README: Group 1

Ai-Linh Alten, Rajeshwari Chandratre, Scott Vuong Tran

Predicting survivors of the Titanic

Our project builds and trains a model using training data pertaining to passengers of the Titanic and works to make predictions on a set of unlabelled testing data.

How to run?

If you plan to run the project using **Enthought Canopy**, confirm that the entire project is placed in your home directory.
Once confirmed, **run the TitanicMain.py module**
or
If using a **terminal**, navigate to the project folder and execute the following command: **python3 TitanicMain.py** or **py TitanicMain.py** (specifically for Window machines)

Optional --gridsearch parameter (May take some time to run to completion)
To **run** our project **with GridSearch**, in your terminal, execute the following command: **python3 TitanicMain.py --gridsearch** or **py TitanicMain.py --gridsearch**. Otherwise, open up the Canopy Command Prompt and execute the two previously mentioned commands.

Main module - /group1projectcode/TitanicMain.py

The main module where the training data and testing data are cleaned & visualized, and a model is built to do predictions on the testing data based on the training data..

Data - /group1projectcode/Data/…

- train.csv The training data set to be used to train the model
- test.csv The testing data set to test the model
- TitanicDataClean.csv
  The cleaned training data saved to a CSV file
- TitanicDataTestClean.csv
  The cleaned testing data saved to a CSV file
- gender_submission.csv
  The target column that corresponds to the test data

Modules - /group1projectcode/Modules/…

- **TitanicClean.py**
  Includes the functions to clean the data i.e. log transformations, KNN imputations on specific columns, make_dummies, drop columns, sort columns,
- **TitanicPlotting.py**
  Includes the functions to visualize the data (Histograms, Heatmap)
  Include the function to do Pearson's correlation against all features
- **TitanicModel.py**
  Includes the code to build a model using training data, test the model using testing data. Additionally includes k-fold cross validation and ROC plot
- **KNNimpute.py**
  The code that does KNN imputations on certain columns
- **TTestPearson.py**
  The code that does the hypothesis testing on all features using Pearson's Correlation Coefficient

- **GridSearch.py**
  This code is optionally ran if --gridsearch is specified as an option when running our project code. This module is used for cross validation and to do parameter estimation.

Visualizations - /group1projectcode/Visualizations/...

This directory contains a few selected visualizations including the histogram for the uncleaned + cleaned data as well as the correlation heatmap and the generated ROC plot.

Sample Output - /group1projectcode/Sample-Output/...

- **#OUTPUT.txt**
  - This file contains sample output that you should expect to see
- **#OUTPUT-GRIDSEARCH.txt**
  - This file contains sample output that you should expect to see if you ran the project with the optional --gridsearch parameter