CRUDE OIL PRICE PREDICTION

NALAIYA THIAN PROJECT REPORT

TEAM ID: PNT2022TMID43298

Submitted by:

DHARSHINI S 715519104011

THARAN S 715519104054

BHUVANESWARI P 715519104009

ANITHA PETRICIA 715519104302

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

Computer Science and Engineering

PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH

Neelambur, Coimbatore - 641062

1. INTRODUCTION

Crude oil, also known as petroleum, is an energy-rich liquid consisting mainly of hydrocarbons. Oil is an important part of daily life in American households and all over the world. This powerful source of energy moves us, heats our homes and creates jobs, and makes up an important component of everyday consumer products. Its components are used to manufacture almost all chemical products, such as plastics, detergents, paints, and even medicines. Depending on the price of crude oil per barrel, the prices for our necessities and luxuries fluctuate. When the cost of crude oil per barrel is high, prices for necessities and luxuries increase. When the cost of crude oil per barrel is low, then the price for which we pay for our necessities and luxuries decreases. Data collected from crude oil prices over time can be analyzed to predict what the prices of crude oil will look like in future years. This gives us insight into how the fluctuation in prices and the fluctuation in the amount of money spent in the future as crude oil prices continue to fluctuate. we have used LSTM-based recurrent neural networks for the purpose of crude oil price prediction. Recurrent neural networks (RNN) have been proven to be one of the most powerful models for processing time-series-based sequential data. LSTM is one of the most successful RNN architectures. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With these memory cells, networks are able to effectively associate memories and input remotely in time, hence suit to grasp the structure of data dynamically over time with high prediction capacity.

1.1. Project Overview

Crude oil price prediction 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 implement RNN(Recurrent Neural Network) with LSTM(Long Short Term Memory) to achieve the task.

1.2. Purpose

Since the demand for crude oil is increasing, there is a rise in its prices. Forecasting the prices would really help in maximizing the profit and excelling in the market.

- The trend that the stock has been following in the previous days, maybe a downtrend or an uptrend.
- The price of the stock on the previous day, because many traders compare the stock's previous day price before buying it.
- The factors that can affect the price of the stock for today. This can be a new company policy that is being criticized widely, or a drop in the company's profit, or maybe an unexpected change in the senior leadership of the company.

2. LITERATURE SURVEY

2.1. Existing problem

The continuous usage of statistical and econometric techniques for crude oil price prediction might demonstrate demotions to the prediction performance.

Hiring experts for market analysis is expensive.

2.2. References

Crude Oil Price Prediction Using Deep Learning–Y. Jeevan Nagendra Kumar; Partapu Preetham; P. Kiran Varma; P. Rohith; P. Dilip Kumar

IEEE Xplore Conferences 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)

Machine learning approach for crude oil price prediction with Artificial Neural Networks-Quantitative (ANN-Q) model–S. N. Abdullah X. Zeng

IEEE Xplore Conferences The 2010 International Joint Conference on Neural Networks (IJCNN)

Text-based crude oil price forecasting: A deep learning approach--XuerongLi WeiShang ShouyangWang

International Journal of Forecasting

CPPCNDL: Crude oil price prediction using complex network and deep learning algorithms–Makumbonori Bristone, Rajesh Prasad ,Adamu AliAbubakar

International Conference on Advanced Science and Engineering (ICOASE)

Oil Price Forecast Using Deep Learning and ARIMA–Junhui Guo

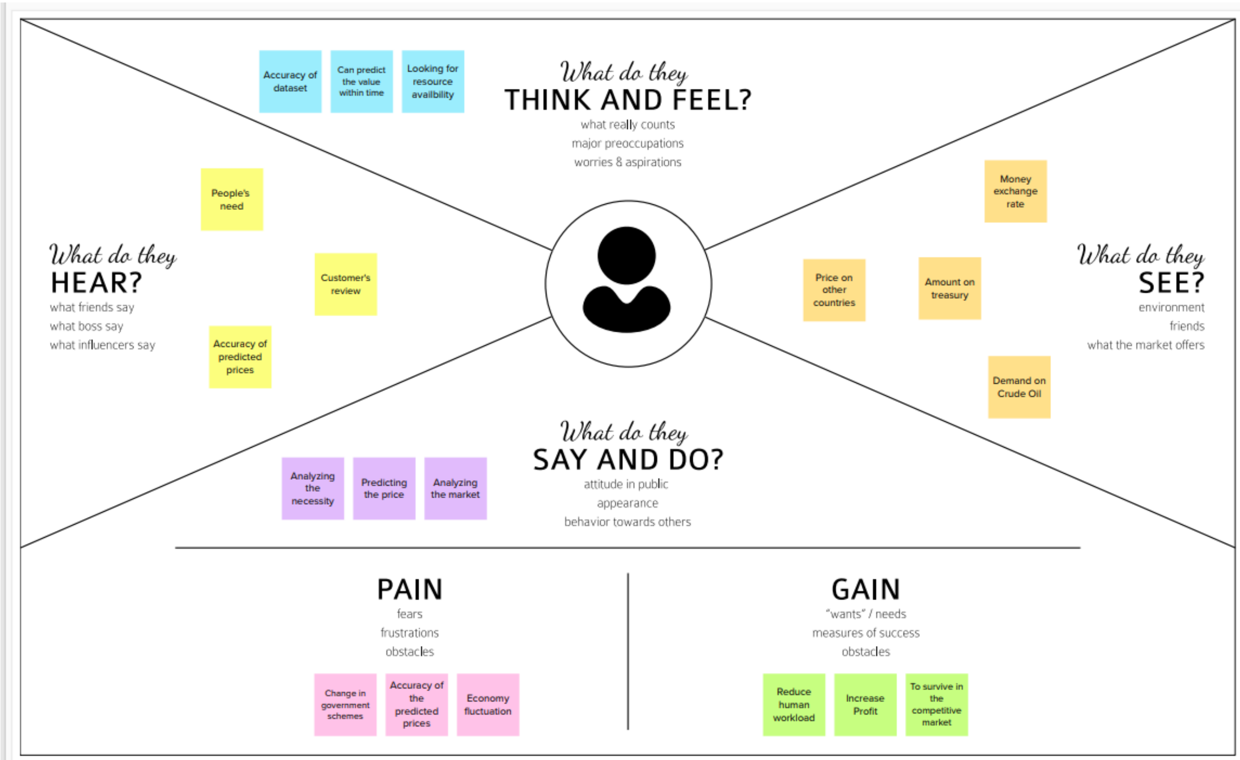
IEEE xplore 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)

2.3. Problem statement definition

To help in buying the crude oil at the proper time by predicting the crude oil price by applying neural networks by implementing recurrent neural network and Long short term memory(LSTM).

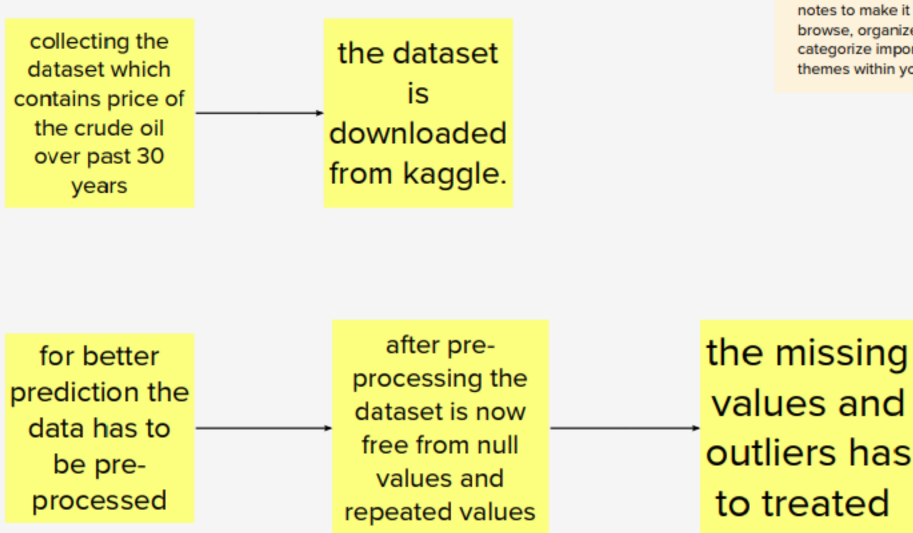
3. IDEATION AND PROPOSED SOLUTION

3.1.Empathy map canvas

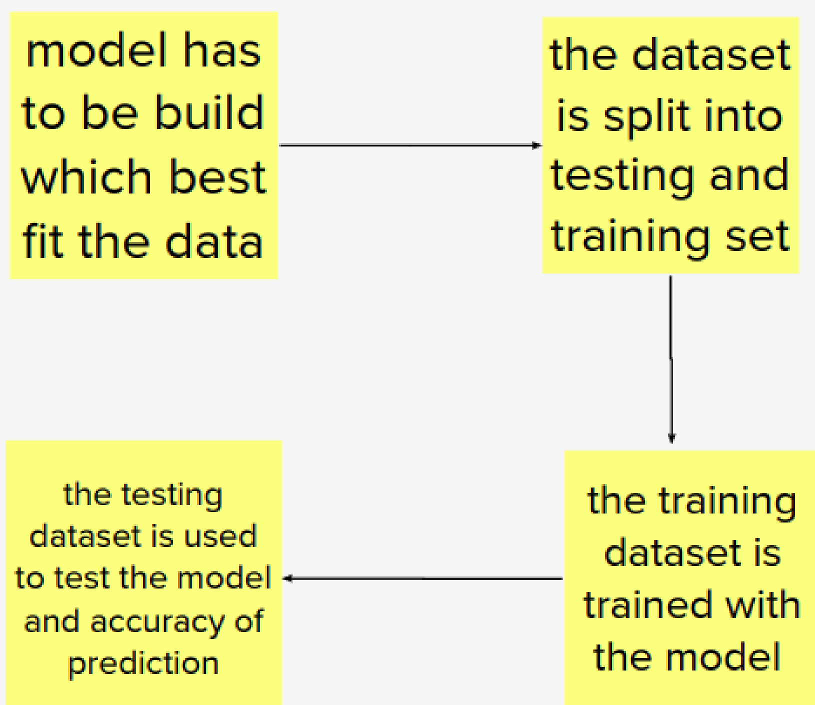


3.2.Ideation and brainstorming

DATA COLLECTION :



MODEL BUILDING AND PREDICTING:



Collect information about the current prices of crude oil in several countries
LSTM and RNN can be used for predicting
LSTM is for deployiing the model
Collecting the past crude oil prices
capacity to handle extremely large dataset
RNN is used for training the model
Updates the database of crude oil price regularly
Exploring several machine learning algorithms
model has to be build which best fit the data
Analyzing the financial market
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 H key on the keyboard.
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3.Proposed solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	This project aims to predict the crude oil price based on the dataset and the current scenario of importing and exporting countries. The dataset contains the price of crude oil of past 30 years.
2.	Idea / Solution description	The idea is to apply the neural networks to predict the crude oil price with more accuracy. Recurrent Neural network (RNN) and Long Short term memory (LSTM) is used as both models are useful in time series prediction.
3.	Novelty / Uniqueness	This project deals with the prediction of crude oil prices using prediction algorithms in neural networks.
4.	Social Impact / Customer Satisfaction	Provides the predicted value faster which helps the customers to buy the crude oil on time. Provides the crude oil price with more accuracy.
5.	Business Model (Revenue Model)	This project mainly focuses to find the powerful model for processing time series data.
6.	Scalability of the Solution	As the dataset is big(contains nearly 30 years of data) the accuracy of the model will be high so that it can be used for large scale purpose.

3.4.Problem Solution fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

CS

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

Oil merchants, investors, traders

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.

Budget constraints, logistics, natural calamities.

5. AVAILABLE SOLUTIONS

AS

Which solutions are available to the customers when they face the pro 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

Predicting the crude oil price using the past data using ML AI forecasting algorithms.

Doing fundamental research

Qualitative techniques using subjective data

pros: easy forecasting of prices without any physical effort, handy results.

cons: No guarantee of accurate results

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.

Predicting the prices before hand and with more accuracy.

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.

Fluctuation in crude oil price, economy fluctuation.

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)

Investors keep an eye on the fluctuation in prices, analyze the market and invest money on it according to the demand and supply.

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

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.

Demand and constant changes in crude oil price, high profit in investment and trading in crude oil.

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.

Before – Insecure about the price fluctuations, manual forecasting may lead to inaccurate predictions.

After – Confident speculation and funding.

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.

In ML, using forecasting algorithms like RNN and LSTM we can predict the price beforehand by processing historic data.

8. CHANNELS of BEHAVIOUR

CH

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

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

Online: User can access this site and resources through internet and use it.
Offline: Investors do manual forecasting from various resources, analyze the market pattern and act accordingly.

4. REQUIREMENT ANALYSIS

4.1. Functional requirements

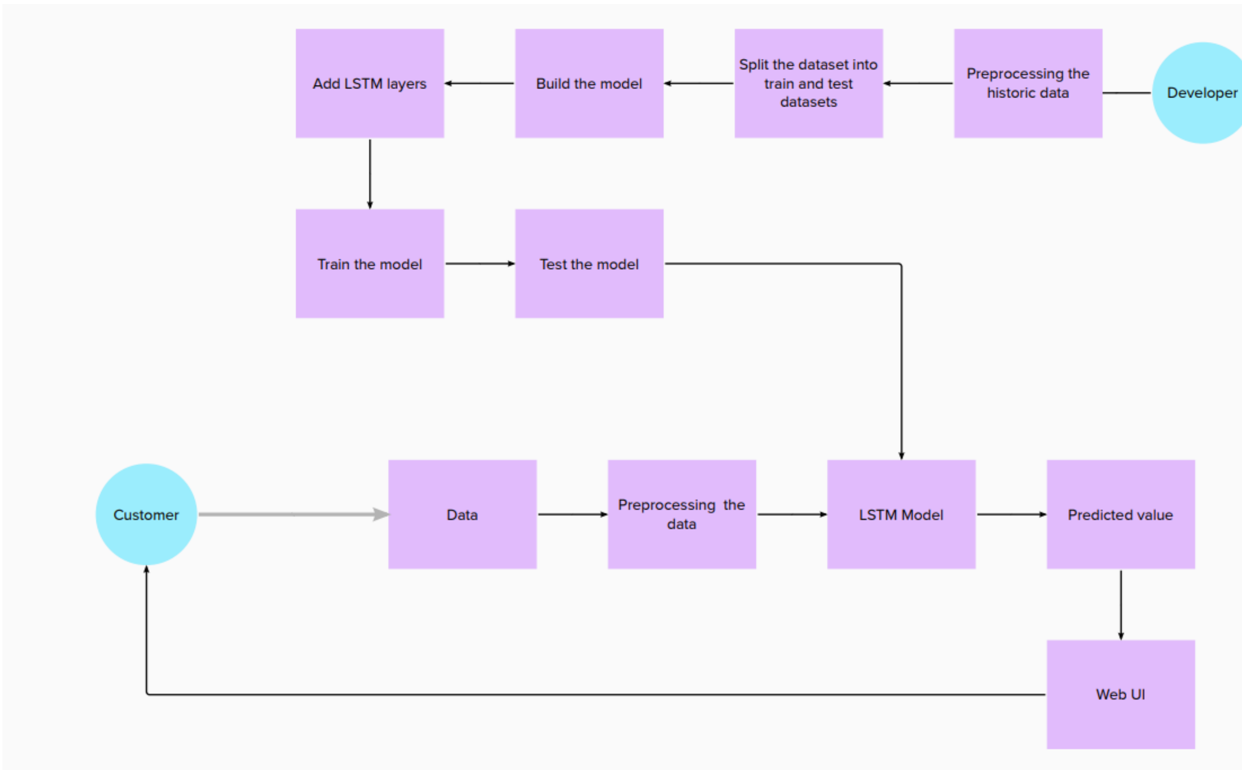
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Confirmation	Confirmation via Email
FR-3	User login	Login via user credentials
FR-4	Primary specifics	The user has to specify the location from where crude oil is to be purchased and the date of purchase.
FR-5	Additional requirements	Users must have a proper internet connection.

4.2. Non functional requirements

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The user is provided with better user interface so that they can access the application effectively.
NFR-2	**Security**	Certain encryption algorithms are used to securely maintain user's data.
NFR-3	**Reliability**	Users review and feedback are taken and is used to maintain reliability.
NFR-4	**Performance**	The accuracy of the prediction will be good irrespective of the size of dataset and number of users.
NFR-5	**Availability**	The application is available to all users at any time.
NFR-6	**Scalability**	As the model is trained with huge data, it is capable of handling growing data without affecting performance.

5. PROJECT DESIGN

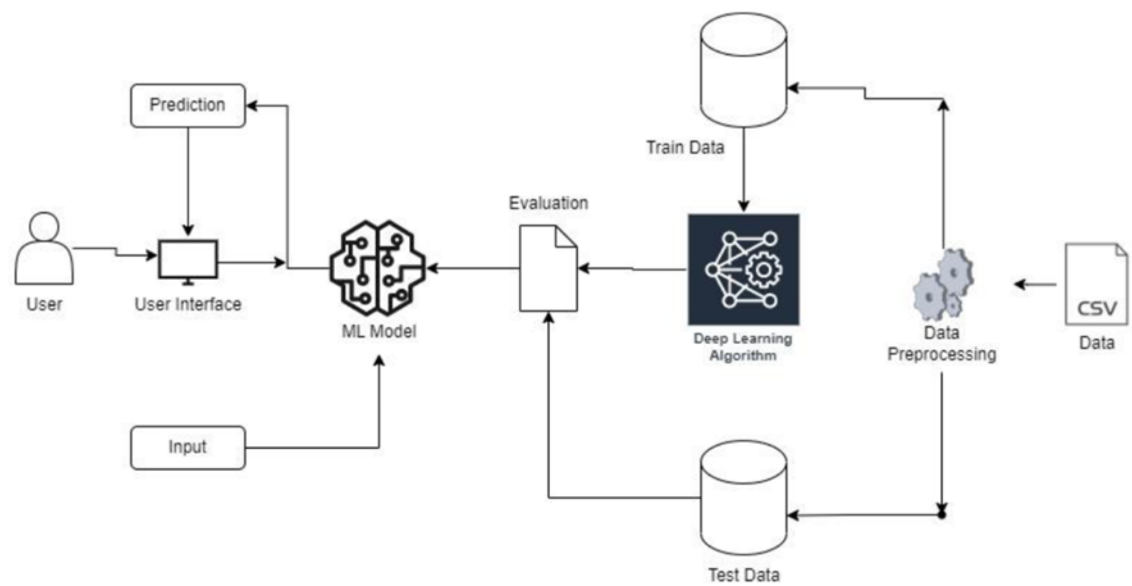
5.1. Data flow diagram



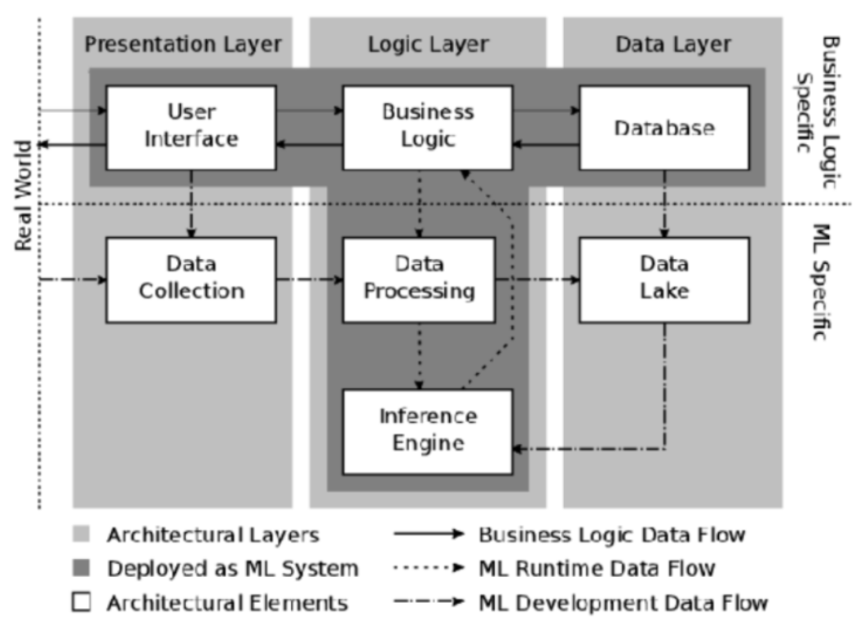
5.2. Solution architecture:

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



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/ Displays Line graph / Bar graph.	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 Gmail	I can register through already logged in Gmail account.	Medium	Sprint-1
	Login	USN-4	As a user I can log into the application by entering email & password	After registration, I can log in by only email & password.	High	Sprint-1
	Line\Bar gragh		After entering the inputs the model will display predictions in Line\Bar Graph Format.	I can get the expected prediction in various formats.	High	Sprint-3
Customer (Web user)	Login	USN-1	As the web user I can login simply by using Gmail account.	Already created Gmail can be used for Login.	Medium	Sprint-2
Customer Care Executive	Support		The Customer care service will provide solutions for any FAQ and also provide Chat Bot.	I can solve the problems raised by Support.	Low	Sprint-3
Administrator	News		Admin will give the recent news of Oil Prices.	Provide the recent oil prices.	High	Sprint-4
	Notification		Admin will notify when the oil prices changes.	Notification by Gmail.	High	Sprint-4
	Access Control		Admin can control the access of users.	Access permission for Users.	High	Sprint-4
	Database		Admin can store the details of users.	Stores User details.	High	Sprint-4

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	Download Crude Oil Price Dataset	2	Medium	Dharshini S
Sprint-1	Data Preprocessing	USN-2	Importing The Dataset into Workspace	1	Low	Bhuvaneswari P
Sprint-1		USN-3	Handling Missing Data	3	Medium	Tharan S
Sprint-1		USN-4	Feature Scaling	3	Low	Anitha Petricia
Sprint-1		USN-5	Data Visualization	3	Medium	Dharshini S
Sprint-1		USN-6	Splitting Data into Train and Test	4	High	Tharan S
Sprint-1		USN-7	Creating A Dataset with Sliding Windows	4	High	Dharshini S
Sprint-2	Model Building	USN-8	Importing The Model Building Libraries	1	Medium	Anitha Petricia
Sprint-2		USN-9	Initializing The Model	1	Medium	Bhuvaneswari P
Sprint-2		USN-10	Adding LSTM Layers	2	High	Bhuvaneswari P
Sprint-2		USN-11	Adding Output Layers	3	Medoum	Tharan S
Sprint-2		USN-12	Configure The Learning Process	4	High	Anitha Petricia
Sprint-2		USN-13	Train The Model	2	Medium	Dharshini S
Sprint-2		USN-14	Model Evaluation	1	Medium	Tharan S
Sprint-2		USN-15	Save The Model	2	Medium	Bhuvaneswari P
Sprint-2		USN-16	Test The Model	3	High	Anitha Petricia
Sprint-3	Application Building	USN-17	Create An HTML File	4	Medium	Tharan S
Sprint-3		USN-18	Build Python Code	4	High	Tharan S
Sprint-3		USN-19	Run The App in Local Browser	4	Medium	Dharshini S
Sprint-3		USN-20	Showcasing Prediction On UI	4	High	Bhuvaneswari P
Sprint-4	Train The Model On IBM	USN-21	Register For IBM Cloud	4	Medium	Anitha Petricia
Sprint-4		USN-22	Train The ML Model On IBM	8	High	Dharshini S
Sprint-4		USN-23	Integrate Flask with Scoring End Point	8	High	Tharan S

6.2. Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	29 Oct 2022	1 Nov 2022	20	10 Nov 2022
Sprint-2	20	3 Days	2 Nov 2022	7 Nov 2022	20	13 Nov 2022
Sprint-3	20	6 Days	8 Nov 2022	10 Nov 2022	20	19 Nov 2022
Sprint-4	20	3 Days	11 Nov 2022	14 Nov 2022	20	22 Nov 2022

`7. CODING AND SOLUTIONING

7.1 Feature 1

This UI takes the price values of the past 10 days, analyzes and predict the next day's price. To achieve this benchmark, this project uses the LSTM model for forecasting crude oil prediction. LSTM is a time series model that can look at the history of a sequence of data and correctly predict the future sequence. The LSTM network enables faster and better training on long sequence data and avoids the problem of vanishing gradients.

Code:

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

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_3 (LSTM)               (None, 10, 50)            10400

 lstm_4 (LSTM)               (None, 10, 50)            20200

 lstm_5 (LSTM)               (None, 50)                20200

 dense_1 (Dense)             (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

7.2 Feature 2

Provides crude oil live stock screener and fundamental analysis tools, candlestick charts for market pattern analysis and so on. When It Comes To Technical Analysis Of Stock Charts Smart investors who want to buy stocks will make sure that they have done all the research possible to ensure that the stock will see an increase in price over time.

<u>Code:</u>

```
<div class="tradingview-widget-container" style="margin-top: 10px;">
        <div id="tradingview_47972"></div>
        <!-- <div class="tradingview-widget-copyright"><a href="https://in.tradingview.com/symbols/CURRENCYCOM-OIL_CRUDE/" rel="noopener" target="_blank"><span class="blue-text">OIL_CRUDE Chart</span></a></div> -->
        <script type="text/javascript" src="https://s3.tradingview.com/tv.js"></script>
        <script type="text/javascript">
        new TradingView.widget(
        {
            // "autosize": true,
            "height": 500,
            "width" : 980,
            "symbol": "CURRENCYCOM:OIL_CRUDE",
            "interval": "D",
            "timezone": "Asia/Kolkata",
            "theme": "dark",
            "style": "2",
            "locale": "in",
            "toolbar_bg": "#f1f3f6",
            "enable_publishing": false,
            "allow_symbol_change": true,
            // "container_id": "tradingview_47972"
        }
        );
        </script>
    </div>
```

8.TESTING

8.1Test cases

Component	**Test Scenario**	**Prerequisite**	**Steps To Execute**	**Test Data**	**Expected Result**
Dataset collection	verify the presence of data	kaggle account	download the required dataset from the link provided	Dataset file	data must be present
Importing dataset	Verify dataset loading	jupyter notebook /google colab	Import dataset	Dataset file	dataset is imported
Feature scaling	Verify whether dataset is ranged as 0-1	loaded dataset	use minmaxscaler to convert	Dataset file	dataset is converted
Training and testing	Verify the splitting of dataset	Loaded Dataset	split 80% as training and remaining as testing dataset	Dataset file	Splitting is correct
Model building	Verify the model builded	Loaded Dataset	create a LSTM model with needed layers	loaded model	model builded correctly
Model training	Train the model	Loaded Model	Train using model and dataset loaded	Dataset file	Training the model with better accuracy
Model evaluation	Verify the error ratio	Loaded Model	Predict using model	Input data	model must be with less error rate
Model testing	Verify predictions	Loaded Model	Predict using model	Input data	model must predict the

					price of 10th day
Importing necessary libraries	verify whether the necessary libraries requirement are satisfied	Python and pip	install the libraries using the pip module	Libraries	Libraries installed
Load the model	Verify the model loaded	The trained model	Import the model	loaded model	Model loaded successfull y
Routing	Route the web pages using flask	Flask module	render the necessary templates	HTML files	Display of web pages
Run the server	Run the flask server	Flask module	Execute using python	Input data	Predicted value
Model	Verify COS creation	IBM Account	create a cloud object storage bucket	Dataset file	Bucket is created
Model	Verify dataset loading	COS dataset object	import dataset from COS	Dataset file	COS dataset is imported
Model	Verify if Model is deployed	Watson ML service	check namespace logs	Pickled model	model is pushed to namespace
Model	Verify predictions	Loaded model	predict using model	Input data	prediction is correct

8.2 User Acceptance Testing

1. **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	**12**	**2**	**5**	**1**	**20**
Duplicate	**1**	**2**	**3**	**1**	**7**
External	**0**	**3**	**0**	**1**	**4**
Fixed	**10**	**2**	**5**	**20**	**37**
Not Reproduced	**1**	**0**	**1**	**1**	**3**
Skipped	**0**	**0**	**1**	**1**	**2**
Won't Fix	**0**	**3**	**2**	**1**	**6**
Totals	**24**	**12**	**17**	**26**	**79**

2. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	**3**	**0**	**0**	**3**
Client Application	**2**	**0**	**0**	**2**
Security	**2**	**0**	**0**	**2**
Outsource Shipping	**0**	**0**	**0**	**0**
Exception Reporting	**1**	**0**	**0**	**1**
Final Report Output	**4**	**0**	**0**	**4**
Version Control	**0**	**0**	**0**	**0**

9. ADVANTAGES AND DISADVANTAGES:

ADVANTAGES

As a very powerful prediction tool, LSTM has been widely used in prediction-related fields. Therefore, to forecast crude oil prices more accurately, we have selected the LSTM model for this study. Usage of LSTM model which helps in predicting accurate results of the next day price using the past 10 days' data, which also reduces the manual workforce of the analysts/customers.

The different gates inside LSTM boost its capability for capturing nonlinear relationships for forecasting. Causal factors generally have a non-linear impact on demand. When these factors are used as part of the input variable, the LSTM could learn the nonlinear relationship for forecasting.

DISADVANTAGES

The prices of crude oil not only depend on past data but also on external factors as well(like GDP, demand and supply, etc.). As the predicted value depends upon the past 10 days' data, the rightness of the prediction may divert.

10. CONCLUSION

LSTM model predicts the price with better accuracy while compared to other conventional models. As LSTM model works better with time series data. LSTM has the property of selectively remembering patterns for long durations of time.Unlike standard feedforward neural networks, LSTM has feedback connections. Such a recurrent neural network (RNN) can process not only single data points (such as images) but also entire sequences of data.

11. FUTURE SCOPE

The current project only deals with the past 10 days of data which would not be righteous all the time. And there are many external factors that influence the crude oil market. In future enhancements, we would like to include and train the model with external factors(demand and supply GDP, etc.) as well. And this project only deals with the number range between 0 to 1, which will be rectified in future enhancements.

12. APPENDIX

Source code:

Crude_oil_price_prediction.ipynb

```
# -*- coding: utf-8 -*-
"""crude_oil_price_prediction-checkpoint.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1xvGvKCQMpYu_rfyONZN6M_SK4F4LUmGI
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import os

data=pd.read_excel(r"D:\Crudeoil price prediction\Crude Oil Prices Daily.xlsx")
data.head()

data.isnull().any()

data.isnull().sum()

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

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

data.head()

from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
data=scaler.fit_transform(np.array(data).reshape(-1,1))
```

```
data

len(data)

plt.plot(data)
plt.xlabel("Days")
plt.ylabel("Price")
plt.show()

training_size=int(len(data)*0.80)
testing_size=len(data)-training_size
train_data,test_data=data[0:training_size],data[training_size:len(data)]

train_data.shape
len(test_data)

def create_dataset(dataset,timestep):
    dataX,dataY=[],[]
    for i in range(len(dataset)-timestep-1):
        a=dataset[i:(i+timestep),0]
        dataX.append(a)
        dataY.append(dataset[i+timestep,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)

y_train=y_train.reshape(-1,1)
y_test=y_test.reshape(-1,1)

print(x_train.shape)
print(y_test.shape)

from tensorflow.keras.models import Sequential
```

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

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')

history=model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=50,batch_size
=64,verbose=1)

train_predict=model.predict(x_train)
test_predict=model.predict(x_test)

train_predict=scaler.inverse_transform(train_predict)
y_train=scaler.inverse_transform(y_train)
test_predict=scaler.inverse_transform(test_predict)
y_test=scaler.inverse_transform(y_test)

import math
from sklearn.metrics import mean_squared_error
math.sqrt(mean_squared_error(y_train,train_predict))

plt.figure(figsize=(8,4))
plt.plot(history.history['loss'],label='Train loss')
plt.plot(history.history['val_loss'],label='Test loss')
plt.show()

from tensorflow.keras.models import load_model
```

```
model.save("crude_oil_forecasting.h5")

# Saving in tar

!tar -zcvf crude_oil.tgz crude_oil_forecasting.h5

!pip install watson-machine-learning-client

!pip install ibm_watson_machine_learning

from ibm_watson_machine_learning import APIClient

wml_credentials ={
    "url":"https://eu-gb.ml.cloud.ibm.com",
    "Apikey": API_KEY
}

client = APIClient(wml_credentials)
client

client.spaces.get_details()

def guid_space_name(client,crudeoil):
  space = client.spaces.get_details()
  return(next(item for item in space['resources'] if
item['entity']['name']==crudeoil)['metadata']['id'])

space_uid = guid_space_name(client,'Crude oil prediction')
space_uid

client.set.default_space(space_uid)

client.software_specifications.list()

software_space_uid =
client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
```

```
software_space_uid

model_details = client.repository.store_model(model='crude_oil.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"Crude oil prediction",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})

model_details

model_id = client.repository.get_model_id(model_details)
model_id

client.repository.download(model_id,'fetch.tar.gz')
```

Main.py

```
import numpy as np
import json
from flask import Flask, render_template, request

from tensorflow.python.keras.models import load_model

app = Flask(__name__)
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM
Cloud account.
API_KEY = API_KEY
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# model = load_model('crude_oil_forecasting.h5')
ent = []

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

@app.route('/predict', methods=["GET", "POST"])
def home1():
    if request.method == "POST":
        data = request.form['data']
        # print(data)
        ent.append(data)
        inp = str(request.form['data'])
        inp = inp.split(',')
        for i in range(0, len(inp)):
```

```
            inp[i] = float(inp[i])
        inp = np.array(inp).reshape(1,-1)
        temp = list(inp)
        temp = temp[0].tolist()
        # print(temp)
        outpt = []
        n_steps = 10
        i=0
        while(i<1):
            if(len(temp) > 10):
                #some prblm
                inp = np.array(temp[1:])
                print("{} day input {}".format(i, inp))
                inp = inp.reshape(1,-1)
                inp = inp.reshape((1, n_steps, 1))

                payload_scoring = {"input_data": [{"fields": ["price"], "values":
inp}]}
                response_scoring =
requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2006243c-d8cb-4ad4-9f7
9-69698be3f014/predictions?version=2022-11-18', json=payload_scoring,
                headers={'Authorization': 'Bearer ' + mltoken})
                print("Scoring response")
                yhat = response_scoring.json()

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

                payload_scoring = {"input_data": [{"fields": ["price"], "values":
inp}]}
```

```
                response_scoring =
requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2006243c-d8cb-4ad4-9f7
9-69698be3f014/predictions?version=2022-11-18', json=payload_scoring,
                headers={'Authorization': 'Bearer ' + mltoken})
                print("Scoring response")
                yhat = response_scoring.json()
                outpt.extend(yhat.tolist())
                temp.extend(yhat[0].tolist())
                outpt.extend(yhat.tolist())
                i+=1
        return render_template("predict_opt.html", entries = outpt)
    else:
        return render_template("predict_opt.html")

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

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

GITHUB Link:

[IBM-EPBL/IBM-Project-39189-1660399864: Crude Oil Price Prediction (github.com)]

DEMO Link:

https://drive.google.com/file/d/1eYJ2r_2MugR3-NFB-R9Ygz005q-OM-sH/view?usp=share_link