# Problem1

## Suppose

1. Directly trade the SPX index.
2. Treat this problem as a dynamic programming algorithm, with the ultimate goal of maximising the sum of each `sale price - buy price` (considering transaction fees or not).

## Result

Run Question1.py, you can get the following result

| Scenario | Profit |
|---|---|
| Scenario1: Max profit with fee | 17622.68 |
| Scenario2: Max profit with transaction limits | 5998.48 |

# Problem2

## Data Collection

Choose the following data to help predict return.

| Factor name | Category | Source | Way to download | Frequency | Start from |
|---|---|---|---|---|---|
| VIX | Market | Yahoo finance | yfinance package | daily | 1990-01-02 |
| SPX yesterday's close price | Market | Yahoo finance | yfinance package | daily | 1928-01-03 |
| SPX volume | Market | Yahoo finance | yfinance package | daily | 1950-01-03 |
| 2-Year Treasury Yield | Market | Yahoo finance | yfinance package | daily | 1960-01-04 |
| 10-Year Treasury Yield | Market | Yahoo finance | yfinance package | daily | 1962-01-02 |
| Equity Risk Premium | Market | Yahoo finance | calculate | daily | 1962-01-02 |

| | | | | | |
|---|---|---|---|---|---|
| GDP | Macro | BEA | Download online | quarterly | 1929A |
| PCE | Macro | BEA | Download online | quarterly | 1929A |
| Non-Farm Payrolls | Macro | FRED | Fred api | monthy | 1939-01 |
| CPI | Macro | FRED | Fred api | monthy | 1947-01 |
| MA50 | Technical | calculate | calculate | daily | 1928-01-03 |
| RSI | Technical | calculate | calculate | daily | 1928-01-03 |
| MACD | Technical | calculate | calculate | daily | 1928-01-03 |

## Model

Due to time constraints, I will temporarily establish the following assumptions:

1. Assume the strategy involves periodic buying and selling, with the goal shifting to finding factors that indicate when to buy.
2. Based on the first question of buying and selling 100% of the SPX, assume direct trading of the index without searching for additional products.
3. Due to the strategy allowing a maximum of 20 trades per year, my `ylabel` is labeled as the 6-12 day return of the SPX.

### Model1: LSTM

(Code on Colab due to the usage of GPU: https://colab.research.google.com/drive/19hCnlJ_36DwAND0XyBlvwev42W1NX6Nb?usp=sharing)
Data Explanation:

1. The selected factor pool consists of the 13 factors mentioned in the Data Collection.
2. Based on their overlapping valid time periods, the training set is tentatively set from 1990 to 2017, the validation set from 2018 to 2019, and the test set from 2019 to 2023.

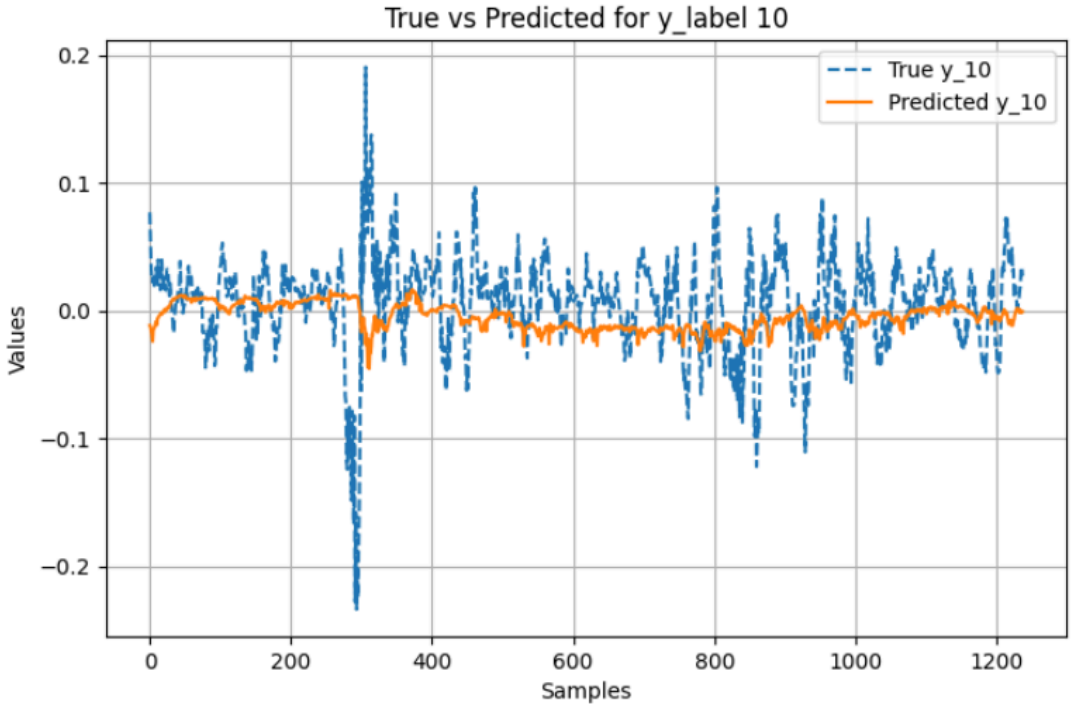
Order of Operations:

1. Standardize the independent variables.
2. Select time steps to construct the sequential X and y.
3. Set the parameter pool, and select the model through machine tuning by observing the validation loss.
4. Test the model performance on the test dataset.

Parameter tuning analysis:

1. batch_sizes = [16,32,64]
2. layers = [1,2,3]
3. time_steps_values = [10,20,30,40,60]

I attempted periodic holding durations of 4, 6, 8, 10, and 12 business days, so the predicted values were set as the 4, 6, 8, 10, and 12-day return rates. For each period, the optimal model was selected through machine tuning, and based on the validation dataset loss, the holding period for model1 was chosen to be 10 business days. The chart drawn on the test set is as follows:




## Model2: LR

Data Explaination:

1. The selected factor pool consists of the 13 factors mentioned in the Data Collection.
2. Based on their overlapping valid time periods, the training set is tentatively set from 1990 to 2017, the validation set from 2018 to 2019, and the test set from 2019 to 2023.

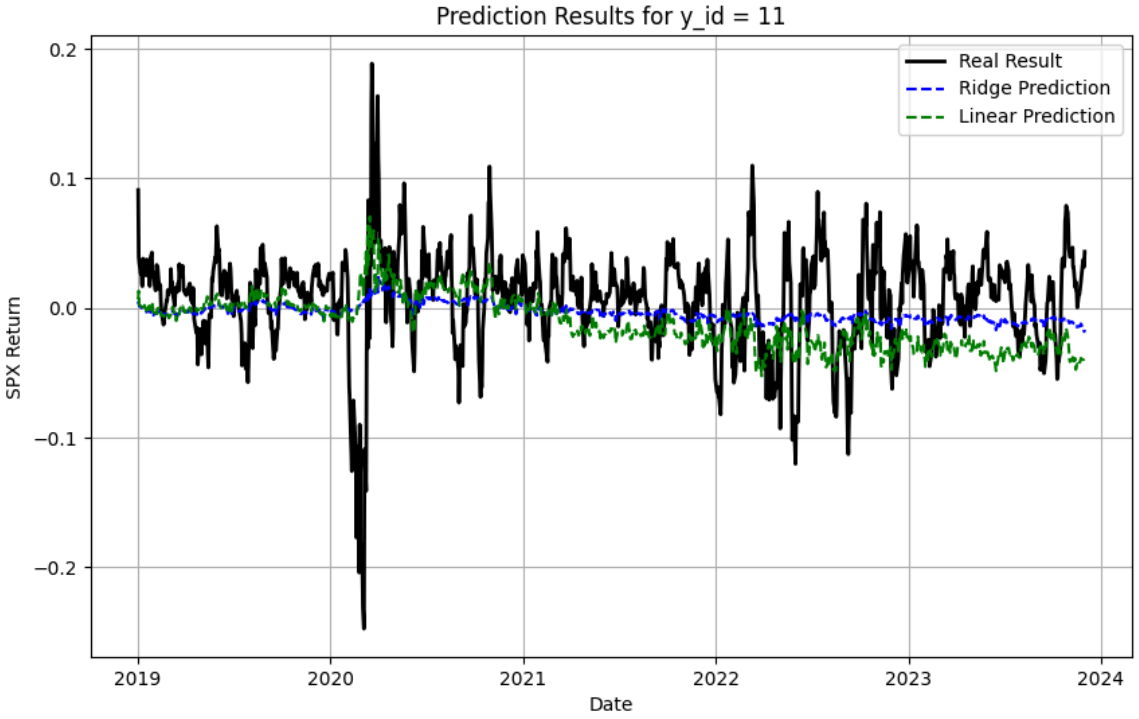
Order of Operations:

1. Standardize the independent variables.
2. Establish two linear regression models.
3. Train the data on the training set, observe the training effect on the validation dataset, and then train on both the training set and validation dataset.
4. Predict results on the test set.

Parameter tuning analysis:

1. Cycle：Holding periods of 4, 6, 8, 10, and 12 business days.
2. Model：Linear Regression，Ridge Regression

Analyze the error (MSE, $R^2$) on the validation set for each holding period and model. The final selected model is a linear regression model with a holding period of 4 business days. The model is then trained on the entire training dataset and validation dataset, and the prediction chart on the test dataset is shown below:




## Portfolio

Based on the distribution of n-day return rates in the test set, select the threshold at which a buy operation is executed when the predicted value exceeds this threshold, and sell after n days.

For the percentage selection of the threshold distribution, use the formula `(252/n - 18) / (252/n)` to control the average number of operations per year to not exceed 20.

The final results of the two models are shown in the table below:

| | Return | Win rate |
|---|---|---|
| Model1_LSTM | 25% | 70% |
| Model2_LR | 4% | 67% |

## Result Analyse

1. First, I chose these two models to compare the predictive results of linear and non-linear models.
2. Based on the charts and real trading returns, although the win rate is decent, the return results are quite average.

**Model 1 (LSTM):** LSTM is generally an ideal model for predicting complex time series. However, due to time and GPU cost constraints, I tested a limited set of parameters. From the chart, the model appears overly smoothed and does not capture significant fluctuations, with a noticeable lag. Next steps for optimization: Increase the number of layers in training; Use a more complex model structure (e.g., Bidirectional LSTM).

**Model 2 (Linear Regression):** Linear regression can only capture linear factors, so its capabilities are relatively limited. However, its short-term performance exceeded my expectations, and it was able to capture large fluctuations. However, its long-term prediction performance was quite average, which is why the 4-day holding period outperformed the 10-day prediction significantly. Optimization direction: Update the model iteratively over time. For example, adding the previous month's data to the train_dataset every month to update the model.

3. Overall, I believe that with further tuning, LSTM will certainly outperform Linear Regression, primarily due to its superior capability in handling time series.

### Other Models

If I had enough time, I would also try the following models for this task:

1. Scraping financial news from mainstream media, using a large language model to derive daily sentiment factors, and then converting these factors into buy/sell signals for trading.
2. Prediction using the XGBoost model.
3. Time series model predictions.