Statistical Data Analysis

Author: Varsha Gopalakrishnan

In the exploratory data analysis, we identified several features that were correlated with the Black Carbon (BC) and Nitrogen dioxide (NO2) concentrations. We observed strong positive or negative correlation for some features, and weak correlations for this rest. In this section, we perform statistical tests to identify the degree of correlation between the predictor and independent variable, and determine if there is multicollinearity between the predictor variables. We also identify the model that best fits the data for BC and NO2.

Identify features that have a high correlation, with a correlation coefficient above 0.9 or below -0.9, drop those features and fit a simple Linear Regression Model using OLS

Methodology:

First, we identify features that have a high correlation by setting a threshold of 0.9, and comparing each feature against all other features to check if the correlation matrix resulted in a value of >= 0.9 or <= -0.9, indicating a strong positive or negative correlation between features.

This test for multicollinearity will identify features that are highly correlated with each other. However, one of the assumptions of a linear regression model is that independent variables should be uncorrelated. Preserving highly correlated variables may result in coefficients from the regression not to be estimated precisely, and the standard errors are likely to be high.

Results:

BC Dataset:
By running the above test, we identified that for the BC dataset, the following features were not highly correlated
1. high-AsphaltPlant-10510811_dist
2. high-AsphaltPlant-808611_dist
3. high-AutoRepair-15714511_dist
4. Precip
5. Radiation
6. Maxtemp
7. Mintemp
8. Pressure
9. Number_intersections

Fitting a linear model using statsmodel with the above features resulted in the following summary statistics:

OLS Regression Results

Dep. Variable:	BC_Value	R-squared (uncentered):	0.682
Model:	OLS	Adj. R-squared (uncentered):	0.682
Method:	Least Squares	F-statistic:	5121.
Date:	Sun, 31 May 2020	Prob (F-statistic):	0.00
Time:	11:07:14	Log-Likelihood:	-16976.
No. Observations:	21488	AIC:	3.397e+04
Df Residuals:	21479	BIC:	3.404e+04
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
high-AsphaltPlant-10510811_dist	-0.0399	0.005	-8.699	0.000	-0.049	-0.031
high-AsphaltPlant-808611_dist	0.0155	0.004	4.270	0.000	0.008	0.023
high-AutoRepair-15714511_dist	0.0836	0.004	23.350	0.000	0.077	0.091
Precip	-2.7197	0.077	-35.422	0.000	-2.870	-2.569
Radiation	0.0899	0.002	36.860	0.000	0.085	0.095
Maxtemp	-1.5640	0.039	-40.448	0.000	-1.640	-1.488
Mintemp	0.2596	0.134	1.941	0.052	-0.003	0.522
Pressure	0.0039	0.002	2.092	0.036	0.000	0.008
number_intersections	-0.0121	0.001	-16.788	0.000	-0.013	-0.011

| Omnibus:	5567.049	Durbin-Watson:	0.955
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18211.398
Skew:	1.309	Prob(JB):	0.00
Kurtosis:	6.672	Cond. No.	3.97e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.97e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Actual vs. Predicted BC Concentration
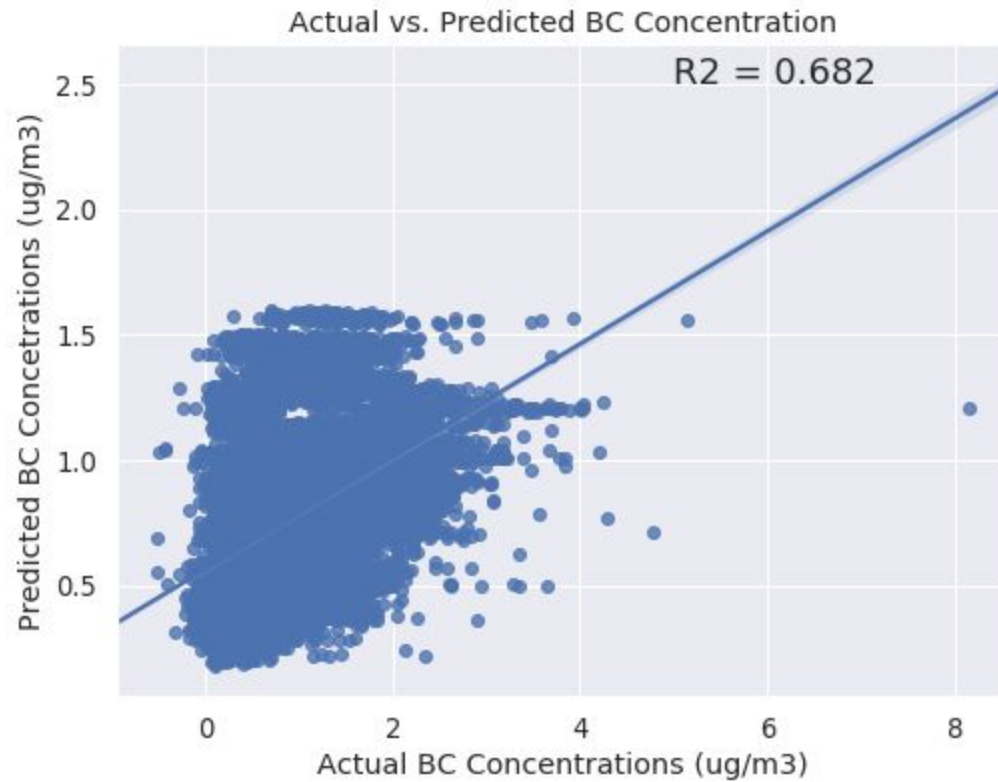
A plot of the actual vs predicted BC concentration shows that a **linear regression model does a decent job** of predicting concentrations for BC, especially when the concentrations are below 1.5 ug/m3, but does not do a good job at higher concentrations.

Correlation heatmap of the above selected features is shown here:

Correlation Matrix - Black Carbon - Simple OLS
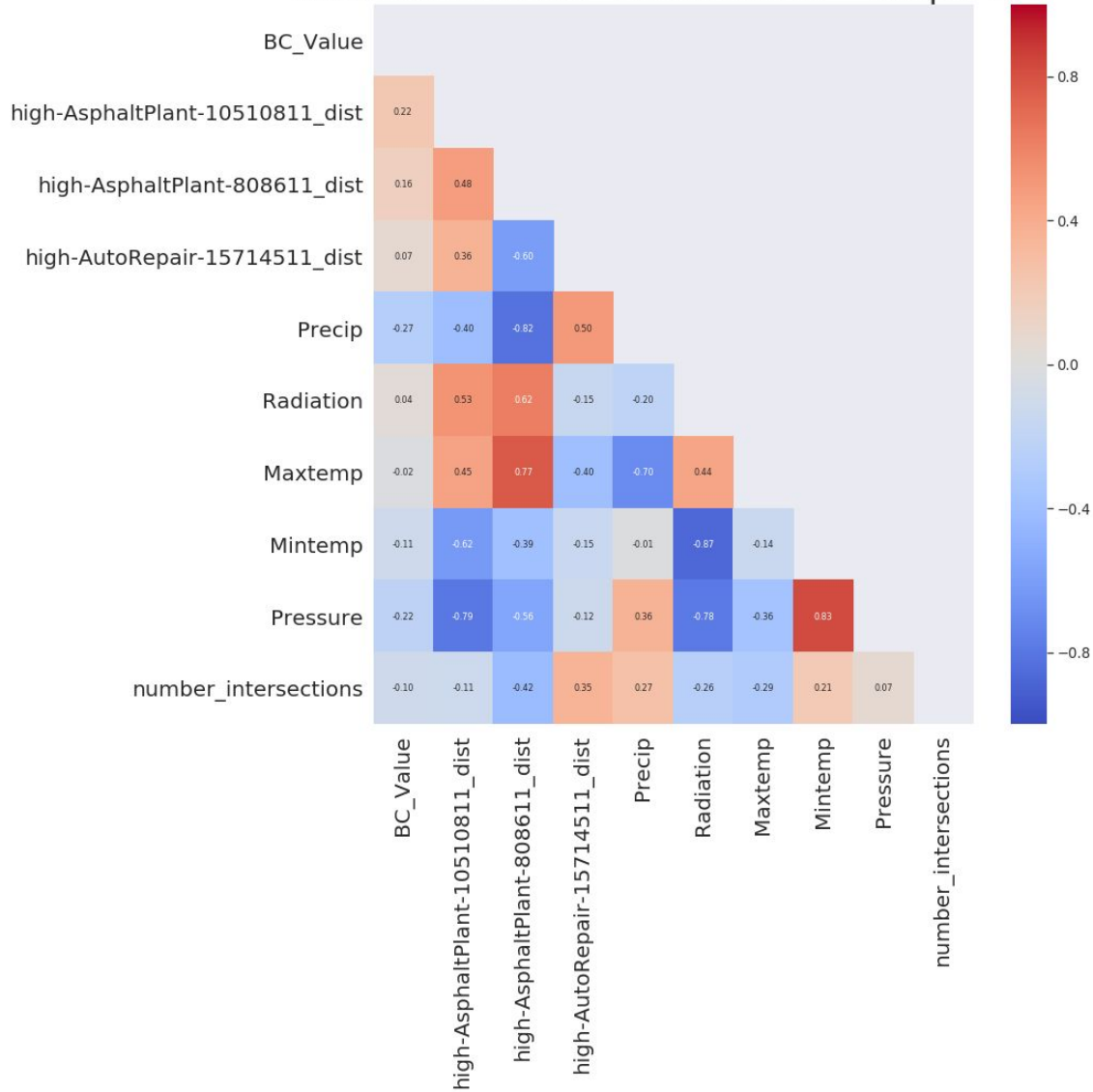
NO2 Dataset:

Applying the above test for the NO2 dataset, we identify the following features were not highly correlated

1. high-AsphaltPlant-10510811_dist
2. high-AsphaltPlant-808611_dist
3. high-FoodPlant-340611_dist
4. Precip
5. Radiation
6. Maxtemp
7. Mintemp
8. Pressure
9. number_intersections

Fitting a linear model using statsmodel with the above features resulted in the following summary statistics:

OLS Regression Results

Dep. Variable:	NO2_Value	R-squared (uncentered):	0.818
Model:	OLS	Adj. R-squared (uncentered):	0.818
Method:	Least Squares	F-statistic:	1.075e+04
Date:	Sun, 31 May 2020	Prob (F-statistic):	0.00
Time:	11:07:23	Log-Likelihood:	-76254.
No. Observations:	21488	AIC:	1.525e+05
Df Residuals:	21479	BIC:	1.526e+05
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
high-AsphaltPlant-10510811_dist	0.1445	0.072	2.017	0.044	0.004	0.285
high-AsphaltPlant-808611_dist	-0.1970	0.057	-3.447	0.001	-0.309	-0.085
high-FoodPlant-340611_dist	0.9573	0.057	16.924	0.000	0.846	1.068
Precip	-43.8396	1.213	-36.156	0.000	-46.216	-41.463
Radiation	1.6794	0.038	43.681	0.000	1.604	1.755
Maxtemp	-29.0032	0.610	-47.551	0.000	-30.199	-27.808
Mintemp	8.8408	2.111	4.189	0.000	4.704	12.978
Pressure	0.0129	0.029	0.442	0.658	-0.044	0.070
number_intersections	-0.0728	0.011	-6.411	0.000	-0.095	-0.051

| Omnibus:	1881.925	Durbin-Watson:	0.815
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2437.379
Skew:	0.772	Prob(JB):	0.00
Kurtosis:	3.579	Cond. No.	3.97e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.97e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Actual vs. Predicted NO2 Concentration
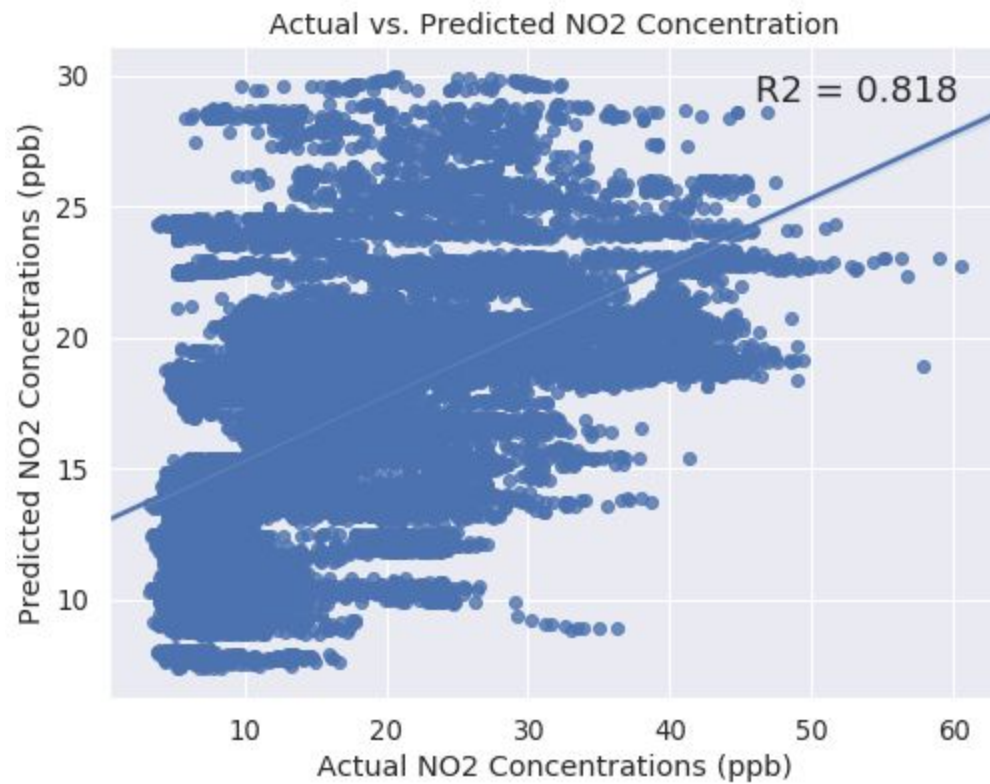
Once again, the above plot shows that the simple linear model does a decent job of predicting concentrations below 30 ppb, but does not perform well at high concentrations.

A correlation heatmap of selected features is shown here:

Correlation Matrix - Nitrogen Dioxide - Simple OLS
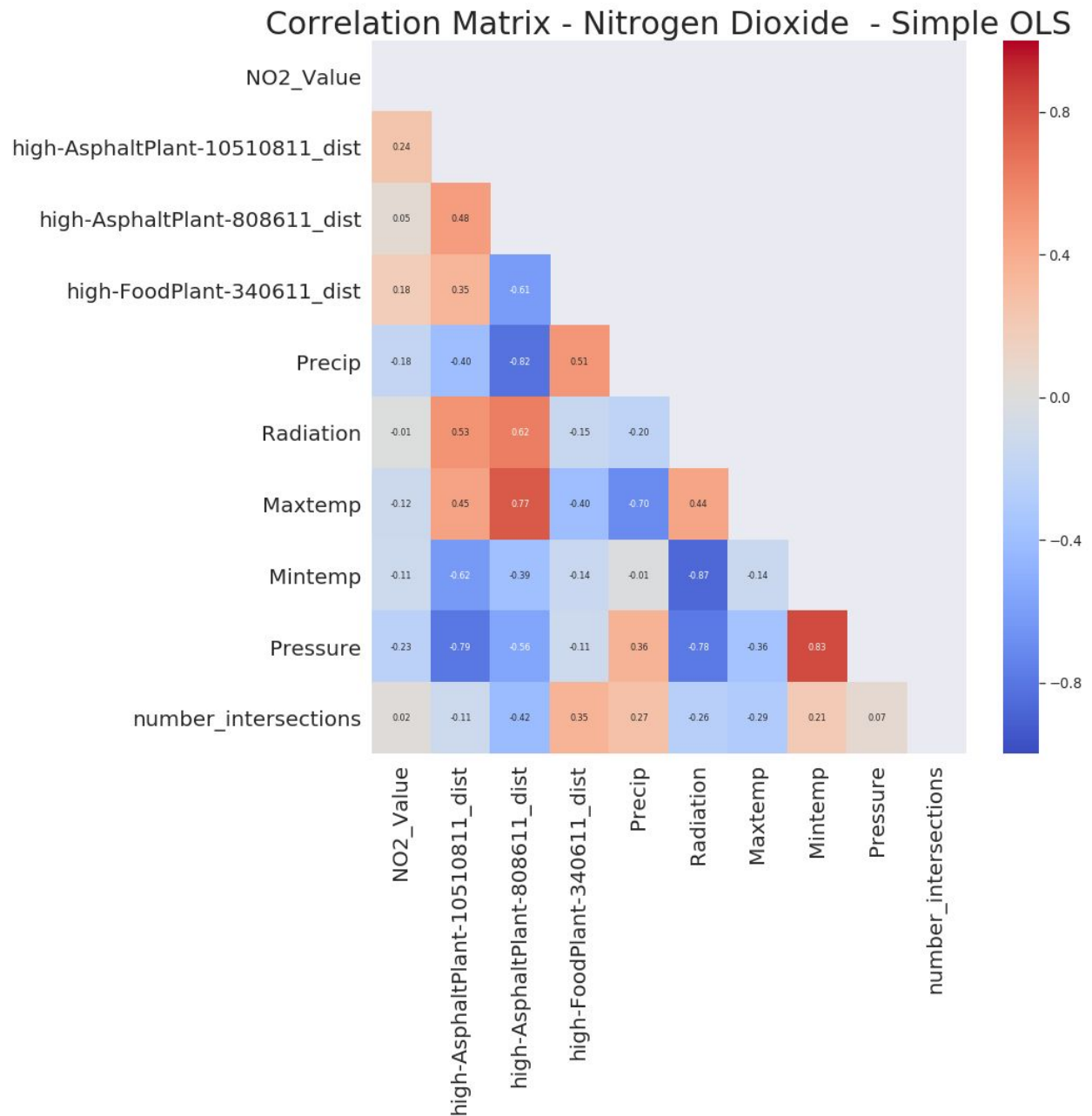
Summary:
The OLS for both the BC and NO2 datasets results in a model with a decent R2 value indicating that linear regression model may result in the good fit, at least with just the features that are not highly correlated. However, these features have a very high Variance Inflation Scores resulting in multicollinearity. We can try better algorithms that may result in a better fit.

Step Forward Approach with VIF Scores and R2 estimation

The approach here is to start with the feature that results in the best R2 value, and then loop through all the features by adding features sequentially. Each time a new feature is added, we calculate the R2 for the new model, and determine if the new R2 is higher than the previously estimated R2 value. If the R2 is higher, then we calculate the VIF scores with addition of each feature. If the VIF score of any of the features increases above threshold of 10, then we drop the newly added feature. This process continues until we determine the best set of features that results in highest R2 value, and VIF scores < 10.

A VIF is the quotient of the variance in a model with multiple terms by the variance of a model with one term alone. It quantifies the severity of multicollinearity in an ordinary least squares regression analysis.

Methodology:
Prior to applying the VIF methodology, the column names for the predictor variables were transformed to the format Q('column name') since some column names contain numbers. The data was then split into test, train dataset to get the best generalizable features.

The first step is to loop through all the features, and fit a simple OLS model for each feature individually. Determine the feature that results in the largest R2 value. Next, we loop through the features again, and recalculate the R2 value for the addition of each new feature. If the R2 value for the model with newly added feature is larger than the previously estimated R2, we calculate the VIF scores and determine if all the VIF scores are below 10. If *any* score exceeds 10, then we drop the newly added feature.

Finally, we recalculate VIF score for the remaining features, and fit an OLS on the remaining features.

Results:

BC dataset:
For the BC dataset, only 'Radiation' as a feature was selected and the OLS fit resulted in an R2 value of 0.593 for the training data and 0.584 for the test data.

Results of the OLS model fit with the selected features are shown below.

Training data:

```
(     VIF Factor          features
0   46676.348204        Intercept
1       1.000000   Q('Radiation'), <class 'statsmodels.iolib.summary.Summary'>
"""
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              BC_Value   R-squared (uncentered):              0.593
Model:                           OLS   Adj. R-squared (uncentered):         0.593
Method:                Least Squares   F-statistic:                      2.190e+04
Date:               Sun, 14 Jun 2020   Prob (F-statistic):                   0.00
Time:                       23:17:33   Log-Likelihood:                    -13655.
No. Observations:              15041   AIC:                              2.731e+04
Df Residuals:                  15040   BIC:                              2.732e+04
Df Model:                          1
Covariance Type:           nonrobust
==============================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Q('Radiation')    0.0021   1.42e-05    147.979      0.000       0.002       0.002
==============================================================================
Omnibus:                    3927.252   Durbin-Watson:                       2.017
Prob(Omnibus):                 0.000   Jarque-Bera (JB):                 8782.050
Skew:                          1.502   Prob(JB):                             0.00
Kurtosis:                      5.234   Cond. No.                             1.00
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
""")
```

Test data:

```
(     VIF Factor          features
0   46894.386704        Intercept
1       1.000000   Q('Radiation'), <class 'statsmodels.iolib.summary.Summary'>
"""
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              BC_Value   R-squared (uncentered):              0.584
Model:                           OLS   Adj. R-squared (uncentered):         0.584
Method:                Least Squares   F-statistic:                         9059.
Date:               Sun, 14 Jun 2020   Prob (F-statistic):                   0.00
Time:                       23:18:10   Log-Likelihood:                    -6045.7
No. Observations:               6447   AIC:                              1.209e+04
Df Residuals:                   6446   BIC:                              1.210e+04
Df Model:                          1
Covariance Type:           nonrobust
==============================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Q('Radiation')    0.0021   2.24e-05     95.176      0.000       0.002       0.002
==============================================================================
Omnibus:                    2496.187   Durbin-Watson:                       1.995
Prob(Omnibus):                 0.000   Jarque-Bera (JB):                12776.889
Skew:                          1.804   Prob(JB):                             0.00
Kurtosis:                      8.877   Cond. No.                             1.00
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
""")
```

NO2 dataset:

For the NO2 dataset, only 'Radiation' as a feature was selected and the OLS fit resulted in an R^2 value of 0.759 for the training data and 0.757 for the test data.

Results of the OLS model fit with the selected features are shown below.

Training data:

```
(    VIF Factor        features
0  46676.348204       Intercept
1      1.000000  Q('Radiation'), <class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                NO2_Value   R-squared (uncentered):          0.759
Model:                              OLS   Adj. R-squared (uncentered):     0.759
Method:                   Least Squares   F-statistic:                 4.748e+04
Date:                  Sun, 14 Jun 2020   Prob (F-statistic):               0.00
Time:                          23:25:17   Log-Likelihood:                -55437.
No. Observations:                 15041   AIC:                         1.109e+05
Df Residuals:                     15040   BIC:                         1.109e+05
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Q('Radiation')   0.0499      0.000    217.896      0.000       0.049       0.050
==============================================================================
Omnibus:                     1777.290   Durbin-Watson:                   2.016
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2467.291
Skew:                           0.969   Prob(JB):                         0.00
Kurtosis:                       3.425   Cond. No.                         1.00
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
""")
```

Test data:

```
(    VIF Factor        features
0  46894.386704       Intercept
1      1.000000  Q('Radiation'), <class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                NO2_Value   R-squared (uncentered):          0.757
Model:                              OLS   Adj. R-squared (uncentered):     0.757
Method:                   Least Squares   F-statistic:                 2.004e+04
Date:                  Sun, 14 Jun 2020   Prob (F-statistic):               0.00
Time:                          23:26:25   Log-Likelihood:                -23872.
No. Observations:                  6447   AIC:                         4.775e+04
Df Residuals:                      6446   BIC:                         4.775e+04
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Q('Radiation')   0.0504      0.000    141.555      0.000       0.050       0.051
==============================================================================
Omnibus:                      732.467   Durbin-Watson:                   1.995
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1003.036
Skew:                           0.950   Prob(JB):                    1.56e-218
Kurtosis:                       3.348   Cond. No.                         1.00
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
""")
```

Summary:

The stepforward approach with VIF and R2 estimation resulted in the selection of only one single feature, which is Radiation. This feature seems to result in the highest R2 with concentration, and addition of other features to the model does not improve performance.

Lasso Regularization with GridSearch

Methodology:
In this method, we do feature selection using Lasso Regularization. The lasso regularization will penalize a feature's coefficient by making it 0, if a feature is irrelevant. In this case, all features with coefficient = 0 can be removed from the model. In this approach, we first split the data into test and train set, perform a 5-fold Lasso cross validation on the training data. We then determine the optimal value for alpha that gives the best fit for the training data, use that value of alpha to refit the training data and predict on the test set. We compute the mean squared error on the test data, and the model score. Once the features are determined by Lasso, we again apply the step forward feature selection approach with R2 and VIF estimation to select features that are not highly collinear.

Results:
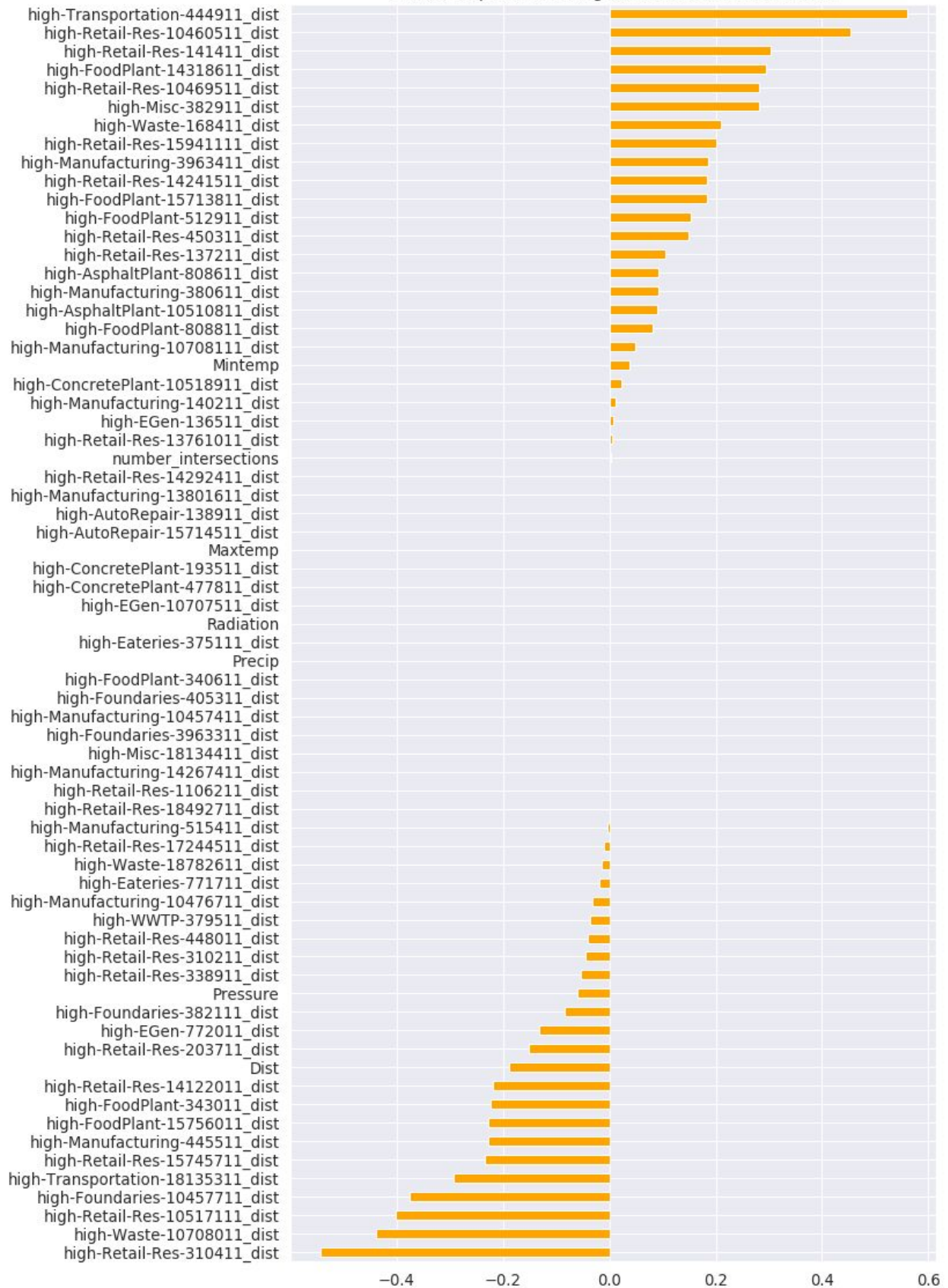
BC dataset:
For the BC dataset,

```
Best Alpha using LassoCV =  0.00015274253372807208
MSE on test data =  0.07799639454945655
Best score using LassoCV:  0.5345421171448599
```

The features that were selected are as follows. In total, Lasso picked 49 features out of the 68 features, as shown below:

Feature importance using Lasso Model for BC Dataset

Results of the OLS model fit for BC:

OLS Regression Results

Dep. Variable:	BC_Value	R-squared (uncentered):	0.827	
Model:	OLS	Adj. R-squared (uncentered):	0.827	
Method:	Least Squares	F-statistic:	2099.	
Date:	Sun, 31 May 2020	Prob (F-statistic):	0.00	
Time:	11:13:46	Log-Likelihood:	-10408.	
No. Observations:	21488	AIC:	2.091e+04	
Df Residuals:	21439	BIC:	2.131e+04	
Df Model:	49			
Covariance Type:	nonrobust			

	coef	std err	t	P>|t|	[0.025	0.975]
high-AsphaltPlant-10510811_dist	4.0567	0.436	9.295	0.000	3.201	4.912
high-AsphaltPlant-808611_dist	-41.9524	3.269	-12.835	0.000	-48.359	-35.546
high-ConcretePlant-10518911_dist	-1.0495	0.095	-11.074	0.000	-1.235	-0.864
high-EGen-136511_dist	4.4218	0.398	11.120	0.000	3.642	5.201
high-EGen-772011_dist	-0.9109	0.106	-8.580	0.000	-1.119	-0.703
high-Eateries-771711_dist	4.5713	0.659	6.935	0.000	3.279	5.863
high-FoodPlant-14318611_dist	15.2293	1.193	12.762	0.000	12.890	17.568
high-FoodPlant-15713811_dist	-0.3450	0.094	-3.684	0.000	-0.529	-0.161
high-FoodPlant-15756011_dist	-2.8975	0.524	-5.528	0.000	-3.925	-1.870
high-FoodPlant-343011_dist	-1.6590	0.178	-9.327	0.000	-2.008	-1.310
high-FoodPlant-512911_dist	-4.0435	0.833	-4.851	0.000	-5.677	-2.410
high-FoodPlant-808811_dist	0.5232	0.082	6.355	0.000	0.362	0.685
high-Foundaries-10457711_dist	-4.4442	1.363	-3.260	0.001	-7.116	-1.772
high-Foundaries-382111_dist	0.2678	0.113	2.372	0.018	0.047	0.489
high-Manufacturing-10476711_dist	3.1388	0.230	13.655	0.000	2.688	3.589
high-Manufacturing-10708111_dist	0.4735	0.197	2.408	0.016	0.088	0.859
high-Manufacturing-140211_dist	1.2277	0.833	1.474	0.140	-0.405	2.860

high-Manufacturing-380611_dist	2.2972	0.328	7.010	0.000	1.655	2.939
high-Manufacturing-3963411_dist	10.2872	1.892	5.437	0.000	6.579	13.996
high-Manufacturing-445511_dist	21.6157	4.852	4.455	0.000	12.106	31.125
high-Manufacturing-515411_dist	-1.4269	0.419	-3.407	0.001	-2.248	-0.606
high-Misc-382911_dist	0.2670	0.038	7.094	0.000	0.193	0.341
high-Retail-Res-10460511_dist	1.4084	0.284	4.966	0.000	0.852	1.964
high-Retail-Res-10469511_dist	1.7054	0.103	16.523	0.000	1.503	1.908
high-Retail-Res-10517111_dist	35.1287	3.896	9.016	0.000	27.492	42.766
high-Retail-Res-137211_dist	0.8309	0.080	10.365	0.000	0.674	0.988
high-Retail-Res-13761011_dist	0.5070	0.107	4.750	0.000	0.298	0.716
high-Retail-Res-14122011_dist	-0.5406	0.055	-9.804	0.000	-0.649	-0.432
high-Retail-Res-141411_dist	0.3273	0.040	8.281	0.000	0.250	0.405
high-Retail-Res-14241511_dist	0.5724	0.106	5.385	0.000	0.364	0.781
high-Retail-Res-15745711_dist	-0.0809	0.107	-0.754	0.451	-0.291	0.129
high-Retail-Res-15941111_dist	0.1098	0.099	1.111	0.267	-0.084	0.304
high-Retail-Res-17244511_dist	-11.4232	3.319	-3.442	0.001	-17.928	-4.919
high-Retail-Res-203711_dist	34.4138	2.578	13.351	0.000	29.361	39.466
high-Retail-Res-310211_dist	0.3069	0.060	5.112	0.000	0.189	0.425
high-Retail-Res-310411_dist	-27.4632	1.394	-19.703	0.000	-30.195	-24.731
high-Retail-Res-338911_dist	-9.9571	1.077	-9.248	0.000	-12.068	-7.847
high-Retail-Res-448011_dist	-21.4297	2.174	-9.855	0.000	-25.692	-17.168
high-Retail-Res-450311_dist	0.3393	0.026	13.079	0.000	0.288	0.390
high-Transportation-18135311_dist	-1.0254	0.107	-9.554	0.000	-1.236	-0.815
high-Transportation-444911_dist	2.6736	0.415	6.449	0.000	1.861	3.486
high-WWTP-379511_dist	0.0058	0.167	0.035	0.972	-0.322	0.334
high-Waste-10708011_dist	-13.0775	2.679	-4.882	0.000	-18.328	-7.827
high-Waste-168411_dist	1.0897	0.102	10.725	0.000	0.891	1.289
high-Waste-18782611_dist	-3.8877	1.119	-3.474	0.001	-6.081	-1.694
Mintemp	1.0414	0.116	9.004	0.000	0.815	1.268
Pressure	-0.0284	0.002	-17.120	0.000	-0.032	-0.025
Dist	-0.1226	0.015	-8.152	0.000	-0.152	-0.093
number_intersections	0.0018	0.001	2.057	0.040	8.62e-05	0.004

Omnibus:	7523.079	Durbin-Watson:	1.483
Prob(Omnibus):	0.000	Jarque-Bera (JB):	55141.437
Skew:	1.493	Prob(JB):	0.00
Kurtosis:	10.258	Cond. No.	2.19e+06

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.19e+06. This might indicate that there are strong multicollinearity or other numerical problems.

A scatter plot of the predicted vs actual concentrations are shown below, with an R2 value of 0.827.

Actual vs. Predicted BC Concentration
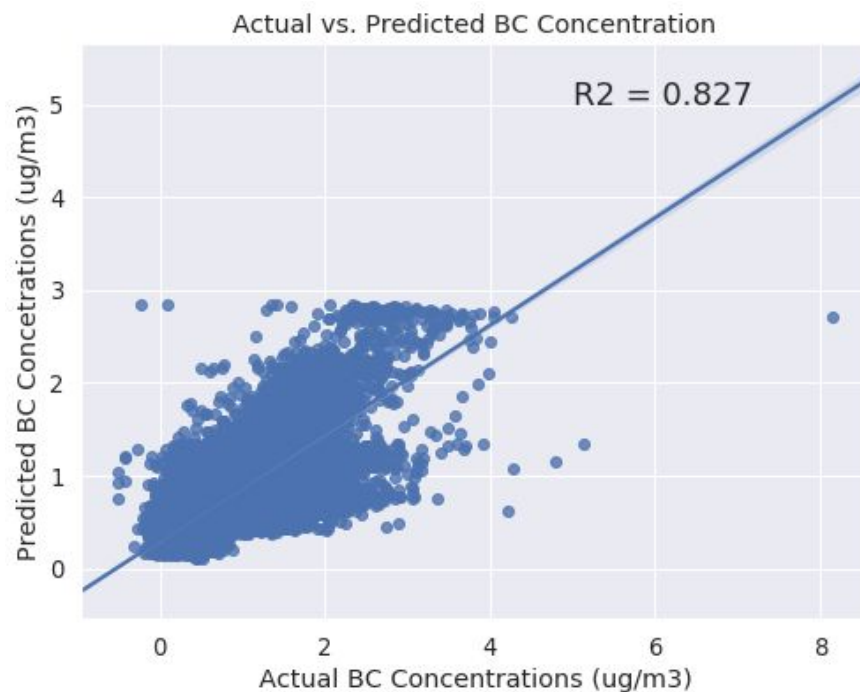
This indicates that the Lasso regression approach outperforms both the simple OLS and the Step forward VIF approach.

However, most of the correlation coefficients are really high and the OLS model indicates that there is strong multicolinearity present. To overcome this, once again we perform a stepforward VIF approach, where we start with the 49 features that were selected by the Lasso model, and sequentially add features that results in the maximum R2 and lowest VIF scores.

Results of the step forward VIF approach for Lasso:

Only the 'Mintemp' feature is selected by the step forward VIF approach and the R2 for the training set was 0.592, and the test set was 0.583.

NO2 dataset:
For the NO2 dataset,

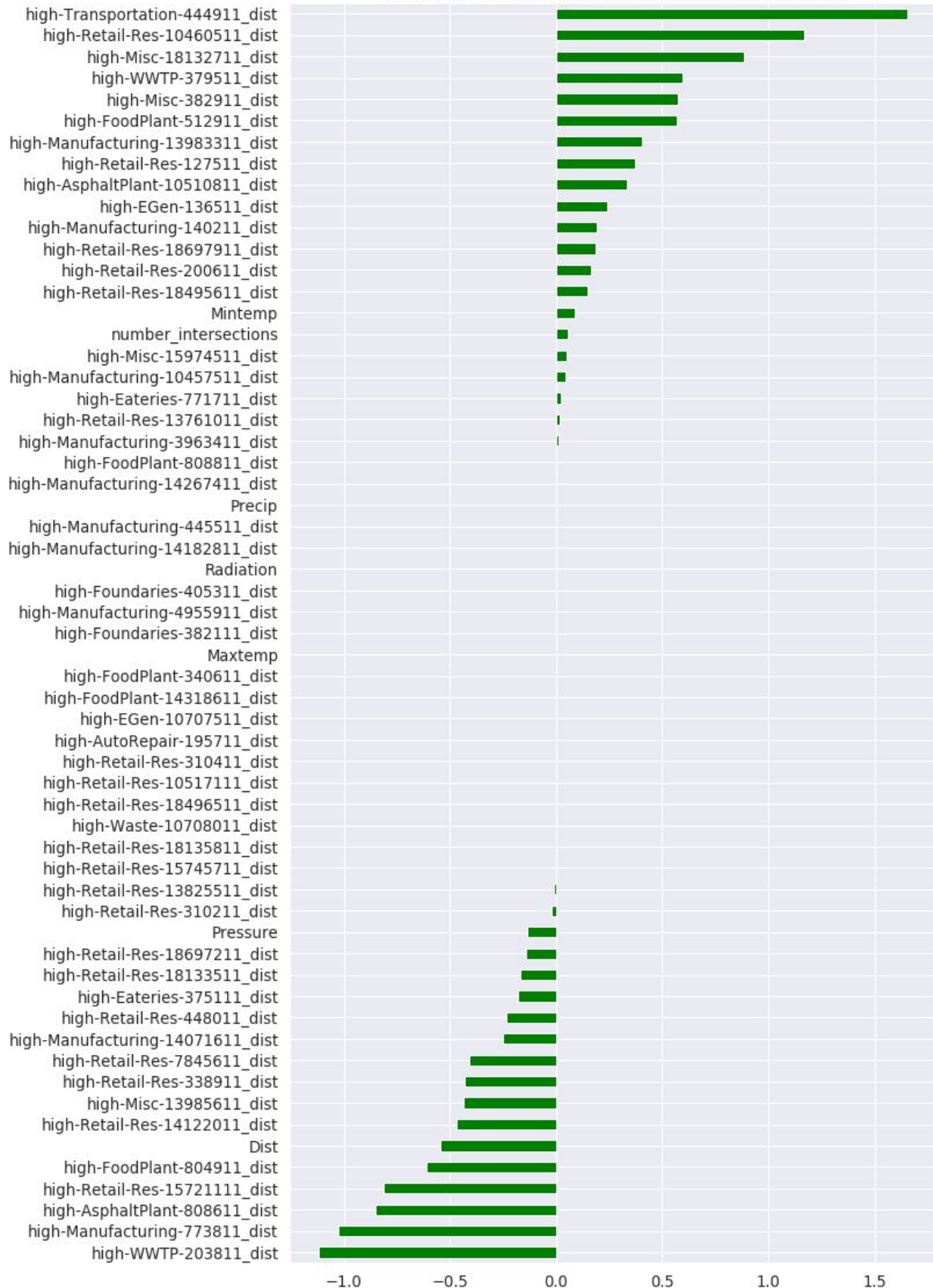```
Best Alpha using LassoCV = 0.00038952008419926016
MSE on test data = 0.20860434953284926
Best score using LassoCV: 0.6395777236551858
```

The features that were selected are as follows. In total, the lasso approach selected 42 features out of 61 features.

Feature importance using Lasso Model for NO2 Dataset

# Results of the OLS model fit:

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | NO2_Value | R-squared (uncentered): | 0.912 |
| Model: | OLS | Adj. R-squared (uncentered): | 0.911 |
| Method: | Least Squares | F-statistic: | 5268. |
| Date: | Sun, 31 May 2020 | Prob (F-statistic): | 0.00 |
| Time: | 11:13:52 | Log-Likelihood: | -68511. |
| No. Observations: | 21488 | AIC: | 1.371e+05 |
| Df Residuals: | 21446 | BIC: | 1.374e+05 |
| Df Model: | 42 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| high-AsphaltPlant-10510811_dist | 32.3134 | 2.068 | 15.622 | 0.000 | 28.259 | 36.368 |
| high-AsphaltPlant-808611_dist | -8249.1066 | 1179.534 | -6.994 | 0.000 | -1.06e+04 | -5937.132 |
| high-EGen-136511_dist | 37.1751 | 3.570 | 10.413 | 0.000 | 30.178 | 44.173 |
| high-Eateries-375111_dist | 4945.4679 | 731.520 | 6.761 | 0.000 | 3511.634 | 6379.301 |
| high-Eateries-771711_dist | -16.3971 | 3.835 | -4.276 | 0.000 | -23.913 | -8.881 |
| high-FoodPlant-512911_dist | -74.7029 | 11.637 | -6.420 | 0.000 | -97.512 | -51.894 |
| high-FoodPlant-804911_dist | -39.5776 | 4.700 | -8.421 | 0.000 | -48.789 | -30.366 |
| high-FoodPlant-808811_dist | -11.2241 | 2.503 | -4.483 | 0.000 | -16.131 | -6.317 |
| high-Manufacturing-10457511_dist | 9.1533 | 2.327 | 3.934 | 0.000 | 4.593 | 13.714 |
| high-Manufacturing-13983311_dist | 13.4396 | 0.718 | 18.713 | 0.000 | 12.032 | 14.847 |
| high-Manufacturing-140211_dist | 32.5681 | 4.404 | 7.395 | 0.000 | 23.936 | 41.201 |
| high-Manufacturing-14071611_dist | 732.8911 | 86.306 | 8.492 | 0.000 | 563.725 | 902.057 |
| high-Manufacturing-14267411_dist | -4.7073 | 1.479 | -3.182 | 0.001 | -7.607 | -1.808 |
| high-Manufacturing-3963411_dist | 148.9537 | 25.301 | 5.887 | 0.000 | 99.362 | 198.545 |
| high-Manufacturing-773811_dist | -81.1912 | 3.262 | -24.888 | 0.000 | -87.586 | -74.797 |
| high-Misc-13985611_dist | -17.7073 | 2.432 | -7.281 | 0.000 | -22.474 | -12.940 |
| high-Misc-15974511_dist | -18.6592 | 2.763 | -6.753 | 0.000 | -24.075 | -13.243 |
| high-Misc-18132711_dist | 18.4679 | 1.443 | 12.794 | 0.000 | 15.639 | 21.297 |
| high-Misc-382911_dist | 2.1365 | 0.513 | 4.166 | 0.000 | 1.131 | 3.142 |
| high-Retail-Res-10460511_dist | -8.7317 | 6.954 | -1.256 | 0.209 | -22.363 | 4.900 |
| high-Retail-Res-127511_dist | 3.4740 | 1.077 | 3.225 | 0.001 | 1.362 | 5.586 |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| high-Retail-Res-13761011_dist | -8.7124 | 1.868 | -4.664 | 0.000 | -12.374 | -5.051 |
| high-Retail-Res-13825511_dist | -549.0568 | 80.329 | -6.835 | 0.000 | -706.508 | -391.605 |
| high-Retail-Res-14122011_dist | -4.4559 | 0.681 | -6.539 | 0.000 | -5.792 | -3.120 |
| high-Retail-Res-15721111_dist | -52.3147 | 3.257 | -16.060 | 0.000 | -58.699 | -45.930 |
| high-Retail-Res-18133511_dist | 7.1740 | 4.101 | 1.749 | 0.080 | -0.864 | 15.212 |
| high-Retail-Res-18495611_dist | 5.0519 | 1.102 | 4.584 | 0.000 | 2.892 | 7.212 |
| high-Retail-Res-18697211_dist | 1.1972 | 1.982 | 0.604 | 0.546 | -2.688 | 5.083 |
| high-Retail-Res-18697911_dist | -0.9889 | 1.588 | -0.623 | 0.533 | -4.101 | 2.123 |
| high-Retail-Res-200611_dist | -1.8877 | 1.461 | -1.292 | 0.196 | -4.751 | 0.975 |
| high-Retail-Res-310211_dist | 3.4964 | 1.734 | 2.017 | 0.044 | 0.098 | 6.894 |
| high-Retail-Res-338911_dist | 343.1114 | 48.150 | 7.126 | 0.000 | 248.733 | 437.490 |
| high-Retail-Res-448011_dist | 295.8299 | 41.376 | 7.150 | 0.000 | 214.730 | 376.929 |
| high-Retail-Res-7845611_dist | 3214.2432 | 443.922 | 7.241 | 0.000 | 2344.123 | 4084.364 |
| high-Transportation-444911_dist | 80.2390 | 4.880 | 16.441 | 0.000 | 70.673 | 89.805 |
| high-WWTP-203811_dist | -796.7642 | 98.315 | -8.104 | 0.000 | -989.468 | -604.060 |
| high-WWTP-379511_dist | 10.4299 | 1.830 | 5.699 | 0.000 | 6.842 | 14.017 |
| Precip | 3.2921 | 2.402 | 1.370 | 0.171 | -1.416 | 8.000 |
| Mintemp | 12.1718 | 2.307 | 5.277 | 0.000 | 7.650 | 16.693 |
| Pressure | -0.2900 | 0.033 | -8.854 | 0.000 | -0.354 | -0.226 |
| Dist | -2.2286 | 0.417 | -5.350 | 0.000 | -3.045 | -1.412 |
| number_intersections | 0.1638 | 0.013 | 12.745 | 0.000 | 0.139 | 0.189 |

| | | | |
|---|---|---|---|
| Omnibus: | 3716.569 | Durbin-Watson: | 1.331 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 8745.204 |
| Skew: | 0.986 | Prob(JB): | 0.00 |
| Kurtosis: | 5.425 | Cond. No. | 3.34e+07 |

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.34e+07. This might indicate that there are strong multicollinearity or other numerical problems.

A plot of the predicted vs actual concentrations are shown below, and the model has an R2 value of 0.912.

Actual vs. Predicted NO2 Concentration
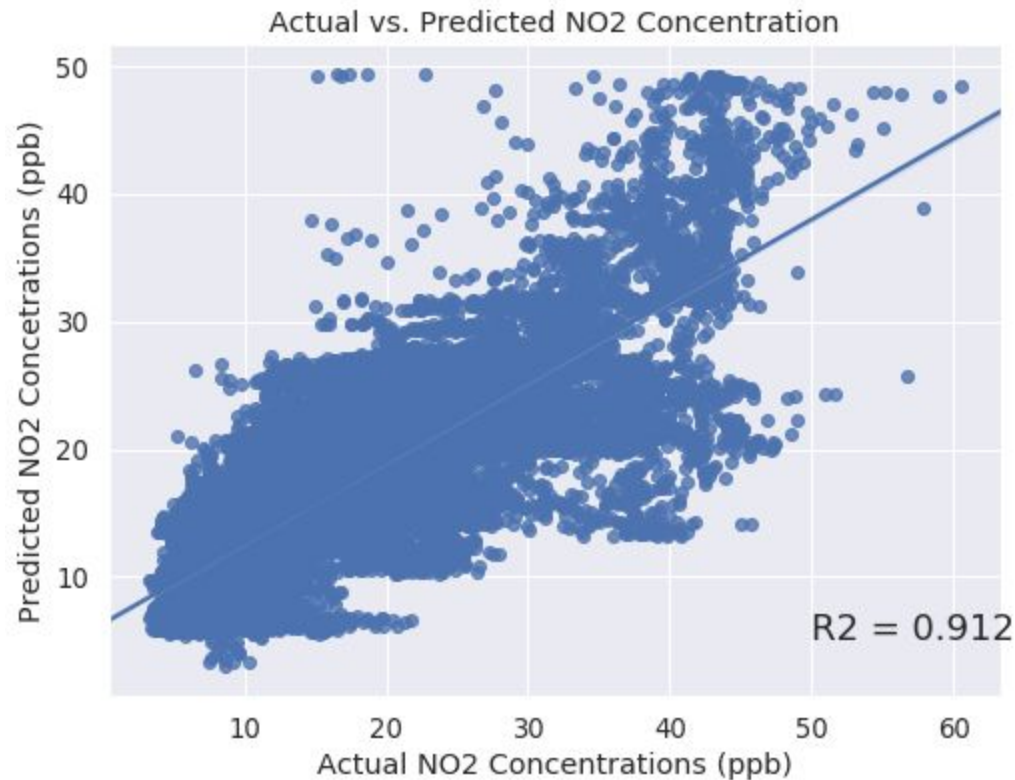
The above approach indicates that the Lasso approach outperforms the simple OLS and the stepforward VIF methodology, resulting in a much better prediction, even at high concentrations.

However, most of the correlation coefficients are really high and the OLS model indicates that there is strong multicolinearity present. To overcome this, once again we perform a stepforward VIF approach, where we start with the 42 features that were selected by the Lasso model, and sequentially add features that results in the maximum R2 and lowest VIF scores.

**Results of the step forward VIF approach for Lasso:**
Similar to the BC dataset, only min temp was selected as feature and R2 value for the training set was 0.759, and test set was 0.756.

<u>**Summary:**</u>
The lasso approach results in a much better model fit than simple OLS. But the features selected by the lasso model are still highly correlated with each other which may not result in a great fit for a validation set. The step forward VIF approach with lasso resulted in the selection of features that had lower model coefficients, and also gave us some insight on feature importance.

# Feature Selection Using Random Forest

Feature selection using the Random Forest approach comes under the category of embedded methods as it combines qualities of filter and wrapper methods. Feature selection in random forest is done by building sevel decision trees, and each tree is built over a random extraction of the observations from the dataset and a random extraction of the features. Not every tree sees all the features or all the observations, and this guarantees that the trees are de-correlated and therefore less prone to over-fitting.

## Methodology:

Once again, we split the data into test and train set and apply the RandomForestRegressor() function, and the feature importance attribute to understand the importance of features. Here, n_estimators is set at 100.

## Results:
**BC dataset:**

Graphs below show the feature importance for training and test data



Feature selection using training data for BC Dataset

Feature selection using test data for BC Dataset
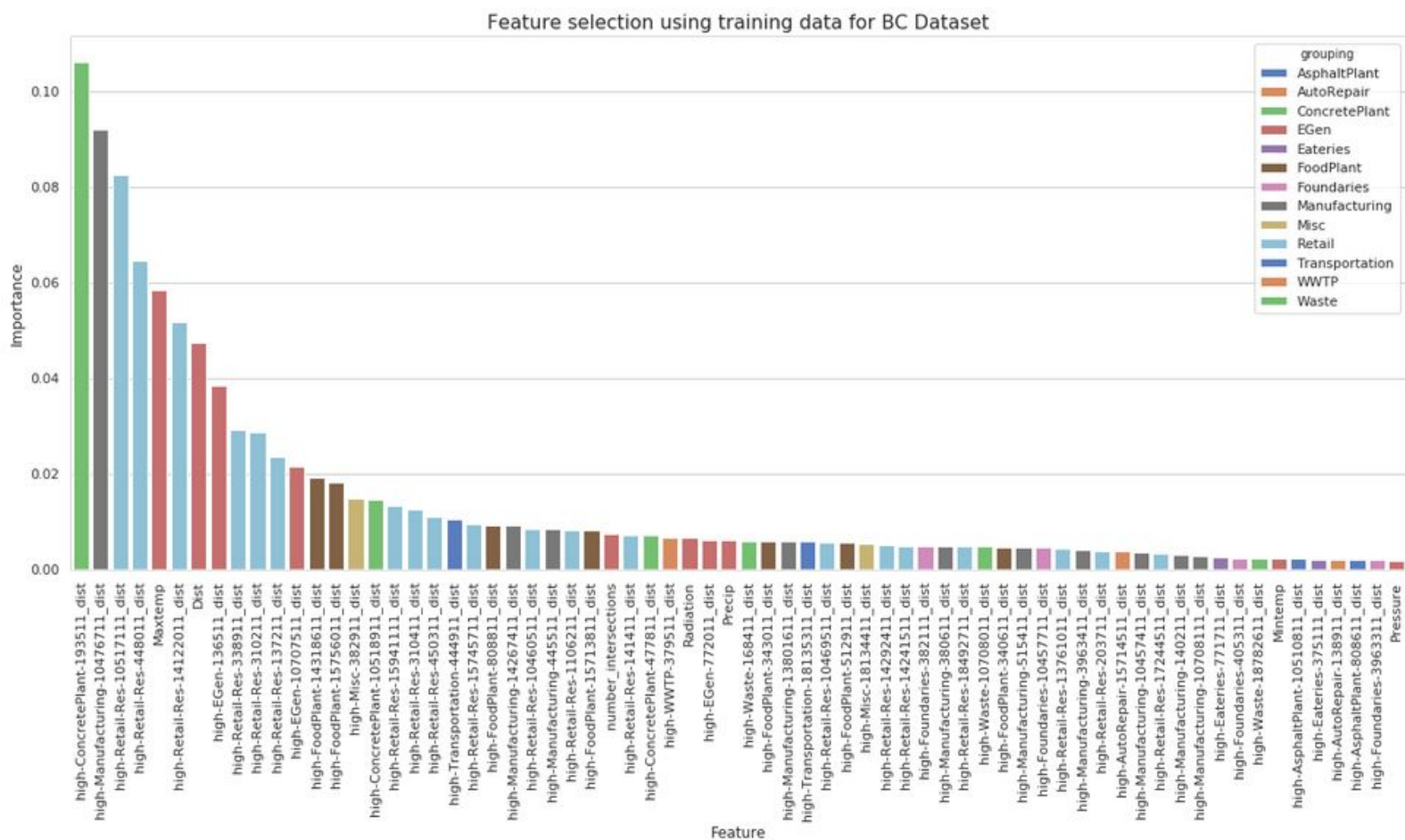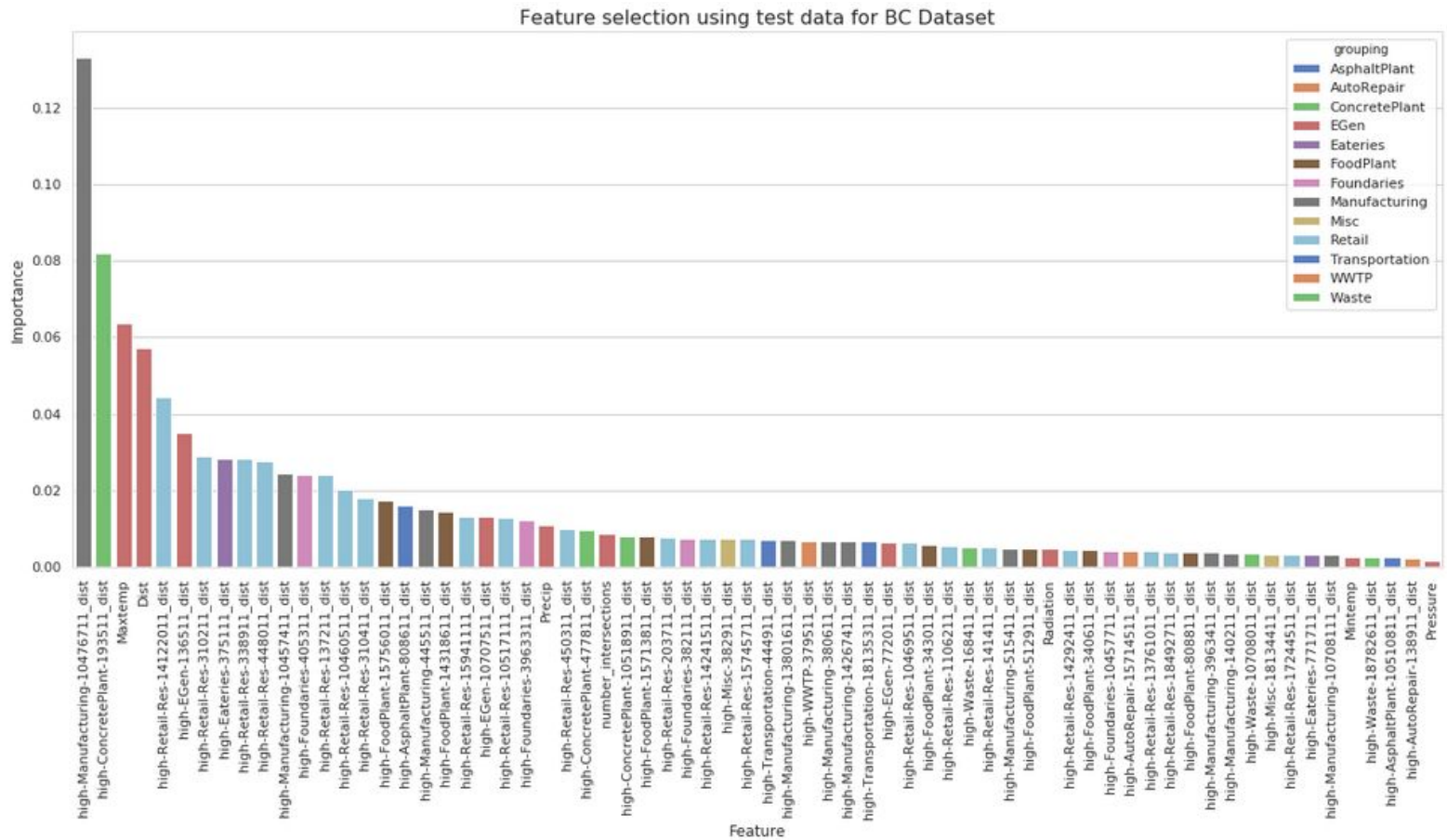
Results show that for the BC dataset, **'Manufacturing-10476711', 'ConcretePlant-193511'** are the top two features.

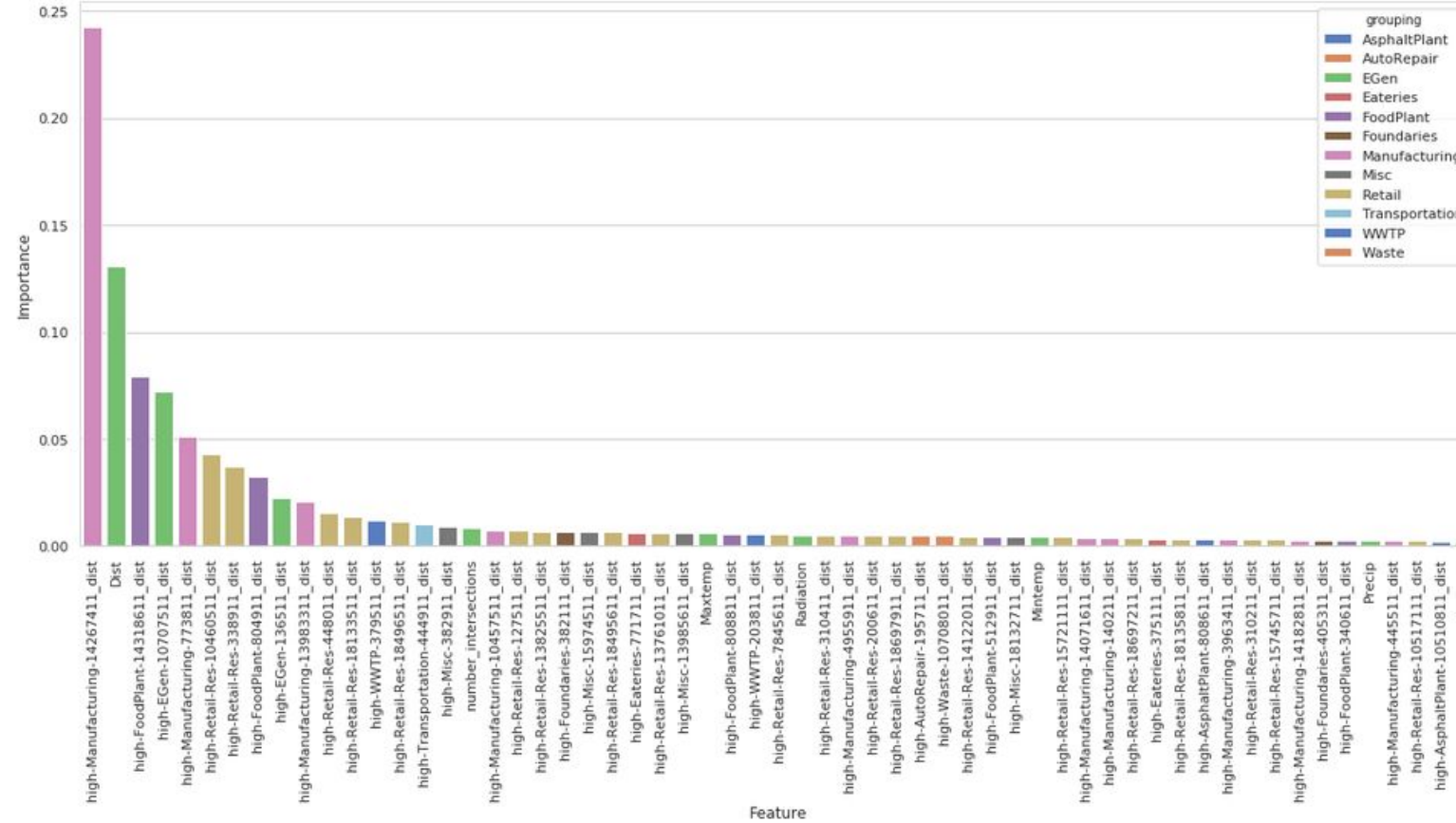**NO2 dataset:**

Graphs below show the feature importance for training and test data. Results show that for the NO2 dataset, **'Manufacturing-14267411', 'Dist', 'EGen-10707511' and 'FoodPlant-14316811'** are the top few features.

Feature selection using training data for NO2 Dataset
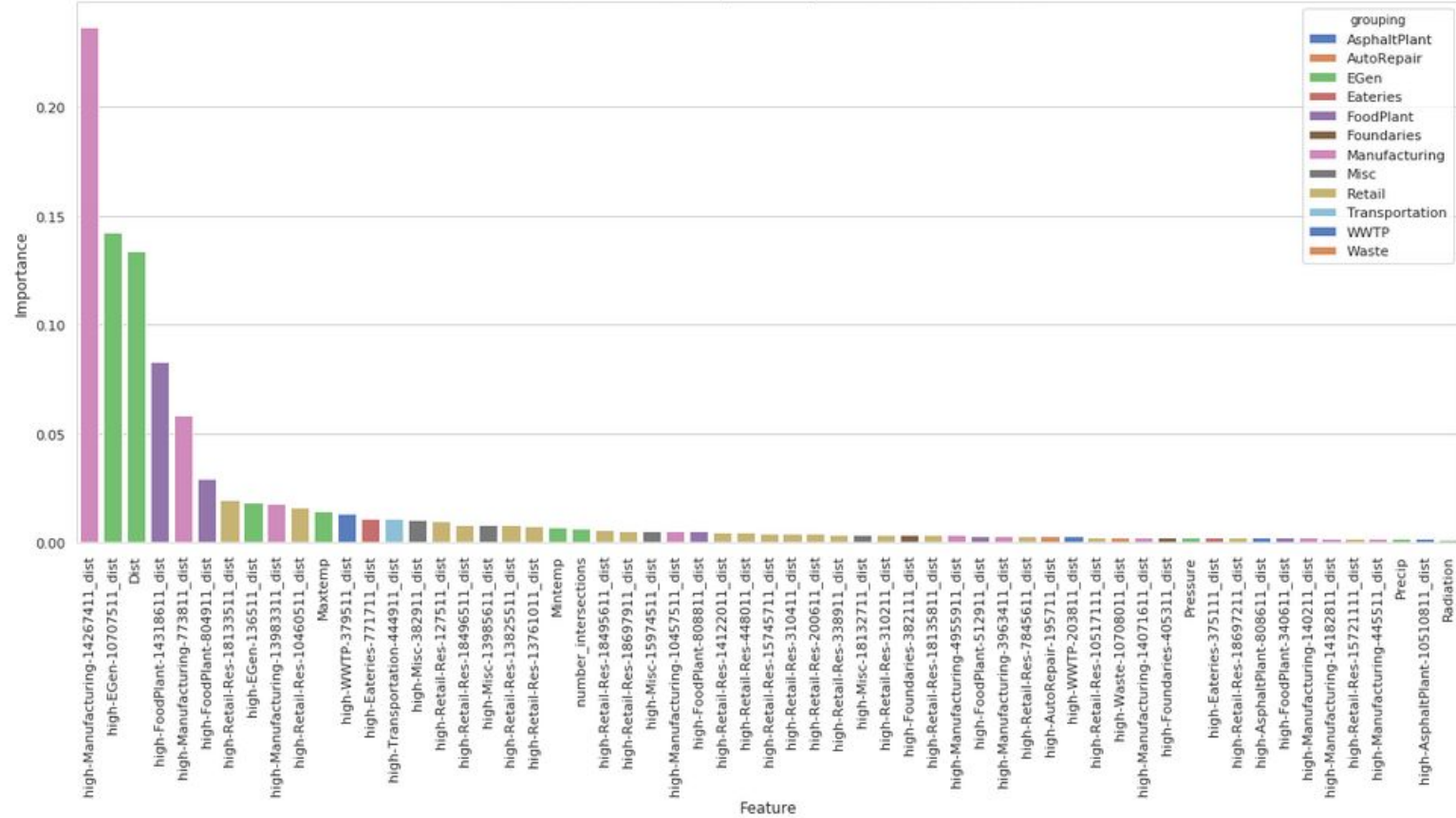


Feature selection using test data for NO2 Dataset

# Apply PCA And Test Performance of Several Models

We performed (results not shown here) a  PCA + DecisionTree Regressor that resulted in the selection of 31 components, and the mean cross validation score for the BC test dataset was 0.60. Similarly, the NO2 dataset resulted in the selection of 50 PCs with a cross validation score of 0.78. In this case, we test whether applying PCA really improves the model performance. We test several different models on the training and test set and calculate the R2 for the model with PCA and without PCA.

## Methodology:

Similar to the above approach, we first scale the dataset using StandarScaler() method such that each column has a mean = 0 and variance = 1. After this, we create a PCA object and set the variance to be 0.99. Next, we test different models using 5-fold cross validation on the test dataset. The different models that were tested include:
1) Linear Regression
2) Ridge Regression
3) RidgeCV Regression
4) Huber Regression
5) Bayesian Ridge
6) Bagging
7) Random Forest

## Results:

**BC Dataset:**
The R2 score for the training data with and without PCA for the BC dataset is shown below. The results show that applying PCA does not really improve the model performance, at least for the test data.

### R2 with PCA for training data:

|  | R Square Score |
|---|---|
| Linear | 0.588995 |
| Ridge | 0.546910 |
| RidgeCV | 0.578782 |
| Bayesian Ridge | 0.578883 |
| Hubber | 0.492165 |
| Bagging | 0.775148 |
| RandomForest | 0.790851 |

### R2 with PCA for test data

|  | R Square Score |
|---|---|
| Linear | 0.574383 |
| Ridge | 0.527219 |
| RidgeCV | 0.563863 |
| Bayesian Ridge | 0.562670 |
| Hubber | 0.479324 |
| Bagging | 0.714963 |
| RandomForest | 0.743671 |

### R2 without PCA for test data:

|  | R Square Score |
|---|---|
| Linear | 0.555311 |
| Ridge | 0.541292 |
| RidgeCV | 0.550310 |
| Bayesian Ridge | 0.545387 |
| Hubber | 0.522175 |
| Bagging | 0.696226 |
| RandomForest | 0.715134 |

**NO2 Dataset:**

The R2 score for different models are shown below:

### R2 with PCA for training data:

|  | R Square Score |
| --- | --- |
| Linear | 0.649465 |
| Ridge | 0.616807 |
| RidgeCV | 0.636670 |
| Bayesian Ridge | 0.635459 |
| Hubber | 0.540464 |
| Bagging | 0.884280 |
| RandomForest | 0.893266 |

### R2 with PCA for test data:

|  | R Square Score |
| --- | --- |
| Linear | 0.637097 |
| Ridge | 0.598221 |
| RidgeCV | 0.627137 |
| Bayesian Ridge | 0.624936 |
| Hubber | 0.528650 |
| Bagging | 0.863286 |
| RandomForest | 0.876086 |

### R2 without PCA for test data:

| | R Square Score |
| --- | --- |
| Linear | 0.663888 |
| Ridge | 0.641820 |
| RidgeCV | 0.654200 |
| Bayesian Ridge | 0.643589 |
| Hubber | 0.626353 |
| Bagging | 0.878333 |
| RandomForest | 0.889857 |

The above results show that applying PCA does not really improve the model performance, at least for the test data.


# Gridsearch different models for comparison

We adopt a gridsearch and cross validation to test the performance of different model based on the prediction score. Here, we first split the data into test train split, and then test Ridge regression, Random Forest and the XGBoost approach. For each model, we perform a gridsearch to identify the best parameters (such as optimal tree depth, number of estimators etc.) in order to find the best parameters for each model. The ridge regression is also tested with and without PCA.

**Methodology:**

First , split the data into test and train set. Next,  we first scale the dataset using StandarScaler() method such that each column has a mean = 0 and variance = 1. Next, define grid parameters for different types of model and perform a gridsearch CV over each. The first model is a RandomforestRegressor. Grid parameters are number of estimators, max_features, and max_depth. Second model is a Ridge Regression model. Grid parameters include Alpha and fit_intercept. Third model is XGBoost, with grid parameters like max_depth, learning rate and number of estimators.

## Results:

### BC Dataset

| Model Type | Best Parameter | Mean CV score for best estimator | Training R2 | Test R2 |
|---|---|---|---|---|
| Ridge regression with PCA | Alpha = 0 | 0.588 | 0.594 | 0.577 |
| Ridge regression without PCA | Alpha = 0 | 0.588 | 0.594 | 0.577 |
| Random Forest | Max depth = 15 Estimators = 300 | 0.782 | 0.921 | 0.769 |
| XGBRegressor | Learning rate = 0.1 max depth = 10 estimators = 100 | 0.798 | 0.925 | 0.791 |

### NO2 Dataset

| Model Type | Best Parameter | Mean CV score for best estimator | Training R2 | Test R2 |
|---|---|---|---|---|
| Ridge regression with PCA | Alpha = 0 | 0.649 | 0.652 | 0.640 |
| Ridge regression without PCA | Alpha = 0 | 0.649 | 0.652 | 0.640 |
| Random Forest | Max depth = 50 Estimators = 400 | 0.888 | 0.985 | 0.898 |
| XGBRegressor | Learning rate = 0.1 | 0.899 | 0.967 | 0.91 |

| | max depth = 10 estimators = 100 | | | |
| --- | --- | --- | --- | --- |

**Summary:**

XGBRegressor performs the best, followed by random forest