1 **Multi-Time Series Averaging of Ensemble Machine Learning Models**

2 **Towards Crude Oil Price Forecasting**

3 Farid Javadnejad, Ph.D.[1]

4

5 1. Introduction

6

7   Crude oil and other refined liquid products from fossil fuels are critical contributors to the

8 world economy. Petroleum has been the largest energy source for all countries. Its products run

9 vehicles, heat buildings, and produce electricity. Moreover, various industries use petroleum as a

10 raw material to produce intermediate or end-user products that we use daily (EIA 2022b; Lu et al.

11 2021; Deng, Ma, and Zeng 2021; Kilian and Murphy 2014). In 2019, global petroleum

12 consumption neared 100 million barrels per day (Table 1).

13 **Table 1. The largest oil consumers and their share of total world consumption** (EIA 2022b)

Ranking	Country	Million barrels per day	Share of world total
1	United States	20.54	20%
2	China	14.01	14%
3	India	4.92	5%
4	Japan	3.74	4%
5	Russia	3.70	4%
6	Saudi Arabia	3.18	3%
7	Brazil	3.14	3%
8	Canada	2.63	3%
9	South Korea	2.60	3%
10	Germany	2.35	2%
	World total	100.23	

[1] Geospatial Consultant, Aerogeospatial LLC, 1401 21st Street Suite R, Sacramento, CA 95811, e-mail: nejad.fj@gmail.com.

1.1. Problem statement

Crude oil prices are difficult to predict accurately due to the number of influencing factors and the highly complex behavior of such influences. Global economic and social activities can be substantially impacted by fluctuations in crude oil prices. Threfore, despite challenges for prediction of oil price, accurate oil price forecasting is crucial for decision-making support for the manufacturing, logistics, and government sectors to guide industrial and social policies and practices (Kilian and Murphy 2014; Deng, Ma, and Zeng 2021; Lu et al. 2021).

1.2. Background

Brent, West Texas Intermediate (WTI), Dubai/Oman, and Shanghai crude oil prices are the major benchmarks of the crude oil market and are reported is USD per barrel unit. Factors such as supply and demand, financial markets and economics, politics, global events, renewable energy and alternative resources, new resources and development of new oil extraction technologies, social & environmental policies, and consumption patterns may influence the crude oil market dynamics. Such impacts and resultant price fluctuations might be very complex and may occur at different frequencies.

Classical econometric models such as random walk, autoregressive integrated moving average (ARIMA), error correction model (ECM), generalized autoregressive conditional heteroscedasticity (GARCH) model are used for crude oil price prediction. Recently, machine learning (ML) methods such as artificial neural network (ANN) and support vector machine (SVM) are used for the crude oil price prediction, which provide powerful tools to model nonlinear behavior or crude oil market dynamics (Jammazi and Aloui 2012; Lanza, Manera, and Giovannini 2005; Hou and Suardi 2012; Basiri 2015; Yu, Zhao, and Tang 2017; Murat and Tokat 2009; Kilian and Murphy 2014; Javadnejad 2012).

1.3. Objectives

In this work, an ML model is proposed to predict crude oil price using multiple infulencing factors. The predictions are casted on multiple time-series to consider for complex factors that imact the market dynamics in differenct frequencies.

This report is structured as follows. Section describes the metholdogy, where Sub-section 2.1 summarizes the datasets that are used in this study. Sub-section 2.2 describes the data preparation and data wrangling procedures. In Sub-section 2.3, the exploratory data analysis and feature engineering approcahes for ML training are described. Sub-section 2.4 covers the ML pre-processing, training, models selection, model metrics. In Sub-section 2.5, the final results for model training and validation are presented, as well as the predictions for 6-month time frequencies. We present the discussion of our results in Section 3. Finally, in Section 4, we summarise our recoomendaiton for future work.

2. Methodology

2.1. Datatets

The factors that influence the crude oil market dynamics include supply and demand, financial markets, politics, global events, alternative resources, development technologies, policies, and consumption patterns (Hamilton 2008; Hamilton 2009; Kilian and Murphy 2014; Zhao, Li, and Yu 2017; Lu et al. 2021; Wang, Wu, and Yang 2015). We use the crude oil prices of West Texas Intermediate (WTI) benchmark as the target feature. To take into account the aforementioned influencing factors a total of 32 feature variables were selected from publicly accessible data sources (EIA 2022a; FRED 2022; Investing 2022; WSJ 2022). Table 2 provides a list of the selected features, a description about each feature, and the sources of data.

Table 2. Selected dataset of feature variables for crude oil price

Category	Symbol	Variable	Unit	Source
Crude Oil Price	WTIPUUS	West Texas Intermediate Crude Oil Price	dollars per barrel	EIA
Supply	COPR_OPEC	Crude Oil Production, Total OPEC	million barrels per day	EIA
	PAPR_NONOPEC	Crude Oil Production, Total non-OPEC	million barrels per day	EIA
	INTL.55-1-WORL-TBPD	Crude Oil Production, NGPL, and other liquids production, World	thousand barrels per day	EIA
	COPRPUS	Crude Oil Production, U.S.	million barrels per day	EIA
Replacement Cost	RNGWHHD	Henry Hub Natural Gas Spot Price	dollars per million btu	EIA
Demand	PATC_OECD	Liquid Fuels Consumption, Total OECD	million barrels per day	EIA
	PATC_NON_OECD	Liquid Fuels Consumption, Total non-OECD,	million barrels per day	EIA
	FEDFUNDS	Federal Funds Effective Rate	percent, not seasonally adjusted	FRED
	IGREA	Index of Global Real Economic Activity	index, not seasonally adjusted	FRED
	CICPIUS	US Consumer Price Index (CPI): All Commodities	index, 1982-1984=1.00	EIA
	USACPIENGMINMEI	US Consumer Price Index (CPI): Energy for the United States	index 2015=100, not seasonally adjusted	FRED
	WPCPIUS	US Producer Price Index (PPI): All Commodities	index, 1982=1.00	EIA
	WP57IUS	US Producer Price Index (PPI): Petroleum	index, 1982=1.00	EIA
	EA19PIEAMI01GPM	roducer Price Index (PPI) of Euro Area (19 Countries)	index 2015=100, not seasonally adjusted	FRED
	ZOMNIUS	US Manufacturing Production Index (PMI)	index, 2017=100 (seasonally adjusted)	EIA
Inventory	PASC_OECD_T3	Petroleum Inventory, Total OECD	million barrels, end-of-period	EIA
	PASXPUS	Petroleum Inventory, US Total	million barrels, end-of-period	EIA
	COSQPUS	US Crude Oil Inventory: Strategic Petroleum Reserve (SPR)	million barrels, end-of-period	EIA
	COSXPUS	US Crude Oil Inventory: Non-SPR	million barrels, end-of-period	EIA
Monetary Market	RTWEXBG	Real Broad Dollar Index	index Jan 2006=100, not seasonally adjusted	FRED
	DXY	US Dollar Index (DXY)	index	Investing
	DEXUSEU	U.S. Dollars to Euro Spot Exchange Rate (DEXUSEU)	US dollars to one euro, not seasonally adjusted	FRED
Stock Market	SPX	S&P 500 Index	index	WSJ
	DJI	Dow Jones Industrial Index	index	WSJ
	COMP	NASDAQ index	index	WSJ
Commodity Market	Gold_Future	Gold Futures Historical Data	dollar per ounce	Investing

	Copper_Future	Copper Futures Historical Data	dollar per pound	Investing
Policy Uncertainty	GEPUCURRENT	Global Economic Policy Uncertainty Index: Current Price Adjusted GDP	index, not seasonally adjusted	FRED
Technology	MGWHUUS	Refiner Wholesale Gasoline Price	cents per gallon	EIA
	DSWHUUS	Diesel Fuel Refiner Wholesale Price	cents per gallon	EIA
	BREPUUS	Brent Crude Oil Spot Price	dollars per barrel	EIA

62

2.2. Data Cleaning and Data Wrangling

64   We used Jupyter Notebook 6.5.2 (Kluyver et al. 2016) and Python 3.9.15 (Python Software

65 Foundation 2022) to process the data. The features in Table 2 were read through APIs (if available)

66 or were downloaded directly from the data source. The data were initially set to be imported with

67 monthly intervals or averaged to monthly values, then were limited to the target time frame of

68 January 2000 and December 2022. The features were index based on their date values and then all

69 merged together on the date values to create corresponding feature values for each month. Figure

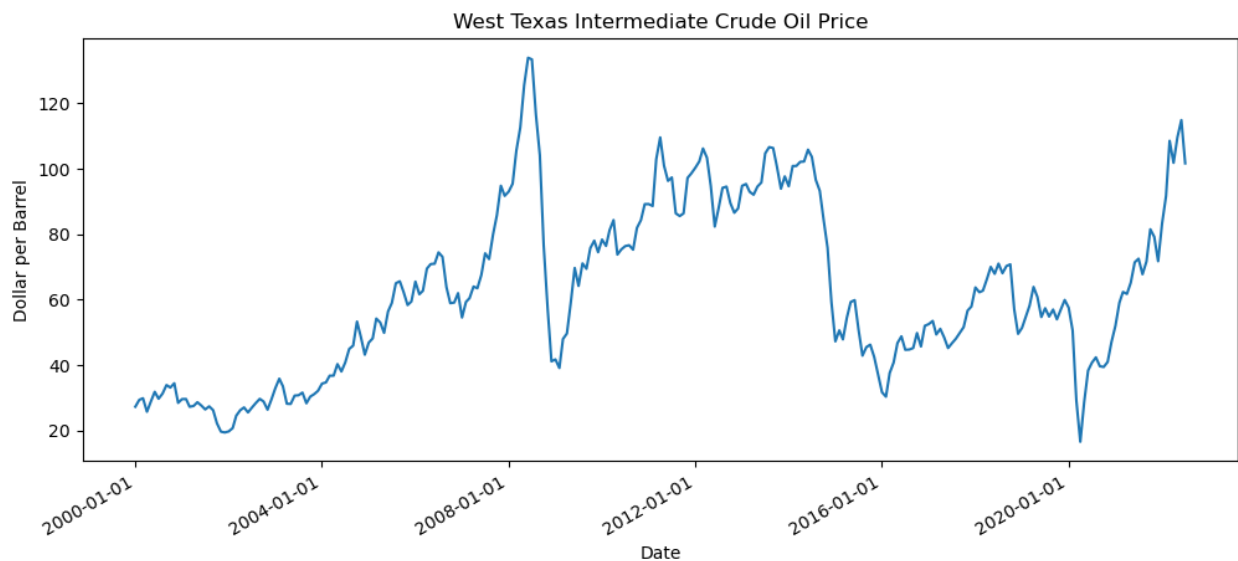70 1 shows the monthly West Texas Intermediate (WTI) crude oil price in the target time frame.

71

72 **Figure 1. West Texas Intermediate (WTI) crude oil price**

73

74   We used Pandas 1.5.2 (McKinney 2011) that is an open-source, simple, powerful, and

75   flexible library for data analysis and data manipulation, Pandas is for Python programming

76   language. The final dataset sized (274, 32) that represent 274-month records (rows) for 32 feature

77   variables (columns).

78   An important step of data wrangling is dealing with missing data. Table 3 shows the

79   summary statistics of missing data in feature variables. Missingno (Bilogur 2018) is also useful

80   tool that provides a series of visualisations for presence and distribution of the missing data within

81   a pandas dataframe. Figure 2 visually shows the distribution of the missing. To handle the missing

82   data, features with more than the 10% of missing data were dropped from the dataset. The columns

83   that had less than 1% missing features were imputed by using back and forward fill methods. For

84   the remaining missing data between 1% and 10%, the rows for all features were dropped to create

85   a dataset with no missing data. After treating missing data, the final dataframe sized (271, 30).

86

87   **Table 3. Summary statistics of missing data in feature variables**

Variable	Count	Percentage
oil_production_world	3	1.1%
petroleum_inventory_oecd	36	13.1%
global_real_econ_activity_index	1	0.4%
us_cpi_energy	1	0.4%
eu19_ppi	2	0.7%
real_dollar_index	72	26.3%
global_econ_policy_uncert_index	1	0.4%

88
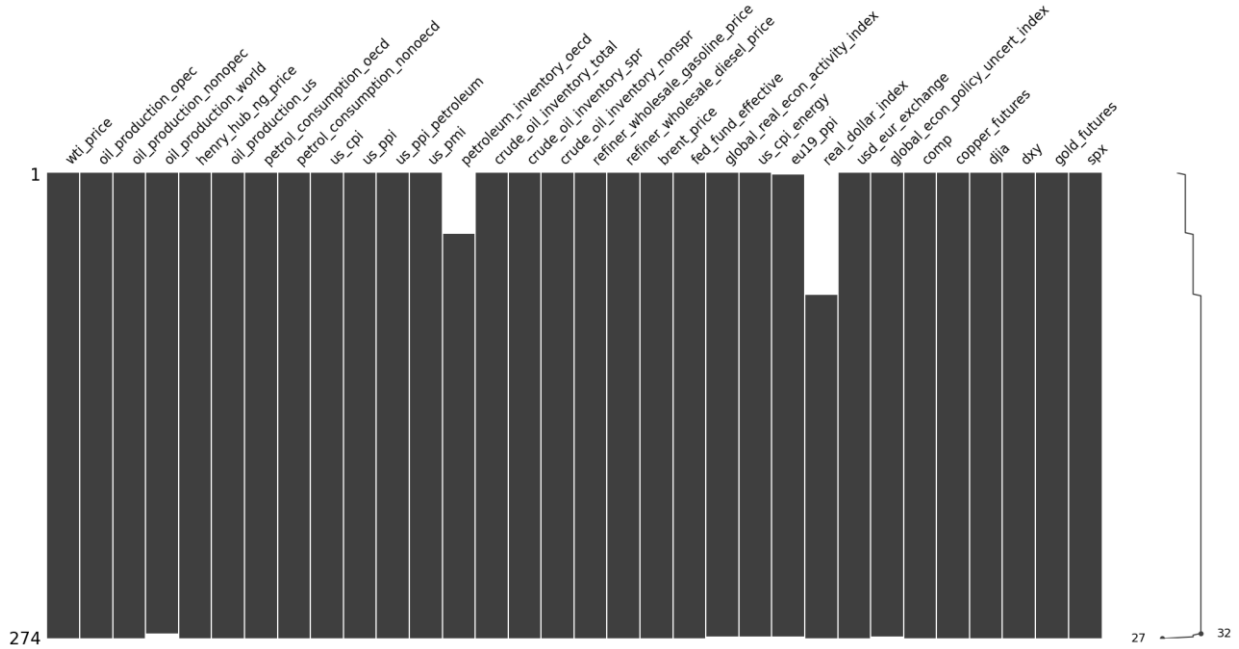
Figure 2. Missing data plot

2.1. Exploratory Data Analysis and Feature Engineering

WTI-Brent Spread is defined as difference between Brent and the WTI Crude Oil spot prices. WTI is produced from the Gulf of Mexico basin and is the benchmark for oil price in American market. On the other hand, Brent Oil is produced from the North Sea and Atlantic basin and is the pricing benchmark for European market. While, the prices for these two benchmarks are highly correlated; difference between the prices often reflect technical, supply/demand or geopolitical issues.

$$\text{WTI-Brent spread} = \text{WTI spot price} - \text{Brent spot price} \qquad (1)$$

The Crack spread is defined as the price difference between crude oil and its refined oil, reflecting the supply and demand relationship between the crude oil market and its refined product market   (Wang, Wu, and Yang 2015).

$$\text{WTI crack spread} = 3 \times \text{WTI spot price} - 2 \times \text{Gasoline Price} - 1 \times \text{Diesel Fuel Price} \qquad (2)$$

$$\text{Brent crack spread} = 3 \times \text{Brent spot price} - 2 \times \text{Gasoline Price} - 1 \times \text{Diesel Fuel Price} \qquad (3)$$

7

101   After calculating the spreads, Brent price, refiner wholesale gasoline price, and refiner

102   wholesale diesel price were dropped from the dataset.

103   Figure 3 shows the histograms of initial feature variables and feature engineered variables.

104   Figure 4 includes scatterplots of feature variable against the target variable of WTI oil price. Figure

105   5 is the Pearson correlation coefficients matrix for feature variables, and Figure 6 shows the

106   Predictive Power Score (PPS) matrix for feature variables.
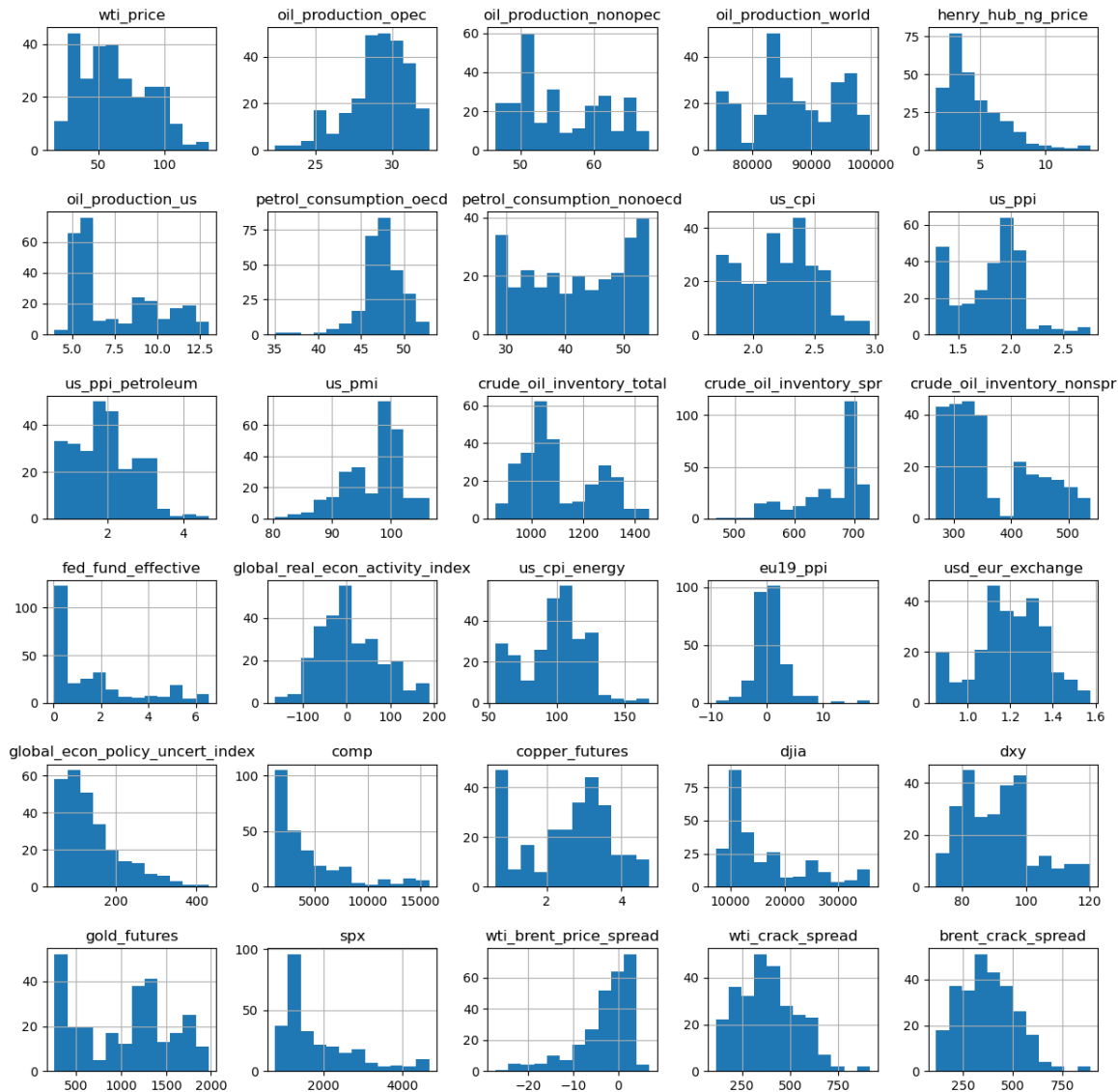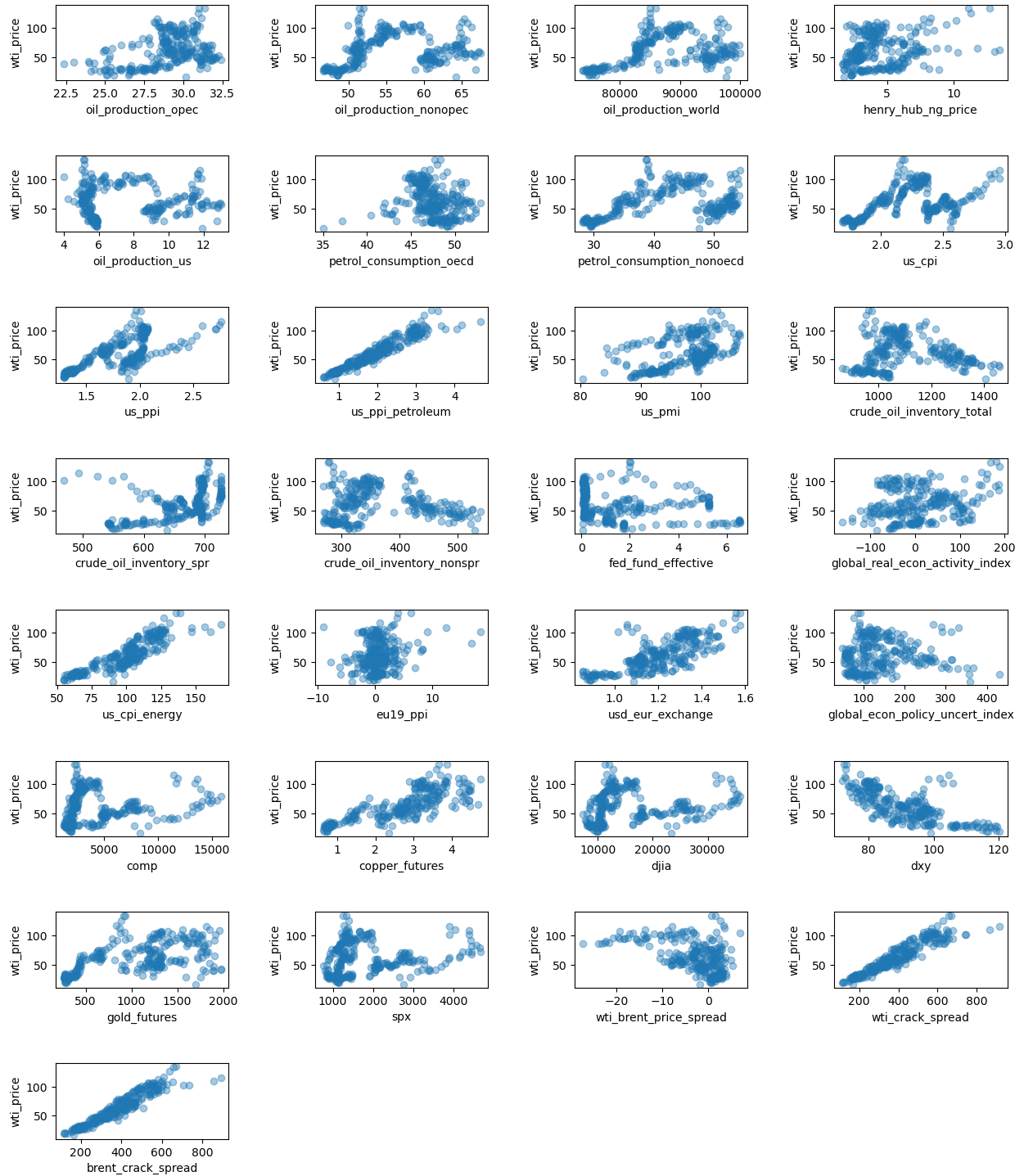
107

Figure 3. Histograms of feature variables

8

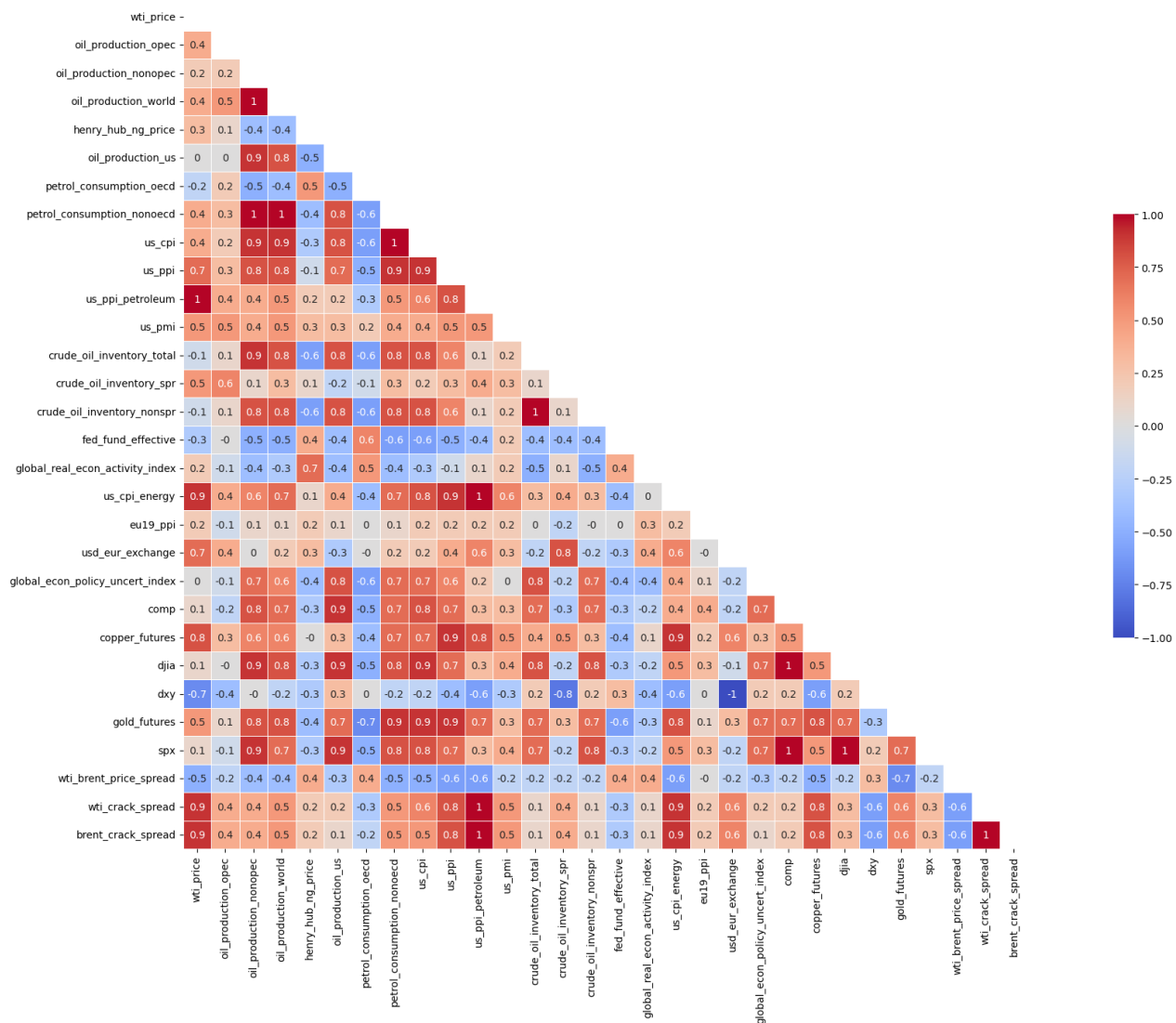**Figure 4. Scatter plots of feature variable against the target variable**

113
114   **Figure 5. Pearson correlation coefficients matrix for feature variables**
115

Figure 6 (PPS matrix):

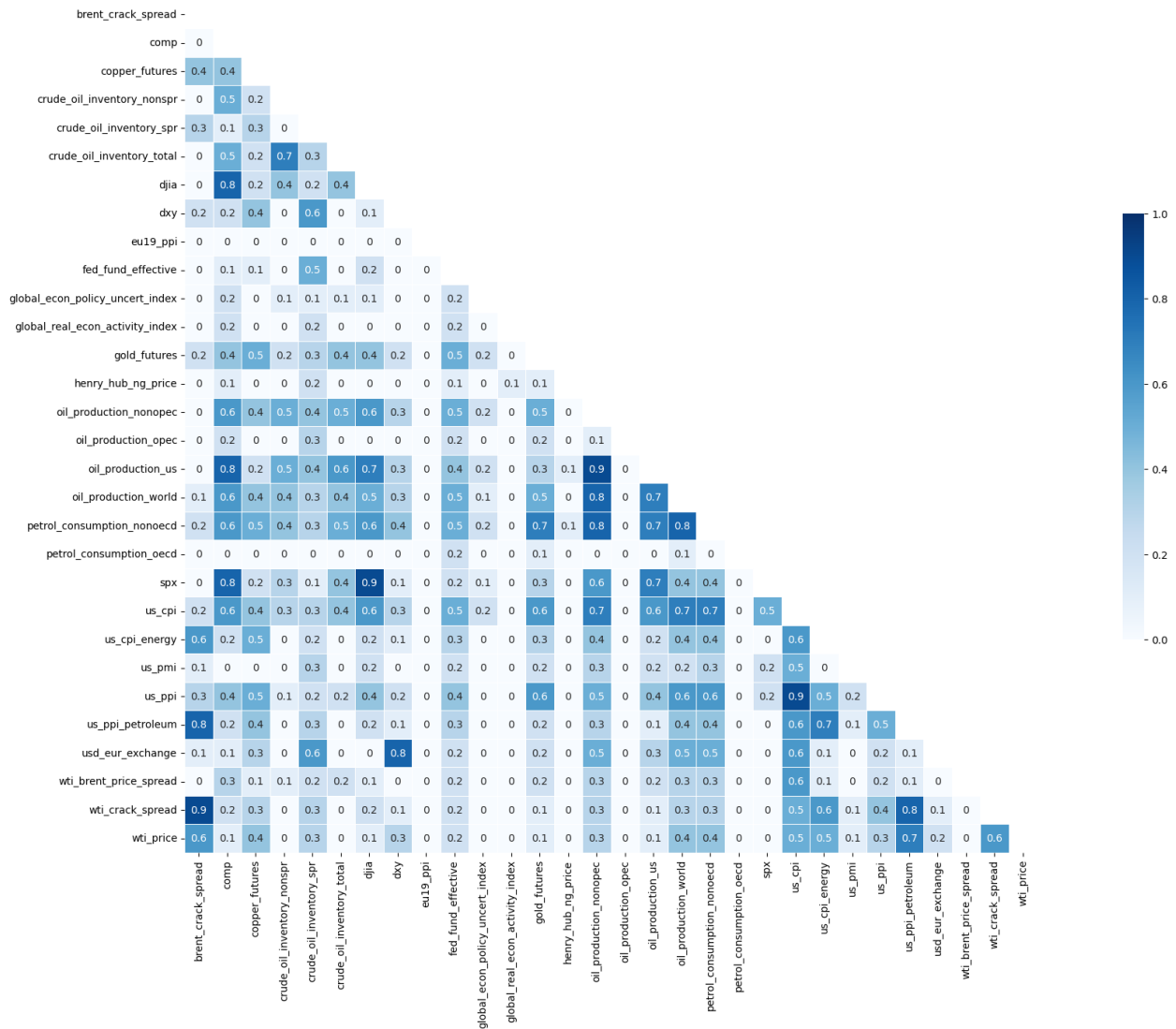|	brent_crack_spread	comp	copper_futures	crude_oil_inventory_nonspr	crude_oil_inventory_spr	crude_oil_inventory_total	djia	dxy	eu19_ppi	fed_fund_effective	global_econ_policy_uncert_index	global_real_econ_activity_index	gold_futures	henry_hub_ng_price	oil_production_nonopec	oil_production_opec	oil_production_us	oil_production_world	petrol_consumption_nonoecd	petrol_consumption_oecd	spx	us_cpi	us_cpi_energy	us_pmi	us_ppi	us_ppi_petroleum	usd_eur_exchange	wti_brent_price_spread	wti_crack_spread	wti_price
brent_crack_spread																														
comp	0																													
copper_futures	0.4	0.4																												
crude_oil_inventory_nonspr	0	0.5	0.2																											
crude_oil_inventory_spr	0.3	0.1	0.3	0																										
crude_oil_inventory_total	0	0.5	0.2	0.7	0.3																									
djia	0	0.8	0.2	0.4	0.2	0.4																								
dxy	0.2	0.2	0.4	0	0.6	0	0.1																							
eu19_ppi	0	0	0	0	0	0	0	0																						
fed_fund_effective	0	0.1	0.1	0	0.5	0	0.2	0	0																					
global_econ_policy_uncert_index	0	0.2	0	0.1	0.1	0.1	0.1	0	0	0.2																				
global_real_econ_activity_index	0	0.2	0	0	0.2	0	0	0	0	0.2	0																			
gold_futures	0.2	0.4	0.5	0.2	0.3	0.4	0.4	0.2	0	0.5	0.2	0																		
henry_hub_ng_price	0	0.1	0	0	0.2	0	0	0	0	0.1	0	0.1	0.1																	
oil_production_nonopec	0	0.6	0.4	0.5	0.4	0.5	0.6	0.3	0	0.5	0.2	0	0.5	0																
oil_production_opec	0	0.2	0	0	0.3	0	0	0	0	0.2	0	0	0.2	0	0.1															
oil_production_us	0	0.8	0.2	0.5	0.4	0.6	0.7	0.3	0	0.4	0.2	0	0.3	0.1	0.9	0														
oil_production_world	0.1	0.6	0.4	0.4	0.3	0.4	0.5	0.3	0	0.5	0.1	0	0.5	0	0.8	0	0.7													
petrol_consumption_nonoecd	0.2	0.6	0.5	0.4	0.3	0.5	0.6	0.4	0	0.5	0.2	0	0.7	0.1	0.8	0	0.7	0.8												
petrol_consumption_oecd	0	0	0	0	0	0	0	0	0	0.2	0	0	0.1	0	0	0	0	0.1	0											
spx	0	0.8	0.2	0.3	0.1	0.4	0.9	0.1	0	0.2	0.1	0	0.3	0	0.6	0	0.7	0.4	0.4	0										
us_cpi	0.2	0.6	0.4	0.3	0.3	0.4	0.6	0.3	0	0.5	0.2	0	0.6	0	0.7	0	0.6	0.7	0.7	0	0.5									
us_cpi_energy	0.6	0.2	0.5	0	0.2	0	0.2	0.1	0	0.3	0	0	0.3	0	0.4	0	0.2	0.4	0.4	0	0	0.6								
us_pmi	0.1	0	0	0	0.3	0	0.2	0	0	0.2	0	0	0.2	0	0.3	0	0.2	0.2	0.3	0	0.2	0.5	0							
us_ppi	0.3	0.4	0.5	0.1	0.2	0.2	0.4	0.2	0	0.4	0	0	0.6	0	0.5	0	0.4	0.6	0.6	0	0.2	0.9	0.5	0.2						
us_ppi_petroleum	0.8	0.2	0.4	0	0.3	0	0.2	0.1	0	0.3	0	0	0.2	0	0.3	0	0.1	0.4	0.4	0	0	0.6	0.7	0.1	0.5					
usd_eur_exchange	0.1	0.1	0.3	0	0.6	0	0	0.8	0	0.2	0	0	0.2	0	0.5	0	0.3	0.5	0.5	0	0	0.6	0.1	0	0.2	0.1				
wti_brent_price_spread	0	0.3	0.1	0.1	0.2	0.2	0.1	0	0	0.2	0	0	0.2	0	0.3	0	0.2	0.3	0.3	0	0	0.6	0.1	0	0.2	0.1	0			
wti_crack_spread	0.9	0.2	0.3	0	0.3	0	0.2	0.1	0	0.2	0	0	0.1	0	0.3	0	0.1	0.3	0.3	0	0	0.5	0.6	0.1	0.4	0.8	0.1	0		
wti_price	0.6	0.1	0.4	0	0.3	0	0.1	0.3	0	0.2	0	0	0.1	0	0.3	0	0.1	0.4	0.4	0	0	0.5	0.5	0.1	0.3	0.7	0.2	0	0.6	

Figure 6. Predictive Power Score (PPS) matrix for feature variables

2.2. Pre-processing and Training

- *Pre-processing*

- Shift n-month
- Handle missing values
- Score association with target feature

Figure 5.a presents a bar chart of sorted correlation coefficients against WTI oil price where values larger than 0.6 and smaller than -0.6 are highlighted in red. Figure 6.a shows the Predictive

126   Power Score (PPS) matrix for feature variables, Figure 6.b is bar chart of sorted PPS association

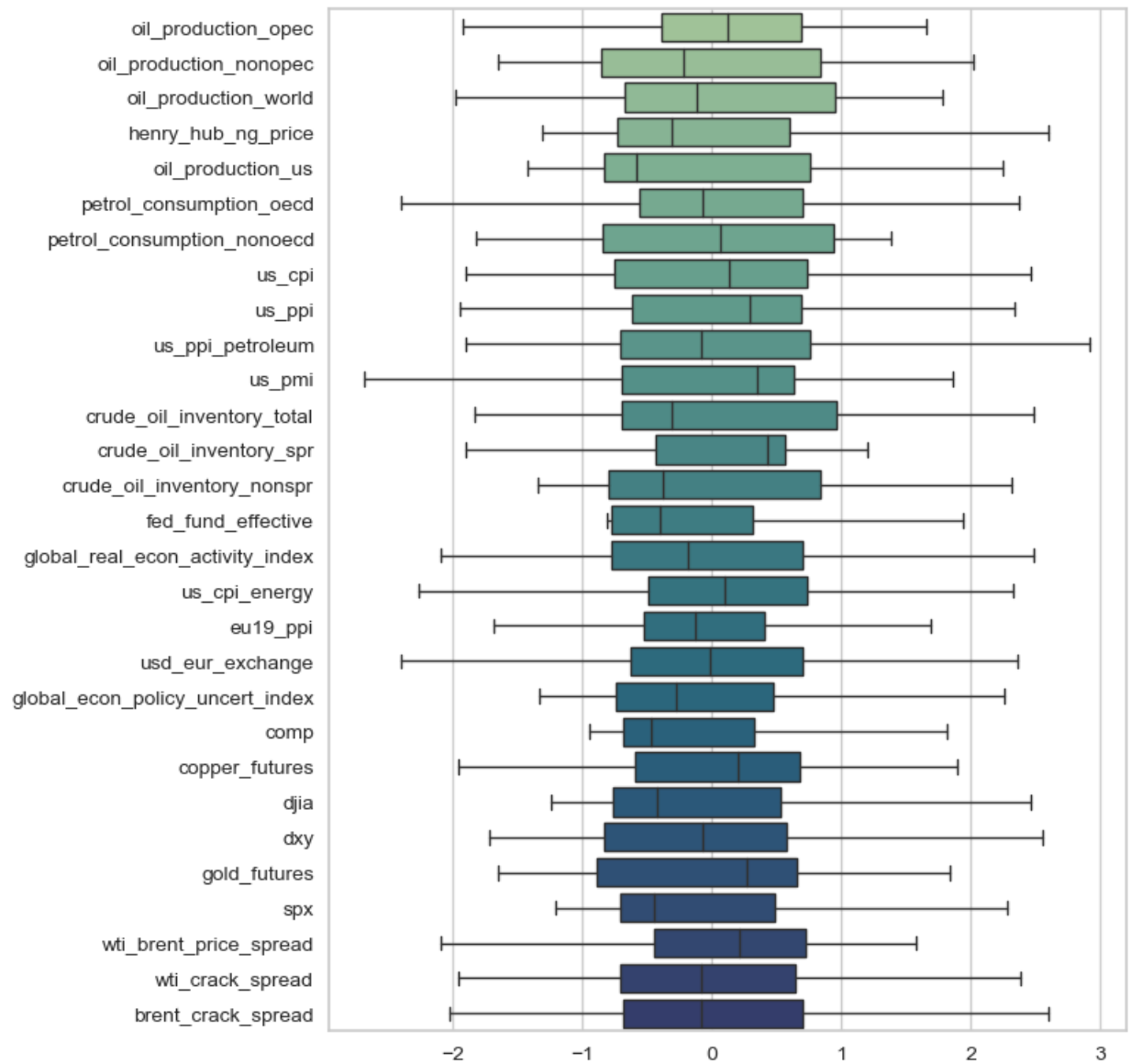127   against WTI oil price, where values larger than 0.6 are highlighted in red.

, b) Bar chart of correlation coefficients with the target feature, where values larger than 0.6 and smaller than -0.6 are highlighted in red

b) Bar chart of PPS association with the target feature, where values larger than 0.6 are highlighted in red.

128
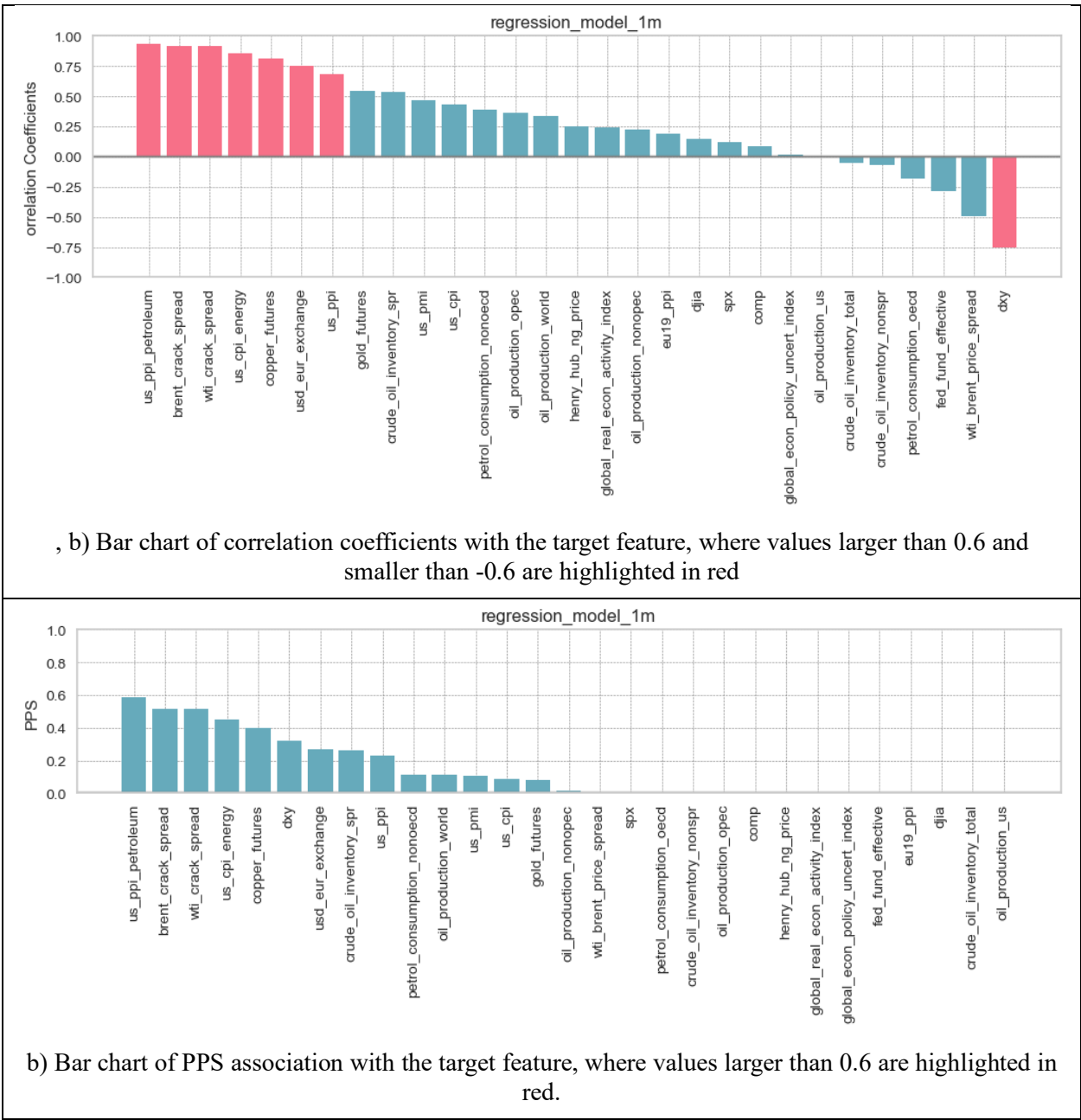
129   -   Train/Test Split

130   -   scaling

Figure 7

- *Training*

We use Pycaret, an open-source low code Python library that automates machine learning (AutoML) models to construct and deploy the models (Moez 2022). The library manages twenty-five different algorithms for regression, such as Extra Trees Regressor (ET), Gradient Boosting Regressor (GBR), Extreme Gradient Boosting Regressor (XGB), Random Forest Regressor (RF),

139 Linear Regression (LR), AdaBoost Regressor (ADA), and eighteen other algorithms for

140 classification.

141   We compare the performance of twenty-five AutoML models based on coefficient of

142 determination (R^2), Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean

143 Squared Error (RMSE). Then, we select the top five models for each time-series and build

144 ensemble models. Ensemble methods benefit different training algorithms for increasing the

145 training accuracy for reaching a higher testing accuracy to substantially improve the accuracy of

146 the integerated model (Ardabili, Mosavi, and Várkonyi-Kóczy 2020).

147

148 #Setup Pycaret Regression session
149 #Transform dataset, normalize and split the dataset.
150

Description	Value
Session id	786
Target	wti_price
Target type	Regression
Data shape	(270, 30)
Train data shape	(188, 30)
Test data shape	(82, 30)
Numeric features	29
Preprocess	True
Imputation type	simple
Numeric imputation	mean
Categorical imputation	constant
Low variance threshold	0
Transformation	True
Transformation method	yeo-johnson
Normalize	True
Normalize method	zscore
Fold Generator	KFold
Fold Number	10
CPU Jobs	-1
Use GPU	FALSE
Log Experiment	FALSE

Experiment Name	reg-default-name
USI	79e4

151

152   # Use compare_models() function to compare ML models and their performance on the data
153   # Use the default fold value from 10 to 5 for cross-validation generator

154
155   # Store the top 5 models in top5.
156   # We plan to use this for Stacking and Blending purposes.

157
158

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
et	Extra Trees Regressor	4.51	40.47	6.32	0.95	0.12	0.08
gbr	Gradient Boosting Regressor	5.10	48.24	6.85	0.94	0.12	0.09
lightgbm	Light Gradient Boosting Machine	5.22	53.37	7.25	0.93	0.12	0.09
rf	Random Forest Regressor	5.30	55.06	7.38	0.93	0.13	0.10
lr	Linear Regression	5.90	56.68	7.49	0.93	0.16	0.12
ada	AdaBoost Regressor	5.80	59.57	7.63	0.92	0.13	0.11
huber	Huber Regressor	5.81	58.89	7.64	0.92	0.16	0.12
ridge	Ridge Regression	5.94	59.68	7.69	0.92	0.16	0.12
br	Bayesian Ridge	6.02	62.60	7.88	0.92	0.16	0.12
lasso	Lasso Regression	6.26	68.05	8.22	0.91	0.16	0.12
knn	K Neighbors Regressor	5.89	73.74	8.42	0.90	0.14	0.11
omp	Orthogonal Matching Pursuit	6.31	75.90	8.70	0.90	0.17	0.12
en	Elastic Net	6.86	79.07	8.86	0.89	0.17	0.13
dt	Decision Tree Regressor	6.64	84.37	9.07	0.88	0.15	0.12
par	Passive Aggressive Regressor	8.66	132.63	10.90	0.82	0.31	0.19
llar	Lasso Least Angle Regression	12.88	249.70	15.76	0.67	0.28	0.26
dummy	Dummy Regressor	23.21	756.93	27.45	-0.01	0.48	0.50

159

160   Hyper-parameterization
161   Tune the top 5 models dynamically with a higher iteration rate (n_iter).
162   This is to find more optimal hyper parameters over a larger search space.

163
164   tuned_top5[i] = tune_model(top, n_iter=120, optimize='RMSE', fold=5)
165   `Fitting 5 folds for each of 120 candidates, totalling 600 fits`
166   `Original model was better than the tuned model, hence it will be returned.`
167   `NOTE: The display metrics are for the tuned model (not the original one).`

168
169
170   ### Combine top 5 models
171   #### Build blend models
172   # Building a blending model from our top 5 models and evaluate the results.
173   # Blending models is an ensemble method of combining different machine learning algorithms.

```
174    # Use a majority vote to build consensus of final prediction values.
175
176    blender_specific = blend_models(estimator_list=tuned_top5[0:], fold=5, optimize='RMSE',
177    choose_better=False)
178
```
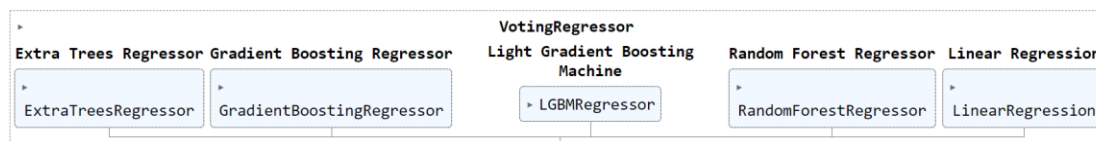
Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	4.7849	36.0137	6.0011	0.9562	0.0872	0.0736
1	4.4661	47.8932	6.9205	0.9324	0.1016	0.0757
2	4.3926	28.0386	5.2951	0.9577	0.0908	0.0752
3	5.4943	53.3835	7.3064	0.9415	0.1802	0.1332
4	3.8538	29.3091	5.4138	0.9554	0.1047	0.0748
Mean	4.5984	38.9276	6.1874	0.9486	0.1129	0.0865
Std	0.5388	10.0889	0.8024	0.01	0.0343	0.0234

```
179
180
```



```
181
182
183
184    #### Build stack models
185    # Build a stacking model from our top 5 models and evaluate the results.
186    # Stacking models is an ensemble method of using meta learning.
187    # A meta model is created using multiple base estimators to generate the final prediction.
188
189    stacker_specific = stack_models(estimator_list=tuned_top5[1:],
190                    meta_model=tuned_top5[0],
191                    fold=5,
192                    optimize='RMSE',
193                    choose_better=False)
194
```

Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	3.9086	23.294	4.8264	0.9717	0.0643	0.0563
1	5.3399	61.0785	7.8153	0.9138	0.1114	0.0851
2	4.1576	26.6576	5.1631	0.9597	0.0903	0.0729
3	5.7948	61.4996	7.8422	0.9326	0.1865	0.138
4	3.7863	26.4179	5.1398	0.9598	0.1006	0.0716
Mean	4.5974	39.7895	6.1574	0.9475	0.1106	0.0848
Std	0.8137	17.5949	1.3699	0.0212	0.041	0.0281

```
195
196
```

197
198
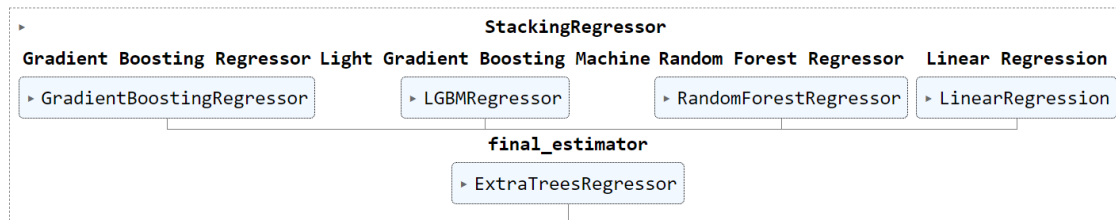
Evaluate the performances

(a)

(b)

(c)

(d)

(e)

(f)
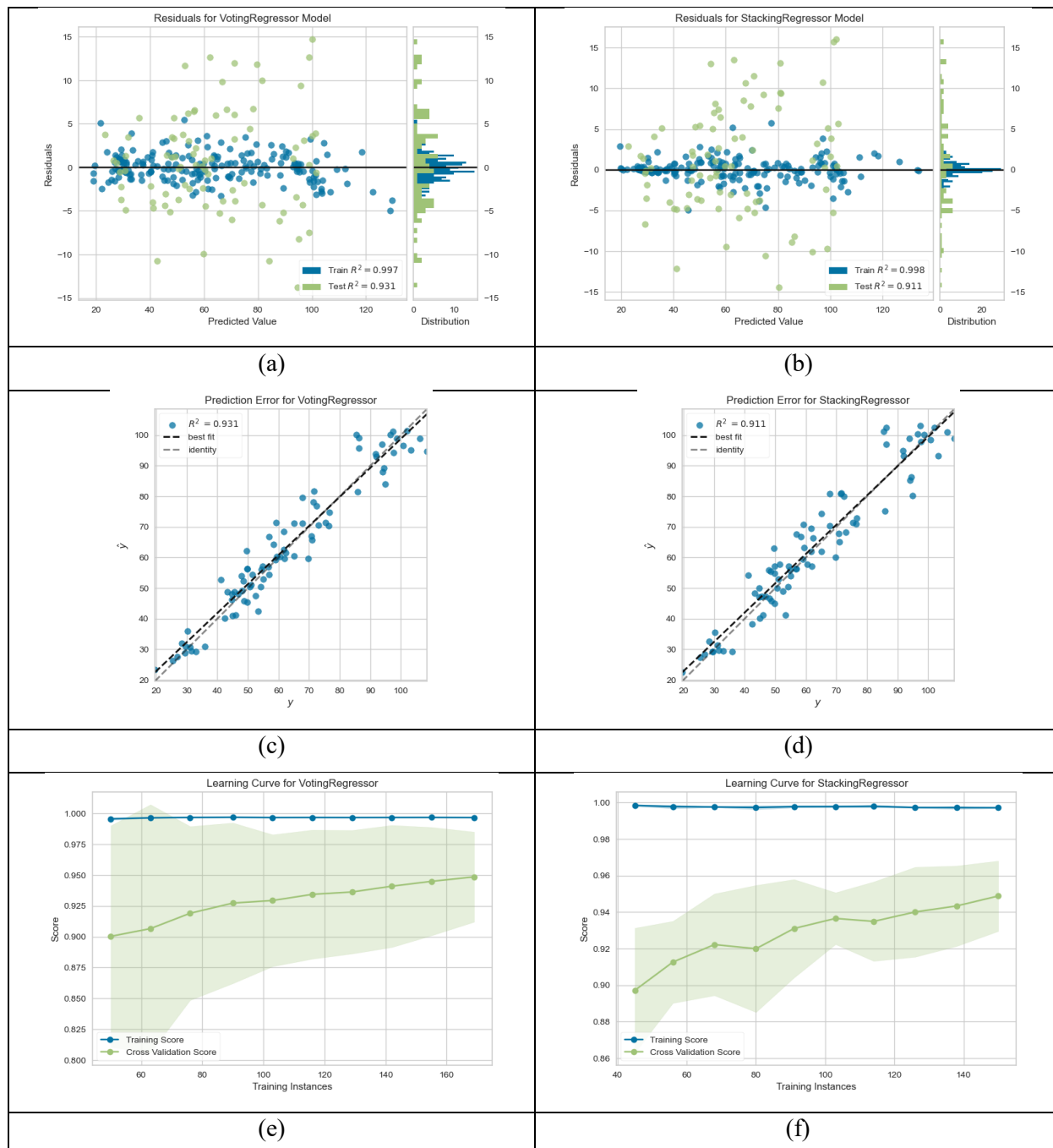
200

201   # Execute the predict_model() function to use the blender model to generate the predicted values.

202   # Execute the predict_model() function to use the stack model to generate the predicted values.
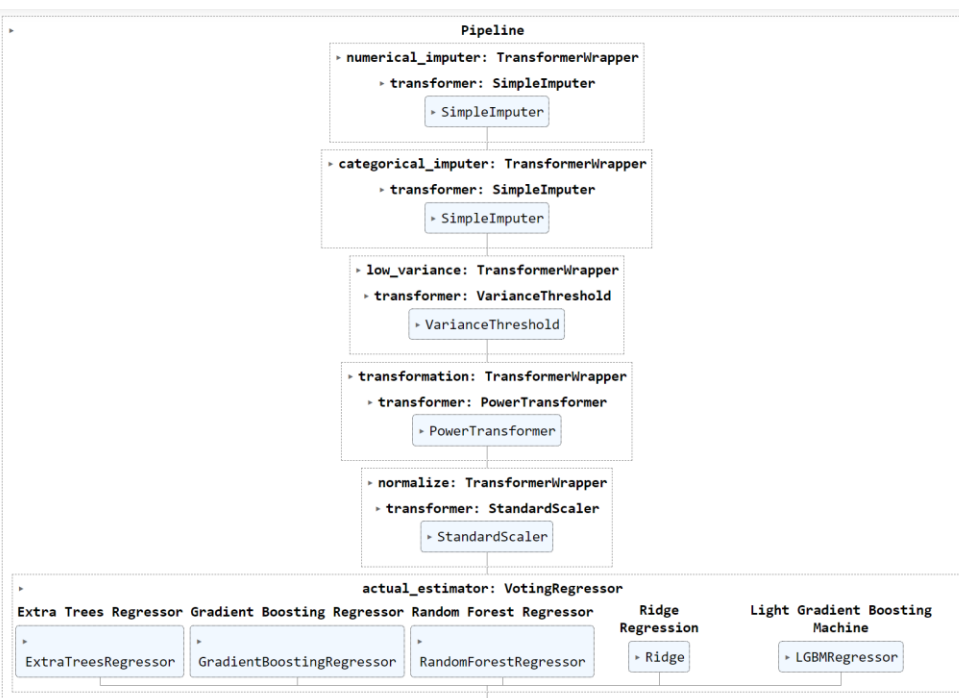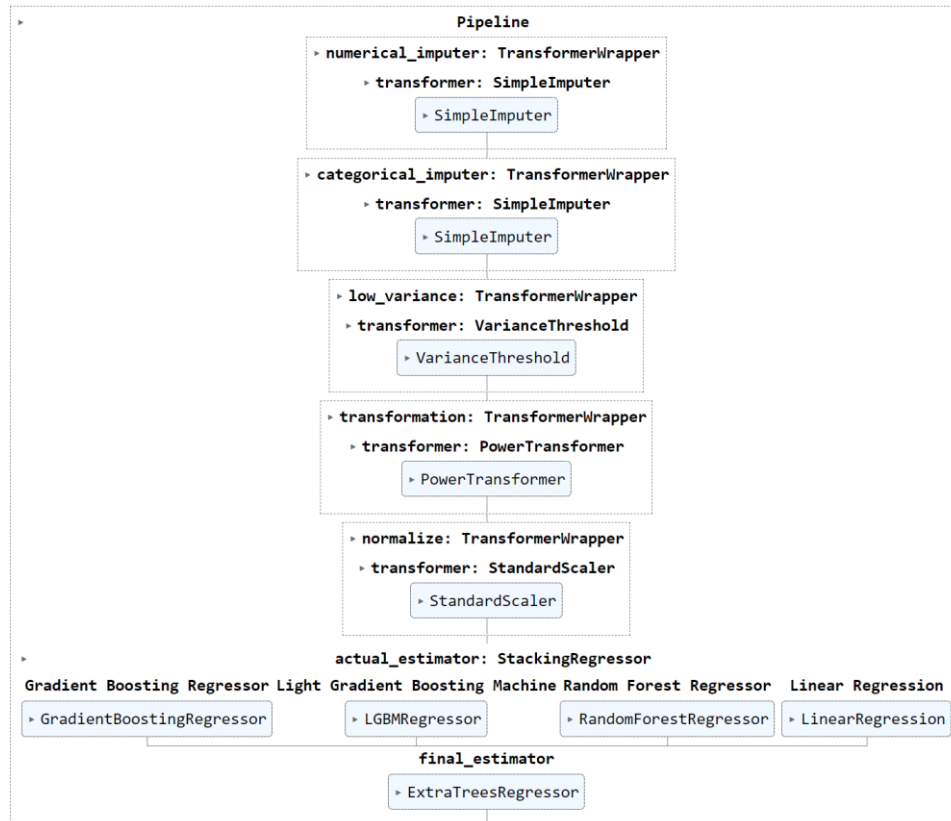
203

Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
Voting Regressor	4.5871	34.2862	5.8554	0.9308	0.093	0.0773
Stacking Regressor	5.2775	44.2081	6.6489	0.9107	0.1042	0.0881

204

205   Save model

206

207   **2.3. Modelling**

208   Finally, the multi-frequnecy prediction time-series are weight-averaged based on the

209   perfromance of the ML model into a single integrated predication series that represent the final oil

210   price predictions.

211

212   ## Analyze the Performance of Final Models on Entire Dataset

Pipeline

numerical_imputer: TransformerWrapper

transformer: SimpleImputer

SimpleImputer

categorical_imputer: TransformerWrapper

transformer: SimpleImputer

SimpleImputer

low_variance: TransformerWrapper

transformer: VarianceThreshold

VarianceThreshold

transformation: TransformerWrapper

transformer: PowerTransformer

PowerTransformer

normalize: TransformerWrapper

transformer: StandardScaler

StandardScaler

actual_estimator: StackingRegressor

Gradient Boosting Regressor	Light Gradient Boosting Machine	Random Forest Regressor	Linear Regression
GradientBoostingRegressor	LGBMRegressor	RandomForestRegressor	LinearRegression

final_estimator

ExtraTreesRegressor

213

Pipeline

numerical_imputer: TransformerWrapper

transformer: SimpleImputer

SimpleImputer

categorical_imputer: TransformerWrapper

transformer: SimpleImputer

SimpleImputer

low_variance: TransformerWrapper

transformer: VarianceThreshold

VarianceThreshold

transformation: TransformerWrapper

transformer: PowerTransformer

PowerTransformer

normalize: TransformerWrapper

transformer: StandardScaler

StandardScaler

actual_estimator: VotingRegressor

Extra Trees Regressor	Gradient Boosting Regressor	Random Forest Regressor	Ridge Regression	Light Gradient Boosting Machine
ExtraTreesRegressor	GradientBoostingRegressor	RandomForestRegressor	Ridge	LGBMRegressor

214

215

216

217   regression_model_1m

19

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Stacking Regressor	0.9828	1.9863	1.4094	0.9971	0.0246	0.0164

218

219   `regression_model_2m`

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Voting Regressor	1.4974	4.0136	2.0034	0.9941	0.0385	0.0273

220

221

222 `regression_model_3m`

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Stacking Regressor	1.3449	4.1664	2.0412	0.9938	0.0344	0.0224

223

224 `regression_model_4m`

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Voting Regressor	1.8882	6.3462	2.5192	0.9905	0.0503	0.0346

225   `regression_model_5m`

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Stacking Regressor	1.2967	5.0315	2.2431	0.9925	0.0553	0.025

226

227 `regression_model_6m`

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Stacking Regressor	1.2893	5.3536	2.3138	0.992	0.0585	0.0253

228

229

230

21

Linear Relationship between Actual Value and Predicted Value

231

Residuals for the Predicted values in Final Model

232

233
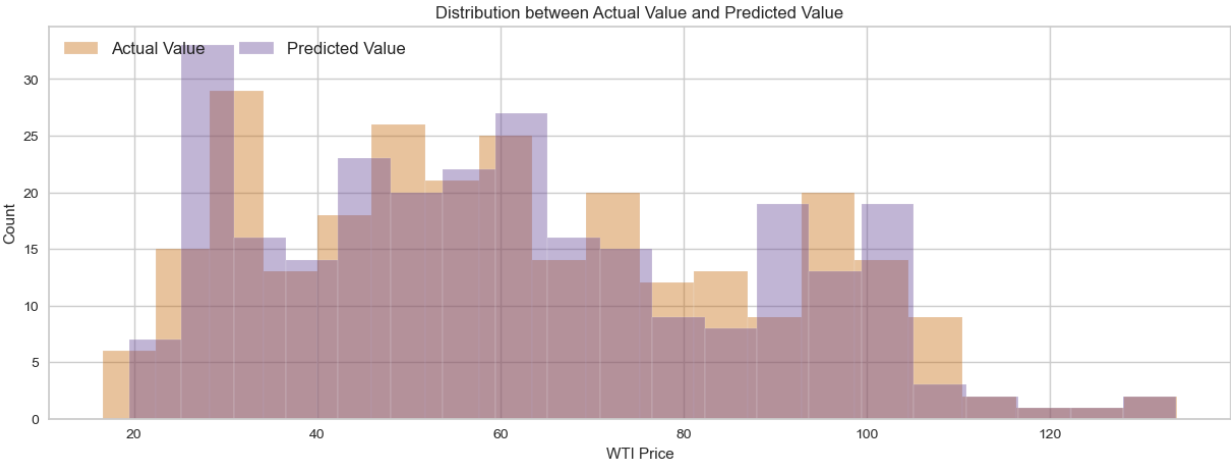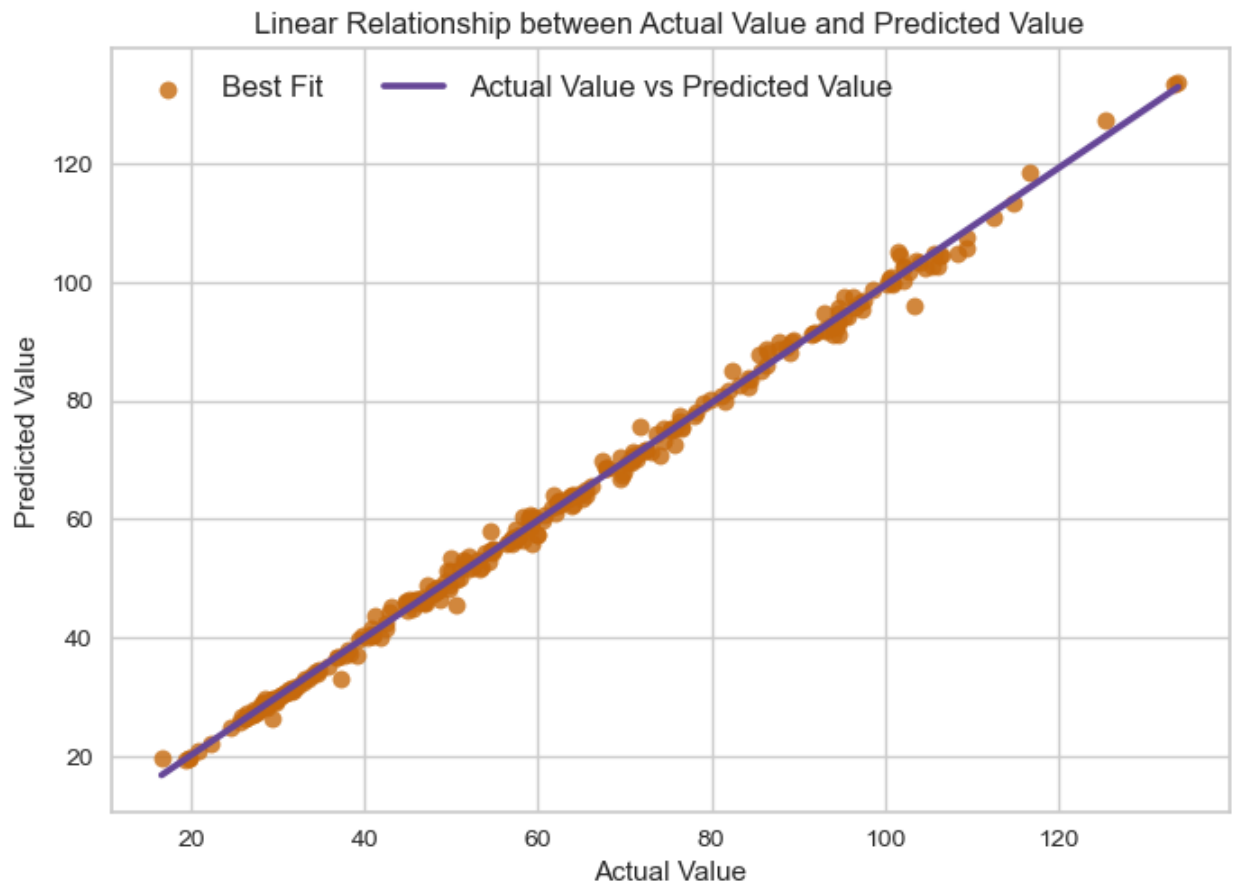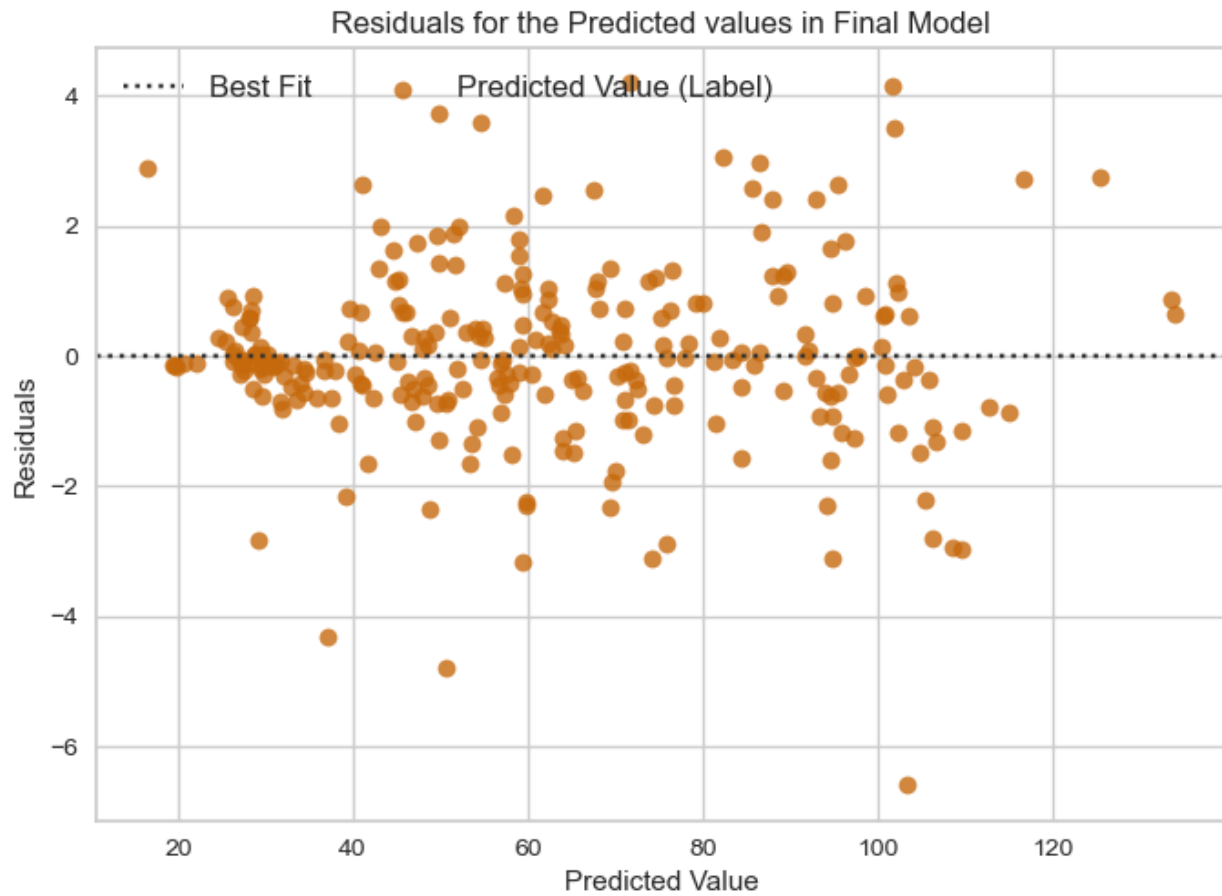
234
Extend Date index for predicted months
Cast predictions to the dataframe
Plot predictions

238

WTI Price Prediction

239

WTI Price Prediction
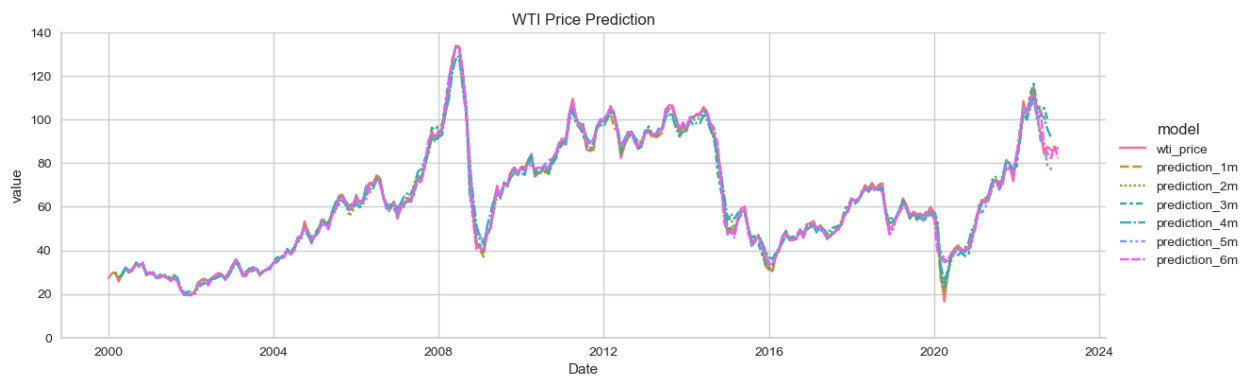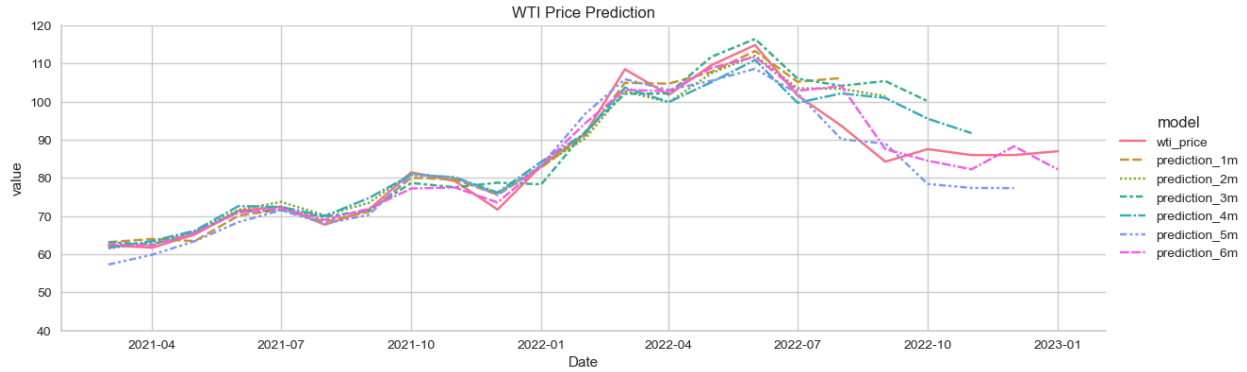
240
241
242
243   ## Calculate Expected Average
244   #Make a copy to store average predictions
245   df_average = dates_extend_df.copy()
246
247
248   ### Calculate weights
249   #Calcluate weight as 1/RMSE
250   weights = power(rmse_all, -1)
251
252   #Calclate weighted mean of predictions
253   df_average['avg_prediction'] = (df_average.iloc[:,3:]*weights_df).sum(axis=1)
254
255
256   ### Calcalte average RMSE and upper and lower bounds
257   2.5758 is used because the 99% confidence interval has only 1% on each side. The probability
258   for a z score below −2.5758 is 1%, and similarly for a z score above +1.96; added together this is
259   1%.
260
261   #Calcluate average RMSE of multiple models
262   count_rmse = dates_extend_df.notnull().iloc[:,3:].sum(axis=1)
263   df_average['avg_rmse'] = np.sqrt(sum(power(rmse_all,2))/count_rmse)
264
265

24

WTI Price Prediction

266
267
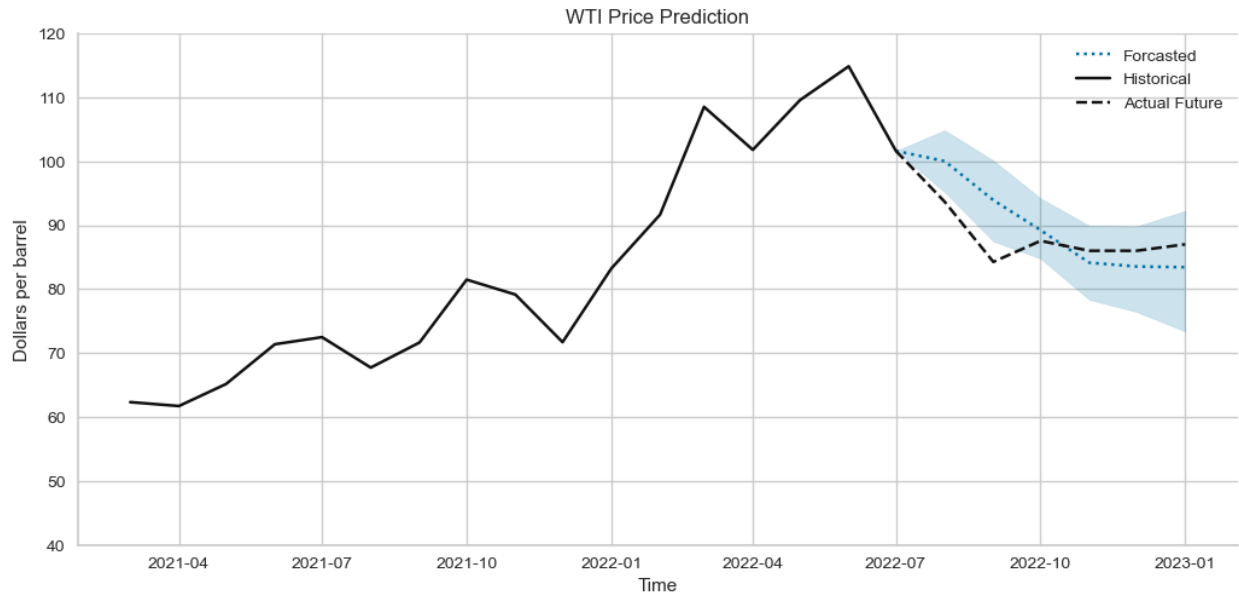
3. Results and Discussion

268

269

270

271

272

4. Recommendations and Future Work

273

274

275 • Fused TIR and RGB 3D models generated from UAS imagery offer great potential

276 for mapping heat loss, supplementing non-destructive testing of structures, aiding

277 in the inspection of electrical parts, and more.

278 • This study tested a simplified approach for generating 3D TIR point clouds from

279 coacquired TIR and RGB images for remote sensing applications. The constructed

280 TIR point clouds are georeferenced to the same coordinate system as the RGB

25

clouds. The resultant point cloud preserves the spatial density and resolution of the RGB point cloud while adding TIR attributes.

- The integrated visualization approach tested in this study enables 3D point cloud and 2D raster representation of RGB and TIR data in one model, enhancing the visual interpretation and analysis of the remotely-sensed data.

- The approach does not require additional depth sensors, such as lidar, or GNSS-aided INS for registration purposes.

- In general, the approach is appropriate for cases when….. For evaluation, and as examples of implementation…. While the SfM processing of RGB images was able to generate reliable….

- In future work, the proposed integration and visualization can be integrated into standard …... Radiometric calibration was considered beyond the scope of the present study; however, in-situ radiometric calibration of the thermal camera might improve the spectral content of the data. As an alternative …...

- TIR-RGB image feature matching and auto-registration can handle non-synchronized dual-head camera captures; however, extraction of identical features and co-registration based on the extracted pair is challenging for images of different spectral bands at the scene without well-designed calibration patterns.

- It is recommended that follow-on studies be conducted to address these topics…..

Acknowledgments

References

308 Ardabili, Sina, Amir Mosavi, and Annamária R. Várkonyi-Kóczy. 2020. "Advances in Machine
309   Learning Modeling Reviewing Hybrid and Ensemble Methods." In , 215–227.
310   doi:10.1007/978-3-030-36841-8_21.

311 Basiri, Mohammad Hossein; Javadnejad, Farshid; Saeedi, Azita. 2015. "Forecasting Crude Oil
312   Price with an Artificial Neural Network Model Based on a Regular Pattern for Selecting of
313   the Training and Testing Sets Using Dynamic Command-Line Functions." In *24th*
314   *International Mining Congress and Exhibition of Turkey-IMCET'15*.

315 Bilogur, Aleksey. 2018. "Missingno: A Missing Data Visualization Suite." *The Journal of Open*
316   *Source Software* 3 (22). The Open Journal: 547. doi:10.21105/JOSS.00547.

317 Deng, Chao, Liang Ma, and Taishan Zeng. 2021. "Crude Oil Price Forecast Based on Deep
318   Transfer Learning: Shanghai Crude Oil as an Example." *Sustainability* 13 (24): 13770.
319   doi:10.3390/su132413770.

320 EIA. 2022a. "Opendata - U.S. Energy Information Administration (EIA)."
321   https://www.eia.gov/opendata/.

322 EIA. 2022b. "International Energy Statistics, Total Oil Production." *U.S. Energy Information*
323   *Administration*.

324 FRED. 2022. "St. Louis Fed Web Services: FRED® API."
325   https://fred.stlouisfed.org/docs/api/fred/.

326 Hamilton, James. 2008. *Understanding Crude Oil Prices*. Cambridge, MA. doi:10.3386/w14492.

327   Hamilton, James. 2009. *Causes and Consequences of the Oil Shock of 2007-08*. Cambridge, MA.

328   doi:10.3386/w15002.

329   Hou, Aijun, and Sandy Suardi. 2012. "A Nonparametric GARCH Model of Crude Oil Price Return

330   Volatility." *Energy Economics* 34 (2): 618–626. doi:10.1016/j.eneco.2011.08.004.

331   Investing. 2022. "Investing.Com - Stock Market Quotes & Financial News."

332   https://www.investing.com/.

333   Jammazi, Rania, and Chaker Aloui. 2012. "Crude Oil Price Forecasting: Experimental Evidence

334   from Wavelet Decomposition and Neural Network Modeling." *Energy Economics* 34 (3):

335   828–841. doi:10.1016/j.eneco.2011.07.018.

336   Javadnejad, Farshid. 2012. "Presenting a Model for Prediction of Crude Oil Price Based on

337   Artificial Intelligent Hybrid Methods and Time-Series." Tarbiat Modares University.

338   Kilian, Lutz, and Daniel P. Murphy. 2014. "THE ROLE OF INVENTORIES AND

339   SPECULATIVE TRADING IN THE GLOBAL MARKET FOR CRUDE OIL." *Journal of*

340   *Applied Econometrics* 29 (3): 454–478. doi:10.1002/jae.2322.

341   Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias

342   Bussonnier, Jonathan Frederic, Kyle Kelley, et al. 2016. *Jupyter Notebooks: A Publishing*

343   *Format for Reproducible Computational Workflows*. Vol. 2016. https://jupyter.org/.

344   Lanza, Alessandro, Matteo Manera, and Massimo Giovannini. 2005. "Modeling and Forecasting

345   Cointegrated Relationships among Heavy Oil and Product Prices." *Energy Economics* 27 (6):

346   831–848. doi:10.1016/j.eneco.2005.07.001.

347   Lu, Quanying, Shaolong Sun, Hongbo Duan, and Shouyang Wang. 2021. "Analysis and

348   Forecasting of Crude Oil Price Based on the Variable Selection-LSTM Integrated Model."

349   *Energy Informatics* 4 (S2): 47. doi:10.1186/s42162-021-00166-4.

350   McKinney, Wes. 2011. "Pandas: A Foundational Python Library for Data Analysis and Statistics."

351   *Python for High Performance and Scientific Computing* 14 (9). Seattle: 1–9.

352   https://pandas.pydata.org/.

353   Moez, Ali. 2022. "PyCaret: An Open Source, Low-Code Machine Learning Library in Python."

354   Murat, Atilim, and Ekin Tokat. 2009. "Forecasting Oil Price Movements with Crack Spread

355   Futures." *Energy Economics* 31 (1): 85–90. doi:10.1016/j.eneco.2008.07.008.

356   Python Software Foundation. 2022. "Python." Python Software Foundation (PSF).

357   https://www.python.org/.

358   Wang, Yudong, Chongfeng Wu, and Li Yang. 2015. "Forecasting the Real Prices of Crude Oil: A

359   Dynamic Model Averaging Approach." *SSRN Electronic Journal*. doi:10.2139/ssrn.2590195.

360   WSJ. 2022. "The Wall Street Journal - Breaking News, Business, Financial & Economic News,

361   World News and Video." https://www.wsj.com/.

362   Yu, Lean, Yang Zhao, and Ling Tang. 2017. "Ensemble Forecasting for Complex Time Series

363   Using Sparse Representation and Neural Networks." *Journal of Forecasting* 36 (2): 122–138.

364   doi:10.1002/for.2418.

365   Zhao, Yang, Jianping Li, and Lean Yu. 2017. "A Deep Learning Ensemble Approach for Crude

366   Oil   Price   Forecasting."   *Energy   Economics*   66   (August):   9–16.

367   doi:10.1016/j.eneco.2017.05.023.

368