Prepared by: Aditya Sunil Choksi **Date: 07/06/2023**

BUAN 6341 Applied Machine Learning

Assignment 1

Executive Summary –

- **Linear regression analysis shows the impact of learning rate and threshold on convergence and Root Mean Square Error (RMSE). The model with all features performs the best.**
- **The experiments conducted using different learning rates and thresholds show that the model with all features performs the best in terms of predicting bike rentals, followed by the model with eight important variables. The model with randomly selected features performs the worst.**
- **Important findings include days with no bike rentals and the importance of feature selection and parameter tuning in machine learning.**
- **Insights from the analysis can inform future predictions and research in bike rental trends.**

Introduction –

In numerous urban cities, rental bikes have been implemented to improve mobility convenience. Ensuring the timely availability and accessibility of rental bikes is crucial in reducing waiting times and addressing the challenge of maintaining a stable supply. Accurately predicting the required bike count for each hour is a vital aspect of achieving this stability.

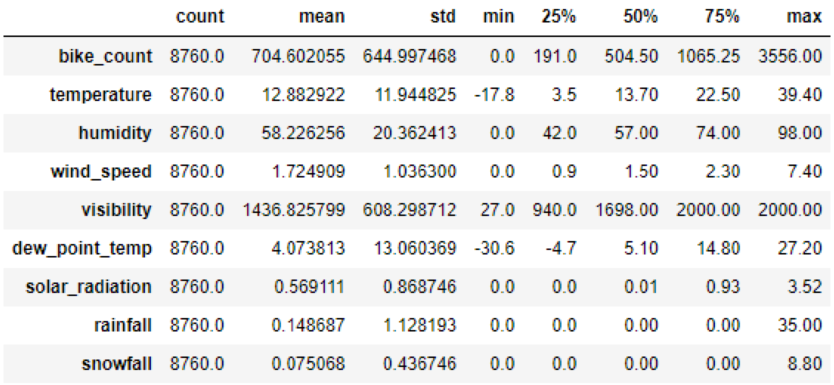
About the Data –

The dataset consists of 14 features and 8760 records. The dataset contains the number of bikes rented per hour and date information along with the weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall) which can be used to predict the bike count required at each hour for the stable supply of rental bikes.

Exploratory Data Analysis –

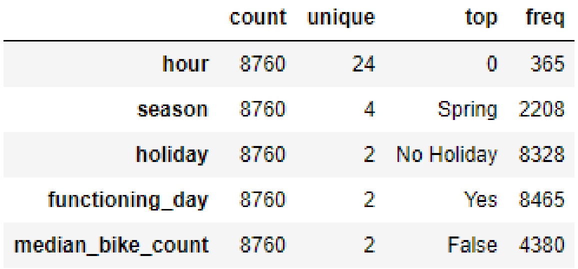
The Seoul Bike dataset offers an extensive range of data, presenting detailed records of bike rentals on an hourly basis throughout a year. Moreover, it includes accompanying weather conditions for each of those hours. Conducting a comprehensive examination of every variable within this dataset is crucial in order to gain a comprehensive understanding of its implications. Exploratory Data Analysis (EDA), an essential procedure in any data analysis undertaking, facilitates this process. EDA aids in uncovering variable distributions, detecting potential outlier values, and revealing interrelationships among different variables.

Summary of Numerical Variables –



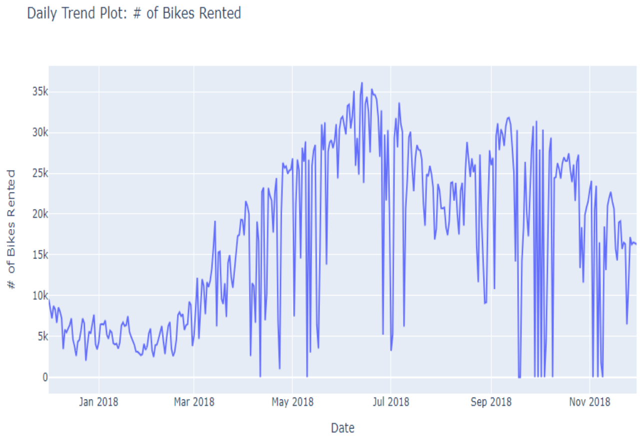
	count	mean	std	min	25%	50%	75%	max
bike_count	8760.0	704.602055	644.997468	0.0	191.0	504.50	1065.25	3556.00
temperature	8760.0	12.882922	11.944825	-17.8	3.5	13.70	22.50	39.40
humidity	8760.0	58.226256	20.362413	0.0	42.0	57.00	74.00	98.00
wind_speed	8760.0	1.724909	1.036300	0.0	0.9	1.50	2.30	7.40
visibility	8760.0	1436.825799	608.298712	27.0	940.0	1698.00	2000.00	2000.00
dew_point_temp	8760.0	4.073813	13.060369	-30.6	-4.7	5.10	14.80	27.20
solar_radiation	8760.0	0.569111	0.868746	0.0	0.0	0.01	0.93	3.52
rainfall	8760.0	0.148687	1.128193	0.0	0.0	0.00	0.00	35.00
snowfall	8760.0	0.075068	0.436746	0.0	0.0	0.00	0.00	8.80

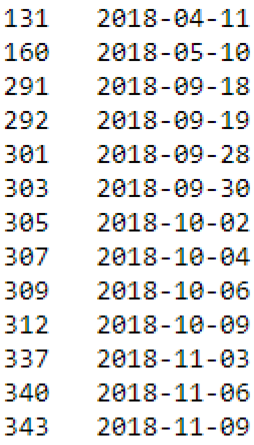
Summary of Categorical Variables -



	count	unique	top	freq
hour	8760	24	0	365
season	8760	4	Spring	2208
holiday	8760	2	No Holiday	8328
functioning_day	8760	2	Yes	8465
median_bike_count	8760	2	False	4380

Analysis of Date and Time Variables -



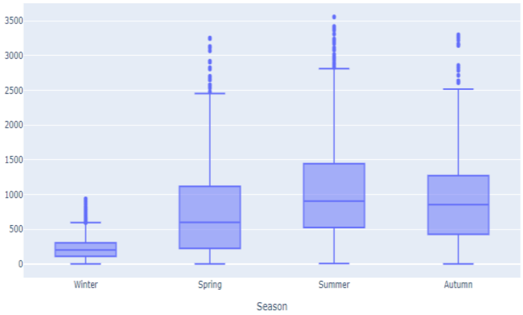


```
131    2018-04-11
160    2018-05-10
291    2018-09-18
292    2018-09-19
301    2018-09-28
303    2018-09-30
305    2018-10-02
307    2018-10-04
309    2018-10-06
312    2018-10-09
337    2018-11-03
340    2018-11-06
343    2018-11-09
```

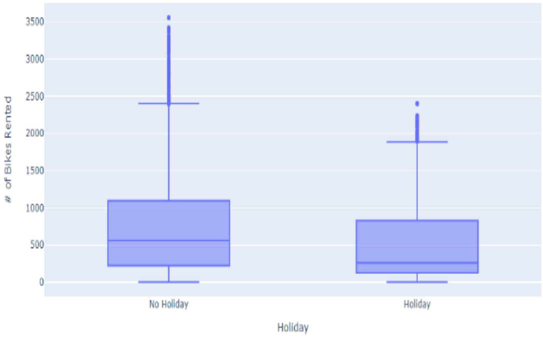
The trend plot above visually illustrates the daily bike rental patterns over time. It is noteworthy that there were specific days when no bikes were rented at all, which is unexpected as some level of demand is typically anticipated every day. This occurrence can be attributed to the 'functioning day' variable, which indicates whether the bike rental service operated on a particular day. On non-functioning days, the bike count is zero, clarifying the absence of rentals on those specific dates mentioned on the right side of the plot.

Implication - We do not need to include 'functioning_day' while training our models and on those days we do not predict the number of bikes rented.

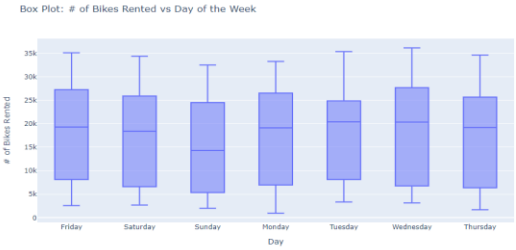
Furthermore, an additional insight derived from the trend plot is the negative impact of colder weather conditions (specifically during Winter, early Spring, and late Autumn) on bike rentals. This finding is further supported by the analysis of boxplots, which illustrate the relationship between the season and bike count.



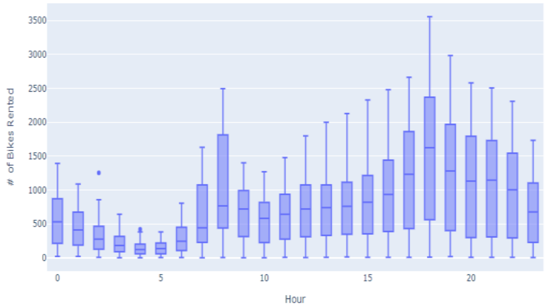
In the annual calendar of Seoul, there were a total of 18 holidays, including weekends. Notably, the box plots demonstrate a lower number of bike rentals on holidays compared to working days. This suggests that people tend to prefer staying at home during holidays, resulting in reduced bike rental activity. However, it is important to note that this correlation does not imply a causal relationship, as the bike rental sample does not represent the entire population.



Analyzing the relationship between the number of bikes rented and the days of the week can yield valuable insights, considering that holidays can occur on both weekdays and weekends. The median values of bike rentals are lower on weekends compared to weekdays. However, determining the significance of the expected value is beyond the scope of this report.



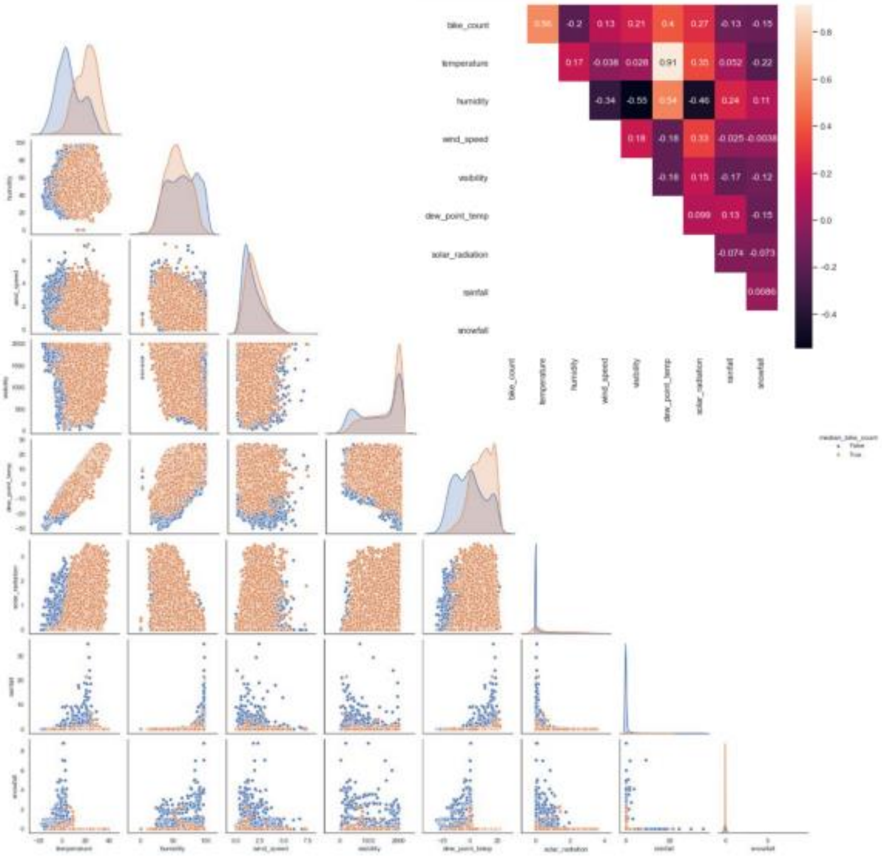
The dataset encompasses hourly data on bike counts, allowing insights into the influence of peak commute hours on the number of bikes rented. Given the speculation that bikes are primarily utilized for daily commutes, particularly during office hours, examining the hourly distribution of bike counts can provide valuable information. Analysis of the box plots reveals that bike sharing is most prominent between 5 p.m. and 10 p.m.



Implication - The dataset does not contain day of week as variable. Hence a new variable is created to include in machine learning models.

Weather Conditions –

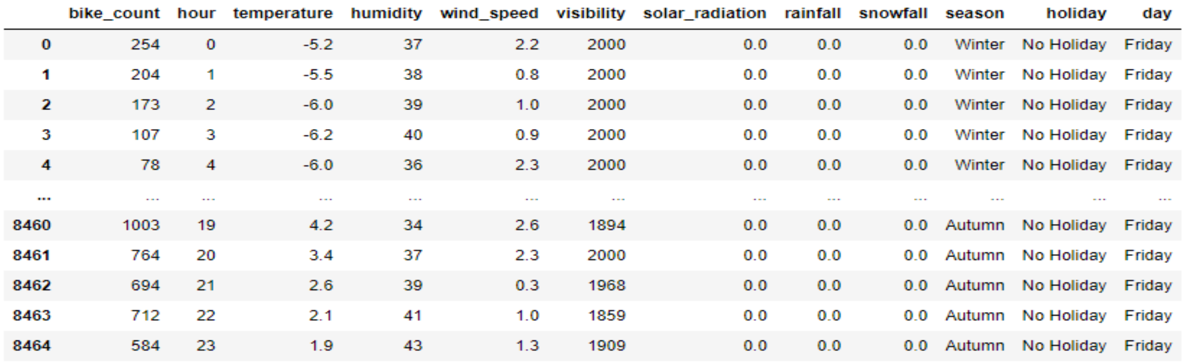
Correlation heatmap between weather characteristics and bike count are shown below.



Implication - The dew point temperature exhibits a strong correlation of 0.91 with temperature, raising concerns about potential multicollinearity issues if both variables are included in the model. However, after removing the dew point temperature, the variance inflation factor (VIF) for all remaining variables remains below 10. Previously, when including dew point temperature, the VIF values for temperature (87) and humidity (20), in addition to dew point temperature (116), exceeded the threshold of 10. Consequently, dew point temperature is excluded from the model construction process.

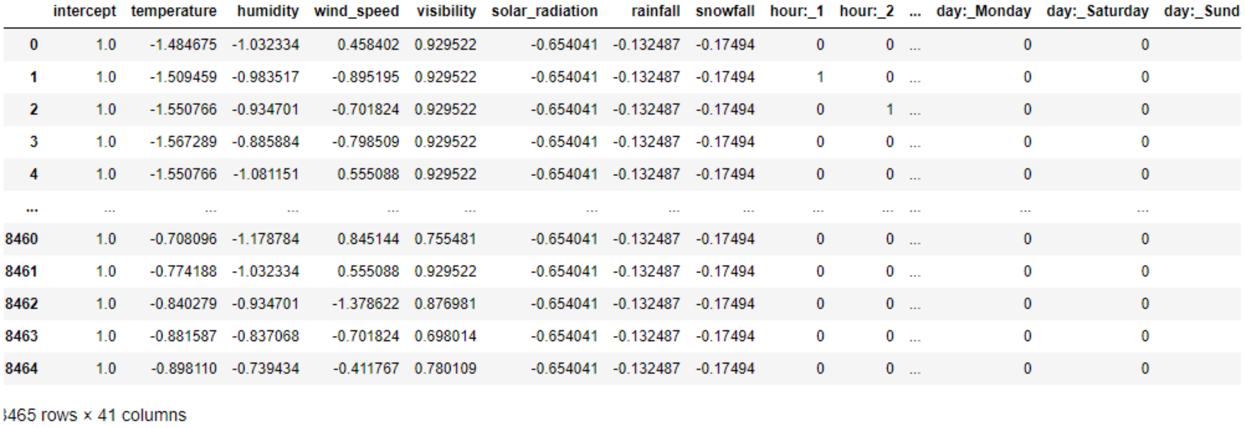
Data Preprocessing –

Final Dataset: The final dataset utilized for constructing the linear model consists of 12 variables, including 'day,' and excludes 'date,' 'functioning_day,' and 'dew_point_temp.' The dataset comprises 8465 records, excluding the non-functioning days.



	bike_count	hour	temperature	humidity	wind_speed	visibility	solar_radiation	rainfall	snowfall	season	holiday	day
0	254	0	-5.2	37	2.2	2000	0.0	0.0	0.0	Winter	No Holiday	Friday
1	204	1	-5.5	38	0.8	2000	0.0	0.0	0.0	Winter	No Holiday	Friday
2	173	2	-6.0	39	1.0	2000	0.0	0.0	0.0	Winter	No Holiday	Friday
3	107	3	-6.2	40	0.9	2000	0.0	0.0	0.0	Winter	No Holiday	Friday
4	78	4	-6.0	36	2.3	2000	0.0	0.0	0.0	Winter	No Holiday	Friday
...	...	...	...	...	...	...	...	...	...	...	...	...
8460	1003	19	4.2	34	2.6	1894	0.0	0.0	0.0	Autumn	No Holiday	Friday
8461	764	20	3.4	37	2.3	2000	0.0	0.0	0.0	Autumn	No Holiday	Friday
8462	694	21	2.6	39	0.3	1968	0.0	0.0	0.0	Autumn	No Holiday	Friday
8463	712	22	2.1	41	1.0	1859	0.0	0.0	0.0	Autumn	No Holiday	Friday
8464	584	23	1.9	43	1.3	1909	0.0	0.0	0.0	Autumn	No Holiday	Friday

The numeric features are normalized, and the categorical variables are one-hot encoded as shown below.



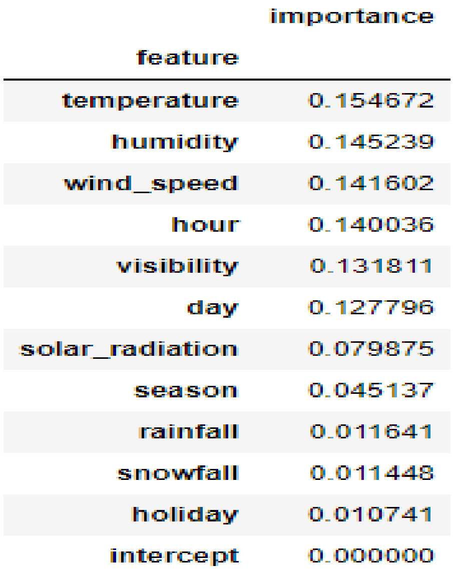
	intercept	temperature	humidity	wind_speed	visibility	solar_radiation	rainfall	snowfall	hour:_1	hour:_2	...	day:_Monday	day:_Saturday	day:_Sund
0	1.0	-1.484675	-1.032334	0.458402	0.929522	-0.654041	-0.132487	-0.17494	0	0	...	0	0	
1	1.0	-1.509459	-0.983517	-0.895195	0.929522	-0.654041	-0.132487	-0.17494	1	0	...	0	0	
2	1.0	-1.550766	-0.934701	-0.701824	0.929522	-0.654041	-0.132487	-0.17494	0	1	...	0	0	
3	1.0	-1.567289	-0.885884	-0.798509	0.929522	-0.654041	-0.132487	-0.17494	0	0	...	0	0	
4	1.0	-1.550766	-1.081151	0.555088	0.929522	-0.654041	-0.132487	-0.17494	0	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
8460	1.0	-0.708096	-1.178784	0.845144	0.755481	-0.654041	-0.132487	-0.17494	0	0	...	0	0	
8461	1.0	-0.774188	-1.032334	0.555088	0.929522	-0.654041	-0.132487	-0.17494	0	0	...	0	0	
8462	1.0	-0.840279	-0.934701	-1.378622	0.876981	-0.654041	-0.132487	-0.17494	0	0	...	0	0	
8463	1.0	-0.881587	-0.837068	-0.701824	0.698014	-0.654041	-0.132487	-0.17494	0	0	...	0	0	
8464	1.0	-0.898110	-0.739434	-0.411767	0.780109	-0.654041	-0.132487	-0.17494	0	0	...	0	0	

8465 rows × 41 columns

Randomly selected features –

To evaluate the training and test errors against the full model, eight random features are chosen for model construction. The selected features include 'day,' 'temperature,' 'hour,' 'visibility,' 'holiday,' 'rainfall,' 'solar_radiation,' and 'season.'

Important Features - The top eight features are determined using a random forest classifier. The default method for calculating variable importance is the mean decrease in impurity (or Gini importance). This mechanism assigns an importance measure to the splitting variable based on the improvement in the split-criterion at each split in each tree. The importance measures are then accumulated across all trees in the forest, resulting in a table displaying the features in decreasing order of importance. The selected features include 'temperature,' 'humidity,' 'wind_speed,' 'hour,' 'visibility,' 'day,' 'solar_radiation,' and 'season.'



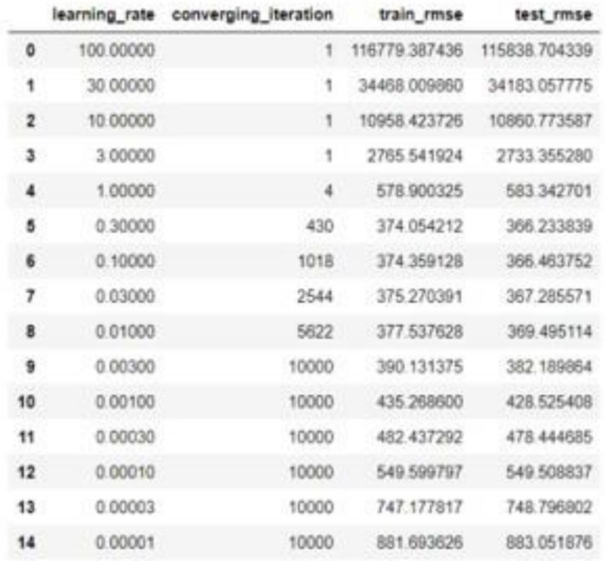
	importance
feature	
temperature	0.154672
humidity	0.145239
wind_speed	0.141602
hour	0.140036
visibility	0.131811
day	0.127796
solar_radiation	0.079875
season	0.045137
rainfall	0.011641
snowfall	0.011448
holiday	0.010741
intercept	0.000000

Train and test sets - The datasets are randomly split in 70/30 ratio to create training and test sets.

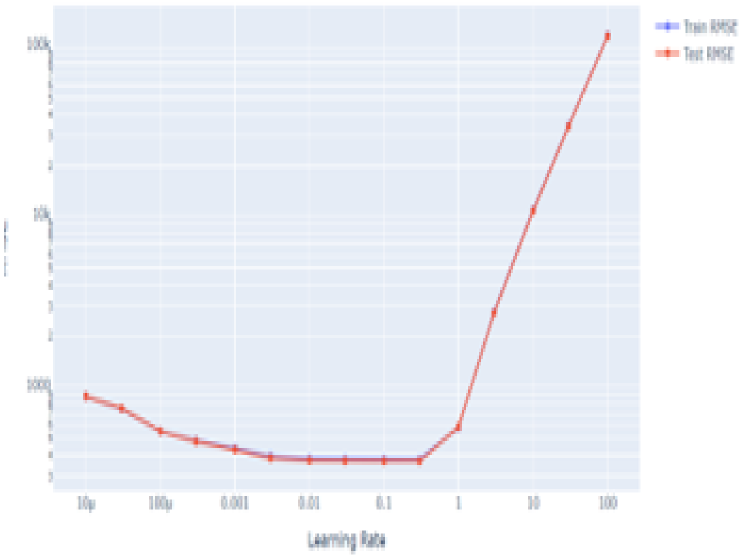
Linear Regression –

The gradient descent algorithm used in this analysis has a maximum limit of 10,000 iterations. This is a common practice in machine learning, as it prevents the algorithm from running indefinitely if it is unable to find a solution. The threshold in this context refers to the minimum percentage change in the cost function that is required at each update. If the change in the cost function is less than this threshold, the gradient descent algorithm terminates the loop and reports the cost function and thetas at that point. This point is considered to be the 'convergence' of the algorithm, as it indicates that the algorithm has found the minimum of the cost function (or at least, a point that is close enough to the minimum based on the threshold).

Experiment 1 - For threshold = 0.001



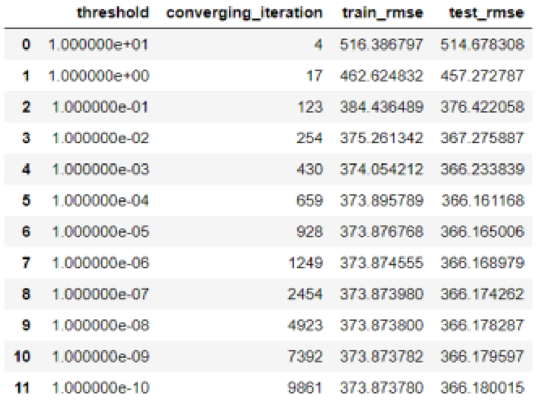
	learning_rate	converging_iteration	train_rmse	test_rmse
0	100.00000	1	116779.387436	115838.704339
1	30.00000	1	34468.009860	34183.057775
2	10.00000	1	10958.423726	10860.773587
3	3.00000	1	2765.541924	2733.355280
4	1.00000	4	578.900325	583.342701
5	0.30000	430	374.054212	366.233839
6	0.10000	1018	374.359128	366.463752
7	0.03000	2544	375.270391	367.285571
8	0.01000	5622	377.537628	369.495114
9	0.00300	10000	390.131375	382.189864
10	0.00100	10000	435.268600	428.525408
11	0.00030	10000	482.437292	478.444685
12	0.00010	10000	549.599797	549.508837
13	0.00003	10000	747.177817	748.796802
14	0.00001	10000	881.693626	883.051876



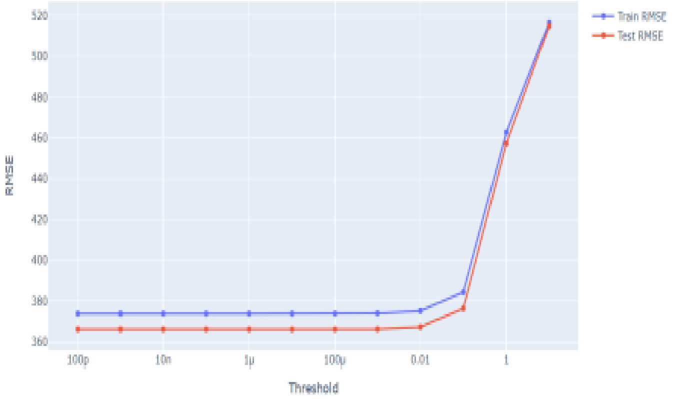
Learning Rate vs Train and Test RMSE

Using higher learning rates, the gradient descent fails to converge to the minimum and instead overshoots, resulting in substantially high train and test root mean square error (RMSE) values, with convergence occurring in less than five iterations. Conversely, lower learning rates require a greater number of iterations for convergence. As the maximum iteration is limited to 10,000, the algorithm does not reach the global minimum, leading to relatively higher train and test RMSE values compared to the optimal learning rate. In this analysis, a learning rate of 0.3 is selected as the optimum choice.

Experiment 2- For learning rate = 0.3



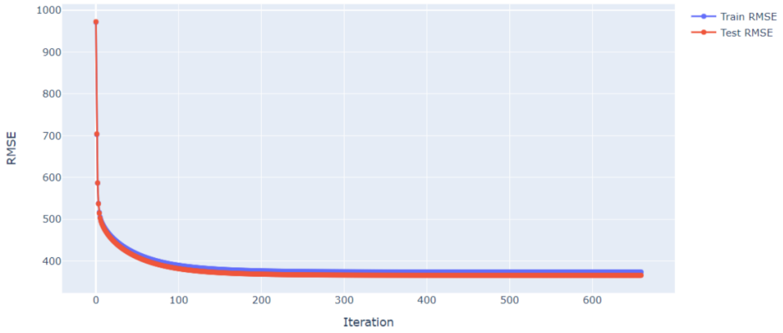
	threshold	converging_iteration	train_rmse	test_rmse
0	1.000000e+01	4	516.386797	514.678308
1	1.000000e+00	17	462.624832	457.272787
2	1.000000e-01	123	384.436489	376.422058
3	1.000000e-02	254	375.261342	367.275887
4	1.000000e-03	430	374.054212	366.233839
5	1.000000e-04	659	373.895789	366.161168
6	1.000000e-05	928	373.876768	366.165006
7	1.000000e-06	1249	373.874555	366.168979
8	1.000000e-07	2454	373.873980	366.174262
9	1.000000e-08	4923	373.873800	366.178287
10	1.000000e-09	7392	373.873782	366.179597
11	1.000000e-10	9861	373.873780	366.180015



Threshold for Convergence vs Train and Test RMSE (Learning Rate = 0.3)

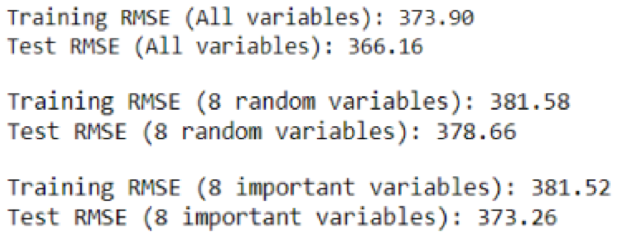
The threshold parameter defines the minimum percentage change required in the cost function. Using higher thresholds may prevent the gradient descent algorithm from converging at the global minimum. As the threshold decreases, the change in the cost function becomes less significant. However, excessively small thresholds prolong the convergence process, requiring a larger number of gradient descent iterations. Decreasing the threshold beyond the optimal value does not have a substantial impact on the train and test RMSE. In this analysis, the optimal threshold is determined to be 0.0001. The plot below illustrates the train and test RMSE at each iteration for a chosen learning rate of 0.3 and a threshold of 0.0001.

Train and Test RMSE at various Iterations (Learning Rate = 0.3, Threshold = 0.0001)



The cost function consistently decreases with each iteration, resulting in a reduction in both the train and test root mean square error (RMSE) values. The gradient descent algorithm terminates at the 659th iteration, reaching the minimum cost function and providing the corresponding thetas.

Experiment 3 and 4 - For learning rate = 0.3 and threshold = 0.0001



```
Training RMSE (All variables): 373.90
Test RMSE (All variables): 366.16

Training RMSE (8 random variables): 381.58
Test RMSE (8 random variables): 378.66

Training RMSE (8 important variables): 381.52
Test RMSE (8 important variables): 373.26
```

The model that incorporates all features exhibits the lowest train and test root mean square error (RMSE) values among all the models. It is followed by the model containing 8 important variables, which also performs well but not as effectively. In contrast, the model with randomly selected features demonstrates the highest train and test RMSE. This disparity can be attributed to the fact that the features selected through the feature importance method offer better explanations for the variation in the target variable compared to randomly selected features. These results support the notion that including important variables leads to a greater mean decrease in impurity (measured by Gini index) and enhances predictive power.

Questions –

What do you think matters the most for predicting the rented bike count?

The categorical variables related to date and time, including hour, season, and day of the week, along with weather conditions such as temperature, humidity, wind speed, and visibility, are highly significant predictors. Notably, bike rentals experience a substantial increase during the summer season, and between 5 p.m. and 10 p.m. daily, there is a noticeable surge in business. Weekdays witness higher bike usage among commuters compared to weekends and holidays**.**

What other steps you could have taken with regards to modelling to get better results?

The 'bike_count' variable exhibits a significant right skew, which introduces heteroskedastic errors when applying a linear regression model. To address this issue, variable transformation can be employed, although it alters the interpretation of the coefficients. Alternatively, treating the dataset as a time series model and implementing multivariate time series forecasting models can provide a more effective approach to modeling.