Linear Regression Using for Stock Price Prediction

A MAJOR PROJECT REPORT

SUBMITED BY [Team 13]

CH.EN.U4AIE20011 **D.MANISH**

CH.EN.U4AIE20012 **D.MEHUL**

CH.EN.U4AIE20061 **SHAIK NAWAB MUDASIR**

CH.EN.U4AIE20062 **SIRIGIRI NAGA PAVAN SATHVIK REDDY**

CH.EN.U4AIE20067 **MADISETTY THARUN KUMAR**

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, CHENNAI, 601 103

May 2022

BONAFIDE CERTIFICATE

This is to certify that the major project report entitled

"Linear Regression Using for Stock Price Prediction"

submitted by [Team 13]

CH.EN.U4AIE20011 D.MANISH

CH.EN.U4AIE20012 D.MEHUL

CH.EN.U4AIE20061 SHAIK NAWAB MUDASIR

CH.EN.U4AIE20062 SIRIGIRI NAGA PAVAN SATHVIK REDDY

CH.EN.U4AIE20067 MADISETTY THARUN KUMAR

In partial fulfillment of the requirements for the award of the **Degree of Bachelor of Technology** in **ARTIFICIAL INTELLIGENCE** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering, Chennai.

Signature

Dr. Soumyendra Singh,

This project report was evaluated by us on ……...

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We would like to offer our sincere pranams at the lotus feet of Universal guru, **MATA AMRITANANDAMAYI DEVI** who blessed us with her grace to make this a successful major project.

We express our deep sense of gratitude to **Dr. Prasanna Kumar R,** Chairperson, for his constant help, suggestions, and inspiring guidance. We are grateful to our guide **Dr. Soumyendra Singh** Department of ASE, Chennai for his invaluable support and guidance during the major project work.

We would also like to extend our gratitude to our director **Shri. Manikandan**, Principal **Dr. Shankar** who has always encouraged us. We are also thankful to all our classmates who have always been a source of strength, for always being there and extending their valuable bits of help to the successful completion of this work.

Table of contents		

Abstract

Researchers have been studying different methods to effectively predict the stock market price. Useful prediction systems allow traders to get better insights about data such as: future trends. Also, investors have a major benefit since the analysis give future conditions of the market. One such method is to use machine learning algorithms for forecasting. This project's objective is to improve the quality of output of stock market predicted by using stock value. A number of researchers have come up with various ways to solve this problem, mainly there are traditional methods so far, such as artificial neural network is a way to get hidden patterns and classify the data which is used in predicting stock market. This project proposes a different method for prognosting stock market prices. It does not fit the data to a specific model; rather we are identifying the latent dynamics existing in the data using machine learning architectures. In this work we use Machine learning regression model like linear regression, lasso, Ridge using or the price forecasting of Tesla stock price. On a long term basis, sling window approach has been applied and the performance was assessed by using root mean square error.

Introduction

Stock price analysis has been a critical area of research and is one of the top machine learning applications. This paper will show you how to predict stock prices with machine learning and deep learning techniques. Then will train the model with Tesla stocks price using an regression models.

A stock market is a public market in which shares of publicly traded companies can be bought and sold. Stocks, also known as equities, are ownership stakes in a company. The stock exchange acts as a middleman, facilitating the purchase and sale of shares.

Importance of Stock Market

- Stock markets help companies to raise capital.
- It helps generate personal wealth.
- Stock markets serve as an indicator of the state of the economy.
- It is a widely used source for people to invest money in companies with high growth potential.

Stock Price Prediction with Machine Learning assists you in determining the future value of a company's stock and other financial assets traded on an exchange. The entire point of predicting stock prices is to make large profits. It is difficult to predict how the stock market will perform.

The algorithm was developed from the scratch, Machine learning algorithm used are linear regression, lasso, Ridge

Objectives

- Implementation of stock prediction using Linear Regression, Lasso using tesla dataset
- Analysing Prediction
- Difference b/w Regression

Dataset (tesla stock price)

In dataset it consist of date for knowing data for particular day. Open and close colomns for knowing the opening price of stock and closing price of stock, And Adjclose (Adjusted close price) This makes closing price is the raw price, which is just the cash value of the last transacted price before the market closes. The adjusted closing price factors in anything that might affect the stock price after the market closes. Volume represent the number of shares sold/bought by the closing day

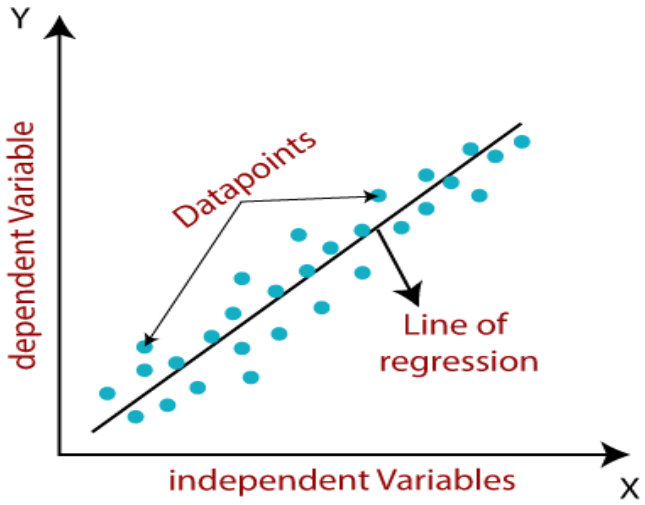
Date	Open	High	Low	Close	Adj Close	Volume
########	19	25	17.54	23.89	23.89	18766300
########	25.79	30.42	23.3	23.83	23.83	17187100
########	25	25.92	20.27	21.96	21.96	8218800
########	23	23.1	18.71	19.2	19.2	5139800
########	20	20	15.83	16.11	16.11	6866900
########	16.4	16.63	14.98	15.8	15.8	6921700
########	16.14	17.52	15.57	17.46	17.46	7711400
########	17.58	17.9	16.55	17.4	17.4	4050600
########	17.95	18.07	17	17.05	17.05	2202500
########	17.39	18.64	16.9	18.14	18.14	2680100
########	17.94	20.15	17.76	19.84	19.84	4195200
########	19.94	21.5	19	19.89	19.89	3739800
########	20.7	21.3	20.05	20.64	20.64	2621300
########	21.37	22.25	20.92	21.91	21.91	2486500
########	21.85	21.85	20.05	20.3	20.3	1825300
########	20.66	20.9	19.5	20.22	20.22	1252500
########	20.5	21.25	20.37	21	21	957800
########	21.19	21.56	21.06	21.29	21.29	653600

(i) Linear Regression

Linear regression is used in business, science, and nearly every other field where predictions and forecasting are important. It aids in the identification of relationships between one or more independent variables and a dependent variable. A feature is used to predict an outcome in simple linear regression. That is exactly what we will do here.

Forecasting the stock market is an appealing application of linear regression. These analyses can now be implemented in a few lines of code using modern machine learning packages such as scikit-learn.

As simple as these analyses are to implement, choosing features with sufficient predictive power to generate a profit is more of an art than a science. We'll look at how to easily add common technical indicators to our data to use as features in training our model during training. Let's take this step by step, beginning with gathering our pricing data.



The simplest form of the regression equation with one dependent and one independent variable.

$$y = m * x + b$$

y = estimated dependent value.
b = constant or bias.
m = regression coefficient or slope.
x = value of the independent variable.

Loss functions for regression

Regression involves predicting a specific value that is continuous in nature. Estimating the price of a house or predicting stock prices are examples of regression because one works towards building a model that would predict a real-valued quantity.

Let's take a look at some loss functions which can be used for regression problems and try to draw comparisons among them.

Mean Absolute Error (MAE)

Mean Absolute Error (also called L1 loss) is one of the most simple yet robust loss functions used for regression models.

Regression problems may have variables that are not strictly Gaussian in nature due to the presence of outliers (values that are very different from the rest of the data). Mean Absolute Error would be an ideal option in such cases because it does not take into account the direction of the outliers (unrealistically high positive or negative values).As the name suggests, MAE takes the average sum of the absolute differences between the actual and the predicted values. For a data point x_i and its predicted value y_i, n being the total number of data points in the dataset, the mean absolute error is defined as:

$$\mathbf{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Mean Squared Error (MSE)

Mean Squared Error (also called L2 loss) is almost every data scientist's preference when it comes to loss functions for regression. This is because most variables can be modeled into a Gaussian distribution.

Mean Squared Error is the average of the squared differences between the actual and the predicted values. For a data point Y_i and its predicted value $\hat{Y}_i$, where n is the total number of data points in the dataset, the mean squared error is defined as:

$$\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Optimization Algorithm

Optimization algorithms are used to find the optimal set of parameters given a training dataset that minimizes the loss function, in our case we need to find the optimal value of slope (m) and constant (b).

One such Algorithm is Gradient Descent.

Gradient descent is by far the most popular optimization algorithm used in machine learning.

Using gradient descent we iteratively calculate the gradients of the loss function with respect to the parameters and keep on updating the parameters till we reach the local minima.

Implementing Gradient Descent

Let's try applying gradient descent to **m** and **c** and approach it step by step:

Initially let m = 0 and c = 0. Let L be our learning rate. This controls how much the value of **m** changes with each step. L could be a small value like 0.0001 for good accuracy.

Calculate the partial derivative of the loss function with respect to m, and plug in the current values of x, y, m and c in it to obtain the derivative value **D**.

$$D_m = \frac{1}{n}\sum_{i=0}^{n} 2(y_i - (mx_i + c))(-x_i)$$

$$D_m = \frac{-2}{n}\sum_{i=0}^{n} x_i(y_i - \bar{y}_i)$$

Derivative with respect to m

D_m is the value of the partial derivative with respect to **m**. Similarly lets find the partial derivative with respect to **c**, Dc :

$$D_c = \frac{-2}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)$$

Derivative with respect to c

Now we update the current value of **m** and **c** using the following equation:

$$m = m - L \times D_m$$

$$c = c - L \times D_c$$

We repeat this process until our loss function is a very small value or ideally 0 (which means 0 error or 100% accuracy). The value of **m** and **c** that we are left with now will be the optimum values.

Now going back to our analogy, **m** can be considered the current position of the person. **D** is equivalent to the steepness of the slope and **L** can be the speed with which he moves. Now the new value of **m** that we calculate using the above equation will be his next position, and **L×D** will be the size of the steps he will take. When the slope is more steep (**D** is more) he takes longer steps and when it is less steep (**D** is less), he takes smaller steps. Finally he arrives at the bottom of the valley which corresponds to our loss = 0.

Now with the optimum value of **m** and **c** our model is ready to make predictions

(ii) Lasso

In statistics and Machine Learning Lasso(Least absolute shrinkage and selection operator) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

Lasso Regression uses L1 regularization technique. It is used when we have more number of features because it automatically performs feature selection.

L1 regularization

If a regression model uses the L1 Regularization technique, then it is called Lasso Regression. If it used the L2 regularization technique, it's called Ridge Regression.

L1 regularization adds a penalty that is equal to the absolute value of the magnitude of the coefficient. This regularization type can result in sparse models with few coefficients. Some coefficients might become zero and get eliminated from the model. Larger penalties result in coefficient values that are closer to zero (ideal for producing simpler models).

Mathematical Equation
Residual Sum of Squares + λ * (Sum of the absolute value of the magnitude of coefficients)

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

(iii) Ridge

Prerequisites:

Linear Regression

Gradient Descent

Ridge Regression (or L2 Regularization) is a variation of Linear Regression. In Linear Regression, it minimizes the Residual Sum of Squares (or RSS or cost function) to fit the training examples perfectly as possible. The cost function is also represented by J.

Cost Function for Linear Regression:

$$\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - h\left(x^{(i)}\right)\right)^2$$

Here, h(x(i)) represents the hypothetical function for prediction. y(i) represents the value of target variable for ith example.

m is the total number of training examples in the given dataset.

Linear regression treats all the features equally and finds unbiased weights to minimizes the cost function. This could arise the problem of overfitting (or a model fails to perform well on new data). Linear Regression also can't deal with the collinear data (collinearity refers to the event when the features are highly correlated). In short, Linear Regression is a model with high variance. So, Ridge Regression comes for the rescue. In Ridge Regression, there is an addition of l2 penalty (square of the magnitude of weights) in the cost function of Linear Regression. This is done so that the model does not overfit the data. The Modified cost function for Ridge Regression is given below

$$\frac{1}{m}\left[\sum_{i=1}^{m}\left(y^{(i)} - h\left(x^{(i)}\right)\right)^2 + \lambda \sum_{j=1}^{n} w_j^2\right]$$

Here, wj represents the weight for jth feature.

n is the number of features in the dataset.

Mathematical Intuition:

During gradient descent optimization of its cost function, added l2 penalty term leads to reduces the weights of the model to zero or close to zero. Due to the penalization of weights, our hypothesis gets simpler, more generalized, and less prone to overfitting. All weights are reduced by the same factor lambda. We can control the strength of regularization by hyperparameter lambda.

Different cases for tuning values of lambda.

If lambda is set to be 0, Ridge Regression equals Linear Regression

If lambda is set to be infinity, all weights are shrunk to zero.

So, we should set lambda somewhere in between 0 and infinity.

Results

Regression models and actual, predicted value

	Actual	Predicted
0	214.020004	212.283167
1	28.580000	28.767654
2	24.879999	24.943301
3	348.619995	336.121338
4	200.899994	203.164383
..	...	...
479	192.699997	192.894362
480	363.299988	363.120997
481	27.490000	27.573598
482	186.580002	188.408598
483	32.200001	32.397084

[484 rows x 2 columns]
Mean Absolute Error: 1.2604825865214917
Mean Squared Error: 4.401987987715675
Root Mean Squared Error: 2.0980915108058737

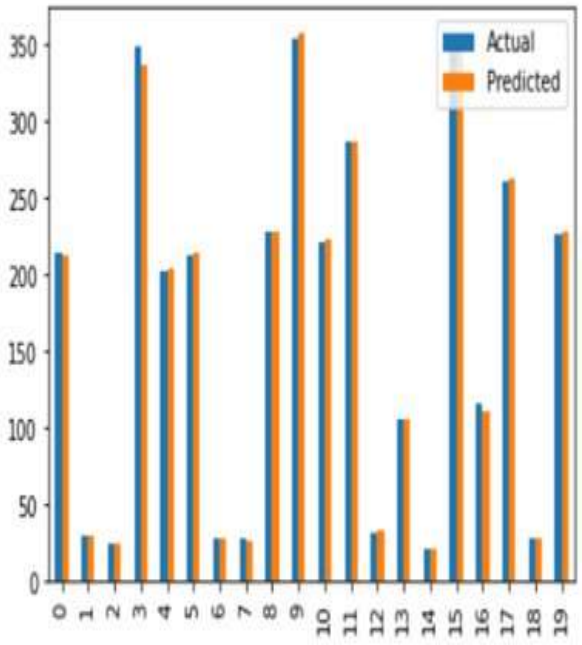


Fig. **Linear Regression**

```
        Actual   Predicted
0    214.020004  212.346722
1     28.580000   28.789360
2     24.879999   24.969353
3    348.619995  336.801723
4    200.899994  202.979873
..          ...         ...
479  192.699997  192.784202
480  363.299988  362.891586
481   27.490000   27.623724
482  186.580002  188.267238
483   32.200001   32.425411

[484 rows x 2 columns]
Mean Absolute Error: 1.138661977069429
Mean Squared  Error: 3.5576012910836377
Root Mean Squared Error: 1.8861604627082071
```

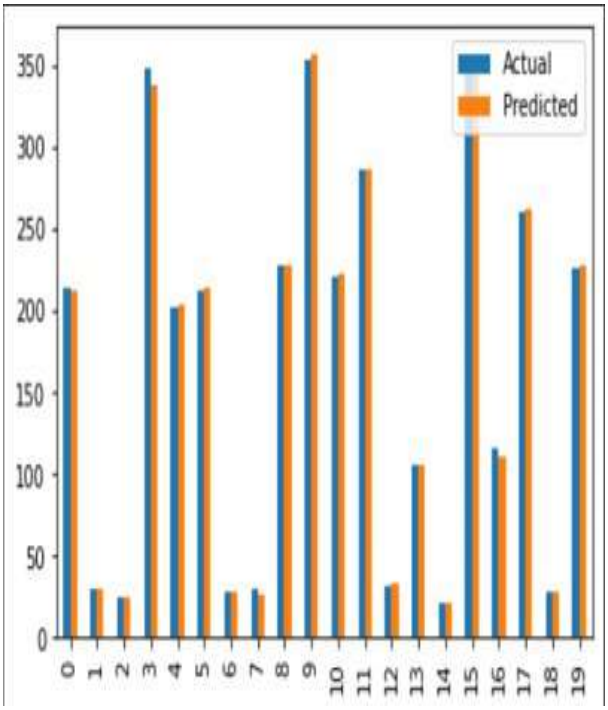


Fig. **Lasso**

```
        Actual   Predicted
0    214.020004  212.000398
1     28.580000   30.825819
2     24.879999   27.050105
3    348.619995  334.235915
4    200.899994  203.008032
..          ...         ...
479  192.699997  192.866195
480  363.299988  360.928445
481   27.490000   29.645737
482  186.580002  188.438507
483   32.200001   34.408693

[484 rows x 2 columns]
Mean Absolute Error: 1.8194856269015536
Mean Squared  Error: 6.828898858846586
Root Mean Squared Error: 2.6132161906062397
```

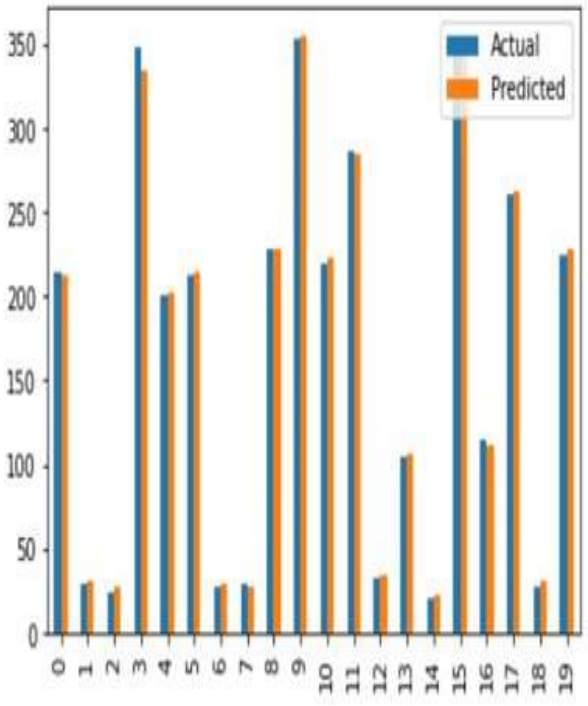


Fig. **Ridge**

Conclusion :-

We have successfully implemented & predicted tesla stock price by taking dataset with Machine learning algorithm regression models with high accuracy. Implemented from scratch

Future Scope :-

Implementing some more machine learning algorithms to check whether it predict best results. Developing GUI, Website and app application

Bibliography

[1] Kumari, Khushbu & Yadav, Suniti. (2018). Linear regression analysis study. Journal of the Practice of Cardiovascular Sciences. 4. 33. 10.4103/jpcs.jpcs_8_18.

[2] Kaya Uyanık, Gülden & Güler, Neşe. (2013). A Study on Multiple Linear Regression Analysis. Procedia - Social and Behavioral Sciences. 106. 234–240. 10.1016/j.sbspro.2013.12.027.

[3] Sunthornjittanon, Supichaya, "Linear Regression Analysis on Net Income of an Agrochemical Company in Thailand" (2015). University Honors.