**CRUDE OIL PRICE PREDICTION**

**Project Report Format**

**1. INTRODUCTION**

1.1 Project Overview

1.2 Purpose

**2. LITERATURE SURVEY**

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

**3. IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

**4. REQUIREMENT ANALYSIS**

4.1 Functional requirement

4.2 Non-Functional requirements

**5. PROJECT DESIGN**

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

**6. PROJECT PLANNING & SCHEDULING**

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

**7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

**8. TESTING**

8.1 Test Cases

8.2 User Acceptance Testing

**9. RESULTS**

9.1 Performance Metrics

**10. ADVANTAGES & DISADVANTAGES**

**11. CONCLUSION**

**12. FUTURE SCOPE**

**13. APPENDIX**

**Source Code**

**GitHub & Project Demo Link**

**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. Our work mainly focuses on applying Recurrent Neural Networks to predict the Crude Oil Price.

This decision helps common people 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 price of the crude oil. So we would be implementing RNN (Recurrent Neural Network) with LSTM (Long Short Term Memory) to achieve the task.

We will be experimenting with different types of models with varying number of epochs, look backs and other tuning methods.

**1.2 PURPOSE**:

The paper summaries about LSTM network is an improved method as compared to other ordinary neural network for prediction of oil prices as an objective in the motion of back propagation model.

Traditional or ordinary neural network such as RRN or CNN on contrast assumes the next outgoing but can’t essentially store the previous data or connection that is dependent on feed-forwarding, in the sense the previous data is not compulsory to forecasting the later data.

LSTM clears about keeping the previous data and prediction which might be encouraging and more accurate. The possible results are comparatively inspiring .

This outcomes shows that the huge process may not definitely work on the correctness of the prediction of crude oil prices. Thus, it might be finished and thus the model with single LSTM model surely be highly accurate.

**2. LITERATURE SURVEY**

**2.1 Existing problem**

**TITLE**: Predictive Analytics for crude oil prediction using RNN-LSTM Neural network

**AUTHORS** : Norshakirah aziz and mohd hafizul afifi abdullah

**YEAR**:2020

Prediction of future crude oil price is considered a significant challenge due to the extremely complex, chaotic, and dynamic nature of the market and stakeholder's perception. The crude oil price changes every minute, and millions was crude oil is influenced by many factors including news, supply-and-demand gap, labour costs, amount of remaining resources, as well as stakeholders' perception. Therefore, various indicators for technical analysis have been utilized for the purpose of

predicting the future crude oil price. Recently, many researchers have turned to machine learning approached to cater to this problem. This study demonstrated the use of RNN-LSTM networks for predicting the crude oil price based on historical data alongside other technical analysis indicators. This study aims to certify the capability of a prediction model built based on the RNN-LSTM network to predict the future price of crude oil. The developed model is trained and evaluated against accuracy matrices to assess the capability of the network to provide an improvement of the accuracy of crude oil price prediction as compared to other strategies. The result obtained from the model shows a promising prediction capability of the RNN-LSTM algorithm for predicting crude oil price movement.

**TITLE:**Crude oil price using Artificial Neural network

**AUTHORS**: Nalini gupta and shobhit nigam, pandit deen dayal petroleum university.

**YEAR**:2020

Crude oil is amongst the most important resources in today’s world, it is the chief fuel and its cost has a direct effect on the global habitat, our economy and oil exploration, exploitation and other activities. Prediction of oil prices has become the need of the hour, it is a boon to many large and small industries, individuals, the government. The evaporative nature of crude oil, its price prediction becomes extremely difficult and it is hard to be precise with the same. Several different factors that affect crude oil prices. We propose a contemporary and innovative method of predicting crude oil prices using the artificial neural network (ANN). The main advantage of this approach of ANN is that it continuously captures the unstable pattern of the crude oil prices which have been incorporated by finding out the optimal lag and number of the delay effect that controls the prices of crude oil. Variation of lag in a period of time has been done for the most optimum and close results, we then have validated our results by evaluating the root mean square error and the results obtained using the proposed model have significantly outperformed.

**TITLE**: Forecasting crude oil price using Artificial Neural network model

**AUTHORS**: Sivaprakash j and manu ks

**YEAR**:2021

In the advanced global economy, crude oil is a commodity that plays a major role in every economy. As Crude oil is highly traded commodity it is essential for the investors, analysts, economists to forecast the future spot price of the crude oil appropriately. In the last year the crude oil faced a historic fall during the pandemic and reached all time low, but will this situation last? There was analysis such as fundamental analysis, technical analysis and time series analyses which were carried out for predicting the movement of the oil prices but the accuracy in such prediction is still a question. Thus, it is necessary to identify better methods to forecast the crude oil prices. This study is an empirical study to forecast crude oil prices using the neural networks. This study consists of 13 input variables with one target variable. The data are divided in the ratio 70:30. The 70% data is used for training the network and 30% is used for testing. The feed forward and back propagation algorithm are used to predict the crude oil price. The neural network proved to be efficient in forecasting in the modern era. A simple neural network performs better than the time series models. The study found that back propagation algorithm performs better while predicting the crude oil price. Hence, ANN can be used by the investors, forecasters and for future researchers.

**2.2 References**

[1] Mohammad Reza Mahdiani and Ehsan Khamehchi, “A modified neural network model for predicting the crude oil price”, Intellectual Economics, vol. 10, no. 2, pp. 71-77, Aug. 2016. [2] Manel Hamdi and Chaker Aloui, \"Forecasting Crude Oil Price Using Artificial NeuralNetworks: A Literature Survey,\" Economics Bulletin, AccessEcon, vol. 35, no. 2, pp. 1339-1359, 2015. [3] [3] Yu Runfang, Du Jiangze and Liu Xiaotao, “Improved Forecast Ability of Oil Market Volatility Based on combined Markov Switching and GARCH-class Model, Procedia Computer Science, vol. 122, pp. 415-422, 2017. [4] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, \"LSTM: A Search Space Odyssey,\" IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232,Oct. 2017. [5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [6] Jammazi, R., Aloui, C.: Crude oil price forecasting: experimental evidence from wavelet decomposition and neural network modeling. Energy Econ. 34(3), 828–841 (2012) [7] S. Moshiri, and F. Foroutan, “Forecasting nonlinear crude oil futures prices,” The Energy Journal vol. 27, pp. 81-95, 2005. [8] Siddhi Vinayak Kulkarni and Imad Haidar, Forecasting Model for Crude Oil Price Using Artificial Neural Networks and Commodity Futures Prices. International Journal of Computer Science and Information Security, vol. 2, no.1, June 2009. [9] Hamdi and Aloui, \"Machine learning approach for crude oil price prediction with Artificial Neural Networks-Quantitative (ANN-Q) model,\" The 2010 International JointConference on Neural Networks (IJCNN), Barcelona, pp. 1-8, 2010. [10] Abdullah and Zeng.: Exploring the core factors and its dynamic effects on oil price: An application on path analysis and BVAR-TVP model. Energy Policy 39(12), 8022–8036 (2011) [11] Chen et al.: Forecasting the crude oil spot price by wavelet neural networks using OECD petroleum inventory levels. New Math. Nat. Comput. 07(02), 281–297 (2011)

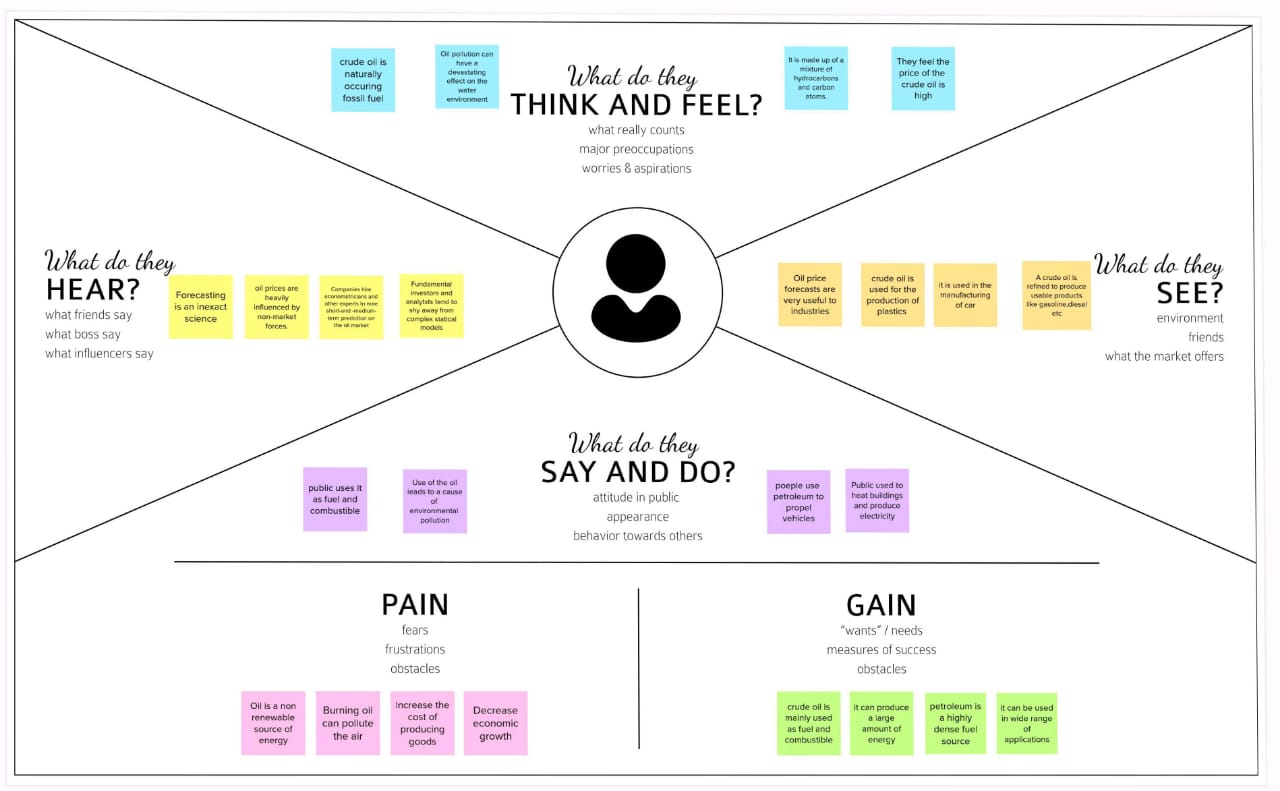
**2.3 Problem Statement Definition**

The existing patrimony model of oil price prediction is not capable enough to deliver the accurate predicted prices as expected. Another problem arises is the factors that are being considering in the prediction model. Few factors can be described as the conjectural buying and selling , geopolitical, OPEC output, increased demand from important role in the prediction of the oil prices. Now problem arising with the current ANN and CNN models that are used as prediction model’s are that they can’t provide accurate results when the data is too big. The big reason of not being successful enough is that these models uses backward propagation which lead to only derivative error , where we need the model to propagate forward as well to get the desired output and it can be compared with real value to fetch the errors occurring in the models.To over come this problem LSTM (long short term memory) algorithm was proposed which uses backward and feed-forward propagation which helps to get more accurate results.

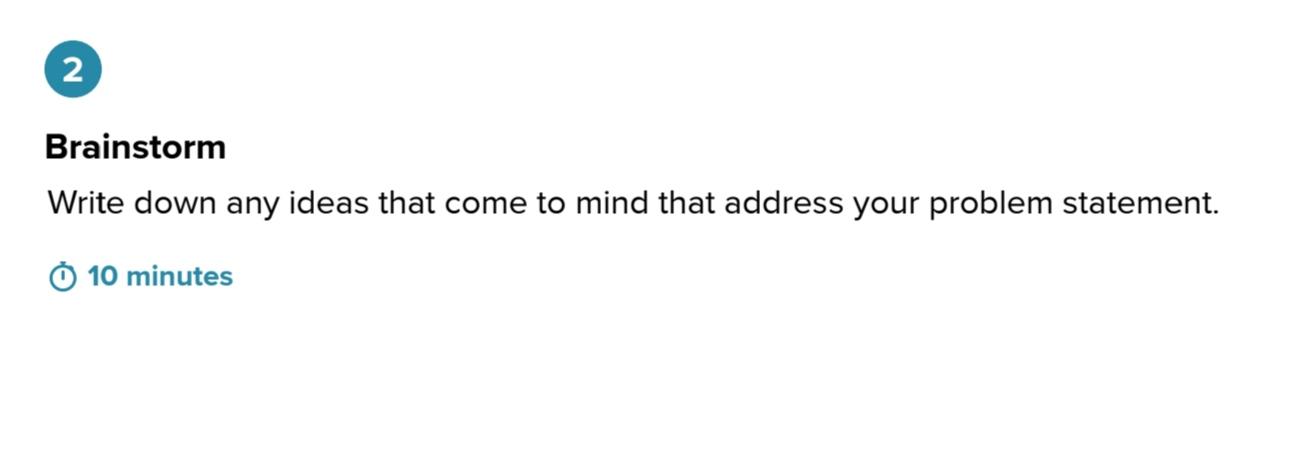
**Solution**:

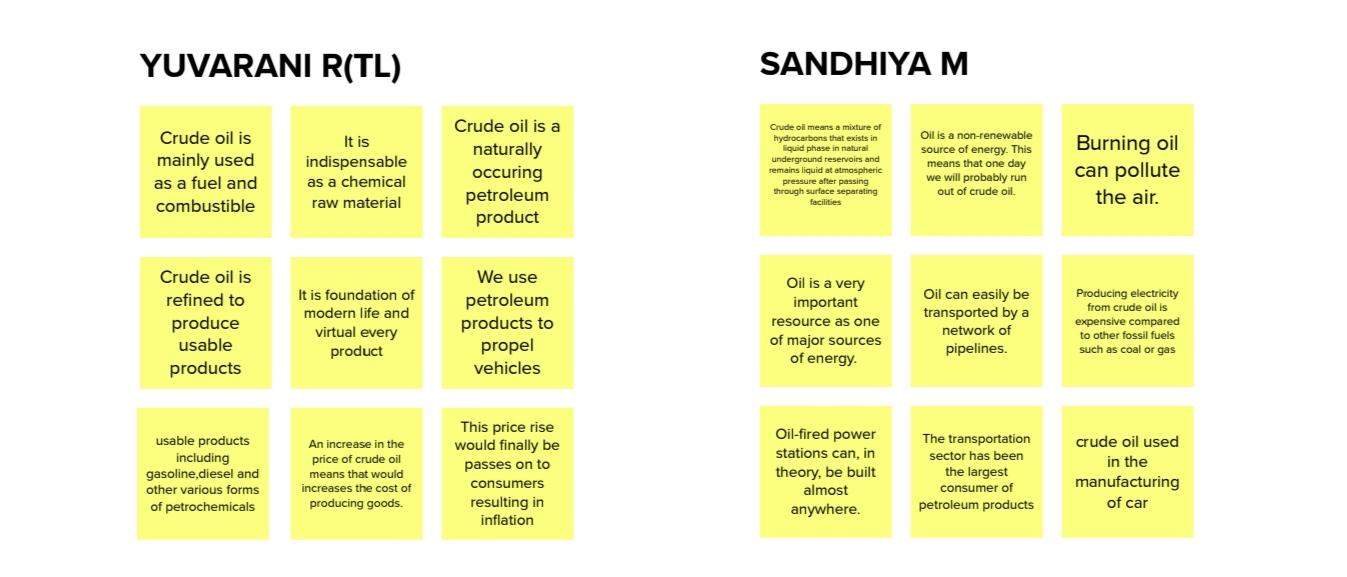
The paper summaries about LSTM network is an improved method as compared to other ordinary neural network for prediction of oil prices as an objective in the motion of back propagation model. Traditional or ordinary neural network such as rrn or cnn on contrast assumes the next outgoing but can’t essentially store the previous data or connection that is dependent on feed-forwarding, in the sense the previous data is not compulsory to forecasting the later data. LSTM clears about keeping the previous data and prediction which might be encouraging and more accurate. The possible results are comparatively inspiring .This outcomes shows that the huge process may not definitely work on the correctness of the prediction of crude oil prices. Thus, it might be finished and thus the model with single LSTM model surely be highly accurate.

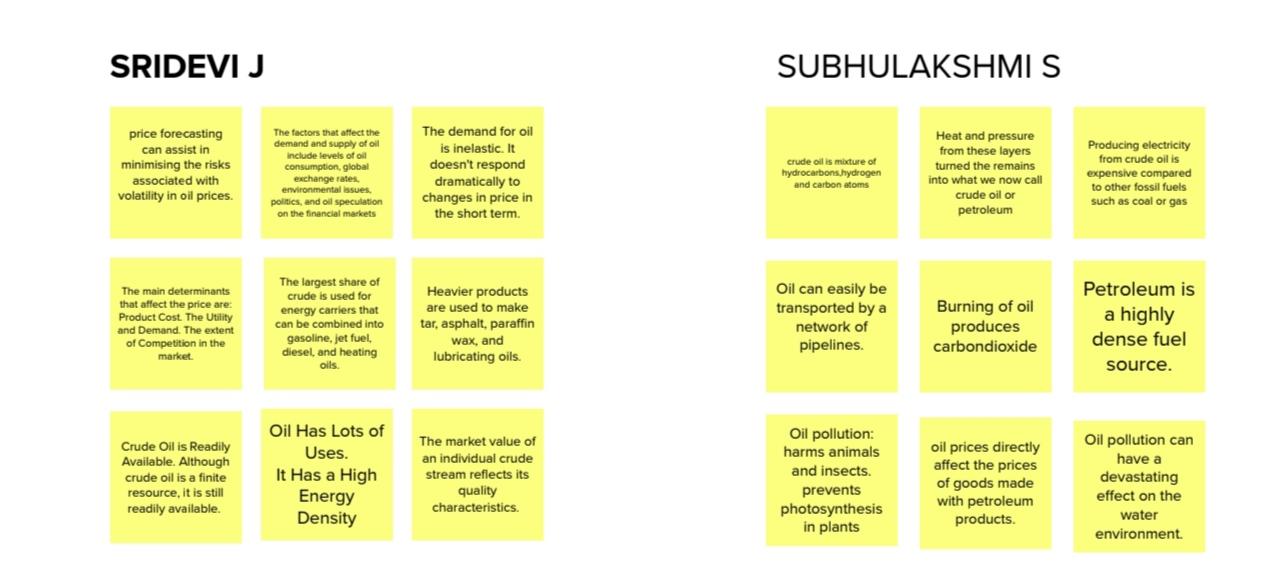
**3. IDEATION & PROPOSED SOLUTION**

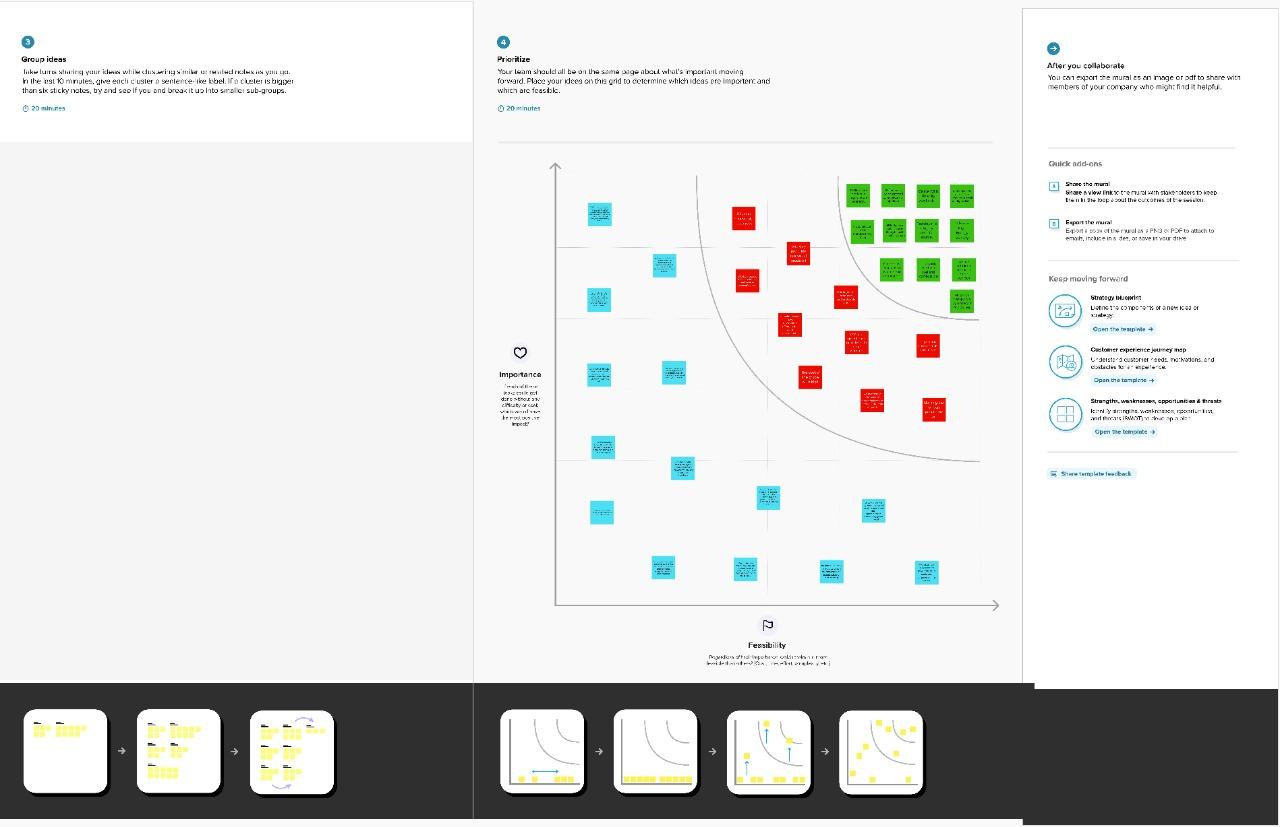
**3.1 Empathy Map Canvas**

**3.2 Ideation & Brainstorming**

****

****

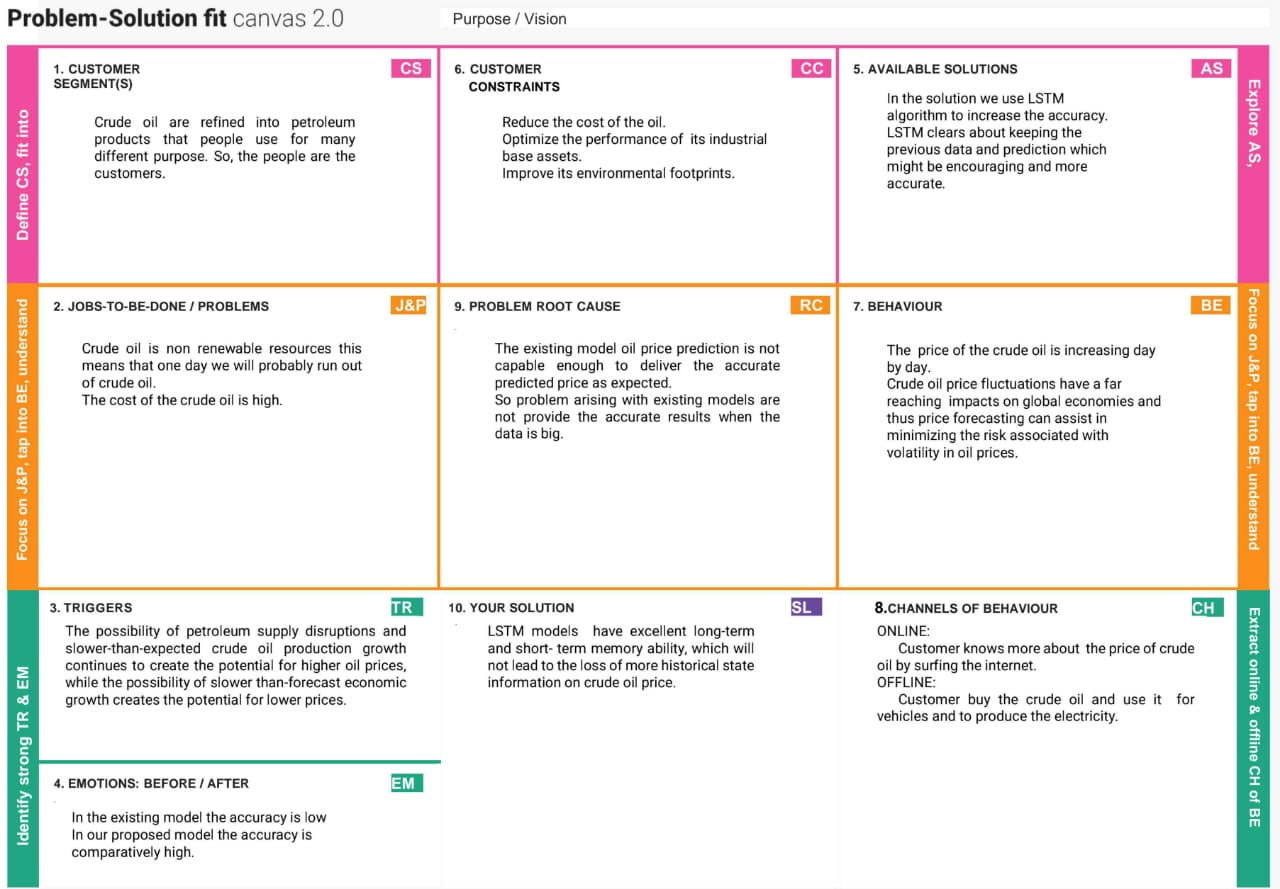
****

****

**3.3 PROPOSED SOLUTION**

| **S.No.** | **Parameter** | **Description** |
| --- | --- | --- |
| 1. | Problem Statement (Problem to be solved) | The existing patrimony model of oil price prediction is not capable enough to deliver the accurate predicted prices as expected.Few factors can be described as the conjectural buying and selling , geopolitical, OPEC output, increased demand from important role in the prediction of the oil prices. Now problem arising with the current **ANN** and **CNN** models that are used as prediction model’s are that they can’t provide accurate results when the data is too big. |
| 2. | Idea / Solution description | 1)**LSTM** clears about keeping the previous data and prediction which might be encouraging and more accurate. The possible results are comparatively inspiring.  2)The **LSTM** model will be updated whenever new oil price data are available, and provided to model, so the model continuously evolves over time, and can capture the changing pattern of oil prices. |
| 3. | Novelty / Uniqueness | 1)Price forecasting can assist in minimising the risks associated with volatility in oil prices.  2)Price forecasts are very important to various stakeholders: governments, public and private enterprises, policymakers, and investors. |
| 4. | Social Impact / Customer Satisfaction | 1)Brand activation  2) Innovative and schemes  3) Instant reward schemes  4) Personalized consumer purchase exchanges  5) Capability building of sales personnel |
| 5. | Business Model (Revenue Model) | 1)The price of crude oil should be easily predictable from the equilibrium between demand and supply.  2)Traders analyze demand and supply factors and take calculated positions. If their prediction comes true, traders close their position to book profits way before expiry.  3) price of crude oil are changeable based from time to time. |
| 6. | Scalability of the Solution | 1)hydrodynamic conditions in oilfield operations is suggested.  2)Modern refineries typically use a high number of sensors that generate an enormous amount of data.  3)Sustainable Solution for Crude Oil using Concentrated Solar Power Technology. |

**3.4 Problem Solution fit**

****

**4. REQUIREMENT ANALYSIS**

**4.1 Functional requirement**

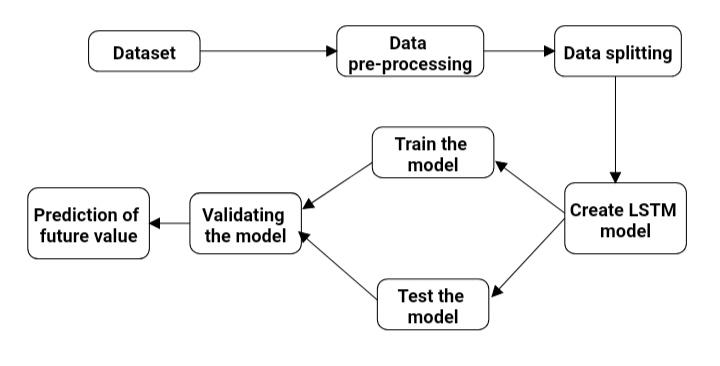
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| --- | --- | --- |
| FR-1 | **User Registration** | * Registration through Form * Registration through Gmail |
| FR-2 | **User Confirmation** | * Confirmation via Email * Confirmation via OTP |
| FR-3 | **Authentication** | * Verifying the identity of the user (ie)checking the email and password is correct. |
| FR-4 | **Authorization levels** | * User has been properly identified and authenticated. authorization levels determine the extent of system rights that the user has access to. |
| FR-5 | **Historical data management** | * Historical data to forecast future performance of the company. * Historical data includes your company's financial statements, client invoices and any information you believe has relative predictive value to the future success of your company. |

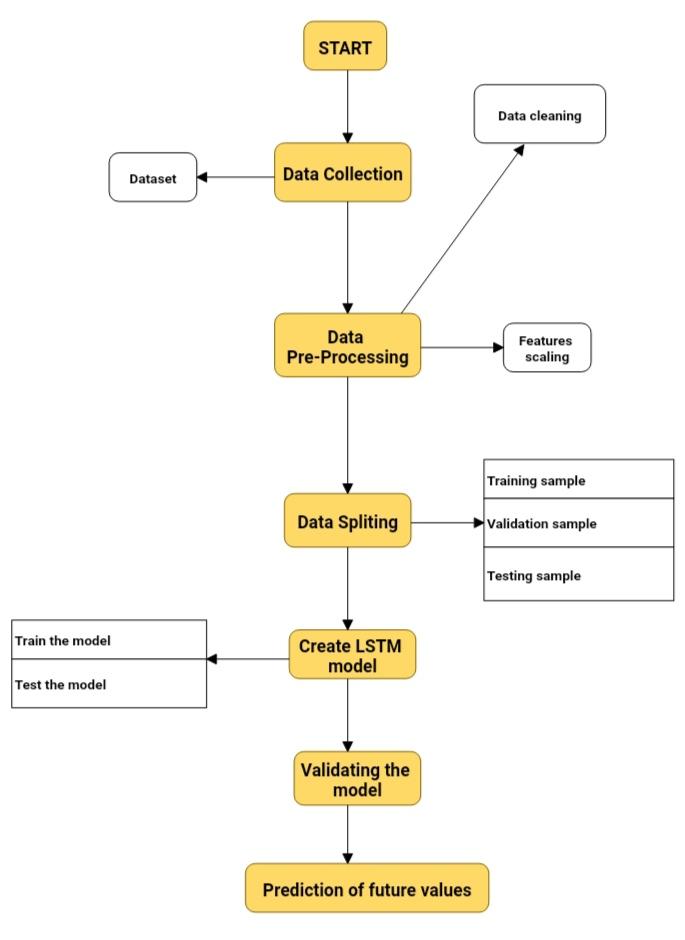
**4.2 Non-Functional requirements**

| **FR No.** | **Non-Functional Requirement** | **Description** |
| --- | --- | --- |
| NFR-1 | **Usability** | * Crude oil price fluctuations have a far reaching impact on global economies and thus price forecasting can assist in minimising the risks associated with volatility in oil prices. |
| NFR-2 | **Reliability** | * Price forecasts are very important to various stakeholders, governments, public and private enterprises, policymakers, and investors. |
| NFR-3 | **Performance** | * Using the LSTM model the accuracy of crude oil price prediction is increased. |
| NFR-4 | **Availability** | * Crude oil is not in infinite supply. After all, it took millions of years to "brew". Estimates vary, but if our current consumption continues apace, we may well see a time in the near future when it is completely exhausted. * Oil reserves are found all over the world. The top oil producing countries are Saudi Arabia, Russia, the United States, Iran, and China. |
| NFR-5 | **Scalability** | * Hydrodynamic conditions in oilfield operations is suggested. * Modern refineries typically use a high number of sensors that generate an enormous amount of data. * Sustainable Solution for Crude Oil using concentrated Solar Power Technology. |

**5. PROJECT DESIGN**

**5.1 Data Flow Diagrams**

****

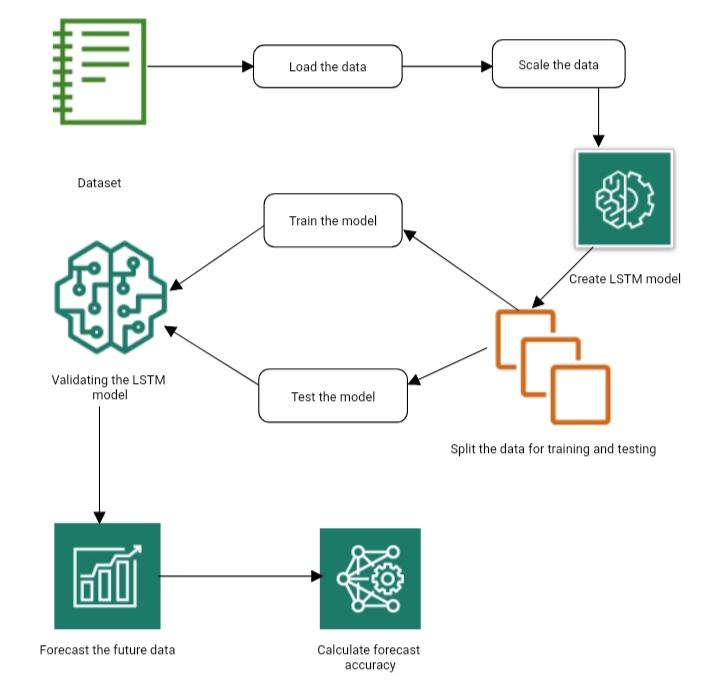
****

**5.2 Solution & Technical Architecture**

**Solution 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 problems in the organizations - LSTM technique.
* Describe the structure, characteristics, behavior, and other aspects of the software is move to LSTM for predicting more accurate Crude Oil Prices.
* Define features, development phases, and solution requirements of the project.
* Provide specifications according to which the solution is defined, managed, and delivered.

****

**5.3 User Stories**

| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| --- | --- | --- | --- | --- | --- | --- |
| Customer (People) | Registration | USN-1 | As a user can register for the application by entering my email, password, and confirming password. | I can access my account / dashboard | High | Sprint-1 |
|  |  | USN-2 | As a user will receive confirmation email once that have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
|  |  | USN-3 | As a user can register for the application through Gmail | I can register & access the dashboard with Facebook Login | Low | Sprint-2 |
|  | Login | USN-4 | As a user, I can log into the application by entering email & password |  | High | Sprint-1 |
|  | Authentication | USN-5 | Verifying the identity of the user (ie)checking the email and password is correct. |  |  |  |
|  | Authorization levels | USN-6 | As a user has been properly identified and authenticated. Authorization levels determine the extent of system rights that the user has access to. |  |  |  |
|  | Historical data management | USN-7 | Historical data to forecast future performance of the company. |  |  |  |
|  |  | USN-8 | Historical data includes your company's financial statements, client invoices and any information you believe has relative predictive value to the future success of your company. |  |  |  |
|  | Build the model | USN-9 | Build the model is trained and tested for prediction. |  |  |  |
| Administrator | Login | USN-1 | As an Administrator, I can login into the  analysis page. |  |  |  |
|  | Dashboard | USB-2 | As an Administrator, I can access the  Dashboard. |  |  |  |

**6. PROJECT PLANNING & SCHEDULING**

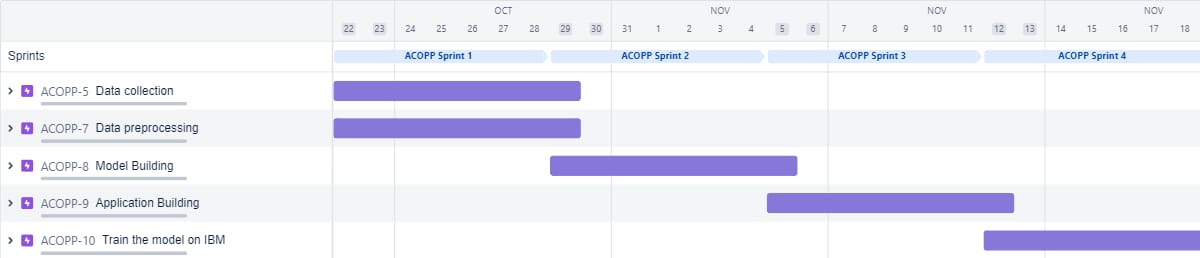
**6.1 Sprint Planning & Estimation**

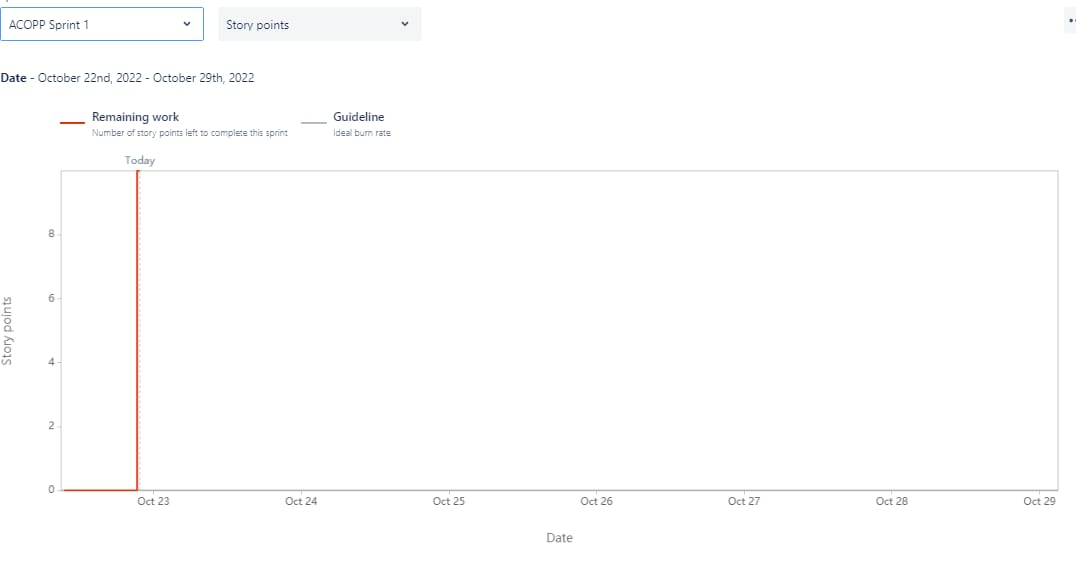
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | Data Collection | USN-1 | As a user,I can collect the dataset from kaggle. | 5 | High | Sridevi. J |
| Sprint-1 | Data Preprocessing | USN-2 | As a user,I can load the data set,handling the missing data, scaling and split data into train and test. | 5 | High | Sandhiya.M |
| Sprint-2 | Model Building | USN-3 | As a user,I can initialize the model,adding the LSTM layer and output layer,train,evaluate,save and test the model. | 10 | High | Yuvarani. R |
| Sprint-3 | Application Building | USN-4 | As a user,I create a HTML file,build a python code and run the app and showcasing the prediction. | 10 | High | Subulakshmi. S |
| Sprint-4 | Train the model on IBM | USN-5 | As a user,I train the model on IBM and integrate flask with scoring end point. | 10 | Medium | Yuvarani. R |

**6.2 Sprint Delivery Schedule**

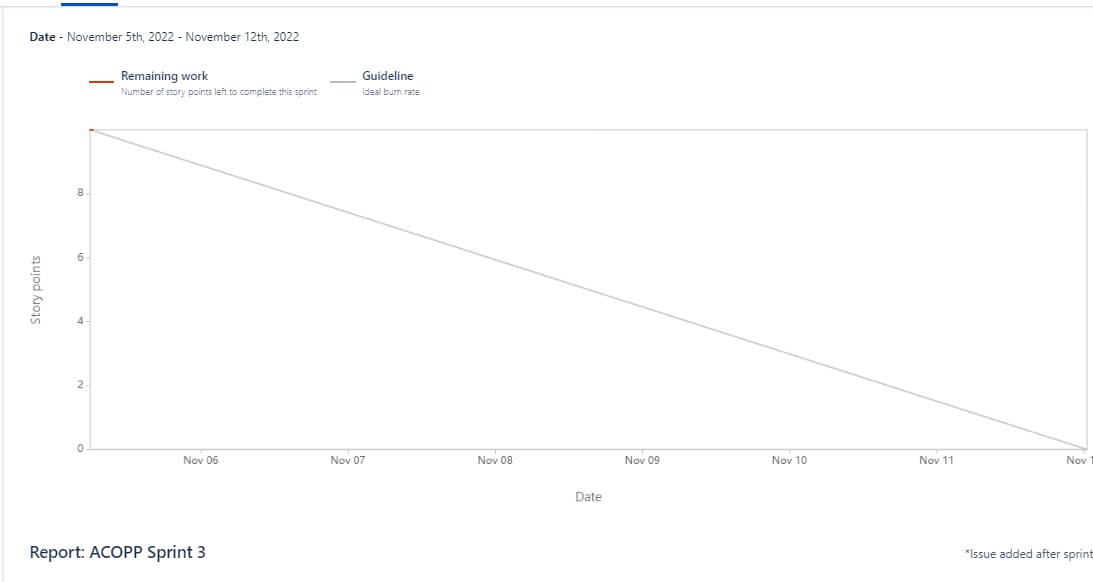
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | 10 | 5 Days | 24 Oct 2022 | 29 Oct 2022 | 10 | 29 Oct 2022 |
| Sprint-2 | 10 | 5 Days | 31 Oct 2022 | 05 Nov 2022 | 10 | 05 Nov 2022 |
| Sprint-3 | 10 | 5 Days | 07 Nov 2022 | 12 Nov 2022 | 10 | 12 Nov 2022 |
| Sprint-4 | 10 | 5 Days | 14 Nov 2022 | 19 Nov 2022 | 10 | 19 Nov 2022 |

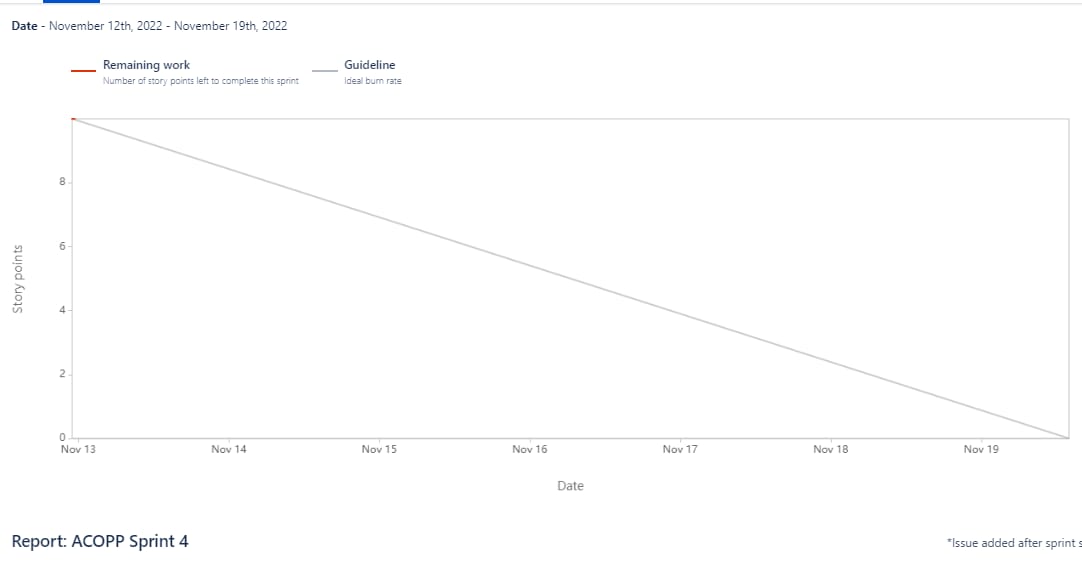
**6.3 Reports from JIRA**

****

****

****

****

****

**7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

**7.1 Feature 1**

**Data Preprocessing and model building:**

**Import Libraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

**unzip the file**

!unzip '/content/archive.zip**'**

**Load the dataset**

Data = pd.read\_excel('/content/Crude Oil Prices Daily.xlsx')

Data

**Handling Missing Data**

Data.isnull().any()

Data.isnull().sum()

Data.dropna(axis=0,inplace=True)

Data.isnull().sum()

data\_oil=Data.reset\_index()['Closing Value']

data\_oil

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature\_range=(0,1))

data\_oil=scaler.fit\_transform(np.array(data\_oil).reshape(-1,1))

**Data Visualization**

plt.plot(data\_oil)

**Split Data into Train and Test**

training\_size=int(len(data\_oil)\*0.65)

test\_size=len(data\_oil)-training\_size

train\_data,test\_data=data\_oil[0:training\_size,:],data\_oil[training\_size:len(data\_oil),:1]

training\_size,test\_size

train\_data.shape

**creating dataset with sliding windows**

def create\_dataset(dataset,time\_step=1):

dataX, dataY = [], []

for i in range(len(dataset)-time\_step-1):

a = dataset[i:(i+time\_step),0]

dataX.append(a)

dataY.append(dataset[i+time\_step,0])

return np.array(dataX), np.array(dataY)

time\_step = 10

X\_train, y\_train = create\_dataset(train\_data, time\_step)

X\_test, ytest = create\_dataset(test\_data, time\_step)

print(X\_train.shape), print(y\_train.shape)

print(X\_test.shape), print(ytest.shape)

X\_train

X\_train = X\_train.reshape(X\_train.shape[0],X\_train.shape[1] , 1)

X\_test = X\_test.reshape(X\_test.shape[0],X\_test.shape[1] , 1)

**Import the Model Building Libraries**

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense

**Initializing model**

model= Sequential()

**Adding LSTM Layers**

model.add(LSTM(50,return\_sequences=True,input\_shape=(10,1)))

model.add(LSTM(50,return\_sequences=True))P

model.add(LSTM(50))

**Adding Output Layers**

model.add(Dense(1))

model.summary()

**Configure The Learning Process**

model.compile(loss='mean\_squared\_error',optimizer='adam')

**Train The Model**

model.fit(X\_train,y\_train,validation\_data=(X\_test,ytest),epochs=50,batch\_size=64,verbose=1)

**Model Evaluation**

import tensorflow as tf

train\_predict = model.predict(X\_train)

test\_predict = model.predict(X\_test)

train\_predict=scaler.inverse\_transform(train\_predict)

test\_predict=scaler.inverse\_transform(test\_predict)

import math

from sklearn.metrics import mean\_squared\_error

math.sqrt(mean\_squared\_error(y\_train,train\_predict))

math.sqrt(mean\_squared\_error(ytest,test\_predict))

**Save the Model**

from tensorflow.keras.models import load\_model

model.save("crude\_oil.h5")

**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(scaler.inverse\_transform(data\_oil))

plt.plot(trainPredictPlot,label="traindata")

plt.plot(testPredictPlot,label="testdata")

plt.show()

print("Green indicates predicated data")

print("Blue indicates complete data")

print("Orange indicates train data")

len(test\_data)

X\_input=test\_data[2866:].reshape(1,-1)

X\_input.shape

temp\_input=list(X\_input)

temp\_input=temp\_input[0].tolist()

temp\_input

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))

X\_input=X\_input.reshape(1,-1)

X\_input=X\_input.reshape((1,n\_steps,1))

yhat=model.predict(X\_input, verbose=0)

print("{} Day output {}".format(i,yhat))

temp\_input.extend(yhat[0].tolist())

temp\_input=temp\_input[1:]

lst\_output.extend(yhat.tolist())

i=i+1

else:

X\_input=X\_input.reshape((1,n\_steps,1))

yhat=model.predict(X\_input, verbose=0)

print(yhat[0])

temp\_input.extend(yhat[0].tolist())

print(len(temp\_input))

lst\_output.extend(yhat.tolist())

i=i+1

day\_new=np.arange(1,11)

day\_pred=np.arange(11,21)

len(data\_oil)

plt.plot(day\_new,scaler.inverse\_transform(data\_oil[8206:]))

plt.plot(day\_pred,scaler.inverse\_transform(lst\_output))

df3=data\_oil.tolist()

df3.extend(lst\_output)

plt.plot(df3[8100:])

df3=scaler.inverse\_transform(df3).tolist()

plt.plot(df3)

**7.2 Feature 2**

**Application building:**

**Index. Html:**

<!DOCTYPE html>

<html>

<head>

<title>Home</title>

<style>

body

{

background-image:url("https://thumbs.gfycat.com/FluidCorruptArabianoryx-size\_restricted.gif");

background-size: cover;

}

.pd{

padding-bottom:100%;}

.navbar

{

margin: 0px;

padding:20px;

background-color:white;

opacity:0.6;

color:black;

font-family:'Roboto',sans-serif;

font-style: italic;

border-radius:20px;

font-size:25px;

}

a

{

color:grey;

float:right;

text-decoration:none;

font-style:normal;

padding-right:20px;

}

a:hover{

background-color:black;

color:white;

border-radius:15px;0

font-size:30px;

padding-left:10px;

}

p

{

color:turqouise;

font-style:italic;

font-size:30px;

}

</style>

</head>

<body>

<div class="navbar">

<a href="/predict" >Predict</a>

<a href="/about">Home</a>

<br>

</div>

<br>

<center><b class="pd"><font color="white" size="15" font-family="Comic Sans MS" >Crude Oil Prediction</font></b></center><br><br>

<div>

<br>

<center>

<p><font color="white">Demand for oil 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.</p>

</center>

</div>

</body>

</html>

**Web. Html:**

<html>

<style>

div.header{

top: 0;

position: fixed;

padding-left: 400px;}

div.header1{

top:20;

position: fixed;

padding-left: 490px;

}

\*{

margin:0;

padding:0;

border:0;

outline:0;

text-decoration:none;

font-family:montserrat;

}

body

{

background-image:url('https://img.freepik.com/free-vector/trading-background-with-graphic-illustration-blue-heart-rate-that-is-rising-upwards\_159711-165.jpg?size=626&ext=jpg');

background-position: center;

font-family:sans-serif;

background-size:cover;

margin-top:40px;

}

.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"]{

border:0;

background:none;

display:block;

margin:20px auto;

text-align:center;

border:2px solid #800080;

padding:15px 3px;

width:400px;

outline:none;

color:white;

border-radius:100px;

transition:0.25s;

font-size:20;

}

.bor{

border:0;

background:none;

display:block;

margin:20px auto;

text-align:center;

border:2px solid #800080;

padding:10px 3px;

width:500px;

outline:none;

color:white;

transition:0.25s;}

.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus{

width:280px;

border-color:#8e44ad;

}

.logbtn{

display:block;

width:35%;

height:50px;

border:none;

border-radius:24px;

background:linear-gradient(120deg,#3498db,#8e44ad,#3498db,#8e44ad);

background-size:200%;

color:#fff;

outline:none;

cursor:pointer;

transition:.5s;

font-size:25;

}

.logbtn:hover{

background-center;

}

input::placeholder{

color:#F5FFFA;

}

.bottom-text{

margin-top:60px;

text-align:center;

font-size:13px;

}

</style>

<body>

<div class="navbar">

<a href="/predict" >Predict</a>

<a href="/about">Home</a>

<br>

</div>

<center><div><font color="Powderblue" font-family="sans-serif" size=8 ><b>Crude Oil Prediction</b></font></div></center>

<br><br><br><br>

<form class="main" action="/login" method="post">

<br>

<font size=10><input type="text" name="year" placeholder="Enter the crude oil prices for the first 10 days"/></font>

<center><input type="submit" class="logbtn" value="Predict"></center>

<div class="bor"><b><font color="white" size=5>{{showcase}}</font></b</div>

</form>

</div>

</body>

</html>

**app. py:**

import numpy as np

from flask import Flask, render\_template, request

from tensorflow.keras.models import load\_model

app = Flask(\_\_name\_\_)

model = load\_model('crude\_oil.h5')

@app.route('/')

def home():

return render\_template("index.html")

@app.route('/about')

def home1():

return render\_template("index.html")

@app.route('/predict')

def home2():

return render\_template("web.html")

@app.route('/login', methods=['POST'])

def login():

x\_input = str(request.form['year'])

x\_input = x\_input.split(',')

temp\_input = [eval for i in x\_input]

x\_input = np.zeros(shape=(1, 10))

lst\_output = []

n\_steps = 10

i = 0

while i < 1:

if len(temp\_input) > 10:

x\_input = np.array(temp\_input[1:])

print("{} day input {}".format(i, x\_input))

x\_input = x\_input.reshape(1, -1)

x\_input = x\_input.reshape((1, n\_steps, 1))

yhat = model.predict(x\_input, verbose=0)

print("{} day output {}".format(i, yhat))

temp\_input.extend(yhat[0].tolist())

temp\_input = temp\_input[1:]

lst\_output.extend(yhat.tolist())

i = i+1

else:

x\_input = x\_input.reshape((1, n\_steps, 1))

yhat = model.predict(x\_input, verbose=0)

print(yhat[0])

temp\_input.extend(yhat[0].tolist())

print(len(temp\_input))

lst\_output.extend(yhat.tolist())

i = i+1

print(lst\_output)

return render\_template("web.html", showcase='The next day predicted value is:' + str(lst\_output))

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True, port=5000)

**7.3 Database Schema (if Applicable)**

**Import Libraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

**unzip the file**

!unzip '/content/archive.zip**'**

**Load the dataset**

Data = pd.read\_excel('/content/Crude Oil Prices Daily.xlsx')

Data

**Handling Missing Data**

Data.isnull().any()

Data.isnull().sum()

Data.dropna(axis=0,inplace=True)

Data.isnull().sum()

data\_oil=Data.reset\_index()['Closing Value']

data\_oil

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature\_range=(0,1))

data\_oil=scaler.fit\_transform(np.array(data\_oil).reshape(-1,1))

**Data Visualization**

plt.plot(data\_oil)

**Split Data into Train and Test**

training\_size=int(len(data\_oil)\*0.65)

test\_size=len(data\_oil)-training\_size

train\_data,test\_data=data\_oil[0:training\_size,:],data\_oil[training\_size:len(data\_oil),:1]

training\_size,test\_size

train\_data.shape

**creating dataset with sliding windows**

def create\_dataset(dataset,time\_step=1):

dataX, dataY = [], []

for i in range(len(dataset)-time\_step-1):

a = dataset[i:(i+time\_step),0]

dataX.append(a)

dataY.append(dataset[i+time\_step,0])

return np.array(dataX), np.array(dataY)

time\_step = 10

X\_train, y\_train = create\_dataset(train\_data, time\_step)

X\_test, ytest = create\_dataset(test\_data, time\_step)

print(X\_train.shape), print(y\_train.shape)

print(X\_test.shape), print(ytest.shape)

X\_train

X\_train = X\_train.reshape(X\_train.shape[0],X\_train.shape[1] , 1)

X\_test = X\_test.reshape(X\_test.shape[0],X\_test.shape[1] , 1)

**Import the Model Building Libraries**

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense

**Initializing model**

model= Sequential()

**Adding LSTM Layers**

model.add(LSTM(50,return\_sequences=True,input\_shape=(10,1)))

model.add(LSTM(50,return\_sequences=True))

model.add(LSTM(50))

**Adding Output Layers**

model.add(Dense(1))

model.summary()

**Configure The Learning Process**

model.compile(loss='mean\_squared\_error',optimizer='adam')

**Train The Model**

model.fit(X\_train,y\_train,validation\_data=(X\_test,ytest),epochs=50,batch\_size=64,verbose=1)

**Model Evaluation**

import tensorflow as tf

train\_predict = model.predict(X\_train)

test\_predict = model.predict(X\_test)

train\_predict=scaler.inverse\_transform(train\_predict)

test\_predict=scaler.inverse\_transform(test\_predict)

import math

from sklearn.metrics import mean\_squared\_error

math.sqrt(mean\_squared\_error(y\_train,train\_predict))

math.sqrt(mean\_squared\_error(ytest,test\_predict))

**Save the Model**

from tensorflow.keras.models import load\_model

model.save("crude\_oil.h5")

!pip install watson-machine-learning-client

!pip install ibm\_watson\_machine\_learning

from ibm\_watson\_machine\_learning import APIClient

wml\_credentials = {

"url":"https://eu-gb.ml.cloud.ibm.com",

"apikey":"KaV\_3TkCT6UHi6BB8P1qBqvpDKNkCLLor76vWllPRdCz"

}

Client = APIClient(wml\_credentials)

Client

Client.spaces.get\_details()

Client.spaces.list()

space\_uid = "7c472a47-2d81-4767-bac5-e45218390e87"

space\_uid

Client.set.default\_space(space\_uid)

Client.software\_specifications.list()

software\_space\_uid = Client.software\_specifications.get\_uid\_by\_name("tensorflow\_rt22.1-py3.9")

software\_space\_uid

model\_details = Client.repository.store\_model(model="crude-oil-price-prediction.tgz", meta\_props={

Client.repository.ModelMetaNames.NAME:"crude oil price prediction model",

Client.repository.ModelMetaNames.TYPE:"tensorflow\_2.7",

Client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID:software\_space\_uid

})

model\_details

model\_id = Client.repository.get\_model\_id(model\_details)

model\_id

Client.repository.download(model\_id,'CRUDE OIL PRICE PREDICTION\_IBM\_MODEL.tgz')

**8. TESTING**

**8.1 Test Cases**

1.To view Crude oil home page,home and predict navigation bar

2.To view the box for entering crude oil prices for first ten days

3.To view the predicted price for first ten days

**8.2 User Acceptance Testing**

# **Purpose of Document**

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

# **Defect Analysis**

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

| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Subtotal** |
| --- | --- | --- | --- | --- | --- |
| By Design | 10 | 4 | 2 | 3 | 20 |
| Duplicate | 0 | 5 | 8 | 0 | 43 |
| External | 2 | 3 | 0 | 1 | 6 |
| WFixed | 13 | 9 | 4 | 20 | 37 |
| Not Reproduced | 5 | 0 | 7 | 0 | 1 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 56 | 5 | 2 | 0 | 8 |
| Totals | 20 | 14 | 28 | 26 | 64 |

# **3.Test Case Analysis**

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

| **Section** | **Total Cases** | **Not Tested** | **Fail** | **Pass** |
| --- | --- | --- | --- | --- |
| Print Engine | 12 | 0 | 0 | 12 |
| Client Application | 12 | 0 | 0 | 12 |
| Security | 12 | 0 | 0 | 12 |
| Outsource Shipping | 12 | 0 | 0 | 12 |
| Exception Reporting | 3 | 0 | 0 | 3 |
| Final Report Output | 3 | 0 | 0 | 3 |
| Version Control | 3 | 0 | 1 | 2 |

**9. RESULTS**

**9.1 Performance Metrics**

This paper that pastforecasting performance is not a good indicator for future performance as oil prices tend to behave very differently over time. For example, it might be that if oil prices remain relatively stable as they were in the 1990s, the futures-based forecast is the best forecast method available. Conditional upon the fact, however, that we do not know how oil prices will behave in the future, the forecast combination will shield better against structural changes compared to stand-alone models. A central banker will have to re-evaluate the forecasting performance of the preferred model given recent oil price behavior, taking into account what a specific forecasting approach can capture and what not.

**10. ADVANTAGES & DISADVANTAGES**

**Advantages:**

1)Crude oil price fluctuations have a far reaching impact on global economies and thus price forecasting can assist in minimising the risks associated with volatility in oil prices.

2) Crude oil is amongst the most important resources in today’s world, it is the chief fuel and its cost has a direct effect on the global habitat, our economy and oil exploration, exploitation and other activities.

3) It is powerful and highly suggested because investors can use it not only to initiate trades but also as an effective tool to judge various strategies relating to investment.

**Disadvantages**:

1)Producing electricity from crude oil is expensive compared to other fossil fuels such as coal or gas.

2)Burning oil produces carbon dioxide gas. This is a greenhouse gas that contributes towards climate change and Burning oil can pollute the air.

3)Other environmental impacts include intensification of the greenhouse effect, acid rain, poorer water quality, groundwater contamination, among others. The oil and gas industry may also contribute to biodiversity loss as well as to the destruction of ecosystems.

**11. CONCLUSION:**

This paper has clears that an LSTM network is better than other traditional neural network for forecasting prices as it aims in using back propagation model. Traditional neural network such as CNN on the other hand predicts the next outgoing but doesn’t necessarily save the previous data or connection which is based on feed-forwarding, in the sense the previous data is not necessary to predict the future data. LSTM focuses on storing the previous data and prediction which is rather encouraging and more approximate. The outcome derived are relatively encouraging. The results show that large look ups do not necessarily improve the accuracy of the predictions of crude oil prices. Hence it can be concluded, the model with single LSTM model is definitely the most accurate.

**12. FUTURE SCOPE**

The proposed model is powerful and highly suggested because investors can use it not only to initiate trades but also as an effective tool to judge various strategies relating to investments. This work is carried out on the closing price of crude oil; however there are various other factors which also affect the crude oil prices like change in the prices and quantities (demand and supply), change in the economy and current affairs as shown by the media. The main advantages of this research is in capturing the changing pattern of these prices. In the coming future, fundamentals indicators and market trends have been planned to be incorporated into a model which will help the proposed model perform more efficiently.

**13. APPENDIX**

The repository for the code used to generate the evaluation of this article will be made available in a GitHub repository.

**Source Code**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

!unzip '/content/archive.zip**'**

Data = pd.read\_excel('/content/Crude Oil Prices Daily.xlsx')

Dat

Data.isnull().any()

Data.isnull().sum()

Data.dropna(axis=0,inplace=True)

Data.isnull().sum()

data\_oil=Data.reset\_index()['Closing Value']

data\_oil

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature\_range=(0,1))

data\_oil=scaler.fit\_transform(np.array(data\_oil).reshape(-1,1))

plt.plot(data\_oil)

training\_size=int(len(data\_oil)\*0.65)

test\_size=len(data\_oil)-training\_size

train\_data,test\_data=data\_oil[0:training\_size,:],data\_oil[training\_size:len(data\_oil),:1]

training\_size,test\_size

train\_data.shape

def create\_dataset(dataset,time\_step=1):

dataX, dataY = [], []

for i in range(len(dataset)-time\_step-1):

a = dataset[i:(i+time\_step),0]

dataX.append(a)

dataY.append(dataset[i+time\_step,0])

return np.array(dataX), np.array(dataY)

time\_step = 10

X\_train, y\_train = create\_dataset(train\_data, time\_step)

X\_test, ytest = create\_dataset(test\_data, time\_step)

print(X\_train.shape), print(y\_train.shape)

print(X\_test.shape), print(ytest.shape)

X\_train

X\_train = X\_train.reshape(X\_train.shape[0],X\_train.shape[1] , 1)

X\_test = X\_test.reshape(X\_test.shape[0],X\_test.shape[1] , 1)

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense

model= Sequential()

model.add(LSTM(50,return\_sequences=True,input\_shape=(10,1)))

model.add(LSTM(50,return\_sequences=True))

model.add(LSTM(50))

model.add(Dense(1))

model.summary()

model.compile(loss='mean\_squared\_error',optimizer='adam')

model.fit(X\_train,y\_train,validation\_data=(X\_test,ytest),epochs=50,batch\_size=64,verbose=1)

import tensorflow as tf

train\_predict = model.predict(X\_train)

test\_predict = model.predict(X\_test)

train\_predict=scaler.inverse\_transform(train\_predict)

test\_predict=scaler.inverse\_transform(test\_predict)

import math

from sklearn.metrics import mean\_squared\_error

math.sqrt(mean\_squared\_error(y\_train,train\_predict))

math.sqrt(mean\_squared\_error(ytest,test\_predict))

from tensorflow.keras.models import load\_model

model.save("crude\_oil.h5")

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(scaler.inverse\_transform(data\_oil))

plt.plot(trainPredictPlot,label="traindata")

plt.plot(testPredictPlot,label="testdata")

plt.show()

print("Green indicates predicated data")

print("Blue indicates complete data")

print("Orange indicates train data")

len(test\_data)

X\_input=test\_data[2866:].reshape(1,-1)

X\_input.shape

temp\_input=list(X\_input)

temp\_input=temp\_input[0].tolist()

temp\_input

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))

X\_input=X\_input.reshape(1,-1)

X\_input=X\_input.reshape((1,n\_steps,1))

yhat=model.predict(X\_input, verbose=0)

print("{} Day output {}".format(i,yhat))

temp\_input.extend(yhat[0].tolist())

temp\_input=temp\_input[1:]

lst\_output.extend(yhat.tolist())

i=i+1

else:

X\_input=X\_input.reshape((1,n\_steps,1))

yhat=model.predict(X\_input, verbose=0)

print(yhat[0])

temp\_input.extend(yhat[0].tolist())

print(len(temp\_input))

lst\_output.extend(yhat.tolist())

i=i+1

day\_new=np.arange(1,11)

day\_pred=np.arange(11,21)

len(data\_oil)

plt.plot(day\_new,scaler.inverse\_transform(data\_oil[8206:]))

plt.plot(day\_pred,scaler.inverse\_transform(lst\_output))

df3=data\_oil.tolist()

df3.extend(lst\_output)

plt.plot(df3[8100:])

df3=scaler.inverse\_transform(df3).tolist()

plt.plot(df3)

**GitHub & Project Demo Link**

https://drive.google.com/file/d/1ZwfDKVF8wa\_sNc0P2OF1Pu6NccpEWkEU/view?usp=drivesdk