Tesla Stock Price Prediction Using Machine Learning

Abstract

In this project, Tesla stock prices (Time series data) are predicted using machine learning techniques. For binary prediction of stock price whether price will go higher or lower was done with help of Random Forest Classifier but as there was very low accuracy and precision to avoid that problem back testing is done to improve precision and avoid data leakage. XG-boost model is implemented to visualize difference between original stock price and predicted stock price and it shows better performance in handling non-linear patterns. After that feature scaling is done for Long Short Term Memory (LSTM) then sequential model is initialized and complex dependencies are identified. Lasso regression is also applied both without and with Principal Component Analysis (PCA) and result is compared at end.

I. Dataset and Data preprocessing:

Tesla's stock data is downloaded from yfinance with maximum available history of Tesla's stock data with a size of 3632 rows × 7 columns.

In the data cleaning process, columns named Dividends and Stock Splits are deleted from the dataset as they have no values present and can affect data integrity and lower model performance.

In data visualization first, we will plot Close price over Time to see the ongoing pattern of Tesla stock price. As shown in Figure-1, it is clearly noticeable that stock prices have increased after 2020.

II. Feature Engineering

A new column, Tomorrow is added to store the next day's closing price also another column Target was added with binary labels (0 or 1) showing next day's price is higher (1) or lower (0) when compared to today's price.

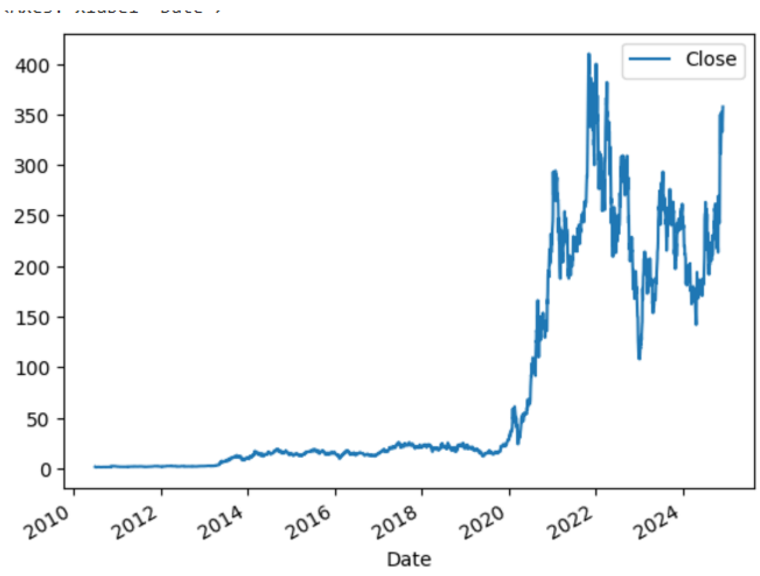
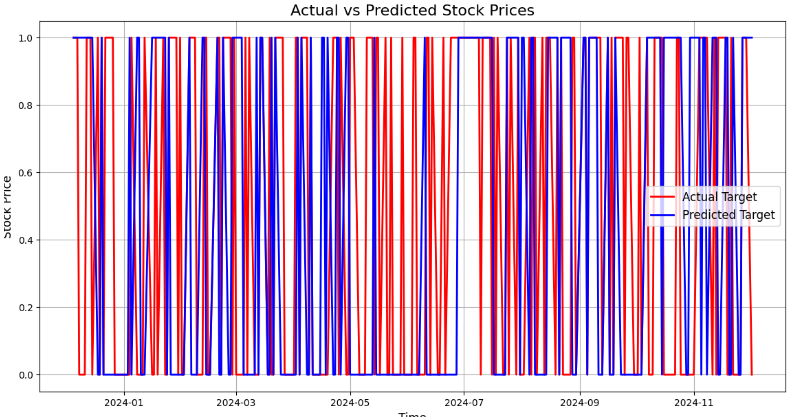


Fig. 1. Close price vs Time

III. Random Forest Classifier

Now initializing random forest classifier where n_estimator = number of decision tree as we want to improve accuracy and min_sample_split is introduced to protect against overfitting.

Considering all of the rows except last 250 rows in training and consider last 250 rows in test this will work best in time series. Here we will receive predicted target coloumn and then we will compare predicted target with actual target and got result as shown in Figure-2.



performance metrics and confusion matrix for predicted label vs actual label

performance metrics	**Values**
Precision Score	0.531
Accuracy	0.52
Confusion Matrix	
True Positive (TP)	False Positive (FP)
72	52
False Negative (FN)	True Negative (TN)
67	59

Backtesting

Aa seen above precision score is very low it can be improved by doing back test and adding additional predictors.Sliding window backtesting with a step of 150 was used to validate the model over different time periods.

Results after doing back testing are shown here there is significant increment in precision seen :

performance metrics	**Values**
Precision Score	0.67
Accuracy	0.52
Confusion Matrix	
True Positive (TP)	False Positive (FP)
56	5
False Negative (FN)	True Negative (TN)
60	10

Line graph after performing backtesting for predicted label vs actual label Figure-3 :

IV. Long Short Term Memory (LSTM)

LSTM is part of recurrent neural network (RNN) architecture mainly designed to handle sequential data like stock prices and it is one of the best for time series predcition .Multiple LSTM layers are present with 50 units each and also dropout layer is used for regularization then we will consider close column as training set and covert it to numeric form and also rescale data and keep values between 0 and 1 for better performance.Here last 60 days closing price is used to predict price of coming day.

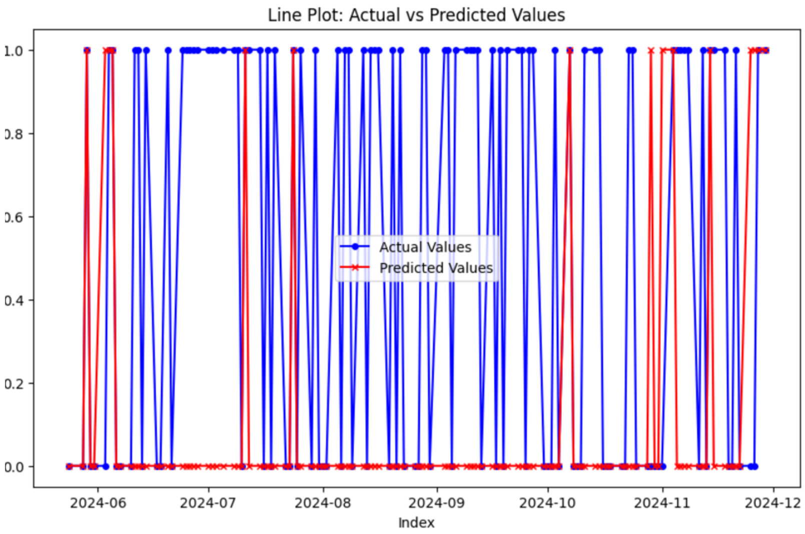


Fig. 3. Actual Label vs Predicted label

Sequential Model:

In sequential model optimizer is adam and loss function is mean_sqaure_error. Here model is trained for 20 epochs with a batch size of 32. Then as shown in Figure-4 model loss is as curve goes down model is learning parameter and perform well

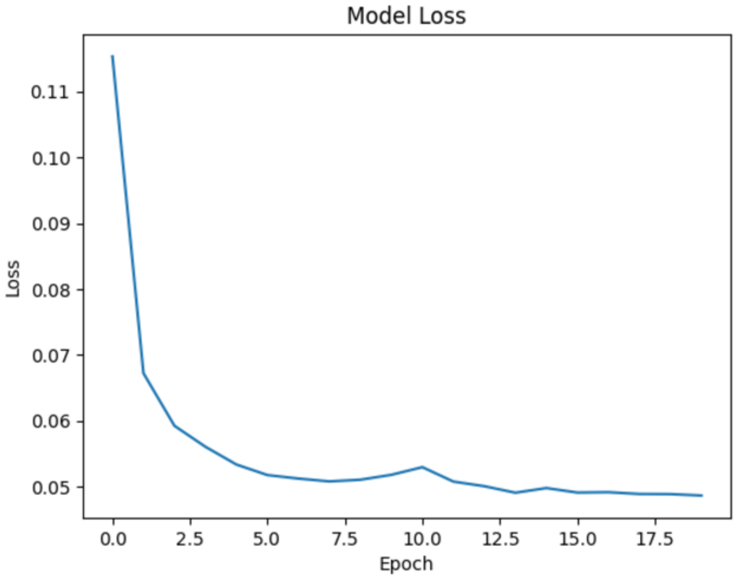


Fig. 4. Model loss

scaler.inverse_transform is used to get desidered predicted price and then actual price is compared with predicted price as shown in Figure-5 and performance metrics is as :

Metric	Value
Precision and Accuracy	
Accuracy	61.05%
Precision	61.05%
Error Metrics	
Mean Absolute Error (MAE)	11.97
Root Mean Squared Error (RMSE)	17.51
R² Score	0.85

TABLE I

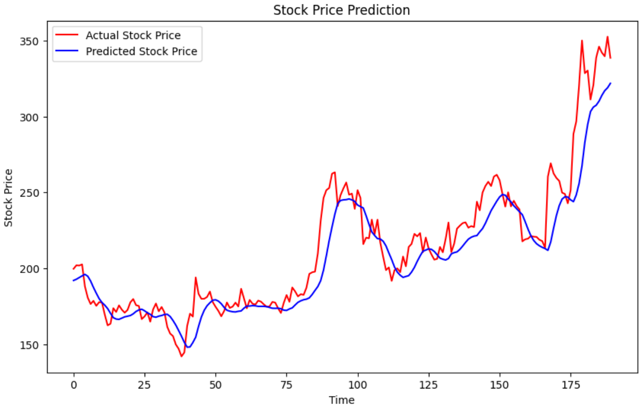


Fig. 5. LSTM

V. Gradient Boosting Approach: XGBoost

XGBoost is a powerful tree-based method which can handle non-linear relationships ver well and also give robust predictions for structured time-series data.Here also as similar as LSTM data of last 60 days is used to predict value of next day. parameters used here are n_estimators=100, learnin_rate=0.05, max_depth=6.

here training is done on scaled data and prediction is done on test data finally comparing predicted value with actual value and got result as shown is Figure-6 and precision , accuracy is as shown here in Table 2 .

Precision and Accuracy	
Accuracy	81.48%
Precision	81.48%
Error Metrics	
Mean Squared Error (MSE)	118.61
Mean Absolute Error (MAE)	7.28
R² Score	0.94

TABLE II
Model Evaluation Metrics

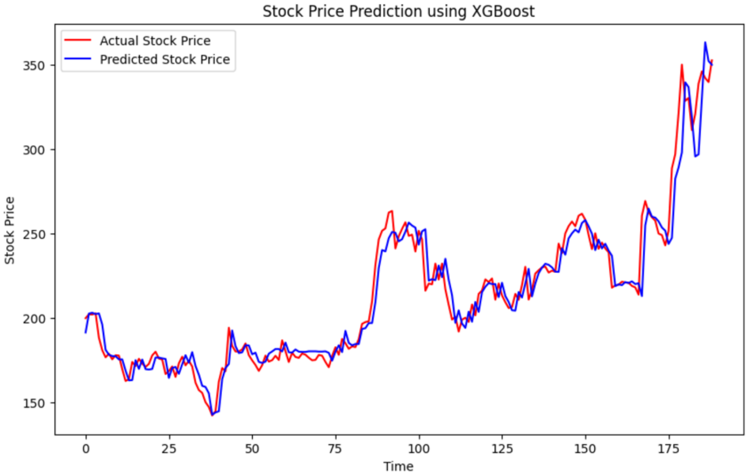


Fig. 6. XG-boost

VI. Lasso Regression (Without PCA)

: Lasso regression is applied using Lasso(alpha=0.001) it is a type of linear regression that applies regularization here firstly model is trained on flattend data and then predictions are made here predicted values are then rescaled to the original scale using scaler.inverse_transform as data was normalized using MinMaxScale.Actual value vs predicted value is plotted as shown in figure 7 and performance metrics is as shown here in table 3 :

Accuracy and Precision	
Accuracy	82.54%
Precision	82.54%
Error Metrics	
Mean Squared Error (MSE)	99.21
Mean Absolute Error (MAE)	6.73
R² Score	0.95

TABLE III
Model Evaluation Metrics for Lasso Regression

VII. Lasso regression +PCA :

PCA(Prinicipal component) is applied to reduce the dimensionality of the dataset it basically tries to

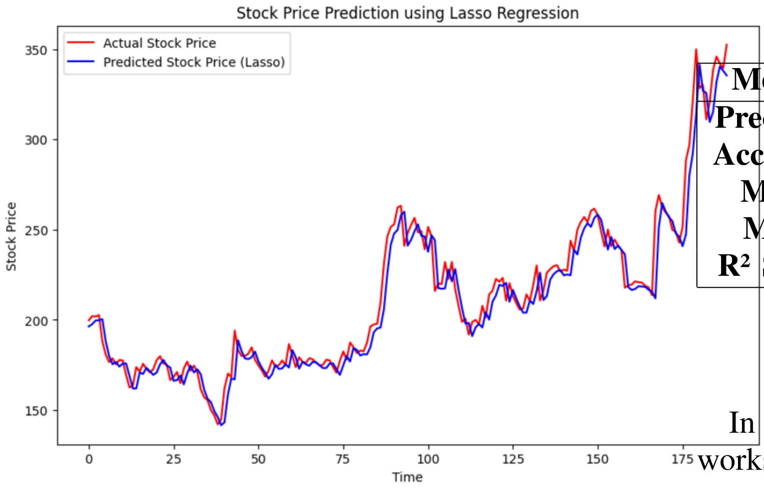


Fig. 7. Lasso without PCA

retain the maximum information . here considering that 95 percentage of variance should be retrained and lasso regression model is used on reduced-dimension data and used for predictions on the PCA-transformed test data . Result of predicted value vs actual value is shown in figure-8 and precision and accuracy is

PCA + Lasso	
Accuracy	39.68%
Precision	39.68%

TABLE IV
PCA + LASSO MODEL EVALUATION METRICS

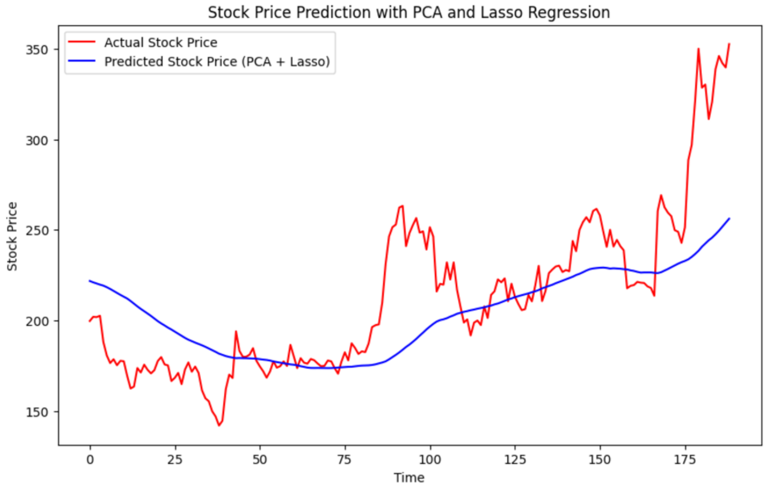


Fig. 8. Lasso with PCA

VIII. RESULT AND CONCLUSION:

Metric	Random Forest	LSTM	XGBoost	Lasso
Precision	0.67	61.05	81.48	82.54
Accuracy	0.52	61.05	81.48	82.54
MAE	N/A	5.12	17.51	6.73
MSE	N/A	8.43	11.971	99.21
R² Score	N/A	0.85	0.94	0.95

TABLE V
RESULT

In conclusion Lasso regression without PDA works best in prediction model