Predicting Profit Opportunities in S&P 500 ETF (SPY)

Finance and Structures for Data Science Final Project

Sofia De la Mora Tostado
Anna Joen
Sushant Prabhu

In this project, the goal was to predict profit opportunities in the S&P 500 ETF (SPY) by estimating the probability of achieving a 1% profit. To achieve this, we have built multiple models using neural networks - LSTM and CNN.

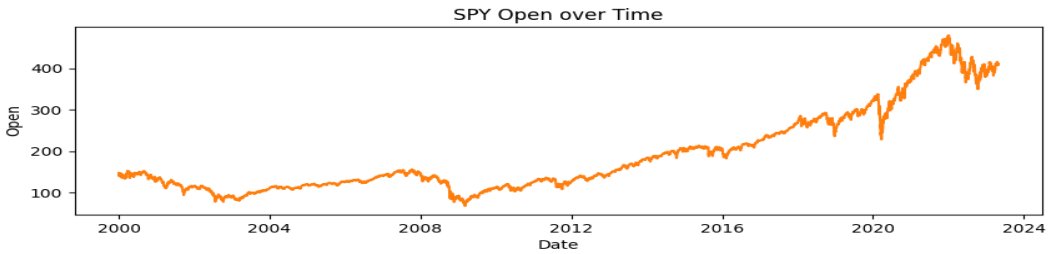
LSTM was chosen as it is well-suited for processing sequential data, and can capture temporal dependencies in the data. Different time window sizes were explored to evaluate the model's performance. A time window refers to the number of previous days' data that is used as input to the model. The time window sizes explored were 3, 50, 100, and 1000 days. The results showed that a time window of 50 days yielded the highest accuracy while the window size of 3 had a comparatively better F1 score. To evaluate the model's performance, we used various metrics such as accuracy, precision, recall, and F1-score.

Accuracy measures the overall correctness of the model's predictions, precision measures the proportion of true positive predictions among all positive predictions, recall measures the proportion of true positive predictions among all actual positives, and F1-score is a weighted average of precision and recall. Overall, our results suggest that it is possible to predict profit opportunities in the SPY 500 ETF using neural networks. However, it was important to carefully select the time window size and other hyperparameters of the model to achieve the best performance.

Introduction

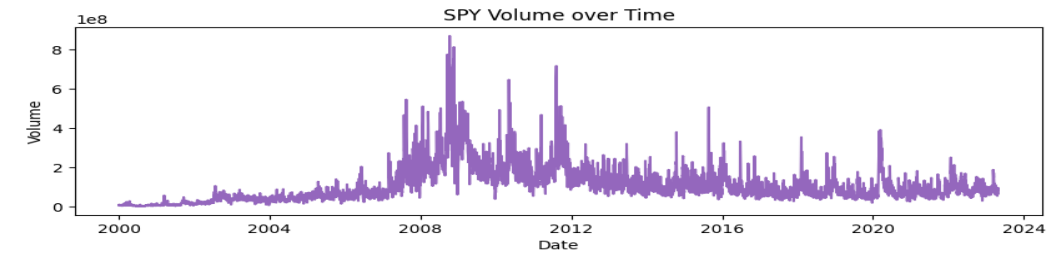
The objective of this project is to estimate the probability that the high price of the SPY ETF will be greater than or equal to 1.01 times the opening price. A trading strategy is to be developed that enters a trade if this probability is greater than 60%. Our input variables include the opening, closing, high, and low prices, as well as the volume of the previous n days. We used a neural network to solve this classification problem and implemented the softmax activation function to estimate the probability of making a profit of 1%.


SPY Close over Time
Close
400
300
200
100
2000
2004
2008
2012
2016
2020
2024
Date


SPY Open over Time
Open
400
300
200
100
2000
2004
2008
2012
2016
2020
2024
Date


SPY High over Time
High
500
400
300
200
100
2000
2004
2008
2012
2016
2020
2024
Date


SPY Low over Time
Low
400
300
200
100
2000
2004
2008
2012
2016
2020
2024
Date


SPY Volume over Time
1e8
Volume
8
6
4
2
0
2000
2004
2008
2012
2016
2020
2024
Date

Methodology

Approach 1: **LSTM Model NN Architecture**

We used historical data for the SPY ETF from Yahoo Finance, from January 2000 to March 2023. The data was preprocessed, and the target variable was created, indicating whether the high price was at least 1.01 times the opening price. The SMOTE technique was employed to balance the dataset. We used K-Fold cross-validation with time-series splits and experimented with window sizes of 3, 50, 100, and 1000. The model was built using LSTM layers with tanh activation, dropout layers, and a dense layer with softmax activation for the output. Adam optimizer was selected with a learning rate of 0.001, and binary cross-entropy was used as the loss function. Early stopping and learning rate reduction were incorporated.

Approach 2: **CNN Architecture**

For predicting whether the high price of the S&P 500 index ETF (SPY) will increase by at least 1% on the next trading day, based on historical data of the opening price, closing price, high price, low price, trading volume, and volatility, we train the data after preprocessing and splitting into training and testing sets, standardize and reshaped to a 3D format for input to a 1D convolutional neural network (CNN). The CNN model is trained with the Adam optimizer and evaluated on the testing set for 20% and then future forecasted. Ingesting a 2D image and using convolutional layers to extract patterns and features, this approach allows the model to capture spatial and temporal relationships between the different inputs.

Results

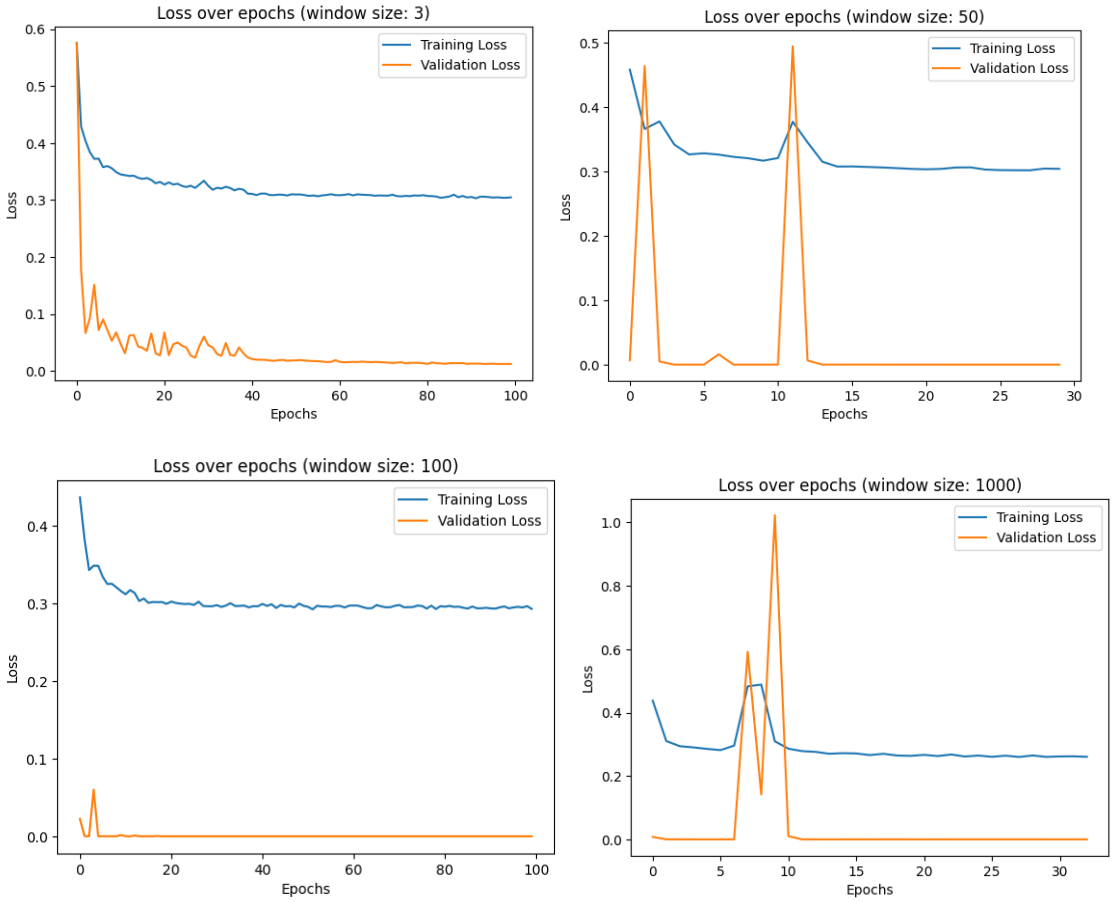
Approach 1 :

The model was evaluated using accuracy, precision, recall, and F1 score. The results are as follows:

Window size	Accuracy	Precision	Recall	F1
3	0.841244	0.514894	0.473559	0.491930
50	0.847220	0.569231	0.403202	0.406286

100	0.841650	0.600000	0.399797	0.401207
1000	0.782729	0.400000	0.400000	0.400000

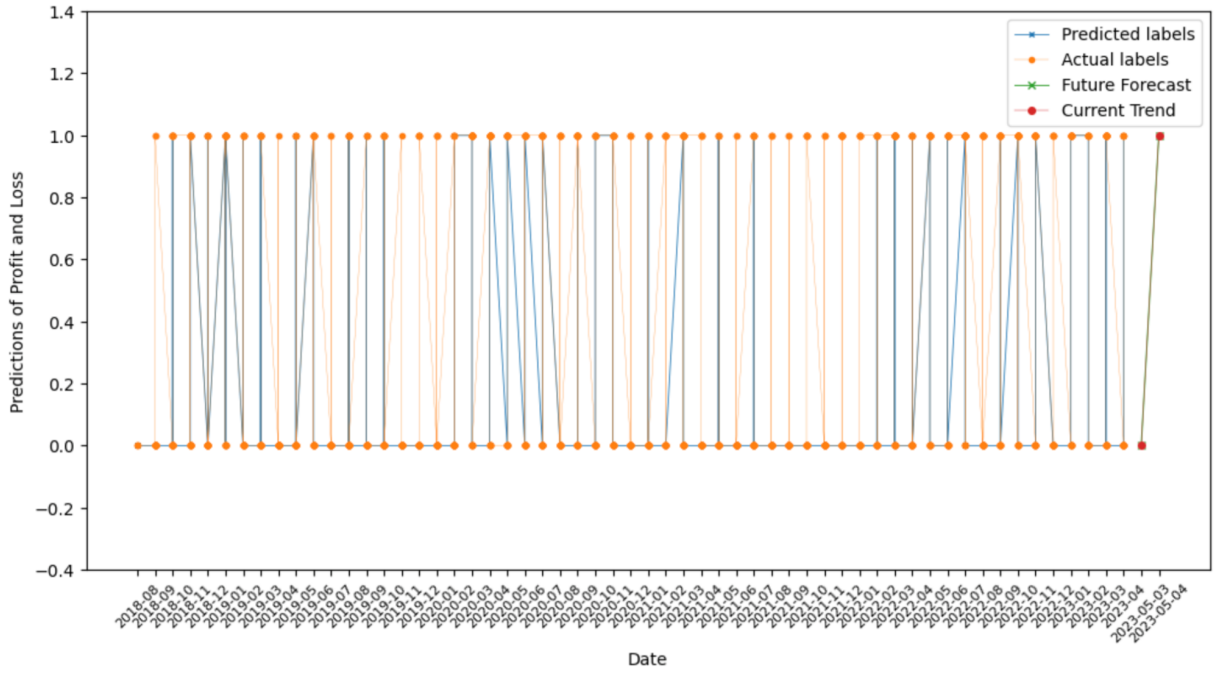
The model achieved the highest accuracy and F1 score with a window size of 3. The training and validation loss plots for different window sizes show that the model's performed the best with window size 3.



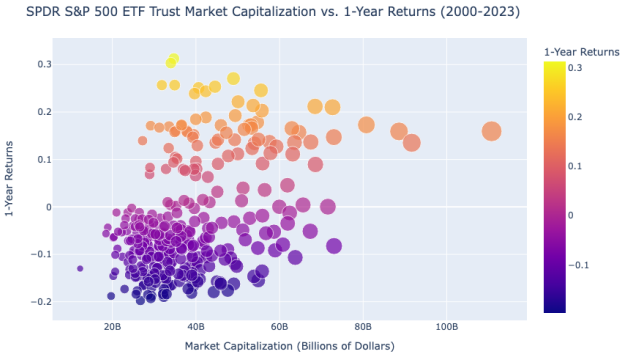
Approach 2 :

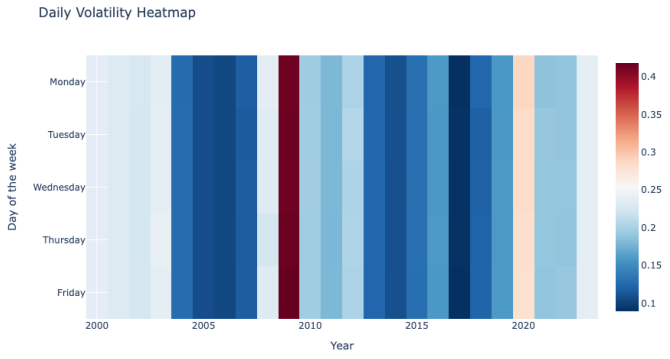
We identified n = 3 window approach works efficiently so leveraged that for CNN model architecture and the results looked like -
Test accuracy: 74%

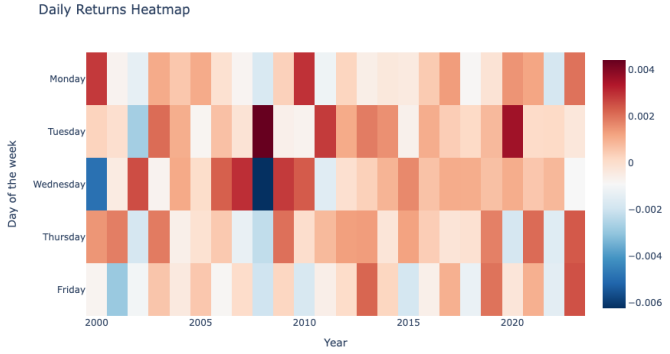
Epochs	Accuracy	Precision	Recall	F1
100	0.738	0.73	0.718	0.677



Exploratory Data Analysis (Interactive on Jupyter Notebook)







Conclusion

In conclusion, this project showcased the potential of utilizing neural network model architectures, specifically LSTM and CNN, to predict profit opportunities in the S&P 500 ETF. By employing different window sizes and carefully selecting the appropriate time window size and other hyperparameters, our model was able to estimate the probability of making a profit of 1% based on historical data. Additionally, we explored the possibility of using state-of-the-art models like Prophet for our use case to develop an alternative trading strategy. This project serves as a solid foundation for further research and development in the field.

Future Steps

Moving forward, we plan to integrate this model on the Alpaca trading platform to test out the performance and keep CI/CD the model performance for future S&P movements. We might experiment with different feature engineering techniques and investigate the impact of using alternative data sources, such as macroeconomic indicators, market sentiment, or news articles, to enhance our model's predictive capabilities.