**計量財務與機器學習**

**期末報告**

**Prepared by**

**109306091 資管三 蘇希甫**

1. **Introduction**
2. **Main axis of experiment-Time-Series**
   * 1. **Introduction of Time-Series**
     2. **Challenges in time prediction**
3. **Experiment Idea**
   * 1. **Objective**
     2. **background**
4. **Methodology**
   * 1. **How I conduct the experiment**
     2. **Variable definition**
5. **Model detail:**
   * 1. **Linear Methodology**
     2. **Random Forest**
     3. **Support Vector Machine**
     4. **Ridge regression**
     5. **Lasso regression**
6. **Result Comparison**
7. **Conclusion**
8. Introduction:
   1. This report investigates the prediction of financial time series data using various statistical and machine learning models. We specifically focus on Linear Regression, Random Forest, Support Vector Machines (SVM), Ridge Regression, and Lasso Regression models. We apply these models to historical financial data using an expanding window approach, which updates and retrains the models as new data becomes available. This strategy is preferred for its adaptability in dynamic financial environments. The models' performances are evaluated using metrics including Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Symmetric Mean Absolute Percentage Error (SMAPE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The report is structured to detail our methodology, present the results, discuss their implications, and conclude with the key findings.
9. The main axis of this experiment - Time Series:
   1. Introduction of time series:
      1. The CAD/JPY exchange rate returns, like many other financial time series, demonstrate time dependence. This means that past behaviors, fluctuations, and trends significantly influence future returns. Employing time series forecasting models allows us to exploit this inherent nature of financial data and capture these time dependencies, enhancing the efficacy of our forecasting.
      2. Moreover, time series data often exhibit specific characteristics such as trends, seasonality, and autocorrelation. A trend is a long-term increase or decrease in the data. Seasonality refers to periodic fluctuations, for instance, certain months may consistently exhibit higher returns. Autocorrelation, or serial correlation, describes the relationship of a variable with a lagged version of itself. In our context, it refers to today's returns being influenced by the returns of previous days or weeks.
   2. Challenge in time series prediction:
      1. Non-Stationarity: Financial time series data often exhibit non-stationarity, which violates the assumptions of many statistical models and can lead to unreliable forecasts.
      2. Autocorrelation: Autocorrelation stands for correlation between a time series and its own lagged version. This is addressed in our model by using lagged returns as the predictor variable.
      3. Overfitting: In time series forecasting, overfitting occurs when the model is too close to the historical data, capturing noise and signals. This may lead to poor prediction performance on new data. The regularization techniques we will discuss later, such as Ridge and Lasso regressions, are used to prevent overfitting.
10. Experiment Idea:
    1. Objective: Our primary objective is to predict the return on the CAD/JPY exchange rate using Lagged return as the forecast variable. The ultimate goal is to determine the most effective and efficient model to predict these returns.
    2. Background: Financial forecasting is an important activity in the financial world. It is critical to various financial operations and activities such as trading, risk management, budgeting, and strategic planning. For this project, we focus on CAD/JPY exchange rate returns as these affect a wide range of stakeholders from individual investors to multinational corporations.
11. Methodology:
    1. How I conduct my experiment:
       1. Firstly, we downloaded the data from Yahoo Finance, we cleaned the data and constructed new features like Weekly data and one week prediction and 12 weeks prediction. Then define x is the payoff using the last 12 weeks as the explanatory variable, and y is the current return, which is the explained variable. We use the rolling window prediction method which is “Out-of-sample 1-week rolling prediction”. We train our model in a 150-week window and then predict the next week's return. The process is repeated, moving the window forward one week at a time. Within the model that were used in the experiment, including Linear Regression, Random Forest, Support Vector Machines, Ridge Regression, Lasso Regression. Besides, in total, we did four different prediction methods:1-week rolling forecast, 12-week rolling forecast, 1-week extended rolling forecast, 12-week extended rolling forecast. Lastly, use five different metrics to evaluate all the models’ performance.
    2. Variable definition:
12. **data** - This is a dataset imported from the file "CADJPY.csv", which presumably contains data about the CAD/JPY exchange rate.

**Data Cleaning and Preparation**

1. **Date** - The 'Date' column from the data. The data is filtered to only include rows where 'Date' is later than "2013-12-31".
2. **yr** - The year component of the 'Date' column.
3. **wk** - The week number of the 'Date' column.
4. **d** - The day of the week from the 'Date' column.
5. **return** - Calculated as the percentage change in the 'Close' price from the previous row.
6. **return\_lagX** (where X ranges from 1 to 12) - The lagged returns, essentially the return from X weeks ago.

**Models and Predictions**

1. **train\_data** - The training dataset, which includes all columns except the first ten. This dataset changes as the rolling prediction window moves.
2. **test\_data** - The test data point, which is the row immediately following the current training dataset.

**Following models are trained and tested:**

1. **mod\_reg** - A linear regression model built with the lm() function, which predicts the 'return' based on all other variables in the training data.
2. **mod\_random** - A random forest model built with the randomForest() function, which also predicts the 'return' based on all other variables in the training data.
3. **mod\_svm** - A support vector machine model built with the svm() function, which again predicts the 'return' based on all other variables in the training data.
4. **mod\_ridge** - A ridge regression model built with the glmnet() function, which again predicts the 'return' based on all other variables in the training data. The lambda for the model is selected using cross-validation with the cv.glmnet() function.
5. **mod\_lasso** - A lasso regression model, also built with the glmnet() function and also using cross-validation to select the lambda value.

**Evaluation Metrics**

Several metrics are calculated to evaluate the accuracy of the predictions:

1. **MAE** - Mean absolute error.
2. **MAPE** - Mean absolute percentage error.
3. **SMAPE** - Symmetric mean absolute percentage error.
4. **MSE** - Mean squared error.
5. **RMSE** - Root mean squared error.
   1. Models that use in the experiment:
      1. Linear regression:
         1. Linear Regression is a simple algorithm, but it has some limitations: It assumes a linear relationship between features and objectives, which may not hold in the real world. It is particularly sensitive to outliers. If the dataset contains highly correlated predictive variables, it may show multiple co-linearities. Despite its simplicity, using linear regression as one of the models to predict CAD/JPY returns can be a good starting point for establishing baseline forecasts.
      2. Random Forest:
         1. Random forest method: creates a bootstrap dataset. it creates a decision tree by randomly drawing a sample of data and creating a new dataset of the same size but with different composition. the difference is that it takes into account all the features of each split in the tree, based on the randomness of the dataset creation and the features of the split. Each of these trees will be different.
         2. Random Forest approaching detail:
            1. data = na.omit(train\_data[,-(1:10)]) specifies the training dataset after removing rows with missing values and excluding the first 10 columns.
            2. mtry = ncol(na.omit(train\_data[,-(1:10)]))-1 Specifies the number of randomly selected prediction variables to be considered for each split. It is set to the number of columns in the training dataset minus one.
            3. subset = sample(1:nrow(na.omit(train\_data[,-(1:10)]))) Specifies the random subset of rows to be used for constructing each tree in the forest.
            4. ntree = 300 Specifies the number of trees to grow in the random forest.
      3. Support Vector Machine:
         1. SVM is a kind of supervised learning that uses the principle of statistical risk minimization to estimate the hypersequence of a classification, that is, to find a decision boundary that maximizes the margins between two classes so that they can be perfectly separated.
         2. Support Vector Machine approaching detail:
            1. SVM is a kind of supervised learning that uses the principle of statistical risk minimization to estimate the hypersequence of a classification, that is, to find a decision boundary that maximizes the margins between two classes so that they can be perfectly separated.
      4. Ridge Regression and Lasso regression:
         1. Cross Validation:
            1. Cross validation in Ridge and Lasso regressions: Cross validation is used to determine the optimal value of lambda (λ), which is the Regularization parameters of Lasso and Ridge. The principle of Cross Validation is to divide the training set into subsets. Then, the model is trained on all but one of the subsets and tested on the remaining subsets. This process is repeated for each subset.
         2. Ridge Regression:
            1. Ridge adds a penalty term to the linear regression cost function to shrink the coefficient to zero, thereby eliminating relatively useless variables. The cv.glmnet function is used to determine the optimal value of lambda by cross-validating Ridge. Select the lambda value of (mod1$lambda.min) and use the glmnet function to fit (mod\_ridge) with the selected lambda value.
         3. Lasso Regression:
            1. Lasso Regression: Lasso regression is similar to Ridge regression, but the penalty estimation method is different, and L1 Regularization is used instead of L2 Regularization.
            2. 13 x 1 sparse Matrix, from lag1 - lag12 are all nearly rounded down to 0
6. Performance Comparison:
   1. One week rolling prediction:
      1. Mean Absolute Error (MAE): This measures the average magnitude of the errors in a set of predictions, without considering their direction. Lower values of MAE mean better prediction performance. In my case, the Lasso model performs the best with the lowest MAE of 0.01016, followed closely by the Ridge model with an MAE of 0.01018.
      2. Mean Absolute Percentage Error (MAPE): This measures the average percentage error in the predictions. The Ridge model significantly outperforms the others with a MAPE of 106.21176. The Lasso model is a close second with a MAPE of 107.03701. The lower the MAPE, the better the model's performance.
      3. Symmetric Mean Absolute Percentage Error (SMAPE): This is an improvement over MAPE that adjusts the formula to handle both over-forecasts and under-forecasts equally well. Higher values indicate worse prediction performance. The Regular Regression (Reg) model performed best on this metric, with the lowest SMAPE of 1.55586.
      4. Mean Squared Error (MSE): This measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. The Lasso model performed the best in this metric, with the lowest MSE of 0.00018, followed closely by the Ridge model.
      5. Root Mean Squared Error (RMSE): This is the square root of the mean square error. RMSE is even more sensitive to large errors than MAE or MSE and heavily penalizes large errors in prediction. Again, the Lasso model performed the best with an RMSE of 0.01334, closely followed by the Ridge model.
      6. In summary, the Lasso model performed best across most metrics, indicating it might be the most accurate for predicting the CAD/JPY exchange rate based on your one-week rolling prediction analysis. The Ridge model also performed well, making it a close competitor. The Random Forest and SVM models seem to have underperformed compared to the other models.
   2. 12 weeks rolling prediction:
      1. Mean Absolute Error (MAE): For all models, the MAE of the 12-week forecast is slightly lower than the 1-week forecast. However, the difference is very small. This indicates that, on average, the models have a slightly smaller absolute error after 12 weeks of forecasting compared to 1 week of forecasting.
      2. Mean Absolute Percentage Error (MAPE): Again, all models have a lower MAPE for the 12-week forecast. This indicates that the model has a smaller percentage error in predicting the next 12 weeks.
      3. Symmetric Mean Absolute Percentage Error (SMAPE): The regression and SVM models have a slightly lower SMAPE for the 12-week forecast, while the Ridge and Lasso models have a slightly higher SMAPE. This indicates that the symmetry of the prediction errors on a percentage basis improves for the regression and SVM but deteriorates for the Ridge and Lasso over the 12-week period.
      4. Mean Square Error (MSE): The MSE of the 1-week and 12-week forecasts are the same for all models, indicating that the mean squared error is the same in both cases.
      5. Root mean squared error (RMSE): The RMSE of the 12-week forecast is slightly lower for all models. This indicates that the models have slightly lower forecast errors when forecasting 12 weeks ahead compared to 1 week ahead.
      6. In summary, the models perform slightly better for the 12-week forecast compared to the 1-week forecast, with the exception of SMAPE. However, these differences are very small, indicating that the model's performance is quite stable across the forecast horizon.
   3. One week expanding rolling prediction:
      1. Mean Absolute Error (MAE): The 1 week expanded prediction shows that the MAE is slightly lower than the 1 week and 12 week predictions for the regression, and similar values for the stochastic, SVM, Ridge, and Lasso models. This indicates that the mean absolute error of the model is slightly smaller when using the expanded rolling window for the regression model.
      2. Mean Absolute Percentage Error (MAPE): The MAPE of the 1-week expanded prediction is lower than the 1-week and 12-week predictions for all models. This indicates that the percentage error of the model is smaller when using the extended rolling window.
      3. Symmetric Mean Absolute Percentage Error (SMAPE): SMAPE is slightly higher for the 1 week expanded prediction for the Ridge and Lasso models, but lower for the regression, stochastic, and SVM models. When using the expanded rolling window, the prediction error symmetry improves for the regression, random, and SVM models, but deteriorates for the Ridge and Lasso models.
      4. Mean Squared Error (MSE): MSE is slightly lower for the 1 week expanded prediction for the regression and SVM models, as well as for the random, Ridge, and Lasso models. This indicates that the mean squared error is slightly smaller or the same when using the extended rolling window.
      5. Root Mean Square Error (RMSE): The RMSE is lower for the 1 week expanded prediction for the regression and SVM models, and similar for the random, Ridge, and Lasso models. This indicates that the prediction errors of the models are smaller or the same when using the extended rolling window.
      6. In summary, prediction using the extended rolling window slightly improves the performance of some models (Reg and SVM) without significantly changing the performance of others. These improvements are more pronounced for MAPE.
   4. 12 weeks expanding rolling prediction:
      1. Mean Absolute Error (MAE): The MAE of all models in the 12-week expanding rolling prediction is similar to the MAE of the 1-week expanding prediction. This indicates that expanding the rolling prediction window from 1 week to 12 weeks does not significantly change the mean absolute error of the models.
      2. Mean Absolute Percentage Error (MAPE): For all models, the MAPE in the 12-week expanding rolling prediction is slightly higher than the 1-week expanding prediction, but still lower than the 1-week and 12-week predictions. This indicates that the percentage error of the model for the 12-week expanding rolling window increases slightly compared to the 1-week expanding rolling window, but is still lower than the unexpanded window.
      3. Symmetric Mean Absolute Percentage Error (SMAPE): The SMAPE for all models in the 12-week expanding rolling prediction is slightly lower than the 1-week expanding prediction. This indicates that, on a percentage basis, the prediction error symmetry is slightly improved when using the 12-week expanding rolling window compared to the 1-week expanding rolling window.
      4. Mean Square Error (MSE): The MSE for all models in the 12-week expanding rolling prediction is the same as the MSE for the 1-week expanding prediction. This indicates that expanding the rolling prediction window from 1 week to 12 weeks does not change the mean squared error of the models.
      5. Root mean squared error (RMSE): The RMSE of all models in the 12-week expanding rolling prediction is similar to the RMSE of the 1-week expanding prediction. This indicates that the prediction errors of the models are similar when using a 12-week expanding rolling window compared to using a 1-week expanding rolling window.
      6. In conclusion, extending the rolling forecast window from 1 week to 12 weeks does not significantly change the performance of the model. While MAPE increases slightly, SMAPE improves slightly, while MAE, MSE, and RMSE remain essentially unchanged. This may indicate that the time scale of the rolling window does not significantly affect the predictive accuracy of the model.
7. Conclusion:

In summary, Ridge and Lasso regression models outperformed SVM, Random Forest, and Linear Regression models in our experiments based on the error metrics. Lasso regression slightly outshone others, with Ridge regression demonstrating potential in specific scenarios. Performance of SVM, Random Forest, and Linear Regression varied considerably across different experimental conditions. Experiments with extended rolling forecast windows (1-week and 12-week) presented marginally better outcomes than their non-extended counterparts. Overall, Lasso regression proved the most efficient, closely followed by Ridge regression, and extending the rolling prediction window showed minor advantages in terms of predictive accuracy, which the model that have the best performance is “ 12 week expanding rolling prediction with Lasso model”.

1. Thoughts after conducting this experiment:

This study has substantially contributed to my proficiency in applying machine learning methodologies to financial forecasting. However, it has also highlighted the importance of complex approaches and meticulous tuning of hyperparameters for optimal outcomes. This reinforces the understanding that the quest for excellence in machine learning is a continual process, necessitating ongoing learning and refinement of techniques.

1. Graphs of Rolling Prediction results(2017-2022):

一張含有 文字, 螢幕擷取畫面, 圖表, 繪圖 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 圖表, 繪圖 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 圖表, 繪圖 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 圖表, 繪圖 的圖片

自動產生的描述

1. Graph of Prediction results(2022): Data ahead the red line is pure out of sample forecasts.

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述 一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述 一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述