Project Report On

Stock price predictions using Apache airflow

ML and Flask frame work



### Submitted in partial fulfilment for the award of

Post Graduate Diploma in Big Data Analytics (PG-DBDA) From C-DAC,Patna

**Guided by:**

**Mr. Ranjan Sinha Sir**

**Submitted By**

Shreya Hote **(**220980725006)

kancharla sai sidhartha (220980725005)

Nitish Kumar (220980725003)

Rachit Moghe (220980725001)

**CERTIFICATE**

**TO WHOMSOEVER IT MAY CONCERN**

**This is to certify that**

Shreya Hote **(**220980725006)

kancharla sai sidhartha (220980725005)

Nitish Kumar (220980725003)

Rachit Moghe (220980725001)

**Have successfully completed their project on**

Stock Price Prediction Using using Apache airflow

,ML and Flask frame work

**Under the guidance of Mr. Ranjan Sinha Sir**

**ACKNOWLEDGEMENT**

This project **“Stock Price Prediction Using Apache Airflow,ML and Flask Framework”** was a great learning experience for us and we are submitting this work to CDAC ,Patna.

We all are very glad to mention the name of **Mr. Ranjan Sinha sir** for their valuable guidance to work on this project. Their guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

We are highly grateful to **Mr.Saket Jha Sir** Manager, C-DAC,Patna for his guidance and support whenever necessary while doing this course Post Graduate Diploma in Big Data

Analytics (PG-DBDA) through C-DAC ,Patna.

Our most heartfelt thanks goes to **Mr.Ranjan Sinha Sir**(Course Coordinator, PG-DBDA) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC ,Patna.

**From:**

Shreya Hote **(**220980725006 **)**

kancharla sai sidhartha (220980725005)

Nitish Kumar (220980725003)

Rachit Moghe (220980725001)

**TABLE OF CONTENTS**

**ABSTRACT**

1. **INTRODUCTION**
2. **SYSTEM REQUIREMENTS**
   1. Software Requirements
   2. Hardware Requirements
3. **FUNCTIONAL REQUIREMENTS**
4. **SYSTEM ARCHITECTURE**
5. **METHODOLOGY**
6. **DATA VISUALIZATION AND REPRESENTATION**
7. **RESULT AND FINDING**
8. **CONCLUSION AND FUTURE SCOPE**
9. **REFERENCES**

**Abstract**

Stock price analysis has been a critical area of research and is one of the top applications of machine learning. In this project we will be looking at data from the stock market. It can be used to provide additional insights and context on likely trends in stock performance, based on past performance. For having more understanding in graphical view we are using the Flask-application which helps us to see forecasting graphs of the stocks. We will be talking about predicting the returns on stocks.  For retrieving the data we have used Airflow, for processing the data we have used Spark and Flask Application for Visualization.

## INTRODUCTION

A stock market is a public market where you can buy and sell shares for publicly listed companies. The stocks, also known as equities, represent ownership in the company. The stock exchange is the mediator that allows the buying and selling of shares.



Importance of Stock Market

* Stock markets help companies to raise capital.
* It helps generate personal wealth.
* Stock markets serve as an indicator of the state of the economy.
* It is a widely used source for people to invest money in companies with high growth potential.

Stock Price Prediction

* Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile.

## Datasets and features:

Data used in the project is 5 year data and updates after every fifteen minutes. It was collected [www.finance.yahoo.com](http://www.finance.yahoo.com).To create model & predict stock price algorithm used are KNN,Linear regression ,random forest , etc.

## SYSTEM REQUIREMENTS

**Hardware Requirements:**

 Platform – Windows 10

 RAM – 8 GB of RAM,

 Peripheral Devices – Mouse, Keyboard, Monitor

 A network connection for data recovering over network.

**Software Requirements:**

 OS – Windows 10

 VMWARE Oracle virtual Box

 Python 3

 Apache Spark

 Apache Airflow

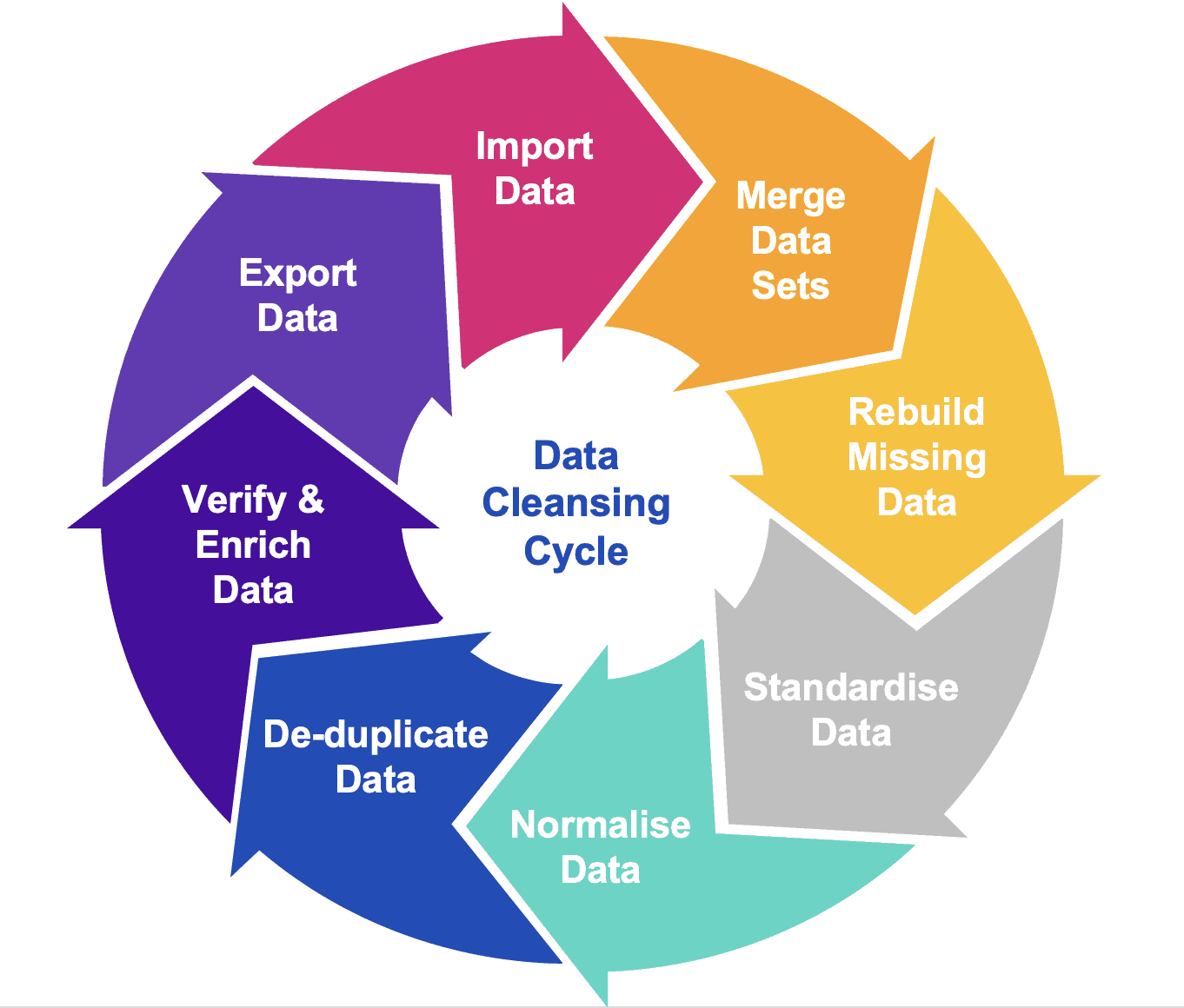
 MongoDB

 Flask-application

## FUNCTIONAL REQUIREMENTS

1. **Python 3:**
   * Python is a general purpose and high level programming language.
   * It is use for developing desktop GUI applications, websites and web applications.
   * Python allows to focus on core functionality of the application by taking care of common programming tasks.
   * Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68,
   * Small Talk, and UNIX shell and other scripting languages.
2. **Apache Airflow:**
   * **Airflow** in Apache is a popularly used tool to manage the automation of tasks and their workflows.
   * They are also primarily used for scheduling various tasks.
   * Apache Airflow automate tasks by training your machine learning model to serve these kinds of tasks on a regular interval specified while training it.
   * Airflow allows you to easily resolved the issue of automating time-consuming and repeating task.
   * Airflow primarily written in SQL and Python.
3. **Apache Spark:**
   * Apache Spark is an open-source cluster computing system that provides high-level API in Java, Scala, Python and R.
   * Apache Spark is one of the fastest-growing big data projects in the history of the Apache Software Foundation. With its memory-oriented architecture, flexible processing libraries, and ease-of-use, Spark has emerged as a leading distributed computing framework for real- time analytics.
   * Spark is used for many types of data processing – it comes packaged with support for machine learning, interactive queries (SQL), statistical queries with R, graph processing, ETL, and streaming.
   * For loading and storing data, Spark integrates with a number of storage MongoDB, and more.
4. **MongoDB:**
   * **MongoDB**, the most popular NoSQL database, is an open-source document-oriented database.
   * MongoDB allows a highly flexible and scalable document structure.
   * MongoDB has built in solution for partitioning and sharing your database.
   * MongoDB provides a variety of storage engines, allowing you to choose one most suited to your application.
   * A real-life scenario for this kind of data manipulation is storing and querying real-time, intraday market data in MongoDB.
5. **Flask-application:**
   * Data visualization is the graphical representation of information and data.
   * It helps create interactive elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.
   * All of this is made possible with gestures as simple as drag and drop.

**Data Cleaning Process:**

****

**Fig: Data Cleaning Process**

Data cleansing or data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Data cleansing may be performed interactively with data wrangling tools, or as batch processing through scripting. After cleansing, a data set should be consistent with other similar data sets in the system. The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores. Data cleaning differs from data validation in that validation almost invariably means data is rejected from the system at entry and is performed at the time of entry, rather than on batches of data.

## SYSTEM ARCHITECTURE

**FLASK APPLICATION**

**JUPYTER-NOTEBOOK**

**MONGODB**

**AIRFLOW**

**DATA GATHERING**

**DATA PROCESSING**

**MAC. LEARNING**

**DATA STORAGE**

**VISUALIZATION**

**Fig: System Architecture of Price Prediction**

1. **METHODOLOGY**

STEP – I : Data Gathering Using Apache AirFlow :

Here we have used Apache Airflow as our initial staring point of the project. In this step we have constructed dags for automating the data downloading from the Y-Finance API.

The dag also has the operator for dumping / storing the data into our database Mongo DB. The dag tasks hierarchy is as Follows -

STEP – II : Use of the Stored Data for Machine Learning :

Here we have extracted the data which we stored in Mongo DB in the first step for our Machine Learning Algorithms. Here we have used various algorithms such as LSTM, Linear Regression, SVR, Elastic Net, KNN, Random Forest etc. for predicting the stock prices based on the analysis of the historical stock prices.

* 1. **Long Short-term Memory :**
  + Long short-term memory (LSTM) is one of the most popular subjects in Deep Learning. There are many works about predicting financial markets with this method since it works well with time series data.
  + LSTM is an artificial RNN architecture that was designed to solve the vanishing gradient problem that can be encountered during the training phase of traditional RNNs.
  + The aim of this article is introducing LSTM basics and measuring LSTM efficiency when there is a limited amount of data for predicting future movements of Stocks prices.
  + For this purpose a simple LSTM network application was constructed. The following part describes the artificial and recurrent neuron networks, vanishing gradient problem and LSTMs.
  + Result part reveals the results of application results and discusses its performance analysis.
  + Conclusion part is about consequences that are rendered with this study and future improvements that can be done about this research.

**Description:**

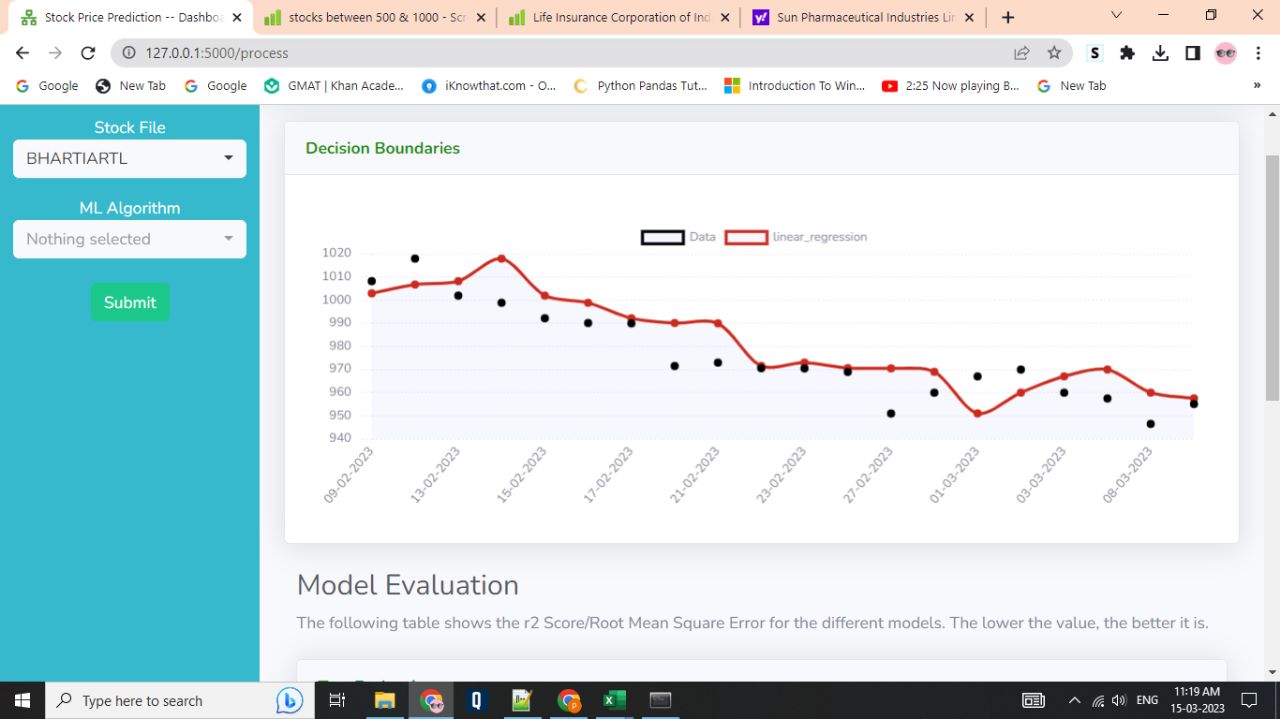
* + To predict Stock prices with the LSTM architecture, a predictor system for experimental results was developed. A data set containing last 5 years Stocks Price daily prices was used for that purpose. 5 years data was preferred because it is the time when stocks started to be popular all around the world with high trading volume.
  + The data was divided in 80% for training, and 20% for testing. Train set was used to train our LSTM network. Validation set was used to measure the loss and performance of our system. Finally, the test data set was used for prediction.
  + The application is predicting the price of future days using past 30 days price for the selected date using patterns or sequences of the trained LSTM network. It doesn’t mean it takes into consideration only 60 days data since it uses each day of the data set to train the system.
  + For implementing LSTM application, Keras library was used with Tensorflow backend. Keras is a high-level neural networks API, written in Python while Tensorflow is an end-to-end open source machine learning platform.
  + Linear was used as an activation function. Optimizer algorithm was selected as Adam (adaptive moment estimation).
  + Adam algorithm was selected because it is computationally efficient and requires little tuning. Mean square error method was used as loss function which is the average squared difference between the predicted and the real values.
  + Input layer was composed of 50 inputs for each time-step and 1 input feature which is STOCKS‘Close’ price. 2 hidden layer was constructed having 50, 50 neurons respectively. Final layer was output layer which produces 1 outputs for each time-step.
  + Epoch number was 100 for the system which is the number of how many times the network will train itself repeatedly. Different number of epochs were experienced and 64 batch size used for the application.
  + Two approaches were used for evaluating the performance of the LSTM network which are comparing loss and R2 score of the validate set. Although they are useful to calculate the effectiveness of our system, the graphs also help us to detect the conditions like underfitting or overfitting.
  + Underfitting means a model can’t learn from the training data set while overfitting means a model learns the data set very well so it can’t interpret and generalize a new data different from the training set is present.
  1. **KNN :**
* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
  1. **Linear Regression :**
* **Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables.
* It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.
* Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.
  1. **Random Forest :**
* Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
* As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
* **The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

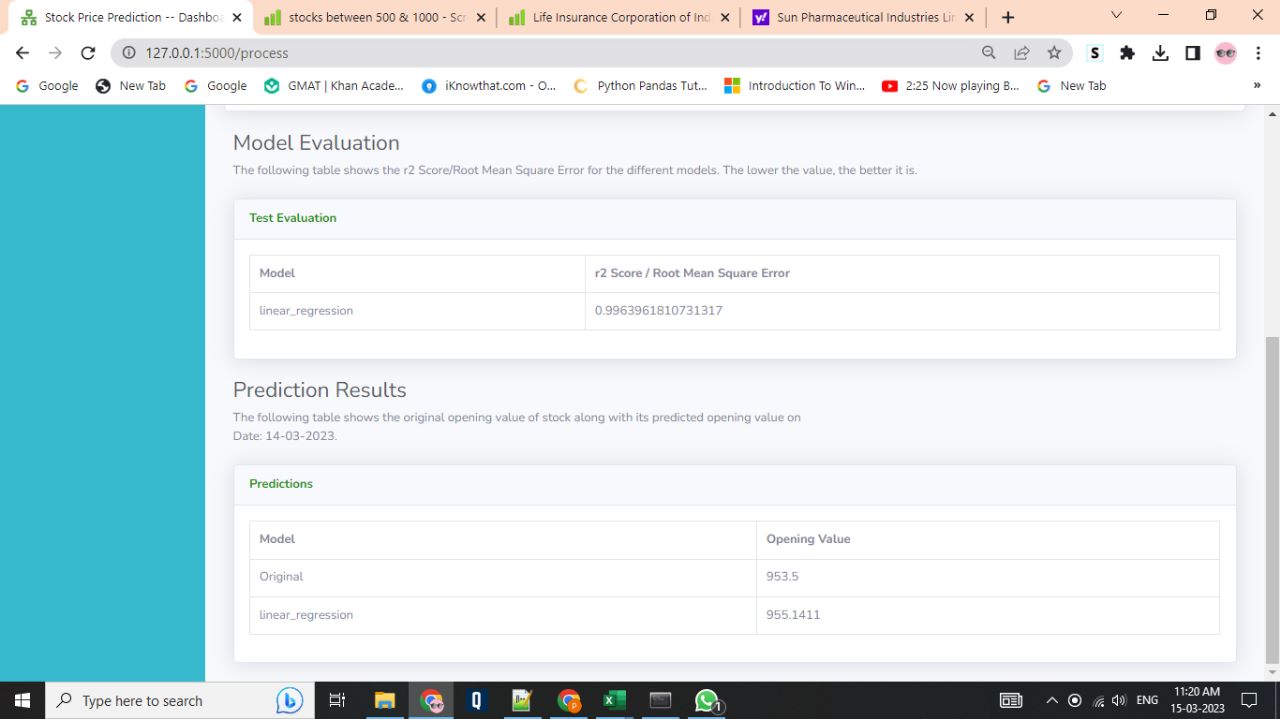
And also some more algorithms have been applied for testing which model fits best for Stock Price prediction.

STEP – III : Use of FLASK to visualize the data that we have analyzed using Machine Learning Algorithms. In this application the user can select the stock name for which he/she wishes to check the stock prices along with the Algorithm he / she wishes to apply to the stock for further prediction.

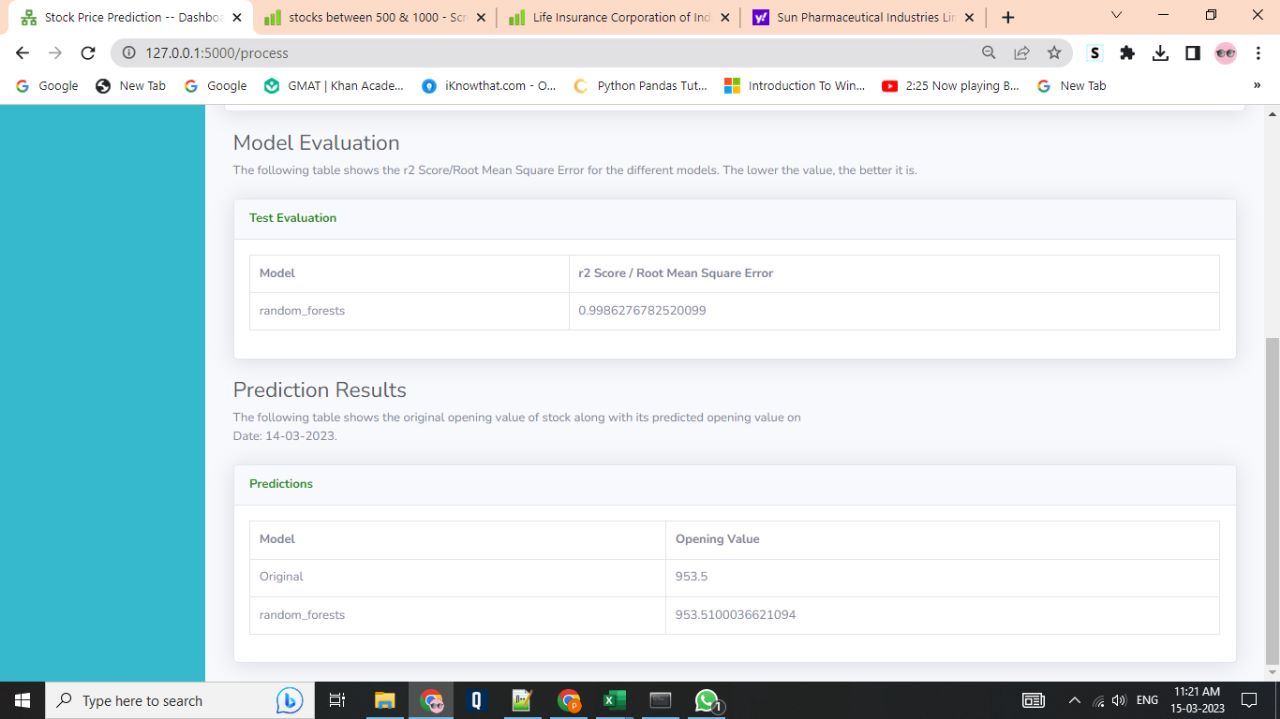
**OUR FINDINGS BY APPLYING VARIOUS ALGORITHMS**

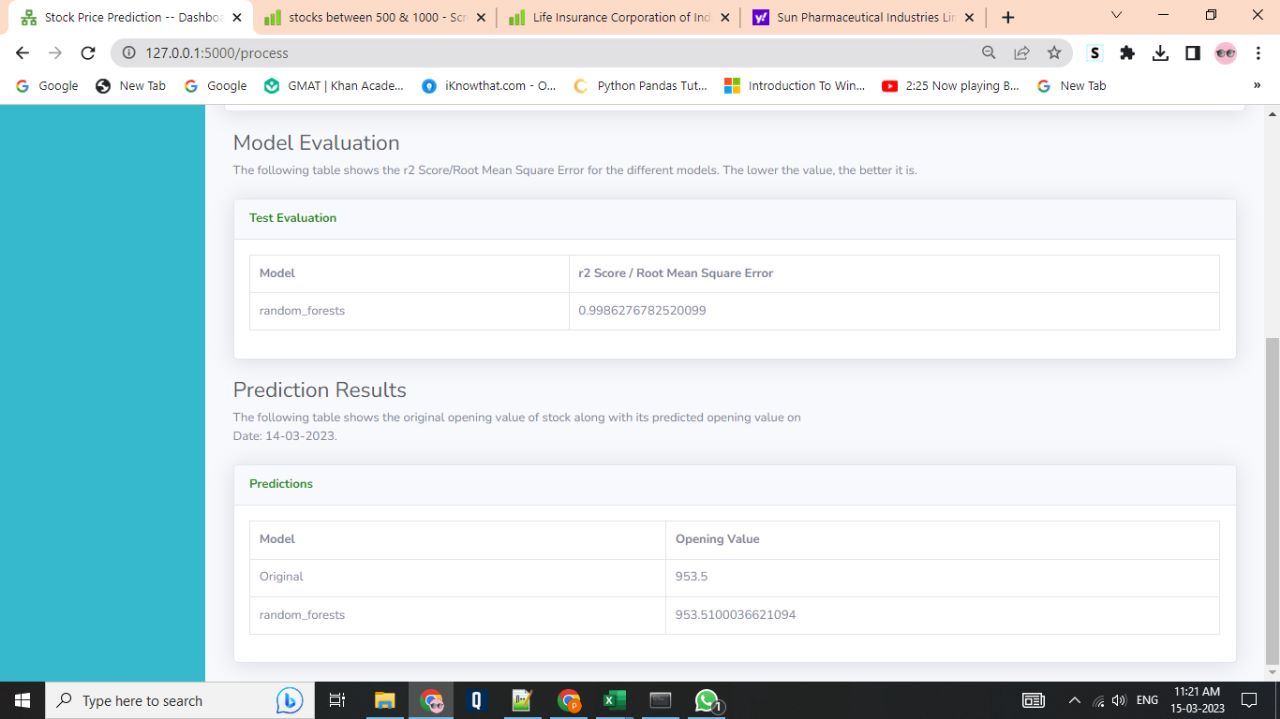
1. Linear Regression:

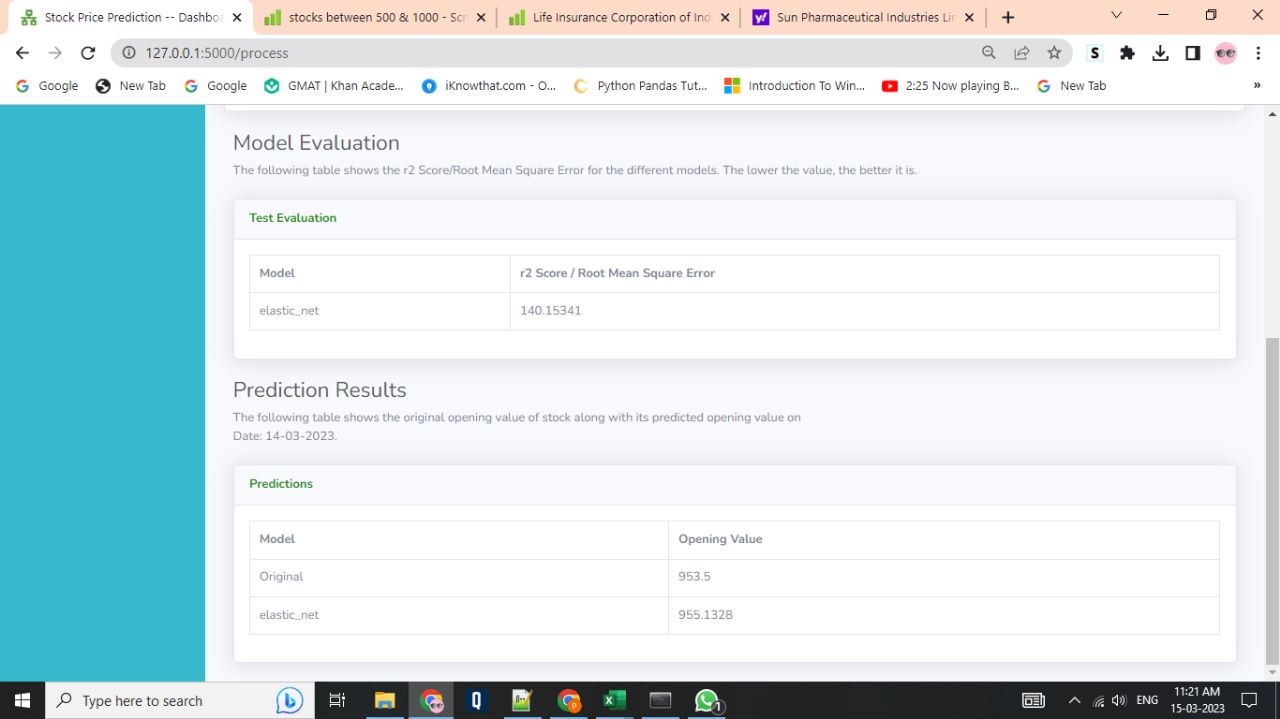


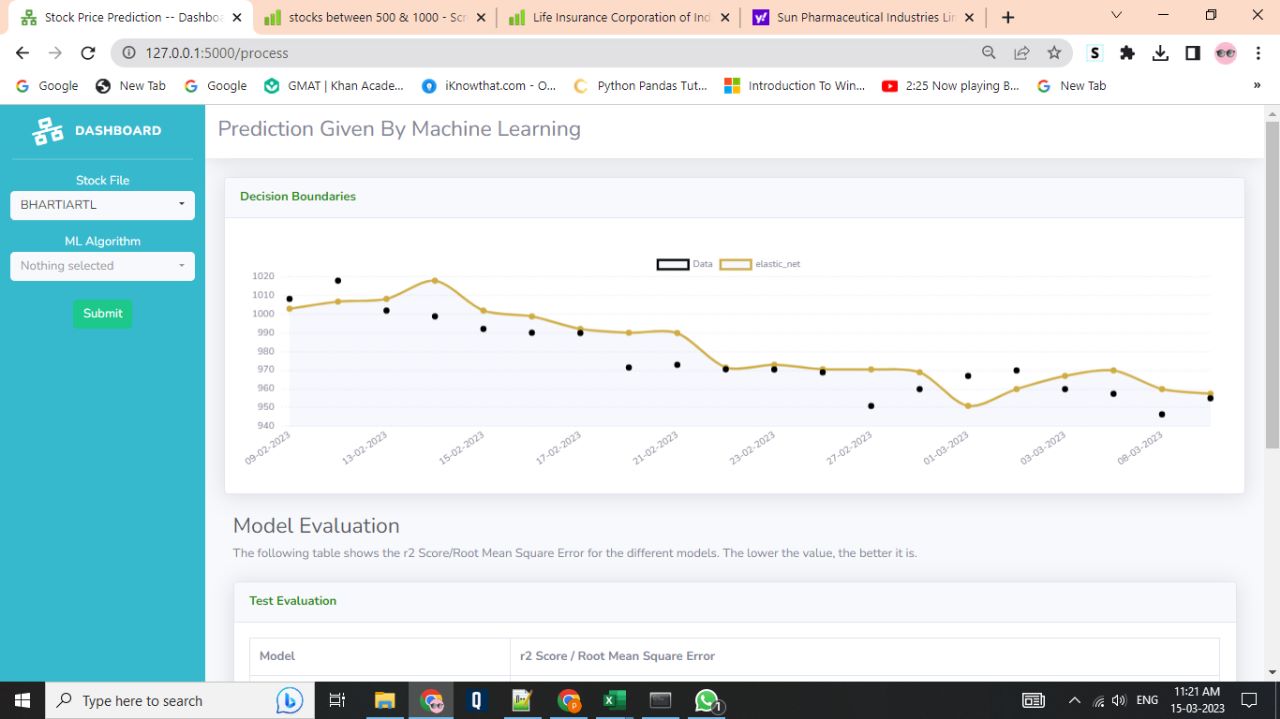


2. Random Forest:

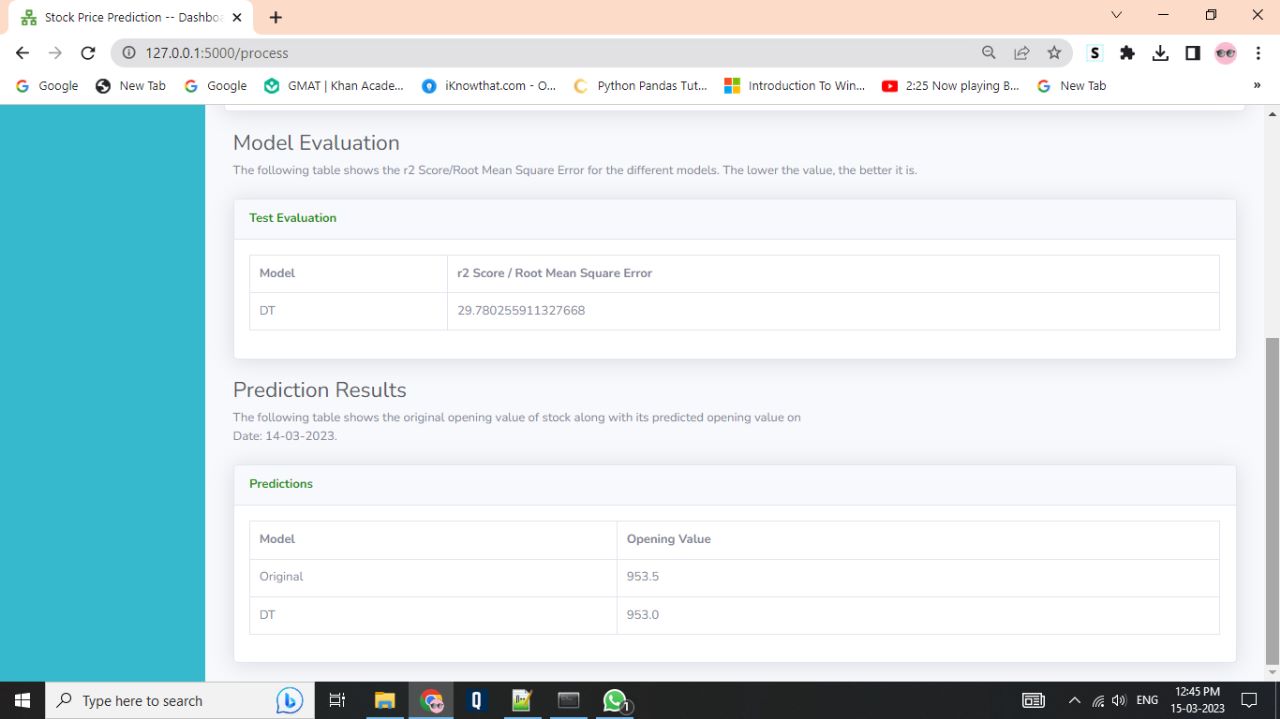


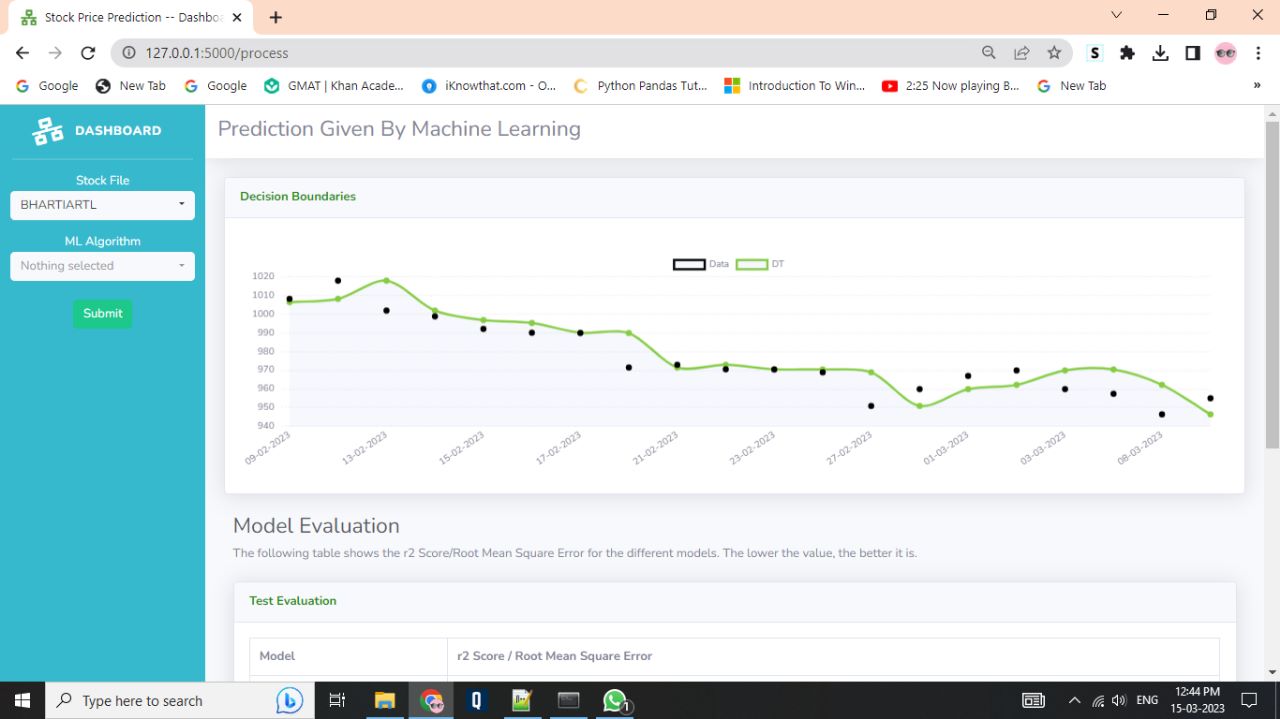
3. Elastic-net:

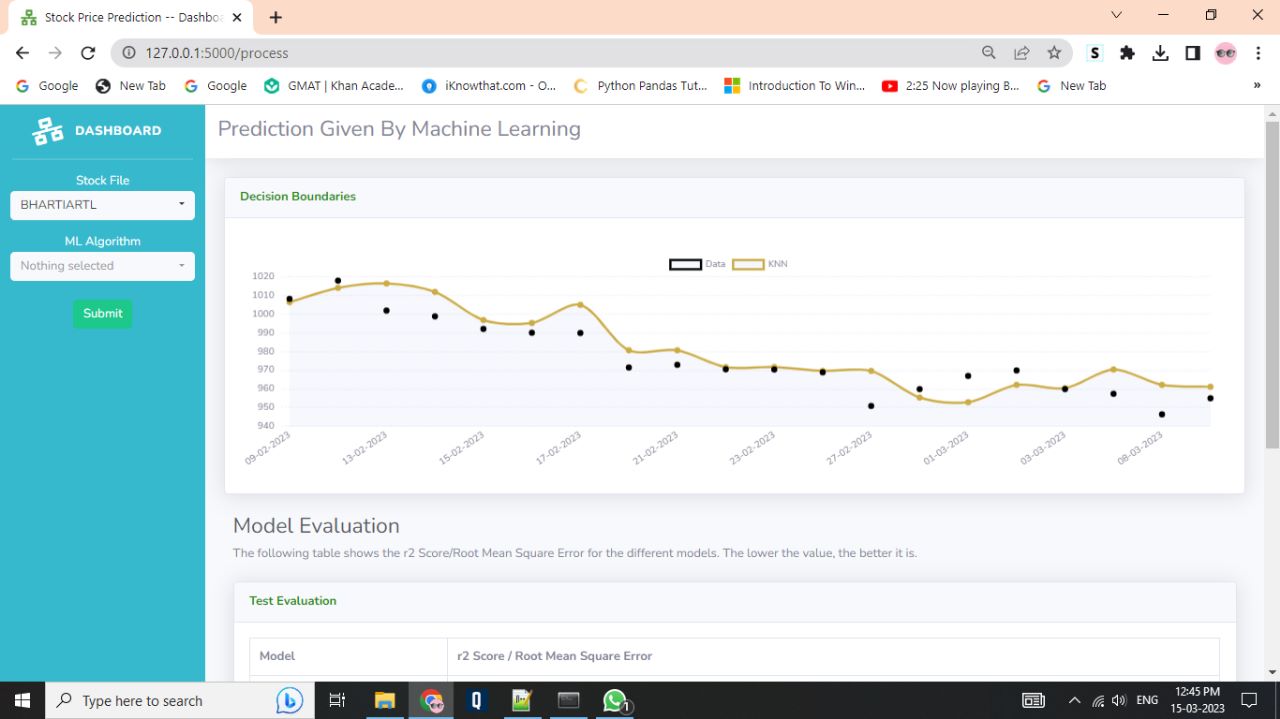
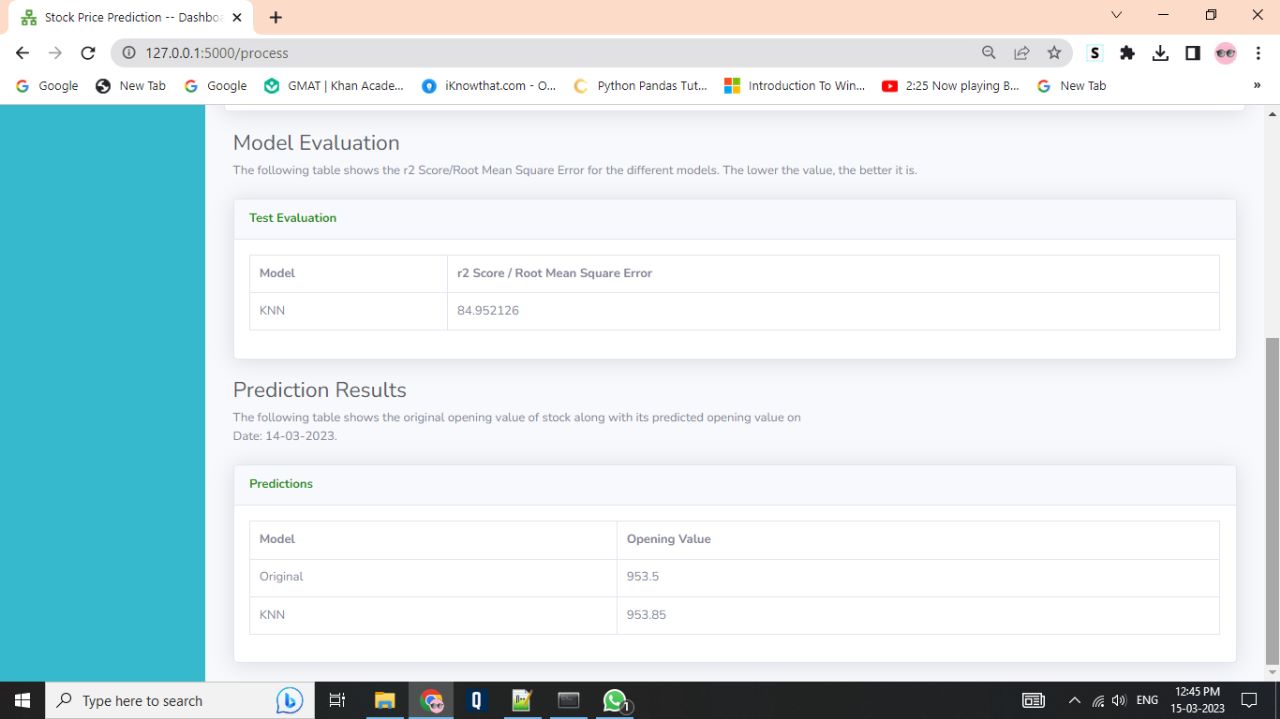




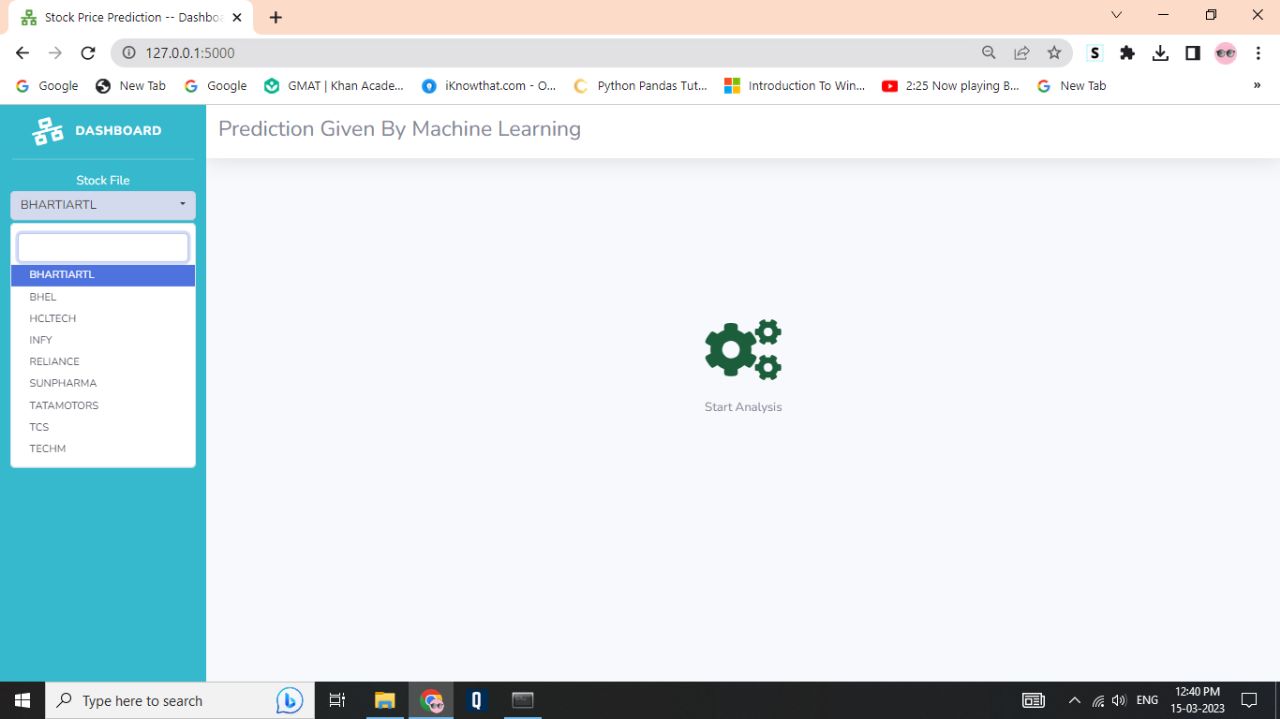
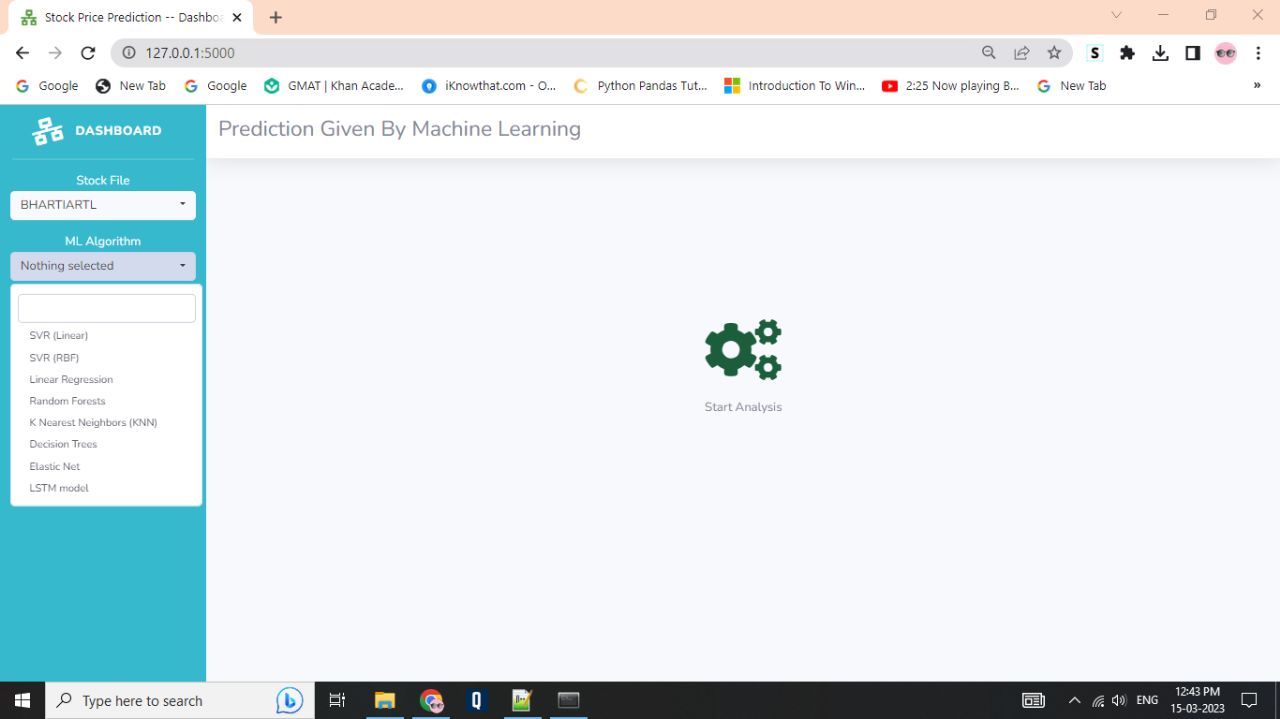
4.Decision Tree:



5. KNN:



STEP – III : Use of FLASK to visualize the data that we have analyzed using Machine Learning Algorithms. In this application the user can select the stock name for which he/she wishes to check the stock prices along with the Algorithm he / she wishes to apply to the stock for further prediction.



**Result and Finding**

|  |  |
| --- | --- |
| **ALGORITHM** | **Root Mean Squared Error** |
| **SVR-rbf** | **95.646** |
| **Linear Regression** | **0.9963** |
| **Random Forest** | **0.9986** |
| **KNN** | **84.952** |
| **DT** | **29.780** |
| **Elastic Net** | **140.153** |

1. **CONCLUSION AND FUTURE SCOPE**

To further improve the prediction accuracy, more variability needs to be considered and modeled. Since stock prices are very prone to speculations and respond these speculations with sharp increases and decreases in their prices, Sentiment analysis can also be conducted for detecting increase or decrease of the prices. Last part that can be improved is data size. For overcoming this problem, the prices of each day can be enhanced with splitting prices hourly.

For the future studies, research will focus on, modified LSTM layers, adding dropout and modified number of epochs, and using different instability dataset to test how good the prediction results or try to use sentiment analysis combined with LSTM method to see the impact of the uncertainty in value of the Stock

# References

1. [**https://www.sciencedirect.com/science/article/pii/S1877050920307924**](https://www.sciencedirect.com/science/article/pii/S1877050920307924)
2. [**https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6**](https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6)
3. [**https://jfin-swufe.springeropen.com/articles/10.1186/s40854-019-0131-7**](https://jfin-swufe.springeropen.com/articles/10.1186/s40854-019-0131-7)