Stock Image Classification

1. Introduction

This project conducts a comprehensive analysis of historical data obtained from Yahoo Finance. The goal was to predict future stock prices using a combination of historical data and a moving average (MA) model. The project covers data collection, preprocessing, image generation, and the implementation of a Convolutional Neural Network (CNN) for binary classification.

2. Data Collection & Preprocessing

The historical stock data was gathered for multiple companies from January 1, 2010, to December 30, 2019, using Yahoo Finance library. We then used this data, along with a moving average (MA) approach, to make the best predictions for future stock prices.

2.1 Data Preprocessing Steps:

- Data Organization:
    - Stock data is structured into a pandas dataframe for systematic analysis.
    - Date and Moving Average (MA) columns are added and normalized.
- Consistent Data Enhancement:
    - Additional columns like high and low are derived from a data subset and normalized for consistency.
- Stock Movement Calculation:
    - Movement (result) is computed based on the difference between the closing prices of the current day and the next day.
- Binary Indicator Assignment:
    - If the stock movement is positive, the binary indicator 'up_dn' is set to one; otherwise, it is set to zero.
- Visual Representation:
    - A Candlestick chart of the stock price is visually presented in a 2x2 format, along with the moving average lines.

2. 2 Image Generation, Storage & Splitting:

- Image Generation:
    - A Candlestick chart, accompanied by a moving average, is created and saved as an image.
    - The generated image undergoes processing, involving cropping and the preservation of the cropped version.
- Data Storage:
    - Processed data including chart images, labels, investment returns, trade dates, and stock symbols, organized into arrays/lists.
    - Information from each iteration, such as location, result, and index of the cropped image, is systematically stored in a DataFrame named "results."
- Data Splitting:
    - The generated data is splitted into training, validation, and testing sets based on predefined cutoff dates.

2.3 Generate Pickle Files:

- The data is then stored in the form of arrays/lists into separate pickle files for later use.
- The saved data includes
    - x_train, x_valid, x_test
    - y_train, y_valid, y_test,
    - invest_return_test, invest_return_valid, invest_return_train,
    - trade_dates, ticker, results,
    - true_return_test, true_return_valid true_return_train.

3. CNN Architecture for Binary Classification:

3.1 Model Architecture

3.1.1 Input Layer

Shape: (107, 107, 1)

3.1.2 Convolutional Layers:

- Three Conv2D layers with neurons: 64, 32, 10, each incorporating the Rectified Linear Unit (ReLU) as activation function.

3.1.3 Pooling Layer:

- Conv2D layers block is succeeded by a MaxPooling2D layer with a pool size of (2, 2)

3.1.4 Regularization Layer:

- A Dropout layer with a 20% dropout rate is introduced after the pooling layer to prevent overfitting.

3.1.5 Flatten Layer

- A Flatten layer is succeeded by the Dropout layer to flatten the 2D vectors.

3.1.6 Dense Layers:

- Three Dense layers are integrated into the model.
    - The first two have 128 and 50 units, employing the ReLU activation function.
    - The final layer consists of two units with the softmax activation function, serving as the output layer.

3.2 Training Parameters:

- Epochs set to 3.
- Optimizer: Adam.
- Loss function: Categorical Cross-Entropy.

4. Original Results and Reproduced Results:

Both the results are almost similar but due to the scaling factor in plot axes a little variation can be observed in the graphs.

a. Percentage of winning Trades:

Both orange & blue lines show that stocks with a confidence score above 0.5 have a higher percentage of winning trades than stocks with a confidence score below that.

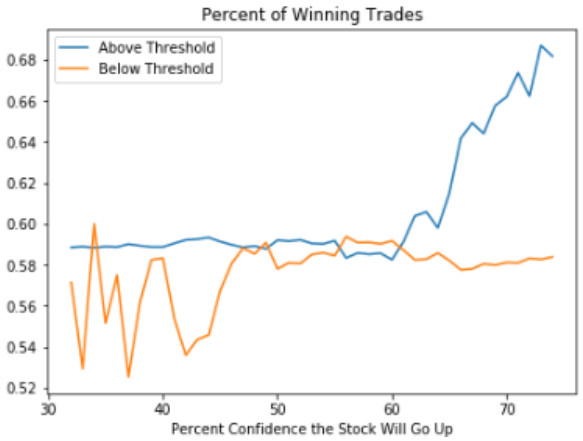


Fig. OriginalResults

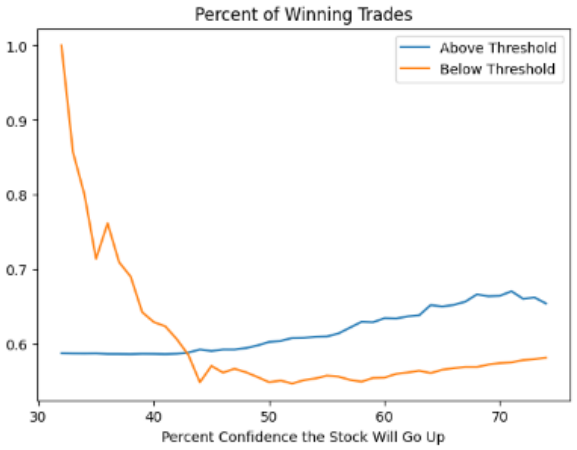


Fig. Reproduced Results

b. Average Return:

The blue and green lines show that as the confidence increases, so does the average return. The red and orange lines show the mirror that as confidence decreases, so does the average return with the exception of some outliers.

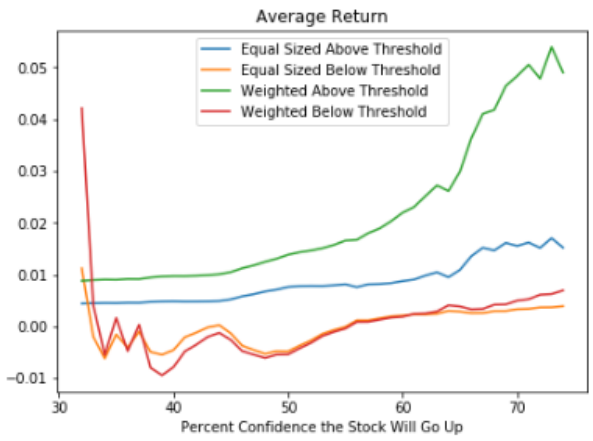


Fig. Original Results

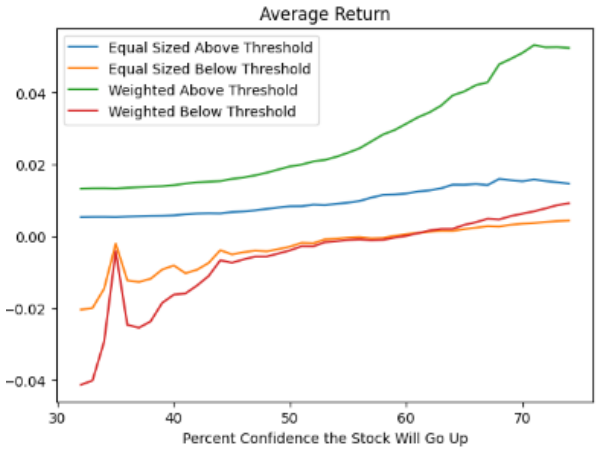


Fig. Reproduced Results

c. Sd of Returns:

The red and green lines show that as the trades become more heavily weighted, so too do the standard deviations of their returns. Interestingly the blue line decreases slightly above 65, which makes sense because as the percentage of winners increases, there will be fewer negative returns to expand the standard deviation.

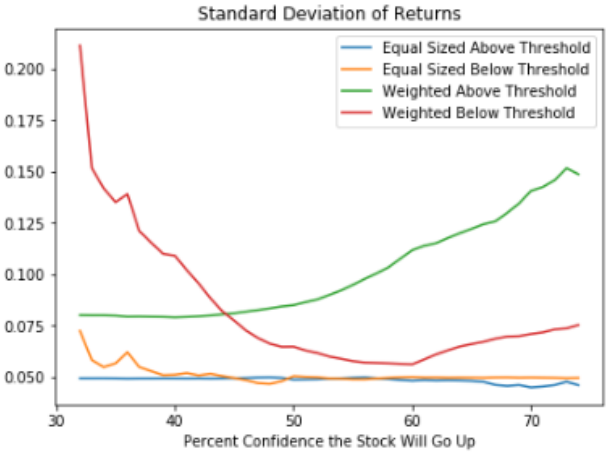


Fig. Original Results

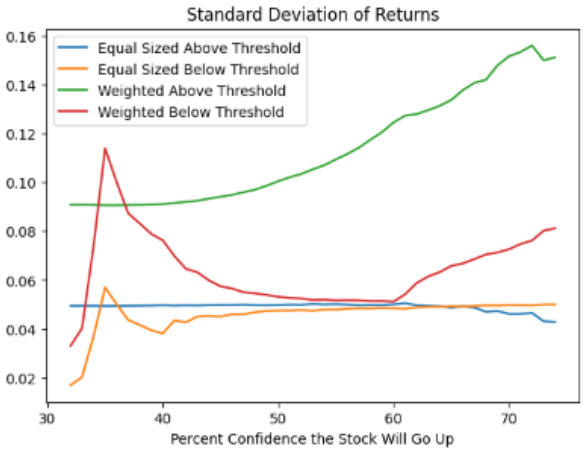


Fig. Reproduced Results

d. Risk Compare matrix:

The risk metric quantifies how much value the model is adding and we can clearly see that risk adjusted return increases as the model's confidence increases.

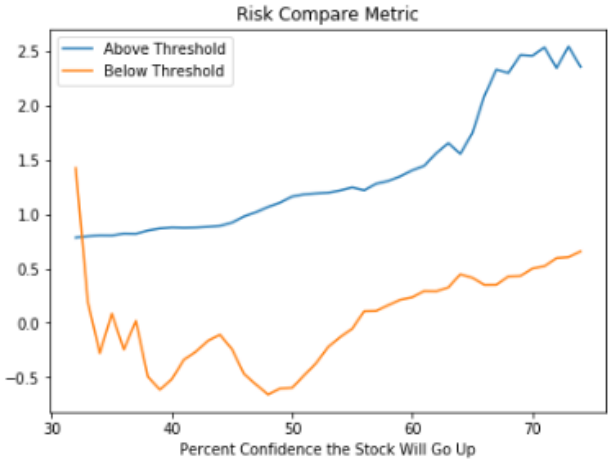


Fig. Original Results

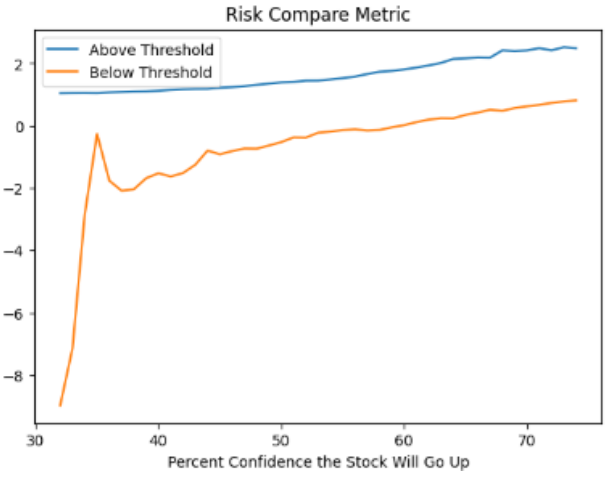


Fig. Reproduced Results

e. Weighted Average Return per Share:

This chart standardizes all of the weighted trades so that we are looking at returns on equal risk. This shows that the model identifies better trades and does not simply benefit from "betting more."

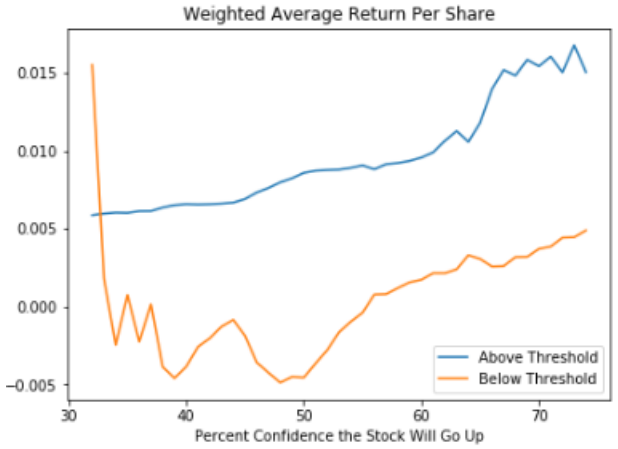


Fig. Original Results

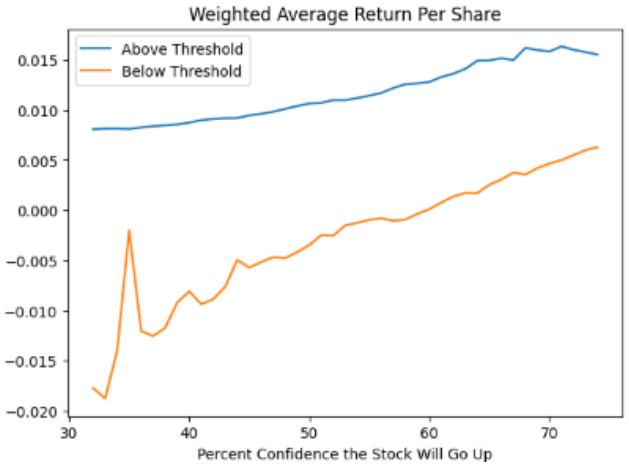


Fig. Reproduced Results

f. Percent of All trades taken:

The graph shows that the percentage of all trades taken decreases as the confidence level increases. This means that there are fewer trades with a higher confidence level.

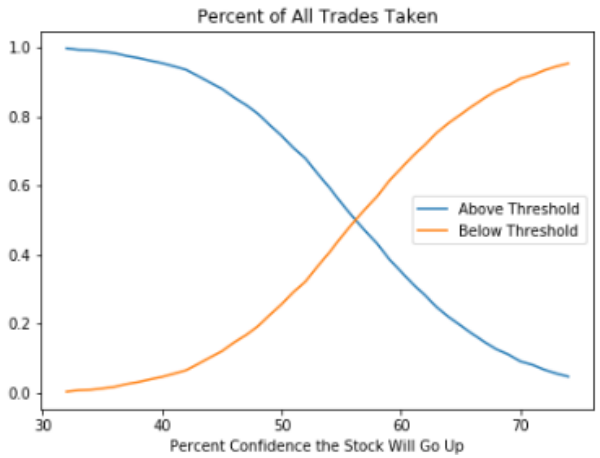


Fig. Original Results

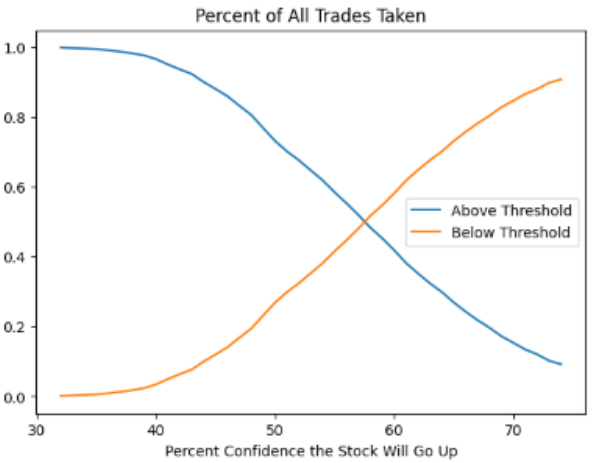


Fig. Reproduced Results

g. welch ‘s t-test Coefficient

This depicts the t-test coefficient for the isolated trades compared with all trades. Not surprisingly, the coefficient increases as the confidence goes up.

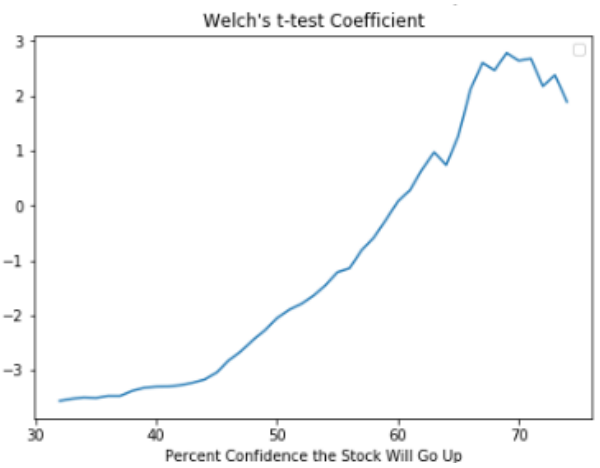


Fig. Original Results

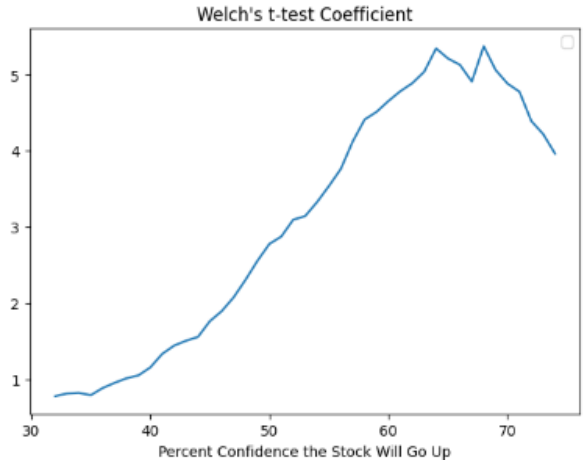


Fig. Reproduced Results

h. Welch’s t-test p-value:

This means that the stocks with a higher confidence score tend to perform better than stocks with a lower confidence score.

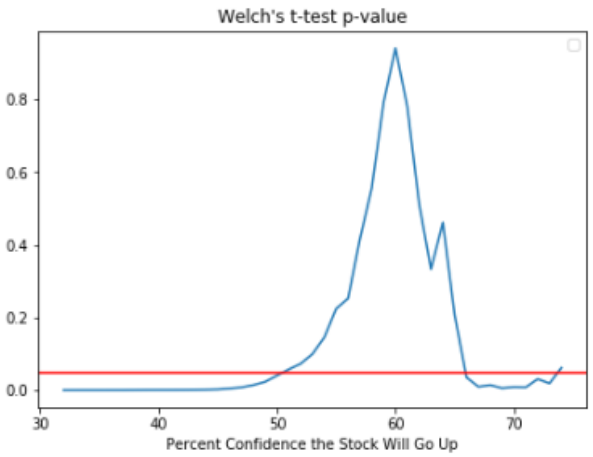


Fig. Original Results

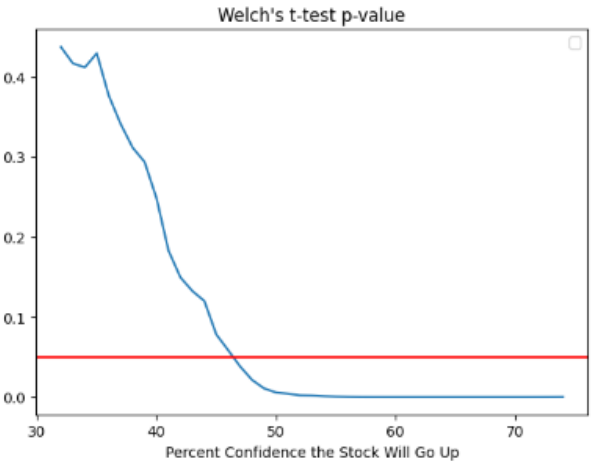


Fig. Reproduced Results

5. Results on Extended Data till DEC-2023:

a. Stock Data Range: The stock data under consideration spans from August 19, 2004, to December 11, 2023.

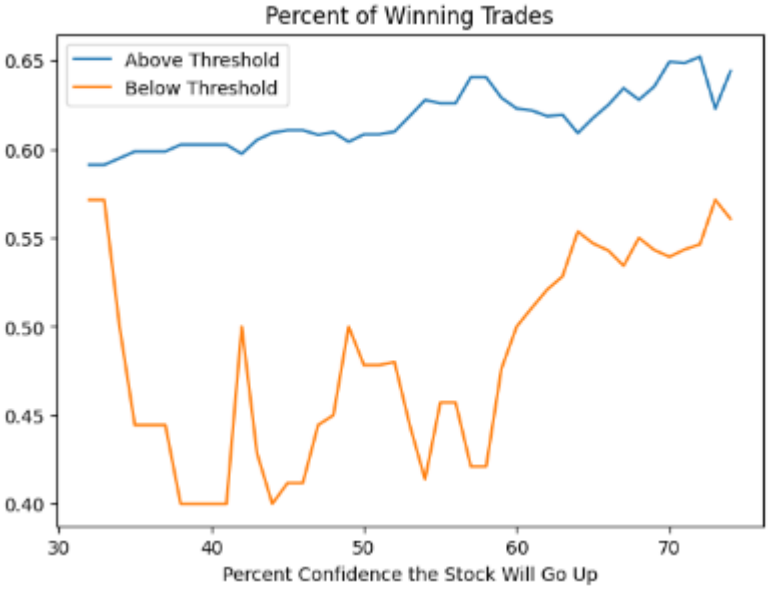
b. Train-Test-Validation Split: For the purpose of training and evaluating the model, a split of 70%-20%-10% is employed, where:

   i. Training Data: Covers the period from August 19, 2004, to February 29, 2018.

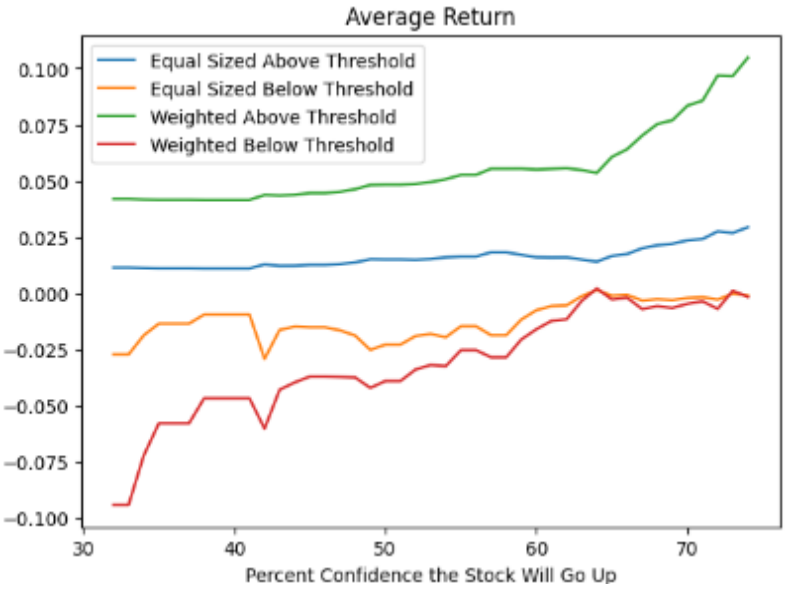
   ii. Validation Data: Encompasses the timeframe from March 1, 2018, to March 31, 2020.

   iii. Testing Data: Spans from April 1, 2020, to December 11, 2023.

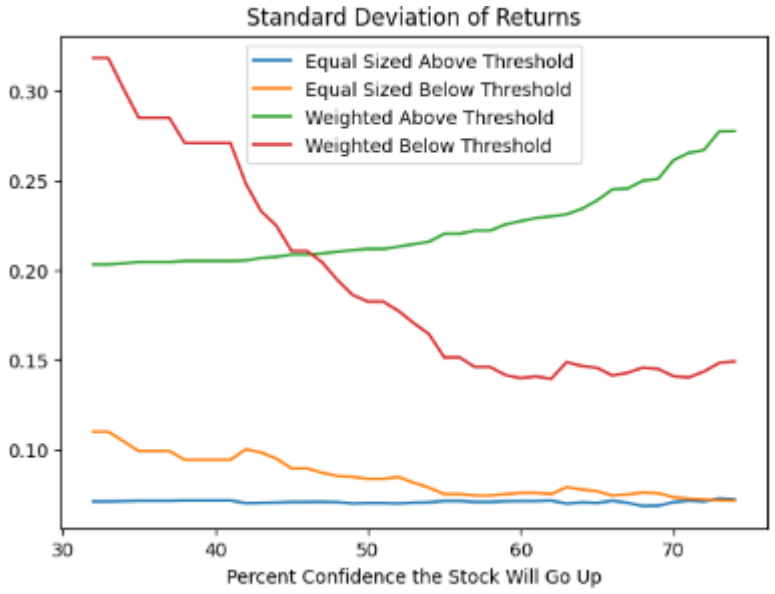
a. Percent of Winning Trades: This graph shows the percentage of winning trades achieved at different levels of confidence in the trading model. Both "Above Threshold" and "Below Threshold" lines are shown, suggesting that the model performs better above certain confidence thresholds.



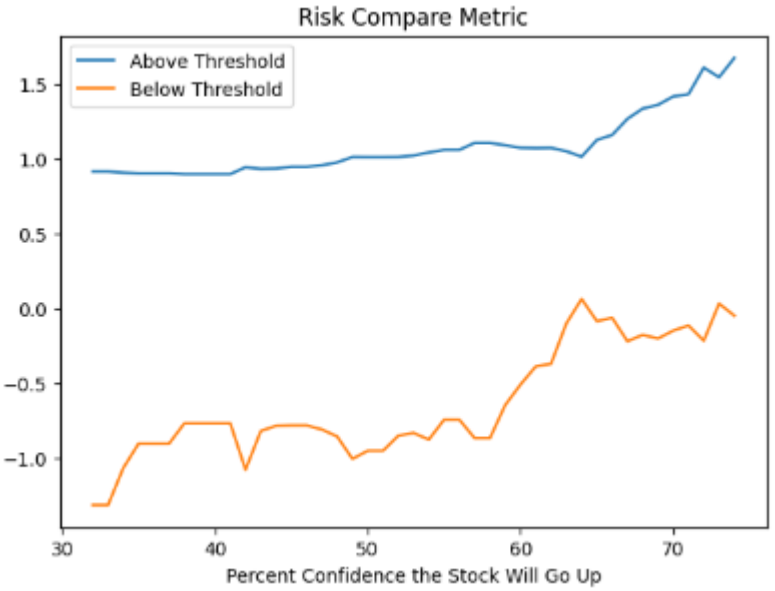
b. Average Return: This graph displays the average return on investment (ROI) for both "Equal Sized" and "Weighted" trades at different confidence levels. The "Equal Sized" lines represent trades with equal amounts invested regardless of confidence, while the "Weighted" lines prioritize trades with higher confidence. Interestingly, the "Weighted Above Threshold" line generally shows higher returns, suggesting that focusing on high-confidence trades can be beneficial.



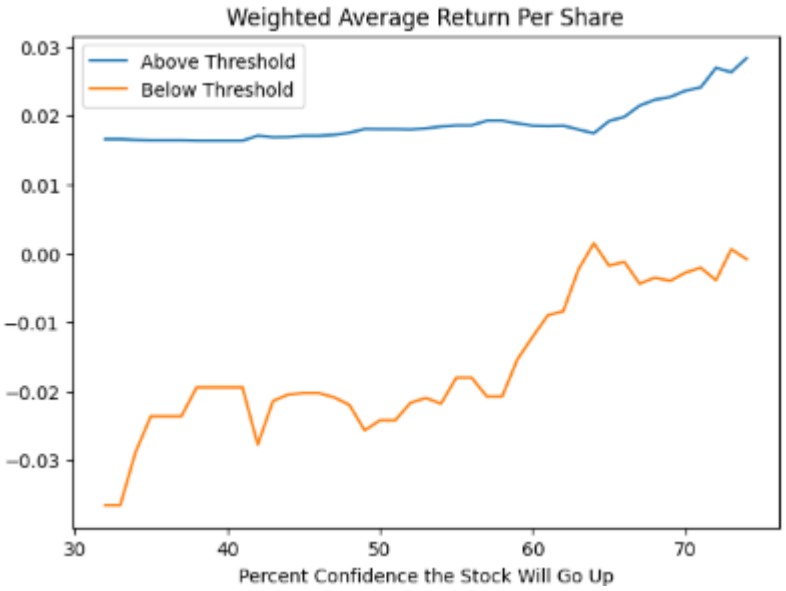
c. Standard Deviation: This graph depicts the standard deviation of returns for both "Equal Sized" and "Weighted" trades at different confidence levels. Higher standard deviation indicates greater variability in returns, and the graph seems to show that "Below Threshold" trades have higher variability, especially for "Equal Sized" trades.



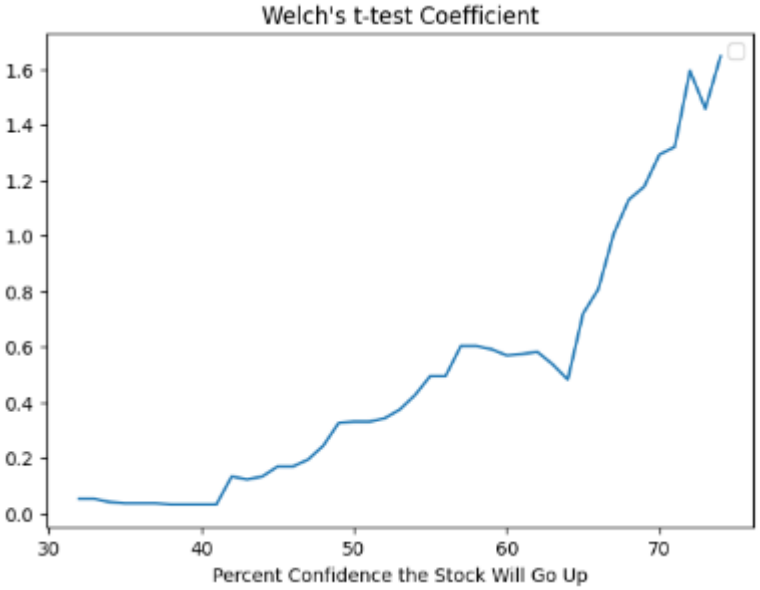
d. Risk Compare Metric: This graph is currently a placeholder and requires your specific data and metric definition to be interpreted. Please provide details about the risk comparison metric you'd like to analyze.



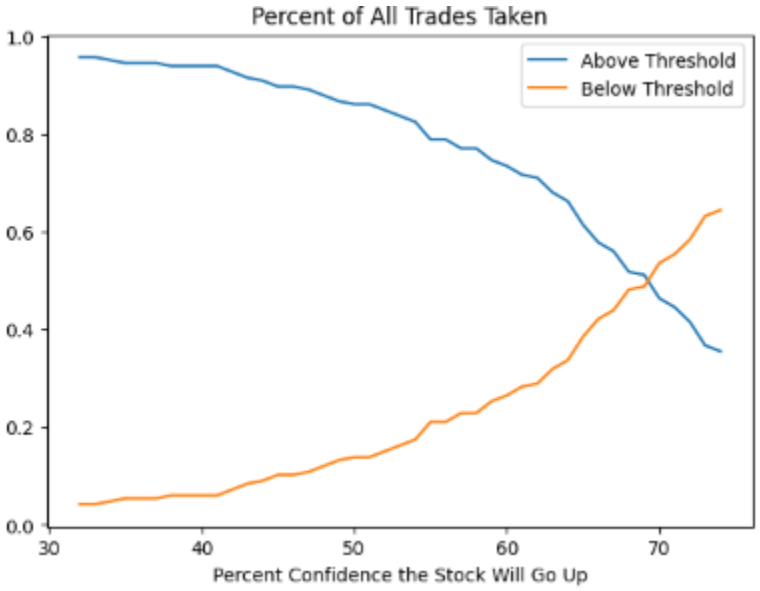
e. Weighted Average Return per Share: Similar to the previous placeholder, this graph requires your actual data and definition of the "Weighted Average Return per Share" metric for interpretation.



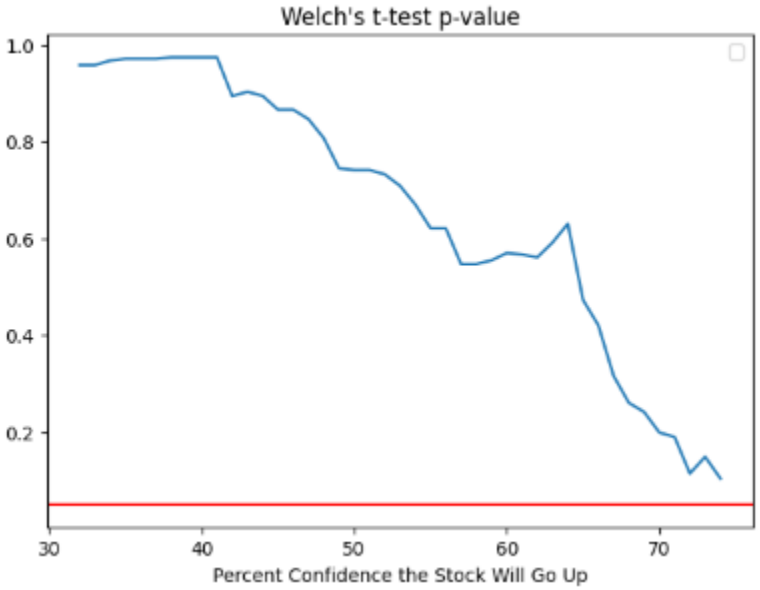
f. Percent of All Trades Taken: This graph shows the percentage of all trades taken at different confidence levels. It seems that the model takes more trades as confidence increases, potentially indicating a more aggressive trading strategy at higher confidence levels.



g. Welch's t-test Coefficient: This graph displays the Welch's t-test coefficient, a statistical measure of the difference in means between the "Above Threshold" and "Below Threshold" groups for each confidence level. Values closer to 1 indicate a stronger difference in means, suggesting that the model's performance significantly differs between the two groups at those confidence levels.



h. Welch's t-test p-value: This graph shows the p-value associated with the Welch's t-test. Lower p-values (typically below 0.05) indicate statistically significant differences between the groups, confirming the observations from the previous graph.



6. Model Architecture:

a. The architecture of the improved model is as follow:

Layer	Type	Units	Activation	Kernel Size	Pooling Size	Dropout
Conv2D	Convolutional	128	relu	(3, 3)		0.3
BatchNormalization	Normalization	-	-	-	-	-
Conv2D	Convolutional	64	relu	(3, 3)		-
Conv2D	Convolutional	32	relu	(2, 2)		0.3
MaxPooling2D	Pooling	-	-	(2, 2)		-
Flatten	Reshaping	-	-	-	-	-
Dense	Fully-connected	256	relu	-	-	0.3
Dense	Fully-connected	128	relu	-	-	0.3
Dense	Fully-connected	64	relu	-	-	-
Dense	Output	num_classes	softmax	-	-	-

b. Hyper-parameters:

Key	Value
Optimizer	Adam
Learning Rate	0.001
Batch Size	batch_size
Epochs	epochs
Loss Function	categorical_crossentropy
Metrics	Accuracy
Validation Data	Yes (x_valid_mod, y_valid_mod)