Name: Daniel Nehemiah Peter Katam

Stock Market Analysis Using Apache Spark

Introduction

we delve into of real-time daily stock data, specifically focusing on the symbol "Amazon." The data is sourced directly from the Alpha Vantage API, a prominent financial data provider known for its accuracy and timeliness. Our overarching goals in this analysis revolve around gaining a profound understanding of the stock's trends, employing thorough exploratory data analysis (EDA) techniques, and ultimately constructing a robust machine learning model for stock prices.

To accomplish these goals, we make use of real-time data, which allows us to obtain the most recent data regarding Amazon's daily stock performance. By using the Alpha Vantage API, we can make sure that our research is based on current and dependable market data, which allows for a more accurate representation of the stock's activity.

Data Source:

- Real-Time API: Use an API to fetch real-time data (e.g., stock market data API).

Tasks

- Fetch real-time daily stock data from Alpha Vantage API.
- Transform data into a structured Pandas Data Frame.
- Set up a Spark environment for streaming analysis.
- Conduct EDA to gain insights into stock trends.
- Build and evaluate a machine learning model for stock price prediction.

Your approach and architecture design:

We will utilize the Alpha Vantage API as the primary source for obtaining real-time daily stock data for the symbol "Amazon." This API offers a comprehensive set of financial data, including historical stock prices. We will implement a script to interact with the Alpha Vantage API, fetching relevant daily stock data for Amazon. Ensure that the data retrieval process is automated and scheduled to maintain up-to-date information. The data is first converted to a Pandas Data Frame, which includes column renaming, appropriate indexing, and data type conversion. We Perform data cleaning to handle missing values, outliers, and any inconsistencies in the acquired stock data. Ensure uniform formatting for dates, prices, and other relevant metrics. Now we Derive additional features that may contribute to the predictive power of the machine learning model. Conduct descriptive statistics to summarize key characteristics of the data. Generate visualizations, including time series plots, candlestick charts, and other relevant graphs to explore trends, seasonality, and anomalies in

the stock data. After that, this Data Frame is transformed into a Spark Data Frame for additional examination. We then will initiate a Spark session, build a app and implement Window Operations. Lastly we will split the data and build regression and predict on test data and check for the performance of the model built.

Challenges encountered and solutions implemented.

Data Quality Issues:

- Challenge: Incomplete or inaccurate data retrieved from the Alpha Vantage API.
- Solution: Implementing robust data cleaning and preprocessing steps. This involved handling missing values, outliers, and ensuring data consistency. Regularly check and update the data retrieval process to address any changes in the API structure.

Feature Engineering Complexity:

- Challenge: Deriving relevant features that contribute meaningfully to the model's predictive power can be challenging.
- Solution: Experimented with a variety of features, including technical indicators, market sentiment data, or macroeconomic factors. Leveraged domain knowledge and collaborate with financial experts to identify and incorporate relevant features.

Data Overfitting in EDA:

- Challenge: Drawing incorrect conclusions from exploratory data analysis due to overfitting to historical data patterns.
- Solution: Separate the data used for EDA from the data used for model training and testing. This ensures that insights gained during EDA are more likely to generalize to unseen data.

Insights and findings from the data analysis.

- By using API package we were able to load the live data into a dataset without error
- We saw that there were no null values, duplicate values and the data is ready for our spark application building and for model building
- We have then transformed the data index to column by resetting the index
- Below are the list of columns used for this project
  ['Date', 'Open', 'High', 'Low', 'Close', 'Volume'], dtype='object')
- We have then used cufflinks package to plot graphs for close, open and high from them we can infer the incremental direction of the stock prices going upwards.
- We also plotted a shaded plot to see the distribution of then dataset
- We have also used seasonal plot to check for the seasonal decompose of the close column in additive process giving period = 30. From that we could see the fluctuation in the close column and how the stock prices

have change till now. From this chart the main things which we were able to analyse are the Trend and Seasonality

- We also used the 50_MA, 200_MA by using the rolling window function to check the spread of the data for past 50 and 200 days

Details and evaluation of the machine learning model and any notable data patterns or anomalies identified.

We have first installed the Java and Pyspark (Apache Spark) into our notebook to build spark app and java to eliminate any Exceptions errors which may raise.

Then built an app **"StockDataAPP"**

Then loaded the dataframe into the app

Then defined the window function and implemented into the app to check for the descpritive analytics of the dataset

We also checked for the schema of the app built

We used Machine learning to build linear Regression model on training dataset, Used VectorAssembler to divide the columns as per the requirements. We then used built model on test dataset and used RMSE values and R2-sqaured values to check the accuracy of the model build. From the output we can see that the model was built with 98.85 ~ 99% confidence. Which tells us that we don't have to perform the optimization technique on the model.