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 DATA FEATURE SELECTION, STANDARDISATION AND FINDING THE BEST TRAIN-TEST COMBINATION FOR STOCK PRICE PREDICTION USING LINEAR REGRESSION

[1] **Paul Vettukallel** & [2] **Ashwini Madhusudan**
[1 2] Students MSc in Artificial Intelligence
[1]REVA Academy for Corporate Excellence (RACE)
[1]REVA University, Bengaluru, India

Abstract : Stock price prediction is one of the challenging problems in the field of Machine Learning as stock markets remain extremely fluid due to a varying number of external factors which are fleeting in nature. The performance of the Machine Learning (ML) Models largely depends on the features selected to make predictions. Identification & selection of features and the most appropriate ML Model plays a critical role in prediction accuracy and Model versatility across different stocks and stock markets. This study explores Basic Features as well as Technical Indicators for selection of features for stock price prediction based on market best practices and seeks to identify the best Train-Test-Scaling combination for modeling the time-series OHLCAV (Open-High-Low-Close-AdjClose-Volume) data that achieves optimum test-metric scores for Linear Regression Models. For this study we used the historical stock prices data from yahoo finance and the prediction process is carried out using Linear Regression, Polynomial Regression and ElasticNetCV Models of ML from the Scikit-Learn library of Python. Our findings reveal a reliable process to identify the best performing Train-Test-Scaler combination for Linear Regression models, for stock price prediction among various other insights.

Index Terms **– Machine Learning, Stock Price Prediction, Stock Market Prediction, Linear Regression, Polynomial Regression, Elastic Net, Feature Selection, Standardization, Scaling, Train-Test-Scaler Combination**

I. INTRODUCTION

The movement of any stock price is governed by two broad factors, mainly the Fundamentals and the Technicals. The Fundamentals are well suited for long-term investors whereas, the Technicals are more suited for a short-term investor or a day-trader which may be used with no particular attention to the Fundamentals. Among the different assets, equities are the most interesting with respect to the prediction of short or long-term market prices, returns, and portfolio management.

1.1 Feature Selection

The selection of appropriate features, or independent variables, is a critical step in model construction. The relevance and quality of these features have a direct bearing on the efficacy of the model's learning process. Htet Htet Htun et al (2023) observed that structured-type inputs are mainly used as features in various stock market applications, because their information is systematic and the processing techniques are well-defined. Three main types of structured inputs are used in stock market prediction, namely Basic features, Technical indicators, and Fundamental indicators:-

(a) Basic Features are stock values such as OHLCAV data. Closing prices are the most commonly used information to predict the prices of the next trading day.

(b) Technical Indicators are extracted from historical price series using mathematical formulae and are used to analyze particular patterns of past prices and predict future movements.

(c) Fundamental Indicators are economic indicators (Bustos and Pomares-Quimbaya 2020) ranging from macroeconomic factors, such as a country's or government's overall economic status, to microeconomic factors, such as the information of an individual company. Macroeconomic factors, such as interest rates, consumer price index, and the overall state of the economy, are the most commonly used fundamental indicators. Forecasting based on fundamental indicators is less common in the literature because of the difficulty in building models that explain why a stock's price fluctuates.

In terms of the outputs from ML models, the two targets for prediction are value/ return and the direction of the stock. Predicting value/return is a regression task while predicting direction (up or down) is a classification task (Htet Htet Htun et al 2023).

1.2 Regression Analysis

Regression analysis is a set of statistical processes which investigates the relationship between dependent (target) variable and independent variables (predictors). This technique is used for forecasting, time-series modeling, and finding the causal-effect relationship between the variables. The most common form of regression analysis is Linear Regression, which aims to find the line that most closely fits the data according to a specific mathematical criterion. Linear regression is one of the foundational tools in data science. It is extremely powerful due to its simplicity, interpretability, and applicability to a wide variety of problems.

There are various kinds of regression techniques available to make predictions. These techniques are mostly driven by three metrics i.e. the number of independent variables, type of dependent variable, and shape of regression line:-

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Ridge Regression
- Lasso Regression
- ElasticNet Regression
- Logistic Regression

1.3 Simple Linear Regression

The case of one independent variable is called simple linear regression, while for more than one independent variable, the method is known as multiple linear regression. In simple linear regression, the relationship is summarized with a straight line. In a simple linear regression model, we predict the dependent variable y as a linear function of the independent variable x. If we have one independent variable, the equation of the line is: $y=mx+c$

Where: y is the dependent variable (output)
m is the slope of the line (also known as the coefficient or parameter)
x is the independent variable (input)
c is the y-intercept

1.4 Multiple Linear Regression

This equation can take many forms, for Multiple Linear Regression in ML, we usually write this equation as: $yi=\beta 0+\beta ixi$

Where: yi is the i^{th} dependent variable (output)
$\beta 0$ is the y-intercept/constant term
βi is the slope of the line with respect to the i^{th} independent variable
xi is the i^{th} independent variable (input)

We use Scikit-Learn's LinearRegression module to automatically implement Multiple Linear Regression in this paper.

1.5 Polynomial Regression

Polynomial regression is a special case of multiple linear regression. Instead of fitting a straight line or a plane in a high-dimensional space, polynomial regression fits a curved line or a surface in the case of multiple variables. In polynomial regression, we transform our features into polynomial features before applying linear regression. This way, we're still using a linear model, the coefficients attached to our features are still linear, but we've transformed our features themselves to capture non-linear relationships. This approach gives the model the flexibility it needs to fit the data more accurately, while still retaining the computational benefits of a linear regression model.

We use Scikit-Learn's PolynomialFeatures function to automatically implement a polynomial combination of features to the 2^{nd} degree. The PolynomialFeatures function with degree=2 creates all possible interaction terms and squares of the features. Specifically, it will generate the following feature combinations, for a set of six features a, b, c, d, e & f the PolynomialFeatures function generates a total of 28 feature combinations :-

[1,a,b,c,d,e,f,a2,b2,c2,d2,e2,f2,ab,ac,ad,ae,af,bc,bd,be,bf,cd,ce,cf,de,df,ef]

This new feature space allows the model to capture the relationship between the interaction of the various variables.

1.6 Evaluation Parameters

For assessing the quality of our model's predictions, we need to examine some key performance metrics. The commonly used metrics for regression tasks are Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and the Coefficient of Determination, often called the R-squared (R^2) score.

With these metrics, we can assess the performance of our model. However, they may not always provide a full picture of the model's strengths and weaknesses. The ultimate goal of machine learning isn't just about achieving the lowest error or highest accuracy, interpretability and understanding of the model are also important.

1.7 Over-fitting vs Under-fitting

It is important to strike a balance between model complexity and predictive power. We should avoid overfitting, which happens when the model is too complex and starts to learn the noise in the data, while underfitting results from a model that is too simple to capture the underlying relationships in the data. To find the optimal complexity, we could utilize techniques such as cross-validation or regularization.

1.8 Regularization

Regularization is a technique used in machine learning models to prevent overfitting, which occurs when a model captures not only the underlying trend but also the noise in the dataset. Regularization discourages overly complex models by adding a penalty term to the loss function that the algorithm optimizes. This penalty term reduces the values of the model coefficients, which in turn simplifies the model. There are several types of regularization techniques, three commonly used methods are Ridge, Lasso, and Elastic Net.

1.9 Elastic Net Regression (L1 and L2 Regularization)

Elastic Net regression is a middle ground between Ridge Regression and Lasso Regression. It is a hybrid model that includes both L1 and L2 regularization. It can shrink some coefficients and set some to 0 for sparse selection. Elastic Net is useful when there are multiple features which are correlated with one another. Lasso might just select one of these randomly, while Ridge would shrink all of them, but keep all of them in the model. In this case, Elastic Net will pick up some of the variables and leave out the others, leading to a potentially more meaningful model.

We use ElasticNetCV module from Scikit-Learn library of Python for the purpose of the study carried out in this paper.

1.10 Feature Standardisation using Scaling

Feature scaling improves the convergence of steepest descent algorithms, which do not posses the property of scale invariance. If features are on different scales, certain weights may update faster than others since the feature values play a role in the weight updates. The critical benefit of feature scaling is related to gradient descent. Scaling the features so that their respective ranges are uniform is important in comparing measurements that have different units. It allows us directly to compare model coefficients to each other.

To study the effect of standardisation using different scalers with different ML models, following Scalers from Scikit Learn library of Python are used to standardise the data for the purpose of this study:-

(a) MinMax Scaler.
(b) Standard Scaler.
(c) Robust Scaler.

II. Exploring Insights & Analysis of Researches so far

2.1 Feature Selection

A combined analysis of technical and fundamental indicators was conducted in Nti et al. (2020a), Thakkar and Chaudhari (2021) by using various artificial intelligence algorithms. These theories are challenged by the widely accepted random walk hypothesis (Fama 1995) and efficient market hypothesis (Malkiel 2003), which suggest that future changes in stock prices cannot be predicted from the historical data as fluctuations are independent and random. Therefore, future stock price changes are widely known to be unpredictable. However, many financial economists, researchers, and traders believe that stock prices are at least partially predictable because price changes tend to repeat themselves owing to the collective and patterned behaviour of investors (Zhang et al. 2018).

The most common technical indicators (Alsubaie et al. 2019) are the Relative Strength Index (RSI), stochastic oscillator, and moving average convergence-divergence. Some studies such as Botunac et al. (2020) and Qolipour et al. (2021), used a combination of basic features and technical indicators to forecast stock market direction.

The diversity of features presents a challenge in achieving higher prediction accuracy. Thus, a feature selection process should be performed to select key features from the original feature set before applying an ML model to predict outcomes. The feature selection process also helps to reduce irrelevant variables, computational cost, and the overfitting problem and improves the performance of ML models (Cai et al. 2012). If we select only a small number of features as input for an ML model, the information may not be enough to make predictions. A large number of features also increase the running time and causes the generalization performance to deteriorate owing to the curse of dimensionality (Kim 2006). Therefore, only the most significant features that affect the results should be selected to achieve successful predictions.

Feature selection and extraction techniques helped obtain better predictions over periods of 10 min up to 1 month ahead in terms of absolute price or direction. Therefore, ignoring feature selection in stock market analysis can have negative effects, such as overfitting, which is likely to damage the overall prediction results of a given ML model (Htet Htet Htun et al 2023).

Regarding the types of features, most studies considered either basic features or technical or fundamental indicators. The number of studies that applied both basic features and technical indicators was lower than the number of studies that applied one

type of feature. Therefore, further research is required to employ multiple feature types from different categories. In Rana et al. (2019), closing price was found to be the most significant feature among the basic features; therefore, future work should consider applying closing price and technical indicators as input features to the model.

In the light of the above insights, this paper explores the use of Basic Features and Technical Indicators to predict stock prices. A combination of Basic features and Technical Indicators derived from traditional use by stock brokers and brokerage houses have been considered for the purpose of the study. Close Price and Volume have been selected directly from OHLCAV data and the remaining are derived features:-

(a) Close Price (Close) : is the Target value considered for prediction.
(b) Volume : of stocks traded on a daily basis.
(c) Standard Deviation (Std) : 20 day rolling standard deviation of Close price.
(d) Relative Volume (Rel_Vol) : Daily volume divided by 3 month rolling average.
(e) Relative Strength Index (RSI) : 100 - 100 / (1 + [Rolling average of 14 gain days divided by rolling average of 14 loss days]).
(f) Actual True Range (ATR) : 14 day rolling mean of daily High - daily Low.
(g) Market Capitalisation (Market_Cap) : Daily volume multiplied by the Adj Close.

2.2 Splitting Train & Test Data

Htet Htet Htun et al (2023), observed that most studies divided the experimental datasets into 70% training and 30% testing datasets to evaluate the performance of the predictive models. To consider a more practical problem of stock market forecasting, future research should use the sliding window method in splitting the sample into different groups of training and testing periods. The primary reason for using this method is that investors are always interested in the most recent stock trends but not in long-term historical data. Therefore, the predictive models should be updated periodically throughout the process. Future studies should examine the performance of the results based on different widths of the sliding window (one month, three months, six months, and one year) for the training and testing data because the movement of stock prices displays periodic behaviour over various time scales

In the light of the above, for the study carried out in this paper the train and test data are split based on the Train-Test combination of n-number of continuous trading days as training data and subsequent 7 continuous trading days as test data.

2.3 Performance Parameters

Deepak Kumar et al (2020), in their insights from survey of various studies on Stock Market prediction model performance observed that:-

(a) **Root mean square error (RMSE)**. The RMSE is used to calculate the difference between the expected model values and the retained values (M.M. Pai et al. 2016). RMSE is very close to the training and assessment database. 20% of the selected studies used the RMSE parameter for prediction performance.

(b) **Mean absolute error (MAE)**. MAE is used in regression values (S. Selvin et al 2017 & Ritika Singh et al 2017). In this case, error prediction is the sum of the absolute differences between the expected and actual variables, divided by the number of data points above all data points. MAE refers to calculating the difference between two continuous variables. 16% of the selected studies used the MAE parameter for prediction performance.

(c) **Mean squared error (MSE**). MSE is squared average error used as a loss function for calculating the minimum square regression (M.R. Vargas et al 2017 & K.A. Althelaya et al 2018). Also, it is the sum of the squared differences between the expected and actual variables, divided by the number of data points above all data points. 21% of the selected studies used the MSE parameter for prediction performance.

(d) **Mean absolute percentage error (MAPE)**. MAPE is most widely used in key performance index (KPI) for calculating the stock market forecasting (R. Ray et al 2018). It is the sum of absolute individual errors separated by the demand (E. Guresen et al 2011). It is a percentage error average. 11% of the selected studies used the MAPE parameter for prediction performance.

Thus, in this study, evaluation of the models are carried out based on reliable financial data test metrics namely, Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). In addition, we have included R2-Score as well, which is a measure of goodness of fit for Regression models. It is pertinent to mention here that R2-Score is not a conclusive test-metric and need to be interpreted in conjunction with all other test metrics. Here we seek to explore the suitability of this test metric on Stock prices data.

2.4 Over-fitting vs Under-fitting

In our study we analyse the models for their performance with the help of the test-metrics. We should avoid overfitting, which happens when the model is too complex and starts to learn the noise in the data, this implies that the model would display low error on training data and high error on test data. While underfitting results from a model that is too simple to capture the underlying relationships in the data, this implies that the model would poorly perform on both training and test data.

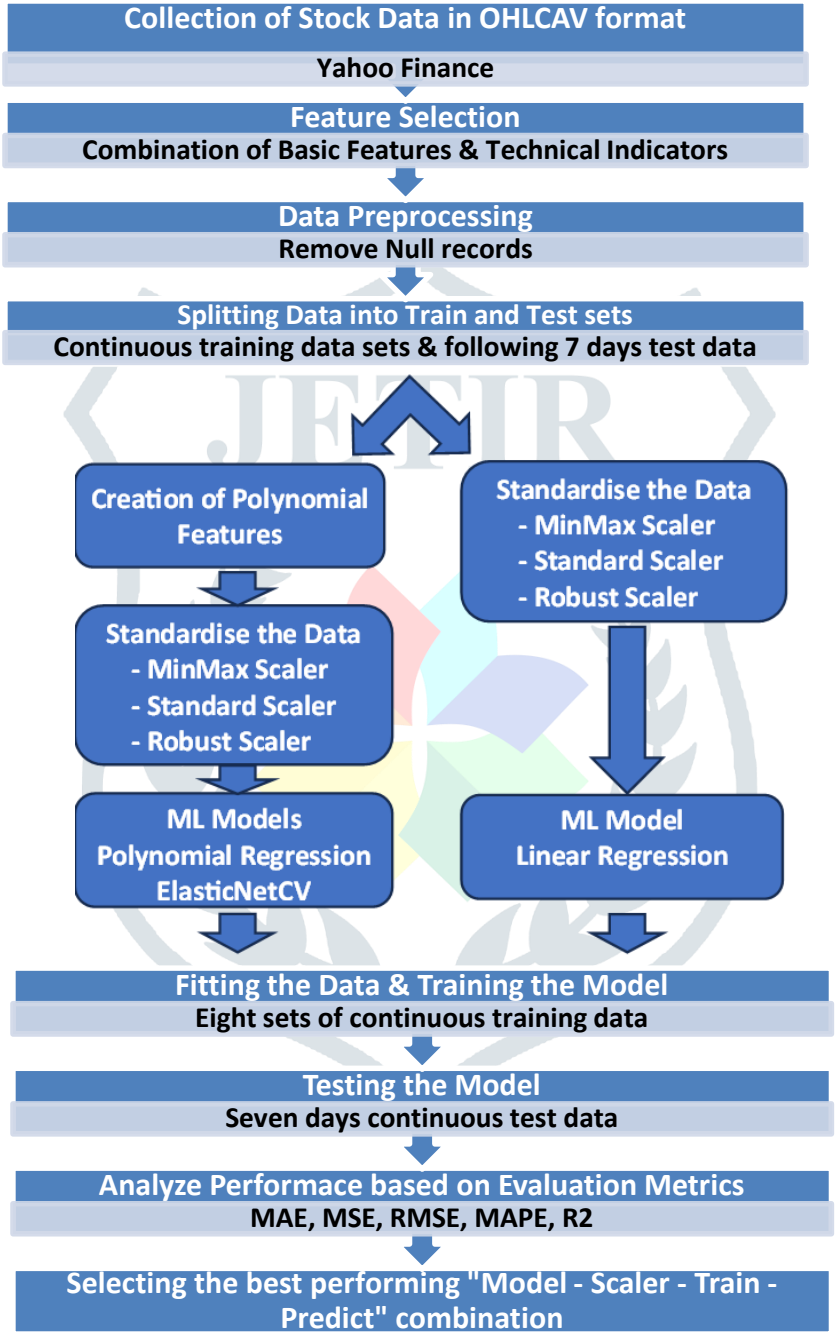
III. RESEARCH METHODOLOGY

3.1 Data

The study was conducted on four historical datasets of four different Stocks, namely AXIS BANK, TATA MOTORS, GE and APPLE beginning from 01 April 2020 till 26 February 2024, a total of 968 trading days, data loaded directly from Yahoo Finance using Pandas DataReader. We will discuss the study carried out using AXIS BANK data, ticker - 'AXISBANK.NS', for the major portion of this paper.

3.2 Work Flow

The study was carryout using the following work flow:-



3.3 Feature Selection & Data Preprocessing

Feature Selection was carried out as discussed in the preceding sections and records with null values were removed. After feature selection and dropping the resultant null values the final dataframe for the study consisted of 879 rows and 7 columns / features created from OHLCAV stock prices - period from 12 August 2020 till 26 February 2024.

```
DatetimeIndex: 879 entries, 2020-08-12 to 2024-02-26
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Close       879 non-null    float64
 1   Volume      879 non-null    int64
 2   Std         879 non-null    float64
 3   Rel_Vol     879 non-null    float64
 4   RSI         879 non-null    float64
 5   ATR         879 non-null    float64
 6   Market_cap  879 non-null    float64
dtypes: float64(6), int64(1)
```

Table -1 : Final dataframe considered for the study

3.4 Splitting Data into Train-Test sets

The above data was then split into Train-Test combinations of n-number of continuous trading days as training data and subsequent 7 continuous trading days as test data in the following manner:-

#	**Train-Test Combination** (Trading Days)	**Train Data Shape & Range**	**Test Data Shape & Range**
1.	Train on complete data and Test 7 days	(872, 7) ('2020-08-12', '2024-02-15')	(7, 7) ('2024-02-16', '2024-02-26')
2.	Train on half of the data and Test 7 days	(439, 7) ('2022-05-11', '2024-02-15')	(7, 7) ('2024-02-16', '2024-02-26')
3.	Train on one-fourth of the data and Test 7 days	(220, 7) ('2023-03-24', '2024-02-15')	(7, 7) ('2024-02-16', '2024-02-26')
4.	Train on 100 days of data and Test 7 days	(100, 7) ('2023-09-20', '2024-02-15')	(7, 7) ('2024-02-16', '2024-02-26')
5.	Train on 35 days of data and Test 7 days	(35, 7) ('2023-12-27', '2024-02-15')	(7, 7) ('2024-02-16', '2024-02-26')
6.	Train on 21 days of data and Test 7 days	(21, 7) ('2024-01-16', '2024-02-15')	(7, 7) ('2024-02-16', '2024-02-26')
7.	Train on 14 days of data and Test 7 days	(14, 7) ('2024-01-26','2024-02-15')	(7, 7) ('2024-02-16', '2024-02-26')
8.	Train on 7 days of data and Test 7 days	(7, 7) ('2024-02-07', '2024-02-15')	(7, 7) ('2024-02-16', '2024-02-26')

Table -2 : Test-Train combinations considered for the study

3.5 Creation of Polynomial Features

After the Train-Test splits into eight different combinations as described above, each Train-Test combination is further split into X_train, X_test, y_train and y_test with the '**Close**' column as the target feature, which we will be predicting using the ML models. Two copies of X_train & X_test were made. One copy of this X_train & X_test was used for the creation of polynomial features using the PolynomialFeatures function from sklearn.preprocessing module. We will call this polynomial feature set as "Polynomial Features" (having 28 features / predictors), and the other set as "Regular Features" (having 6 features / predictors).

3.6 Standardisation

At this stage, both the Polynomial Features and Regular Features are scaled using MinMaxScaler, StandardScaler and RobustScaler from the sklearn.preprocessing module which is carried out using a Python for-loop and the respective scaled data passed to the model for prediction. One additional set of the same data which is kept un-scaled is also passed to the model for comparing the model prediction with both the scaled data and un-scaled data. Here, only the X_train data is fitted and transformed whereas the X_test data is only transformed in order to maintain the scaling uniformity.

```
#Perform scaling on train and test data
for scaler in [Standard_Scaler, Min_Max_Scaler, Robust_Scaler,'No Scaling']:
    if scaler == 'No Scaling':
        X_train_scaled = X_train
        X_test_scaled = X_test
    else:
        X_train_scaled = scaler.fit_transform(X_train)
        X_test_scaled  = scaler.transform(X_test)
```

3.7 Linear Regression Models

Three Linear Regression models from Scikit Learn library of Python were selected as under:-

(a) **Linear Regression**. Here Regular Features are passed to the model as X_train.

```
from sklearn.linear_model import LinearRegression
model_lr = LinearRegression()
model_lr.fit(X_train, y_train)
```

(b) **Polynomial Regression**. Here the process is same as in Linear Regression, however, Polynomial Features are passed to the model as X_train.

```
from sklearn.linear_model import LinearRegression
model_lr = LinearRegression()
model_lr.fit(X_train, y_train)
```

(c) **ElasticNetCV**. Here also, Polynomial Features are passed to the model as X_train. L1 ratio is the ElasticNet scaling between l1 and l2 penalties, a good choice of list of values for l1_ratio is to put more values close to 1 (i.e. Lasso) and less values close to 0 (i.e. Ridge), as in [.1, .5, .7, .9, .95, .99, 1].

```
from sklearn.linear_model import ElasticNetCV
elastic_model = ElasticNetCV(l1_ratio=[.1, .5, .7,.9, .95, .99, 1],eps=0.01,n_alphas=100,max_iter=100000,cv=2)
elastic_model.fit(X_train,y_train)
```

The models were fitted and trained on the training data set and tested on the test data i.e. X_test for each Train-Test combination as listed above. The test metrics for the evaluation of the models are compiled and the resultant dataframe was analysed for insights.

IV. RESULTS AND DISCUSSION

The results obtained for the eight combinations of the Train-Test data are as shown in Table -3. We found that the MSE values (highlighted in red box) decreases as the number of data values in the train data decreases and reaches the min value for the Train-Test combination of “Train on 35 days of data and Test 7 days” and further increases as the number of data values in the train data decreases for each model considered by us. For the Train-Test combination of “Train on 35 days of data and Test 7 days”, the MSE value is the highest for ElasticNetCV, followed by Linear Regression and Polynominal Regression. We also found that the same is true of other test metrics MAE, RMSE and MAPE. Interestingly we also found that the value of R2-Score has moved to the negative for all the three models for certain combinations of the Train-Test data, indicating that R2-Score may not be a good metric for gauging the performance of Linear Regression models on Stock prices data.

Table -3 : Results obtained for eight combinations of the Train-Test data

	Train-Test Combination	Test Metric	Scaler	ELASTIC_NET_CV	LINEAR REGRESSION	POLYNOMIAL REGRESSION
0	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MAPE	MinMaxScaler()	0.11	0.11	0.05
1	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MAPE	No Scaling	0.27	0.11	0.08
2	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MAPE	RobustScaler()	0.1	0.11	0.05
3	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MAPE	StandardScaler()	0.09	0.11	0.05
4	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MAE	MinMaxScaler()	122.43	123.42	52.13
5	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MAE	No Scaling	292.66	123.42	86.29
6	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MAE	RobustScaler()	104.68	123.42	52.13
7	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MAE	StandardScaler()	94.09	123.42	52.13
8	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test RMSE	MinMaxScaler()	127.7	133.79	59.57
9	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test RMSE	No Scaling	292.94	133.79	93.17
10	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test RMSE	RobustScaler()	112.97	133.79	59.57
11	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test RMSE	StandardScaler()	102.65	133.79	59.57
12	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MSE	MinMaxScaler()	16307.25	17899.2	3548.41
13	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MSE	No Scaling	85814.62	17899.2	8681.11
14	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MSE	RobustScaler()	12762.8	17899.2	3548.41
15	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test MSE	StandardScaler()	10537.18	17899.2	3548.41
16	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test R2	MinMaxScaler()	-98.46	-108.17	-20.64
17	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test R2	No Scaling	-522.39	-108.17	-51.95
18	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test R2	RobustScaler()	-76.84	-108.17	-20.64
19	TRAIN_COMPLETE_DATA_TEST_7_DAYS	Test R2	StandardScaler()	-63.27	-108.17	-20.64
20	TRAIN_HALF_DATA_TEST_7_DAYS	Test MAPE	MinMaxScaler()	0.03	0.04	0.02
21	TRAIN_HALF_DATA_TEST_7_DAYS	Test MAPE	No Scaling	0.17	0.04	0.02
22	TRAIN_HALF_DATA_TEST_7_DAYS	Test MAPE	RobustScaler()	0.02	0.04	0.02
23	TRAIN_HALF_DATA_TEST_7_DAYS	Test MAPE	StandardScaler()	0.02	0.04	0.02
24	TRAIN_HALF_DATA_TEST_7_DAYS	Test MAE	MinMaxScaler()	31.68	45.48	19.35
25	TRAIN_HALF_DATA_TEST_7_DAYS	Test MAE	No Scaling	189.02	45.48	21.05
26	TRAIN_HALF_DATA_TEST_7_DAYS	Test MAE	RobustScaler()	21.96	45.48	20.85
27	TRAIN_HALF_DATA_TEST_7_DAYS	Test MAE	StandardScaler()	24.84	45.48	20.55
28	TRAIN_HALF_DATA_TEST_7_DAYS	Test RMSE	MinMaxScaler()	35.43	53.62	22.19
29	TRAIN_HALF_DATA_TEST_7_DAYS	Test RMSE	No Scaling	189.37	53.62	25.52
30	TRAIN_HALF_DATA_TEST_7_DAYS	Test RMSE	RobustScaler()	27.67	53.62	22.48
31	TRAIN_HALF_DATA_TEST_7_DAYS	Test RMSE	StandardScaler()	29.14	53.62	22.03
32	TRAIN_HALF_DATA_TEST_7_DAYS	Test MSE	MinMaxScaler()	1255.61	2875.4	492.61
33	TRAIN_HALF_DATA_TEST_7_DAYS	Test MSE	No Scaling	35860.53	2875.4	651.48
34	TRAIN_HALF_DATA_TEST_7_DAYS	Test MSE	RobustScaler()	765.82	2875.4	505.55
35	TRAIN_HALF_DATA_TEST_7_DAYS	Test MSE	StandardScaler()	849.34	2875.4	485.33
36	TRAIN_HALF_DATA_TEST_7_DAYS	Test R2	MinMaxScaler()	-6.66	-16.54	-2
37	TRAIN_HALF_DATA_TEST_7_DAYS	Test R2	No Scaling	-217.72	-16.54	-2.97
38	TRAIN_HALF_DATA_TEST_7_DAYS	Test R2	RobustScaler()	-3.67	-16.54	-2.08
39	TRAIN_HALF_DATA_TEST_7_DAYS	Test R2	StandardScaler()	-4.18	-16.54	-1.96
40	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MAPE	MinMaxScaler()	0.02	0.03	0.03
41	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MAPE	No Scaling	0.09	0.03	0.01
42	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MAPE	RobustScaler()	0.03	0.03	0.02
43	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MAPE	StandardScaler()	0.02	0.03	0.02
44	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MAE	MinMaxScaler()	21.18	28.48	33.16
45	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MAE	No Scaling	93.57	28.48	13.77
46	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MAE	RobustScaler()	27.26	28.48	18.71
47	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MAE	StandardScaler()	18.63	28.48	20.16
48	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test RMSE	MinMaxScaler()	23.19	32.48	35.82
49	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test RMSE	No Scaling	94.47	32.48	18.22
50	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test RMSE	RobustScaler()	29.01	32.48	24.31
51	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test RMSE	StandardScaler()	20.75	32.48	21.24
52	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MSE	MinMaxScaler()	537.71	1055.16	1282.81
53	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MSE	No Scaling	8924.83	1055.16	331.8
54	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MSE	RobustScaler()	841.39	1055.16	591.18
55	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test MSE	StandardScaler()	430.56	1055.16	451.1
56	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test R2	MinMaxScaler()	-2.28	-5.44	-6.82
57	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test R2	No Scaling	-53.43	-5.44	-1.02
58	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test R2	RobustScaler()	-4.13	-5.44	-2.61
59	TRAIN_ONEFOURTH_DATA_TEST_7_DAYS	Test R2	StandardScaler()	-1.63	-5.44	-1.75
60	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MAPE	MinMaxScaler()	0.03	0.01	0.01
61	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MAPE	No Scaling	0.03	0.01	0
62	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MAPE	RobustScaler()	0.03	0.01	0
63	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MAPE	StandardScaler()	0.03	0.01	0.01
64	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MAE	MinMaxScaler()	29.78	12.47	9.06
65	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MAE	No Scaling	30.07	12.47	2.81
66	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MAE	RobustScaler()	29.78	12.47	4.58
67	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MAE	StandardScaler()	29.78	12.47	15.86
68	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test RMSE	MinMaxScaler()	32.42	13.89	9.56
69	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test RMSE	No Scaling	32.52	13.89	3.42
70	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test RMSE	RobustScaler()	32.42	13.89	6.05
71	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test RMSE	StandardScaler()	32.42	13.89	17.02
72	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MSE	MinMaxScaler()	1050.85	193.01	91.44
73	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MSE	No Scaling	1057.58	193.01	11.68
74	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MSE	RobustScaler()	1050.85	193.01	36.61
75	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test MSE	StandardScaler()	1050.85	193.01	289.6

Table -3 : Results obtained for eight combinations of the Train-Test data

	Train-Test Combination	Test Metric	Scaler	ELASTIC_NET_CV	LINEAR REGRESSION	POLYNOMIAL REGRESSION
76	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test R2	MinMaxScaler()	-5.41	-0.18	0.44
77	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test R2	No Scaling	-5.45	-0.18	0.93
78	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test R2	RobustScaler()	-5.41	-0.18	0.78
79	TRAIN_100_DAYS_DATA_TEST_7_DAYS	Test R2	StandardScaler()	-5.41	-0.18	-0.77
80	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MAPE	MinMaxScaler()	0.01	0	0
81	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MAPE	No Scaling	0.01	0	0
82	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MAPE	RobustScaler()	0.01	0	0
83	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MAPE	StandardScaler()	0.01	0	0
84	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MAE	MinMaxScaler()	9.07	3.81	1.44
85	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MAE	No Scaling	11.64	3.81	5.35
86	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MAE	RobustScaler()	8.69	3.81	1.47
87	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MAE	StandardScaler()	9.15	3.81	1.56
88	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test RMSE	MinMaxScaler()	10.65	4.82	1.62
89	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test RMSE	No Scaling	14.71	4.82	6.04
90	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test RMSE	RobustScaler()	10.04	4.82	1.64
91	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test RMSE	StandardScaler()	10.65	4.82	1.7
92	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MSE	MinMaxScaler()	113.33	23.26	2.63
93	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MSE	No Scaling	216.47	23.26	36.51
94	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MSE	RobustScaler()	100.87	23.26	2.69
95	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test MSE	StandardScaler()	113.39	23.26	2.9
96	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test R2	MinMaxScaler()	0.31	0.86	0.98
97	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test R2	No Scaling	-0.32	0.86	0.78
98	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test R2	RobustScaler()	0.38	0.86	0.98
99	TRAIN_35_DAYS_DATA_TEST_7_DAYS	Test R2	StandardScaler()	0.31	0.86	0.98
100	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MAPE	MinMaxScaler()	0.01	0.01	0
101	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MAPE	No Scaling	0.01	0.01	0
102	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MAPE	RobustScaler()	0.01	0.01	0
103	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MAPE	StandardScaler()	0.01	0.01	0
104	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MAE	MinMaxScaler()	10.69	5.82	2.9
105	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MAE	No Scaling	15.78	5.82	3.26
106	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MAE	RobustScaler()	12.7	5.82	4.32
107	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MAE	StandardScaler()	11.46	5.82	2.81
108	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test RMSE	MinMaxScaler()	12.8	7.09	3.38
109	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test RMSE	No Scaling	18.05	7.09	3.85
110	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test RMSE	RobustScaler()	15.27	7.09	4.8
111	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test RMSE	StandardScaler()	13.61	7.09	3.32
112	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MSE	MinMaxScaler()	163.85	50.21	11.45
113	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MSE	No Scaling	325.91	50.21	14.79
114	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MSE	RobustScaler()	233.1	50.21	23.05
115	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test MSE	StandardScaler()	185.27	50.21	11
116	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test R2	MinMaxScaler()	0	0.69	0.93
117	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test R2	No Scaling	-0.99	0.69	0.91
118	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test R2	RobustScaler()	-0.42	0.69	0.86
119	TRAIN_21_DAYS_DATA_TEST_7_DAYS	Test R2	StandardScaler()	-0.13	0.69	0.93
120	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MAPE	MinMaxScaler()	0.01	0.01	0
121	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MAPE	No Scaling	0.02	0.01	0.01
122	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MAPE	RobustScaler()	0.01	0.01	0
123	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MAPE	StandardScaler()	0.01	0.01	0
124	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MAE	MinMaxScaler()	11.53	11.23	3.65
125	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MAE	No Scaling	24.32	11.23	6.77
126	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MAE	RobustScaler()	12.28	11.23	3.47
127	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MAE	StandardScaler()	11.71	11.23	3.57
128	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test RMSE	MinMaxScaler()	13.22	13.01	4.94
129	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test RMSE	No Scaling	27.1	13.01	7.97
130	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test RMSE	RobustScaler()	14.06	13.01	4.43
131	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test RMSE	StandardScaler()	13.4	13.01	5.06
132	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MSE	MinMaxScaler()	174.71	169.14	24.45
133	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MSE	No Scaling	734.27	169.14	63.44
134	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MSE	RobustScaler()	197.69	169.14	19.6
135	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test MSE	StandardScaler()	179.48	169.14	25.55
136	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test R2	MinMaxScaler()	-0.07	-0.03	0.85
137	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test R2	No Scaling	-3.48	-0.03	0.61
138	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test R2	RobustScaler()	-0.21	-0.03	0.88
139	TRAIN_14_DAYS_DATA_TEST_7_DAYS	Test R2	StandardScaler()	-0.09	-0.03	0.84
140	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MAPE	MinMaxScaler()	0.02	0.15	0.01
141	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MAPE	No Scaling	0.02	0.15	0.04
142	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MAPE	RobustScaler()	0.03	0.15	0.01
143	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MAPE	StandardScaler()	0.02	0.15	0.01
144	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MAE	MinMaxScaler()	17.14	163.93	11.46
145	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MAE	No Scaling	24.64	163.93	42.47
146	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MAE	RobustScaler()	27.88	163.93	15.39
147	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MAE	StandardScaler()	18.41	163.93	11.88
148	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test RMSE	MinMaxScaler()	19.62	172.24	15.67
149	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test RMSE	No Scaling	27.34	172.24	44.1
150	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test RMSE	RobustScaler()	30.02	172.24	19.85
151	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test RMSE	StandardScaler()	20.75	172.24	16.2
152	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MSE	MinMaxScaler()	385.12	29666.77	245.54
153	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MSE	No Scaling	747.45	29666.78	1944.78
154	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MSE	RobustScaler()	901.19	29666.77	393.96
155	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test MSE	StandardScaler()	430.76	29666.77	262.28
156	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test R2	MinMaxScaler()	-1.35	-179.94	-0.5
157	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test R2	No Scaling	-3.56	-179.94	-10.86
158	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test R2	RobustScaler()	-4.5	-179.94	-1.4
159	TRAIN_7_DAYS_DATA_TEST_7_DAYS	Test R2	StandardScaler()	-1.63	-179.94	-0.6

When we plot the data in Table -3, the plot was found to subdue certain low value test metric due to some large values of MSE, so we dropped the MSE values and re-plotted the values in Table -3. We got the Regressor vs Test-Score plot in Fig-1.

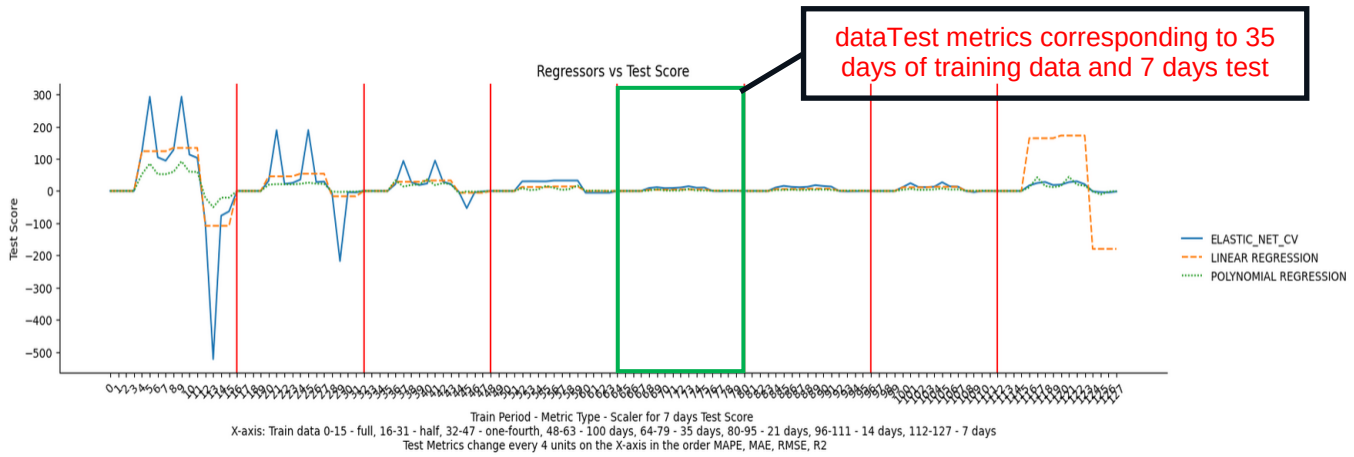


Fig -1 : Plot of test-metric results for eight combination of the Train-Test data

From Fig -1 and Table -3 it can be observed that:-

(a) The Test-Train combination of "Train on 35 days of data and Test 7 days" offers the best stable test-metrics for Axis Bank stock data prediction using Linear Regression models.

(b) For Linear Regression models, R2-Scores for all models go into the negative for large training data set above the "Train on 35 days of data and Test 7 days" combination and for certain models below this combination. Thus, indicating that R2-Score may not be a good metric for gauging the performance of Linear Regression models on Stock data.

Now we drop the R2-Score from Table -3 and plot again for a clearer picture shown in Fig – 2.

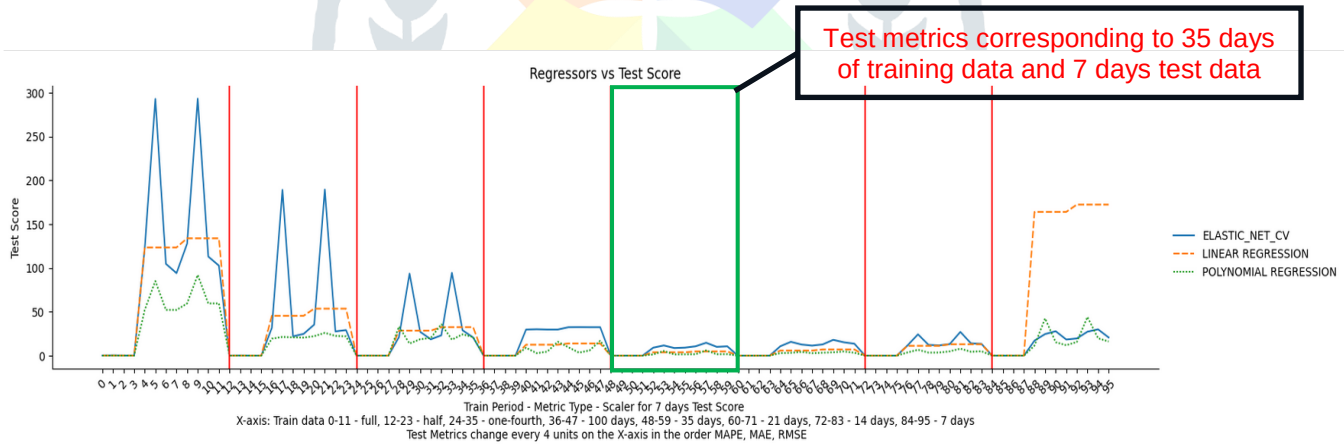


Fig -2 : Plot of test-metric results for eight combination of the Train-Test data

From Fig -2 and Table -3 it can be observed that:-

(a) For Test-Train combination above and below "Train on 35 days of data and Test 7 days", i.e. for large training data above 35 days and smaller number data below, the test-metrics show large variance for all the considered models, irrespective of the scaling method used.

(b) Linear Regression test metrics display comparable results for all three scalers within this Train-Test combination for all the three different models.

Table -4 displays all the test-metrics for the Train-Test combination "Train on 35 days of data and Test 7 days" for which we observed that test-metrics for all three models displayed the best comparable results. Fig -3 displays the predicted values (red dots) against the actual values (blue dots) and the test-residuals against the Close (target) values.

Train on 35 days of data and Test 7 days - Test-Metrics

		Train MAPE	Test MAPE	Train MAE	Test MAE	Train RMSE	Test RMSE	Train MSE	Test MSE	Train R2	Test R2	Intercept	Alpha	Best l1
LINEAR REGRESSION	StandardScaler()	0	0	4.65	3.81	5.84	4.82	34.11	23.26	0.96	0.86	1087.785714	-	-
	MinMaxScaler()	0	0	4.65	3.81	5.84	4.82	34.11	23.26	0.96	0.86	1099.654968	-	-
	RobustScaler()	0	0	4.65	3.81	5.84	4.82	34.11	23.26	0.96	0.86	1105.063295	-	-
	No Scaling	0	0	4.65	3.81	5.84	4.82	34.11	23.26	0.96	0.86	1114.174477	-	-
POLYNOMIAL REGRESSION	StandardScaler()	0	0	0.41	1.56	0.62	1.7	0.39	2.9	1	0.98	1087.785714	-	-
	MinMaxScaler()	0	0	0.44	1.44	0.64	1.62	0.4	2.63	1	0.98	1081.25448	-	-
	RobustScaler()	0	0	0.41	1.47	0.62	1.64	0.38	2.69	1	0.98	1122.187284	-	-
	No Scaling	0	0	2.73	5.35	3.44	6.04	11.81	36.51	0.98	0.78	1099.256841	-	-
ELASTIC_NET_CV	StandardScaler()	0.01	0.01	8.11	9.15	9.95	10.65	99	113.39	0.87	0.31	-	0.610854574	1
	MinMaxScaler()	0.01	0.01	8.24	9.07	10.1	10.65	102.06	113.33	0.87	0.31	-	0.221904744	1
	RobustScaler()	0.01	0.01	7.41	8.69	9.15	10.04	83.72	100.87	0.89	0.38	-	0.362002097	1
	No Scaling	0.02	0.01	24.05	11.64	27.07	14.71	732.77	216.47	0.05	-0.32	-	4.7667E+20	0.99

Table -4 : Results obtained for Train-Test combination : "Train on 35 days of data and Test 7 days"

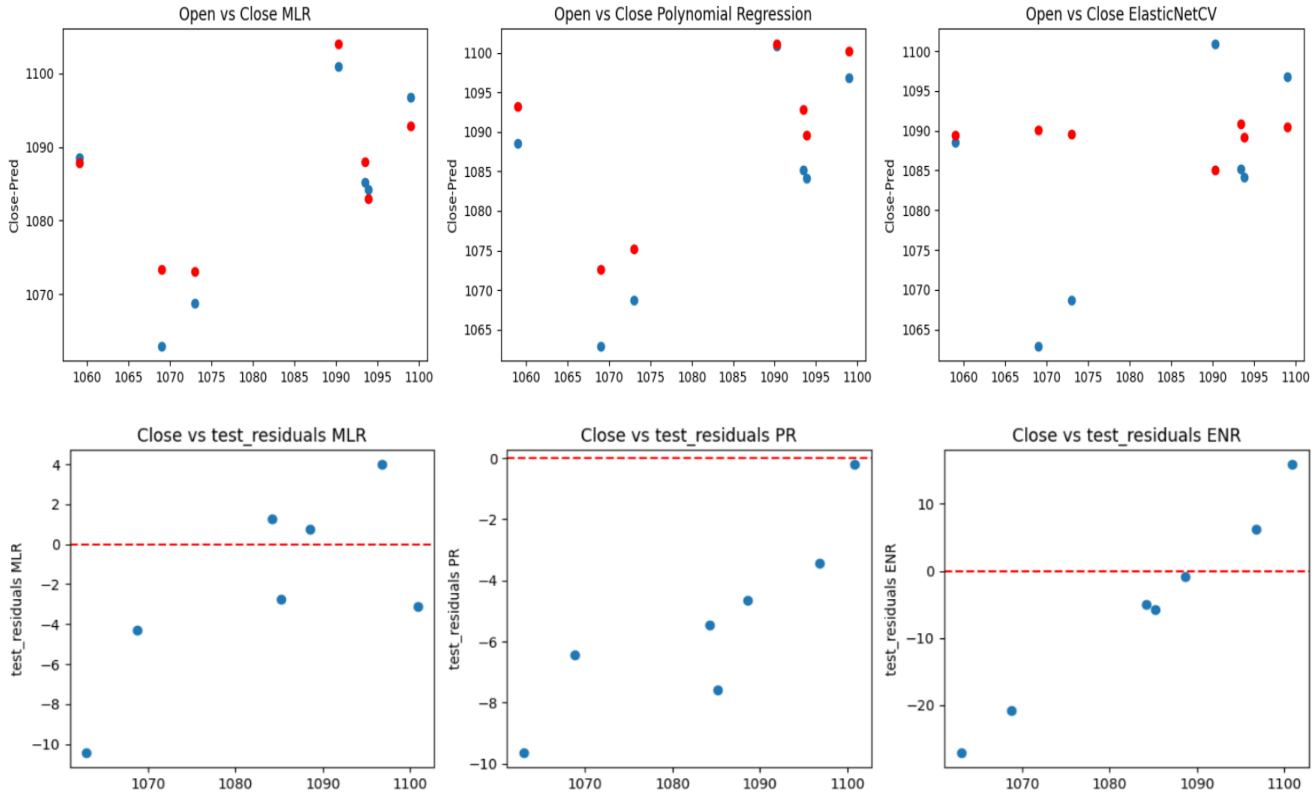


Fig -3 : Plot of Predicted values against actual Close values and test-residuals against Close for Train-Test combination : "Train on 35 days of data and Test 7 days"

From Table -4 and Fig -3 it can be observed that :-

(a) Though the Polynomial Regression is giving the best test-metrics, it may likely be very slightly overfitting on the data which is also the case with the ElasticNetCV.

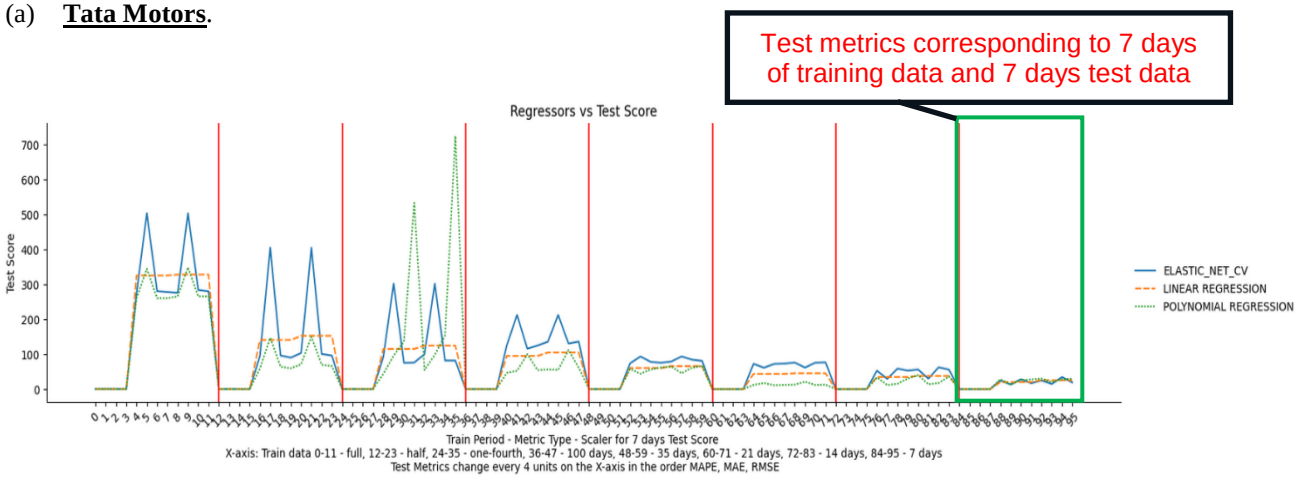
(b) The Linear Regression model shows better performance on test data compared to the training data, however its performance is way lesser than Polynomial Regression.

(c) Train and Test MAPE closer to 0% for all three models can be considered as excellent metrics for stock price prediction performance of the models.

(d) Polynomial Regression model with MinMax Scaler gives the best prediction performance with Test RMSE = 1.62, Test MAE = 1.44 and Test MSE = 2.63.

43. At this stage we went on to carry out the study of three other stocks viz. 'TATAMOTORS.NS', 'GE' & "AAPL' in a similar fashion as the 'AXISBANK.NS' stocks above. The results are summarised in Figs – 4, 5 & 6 below:-

(a) **Tata Motors**.

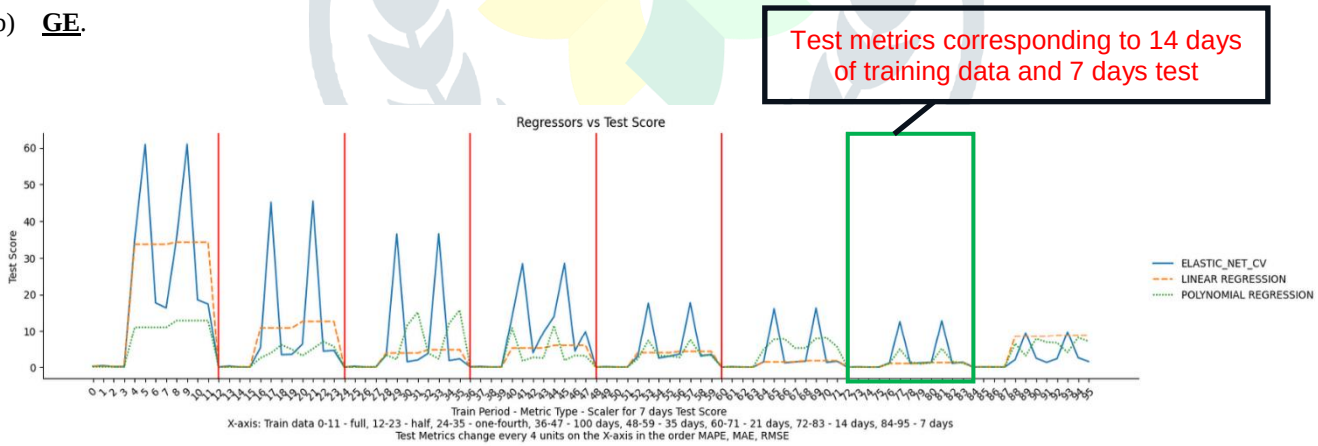


Train on 7 days of data and Test 7 days : Test-Metrics

		Train MAPE	Test MAPE	Train MAE	Test MAE	Train RMSE	Test RMSE	Train MSE	Test MSE	Train R2	Test R2	Intercept	Alpha	Best l1
LINEAR REGRESSION	StandardScaler()	0	0.02	0	20.22	0	25.5	0	650.36	1	-17.26	918.6357073	-	-
	MinMaxScaler()	0	0.02	0	20.22	0	25.5	0	650.36	1	-17.26	914.4892282	-	-
	RobustScaler()	0	0.02	0	20.22	0	25.5	0	650.36	1	-17.26	914.4195064	-	-
	No Scaling	0	0.02	0	20.22	0	25.5	0	650.36	1	-17.26	845.4126355	-	-
POLYNOMIAL REGRESSION	StandardScaler()	0	0.03	0	27.87	0	29.85	0	891.3	1	-24.03	918.6357073	-	-
	MinMaxScaler()	0	0.03	0	28.35	0	30.41	0	924.49	1	-24.96	928.5234241	-	-
	RobustScaler()	0	0.03	0	26.16	0	28.28	0	799.67	1	-21.46	919.4138698	-	-
	No Scaling	0	0.01	0	12.78	0	20.01	0	400.36	1	-10.24	945.4022543	-	-
ELASTIC_NET_CV	StandardScaler()	0	0.02	4.24	16.87	4.96	18.94	24.65	358.84	0.63	-9.08	-	3.928363565	1
	MinMaxScaler()	0	0.03	3.7	24.74	4.84	24.97	23.44	623.34	0.65	-16.5	-	1.261507843	1
	RobustScaler()	0	0.03	4.33	28.4	4.99	34.75	24.86	1207.4	0.63	-32.91	-	4.990644045	1
	No Scaling	0.01	0.01	6.5	13.54	8.16	14.8	66.61	219.02	0	-5.15	-	4.58667E+20	0.1

Fig - 4 : TATAMOTORS : Plot of test-metric results for eight combination of the Train-Test data & Table of Results obtained for Train-Test combination : "Train on 7 days of data and Test 7 days"

(b) **GE**.

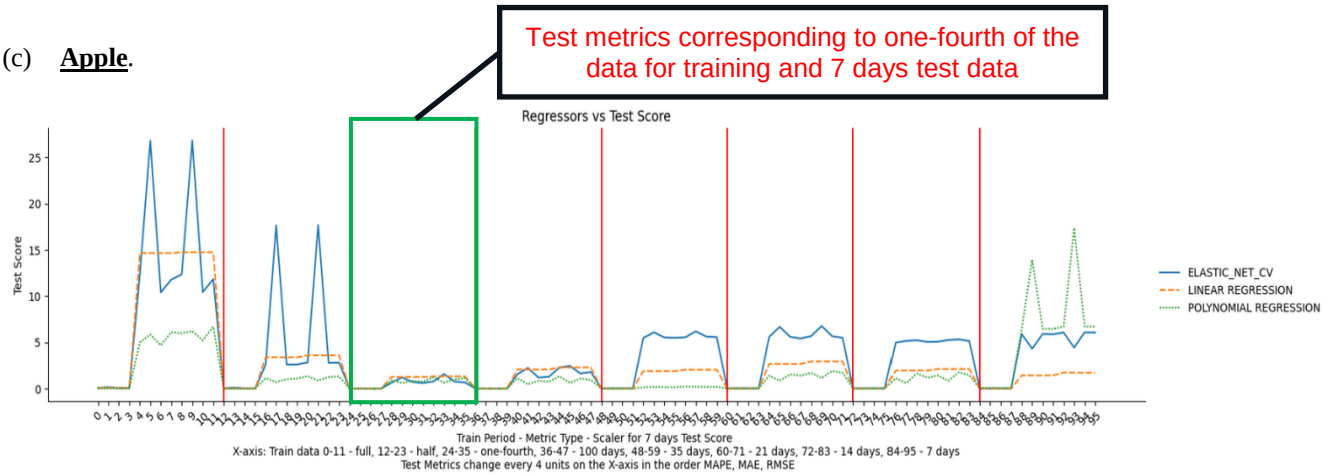


Train on 14 days of data and Test 7 days : Test-Metrics

		Train MAPE	Test MAPE	Train MAE	Test MAE	Train RMSE	Test RMSE	Train MSE	Test MSE	Train R2	Test R2	Intercept	Alpha	Best l1
LINEAR REGRESSION	StandardScaler()	0	0.01	0.48	0.99	0.53	1.24	0.29	1.53	0.99	0.72	137.9680003	-	-
	MinMaxScaler()	0	0.01	0.48	0.99	0.53	1.24	0.29	1.53	0.99	0.72	129.2035402	-	-
	RobustScaler()	0	0.01	0.48	0.99	0.53	1.24	0.29	1.53	0.99	0.72	137.2184308	-	-
	No Scaling	0	0.01	0.48	0.99	0.53	1.24	0.29	1.53	0.99	0.72	103.0522114	-	-
POLYNOMIAL REGRESSION	StandardScaler()	0	0.01	0	0.89	0	1.11	0	1.23	1	0.77	137.9680003	-	-
	MinMaxScaler()	0	0.01	0	0.95	0	1.17	0	1.38	1	0.75	117.2867118	-	-
	RobustScaler()	0	0.01	0	1.38	0	1.52	0	2.31	1	0.58	138.0449117	-	-
	No Scaling	0	0.03	0.02	4.92	0.03	5.09	0	25.86	1	-3.73	138.7913354	-	-
ELASTIC_NET_CV	StandardScaler()	0	0.01	0.58	1.18	0.7	1.35	0.49	1.82	0.98	0.67	-	0.220302529	1
	MinMaxScaler()	0	0.01	0.64	1.13	0.78	1.29	0.61	1.67	0.97	0.69	-	0.083869456	1
	RobustScaler()	0.01	0.01	0.71	0.97	0.84	1.01	0.71	1.02	0.97	0.81	-	0.186639532	0.7
	No Scaling	0.02	0.08	3.06	12.57	3.74	12.81	13.97	164.08	0.41	-29.01	-	2.74674E+16	0.9

Fig - 5 : GE : Plot of test-metric results for eight combination of the Train-Test data & Table of Results obtained for Train-Test combination : "Train on 14 days of data and Test 7 days"

(c) **Apple**.



Train on onefourth of the data and Test 7 days : Test-Metrics

		Train MAPE	Test MAPE	Train MAE	Test MAE	Train RMSE	Test RMSE	Train MSE	Test MSE	Train R2	Test R2	Intercept	Alpha	Best l1
LINEAR REGRESSION	StandardScaler()	0.01	0.01	1.64	1.26	2.42	1.32	5.88	1.73	0.94	-0.69	181.2034074	-	-
	MinMaxScaler()	0.01	0.01	1.64	1.26	2.42	1.32	5.88	1.73	0.94	-0.69	186.8686719	-	-
	RobustScaler()	0.01	0.01	1.64	1.26	2.42	1.32	5.88	1.73	0.94	-0.69	181.9817544	-	-
	No Scaling	0.01	0.01	1.64	1.26	2.42	1.32	5.88	1.73	0.94	-0.69	186.0977578	-	-
POLYNOMIAL REGRESSION	StandardScaler()	0	0	0.59	0.76	0.83	1.13	0.69	1.27	0.99	-0.24	181.2034074	-	-
	MinMaxScaler()	0	0	0.66	0.89	0.87	1.24	0.75	1.54	0.99	-0.5	195.2812072	-	-
	RobustScaler()	0	0	0.83	0.83	1.18	1	1.39	1.01	0.99	0.02	186.8647806	-	-
	No Scaling	0	0	0.58	0.61	0.8	0.65	0.63	0.43	0.99	0.58	185.7805235	-	-
ELASTIC_NET_CV	StandardScaler()	0.01	0	1.3	0.61	1.85	0.67	3.41	0.45	0.96	0.56	-	0.023705146	1
	MinMaxScaler()	0.01	0	1.59	0.65	2.29	0.78	5.26	0.6	0.95	0.41	-	0.005535552	1
	RobustScaler()	0.01	0	0.9	0.72	1.24	0.76	1.53	0.57	0.98	0.44	-	0.015803707	1
	No Scaling	0.05	0.01	8.38	1.21	9.8	1.57	96.01	2.47	0	-1.42	-	9.99759E+19	0.7

Fig - 6 : AAPL : Plot of test-metric results for eight combination of the Train-Test data & Table of Results obtained for Train-Test combination : "Train on one-fourth of the data and Test 7 days"

4.1 Findings

By the study of the results of test-metrics of the above stocks it becomes evident that the Regressors vs Test-Score plot will adequately indicate to us regarding the Train-Test combination that performs the best for a given set of Linear Regression models for a particular stock. Thereafter, a scrutiny of the dataframe for the test-metrics of the same Train-Test combination reveals models which are overfitting or underfitting and will give us the best performing Linear Regression model with the most suitable Train-Test-Scaler combination for prediction of stock prices of that particular stock.

V. CONCLUSION

In this study feature-selection of Stock data was carried out using Basic Features and Technical Indicators based on market best practices. We further went on to identify the best Train-Test-Scaling combination for modeling the time-series OHLCAV (Open-High-Low-Close-AdjClose-Volume) data that achieves optimum test-metric scores for Linear Regression Models.

During the course of our study we found that, features selected based on existing stock market best practices when modeled using Linear Regression can achieve very good prediction performance. The best performing Train-Test-Scaler combination for a given Linear Regression model can be discovered using the process as described in this paper. Train-Test combination of n-number of continuous trading days as training data and subsequent 7 continuous trading days as test data was used for the purpose of this study. The reader may try various other combinations using this method.

Theoretically, as well as intuitively, the best Train-Test combinations can be used as a scaling window for the prediction of stock prices. Studies on the same may be undertaken to conclusively ascertain the same.

R2-Score has not proved to be a good metric for gauging the performance of Linear Regression models in the prediction of stock prices.

Linear Regression model test metrics display comparable results for all three scalers used in the study viz. MinMax Scaler, Standard Scaler and Robust Scaler, within the best Train-Test combination.

REFERENCES

1. Alsubaie Y, Hindi KE, Alsalman H (2019) Cost-sensitive prediction of stock price direction: selection of technical indicators. IEEE Access 7:146876–146892
2. auxeno Alex; Linear Regression Masterclass – ML https://www.kaggle.com/code/auxeno/linear-regression-masterclass-ml?scriptVersionId=142416708
3. Botunac I, Panjkota A, Matetic M (2020) The effect of feature selection on the performance of long short-term memory neural network in stock market predictions, In 31st DAAAM ISIMA, Vienna, Austria, pp 0592-0598
4. Bustos O, Pomares-Quimbaya A (2020) Stock market movement forecast: a systematic review. Expert Syst Appl 156:113464
5. Cai X, Hu S, Lin X (2012) Feature extraction using restricted Boltzmann machine for stock price prediction, IEEE CSAE, Zhangjiajie, China, pp 80–83
6. Deepak Kumar, Pradeepta Kumar Sarangi and Rajit Verma A systematic review of stock market prediction using machine learning and statistical techniques https://doi.org/10.1016/j.matpr.2020.11.399 2214-7853/© 2020 Elsevier.
7. E. Guresen, G. Kayakutlu, T.U. Daim, Using artificial neural network models in stock market index prediction, Expert Syst. Appl. 38 (8) (2011) 10389–10397.
8. Fama EF (1995) Random walks in stock market prices. Financ Anal J 51(1):75–80
9. Htet Htet Htun, Michael Biehl and Nicolai Petkov Survey of feature selection and extraction techniques for stock market prediction Htun et al. Financial Innovation (2023) 9:26 https://doi.org/10.1186/s40854-022-00441-7 Springer Open.
10. K.A. Althelaya, E.S.M. El-Alfy, S. Mohammed. (2018, April). Evaluation of bidirectional lstm for short-and long-term stock market prediction. In 2018 9th international conference on information and communication systems (ICICS) (pp. 151-156). IEEE.
11. Kim Y (2006) Toward a successful CRM: variable selection, sampling, and ensemble. Decis Support Syst 41:542–55338:103–123
12. M.R. Vargas, B.S. De Lima, A.G. Evsukoff. (2017, June). Deep learning for stock market prediction from financial news articles. In 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) (pp. 60-65). IEEE.
13. M.M. Pai, A. Nayak, R.M. Pai, Prediction models for the Indian stock market, Procedia Comput. Sci. 89 (2016) 441–449.
14. Malkiel BG (2003) The efficient market hypothesis and its critics. J Econ Perspect 17(1):59–82
15. Nikou M, Mansourfar G and Bagherzadeh J. Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. Intell Sys Acc Fin Mgmt. 2019;1–11. https://doi.org/10.1002/isaf.1459 Wiley.
16. Nti IK, Adekoya AF, Weyori BA (2020a) A systematic review of fundamental and technical analysis of stock market predictions.
17. Qolipour F, Ghasemzadeh M, Mohammad-Karimi N (2021) The predictability of tree-based machine learning algorithms in the big data context. Inter J Eng 34(01):82–89
18. R. Ray, P. Khandelwal, B. Baranidharan, 2018, December. A survey on stock market prediction using artificial intelligence techniques. In 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 594-598). IEEE.
19. Rana M, Uddin MM, Hoque MM (2019) Effects of activation functions and optimizers on stock price prediction using LSTM recurrent networks, CSAI, Beijing, China, pp 354–358
20. Ritika Singh, Shashi Srivastava, Stock prediction using deep learning, Multimedia Tools Appl. 76 (18) (2017) 18569–18584.
21. S. Selvin, R. Vinayakumar, E.A. Gopalakrishnan, V.K. Menon, K.P. Soman. Stock price prediction using LSTM, RNN, and CNN-sliding window model. In2017 international conference on advances in computing, communications, and informatics (icacci) 2017 Sep 13 (pp. 1643-1647). IEEE.
22. Thakkar A, Chaudhari K (2021) Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions. Inf Fusion 65:95–107
23. Thakkar A, Chaudhari K (2021) A comprehensive survey on deep neural networks for stock market: the need, challenges, and future directions. Expert Syst Appl 177:114800
24. https://finance.yahoo.com/
25. Zhang J, Cui S, Xu Y, Li Q, Li T (2018) A novel data-driven stock price trend prediction system. Expert Syst Appl 97:60–69