CQF Exam Three

June 2022 Cohort

Zhizhao Wang

This is the main report for Exam 3.

First part (A. Maths) consists of the answers for the math questions of the exam.

Second part (B. Models) includes the whole process of building the model for short-term asset move.

A. Maths

A.1

- (a) Yes. MSE consists of two parts, which are variance and bias. Least Square estimator finds the set of coefficients which minimise the MSE with no bias, but there exist biased estimators (such as Ridge estimator and Lasso estimator) which could have smaller MSE by trading a little bias for a larger reduction in variance. This can be achieved by shrinking or setting to zero some of the least squared coefficients.

- (b) Yes, MSE measures the model error in the ideal situation, model error plus irreducible error in practice. Model error can be decreased by adjusting the coefficients of the model. Irreducible error comes from the noise of the data itself and cannot be reduced by any model. According to the definition of MSE, it is the mean of the squared errors between the estimated values from the model and the true values. MSE can be decreased by adjusting the coefficients of the model, so it measures the model error. However, in practice the data values measured from the real world contain noise and these noisy values are regarded as the "true value" to train the model; in this case, MSE actually contains both model error and irreducible error.

A.2

The answer is **(a) $k(z_1,\, x)$ will be close to 1 and $k(z_2,\, x)$ will be close to 0**.

Gaussian RBF kernel has the form of:

$$k(x_i,\, x_j) = exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$

and it can be rewritten as:

$$RBF(d) = k(x_i,\, x_j) = exp\left(-\gamma d^2\right)$$

where $d = ||x_i - x_j||$, which is the geometrical distance between x_i and x_j; $\gamma = \frac{1}{2\sigma^2}$ is a positive constant.

As d is the distance, it is a non-negative value. So $RBF(d)$ is a monotonically decreasing function in its domain, and has range of $(0,\, 1]$. Because z_1 is geometrically very close to x, their distance is close to 0, the kernal value:

$$k(z_1,\, x) = \lim_{d \to 0} RBF(d) = 1$$

z_2 is geometrically far away from x, their distance is close to infinity, so:

$$k(z_2,\, x) = \lim_{d \to \infty} RBF(d) = 0$$

A.3

Considering the decision tree for classification problem: K is the number of classes, k indicates the k^{th} class; M is the number of leaves/braches at a certain depth of the tree, m indicates the m^{th} leaf/branch; $p_{m,\,k}$ is the portion between the number of observations of class k in the m^{th} leaf/branch and the number of observations in the m^{th} leaf/branch:

$$p_{m,\,k} = \frac{N_{m,\,k}}{N_m}$$

- (1) Entropy

  Entropy is used in some kinds of decision trees (such as ID3, C4.5) to represent the information. ID3 decision tree uses information gain as its loss function, which is the change of entropy after one more split using a certain feature. The selected feature for the split is the feature which makes the information gain highest. C4.5 tree is similar to ID3, the difference is that it uses the ratio of information gain as the loss function. The entropy of the m^{th} leaf/branch is given as:

$$Entropy(m) = -\sum_{k=1}^{K} p_{m,\,k} log_2(p_{m,\,k})$$

- (2) Gini Impurity Index

Gini impurity index measures the impurity level of the model. If the samples in a leaf/branch are of the same class, the impurity is 0. It is used in CART as the loss function. The Gini of the m^{th} leaf/branch is given as:

$$Gini(m) = \sum_{k=1}^{K} p_{m,\,k}(1 - p_{m,\,k})$$

- (3) Misclassification Error

  Misclassification error measures the possibility of that a sample is being misclassified by the model. The lack of differentiability means it is not suitable for more complex set up. The misclassification error of the m^{th} leaf/branch has the form of:

  $$Error(m) = 1 - \underset{k}{max}\,(p_{m,\,k})$$

  where the function $\underset{k}{max}\,(p_{m,\,k})$ gives the maximal $p_{m,\,k}$ for all possible k.

B. Models

Short-term asset return is a challenging quantity to predict. Efficient markets produce near-Normal daily returns with no significant correlation between r_t, r_{t-1}. The B part of the project is to produce a model to predict **one-day move trend** for gold ETF (GLD) using Support Vector Machine.

The project will follow the process below:

- Step 1: Understand and Collect the data
- Step 2: Data Exploration
  - Daily Return (1D_Return)
  - Model Target (Target)
- Step 3: Feature Engineering
  - Feature Generation
  - Feature Selection
- Step 4: Clean and Transform data
- Step 5: Modelling and Hyperparams Tuning
  - General View of SVC and SVM
  - Hyperparameter Tuning
  - Final Model
- Step 6: Validation and Evaluation
  - Accuracy
  - Confusion Matrix
  - Classification Report
  - Area under ROC Curve
- Step 7: Backtesting and Trading Strategy
  - Without Short Position
  - With Short Position

Numerical results and plots will be included in this report with explanations; the corresponding codes can be found in the auxiliary file **WANG_CODE.ipynb**. The final model gives the accuracy of **0.5291** and AUC-ROC score of **0.525** on the test set.

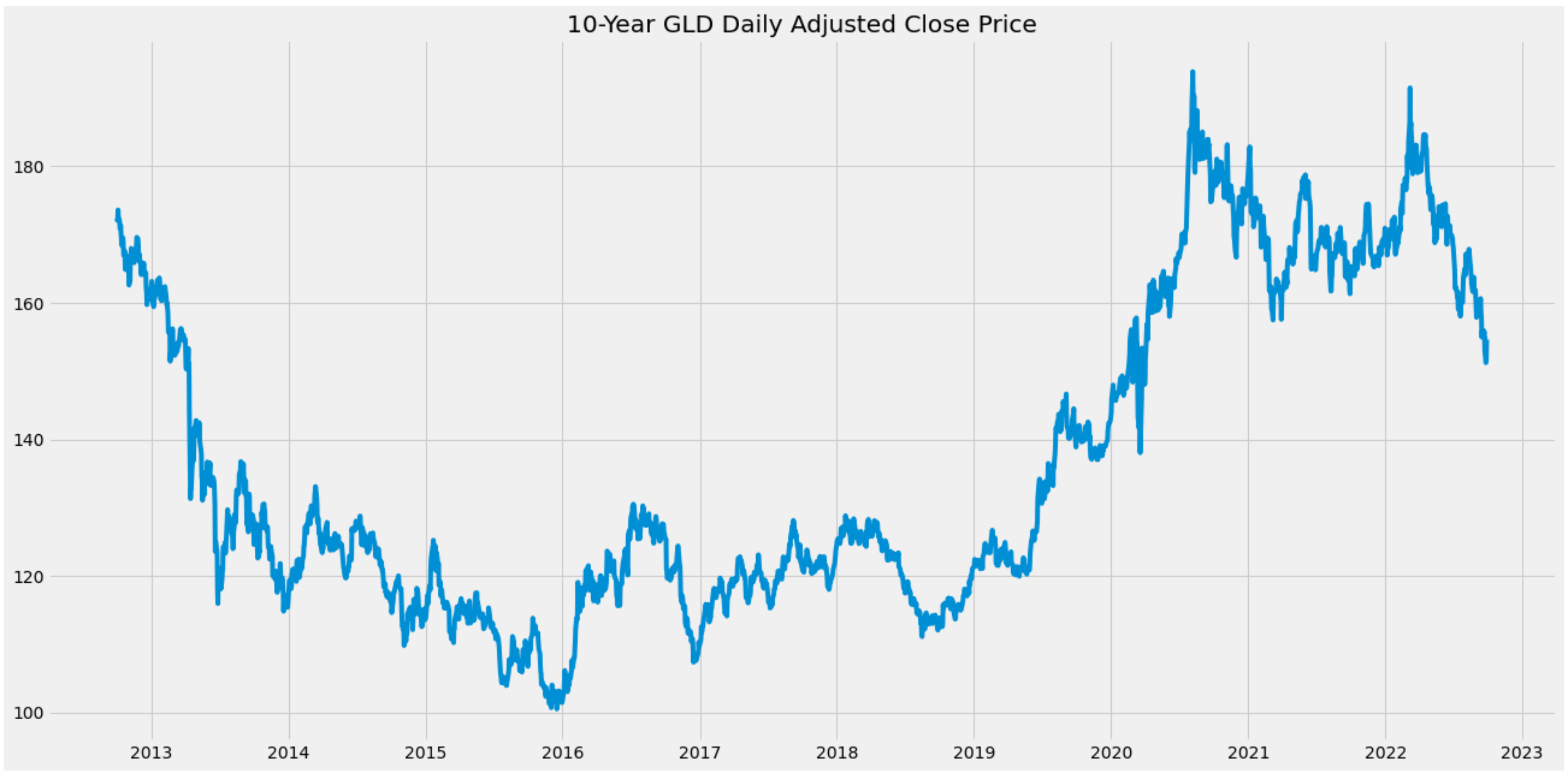
B.1 Understand and Collect the data

As an asset class, gold is unique. The economic forces that determine the price of gold are different from the economic forces that determine the price of many other asset classes such as equities, bonds or real estate. Gold offers investors an attractive opportunity to diversify their portfolios.

SPDR® Gold Shares (NYSEArca: GLD) offer investors an innovative, relatively cost efficient and secure way to access the gold market. Originally listed on the New York Stock Exchange in November of 2004, and traded on NYSE Arca since December 13, 2007, SPDR® Gold Shares is the largest physically backed gold exchange traded fund (ETF) in the world. The net asset value (NAV) of GLD is determined using LBMA Gold Price PM, so GLD price is closely related to the spot price of gold.

In this project, the data used is the 10 years GLD daily prices from 2012-10-01 to 2022-10-01. The data is collected and downloaded through yahoo finance API and saved in file **GLD.csv** in the folder.

The original data consists of 2517 daily data with 6 features: "Open", "High", "Low", "Close", "Adj Close" and "Volume". There is no missing values in any feature column. Below is the plot for 10-year GLD daily adjusted close price:



B.2 Data Exploration

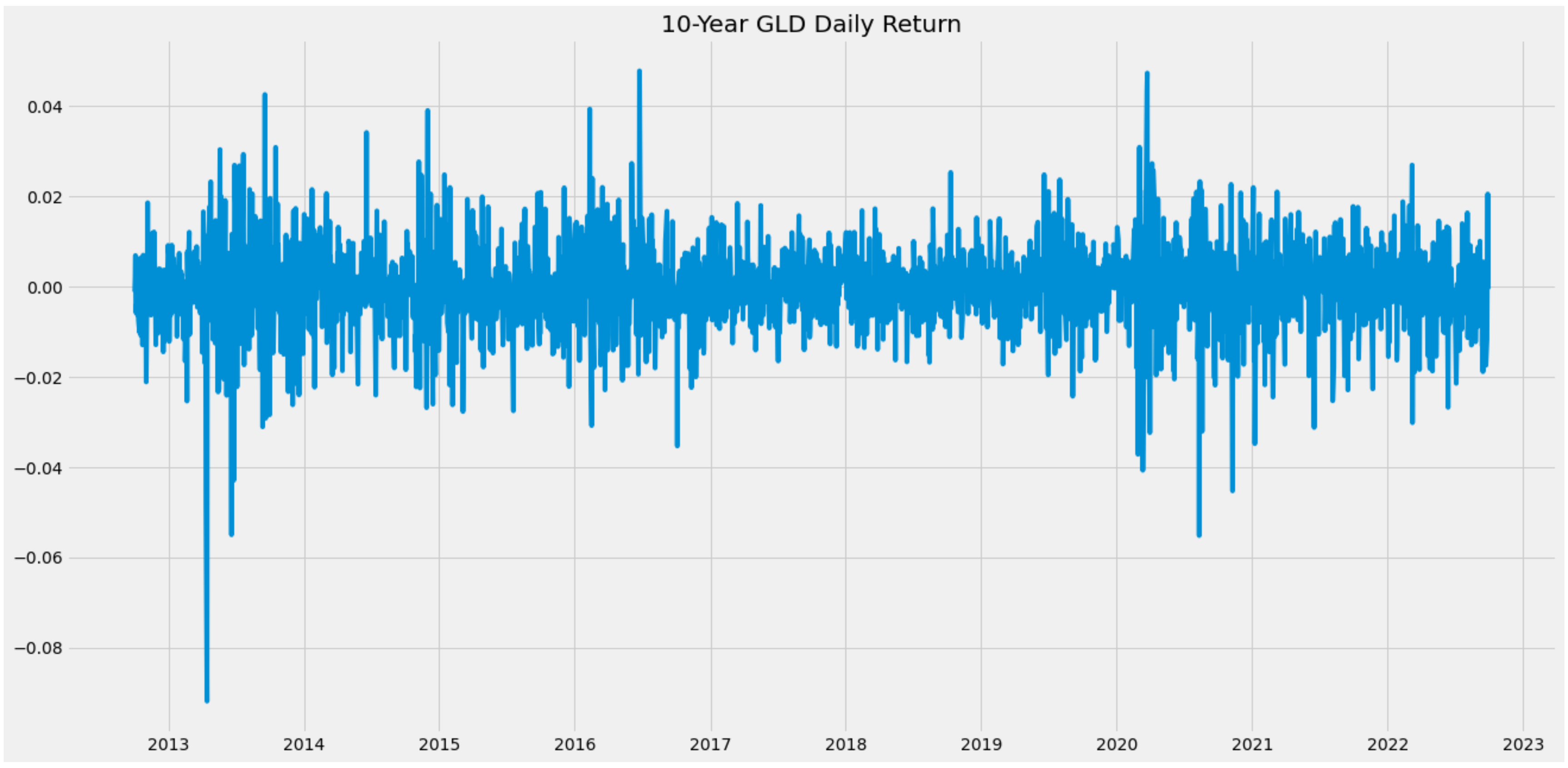
- **B.2.1 Daily Return (1D_Return)**

First, daily return of the GLD is calculated by the difference between the log of today's adjusted close price and the log of the previous day's adjusted close price:

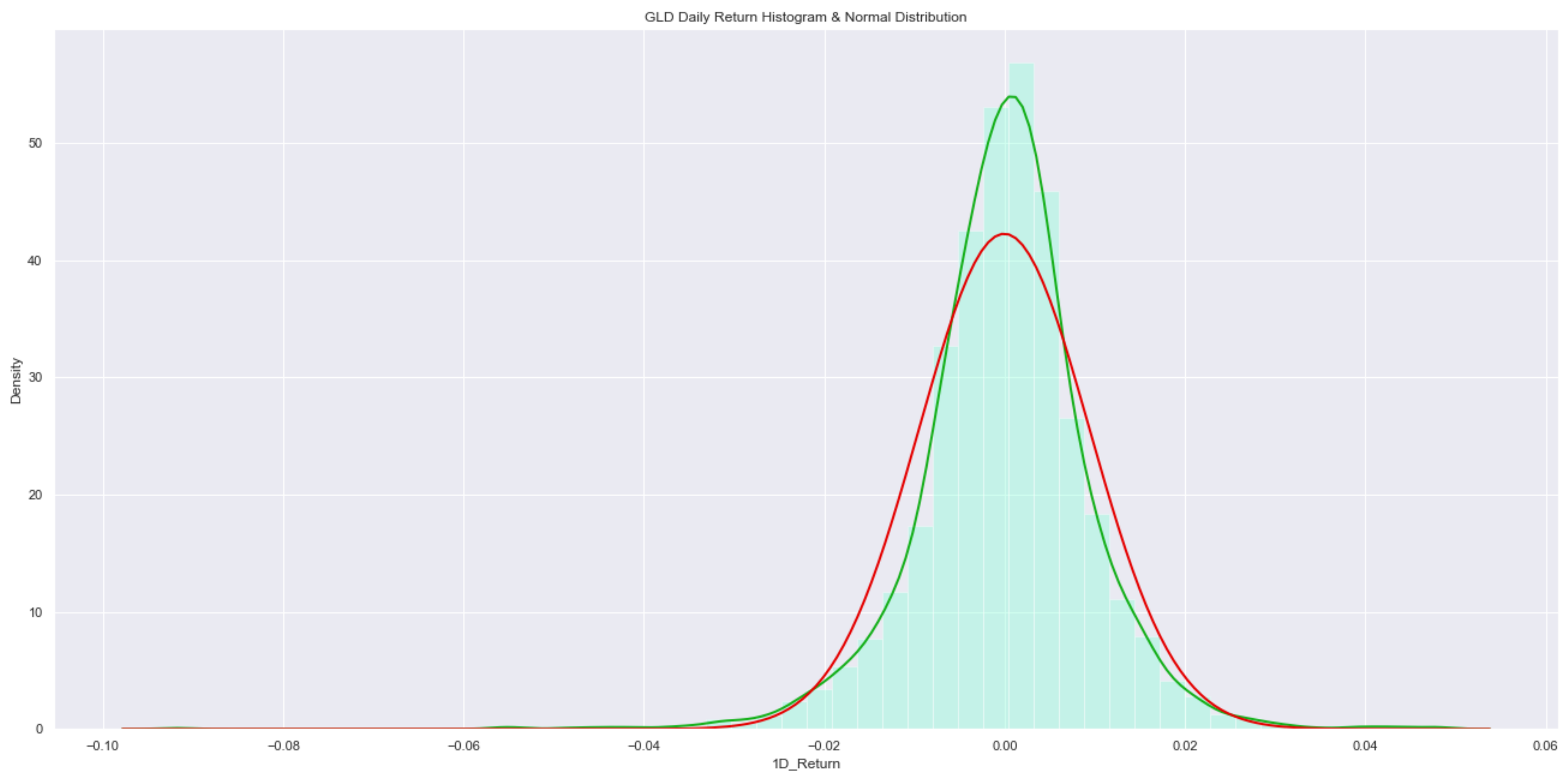
$$r_t = log\left(\frac{p_t}{p_{t-1}}\right)$$

where r_t is the daily return of the t^{th} day and p_t is the adjusted close price of the t^{th} day.

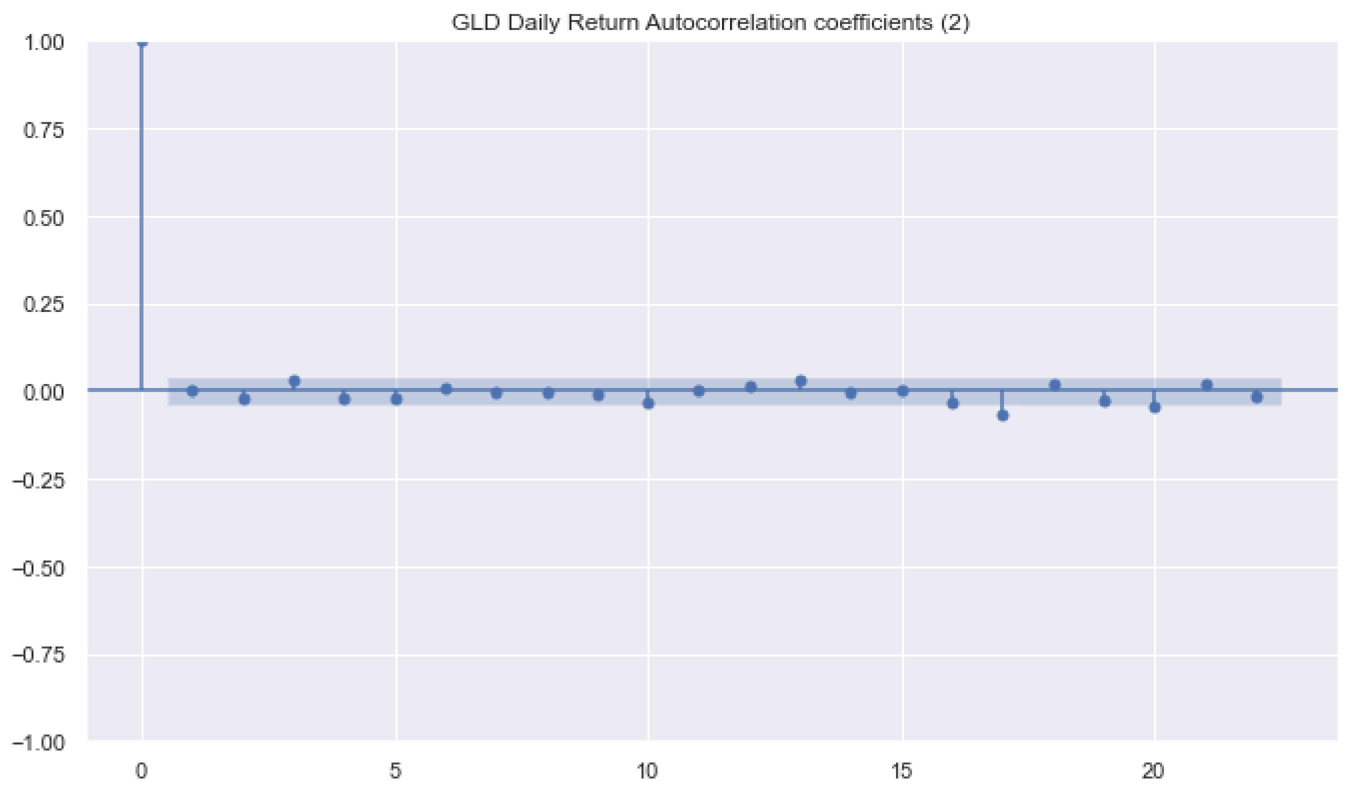
The plot below shows the daily return of GLD during the 10 years. From the plot, it can be seen that the return is around 0 (with a small number of big negative values) and has the property of mean reversion.

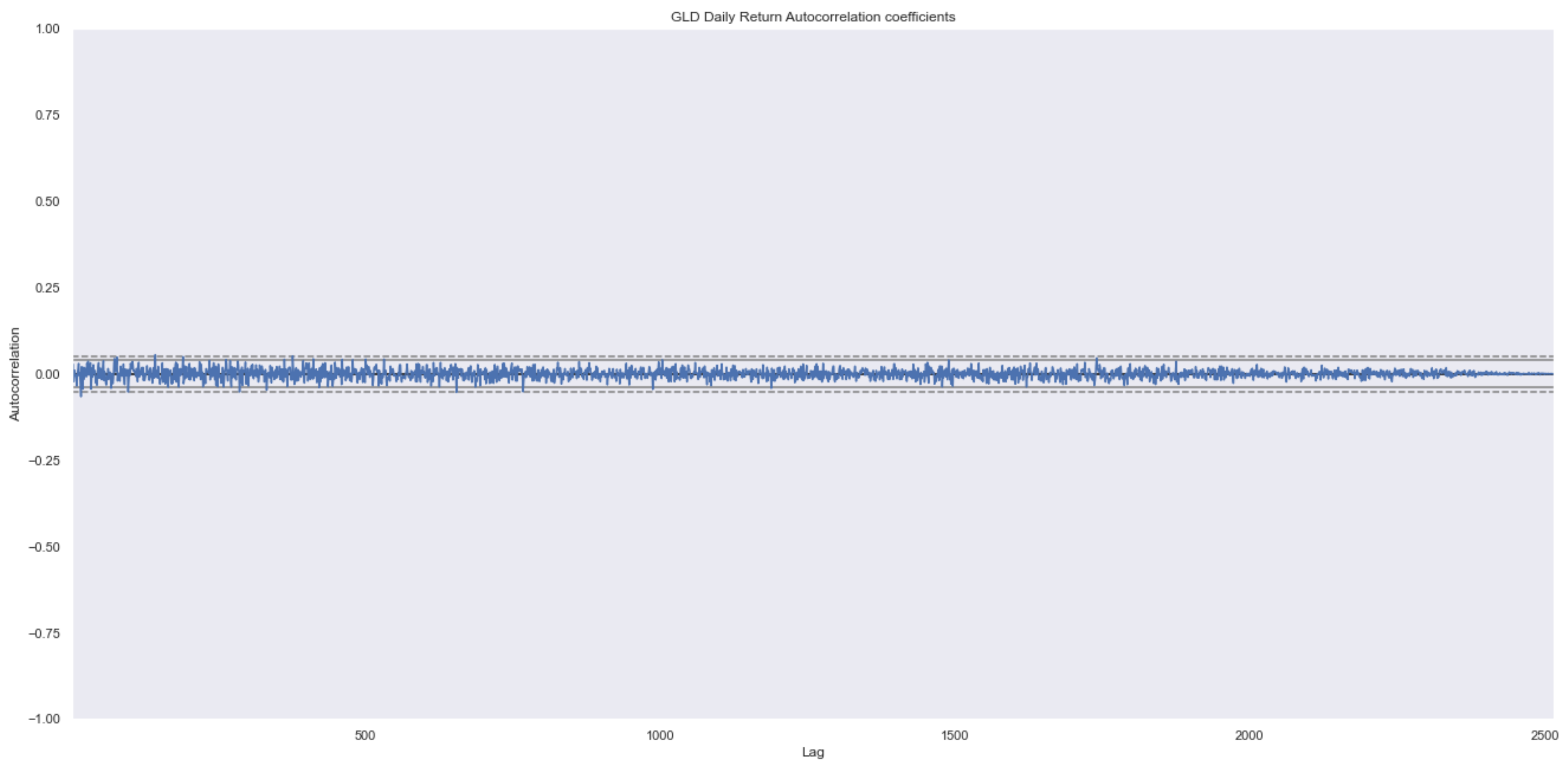


The following plot compares the KDE of the histogram of GLD daily returns (Green) and the Normal distribution curve (Red). It shows that the distribution of GLD daily returns is near Normal, but with higher peak and fatter tails (especially the negative tail) than the Normal distribution.



The next two plots analyse the condition of autocorrelation of GLD daily returns. The first plot shows the autocorrelation coefficients of lags from 0 to 22. The autocorrelation is very low (close to 0) for lag of more than 0. The second plot shows the autocorrelation of lags from 1 to the length of the dataset (10 years). The level of autocorrelation is always very low and even lower when the lag becomes large (more than around 1800 days). So the conclusion is that there is no significant correlation between r_t, r_{t-1}. Considering the fact of that GLD data has near Normal return with no significant one-day autocorrelation, it satisfies the Efficient Markets assumption.





- **B.2.2 Model Target (Target)**

The mean of the daily return is -0.00429% and the median is 0.0294%. It satisfies the observation of that the distribution of daily return has fatter and longer negative tail than the positive tail. Here, daily return feature (1D_Return) is used to determine the target of the model, the threshold is the median, which means: **target value is set to be 0 if the return of previous day is equal or below the median, target value is set to be 1 if the return of previous day is above the median.** The target column is given name as "Target" in the data.

B.3 Feature Engineering

In B.3 part, the project will follow the process of **Feature Generation** and **Feature Selection**.

- **B.3.1 Feature Generation**

In Feature Generation part, several new features are constructed using the six features of the original data and "1D_Return". The new features are not expected to have the predictive powers, in the next part B.3.2 they will be selected according to several criterion. The main purpose of this part is to generate the possible new features. The new features can be classified as following:

- B.3.1.1 Intraday Price Range

  Open-Close range and High-Low range are used to represent the information of the intraday price range, given names of "O-C" and "H-L" respectively. "O-C" is the difference between the open price and closed price for each day; "H-L" is the difference between the highest price and lowest price of each day.

- B.3.1.2 Sign of Return

  Sign of return indicates the positiveness or negativeness of the return of each day. It is given name of "Sign" and set to be -1 if the return is 0 or negative and 1 otherwise.

- B.3.1.3 Past Returns

  In this project, the n-day lagged returns are taking into account as features with n equals to 1, 5, 22 corresponds to 1-day, 1-week and 1-month lags respectively. These features have names of "1D_LagReturn", "5D_LagReturn" and "22D_LagReturn".

- B.3.1.4 Momentum

  Momentum indicates the price change over k period and has the formula of: $p_t - p_{t-k}$. Same as Past Returns, the period k has the value of 1, 5, 22 with feature names "1D_Momentum", "5D_Momentum" and "22D_Momentum" respectively.

- B.3.1.5 Simple Moving Average

  Simple moving average is the arithmetic average of the latest n days' adjusted closed price. It is given formula of:

$$SMA_t = \frac{1}{n}\sum_{i=0}^{n-1} p_{t-i}$$

  Here n has values of 3, 5, 10, 15, 22 with feature names "3D_SMA", "5D_SMA", "10D_SMA", "15D_SMA" and "22D_SMA" respectively.

- B.3.1.6 Exponential Moving Average

  Exponential moving average has the formula:

$$EMA_t = \beta\, EMA_{t-1} + (1-\beta)\, p_t$$

  with $EMA_0 = p_0$ and β is a parameter, high $beta$ results in the higher influence of the histroy. β takes the value of 0.1, 0.3, 0.5, 0.7 and 0.9. These feature is named as "0.1_EMA", "0.3_EMA", "0.5_EMA", "0.7_EMA" and "0.9_EMA".

- B.3.1.7 Cumulative Return

  Cumulative return is the sum of the latest n days. n takes values of 5, 10, 15, 22 corresponding to one week, two weeks, three weeks and one month cumulative returns respectively. The feature names are "5D_CR", "10D_CR", "15D_CR" and "22D_CR".

Until now, there are 30 features. The dimensions is high, for the consideration of model performance and computational speed, it is necessary to reduce the dimensionality. In the next part, a feature selection process is applied.

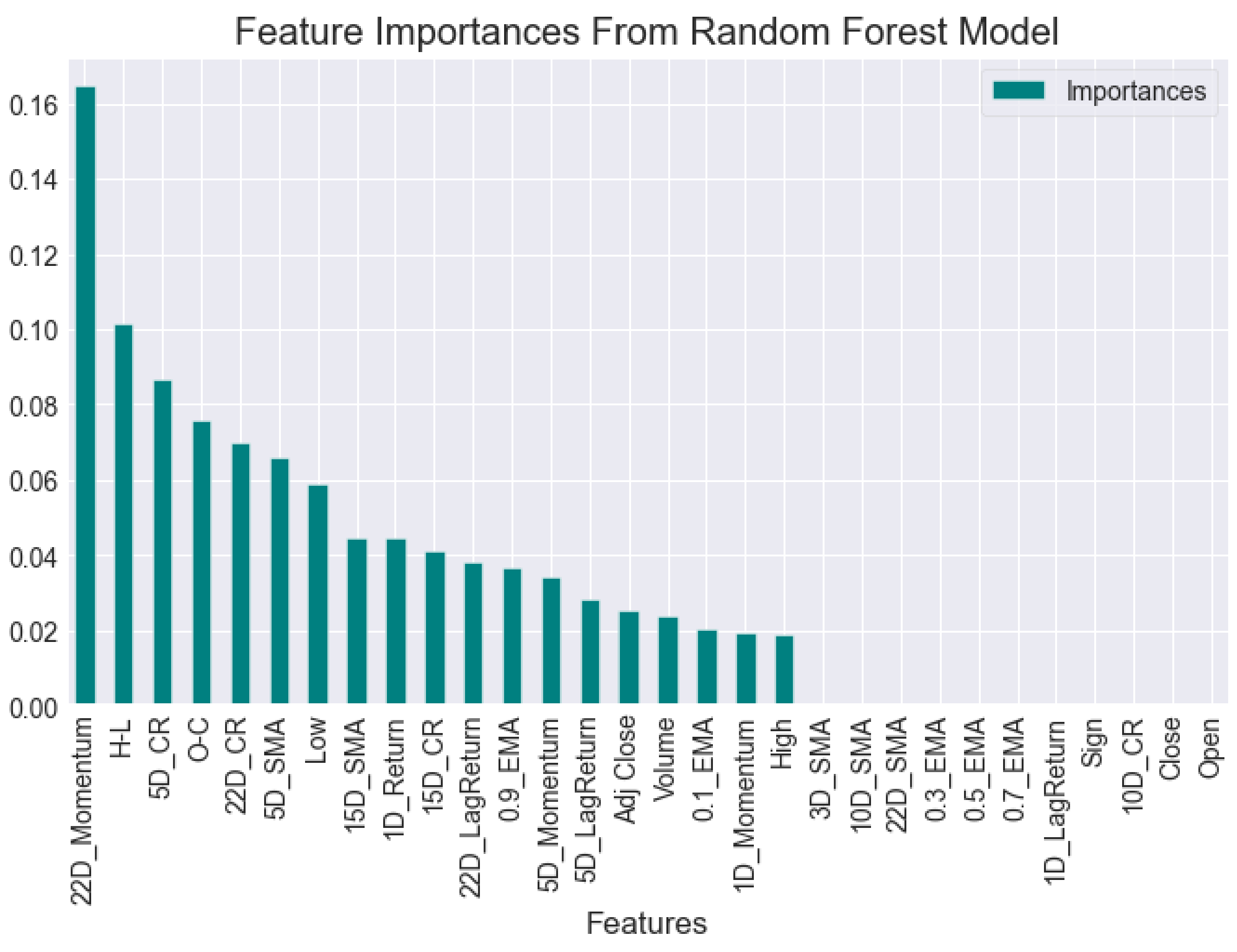
- **B.3.2 Feature Selection**

- B.3.2.1 Random Forest Feature Importance

The method **Random Forest Feature Importance** is used to have a look at all the features importance in this classification task. The feature importance is determined by the Random Forest Classifier according to the amount of purity improvement given by the feature (if Gini Impurity loss is used) or the amound of entropy decreased by the feature (if entropy loss is used). As tree based models are quite robust and convenient in sense of that:

1. they can be applied on both continuous features and categorical feature (in this task is the "Sign");
2. they are not sensitive to feature scaling and missing values;

which makes them the best models to apply at the first place. Here I used random forest model with number of estimators (trees) of 20 and maximal depth of 2. These two parameters are found by GridSearch of cross-validation of the whole dataset using the scoring metric of ROC-AUC. The candidate values of hyperparameter n_estimators are 5,10,20,30,40 and 50; the candidate values of the hyperparameter max_depth are 2,4,6,8,10,20 and 30. The best hyperparameters are n_estimators=10 and max_depth=2 with cross-validation ROC-AUC score of 0.5227. The feature importances given by the model with the best parameters are displayed in the plot:

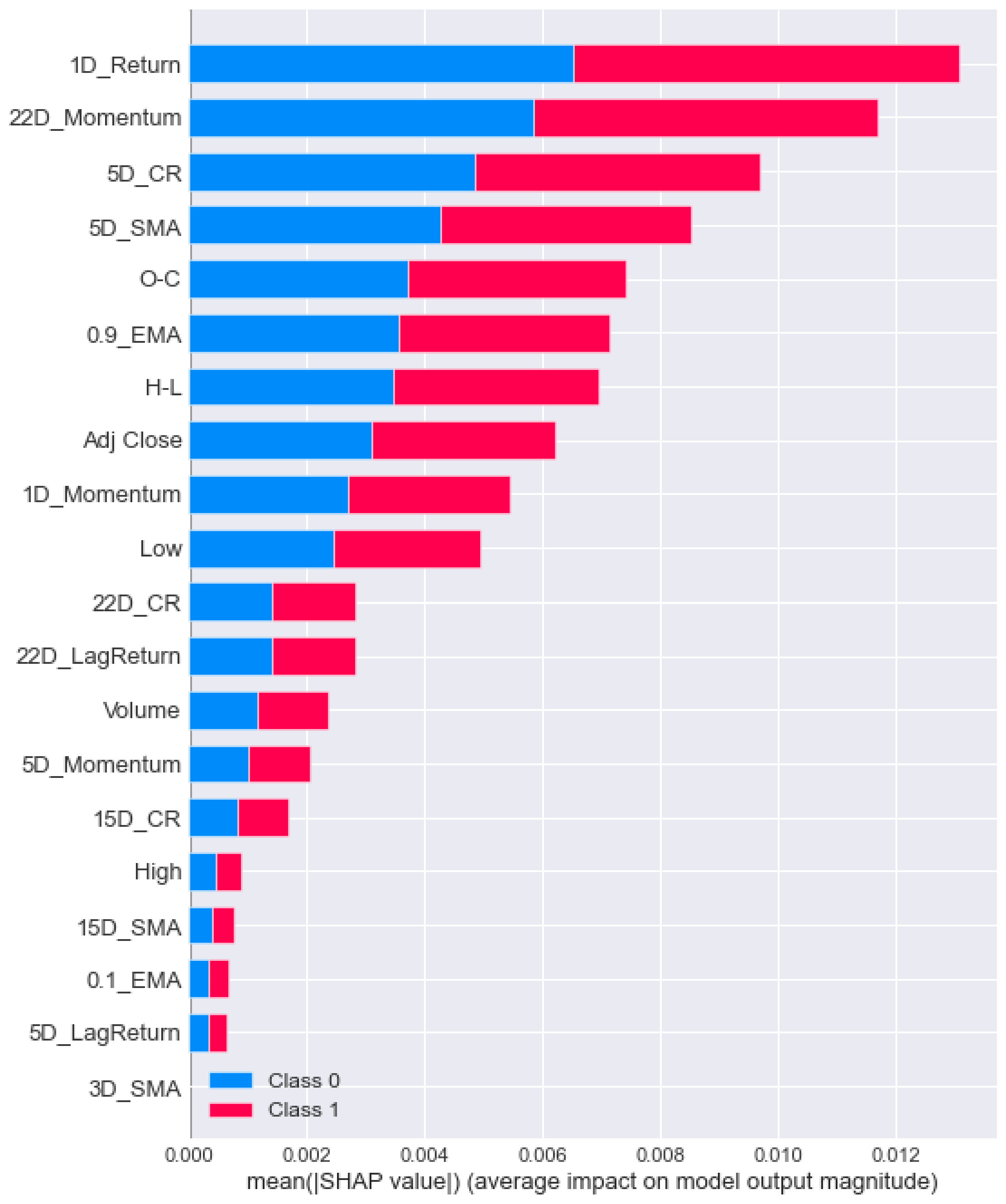


According to the plot there are 11 features of nearly 0 importance and can be removed from the dataset for the lack of predictive powers.

- B.3.2.2 SHAP

Shapley Additive Explanations (SHAP) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions.

Shapley values are a widely used approach from cooperative game theory that come with desirable properties and is the average marginal contribution of a feature value across all possible coalitions.



The above plot is the feature importance visualization from the SHAP. The conclusion is actually the same as B.3.2.1.

- B.3.2.3 Multicolinearity

After B.3.2.1, the left features are all numerical and continuous features. Many of them are calculated from the same or same set of the orignal features, so there might be high multicolinearity problem. Multicollinearity occurs when two or more independent variables are highly correlated with one another. It affects the performance of the regression models a lot but hardly influences the classification model, so there is no need to apply feature selection for this reason.

- B.3.2.4 ANOVA & t-test

As all the remaining features are numerically continuous and the target is categorical, it is interesting to know if there is any differences between the feature values of groups of that target equals to 1 and equals to 0. So I decided to use the One-Factor Analysis of Variance (ANOVA) and t-test methods to find out the conclusion.

Analysis of variance, or ANOVA, is a statistical method that separates observed variance data into different components to use for additional tests. One-way ANOVA is used for different groups of data to gain information about the relationship between the variables. It is used to compare whether the means of the samples from multiple groups are the same/whether multiple-group samples are of the same distribution. ANOVA uses F value (ANOVA coefficients) as the measurement. High F values with a low p-value indicate the different means of the feature values of different groups (there may exist predictive powers).

t-test is very similar to ANOVA in this case. It also used for whether the means of two groups are significantly different. And same as ANOVA, High t values with a low p-value indicate the different means of the feature values of different groups (there may exist predictive powers).

In this task, the data is seperated into two groups according to the value of "Target". The the samples of the same feature from the two groups are compared using both ANOVA and t-test. The result is sorted in the descending order of F and displayed below:

	F	ANOVA p	t	t-test p
Features				
1D_Return	5.354165	0.020754	2.313907	0.020754
1D_Momentum	3.632961	0.056761	1.906033	0.056761
22D_CR	3.347360	0.067432	1.829579	0.067432
22D_Momentum	3.153296	0.075896	1.775752	0.075896
O-C	2.697328	0.100643	-1.642354	0.100643
15D_CR	1.772359	0.183212	1.331300	0.183212
5D_LagReturn	1.171888	0.279118	-1.082538	0.279118
5D_Momentum	0.658139	0.417295	0.811257	0.417295
Volume	0.517748	0.471872	-0.719547	0.471872
5D_CR	0.503890	0.477862	0.709852	0.477862
H-L	0.124704	0.724017	0.353135	0.724017
22D_LagReturn	0.090857	0.763116	-0.301424	0.763116
0.9_EMA	0.056643	0.811902	-0.237998	0.811902
15D_SMA	0.047315	0.827820	-0.217521	0.827820
5D_SMA	0.036469	0.848565	-0.190969	0.848565
Low	0.019204	0.889795	-0.138578	0.889795
High	0.015623	0.900541	-0.124991	0.900541
0.1_EMA	0.011458	0.914765	-0.107042	0.914765
Adj Close	0.009176	0.923695	-0.095791	0.923695

The result shows that the p-value of ANOVA and t-test for the same feature is the same, which is reasonable. And the F value and the absolute value of t value also have the same ranking order for all the features, which is reasonable as well. Because of the fact that there is no single feature having the significant predictive power in this classification task, there is no feature with high enough F / t value and low enough p-value.

```
- At significance level of 1% (p-value < 0.01), no feature is relevant.
- At significance level of 5% (p-value < 0.05), "1D_Return" is relevant.
- At significance level of 10% (p-value < 0.1), "1D_Return",  "1D_Momentum", "22D_CR" and "22D_Momentum"
are relevant.
```

However, here I just select the features which are significant relevant at 90% of confidence (with p-value < 0.1) as the final features together with 3 more features considering the results given by the Random Forest model. These 7 features will be used in the SVM classification modelling. The features used for building the model are:

- **1D_Return**: the daily return of GLD ETF
- **1D_Momentum**: the price change over 1 day period
- **22D_CR**: the one month cumulative return of GLD ETF
- **22D_Momentum**: the price change over 1 month period

According to the remained features, it seems that the return of 1 day, 1 month and the price change of 1 day, 1 month are relevant. And actually the return and momentum are both indicators of the price change in a period. This is intuitively reasonable considering that the target is actually also the price change (of the next day).

B.4 Clean and Transform data

In this step, the data is cleaned (rows with NaN deleted and the features not needed are deleted) and saved locally as "**GLG_cleaned.csv**". Then the dataset is divided into the feature set X and target vector y.

B.5 Modelling and Hyperparams Tuning

- **B.5.1 General View of SVC and SVM**

Support Vector Classifier (SVC), also called as soft margin classifier, is a classification method using hyperplane with maximal soft margin as the decision boundary. The hyperplane is determined by some of the samples called Support Vectors. Support Vector Machine is an extension of support vector classifier.It enlarges feature space using kernels and is able to accommodate non-linear boundaries between the classes. It implicitly creates features to project into higher dimensional space. The hyperplane divides the classes in the higher dimension space and results in the decision boundary on the original dimensional space is non-linear.

In this task, a SVM is applied at first to have a general view of the performance. The scaling method and parameters are as below:

```
Scaling method: MinMaxScaler
C: 1000.0
kernal: rbf
gamma: 0.1
```

The ROC-AUC score of this hyperparameter set is 0.522.

- **B.5.2 Hyperparameters Tuning**

According to the scikit learn offical web, SVC has many hyperparameters, and some of them are usage restricted with respect to the kernal selected: for example, "degree" is only valid for polinomial kernal function and ignored by the others. Next the general hyperparameters of SVC is listed:

```
C: Regularization parameter. The strength of the regularization is inversely proportional to C. Must be
strictly positive. The penalty is a squared l2 penalty.

kernel: Specifies the kernel type to be used in the algorithm. If none is given, ‘rbf’ will be used. If
a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should
be an array of shape (n_samples, n_samples).

gamma: Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’. If gamma='scale' (default) is passed then it
uses 1 / (n_features * X.var()) as value of gamma. If ‘auto’ it uses 1 / n_features.
```

So in the first step, these three hyperparameters are tuned. The hyperparameter tuning consists of GridSearch using the ROC-AUC score of the 5-folds cross-validation. The 5 time series folds used in the cross-validation are determined by the Forward Chaining method.

The search grid is:

```
Scaler: [StandardScaler, MinMaxScaler, RobustScaler]
C: [0.01, 0.1, 1, 10, 100, 1000, 10000]
kernal: ["linear", "poly", "rbf", "sigmoid"]
gamma: ["scale", "auto", 1e-07, 1e-06, 1e-05, 0.0001, 0.001, 0.01, 0.1]
```

The GridSerch method gives the result of ROC-AUC score of 0.555 and hyperparameter combination of:

- **Scaler: StandardScaler()**
- **C: 100**
- **kernal: 'poly'**
- **gamma: 'scale'**

Applying the best hyperparameters above, the model is trained using the whole training set in the next part.

- **B.5.3 Final Model**

The ROC-AUC score on the test set given by the SVM model with best hyperparameters is **0.525**.

- **Scaler: StandardScaler()**
- **C: 100**
- **kernal: 'poly'**
- **gamma: 'scale'**

The model with this set of hyperparameters is called "the model" in the later part of report.

There is an interesting observation of that: SVM is a relatively slow algorithm (at least comparing to Random Forest) according to the experiment. In the feature selection part (B.3.2.1), I used the Random Forest model to get the feature importance of 30 features and even more training dataset (I didn't split the dataset into train and test sets) and the time for training the model is less in 3s. The SVM model of only 4 features and smaller training dataset (0.8 of the whole) took about 20s for training. I privately tried to train the SVM with all the 30 features with the 0.8 of whole dataset: the training time is unbearably long. This dues to the high computational time and storage consumption of the $n \times m$-order matrix calculations for getting the support vectors, where n is the number of training samples and m is the number of features.

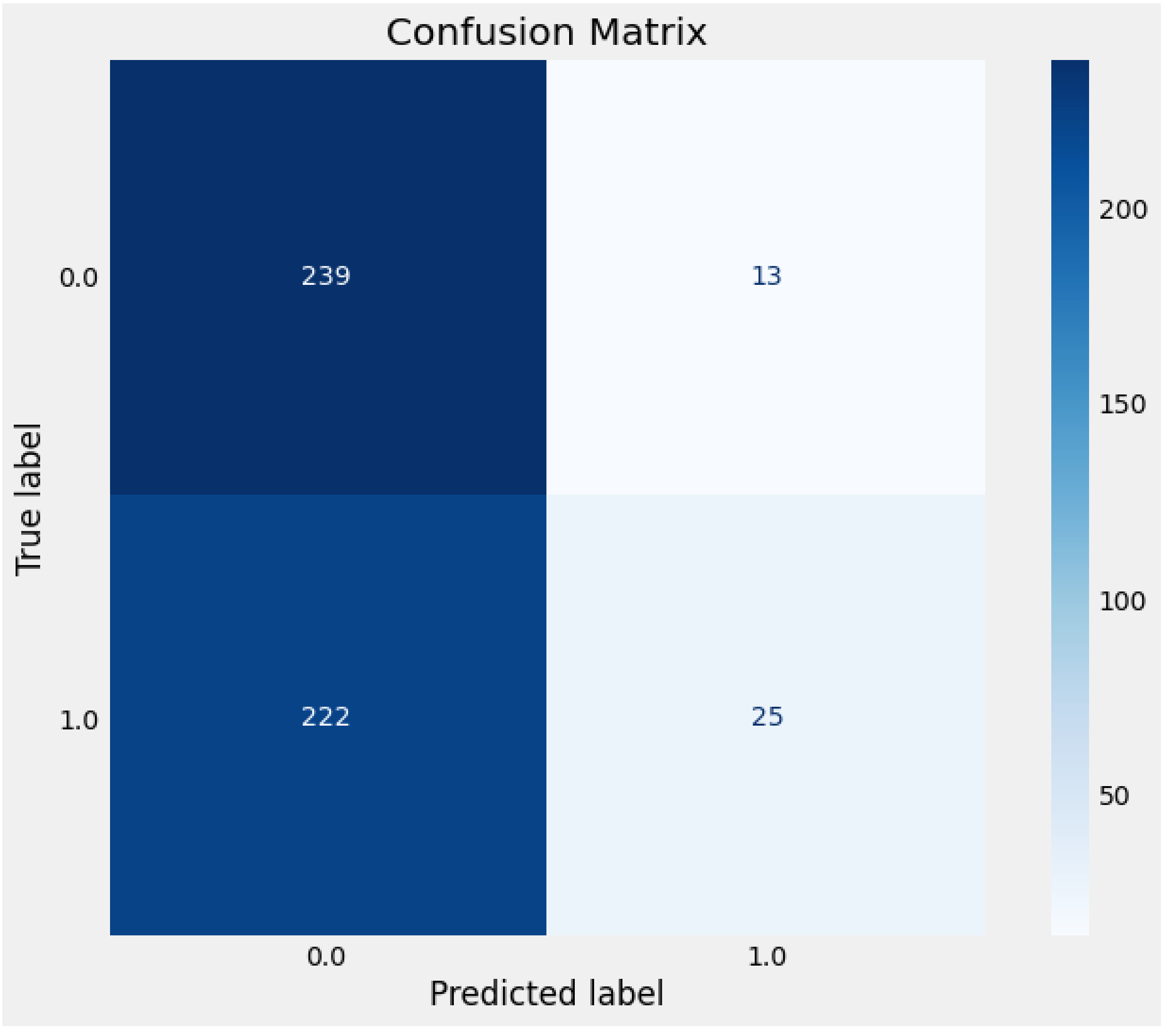
B.6 Validation and Evaluation

In general, the prediction quality of the model is not outstanding. However, it gives relatively good and reasonable result in the financial domain of one-day stock & EFT move trend prediction.

- **B.6.1 Accuracy**

Accuracy is a metric for measuring the predict correctness of the model. The model has the accuracy of 0.5283 and **0.5291** for the training and test set respectively. The accuracy on the test set is quite satisfactory.

- **B.6.2 Confusion Matrix**



The plot above is the Confusion Matrix of the model: numbers in each quadrant represent the number of test samples predicted with respect to their true label. The four quadrants are true positive, false positive, true negative and false negative:

- True Positive (TP): the samples with true label of 1 and predicted as label 1, for the model is 25
- False Positive (FP): the samples with true label of 0 and predicted as label 1, for the model is 13
- True Negative (TN): the samples with true label of 0 and predicted as label 0, for the model is 239
- False Negative (FN): the samples with true label of 1 and predicted as label 0, for the model is 222

From the result, it seems that the model is more likely to give negative prediction (0) while the data set is balanced in sense of the target (the median of the return is used to form the target). In this sense, the model is more "**Conservative**": it tends to recall all the negative trends in order to prevent the users from making a loss, while giving the positive prediction in a more prudent way.

- **B.6.3 Classification Report**

There are several more metrics to evaluate the performance of the model:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall\,(Sensitivity\,/\,True\,Positive\,Rate) = \frac{TP}{TP + FN}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Precision and recall are two numbers which together are used to evaluate the performance of classification or information retrieval systems. Precision is defined as the fraction of relevant instances among all retrieved instances. Recall, sometimes referred to as ‘sensitivity, is the fraction of retrieved instances among all relevant instances. A perfect classifier has precision and recall both equal to 1. Precision and recall should always be reported together. Precision and recall are sometimes combined together into the F1 score, if a single numerical measurement of a system's performance is required. Following is the classification report of the model which mentions the cited metrics:

```
              precision    recall  f1-score   support

         0.0       0.52      0.95      0.67       252
         1.0       0.66      0.10      0.18       247

    accuracy                           0.53       499
   macro avg       0.59      0.52      0.42       499
weighted avg       0.59      0.53      0.43       499
```

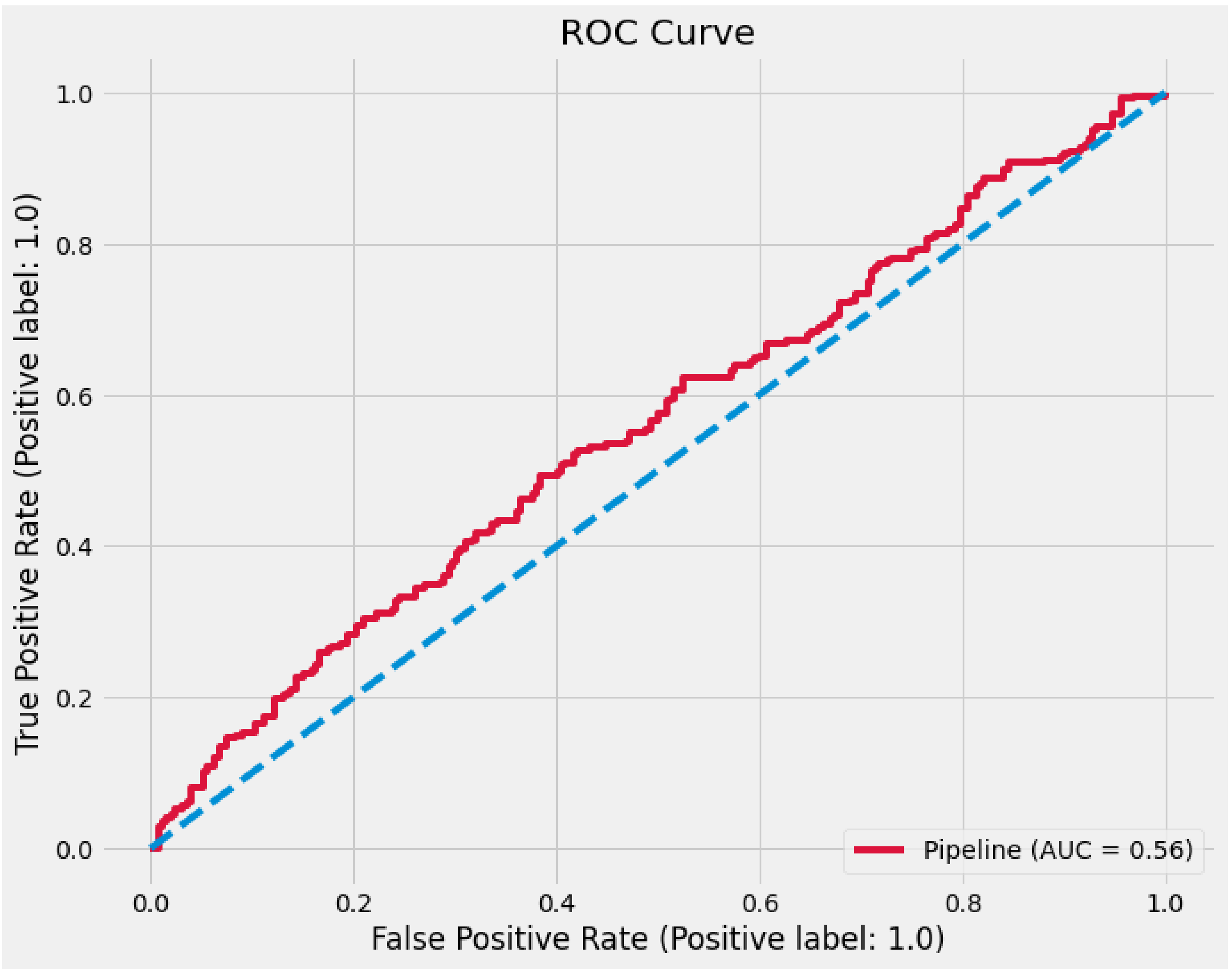
The report shows the same conclusion derived in B.6.2: as the more tend to give negative predictions, the recall of class 0 is very good (0.95). The prediction of 1 is made more carefully and accurate with precision of class 1 is 0.66.

- **B.6.4 Area under ROC Curve**

Receive Operating Characteristic (ROC) curve is a plot of the false positive rate versus the true positive rate (Recall). False positive rate is defined as the inverted specificity (ie., 1-Specificity) where Specificity is defined as:

$$Specificity = \frac{TN}{TN + FP}$$

ROC curves are used where there is equal number of observations of each class, which is satisfied in this task. The area under the ROC curve (AUC) is a measure of how well a model can distinguish between two classes. The model with higher AUC has better predictive ability.



The plot above is the ROC curve, where the red line represents the performance of the model and the blue dashed line represents the performance of the random prediction. From the plot, it shows that the predictive power of the model is better than the random model, although not much better.

B.7 Backtesting and Trading Strategy

Now a trading strategy is defined. There will be strategies for both markets that allow and not allow to short or people choose to and not to short.

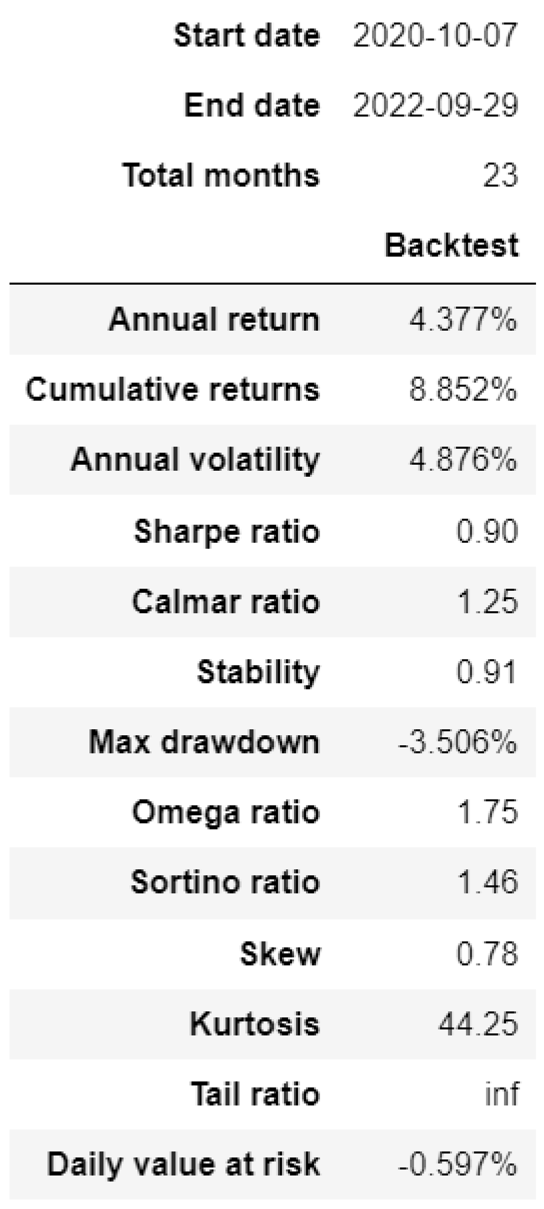
The backtesting period is from 2020-10-07 to 2022-09-29. The cumulative returns of both the strategy and "buy and hold" in both markets of the backtesting period is listed as:

	Without Short	**With Short**
Buy and Hold	-12.77%	-12.77%
Strategy	**9.11%**	**36.41%**

The results are quite good.

- **B.7.1 Without Short Position**

In the market of that short is not allowed or people choose not to short, a value of +1 is assigned for the buy signal and 0 otherwise from the predicted price y_pred for the outsample horizon. Then there is a comparison of the result of this strategy with the buy and hold and visualize the performance of the strategy.



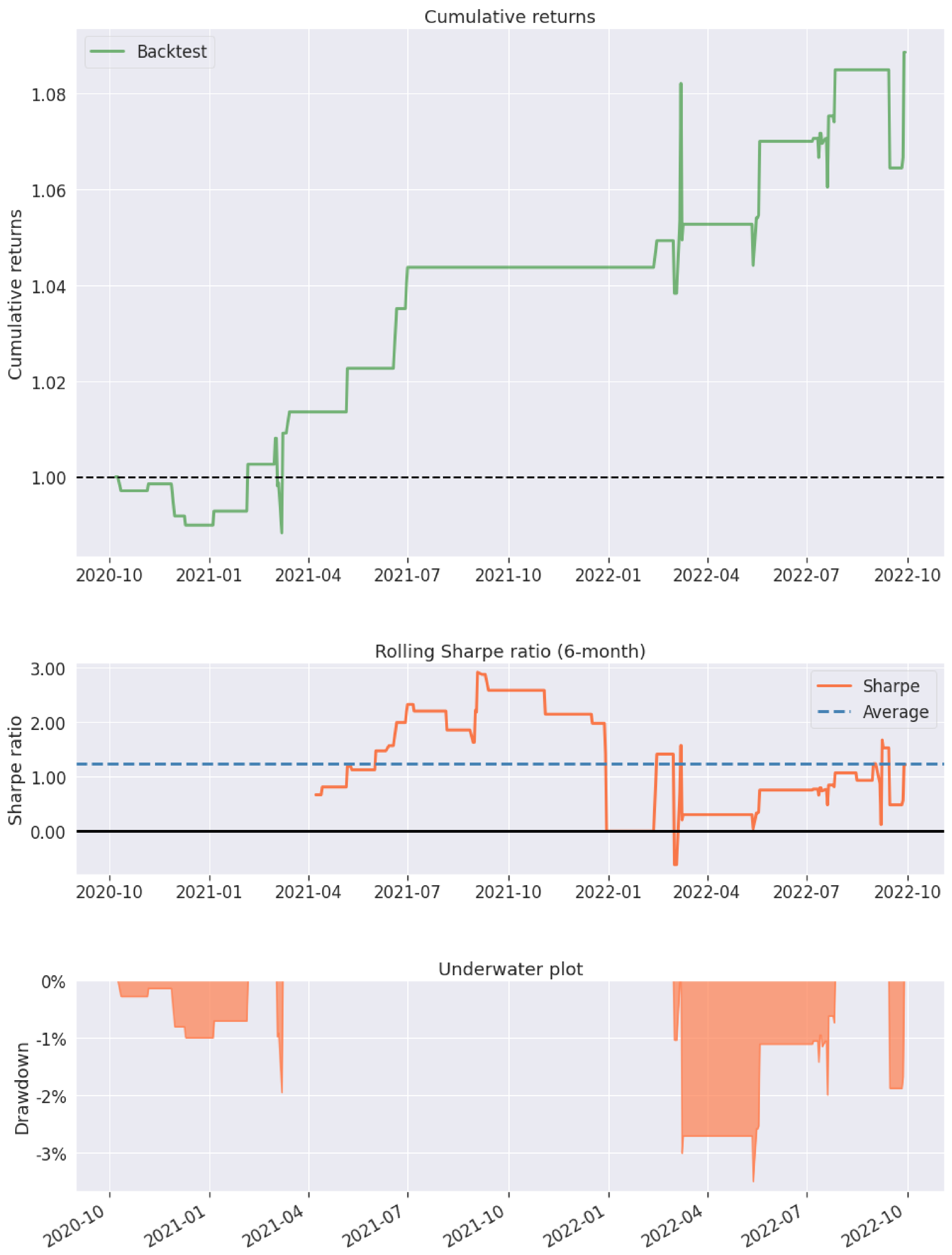
Start date	2020-10-07
End date	2022-09-29
Total months	23

	Backtest
Annual return	4.377%
Cumulative returns	8.852%
Annual volatility	4.876%
Sharpe ratio	0.90
Calmar ratio	1.25
Stability	0.91
Max drawdown	-3.506%
Omega ratio	1.75
Sortino ratio	1.46
Skew	0.78
Kurtosis	44.25
Tail ratio	inf
Daily value at risk	-0.597%

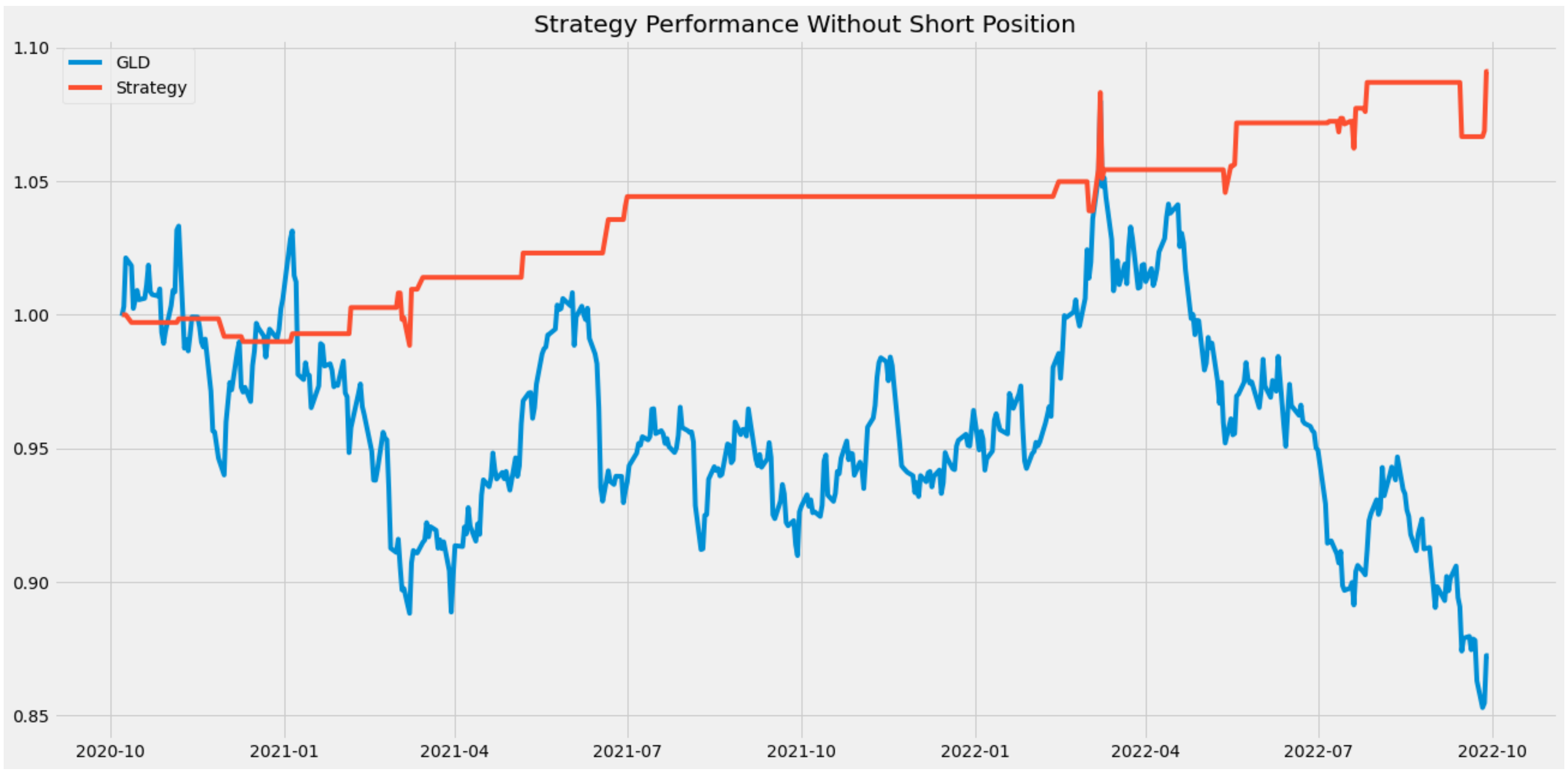
The table above shows the performance of the trading strategy, some important metrics are:

- Annual return 4.377%
- Cumulative returns 8.825%
- Annual volatility 4.876%
- Sharpe ratio 0.90
- Max drawdown -3.506%
- Daily value at risk -0.597%

It does not make a gain so much but gives a relatively good return with respect to the risk (Sharpe ratio of 0.9 with max drawdown of only 3.506%).



Above are the plots of Cumulative returns, Rolling Sharpe ratio (6 month) and Underwater obtained from the strategy. It can be seen that in most of the time, the cumulative return of the strategy is higher than the cumulative return of "buy and hold". Since 2021-03 the cumulative return of the strategy keeps outperforming the "buy and hold", which means the model's predictions are relatively more accurate since then. The Rolling Sharpe ratio is relatively high during the period from 2021-05 to 2022-01 (higher than 1).



Above plot makes a clear compare of the cumulative returns of the strategy and "buy and hold". From the plot, there is a cross between the curves of the cumulative returns of the strategy and "buy and hold" bwtween 2021-01 and 2021-02. After that, the strategy outperforms the "buy and hold".

- **B.7.2 With Short Position**

In the market of that short is allowed or people choose to short, a value of +1 is assigned for the buy signal and -1 otherwise from the predicted price y_pred for the outsample horizon. Then there is a comparison of the result of this strategy with the buy and hold and visualize the performance of the strategy.

Start date	2020-10-07
End date	2022-09-29
Total months	23

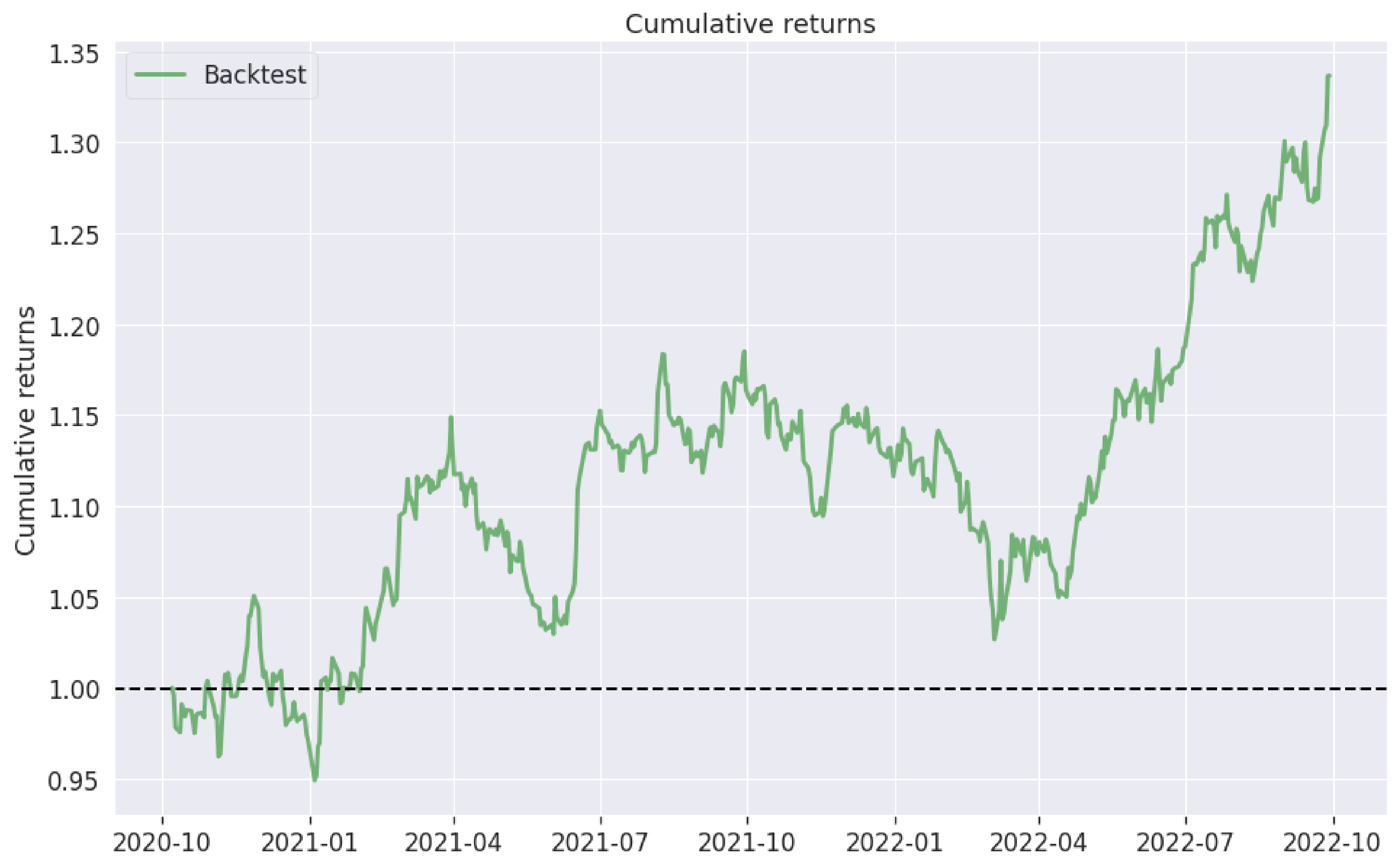
	Backtest
Annual return	15.774%
Cumulative returns	33.647%
Annual volatility	14.395%
Sharpe ratio	1.09
Calmar ratio	1.18
Stability	0.62
Max drawdown	-13.358%
Omega ratio	1.20
Sortino ratio	1.76
Skew	0.56
Kurtosis	1.73
Tail ratio	1.28
Daily value at risk	-1.751%

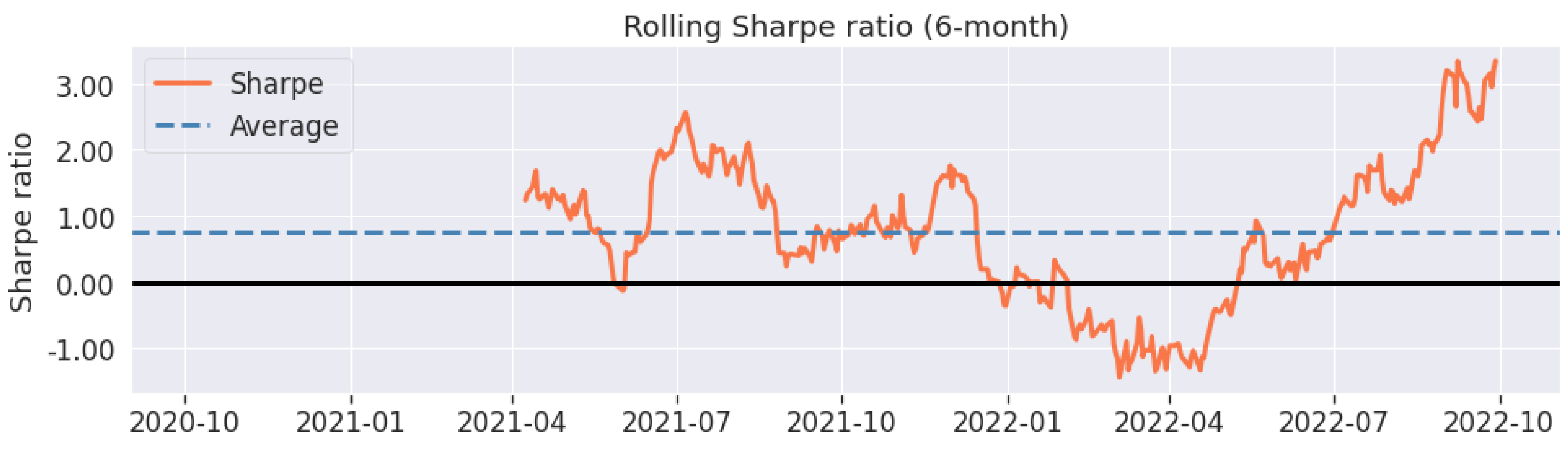
The table above shows the performance of the trading strategy, some important metrics are:

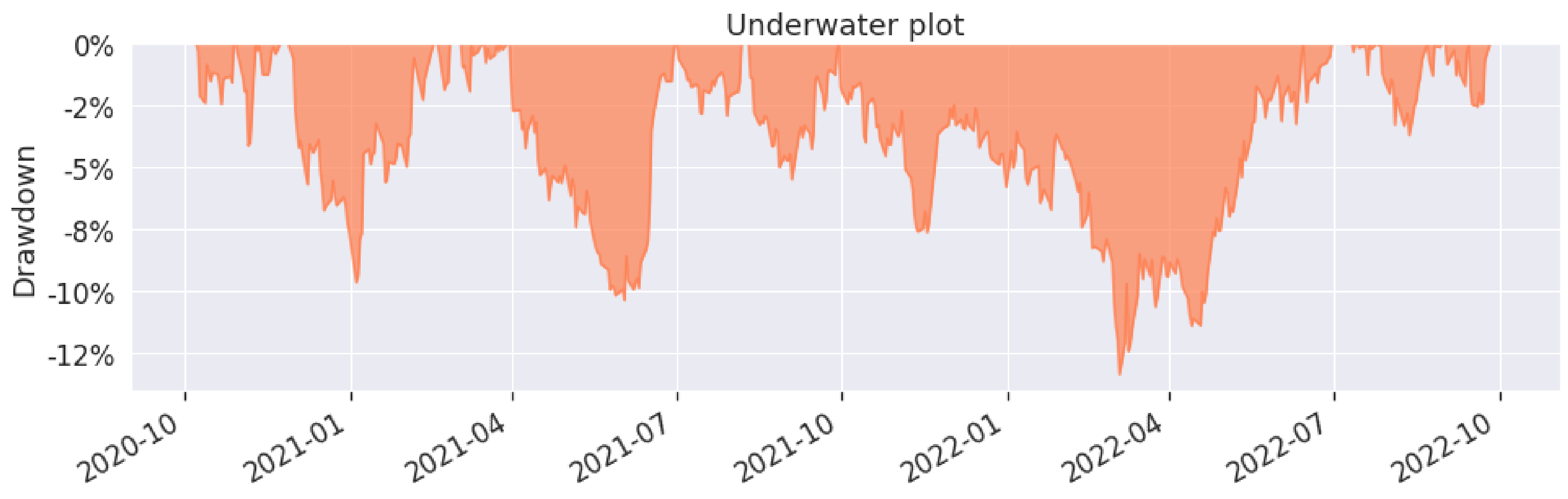
- Annual return 15.774%
- Cumulative returns 33.647%
- Annual volatility 14.395%
- Sharpe ratio 1.09
- Max drawdown -13.358%
- Daily value at risk -1.751%

In the market with short positions, the strategy is clearly much better than in the market without short positions. The strategy gives positive cumulative returns of 33.647% while the simple "buy and hold" gives negative returns of -12.77% during the period from 2020-10-07 to 2022-09-29. The strategy is quite good. This result is reasonable considering that the model have 0.95 of recall for predicting the negative class, which is much higher than the recall of predicting positive class (which is 0.1).

However, comparing to the strategy performance of the market without short position, the risk level becomes higher. The annual volatility rises from 7.069% to 14.395%, max drawdown rises from 3.506% to 13.358% and daily VaR rises from 0.597% to 1.751%. As the Sharpe ratio also increases, it seems to worth the price. But pay attention to the increase of daily VaR and max drawdown, short positions make these two risk measurements triplex and quadplex respectively.

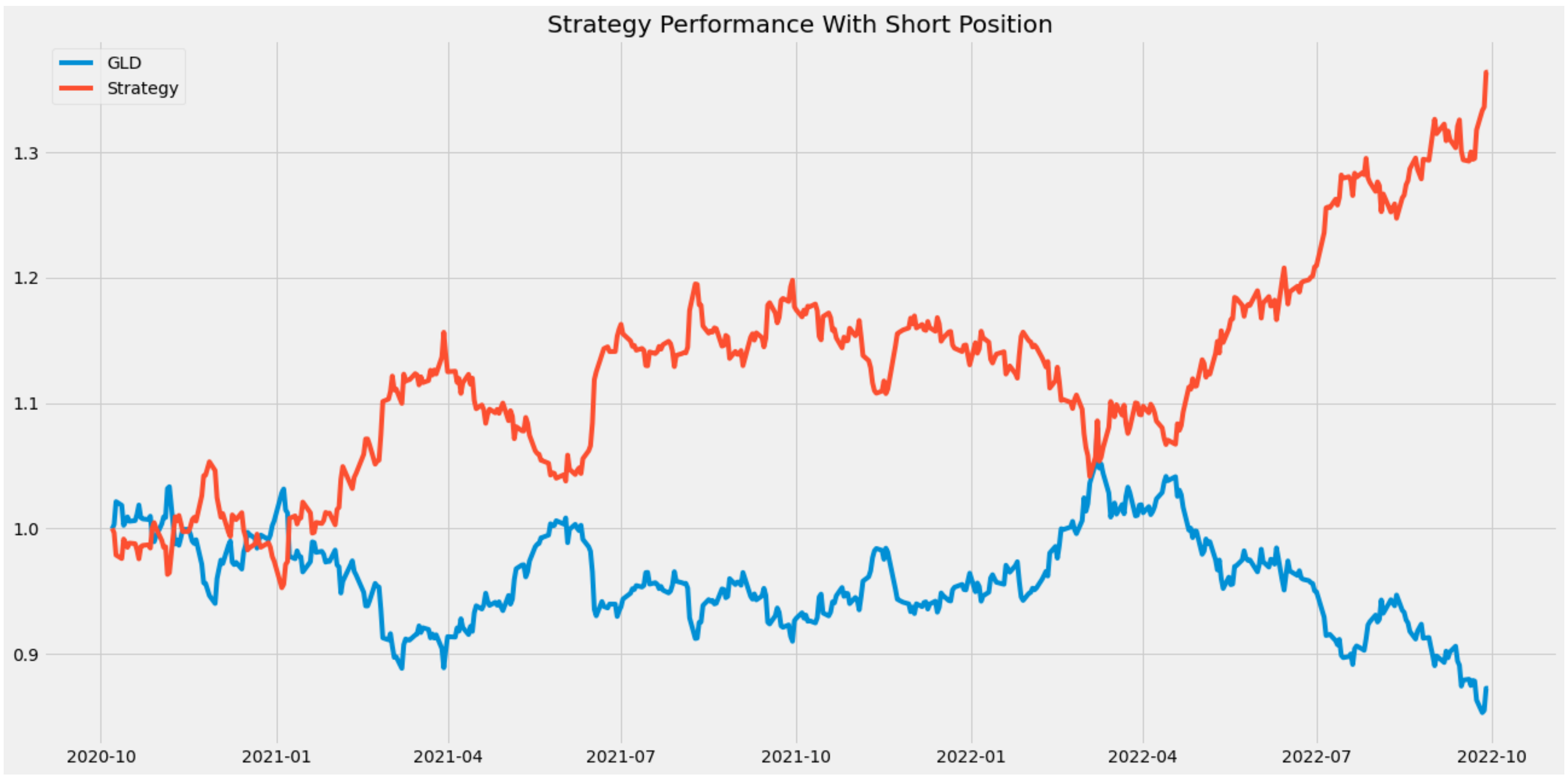






Above are the plots of Cumulative returns, Rolling Sharpe ratio (6 month) and Underwater obtained from the strategy. It can be seen that in most of the time, the cumulative return of the strategy is higher than the cumulative return of "buy and hold". After around 2022-02, the cumulative returns of the strategy keep outperforming the "buy and hold", which means the model's predictions are relatively more accurate since then.

However, compared with the same plot of in market without short, the curves seem to be more fluctuant.



The above plot makes a clear compare of the cumulative returns of the strategy and "buy and hold". From the plot, there are around 5 crosses between the two curves of the cumulative returns of the strategy and "buy and hold". However, the strategy almost always outperform the "buy and hold" during the whole period and rapidly rises after 2022-03.

References

- GLD ETF Introduction: https://www.spdrgoldshares.com/
- Paper Efficient feature selection filters for high-dimensional data by Artur J. Ferreira , Mário A.T. Figueiredo.
- One-Factor ANOVA (Between Subjects). Retrieved from http://onlinestatbook.com/2/analysis_of_variance/one-way.html
- Analysis of variance. (2019, February 04). Retrieved from https://en.wikipedia.org/wiki/A
- Scikit Learn SVC: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html?highlight=svc#sklearn.svm.SVC