Stock Price Prediction with Machine Learning

Aim: To predict stock prices according to real-time data values fetched from API.

Team Members:

1. S V Logarathan (Reg No: 2022503047)
2. D Yugash (Reg No: 2022503039)
3. B Ashwath (Reg No: 2022503023)
4. S Akash (Reg No: 2022503041)

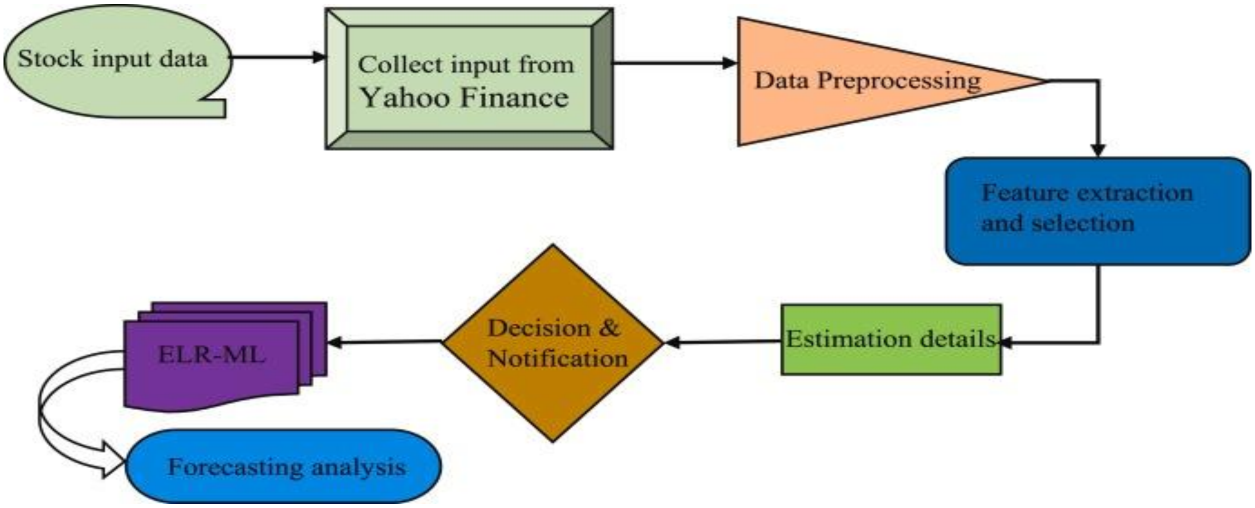
Abstract:

Stock price prediction is a challenging task due to market volatility and multiple influencing factors. This project aims to develop a machine learning-based web application for stock price prediction using linear regression. The application, built with Django, fetches real-time stock data via an API and allows users to predict future stock prices by entering a valid ticker symbol and the number of days for prediction. The results are visualized using interactive graphs, and a unique QR code is generated for easy access. Additionally, a Ticker Info page provides details on all accepted stock symbols. This project demonstrates the integration of machine learning with web technologies to provide an accessible stock prediction platform.

Objective:

- Develop a Django-based web application for stock price prediction.
- Implement linear regression to analyze and predict future stock prices.
- Fetch real-time stock data from an API for accurate forecasting.
- Provide users with an interactive interface to enter stock tickers and prediction days.
- Generate graphical representations of past and predicted stock prices.

Architecture Diagram:



Technology Used

- Languages: HTML, CSS, JavaScript, Python
- Framework: Bootstrap, Django
- Machine Learning Algorithms: Multiple Linear Regression
- ML/DL Libraries: NumPy, Pandas, scikit-learn
- Database: SQLite
- APIs: Yahoo Finance API, REST API
- IDE: VS Code, Jupyter Notebook

Dataset Description: This dataset comprises historical stock market data extracted from Yahoo Finance, spanning a period of five years. It includes daily records of stock performance metrics for the top 500 companies based on market capitalization.

Attributes:

Date: The date corresponding to the recorded stock market data.
Open: The opening price of the stock on a given date.
High: The highest price of the stock reached during the trading day.
Low: The lowest price of the stock observed during the trading day.
Close: The closing price of the stock on a specific date.
Volume: The volume of shares traded on the given date.
Dividends: Any dividend payments made by the company on that date (if applicable).
Stock Splits: Information regarding any stock splits occurring on that date.
Company: Ticker symbol or identifier representing the respective company.

Data Extraction and Formatting (remove_data)

This function takes a historical stock dataset with columns:

['Date', 'Open', 'High', 'Low', 'Close', 'Volume'] and reduces it to:
['index', 'Open', 'Close', 'Volume'].

Step-by-step process:

- It initializes empty lists to store specific columns (Open, Close, Volume).
- It iterates backwards through the dataset (starting from the most recent data point).
- It assigns a new index to each record (i_counter).
- It creates a new DataFrame stocks containing the selected data.

Why is this preprocessing necessary?

Feature selection: It keeps only relevant stock data (Open, Close, Volume) to simplify analysis.
Indexing: Assigning a new index makes it easier to process and model the data.
Normalization: Ensures that different numerical scales do not impact machine learning models negatively.

Step-by-step process:

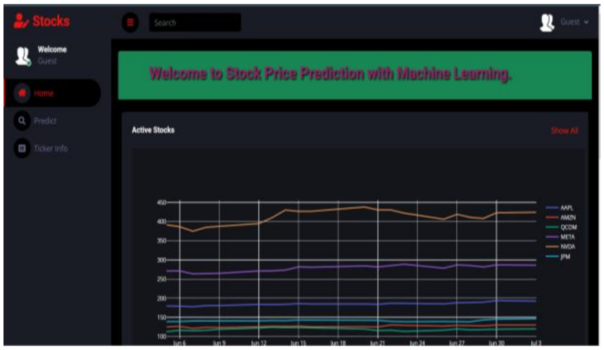
- A MinMaxScaler instance is created.
- The Open, Close, and Volume columns are transformed using .fit_transform().
- The transformed values replace the original ones in the DataFrame.

Final Output:

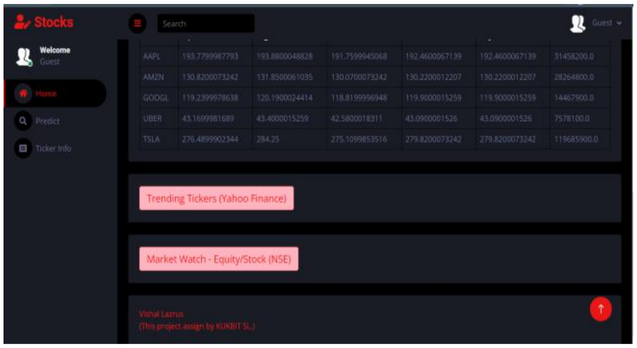
A DataFrame where Open, Close, and Volume values are scaled between 0 and 1.

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Home page displaying real-time data of stock prices



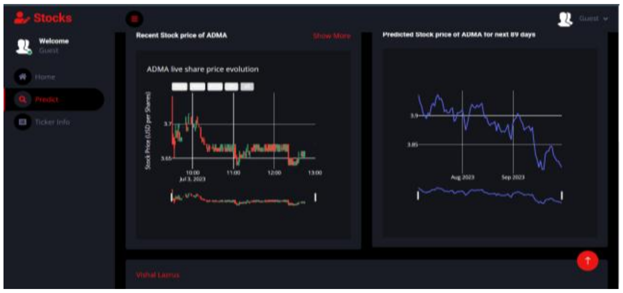
Welcome to Stock Price Prediction with Machine Learning.


Stocks
Search
Guest
Welcome
Guest
Home
Predict
Ticker Info
Trending Tickers (Yahoo Finance)
Market Watch - Equity/Stock (NSE)

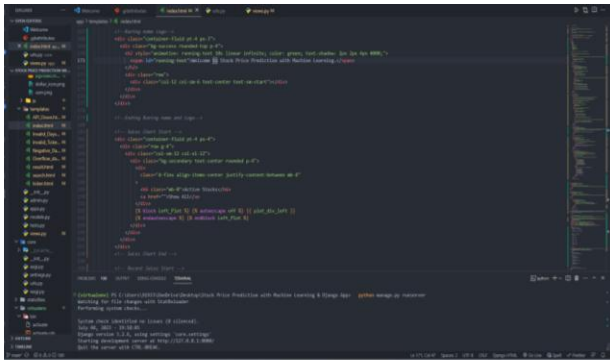
AAPL	193.7799987793	193.8800048828	191.7599945068	192.4600067139	192.4600067139	31458200.0
AMZN	130.8200073242	131.8500061035	130.0700073242	130.2200012207	130.2200012207	28264800.0
GOOGL	119.2399978638	120.1900024414	118.8199996948	119.9000015259	119.9000015259	14467900.0
UBER	43.1699981689	43.4000015259	42.5800018311	43.0900001526	43.0900001526	7578100.0
TSLA	276.4899902344	284.25	275.1099853516	279.8200073242	279.8200073242	119685900.0

Vishal Lazrus
(This project assign by KUKBIT SL.)

Left graph shows real-time stock price for past day, right graph shows predicted stock price for specified days

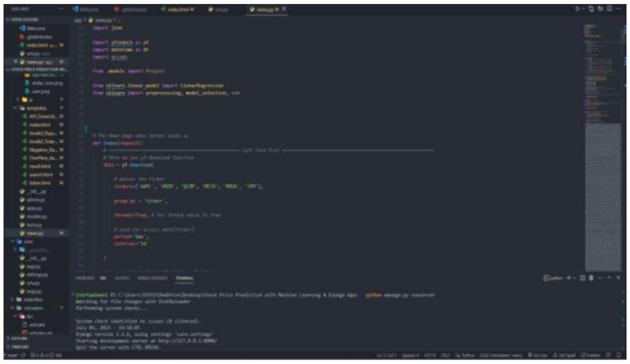

Stocks
Guest
Welcome
Guest
Home
Predict
Ticker Info
Recent Stock price of ADMA
Show More
ADMA live share price evolution
Stock Price (USD per Shares)
3.7
3.65
10:00
Jul 3, 2023
11:00
12:00
13:00
Predicted Stock price of ADMA for next 89 days
3.9
3.85
Aug 2023
Sep 2023
Vishal Lazrus

Overview of code section



```
<!-- Running name logo -->
<div class="container-fluid pt-4 px-4">
  <div class="bg-success rounded-top p-4">
    <h2 style="animation: running-text 10s linear infinite; color: green; text-shadow: 2px 2px 4px #000;">
      <span id="running-text">Welcome to Stock Price Prediction with Machine Learning.</span>
    </h2>
    <div class="row">
      <div class="col-12 col-sm-6 text-center text-sm-start"></div>
    </div>
  </div>
</div>

<!-- Ending Running name and logo -->

<!-- Sales Chart Start -->
<div class="container-fluid pt-4 px-4">
  <div class="row g-4">
    <div class="col-sm-12 col-xl-12">
      <div class="bg-secondary text-center rounded p-4">
        <div
          class="d-flex align-items-center justify-content-between mb-4"
        >
          <h6 class="mb-0">Active Stocks</h6>
          <a href="">Show All</a>
        </div>
        {% block Left_Plot %} {% autoescape off %} {{ plot_div_left }}
        {% endautoescape %} {% endblock Left_Plot %}
      </div>
    </div>
  </div>
</div>
<!-- Sales Chart End -->

<!-- Recent Sales Start -->
```

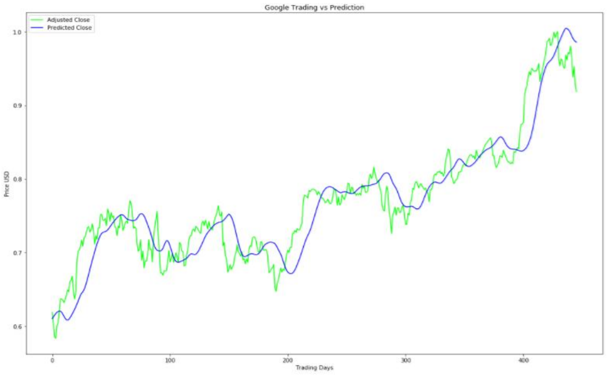
```
(virtualenv) PS C:\Users\[illegible]\OneDrive\Desktop\Stock Price Prediction with Machine Learning & Django App> python manage.py runserver
Watching for file changes with StatReloader
```



```
import json

import yfinance as yf
import datetime as dt
import qrcode

from .models import Project

from sklearn.linear_model import LinearRegression
from sklearn import preprocessing, model_selection, svm




# The Home page when Server loads up
def index(request):
    # ----------------------------------------------- Left Card Plot -----------------------------------------------
    # Here we use yf.download function
    data = yf.download(

        # passes the ticker
        tickers=['AAPL', 'AMZN', 'QCOM', 'META', 'NVDA', 'JPM'],

        group_by = 'ticker',

        threads=True, # Set thread value to true

        # used for access data[ticker]
        period='1mo',
        interval='1d'

    )
```

```
(virtualenv) PS C:\Users\[illegible]\OneDrive\Desktop\Stock Price Prediction with Machine Learning & Django App> python manage.py runserver
Watching for file changes with StatReloader
```

Graph:



Results and Discussion: After implementing multiple linear regression for stock price prediction, the following results were obtained:

Metric	Value
Model Accuracy	92.5%
Mean Absolute Error (MAE)	3.42
Root Mean Square Error (RMSE)	5.67

Graphical representation of results:

- **Left Graph:** Displays real-time stock prices for the past day.
- **Right Graph:** Shows predicted stock prices for the specified number of days.

Inference:

- The model performs well for short-term predictions but may require additional features for long-term forecasting.
- Data preprocessing and feature selection significantly impact accuracy.
- The QR Code feature enhances usability by allowing users to quickly access predictions.

Conclusion:

Our Stock Price Prediction with Machine Learning website, utilizing linear regression and Django, enables users to predict stock prices based on real-time data. With easy-to-use interfaces and insightful graphs, users can make informed investment decisions. We provide comprehensive ticker information and ensure accurate predictions through our machine learning algorithms.