**Nalaiya Thiran**

Batch No: B2 – 2M4E

PSG Institute of Technology and Applied Research
Department of Computer Science and Engineering

# Crude Oil Price Prediction

**Team ID:** PNT2022MID43299

**Team Members:**

| | |
|---|---|
| 715519104012 | Dharun R |
| 715519104018 | Kavin S V |
| 715519104041 | Roopakumar |
| 715519104048 | Shyam Ganesh T |

**Project Guide**:

Industry mentor: Ms. Shanthi

Faculty Mentor: Ms. P. Priya Ponnusamy

# INDEX

# CHAPTER 1: INTRODUCTION

## 1.1 Project Overview

Oil demand is inelastic, therefore the rise in price is good news for producers because they will see an increase in their revenue. Oil importers, however, will experience increased costs of purchasing oil. Because oil is the largest traded commodity, the effects are quite significant. A rising oil price can even shift economic/political power from oil importers to oil exporters. The crude oil price movements are subject to diverse influencing factors.

This Project mainly focuses on applying Neural Networks to predict the Crude Oil Price. This decision helps us to buy crude oil at the proper time. Time series analysis is the best option for this kind of prediction because we are using the Previous history of crude oil prices to predict future crude oil. So we would be implementing RNN(Recurrent Neural Network) with LSTM(Long Short Term Memory) to achieve the task.

## 1.2 Purpose

Crude oil prices are considered one of the most important indicators in the global economy. Governments and businesses spend a lot of time and energy to figure out where oil prices are headed next, but forecasting is an inexact science.

# CHAPTER 2: LITERATURE SURVEY

## 2.1 Existing Problem

**Survey 1**

**Ganiyu Adewale Busari**

**Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance**

We proposed a hybrid model that combined AdaBoost algorithm with GRU and compare its forecasting performance with the existing AdaBoost-LSTM model. The proposed work is the first that applied to the crude oil prices prediction. Comparison of the analysis results of the models are presented with the benchmarked models, that is, LSTM and GRU. The evaluation metrics considered are both scale –dependent metrics, that is, MAE and RMSE and scale-independent metrics, that is, MAPE and WMAPE. We also applied Scatter index as normalized RMSE. The proposed method, AdaBoost-GRU outperforms the single methods and AdaBoost-LSTM ensemble model in this study.

**Survey 2**

**HuiziHe,MeiSun,XiumingLi,Isaac AdjeiMensah**

**A novel crude oil price trend prediction method: Machine learning classification algorithm based on multi-modal data features**

A novel hybrid prediction model is proposed based on VMD and ML algorithms. Five multi-modal data feature indices are established based on IMFs of prices. The forecasting accuracy is elevated by introducing multi-modal data features. Trend symbols of prices are predicted by ML multi-classifiers. Classification performs better than regression in forecasting price trend.

**Survey 3**

**Prediction of crude oil prices in COVID-19 outbreak using real data**

**ÖznurÖztunç Kaymak**

The world has been undergoing a global economic recession for almost two years because of the health crisis stemming from the outbreak and its effects have still continued so far. Especially, COVID-19 reduced consumer spending due to social isolation, lockdown and travel restrictions in 2020. As a result of this, with social and economic life coming to a standstill, oil prices plummeted. With the ongoing uncertainty concerning the COVID-19 pandemic, it has been of great importance for all economic agents to predict crude oil prices. The objective of this paper is to improve a model in order to make more accurate predictions for crude oil price movements. The performance of this model is assessed in terms of some significant criteria comparing our model with its counterparts as well as artificial neural networks (ANNs) and support vector machine (SVM) methods. As for these criteria, root mean square error (RMSE) and mean absolute error (MAE) results show that this model outperforms other models in forecasting crude oil prices. Further, the simulation results for 2021 show that the daily crude oil price forecasts are almost close to the real oil prices.

**Survey 4**

**Taiyong Li,Zijie Qian,Shuheng Wang**

**Forecasting crude oil price with multilingual search engine data**

In the big data era, search engine data (SED) have presented new opportunities for improving crude oil price prediction; however, the existing research were confined to single-language (mostly English) search keywords in SED collection. To address such a language bias and grasp worldwide investor attention, this study proposes a novel multilingual SED-driven forecasting methodology from a global perspective. The proposed methodology includes three main steps: (1) multilingual index construction, based on multilingual SED; (2) relationship investigation, between the multilingual index and crude oil price; and (3) oil price prediction, with the multilingual index as an informative predictor. With WTI spot price as studying samples, the

empirical results indicate that SED have a powerful predictive power for crude oil price; nevertheless, multilingual SED statistically demonstrate better performance than single-language SED, in terms of enhancing prediction accuracy and model robustness.

**Survey 5**

**Binrong Wu,Lin Wang**

**Effective crude oil price forecasting using new text-based and big-data-driven model**

This study proposes a novel data-driven crude oil price prediction methodology using Google Trends and online media text mining. Convolutional neural network (CNN) is used to automatically extract text features from online crude oil news to illustrate the explanatory power of text features for crude oil price prediction. Specifically, our findings contribute to the methodological and theoretical insights for information processing, in that variational mode decomposition is used to construct useful time series indicators based on the outputs of CNN. Experimental results imply that the proposed text-based and online-big-data-based forecasting methods outperform other techniques. A total of 4837 and 3883 news headlines are collected in two cases, respectively. The mean absolute percentage error of the proposed model is 0.0571 and 0.0459 for crude oil price forecasting of two cases, respectively.

## 2.2 References

1. Ganiyu Adewale Busari,"Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance".
2. HuiziHe,MeiSun,XiumingLi,Isaac AdjeiMensah**,** "A novel crude oil price trend prediction method: Machine learning classification algorithm based on multi-modal data features"

3. ÖznurÖztunç Kaymak, “Prediction of crude oil prices in COVID-19 outbreak using real data”.

4. Taiyong Li,Zijie Qian,Shuheng WangForecasting crude oil price with multilingual search engine data.

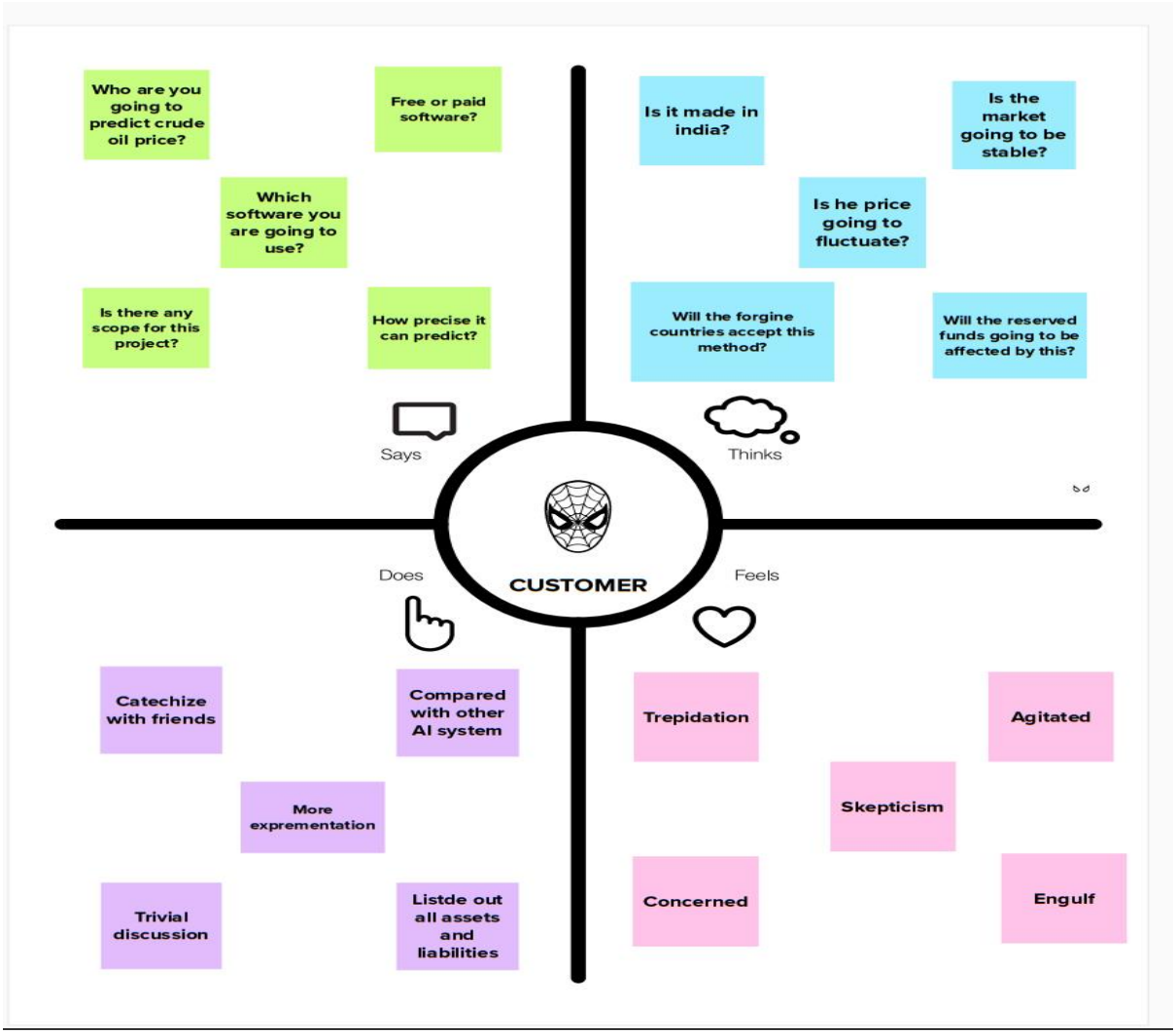
5. Binrong Wu,Lin Wang,”Effective crude oil price forecasting using new text-based and big-data-driven model”.

### 2.3 Problem Statement

Why Oil companies need to assess new fields or prospects where very little hard data exists? Based on seismic data, analysts can estimate the probability distribution of the reserve size. The oil price cannot not be figured on a day-today basis. Demand forecasts are usually made from GDP, exchange rates and domestic prices, and supply is predicted from past production data and reserve data. With rising global demand, highly volatile prices and increasingly stringent environmental regulations, the oil and gas industry faces three major challenges : reduce costs, optimize the performance of its industrial base assets. These factors include:Demand, Supply, Quality of Oil, Speculation, Demand for Oil, Temporary Price Fluctuations, Investing in Oil and Gas Drilling. High oil prices can drive job creation and investment as it becomes economically viable for oil companies to exploit higher-cost shale oil deposits. However, high oil prices also hit businesses and consumers with higher transportation and manufacturing costs. As with any commodity, stock, or bond, the laws of supply and demand cause oil prices to change. When supply exceeds demand, prices fall; the inverse is also true when demand outpaces supply. Our project is to built an efficient oil predicting system powered by AI especially to predict the oil price in the world and in the west asian countries especially gulf countries.

# CHAPTER 3: IDEATION & PROPOSED SOLUTION

## 3.1 Empathy Map Canvas




## 3.2 Ideation & Brainstorming

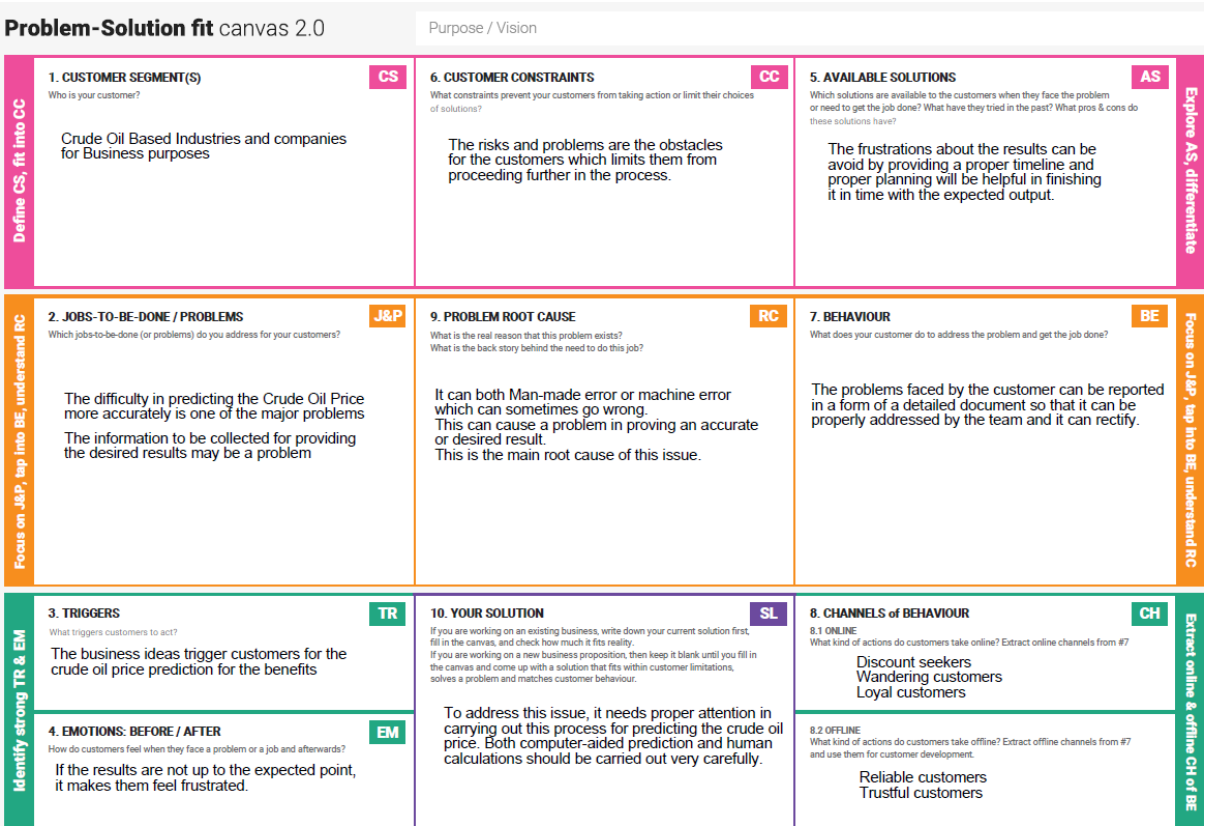
### 3.2.1 Brainstorm by team members

### 3.3 Proposed Solution

| S.NO. | PARAMETER | DESCRIPTION |
|---|---|---|
| 1. | Problem statement (Problem to be solved) | Predicting the price of the crude oil is very difficult in recent days due to war time.The oil price which is controlled by the west- Asian countries. |
| 2. | Idea/Solution description | This system is built by using the AI. By using this system we can. Predict the oil price which can be useful for countries to buy the crude oil. |
| 3. | Novelty/Uniqueness | The application will make understand people how oil price is predicted. There is no mediator between buyer and seller. |
| 4. | Social Impact/Customer Satisfaction | This system provides an effective support to both the buyer and seller & create a satisfaction between them.So the oil price can be low compared to older price. |
| 5. | Business Model (Revenue Model) | This system covers a wide range because we can predict both crude oil, petroleum products & natural gas. |
| 6. | Scalability of the solution | Implementing this system provides the performance and fulfillness within the market and creating a |

| | | |
|---|---|---|
| | | platform which can be used worldwide. |

## 3.4 Proposed Solution fit



# CHAPTER 4: REQUIRMENT ANALYSIS

## 4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Application | User Direct Open With Google Play Store App<br>User Can Download The Crude Oil Price |
| FR-2 | User Products Available | User Using The Application There Are So Many Products In Crude Oil Price App<br>User Update The Energy And Oil Price Instant The Application |
| FR-3 | User Additional Features | User Can Read Latest News And View Oil Price Charts<br>User View Major Energy Quotes<br>User Can Using A Multiple Color Themes |
| FR-4 | User Exceptions | User Can Exchange Rates And Currancy Converter |

## 4.2 Non-functional Requirements:

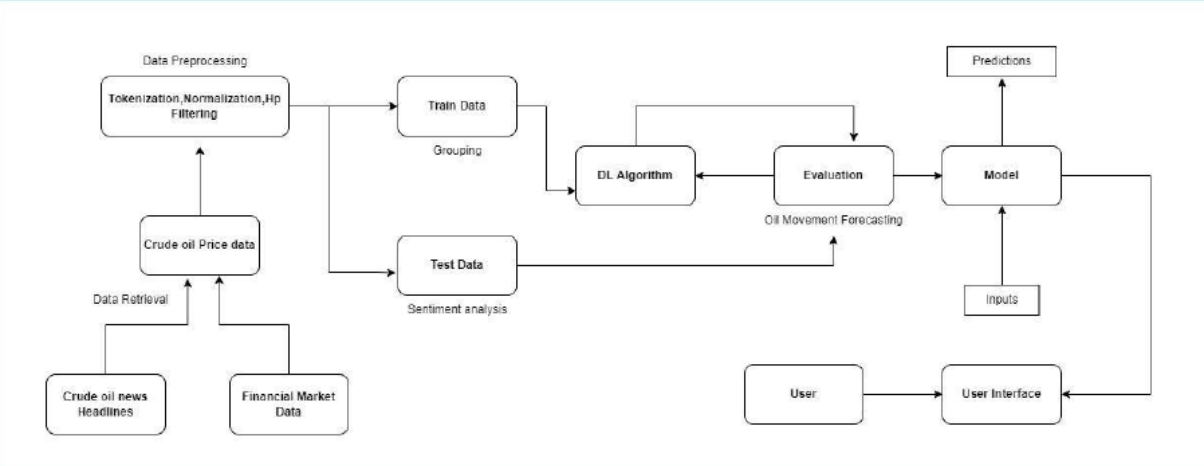
Following are the non-functional requirements of the proposed solution.

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | Used to improve to the Accuracy of crude oil price prediction |
| NFR-2 | **Security** | In the rising oil price can even shift economical/political power from oil importers to oil exporters communications will be secured |
| NFR-3 | **Reliability** | Reliability of the pointing towards high –risk components |
| NFR-4 | **Performance** | Performance of the this project is to improve to the accuracy of crude oil price prediction |
| NFR-5 | **Availability** | The Availability Solution is More Benefit for and the Importers and exporters in the crude oil price prediction. |
| NFR-6 | **Scalability** | The scalability are 90%-95% |

# CHAPTER 5: PROJECT DESIGN

## 5.1 Data Flow Diagram:

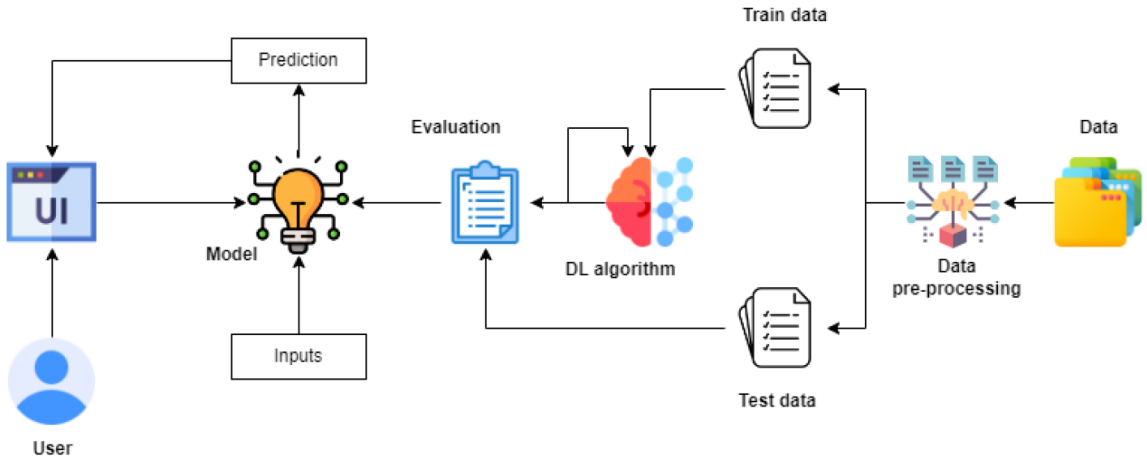
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.




## 5.2 Solution and Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.




## 5.3 User Stories

Use the below template to list all the user stories for the product.

| User Type | Functional Requirement (Epic) | User Story Number | User Story/ Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer (Mobile User) | Registration | USN-1 | As a user,I can register for the application by entering my email, password,and confirming my password. | I can access my account/ Displays Line gragh / Bar gragh. | High | Sprint-1 |
| | | USN-2 | As a user,I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
| | | USN-3 | As a user,I can register for the application through Facebook | I can register & accessthe my Account | Low | Sprint-2 |
| | | USN-4 | As a user,I can register for the application through Gmail | I can register through already logged in gmail account. | Medium | Sprint-1 |
| | Login | USN-5 | As a user,I can log into the application by entering email & password | After registration,I can log in by only email & password. | High | Sprint-1 |
| | Line\Bar gragh | | After entering the inputs,the model will display predictions in Line\Bar Gragh Format. | I can get the expected prediction in various formats. | High | Sprint-3 |
| Customer (Web user) | Login | USN-1 | As the web user,I can login simply by using Gmail or Facebook account. | Already created gmail can be used for Login. | Medium | Sprint-2 |
| Customer Care Executive | Support | | The Customer care service will provide solutions for any FAQ and also provide ChatBot. | I can solve the problems arised by Support. | Low | Sprint-3 |
| Administrator | News | | Admin will give the recent news of Oil Prices. | Provide the recent oil prices. | High | Sprint-4 |
| | Notification | | Admin will notify when the oil prices changes. | Notification by Gmail. | High | Sprint-4 |
| | Access Control | | Admin can control the access of users. | Access permission for Users. | High | Sprint-4 |
| | Database | | Admin can store the details of users. | Stores User details. | High | Sprint-4 |

# CHAPTER 6: PROJECT PLANNING & SCHEDULING

## 6.1 Sprint Planning & Estimation:

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 10 | High | DHARUN R |
| Sprint-1 | | USN-2 | As a user, I will receive confirmation email once I have registered for the application | 10 | High | KAVIN SV |
| Sprint-1 | Login | USN-3 | As a user, I can log into the application by entering email & password. | 15 | High | SHYAM GANESH T |
| Sprint-2 | Input Necessary Details | USN-4 | As a user, I can give Input Details to Predict Likeliness of crude oil | 15 | High | ROOPAKUMAR S |
| Sprint-2 | Data Pre-processing | USN-5 | Transform raw data into suitable format for prediction. | 15 | High | DHARUN R |
| Sprint-3 | Prediction of Crude Oil Price | USN-6 | As a user, I can predict Crude oil using machine learning model. | 20 | High | KAVIN SV |
| Sprint-3 | | USN-7 | As a user, I can get accurate prediction of crude oil | 5 | Medium | SHYAM GANESH T |
| Sprint-4 | Review | USN-8 | As a user, I can give feedback of the application. | 20 | High | ROOPAKUMAR S |

## 6.2 Sprint Delivery Schedule:

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---|---|---|---|---|---|---|
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | | |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | | |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

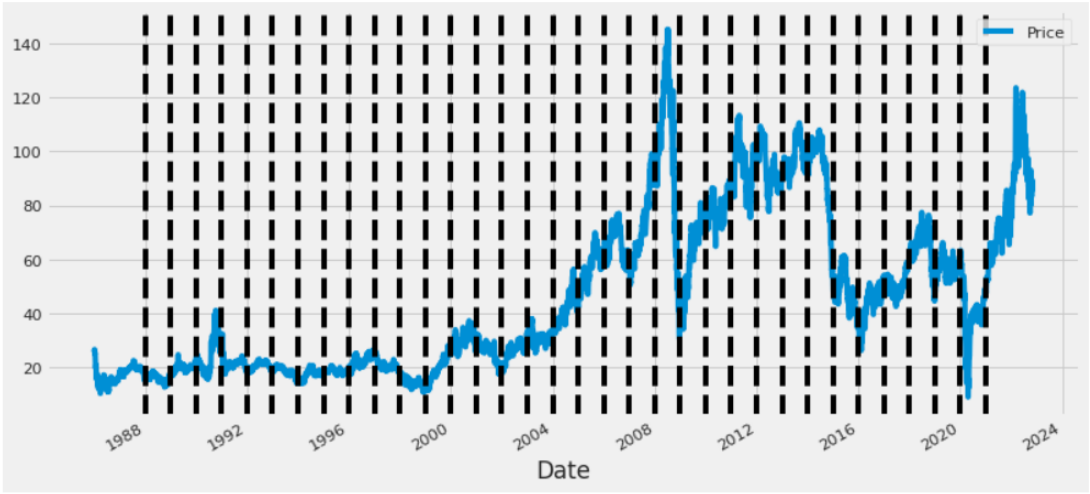
# CHAPTER 7: CODING AND SOLUTIONING

## 7.1 Feature 1:

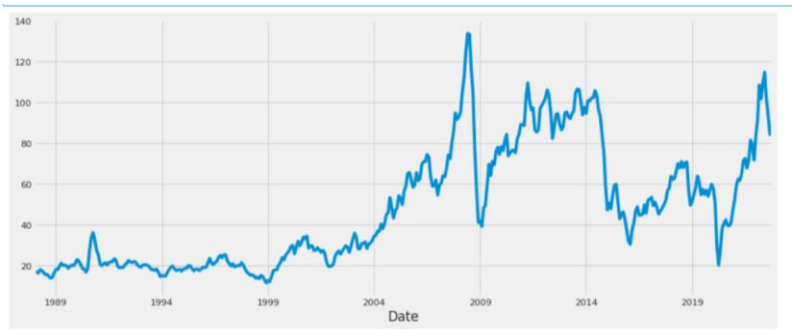
```
import numpy as np
import pandas as pd
import datetime
import matplotlib.pyplot as plt
import warnings
import itertools
import statsmodels.api as sm
import seaborn as sns
import math
from pylab import rcParams
from sklearn.preprocessing import MinMaxScaler

#Jupyter Notebook Specific
sns.set_context("paper", font_scale=1.3)
sns.set_style('white')
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
df = pd.read_csv(r'/content/Crude Oil Prices Daily.csv')
df['Date'] = pd.to_datetime(df['Date'])

#Data visualization
plt_1 = plt.figure(figsize=(15, 6))
time = pd.to_datetime(df['Date'])
data = list(df['Price'])
copdata = pd.Series(data, time)
plt.plot(copdata)
```


```
#observe for any seasonal patterns
ax = df.plot(x='Date', y='Price', figsize=(12,6))
xcoords = ['1988-01-01','1989-01-01','1990-01-01','1991-01-01','1992-01-
01','1993-01-01','1994-01-01',
```

```
          '1995-01-01','1996-01-01','1997-01-01','1998-01-01','1999-01-
01','2000-01-01','2001-01-01',
          '2002-01-01','2003-01-01','2004-01-01','2005-01-01','2006-01-
01','2007-01-01','2008-01-01',
          '2009-01-01','2010-01-01','2011-01-01','2012-01-01','2013-01-
01','2014-01-01','2015-01-01',
          '2016-01-01','2017-01-01', '2018-01-01', '2019-01-01', '2020-01-
01','2021-01-01']
for xc in xcoords:
    plt.axvline(x=xc, color='black', linestyle='--')
```




```
#Decompose the plot
df1 = df
df1 = df1.sort_values('Date')
df1 = df1.groupby('Date')['Price'].sum().reset_index()
df1.set_index('Date', inplace=True)
df1=df1.loc[datetime.date(year=1988,month=1,day=1):]

q = df1['Price'].resample('MS').mean()
q.plot(figsize=(15, 6))
plt.show()
```



```
#Importing regressor = Sequential()
regressor.add(LSTM(units = 60, return_sequences = True, input_shape =
(X_train.shape[1], 1)))
regressor.add(Dropout(0.1))

regressor.add(LSTM(units = 60, return_sequences = True))
regressor.add(Dropout(0.1))

regressor.add(LSTM(units = 60))
regressor.add(Dropout(0.1))

regressor.add(Dense(units = 1))

regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
# early_stopping = EarlyStopping(monitor='val_loss',patience=20)
history =regressor.fit(X_train, Y_train, epochs = 20, batch_size =
64,validation_data=(X_test, Y_test),verbose=1)
# history =regressor.fit(X_train, Y_train, epochs = 100, batch_size =
64,validation_data=(X_test, Y_test),
callbacks=[early_stopping],shuffle=False,verbose=1)
```

```
Epoch 1/20
95/95 [==============================] - 10s 45ms/step - loss: 0.0047 - val_loss: 7.2943e-04
Epoch 2/20
95/95 [==============================] - 3s 30ms/step - loss: 4.9550e-04 - val_loss: 7.6921e-04
Epoch 3/20
95/95 [==============================] - 3s 30ms/step - loss: 4.8898e-04 - val_loss: 8.0942e-04
Epoch 4/20
95/95 [==============================] - 3s 29ms/step - loss: 5.1047e-04 - val_loss: 0.0013
Epoch 5/20
95/95 [==============================] - 3s 30ms/step - loss: 5.9421e-04 - val_loss: 0.0023
Epoch 6/20
95/95 [==============================] - 3s 30ms/step - loss: 5.2523e-04 - val_loss: 0.0013
Epoch 7/20
95/95 [==============================] - 3s 31ms/step - loss: 5.0870e-04 - val_loss: 0.0023
Epoch 8/20
95/95 [==============================] - 4s 38ms/step - loss: 5.7511e-04 - val_loss: 6.1036e-04
Epoch 9/20
95/95 [==============================] - 3s 30ms/step - loss: 4.4533e-04 - val_loss: 6.3446e-04
Epoch 10/20
95/95 [==============================] - 3s 30ms/step - loss: 4.3985e-04 - val_loss: 6.5852e-04
Epoch 11/20
95/95 [==============================] - 3s 31ms/step - loss: 3.8284e-04 - val_loss: 5.7576e-04
Epoch 12/20
95/95 [==============================] - 3s 30ms/step - loss: 4.1176e-04 - val_loss: 6.4895e-04
Epoch 13/20
...
Epoch 19/20
95/95 [==============================] - 3s 31ms/step - loss: 3.4734e-04 - val_loss: 0.0011
Epoch 20/20
```

#Train the model

```
train_predict = regressor.predict(X_train)
test_predict = regressor.predict(X_test)
```

```
189/189 [==============================] - 3s 7ms/step
```

```
102/102 [==============================] - 1s 7ms/step
```

```
# invert predictions
train_predict = sc.inverse_transform(train_predict)
Y_train = sc.inverse_transform([Y_train])
test_predict = sc.inverse_transform(test_predict)
Y_test = sc.inverse_transform([Y_test])
```

#Model evaluation

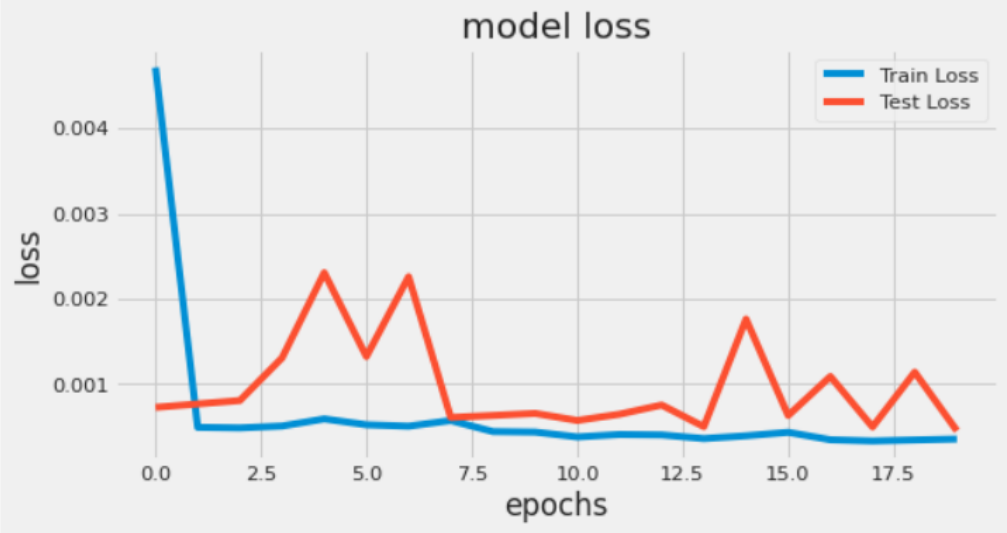
```
print('Train Mean Absolute Error:', mean_absolute_error(Y_train[0],
train_predict[:,0]))
print('Train Root Mean Squared Error:',np.sqrt(mean_squared_error(Y_train[0],
train_predict[:,0])))
print('Test Mean Absolute Error:', mean_absolute_error(Y_test[0],
test_predict[:,0]))
print('Test Root Mean Squared Error:',np.sqrt(mean_squared_error(Y_test[0],
test_predict[:,0])))
plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Test Loss')
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epochs')
plt.legend(loc='upper right')
plt.show();
```

```
Train Mean Absolute Error: 1.5004689769241186
```

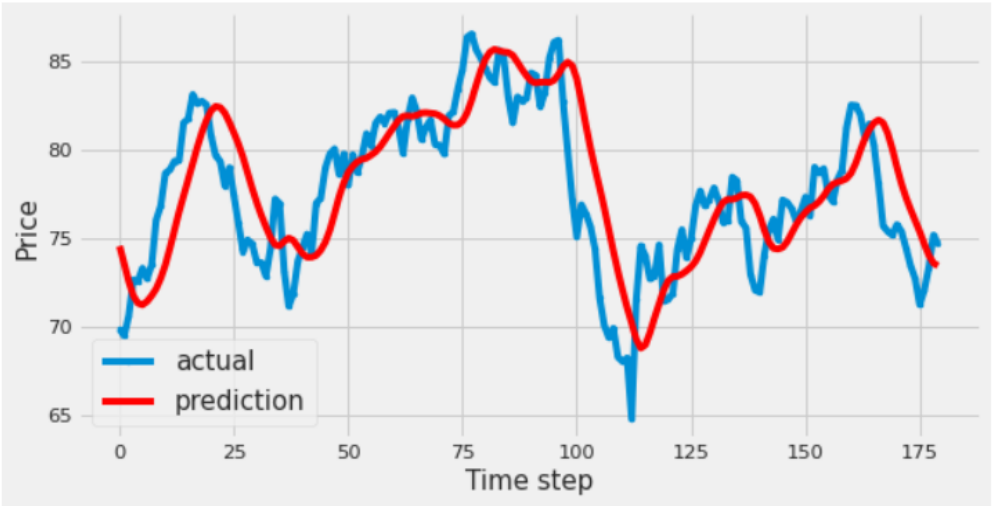
```
Train Root Mean Squared Error: 2.03291747578067
```

```
Test Mean Absolute Error: 2.1357170599548074
```

```
Test Root Mean Squared Error: 2.9096238300262343
```

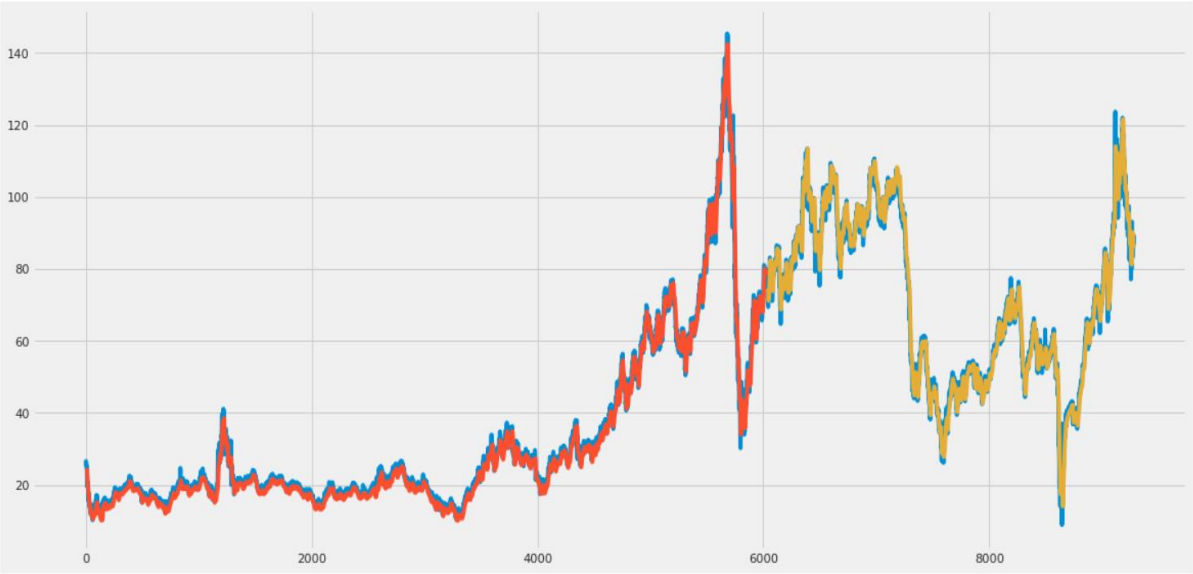



```
 aa=[x for x in range(180)]
plt.figure(figsize=(8,4))
plt.plot(aa, Y_test[0][:180], marker='.', label="actual")
plt.plot(aa, test_predict[:,0][:180], 'r', label="prediction")
plt.tight_layout()
sns.despine(top=True)
plt.subplots_adjust(left=0.07)
plt.ylabel('Price', size=15)
plt.xlabel('Time step', size=15)
plt.legend(fontsize=15)
plt.show();
```



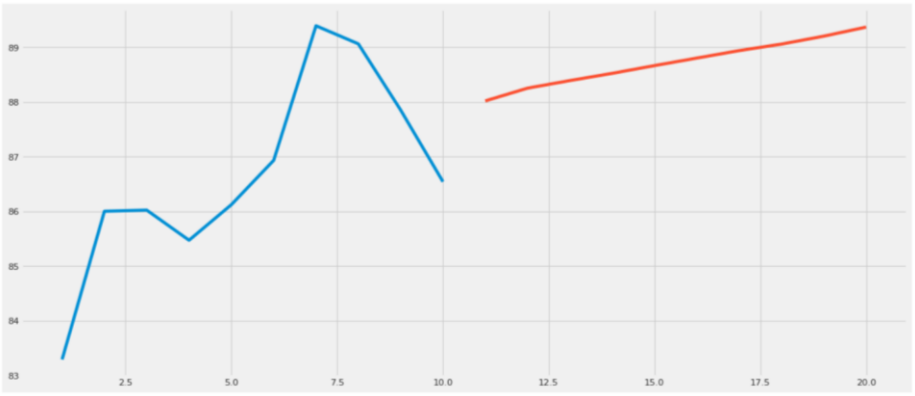

```
#test the model
look_back = 10
trainPredictPlot = np.empty_like(data_oil)
trainPredictPlot[:,:] = np.nan
trainPredictPlot[look_back:len(train_predict)+look_back,:] = train_predict
```

```
testPredictPlot = np.empty_like(data_oil)
testPredictPlot[:,:] = np.nan
testPredictPlot[len(train_predict)+(look_back*2)+1:len(data_oil)-1,:] =
test_predict

plt.plot(sc.inverse_transform(data_oil))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.show()
```




```
lst_output = []
n_steps = 10
i = 0

while(i<10):
  if(len(temp_input) > 10):
    x_input = np.array(temp_input[1:])
    print("{} day input {}".format(i,x_input),end="\n")
    x_input = x_input.reshape(1,-1)
    x_input = x_input.reshape((1,n_steps,1))

    yhat = regressor.predict(x_input,verbose=0)
    print("{} day output {}".format(i,yhat),end="\n")
    temp_input.extend(yhat[0].tolist())
    temp_input = temp_input[1:]
    print("-----------------------",end="\n")
    lst_output.extend(yhat.tolist())
    i = i+1
  else:
    x_input = x_input.reshape((1,n_steps,1))
```

```
    yhat = regressor.predict(x_input,verbose=0)
    print("{} day output {}".format(i,yhat),end="\n")
    temp_input.extend(yhat[0].tolist())
    lst_output.extend(yhat.tolist())
    i = i+1

day_new = np.arange(1,11)
day_pred = np.arange(11,21)
len(data_oil)

print(day_new)
print(day_pred)


plt.plot(day_new,sc.inverse_transform(data_oil[9271:]))
plt.plot(day_pred,sc.inverse_transform(lst_output))
```




### 7.2 Feature 2:

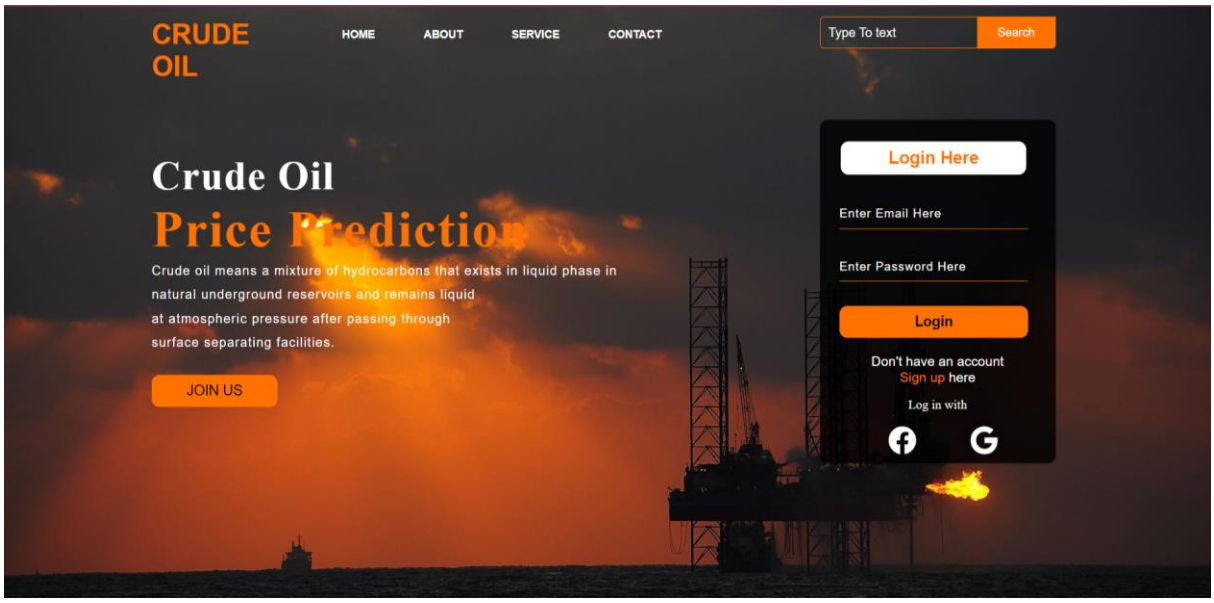
**Index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home page</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
```

```
<div class="main">
    <div class="navbar">
        <div class="icon">
            <h2 class="logo">CRUDE OIL</h2>
        </div>

        <div class="menu">
            <ul>
                <li><a href="#">HOME</a></li>
                <li><a href="#">ABOUT</a></li>
                <li><a href="#">SERVICE</a></li>
                <li><a href="#">CONTACT</a></li>
            </ul>
        </div>

        <div class="search">
            <input class="srch" type="search" name="" placeholder="Type To text">
            <a href="#"> <button class="btn">Search</button></a>
        </div>

    </div>
    <div class="content">
        <h1>Crude Oil<br><span>Price Prediction</span><br></h1>
        <p class="par"> Crude oil means a mixture of hydrocarbons that exists in liquid phase in<br>
```

```
            natural underground reservoirs and remains liquid <br>at atmospheric pressure
            after passing through <br>surface separating facilities.</p>

            <button class="cn"><a href="register.html">JOIN US</a></button>

            <div class="form">
               <h2>Login Here</h2>
               <input type="email" name="email" placeholder="Enter Email Here">
               <input type="password" name="" placeholder="Enter Password Here">
               <button class="btnn"><a href="#">Login</a></button>

               <p class="link">Don't have an account<br>
               <a href="#">Sign up </a> here</a></p>
               <p class="liw">Log in with</p>

               <div class="icons">
                  <a href="#"><ion-icon name="logo-facebook"></ion-icon></a>
                  <a href="#"><ion-icon name="logo-google"></ion-icon></a>
               </div>

            </div>
               </div>
            </div>
      </div>
   </div>
```

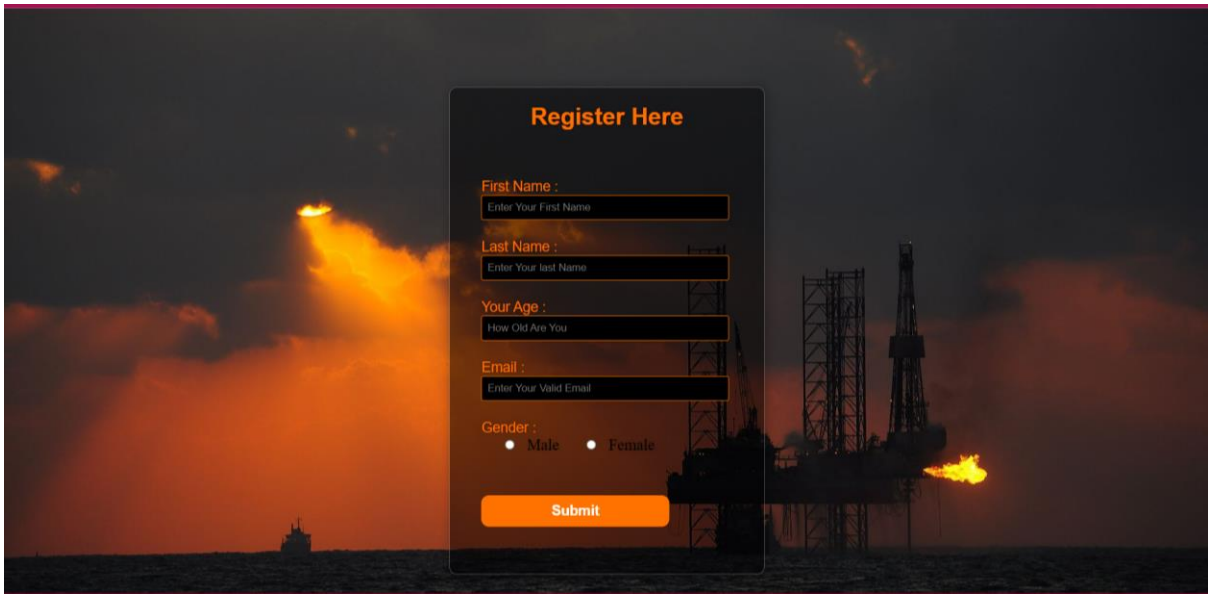
```
    <script src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js"></script>
</body>
</html>
```




**Register.Html**

```
<!DOCTYPE html>
<html>
   <head>
      <title>Registration Form</title>
      <link rel="stylesheet"
      href="register.css" type="text/css">
   </head>
   <body>
      <div class="main">
         <div class="register">
            <h2>Register Here</h2>
            <form id="register" method="post">
               <label>First Name : </label>
               <br>
               <input type="text" name="fname"
               id="name" placeholder="Enter Your First Name">
               <br><br>
               <label>Last Name : </label>
               <br>
               <input type="text" name="lname"
               id="name" placeholder="Enter Your last Name">
```

```
                <br><br>
                <label>Your Age : </label>
                <br>
                <input type="number" name="age"
                id="name" placeholder="How Old Are You">
                <br><br>
                <label>Email : </label>
                <br>
                <input type="email" name="email"
                id="name" placeholder="Enter Your Valid Email">
                <br><br>
                <label>Gender : </label>
                <br>
                     
                <input type="radio" name="gender"
                id="male">
                 
                <span id="male">Male</span>
                     
                <input type="radio" name="gender"
                id="female">
                 
                <span id="female">Female</span>
                <br><br>
                <input type="submit" value="Submit"
                name="submit" id="submit">
            </form>
        </div>
    </div>
  </body>
</html>
```


Register Here
First Name :
Enter Your First Name
Last Name :
Enter Your last Name
Your Age :
How Old Are You
Email :
Enter Your Valid Email
Gender :
Male
Female
Submit

# CHAPTER 8: TESTING

## 8.1 Test cases:

Test case analysis This report shows the number of test cases that have passed, failed, and untested.

| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| ML Model | 4 | 0 | 0 | 4 |
| Flask Application | 4 | 0 | 0 | 4 |
| IBM cloud | 4 | 0 | 0 | 4 |
| Exception Reporting | 2 | 0 | 0 | 2 |
| Final Report output | 4 | 0 | 0 | 4 |

## 8.2 User Acceptance testing

The purpose is to briefly explain the test coverage and open issues of the crude oil price prediction project at the time of the release to user acceptance testing

**Defect Analysis:**

| | | | | | |
|---|---|---|---|---|---|
| Not Reproduced | 0 | 0 | 0 | 0 | 0 |
| Skipped | 0 | 0 | 0 | 0 | 0 |
| Won't fix | 0 | 0 | 0 | 1 | 1 |
| Totals | 8 | 0 | 2 | 2 | 12 |

**Test case analysis**

This report shows the number of test cases that have passed, failed, and untested

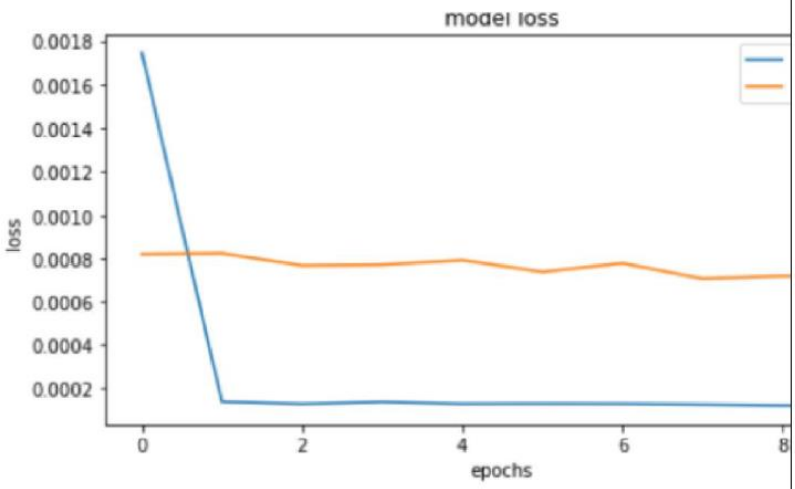
| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| ML Model | 4 | 0 | 0 | 4 |
| Flask Application | 4 | 0 | 0 | 4 |
| IBM Cloud | 4 | 0 | 0 | 4 |
| Exception Reporting | 2 | 0 | 0 | 2 |
| Final Report Output | 4 | 0 | 0 | 4 |

The report shows the number of resolved and closed bugs at each severity level and how they were resolved

| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Subtotal |
|---|---|---|---|---|---|
| By Design | 3 | 0 | 0 | 0 | 3 |
| Duplicate | 1 | 0 | 1 | 0 | 2 |
| External | 0 | 0 | 0 | 0 | 0 |
| Fixed | 4 | 0 | 1 | 1 | 6 |

# CHAPTER 9: RESULTS

| S.No | Parameters | Values | Screenshot |
|---|---|---|---|
| 1. | Model Summary | | Model: "sequential_1"<br>_____<br>Layer (type) Output Shape<br>==========<br>lstm_3 (LSTM) (None, 10, 50)<br>lstm_4 (LSTM) (None, 10, 50)<br>lstm_5 (LSTM) (None, 50)<br>dense_1 (Dense) (None, 1)<br>==========<br>Total params: 50,851<br>Trainable params: 50,851<br>Non-trainable params: 0 |

| | | |
|---|---|---|
| 2. | Accuracy | Train Mean Absolute Error: 1.0278217422906264<br>Train Root Mean Squared Error: 1.42852486399340<br>Test Mean Absolute Error: 2.780526920817909<br>Test Root Mean Squared Error: 3.634823446652373<br>  |

# CHAPTER 10: ADVANTAGES AND DISADVANTAGES

**Advantages:**

• Prediction of crude oil price can help the importers to choose the right time to buy as they wait for the prices to fall down

• Prediction of crude oil prices can help the exporters to increase the demand

• It can even help in shifting the political powers

• can assist in minimizing the risks associated with volatility in oil prices

**Disadvantages**

• The prediction results may lack accuracy

• Volatility in prices may be misleading

# CHAPTER 11: CONCLUSION

Crude oil prices are highly fluctuated time series. It is affected by many economic and political factors. Specially, there are several sudden increase and decrease throughout the time. In order to eliminate the irregular trend. We try several methods, hp-filter, loess filter, log transformation and difference. It shows that log transformation and difference have the best performance. After this manipulation, the data is generally stationary.

# CHAPTER 12: FUTURE SCOPE

Based on the whole modelling and prediction, we can conclude that our model is relatively easy to handle and capture the main feature. But the crude oil price is not that stationary for analysis, and exhibits intensive fluctuation even after transformation. The crude oil price can suffer sudden increase and decrease. Thus, only analysis of the crude oil price itself can hardly predict the sudden change. Maybe, we can find some latent variable to improve modelling and prediction.

The result obtained from simulation study validates the effectiveness of data selection process. This model successfully extracts a comprehensive list of key factors that cause the crude oil price market to volatile. The effectiveness and accurateness of data selection also helps to extensively deliberate the input variables combination for this model. Data represented in process had successfully proved to cleanse and uniform the data from errors and noises hence, the crisp prediction result. This research is now in its extension level to comprehend this quantitative part of prediction with the qualitative part. This work in progress is expected to trigger and show some interesting information and trend for this crude oil price and together will result to a better prospect for crude oil price prediction in the future.

## CHAPTER 13: APPENDIX

### 13.1 Source Code:

### Web Application to get the Live location:

### index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Home page</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>

   <div class="main">
      <div class="navbar">
         <div class="icon">
            <h2 class="logo">CRUDE OIL</h2>
         </div>

         <div class="menu">
            <ul>
               <li><a href="#">HOME</a></li>
               <li><a href="#">ABOUT</a></li>
               <li><a href="#">SERVICE</a></li>
               <li><a href="#">CONTACT</a></li>
            </ul>
         </div>

         <div class="search">
            <input class="srch" type="search" name="" placeholder="Type To
text">
            <a href="#"> <button class="btn">Search</button></a>
         </div>

      </div>
      <div class="content">
         <h1>Crude Oil<br><span>Price Prediction</span><br></h1>
         <p class="par"> Crude oil means a mixture of hydrocarbons that exists
in liquid phase in<br>
```

```
            natural underground reservoirs and remains liquid <br>at atmospheric
pressure
            after passing through <br>surface separating facilities.</p>

            <button class="cn"><a href="register.html">JOIN US</a></button>

            <div class="form">
               <h2>Login Here</h2>
               <input type="email" name="email" placeholder="Enter Email
Here">
               <input type="password" name="" placeholder="Enter Password
Here">
               <button class="btnn"><a href="#">Login</a></button>

               <p class="link">Don't have an account<br>
               <a href="#">Sign up </a> here</a></p>
               <p class="liw">Log in with</p>

               <div class="icons">
                  <a href="#"><ion-icon name="logo-facebook"></ion-icon></a>
                  <a href="#"><ion-icon name="logo-google"></ion-icon></a>
               </div>

            </div>
               </div>
            </div>
      </div>
   </div>
   <script src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js"></script>
</body>
</html>
```

**Index.css**

```
*{
   margin: 0;
   padding: 0;
}

.main{
   width: 100%;
   background:url(cr.jpg);
   background-position: center;
```

```
    background-size: cover;
    height: 100vh;
}
.navbar{
    width: 1200px;
    height: 75px;
    margin: auto;
}
.icon{
    width: 200px;
    float: left;
    height: 70px;
}
.logo{
    color: #ff7200;
    font-size: 35px;
    font-family: Arial;
    padding-left: 20px;
    float: left;
    padding-top: 10px;
    margin-top: 5px
}
.menu{
    width: 400px;
    float: left;
    height: 70px;
}
ul{
    float: left;
    display: flex;
    justify-content: center;
    align-items: center;
}
ul li{
    list-style: none;
    margin-left: 62px;
    margin-top: 27px;
    font-size: 14px;
}
ul li a{
    text-decoration: none;
    color: #fff;
    font-family: Arial;
```

```
    font-weight: bold;
    transition: 0.4s ease-in-out;
}
ul li a:hover{
    color: #ff7200;
}
.search{
    width: 330px;
    float: left;
    margin-left: 270px;
}
.srch{
    font-family: 'Times New Roman';
    width: 200px;
    height: 40px;
    background: transparent;
    border: 1px solid #ff7200;
    margin-top: 13px;
    color: #fff;
    border-right: none;
    font-size: 16px;
    float: left;
    padding: 10px;
    border-bottom-left-radius: 5px;
    border-top-left-radius: 5px;
}
.btn{
    width: 100px;
    height: 40px;
    background: #ff7200;
    border: 2px solid #ff7200;
    margin-top: 13px;
    color: #fff;
    font-size: 15px;
    border-bottom-right-radius: 5px;
    border-bottom-right-radius: 5px;
    transition: 0.2s ease;
    cursor: pointer;
}
.btn:hover{
    color: #000;
}
.btn:focus{
```

```
    outline: none;
}
.srch:focus{
    outline: none;
}
.content{
    width: 1200px;
    height: auto;
    margin: auto;
    color: #fff;
    position: relative;
}
.content .par{
    padding-left: 20px;
    padding-bottom: 25px;
    font-family: Arial;
    letter-spacing: 1.2px;
    line-height: 30px;
}
.content h1{
    font-family: 'Times New Roman';
    font-size: 50px;
    padding-left: 20px;
    margin-top: 9%;
    letter-spacing: 2px;
}
.content .cn{
    width: 160px;
    height: 40px;
    background: #ff7200;
    border: none;
    margin-bottom: 10px;
    margin-left: 20px;
    font-size: 18px;
    border-radius: 10px;
    cursor: pointer;
    transition: .4s ease;

}
.content .cn a{
    text-decoration: none;
    color: #000;
    transition: .3s ease;
```

```
}
.cn:hover{
    background-color: #fff;
}
.content span{
    color: #ff7200;
    font-size: 65px
}
.form{
    width: 250px;
    height: 380px;
    background: linear-gradient(to top, rgba(0,0,0,0.8)50%,rgba(0,0,0,0.8)50%);
    position: absolute;
    top: -20px;
    left: 870px;
    transform: translate(0%,-5%);
    border-radius: 10px;
    padding: 25px;
}
.form h2{
    width: 220px;
    font-family: sans-serif;
    text-align: center;
    color: #ff7200;
    font-size: 22px;
    background-color: #fff;
    border-radius: 10px;
    margin: 2px;
    padding: 8px;
}
.form input{
    width: 240px;
    height: 35px;
    background: transparent;
    border-bottom: 1px solid #ff7200;
    border-top: none;
    border-right: none;
    border-left: none;
    color: #fff;
    font-size: 15px;
    letter-spacing: 1px;
    margin-top: 30px;
    font-family: sans-serif;
```

```
}
.form input:focus{
    outline: none;
}
::placeholder{
    color: #fff;
    font-family: Arial;
}
.btnn{
    width: 240px;
    height: 40px;
    background: #ff7200;
    border: none;
    margin-top: 30px;
    font-size: 18px;
    border-radius: 10px;
    cursor: pointer;
    color: #fff;
    transition: 0.4s ease;
}
.btnn:hover{
    background: #fff;
    color: #ff7200;
}
.btnn a{
    text-decoration: none;
    color: #000;
    font-weight: bold;
}
.form .link{
    font-family: Arial, Helvetica, sans-serif;
    font-size: 17px;
    padding-top: 20px;
    text-align: center;
}
.form .link a{
    text-decoration: none;
    color: #ff7200;
}
.liw{
    padding-top: 15px;
    padding-bottom: 10px;
    text-align: center;
```

```
}
.icons a{
    text-decoration: none;
    color: #fff;
}
.icons ion-icon{
    color: #fff;
    font-size: 40px;
    padding-left: 60px;
    padding-top: 5px;
    transition: 0.3s ease;
}
.icons ion-icon:hover{
    color: #ff7200;
}
@media screen and (max-width:1200px) {
    /*Normal Screen*/

    .navbar{
        width: 100%;
        height: 100px;
    }
    ul{
        margin-left: 30px;
    }
    ul li{
        margin-left: 60px;
    }
    ul li a{
       font-size: 1.6vw;
    }
    .search{
        margin-top: 3px;
        margin-left: 290px;
    }
    .srch{
        height: 40px;
        width: 190px;
        font-size: 14px;
    }
    .btn{
        height:40px;
        width: 80px;
```

```
    }
    .content{
        width: 100%;
    }
    .content h1, .content span{
        font-size: 4.5vw;
    }
    .content .par{
        width: 90%;
        font-size: 1.5vw;
    }
    .content .cn{
        width: 13%;
        height: 3.5vw;
        font-size: 1.8vw;
    }
    .content a{
        font-size: 1.6vw
    }
}

@media screen and (max-width:1170px) {
    /*Login-box*/
    .main{
        padding-left: 20px;
        height: 180vh;
    }
    .form{
        margin-left: -30px;
        width: 250px;
        height: 370px;
        background: linear-gradient(to top,
rgba(0,0,0,0.8)50%,rgba(0,0,0,0.8)50%);
        position: absolute;
        top: 420px;
        left: 50px;
        transform: translate(0%,-5%);
        border-radius: 10px;
        padding: 25px;
    }
    .form input{
        width: 240px;
        height: 35px;
```

```
        background: transparent;
        border-bottom: 1px solid #ff7200;
        border-top: none;
        border-right: none;
        border-left: none;
        color: #fff;
        font-size: 15px;
        letter-spacing: 1px;
        margin-top: 30px;
        font-family: sans-serif;
    }
    .btnn a{
        font-size: 16px;
    }
    .form .link a{
        font-size: 16px;
    }
}

@media screen and (max-width: 830px){
    /*For tablet*/

    .content{
        margin-top: 120px;
        width: 80%;
        margin-left: 40px;
    }
    .content h1, .content span{
        font-size: 6vw;
    }
    .content .par{
        width: 90%;
        font-size: 1.8vw;
    }
    .content .cn{
        width: 15%;
        height: 4.5vw;
        font-size: 2vw;
    }
    .content a{
        font-size: 2vw
    }
    .logo{
```

```
        margin-left: 240px;
        width: 100%;
        margin-top: 15px;
        font-size: 5vw;
    }
    .menu{
        width: 100%;
    }
    ul{
        margin-top: -5px;
        margin-left: 5px;
    }
    ul li{
        margin-left: 60px;
    }
    ul li a{
       font-size: 2vw;
    }
    .search{
        margin-top: -20px;
        margin-left: 60px;
    }
    .srch{
        height: 30px;
        width: 160px;
        font-size: 12px;
    }
    .btn{
        height:30px;
        width: 70px;
    }
    .main{
        padding-left: 20px;
        height: 180vh;
    }
    .form{
        margin-left: -30px;
        width: 250px;
        height: 370px;
        background: linear-gradient(to top,
rgba(0,0,0,0.8)50%,rgba(0,0,0,0.8)50%);
        position: absolute;
        top: 430px;
```

```
        left: 50px;
        transform: translate(0%,-5%);
        border-radius: 10px;
        padding: 25px;
    }
    .form input{
        width: 240px;
        height: 35px;
        background: transparent;
        border-bottom: 1px solid #ff7200;
        border-top: none;
        border-right: none;
        border-left: none;
        color: #fff;
        font-size: 15px;
        letter-spacing: 1px;
        margin-top: 30px;
        font-family: sans-serif;
    }
    .btnn a{
        font-size: 16px;
    }
    .form .link a{
        font-size: 16px;
    }
}

@media screen and (max-width: 600px){
    /*IPAD*/

    .content{
        margin-top: 80px;
        margin-left: 20px;
    }
    .search{
        margin-top: -40px;
        margin-left: 42px;
    }
    .logo{
        margin-left: 180px;
        font-size: 4vw;
    }
    ul{
```

```
        margin-top: -25px;
        margin-left: -5px;
    }
    ul li {
        margin-left: 50px;
    }
    ul li a{
        font-size: 2vw;
    }
}

@media screen and (max-width: 450px){
    /*mobile*/
    .logo{
        margin-left: 140px;
        font-size: 4vw;
    }
    ul{
        margin-top: -25px;
    }
    ul li {
        margin-left: 42px;
    }
    ul li a{
        font-size: 2vw;
    }
    .search{
        margin-top: -40px;
        margin-left: 38px;
    }

}
```

**Main.py**

```
import numpy as np
import requests
from flask import Flask, render_template, request
from tensorflow.python.keras.models import load_model

app = Flask(__name__)

model = load_model("D:\Documents\ibm\fetch.tar.gz")
```

```
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":"1gwmabI1uLfI58p348zk2GOm5l-4Gq-30YwWSKgGvsUY",
"grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}


ent = []

@app.route('/')
def home():
    return render_template("index.html")

@app.route('/predict', methods=["GET", "POST"])
def home1():
    if request.method == "POST":
        data = request.form['data']
        # print(data)
        ent.append(data)

        inp = str(request.form['data'])
        inp = inp.split(',')
        for i in range(0, len(inp)):
            inp[i] = float(inp[i])
        inp = np.array(inp).reshape(1,-1)
        temp = list(inp)
        temp = temp[0].tolist()
        # print(temp)
        outpt = []
        n_steps = 10
        i=0
        while(i<1):
            if(len(temp) > 10):
                #some prblm
                inp = np.array(temp[1:])
                print("{} day input {}".format(i, inp))
                inp = inp.reshape(1,-1)
                inp = inp.reshape((1, n_steps, 1))
                yhat = model.predict(inp, verbose=0)
                temp.extend(yhat[0].tolist())
                temp = temp[1:]
                outpt.extend(yhat.tolist())
```

```
            i += 1
        else:
            inp = inp.reshape((1,len(temp),1)) #n_steps --> len(temp)
            yhat = model.predict(inp, verbose=0)
            temp.extend(yhat[0].tolist())
            outpt.extend(yhat.tolist())
            i+=1


        return render_template("predict_opt.html", entries = outpt)
    else:
        return render_template("predict_opt.html")

@app.route('/market')
def home2():
    return render_template("market.html")


if __name__ == '__main__':
    app.run(debug=True)
```