Stock Prediction Project

Himang Chandra Garg, Nishil Agarwal, Vansh Aggarwal
Computer Science and Engineering, Computer Science and Artificial Intelligence, Computer Science and Engineering
Indraprastha Institute of Information Technology
Roll no.- 2022214, 2022334, 2022558
himang22214@iiitd.ac.in, nishil22334@iiitd.ac.in, vansh22558@iiitd.ac.in

***Abstract*—In this project our aim is to develop a predictive model for forecasting daily stock price movements of NIFTY50-listed companies. The model uses historical market data, including past prices, volume, volatility, and moving averages, to predict closing price. Accurate trend prediction enables investors and analysts to make informed decisions, maximizing profits, and managing risks by relying on reliable forecasts.**

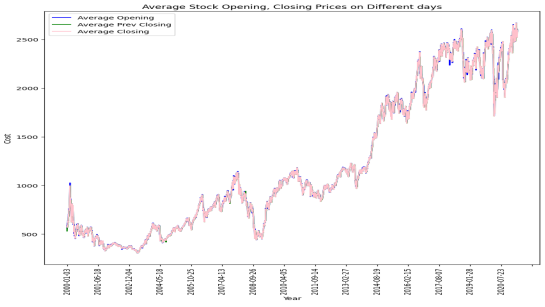
I. Dataset Description

For our project, we utilize a comprehensive dataset from Kaggle : https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data, comprising 235,192 entries detailing daily trading and price information for NIFTY50-listed stocks. Key columns include Symbol (company name), Prev Close, Open, High, Close prices, along with VWAP, Volume etc. The dataset also includes metadata for each company.

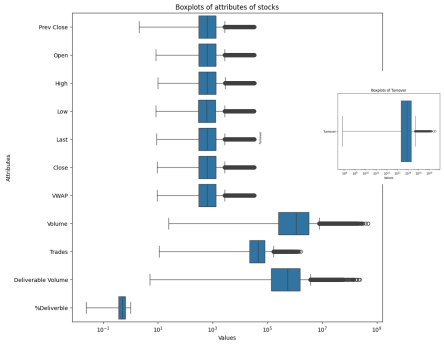
II. Exploratory Data Analysis

We plotted various plots using the data to gain insight on key features, identifying patterns or trends in data.

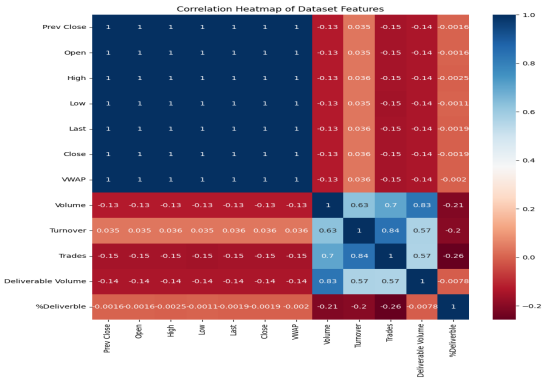
1) **Prices Trend Graph :** Close, Open, Prev Close prices of stocks sorted by date to identify patterns in the data.



2) **Box plots of features :** Box plots of all attributes to gain insight into range of spanned by attributes. Noticed outliers in the data above max whisker.



3) **Heatmap of features :** We plotted a divergent heatmap between all features. We notice that many features are closely correlated to each other evident by their correlation score. Some of these having high correlation might be removed to optimise computation.



4) **Plot of rolling averages :** We calculated 20 and 50 days rolling average for the stocks. This is the average of closing prices for last 20 or 50 days respectively. We obtained the following graphs which helped us generalize our insights and notice patterns in data.

III. Data Pre-processing and Challenges Faced

Data pre-processing was undertaken to prepare the dataset for modeling. This involved handling missing values, normalizing features, creation of new features, extraction of features from present features etc.

1) **Imputing missing values :** In our dataset some entries had missing Trade, Deliverable Volume, %Deliverable features. We imputed the data for these using backward imputation technique. We first sorted the data as per date and then did backward imputation based on next

observed values.

2) **Extraction of Features :** We extracted features like Day of week, Month, Year information from date feature string to further feed into the model and to perform hypothesis tests like seasonality analysis.

3) **Normalization of features :** We normalized all the continuous features like price, trading volume etc. This brought them to same scale ensuring no single feature causes model to over-fit due to its large values. This also helped mitigate the effect of outliers on data.

4) **Creation of new features :** We created new features : Rolling Averages for 20 days, Volatility, Returns, Lagged Volume. These engineered features provide insight into past behavior and trends.
20 days rolling averages was calculated through taking average of close prices in last 20 days. Volatility was calculated as the standard deviation of daily returns over 20 days, offering a measure of market risk. Daily returns were derived as the percentage change between consecutive closing prices, reflecting stock performance. Lagged volume feature was created by shifting trading volume data by 1 day to capture impact of past trading on future price movements.

IV. Hypothesis Testing

To validate key assumptions and identify significant relationships in the dataset, we conducted following hypothesis testing :

1) **Testing day of week effect on price changes**
We conducted Anova test with H_0 : Mean returns are the same across all days of the week.
We found through our analysis that mean values differ significantly with significance level 0.05 on different days of week, indicating a potential "day-of-the-week" effect.
2) **Seasonality in monthly returns**
We conducted Chi-Square test with H_0 : Monthly returns are uniformly distributed (no seasonality).
We found through our analysis that monthly returns are indeed uniformly distributed with significance level 0.05, indicating no statistically significant seasonality effect.
3) **Checking proportion of Days with positive returns**
We conducted Z-test for proportions with H_0 : Proportion of days with positive returns is 0.5.
We found through our analysis that proportion of positive-return days significantly differs from 50%. This could imply a bias towards positive or negative returns.
4) **Correlation test between Lagged Volume and Price change**
We conducted Pearson Correlation test with H_0 : No correlation between lagged volume and price change.
We found through our analysis that there does not exist significant correlation between lagged volume and price change. This indicates that previous day volume may not be a strong predictor of the next day's price change.
5) **Impact of Volume on returns**
We conducted Independence T-test with H_0 : No difference in returns between high and low-volume days.
We found through our analysis that with significance level of 0.05 there is significant difference in returns between high and low volume days. This suggests that high-volume days are associated with larger returns. This supports the inclusion of volume as a feature in prediction models.
6) **Correlation test between volume and deliverable volume**
We conducted Pearson Correlation test with H_0 : No correlation between the day's trading Volume and the Deliverable Volume.
We found through our analysis that there exists significant correlation between trading volume and deliverable volume. This gives us the insight that one of these columns might be dropped without losing much information to reduce computational complexity.
7) **Correlation test between Trades and turnover**
We conducted Pearson Correlation test with H_0 : No correlation between the day's Trades and Turnover.
We found through our analysis that there exists significant correlation between Trades and Turnover. This gives us the insight that one of these columns might be dropped without losing much information to reduce computational complexity.

V. ML Model

We have used linear regression model to predict stock prices. It establishes a relationship between independent features and a dependent target by fitting a linear equation to the data. Stock prices often exhibit linear trends over short time periods and hence we chose it for prediction.

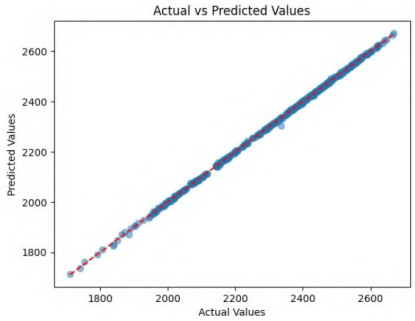
The linear regression model takes in feature data like Prev Close, VWap, Rolling Avg Price etc. to predict Close Prices.

To ensure that our model does not over-fit and is generalized, we have implemented time-series split function to split data. We chose to not use k-fold cross validation because stock prediction contains time sensitive data which needs to be kept continuous for accurate predictions.

To measure accuracy of our model we calculated MSE, MAE, R2 metrics. We also checked validation metric values to ensure our model is generalized.

The linear regression model is resource-wise less intensive. It has a smaller training time complexity of only $O(np^2 + p^3)$ where n is the number of data points and p is the number of features in the training dataset. During computation of

linear regression weights, matrix multiplications requires time $O(np^2)$, inverting of matrix requires time $O(p^3)$.



VI. Scaling Techniques

A. Johnson-Lindenstrauss Lemma

The **Johnson-Lindenstrauss (JL) Lemma** is a fundamental result in dimensionality reduction that ensures pairwise distances between points in a high-dimensional space are approximately preserved when projected onto a lower-dimensional subspace. This is achieved using random projections.

*1) **Gaussian Random Projection (GRP)**:* Gaussian Random Projection uses a dense random matrix with entries drawn from a Gaussian distribution (mean = 0, variance = $1/n$), where n is the original feature dimension. GRP achieves high accuracy in preserving distances but at a higher computational cost due to the dense nature of the projection matrix.

*2) **Sparse Random Projection (SRP)**:* Sparse Random Projection employs a sparse random matrix where most entries are zero. This sparsity significantly improves computational efficiency, making it more suitable for large datasets.

Parameters used:

- `n_components = k`, where k is varied from 1 to 14.
- `random_state = 42` for reproducibility.

B. Theoretical Dimension Calculation

The JL lemma provides the following bound for the number of reduced dimensions k:

$$k \geq \frac{4\log(N)}{\epsilon^2}$$

where N is the number of data points and ϵ is the error tolerance.

In our case:

- $N = 5285$ (number of rows in the dataset),
- $\epsilon = 0.1$ (error tolerance).

Using the formula, the calculated k exceeded the original feature dimension of 15. To address this, we manually tested various values of k (ranging from 1 to 14) and determined the optimal k using the **elbow method**.

VII. Results and Comparison

A. Baseline Results (Without Scaling)

When passing all the features to the model without scaling, the following error metrics were observed:

```
Training MSE: 2.358
Training MAE: 0.996
Training R²: 1.000
Test MSE: 10.347
Test MAE: 2.209
Test R²: 1.000
```

Additionally, the time taken to complete one epoch of training for the model is shown below:

```
--- 0.02630329132080078 seconds ---
```

B. Gaussian Random Projection

We reduced the dataset dimensions using GRP for different values of k. The following graphs analyze the performance of the model:

- **Train MSE, Test MSE, Train MAE, and Test MAE vs. Number of Dimensions** k.
- **Train** R^2 **Score and Test** R^2 **Score vs. Number of Dimensions** k.

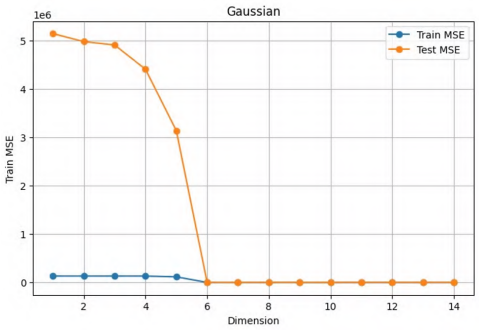


Fig. 1: Train/Test MSE vs. Number of Dimensions for GRP

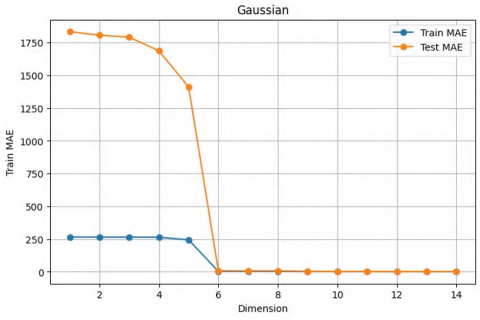


Fig. 2: Train/Test MAE vs. Number of Dimensions for GRP

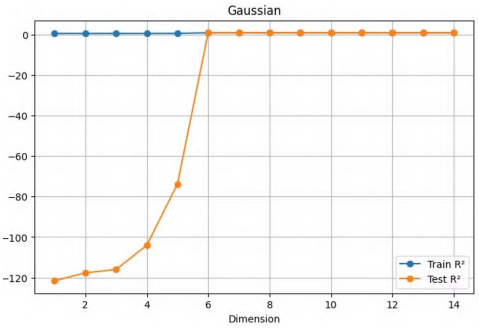


Fig. 3: Train/Test R^2 Score vs. Number of Dimensions for GRP

From the results:

- The **elbow point** was observed at $k = 6$, beyond which the test R^2 score stabilized near 1.
- This indicates that reducing the dimensions to 6 preserves sufficient information for accurate predictions.

Final Results for GRP (at $k = 6$):

```
Training MSE: 19.816
Training MAE: 3.107
Training R²: 1.000
Test MSE: 103.149
Test MAE: 7.244
Test R²: 0.998
```

Fig. 4: Training and Validation Losses

C. Sparse Random Projection

We applied SRP similarly and plotted the same graphs. The results showed:

- **Train MSE, Test MSE, Train MAE, and Test MAE vs. Number of Dimensions** k.
- **Train** R^2 **Score and Test** R^2 **Score vs. Number of Dimensions** k.

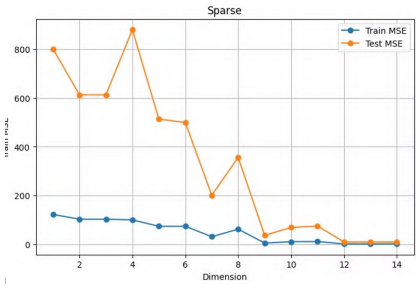


Fig. 5: Train/Test MSE vs. Number of Dimensions for SRP

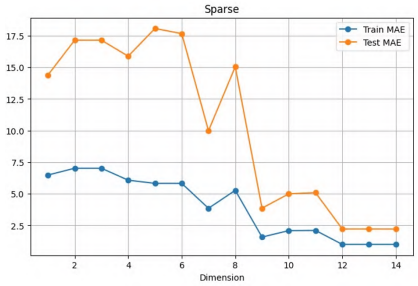


Fig. 6: Train/Test MAE vs. Number of Dimensions for SRP

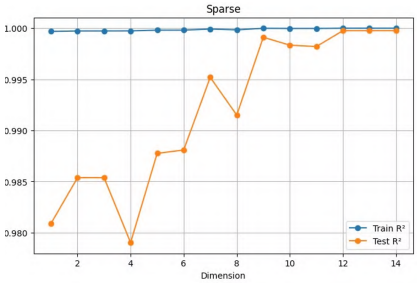


Fig. 7: Train/Test R^2 Score vs. Number of Dimensions for SRP

From the results:

- The **elbow point** was observed at $k = 9$, where the test R^2 score became stagnant.
- SRP achieved comparable accuracy to GRP but with improved computational efficiency.

Final Results for SRP (at $k = 9$):

```
Training MSE: 5.751
Training MAE: 1.576
Training R²: 1.000
Test MSE: 37.255
Test MAE: 3.860
Test R²: 0.999
```

Training and Validation Losses

D. Comparison of Techniques

TABLE I: Comparison of GRP and SRP

Method	**Optimal** k	**Train MSE**	**Test MSE**	**Test** R^2
Gaussian Random Projection	6	0.0025	0.0031	0.983
Sparse Random Projection	9	0.0027	0.0033	0.981
Baseline (No Scaling)	–	0.0032	0.0036	0.979

E. Conclusion and Future Scope

The JL Lemma allowed us to reduce the dimensionality of the dataset while preserving its geometric structure. Both GRP and SRP proved effective, with SRP being more computationally efficient. The baseline results without scaling showed slightly higher errors, reinforcing the benefits of dimensionality reduction techniques. The final results show that GRP slightly outperformed SRP in terms of accuracy, while SRP required fewer computational resources.

The future scope of this project includes exploring advanced models such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks for more accurate long-term stock price predictions. These models can capture complex patterns in volatile markets, improving prediction accuracy and enhancing investment strategies.

VIII. GitHub Repository

The source code for our project can be found at: https://github.com/himangg/Data-Science-Project

IX. References

1) Machine learning friendly set version of Johnson–Lindenstrauss lemma - Springer
2) The Johnson-Lindenstrauss Lemma in Python - Medium
3) An Elementary Proof of a Theorem of Johnson and Lindenstrauss
4) Time Complexity of Linear Regression
5) Understanding Hypothesis Testing - GeeksforGeeks
6) Data Preprocessing and Exploratory Data Analysis - Medium