

# **The US Inflation Phenomenon | It's Oil, silly**

**Author | Rand Sobczak Jr.**

**Date | 14 September 2021**

# Table of Contents

- 1.0 [Problem Identification Overview](#)
- 2.0 [Generated Deliverables](#)
- 3.0 [Data Pre-Processing Steps](#)
- 4.0 [Model Description](#)
- 5.0 [Model Findings](#)
- 6.0 [Next Steps](#)

## 1.0 Problem Identification Overview

**The United States Consumer Price Index ( "Inflation" )** is calculated by the U.S. Bureau of Labor Statistics. It has gone through various periods of prominent increases; notably in the 1920's, 1940's, & 1970's. Otherwise, it has remained relatively constant or declining.

**Inflation is important in all facets of life but the financial world pays special attention to it** as the key objectives of the Federal Reserve are maximizing employment, stabilizing prices & moderating long-term interest rates; the second of which is Inflation & the third of which is generally decided by the condition of the other two ( 2 ). Their decisions can move financial markets around the world; we won't go into that here. In short, **Inflation is an important component of developing investment strategies for portfolios across the world. The view on inflation becoming positive or negative in the future is not agreed upon nor are the variables which influence it.**

**The purpose of this Data Science project is to develop a model to help understand the phenomenon of Inflation. Nineteen ( 19 ) variables ( "Variables" ) were shortlisted to determine their influence on Inflation; [Appendix I](#).**

## 2.0 Generated Deliverables

**The following Application Programming Interface's ( "API" ) were used to pull the relevant data:**

1. Quandl
2. Investing.com ( "Investpy" )
3. Federal Reserve Economic Data ( "FRED" )

**The data retrieved by the API's are from sources highlighted in [Appendix I](#).**

Using these, **three ( 3 ) deliverables were generated:**

1. The source code for the modeling developed to analyze the aforementioned problem ( [link](#) )
2. This document outlining the process
3. A PDF presentation with our findings ( [link](#) )

## 3.0 Data Pre-Processing Steps

### Data Cleaning:

1. The data for the Variables & Inflation were first pulled using the API's
2. The result produced a data frame with non-congruent lengths in time ( [Appendix II](#) )
3. It was required to draw the line at the 18 April 1990 to ensure they aligned appropriately ( [Appendix II](#) )
4. This resulted in a Daily ( Mon-Fri ( excluding holidays ) ) comprised of 9,752 non-null float values ( [Appendix II](#) )
5. The Variables were concatenated with Inflation which resulted in a data frame with 321 observations ( [Appendix II](#) )
6. This data frame was used to cross reference the accuracy of the next steps in Exploratory Data Analysis

## Exploratory Data Analysis:

Which time periods for percent changes to use for the Variables & Inflation **wasn't confirmed**. The first step was to **run** through the process on multiple configurations with **Winsorization on Inflation** as per below:

1. **Quarter-on-Quarter**
  - a. **Description** | Looking at Quarterly percent changes in both Variables & Inflation
  - b. **Inflation Winsorization** | +3% & -2%
  - c. **Average Pearson Coefficient** | 22.64%
2. **Month-on-Month**
  - a. **Description** | Looking at Monthly percent changes in both Variables & Inflation
  - b. **Inflation Winsorization** |  $\pm 1\%$
  - c. **Average Pearson Coefficient** | 18.22%
3. **Quarter-on-Quarter for Variables ( past ) & Inflation ( forwards )**
  - a. **Description** | Looking at Quarterly percent changes on Variables looking backwards while forward looking Quarterly changes on Inflation to ascertain if changes in the Variables took time to reach Inflation
  - b. **Inflation Winsorization** | +2.51% & -2.84%
  - c. **Average Pearson Coefficient** | 12.18%
4. **Quarter-on-Quarter w/ Rolling Averages on Daily, Weekly & Monthly Variables**
  - a. **Description** | This approach is similar to # 1 albeit used a rolling average for those that reported more often than once a Quarter. The rational was that a Variable may have had a bad week or day when the Quarter ended; as such, the entire changes throughout the Quarter may need to be accounted for evenly
  - b. **Inflation Winsorization** |  $\pm 3\%$
  - c. **Average Pearson Coefficient** | 30.21%

It was also observed that Winsorization on Inflation in #4 above ( Best ) reduced the average Pearson Coefficient score. The results before Winsorization are below:

5. **Quarter-on-Quarter w/ Rolling Averages ( without Inflation Winsorized )**
  - a. **Description** | Same as #4 above less Winsorization on Inflation
  - b. **Inflation Winsorization** | n/a
  - c. **Average Pearson Coefficient** | 30.95%

It was decided that Winsorization did not work on the Inflation data herein; thus, Inflation was not Winsorized & this data frame was chosen to move on.

While Winsorization may not work on Inflation, that does not mean it doesn't work on the Variables. Eight (8) variables were chosen to be Winsorized. These saw an average increase in their Pearson correlation coefficients of 173 bps with one seeing a 460 bps increase. Our final round presented the following:

6. **Quarter-on-Quarter w/ Rolling Averages ( without Inflation Winsorized & 8 Variables Winsorized )**
  - a. **Description** | Same as #5 above but 8 variables Winsorized
  - b. **Inflation Winsorization** | n/a
  - c. **Average Pearson Coefficient** | 31.67%

The "Win\_plus" column in the chart on the right, shows each of the variables improvement with Winsorization. If it's a Zero, the Variable wasn't chosen.

	NoWinzor_p	Win_p	Win_plus	Winsorized?
Wage CPI	0.595358	0.595358	0.000000	n/a
WTI	0.540347	0.541041	0.000694	Winsorized
Heating Oil	0.542445	0.542445	0.000000	n/a
Copper	0.372813	0.372813	0.000000	n/a
Sugar	0.219235	0.220639	0.001405	Winsorized
Natural Gas	0.289579	0.292725	0.003146	Winsorized
Cattle	0.162077	0.162077	0.000000	n/a
Lean Hogs	0.291943	0.291943	0.000000	n/a
Soybeans	0.199634	0.199634	0.000000	n/a
Lumber	0.255493	0.255685	0.000191	Winsorized
Capacity Utilization	0.349075	0.367072	0.017997	Winsorized
Corn	0.293481	0.293481	0.000000	n/a
M2 Velocity	0.281344	0.281344	0.000000	n/a
GDP	0.343129	0.387090	0.043961	Winsorized
Wheat	0.125293	0.125293	0.000000	n/a
PMI	0.223222	0.223222	0.000000	n/a
USD Index	0.307158	0.307158	0.000000	n/a
Unemployment Rate	0.215974	0.261978	0.046005	Winsorized
Initial Jobless Claims	0.272464	0.297205	0.024741	Winsorized

The **Feature Correlation Heat Maps with the Pearson correlation coefficients of each variable against Inflation are found in [Appendix III](#)**; these go into greater detail on the Average Pearson Coefficient ( c ) above. Scatter Plots for those Winsorized are in [Appendix IV](#); all others are in the source code.

In summary, **the Best Configuration ( #6 ) was chosen as it presented the most optimal Pearson correlation coefficients to achieve our goal of developing a model to understand the phenomenon of Inflation.**

## Pre-processing:

1. The first step in pre-processing, per the Machine Learning process, was to create a “Best Guess” number. The standard mean & DummyRegressor functions were used. The data frame has no categorical data so **this step** was purely to “go through the process” but **is irrelevant; please disregard when reading the source code**
2. The next step was to **split the data into training & testing data frames, 70% & 30% respectively**
3. Next was to **scale the data. There is no certainty to which scaling technique to use & thus, the following were all applied on both the X & y variables & X only** ( x6 in total ):
  - a. Standard Scaling ( SS )
  - b. MinMax Scaling ( MM )
  - c. Log Transformation ( LG )
4. **The Variables were divided into selected groups to apply SS, MM & LG separately to where SS, MM & LG may be better suited**; they were concatenated with Inflation
5. The **initial results** ( [Appendix V](#) ) showed that MM presented negative results for  $R^2$  on the test set. The Mean Absolute Error ( MAE ) & the Root Mean Squared Error ( RMSE ) presented a mixed bag for the MM structure. It was decided that due to this relative uncertainty & negative  $R^2$ , the MM structure was taken out of consideration.
6. Five ( 5 ) scaling approaches were used & it was determined that all showed results that should be put to the process of the Random Forest Generator to identify the best Variables presented below & in [Appendix VI](#).

`R2 results for X & y scaled below`

`SS Train | 0.5055    Test 0.2962`

`LG Train | 0.4983    Test 0.2781`

`R2 results for X only scaled below`

`SS Train | 0.5133    Test 0.2925`

`LG Train | 0.5005    Test 0.2732`

`R2 results for the LG & SS combination below`

`SS Train | 0.5053    Test 0.2788`

## 4.0 Model Description

Although Dummy Regressor & Linear Regression analysis was undertaken, they were discounted due to the data frame being entirely float based & displayed comparatively worse performance respectively. Therefore, **the Random Forest Model was used with the goal of determining what variables best explain Inflation.**

## 5.0 Model Findings

As the purpose of this Data Science project is to develop a model to explain & understand the phenomenon of Inflation, **the following process was undertaken on the five ( 5 ) shortlisted scaling approaches:**

1. Grid Search
2. Random Forest
3. Hyperparameter search using Grid Search CV

The final outcomes can be seen in [Appendix VII](#).

**WTI held a ubiquitous position as being the dominate Variable on all scaling approaches. This may be justified due to its connection with many activities in the United States.** Heating Oil, a byproduct using WTI, & Wages CPI stayed in second & third place on many respectively.

Nevertheless, they weren't the only ones helping improve the k-value. The total number of variables ranged from 9 to 11 & the breakdown is in [Appendix VII](#) for review.

The variables that were not helping were then removed for each & the process was run the last time. The results are below:

R <sup>2</sup> results for X & y scaled below		MAE results for X & y scaled below		RMSE results for X & y scaled below	
SS Train	0.492	Test 0.2706	SS Train	0.5143	Test 0.6133
LG Train	0.4682	Test 0.2862	LG Train	0.5261	Test 0.5955
R <sup>2</sup> results for X only scaled below		MAE results for X only scaled below		RMSE results for X only scaled below	
SS Train	0.492	Test 0.2734	SS Train	0.4526	Test 0.6034
LG Train	0.7563	Test 0.6524	LG Train	0.2229	Test 0.294
R <sup>2</sup> results for the LG & SS combination below		MAE results for the LG & SS combination below		RMSE results for the LG & SS combination below	
SS Train	0.4776	Test 0.2918	SS Train	0.2229	Test 0.294
					SS Train   0.4343 Test 0.5702

After review of the **Test set results ( the arbiter )**, the **LG on X only was chosen** amongst the other scaling approaches **given it had the highest R<sup>2</sup> & tied as the lowest MAE & RMSE.**

So **how do these test results compare to those presented in the Pre-processing step?**

Comparing final to the averages in the Pre-processing Step

37.92 bps increase in R<sup>2</sup>

A -23.52 bps decrease in MAE

A -15.28 bps decrease in RMSE

**Our conclusion on how to explain & understand the phenomenon of Inflation** is a reconfiguration of words used by James Carville ( [link](#) ) | **It's Oil, silly.**

## 6.0 Next Steps

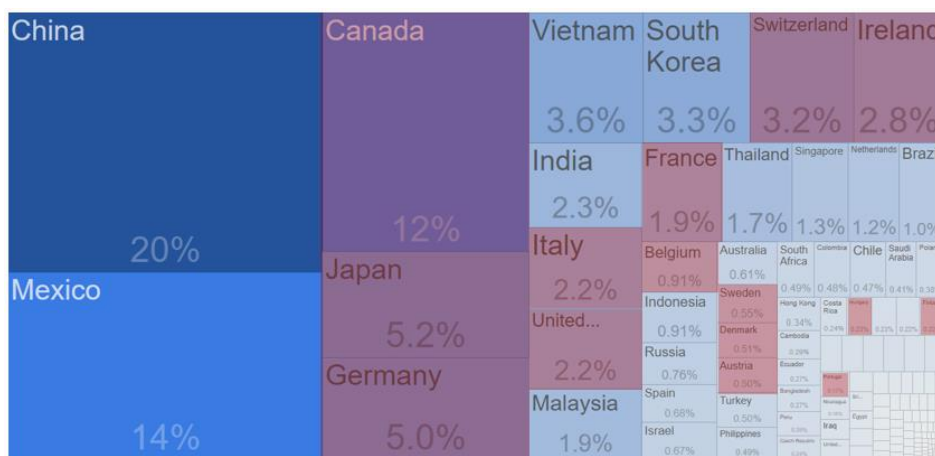
**Inflation is a difficult & highly disputed financial beast but the closer you get to taming it, your eyes will open wider.**

Throughout the process there were a number of actions & variables that may have been done or included. **Highlighted below are some Variables which were not included** due to financial & time constraints:

1. Steel
  - Could only identify data back to 2008
2. Gasoline
  - Could only identify data back to 2005
3. Growth of M2
  - Put it aside to mitigate any collinearity with M2 Velocity
4. US Wages Hourly Earnings
  - Limited data as well
5. US Dollar Index: Broad, Goods and Services
  - This only goes back to 2006

Further to these, **further consideration may be applicable to the below:**

1. The big set back would be the size of the data frame. With only 321 observations, machine learning is limited
2. Winsorization on Inflation & other variables may be re-examined
3. Reassess the Variables chosen in the SS & LG divide; discussed in Pre-processing above
4. Develop a model to predict Wages CPI itself in order to minimize reliance on US gov't reporting
5. It is believed that the US Dollar Index Variable ( DXY ) does not correctly address the relationship of US Imports on Inflation. This may be because the DXY's weighting does not correctly align to US trade. In short, it's a weighted geometric mean of the currencies in the Eurozone ( EUR ), Japan ( JPY ), the United Kingdom ( GBP ), Canada ( CAD ), Sweden ( SEK ) & Switzerland ( CHF ) but that does not correctly align with US trade with the world
  - Highlighted below is the US Import Trade in 2020 by country; in red are those in the DXY
  - It only accounts for under ~40%



There are a number of Variables to take into consideration. **For now, the machine recommends keeping an eye on Oil.**

## Appendix I

Items	Reported	API	API Source		Comments
Inflation	Monthly	Quandl	U.S. Bureau of Labor Statistics		The target variable
Wages CPI	Monthly	FRED	U.S. Bureau of Labor Statistics		A component of the target variable
WTI	Daily	Quandl	CME	West Texas Intermediate -	One of many commodities
Heating Oil	Daily	Investpy	Investing.com		One of many commodities
Copper	Daily	Investpy	Investing.com		One of many commodities
Sugar	Daily	Investpy	Investing.com		One of many commodities
Natural Gas	Daily	Investpy	Investing.com		One of many commodities
Cattle	Daily	Investpy	Investing.com		One of many commodities
Lean Hogs	Daily	Investpy	Investing.com		One of many commodities
Soybeans	Daily	Investpy	Investing.com		One of many commodities
Lumber	Daily	Investpy	Investing.com		One of many commodities
Capacity Utilization	Monthly	FRED	Board of Governors of the Federal Reserve		The % of resources used by corporations
Corn	Daily	Investpy	Investing.com		One of many commodities
M2 Velocity	Quarterly	FRED	Federal Reserve Bank of St. Louis	Movement of money; state of the economy proxy	
GDP	Quarterly	FRED	U.S. Bureau of Economic Analysis		A proxy for the state of the economy
Wheat	Daily	Investpy	Investing.com		One of many commodities
PMI	Monthly	Quandl	Institute of Supply Management	Manufacturing PMI - A proxy for the economy	
USD Index	Daily	Quandl	Intercontinental Exchange Inc	( DXY ) Proxy for potentially importing inflation	
Unemployment Rate	Monthly	Quandl	U.S. Bureau of Labor Statistics		A proxy for the state of the economy
Initial Jobless Claims	Weekly	Quandl	U.S. Employment and Training Administration		A proxy for the state of the economy

( [back](#) )



## Appendix II

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 14302 entries, 1946-01-01 to 2021-09-03
Data columns (total 19 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Wage CPI                             14293 non-null  float64
1   WTI                                  12088 non-null  float64
2   Heating Oil                          13087 non-null  float64
3   Copper                              10440 non-null  float64
4   Sugar                               13087 non-null  float64
5   Natural Gas                          9915 non-null   float64
6   Cattle                              13084 non-null  float64
7   Lean Hogs                           13089 non-null  float64
8   Soybeans                            9999 non-null   float64
9   Lumber                              13089 non-null  float64
10  Capacity Utilization                 14033 non-null  float64
11  Corn                                13086 non-null  float64
12  M2 Velocity                         14151 non-null  float64
13  GDP                                 14295 non-null  float64
14  Wheat                              10001 non-null  float64
15  PMI                                14281 non-null  float64
16  USD Index                          11273 non-null  float64
17  Unemployment Rate                   14281 non-null  float64
18  Initial Jobless Claims              14030 non-null  float64
dtypes: float64(19)
memory usage: 2.2 MB
```

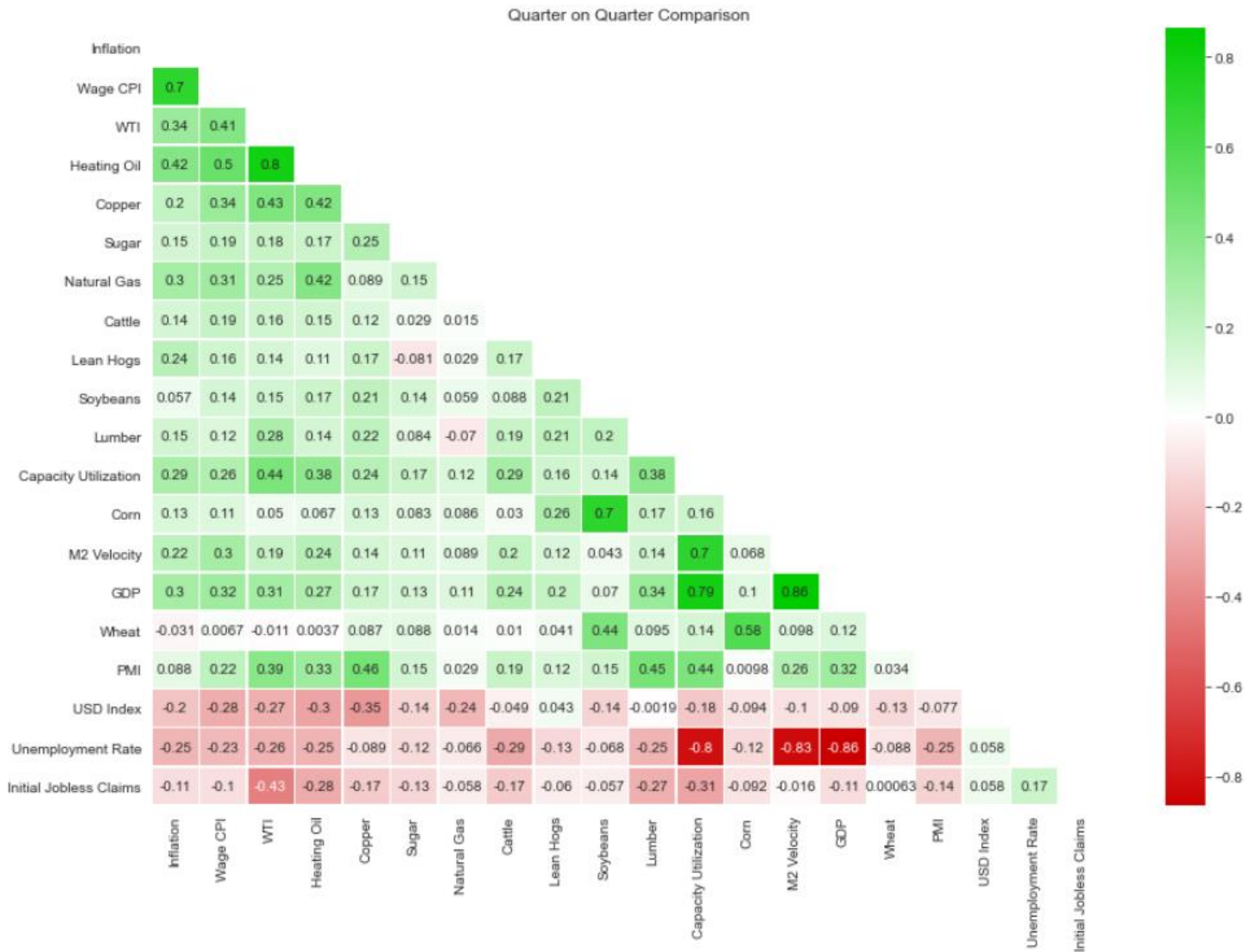
```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 9752 entries, 1991-04-18 to 2021-09-03
Data columns (total 19 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Wage CPI                             9752 non-null  float64
1   WTI                                  9752 non-null  float64
2   Heating Oil                          9752 non-null  float64
3   Copper                              9752 non-null  float64
4   Sugar                               9752 non-null  float64
5   Natural Gas                          9752 non-null  float64
6   Cattle                              9752 non-null  float64
7   Lean Hogs                           9752 non-null  float64
8   Soybeans                            9752 non-null  float64
9   Lumber                              9752 non-null  float64
10  Capacity Utilization                 9752 non-null  float64
11  Corn                                9752 non-null  float64
12  M2 Velocity                         9752 non-null  float64
13  GDP                                 9752 non-null  float64
14  Wheat                              9752 non-null  float64
15  PMI                                9752 non-null  float64
16  USD Index                          9752 non-null  float64
17  Unemployment Rate                   9752 non-null  float64
18  Initial Jobless Claims              9752 non-null  float64
dtypes: float64(19)
memory usage: 1.5 MB
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 321 entries, 1991-04-30 to 2021-07-31
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Inflation                             321 non-null    float64
1   Wage CPI                             321 non-null    float64
2   WTI                                  321 non-null    float64
3   Heating Oil                          321 non-null    float64
4   Copper                              321 non-null    float64
5   Sugar                               321 non-null    float64
6   Natural Gas                          321 non-null    float64
7   Cattle                              321 non-null    float64
8   Lean Hogs                           321 non-null    float64
9   Soybeans                            321 non-null    float64
10  Lumber                              321 non-null    float64
11  Capacity Utilization                 321 non-null    float64
12  Corn                                321 non-null    float64
13  M2 Velocity                         321 non-null    float64
14  GDP                                 321 non-null    float64
15  Wheat                              321 non-null    float64
16  PMI                                321 non-null    float64
17  USD Index                          321 non-null    float64
18  Unemployment Rate                   321 non-null    float64
19  Initial Jobless Claims              321 non-null    float64
dtypes: float64(20)
memory usage: 52.7 KB
```

( [back](#) )

## Appendix III ( Feature Correlation Heatmap )

#1

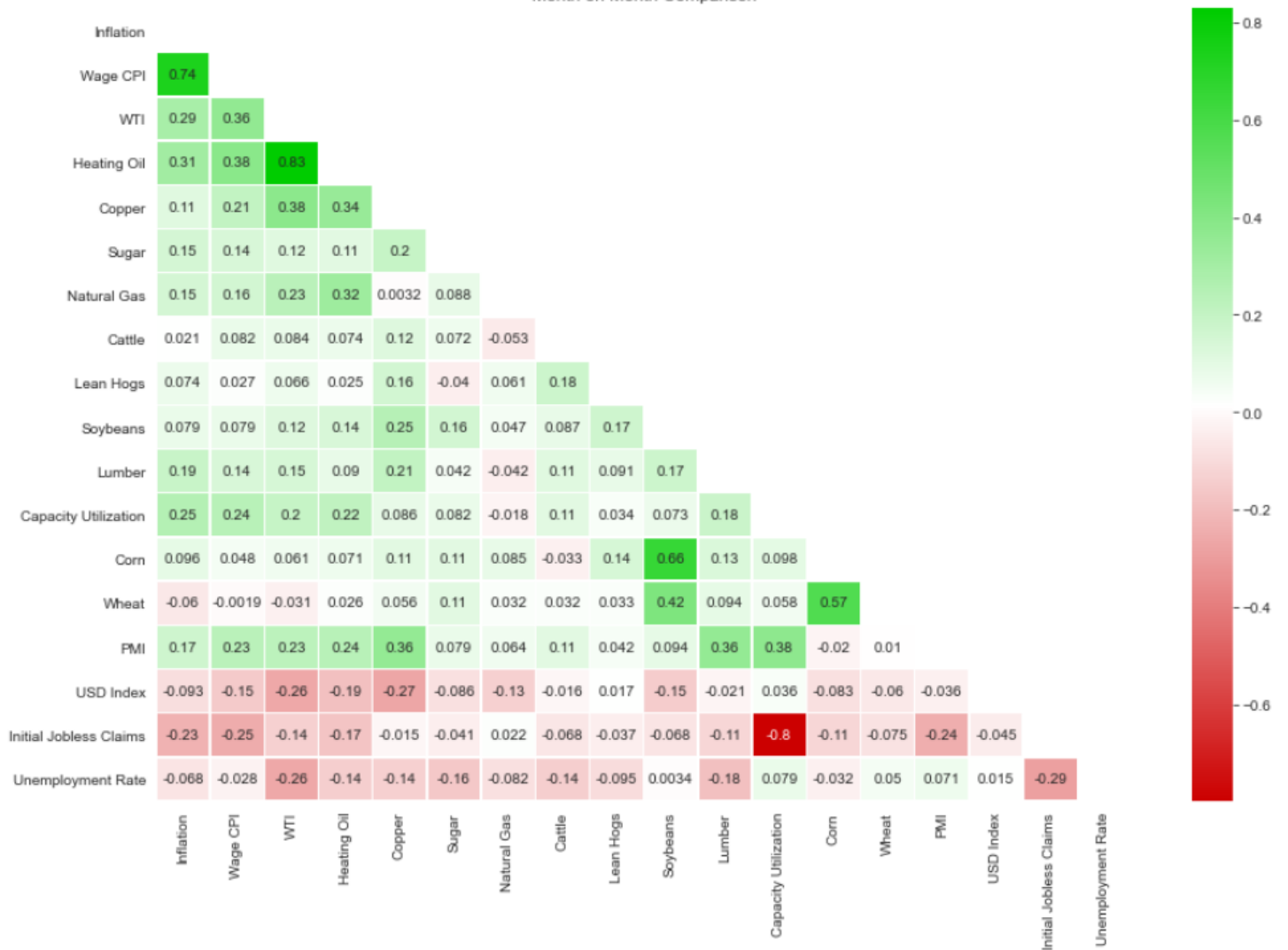


( [back](#) )

## Appendix III ( Feature Correlation Heatmap )

#2

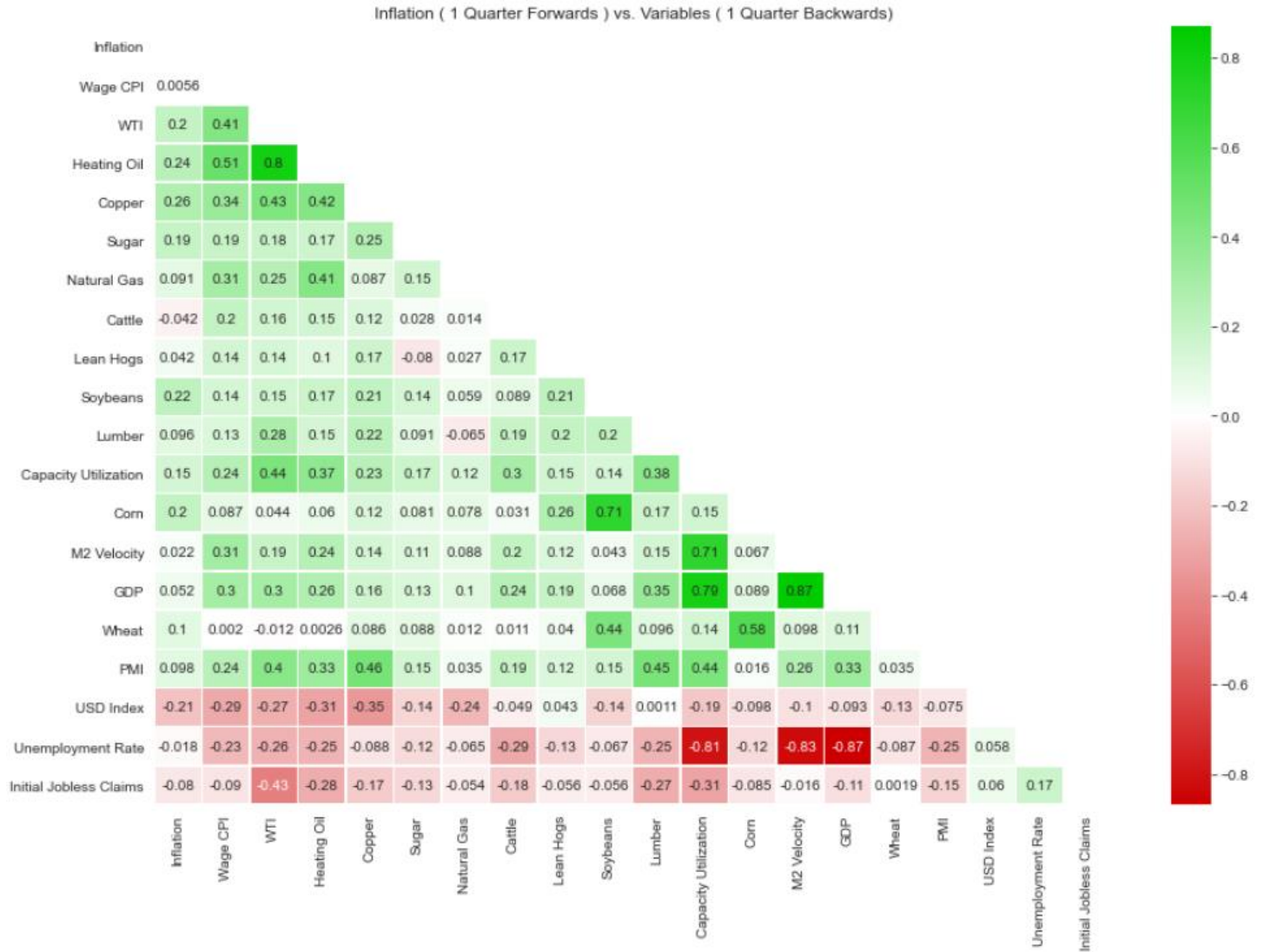
Month on Month Comparison



( [back](#) )

## Appendix III ( Feature Correlation Heatmap )

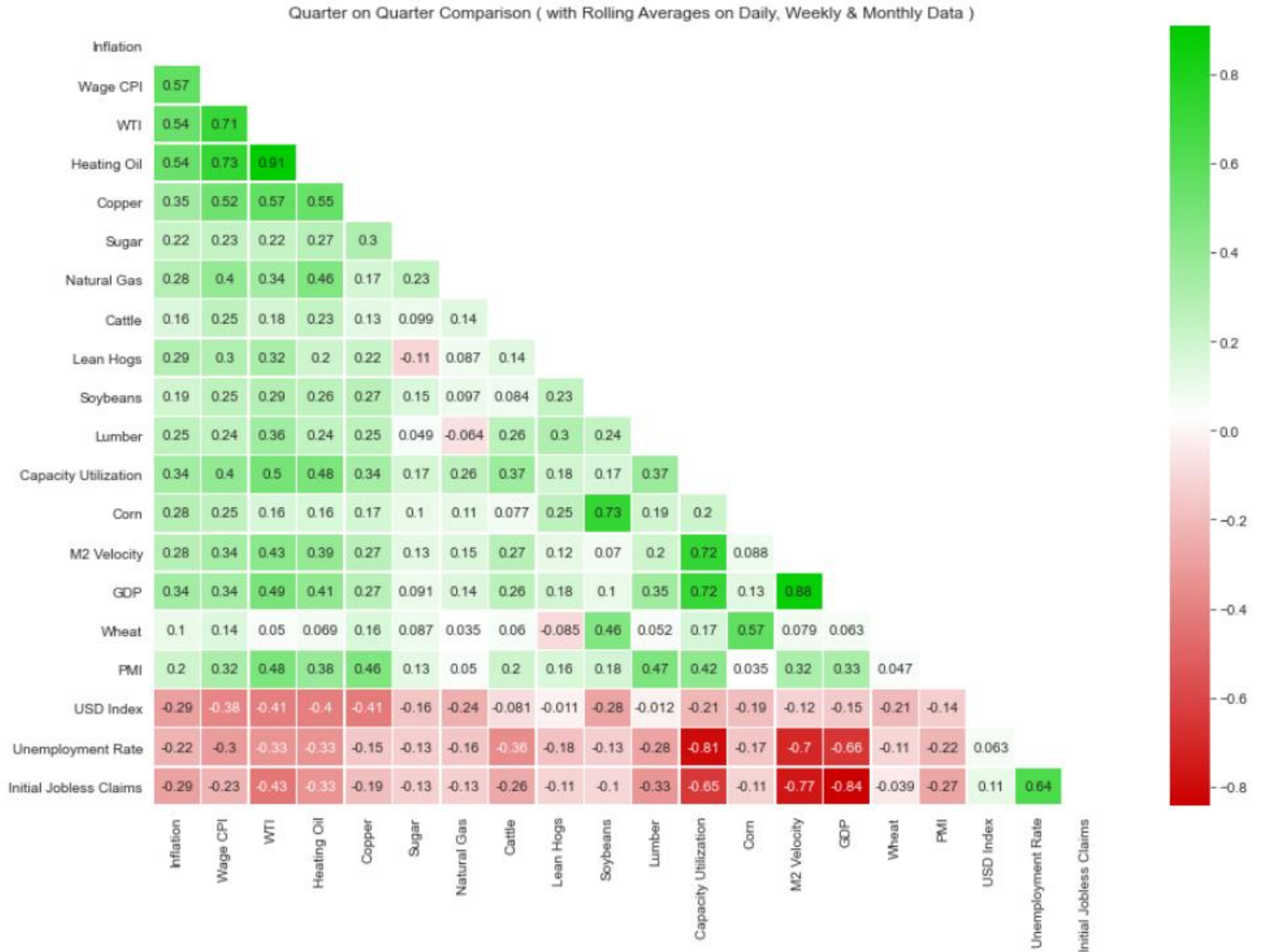
#3



( [back](#) )

## Appendix III ( Feature Correlation Heatmap )

#4

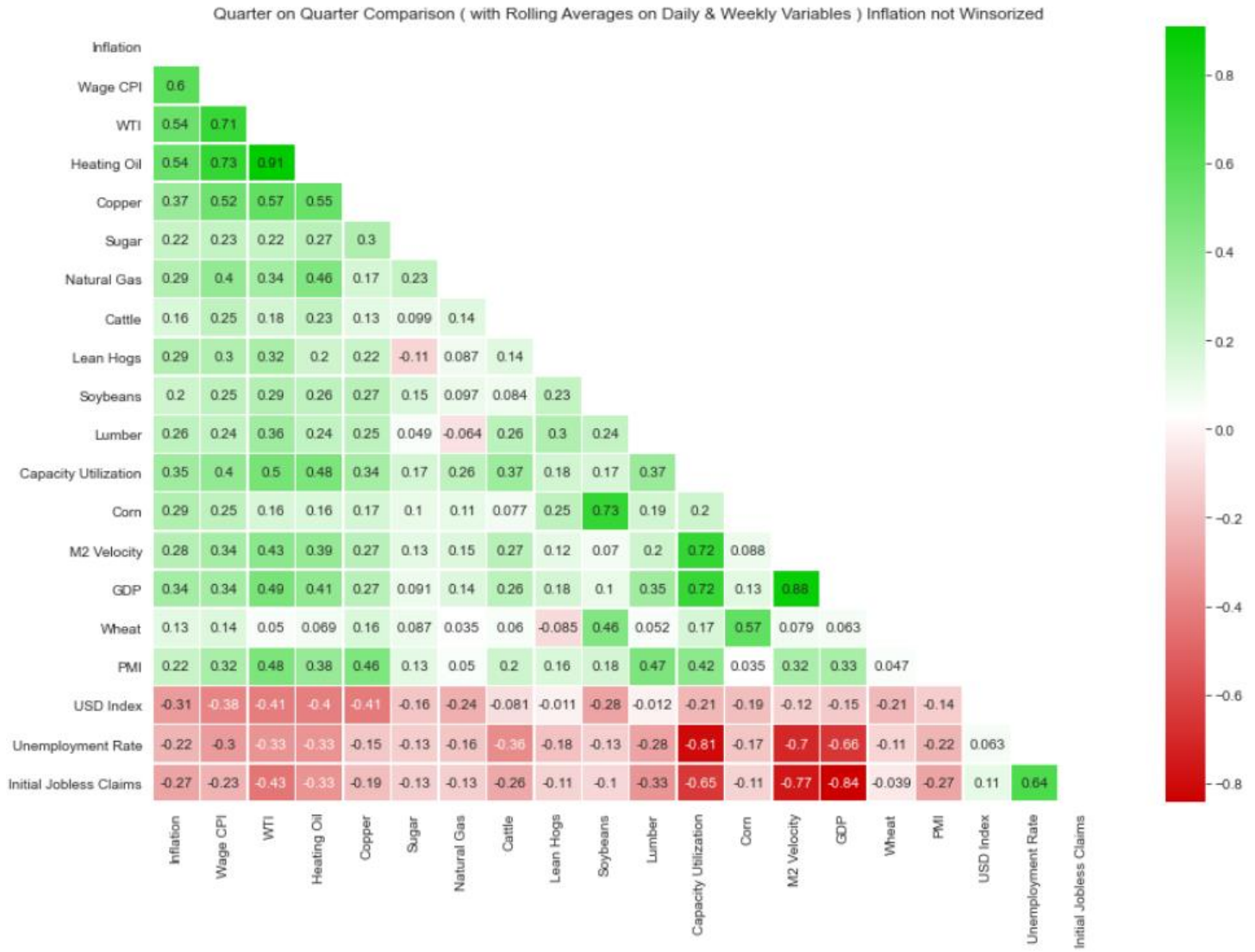


( [back](#) )



## Appendix III ( Feature Correlation Heatmap )

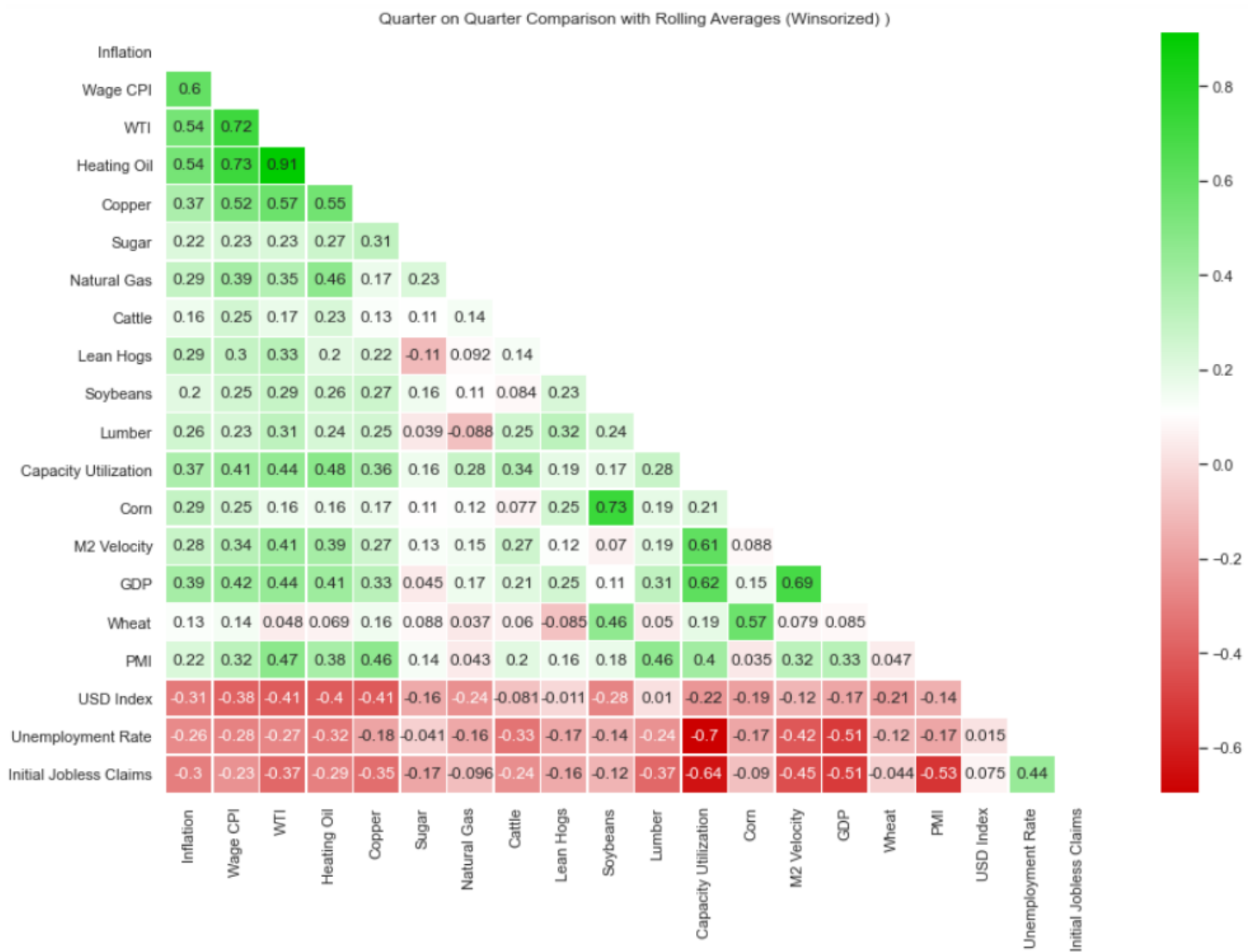
#5



( [back](#) )

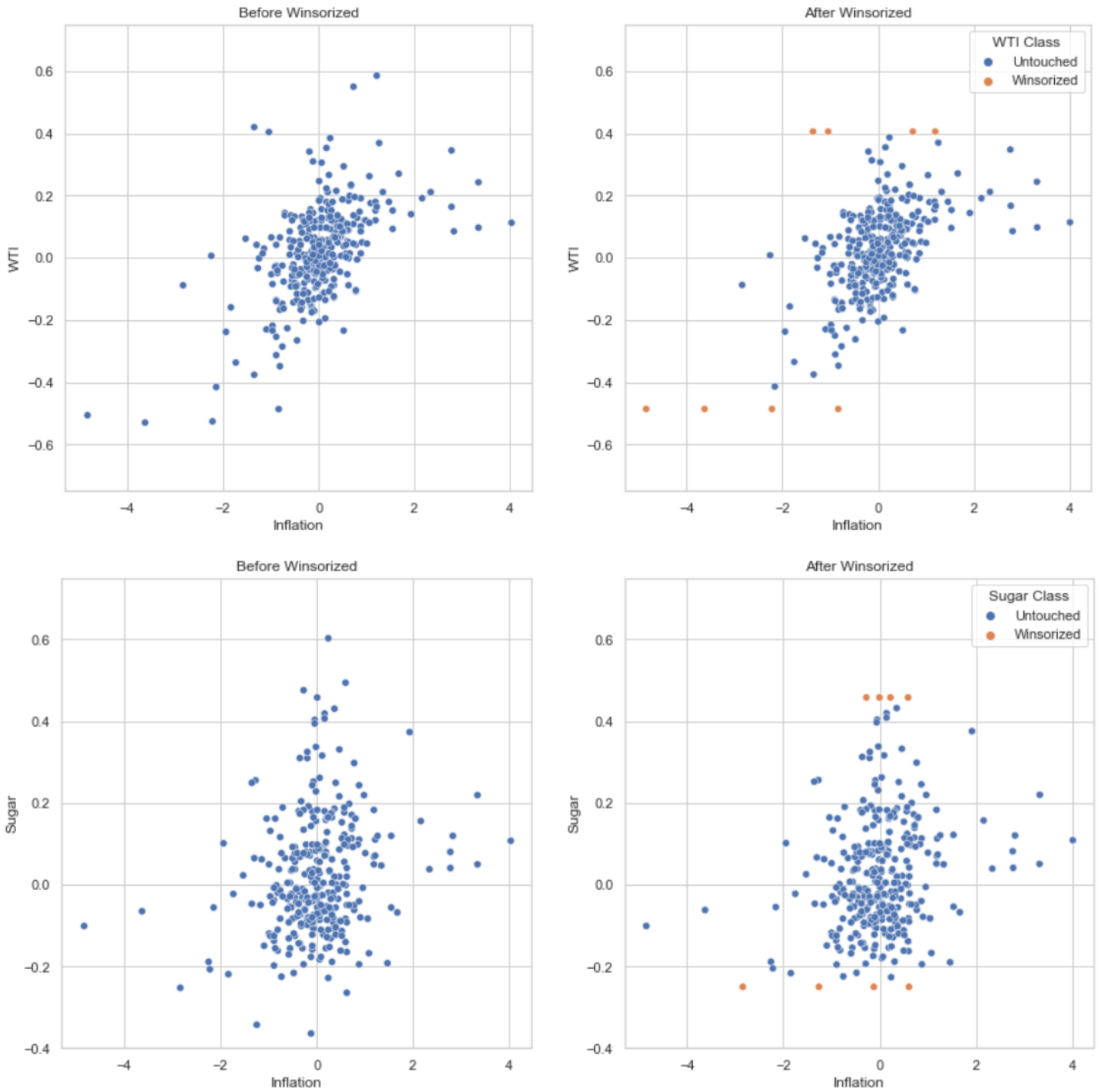
## Appendix III ( Feature Correlation Heatmap )

#6



( [back](#) )

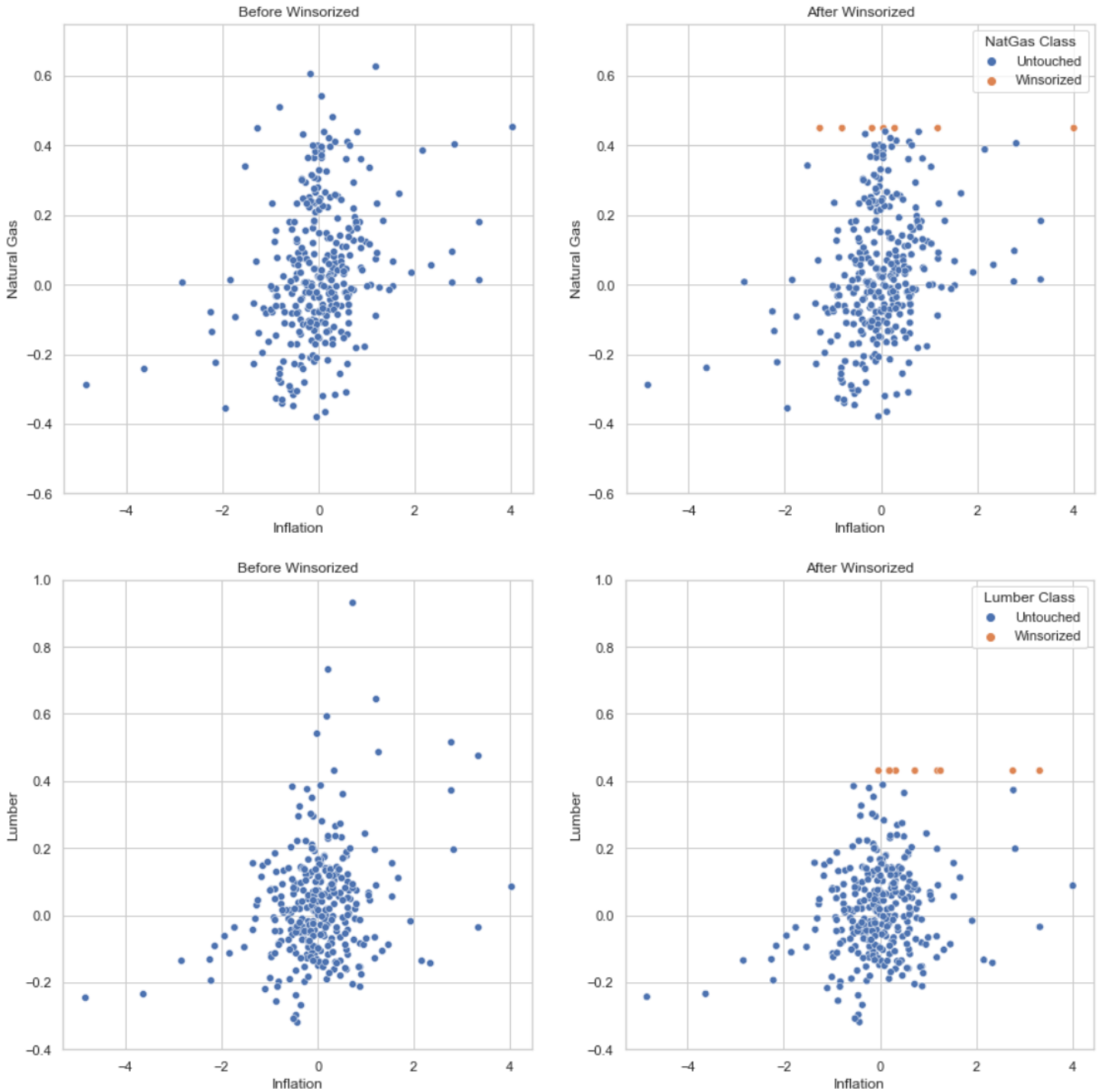
## Appendix IV (Scatter Plots)



( [back](#) )

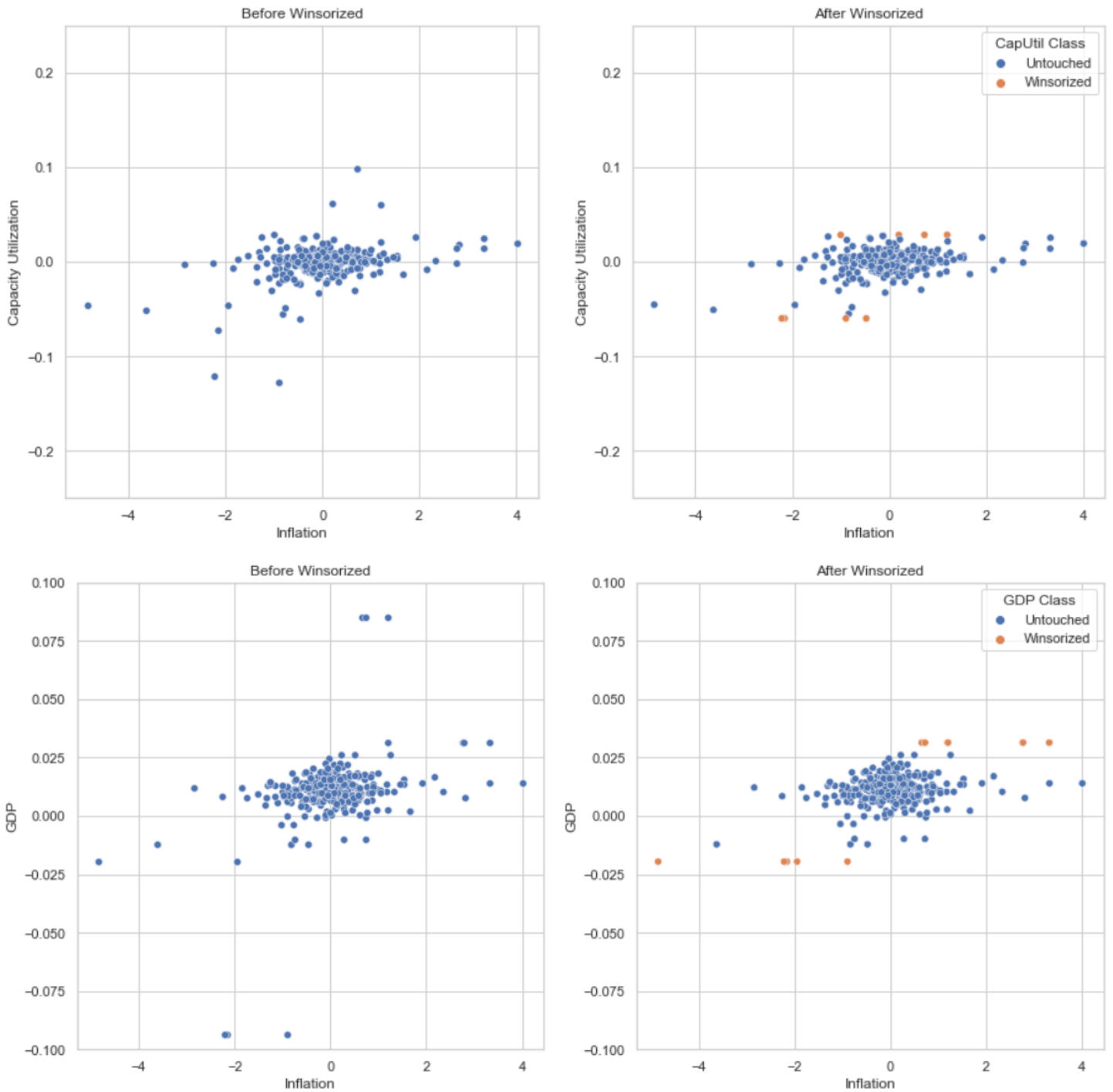


## Appendix IV (Scatter Plots)



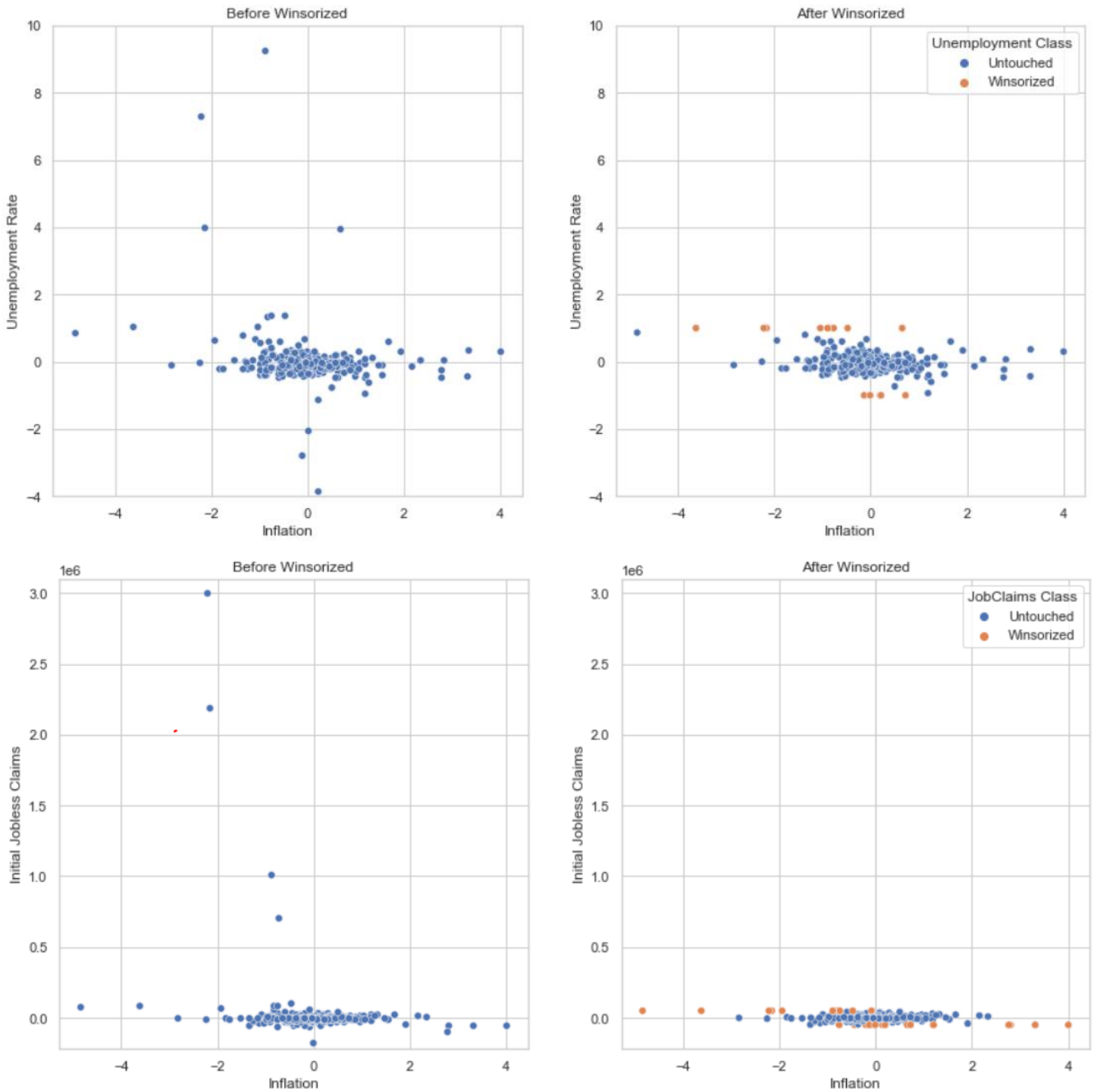
( [back](#) )

## Appendix IV (Scatter Plots)



( [back](#) )

## Appendix IV (Scatter Plots)



( [back](#) )

## Appendix V

R<sup>2</sup> results for nothing scaled below  
Test 0.2925 ( nothing scaled )

R<sup>2</sup> results for X & y scaled below  
SS Train | 0.5055 Test 0.2962  
MM Train | -6.3454 Test -6.8587  
LG Train | 0.4983 Test 0.2781

R<sup>2</sup> results for X only scaled below  
SS Train | 0.5133 Test 0.2925  
MM Train | 0.057 Test -0.042  
LG Train | 0.5005 Test 0.2732

R<sup>2</sup> results for the LG & SS combination below  
SS Train | 0.5053 Test 0.2788

R<sup>2</sup> averages of LG & SS X only scaled below  
Av. Train | 0.5069 Test 0.2828

~~~~~

MAE results for nothing scaled below  
Test 0.5214 ( nothing scaled )

MAE results for X & y scaled below  
SS Train | 0.5085 Test 0.5859  
MM Train | 0.2581 Test 0.2538  
LG Train | 0.5172 Test 0.603

MAE results for X only scaled below  
SS Train | 0.4461 Test 0.5214  
MM Train | 0.5971 Test 0.6354  
LG Train | 0.4545 Test 0.5291

MAE results for the LG & SS combination below  
SS Train | 0.4488 Test 0.5229

MAE averages of LG & SS X only scaled below  
Av. Train | 0.4503 Test 0.5253

~~~~~

RMSE results for nothing scaled below  
Test 0.7133 ( nothing scaled )

RMSE results for X & y scaled below  
SS Train | 0.7032 Test 0.8086  
MM Train | 0.2694 Test 0.2685  
LG Train | 0.7083 Test 0.8218

RMSE results for X only scaled below  
SS Train | 0.6139 Test 0.7133  
MM Train | 0.8545 Test 0.8657  
LG Train | 0.6219 Test 0.723

RMSE results for the LG & SS combination below  
SS Train | 0.6189 Test 0.7202

RMSE averages of LG & SS X only scaled below  
Av. Train | 0.6179 Test 0.7182

( [back](#) )

## Appendix VI

R<sup>2</sup> results for X & y scaled below

SS Train | 0.5055    Test 0.2962

LG Train | 0.4983    Test 0.2781

R<sup>2</sup> results for X only scaled below

SS Train | 0.5133    Test 0.2925

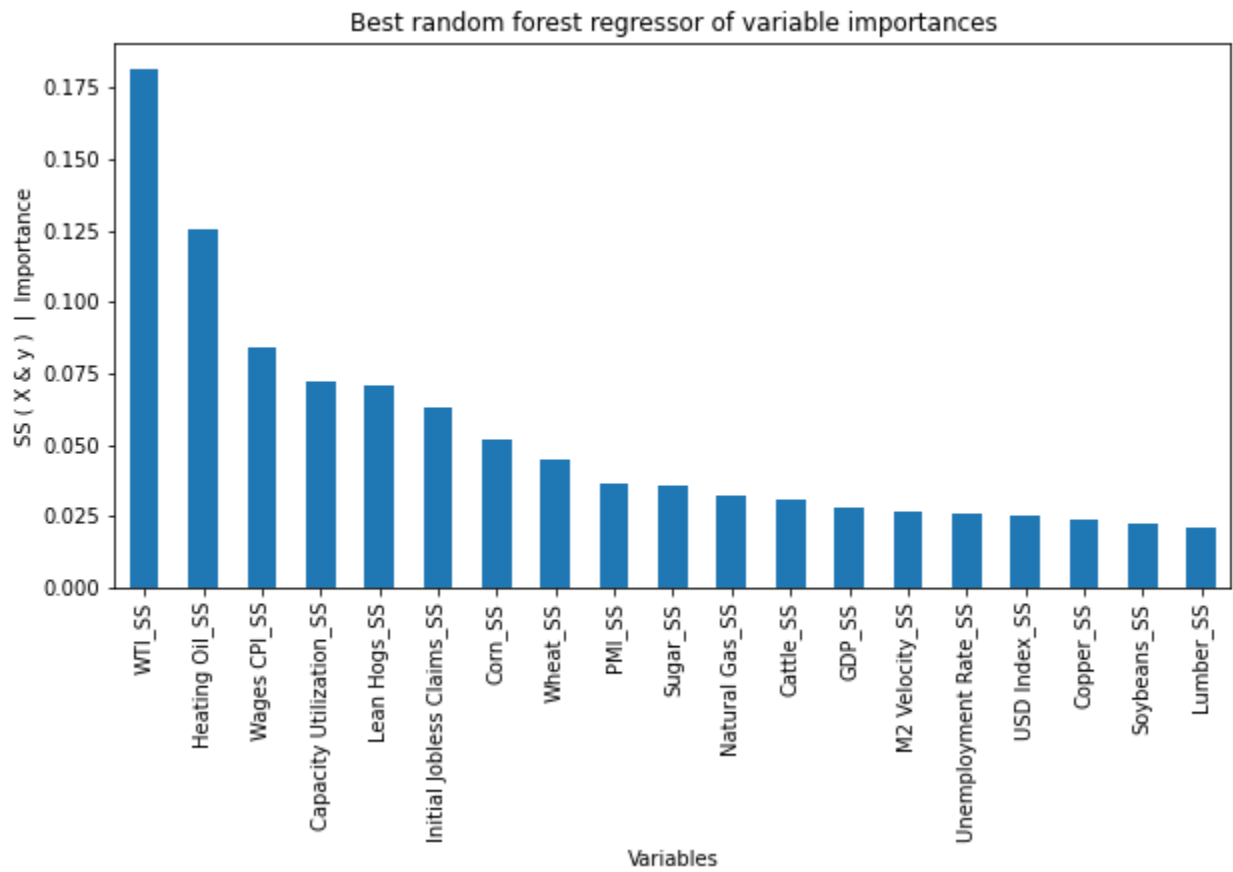
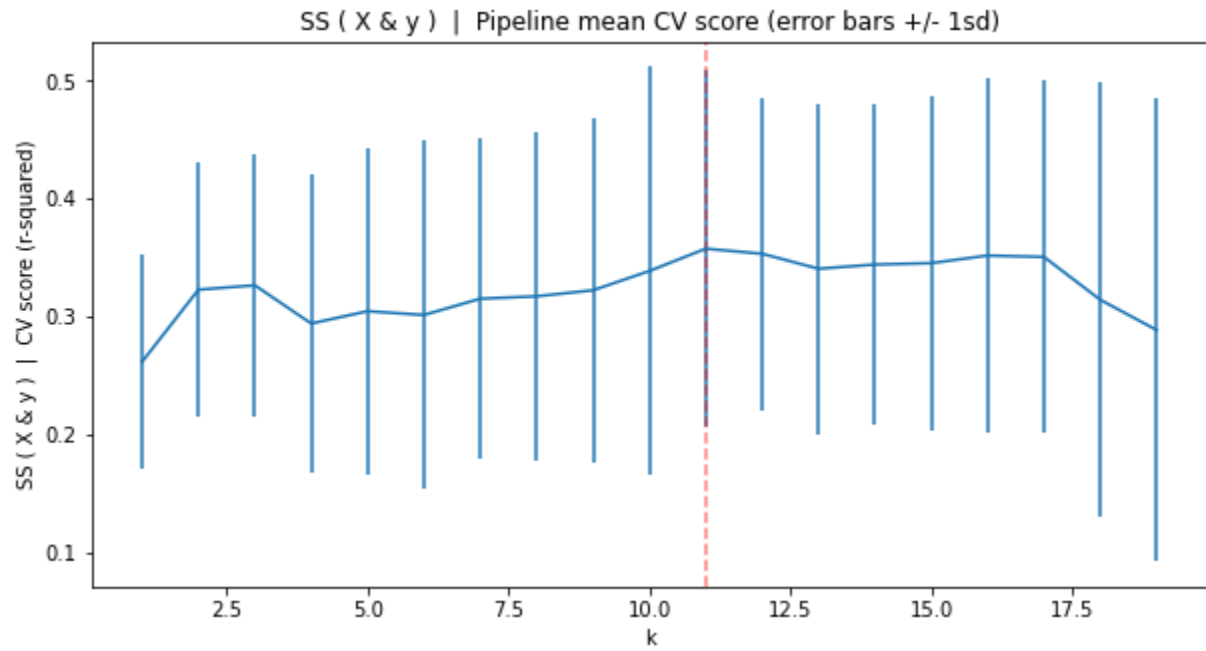
LG Train | 0.5005    Test 0.2732

R<sup>2</sup> results for the LG & SS combination below

SS Train | 0.5053    Test 0.2788

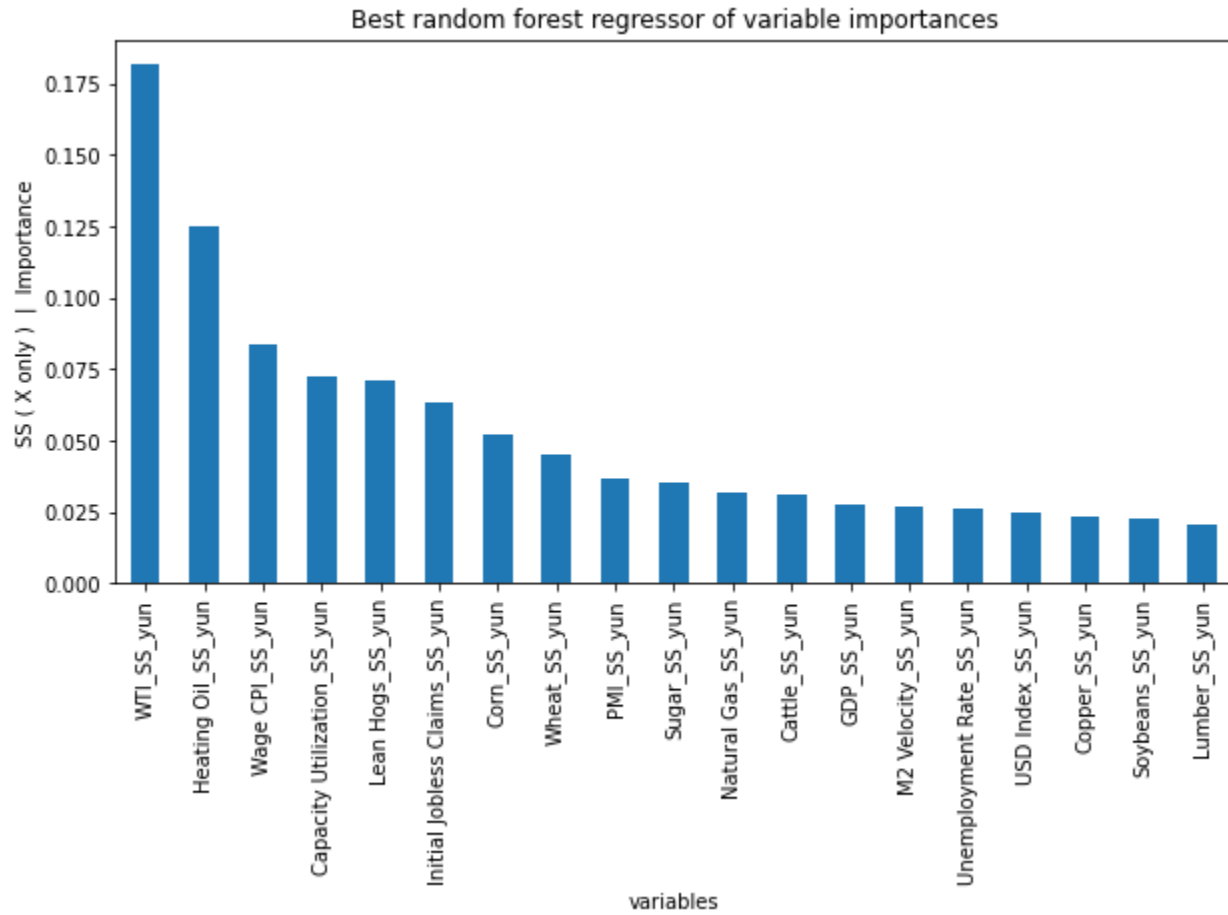
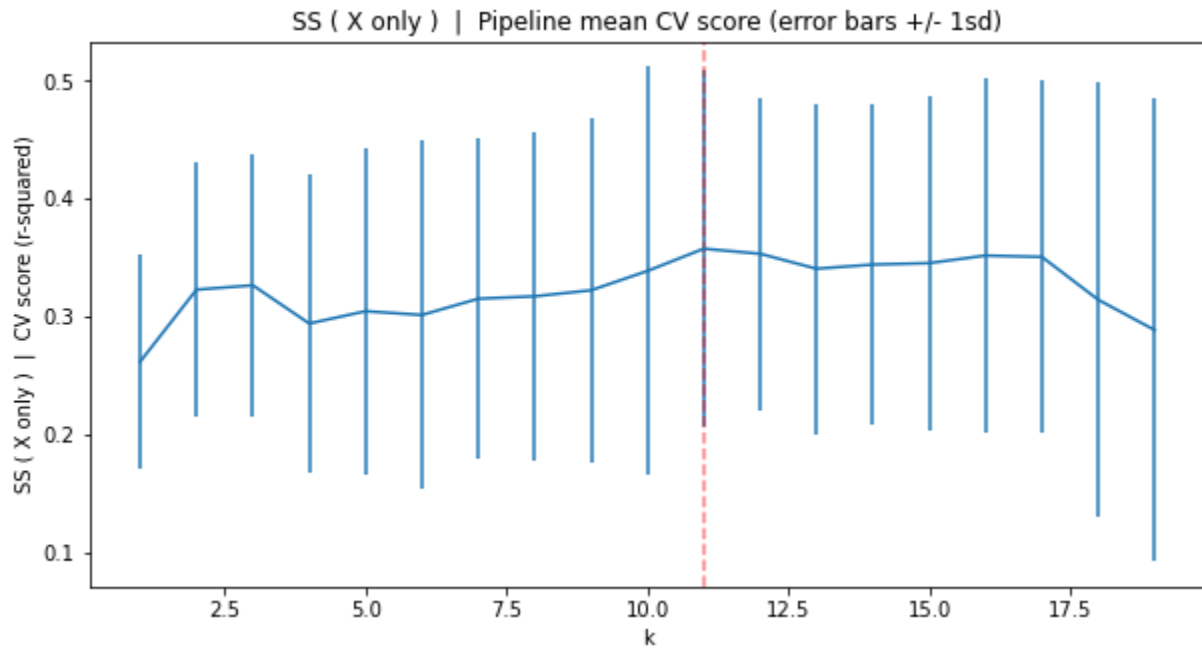
( [back](#) )

## Appendix VII



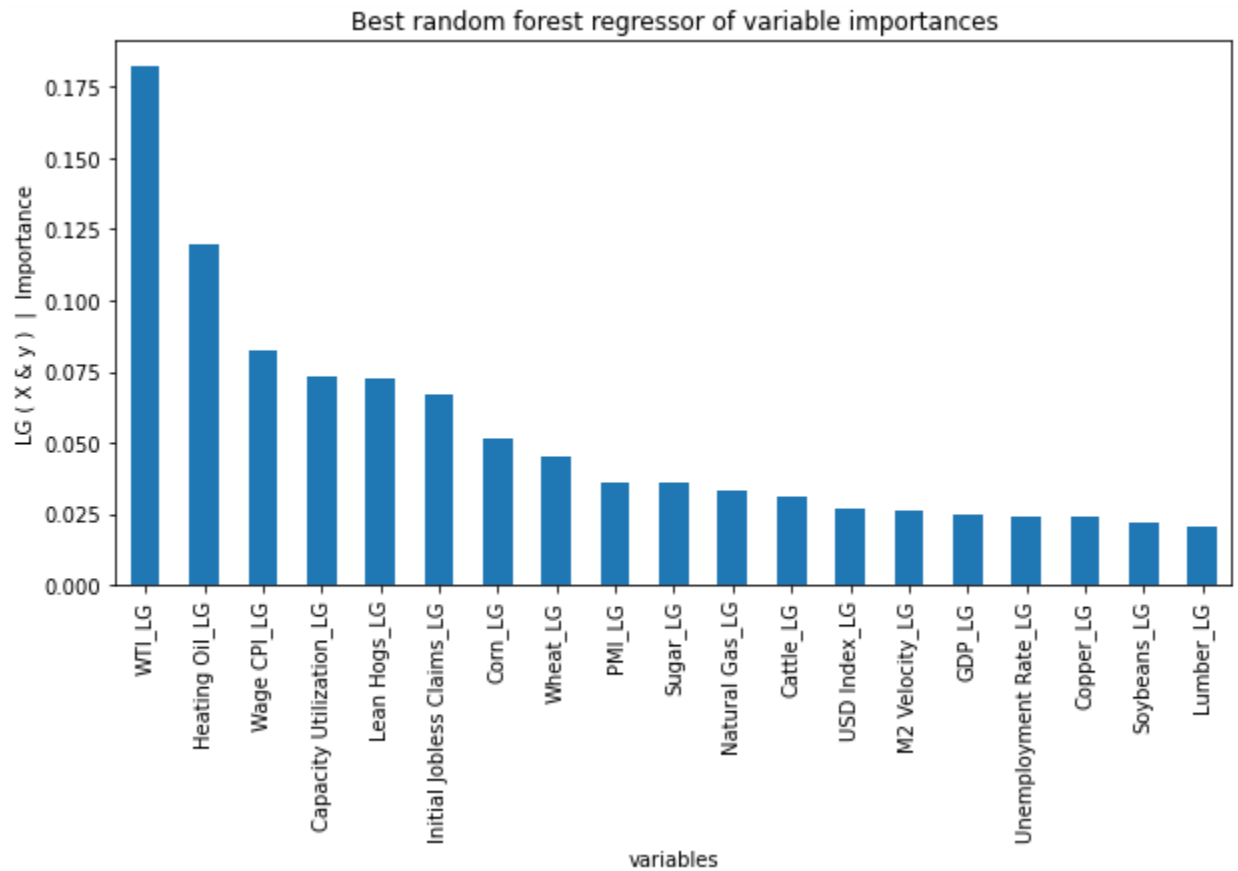
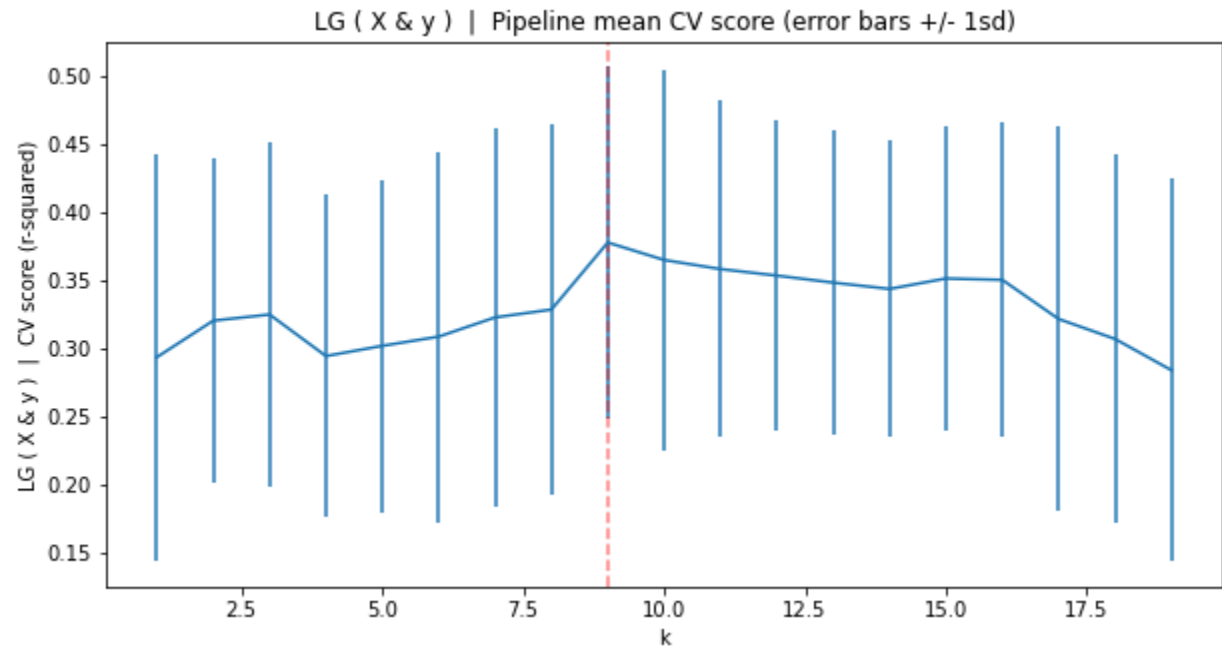
( [back](#) )

## Appendix VII



( [back](#) )

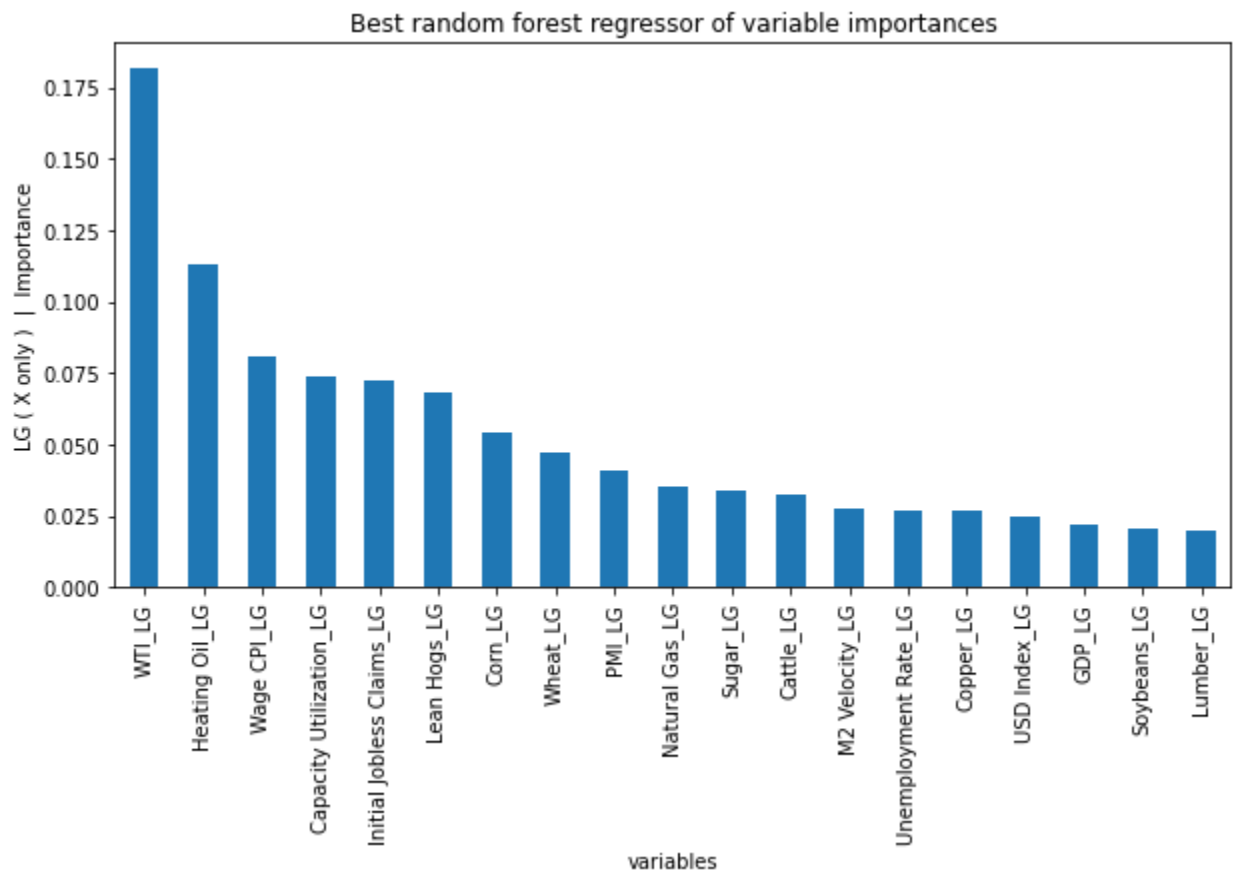
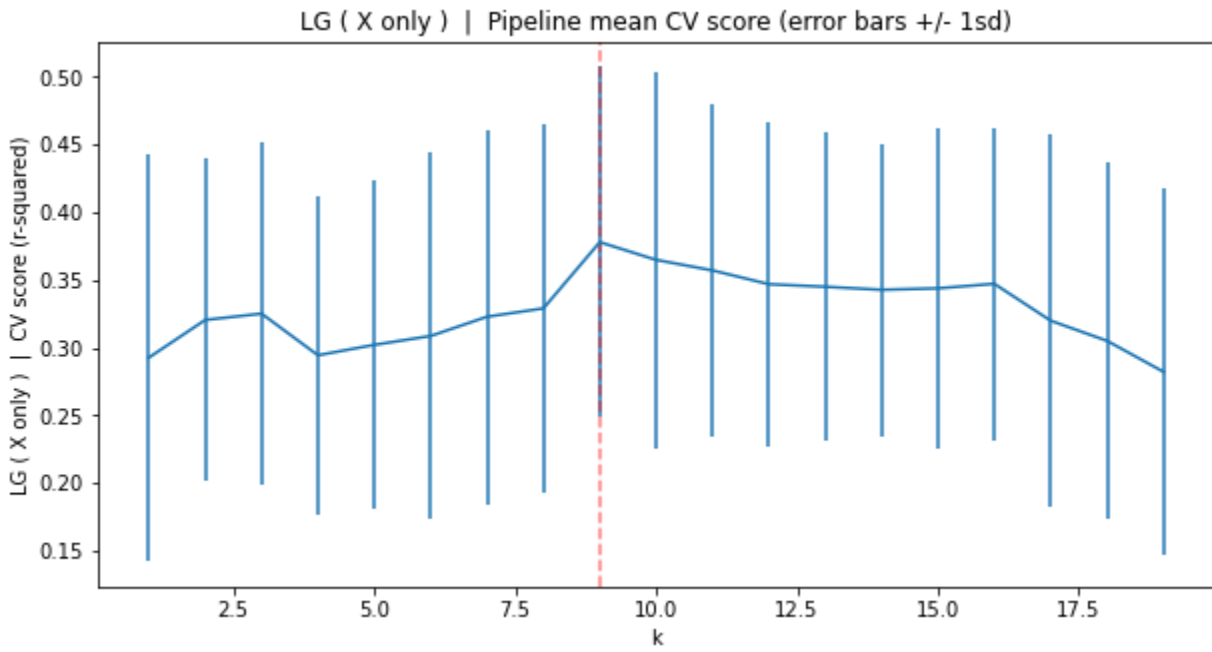
## Appendix VII



( [back](#) )

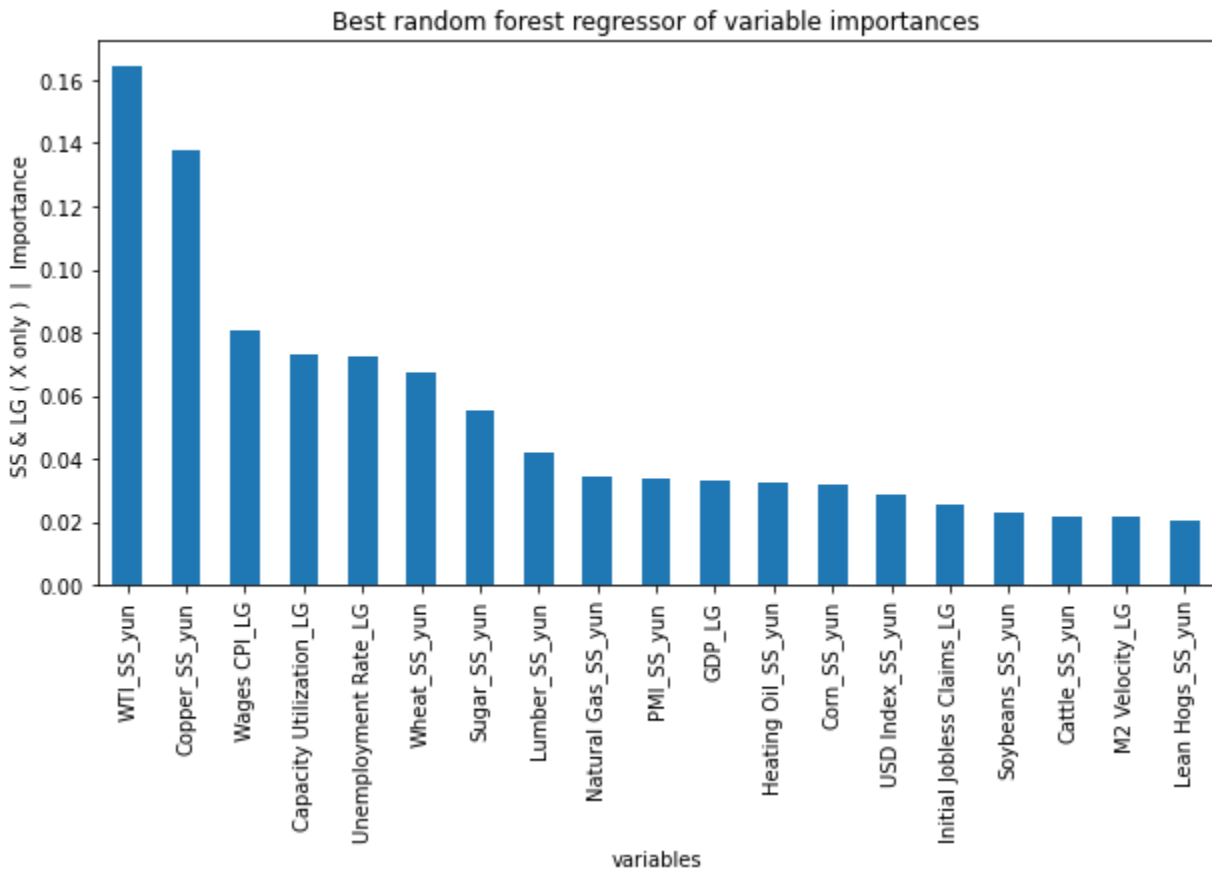
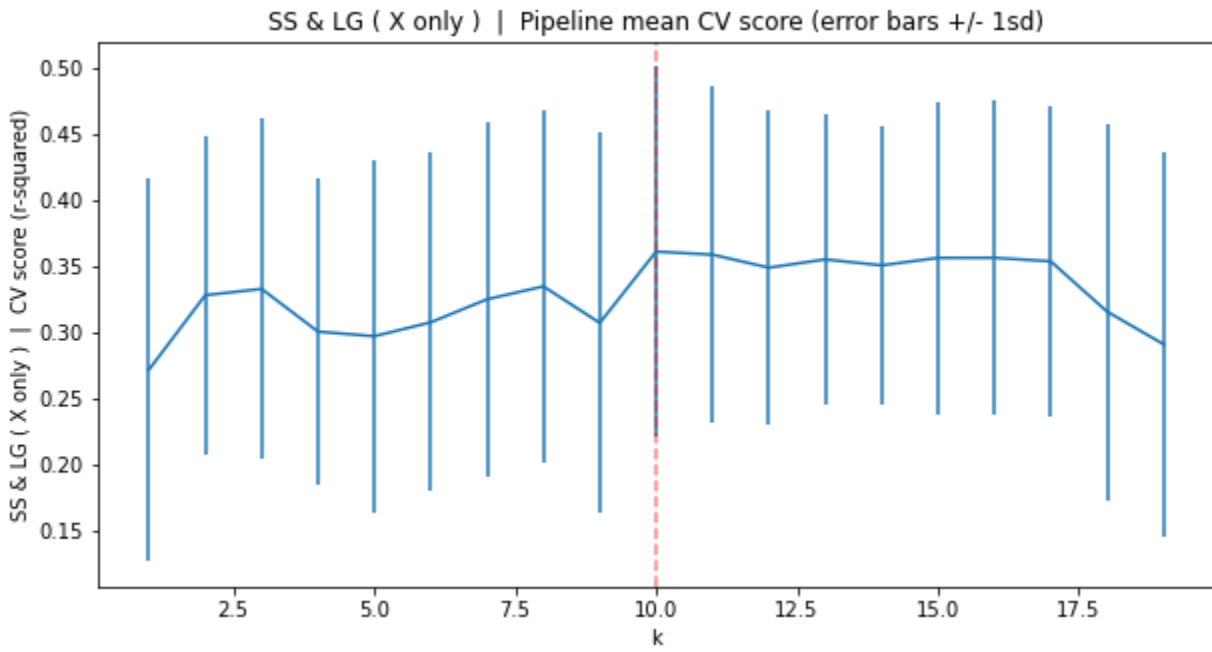


## Appendix VII



( [back](#) )

## Appendix VII



( [back](#) )