ENME743
Haixiang Fang
116293242

Problem Formulation

1. Which of the following types of machine learning problems does your project fit under and why (if multiple, explain each):

   Regression (Supervised)
   K Nearest Neighbors Regression and Linear (and Polynomial) Regression are used to model the relationship between time and rental counts data.

Model Choice

2. For the type of problem you identified above, what would be an appropriate "Baseline" model that we learned about in class (or via the readings) that you could use as a performance benchmark? Put another way, what is an easy-to-implement benchmark approach (whether naive or state-of-the-art) that you can use as a "first pass" at solving the problem, and why did you pick it? Your final model at the end of the semester will try to improve upon this baseline.

   First, we start by prediction the outcome variable with its mean to set the baseline. Later, Linear Regression, KNN, and decision tree will be implemented as benchmark to improve upon the baseline.

3. Look back at your data visualization: which assumptions from that (if any) are reflected in either the modeling choices or performance of the baseline model? Which assumptions are not reflected in the baseline model? What kind of things might you try for your final "improved" model at the end of the semester that might account for those assumptions?

   The average rental counts by datetime fluctuation assumption are reflected in the modeling choices. The usage pattern of each day within the week will not be modeled since its high dimensionality.

Testing Methodology

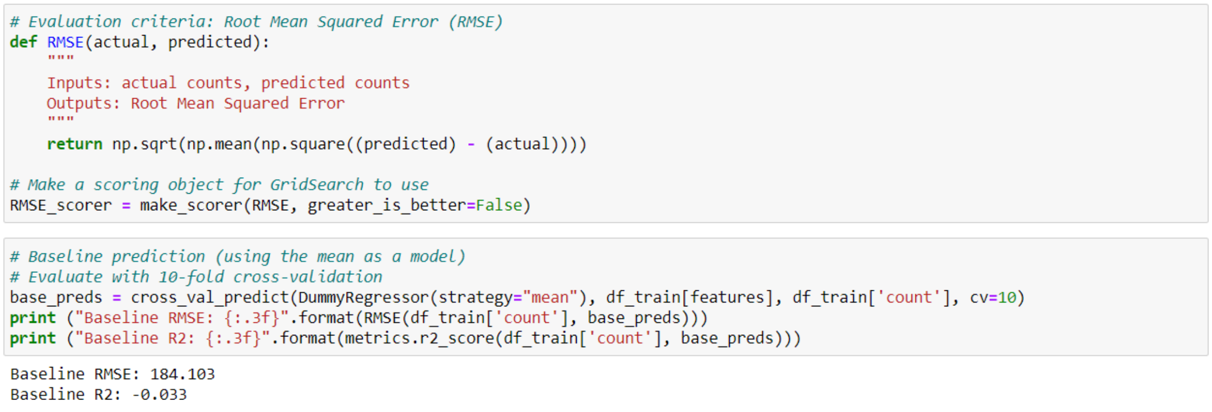
4. What constitutes "success" for your model? How will you know (i.e., quantitatively measure) if the model has done a good job? Put another way, what specific loss or score function will you use to compare your baseline model and your final model (either write it out and show how you would calculate it)?

   Root Mean Squared Error (RSME) and R^2 (R Squared) metric will be used to determine the accuracy of the model in predicting the target values. K-fold Cross Validation will be used to evaluate the RSME and R^2 and avoid over-fitting.

ENME743
Haixiang Fang
116293242

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

5. Implement (using either an existing library or your own code) your above choice of baseline model on your currently available data, and assess its performance using the "success" criteria (i.e., cost function) you selected above. Where applicable, use the appropriate type of cross-validation to differentiate performance on training data versus held-out test data, and report how well your baseline model does on your problem, making sure to describe how you do the cross-validation. This train/test performance will become "the bar" that your improved model at the end of the semester will try to exceed. Include either tables or figures of the performance where appropriate.



```
# Evaluation criteria: Root Mean Squared Error (RMSE)
def RMSE(actual, predicted):
    """
    Inputs: actual counts, predicted counts
    Outputs: Root Mean Squared Error
    """
    return np.sqrt(np.mean(np.square((predicted) - (actual))))

# Make a scoring object for GridSearch to use
RMSE_scorer = make_scorer(RMSE, greater_is_better=False)
```

```
# Baseline prediction (using the mean as a model)
# Evaluate with 10-fold cross-validation
base_preds = cross_val_predict(DummyRegressor(strategy="mean"), df_train[features], df_train['count'], cv=10)
print ("Baseline RMSE: {:.3f}".format(RMSE(df_train['count'], base_preds)))
print ("Baseline R2: {:.3f}".format(metrics.r2_score(df_train['count'], base_preds)))
```

```
Baseline RMSE: 184.103
Baseline R2: -0.033
```