Predictive Modeling in Financial Markets: A Data Warehousing Approach

Sal Mourad
Computing Science
University of the Fraser Valley
Abbotsford, Canada
salman.mourad@student.ufv.ca

Callum Brezden
Computing Science
University of the Fraser Valley
Abbotsford, Canada
callum.brezden@student.ufv.ca

Andrew Meier
Computing Science
University of the Fraser Valley
Abbotsford, Canada
andrew.meier1@student.ufv.ca

Allen Conolly
Computing Science
University of the Fraser Valley
Abbotsford, Canada
allen.conolly@student.ufv.ca

***Abstract*—This report outlines our development of a financial analysis platform focusing on the stock market. Utilizing Python scripts, we efficiently extract, transform, and load data into our data warehouse. Our warehouse design includes dimensions such as stock, company, date, currency, index fund, bond, and commodity, supporting an efficient fact table containing measures and various quantified values. We've adopted a strategic approach to data collection, leveraging online datasets and APIs. Python enables seamless interaction with our Oracle data warehouse. Progress has been made in ETL scripting, with machine learning integration intended to be applied in future. Our aim is to provide a valuable resource for financial analysis, benefiting personal investments and market insights.**

***Index Terms*—Financial Analysis, Stock Market, Data Warehouse, Python Scripts, Data Extraction, Oracle, ETL, Machine Learning.**

I. Introduction

In the world of financial analysis, the importance of efficient and robots data management cannot be overstated. It forms the foundations of both informed decision-making and predictive modelling. Our project focuses on the development of a comprehensive financial analysis platform, built on the principles of data warehousing.

To accomplish many of our goals for this project, we centred our efforts on the efficient extraction, transformation, and loading of financial data into our robust data warehouse. This warehouse serves as the focal point of our operations, housing dimensions ranging from stock and company details, to index and commodity prices. Furthermore, we leveraged the capabilities of Oracle to ensure the reliability and scalability of our data infrastructure.

Additionally, in order to ensure seamless data collection, we utilised Python scripts to streamline various aspects of our project. The scripts play a crucial role in facilitating the extraction, transformation and loading data into our data warehouse. With Python, we were able to interact effectively with our Oracle data warehouse, ensuring smooth data management processes.

Through strategic integration of Python scripting and Oracle's proven robust capabilities, our project intends on providing a comprehensive financial analysis platform that provides users with actionable insights. While machine learning is a potential avenue for future exploration and is a topic of discussion in our report, we determined that due to time constraints, our immediate focus will lie on establishing a solid foundation for data-driven financial analysis via data warehousing and efficient Python ETL scripting.

II. Methodology

A. ETL Data Collection and Warehouse Population

Our methodology utilises meticulous data collection from diverse sources, including online datasets and API's. We gained access to extensive financial datasets primarily through well populated and maintained csv files from reputable sources (*see references for more details*), allowing us to populate our data warehouse with approximately 1.8 million records of financial data.

Our team focused on particular stock market elements, including individual stocks, indexes, commodities, and bonds. To streamline the data collection process and ensure consistency and efficiency, we developed Python ETL (Extract, Transform, Load) scripts. These scripts played a crucial role in extracting data from various sources, transforming it into a standardized format, and loading it into our data warehouse.

Following a few iterations of our scripts, we were able to develop a very well structured and intuitive method of extracting data from our data sources and have them placed directly into our data warehouse, the finer details of which are discussed in later sections of this report.

III. Data Warehouse Schema and Design

Our database schema, depicted in Figure 1 below, serves as a reference for the majority of this section's discussion.

We designed our data warehouse in a sophisticated and intuitive way intended to meet the needs of financial analysis. We opted to implement a snowflake schema as it is highly flexible and scalable. This schema structure facilitates the normalisation of our data warehouse, which ensures efficient data storage and retrieval. Our schema is designed to be modular and maintain relational integrity. This means we can effortlessly modify, expand or update our tables as needed. This design ensures our data remains accurate and reliable.

Within our snowflake schema, we selected particular dimensions that cover a diverse spectrum of financial data, which in turn provide deep insights that can be leveraged into high level decision making and analytics. These dimensions serve as the foundations of our data warehouse, providing a structured framework for organising and categorising information.

The following are a list of dimensions found in our data warehouse:

- **Stock:** Incorporates another dimension within it to isolate company-specific information, enhancing flexibility for future data reuse.
- **Company:** Holds company-specific details for analysis, providing deeper insights into individual companies.
- **Date:** Archives data associated with time and serves as temporal reference.
- **Currency:** Provides currency type and exchange rate to USD, simplifying interpretation of values.
- **Index Fund:** Captures yield, management company, index type, and net asset amount, offering a comprehensive overview.
- **Bond:** Contains bond information similar to that of the index fund table.
- **Commodity:** Holds data for commodities including type and unit of measure, offering a view of each commodity's specifics.

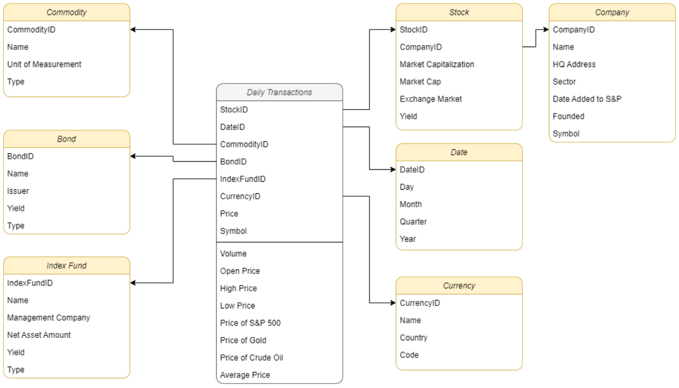


Fig. 1. Financial Data Warehouse Schema

Complementing the aforementioned dimensions is the Fact Table, which serves as the central repository for measures and quantitative data that is essential for financial analysis. We ensured our fact table incorporated key metrics highly relevant to the stock market and deeply insightful.

The following are a list of measures found in our data warehouse:

- **Volume:** Represents the quantity of stocks traded in a transaction.
- **Open Price:** Denotes the starting price of a stock within a trading period.
- **Low Price:** Indicates the lowest price of a stock within a trading period.
- **High Price:** Signifies the highest price of a stock within a trading period.
- **Price of Gold:** Reflects the current market value of gold.
- **Price of S&P 500:** Represents the current market value of the S&P 500.
- **Price of Crude Oil:** Indicates the current market value of crude oil.
- **Average Price:** Denotes the cumulative average price of the specific item up until the date associated with the record.

Through the integrated arrangement of dimensions and the fact table within our snowflake schema, we establish a structured foundation for extracting actionable insights in the realm of financial analysis.

IV. Github Repository & Oracle Connection

Our team utilized Github as a platform to store and share our code, allowing others to learn from and build upon our work. The repository contains a comprehensive collection of resources that were instrumental in the creation and operation of our financial data warehouse. The repository includes:

- **ETL Scripts:** These Python scripts are responsible for the extraction, transformation, and loading of data from various financial markets into our data warehouse. They ensure that the data is clean, consistent, and ready for analysis.
- **SQL Scripts:** These scripts were used to create the tables for the data warehouse. They define the structure of the database and ensure that it is optimized for the types of queries that will be run against it.
- **Diagrams:** We have included diagrams to provide a visual representation of the system architecture and data flow. These diagrams help to understand the overall structure and operation of the data warehouse.
- **Machine Learning Prototype Code:** This represents a prototype of our machine learning model, designed to leverage the data within our warehouse for making investment decisions. As a preliminary version, it forms the foundation of the predictive analytics component of our project, showcasing the potential of data-driven decision-making in finance.

Connecting to the Oracle data warehouse is achieved through the use of Oracle Wallets, a secure and convenient way to store database credentials. We have provided detailed instructions in our documentation on how to set up and connect an Oracle Wallet, enabling users to run our scripts on any Oracle instance.

V. Graphical Analysis of Collected Data

We capitalized on the vast amount of data we were able to collect and store in our data warehouse to plot visual representations of particular categories of financial trends. We leveraged Python and the MatPlotLib library to create our plots. Additionally, we focused our efforts on areas surrounding popular stocks such as Apple, Microsoft and Google, as well on specific commodities and indexes, among other items.

The first graph we plotted shown in figure 2 illustrates the trajectory of stock prices of large organisations such as Best Buy, Amazon and Apple between the years 2010 and 2022 as compared to their prices over those specific years. We can see from the plot that Amazon and Apples stock price fluctuated over the years with some large up and down trends, while Best Buys stock price has remained relatively less volatile over the same amount of years. This chart can provide insights into these particular companies market performance over a large time span.

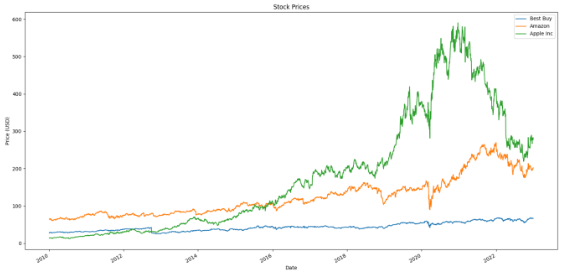


Fig. 2. Stock Price Trends of Best Buy, Amazon, and Apple (2010-2022)

The second graph shown in figure 3 offers a comprehensive view of commodity prices over a 24 year time period between the year 2000 to the year 2024. We focused on commodities like Gold, HRW Wheat, Soybeans, and Wheat for this graph as these particular commodities are representative of key aspects of the economy.

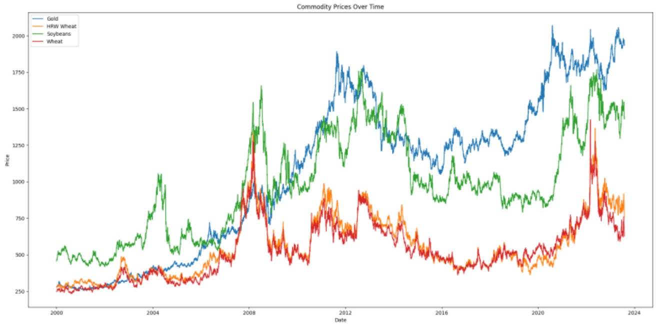


Fig. 3. Commodity Price Trends (2000-2024): Gold, HRW Wheat, Soybeans, and Wheat

The third graph shown in figure 4 presents a comparative analysis between tech giants Microsoft, Google and Apple, as well as incorporating the price trend of Oil. This can highlight potential relationships and connections between these entities within the financial market.

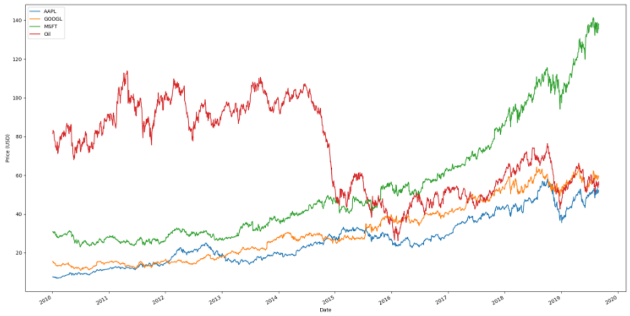


Fig. 4. Comparative Analysis: Apple, Google, Microsoft, and Oil Prices

The fourth graph shown in figure 5 showcases a time series plot of key financial indicators, including the stock prices of fortune 500 companies such as Apple and Google, as well as the performance of prominent indexes such as the S&P 500. Additionally, the plot illustrates the fluctuating nature of commodities like gold and oil.

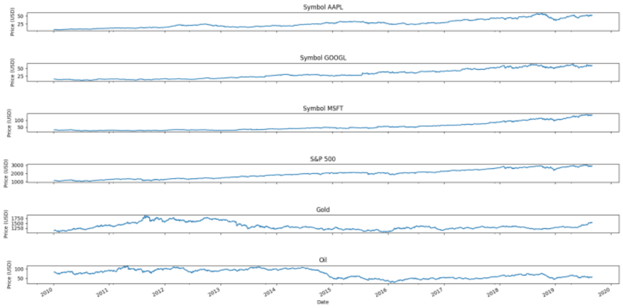
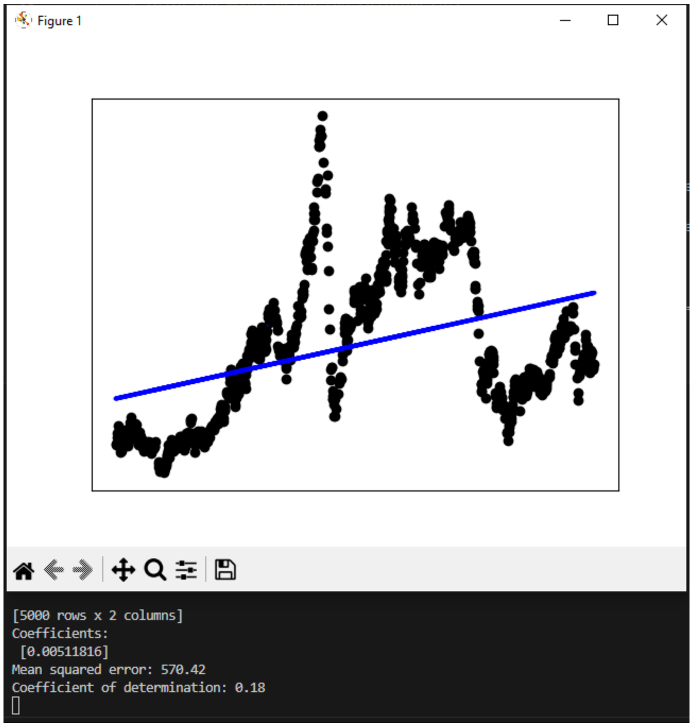


Fig. 5. Time Series Plot for Apple, Google, Microsoft, S&P 500, Gold, Oil

The purpose of the plots is to highlight the multitude of insights one can obtain from the vast set of data within our financial data warehouse. These insights can be the converted into actionable decisions regarding investment or simply understanding the general market sentiment around many stock market elements.

VI. Future Work: Machine Learning Applications for Financial Data Analysis

Using our collected and aggregate data machine learning algorithms can be utilized to predict the price of stocks, bonds, or commodities. Some of the Machine Learning models that could be used to forecast prices are: Support Vector Regression, Neural Networks and Random Forest Regression, and Linear Regression. Using our warehouse we were able to test these models to show our data warehouse's usefulness. Below is the result of a linear regression on the price of gold and it's associated evaluation metrics Mean Square Error and Coefficient of Determinant:



```
[5000 rows x 2 columns]
Coefficients:
 [0.00511816]
Mean squared error: 570.42
Coefficient of determination: 0.18
```

Fig. 6. Linear Regression of gold using data from 1968 to 2023

This prediction had a mean square error of 570.62 and a Coefficient of Determinant of 0.18. This shows that this model does not predict the data well.

The mean square error is used to evaluate how precise the model is. It is the average error for any one data point and is calculated by:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

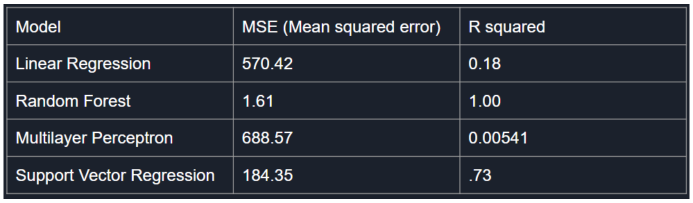
Fig. 7. Mean Square Error formula

The Coefficient of Determinant (also known as R squared) valued tells us how well this particular model fits the test data.A value of 1 means the prediction was accurate and values close to zero means the prediction was not accurate. R squared is calculated by:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \mu)^2}.$$

Fig. 8. R squared formula

The results from tests on the same gold data are:



Model	MSE (Mean squared error)	R squared
Linear Regression	570.42	0.18
Random Forest	1.61	1.00
Multilayer Perceptron	688.57	0.00541
Support Vector Regression	184.35	.73

Fig. 9. model comparison

From these results we can see Random Forest performed exceptionally well in both MSE and R squared at predicting results. The Multilayer Perceptron preformed very poorly. The Support Vector Regression was second in performance and The Linear Regression was third in performance. The Linear Regression shows a slow positive increase in gold prices which would make our expectations for a stable commodity. The bad performance of the Perceptron was unexpected, and more testing with it may be needed with different settings for the model.

VII. Conclusion

This report outlines our development of a well structured, robust and comprehensive financial data warehouse. Leveraging Python scripts to perform our extract, transform and load operations to efficiently and effectively populate our data warehouse with a rich collection of financial data revolving around dimensions such as stock, company, index fund, bonds and commodities, as well as other functional dimensions such as time and currency. Our data collection approach involved utilizing online datasets in the form of csv files for their simple and efficient extractability, as well as API calls when required. This allowed us to populate our data warehouse with approximately 1.8 million rows of data, which in turn can be leveraged to produce actionable insights. Our main focus was ensuring we were able to create a fully functional financial data warehouse and populate it with data from reputable sources, and with that part having been successfully completed, the next steps our team would hope to work on in future involve integrating machine learning techniques to further leverage the data in our warehouse to provide deeper value and potentially predictive insights into the nature of financial markets.

References

[1] SP 500. [Online]. Available: https://data.world/chasewillden/stock-market-from-a-high-level/workspace/file?filename=SP500.csv

[2] DowJones. [Online]. Available: https://data.world/chasewillden/stock-market-from-a-high-level/workspace/file?filename=DowJones.csv

[3] Apple, Microsoft, Tesla, Amazon 2023-2024. [Online]. Available: https://data.world/shamiya-lin/appl-msft-tsla-amzn-23-24-stock/workspace/file?filename=stock_data.csv

[4] Stock Market Dataset. [Online]. Available: https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset?resource=download

[5] Most Popular Historical Data Pages. [Online]. Available: https://www.nasdaq.com/market-activity/quotes/historical

[6] Gold Prices. [Online]. Available: https://www.kaggle.com/datasets/kamyababedi/lbma-gold-prices-1968-2023

[7] Oil Prices. [Online]. Available: https://www.kaggle.com/datasets/mabusalah/brent-oil-prices

[8] Currency. [Online]. Available: https://github.com/datasets/currency-codes/blob/master/data/codes-all.csv

[9] Exchange rate. [Online]. Available: https://app.exchangerate-api.com/dashboard

[10] Repo of relevant data. [Online]. Available: https://github.com/datasets/s-and-p-500-companies/blob/main/data/constituents.csv