Natural Gas Price Prediction System using Machine Learning

SMARTBRIDGE EXTERNSHIP- ARTIFICIAL INTELLIGENGE TRACK

INDEX

INTRODUCTION

Natural gas, which is one of the most important energy resources, is going to play an expanded role in the future of global energy due to its significant environmental benefits. Forecasting natural gas prices is a powerful and essential tool which has become more important for different stakeholders in the natural gas market, allowing them to make better decisions for managing the potential risk, reducing the gap between the demand and supply, and optimizing the usage of resources based on accurate predictions. As indicated in the abovementioned existing studies that exploited machine learning tools for natural gas price prediction, ANN is widely used machine learning methods in forecasting natural gas prices.

PROBLEM STATEMENT AND SOLUTION

The goal of this project is to devise a machine learning algorithm for accurate prediction of natural gas prices.

Steps to be followed:

- Collection of data to form the dataset
- Data preprocessing- preparing the data for machine learning algorithm
- Data visualization- understanding the factors on which the data depends heavily
- Splitting the dataset into train and test set
- Devising a sequential type regression model to predict the values
- Training of the model
- Deployment of the model
- Building a Flask web application for users to easily interact with the trained model

LITERATURE SURVEY

A study on the prediction of prices was read and the following observations were made from the research paper.

The poor track record of energy price forecasting models has encouraged analysts to turn to other sources of information about future energy prices, including most prominently, energy futures markets. Energy futures markets are ‘hubs’ that price and market natural gas. On examining several years of spot prices one finds that, in general, gas futures are unbiased predictors of future spot prices whereas the author finds bias in natural gas futures prices where futures prices are greater than realized spot prices. Chinn et al (2005) finds futures prices to be unbiased predictors of future spot prices, with the exception of those in the natural gas market at the three-month horizon and they slightly outperform time series models. This study builds upon the existing literature by investigating the accuracy of forecast methods up to the 24-month horizon.

Based on the before-mentioned machine learning methods, prepared data, forecasting performance evaluation criteria, model validation technique, and selected model parameters, the empirical study is carried out. Observing data of four criteria can easily find that the forecasting performance of ANN and SVM is better than that of GBM and GPR. In particular, ANN is obviously superior to other methods while GBM has the worst behaviour. Overall, the performance ranking is ANN, SVM, GPR, and GBM from strong to weak. intuitively compares the prediction suitcases of four machine learning methods, which contains 214 observed values from January 2001 to October 2018. From the view of the whole tendency, ANN outperforms others, in particular, for the prediction of abnormal values at the beginning of 2009 and the second half of 2010. SVM and GPR are inferior to ANN, but SVM excels the other three methods in terms of price prediction in the middle of 2008. GBM behaves the worst prediction ability among the methods, especially for outliers.

This study motivated us to believe that ANN is one of the most accurate methods of Natural Gas Price Prediction.

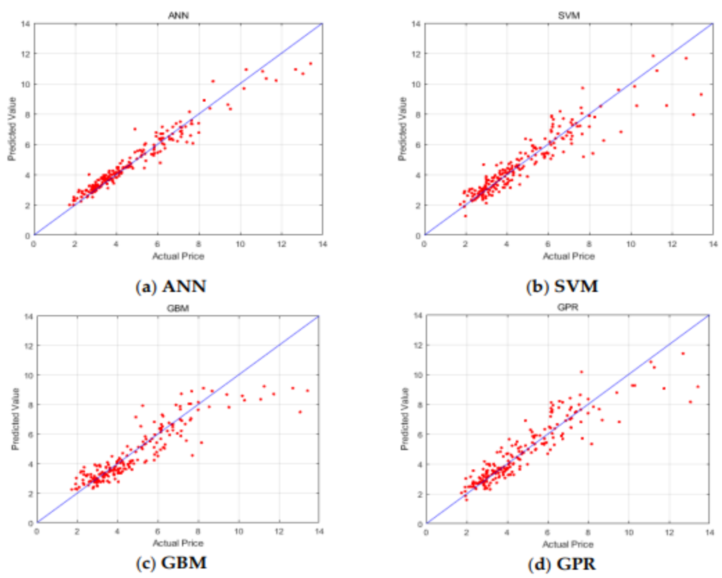


Figure 5. Distribution suitcases between predictive values and actual values.

SOFTWARE SPECIFICATIONS

- Jupyter Notebook- open-source web application that you to create and share documents that contain live code, equations, visualizations and narrative text.
- Spyder- web API using the Flask framework

DATASET

	A	B	C
1	Date	Price	
2	07-01-1997	3.82	
3	08-01-1997	3.8	
4	09-01-1997	3.61	
5	10-01-1997	3.92	
6	13-01-1997	4	
7	14-01-1997	4.01	
8	15-01-1997	4.34	
9	16-01-1997	4.71	
10	17-01-1997	3.91	
11	20-01-1997	3.26	
12	21-01-1997	2.99	
13	22-01-1997	3.05	
14	23-01-1997	2.96	
15	24-01-1997	2.62	
16	27-01-1997	2.98	
17	28-01-1997	3.05	
18	29-01-1997	2.91	
19	30-01-1997	2.86	
20	31-01-1997	2.77	
21	03-02-1997	2.49	
22	04-02-1997	2.59	
23	05-02-1997	2.65	
24	06-02-1997	2.51	
25	07-02-1997	2.39	
26	10-02-1997	2.42	
27	11-02-1997	2.34	
28	12-02-1997	2.42	
29	13-02-1997	2.22	
30	14-02-1997	2.12	
31	18-02-1997	1.84	
32	19-02-1997	1.95	

Date Column-

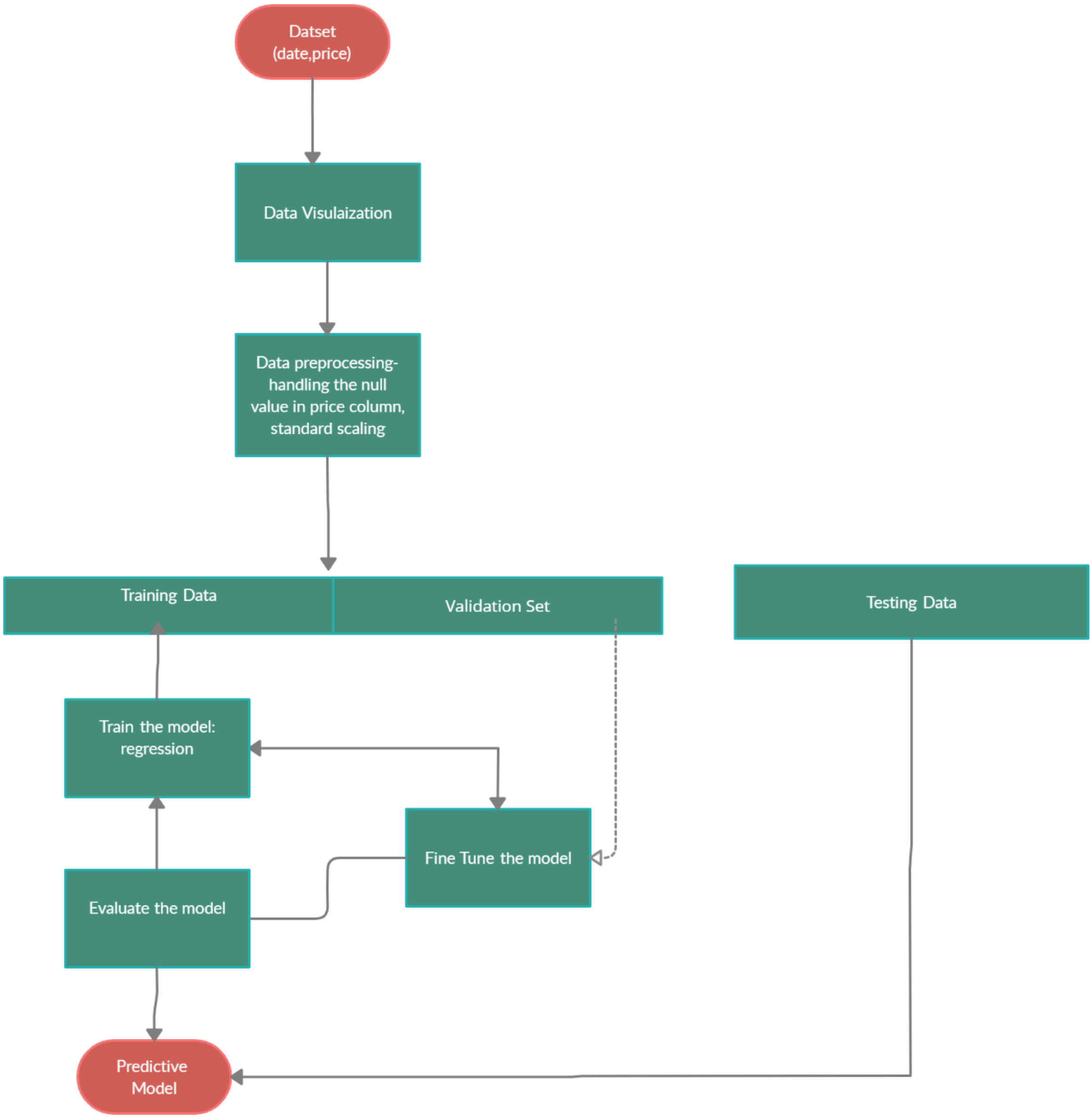
- Day
- Month
- Year

Price Column-

Corresponding price of natural day for a particular date

Data available from 1997 to 2020

FLOWCHART OF THE MODEL



MODEL

1. DATA HANDLING

In [4]:
```
#splitting the year,month and day into seperate columns
dataset['Year'] = pd.DatetimeIndex(dataset['Date']).year
dataset['Month'] = pd.DatetimeIndex(dataset['Date']).month
dataset['Day'] = pd.DatetimeIndex(dataset['Date']).day
```

In [5]:
```
dataset.head()
```

Out[5]:

	Date	Price	Year	Month	Day
0	07-01-1997	3.82	1997	7	1
1	08-01-1997	3.80	1997	8	1
2	09-01-1997	3.61	1997	9	1
3	10-01-1997	3.92	1997	10	1
4	13-01-1997	4.00	1997	1	13

In [6]:
```
dataset.drop('Date',axis=1,inplace=True) # deleting the date column as it is unnecessary
```

In [7]:
```
dataset.head()
```

Out[7]:

	Price	Year	Month	Day
0	3.82	1997	7	1
1	3.80	1997	8	1
2	3.61	1997	9	1
3	3.92	1997	10	1
4	4.00	1997	1	13

In [8]:
```
dataset.isnull().any()
```

Out[8]:
```
Price     True
Year     False
Month    False
Day      False
dtype: bool
```

In [9]:
```
dataset.isnull().sum()
```

Out[9]:
```
Price    1
Year     0
Month    0
Day      0
dtype: int64
```

In [10]:
```
#handling of null values
dataset['Price'].fillna(dataset['Price'].mean(),inplace=True)
```

2. BUILDING THE MODEL

Creating the model for price prediction-regression

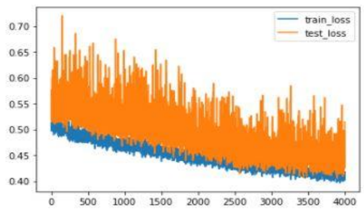
```
In [19]: from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Dense, Dropout

In [20]: reg=Sequential()

In [21]: reg.add(Dense(units=3,kernel_initializer="random_uniform",activation="relu"))

In [22]: reg.add(Dense(units=100,kernel_initializer="random_uniform",activation="relu"))

In [23]: reg.add(Dense(units=100,kernel_initializer="random_uniform",activation="relu"))

In [24]: reg.add(Dense(units=1,kernel_initializer="random_uniform"))

In [25]: reg.compile(optimizer="rmsprop",loss="mse",metrics=['mse'])

In [44]: history= reg.fit(x_train,y_train,batch_size=200,epochs=4000,validation_data=(x_test,y_test))
```

```
24/24 [==============================] - 0s 7ms/step - loss: 0.4978 - mse: 0.4978 - val_loss: 0.5590 - val_mse: 0.5590
Epoch 2/4000
24/24 [==============================] - 0s 5ms/step - loss: 0.5080 - mse: 0.5080 - val_loss: 0.5768 - val_mse: 0.5768
Epoch 3/4000
24/24 [==============================] - 0s 5ms/step - loss: 0.5050 - mse: 0.5050 - val_loss: 0.5333 - val_mse: 0.5333
Epoch 4/4000
24/24 [==============================] - 0s 5ms/step - loss: 0.5083 - mse: 0.5083 - val_loss: 0.5169 - val_mse: 0.5169
Epoch 5/4000
24/24 [==============================] - 0s 5ms/step - loss: 0.5081 - mse: 0.5081 - val_loss: 0.5469 - val_mse: 0.5469
Epoch 6/4000
24/24 [==============================] - 0s 5ms/step - loss: 0.5133 - mse: 0.5133 - val_loss: 0.5664 - val_mse: 0.5664
Epoch 7/4000
24/24 [==============================] - 0s 5ms/step - loss: 0.5061 - mse: 0.5061 - val_loss: 0.5389 - val_mse: 0.5389
Epoch 8/4000
24/24 [==============================] - 0s 6ms/step - loss: 0.4976 - mse: 0.4976 - val_loss: 0.5517 - val_mse: 0.5517
Epoch 9/4000
24/24 [==============================] - 0s 5ms/step - loss: 0.5134 - mse: 0.5134 - val_loss: 0.5466 - val_mse: 0.5466
Epoch 10/4000
24/24 [==============================] - 0s 5ms/step - loss: 0.5000 - mse: 0.5000 - val_loss: 0.5531 - val_mse: 0.5531
Epoch 11/4000
```

```
In [45]: y_pred=reg.predict(x_test)
```

3. VERIFYING THE MODEL

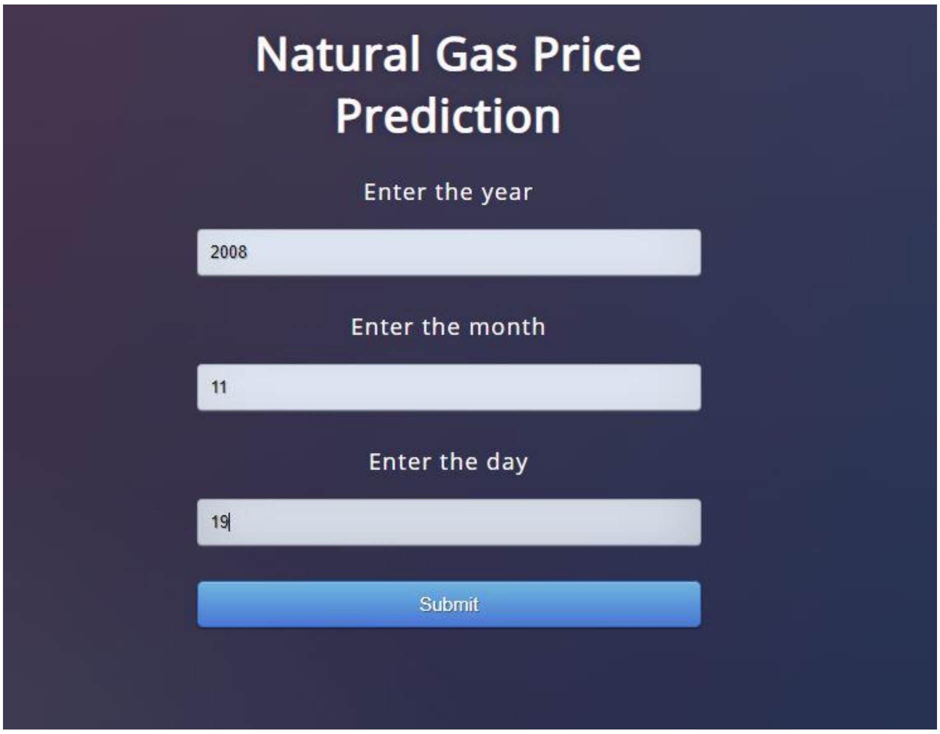
```
In [46]: from sklearn.metrics import r2_score
         accuracy=r2_score(y_test,y_pred)
         accuracy
Out[46]: 0.9029401956732639

In [54]: import matplotlib.pyplot as plt
         plt.plot(history.history['loss'],label="train_loss")
         plt.plot(history.history['val_loss'],label='test_loss')
         plt.legend()
         plt.show()
```

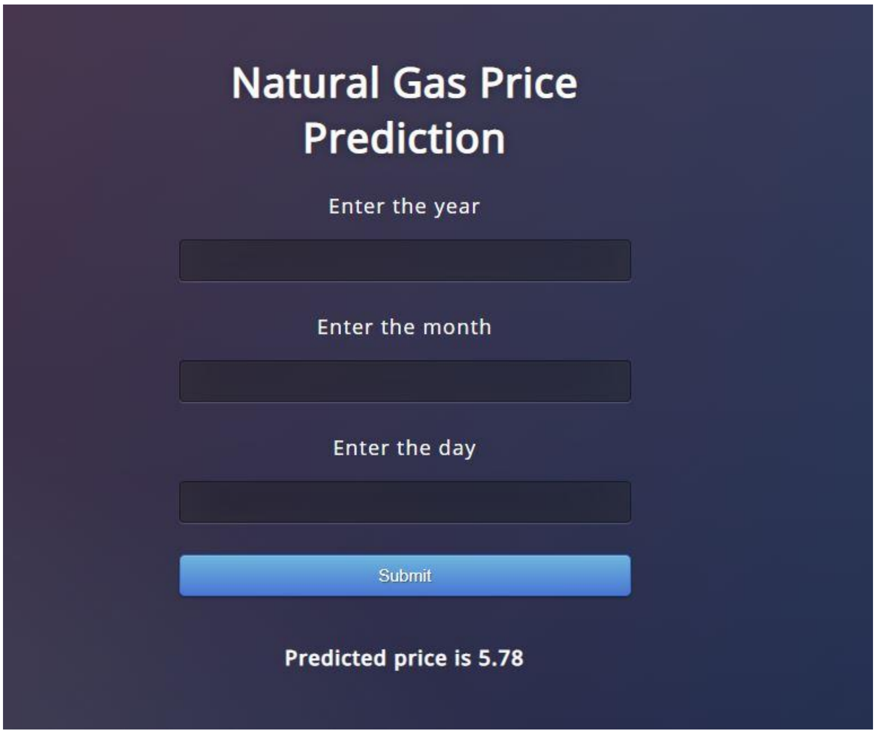


Verifying the model

```
In [ ]:  yp=reg.predict(sc.transform([[2008,7,30]]))

In [49]: yp
Out[49]: array([[8.949772]], dtype=float32)

In [50]: yp1=reg.predict(sc.transform([[2011,7,19]]))

In [51]: yp1
Out[51]: array([[3.9736688]], dtype=float32)

In [63]: reg.save("price.h5")

In [64]: import joblib
         joblib.dump(sc,"scaler")
Out[64]: ['scaler']
```

FLASK APPLICATION OUTPUT





CONCLUSION

Accuracy of the model- 90.29%

The model was successfully deployed by all the team members with the valuable guidance of the mentor under SmartBridge**.** Models such as ANN to determine which approach, those based on economic models or on the market, is more accurate and unbiased

FUTURE SCOPE

- For future work, the effect of emerging machine learning algorithms can be evaluated, such as deep learning and reinforcement learning, on energy price and correlation prediction.
- The next step could be to determine how these results can inform researchers and policy makers who utilize natural gas prices to develop federal energy policy.
- Additionally, more factors on which the natural gas price depend like availability based on location could be included in the dataset.

BIBLIOGRAPHY

1. International Energy Agency (IEA). Key World Energy Statistics 2018. Available online: https://webstore.iea. org/key-world-energy-statistics-2018 (accessed on 30 April 2019).

2. Afgan, N.H.; Pilavachi, P.A.; Carvalho, M.G. Multi-criteria evaluation of natural gas resources. Energy Policy 2007, 35, 704–713. [CrossRef]

3. Abrishami, H.; Varahrami, V. Different methods for gas price forecasting. Cuad. Econ. 2011, 34, 137–144. [CrossRef] Energies 2019, 12, 1680 15 of 17

4. Busse, S.; Helmholz, P.; Weinmann, M. Forecasting day ahead spot price movements of natural gas— An analysis of potential influence factors on basis of a NARX neural network. In Proceedings of the Tagungsband der Multikonferenz Wirtschaftsinformatik (MKWI), Braunschweig, Germany; Available online: https://publikationsserver.tu-braunschweig.de/servlets/MCRFileNodeServlet/dbbs_derivate_00027726/ Beitrag299.pdf (accessed on 30 April 2019).

5. Azadeh, A.; Sheikhalishahi, M.; Shahmiri, S. A Hybrid Neuro-Fuzzy Approach for Improvement of Natural Gas Price Forecasting in Vague and Noisy Environments: Domestic and Industrial Sectors.