MEF UNIVERSITY

PREDICTION OF BRENT OIL SPOT PRICES USING COUNTRY BASED INVENTORY AND TRADING DATA

Capstone Project

İsmail Batur Usta

İSTANBUL, 2019

MEF UNIVERSITY

PREDICTION OF BRENT OIL SPOT PRICES USING COUNTRY BASED INVENTORY AND TRADING DATA

Capstone Project

İsmail Batur Usta

Advisor: Prof. Semra Ağralı

İSTANBUL, 2019

MEF UNIVERSITY

Name of the project: Prediction of Brent Oil Spot Prices Using Country Based Inventory and Trading Data
Name/Last Name of the Student: İsmail Batur Usta
Date of Thesis Defense: 04/09/2019

I hereby state that the graduation project prepared by İsmail Batur Usta has been completed under my supervision. I accept this work as a “Graduation Project”.

04/09/2019
Asst. Prof. Tuna Çakar

I hereby state that I have examined this graduation project by İsmail Batur Usta which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

04/09/2019
Prof. Özgür Özlük

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of İsmail Batur Usta and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Asst. Prof. Tuna Çakar	………………………..
2. Prof. Özgür Özlük	………………………..

Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

İsmail Batur Usta 04/09/2019

EXECUTIVE SUMMARY

PREDICTION OF BRENT OIL SPOT PRICES USING COUNTRY BASED INVENTORY AND TRADING DATA

İsmail Batur Usta

Advisor: Prof. Semra Ağralı

AUGUST, 2019, 25 Pages

Crude oil price forecasting has been the focus of numerous authorities, yet the task still persists on being a challenging one. The extremely volatile nature of oil market and high number of active players in it makes establishing a solid forecasting model that is constantly relevant to time very difficult. Recent advancements on data technologies, mainly ever-increasing computing power and trending big data technologies allowed new approaches to be born. From online learners to natural language processing, advanced data analytics models were employed with the help of easily accessible and diverse data. This project is an attempt on making use of such available data in order to forecast Brent oil spot price. By using monthly country by country inventory, trading and economic data, strong drivers of crude price was explored. The data used in this project comes from various sources and in multiple formats, with the final merged data frame has over 17000 observations and contains information on 86 countries. To enhance prediction power, a specialized learner is fit on each country individually and then the predictions are accumulated and filtered before outputting a single prediction. Compared to a single predictor, this approach enhanced the predictive power of the algorithm by adapting to dynamics of each country.

Key Words: Crude oil, Brent Crude, forecasting, support vector machines, gradient boosting, decision tree, inventory, import, export

ÖZET

ÜLKE BAZLI ENVANTER VE TİCARİ VERİLERİ KULLANARAK BRENT HAM PETROLÜ SPOT FİYATLARI TAHMİNİ

İsmail Batur Usta

Tez Danışmanı: Prof. Semra Ağralı

AĞUSTOS, 2019, 25 Sayfa

Ham petrol fiyat tahmini birçok çalışmanın odak noktası olmuş olmasına rağmen zorlu bir iş olma özelliğini sürdürmektedir. Ham petrol marketinin istikrarsız doğası ve çok sayıda oyuncuya sahip olması sağlam temellere dayanan ve zaman ile geçerliliğini yitirmeyen bir tahmin edicinin yaratılmasını hayli zor kılmaktadır. Veri teknolojilerindeki yeni gelişmeler, özellikle sürekli artan işlem gücü ve büyük very teknolojilerinin gündemde önemli yer tutmaya başlaması, yeni yaklaşımların doğmasına ortam sağlamıştır. Kolay ulaşılabilir ve çeşitli veri ile çevrimiçi öğrenicilerden doğal dil işlemeye, ileri veri analitiği modelleri uygulanmaya başlanmıştır. Bu çalışma da bu şekilde mevcut veriyi kullanarak Brent ham petrolü spot fiyatını tahmin etmeye çalışmak için yapılmıştır. Ülke bazlı envanter, ticaret ve ekonomik veri kullanılarak ham petrol fiyatının sürücü güçleri tespit edilmeye çalışılmştır. Bu çalışmada kullanılan veri birçok kaynaktan ve farklı formatlarda gelmektedir. Veri tablosunun işlemeye hazır hali 17000'in üzerinde gözlem ve 86 farklı ülkeye ait veriye sahiptir. Modelin tahmin etme gücünü arttırmak için her ülkeye özgü şekilde oluşturulmuş modeler yaratılarak tahminleri bir araya getirilip filtrelendikten sonra ana tahmin ortaya çıkarılmıştır. Her ülkenin dinamiklerine adapte olmayı başararak, bu model tekil bir tahmin ediciye kıyasla daha iyi bir tahmin etme gücüne sahiptir.

Anahtar Kelimeler: Ham petrol, Brent ham petrol, fiyat tahmini, destek vektör makinesi, gradyan artırma, karar ağacı, envanter, ithalat, ihracat

TABLE OF CONTENTS

1. INTRODUCTION

Crude oil is a leading commodity that effectively drives global economy. While the range of products and sub products that are derived from crude oil are extremely wide, the majority of crude oil products consist of fuels, which are produced by the refining industry. In the current setting, leading players of crude oil trade are usually among the largest producers of crude oil, some of which are the USA, Russia and the countries of Arabian Peninsula. Crude oil is a substance that naturally occurs over very long time periods and the characteristics of the crude can vary heavily depending on the location, the method of production and even the age of the well, meaning that the same crude may have different qualities throughout years. As a result, there are numerous commercial crudes with different names and prices on the market at any moment. Rather than setting the price of each individual crude separately, they are usually determined relatively to the benchmark crudes, which are Brent Crude for Europe, Asia and Africa regions and WTI for North America. According to Scheitrum et al. (2018), Brent recently overtook WTI as the crude with the highest trading volume [1]. This project mainly focuses on the factors that influence Brent oil prices.

Miao et al. (2017) state that crude oil prices depend on already complex and multi-layered factors, which are difficult to guess in the first place [2]. It is also not helpful that unexpected events may occur rather frequently and violently to the extent of disrupting oil trade globally or in a region, such as political crises, production disruptions or terrorism. This, in conclusion, introduces a risk factor that producers and traders have to factor in to keep their businesses running. The commodity market is the main driving force that determines the crude oil prices. Actions of traders in the commodity market, their futures contacts bids, and their hedging strategies are among the direct reasons that affect the crude price for a particular date. All other factors that have been explained above affect traders' decisions. Amadeo (2019) splits the factors that affect the decision processes of traders in three categories; current global crude supply, situation on oil reserves and global oil demand [3]. For example, if there is a political tension between a major oil producer and a major oil consumer that can affect trade or implies sanctions in the future, traders will respond in order to cover potential losses and the prices will shift. How drastic the shift will be is obscured at best and that is why forecasting crude oil price is a tedious task.

2. LITERATURE REVIEW

Crude oil price forecasting has been the subject of many scholars and companies. The nature of the models that have been developed to forecast oil prices can be categorized in various ways. Li et al. (2016), for example, divide these works into two fields: AI models, those that incorporate technologies such as machine learning and deep learning; and statistical and econometric models, which are arguably simpler but much more easily explainable and justifiable than AI solutions [4]. On the other hand, Natarajan and Ashok (2018) make this classification not by the methodology but by the structure of the data used. They state that there are three distinct methods, time series analysis, which uses the historical data of the prices and predicts the future, econometric models, which factor in different economic factors, and multivariate models that use various variables that can come from any source or discipline [5]. The most encountered methodology consists of different applications of time series analysis. Time series analysis, such as ARIMA, generally performs very well on forecasting near time results, and that is enough for most companies. The simplicity of the model is also an advantage, and such models can usually be created using a spreadsheet application. On the other hand there is no shortage of more advanced and fundamentally different solutions in the literature.

Models that only use historical prices usually apply ARIMA like models as a benchmark. Xie et al. (2006) perform such a comparison with their Support Vector Machine (SVM) model, and report that their SVM model that uses radial basis function kernel outperforms ARIMA in accurately predicting future prices and even more so in accurately predicting the direction of future prices [6]. Li et al. (2018) stated that due to its volatile nature, feeding raw oil price data into a model would hurt the performance of the model and applied a “decomposition and ensemble” methodology, combining ensemble empirical mode decomposition, sparse Bayesian learning and simple addition into a forecasting algorithm to boost prediction performance [7]. Mahdiani and Khamehchi (2016) use a modified neural networks model, which is modified with a genetic algorithm, and claim that their method, even when using limited data, results in better fits than general artificial neural networks [8].

Many other approaches are used in order to establish a better model. For example, Li et al. (2018) offer a convolutional neural network model that takes text data retrieved from a popular financial website’s news and headlines [9]. Gao and Lei (2017) use a stream learning method that re-trains the model every time a new price data set is available, claiming that this would help forecast the fluctuations in the market better [10]. Ye et al. (2005) claim that adding inventory data into a simple dynamic model results in a commercially usable algorithm [11]. Natarjan and Ashok (2018) use a more elaborate approach that incorporates supply and demand data of different markets into a neural network model in order to predict future prices [5].

3. DATA ANALYSIS

This project uses data from different sources with different formatting conventions. Therefore, it is necessary to transform and filter the data into a unified structure before proceeding with analytics.

3.1 JODI Oil Data

Majority of the data is acquired from Joint Organizations Data Initiative (JODI, https://www.jodidata.org/oil/). This initiative is dedicated to provide transparency to the energy market to grant stability and reduce volatility. While this project only uses the crude oil data of the database, it also contains information about downstream products as well, such as LPG, Fuel Oil, and Gasoline. Table 1 shows the initial structure of the data. The elements of the columns are as following [12]:

REF_AREA:

This column contains the abbreviations of the countries. Refer to APPENDIX A for country codes.

TIME_PERIOD:

In Year-Month format, starting from January 2002.

ENERGY_PRODUCT:

Specifies which type of product the data belongs to:

CRUDEOIL: Crude Oil

NGL: Natural Gas

OTHERCRUDE: Other Oil Products

TOTCRUDE: Total

FLOW_BREAKDOWN:

Specifies what the numerical data represents in terms of source of the flow.

INPROD: Production

OSOURSCES: From other sources

TOTIMPSB: Imports

TOTEXPSB: Exports

TRANSBAK: Products transferred/Backflows

DIRECUSE: Direct Use

STOCKCH: Stock change

REFINOBS: Refinery intake

STATDIFF: Statistical difference

CLOSTLV: Closing Stocks

UNIT_MEASURE:

Measurement units used in the numerical data.

KBD: Thousand barrels per day

KBBL: Thousand barrels

KL: Thousand kilolitres

KTONS: Thousand metric tons

CONVBBL: Conversion factor barrels/ktons

ASSESSMENT_CODE:

Results of the assessment of the data.

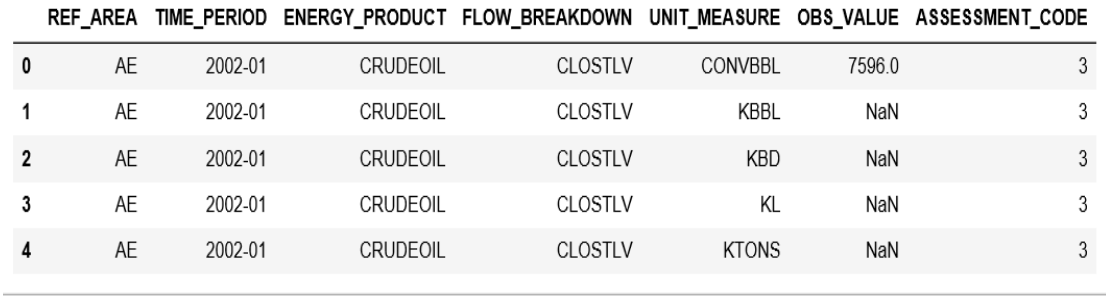
1: Reasonable levels of comparability

2: Consult metadata

3: Data has not been assessed

4: Data under verification

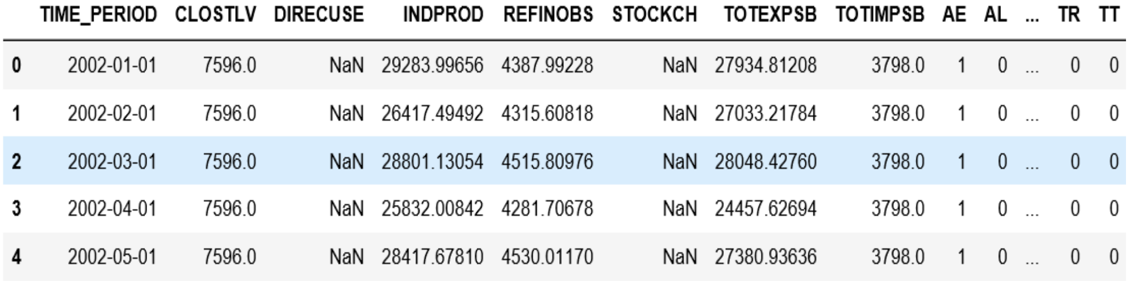
Table 1. JODI oil database original structure



	REF_AREA	TIME_PERIOD	ENERGY_PRODUCT	FLOW_BREAKDOWN	UNIT_MEASURE	OBS_VALUE	ASSESSMENT_CODE
0	AE	2002-01	CRUDEOIL	CLOSTLV	CONVBBL	7596.0	3
1	AE	2002-01	CRUDEOIL	CLOSTLV	KBBL	NaN	3
2	AE	2002-01	CRUDEOIL	CLOSTLV	KBD	NaN	3
3	AE	2002-01	CRUDEOIL	CLOSTLV	KL	NaN	3
4	AE	2002-01	CRUDEOIL	CLOSTLV	KTONS	NaN	3

The original dataset contains more than 4.5M data points but since each unit of measurement has a different row, some of the data represent the same value. Therefore, converting all measurements into the same type of unit would reduce this number. Unit of measurement usually changes depending on the reporting country and there is only one value, which means the column contains a high number of null values. In order to infer a meaning from the database structure, some transformations are required. With the aim of distinguishing country and flow type data, the structure of the database is changed in a way that every flow type has its own column, and country codes are one hot encoded. Table 2 shows the new format of the database.

Table 2. Transformed format of the JODI database



	TIME_PERIOD	CLOSTLV	DIRECUSE	INDPROD	REFINOBS	STOCKCH	TOTEXPSB	TOTIMPSB	AE	AL	...	TR	TT
0	2002-01-01	7596.0	NaN	29283.99656	4387.99228	NaN	27934.81208	3798.0	1	0	...	0	0
1	2002-02-01	7596.0	NaN	26417.49492	4315.60818	NaN	27033.21784	3798.0	1	0	...	0	0
2	2002-03-01	7596.0	NaN	28801.13054	4515.80976	NaN	28048.42760	3798.0	1	0	...	0	0
3	2002-04-01	7596.0	NaN	25832.00842	4281.70678	NaN	24457.62694	3798.0	1	0	...	0	0
4	2002-05-01	7596.0	NaN	28417.67810	4530.01170	NaN	27380.93636	3798.0	1	0	...	0	0

This format will allow us to easily perform exploratory analysis on the data. As an example, Turkey's data is presented in Figure 1.

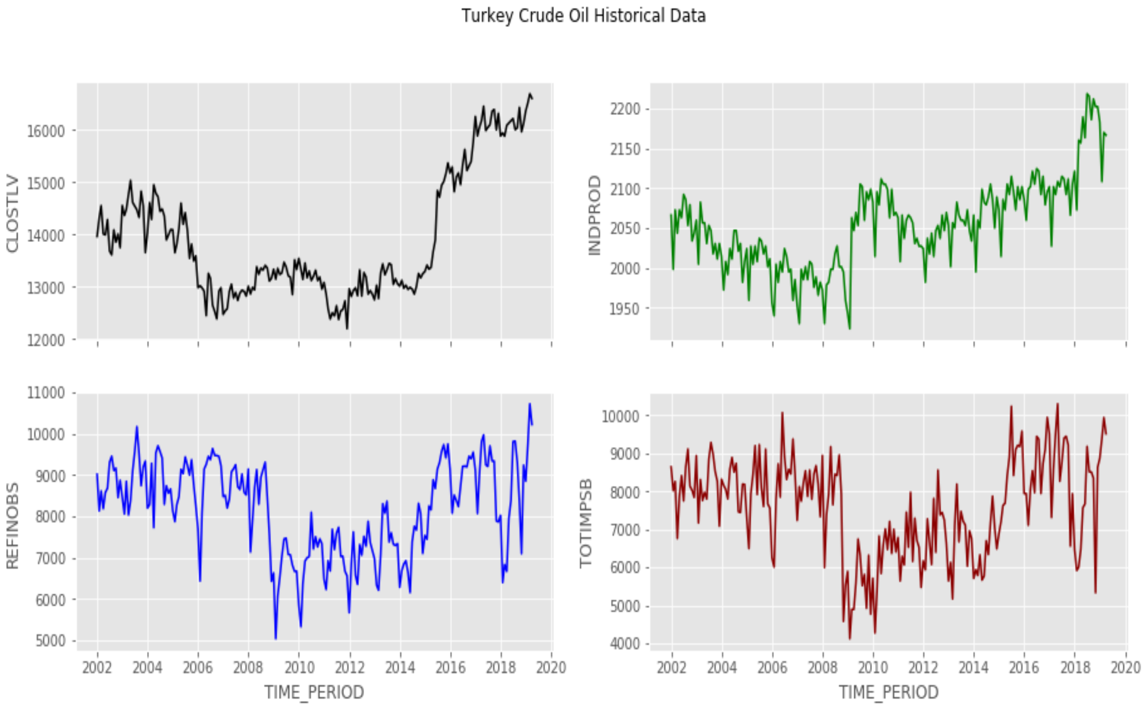


Figure 1. Turkey's crude oil information

From the graphs we can see that Turkey's crude oil production has increased in the later years to the levels of 2.2 million metric tons per month, while the refinery intake is between 7 million tons and over 10 million tons lately. This shows that Turkey needs to import large portion of its crude supply, which is proven by the net import graph. In Figure 2, we show how the flow type data are correlated overall.

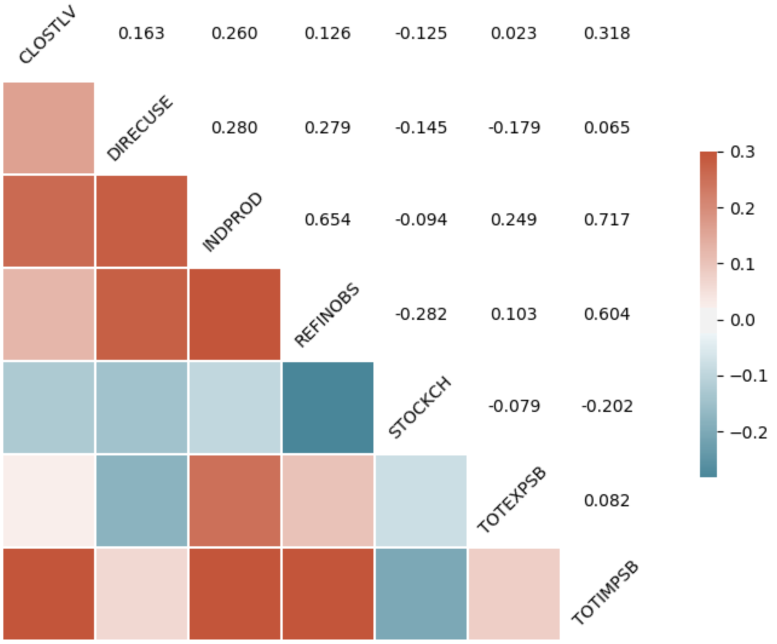
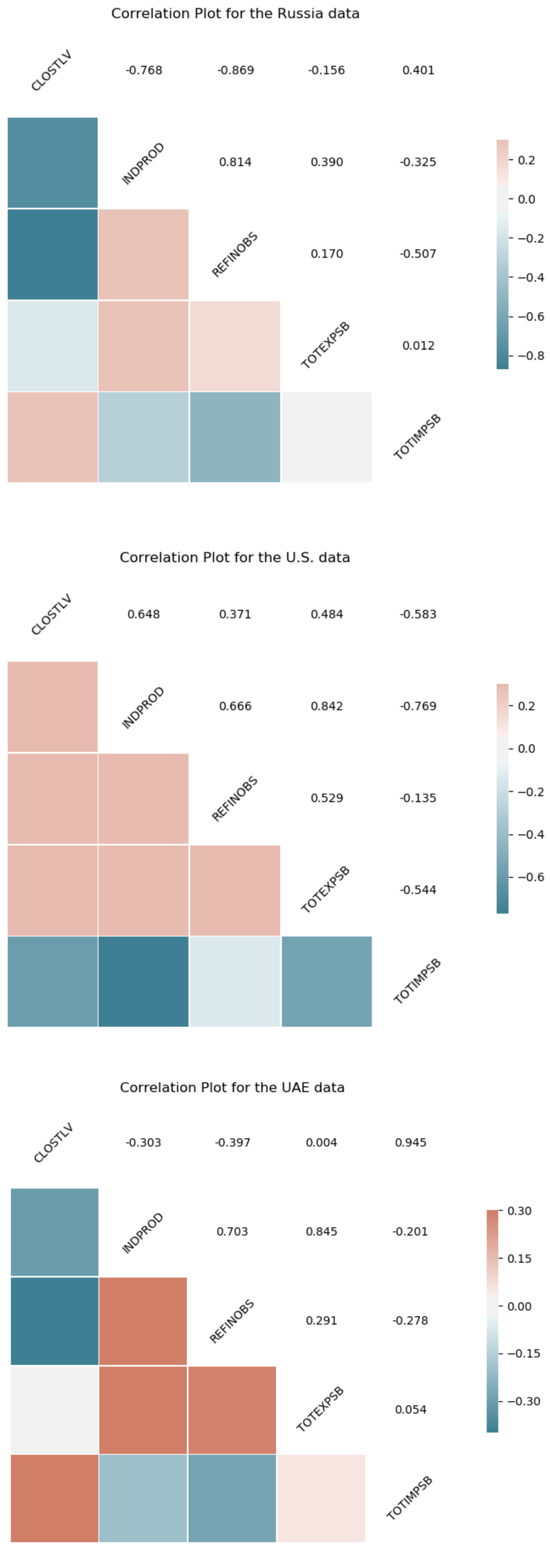


Figure 2. Correlation plot for overall monthly data

This data suggests that refinery intake is closely related to the country's production and crude import. It means that when global crude production capacities are high, we can expect that the crude oil market will be active. It might be helpful to look at the data on a country basis to see how this outlook diverges when we close down on a country. Figure 3 supports this idea, as Russia, UAE and the U.S data show different correlations on their own.



Correlation Plot for the Russia data

Correlation Plot for the U.S. data

Correlation Plot for the UAE data

Figure 3. Correlation plots for Russia, USA and UAE

In United Arab Emirates (UAE), the crude production is highly correlated with total exported crude, while closing stocks and total imports are related. Some of the correlations are similar; however, for example refinery intake is highly and positively correlated with crude production. A closer interaction with these correlations would help us determine plenty of information about crude oil market such as which countries are net importers and which are net exporters over a given time.

3.2 U.S. Dollar Index

The U.S. dollar index represents where U.S. dollar stands in terms of value compared to a selected number of currencies with which the U.S. dollar have close interactions [13]. These currencies currently are Euro, Japanese yen, Canadian Dollar, Swedish Krona, Swiss Franc and British Pound [13]. The base index is 100, which is set when the index is first established in 1973, and its current value shows the power of the U.S. dollar compared to that year [13]. Including the dollar index in this project helps identify points where the crude price shift is due to an indirect effect on the U.S. dollar and not something originated from crude oil market itself. Figure 4 shows the change of the U.S. dollar index since 1973. The index saw a drop from 2002, and then started to climb back after second half of 2010.

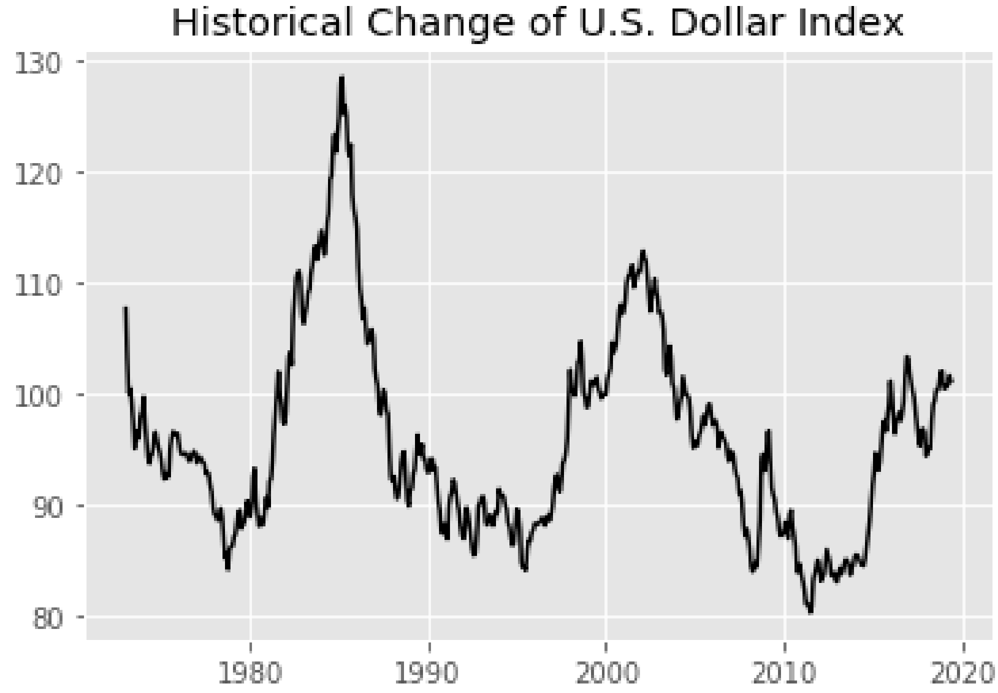


Figure 4. Historical change of the dollar index

3.3 Brent Crude Spot Prices

The Brent crude spot price data is retrieved from Energy Information Administration (EIA) and has a custom granulation. Figure 5 shows the Brent spot prices over the years. The oil prices saw a gradual rise after year 2000 and it is obvious that after year 2008, the already volatile crude oil market have become extremely unpredictable and a high risk zone. Those events in 2009 and after 2014 caused great calamity on the market and actually caused some major oil companies go bankrupt. This trend supports the idea that correctly predicting future prices is very crucial for major companies.

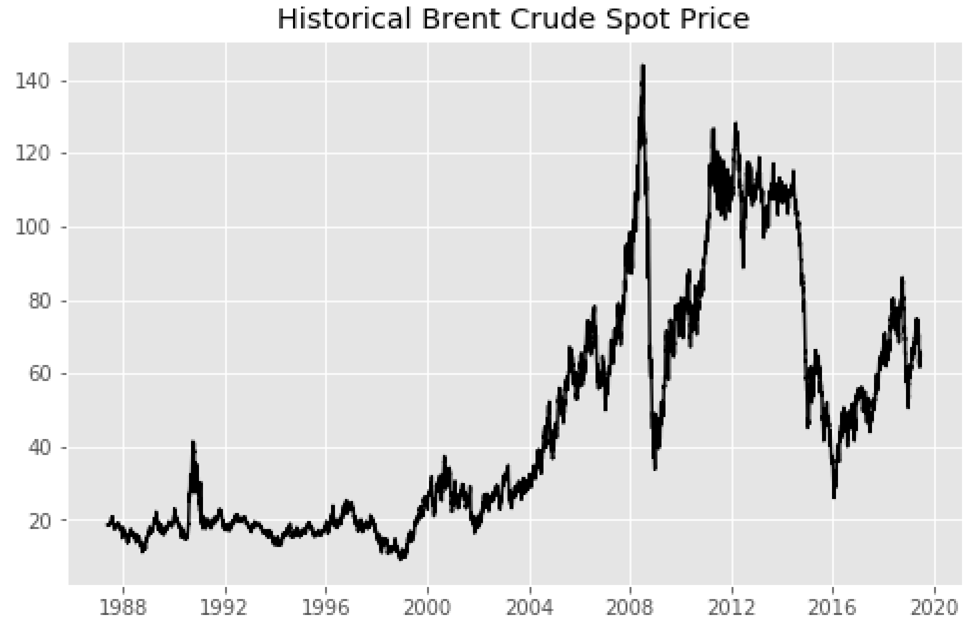


Figure 5. Brent crude spot price history

4. PROJECT DEFINITION

4.1. Objective and Problem Statement

The objective of this Project is to establish a stable machine learning model that predicts the Brent Crude Oil spot prices using commercially available and relevant inventory, trading and econometric data.

The models that are being widely used in the industry are mostly based on linear correlations and reportedly cannot perform well when it comes to non-linearity of the crude prices. However, while addressing that issue with a nonlinear model, one must keep in mind that it also would not be feasible to create a very complex model with high processing power demands, or a model that has to be constantly monitored and updated. Therefore, a middle ground has to be found.

This project is focused on establishing a correlation between crude oil spot prices and information from the crude market with the ultimate goal of utilizing machine learning algorithms to create a reliable price-forecasting model. The data used is crude oil inventories by country, oil trade market data, U.S. dollar index, and historical Brent spot prices. Given this data, an ensemble regression algorithm is created and evaluated with out of sample data.

4.2 Project Scope

This project will not weigh on time series analysis; instead, it will focus on direct correlations between relative data and the prices. The data from JODI database goes as far as 2002; therefore, only the data from present time to 2002 will be used in other data sources as well.

The project includes data from all countries in order not to lose any precious information. The U.S. price index is also included to weed out price fluctuations that might have caused by the currency. Eventually, U.S. price index is proven to be a strong drive on crude prices and adding it increases the model performance drastically. Inventory data from each country is used in reducing the error rate.

5. METHODOLOGY

The main approach that is used for the problem studied is a regression algorithm. During the initial exploration of the JODI database, a benchmark algorithm is performed to establish how well the dataset reacts to a regression solution. In this experiment, 5 attributes, namely CLOSTLV, INDPROD, TOTEXPSB, TOTIMPSB and REFINOBS are used to perform a plain principal component analysis (PCA). We find that two principal components explain over 94% of the variance in the data. These principal components are then fed into several regression algorithms, where the Random Forest Regressor perform the best. Figure 5 shows the initial results of the benchmark model. It must be noted that this model is not very reliable because it suffers from overfitting, but it will only be used as a starting point for further analytics.

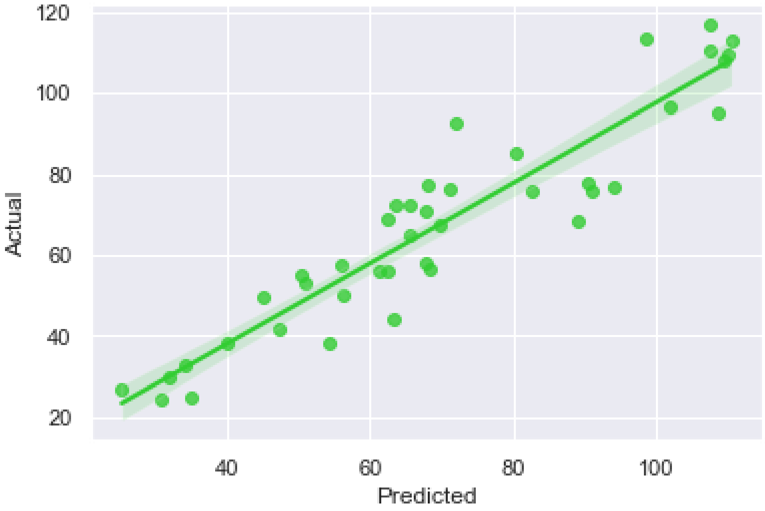


Figure 5. Actual vs Predicted for benchmark model

The results of the benchmark model show that the inventory data is somehow reliable to predict the crude spot prices, or at least that it will help increase the performance. The main challenge while preparing the real algorithm is to preprocess the data frame correctly. It is established before that the original data frame is somewhat chaotic and difficult for a machine learning algorithm to process. In order to address that issue, several transformations are made.

5.1 Data Preprocessing

Due to each country or region having different measurement units and metrics in their reports, the JODI database contains several units of measurements in the data frame for the same element. In some instances measurements for all units are present, but in majority only two or three of them are reported. In order to both verify accuracy and have all measurements in a single unit, a simple function is written. Please refer to APPENDIX B-Script 1 for the python code that is used. The script simply converts all measurements into thousand metric tons, which is the chosen measurement for this project.

Thousand barrels per day into thousand barrels:

$$KBBL = KBDx24$$

Thousand kiloliters into thousand barrels:

$$KBBL = KLx6.2898$$

And then thousand barrels are converted into thousand metric tons by using the conversion factor:

$$KTONS = \frac{KBBL}{CONVBBLx1000}$$

As a result, all measurements are now in thousand metric tons and an information loss is prevented.

Another transformation is made on date data. Since the information fed into the model is on a monthly basis, year and month of the data are separated into two columns. This introduces an additional issue where the month column simply contains numbers from 1 to 12 and in this state the model would not be able to relate January with December. Kaleko (2017) offers a solution to this issue by converting month data into cyclic features by calculating sine and cosine of the month and replacing month column with two columns, representing each component [14].

$$Month\ Sine = \sin\left((Month - 1)\ x\ \left(2\ x\frac{\pi}{12}\right)\right)$$

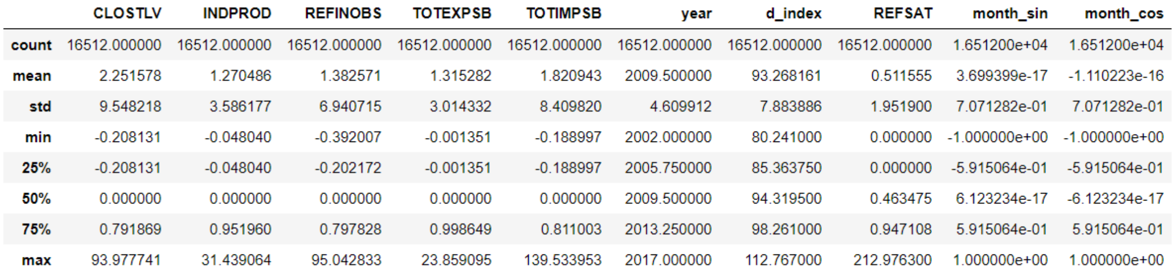
$$Month\ Cosine = \cos\left((Month - 1)\ x\ \left(2\ x\frac{\pi}{12}\right)\right)$$

A new feature is also added to the data frame called REFSAT, which represents how much of country's crude import is to saturate their refinery intake. It is an indicator of country's static crude demand.

$$REFSAT = TOTIMPSB/REFINOBS$$

The dataset originally contains 121 countries. However, 39 of those countries found to have very little to no information on inventory and trade data. Therefore, these countries and rows associated with them are dropped from the main data frame in order to prevent them from reducing the accuracy of the model. As a result, the final database on which the machine learning models are trained contains 86 countries. Finally, columns CLOSTLV, INDPROD, REFINOBS, TOTEXPSB, TOTIMPSB and REFSAT are scaled using Robust Scaler of sklearn [15]. Table 3 shows the descriptive information on the data frame that is used on training the models.

Table 3. Descriptive information of the final dataframe



	CLOSTLV	INDPROD	REFINOBS	TOTEXPSB	TOTIMPSB	year	d_index	REFSAT	month_sin	month_cos
count	16512.000000	16512.000000	16512.000000	16512.000000	16512.000000	16512.000000	16512.000000	16512.000000	1.651200e+04	1.651200e+04
mean	2.251578	1.270486	1.382571	1.315282	1.820943	2009.500000	93.268161	0.511555	3.699399e-17	-1.110223e-16
std	9.548218	3.586177	6.940715	3.014332	8.409820	4.609912	7.883886	1.951900	7.071282e-01	7.071282e-01
min	-0.208131	-0.048040	-0.392007	-0.001351	-0.188997	2002.000000	80.241000	0.000000	-1.000000e+00	-1.000000e+00
25%	-0.208131	-0.048040	-0.202172	-0.001351	-0.188997	2005.750000	85.363750	0.000000	-5.915064e-01	-5.915064e-01
50%	0.000000	0.000000	0.000000	0.000000	0.000000	2009.500000	94.319500	0.463475	6.123234e-17	-6.123234e-17
75%	0.791869	0.951960	0.797828	0.998649	0.811003	2013.250000	98.261000	0.947108	5.915064e-01	5.915064e-01
max	93.977741	31.439064	95.042833	23.859095	139.533953	2017.000000	112.767000	212.976300	1.000000e+00	1.000000e+00

5.2 Model Training and Parameter Tuning

Figure 3 gives the correlation plots for different countries. These plots support the idea that each country has distinctive relations between their crude oil related information and for the most part, these correlations stay relevant throughout extended periods of time. We find that instead of feeding all the data into an overarching model and returning a single prediction, establishing an ensemble model that contain weak learners trained on the dynamics and information of each country is a better choice. This structure can also be enhanced with a best model selection algorithm. One downside to this would be the increased workload and runtime of training model. As a result, a class is written, which feeds country data into several regression algorithms and chooses the best one. Training is done by performing grid search and 10 k-fold cross validation. The model then stores mean absolute errors and cross validation scores, and selects the best algorithm to run on that particular country. The model is then saved on a local computer with the joblib library [15]. Figure 6 represents the framework of the algorithm. Please refer to APPENDIX B-Script 3 for python code that is written for this task.

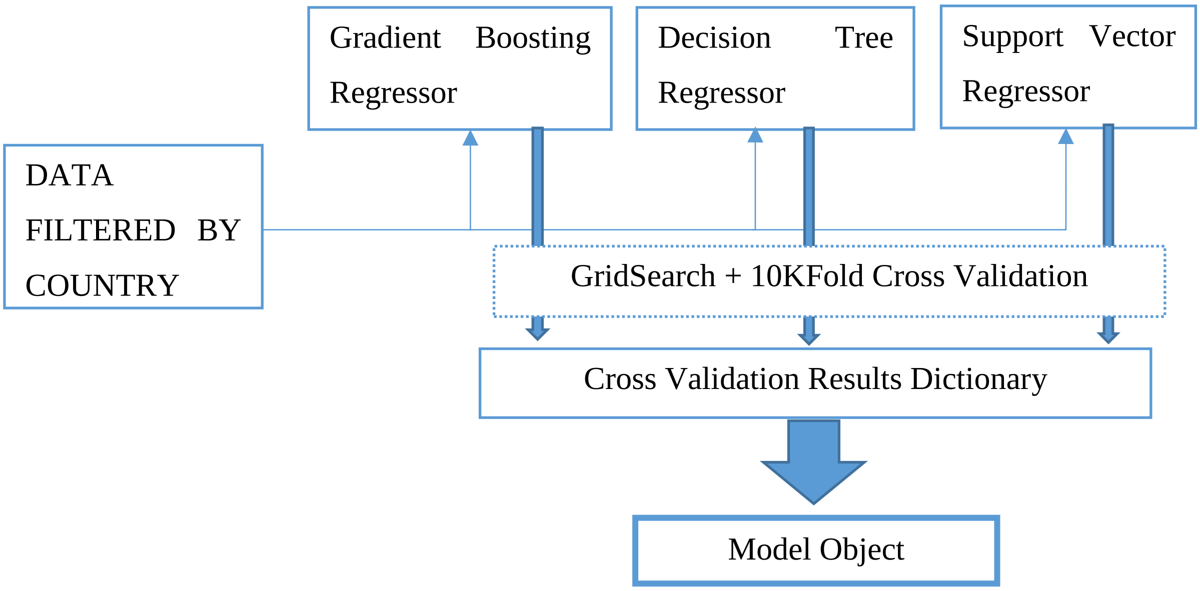

Gradient Boosting Regressor
Decision Tree Regressor
Support Vector Regressor
DATA FILTERED BY COUNTRY
GridSearch + 10KFold Cross Validation
Cross Validation Results Dictionary
Model Object

Figure 6. Best Model Selection Framework

With the models that are created, it is now possible to predict crude price individually. In order to consolidate these predictions, a collector algorithm is written. For each country, the selected model is run and the prediction is stored in an array. Out of the items that are listed in this array, the elements that are out of 10%-90% range are left out of the average calculation. The average of remaining predictions is the overall prediction of the algorithm. For testing purposes, year 2019 and 2018 are left out of the training sample. Due to low number of points in the hold out sample, predictions from training dataset is also included.

6. RESULTS

Creation of 86 models -one for each country- take a long time to complete with the current hardware. Over several iterations, the average time it takes to finish these tasks is about 90 minutes. This makes the re-iteration of model creation process a challenging one. With the final iteration the models are created and saved in order to be run over 2018 and 2019 data. However, since the predictions are done on monthly basis, there are only 16 values for out of sample testing. At the time the data is retrieved, the reports were only until April 2019. In order to generate more information on model accuracy, along with out of sample results, predictions for the whole training set is also provided. Figure 7 shows the prediction fit and the residuals graphs.

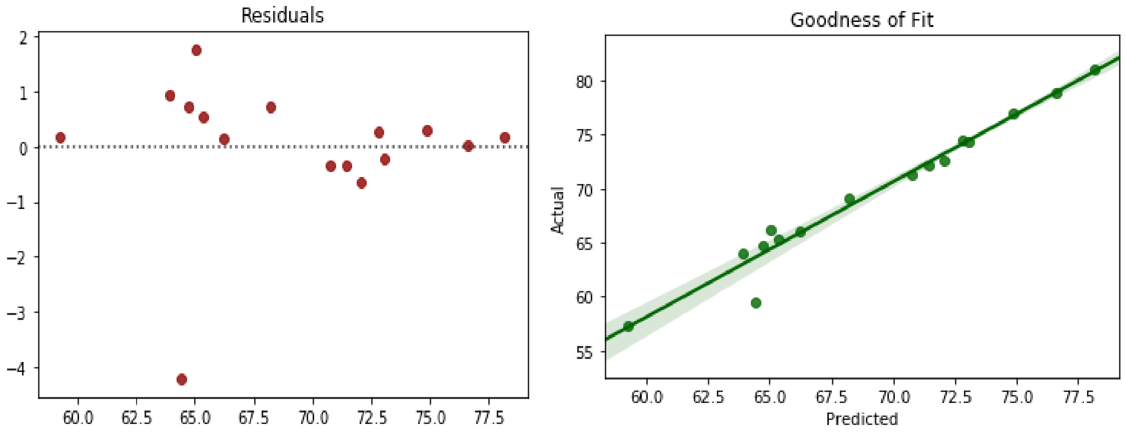


Figure 7. Residuals and Goodness of Fit graphs for hold out sample

For 16 points from 2018 and 2019, the prediction power of the model seems high, yet residual plot clearly shows that the model struggles to predict when crude prices are in the lower range. Table 4 shows a few metrics for the predictions. As of the final results, the lowest mean average error achieved is $1.298

Table 4. Metrics for Hold Out Sample predictions

R2 Score	Mean Average Error	Mean Squared Error
92%	1.298	3.341

Additionally, the models are also used to predict prices on training dataset. Figure 8 represents the results. The goodness of fit graph suggests a degree of overfitting, however the residuals plot employs randomness enough to assume that the model fitting is healthy.

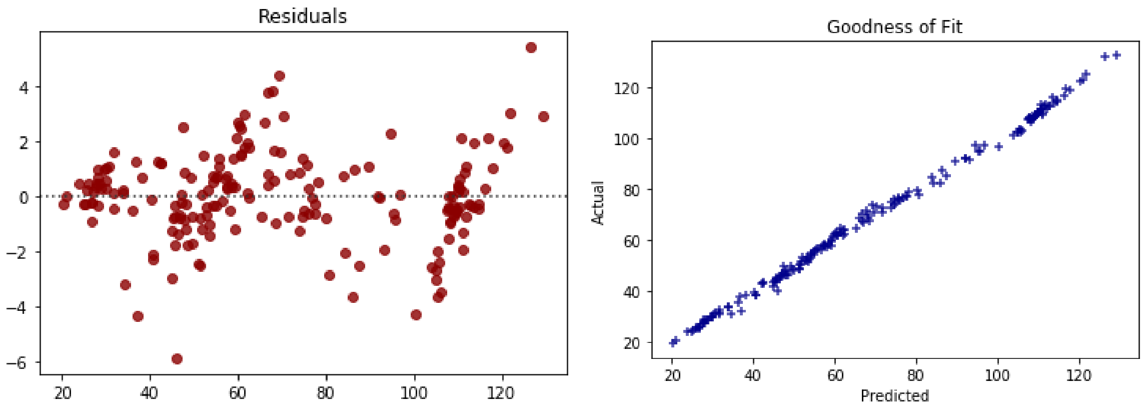


Figure 8. Residuals and Goodness of Fit graphs for train sample

Train set predictions has an R squared value of almost 99%, and a mean absolute error of 1.186. Using the outlier reduction function (APPENDIX B-Script 4) reduces the mean absolute error in both predictions, albeit only by a small amount. Given these results, the model has assumingly satisfactory results although there are numerous points to be discussed.

7. DISCUSSION AND FUTURE RESEARCH

In this project, the informative power of crude oil inventories, refinery capacities and monthly trading summaries is explored in terms of spot crude oil price predictions. Using the data from 2002 to current, a set of models is created, which contributes to the overall price prediction of the month in tandem. The initial studies show that this information, given that it is easily accessible, will amplify the performance of a predictor model as the error of the predictions is gradually reduced as the models grow more complex. However, there are many issues to be discussed if this kind of study is to be effectively used in a professional environment.

During the initial model fitting period, it is discovered that excluding recent years in this study drastically reduces the predictive power of the model. This suggests that while the data used in this project assists in prediction, it cannot be solely used to create a reliable model. An approach where only the few most recent years were included in training the model and employing a self-updating scheme to the model could both reduce training time and keep the model relevant through time. Furthermore, it is stated in project scope section that this study would not include time series analysis. For future reference, combining this model with a time series study or also feeding historical brent spot prices directly into the model at hand has the potential to return good results. Additionally, the models at hand do not have very strong forecasting implications due to the fact that they use information from 86 other countries. As a result, another future study opportunity would be to looking into how to reduce this number while keeping the model accuracy relatively high.

This project can be a solid starting point on establishing a usable and reliable forecasting algorithm whose methodology can also be implemented on the forecast of prices of other commodities or products. Making use of a constant stream of information that is being more and more widely available to public will eventually be a necessity if one wants to keep being relevant to the current establishments. The world is rapidly transitioning into an era of both data transparency and data privacy and it is important to figure out the available data and make good use of it

APPENDIX A – JODI OIL COUNTRY ABBREVIATIONS

The country codes are based on ISO-3166-1 alpha-2 standard:

http://www.iso.org/iso/home/store/publication_item.htm?pid=PUB500001%3aen

Code	Country	Code	Country	Code	Country
AE	United Arab Emirates	FR	France	OM	Oman
AO	Angola	HU	Hungary	PA	Panama
AR	Argentina	ID	Indonesia	PE	Peru
AT	Austria	IE	Ireland	PG	Papua New Guinea
AU	Australia	IN	India	PH	Philippines
AZ	Azerbaijan	IQ	Iraq	PL	Poland
BE	Belgium	IR	Iran	PT	Portugal
BG	Bulgaria	IT	Italy	QA	Qatar
BH	Bahrain	JM	Jamaica	RO	Romania
BN	Brunei Darussalam	GA	Gabon	RU	Russian Federation
BO	Bolivia	GB	United Kingdom	SA	Saudi Arabia
BR	Brazil	GQ	Equatorial Guinea	SE	Sweden
BY	Belarus	GR	Greece	SG	Singapore
CA	Canada	HR	Croatia	SK	Slovakia
CH	Switzerland	JP	Japan	SV	El Salvador
CL	Chile	KR	South Korea	TH	Thailand
CN	China	KW	Kuwait	TN	Tunisia
CO	Colombia	KZ	Kazakhstan	TR	Turkey
CR	Costa Rica	LT	Lithuania	TT	Trinidad and Tobago
CU	Cuba	LY	Libya	TW	Taiwan
CZ	Czechia	MA	Morocco	UA	Ukraine
DE	Germany	MM	Myanmar	US	United States of America
DK	Denmark	MX	Mexico	UY	Uruguay
DO	Dominican Republic	MY	Malaysia	VE	Venezuela
DZ	Algeria	NE	Niger	VN	Viet Nam
EC	Ecuador	NG	Nigeria	YE	Yemen
EG	Egypt	NI	Nicaragua	ZA	South Africa
ES	Spain	NL	Netherlands		
FI	Finland	NZ	New Zealand		

APPENDIX B – PYTHON SCRIPTS

Libraries Used

```
import pandas as pd                                                   [17]
import numpy as np                                                    [18]
import matplotlib.pyplot as plt                                       [19]
import seaborn as sns                                                 [19]
import joblib                                                         [16]
from sklearn.model_selection import KFold, GridSearchCV               [15]
from sklearn.svm import SVR
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.preprocessing import RobustScaler
%matplotlib inline
```

Script 1-Measurement Conversion

```
# Convert all measurements into ktons in order to not lose any values
import warnings
warnings.filterwarnings('ignore')
def convert_measurement(data):
    # Directly convert KBD into KBBL
    data.update(pd.DataFrame(data[data.UNIT_MEASURE=="KBD"].OBS_VALUE*data[data.UNIT_MEASURE=="KBD"].date.dt.daysinmonth, columns=["OBS_VALUE"]))
    # Directly convert KL into KBBL (1 KBBL = 6.2898 KL)
    data.update(pd.DataFrame(data[data.UNIT_MEASURE=="KL"].OBS_VALUE*6.2898, columns=["OBS_VALUE"]))
    # Convert thousand barrels into kilotons with the conversion factor
    for area in data.REF_AREA.unique():
        data.update(pd.DataFrame(data[data.REF_AREA==area][(data.UNIT_MEASURE!="KTONS") & (data.UNIT_MEASURE!="CONVBBL")].OBS_VALUE/data[data.REF_AREA==area][data.UNIT_MEASURE=="CONVBBL"].OBS_VALUE.unique()[0]*1000, columns=["OBS_VALUE"]))
```

Script 2–Transformations and Train-Test splits

```
df=pd.read_csv("project_data.csv")
# Final transformations
# Convert month to cyclic variable so that december and january are more closely related
# http://blog.davidkaleko.com/feature-engineering-cyclical-features.html
df['month_sin'] = np.sin((df.month-1)*(2.*np.pi/12))
df['month_cos'] = np.cos((df.month-1)*(2.*np.pi/12))
df.drop('month', axis=1, inplace=True)
df=df.iloc[:,1:]
# year 2019 and 2018 is left out of sample
oos_df=df[df.year.isin([2019,2018])]       # Out of Sample
train_df=df[~df.year.isin([2019,2018])]    # Training Data
# Scale the data
scl=RobustScaler()
scl_cols=train_df.iloc[:,:5]
train_df[scl_cols.columns] = scl.fit_transform(train_df[scl_cols.columns])  #Fit and transform training data
```

```python
oos_df[scl_cols.columns]=scl.transform(oos_df[scl_cols.columns])            #Transform hold out sample
# Just to make sure
train_df.dropna(axis=0, inplace=True)
oos_df.dropna(axis=0, inplace=True)
# Establish target and inputs
oos_target=oos_df.brent
oos_inputs=oos_df.drop('brent', axis=1)
train_target=train_df.brent
train_inputs=train_df.drop('brent', axis=1)
# drop country columns where there is no information (one hot encodes only has zeros)
numbers={}
cols_to_drop=[]
for c in train_inputs.iloc[:,5:118].columns:
    numbers[c]=sum(train_inputs[train_inputs[c]==1][c])
    if sum(train_inputs[train_inputs[c]==1][c])==0:
        cols_to_drop.append(c)
train_inputs.drop([k for k,v in numbers.items() if v == 0], axis=1, inplace=True)
```

Script 3-Best Model Selection Class

```python
# A class that fits three models with gridsearch and selects the best one for the current data
kfold=KFold(n_splits=10, shuffle=False, random_state=15)
class model_selector:
    def __init__(self, data, target):
        self.data=data
        self.target=target
    def dtree_selector(self, data, target):
        model=DecisionTreeRegressor()
        params={'max_depth':[3,5,10,16,22,30],'max_leaf_nodes':[None,2,3,4,5,6], 'min_samples_split':[3,5,10,12,15]}
        gridsearch=GridSearchCV(model, params, cv=kfold, return_train_score=False, scoring='neg_mean_absolute_error')
        gridsearch.fit(self.data, self.target)
        return(gridsearch)
    def svr_rbf_selector(self, data, target):
        model=SVR(kernel='rbf')
        params={'C':[1,2,3,4,5], 'gamma':[0.001,0.01,0.1,1]}
        gridsearch=GridSearchCV(model, params, cv=kfold, return_train_score=False, scoring='neg_mean_absolute_error')
        gridsearch.fit(self.data, self.target)
        return(gridsearch)
    def gbr_selector(self, data, target):
        model=GradientBoostingRegressor()
        params={'learning_rate':[0.05, 0.1,0.3,0.5], 'n_estimators':[100,200,300], 'max_depth':[3,5,6,8,11]}
        gridsearch=GridSearchCV(model, params, cv=kfold, return_train_score=False, scoring='neg_mean_absolute_error')
        gridsearch.fit(self.data, self.target)
        return(gridsearch)
    def evaluate(self):
        d1=self.dtree_selector(self.data, self.target)
        d2=self.svr_rbf_selector(self.data, self.target)
        d4=self.gbr_selector(self.data, self.target)
        results={'DTREE':d1.best_score_, 'RBF': d2.best_score_, 'GBR':d4.best_score_}
        return d1, d2, d4, results
    def output_model(self):
        res=self.evaluate()
        if max(res[3], key=res[3].get)=='DTREE':
            return res[0]
        elif max(res[3], key=res[3].get)=='RBF':
            return res[1]
        elif max(res[3], key=res[3].get)=='GBR':
            return res[2]
```

Script 4-Outlier Reducer Function

```
# Remove predictions that are outsite of 10-90% range then output the prediction
def outlier_reduce(results_df):
    predictions=np.zeros(results_df.shape[0])
    for i in range(results_df.shape[0]):
        res_arr=results_df[i]
        q90, q10 = np.percentile(res_arr, [90 ,10])
        iqr=q90-q10
        avg=res_arr.mean()
        predictions[i]=res_arr[(res_arr <= avg+iqr) & (res_arr >= avg-iqr)].mean()
    return predictions
```

Script 5-Model Training and Out of Sample Testing

```
# Create fitted models
countries=train_inputs.iloc[:,5:91].columns
for c in countries:
    train=train_inputs[train_inputs[c]==1]
    train=train.drop(countries, axis=1)
    target=train_target.loc[train.index]
    model=model_selector(train, target).output_model()
    modelname=str(c+"_model")                    # Save model with the country name for easy referencing
    joblib.dump(model, modelname)
# Apply models to out of sample
oos_inputs_fin=oos_inputs.drop(cols_to_drop, axis=1)
scores_oos=np.zeros(countries.shape[0])
y_results_oos=np.zeros((int(oos_inputs.shape[0]/countries.shape[0]), countries.shape[0]))
for c in countries:
    step=oos_inputs_fin[oos_inputs[c]==1]
    step.drop(countries, axis=1, inplace=True)
    goal=oos_target.loc[step.index]
    modelname=str(c+"_model")
    step_result=joblib.load(modelname).predict(step)
    scores_oos[np.where(countries==c)[0][0]]=mean_absolute_error(goal, step_result)
    y_results_oos[:,np.where(countries==c)[0][0]]=step_result
averages_oos=outlier_reduce(y_results_oos)          # Apply outlier reduce function to remove outliers from the result and return predictions
#averages_oos=pd.DataFrame(y_results_oos, columns=countries).mean(axis=1)
mean_absolute_error(goal, averages_oos)
mean_squared_error(goal, averages_oos)
r2_score(goal, averages_oos)
```

REFERENCES

[1] Scheitrum, D. P., Carter, C. A., Raworedo-Giha, C. (2018). WTI and Brent Pricing Structure. *Energy Economics* 72 462-469

[2] Miao, H., Ramchander, S., Wang, T., Yang, D. (2017). Influential factors in Crude Oil Price Forecasting. *Energy Economics* 68 77–88.

[3] Amadeo, K. (2019, May). What Affects Oil Prices? Three Critical Factors. Retrieved from https://www.thebalance.com/how-are-oil-prices-determined-3305650

[4] Li. J., Xu, Z., Yu, L., Tang, L. (2016). Forecasting oil price trends with sentiment of online news articles Procedia Computer Science 91 1081-1087

[5] Natarjan, S. G., Ashok, A. (2018). Multivariate Forecasting of crude Oil Spot Prices using Neural Networks. *ArXiv* abs/1811.08963.

[6] Xie, W., Yu, L., Xu, S., Wang, S. (2006). A New Method for Crude Oil Price Forecasting Based on Support Vector Machines. *LNCS* 3994 444-451.

[7] Li, T., Hu, Z., Jia, Y., Wu, J., Zhou, Y. (2018). Forecasting Crude Oil Prices Using Ensemble Empirical Mode Decomposition and Sparse Bayesian Learning. *Energies* 11 1882

[8] Mahdiani, M. R., Khamehchi, E. (2016). A modified neural network model for predicting the crude oil price. *Intellectual Economics* 71-77.

[9] Li, X., Shang, W., Wang, S. (2018). Text-Based crude oil price forecasting: A deep learning approach. *International Journal of Forecasting* https://doi.org/10.1016/j.ijforecast.2018.07.006

[10] Gao, S., Lei, Y. (2017). A new approach for crude oil price prediction based on stream learning. *Geoscience Frontiers 8* 183-187.

[11] Ye, M., Zyren, S., Shore, J. (2005). A monthly crude oil spot price forecasting using relative inventories. *International Journal of Forecasting 21* 491-501

[12] Jodi Oil Manual, 2nd Edition. Retieved from https://www.jodidata.org/oil/support/jodi-oil-manual.aspx

[13] Chen, J. (2019, April). U.S. Dollar Index - USDX Definition. Retrieved from https://www.investopedia.com/terms/u/usdx.asp

[14] Kaleko, D. (2017, October 30). Feature Engineering – Handling Cyclical Features [Blog Post]. Retrieved from http://blog.davidkaleko.com/feature-engineering-cyclical-features.html

[15] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[16] Varoquaux, G. et al. Joblib: Running Python Functions as Pipeline Jobs, 2008-, https://github.com/joblib/joblib

[17] Wes McKinney. **Data Structures for Statistical Computing in Python**, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

[17] Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37

[18] John D. Hunter. **Matplotlib: A 2D Graphics Environment**, Computing in Science & Engineering, **9**, 90-95 (2007),DOI:10.1109/MCSE.2007.55