# CS 4372

# ASSIGNMENT 1 - DATA ANALYSIS REPORT

## Names of students in your group:

## Scott Vu - SMV210000

## Philip Wallis - PTW190000

Number of free late days used: 0. 4, 4 Remaining  
Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

## Please list clearly all the sources/references that you have used in this assignment.

Dataset Used:

Seoul Bike Sharing Demand

<https://archive.ics.uci.edu/dataset/560/seoul+bike+sharing+demand>

Libraries used:

numpy, pandas, matplotlib, seaborn, sklearn, statsmodels

<https://scikit-learn.org/stable/modules/generated/sklearn.metrics.explained_variance_score.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html>

<https://realpython.com/linear-regression-in-python/>

<https://en.wikipedia.org/wiki/Coefficient_of_determination>

<https://seaborn.pydata.org/generated/seaborn.lineplot.html>

<https://pypi.org/project/matplotlib/>

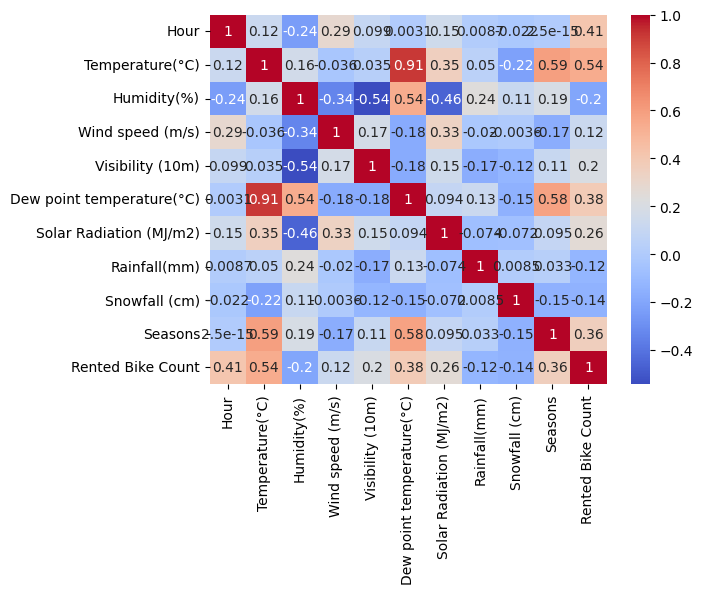
## Introduction

In this assignment, we decided to use the *Seoul Bike Sharing Demand* dataset from the UCI ML Repository (<https://archive.ics.uci.edu/dataset/560/seoul+bike+sharing+demand>). This specific dataset was chosen as it is a regression-type task with various numerical attributes that are highly likely to have correlation with the intended output of bikes rented in Seoul. With the linear regression models provided by the sklearn and statsmodels libraries, the SGD Regression Model and the OLS Model, we were successfully able to train a multitude of linear regression models and analyze them. By analyzing the chart through visualizations, we can see that the data is not really linear, and our results reflect that with a low R2 value.

## Preparing the Data

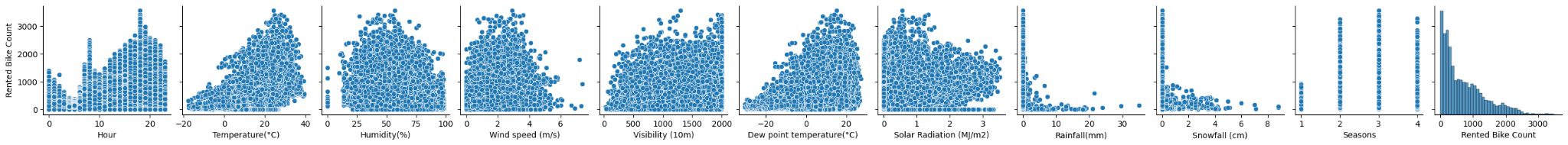
Before creating the models, the first step is to clean the data. This was achieved by dropping rows with empty values and redundant rows, then removing columns that we felt were unnecessary, such as “Date”, “Holiday”, and “Functioning Day”. Then we moved the “Number of Bikes Rented” to the last column so the table is easier to read and be processed.

Using seaborn’s heatmap function, we obtained a visualization of the heatmap of every column. At the very bottom row, you can see the “Rented Bike Count” row, which is what we will be focusing on.



The higher values show that variables such as 'Temperature' and 'Seasons' have a strong positive correlation with 'Rented Bike Count.' On the other hand, the lower values indicate that variables like 'Humidity' have a weaker or negative correlation.

Another plot to help us visualize the data is using seaborn’s pairplot.



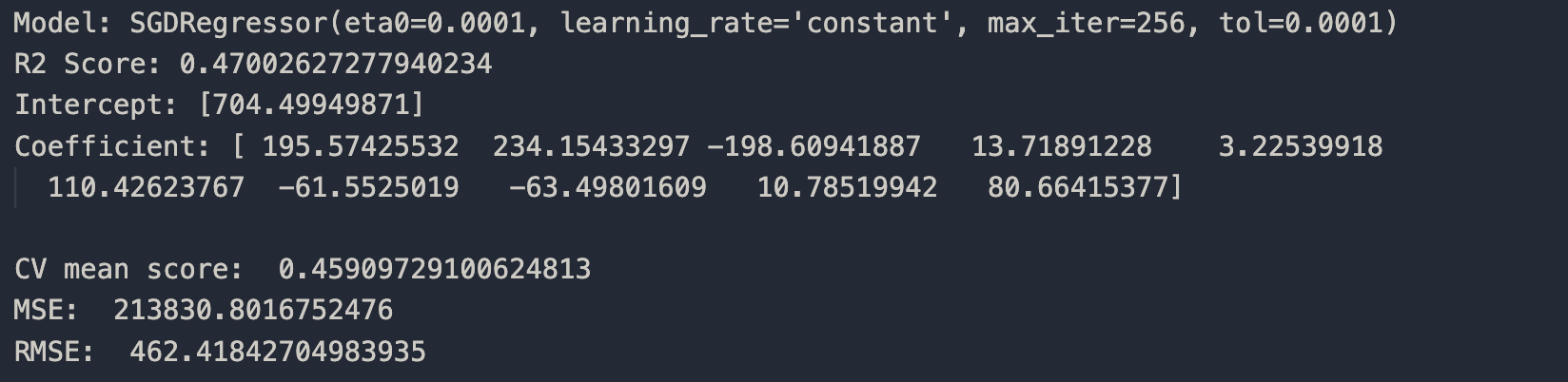
By specifying the y\_var to “Rented Bike Count”, we can get the plots to visualize their correlations to each of the data columns. Here, we see that not many of the actual values seem to have a linear correlation; this will be important later in our model generation since we are creating linear regression models.

Using sklearn function train\_test\_split, the datasets were split into a training set and a testing set. Using the test size of 0.2, we use 80% of the data for training, and 20% of the data for testing.

## SGD Regression Model

To train various models to obtain the best-performing one, we fine-tuned three hyper-parameters in the creation of the SGDRegressor class. The train\_learning\_rate (constant, eta0), train\_max\_iterations, and train\_tolerance. Through 3 nested for loops, all combinations of parameters provided were trained and tested on the same dataset, providing MSE, RMSE, R2 Scores, and other metrics. Our definition of best model is the model with the highest R2 score.

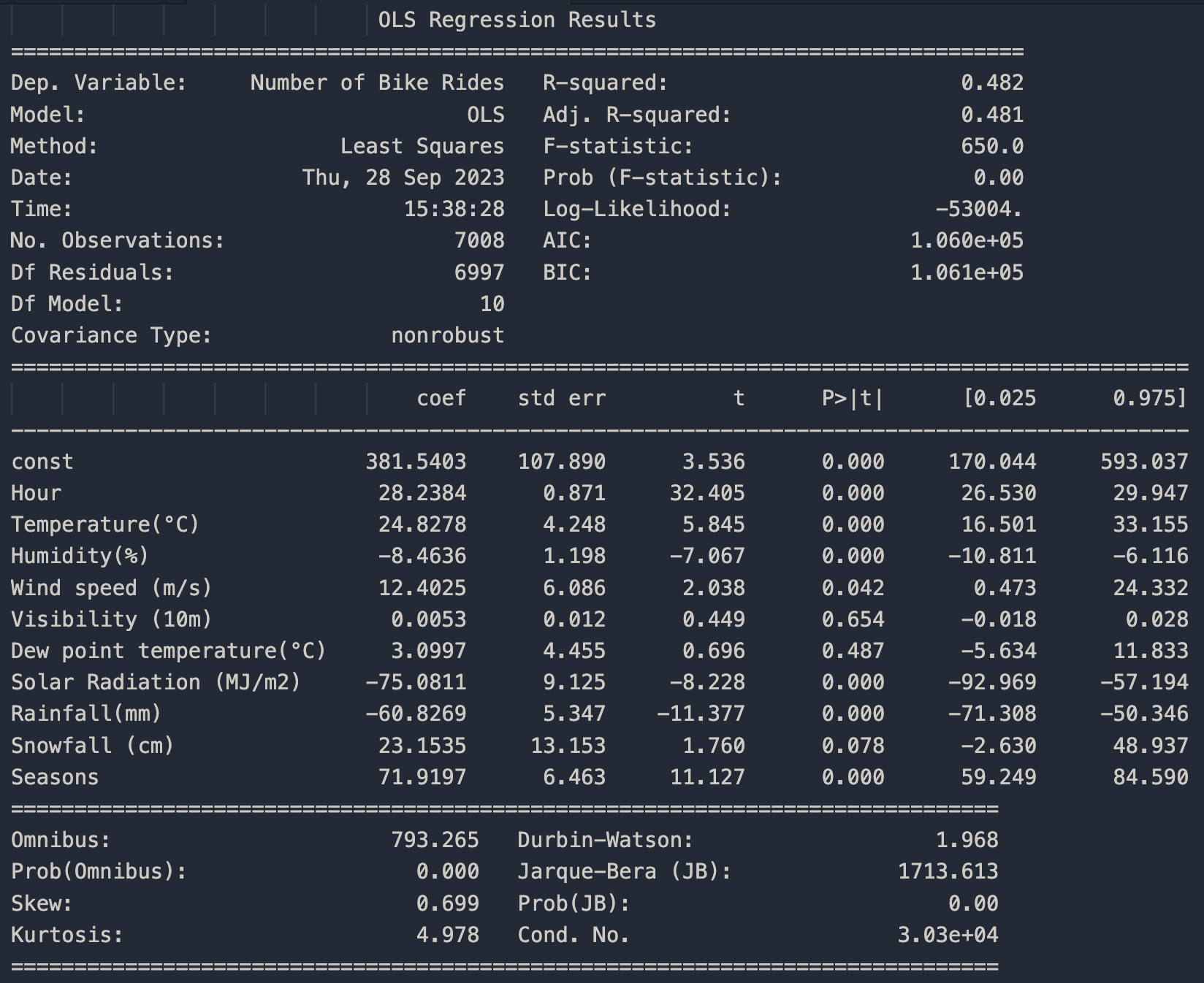
We discovered that while some hyper-parameters changed on each run, such as max\_iterations and tolerance, we still get similar R2 score results. This means that the model performs similarly without a large iteration size or extremely small tolerances. In other words, this likely means that the model has plateaued at lower values and is likely to be overfitted or overtrained at higher hyper-parameter values.



The R2 score of around 0.47 was generated for most of the trained models.

## OLS Model

To train the OLS Model, we simply used the default hyper-parameters with the same scaled training and testing data set as in the SGD Model. The default parameters of the OLS Model consists of basically everything being set to None. The metrics gathered and presented by the OLS model contain R2, AdjR2, F, and many others. Note that we forgot to get the MSE and RMSE values for direct comparisons, however.



The R2 score was 0.482, AdjR2 was 0.481, and F-statistic was 650.0. In theory, this model should be slightly better due to the higher R2 score.

The coefficients refer to the weighted values within the linear model, w0, w1, … wn, where w0 is the constant, bias, or intercept. Positive weights indicate more likely to rent, negative meaning less likely.

The standard error values represent the standard deviation of the data, where +- 2 SE would mean the 97.5th and 2.5th percentile. Smaller standard error means better fit of the weighted line.

The t-value represents the statistical significance of the weighted line, in other words, how many standard errors there are to 0, helping show what attributes may be contributing most to the prediction.

P-value being very low in most of the weights (under 0.05), means that the values are indeed correctly correlated and can object to the null’s hypothesis. In the model, variables like “Hour”, “Temperature”, and “Seasons” would be considered significant, and variables like “Visibility” and “Dew point temperature” would be considered insignificant due to its high p value.

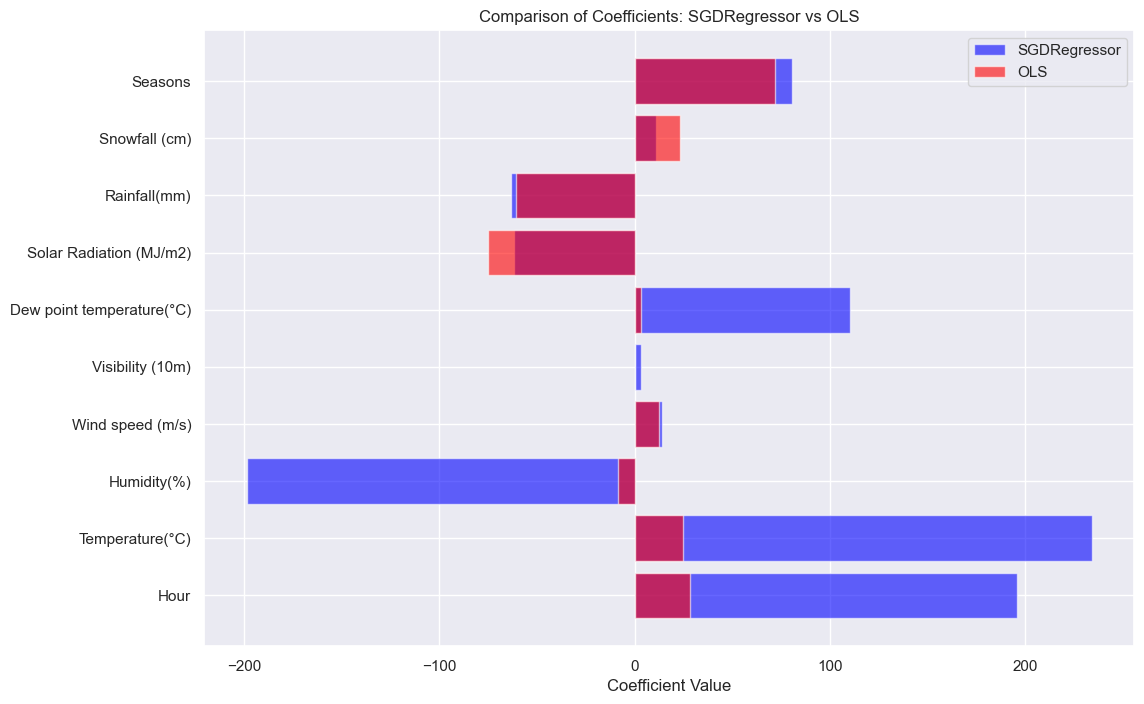
The 0.482 R-squared value tells us that approximately 48.2% of the “Number of Bike Rides” are dependent on the variables. The value is used to measure how well the model has been fit to the data.

Adjusted R-squared, 0.481 takes into account the number of predictors in the model. Being very close to the original R-squared value, it suggests that the variables included in the model are useful.

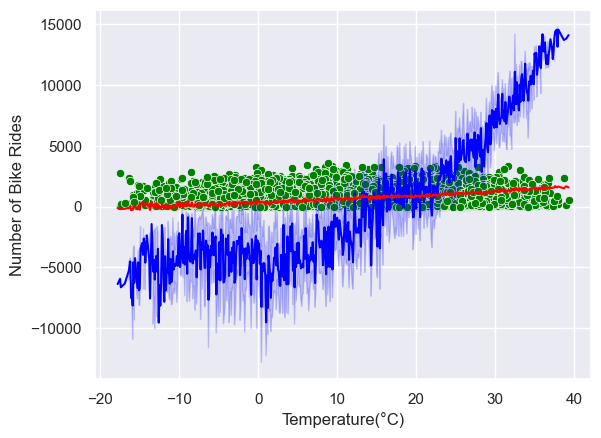
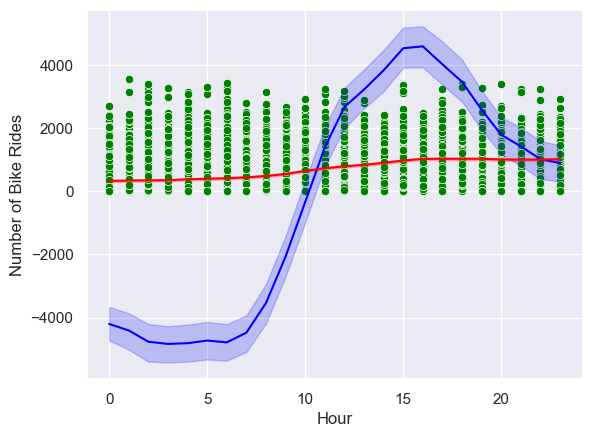
The F-statistic is used to test the hypothesis that the coefficients for the variables are equal to zero. A large F-statistic (650.0) and a low Prob (F-statistic) (0.00) both usually suggest that we can reject the null hypothesis.

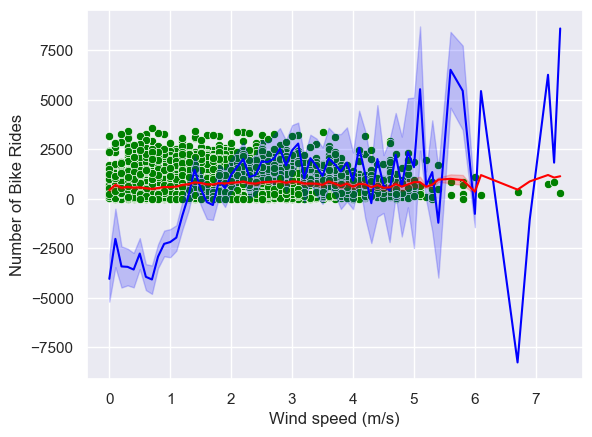
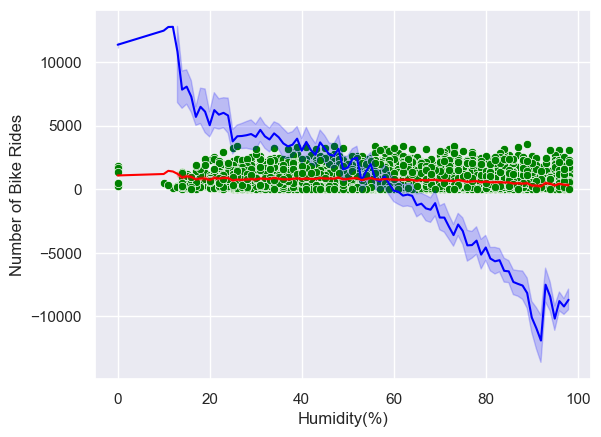
## Comparing the Two Models

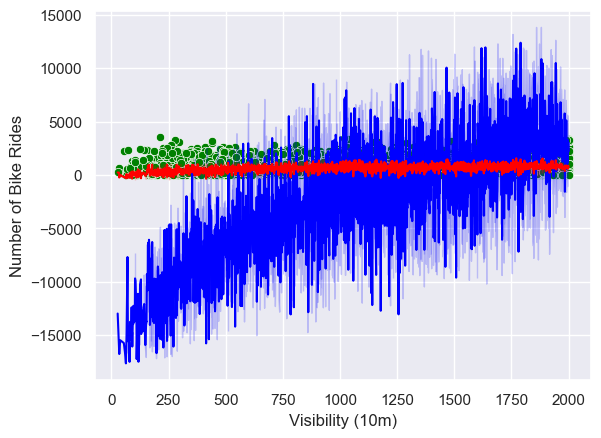
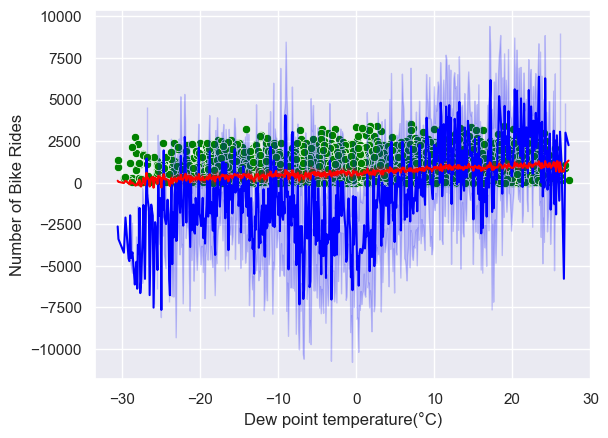
Using seaborn’s plot library, we can visualize the plotted coefficient lines of the SGDRegressor and OLS. Variables like “Temperature” and “Hour”, which have high coefficient values, tell us that they are likely the most important attributes pertaining to the prediction calculation. Humidity on the other hand, which has a high negative value tells us that it is also a significant part of the calculation, but it will be weighted negatively, lowering the calculation. Variables like “Visibility” and “Snowfall” have low coefficient values, meaning that when predicting using the model, it is going to give only a small impact on the predicted result.

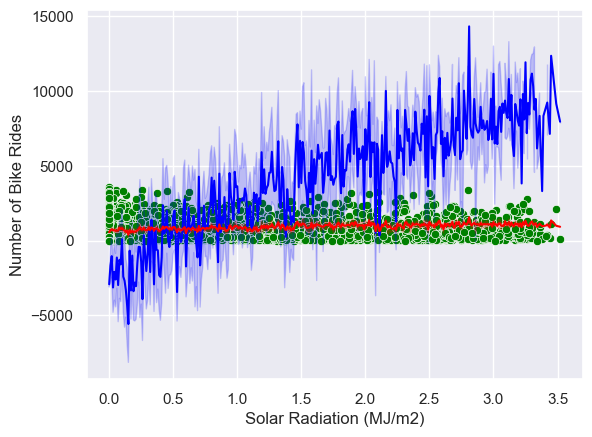
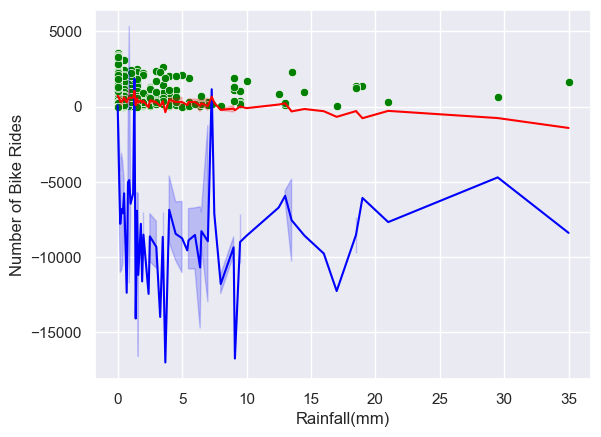


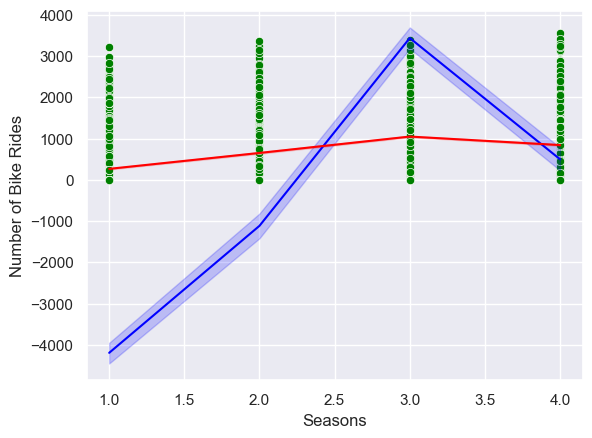
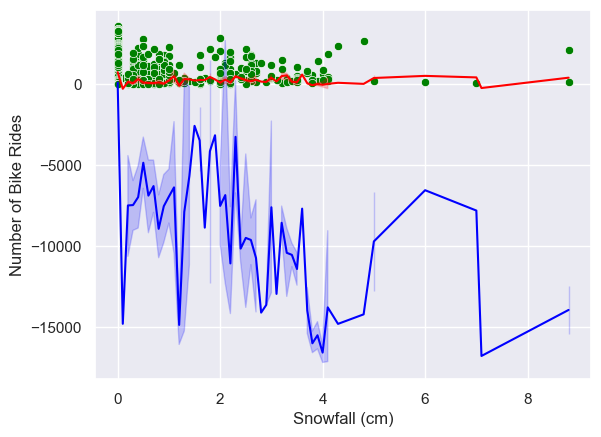
## 







In the figures above, Green dots are actual data points, red line is OLS model estimation, and blue line is the SGD Regression Model.

We can see the varying scales and weight types of the two model’s lines, and how they, while showing negatives and positives, or closer to 0 in OLS’s case, still correlate towards similar results in determining the importance of certain attributes’ and their values.

While the OLS model provides a slightly higher R-squared value, indicating better fit for average predictions, the SGDRegressor model is more robust in predicting actual values. This could be due to the stochastic nature of SGD, which allows it to escape local minima and potentially find a more global solution.

## Conclusion

To summarize, the analysis of the data to predict the bike sharing demand in Seoul using the two linear regression models, SGRRegressor and OLS, have been insightful. It highlighted some limitations of linear approaches for this specific dataset, much in tune with the saying that we do not live in a linear world.

While both SGDRegressor and OLS models resulted in a modest predictive power, with R-squared values around 0.47 to 0.48, their strengths have become apparent upon a closer look into the figures.

The OLS model was slightly more effective in capturing the average behavior based on the data, and it provides more statistics for us to explore, making it better suited for understanding trends and relationships among the variables.

On the other hand, the SGDRegressor’s stochastic nature made it more robust in real value prediction. It suggests that depending on the application or the dataset, if the goal is to provide accurate results, the model would be appropriate.

Overall, the assignment affirmed the importance of choosing the right models and the effect of micro tuning the hyper-parameters carefully, when the data does not strictly adhere to linear assumptions.