Crude oil price prediction

Team ID	PNT2022TMID48733
Project name	Crude oil price prediction
Team members name	R.Muniyappan K.Murugesan K.Karuppasami T.Iyappan

1. INTRODUCTION

1.1. Project Overview

Artificial intelligent methods are being extensively used for oil price forecasting as an alternate approach to conventional techniques. There has been a whole spectrum of artificial intelligent techniques to overcome the difficulties of complexity and irregularity in oil price series. The potential of AI as a design tool for oil price forecasting has been reviewed in this study.

1.2. Purpose

Increases in oil prices can depress the supply of other goods because they increase the costs of producing them. In economics terminology, high oil prices can shift up the supply curve for the goods and services for which oil is an input.

2. LITERATURE SURVEY

2.1. Existing problem

The price of oil fluctuates according to three main factors: current supply, future supply, and expected global demand. Members of OPEC control 40% of the world's oil.

As a result, these will be passed down to the consumers, increasing the overall prices of goods and services, and causing inflation. In fact, Bloomberg predicts that a “10% rise in oil prices could add a 0.4 percentage points” to Philippine inflation.

2.2. Reference

Barsky RB, Kilian L (2001) Do we really know that oil caused the great stagflation? A monetary alternative. NBER Macroecon Annu 16:137–183. https://doi.org/10.1086/654439

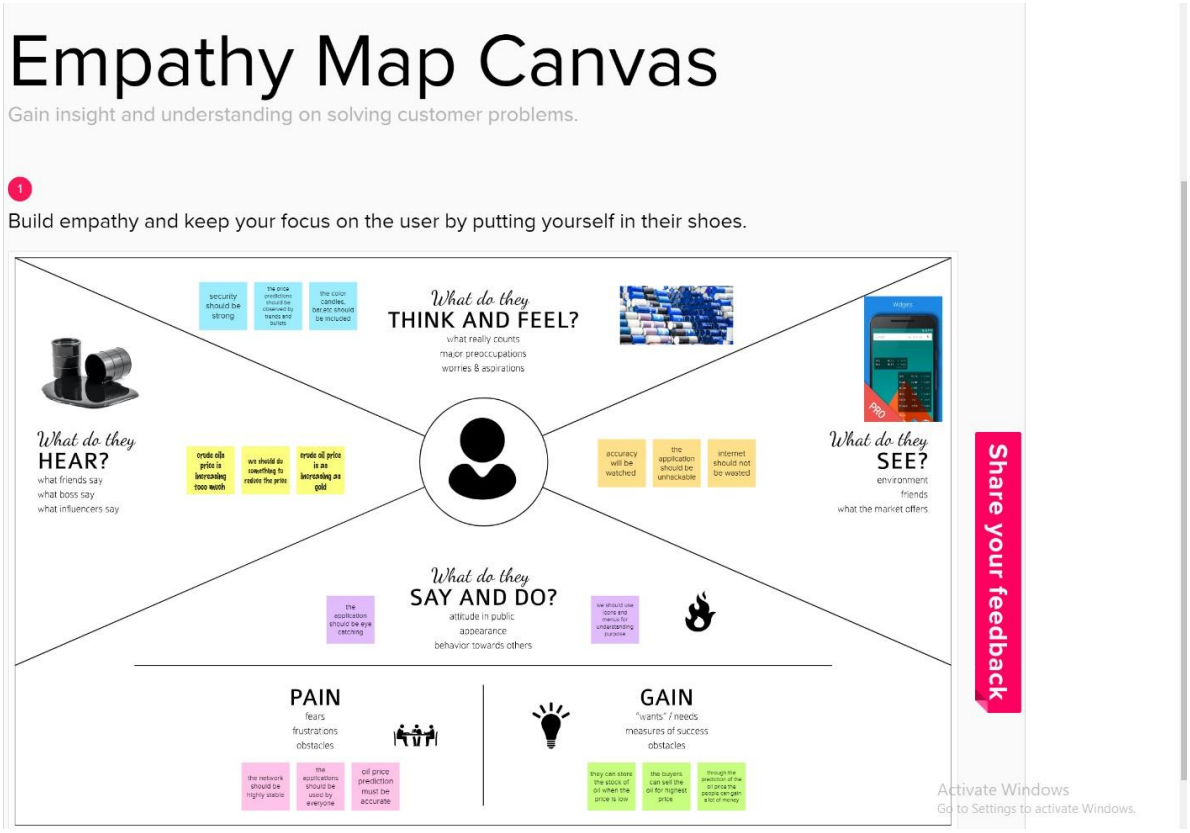
Castle JL, Qin X, Reed WR (2009) How to pick the best regression equation: a review and comparison of model selection algorithms. Working Papers in Economics 32(5):979–986

2.3. Problem statement definition

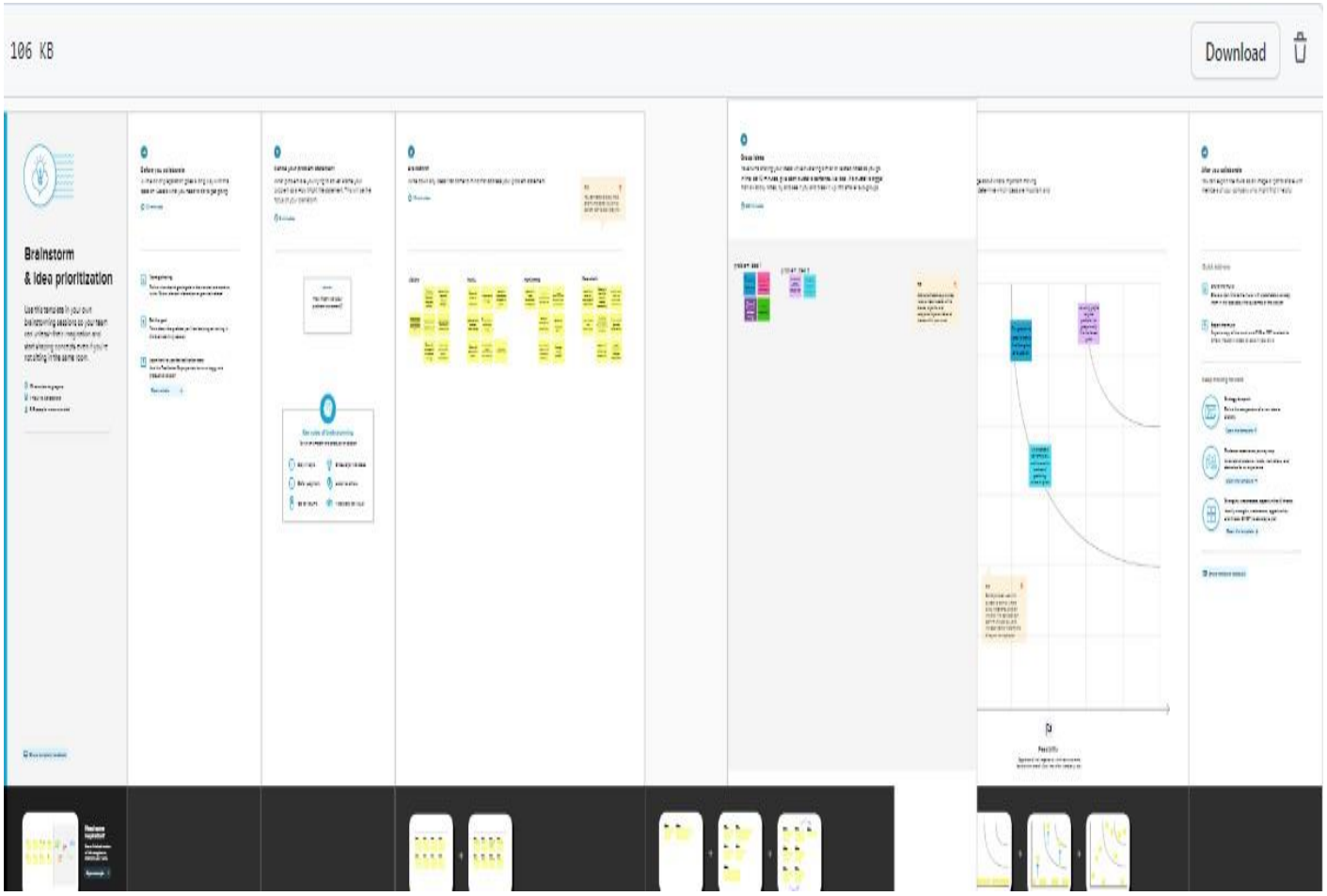
Crude oil price prediction is a challenging task in oil producing countries. Its price is among the most complex and tough to model because fluctuations of price of crude oil are highly irregular, nonlinear and varies dynamically with high uncertainty.

3. IDEATION AND PROPOSED SOLUTION

3.1. Empathy map canvas

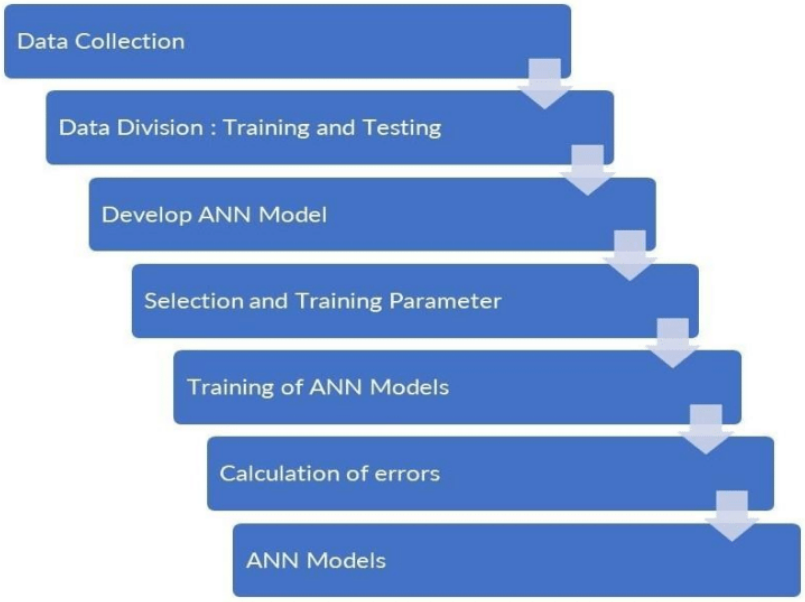


3.2. Ideation and Brainstorming



3.3. Proposed Solution

According to economic theory, the price of crude oil should be easily predictable **from the equilibrium between demand and supply**, wherein demand forecasts are usually made from GDP, exchange rates and domestic prices, and supply is predicted from past production data and reserve data.



3.4. Problem Solution Fit

a. High oil rate can be solved by decreasing the use of personal vehicle.

b. Stop using vehicle in unnecessary work.

C. Govt should import more and more oil from Foreign.

d. Govt should manage this cost with other things and decrease the cost of oil.

4. REQUIREMENT ANALYSIS

4.1. Functional Requirements

User Type User Story / Task	Type Functional Requirement (Epic)	User Story Number	User Story /Task
Predictors	Data analysis and collection	USN-1	AS A USER I CAN ANALYSIS OF influencing factors of curde oil price through various web pages
		USN-2	Data samples are collected from various web sites
		USN-3	The collected data will be proceed to remove the raw data from the data site
	Pre-processing	USN-4	The preprocessed data will be put in for the training and splitting
	Training model	USN-5	The trained data will be checked for the target page that is set by user
		USN-6	If user range is not yet meet again the data is put in for training
		USN-7	Attempt is made until the range is reached
	Predicted value	USN-8	Once the range is reached will be compared with previous result to obtain the correct predicted value

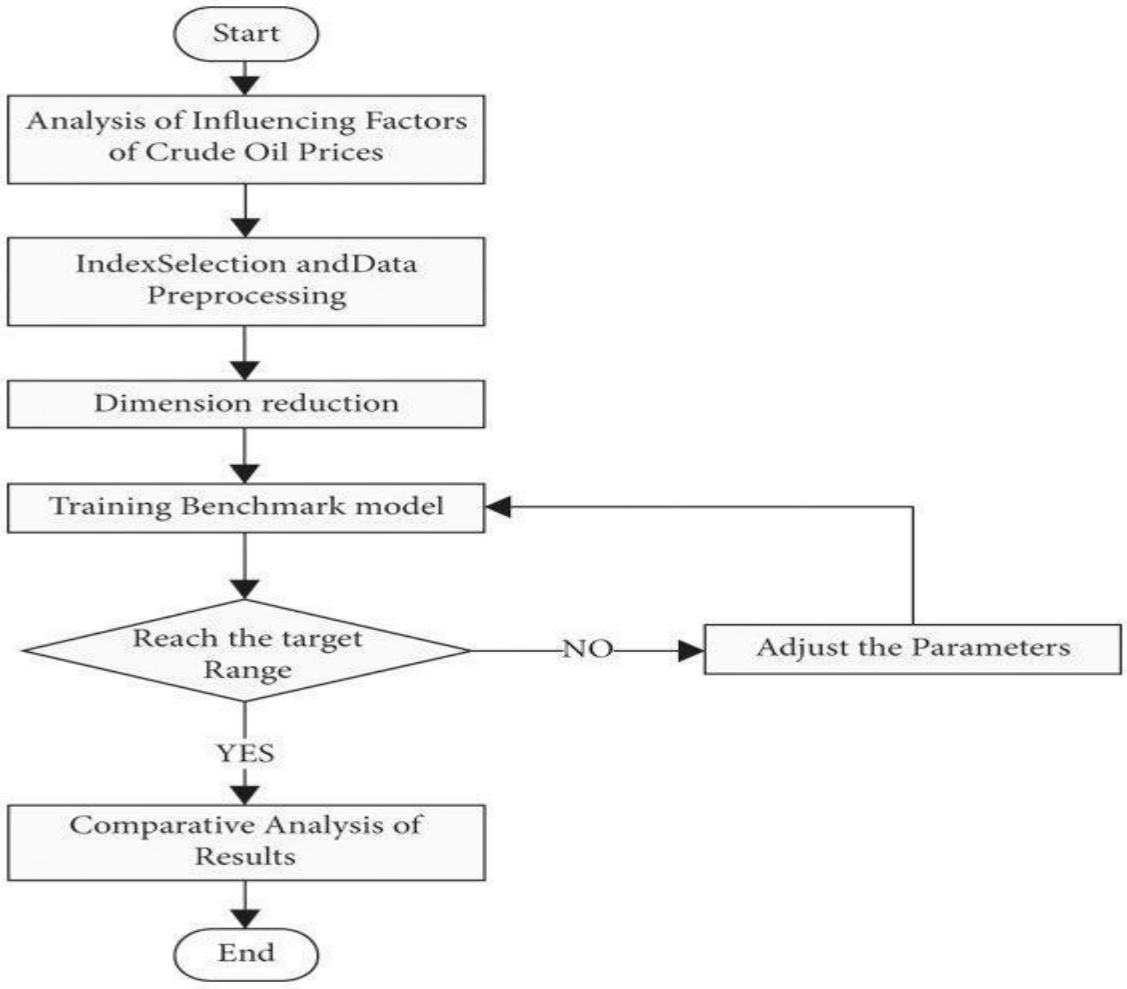
The price of crude oil should be easily predictable from the equilibrium between demand and supply, wherein demand forecasts are usually made from GDP, exchange rates and domestic prices, and supply is predicted from past production data and reserve data.

4.2. Non functional requirements

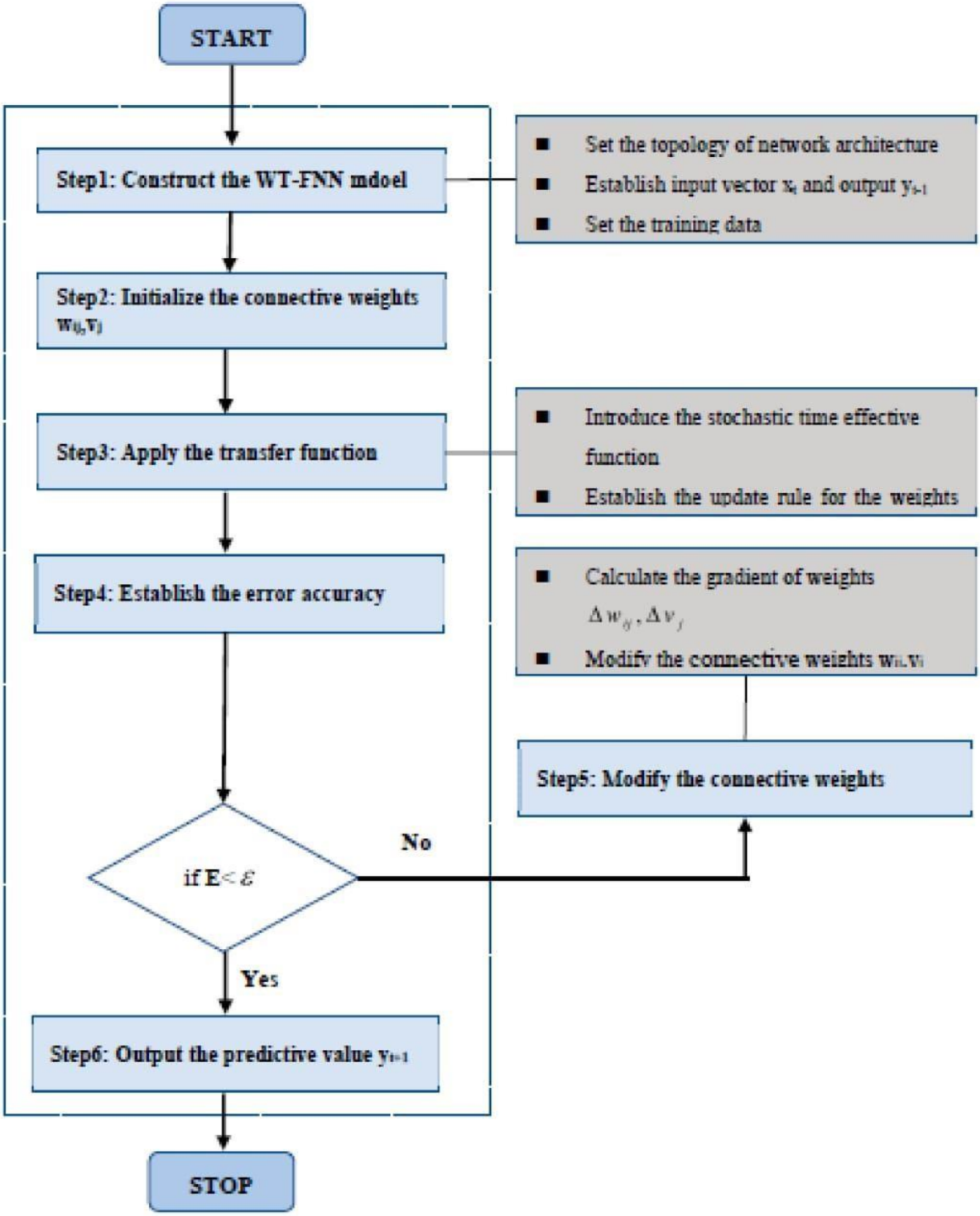
We use two standard performance metrics in the oil price prediction literature for comparing different oil price prediction models. The first metric is Mean Squared Prediction Error (MSPE). MSPE of a prediction model measures the average of the squares of the prediction errors. The prediction error is the difference between the true value and the predicted value.

5. Project Design

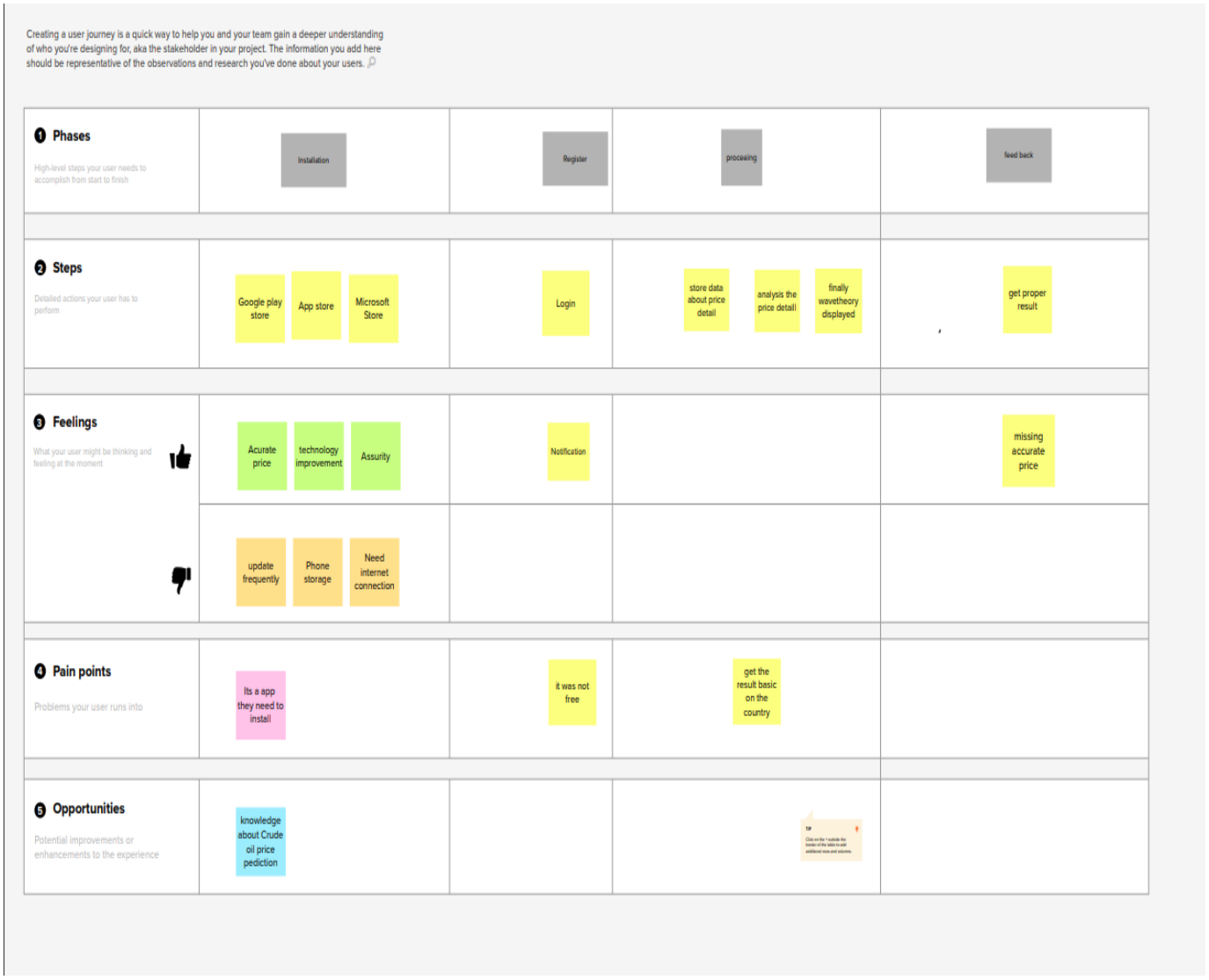
5.1. Data Flow Diagram



5.2. Solution and Technical Architecture



5.3. User Stories



6. PROJECT PLANNING AND SCHEDULING

6.1. Sprint Planning and Estimation

Planning and Estimation are essential in software projects to achieve predictability, reduce the risks involved, and set a basic expectation for all stakeholders. Planning brings a lot of focus on preparation and forecasting whereas Estimation is a process to forecast project-related variables i.e., effort, scope, schedule, etc.

Planning: Planning is required irrespective of the project management methodologies that the team follows, whether it is Waterfall or Agile. Planning gives the project team a perspective on how to meet the objective in a systematic way and helps project stakeholders to keep a tab on the project progress and investments done.

6.2. Sprint Delivery Schedule

Manage result for sprint delivery schedule

In case you're unfamiliar, a sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.28-Jan-2022

6.3. Report From JIRA

Crude oil prices hit multiyear lows recently. Unlike previous dips, the impact of this market shock is expected to persist through 2015 and perhaps beyond.

GEP's commodity experts expect no significant change in supply-demand factors for 2015 and forecast prices in a range of $55/bbl to $65/bbl for the upcoming year. No doubt, such a fundamental shift in the crude oil sector will have a major impact on other commodities and supply chains overall.

This paper explores the key drivers behind the recent decline in crude prices and assesses the impact on other major commodities. The paper also describes four key strategies that would help enterprise sourcing and procurement teams capitalize on this global phenomenon.

7 .CODING AND SOLUTIONING

7.1. Feature1

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder
```

```
from keras.models import Model,Sequential
from keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding
from keras.optimizers import Adam
from keras.preprocessing.text import Tokenizer
from keras.preprocessing import sequence
from keras.utils import pad_sequences
from keras.utils import to_categorical
from keras.callbacks import EarlyStopping
%matplotlib inline
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

Reading the dataset

```
df = pd.read_csv('/content/drive/MyDrive/spam.csv',encoding='latin-1')
df.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN

1 ham Ok lar... Joking wif u oni... NaN NaN NaN

2 spam Free entry in 2 a wkly comp to win FA Cup fina... NaN NaN NaN

3 ham U dun say so early hor... U c already then say... NaN NaN NaN

4 ham Nah I don't think he goes to usf, he lives aro... NaN NaN NaN

```
df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],axis=1,inplace=True)

df.info()
```

RangeIndex: 5572 entries, 0 to 5571

Data columns (total 2 columns):

#	Column	Non-Null Count	Dtype
0	v1	5572 non-null	object
1	v2	5572 non-null	object

dtypes: object(2)

memory usage: 87.2+ KB

```
sns.countplot(df.v1)

plt.xlabel('Label')

plt.title('Number of ham and spam messages')
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Text(0.5, 1.0, 'Number of ham and spam messages')

```
X = df.v2

Y = df.v1

le = LabelEncoder()

Y = le.fit_transform(Y)

Y = Y.reshape(-1,1)

X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.15)
```

Process the data

```
max_words = 1000

max_len = 150

tok = Tokenizer(num_words=max_words)

tok.fit_on_texts(X_train)
```

```
sequences = tok.texts_to_sequences(X_train)

sequences_matrix = pad_sequences(sequences,maxlen=max_len)

sequences_matrix

array([[  0,    0,  0,  ..., 841,  17, 428],
       [  0,   0,   0, ..., 370,  12,  14],
       [  0,   0,   0, ...,  39,  38,  32],
       ...,
       [  0,   0,   0, ...,  13, 608, 180],
       [  0,   0,   0, ...,  20,  6, 204],
       [  0,   0,   0, ...,  18,   5, 144]], dtype=int32)

tok.index_word

{1: 'i',
 2: 'to',
 3: 'you',
 4: 'a',
 5: 'the',
 6: 'u',
 7: 'and',
```

8: 'in',

9: 'is',

10: 'me',

11: 'my',

12: 'for',

13: 'your',

14: 'it',

15: 'of',

16: 'call',

17: 'have',

18: 'on',

19: 'that',

20: '2',

21: 'are',

22: 'now',

23: 'so',

24: 'not',

25: 'but',

```
 26: 'or',

 27: 'at',

 28: 'can',

 29: 'do',

 30: 'with',

 31: 'be',

 32: 'get',

 33: 'if',

 34: 'just',

 35: "i'm",

 36: 'no',

 37: 'ur',

 38: 'will',

 39: 'we',

 40: 'gt',

 41: 'lt',

TOT_SIZE = len(tok.word_index)+1

lstm_model = Sequential()
```

```
lstm_model.add(Embedding(TOT_SIZE, 32, input_length=max_len))

lstm_model.add(LSTM(100))

lstm_model.add(Dropout(0.4))

lstm_model.add(Dense(20, activation="relu"))

lstm_model.add(Dropout(0.3))

lstm_model.add(Dense(1, activation = "sigmoid"))
```

Compile the model

```
lstm_model.compile(loss = "binary_crossentropy", optimizer = "adam", metrics = ["accuracy"])

lstm_model.summary()
```

Model: "sequential"

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 150, 32)           263648

 lstm (LSTM)                 (None, 100)               53200
```

dropout (Dropout) (None, 100) 0

dense (Dense) (None, 20) 2020

dropout_1 (Dropout) (None, 20) 0

dense_1 (Dense) (None, 1) 21

==========

Total params: 318,889

Trainable params: 318,889

Non-trainable params: 0

Fit the model

```
lstm_model.fit(sequences_matrix,Y_train,batch_size=128,epochs=10,
        validation_split=0.2,
```

```
        workers=10,
         callbacks=[EarlyStopping(monitor='val_loss',min_delta=0.0001)])
```

```
Epoch 1/10
30/30 [==============================] - 14s 379ms/step - loss: 0.4631 -
accuracy: 0.8524 - val_loss: 0.3151 - val_accuracy: 0.8650
Epoch 2/10
30/30 [==============================] - 11s 359ms/step - loss: 0.2174 -
accuracy: 0.9311 - val_loss: 0.1200 - val_accuracy: 0.9705
```

Save the model

```
lstm_model.save('sms.h5')
```

Test the model

```
test_sequences = tok.texts_to_sequences(X_test)
test_sequences_matrix = pad_sequences(test_sequences,maxlen=max_len)
acc = lstm_model.evaluate(test_sequences_matrix,Y_test)
27/27 [==============================] - 1s 36ms/step - loss: 0.1192 -
accuracy: 0.9749
print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(acc[0],acc[1]))
```

Test set

Loss: 0.119

Accuracy: 0.975

7.2. Feature2

Basic Python

1. Split this string

s = "Hi there Sam!"

print(s.split())

['Hi', 'there', 'Sam!']

2. Use .format() to print the following string.

Output should be: The diameter of Earth is 12742 kilometers.

planet = "Earth"

diameter = 12742

a="The daimeter of {planet}is{diameter}kilometers"

print(a.format(planet="Earth",diameter=12742))

The daimeter of Earthis12742kilometers

3. In this nest dictionary grab the word "hello"

d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}

print(d)

{'k1': [1, 2, 3, {'tricky': ['oh', 'man', 'inception', {'target': [1, 2, 3, 'hello']}]}]}

Numpy

import numpy as np

b=np.zeros(10)*0

print(b)

[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

4.1 Create an array of 10 zeros?

4.2 Create an array of 10 fives?

import numpy as np

b=np.ones(10)*5

print(b)

[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]

5. Create an array of all the even integers from 20 to 35

import numpy as np

a=np.arange(20,35,2)

print(a)

[20 22 24 26 28 30 32 34]

6. Create a 3x3 matrix with values ranging from 0 to 8

```
import numpy as np

a=np.arange(0,9).reshape(3,3)

print(a)
```

```
[[0 1 2]

 [3 4 5]

 [6 7 8]]
```

7. Concatenate a and b

```
a = np.array([1, 2, 3]), b = np.array([4, 5, 6])

import numpy as np

arr1=np.array([1,2,3])

arr2=np.array([4,5,6])

arr=np.concatenate((arr1,arr2))

print(arr)
```

```
[1 2 3 4 5 6]
```

Pandas

8. Create a dataframe with 3 rows and 2 columns

```
import pandas as pd
```

```
data=[{'a':12,'b':45},{'a':54,'b':23},{'a':94,'b':76}]

df=pd.DataFrame(data)

print(df)

   a  b

0  12  45

1  54  23

2  94  76
```

9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023

```
import pandas as pd

a=pd.date_range(start='1/1/2023',end='10/2/2023')

print(a)

DatetimeIndex(['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',

               '2023-01-05', '2023-01-06', '2023-01-07', '2023-01-08',

               '2023-01-09', '2023-01-10',

               ...

               '2023-09-23', '2023-09-24', '2023-09-25', '2023-09-26',

               '2023-09-27', '2023-09-28', '2023-09-29', '2023-09-30',

               '2023-10-01', '2023-10-02'],
```

dtype='datetime64[ns]', length=275, freq='D')

10. Create 2D list to DataFrame

```
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

```
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
import pandas as pd
lst=[[1,'aaa',22],[2,'bbb,25'],[3,'ccc',24]]
df=pd.DataFrame(lst)
print(df)
```

```
  0      1     2
0  1     aaa  22.0
1  2 bbb,25   NaN
2  3     ccc  24.0
```

8. TESTING

8.1. Test Cases

Crude oil prediction using computational intelligence techniques aims at enhancing the crude oil industry. Using test cases processes, we are able to discover information, more effective for different classes of

and prove strictness prediction results. In the case of crude oil prediction, the prediction results are going to be so conservative that it is often felt useless for decision-making, using test cases and clustering functions of the predicted results and empirical values prove to have more precision and efficiency.

8.2. User Acceptance Testing

a. Analyze product requirements and define key deliverables. ...

b. Choose the time and form of end-user testing. ...

c. Recruit users and form UAT team.

d. Implement end-user testing tools and onboard testers.

9. RESULTS

9.1. Performance Metrics

The crude oil prices are taken out by implementing the performance of the proposed model. The proposed model is here used to predict the closing price of crude oil.

10. ADVANTAGES

Quite a lot of plastics and other synthetic materials are derived from oil and higher prices ripple through the economy. With high oil

prices, then, comes increased interest and R&D into non-oil alternative feedstocks for these materials. This process has a lot of fringe benefits for the economy as a whole.

- It can produce a large amount of energy.
- Petroleum is a highly dense fuel source.
- Its technology and infrastructure are already in place.
- It can be used in a wide range of applications.

DISADVANTAGES

- Crude Oil Price prediction can not be accurate at all the time.
- Predicting the crude oil price is difficult.

11. CONCLUSION

In this paper, an artificial intelligence model is presented with the task of determining the most favorable lag

in the crude oil price data. It is evident, the result is shown in the figure, the prediction is accurate till there is a

massive and sudden change in the actual data, where it becomes challenging to predict the exact new price with the change, however,

the proposed model has efficiently taken into consideration these patterns.

12. FUTURE SCOPE

This work is carried out on the closing price of crude oil; however, there are various other factors

which also affect the crude oil prices like change in the prices and quantities (demand and supply), change in the

economy and current affairs as shown by the media. The main advantage of this research is in capturing the changing

pattern of these prices. In the coming future, fundamental indicators and market trends have been planned to be

incorporated into a model which will help the proposed model perform more efficiently.

13. APPENDIX

Source code

Loading the excel data file into pandas using the read_excel() function. Locating the directory of the excel file at first which is more efficient to keep the dataset in the same directory.

Data= pd.read_excel(r”File_location”)

r stands for "raw" and will cause backslashes in the string to be interpreted as actual backslashes rather than special characters.

```
import pandas as pd
import os
pwd
'/content'
variable = pd.read_csv("/content/Crude Oil Prices Daily.csv")
variable.head()
```

	Date	Closing Value
0	1/2/1986	25.56
1	1/3/1986	26.00
2	1/6/1986	26.53
3	1/7/1986	25.85

4 1/8/1986 25.87

variable = pd.read_csv(r"/content/Crude Oil Prices Daily.csv")

variable.head()

Date Closing Value

0 1/2/1986 25.56

1 1/3/1986 26.00

2 1/6/1986 26.53

3 1/7/1986 25.85

4 1/8/1986 25.87

Demo video link : https://drive.google.com/file/d/1tH5IvRxGgYvJSluaCFQmtyKzqbAcpkQL/view?usp=share_link

Github link : https://github.com/IBM-EPBL/IBM-Project-47648-1660800767/blob/2e9f8203ed71a450f36cad56fbd3a555be2e8ed4/Final%20deliverable/model%20video%20.mp4