Jeffrey Chang
jlc190004

BUAN 6341 APPLIED MACHINE LEARNING ASSIGNMENT 1

The purpose of this assignment is to implement linear regression models on the dataset, "Seoul Bike-sharing Demand Data." Predicting "rented bike count" with linear regression and using a logistic regression model to classify the day whether "Holiday" or not.

Viewing the Seoul Bike-sharing Demand Dataset, which is 13 variables. (Exclude date)

	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
1	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
2	01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	Winter	No Holiday	Yes

Seoul Bike-sharing Demand Dataset does not have missing values after using *.isnull()*.

```
[1] dataset.isnull().sum() # check the Missing Data

Date                       0
Rented Bike Count          0
Hour                       0
Temperature(°C)            0
Humidity(%)                0
Wind speed (m/s)           0
Visibility (10m)           0
Dew point temperature(°C)  0
Solar Radiation (MJ/m2)    0
Rainfall(mm)               0
Snowfall (cm)              0
Seasons                    0
Holiday                    0
Functioning Day            0
dtype: int64
```

Linear Regression Model with multiple variables by Gradient Descent

There are three categorical variables (Seasons, Holiday, Functioning Day) for this dataset. Thus, the first step was creating the new dummy variables for three categorical variables and scaled all variables to avoid the larger feature(s) that will dominate.

	Autumn	Summer	Spring	Winter	No	Yes	Holiday	No Holiday	Hour	Temperature	Humidity	Wind speed	Visibility	Dew point temperature	Solar Radiation	Rainfall	Snowfall
0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.000000	0.220280	0.377551	0.297297	1.0	0.224913	0.0	0.0	0.0
1	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.043478	0.215035	0.387755	0.108108	1.0	0.224913	0.0	0.0	0.0
2	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.086957	0.206294	0.397959	0.135135	1.0	0.223183	0.0	0.0	0.0

Partition dataset randomly into Train (70%) and Test (30%) set. Notice: here I add one at 1st column for β_0. The shape of each train and test set:
[X_train, Y_train] : [(6312,18), (6132,)] and [X_test, Y_test] : [(2628,18), (2628,)]

There are 18 different independent variables and 1 dependent variable (Rented bike count).
My original design regression model equation:

$$Rented\_bike\_count = \beta_0+\beta_1*(Autumn)+\beta_2*(Summer)+\beta_3*(Spring)+\beta_4*(Winter)+\beta_5*(NoFunc)+\beta_6*(YesFunc)+\beta_7*(Holiday)+\beta_8*(No\_Holiday)+\beta_9*(Hour)+\beta_{10}(Temperature)+\beta_{11}*(Humidity)+\beta_{12}*(Wind\_speed)+\beta_{13}*(Visibility)+\beta_{14}*(Dew\_point\_temperature)+\beta_{15}*(Solar\_Radiation)+\beta_{16}*(Rainfall)+\beta_{17}*(Snowfall)$$

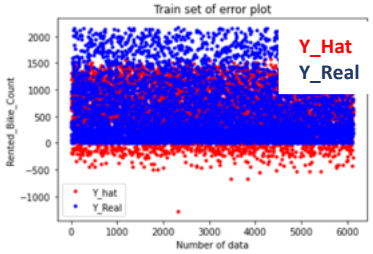
Jeffrey Chang
jlc190004

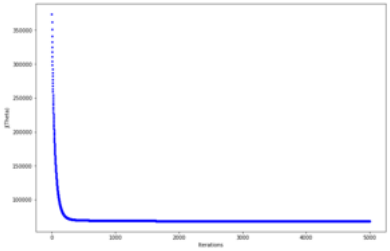
Experiment with all variables by different various parameters

Implement the gradient descent algorithm with batch update rule. Use the same cost function as in the class (sum of squared error). Trying different α and iteration times to find the best R^2.

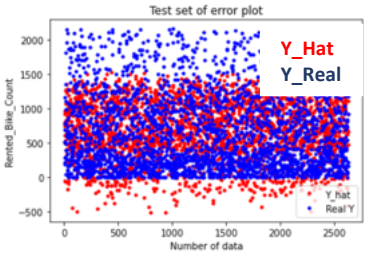
1st: All variables, α = 0.4, iteration = 5000. (Best values of the parameters)

Rented_bike_count = (-97.18) + (133.03)*(*Autumn*) + (10.11)*(*Summer*) − (11.22)*(*Spring*) − (229.1) *(*Winter*) − (484.94)*(*NoFunc*) + (387.76)*(*YesFunc*) − (104.88)*(*Holiday*) + (7.7)*(*No_Holiday*) + (566.02)*(*Hour*) + (670.74)*(*Temperature*) − (1037.91)*(*Humidity*) + (48.54)*(*Wind_speed*) + (11.77)*(*Visibility*) + (699.6)*(*Dew_point_temperature*) − (190.1)*(*Solar_Radiation*) − (1601.69) *(*Rainfall*) + (276.35)*(*Snowfall*)



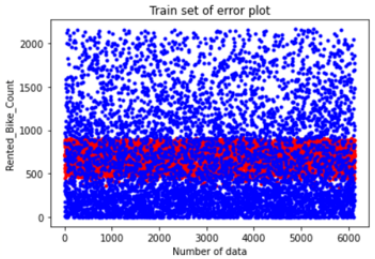


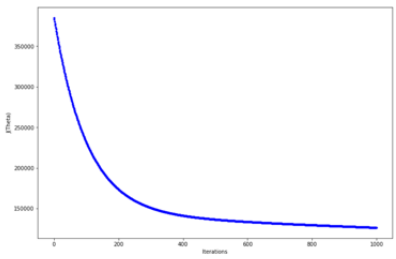
Final Cost = 68479
R^2 = 0.5852



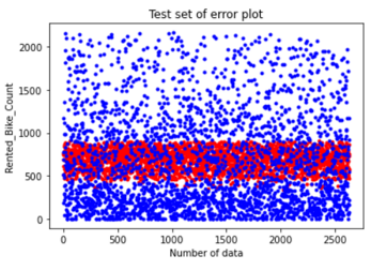
The 2nd try starting form small α and iteration.

2nd: All variables, α = 0.001, iteration = 1000.



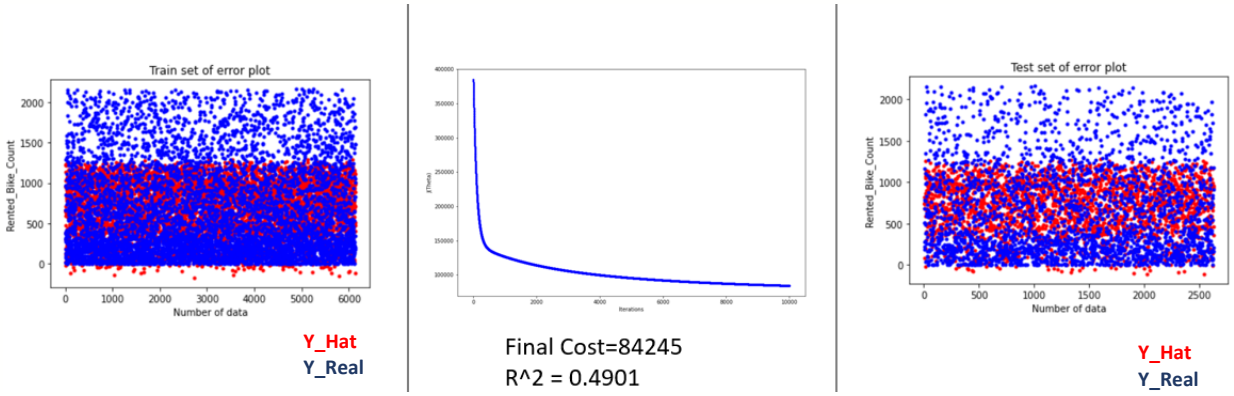


Final Cost=126538
R^2 = 0.2344



The 2nd try had a very small R^2 score and higher cost. Thus, doing more iteration might help to optimize the cost function and R^2 by closing the minima

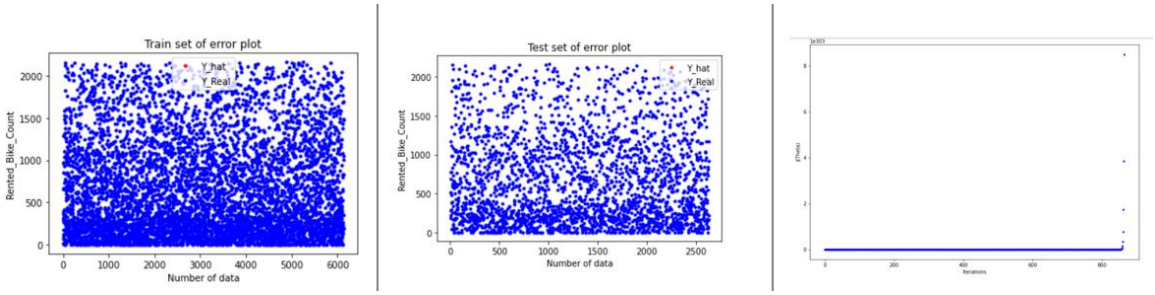
3rd: All variables, α = 0.001, iteration = 10000.



The 3rd try had a big improvement in R^2 score and final cost. Thus, increasing the learning rate to check whether it helps to optimize the cost function and R^2 by closing the minima.

4th: All variables, α = 0.5, iteration = 10000.

The learning rate α is too big, so the cost function value cannot guarantee convergence.



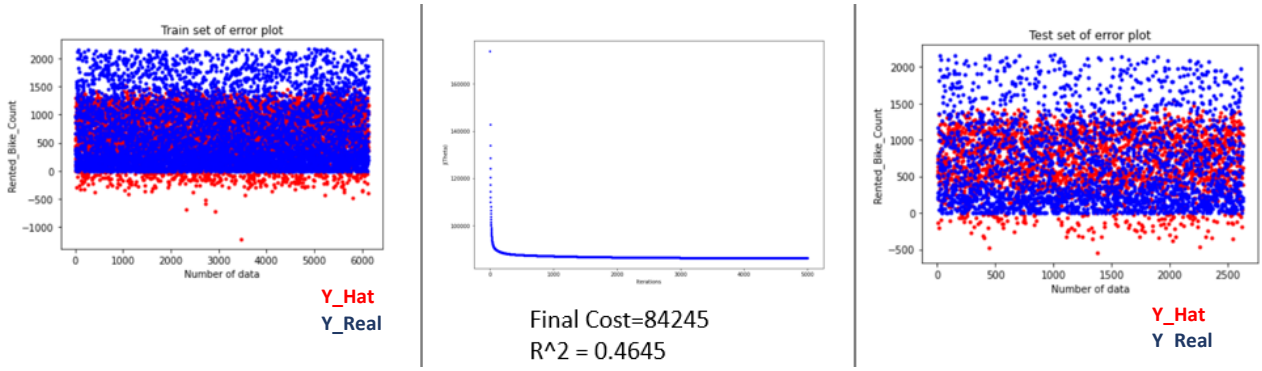
Conclusion of all variables by various parameters

- Increasing the iteration times will help to optimize the cost function and R^2 by closing to the minima. However, the iterations times over a specific number, the improvement is tiny. Thus, lower the iteration time will help decrease the program running time.
- If the learning rate α is too big, the cost function value cannot support convergence because of too large steps for gradient descent
- If the learning rate α is too small, the R^2 is small because the small step of gradient descent causes iteration numbers might not be enough.

Experiment with 8 random variables

The random 8 variables are Hour, Humidity (%), Wind speed (m/s), Visibility (10m), Rainfall(mm), Snowfall (cm), Seasons, Holiday. There are two categorical variables, Seasons and Holiday. Therefore, let's created the new dummy variable and scaled the new random dataset. Setting seed (30) for the same output.

Random 8 variables, α = 0.4, iteration = 5000.



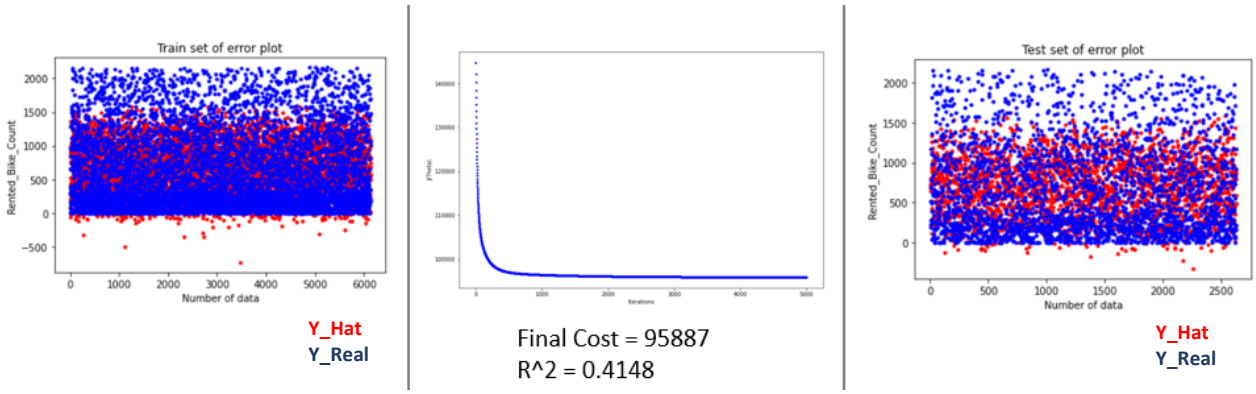
Experiment with 8 select variables

The reasons for choosing these 8 variables:

1. Removed 'Holiday' and 'Functioning Day.' They have small variations within because there are many 0 and seldom 1.

2. Removed 'Hours' and 'Seasons' is because they are continuing variables from 1 to 23 or solid variable (4 seasons). Thus, let's assumed their correlation with the dependent variable is small.

Select 8 variables, α = 0.4, iteration = 5000.



Conclusion of experiment with 8 random and select variables:

For both random and select experiments, using the same α = 0.4 and iteration = 5000 because it has the best values of the parameters with all variables. According to the results, if we reduced the variables, the R^2 will decrease and final J(θ) will increase. For this Seoul Bike-sharing Demand Dataset, we might produce a better prediction on "rank bike count" with all variables. Moreover, the select 8 variables model has an overfitting problem compare to the random 8 variables model. At the select 8 variables model, the error plot of the Train set in which the distribution of red dots seems to coincide with the blue dots. However, the random 8 variables model has better R^2 and J(θ) values.

Logistic Regression Model with multiple variables by Gradient Descent

In "Seoul Bike-sharing Demand Data," binary classification is using the variable of "Holiday." The setup of two categories is "No Holiday" = 0 and "Holiday" = 1.

	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
1	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
2	01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	Winter	No Holiday	Yes

Partition Seoul Bike-sharing Demand Dataset randomly into two datasets, which are the Train (70%) and Test (30%). The first experiment, design to use all variables. Thus, encoding two variables "Seasons" and "Functioning day" into the number, and scaling the dataset both make the output more precise. It is unnecessary for logistic regression to scale. Notice: Adding one at the 1st column for ω_0 needs. The shapes of each Train and Test sets are:

[X_train_L, Y_train_L] : [(6312,17), (6132,)] and [X_test_L, Y_test_L] : [(2628,17), (2628,)]

Original Logistic regression model equation design: *(Default p > 0.5 = 1, p < 0.5 = 0)*

$Holiday = \omega_0 + \omega_1 * (Autumn) + \omega_2 * (Summer) + \omega_3 * (Spring) + \omega_4 * (Winter) + \omega_5 * (NoFunc) + \omega_6 * (YesFunc) + \omega_7 * (Rented\_bike\_count) + \omega_8 * (Hour) + \omega_9 (Temperature) + \omega_{10} * (Humidity) + \omega_{11} * (Wind\_speed) + \omega_{12} * (Visibility) + \omega_{13} * (Dew\_point\_temperature) + \omega_{14} * (Solar\_Radiation) + \omega_{15} * (Rainfall) + \omega_{16} * (Snowfall)$

Experiment with various parameters by all variables

The experiment implemented the logistic regression model with all variables by gradient descent, used the same cost function in the class (sum of squared error), and tried out different α and iteration combinations to find the best Accuracy Score as possible.

1st: All variables, α = 0.001, iteration = 1000, default (p>0.5 = 1, p<0.5 = 0).

Using the new regression coefficient into the Test set to calculate the confusion matrix.

Accuracy score: 0.972.

```
[[2538   30    0    0    0    0    0    0    0]
 [   0   17    0    0    0    0    0    0    0]
 [   0   24    0    0    0    0    0    0    0]
 [   0   11    0    0    0    0    0    0    0]
 [   0    4    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]]

0.9722222222222222
```

Next try, trying to increase iteration times and α to optimize logistic regression cost function and find a higher accuracy score.

2nd: All variables, α = 0.005, iteration = 10000, default (p>0.5 = 1, p<0.5 = 0).

```
[[2505   63    0    0    0    0    0    0    0]
 [   0   17    0    0    0    0    0    0    0]
 [   0   24    0    0    0    0    0    0    0]
 [   0   11    0    0    0    0    0    0    0]
 [   0    4    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]]
0.9596651445966514
```

Increased both iteration times and α did not optimize logistic regression cost function and accuracy score, so next try will use small α to find accuracy score.

3rd: All variables, α = 0.001, iteration = 10000, default (p>0.5 = 1, p<0.5 = 0).

```
[[2514   54    0    0    0    0    0    0    0]
 [   0   17    0    0    0    0    0    0    0]
 [   0   24    0    0    0    0    0    0    0]
 [   0   11    0    0    0    0    0    0    0]
 [   0    4    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]]
0.963089802130898
```

In 3rd try, the accuracy is better than 2nd try but lower than 1st try. The followed try is to change decision boundary, and 4th try is setting p > 0.3 = 1, p < 0.3 = 0.

4th: All variables, α = 0.001, iteration = 1000, default (p>0.3 = 1, p<0.3 = 0).

```
[[2526   42    0    0    0    0    0    0    0]
 [   0   17    0    0    0    0    0    0    0]
 [   0   24    0    0    0    0    0    0    0]
 [   0   11    0    0    0    0    0    0    0]
 [   0    4    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]]
0.9676560121765602
```

In 4th try, the accuracy is better than 3rd try but lower than 1st try. The followed try increased iteration times trying to optimize logistic regression cost function and accuracy score.

5th: All variables, α = 0.001, iteration = 10000, default (p>0.3 = 1, p<0.3 = 0).

```
[[2510   58    0    0    0    0    0    0    0]
 [   0   17    0    0    0    0    0    0    0]
 [   0   24    0    0    0    0    0    0    0]
 [   0   11    0    0    0    0    0    0    0]
 [   0    4    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]]
0.9615677321156774
```

6th: All variables, α = 0.005, iteration = 10000, default (p>0.3 = 1, p<0.3 = 0).

```
[[2504   64    0    0    0    0    0    0    0]
 [   0   17    0    0    0    0    0    0    0]
 [   0   24    0    0    0    0    0    0    0]
 [   0   11    0    0    0    0    0    0    0]
 [   0    4    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]]
0.9592846270928462
```

The 6th try's accuracy score is the worst.

7th: All variables, α = 0.001, iteration = 1000, p>0.8 = 1, p<0.3 = 0 (Best)

```
[[2568    0    0    0    0    0    0    0    0]
 [   4   13    0    0    0    0    0    0    0]
 [   0   24    0    0    0    0    0    0    0]
 [   0   11    0    0    0    0    0    0    0]
 [   0    4    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]]
0.9821156773211568
```

After trying different combination, when α = 0.001, iteration = 1000, p>0.8 = 1, p<0.3 = 0, the best accuracy score = 0.98212. The best logistic regression for this experiment is:

Holiday (p > 0.8 = 1, p < 0.3 = 0) = (-0.375) + (-0.01) *(*Autumn*) + (-0.034)*(*Summer*) + (-0.025)*(*Spring*) + (0.068)*(*Winter*) + (-0.01)*(*NoFunc*) + (0.01)*(*YesFunc*) + (-0.036)*(*Rented_bike_count*) + (-0.003)*(*Hour*) + (-0.06)*(*Temperature*) + (0.039)*(*Humidity*) + (-0.001) *(*Wind_speed*) + (-0.04)*(*Visibility*) + (-0.038)*(*Dew_point_temperature*) + (-0.016)*(*Solar_Radiation*) + (-0.04)*(*Rainfall*) + (0.382)*(*Snowfall*)

Experiment with 8 random variables

The random 8 variables are: Rented Bike Count, Temperature(°C), Humidity (%), Wind speed (m/s), Visibility (10m), Solar Radiation (MJ/m2), Rainfall(mm), Seasons. One categorical variable, "Seasons," so created the new dummy variables and scaled the 8 random variables. The accuracy score of 8 random variables try is lower than the best accuracy score of all variables try. Using the same parameters when α = 0.001, iteration = 1000, p>0.8 = 1, p<0.3 = 0.

```
[[2568    0    0    0    0    0    0    0    0]
 [  17    0    0    0    0    0    0    0    0]
 [  24    0    0    0    0    0    0    0    0]
 [  11    0    0    0    0    0    0    0    0]
 [   4    0    0    0    0    0    0    0    0]
 [   1    0    0    0    0    0    0    0    0]
 [   1    0    0    0    0    0    0    0    0]
 [   1    0    0    0    0    0    0    0    0]
 [   1    0    0    0    0    0    0    0    0]]
0.9771689497716894
```

accuracy score = 0.97717

Experiment with 8 select variables

The select 8 variables are: Rented Bike Count, Temperature(°C), Humidity (%), Visibility (10m), Dew point temperature(°C), Solar Radiation (MJ/m2), Rainfall(mm), Snowfall (cm). The select variables eliminated the lower value of the coefficient of the best logistic regression of all variables. For example, Wind Speed and Hour. Moreover, remove the category variables The accuracy score of 8 select variables try is higher than the 8 random variables model but still lower than the best accuracy score of all variables try. Using the same parameters when α = 0.001, iteration = 1000, p > 0.8 = 1, p < 0.3 = 0.

```
[[2568    0    0    0    0    0    0    0    0]
 [   9    8    0    0    0    0    0    0    0]
 [   0   24    0    0    0    0    0    0    0]
 [   0   11    0    0    0    0    0    0    0]
 [   0    4    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0]]
0.9802130898021308
```

Conclusion of experiment with 8 random and select variables:

Used the same parameters (α = 0.001, iteration = 1000, p > 0.8 = 1, p < 0.3 = 0) for both experiences. According to the result above, if we reduce the variables, the accuracy score will decrease. For this Seoul Bike-sharing Demand Dataset, we might produce a better classification

on "Holiday" with all variables by logistic regression model with multiple variables by gradient descent. Moreover, there are some reasons for the select model has a better score than a random model. First, tuning the parameters by removing the lower value. Second, there might be a lower correlation between category variables.

Summary

Why both all variables models have better output?

First, each variable in Seoul Bike-sharing Demand Data" might have a correlation coefficient ($\neq 0$). Therefore, when we build the regression models, removing any one of the variables will influence the output. Second, the Gradient Descent algorithm might lead to local minima instead of not global minima. Third, the miscalculation or misoperation caused by human error.

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

1. Collect more information (data or variables) to improve the dataset.
2. Computing the correlation coefficient and remove or input the proper variables.
3. Using multiple algorithms to build the regression models.

The conclusion from previous experiments:

- Increasing the iteration times in a reasonable range will help to optimize the cost function and coefficient of determination(R^2) by closing to the minima.
- It is essential to set the proper learning rate α. If α is too big, the cost function value cannot support the convergence, and it will skip the true local minima. If α is too small, it takes too many steps and causes a running time problem by increasing iteration times.
- For the "Seoul Bike-sharing Demand Data" experiment, both Linear Regression Model and Logistic Regression Model, all variables models produced a better prediction with a higher coefficient of determination(R^2) or better classification with a superior accuracy score.
- The select 8 variables model might have an overfitting problem compare to the random 8 variables model. It will cause a lower prediction problem.
- If we can remove the lower correlation variables between independent and dependent variables, the output will better than choosing random variables.