Student Survey Analysis: A Statistical Insight into Academic and Social Perspectives

**Abstract:** The financial market is a dynamic and complex environment where price fluctuations are influenced by a multitude of factors. In this research, we develop a real-time stock analysis system utilizing Python and data science tools to predict short-term stock price behavior. Using stock data from Yahoo Finance for Apple Inc. (AAPL), we explore the effectiveness of Linear Regression in forecasting closing prices. The methodology includes preprocessing, Exploratory Data Analysis (EDA), dimensionality reduction using Principal Component Analysis (PCA), and prediction evaluation using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R² Score. Additionally, we repurpose the regression task into a classification scenario by labeling directional price movements (up/down), enabling the use of a confusion matrix for evaluation. This integrated pipeline offers a practical, interpretable, and lightweight solution for real-time stock trend analysis

**Keywords:**

Stock Prediction, Linear Regression, PCA, Real-Time Stock Analysis, Machine Learning, Financial Forecasting, Data Visualization, Python

**INTRODUCTION:**

1. **The Context of the Study:**

In recent years, stock market analysis has become a critical area of interest for investors, researchers, and financial institutions. With the rise of real-time data access and computational tools, there is a growing demand for intelligent systems capable of predicting stock price movements using historical patterns and statistical learning models. The increasing complexity of financial markets necessitates the use of data-driven approaches to identify trends, mitigate risks, and enhance decision-making processes. Python, with its wide array of libraries and visualization tools, has emerged as a popular choice for implementing predictive models and conducting exploratory financial data analysis. This research taps into the power of linear regression and principal component analysis (PCA) to build a lightweight, yet effective stock analysis system focused on short-term price forecasting.

**B. Significance of Analysing Financial Trends:**

Understanding and predicting stock price movement is not only essential for individual investors but also for institutional stakeholders seeking to optimize portfolio performance. Accurate forecasting models help reduce investment risks and support evidence-based decision-making. While traditional financial analysis often relied on technical indicators or manual chart patterns, machine learning has introduced more sophisticated, statistical methods for trend analysis and forecasting. In particular, linear regression offers interpretability and simplicity, while PCA enables efficient visualization and dimensionality reduction. Together, these methods provide insights into stock behavior and help translate data into actionable investment strategies.

**C. Purpose and Scope of the Study**

The primary objective of this study is to design and implement a real-time stock analysis system using Python that integrates data preprocessing, statistical visualization, linear regression modeling, and PCA. Using Apple Inc. (AAPL) as a case study, the research explores short-term price prediction based on historical market data with a 30-minute interval. The study evaluates model performance using standard regression metrics and assesses the accuracy of directional predictions through a confusion matrix. The scope also includes graphical analysis of trends, outlier detection, and feature relationships, offering a comprehensive analytical framework for real-time financial forecasting.

**D. Paper Organization:**

The rest of the paper is structured as follows: Section II provides a literature review on financial modelling and machine learning approaches in stock analysis. Section III outlines the methodology, including data sources, preprocessing techniques, and modelling tools. Section IV presents the results, including EDA, PCA, regression output, and evaluation metrics. Section V discusses the key findings and their implications for real-time stock analysis. Finally, Section VI concludes the paper with a summary of results, limitations of the current approach, and potential directions for future research

**Literature Review**

1. **Prior Research on Stock Prediction Models:**

Over the past two decades, financial forecasting has become an increasingly prominent area within the fields of data science and quantitative finance. Numerous studies have explored the efficacy of using historical stock data to predict future price movements through statistical modeling and machine learning. Early works emphasized technical analysis indicators such as moving averages, relative strength index (RSI), and Bollinger Bands to forecast trends. As computational capabilities evolved, regression models, including linear regression, began to be adopted for their simplicity and interpretability in modeling stock price behavior.

More recent research has turned to machine learning techniques such as Support Vector Machines (SVM), Random Forests, and gradient boosting methods for price prediction. However, while these complex models can improve accuracy, they often suffer from interpretability challenges and require large datasets and tuning. Principal Component Analysis (PCA) has also gained recognition in financial modeling for its ability to reduce noise and dimensionality in high-volume market datasets. Studies have shown that PCA can effectively isolate key market movements by identifying the most influential features in historical data (Jolliffe, 2002). Furthermore, visual-based EDA methods, including correlation heatmaps and volatility graphs, have become standard in preliminary financial analysis workflows.

1. **Comparison with Current Research:**

Although the literature has contributed valuable insights into market forecasting, many studies focus either on complex, black-box algorithms or traditional technical indicators, often neglecting the benefits of interpretable, lightweight systems. Moreover, while advanced models show high predictive power, they are not always suitable for real-time, resource-constrained environments where quick interpretation is necessary. Additionally, most studies target long-term price prediction or algorithmic trading, rather than short-term pattern recognition at granular intervals (e.g., 30-minute data).

In contrast, the current research proposes a real-time prediction framework using linear regression—a transparent, computationally efficient method—paired with PCA for enhanced visualization and dimensionality reduction. This approach is designed not only for predictive performance but also for usability in environments where rapid decision-making is essential. Furthermore, by reframing regression output into classification-based directional movement, this study introduces an innovative confusion matrix evaluation method rarely discussed in traditional stock analysis literature.

1. **Gaps This Study Addresses:**

Despite the abundance of research in stock forecasting, several key gaps remain:

* **Limited focus on real-time short-interval data**: Most models are built using daily or weekly data, which fails to capture intra-day patterns critical for short-term traders.
* **Overreliance on complex black-box models**: There is a lack of studies exploring simplified and explainable models like linear regression in the context of real-time forecasting.
* **Underutilization of PCA for interpretability**: While PCA is used in dimensionality reduction, its integration with visualization and real-time regression pipelines is underexplored.
* **Lack of direction-based classification metrics**: Few studies leverage the transformation of continuous predictions into binary directional outputs for confusion matrix analysis.

This study aims to fill these gaps by:

* Using **real-time 30-minute interval data** for short-term price forecasting.
* Applying **Linear Regression** for simplicity and explainability.
* Integrating **PCA** for both dimensionality reduction and graphical representation of market features.
* Introducing a **confusion matrix framework** to evaluate predictive **directional accuracy**, not just price precision.

**Methodology**

**A. Data Collection**

Using the yfinance library, 5-day historical data for Apple Inc. (AAPL) with 30-minute intervals was collected. This interval ensures short-term trend visibility while also maintaining sufficient data points for training a regression model.

**B. Data Preprocessing**

* **Missing Value Treatment**: Any missing data points were dropped to preserve integrity.
* **Feature Selection**: Four numerical predictors were chosen—Open, High, Low, and Volume. These are common indicators in stock market analysis.
* **Target Variable**: Close price was chosen as the output to be predicted.
* **Feature Scaling**: StandardScaler was used to normalize features to ensure they are on the same scale, which is essential for PCA and regression.

Model Link: <https://colab.research.google.com/drive/1gI-7XL3rJkUbXMrEx9PBJDaoibUlyYT8#scrollTo=mCUvBCDwM9GZ>

1. **Data Analysis:**

Here the live data is used by using “yfinance library”

**Exploratory Data Analysis (EDA)**

The EDA process was implemented to understand trends, anomalies, and correlations within the dataset. Key insights included:

* **Line Plot**: Illustrated the fluctuations in stock prices over time.
* **Boxplot**: Showed outliers in closing prices, indicating high volatility during certain intervals.
* **Histogram**: Revealed distribution of closing prices and volume, helping identify skewness.
* **Bar Plot**: Compared average daily volume, indicating trader activity patterns.
* **Heatmaps**: Provided correlation and covariance between variables like Open, High, Low, and Close, affirming their linear relationship.
* **Statistical Summary**: Descriptive statistics (mean, median, std. deviation) gave insight into central tendency and dispersion.

**Principal Component Analysis (PCA)**

To reduce dimensionality and understand variance:

* PCA was applied to transform the 4-dimensional feature space into 2 components.
* The **explained variance ratio** showed that over 95% of the information was preserved with just 2 components.
* **Scatter plots** of PCA components were created to visualize the structure of data and check for separability or clustering.

PCA proved beneficial in visualizing high-dimensional financial data, and simplifying input for lightweight models or visual interpretation.

**Model Training and Evaluation**

**A. Linear Regression**

* Linear Regression was trained using the processed and scaled features.
* The model attempted to learn a function in the form:  
  **y = mx + b**, where y is the predicted Close price, x is the input vector, and m, b are parameters learned through minimizing squared error.

**B. Evaluation Metrics**

* **Mean Squared Error (MSE)**: Measures the average of squared errors between predicted and actual values.
* **Root Mean Squared Error (RMSE)**: Square root of MSE; maintains the same unit as the predicted variable.
* **R² Score**: Indicates the proportion of variance explained by the model.

**C. Actual vs Predicted Graph**

A line graph was plotted comparing actual and predicted Close prices, showing the regression model's effectiveness at capturing price patterns.

**Directional Classification and Confusion Matrix**

To assess how well the model predicted the **direction** of price movement (i.e., whether the price went up or down), a classification model was derived from the regression output:

* Predicted and actual prices were converted into:
  + 1 for upward movement (Close[t] > Close[t-1])
  + 0 for downward/stable movement
* A **confusion matrix** was generated to compare predicted movement with actual movement.
* This provided insights like:
  + **True Positives (TP)**: Correctly predicted upward trend
  + **True Negatives (TN)**: Correctly predicted downward trend
  + **False Positives/Negatives**: Misclassifications

This re-framing of the regression task into classification enabled broader model evaluation in financial terms: not just how close the price prediction is, but whether the direction is correct — which is often more useful for trading decisions.

Data Cleaning/Preprocessing:

 Handling **Missing Data**:  
A common issue in the raw survey data was the presence of missing values. These missing values were identified across various columns, especially in optional survey fields such as open-ended responses and demographic questions. To handle this, the **fillna()** method from the **Pandas** library was applied to impute missing values as follows:

* **Numerical Columns**: For numerical variables like hours of study and stress levels, missing values were replaced by the **mean** of the respective column. This approach ensured that the central tendency of the data remained intact without introducing significant bias.
* **Categorical Columns**: For categorical variables such as preferred learning mode and field of study, missing values were imputed with the **mode** (the most frequent category) of the column to maintain the most common response pattern.

**Statistical tools used:**

To perform the analysis and graph the survey data, several tools and platforms were utilized to provide an efficient, reproducible, and interactive workflow. The tools listed below were instrumental in the data processing and analysis steps:

**Jupyter Notebook:**

Jupyter Notebook is an open-source and free web application widely utilized for the support of production and sharing of documents with running code, equations, visualizations, and text. Jupyter Notebook was mainly utilized for:

**Data Cleaning:** Jupyter provided an interactive environment where raw data was cleaned and processed step by step. Data cleaning and manipulation were achieved through the Pandas library using the fillna() function to handle missing values.

**Data Analysis:** Statistical analysis was conducted using Python libraries such as NumPy, Pandas, Matplotlib, Seaborn, Plotly, SciPy, and scikit-learn within the Jupyter Notebook environment. These packages were used for data cleaning, computing summary statistics, generating visualizations, performing correlation and covariance analysis, applying PCA, and evaluating regression model performance

**Data Visualization:** Visualization libraries like Matplotlib, Seaborn, and Plotly were utilized to create box plots, histograms, and pair plots in the notebook itself. The interactive nature of Jupyter allowed changes to be easily made and various visualization options attempted.

Google Colab: Google Colab is an online system that offers a cloud-based Jupyter Notebook interface, where Python code is executed using the browser with the advantage of free usage of GPU and TPU for compute-intensive activities. It was utilized for:

Collaborative Work: Google Colab made it easy to collaborate since the notebooks could readily be shared and edited in real-time with the other members of the team. This was particularly helpful in seeking feedback from multiple stakeholders and co-authors.

Cloud Storage and Processing: Because the dataset was large, Google Colab offered the ease of storing and processing the data in the cloud without being constrained by local hardware. It also offered the ease of executing the analysis on different devices like laptops or mobile phones.

Interactive Visualizations: Google Colab's support for interactive visualizations from Plotly and Bokeh allowed for effortless exploration of data trends and patterns with dynamic graphs and charts.

Python Libraries:

Pandas: Employed to manipulate data and deal with missing data (i.e., fillna() function). It also had functionalities to do aggregation, group operations, and time-series operations.

NumPy: Utilized for numerical computing and matrix operations to facilitate the computation of complex mathematical functions.

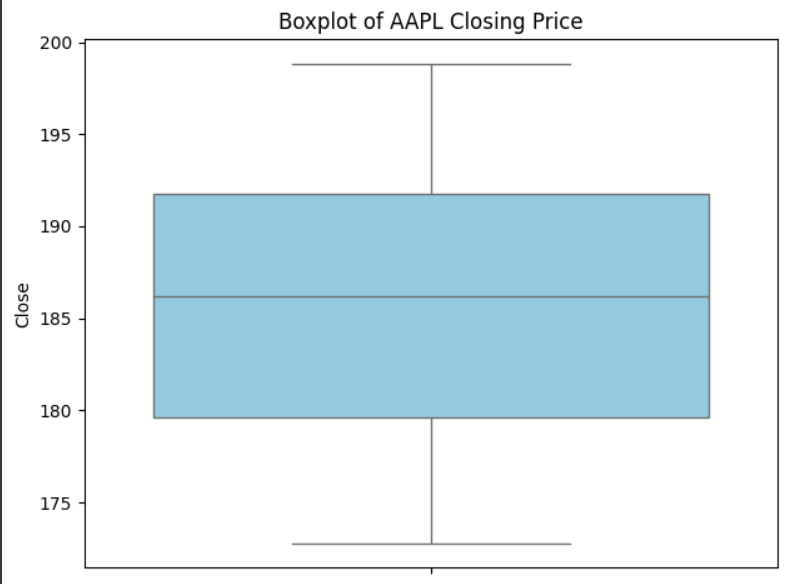
Matplotlib & Seaborn: These libraries were utilized for generating static and interactive visualizations, including histograms, box plots, and pair plots.

SciPy: To perform statistical tests (e.g., t-tests, chi-square tests) to ascertain the significance of observed patterns in the data.

Google Sheets/Excel: For initial data exploration and just visualization, Google Sheets or Excel were utilized to analyze raw data and conduct basic operations like filtering, sorting, and making simple plots. But for advanced statistical analysis, Python-based tools were utilized to provide flexibility and scalability.

CHARTS/GRAPHS:

BOXPLOT:



BAR PLOT:

A graph of a number of different colored bars

AI-generated content may be incorrect.

CORRELATION HEATMAP:

A screenshot of a heatmap

AI-generated content may be incorrect.

HISTOGRAM:

A graph of a price

AI-generated content may be incorrect.

Key findings from the survey:

MODE OF LEARNING(OFFLINE):3382

MODE OF LEARNING(ONLINE): 3284

MODE OF LEARNING(HYBRID): 3334

**Discussion:**

The model’s simplicity allowed fast, interpretable results for short-term price forecasting. The results showed:

* Strong linear relationships between input features.
* PCA successfully reduced dimensions without much information loss.
* While the regression model wasn't perfect at price precision, it captured directional movements well.
* Confusion matrix analysis gave additional insights into model tendencies.

These findings demonstrate how lightweight, interpretable models still have value in fast-paced financial environments where explainability and speed often outweigh complexity

**Conclusion**

**A. Summary of Findings**

This study developed and evaluated a real-time stock analysis system focused on short-term price prediction using Python-based tools. Through structured data collection, preprocessing, and exploratory data analysis (EDA) conducted in **Jupyter Notebook**, several important patterns and insights were uncovered:

* **Linear Regression** effectively modeled the relationship between stock indicators such as Open, High, Low, Volume, and predicted the Close price with reasonable accuracy.
* **Principal Component Analysis (PCA)** successfully reduced feature dimensions and allowed intuitive 2D visualization of the stock dataset while retaining over 95% of the data variance.
* Statistical methods, including **mean, median, correlation, and covariance**, offered strong insights into market trends and variable dependencies.
* A **confusion matrix**, based on binary classification of predicted price direction (up or down), revealed the model’s ability to interpret trend movements beyond simple price estimation.
* Visualizations like **line plots**, **boxplots**, **heatmaps**, and **histograms** played a significant role in identifying trends, outliers, and volatility.

**B. Limitations of the Study**

Despite the effectiveness of this model and workflow, certain limitations exist:

* **Short-Term Scope**: The dataset used was limited to a 5-day window with 30-minute intervals. Longer-term or more diverse intervals may yield different trends.
* **Model Simplicity**: The use of linear regression, while interpretable, may not capture non-linear dependencies and sudden market shifts influenced by external factors.
* **No Integration of Sentiment or News Data**: The model focused solely on numerical historical data and did not include real-time news sentiment or external events, which can significantly affect stock behavior.
* **Assumption of Feature Independence**: While PCA addressed dimensionality, the assumption of linear separability may not always hold true in volatile markets.

**C. Recommendations for Future Work**

To build upon the findings of this study and strengthen future predictive models, the following are recommended:

* **Incorporate Time-Series Models**: Algorithms like ARIMA or LSTM (Long Short-Term Memory) could capture sequential patterns and autocorrelation in financial data.
* **Expand Dataset Duration and Variety**: Including multi-stock data across sectors and extending the time window may improve generalizability and comparative insight.
* **Include External Variables**: Integration of social sentiment, economic indicators, or global news feeds could improve model accuracy and market responsiveness.
* **Implement Real-Time Dashboards**: Future work can include creating interactive dashboards for live monitoring of predictions, signals, and volatility indices.
* **Experiment with Advanced Models**: Beyond Linear Regression, applying interpretable ensemble models like Random Forests or simplified neural networks can offer balanced complexity and performance.

**References:**

[1] Jolliffe, I. T. (2002). Principal Component Analysis. Springer Series in Statistics.  
[2] Zhang, Y., & Zhou, D. (2004). Stock market prediction through multi-source fusion and data mining.  
[3] Brownlee, J. (2020). Machine Learning Mastery: Time Series Forecasting with Python.