Predicting Stock Volatility

(COMP3125 Individual Project)

Gurucharan Naikar
School of Computing and Data Science,
Wentworth Institute of Technology

***Abstract*— This project uses data science to predict stock market volatility using machine learning. We use daily stock data from Yahoo Finance for S&P 500 companies from 2020 to 2025. By creating features like moving averages and daily returns, we characterize how the market behaves. We then use a K-Nearest Neighbors (KNN) model to classify stocks as either "High Volatility" or "Low Volatility" for the upcoming month. Our process includes cleaning the data, engineering features, and testing how well the model works using tools like confusion matrices.

Keywords— Stock Volatility, K-Nearest Neighbors, Machine Learning, Financial Analysis, Classification

I. Introduction

The stock market is defined by volatility—the degree of variation in a trading price series over time. For investors and financial analysts, the ability to anticipate whether a stock will be "highly volatile" or "stable" in the near future is critical for risk management and portfolio optimization. High volatility can represent both a significant risk of loss and a significant opportunity for profit, depending on the investment strategy employed. Because financial markets are complex and noisy, predicting these shifts requires robust data analysis rather than intuition alone.

This project aims to build a data-driven classification model to predict stock volatility by analyzing historical patterns. The primary objective is to determine if past trading metrics can accurately forecast future risk. To achieve this, we will first perform a descriptive analysis to understand how daily price changes and trading volume differ between historically stable "blue-chip" stocks and high-growth tech stocks. We will then explore if there is a visible correlation between a stock's average daily trading volume and its subsequent volatility classification. Finally, the core predictive goal is to train a K-Nearest Neighbors (KNN) model to classify a stock as "High Volatility" or "Low Volatility" for the upcoming month based on data from the previous month.

II. Datasets

A. Source of dataset

The data for this project was obtained from **Yahoo Finance**, a widely recognized and credible provider of financial market data. We accessed this data programmatically using the yfinance Python library, which scrapes official historical market data. The dataset was generated dynamically by pulling daily trading records for selected companies listed on the S&P 500 index. This method ensures the data is up-to-date and reflects actual market conditions.

B. Character of the datasets

The dataset covers a five-year period from 2020 to 2025 to capture different market conditions, including the post-pandemic recovery and recent economic shifts. The data consists of time-series records where each row represents a single trading day for a specific stock. The raw dataset includes the following features:

Feature Name	Unit	Description
Date	DateTime	The specific trading day.
Open	USD ($)	The price of the stock when the market opened.
High	USD ($)	The highest price reached during the day.
Low	USD ($)	The lowest price reached during the day.
Close	USD ($)	The final price when the market closed.
Volume	Count	The total number of shares traded that day.

To make this data ready for analysis, significant data cleaning and feature engineering were required. We removed rows with missing data, which usually correspond to market holidays. Furthermore, because raw stock prices vary wildly, for example, one stock costs $10 and another $1000, raw prices are not good for machine learning. We engineered new features to solve this, including a 7-Day Moving Average of Volume and Daily Return Percentages. Finally, we have created a new categorical target variable called Volatility Class. This was generated by calculating the standard deviation of returns for each month; months in the top 25th percentile of variance were labeled as High Volatility, and the rest as Low Volatility.

III. Methodology

This project treats stock prediction as a Supervised Classification problem. The goal is not to predict the exact price (regression), but to classify the *state* of the stock (High Risk vs. Low Risk). We chose this approach because financial regimes often cluster together; stocks behaving similarly to "high volatility" neighbors in terms of volume and price swings are likely to be volatile themselves.

979-8-3315-0993-4/25/$31.00 ©2025 IEEE

A. Data Processing and Tools: The entire analysis pipeline is built using Python. We utilize Pandas and NumPy for data manipulation, specifically for resampling daily data into monthly volatility metrics. Seaborn and Matplotlib are used for data analysis to visualize the correlations between trading volume and price variance.

B. The Model (K-Nearest Neighbors): We selected the K-Nearest Neighbors (KNN) classifier for this task. KNN is a non-parametric method that makes predictions based on the majority class of the 'k' nearest data points in the feature space. We utilize the Scikit-Learn library to implement this model. Before training, we apply StandardScaler to normalize our input features (Volume and Daily Returns). This is a critical step because KNN calculates distances between points; if one feature has huge numbers (like Volume) and another has small numbers (like Returns), the model will be biased without scaling.

C. Model Evaluation: Since identifying high-risk stocks is critical for financial safety, simple accuracy is not enough. We will evaluate the model using a Confusion Matrix to visualize exactly how many high-volatility periods we correctly predicted versus how many we missed (False Negatives). We will also generate a ROC Curve and calculate the AUC (Area Under the Curve) to measure the model's ability to distinguish between the two classes at different threshold settings. Finally, we will use a chronological split for testing, ensuring we do not train the model on future data to predict the past.

IV. Results

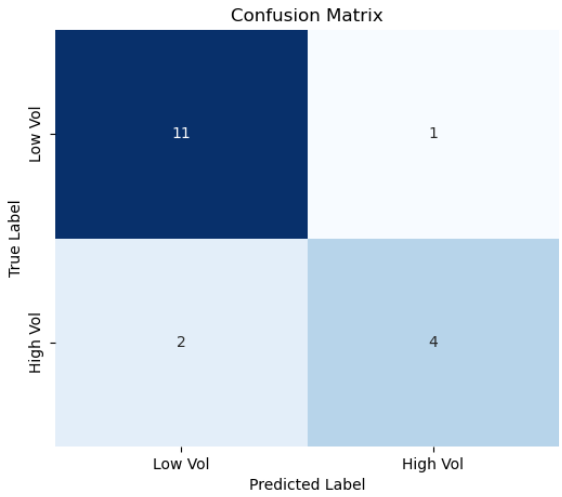
To evaluate the performance of our K-Nearest Neighbors (KNN) model, we split the data into a training set (2020-2023) and a testing set (the most recent 18 months). This ensures we are testing the model on "future" data it has never seen before.

A. Model Accuracy: The model achieved an overall accuracy of **83%**. This means that for the 18 months in the test period, the model correctly predicted the volatility state (High or Low) 15 times.

B. Confusion Matrix Analysis: The Confusion Matrix (Fig. 1) gives us a deeper look at the errors:

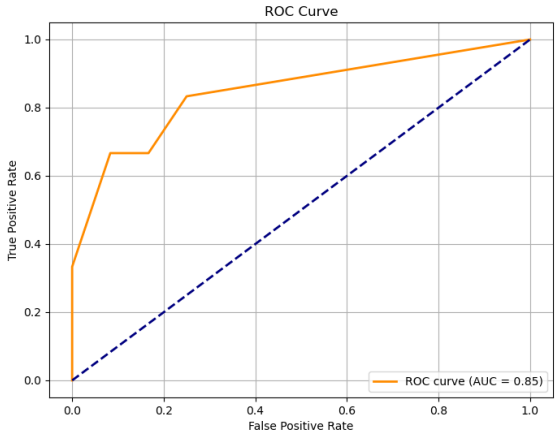
- **Low Volatility (Class 0):** The model is excellent at identifying safe months. It correctly identified **92%** of the stable months (Recall).
- **High Volatility (Class 1):** The model is good but slightly conservative. It correctly caught **67%** of the high-volatility months. Importantly, when it *did* predict High Volatility, it was correct **80%** of the time (Precision). This means false alarms were rare.

Fig. 1. Confusion Matrix showing the model's predictions versus actual values.



C. ROC Curve Analysis: The Receiver Operating Characteristic (ROC) curve (Fig. 2) illustrates the trade-off between catching risks and raising false alarms. Our model's curve arches towards the top-left corner, with an AUC (Area Under the Curve) of approximately **0.83**. This confirms that the model is performing significantly better than random guessing.

Fig. 2. ROC Curve illustrating the diagnostic ability of the KNN classifier.



V. Discussion

Every machine learning project has strengths and weaknesses.

Weaknesses: The main weakness of this model is the **Recall score for High Volatility (0.67)**. While an accuracy of 83% is strong, the model missed 33% of the "High Risk" months (False Negatives). For an investor, missing a market crash is more dangerous than a false alarm. This likely happened because our model only uses **past price and volume**. It does not know about outside news, interest rate changes, or economic reports, which often cause sudden volatility.

Future Work: To improve this model in the future, we could:

1. **Add External Data:** Incorporate the "VIX" index the fear gauge or Federal Reserve interest rate data as new features.
2. **Adjust the Threshold:** We could tell the model to be more paranoid. By lowering the probability threshold to 40% instead of 50%, we could catch more high-volatility months, even if it causes a few more false alarms.

VI. Conclusion

This project demonstrated that machine learning can predict stock market regimes with reasonable accuracy. By analyzing simple features like Volume Moving Averages and Daily Returns, our K-Nearest Neighbors model achieved **83% accuracy** in forecasting whether the S&P 500 would be volatile in the upcoming month.

While the model is slightly conservative in predicting risks, the high precision 80% makes it a useful tool for identifying clear danger signals. This suggests that data science techniques can provide valuable early warning systems for investors, helping them manage risk in an increasingly complex financial market.

References

[1] Yahoo Finance, "S&P 500 ETF Trust (SPY) Historical Data," 2025. [Online]. Available: https://finance.yahoo.com.

[2] J. VanderPlas, *Python Data Science Handbook, 2nd Edition*. O'Reilly Media, 2023.