A Course end Project report on

# Stock Price Predictor using Random Forest

**A6413-Fundamentals of Machine Learning**

Submitted by Batch

|  |  |  |
| --- | --- | --- |
| S. No | Roll Number | Name of the student |
| 1 | 20881A0477 | G.Yuvaraj |
| 2. | 20881A04C0 | V.Vishnesh |

**Course Instructor**

**Mrs. S.Sujana**

**Assistant Professor, Dept of ECE**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING** VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

Autonomous institute, affiliated to JNTUH

**2022 – 23**

**🡪Contents**

* Abstract
* Introduction
* Proposed Method
* Results
* Conclusion
* References
* Code

**1.Abstract**

Stock Market Prediction provides wide area of Research which is revenue for the country and useful for day-to-day analysis. A lot of financial specialists performs predictive Analysis and finds the Market trends for their business. Analysts uses this Prediction for long time because of its unpredictable, composite, and consistently varying in character which was very hard to build solid expectations. This work proposes a methodology towards the expectation of pattern matching using AI methods like Random Forest and Support Vector Machine (SVM). The Random Forest method is a group learning strategy which is an extremely effective method for order & relapse. Support vector mechanism is an AI representation for the order and this model is generally utilized for arrangement. These procedures are utilized to determine whether the cost of stock will be higher than its cost on a given day to make profitable trading strategies. The main aim of this work is to create and evaluate a stock price predictor to make profitable trading strategies using AI and Data mining approaches.

**2. Introduction**

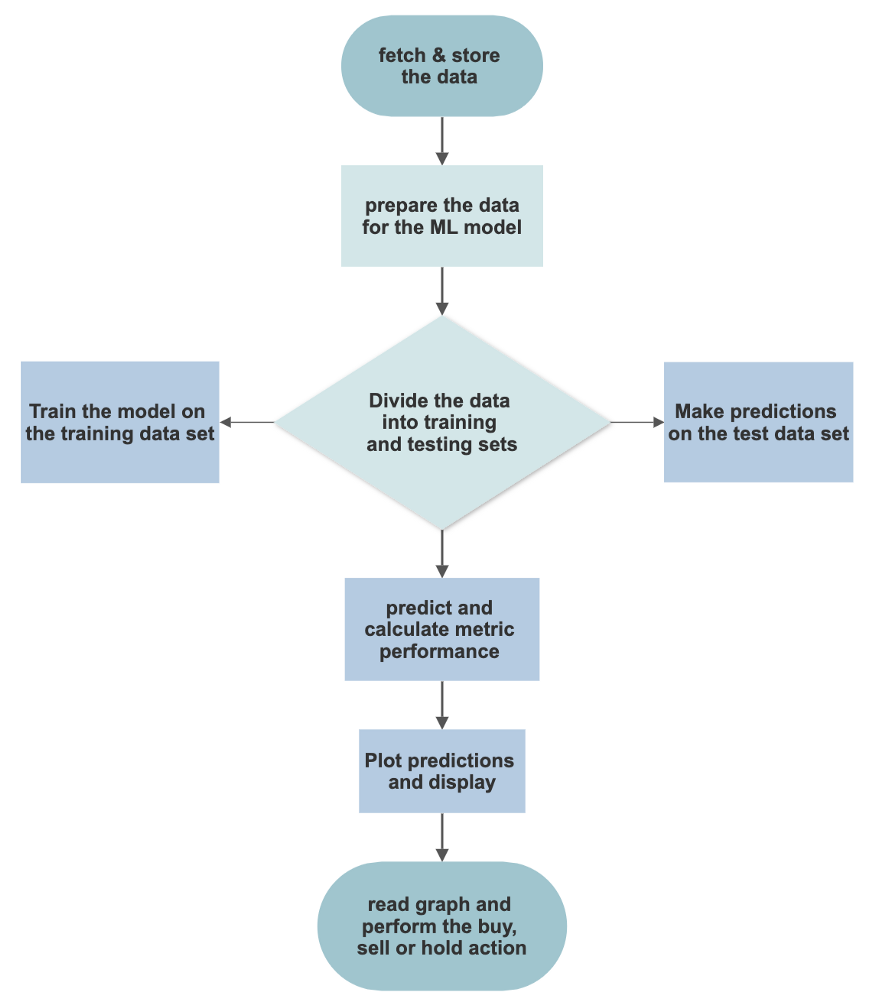
My interest in machine learning, data analytics, and the stock market led me to choose this topic for my final year capstone project. I focused this project on one of many applications of machine learning in finance. The financial industry is growing and looking for ways to use machine learning because it has created efficient services that add value to the customers and clients with financial investments.

**Stock price prediction** is the act of forecasting stock prices based on historical data. I used **historical data** in machine learning to recognize trends and understand the current market**. Machine learning**automates the trading process by using statistical models to draw insights and make predictions. Machine learning can collect and test a large amount of data, both structured and unstructured. It can apply suitable algorithms, transform, search for patterns, and make decisions based on the new data.

I have done many works to apply the machine learning approach in pattern analyzing and prediction. Because of the rise and importance of machine learning in the industry, this is promising. Predicting accurate stock prices is challenging because of the nature of the financial stock market, which includes the current trends, politics, and the economy. They create a heavy impact on the prices by making it hard to decide on buying, selling, or holding stock. **Therefore, risks are inevitable and can only be reduced.**

This project shows how various efforts have been taken to apply machine learning in stock forecasting for the **S&P500 index**. I implemented it in python with open-source libraries. I got the historical data from yahoo finance and applied pre-processing methods to make the data relevant. Also, using the randomized grid search cross-validation, a hyper tuning process to validate the model for building, fitting, and training for prediction. After prediction, error analysis is crucial for identifying how the model performs and how accurate the predicted values are.

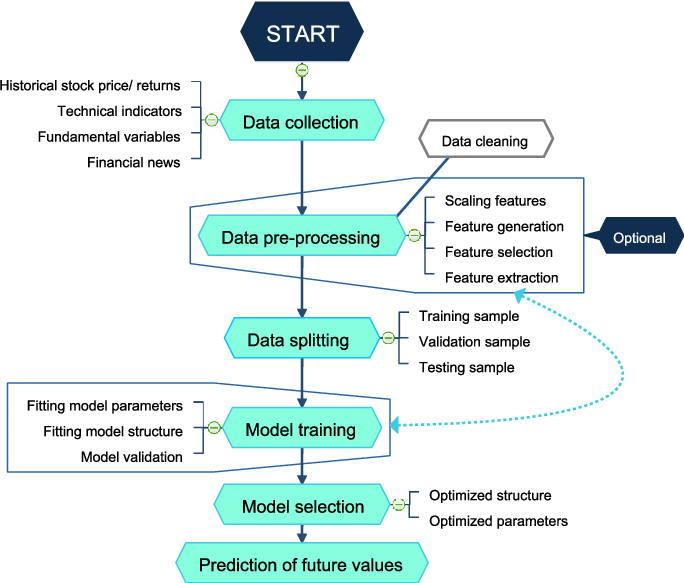
The**random forest regression model**is used for prediction. This will predict the low and high values of the next trading days, which includes the future prices for the next five days, one month, and one year of the S&P500. The outcome of buying, selling, or holding a stock will be based on the predicted values. **The objective of this project is data collection, data processing, and building the trading algorithm for prediction.**



**Fig 1.proposed method to predict stock price**

**3.Proposed Method**

The random forest regression model is used for prediction. This will predict the low and high values of the next trading days, which includes the future prices for the next five days ,one month ,and one year of the s&P500.The outcome of buying ,selling or holding a stock will be based on the predicted values



**fig 2.Block diagram of algorithm**

More trees usually means higher accuracy at the cost of slower learning .If you wish to speed up your random forest ,lower the number of estimations .If you want to increase the accuracy of your model ,increase the number of trees.

Use random forest regression to model your operations .For example ,you can input your investment data(advertisements, sales materials, cost of hours worked on long term enterprise deals, etc.) and your revenue data ,and random forest will discover the connection between the input and output .This connection can be used to predict how much revenue you will generate based on the growth activity that you pick and how much you are willing to spend on it

### **Methodology**

**1.Data collection**: Collecting data from the source. Random Forest Algorithm Random forest algorithm is being used for the stock market prediction. Since it has been termed as one of the easiest to use and flexible machine learning algorithm, it gives good accuracy in the prediction. This is usually used in the classification tasks. Because of the high volatility in the stock market, the task of predicting is quite challenging. In stock market prediction we are using random forest classifier which has the same hyper parameters as of a decision tree.

Feature are columns in the darkest .Feature selection is one of the essential concepts of machine Learning applications that have a tremendous impact on the model’s performance .In feature selection, not every column is necessary .These features selected have an impact and contribute to the prediction output .Unnecessary features decrease the general performance of the test data.

**2.Feature selection**:

Feature are columns in the darkest .Feature selection is one of the essential concepts of machine Learning applications that have a tremendous impact on the model’s performance .In feature selection, not every column is necessary.These features selected have an impact and contribute to the prediction output.Unnecessary features decrease the general performance of the test data.

**3.Data Pre-processing:**

The step is the most important part of this project.Data preprocessing are steps taken to make the data ready for the machine learning model.Preprocessing includes transforming the raw data into a format that the model can take from and operate on.

**4.Divide into train and test datasets:**

**Training set**: is a subset of the dataset used to build and fit predictive models.A training set is generated by building a training dataset script,which generates the trainingset features from the input options and the raw stock price data.The data is fed into the model for training .

**Testing data**: is a subset of the data set to assess the likely future performance of amodel . It is good standard for evaluating the model that is trained.This is called data standardization .Sklearn has a function called standard scaler that is used for standardizing the dataset.Standardisation is known to improve the numerical stability of the model and increase training speed.

**5.Scaling the features**

This is called data standardization .Sklearn has a function called standard scaler that is used for standardizing the dataset.Standardisation is known to improve the numerical stability of the model and increase training speed.

Created data frames with the predicted values for the next one year,one month.There are 252 trading days in a year,21 trading days in one month.I extracted the needed future days from predicted that held,341 days.

**4.Results**

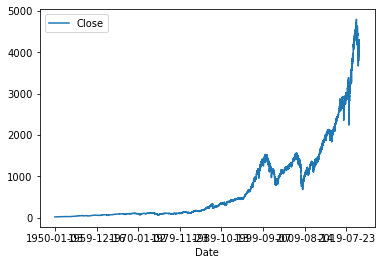


Fig3.Price of stock

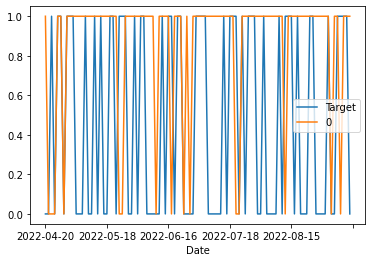


Fig4. Expected and predicted outputs

5.Observation :

* (8238rowsx7columns)

prediction score is : 0.47058823

# 6.Conclusion

There are different approaches to solving these problems. Their performance can differ from mathematical analysis for prediction to sentiment analysis, financial news articles, and expert reviews. However, because of how fickle the stock market is, there are no perfect or accurate prediction systems. The algorithm is also a great choice for anyone who needs to develop a model quickly. It provides a pretty good indicator of the importance it assigns to your features. Overall, random forest is mostly fast, simple, and flexible, but not without some limitations.

**7. References**

[1] Ashish Sharma, Dinesh Bhuriya, Upendra singh. “survey of exchange Prediction using Machine Learning Approach”,ICECA,2017.

[2] Loke .K.S.”impact of monetary Ratios And Technical Analysis On stock price

Prediction Using Random Forests”,IEEE,2017.

[3[ Xi zhang1,Siyu Qu1, Jieyum Huang1,Binxing Fang1,Philip Yu2,”stock Market

Prediction via Multi-source MultipleInstance Learning.” IEEE 2018.

[4] vivekanade, BhausahebDevikar , sayaliPhadatre , PranaliMunde , Shubhangi sonone.

“stock Prediction : Using Historical Data Analysis”,IJARCSSE 2017.

**8.Code**

import pandas as pd

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import precision\_score

sp500 = pd.read\_csv("C:/Users/vishnesh/OneDrive/Desktop/abcd/sp500.csv", index\_col=0)

sp500.plot.line(y="Close", use\_index=True)

del sp500["Dividends"]

del sp500["Stock Splits"]

sp500["Tomorrow"] = sp500["Close"].shift(-1)

sp500["Target"] = (sp500["Tomorrow"] > sp500["Close"]).astype(int)

sp500 = sp500.loc["1990-01-01":].copy()

print(sp500)

model = RandomForestClassifier(n\_estimators=100, min\_samples\_split=100, random\_state=1)

train = sp500.iloc[:-100]

test = sp500.iloc[-100:]

predictors = ["Close", "Volume", "Open", "High", "Low"]

model.fit(train[predictors], train["Target"])

preds = model.predict(test[predictors])

preds = pd.Series(preds, index=test.index)

print(‘prediction score is:’)

p=precision\_score(test["Target"], preds)

print(p)

**THANK YOU**