Oil & Gas Drilling Activity Prediction:

A solution for Resource Planning

Antonio Jose Dagnino Mendez
Big Data Analytics Certificate, Ryerson University
Course: CIND-820
Supervisor: Zekiye Erdem
November 29th, 2021

Ryerson University

Table of Contents

List of Tables and Figures

Abstract

The Oil & Gas industry is the primary source of energy in the United States. According to (U.S. Department of Energy, 2021), 35% of the energy consumed in the mentioned country comes from petroleum, while 34% from gas. In Addition, there are several consumer goods that come from oil such as plastics, cosmetics, gasoline, asphalt, polyester clothes, lubricants, rubbers, and many other products that are essential for today's way of living. Therefore, it is of national interest to produce and store crude oil and natural gas to always satisfy the demand.

In order to satisfy the oil and gas demand, United States produces around 11.3 million barrels per day and 82.5 billion cubic feet of natural gas based on the last Short-Term Energy Outlook from (U.S Department of Energy, 2021). These amounts are obtained and sustained based on the drilling activity across all the country.

The drilling activity consists in the exploration and development of wells, with operational, tasks like directional drilling, cementing, casing running and completions. After the well is put in production, the oil and gas are extracted and refined before they turn into the final product. However, there are unexpected changes that oil and gas companies usually face while developing wells, which makes the planning of resources extremely challenging. Those unexpected changes could be operationally related due to geological uncertainties, equipment failure, well control, unexpected subsurface conditions, etc. or economically related based on supply, demand, market prices, imports, exports, storage, CO2 emissions, climate change, etc. Consequently, oil and gas companies often struggle to determine the following: how many resources they need to satisfy job demand? how can they optimize the utilization of resources, lower the costs and plan efficiently? and last, how can they prepare for the lookahead? Failure to properly plan resources often leads

into service quality, environment and safety incidents, non-productive times, loss of revenue and several other undesired outcomes.

As a result, if these companies could predict the drilling activity based on the unexpected changes mentioned above, the planning of resources could be significantly optimized, and many of the issues could be prevented and mitigated through a most efficient and reliable planning tool. To be more specific, the drilling activity could be predicted if we had the rotary rig count in the United States in the future. Since historical data of variables like rig count, CO2 emissions, oil and gas prices, imports, exports, storage amounts, production and consumption are well known, tracked in time, and publicly reported by the U.S. Department of Energy, we could create a model of Predictive Analytics for pattern mining / time-series using general forecasting models like Auto Regressive Integrated Moving Average (ARIMA) or vector auto regression (VAR). Then, the models could be compared to take the one that performs best. These methods could be created using Python libraries like 'pdmarima, 'statsmodel', 'Pandas', 'NumPy', 'Matplotlib', 'Seaborn' and 'Requests' to get the data from (U.S. Deparment of Energy, 2021) Open Data portal (https://www.eia.gov/opendata/). The final output would be a number, dependent of time in the future, that represents the number of rotary rigs operating in the United States at a given point in time.

To summarize, the Oil & Gas industry is vital for the supply of energy and consumer goods to the population nowadays. However, operations are dramatically affected by several factors that fluctuate in time with random patterns and in exchange, causes the operational planning of resources to often get compromised with unexpected results. To overcome this challenge, predictive analytics could be applied to make estimates of rotary rig count inside United States in the near future using time-series data available.

Literature Review

The energy demand is satisfied through a combination of sources currently available in the United States. These sources of energy are based on coal, nuclear, renewables and the biggest contributor, fossil fuels. Oil and gas account for 69% of the supply of the overall demand, which means the activities involved in the exploration and development of oil/gas wells (often known as drilling activity) are of high national interest. Therefore, it is often observed the drilling activity is closely monitored by government officials and corporations to ensure success in the operations, estimate resources and project management.

The drilling activity is usually represented through the rig count. The rig count is essentially what it sounds like, a tally or count of rigs. According to (Brenner, 2008), from the Department of Natural Resources in the state of Louisiana; "when it comes to analyzing drilling activity, rig count is a simple and accurate method. In fact, it is widely used due to its simplicity. Rig count is a present and solid gauge for drilling activity". At the end, what we know about the topic, is the rig count could work as a dependent variable, because it could be used by oil companies to predict overall drilling activity.

Nevertheless, this value is difficult to estimate given the fact the fossil fuels industry is very unstable. Based on (Woods, 1988), U.S. drilling activity has fluctuated substantially over the past 40 years. This constant fluctuation is determined by several factors, including economic, political, and social. To mention more in detail, oil and gas prices, energy policies, success in exploration of oil and gas wells (proven reserves), production, imports, etcetera, are the variables influencing in rig count fluctuations. Moreover, economically speaking, the demand and supply are other elements that dramatically influence rig counts, since they directly affect oil gas market prices. The U.S. Department of Energy shows that oil price is driven by the supply and demand

from organizations of petroleum exporting countries (OPEC), and organizations for economic co-operation and development (OECD). In other words, it can be said the spot price is an outcome of the worldwide supply/demand of oil and gas. Overall, the rig count has fluctuated over time due to several factors. Therefore, we could critically say that if we use the information available in Energy Information Administration API website to obtain the values those factors (oil price, gas price, imports, exports, proven reserves, demand, supply, storage, etcetera), then we should be able to create a model that helps us forecast the rig count using time-based data.

Although rig count prediction using applied analytics does not seem it was done in the past, several sources use rig count to forecast drilling activity, calculate oil and gas market prices and evaluate overall status of the industry. For example, (Seeking Alpha, 2015) released an article that analyzes the relationship between rig count, oil price and production to predict in which direction the rig count will move next. It can be inferred from the article that production, storage, spot prices are related in a positive and negative way, thus supporting the use of those factors as features for rig count prediction. Also, another example is observed in the U.S crude oil production forecast for 2021 provided by (U.S Department of Energy, 2020), where it is observed how the rig count and West Texas Intermediate crude oil prices were used for this analysis. In general, several magazines, web sites, financial studies, and other sources of information use rig count as an indicator for the industry outlook and define whether it will increase or decrease. However, they often are interested in the trend more than the actual number without applying predictive analysis for this. Instead, they account for financial and industry advisors that manually evaluate trends and factors that influence the overall behavior of the drilling activity before determining the forecasts.

Forecasting using time-based data is a well-known procedure in the world of data analytics. In fact, there are several forecasting models and methods that are used nowadays such as vector

Auto Regression (VAR), Holt-Winters or Auto Regressive integrated Moving Average (ARIMA). In addition, we could use deep learning for time-series forecasting. There are several projects that were done in the past using time-series data to make predictions. For instance, a former Data Scientist from BP (Perry, 2019), made a project that predicts electricity prices in time (url: https://github.com/kperry2215/electricity_price_time_series_analysis). The mentioned project pulls data from Energy Information Administration API to perform time series analysis on it, including time series decomposition and vector Auto Regression for forecasting. More in detail, Perry used Python as programming language, and she used several libraries to finish the project. She used 'eia' library to obtain the data from U.S. Department of Energy API. Also, she handled the data using 'pandas' and 'matplotlib' to visualize the data. Then, she used 'statsmodels' library to call functions that allows the calculation of trends, filters and create the vector autoregression. Finally, Perry used 'scikit-learn' to apply evaluation metrics tools like mean squared error and mean absolute error to determine overall performance of the model.

Similarly, (Taylor, 2019) did a project to forecast future values of electricity loads using a dataset that contained 3 years of hourly electricity load and temperature values. However, he used Deep Learning for this. He got the data from an API, then prepared the data, trained the data, implemented a convolutional neural network, and enabled early stopping hyperparameter to reduce the likelihood of model overfitting (https://github.com/Azure/DeepLearningForTimeSeriesForecasting/blob/master/1_CNN_dilated.ipynb).

Both approaches are able to be used both for the current project. However, there might be advantages and disadvantages in taking one or the other. Talking about forecasting models, we have enough tools that can help us making good predictions, several information of the procedure

is found online and even projects with time series data are available. Also, these forecasting models offer more simplicity and easier interpretation of what the overall algorithm is doing. But it has some constraints. These models are not 100% accurate. They do have a margin of error in the predictions depending on the forecasting calculation used. Thus, it should be used wisely to avoid issues or concerns.

On the other hand, when it comes to the deep learning approach, the model learns overall behavior of variables to be predicted and although they tend to provide accurate results, they need large amounts of data so the neural network could be properly trained. Besides this need, the complexity increases, and the interpretation of results are often difficult to explain.

After having a big picture of both approaches, we will use different forecasting models like vector Auto Regression (VAR), vector auto Regression Moving Average (VARMA), Auto Regression Integrated Moving Average (ARIMA) and Holt-Winters forecasting models since we do not have a large dataset for deep learning. At the end, we will compare all models before getting a conclusion on what model to use. The overall time-series forecasting methodology consists of doing exploratory data analysis, data modeling, and evaluate models against test sets to get metrics. Then, the model is re-fit based on the entire data set before forecasting for future data.

To sum up, if we ask if someone else did anything related before, the answer is yes. As matter of fact, forecasting using time-series data is a popular procedure and has many tools that exist nowadays to help make better predictions. Then, going back to Drilling Activity Prediction project, we could use any of the models mentioned before, and even use some of them to finally compare their evaluation metrics to see which one performs better given the existing dataset. Either way, since the dataset does not have substantial amount of data to properly train a neural network, we advocate for using forecasting models like VAR, VARMA, ARIMA and Holt-Winters.

The Oil and Gas industry has been considerably unstable over the last 40 years or more, and most likely will continue the same way. One example happened last year, where the Covid-19 caused the drilling activity to drastically drop until the end of year. As of right now, it is slowly recovering, but would be of great support to determine the drilling activity recovery rate for better business preparedness. It is important to mention that all entities involved in this industry are executing their duties in a reactive manner instead of a proactive manner. The reason is because still there is no tool that is accurate enough to give an indication of how activity will behave in the long term. Consequently, the elaboration of a drilling activity prediction model would be of great fit to help the oil and gas industry smoothen all fluctuations in activity.

The development of a model that predicts drilling activity in the future will allow the entire industry to have better operational planning, assign resources efficiently and execute operations avoiding several issues that leads to loss of revenue, increase in non-productive times, service quality incidents, wrong assignment of resources, etcetera. Several oil and gas journals, magazines and web sites use the rig count to assess overall industry health and give a short-term outlook. Nonetheless, an automated model will allow organizations not only to do this but could be further adapted to their individual conditions to help them improve their project management. The value of this research will create a precedent regarding the evaluation of oil and gas activity and improve overall performance of operations through better planning. As a result, it is worth kicking off and delve into the matter to make progress in future operations.

Data Description

The data is obtained from EIA API portal. Also, it is important to mention that there is no pre-built data set. Instead, we need to separately obtain the features that we consider relevant to

the investigation, and then merge them all based on time. After doing some research and using domain expertise in the topic, the dataset built is the following:

	Description	Units	Data Type	mean	std	min	max
date	MM/DD/YY	-	datetime	-	-	-	-
oil_price	spot price in time	USD	float	44.4800	28.7184	-36.98	145.31
gas_price	spot price in time	USD	float	4.14823	2.18031	0	23.86
oilrig_count	active oil rigs	-	integer	497.638	361.820	108	1596
gasrig_count	active gas rigs	-	integer	586.156	388.729	70	1585
totalrig_count	total rig count	-	integer	1390.76	746.262	250	4521
onshore_count	onshore rigs	-	integer	1286.39	705.074	237	4238
offshore_count	offshore rigs	-	integer	104.388	63.5545	12	283
totalwell_count	total active wells	-	integer	3482.34	1654.49	1268	8556
oilwell_count	oil wells	-	integer	1363.82	898.651	291	3945
gaswell_count	gas wells	-	integer	1259.94	626.667	454	3200
oilimports_amount_x	oil imports monthly	Thousand barrels per day	float	9150.44	2478.59	3689.51	14696.6
oilimports_amount_y	oil imports weekly	Thousand barrels per day	float	10423.7	1853.61	5775	15217
oilexports_amount_x	oil exports monthly	Thousand barrels per day	float	1944.59	2251.62	155.774	9589.40
oilexports_amount_y	oil exports weekly	Thousand barrels per day	float	2545.68	2363.00	723	10134
oilproduction_amount_x	oil production monthly	Thousand barrels per day	float	7599.54	1772.41	3973.58	12966.1
oilproduction_amount_y	oil production weekly	Thousand barrels per day	float	7338.09	1897.94	3813	13100
oilconsumption_amount_x	oil consumption monthly	Thousand barrels per day	float	18316.9	1682.56	14504.7	21666.0
oilconsumption_amount_y	oil consumption weekly	Thousand barrels per day	float	19210.4	1382.68	13797	22820
oilstorage_amount_x	oil storage monthly	Million Barrels	float	839.682	261.119	233.035	1230.10

	Description	Units	Data Type	mean	std	min	max
oilstorage_amount_y	oil storage weekly	Thousand barrels	float	945275.	123539	607781	1227678
oilstoragechange_amount	change in storage	Thousand barrels	float	1206.18	11206.9	-34924	48553
oildaysofsupply	oil days of supply	days	integer	23.4161	4.0545	16.1	42
gasimports_amount	gas imports	Billion cubic feet	float	202.998	103.105	42.512	426.534
gasexports_amount	gas exports	Billion cubic feet	float	70.4393	111.209	1.614	595.375
gasproduction_amount	gas production	Billion cubic feet	float	91.3071	40.3872	59	244.674
gasconsumption_amount	gas consumption	Billion cubic feet	float	1851.26	472.992	939.93	3417.27
gasstorage_amount	gas storage amount	Billion cubic feet	float	6492.78	829.360	4446	8384.08
uspopulation	us population	thousand	integer	253787	44937.7	180671	328330
usemissions	us CO2 emissions	million metric tons CO2	float	2269.59	162.139	1978.17	2594.01

Table 1 Descriptive statistics of data set

The code for the built dataset could be found at https://github.com/antoniodagnino/Oil-Gas-Drilling-Activity-Prediction.

Methodology

The development of a forecasting model is divided into different levels or stages that will enable better understanding of the project, flexibility to make modifications, and organization of the tasks been performed. The methodology used for time series forecasting will be done by going through the following steps shown in the diagram below:

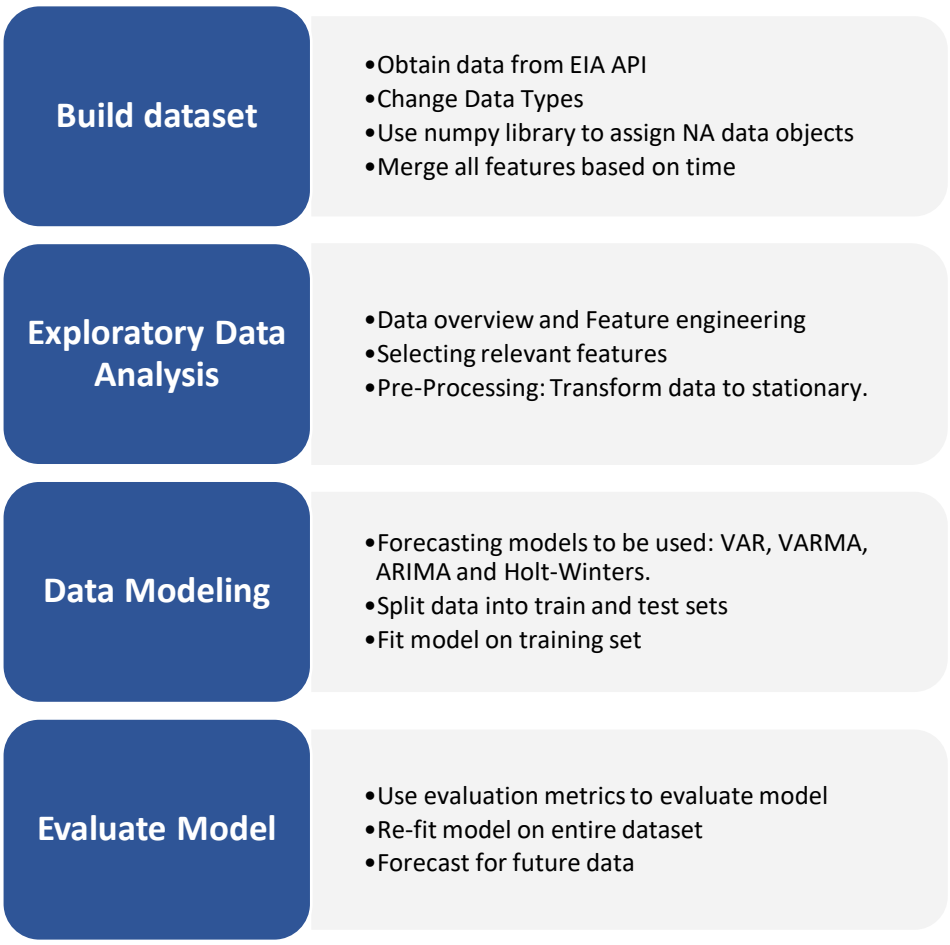

Figure 1 Methodology workflow

First, the data is built by retrieving information from Energy Information Administration to our programming environment through API's. We will use Python as programming language and Jupyter Notebook as our Integrated Development Environment (IDE).

Then, the data is further explored to determine whether it could be used or not for Machine Learning. Once those results are obtained, it is studied what feature engineering actions are needed to fix the data and make it readable for forecasting models.

Finally, the data is split into train and test sets so models can be avaluated. Those models are tested against known data to see their overall performance using specific evaluation metrics. The models to be used will be VAR, VARMA, ARIMA and Holt-Winters.

Build dataset

To build the dataset, we obtain data online from EIA API portal (U.S. Deparment of Energy, 2021), and all variables are merged into a single data frame using a common time stamp. For this, we need an API key, and the time series ID of the data. The API key is obtained for free, and it's just required to input an email address inside EIA website. Then, we use the 'requests' Python library to get the data. Initially, the data is obtained in json structure and handled with Python like if it was a dictionary object.

Since we need to get data from several time series, we create a user-defined function called 'download_convert' that will obtain the data for us, and it outputs a dataframe object with two columns: date and feature name. To make the function work properly, the user has to input the series ID to be downloaded, desired time format and column names. Then, the function will attempt to download the data, transform the time variable to date time object with the specified time format, deal with null values by assigning 'NA' objects, and convert string type numbers into float data types before turning the entire timeseries into dataframe.

Finally, all data frames are merged into a single data frame that contains all features. The merge is based on the date column and by keeping values on both data frames in every single time data point. If some data is missing in one of them, a null data value will be put instead.

To summarize, the data is obtained and built by following the following steps:

1. Obtain EIA key: Go to EIA website, get API key by following the instructions.
2. Search variables in EIA website and obtain series ID of chosen variables.
3. Create function that will download the data and convert into dataframe
4. Merge all data frames into one single data frame based on date and keeping data from both data frames (outer merge type).

5. Export data frame in csv file so data is available offline (optional step).

Exploratory data analysis

After data is obtained and in data frame format, further analysis can be performed. The data is studied in higher level of detail, and several functions are applied to the data to alter it in a way that forecast models can interpret. Also, statistical tools are used to determine the influence of the features in the dependent variable. The Exploratory data analysis involves data overview and feature engineering, feature selection and pre-processing activities. The result is a data frame with sufficient data, without null values, with proper data types and transformed in a way that forecasting models can use to train.

To begin with, the data is explored by using Python libraries like 'Pandas', 'Matplotlib', 'Seaborn' and 'Numpy'. More in detail, the head of the data is observed, descriptive statistics, missing data, and data types. The index of the data frame is the date, and the columns are the features. By having a first look into the data set, we can identify that the potential dependent variables for our model could be 'Oil Rig Count', 'Gas Rig Count, or 'Total Rig Count'.

date	Oil price	Gas price	Oilrig count	Gasrig count	Totalrig count	Oilstorage change amount	Oil days of supply	Gas imports amount	Gas exports amount
10/25/2021	84.64	5.72	407	101	508	-9035	28.3	227.779	565.564
10/24/2021	84.60333	5.513333	407	101	508	-9035	28.3	227.779	565.564
10/23/2021	84.56667	5.306667	407	101	508	-9035	28.3	227.779	565.564
10/22/2021	84.53	5.1	407	101	508	-9035	28.3	227.779	565.564
10/21/2021	82.64	4.94	407	101	508	-9035	28.24286	227.779	565.564

Table 2 Data Overview. Last rows of data set and 9 out of 25 columns

Moreover, it was noticed some constraints in the data, which are missing values, different units, and redundant features. All those constraints are addressed before data can be processed, so feature engineering is applied for this.

Regarding missing data, this is identified by plotting a heatmap with 'Seaborn' library. It can be observed there are several null values in the raw data. The main reason is because features have different time frequencies. Our data set time frequency is in days, and some features have data points updated in either weekly or monthly frequencies. This problem is addressed with interpolation, merging some features, resampling time series, and dropping remaining null values. However, before interpolating features it is important to identify their units and make sure they are not cumulative values. Cumulative values cannot be interpolated. Instead, a function should be created to evenly distribute cumulative values in a specific time interval. In other words, this could be explained with the following example:

"If U.S. produces 12 million barrels per day, that means 360 million barrels per month". If the production variable is in barrels per day, we will be able to interpolate since it will fill average values in gaps. However, if it's cumulative, we'll need to apply the cumulative sum of the values (12 on day one, 24 on day 2, 36 on day 3 and so on). If that is the case, interpolation is not the right tool to be used. After confirming the features are in rate expression, we applied interpolation to fill missing data in between data points.

Besides interpolating data, it was noticed some features that were basically the same information but represented in different time frequencies. For instance, 'oil imports amount x' and 'oil imports amount y' were both same data but one represented weekly and the other monthly. However, the one expressed monthly had data from 1973 to 2021 while the one expressed weekly had data from 1991 to 2021.

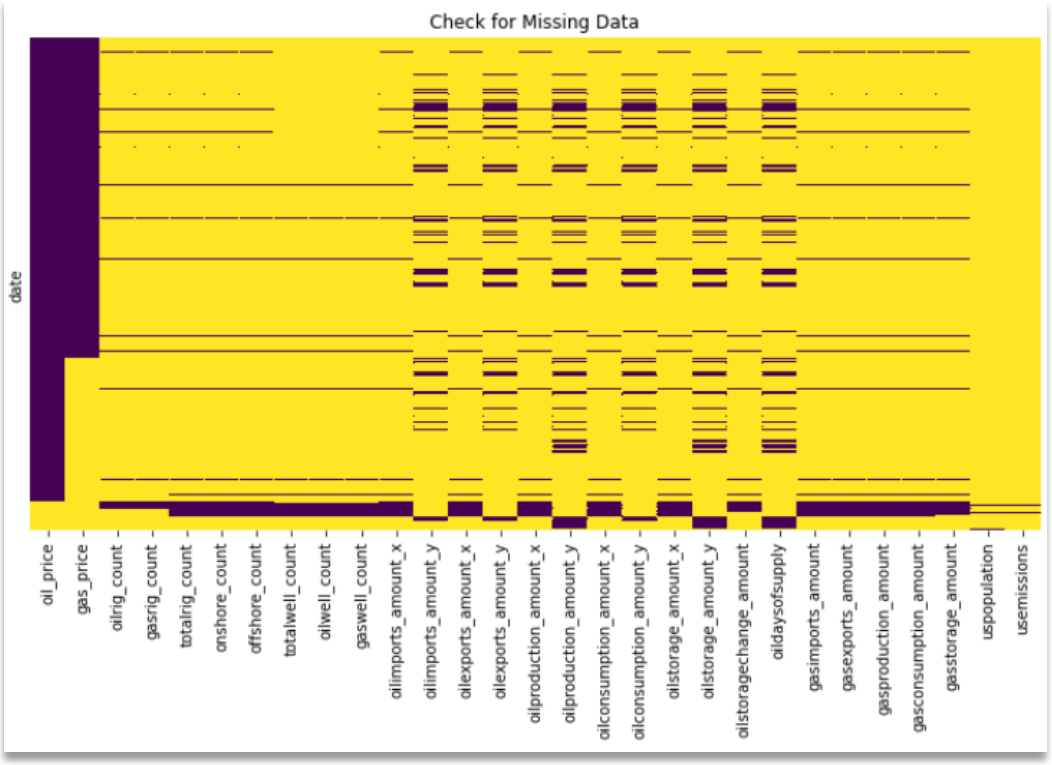

Figure 2 Heatmap that shows missing data raw dataset

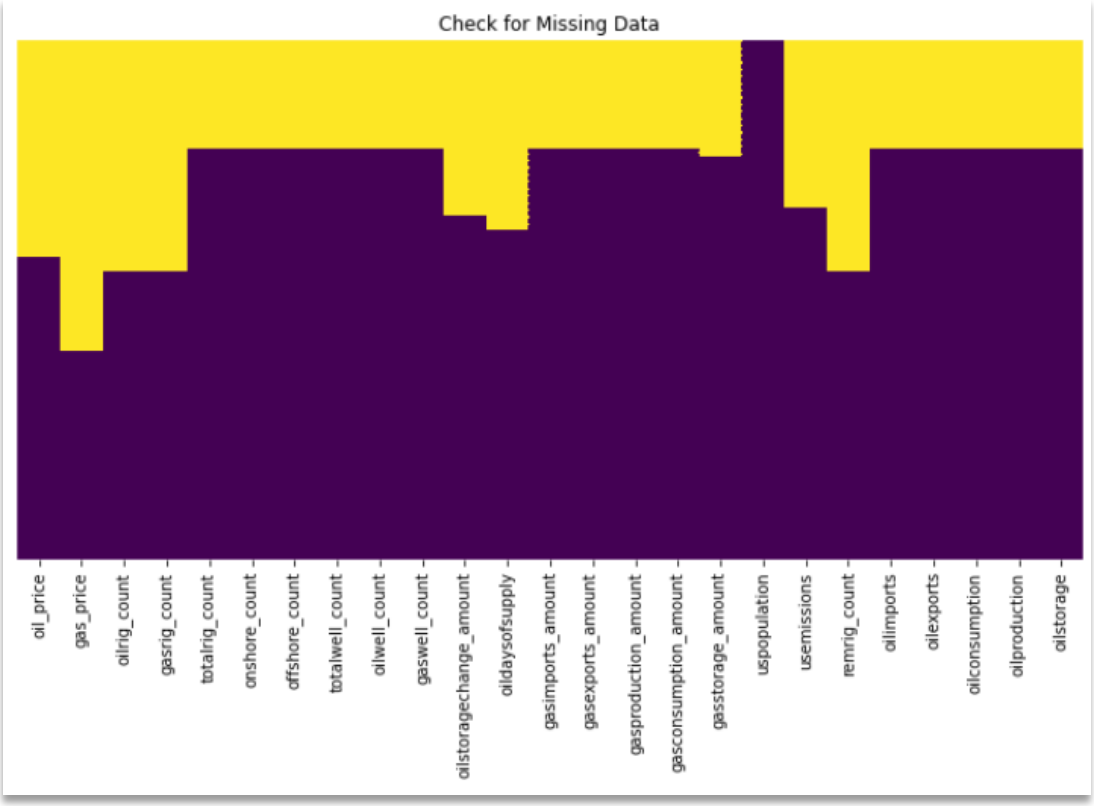

Figure 3 Missing Data heatmap after some feature engineering

Therefore, we created a user defined function to merge these columns and have more data density. The function called ‘mergecolumns’ basically joins these two features into one. Also, it takes preference in keeping weekly data points in case there is data on a same point in time.

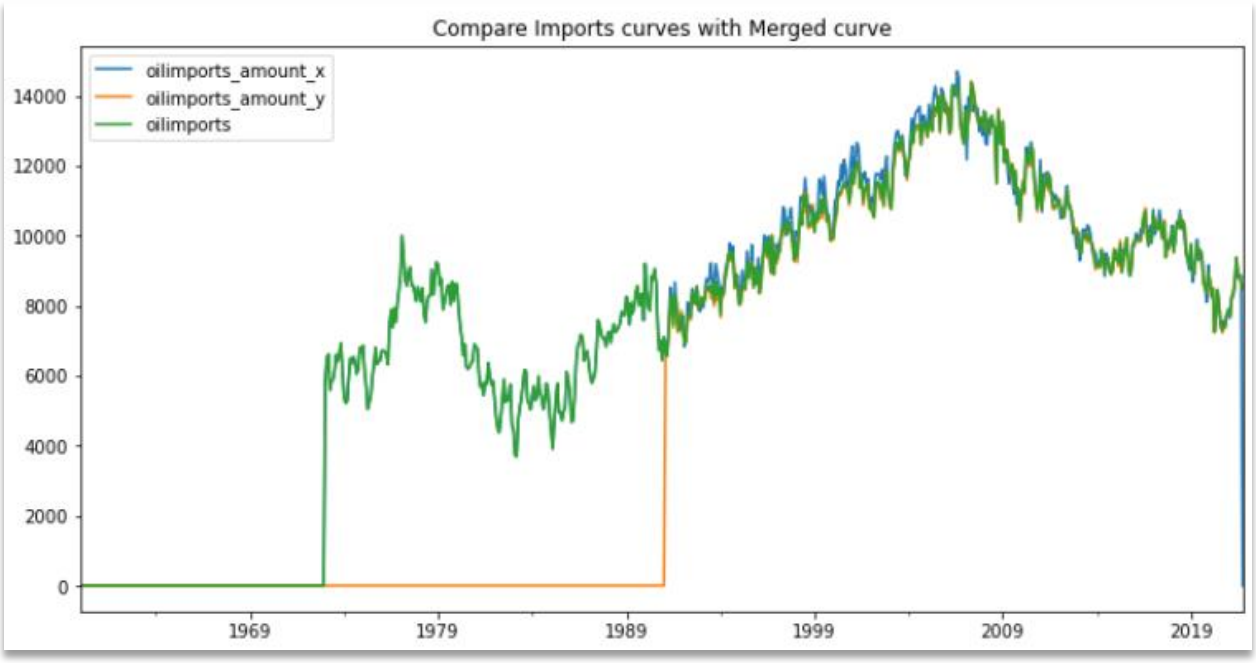

Figure 4 Oil Imports features merged into a single feature (green curve).

To end with dealing with missing data, the values that remain null after applying interpolations and column merge can be solved by either dropping them or doing ‘Backcasting’. Since the null data is data in the past (starting from 1960), an auto-regressive model could be applied to these features so data will be filled with moving averages that will follow trends and seasonality from the same feature. This process is same as forecasting but backwards. To sum up, missing data could be solved by interpolating data, create user defined functions, dropping null values or doing other mathematical applications such as applying Moving Averages.

After that, we continued doing feature engineering to data that had different units. For example, ‘Oil Storage Amount X’ and ‘Oil Storage Amount Y’ not only had the same time frequency

problem that we discussed earlier, but the units were different. While ‘Oil Storage Amount Y’ is in Thousand Barrels, ‘Oil Storage Amount Y’ is in Million Barrels. Thus, before merging the columns we used lambda function to convert ‘Oil Storage Amount X’ in Thousand Barrels. Once both features had same units, ‘mergecolumns’ was used one more time to get a feature with higher density.

After data is free of null values and feature engineering was applied, it is time to determine what features are important for the forecasting model. In other words, we need to select features that will help models perform better. For this, we use statistical tools like correlation and Granger Causality Tests. The features with highly correlated (negatively or positively) will be chosen. Then from those chosen, the Granger Causality test will determine the best ones to be used to predict fluctuations in Rig Counts.

Firstly, we use ‘Pandas’ correlation function on the data frame and ‘Seaborn’ heatmap one more time to have a visual appreciation of the correlations between features. At this point we discovered that there were some features that correlated with ‘Oil Rig Count’ and other features that correlated with ‘Gas Rig Count’. As a result, we decided to split the dataset into two, and thus have two dependent variables, ‘Oil Rig Count’ and ‘Gas Rig Count’. In other words, we will predict ‘Oil Rig Count’ and ‘Gas Rig Count’ separately with different features.

Secondly, we apply Granger Causality Tests to the selected correlated values since correlation does not mean causality. The Granger causality test is a hypothesis test to determine if one time series is useful in forecasting another. While it is easy to measure correlations between series – (when one goes up the other goes up, or vice versa), it's another thing to observe changes in one series correlated to changes in another after a consistent amount of time.

This may indicate the presence of causality. For instance, changes in the first series influenced the behavior of the second. However, it may also be that both series are affected by some third factor, just at different rates. Still, it can be useful if changes in one series can predict upcoming changes in another, whether there is causality or not. In this case we say that one series "Granger-causes" another. In the case of two series, 'y' and 'x', the null hypothesis is that lagged values of 'x' do not explain variations in 'y'. In other words, it assumes that 'xt' doesn't Granger-cause 'yt' (null hypothesis).

Oil Rig Count Correlation	Gas Rig Count Correlation
oilrig_count 1.000000	gasrig_count 1.000000
oil_price 0.668226	Oilimports 0.882810
oilwell_count 0.661305	gasimports_amount 0.838126
onshore_count 0.576134	gaswell_count 0.738234
uspopulation 0.566449	gas_price 0.732487
totalrig_count 0.534911	totalrig_count 0.591709
Oilstorage 0.434567	onshore_count 0.547746
Oilexports 0.361687	remrig_count 0.494310
Oilproduction 0.331508	offshore_count 0.471420
gasproduction_amount 0.302828	totalwell_count 0.471094
gasconsumption_amount 0.272281	Usemissions 0.386313
gasexports_amount 0.237308	Oilconsumption 0.241871
gasstorage_amount 0.217094	oil_price 0.169827
Oildaysofsupply 0.178339	oilstoragechange_amt 0.064229
totalwell_count 0.150101	gasstorage_amount -0.198323
remrig_count 0.090129	oilwell_count -0.280672
oilstoragechange_amount -0.001197	oilrig_count -0.364592
Oilconsumption -0.159903	gasconsumption_amount -0.441030
gas_price -0.197755	Oilstorage -0.457413
gaswell_count -0.201369	Uspopulation -0.506626
gasrig_count -0.364592	gasexports_amount -0.622342
Oilimports -0.379046	Oildaysofsupply -0.677238
offshore_count -0.477353	Oilexports -0.733829
gasimports_amount -0.525591	gasproduction_amount -0.759580
Usemissions -0.696405	Oilproduction -0.826225

Table 3 Correlation Table. Highly correlated features highlighted in red.

In order to evaluate all features against the dependent variable ('Oil Rig Count' or 'Gas Rig Count'), a for loop was created. The loop will take feature by feature in a list type variable and compare against the dependent variable using 'grangercausalitytests' function available in 'statsmodels' library. The output will show p-values. Essentially, we are looking for p-values close to '0', meaning we can reject the null hypothesis and thus say the feature been evaluated has 'granger causality'. After running the loops, we concluded the features to be used are the following:

Oil Rig Count	Gas Rig Count
Oil Price Oil Consumption Oil Production Oil Storage	Gas Price Gas Wells Count Gas Imports Gas Consumption

Table 4 Granger Causality Test Results

To end with the feature selection process, the correlation results helped determine what features to be used in a primary stage and split our dataset into two that will predict two variables: 'Oil Rig Count' and 'Gas Rig Count'. Then, the Granger Causality test helped filter further the features and choose the ones that affect the trends of the others. With the resulted data sets, the final pre-processing activities can be applied.

Before fitting the forecasting models, we need to know that models like VAR, VARMA assume features are Stationary. As a result, we need to make one more test to determine whether or data is stationary or not. A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary. Stationarity can be checked by using 'Augmented Dickey Fuller' test statistic. Essentially, this is another statistical tool based on Hypothesis Testing that will determine whether the feature is stationary or not. In case data is not stationary, we need to transform it so it becomes stationary.

The procedure of transforming the data is called Differencing. After creating a function that has a loop that tests every single feature for stationarity using ADF function available in 'Statsmodels' library, we realized that some features were non-stationary. As a result, we needed to apply Differencing. Differencing consists of computing the differences between consecutive observations. This can be done with 'Pandas' or 'Statsmodels' libraries. The differenced df is stored in a new variable so original values are kept for later use when the transformed data frame needs to be transformed back into its original version. The differenced data frame will be the one used to fit the VAR and VARMA models.

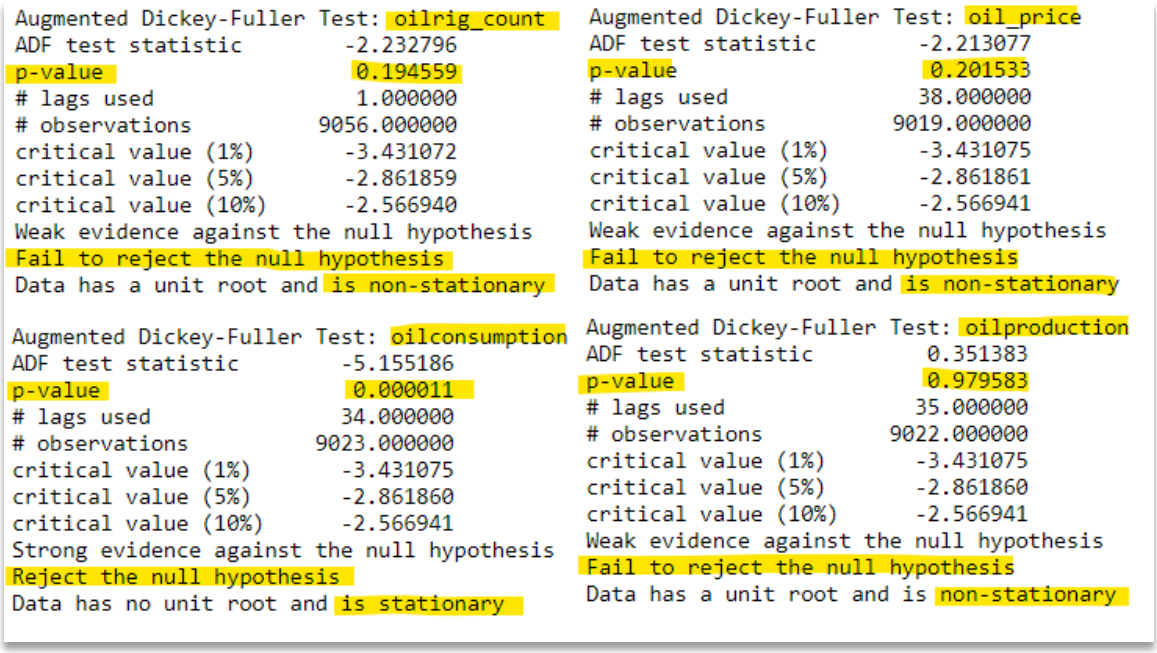

Figure 5 ADF Test Results on four features.

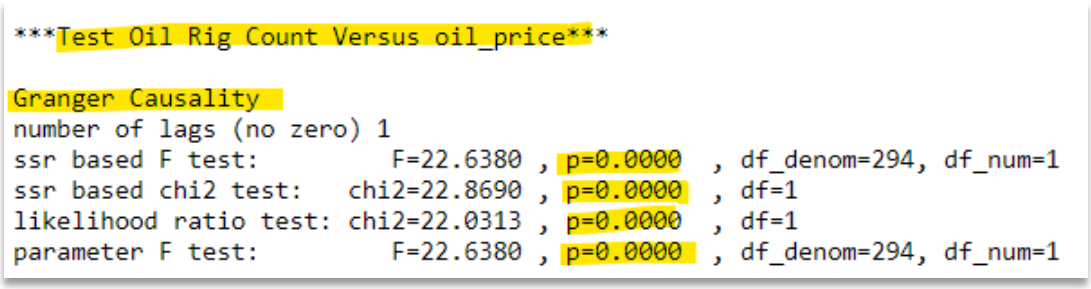

Figure 6 Granger Causality Test: Oil Rig Count Vs Oil Price.

To conclude with the Exploratory Data Analysis, there were 3 main activities: Data Overview and Feature Engineering, feature selection process and pre-processing activities. In the data overview stage, we identified are dependent variables, independent variables, and some constraints in data: missing values, data in different units and redundant data. All these constraints were addressed with user defined functions and interpolations. After that, the selection of relevant features was done with correlation functions and Granger Causality Tests. Finally, the data was transformed to its stationary version by using Augmented Dickey-Fuller Tests and Differencing.

Exploratory Data Analysis	
Steps	Description of Activities
Data Overview and Feature Engineering	Identify Missing Data Identify features with different time frequencies Identify features with different units Apply Interpolations Apply user defined function to merge columns Apply lambda function to change scale of feature
Feature Selection Process	Use of correlation function Use of Granger Causality Test
Pre-Processing	Augmented Dickey-Fuller Test Apply Differencing (Transform data to stationary) Save transformed data frames and original data frames

Table 5 Steps Performed in Exploratory Data Analysis.

Data Modeling

The data from EIA was analyzed, missing data replaced, units were adjusted, overall data cleaned and then, it was split into two data sets: ‘Oil df’ and ‘Gas df’. These data sets were prepared to fit the forecasting models to make predictions of the Oil Rig Count and Gas Rig Count respectively. For this purpose, four forecasting models were created and used to test their

performance against the data. These models are Vector Auto Regression (VAR), Vector Auto Regression Moving Average (VARMA), Auto Regressive Integrated Moving Average (ARIMA) and Holt-Winters. Overall, the procedure for forecasting with these models include split the data into train and test sets, define hyperparameters to be used in each of the models, fit the models with training dataset, forecast against test dataset, re-fit with the entire data set and create real forecasts. Finally, evaluation metrics and plots are observed to interpret the results and make conclusions.

The train-test split in time series is done without shuffling data (not random split). Instead, the data will be split in a specific point in time. Then the test set will be the interval with most recent data whereas the train set will have the oldest data. Moreover, as a rule of thumb the test set usually has the same time interval that we are planning to forecast or predict. For instance, if we want to predict 12 months in the future, then our test set should have at least 12 months of data (or 1 year). In other words, the test set should be as large as our predictions in terms of time. Thus, we created a variable called ‘nobs’, which stands for ‘number of observations’, and will represent the number of data points the test set will have. Since data sets are resampled to monthly frequency, then nobs will represent the number of months to predict and the number of months that the test set will have. For this project, we will use 12 months in our train-test split for all models. In that way, we can compare evaluation metrics at same time interval predictions.

The first model used; vector Auto Regression (VAR) is a statistical model used to capture the relationship between multiple variables as they change over time. To explain in more detail, VAR will use the past data (also known as lagged values) of the dependent variable and independent variables to create forecasts. The model assumes the variables affect each other (or can be hypothesized to affect each other), and their behavior in time is stationary. At the end, the

variables are collected in vectors of K-dimensions, being k equal to the number of variables in the model.

When coding, we instantiate the VAR model, and we pass in the transformed train set (dataset transformed to its stationary form). Then, we fit the model with the right hyperparameter values. In this case, we assign the 'maxlags' value based on the lowest Akaike Information Criterion (AIC). AIC evaluates a collection of models and estimates the quality of each model relative to the others. Also, it penalizes for the number of parameters used to thwart overfitting. Essentially, AIC is a common metric that can be used to evaluate the performance based on multiple numbers of 'maxlags'. After fitting the model with several 'maxlags' values, it was observed the number that best performs is 4 (order is 4), because it had the lowest AIC value. Once the VAR model is fitted with proper hyperparameter values, it is ready to make predictions by using the forecast method. However, we might not be able to properly interpret the values because we need to make two more things: assign timestamps to the forecasted values, and invert transformation to its original form. For that reason, we reindex the series output given by the model with the timestamps of the test set. From there, we invert the differencing values of the features we need so we can interpret the real results (Oil Rig Count). Finally, we plot Test set Versus the forecasts, we use the RMSE, MSE and MAE functions from 'statsmodels' to see our first results before re-fitting the model again using the entire dataset. With the entire dataset, we will be able to forecast to the real future instead of the test set.

The vector Auto Regression model makes predictions based on past data of all the features in the dataset, and the result is a 'Numpy' series data type object. Depending on the time frequency, we reindex this series output with the months, years or days that were used, and from there we back

transform the data set to its original form. In this case, we requested the model to make 12 months forecast.

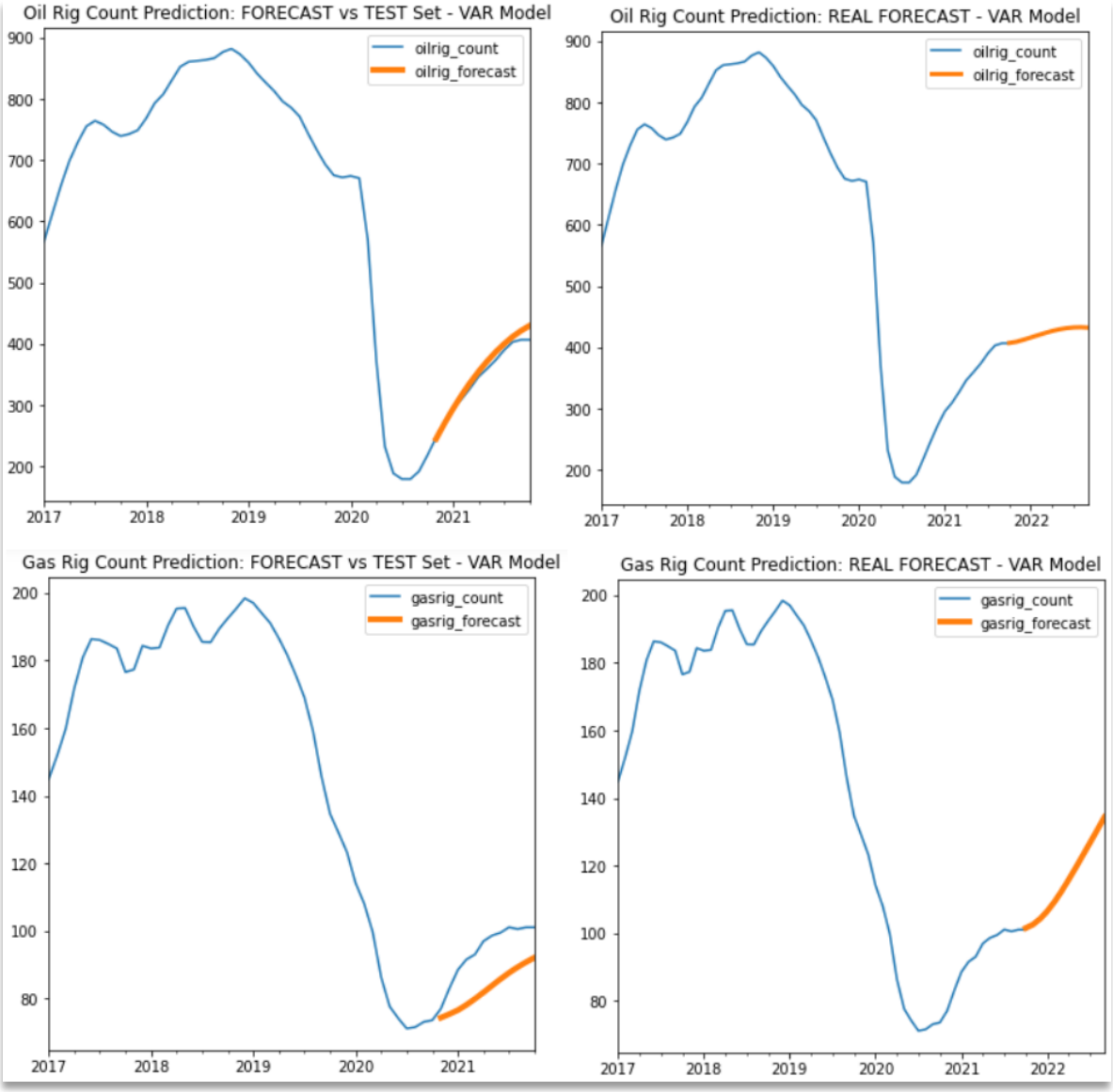

Figure 7 VAR model prediction plots in oil and gas data sets.

The second model used, Vector Auto Regression Moving Average (VARMA), is one of the statistical analyses used in several studies of multivariate time series data in economy, finance, and business. This model is an extension of VAR since it includes the Moving Average component. The Moving Average is a calculation to analyze data points by creating a series of averages of

different subsets of the full data set. A Moving Average is commonly used with the time series data to smooth out short-term fluctuations and highlight longer-term trends or cycles.

The hyperparameters needed to create the model are order, trend, and data assignment. The order is a tuple where we input the order for AR and MA components. We recall we used second-order differencing in our original dataset to obtain all features to be stationary. Therefore, we will use 2 for 'p' term (AR), and 2 for 'q' (MA). The trend by default is 'constant', but we have couple options available, constant trend, constant linear and quadratic term, and no constant.

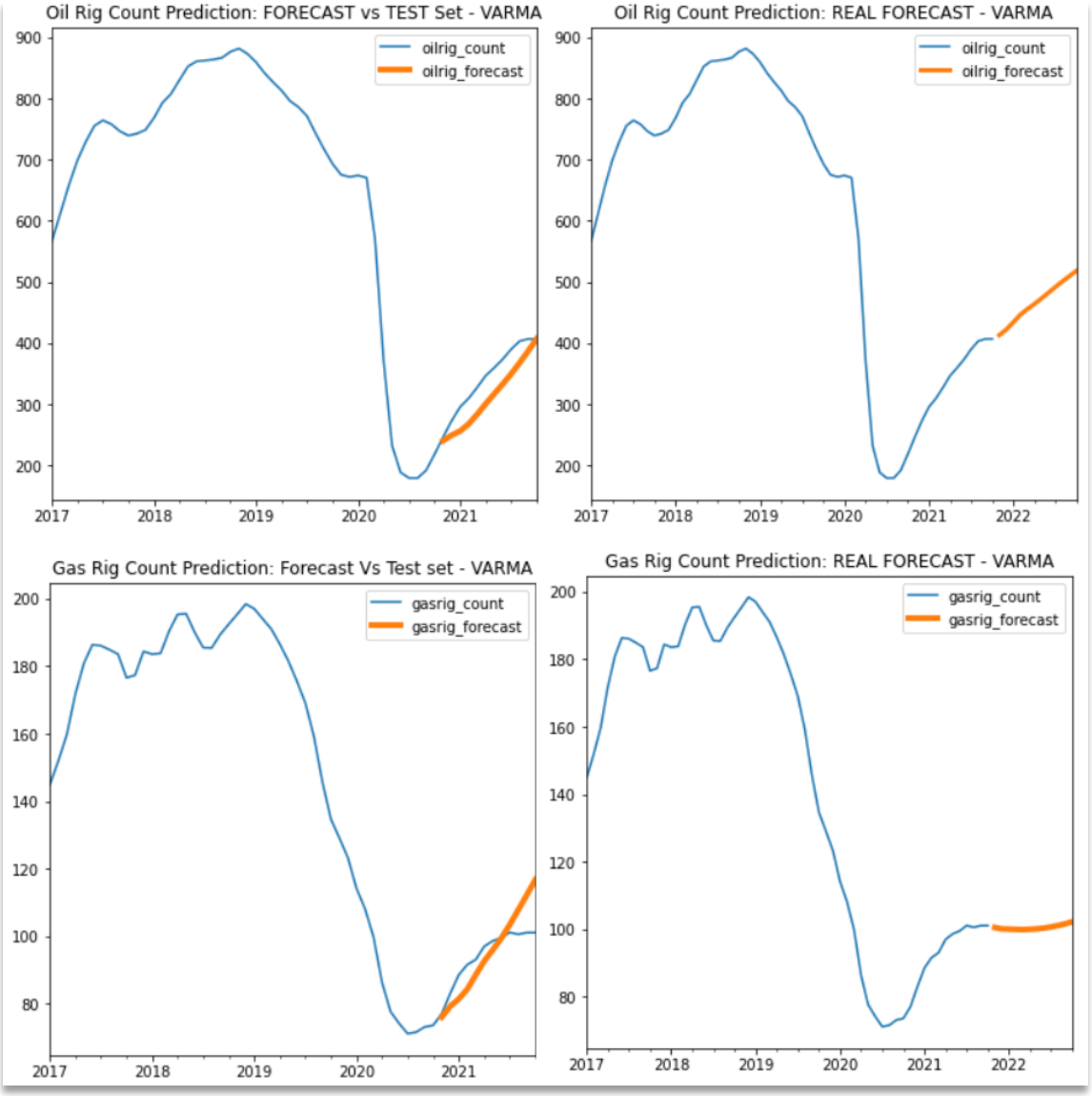

Figure 8 VARMA model prediction plots in oil and gas data sets.

Essentially, this parameter could be chosen based on trial an error and use the one that performs best with the data. In our case, we used 'ctt' (constant linear and quadratic term). We then instantiate the VARMA model, use transformed dataset and tune it with the hyperparameters. After that, we fit the model, but we pass in 'maxiter'=1000 and 'disp'=False. The 'disp' parameter stands for display and will just show several messages. From there, we create forecasts and pass in the number of datapoints to predict in the future, depending on time frequency (I recall we were using 12 months).

At the end, you will see forecasted values along with timestamp indices already built-in with the results. That is one difference VARMA has with respect to VAR when having forecast values. Finally, we invert the transformation back to its original form before plotting results and calculate the evaluation metrics.

The third model used, Auto Regressive Integrated Moving Average (ARIMA) is a statistical tool used for time series data for understanding and, perhaps, predicting values in the series. It has three components Auto Regression (also known as AR, or 'p'), Integrated (also known as I or 'd') and Moving Average (also known as MA or 'q'). The AR component involves regressing the variable on its own lagged (past values). The I component refers to the number of times the raw observations are differenced (transform data to stationary). The MA component is the size of the moving average window, also called the order or moving average. However, aside to VAR or VARMA, this model is for univariate time series data, which means only one variable will be used for forecasting. Therefore, this model will only rely on Oil Rig Count and Gas Rig Count data to make predictions.

When instantiating the model we pass in the order of p,d,q and the raw data. The better p,d,q orders could be found by running the 'auto_arima' function available in 'pmdarima' library. This function

will iterate with different orders with the ARIMA instance and determine the best order to be assigned based on the lowest AIC. After creating the ARIMA model, we make predictions by using the predict function and passing in the following hyperparameters; 'start' for start datetime, 'end' for end datetime, 'dynamic' set to False to disable dynamic start date assignment, and 'typ' set to levels to make sure we get the data in its original version (endogenous values), and not the differenced values. It is important to mention for ARIMA we do not need to transform the data since the Integrated component does that for us. As a result, we just have to make sure we get predictions in it's raw version for better interpretation of the results.

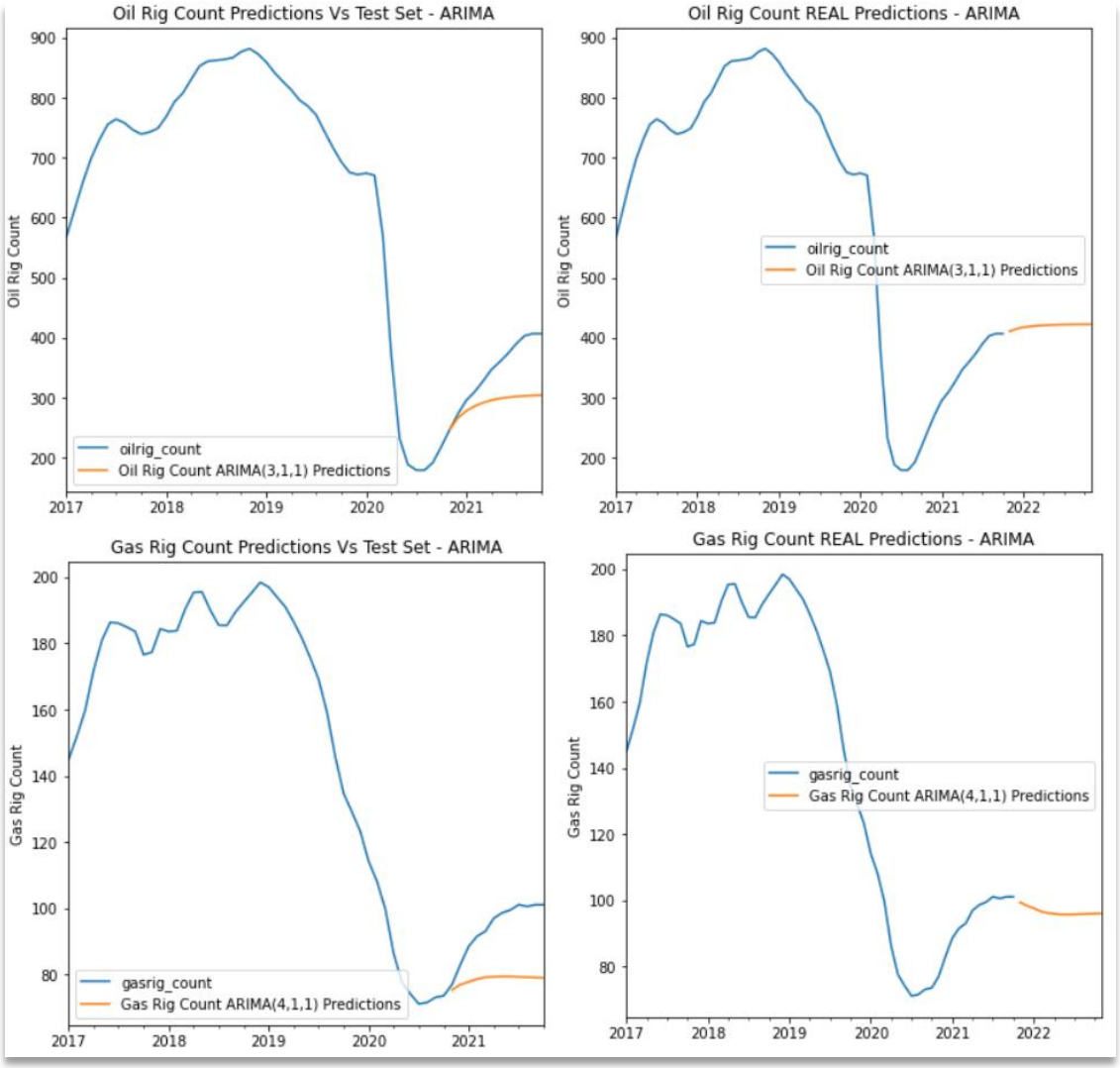

Figure 9 ARIMA model prediction plots in oil and gas data sets.

The last model used, Holt-Winters's method is a statistical tool used to model three aspects of univariate time-series data: level, trend, and seasonality (also known as alpha, beta, gamma). The Level represents values (average) of time series data, Trend is the tendency (increasing, decreasing), and seasonality refers to cyclical repeating patterns. Also, there are two variations of the Holt-Winters model: 'additive' and 'multiplicative'. The additive method is preferred when seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series.

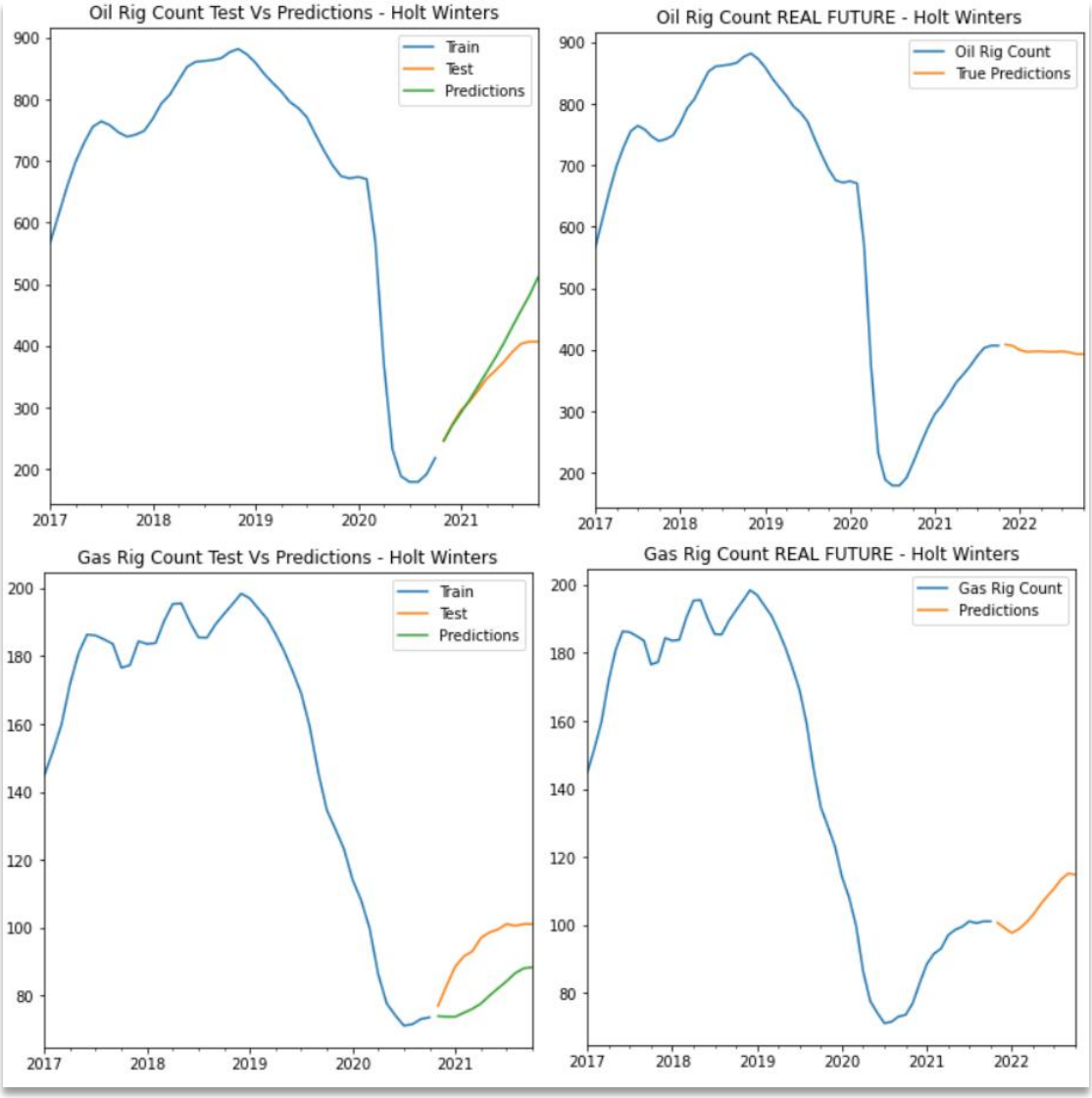

Figure 10 Holt-Winters model prediction plots in oil and gas data sets.

To create the model, we need to create the instance of the model just like the previous models, and pass in the following hyperparameters: dataset to choose, 'trend' set to additive, 'seasonal' set to multiplicative and 'seasonal_periods' to 12 (the months we will predict). After that, we create predictions with the predict method, and assign 12 since we want 12 months to predict. The Holt-Winters method is like ARIMA in terms it is for univariate data and does not consider the impact of other variables.

Evaluation Measures and interpretations

After completing all models and generated results, it's time to evaluate results based on Effectiveness, Efficiency and Stability. Talking about effectiveness, we will be select and applying the use of RMSE, MSE and MAE as a set of valid measures to test the performance of the models. Then, we will time the performance for every model as part of efficiency. Finally, we will do a 10-fold Cross Validation in our models to measure stability. At the end, we will compare them and interpret the results.

Talking about Effectiveness, we compare all forecasted values versus real values in three ways:

- Mean Absolute Error (MAE): The Absolute Difference between forecasted values and real values divided by the number of data points in the set. This is easy to understand since it tells you how far off forecasted values are from real values in average. However, this value won't alert us if the forecast was off for a few points. In other words, the MAE won't be able to detect specific offsets in the data since it averages all datapoints and might smoothen the error in some intervals.

- Mean Square Error (MSE): This is just the square of mean errors. Instead of having the absolute difference like MAE, we square this difference, so we always have a positive value. Also, by squaring the difference larger errors are noted more than MAE, making MSE more popular. However, since we squared the residuals, the units will be squared as well. As a result, sometimes it turns difficult to interpret.

- Root Mean Square Error (RMSE): We fix the problem of squared units in MSE by applying the square root, so the units are equal to its original form.

Overall, the lower the errors the better, but there is no magic number for this. It will depend on the model being evaluated to determine whether it performs good or not. For Example, the model that performed best in forecasting Oil Rig Counts is VAR model with a RMSE of 11.013 while Gas Rig Counts best model is VARMA with 7.045 RMSE.

To further explain the current project, those values mean the predictions in Oil Rig Counts would be off around 11 rigs (worst case scenario). This is a good indication since we are trying to predict values between 108 – 1596 rigs.

Model	RMSE	MSE	MAE
VAR Oil Rig Count **VAR Gas Rig Count**	11.013 (best) 11.890	121.297 141.381	9.293 11.385
VARMA Oil Rig Count **VARMA Gas Rig Count**	35.022 7.045 (best)	1226.545 49.629	31.681 5.611
ARIMA Oil Rig Count **ARIMA Gas Rig Count**	66.322 17.085	4398.668 291.883	55.252 15.785
HW Oil Rig Count **HW Gas Rig Count**	43.342 15.040	1878.522 226.195	29.241 14.389

Table 6 Evaluate Model: Effectiveness

Also, if we compare the error in predictions versus the standard deviation of the test set, we will have an idea of the normal fluctuations in the test set versus the error. On this case, the standard deviation of test set is 54.71 rigs versus 11.013 rigs RMSE. Therefore, we have lower variation than standard deviation of the set.

While continuing with the evaluation of Drilling Activity Prediction models, the efficiency is measured by obtaining the time the models take to fit, predict data. In that way, we measure the performance.

Model	Time in seconds
VAR Model	0.015 seconds
VARMA Model	147.76 seconds
ARIMA Model	0.25 seconds
Holt-Winters model	0.45 seconds

Table 7 Evaluate Model: Efficiency

The VAR model is the most efficient model when comparing to the rest of them, and the least efficient is VARMA. The reason could be related to the hyperparameter ‘maxiter’ in VARMA, which is set to 1000. If we lower this value, we could see more efficiency but probably at the cost of effectiveness.

Finally, when evaluating the models to guarantee stability, we use the 10-fold Cross Validation method, which basically consists of a resampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in settings where the goal

is prediction, and one wants to estimate how accurately a predictive model will perform in practice.

In time series, the train test split is done the following way:

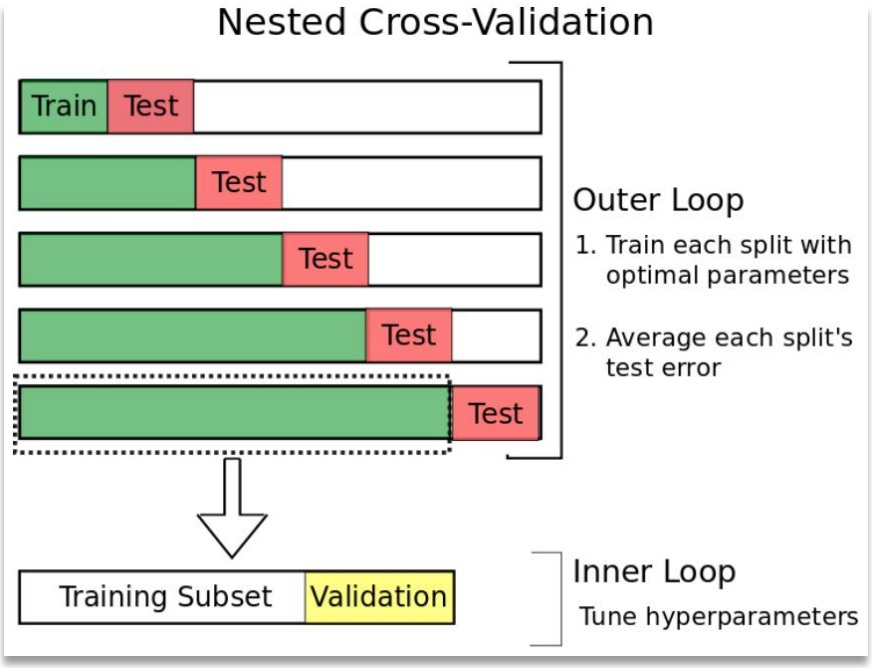

Figure 11 Cross Validation for Time Series (Example).

In brief, the train-test split is done in a specific order, and every time the train set is done it will have larger and larger data. At the end, we average all errors to observe results.

10-fold Cross Validation			
Model	RMSE	MSE	MAE
VAR Oil Rig Count **VAR Gas Rig Count**	515.152 190.62	382589.856 66785.434	488.906 187.414
VARMA Oil Rig Count **VARMA Gas Rig Count**	543.836 213.752	437414.838 87621.795	521,412 209.565
ARIMA Oil Rig Count **ARIMA Gas Rig Count**	195.867 64.389	77269.604 8689.194	170.945 57.037
HW Oil Rig Count **HW Gas Rig Count**	586.111 198.491	521892.011 86031.714	574.394 195.363

Table 8 Evaluate Model: Stability

In terms of stability, the models that best performed were ARIMA for Oil Rig Count and Holt-Winter for Gas Rig Count. This means that although they are not the more effective ones, they are more stable. However, all models perform poor in general in terms of Stability. The main reason might be related to the behavior of the model in the different scenarios of the rig count in different points of time (rig counts in ascending, descending or lateral trends). Also, there is no clear seasonality in the curve, turning it more challenging for the models to find patterns. Finally, more model hyperparameter tuning and further feature engineering could be applied to improve overall evaluation metrics.

Conclusions

Drilling Activity Predictions consist of forecasting the oil and gas rig counts in the United States. These rigs oversee the exploration and development of wells and is a good indicator of overall drilling activity and industry status. However, this rig count has fluctuated dramatically over the last years because of several factors including economic, political, social, etc. As a result, oil and gas companies need to quickly react, provide resources and always deliver quality services, which end up turning into a difficult task. Drilling Activity Predictor tries to smoothen the fluctuations in oil and gas activity by forecasting future values based on economic factors like oil-gas prices, imports, exports, storage, production, etc. and past data (lagged values) using forecasting models like VAR, VARMA, ARIMA or Holt-Winters.

Overall, after interpreting the results of the models, we could say there is no best or worst model. There are certain conditions where one model performs best than the other. For example, VAR model performed best in Oil dataset, but VARMA performed best in Gas Dataset. That means that

the data has an impact in the overall effectiveness of the predictive models. Moreover, all the models have certain error, which means they are not perfect. These forecasting models just try to predict based on trends, seasonality, patterns and averaging the data.

Also, in time-series, the frequency plays an important role. If the data is resampled to daily frequency, it could lead to totally different results. Therefore, it is important to select the most convenient frequency based on final row count and values. For instance, if daily data fluctuations are too high, a weekly or monthly data might smoothen the time-series and provide more accurate results. Otherwise, too much noise could lead to inaccurate results. Finally, it was noticed the forecasting models behave different in various points of time. This means, we obtain unequal results in scenarios where time series data shows an ascending trend, descending or lateral. This could lead to lower overall stability of the models. However, this could be mitigated by the following (see table 9 below):

Way Forward / Actions to take to improve the model
Missing Data: Apply 'backcasting' to raw data to fill past missing data in the features. More data lead to better predictions.
Find more features: Supply and demand from OPEC, Non-OPEC organizations, pandemic flags, changes in politics, etc. that might affect oil and gas industry.
Use Rolling Averages: Use Simple Moving Averages to fit models instead of raw data.
Use Deep Learning with early-stopping parameter to avoid overfitting.
Continue looking for better combinations of hyperparameters to improve models.
Resample data to weekly frequency and refit models with this. Observe results.

Table 9 Concluding remarks on the continuity of the work.

References

Brenner, M. (2008, April). *Rig Count.* Retrieved from Department of Natural Resources - State of Louisiana: http://www.dnr.louisiana.gov/assets/docs/energy/reports/Rig_Counts_Report_2008.pdf

Cochrane, C. (2018, May 18). *Time Series Nested Cross Validation*. Retrieved from Towards Data Science: https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9

Perry, K. (2019, July). *Electricity Price Time Series Analysis*. Retrieved from github.com: https://github.com/kperry2215/electricity_price_time_series_analysis

Seeking Alpha. (2015, August 17). *Crude Oil: The Divergence of Rig Count, Oil Price and Production*. Retrieved from seekingalpha.com: https://seekingalpha.com/article/3444326-crude-oil-divergence-of-rig-count-oil-price-and-production-explained?gclid=CjwKCAjwk6-LBhBZEiwAOUUDp3saTgModVrPgxdhXvo2rdQpg5ZR2xZSAE2-1dtDGVyUEBd_EX7qVxoCeysQAvD_BwE&utm_campaign=14049528666&utm_medium=cpc&utm_sou

Taylor, A. (2019, June 18). *Deep Learning for Time Series Forecasting*. Retrieved from github.com: https://github.com/Azure/DeepLearningForTimeSeriesForecasting/blob/master/1_CNN_dilated.ipynb

U.S Department of Energy. (2020, May 14). *EIA forecasts U.S. crude oil production to fall in 2020 and 2021*. Retrieved from eia.goc: https://www.eia.gov/todayinenergy/detail.php?id=43735

U.S Department of Energy. (2021, September 02). *Short-Term Energy Outlook.* Retrieved from Energy Information Administration (EIA): https://www.eia.gov/outlooks/steo/pdf/steo_text.pdf

U.S. Deparment of Energy. (2021, September 01). *Open Data API Portal*. Retrieved from Energy Information Administration (EIA): https://www.eia.gov/opendata/

U.S. Department of Energy. (2021, April 01). *U.S. Energy Facts*. Retrieved from Energy Information Administration (EIA): https://www.eia.gov/energyexplained/us-energy-facts/

U.S. Department of Energy. (2021, September). *What Drives Crude Oil Prices?* Retrieved from eia.gov: https://www.eia.gov/finance/markets/crudeoil/spot_prices.php

Woods, T. (1988, October). *Economic Factors Controlling Drilling Activity.* doi:https://doi.org/10.2118/18105-MS