PROJECT REPORT

Crude Oil Price Prediction

DOMAIN	ARTIFICIAL INTELLIGENCE
TOPIC	Crude Oil Price Prediction
TEAM ID	PNT2022TMID15017
TEAM MEMBERS	RETHIKA R A, SURIYA PRAKASH S K, VAWNIKA M S, NARAIN SRINIVAS T

CRUDE OIL PRICE PREDICTION

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROJECT OVERVIEW:

Oil the world economy's most important source of energy and it is therefore critical to economic growth. Its value is driven by demand for refined petroleum products particularly in the transportation sector.

1.2 PURPOSE:

- 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.
- RNN (Recurrent Neural Network) with LSTM (Long Short Term Memory) to achieve the task.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

S.NO	NAME OF PAPER	AUTHOR	YEAR OF PUBLICATION	TECHNOLOGY USED	DRAWBACKS
1	Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU.	Ganiyu Adewale Busari	2021	AdaBoost algorithm	The proposed method, AdaBoost-GRU outperforms the single methods and AdaBoost-LSTM ensemble model in this study.
2	A novel crude oil price trend prediction method.	HuiziHe,MeiSun,XiumingLi,Isaac AdjeiMensah	2022	Machine learning classification algorithm	Regression in forecasting price trend.
3	Prediction of crude oil prices in COVID-19 outbreak using real data	ÖznurÖztunç Kaymak	2020	Artificial neural networks (ANNs) and support vector machine (SVM) methods.	Hourly dataset is used.
4	Forecasting crude oil price with multilingual search engine data	Taiyong Li,Zijie Qian,Shuheng Wang	2020	Search engine data (SED)	No known competing financial interests.

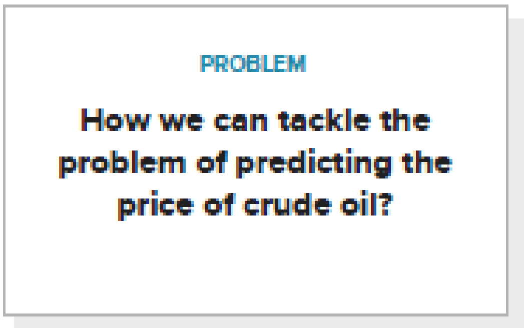
5	Effective crude oil price forecasting.	Binrong Wu,Lin Wang	2021	Convolutional neural network (CNN)	Absolute percentage error.

2.2 REFERENCES

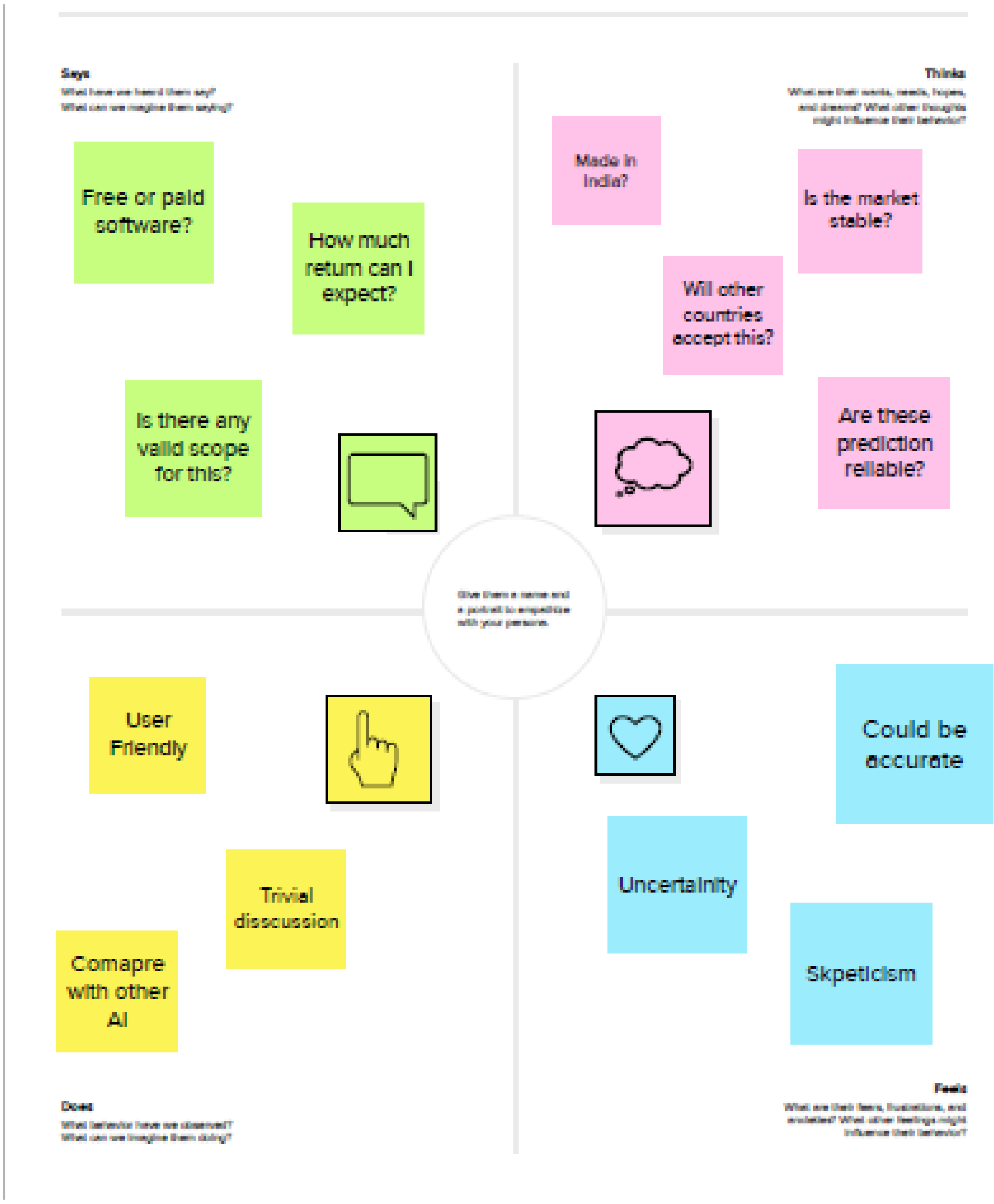
1. Ganiyu Adewale Busari,"Crude oil price prediction: A comparison between AdaBoost- LSTM and AdaBoost-GRU for improving forecasting performance".
2. HuiziHe,MeiSun,XiumingLi,Isaac AdjeiMensah, "A novel crude oil price trend prediction method: Machine learning classification algorithm based on multi-modal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

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.

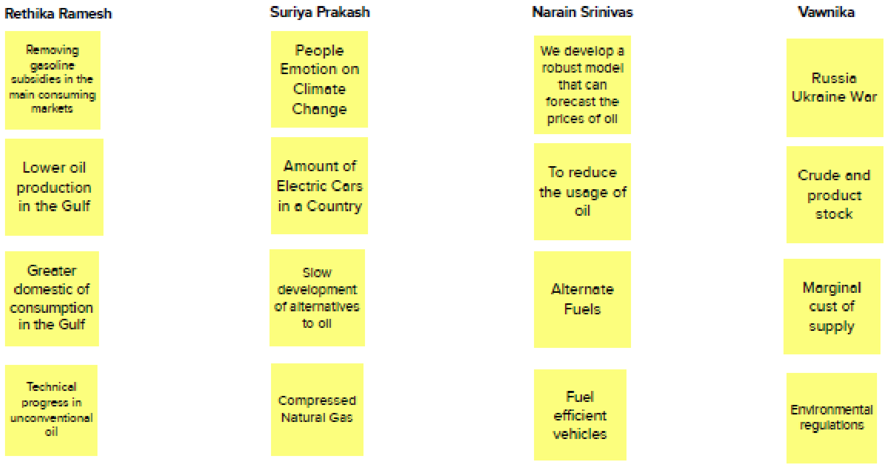


3.1 EMPATHY MAP CANVAS

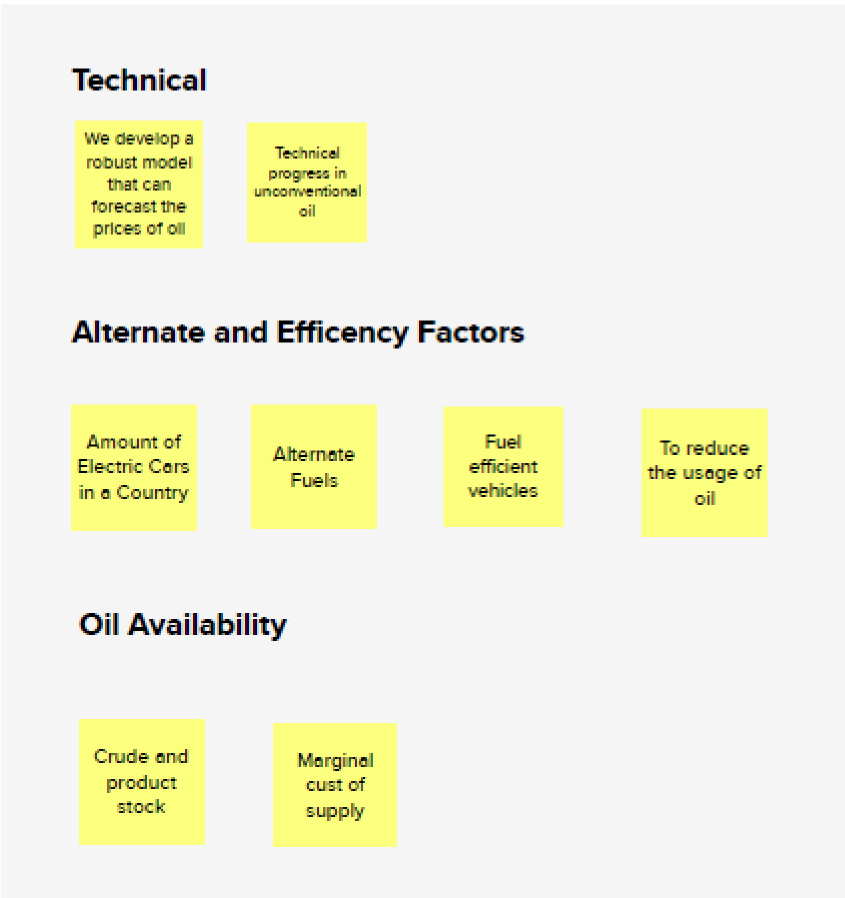


3.2 IDEATION AND BRAINSTORMING

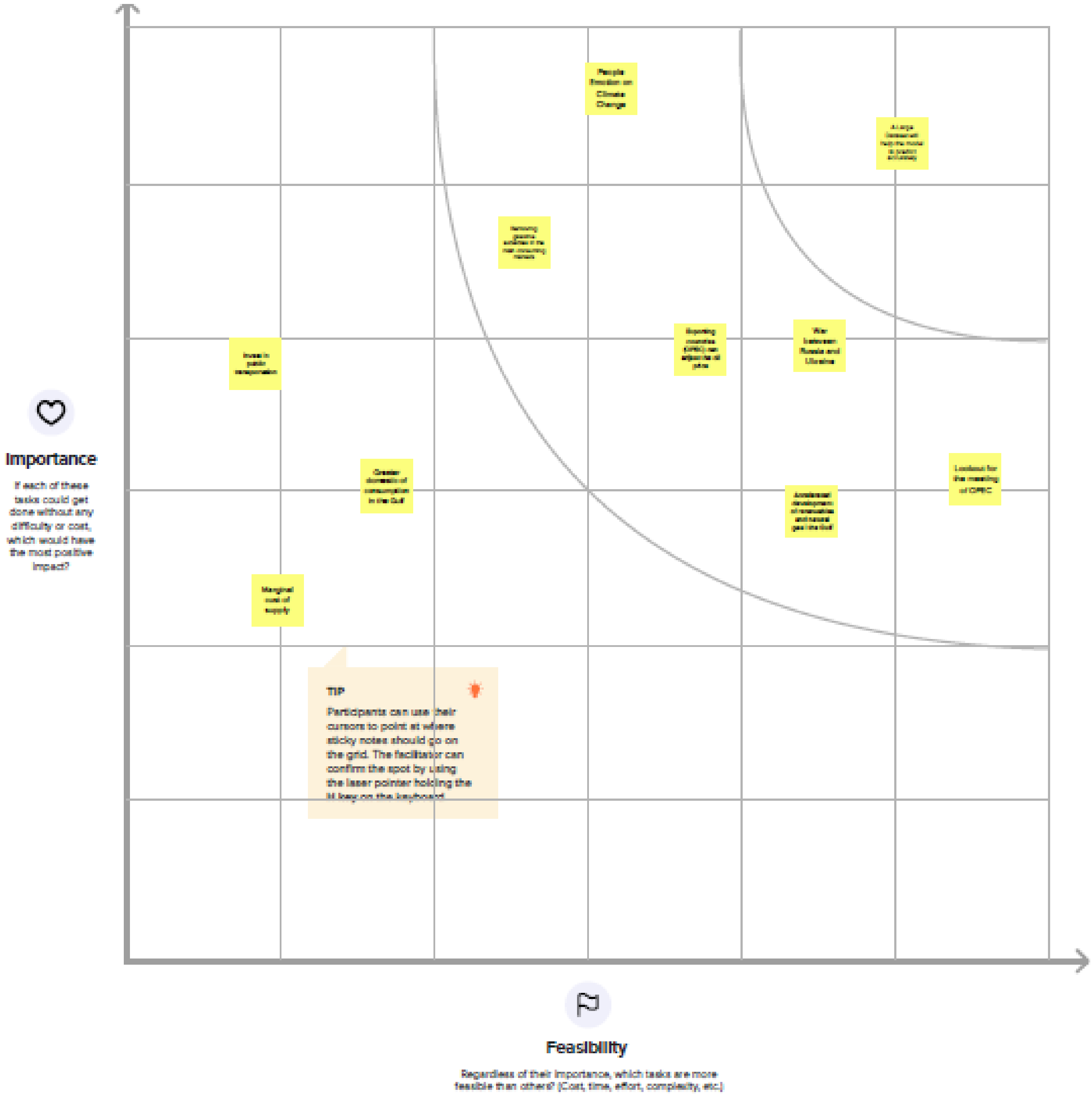
BRAINSTORM



GROUP IDEAS



PRIORITIZE



People Emotion on Climate Change
War between Russia and Ukraine
Lookout for the meeting of OPEC
Greater domestic of consumption in the Gulf
Marginal cost of supply
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 PROPOSED SOLUTION

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Crude oil is the world’s leading fuel, and its prices have a big impact on the global environment its forecasts are very useful to governments, the industry is individuals. • The continuous usage of statistical and econometric techniques including AI for crude oil price prediction might demonstrate demotions to the prediction performance.
2.	Idea / Solution description	• In order to predict future crude oil using historical data on crude oil, RNN is utilised with long short-term memory. • The effectiveness of the cost is calculated using the mean squared error. Using the pricing information in the WTO crude oil materials, the proposed model's performance is assessed.
3.	Novelty / Uniqueness	• Crude oil price variations have a significant impact on the world's economies; thus price forecasting can help reduce the risks brought on by this volatility. • For a variety of stakeholders, including governments, public and private businesses, legislators, and investors, price projections are crucial.
4.	Social Impact / Customer Satisfaction	• It is used to predict the future price and use the oil according to the prices. • This price directly influences a variety of items, and its variations have an impact on the capital markets.
5.	Business Model (Revenue Model)	• It can help decision makers – either firms, private investors, or individuals – when choosing to buy or sell the crude oil. • RNN and LSTM models are used as the benchmark model to predict crude oil prices.
6.	Scalability of the Solution	• PCA, MDS, and LLE methods are used to reduce the dimensions of the data. • Improve the accuracy of the RNN and LSTM models.

3.4 PROBLEM SOLUTION FIT

Problem-Solution fit canvas 2.0

Purpose / Vision Rethika Suriya Narain Vawnika

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

1.Our project mainly focuses on the continuous usage of statistical and econometric techniques including AI for crude oil price prediction might demonstrate demotions to the prediction performance.

2. Our project is used to predict the future price and use the oil according to the prices.People from any age group can use this application.

6. CUSTOMER CONSTRAINTS — CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

1.Proper internet connectivity is required.

2.User must enter appropriate details for accurate results.

3.Must read the guidelines for better usage.

5. AVAILABLE SOLUTIONS — AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

1. if crude oil price goes low ,the easiest way to take advantage of the low prices is to fleece the bears.
2. Simply buying oversold oil or gas stocks can be a great way to take advantage now and reap the benefits when the bears realize their mistake and oil prices rebound.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

1.Websites crashes should be avoided.

2,Application interface should be user-friendly.

3.Precision of results delivered.

9. PROBLEM ROOT CAUSE — RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

1. Changing pattern of oil prices.

2. Inexperienced professionals.

7. BEHAVIOUR — BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

1.Closing price is the last price at which a stock trades during a regular trading session.

2.The Closing Price helps the investor understand the market sentiment of the stocks over time. It is the most accurate matrix to determine the valuation of stock until the market resumes trading the next day.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS — TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

1.Cost Effective.
2.Early prediction can avoid serious problems.

4. EMOTIONS: BEFORE / AFTER — EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

1.Trust, Profit gain or loss fear, insecurity.

10. YOUR SOLUTION — SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

1.This Guided Project mainly focus on applying Neural Networks to predict the crude oil price.
2.This decision helps us to buy crude oil at proper time.
3.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.
4.So we would be implementing RNN(Recurrent Neural Network) with LSTM(Long Short Term Memory) to achieve the task.

8. CHANNELS of BEHAVIOUR — CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

1.Searching online for current crude oil prices.

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

1.Performing fundamental analysis.
2.Technical analysis.
3.Risk Management

Extract online & offline CH of BE

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub- Task)
FR-1	User Registration	User Direct Open With Google Play Store App User Can Download The Crude Oil Price
FR-2	User Confirmation	User Using The Application There Are So Many Products In Crude Oil Price App. 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 . User View Major Energy Quotes. User Can Using A Multiple Color Themes.
FR-4	User Exceptions	User Can Exchange Rates And Currancy Converter.

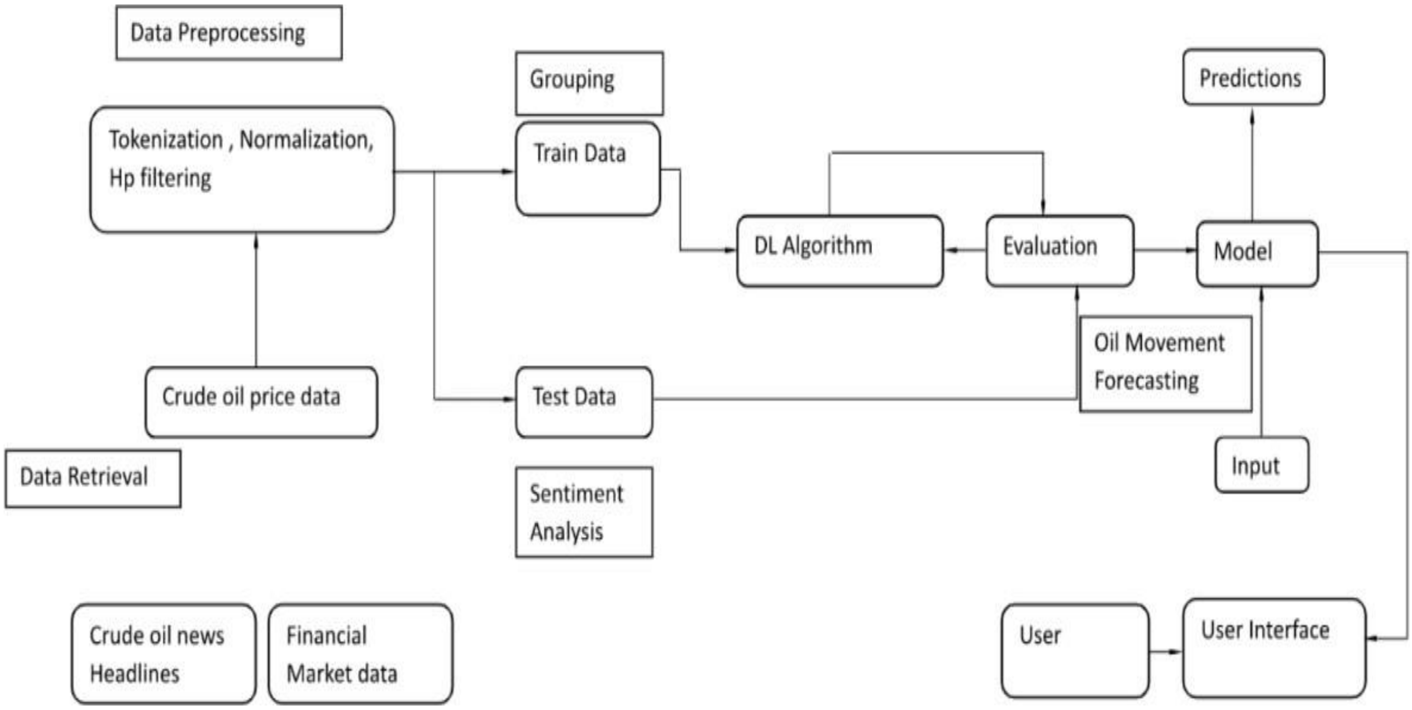
4.2 NON-FUNCTIONAL REQUIREMENTS

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
NFR- 3	Reliability	Reliability of the pointing towards high risk Components.
NFR- 4	Performance	Performance of 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 is 90%-95%.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

- 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.

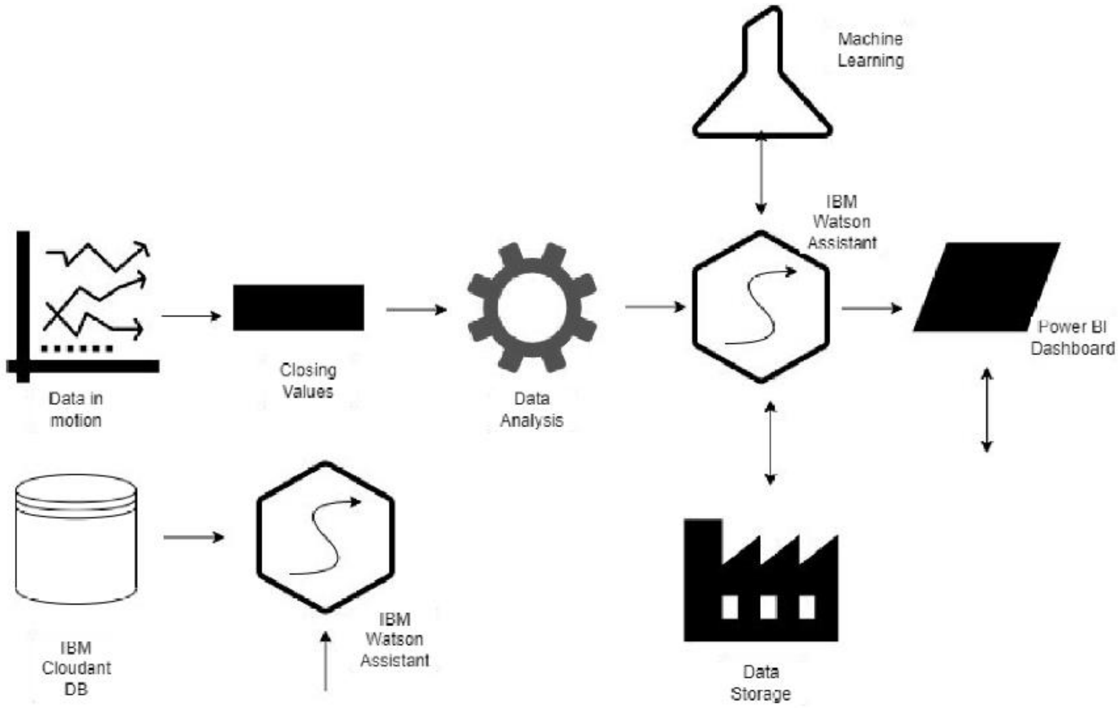


5.2 SOLUTION AND TECHNICAL ARCHITECHTURE

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



TECHNICAL ARCHITECHTURE:

PROJECT FLOW:

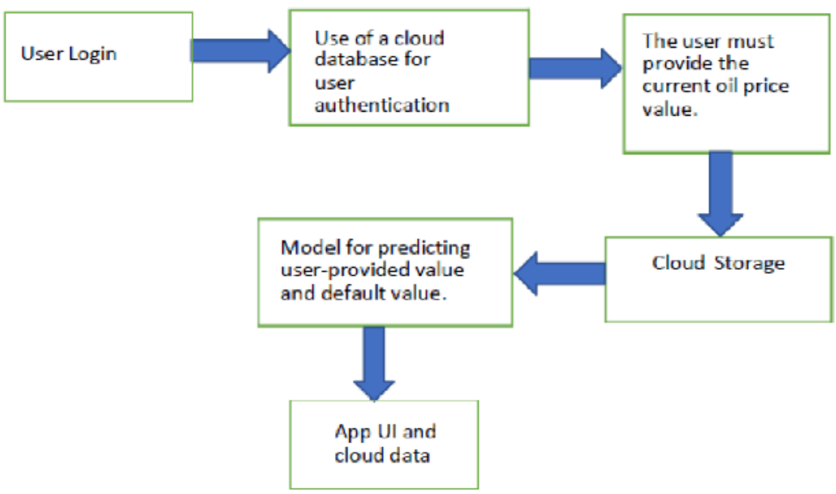


Table-1 : Components & Technologies:

S.no	Component	Description	Technology
1.	User Interface	Web application	HTML, CSS, JavaScript , Angular Js
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson Assistant
4.	Database	Data Type, Configurations	MySQL
5.	Cloud Database	Database Service on Cloud	IBM cloud
6.	File Storage	File storage requirements	IBM Block Storage, Local Filesystem
7.	External API-1	Purpose of External API used in the application	Firebase
8.	Machine Learning Model	Purpose of Machine Learning Model	Recurrent neural network & LSTM
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Firebase

Table-2: Application Characteristics:

S.no	Characteristics	Description	Technology
1.	Open-Source Frameworks-1	Python,	Pandas, flask, NumPy, TensorFlow
2.	Open-Source Frameworks-2	JavaScript, Angular Js.	App module, component module
3.	Security Implementations	User data will be stored according to CIA model.	End to end encryption (SHA-256)
4.	Scalable Architecture	IBM cloud and firebase both used for better performance in storage and authentication.	IBM Watson , Firebase, MySQL
5.	Availability	Handle huge requests, avoid DDOS and XSS attack.	Effective coding and restrictive user access based on need
6.	Performance	Handle more than 1000 users to use server at a time.	Flask

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	You can download the crude oil price by opening the Google Play Store app directly as a user.	I can access my account / dashboard	High	Sprint-1
	Additional Features	USN-2	Users can read the most recent news and see oil price charts. Major Energy Quotes User View The user may use many colour schemes.	I can view then read the price prediction.	High	Sprint-1
	Available Products	USN-3	Users of the application may instantly update the energy and oil prices while using it because there are so many different products in the crude oil price app.	I can receive the data once click then confirm	High	Sprint-2
	Expectations	USN-4	User Can Convert Currency And Exchange Rates	I can expect	Medium	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Login	USN-5	Log in as a user without using your email address, username, or password.		High	Sprint-1
	Dashboard					
Customer (Web user)			I can see the price of crude oil as a consumer.	I can view the price directly	High	Sprint - 1
Customer Care Executive			I am the user and I executive the pricing history.	can accept the terms	High	Sprint - 1
Administrator			As a manager, it anticipates the results.	Show the result	High	Sprint - 1

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	Data Collection	USN-1	Collecting the Dataset	10	High	Rethika Ramesh Suriya Prakash Narain Srinivas Vawnika
Sprint-1		USN-2	Data Pre-Processing	7	Medium	Rethika Ramesh Suriya Prakash Narain Srinivas Vawnika
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model.	10	High	Rethika Ramesh Suriya Prakash Narain Srinivas
Sprint-2		USN-4	Training the data classification model using RNN and others systems.	7	Medium	Rethika Ramesh Suriya Prakash Narain Srinivas Vawnika

Sprint-3		USN-5	Training the model and testing the model's performance.	10	High	Suriya Prakash Narain Srinivas Vawnika
Sprint-4	Training and Testing	USN-6	Build the system and deploy the model in IBM cloud	7	Medium	Rethika Ramesh Suriya Prakash Vawnika

Project Tracker, Velocity & Burndown Chart:

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	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	7	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity}$$

Average Velocity of Our Team= 6/10

= 0.6

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

6.2 MILESTONE AND ACTIVITY LIST

S.No	Milestone	Activities	Team Members
1.	Data Collection	Create Train and Test Folders	Rethika Ramesh Suriya Prakash Vawnika
2.	Data Preprocessing	Import Library and Configure	Rethika Ramesh Suriya Prakash Narain Srinivas
3.	Data Preprocessing	Analyze the data functionality to Train and Test Set	Suriya Prakash Narain Srinivas Vawnika
4.	Model Building	Import the required model building libraries	Rethika Ramesh Narain Srinivas Vawnika
5.	Model Building	Initialize the model	Rethika Ramesh Suriya Prakash Narain Srinivas Vawnika
6.	Model Building	Add LSTM Layers	Rethika Ramesh Suriya Prakash Narain Srinivas
7.	Model Building	Adding output layers	Rethika Ramesh Suriya Prakash Vawnika
8.	Model Building	Compile the model	Rethika Ramesh Suriya Prakash Narain Srinivas Vawnika
9	Model Building	Fit and save the model	Rethika Ramesh Narain Srinivas Vawnika
10.	Test the model	Import the packages and load the saved Model	Rethika Ramesh Narain Srinivas Vawnika
11.	Test the model	Load the test data, pre- process it and predict	Suriya Prakash Narain Srinivas Vawnika
12.	Application Building	Build a flask application	Suriya Prakash Narain Srinivas Vawnika
13.	Application Building	Build the HTML page	Rethika Ramesh Suriya Prakash Vawnika
14.	Application Building	Output	Rethika Ramesh Suriya Prakash Narain Srinivas
15.	Train RNN Model on IBM	Register for IBM Cloud	Rethika Ramesh Suriya Prakash Vawnika
16.	Train RNN Model on IBM	Train Data classification Model	Rethika Ramesh Suriya Prakash Narain Srinivas

7.CODING AND SOLUTIONING

7.1 DATA PREPROCESSING

Importing the libraries:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import tensorflow as tf

data=pd.read_excel(r"Crude Oil Prices Daily.xlsx")

data.head()

	Date	Closing Value
0	1986-01-02	25.56
1	1986-01-03	26.00
2	1986-01-06	26.53
3	1986-01-07	25.85
4	1986-01-08	25.87

Handling missing values

data.isnull().any()

Date False

Closing Value True

dtype: bool

data.isnull().sum()

Date 0

Closing Value 7

dtype: int64

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

data_oil=data.reset_index()['Closing Value']

data_oil

0 25.56

1 26.00

2 26.53

3 25.85

4 25.87

...

8211 73.89

8212 74.19

8213 73.05

8214 73.78

8215 73.93

Name: Closing Value, Length: 8216, dtype: float64

data.isnull().any()

Date False

Closing Value False

dtype: bool

Feature Scaling

from sklearn.preprocessing import MinMaxScaler

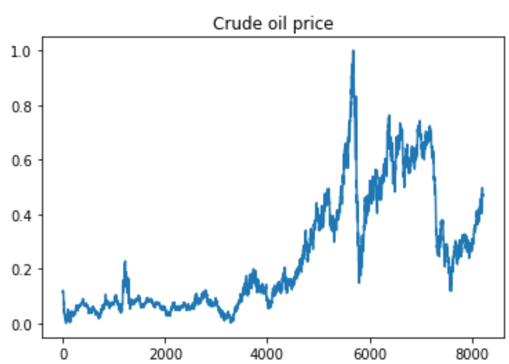
scalar=MinMaxScaler(feature_range=(0,1))

data_oil=scalar.fit_transform(np.array(data_oil).reshape(-1,1))

Data Visualization

plt.title('Crude oil price')

plt.plot(data_oil)



Splitting data into Train and Test Data

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

(5340, 2876)

train_data.shape

(5340, 1)

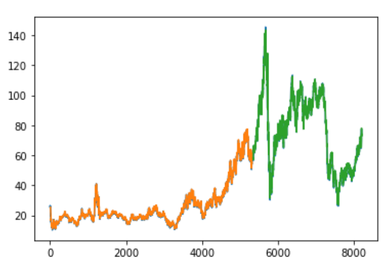
Creating a dataset with sliding windows

```
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, y_test = create_dataset(test_data,time_step)
print(X_train.shape),print(y_train.shape)
(5329, 10)
(5329,)
(None, None)
print(X_test.shape),print(y_test.shape)
(2865, 10)
(2865,)
(None, None)
X_train
array([[0.11335703, 0.11661484, 0.12053902, ..., 0.10980305, 0.1089886 ,
        0.11054346],
       [0.11661484, 0.12053902, 0.11550422, ..., 0.1089886 , 0.11054346,
        0.10165852],
       [0.12053902, 0.11550422, 0.1156523 , ..., 0.11054346, 0.10165852,
        0.09906708],
       ...,
       [0.36731823, 0.35176958, 0.36080261, ..., 0.36391234, 0.37042796,
        0.37042796],
       [0.35176958, 0.36080261, 0.35354657, ..., 0.37042796, 0.37042796,
        0.37879461],
       [0.36080261, 0.35354657, 0.35295424, ..., 0.37042796, 0.37879461,
        0.37916482]])
X_train.shape
(5329, 10)
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)
```

7.2 MODEL BUILDING:

```
import tensorflow

import keras

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

model=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: "sequential_1"

_________________________________________________________________

 Layer (type)              Output Shape          Param #

=================================================================

 lstm (LSTM)               (None, 10, 50)        10400


 lstm_1 (LSTM)             (None, 10, 50)        20200


 lstm_2 (LSTM)             (None, 50)            20200


 dense (Dense)             (None, 1)             51


=================================================================

Total params: 50,851

Trainable params: 50,851

Non-trainable params: 0

_________________________________________________________________

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)

from sklearn.model_selection import train_test_split

import tensorflow as tf

train_predict = model.predict(X_train)

test_predict = model.predict(X_test)
```

```
167/167 [==============================] - 3s 7ms/step
90/90 [==============================] - 1s 6ms/step
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))
29.607859180352207
math.sqrt(mean_squared_error(ytest,test_predict))
78.82827278932622
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")
```




```
Green indicates predicated data
Blue indicates complete data
Orange indicates train data
len(test_Data)
2876
```

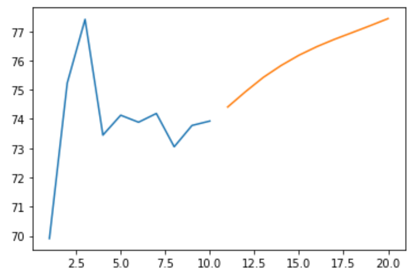
```
X_input=test_Data[2866:].reshape(1,-1)

X_input.shape

(1, 10)

temp_input=list(X_input)

temp_input=temp_input[0].tolist()

temp_input

[0.44172960165852215,
 0.48111950244335855,
 0.49726047682511476,
 0.4679401747371539,
 0.4729749740855915,
 0.47119798608026064,
 0.47341922108692425,
 0.4649785280616022,
 0.4703835332444839,
 0.47149415074781587]

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
[0.47505158]
11
1 Day input [0.4811195  0.49726048 0.46794017 0.47297497 0.47119799 0.47341922
 0.46497853 0.47038353 0.47149415 0.47505158]
1 Day output [[0.47893462]]
2 Day input [0.49726048 0.46794017 0.47297497 0.47119799 0.47341922 0.46497853
 0.47038353 0.47149415 0.47505158 0.47893462]
2 Day output [[0.482561]]
3 Day input [0.46794017 0.47297497 0.47119799 0.47341922 0.46497853 0.47038353
 0.47149415 0.47505158 0.47893462 0.48256099]
3 Day output [[0.48557332]]
4 Day input [0.47297497 0.47119799 0.47341922 0.46497853 0.47038353 0.47149415
 0.47505158 0.47893462 0.48256099 0.48557332]
4 Day output [[0.48816994]]
5 Day input [0.47119799 0.47341922 0.46497853 0.47038353 0.47149415 0.47505158
 0.47893462 0.48256099 0.48557332 0.48816994]
5 Day output [[0.4903399]]
6 Day input [0.47341922 0.46497853 0.47038353 0.47149415 0.47505158 0.47893462
 0.48256099 0.48557332 0.48816994 0.49033991]
6 Day output [[0.49222207]]
7 Day input [0.46497853 0.47038353 0.47149415 0.47505158 0.47893462 0.48256099
 0.48557332 0.48816994 0.49033991 0.49222207]
7 Day output [[0.49392977]]
8 Day input [0.47038353 0.47149415 0.47505158 0.47893462 0.48256099 0.48557332
 0.48816994 0.49033991 0.49222207 0.49392977]
8 Day output [[0.49566177]]
9 Day input [0.47149415 0.47505158 0.47893462 0.48256099 0.48557332 0.48816994
 0.49033991 0.49222207 0.49392977 0.49566177]
9 Day output [[0.4974547]]
day_new=np.arange(1,11)
day_pred=np.arange(11,21)
len(Data_oil)
8216
```

plt.plot(day_new,scaler.inverse_transform(Data_oil[8206:]))

plt.plot(day_pred,scaler.inverse_transform(lst_output))

[]

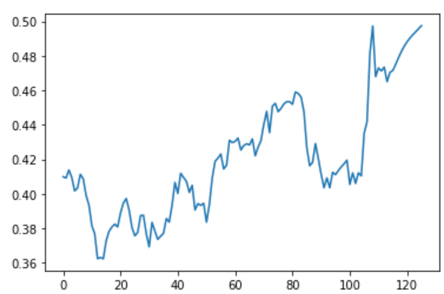


df3=Data_oil.tolist()

df3.extend(lst_output)

plt.plot(df3[8100:])

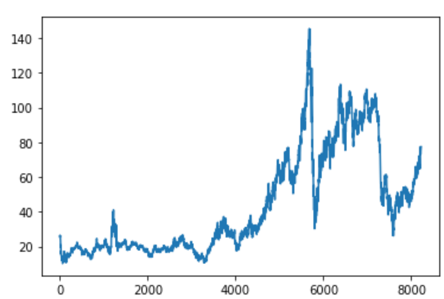
[]



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

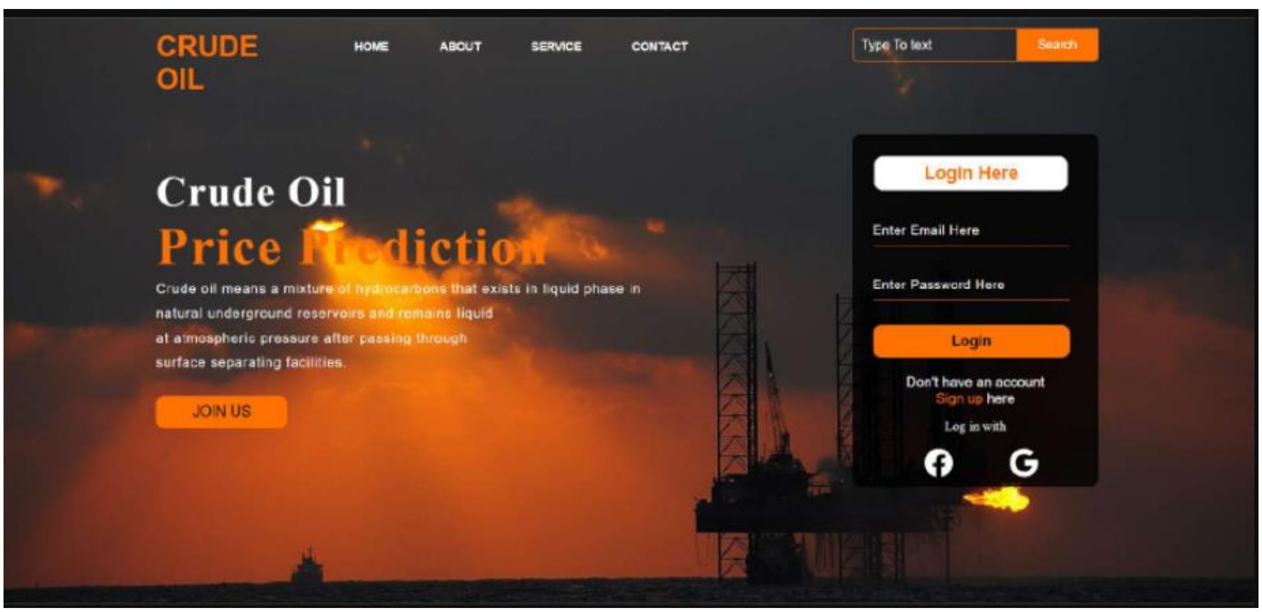
plt.plot(df3)

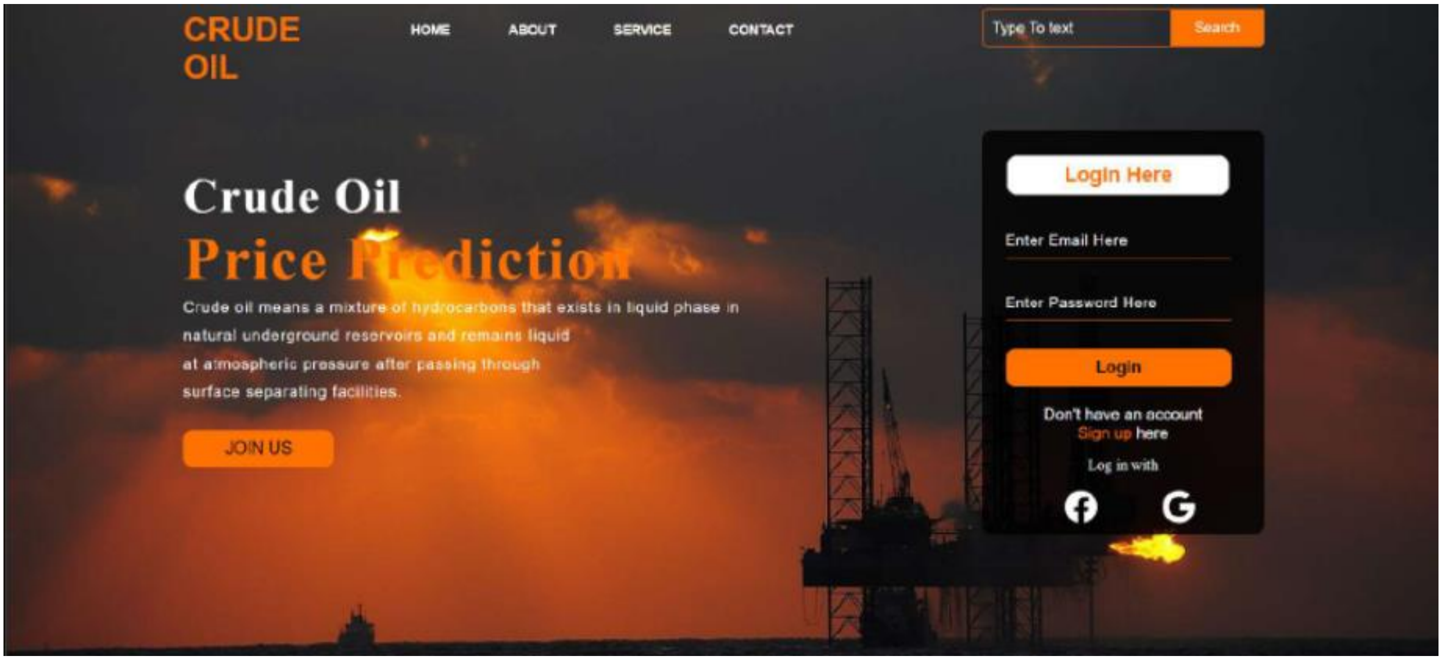
[]



8. RESULTS


CRUDE OIL PRICE PREDICTION
ENTER PRICE: dd-mm-yyyy Submit
PREDICTED PRICE:


CRUDE OIL
HOME
ABOUT
SERVICE
CONTACT
Type To text
Search
Crude Oil
Price Prediction
Crude oil means a mixture of hydrocarbons that exists in liquid phase in natural underground reservoirs and remains liquid at atmospheric pressure after passing through surface separating facilities.
JOIN US
Login Here
Enter Email Here
Enter Password Here
Login
Don't have an account
Sign up here
Log in with



CRUDE OIL
HOME
ABOUT
SERVICE
CONTACT
Type To text
Search
Crude Oil
Price Prediction
Crude oil means a mixture of hydrocarbons that exists in liquid phase in natural underground reservoirs and remains liquid at atmospheric pressure after passing through surface separating facilities.
JOIN US
Login Here
Enter Email Here
Enter Password Here
Login
Don't have an account
Sign up here
Log in with

9.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- The proposed technique can be extended by considering other factors that affect crude oil price volatilities such as, financial market, economic growth, exchange rate, demand and supply and the weather.
- And the horizon of the prediction can be widened by considering daily data.
- The proposed technique can be implemented with different dataset such as the stock market data in the future to further check the validity of the proposed technique.

DISADVANTAGES:

- The oil price hike is due to the cost of shipping goods of all types rises, since oil is used in nearly all methods of transports.
- The cost of materials that are made from oil, such as asphalt and chemical products, also rises. If the cost of oil rises, it tends to raise the cost of other fossil fuel.
- The Artificial intelligence may vary by getting the inputs of the daily updates on the other areas such as transportation costs, resource material demands.

10. CONCLUSION

An artificial neural network model is presented with the task of determining the most favourable lag in the crude oil price data. It is evident, the result is shown in the figure, the prediction is accurate till there is a massive and sudden change in the actual data, where it becomes challenging to predict the exact new price with the change, however, the proposed model has efficiently taken into consideration these patterns. Else ways, this also proves the theory that financial markets are unpredictable and change anytime because of known and unknown factors. This work indicates that the ANN model is an effective tool for crude oil price prediction and can be efficiently used for short term price forecasting by determining the optimal lags. 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 original and Predicted Closing prices with time.

11.FUTURE SCOPE

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 advantage of this research is in capturing the changing pattern of these prices. In the coming future, fundamental indicators and market trends have been planned to be incorporated into a model which will help the proposed model perform more efficiently.

12.APPENDIX

GITHUBLINK:

https://github.com/IBM-EPBL/IBM-Project-11132-1659269583

PROJECT DEMO VIDEO LINK:

https://github.com/IBM-EPBL/IBM-Project-11132-1659269583/blob/main/Final%20Deliverable/Output/Demo%20video.mp4