# HDSC December ’22 Premiere Project Presentation: Stock Portfolio Performance

## A Project by Team Gitkraken

21.12.2022

Portfolio performance measures are a key factor in the investment decision. The recent trends in stock market prediction technologies are making the use of machine learning which makes predictions based on the values of current stock market by training data on the obtained previous values. With the advancements in machine learning, a variety of algorithms are used for various purposes like data mining, image processing, predictive analytics, etc.

In this project, regression analysis is carried out using the training data set in order to use the correct model for better prediction and accuracy.

The datasets are simulated with US stock market historical data to obtain their performances. Performance prediction models were built with the simulated performance data set.


Machine Learning in Stock Prediction

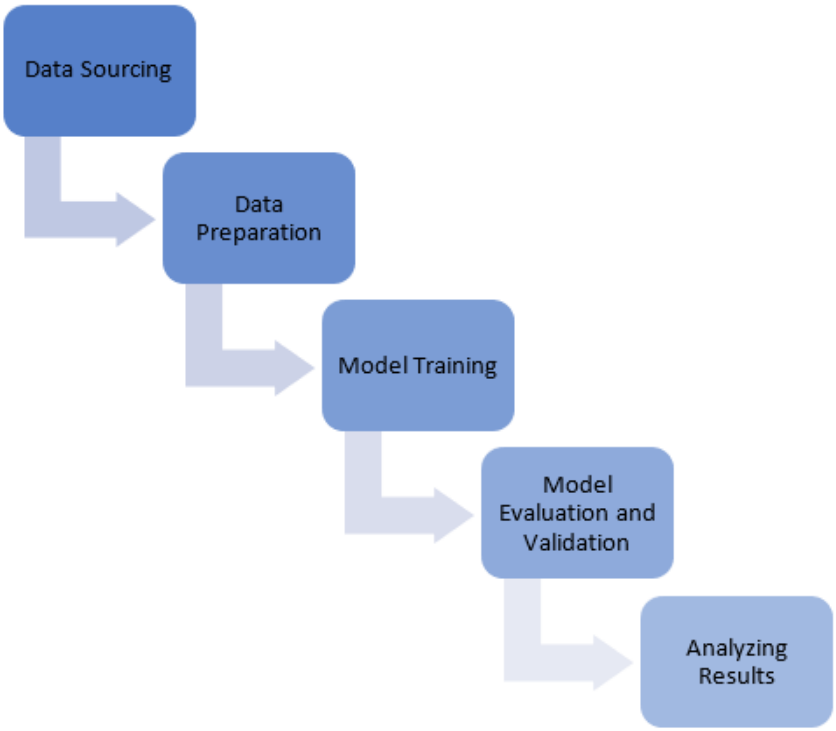
### Aims and objectives

The project aims to analyze the performance of stock, annual returns, risks, win rates, find the relationship between stock pricing concepts and performances of the portfolios.

Our main objective is to discover the optimal combination of weights of stock pricing concepts and predict the absolute win rate of a stock.

## Flow Process

There are 5 stages involved in the project.




Flow Process of the project

1. Data Sourcing—This stage implicates gathering the dataset required for this project. The data mining process is a subset of this stage.
2. Data Preparation—This stage involves data wrangling, data cleaning and removal of outliers. It also includes EDA(Exploratory Data Analysis), by which we can derive insights from the dataset.
3. Model Training—In this stage, the cleaned data is now fed to the model, and the model learns patterns from our dataset.
4. Model Evaluation and Validation—Now, after training the model(s), it is used to make some predictions. Then its performance would be evaluated and validated.
5. Analyzing results—After we are done making predictions with some models, we consider the model with the best accuracy to make accurate predictions.

## Understanding the dataset

The dataset which we have obtained has six worksheets.

The first five worksheets are recordings which contain the actual data and features, divided in certain time frames. While the last worksheet has information related to the holding period of the stock in years.

Each of the performance indicators adopted in this project has been normalized into the same scale during the same time frame of 20 years (80 quarters).

The entire stock data provided has been classified to 3 parts, each part having its own features:

1. The weight of the stock-picking concept.
2. The original investment performance indicator : This refers to the relative return rates of the firm's stock and does not consider the market's tendency. This can be solved if the covered time period is shorter, but in our dataset, we are considering the whole period.
3. The normalized investment performance indicator : This refers to the absolute return rates of the firm's stock and this results in precise performance of prediction for a specific time frame.

| The weight of the stock-picking concept. | | | | | | | Original investment performance indicator | | | | | | Normalized investment performance indicator | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | Large B/P | Large ROE | Large S/P | Large Return Rate in the last quarter | Large Market Value | Small systematic Risk | Annual Return | Excess Return | Systematic Risk | Total Risk | Abs. Win Rate | Rel. Win Rate | Annual Return.1 | Excess Return.1 | Systematic Risk.1 | Total Risk.1 | Abs. Win Rate.1 | Rel. Win Rate.1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0.139 | 0.01 | 1.33 | 0.149 | 0.663 | 0.525 | 0.531875 | 0.478116 | 0.738015 | 0.8 | 0.52 | 0.411765 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0.143 | 0.01 | 1.17 | 0.108 | 0.663 | 0.65 | 0.549712 | 0.487595 | 0.571579 | 0.412231 | 0.52 | 0.764706 |
| 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0.173 | 0.018 | 1.3 | 0.144 | 0.638 | 0.513 | 0.692625 | 0.629895 | 0.703051 | 0.756879 | 0.44 | 0.376471 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0.096 | -0.002 | 1.39 | 0.144 | 0.613 | 0.475 | 0.324351 | 0.255634 | 0.8 | 0.756046 | 0.36 | 0.270588 |
| 5 | 0 | 0 | 0 | 0 | 1 | 0 | 0.096 | 0.001 | 1.04 | 0.087 | 0.725 | 0.538 | 0.326615 | 0.306501 | 0.432452 | 0.209289 | 0.72 | 0.447059 |

Features of the dataset

## Data Source

The dataset has been obtained from this link.

The data set of performances of weighted scoring stock portfolios are obtained with mixture design from the US stock market historical database.

## Features of interest

Some of the important features given in the dataset are described below.

● *B/P*—Companies use the price-to-book ratio (P/B ratio) to compare a firm's market capitalization to its book value. The price-to-book ratio is often used by value investors looking for stocks that are underpriced by the market.

● *ROE*—Return on equity (ROE) is calculated by dividing a company's net income by its shareholders' equity, thereby arriving at a measure of how efficient a company is in generating profits

● *Rate of return*—It is the gain or loss of an investment over a specified period of time, expressed as a percentage of the investment's cost.

● *Systematic risk*, also known as market risk—This is the risk that is inherent to the entire market, rather than a particular stock or industry sector.

● *The win-rate*—This is a number of profitable trades during a certain period of time in the general number of executed trades for the same period of time.

● *Annual return*—The annual rate of return is the profit or loss on an investment over a one-year period.

● *Excess return*—The excess returns are the return earned by a stock (or portfolio of stocks) and the risk-free rate.

● *Market value*—This can be said as the investment given to specific equity or a business; the price an asset can fetch in the marketplace.

● *Absolute Winning Rate* -The ratio between the number of portfolios holding periods with positive return rate and the total number of portfolios holding periods.

● *Relative Winning Rate*—The ratio between the number of portfolios holding periods with a return rate greater than the market return rate and the total number of portfolio holding periods.

## Data Preparation

The first look of the imported dataset looks somewhat untidy.

```
df= pd.read_excel('stock portfolio performance data set (2).xlsx', sheet_name ='all period')
df.head(5)
```

| | Unnamed: 0 | the weight of the stock-picking concept | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 | the original investment performance indicator | Unnamed: 8 | Unnamed: 9 | Unnamed: 10 | Unnamed: 11 | Unnamed: 12 | the normalized investment performance indicator | Unnamed: 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ID | Large B/P | Large ROE | Large S/P | Large Return Rate in the last quarter | Large Market Value | Small systematic Risk | Annual Return | Excess Return | Systematic Risk | Total Risk | Abs. Win Rate | Rel. Win Rate | Annual Return.1 | Excess Return.1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0.139 | 0.01 | 1.33 | 0.149 | 0.663 | 0.525 | 0.531875 | 0.478116 |
| 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0.143 | 0.01 | 1.17 | 0.108 | 0.663 | 0.65 | 0.549712 | 0.487595 |
| 3 | 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0.173 | 0.018 | 1.3 | 0.144 | 0.638 | 0.513 | 0.692625 | 0.629895 |
| 4 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0.096 | -0.002 | 1.39 | 0.144 | 0.613 | 0.475 | 0.324351 | 0.255634 |

First impression of imported dataset

We cannot understand the features from above obtained column names. Therefore, we assign a new header to the data frame.

```
new_header = df.iloc[0]
df = df[1:]
df.columns = new_header
```

```
#Check the first 5 columns
df.head()
```

| | ID | Large B/P | Large ROE | Large S/P | Large Return Rate in the last quarter | Large Market Value | Small systematic Risk | Annual Return | Excess Return | Systematic Risk | Total Risk | Abs. Win Rate | Rel. Win Rate | Annual Return.1 | Excess Return.1 | Systematic Risk.1 | Total Risk.1 | Abs. Win Rate.1 | Rel. Win Rate.1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0.139 | 0.01 | 1.33 | 0.149 | 0.663 | 0.525 | 0.531875 | 0.478116 | 0.738015 | 0.8 | 0.52 | 0.411765 |
| 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0.143 | 0.01 | 1.17 | 0.108 | 0.663 | 0.65 | 0.549712 | 0.487595 | 0.571579 | 0.412231 | 0.52 | 0.764706 |
| 3 | 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0.173 | 0.018 | 1.3 | 0.144 | 0.638 | 0.513 | 0.692625 | 0.629895 | 0.703051 | 0.756879 | 0.44 | 0.376471 |
| 4 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0.096 | -0.002 | 1.39 | 0.144 | 0.613 | 0.475 | 0.324351 | 0.255634 | 0.8 | 0.756046 | 0.36 | 0.270588 |

Assigning new header to make the frame look clean

After we have obtained the better data frame, we check the data types of all the features. We get to know that all the types are of "object" type.

Machine Learning algorithms usually rely on mathematical operations which require their inputs to be of numeric type.

```
RangeIndex: 63 entries, 1 to 63
Data columns (total 19 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   ID                                      63 non-null     object
 1    Large B/P                              63 non-null     object
 2    Large ROE                              63 non-null     object
 3    Large S/P                              63 non-null     object
 4    Large Return Rate in the last quarter  63 non-null     object
 5    Large Market Value                     63 non-null     object
 6    Small systematic Risk                  63 non-null     object
 7   Annual Return                           63 non-null     object
 8   Excess Return                           63 non-null     object
 9   Systematic Risk                         63 non-null     object
 10  Total Risk                              63 non-null     object
 11  Abs. Win Rate                           63 non-null     object
 12  Rel. Win Rate                           63 non-null     object
 13  Annual Return.1                         63 non-null     object
 14  Excess Return.1                         63 non-null     object
 15  Systematic Risk.1                       63 non-null     object
 16  Total Risk.1                            63 non-null     object
 17  Abs. Win Rate.1                         63 non-null     object
 18  Rel. Win Rate.1                         63 non-null     object
dtypes: object(19)
memory usage: 9.5+ KB
```

Initial data type of features

Now, these types must be converted to numeric values, so that they can be worked upon.

```
df1 = df.apply(pd.to_numeric)
```

Changing the datatype

The datatype of all the features has been changed to numeric.

```
dtypes: float64(12), int64(1)
memory usage: 6.5 KB
```

To perform some visualization and plots, we need to rename the columns to better looking column names.

```
#Replacing the characters in column names
df.columns = [c.replace(' ', '_') for c in df.columns]
df
```

```
#Renaming columns to aid visualization
df.columns = [c.replace('.', '') for c in df.columns]
df.columns
```

Renaming the columns

After renaming the columns properly using the above two functions, our data is now produced, and we are now ready to perform some analysis and visualize the data.

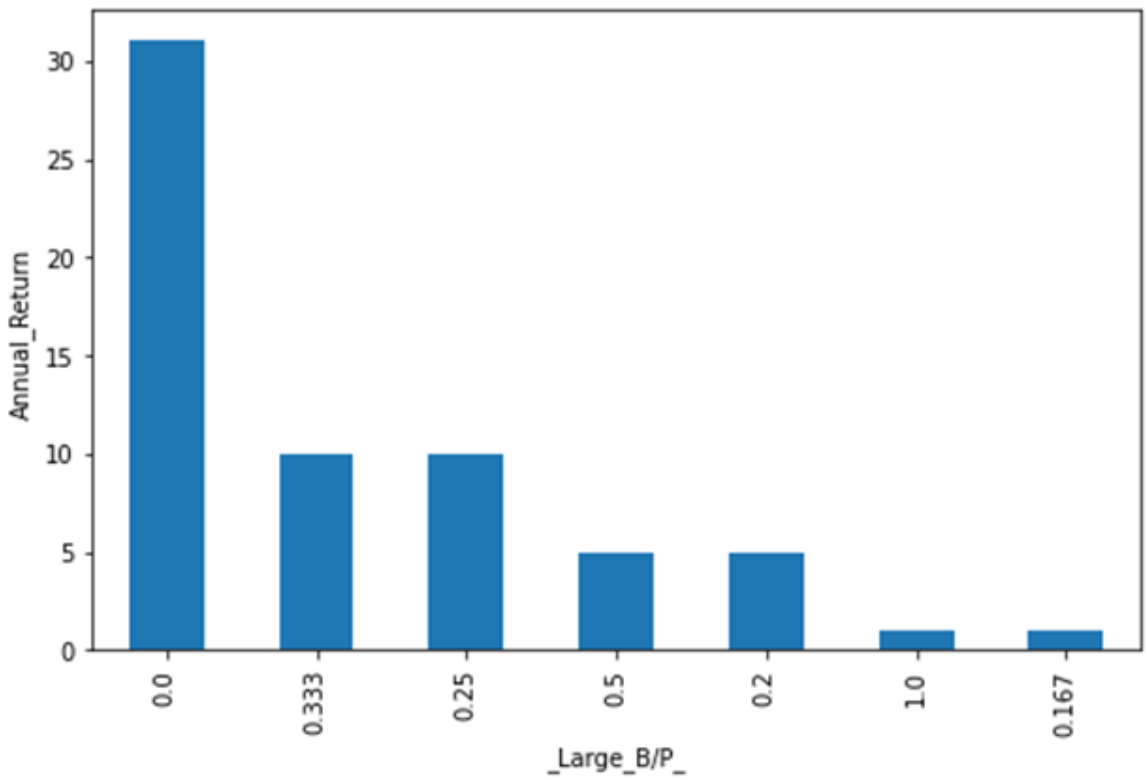
Before splitting the data into training sets and testing sets, we decided to drop the original portfolio investment performance indicators since we already have the normalized investments indicators.

With this understanding and considering our project circumstances, the original indicators were dropped and limitations of our project were set to the timeframe of the dataset.

```
df1 = df.drop(columns = ['Annual_Return', 'Excess_Return', 'Systematic_Risk', 'Total_Risk', 'Abs_Win_Rate', 'Rel_Win_Rate'])
df1.head(3)
```

Dropping the original indicators

We now visualize the dataset by making use of different bar plots, histogram, and pie charts, to gain some insights.

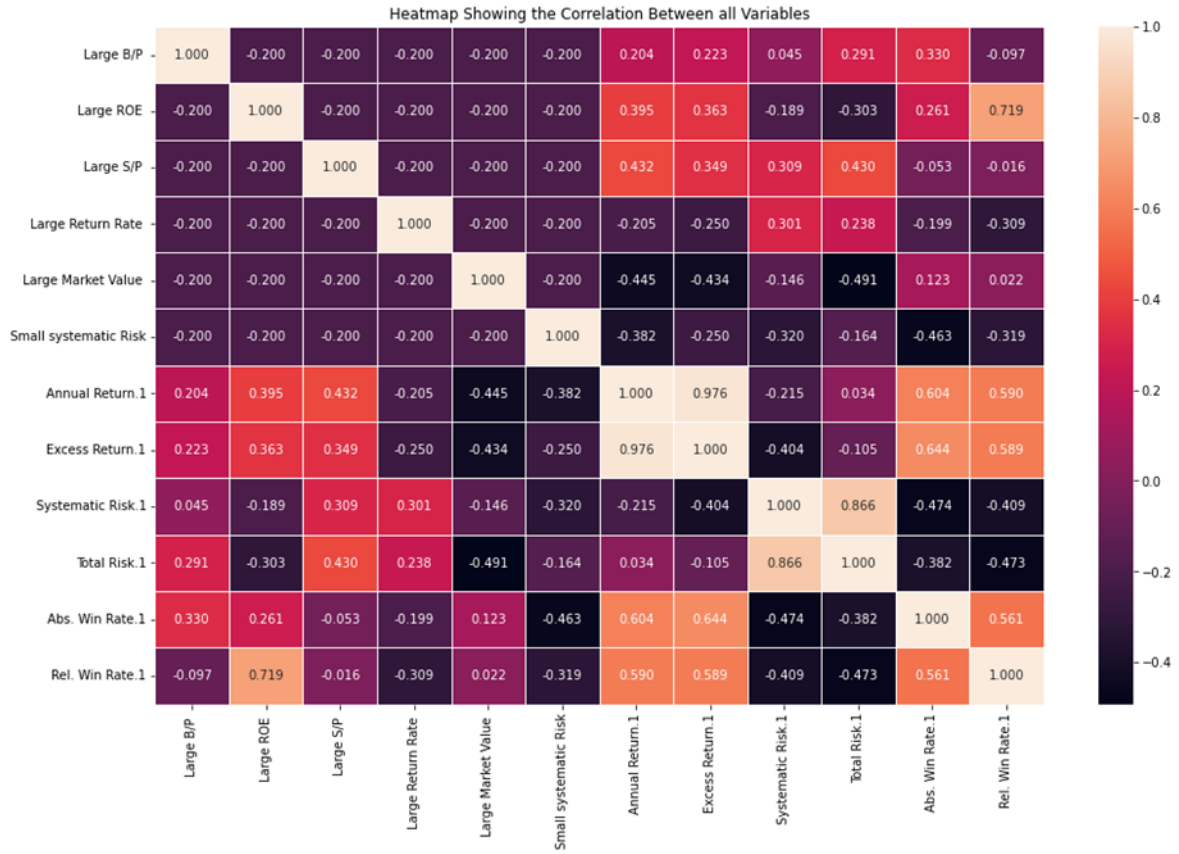



A sample bar plot of B/P ratio to Annual Return reveals that the two features are in inverse proportion

The data is now cleaned, and all the necessary features have been obtained. Now, the most important part,

## Model Training, Evaluation, and Validation

In order to build the model, we used heatmap to inspect the contribution of each of the variables to find out some correlations between the features.




Heatmap showing correlation between all the variables

First, we split the data into dependent variable y, in this case it's **Abs_Win_Rate1** the variable that we want to predict, and independent variables X, where we performed a featured selection based on the correlation matrix to optimize the learning and to avoid the heteroskedasticity, we didn't include many columns for that reason.

A base CatBoost Classifier was trained, a function was defined to fit and evaluate the models and corresponding scores were obtained. The models used were as follows:

1. XGBoost Regressor

2. Decision Tree Regressor

3. Random Forest Regressor

4. CatBoost Regressor

5. Gradient Boosting Regressor

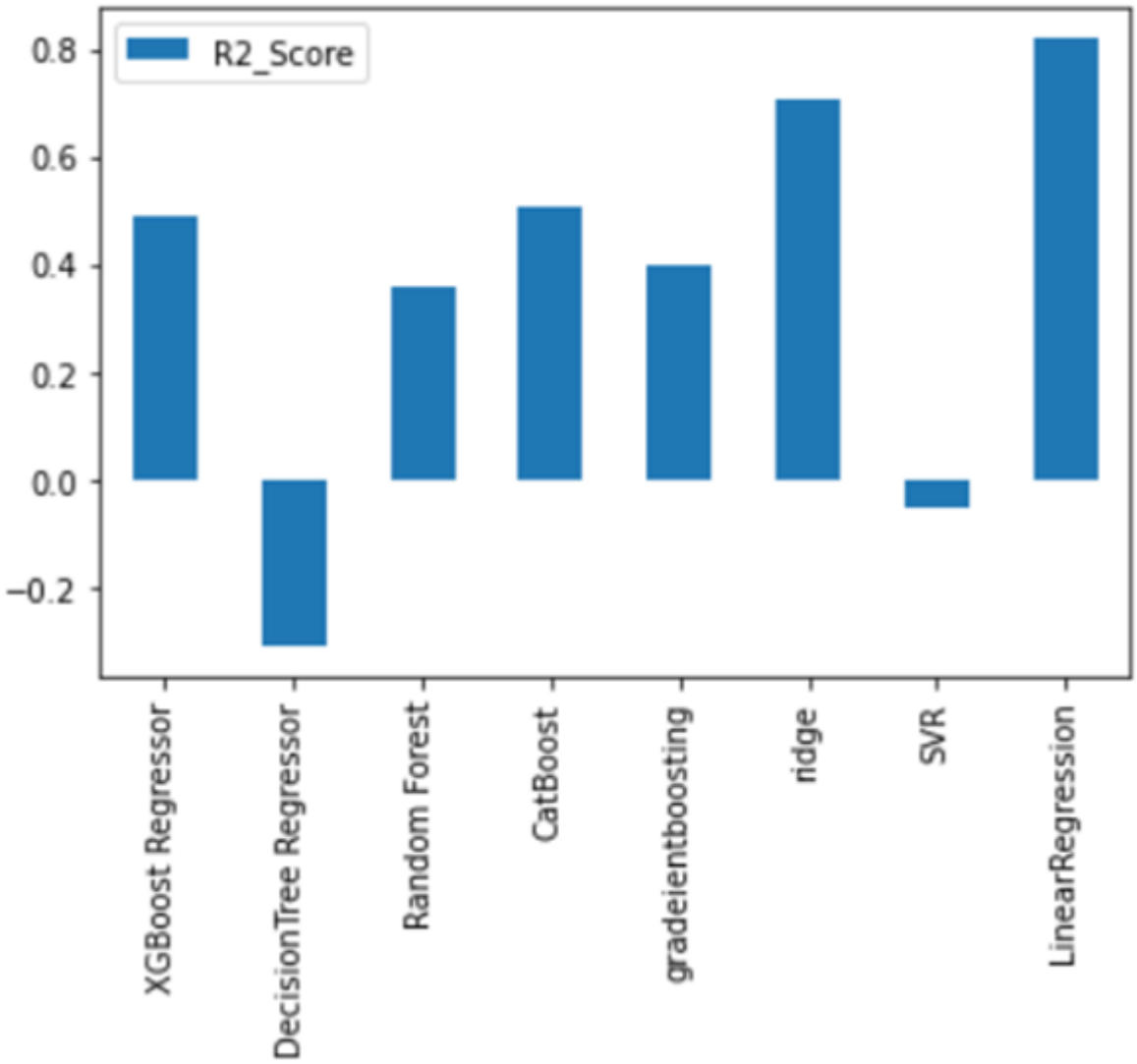
6. Ridge Regressor

7. Support Vector Regressor

8. Linear Regressor

## Results

The scores obtained for the different models were as follows:

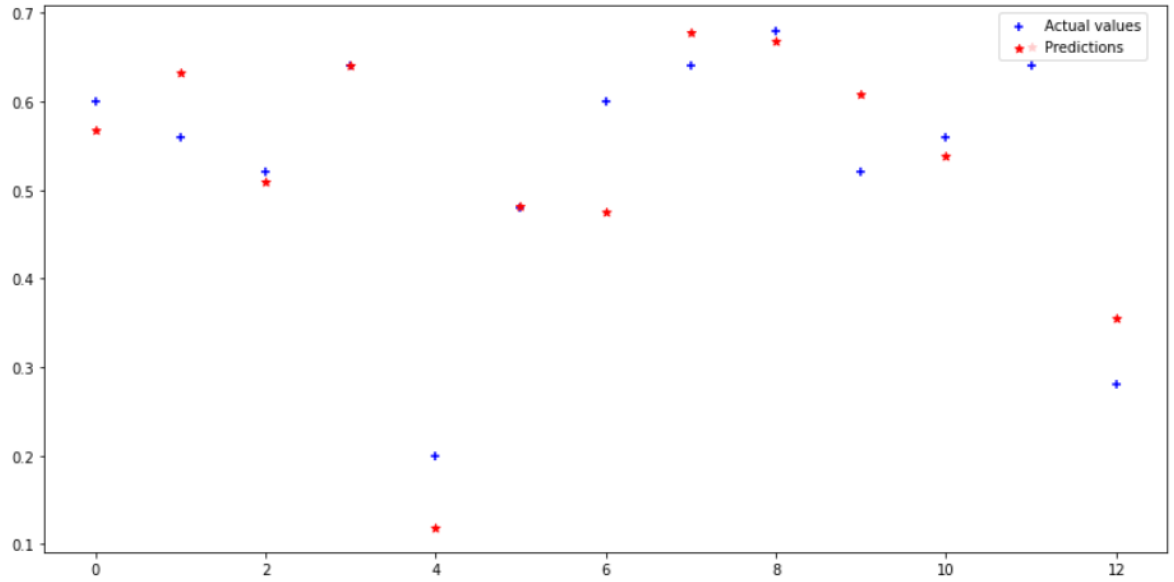



Models VS r2 Scores

| Model | R2 Score |
|---|---|
| XGBoost Regressor | 0.4893827873169776 |
| Decision Tree Regressor | -0.309164149043303 |
| Random Forest | 0.35924924471299136 |
| CatBoost | 0.5066279465594087 |
| Gradient Boosting | 0.3972664020324649 |
| Ridge | 0.7051131719147254 |
| SVR | -0.05518071987522877 |
| Linear Regression | 0.8187618435005503 |

The scores of all the different regression models used were compared. It is clearly evident that the Linear model gave the best r2 score.

When we compare the actual and predicted values of absolute win rates by visualizing the errors, we observe that our predictions are exquisite.




Visualizing the errors

## Conclusion

According to the evaluation of performance prediction models based on regression methods, absolute win rates can be predicted more precisely than relative win rates. These may be attributed to the reason that an individual firm's fundamental analysis is only effective for stock picking but not useful for market timing. The accuracy of the results largely depends on the quality of the dataset fed into the model. Portfolio performance measures are a key factor in the investment decision. These tools also assist to provide insights and information to investors.

## Team Members

1. Hannah Kasali (Team Leader)
2. Christian Tan (Assistant Team Leader)
3. Apurv Deshpande (Query Analyst)
4. Zakaria Jnayni
5. Lovette Duke
6. Fachi Okolo