US Drilling Rig Count

World Drilling Rig Analysis and Prediction of Drilling Rig Count in the US

Tamey Washington

Andrey Tokarev

Jonathan Ezeugo

Simon Feinsilver

Content

Executive Summary

When drilling rigs are active, they consume products and services produced by the oil service industry. The active rig count acts as a leading indicator of demand for products used in drilling, completing, producing and processing hydrocarbons.

The US Oil&Gas industry market is very dynamic and competitive. The industry in general and especially US O&G industry is very resources heavy business, requiring substantial amount of fixed assets, workforce and maintenance budget.

Problem Statement

Many businesses rely on the drilling rig count forecast to plan budget, workforce and resources.

As the domestic rig count is directly related to the health of the oil industry, our group has set out to predict the number of drilling rigs in the US based on some inputs. We believe that insight of this nature would help the industry and its stakeholders at large have a better grasp on the market swings of a fairly volatile market and help in financial and production planning of all types.

Data Gathering

In order to do this, we analyzed supply and demand data from around the world, the price of brent crude, and oil rig counts all over the globe.

In choosing to analyze the US Rig Count, we quickly realized that it would be hard to find a single dataset that would provide enough relevant information for such a grand task. To combat this problem, we decided to combine 3 different csv files, 2 from kaggle and 1 provided by Baker Hughes. The 3 different sets of information we gathered were oil supply/demand data, brent crude price, and rig count.

One of the issues we ran into with having 3 different data sources is that not all the data was provided with the same time intervals. To combat this, we used pivot charts and different Microsoft Excel skills to aggregate the data into the largest common time interval, quarter year. While this reduced the total number of rows of data, we still feel that there is enough data for our machine learning algorithms to accurately predict our target. From there, we combined the data into a single csv file, and then read it into a pandas dataframe using the Jupyter Notebook software. Let the fun begin!

Data Gathering

```
1 file = "../Data/Capstone_Data.csv"
```

```
1 oil_df = pd.read_csv(file)
```

```
1 oil_df.head()
```

	Year	Quarter	Canada_D	Europe_D	Japan_D	US_D	China_D	Soviet_D	Asia_D	Other_D	Total_World_D	Canada_S	Mexico_S	North_Sea_S	Other_S
0	2019	Q3	2.57	15.44	3.43	20.88	14.37	5.58	13.74	26.24	102.25	5.47	1.93	2.96	4.64
1	2019	Q2	2.32	14.95	3.39	20.63	14.65	5.19	14.11	25.77	101.00	5.47	1.91	2.96	4.59
2	2019	Q1	2.31	14.82	4.06	20.55	14.46	5.15	13.95	25.22	100.49	5.43	1.91	2.96	4.85
3	2018	Q4	2.58	14.93	3.89	20.75	14.10	5.36	13.82	25.29	100.73	5.62	1.95	2.95	4.89
4	2018	Q3	2.65	15.47	3.53	20.86	13.88	5.50	13.48	25.88	101.25	5.41	2.09	2.84	4.65

Data Preparation

As stated above, we now had all of our data broken down by quarter. In prepping our data for machine learning, we learned that our quarter column was actually a string. For example, instead of quarter 1 being displayed simply as integer "1", it was "Q1". Due to this it became necessary to encode the data using either label encoding or one-hot-encoding. To ensure that our machine learning algorithms would not erroneously assign more values to quarters 3 and 4 over quarters 1 and 2, we used we used pandas get_dummies method to one-hot-encode the quarter data.

```
#One Hot Encoding
dummies = pd.get_dummies(oil_df.Quarter)
oil_encode_df = pd.concat([oil_df, dummies], axis='columns')
oil_encode_df.head()
```

irent_Crude_Price	Latin_America_Rigs	Europe_Rigs	Africa_Rigs	Middle_East_Rigs	Asia_Pacific_Rigs	Total_Intl_Rigs	Canada_Rigs	US_Rigs	Q1	Q2	Q3	Q4
61.95	195	190	114	422	224	1144	132	920	0	0	1	0
69.04	186	159	122	412	230	1109	83	989	0	1	0	0
63.10	188	92	116	398	235	1030	186	1046	1	0	0	0
68.76	193	90	106	397	225	1011	177	1072	0	0	0	1
75.07	191	84	104	399	225	1003	208	1051	0	0	1	0

To explore our data, we created histograms of each of our columns and found that while some features displayed normal distributions, there were quite a bit that were either left or right skewed. We used sklearn's MinMaxScaler import to scale our data for this issue.

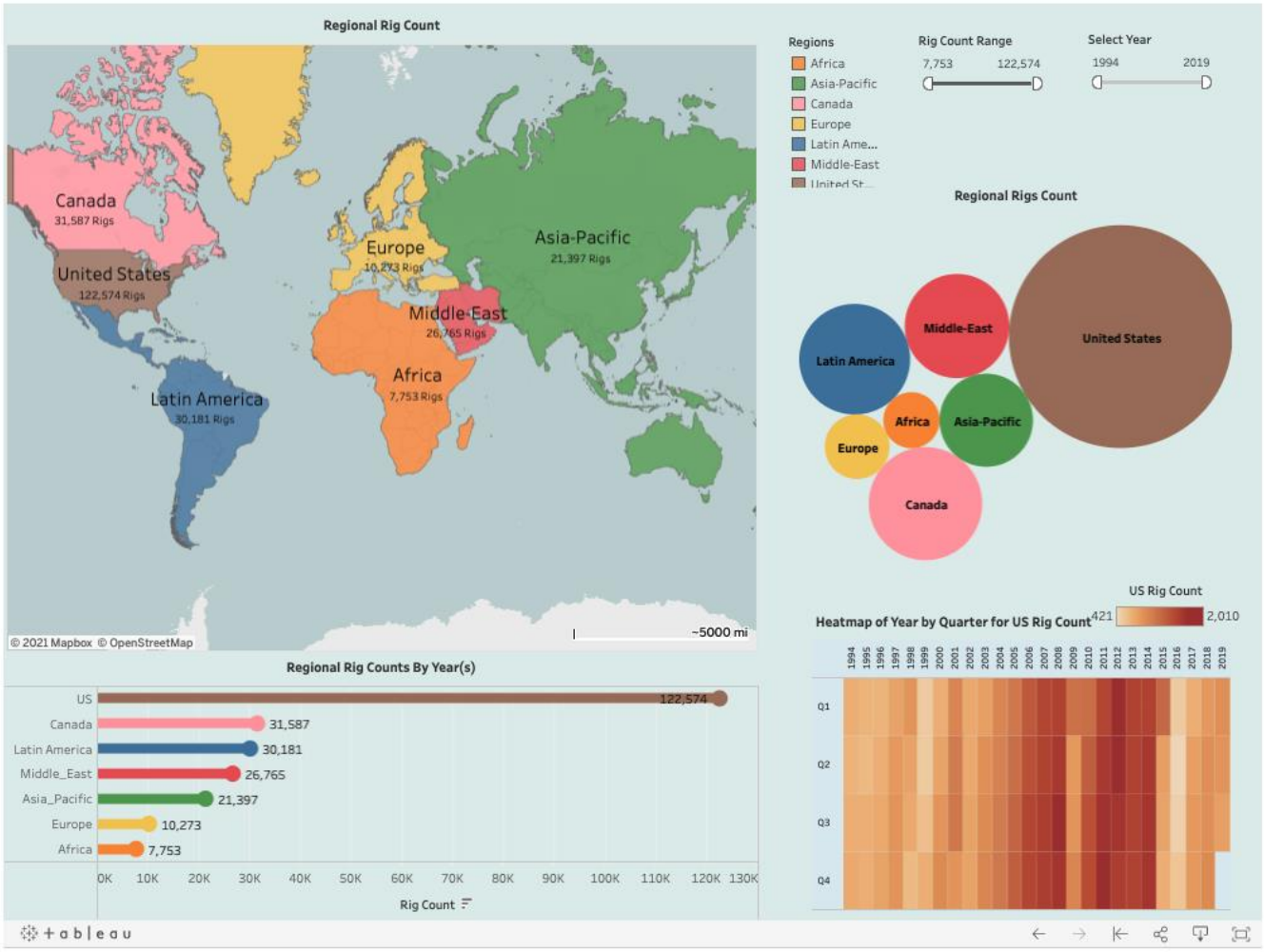
```python
#Scaling
from sklearn.preprocessing import MinMaxScaler
x_scaler = MinMaxScaler()
features = oil_encode_df.drop(["US_Rigs", "Quarter"], axis=1)
features_new = features[['Total_Intl_Rigs','Asia_Pacific_Rigs', 'Brent_Crude_Price', 'Latin_America_Rigs']]
x_scaler.fit(features_new)
features_scaled = x_scaler.transform(features_new)
df_scaled = pd.DataFrame(features_scaled, columns=features_new.columns)
df_scaled["US_Rigs"] = oil_encode_df["US_Rigs"]
df_scaled.head()
```

	Total_Intl_Rigs	Asia_Pacific_Rigs	Brent_Crude_Price	Latin_America_Rigs	US_Rigs
0	0.739464	0.641892	0.461315	0.058140	920
1	0.694764	0.682432	0.525775	0.023256	989
2	0.593870	0.716216	0.471770	0.031008	1046
3	0.569604	0.648649	0.523229	0.050388	1072
4	0.559387	0.648649	0.580598	0.042636	1051

Data Exploration and Analysis with Tableau

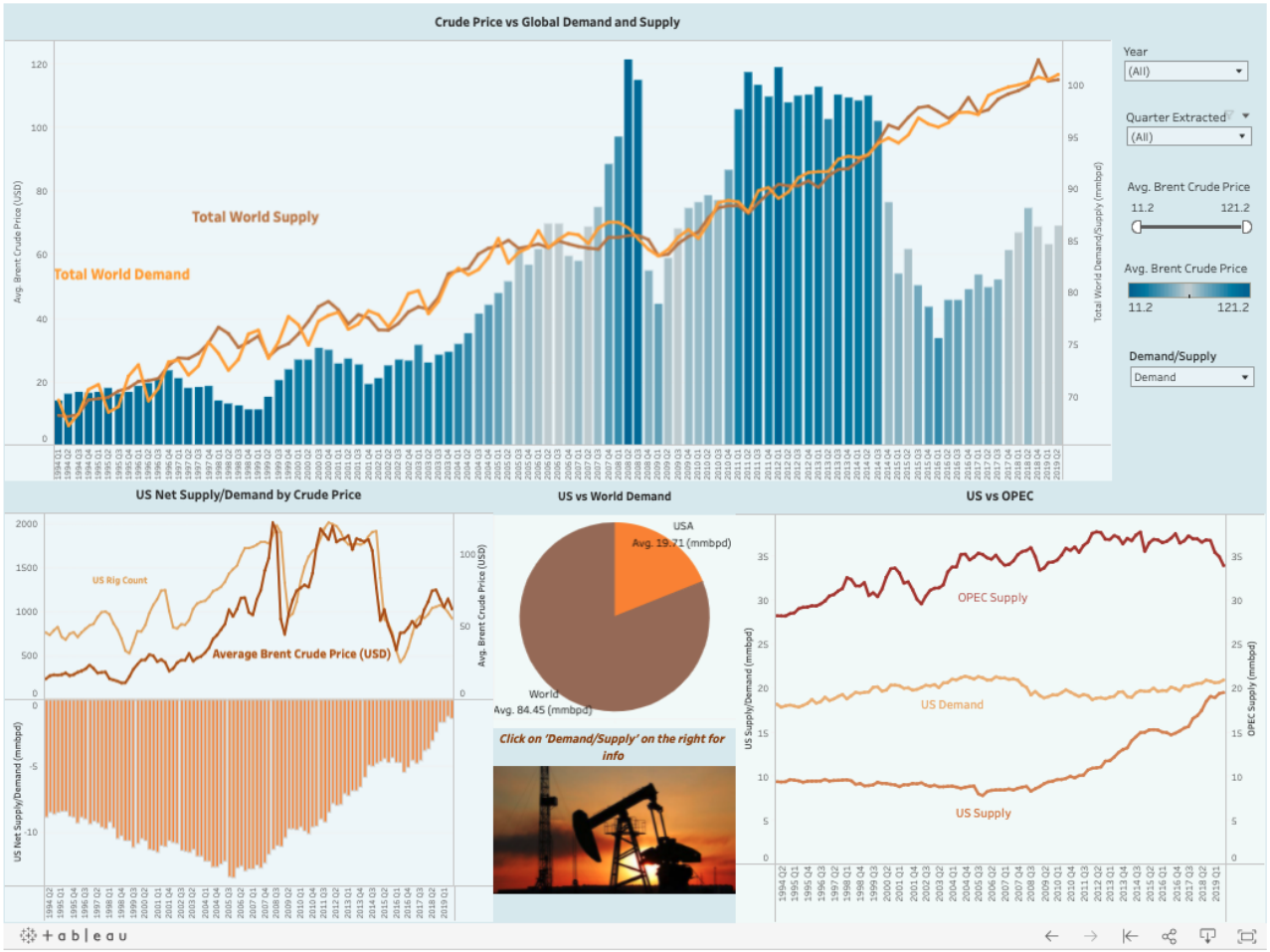
Mapping Global Regional Rig Counts With US In Spotlight

On this dashboard, we explore global regional rig counts over the analysis period. We also in the heatmap analyze the most active US quarters and years for rig counts.


Regional Rig Count
Canada
31,587 Rigs
United States
122,574 Rigs
Latin America
30,181 Rigs
Europe
10,273 Rigs
Middle-East
26,765 Rigs
Africa
7,753 Rigs
Asia-Pacific
21,397 Rigs
© 2021 Mapbox © OpenStreetMap
~5000 mi
Regions
Africa
Asia-Pacific
Canada
Europe
Latin Ame...
Middle-East
United St...
Rig Count Range
7,753
122,574
Select Year
1994
2019
Regional Rigs Count
Latin America
Middle-East
United States
Africa
Asia-Pacific
Europe
Canada
Regional Rig Counts By Year(s)
US
Canada
Latin America
Middle_East
Asia_Pacific
Europe
Africa
122,574
31,587
30,181
26,765
21,397
10,273
7,753
0K
10K
20K
30K
40K
50K
60K
70K
80K
90K
100K
110K
120K
130K
Rig Count
US Rig Count
Heatmap of Year by Quarter for US Rig Count
421
2,010
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
Q1
Q2
Q3
Q4
+ableau

Demand and Supply Analysis

Often we wonder why US oil consumption impacts global geopolitics and crude pricing. On this dashboard, we analyze US Demand and Supply metrics in relation to global perspectives.

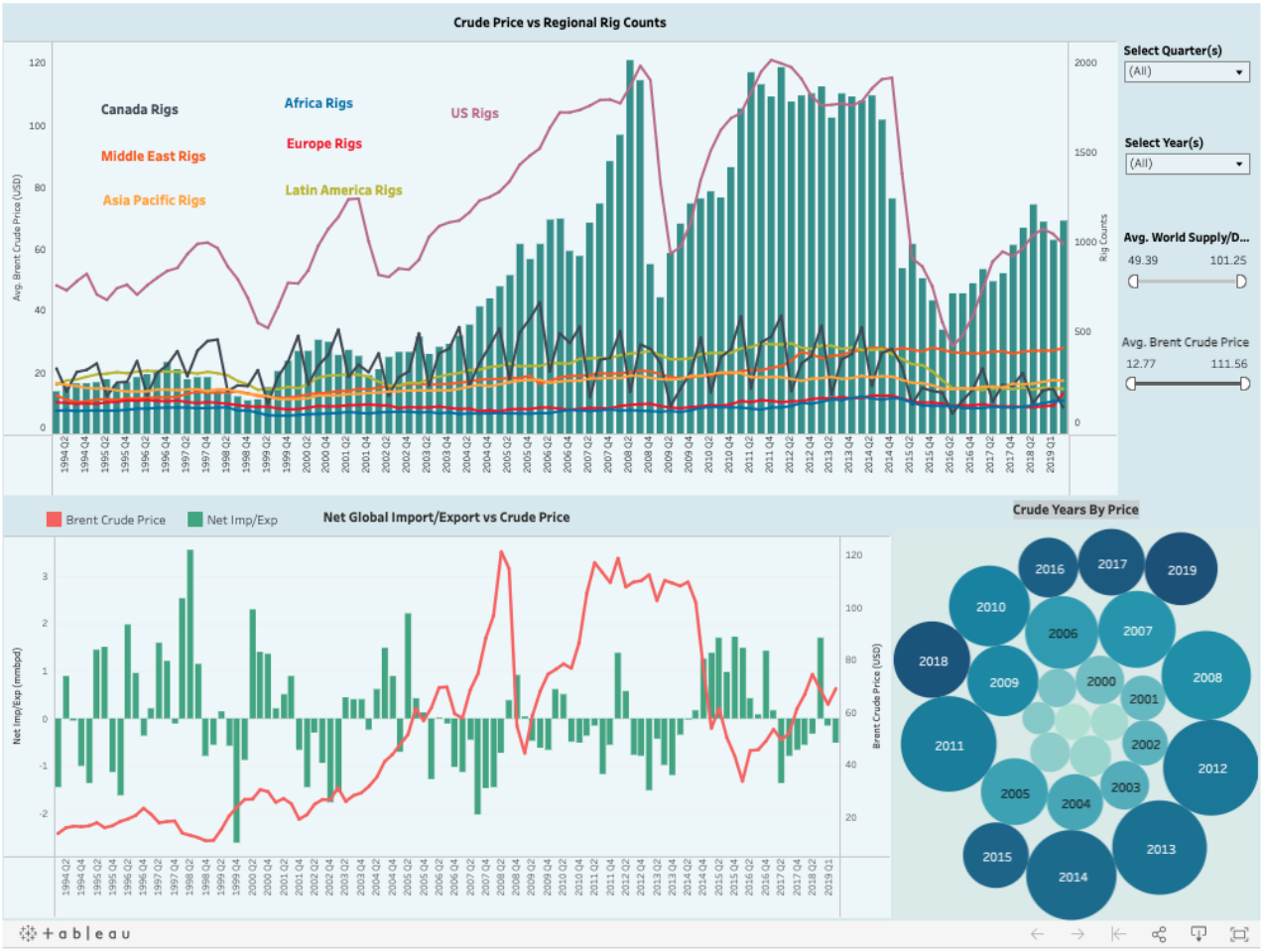


It is fascinating how quickly the US was able to close the domestic supply/demand gap. The rapid growth of US supply was possible by combining two technologies: horizontal drilling and hydraulic fracturing. While it allowed much lower energy prices for the US residents, it created substantial competition for the word producers and created and created serios challenges for the US O&G industry. The market was flooded with crude oil which crushed oil prices forcing many US based companies going under. The point here is supply and demand is, of

course, the fundamental criteria for oil price and therefore oil rigs and, over the past decade, US became not only key oil consumer but also key oil producer.

Crude Price, Regional Rig Counts and Net Demand/Supply Relationships

On this dashboard, we explore the relationship between crude price and deployment of rigs across 7 global regions, as well as global net crude import (demand) and export (supply) analyzed with crude price.



The graph visualization above greatly demonstrates how sensitive the US Drilling Rigs are to the crude price while the rest of the world is more consistent. The US is able to rapidly pick up and lay down the drilling rigs. The number of drilling rigs in the US is much larger than anywhere in the word making it a very

attractive market for Oil Field Service Companies which also drives very high competition. Recent low crude prices and the high competition forced all companies the US to challenge status quo and develop new creative technologies aimed to reduce cost of production.

Machine Learning

Feature Exploration & Correlation

Due to our desire to create a more usable model for the end user, we wanted to try and eliminate some of the columns that appeared to be less highly correlated with our US Rig Count target. Through trial and error, we landed on a correlation benchmark of .6, both positive and negative. Creating feature columns with a correlation of .6 or higher, we narrowed our feature columns to 'Total Intl. Rigs', 'Asia Pacific Rigs', 'Brent Crude Price', and 'Latin America Rigs'.

```
#Get Highly Correlated Features
corrs = abs(df_scaled.corr()["US_Rigs"]).sort_values()

predictive_cols = []
for name, col in corrs.iteritems():
    if col > .6 or col < -.6:
        predictive_cols.append(name)

predictive_cols
```

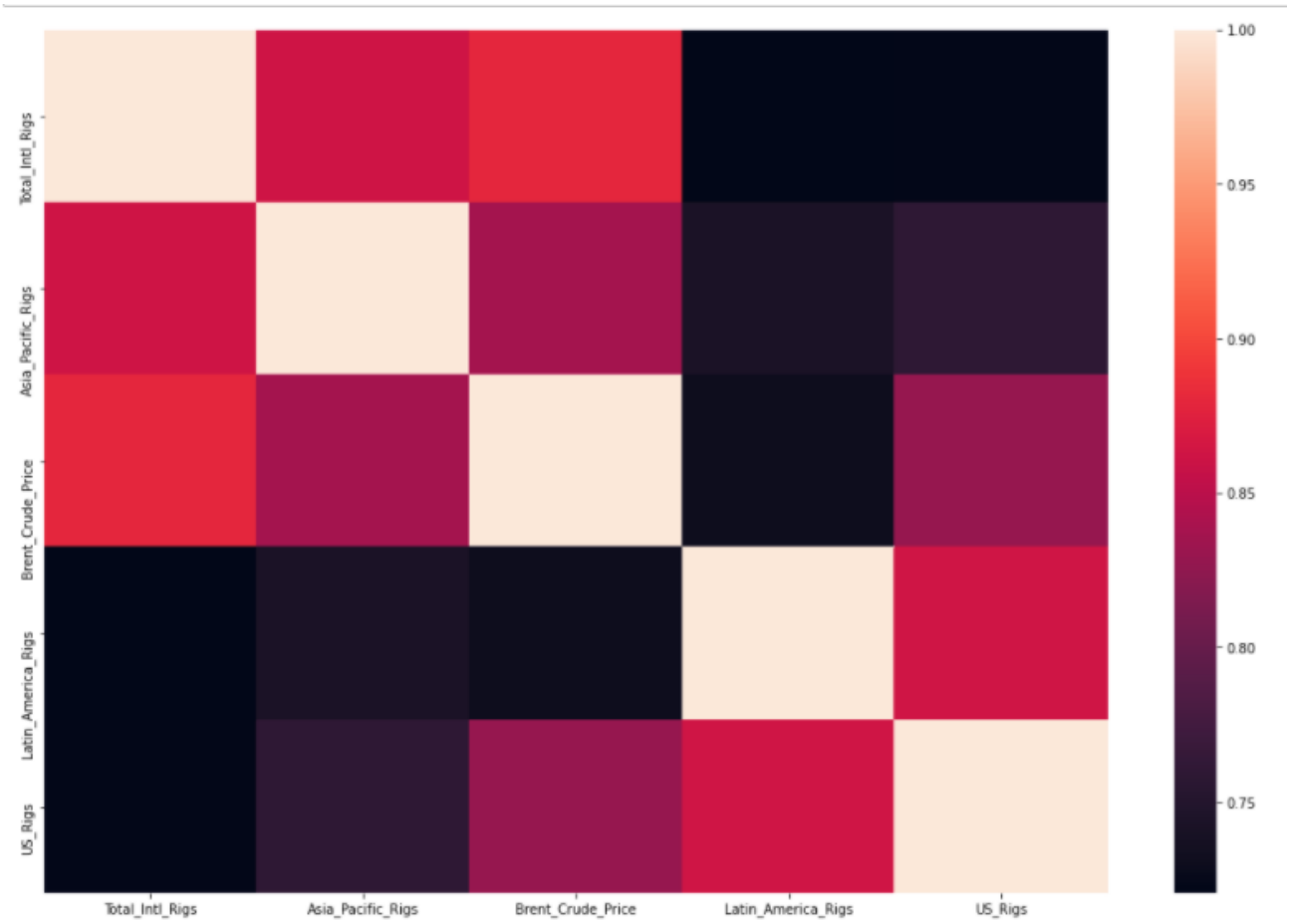
```
['Total_Intl_Rigs',
 'Asia_Pacific_Rigs',
 'Brent_Crude_Price',
 'Latin_America_Rigs',
 'US_Rigs']
```

Using our new feature columns, we ran a correlation table to further inspect the figures and ensure that we were on the right track.

Correlations

```
1 df_scaled.corr()
```

	Total_Intl_Rigs	Asia_Pacific_Rigs	Brent_Crude_Price	Latin_America_Rigs	US_Rigs
Total_Intl_Rigs	1.000000	0.861696	0.879123	0.720542	0.721919
Asia_Pacific_Rigs	0.861696	1.000000	0.837156	0.741667	0.759682
Brent_Crude_Price	0.879123	0.837156	1.000000	0.730043	0.829238
Latin_America_Rigs	0.720542	0.741667	0.730043	1.000000	0.862802
US_Rigs	0.721919	0.759682	0.829238	0.862802	1.000000



Machine Learning Regression Models

As we were looking to predict continuous values, we ran various machine learning regression models in order to find the model that produced the highest correlation and lowest error estimator combination. In order to evaluate correlation, we used the r-squared metric. To evaluate error, we used the Mean Squared Error metric. After splitting the data into training and testing data, we ran both sets of data through the linear regression, ridge, lasso, ElasticNet, decision tree, random forest, AdaBoost, GradientBoost, KNN, and SVN models.

Best Linear Regression Model: Ridge

The best linear regression model that we came across was the ridge model. The ridge model actually performed better on the testing data then the training data with a r-squared correlation of .87 and a mean squared error of 29220 when applied to the testing data, and a .81 r-squared correlation and mean squared error value of 36590 when applied to the training data.

```
#Initialize Ridge Model
ridge = Ridge()

# fit
ridge.fit(X_train, y_train)

# predict
in_preds = ridge.predict(X_train)
out_preds = ridge.predict(X_test)

#evaluate
print("Model Evaluation Report")
print(f"The In Sample R2 Score: {r2_score(y_train, in_preds)}")
print(f"The In Sample MSE: {mean_squared_error(y_train, in_preds)}")
print()
print(f"The Out Sample R2 Score: {r2_score(y_test, out_preds)}")
print(f"The Out Sample MSE: {mean_squared_error(y_test, out_preds)}")
```

```
Model Evaluation Report
The In Sample R2 Score: 0.8106999537075118
The In Sample MSE: 36509.4529768271

The Out Sample R2 Score: 0.8707207961732302
The Out Sample MSE: 29220.132409712533
```

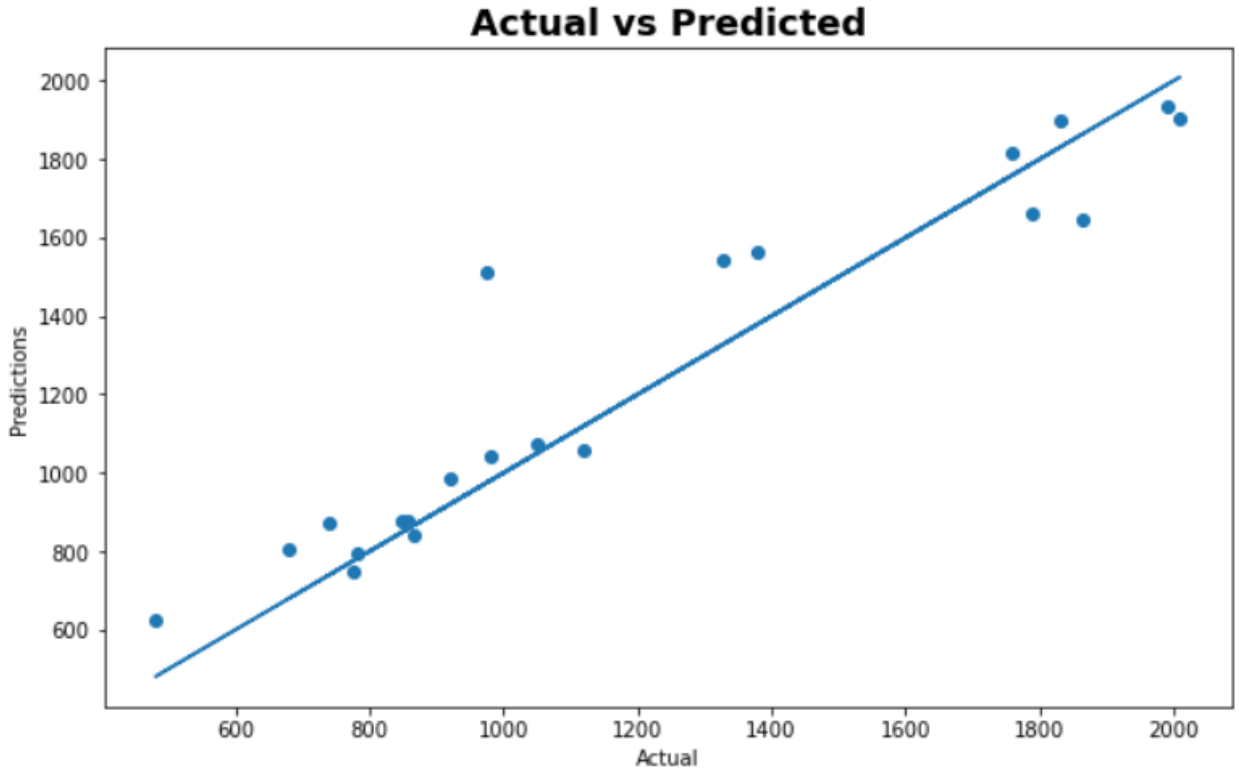
Best Tree Model: Random Forest

The best tree model, and ultimately the best model overall, proved to be the random forest model. Although it seemed that the model was overfitting the training data with a r-squared correlation value of .97 and a mean squared error of just 74, the random forest actually proved to be quite predictive for the testing data as well. Ultimately, with a r-squared correlation value of .89 and a mean squared error of

25225, the random forest model was the model we decided to choose to conduct our final analysis.

```
Model Evaluation Report
The In Sample R2 Score: 0.9714453338480108
The In Sample RMSE: 74.21058350167135

The Out Sample R2 Score: 0.8883958101001377
The Out Sample MSE: 25225.164680952377
```



Finalizing Model

Upon deciding on the random forest model, our last step in the model building process was to fully fit the data to the model and run the algorithm. We were happily surprised to see a high r-squared correlation of .98 and a relatively low mean squared error figure of 4761.

Model Selection

```
# We chose the Random Forest Model because it had one of the hig
```

```
#Initialize Random Forest Model
rf_final = RandomForestRegressor()

# fit
rf_final.fit(X, y)

# predict
in_preds = rf_final.predict(X)

#evaluate
print("Model Evaluation Report")
print(f"The In Sample R2 Score: {r2_score(y, in_preds)}")
print(f"The In Sample MSE: {mean_squared_error(y, in_preds)}")
```

```
Model Evaluation Report
The In Sample R2 Score: 0.9761492510125006
The In Sample MSE: 4761.234212621359
```

Readying for Deployment

Because data analysis is only as good as its ability to produce actionable intelligence, our work was not over. Paramount to the machine learning process is ensuring that the end user is able to successfully use the model him or herself. In order to ensure this, we tested some of our deployment code to guarantee it passed the common sense test.

Testing the Model

In our initial test, we found that we had to input scaled data into the model in order for it to produce a common sense output. We remembered that this was due to us scaling the data in the first place.

Test Model

```
my_new_model.predict(X_test)
```

```
array([1973.92,  844.64, 1097.33, 1731.36, 1396.47, 1229.3 ,  864.1 ,
       1824.97,  745.59,  943.45, 1352.88, 1974.33,  727.02,  855.36,
       1013.99, 1072.51,  774.08, 1847.47,  538.99, 1799.13,  786.8 ])
```

```
#Prediction
prediction = my_new_model.predict([[.6, .7, .5, .03]])
prediction
```

```
array([1092.2])
```

Testing the Scaler

Because of this, we realized we would have to programmatically transform the user input using the same scaler we used in our code before we entered the users input into our model.

Testing User Input Scale Code

```
#User Input Scaler - Test
user_input = [2000, 300, 70, 350]
```

```
user_input_scaled = x_scaler.transform([user_input])
user_input_scaled
```

```
array([[1.83269476, 1.15540541, 0.53450314, 0.65891473]])
```

```
new_prediction = my_new_model.predict(user_input_scaled)
new_prediction
```

```
array([1281.92])
```

Conclusion

America's booming oil industry has allowed the US to achieve a type of energy independence

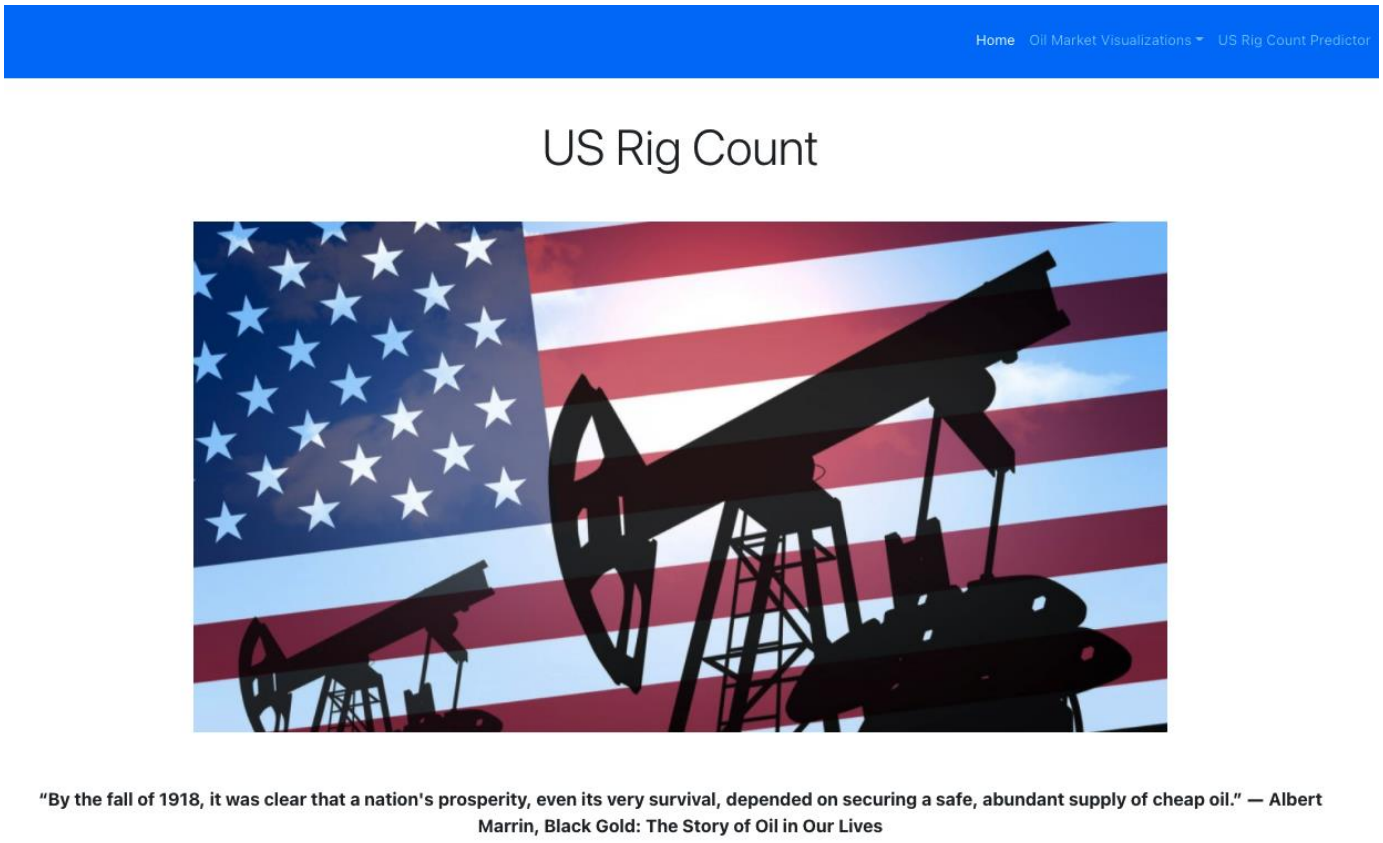
As a country, we have gone from net imports of our energy needs to exporting more than we import over the last 14 years. This is a truly remarkable shift and begs the question: where did it come from? In short, the shale revolution has created US energy independence.

Historically, American energy consumption rose as the economy and population expanded, but over the last two decades that hasn't been true. Total American energy consumption has been basically flat since the late 1990s, despite economic and population expansions. At the same time, the US production has risen significantly allowing the US to sell oil abroad.

The US Oil & Gas industry is very dynamic. Vast majority of the production is coming from shale formations requiring a lot of fixed assets (drilling rigs, hydraulic fracturing fleets). The burden of the fixed assets forces the US Oil & Gas industry to rapidly adjust every time there is a change to crude price and it’s demonstrated well in the visualizations above. The main leading indicator for Oil Field Service companies is number of drilling rigs. It’s critical to understand what the number will be for a proper business planning and development purposes.

Appendix

Website:



Resources

Data:

https://www.kaggle.com/nroll12/global-oil-supplydemand
https://www.kaggle.com/mabusalah/brent-oil-prices
https://rigcount.bakerhughes.com/intl-rig-count

Images:

US Oil
Logo: https://www.google.com/search?q=us+oil+logo&sxsrf=ALeKk011zF2CnYxEtXnT1H0-tRbbaKhGRA:1619891718694&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiIgt7zhqnwAhXhMn0KHZFdB_IQ_AUoAXoECAEQAw&biw=1536&bih=754#imgrc=lb0QCNYYPPduWM
Rig w/ American Flag
Backdrop: https://www.google.com/search?q=United+States+Oil+Picture&sxsrf=ALeKk016yVIPxq-Dphm3hO6PsgHM3noSiw:1619891899909&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiqz5LKh6nwAhW_JzQIHSLbC3UQ_AUoAXoECAEQAw&biw=1536&bih=754#imgrc=B685Sl4wa2sKvM

Inspiration:

https://public.tableau.com/en-us/gallery/uk-exports-after-brexit?tab=viz-of-the-day&type=viz-of-the-day

https://public.tableau.com/en-us/gallery/life-span-animals?tab=viz-of-the-day&type=viz-of-the-day
https://public.tableau.com/en-us/gallery/birth-control-every-age?tab=viz-of-the-day&type=viz-of-the-day