SVM vs MLP for stock prediction

INM427 Neural Computing

RAJANI MOHAN JANIPALLI

Rajani.Janipalli@city.ac.uk

1. Brief description and motivation of the problem

This paper aims to present a comparative study of stock prediction to predict whether or not a stock would perform well in future, a binary classification task, by comparing predictions with Support Vector Machines and Multi-Layer Perceptron. The paper tries to find which of the two algorithms gives a better prediction for the given kind of data. The study would be useful to investors of quant hedge funds through such predictive models.

2. Description of the dataset

The dataset is cleaned, regularized and obfuscated global equity data taken from Kaggle [1]. Originally this is a standard financial market data of global equities assembled by Numerai [2], who offer this data to data scientists around the world for free so that they can have hedge fund quality data to build their machine learning models. It then builds a quant hedge fund from thousands of those crowdsourced models. For confidentiality and simultaneously retention of unique features, the data is obfuscated.

Each instance in the dataset corresponds to a stock in a particular era [3]. The features are various quantitative attributes of the stock at that time. The target indicates an abstract measure of stock performance in the future.

The dataset has 96,320 rows or instances and 22 columns, of which the first 21 are features or independent variables and the last one is the target or the dependent variable. All the features are continuous numeric values between 0 and 1. The target is a binary column of 0s and 1s. There are neither any missing values nor any outliers as shown in Fig.1. As shown in the figure, the quartiles and the median of all the features are also in close range.

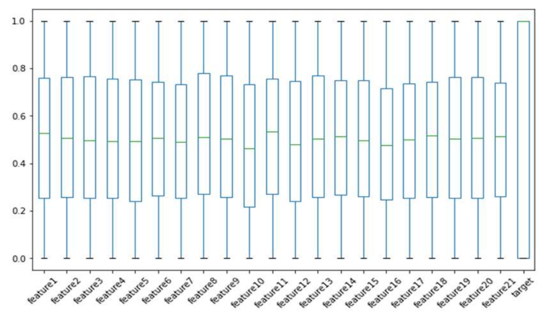


Figure 1: Boxplot of all columns.

A series of scatter plots of all the 21 features have shown that the features are uniformly distributed. The data is almost balanced with respect to the target class, having 48,658 target values as 1s and 47,662 of them as 0s. So, the exploratory data analysis showed that the data is indeed cleaned and regularized. The descriptive statistics of the data indicates that the mean for all the 21 features is in range 0.45 – 0.55 and the standard deviation is in the range 0.28 – 0.30.

3. Brief summary of the two neural network models

Support vector machines (SVM)

It is a supervised machine learning algorithm established on statistical learning theory [4], used for classification and regression tasks [5]. SVM uses the training data to maximize the existing separation between the decision borders in a high dimensional space called feature space, to determine the decision functions. In high dimensional space, the decision boundary is called a hyperplane. The extreme points on either side of the hyperplane called the support vectors [6] are key to the algorithm. For a given set of weights and bias, the separation between the hyperplane and the support vectors is called the margin of separation. The aim of SVM is to find a hyperplane for which the margin of separation is maximized, under which condition the hyperplane would be referred to as the optimal hyperplane.

Pros of SVM:

It can easily handle high dimensionality of the data [7], resistant to over fitting, has very good generalization abilities, and has shown promising empirical performance.

Cons of SVM:

Due to algorithmic complexity, for large datasets the training process is very slow [5]. For multi class classification, development of optimal solution is not an easy task. Correct classification of minority class in an imbalanced data is tough problem.

Multi-layer perceptron (MLP)

It is a feed forward neural network made of layers of neurons that are densely connected [8]. One or more layers are contained in the network that are hidden from both the input and output nodes. A non-linear activation function that is differentiable is included in the model of each neuron in the network [6]. One of the well know learning methods of MLP is the back-propagation algorithm, consisting of two phases. In the first phase called the forward phase, the input signal is propagated through the network, layer by layer, until it reaches the output, with fixed weights and changing activation potentials. In the second phase called the backward phase, the output of the network is compared with a desired response to produce an error signal, that is propagated through the network, again layer by layer, but in the backward direction and making successive adjustments to the weights of the network. These two phases are repeated in cycle till the point where there is no further update of the weights of the network.

Pros of MLP:

Easy to work with non-linear data [9], produces good results for both small and large datasets. The training process is fast.

Cons of MLP:

In some cases, it is computationally expensive. Training influences the model functioning.

4. Hypothesis statement

As a literature review for the study of this paper, two papers which compared the same two algorithms for stock prediction but on a different kind of data, were referred to. The primary paper of reference is "A comparison between SVM and Multilayer Perceptron in predicting an emerging financial market: Colombian stock market" by Bustos et al. and others [10]. The secondary paper of reference is "Comparison between SVM and MLP in Predicting Stock Index Trends" by Mantri [11]. Both the papers concluded that SVM was a better predictor than MLP for the data in their studies.

Going with the above-mentioned conclusions and the common notion that SVM performs better than MLP, the Hypothesis for this study is that SVM will perform better than MLP for the prediction task of this paper.

5. Description of choice of training and evaluation methodology

Training

To perform this study a slight deviation was made from the primary reference paper and 80% of the data was used for training. Following the primary reference paper, 5-fold cross-validation was performed on baseline models and penultimate improvement before the best model, for obtaining generalization of the models over the training data. Then, the mean cross-validation score of all the 5 splits of the data was considered to measure the performance of the model. Also, like the primary reference paper, the hyperparameters were tuned through an exhaustive grid search, which performs a 5-fold cross-validation by default. After tuning all the possible hyperparameters, the best model with all the best parameters from the grid search was created for both algorithms.

Evaluation

The test data was exposed only to the best models of both algorithms. The best models of the two algorithms were compared and evaluated in different terms using different metrics. To check for any overfitting the training score, as well as the testing score of both the algorithms, were compared. To understand the computational efficiency, the training time for both algorithms was compared. To see how correctly the classification of both the target classes was done, other metrics viz., Receiver Operating Characteristic (ROC) curve [12], Area Under Curve (AUC), confusion matrix [13] and classification report were for the two algorithms were compared.

6. Choice of parameters and experimental results

SVM

The baseline model was trained without the assignment of any specific parameters. Then, the regularization parameter C, the type of kernel and the kernel coefficient gamma were tuned step by step through grid search. It was observed that the model scores for grid search improved a tiny bit for the best values of gamma and C.

MLP

The baseline model was trained without the assignment of any specific parameters. Then a number of parameters, viz., hidden layer sizes, activation function, weight optimization solver, L2 penalty parameter alpha, the batch size for stochastic optimization, learning rate type, learning rate initial value, the maximum number of iterations, momentum and early stopping were tuned step by step through grid search. It was observed that the model score for grid search improved a tiny bit for the best values of hidden layer size, activation function and alpha.

7. Analysis and critical evaluation of results

Before proceeding to modelling, a correlation matrix of all the features was plotted, which revealed that for some pairs of features there was a correlation as high as 0.86. To deal with this problem of multicollinearity, the L2 penalty [14] parameters were to be tuned appropriately while modelling.

SVM

As a start, a baseline model was created without assigning any parameters to the SVM classifier object. Then, a 5-fold cross-validation of the baseline model was performed over the training data and the mean score of the 5 splits was 0.5192.

Taking inspiration from a paper on “An efficient machine learning approach for indoor localization” by Zhang et al. [15], a grid search was done for hyperparameter tuning on principal components , to reduce the computation time for SVM, as a result of dimension reduction. So, a principal component analysis was performed to obtain components that explained more than 70% variance, which had a mean cross-validation score of 0.5188 for a 5-fold cross-validation, quite close to that of the baseline model for original data.

The first hyperparameter to be tuned was the kernel, a grid search of which directed to ‘rbf’, the Radial Basis Function kernel. Theoretically, since the data is nonlinear and there is no prior knowledge of the data, ‘rbf’ is the kernel suitable for the case [5].

The next hyperparameter to be tuned is the kernel coefficient gamma, a grid search of which concluded that 0.001 is the best value for it. A very high value of gamma would lead to the overfitting of the model. This took the mean cross-validation score of the model to 0.5190.

The final parameter to be tuned is the regularization parameter C, for which a grid search was made to find the best C among 0.5, 1.0 (default) and 1.5 to get a sense of the direction in which the regularization increases. At 0.5 gave the best mean test score and also a better regularization. This pushed the mean cross-validation score of the model further to 0.5195.

Then, an SVM classifier object was created with all these best parameters, but it was fit to the original training data of 21 features, instead of the six principal components, and the mean cross-validation score achieved was 0.51944, which is an improvement of 0.02998 % over the original baseline SVM model. So, the best SVM model was made from these parameters and trained over the training data, which yielded results as shown in the Fig.3.

MLP

Similar to SVM a baseline model for MLP was also created, that gave a mean score of 0.5148 for 5-fold cross validation over training data, not using PCA for MLP anywhere.

The first hyperparameter to be tuned was the number of hidden layers and the nodes in them. A grid search was performed on different number of nodes for a single hidden layer and different sets of combinations of number of nodes for two hidden layers. The best mean score was obtained for a single hidden layer of 16 neurons, which would be resistant to overfitting. This took the mean cross-validation score of the model to 0.5224.

Coming the activation function, for the binary classification into 0 and 1, three activation functions viz., logistic, tanh and relu are equally good contenders, theoretically [16]. But a grid search on all of them proved tanh to be consistently the best in terms of mean test score. This gave a tiny kick to the mean cross-validation score of the model to 0.5228.

For weight optimization, SGD and Adam which is an extension of SGD are two natural choices for the large dataset in the study, with a trade-off between generalization and performance, respectively [17]. A grid search gave mean test scores of 0.5175 and 0.5228 for SGD and Adam respectively, which are not greatly different in terms of generalization and hence Adam was chosen as the best parameter for the edge of performance.

Another step towards generalization of the model is the L2 penalty parameter alpha, for which a grid search was performed on values 0.0001 (default), 0.001 and 0.01, resulting in mean test scores of 0.5228, 0.5229 and 0.5225 respectively. In terms of generalization, these scores aren’t too different. But in terms of performance, 0.001 was the best pushed the mean cross-validation score further.

Then, the batch size for stochastic optimization was played with different values and it turned out that one of the default values, 200, produced the best mean score.

Learning rate type is an important factor that would determine that stability of model and the chance to reach global minima while optimizing. Strangely, the mean scores for all the type in the grid search were exactly same. So, constant type, which was indicate as the best by grid search, was indeed considered the best. Similarly, the default value 0.001 was considered the best parameter for the initial learning rate.

After that, a grid search on the maximum number of iterations was done over the values in a wide range and the default number 200 outperformed the rest. So, to avoid computational cost after the convergence of stochastic gradient descent, this was chosen as the best.

A gird search was performed on momentum parameter which aids the optimization by guiding it in the direction of minima, was performed over a range of values. Surprisingly, the mean test score for all the values was exactly same and hence the default values 0.9 was chosen as the best parameter.

The final parameter tuned was the early criteria, which was set to True in view of generalization, as in case of False the mean test score isn't greatly high.

Then, a MLP classifier object was created with all these best parameters and the mean cross-validation score achieved was 0.5227, which is an improvement of 1.5454 % over the original baseline MLP model. So, the best MLP model was made from these parameters and trained over the training data, which returned results as shown in the Fig.3.

Results

Looking at the performance metrics Fig.2. and Fig.3, its is clear that MLP outperformed SVM in terms of both training and test scores. The difference in training time proves MLP to be computationally extremely efficient than SVM.

For this study, the target value of 1 is considered a positive class and 0 as a negative class, with 1 meaning stock would perform well and 0 meaning stock wouldn't perform well. So, proper classification of 0 would be a priority as it would mean avoiding loss to the investor. The true negatives are high and false positives are low for MLP, making it better model than SVM. This is also supported by the higher recall of negative class for MLP. Strangely, the precision of both classes is same for both the algorithms. F1 scores are little dicey, but it doesn't contradict any of the observations above.

8. <u>Conclusions, lessons learned, references and future work</u>

The study concluded that refuting the hypothesis, MLP outperformed SVM in terms of multiple metrics. Although the baseline model of SVM performed better than MLP, the best model of MLP turned better than SVM, which can be attributed for the huge number of hyperparameters available for tuning in case of MLP.

Major lesson learnt was that depending on the data the advantage of hyperparameter optimization can have an edge over empirical performance history.

Future work would include playing more with hidden layer configurations for MLP and making a similar study for multi class classification for similar kind of data.

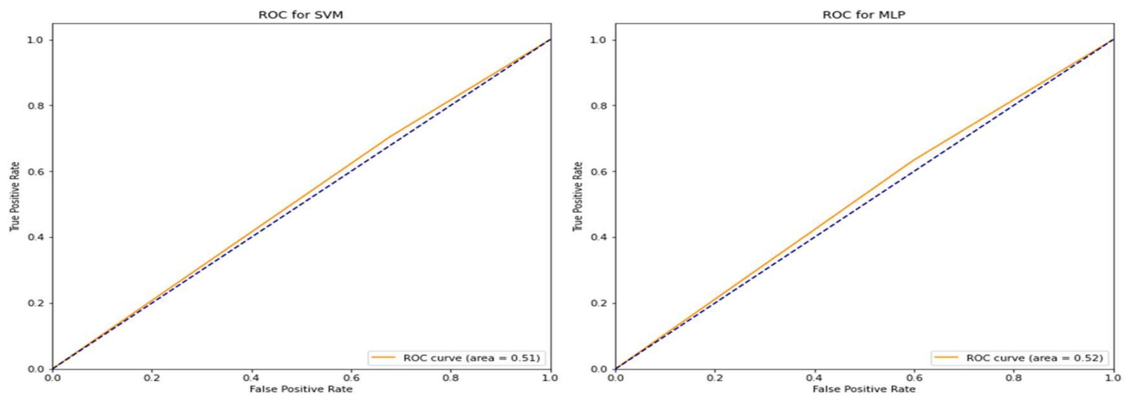
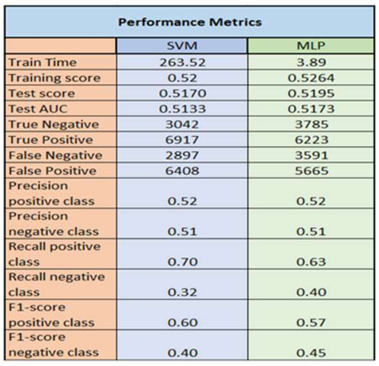


Figure 2: ROC curves of best models of SVM & MLP for test data.



Performance Metrics		
	SVM	MLP
Train Time	263.52	3.89
Training score	0.52	0.5264
Test score	0.5170	0.5195
Test AUC	0.5133	0.5173
True Negative	3042	3785
True Positive	6917	6223
False Negative	2897	3591
False Positive	6408	5665
Precision positive class	0.52	0.52
Precision negative class	0.51	0.51
Recall positive class	0.70	0.63
Recall negative class	0.32	0.40
F1-score positive class	0.60	0.57
F1-score negative class	0.40	0.45

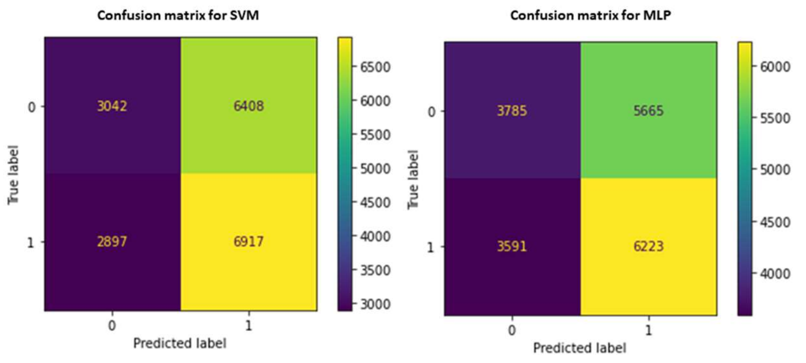


Figure 3: Performance metrics for best models of SVM & MLP.

References

[1] "Encrypted Stock Market Data from Numerai." https://www.kaggle.com/numerai/encrypted-stock-market-data-from-numerai (accessed May 05, 2022).

[2] "Numerai Hedge Fund." https://numerai.fund/ (accessed May 05, 2022).

[3] "Numerai Tournament Overview." https://docs.numer.ai/tournament/learn (accessed May 05, 2022).

[4] N. Tripathy, "Stock Price Prediction Using Support Vector Machine Approach," presented at the International Academic Conference on Management and Economics, Nov. 2019. doi: 10.33422/conferenceme.2019.11.641.

[5] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, Sep. 2020, doi: 10.1016/j.neucom.2019.10.118.

[6] S. S. Haykin and S. S. Haykin, *Neural networks and learning machines*, 3rd ed. New York: Prentice Hall, 2009.

[7] Y. Tian, Y. Shi, and X. Liu, "Recent advances on support vector machines research," *Technol. Econ. Dev. Econ.*, vol. 18, Mar. 2012, doi: 10.3846/20294913.2012.661205.

[8] M. P. Naeini, P. Branch, H. Taremian, and H. B. Hashemi, "Stock Market Value Prediction Using Neural Networks."

[9] B. Akkaya and N. Çolakoğlu, "Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases," Sep. 2019.

[10] O. Bustos, A. Pomares, and E. Gonzalez, "A comparison between SVM and multilayer perceptron in predicting an emerging financial market: Colombian stock market," in *2017 Congreso Internacional de Innovacion y Tendencias en Ingenieria (CONIITI)*, Oct. 2017, pp. 1–6. doi: 10.1109/CONIITI.2017.8273335.

[11] J. K. Mantri, "Stock Index Trends."

[12] J. Brownlee, "How to Use ROC Curves and Precision-Recall Curves for Classification in Python," *Machine Learning Mastery*, Aug. 30, 2018. https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/ (accessed May 07, 2022).

[13] "AUC-ROC Curve in Machine Learning Clearly Explained," *Analytics Vidhya*, Jun. 15, 2020. https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/ (accessed May 07, 2022).

[14] "L2 vs L1 Regularization in Machine Learning | Ridge and Lasso Regularization." https://www.analyticssteps.com/blogs/l2-and-l1-regularization-machine-learning (accessed May 07, 2022).

[15] L. Zhang, Y. Li, Y. Gu, and W. Yang, "An efficient machine learning approach for indoor localization," *China Commun.*, vol. 14, no. 11, pp. 141–150, Nov. 2017, doi: 10.1109/CC.2017.8233657.

[16] "12 Types of Neural Networks Activation Functions: How to Choose?" https://www.v7labs.com/blog/neural-networks-activation-functions, https://www.v7labs.com/blog/neural-networks-activation-functions (accessed May 08, 2022).

[17] Synced, "ICLR 2019 | 'Fast as Adam & Good as SGD' — New Optimizer Has Both," *SyncedReview*, Mar. 07, 2019. https://medium.com/syncedreview/iclr-2019-fast-as-adam-good-as-sgd-new-optimizer-has-both-78e37e8f9a34 (accessed May 08, 2022).

APPENDIX

INM427 Neural Computing | RAJANI MOHAN JANIPALLI | City University of London

Glossary

Features/ independent variable	The variables which will be used to produce the outcome of the model prediction.
Target column/response variable	The dependent variable which will be the outcome of the model prediction.
Instance	Row of table contained data
Supervised machine learning algorithms	Algorithms in which labelled datasets are used to train the model, for it to classify the data or predict the output. The model would know the target it has to achieve.
Decision boundary	A boundary line created by the classifier to segregate the data into different classes.
Weights	A parameter indicating the amount of influence an input will on the output.
Bias	A constant that is added to the product of an input and its weight, before the application of activation function.
Overfitting	This is a condition where a model tries to fit all the data during the training, but is unable to predict the relationship among the data precisely.
Imbalanced data	Data in which the distribution of target values is uneven.
Feed forward neural network	Neural network in which information or signal propagates from input to output through functions.
Non-linear activation function	Functions inside neural networks that modify the data to make it suitable for non-linear predictions.
Gradient descent	It is an optimization method used to minimizer function by following the gradients of a cost function plot. Cost function is a measure of how bad the model that uses the estimation of losses of the model during the training phase.
Cross Validation	A method to estimate how well a machine learning model can perform in general for any data.
Hyperparameters	It is a constituent of model configuration which helps the model in learning process and can be used to tune or optimize the model.
Exhaustive grid search	An approach in which hyperparameter tuning is done for a grid of parameter values.
Kernel	A function that converts the data into a high dimensional feature space.
SGD – Stochastic gradient descent	An approach in which randomly selected data are used instead of whole dataset while trying the minimize the loss through derivation.
L2 penalty	Penalty applied on model when there are some features in data that are highly correlated.
ROC curve	Receiver Operating Characteristic curve is a plot of false positive rate against true positive rate of prediction of classes.
AUC	Area Under Curve represents the ability of the model to precisely predict the labelled classes.
Confusion matrix	A matrix indicating the classification quality in terms of True positives, False positives, True negatives and False negatives.
True positives	It is the value of the positive class data of a classification problem that have been predicted as positive and are actually positive.
True negatives	It is the value of the negative class data of a classification problem that have been predicted as negative and are actually negative.

False positive	It is the value of the negative class data of a classification problem that have been predicted as positive but are actually negative.
False negative	It is the value of positive class data of a classification problem that have been predicted as negative but are actually positive.
Recall	It is the fraction or proportion of positive class of a classification problem that have been correctly identified as positive.
Precision	It is the fraction of positive class that is identified correctly out of all the positives including false positives.
F1- score	It is the harmonic mean of precision and recall

Implementation details

SVM

On training the baseline model, it was observed that the process was very slow and time consuming. So, based on a reference paper mentioned in the report, it was decided to do hyperparameter tuning on principal components to take the computational advantage of reduced dimensions.

While doing PCA, initially it was done for explaining 95% variance which was shown by 13 components. It was observed that out 13, 6 components were explaining a variance of more than 70%. So, a second PCA was done with to obtain those 6 components and that was then used in grid search for tuning hyperparameters.

MLP

The baseline model of MLP trained so fast that there was no need to use principal components in grid search for hyperparameter tuning.