IPython Notebook Table of Contents

**BDI to Predict Coal and Freight**

* Uploaded file: coalfreightbdi.csv
  + Columns names: Date, BDI, Freight, (log\_BDI and log\_freight added later)
  + Dates from 1/5/1996 – 4/10/2015 (all x variables and y variable available)
  + Dates from 4/17/2015 – 3/25/2016 missing freight (this is what we want to predict)
  + Had to dropna so then had to put into numpy array back into pandas data frame
  + Put dates into datetime form

BDI to Predict Freight

* Analysis used on original data
  + Describe, corr, boxplot, scatterplot matrix
  + Added log\_BDI and log\_Freight so did scatterplot matrix of that too
  + Looked at basic trend graph
* Models created
  + MLR using BDI as x and Freight as y, also graphed normal probability plot
  + MLR using log\_BDI and log\_freight, also graphed normal probability plot
  + RDF using BDI to predict Freight, had to determine #trees, and depth for best model
  + RDF using log\_BDI and log\_freight, had to determine #trees, and depth for best model
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (4/17/2015 – 3/25/2016) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** freight.csv
  + Plotted each of the models vs the actual data
* Uploaded file: coalfreightbdi.csv
  + Column names: Date, BDI, Freight, (log\_BDI, and log\_coal added later)
  + Dates from 2/18/2000 - 3/25/2016 (all x variables and y variable available)
  + Dates from 1/5/1996 – 2/11/2000 missing coal (this is what we want to predict)
  + Had to dropna so put into numpy array to put back into pandas data frame
  + Put dates into datetime form
* Analysis used on original data
  + Describe, corr, boxplot, scatterplot matrix with log values too
  + Looked at basic trend graph

BDI to Predict Coal

* Models created
  + MLR using BDI as x and Coal as y, also graphed normal probability plot
  + MLR using log\_BDI as x and log\_coal as y, also graphed normal probability plot
  + RDF using BDI to predict Coal, had to determine #trees, and depth for best model
  + RDF using log\_BDI to predict log\_Coal, had to determine #trees, and depth
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (1/5/1996 – 2/11/2000) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coal2.csv
  + Plotted each of the models vs the actual data

BDI leading to predict coal

* Uploaded file: coalfreightbdi.csv
  + Column names: Date, BDI\_lead, Coal
  + Dates from 2/18/2000 - 3/25/2016 (all x variables and y variable available)
  + Analysis used on original data :Kept changing lead and lag but doing that didn’t help corr at all (did same as above)
* Uploaded file: coalfreightbdi.csv
  + Column names: Date, BDI, Coal, Freight
  + Dates from 2/18/2000 - 4/10/2015 (all x variables and y variable available)
  + Dates from 1/5/1996 – 2/11/2000 missing coal (this is what we want to predict)
  + Had to dropna so put into numpy array to put back into pandas data frame
  + Put dates into datetime form
* Analysis used on original data

BDI & Freight to predict coal

* + Corr and basic trend graph
* Models created
  + MLR using BDI and freight as x and Coal as y
  + MLR using freight as x and Coal as y
  + RDF using BDI and Freight to predict coal, had to determine #trees, and depth for best model
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (1/5/1996 – 2/11/2000) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coalall.csv
  + Plotted each of the models vs the actual data
* Uploaded file: coalfreightbdi.csv
  + Column names: Date, BDI, Freight
  + Dates from 1/3/1997 - 4/10/2015 (previous freight predicting model bad at predicting 1996 so I thought I would get rid of that year when making models)
  + Dates from 4/17/2015 – 3/25/2016 missing freight (this is what we want to predict)
  + Had to dropna so put into numpy array to put back into pandas data frame
  + Put dates into datetime form

BDI no 1996 values to predict Freight

* Analysis used on original data
  + Corr, describe, scatter plot and basic trend graph
* Models created
  + MLR using BDI as x and Freight as y, also graphed normal probability plot, and got rid of 3 outliers (did analysis again to see if anything changed significantly)
  + RDF using BDI to predict freight, had to determine best parameters
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (4/17/2015 – 3/25/2016) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** freightno96.csv
* Uploaded file: coalfreightbdi.csv
  + Column names: Date, BDI, Coal, log\_BDI, log\_Coal
  + Dates from 2/18/2000 - 2/27/2014 (only using early dates since past predicting)
  + Dates from 1/5/1996 – 2/11/2000 missing coal (this is what we want to predict)
  + Had to dropna so put into numpy array to put back into pandas data frame
  + Put dates into datetime form
* Analysis used on original data: Scatterplot matrix
* Models created

Early BDI to Predict Coal

* + MLR using BDI as x and Coal as y, also graphed normal probability plot
  + MLR using log\_BDI as x and log\_coal as y, also graphed normal probability plot
  + RDF using BDI to predict Coal, had to determine #trees, and depth for best model
  + RDF using log\_BDI to predict log\_Coal, had to determine #trees, and depth
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (1/5/1996 – 2/11/2000) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coalearly.csv
  + Plotted each of the models vs the actual data
* Uploaded file: coalfreightbdi.csv
  + Column names: Date, BDI, Coal, Freight, Dollar
  + Dates from 2/18/2000 - 4/10/2015 (dates when coal and freight present), missing (some 2005-2012 because no dollar value)
  + Dates from 1/5/1996 – 2/11/2000 missing coal (this is what we want to predict)

BDI, Dollar, Freight to Predict Coal

* + Had to dropna so put into numpy array to put back into pandas data frame
  + Put dates into datetime form
* Analysis used on original data: corr and scatterplot matrix
* Models created
  + MLR using BDI, dollar, Freight as x and coal as y, also graphed normal probability plot
  + RDF using BDI, dollar, freight to predict coal, had to determine best parameters
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (1/5/1996 – 2/11/2000) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coalearlydollarfreight.csv

BDI, Dollar\_All, Freight to Predict Coal

* Uploaded file: coalfreightbdi.csv
  + Column names: Date, BDI, Coal, Freight, Dollar\_All
  + Dates from 2/18/2000 - 4/10/2015 (dates when coal and freight present),now I have those missing dollar values I didn’t have in previous part
  + Everything else same as BDI, dollar, freight to predict coal
  + **Combined csv saved as:** newcoal.csv
* Uploaded file: coalfreightbdi.csv
  + Column names: Date, BDI, Coal, Dollar\_All
  + Dates from 2/18/2000 - 3/25/2016 (all x variables and y variable available)
  + Dates from 1/5/1996 – 2/11/2000 missing coal (this is what we want to predict)
  + Had to dropna so put into numpy array to put back into pandas data frame

BDI, Dollar\_All to Predict Coal, then freight

* + Put dates into datetime form
* Analysis used on original data: corr and describe
* Models created
  + MLR using BDI, and dollar\_all as x and coal as y
  + RDF using BDI, dollar\_all to predict coal, had to determine best parameters
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (1/5/1996 – 2/11/2000) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** newcoal.csv
* Started to predict freight but already had a good enough model so I stopped

**CF\_Urea Lead-Lag**

* Uploaded file: CFUrea.csv
  + Column names: Date, CF, Urea
  + Used dates: 8/12/2005-6/24/2016
* Initial Analysis used: correlation, rainbow scatterplot, general trend
* Model used: simple linear regression
* Looked at VAR lag selection
* Graphed both variables trend line and then below cross correlation graph
* Used lag correlation definition to see if correlation improved if one of the variables lead or lagged
* 2nd part looked at 2015 on since looked like more of a correlation looking at rainbow scatterplot and then did the same things as above

**Comparing Groups**

* Tukey’s HSD
* T-test
* Chi-Square

**Comparing Spark to Python**

* Uploaded file: project1revisit2.csv
  + Column names: latedate, Spotdollar, 2monthcrude, 2monthcorn, 2monthurea, earlydate, dollar, crude, corn, urea, crude.1, urea.1
  + More graphs found on Project 1 Revisit 2.ipynb
  + If you want to see Spark outputs for different models look at UREA Predictions.ipynb
  + Use dates from 1/5/2015 – end of that year

Spotdollar, 2month-crude, 2month-corn to predict 2month-urea 2015

* + Used Spotdollar, 2monthcrude, 2monthcorn, 2monthurea
* Analysis used on original data: describe, corr, scatterplot matrix
* Models created
  + RDF using spotdollar, 2monthcrude, 2monthcorn to predict 2monthurea, selected best parameters
* Comparing predicted to actual
  + Used RDF to predict a few days in March given we know what the x variables are
  + Looked at the error between those predicted values and what it actually was
* Technically this part of the notebook doesn’t compare RDF from scikitlearn to Pyspark
* Uploaded file: project1revisit2.csv
  + Column names: latedate, Spotdollar, 2monthcrude, 2monthcorn, 2monthurea, earlydate, dollar, crude, corn, urea, crude.1, urea.1
  + Dates used: 1/19/96 – end 2005
  + Columns used: earlydate, dollar, crude, corn, urea
* Analysis used on original data: corr, describe, scatterplot matrix, heat map
* Model used

Dollar, crude, corn to predict urea 1996-2005

* + RDF model using same number of trees and same depth as in Spark document (200 trees, and max depth = 10), dollar, crude, corn to predict urea
  + RDF model based on best parameters given by scikitlearn
* Comparing predicted to actual
  + Graph of actual vs predicted
  + Used scikitlearn RDF’s to predict future values 2/26/2016 – 3/8/2016 (not in model but I know what x values are and what urea actually is) then looked at error
* Other graphs
  + Performance vs Training size (shows test error and training error)
  + Performance vs Number of Trees (shows test error and training error)

Spot-dollar, 2month-crude, 2month-corn to predict2month-urea China

* Uploaded file: project1revisit.csv
  + Column names: latedate, Spotdollar, 2monthcrude, 2monthcorn, 2monthurea, earlydate, dollar, crude, corn, urea, latestdate, crude.1, urea.1
  + Dates used: 8/3/2015 – 2/25/2016 (China Influence)
  + Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
  + Dropna values
* Everything else the same as above except different variables used and only describe() used
* Uploaded file: project1revisit.csv
  + Column names: latedate, Spotdollar, 2monthcrude, 2monthcorn, 2monthurea, earlydate, dollar, crude, corn, urea, latestdate, crude.1, urea.1
  + Dates used: 3/1/2012 -2/25/2016 (Sec1)

2012 – 2/25/166

* + Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
  + Dropna values
* Everything else the same as above except different dates selected
* Uploaded file: project1revisit.csv
  + Column names: latedate, Spotdollar, 2monthcrude, 2monthcorn, 2monthurea, earlydate, dollar, crude, corn, urea, latestdate crude.1, urea.1
  + Dates used: 2/1/2014 -2/25/2016 (Green Markets) - weekly

Green Markets

* + Columns used: latestdate, crude.1, urea.1
  + Dropna values
* Everything else the same as above except different dates selected and only crude used to predict urea

**Corn**

* Uploaded file: corn2.csv
  + Column names: year, state, planted, acres treated with N, acres treated with P, acres treated with K, N applied, P applied, K applied, PDSI, Precip, Avg temp
  + The data was taken from USDA survey data and the weather was taken from NOAA
* Grouping
  + Showed how to group data by state two ways
* Iowa Corn All
  + Used all columns (all variables)
  + Analysis used on Iowa data
    - Describe, corr, spearman corr, heat map, scatterplot matrix,boxplot, general trend graph, made subplot comparing all lines on trend (so plot for each variable)
* Uploaded file: planted.csv
  + Column names: year, state, commodity, Agdistrict, agdistrictcode, county, cornacresplanted,corngrainacresharvested, temp, precipitation
* Iowa Planted vs harvested
  + Grouping
    - Average amount planted grouped by district
    - Average amount harvested grouped by district
    - Data grouped by year and district in one
    - In the end compared Kossuth county to Emmet county
  + Graphs
    - On two separate plots for each of the counties graphed cornacresplanted, corngrainacresharvested, temp, and precipitation
    - Then made subplot, Kossuth on one side and Emmet on other and included plot for the variables listed above
  + Analysis used
    - Boxplots, scatterplot matrices, heat maps, used machine learning to graph variable importance
  + Models created
    - 2 MLRs (both counties) using cornacresplanted, temp, and precipitation to predict corngrainacresharvested
    - 2 MLRs (both counties) using cornacresplanted to predict corngrainacresharvested

**Customer Segmentation**

* Uploaded file: segmentation.csv
  + Columns used: parent name, region, 2013 urea, 2013 UAN, 2013 ammonia, 2013 amm sulfate, 2013 DAP, 2013 MAP-MAP-50-MES, 2013 TSP, 2013 10-34-00, 2013 potassium, 2013 dry blends, 2013 other liq blends, 2013 specialty products, 2013 liquid blends, 2013 LILLY, 2013 total, 2013 9-Box
    - Same but for 2014 and 2015 too
    - In addition Nitrogen 2015, Phosphate 2015, Potassium 2015, other, specialty
    - 65 columns all together
* Grouping
  + Grouped all variables by year so 3 different segments
  + Then each segment grouped total by regions
  + Then each segment grouped total by 9 box colors
* Uploaded file: seg2.csv
  + Columns used: Parent Name, Region Account Manager, year, Urea, UAN, Ammonia, AmmSulfate, DAP, MAP-MAP-50-MES, TSP, 10-34-00, Potassium, DryBlends, OtherLiqBlends, SpecialtyProducts, LiquidBlends, LILLY, Total, MarginScore TotalStrat,,MarginStrat, 9-Box
* Grouping
  + Grouped by parent name, year, and total
  + Sorted values by largest total and margin score (descending)
  + Sorted values by 9 box color so that means would be GreenGreen at top since the previous sorting put large total and large margin score at top

**DAP Model**

* Uploaded file: UANDAP.csv
  + Column names: Date, Dollar, Crude, Corn, Gas, UAN, DAP, Urea, Wheat, Transportation index
  + Columns used: Date, Dollar, Crude, Corn, Gas, DAP
  + Dates used: 3/1/2012 – 3/29/2016
* Analysis used on original data: boxplot, describe, corr, heat map, scatterplot matrix, trend graph
* Models used
  + MLR model: x variables = dollar, crude, corn, gas ; y variable = DAP. Normal probability plot included.
  + MLR model: x variables = Dollar, corn, gas ; y variable = DAP. Normal probability plot included.
  + MLR model: x variables = crude, corn, gas; y variable = DAP. Normal probability plot included.
  + MLR model: x variables = dollar, crude, corn; y variable = DAP.
  + LR model: x variable = corn; y variable = DAP
  + RDF model: x variables = dollar, crude, corn, gas; y variable = DAP, needed to find best parameters
  + RDF model: x variables = corn, gas; y variable = DAP, needed to find best parameters
  + RDF model: x variables = dollar, corn, gas; y variable = DAP, needed to find best parameters
  + RDF model: x variables = crude, corn, gas; y variable = DAP, needed to find best parameters
  + RDF model: x variable = corn; y variable = DAP, needed to find best parameters
* Comparing predicted to actual
  + Graph of actual vs predicted of all models
  + Used scikitlearn RDF’s to predict known values since I didn’t put future values to 2020 yet (I plan on it) then looked at error

**DAP-MAP Predictions May**

* Lead or Lag in PCS and Mosaic Stock
  + Uploaded file: ureamorevariables.csv
    - Dates used: 1/5/1996-5/27/2016
    - Columns used: Date, DAP, PCS\_Stock, Mosaic\_Stock
  + Initial data analysis: scatterplot matrix, rainbow scatterplots, general trends
  + MLR models used: both stocks, then each stock individually to DAP
  + Rolling mean and standard deviation graph
  + VAR analysis
    - Results showed 5 week lag of DAP, and did same analysis as above
  + Cross Correlation
    - Graphed two stocks and the cross correlation graph
    - Used definition that shows different lags and best correlation, which showed 7 weeks lead of both stocks so did same analysis but with DAP up 7 weeks
  + Looked at percentage change but that is currently a lost cause
* Exploratory Analysis (including stock info and no lead or lag)
  + Uploaded file: ureamorevariables.csv
  + Columns used: Date, Urea, Phos\_Inv, Phos\_Exports, Phos\_Domestic\_Use, PCS\_Stock, Mosaic\_Stock, DAP
  + Dates used: 1/5/1996-5/27/2016
  + Initial analysis: describe, boxplot, corr, heatmap, scatterplot matrix, general trend
  + Models used:
    - MLR1: all variables using statsmodels
    - MLR2: all variables using scikitlearn
    - MLR3: all variables using LASSO
    - MLR4: x variables = urea, PCS\_stock, Mosaic\_Stock (RFECV choice)
    - Model 1: RDF all variables
    - Model 2: RFECV/top3, x variables = urea, pcs\_stock, mosaic\_stock
    - Model 3: Top 4, so same as model 2 but added phos\_inv
    - Model 4: no exports, same as model 3 but add domestic\_use
    - Model 5: no correlation between x variables, x variables = phos\_inv, phos\_exports, phos\_domestic\_use, mosaic\_stock
  + Could not use these models to predict DAP yet since don’t have predictions for stocks
* Exploratory Analysis with Stocks Leading DAP by 7 Weeks
  + Moved DAP up 7 weeks so dates used = 1/5/1996-4/8/2016
  + Initial Analysis: corr
  + Models Used:
    - MLR1: all variables using statsmodels
    - MLR2: no exports
    - MLR3: Top 3/ RFECV, x variables = urea, pcs\_stock, mosaic\_stock
    - MLR4: no correlation among x variables x variables = phos\_inv, phos\_exports, phos\_domestic\_use, mosaic\_stock
    - Model 1: RDF all variables
    - Model 2: RFECV/top3, x variables = urea, pcs\_stock, mosaic\_stock
    - Model 3: Top 4, so same as model 2 but added phos\_inv
    - Model 4: no exports, same as model 3 but add domestic\_use
    - Model 5: no correlation between x variables, x variables = phos\_inv, phos\_exports, phos\_domestic\_use, mosaic\_stock
  + Could not use these models to predict DAP ye since don’t have predictions for stocks
* Exploratory Analysis (no stocks, no lead or lag of DAP)
  + Initial analysis: corr
  + Models used:
    - MLR1: all variables using statsmodels
    - MLR2: no exports
    - MLR3: RFECV, x variables = urea
    - MLR4: Top 2, x variables = urea and phos\_inv
    - Model 1: RDF all variables
    - Model 2: No domestic use
    - Model 3: No exports
    - Model 4: no correlation, x variables = urea, phos\_domestic\_use
  + Then predicted future values of DAP to 12/25/2020 for the avg, best and worst case (also included graph that showed predictions.
    - Uploaded files for the 3 cases were DAPmayavg, DAPmaybest, DAPmayworst
    - Output files: DAPmayavgresults, DAPmaybestresults, DAPmayworstresults

**DAP-Predictions May Using UAN**

* This notebook uses assumptions made in the DAP-MAP Predictions May one
* Exploratory Analysis (no lag, stocks included)
  + Uploaded file: ureamorevariables.csv
  + Dates used: 1/5/1996-5/27/2016
  + Columns used: Date, UAN, Phos\_Inv, Phos\_Exports, Phos\_Domestic\_Use, PCS\_Stock, Mosaic\_Stock, DAP
  + Initial analysis: corr
  + Models used:
    - MLR1: all variables using statsmodels
    - MLR2: forward selection, x variables = uan, mosaic\_stock, phos\_inv, phos\_domestic\_use
    - MLR3: RFECV, x variables = uan, pcs\_stock, mosaic\_stock
    - Model 1: RDF all variables
    - Model2: Same variables as MLR2 forward selection
    - Model3: top3/RFECV, x variables = uan, mosaic\_stock, pcs\_stock
    - Model4: no correlation with UAN to other x variables, x variables = uan, phos\_domestic\_use
  + No DAP predictions since no future predictions for stocks at this time
* Exploratory Analysis with Stocks Leading DAP by 7 Weeks
  + Since there is a shift the dates used are 1/5/1996-4/8/2016
  + Initial analysis: corr
  + Models used:
    - MLR1: all variables using statsmodels
    - MLR2: forward selection, x variables = pcs\_stock, mosaic\_stock, uan, phos\_inv
    - MLR3: RFECV, x variables = uan, pcs\_stock, mosaic\_stock
    - MLR4: no correlation to uan, x variables = phos\_domestic\_use, pcs\_stock
    - Model 1: RDF all variables
    - Model 2: RFECV/Top3, x variables = uan, pcs\_stock, mosaic\_stock
    - Model 3: Top4/Forward selection, x variables = uan, phos\_inv, pcs\_stock, mosaic\_stock
    - Model 4: x variables = phos\_domestic\_use and pcs\_stock
  + Can’t use these models to predict DAP yet since no predictions for stocks
* Exploratory Analysis (no stocks, no lead or lag of DAP)
  + Dates used: 1/5/1996-5/27/2016
  + Initial analysis: corr
  + Models used:
    - MLR1: all variables using statsmodels (no stocks)
    - MLR2: forward selection, x variables = uan, phos\_inv, phos\_exports
    - MLR3: RFECV, x variable = uan
    - MLR4: no correlation among x variables, x variables = uan, phos\_domestic\_use
    - Model 1: RDF all variables used
    - Model 2: No domestic use, forwards selection/top3
    - Model 3: RFECV/ just uan
    - Model 4: No correlation among x variables, x variables = uan, phos\_domestic\_use
  + Then predicted future values of DAP to 12/25/2020 for the avg, best and worst case (also included graph that showed predictions.
    - Uploaded files for the 3 cases were DAPmayavg, DAPmaybest, DAPmayworst
    - Output files: DAPmayavgresultsusinguan, DAPmaybestresultsusinguan, DAPmayworstresultsusinguan

**Helpful Python Scripts**

* Basics about executing code, opening notebook, print difference between Python 2 & 3
* Uploading files, csv in particular
  + Printing out head and shape
* Grouping and finding certain attribute
  + Ex.) looking for a particular state, group amount by state
* General trend graph: not using ax format, just using plt
* Descriptive statistics& boxplot
* Correlation and heat map
* Scatterplots and scatterplot matrix
* Histograms
* Subplots
* OLS Regression and Outlier Diagnostics
* Machine learning: random forest, feature importance bar graph, training & test data graph or residuals vs predicted values

**International Template**

* Uploaded file: InternationalUrea.csv
  + Column names: date, dollar, crude, corn, gas, Coal\_RDF(MLR), Coal\_RDF(RDF), UreaPrillYuz, UANRouen, UreaPrillChina, UreaGranEgypt, UreaGranBrazil
  + Dates used: 8/1/2013 – 3/31/2016
* Analysis used on original data: describe, corr, trend graph
* Urea Prill Yuz
  + Columns used from file: date, dollar, crude, corn, gas, Coal\_RDF(MLR), Coal\_RDF(RDF), UreaPrillYuz
  + Dates used: both 2014 and 2016 and later plan on separating those years
  + Dropna, put into numpy array, put back into pandas data frame
  + Put date into datetime format
  + Analysis on data
    - Scatterplot matrix, corr, and boxplot
  + Models used
    - MLR model: x variables = dollar, crude, corn, gas, coal\_RDF(MLR); y variable = UreaPrillYuz. Normal probability plot included.
    - MLR model: x variables = dollar, crude, corn, gas, coal\_RDF(RDF); y variable = UreaPrillYuz.
    - MLR model: x variables = dollar, crude, gas, coal\_RDF(MLR); y variable = UreaPrillYuz
    - MLR model: x variables = dollar, crude, gas, coal\_RDF(RDF); y variable = UreaPrillYuz
    - MLR model: x variables = dollar, corn, gas, coal\_RDF(MLR); y variable = UreaPrillYuz
    - MLR model: x variables = dollar, corn, gas, coal\_RDF(RDF); y variable = UreaPrillYuz
    - RDF model: x variables = dollar, crude, corn, gas, coal\_RDF(MLR), y variable = UreaPrillYuz, needed to find best parameters
    - RDF model: x variables = dollar, crude, corn, gas, coal\_RDF(RDF), y variable = UreaPrillYuz, needed to find best parameters
  + Comparing models
    - I didn’t get rid of coal but I plan to do that
    - Graph of actual vs predicted
  + Then I just looked at 2016 since 2015 there was a gap and looked at data, dollar, ureaprillyuz
    - Describe, boxplot, and correlation
* UreaGranEast, UreaPrillChina, UreaGranEgypt, UreaGranBrazil, UANRouen
  + So far I only used dollar for x variable and those corresponding for y variables
  + Analysis used: describe, boxplot, trends, corr
  + Models used
    - MLR model: x variable = dollar, y variable = above listed. Normal probability plot included
    - RDF model: x variable = dollar, y variable = above listed. Needed to find best parameters.
  + Comparing models
    - Graph actual vs predicted
    - Found test and train errors, no predicting into future yet
  + Looked at just 2016 since no 2015
    - Describe, boxplot, corr

**Machine Learning**

* Uploaded file: project1revisit2csv.csv
  + Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, precipitation, temp, 2monthurea
  + Dates used: 3/1/2012 – 12/31/2015
* Analysis used on original data: scatterplot matrix, heat map
* Machine learning
  + RDF model: x variables spotdollar, 2monthcrude, 2monthcorn, precipitation, temp; y variable = 2monthurea
  + Training data and testing data, in residual vs predicted value graph
  + Bar graph of random forest importance
  + Gradient boosting regressor graph
  + # misclassified samples and accuracy
  + Test accuracy for logistic regression
  + K-fold cross validation for logistic regression
  + 3 hypothesis: linear svm, rbf SVC, poly SVC
  + MAE
  + Density plot or all variables
  + Training and cross validation on graph of accuracy vs alpha parameter
  + GBT partial dependence plots

**Monthly Import Predictions (Models use data from 96-Mar 25th 2016) to 2020**

* Uploaded file: ureamarchupdateavg2.csv
  + Column names/used: Date, Dollar, Crude, Corn, Gas, Coal\_RDF(MLR), Coal\_RDF(RDF), Urea\_RDF(MLR), Urea\_RDF(RDF), Monthly Imports
    - The (MLR) uses the freight values from the MLR model, needed freight to predict coal, therefore to predict urea too
  + Dates used: 1/5/1996 - 3/25/2016
  + Put date into datetime format
* Analysis used on original data: describe, corr, scatterplot matrix
* Models used
  + MLR model: x variables = dollar, crude, corn, gas, Coal\_RDF(MLR), Urea\_RDF(MLR); y variable = monthly imports
  + RDF model: x variables = dollar, crude, corn, gas, Coal\_RDF(MLR), Urea\_RDF(MLR); y variable = monthly imports. RDF model: x variables = dollar, crude, corn, gas, Coal\_RDF(RDF), Urea\_RDF(RDF); y variable = monthly imports.

Average Case

* Comparing models
  + Didn’t include predictions using MLR model since it was bad
  + Predicted monthly imports using (MLR) and (RDF) values using RDF models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (3/26/2016 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** monthlyimportsavgmarch2.csv
  + Had to change combined into numpy array back into pandas dataframe so that it went from 0-2797 with no starting over at 0 in prediction part
  + Split data into months
  + Found std deviation of the models so I could put that on the graph
  + Graphed actual vs predicted monthly of each model
  + Graphed actual vs predicted daily of each model
* Uploaded file: ureamarchupdatebest2.csv
  + Column names/used: Date, Dollar, Crude, Corn, Gas, Coal\_RDF(MLR), Coal\_RDF(RDF), Urea\_RDF(MLR), Urea\_RDF(RDF), Monthly Imports
    - The (MLR) uses the freight values from the MLR model, needed freight to predict coal, therefore to predict urea too
  + Dates used: 1/5/1996 - 3/25/2016
  + Put date into datetime format
* Analysis used on original data: describe, corr, scatterplot matrix
* Models used
  + RDF model: x variables = dollar, crude, corn, gas, Coal\_RDF(MLR), Urea\_RDF(MLR); y variable = monthly imports.
  + RDF model: x variables = dollar, crude, corn, gas, Coal\_RDF(RDF), Urea\_RDF(RDF); y variable = monthly imports.

Best Case

* Comparing models
  + Didn’t include predictions using MLR model since it was bad
  + Predicted monthly imports using (MLR) and (RDF) values using RDF models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (3/26/2016 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** monthlyimportsbestmarch2.csv
  + Had to change combined into numpy array back into pandas dataframe so that it went from 0-2797 with no starting over at 0 in prediction part
  + Split data into months
  + Found std deviation of the models so I could put that on the graph
  + Graphed actual vs predicted monthly of each model
  + Graphed actual vs predicted daily of each model
* Uploaded file: ureamarchupdateworst2.csv
  + Column names/used: Date, Dollar, Crude, Corn, Gas, Coal\_RDF(MLR), Coal\_RDF(RDF), Urea\_RDF(MLR), Urea\_RDF(RDF), Monthly Imports
    - The (MLR) uses the freight values from the MLR model, needed freight to predict coal, therefore to predict urea too
  + Dates used: 1/5/1996 - 3/25/2016
  + Put date into datetime format
* Analysis used on original data: describe, corr, scatterplot matrix
* Models used
  + RDF model: x variables = dollar, crude, corn, gas, Coal\_RDF(MLR), Urea\_RDF(MLR); y variable = monthly imports.
  + RDF model: x variables = dollar, crude, corn, gas, Coal\_RDF(RDF), Urea\_RDF(RDF); y variable = monthly imports.
* Comparing models

Worst Case

* + Didn’t include predictions using MLR model since it was bad
  + Predicted monthly imports using (MLR) and (RDF) values using RDF models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (3/26/2016 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** monthlyimportsworstmarch2.csv
  + Had to change combined into numpy array back into pandas dataframe so that it went from 0-2797 with no starting over at 0 in prediction part
  + Split data into months
  + Found std deviation of the models so I could put that on the graph
  + Graphed actual vs predicted monthly of each model
  + Graphed actual vs predicted daily of each model

**More Efficient way to do Notebooks**

* Now I know how to split up in months without having to figure out each month chunk, so less code

**Most recent working RSI**

* Ways to get stock values from Yahoo finance
  + fetch\_historical\_yahoo
  + web.DataReader
* Types of candlestick graphs using Yahoo finance data
  + Candlestick2\_ohlc
  + Plot\_day\_summary2\_ochl (this one looks nicer)
* Subplotting close and volume
  + Print min, max, and average of the volume
  + Subplot of close above volume, sharing x axis
* Moving average graphs
  + Graph of close, ma50, ma200
  + Subplot of moving avg graph above volume graph
* Interactive graph which lets you enter Yahoo ticker
  + Fancy graph (dark graph)
  + Has MACD, RSI, volume, close, etc.
* Graphs that pulls from Excel file (applicable in future)
  + File needs to have date, open, high, low, close, volume, adj\_close columns
  + Graph shows moving average, rsi, MACD, price, volume (matplotlib format)
* Comparing
  + I compared the csv file graph to the matplotlib format one to make sure you get the same graph and I did

**New**

* Uploaded file: project1revisit2.csv
  + Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
  + Dates used: 3/1/2012 – 2/25/2016
  + Dropna()
* Bokeh.plotting (plot that you can zoom in on, etc)
* Analysis used on original data: describe, scatterplot matrix, kde plot matrix, heat map
* Models used
  + RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
* Analysis on RDF model
  + Print model accuracy
  + Compare values to predict2.csv file that has some values
  + Print feature importance and graph
  + Calculate error of those predicted values to the actual
  + Then found the best parameters using script and did everything again, in addition added graph of actual vs predicted
* Other graphs
  + Random forest performance vs training size (test error and training error), 200 trees, depth = 6
  + Random forest performance vs training size (200 trees, depth =10)
  + RDF performance vs number of trees (test error and training error)

**Predictions for Coal used to predict urea-mar25**

* Uploaded file: bdifreightdollarpredictcoalavg.csv
  + Column names/used: Date, BDI, Freight\_MLR, Freight\_RDF, Dollar\_All, Coal
  + Dates used: 2/18/2000 - 3/25/2016
  + Put date into datetime format
* Analysis used on original data: describe
* Models used
  + MLR model: x variables = BDI, Freight\_MLR, Dollar\_All; y variable = coal
  + MLR model: x variables = BDI, Freight\_RDF, Dollar\_All; y variable = coal
  + RDF model: x variables = BDI, Freight\_MLR, Dollar\_All; y variable = coal. Had to determine best parameters.
  + RDF model: x variables = BDI, Freight\_RDF, Dollar\_All; y variable = coal. Had to determine best parameters

Average Case

* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (3/26/2016 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coalavg2.csv
* Uploaded file: bdifreightdollarpredictcoalbest.csv
  + Column names/used: Date, BDI, Freight\_MLR, Freight\_RDF, Dollar\_All, Coal
  + Dates used: 2/18/2000 - 3/25/2016
  + Put date into datetime format
* Analysis used on original data: describe

Best Case

* Models used
  + RDF model: x variables = BDI, Freight\_MLR, Dollar\_All; y variable = coal. Had to determine best parameters.
  + RDF model: x variables = BDI, Freight\_RDF, Dollar\_All; y variable = coal. Had to determine best parameters
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (3/26/2016 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coalbest2.csv
* Uploaded file: bdifreightdollarpredictcoalworst.csv
  + Column names/used: Date, BDI, Freight\_MLR, Freight\_RDF, Dollar\_All, Coal
  + Dates used: 2/18/2000 - 3/25/2016
  + Put date into datetime format
* Analysis used on original data: describe
* Models used

Worst Case

* + RDF model: x variables = BDI, Freight\_MLR, Dollar\_All; y variable = coal. Had to determine best parameters.
  + RDF model: x variables = BDI, Freight\_RDF, Dollar\_All; y variable = coal. Had to determine best parameters
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (3/26/2016 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coalworst2.csv

**Predictions for Coal used to predict urea**

* Uploaded file: bdifreightdollarpredictcoalavg.csv
  + Column names/used: Date, BDI, Freight\_MLR, Freight\_RDF, Dollar\_All, Coal
  + Dates used: 2/18/2000 – 4/10/2015
  + Put date into datetime format
* Analysis used on original data: describe
* Models used
  + MLR model: x variables = BDI, Freight\_MLR, Dollar\_All; y variable = coal

Average Case

* + RDF model: x variables = BDI, Freight\_MLR, Dollar\_All; y variable = coal. Had to determine best parameters.
  + RDF model: x variables = BDI, Freight\_RDF, Dollar\_All; y variable = coal. Had to determine best parameters
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (4/17/2015 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coalavg.csv
* Uploaded file: bdifreightdollarpredictcoalbest.csv
  + Column names/used: Date, BDI, Freight\_MLR, Freight\_RDF, Dollar\_All, Coal
  + Dates used: 2/18/2000 – 4/10/2015
  + Put date into datetime format
* Analysis used on original data: describe
* Models used

Best Case

* + RDF model: x variables = BDI, Freight\_MLR, Dollar\_All; y variable = coal. Had to determine best parameters.
  + RDF model: x variables = BDI, Freight\_RDF, Dollar\_All; y variable = coal. Had to determine best parameters
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (4/17/2015 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coalbest.csv
* Uploaded file: bdifreightdollarpredictcoalworst.csv
  + Column names/used: Date, BDI, Freight\_MLR, Freight\_RDF, Dollar\_All, Coal
  + Dates used: 2/18/2000 – 4/10/2015
  + Put date into datetime format
* Analysis used on original data: describe
* Models used

Worst Case

* + RDF model: x variables = BDI, Freight\_MLR, Dollar\_All; y variable = coal. Had to determine best parameters.
  + RDF model: x variables = BDI, Freight\_RDF, Dollar\_All; y variable = coal. Had to determine best parameters
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (4/17/2015 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** coalworst.csv

**Predictions for Freight to use to predict coal-Copy 1**

* Uploaded file: BDIdollarpredictfreightavg.csv
  + Column names: date, BDI, corr, Dollar\_All, Freight
  + Dates used: 1/5/1996 – 4/10/2015
  + Put into datetime format
* Analysis used on original data: corr
* Models used
  + MLR model: x variables = BDI and dollar\_all; y variable = freight. Got model from “BDI to Predict Coal & Freight ipynb”

Average Case

* + RDF model: x variables = BDI and Dollar\_All; y variable = freight. Got best parameters from same ipynb listed above.
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (4/17/2015 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** freightdollavg.csv
* Uploaded file: BDIdollarpredictfreightbest.csv
  + Column names: date, BDI, corr, Dollar\_All, Freight
  + Dates used: 1/5/1996 – 4/10/2015
  + Put into datetime format

Best Case

* Analysis used on original data: corr
* Models used
  + MLR model: x variables = BDI and dollar\_all; y variable = freight. Got model from “BDI to Predict Coal & Freight ipynb”
  + RDF model: x variables = BDI and Dollar\_All; y variable = freight. Got best parameters from same ipynb listed above.
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (4/17/2015 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** freightdollbest.csv
* Uploaded file: BDIdollarpredictfreightworst.csv
  + Column names: date, BDI, corr, Dollar\_All, Freight
  + Dates used: 1/5/1996 – 4/10/2015
  + Put into datetime format
* Analysis used on original data: corr
* Models used
  + MLR model: x variables = BDI and dollar\_all; y variable = freight. Got model from “BDI to Predict Coal & Freight ipynb”

Worst Case

* + RDF model: x variables = BDI and Dollar\_All; y variable = freight. Got best parameters from same ipynb listed above.
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Then added the prediction part (4/17/2015 – 12/31/2020) of every model (no error since not applicable) to original data
  + **Combined csv saved as:** freightdollworst.csv

**Project 1 Revisit 2**

* Uploaded file: project1revisit2.csv
  + Column names: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea, earlydate, dollar, crude, corn, urea, latestdate, crude.1, urea.1
  + Dates: 3/1/2012 – 2/25/2016 or 1/1/1996 – 12/31/2005
* Uploaded file: project1revisit2.csv
  + Columns used: earlydate, dollar, crude, corn, urea
  + Dates used: 1/1/1996 – 12/31/2005

1996-2005

* Analysis used on original data: describe, boxplot, corr, scatterplot matrix
* Models used
  + MLR model: x variables = dollar, crude, corn; y variable = urea. Normal probability plot included, and outlier diagnostics
* Uploaded file: project1revisit2.csv
  + Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
  + Dates used: 3/1/2012 – 2/25/2016
  + Dropna

Section 1 3/1/12 – 2/25/16

* Analysis used on original data: trend graph, describe, boxplot, corr, spearman corr, heat map, scatterplot matrix, scatterplots, histograms
* Models used
  + MLR model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
  + MLR model: x variables = 2monthcrude, 2monthcorn; y variable = 2monthurea. Normal probability plot included
  + MLR model: x variables = spotdollar, 2monthcorn; y variable = 2monthurea. Normal probability plot included and outlier detection
  + MLR model: x variables = spotdollar, 2monthcorn; y variable = 2monthurea. Got rid of the outliers found in previous model.
* Uploaded file: project1revisit2.csv
  + Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
  + Dates used: 8/1/2015 – 2/25/2016
  + Dropna
* Analysis used on original data: trend graph, describe, boxplot, corr, spearman corr, heat map, scatterplot matrix, scatterplots, histograms

China Influence 8/1/15 – 2/25/16

* Models used
  + MLR model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea. Normal probability plot included and outlier detection.
  + MLR model: x variable = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea. Got rid of the outliers found in the previous model and normal probability plot included.
  + MLR model: x variables = spotdollar, 2monthcrude; y variable = 2monthurea. Normal probability plot included
* Uploaded file: project1revisit2.csv
  + Columns used: latestdate, crude.1, urea.1
  + Dates used: 2/1/2014 - 2/25/2016

Green Markets

* Analysis used on original data: trend graph, describe, boxplot, corr, scatterplot matrix, scatterplot, histogram
* Models used
  + MLR model: x variable = crude.1, y variable = urea.1. Normal probability plot and outlier detection included.
  + MLR model: x variable = crude.1, y variable = urea.1. Got rid of outliers of previous model and normal probability plot is included.
* Uploaded file: project1revisit2csv.csv
  + Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea, precipitation, temp
  + Dates used: 3/1/2012 - 12/31/2015
  + dropna
* Analysis used on original data: trend graph, describe, boxplot, corr, spearman corr, heatmap, scatterplot matrix, scatterplots, histograms

Prec & Temp Included

* Models used
  + MLR model: x variables = spotdollar, 2monthcrude, 2monthcorn, precipitation, temp; y variable = 2monthurea. Normal probability and outlier detection included.
  + MLR model: x variables = spotdollar, 2monthcrude, temp; y variable = 2monthurea. Normal probability plot included.
  + MLR model: x variables = spotdollar, 2monthcrude, 2monthcorn, precipitation, temp; y variable = 2monthurea. Got rid of outliers.
  + MLR model: x variables = spotdollar, precipitation, temp; y variable = 2monthurea. Got rid of outliers.
  + MLR model: x variables = spotdollar, 2monthcorn, precipitation, temp; y variable = 2monthurea. Got rid of outliers
  + MLR model: x variables = spotdollar, 2monthcorn, temp; y variable = 2monthurea. Got rid of outliers.
  + MLR model: x variables = spotdollar, 2monthcorn, precipitation; y variable = 2monthurea. Got rid of outliers.
  + MLR model: x variables: 2monthcrude, 2monthcorn, temp; y variable = 2monthurea. Got rid of outliers.

**Project1revisit**

* Uploaded file: project1revisitcsv.csv
  + Column names: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea, earlydate, dollar, crude, corn, urea
  + Dates used: 3/1/2012 – 2/25/2016 or 1/1/1996 – 12/31/2005
* Analysis used on original data: scatterplots, scatterplot matrix, describe, corr, boxplot, histogram
* Models used
  + MLR model: x variables = dollar, corn, crude; y variable = urea. Normal probability plot, outlier detection, and partial regression plots included.
* Other random things
  + Heat map with no corr values
  + Filling na values with the mean

**Quandl**

* Certain stock/database: Quandl.get
* Getting weekly stocks of GOOG from 1/1/2001 – 1/1/2010 (open, high, low, close, volume)
* Comparison of prices for Microsoft to Apple (close values)
* Import and export values for US phosphate fertilizers
* Getting Dubai, Brent, Nigerian Forcados, West Texas Intermediate crude values
  + Filter out just Brent
* Percentage change between Apple and Microsoft close values (column 4)

**Random stuff from book**

* Guassian matrix
* Vectors and arrays
  + Multiplication
  + Slicing and indexing
* Iris Data
  + Describe feature names
  + Multicolored scatterplot matrix
  + Covariance matrix and comparing that to PCA
    - PCA and variation of PCA for big data graphs
* Housing data
  + KNN regression
* Boston data
  + Outlier detection
    - Z-score (abs>3)
    - Using PCA and elliptic envelope (fits hypothetical guassian distribution) graph
    - Using oneclass SVM graph
* Scoring functions (performance of model)
  + Multilabel classification (when your task is to predict more than a single label)
  + Decision tree classifier
  + Confusion matrix
  + Precision, accuracy, etc output
  + Binary classification (ex. Male or female categories)
  + Mean absolute error, mean squared error, R^2
* Testing and validating
  + Linear SVC
  + RBF SVC
  + Poly SVC
* Cross-validation
  + Use just for optimization, it just points to the best possible algorithm and parameter choice on the basis of the default (best average result), however can change default
* Linear and Logistic regression
  + Precision, recall, f1 score, support
  + Look at Boston data
* Word tokenization
* Subplot and density plot
* Learning curve visualization and cross-validation
  + Accuracy vs sample size (training and cross-validation lines)
  + Accuracy vs alpha parameter (training and cross-validation lines)
* Random Forest
  + Feature importance and histogram
  + GBT partial dependence plot

**Regression Basics**

* Simple Linear Regression
  + Uploaded file: CrudeUreaCorrelation.csv
  + Created model using statsmodels
  + Scatterplots created: one that shows R^2 and opacity, rainbow one
  + Mentioned cointegration vs correlation
* Multiple Linear Regression
  + Uploaded file: ureamorevariables.csv
  + Columns used: Date, Dollar, Crude, Corn, Gas, My\_Coal, Monthly Imports, Urea\_Prod, Urea\_Inventory, UAN\_Inventory, UAN, Urea
  + Models used: scikitlearn MLR, statsmodels MLR, forward stepwise regression, RFECV
  + Diagnostics/Result Outputs:
    - Print parameters, standard errors, and predicted values
    - Plot of normalized residuals squared by leverage
    - Multicollinearity tests
    - Heteroscedascity tests
    - Cointegration heat map
* Ridge and LASSO Regression
  + Uploaded file: ureamorevariables.csv
  + Columns used: Dollar, Crude, Corn, My\_Coal, Gas, Urea\_Inventory, UAN\_Inventory
  + Models used:
    - Used scikitlearn for LASSO and LassoCV, LassoLarsCV
    - Used scikitlearn for Ridge, BayesianRidge (included choose best alpha)
  + Plots created:
    - Regression Coefficients Progression for Lasso paths
    - Mean Squared error on each fold
* GLSAR
  + Uploaded file: ureamorevariables.csv
  + Columns used: Date, Dollar, Crude, Corn, Gas, My\_Coal, Urea\_Inventory, UAN\_Inventory, UAN, Urea
  + GLSAR uses statsomodel package, also looked at log model too
* VAR
  + Used data from the packages themselves, printed data results
  + Graphs created:
    - Sample autocorrelation
    - Impulse responses
    - Forecasts
    - Forecast error variance decomposition
  + Lag order selection
* Polynomial Features
  + Have yet to explore in detail so nothing really there

**SQLite**

* Connect to sqlite
* Looked at tutorial database and created a table where I later selected and joined certain things

**Stuff Learning from Safari Books**

* Modeling in Predictive Analytics with Python and R
  + Analysis of movie taglines, couldn’t get to work since I don’t have upload file
  + Game day simulator for baseball
* Mastering Python in Finance
  + Call methods:
    - Bisection method
    - Newton’s method
    - Secant method
    - Brent’s method
  + Definition for storing common attributes of a stock quote, using binomial European Option
  + Definition that lets you choose a European or an American Option
  + Definition that prices an option by the leisen-reimer tree
  + Definition that computes option price, delta, and gamma by the LR tree
  + Definition that prices an option by the Boyle trinomial tree
  + Definition that gets implied volatility from LR binomial tree using bisection method
  + Looking at dividends and bootstrapping the yield curve
* Learning Predictive Analytics with Python
  + Calculated VIF, cross validation and ROC curve, clustering
* Marketing data Analysis
  + Graph of product positioning of movies
* Python Business Intelligence Cookbook
  + Cumulative distribution
* Mastering Python Data Visualization
  + Gross Returns graph
  + Max profit graph
  + Birthday problem
* Learning Data Mining with Python
  + Feature selection using variance threshold
* Regression Analysis with Python
  + Eigenvalues
  + SGD classifier with printed test set accuracy, confusion matrices, and train set accuracy
* Python for data science for dummies
  + RFECV
* Web Scraping in Python
  + Definitions that retrieves a list of all internal links found on a page and retrieves a list of all external links found on page
* Mastering Matplotlib
  + Heat Map with Histogram Mean Monthly Precipitation (it didn’t print but I saved figure of how it looks)

**UAN Model**

* Uploaded file: UANDAP.csv
  + Columns used: date, dollar, crude, corn, gas, UAN
  + Dates used: 3/1/2012 – 3/29/2016
  + Dropna to numpy array back into pandas dataframe
  + Put date into datetime format
* Analysis used on original data: boxplot, describe, corr, heat map, scatterplot matrix, trend graph
* Models used
  + MLR model: x variables = dollar, crude, corn, gas; y variable = UAN. Normal probability plot included
  + MLR model: x variables = dollar, corn, gas; y variable = UAN. Normal probability plot included
  + MLR model: x variables = crude, corn, gas; y variable = UAN. Normal probability plot included
  + RDF model: x variables = dollar, crude, corn, gas; y variable = UAN. Needed to find best parameters
  + RDF model: x variables = dollar, crude, corn; y variable = UAN. Needed to find best parameters
  + RDF model: x variables = dollar, corn, gas; y variable = UAN. Needed to find best parameters
  + RDF model: x variables = crude, corn, gas; y variable = UAN. Needed to find best parameters
* Comparing models
  + Added to the original data the predictions for each of the models and each of their errors
  + Graphed predicted vs actual for all the models
  + Eventually will be able to use this model to predict future UAN model

**UAN to end of April Prediction (Crude, Corn, Gas, Coal, UAN Inventory)**

* Exploratory Analysis
  + Uploaded file: ureamorevariables.csv
  + Columns used: Date, Crude, Corn, Gas, My\_Coal, UAN\_Inventory, UAN
  + Dates used: 1/5/1996 – 4/30/2016
  + Initial analysis: boxplots, describe, corr, heatmap, scatterplot matrix, trend graph
  + MLR models, forward selection selected by AIC one by one then looked at RDF models
  + Models Used:
    - MLR\_All: all variables included
    - MLR\_NoInv: all variables but uan\_inventory
    - RDF\_All: all variables included
    - RDF\_NoCrude: no crude since highly correlated to other variables
    - RDF\_NoCrudeCorn: no crude or corn in this model
* Predictions Part
  + Used 3 different cases: avg, best, worse
    - Uploaded files: ureaaprilavgrandomness, ureaaprilbestrandomness, ureaaprilworstrandomness
  + Dates used in prediction part: 5/1/2016 – 12/31/2020
  + Daily output files: uanaprilavgresultsrandomness, uanaprilbestresultsrandomness, uanaprilworstresultsrandomness
  + Then split data up into months and output files: uanaprilavgresultsmonthlyrandomness, uanaprilbestresultsmonthlyrandomness, uanaprilworstresultsmonthlyrandomness
  + Created graphs for predicted values for each case

**UAN to end of April Prediction (Everything)**

* Exploratory Analysis
  + Uploaded file: ureamorevariables.csv
  + Columns used: Date, Dollar, Crude, Corn, Gas, My\_Coal, Monthly Imports, Urea\_Prod, Urea\_Inventory, Urea, UAN\_Inventory, UAN
  + Dates used: 1/5/1995-4/30/2016
  + Initial analysis: boxplots, describe, corr, heatmap, scatterplot matrix, trend
  + MLR models found using forward stepwise regression, one by one selction based off of AIC
  + RDF Models looked at MSE and R^2 to select
  + Models Used:
    - UAN\_MLR\_Urea: use urea to predict uan
    - UAN\_MLR\_Forward: x variables = urea, my\_coal, urea\_prod, urea\_inventory, corn, and gas,
    - UAN\_MLR\_Top4: x variables = urea, my\_coal, urea\_prod, urea\_inventory
    - RDF\_All: all variables included
    - RDF\_Top3: x variables = urea, my\_coal, corn
    - RDF\_LeastCorr: x variables = urea, my\_coal, urea\_inventory, uan\_inventory, gas
    - RDF\_CoalGas\_UANInv: x vairables = my\_coal, gas, uan\_inventory
    - RDF\_UreaGas: x variables = urea and gas
* Predictions Part
  + To go any further I need to first predict urea

**UREA Predictions**

* Uploaded file from Hadoop: project2nolabel
  + Once connected using Pyspark, counted # values
  + Printed off first row of values
* Select dollar, crude, corn, and urea columns
  + Then say x values are all but urea, and urea is y
* Split data into training and test datasets
  + Count test and train data
* Train a random forest model
  + Print MSE and save the Random forest model as **olderurea4**
* Load RDF file and **predict2.csv** which has values for predictions we want to get
* Predict the prediction file values using RDF and print out values
  + Look at error between actual and predicted values
* Repeated all the things above except then used dates from 8/3/2015 – 2/25/2016 and will compare to MLR too
* Repeated again but use all of section 1 and compared to MLR too
* Compared all models and plotted the predicted vs actual and compared errors of the models by graphing both
* Summarized errors at the bottom

**Weather, Choropleth, Sentiment Analysis Investing**

* Weather from wunderground, just need to know zipcode you are interested in
* Choropleth maps most of the things I tried failed so lots of empty outputs, except Texas Unemployment
* Python and Pandas for Sentiment Analysis and Investing
  + Graph of single stock price and moving average
  + Definitions for outlier fixing, single stock auto MA, position, and backtest
  + Output of holdings, funds, starting funds, current valuation, strategy percent growth
* Stock Market Predictions with Time Series Estimators
  + Created a time series estimator.py
    - However couldn’t figure out how to download pickle files so couldn’t use it
* Automated Trading in Python
  + Graphed close, MA, signal, position, and return using stock data from Quandl
  + Created a backtest also
  + Used ffn to get stock stats like start, end, risk free rate, total return, daily sharpe, CAGR, max drawdown, MTD, 3 month, 6 month, YTD, 1y,3y,5y,10y, mean, volume, skew, kurtosis, best day, worst day
* Other
  + Look at articles of clothing people buy together
  + Text analysis

**Web Scraping, Clustering, Movie Sentiment, Plotly**

* Web Scraping
  + Use beautifulsoup for scraping stock data from yahoo finance
  + Use googlemaps API look at different distances to find routes however output doesn’t make sense
* Customer Segmentation
  + Uploaded file: Wholesalecustomersdata.csv
  + Cluster data based on meanshift
  + Graph created: Centroids of clusters for milk and groceries
* Clustering with Our Data
  + Used 2015 customer segmentation from survey, the only problem is that everything needs to be in number format which it is not so I need to figure that out yet
* Movie Sentiment
  + Use nltk to extract words to see if positive or negative
  + Use naïve bayes classifier to see how accurate the classifier made is in determining if comments were positive or negative in nature
* Plotly
  + Used data from the site for choropleth graphs
  + Graphed time series with range slider and selectors for Mosaic
  + Graphed a scatterplot that shows density and has histograms for each of the variables
  + Graphed a bubble plot
  + Graphed a candlestick chart

**Random ipynb folder**

* Untitled 1: different ways to format dates
* untitled0: a python script that gets you connected to ARMS of USDA
* untitled3: a python script that uses the API to pull data from ARMS of USDA

**Urea Cases folder**

* Average Case China Model-2020
  + Uploaded file: averagerevisit3.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 8/3/2015 – 2/29/2016
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2020) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted
* Average Case Revisit 3
  + Uploaded file: averagerevisit3.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, demanddaily, 2monthurea
    - Dates used: 3/1/2012 – 2/29/2016
    - Dropna to numpy array back to pandas dataframe
    - Date to datetime format
  + Analysis used on original data: describe, dtypes, boxplot, corr, scatterplot matrix, kde matrix, trend of urea and demand
  + Did same things as above but put demand in monthly format
  + Graphed actual monthly avg vs monthly demand
* Avg Case-2020
  + Uploaded file: averagerevisit3.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 3/1/2012 – 2/29/2016
    - Dropna to numpy array to pandas dataframe
    - Date to datetime format
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea. Best parameters found in **“Avg case” ipynb**
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2020) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted
* Avg Case
  + Uploaded file: sec1avgcase.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 3/1/2012 – 2/29/2016
    - Dropna to numpy array to pandas dataframe
    - Date to datetime format
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea. Best parameters found in **“Avg case” ipynb**
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2016) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted
* Best Case China Model-2020
  + Uploaded file: bestrevisit3.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 8/3/2015 – 2/29/2016
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2020) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted
* Best Case China Model
  + Uploaded file: sec1bestcase.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 8/3/2015 – 2/29/2016
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2016) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted
* Best Case-2020
  + Uploaded file: bestrevisit3.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 8/3/2015 – 2/29/2016
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2020) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted
* Best Case
  + Uploaded file: sec1bestcase.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 8/3/2015 – 2/29/2016
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2016) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted
* **Urea Model Using My Predictions, crude, corn, coal**
  + Uploaded file: ureamarchupdateavg2.csv
    - Columns used: date, crude, corn, Coal\_RDF(MLR), Coal\_RDF(RDF), urea
    - Dates used: 1/5/1996 – 3/25/2016
    - Date to datetime format
  + Analysis used on original data**:** describe
  + Models used
    - MLR model: x variables = crude, corn, coal\_RDF(MLR); y variable = urea
    - MLR model: x variables = crude, corn, coal\_RDF(RDF); y variable = urea

Avg Case

* + - RDF model: x variables = crude, corn, coal\_RDF(MLR); y variable = urea. Best parameters found using “Urea model, monthly import model”
    - RDF model: x variables = crude, corn, coal\_RDF(RDF); y variable = urea. Best parameters found using “Urea model, monthly import model”
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Then added the prediction part (3/26/2016 – 12/31/2020) of every model (no error since not applicable) to original data
    - **Combined saved as**:ureaavgmarch2.csv
    - Combined grouped into months
    - Graphed all models monthly actual vs predicted
    - Graphed all models daily actual vs predicted
  + Uploaded file: ureamarchupdatebest2.csv
    - Columns used: date, crude, corn, Coal\_RDF(MLR), Coal\_RDF(RDF), urea
    - Dates used: 1/5/1996 – 3/25/2016
    - Date to datetime format
  + Analysis used on original data**:** describe
  + Models used
    - MLR model: x variables = crude, corn, coal\_RDF(MLR); y variable = urea
    - MLR model: x variables = crude, corn, coal\_RDF(RDF); y variable = urea
    - RDF model: x variables = crude, corn, coal\_RDF(MLR); y variable = urea. Best parameters found using “Urea model, monthly import model”
    - RDF model: x variables = crude, corn, coal\_RDF(RDF); y variable = urea. Best parameters found using “Urea model, monthly import model”

Best Case

* + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Then added the prediction part (3/26/2016 – 12/31/2020) of every model (no error since not applicable) to original data
    - **Combined saved as**:ureabestmarch2.csv
    - Combined grouped into months
    - Graphed all models monthly actual vs predicted
    - Graphed all models daily actual vs predicted
  + Uploaded file: ureamarchupdateworst2.csv
    - Columns used: date, crude, corn, Coal\_RDF(MLR), Coal\_RDF(RDF), urea
    - Dates used: 1/5/1996 – 3/25/2016
    - Date to datetime format
  + Analysis used on original data**:** describe
  + Models used
    - MLR model: x variables = crude, corn, coal\_RDF(MLR); y variable = urea
    - MLR model: x variables = crude, corn, coal\_RDF(RDF); y variable = urea

Worst Case

* + - RDF model: x variables = crude, corn, coal\_RDF(MLR); y variable = urea. Best parameters found using “Urea model, monthly import model”
    - RDF model: x variables = crude, corn, coal\_RDF(RDF); y variable = urea. Best parameters found using “Urea model, monthly import model”
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Then added the prediction part (3/26/2016 – 12/31/2020) of every model (no error since not applicable) to original data
    - **Combined saved as**:ureaworstmarch2.csv
    - Combined grouped into months
    - Graphed all models monthly actual vs predicted
    - Graphed all models daily actual vs predicted
* Urea Model, Monthly Import Model
  + Uploaded file: ureamorevariables.csv
    - Columns used: date, dollar, crude, corn, gas, coal, monthly imports, urea
    - Dates used: 1/5/1996 – 3/25/2016
    - Put date in datetime format
  + Analysis on original data: boxplot, describe, corr, heat map, scatterplot matrix, trend graph
  + Models used
    - MLR model: x variables = dollar, crude, corn, gas, coal, monthly imports; y variable = urea
    - MLR model: x variables = dollar, crude, corn, gas, coal; y variable = urea. Normal probability plot included.
    - MLR model: x variables = dollar, gas, monthly imports; y variable = urea. Normal probability plot included.
    - MLR model: x variables = crude, gas, monthly imports; y variable = urea. Normal probability plot.

Predict Urea

* + - MLR model: x variables = corn, gas, monthly imports; y variable = urea. Normal probability plot included.
    - MLR model: x variables = coal, gas, monthly imports; y variable = urea. Normal probability plot included.
    - MLR model: x variables = dollar, crude, corn, gas, coal; y variable = urea. Normal probability plot included.
    - MLR model: x variables = crude, corn, coal; y variable = urea. Normal probability plot included.
    - RDF model: x variables = dollar, crude, corn, gas, coal, monthly imports; y variable = urea. Needed to find best parameters.
    - RDF model: x variables = dollar, crude, corn, gas, coal; y variable = urea. Needed to find best parameters.
    - RDF model: x variables = crude, corn, coal; y variable = urea. Needed to find best parameters.
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months
  + Uploaded file: ureamorevariables.csv
    - Columns used: date, dollar, crude, corn, gas, coal, urea, monthly imports
    - Dates used: 1/5/1996 – 3/25/2016
    - Date to datetime format

Predict Monthly Imports

* + Analysis used on original data: heat map,
  + Models used
    - RDF model: x variables = dollar, crude, corn, gas, coal, urea; y variable = monthly imports. Needed to find best parameters.
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months
    - Looked to see if any months similar (means for each month)
    - Graphed to see if any lags in monthly imports
    - Graphed monthly actual vs predicted of monthly imports
    - Graphed daily actual vs predicted of monthly imports
  + Like the urea model repeated but used my coal predictions instead
  + Then like Monthly imports model repeated but used coal, urea and dollar for x variables
* Worse Case China Model-2020
  + Uploaded file: worserevisit3.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 8/3/2015 – 2/29/2016
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2020) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted
* Worse Case-2020
  + Uploaded file: worserevisit3.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 8/3/2015 – 2/29/2016
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2020) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted
* Worse Case
  + Uploaded file: sec1worstcase.csv
    - Columns used: latedate, spotdollar, 2monthcrude, 2monthcorn, 2monthurea
    - Dates used: 8/3/2015 – 2/29/2016
  + Models used
    - RDF model: x variables = spotdollar, 2monthcrude, 2monthcorn; y variable = 2monthurea
  + Comparing models
    - Added to the original data the predictions for each of the models and each of their errors
    - Split data up into months and found std dev
    - Graphed monthly actual vs predicted
    - Then added the prediction part (3/1/2016 – 12/31/2016) to RDF model (no error since not applicable) to original data
    - Split data up into months again
    - Graphed monthly actual vs predicted, now have future prediction section
    - Graphed daily actual vs predicted