PROJECT DOCUMENTATION

TEAM ID: PNT2022TMID08579

PROJECT TITLE: CRUDE OIL PRICE PREDICTION

TEAM MEMBERS:

ROJER
MANOJ
GURU PRASATH
JEROME JOHNIS
GURU PRASATH

1.INTRODUCTION:

1.1.PROJECT OVERVIEW:

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

1.2 PURPOSE:

Applying Neural Networks to predict the Crude Oil Price helps us to buy crude oil at the proper time. Time series analysis is the best option for this kind of prediction because we are using the Previous history of crude oil prices to predict future crude oil. So we would be implementing RNN(Recurrent Neural Network) with LSTM(Long Short Term Memory) to achieve the task.

2.LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

S.NO	Title	Authors	Publication Date	Methodology	Merits	Demerits
1	Brent crude oil price forecast utilizing Deep Neural Network Architectures	Amir Daneshvar and Maryam Ebrahimi	05 May 2022	Artificial Neural Network, Deep Learning	The LSTM layers results in more accurate result.	Crude oil price signals exhibit highly nonlinear and complex behavior.
2	Crude oil prices and volatility prediction by a hybrid model based on kernel extreme learning machine	Hongli Niu and Yazhi Zhao	17 September 2021	VMD-KELM	The VMD-KELM model shows a more powerful ability than other models in improving the precision of forecasting crude oil volatility.	-
3	Crude oil price prediction using ANN	Nalini Gupta and Shobhit Nigam	January 2020	Artificial Neural Network	ANN model is effective. This capture the changing pattern of prices. Prediction is accurate.	Market trends have to be planned, then the ANN model will perform.
4	Crude oil price prediction using complex network and deep learning algorithms	Makumbonori Bristone, Rajesh Prasad, Adamu Ali Abubakar	19 June 2019	Artificial Neural Network, Deep Learning	The appropriate number of LSTM layers can effectively improve the model.	The other factors that affect the crude oil price volatilities such as economic growth, exchange rate demand are not considered.

5	Daily crude oil price forecasting using Hybridizing wavelet and Artificial Neural Network Model	Ani Shabri and Ruhaidah Samsudin	16 July 2014	Artificial Neural Network	The hybrid model showed a great improvement in crude oil price modeling and produced better forecasts than ANN model alone.	-
6	Machine Learning Approach for crude oil price prediction with Artificial Neural Networks-Quantitative (ANN-Q) model	Abdullah	-	Artificial Neural Network	Returns function had successfully proved to cleanse and uniform the data from errors and noises hence, the crisp prediction result.	
7	A novel look back N feature approach towards prediction of crude oil price	Rudra Kalyan Nayak	-	ARIMA, LBNF Algorithm	Attained better training and accuracy by shifting the dataset into n class problem and more scope to classifier.	-

2.2 REFERENCES:

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9098271/
2. https://www.aimspress.com/article/doi/10.3934/mbe.2021402
3. https://www.sciencedirect.com/science/article/pii/S1877050920305913
4. https://www.sciencedirect.com/science/article/pii/S2405656119301117
5. https://www.hindawi.com/journals/mpe/2014/201402/
6. https://ieeexplore.ieee.org/document/5596602

7.https://www.researchgate.net/publication/328074349_A_novel_look_Back_N_feature_approa

ch_to wards_prediction_of_crude_oil_price

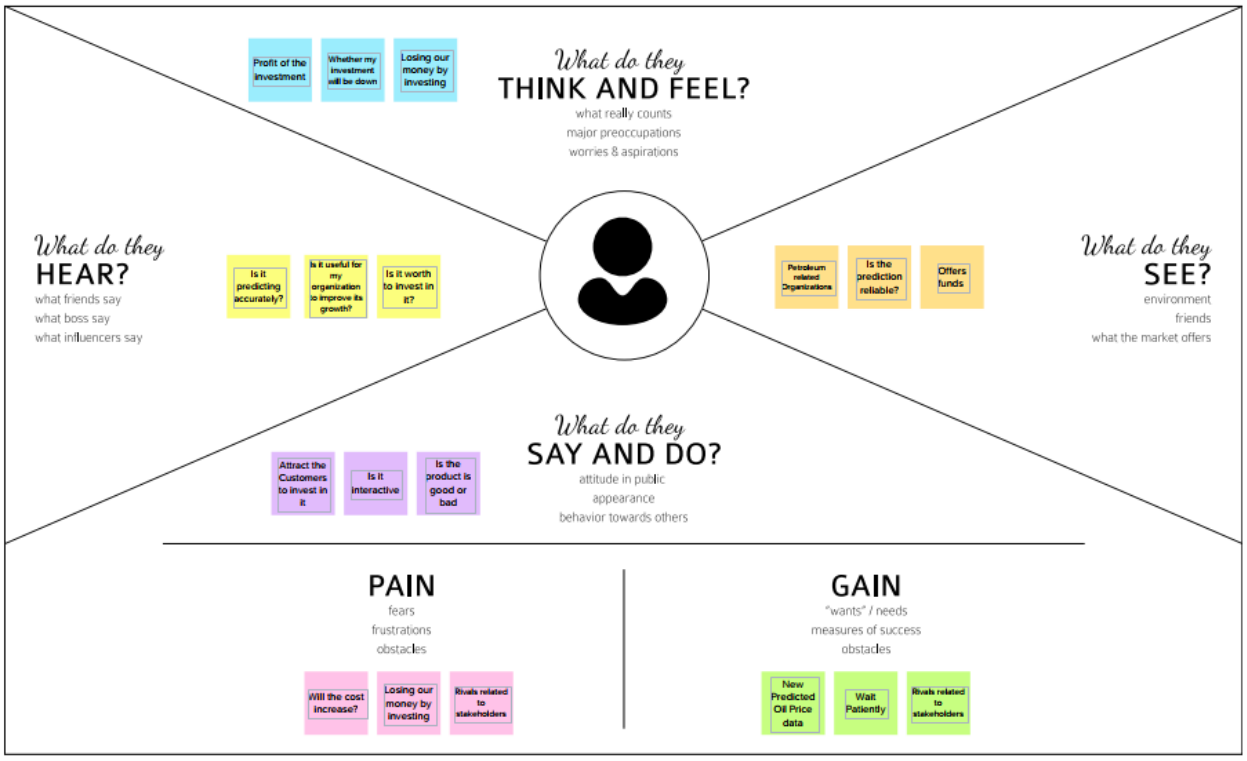
2.3 PROBLEM STATEMENT DEFINITION:

Crude oil is the world's leading fuel, and its prices have a big impact on the global environment, economy as well as oil exploration and exploitation activities. Oil price forecasts are very useful to industries, governments and individuals. Although many methods have been developed for predicting oil prices, it remains one of the most challenging forecasting problems due to the high volatility of oil prices. Now-a-days the oil price has been increasing in leaps and bounds due to certain reasons like inflation throughout the world. Hence these are derived or extracted from petroleum. To predict the values of petroleum like petroleum and Diesel within the future, we've decided to use the Machine Learning algorithms
A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

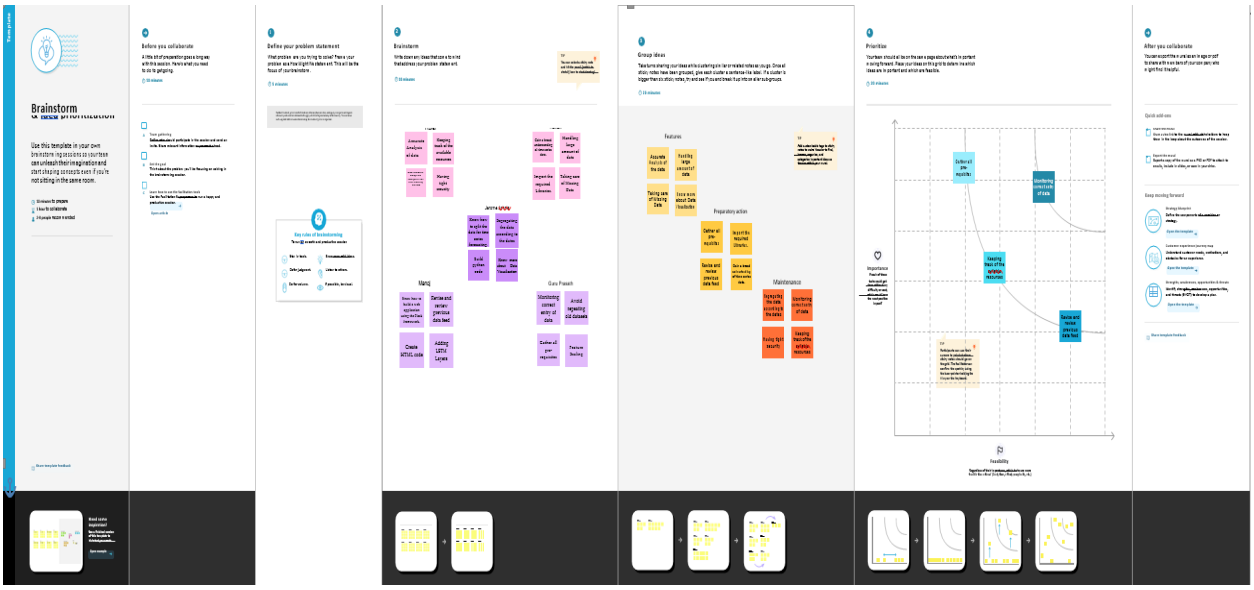
So, to maintain proper economy knowing crude oil price earlier became indispensable. In this project we will forecast the future crude oil price using Artificial Neural Network models and analyse the impact of production, consumption, exports and imports of crude oil. In this we will understand the similarity in the extent of impact of crude-oil prices on the economy. Crude oil price fluctuations have a far reaching impact on global economies and thus price forecasting can assist in minimizing the risks associated with volatility in oil prices. Price forecasts are very important to various stakeholders: governments, public and private enterprises, policymakers, and investors. Trading will be taken in efficient and cautiously way. Demand and supply will be functioned without any hindrances. Hence, Country Economy will gradually rise.

3 IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas:



3.2 IDEATION AND BRAINSTORMING:

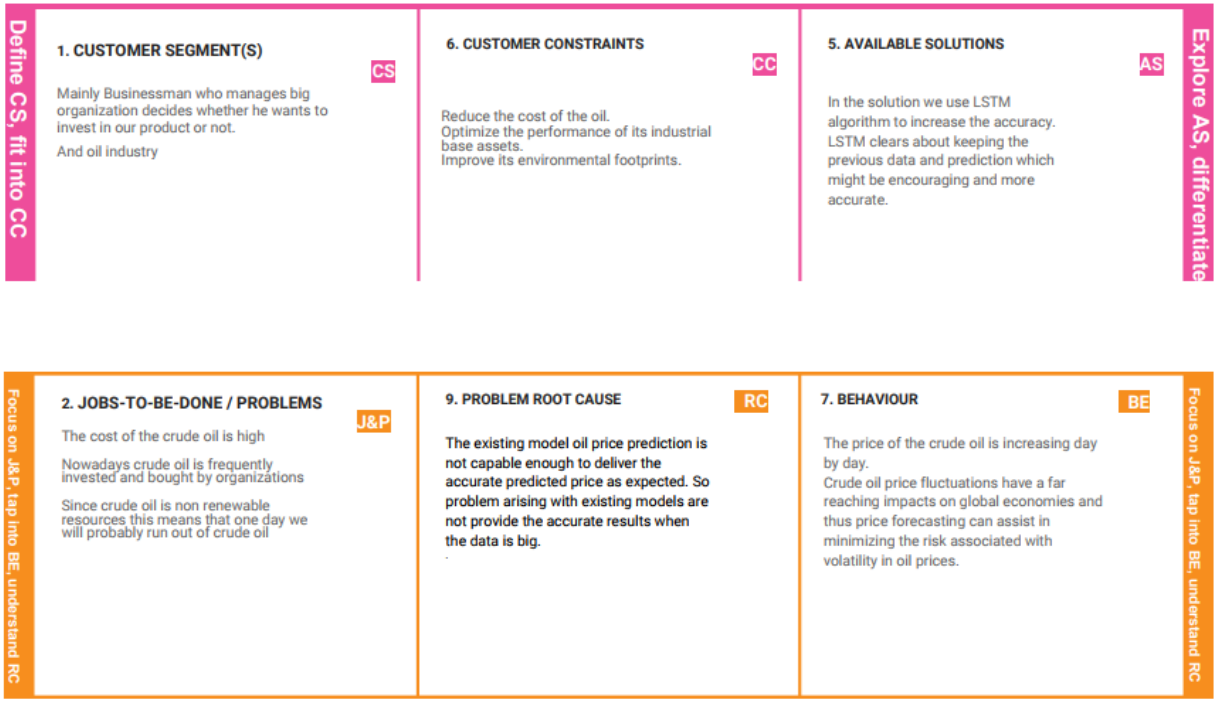


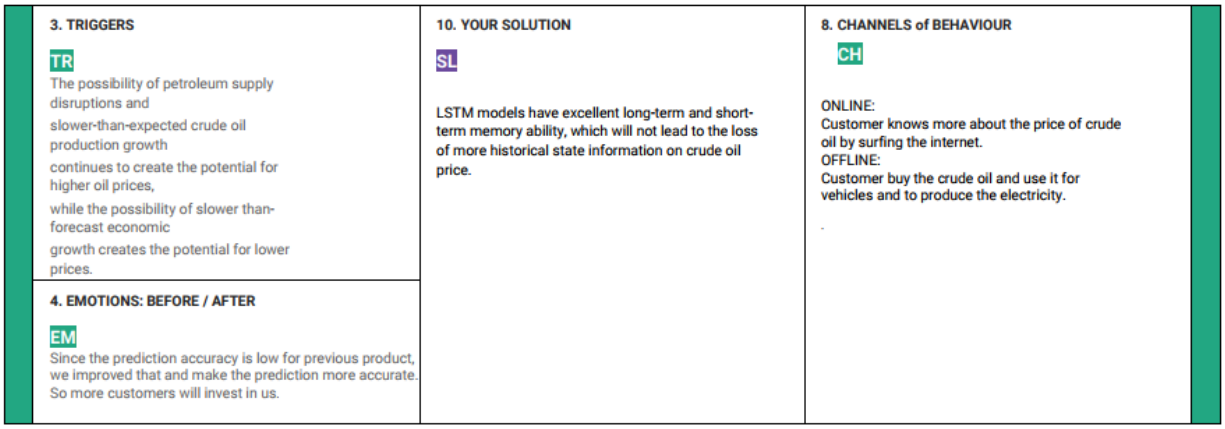
3.3 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The existing predicted models are not accurate as expected. Few factors can be described as the conjectural buying and selling, geopolitical, OPEC output, increased demand from important role in the prediction of the oil prices. Now problem arising with the current ANN and CNN models that are used as prediction model's are that they can't provide accurate results when the data is too big.
2.	Idea / Solution description	1)LSTM clears about keeping the previous data and prediction which might be encouraging and more accurate. The possible results are comparatively inspiring. 2)The LSTM model will be updated whenever new oil price data are available, and provided to model, so the model continuously evolves over time, and can capture the changing pattern of oil prices.
3.	Novelty / Uniqueness	1)Price forecasting can assist in minimising the risks associated with volatility in oil prices. 2)Price forecasts are very important to various stakeholders: governments, public and private enterprises, policymakers, and investors.
4.	Social Impact / Customer Satisfaction	1)Brand activation 2) Innovative and schemes 3) Instant reward schemes 4) Personalized consumer purchase exchanges 5) Capability building of sales personnel

5.	Business Model (Revenue Model)	1)The price of crude oil should be easily predictable from the equilibrium between demand and supply. 2)Traders analyze demand and supply factors and take calculated positions. If their prediction comes true, traders close their position to book profits way before expiry. 3) price of crude oil are changeable based from time to time.
6.	Scalability of the Solution	1)hydrodynamic conditions in oilfield operations is suggested. 2)Modern refineries typically use a high number of sensors that generate an enormous amount of data. 3)Sustainable Solution for Crude Oil using Concentrated Solar Power Technology

3.4 PROBLEM SOLUTION FIT:





4 REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

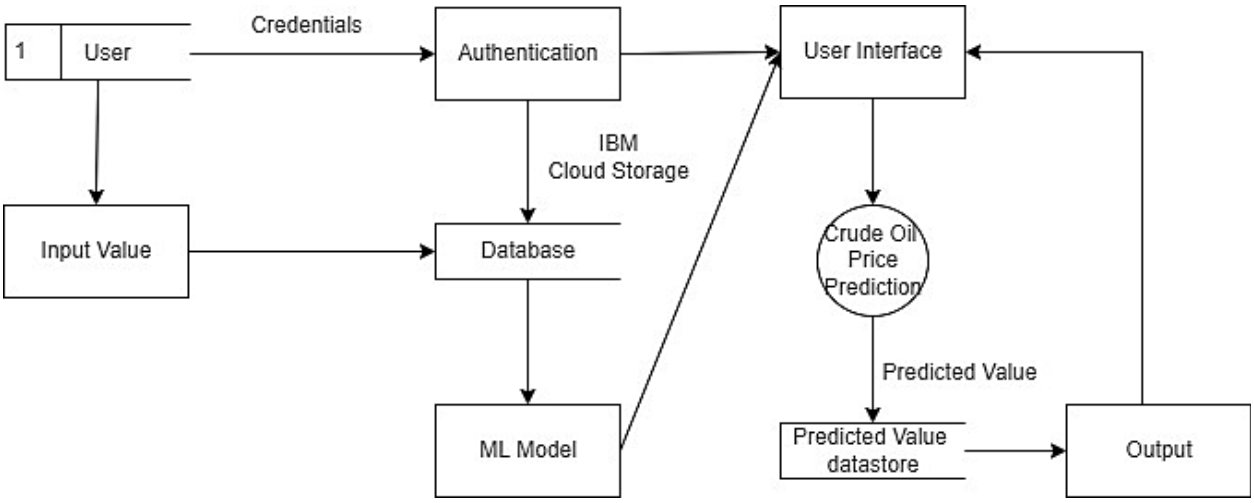
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Application	Users can download this application in play store or he can view in the browser itself.
FR-2	User Products available	The prices of the products that are available in crude oil is constantly updated.
FR-3	User Additional Features	The user can invest on the product based on the oil price charts
FR-4	User Exceptions	The user can exchange rates and currency converter

4.2 NON-FUNCTIONAL REQUIREMENTS:

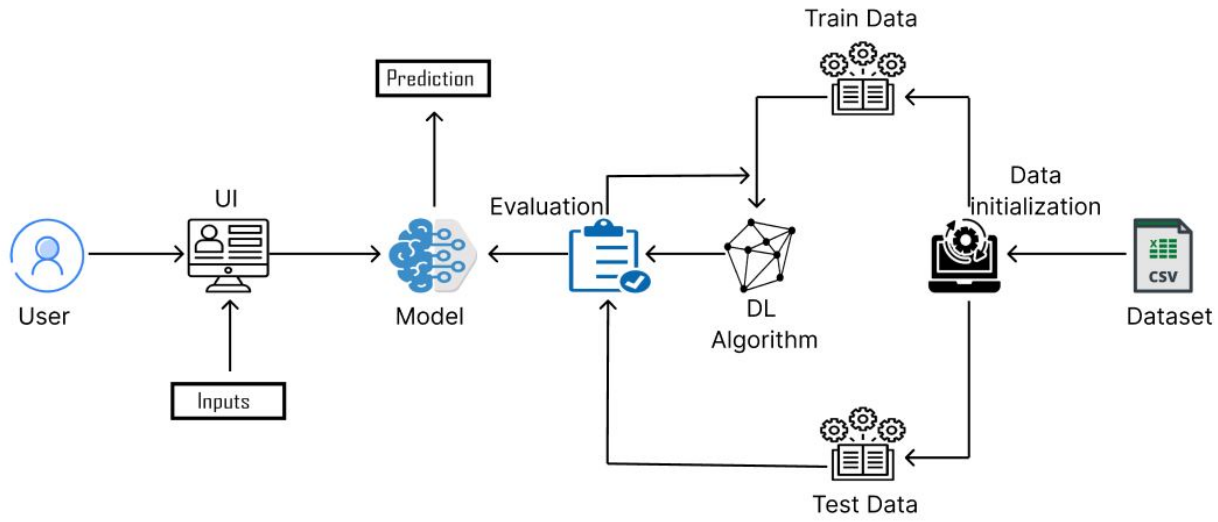
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Based on the predicted data we can decide whether we need to invest in it or to buy the product.
NFR-2	**Security**	Communication will be secure even if rising oil prices cause economic and political power to shift from oil importers to oil exporters.
NFR-3	**Reliability**	We can trust this product and its predicted value because we performed many analysis to improve the accuracy
NFR-4	**Performance**	Performance of this product is way better than the other product because we performed many test cases
NFR-5	**Availability**	Availability solution is more benefit for the importers and exporters around the world
NFR-6	**Scalability**	Scalability is much higher (90% - 95%)

5 PROJECT DESIGN:

5.1 DATA FLOW DIAGRAM:



5.2 SOLUTION AND TECHNICAL ARCHITECTURE:



5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (common People, Business man)	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 / dashboard	High	Sprint-1
		USN-2	As a user, I can register for the application by entering my Facebook id.	I can register and access the dashboard with Gmail	High	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering email & password	I can login and access using the email and password	Low	Sprint-2
	Dataset Management	USN-4	As a user, I split the dataset for training and testing.	Dataset split for training and testing.	High	Sprint-1
	Model Building	USN-5	As a user J build a model and train the model, test the model to predict the future price.	The model is build and model is trained and tested.	High	Sprint-1
Administrator	Login	USN-1	As an Administrator, I can login into the analysis page.	I can login using email and password.	High	Sprint-1
	Dashboard	USN-2	As an Administrator, I can access the Dashboard.	I can update the details of the features.	Medium	Sprint-2
	Authentication	USN-3	As an Administrator, I can verify the identity of the user.	I can check the email and password is correct or not.	High	Sprint-1
	Authorization levels	USN-4	As an Administrator, I can determine the extent of system rights that the user has access to.	I can verify the user has been properly identified and authenticated.	Medium	Sprint-2

6.PROJECT PLANNING AND SCHEDULING:

6.1 SPRINT PLANNING AND ESTIMATION:

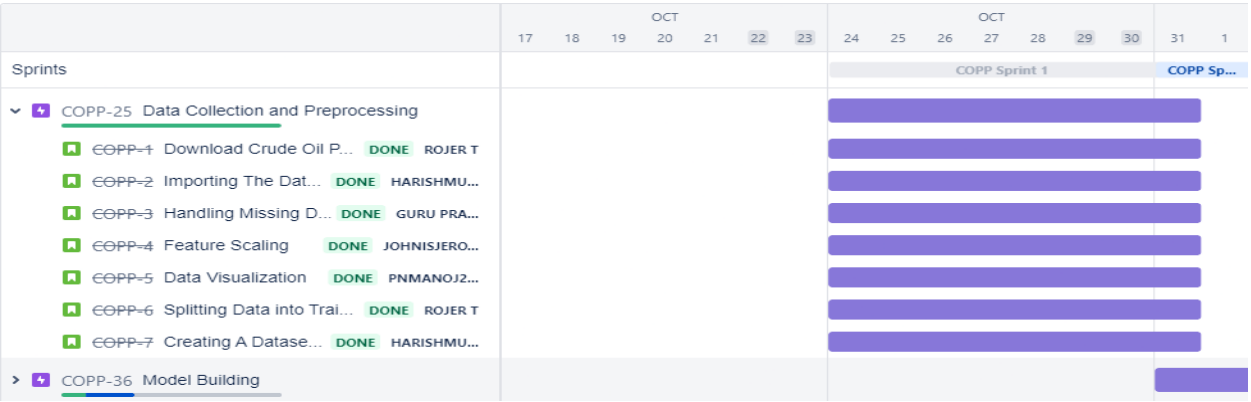
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Download Crude Oil Price Dataset	2	Medium	Rojer
Sprint-1	Data Preprocessing	USN-2	Importing The Dataset into Workspace	1	Low	Guru Prasath
Sprint-1		USN-3	Handling Missing Data	3	Medium	Jerome Johnis
Sprint-1		USN-4	Feature Scaling	3	Low	Manoj
Sprint-1		USN-5	Data Visualization	3	Medium	Harish
Sprint-1		USN-6	Splitting Data into Train and Test	4	High	Rojer
Sprint-1		USN-7	Creating A Dataset with Sliding Windows	4	High	Guru Prasath
Sprint-2	Model Building	USN-8	Importing The Model Building Libraries	1	Medium	Jerome Johnis
Sprint-2		USN-9	Initializing The Model	1	Medium	Manoj
Sprint-2		USN-10	Adding LSTM Layers	2	High	Harish
Sprint-2		USN-11	Adding Output Layers	3	Medium	Rojer
Sprint-2		USN-12	Configure The Learning Process	4	High	Guru Prasath
Sprint-2		USN-13	Train The Model	2	Medium	Jerome Johnis
Sprint-2		USN-14	Model Evaluation	1	Medium	Manoj
Sprint-2		USN-15	Save The Model	2	Medium	Harish
Sprint-2		USN-16	Test The Model	3	High	Rojer
Sprint-3	Application Building	USN-17	Create An HTML File	4	Medium	Guru Prasath
Sprint-3		USN-18	Build Python Code	4	High	Jerome Johnis
Sprint-3		USN-19	Run The App in Local Browser	4	Medium	Manoj
Sprint-3		USN-20	Showcasing Prediction On UI	4	High	Harish
Sprint-4	Train The Model OnIBM	USN-21	Register For IBM Cloud	4	Medium	Rojer
Sprint-4		USN-22	Train The ML Model On IBM	8	High	Guru Prasath
Sprint-4		USN-23	Integrate Flask with Scoring End Point	8	High	Jerome Johnis

6.2 SPRINT DELIVERY SCHEDULE:

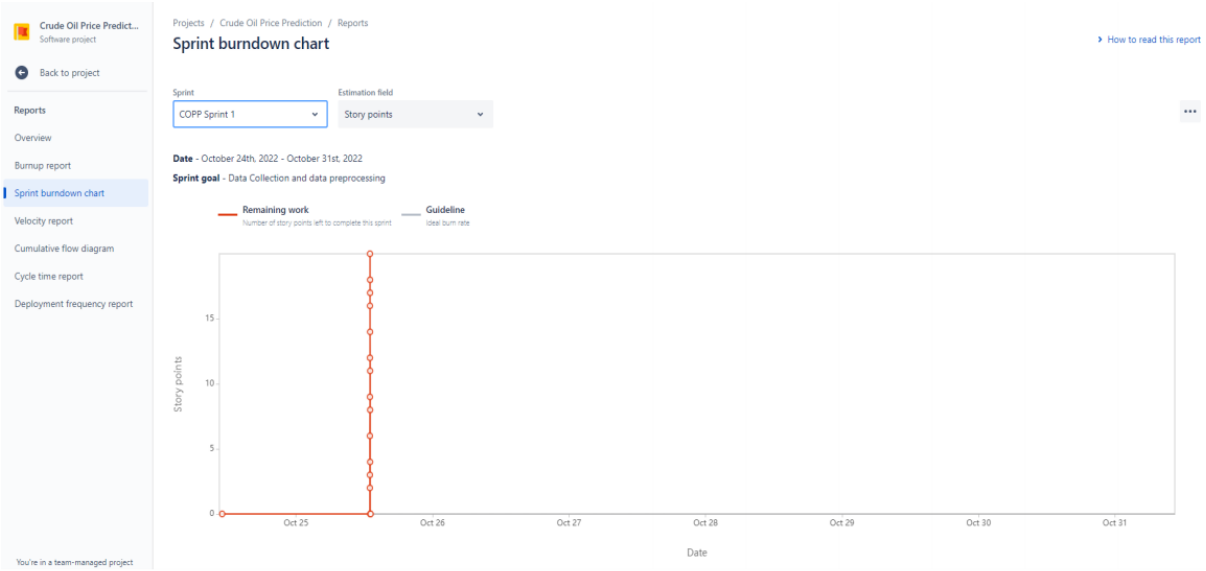
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	19	6 Days	31 Oct 2022	05 Nov 2022	19	05 Nov 2022
Sprint-3	16	6 Days	07 Nov 2022	12 Nov 2022	16	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 Nov 2022

6.3 REPORTS FROM JIRA:

SPRINT 1:

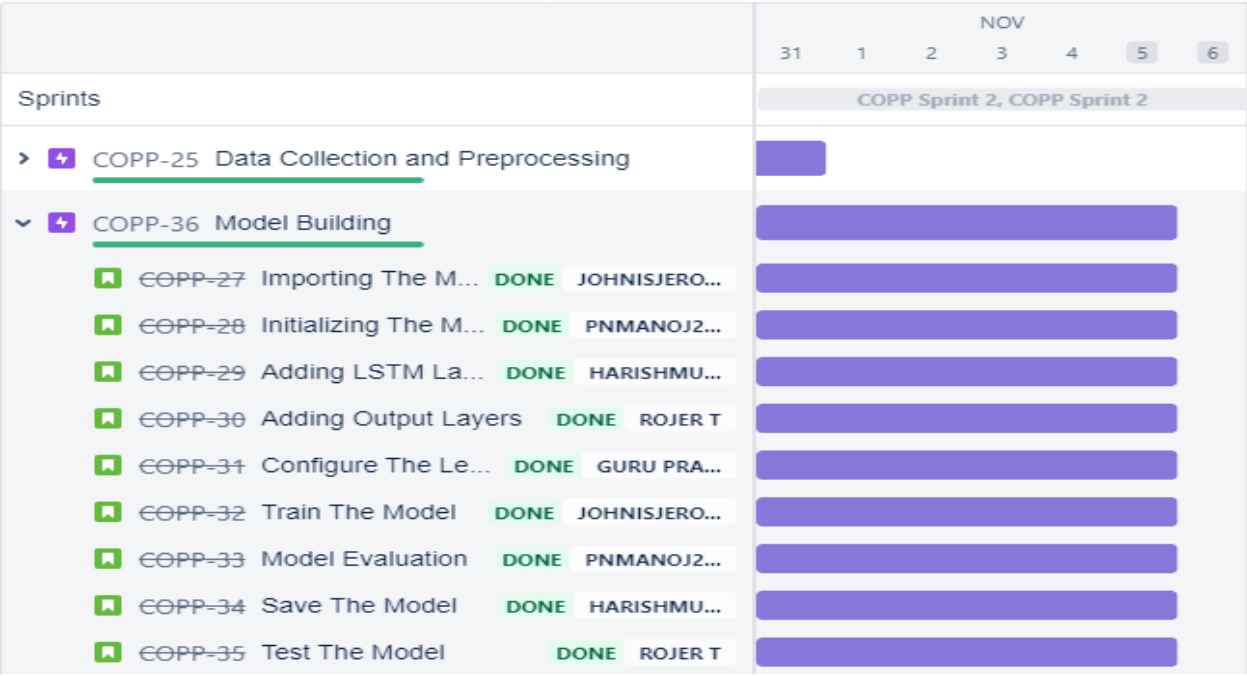


- SPRINT BURNDOWN CHART:

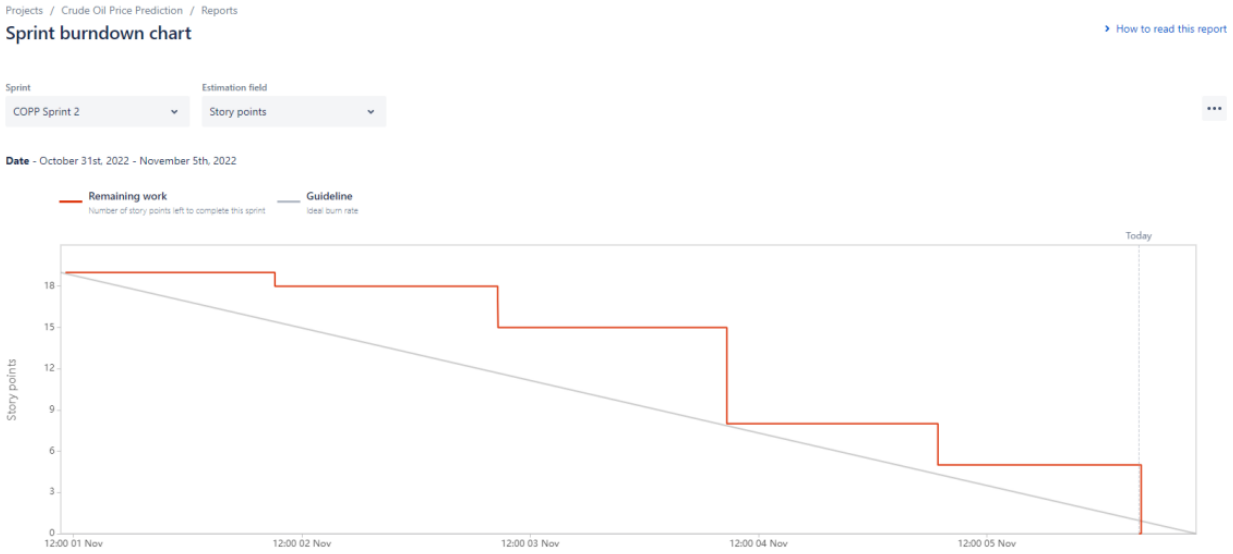


- VELOCITY REPORT:

SPRINT 2:



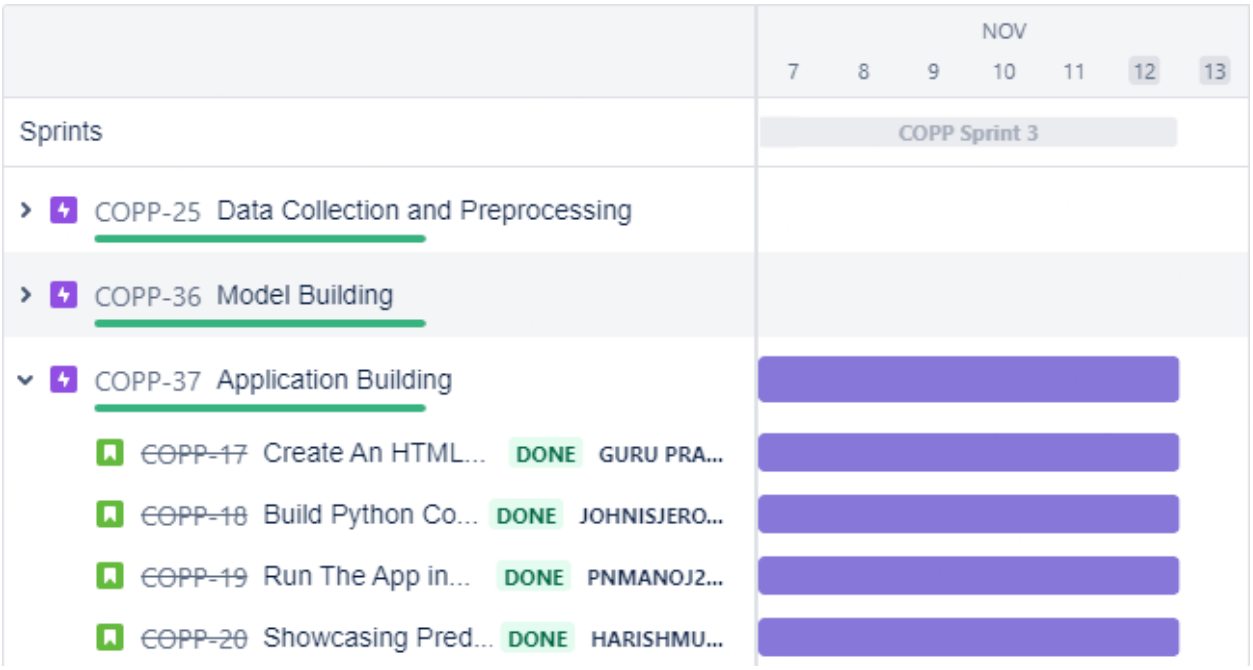
- SPRINT BURNDOWN CHART:



- VELOCITY REPORT:

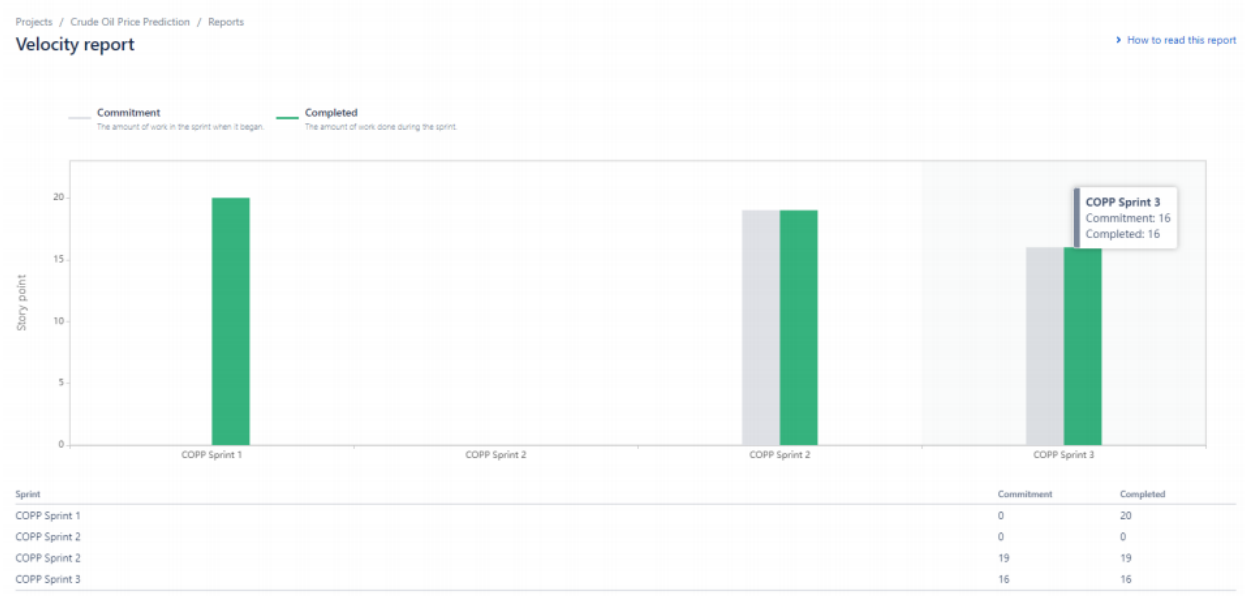
Sprint	Commitment	Completed
COPP Sprint 1	0	20
COPP Sprint 2	0	0
COPP Sprint 2	19	19

SPRINT 3:



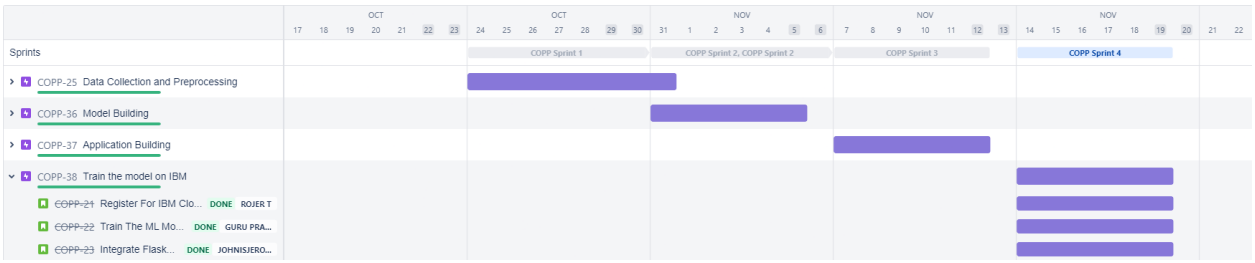
- SPRINT BURNDOWN CHART:

- VELOCITY REPORT:

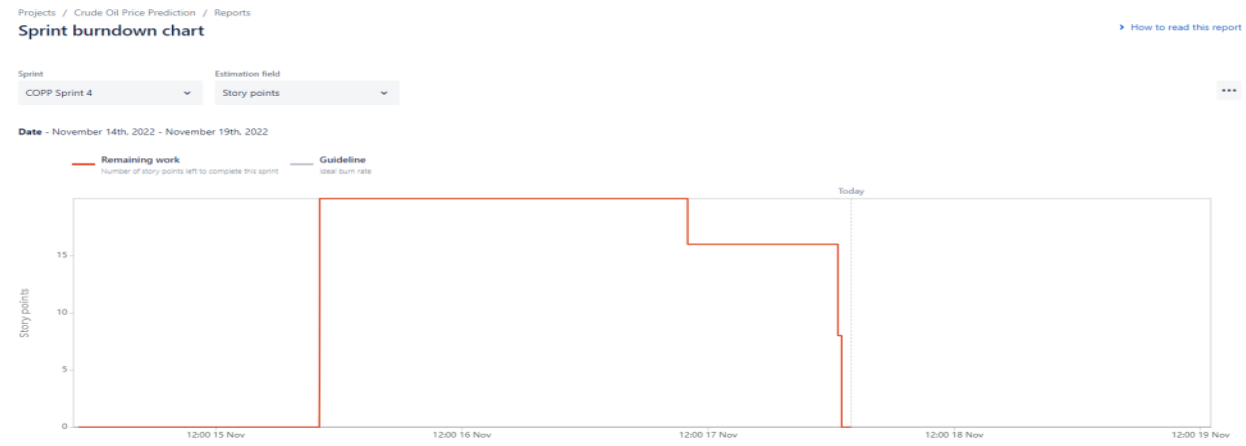


Sprint	Commitment	Completed
COPP Sprint 1	0	20
COPP Sprint 2	0	0
COPP Sprint 2	19	19
COPP Sprint 3	16	16

SPRINT 4:



- SPRINT BURNDOWN CHART:



- VELOCITY REPORT:

Sprint	Commitment	Completed
COPP Sprint 1	0	20
COPP Sprint 2	0	0
COPP Sprint 2	19	19
COPP Sprint 3	16	16
COPP Sprint 4	0	20

7 CODING AND SOLUTIONING:

7.1 FEATURE 1:

Long Short-Term Memory (LSTM) is one type of recurrent neural network which is used to learn order dependence in sequence prediction problems. Due to its capability of storing past information, LSTM is very useful in predicting oil prices.

7.2 FEATURE 2:

A User Interface for forecasting based on the past 10 days price

8 TESTING:

8.1 TEST CASES:

SPRINT 1:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Sprint_01_TC_01	Functional	Model	Verify whether the data is loaded properly				Data should be loaded properly	Working as expected	Pass				Rojer T
Sprint_01_TC_02	Functional	Model	Verify whether any data is missing				Missing Data should be founded and handled	Working as expected	Pass				Manoj P N
Sprint_01_TC_03	Functional	Model	Verify the feature scaling scenario				Standard featurizing and scaling is performed.	Working as expected	Pass				GuruPrasath M
Sprint_01_TC_04	Functional	Model	Verify the splitting of data into train and test				Data should be split.	Working as expected	Pass				Jerome Johnis J
Sprint_01_TC_05	Functional	Model	Creating the dataset with sliding windows				Dataset with siding windows should be created	Working as expected	Pass				Harish S

SPRINT 2:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Sprint_02_TC_01	Functional	Model	Verify whether the LSTM Layers are added				The Layers are to be added implicitly'	Working as expected	Pass				Rojer T
Sprint_02_TC_02	Functional	Model	Verify if the model is configured				The Model Configuration should be immaculte.	Working as expected	Pass				Manoj P N
Sprint_02_TC_03	Functional	Model	Verify the model is trained				The model should be trained	Working as expected	Pass				GuruPrasath M
Sprint_02_TC_04	Functional	Model	Verify whether the model is evaluated on proper metrics				The Model Evaluation should be performed	Working as expected	Pass				Jerome Johnis J
Sprint_02_TC_05	Functional	Model	Verify whether the model is saved				The Model should be saved.	Working as expected	Pass				Harish S

SPRINT 3:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Sprint_03_TC_01	Functional	Model	Verify if the navigation is working properly				Navigations are working properly.	Working as expected	Pass				Rojer T
Sprint_03_TC_02	Functional	Model	Verify if the model is loaded properly				Model is loaded properly.	Working as expected	Pass				Manoj P N
Sprint_03_TC_03	Functional	Model	Verify scalar.save is working properly				Scalar.save is working properly	Working as expected	Pass				GuruPrasath M
Sprint_03_TC_04	Functional	Model	Verify is the form data sent from the UI is properly received at the Python Flask				The data transmission is done perfectly.	Working as expected	Pass				Jerome Johnis J
Sprint_03_TC_05	Functional	Model	Verify if the predictions are correlated displayed				The prediction is displayed wrongly.	Working as expected	Pass				Harish S

SPRINT 4:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Sprint_01_TC_01	Functional	Model	Verify whether the cloud storage is created				Cloud Storage is present	Working as expected	Pass				Rojer T
Sprint_01_TC_02	Functional	Model	Verify whether the dataset is loaded for training				Dataset is loaded	Working as expected	Pass				Manoj P N
Sprint_01_TC_03	Functional	Model	Verify whether the deployment space is created				Deployment space is created	Working as expected	Pass				GuruPrasath M
Sprint_01_TC_04	Functional	Model	Verify if the DL Model is deployed				The DL Model is deployed	Working as expected	Pass				Jerome Johnis J
Sprint_01_TC_05	Functional	Model	Verify if the predictions are made				The Predictions are made	Working as expected	Pass				Harish S

8.2 USER ACCEPTANCE TESTING:

DEFECT ANALYSIS:

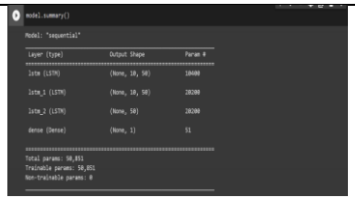
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	3	0	1	0	4
Duplicate	1	0	1	0	2
External	0	0	0	0	0
Fixed	4	0	1	1	6
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	1	1
Totals	8	0	3	2	13

TEST CASE ANALYSIS:

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.	Parameter	Values	Screenshot
1.	Model Summary	-	
2.	Accuracy	Training Accuracy - 2.03291747578067 Validation Accuracy - 2.9096238300262343	

10 ADVANTAGES AND DISADVANTAGES:

- The model is able to forecasting the price accurately based on the past 10 days data.
- But whenever the commodity is affected by external factors which are caused naturally then the predictions are bad. This cannot be predicted by machine learning model

11 CONCLUSION:

Therefore the ml was deployed as a web app and the user interface is handy for stakeholders who do not have much knowledge in programming. The predicted value is displayed in the user interface.

12 FUTURE SCOPE:

The machine learning model can be improved by converting it from univariate to multi variate model for better understanding the price value. This can result in a great boost in the prediction accuracy.

13 APPENDIX:

SOURCE CODE:

app.py:

```
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
import requests


# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "qBVjfLpxSGroBr5cd8BNRjoC3pZhoHyrC9bKxMZ1IePl"
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}


app = Flask(__name__)
model = load_model('D:/19BCS018_ROJER T/IBM Proj/Project Development Phase/Sprint 4/Integrate flask with scoring end points/crudeoilprediction.h5')

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


@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("D:/19BCS018_ROJER T/IBM Proj/Project Development Phase/Sprint 4/Integrate flask with scoring end points/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))
        b = x_input.tolist()
        print(b)

        model = load_model('D:/19BCS018_ROJER T/IBM Proj/Project Development Phase/Sprint 4/Integrate flask with scoring end points/crudeoilprediction.h5')
        output = model.predict(b)
        print(output[0][0])

        val = sc.inverse_transform(output)
        payload_scoring = {"input_data": [{   "values": [[b]]    }]}

        response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7179b8e9-1924-45b8-9dec-5f1ea85e6aa2/predictions?version=2022-11-14', json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
        predictions = response_scoring.json()
        print(response_scoring.json())
        return render_template('index.html' , prediction = val[0][0])

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

index.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>Crude Oil Price Prediction</title>
    <link rel="icon" type="image/png" href="../static/css/oil-icon.png">
    <link rel="stylesheet" href="../static/css/styles.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:  rgb(237, 144, 12);;
            box-shadow: 0 0 8px 0  rgb(237, 144, 12);;
        }
    </style>
</head>
<body>
    <header class="homepage" >
        <div class="navbar">
            <a href="#"><h1>Crude Oil <span>Price Prediction</span></h1></a>
            <!-- <div class="nav-list">
                <ul>
                    <li><a href="#">Home</a></li>
                </ul>
            </div> -->
            <!-- <img src="../static/css/menu-icon.png" alt="menu" class="menu-icon"> -->
        </div>
        <div class="Div">
            <h1><span>Crude </span>oil <span>Price</span> Prediction</h1>
            <p>
                With our web app, you can easily predict the prices of crude oil by entering the prices of
crude oil for 10 days as input!!
            </p>
            <form action="/predict" method="POST" enctype = "multipart/form-data">
                <input type="text" name="val" placeholder="Enter the crude oil price for first 10 days" >
                 <br> <br> <br>
                <input type="submit" class="button-class" placeholder="Predict">
```

```
        </form><br> <br>
        <div>
            <h2>Predicted price is: {{prediction}}</h2>

        </div>
    </div>
    <img style="margin-top: 200px;margin-right: 10px"  alt="crude" src="../static/css/crude.png"
ALIGN="right" width="200px" height="200px">
  </header>

</body>
</html>
```

GITHUB LINK:https://github.com/IBM-EPBL/IBM-Project-2114-1658457925

PROJECT DEMO LINK: https://vimeo.com/772467830/e6f9b51ca0