CRUDE OIL PRICE PREDICTION USING LSTM

A PROJECT REPORT

Submitted by

TEAM ID: PNT2022TMID28643

KARAN KUMAR R	142219104049
KARTHIKRAJA N	142219104051
KAVIMUKILAN M	142219104057
KARNA M	142219104050

BACHELOR OF ENGINEERING
IN

COMPUTER SCIENCE AND ENGINEERING

DHANALAKSHMI COLLEGE OF ENGINEERING

TABLE OF CONTENTS

ABSTRACT

Research on crude oil price forecasting has attracted tremendous attention from scholars and policymakers due to its significant effect on the global economy. Besides supply and demand, crude oil prices are largely influenced by various factors, such as economic development, financial markets, conflicts, wars, and political events. Most previous research treats crude oil price forecasting as a time series or econometric variable prediction problem. Numerous research has recently been conducted in an effort to analyze the difficulty of predicting oil prices and find the best solutions. Although recently there have been research considering the effects of real-time news events, most of these works mainly use raw news headlines or topic models to extract text features without profoundly exploring the event information. It will be beneficial for our government, businesses, and investors to anticipate its demands. As part of this research, artificial neural networks (ANNs) will be built to forecast crude oil prices. In this study, we suggest a cutting-edge method for predicting the price of crude oil using analytical. The future price of the crude oil will be predicted on basis of the inputs given by the user. The predicted price would be for the next day. Hence, it is concluded that the proposed model achieved higher forecasting accuracy and takes less computational time with the modes' reconstruction as opposed to using all the decompose modes. As a part of future scope, there is being an idea to improve the model by considering the latest news, disaster, tweet, and social media sensitive messages.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Project Overview

Owing to the fact that crude oil provides around one-third of the world's energy needs, crude oil is important to the global economy. Additionally, changes in oil prices have a big impact on both countries' economies that export and buy oil. Forecasting the oil price accurately would assist policymakers in enacting the right legislation and selecting the best energy sources. However, because there are numerous factors that affect oil prices, forecasting researchers have found it difficult to estimate the price of crude oil. Economic growth, conflicts, wars, and breaking news all have a significant impact on oil price fluctuations in addition to the basic market elements like supply, demand, and inventory. For instance, oil producers were paying buyers to take the commodity off their hands because they were concerned that storage space might be depleted in May 2020. On April 20, 2020, the price of WTI oil even became negative for the first time ever. Another recent example is the higher association between changes in crude oil prices and the severity of the COVID-19 epidemic. Since the majority of this information is found in unprocessed texts, characterizing and modelling these nonlinear and non quantitative factors is difficult.

1.2 Purpose

The three primary factors that impact the price of oil are:

- **Supply and demand**

The idea of supply and demand is rather simple. Price should rise as demand (or supply) rises or falls. Price should decrease when supply grows or as demand declines. Actually, the oil futures market is where the price of oil as we know it is set. A legally binding agreement known as an oil futures contract offers one the right to buy oil by the barrel at a specified price on a specified date in the future. In a futures contract, each party is responsible for carrying out their portion of the deal before the deadline.

- **Cost of production**

Cost of production refers to the total cost incurred by a business to produce a specific quantity of a product or offer a service. Production costs may include things such as lab our, raw materials, or consumable supplies. In other words, the cost of production is defined as the expenditures incurred to obtain the factors of production such as lab our, land, and capital, that are needed in the production process of a product.

- **Market sentiment**

Sentiment is the other important factor that impacts oil prices. The simple expectation that oils demand would rise sharply at some point in the future can cause speculators and hedgers to buy up oil futures contracts, driving up oil prices now.

There used to be a recognisable seasonal swing in oil prices. As oil dealers anticipated a large demand for driving over the summer vacation, they increased in the spring. Prices fell in the fall and winter once the demand peaked.

Geopolitical instability and civil upheaval also have a significant impact on global supply and prices.

There are several reasons why oil prices are more unpredictable now, but five are the most significant.

- **The Russian Invasion of Ukraine**

Russia is the third-largest producer of liquid fuels and petroleum, so when the country invaded Ukraine in late February 2022, it had immediate impact on Brent crude oil futures prices.10 As the conflict continued, the prices of crude oil settled in out on an upward trajectory, reaching nearly $130/b in early March, and staying well above $100/b into April.

- **US Oil Supply**

The coronavirus pandemic and natural events are still affecting oil demand and supply. The U.S. experienced a drop in production following Hurricane Ida in September as the storm shut at least nine refineries.

The EIA estimates that U.S. crude oil production will average 12.01 million b/d in 2022 and 12.95 million b/d in 2023.11

- **Diminished OPEC Output**

Oil price increases also reflect supply limitations by the Organization of the Petroleum Exporting Countries (OPEC) and OPEC partner countries. In 2020, OPEC cut oil production due to decreased demand during the pandemic. It gradually increased oil output through 2021 and into 2022. Supply chain disruptions in late 2021 affected global trade as well.

At its most recent meeting in December 2021, OPEC stated it would continue to gradually adjust oil production upward by 0.4 million barrels per day (mb/d) in January 2022.

- **Natural Gas**

Countries in Asia have relied on coal to generate power, but recent shortages have turned them to natural gas. Higher temperatures in parts of Asia and Europe have led to high demand for natural gas to generate power.

COVID-19 has hampered Europe's natural gas production, and a colder-than-expected heating season in early 2021 reduced supplies further.

As a result, natural gas prices soared in 2021 and are expected to remain high in 2022 and affected countries have turned to gas-to-oil switching to reduce power generation costs.

- **Global Inventory Draw**

As a reduction in oil production continues globally, countries are forced to draw from their stored reserves (not including the strategic petroleum reserves). This steady draw of oil is contributing to the increase in prices because inventories are decreasing.

Models incorporating economic parameters such as supply, and demand and their determinants are known as structural models. Even though structural models are found to be the most logical ways of modelling the prices of industrial products, the price of crude oil is affected by many other factors. One of these factors is that the price of crude oil is determined in the futures market which enables the purchase of a predefined amount of oil at a particular price in the future. Additionally, only 1% of the crude oil traded in futures contracts results in the actual purchase of a physical commodity; its chief purpose is to make money out of price fluctuations in crude oil. Hence the price of crude oil behaves more like a financial asset and therefore is more representative of the expectations of traders rather than just predictions based on economic theories of supply and demand.

There are other categories of models which are non-structural and consider time variation of crude oil prices, known as time series models. It is difficult to obtain reliable data to formulate a structural model, while time series data for crude oil prices is easily available and hence it is easier to build a time series model. We focus on time series modelling of crude oil prices in this article.

In time series models, it is assumed that the current price of crude oil reflects the effects of all influencing factors, and that price forecasting can be done based on the behaviour of past crude oil prices. The main assumption in such models is that the past behaviour of oil prices can explain future prices. Although time series models can capture trends or any cyclical patterns in the data, there are limitations to the forecasting capability of these models when trend reversals are observed in the data, or the repeating pattern captured in the model is not followed in future prices. Different trends in a time series can be classified as increasing, decreasing and periodic patterns. Time series models are quite useful and forecast reasonably well when the data follows any of these types of trends.

We can easily observe the downtrends, uptrends and repeating patterns in crude oil prices within specific years. Crude oil monthly price data is obtained from the US Energy Information Administration (EIA) website.1 Different subsets of crude oil price data are formed to demonstrate the utility of time series modelling and its limitations in some scenarios.

Time Series Modelling Techniques

Several methods are proposed in the literature to build time series models. They include autoregressive integrated moving average (ARIMA), generalised auto regressive conditional heteroscedastic (GARCH), Holt-Winters, autoregressive neural networks, and support vector regression.2 Various hybrid models are also suggested such as combination of ARIMA and neural networks with support vector regression, genetic algorithms and wavelets.3-7 Discussion of various methodologies applied for crude oil price modelling can be found in review articles available in the literature.8,7 We have used ARIMA and auto regressive neural networks for modelling oil prices, as these techniques cover both linear and non-linear types of modelling. A short description of these methods is given below.

ARIMA

ARIMA is the most widely used and well-known technique for time series analysis, developed by Box and Jenkins. In an ARIMA model, future values are predicted as a linear combination of previous oil prices and the associated errors. This model consists of three parts: the AR (auto regressive) component is a linear combination of past observations; MA (moving average) is a linear combination of lagged error terms; and I (integrated) replace the original series with differenced series.

Auto regressive Neural Network

An autoregressive neural network (ANN) is a non-linear model in which future prices are expressed as a non-linear function of lagged prices in the series, in contrast to linear modelling in ARIMA. Additionally, neural network-based models have the ability to learn and capture patterns in data sets without the need to specify the exact model form. Multilayer perceptron (MLP) is the most widely used ANN in forecasting problems. Typically, the model is composed of input layer, hidden layer and output layer. The connecting nodes in these layers are called neurons. Input to the neurons is mapped using transfer functions and the weighted average of output from all the nodes is sent to next layer. There are various parameters that need to be specified for an ANN model: number of hidden layers, number of neurons in each layer, type of transfer function, and number of lags. The selection of appropriate network parameters is crucial to the fitting and forecast accuracy of an ANN model. We have used the nnetar function in R to build a neural network model.

Benefits of predicting crude oil prices:

- Some Sectors Thrive It probably counts as obvious that there are sectors that thrive when oil prices march upward. High prices for oil fuel the same sort of process as in any other sector; suppliers look for ways to provide more of the product and take advantage of those higher prices. For energy, then, that means opportunities for companies involved in exploration (seismic survey, for instance), drilling, production and servicing.

- New Technologies Become Viable Cheap oil is problematic for companies and industries looking to supplant oil. While most people can agree that there are vague and nebulous costs associated with accessing and utilizing oil (pollution, for starters), the United States has been reticent to translate those costs into higher energy taxes. What's more, it is not clear that higher taxes on fossil fuels in Europe and much of Asia really do anything to mitigate environmental damage beyond reducing consumption. All in all, then, when oil prices are low it is very hard for cleaner energy technologies to compete effectively on price.

- Changes in Behaviour For those who believe that burning oil (and other hydrocarbons) is generally a bad thing, higher prices that lead to lower use has to be counted as a benefit. When people are faced with higher prices and no obvious substitutes, they will consume less assuming that their demand is relatively elastic.

- Alternatives Come to the Fore If increased exploration and production is a normal by-product of higher oil prices, so too is substitution. When Nazi Germany faced oil shortages in World War II, methods of producing oil, diesel and gasoline substitutes from vegetable oils, animal fats and coal were thoroughly explored. Likewise, the oil crisis of the 1970s gave the development of ethanol in Brazil a major boost.

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem:

The existing problem can be broadly classified into the following

- Predictive Analytics
- Determining the Crude Oil Price
- Neural Network for Predictive Analytics
- RNN LSTM Network

A. Predictive Analytics

Predictive analytics is a cutting-edge field of study that employs statistical models and other scientific methods to assess hazy future opportunities with a view to producing actual forecasts and verifying the accuracy of these forecasts in the real world [2]. The predictive analytics model can provide meaningful insights by extracting knowledge from data and use statistical or machine learning methods to assist with the analytical task.

B. Determining the Crude Oil Price

Various significant elements, including a supply and demand curve, the present financial market, the commodities market, speculative factor, and geopolitical factor, may have an impact on fluctuations in crude oil prices, according to Miao et al. [3]. Each of these variables has a number of determining factors (sub-variables) that impact the price of the commodity.

According to an article published on the Caltex website [4], the fuel (such as petrol) prices change is closely related to the cost of crude oil–and it has a long-term effect on the fluctuation of the commodity price. Additionally, the cost of crude oil alone has contributed to nearly 50 percent of the retail petroleum price [4].

C. Neural Network for Predictive Analytics

The neural network contains a set of neurons (or perceptron's) which acts as processing units [5], interlinked, and may reside within an extensive network.
The most basic form of the neural network consists of an input layer, one hidden layer, and an output layer [6], as visualized in Figure 1. The number of hidden layers may vary based on the complexity of computation.

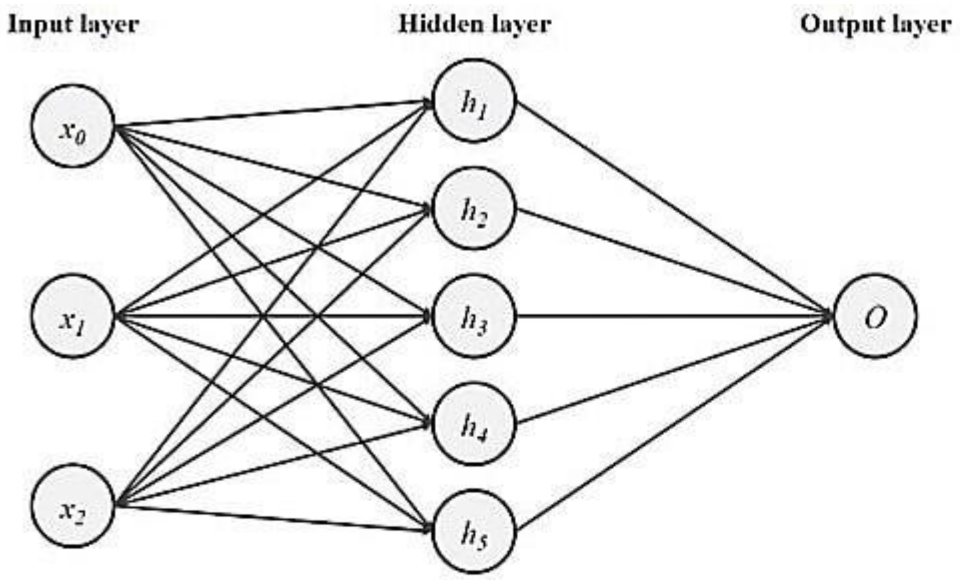

Figure 1 - A neural network

D. RNN-LSTM Network

Traditional neural network techniques function well for applications requiring prediction, but they cannot store memories. On the other hand, the Recurrent Neural Network (RNN) is a section of a neural network that has been converted into a loop, providing it the ability to retain knowledge from its previous state.

Hochreiter & Schmidhuber [7] have introduced the concept of Long-Short Term Memory (LSTM), which has proven its accuracy across various domains [7]. LSTM is a type of Recurrent Neural Network (RNN) that can learn long-term dependencies and is useful for a sequence-to-sequence prediction–such as prediction of upcoming crude oil prices using time-series data.

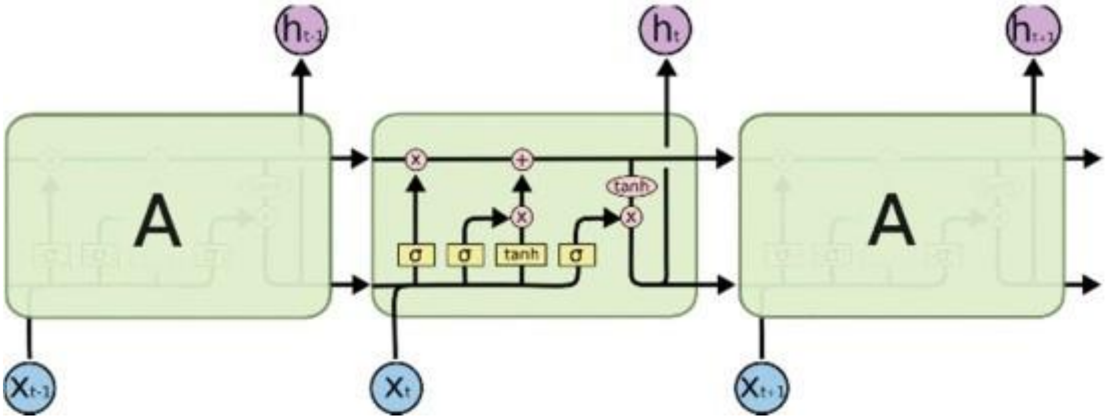

Figure 2 - The RNN-LSTM architecture

In our project “Crude Oil Price Prediction”, we proposed a solution which uses the RNN LSTM method to solve the existing problem. Time series analysis algorithm is used to combine all the advantages of the above methods and to remove some of the disadvantages discussed in the above methods.Time series analysis is a specific way of analysing a sequence of data points collected over an interval of time. In time series analysis, analysts record data points at consistent intervals over a set period of time rather than just recording the data points intermittently or randomly. This model is also trained using the Long Short Term Memory method in the Recurrent Neural Network algorithm which would have a greater efficiency.

2.2 References

1. N. Aziz, M. H. A. Abdullah and A. N. Zaidi, "Predictive Analytics for Crude Oil Price Using RNN-LSTM Neural Network," 2020 International Conference on Computational Intelligence (ICCI), 2020, pp. 173-178, doi: 10.1109/ICCI51257.2020.9247665.

2. G. Shmueli and O. R. Koppius, "Predictive analytics in information systems research", *MIS Q*, pp. 553-572, 2011.

3. H. Miao, S. Ramchander, T. Wang and D. Yang, "Influential factors in crude oil price forecasting", *Energy Econ*, vol. 68, pp. 77-88, 2017.

4. "Determining Fuel Prices", Mar 2020, [online] Available: https://www.caltex.com/my/motorists/tips-resources/determining-fuel-prices.html.

5. M. H. A. Abdullah, "Comparative Analysis of Spatio/Spectro-Temporal Data Modelling Techniques", *Data Engineering And Information Security*, 2017.

6. M. H. A. Abdullah, M. Othman, S. Kasim and S. A. Mohamed, "Evolving Spiking Neural Networks Methods for Classification Problem : A Case Study in Flood Events Risk Assessment Evolving spiking neural networks methods for classification problem : a case study in flood events risk assessment", *Indones. J. Electr. Eng. Comput. Sci*, vol. 16, no. 1, pp. 222-229, 2019.

7. S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Comput*, vol. 9, no. 8, pp.

2.3 Problem Statement Definition

The price of crude oil has a significant impact on the environment globally, and its forecasts are particularly helpful to governments and industry. Crude oil is the most widely used fuel in the world. The ongoing application of statistics and econometric methods for crude oil, including AI Price forecasting could show reductions in the accuracy of the prediction.

In order to predict future crude oil using historical data on crude oil, RNN (Recurrent Neural Network) is utilised with long short-term memory. The effectiveness of the cost is calculated using the mean squared error. Using the pricing information in the crude oil materials, the proposed model's performance is assessed.

Since changes in the price of crude oil have a significant impact on national economies around the world, price forecasting can help reduce the risks brought on by oil price volatility.

Governments, public and private businesses, legislators, and investors all place a high value on price estimates.

The project "Crude Oil Price Prediction", has the following uniqueness and novelty:

- This model is used to forecast future pricing and to manage oil use.
- This price directly influences many different items and goods, and its variations have an impact on the capital markets.
- Important events also have an impact on oil prices, in addition to economic factors.

The project "Crude Oil Price Prediction", has the following business model:

- It can assist those who are making decisions about whether to buy or sell crude oil, whether they are businesses, private investors, or individuals.
- The benchmark model for predicting crude oil prices uses RNN and LSTM models.

The scalability of the solution of this project are:

- The dimensions of the data are reduced using the PCA, MDS, and LLE methods.
- Enhance the RNN and LSTM models' accuracy.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map canvas is a more in-depth version of the original empathy map, which helps identify and describe the user's needs and pain points. And this is valuable information for improving the user experience.

Teams rely on user insights to map out what is important to their target audience, what influences them, and how they present themselves. This information is then used to create personas that help teams visualize users and empathize with them as individuals, rather than just as a vague marketing demographic or account number.

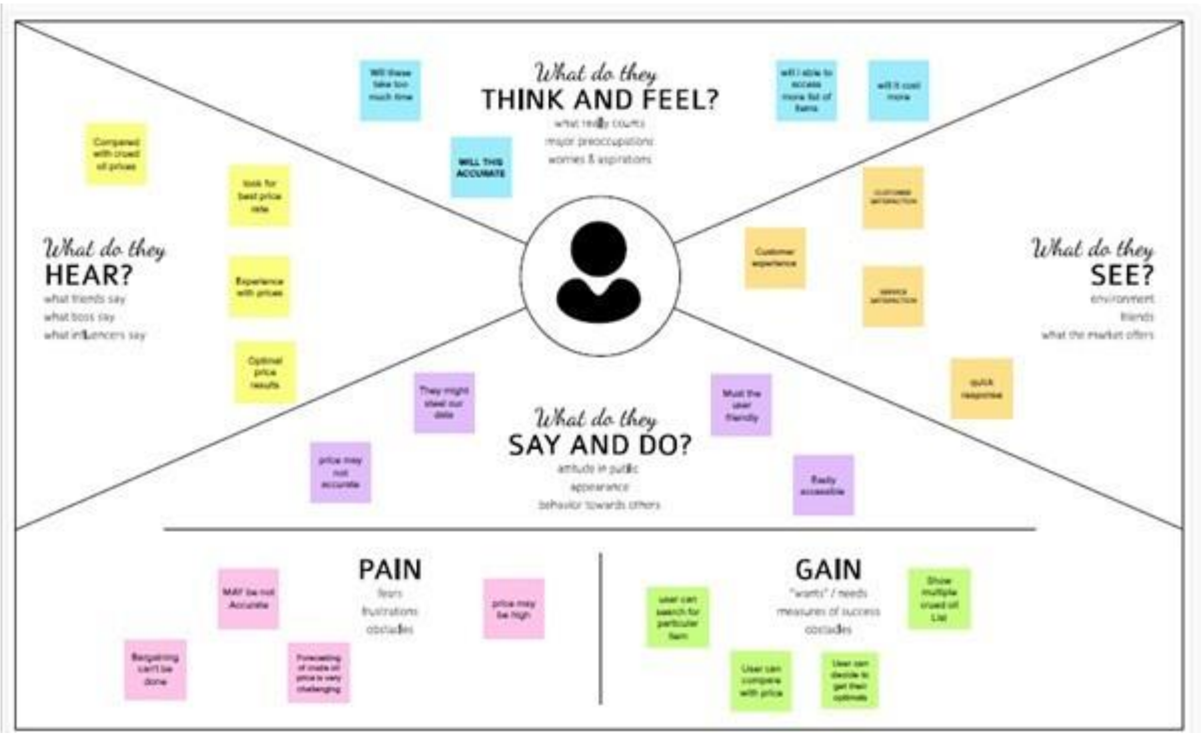

Figure 3 – Empathy Map Canvas

3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions.

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.

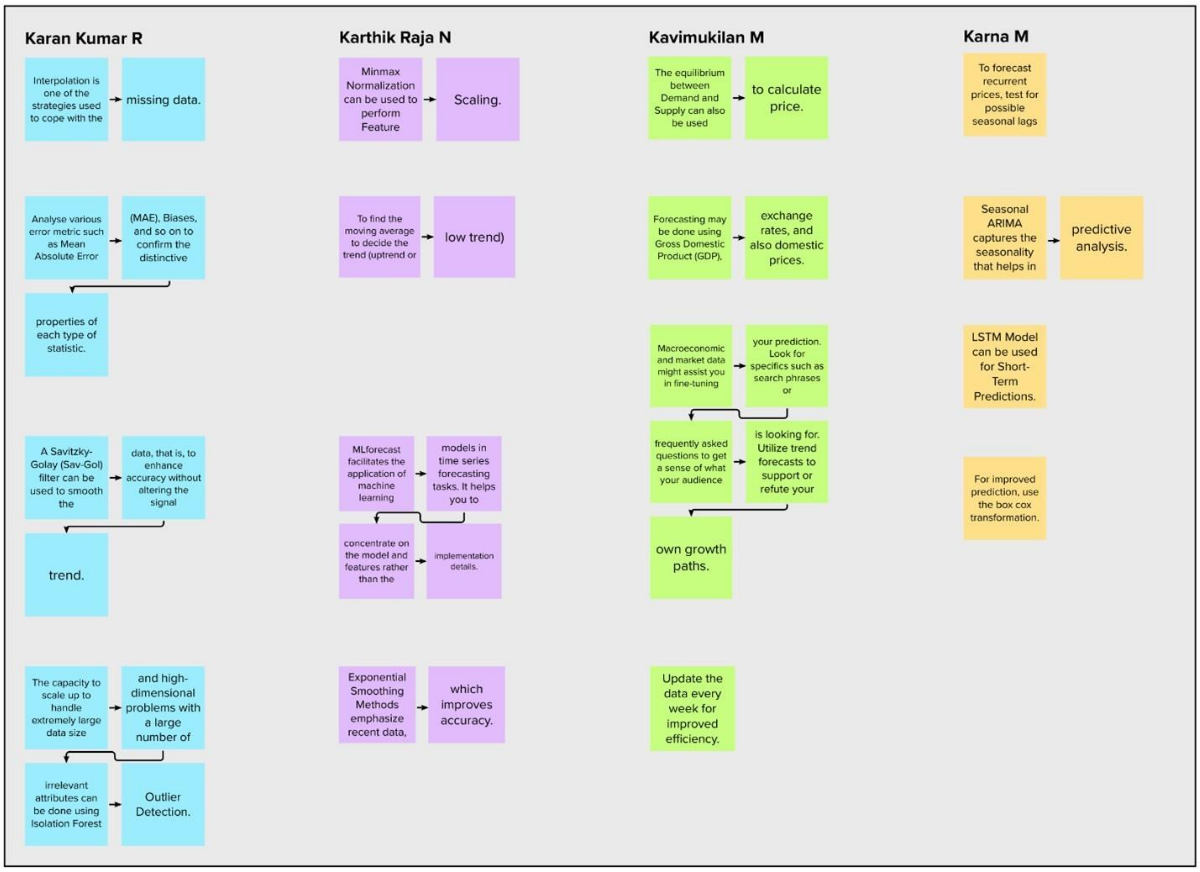

Figure 4 - Brainstorm

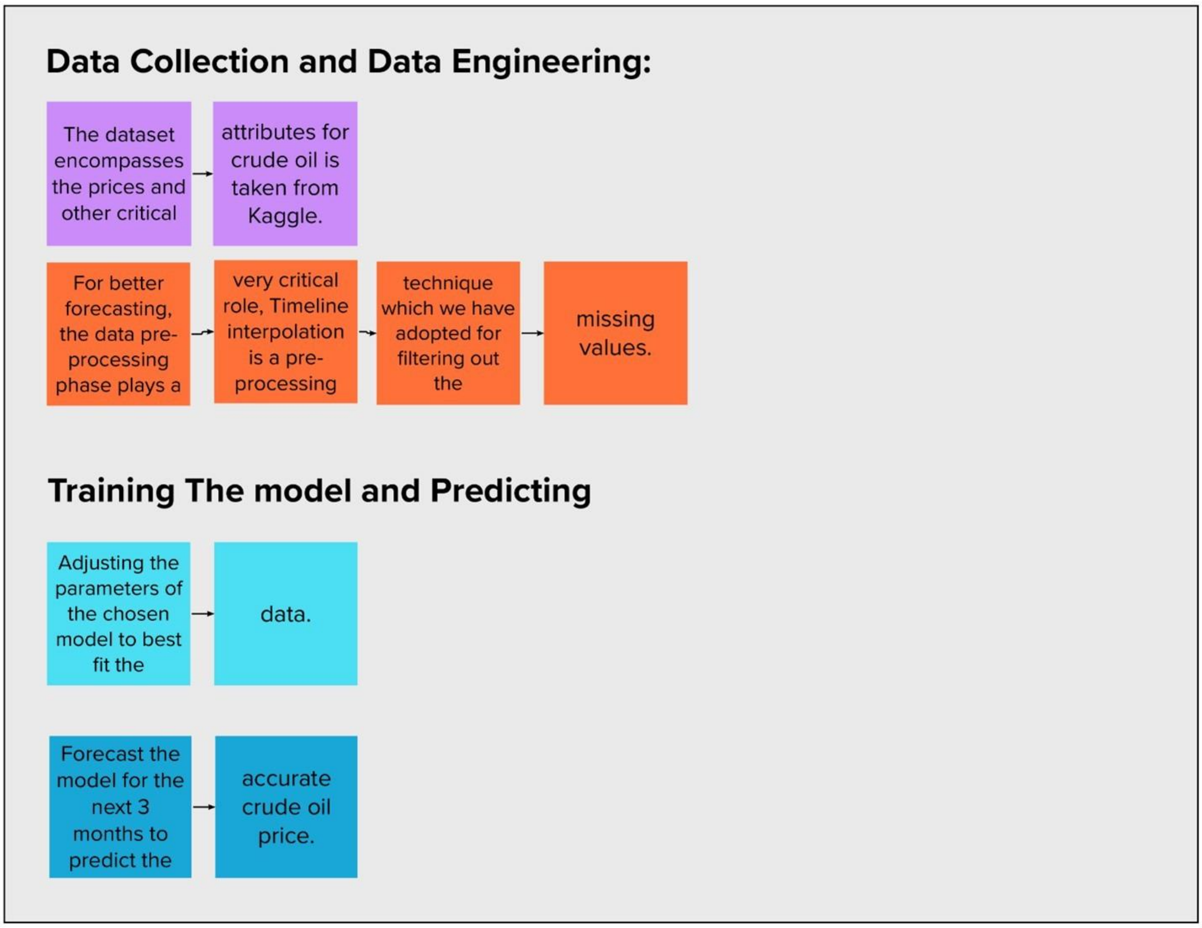

Figure 5 – Group Ideas

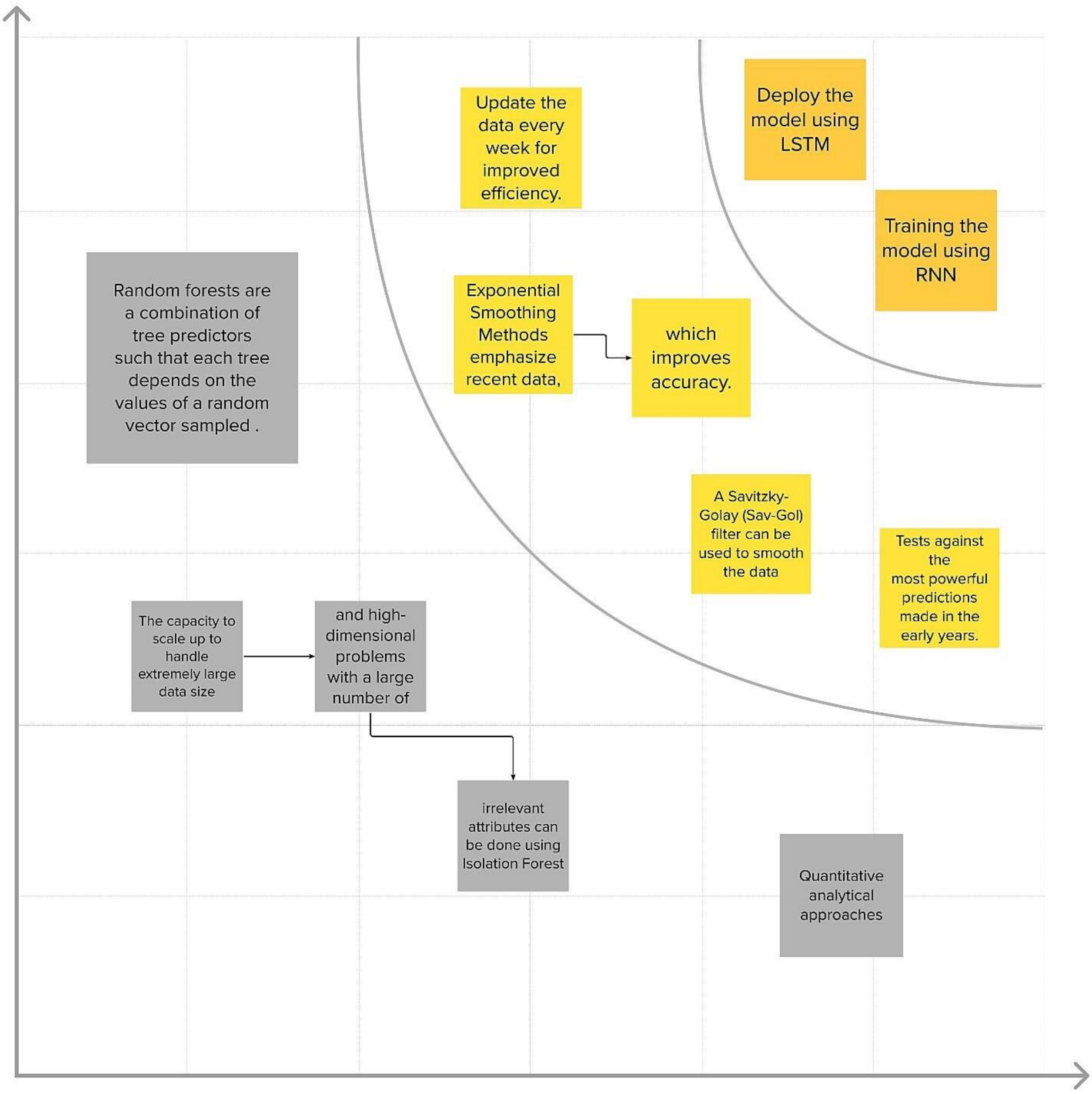

Figure 6 – Prioritisation

3.3 Proposed Solution

The price of crude oil has a significant impact on the environment globally, and its forecasts are particularly helpful to governments and industry. Crude oil is the most widely used fuel in the world. The ongoing application of statistics and econometric methods for crude oil, including AI Price forecasting could show reductions in the accuracy of the prediction.

In order to predict future crude oil using historical data on crude oil, RNN (Recurrent Neural Network) is utilised with long short-term memory. The effectiveness of the cost is calculated using the mean squared error. Using the pricing information in the crude oil materials, the proposed model's performance is assessed.

Since changes in the price of crude oil have a significant impact on national economies around the world, price forecasting can help reduce the risks brought on by oil price volatility.

Governments, public and private businesses, legislators, and investors all place a high value on price estimates.

The project "Crude Oil Price Prediction", has the following uniqueness and novelty:

- This model is used to forecast future pricing and to manage oil use.
- This price directly influences many different items and goods, and its variations have an impact on the capital markets.
- Important events also have an impact on oil prices, in addition to economic factors.

The project "Crude Oil Price Prediction", has the following business model:

- It can assist those who are making decisions about whether to buy or sell crude oil, whether they are businesses, private investors, or individuals.
- The benchmark model for predicting crude oil prices uses RNN and LSTM models.

The scalability of the solution of this project are:

- The dimensions of the data are reduced using the PCA, MDS, and LLE methods.

- Enhance the RNN and LSTM models' accuracy.

3.4 Problem Solution fit

Problem-Solution fit canvas is not just a mapping tool, but an actionable translation template, where you turn problems into solution and communication strategy, taking into account customer behaviour to increase your chances of solution adoption. It gives you insights into how your idea could fit the reality.

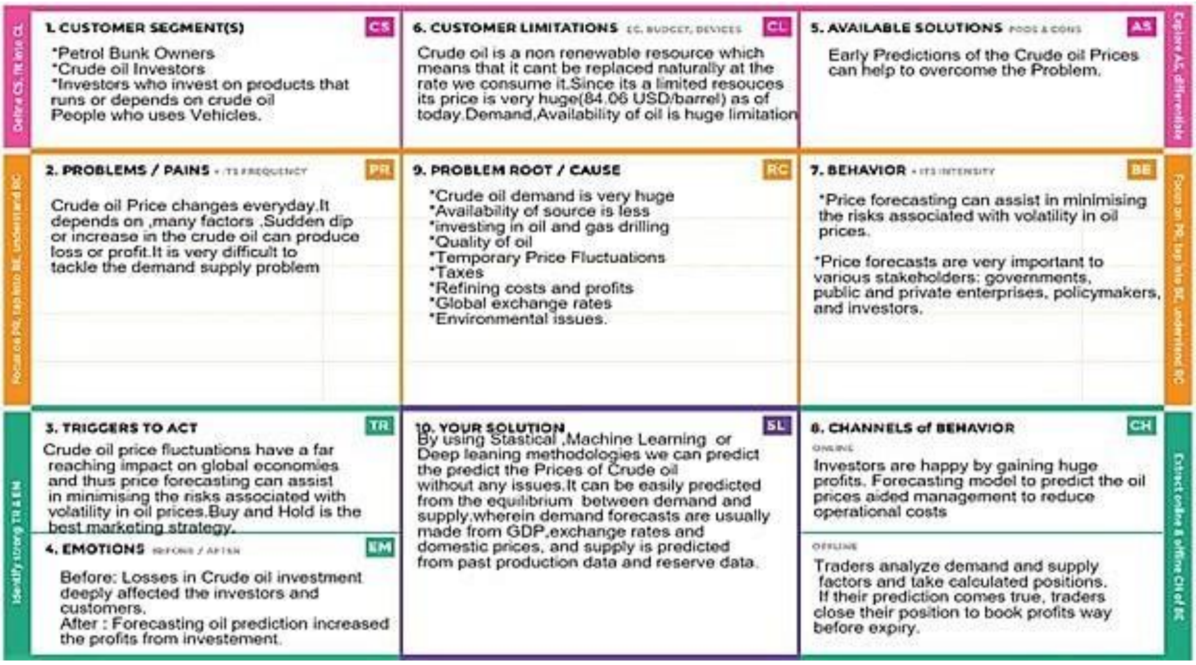

Figure 7 – Problem Solution Fit

CHAPTER 4

REQUIREMENT ALANLYSIS

CHAPTER 4

REQUIREMENT ALANLYSIS

4.1 Functional Requirement

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 • Registration through LinkedIn
FR-2	User Confirmation	• Confirmation via Email • Confirmation via OTP
FR-3	User Login	• Login through username and password • Login through Gmail • Login through LinkedIn
FR-4	Primary specifics	• Sync oil price every second • Show Up and Down graph in real time in accordance with the oil price
FR-5	Additional Requirement	• Read latest news • View price charts • Review futures on selected quotation • Analyse historical price trends • Check exchange rates and commodities futures
FR-6	System Responsibility	• Allowing the user to select a date • Track the precious results • The pricing news should be updated

Table 1 – Functional Requirements

4.2 Non-Functional Requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• To utilise a system easily and accelerate routine operations, it must have a logical user interface. • Anyone who registers on the portal can utilise the system.
NFR-2	**Security**	The following is a list of some of the factors that have been found to prevent malicious or unintentional access, usage, modification, destruction, or disclosure of the software: • Maintain particular log or historical data sets. • Apply specific cryptography methods. • Limit the number of devices that can access the website for predicting the price. • Verify the integrity of the data.
NFR-3	**Reliability**	• At the time of entry, all user variable data will be committed to the database. • By using the available backup procedures and techniques, data corruption is avoided.

NFR-4	**Performance**	● The system must allow for the simultaneous use of many users at all times. ● The accuracy of the price should be at the maximum.
NFR-5	**Availability**	● The system should always be accessible, allowing for simple user access. ● A replacement page will be displayed in the event that hardware or data base failure increases, and data should be obtained to restore the system.
NFR-6	**Scalability**	● Identifies the maximum workloads at which the system will still operate well. ● Focus on the measurement of the system's response time under various load levels.

Table 2 – Non-Functional Requirements

CHAPTER 5

PROJECT DESIGN

CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

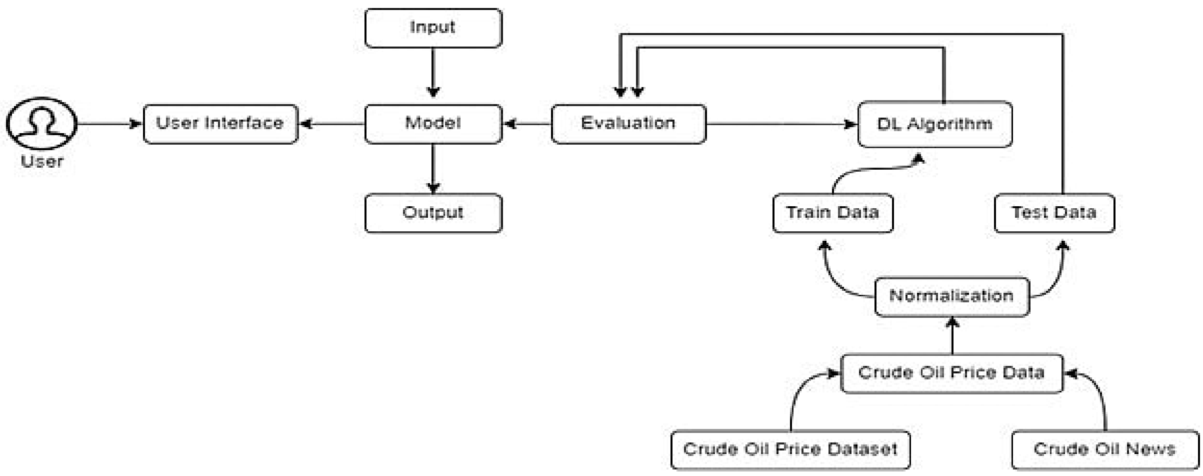

Figure 8 – Data Flow Diagram

5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.

- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:

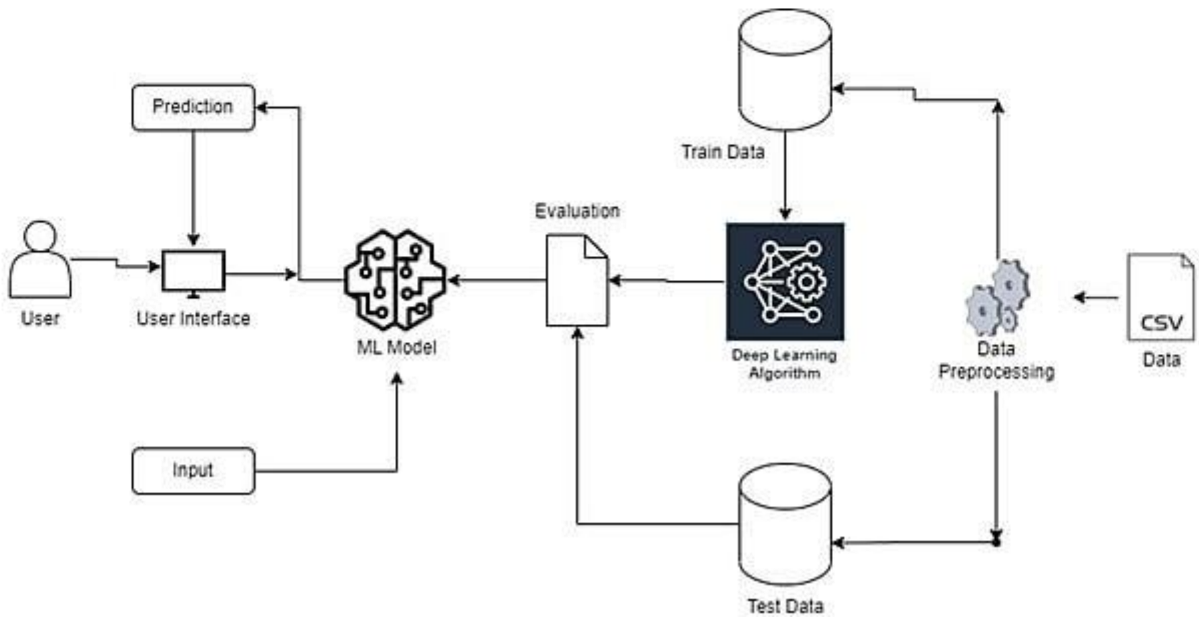

Figure 9 - Architecture Diagram

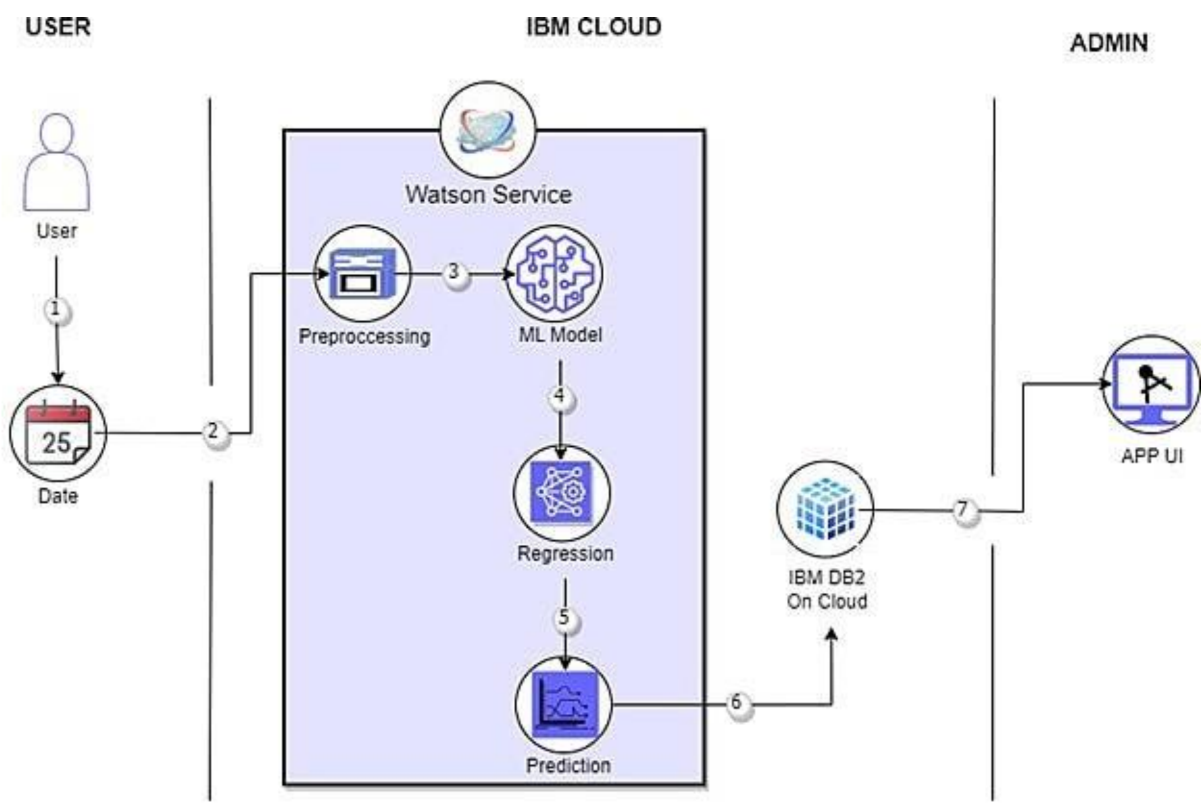

Figure 10 – Technical Architecture Diagram

Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	Through a web UI, the user can engage with the application.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	It has many in built libraries which helps in machine learning	Python
3.	Application Logic-2	It helps to build machine learning model	IBM Watson Jupyter Notebook service
4.	Application Logic-3	It is fast and accurate	IBM Watson Assistant
5.	Database	MySQL is used to store the user information and warehouse the crude oil price	MySQL
6.	Cloud Database	IBM Db2 is reliable and scalable	IBM DB2
7.	File Storage	Maintain files easily	Local Filesystem
8.	External API-2	Aadhar and customer KYC verification takes a little amount of time	Aadhar API, etc.
9.	Machine Learning Model	To recognise the patterns and trends	Sequential, Dense & LSTM Model

10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration	Local System and IBM Watson

Table 3 – Components & Technologies

Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Tensor flow – Implements model building and training. Flask – Can handle multiple user request simultaneously. Scikit learn – Contains model for classification, regression, clustering.	Tensor flow, Flask, Scikit learn.
2.	Security Implementations	SHA-256 doesn't have any known vulnerabilities	SHA-256.
3.	Scalable Architecture	MySQL can store huge amount of data and it is easily scalable.	MySQL
4.	Availability	This application can be accessed from anywhere easily and it is easily scalable.	IBM Watson Cloud.

5.	Performance	Flask can handle multiple user request simultaneously.	Flask

Table 4 – Application Characteristics

5.3 User Stories

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the	I can receive confirmation email & click confirm	High	Sprint-1

			application			
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register through already existing mail account.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	After registration, I can log in via only email & password.	High	Sprint-1

	Dashboard	USN-6	Display the oil price, line graph / bar graph real time.	I can expect the prediction in various formats.	Low	Sprint-3
Customer (Web user)	Login	USN-7	As the user, I can login by using Gmail or Facebook account or LinkedIn or by registering.	Existing users can easily login.	High	Sprint-2
Customer Care Executive	Support	USN-8	The Customer care service will provide solutions for any FAQ and also provide Chat-Bot.	I can solve the problems raised.	High	Sprint-3
Administrator	Access Control	USN-9	Admin can control the access of users.	Access permission for Users.	High	Sprint-4

	Database	USN-10	Admin can store the details of users.	Stores User details.	Medium	Sprint-4
	News	USN-11	Admin will give the recent news of Oil Prices.	Provide the recent oil prices.	Medium	Sprint-4
	Notification	USN-12	Admin will notify when the oil prices changes.	Notification by Gmail.	High	Sprint-4

Table 5 – User Stories

CHAPTER 6

PROJECT PLANNING & SCHEDULING

CHAPTER 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	Download Crude Oil Price Dataset	2	Medium	Karna M
Sprint-1	Data Preprocessing	USN-2	Importing The Dataset into Workspace	1	Low	Kavimukilan M
Sprint-1		USN-3	Handling Missing Data	3	Medium	Karthikraja N
Sprint-1		USN-4	Feature Scaling	3	Low	Karan Kumar R
Sprint-1		USN-5	Data Visualization	3	Medium	Karna M
Sprint-1		USN-6	Splitting Data into Train and Test	4	High	Kavimukilan M
Sprint-1		USN-7	Creating A Dataset with Sliding Windows	4	High	Karthikraja N

Sprint-2	Model Building	USN-8	Importing The Model Building Libraries	1	Medium	Karna M
Sprint-2		USN-9	Initializing The Model	1	Medium	Karthikraja N
Sprint-2		USN-10	Adding LSTM Layers	2	High	Kavimukilan M
Sprint-2		USN-11	Adding Output Layers	3	Medoum	Karan Kumar R
Sprint-2		USN-12	Configure The Learning Process	4	High	Karna M
Sprint-2		USN-13	Train The Model	2	Medium	Karan Kumar R
Sprint-2		USN-14	Model Evaluation	1	Medium	Karan Kumar R
Sprint-2		USN-15	Save The Model	2	Medium	Karthikraja N
Sprint-2		USN-16	Test The Model	3	High	Karna M
Sprint-3	Application Building	USN-17	Create An HTML File	4	Medium	Kavimukilan M
Sprint-3		USN-18	Build Python Code	4	High	Karan Kumar R
Sprint-3		USN-19	Run The App in Local	4	Medium	Karna M

			Browser			
Sprint-3		USN-20	Showcasing Prediction On UI	4	High	Karan Kumar R
Sprint-4	Train The Model On IBM	USN-21	Register For IBM Cloud	4	Medium	Kavimukilan M
Sprint-4		USN-22	Train The ML Model On IBM	8	High	Karthikraja N
Sprint-4		USN-23	Integrate Flask with Scoring End Point	8	High	Karan Kumar R

Table 6 – Sprint Plan

6.2 Sprint Delivery Schedule

Title	**Description**	**Date**
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	19 September 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	23 September 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility &	25 September 2022

	importance.	
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	27 September 2022
Problem Solution Fit	Prepare problem - solution fit document.	29 September 2022
Solution Architecture	Prepare solution architecture document.	01 October 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	04 October 2022
Functional Requirement	Prepare the functional requirement document.	06 October 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	08 October 2022
Technology Architecture	Prepare the technology architecture diagram.	11 October 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	23 October 2022
Sprint Schedule	Prepare spring plan	23 October 2022
Delivery of Sprint-1	Develop & submit the developed code.	29 October 2022
Delivery of Sprint-2	Develop & submit the developed code.	05 November 2022
Delivery of Sprint-3	Develop & submit the developed code.	12 November 2022
Delivery of Sprint-4	Develop & submit the developed code.	17 November 2022

Table 7 – Sprint Plan 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	20	6 Days	31 Oct 2022	05 Nov 2022	20	03 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	10 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	17 Nov 2022

Table 8 – Sprint Delivery Schedule

6.3 Reports From JIRA:

Velocity:

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

Figure 11 – Velocity Chart

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

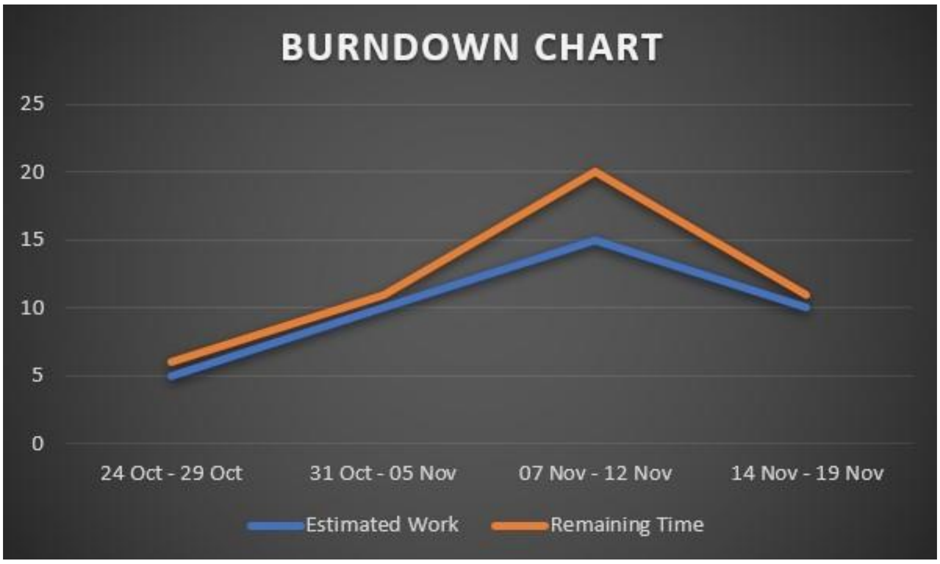

Figure 12 – Burndown Chart

CHAPTER 7

CODING & SOLUTIONING

CHAPTER 7

CODING & SOLUTIONING

7.1 Interactive UI

The area where interactions between people and machines take place is known as a user interface (UI) in the subject of industrial design known as human-computer interaction. This interaction's purpose is to enable efficient machine operation and control from the human end, while the machine also feeds information back to the operators to support their decision-making. The general objective of user interface design is to provide an interface that makes it simple, effective, and pleasurable (user-friendly) to operate a machine in a way that yields the desired outcome (i.e., maximum usability). This typically means that the machine reduces undesirable outputs to the user while simultaneously requiring the operator to input as little as possible to produce the desired output.

We have included a user interface in our project to make it easier for users to forecast the price of crude oil in the future. Users simply need to visit the website to access the interface and can click a button to forecast the price. Once the button has been clicked, the user will be taken to another website where they can enter the price of crude oil for 10 days. In that case, the user should click Predict. The user can then view the price of crude oil after ten days.

7.2 Cloud Integration

The on-demand availability of computer system resources, in particular data storage (cloud storage) and processing power, without direct active supervision by the user, is known as cloud computing. Functions in large clouds are frequently dispersed over several sites, each of which is a data centre. Cloud computing often uses a "pay as you go" model, which can help reduce capital expenses but may also result in unanticipated running expenses for users. Cloud computing depends on resource sharing to accomplish coherence.

Our project is cloud-integrated, allowing it to run anywhere and be accessible at any time. Anytime the user desires, they will be able to forecast the price of crude oil. Through the IBM Cloud, this is accomplished. On the IBM Watson Studio, which makes use of the Watson Machine Learning Platform, we developed and trained the model. We generated a deployment space and ran the code using the API key to deploy the model. The Flask app, which is used to link to the backend and frontend, was then finally integrated.

CHAPTER 8

TESTING

CHAPTER 8

TESTING

8.1 Test Cases

The following test scenarios were tested successfully.

	Test Scenarios
1	Verify the UI elements in the home page
2	Verify whether the user can navigate to the prediction page
3	Verify the UI elements in the prediction page
4	Verify user is able to enter value in the text box.
5	Verify user is able to enter numbers in the text box
6	Verify model can handle with no inputs
7	Verify model can handle multiple input
8	Verify model can handle unsupported input
9	Verify model can predict the output
10	Verify the predicted results are displayed
11	Verify user can enter the value after prediction

Date	15-Nov-22
Team ID	PNT2022TMID21644
Project Name	Project - Crude Oil Price Predictio
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	BUG ID	Executed By
COPP_TC_001	UI	Index.html	Verify the UI elements in the home page	1. Enter URL 2. Check if all elements are displayed	http://127.0.0.1:5000/	The UI elements should be displayed properly	Working as expected	Pass		Karan Kumar R
COPP_TC_002	Functional	Index.html	Verify whether the user can navigate to the prediction page	1. Enter URL 2. Check whether the user can navigate to the prediction page after clicking	http://127.0.0.1:5000/	The user should be able to navigate to the prediction page after clicking the predict button	Working as expected	Pass		Karthik Raja N
COPP_TC_001	UI	Web.html	Verify the UI elements in the prediction page	1. Enter URL 2. Click the predict button 3. Check if all elements are displayed	http://127.0.0.1:5000/	The UI elements should be displayed properly	Working as expected	Pass		Kavikukilan M
COPP_TC_003	Functional	Web.html	Verify user is able to enter value in the text box	1. Enter URL 2. Click the predict button 3. Checck whether the user can enter values in the text	http://127.0.0.1:5000/	User should be able to enter the values in the text box	Working as expected	Pass		Karna M
COPP_TC_004	Functional	Web.html	Verify user is able to enter numbers in the text box	1. Enter URL 2. Click the predict button 3. Checck whether the user can enter values in the text	http://127.0.0.1:5000/	User should be able to enter numbers in the text box	Working as expected	Pass		Karan Kumar R
COPP_TC_005	Functional	Model	Verify model can handle with no inputs	1. Enter URL 2. Click the predict button 3. Checck whether the user can enter values in the text box	http://127.0.0.1:5000/	The predicted output should be displayed	Exception is thrown	Fail	COPP_B_001	Karthik Raja N
COPP_TC_006	Functional	Model	Verify model can handle multiple input	1. Enter URL 2. Click the predict button 3. Checck whether the user can enter values in the text box	http://127.0.0.1:5000/	The model should predict the output for the input data	Working as expected	Pass		Kavikukilan M
COPP_TC_007	Functional	Model	Verify model can handle unsupported input	1. Enter URL 2. Click the predict button 3. Checck whether the user can enter values in the text box	http://127.0.0.1:5000/	The predicted output should be displayed	Exception is thrown	Fail	COPP_B_002	Karna M
COPP_TC_008	Functional	Model	Verify model can predict the outpu	1. Enter URL 2. Click the predict button 3. Checck whether the user can enter values in the text box	http://127.0.0.1:5000/	The model should predict the output for the input data	Working as expected	Pass		Karan Kumar R
COPP_TC_009	Functional	Web.html	Verify the predicted results are displayed	1. Enter URL 2. Click the predict button 3. Checck whether the user can enter values in the text box	http://127.0.0.1:5000/	The predicted output should be displayed	Working as expected	Pass		Karthik Raja N
COPP_TC_003	Functional	Web.html	Verify user can enter the value after prediction	1. Enter URL 2. Click the predict button 3. Checck whether the user can enter values in the text	http://127.0.0.1:5000/	User should be able to enter the values in the text box	Working as expected	Pass		Kavikukilan M

Figure 13 – Test Cases

8.2 User Acceptance Testing:

Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	0	1	0
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	0	0
Skipped	1	0	0	0	1
Won't Fix	1	0	1	1	3
Totals	7	1	3	3	12

Table 9 – Defect Analysis

Test Case Analysis

Section	Total Cases	Not Tested	Fa il	Pass
Print Engine	10	0	2	8
Client Application	5	0	0	5
Security	1	0	0	1
Outsource Shipping	3	0	0	3
Exception Reporting	2	0	2	0
Final Report Output	4	0	0	4

Table 10 – Test Case Analysis

CHAPTER 9

RESULTS

CHAPTER 9

RESULTS

9.1 Performance Metrics:

We attempted to forecast the output of the crude oil by entering various input variables in order to assess the accuracy and performance of this project. These are the input values.

[0.44172960165852215, 0.48111950244335855, 0.49726047682511476, 0.4679401747371539, 0.4729749740855915, 0.47119798608026064, 0.47341922108692425, 0.4649785280616022, 0.4703835332444839, 0.47149415074781587]

The anticipated outcome after providing the input values is 0.46976325.

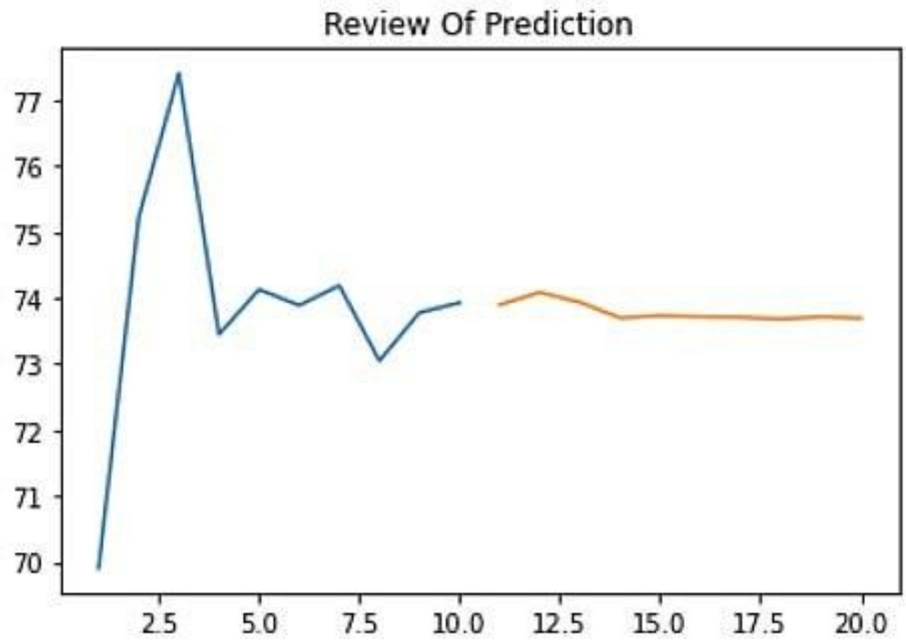

Figure 14 – Review of Prediction

Figure 13 gives a review of prediction how the system has predicted the future price based on the given input values.

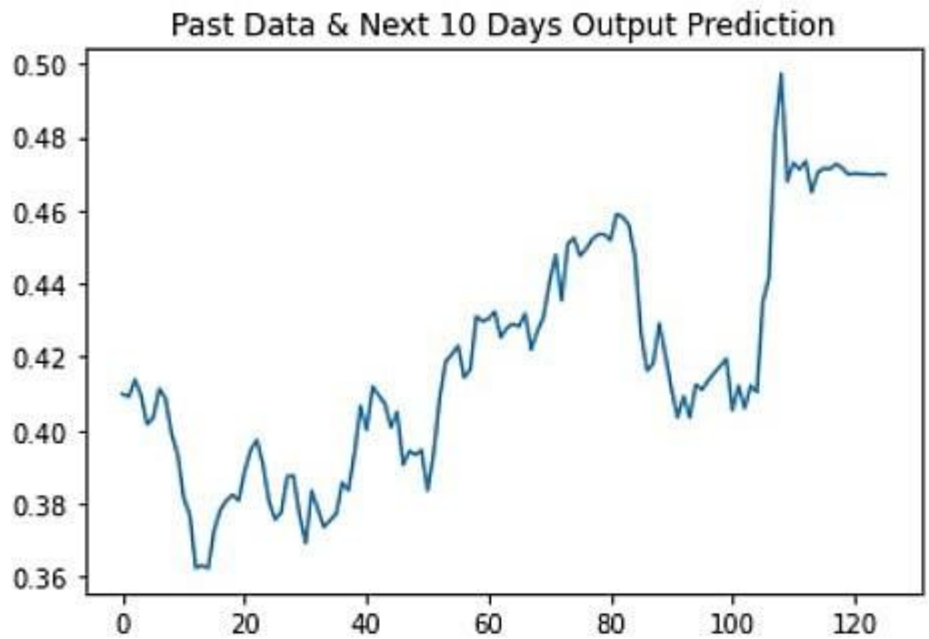

Figure 15 – Next 10 Days Prediction

It can be seen that the graph was drawn using the provided data and a projection for the next 10 days. There was a little discrepancy between the output and the real pricing.

The developed system shows a clear prediction of the future prices which has very less deviations from the true prices by using LSTM in tensorflow and keras in python. There is always a thin line between the overfitting of the model and its best performance. This project helps a lot to learn about the developed model and the algorithm and using this model as a base, a much more complicated model can be easily developed. The facet of more prediction algorithms for crude oil can concoct with the help of this system.

This system concludes that the machine learning model LSTM (Long Short-Term Method) predicts the future price of crude oil by bordering the actual price of the crude oil price.

CHAPTER 10

ADVANTAGES & DISADVANTAGES

CHAPTER 10

ADVANTAGES & DISADVANTAGES

Advantages

- High Accuracy
- Removes the investment bias
- Develop the habit of complete analysis
- Minimise our losses
- Allows smart way of making money

High Accuracy:

The model which we predicted had a high accuracy of above 90 per cent in all aspects. The other advantages of predicting the price of crude oil are discussed below.

Removes the investment bias:

The Indian stock market offers a variety of chances for traders and investors, but it is also helpful to be aware of the market environment before taking a position in a particular stock. Take the weather prediction as an example to help you comprehend this; being aware of the weather forecast for the coming week enables you to make appropriate plans. The situation with stock market investments is comparable. Let's look at a few of the major benefits connected with stock market prediction now to help you grasp.

Develop the habit of complete analysis:

Investors don't always conduct a thorough research of the stock before learning how to anticipate the stock market and putting what they have learned into practise. They only start to establish the habit of comprehensive analysis before making any investing decisions after they learn how to apply formulae and procedures to forecast stock market movements. Once or initially, making a successful stock market prediction gives investors the confidence to form the habit of conducting a thorough analysis each time. Here, "complete analysis" refers to both the fundamental and the technical analysis of the stocks because the combination of these two forecasting methods results in predictions that are more precise.

Minimise our losses:

Another benefit of stock market prediction is that it significantly reduces your losses or restricts them. Investors sometimes make the error of not doing their studies thoroughly before learning how to anticipate, which results in them frequently employing the incorrect prediction strategies. As a result, many put their money into the stocks based solely on intuition or merely wild estimates in the hopes that the prices will rise, and they will profit. They lose most of the time because it doesn't happen. They can reduce their losses by correctly implementing and using the appropriate forecast strategies. The converse of this is also true, and given the information provided, you can make wise selections.

Allows smart way of making money:

Making steadily increasing profits through the use of your trading expertise and knowledge is the smart method to make money. The most desired and ideal approach to make money in the stock market is to become a day trader and make money every day, unless of course a person has long-term aspirations. But in order to do that, you must be aware of the various difficulties and difficulties that come with intraday trading, as well as how to deal with them. That can only occur when you understand how to forecast the stock market using a variety of tools and tactics and how to maximise intraday trading, enabling yourself to consistently make money.

Disadvantages

- Forecasts are never 100% accurate
- It can be time-consuming and resource-intensive

Forecasts are never 100% accurate:

Let's face it: it's hard to predict the future. Even if you have a great process in place and forecasting experts on your payroll, your forecasts will never be spot on. Some products and markets simply have a high level of volatility. And in general, there is just an endless number of factors that influence demand.

It can be time-consuming and resource-intensive:

Forecasting involves a lot of data gathering, data organizing, and coordination. Companies typically employ a team of demand planners who are responsible for coming up with the forecast. But in order to do this well, demand planners need substantial input from the sales and marketing teams. In addition, it's not uncommon for processes to be manual and labour-intensive, thus taking up a lot of time. Fortunately, if you have the right technology in place, this is much less of an issue.

CHAPTER 11

CONCLUSION

CHAPTER 11

CONCLUSION

In today's world and in such a dynamic atmosphere where everyone wants to know what will happen in the future, artificial intelligence and deep learning are the foundation for upgrading technology. The path to future prediction has been established by several facilities. It previously hard to predict the prices of cryptocurrencies since they change randomly, but machine learning has made it feasible.

By integrating LSTM in TensorFlow and keras in Python, the constructed model demonstrates a clear prediction of the future prices with very little variance from the genuine prices. Between the model being overfitted and performing at its optimum, there is always a fine line. With a few minor adjustments, the model may be applied to different time series data. With the knowledge gained from this research, a far more complex model may be created with relative ease utilising the generated model and algorithm as a foundation. With the aid of this model, more prediction algorithms for bitcoin may be developed.

This project comes to the conclusion that the LSTM (Long Short-Term Method) machine learning algorithm predicts the future price of crude oil by edging the current price of the oil with high accuracy.

CHAPTER 12

FUTURE SCOPE

CHAPTER 12

FUTURE SCOPE

The Long Short-Term Method (LSTM) machine learning algorithm is shown to have a high degree of accuracy in predicting the future price of crude oil by edging the current price of the oil.

In the future, it will be possible to estimate crude oil prices by taking into account additional variables that influence the price, such as tweets, national news, natural disasters, the cost of forecasting, conflict, demand, and floods. By doing this, the model's precision and accuracy would both be enhanced.

The dataset will be obtained from Kaggle, a sizable platform that is frequently used for data mining and doing analysis. The model would similarly be created using these elements. If this is carried out, the accuracy of forecasting the price of crude oil will exceed 98 percent.

CHAPTER 13

APPENDIX

CHAPTER 13

APPENDIX

Source Code

Building the model:

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

data = pd.read_excel("Crude Oil Prices Daily.xlsx")
data.head()

data.isnull().any()

data.isnull().sum()

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

data_oil = data.reset_index()["Closing Value"]
data_oil

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

plt.title('Crude OIl Price')
plt.plot(data_oil)
```

```
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

import numpy
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, ytest = create_dataset(test_data, time_step)

print(X_train.shape), print(y_train.shape)

print(X_test.shape), print(ytest.shape)

X_train

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)

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

model = Sequential()

model.add(LSTM(50,return_sequences = True, input_shape = (10,1)))
model.add(LSTM(50,return_sequences = True))
model.add(LSTM(50))

model.add(Dense(1))
model.summary()

model.compile(loss='mean_squared_error', optimizer = 'adam')

model.fit(X_train, y_train, validation_data = (X_test, ytest), epochs = 10, batch_size =
64, verbose = 1)

train_predict=model.predict(X_train)
test_predict=model.predict(X_test)

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

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

```
from tensorflow.keras.models import load_model
model.save("Crude_oil.h5")

look_back = 0
trainPredictPlot = np.empty_like(data_oil)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(train_predict) + look_back, :] = train_predict

testPredictPlot = np.empty_like(data_oil)
testPredictPlot[:,:] = np.nan
testPredictPlot[len(train_predict)+(look_back*2)+1: len(data_oil)-1, :] = test_predict

plt.plot(scaler.inverse_transform(data_oil))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.title("Testing The Model")
plt.show()

len(test_data)

x_input = test_data[2866:].reshape(1,-1)
x_input.shape

temp_input = list(x_input)
temp_input = temp_input[0].tolist()
temp_input

lst_output = []
n_steps = 10
i=0
```

```
while(i<10):
    if(len(temp_input)>10):
        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))

        yhat = model.predict(x_input, verbose = 0)
        print("{} day output {}".format(i,yhat))
        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)
        print(yhat[0])
        temp_input.extend(yhat[0].tolist())
        print(len(temp_input))
        lst_output.extend(yhat.tolist())
        i=i+1

day_new = np.arange(1,11)
day_pred = np.arange(11,21)

len(data_oil)

plt.plot(day_new,scaler.inverse_transform(data_oil[8206:]))
plt.title("Review Of Prediction")
```

```
plt.plot(day_pred,scaler.inverse_transform(lst_output))
plt.show()

df3 = data_oil.tolist()
df3.extend(lst_output)
plt.title("Past Data & Next 10 Days Output Prediction")
plt.plot(df3[8100:])

df3 = scaler.inverse_transform(df3).tolist()
plt.title("Past Data & Next 10 Days Output Prediction After Reversing The Scaled Values")
plt.plot(df3)
```

Deploying on IBM Cloud:

```
get_ipython().system('pip install ibm_watson_machine_learning')

from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "uVEty-CB4dYcccQ_Jq9V-atVXmL1dByE_wiDm95lcyTQ"
}

client = APIClient(wml_credentials)

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

space_uid = guid_from_space_name(client, 'NewSpace')
print("Space UID = " + space_uid)

client.set.default_space(space_uid)
```

```
client.software_specifications.list()

software_spec_id =
client.software_specifications.get_id_by_name('tensorflow_rt22.1-py3.9')
print(software_spec_id)

model.save('crude.h5')

get_ipython().system('tar -zcvf crude-oil.tgz Crude.h5')

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

model_details = client.repository.store_model(model='crude.tgz',meta_props={
client.repository.ModelMetaNames.NAME:"crude_oil_model",
client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_id }
                                    )
model_id = client.repository.get_model_uid(model_details)
model_id

client.repository.download(model_id,'crude_oil_model.tar.gb')
```

INTEGRATE FLASK WITH SCORING END POINT

App.py

```
from flask import Flask,render_template,request,redirect
import pandas as pd
import numpy as np
from flask import Flask, render_template, Response, request
import pickle
from sklearn.preprocessing import LabelEncoder
import requests
```

```
# NOTE: you must manually set API_KEY below using information retrieved from
your IBM Cloud account.
API_KEY = "uVEty-CB4dYcccQ_Jq9V-atVXmL1dByE_wiDm95lcyTQ"
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_)

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


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

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

        lst_output = []
        n_steps = 10
        i=0
        while(i<10):
            if(len(temp_input)>10):
                x_input = np.array(temp_input[1:])
                x_input = x_input.reshape(1,-1)
                x_input = x_input.reshape((1,n_steps, 1))
                yhat = model.predict(x_input, verbose = 0)
                temp_input.extend(yhat[0].tolist())
                temp_input = temp_input[1:]
```

```
            lst_output.extend(yhat.tolist())
            i=i+1

        else:
            x_input = x_input.reshape((1, n_steps,1))
            yhat = model.predict(x_input, verbose = 0)
            temp_input.extend(yhat[0].tolist())
            lst_output.extend(yhat.tolist())
            i=i+1


        # NOTE: manually define and pass the array(s) of values to be scored in the
next line
        payload_scoring = {"input_data": [{   "values": [[x_input]]    }]}

        response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/7f67cbed-6222-413b-9901-
b2a72807ac82/predictions?version=2022-10-30', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
        predictions = response_scoring.json()
        print(response_scoring.json())

        val = lst_output[9]
        return render_template('web.html' , prediction = val)



    if request.method=="GET":
        return render_template('web.html')

if name ==" main ":
    model = load_model('C:/Users/rkara/IBM/Sprint - 4/Crude_oil.tar.gz')
    app.run(debug=True)
```

INDEX.HTML

```
<!DOCTYPE html>
<head>
   <title>Crude Oil Price Prediction </title>
   <link rel="stylesheet" href="{{ url_for('static', filename='css/index.css') }}">
</head>
<body>
   <h1> Crude Oil Price Prediction</h1>
   <p> Demand for oil is inelastic, therefore the rise in price is good news
    for producers because they will see an increase in their revenue. Oil
    importers, however, will experience increased costs of purchasing oil.
    Because oil is the largest traded commodity, the effects are quite
    significant. A rising oil price can even shift economic/political
    power from oil importers to oil exporters. The crude oil price movements
    are subject to diverse influencing factors.
   </p><br><br>
   <a href="{{url_for('predict')}}">
   Predict Future Price</a>
</body>
```

WEB.HTML

```
<!DOCTYPE html>
<head>
   <title>Crude Oil Price Prediction </title>
   <link rel="stylesheet" href="{{ url_for('static', filename='css/web.css') }}">
</head>
<body>
   <h1>
```

```
    Crude Oil Price Prediction </h1>
    <form action="/predict" method="POST" enctype = "multipart/form-data">
        <input type="text" name="val" placeholder="Enter the crude oil price for first 10 days" >
         <br> <br> <br>
        <input type="submit"/>
    </form><br> <br>
    <div>
        {{prediction}}
    </div>

</body>
```

GitHub Link:

The "Crude Oil Price Prediction" project has been posted on GitHub.

Link: https://github.com/IBM-EPBL/IBM-Project-12033-1659367379

Project Demonstration:

The following link will take you to a demonstration of the "Crude Oil Price Prediction" project.

Link: Demo Link