ratio of 80% for training and 20% for testing.
b. Train a logistic regression classifier on the training partition and report the accuracy on the test partition. Use the `LogisticRegression()` method.
c. Train a random forest classifier on the training partition and report the accuracy on the test partition. Use the `RandomForestClassifier()` method.
d. Repeat b. using 5 fold cross validation (skip step a). Calculate the mean accuracy of the 5 folds and the standard deviation.
e. Repeat c. using 5 fold cross validation (skip step a). Calculate the mean accuracy of the 5 folds and the standard deviation.
f. Inspect and draw the confusion matrices (of the two classifiers.
g. Calculate the precision, recall and F1 scores for the two classifiers.
h. Draw the precision-recall and ROC curves for the two classifiers.

Step 4: Comment on your results

a. Which classifier gave better results?
b. Which data imputation method gives better results?
c. How can you further improve the performance?

What should you submit?

- Submit your .ipynb file containing your code and expalanations.
- Also submit your .html (or .pdf) file showing the result of each cell.

References:

[1] “Titanic-Machine Learning from Disaster”, Kaggle, https://www.kaggle.com/competitions/titanic

[2] “A Beginner’s guide to Kaggle’s Titanic Problem”, S. Mukhija, https://towardsdatascience.com/a-beginners-guide-to-kaggle-s-titanic-problem-3193cb56f6ca

[3] pandas. DataFrame.groupby: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.groupby.html

[4] Working with missing data: https://pandas.pydata.org/docs/user_guide/missing_data.html

[5] pandas.merge: https://pandas.pydata.org/docs/reference/api/pandas.merge.html#pandas.merge