Assignment 4 (due by midnight MST the day prior to Live Session 8)

Part 1:

Data Exploration

- Read in bike_share_hour.csv as a pandas dataframe. The columns are described in the bike_share_readme.txt if you need more information about them.
- Look at the dataset, and convert the columns that are categorical to a pandas “category” type.
- Look for non-null values in the dataset.
- Do a descriptive analysis of the numeric columns.
- Implement a bar plot of cnt versus season. Document which season has the most bike rides and which season has the least.
- Implement a bar chart for working day versus count. Document how bike rides are distributed across these two classes.
- Implement a bar chart for month versus count. Document which months have the most bike rides.
- Implement code to figure out which months belong to which seasons.
- Implement a bar plot of weathersit versus cnt. Document which weather situation has less bike rentals.
- Implement a point plot of weathersit on the x-axis, count on the y-axis, and the season as the hue. Document how season and weathersit are related.
- Implement a bar plot of hour versus count. Are there any specific hours that are busier than others?
- Implement a bar plot of hour versus count on weekends and holidays (when workingday = 0). Does the hourly trend change on weekends?

Part 2:

Data Preparation

- Implement and graph a correlation matrix with the remaining numeric features. Any interesting relationships?
- Scale the numerical features using StandardScaler(), and replace the original columns in your dataframe.
- Drop the following columns from your dataset: casual, registered, dteday, instant.
- Implement a histogram of the count column. What can be said based on the resulting distribution?
- Implement a train/test split with a test size of 33%.

- Implement a baseline linear regression algorithm. Use cross-validation to output r2 and mse. Calculate RMSE base on mse. Document your scores.

Part 3:

Model Training (Hint: trained all of these with a for loop and added my results to a PrettyTable.)

- Create one-hot-encoded values for your categorical columns using get_dummies and add them to your source dataset.
- Drop the original categorical columns from your source dataset.
- Do a test/train split based on your new source dataset. Implement and fit a new linear model on your new training set.
- What are the new values for r2, mse, and rmse?
- Implement and score a decision tree regressor with random_state=0.
- Implement and score a RandomForestRegressor with random_state=0 and n_esitmators=30.
- Implement and score an SGDRegressor with max_iter=1000 and tol=1e-3).Implement and score a Lasso Regressor with alpha=0.1.
    - Implement and score an ElasticNet Regressor with random_state=0.
    - Implement and score a Ridge Regressor with alpha=0.5.
    - Implement and score a BaggingRegressor.

Part 4: Model Tuning

- Take the top three performing models and implement cross-validation on them.Hint: They should be Decision Tree Regressor, RandomForestRegressor, and BaggingRegressor.

- Take your top performing model (mine was the RandomForestRegressor) and do a randomize search cv with 20 iterations and three folds.

    - I found it is best to set your n_jobs = (# of cpu's you have – 1). This took about 10 minutes on my MacBook with 4 CPUs and 8 GB of memory.
    - Your param distributions should include the following:
        - Bootstrap: true, false
        - Max_depth: 10-110, number of bins 11
        - Max_features: auto, sqrt
        - Min_samples_split: 2,5,10
        - Min_samples_leaf: 1,2,4
        - N_estimators: 200 – 2000, number of bins 10

- Take your best_estimator_ and see how it compares by doing cross_vals for r2, mse, and calculating rmse.
- Finally, run predictions on your test set with this model, and see how your r2 score and RMSE look.