COMP4432 - Assignment 3

(due by midnight MST the day prior to Live Session 6)

Part 1: Data Exploration.

Load the titanic dataset from Seaborn by using the *load_dataset(‘titanic’)* method. Document the columns that are missing data both numerically (via a count) and visually (via an sns heatmap). Document which values are categorical. Explore the data and answer the following questions: Did more women or men die on the *Titanic*? Which passenger class was more likely to survive? What does the distribution of fare look like? What does the distribution of non-null age values look like? What is the median age of each passenger class (pclass)? Visualize this in a box plot.

Part 2: Data Cleansing.

Since there are so many missing values in Cabin, get rid of the cabin feature. Define a function to impute age using the median of the passenger class you computed earlier. To call it, use *train[[‘age’, ‘pclass]].apply(impute_age,axis=1)*. Drop the remaining records containing null values. Show there are no remaining null values. Convert categorical variables to numeric dummies using pandas’ *get_dummies()* method. Add these to your training dataframe. Drop the categorical columns you converted earlier as well as *name, ticket, and passengerId*. Create a feature set by dropping “Survived.” Your resulting feature set should include pclass, age, sibsp, parch, fare, and the categorical dummy columns you created earlier. Implement a label dataframe by copying the contents of the Survived column of your training set to a new dataframe. Split your clean data into a training and test set.

Part 3: Model Training.

Implement a logistic regression model. Implement a support vector classifier. Implement an sgd classifier. Print out the classification reports, confusion matrices, and roc score and chart for each of these. Remember to set Probability=True for SVM and use method=decision_function in a cross_val_predict instead of predict_proba for the SGD ROC plot.

Part 4: Model Tuning

- See if scaling your input data affects your SVC model (implement a sklearn pipeline to combine scaling and instantiation of your model).

- Do a grid search of your pipeline classifier using the following parameter grid:
  {‘<your_svc_model_name>_________kernel’: [‘rbf’],
  ‘<your_svc_model_name> _________gamma’: [0.0001, 0.001, 0.01, 0.1, 1],
  ‘{‘<your_svc_model_name>__C’: [1,10,50,100,200,300]}.
  - Print the best estimator, its parameters, and the resulting score. Apply this estimator to your test set
  - Implement a learning curve using your best estimator from the grid search.
    - The figure should have a title of “learning curve.”
    - Label the y-axis with “Score.”
    - Label the x-axis with “Training Examples.”
    - Make the training score red.
    - Make the validation score green.
    - What does this learning curve tell you?