CE 475 - Term Project Report

Inan Evin

2014 060 1024

Identification and Significance of the Problem

The project is based on a dataset consisted of 100 observations. These observations include 5 different predictors and their values for each observation. We are simply requested to develop a model, using the data in hand to predict another set of points using only the predictors.

Although it might seem like a simple task, it is the very basis of the fundamentals of understanding how machine learning works. In simpler forms, our problem is predicting possible future outcomes that can be generated with a set of given input variables. We are required to develop a model, more basically a function, that can be used to determine the response using various inputs, with the aim of achieving high accuracy, without knowing the actual underlying function, just like it is in most of the real world fields.

The significance of the problem derives from the very fact that our case is similar to the cases in real world fields. Nowadays, with technologies like deep learning, data-mining and gathering, the big data etc. we are bound to live in a world in which the most valuable currency is information. Stock markets, real-estates, gambling, global aid projects, health & living related fields and many more, depend on future prediction of possible outcomes. Everywhere, in every computer, data is gathered and collected throughout almost every single interaction done by any casual user online. All these data, serve great purpose for statistics and inspection, however, most fields are not only interested in past data, they are mostly focused on predicting the future ones. This is where machine learning comes into place. We are using machine learning algorithms to inspect the past data and make assumptions on the future ones. And the problem given to us in this project, even though it is not a very complex one, it is the underlying basis of understanding how more complicated applications of machine learning works. Thus, being able to understand, inspect, work on, study and produce solutions for our given problem, will give us a better view and understanding of further studies in machine learning fields.

Methodology

Given our task, we are firstly required to inspect the data points, in order to understand how the predictors have an effect on the response value, as well as to acquire hints about the underlying model. The very first thing I have tried to resolve was the linearity of the underlying model. In order to do this, I have started thinking the model as a linear one. In more technical means, my first hypothesis was the model being linear. Even though it was unlikely to be linear by looking at the spreadsheet as a draft, but it was still a good point to start so that I can eliminate different states of the underlying model from smaller to biggest pieces.

Non-Linearity

Since we had multiple predictors, I had run some multiple linear regression tests. I had separated the data into test and train sets, and used a simple Ridge to train a basic linear model. Then I had used **Residuals Plotting** to inspect the dataset and understand that the underlying model was more complicated than a simple linear model. ***(Results 1.1)***

In order to become more sure about the non-linearity, I had run a few more tests about the predictors. I had plotted predictors individually against the responses to detect if any linearity, or any other obvious relation is present between them that would help me in my further analysis.

Predictor Analysis - Collinearity

After deciding that the model I am supposed to develop is a non-linear one, without taking any further steps, I needed to decide on whether one or more of the predictors were unrelated to the response values. This is a very crucial step since if we more further with the wrong predictors, all of our way of thinking and analysis would be affected in the ways that we would not want.

In order to assess the collinearity, firstly I had checked the **correlation matrix** of the predictors. (***Results 1.2***) This way, I would be able to detect negative or positive collinearity between a pair of predictors. I found out that there were no significantly correlated predictors. However, even though checking correlation matrix is a great way of gathering information, it was not enough. This is due to the fact that it is possible for a pair of variables not to have any correlation, but a set of 3 variables, or more might have a strong relation between them, which is called **Multicollinearity** and correlation matrix does not show us this data. So I had to calculate **Variance Inflation Factor (VIF)** to see if multicollinearity between the predictors exists. The biggest value I got from VIF was around 3.6, which does not indicate a multicollinearity strong enough to remove one variable. (***Results 1.3***)

Model & Subset Selection, Evaluation

Linear Regression

After I had my initial inspection on the predictors, I decided to move on to selecting the right subset of predictors to develop my model. The reason to classify subsets was simple, the more predictors we had, we would have less RSS on the training data, which is fine. However, while developing a model what we should focus on is the test data, since focusing on the training data might introduce overfitting or underfitting in most circumstances. Thus, I started looking at the options for subset selection. The first idea was using the **Best Subset Selection**, as it was the simplest one to go. Even though best subset selection is still a simple and efficient method to go, I did not choose it due to some disadvantages it presents. The biggest one is the fact that there are computational restrictions to it, as the number of predictors increase the computational time grows devastatingly large. Even though we only have 5 predictors and 32 possible models to assess and it would not take too much computational effort, I genuinely do not trust hard-to-scale methods in general. So, I had decided to go with a more concrete approach, which was stepwise selection. According to my first research, I decided to try out **forward selection** and **backward selection** to check my subsets and compare these different stepwise selection methods to figure out which subset of predictors I should use. But before doing so, I had one more thing to decide. What kind of a criteria should I use to check the accuracy of these methods? It is not feasible to look at RSS and R^2 directly, as they are measurements mostly related with the training error. There are few approaches like Cp, Akaike information criterion (AIC), Bayesian information criterion (BIC), and adjusted R^2. Honestly, at first glance, I did not know which approach to take. Upon some research, BIC, AIC and Cp all looked like they would round around at very similar results, since I only had a small number of predictors for them to actually derive huge differences. I decided on using p values and adjusted R^2. However assessing my subset selection results with a directly test data related approach looked better on paper, so I had decided to use Cross Validation as well. It definitely has an advantages over other methods, since it provides a direct estimation of the test error and it makes fewer assumptions on the underlying model. I had used **Negative Mean Squared Error** while scoring the cross validation since it is a flexible approach when non-linearity is considered.

Initially, I had tried fitting a multiple linear regression without polynomial features and running stepwise elimination according to the p-values and adjusted R^2 scores to gain an insight about the candidate subsets. After running backwards elimination, the subset of X2 and X3 was found to be the best result. (***Results 1.4***) Then I had run a forward selection, which has given me the same result, the subset of X3 and X2 according to the p threshold of 0.05. However, I had run this test just for an insight, since it did not have any polynomial features added it would not be a direct result to choose.

Secondly, I had tried fitting a multiple linear regression model with polynomials, running different subset selections and k-fold cross validation to find out which subsets and what degree of polynomials were the best fit. I had not used Leave One Out Cross Validation. The advantage of using LOOCV is that it provides fairly accurate results, however it brings the drawback of spending too much computational power. Instead, I had used k-fold cross validation with different k's to figure out the best subset. Tests with k = 5,10, 20 showed that the subset [X2,X3] with the polynomial degree 3 had the best test results. As for the others, k value of 10 and 20 showed that the subset [X2,X3,X5] with the degree 3 was the second best result, whilst k value of 5 showed that the subset [X2,X3,X4] with the degree 2 had the best result. (***Results 1.5***)

Lasso and Ridge CV

Upon running the tests above, I still had my doubts about the subsets. So I had tried 2 different shrinkage methods to see what else can I do about minimizing the effects of the predictors. I had tried these shrinkage methods with possible subsets from the tests before, with their relative polynomial features added. The best test score I got was the subset [X2,X3] with polynomial of degree 2 with Ridge CV. Second best result was the same subset with Lasso CV. The third best result I had achieved was the subset [X2,X3,X5] with 3rd degree polynomial features added with Lasso CV. (***Results 1.6***)

Support Vector Machines

In the further chapters of our book, solid algorithms about support vector machines are mentioned to run predictions. SVM's are great for both classification and regression problems. Even though they have a disadvantage of their training time being too much and they require computational power, since our dataset is not a huge one, I had decided to try to run regression with SVM to see results. I had the best result so far with the SVM with the subset [X2,X3]. Also, there was a huge improvement with the subset [X2,X3 and X5] which became the second best result so far. (***Results 1.7***)

Trees

Even though prediction using Trees have few considerable disadvantages like accuracy problems, the ease of use they bring and the algorithms provided to increase their performance and accuracy made me have a look into the tree section of our book for the model selection. I have decided on trying out the Random Forest algorithm to see if I would get a better results than SVM. I had tried trees with different hyperparameters to check various results. I had used Feature Importances with my defined threshold of 0.05 to lastly see the best subset. Even though I did not want to pick so less predictors, the subset [X2,X3] gave the best results, as with all other regressions I had run so far. The predictors X1, X4 and X5 had stayed below my threshold (***Results 1.9***), and I had a tiny insight of this when I had plotted the individual predictors versus the outputs. The best hyperparameter I got was **500** with the mean of -0.20434~**,** however I felt more comfortable using a smaller number, so I chose 100 as it gave the second best results with the mean test score of -0.2063~(***Result 1.8***), which was the smallest error I have seen so far. So I decided to go with the Random Forest algorithm for my prediction. (***Results 2.0***)

Implementation

I had used Python to develop my model and my preference of IDE was PyCharm by JetBrains. I had run my observations and algorithms on a Windows 10 machine with 8 processors, over more than 4.8 ghz of clock speed. Mostly, I did not run into the issue of waiting for long times for my algorithms to complete, since our dataset was relatively small.

I had used many libraries, for plotting, inspection, utility and regression purposes. These libraries can be listed below:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import statsmodels.api as sm
from sklearn.linear_model import Ridge
from sklearn.model_selection import train_test_split
from yellowbrick.regressor import ResidualsPlot
from statsmodels.stats.outliers_influence import
variance_inflation_factor
from sklearn.linear_model import LinearRegression, LassoCV, RidgeCV
from sklearn.svm import SVR
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import cross_val_score, cross_validate
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
```

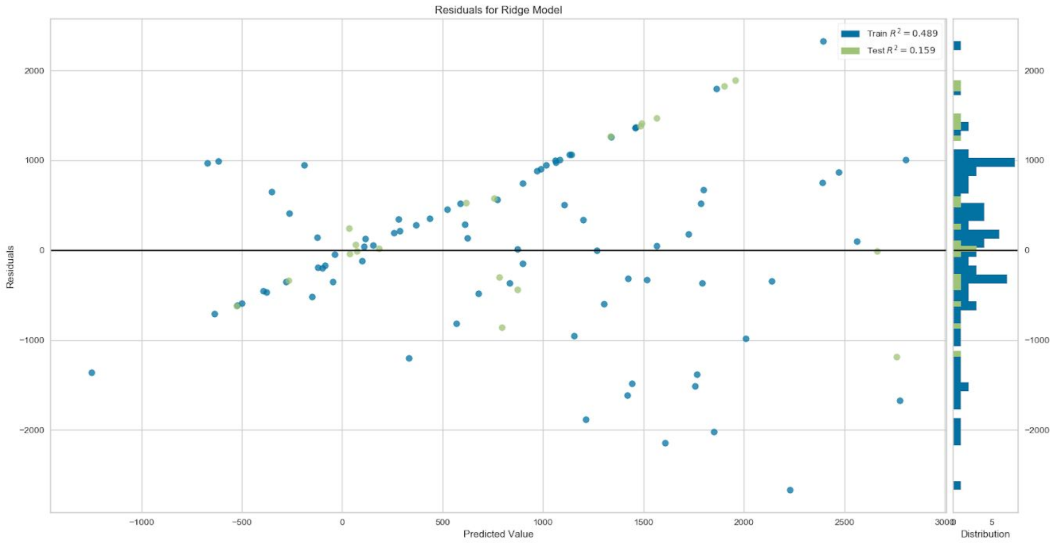
I had used matplotlib, yellowbrick and seaborn for mostly plotting and inspection purposes. I used pandas for data manipulation and data frames as well as numpy. I had used sklearn for model selection, adding polynomials, cross validation and normalization, as well as Ridge, Lasso, SVM and Random Forest machines.. I had used statsmodels for OLS resulting and VIF calculation.

I had mainly worked on a single .py file with different functions for observation purposes. Sometimes I had to run my methods on Python Console without stopping the execution of the current main file to see how my data changes, especially while doing backward elimination. Other than that the inspection of variables through the event log and python console was a big help while trying to understand the data.

All implementations can be found inside the file I have provided (Main.py). I have separated everything into corresponding functions for ease of inspection and commented the necessary lines for descriptions.

Results

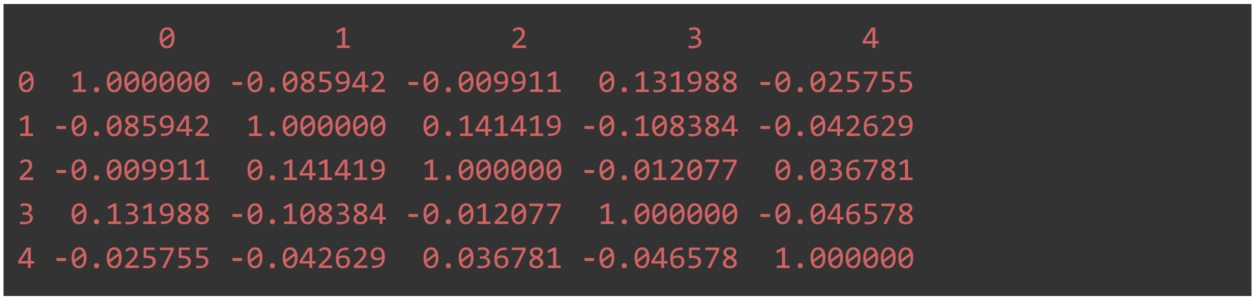
Results 1.1 - Residuals Plot For Multiple Linear Regression



Checking the residuals plot for patterns to understand linearity, as well as predictors vs responses to understand their weight.

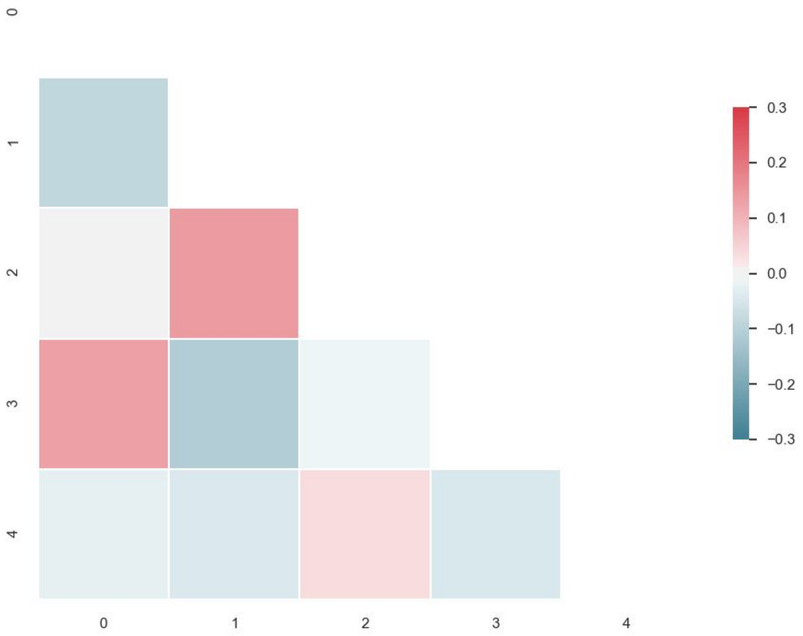
Results 1.2 - Correlation Matrix of the Predictors

Upon printing the correlation matrix, I had the results:



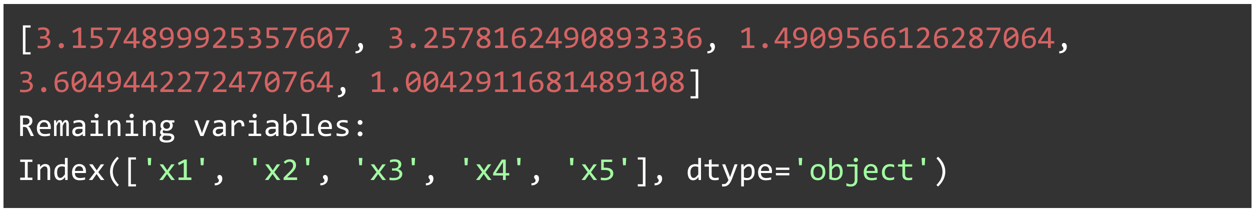
```
          0         1         2         3         4
0  1.000000 -0.085942 -0.009911  0.131988 -0.025755
1 -0.085942  1.000000  0.141419 -0.108384 -0.042629
2 -0.009911  0.141419  1.000000 -0.012077  0.036781
3  0.131988 -0.108384 -0.012077  1.000000 -0.046578
4 -0.025755 -0.042629  0.036781 -0.046578  1.000000
```

These indicate that there is not any strong correlation between the pairs of predictors. Below can be found the heatmap, ranged from -0.3 to 0.3 for convenient viewing.



Results 1.3 - Variance Inflation Factor (VIF)

As mentioned, the correlation matrix detects the collinearity with the pairs. However, we still had multicollinearity to worry about, so I had calculated VIF.

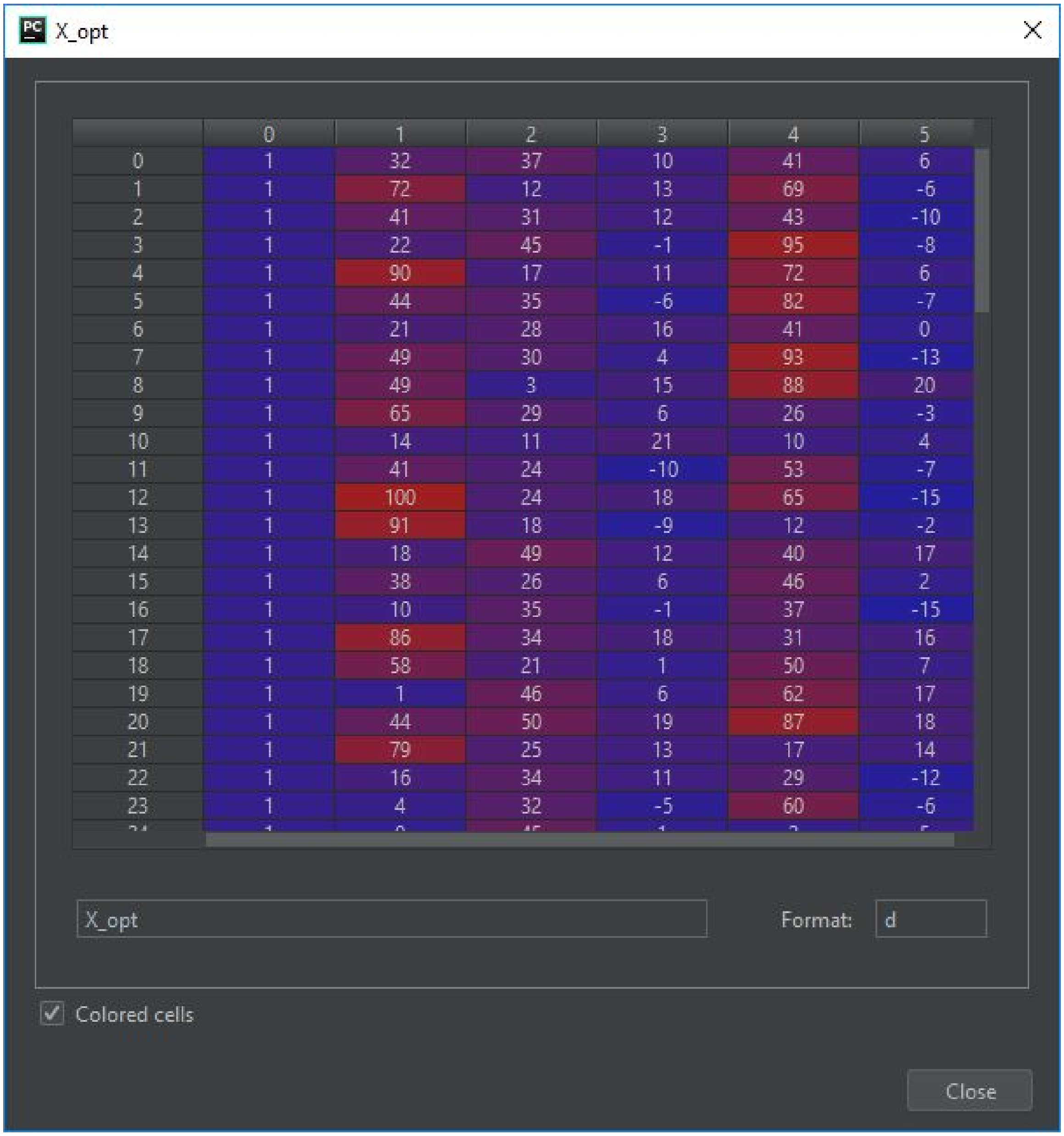


```
[3.1574899925357607, 3.2578162490893336, 1.4909566126287064,
3.6049442272470764, 1.0042911681489108]
Remaining variables:
Index(['x1', 'x2', 'x3', 'x4', 'x5'], dtype='object')
```

Upon checking VIF values, I could not see any variable to remove just yet.

Results 1.4 - Subset Selection - Backwards Elimination

Before any elimination, I had my predictors as:



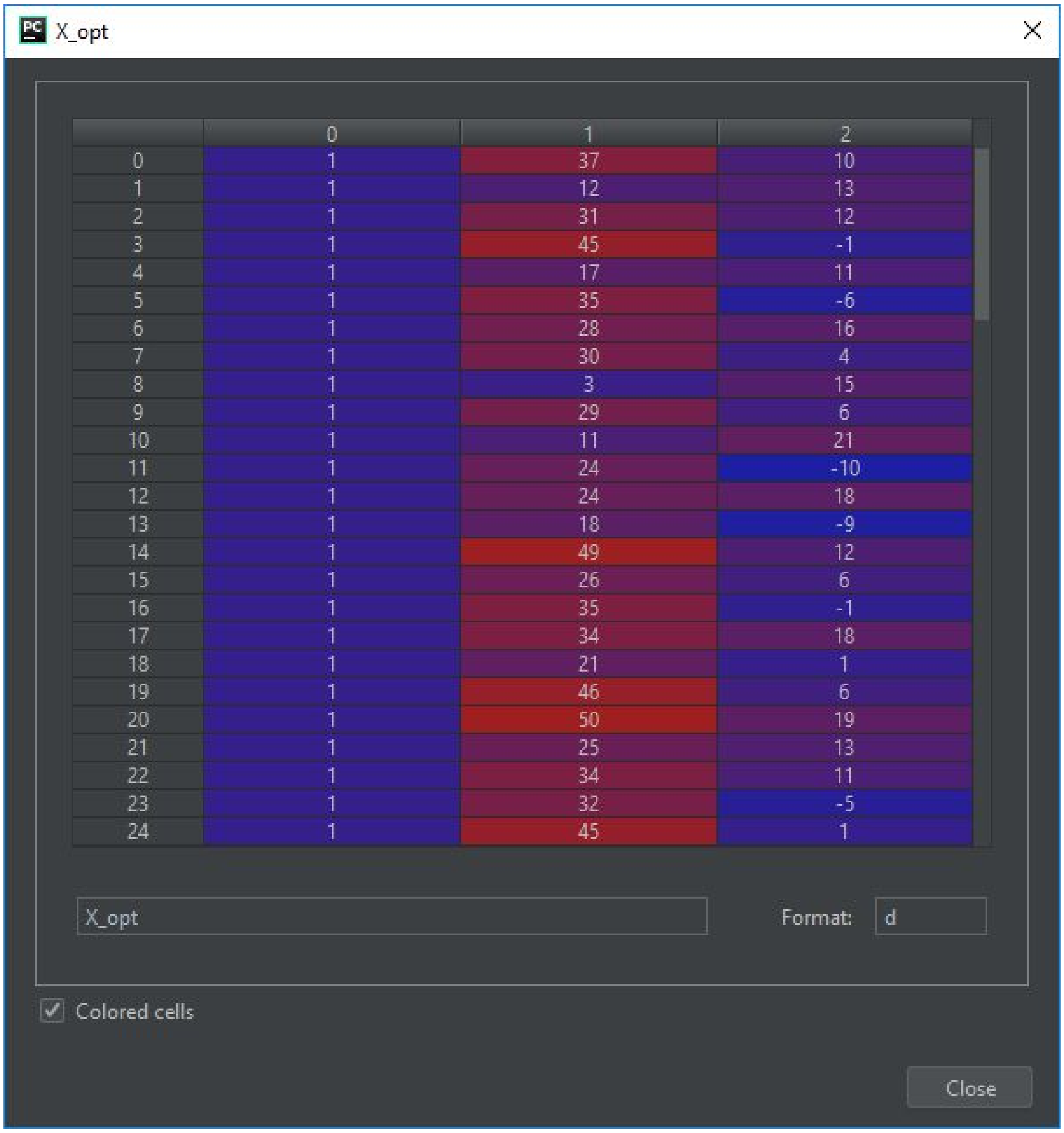
	0	1	2	3	4	5
0	1	32	37	10	41	6
1	1	72	12	13	69	-6
2	1	41	31	12	43	-10
3	1	22	45	-1	95	-8
4	1	90	17	11	72	6
5	1	44	35	-6	82	-7
6	1	21	28	16	41	0
7	1	49	30	4	93	-13
8	1	49	3	15	88	20
9	1	65	29	6	26	-3
10	1	14	11	21	10	4
11	1	41	24	-10	53	-7
12	1	100	24	18	65	-15
13	1	91	18	-9	12	-2
14	1	18	49	12	40	17
15	1	38	26	6	46	2
16	1	10	35	-1	37	-15
17	1	86	34	18	31	16
18	1	58	21	1	50	7
19	1	1	46	6	62	17
20	1	44	50	19	87	18
21	1	79	25	13	17	14
22	1	16	34	11	29	-12
23	1	4	32	-5	60	-6

X_opt

Format: d

☑ Colored cells

After the elimination, my final matrix looked like this:



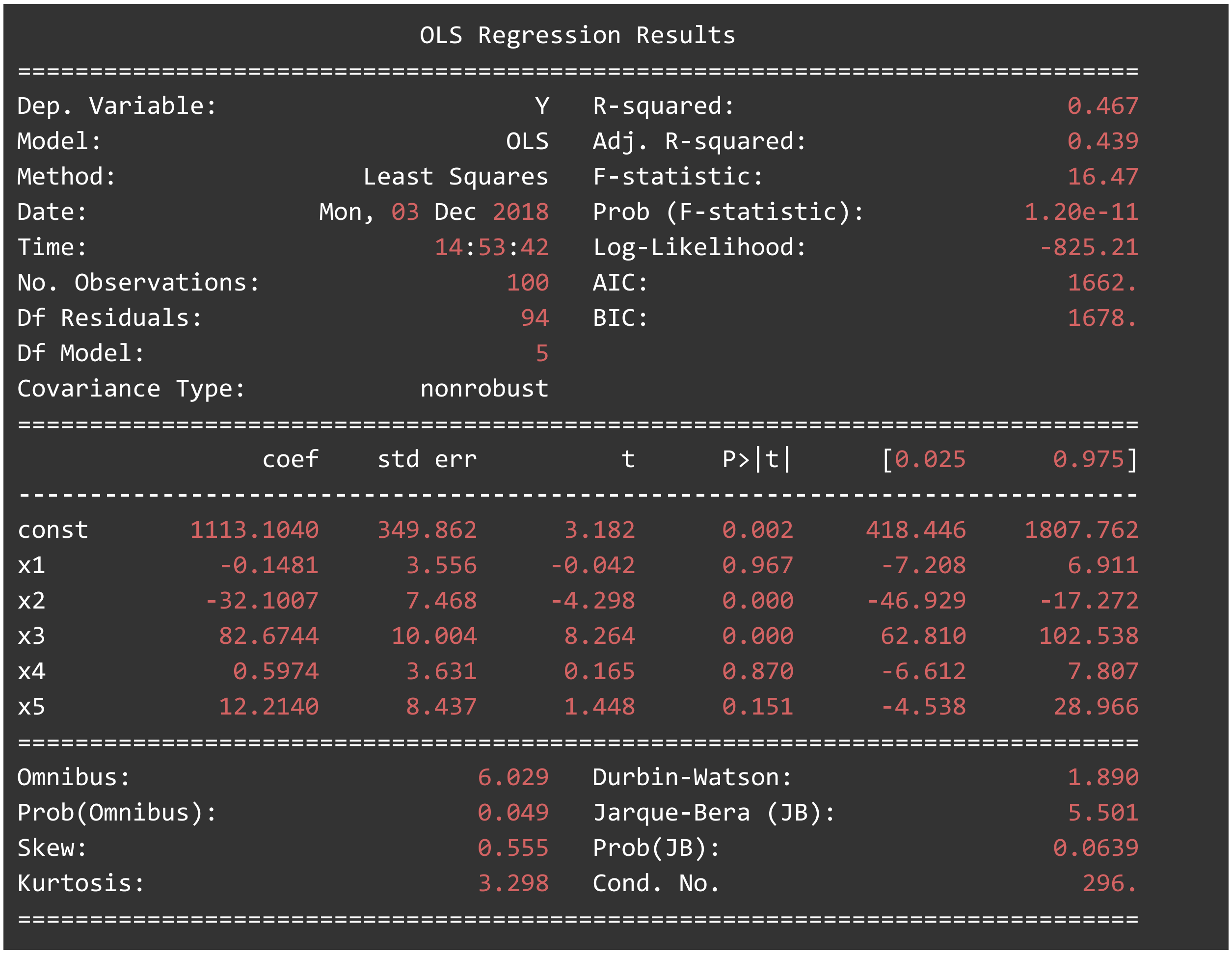
	0	1	2
0	1	37	10
1	1	12	13
2	1	31	12
3	1	45	-1
4	1	17	11
5	1	35	-6
6	1	28	16
7	1	30	4
8	1	3	15
9	1	29	6
10	1	11	21
11	1	24	-10
12	1	24	18
13	1	18	-9
14	1	49	12
15	1	26	6
16	1	35	-1
17	1	34	18
18	1	21	1
19	1	46	6
20	1	50	19
21	1	25	13
22	1	34	11
23	1	32	-5
24	1	45	1

X_opt

Format: d

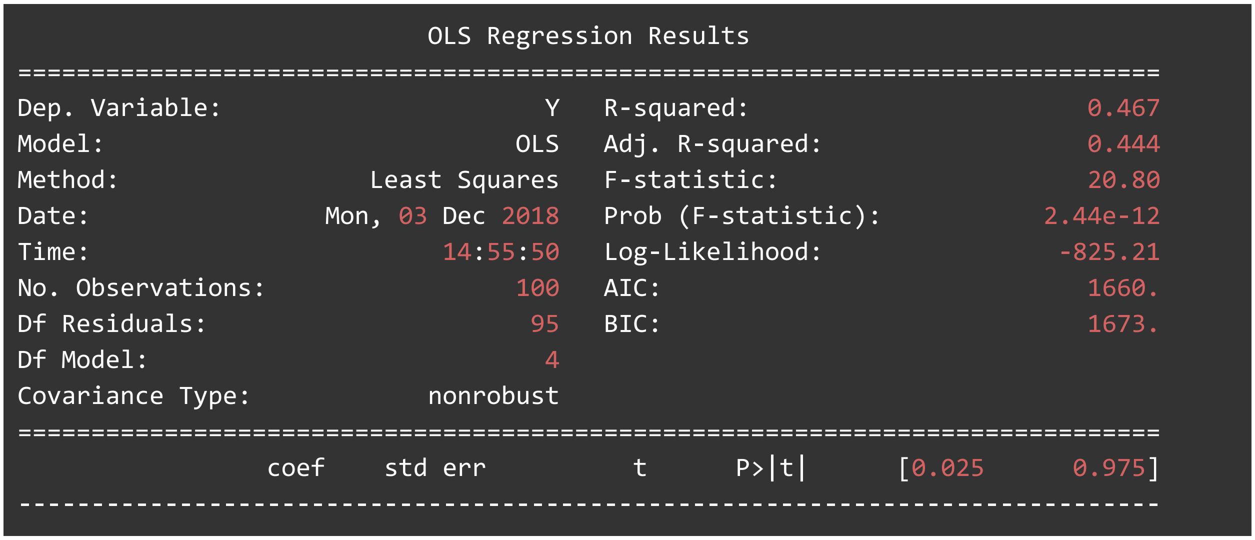
☑ Colored cells

In which the first column was a constant column for OLS resulting. Next page includes step by step results from the OLS table.

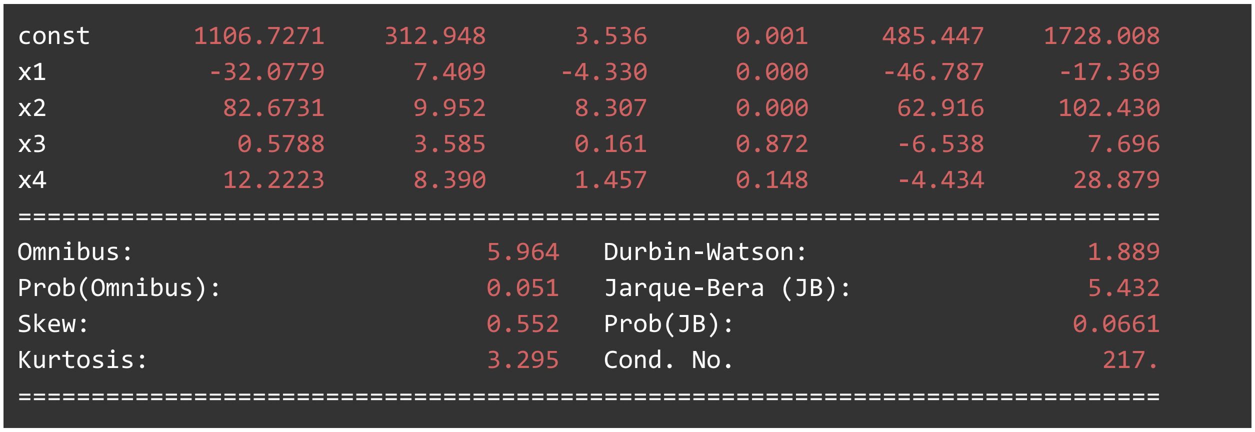


```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.467
Model:                            OLS   Adj. R-squared:                  0.439
Method:                 Least Squares   F-statistic:                     16.47
Date:                Mon, 03 Dec 2018   Prob (F-statistic):           1.20e-11
Time:                        14:53:42   Log-Likelihood:                -825.21
No. Observations:                 100   AIC:                             1662.
Df Residuals:                      94   BIC:                             1678.
Df Model:                           5                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1113.1040    349.862      3.182      0.002     418.446    1807.762
x1            -0.1481      3.556     -0.042      0.967      -7.208       6.911
x2           -32.1007      7.468     -4.298      0.000     -46.929     -17.272
x3            82.6744     10.004      8.264      0.000      62.810     102.538
x4             0.5974      3.631      0.165      0.870      -6.612       7.807
x5            12.2140      8.437      1.448      0.151      -4.538      28.966
==============================================================================
Omnibus:                        6.029   Durbin-Watson:                   1.890
Prob(Omnibus):                  0.049   Jarque-Bera (JB):                5.501
Skew:                           0.555   Prob(JB):                       0.0639
Kurtosis:                       3.298   Cond. No.                         296.
==============================================================================
```

Seeing that the p value for X1 is way too higher than our threshold of 0.05, we eliminate X1 and run the process again.

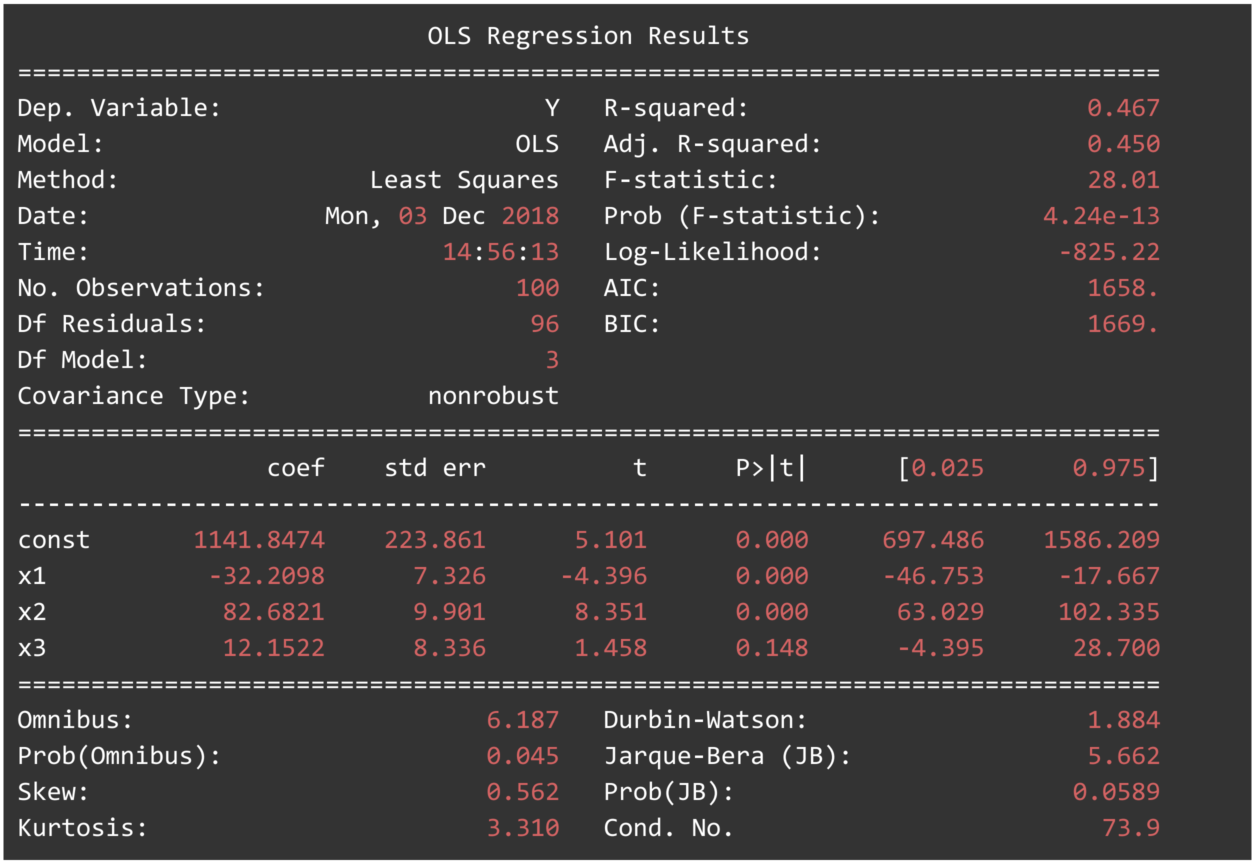


```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.467
Model:                            OLS   Adj. R-squared:                  0.444
Method:                 Least Squares   F-statistic:                     20.80
Date:                Mon, 03 Dec 2018   Prob (F-statistic):           2.44e-12
Time:                        14:55:50   Log-Likelihood:                -825.21
No. Observations:                 100   AIC:                             1660.
Df Residuals:                      95   BIC:                             1673.
Df Model:                           4                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
```



```
const       1106.7271    312.948      3.536      0.001     485.447    1728.008
x1           -32.0779      7.409     -4.330      0.000     -46.787     -17.369
x2            82.6731      9.952      8.307      0.000      62.916     102.430
x3             0.5788      3.585      0.161      0.872      -6.538       7.696
x4            12.2223      8.390      1.457      0.148      -4.434      28.879
==============================================================================
Omnibus:                        5.964   Durbin-Watson:                   1.889
Prob(Omnibus):                  0.051   Jarque-Bera (JB):                5.432
Skew:                           0.552   Prob(JB):                       0.0661
Kurtosis:                       3.295   Cond. No.                         217.
==============================================================================
```

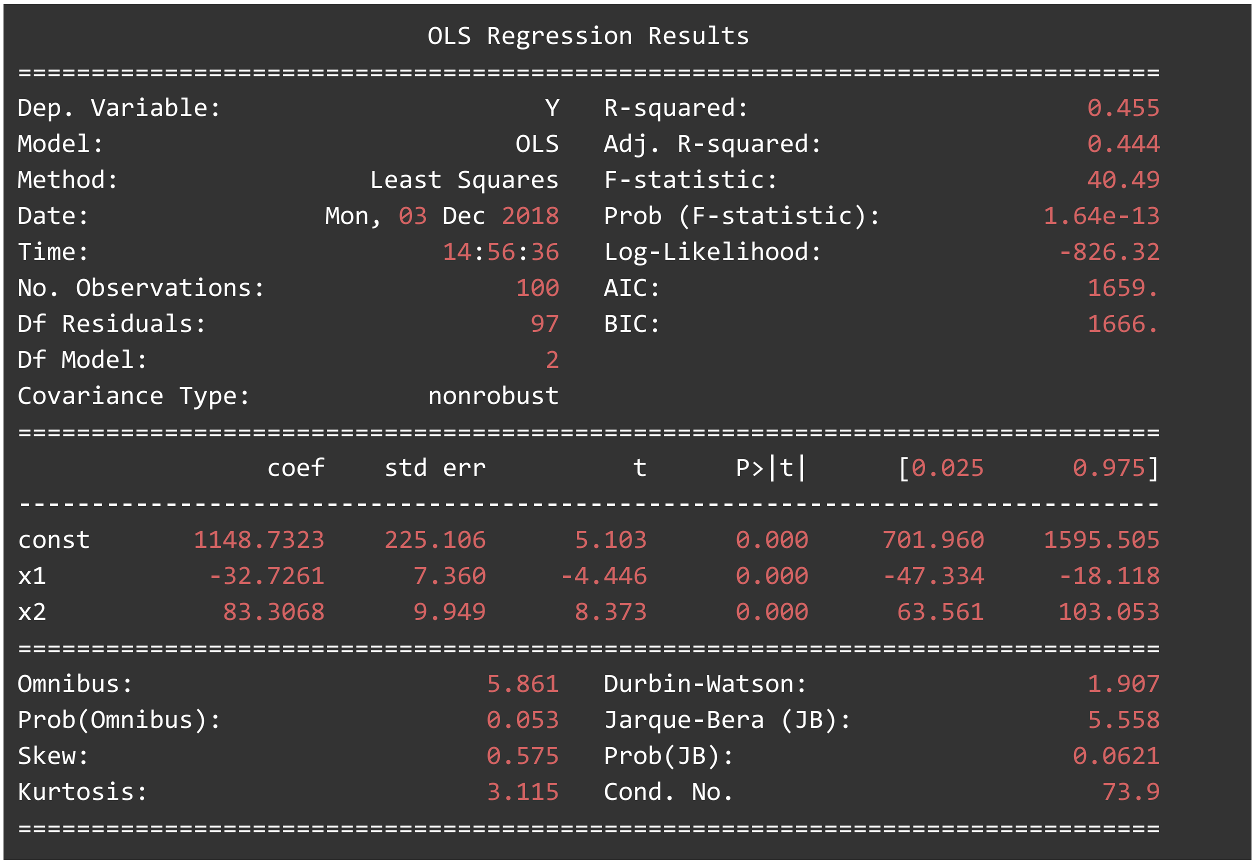
This time, we have our x3 (originally X4) to eliminate. We remove it from the subsets and run again.



```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.467
Model:                            OLS   Adj. R-squared:                  0.450
Method:                 Least Squares   F-statistic:                     28.01
Date:                Mon, 03 Dec 2018   Prob (F-statistic):           4.24e-13
Time:                        14:56:13   Log-Likelihood:                -825.22
No. Observations:                 100   AIC:                             1658.
Df Residuals:                      96   BIC:                             1669.
Df Model:                           3                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1141.8474    223.861      5.101      0.000     697.486    1586.209
x1           -32.2098      7.326     -4.396      0.000     -46.753     -17.667
x2            82.6821      9.901      8.351      0.000      63.029     102.335
x3            12.1522      8.336      1.458      0.148      -4.395      28.700
==============================================================================
Omnibus:                        6.187   Durbin-Watson:                   1.884
Prob(Omnibus):                  0.045   Jarque-Bera (JB):                5.662
Skew:                           0.562   Prob(JB):                       0.0589
Kurtosis:                       3.310   Cond. No.                         73.9
```

Whilst the x1 and x2 (originally X2 and X3) have zeroed out their p-values, we still have x3 (originally X5) to eliminate.

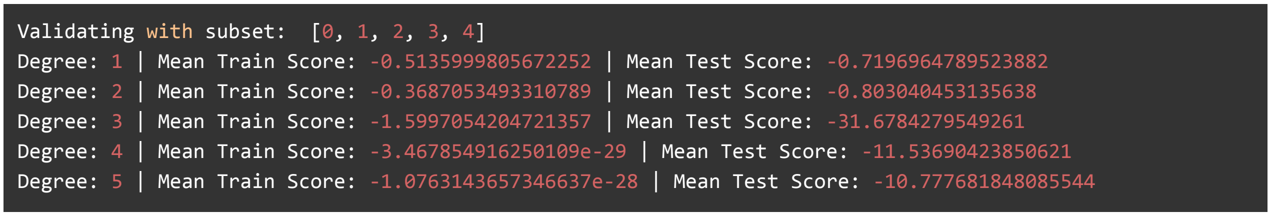
Below is the final OLS table, and the left subset is: [X2,X3]



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.455
Model:                            OLS   Adj. R-squared:                  0.444
Method:                 Least Squares   F-statistic:                     40.49
Date:                Mon, 03 Dec 2018   Prob (F-statistic):           1.64e-13
Time:                        14:56:36   Log-Likelihood:                -826.32
No. Observations:                 100   AIC:                             1659.
Df Residuals:                      97   BIC:                             1666.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1148.7323    225.106      5.103      0.000     701.960    1595.505
x1           -32.7261      7.360     -4.446      0.000     -47.334     -18.118
x2            83.3068      9.949      8.373      0.000      63.561     103.053
==============================================================================
Omnibus:                        5.861   Durbin-Watson:                   1.907
Prob(Omnibus):                  0.053   Jarque-Bera (JB):                5.558
Skew:                           0.575   Prob(JB):                       0.0621
Kurtosis:                       3.115   Cond. No.                         73.9
==============================================================================
```

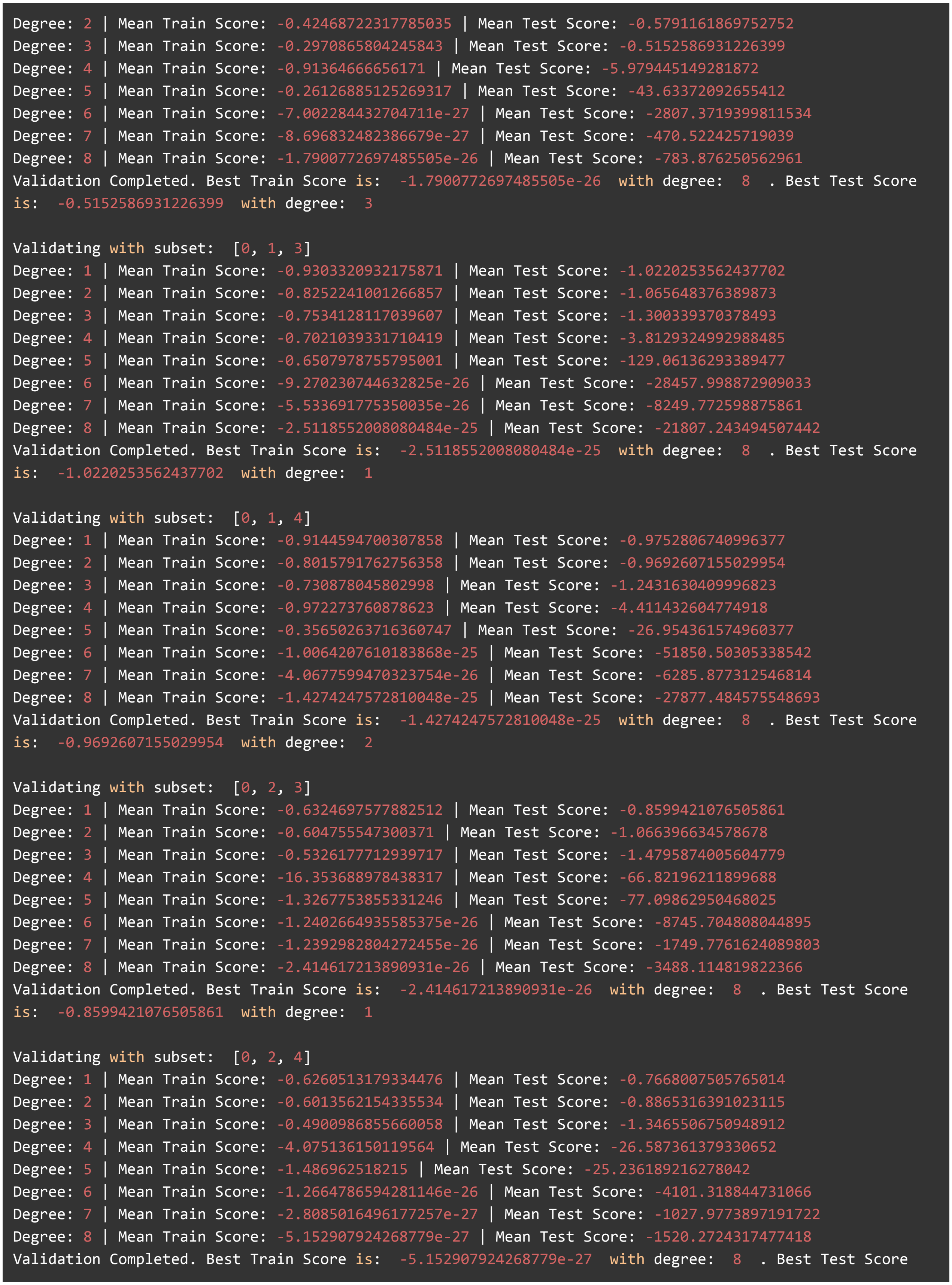
Results 1.5 - Subset Selection - Poly Features Cross Validation

Next, I had run multiple linear regression with polynomial features to have a better insight about the predictors and their weights. I had used k-fold cross validation to measure the scores and combinations of different subsets. Below are the results for all the steps when k = 5.



```
Validating with subset:  [0, 1, 2, 3, 4]
Degree: 1 | Mean Train Score: -0.5135999805672252 | Mean Test Score: -0.7196964789523882
Degree: 2 | Mean Train Score: -0.3687053493310789 | Mean Test Score: -0.803040453135638
Degree: 3 | Mean Train Score: -1.5997054204721357 | Mean Test Score: -31.6784279549261
Degree: 4 | Mean Train Score: -3.467854916250109e-29 | Mean Test Score: -11.53690423850621
Degree: 5 | Mean Train Score: -1.0763143657346637e-28 | Mean Test Score: -10.777681848085544
```

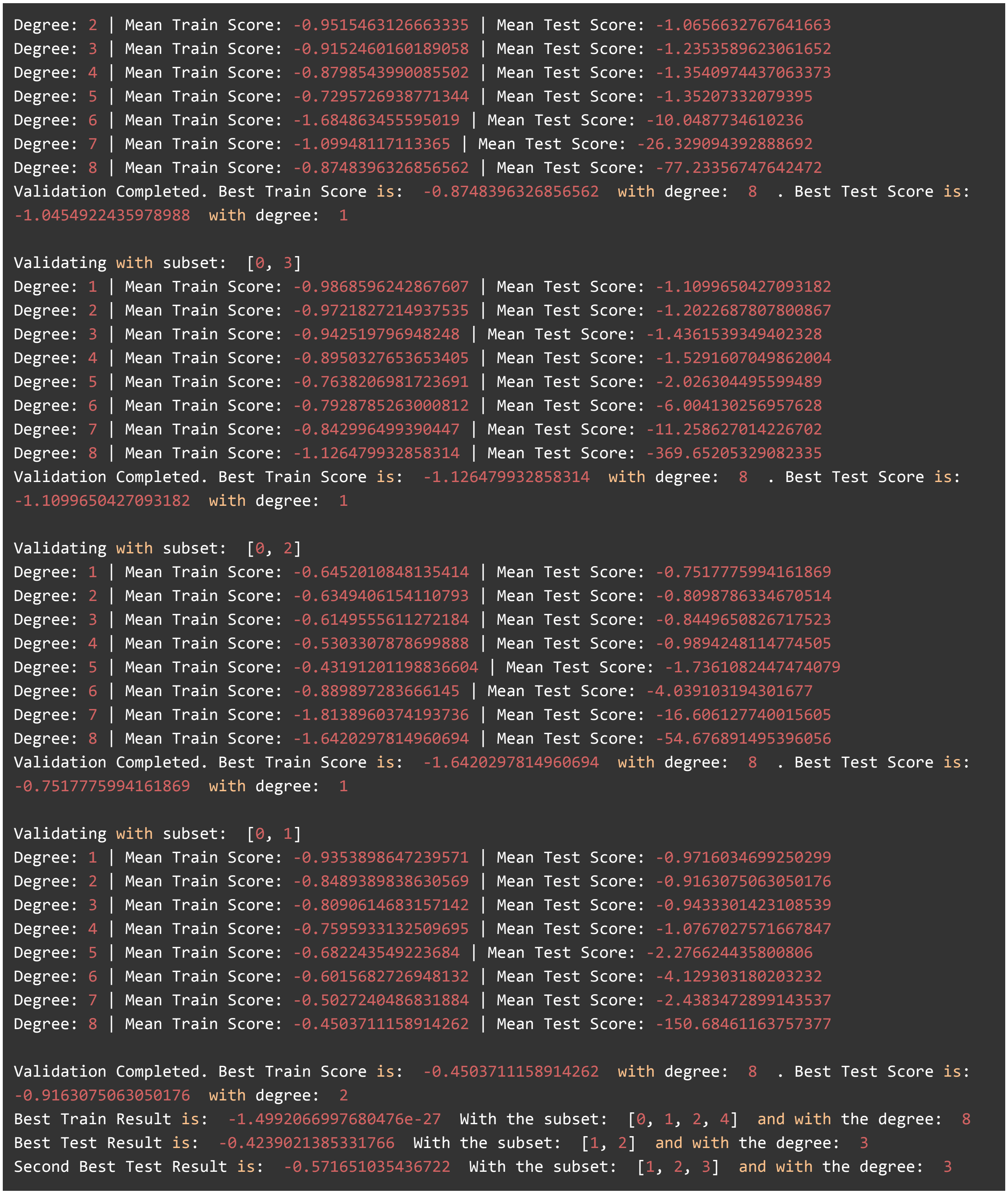
```
Degree: 6 | Mean Train Score: -2.4989986964865995e-28 | Mean Test Score: -27.700631789613475
Degree: 7 | Mean Train Score: -9.06496191037924e-28 | Mean Test Score: -41.92248550375895
Degree: 8 | Mean Train Score: -3.571847362962289e-27 | Mean Test Score: -108.83465609855088
Validation Completed. Best Train Score is:  -3.571847362962289e-27  with degree:  8  . Best Test Score
is:  -0.7196964789523882  with degree:  1

Validating with subset:  [0, 1, 2, 3]
Degree: 1 | Mean Train Score: -0.5289215152486676 | Mean Test Score: -0.7007283202327232
Degree: 2 | Mean Train Score: -0.40053110191735347 | Mean Test Score: -0.6970953704293088
Degree: 3 | Mean Train Score: -2.106847465504754 | Mean Test Score: -9.093631834284132
Degree: 4 | Mean Train Score: -0.1816692272793831 | Mean Test Score: -46.15538798843724
Degree: 5 | Mean Train Score: -1.718375575027834e-28 | Mean Test Score: -45.221636903679205
Degree: 6 | Mean Train Score: -6.504432849305561e-28 | Mean Test Score: -58.32323005411606
Degree: 7 | Mean Train Score: -2.2873429665048758e-27 | Mean Test Score: -172.347563888868
Degree: 8 | Mean Train Score: -8.296479900641847e-27 | Mean Test Score: -251.4826858002652
Validation Completed. Best Train Score is:  -8.296479900641847e-27  with degree:  8  . Best Test Score
is:  -0.6970953704293088  with degree:  2

Validating with subset:  [0, 1, 2, 4]
Degree: 1 | Mean Train Score: -0.5219483222932009 | Mean Test Score: -0.6371359898570456
Degree: 2 | Mean Train Score: -0.40132747501180904 | Mean Test Score: -0.646990783661793
Degree: 3 | Mean Train Score: -1.1920491735896104 | Mean Test Score: -3.6318515081437375
Degree: 4 | Mean Train Score: -0.13715059902455629 | Mean Test Score: -24.746598396882266
Degree: 5 | Mean Train Score: -1.2461830467404889e-28 | Mean Test Score: -23.978944041189656
Degree: 6 | Mean Train Score: -2.334865466151521e-28 | Mean Test Score: -36.53722060128698
Degree: 7 | Mean Train Score: -7.815055380221539e-28 | Mean Test Score: -73.46758492670928
Degree: 8 | Mean Train Score: -1.4992066997680476e-27 | Mean Test Score: -131.34621882432856
Validation Completed. Best Train Score is:  -1.4992066997680476e-27  with degree:  8  . Best Test Score
is:  -0.6371359898570456  with degree:  1

Validating with subset:  [0, 1, 3, 4]
Degree: 1 | Mean Train Score: -0.9090375000816244 | Mean Test Score: -1.0276765822152996
Degree: 2 | Mean Train Score: -0.7703938344638657 | Mean Test Score: -1.2324011347287327
Degree: 3 | Mean Train Score: -0.8445147178672915 | Mean Test Score: -5.448101200588394
Degree: 4 | Mean Train Score: -0.36673430685132236 | Mean Test Score: -136.49522667063735
Degree: 5 | Mean Train Score: -9.988292062837595e-28 | Mean Test Score: -318.81460129212365
Degree: 6 | Mean Train Score: -2.350308148419042e-27 | Mean Test Score: -600.2237301198005
Degree: 7 | Mean Train Score: -8.32156578479847e-27 | Mean Test Score: -1243.4824756375544
Degree: 8 | Mean Train Score: -1.506743792528484e-26 | Mean Test Score: -2633.365870688376
Validation Completed. Best Train Score is:  -1.506743792528484e-26  with degree:  8  . Best Test Score
is:  -1.0276765822152996  with degree:  1

Validating with subset:  [1, 2, 3, 4]
Degree: 1 | Mean Train Score: -0.5154799690749945 | Mean Test Score: -0.697372078483268
Degree: 2 | Mean Train Score: -0.3844803619757638 | Mean Test Score: -0.690313280234782
Degree: 3 | Mean Train Score: -3.9223146581769877 | Mean Test Score: -16.38959301522025
Degree: 4 | Mean Train Score: -0.20008237345411944 | Mean Test Score: -16.460770828589446
Degree: 5 | Mean Train Score: -1.929932734590927e-28 | Mean Test Score: -76.3358302484386
Degree: 6 | Mean Train Score: -3.466784193090115e-28 | Mean Test Score: -61.96575754423547
Degree: 7 | Mean Train Score: -8.145355341245714e-28 | Mean Test Score: -208.65889922823962
Degree: 8 | Mean Train Score: -3.313086029398219e-27 | Mean Test Score: -182.1045728635841
Validation Completed. Best Train Score is:  -3.313086029398219e-27  with degree:  8  . Best Test Score
is:  -0.690313280234782  with degree:  2

Validating with subset:  [0, 1, 2]
Degree: 1 | Mean Train Score: -0.5371015666733647 | Mean Test Score: -0.6186341095814323
```



```
Degree: 2 | Mean Train Score: -0.42468722317785035 | Mean Test Score: -0.5791161869752752
Degree: 3 | Mean Train Score: -0.2970865804245843 | Mean Test Score: -0.5152586931226399
Degree: 4 | Mean Train Score: -0.91364666656171 | Mean Test Score: -5.979445149281872
Degree: 5 | Mean Train Score: -0.26126885125269317 | Mean Test Score: -43.63372092655412
Degree: 6 | Mean Train Score: -7.002284432704711e-27 | Mean Test Score: -2807.3719399811534
Degree: 7 | Mean Train Score: -8.696832482386679e-27 | Mean Test Score: -470.522425719039
Degree: 8 | Mean Train Score: -1.7900772697485505e-26 | Mean Test Score: -783.876250562961
Validation Completed. Best Train Score is:  -1.7900772697485505e-26  with degree:  8  . Best Test Score
is:  -0.5152586931226399  with degree:  3

Validating with subset:  [0, 1, 3]
Degree: 1 | Mean Train Score: -0.9303320932175871 | Mean Test Score: -1.0220253562437702
Degree: 2 | Mean Train Score: -0.8252241001266857 | Mean Test Score: -1.065648376389873
Degree: 3 | Mean Train Score: -0.7534128117039607 | Mean Test Score: -1.300339370378493
Degree: 4 | Mean Train Score: -0.7021039331710419 | Mean Test Score: -3.8129324992988485
Degree: 5 | Mean Train Score: -0.6507978755795001 | Mean Test Score: -129.06136293389477
Degree: 6 | Mean Train Score: -9.270230744632825e-26 | Mean Test Score: -28457.998872909033
Degree: 7 | Mean Train Score: -5.533691775350035e-26 | Mean Test Score: -8249.772598875861
Degree: 8 | Mean Train Score: -2.5118552008080484e-25 | Mean Test Score: -21807.243494507442
Validation Completed. Best Train Score is:  -2.5118552008080484e-25  with degree:  8  . Best Test Score
is:  -1.0220253562437702  with degree:  1

Validating with subset:  [0, 1, 4]
Degree: 1 | Mean Train Score: -0.9144594700307858 | Mean Test Score: -0.9752806740996377
Degree: 2 | Mean Train Score: -0.8015791762756358 | Mean Test Score: -0.9692607155029954
Degree: 3 | Mean Train Score: -0.730878045802998 | Mean Test Score: -1.2431630409996823
Degree: 4 | Mean Train Score: -0.972273760878623 | Mean Test Score: -4.411432604774918
Degree: 5 | Mean Train Score: -0.35650263716360747 | Mean Test Score: -26.954361574960377
Degree: 6 | Mean Train Score: -1.0064207610183868e-25 | Mean Test Score: -51850.50305338542
Degree: 7 | Mean Train Score: -4.0677599470323754e-26 | Mean Test Score: -6285.877312546814
Degree: 8 | Mean Train Score: -1.4274247572810048e-25 | Mean Test Score: -27877.484575548693
Validation Completed. Best Train Score is:  -1.4274247572810048e-25  with degree:  8  . Best Test Score
is:  -0.9692607155029954  with degree:  2

Validating with subset:  [0, 2, 3]
Degree: 1 | Mean Train Score: -0.6324697577882512 | Mean Test Score: -0.8599421076505861
Degree: 2 | Mean Train Score: -0.604755547300371 | Mean Test Score: -1.066396634578678
Degree: 3 | Mean Train Score: -0.5326177712939717 | Mean Test Score: -1.4795874005604779
Degree: 4 | Mean Train Score: -16.353688978438317 | Mean Test Score: -66.82196211899688
Degree: 5 | Mean Train Score: -1.3267753855331246 | Mean Test Score: -77.09862950468025
Degree: 6 | Mean Train Score: -1.2402664935585375e-26 | Mean Test Score: -8745.704808044895
Degree: 7 | Mean Train Score: -1.2392982804272455e-26 | Mean Test Score: -1749.7761624089803
Degree: 8 | Mean Train Score: -2.414617213890931e-26 | Mean Test Score: -3488.114819822366
Validation Completed. Best Train Score is:  -2.414617213890931e-26  with degree:  8  . Best Test Score
is:  -0.8599421076505861  with degree:  1

Validating with subset:  [0, 2, 4]
Degree: 1 | Mean Train Score: -0.6260513179334476 | Mean Test Score: -0.7668007505765014
Degree: 2 | Mean Train Score: -0.6013562154335534 | Mean Test Score: -0.8865316391023115
Degree: 3 | Mean Train Score: -0.4900986855660058 | Mean Test Score: -1.3465506750948912
Degree: 4 | Mean Train Score: -4.075136150119564 | Mean Test Score: -26.587361379330652
Degree: 5 | Mean Train Score: -1.486962518215 | Mean Test Score: -25.236189216278042
Degree: 6 | Mean Train Score: -1.2664786594281146e-26 | Mean Test Score: -4101.318844731066
Degree: 7 | Mean Train Score: -2.8085016496177257e-27 | Mean Test Score: -1027.9773897191722
Degree: 8 | Mean Train Score: -5.152907924268779e-27 | Mean Test Score: -1520.2724317477418
Validation Completed. Best Train Score is:  -5.152907924268779e-27  with degree:  8  . Best Test Score
```

```
is:  -0.7668007505765014  with degree:  1

Validating with subset:  [1, 2, 3]
Degree: 1 | Mean Train Score: -0.5309059824910606 | Mean Test Score: -0.6776840933131304
Degree: 2 | Mean Train Score: -0.4104464872218666 | Mean Test Score: -0.6175790673268974
Degree: 3 | Mean Train Score: -0.28670830148640325 | Mean Test Score: -0.571651035436722
Degree: 4 | Mean Train Score: -0.379934623441685 | Mean Test Score: -1.7202932543155751
Degree: 5 | Mean Train Score: -0.14124151638766755 | Mean Test Score: -5.532452931130908
Degree: 6 | Mean Train Score: -6.080691216855667e-26 | Mean Test Score: -39823.51574993976
Degree: 7 | Mean Train Score: -1.2819875619801009e-26 | Mean Test Score: -1242.602632331565
Degree: 8 | Mean Train Score: -6.046909078617235e-26 | Mean Test Score: -3601.3708452043056
Validation Completed. Best Train Score is:  -6.046909078617235e-26  with degree:  8  . Best Test Score
is:  -0.571651035436722  with degree:  3

Validating with subset:  [1, 2, 4]
Degree: 1 | Mean Train Score: -0.522855902671014 | Mean Test Score: -0.6273899938810343
Degree: 2 | Mean Train Score: -0.41058629101633654 | Mean Test Score: -0.6051054059757976
Degree: 3 | Mean Train Score: -0.264856684001356 | Mean Test Score: -0.7504601942091519
Degree: 4 | Mean Train Score: -0.5783411218713563 | Mean Test Score: -2.426021542872602
Degree: 5 | Mean Train Score: -0.11861706064605096 | Mean Test Score: -10.054433737687685
Degree: 6 | Mean Train Score: -9.951924610697487e-27 | Mean Test Score: -2839.2697826446133
Degree: 7 | Mean Train Score: -2.3771519813454842e-27 | Mean Test Score: -1075.8567207771105
Degree: 8 | Mean Train Score: -4.964793360211357e-27 | Mean Test Score: -424.30915247559795
Validation Completed. Best Train Score is:  -4.964793360211357e-27  with degree:  8  . Best Test Score
is:  -0.6051054059757976  with degree:  2

Validating with subset:  [1, 3, 4]
Degree: 1 | Mean Train Score: -0.9113898470007002 | Mean Test Score: -1.0032033384361818
Degree: 2 | Mean Train Score: -0.8031078648228691 | Mean Test Score: -1.0171631692258594
Degree: 3 | Mean Train Score: -0.7212017375648673 | Mean Test Score: -1.205199869715814
Degree: 4 | Mean Train Score: -1.2620766220513062 | Mean Test Score: -4.649431495252136
Degree: 5 | Mean Train Score: -0.5493231077763977 | Mean Test Score: -74.51221397970653
Degree: 6 | Mean Train Score: -1.631980821627393e-25 | Mean Test Score: -180107.68535961973
Degree: 7 | Mean Train Score: -7.137038937466874e-26 | Mean Test Score: -6608.697197404736
Degree: 8 | Mean Train Score: -3.1039547973231315e-25 | Mean Test Score: -12450.499472079087
Validation Completed. Best Train Score is:  -3.1039547973231315e-25  with degree:  8  . Best Test Score
is:  -1.0032033384361818  with degree:  1

Validating with subset:  [2, 3, 4]
Degree: 1 | Mean Train Score: -0.6171377965766236 | Mean Test Score: -0.8324617664375664
Degree: 2 | Mean Train Score: -0.5893885807601867 | Mean Test Score: -1.0468350794501249
Degree: 3 | Mean Train Score: -0.5099816247019348 | Mean Test Score: -1.6789185639733681
Degree: 4 | Mean Train Score: -6.29789189913608 | Mean Test Score: -29.132513751095225
Degree: 5 | Mean Train Score: -2.475541941329008 | Mean Test Score: -75.47798169857703
Degree: 6 | Mean Train Score: -1.432545648420211e-26 | Mean Test Score: -5692.137450162127
Degree: 7 | Mean Train Score: -6.234902958078404e-27 | Mean Test Score: -1125.9736812020265
Degree: 8 | Mean Train Score: -1.475648380151473e-26 | Mean Test Score: -2250.265837919109
Validation Completed. Best Train Score is:  -1.475648380151473e-26  with degree:  8  . Best Test Score
is:  -0.8324617664375664  with degree:  1

Validating with subset:  [3, 4]
Degree: 1 | Mean Train Score: -0.9657488272328424 | Mean Test Score: -1.0783886998679673
Degree: 2 | Mean Train Score: -0.9561905776093317 | Mean Test Score: -1.1335528454695694
Degree: 3 | Mean Train Score: -0.9313866298235173 | Mean Test Score: -1.263668410619291
Degree: 4 | Mean Train Score: -0.8980064255630019 | Mean Test Score: -1.4380653263945575
Degree: 5 | Mean Train Score: -0.7592090285210247 | Mean Test Score: -3.1574845381711025
```

```
Degree: 6 | Mean Train Score: -1.1020514587865828 | Mean Test Score: -9.579428682889539
Degree: 7 | Mean Train Score: -0.9230707489306418 | Mean Test Score: -26.811749514241065
Degree: 8 | Mean Train Score: -0.7068083308049778 | Mean Test Score: -297.82962757965595
Validation Completed. Best Train Score is:  -0.7068083308049778  with degree:  8  . Best Test Score is:
-1.0783886998679673  with degree:  1

Validating with subset:  [2, 4]
Degree: 1 | Mean Train Score: -0.6288644307047508 | Mean Test Score: -0.746350336097728
Degree: 2 | Mean Train Score: -0.6161238122963868 | Mean Test Score: -0.853134278493188
Degree: 3 | Mean Train Score: -0.5695874842845582 | Mean Test Score: -1.1665273674030263
Degree: 4 | Mean Train Score: -0.5141730064885406 | Mean Test Score: -1.6085221163849965
Degree: 5 | Mean Train Score: -0.4386354155620745 | Mean Test Score: -2.7175228375274183
Degree: 6 | Mean Train Score: -1.6089831133739445 | Mean Test Score: -14.92752975412472
Degree: 7 | Mean Train Score: -5.989946244701366 | Mean Test Score: -51.046088437133626
Degree: 8 | Mean Train Score: -1.5607888307307032 | Mean Test Score: -111.14230368732203
Validation Completed. Best Train Score is:  -1.5607888307307032  with degree:  8  . Best Test Score is:
-0.746350336097728  with degree:  1

Validating with subset:  [2, 3]
Degree: 1 | Mean Train Score: -0.6368593364743058 | Mean Test Score: -0.8162421165192983
Degree: 2 | Mean Train Score: -0.622506771311999 | Mean Test Score: -0.9208661342716906
Degree: 3 | Mean Train Score: -0.5890465782812735 | Mean Test Score: -0.9055193887784883
Degree: 4 | Mean Train Score: -0.5473091072814203 | Mean Test Score: -1.1000419886462436
Degree: 5 | Mean Train Score: -0.4802526443713007 | Mean Test Score: -1.6289242736648457
Degree: 6 | Mean Train Score: -1.0273006403340603 | Mean Test Score: -4.218659570783769
Degree: 7 | Mean Train Score: -2.0651628606338663 | Mean Test Score: -7.958454493455305
Degree: 8 | Mean Train Score: -3.060864170880059 | Mean Test Score: -171.8832929075524
Validation Completed. Best Train Score is:  -3.060864170880059  with degree:  8  . Best Test Score is:
-0.8162421165192983  with degree:  1

Validating with subset:  [1, 4]
Degree: 1 | Mean Train Score: -0.9161081380995807 | Mean Test Score: -0.9596992788216306
Degree: 2 | Mean Train Score: -0.8178167115335032 | Mean Test Score: -0.9142497345808008
Degree: 3 | Mean Train Score: -0.7715737680520622 | Mean Test Score: -1.008942284663872
Degree: 4 | Mean Train Score: -0.7149664616014436 | Mean Test Score: -0.9182129970905721
Degree: 5 | Mean Train Score: -0.6734558805722335 | Mean Test Score: -1.1413393620576753
Degree: 6 | Mean Train Score: -0.6219579268180634 | Mean Test Score: -3.073952346311185
Degree: 7 | Mean Train Score: -0.5671765926612864 | Mean Test Score: -15.216877874104231
Degree: 8 | Mean Train Score: -1.1351747500050737 | Mean Test Score: -102.89560684972503
Validation Completed. Best Train Score is:  -1.1351747500050737  with degree:  8  . Best Test Score is:
-0.9142497345808008  with degree:  2

Validating with subset:  [1, 2]
Degree: 1 | Mean Train Score: -0.5380720319303715 | Mean Test Score: -0.6086388651307335
Degree: 2 | Mean Train Score: -0.43160878706521977 | Mean Test Score: -0.5522373189153529
Degree: 3 | Mean Train Score: -0.3233241704618768 | Mean Test Score: -0.4239021385331766
Degree: 4 | Mean Train Score: -0.265836816479556 | Mean Test Score: -0.49397720756429314
Degree: 5 | Mean Train Score: -0.19324777390239123 | Mean Test Score: -0.4673439491003082
Degree: 6 | Mean Train Score: -0.16802813399588076 | Mean Test Score: -1.6846603219971243
Degree: 7 | Mean Train Score: -0.10261078413450156 | Mean Test Score: -13.86838533230593
Degree: 8 | Mean Train Score: -0.08470408613085231 | Mean Test Score: -18.568987035148098
Validation Completed. Best Train Score is:  -0.08470408613085231  with degree:  8  . Best Test Score is:
-0.4239021385331766  with degree:  3

Validating with subset:  [0, 4]
Degree: 1 | Mean Train Score: -0.9707291343981082 | Mean Test Score: -1.0454922435978988
```



```
Degree: 2 | Mean Train Score: -0.9515463126663335 | Mean Test Score: -1.0656632767641663
Degree: 3 | Mean Train Score: -0.9152460160189058 | Mean Test Score: -1.2353589623061652
Degree: 4 | Mean Train Score: -0.8798543990085502 | Mean Test Score: -1.3540974437063373
Degree: 5 | Mean Train Score: -0.7295726938771344 | Mean Test Score: -1.35207332079395
Degree: 6 | Mean Train Score: -1.684863455595019 | Mean Test Score: -10.0487734610236
Degree: 7 | Mean Train Score: -1.09948117113365 | Mean Test Score: -26.329094392888692
Degree: 8 | Mean Train Score: -0.8748396326856562 | Mean Test Score: -77.23356747642472
Validation Completed. Best Train Score is:  -0.8748396326856562  with degree:  8  . Best Test Score is:
-1.0454922435978988  with degree:  1

Validating with subset:  [0, 3]
Degree: 1 | Mean Train Score: -0.9868596242867607 | Mean Test Score: -1.1099650427093182
Degree: 2 | Mean Train Score: -0.9721827214937535 | Mean Test Score: -1.2022687807800867
Degree: 3 | Mean Train Score: -0.942519796948248 | Mean Test Score: -1.4361539349402328
Degree: 4 | Mean Train Score: -0.8950327653653405 | Mean Test Score: -1.5291607049862004
Degree: 5 | Mean Train Score: -0.7638206981723691 | Mean Test Score: -2.026304495599489
Degree: 6 | Mean Train Score: -0.7928785263000812 | Mean Test Score: -6.004130256957628
Degree: 7 | Mean Train Score: -0.842996499390447 | Mean Test Score: -11.258627014226702
Degree: 8 | Mean Train Score: -1.126479932858314 | Mean Test Score: -369.65205329082335
Validation Completed. Best Train Score is:  -1.126479932858314  with degree:  8  . Best Test Score is:
-1.1099650427093182  with degree:  1

Validating with subset:  [0, 2]
Degree: 1 | Mean Train Score: -0.6452010848135414 | Mean Test Score: -0.7517775994161869
Degree: 2 | Mean Train Score: -0.6349406154110793 | Mean Test Score: -0.8098786334670514
Degree: 3 | Mean Train Score: -0.6149555611272184 | Mean Test Score: -0.8449650826717523
Degree: 4 | Mean Train Score: -0.5303307878699888 | Mean Test Score: -0.9894248114774505
Degree: 5 | Mean Train Score: -0.43191201198836604 | Mean Test Score: -1.7361082447474079
Degree: 6 | Mean Train Score: -0.889897283666145 | Mean Test Score: -4.039103194301677
Degree: 7 | Mean Train Score: -1.8138960374193736 | Mean Test Score: -16.606127740015605
Degree: 8 | Mean Train Score: -1.6420297814960694 | Mean Test Score: -54.676891495396056
Validation Completed. Best Train Score is:  -1.6420297814960694  with degree:  8  . Best Test Score is:
-0.7517775994161869  with degree:  1

Validating with subset:  [0, 1]
Degree: 1 | Mean Train Score: -0.9353898647239571 | Mean Test Score: -0.9716034699250299
Degree: 2 | Mean Train Score: -0.8489389838630569 | Mean Test Score: -0.9163075063050176
Degree: 3 | Mean Train Score: -0.8090614683157142 | Mean Test Score: -0.9433301423108539
Degree: 4 | Mean Train Score: -0.7595933132509695 | Mean Test Score: -1.0767027571667847
Degree: 5 | Mean Train Score: -0.682243549223684 | Mean Test Score: -2.276624435800806
Degree: 6 | Mean Train Score: -0.6015682726948132 | Mean Test Score: -4.129303180203232
Degree: 7 | Mean Train Score: -0.5027240486831884 | Mean Test Score: -2.4383472899143537
Degree: 8 | Mean Train Score: -0.4503711158914262 | Mean Test Score: -150.68461163757377

Validation Completed. Best Train Score is:  -0.4503711158914262  with degree:  8  . Best Test Score is:
-0.9163075063050176  with degree:  2
Best Train Result is:  -1.4992066997680476e-27  With the subset:  [0, 1, 2, 4]  and with the degree:  8
Best Test Result is:  -0.4239021385331766  With the subset:  [1, 2]  and with the degree:  3
Second Best Test Result is:  -0.571651035436722  With the subset:  [1, 2, 3]  and with the degree:  3
```

For the sake of leaving the report clean, I have not included other folds in here. The results for other folds can be found inside the .txt files, which can be found under the results folder inside the provided files. According to all the folds, the subset [X2,X3] with the polynomial degree 3 had the best test results for all k values. For the second place, k value of 10 and 20 showed that the subset [X2,X3,X5] with the degree 3 was better, while k value of 5 showed that the subset [X2,X3,X4] with the degree 2 was the second winner.

Results 1.6 - Lasso and Ridge CV

The results for Lasso CV for different subsets were:

```
Predictors: X2 and X3
Degree: 3
Mean Train Score: -0.330968774541359O3  Mean Test Score: -0.4338286227721304

Predictors: X2, X3 and X4
Degree: 2
Mean Train Score: -0.4543572983110679  Mean Test Score: -0.5366836654190827

Predictors: X2, X3 and X5
Degree: 3
Mean Train Score: -0.32989314414319254  Mean Test Score: -0.5040878596296313
```

Here it can be seen that the subset [X2,X3] with polynomial degree of 3 was again once more the clear winner with the test score. The second place winner was the subset [X2,X3,X5] with the degree of 3, while the subset [X2, X3, X4] had the biggest test score amongst three.

Below are the results for Ridge CV:

```
Predictors: X2 and X3
Degree: 3
Mean Train Score: -0.32893774864422076 | Mean Test Score: -0.4256606715643283

Predictors: X2, X3 and X4
Degree: 2
Mean Train Score: -0.4285054279376377 | Mean Test Score: -0.579948431774464

Predictors: X2, X3 and X5
```

```
Degree: 3
Mean Train Score: -0.3108737978845267 | Mean Test Score: -0.5221750650944521
```

Again, just like the Lasso CV, the subset [X2, X3] with the degree of 3 had the best test score. However with the Ridge CV, the subset [X2, X3, X5] got into the second place unlike the fact that it was 3rd in Lasso CV tests. To summarize the results, we can say that the top results are:

- [X2,X3] with degree 3, Ridge CV
- [X2, X3] with degree 3, Lasso CV
- [X2, X3, X5] with degree 3, Lasso CV
- [X2, X3, X5] with degree 3, Ridge CV
- [X2, X3, X4] with degree 2, Lasso CV
- [X2, X3, X4] with degree 2, Ridge CV

Results 1.7 - SVM

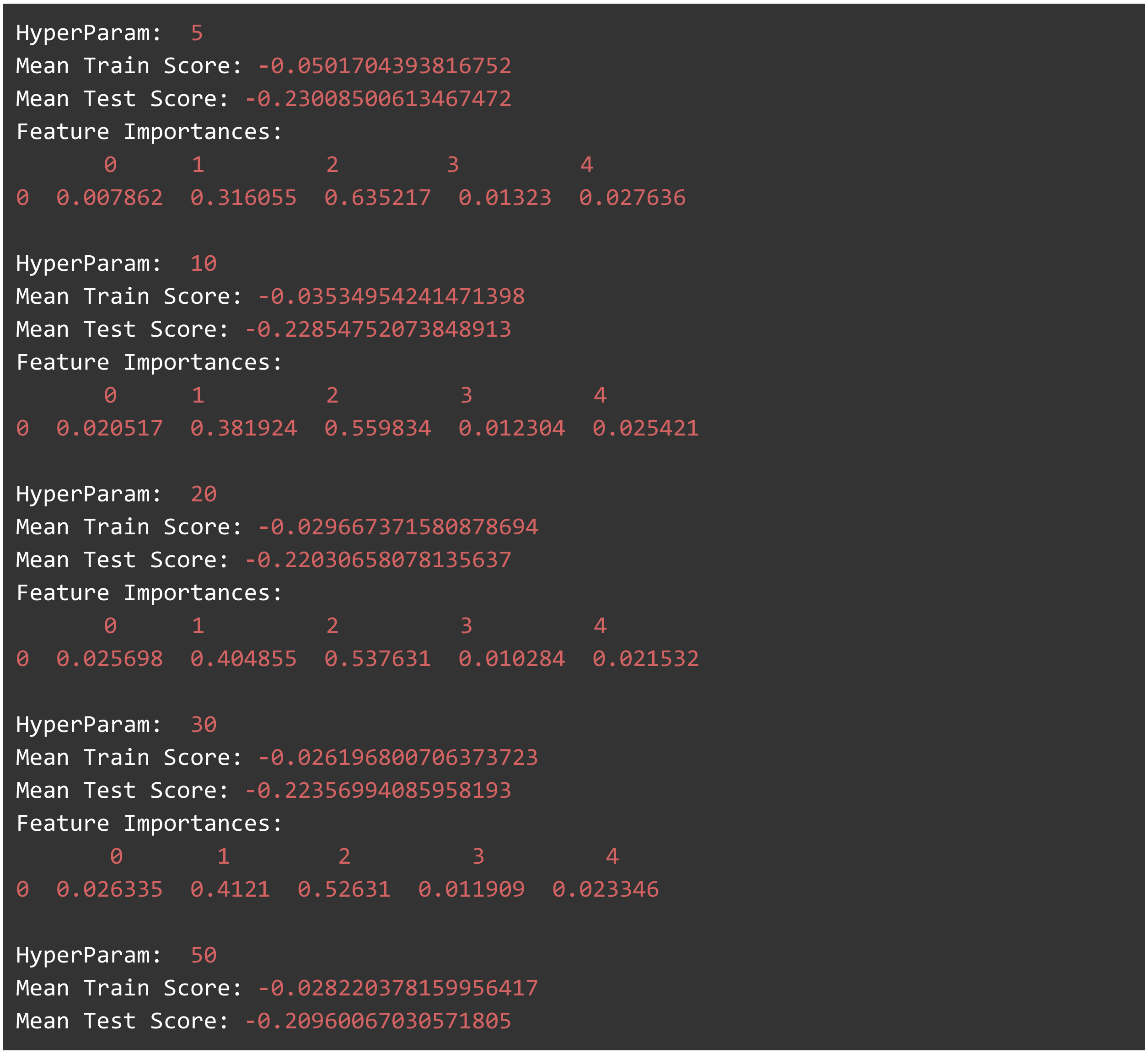
Moving beyond linearity, I had my tests with various machines as mentioned in the Methodology section. The results obtained by Support Vector Machine are below:

```
Subset: [X2, X3]
Mean Train Score: -0.26112950014608055 | Mean Test Score: -0.401293732381754

Subset: [X2, X3, X4]
Mean Train Score: -0.3175379715926245 | Mean Test Score: -0.48645663101828573

Subset: [X2, X3, X5]
Mean Train Score: -0.24703315691451982 | Mean Test Score: -0.42441381260931577
```

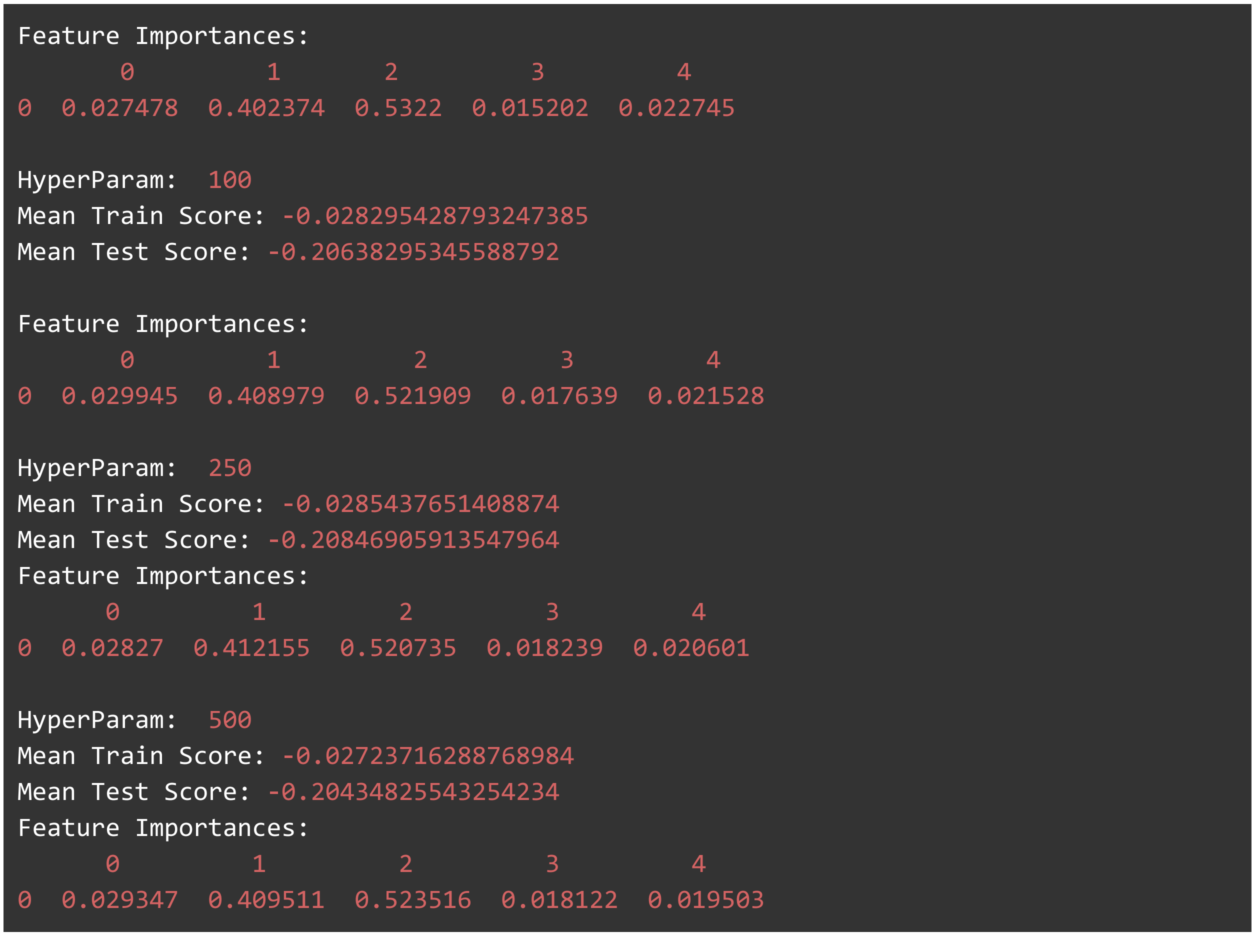
As can be seen here, the subset [X2, X3] performed the best result again, while the subset [X2, X3, X5] was the second place, in the means of test scores. One more important point is that, with SVM, I had achieved the best test results so far which is -0.40129~, better than those I had achieved with polynomial linear regression and shrinkage methods.

Results 1.8 - Random Forest Scores

Even though I was pretty much convinced to use the subset [X2, X3] after I had seen the results in SVM; I still wanted to check the predictors and their weights using the feature importance in random forest package. So I had run some tests with different trees of [5,10,20,30,50,100,250,500]. Below are the results.



```
HyperParam:  5
Mean Train Score: -0.0501704393816752
Mean Test Score: -0.23008500613467472
Feature Importances:
       0      1         2        3         4
0  0.007862  0.316055  0.635217  0.01323  0.027636

HyperParam:  10
Mean Train Score: -0.03534954241471398
Mean Test Score: -0.22854752073848913
Feature Importances:
       0      1         2         3         4
0  0.020517  0.381924  0.559834  0.012304  0.025421

HyperParam:  20
Mean Train Score: -0.029667371580878694
Mean Test Score: -0.22030658078135637
Feature Importances:
       0      1         2         3         4
0  0.025698  0.404855  0.537631  0.010284  0.021532

HyperParam:  30
Mean Train Score: -0.026196800706373723
Mean Test Score: -0.22356994085958193
Feature Importances:
        0       1        2         3         4
0  0.026335  0.4121  0.52631  0.011909  0.023346

HyperParam:  50
Mean Train Score: -0.028220378159956417
Mean Test Score: -0.20960067030571805
```

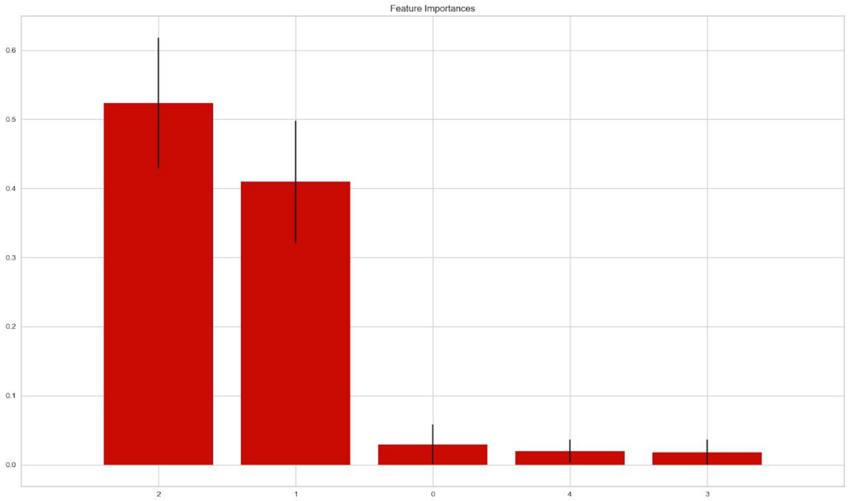


```
Feature Importances:
          0         1        2         3         4
0  0.027478  0.402374  0.5322  0.015202  0.022745

HyperParam:  100
Mean Train Score: -0.028295428793247385
Mean Test Score: -0.20638295345588792

Feature Importances:
          0         1         2         3         4
0  0.029945  0.408979  0.521909  0.017639  0.021528

HyperParam:  250
Mean Train Score: -0.0285437651408874
Mean Test Score: -0.20846905913547964
Feature Importances:
         0         1         2         3         4
0  0.02827  0.412155  0.520735  0.018239  0.020601

HyperParam:  500
Mean Train Score: -0.027237162887689 84
Mean Test Score: -0.20434825543254234
Feature Importances:
         0         1         2         3         4
0  0.029347  0.409511  0.523516  0.018122  0.019503
```

As can be seen above, the best test result came with the estimator count of 500, which had the mean test score of -0.20434825543254234. However, I did not want to choose 500 estimators as it was relatively a big number, so I had the hunch of choosing 100, which was the second best with the mean score of -0.20638295345588792.

Results 1.9 - Random Forest Features

Above we can see the feature importances of predictors, for the sake of simplicity, I had plotted those importances.



And from the plot and the results, we can see that the parameters X2 and X3 had the most significance, while X1, X4 and X5 were lesser than the threshold of 0.05.

Results 2.0 - Final Prediction

After running tests on various machines, the best cross-validated mean test score I had was with the random forest machine, using 500 estimators. But my preference was again the random forest machine, however with 100 trees, which gave the cross validation results of -0.2064~. By looking at the feature importances, and my previous tests with plotting, linear regression, multiple linear regression, I had came up with the final decision of using X2 and X3 to predict the response values. So my final subset would be [X2, X3]. I had removed other predictors from the matrices, and run Random Forest to predict the response values. The prediction results are provided with the spreadsheet.

Conclusion

This project had made me learn more than I thought it would do. Initially, it was a great refreshment on the fundamentals of regression, especially with the linear regression. Because in order to take further steps for developing model, I first had to pass through the processes of analyzing and understanding the data. I had to read again and study about basic regression algorithms, error estimates, coefficients and the math behind to be able to actually start my project. Furthermore, I definitely had learned a lot about subset selection and error estimation. Especially the research I had to make on cross validation was very informative, and throughout the development of the project I had seen the practical approaches and applications of cross validation rather than just the logical part of it.

About my work, I think I could have gone with a number of different choices along my development process. The subset selection was the hardest choice I had to make. All the linear regression, with or without polynomials, suggested the usage of the predictors X2 and X3. However, due to collinearity and multicollinearity issues, I always thought I would get a better candidate with the non-linear algorithms once I move onto them. However, seeing that 3 of the predictors were not as much significant as the others on non-linear algorithms too, I settled my decision. Honestly, personally I did not want to choose only 2 predictors, because it felt missing. However, as the final results go, I trusted the scores from SVM and Random Forest.

For the estimator picking, I made the other toughest decision, relying on my hunch. I really would like to compare the results that I would have got with the 500 estimators in Random Forest once the actual responses are released.

Maybe there were other points I have missed, about subset selection, or cross-validation, model selection and analysis, that I should have taken into account. However, regardless of the accuracy of my model, I think I had learned a lot of new things and I will definitely try to get a feedback about my model to improve it after the project and learn more about the points I had missed.