1. Project Title and Team Members

Feature Engineering and Machine learning for Stock Analysis and Prediction

Group 1: Paul Phillips, Qi Cai, Xin Gao

GitHub link at https://github.com/pcp0019/CSCE-5222-Group-1/

2. Goals and Objectives:

Our goal is to use feature engineering, data preprocessing, and machine learning together to predict stock price. We will download a dataset, adjust, add and remove features as needed, then use machine learning models to make predictions and test how salient these predictions are based on metrics such as r2 score.

Motivation

There are a number of factors that affect a stock's price, and these factors change every second. Finding the relationships between these factors is key to determining the value of a stock. This raises the question: how can we know which feature is more important than the other? How can we know how one feature affects another feature? What we need to do is to use feature engineering techniques to get this answer, and alter features as needed to improve predictive modeling. After we find out or get some patterns among them, then we may use these patterns and several regression machine learning models to apply to make stock predictions.

Significance

A stock's price changes often, and if we can use feature engineering to improve predictive models it will enable a better understanding about what goes into a stock's valuation and the factors that can cause this valuation to change. This knowledge can be useful for companies looking to improve their present or future stock value and prevent factors that will lead to valuation decline.

Objectives

Perform feature engineering techniques to examine and alter a feature set to create a better machine learning model for predicting stock price.

Features

The features we start with from our dataset are: High, Low, Open, Close, Volume, Adj Close. High represents the highest value of the stock for that day and Low is the lowest value. Close represents the value of the stock before trading ends for that day. Volume represents how much of that stock (represented by shares, not cost) was actually bought or sold in a trading day. Adj Close is similar to Close but it has some modifiers applied to it to take into account actions like dividend distribution. Different mathematical methods are then applied to these features to get new features for use in our dataset, such as H-L (High price minus Low price), and O-C (Open price minus Close price), and Daily Value Proportion.

3.Increment 1:

Related Work:

Sidra Mehtab, Jaydip Sen & Abhishek Dutta have used NIFTY 50 index values of the National Stock Exchange (NSE) of India and built eight regression models that predicted the open values of NIFTY 50 for the period December 31, 2018 till July 31, 2020. [1]

Rebwar M. Nabi and his research team collected data from NASDAQ and S&P, constructing new features and used the WEKA evaluation method. They finally proposed a new feature engineering approach for stock prediction.[2]

Jigar Patel and his research team used ANN, SVC, random forest and naïve-Bayes to predict direction of movement of stock price index for Indian stock markets by analyzing data like open, high, low & close prices. They found the random forest model showed a best performance than the other three.[3]

Tsai and Wang focus on combining ANN and decision trees to create a stock price forecasting model to predict Taiwan stock prices. The experimental result shows that the combined DT+ANN model has 77% accuracy, which is higher than the single ANN and DT models over the electronic industry. [4]

EK Ampomah, Z Qin, G Nyame compare the effectiveness of tree-based ensemble ML models (Random Forest (RF), XGBoost Classifier (XG), Bagging Classifier (BC), AdaBoost Classifier (Ada), Extra Trees

Classifier (ET), and Voting Classifier (VC)) in forecasting the direction of stock price movement. They use eight different stock data from three stock exchanges (NYSE, NASDAQ, and NSE). They found that AdaBoost Classifier has the best performance on their training and test dataset. [5]

Mojtaba Nabipour and his research team compare nine machine learning models and two l deep learning methods to reduce the risk of trend prediction by using stock price data from Tehran stock exchange. They found that RNN and LSTM outperform other prediction models with a considerable difference [6]

Dataset:

Our dataset was downloaded from Yahoo finance. Using Costco stock price from 2013-1-1 to 2023-11-1.

Detail design of Features:

We used feature engineering to calculate additional features to add to the original dataset. Those features are:

1) Daily Value Proportion, an original feature that is calculated by first calculating the 75^{th} percentile value and the 25^{th} percentile closing stock value. After that, we perform the calculation:

Daily Value Proportion = (Daily Close Price – 25^{th} percentile)/ (75^{th} percentile – 25^{th} percentile)

This calculation is performed for all the stock values in the entire timeframe of our dataset. The numerator's calculation is to keep the stock price in the range of -2 to 2, a form of normalization. The 75-25 percentile difference in the denominator was chosen to account for a stock's regular pattern of value across a period of time, without regard for the higher or lower extremes. In this way, Daily Value Proportion demonstrates how the daily stock price relates to its regular pattern of value, allowing it to become useful for predicting the overall price of the stock over time.

For example, if we assume the prices are 100, 110, 115, 120, 125, 130, 140, 150, 160, 170, the 75% percentile is 157.5, the 25% percentile is 115. If daily value is 170, then the Daily Value Proportion is (170 – 115)/(157.5 - 115) = 1.29, showing that it is higher at this point in time than the stock's regular value.

2) H–L is the High Low Difference, calculated as the stock's price at the high point of the day minus its price at the low point of the day. It will show the difference between the maximum and minimum value that could have been gotten from trading that stock in a certain day.

3) O-C is the Open Close Difference, calculated as the stock's daily opening price minus it's daily closing price. This shows how much value it had at the earliest tradable moment as compared to its last tradable moment for a given day.

With these three new features, machine learning models should be able to predict the stock prices more clearly.

```
percentile75 = octdf.Close.quantile(0.75) #obtains the 75th percentile value of the closing price
percentile25 = octdf.Close.quantile(0.25) #obtains the 25th percentile value of the closing price

#equation for added feature is (daily close price - percentile25)/(percentile75 - percentile 25)
dailyValueProportion = [] #list to become part of the feature set
highlowDifference = [] #list to become part of the feature set
opencloseDifference = [] #list to become part of the feature set

for i in range (0, len(octdf) ): #loop that takes in all the rows of closing values and calculates the Daily Value Proporti
  dailyClosePrice = octdf.iloc[i].Close
  calculation = (dailyClosePrice - percentile25)/(percentile75 - percentile25) #
  dailyValueProportion.append(calculation)

for i in range (0, len(octdf) ): #obtains the high - low daily price
  highPrice = octdf.iloc[i].High
  lowPrice = octdf.iloc[i].Low
  calculation = (highPrice - lowPrice)
  highlowDifference.append(calculation)

for i in range (0, len(octdf) ): #obtains the close - open daily price
  openPrice = octdf.iloc[i].Close
  closePrice = octdf.iloc[i].Open
  calculation = (openPrice - closePrice)
  opencloseDifference.append(calculation)

octdf['Daily_Value_Proportion'] = dailyValueProportion #adds the Daily Value Proportion to the dataframe as a feature
octdf['H-L'] = highlowDifference #adds the difference between the highest and lowest price for the day as a feature to our
octdf['O-C'] = opencloseDifference #adds the difference between the opening and closing price for the day as a feature to c
print(octdf)
```

As displayed in a table:

Date	Open	High	Low	Close	Adj Close	Volume	Daily_Value_Proportion	H-L	O-C
2013-01-02	100.599998	101.449997	100.209999	101.449997	82.717537	3153800	-0.207033	1.239998	0.849998
2013-01-03	102.110001	103.019997	101.760002	102.489998	83.565491	3872400	-0.202371	1.259995	0.379997
2013-01-04	102.550003	102.910004	101.550003	102.160004	83.296448	1989000	-0.203851	1.360001	-0.389999
2013-01-07	101.089996	101.730003	100.900002	101.370003	82.652313	1663900	-0.207392	0.830002	0.280006
2013-01-08	101.000000	101.790001	100.730003	101.180000	82.497391	2189900	-0.208244	1.059998	0.180000
...	...	...	...	...	...	...	...	...	...
2023-10-25	548.549988	553.830017	545.609985	549.989990	548.982483	1757500	1.803635	8.220032	1.440002
2023-10-26	549.650024	554.659973	545.530029	547.599976	546.596863	1925300	1.792922	9.129944	-2.050049
2023-10-27	547.599976	548.030029	540.229980	543.030029	542.035278	1503100	1.772436	7.800049	-4.569946
2023-10-30	545.739990	556.359985	543.640015	554.880005	553.863525	1696200	1.825556	12.719971	9.140015
2023-10-31	552.159973	554.030029	549.059998	552.440002	551.427979	1394700	1.814618	4.970032	0.280029

Other than feature additions, we also removed some features, the columns Adj Close and Volume. Adj Close was too similar to Close Price (which is our target) and volume was dropped after it was found to be skewing results.

Analysis, implementation and preliminary results:

We use this dataset to train different machine learning models. One of these models is K-Nearest Neighbor, a common machine learning technique that makes predictions based on identifying a certain number of close datapoints. The number of these datapoints is assigned by the user (the K value, set to 4 in this case). While this method can be used for classification, predicting the value of a stock is a regression problem so an average of values is from this number of datapoints is used to make the

regression prediction. R2 score will be the metric used to measure how well the model has performed this task. The screenshot below demonstrates results:

```
            Daily_Value_Proportion        H-L       O-C
Date
2013-01-02               -0.207033   1.239998  0.849998
2013-01-03               -0.202371   1.259995  0.379997
2013-01-04               -0.203851   1.360001 -0.389999
2013-01-07               -0.207392   0.830002  0.280006
2013-01-08               -0.208244   1.059998  0.180000
...                            ...        ...       ...
2023-10-25                1.803635   8.220032  1.440002
2023-10-26                1.792922   9.129944 -2.050049
2023-10-27                1.772436   7.800049 -4.569946
2023-10-30                1.825556  12.719971  9.140015
2023-10-31                1.814618   4.970032  0.280029

[2727 rows x 7 columns]
Using 4 neigbors, the r2score result is:
0.9998712984773843
```

The r2 score of 0.998 shows that the predictions were very close to the actual values.

Naïve Bayes is a very different model attempted for this dataset. It makes a prediction by calculating the most likely outcome from a Bayes Theorem probability calculation. However, since it assumes feature independence and our features have relations to one another, we do not expect it to yield very accurate results in this case. Since this model is used for classification and not regression, we create a new y set called rise or fall. We use the next day's close price to minus today's close price to know if the close price between these two days is up or down. Up will be recorded as 1 and down will be 0. A new y set is created full of 1s and 0s, and we remove the last row to remove NaN value.

With this feature as our new target, we trained and tested the Naïve Bayes model. The results can be seen in the screenshot below:

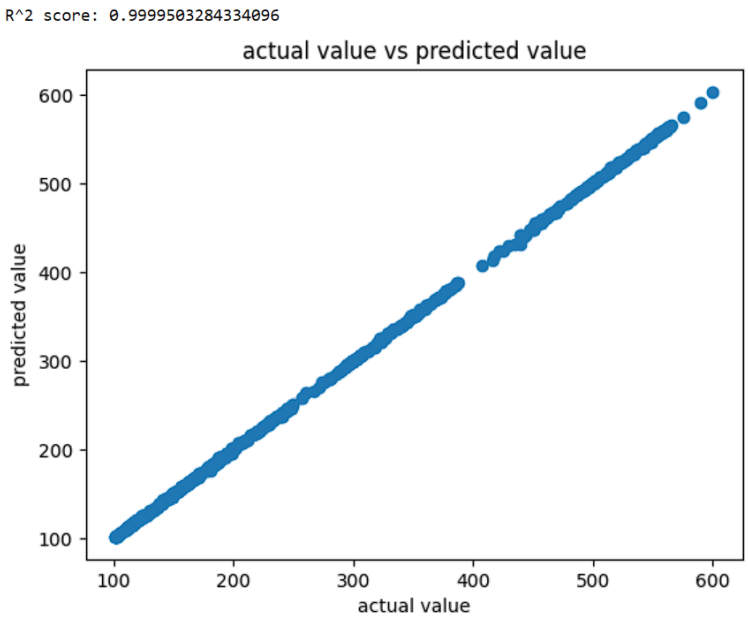
```
Accuracy: 0.510989010989011
              precision    recall  f1-score   support

           0       0.43      0.14      0.21       255
           1       0.53      0.84      0.65       291

    accuracy                           0.51       546
   macro avg       0.48      0.49      0.43       546
weighted avg       0.48      0.51      0.44       546
```

The accuracy of 0.51 was to be expected given the inherent assumptions of Naïve Bayes. The F1 score in predicting fall is 0.21, while in predicting rise it is 0.65, much higher. This difference is also shown in the recall value: 0.84 in predicting rise and 0.14 in predicting fall.

The last model we used was Gradient Boost Regressor. This method makes predictions based on decision trees, with a special focus on lowering the prediction error rate. Since the gradient boost method is being used for regression and not classification, the target will be the closing price, like in the K-Neighbors model. The data was trained on 70% of the data and tested on 30%, using decision trees of depths 3 and 100 estimators. The results are demonstrated in the graph screenshot below:



We also got a very high R2 score as 0.9999. The model showed a good linear relationship and the predicted price matched very closely with the actual price.

4.Project Management

Work completed:

Paul Phillips: Data preprocessing, Daily Value Proportion implementation, K-Nearest-Neighbors model implementation, PPT making, project draft report writing

33.3%

Xin Gao: Comment modification, Daily Value Proportion concept creation, idea generation, Naïve Bayes model implementation, project draft report writing

33.3%

Qi Cai: Comment modification, Daily Value Proportion concept creation, project planning, Gradient boost regression model implementation, project draft report writing

33.4%

Future Tasks:

To improve our performance, we plan to focus on two different areas of our project. One is to implement new methods of data processing such as rescaling or normalizing the data to reduce dataset imbalance. Using models that have been trained on normalized data may produce a different result than what have had previously. The other objective is to attempt different machine learning models on the dataset to discover how their performances differ from what we have demonstrated thus far.

5. References

[1] Mehtab, Sidra, Jaydip Sen, and Abhishek Dutta. "Stock price prediction using machine learning and LSTM-based deep learning models." Machine Learning and Metaheuristics Algorithms, and Applications: Second Symposium, SoMMA 2020, Chennai, India, October 14– 17, 2020, Revised Selected Papers 2. Springer Singapore, 2021. (Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models | SpringerLink)

[2] Nabi, Rebwar M., Soran AB Saeed, and Abdulrahman MW Abdi. "Feature Engineering for Stock Price Prediction." Int. J. Adv. Sci. Technol 29.12s (2020): 2486-2496.(https://www.researchgate.net/profile/Rebwar- Nabi/publication/342339410

[3] Patel, Jigar, et al. "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques." Expert systems with applications 42.1 (2015): 259-268.

[4] Tsai, Chih F., and Sammy P. Wang. "Stock price forecasting by hybrid machine learning techniques." Proceedings of the international multiconference of engineers and computer scientists. Vol. 1. No. 755. 2009.

[5] Ampomah, Ernest Kwame, Zhiguang Qin, and Gabriel Nyame. "Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement." Information 11.6 (2020): 332.

[6] Nabipour, Mojtaba, et al. "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis." IEEE Access 8 (2020): 150199-150212.