**Stock prediction and analysis**

**About the project:**

This experimental project of stock prediction by analyzing already existing historical stocks/markers. We will be using several prediction models aimed at further in-depth prediction using concepts like linear regression, time series concepts, Sentiment Analysis combining complex models into one entire package to improve prediction analysis. Throughout the 6 months our plans will keep varying but there will be consistent effort placed in creating this project.

This is a passion project although we may have been inspired by code on online sources, I may take this up into future startup projects. As a visual based learner this project has been tough because of the countless amounts of theoretical concepts and programming. It’s a project with high burnout rate there may be inconsistencies or miscalculations throughout the project but we have put in the best work.

Concepts involved related to data science like interpolation, statistics, datatypes, data distribution, graph plotting are also covered. Many fields converged into one project.

***Sincere thanks to ChatGPT.***

TERMS TO KNOW:

1. Marker/Ticker: It’s your stock name listed on the stock market. Mentioned as markers or tickers when writing code
2. OHLC: Open, High, Low and Close meaning how much opening price the marker hit, high means the highest it went, Low means the lowest it dropped and close means prices before stock market closed

Tools used:

* Python libraries:

Data fetching/cleaning: YFinance, Pandas, OS

Exploratory Data Service:

Goal:  
To predict partially accurate results for future stock prices.

**Data fetching and cleaning:**

The most boring thing in the entire project. Collect data and clean them. What we did was fetch only 100 random markers because API’s like yfinance do not allow downloading many markers and everything in this project is done without spending a dime. Several companies provide accurate, fast downloads for a price but we used yfinance here to get a random of 100 markers out of 3300 registered markers. This block took us 4 days to develop the code, understand partially borrowed code, edited it and had it run successfully.

Files needed:

1. fetch\_data.py
2. clean\_data.py

fetch\_data.py:

Here we fetch 100 random markers from yfinance library in python.

import yfinance as yf

import pandas as pd

import random

import os

# Step 1: Get all tickers from yfinance's built-in S&P 500 list

sp500= pd.read\_html('https://en.wikipedia.org/wiki/List\_of\_S%26P\_500\_companies')[0]

all\_tickers = sp500['Symbol'].tolist()

# Optional: clean up symbols like BRK.B -> BRK-B (for Yahoo Finance format)

all\_tickers = [ticker.replace('.', '-') for ticker in all\_tickers]

# Step 2: Randomly pick 100 tickers

random\_100 = random.sample(all\_tickers, 100)

# Step 3: Make directory

os.makedirs('data/raw', exist\_ok=True)

# Step 4: Download historical data

for ticker in random\_100:

    try:

        df = yf.download(ticker, period='max')

        df.to\_excel(f'data/raw/{ticker}.xlsx')

        print(f'[+] Saved: {ticker}')

    except Exception as e:

        print(f'[!] Failed: {ticker} - {e}')