CRUDE OIL PRICE PREDICTION

TEAM ID: PNT2022TMID04569

KONGU ENGINEERING COLLEGE
IBM NALAIYA THIRAN
PROJECT REPORT

Submitted by

Role	Name	Roll Number
Team Leader	DHARSAN S	737819ECR031
Team Member	DHAMODHARAN S	737819ECR026
Team Member	DHARUNASH AS	737819ECR032
Team Member	HARIPRASAD	737819ECR052

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

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

This Guided 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

This project helps the People working in the investment of crude oil needs earlier crude oil price prediction system, which can help them to find the right time to buy crude oil so that they can increase profit from the purchase and reduce any substantial loss

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Crude oil is one of the major products which includes global measurements. The origin of crude oil prediction errors involves composite supply-demand structures. Scientists have come across unique modes for exploring and forecasting crude oil prices. Studies related to the prediction of prices play an important role in the economic crisis. One of the features of the imperfection of all the methodologies was that the upcoming movement of oil price was derived from the prior data. Machine learning strategies came into existence for oil price prediction. In recent times, many studies have given more focus on the Convolutional neural network which is a neural network based on deep learning concepts.

Crude oil price fluctuations have a far-reaching impact on global economies and thus price forecasting can assist in minimizing the risks associated with volatility in oil prices. Price forecasts are very important to various stakeholders: governments, public and private enterprises, policymakers, and investors. 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. Predicting demand for oil is usually straightforward, however supply is heavily affected by political activity such as cartelization by OPEC to regulate prices, technological advances leading to the extraction of higher amounts of oil, and wars and other conflicts which can affect supply unpredictably.

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 Neural Networks: A Literature Survey,” Economics Bulletin, AccessEcon, vol. 35, no. 2, pp. 1339-1359, 2015.

[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 Computing, 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 modelling. 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 Joint Conference 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).

2.3 PROBLEM DEFINITION

The business people who invest or works in the crude oil field and petrol bunk owners requires a way to predict the crude oil price for the next day so that they can take some major investment decisions which can lead to business profit or reduce loss.

Persons working in the investment of crude oil needs a earlier crude oil price prediction system, which can help them to find the right time to buy crude oil so that they can increase profit from the purchase and reduce any substantial loss.

To develop a system that predicts the crude oil prices using LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) and to display the results using python-flask app.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP

The primary purpose of the empathy map is to bridge the understanding of the user and developer. Figure 3.1 represents the empathy map for the Crude oil Price Prediction System.

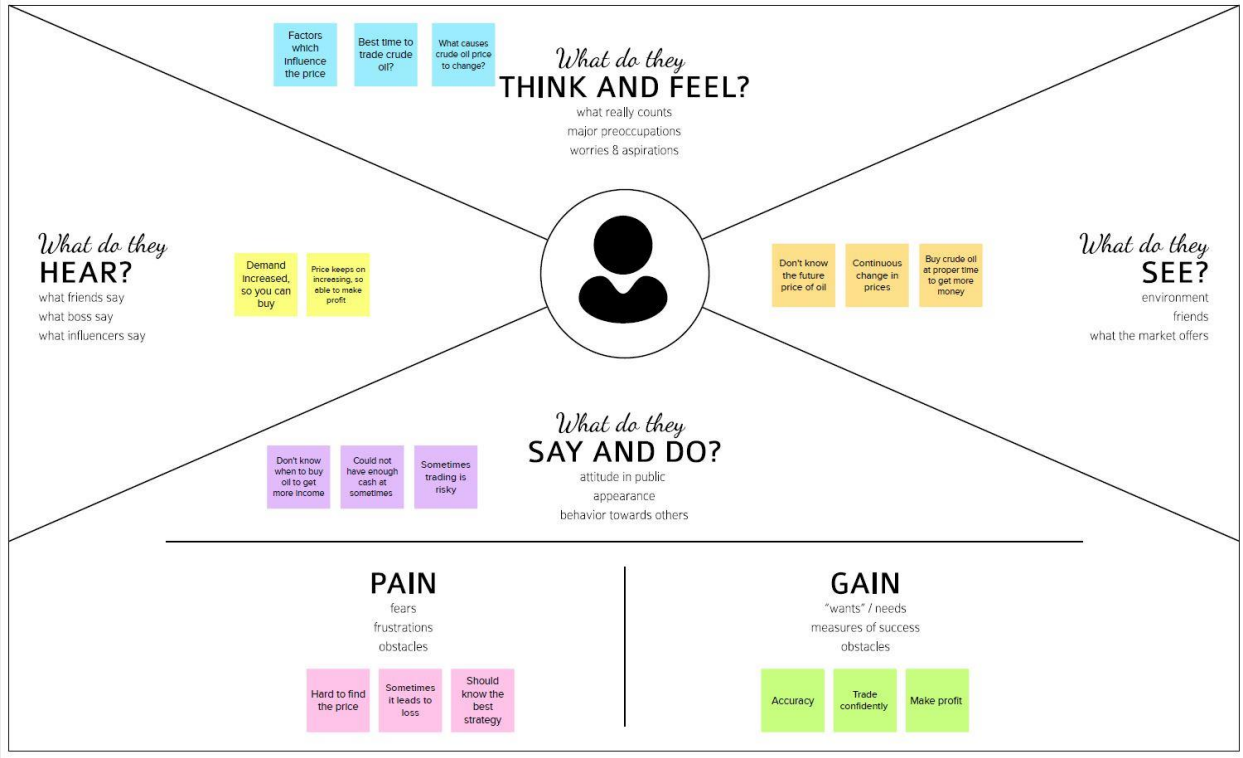


Figure 3.1 – Empathy Map

3.2 Ideation and Brainstorming

This is often the most exciting stage in a project, because during Ideation and brainstorming, the aim is to generate a large quantity of ideas that the team can then filter and cut down into the best, most practical, or most innovative ones to inspire new and better design solutions and products. Figure 3.2 shows the stages of ideation and brainstorming for the Crude oil Price Prediction System.

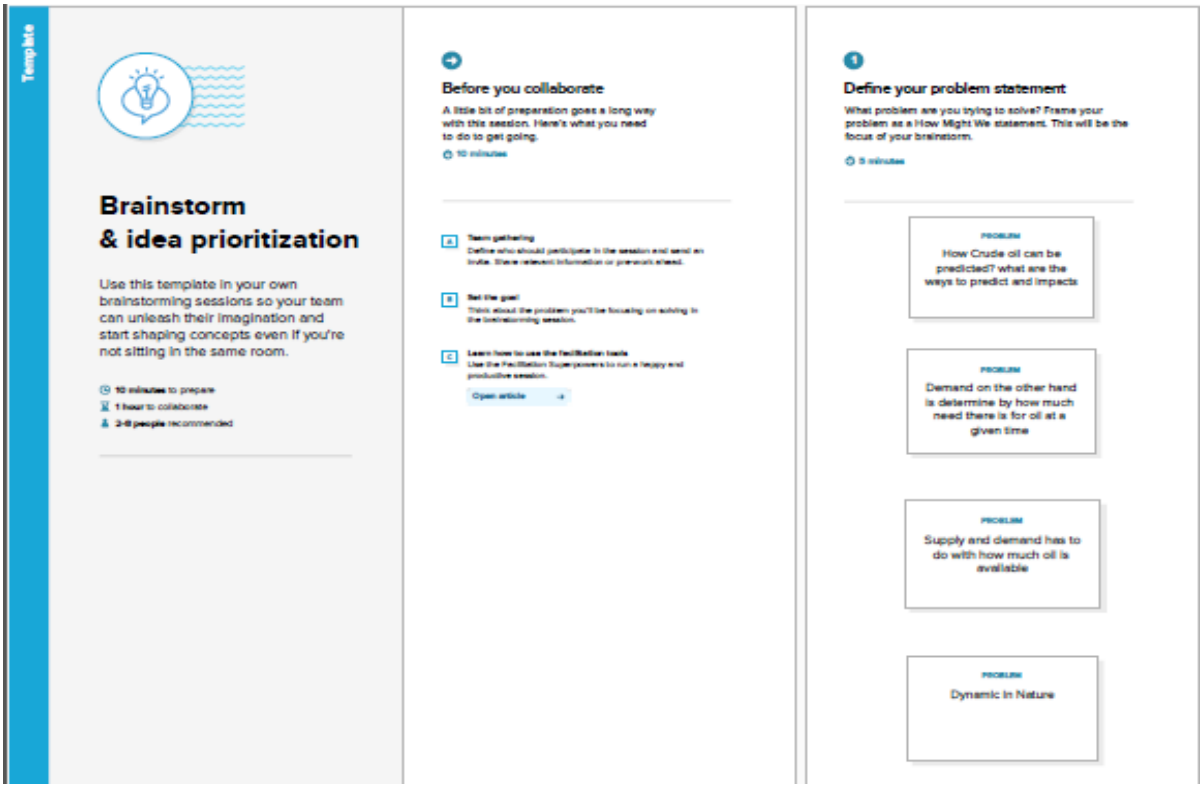


Figure 3.2 – Ideation & Brainstorming

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Predicting the price of the crude oil
2.	Idea / Solution description	By using the Artificial Neural network , RNN (Recurrent neural network) and LSTM algorithms and technology predicting the accurate prices of crude oil.
3.	Novelty / Uniqueness	LSTM performs uniquely well compared to other type of deep learning, including more vanilla RNN. LSTM leads to many more successful runs , and learns much faster. LSTM also solves complex, artificial long time lag tasks that have never been solved by previous recurrent network algorithms.
4.	Social Impact / Customer Satisfaction	As the prices of the crude oil is predicted by using these algorithms , they get accurate results and can invest in their stocks.
5.	Business Model (Revenue Model)	By using this prediction techniques they stakeholders and investors can invest their money without any fear of loss . By without using these stakeholders will face some miserable loss if the prices of the crude oil is unpredictable to them.
6.	Scalability of the Solution	It is well advanced in its technology and algorithms used are very strong hence predict the accurate prices and if there exist any new technology we add those with solution and make our prediction even more better since algorithms used are scalable.

3.4 Problem Solution Fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

Purpose:

❑ Solve complex problems in a way that fits the state of your customers.

❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.

❑ Sharpen your communication and marketing strategy with the right triggers and messaging.

❑ Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.

❑ Understand the existing situation in order to improve it for your target group.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirements

Table 4.1 are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**Graph**	Showing Graph by obtaining the data from the Excel sheet.
FR-2	**News**	Information of all oil prices will be updated by the admin
FR-3	**Database**	Information of the crude oil price will be updated stored in excel sheet

Table 4.1 – Functional Requirements

4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It can use by wide variety of client as it is very simple to learn and not complex to proceed.
NFR-2	**Security**	The information will be hashed so that it will be very secure to use.
NFR-3	**Reliability**	It will be reliable that it can update with very time period so that the accuracy will be good.
NFR-4	**Performance**	It will be performed fast and secure even at the lower bandwidth.
NFR-5	**Availability**	Prediction will be available for every user.
NFR-6	**Scalability**	we are going to use data in excel so it will be easily scalable.

Table 4.2 – Non-Functional Requirements

CHAPTER 5

PROJECT DESIGN

5.1 Dataflow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of how information flows within a system. A neat and clear DFD can thus depict the right amount of the system requirements graphically. It not only shows how data enters and leaves the system, but also what changes the information and where the data is stored. Figure 5.1 represents the DFD for the given project.

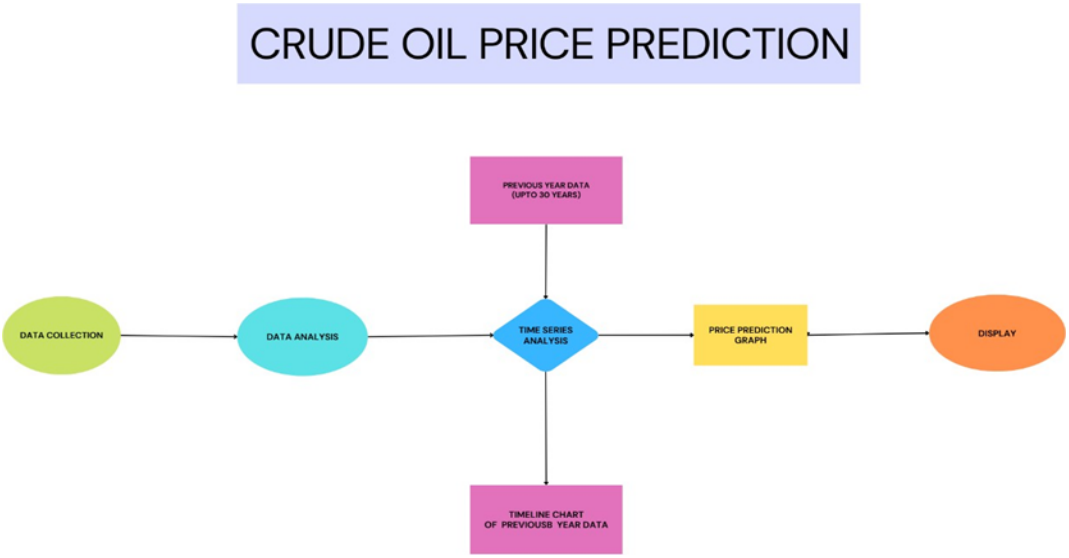


Figure 5.1 – Dataflow Diagram

5.2 Technical Architecture

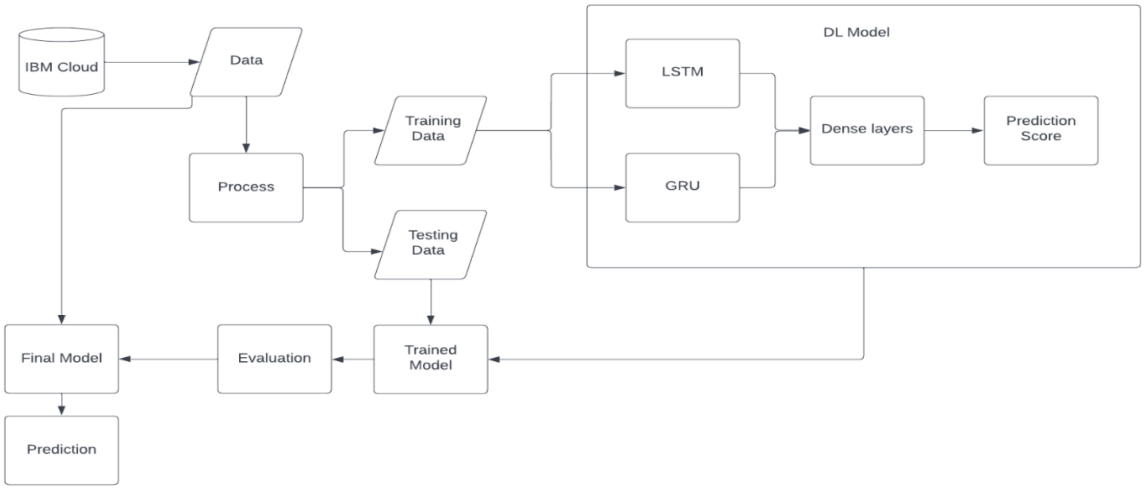


Figure 5.2 Technical Architecture

5.2.1 Component and Technologies

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Flask
2.	Loading data	Converting the csv file to python object	Python
3.	Pre-Processing of data	Pre-Processing and normalizing the data to getaccurate results	Python
4.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
5.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
6.	Machine Learning Model	Long short-term memory (LSTM) is an artificial neural network. Unlike standard feedforward neural networks, LSTM has feedback connections GRU Gated recurrent units is like a long short-termmemory (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate.	Object Recognition Model, etc
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: 2.5Ghz processor, 8GB RAM Cloud Server Configuration: 4 GB GPU	Local, Cloud Foundry, Kubernetes, etc.

Table 5.2.1 – Components and Technologies

5.2.2 Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Flask
2.	Scalable Architecture	Justify the scalability of architecture (3 – tier,Micro-services)	Cloud Foundry, IBM Cloudant

3.	Availability	Justify the availability of application (e.g., use of load balancers, distributed servers etc.)	Cloud Foundry
4.	Performance	Design consideration for the performance of theapplication (number of requests per sec, use of Cache, use of CDN's) etc.	Cloud Foundry

Table 5.2.2 – Application Characteristics

5.3 User Stories

Sprint	Functional requirement	USN number	User story	Story points	Priority
Sprint 1	Mobile Registration	USN 1	As a user, I can register for the application Through mobile number	10	HIGH
Sprint 2	Registration	USN 2	As a user, I can register for the application by entering my email, password, and confirming my password.	10	HIGH
Sprint 3	Internet facility	USN 3	As a user I can give input to the model through the website	15	HIGH
Sprint 4	Access	USN 4	As a user I can access Application using Interactive UI	10	HIGH

Table 5.3 – User Stories

CHAPTER 6

PROJECT PLANNING AND 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	Download Crude Oil Price Dataset	2	Medium	All
Sprint-1	Data Preprocessing	USN-2	Importing The Dataset into Workspace	1	Low	All
Sprint-1		USN-3	Handling Missing Data	3	Medium	All
Sprint-1		USN-4	Feature Scaling	3	Low	All
Sprint-1		USN-5	Data Visualization	3	Medium	All
Sprint-1		USN-6	Splitting Data into Train and Test	4	High	All
Sprint-1		USN-7	Creating A Dataset with Sliding Windows	4	High	All
Sprint-2	Model Building	USN-8	Importing The Model Building Libraries	1	Medium	All
Sprint-2		USN-9	Initializing The Model	1	Medium	All
Sprint-2		USN-10	Adding LSTM Layers and GRU Layers	2	High	All
Sprint-2		USN-11	Adding Output Layers	3	Medium	All
Sprint-2		USN-12	Configure The Learning Process	4	High	All

Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-2		USN-13	Train The Model	2	Medium	All
Sprint-2		USN-14	Model Evaluation	1	Medium	All
Sprint-2		USN-15	Save The Model	2	Medium	All
Sprint-2		USN-16	Test The Model	3	High	All
Sprint-3	Application Building	USN-17	Create An HTML File	4	Medium	All

Sprint-3		USN-18	Build Python Code	4	High	All
Sprint-3		USN-19	Run The App in Local Browser	4	Medium	All
Sprint-3		USN-20	Showcasing Prediction On UI	4	High	All
Sprint-4	Train The Model On IBM	USN-21	Register For IBM Cloud	4	Medium	All
Sprint-4		USN-22	Train The ML Model On IBM	8	High	All
Sprint-4		USN-23	Integrate Flask with Scoring End Point	8	High	All

Table 6.1 – Sprint Planning

6.2 Sprint Delivery Schedule

Sprint	Story Points	Duration (days)	Sprint Start Date	Story Points Completed	Sprint Release Date
Sprint 1	20	6	24 Oct 2022	20	29 Oct 2022
Sprint 2	20	6	31 Oct 2022	20	03 Nov 2022
Sprint 3	20	6	07 Nov 2022	20	10 Nov 2022
Sprint 4	20	6	14 Nov 2022	20	17 Nov 2022

Table 6.2 – Sprint Delivery Schedule

6.3 Reports for JIRA

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)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Figure 6.1 Velocity chart

Burndown Chart:

A burndown 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.

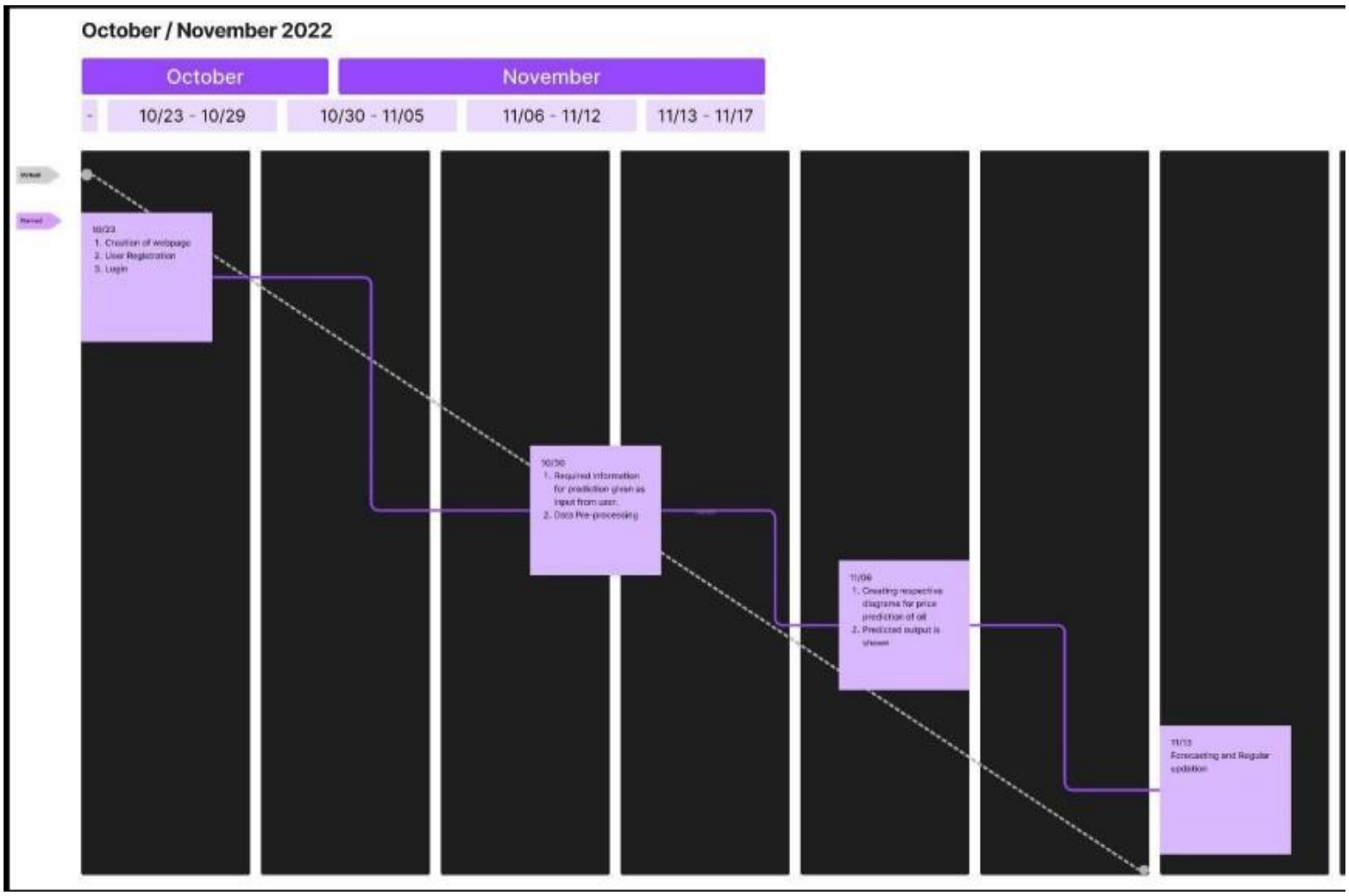


Figure 6.2 Burndown chart

CHAPTER 7

CODING AND SOLUTION

7.1 Feature

FR No.	Feature	Description
FR-1	Crude oil Price Graph	Showing the price of crude oil for respective dates in a graph with dates in x axis and crude oil prices on y axis
FR-2	Current Price Prediction	Showing the last predicted price of the crude oil
FR-3	Prediction based on user provided values	When user provided with the three days prices of the crude oil the application will give predicted price for the next day

Table 7.1 – Description for Feature

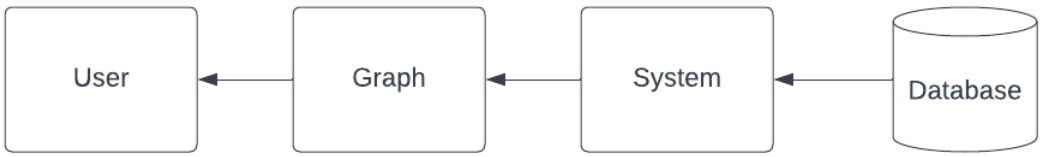


Figure 7.1 – Dataflow Diagram for Feature 1

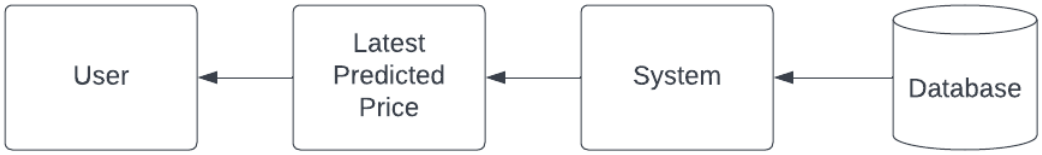


Figure 7.2 – Dataflow Diagram for Feature 2

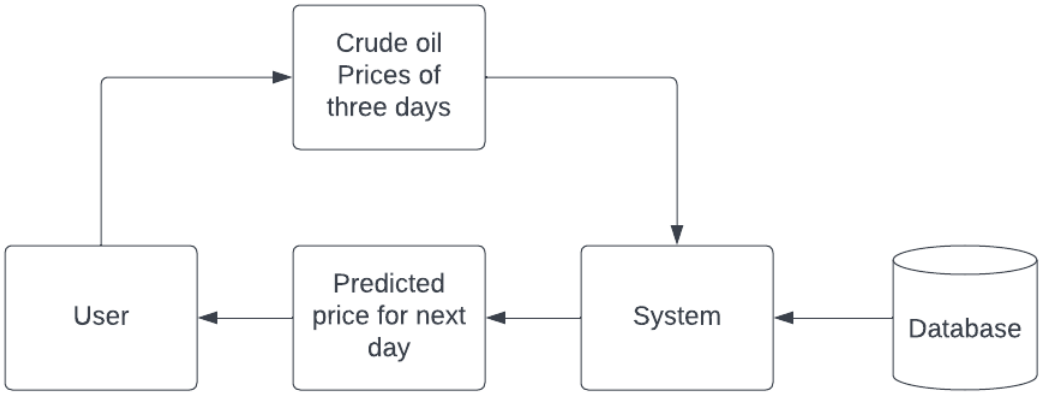


Figure 7. – Dataflow Diagram for Feature 3

CHAPTER 8

TESTING

8.1 Test Cases

The test cases are window of closing prices, where the window size is 3. The test cases are sent to the model and the prediction is compared with the original closing price. The loss metric is used to analyze the performance of the model. Figure 8.1 shows the result after the testing. The blue line in the bottom shows the true closing prices. The orange lines denote the prediction using the training data. The green line denotes the prediction based on testing data.

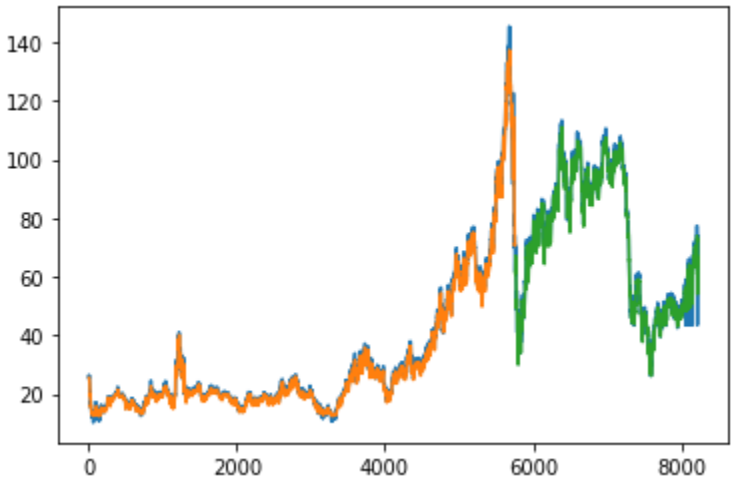


Figure 8.1 – Test Cases Run

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Crude Oil Price Prediction project at the time of the release to User Acceptance Testing (UAT).

2. 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	0	0	2	1	3
Duplicate	1	0	0	0	1
External	0	0	1	0	1
Fixed	0	1	1	0	2
Not Reproduced	0	1	0	0	1
Skipped	0	0	0	0	0
Won't Fix	0	1	0	0	1
Totals	1	3	4	1	9

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
Navigation bar	3	0	0	3
View the current crude oil price	10	0	1	9
View graph of last 100 days prices	5	0	1	4
Enter the prediction prices for 3 days	3	0	0	3
Viewing the predicted output	10	0	0	10

CHAPTER 9

RESULTS

9.1 Performance Metrics

The performance metric used to measure the model is RMSE (Root Mean Square Error). RMSE is measured on both the prediction on training data and the testing data. Lower the RMSE score, better is the accuracy of the model. The results of the RMSE are given in the figure 9.1. From the figure 9.1 it can be observed that the RMSE are quite low. This indicates the model is working better and the predictions are quite accurate.

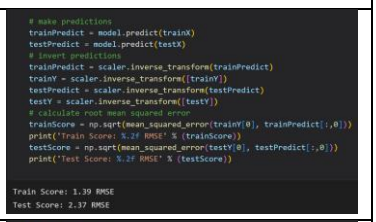
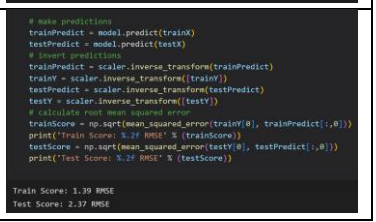
S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** RMSE – 1.39	
2.	Tune the Model	Hyperparameter Tuning – window size = 3 Validation Method – using test set	

Figure 9.1 – Performance Metrics

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

- The application we have created is user friendly
- This application is flexible as user can choose the way they need to predict the price
- User can either give crude oil prices of any three continues dates or can get the latest predicted price of crude oil

10.2 DISADVANTAGES

As the data used in this project is not up to date the prices the model predicted will not be applicable for using in real world crude oil price prediction

CHAPTER 11

CONCLUSION

The prediction system works using the model that is built by combination of LSTM and GRU. The RMSE score for both the training and testing data is quite low. This shows that the accuracy of the model is good. A website is served using flask framework, which helps to enable the users to interact with the model. It helps the user to see the current predicted price the crude oil. And it helps to do prediction for manually entered crude oil closing price values.

CHAPTER 12

FUTURE WORKS

The model currently cannot update the prices to the current data automatically. Web automation can be enables to let the system update its database to current prices. And the model can be retrained on the updated data.

CHAPTER 13

APPENDIX

13.1 Source Code

DATA PREPROCESSING

```
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()
```

HANDLING THE MISSING VALUES

```
data.isnull().any()
data.isnull().sum()
data.dropna(axis=0,inplace=True)
data_oil=data.reset_index()['Closing Value']
data_oil
data.isnull().any()
```

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 VISUALISATION

```
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(da
ta_oil),:1]
training_size,test_size
train_data.shape
```

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)
X_train
X_train.shape
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)
```

MODEL BUILDING

IMPORTING THE MODEL BUILDING LIBRARIES

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
```

INITIALIZING THE 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,y_test),epochs=10,batch_
size=64,verbose=1)
```

MODEL EVALUATION

```
train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
train_predict=scalar.inverse_transform(train_predict)
test_predict=scalar.inverse_transform(test_predict)
import math
from sklearn.metrics import mean_squared_error
math.sqrt(mean_squared_error(y_train,train_predict))
```

SAVE THE MODEL

```
from tensorflow.keras.models import load_model
model.save("crudeoilprediction.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(scalar.inverse_transform(data_oil))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.show()
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):
#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
print (lst_output)
day_new=np.arange(1,11)
day_pred=np.arange(11,21)
len(data_oil)
plt.plot(day_new,scalar.inverse_transform(data_oil[8206:]))
plt.title("Review of prediction")
plt.plot(day_pred,scalar.inverse_transform(lst_output))
plt.show()
df3=data_oil.tolist()
df3.extend(lst_output)
plt.title("Past data nad next 10 days output prediction")
plt.plot(df3[8100:])
df3=scalar.inverse_transform(df3).tolist()
plt.title("Past data nad next 10 days output prediction after reversing the
scaled values")
plt.plot(df3)
```

CODE FOR INDEX.HTML

```
<!DOCTYPE html>
<head>
    <title>Crude Oil Price Prediction </title>
    <link rel="stylesheet" href="index.css">
</head>
<body>
    <h1> Crude Oil Price Prediction</h1>
    <p>   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><br><br>
    <a href="{{url_for('predict')}}"> Predict Future Price</a>
</body>
```

CODE FOR WEB.HTML

```
<!DOCTYPE html>
<head>
   <title>Crude Oil Price Prediction </title>
   <link rel="stylesheet" href="web.css">
</head>
<body>
   <h1>
   Crude Oil Price Prediction </h1>
   <form action="/predict" method="POST" enctype = "multipart/form-
data">
       <input type="text" name="val" placeholder="Enter the crude oil price
for first 10 days" >
        <br> <br> <br>
       <input type="submit"/>
   </form><br> <br>
   <div>
      {{prediction}}
   </div>
   </body>
   </html>
```

CODE FOR INDEX.CSS

```
h1 {
  text-align: center;
          color: floralwhite;
          font-size: 50px;
          font-family: cursive;
}

p {
   font-family: cursive;
          color: ghostwhite;
          margin-right: 30px;
          margin-left: 30px;
```

```
    text-align: center;
            font-size: 20px;
            font-weight: bold;
}
body{
            background: url(back2.jpg);
            background-repeat: no-repeat;
            background-size: 100% 275%;
        }

.button {
    display: inline-block;
    border-radius: 4px;
    background-color: black;
    border: none;
    color: #FFFFFF;
    text-align: center;
    font-size: 20px;
    padding: 12px;
    width: 100px;
    transition: all 0.5s;
    cursor: pointer;
    margin: 5px;
}

a{
    font-size: 20px;
    font-family: cursive;
            color: ghostwhite;
            margin-right: 30px;
            margin-left: 30px;
            text-align: center;
            font-size: 20px;
            font-weight: bold;
}
```

CODE FOR WEB.CSS

```
body{
background:  url(stat.jpg.jpg);background-repeat: no-repeat;background-
size: 100% 275%;
}

h1{
color: white;font-size: 50px;font-family: cursive;
}

input{
border-radius: 18px;;padding: 20px;width: 300px;height: 15px;text-align:
center;align:center;
}

submit{
border-radius: 9px;;padding: 10px;width: 150px;height: 40px;text-align:
center;background: #003d66;font-color: white
}

div{
color:white;font-size:50px;font-family:cursisve;
}
```

CODE FOR APP.PY

```
from flask import Flask,render_template,request,redirect
import numpy as np
from tensorflow.keras.models import load_model
# from tensorflow.k

app = Flask(__name__)

@app.route('/',methods=["GET"])
def index():
    return render_template('index1.html')
```

```
@app.route('/predict',methods=["POST","GET"])
def predict():
    if request.method == "POST":
        string = request.form['val']
        string = string.split(',')
        temp_input = [eval(i) for i in string]

        x_input = np.zeros(shape=(1, 10))
        x_input.shape

        lst_output = []
        n_steps = 10
        i=0
        while(i<10):
            if(len(temp_input)>10):
                x_input = np.array(temp_input[1:])
                x_input = x_input.reshape(1,-1)
                x_input = x_input.reshape((1,n_steps, 1))
                yhat = model.predict(x_input, verbose = 0)
                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)
                temp_input.extend(yhat[0].tolist())
                lst_output.extend(yhat.tolist())
                i=i+1
        val = lst_output[9]
        return render_template('web1.html' , prediction = val)
    if request.method=="GET":
        return render_template('web1.html')
```

```
if __name__=="__main__":
    model = load_model('D:\IBM\Project Development Phase\Sprint 1\Data Collection\crudeoilprediction.h5')
    app.run(debug=True)
```

13.2 GitHub & Project Demo Link

Github link: https://github.com/IBM-EPBL/IBM-Project-22280-1659846200