PROJECT REPORT

CRUDE OIL PRICE PREDICTION

Submitted by PNT2022TMID52700

Darshini B - CITC1907006

Deveshwar R- CITC1907008

Kausik Subramanaiyam G - CITC1907019

Shunmuga Sundaram.K – CITC2007206

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

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

The purpose of this project is to help the society use the crude oil In an efficient way and also to help the vendors and buyers in the crude oil industry by providing them with an user friendly platform so that they will be able to know about the immediate future through which they will be able to make decisions wisely and avoid wastage/loss in business as well as resources. This can also lead to the reduction in scarcity of the resources since it is utilised efficiently.

CHAPTER 2 – LITERATURE SURVEY

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

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 and improve its environmental footprint.The oil and gas sector in Kenya faces various challenges such as inadequate capacity, weak institutional framework, revenue management, weak bargaining power, lack of transparency and accountability. However these challenges can be strengthened through reinforcement.

2.2 REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] Manel Hamdi and Chaker Aloui, "Forecasting Crude Oil Price Using Artificial Neural Networks: A Literature Survey," Economics Bulletin, AccessEcon, vol. 35, no. 2, pp. 1339-1359, 2015.

[5] Aloui, Chaker & Hamdi, Manel. (2015). Forecasting Crude Oil Price Using Artificial Neural Networks: A Literature Survey. Economics Bulletin. 35. 1339-1359.

2.3 PROBLEM STATEMENT DEFINITION

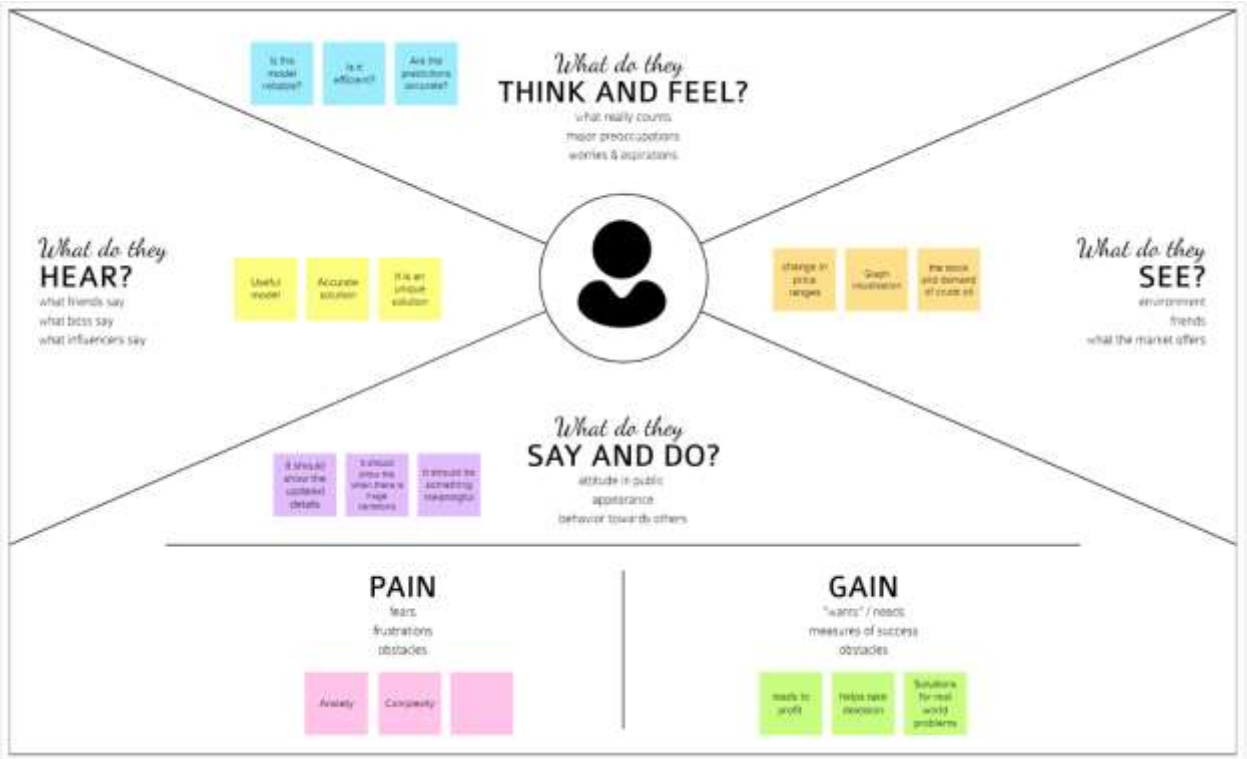
The Government, private and public enterprise, policymakers and investors are mainly dependent on the price fluctuations and it is an very important aspect to them. Thus the crude oil price causes a global economic impact and it can be prevented by predicting the price which minimises the risks associated with the volatility in oil prices. Petrol bunk owners, Crude oil investors , Investors who invest on products the runs or depends on crude oil are the ones who are the most affected. Crude oil price changes every day. It depends on many factors. Sudden fluctuations in the crude oil can result in major changes. So, prediction of the crude oil price can help to cut the loss percentage, and increase the profit margin. It can

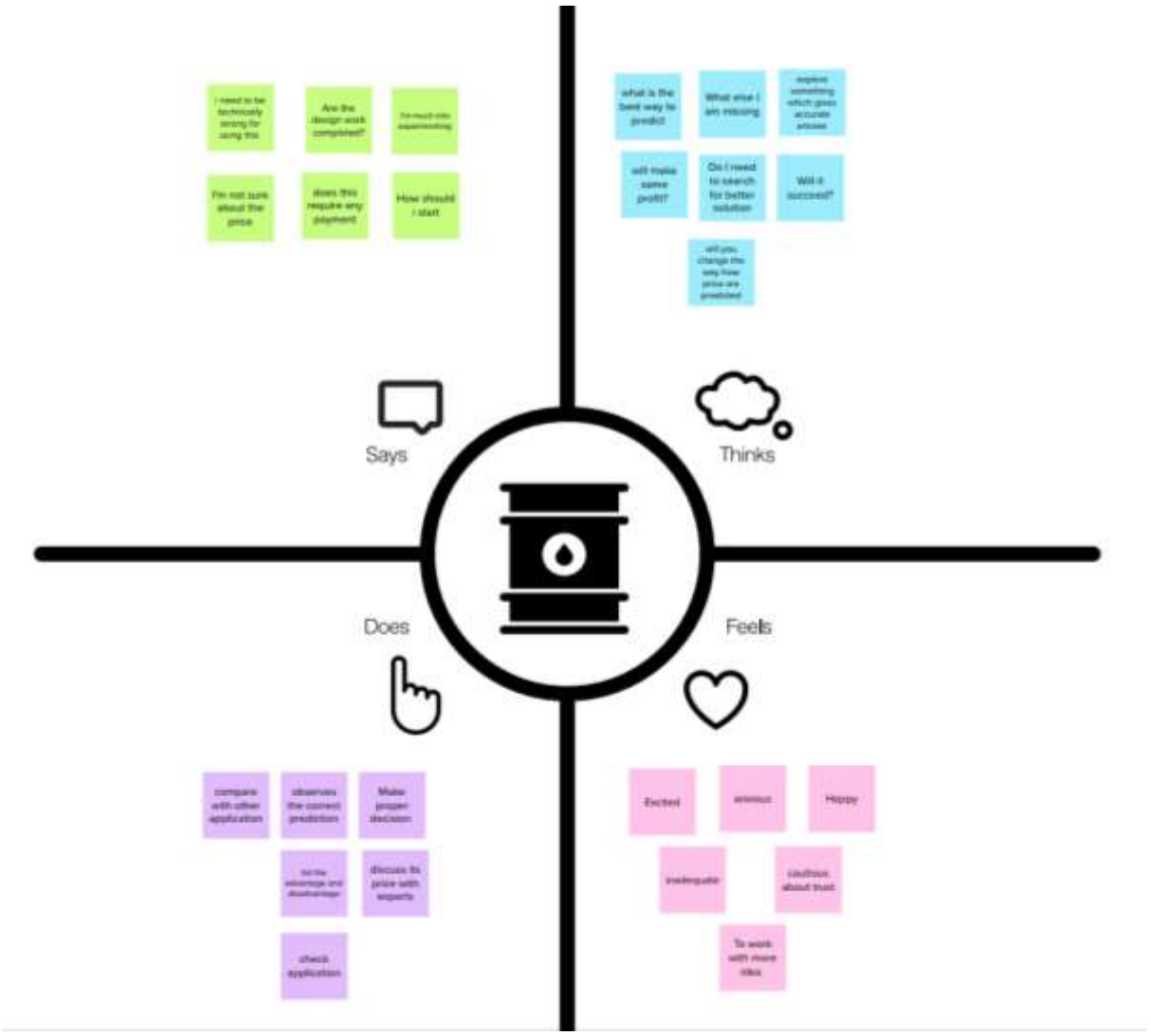
also help to overcome the demand supply issues. This issue occurs in the investment related areas and also affects the product development and functioning that depends on crude oil. Solving this problem can increase the profits from investment. It may not only help individuals but also can affect national level income. Machine learning and deep learning algorithms can be used in order to provide the solution for this problem.

CHAPTER 3 – IDEATION AND PROPOSED SOLUTION

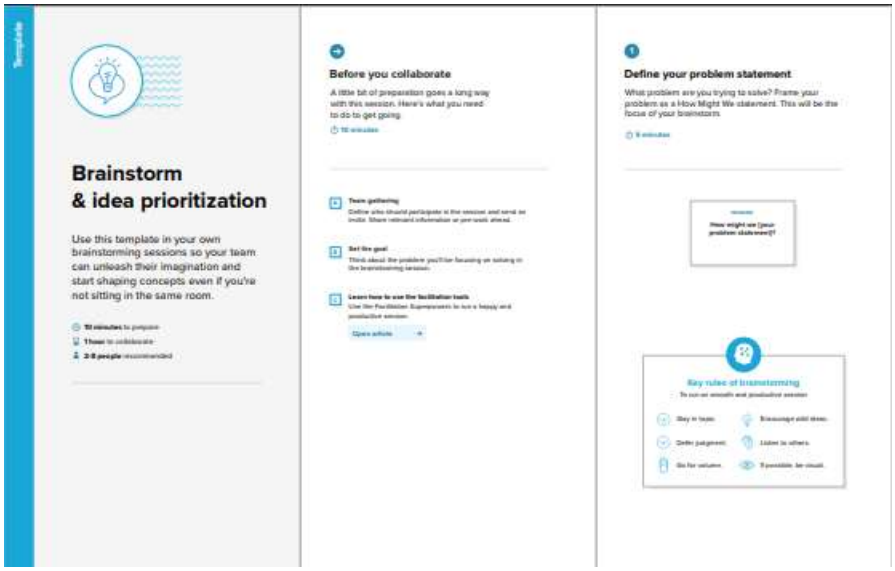
3.IDEATION AND PROPOSED SOLUTION

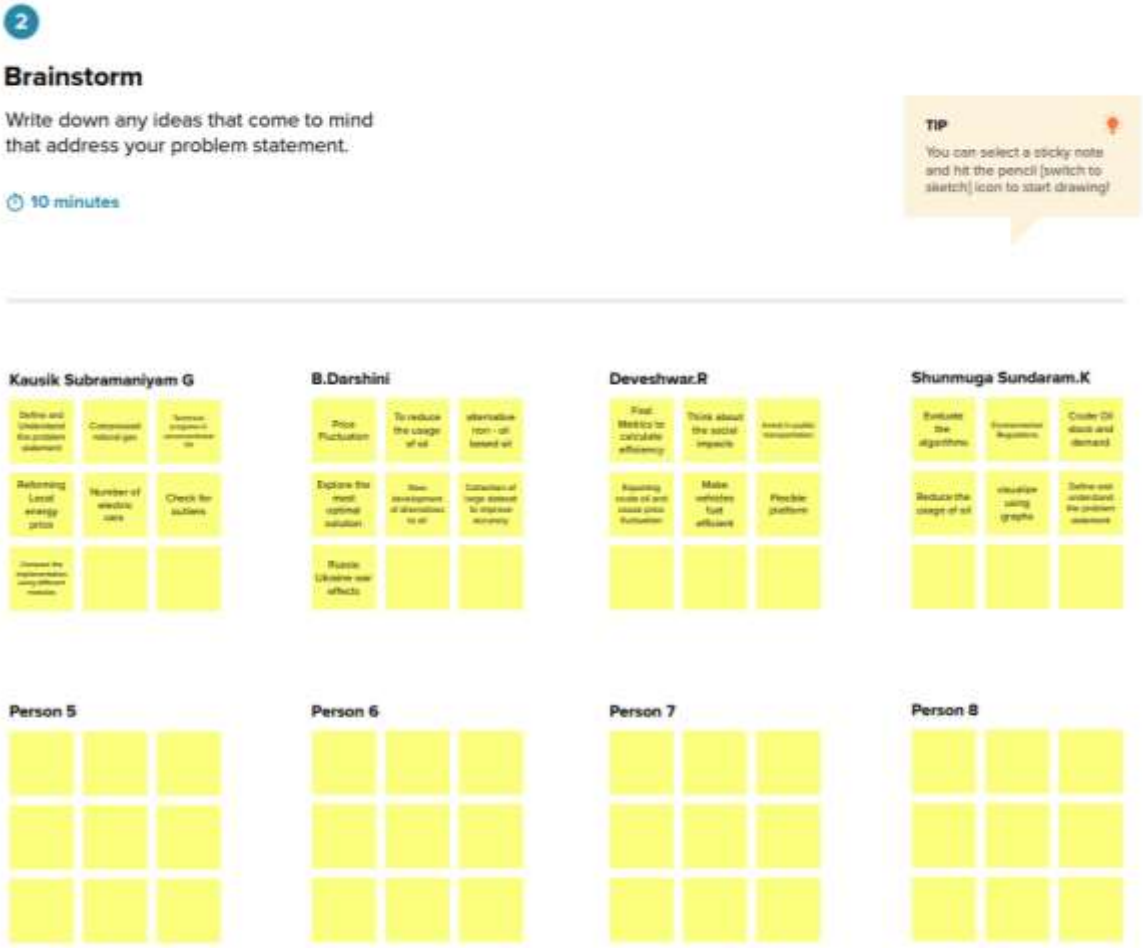
3.1 EMPATHY MAP CANVAS

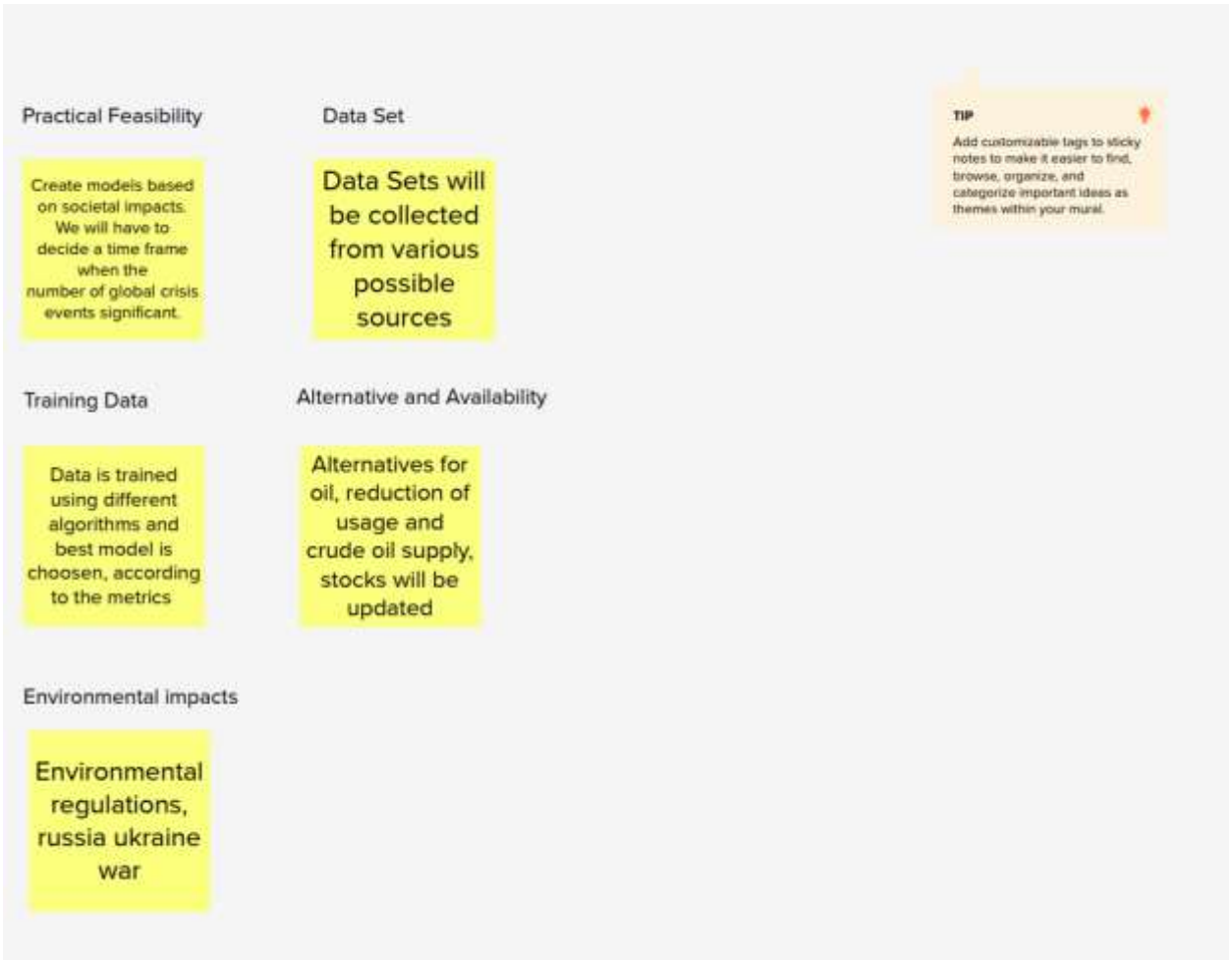


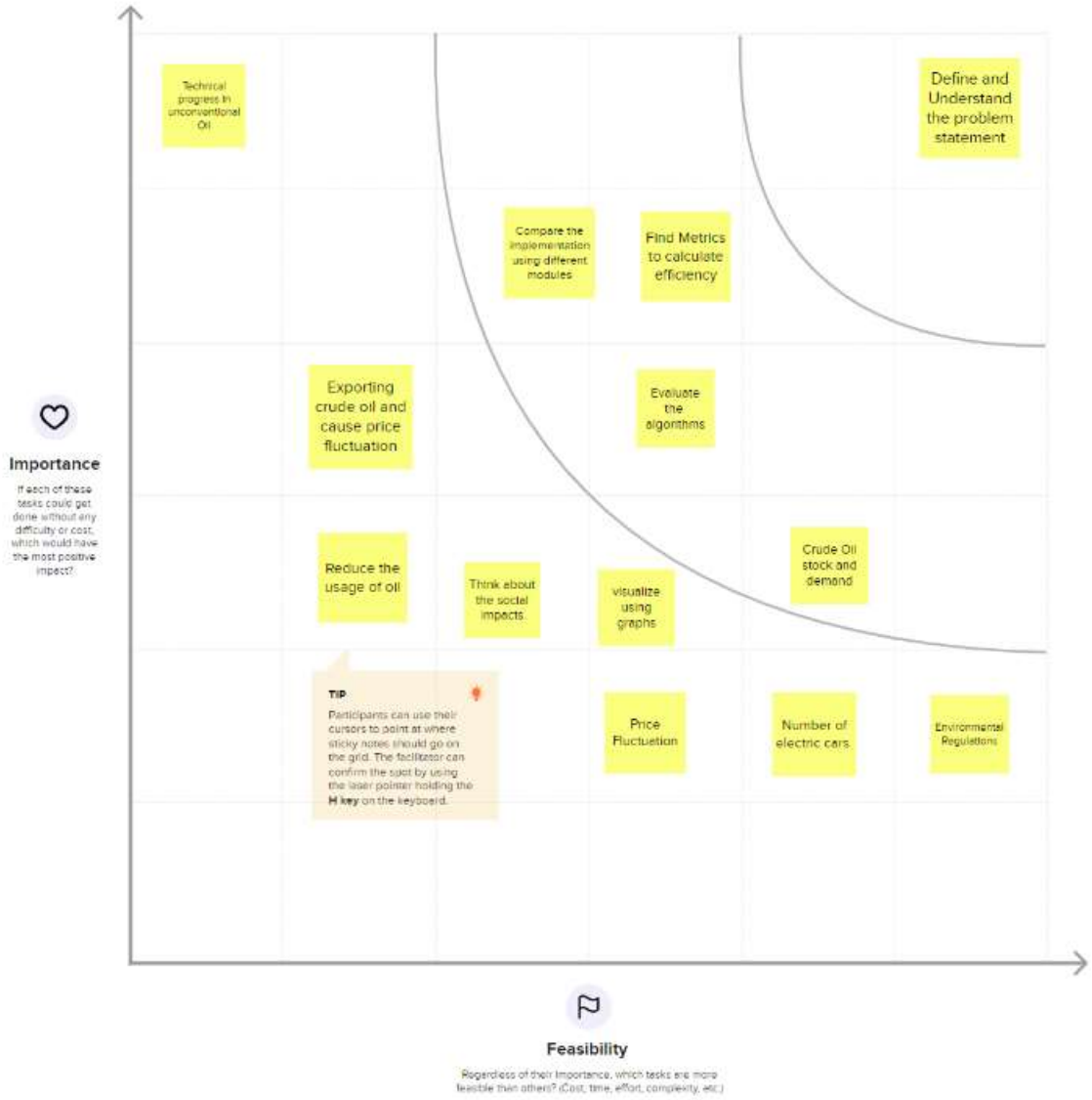

Says
Thinks
Does
Feels

3.2 IDEATION & BRAINSTORMING






Practical Feasibility
Create models based on societal impacts. We will have to decide a time frame when the number of global crisis events significant.
Data Set
Data Sets will be collected from various possible sources
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
Training Data
Data is trained using different algorithms and best model is choosen, according to the metrics
Alternative and Availability
Alternatives for oil, reduction of usage and crude oil supply, stocks will be updated
Environmental impacts
Environmental regulations, russia ukraine war

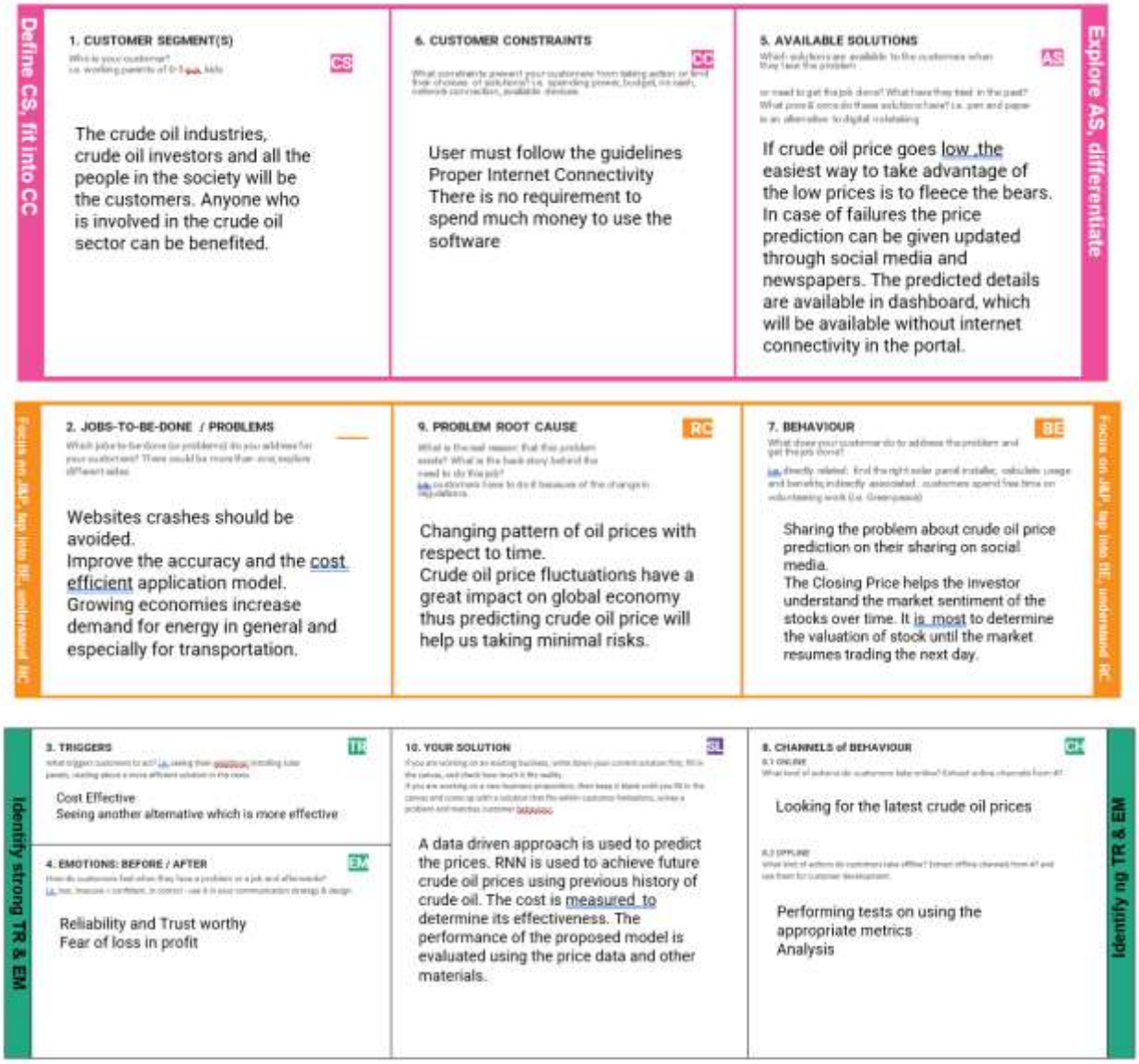


3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Crude oil is one the most important need in day to day life, and its prices have a big impact on the global environment and the prediction of prices will be useful to government, industry and individuals. The continuous usage of statistical and econometric techniques including AI for crude oil price prediction might demonstrate demotions to the prediction performance. The price might the used to be the

		time analysis prediction which will help in buying the crude oil at the best time period. This is enable the industries to increase the profit rate.
2.	Idea / Solution description	A data driven approach is used to predict the prices. RNN is used to achieve future crude oil prices using previous history of crude oil. The cost is measured to determine its effectiveness. The performance of the proposed model is evaluated using the price data and other materials.
3.	Novelty / Uniqueness	Building an user friendly interface, to enable any kind of user to make use of the application. The prediction will be made not only based on the prices but also considering other factors affecting the price fluctuation. Application will help the industries and the government benefit in many ways.
4.	Social Impact / Customer Satisfaction	By predicting the crude oil prices it will help the users be prepared for uncertainties and improve the performance. It will also help to stabilise the economy. Price has direct effect on several goods and products and its fluctuations affect the stock markets. The prices are also affected by other factors.
5.	Business Model (Revenue Model)	These days the business using supply chain rely on crude oil for transportation. So the prediction of crude oil prices will help the firms increase the profit rate and also result in growth of the firm. It can help decision makers such as firms, private investors, or individuals chose to buy or sell the crude oil . Crude oil is one of the most profitable trading commodities for traders. RNN and LSTM models are used as the key model to predict the crude oil prices.
6.	Scalability of the Solution	User friendly application to enable users to access from anywhere. Algorithms and methods are used to improve the accuracy and prediction.

3.4 PROBLEM SOLUTION FIT



CHAPTER 4 – REQUIREMENT ANALYSIS

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 Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Additional Features	User Can Read Latest information And View Oil Price User friendly records and history of prices are visible
FR-4	User Exceptions	User Can Exchange Rates And convert currencies
FR-5	User Support	FAQ's and any other queries raised will be answered
FR-6	Database	The Information given by the user is stored
FR-7	Notifications	Daily updates and notifications are sent to the user
FR-8	Generating Results	Oil Price Prediction

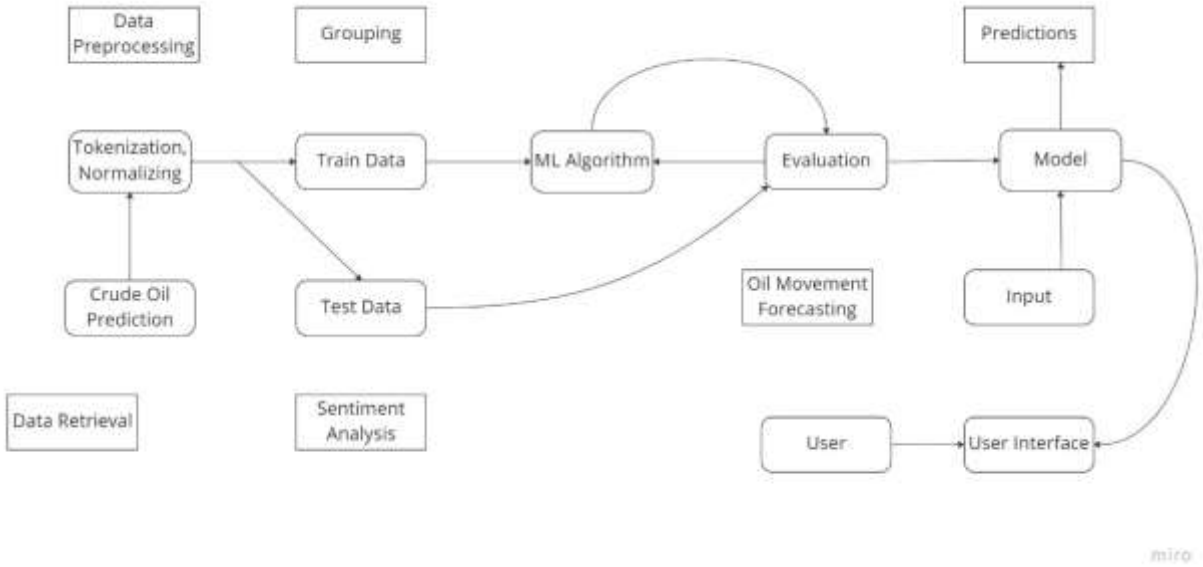
4.2 NON-FUNCTIONAL REQUIREMENT

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is easy to use for a client and the portal will be user friendly. It can be used by clients of any age group
NFR-2	**Security**	Sensitive data is being protected and the login information is hashed. Some data require authentication in order to view
NFR-3	**Reliability**	The data will be reliable since it is updated on the day to day basis with accurate information from multiple resources.
NFR-4	**Performance**	The portal can be used at a low bandwidth network connection in a hassle free manner. Fast and smooth working of the portal is ensured.
NFR-5	**Availability**	The history of the previous predictions will be available, updates are also sent regularly.
NFR-6	**Scalability**	The storage space can be expanded and the required data can be given on demand

CHAPTER 5 – PROJECT DESIGN

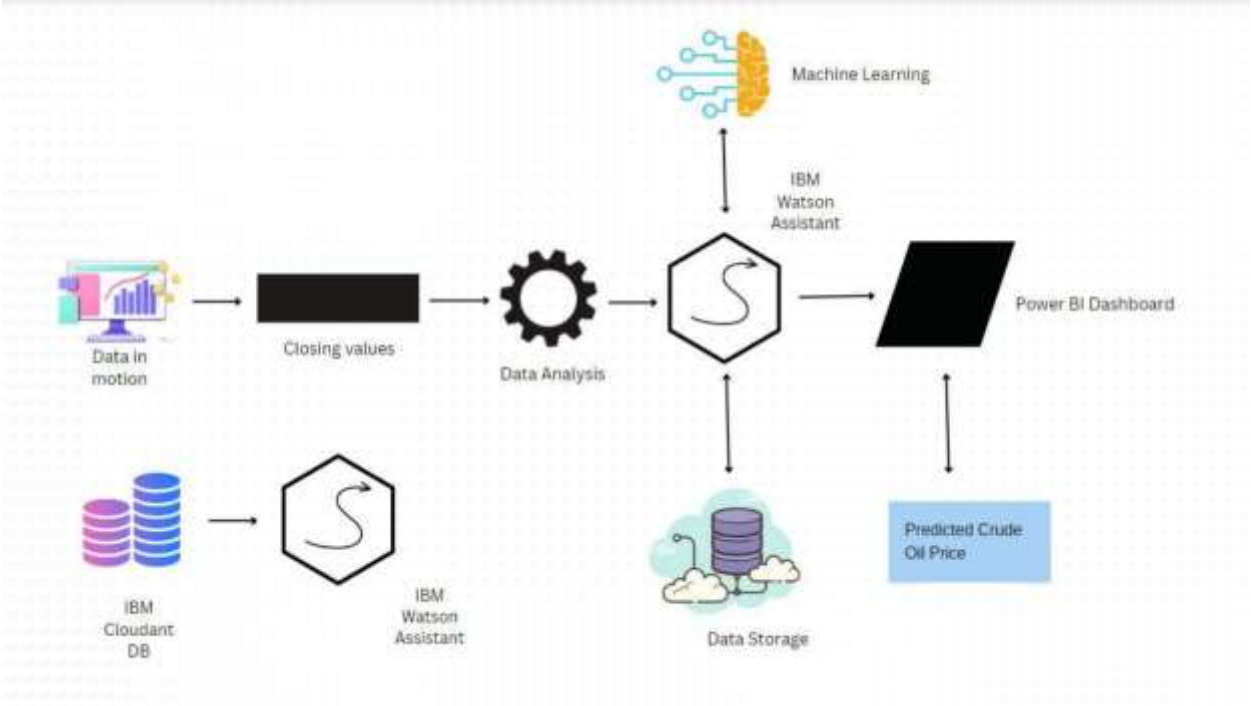
5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

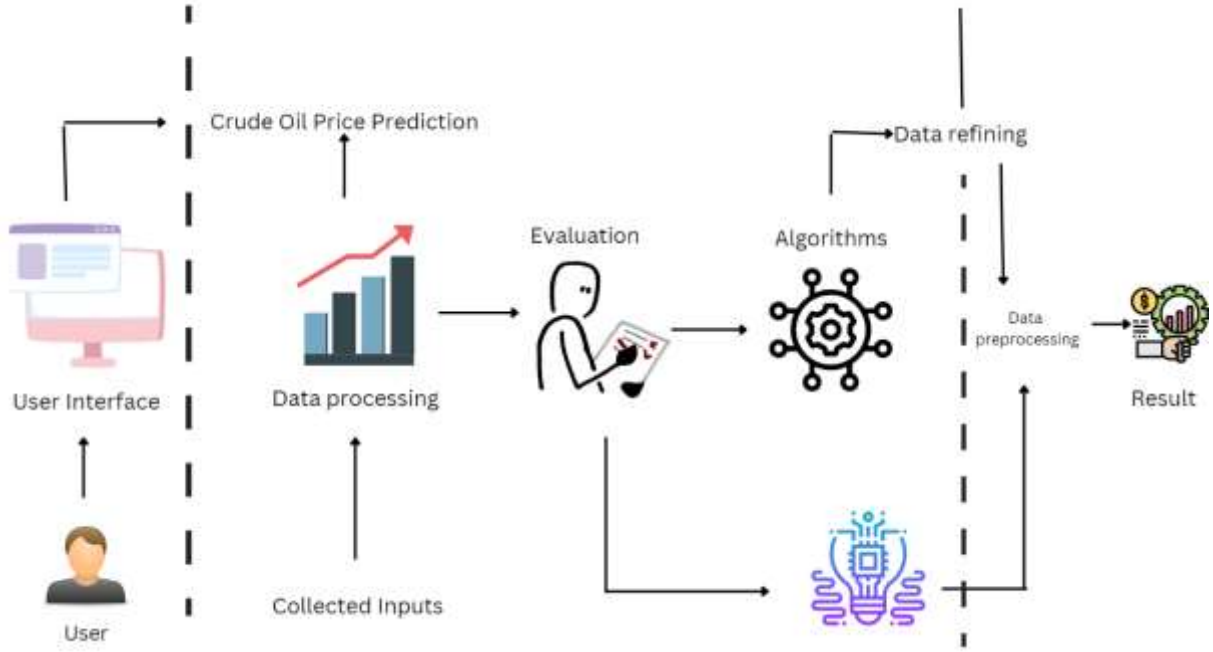


5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution Architecture :



Technical Architecture :



5.3 USER STORIES

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 / dashboard	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 & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register	Medium	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
				through existing gmail account.		
	Login	USN-5	As a user, I can log into the application by entering email & password	after registration,I can log in by only registered details	High	Sprint-1
	Dashboard (graphs)		After entering the inputs,the model will statistical results	Can be logged in through gmail or other social media account	High	Sprint-3
Customer (Web user)	Login	USN-1	As the web user,I can login simply by using Social Media Account.(gmail,facebook,linkedin)	An existing account can be used for login	Medium	Sprint-2
Customer Care Executive	Support		The help and support section will work towards the FAQ and the chatbot will reply to basic questions	I can solve the problems arised by Customers	Low	Sprint-3
Administrator	News		Admin will give the recent updates of Oil Prices.	Provide the recent oil prices	High	Sprint-4
	Notification		Admin will notify the customers with the oil prices changes and other updates.	Notification by Gmail and SMS	High	Sprint-4
	Accessibility		Admin can control the access of users	Access permission for Users	High	Sprint-4
	Database		Admin can store the details of users.	Admin can store the details of users.	High	Sprint-4

CHAPTER 6 – PROJECT PLANNING AND SCHEDULING

6.PROJECT PLANNING AND SCHEDULING

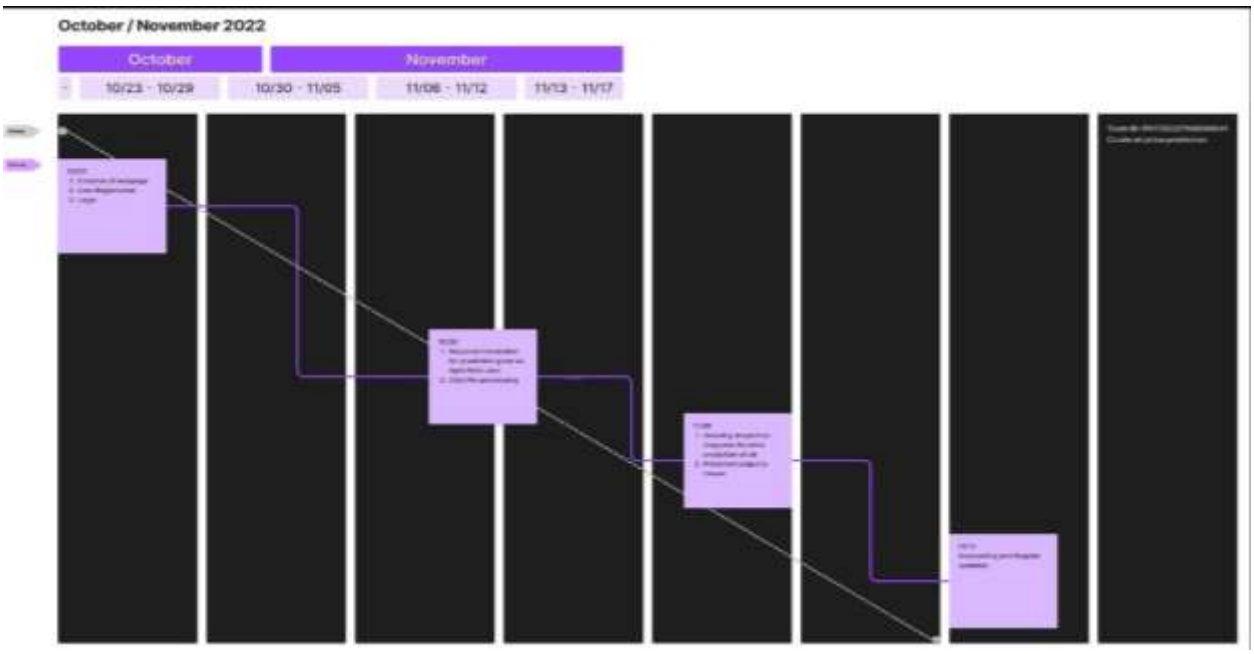
6.1 SPRINT PLANNING AND ESTIMATION

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	Kausik Subramaniyam.G ,B.Darshini,Deveshwar.R, Shunmuga Sundaram.K
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	10	High	Kausik Subramaniyam.G ,B.Darshini,Deveshwar.R, Shunmuga Sundaram.K
Sprint-1	Login	USN-3	As a user, I can register for the application through Facebook	15	High	Kausik Subramaniyam.G ,B.Darshini,Deveshwar.R, Shunmuga Sundaram.K
Sprint-2	Input	USN-4	As a user, I can give Input Details to Predict Likeliness of crude oil	15	High	Kausik Subramaniyam.G ,B.Darshini,Deveshwar.R, Shunmuga Sundaram.K
Sprint-2	Data pre-processing	USN-5	Transform raw data into suitable format for prediction.	15	High	Kausik Subramaniyam.G ,B.Darshini,Deveshwar.R, Shunmuga Sundaram.K
Sprint-3	Prediction of crude oil price	USN-6	As a user, I can predict Crude oil using machine learning model.	20	High	Kausik Subramaniyam.G ,B.Darshini,Deveshwar.R, Shunmuga Sundaram.K
Sprint-3		USN-7	As a user, I can get accurate prediction of crude oil	5	Medium	Kausik Subramaniyam.G ,B.Darshini,Deveshwar.R, Shunmuga Sundaram.K
Sprint-4	Review, Deployement	USN-8	As a user, I can give feedback of the application	20	High	Kausik Subramaniyam.G ,B.Darshini,Deveshwar.R, Shunmuga Sundaram.K

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	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 Nov 2022

6.3 REPORTS FROM JIRA



CHAPTER 7 – CODING AND SOLUTIONING

7.CODING AND SOLUTIONING

7.1 STATISCAL VISUALIZATION

The statistics of the crude oil price trends are displayed on the website which will enable the users to visualize the trends of the crude oil price. Various trends can be seen rather than having only a single graph. Multiple trends based on the time is recorded and displayed respectively.

CODE :

```
<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("{{url_for('static', filename='zp.png')}}");
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 black;
        padding:15px 3px;
        width:400px;
        outline:none;
        color:black;
        border-radius:0px;
        transition:0.25s;
```

```
        font-size:20;


}
.bor{
border:0;
        background:none;
        display:block;
        margin:20px auto;
        text-align:center;
        border:2px solid black;
        padding:10px 3px;
        width:500px;
        outline:none;
        color:black;
        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;
        color: black;
        border-color:black;
}
.logbtn{
        display:block;
        width:35%;
        height:50px;
        border:none;
```

```
        border-radius:24px;
        background:linear-gradient(120deg,#3498db,#8e44ad,#3498db,#8e44ad);
        background-size:200%;
        color:black;
        outline:none;
        cursor:pointer;
        transition:.5s;
        font-size:25;
}
.logbtn:hover{
        background-center;
}

input::placeholder{
        color:purple;
        font-family: verdana;
        font-size: 15;
}
.bottom-text{
        margin-top:60px;
        text-align:center;
        font-size:13px;

}

</style>
```

```
<body>

<div class="navbar">

<a href="/contact" style="font-size: 20px;color: blue;font-family: verdana;" >Contact | </a>

<a href="/" style="font-size: 20px;color: blue;font-family: verdana;" >Home</a>

<br>

<center>

<h1 style="color: white;font-family: verdana;font-size:
50px;">STATISTICS</h1></center><br><br><br><br>

<center><img src="{{url_for('static', filename='trends.png')}}" width="600px"><br></center><br><br>

<center><img src="{{url_for('static', filename='overyears.png')}}"
width="600px"><br></center><br><br>

<center><img src="{{url_for('static', filename='rangedaste.png')}}" width="600px"></center>




</body>


</html>
```

7.2 LSTM WITH DROP OUT

We have built another layer with drop out to compare the accuracy of different models.

CODE :

```
import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import datetime

import warnings

import itertools
```

```
import statsmodels.api as sm
import seaborn as sns
data=pd.read_excel("Crude Oil Prices Daily.xlsx")
data.isnull().any()
data.dropna(axis=0,inplace=True)
data.isnull().sum()
data.head()
oil_price=data.reset_index()['Closing Value']
oil_price
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
oil_price=scaler.fit_transform(np.array(oil_price).reshape(-1,1))
lst_output=[]
n_steps=10
i=0
while(i<10):
    if(len(temp_input)>10):
        #print(temp_input)
        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))
        #print(x_input)
        yhat = Model.predict(x_input, verbose=0)
        print("{} day output {}".format(i,yhat))
        temp_input.extend(yhat[0].tolist())
```

```
        temp_input=temp_input[1:]
        #print(temp_input)
        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
df3=oil_price.tolist()
df3.extend(lst_output)
plt.plot(df3[8100:])
```

CHAPTER 8 – TESTING

8.TESTING

8.1 TEST CASES

Sprint 1 :

Test Scenario	**Pre-Requisite**	**Steps To Execute**
Collect all the necessary datasets for the project	Dataset	1. Collect Dataset reqiured for the project.
Check if the login details is appropriate and login in to the portal	Registered credentials	1. Enter the appropriate details
Accept only if the details are not existing	User data	Enter a new detail which is not registered with the portal

Sprint 2 :

Test Scenario	**Pre-Requisite**	**Steps To Execute**
Check if the data is able to be normalized and fitted	Data Set	1. Data preprocessing
Verify whether the model gives accurate statiscal results	Data Set	1. Image Processing 2. Compiling the Deep Learning Model
Verify whether the data is splitted properly into train and test data	Visualization	Compile the model, Run the code Check for errors

Sprint 3 :

Test Scenario	**Pre-Requisite**	**Steps To Execute**
To Test the Model with the Test Data	Model Building	1. Fit the test data to the model 2.Calculate the accuracy
To Test the Model with Different algorithms	Model Building	1. Fit the data to the model

8.2 USER ACCEPTANCE TESTING

DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	3	3	4	19
Duplicate	2	1	2	0	5
External	1	5	1	0	7
Fixed	8	4	5	17	34
Not Reproduced	0	0	1	0	1
Skipped	0	1	0	0	1
Won't Fix	0	4	3	2	9
Totals	20	18	15	23	76

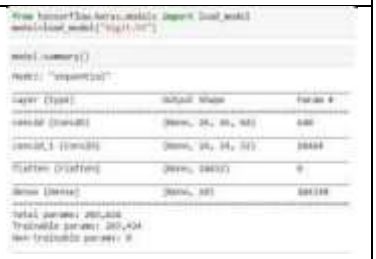
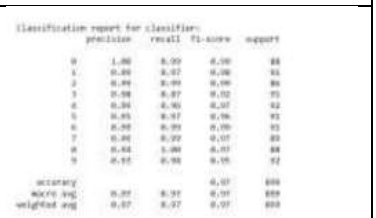
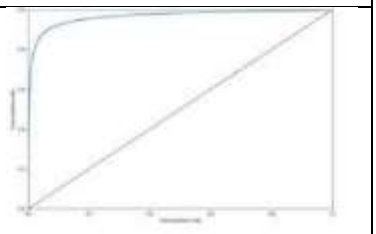
TEST CASE ANALYSIS

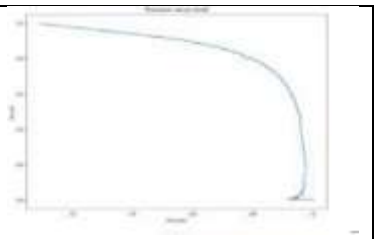
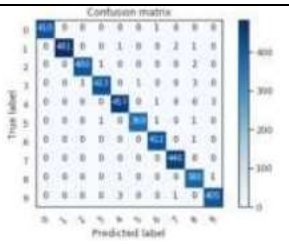
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	2	0	0	2
Client Application	1	0	0	1
Security	2	0	0	2
Outsource Shipping	0	0	0	0
Exception Reporting	1	0	0	1
Final Report Output	3	0	0	3
Version Control	0	0	0	0

CHAPTER 9 – RESULTS

9. RESULTS

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Model: "sequential" Layer (type) Output Shape Param # conv2d (Conv2D) (None, 26, 26, 64) 640 conv2d_1 (Conv2D) (None, 24, 24, 32) 18464 flatten (Flatten) (None, 18432) 0 dense (Dense) (None, 10) 184330 Total params: 203,434 Trainable params: 203,434 Non-trainable params: 0	
2.	Accuracy	Training Accuracy - 0.9979166388511658 Validation Accuracy -0.98089998960495	
3.	Metrics	Classification Model: precision,recall,f1-score,support	
4.	Metrics	ROC (Receiver Operating Characteristics) curve	

5.	Metrics	Precision-Recall or PR curve	
6.	Metrics	Confusion Matrix	

CHAPTER 10 – ADVANTAGES AND DISADVANTAGES

10. ADVANTAGES

According to economic theory, the price of crude oil should be easily predictable from the equilibrium between demand and supply, wherein demand forecasts are usually made from GDP, exchange rates and domestic prices, and supply is predicted from past production data and reserve data.

Prediction of future crude oil price with the history

The statistical records are displayed which will improve the users usability

DISADVANTAGES

Only if the data for the past 10 days is available the future can be prediction can be made. So the previous data is crucial.

CHAPTER 11 – CONCLUSION

11.CONCLUSION

For predicting the crude oil price, we have chosen the machine learning approach to study the crude oil price dataset. We have applied various deep learning algorithms to decide which one will be the best for applying on the dataset to get the result with the highest accuracy.

CHAPTER 12 – FUTURE SCOPE

12.FUTURE SCOPE :

In the future modules will be developed and the project will be made use as a tool to make the users more efficient and help them take wiser decisions. The model will be developed with various features which includes the admin module to manage the usage and the data more wisely. It will also be made into a live portal which shows the statistical data of the crude oil price as well the updates will be sent to the users registered dashboard as well as the email and contact credentials. By building such a model some of the people in the society might be use it for a good cause.

CHAPTER 13 – APPENDIX

13.APPENDIX

SOURCE CODE :

```
import numpy as np  # used for numerical analysis

from flask import Flask, render_template, request, url_for, redirect  # Flask is a application used to run/serve our application

# request is used to access the file which is uploaded by the user in our application

# render_template is used for rendering the html pages

from tensorflow.keras.models import load_model  # we are loading our model from keras

app = Flask(__name__)  # our flask app

model = load_model('crude_oil_price_prediction.h5')  # loading the model in the flask app

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

def home():

    error = None

    if request.method == 'POST':

        if request.form['username'] != "admin" or request.form['password'] != "admin":

            error = 'Invalid Credentials. Please try again.'

        else:

            return redirect(url_for('mains'))

    return render_template('login.html', error=error)
```

```
@app.route('/mains', methods=['GET', 'POST'])
def mains():
    return render_template('index.html')
@app.route('/stats', methods=['GET', 'POST'])
def stats():
    return render_template('stats.html')
@app.route('/about')
def home1():
    return render_template("index.html")  # rendering html template
@app.route('/predict')
def home2():
    return render_template("web.html")  # rendering html template


@app.route('/contact')
def contact():
    return render_template("contact.html")
@app.route('/login', methods=['POST'])  # route for our prediction
def login():
    a = request.form['year1']
    b = request.form['year2']
```

```
    c = request.form['year3']
    d = request.form['year4']
    e = request.form['year5']
    f = request.form['year6']
    g = request.form['year7']
    h = request.form['year8']
    i = request.form['year9']
    j = request.form['year10']  # requesting the file
    x_input = [[float(a), float(b), float(c), float(d), float(e), float(f), float(g), float(h), float(i), float(j)]]
    print(x_input)
    lst_output = model.predict(x_input)
    lst_output = np.round(lst_output[0][0], 2)
    return render_template("web.html", showcase='The Predicted crude oil price is : Rs. '+str(lst_output))
if __name__ == '__main__':
    app.run(debug=False)
```

MODEL BUILDING

```
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
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,y_test),epochs=30,batch_size=64,verbose=1)

train_predict=scaler.inverse_transform(train_data)

test_predict=scaler.inverse_transform(test_data)

import math

from sklearn.metrics import mean_squared_error

math.sqrt(mean_squared_error(train_data,train_predict))

from tensorflow.keras.models import load_model

model.save("crude_oil_price_prediction.h5")

look_back=10

trainpredictPlot = np.empty_like(oil_price)

trainpredictPlot[:, :]= np.nan

trainpredictPlot[look_back:len(train_predict)+look_back, :] = train_predict


testPredictplot = np.empty_like(oil_price)

testPredictplot[:,: ] = np.nan
```

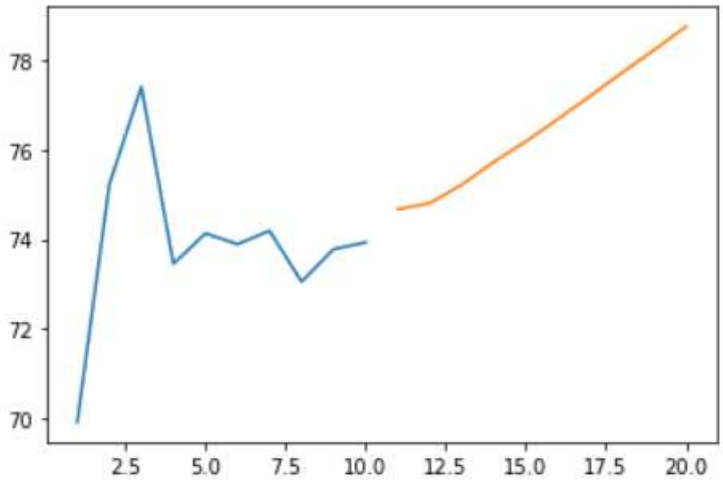
```
testPredictplot[look_back:len(test_predict)+look_back, :] = test_predict


plt.plot(scaler.inverse_transform(oil_price),color="red")
plt.show()
len(test_data)
x_input=test_data[2866:2876].reshape(1,-1)
x_input.shape
temp_input=list(x_input)
temp_input=temp_input[0].tolist()
lst_output=[]
n_steps=10
i=0
while(i<10):
    if(len(temp_input)>10):
        x_input=np.array(temp_input[1:11])
        print("{} day input {}".format(i,x_input))
        x_input=x_input.reshape(1,-1)
        x_input = x_input.reshape((1, n_steps, 1))
        #print(x_input)
        yhat = model.predict(x_input, verbose=0)
        print("{} day output {}".format(i,yhat))
```

```
        temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:]
        #print(temp_input)
        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(oil_price)
plt.plot(day_new, scaler.inverse_transform(oil_price[8206:]))
plt.plot(day_pred, scaler.inverse_transform(lst_output))
df3=oil_price.tolist()
df3.extend(lst_output)
plt.plot(df3[8100:])
```

df3=scaler.inverse_transform(df3).tolist()

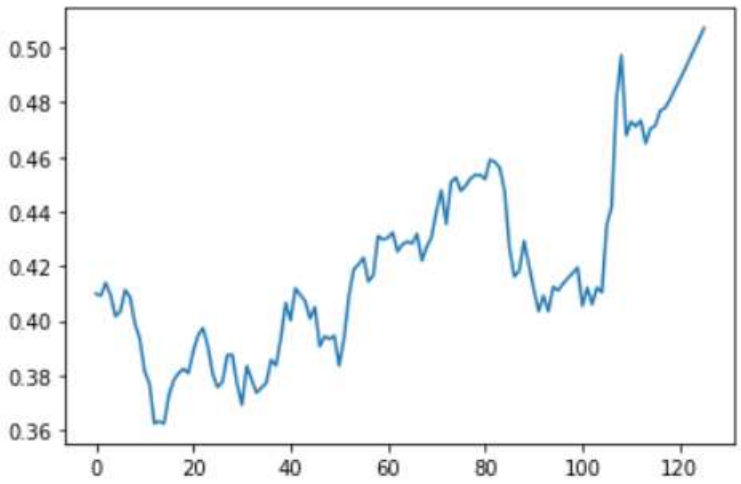
plt.plot(scaler.inverse_transform(oil_price))

Out[46]: []



In [47]:
```
df3=oil_price.tolist()
df3.extend(lst_output)
plt.plot(df3[8100:])
```

Out[47]: []



In [48]:
```
df3=scaler.inverse_transform(df3).tolist()
```

GITHUB AND PROJECT DEMO LINK:

Github :

https://github.com/IBM-EPBL/IBM-Project-13344-1659516895

Project Demo Link:

https://drive.google.com/file/d/1aGhMBd6bYX4x548pyD8KPqASCaRv_8Ou/view?usp=share_link