Best greenfield location for new U.S. refinery

Capstone II Project

Aibek Uraimov

1 The Problem Statement

Develop a data-driven methodology to validate economic feasibility study of building and operating a small (under 50kBpd) crude oil refinery in the U.S., using datasets from the U.S. Census and EIA.gov websites. Identify 2-3 key parameters that have high impact on the economic performance of a refinery, located in the pre-selected PAD-2-2 district, optimizing for anticipated gross margin ($/Bbl crude oil refined), based on at least 8 years of historical data.

1.1 Context

Green Light Co. (GLC), a California startup, wishes to validate its preliminary economical model for building a greenfield (i.e., 'completely new construction') crude oil refineries in the U.S. It recognizes that no new refinery with a significant downstream capacity has been built in the U.S. in 40 years; the existing refinery plants are relying on outdated technology to produce petroleum products. Refineries are shutting down and some are 60-70 years old.

The recent technological advancements in petroleum refining make it possible to build small-size operations at strategic locations, making products cheaper for consumers, while meeting the EPA requirements that are getting more stringent each year. The Green Light Co.'s approach is to utilize the latest technology in its refinery projects, i.e.: processing light API crude oil, reducing pollutant emissions by 7-8 times and greenhouse by 50% of the U.S. average.

GLC preliminary has selected several potential locations near major sources of domestic crude oil supply and theorizes that these locations will bring the highest profit margins. I developed a methodology that could be used to forecast the margins, based on the geographical location of a proposed refinery site, quality of crude oil (feedstock), plant equipment configuration, and proximity to destination markets.

1.2 Criteria for success

I ran a comparative analysis of data on demand and supply of energy in the US, by designated petroleum market districts (PADDs), makeup of the energy production and consumption rates, historical feedstock, and refinery product prices by markets – to validate a theory on correlation of the sustainable profitability of a greenfield refinery in the U.S. and its proposed location.

The criterion for success is to validate the anticipated gross profit margins in the range of $25-$30 per 1 U.S. barrel of crude oil processed, given the pre-selected sites for new refinery construction, type of crude available, and proximity to destination markets.

2 About Data

2.1 Key data sources

I have used datasets obtained from the US Census and EIA.gov websites to identify key metrics:

- major metro areas and specific region population
- historical energy production and consumption rates
- historical energy source composition dynamics (oil & gas, solar, hydro, nuclear, etc.)
- refinery operational throughput dynamics (to determine trends over the last 10 years)
- refinery product yields and utilization rates - to assess their historical profitability
- crude oil actual (refinery) crack-spreads by type of refining configurations,
- finally, predictive price modeling for ready products, in relation to crude oil price indices

2.2 Constraints

Some of the constraints on data availability and level of accuracy I had to accept were as follows:

- Much of the actual refinery sale prices for ready products is not publicly available data; there are agencies that sell historical petroleum pricing data, but their costs were prohibitive for an academic exercise; we will just have to rely on the EIA.gov data that is the next best thing
- Great level of uncertainty in the future of gasoline markets due to politically driven decisions by the U.S. government, which may have a negative impact on the U.S. energy sector, e.g., the 'Carbon Free by 2035' plan
- There's uncertainty on the gasoline and possibly, diesel demand, not due to politics and government, but the free market, i.e., an anticipated proliferation of Electrical Vehicles within the next 10-15 years – the minimum runway for refinery project investment returns
- Finally, the COVID-19 related lockdowns have demonstrated that a significant office-based workforce might successfully transition to remote work attendance practices, leading to lower commute and possibly, lower consumption of petroleum-based fuels

2.3 Data requirements:

My initial thoughts on the data required were as follows; however, I abandoned many items, as they were either not very useful for narrowing down the solution scope, or were becoming too large of a task, i.e., outside of the stated scope:

- Categorization of all raw data into designated sections by geographical regions, such as:

    - PADD – Petroleum Administration for Defense Districts (PADDs) are geographic aggregations of the 50 States and the District of Columbia into five districts
    - States and major metro areas
    - Such categorization was required to optimize processing of energy data, which is customarily is sourced and classified by PADDs

- Historical data on energy production and consumptions - we will use a 5-year datasets:
    - Crude Oil production + Net Imports and Crude Oil price indices - WTI for the U.S., and Brent for the global markets, to determine correlation with the prices of petroleum products produced by refineries and net imports
    - Historical refinery operations parameters: average plant utilization rates across the U.S., product yields by input crude and refinery configuration types, and volumes produced - to determine indicative profit margins for the existing refineries and build comparative analysis
    - (abandoned) Make-up of energy produced and consumed by source - to determine where the sustainable production of crude oil based refineries are the strongest and to analyze the patterns of consumption by regions
    - (abandoned) Amount of energy generated, as well as its net imports into the U.S., measured in British Thermal Units
    - (abandoned) Amount of energy consumed in the U.S., also measured in British Thermal Units

3 Data Wrangling

3.1 Raw Data

I had downloaded and inspected all datasets as listed in '2.3 Data requirements' section above and after having reviewed and assessed their usefulness and appropriateness for the task at hand, narrowed the raw datasets to the following lists:

- Active U.S. refineries with location, operating capacity, and PAD designation
- U.S. states and major metropolitan areas with 2010-2019 population data
- Distances by road from U.S. refineries to major metro areas and state capitals
- Operating and idle capacities of U.S. refineries by process and equipment types, in barrels processed per streaming and per calendar days
- Monthly Refined Product Yields by % Volume, by PAD and sub-PAD regions in 2013-2020
- U.S. refinery monthly production rates – by main product types (various grades and types of gasoline, diesel, fuel oil, propane, etc.) in 2013-2021
- Ready petroleum products wholesale resale prices by U.S. refineries by PAD and sub-PAD in 2013-2020
- Crude API by refinery PAD districts, indicative of type of crude processed in 2013-2020
- Crude oil pricing – monthly average market indices for WTI (West Texas Index) and Brent (Europe) 2013-2020
- Crude oil acquisition pricing by U.S. refineries – to calculate the implied crude purchase discount or premiums, using WTI monthly averages for domestic supply and Brent for imported oil

3.2 Wrangling

3.2.1 Cleaning

The datasets required minimal cleaning and mostly involved separating multi-level data fields from aggregate and sub-total groups, e.g., crude API data in was presented at top (U.S.), middle (PAD district), and lowest (PAD sub-district) levels and needed extracting into distinct categories to conduct proper analysis.

The datasets had negligible number of missing values and were mostly of the refinery resale pricing. There's a variety of seasonal specifications, under which products are made and hence, they will only be sold and have the pricing data in the winter or summer periods – I combined them under same product category.

The little amount of missing pricing data, especially for small volume production, which was not always reported to EIA.gov, were imputed using pandas 'fillna()' function.

3.2.2 Combining, melting, merging datasets

I identified the common categorical features among the datasets to combine and merge on with each other. The most common categorical feature was geographical data – specifically, the State, City, PADD district, and PAD sub-district.

Other data fields that were used to combine tables and merging datasets include:

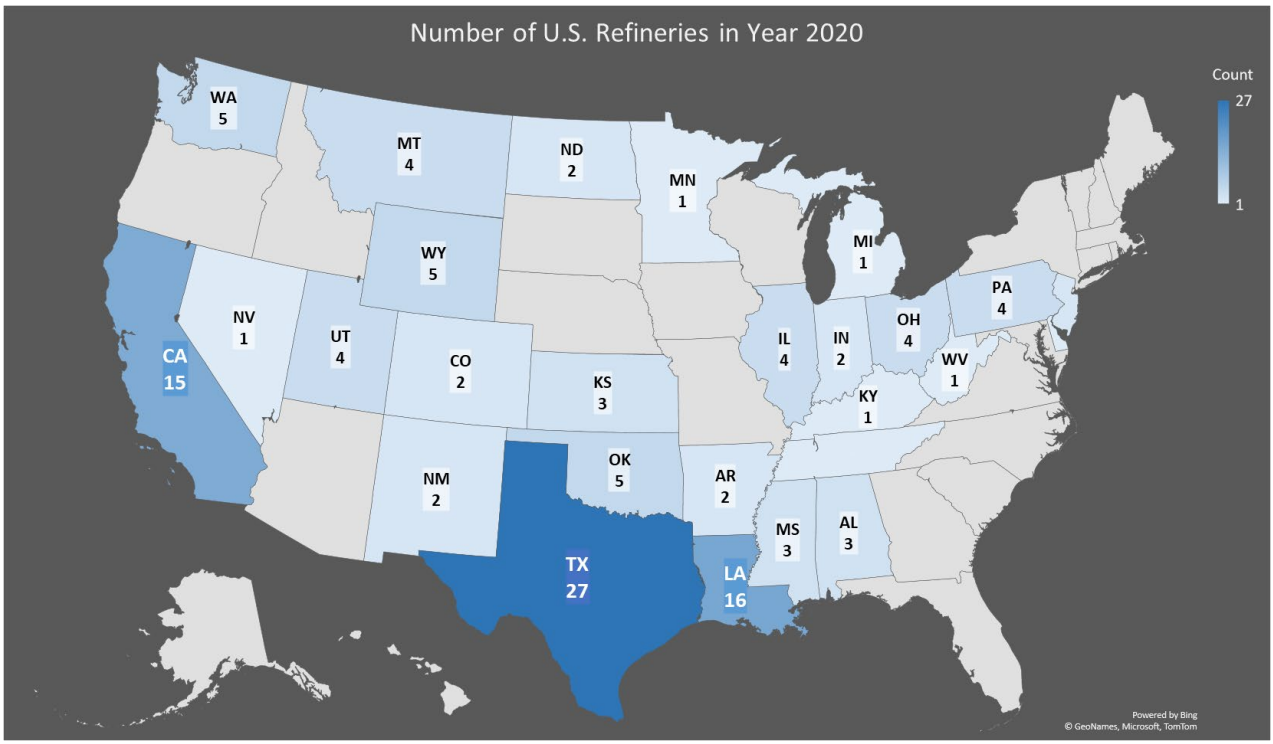
- Refinery capacities:
  - Atmospheric distillation
  - Process types specific, e.g., catalytic reformer, catalytic cracker, coking, vacuum, etc.
- Refinery product yields by volume
- Feedstock purchase and product resale pricing
- Distances from refineries to selected market destinations

Distance from refineries to market destination took quite some time, as I had to rely on free Google API, which limits the number of geolocation queries per days – I was looking for distances between 108 refinery locations and 66 major metro areas. It took me about a week to gradually collect the required information. Once the data was collected, I used it to merge with the rest of the datasets.

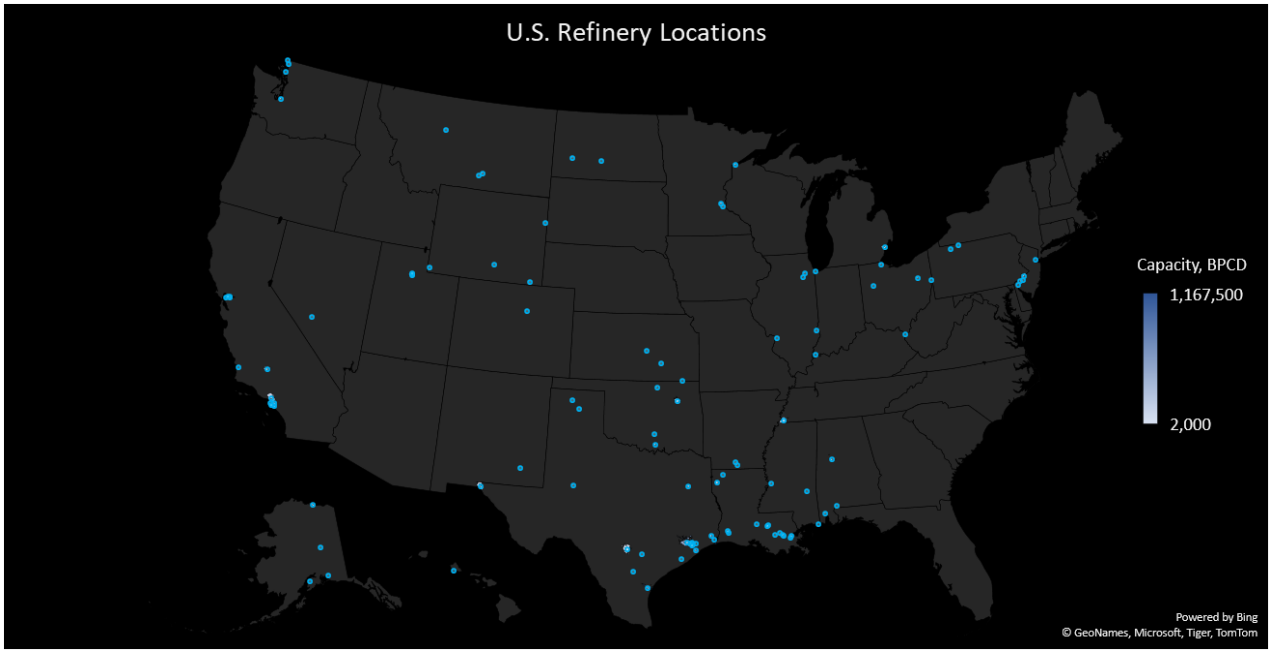
As result of data wrangling, I was able to produce a single dataframe in the shape convenient for further exploration: 969 entries (rows) with 88 data fields (columns)

4 Exploratory Data Analysis

Here's a map of the U.S. states with labels indicating the count of active refineries as of 2020.



Here are their locations shown as blue dots in the night view map. However, in this exercise we are only looking at the refineries within the contiguous U.S., i.e., the 48 adjoining states on the continent of North America.



4.1 Feature values and some definitions

The initial dataframe I was to explore consisted of variables that were refinery specific and destination market specific, with derived variables in between:

- Refineries: Capacity, Configuration, Yields, Distance to Markets, *Refinery Pressure to Markets*, and *Crack Spreads*
- Destination Markets: Distance to Markets, *Market Resistance*, *Market Value of Products* (combined Sale Prices per each volume of products yielded from each volume of crude processed

The *italicized variables* above were derived from combining the refinery specific and market specific variables, as well as calculating the abstract metrics, such as *Refinery Pressure to Markets and Market Resistance:*

- Market Resistance, labeled 'MRKT_RESIST_PAD_' - a perceived resistance of each destination market to aggregate supply from all refineries; the resistance is a measure of how much barrels per calendar day (BPCD) needs to travel to a given destination market per 1MM of its population.
- Refinery Pressure to Markets is similar to Market Resistance, but measuring the potential efforts needed to 'squeeze in' the products to a market, given that competitor refineries are doing the same thing
- Crack Spread is the most important feature here, as it is what we optimize for – it is a gross profit margin per 1 US barrel of crude refined, calculated as the difference between the Revenues generated from the sale of all refined products and the WTI crude price market index.
- Specific Refinery Crack spread is the main indicator of economic feasibility of a given refinery, given the potential range of crack spreads it may achieve in each of the 7 PAD districts, conditioned by the market resistance. **Specific Refinery Crack Spread will be used as the target variable in the modeling stage**.

Other feature values are:

- 'RFY_BPCD' - refinery operating capacity in Barrels per Calendar Day
- 'DIST_PAD_' - cumulative distances from each refinery to each of the destination markets (PAD Districts)
- 'CATCRKRECYL' through 'VACMDIST' are refinery configuration specifics, indicating the type of petroleum products they're capable of yielding from processing raw crude oils

- 'Brent Price' and 'WTI Price' - are crude oil market prices, and the 'Diff' refers to perceived discounts/premiums refineries get. The 'Diffs' are dictated by the going market, but mainly depend on the refineries' geographical locations, i.e., proximity to crude source, whether domestic or imported.
- 'Crude API' - indicates the type of crude oil, e.g., heavy, light, sweet, etc.; this determines the type and yield ratio of petroleum products that could be extracted from it, depending on refinery configuration
- 'Yields' are the expected volumetric ratios of refined petroleum products at refineries; here, grouped into just 4 categories - NG (Propane, Butane), GSL (Naphtha, Gasolines), DSL (kerosene, jet fuel, diesel, some lighter fuel oils), RESID (heavier fuel oils, lubricants, residual oil, etc.)
- 'REVNUE_PAD_' - perceived market value of the refined products at the destination markets; the calculation is based on historical wholesale prices, multiplied by the volumes in a given market district

4.1.1 Categorical Features

Here are the categorical variables:

- Series: 8 groups of observations per each of the 121 observed and 1 modeled refineries
- RFY-PADD: 5 high-level grouping of refinery locations per the U.S. petroleum allocation districts, (PADD = "Petroleum Administration for Defense Districts")
- RFY-PAD-Sub: 7 second-level grouping of PADD
- RFY-PAD-District: 7 lowest level grouping of PADD, there's some overlapping with RFY-PAD-Sub, as some states are divided among PAD-District groups
- RFY_ID: specific refinery ID; there are 122 unique refineries being observed
- RFY_State: states, where specific refineries are located

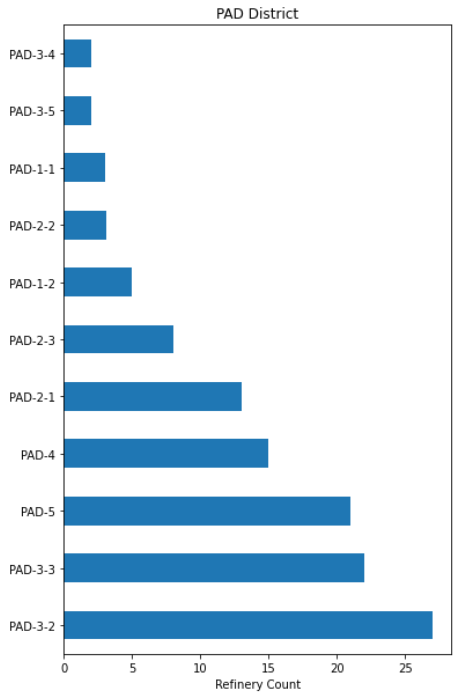
4.1.2 Numerical Features

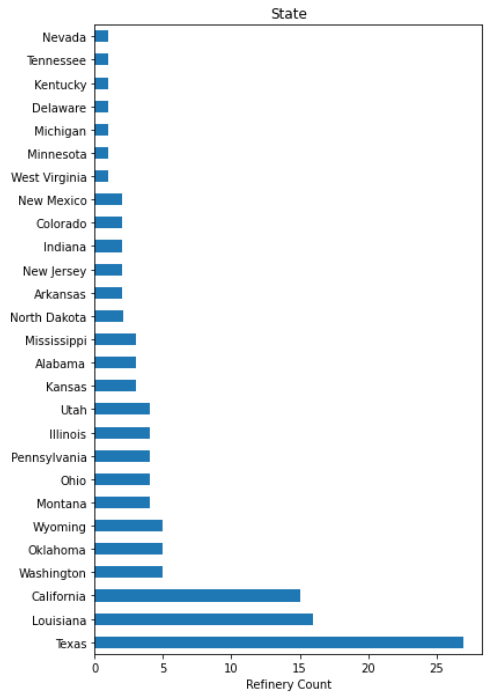
The key numerical features specific:

- Refinery specific metrics: Crude API, Product Yields (Gasoline, Diesel, Residue, Gas), Capacities by Process Type, Crude pricing, Crude Differentials (premium or discount), and Specific Crack Spreads
- Market specific metrics: Population Density in each destination market, Market Resistance, and Product Pricing

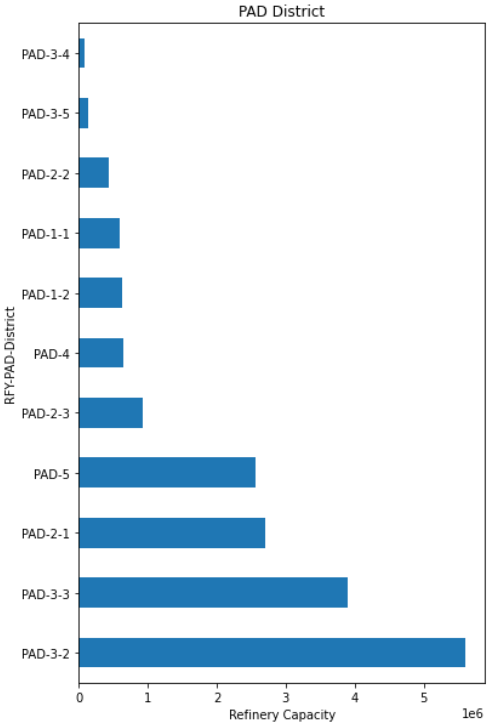
4.2 Distributions

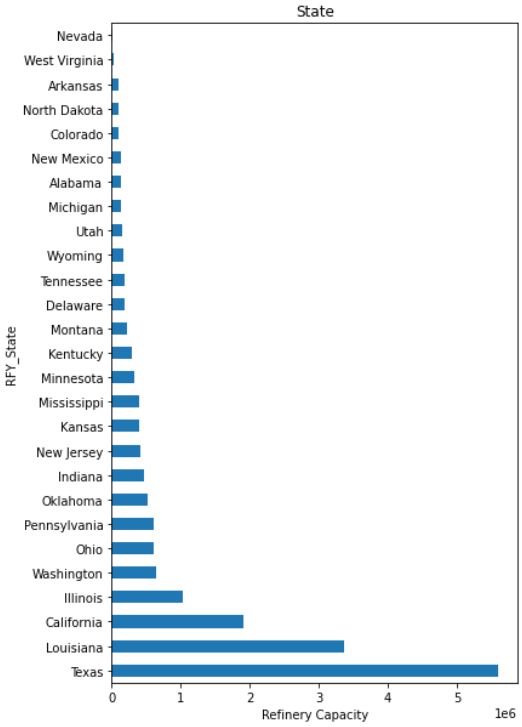
Texas, Louisiana, and California have the highest count of active refineries:





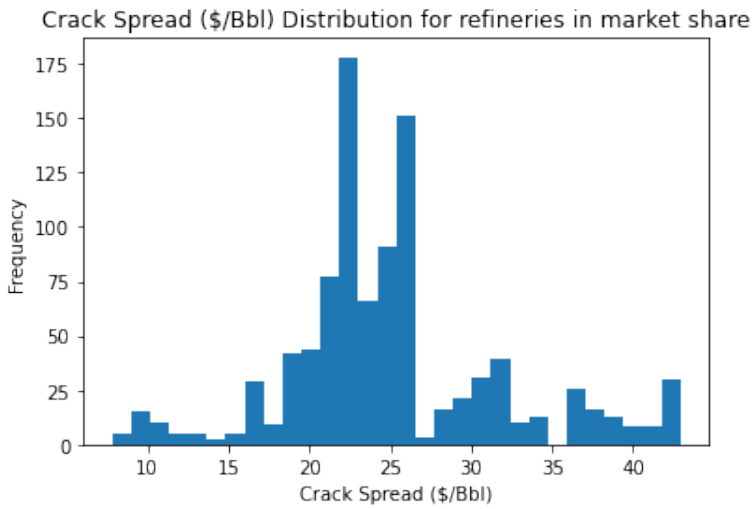
What about the processing capacities?

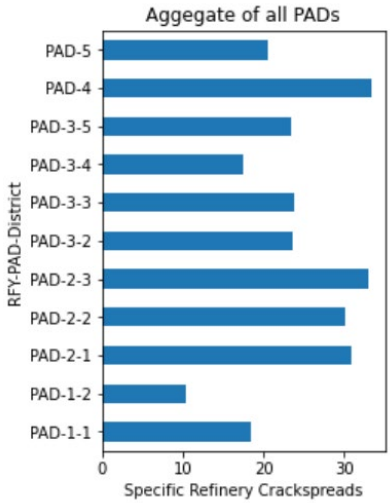




Capacity distribution is pretty similar to the count, except for CA and LA. LA refinery capacities are higher than the CA ones on average and TX holds the largest capacity of 5.6MM BPD.

Distributions of average refinery crack spreads and their aggregates by PAD districts:

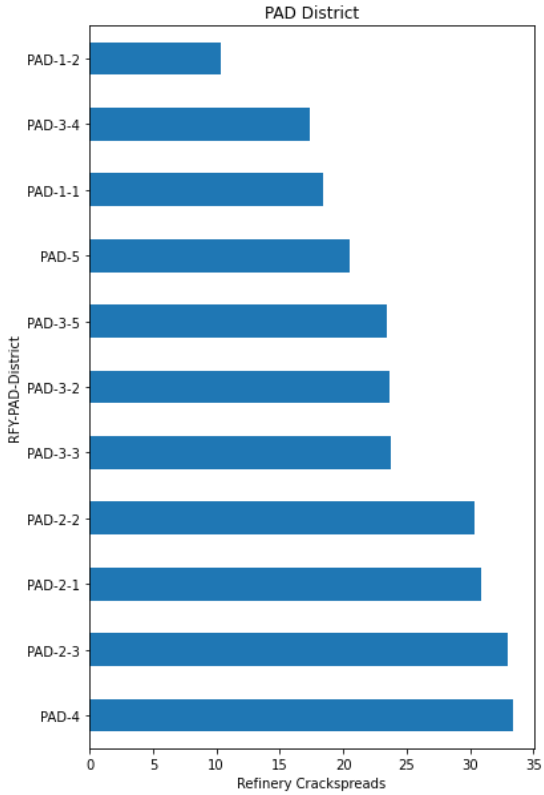


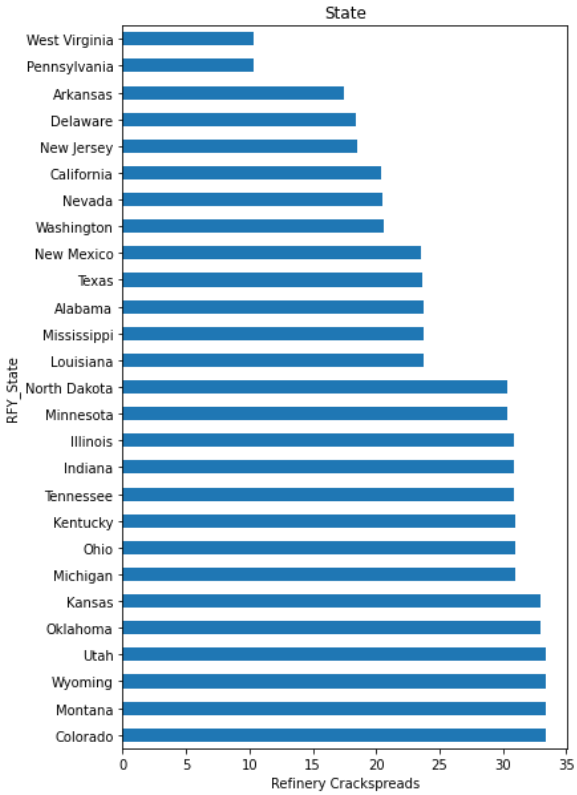


The distribution of average refinery crack spreads shows that the crack spreads range from the low value of $10/Bbl to the high value of $33/Bbl, averaging at $24/Bbl and standard deviation of $7/Bbl.

The distribution of expected crack spreads by PAD indicates that refineries in the PAD-4, PAD-2-3, and PAD-2-1 districts have the highest crack-spread averages.

Now let's look refineries in which U.S. locations have the highest crack spreads on average?





4.3 Hypothesis and Insights

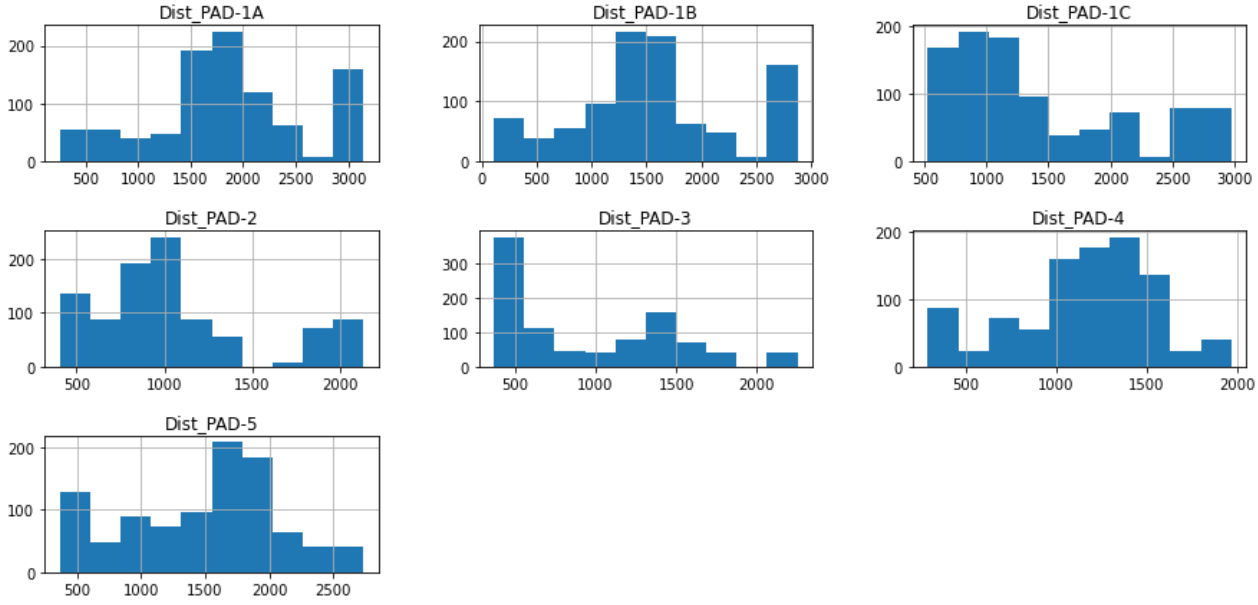
Crack spreads by PADs

The first thing we notice is that the PAD districts with the highest concentration of refineries, i.e., TX (PAD-3), LA (PAD-3), and CA (PAD-5) **are NOT in the list of "high crack spread" candidates**. The 'highest gross margins' are shown in PAD-4 and PAD-2.

The obvious reason for that is probably a higher supply to demand ratio, compared to other PAD districts. Since refineries target as many destination markets, as it's economically viable, let us look at what really drives the "supply to demand" pressure, i.e., distances from refineries to target markets.

Cumulative Refinery to Market Distances

Below histogram of cumulative 'Refinery to Destination Market' distances by PAD Districts confirm only a part of the above hypothesis.



Refineries in the PAD-3 primarily do have close proximity (500-750 miles) to target markets and hence, create the highest supply pressure.

However, gross margins of refineries in the PAD-5, particularly, in CA, seem to struggle for different reasons - there isn't much supply pressure and it's probably due to considerably lower production capacity per population, compared to those of TX and LA refineries.

Refineries in PAD districts 4 and 2 have a low concentration, relative to closest markets and hence, low supply pressure - we can guess that is the reason for potentially higher crack spread for the local production. Let's look at some other correlations

4.4 Visualizing Correlations

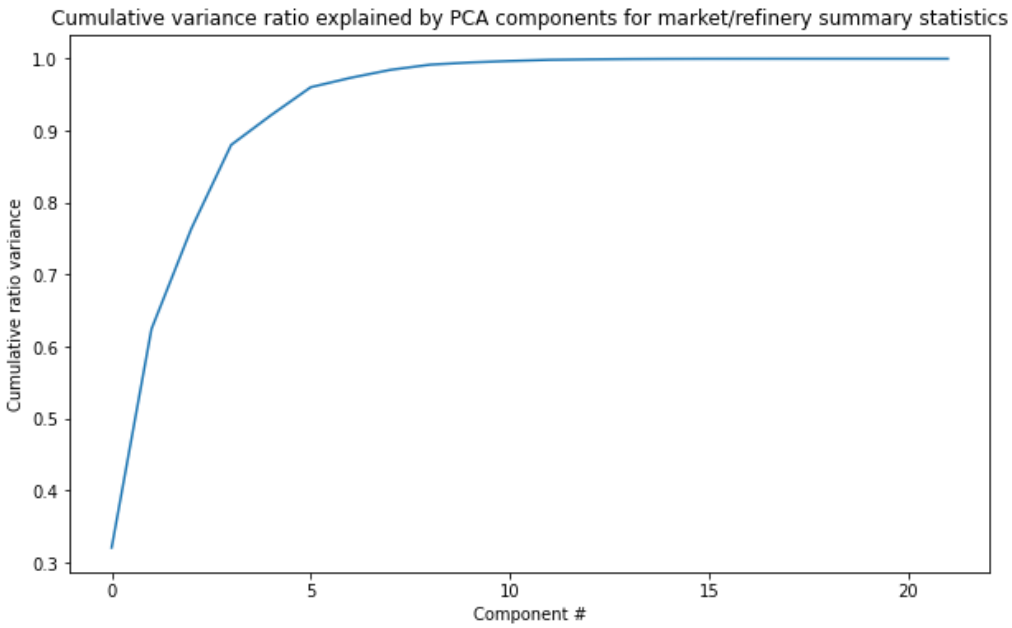
We will look at the following correlations

- refinery and market PAD districts
- specific refinery variable and the features

Refinery and Market Variables Correlations

The combined subset of refinery and market specific data used for identifying correlations include products supply density by specific refinery in barrels per day per PAD, market resistance to specific refinery supply per PAD, cumulative revenue value of all products refined and by specific refinery per targeted PAD market.

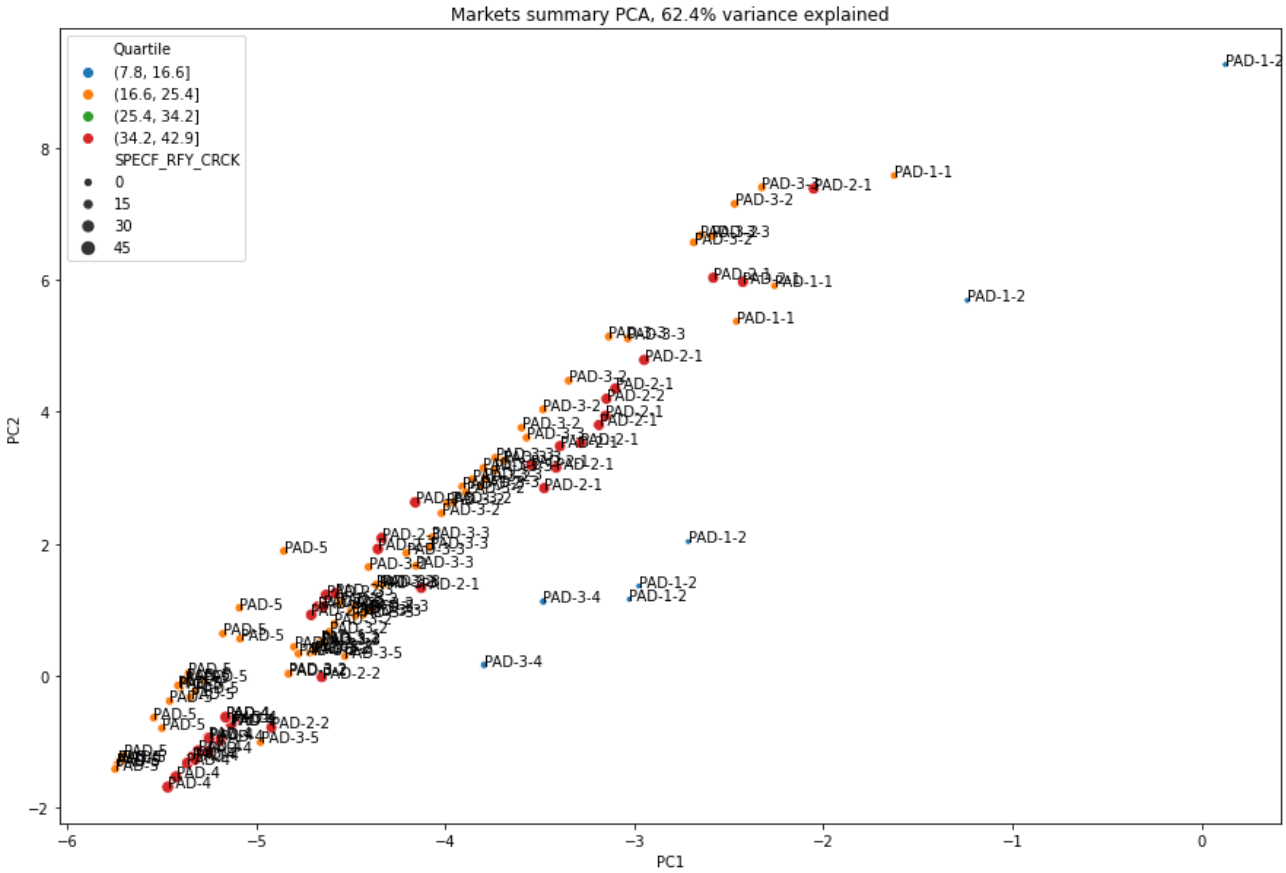
I started with scaling this data subset, using ***scale*** from ***sklearn.preprocessing*** library and then fit it to the principal component analysis (PCA) linear dimensionality reduction:



After applying PCA transformation to the subset data provided derived features I added specific refinery aggregate crack spreads, separated into quartiles, to PCA scatter plot:

RFY-PAD-District	PC1	PC2	SPECF_RFY_CRCK	Quartile
PAD-3-3	-4.084519	1.946419	22.575311	(16.6, 25.4]
PAD-3-3	-4.157284	1.665056	22.567569	(16.6, 25.4]
PAD-3-3	-4.394439	0.98852	22.571499	(16.6, 25.4]
PAD-3-4	-3.482732	1.123108	14.761021	(7.8, 16.6]
PAD-3-4	-3.7964	0.164473	14.760297	(7.8, 16.6]

Then I plotted just one time series of eight, for a clearer chart – and noticed a pattern how the quartile values of crack spreads, achieved by specific refineries are distributed across PADDs:



The PCA analysis shows that there is a clear grouping of refinery economic performance, dependent on their PAD location and the target markets.

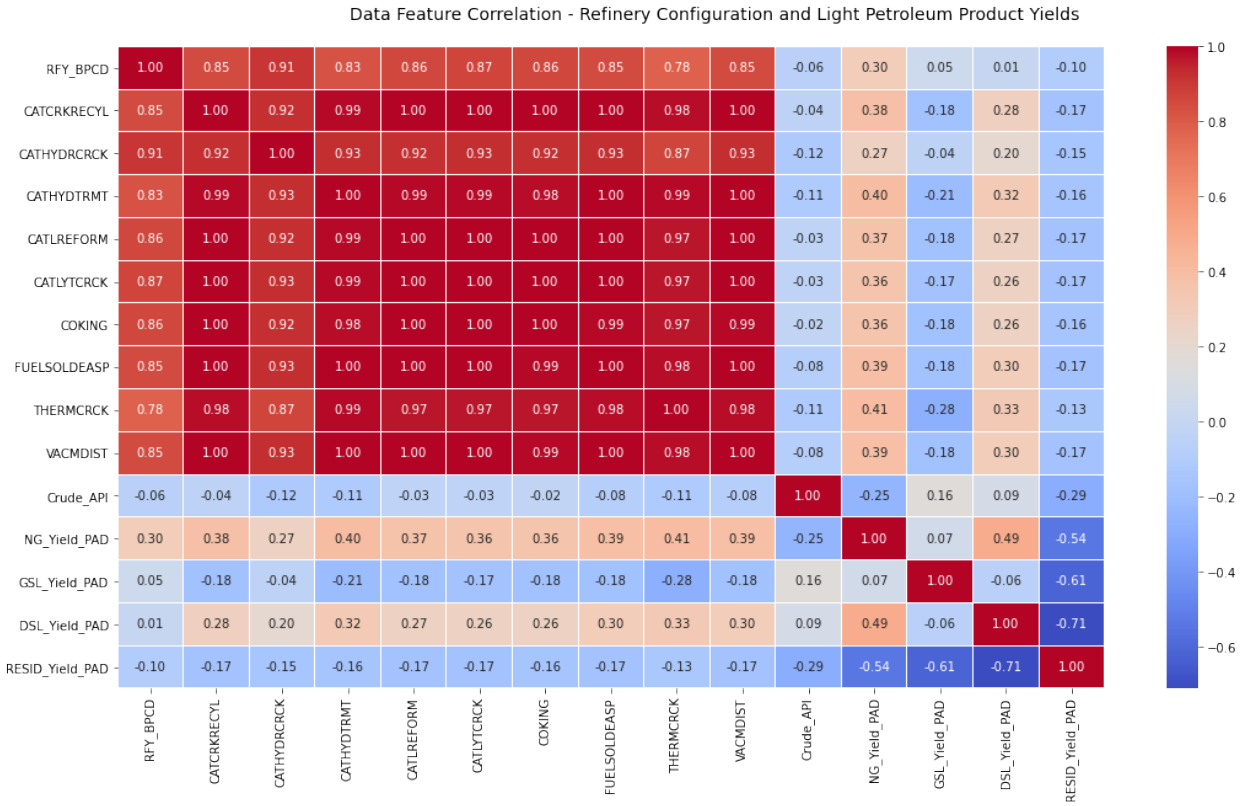
Refinery Variables Correlations

I then selected the following refinery specific variables to identify correlations:

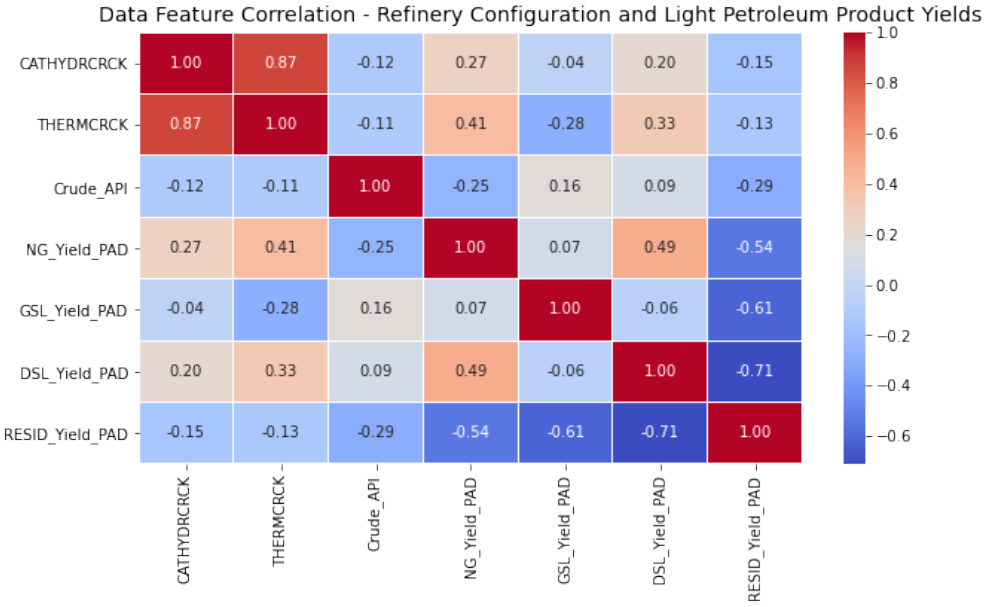
- Refinery configuration variables: 'CATCRKRECYL' through 'VACMDIST'
- Crude characteristics: 'CRUDE_API'
- Yields: 'GSL_Yield' and 'DSL_Yield'; we will omit the 'NG' and 'RESID', as in this analysis, we mainly are concerned with the high-value products, which are gasolines and diesel fractions

I used pandas corr() function, using 'pearson' method to plot the Data Feature Correlation chart. The correlations I focused were the **impact of refinery configuration**, i.e., specific process type capacities and crude API **(input) on the product yields (output)** as volume percentage of the crude processed (input).

The first chart showed a range of varying impacts of refinery configuration equipment capacities by processing type and Crude PAI on the product yields and in the second chart.

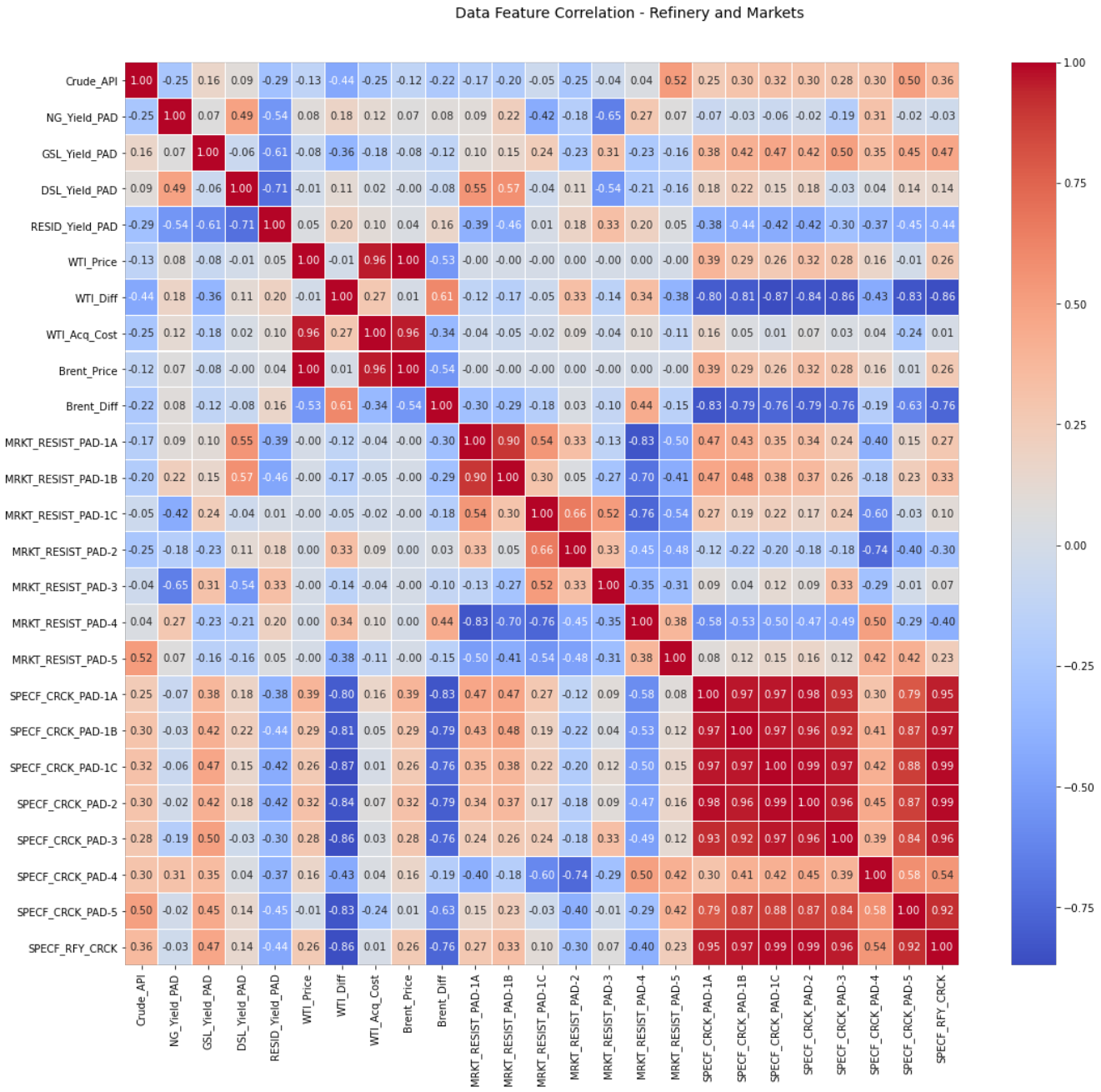


I then re-ran the correlation matrix with only those process types that seemed to have had the highest impacts; namely, the Catalytic Hydrocracking and Thermal Cracking processes:



4.5 Feature Engineering

Having merged the refinery and market specific datasets and analyzing the correlations, I was another step closer to narrow down the features to a dataframe, ready to be used for modeling. Here's the correlation matrix for the merged refinery and market dataframe:



It looks like the Specific Crack Spread values in PAD-Districts are highly correlated with the Market Resistance in the same PAD-Districts, where refineries are located, except for refineries located in PAD-2. These were clearly more concerned with the market resistance in PAD-1A and PAD-1B, which could indicate that their target customer base is located outside of their locale.

The aggregate Specific Crack Spread value, with the location factor ignored, is correlated with the following features: Gasoline Yield (GSL_Yield_PAD), type of Crude (Crude API), and the Crude Market Indices - both the WTI and Brent equally (0.26 each).

Now that we have identified our main features, we are ready for the data preprocessing and training.

5 Pre-processing and training

The final dataframe to be used for modeling included the following entries and fields:

```
#    Column               Non-Null Count  Dtype
0    RFY-PADD             969 non-null    object
1    RFY-PAD-Sub          969 non-null    object
2    RFY-PAD-District     969 non-null    object
3    RFY_ID               969 non-null    object
4    RFY_BPCD             969 non-null    int64
14   Crude_API            969 non-null    float64
15   NG_Yield_PAD         969 non-null    float64
16   GSL_Yield_PAD        969 non-null    float64
17   DSL_Yield_PAD        969 non-null    float64
18   RESID_Yield_PAD      969 non-null    float64
19   WTI_Price            969 non-null    float64
20   WTI_Diff             969 non-null    float64
21   WTI_Acq_Cost         969 non-null    float64
22   Brent_Price          969 non-null    float64
23   Brent_Diff           969 non-null    float64
71   BPD_DENS_SPECF       969 non-null    float64
38   SPECF_RFY_CRCK       969 non-null    float64
```

5.1 Initial Model (simple Linear Regression)

5.1.1 Split data into train/test partitions

I split the data into 70/30 train/test partitions:

```
X_train, X_test, y_train, y_test = train_test_split(rfy_data.drop(columns=['SPECF_RFY_CRCK', 'RFY-PADD',
                                                    'RFY-PAD-Sub', 'RFY-PAD-District', 'RFY_ID']),
                                                    rfy_data['SPECF_RFY_CRCK'], test_size=0.3, random_state=47)
```

5.1.2 Scale the features

I fit the Standard Scaler on the training data:

```
scaler = StandardScaler()

scaler.fit(X_train)
```

and used 'transform()' method to apply the scaling on the train and test splits:

```
X_tr_scaled = scaler.transform(X_train)

X_test_scaled = scaler.transform(X_test)
```

Verifying and comparing the scaling showed the following mean and standard deviation values:

- for the training set:

```
X_tr_scaled.mean(), X_tr_scaled.std()
```

```
(-3.7969518114457595e-16, 1.0)
```

- and for the test set:

```
X_test_scaled.mean(), X_test_scaled.std()
```

```
(-0.02514975534107685, 0.9727552245112391)
```

5.1.3 Train the model on the train split

I used a simple Linear Regression model:

```
lm = LinearRegression().fit(X_tr_scaled, y_train)
```

5.1.4 Make predictions using the model on both the train and test splits

```
y_tr_pred = lm.predict(X_tr_scaled)
```

```
y_test_pred = lm.predict(X_test_scaled)
```

5.2 Initial Model Metrics

The results of model training on the train split and making predictions using the model on the train and test splits were assessed as follows.

5.2.1 R-squared scores:

R-squared scores on the training and test data:

```
median_r2 = r2_score(y_train, y_tr_pred), r2_score(y_test, y_test_pred)
```

```
(0.9665666627253064, 0.9571428396445956)
```

> We see that the simple Linear Regression model explains 96.6% of the variance on the train and 95.7% on the test sets.

5.2.2 Mean Absolute Error scores:

Mean absolute error scores on the train and test splits:

```
median_mae = mean_absolute_error(y_train, y_tr_pred), mean_absolute_error(y_test, y_test_pred)
```

```
(1.0695864137409379, 1.1543230831643632)
```

> This model is expected to estimate the aggregate crack spread for a given refinery within $1.0/Bbl of the real crack spread value.

5.2.3 Mean Squared Error scores:

Mean squared error scores on the train and test splits:

```
median_mse = mean_squared_error(y_train, y_tr_pred), mean_squared_error(y_test, y_test_pred)
```

```
(1.5384724087104205, 1.7795594181981933)
```

5.3 Linear Regression Model with cross-validation

5.3.1 Pipeline for Linear Regression

I created a pipeline for rerunning the above steps and added a feature selection step:

- Scale the data to zero mean and unit variance
- Select the k best features using *sklearn's* function *SelectKBest* passing the scoring function *f_regression*
- train a linear regression model

I ran the pipeline and assessed the model performance:

```
pipe = make_pipeline(SimpleImputer(strategy='median'),
                     StandardScaler(), SelectKBest(f_regression, k=10),  LinearRegression())
y_tr_pred = pipe.predict(X_train)
y_test_pred = pipe.predict(X_test)
```

R-squared:

```
r2_score(y_train, y_tr_pred), r2_score(y_test, y_test_pred)
(0.9652249516553245, 0.9588558852980456)
```

MAE:

```
mean_absolute_error(y_train, y_tr_pred), mean_absolute_error(y_test, y_test_pred)
(1.0838636698460078, 1.1325117335451889)
```

> The improvement was negligible

The next task is to generalize to new data using cross-validation.

5.3.2 Cross-Validation

In cross-validation technique we build *k* models on *k* sets of data with *k* estimates of how the model performed on the unseen data, without having to touch the test set. I used *sklearn's cross_validate()* function to partition the training set into *k* folds, train the model on *k-1* of those folds, and assess performance on the fold not used in training – passing the previously built *pipeline* as the estimator object and setting cross-validation folds to 5.

5.3.3 Hyperparameter search using GridSearchCV

I used the *GridSearchCV()* function to run cross-validation for multiple folds and pick the *k*-value that returned the best performance of the model.

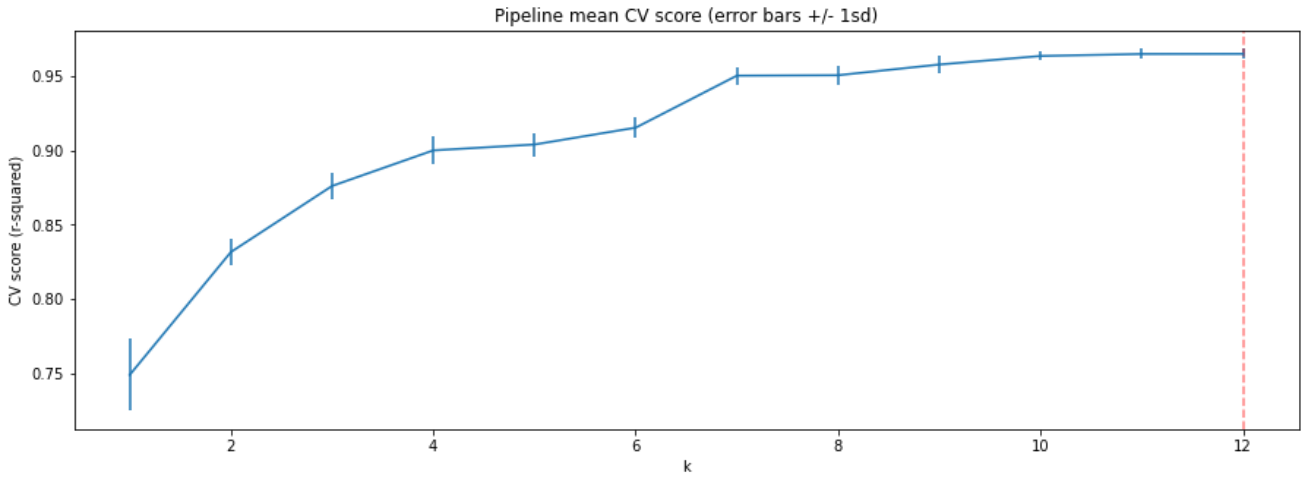
```
lr_grid_cv = GridSearchCV(pipe, param_grid=grid_params, cv=5, n_jobs=-1)
```

```
lr_grid_cv.fit(X_train, y_train)

GridSearchCV(cv=5, estimator=Pipeline(steps=[('simpleimputer',  SimpleImputer(strategy='median')),
                                             ('standardscaler', StandardScaler()), ('selectkbest',
                                              SelectKBest(score_func=<function f_regression at 0x00000155D31D1F70>)),
                                             ('linearregression', LinearRegression())]), n_jobs=-1,
                                              param_grid={'selectkbest__k': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]})
```

Performance:

```
score_mean = lr_grid_cv.cv_results_['mean_test_score']

        array([0.7490734, 0.83168343, 0.87611141, 0.90003385, 0.90398553, 0.91530349,0.95027318, 0.95056109,
        0.95774825, 0.96348883, 0.96482478, 0.96488315])

score_std = lr_grid_cv.cv_results_['std_test_score']

        array([0.0241717, 0.00857218, 0.00917778, 0.00926917, 0.00782644, 0.00672621, 0.00581759, 0.00612649,
        0.00615808, 0.00300102, 0.00341879, 0.00343075])

lr_grid_cv.best_params_

        {'selectkbest__k': 12}
```

The hyperparameter search has determined that the best value for *k* is 12:



5.4 Random Forest Model

5.4.1 Pipeline for Random Forest

Using similar approach to the Linear Regression model with *GridSearchCV*, I used a pipeline to run a Random Forest model, using *sklearn's RandomForestRegressor* class.

```
RF_pipe = make_pipeline(SimpleImputer(strategy='median'),

   StandardScaler(), RandomForestRegressor(random_state=47))
```

Passing the *pipe* object to *cross_validate()* function here will perform the fitting and assess the performance of the model.

```
rf_default_cv_results = cross_validate(RF_pipe, X_train, y_train, cv=5)

rf_cv_scores = rf_default_cv_results['test_score']
```

```
array([0.99884595, 0.99665714, 0.99778002, 0.98768229, 0.99465665])
```

```
np.mean(rf_cv_scores), np.std(rf_cv_scores)
```

```
(0.9951244101223411, 0.003971279419302659)
```

5.4.2 Hyperparameter search for Random Forest

In the grid parameters, I tried both the 'mean' and 'median' as methods for imputing missing values; also, set them to try with and without feature scaling.

```
n_est = [int(n) for n in np.logspace(start=1, stop=3, num=20)]

grid_params = {'randomforestregressor__n_estimators': n_est, 'standardscaler': [StandardScaler(), None],

    'simpleimputer__strategy': ['mean', 'median'] }
```

The best parameters from the GridSearchCV() with the Random Forest pipeline showed that feature scaling did not make a difference and that imputing with 'mean' was a better choice:

```
rf_grid_cv.best_params_
```

```
randomforestregressor__n_estimators': 615,
'simpleimputer__strategy': 'mean',
'standardscaler': None}
```

Using hyperparameters made very little difference over default CV results:

```
np.mean(rf_best_scores), np.std(rf_best_scores)
```

```
(0.9955247874974089, 0.0035313393593243963)
```

6 Modeling Results and Analysis

Here are the most useful features as determined by the models.

6.1 Linear Regression Model Results

Stepping into fitted grid-search object, I pulled the linear model coefficients from the 'coef_' attribute and feature names from the corresponding column names in the training set.

```
coefs = lr_grid_cv.best_estimator_.named_steps.linearregression.coef_

features = X_train.columns[selected]

pd.Series(coefs, index=features).sort_values(ascending=False)
```

Here's how the features ranked:

Brent_Price	21.681405
DSL_Yield_PAD	3.691864
GSL_Yield_PAD	3.178043
RESID_Yield_PAD	3.160543
NG_Yield_PAD	0.639637
BPD_DENS_SPECF	0.233863
Brent_Diff	0.011767
Crude_API	-0.079089
RFY_BPCD	-0.158277
WTI_Diff	-3.367532
WTI_Price	-9.779571
WTI_Acq_Cost	-10.433606

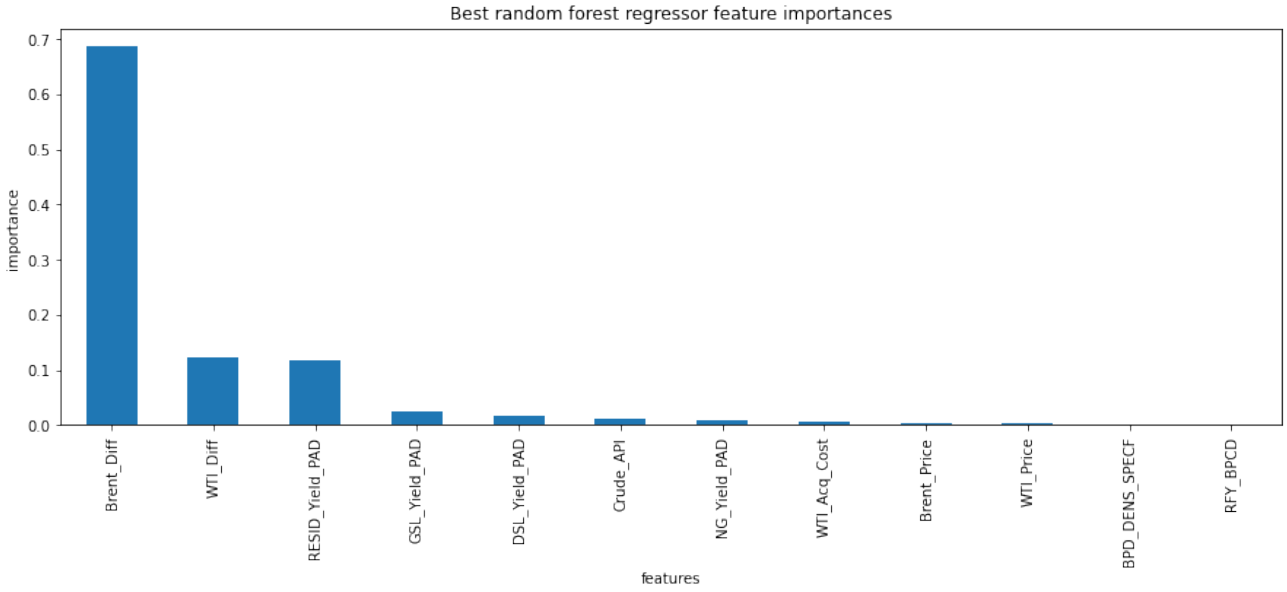
This Linear Regression model indicates that the Specific Crack Spreads are mostly driven by:

- Brent Price: quite possibly, it is - we see a highly negative correlation with the domestic WTI Price, so it could be that the cheaper imported crude does have a big impact on the refinery economics
- Residue Yield - this is the heavier fraction and is in negative correlation with the lighter fractions, i.e., the Diesel and Gasoline yields
- Diesel and Gasoline Yields - these are light products, which are sought out most on the market and bring in the bulk of refineries profits

- There's a strong negative correlation with domestic crude oil WTI price, as well as WTI price differential

6.2 Random Forest Model Results

I plotted a chart showing Random Forest's features by importance, using *'feature_importances_'* attribute to pull the feature values and corresponding field names from the train data columns



The Random Forest returned a much more sensible ranking of features by importance than the linear regression model:

- Brent_Diff - discount off/premium on the imported Brent crude oil, which does have effect on the domestic crude sales
- WTI_Diff - discount/premimium for domestic WTI crude oil price
- RESID_Yield_PAD - as opposed to the Linear Regression model, Random Forest model placed the Residue Yield as of a higher importance than Diesel and Gasoline yields. This makes sense, as the heavier fractions of products, refined from crude what indicates how much light product yields a refinery was able to extract
- GSL_Yield_PAD - how much gasoline a given refinery could produce per each barrel of crude processed
- DSL_Yield_PAD - how much diesel a given refinery could produce per each barrel of crude processed
- Crude_API - quality of crude
- WTI_Acq_Cost - actual crude oil price paid by refineries

6.3 Model Selection

I have tried a simple Linear Regression model, a Linear Regression model with cross-validation and a Random Forest model with cross-validation. The last two have been further improved with grid-search and built with best parameters. Let's compare their performance.

6.3.1 Linear regression model performance

```
lr_mae_mean = np.mean(-1 * lr_neg_mae['test_score'])

lr_mae_std = np.std(-1 * lr_neg_mae['test_score'])
```

```
(1.0908461747848313, 0.04277474406145815)
```

```
mean_absolute_error(y_test, lr_grid_cv.best_estimator_.predict(X_test))
```

```
1.1543230831643632
```

6.3.2 Random forest regression model performance

```
rf_mae_mean = np.mean(-1 * rf_neg_mae['test_score'])

rf_mae_std = np.std(-1 * rf_neg_mae['test_score'])
```

```
(0.1334513516888089, 0.05712131291841946)
```

```
mean_absolute_error(y_test, rf_grid_cv.best_estimator_.predict(X_test))
```

```
0.0806464774716703
```

Clearly, the random forest model shows less variability and lower cross-validation mean absolute error by almost $1.00 – so, I selected the best random forest model for running prediction scenarios.

6.4 Model Scenarios

6.4.1 Starting Point

The ultimate goal, as outlined in the Problem Statement was to validate the feasibility studies for building a new refinery in PAD-2-2 district by identifying 2-3 key parameters that have the highest impact on the Specific Crack Spread.

I had already confirmed the key parameters – i.e., the **top features by importance**. The Baseline Random Forest model prediction **($25.97/Bbl)** for the Greenlight Refinery came out as follows:

Brent_Diff	WTI_Diff	Crude_API	RFY-PAD-District	RFY_BPCD	WTI_Price	PAD-2-2	PAD-3-3	PAD-4	SPECF_RFY_CRCK
(2.08)	(6.05)	26.12	PAD-2-2	50,000	38.52	1	0	0	25.97

I then ran several scenarios to change the values of the key features to gauge the impact. Here are **three example scenarios** that I have selected to illustrate the final conclusion of this project.

GIVEN:

The new refinery is built in PAD-2-2 district, as initially suggested by various distribution charts, plotted during EDA; Brent price differential is $(2.08)/Bbl discount, WTI price differential is $(6.05)/Bbl discount, and the predicted Specific Refinery **Crack Spread is $25.97/Bbl**.

6.4.2 Scenario 1

CHANGES:

- Changed Brent Discount to $0/Bbl
- Changed Brent Discount to $0/Bbl AND WTI Discount to $0/Bbl
- Changed WTI Discount to $(7.5)/Bbl, keeping Brent Discount at $0/Bbl
- Changed Brent Discount to $(5.5)/Bbl keeping WTI Discount at $0/Bbl
-

6.4.3 Scenario 2

CHANGES:

- Changed the Crude API from the baseline value of 26.11 to 39
- Changed the Crude API from the baseline value of 26.11 to 41

6.4.4 Scenario 1

CHANGES:

- Changed location of the refinery from the Baseline PAD-2-2 to PAD-4
- Changed location of the refinery from the Baseline PAD-2-2 to PAD-3-3

7 Summary and Conclusion

7.1 Key Findings

7.1.1 Scenario 1

OUTCOME:

- Lowering Brent Discount from $(2.08)/Bbl to $0/Bbl lead the Crack Spread to drop from $25.97/Bbl to $25.96/Bbl
- Lowering both the WTI and Brent Discounts $0/Bbl, lead the Crack Spread to drop even further – to $20.58/Bbl
- Increasing WTI Discount from the Baseline $(6.05)/Bbl to $(7.5)/Bbl, while keeping the Brent Discount at zero, added a modest increase in the Crack Spread – from $20.58/Bbl to $22.99/Bbl, but still lower than the Baseline of $25.97/Bbl
- Increasing Brent Discount from the Baseline $(2.08) to $(5.5), while keeping the WTI Discount at zero had a significant impact, increasing the Crack Spread – to $34.66/Bbl

The conclusion here is that Brent Differential has a stronger impact on the economics than the domestic WTI crude price index. There is also an indication that WTI is dependent on the Brent.

7.1.2 Scenario 2

OUTCOME:

- Changing the Baseline value of the Crude API from the heavier 26.11 to the lighter 39 resulted in a slightly better outcome – the Crack Spread went from $25.97/Bbl to $26.03/Bbl
- Further increase of Crude API to 41 and 42 did not add anything – the Crack Spread stayed at the $26.03/Bbl level

This was expected, as the heavier crude doesn't automatically mean fewer valuable products may be extracted from it than the lighter crude – it is mainly a function of refinery process type and the equipment. However, there is more lighter product yields in the lighter crude oil. It is easier to extract than from the heavy crude. This effect caps out at the lower and higher ranges, in this case, the 39-42 API range.

7.1.3 Scenario 3

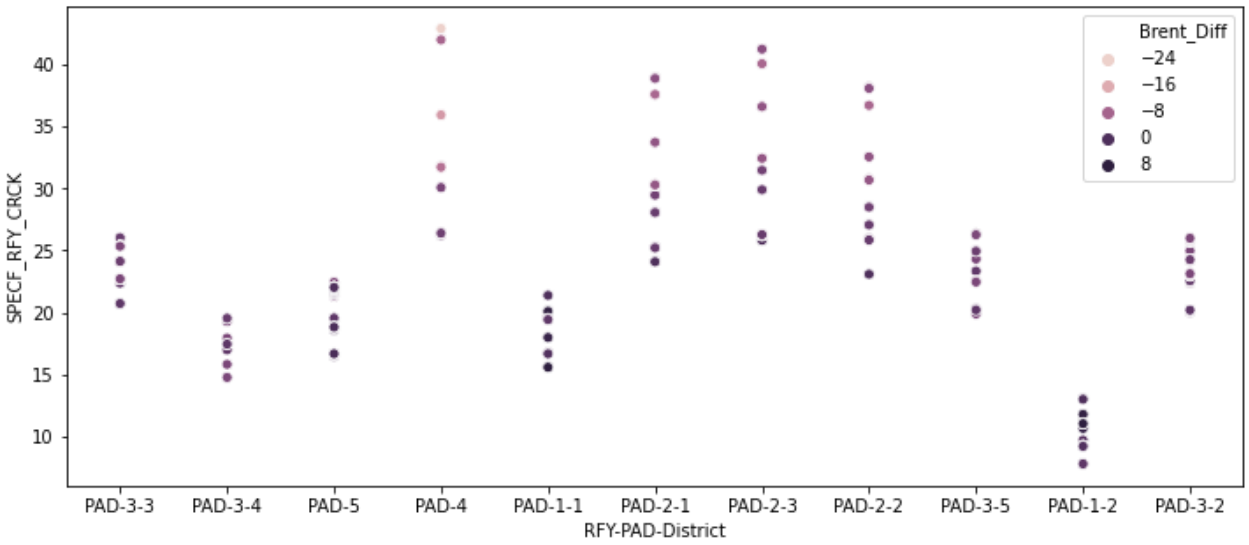
OUTCOME:

- Moving the refinery location from PAD-2-2 to PAD-4 resulted in a very negligible change in the predicted Crack Spread – it went from $25.97/Bbl to $25.99/Bbl
- Similarly, changing the location from PAD-2-2 to PAD-3-3 did not have any further affect

The model did not rank any of the one-hot encoded 'PAD' district features very well in the 'feature importance' assessment. The reality though supports the theory that the PAD location is a very important factor in economic feasibility of a crude oil refinery – an indication that the model definitely needs further tweaking.

7.2 Conclusion

During the EDA the importance of choosing a location for a new refinery in relation to the targeted PAD districts seemed very obvious – here is a scatter plot that illustrates the expected Crack Spreads are distributed by PAD district:

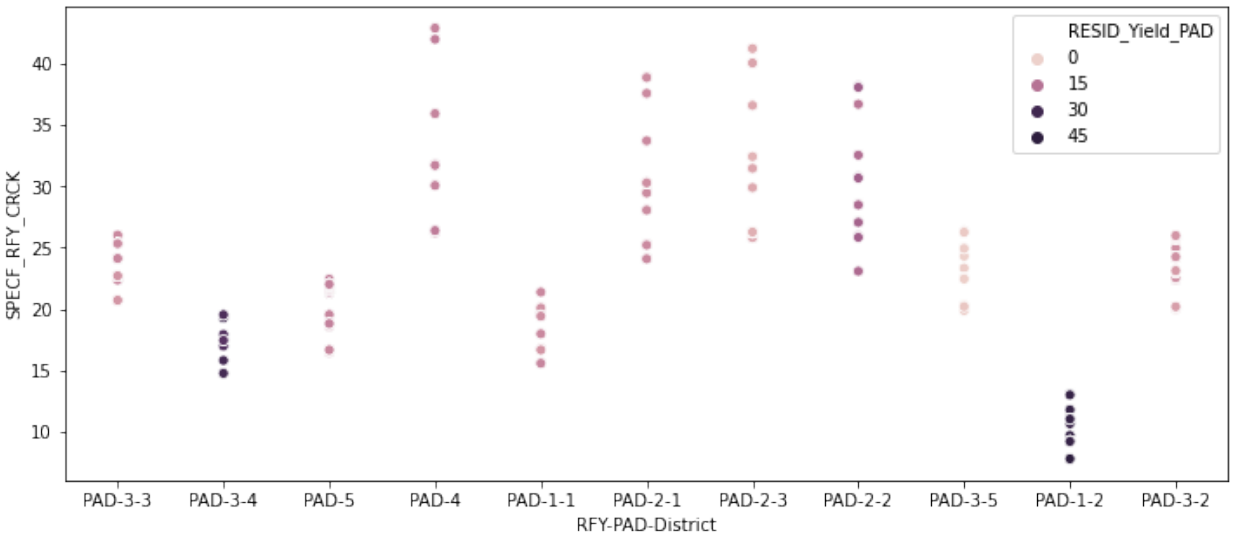


However, the Random Forest model ranked crude purchase price discount, light product yields, and type of crude as the most important features in predicting the gross profit margins:

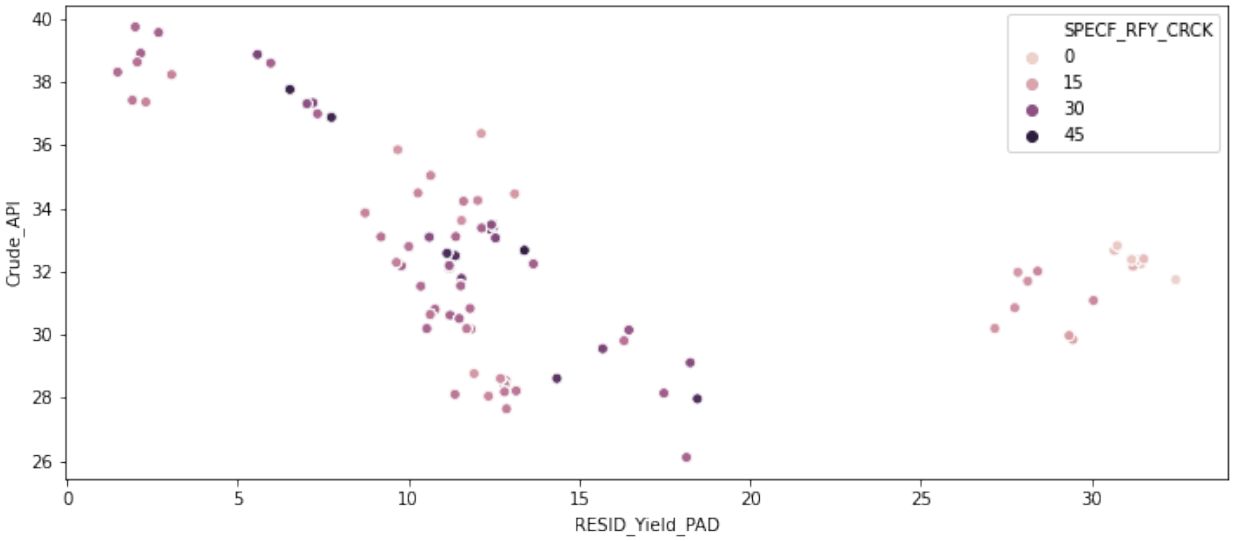
- The same scatter plot above shows that some refineries in PAD-3 (TX and LA) and PAD-1 are actually paying a premium on crude, whereas the discounts are widely common in PAD-3 and PAD-4
- Prevalence of lighter yields means higher value products, so this insight also makes sense – refer to below scatter plot, showing how Crack Spreads are affected by the Residue Yields (which is an inverse of the light products) and distribution by PAD
- Finally, Crude API is highly correlated with the Residue yields and hence, also drives refinery profitability. There are two more scatter plots below – one showing the

correlation of the Crack Spread and the Residue Yield, which in its turn is dependent on the type of Crude it is processed from

CRACK SPREADS AND PRODUCT YIELDS (RESIDUE)



CRUDE API AND RESIDUE YIELDS (AFFECTS CRACK SPREAD)



7.3 Recommendations (Further Work)

When selecting this project as my Capstone II, I did not imagine the amount of work it would require – only after spending days of data collection and wrangling I realized the scale of the scope. It seemed to continue to expand as I kept asking new questions – till I limited the scope.

I will continue to work on this project outside of the course work and am hoping to produce a useful application for the industry. There is plenty of space for improvement – work on both the data and the models.