JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Stock Price Prediction: Machine Learning Models with K-fold Cross Validation

[1]Joseph R, [2]Pramod Kumar Dang
[1 2]Students MTech in Artificial Intelligence
[1]Reva Academy for Corporate Excellence (RACE),
[1]REVA University, Bengaluru, India

Abstract: Stock exchange price prediction is one of the most researched topics, attracting interest from both academics and industry. Various algorithms have been developed since the introduction of Artificial Intelligence (AI) and have been used to forecast equities market movement. Experienced quantitative traders understand that it's often straightforward to create a strategy with excellent predictive capabilities during a back-test. However, such back-tests may obscure the risk of an overfit model, potentially resulting in significant underperformance when the strategy is implemented. less attention has been paid to the use of cross validation (CV) approaches for better stock price prediction. In this article we will attempt to find a partial remedy to the problem of an overfit machine learning model using a technique known as cross-validation**.**

IndexTerms **– Stock Price, Machine Learning, feature engineering, Cross Validation.**

I. INTRODUCTION

The stock market is a regulated marketplace where investors can buy or sell stocks publicly or privately. Companies often turn to the stock market to raise capital for business expansion, making it a popular investment option for investors. To make informed investment decisions, many investors rely on predictions based on past market trends. In recent years, the stock market has changed significantly, making it crucial to anticipate its future value or price given how dynamic the market is. Predictive analysis is frequently carried out using linear regression, a mathematical technique and supervised machine learning method. The linear regression model, which generates linear correlations between independent and dependent variables, is largely consistent with the continuous/real values of mathematical variables. The algorithm makes the predictions in accordance with the guidance provided in the training data set after first training with it.

II. OVERVIEW OF CROSS VALIDATION

The goal of cross-validation is to estimate the test error associated with a statistical model or select the appropriate level of flexibility for a particular statistical method. The training error associated with a model can vastly underestimate the test error of the model. Cross-validation provides us with the capability to more accurately estimate the test error, which we will never know in practice. Cross-validation works by holding out particular subsets of the training set in order to use them as test observations. In this article we will discuss the various ways in which such subsets are held out as well as implement the methods using Python on an example forecasting model based on prior historical data.

k-Fold Cross Validation:

K-fold cross-validation improves upon the validation set approach by dividing the n observations into k mutually exclusive, and approximately equally sized, subsets known as "folds". The first fold becomes a validation set, while the remaining k-1 folds (aggregated together) become the training set. The model is fit on the training set and its test error is estimated on the validation set. This procedure is repeated k times, with each repetition holding out a fold as the validation set, while the remaining k-1 are used for training.

This allows an overall test estimate, CV_k, to be calculated that is an average of all the individual mean-squared errors, MSE_i, for each fold:

$$\mathrm{CV}_k = \frac{1}{k}\sum_{i=1}^{k} \mathrm{MSE}_i$$

The obvious question that arises at this stage is what value do we choose for k? The short answer (based on empirical studies) is to choose k=5 or k=10. The longer answer to this question relates to both computational expense and, once again, the bias-variance trade-off.

III. Research Method

In this work, we consider the prediction performance of machine learning models under cross validation approach, namely K-fold CV. The models considered here are simple linear regression model, Ridge, Lasso and Elastic net Machine learning model. The models were trained on Tata Motors NSE stock data from October 2020 to October 2021, using an 80% training set and a 20% test set. Standard strategic indicators such as, mean squared error, root mean square error, mean absolute error and R2 score is used to evaluate the models.

Mean squared error (MSE):

The MSE measures the amount of error in statistical models. It assesses the average squared difference between the observed and predicted values. When a model has no error, the MSE equals zero. As model error increases, its value increases. The mean squared error is also known as the mean squared deviation (MSD).

$$MSE = \frac{\sum (y_i - y_p)^2}{n}$$

y_i = actual value
y_p = predicted value
n = number of observations/rows

Root Mean Squared Error (RMSE):

RMSE measures the difference of values. These values are the one which are predicted by the model and the actual values. Whatever deviation is measured by RMSE is called residuals. This method is also called root mean square deviation (RMSD). In order to aggregate the magnitude of the errors, RMSD is used to serve the purpose of prediction.

$$RMSE = \sqrt{\frac{\sum (y_i - y_p)^2}{n}}$$

y_i = actual value
y_p = predicted value
n = number of observations/rows

Mean Absolute Error (MAE):

MAE is referred to as the difference between 2 continuous variables. it is used to measure accuracy for continuous variables. When we have a set of predictions, it is used to measure the average magnitude of errors.

$$MAE = \frac{|(y_i - y_p)|}{n}$$

y_i = actual value
y_p = predicted value
n = number of observations/rows

R-Squared:

(R^2 or the coefficient of determination) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, r-squared shows how well the data fit the regression model (the goodness of fit).

$$R^2 = 1 - \frac{\sum_i^N (y_i - y_p)^2}{\sum_i^N (y_i - y_p)^2}$$

y_i = actual value
y_p = predicted value
n = number of observations/rows

IV. Feature Engineering:

Creating lag features (also known as lagged features) is a common technique in time series analysis and stock price prediction. Here's how lag features is created:

1. Lag Window is Defined: Decide on the number of lag features that is created. In this case, lag1 to lag5 features are created.
2. Shift the Time Series: Shifted the stock price time series by the lag window to create the lagged features. Each lag feature will represent the stock price from the previous day(s).

3. Merge Lag Features: Merged the lag features with the original dataset to create a new dataset with lagged features.

The Below code snippet is creating lagged features for a dataframe df based on the 'Close' column.

```
# Define the number of lags
num_lags = 5

# Create lagged features
for i in range(1, num_lags + 1):
df[f'Close_Lag_{i}'] = df['Close'].shift(i)
```

Table 1.1 displays the output of the resulting dataframe after feature engineering

Date	Open	High	Low	Close	Close_Lag_1	Close_Lag_2	Close_Lag_3	Close_Lag_4	Close_Lag_5
16-10-2020	55.5	56.15	54.55	55.849998	54.200001	55.650002	58.299999	59.450001	61.450001
19-10-2020	56.55	57.15	55.45	55.900002	55.849998	54.200001	55.650002	58.299999	59.450001
20-10-2020	55.4	55.7	54.6	55.299999	55.900002	55.849998	54.200001	55.650002	58.299999
21-10-2020	55.65	56.6	54.65	55.549999	55.299999	55.900002	55.849998	54.200001	55.650002
22-10-2020	55.35	57.1	55	56.650002	55.549999	55.299999	55.900002	55.849998	54.200001

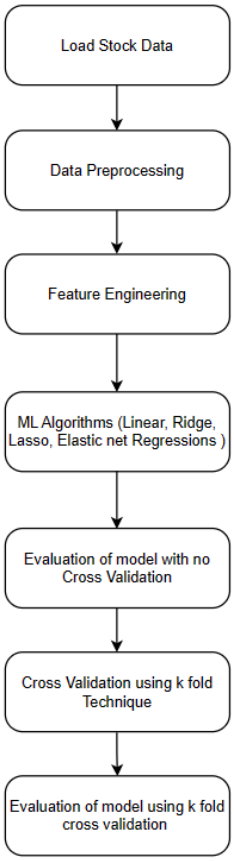
The Below code snippet combines X_test and y_test into a dataframe fdf, then adds columns for predicted values ('predicted') and prediction errors ('prediction_error') based on a model's predictions (y_pred). Finally, it prints the resulting dataframe fdf as shown in resulting table.

```
fdf = pd.concat([X_test,y_test], 1)

fdf['predicted'] = np.round(y_pred,1)
fdf['prediction_error'] = fdf['Close'] - fdf['predicted']

print(fdf)
```

Table 1.2 illustrates an example of the actual closing prices compared to the predicted values generated by the modeling process.

Close_Lag_1	Close_Lag_2	Close_Lag_3	Close_Lag_4	Close_Lag_5	Close	predicted	prediction_error
131.149994	131.449997	128.050003	129.050003	128.649994	132.8	131.6	1.200003
55.299999	56.650002	56.25	55.5	57.450001	55.55	55.6	-0.050001
134.699997	131.199997	132.800003	135.850006	137.75	138.6	135.5	3.100006
129.949997	132.399994	134.25	138.25	133.699997	127.8	130.7	-2.899997
55.849998	54.200001	55.650002	58.299999	59.450001	55.9	56.7	-0.799998
155.100006	156.350006	159.600006	156.449997	154.949997	155.7	155	0.699997
55.549999	55.299999	55.900002	55.849998	54.200001	56.65	56.2	0.450002
138.699997	139.649994	140.25	140.399994	139.449997	138.55	139	-0.449997

V. FLOW CHART



VI. RESULTS

Table 2.1: Model Scores with no Cross Validation

Model	No Cross Validation			
	MSE	RMSE	MAE	R2
Linear	15.2117932	3.900229891	2.399182923	0.988494854
Ridge	15.21340169	3.900436089	2.399216573	0.988493638
Lasso	15.604719	3.950280876	2.419193588	0.988197673
Elastic net	15.68560677	3.960505873	2.41571962	0.988136495

Table 2.2: Model Scores with Cross Validation

Model	K-Fold Cross Validation with k=5			
	MSE	RMSE	MAE	R2
Linear	15.70145694	3.880552388	2.908591651	0.84552767
Ridge	15.7019025	3.880686117	2.908772579	0.84552627
Lasso	15.67918128	3.894231461	2.930333227	0.844702996
Elastic net	15.81026553	3.909902543	2.949863451	0.843715813

VII. CONCLUSION:

In this study, we employed linear regression, Ridge, Lasso, and Elastic Net models to predict stock prices. Cross-validation was utilized to assess the models' generalization performance by testing them on various data subsets. Models that were not cross-validated may have overfit to the training set, showing higher performance on that set but potentially poorer performance on unseen data. This trend is evident in the results presented in Tables 2.1 & 2.2.

This study is significant because it offers researchers suitable models and techniques for predicting stock prices, helping them develop optimal plans for the future. Future research could explore additional important variables for measuring stock prices and consider other cross-validation approaches to improve stock price modelling.

REFERENCES

[1] Ali, A. 2001.Macroeconomic variables as common pervasive risk factors and the empirical content of the Arbitrage Pricing Theory. Journal of Empirical finance, 5(3): 221–240.

[1] Song Y, Lee J. Design of stock price prediction model with various configurations of input features. Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing, Sanya, China, 19-21 December 2019. DOI: 10.1145/3371425.3371432.

[2] Misra M, Yadav AP, Kaur H. Stock market prediction using machine learning algorithms: a classification study. 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering. IEEE, Odisha, India, 27-28 July 2018. DOI: 10.1109/ ICRIEECE44171.2018.9009178.

[3] Jeevan B, Naresh E, Kambli P. Share price prediction using machine learning technique. 2018 3rd International Conference on Circuits, Control, Communication and Computing. IEEE, Bangalore, India, 3-5 October 2018. DOI: 10.1109/ICRIEECE44171.2018.9009178.

[4] Sharma V, Khemnar R, Kumari R et al. Time series with sentiment analysis for stock price prediction. 2019 2nd International Conference on Intelligent Communication and Computational Techniques. IEEE, Jaipur, India, 28-29 September 2019. DOI: 10.1109/ICCT46177.2019.8969060.

[5] Pahwa K, Agarwal N. Stock market analysis using supervised machine learning. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing. IEEE, Faridabad, India, 14-16 February 2019. DOI: 10.1109/ COMITCon.2019.8862225**.**