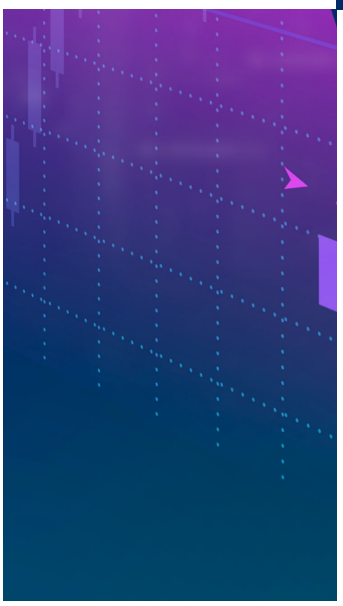

CloudAngles
mlangles
mlangles
Predictive AI
Stock Price Prediction Use Case - Prophet

About mlangles Predictive AI

mlangles is a comprehensive AI platform designed to manage the lifecycle of data and models, offering streamlined solutions for every stage of the process.

Through its Predictive AI component, mlangles provides a suite of tools to navigate efficiently through each phase of AI project development, encompassing data engineering, development, deployment, and monitoring. It facilitates continuous integration, continuous deployment, continuous training, continuous monitoring (CI-CD-CT-CM), enabling enterprises to effectively manage their AI initiatives.

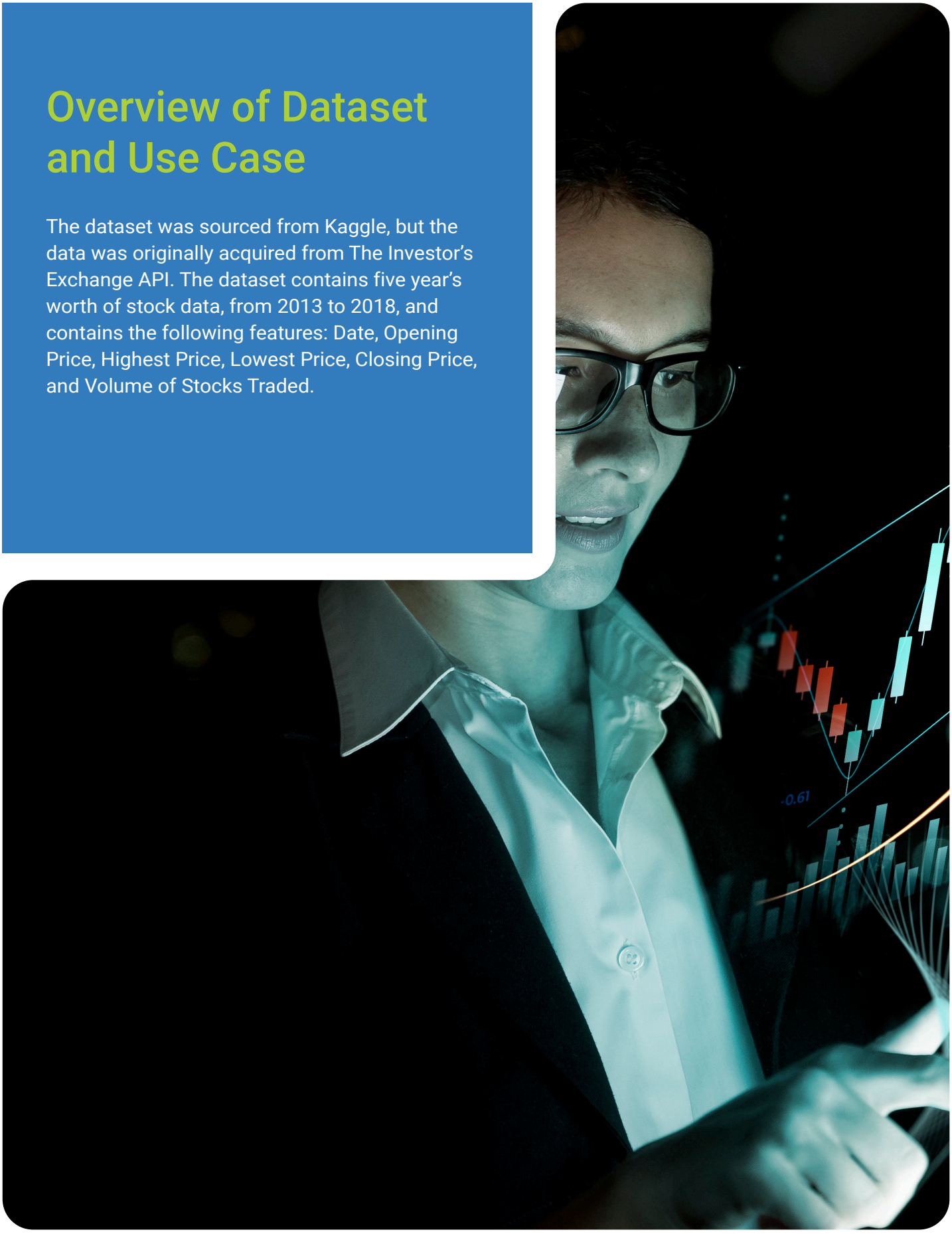


Objective of the Use Case

We are trying to create a machine learning model that can accurately forecast future stock prices based solely on historical stock price data. The objective is to aid investors in making statistically informed trades.

Overview of Dataset and Use Case

The dataset was sourced from Kaggle, but the data was originally acquired from The Investor's Exchange API. The dataset contains five year's worth of stock data, from 2013 to 2018, and contains the following features: Date, Opening Price, Highest Price, Lowest Price, Closing Price, and Volume of Stocks Traded.



Working of the Use Case

Step 1: Data Pipeline

Extract Data: Loaded data from the source (source was in GitHub Repository)

Data Analysis: Performed an analysis of the data to get an overview of the data and determine whether there were any issues with the data.

Data Preprocessing: Prepared data for model development and feature engineering. Dropped redundant data column: name.

Data Visualization: Displayed two graphs, a line graph and a candlestick graph. The line graph focused on the closing price performance of the stock and the candlestick graph showed the entire price movement of the stock.

Feature Engineering: Final preparation of data before experiments are run. In this case, not much was done, just dropped a few more feature columns: Open, High, Low, and Volume.

Data Versioning:

- Various processed data versions can be generated through different transformations applied to the same raw dataset, such as deleting columns or applying various transformations on specific columns.
- Throughout the data pipeline, diverse transformations can be executed at each iteration. Consequently, the resulting data at the pipeline's end is systematically versioned.
- Given that each version of the final data is distinct, models trained on these different versions will exhibit varying behaviors.

An overview of the data Pipeline

fig_1 fig_2

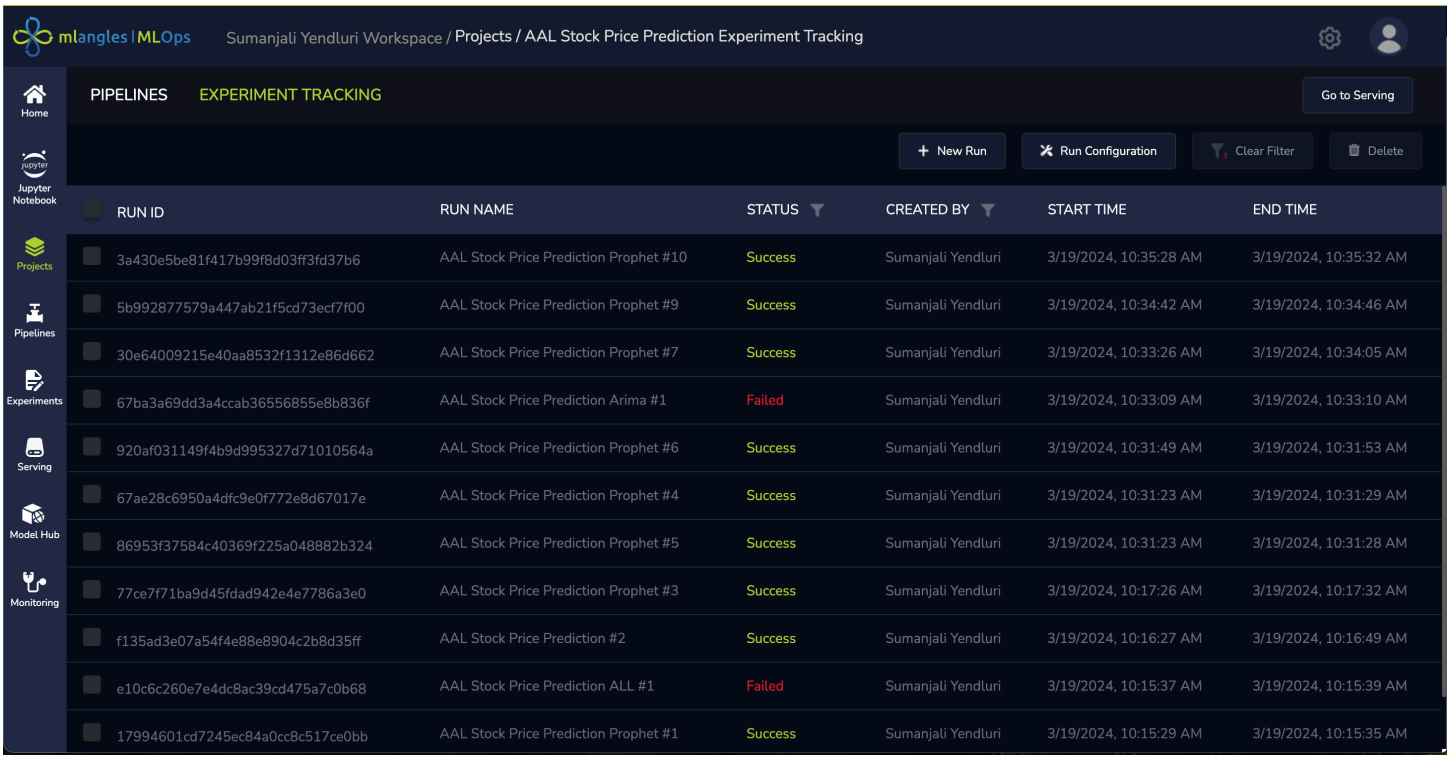
Fig_1 is a line graph displaying the performance of the AAL stock's closing price over a 5-year period.

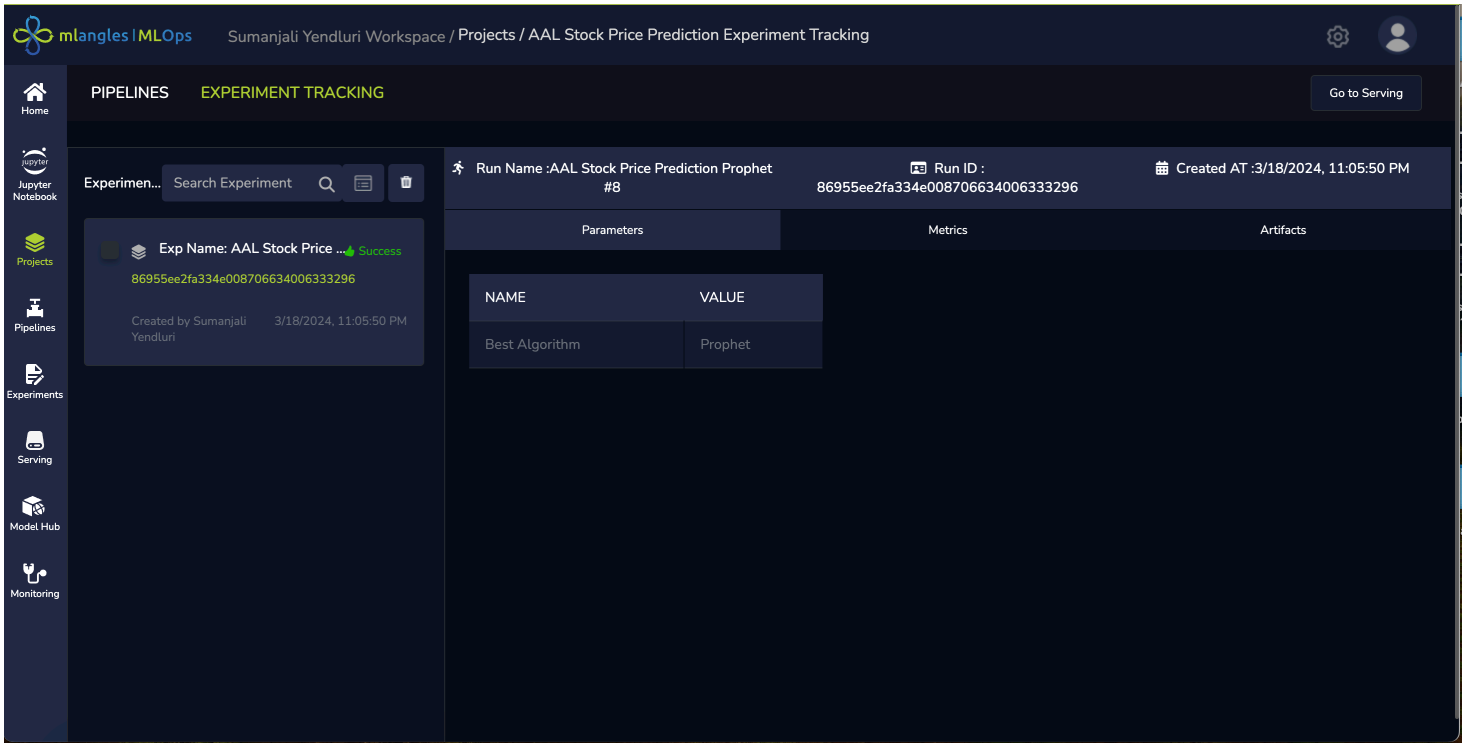
Fig_2 is a candlestick graph displaying the entire price movement of the AAL stock.

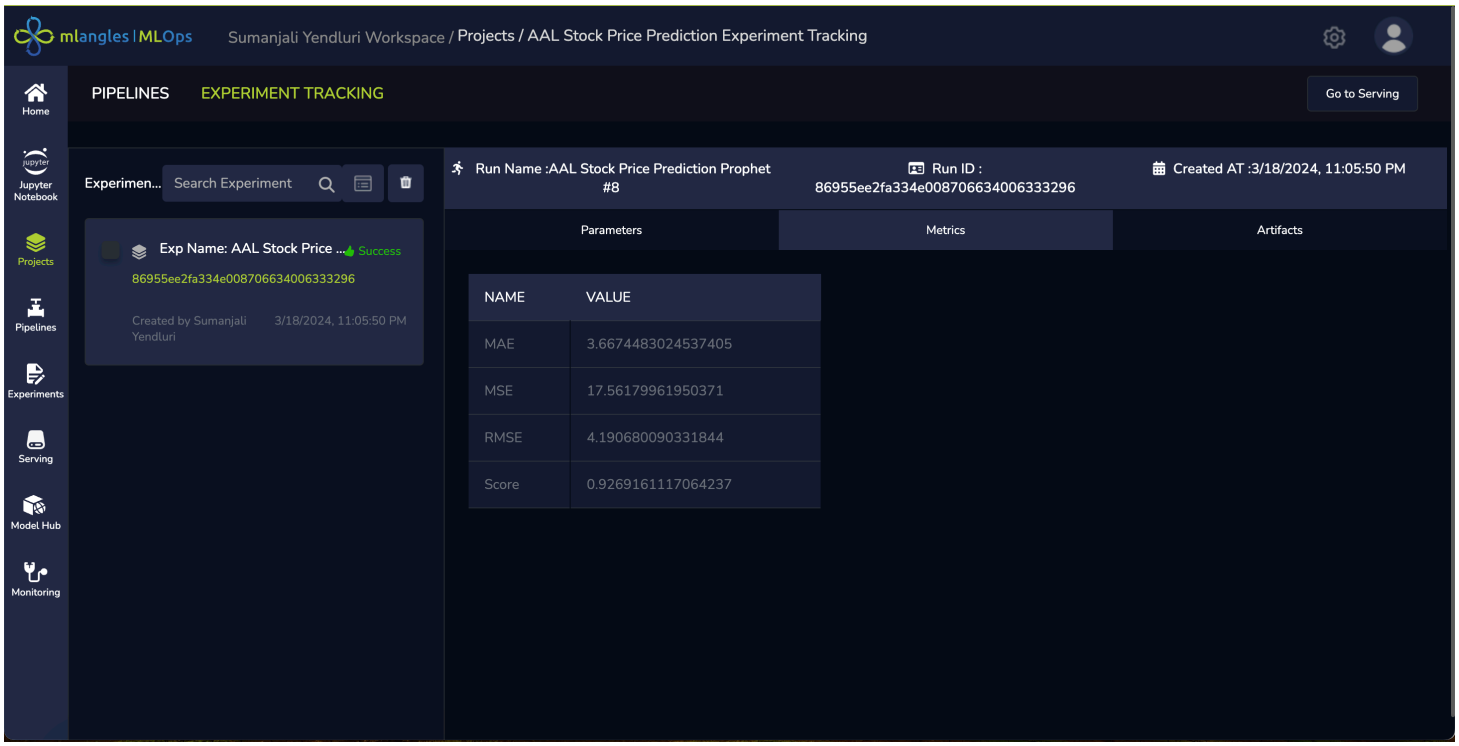
Preview of the data after the data pipeline has been run to completion.

Step2: Experiment Tracking

Multiple experiments were run using the Prophet machine learning model. After comparing the accuracy and RMSE of each experimental run, it was observed that the best results were from using Prophet without any hyperparameters. In the end, the best RMSE was 4.31 with a score of 0.92.





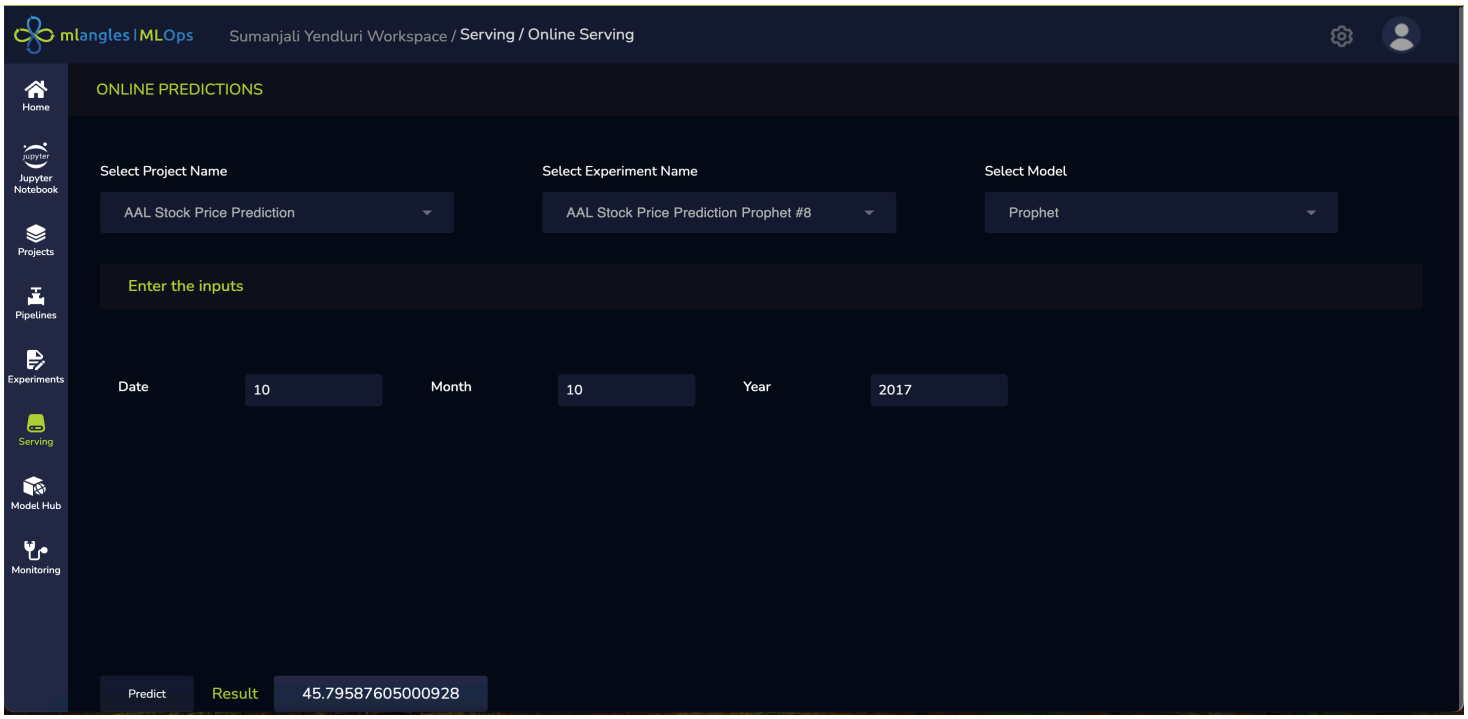


Model Versioning:

- Models are sensitive to a plethora of hyperparameters and parameters, including learning rate, loss function, and optimizers.
- Consequently, a model selected for training, with both the model and final data versions remaining constant but changes in parameters, may yield differing performance metrics.
- These diverse model versions can be uploaded to the model hub, facilitating the management of multiple iterations and variations.

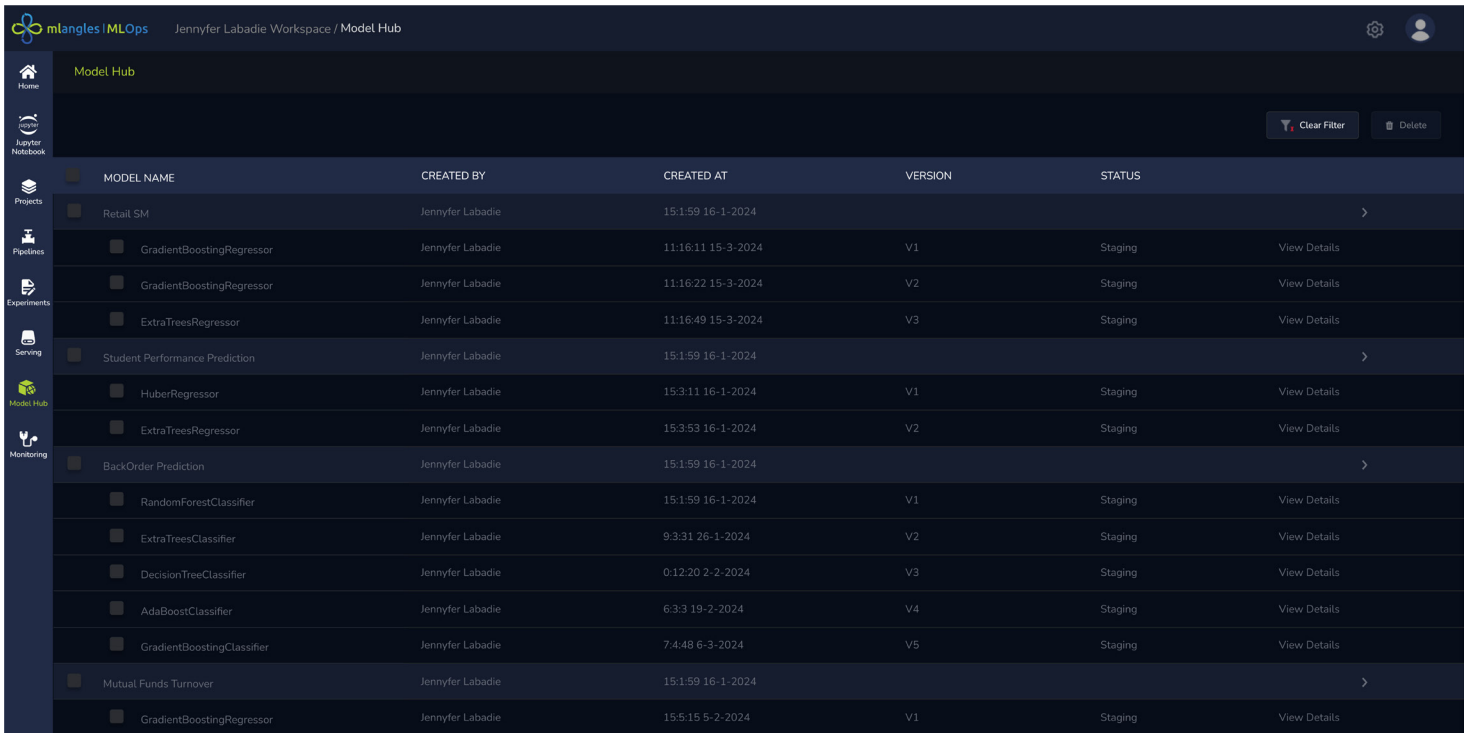
Step 3: Serving

Performed an online prediction using a random date from within the timeline of our data. The forecasted closing price for the stock on October 10, 2017, is $53.09, while the actual closing price for that day was $53.03. Our prediction was off six cents.



Model Hub:

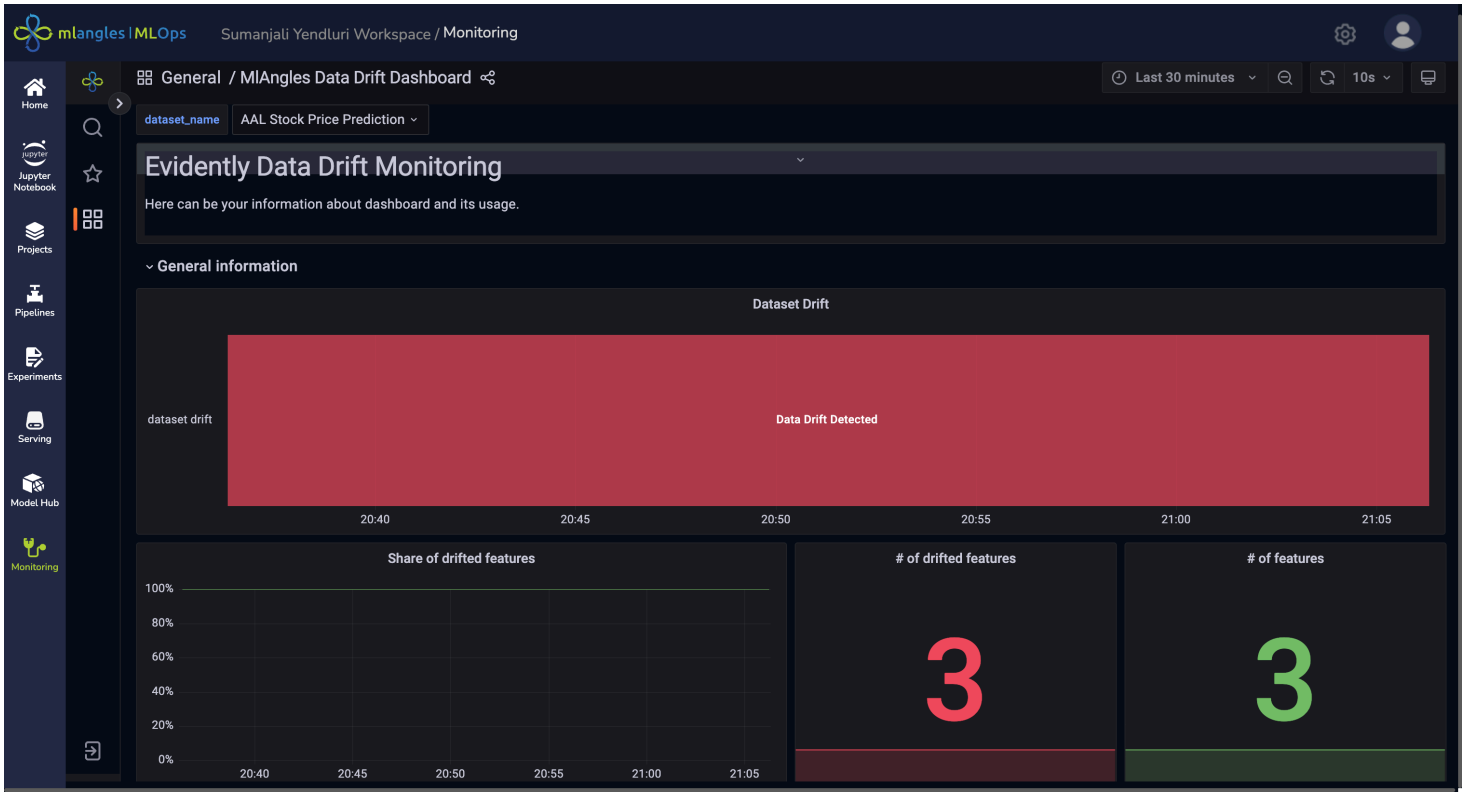
- Trained models are uploaded to the model hub, whereupon deployment, a REST API endpoint is automatically generated.
- Data is transmitted to this endpoint as a request, triggering the model to execute a prediction and return the output as the response to the request.



Step 4: Monitoring

Data drift is the gradual or abrupt change in the statistical properties of training data in a machine learning model. There are three types of data drift: sudden drift, gradual drift, and seasonal drift. Sudden drift occurs when there is a sudden and abrupt change in the distribution of data. Gradual drift, the opposite of sudden drift, is a slow, steady change in the distribution of data. Finally, seasonal drift is when shifts in the data distribution are linked to time-based patterns. Data drift can cause several issues with machine learning models, including reduced model performance, loss of generalization, and model degradation.

We can see here in the monitoring screen that all the features have drifted. Since these features are related to financial matters, which are dependent on a multitude of extraneous factors including seasonality, the data features are prone to data drift. In the future, we can control this issue with continuous monitoring and model retraining.



Conclusion

In conclusion, we can see that the Prophet model is effective at forecasting stock prices. However, the model is also prone to data drift, requiring continuous monitoring and retraining of the model. Still, with care, this model holds excellent potential for use in the investment market.