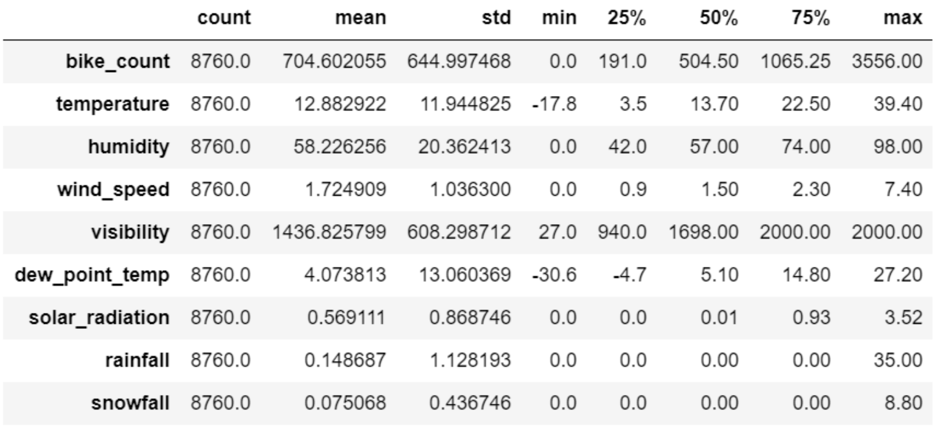
Assignment 1

A. Exploratory Data Analysis:

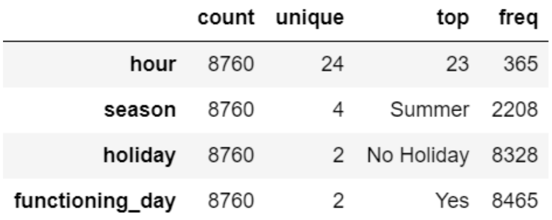
The Seoul Bike dataset contains information regarding the number of bikes rented on an hourly basis for a year and includes prevailing weather conditions. It is important to analyze every variable to understand the dataset better. EDA will provide insights into the distributions of variables, outliers points and, correlation between 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

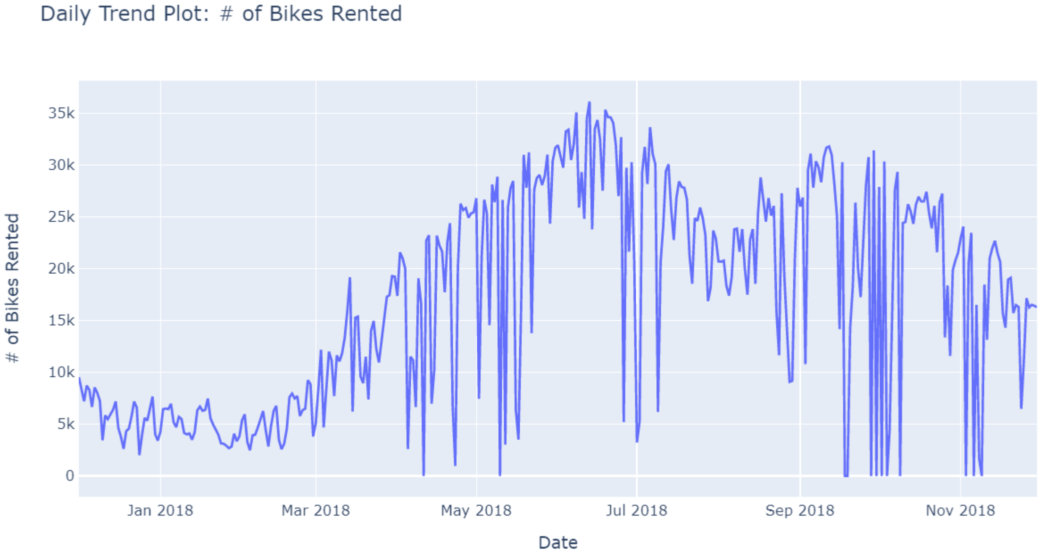
Summary of Categorical Variables:

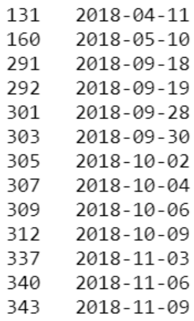


	count	unique	top	freq
hour	8760	24	23	365
season	8760	4	Summer	2208
holiday	8760	2	No Holiday	8328
functioning_day	8760	2	Yes	8465

Implication: The 'hour' variable is considered as a categorical variable and its explanation is given in the next section.

A1. Date and Time Variables:



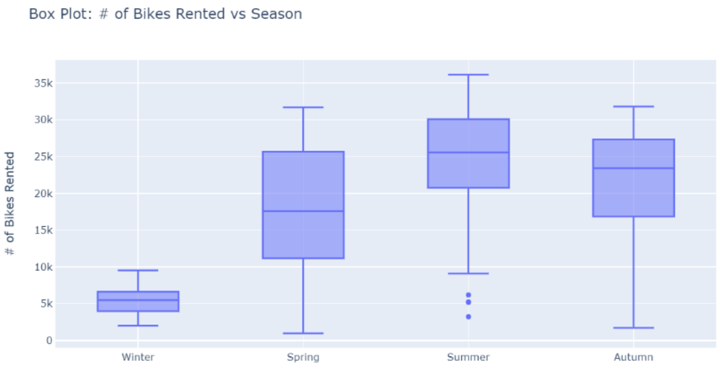


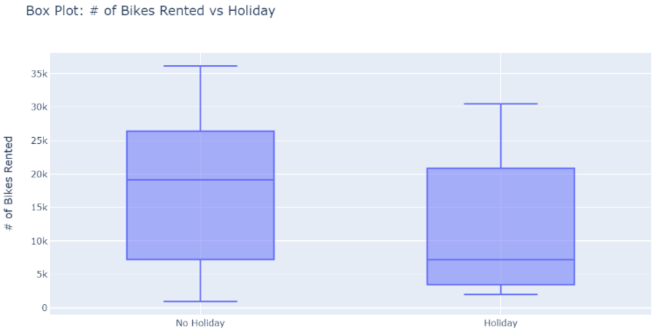
```
131    2018-04-11
160    2018-05-10
291    2018-09-18
292    2018-09-19
301    2018-09-28
303    2018-09-30
305    2018-10-02
307    2018-10-04
309    2018-10-06
312    2018-10-09
337    2018-11-03
340    2018-11-06
343    2018-11-09
```

The above trend-plot shows the number of bikes rented on a particular day. As shown in the plot, there are certain days when no bike is rented. By looking at functioning day variable, one can see that on non-functioning days, bike count is 0. The list of days on the right correspond to the non-functioning dates.

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

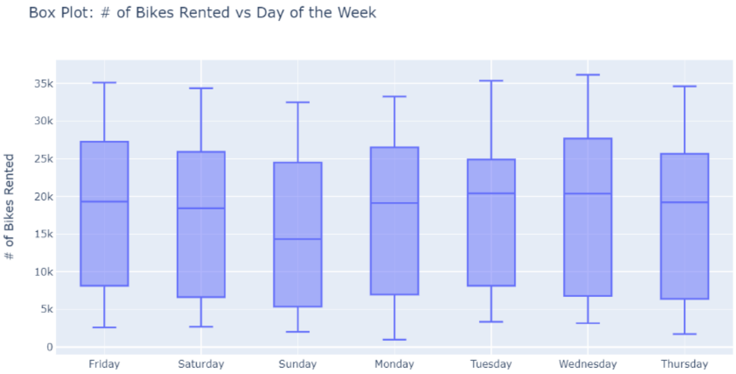
Another observation from the trend plot is that colder weather conditions (entire Winter, early Spring, and late Autumn) hamper bike rentals. This fact is also corroborated from the set of boxplots between season and bike count.

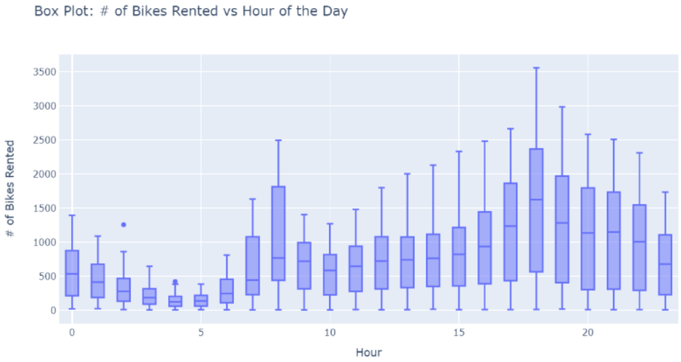




There were a total of 18 holidays in a yearly calendar in Seoul (may or may not include weekend days). It is interesting to see that there are lesser bikes rented on holidays compared to working days as shown in the set of box-plots. It shows that people generally rent bikes to commute and most of them prefer to stay at home during holidays. However, this correlation does not imply causal relationship because pupils who rent bike are not a sample of entire population.

Since holidays may or may not fall on weekends, analysis on number of bikes rented with respect to the days of a week can provide crucial insights. The median values are lower on the weekends compared to the weekdays, but one needs to test the significance of the expected value which is beyond the scope of this report.



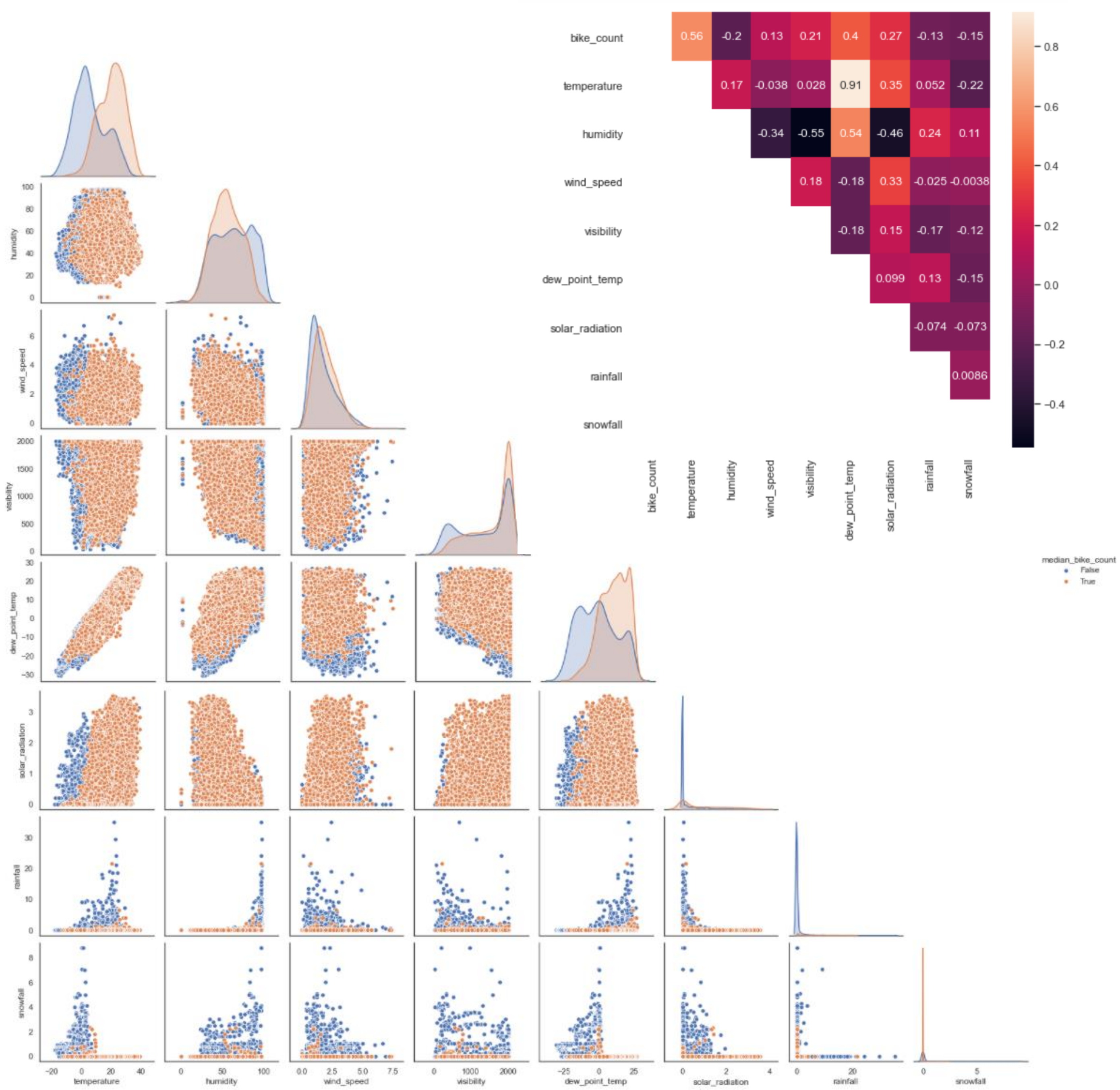


The dataset contains hourly information of bike counts. Since bikes are speculated to be rented for daily commutes (especially around office hours), hourly distribution of bike count can shed some light on the effect of peak commute hours on number of bikes rented. From the set of box-plots, the bike sharing is predominant from 5 p.m. to 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.

A2. Weather Conditions:

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

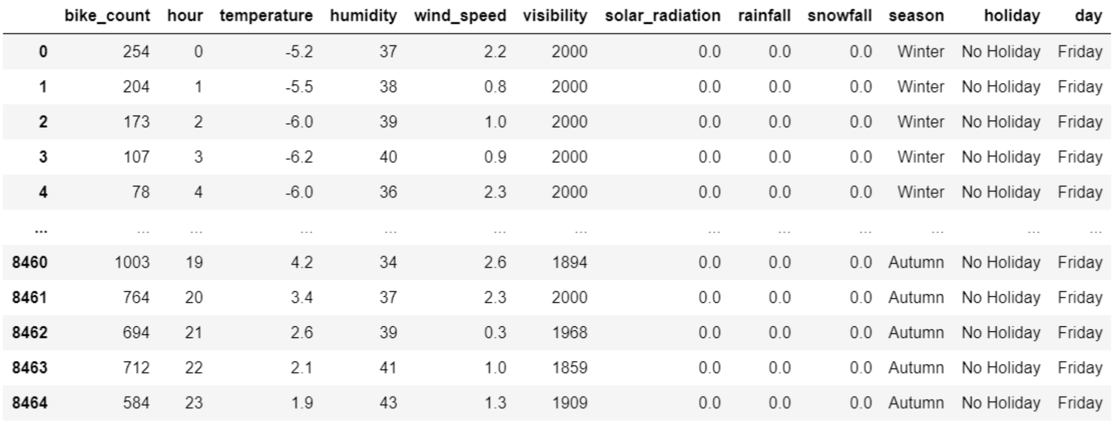


Implication: Dew point temperature is highly correlated with temperature (0.91) and including both these variables might lead to multi-collinearity issues while modelling. In addition, the variance inflation factor of all variables is less than 10 after removing dew-point temperature. Previously, by including that variable, two variables namely – temperature (87) and humidity (20) have VIF greater than 10 in addition to dew point temperature (116). Hence, dew point temperature is not included in the model building.

B. Data Preprocessing:

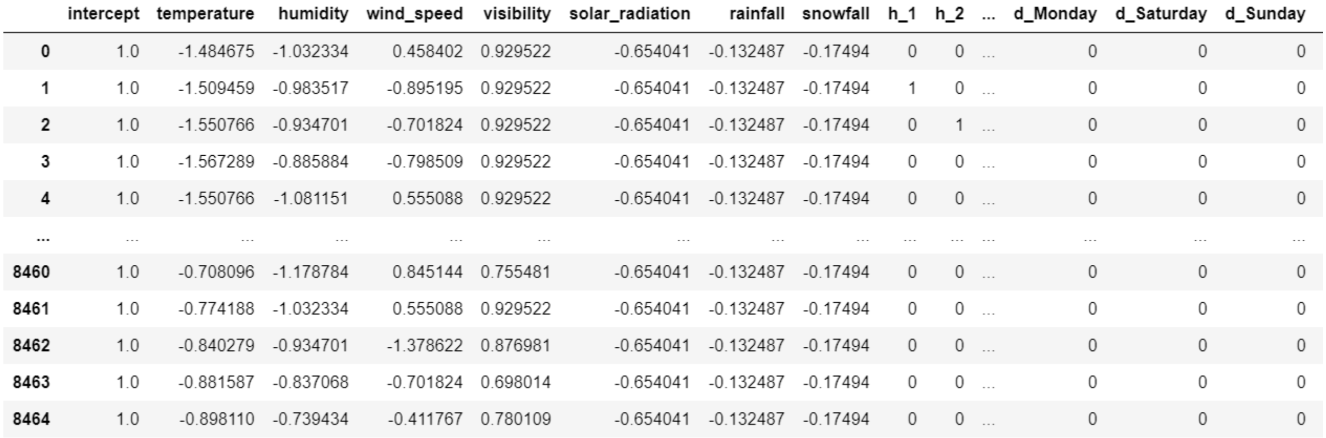
B1. Final Dataset:

The final dataset that is used for building linear and logistic models contains 12 variables (included: [‘day’], excluded: [‘date’, ‘functioning_day’, ‘dew_point_temp’]) with 8465 records (which excludes 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	h_1	h_2	...	d_Monday	d_Saturday	d_Sunday
0	1.0	-1.484675	-1.032334	0.458402	0.929522	-0.654041	-0.132487	-0.17494	0	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	0
2	1.0	-1.550766	-0.934701	-0.701824	0.929522	-0.654041	-0.132487	-0.17494	0	1	...	0	0	0
3	1.0	-1.567289	-0.885884	-0.798509	0.929522	-0.654041	-0.132487	-0.17494	0	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	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8460	1.0	-0.708096	-1.178784	0.845144	0.755481	-0.654041	-0.132487	-0.17494	0	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	0
8462	1.0	-0.840279	-0.934701	-1.378622	0.876981	-0.654041	-0.132487	-0.17494	0	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	0
8464	1.0	-0.898110	-0.739434	-0.411767	0.780109	-0.654041	-0.132487	-0.17494	0	0	...	0	0	0

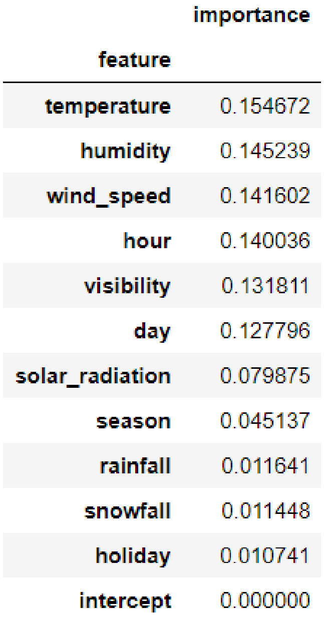
B2. Randomly Selected Features:

8 random features are selected for building models and to compare the training and test errors with the full model. The selected 8 features are: ['day', 'temperature', 'hour', 'visibility', 'holiday', 'rainfall', 'solar_radiation', 'season']

B3. Important Features:

Top 8 features are selected using random forest classifier. The default method to compute variable importance is the mean decrease in impurity (or Gini importance) mechanism: At each split in each tree, the improvement in the split-criterion is the importance measure attributed to the splitting variable and is accumulated over all the trees in the forest separately for each variable. The table to the right shows the decreasing order of feature importance.

Selected features are: ['temperature', 'humidity', 'wind_speed', 'hour', 'visibility', 'day', 'solar_radiation', '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

B4. Train and Test Sets:

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

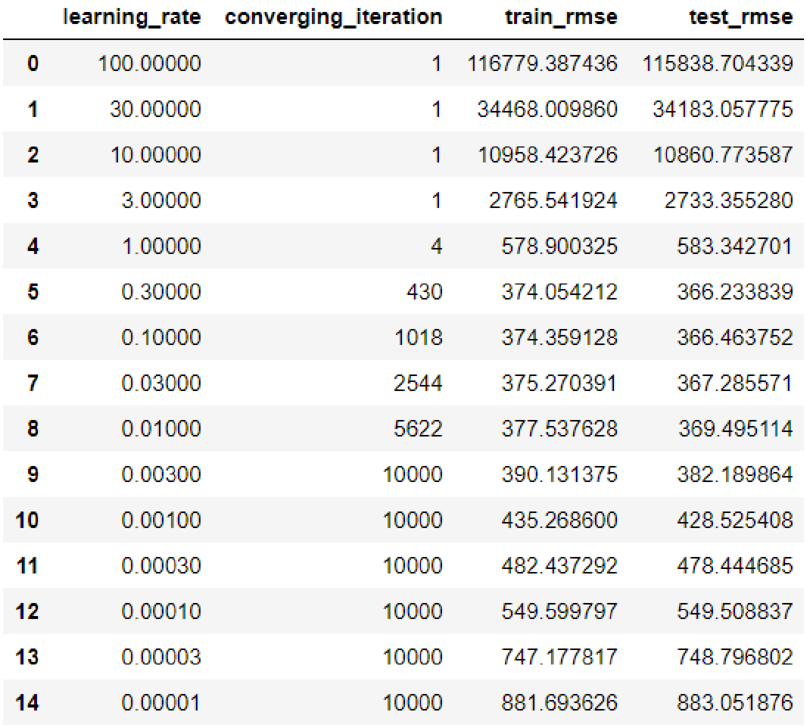
C. Linear Regression:

Maximum number of iterations are capped at 10000.

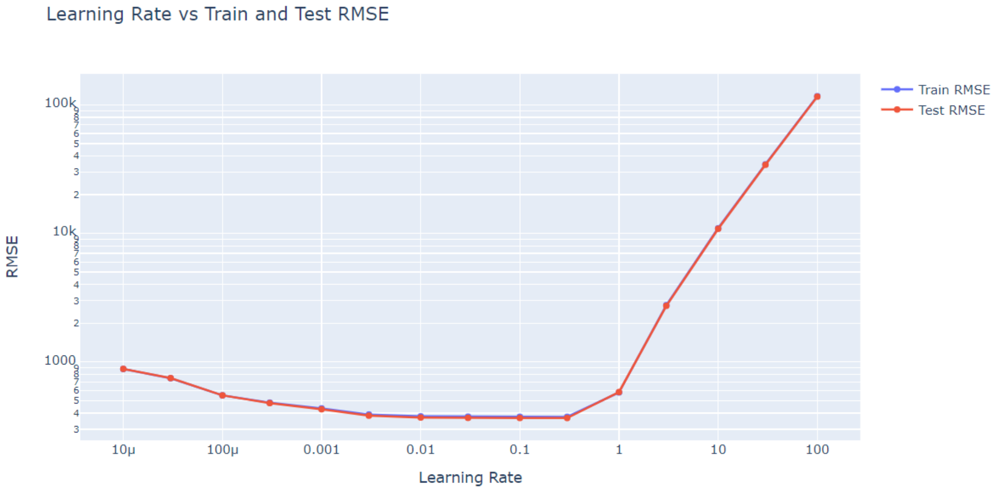
Threshold determines the minimum percentage change in cost function required at each update. If the change is lower than threshold, gradient descent comes out of loop and reports the cost function and thetas at that point (convergence).

C1. 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

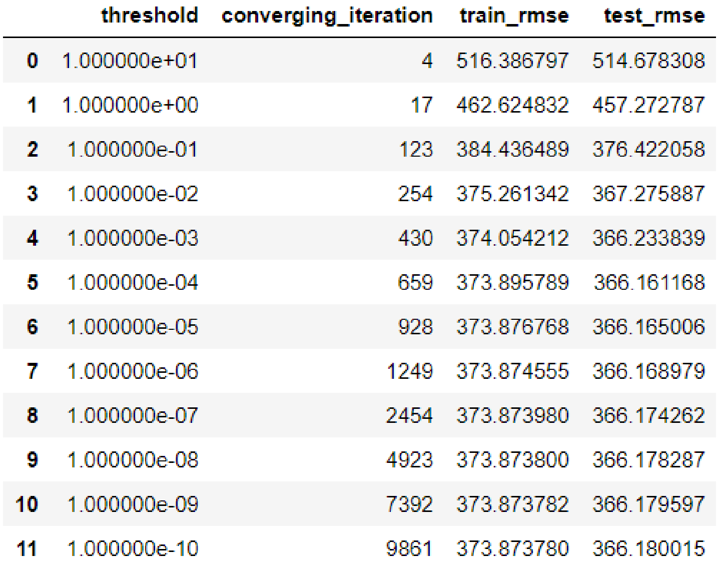


For higher learning rates, the gradient descent does not converge to minimum, and it overshoots. As a result, both train and test RMSE are very high and converging iteration is less than 5. On the other hand, when the learning rate is very low, the gradient descent requires more iterations to converge. Since the maximum iteration is capped at 10000, it has not converged to global minimum and therefore the train and test RMSE are comparatively higher than that at the optimum learning rate.

The chosen learning rate is 0.3.

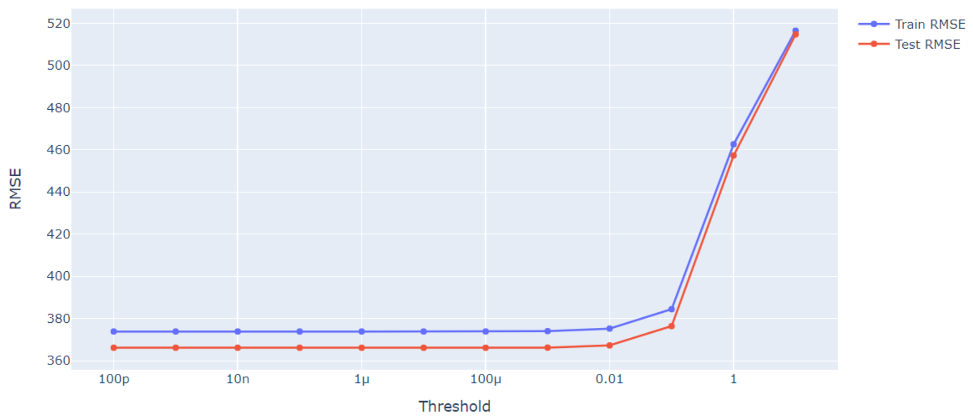
C2. Experiment 2:

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)

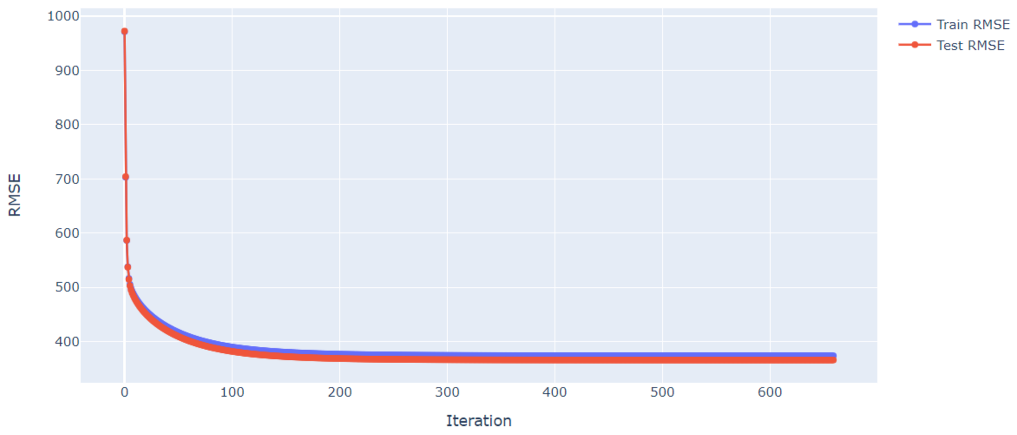


Threshold determines the minimum cut-off percentage change in cost function. Higher thresholds might lead to gradient descent not converging at global minimum. As the threshold decreases, there is not much change in the cost function. However, very small thresholds would take a longer time to converge (lot of gradient descent iterations). The train and test RMSE do not change much by decreasing the threshold any further than optimum threshold.

The optimum threshold is 0.0001.

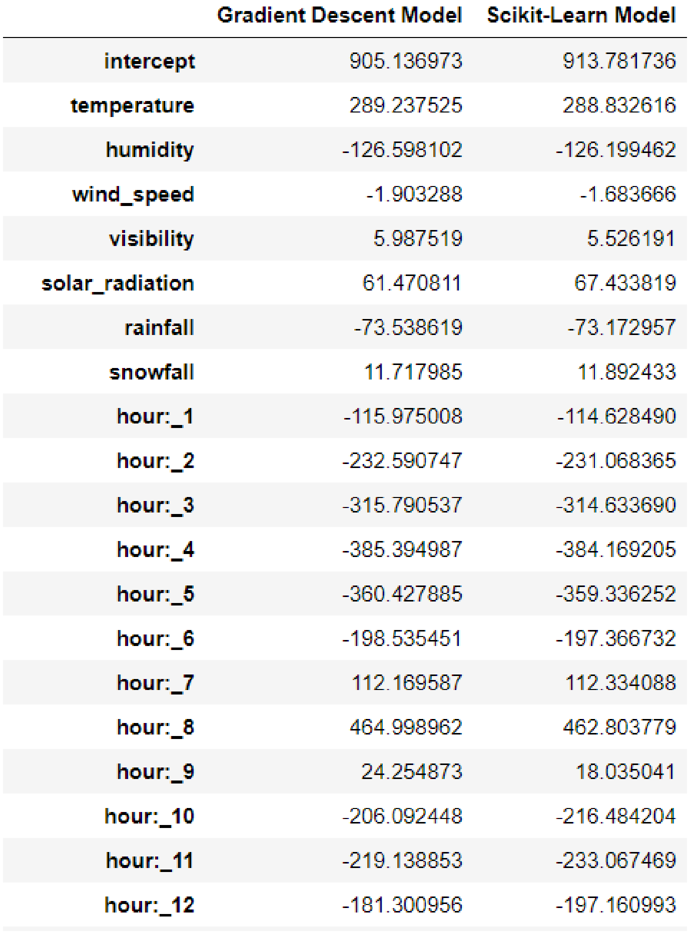
The below plot shows the train and test RMSE at every iteration for chosen learning rate of 0.3 and threshold of 0.0001.

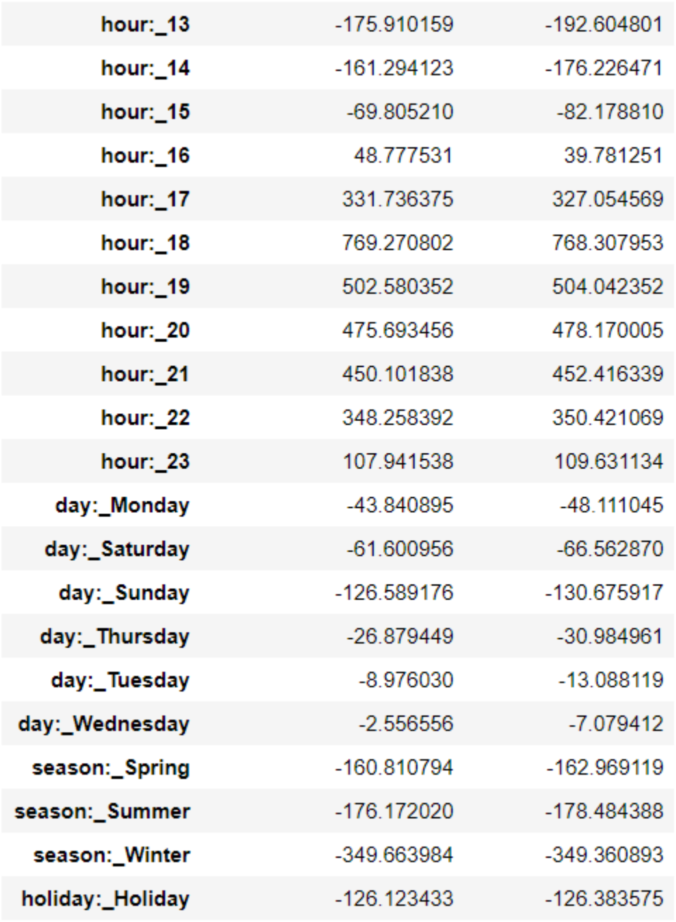
Train and Test RMSE at various Iterations (Learning Rate = 0.3, Threshold = 0.0001)



The cost function decreases at every iteration so does the train and test RMSE. At 659th iteration, the gradient descent comes out of the loop with minimum cost function and reports the thetas at that point.

Thetas of our Linear Regression Model are compared to that of Scikit-Learn Linear Regression Model. Below is the table that summarizes the coefficients.

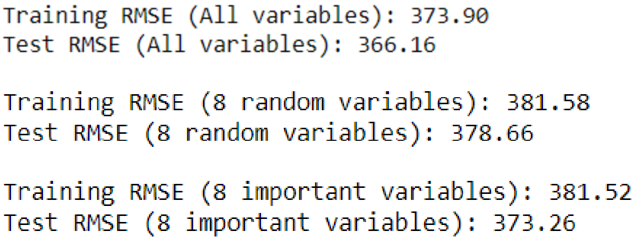




	Gradient Descent Model	Scikit-Learn Model
intercept	905.136973	913.781736
temperature	289.237525	288.832616
humidity	-126.598102	-126.199462
wind_speed	-1.903288	-1.683666
visibility	5.987519	5.526191
solar_radiation	61.470811	67.433819
rainfall	-73.538619	-73.172957
snowfall	11.717985	11.892433
hour:_1	-115.975008	-114.628490
hour:_2	-232.590747	-231.068365
hour:_3	-315.790537	-314.633690
hour:_4	-385.394987	-384.169205
hour:_5	-360.427885	-359.336252
hour:_6	-198.535451	-197.366732
hour:_7	112.169587	112.334088
hour:_8	464.998962	462.803779
hour:_9	24.254873	18.035041
hour:_10	-206.092448	-216.484204
hour:_11	-219.138853	-233.067469
hour:_12	-181.300956	-197.160993
hour:_13	-175.910159	-192.604801
hour:_14	-161.294123	-176.226471
hour:_15	-69.805210	-82.178810
hour:_16	48.777531	39.781251
hour:_17	331.736375	327.054569
hour:_18	769.270802	768.307953
hour:_19	502.580352	504.042352
hour:_20	475.693456	478.170005
hour:_21	450.101838	452.416339
hour:_22	348.258392	350.421069
hour:_23	107.941538	109.631134
day:_Monday	-43.840895	-48.111045
day:_Saturday	-61.600956	-66.562870
day:_Sunday	-126.589176	-130.675917
day:_Thursday	-26.879449	-30.984961
day:_Tuesday	-8.976030	-13.088119
day:_Wednesday	-2.556556	-7.079412
season:_Spring	-160.810794	-162.969119
season:_Summer	-176.172020	-178.484388
season:_Winter	-349.663984	-349.360893
holiday:_Holiday	-126.123433	-126.383575

C3. 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 train and test RMSE of the model containing all features is the lowest of all, followed by the model containing 8 important variables. The model containing random features has the highest train and test RMSE. This can be attributed to the fact that features included in feature importance method explain the variation in target variable better than the one that were selected randomly. This proves the fact that by adding important variables, there is greater mean decrease in impurity (Gini) and better predictive power.

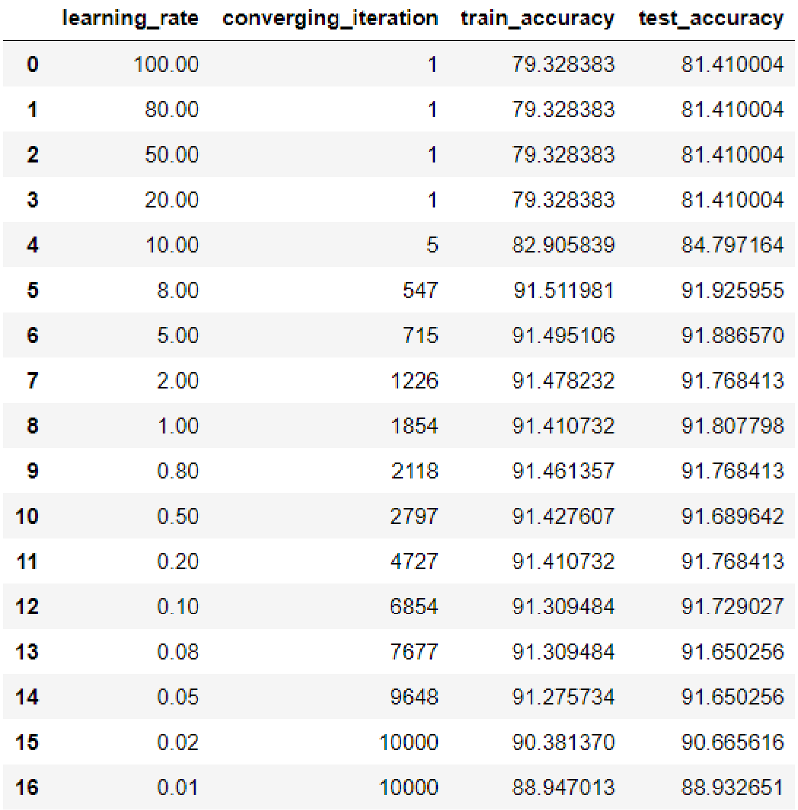
D. Logistic Regression:

Maximum number of iterations are capped at 10000.

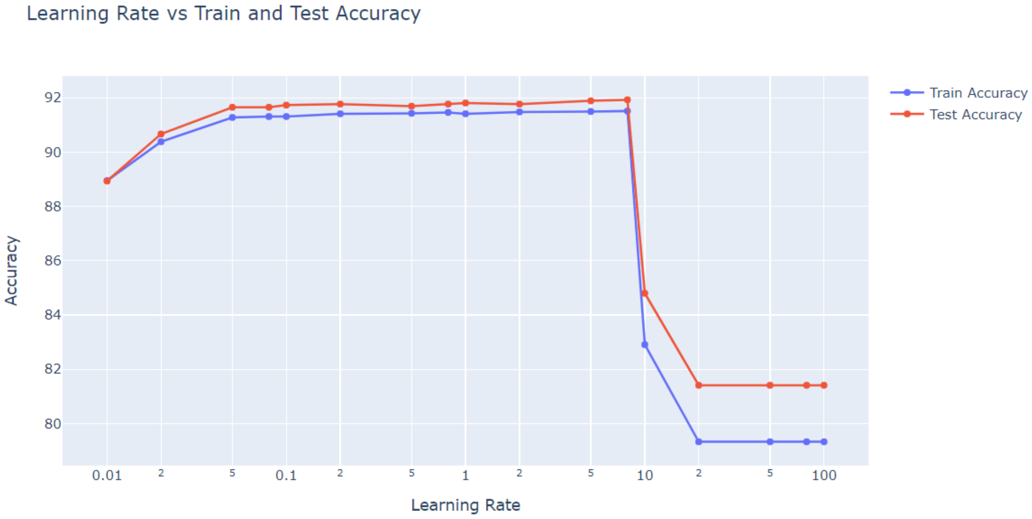
Threshold determines the minimum percentage change in cost function required at each update. If the change is lower than threshold, gradient descent comes out of loop and reports the cost function and thetas at that point (convergence).

D1. Experiment 1:

For threshold = 0.001



	learning_rate	converging_iteration	train_accuracy	test_accuracy
0	100.00	1	79.328383	81.410004
1	80.00	1	79.328383	81.410004
2	50.00	1	79.328383	81.410004
3	20.00	1	79.328383	81.410004
4	10.00	5	82.905839	84.797164
5	8.00	547	91.511981	91.925955
6	5.00	715	91.495106	91.886570
7	2.00	1226	91.478232	91.768413
8	1.00	1854	91.410732	91.807798
9	0.80	2118	91.461357	91.768413
10	0.50	2797	91.427607	91.689642
11	0.20	4727	91.410732	91.768413
12	0.10	6854	91.309484	91.729027
13	0.08	7677	91.309484	91.650256
14	0.05	9648	91.275734	91.650256
15	0.02	10000	90.381370	90.665616
16	0.01	10000	88.947013	88.932651



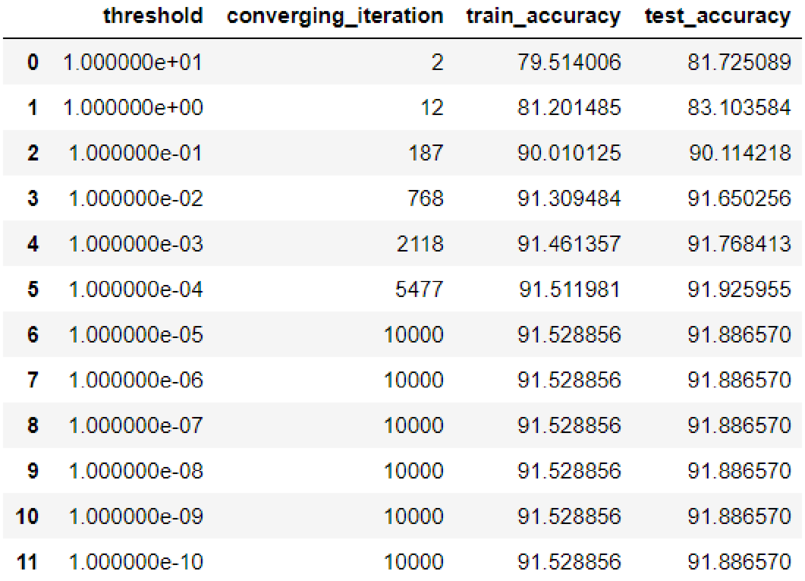
Learning Rate vs Train and Test Accuracy

Similar to Linear Regression Model, the gradient descent does not converge to minimum as it overshoots for higher learning rates. As a result, both train and test Accuracy are very low and converging iteration is less than 10. On the other hand, since the maximum iteration is capped at 10000, the gradient descent has not converged to global minimum and therefore the train and test Accuracy are comparatively lower than that at the optimum learning rate.

The chosen learning rate is 0.8.

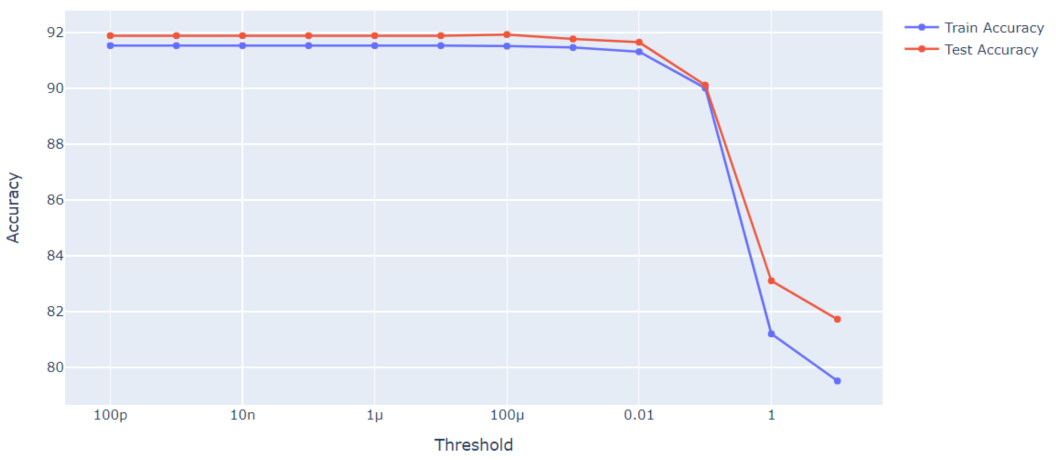
D2. Experiment 2:

For learning rate = 0.8



	threshold	converging_iteration	train_accuracy	test_accuracy
0	1.000000e+01	2	79.514006	81.725089
1	1.000000e+00	12	81.201485	83.103584
2	1.000000e-01	187	90.010125	90.114218
3	1.000000e-02	768	91.309484	91.650256
4	1.000000e-03	2118	91.461357	91.768413
5	1.000000e-04	5477	91.511981	91.925955
6	1.000000e-05	10000	91.528856	91.886570
7	1.000000e-06	10000	91.528856	91.886570
8	1.000000e-07	10000	91.528856	91.886570
9	1.000000e-08	10000	91.528856	91.886570
10	1.000000e-09	10000	91.528856	91.886570
11	1.000000e-10	10000	91.528856	91.886570

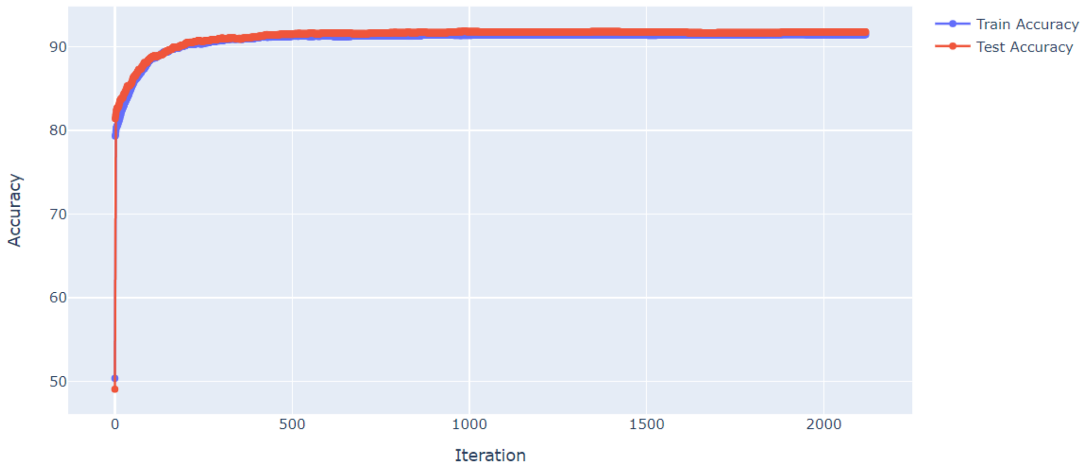
Threshold for Convergence vs Train and Test Accuracy (Learning Rate = 0.8)



Higher thresholds might lead to gradient descent not converging at global minimum. As the threshold decreases, there is not much change in the cost function. The train and test Accuracy do not change much by decreasing the threshold any further than optimum threshold.

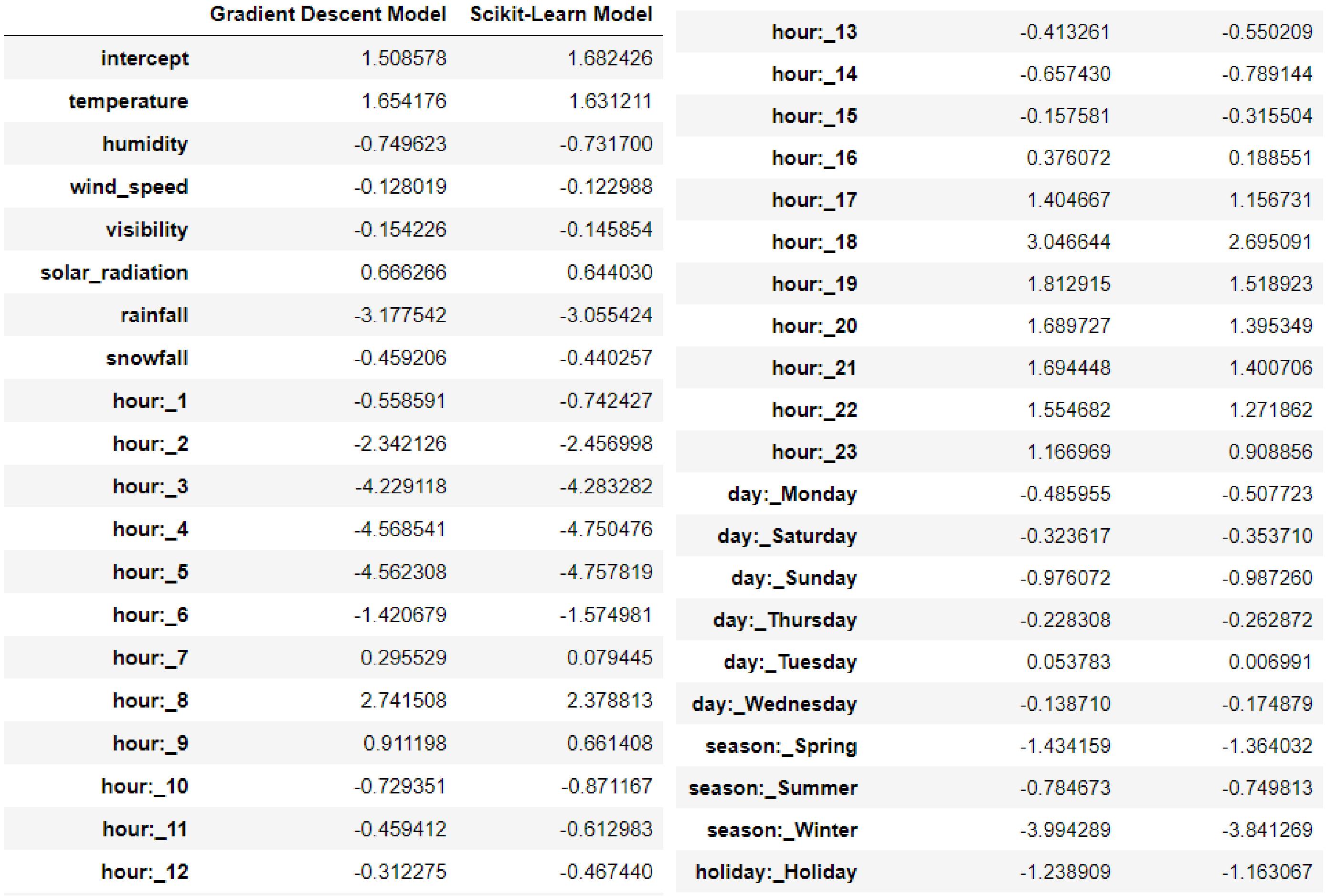
The optimum threshold is 0.001.

Train and Test Accuracy at various Iterations (Learning Rate = 0.8, Threshold = 0.001)



The above plot shows the train and test Accuracy at every iteration for chosen learning rate of 0.8 and threshold of 0.001. At 2118th iteration, the gradient descent comes out of the loop with minimum cost function and reports the thetas at that point.

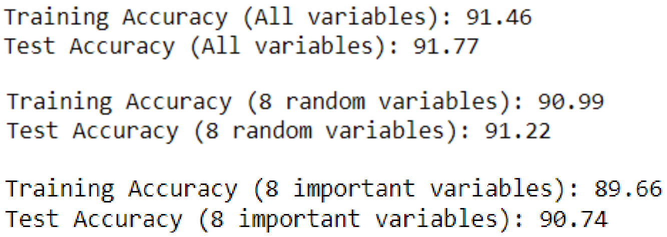
Thetas of our Logistic Regression Model are compared to that of Scikit-Learn Logistic Regression Model. Below is the table that summarizes the coefficients.



	Gradient Descent Model	Scikit-Learn Model
intercept	1.508578	1.682426
temperature	1.654176	1.631211
humidity	-0.749623	-0.731700
wind_speed	-0.128019	-0.122988
visibility	-0.154226	-0.145854
solar_radiation	0.666266	0.644030
rainfall	-3.177542	-3.055424
snowfall	-0.459206	-0.440257
hour:_1	-0.558591	-0.742427
hour:_2	-2.342126	-2.456998
hour:_3	-4.229118	-4.283282
hour:_4	-4.568541	-4.750476
hour:_5	-4.562308	-4.757819
hour:_6	-1.420679	-1.574981
hour:_7	0.295529	0.079445
hour:_8	2.741508	2.378813
hour:_9	0.911198	0.661408
hour:_10	-0.729351	-0.871167
hour:_11	-0.459412	-0.612983
hour:_12	-0.312275	-0.467440
hour:_13	-0.413261	-0.550209
hour:_14	-0.657430	-0.789144
hour:_15	-0.157581	-0.315504
hour:_16	0.376072	0.188551
hour:_17	1.404667	1.156731
hour:_18	3.046644	2.695091
hour:_19	1.812915	1.518923
hour:_20	1.689727	1.395349
hour:_21	1.694448	1.400706
hour:_22	1.554682	1.271862
hour:_23	1.166969	0.908856
day:_Monday	-0.485955	-0.507723
day:_Saturday	-0.323617	-0.353710
day:_Sunday	-0.976072	-0.987260
day:_Thursday	-0.228308	-0.262872
day:_Tuesday	0.053783	0.006991
day:_Wednesday	-0.138710	-0.174879
season:_Spring	-1.434159	-1.364032
season:_Summer	-0.784673	-0.749813
season:_Winter	-3.994289	-3.841269
holiday:_Holiday	-1.238909	-1.163067

D3. Experiment 3 and 4:

For learning rate = 0.8 and threshold = 0.001



```
Training Accuracy (All variables): 91.46
Test Accuracy (All variables): 91.77

Training Accuracy (8 random variables): 90.99
Test Accuracy (8 random variables): 91.22

Training Accuracy (8 important variables): 89.66
Test Accuracy (8 important variables): 90.74
```

The train and test Accuracy of the model containing all features is the highest of all, followed by the model containing 8 random variables. The model containing important features has the lowest train and test Accuracy. The variables selected using feature importance utilizes continuous target variable instead of binary target variable (required for Logistic Regression). Hence these variables not necessarily need to be important predictors for modelling such a target variable. The randomly selected features have higher explanatory power than the features selected using feature importance.

E. Questions:

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

The date and time categorical variables such as hour, season, day of the week and weather condition variables such as temperature, humidity windspeed and visibility are very important predictors. In fact, summer is the season when bike rentals are very high and, in a day, 5 p.m. to 10 p.m is when the business picks up. Commuters use bikes more often during in weekdays than weekends and holidays.

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

The ‘bike_count’ variable is highly skewed to the right. Running a linear regression model on such a target variable leads to heteroskedastic errors. By transforming the variable, we can take care of such errors. However, the interpretation of the coefficients change.

Another way to model this in a better way is to consider the dataset as time series model and run multi-variate time series forecasting models.