CRUDE OIL PRICE PREDICTION

NALAIYA THIRAN PROJECT BASED LEARNING

on

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

Project Report Submitted by

911519104012 – HOORUL AMEEN J

911519104015 – JANARTHANAN S

911519104023 – MOHAMED SUHAIL P N

911519104032 - SANJAY K

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

MOHAMED SATHAK ENGINEERING COLLEGE

KILAKARAI-623 806.

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER – 2022

INDEX

1. INTRODUCTION

1.1ProjectOverview

Oil is the largest traded commodity and its demand is highly inelastic. Both the oil importers and exporters have a major concern towards the increasing or decreasing price. The rise in price is good news for producers because they will see an increase in their revenue. Oil importers, how ever,willexperience increased costs of purchasing oil. A rising oil price can even shift economic or politicalpower from oil importers to oil exporters. The crude oil price movements are subject to diverse influencing factors. There fore, prediction of crude oil prices are highly important.

The proposed work mainly focuses on applying Neural Networks to predict the Crude Oil Price.This decision helps importers to buy crude oil at the proper time. Time series analysis is the best option for this kind of prediction as it utilizes the Previous history of crude oil prices to predict future crude oil.

Purpose

Crude oil prices are controlled by many factors which include the difficulty for industrialists to make profit out of volatility, downfall in the market because of sudden fall in prices, and so on. This problem affects the industrialists, crude oil investors, supply and demand. It is important to fix the problem as it can stabilize the market. The boundaries of the problem are to increase the profit and to predict crude oil investments. There fore, Prediction of crude oil can help in dealing with the sudden rise and fall of prices.

1. LITERATURESURVEY

2.1 Existing problem

A new approach for crude oil price prediction based on stream learning

Shuang Gaoet al., proposed a novel approach based on a new machine learning paradigm called steam learning to predict the price of crude oil. The main advantage of stream learning approach is that the prediction model can capture the changing pattern of oil prices since the model is continuously updated whenever new oil price data are available, with very small constant overhead. The stream learning approach has been compared with other models in order to testthe accuracy of results and it is observed that this model yields a higher performance result interms of mean squared error and directional ratio over a variety of horizons. Two types of prices are predicted which includes the U.S. refiner acquisition cost for crude oil imports and the WTI crude oil spot

2.2References

A. Lu, Q., Sun, S., Duan, H., &Wing,S. (2021). Analysis and forecasting of crude oil price basedon the variable selection – LSTM integrated model.Energy informatics, 4(2), 1-20

B. Ghosh, S., &Sivakumar, N. (2015). Beta lustering of Impact of Crude oil price on the Indianeconomy. J.Appl. ManagementInvestments, 4(1), 24-34

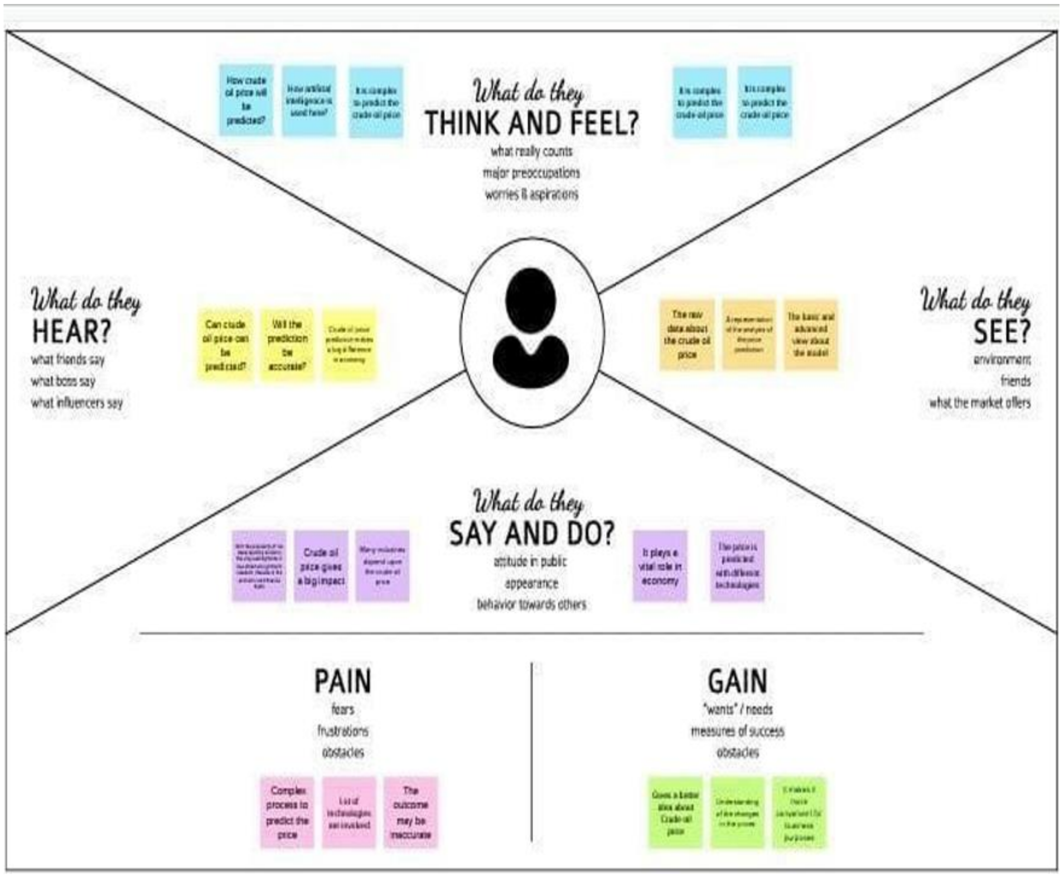
2.3 Problem StatementDefinition

Crude oil price prediction is a Challenging task in oil producing Countries. Its price is among the most complex and tough to model because fluctuations of priceof crude oil are highly irregular, non linear and vary dynamically with high uncertainly The problem statement is proposed to a hybrid model for crude oil price prediction that uses the RNN (Recurrent neural network) with long short –term memory (LSTM)to predict the crude oil price–

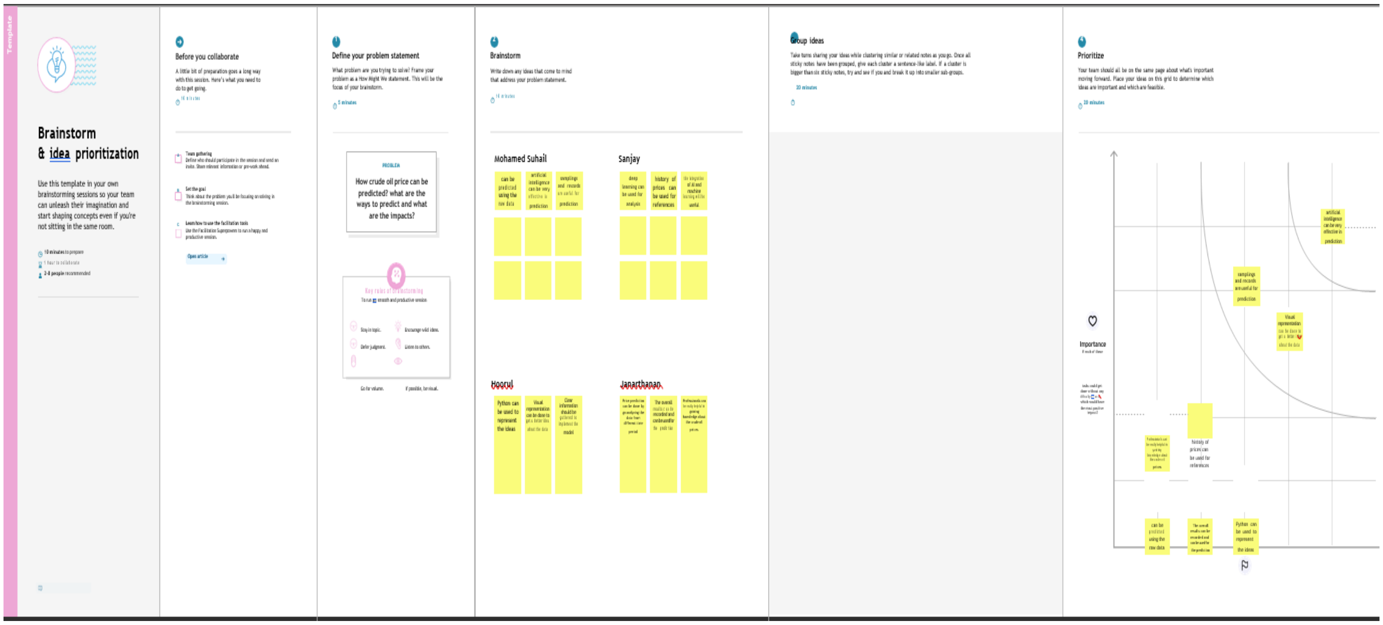
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map canvas

Empathy map canvas :


How crude oil price will be predicted?
How artificial intelligence is used here?
It is complex to predict the crude oil price
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
It is complex to predict the crude oil price
It is complex to predict the crude oil price
What do they
HEAR?
what friends say
what boss say
what influencers say
Can crude oil price can be predicted?
Will the prediction be accurate?
The raw data about the crude oil price
The basic and advanced view about the model
What do they
SEE?
environment
friends
what the market offers
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Crude oil price gives a big impact
Many industries depend upon the crude oil price
It plays a vital role in economy
The price is predicted with different technologies
PAIN
fears
frustrations
obstacles
Complex process to predict the price
Lot of technologies are involved
The outcome may be inaccurate
GAIN
"wants" / needs
measures of success
obstacles
Gives a better idea about Crude oil price
Understanding of the changes in the prices

3.2 Ideation & Brainstorming



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

How crude oil price can be predicted? what are the ways to predict and what are the impacts?

Key rules of brainstorming

To run a smooth and productive session

Stay in topic. Encourage wild ideas.
Defer judgment. Listen to others.
Go for volume. If possible, be visual.

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Mohamed Suhail
can be predicted using the raw data
artificial intelligence can be very effective in prediction
samplings and records are useful for prediction
Sanjay
deep learning can be used for analysis
history of prices can be used for references
the integration of AI and machine learning will be useful
Hoorul
Python can be used to represent the ideas
Visual representation can be done to get a better idea about the data
Clear information should be gathered to implement the model
Janarthanan
Price prediction can be done by grouping the data from different time period
The overall results can be recorded and can be used for the prediction
Professionals can be really helpful in gaining knowledge about the crude oil prices
Importance
Feasibility

3.3 Proposed Solution Template:

The project team shall fill in the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	As with the erratic changes in supply and demand and also the influence of geopolitics, it is very hard to predict the value of crude oil prices in the global market.
2.	Idea / Solution description	We are going to collect the dataset of the past oil prices with time so that by feeding those to the model and training it and compiling it and whenit's achieved the optimal state we can implement it in the web application.
3.	Novelty / Uniqueness	It may be a traditional idea but the implementation of periodic training will have abettereffect on it.

4.	Social Impact / Customer Satisfaction	By using the web app customer can gain knowledge of the crude oil price and get benefits financially.
5.	Business Model (Revenue Model)	It will be used by every individual at ease so that they can have an idea of the crude price so, that the use of the crude will be stable in the market
6.	Scalability of the Solution	The idea we proposed it take the input in the periodic and adjust and train through these so, that it will adapt to very different situations.

Barone-adesi et al. (1998) to short-term forecast of Brent crude oil price. In another work, Tang and Hammoudeh (2002) utilized a nonlinear regression to predict OPEC basket price. Using OECD petroleum inventory levels and relative stock inventories, Ye et al. (2002, 2005) adopted a simple linear regression model for short-term monthly prediction of WTI crude oil spot price. In a related study, Ye et al. (2006) included nonlinear variables such as low- and high- inventory variables to the linear forecasting model suggested by Ye et al. (2002, 2005) to predict short-runWTI crude oil prices.
Zamani (2004) used an econometrics forecasting model to anticipate the short-term quarterly WTI crude oil spot price using OECD stocks, non-OECD demand,and OPEC supply. Using error correction models, Lanza et al. (2005) looked at the pricing of products and crude oil. Sadorsky (2006) used GARCH, TGARCH, AR, and BIGARCH statistical models, among others, to forecast daily volatility in petroleum futures price returns. To predict oil demand, supply, and prices, Dees et al. (2007) created a linear model of the global oil market with a primary focus on OPEC behavior. Murat and Tokat (2009) looked into the connection between futures and spot crude oil prices and used the random walk model to testif futures prices might predict changes in spot prices.

However, more recent research have used GARCH and several models from the GARCH family to forecast oil prices. For instance, the GARCH model was employed by Narayan and Narayan (2007) and Agnolucci(2009) to forecast spotand futures crude oil prices. Ina related study, Mohammadi and Su (2010) investigated the crude oil price predicting outcomes of various GARCH-type models. CGARCH, FIGARCH, and IGARCH models were suggested by Kang etal. (2009) to predict the volatility of crude oil markets.

Wei et al. (2010) enhanced the work of Kang et al. (2009) towards the same goal by using linear and nonlinear GARCH-class models. As a result of the application of linear techniques, a sizable difference between the projected and real price of oilhas been demonstrated. The most often utilised exogenous variables in these models for predicting oil prices are inventories, supply, and demand. The fact that supply and demand are relatively inelastic to price changes and that inventory adjustments can take time to materialism account for a considerable share of the difference between actual and predicted prices, especially in the near run (Hamilton, 2008). However, traditional statistical and economic techniques frequently only detect linear processes in data. data time series. (Weigend and Gershenfeld, 1994). However, the oil prices behavior is characterized by a high non linearity and irregularity. Therefore, the mentioned models are not the appropriate choice to forecast the oil price.

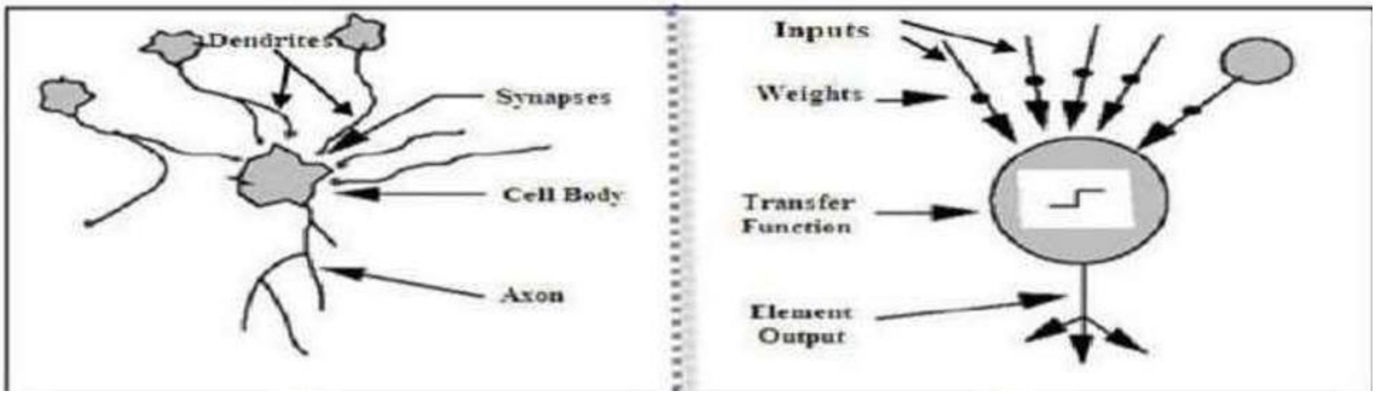
Artificial Neural Network (ANN):

Definition and Neuron

ModelEvolution

Definition

ANN is an input-output mathematical model that mimics how the humanbrain functions by adopting the same strategy for learning new things. An equivalence between biological and an artificial neuron is shown in Fig. 1.



model Figure 1.Analogy between biological neuron (a) and artificial neuron (b)

Neuron Model Evolution

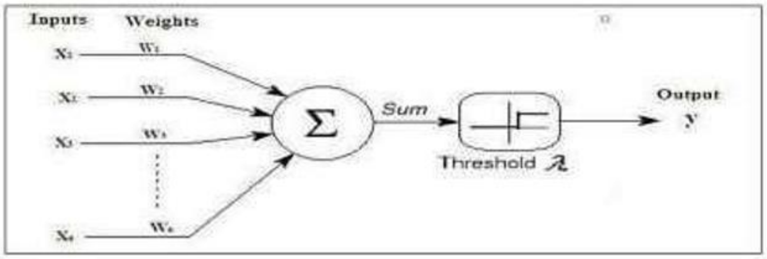
a. McCulloch & Pitts (1943) neuron model

McCulloch & Pitts (1943) neuron model McCullochand Pitts (1943) proposed the first artificial neuron also called formal neuron. Mathematically, theMcCulloch written as follows:

Where 1 2 x , x , ..., n x represent the McCulloch- Pitts neuron inputs that are exclusively binary values (zeros or ones), 1 2 , ,..., w w wn are the connections‟ weights received by the neuron.f is the sign function,
is the threshold and y isthe output of McCulloch - Pitts neuron defined as:

$$f((x_1,\dots,x_n),(w_1,\dots,w_n)) = \begin{cases} 1, & if \sum_{i=1}^{n} w_i x_i \geq \lambda \qquad (2) \\ 0, & if \sum_{i=1}^{n} w_i x_i < \lambda \qquad (3) \end{cases}$$

Figure 2. Illustration of McCulloch & Pitts (1943) neuron



a. Multi layer perception model

Without hidden layers, perception neural networks assume just binary input-output valuesand only two layers, which explains why the model can only handle linearly separable functions. The delta rule was developed by Windrow and Hoff in 1960 and consists of changing the weights of the connections to minimize the discrepancy between the desired and actual output value. As a result, in place of 0 and 1, the output value can take any value. In their book, Minsky and Papert (1969),emphasised the value of including one or more hidden layers to identify the intricate features contained in the inputs. Traditionally, the multi layer perception net was trained using Rumelhart et albackpropagation .'s learning technique (explained in more depth in the following section)(1986).The multi layer perceptionis composed of a layer of input units, one or more hidden layers and an output layer (see Fig .4).

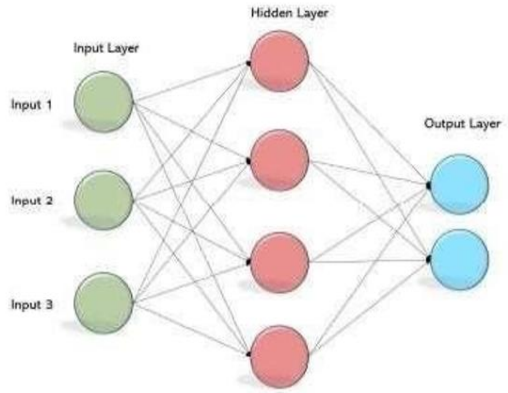


Figure:4

In this network system, the information propagates in a single direction „„forward"": the input units pass the information to the neurons in the first hidden layer, the outputs from the first hidden layer are subsequently

passed to the next layer,and so forth. Thus, the network output(for example, with one hidden layer)is:

Where: i x are the input variables of the network;I is the number of input variables;J is the total number of nodes in the hidden layer; K is the number of neurons in theoutput layer; g and h are, respectively, the transfer/activation function of the first and the second layer; w1 is the weights matrix of the hidden layer; w2 is the weights matrix of the output layer; 1 b and 2 b are the bias vectors of the hidden layer and of the output layer, respectively. To note, at least one transfer function(see the next section for more description of transfer function)of the hidden layer must be nonlinear(Hornik et al., 1989).

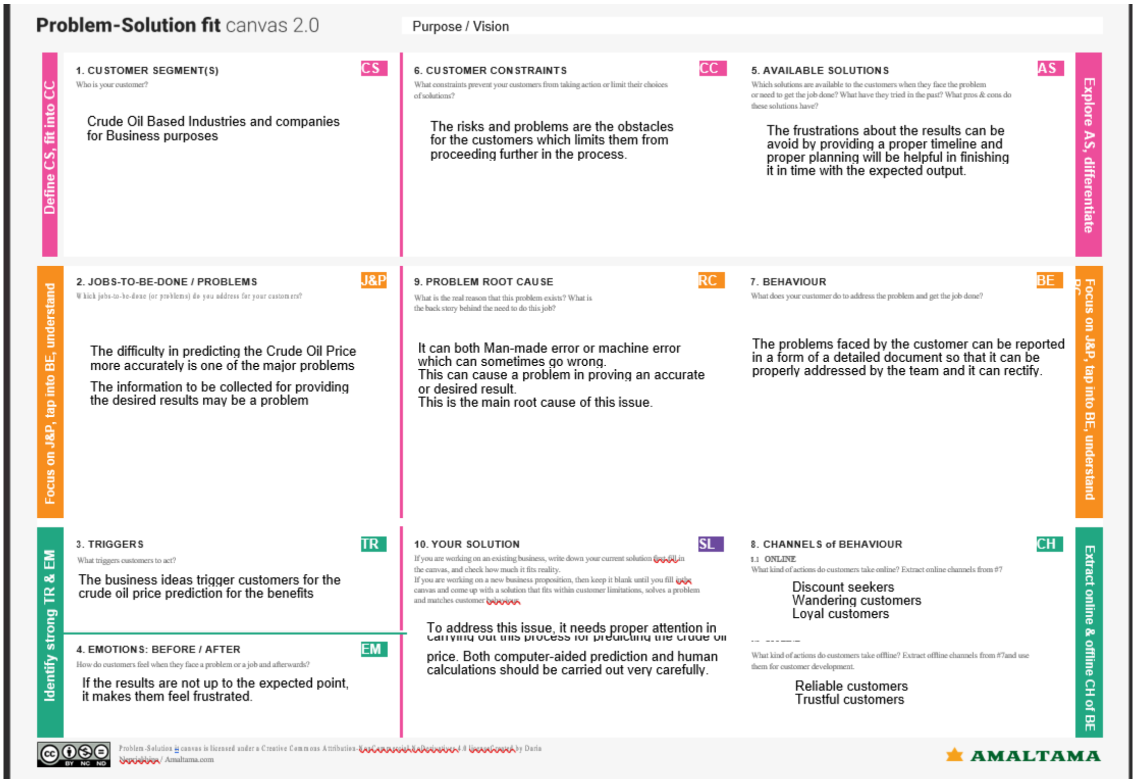
2.2 References:

1. Yu Runfang, Du Jiangze and Liu Xiaotao,"Improved Forecast Ability ofOil Market Volatility Based on combinedMarkov

Switching and GARCH- class Model,Procedia ComputerScience, vol. 122, pp. 415-422,2017.

2. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber,"LSTM: A Search Space Odyssey," IEEE Transactions onNeural Networks and Learning Systems,vol. 28, no. 10, pp. 2222-2232,Oct. 2017.
3. Mohammad Reza Mahdiani and Ehsan Khamehchi, “A modified neural network model for predicting the crude oil price”, Intellectual Economics, vol.10, no. 2, pp. 71-77, Aug. 2016.
4. Manel Hamdi and Chaker Aloui, "Forecasting Crude Oil Price Using Artificial Neural Networks: A Literature Survey," Economics Bulletin, AccessEcon, vol. 35, no. 2, pp. 1339-1359, 2015.
5. Aloui,Chaker & Hamdi, Manel. (2015). Forecasting Crude Oil PriceUsing Artificial Neural Networks: A Literature Survey. Economics Bulletin.35. 1339-1359.

3.4 ProblemSolution fit



Define CS, fit into CC
Focus on J&P, tap into BE, understand
Identify strong TR & EM
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Extract online & offline CH of BE
CS
CC
AS
J&P
RC
BE
TR
SL
CH
EM
AMALTAMA

Problem-Solution fit canvas 2.0

Purpose / Vision

1. CUSTOMER SEGMENT(S)

Who is your customer?

Crude Oil Based Industries and companies for Business purposes

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions?

The risks and problems are the obstacles for the customers which limits them from proceeding further in the process.

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have?

The frustrations about the results can be avoid by providing a proper timeline and proper planning will be helpful in finishing it in time with the expected output.

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers?

The difficulty in predicting the Crude Oil Price more accurately is one of the major problems

The information to be collected for providing the desired results may be a problem

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists? What is the back story behind the need to do this job?

It can both Man-made error or machine error which can sometimes go wrong.
This can cause a problem in proving an accurate or desired result.
This is the main root cause of this issue.

7. BEHAVIOUR

What does your customer do to address the problem and get the job done?

The problems faced by the customer can be reported in a form of a detailed document so that it can be properly addressed by the team and it can rectify.

3. TRIGGERS

What triggers customers to act?

The business ideas trigger customers for the crude oil price prediction for the benefits

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards?

If the results are not up to the expected point, it makes them feel frustrated.

10. YOUR SOLUTION

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.

To address this issue, it needs proper attention in carrying out this process for predicting the crude oil price. Both computer-aided prediction and human calculations should be carried out very carefully.

8. CHANNELS of BEHAVIOUR

8.1 ONLINE

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

Discount seekers
Wandering customers
Loyal customers

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

Reliable customers
Trustful customers

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 Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation viaOTP
FR-3	Graph	Showing graph by obtaining the data from the dataset
FR-4	Support	Providing answers for the queries asked by users.
FR-5	News	Information of the oil prices will be updated by admin
FR-6	Notification	Notification will be sent for the users price alert

Fr-7	Database	Information of theUser will be stored

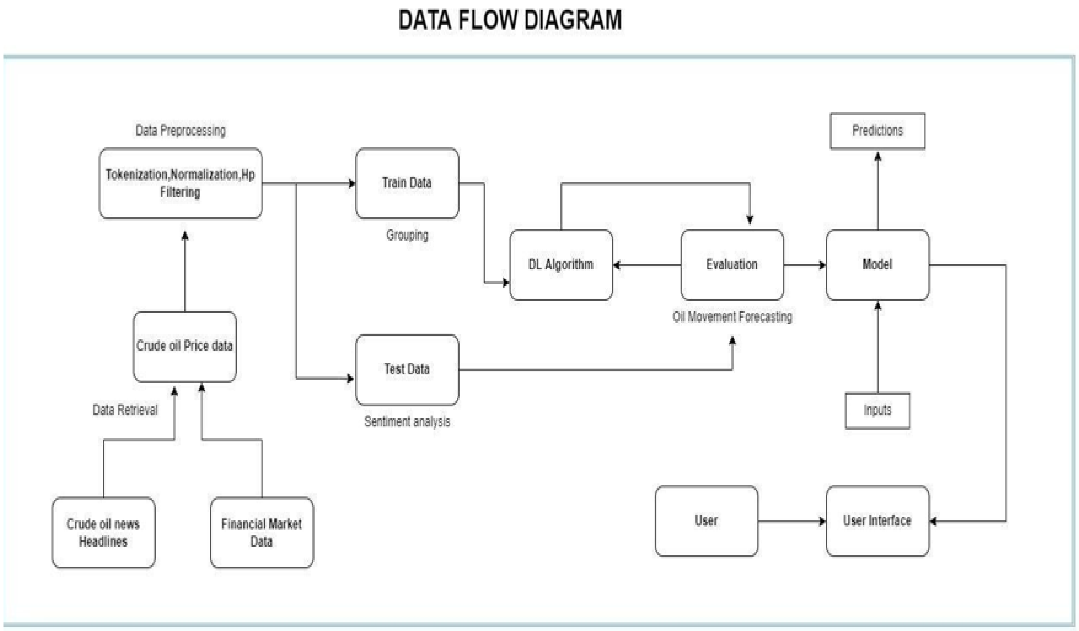
4.2 Non-functionalRequirements:

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

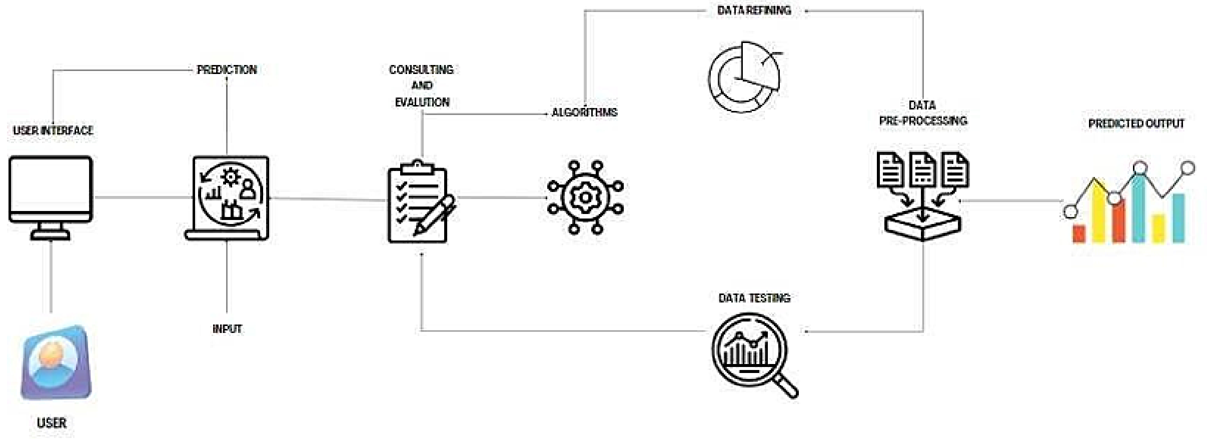
FR No.	Non-Functional Requirement	Description
NFR -1	**Usability**	It can use by wide variety of client a sit is very simple to learn and not complex to proceed.
NFR -2	**Security**	We are using login for the user and the information will be hashed so that it will be very secure to use.
NFR -3	**Reliability**	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 perform fast and secure even at the lower bandwidth.
NFR -5	**Availability**	Prediction will be available for every user but only for premium user news,database and price alert will be alert.
NFR -6	**Scalability**	It is scalable that we are going to use data in kb so that the quite amount of storage is satisfied.

5. PROJECT DESIGN

5.1 Data Flow Diagram

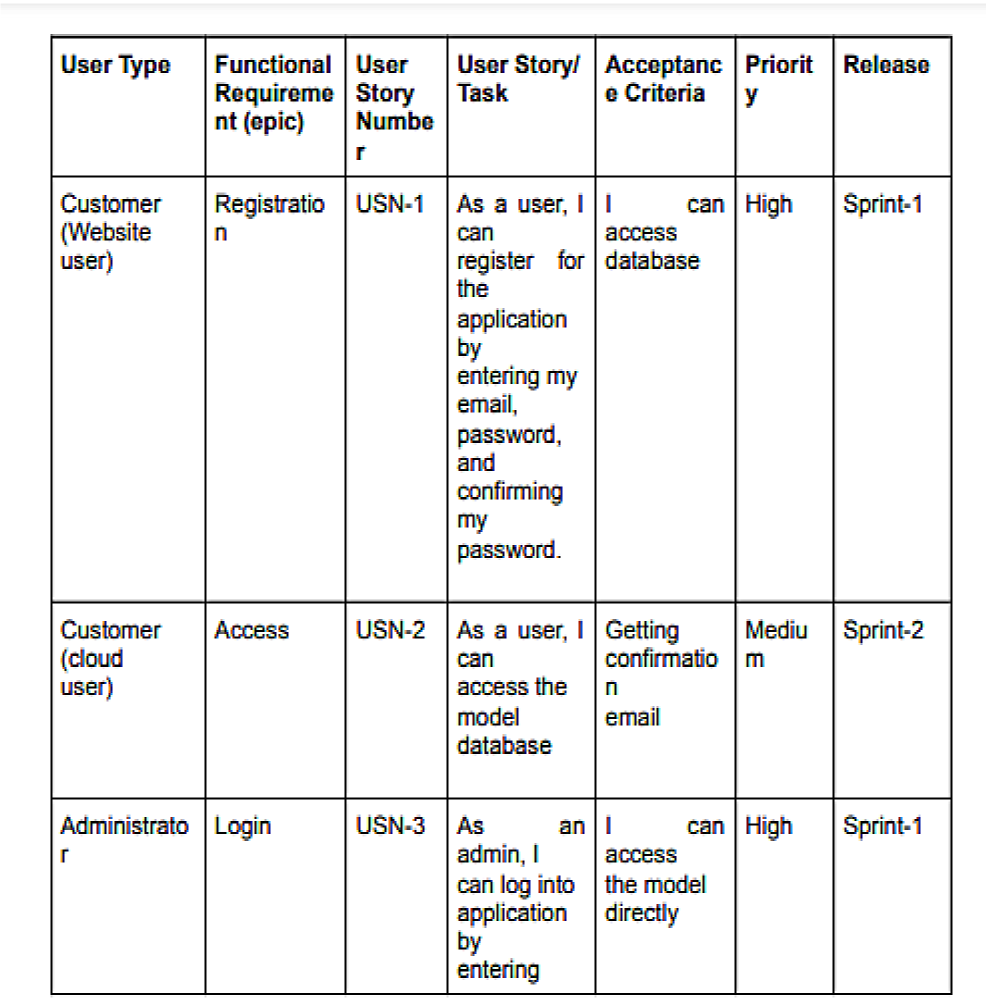


5.2 Solution & Technical Architecture

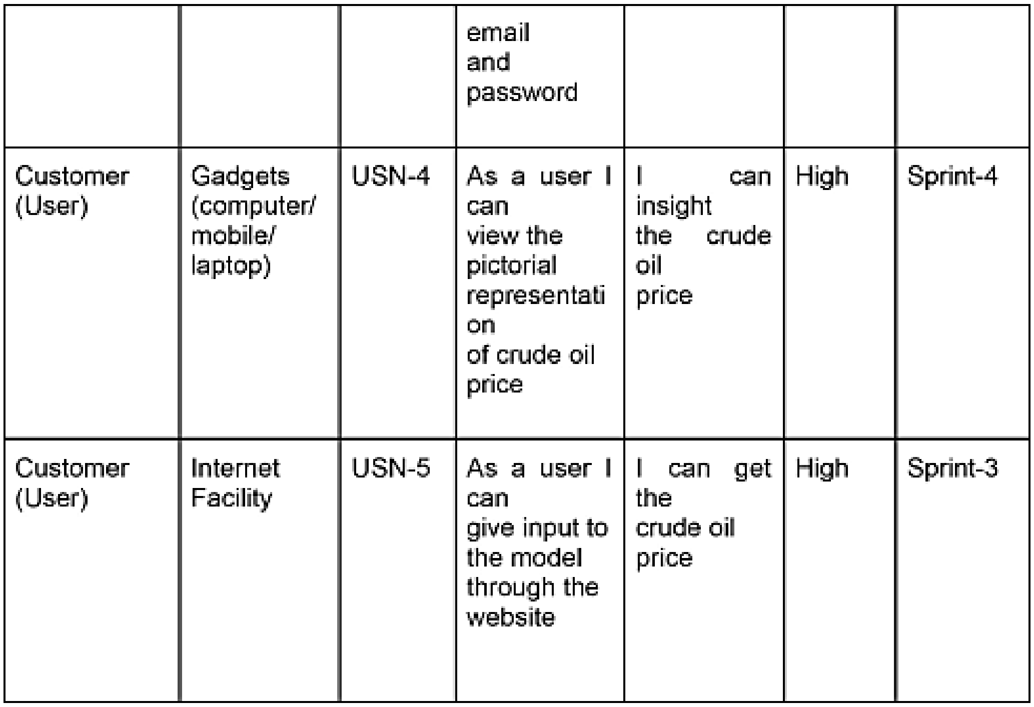


5.3 User Stories

Use the below template to list all the user stories for the product.



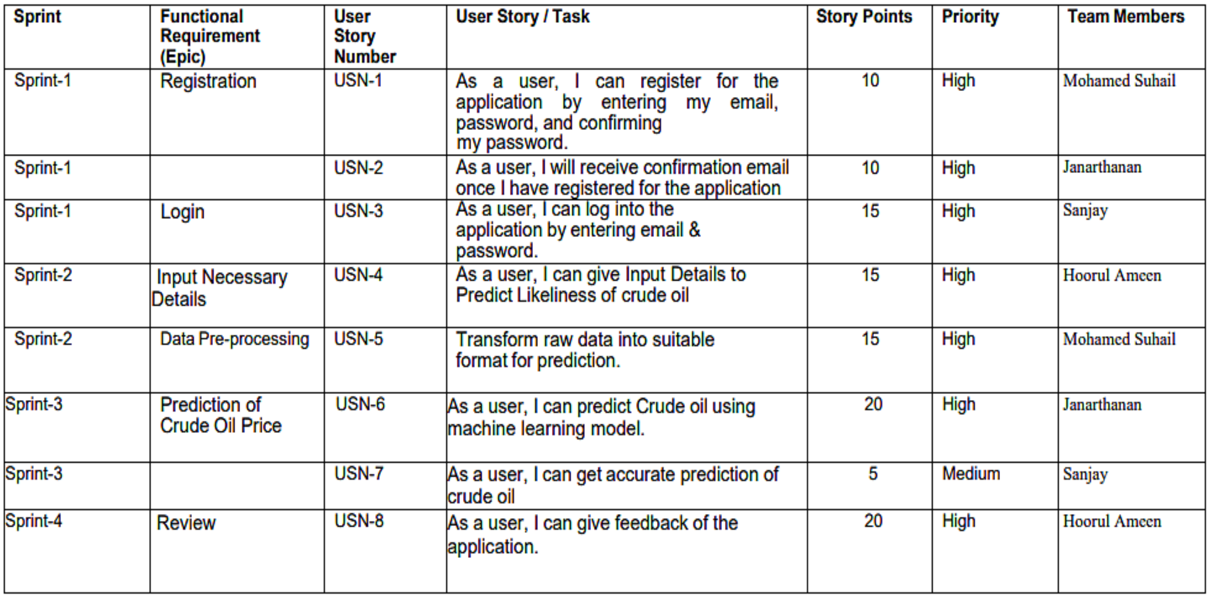
User Type	Functional Requirement (epic)	User Story Number	User Story/ Task	Acceptance Criteria	Priority	Release
Customer (Website user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access database	High	Sprint-1
Customer (cloud user)	Access	USN-2	As a user, I can access the model database	Getting confirmation email	Medium	Sprint-2
Administrator	Login	USN-3	As an admin, I can log into application by entering	I can access the model directly	High	Sprint-1



			email and password			
Customer (User)	Gadgets (computer/ mobile/ laptop)	USN-4	As a user I can view the pictorial representation of crude oil price	I can insight the crude oil price	High	Sprint-4
Customer (User)	Internet Facility	USN-5	As a user I can give input to the model through the website	I can get the crude oil price	High	Sprint-3

6 PROJECT PLANNING & SCHEDULING

6.1Sprint Planning & Estimation



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	10	High	Mohamed Suhail
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	10	High	Janarthanan
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password.	15	High	Sanjay
Sprint-2	Input Necessary Details	USN-4	As a user, I can give Input Details to Predict Likeliness of crude oil	15	High	Hoorul Ameen
Sprint-2	Data Pre-processing	USN-5	Transform raw data into suitable format for prediction.	15	High	Mohamed Suhail
Sprint-3	Prediction of Crude Oil Price	USN-6	As a user, I can predict Crude oil using machine learning model.	20	High	Janarthanan
Sprint-3		USN-7	As a user, I can get accurate prediction of crude oil	5	Medium	Sanjay
Sprint-4	Review	USN-8	As a user, I can give feedback of the application.	20	High	Hoorul Ameen

6 PROJECT PLANNING & SCHEDULING

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

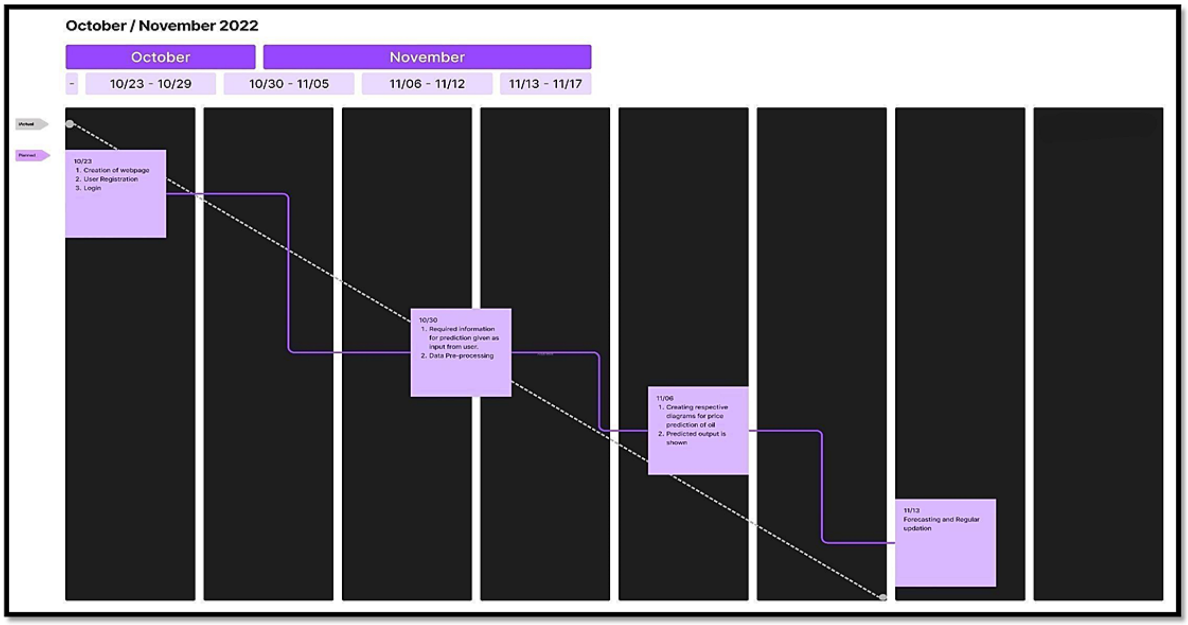
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

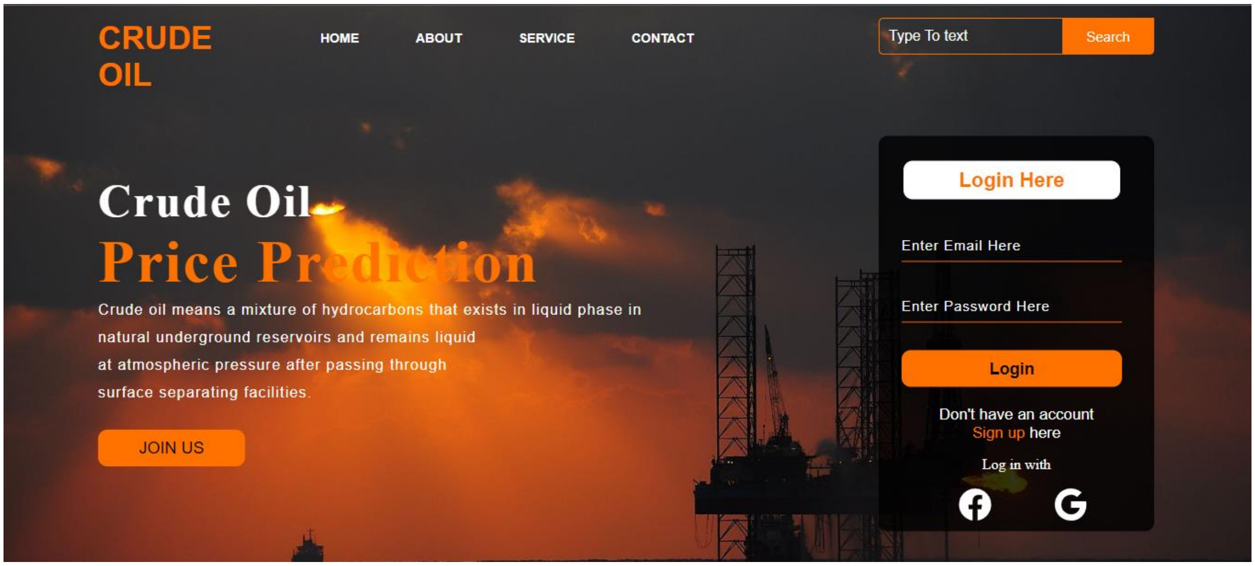
Velocity:

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

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

Burndown Chart:





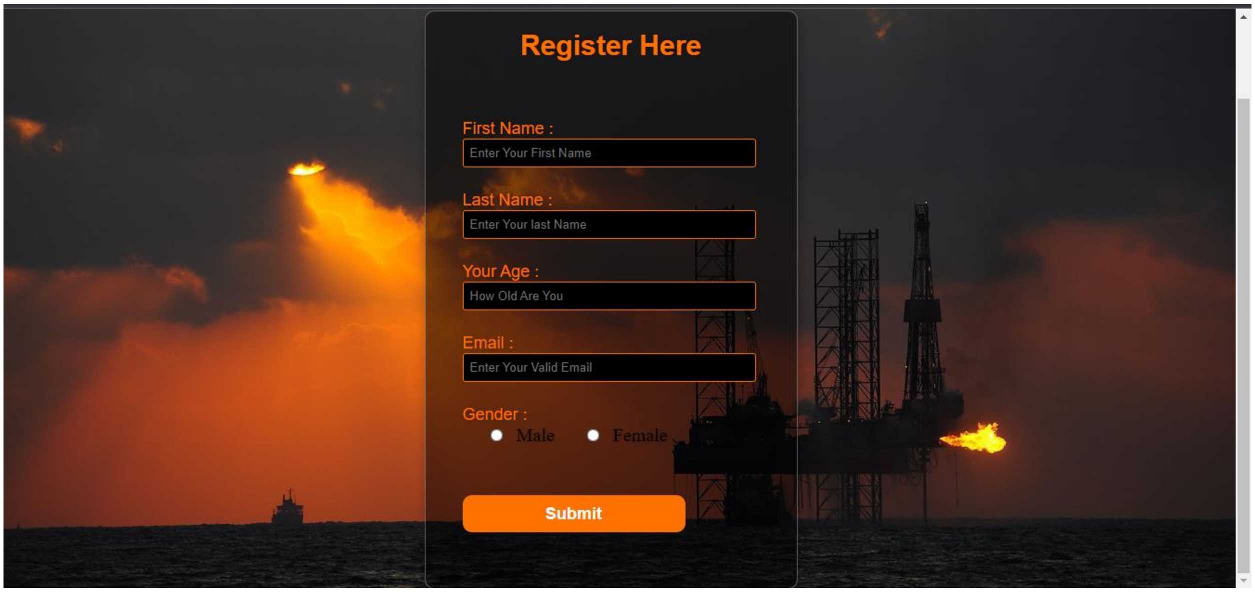
Crude Oil
Price Prediction

Crude oil means a mixture of hydrocarbons that exists in liquid phase in natural underground reservoirs and remains liquid at atmospheric pressure after passing through surface separating facilities.

Enter Email Here

Enter Password Here

Don't have an account
Sign up here



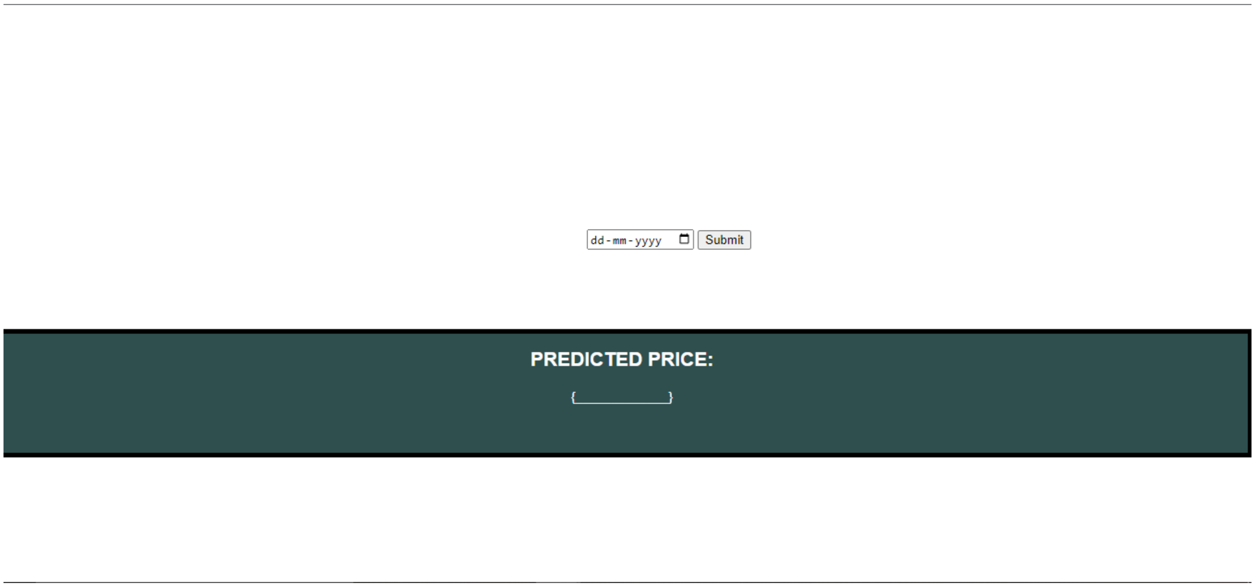
First Name : Enter Your First Name

Last Name : Enter Your last Name

Your Age : How Old Are You

Email : Enter Your Valid Email

Gender : ○ Male ○ Female



Index.html:

```
<!DOCTYPE html>

    <html lang=”en”>

        <head>

        <meta Charset=”UTF-8”>

        <title>Crude oil priceprediction</title>

        <!--<link rel=”stylesheet” href=”in.css”>

        <link
 href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&
 d isplay=swap"rel="stylesheet">
        <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
 <link rel="stylesheet"
 href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.c ss">-->
      <stl
         y
         >
         ul
         {
   list-style-type:
   none;margin:0;
   padding: 0;
   overflow:
   hidden;
   border: 1px solid
   #e7e7e7; background-
   color: #057514;
```

```
}

li {
  float: left;
}

li a {
  display: inline-
  block;color:
  rgb(78,3,3); text-
  align: center;
  padding: 14px
  16px; text-
  decoration: none;
  background-color:rgb(18, 116, 5) ;
}
li a:hover{
    border:1px
    solid;
    background-color:lightseagreen;
}
       </style>
      </head>
    <body>
        <nav class="navbar navbar-inverse">
            <div class="container-fluid">
                <ul>
                  <li class="parts"><a href="#">Home</a></li>
                  <li class="parts"><a href="predict.html">predict</a></li>
               </ul>
             </div>
           </nav>

        <h1 >Crude oil priceprediction</h1>

        <style>
            body {
              background-image:
              url('static/css/image.jpeg');background-
              repeat: no-repeat;
              background-
            attachment:fixed;
```

```
         background-size: 100%
         100%;

         }
         </style>

        <h3 style="font-family:system-ui;">
          Demand for oilis inelastic, there forethe rise in
          price is good news for producers because they will see an increasein
          their revenue.Oil importers, however, will experience
          increased costs of purchasing oil. Because oil is the largest traded commodity,the
          effectsare quite significant. A rising oil price can even shift
economic/political
         power from oil importers to oil exporters. The crude oil rice movements are
subject to
         diverse influencing factors</h3>
      </body>
      </html>
```

Predict.html:

```
<html>
   <head>
      <link rel="stylesheet" href="static/css/style.css">
      <style>
         body {
           background-image:
           url('static/css/image3.jpg');background-
           repeat: no-repeat;
           background-
         attachment:fixed;
         background-size: 100%
         100%;

         }
         </style>

   </head>
   <script>
      document.getElementByID("demo").innerHTM
L =document.getElementById("ten");
   </script>
```

```
<body>
<form action="/method" method="POST" enctype = "multipart/form-data">
<div class="container">
    <!--<div class="brand-logo"></div>-->
    <div class="brand-title">predict the oil price</div>
    <div class="inputs">
      <label>Enter Price</label>
      <input type="text" placeholder="Enter ten days price" id="ten" name="val"/>
      <buttontype="submit">Predict</button><br><br
        >The next day price is : {{prediction}}
    </div>
  </div>
</form>
</body>
</ht

ml>
```

App.

py:

```
from flask import Flask, render_template, request,redirect
import numpy as
np# from
tensorflow.k
from keras.saving.save import load_model
app = Flask(___name___,template_folder='template')

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

@app.route('/predict.html', methods=["POST",

"GET"])

@app.route('/method', methods=["POST",
"GET"])defmethod():
        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('predict.html',
        prediction=val)ifrequest.method == "GET":
            return render_template('predict.html')

    if__name___=="__main__":
        model =
        load_model(r'crudeoilprediction.h5')
        app.run(debug=True)
```

8. TESTING

8.1 Test Case

Test case analysis

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

Section	Total Cases	Not Tested	Fail	Pass
ML Model	4	0	0	4
Flask Application	4	0	0	4
IBM cloud	4	0	0	4
Exception Reporting	2	0	0	2
Final Report output	4	0	0	4

8.2 User Acceptance Testing

The purpose 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

Defect Analysis:

The report shows the number of resolved and closed bugs at each severity level and how theywere resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	3	0	0	0	3
Duplicate	1	0	1	0	2
External	0	0	0	0	0
Fixed	4	0	1	1	6

Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't fix	0	0	0	1	1
Totals	8	0	2	2	12

Test case analysis

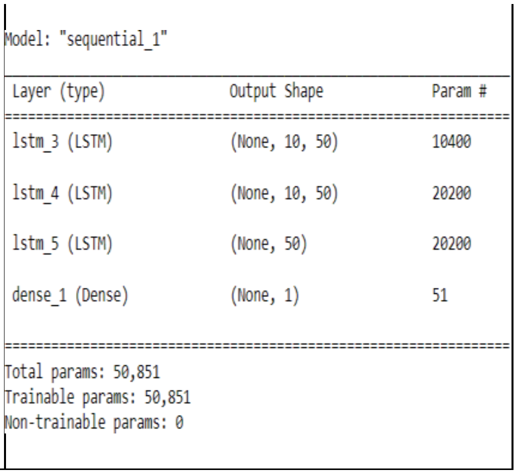
This report shows the number of test cases that have passed,failed, and untested

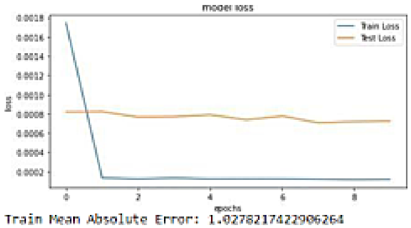
Section	Total Cases	Not Tested	Fail	Pass
ML Model	4	0	0	4
Flask Application	4	0	0	4
IBM Cloud	4	0	0	4
Exception Reporting	2	0	0	2
Final Report Output	4	0	0	4

9. RESULTS

9.1 Performance Metrics

S.No	Parameters	Values	Screenshot

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

2.	Accuracy		 Train Mean Absolute Error: 1.0278217422906264 Train Root Mean Squared Error: 1.4285248639934083 Test Mean Absolute Error: 2.780526920817909 Test Root Mean Squared Error: 3.6348234466523737

10. ADVANTAGES&DISADVANTAGES

Advantages:

1. Prediction of crude oil price can help the importers to choose the right time to buy as theywait for the prices to fall down

2. Prediction of crude oil price scan help the exporters to increase the demand

3. It can even help in shifting the political powers
4. can assist in minimizing the risks associated with volatility in oil prices

Disadvantages

5. The prediction results may lack accuracy

6. Volatility in prices may be misleading

11. CONCLUSION

LSTM network is better than other traditional neural networks for forecasting prices as it aims in using a back propagation model. Traditional neural networks such as CNN on the other hand predicts the next outgoing but doesn’t necessarily save the previous data or connection Code : which is based on feed-forwarding, in the sense the previous data is not necessary to predict the future data. LSTM focuses on storing the previous data and prediction which is rather encouraging and more approximate. The outcomes derived are relatively encouraging. The results show that large lookupsdo not necessarily improve the accuracy of the predictions of crude oil prices. Hence it can be concluded, the model with a single LSTM model is definitely the most accurate.

12. FUTURE SCOPE

The project's future potential is enormous. The project can be implemented with the real-time functionalities that are necessary. Because it is quite versatile in terms of expansion, the project canbe upgraded in the near future as and when the need arises. The complete prediction value can be increased in a much better,accurate, and error-free manner with the proposed approach. The project can be enhanced with real time data.

APPENDIX

Source Code

MODEL:

```
# DATAPREPROCESSING

## Importing the
libraries import pandas
as pd import numpy as
np
import matplotlib.pyplot as
pltimporttensorflow as tf
data=pd.read_excel(r"Crude Oil Prices
Daily.xlsx")data.head()
## Handling 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()
## FeatureScaling
from sklearn.preprocessing import MinMaxScaler
scalar=MinMaxScaler(feature_range=(0,1))
data_oil=scalar.fit_transform(np.array(data_oil).reshape(-
1,1))
```

```
## Data
Visualization
plt.title('Crude oil
price')
plt.plot(data_oil)
## Splitting data into Train and Test Data

training_size=int(len(data_oil)*0.
65)test_size=len(data_oil)-
training_size
train_data,test_data=data_oil[0:training_size,:],data_oil[training_size:len(data_oil),:1]
training_size,test_size
train_data.shape
## Creating a data setwith 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)
print(X_test.shape),print(y_test.shape)
X_train
X_train.sh
ape
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
buildinglibrariesfrom
tensorflow.keras.models import
Sequentialfrom tensorflow.keras.layers
import Dense from tensorflow.keras.layers
importLSTM
# 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()
# ConfigureThe LearningProcess
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,verb
ose=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)
importmath
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_predictplt.plot(scalar.inverse_transform(data_oil))
plt.plot(trainPredictPl
ot)
```

```
plt.plot(testPredictPl
ot) 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=1
0 i=0
while(i<1
0):
    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(1,1
1)
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 10days output
prediction")plt.plot(df3[8100:])
df3=scalar.inverse_transform(df3).tolist()
plt.title("Pastdata nad next 10 days output prediction after reversing the scaledvalues")
plt.plot(df3)
```

Index.html:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Crudeoil price prediction</title>
        <!--<link rel="stylesheet" href="in.css">
        <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&
d isplay=swap"rel="stylesheet">
        <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.c ss">-->
      <sty
         le
         >
         ul
         {
  list-style-type:
  none;margin:0;
  padding: 0;
  overflow:
  hidden;
  border: 1px solid
  #e7e7e7; background-
  color: #057514;
}
```

```
li {
  float: left;
}

li a {
  display: inline-
  block;color:
  rgb(78,3,3); text-
  align: center;
  padding: 14px
  16px; text-
  decoration: none;
  background-color:rgb(18, 116, 5) ;
}
li a:hover{
    border:1px solid;
    background-color: lightseagreen;
}
       </style>
      </head>
    <body>
        <nav class="navbar navbar-inverse">
            <div class="container-fluid">
                <ul>
                  <li class="parts"><a href="#">Home</a></li>
                  <li class="parts"><a href="predict.html">predict</a></li>
              </ul>
             </div>
           </nav>

        <h1 >Crudeoil priceprediction</h1>

        <style>
            body {
              background-image:
              url('static/css/image.jpeg');background-
              repeat: no-repeat;
              background-
            attachment:fixed;
            background-size: 100%
            100%;
            }
            </style>

          <h3 style="font-family:system-ui;">
             Demand for oilis inelastic, thereforethe 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
             effectsare quite significant. A rising oil price can even shift
economic/political
            power from oil importers to oil exporters. The crude oil rice movements are
subject to
            diverse influencing factors</h3>
        </body>
        </html>
```

Predict.html:

```
<html>
    <head>
        <link rel="stylesheet" href="static/css/style.css">
        <style>
            body {
              background-image:
              url('static/css/image3.jpg');background-
              repeat: no-repeat;
              background-
            attachment:fixed;
            background-size: 100%
            100%;

            }
            </style>

    </head>
    <script>
        document.getElementByID("demo").innerHTM
L =document.getElementById("ten");
    </script>
<body>
<form action="/method" method="POST" enctype = "multipart/form-data">
<div class="container">
    <!--<div class="brand-logo"></div>-->
    <div class="brand-title">predict the oil price</div>
    <div class="inputs">
      <label>Enter Price</label>
      <input type="text" placeholder="Enter ten days price" id="ten" name="val"/>
      <buttontype="submit">Predict</button><br><br
        >The next day price is : {{prediction}}
    </div>
  </div>
</form>
</body>
</html>
```

App.py:

```
from flask importFlask,render_template, request,
redirectimport numpy as np
# from tensorflow.k
from keras.saving.save import load_model
app = Flask(___name___,template_folder='template')

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

@app.route('/predict.html', methods=["POST",
"GET"])@app.route('/method', methods=["POST",
"GET"])
def method():
    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('predict.html', prediction=val)
    if request.method == "GET”:
        return render_tempIate('predict.html')

if enamel ==  "        main":
    model
    =Ioad_modeI(r'crudeoiIprediction.h5')
    app.run(debug=True)
```

#cIoud deployment code in ml model

```
!pip installibm_watson_machine_Iearning

from ibm_watson_machine_Iearning import
APICIientwmI_credentiaIs = (
            "url" : "https://us-south.ml.cloud.ibm.com",
            "apikey" : "cRkqykhsnLO1

Ogs_xoYjgLkNTtTS1QxyioMn1GSIQ1P5"client =APICIient(wmI_credentiaIs)

#for creating a deployment  phase
def guid_from_space_name(client,
    space_name):
```

```
    space=cIient.spaces.get_detaiIs()
    #print(space)
return(next(item for item in space['resources'] if item['entity']['name'] ==
space_name)['metadata']['id'])
space_uid =
guid_from_space_name(client,'models')
print("Space UID = "+space_uid)
client.set.defauIt_space(space_uid)
client.software_specifications.list()
software_spec_uid=
client.software_specifications.get_uid_by_name("tensorfIow_rt22.1-py3.9") software_spec_uid
```

GitHub & Project Demo Link:

GITHUB: https://github.com/IBM-EPBL/IBM-Project-10802-1659207450

PROJECT DEMO LINK:
https://drive.google.com/drive/folders/1S4yS5XljfjBB5eTZS7q90K1NMmQN6yB8

SOURCE CODE : https://github.com/IBM-EPBL/IBM-Project-10802-1659207450/tree/main/Application%20Building