SWE4010-Artificial Intelligence

(EmbeddedProject)

AIbasedStockPredictionSystemusingTataGlobal Beverages Data

REVIEW-3

\

SubmittedTo:

Prof TAPAN KUMAR DAS
Associate Professor Grade 2
Schoolof Information Technology andEngineering
VIT

SubmittedBy:

R.SaiKiran-17MIS0422
V.Yashwanth Reddy -17MIS0427
G.Harshavardhan17MIS0454
M.SaiSatyaHarin-17MIS0518
Programme:IntegratedMTECH

Abstract:

In the finance world stock trading is one of the most important activities.AI based Stock price prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. The prediction of a stock market price may serve as an early recommendation system for short-term investors and as an early financial distress warning system for long-term shareholders. Forecasting accuracy is the most important factor in selecting any forecasting methods. Research efforts in improving the accuracy of forecasting models are increasing since the last decade. The appropriate stock selections those are suitable for investment is a very difficult task. The key factor for each investor is to earn maximum profits on their investments.

Introduction:

An AI based stock price prediction in market depicts savings and investments that are advantageous to increase the effectiveness of national economy. The future stock returns have some predictive relationships with the publicly available information of present and historical stock market indices. The investors decide the better time to sell/buy hold a share in stock market based on the former relationship. Financial investors of today are facing this problem of trading us they do it properly understand as to which stocks to buy or which stocks to sell in order to get optimum results. This possesses a major challenge to design and develop an effective and efficient predictive model that assists investors to take appropriate decisions. So, the proposal project will reduce the problem with suitable accuracy faced in such real time scenario.

An AI based Stock price prediction is a classic and important problem. With a successful model for stock prediction, we can gain insight about market behaviour over time, spotting trends that would otherwise not have been noticed. With the increasingly computational power of the computer, machine learning will be an efficient method to solve this problem. However, the advancement in technology, provides an opportunity to gain steady fortune from stock market and also can help experts to find out the most informative indicators to make better prediction. The prediction of the market value is of paramount importance to help in maximizing the profit of stock option purchase while keeping the risk low.

Proposed Model:

Accuracy plays an important role in stock market prediction. Although many algorithms are available for this purpose, selecting the most accurate one continues to be the fundamental task in getting the best results. We have taken Decision Tree Regression algorithm for this stock price prediction in our system. This algorithm allows us to categorize clearly based on previous data of companies over the years. Firstly, in this we have to train the data based on the predefined dataset containing companies' historical data like turnover etc. these all things will be maintained by the admin. Later user will test the data by giving his inputs. The system is here able to provide accurate results.

We have developed this using Machine learning and Python using Django framework. The aim of this project is to predict supervised machine learning algorithms to predict Stock price prediction, data processing has been done and split the data with trained data and fit the values and finally predict the output result. This is also intended to solve the economic dilemma created in individuals that wants to invest in Stock Market.

Advantages of Proposed Model:

- The successful prediction will maximize the benefit of the customer
- Helps the users in detecting the market trend patterns and other conditions.
- Easy to predict
- Time management is done
- System will have a good flexibility
- Easy to understand
- Cost effectiveness will be provided

System Specification

Software Requirements

Functional requirements for a secure cloud storage service are straightforward:

1. The service should be able to store the user’s data;

2. The data should be accessible through any devices connected to the Internet;

3. The service should be capable to synchronize the user’s data between multiple devices (notebooks, smart phones, etc.);

4. The service should preserve all historical changes (versioning);

5. Data should be shareable with other users;

6. The service should support SSO; and

7. The service should be interoperable with other cloud storage services, enabling data migration from one CSP to another.

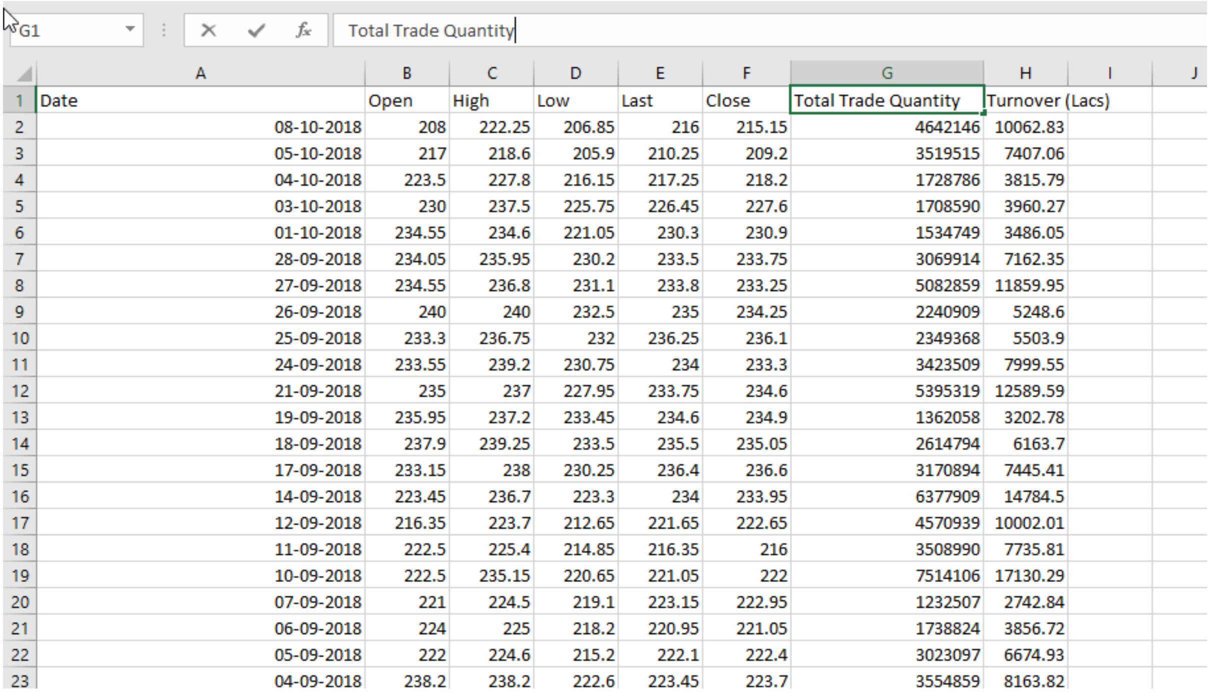
● Operating System	: Windows XP/7/10
● Programming Language	: Python 3x
● Front End or web Technologies	: HTML,CSS,BOOTSTRAP
● Web Frameworks	: Django 2x

Hardware Requirements:

• Micro Processor Type	- Core i3/Core i5
• Micro Processor	- 2.0 GHz
• RAM	- 4GB / 8GB for faster output.
• Hard Disk	- 500 GB

Dataset:

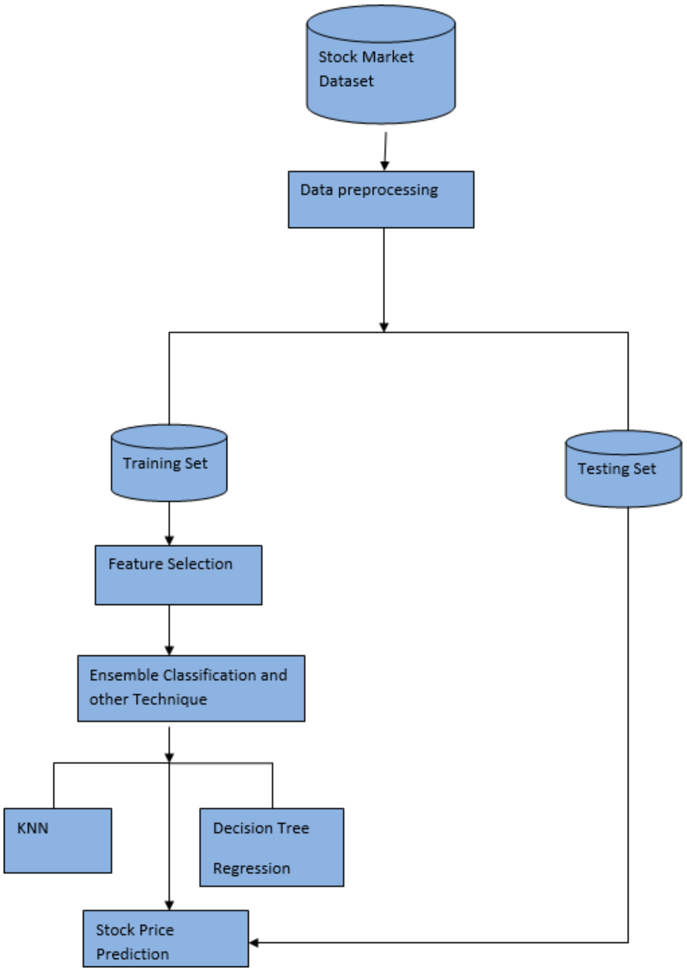
https://www.quandl.com/data/NSE/TATAGLOBAL-Tata-Global-Beverages-Limited
We’ll be using a dataset from Quandl (where we can find historical data for various stocks here) and for this particular project, we have been used the data for ‘Tata Global Beverages’. This dataset contains 8 attributes they are Date, Open, High, Low, Last, Close, Total Trade Quantity and Turnover(Lacs). Turnover is the Class Attribute in this data set.



	A	B	C	D	E	F	G	H	I	J
1	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)		
2	08-10-2018	208	222.25	206.85	216	215.15	4642146	10062.83		
3	05-10-2018	217	218.6	205.9	210.25	209.2	3519515	7407.06		
4	04-10-2018	223.5	227.8	216.15	217.25	218.2	1728786	3815.79		
5	03-10-2018	230	237.5	225.75	226.45	227.6	1708590	3960.27		
6	01-10-2018	234.55	234.6	221.05	230.3	230.9	1534749	3486.05		
7	28-09-2018	234.05	235.95	230.2	233.5	233.75	3069914	7162.35		
8	27-09-2018	234.55	236.8	231.1	233.8	233.25	5082859	11859.95		
9	26-09-2018	240	240	232.5	235	234.25	2240909	5248.6		
10	25-09-2018	233.3	236.75	232	236.25	236.1	2349368	5503.9		
11	24-09-2018	233.55	239.2	230.75	234	233.3	3423509	7999.55		
12	21-09-2018	235	237	227.95	233.75	234.6	5395319	12589.59		
13	19-09-2018	235.95	237.2	233.45	234.6	234.9	1362058	3202.78		
14	18-09-2018	237.9	239.25	233.5	235.5	235.05	2614794	6163.7		
15	17-09-2018	233.15	238	230.25	236.4	236.6	3170894	7445.41		
16	14-09-2018	223.45	236.7	223.3	234	233.95	6377909	14784.5		
17	12-09-2018	216.35	223.7	212.65	221.65	222.65	4570939	10002.01		
18	11-09-2018	222.5	225.4	214.85	216.35	216	3508990	7735.81		
19	10-09-2018	222.5	235.15	220.65	221.05	222	7514106	17130.29		
20	07-09-2018	221	224.5	219.1	223.15	222.95	1232507	2742.84		
21	06-09-2018	224	225	218.2	220.95	221.05	1738824	3856.72		
22	05-09-2018	222	224.6	215.2	222.1	222.4	3023097	6674.93		
23	04-09-2018	238.2	238.2	222.6	223.45	223.7	3554859	8163.82		

System Design:

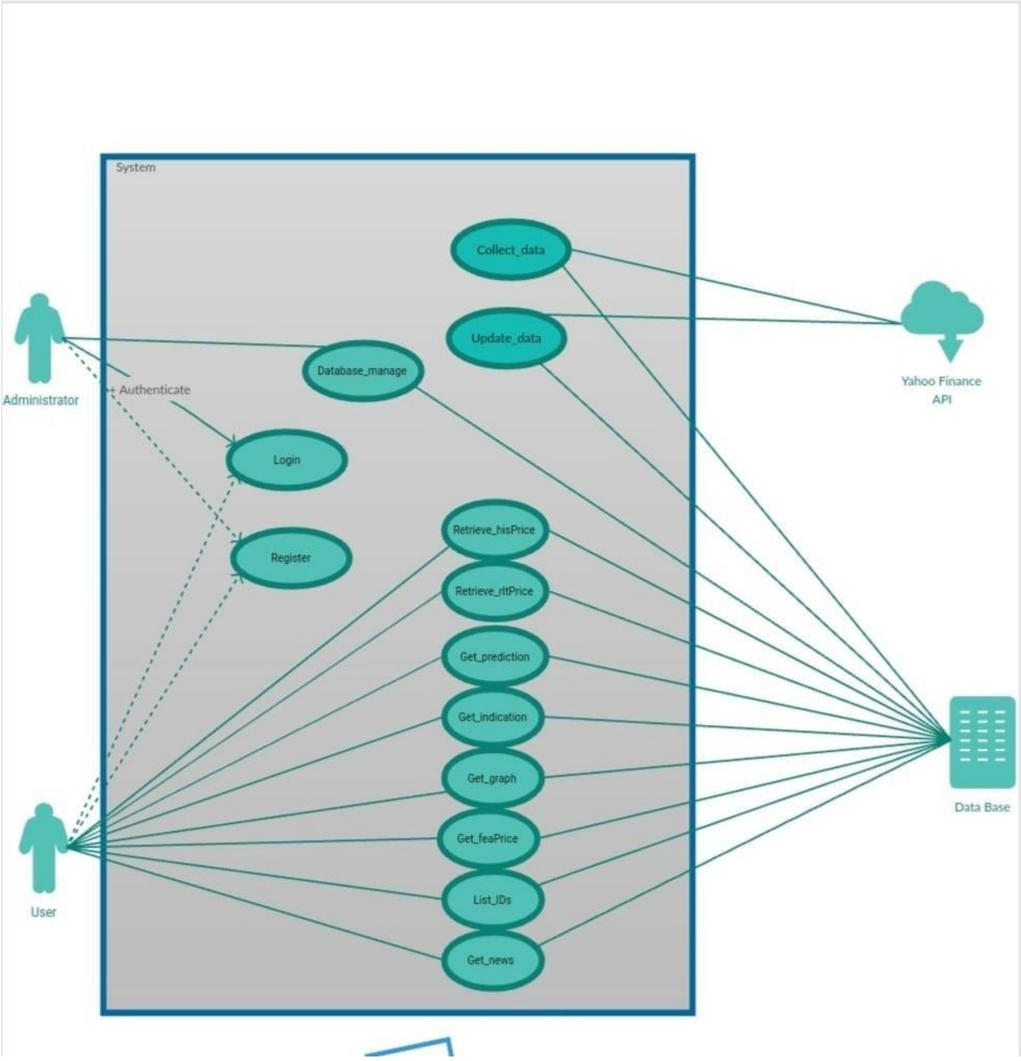
System Architecture



USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphicaloverview of the functionality provided by a system in terms of actors, their goals

(represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor.Roles of the actors in the system can be depicted.


System
Collect_data
Update_data
Database_manage
Login
Register
Retrieve_hisPrice
Retrieve_rltPrice
Get_prediction
Get_indication
Get_graph
Get_feaPrice
List_IDs
Get_news
Administrator
Authenticate
User
Yahoo Finance API
Data Base

IMPLEMENTATION:

views.py

```
from django.shortcuts import render, redirect
#from .models import DoctorReg, predictions,
Regdbfrom django.contrib import messages
from django.contrib.auth.models import User,
authimport pandas as pd
from sklearn.model_selection import
train_test_splitfrom sklearn.tree import
DecisionTreeRegressor
# Create your views
here.def
index(request):
    return render(request,
'index.html')def register(request):
    if request.method == 'POST':
        first_name =
        request.POST['first_name']last_name
        = request.POST['last_name']
        username = request.POST['username']
        password1 =
        request.POST['password1']password2
        = request.POST['password2']email =
        request.POST['email']
 if password1 == password2:
                    if
                User.objects.filter(username=username).exists():
                messages.info(request, 'Username Taken')
                return redirect('register')
```

```
            elif
                User.objects.filter(email=email).exists(
                ):messages.info(request, 'Email Taken')
                return redirect('register')
            else:
                user = User.objects.create_user(username=username,
  password=password1,email=email,


first_name=first_name,
                last_name=last_name)
                user.save();
                print('user created')
                return
                redirect('login')


        else:
            messages.info(request, 'password not
            matching')return redirect('register')
        return
    redirect('/')else:
        return render(request,
  'register.html')def login(request):
    if request.method ==
        'POST': #v =
        DoctorReg.objects.all()
        username =
        request.POST['username']password
        = request.POST['password']
    user = auth.authenticate(username=username, password=password)
```

```
        if user is not None:
            auth.login(request, user)

            return render(request,

        'data.html')else:

            messages.info(request, 'invalid

            credentials')return redirect('login')

    else:

        return render(request, 'login.html')


def data(request):

    return

render(request,"data.html")def

predict(request):

    if (request.method ==

        'POST'): open =

        request.POST['open']high

        = request.POST['high']

        low= request.POST['low']

        last = request.POST['last']

        close =

        request.POST['close']

        trade=request.POST['trade

        ']

        df =

        pd.read_csv(r"static/datasets/Stock.csv")

        df.dropna(inplace=True)

        df.isnull().sum()

        X_train = df[['Open','High','Low','Last','Close','Total Trade

        Quantity']]Y_train = df[['Turnover (Lacs)']]

        tree =

        DecisionTreeRegressor()
        tree.fit(X_train, Y_train)

        prediction = tree.predict([[open,high,low,last,close,trade]])
        return render(request, 'predict.html',

                        {"data": prediction, 'open': open,  'high': high,

                         'close': close, 'last': last,"low":low,'trade':trade

                         })

    else:

        return render(request,

'predict.html')def logout(request):

    return render(request,"logout.html")
```

url.py :

```
from django.urls import

pathfrom .import views

urlpatterns=[

    path('',views.index,name='index'),

    path('register',views.register,name='register'),

    path('login',views.login,name='login'),

    path('data',views.data,name='data'),

    path('predict',views.predict,name='predict'),

    path('logout',views.logout,name='logout')

]
```

settings.py :

```
"""

Django settings for Stockpred project.


Generated by 'django-admin startproject' using Django

3.0.7.For more information on this file, see

https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
```

```
https://docs.djangoproject.com/en/3.0/ref/setting

s/"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__

)))# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/


# SECURITY WARNING: keep the secret key used in production secret!

SECRET_KEY = '4g4hfk=r&)3ti(4i-

kug_bwyu*++bajhqh8gdi1vmb*38(!1^('


# SECURITY WARNING: don't run with debug turned on in

production!DEBUG = True


ALLOWED_HOSTS = []

# Application

definition

INSTALLED_APPS

= [

    'User.apps.UserConfig',

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttyp

    es',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'Stockpred.urls'

TEMPLATES = [
    {
        'BACKEND':
        'django.template.backends.django.DjangoTemplates','DIRS':
        [os.path.join(BASE_DIR,'templates')],
        'APP_DIRS':
        True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages'
                ,
            ],
        },
    },
]
WSGI_APPLICATION =
'Stockpred.wsgi.application'# Database
```

```
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases


DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': os.path.join(BASE_DIR,

        'db.sqlite3'),

    }

}

# Password validation

# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators


AUTH_PASSWORD_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',

    },

    {
        'NAME':
        'django.contrib.auth.password_validation.NumericPasswordValidator',

    },

]

# Internationalization

#

https://docs.djangoproject.com/en/3.0/topics/i18n

/LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'

USE_I18N      =

True

USE_L10N      =

True USE_TZ  =

True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.0/howto/static-

files/STATIC_URL = '/static/'

STATICFILES_DIRS=[

    os.path.join(BASE_DIR,'static')

]
```

<u>manage.py</u> :

```
#!/usr/bin/env python
"""Django's command-line utility for administrative
tasks."""import os
import sys



def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',
    'Stockpred.settings')try:
        from django.core.management import
    execute_from_command_lineexcept ImportError as exc:

        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did
            you ""forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

except ImportError as
```

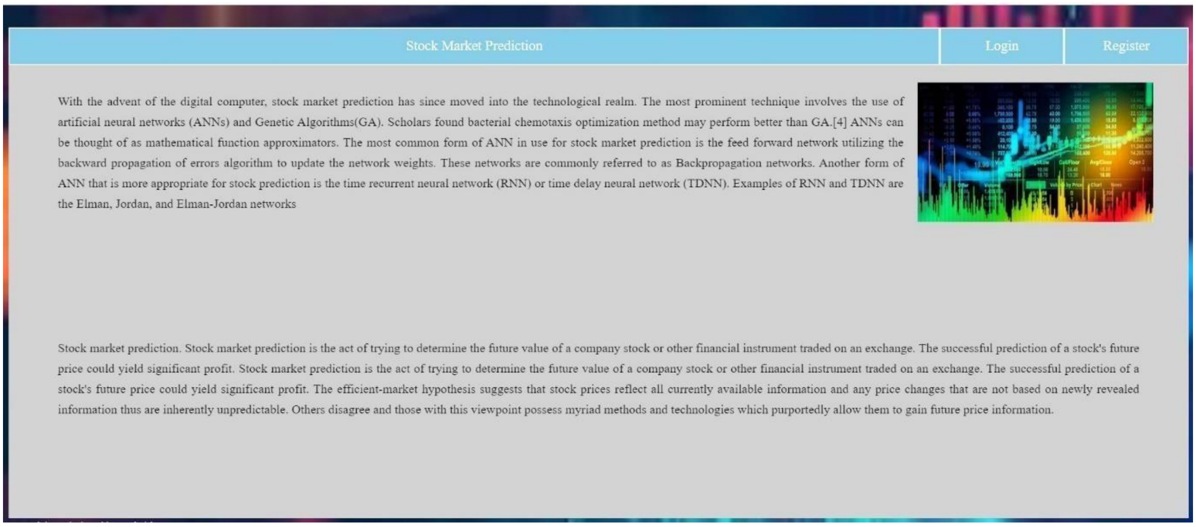
```
exc:raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did
            you ""forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if___name___== '_main_

':main()
```

Results:

I) It is the home page of our stop price prediction system, in this a new user can register by givingdetails and login through the given details. And also have a brief info about stock price prediction.

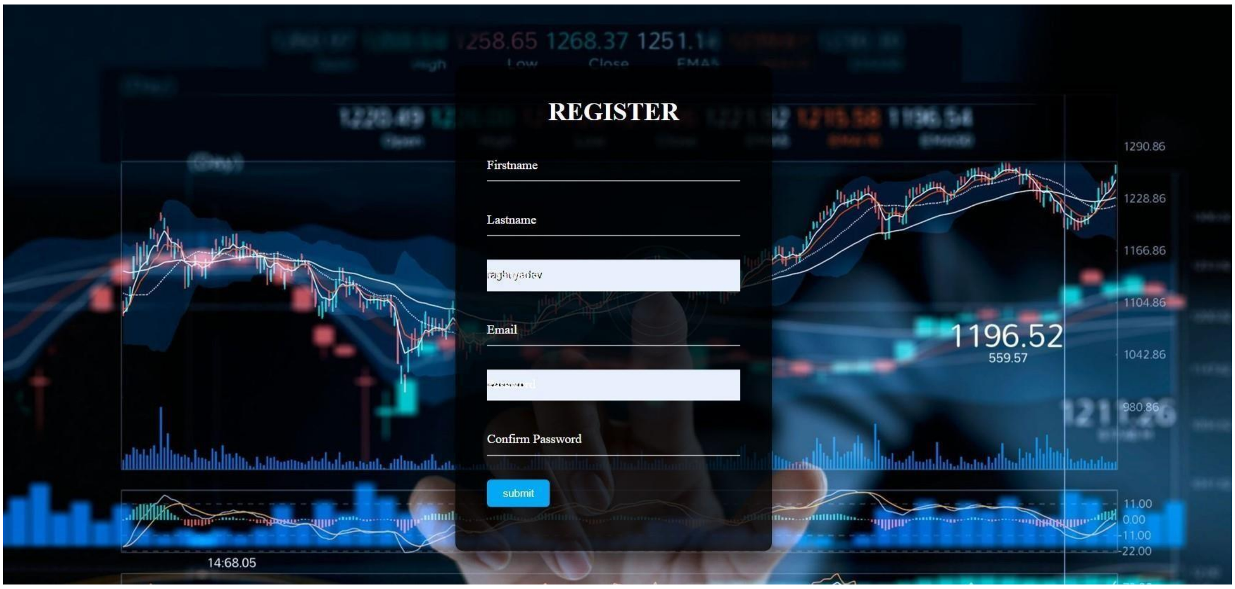


With the advent of the digital computer, stock market prediction has since moved into the technological realm. The most prominent technique involves the use of artificial neural networks (ANNs) and Genetic Algorithms(GA). Scholars found bacterial chemotaxis optimization method may perform better than GA.[4] ANNs can be thought of as mathematical function approximators. The most common form of ANN in use for stock market prediction is the feed forward network utilizing the backward propagation of errors algorithm to update the network weights. These networks are commonly referred to as Backpropagation networks. Another form of ANN that is more appropriate for stock prediction is the time recurrent neural network (RNN) or time delay neural network (TDNN). Examples of RNN and TDNN are the Elman, Jordan, and Elman-Jordan networks

Stock market prediction. Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. The efficient-market hypothesis suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable. Others disagree and those with this viewpoint possess myriad methods and technologies which purportedly allow them to gain future price information.

Home page

II) It is the Registration page through which a new user can register into the application by givinghis name, mobile number, E-mail, etc.



Firstname

Lastname

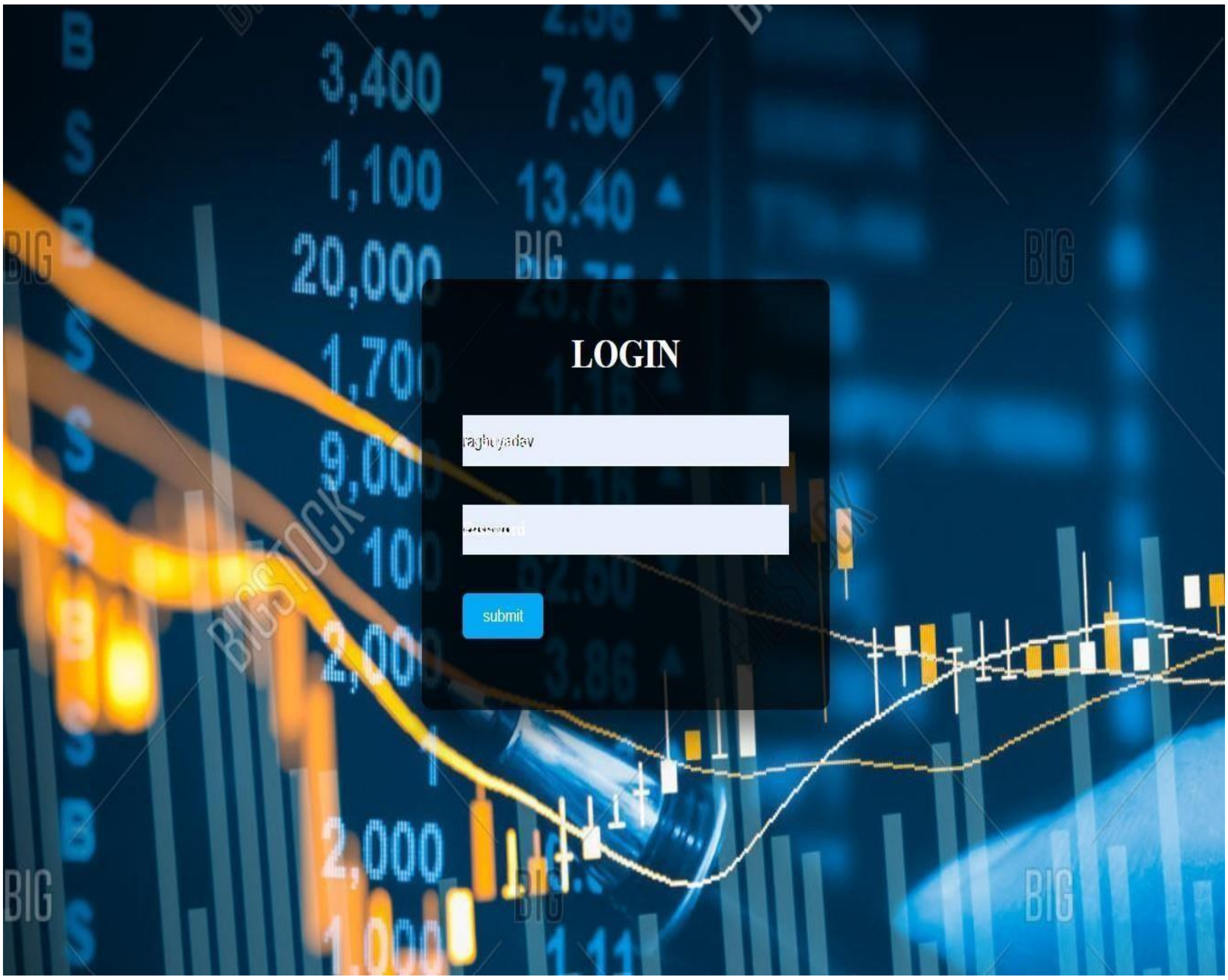
raghuyadav

Email

Confirm Password

Registration page

III) This is login page, user can log in thorough giving username and password.And later click on submit button.

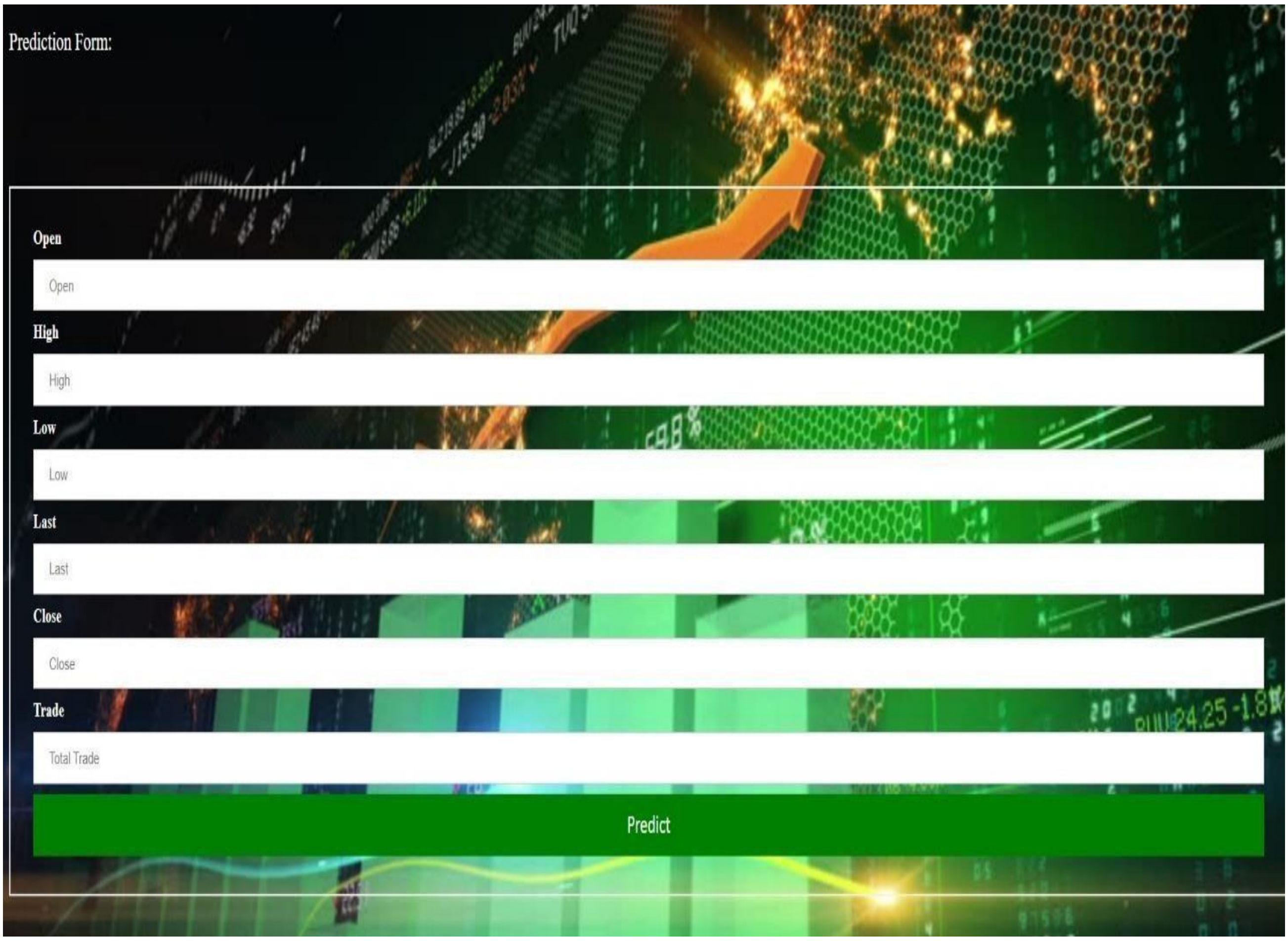


LOGIN

raghuyadav

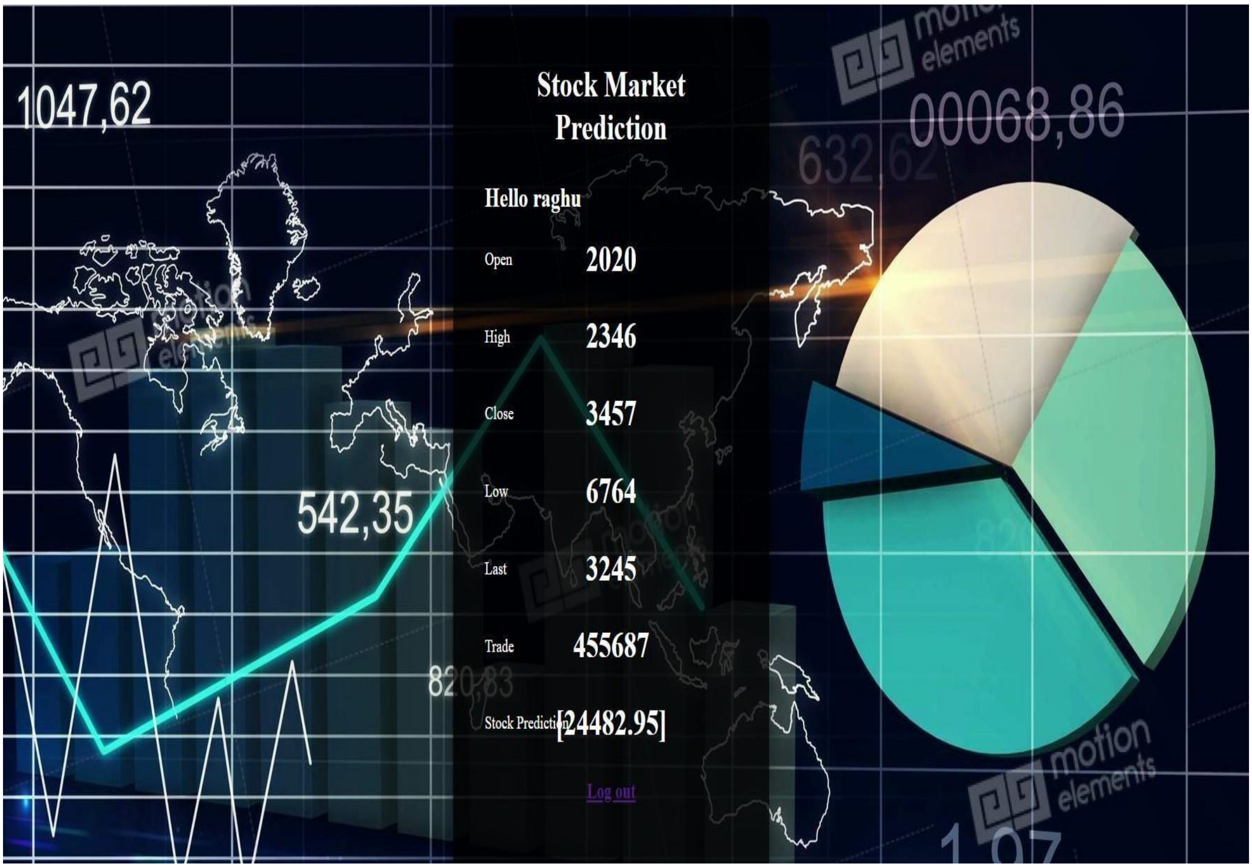
Login page

IV) This is prediction form page where we give the inputs such as open, high, low,last, close, trade for the prediction.



Prediction input page

V) In this page it shows final stock prediction based on inputs we have given tothe system by comparing the inputs with the predefined dataset which some historical data.



Stock Market Prediction

Hello raghu

Open 2020

High 2346

Close 3457

Low 6764

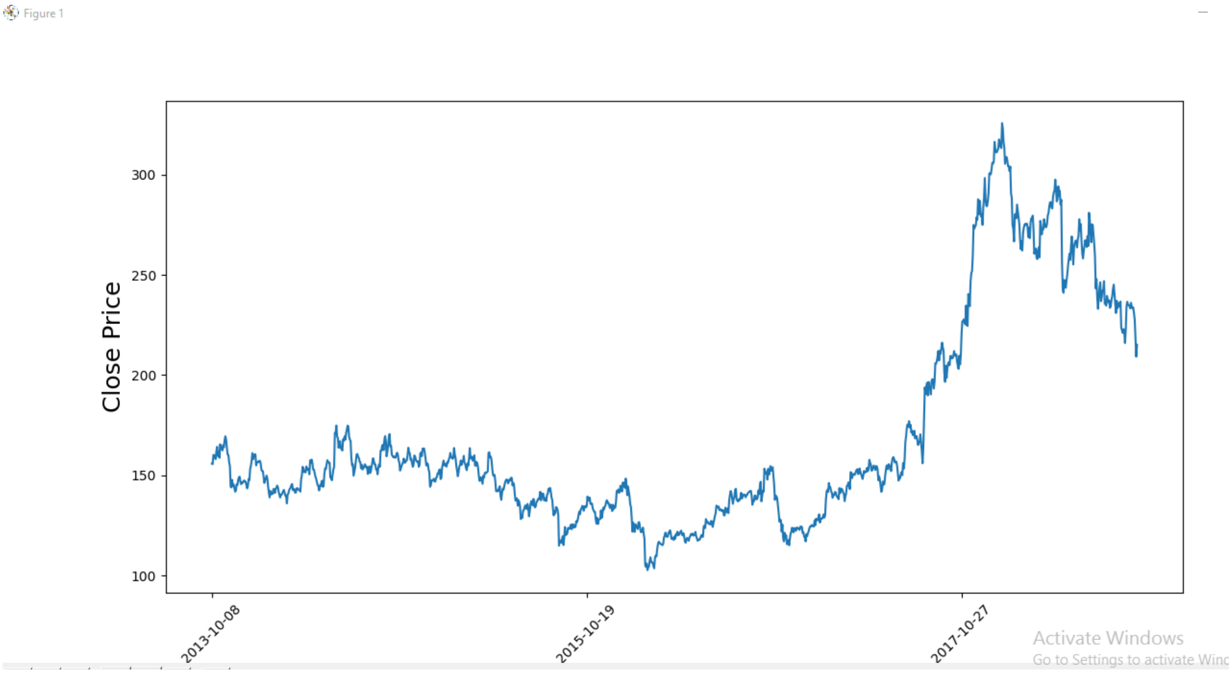
Last 3245

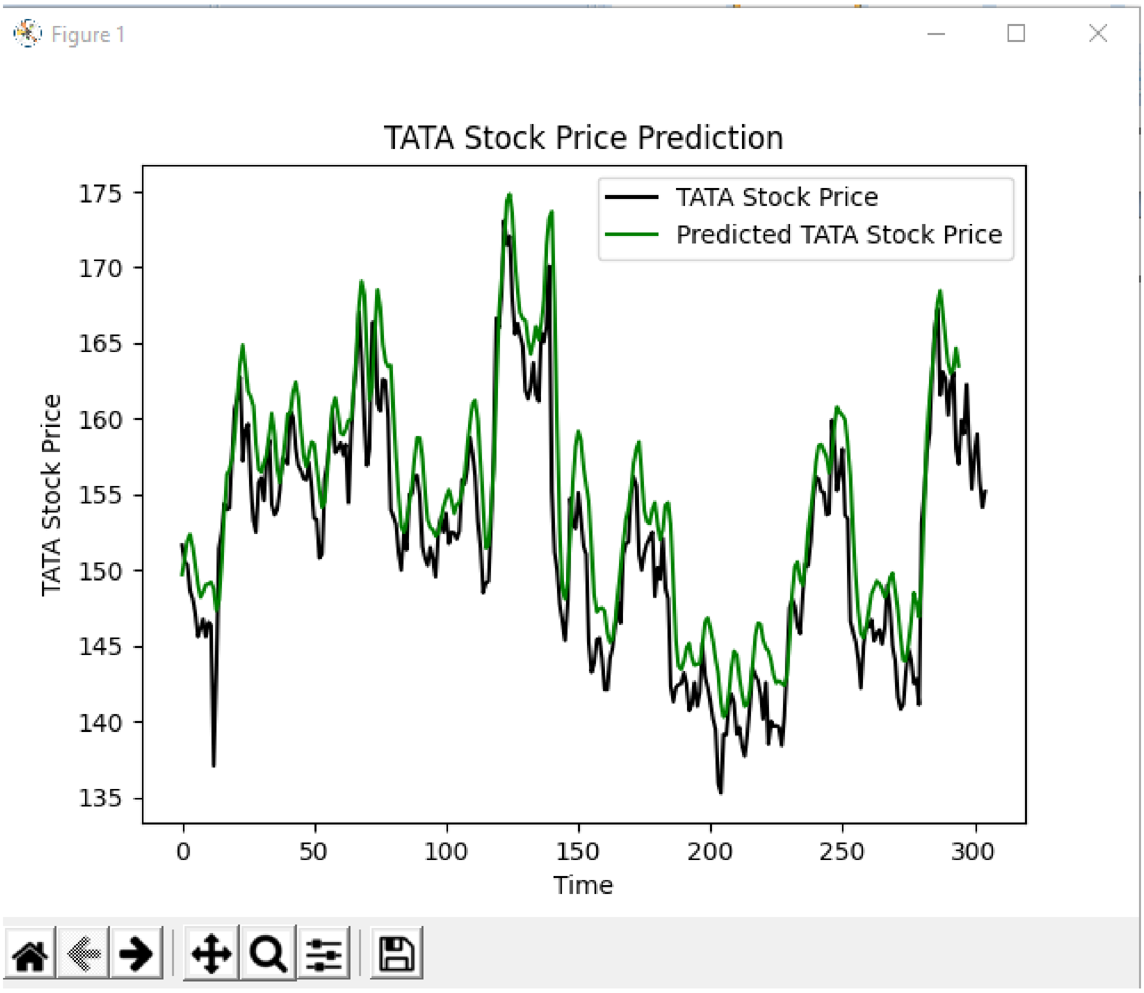
Trade 455687

Stock Prediction [24482.95]

Prediction result

GRAPHICAL ANALYSIS:





CONCLUSION:

In this study, we proposed a Stock Price prediction system which helps the investor for making correct decision regarding buy or sell the stock. In this we have taken Decision Tree Regression algorithm for this stock price prediction in our system. This algorithm allows us to categorize clearly based on previous data of companies over the years. This could guide the investors in the stock market to make profitable investment decisions whether to buy sell hold a share. We have done this using Machine learning and Python using Django framework. The aim of this project isto predict supervised machine learning algorithms to predict Stock price prediction, data processing has been done and split the data with trained data and fit the values and finally predictthe output result. This analysis can be used to reduce the error percentage in predicting the futurestock prices. It increases the chances for the investors to predict the prices more accurately by reducing the error percentage and hence increase their profit in share markets.

FUTURE WORK:

In the future, the stock price prediction system can be further improved by utilizing a muchbigger dataset than the one being utilized currently. Future research can be done with possible improvements such as more refined data and more accurate algorithm. More work on refining key phrases extraction in previously provided results will definitely produce better results, Enhancements in the system. This would help to in the accuracy of our prediction models. Furthermore, other models of Machine Learning could also be studied to check for the accuracy rate resulted by them.

References:

1. Ayo, Charles K. 2014. “Stock Price Prediction Using the ARIMA Model.”
2. Bathla, Gourav. 2020. “Stock Price Prediction Using LSTM and SVR.” : 211–14.
3. Billah, Mustain. 2016. “Stock Market Prediction Using an Improved Training Algorithm of Neural Network.” (December): 8–10.
4. Billah, Mustain. 2016. “Stock Market Prediction Using an Improved Training Algorithm of Neural Network.” (December): 8–10.
5. Chen, H. 2015. “Modelling and Prediction of Stock Price Dynamics Using System Identification Methodology Based on a Popularly Used Technique Analysis Data.” : 889–93.
6. Data, A Stock. 2020. “The Directionality Function Defect of Performance Evaluation Method in Regression Neural Network for Stock Price Prediction.” : 2020–21.
7. Firdaus, Muhammad. “Literature Review on Artificial Neural Networks and as Decision Support Tools.”
8. Gao, Tingwei, Yueting Chai, and Yi Liu. “Applying Long Short Term Momory Neural Networks for Predicting Stock Closing Price.” : 25–28.
9. Graph, Price. 2018. “Stock Price Trend Prediction Model Based on Deep Residual Network and Stock.” : 328–31.
10. Hadavandi, Esmaeil, Hassan Shavandi, and Keywords- Stock Price. 2010. “A Genetic Fuzzy Expert System for Stock Price Forecasting.” (Fskd): 41–44.