CRUDE OIL PREDICTION

PROJECT REPORT

Team Id: PNT2022TMID00817

Submitted by

MUTHU ANNAMALAI.V	**211419104171**
KOULURU NANDA KISHORE REDDY	**211419104142**
MUKESH KUMAR. S	**211419104169**
PATHAKAMURI CHARAN KUMAR	**211419104187**

Panimalar Engineering College

November-2022

Table of Content

1. Introduction

1.1 Project Overview

Due to the inelasticity of oil demand, the rise in price will increase revenue for producers. However, oil importers will have to pay more for their oil purchases. Due to the fact that oil is the most traded commodity, the effects are quite significant. Rising oil prices can even shift economic/political power from oil importers to oil exporters. Diverse factors influence crude oil price movements.

The purpose of this Guided Project is to apply Neural Networks to predict the Crude Oil Price. By making this decision, we are able to buy crude oil at the right time. This kind of prediction is best made using time series analysis since we are predicting crude oil prices based on their past history. As a result, we will implement RNNs (Recurrent Neural Networks) with LSTMs (Long Short Term Memory).

Price forecasting can help minimise risks associated with volatility in oil prices by reducing the impact of price fluctuations on global economies. Investors, policymakers, and governments all rely on price forecasts for various purposes

1.2 Purpose

The cost of crude oil affects the global habitat, our economy, oil exploration, exploitation, and other activities directly, making it one of the most important resources in today's world. Many large and small industries, individuals, and governments rely on the prediction of oil prices, which has become a necessity.

Due to crude oil's evaporative nature, predicting its price becomes extremely difficult and inaccurate. Crude oil prices are affected by a variety of factors. By using artificial neural networks (ANNs), we propose a contemporary and innovative method for predicting crude oil prices.

This approach is able to continuously capture the unstable pattern of crude oil prices that has been incorporated by finding the optimal lag and number of the delay effect that controls crude oil prices.

After varying the lag over time for close and optimum results, we evaluated the root mean square error to validate our results. The results obtained using the proposed model were significantly better than those obtained using the previous model.

2. Literature Survey

2.1 Existing Problem

Crude oil, one of the world's most important commodities, accounts for one-third of global energy consumption. From transportation fuels to plastics, it is a starting material for most products we use every day. Price forecasting can assist in minimizing risks associated with oil price volatility by reducing the impact of fluctuations on global economies.

Various stakeholders, including governments, public and private enterprises, policymakers, and investors, rely heavily on price forecasts. From the equilibrium between demand and supply, crude oil's price should be predictable based on economic theory, where demand forecasts are usually derived from GDP, exchange rates, and domestic prices, while supply forecasts are derived from past production and reserve data.

Usually, oil demand can be predicted easily, but supply is heavily affected by political activity, such as cartelization by OPEC to regulate prices, technological advancements that allow more oil to be extracted, and wars and other conflicts that can affect supply unpredictably.

Economic models that incorporate supply and demand parameters and their determinants are referred to as structural models. Despite structural models being the most logical way to model industrial product prices, crude oil is influenced by many other factors. Among these factors is the fact that crude oil's price is determined by the futures market, which allows the purchase of a predefined amount of oil at a specified price in the future.

2.2 References

1. Sehgal, N.; Pandey, K.K. Artificial intelligence methods for oil price forecasting: A review and evaluation. Energy Syst.2015,6,479–506.
2. Shuang Gao, Yalin Lei. "A new approach for crude oil price prediction based on stream learning" , Geoscience Frontiers, 2017
3. Bashiri Behmiri, Niaz and Pires Manso, José Ramos, Crude Oil Price Forecasting Techniques: A Comprehensive Review of Literature
4. Kulkarni, S., Haidar, I., 2009. Forecasting model for crude oil price using artificial neural networks and commodity future prices. International Journal of Computer Science and Information Security 2 (1)
5. Alquist, R., Kilian, L., Vigfusson, R.J., 2013. Forecasting the price of oil. In: Elliott G
6. Bifet, A., Kirkby, R., Kranen, P., Reutemann, P., 2012. Massive online analysis manual
7. N. Raj Kiran, V. Ravi. "Software reliability prediction by soft computing techniques" , Journal of Systems and Software, 2008
8. Lean Yu. "An EMD-Based Neural Network Ensemble Learning Model for World
9. Crude Oil Spot Price Forecasting" , Studies in Fuzziness and Soft Computing, 2008

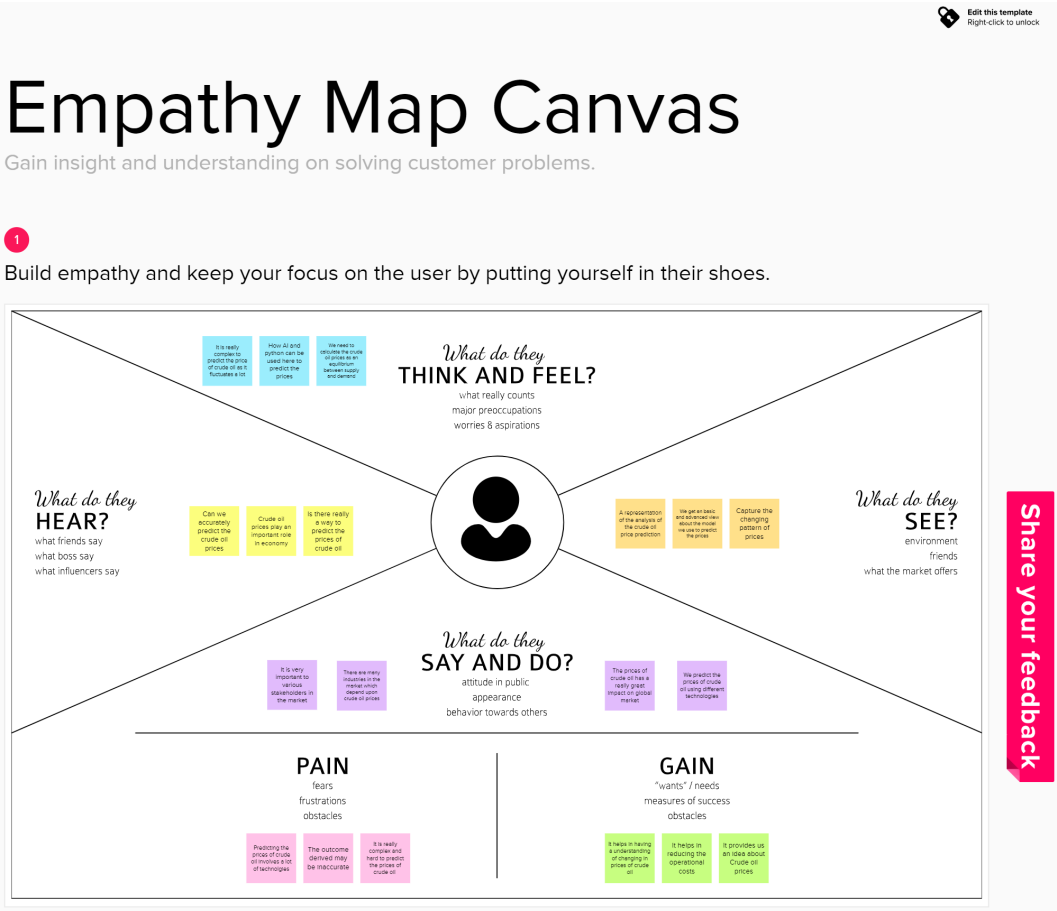
2.3 Problem Statement Definition

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. Ideation & Proposed Solution

3.1 Empathy Map Canvas

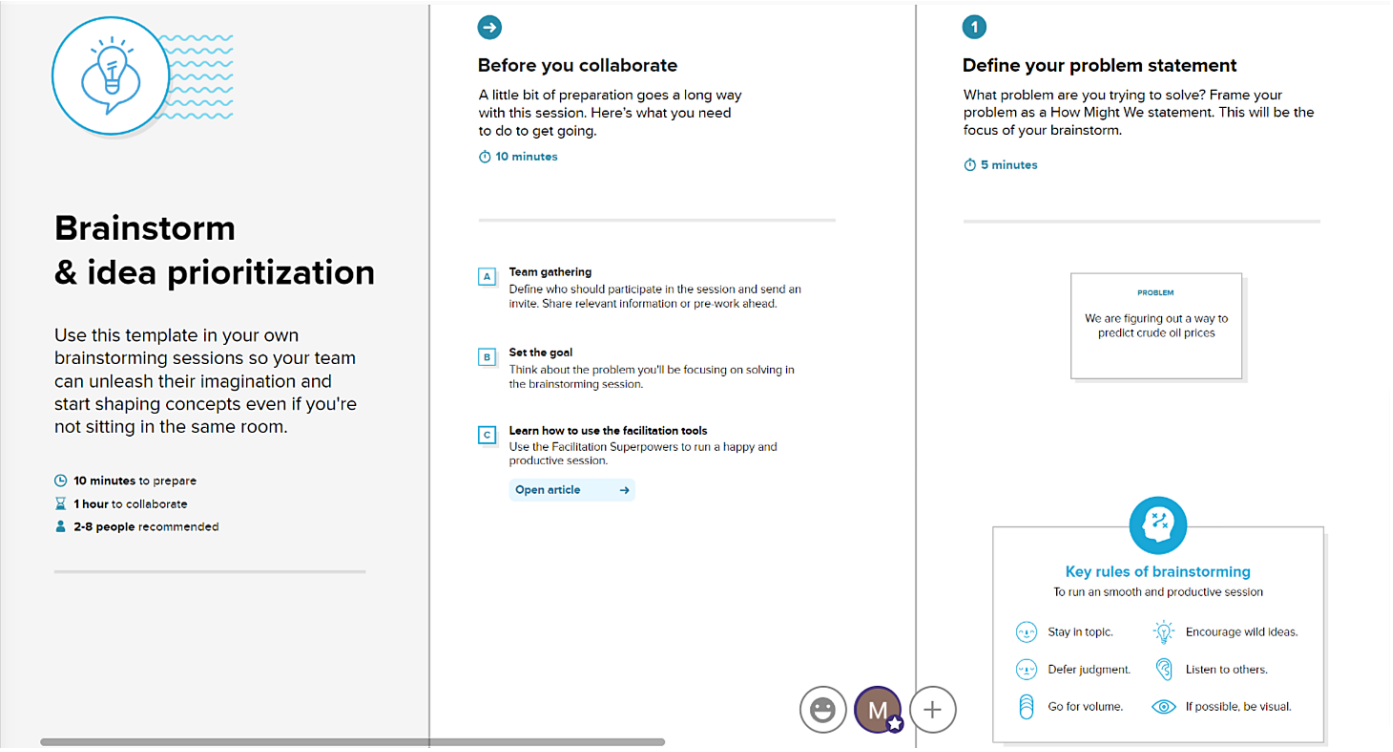
Empathy maps are simple easy-to-digest visuals that convey knowledge about a user's attitude and behavior. Team members can use it to better understand their users with the help of this tool. Understanding the true problem and the person experiencing it is essential for creating an effective solution. Through the process of creating the map, participants gain a better understanding of how the user sees the world, his or her goals, and the challenges he or she faces.



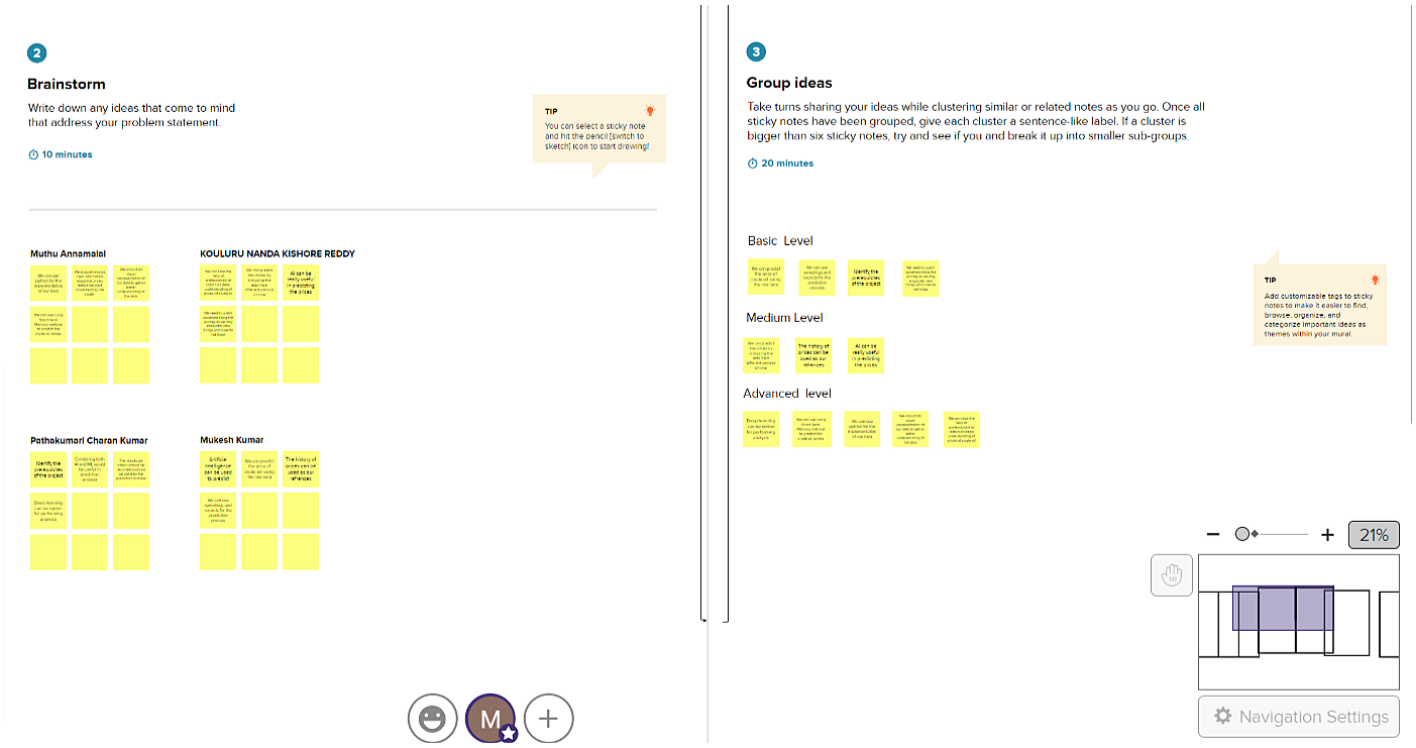
3.2 Ideation & Brainstorming

Brainstorming fosters a creative thinking process that leads to problem solving by providing a free and open environment. We prioritise volume over value, welcome out-of-the-box ideas, and encourage all participants to collaborate in order to develop a rich amount of creative solutions.

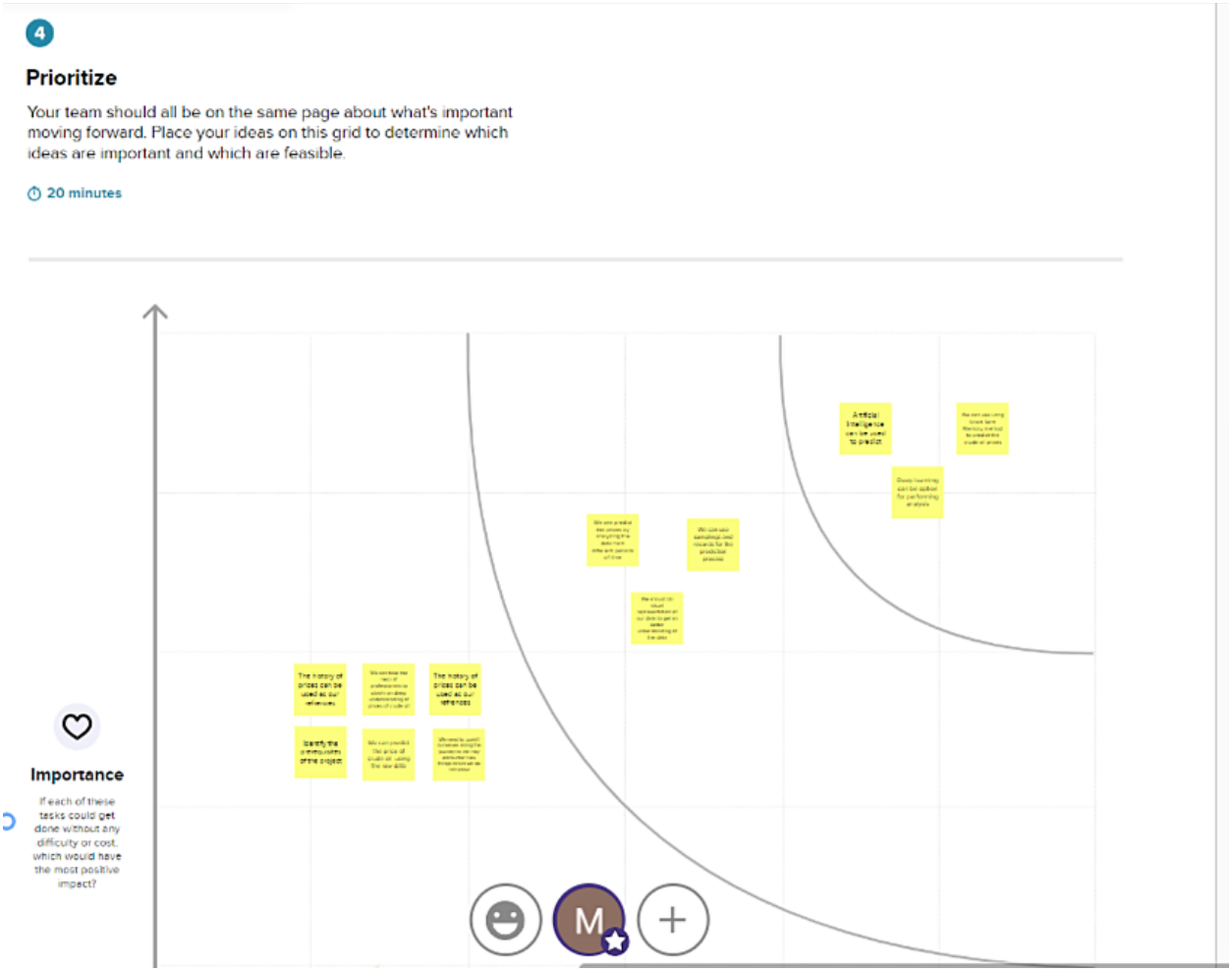
Step 1: Team Gathering, Collaboration and Select the Problem Statement



Step 2: Brainstorm, Idea Listing and Grouping



Step 3: Idea Prioritization



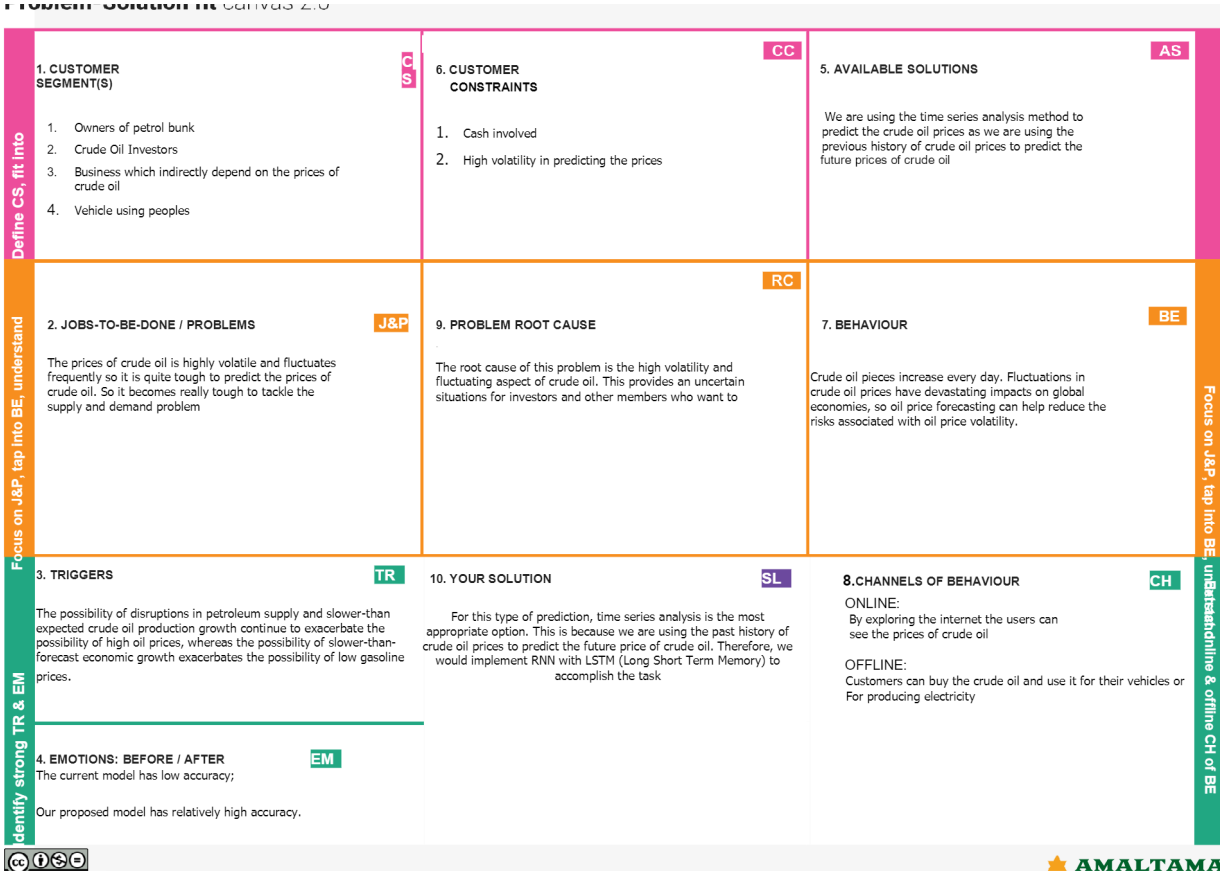
3.3 Proposed Solution

S.NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	Oil prices play a majorrole in determining the global economies but determining it is really tough. We are trying to solve this
2.	Idea/Solution description	The crude oil price can be easilypredicted by usingthe counterbalance between supply, and demand using past data. We are predicting it using Python, RNN, and Deep learning. We have triedto incorporate timeseries analysis methodto predict the prices of crude oil
3.	Novelty / Uniqueness	The main objective of our project is to apply Neural Networks to predict crude oil prices. By making this decision, we can buy crude oil at the righttime. In orderto make this kind of prediction, time series analysis is the best option since we are using the past history of crude oil prices to make predictions about the future. To accomplish the task, we would implement RNN (Recurrent Neural Network) with LSTM (LongShort Term Memory).
4.	Social Impact/ Customer Satisfaction	Oil price changes have important consequences for the global economy, so we try to use our model to predict it in order to help the economy and businesses aroundthe world.
5.	Business Model (Revenue Model)	Our revenuemodel is focused on 1. Pay per month model 2. Pay per year model

6.	Scalability of the Solution	The time series analysis method is used to predict crude oil prices on the basis of previous historical data. We believe that we can provide better and more accurate predictions of crude oil prices, so we tend to satisfy the customer, which in turn makes us more scalable.

3.4 Problem Solution Fit

A Problem-Solution Fit is simply the finding of a problem in your customer's life and the realization of a solution for it that actually solves the problem. Entrepreneurs, marketers, and corporate innovators can identify behavioral patterns and recognize what works and why



4. Requirement Analysis

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Mobile number Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	login	User can login through registered email ID/Mobilenumber

4.2 Non-Functional Requirements

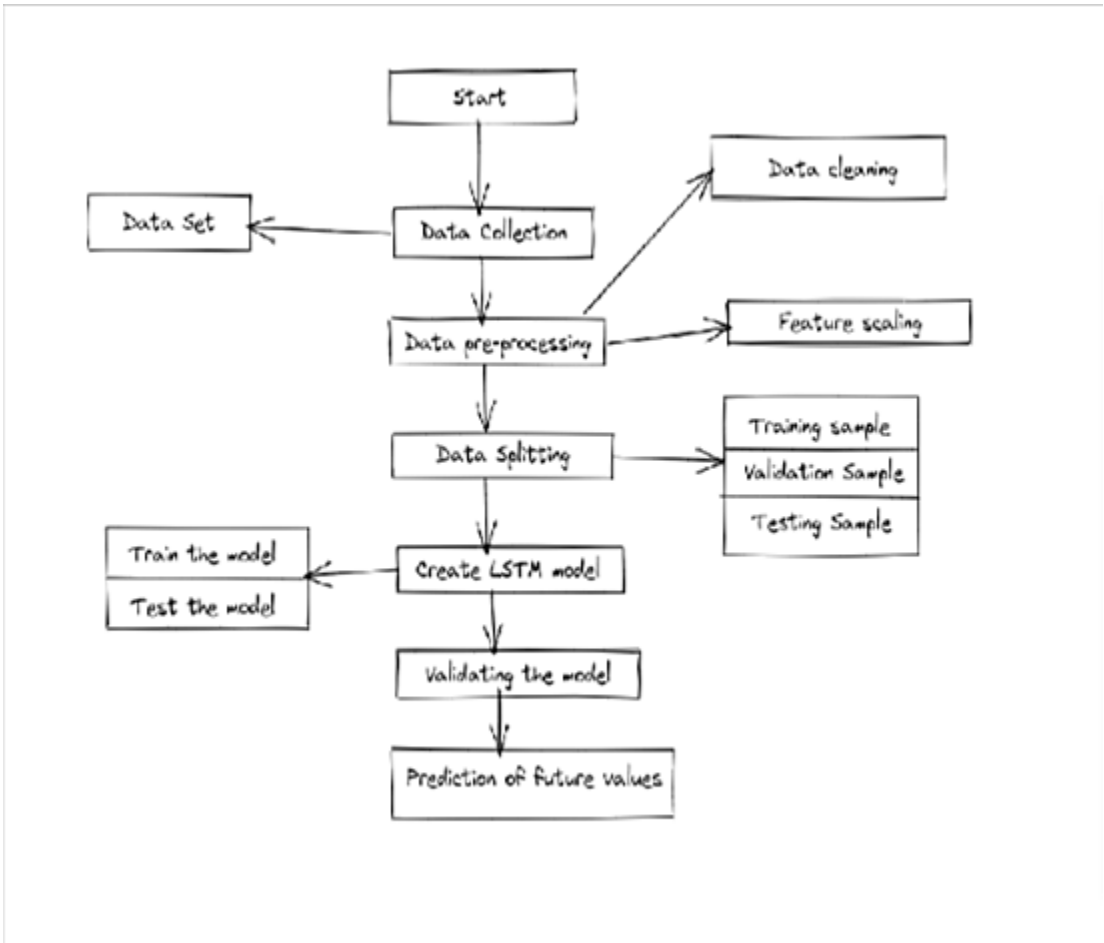
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	UI is user-friendly .we represent the data in charts whichuses clear understanding of price activity
NFR-2	**Security**	We follow certain security protocols likeusing user credentials , OTP verification
NFR-3	**Reliability**	The Data which represented in web app is so accurate and predicting the right data
NFR-4	**Performance**	The performance in this projectis determined through “how accurately you can predict the price ofthe crude oil “
NFR-5	**Availability**	The web appis available to all devices(Android , Mac , windows etc.,)
NFR-6	**Scalability**	According to user basethe project scalability is done

5. Project Design

5.1 Data Flow Diagram

Data Flow Diagrams(DFDs) are traditional visual representations of how information flows within a system. It is possible to illustrate the right amount of the system requirement graphically with a neat and clear DFD. It explains how data enters and leaves the system, what changes the information, and where it is stored.



5.2 Solution & Technical Architecture

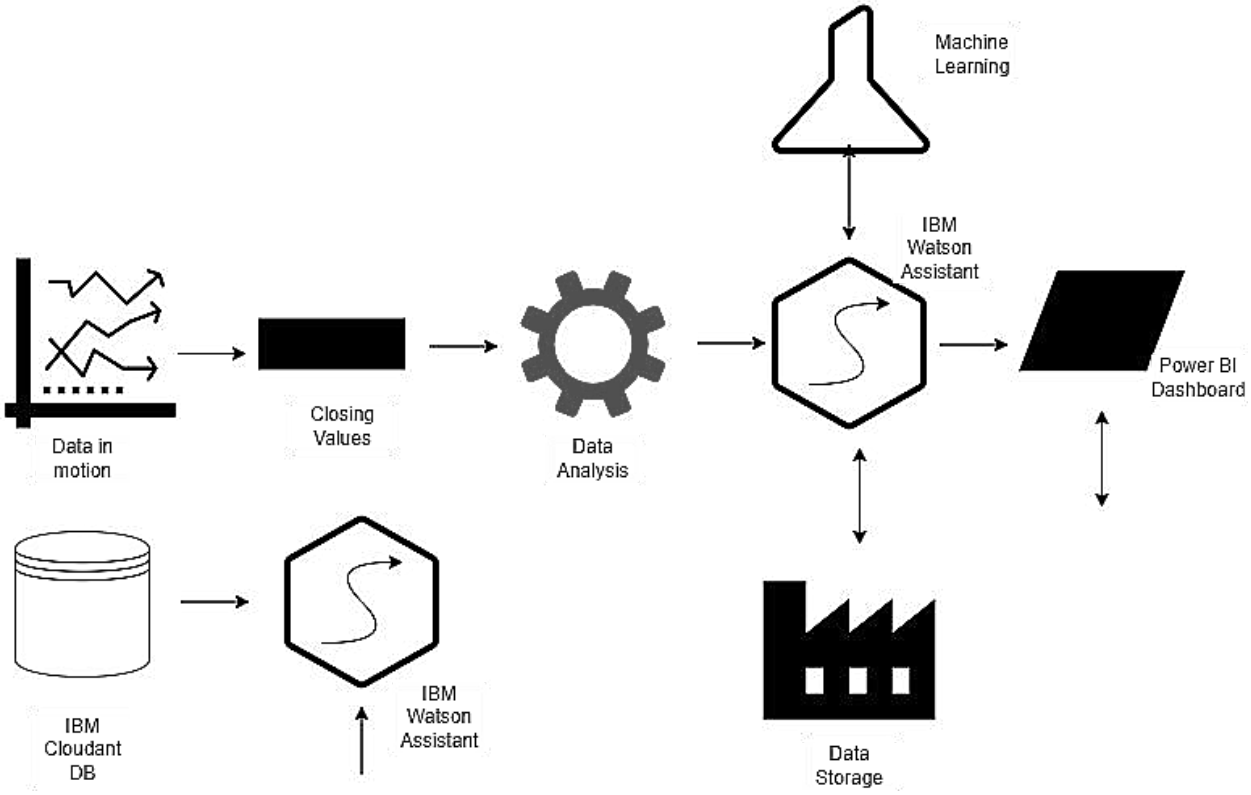
An architectural description of a solution is called a Solution architecture (SA). Enterprise solution architectures (ESAs) combine guidance from various enterprise architecture viewpoints (business, information, and technical). Solution architecture aims to achieve the following overarching goals:

(i) Streamlining day-to-day operations

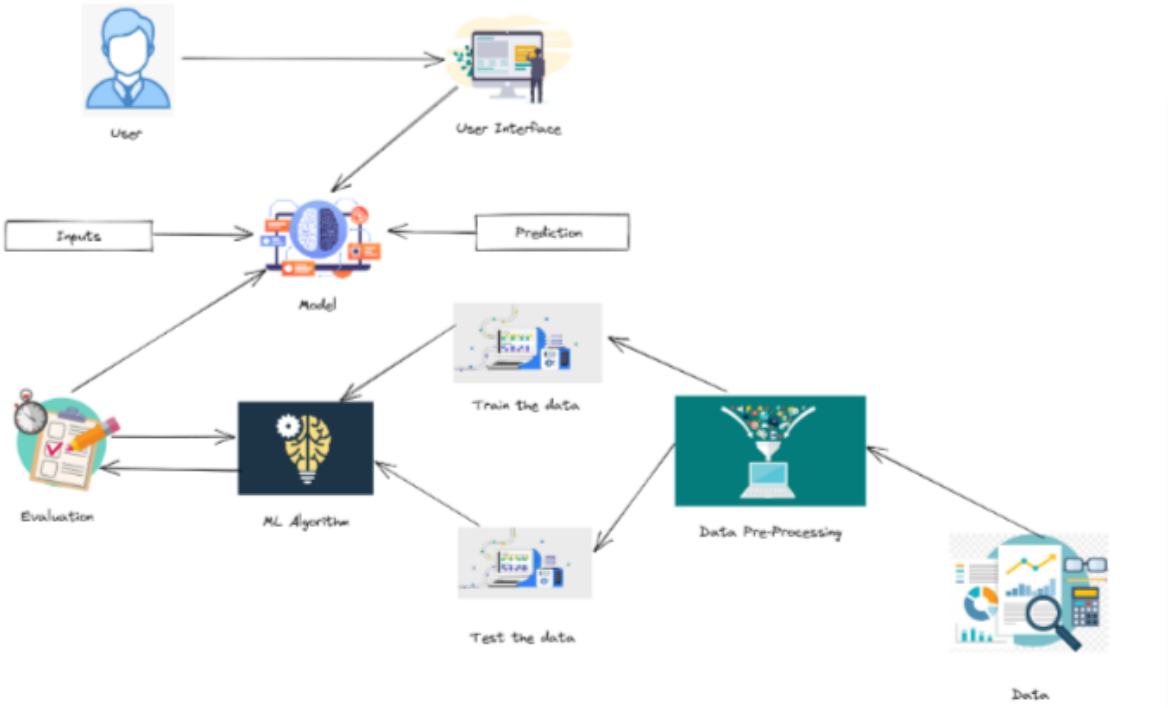
(ii) Providing a more efficientproduction environment

(iii) Lowering costs and gainingcost-effectiveness

(iv) Providing a secure, stable,and supportable environment



Solution Architecture



Technical Architecture

5.3 User Stories

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task
Sprint-1	Data Collection	USN-1	Collecting the Dataset
Sprint-2	DataPre-processing	USN-2	Data Pre-processing
Sprint-3	Model Building	USN-3	Prepare the model by importing thenecessary libraries, adding thelayers, and compiling it.
Sprint-3	Model Building	USN-4	The data classification model is trained using RNNs and other systems.

Sprint-4	Application Building	USN-5	Deploy the modelin the IBMcloudand build the system
Sprint-4	Training and testing	USN-6	Testing the model's performanceand training it

6. Project Planning & Scheduling

6.1 Sprint Planning & Estimation

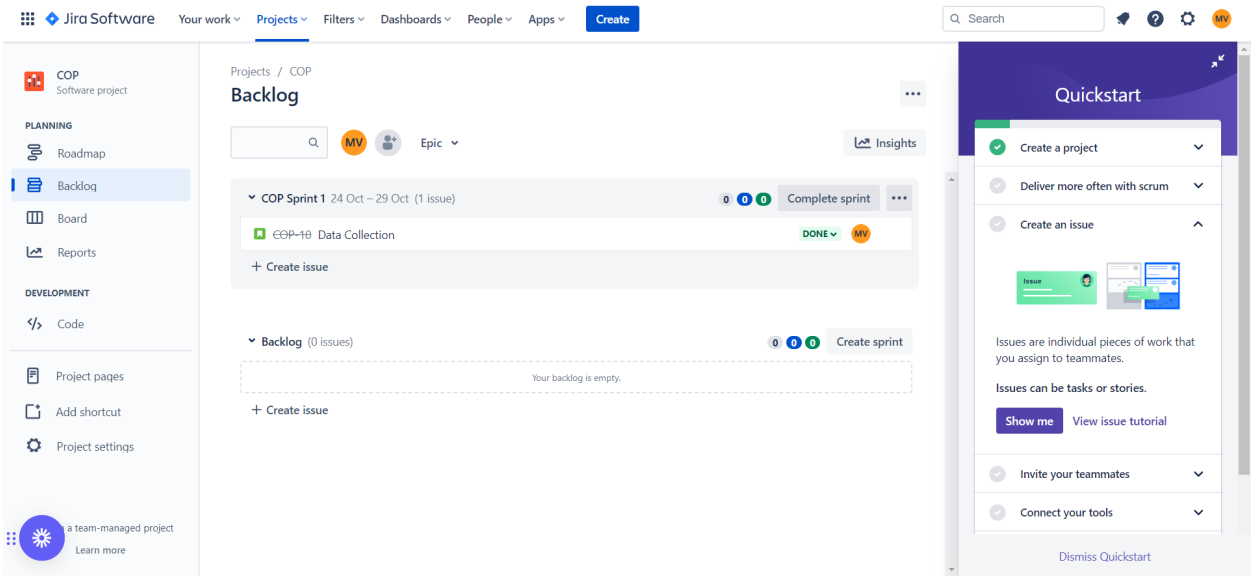
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collecting the Dataset	10	High	Muthu Annamalai KOULURU NANDAKISHORE REDDY MukeshKumar Charan Kumar
Sprint-2	Data Pre-processing	USN-2	Data Pre-processing	7	Medium	Muthu Annamalai KOULURU NANDAKISHORE REDDY MukeshKumar Charan Kumar
Sprint-3	Model Building	USN-3	Prepare the model by importing thenecessary libraries, adding the layers, and compiling it.	10	High	Muthu Annamalai KOULURU NANDAKISHORE REDDY MukeshKumar Charan Kumar

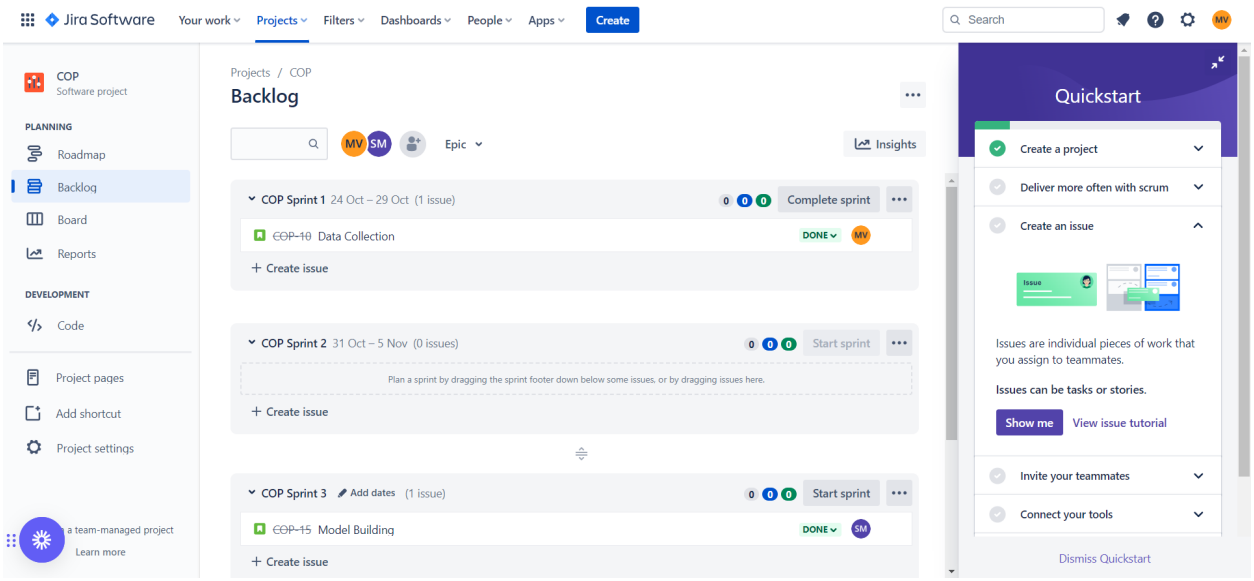
Sprint-3	Model Building	USN-4	The data classification model is trained using RNNs and other systems.	7	Medium	Muthu Annamalai KOULURU NANDAKISHORE REDDY MukeshKumar Charan Kumar
Sprint-4	Application Building	USN-5	Deploy the modelin the IBM cloud and build the system	10	High	Muthu Annamalai KOULURU NANDAKISHORE REDDY MukeshKumar. Charan Kumar
Sprint-4	Training and testing	USN-6	Testing the model's performance and training it	7	Medium	Muthu Annamalai.V KOULURU NANDAKISHORE REDDY MukeshKumar. SCharan Kumar

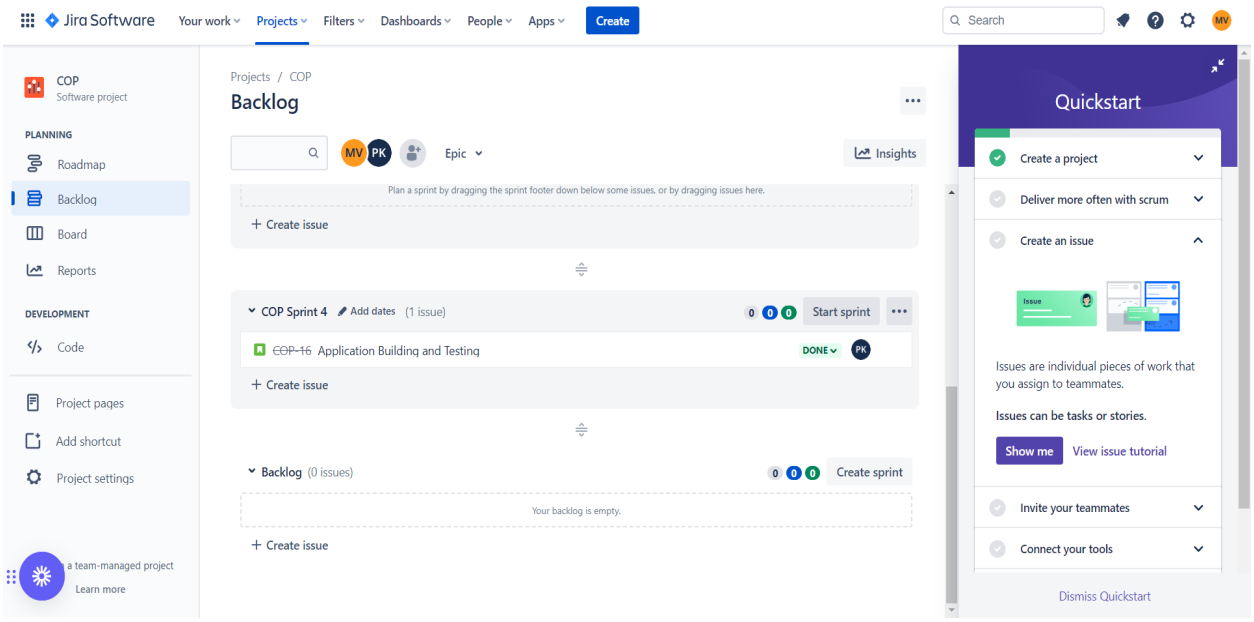
6.2 Sprint Delivery Schedule

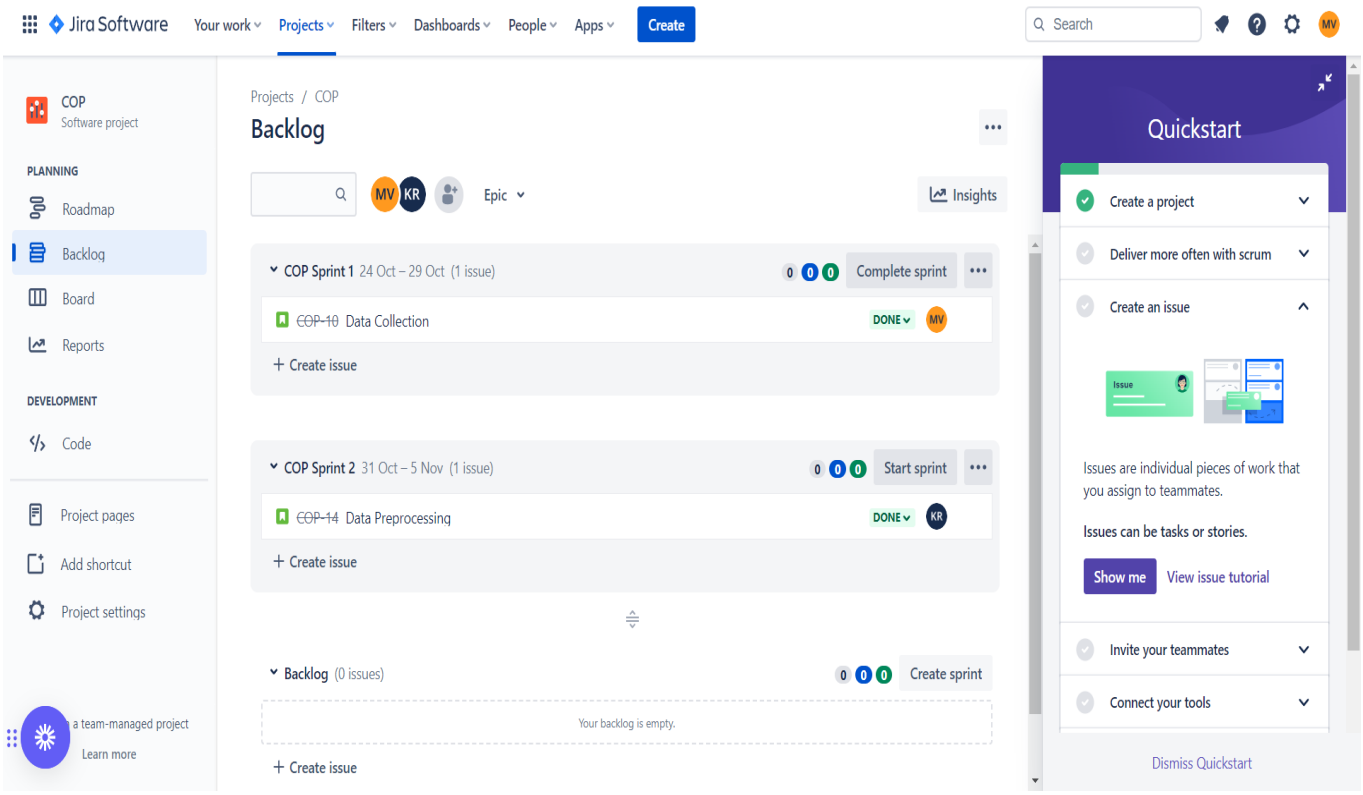
Sprint	Total Story Points	Duration	SprintStart Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	7	19 Nov 2022

6.3 Reports from Jira




Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
COP
Software project
PLANNING
Roadmap
Backlog
Board
Reports
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
a team-managed project
Learn more
Projects / COP
Backlog
Epic
Insights
COP Sprint 1 24 Oct – 29 Oct (1 issue)
Complete sprint
COP-10 Data Collection
DONE
Create issue
COP Sprint 2 31 Oct – 5 Nov (0 issues)
Start sprint
Plan a sprint by dragging the sprint footer down below some issues, or by dragging issues here.
COP Sprint 3 Add dates (1 issue)
COP-15 Model Building
Quickstart
Create a project
Deliver more often with scrum
Create an issue
Issues are individual pieces of work that you assign to teammates.
Issues can be tasks or stories.
Show me
View issue tutorial
Invite your teammates
Connect your tools
Dismiss Quickstart


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
COP
Software project
PLANNING
Roadmap
Backlog
Board
Reports
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
a team-managed project
Learn more
Projects / COP
Backlog
Epic
Insights
Plan a sprint by dragging the sprint footer down below some issues, or by dragging issues here.
Create issue
COP Sprint 4 Add dates (1 issue)
Start sprint
COP-16 Application Building and Testing
DONE
Backlog (0 issues)
Create sprint
Your backlog is empty.
Quickstart
Create a project
Deliver more often with scrum
Create an issue
Issues are individual pieces of work that you assign to teammates.
Issues can be tasks or stories.
Show me
View issue tutorial
Invite your teammates
Connect your tools
Dismiss Quickstart


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
COP
Software project
PLANNING
Roadmap
Backlog
Board
Reports
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
a team-managed project
Learn more
Projects / COP
Backlog
Epic
Insights
COP Sprint 1 24 Oct – 29 Oct (1 issue)
Complete sprint
COP-10 Data Collection
DONE
Create issue
COP Sprint 2 31 Oct – 5 Nov (1 issue)
Start sprint
COP-14 Data Preprocessing
DONE
Create issue
Backlog (0 issues)
Create sprint
Your backlog is empty.
Create issue
Quickstart
Create a project
Deliver more often with scrum
Create an issue
Issue
Issues are individual pieces of work that you assign to teammates.
Issues can be tasks or stories.
Show me
View issue tutorial
Invite your teammates
Connect your tools
Dismiss Quickstart

7. Coding and Solutioning

```python
from flask import Flask, render_template, request, redirect
import numpy as np
from tensorflow import keras
from keras.models import load_model
import joblib
import scipy

app = Flask(__name__)
model = load_model(r'C:\Users\muthu\OneDrive\Desktop\Crude-Oil-Price-Prediction\crude-oil.h5')


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

```python
@app.route('/predict',methods=["POST","GET"])
def predict():
    if request.method == "POST":
        string = request.form['val']
        string = string.split(',')
        x_input = [eval(i) for i in string]
        sc = joblib.load(r'C:\muthu\OneDrive\Desktop\Crude-Oil-Price-Prediction\scaler.save')

        x_input = sc.fit_transform(np.array(x_input).reshape(-1,1))

        x_input = np.array(x_input).reshape(1,-1)

        x_input = x_input.reshape(1,-1)
        x_input = x_input.reshape((1,10,1))
        print(x_input.shape)

        model = load_model(r'C:\Users\muthu\OneDrive\Desktop\Crude-Oil-Price-Prediction\crude-oil.h5')
        output = model.predict(x_input)
        print(output[0][0])

        val = sc.inverse_transform(output)

        return render_template('web.html' , prediction = val[0][0])
    if request.method=="GET":
        return render_template('web.html')

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

8. Testing

8.1 Test cases

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	HomePage	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	HomePage	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	The UI is notdisplayed properly in screen size 2560 x 1801 and 768 x 630	PASS
HP_TC_003	Functional	Web Page	Check if user can enter thepast days	The input price should be updated to the	Working asexpected	PASS

			price	application successfully		
WP_TC_001	Functional	Web Page	Check if user cannot enter anynumber as price	The application should not allow user to enter any number as price	User is able to enterany price	FAIL
WP_TC_002	Functional	Web Page	Check if the page redirects to theresult page oncethe input is given	Thepage should redirect to the results page	Working asexpected	PASS

8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS

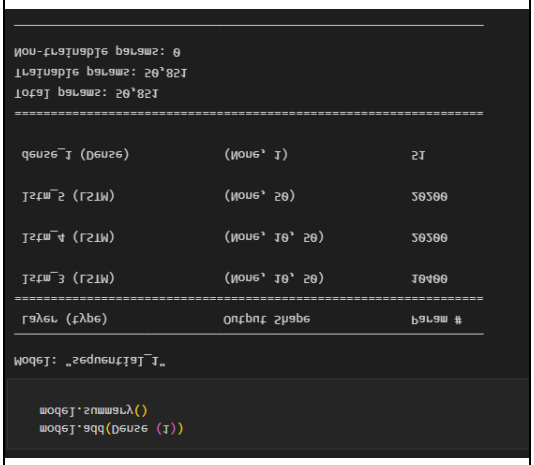
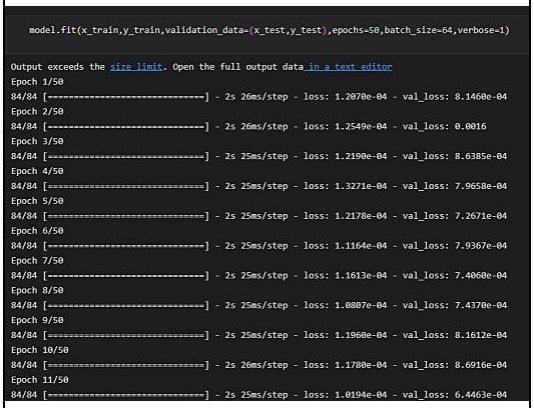
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won’t Fix	1	0	1	0	2
Total	6	1	4	3	14

8.2.2 TESTCASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2
Print Engine	2	0	0	2
Final ReportOutput	2	0	0	2

9. Results

9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary		
2.	Accuracy	Training Accuracy - 1.9685525432167308 Validation Accuracy – 2.201959455277266	
3.	Confidence Score	Class Detected-9 ConfidenceScore9	

10. Advantages and Disadvantages

ADVANTAGES

1. With Price of fallingup full to its lowestlevel in consumer will spendto gasoline in government estimate.
2. Tepid inflationdeclining energy price lamp down inflation.
3. Lowest oil prices economyenergy producing in US, Iran, Venereal.
4. The plungein oil price is roiling market worldwide.
5. Increase fuel mileage for passenger can seemed expensive

DISADVANTAGES

1. Falling oil prices hurt a key sector of stock market.
2. Less business spending automation of energy and equipment firms.
3. Sagging economics higher supply mayor reasonfor oil drop, investor worry persistent declines.
4. Less business spending automation of energy and equipment forms.These facilities high demand of actuatorand values

11. Conclusion

For forecasting prices, LSTM networks are better than traditional neural networks because they use back propagation models. A traditional neural network, such as an RNN, predicts the next outgoing data but does not necessarily save the previous data. RNNs are based on feed-forwarding, so the previous data is not required to predict the future data. The LSTM method stores the previous data and makes predictions based on it, which is rather encouraging and approximate. As a result, relatively encouraging results were derived. It appears that large lookups do not necessarily improve the accuracy of crude oil price predictions. Therefore, it can be concluded that the LSTM model with a single node is the most accurate.

12. Future Scope

This work is carried out on the closing price of crude oil; however, there are various other factorswhich 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 researchis in capturing the changingpattern of theseprices. In the coming future,fundamental indicators and market trendshave been planned to be incorporated into a model which help the proposed model perform more efficiently.

13. Appendix

Source Code

Model Creation

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))
```

Flask App

```python
from flask import Flask, render_template, request, redirect
import numpy as np
from tensorflow import keras
from keras.models import load_model
import joblib
import scipy

app = Flask(__name__)
model = load_model(r'C:\Users\muthu\OneDrive\Desktop\Crude-Oil-Price-Prediction\crude-oil.h5')


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

```python
@app.route('/predict',methods=["POST","GET"])
def predict():
    if request.method == "POST":
        string = request.form['val']
        string = string.split(',')
        x_input = [eval(i) for i in string]
        sc = joblib.load(r'C:\muthu\OneDrive\Desktop\Crude-Oil-Price-Prediction\scaler.save')

        x_input = sc.fit_transform(np.array(x_input).reshape(-1,1))

        x_input = np.array(x_input).reshape(1,-1)

        x_input = x_input.reshape(1,-1)
        x_input = x_input.reshape((1,10,1))
        print(x_input.shape)

        model = load_model(r'C:\Users\muthu\OneDrive\Desktop\Crude-Oil-Price-Prediction\crude-oil.h5')
        output = model.predict(x_input)
        print(output[0][0])

        val = sc.inverse_transform(output)

        return render_template('web.html' , prediction = val[0][0])
    if request.method=="GET":
        return render_template('web.html')

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

RECOGNIZER

```
x_input = [eval(i) for i in string]
sc = joblib.load(r'C:\muthu\OneDrive\Desktop\Crude-Oil-Price-Prediction\scaler.save')

x_input = sc.fit_transform(np.array(x_input).reshape(-1,1))

x_input = np.array(x_input).reshape(1,-1)

x_input = x_input.reshape(1,-1)
x_input = x_input.reshape((1,10,1))
print(x_input.shape)

model = load_model(r'C:\Users\muthu\OneDrive\Desktop\Crude-Oil-Price-Prediction\crude-oil.h5')
output = model.predict(x_input)
print(output[0][0])

val = sc.inverse_transform(output)
```

Home Page - HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Homepage</title>
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css2?family=Ubuntu&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="{{url_for('static',filename='css/style.css')}}">
</head>
<body>
    <header>
        <div>
                <ul class="menu">
                    <li><a href="{{url_for('home')}}">Home</a></li>
                    <li><a href="{{url_for('home')}}">About</a></li>
                </ul>
            </div>
        <div class="banner-text">
            <h1>CRUDE OILL<span> PRICE PREDICTION</span></h1>
            <p>Crude oil is composed of hydrocarbons, which are mainly hydrogen (about 13% by weight) and carbon
            <a href="{{url_for('predict')}}">PREDICT</a>
        </div>
    </header>

</body>
</html>
```

Home Page - CSS

```css
* {
  margin: 0;
  padding: 0;
}
body {
  font-family: "Poppins", sans-serif;
}
.wrapper {
  width: 1170px;
  margin: auto;
}
header {
  background: linear-gradient(rgba(0, 0, 0, 0.8), rgba(0, 0, 0, 0.8)),
    url(1.jpg);
  height: 100vh;
  -webkit-background-size: cover;
  background-size: cover;
  background-position: center center;
  position: relative;
}
.menu {
  float: right;
  list-style: none;
  margin-top: 30px;
}
.menu li {
  display: inline-block;
}
.menu li a {
  color: #fff;
  text-decoration: none;
  padding: 5px 20px;
  font-family: "Poppins", sans-serif;
  font-size: 16px;
  text-transform: uppercase;
}
```

Web Page - HTML

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Web Page</title>
    <link
      rel="stylesheet"
      href="{{url_for('static',filename='css/web.css')}}"
    />
    <style>
      input[type="text"] {
        width: 100%;
        border: 2px solid #aaa;
        border-radius: 4px;
        margin: 8px 0;
        outline: none;
        padding: 8px;
        box-sizing: border-box;
        transition: 3s;
      }
      input[type="text"]:focus {
        border-color: purple;
        box-shadow: 0 0 8px 0 purple;
      }
    </style>
```

Web Page - CSS

```css
* {
  margin: 0;
  padding: 0;
  text-decoration: none;
  list-style: none;
}
body {
  font-family: "poppins", sans-serif;
}
.homepage {
  min-height: 100vh;
  width: 100%;
  position: relative;
  background: linear-gradient(rgba(0, 0, 0, 0.8), rgba(0, 0, 0, 0.8)),
    url(1.jpg);
  background-size: cover;
  background-position: center;
  z-index: 1;
}
.homepage::after {
  content: "";
  position: absolute;
  left: 0;
  top: 0;
  height: 100%;
  width: 100%;
  background-color: rgb(34, 33, 33);
  opacity: 0.3;
  z-index: -2;
}
.navbar {
  position: absolute;
  padding: 40px 40px;
  display: flex;
  justify-content: space-between;
  width: 100%;
  box-sizing: border-box;
```

Github Project Link: https://github.com/IBM-EPBL/IBM-Project-6934-1658843459

Project Demo Link:https://drive.google.com/drive/folders/1Qlq2mmC1XJ_7slnsAuRoW5yLOlM6sR0N?usp=sharing