A CENTER FOR INTER-DISCIPLINARY RESEARCH
2020-21

TITLE

PREDICTION OF INTERNATIONAL NATURAL GAS PRICES

SUPERVISED BY

CHAITRA

GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
AUTONOMOUS

Advanced Academic Center

(A Center For Inter-Disciplinary Research)

This is to certify that the project titled

"PREDICTION OF INTRNATIONAL NATURAL GAS PRICES "

is a bonafide work carried out by the following students in partial fulfilment of the requirements for Advanced Academic Center intern, submitted to the chair, AAC during the academic year 2020-21.

NAME	ROLL NO.	BRANCH
BENDI VANITHA	21241A04E3	ECE
GULLAPANENI VAMSI	21241A0586	CSE
RAO NISHITHA	21241A05J5	CSE
CHAVA SAI SREE PRANEETH	21241A6617	CSE-AIML
SAMALA UDAYENDRA	21241A05Q7	CSE
PRADYUMNA	21241A04H0	ECE

This work was not submitted or published earlier for any study

Dr/Ms./Mr.

Project Supervisor

R.K.Reddy
Program Coordinator

Dr.Ramamurthy Suri
Associate Dean,AAC

ACKNOWLEDGEMENTS

We express our deep sense of gratitude to our respected Director, Gokaraju Rangaraju Institute of Engineering and Technology, for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we extend our appreciation to our respected Principal, for permitting us to carry out this project.

We are thankful to the Associate Dean, Advanced Academic Centre, for providing us an appropriate environment required for the project completion.

We are grateful to our project supervisor who spared valuable time to influence us with their novel insights.

We are indebted to all the above mentioned people without whom we would not have concluded the project.

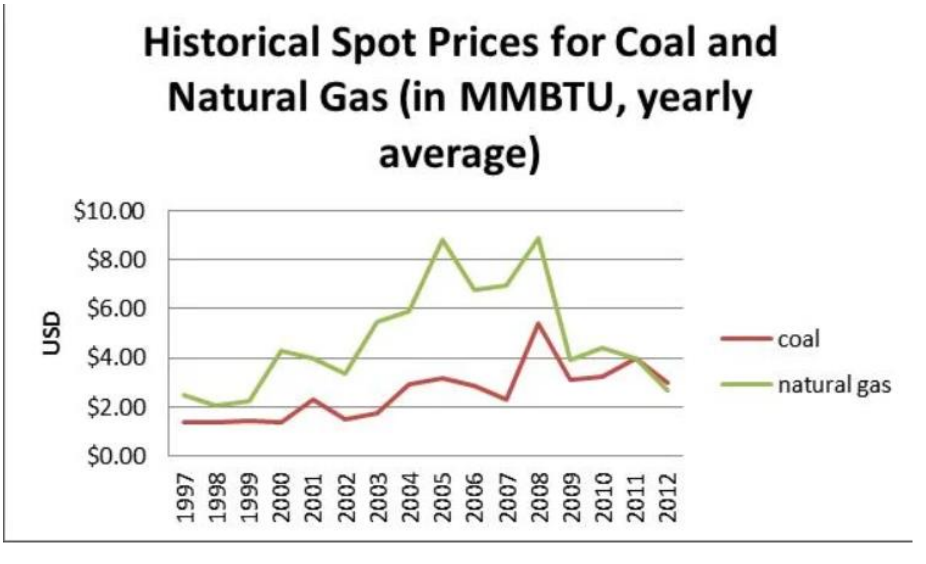
TITLE **PAGE NO**

Abstract:

The natural gas price forecast today, and the natural gas price forecast for the next six months are expected to be very different as market issues play themselves out. Then, the gas price predictions next 5 years will also be interesting to watch because of the way in which the resource is being produced and handled.

Weather, demographics, economic growth, price increases and poverty, fuel consumption, storage, and exports all affect the future gas prices predictions and how the price will move. Some of these factors are quite obvious — such as colder weather meaning more natural gas usage for heating. Temperature also comes into play when looking at the demographics.

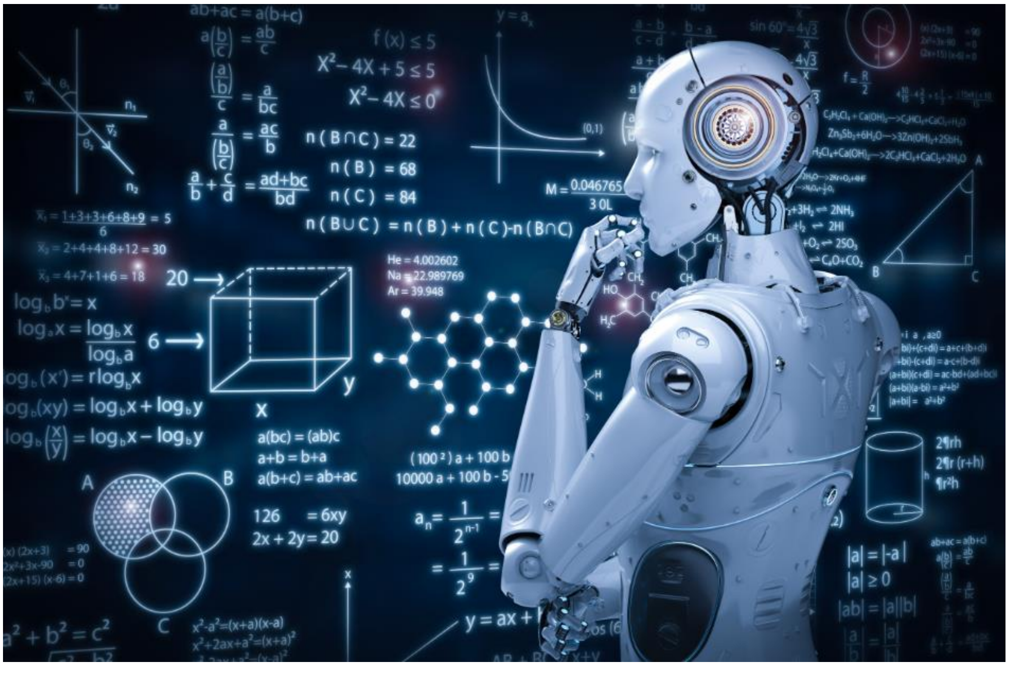
Due to all the above features the natural gas price varies. So a machine learning algorithm is useful to predict prices.



Introduction:

Machine Learning is a system of computer algorithms that can learn from example through self-improvement without being explicitly coded by a programmer. Machine learning is a part of artificial Intelligence which combines data with statistical tools to predict an output which can be used to make actionable insights.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.



Natural Gas:

Natural gas is a fossil energy source that formed deep beneath the earth's surface. Natural gas contains many different compounds. The largest component of natural gas is methane, a compound with one carbon atom and four hydrogen atoms (CH_4). Natural gas also contains smaller amounts of natural gas liquids and nonhydrocarbon gases, such as carbon dioxide and water vapor. We use natural gas as a fuel and to make materials and chemicals.

Three major supply-side factors affect prices:
•Amount of natural gas production
•Level of natural gas in storage
•Volumes of natural gas imports and exports.

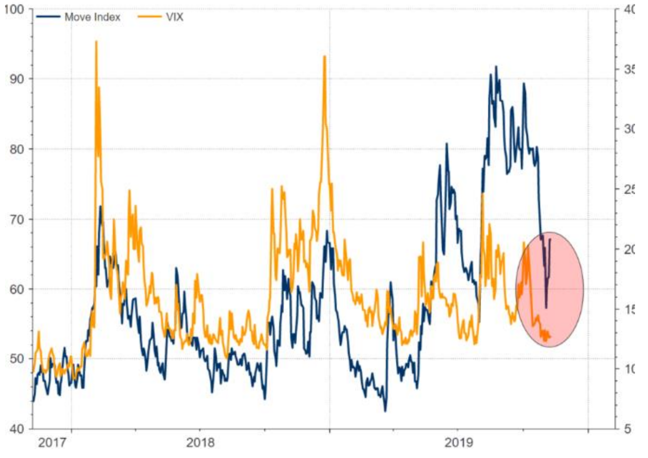
Three major demand-side factors affect prices:
•Variations in winter and summer weather
•Level of economic growth
•Availability and prices of other fuels

Why machine learning to predict natural gas prices?

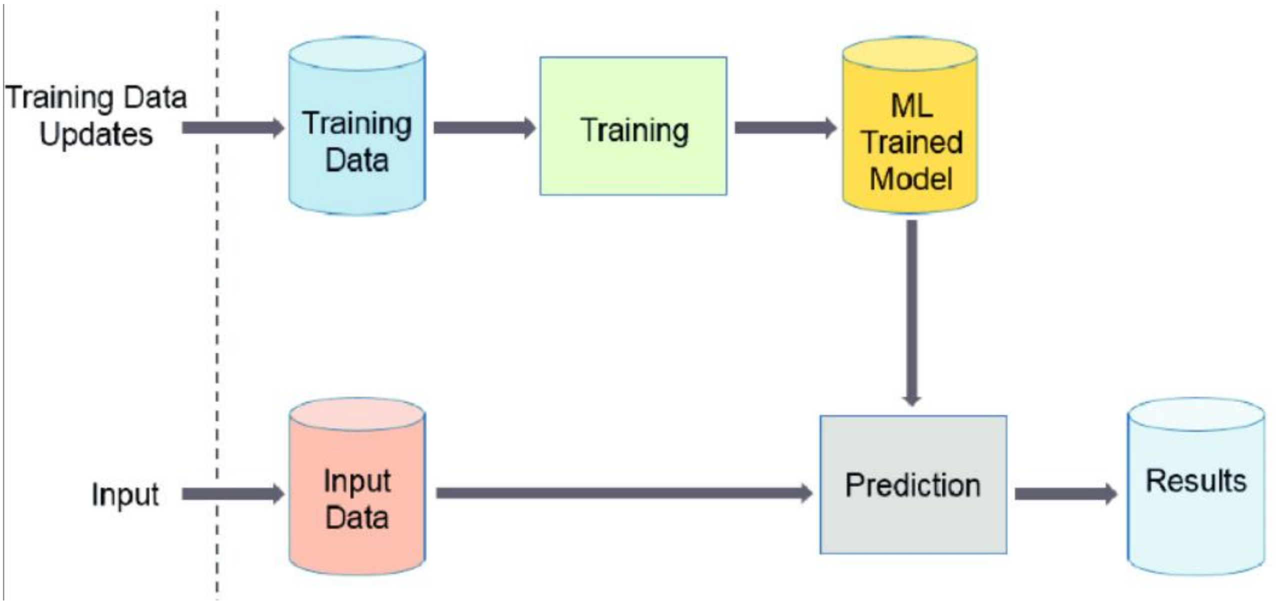
It can be observed that machine learning methodologies produce higher prediction accuracy compared to standard econometric methods . In this project advanced time series techniques are used to predict the natural gas prices. Time series techniques basically focuses on what will be its prediction at particular date in the future . This is exactly what we want, price of natural gas at particular year. So, we use machine learning algorithm here to predict natural gas prices.

Time series techniques in Machine Learning:

Time series analysis is a statistical analysis that deals with trend analysis. Time series analysis is done using a time series data that spans across a period of time. In summary, it involves looking for the correlation between your dependent variable and time.
Facebook Prophet algorithm is an algorithm designed by facebook which is an open source time series forecasting algorithm. It builds a model by finding the best smooth line.



WORK FLOW:



DATASET:

The aim of the machine learning model is to predict international natural gas prices after training on the prices dataset. We require the model to take the data which has dates and price columns and give output.

Dataset contains Natural gas prices from 1997 to 2022. The independent variable is year. The dependent variable is factor to be predicted is natural gas prices.

Code:

```
import numpy as np
```

```
!pip install pystan~=2.14
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pystan~=2.14
  Downloading pystan-2.19.1.1-cp37-cp37m-manylinux1_x86_64.whl (67.3 MB)
     |████████████████████████████████| 67.3 MB 76 kB/s
Requirement already satisfied: Cython!=0.25.1,>=0.22 in /usr/local/lib/python3.7/dist-packages (from pystan~=2.14) (0.29.32)
Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.7/dist-packages (from pystan~=2.14) (1.21.6)
Installing collected packages: pystan
  Attempting uninstall: pystan
    Found existing installation: pystan 3.3.0
    Uninstalling pystan-3.3.0:
      Successfully uninstalled pystan-3.3.0
Successfully installed pystan-2.19.1.1
```

```
!pip install fbprophet
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting fbprophet
  Downloading fbprophet-0.7.1.tar.gz (64 kB)
     |████████████████████████████████| 64 kB 1.5 MB/s
Requirement already satisfied: Cython>=0.22 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (0.29.32)
Collecting cmdstanpy==0.9.5
  Downloading cmdstanpy-0.9.5-py3-none-any.whl (37 kB)
Requirement already satisfied: pystan>=2.14 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (2.19.1.1)
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (1.21.6)
Requirement already satisfied: pandas>=1.0.4 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (1.3.5)
Requirement already satisfied: matplotlib>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (3.2.2)
Requirement already satisfied: LunarCalendar>=0.0.9 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (0.0.9)
Requirement already satisfied: convertdate>=2.1.2 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (2.4.0)
Requirement already satisfied: holidays>=0.10.2 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (0.14.2)
Requirement already satisfied: setuptools-git>=1.2 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (1.2)
Requirement already satisfied: python-dateutil>=2.8.0 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (2.8.2)
Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.7/dist-packages (from fbprophet) (4.64.0)
Requirement already satisfied: pymeeus<=1,>=0.3.13 in /usr/local/lib/python3.7/dist-packages (from convertdate>=2.1.2->fbprophet) (0.5.11)
Requirement already satisfied: korean-lunar-calendar in /usr/local/lib/python3.7/dist-packages (from holidays>=0.10.2->fbprophet) (0.2.1)
Requirement already satisfied: hijri-converter in /usr/local/lib/python3.7/dist-packages (from holidays>=0.10.2->fbprophet) (2.2.4)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from LunarCalendar>=0.0.9->fbprophet) (2022.1)
Requirement already satisfied: ephem>=3.7.5.3 in /usr/local/lib/python3.7/dist-packages (from LunarCalendar>=0.0.9->fbprophet) (4.1.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.0.0->fbprophet) (1.4.4)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.0.0->fbprophet) (0.11.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.0.0->fbprophet) (3.0.9)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from kiwisolver>=1.0.1->matplotlib>=2.0.0->fbprophet) (4.1.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.8.0->fbprophet) (1.15.0)
Building wheels for collected packages: fbprophet
  Building wheel for fbprophet (setup.py) ... done
  Created wheel for fbprophet: filename=fbprophet-0.7.1-py3-none-any.whl size=6638708 sha256=ee9185bdc0eaebe30ae1c8b190e7f6884f3cf56027473a39373bfd3247866400
  Stored in directory: /root/.cache/pip/wheels/cd/a1/12/db63ff624de492fe6cccf676091a0860fdde2ffde4bc3280e2
Successfully built fbprophet
```

```
[ ] import pandas as pd
    from prophet.plot import plot_plotly, plot_components_plotly
    from fbprophet import Prophet
```

```
[ ] df = pd.read_csv("/content/sample_data/DATA SET.csv")
```

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5938 entries, 0 to 5937
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    5938 non-null   object
 1   Price   5937 non-null   float64
dtypes: float64(1), object(1)
memory usage: 92.9+ KB
```

```
df.tail()
```

	Date	Price
5933	05-08-2020	2.23
5934	06-08-2020	2.26
5935	07-08-2020	2.15
5936	10-08-2020	2.18
5937	11-08-2020	2.19

```
[ ] df.dropna(inplace=True)
    df.reset_index(drop=True,inplace=True)
```

```
df.head()
```

	Date	Price
0	07-01-1997	3.82
1	08-01-1997	3.80
2	09-01-1997	3.61
3	10-01-1997	3.92
4	13-01-1997	4.00

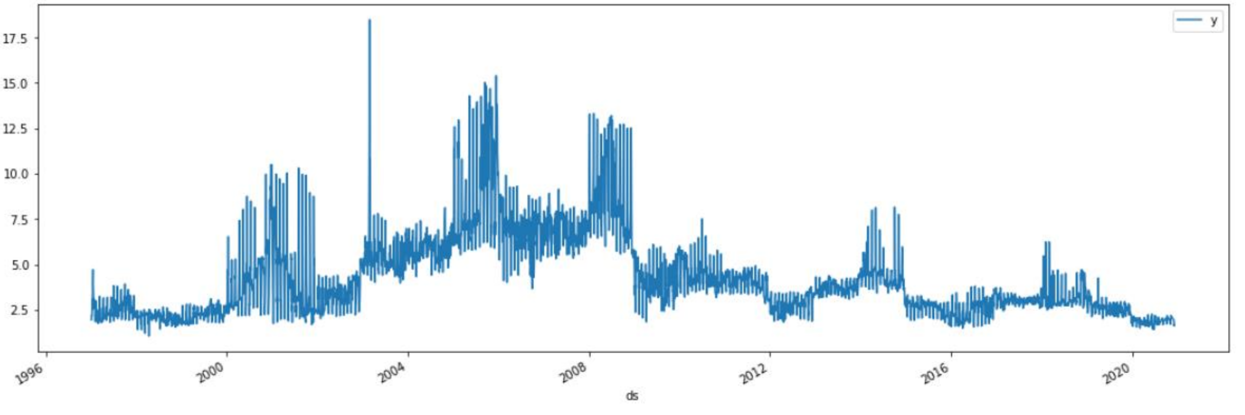
```
[ ] df.columns=["ds","y"]
```

```
df["ds"]=pd.to_datetime(df["ds"])
df.head()
```

	ds	y
0	1997-07-01	3.82
1	1997-08-01	3.80
2	1997-09-01	3.61
3	1997-10-01	3.92
4	1997-01-13	4.00

```
[ ] df.plot(x="ds",y="y",figsize=(18,6))
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fd174e3bf90>




```
[ ] len(df)
```

5937

```
[ ] len(df)
```

5937

```
[ ] train =df.iloc[:len(df)-365]
    test =df.iloc[len(df)-365:]
```

```
m=Prophet()
m.fit(train)
future= m.make_future_dataframe(periods=365)
forecast =m.predict(future)
```

```
forecast.tail()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	weekly	weekly_lower	weekly_upper	yearly	yearly_lower	yearly_upper
5932	2020-11-27	2.469824	0.742076	4.086463	1.844277	3.128230	-0.034819	-0.034819	-0.034819	-0.037182	-0.037182	-0.037182	0.002363	0.002363	0.002363
5933	2020-11-28	2.469437	0.799742	4.325616	1.841128	3.130088	0.033810	0.033810	0.033810	0.013435	0.013435	0.013435	0.020375	0.020375	0.020375
5934	2020-11-29	2.469050	0.734753	4.098386	1.838046	3.131947	0.006207	0.006207	0.006207	-0.034307	-0.034307	-0.034307	0.040514	0.040514	0.040514
5935	2020-11-30	2.468664	0.723502	4.188078	1.834963	3.133806	0.041408	0.041408	0.041408	-0.021128	-0.021128	-0.021128	0.062536	0.062536	0.062536
5936	2020-12-01	2.468277	0.876207	4.262835	1.831881	3.135664	0.120997	0.120997	0.120997	0.034849	0.034849	0.034849	0.086148	0.086148	0.086148

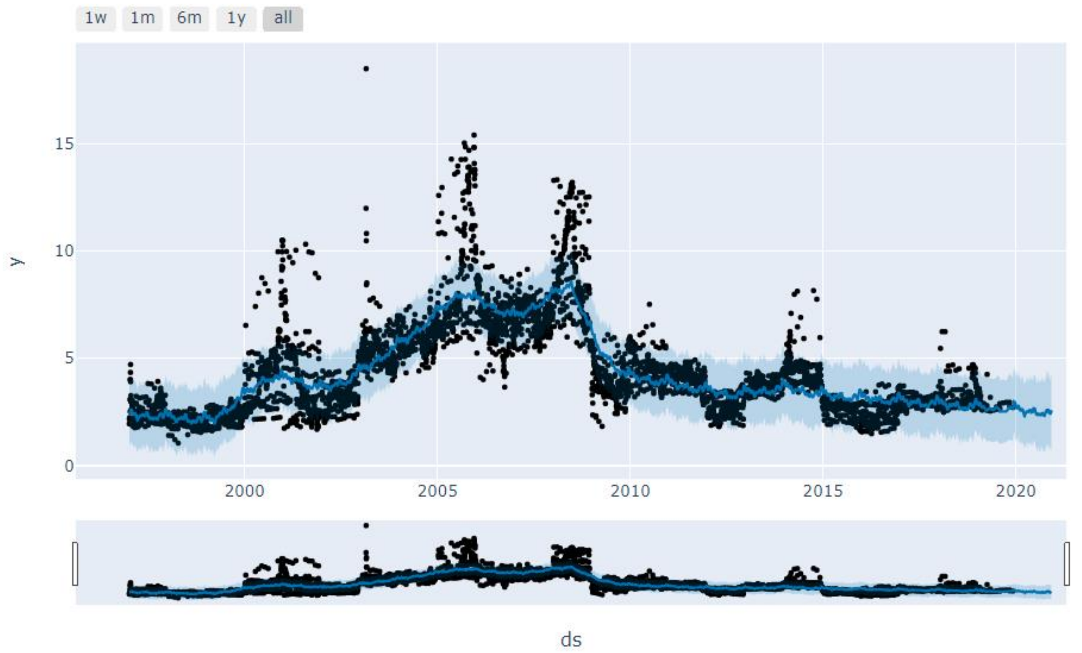
```
forecast[["ds","yhat","yhat_lower","yhat_upper"]].tail(20)
```

	ds	yhat	yhat_lower	yhat_upper
5917	2020-11-12	2.526626	0.893993	4.277876
5918	2020-11-13	2.431680	0.728712	4.071505
5919	2020-11-14	2.471149	0.764336	4.040490
5920	2020-11-15	2.413140	0.785482	4.188206
5921	2020-11-16	2.417196	0.794791	4.003377
5922	2020-11-17	2.465452	0.919994	4.157318
5923	2020-11-18	2.422870	0.792938	4.006996
5924	2020-11-19	2.466334	0.774483	4.099944
5925	2020-11-20	2.381056	0.729019	4.044705
5926	2020-11-21	2.431840	0.826170	4.110951
5927	2020-11-22	2.386679	0.811253	3.991471
5928	2020-11-23	2.404960	0.789320	4.110956
5929	2020-11-24	2.468620	0.684693	4.237493
5930	2020-11-25	2.442389	0.785191	4.210832
5931	2020-11-26	2.502879	0.771307	4.193982
5932	2020-11-27	2.435005	0.742076	4.086463
5933	2020-11-28	2.503247	0.799742	4.325616
5934	2020-11-29	2.475257	0.734753	4.098386
5935	2020-11-30	2.510071	0.723502	4.188078
5936	2020-12-01	2.589274	0.876207	4.262835

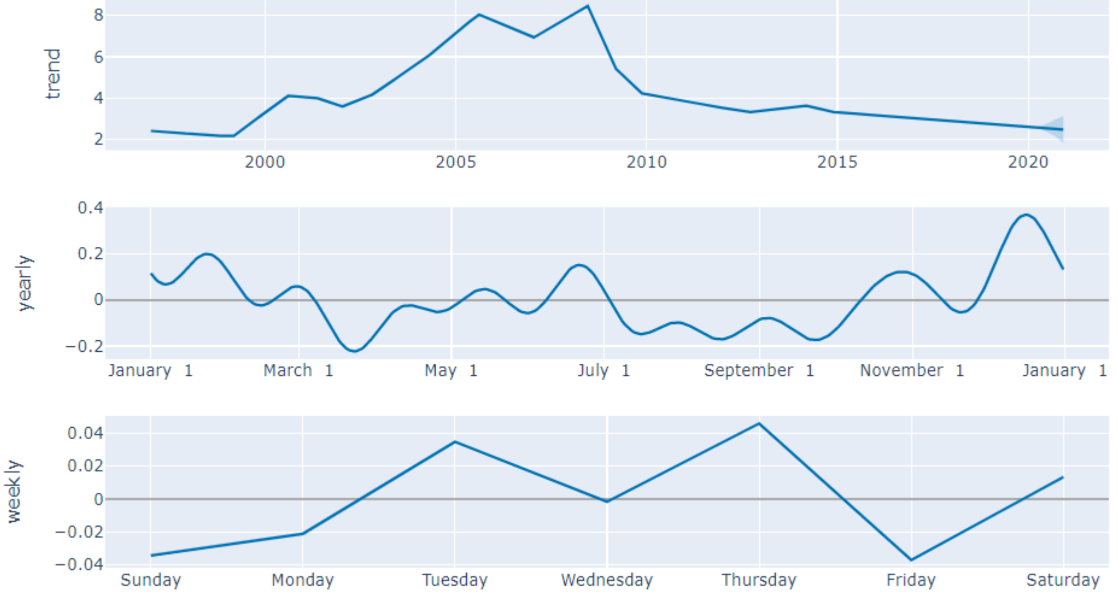
```
test.tail(20)
```

	ds	y
5917	2020-07-15	1.76
5918	2020-07-16	1.79
5919	2020-07-17	1.79
5920	2020-07-20	1.71
5921	2020-07-21	1.66
5922	2020-07-22	1.69
5923	2020-07-23	1.75
5924	2020-07-24	1.77
5925	2020-07-27	1.85
5926	2020-07-28	1.83
5927	2020-07-29	1.77
5928	2020-07-30	1.81
5929	2020-07-31	1.83
5930	2020-03-08	1.95
5931	2020-04-08	2.07
5932	2020-05-08	2.23
5933	2020-06-08	2.26
5934	2020-07-08	2.15
5935	2020-10-08	2.18
5936	2020-11-08	2.19

```
plot_plotly(m,forecast)
```




```
plot_components_plotly(m,forecast)
```



```
[ ] from statsmodels.tools.eval_measures import rmse
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:

pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
```

```
[ ] predictions = forecast.iloc[-365:]["yhat"]
```

```
print("Root Mean Squared Error between actual and predicted values:",rmse(predictions,test["y"]))
print("Mean Value of Test Dataset:",test["y"].mean())
```

```
Root Mean Squared Error between actual and predicted values: 0.46294280780893354
Mean Value of Test Dataset: 2.1910684931506847
```

```
[ ] from fbprophet.diagnostics import cross_validation
df_cv = cross_validation(m, initial='1095 days', period='180 days', horizon = '365 days')
```

```
INFO:fbprophet:Making 39 forecasts with cutoffs between 2000-03-11 00:00:00 and 2018-12-02 00:00:00
```

100% 39/39 [03:13<00:00, 7.69s/it]

```
[ ] df_cv.head()
```

	ds	yhat	yhat_lower	yhat_upper	y	cutoff
0	2000-03-13	2.938737	2.421786	3.468333	2.79	2000-03-11
1	2000-03-14	2.972873	2.415807	3.510403	2.83	2000-03-11
2	2000-03-15	2.976907	2.443938	3.544190	2.76	2000-03-11
3	2000-03-16	2.948398	2.370029	3.499662	2.84	2000-03-11
4	2000-03-17	2.966102	2.380543	3.526837	2.81	2000-03-11

```
[ ] from fbprophet.diagnostics import performance_metrics
df_p = performance_metrics(df_cv)
```

```
[ ] df_p.head()
```

	horizon	mse	rmse	mae	mape	mdape	coverage
0	36 days	2.917997	1.708215	1.239567	0.295761	0.206885	0.735849
1	37 days	2.839766	1.685161	1.243366	0.299697	0.213053	0.729957
2	38 days	2.878726	1.696681	1.251933	0.301013	0.215821	0.726415
3	39 days	2.900039	1.702950	1.252503	0.300972	0.215821	0.729895
4	40 days	2.929758	1.711654	1.255632	0.301385	0.219646	0.730647

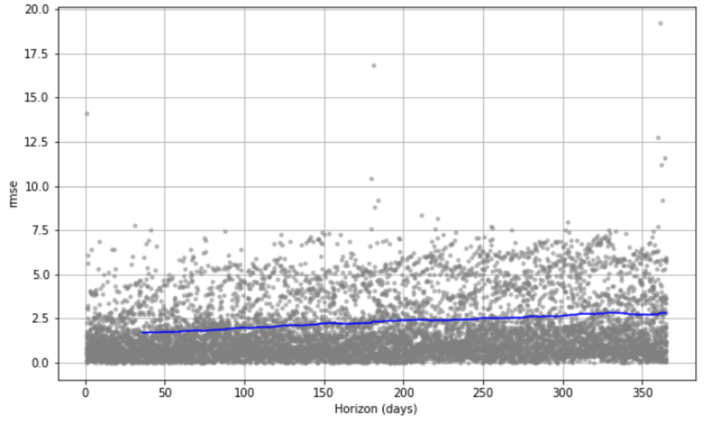
```
from fbprophet.plot import plot_cross_validation_metric
fig = plot_cross_validation_metric(df_cv, metric='rmse')
```

```
/usr/local/lib/python3.7/dist-packages/fbprophet/plot.py:526: FutureWarning:

casting timedelta64[ns] values to int64 with .astype(...) is deprecated and will raise in a future version. Use .view(...) instead.

/usr/local/lib/python3.7/dist-packages/fbprophet/plot.py:527: FutureWarning:

casting timedelta64[ns] values to int64 with .astype(...) is deprecated and will raise in a future version. Use .view(...) instead.
```



Workflow:

1.The dataset is imported using pandas and data is extracted.

2.The complete data is divided into two sets X(independent variables) and Y(dependent variables).

3.The string column(data column) is encoded to make it ready for training. data_time function is used which converts date column into actual date column.

4.prophet library is imported from fbprophet library and object is created for training.

5.The dataset is divided into training and testing datasets and training dataset is used for training the model.

6.The model is tested against testing dataset.

Input:

Input is data time column which are not only random strings but costs of natural gas prices on yearly basis.

Output:

The output displayed is, predicted annual average prices of natural gas in dollars. Root mean square error value for last one year is approximately 2.26 and mean value of test dataset is 4.17.

Future Developments:

With the world looking to source more renewable energy, and leave things like oil and coal behind, natural gas is one such source that is emerging as a good source of electricity for a number of reasons. It releases less carbon compared to other fossil fuels and the infrastructure around gas plants is much quicker and easier.
So this prediction of natural gas prices is very useful to watchout the economy.

References:

Natural Gas Prices Forecast & Predictions for 2022, 2023, 2025-2030 | PrimeXBT
Machine Learning Algorithms – Javatpoint
Facebook Prophet. (Almost) everything you should know to... | by Moto DEI | The Startup | Medium