IOMP Project Report on

CRUDE OIL PRICE PREDICTION USING IBM WATSON STUDIO

SUBMITTED BY

KASU RAHUL	**19UK1A0530**
MITHINTI VYSHNAVI	**19UK1A0549**
BAIREDDY ABHINAVI	**19UK1A0567**
MOHAMMED GHULAM ATHAR MOINUDDIN	**19UK1A0504**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal – 506005

2019– 2023

CONTENTS

CHAPTER 1

INTRODUCTION

Prediction of crude oil prices has been a wide topic for ages. People use their intuition and lot of techniques to guess the prices of crude oil. It takes a lot of knowledge about the crude oil to accurately predict it. Predicting the crude oil price is very significant in various economic, political and industrial areas, both for crude oil importer and exporter countries. Since the crude oil is important strategic resource around the globe; it has become the crucial commodity for the world's economy. Thus, prediction of prices of crude oil has always been considered as a very exciting and challenging task which drew the curiosity of professionals, researchers and organizations all over the world.

1.1 OVERVIEW

This Guided Project mainly focuses on applying Neural Networks to predict the Crude Oil Price. This decision 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.

Recurrent neural networks (RNN) have been proved to be one of the most powerful models for processing time-series based sequential data. LSTM is one of the most successful RNN architectures. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With these memory cells, networks are able to effectively associate memories and input remote in time, hence suit to grasp the structure of data dynamically over time with high prediction capacity.

Recurrent neural network are a type of Neural Network where the output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

LSTMs have chain like structure with the repeating module having a different structure. There are four neural network layers which are interacting to each other in a special way. The key to LSTMs is the cell state, which is the horizontal line running through the top of the diagram. The cell state runs straight down the entire chain, with only some minor linear interactions. The information flows along it unchanged. The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural network layer and a point-wise multiplication operation. An LSTM has three of these gates, to protect and control the cell state.

1.2 PURPOSE

The main objectives of the project is:

- To prepare model to predict crude oil prices for next coming days, weeks months and quarters using LSTM based RNN concepts by using synthetic data.

- To create a dashboard for various prediction activities

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM (OR) Problem Statement

Nowadays, the increased oil prices worldwide are having a great impact on all economic activities. Over the years there has been a fluctuation in petroleum prices, and a close consideration of the demand and supply side effects that sparked these price changes shows there is high probability that these changes will continue in the outlook period and beyond. West African Monetary Agency (2008) concluded that increase in world oil prices have been shown to worsen fiscal deficit positions of oil importing countries like Ghana. For this reason, we believe that if the government can see ahead of monthly petroleum prices, our deficit would not be worsened.

2.2 PURPOSED SOLUTION

Predictive analytics for crude oil price using RNN-LSTM experimental procedure includes five systematic steps as follows:

1. Data Acquisition
2. Data Exploration And Pre-Processing
3. Data Training
4. Test And Evaluate Model
5. Model Improvement.

The method or solution is Jupiter notebook and spyder we used to complete this project. and youwill use this jupiter notebook for you recommended.

To build Machine learning models you must require the following packages

Sklearn: Scikit-learn is a library in Python that provides many unsupervised and supervised learningalgorithms.

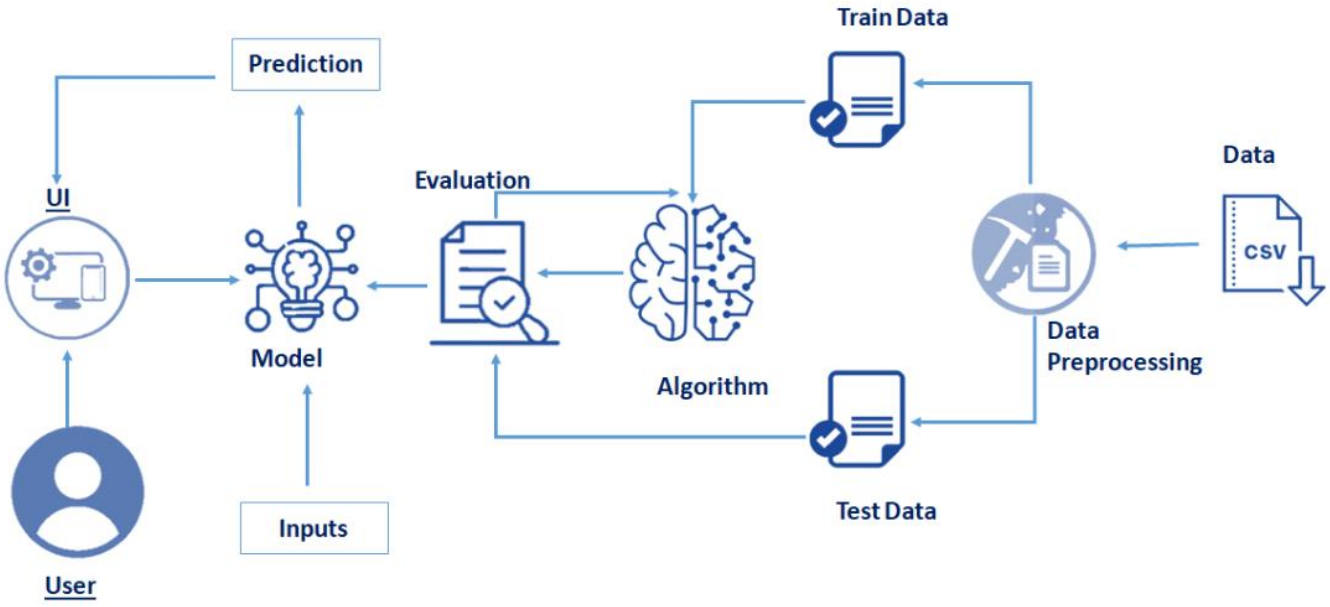
NumPy: NumPy is a Python package that stands for 'Numerical Python'. It is the core library for scientificcomputing, which contains a powerful n-dimensional array object

Pandas: pandas is a fast, powerful, flexible, and easy to use open source data analysis and manipulationtool,built on top of the Python programming language.

CHAPTER 3

THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM



This Project mainly focuses on applying Neural Networks to predict the Crude Oil Price. This decision 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.

3.2 HARDWARE / SOFTWARE DESIGNING

The hardware required for the development of this project is:

Processor : Intel CoreTM i5-9300H
Processor speed : 2.4GHz
RAM Size : 8 GB DDR
System Type : X64-based processor

SOFTWARE DESIGNING:

The software required for the development of this project is:

Desktop GUI : Anaconda Navigator
Operating system : Windows 10
Front end : HTML, CSS,JAVASCRIPT
Programming : PYTHON
Cloud Computing Service : IBM Cloud Services

Anaconda Navigator :

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder.

To build Deep learning models you must require the following packages

1. Tensor flow:

   TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers can easily build and deploy ML-powered applications.

2. Keras:

   Keras leverages various optimization techniques to make high-level neural network API easier and more performant. It supports the following features:

   - Consistent, simple, and extensible API.
   - Minimal structure - easy to achieve the result without any frills.
   - It supports multiple platforms and backends.
   - It is a user-friendly framework that runs on both CPU and GPU.
   - Highly scalability of computation.

Flask: Web framework used for building Web applications

If you are using **anaconda navigator**, follow the below steps to download the required packages:

- open anaconda prompt as administrator
- Type “pip install tensorflow” (make sure you are working on python 64 bit)
- Type “pip install flask”.
- Type "pip install keras

The above steps allow you to install Keras and TensorFlow in the anaconda environment.

CHAPTER 4

EXPERIMENTAL INVESTIGATION

Coming to analysis or investigations on the following Concepts :

1. Supervised and unsupervised learning:
2. Regression Classification and Clustering :
3. Artificial Neural Networks:
4. Recurrent Neural Networks and Long Short–Term Memory:

1. Supervised and unsupervised learning:

   - Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.
   - Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.

2. Regression Classification and Clustering :

   - Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables. More specifically, Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed.
   - Clustering or cluster analysis is a machine learning technique, which groups the un labelled dataset. It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."

3. Artificial Neural Networks:

   - The term "Artificial neural network" refers to a biologically inspired sub-field of artificial intelligence modeled after the brain. An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain. Similar to a human brain has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks. These neurons are known as nodes.

4. Recurrent Neural Networks and Long Short – Term Memory

   - A recurrent neural network (RNN) is a special type of artificial neural network adapted to work for time series data or data that involves sequences. Ordinary feedforward neural networks are only meant for data points that are independent of each other. However, if we have data in a sequence such that one data point depends upon the previous data point, we need to modify the neural network to incorporate the dependencies between these data points. RNNs have the concept of "memory" that helps them store the states or information of previous inputs to generate the next output of the sequence.
   - LSTM is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data.

IMPORT REQUIRED LIBRARIES:

Import the necessary libraries as shown in the figure.

- Spacy is an open-source software library for advanced natural language processing.
- Package string implements simple functions to manipulate UTF-8 encoded strings.

READ DATASET:

We collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository, etc.

The dataset used for this project was obtained from Kaggle. https://www.kaggle.com/datasets/rockbottom73/crude-oil-prices

This dataset contains two columns

- Date
- Closing Value

It contains crude oil prices from 1988 to 2018.

CHAPTER 5

FLOWCHART

PROJECT FLOW:

Input Data: We need to input the last 30 years collected data as input.

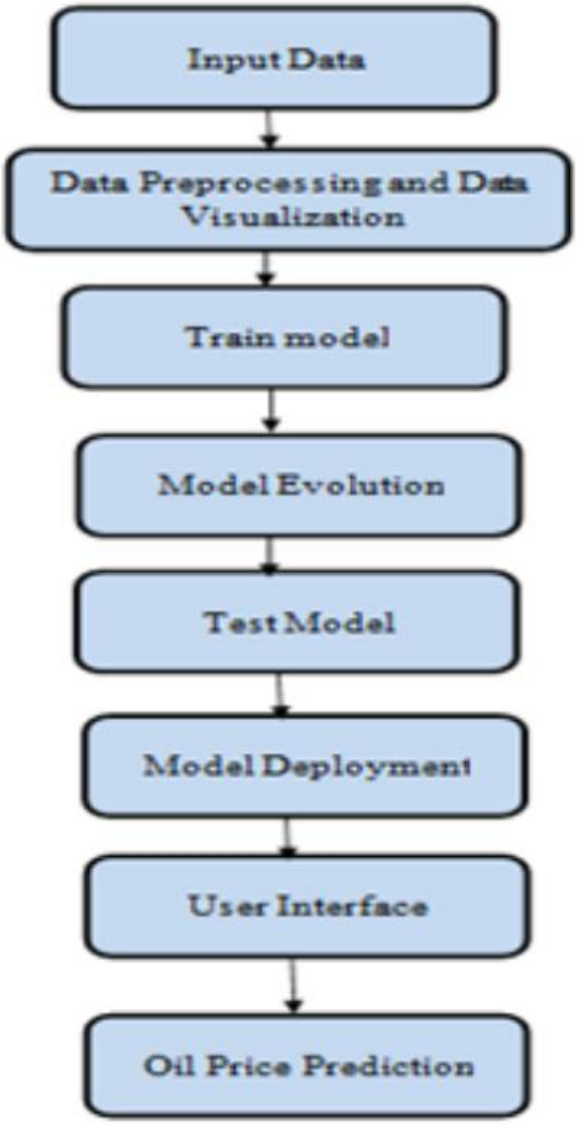
Data pre-processing: This step includes

- Import the Libraries.
- Importing the dataset.
- Analyse the data
- Taking care of Missing Data
- Feature Scaling
- Data Visualization
- Splitting Data into Train and Test.

Training the RNN and LSTM classifier using pretrained Models: The model will be using the technique of Transfer Learning for training the model - "Feature Extraction from pre-trained model and training a classifier using extracted features"

Validation and Testing: Once the model is trained using the train dataset (the sample of data used to fit the model) then validated using validation dataset (The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.) and finally tested using the test dataset.

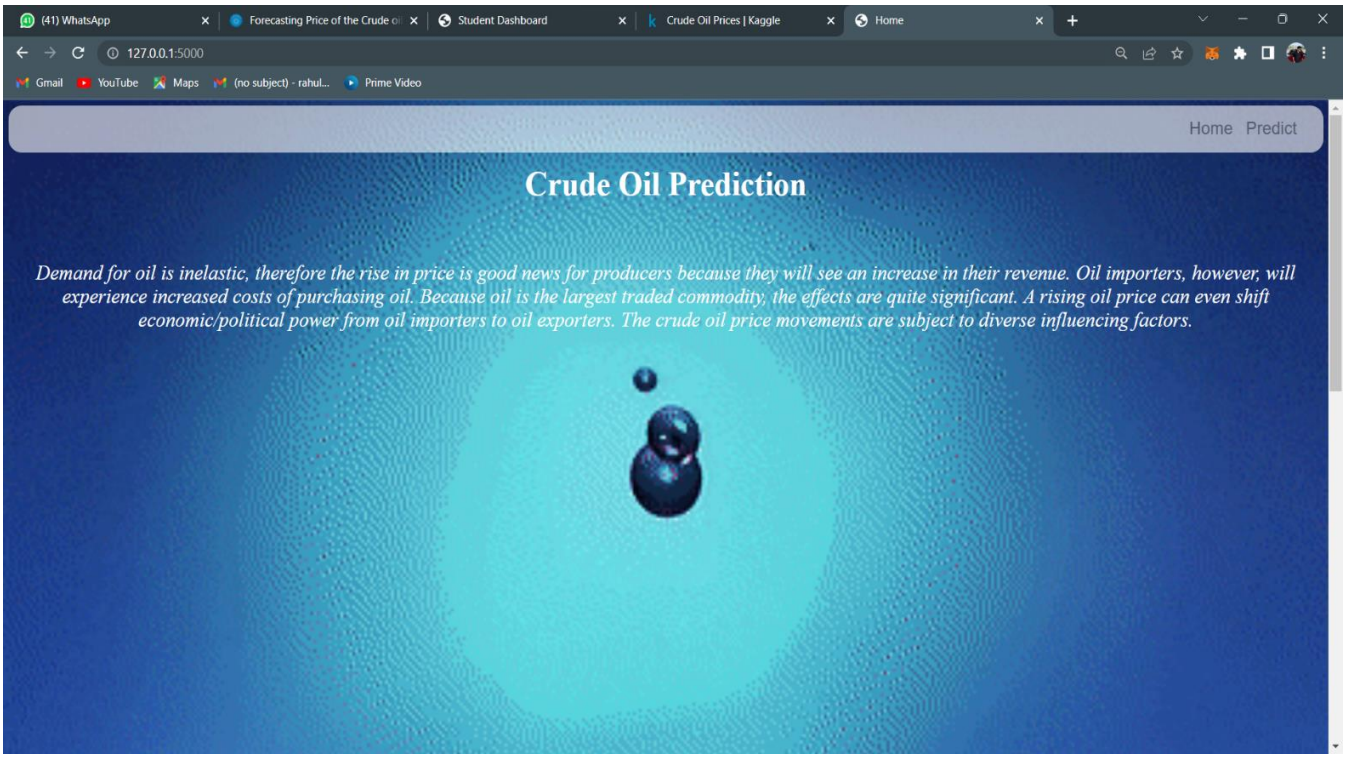
Oil Price Prediction: Finally we the oil price predicted as the output from our implemented system.


Input Data
Data Preprocessing and Data Visualization
Train model
Model Evolution
Test Model
Model Deployment
User Interface
Oil Price Prediction

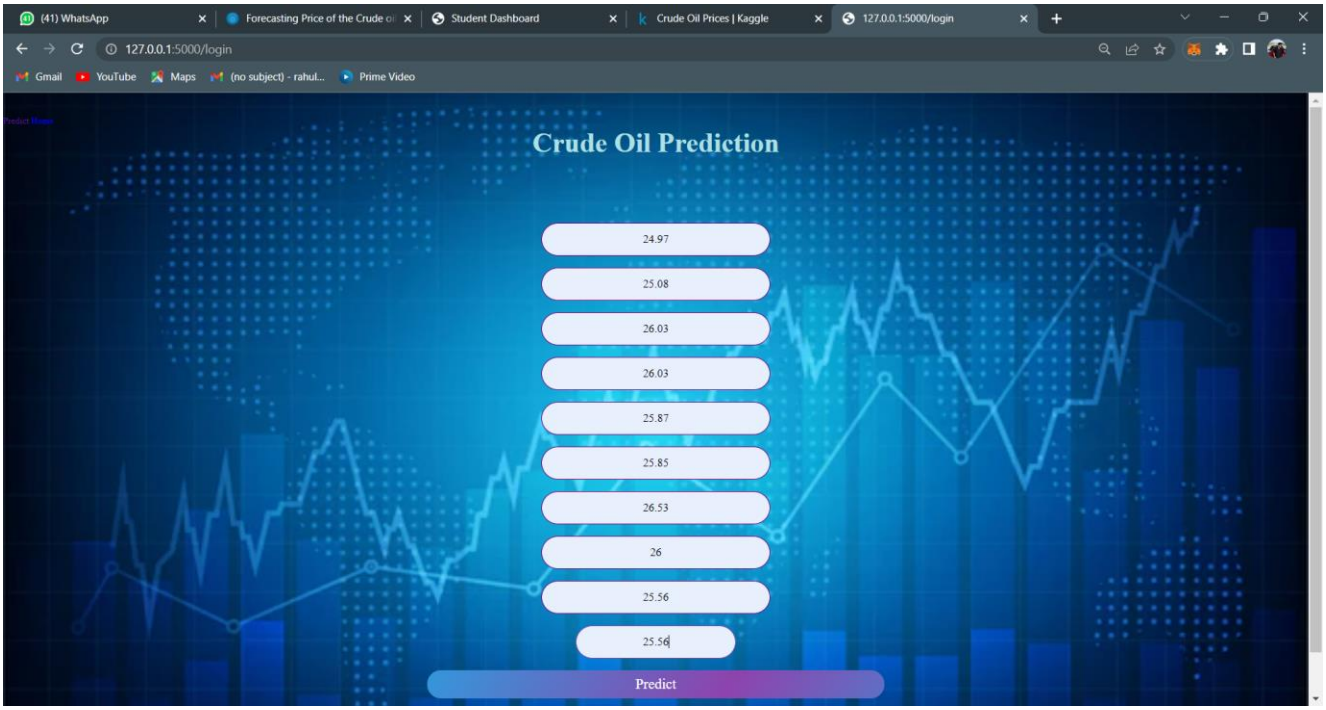
CHAPTER 6

RESULT

Home Page Of Crude Oil Price Prediction



Predicting The Values of Crude Oil



Final Result Of the Project

CHAPTER 7

CONCLUSION

Prediction of crude oil price is considered a challenging task due to the nature of the supply-demand curve, as well as various factors affecting the price volatility and demand in crude oil. Despite this condition, in the long run, the crude oil supply cost is the most significant factor in determining the crude oil price.

The RNN-LSTM model developed using Python and TensorFlow-Keras library has successfully predicted the crude oil price movement accordingly using machine learning to handle the data analytics task to analyze the upcoming price trends. LSTM focuses on storing the previous data and prediction which is rather encouraging and more approximate. The outcome derived are relatively encouraging. The results show that large look ups do not necessarily improve the accuracy of the predictions of crude oil prices. Hence it can be concluded, the model with single LSTM model is definitely the most accurate.

CHAPTER 8

BIBILOGRAPHY

[1] Mohammad Reza Mahdiani and Ehsan Khamehchi, “A modified neural network model for predicting the crude oil price”, Intellectual Economics, vol. 10, no. 2, pp. 71-77, Aug. 2016.

[2] Manel Hamdi and Chaker Aloui, \"Forecasting Crude Oil Price Using Artificial NeuralNetworks: A Literature Survey,\" Economics Bulletin, AccessEcon, vol. 35, no. 2, pp. 1339-1359, 2015.

[3] Yu Runfang, Du Jiangze and Liu Xiaotao, “Improved Forecast Ability of Oil Market Volatility Based on combined Markov Switching and GARCH-class Model, Procedia Computer Science, vol. 122, pp. 415-422, 2017.

[4] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, \"LSTM: A Search Space Odyssey,\" IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232,Oct. 2017.

[5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

APPENDIX

A Source Code of Flask:

https://drive.google.com/drive/folders/1t4GLObgHmC9lX8qQG2fDeKHBLy9ELb5F?usp=share_link

app.py

```
import numpy as np  # used for numerical analysis
from flask import Flask, render_template, request  # Flask is a application used to run/serve our application
# request is used to access the file which is uploaded by the user in our application
# render_template is used for rendering the html pages
from tensorflow.keras.models import load_model  # we are loading our model from keras


app = Flask(__name__)  # our flask app
model = load_model('crude_oil.h5')  # loading the model in the flask app


@app.route('/')  # rendering html template
def home():
    return render_template("index.html")  # rendering html template


@app.route('/about')
def home1():
```

```
    return render_template("index.html")  # rendering html template


@app.route('/predict')
def home2():
    return render_template("web.html")  # rendering html template


@app.route('/login', methods=['POST'])  # route for our prediction
def login():
    a = request.form['year1']
    b = request.form['year2']
    c = request.form['year3']
    d = request.form['year4']
    e = request.form['year5']
    f = request.form['year6']
    g = request.form['year7']
    h = request.form['year8']
    i = request.form['year9']
    j = request.form['year10']  # requesting the file
    x_input = [[float(a), float(b), float(c), float(d), float(e), float(f), float(g), float(h), float(i),
float(j)]]
    print(x_input)
    lst_output = model.predict(x_input)
    lst_output = np.round(lst_output[0][0], 2)
    return render_template("web.html", showcase='The Predicted crude oil price is : Rs.
'+str(lst_output))


if __name__ == '__main__':
    app.run(debug=False)
```