Joseph Loss

XR Trading PC2 Writeup

Predicting Chicago Air Quality Through Linear Regression

Please see the *predictions.csv* file for the predicted values of each record in the "TESTING_DATA.txt" file.

The model itself can be run from the *main.py* file. Please be sure to include the two accompanying files, *create_new_features.py* and *readdata.py* in the same folder because these contain classes and functions that are used by the main file.

Feature Engineering and Variable Transformations:

During my EDA, I plotted regressions of each variable in relation to one another. From the correlation heatmap of covariates (Figure 1), I saw that Temperature.1 and Temperature.2 were extremely highly correlated at 0.99. When this happens, we know that keeping one covariate vs. the other does not add to the predictive power of the linear regression model. As a result, I used the product of the two highly correlated variables as an independent variable (removing the individual temperatures).

I was able to extract 9 new features out of the day.index and time.of.day variables. The exact details of my methodology are explained in the comments of my source code. The short explanation is that my code determined (starting on 01/31/2010) the date for each observation in the dataset. From there, I created a binary feature called "times.of.week" that tracked whether the observation fell on a weekday or on a weekend.

Using a similar methodology, I was also able to track the quarter of the year (starting on 01/01/2010 and adjusting 31 days to the start of the "day.index" variable), as well as the season of the year, which definitely has an impact on the levels of s02 in the air.

Note: the seasons were calculated using the date-ranges for Northern Hemisphere seasons. This is not only to improve the accuracy of my model's predictions but also due to the geographical location of Chicago.

I used a similar extrapolation to compute the time of day for each observation from the time.of.day variable. The head of the dfTrain dataframe is displayed in the figure on the next page.

Note that the “dummy variable” columns *time_night* and *season_winter* are removed from the model; this is standard practice when using pd.get_dummies.

Also note: the variables are not standardized in the image. The model uses X_train_std and X_test_std when fitting and predicting the datasets.

```
In[20]: pprint(X_train[:7])
     car.count  wind.velocity  temperature  times.of.week  quarters  \
0  1430.000396            1.3         0.51              0         1
1  2328.990055            2.5        21.60              1         2
2  2884.999484            2.2        96.04              1         3
3  3542.011269            5.9        18.90              0         2
4  2531.000606            0.5        73.87              0         3
5   989.995106            4.8         2.10              0         1
6  3714.985299            3.5         3.60              0         2

   time_afternoon  time_evening  time_morning  season_spring  season_summer
0               0             1             0              1              0
1               0             1             0              1              0
2               0             1             0              0              1
3               1             0             0              1              0
4               1             0             0              0              1
5               0             0             0              1              0
6               0             0             1              1              0
```

The model and summary statistics are on the following page, along with a plot of the model residuals vs fitted values.

Overall, this project took about a day to complete, but it probably would have taken less time had I known scikit-learn doesn’t output the model statistics or summary information (I first built the entire model using this package and had to re-write it using the statsmodels python library).

```
Model Summary:                            OLS Regression Results
==============================================================================
Dep. Variable:                    s02   R-squared:                       0.408
Model:                            OLS   Adj. R-squared:                  0.388
Method:                 Least Squares   F-statistic:                     19.94
Date:                Mon, 07 Oct 2019   Prob (F-statistic):           6.55e-28
Time:                        21:28:57   Log-Likelihood:                -1415.9
No. Observations:                 300   AIC:                             2854.
Df Residuals:                     289   BIC:                             2894.
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         52.5417      1.596     32.924      0.000      49.401      55.683
x1             8.5122      2.232      3.814      0.000       4.120      12.904
x2            -9.8851      1.701     -5.812      0.000     -13.233      -6.538
x3            -4.8614      1.704     -2.853      0.005      -8.215      -1.507
x4             8.0013      1.753      4.564      0.000       4.550      11.452
x5            -4.1000      3.732     -1.099      0.273     -11.445       3.245
x6             8.1172      2.647      3.066      0.002       2.907      13.328
x7             5.2095      2.226      2.340      0.020       0.828       9.591
x8             9.7867      2.554      3.832      0.000       4.761      14.813
x9             1.9853      4.073      0.487      0.626      -6.031      10.001
x10            2.5408      6.072      0.418      0.676      -9.411      14.492
==============================================================================
Omnibus:                      208.541   Durbin-Watson:                   1.881
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2598.434
Skew:                           2.712   Prob(JB):                         0.00
Kurtosis:                      16.359   Cond. No.                         7.77
==============================================================================

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

R2:  0.4083253857720419

Process finished with exit code 0
```

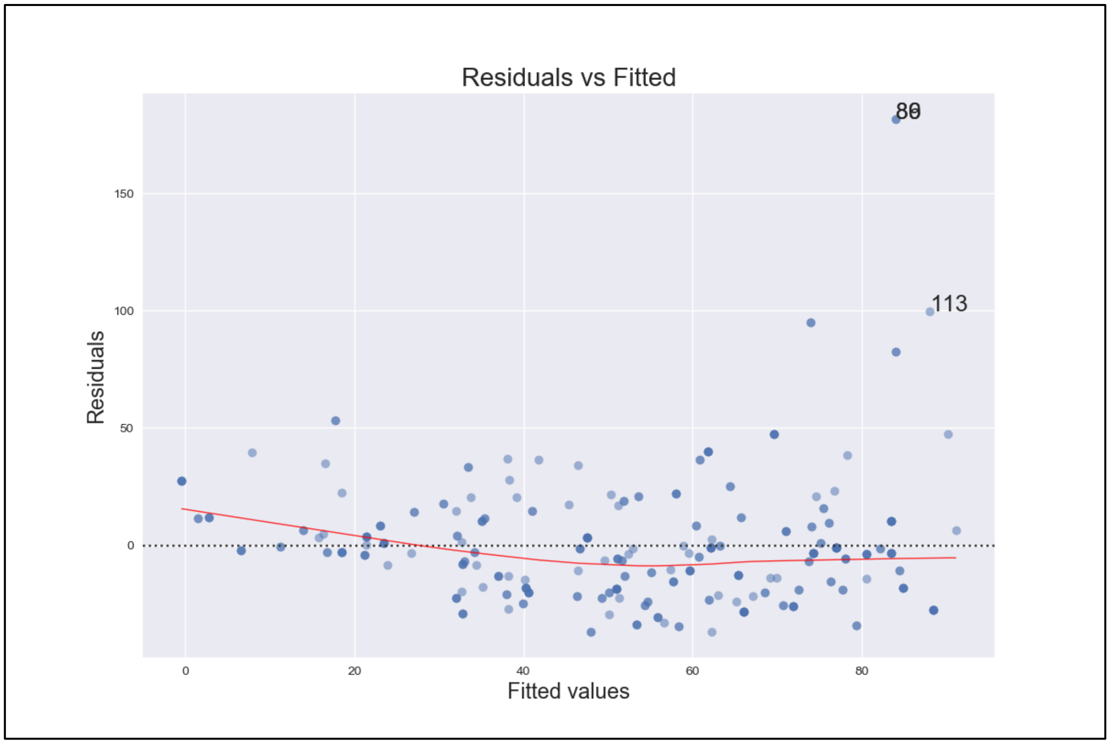

Residuals vs Fitted
80
113
150
100
50
0
Residuals
0
20
40
60
80
Fitted values

Figure 1

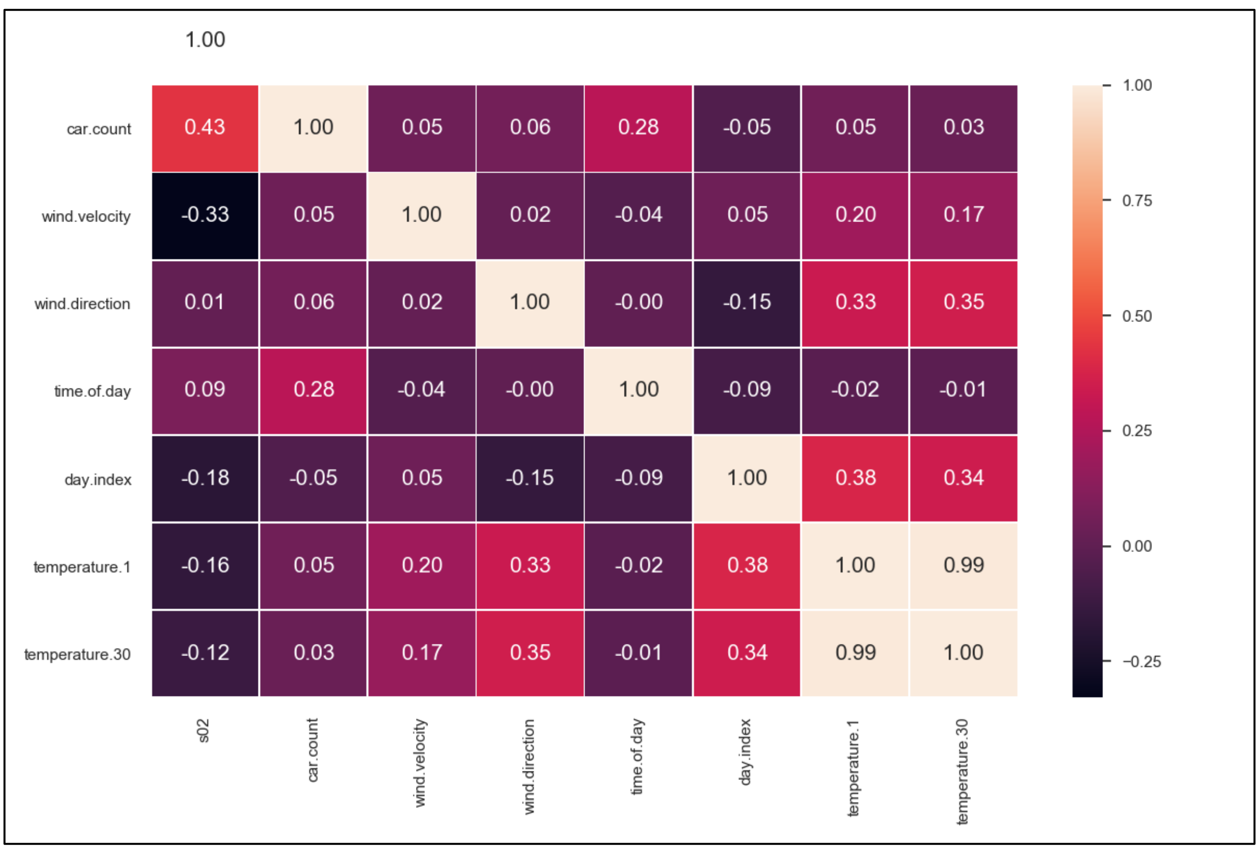
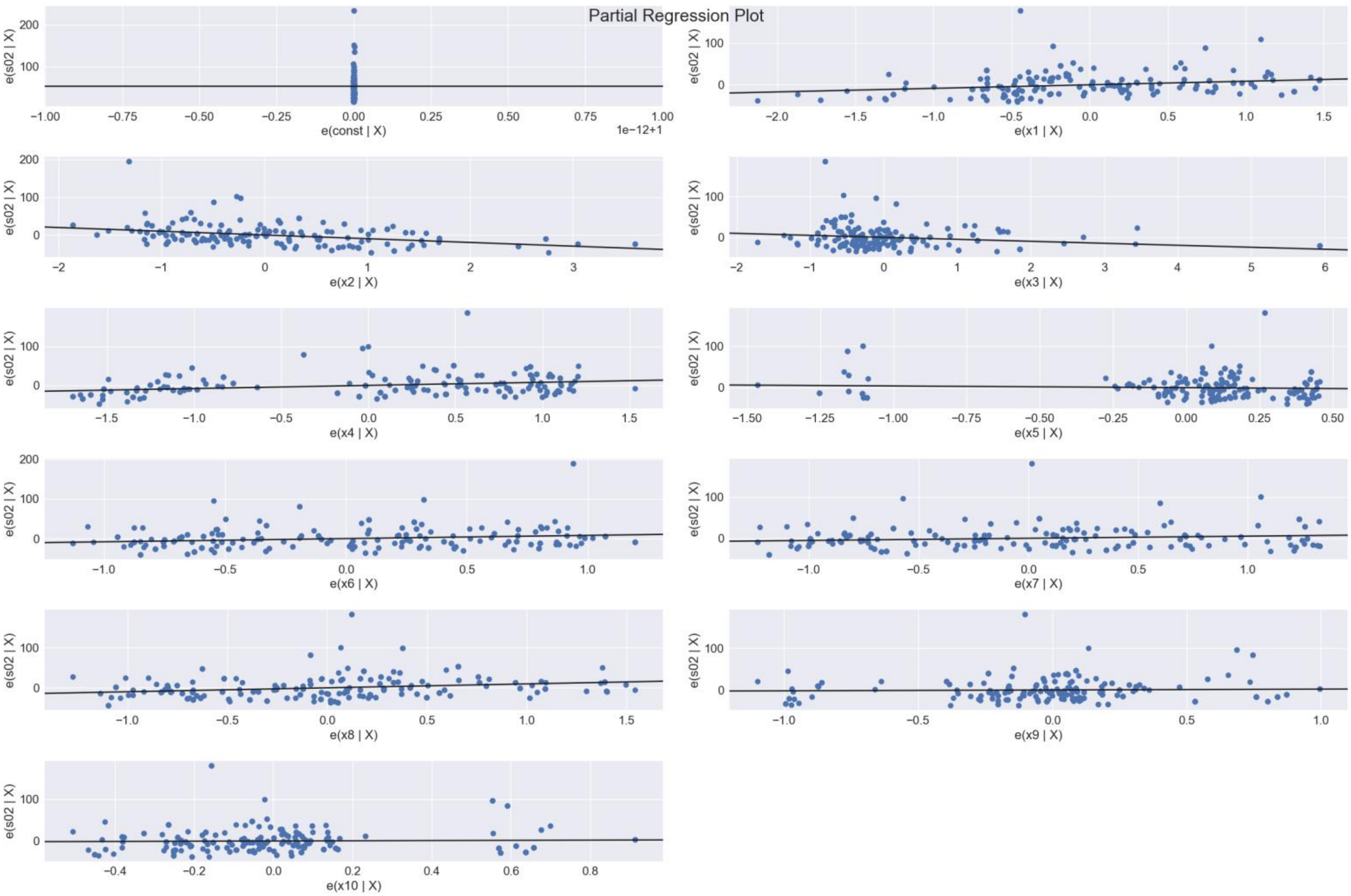

1.00
car.count 0.43 1.00 0.05 0.06 0.28 -0.05 0.05 0.03
wind.velocity -0.33 0.05 1.00 0.02 -0.04 0.05 0.20 0.17
wind.direction 0.01 0.06 0.02 1.00 -0.00 -0.15 0.33 0.35
time.of.day 0.09 0.28 -0.04 -0.00 1.00 -0.09 -0.02 -0.01
day.index -0.18 -0.05 0.05 -0.15 -0.09 1.00 0.38 0.34
temperature.1 -0.16 0.05 0.20 0.33 -0.02 0.38 1.00 0.99
temperature.30 -0.12 0.03 0.17 0.35 -0.01 0.34 0.99 1.00
s02
car.count
wind.velocity
wind.direction
time.of.day
day.index
temperature.1
temperature.30
1.00
0.75
0.50
0.25
0.00
−0.25

Figure 2



Partial Regression Plot
e(s02 | X)
e(const | X)
1e−12+1
e(x1 | X)
e(x2 | X)
e(x3 | X)
e(x4 | X)
e(x5 | X)
e(x6 | X)
e(x7 | X)
e(x8 | X)
e(x9 | X)
e(x10 | X)