DS 5500 Capstone project

**Stock Price Prediction**

Part 1 – Data description:

|  |  |
| --- | --- |
| **Data** | **Apple stock price** |
| Data Range | 2018/01/01 - 2024/01/26 |
| Type | Csv file |
| Interval | Daily |
| volume | 1527 Rows |
| Features | 31 |
| label | 1 (The next day close price) |

Part 2 – Data Clean:

* Use Python or Excel to check whether there’s any missing data, duplication, and outliers.
* For the dataset from yahoo Finance is well structured and with high level of accuracy and quality. there’s no wrong or blank historical data and outliers in this dataset, we don’t need to do much data clean.
* Transformation: Change the date type to standard for future use.

A table with numbers and symbols

Description automatically generated

Part 3 – Data Preprocessing:

3.1 Stock Technical Indicators (STIs)

STIs are statistical calculations based on the price, volume, or significance for a share, security or contract. These does not depend on fundamentals of a business, like earnings, revenue, or profit margins. The active stock traders and technical analysts commonly use STIs to analyze short-term and long-term price movements and to identify entry and exit points. Technical indicators can be useful while predicting the future prices of assets so they can be integrated into automated trading systems. There are two basic types of technical indicators: Overlays and Oscillators.

3.2 Feature extraction from data

In our solution, we consider only the closing price of AAPL. From the data, we calculate 30 indicators. including A simple moving average (SMA), An exponential moving average (EMA) Relative strength index (RSI), The Momentum Indicator (MOM), Stochastic oscillator (SO), The accumulation/distribution indicator (A/D), The Commodity Channel Index (CCI), Williams percentage range (W%R), Moving average convergence divergence (MACD). Formula for calculating the most prevailing Stock Technical Indicators (STIs) is presented in Table.

A table with numbers and letters

Description automatically generated

Part 4 –Summery:

4.1 Methodologies Used for Data Cleaning and Preprocessing

We utilized Python to identify and manage null values and outliers within the dataset. Our analysis confirmed the absence of null values and outliers, ensuring data integrity. Additionally, we employed the Python package "stockdataframe" to compute 30 financial indicators derived from the daily closing prices. These indicators were combined with the closing price itself to form a comprehensive dataset with 31 features.

4.2 Specific Techniques Applied to Enhance the Dataset's Quality and Reliability

The dataset spans from January 1, 2017, to January 26, 2024. To ensure the quality of our features, such as the 100-day Commodity Channel Index (CCI), which necessitates historical data, we utilized records starting from January 1, 2018. This approach guarantees that all features are calculated with sufficient historical depth, thereby maintaining the dataset's quality and integrity.

4.3 Any Transformation or Normalization Processes Applied to the Data

We standardized the date format within the Python dataframe to ensure consistency and facilitate future analyses. This standardization is crucial for time series analysis, allowing for more straightforward manipulation and comparison of dates.

4.4 Challenges Encountered During Data Cleaning and How They Were Addressed

Calculating certain financial features from scratch presented challenges due to the complexity of their formulas and the need for historical data. We overcame these challenges by leveraging specific Python packages designed for financial analysis, such as "stockdataframe," which streamlined the computation of these advanced features.

4.5 Visual aids, like charts or screenshots, to illustrate the process where applicable.

A grid of blue and red squares

Description automatically generated

The correlation heatmap above visualizes the relationships between the different features in AAPL stock dataset. This heatmap is useful for identifying which features are strongly correlated with each other. Strong correlations between features can indicate redundancy (if two features are highly positively correlated) or inverse relationships (if they are highly negatively correlated). This information is valuable for feature selection, especially if we're planning to use this data for predictive modeling, as highly correlated features can affect model performance.