SeoulBikeData Report

Dataset

The dataset examined in this report was obtained from UCI Machine Learning Repository. It contains 8760 instances and 14 attributes.

- Input Variables:
    - 1 - Date: year-month-day
    - 3 - Hour: Hour of the Day
    - 4 - Temperature: Celsius
    - 5 - Humidity: Percent
    - 6 - Wind Speed: m/s
    - 7 - Visibility: 10 m
    - 8 - Dew point Temperature: Celsius
    - 9 - Solar Radiation: MJ/m^2
    - 10 - Rainfall: mm
    - 11 - Snowfall: cm
    - 12 - Seasons: Winter, Spring, Summer Autumn
    - 13 - Holiday: Holiday or No Holiday
    - 14 - Functional Day: NoFunc or Fun
- Response Variable:
    - 2 - Rented Bike Count: Count of Bikes rented at each hour

The input variable of "Functional Day" was eliminated due to every instance being 'Fun' and "Date" was eliminated.

Exploratory Data Analysis

Table 1. Summary Statistics of SeoulBikeData.csv.

	Rented Bike Count	Hour	Temperature	Humidity	Wind speed (m/s)	Visibility	Dew point temperature	Solar Radiation	Rainfall(mm)	Snowfall	Season	Holiday
mean	704.602055	12.50	12.88292	58.226	1.724	1436.82	4.073	1.569111	1.148687	1.075068	2.504110	1.0
std	644.997468	6.922582	11.944825	20.362413	1.036300	608.298712	13.060369	0.868746	1.128193	0.436746	1.114408	0.0
min	0	1.000	-17.8000	0	0	27.00	-30.6	1.00	1.00	1.00	1.00	1.0
25%	191.000000	6.750	3.500000	42.000	0.900	940.	-4.70	1.00	1.000	1.0000	2.00	1.0
50%	504.500000	12.50	13.70000	57.000	1.500	1698.	5.100	1.01	1.000	1.0000	3.00	1.0

75%	1065.250000	18.25	22.50000	74.000	2.300	2000.	14.80	1.93	1.000	1.0000	3.00	1.0
max	3556.000000	24.00	39.4000	98.000	7.400	2000.	27.20	4.52	36.00	9.8000	4.00	1.0

Table 1 displays basic statistics over the dataset. No missing data points or null features were found. For “Seasons” variables, each season was given a value (1,2,3,4) for Winter, Spring, Summer, Autumn. For “Holiday”, the no or yes response was changed to 1 and 2 respectively. As well “Hour”, “Solar Radiation”, “Rainfall” and “Snowfall” all had a numerical one added to each instance in order to eliminate zeros for proper use of some of the models.

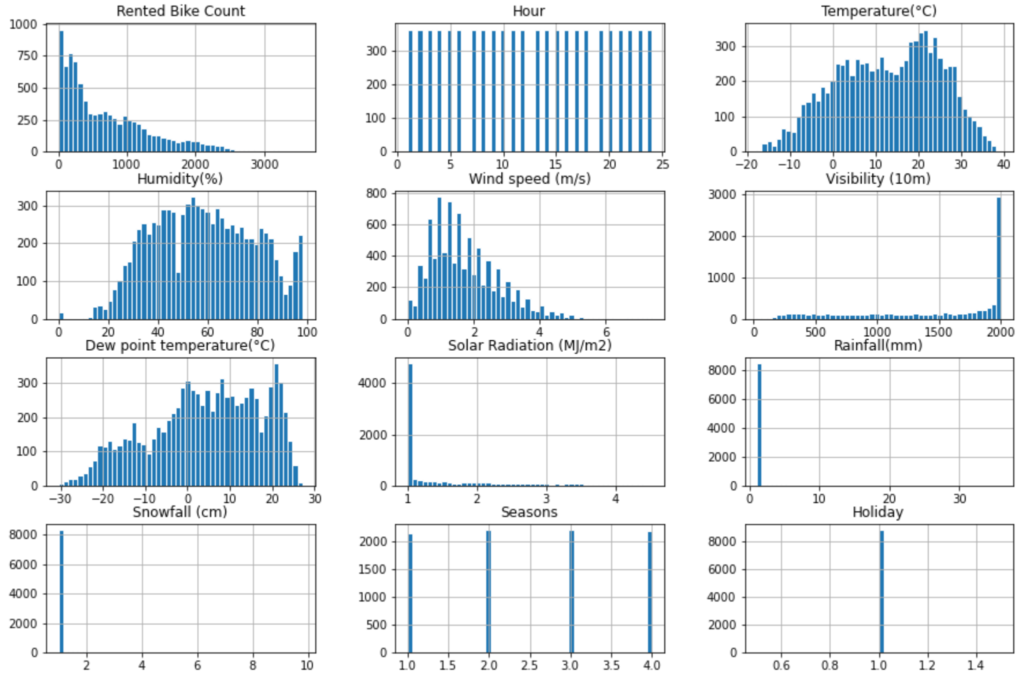

Figure 1: Histogram plots of the distribution of each parameter against the response variable.

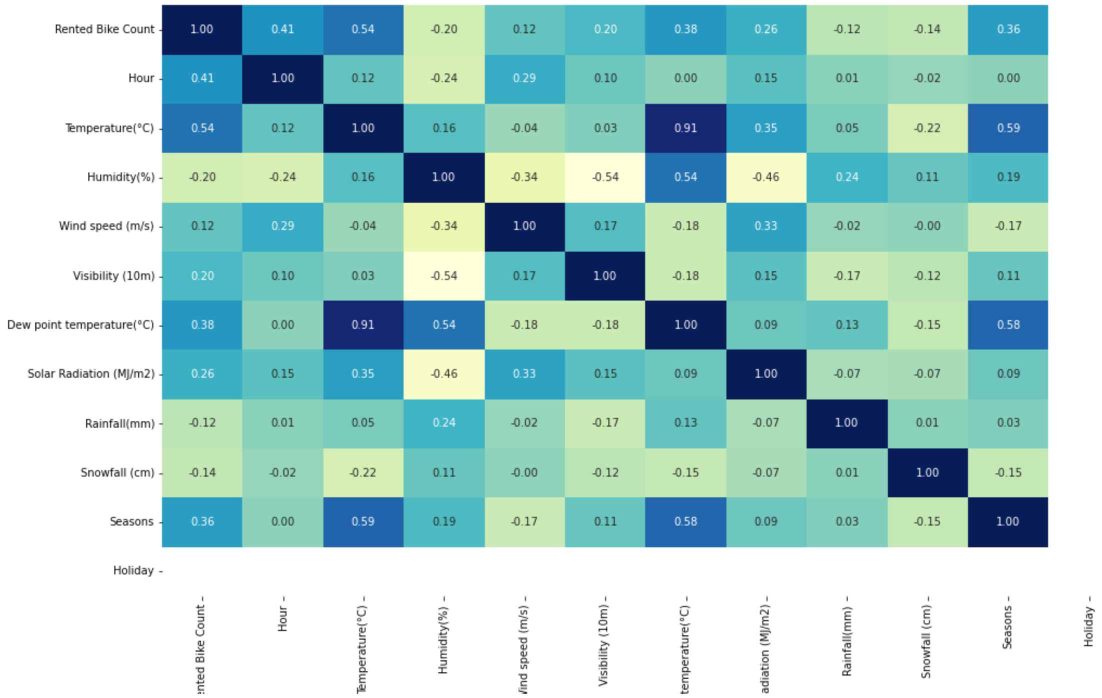

Figure 2: Correlation 'heatmap' of the SeoulBikeData.csv data file.

The histograms in Figure 1 allowed for preliminary insight on what possible variables would impact the response variable the most. It seems, "Temperature", "Dew Point" and "Humidity" show a close resemblance to a uniform distribution while the other variables are either multi-modal due to use of ordinal values or are skewed right like in "Wind Speed". Moreover, the correlation 'heatmap' displayed in Figure 2 gave insights into collinearity in the data set. There are several moderately strong positive correlations between two predictor variables, such as "Humidity" and "Dew Point Temperature". The only probable significant instance for multicollinearity is between "Temperature" and "Dew Point Temperature" which is logical due to "Dew Point Temperature" being a function of temperature itself. These observations, along with the variables that have strong positive correlations with the response variable, give insight on how feature selection algorithms used on the dataset will perform as well as how the models themselves will perform.

Feature Selection

In the Python file, three feature selection methods from sklearn were selected to be used on the models: forward, backward, and recursive selection. The forward and backward selection methods resulted in "Temperature", "Visibility", "Rainfall", "Snowfall" and "Holiday" being chosen to predict "Rented Bike Count". The recursive selection method instead used "Hour", "Wind Speed", "Rainfall", "Snowfall" and "Season" as predictor variables

In the Scala file, the used feature selection methods from Scalation were forward selection, backwards elimination and stepwise elimination. In comparison to how sklearn's feature selection works in Python, Scalation's feature selection was dependent on the model used and thus every model resulted in various used features. The forward selection models often opted to use "Hour","Temperature","Humidity", "Wind Speed", "Dew point temperature", "Rainfall","Snowfall" and "Seasons". Likewise, the stepwise elimination models used all the features that forward selection used. In comparison, the backward elimination models commonly used "Hour" and "Temperature".

Results

The full results, including features used in a model, R^2, adjusted R^2 and R^2 cross-validated , can be found in the accompanied summary file.

In the Scala component of the analysis, both the forward and backward selection methods had the same highest R^2 cross-validated value, 0.56617 from Symbolic Lasso (with both cross terms included) out of all of the models and feature selections tested. Similarly, the stepwise feature selection had its highest R^2 cross-validated value with Symbolic Lasso regression (0.56462) , although slightly lower than the other two feature selection methods. As well, R^2 adjusted was the highest out of all the models in Symbolic Lasso for each feature selection method. Overall, all the models with each feature selection performed either identically or relatively the same with the forward feature selection models tending to do the best.

In comparison to the Scala models, the Python models performed better with Linear, Ridge, Lasso and Quadratic regression. Both the forward and backward selection models performed the same as they chose the same features to be used. The quadratic model from both of these feature selection methods show both the highest R^2 adjusted and R^2 cross-validated with values of 0.5595 and 0.5568 respectively. The recursive selection method, on average, resulted in ~45% lower R^2 adjusted and cross-validated values. Out of all the models in the recursive selection, the quadratic model performed the best with an R^2 adjusted of 0.2982 and an R^2 cross-validated value of 0.2890

Overall it seems forward feature selection performs the best on this data set with Symbolic Lasso Regression performing the best, followed by the Quadratic regression models.

Discussion

Both languages provided moderately strong R^2 values, suggesting a decent relationship between the parameters and the response variable in the SeoulBikeData file. In countied work, a more systematic approach to the data set could yield with better models. Such approaches could include a more in depth exploration into the right-tailed skewing of most of the variables in comparison with the response variable (one such solution could be to mean center the data). Furthermore, a design of Symbolic Lasso regression in Python would be desired. As the quadratic model performed best in Python while in Scala Symbolic Lasso regression followed by Quadratic Lasso regression performed the best and therefore there is an incomplete comparison between the models in each language.