Forecasting Implied Volatility of Advanced Micro Devices Stock Options

Team: Lam Lay, Vlasios Mastrantonis, Ramachandra Rahul Taduri, Nha Tran, Nigel Tucker
GitHub: https://github.com/nha-tran-lsu/spring-2024-stock-options-analysis

Abstract:

We aim to forecast the implied volatility of the Advanced Micro Devices (ADM) stock option. Our dataset spans a little over two years, starting on January 4, 2021, and ending on February 28, 2023. Our base models consist of a Rolling Average model and two ARIMA models of parameters (0,1,5) and (0,0,5), respectively. For our primary model, we opt for a Random Forest model trained on an 80:20 random split. We find that the Random Forest significantly outperforms the base models.

Introduction:

Navigating the complexities of the stock market, especially for those with limited time and capital, presents significant challenges. However, an emerging solution, the **Options Market**, has gained traction in recent years. Unlike traditional stock trading, options offer traders the opportunity to buy and sell contracts tied to specific stocks, providing substantial leverage at a fraction of the cost.

But what is an option? An option is a financial contract between two traders granting the buy or sell of a specific stock at a predetermined price from (or to) the trader who sold the option, by a specific date, regardless of the current market value of the stock. The seller of the option is obliged to fulfill the requested order at the agreed-upon price. Options can be split into two types: 1) a **call option,** which grants the right to buy a stock, and 2) a **put option**, which grants the right to sell a stock. The predetermined price is known as the **strike price**, and the specified date is referred to as the **expiration date**.

For instance, imagine purchasing a put option for AMD stock with a strike price of $100 at $5. If, before expiration, the stock price falls to $90, exercising the option allows selling it at $100 per share, resulting in a net profit of $5 per share ($10 profit over the stock price minus $5 the premium). This flexibility, combined with strategic planning, offers the potential for significant returns compared to traditional stock trading.

Indeed, the price of a given option is intricately tied to the fluctuations in the underlying asset's value. Among various factors influencing the price, the implied volatility represents the expected volatility of the underlying asset, the stock price. Hence, it is essential in the financial portfolio's risk management. In periods of high volatility, traders can implement strategies such as *iron condors* (which entail lower risk but also lower return) or *short strangles* (involving higher risk but offering higher return) to capitalize solely on the heightened volatility. Conversely, when volatility is low, traders may opt for strategies like *short or long vertical spreads* to profit from the subdued volatility. Consequently, by accurately modeling and predicting the implied volatility of the stock, traders can enhance their ability to execute profitable trades with optimal strategy selection and timing.

As a result, our goal is to **forecast** the **implied volatility** (Ganti et al., 2024) of stock option contracts. To this end, we opt to analyze the historical stock options data of **Advanced Micro Devices, Inc. (AMD)**. AMD is a publicly traded company that presents an ideal candidate for our project due to its involvement in a highly volatile industry. Despite that, AMD has exhibited a comparatively lower and more consistent growth rate compared to other companies in the same sector. Using data from AMD allows us to build a model that captures the nuances of a volatile market while avoiding potential overfitting issues associated with data from more volatile companies such as **NVIDIA**, which may not generalize well to the broader industry landscape.

Data: Historical AMD Options Dataset from Wharton Research Data Services (Wharton Research Data Services. (n.d.), n.d.) & AMD Stock Price (Yahoo Finance, n.d.) and Interest Rate Datasets from Yahoo Finance (Yahoo Finance, n.d.).

KPIs: There are several key performance indicators (KPIs) to consider to assess a machine-learning model's performance. These include:

- Accuracy: This measures how correct the model's predictions are overall.
- Precision: This is the proportion of true positive predictions among all positive predictions made by the model.
- Recall: This measures the ratio of true positive predictions to all actual positive instances in a dataset.
- F1-Score: This is a single metric that integrates precision and recall to provide a better understanding of the model's performance.
- Mean Absolute Error (MAE): This is the average difference between predicted and actual stock option values, regardless of direction.
- Mean Squared Error (MSE): This measures the mean of the squared prediction errors, measuring variance.
- Root Mean Squared Error (RMSE): This measures the average size of the prediction errors.

Stakeholders: Traders involved in equity option exchanges, such as Day Traders, Hedge Funds, Prop Trading Firms, etc.

Data Processing

The nature of option data includes categorical features such as Standard Industrial Classification code (sic), exchange identifier (exchange_d), industry group (industry_group), etc., and numerical data of options contracts such as strike price (strike_price), expiration date (exdate), the Greeks (alpha, delta, gamma, vega, theta), implied volatility (sigma), etc.

To begin cleaning the data, we removed columns containing only null values and the rows with null values that remained. Next, we removed columns with unnecessary or repeated information that would not play a role in predicting volatility. We converted numbers and date-time objects for calculation purposes. Given that the option data did not come with the stock price of the underlying asset or the daily risk-free rate, we wrote functions to download and store this information from Yahoo Finance. Before merging these two data frames, we calculated and added financial metrics such as the daily returns of the stock as well as the 30, 60, and 90-day rolling volatility averages to the stock price data frame.

We merged the stock data, risk-free rate, and options data and deleted columns we would not use. We chose only to follow the adjusted closing price of the stock, as this is often the standard protocol for examining historical returns or doing a detailed analysis of past performance. We renamed any necessary columns to standardize our naming convention for the dataset's features. We calculated and added the days to expiration and bid/ask spread. Lastly, the index for the data was set to the date column, the values were sorted by their symbol, and the cleaned file was saved to a new CSV file.

Exploratory Data Analysis

To explore the cleaned data, we split it between the two types of options, call and put options, to better observe the data set and features' relation. The features may behave differently between call and put options.

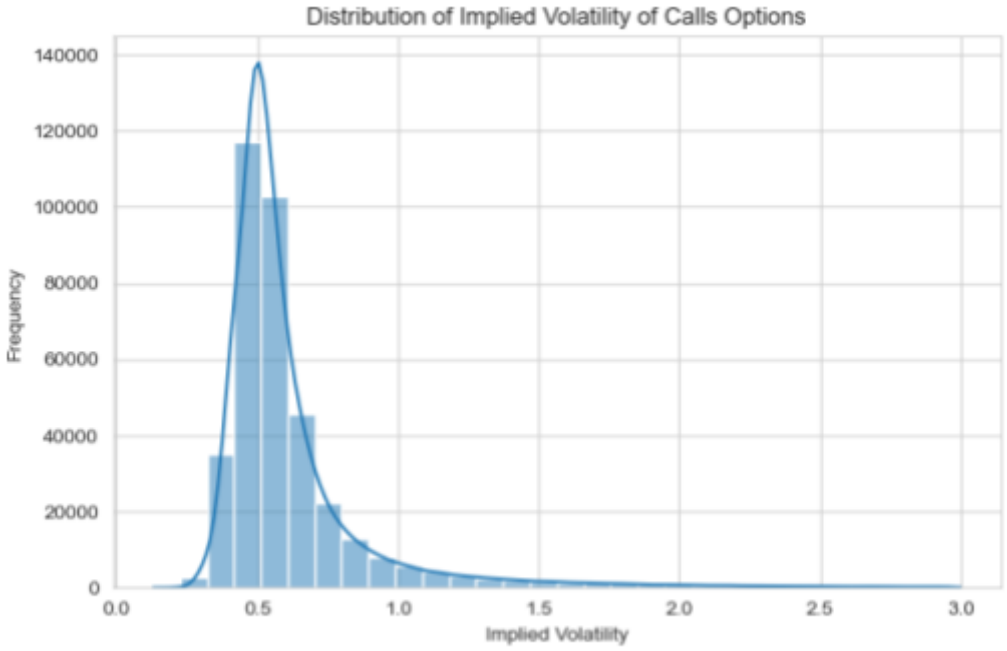


Fig. 1: Distribution of implied volatility of call options

We noticed that the distribution of implied volatility is right-skewed, as in Fig. 1 above. Next, we construct lists that will store the information to graph individual options by their expiration date and symbol. Beginning with calls, we created a scatter matrix to understand better relationships between implied volatility and the other features such as best bid, best offer, days to expiration, volume, open interest, and the options Greeks (delta, gamma, etc.). We built a function to calculate the correlation between the features in our dataset and plotted the results using a heat map, as shown in Fig. 2 below.

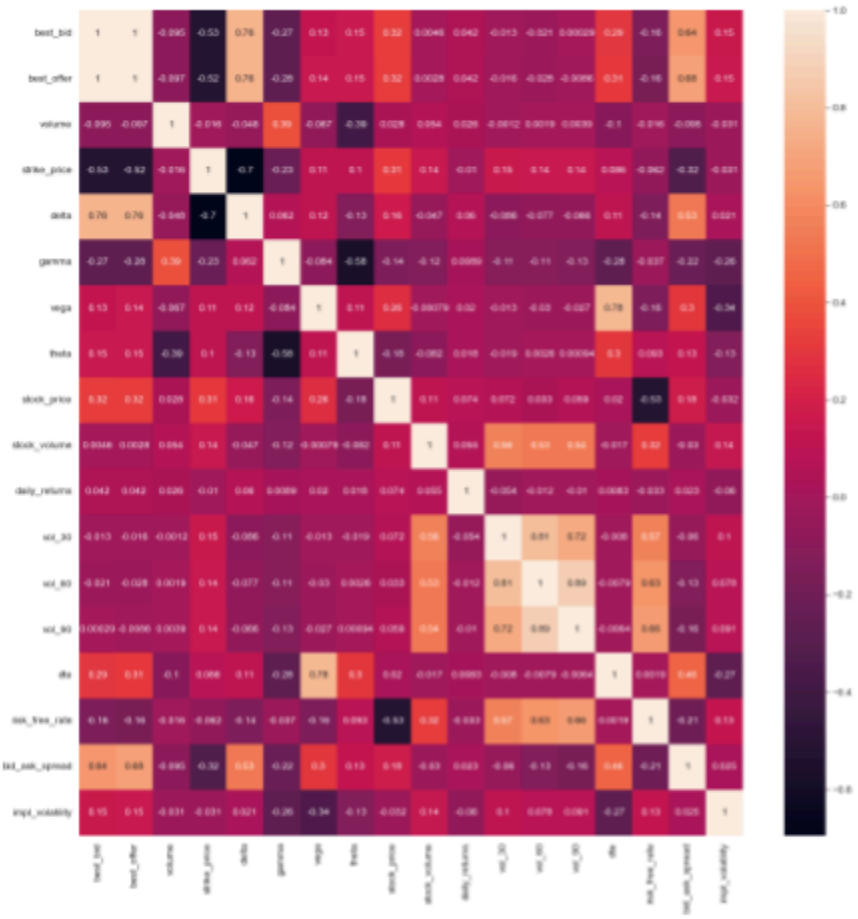


Fig. 2: Correlation matrix of features of call options

From this, we determined that contract size had. Next, we graphed the relationships between the implied volatility and the rest of our features for options with a strike price that was closest to the stock price for that date, as these options tend to have more volume and open interest, and therefore, more data points. The created figures displayed the nuances of different options in the scatter matrix above; for instance, we see that individual options' bid and offer prices have a more linear relationship with implied volatility than logarithmic ones. We added a hue to illustrate if time played an important role in the relationships between implied volatility and the other features.

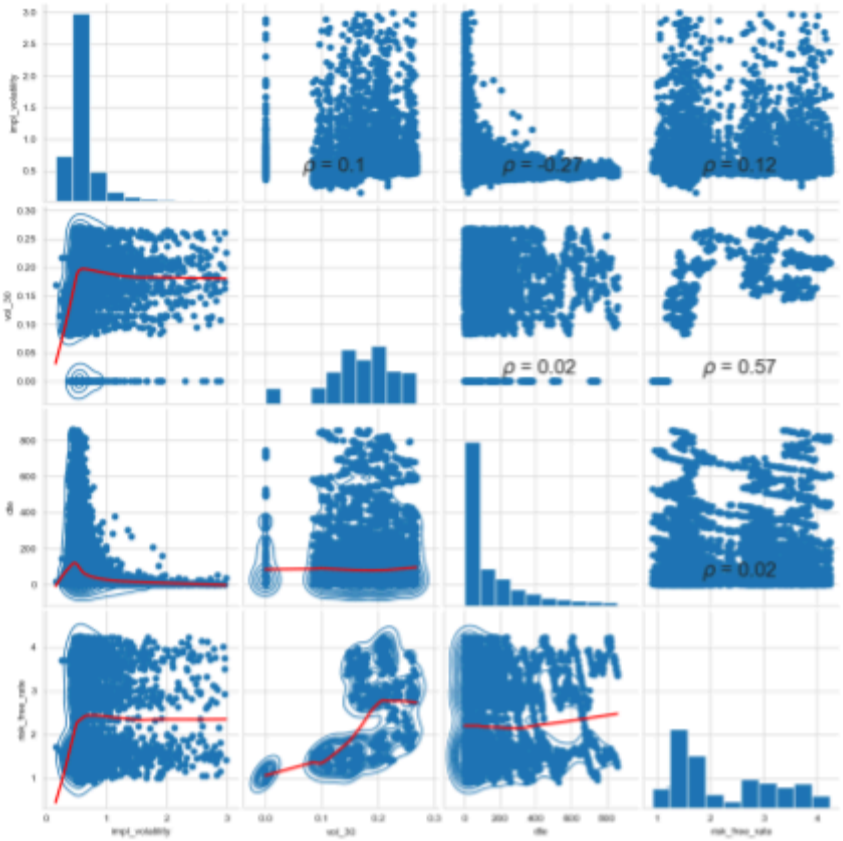


Fig. 3: Pairplot of different features

Upon completing the exploratory data analysis for calls, we repeated the same process for puts and accounted for the differences when constructing our model. We further examine the relationship between implied volatility and other features by plotting Pairplot, which shows the nonlinear relationship between the features. Therefore, machine learning models such as Random Forest, Gradient Boosting, Support Vector Regression, and Neural Networks may be appropriate.

Strategy

Our predictive models aim to estimate volatility. Our base model of choice is the **AutoRegressive Integrated Moving Average (ARIMA)** due to its simplicity of implementation and comprehension. Moreover, it relies solely on the past values of the variable under consideration, rendering it independent of other variables. An ARIMA model is a widely used time series forecasting model. It is composed of three main components:

1. **AutoRegressive Term (AR)**: This part models the relationship between an observation and a number of lagged observations (i.e., its own past values). It implies the current value of a series is dependent on its previous values.
2. **Integrated term (I)**: This component involves differencing that time series data to make it stationary. Stationarity means that the statistical properties of the series (like mean and variance) remain constant over time.
3. **Moving Average term (MA)**: This part models the dependency between an observation and a residual error from a moving average applied to lagged observations.

In our scenario, train two models: one with parameters AR = 0, I = 1, and MA = 5, and another with AR = 0, I = 0, and MA = 5. Furthermore, to streamline the process, for options trading on the same date (after price calculation), we aggregate all entries into one, trading with the mean volatility.

To assess the model's performance, apart from visual inspection of the graphs, we use three metrics: the **Mean Absolute Error (MAE),** which is the average of the absolute differences between forecasted values and the actual values (lower values indicate better accuracy), the **Mean Squared Error (MSE)**, that is the average of the squared differences between forecasted values and the actual values (lower values indicate better accuracy), and the **Root Square Error (RSE)**, which is the root square of MSE and is in the same unit as the original data.

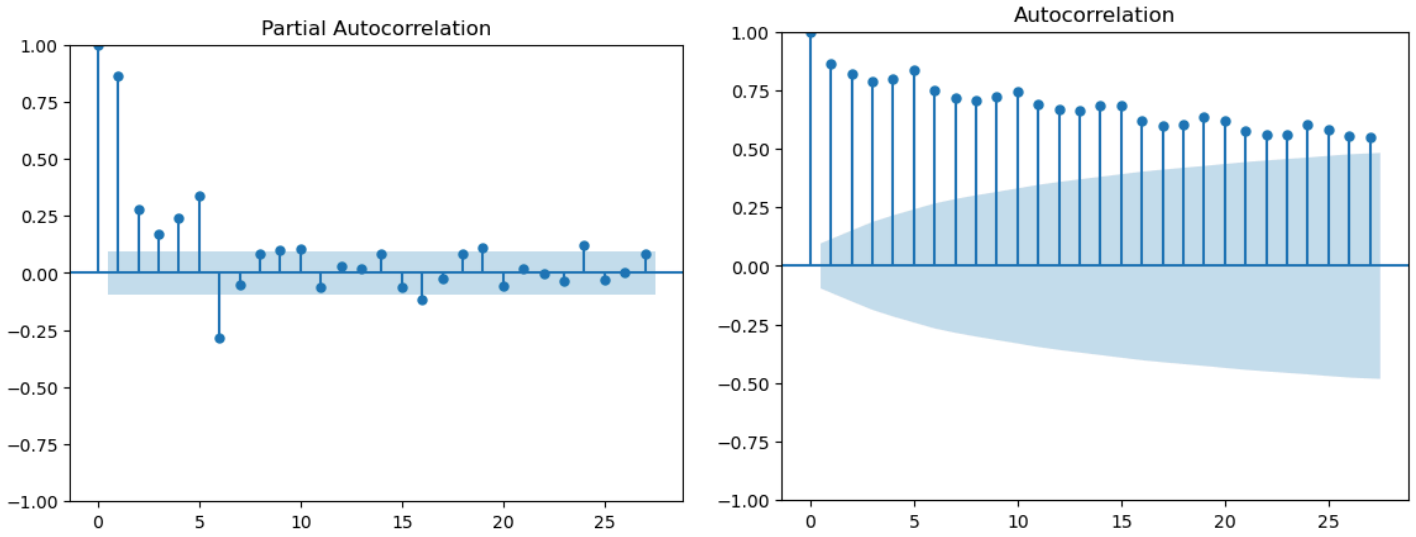
Given some of the linear relationships observed in our visual analysis, we first trained our data on a Linear Regression model and compared it to our baseline models. We found that it was not very effective for analyzing our data. Given that our data was both continuous and categorical in nature we chose to train a **Random Forest Regressor** model next. The train and test sets for these models are obtained by randomly splitting the dataset using the 80/20 split rule. For feature selection, we calculated the correlation of each variable with the ground truth variable - Implied Volatility. We removed the variables that have very low correlation with the Implied Volatility, particularly the ones with correlations between -0.01 and 0.01 since they most likely will not have much effect on predicting Implied Volatility.

We set the maximum depth of each decision tree to be 20 and to help identify the optimal number of n_estimators, or number of trees in the forest, began by training simple Random Forest models with n_estimator ranging from 1 to 30. The optimal value was discovered to be about 20, so we re-trained the model using 20 as our **optimal n_estimator** and made predictions with this model. These predictions were more accurate than our previous models, a fact illustrated by the differences in MAE, MSE, and RSE below. We also constructed an R-squared plot to display the correlation between the observed implied volatility values and our model's predicted implied volatility values.

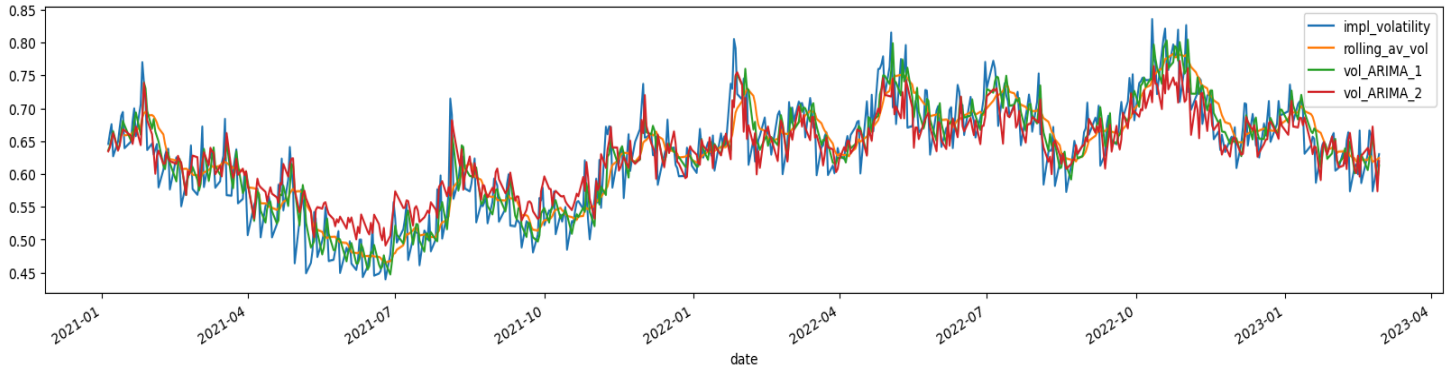
Results

Rolling Average and ARIMA models

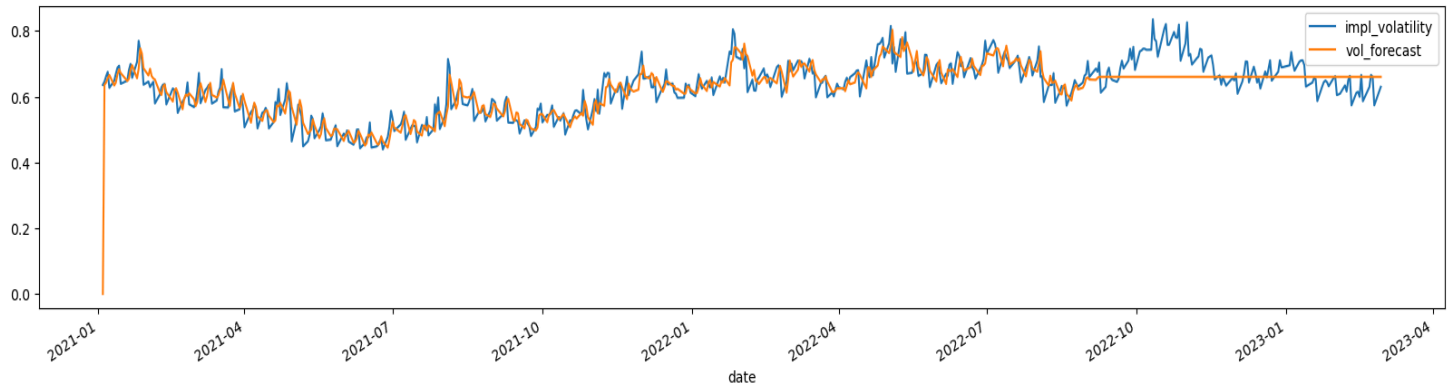
For our baseline models, we start with a simple rolling average with rolling parameter 10. For the ARIMA model, in order to select the parameters we compute the autocorrelation and the partial autocorrelation of the time series:



In addition, the p-value of the ADF test is about 0.47 further indicating highly non-stationary data. To this end, we compute the first difference time series, for which the p-value from the Augmented Dickey-Fuller (ADF) test is near zero. Therefore, we decide to take I=1 in the ARIMA model. In fact, we train two ARIMA models, one with parameters (0,1,5) and another one with parameters (0,0,5).

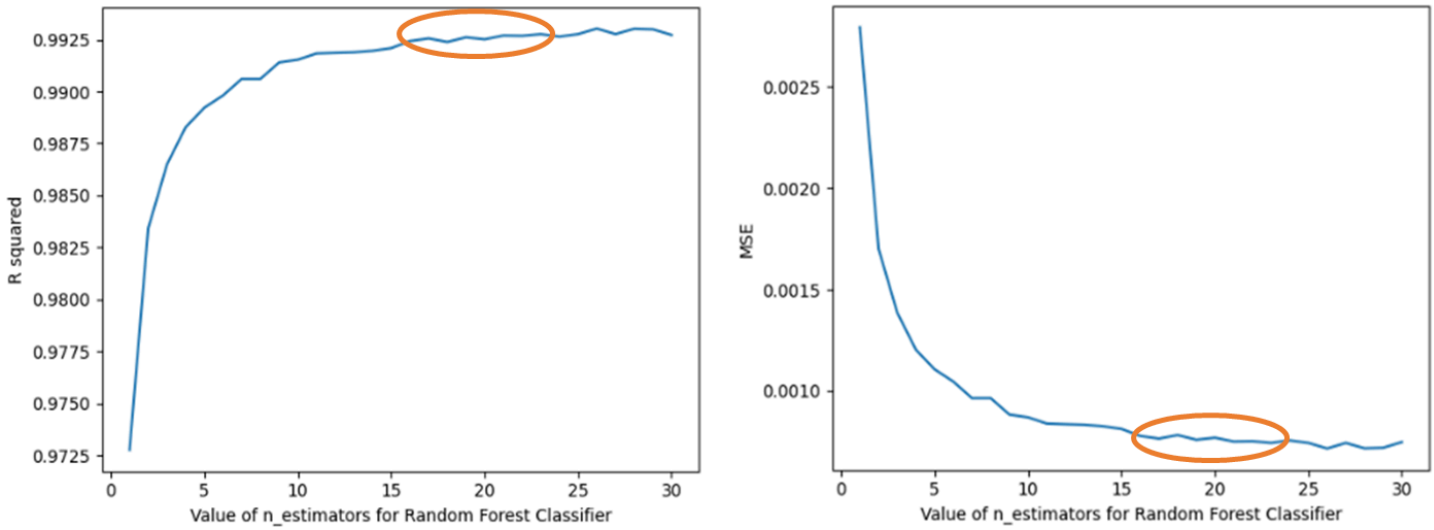


All three baseline models are trained on the whole data set. It is well-known that ARIMA models are not good predictors of long-term behavior as they quickly converge to constant values. We verify this in our case, by training an ARIMA model on a 80:20 split (2021-01-04 to 2022-08-31 for the train set, and 2022-09-01 to 2023-02-28 for the test set).

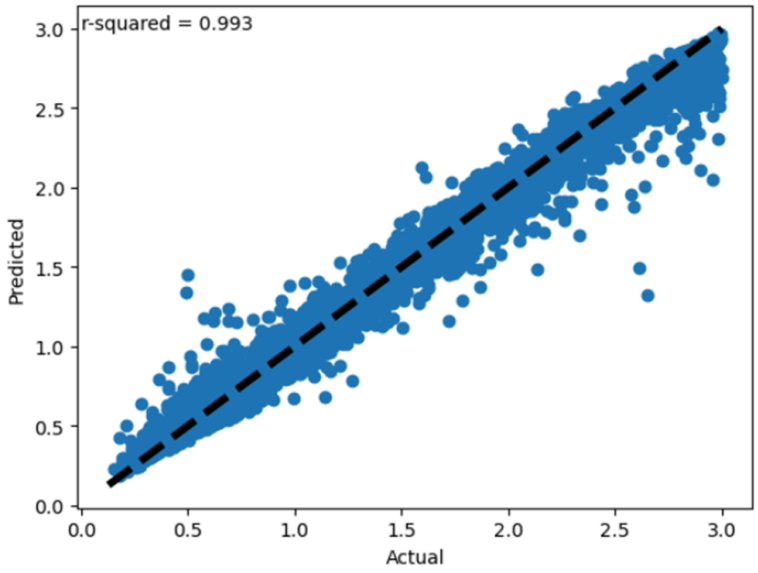


Random forest model for volatility

To find the optimal n_estimator, we trained simple Random Forest models with n_estimator ranging from 1 to 30. We randomly split the dataset into a train set and a test set using the 80/20 split rule. After training a random forest with each n_estimator, we make predictions on the test set and calculate R squared and MSE. These two metrics can be found in the following dataset.



From the plots above, the optimal n_estimator is around 20. We retrain the random forest with this n_estimator with the same train set and test set used in the optimization process. After the prediction, we calculated performance metrics (MAE, MSE, RSE, and R squared) to evaluate the performance of the model. The correlation between actual and predicted implied volatility is provided below, with an R squared of 0.993.



	Actual Volatility	Predicted Volatility
count	147020.000000	147020.000000
mean	0.632206	0.631877
std	0.320403	0.314966
min	0.157611	0.186992
25%	0.470184	0.472746
50%	0.537073	0.538506
75%	0.662465	0.661698
max	2.999485	2.969903

Model comparison

Implied Volatility	Rolling Average	ARIMA(0,1,5)	ARIMA(0,0,5)	Linear Regression	Random Forest
MAE	0.0280	0.0279	0.0323	0.1486	0.0140
MSE	0.0279	0.0019	0.0016	0.063	0.0007
RSE	0.0323	0.0444	0.0407	0.2516	0.0277
R-squared	-	-	-	0.3830	0.9925

Next Steps

Potential next steps for this project would be to first standardize it across the semiconductor industry by expanding the model to include training data from other companies and then to build a pipeline between a brokerage firm's API and a local database on your machine that will automatically call the desired information for options from these brokerage firms, store it in your local database, and then import the updated information into the file we use to build and train the model we have constructed. This would allow for our model to process and analyze the most recent data when making predictions for implied volatility. To maximize the analysis this pipeline can perform, scripts for downloading, storing, and analyzing a company's financial statements are added, giving the trader a better understanding of both a company's fundamental and technical financials. Furthermore, once a pipeline for this information is established, a similar technique can be applied to other equity sectors that trade on options exchanges. Of course, in that case, some attention will need to be paid and changes will need to be made within our process as the exploratory data analysis may lead to a different model using a different variation of the features.

References

Ganti, A., Scott, G., & Velasquez, V. (2024, 04 02). *How Implied Volatility (IV) Works With Options and Examples*. Investopedia. Retrieved 04, 2024, from https://www.investopedia.com/terms/i/iv.asp

Wharton Research Data Services. (n.d.). (n.d.). *Option metrics for AMD.* WRDS - Wharton Research Data Services. Retrieved 04, 2024, from https://wrds-www.wharton.upenn.edu/

Yahoo Finance. (n.d.). *AMD historical data*. https://finance.yahoo.com/quote/AMD/

Yahoo Finance. (n.d.). *CBOE Interest Rate 10 Year T No (^TNX)*. https://finance.yahoo.com/quote/%5ETNX/