The Hong Kong University of Science and Technology
School of Business and Management - Department of Economics

ECON 4305

Machine Learning for Economic and Financial Analysis

Predicting Stock Return with Fundamental Indicators

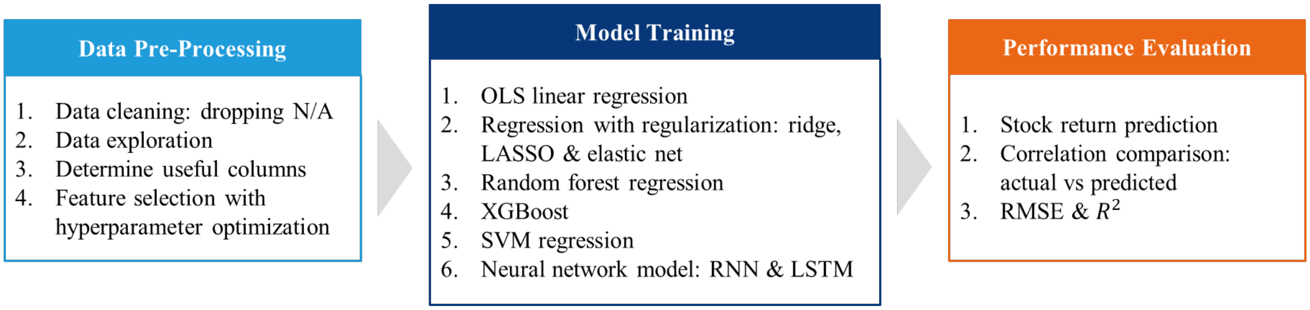
Group 09 Term Paper - 2023/24 Fall

AU, Yik Hau | CHEUNG, Tsz Hin | MAK, Pak Ho | TAN, Chen-yi

1. Introduction

The stock market has always drawn great interest for investors who are looking to make profitable investments. Despite the efforts put into predicting the stock market using traditional prediction methods, the accuracy of these predictions is not always as expected. However, with the advancements in technology, machine learning has emerged as a promising approach to predicting the stock market. In this paper, we aim to explore the effectiveness of different machine learning methods (OLS linear regression without regularization, linear regression with ridge/lasso/elastic net regularization, random forest regression, XGBoost regression, SVM regression, RNN and LSTM) in predicting the stock prices of the United States' market. Our analysis will focus on using stock fundamental indicators to train different regression models and neural networking models. By doing so, we hope to shed light on the potential of machine learning in predicting the stock market and help investors make more informed decisions.

2. Methodology


Data Pre-Processing
1. Data cleaning: dropping N/A
2. Data exploration
3. Determine useful columns
4. Feature selection with hyperparameter optimization
Model Training
1. OLS linear regression
2. Regression with regularization: ridge, LASSO & elastic net
3. Random forest regression
4. XGBoost
5. SVM regression
6. Neural network model: RNN & LSTM
Performance Evaluation
1. Stock return prediction
2. Correlation comparison: actual vs predicted
3. RMSE & R^2

<Figure 1. Methodology flow chart>

2.1 Data pre-processing

For the scope of this paper, we consider each stock return observation in the raw dataset *"US Stock Fundamentals Dataset"* not to be time-series data but but a unique and independent data point with column *"id_name"* used as the primary key which uniquely identifies each record, and we do not take into account company-specific or time-specific effects, regardless of the ticker and timestamp of that observation. Because stock return belongs to time series panel data, in order to exclude the stochastic trend that might lead to poor model forecasting results, we only include financial ratios[1] as our features (67 features in total) to minimize the non-stationary variations over time. Therefore, no stationary transformation such as differencing or detrending is performed in the data pre-processing step.

2.1.1 Data cleaning

Although the raw dataset contains 60000+ observations, many of which are incomplete with missing values. As a result, we first use the command *"dropna()"* to drop all observations with missing data and obtain 11711 valid observations as our cleaned data, which is later split based on the 70-30 rule for the training (70%) and testing (30%) dataset.

2.1.2 Feature selection with hyperparameter optimization

Two feature selection methods, Principal Component Analysis (PCA) and SelectKBest, are employed to reduce the dimension of our cleaned data and prevent overfitting problems. Before directly imposing these two feature selection algorithms onto our training dataset with default parameter settings, we used the Grid search approach to optimize hyperparameters used by PCA (number of components) and SelectKbest (number of features, k).

a. Principal Component Analysis (PCA)

   During the Grid search process, we defined the config scope to be within the range from 1 to 50 and used the Root-Mean-Square Error (RMSE) as the evaluation scores for different numbers of components. Based on the Grid search result for the OLS regression model, RMSE is minimized with a value of 42.5128 when the number of components is equal to 1. The algorithm yields the same result of *"optimal_n_pca_comp = 1"* for all our other linear regression models trained with regularization terms (ridge, lasso, and elastic net regression), which is just a 1-D dimension linear projection. In fact, the top 5 performing number of components are all less than or equal to 5. Despite that we included 67 features in the dataset initially, the Grid search result indicates a lack of representation of most of our features.

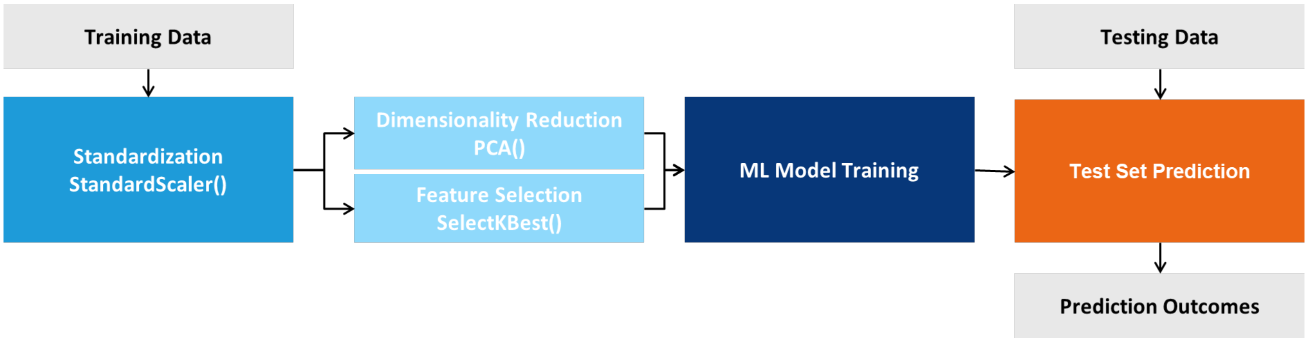
[1] Featuers before "column AT" are dropped to avoid introducing non-stationary properties into the model.

b. SelectKBest

Since we can’t directly interpret the dimension reduction result from PCA, we also employed the SelectKBest method to pin down the feature-specific effect. Unlike the parameter subset we set for PCA, the total config is set to be 67, which is the total number of features included in our cleaned dataset, with RMSE is used for scoring. The Grid search result indicates that the number of features that minimize RMSE is when $k = 3$, with the features selected being *“roa_gp”*, *“turn”*, as well as *“F_lever_chg”* and a RMSE of 43.9266.

2.2 Model training

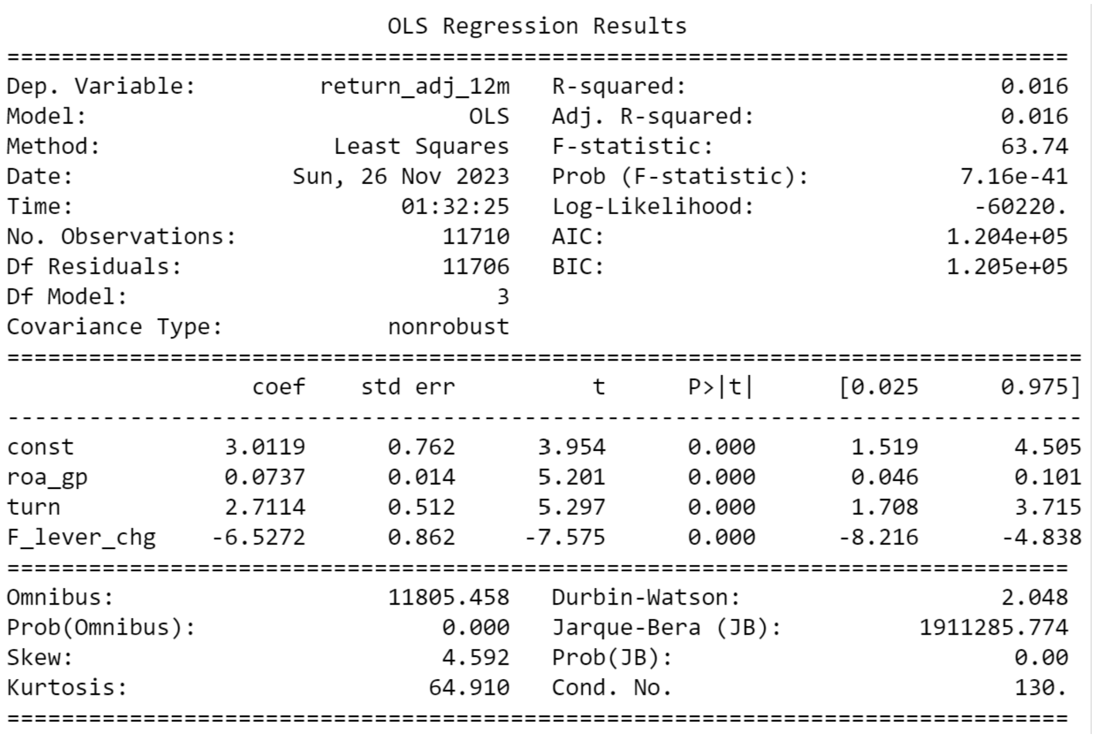
Nine machine learning models are trained by feeding 70% of the data from the cleaned dataset (11711 observations) to our predefined pipeline for each model which generally includes standardizing the data, reducing dimensionality with PCA/selecting common features with SelectKBest, training the machine learning algorithm, and lastly predicting the result using test set data.



<Figure 2. Generalized machine learning pipeline for our 9 models>.

2.3.1 Ordinary Least Squares (OLS) linear regression

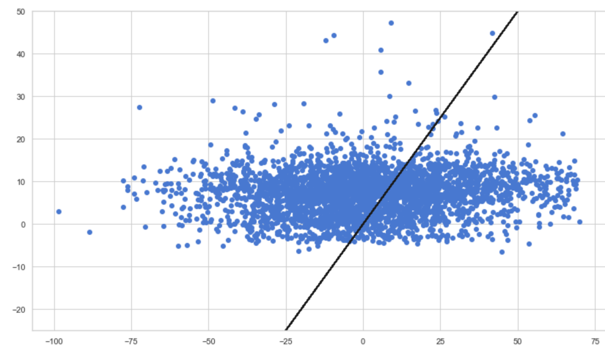
In terms of the correlation between predicted stock return and actual return, the model with features selected by SelectKBest (12.80% correlation) performs better than the one-dimensional PCA model (4.46% correlation), although both are still extremely low. For better result interpretation, we mainly focus on the SelectKBest model for further analysis:

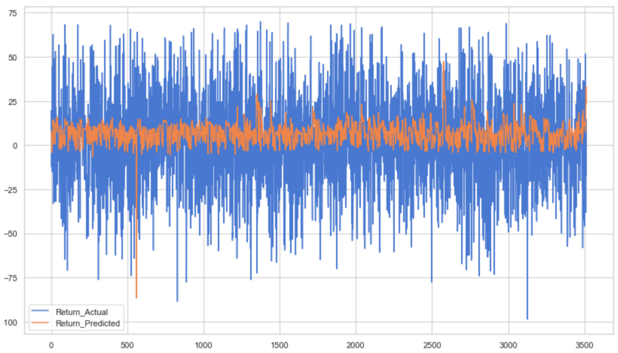


```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:         return_adj_12m   R-squared:                       0.016
Model:                            OLS   Adj. R-squared:                  0.016
Method:                 Least Squares   F-statistic:                     63.74
Date:                Sun, 26 Nov 2023   Prob (F-statistic):           7.16e-41
Time:                        01:32:25   Log-Likelihood:                -60220.
No. Observations:               11710   AIC:                         1.204e+05
Df Residuals:                   11706   BIC:                         1.205e+05
Df Model:                           3                                         
Covariance Type:            nonrobust                                         
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
const           3.0119      0.762      3.954      0.000       1.519       4.505
roa_gp          0.0737      0.014      5.201      0.000       0.046       0.101
turn            2.7114      0.512      5.297      0.000       1.708       3.715
F_lever_chg    -6.5272      0.862     -7.575      0.000      -8.216      -4.838
==============================================================================
Omnibus:                    11805.458   Durbin-Watson:                   2.048
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1911285.774
Skew:                           4.592   Prob(JB):                         0.00
Kurtosis:                      64.910   Cond. No.                         130.
==============================================================================
```

<Figure 3. OLS linear regression ANOVA table with variables chosen by SelectKBest>

As we can observe from the ANOVA table, our model has both low R^2 and low p-values at the same time. On one hand, the F-statistics for the model and t-statistics for all variables selected are all larger than 1.96, which represents statistical significance under a 95% confidence level, proving our model to be useful indeed in capturing the relationships between the predictors and target variable. On the other hand, the low R^2 value also indicates that our model explains the variation of stock return poorly and isn't able to make precise return predictions. As clearly illustrated in the two figures below, while the statistically significant variables are able to estimate the trend among the noisy and variable data, the variation explained by our model is way too low for the highly volatile stock return.

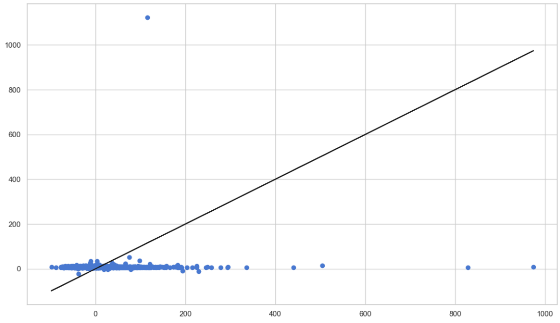


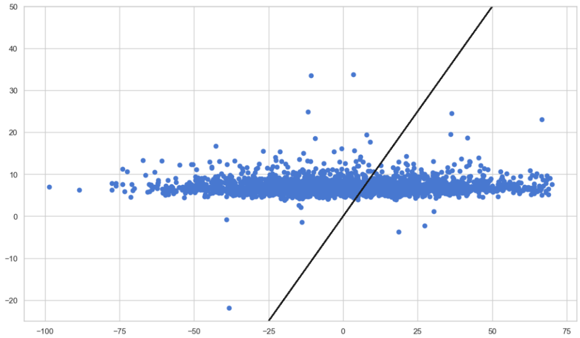


<Figure 4. Actual v.s predicted stock return scatter plot>   *<Figure 5. Actual v.s predicted stock return line chart>*

2.3.2 Random forest regression

Our spectaculation on why OLS Linear Regression failed to predict the result is because the model is too simple to capture the complexities of stock return. Thus, we decided to use Random Forest Regression to better capture the complexity of stock return. However, the poor result obtained using correlation as the standard in the random forest regression is observed. The low correlation level of 4.50% is an indication that the model is not performing as expected. Additionally, the presence of outliers in the figure shown is a major problem that can lead to bias in the results obtained. Although the outliers were removed to address this issue, the improvement in the result was only marginal, and the overall performance of the model remains unsatisfactory. It is not surprising that the result was not performing as expected as regression methods are hard to predict in a market with high volatility. It is clear that more work needs to be done to improve the performance of the random forest regression model, and alternative methods may need to be explored to achieve better results.

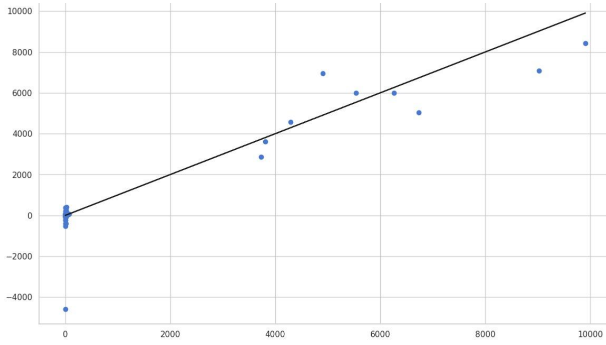




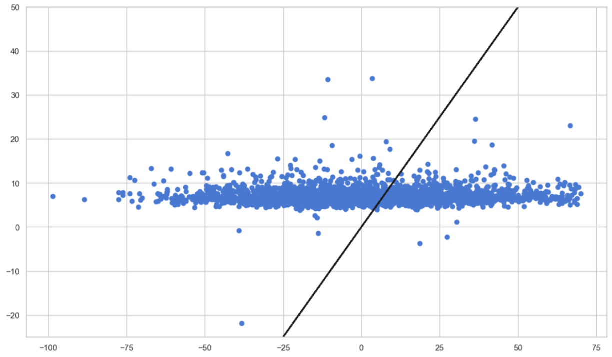
<Figure 6. Actual v.s predicted stock return scatter plot>   *<Figure 7. Figure 6 after removing outliers>*

2.3.3 Extreme Gradient Boosting (XGBoost)

We suspect that existence of model randomness in Random Forest Model may emphasise on the unrelated parameter thus result in low predictive power, therefore we decided to implement a more deterministic model - XGBoost. Nevertheless, the XGBoost model's performance in predicting the utilization metric was underwhelming. With a low correlation of only 4.50% when using correlation as the evaluation metric, the model clearly underperformed compared to expectations. This is further validated by the high RMSE of 44.73 achieved by the model. The presence of outliers in the data also posed a significant issue, as they could introduce bias in the results. Though attempts were made to address this by removing outliers, the improvement to the model was marginal at best, and the overall performance remained unsatisfactory. Given the inherent volatility of the market being predicted, it is unsurprising that regression methods like XGBoost struggle to make accurate predictions in this context. Clearly, more work is needed to strengthen this XGBoost model's predictive abilities. Alternative machine learning approaches should also be explored to attain results superior to an RMSE of 44.73 and aim for a fuller utilization of the available data.



<Figure 8. Actual v.s predicted stock return scatter plot>

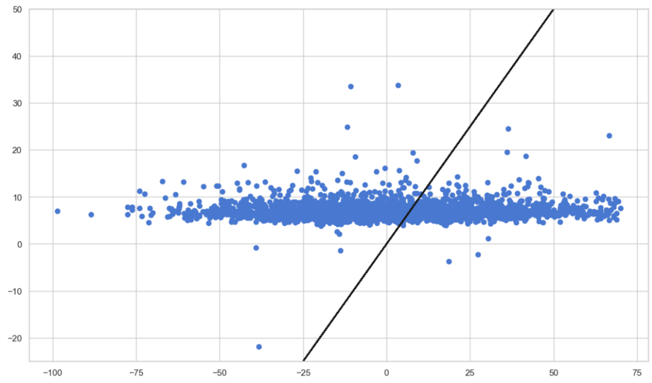


<Figure 9. Figure 8 after removing outliers>

2.3.4 Support Vector Machines (SVM) regression

Since both models with randomness and deterministic based on decision tree failed to achieve the prediction with significant power, we try to implement the SVM Regression Model to attempt to capture the complexity of stock return movement better. We set the kernel function equal to “RBF” which is theoretically creating a hyperplane in infinite dimension (exp() can be expanded in an infinite series using Taylor's series, RBF uses this technique to mimic a polynomial kernel function with dimensions number n tends to infinity) to capture the large amount of features. (Thurnhofer-Hemsi, 2020)

Nevertheless, the SVM regression method has yielded another unsatisfactory result in our study. Despite removing outliers, the correlation with the actual return is still quite low, at only 3.39%. During the hyperparameter grid search, we observed that the number of principal component axes was only 1. This is a significant limitation, as it means that the model was unable to fairly include all the fluctuations of components in the market. As a result, the model's performance suffered, and the accuracy of its predictions was compromised. These findings highlight the need to carefully consider the underlying assumptions and limitations of the model. Further research is needed to identify alternative approaches that can overcome these limitations and improve the accuracy of the model.

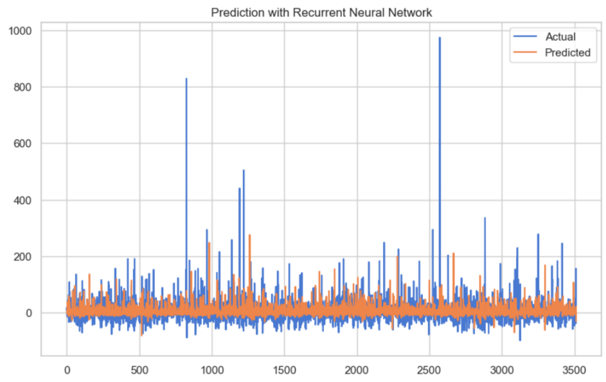


<Figure 10. SVM regression after removing outliers>

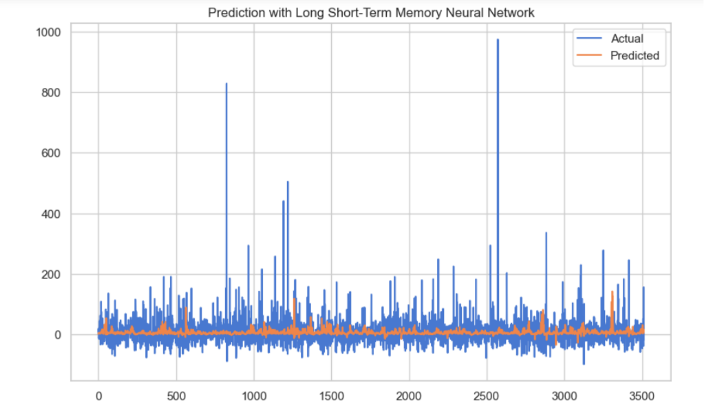
2.3.5 Neural Network model: Recurrent Neural Networks (RNN) and Long-Short Term Memory (LSTM) Neural Networks

Given the fact that all linear regression models trained above yield undesirable results, we also try to implement 2 neural networks models, which are recurrent neural networks and long-short term memory neural networks model, to fit potential non-linear patterns that are more complex. In the initial stage of RNN model training, we encountered the problem of gradient explosion. To address this issue, we transformed the data by batch normalization before running the RNN model. With this modification, the model converged, but the results obtained were not satisfactory, with a root mean square error (RMSE) value of 48.3231. Despite our efforts to repeat with different activation functions (e.g., ReLU and tanh), the outcome does not improve. This outcome suggests that there may be other factors affecting the performance of the model, and the problem of outliers still occurs and needs further examination. Yet, one worth noting point is the fact that among all models trained, RNN is already the model that suffers the least underfitting issue and captures the most volatility of stock return as illustrated in *Figure 11*.

In contrast, we also attempted to use the Long Short-Term Memory (LSTM) Neural Network model, but the results obtained were similarly unsatisfactory. Specifically, the RMSE value for this model was 44.9098, slightly lower as compared to RNN model. Interestingly, while the RNN model tended to overestimate the predicted value, the LSTM neural network model tended to underestimate it. Despite these differences, both models exhibited a high RMSE value of exceeding 40. As a result, neither model can be considered reliable for predicting the stock price of the market. These findings underscore the challenges associated with accurately predicting stock return.



<Figure 11. RNN model prediction>



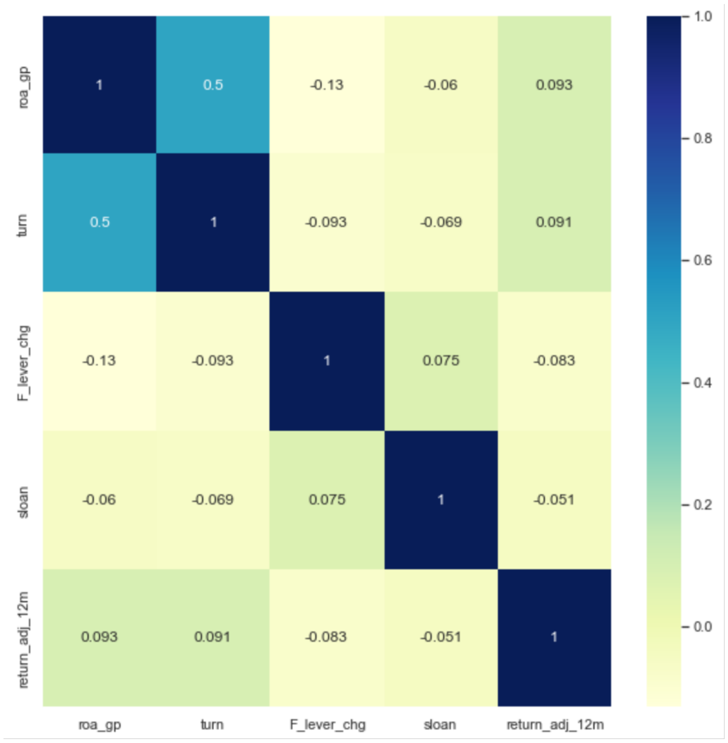
<Figure 12. LSTM neural network model prediction>

3. Discussion

3.1 Limitations

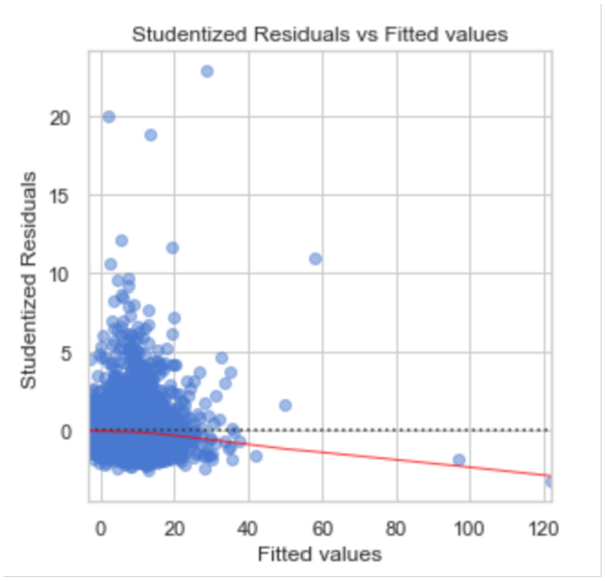
3.1.1 Low correlation

Given the fact that both PCA and SelecKBest methods only select a small number of common components/features to include in the regression model, we find it important to examine the overall correlation among all 67 features and stock return using the correlation heat map. Our investigation using the correlation heatmap supported the selection result of PCA and SelectKBest as there were no significant correlations between any features and the stock return. In fact, there are only four features that have a correlation greater than 5% with the dependent variable, and three of which are identified by the SelectKBest algorithm. This finding also points out that there exist some fundamental predictiveness issues within our model due to the limited representativeness of our data features.



<Figure 13. Correlation heatmap with absolute correlation with "return_adj_12m" ≥ 0.05>

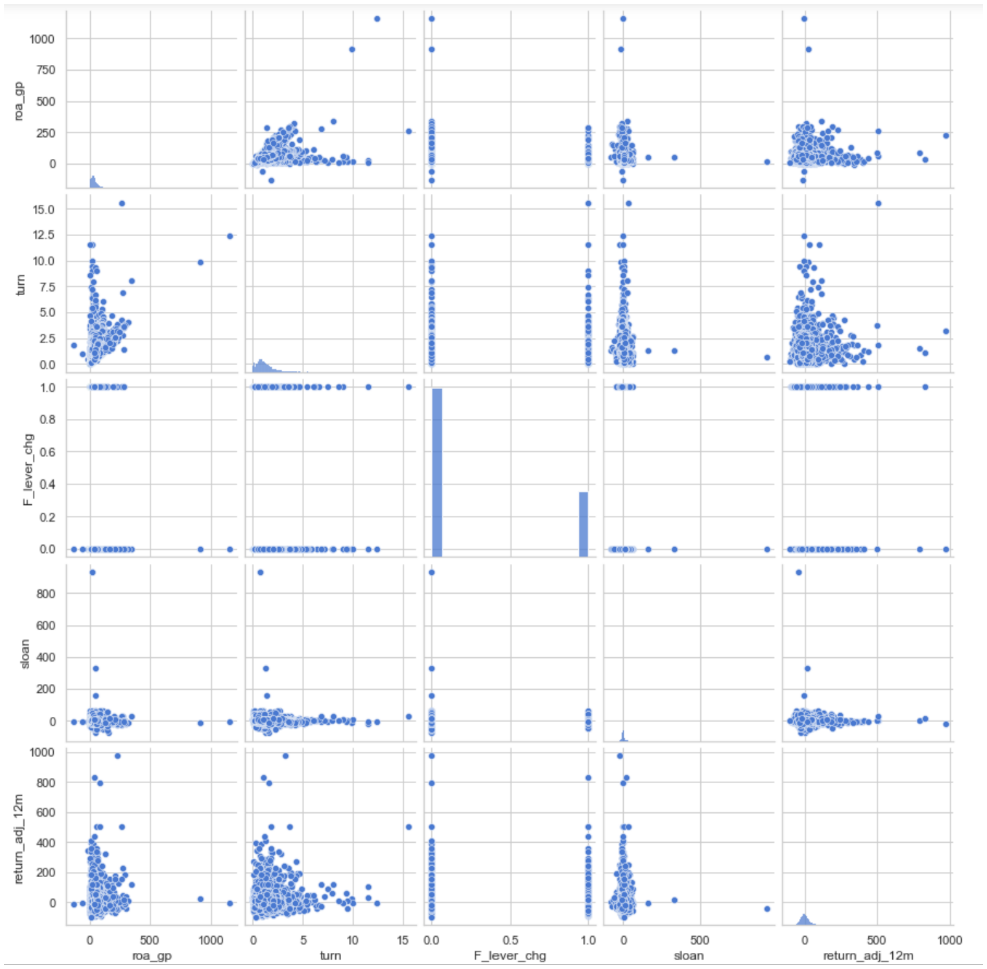
3.1.2 Non-linearity and data dispersion



<Figure 14. Predicted stock return and standardized residual>

In our data pre-processing step, only ratio-type features are included in the model to avoid introducing stochasticity into our models. Nevertheless, because our dependent variable stock return itself is time series data, even for financial ratio variables, we still identify non-stationary characteristics from the data. Take the scatter matrix of the OLS linear regression model with SelectKBest features, for example, we can observe that many data points of the selected features (all of which have *a* ≥ 0.05 absolute correlation with stock return) are clustering around the bottom left corner of the scatter plot (*Figure 15.*), without a clear pattern of their relationship with the target variable. This indicates that data transformation such as taking the first difference of logarithms is required to linearize the curvature among variables and better fit the non-linear patterns. Similarly, still referring to the same OLS model as an example, from its residual plot (*Figure 14.*) we can see that the residuals are far from evenly distributed, and our model might suffer from heteroscedastic trends. Therefore, based on both the data preliminary and residual analysis, we consider it necessary to perform data transformation for better generalization of our models. Nonetheless, after performing non-linear transformation[2] for our features within the raw dataset and dropping data points with N/A or inf values, there are only 10 valid observations left after transformation, making the data sample size too small for ML model training. Due to the limitation within our dispersed dataset, although we recognize that data transformation is necessary, we fail to obtain a useful dataset after transformation.

[2] Please refer to the coding file *"Data Transformation"* for more details.



<Figure 15. Scatter Matrix between features with absolute correlation ≥ 0.05 with stock return>

Another limitation of our current study is that since we treated the panel dataset, which contains time series data, as cross-sectional, ignoring the temporal dimension, our models cannot fully leverage the temporal dependencies and trends within the data, limiting their predictive power. Moving forward, we could adopt time series methods to model the data more precisely. For instance, we could start by conducting an Augmented Dickey-Fuller (ADF) test on the time series components. The ADF test would help determine if the data exhibits stationarity. If it is non-stationary, we may need to apply transformations like differencing to render it stationary, then re-run the ADF test to validate the results. Adopting this approach would enable us to better account for the time-based relationships in the data and potentially strengthen the predictive capabilities of our models.

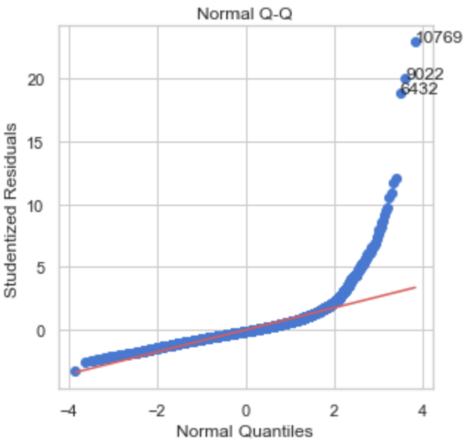
3.1.3 Exclusion of qualitative data

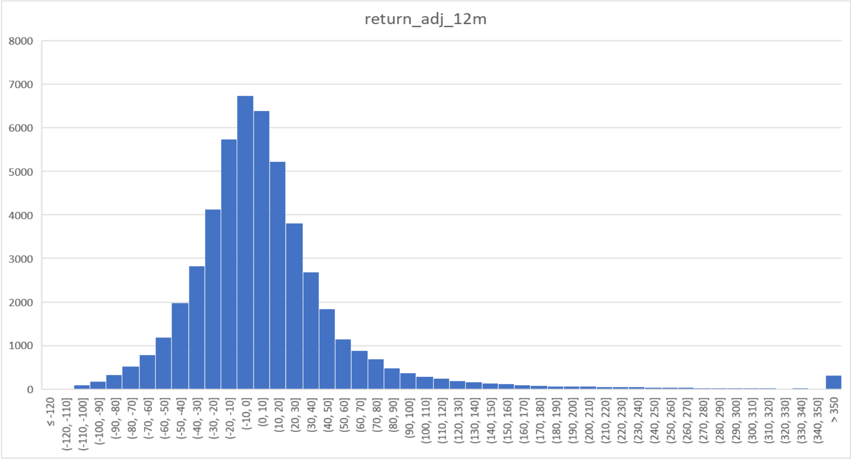
Since all the above machine learning models trained are only capable of processing numerical data, therefore, we did not include categorical data such as time-specific features or differences in regional factors among different companies during the data pre-processing process. It is worth noting that not only numerical features of stock prices can help forecast future prices, but time and environmental factors can also have a significant impact on the change of stock prices. The exclusion of these important factors from the data handling process can lead to various confounding factors that contribute to bias in the results. It is clear that more work needs to be done to improve the accuracy of the model by including these critical factors in the analysis.

3.2 Area of improvements

3.2.1 Outlier detection

Outliers can disproportionately influence the model's behavior and reduce the accuracy of machine learning outcomes, especially in algorithms that are sensitive to extreme values. For example, in linear regression, outliers with large residuals can significantly alter the estimated coefficients and affect the line of best fit. Examining both the residual normal Q-Q plot and distribution histogram of the stock return, we can observe that the target variable itself is highly right-skewed and there exist large positive outliers that could potentially reduce the predictive power of our models.

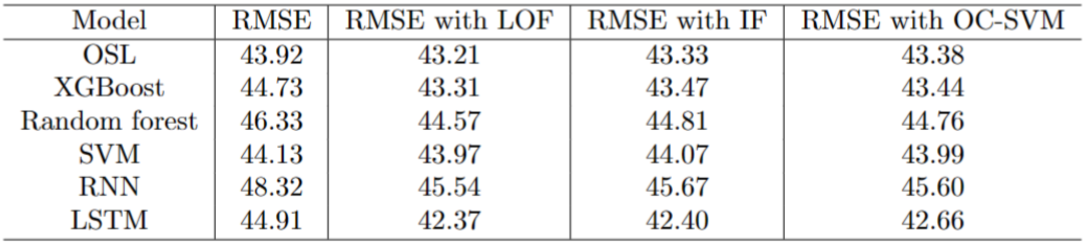




<Figure 16. Residual normal Q-Q plot of

OLS regression> *<Figure 17. Stock return distribution histogram>*

To mitigate the impact of outliers, three outlier detection models are trained, which are local outlier factor, isolation forest, and one-class SVM. Below is the performance summary for our model performance after outlier detection. As clearly stated in the table, not much significant improvement is observed.



Model	RMSE	RMSE with LOF	RMSE with IF	RMSE with OC-SVM
OSL	43.92	43.21	43.33	43.38
XGBoost	44.73	43.31	43.47	43.44
Random forest	46.33	44.57	44.81	44.76
SVM	44.13	43.97	44.07	43.99
RNN	48.32	45.54	45.67	45.60
LSTM	44.91	42.37	42.40	42.66

<Figure 18. Regression model performance summary with outlier detection>

3.2.2 Neural networks model optimization: RNN and LSTM

Due to the constrained computational capacity of our devices, we didn’t train multiple RNN and LSTM models by adjusting the default code arguments to find the optimal configuration. Nevertheless, it’s definitely worth exploring other possible hyperparameters, activation functions, and optimizer combinations to further enhance the model's performance. For example, we may try to modify the number of hidden layers, the number of neurons per layer, and the dropout rate to find a balanced model complexity and overfitting regularization. In addition, experimenting with other activation functions and optimizers such as *tanh* or *sigmoid* activation functions as well as *adamax* optimizer can potentially yield better results as tested in other research. (Qiao et al., 2022)

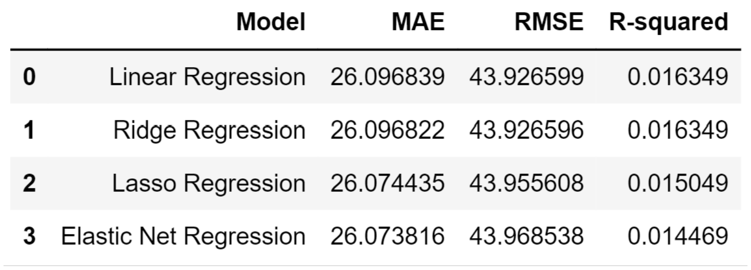
3.2.3 Market efficiency and observation frequency

In Ou and Penman's paper, they have a relatively robust conclusion that since the market doesn't fully impound future earnings implications of current accounting disclosures in current stock prices, the fundamental analysis identifies equity values not currently reflected in stock prices and thus predicts 1-year-ahead "abnormal" returns. (Ou & Penman, 1989) Yet, as far as our term paper finding is concerned, the predictiveness of fundamental indicators on 12-month lagged excessive return is clearly not the case for more recent time frames. With the advancement of information technology, it's fairly reasonable for us to say that the market efficiency has greatly improved and it's highly likely that changes in companies' financials are swiftly incorporated into their stock prices. In fact, there is even a similar study forecasting 1-month-ahead stock returns but still ended up with merely 5% weak correlation between actual and predicted returns. (Abe & Nakayama, 2018) Therefore, using stock return of higher frequency and obtained in a closer timeframe with other financial predictors (eg. 1-day-ahead, 1-week-ahead, etc) may be a more suitable target variable than the 12-month-ahead stock return.

3.3 Alternative models

3.3.1 Linear regression with regularization: ridge, lasso, and elastic net regression

All three regularized regression models yield similar results with the OLS linear regression mode without any significant improvements and fail to explain the response variability well. While RMSE is more or less the same across those models, the R^2 values are even lower than the original OLS model. These results thereby highlight the underfitting issues within our original OLS model and adding regularization terms into the model only makes the underfitting situation worse.



	Model	MAE	RMSE	R-squared
0	Linear Regression	26.096839	43.926599	0.016349
1	Ridge Regression	26.096822	43.926596	0.016349
2	Lasso Regression	26.074435	43.955608	0.015049
3	Elastic Net Regression	26.073816	43.968538	0.014469

<Figure 19. Regularized regression model performance summary>

3.3.2 Multilayer Perceptron (MLP) Neural Network

Unlike RNNs and LSTMs, which have specialized architectures for processing sequential data, MLPs treat each input as independent and disregard any temporal or sequential relationships. This can be a significant limitation when dealing with time series data like stock returns, where the order of data points is crucial. In addition, MLPs are not designed to model long-term dependencies. They may struggle to capture complex temporal patterns or relationships that unfold over a long time horizon. Stock returns often exhibit long-term trends or cyclic patterns, and MLPs may have difficulty capturing these dynamics effectively. Given these weaknesses, it may be worth exploring alternative modeling approaches, such as more advanced RNN architectures (e.g., GRUs or LSTMs with attention mechanisms) or other specialized models for time series forecasting, such as ARIMA, to address the underfitting issue and better capture the temporal dynamics in stock returns.

3.3.3 ARIMA model

Autoregressive Integrated Moving Average (ARIMA) is a statistical analysis model commonly used in professional quantitative analysis on stock market. It's a way of modeling time series data for forecasting time series data in such a way that a pattern of growth/decline, the rate of change in the growth/decline, and noise between consecutive time points in the data are all accounted for. (Abugaber, n.d.) However, the fundamental assumption of the ARIMA model is not suitable for the topic of our analysis. The model assumes that the time series data is stable and its mean and variance remain constant throughout time, which is not the situation in the highly volatile stock market. In addition, the ARIMA model is extremely sensitive to outliers because error terms are one of the predictor variables in the model and are given weights. Once again, the high volatility in stock markets and the positively-skewed stock return distribution will lead to unstable results of the ARIMA model.

4. Conclusion

In sum, the predictiveness of fundamental financial indicators of a company in predicting the stock return 12 months later is far from satisfactory, therefore, it is not recommended to use machine learning to make such forecast and formulate trading strategies of any kind according to the model prediction results. To obtain better generalization outcomes, it's also of great importance that data of higher quality is used in the model. We believe that with a less dispersed dataset and more sophisticated data pre-processing (non-linearity transformation, multicollinearity mitigation, etc), the model performance can be improved. Despite the low correlations with actual returns among all models, we identified that neuron networks models, especially the LSTM model, outperform the traditional linear regression alternatives in terms of RMSE. This could be because neural networks models are more effective in capturing nonlinear relationships and learning complex patterns in the data while generally being more robust to outliers and noisy data compared to linear regression. More neural networks models with different combinations of activation functions and optimizers are definitely worth exploring.

5. References

Abugaber, D. (n.d.). *Chapter 23: Using ARIMA for Time series analysis*. https://ademos.people.uic.edu/Chapter23.html

Qiao, R., Chen, W., & Qiao, Y. (2022). Prediction of stock return by LSTM neural network. *Applied Artificial Intelligence*, *36*(1). https://doi.org/10.1080/08839514.2022.2151159

Rana, M., Uddin, M. M., & Hoque, M. M. (2019). Effects of Activation Functions and Optimizers on Stock Price Prediction using LSTM Recurrent Networks. *CSAI2019: 2019*

3rd International Conference on Computer Science and Artificial Intelligence. https://doi.org/10.1145/3374587.3374622

Ou, J. A., & Penman, S. H. (1989). Financial statement analysis and the prediction of stock returns. *Journal of Accounting and Economics*, *11*(4), 295–329. https://doi.org/10.1016/0165-4101(89)90017-7

Abe, M., & Nakayama, H. (2018). Deep learning for forecasting stock returns in the Cross-Section. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1801.01777

Thurnhofer-Hemsi, K. (2020, July 16). *Radial basis function kernel optimization for Support Vector Machine classifiers*. arXiv.org. https://arxiv.org/abs/2007.08233