Stock Exchange Data: Predictive Analysis of Stock Market Index Movements

October 14, 2024

1 Introduction

1.1 Background & Context

The main objective of this analysis is to **forecast the closing prices of various stock market indices with the focus on if machine learning models can accurately predict the future closing prices of stock indices based on historical data.**
Stock market indices included in the dataset are used to measure the performance of a group of stocks. Predicting their future values can provide valuable insights. Previous studies have used various machine learning techniques to forecast stock prices, including linear regression, support vector machines, and neural networks [San].

1.2 Sub tasks

A standard data analysis sub-tasks are carried out.

1. **Data Exploration:** Thoroughly examine the dataset to understand its characteristics, including data types, range, and distribution of key variables.
2. **Data Cleaning and Preprocessing:** Address missing data, anomalies, and outliers to refine the dataset for accurate analysis.
3. **Feature Engineering:** Develop features that enhance the model's predictive power by encapsulating complex patterns or trends observed in the data.
4. **Model Selection and Implementation:** Evaluate several machine learning models to identify those most capable of capturing the dynamics of stock index movements.
5. **Model Evaluation and Refinement:** Assess model performance using appropriate metrics and refine models through tuning and cross-validation.

1.3 Context & Previous Work

Historically, methods such as ARIMA and linear regression have been extensively used for time series forecasting due to their simplicity and effectiveness in handling linear relationships and seasonal patterns. However, with the advent of big data and computational advancements, more complex models like Random Forest, XGBoost, and neural networks like LSTM have gained prominence. These models have shown potential in capturing nonlinear patterns and long-term dependencies that are characteristic of financial markets [Tad].

1.4 Objectives

This analysis extends the existing research by providing a systematic comparison of multiple models, evaluating their ability of providing accurate predictions, and discuss their practicality and implications, including data quality issues, model over fitting, as well as market volatility.

2 Methodology and Dataset

2.1 technical features of the dataset

It is observed that the dataset includes daily stock prices, volume of trades.

2.2 Summary of dataset Attributes

Region	Specifies the geographical location of the stock exchange.
Exchange	Specifies the name of the stock exchange.
Index	Specifies the unique code representing the index.
Currency	Specifies the currency in which the index values are measured.
Date	Specifies the date for the index data
Open	Specifies the opening value of the index on the given date.
High	Specifies the highest value of the index on the given date.
Low	Specifies the lowest value of the index on the given date.
Close	Specifies the closing value of the index on the given date.
Adj Close	Specifies the adjusted closing value of the index on the given date.
Volume	Specifies the trading volume of the index on the given date.
CloseUSD	Specifies the closing value of the index converted to USD.

2.3 Data Exploration

The distribution and the trends in the data is understood, missing values and outliers are identified, code shown in notebook.

indexProcessed.csv

1. No missing values in any columns.
2. The dataset includes 104,224 rows.
3. The Volume column contains zero values, which may need special handling.
4. The CloseUSD column appears to have a large range, indicating different index scales.

indexData.csv

1. 2,204 missing values in Open, High, Low, Close, Adj Close, and Volume columns.
2. The dataset includes 110,253 rows.
3. Similar to the processed dataset, Volume contains zero values, which may need special handling.

2.4 Data Cleaning

The following steps were taken:

- Handling Missing Values: rows with missing values are dropped
- Handling Zero Volumes: check if zero volumes means non-trading days or missing data
- Ensuring Consistency: consistent date format across dataset, and remove duplicate rows

Figure 1 illustrates the trend of stock price over time after data being cleaned.

2.5 Feature Engineering

Feature engineering is considered as creating additional features may improve model performance. The goal of capturing the patterns, trends in time series data is preserved by selecting lagged features.

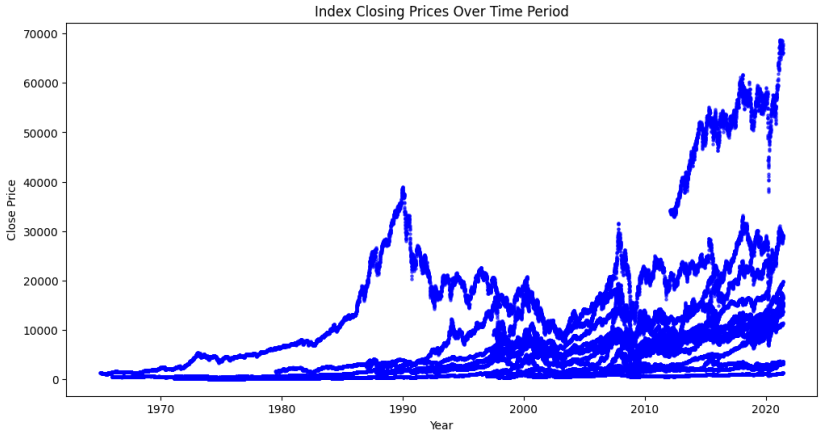

Figure 1: Stock Market Index Price over time.

2.5.1 Lag Features

A common approach in time series data analysis is creating lag features. Using past values of the indices as predictors, can provide significant insights for models. The purpose of lag features is to capture temporal dependencies in time series data. Histograms of each feature is plotted in the notebook, and figure 2 shows the correlation of each feature and closing price.

features	Purpose
Close_Lag_1	Captures the value of the time series from the previous time step. It helps the model to understand how the current value depends on the immediate past value.
Rolling_Mean_7	Captures the trend in the data by smoothing out short-term fluctuations and highlighting longer-term trends over a one-week period.
Rolling_Std_7	Measures the variability or volatility of the time series over a one-week period. It provides insights into how much the values deviate from the mean.

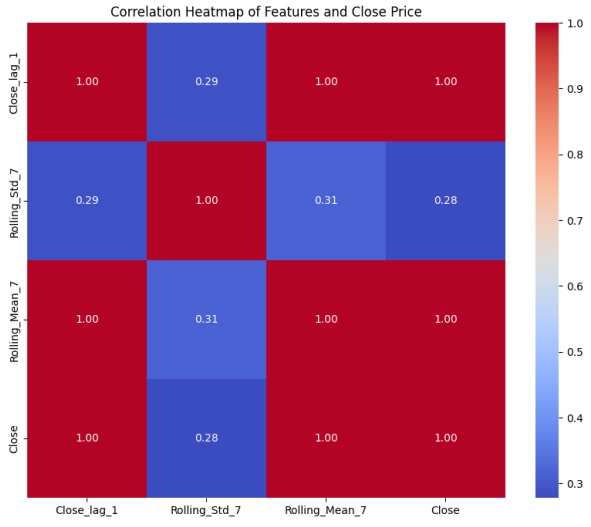

Figure 2: Feature Correlation Heatmap.

2.6 Modelling

Once the lag features are ready, different machine learning models were experimented. In this analysis, the 2 different models will be discussed, the two models will be going through the same steps, in order to have the results compared and discussed.

2.6.1 K-Nearest Neighbors

Using lag features for a K-Nearest Neighbors (KNN) model is a common approach, especially in time series forecasting or when the objective is to capture temporal dependencies in the data [Taj21]. Historical stock data is mapped into a set of vectors, each vectors represent N dimension for each stock features. Then Euclidean distance is calcupated to make a decision [ea13]. KNN model is applied after scaling lag features was done in the previous step. Model is trained as well as evaluated, followed by tuning and validation, this workflow helps to use lag features effectively in KNN model.
A comparison graph is produced to represent the performance of KNN model on the dataset. The visualisation helps to assess if the model's prediction aligns with the actual historical data, this is discussed later on in the analysis.

2.6.2 Support Vector Machines (SVM)

SVM has gained prominence for stock price prediction, hence this model is explored [Tri19]. Since SVMs are sensitive to the scale of the input data, the scaling of lag features are kept, this ensures that no particular feature dominates other due to its scale. Same steps were followed, model tuning and visual representation was produced, as well as Root Mean Squared errors calculated.

2.6.3 Linear Regression

Linear regression was also experimented as an additional option due to its simplicity and interpretability, it can be served as a foundation for more complicated modelling.

3 Results

In order to have fair comparison across all models applied, the same lag features ['Close_lag_1', 'Close_lag_7', 'Close_lag_30', 'Rolling_Mean_7', 'Rolling_Std_7', 'Daily_Change', 'Significant_High'] were kept consistent across all models. This ensures that any differences in model performance are due to the models themselves, and not variations in the data they're trained on [Bro].

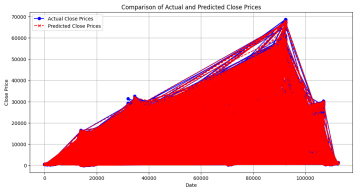

Figure 3: KNN

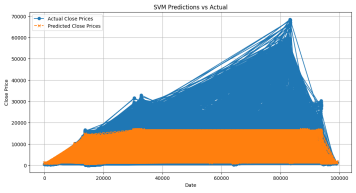

Figure 4: SVM

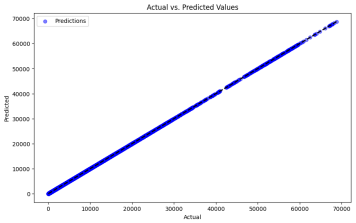

Figure 5: LR

Figure 3,4 and 5 show the comparison of actual closing price vs. Predicted across all models. Where the correlation of the general shape of the graph is consistent.

3.0.1 Hyperparameter

Each model was evaluated with hyper parameter - Grid Search specifically was applied to each model, where it tries every single combination of hyper-parameters provided in a grid.
To note, Grid Search on Linear Regression model application is slightly different, which have multiple hyper-parameters [Lee]. Linear Regression has few hyper-parameters to tune, especially if model is used without regularization, there is no hyper-parameter to tune. But with feature selection, Grid

Search can be applied to determine the best parameters.

K-Nearest Neighbours (KNN)	Support Vector Machine (SVM)	Linear Regression
n_neighbors	C	Ridge
weights	gamma	Lasso
metric	kernel	

Tuning for each model can be found in Tuning section in the notebook. Each model was trained to set a baseline, followed by training and evaluation using the parameters found from the grid search, lastly the performances was compared using RMSE and r^2. to see if there is an improvement. Below summary table illustrates the findings.

K-Nearest Neighbours (KNN)	**Support Vector Machine (SVM)**	**Linear Regression**
Before Tuning RMSE: 273.1390356451318 R^2 : 0.9990838249485683	Before Tuning RMSE: 42623345.315592684 R^2 : 0.4765701928185321	Before Tuning MSE: 4.27227355560098e-23 R^2 : 1.0
After Tuning RMSE: 197.74566821067202 R^2 : 0.9995197969931583	After Tuning	After Tuning RMSE 4.805262294592823e-18 R^2 : 1.0

4 Discussion

Feature engineering as a critical step in the modelling process, as the nature of the component during analysis, it can significantly enhance the performance of the model by introducing new information or distilling existing information into a more useful form [Mum]. Multiple features were considered and all fed into the model for training.

During Model Tuning, when tuning an SVM model particularly, using Grid Search, the process can be super time-consuming. For this particular dataset after cleaning and feature engineering, Grid Search evaluates 36 different combinations of SVM mparameters across 5 folds, resulting in a total of 180 fits. This extensive search is further complicated by the choice of parameters and kernels, such as 'rbf', 'linear', and 'poly', each requiring different computational efforts to assess. The training time in output suggest significant delays.

```
Fitting 5 folds for each of 36 candidates, totalling 180 fits
[CV] END .....................C=0.1, gamma=scale, kernel=rbf; total time= 2.7min
[CV] END .....................C=0.1, gamma=scale, kernel=rbf; total time= 2.8min
[CV] END .....................C=0.1, gamma=scale, kernel=rbf; total time= 2.7min
[CV] END .....................C=0.1, gamma=scale, kernel=rbf; total time= 2.7min
[CV] END .....................C=0.1, gamma=scale, kernel=rbf; total time= 2.9min
[CV] END ..................C=0.1, gamma=scale, kernel=linear; total time= 8.4min
```

4.0.1 Conclusion

In the data table showing results before and after tuning, it is observed that Linear Regression has an r^2 value of 1, it does indicate a perfect fit, however it does suggest overfitting, capturing both actual pattern but also noise in the training data. Whereas KNN model shows a r^2 value close to 1, accompanied by Figure 3, it indicates a good fit.

In this comparative analysis of K-Nearest Neighbors (KNN), Linear Regression, and Support Vector Machine (SVM) models for stock price prediction, the KNN model proves to be the most effective. This conclusion was based on the model's exemplary performance metrics, notably an R^2 value of 0.99 and an RMSE of 197. These metrics indicate that the KNN model not only explains 99% of the variance in stock prices from the dataset but also minimizes prediction errors significantly, and overall, Machine Learning model can accurately predict the stock prices.

References

[Bro] Jason Brownlee. Why do i get different results each time in machine learning?

[ea13] Khalid Alkhatib et al. Stock price prediction using k-nearest neighbor (knn) algorithm. *International Journal of Business, Humanities and Technology*, 3(3):1559–1574, 2013.

[Lee] Wei-Meng Lee. Tuning the hyperparameters of your machine learning model using gridsearchcv.

[Mum] Alhassan Mumuni. Automated data processing and feature engineering for deep learning and big data applications: A survey.

[San] DJ. Margaret Sangeetha. Financial stock market forecast using evaluated linear regression based machine learning technique.

[Tad] Yasin Tadayonrad. A new key performance indicator model for demand forecasting in inventory management considering supply chain reliability and seasonality.

[Taj21] S. et al. Tajmouati. Applying k-nearest neighbors to time series forecasting: two new approaches. *Journal of Forecasting*, 43(5):1559–1574, 2021.

[Tri19] Naliniprava Tripathy. Stock price prediction using support vector machine approach. *International Academic Conference on Mangement Economics*, 3(3):1559–1574, 2019.