**STOCK TREND PREDICTION WEB APPLBICATION**

**IN PYTHON USING STREAMLIT**

**PROJECT DESIGN**

**Eshita Pradhan Riya Yadav**

**CSE Branch CSE Branch**

**Roll No. : 211000018 Roll No.: 211000047**

**Email address :** [**eshita21100@iiitnr.edu.in**](mailto:eshita21100@iiitnr.edu.in) **Email address:** [**riya21100@iiitnr.edu.in**](mailto:riya21100@iiitnr.edu.in)

**Sonali Tudu**

**CSE Branch**

**Roll No. : 211000056**

**Email address :** [**sonalit21100@iiitnr.edu.in**](mailto:sonalit21100@iiitnr.edu.in)

1**. REQUIREMENT ANALYSIS :**

The requirement that the system needs is categorized into :

i)Functional Requirements.

ii) Non-Functional Requirements

1.1 **Functional Requirements -**

Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows:

1. The system should be able to generate an approximate share price.

2. The system should collect accurate data from the stock market website in consistent manner.

3. The prediction shall abide by the following functional requirements.

4. Prior to application of stock recommendations, the database is updated by the latest values.

5.The charts and comparison of the companies would be done only on the latest data stock market data.

6. The user can look previous data Information which was collected.

7. The user can also be recommended on the basis of the trending stocks which would require the data regarding the stocks.

1.2 **Non-Functional Requirements -**

Non-functional requirement is a description of features. characteristics and attribute of the system as well as any constraints that may limit the boundaries of the proposed system. The non- functional requirements are essentially based on the performance, information, economy, control and security efficiency and services.

Based on these the non-functional requirements are as follows:

1. The system should provide better accuracy.

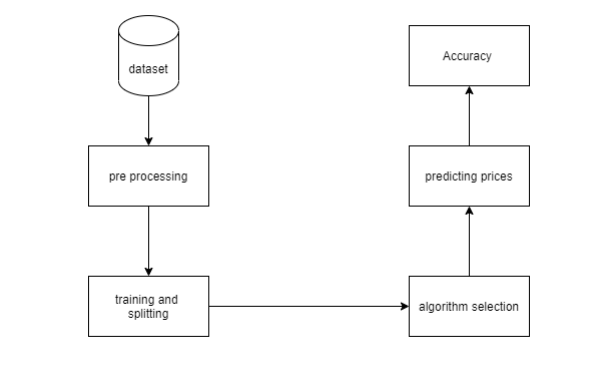
2. The system should have simple interface for users to use.

3. To perform efficiently in short amount of time.

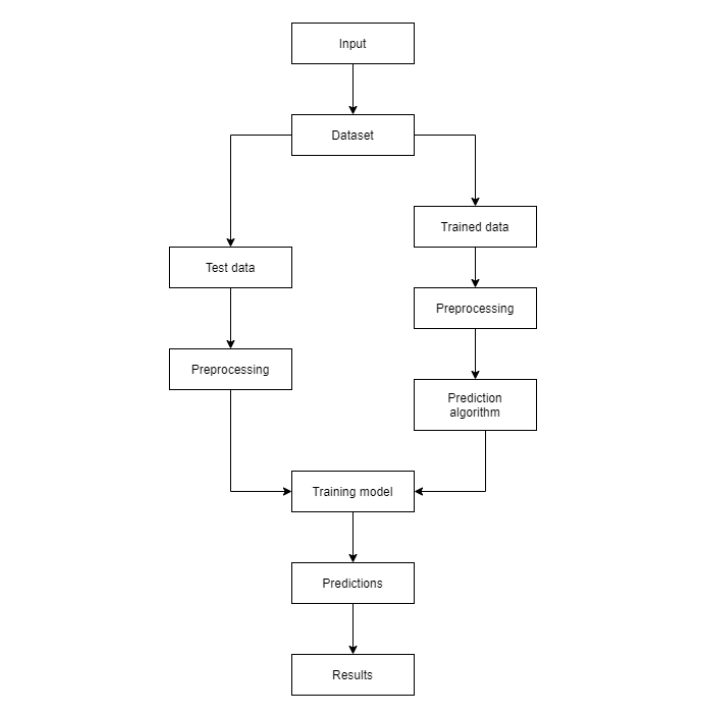
2. **STORYBOARD :**

* **Done** Multiply Svg Png Icon - Cross Sign Clipart Black And White, Transparent Png  - kindpng **Not done yet**
* First, we will learn how to predict stock price using the LSTM neural network**.**
* Then we will build a web application using Streamlit for stock analysis.
* Launch a jupyter notebook through Anaconda navigator and ensure the environment is up to date with all libraries installed properly.
* Upload the dataset
* Analyze the closing prices from dataset
* Sort the dataset on date time and filter “Date” and “Close” columns:
* Normalize the new filtered dataset
* Build and train the LSTM model
* Save your model
* Make a web application using Streamlit
* Make a dashboard for stock analysis

2.1 **Components present**



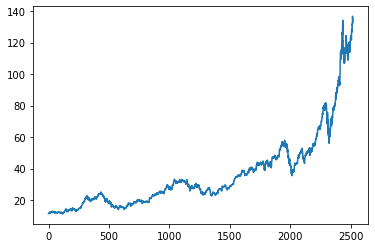
2.2 **Structure chart**



3**.DESIGN OF USER INTERFACE :**

**Apple Inc. (AAPL)**

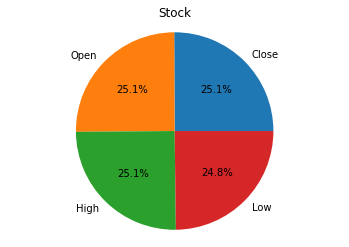
3.1 **Graph of dataset from 2011-2020**



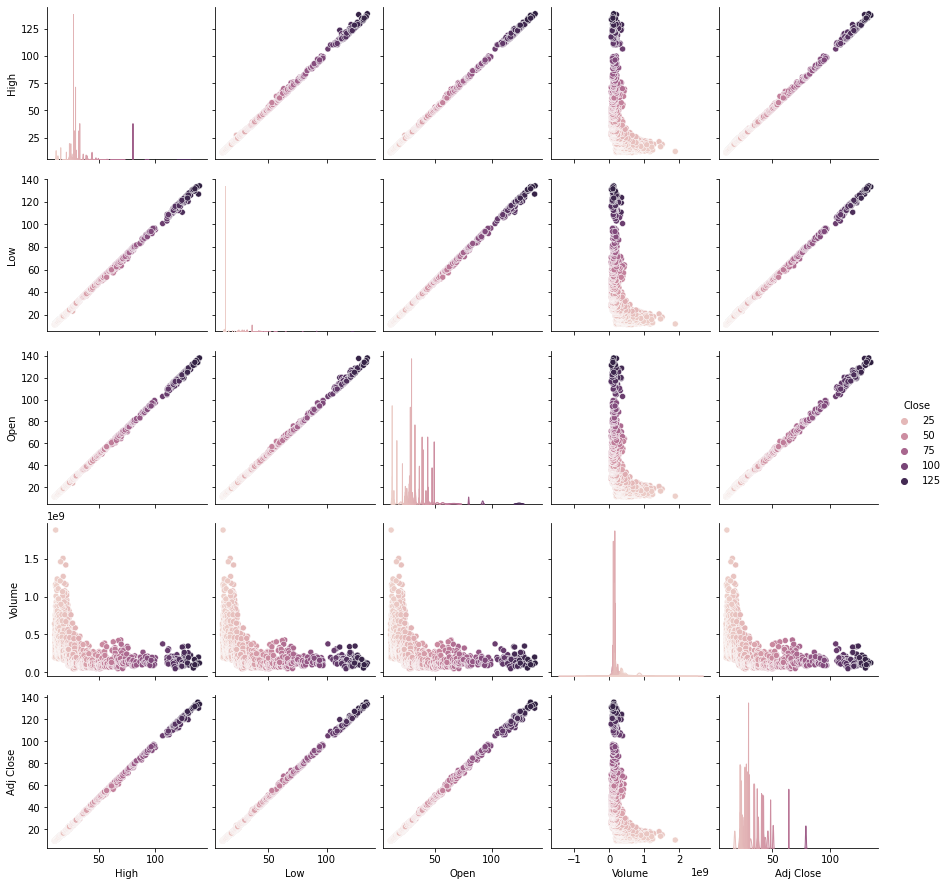
3.2 **Historical data**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Year** | **Sum of Close** | | | **Count of Date** | | **Sum of High** | | **Sum of Low** | | **Sum of Open** | |
| **2011** | | **3276.038932** | **252** | | **3306.812152** | | **3242.679282** | | **3276.552854** | |
| **2012** | | **5143.30108** | **250** | | **5194.869636** | | **5088.581776** | | **5148.684999** | |
| **2013** | | **4253.713923** | **252** | | **4298.750363** | | **4214.223942** | | **4258.153202** | |
| **2014** | | **5812.673213** | **252** | | **5859.797866** | | **5762.958925** | | **5809.849308** | |
| **2015** | | **7562.519994** | **252** | | **7638.404999** | | **7488.375004** | | **7571.072498** | |
| **2016** | | **6590.052501** | **252** | | **6641.912506** | | **6532.487505** | | **6583.989993** | |
| **2017** | | **9447.079986** | **251** | | **9500.730008** | | **9380.350026** | | **9440.417506** | |
| **2018** | | **11863.10253** | **251** | | **11984.88001** | | **11745.76502** | | **11866.74253** | |
| **2019** | | **13120.12499** | **252** | | **13219.39752** | | **12995.1125** | | **13095.75247** | |
| **2020** | | **23990.11995** | **252** | | **24320.68749** | | **23628.21745** | | **23968.64005** | |
| **Grand Total** | | **91058.7271** | **2516** | | **91966.24254** | | **90078.75143** | | **91019.8554** | |
|  | |  |  | |  | |  | |  | |
|  | |  |  | |  | |  | |  | |

3.3 **Pie chart of Historical data (grand total values)**

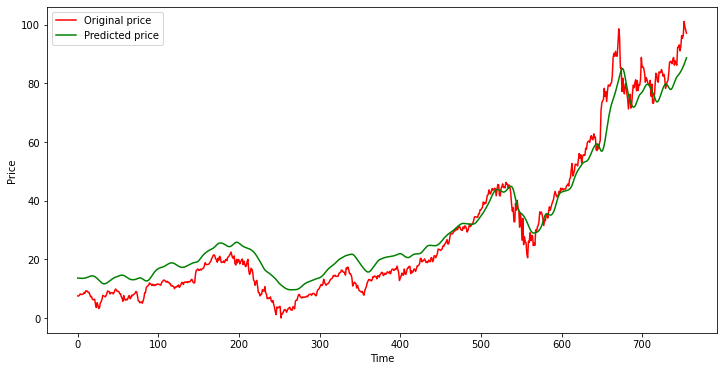


3.4 **Graph according to historical data**



**Link to the pie chart and dashboard:** <http://localhost:8890/notebooks/Downloads/Dashboard%20aapl.ipynb>

3.5 **Comparison between Original and Predicted price**



4. **ATTRIBUTES :**

**JupyterLab :** is the latest web-based interactive development environment for notebooks, code, and data.

Python libraries used :

**Ipynb -** An IPYNB file is a notebook document used by Jupyter Notebook, includes the inputs and outputs of computations, mathematics, machine learning, images, and more.

**Numpy -** NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

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

**matplotlib.pyplot** - It provides an implicit, MATLAB-like, way of plotting. It also opens figures on your screen, and acts as the figure GUI manager ,pyplot is mainly intended for interactive plots and simple cases of programmatic plot generation.

**pandas\_datareader -** is a Python package that allows us to create a pandas DataFrame object by using various data sources from the internet. It is popularly used for working with realtime stock price datasets.

**sklearn.preprocessing -** sklearn. preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators. In general, learning algorithms benefit from standardization of the data set.

**MinMaxScaler -** Transform features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.

**keras.layers -** functional building blocks of Keras Models. Each layer is created using numerous layer\_() functions. These layers are fed with input information, they process this information, do some computation and hence produce the output. Further, this output of one layer is fed to another layer as its input.

**keras.models -** is tightly integrated with TensorFlow, which is used to build machine learning models. Keras' models offer a simple, user-friendly way to define a neural network, which will then be built for you by TensorFlow.

**Streamlit -** is an open-source Python library that makes it easy to create and share custom web apps for machine learning and data science.