**EPS Prediction Model Development Report**

**Azhan Saleem**

Table of Contents

[INTRODUCTION 3](#_Toc165656949)

[Dataset Description 3](#_Toc165656950)

[Data Cleaning and Feature Engineering 3](#_Toc165656951)

[Data Cleaning and Preprocessing: 3](#_Toc165656952)

[Feature Engineering: 5](#_Toc165656953)

[Descriptive Statistics 6](#_Toc165656954)

[Exploratory Data Analysis (EDA): 7](#_Toc165656955)

[Visual Analysis: 7](#_Toc165656956)

[Scatter plot- 7](#_Toc165656957)

[Correlation Analysis 9](#_Toc165656958)

[Current Accuracy Calculation 10](#_Toc165656959)

[Group Comparisons 10](#_Toc165656960)

[Insightful Observations 12](#_Toc165656961)

[Splitting the dataset 12](#_Toc165656962)

[Extraction of Year Information 12](#_Toc165656963)

[Determining the Split Year 12](#_Toc165656964)

[Creating Training and Testing Subsets 13](#_Toc165656965)

[Verification of Data Split 13](#_Toc165656966)

[Modeling Phase 13](#_Toc165656967)

[Backward Selection and Linear Regression 13](#_Toc165656968)

[Data preparation 13](#_Toc165656969)

[Backward Elimination 14](#_Toc165656970)

[Linear regression model 14](#_Toc165656971)

[Results 14](#_Toc165656972)

[KNN Model 15](#_Toc165656973)

[Data Preparation 15](#_Toc165656974)

[KNN and Cross-Validation Technique 15](#_Toc165656975)

[Model Training and K determination. 15](#_Toc165656976)

[Results 16](#_Toc165656977)

[Lasso Model 16](#_Toc165656978)

[Data Preparation 16](#_Toc165656979)

[Lasso Model Implementation 16](#_Toc165656980)

[Results 16](#_Toc165656981)

[Random Forest Model 17](#_Toc165656982)

[Data Preparation 17](#_Toc165656983)

[Random Forest Model Building 17](#_Toc165656984)

[Result 17](#_Toc165656985)

[Evaluating Model Performance with MSPE, RMSE, MAE, MSE, R2 18](#_Toc165656986)

[Assessment of the Benchmark Model 19](#_Toc165656987)

[Comparison of Model Evaluation and Benchmark Model 20](#_Toc165656988)

[Conclusion and Discussion 21](#_Toc165656989)

# **INTRODUCTION**

This project report outlines the development of a predictive model for Earnings Per Share (EPS) for the client, Sink or Swim (SOS). Earnings Per Share (EPS) is used as one of the key financial indicators utilized by businesses to evaluate their profitability and the worth of their stocks. It holds significant importance when making financial decisions for investors by predicting future EPS.

In this project, we use data from the Institutional Brokers' Estimate System (I/B/E/S) provided by Wharton Research Data Services (WRDS). The dataset contains detailed forecasts of EPS made by analysts which allows us to see if projected financial results correspond with actual outcomes. This requires cleaning up messy data, creating features as well as applying advanced statistical methods so that we can come up with a highly accurate model for predicting future values of EPS. This report will go through each step, starting from cleaning the data, pre-processing the data until building and validating models. We aim to improve decision-making for SOS by giving them a tool to forecast EPS efficiently and reliably. We intend to document our approach and findings to provide a complete guide to the project methodology along with insights gained throughout this study.

# **Dataset Description**

The data set for this project is from Wharton research data services through the institutional brokers estimate system which provides comprehensive historical financial indicator forecasts by analysts and among them earnings per share EPS predictions are considered the most important indicators of a company’s financial performance. Below are the primary attributes included in the dataset.

The dataset shape is (2845, 16)

* **TICKER:** The unique identifier assigned to each security, helping in tracking specific companies.
* **CNAME:** The name of the company to which the EPS forecast applies.
* **ACTDATS:** The activation date when the analyst's forecast was recorded in the I/B/E/S database.
* **ESTIMATOR:** Represents the sell side institution (mostly a brokerage house) from which the forecast originated.
* **ANALYS:** The code representing the analyst who made the forecast, preserving anonymity.
* **FPI (Forecast Period Indicator):** This indicator shows if the forecast is for next fiscal quarter or year.
* **MEASURE:** This has the forecasts for Earnings Per Share (EPS).
* **VALUE:** The forecasted EPS value per analyst.
* **FPEDATS:** The forecast period end date, typically December 31st for annual forecasts.
* **REVDATS and REVTIMS:** The most recent dates and timestamps, respectively, when the forecast was reviewed and confirmed by the analyst.
* **ANNDATS and ANNTIMS:** The initial date and time when the forecast was made.
* **ACTUAL:** depicts the actual realized EPS value which is used for forecast accuracy evaluation.
* **ANNDATS\_ACT and ANNTIMS\_ACT:** shows the date and time when the actual EPS was officially announced by the company.

# **Data Cleaning and Feature Engineering**

## Data Cleaning and Preprocessing:

We performed cleaning and preprocessing of the data to ensure reliability and integrity of analysis done.

This involved:

1. **Handling Missing Values: ACTUAL, ANNDATS\_ACT, and ANNTIMS\_ACT** columns had 44 missing values. To keep up quality standards as well as ensure that we have complete records, rows containing missing values were dropped off. By doing this step we prevent any skewness in results caused by using incorrect or partial information.

**Number of missing values in the dataset**

A screenshot of a computer

Description automatically generated

**Number of missing values after removing in dataset**

A screenshot of a computer

Description automatically generated

**Filtering Forecasts:** Only the latest prediction among all forecasts made by an analyst for a given financial year was kept. This means that each year under review will be represented by what each individual thought at last about it since this might have been their best guess considering everything they knew thus far.

The below screen shot shows the dataset info after Filtering Forecast. The area is highlighted.

A screenshot of a computer screen

Description automatically generated

## Feature Engineering:

Several new features were engineered to enhance the model's predictive capabilities:

1. **Past Accuracy Calculation:** We calculated a **past\_accuracy** metric for each analyst to evaluate an individual’s success in the past, we compared their previous year’s forecast (value) against the actual realization EPS (actual) and calculated past\_accuracy. Thus, this function is a good estimator for future performance because history shows that people who predict well typically continue to do so.

We encountered null values in past\_accuracy as each unique pair of the analyst (**ANALYST**) and company (**CNAME**), the first fiscal year recorded in your dataset won't have a preceding year's forecast (**VALUE**) to compare against. Since there is no previous **VALUE** to subtract from this year's **ACTUAL**, **past\_accuracy** will be null for these entries, and decided to remove the rows of null values to keep consistency.

The below screenshot shows how past\_accuracy is calculated by handling null values.

A screenshot of a computer

Description automatically generated

1. **Forecast Horizon (horizon):** To create the horizon feature, the time difference between the forecast announcement date and the end date of the forecast period (ANNDATS\_ACT - ANNDATS) is calculated. This variable measures how far into the future a prediction extends and adds uncertainty when that time is longer.

The below screen shows how the horizon is calculated with horizon values.

A screenshot of a computer

Description automatically generated

1. **Analyst Experience (experience):** This measure indicates how long a forecaster has been making predictions about a single firm, thereby showing his or her understanding of the company’s financial statements.

The below screenshot shows how experience is calculated with few values of experience.

A screen shot of a computer

Description automatically generated

1. **Brokerage House Size (size):** We calculated a variable representing the total number of analysts from the same brokerage house who made predictions for the same company within the same year.

A screenshot of a computer

Description automatically generated

# **Descriptive Statistics**

A screenshot of a computer

Description automatically generated

Descriptive statistics is presented in above table which shows the typical values and spread of values across the variable we have used in our dataset. The Variables which we have taken are: 'MEASURE', 'ACTUAL', 'past accuracy', 'horizon', 'experience', and 'size'. This table provides valuable insights to how the data is distributed and variation for data and size in our dataset.

* Count: "ACTUAL" variable has 345 data points.
* Mean: Average value given in above table is 3.40 which is showing where the center of data is placed.
* Standard Deviation: Larger the standard deviation means that there is more variability in the dataset. The standard deviation for our dataset is 2.45 . A larger standard deviation suggests more variability in the data.
* Minimum: The lowest value observed in the "ACTUAL" variable is -2.40.
* 25th Percentile: 25% of the data points in "ACTUAL" fall below the value of 2.11.
* Median (50th Percentile): The median value of 3.30 indicates that 50% of the data points in "ACTUAL" are below this value and 50% are above it.
* 75th Percentile: 75% of the data points in "ACTUAL" fall below the value of 4.62.
* Maximum: The highest value observed in the "ACTUAL" variable is 8.88.

These numbers above give us an idea of how data is spread out.

# **Exploratory Data Analysis (EDA)**:

## **Visual Analysis:**

### **Scatter plot-**

There is no clear trend or pattern, it suggests that there's no significant relationship between 'past\_accuracy' and 'ACTUAL' values.

A diagram of a graph

Description automatically generated

Below code generates a histogram to visualize the distribution of analyst experience ('experience') in the dataset.

A graph of a number of people

Description automatically generated with medium confidence

The histogram provides a visual depiction of how analyst experience is distributed based on the number of years. It shows the concentration of experience levels present in the dataset. Additionally, the KDE plot superimposed on the histogram offers supplementary details regarding the overall shape of the distribution. if the count is decreasing towards higher experience levels, it indicates that fewer analysts have more years of experience.

The Box plots below shows the distribution of the forecast horizon (in days) for the data in the 'clean\_data' dataset.

A screen shot of a graph

Description automatically generated

The box represents the interquartile range (IQR), with the median indicated by the line inside the box.

The "whiskers" extend to 1.5 times the IQR above the upper quartile (Q3) and below the lower quartile (Q1).

In the graph above, there are data points beyond the whiskers which are d outliers and are displayed individually as points.

This graph is a box plot representing the distribution of actual EPS (Earnings Per Share) values in the 'clean\_data' dataset.

A screen shot of a graph

Description automatically generated

## **Correlation Analysis**

Correlation matrix as shown below. Below code creates a heatmap to visualize the correlation matrix. Each cell in the heatmap represents the correlation coefficient between numeric features.

The strong link between `VALUE’ and ‘ACTUAL’ indicates that this could be a direct measure of EPS, hence it is highly predictive and probably the most important one for modeling.  
  
It is interesting to note that negative correlations were found with ‘past\_accuracy’, ’FPEDATS’, and ‘ACTDATS’. This means that they may reflect more complicated aspects of patterns or shifts affecting EPS inversely.  
  
For instance, weak associations with variables like ‘size’ or ‘experience’ show that these alone cannot act as good predictors for ’ACTUAL’. But still, they could contribute to predictive models with other features.

A screenshot of a graph

Description automatically generated

## **Current Accuracy Calculation**

The graph below shows the distribution of the current accuracy values calculated by subtracting the actual EPS from the forecasted EPS for each observation in the dataset.

A screen shot of a graph

Description automatically generated

The distribution of current accuracy values does not exhibit a perfect symmetrical bell shape.

There seems to be some skewness in the distribution, as indicated by the longer tail on one side of the distribution compared to the other.

### **Group Comparisons**

We are calculating the mean current accuracy for different groups based on 'experience', 'size', and 'horizon' columns.

A screenshot of a computer

Description automatically generated

A graph of blue squares

Description automatically generated with medium confidence

Bars positioned above the zero line signify positive mean current accuracy values, indicating that, on average, the forecasted EPS values exceed the actual EPS values. This implies a propensity to overestimate earnings. On the other hand, bars below the zero line denote negative mean current accuracy values, suggesting that, on average, the forecasted EPS values are lower.

A graph of a bar chart

Description automatically generated with medium confidence

Size 2: The bar with a value of 2 indicates that there is a group of analysts with a size of 2.

Size 1: The bar with a value of 1 indicates that there is another group of analysts with a size of 1.

A graph showing a number of blue lines

Description automatically generated

The above graph shows analysts may be more accurate in predicting earnings for longer-term forecasts compared to shorter-term forecasts, on average.

### **Insightful Observations**

• Analyst teams with two members tend to make more accurate predictions than those with just one member. This suggests that working together or having more expertise in larger teams can help make better forecasts.

• Analysts who have been doing their job for a longer time tend to make more accurate predictions. So, the more experience they have, the better they are at forecasting.

• Predicting earnings far into the future is harder for analysts, and their forecasts tend to be less accurate compared to shorter-term predictions.

# **Splitting the dataset**

To train and validate our models that predict future outcomes, we had to split the data set into two parts: training and testing. The division was made according to a column called FPEDATS. By doing this we wanted our algorithms learn from historical records while testing them against more recent ones. We used the last four years of data as the testing subset.

## Extraction of Year Information

Firstly, we extracted the year from ‘FPEDATS’ column. We did this by converting dates into strings, slicing off the first four characters representing years and then re-converting this sequence to integer values again. So now we have a new column called ‘year’ with year numbers in it.



## Determining the Split Year

Next, we identified the unique years present in the dataset and determined the split year to distinguish between training and testing data. We sorted the unique years and selected the fourth-last year as our split point, ensuring that the testing dataset is representative of the most recent data.

A computer code with green and blue text

Description automatically generated

## Creating Training and Testing Subsets

Using the determined split year, the dataset was divided into two subsets:

* **Training Data:** Consists of all entries from years earlier than the split year.
* **Testing Data:** Comprises all entries from the split year and later.

This split was implemented using the following code:

A close-up of a computer code

Description automatically generated

## Verification of Data Split

Finally, we verified to ensure the split was executed as intended:

* **Training Data Years:** [1996, 1997, 1998, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2014, 2015, 2016]
* **Testing Data Years:** [2017, 2018, 2019, 2020]

These ranges confirm that the training data includes historical records up to 2016, while the testing data consists of the most recent records from 2017 onward.

A white background with black numbers

Description automatically generated

# **Modeling Phase**

## **Backward Selection and Linear Regression**

### Data preparation

To start, In the above dataset we removed predictors like ‘current\_accuracy’ as it can influence the model After that, we selected our numerical predictors and dropped the variable ‘ACTUAL’ which we will be predicting in our model.

A screenshot of a computer code

Description automatically generated

### Backward Elimination

We implemented the backward elimination method, selecting features based on their statistical significance with a threshold p-value of 0.05. This approach iteratively removes the least significant predictor until all remaining predictors are statistically significant.

A computer screen shot of a program code

Description automatically generated

### Linear regression model

The **linearModel\_Team1** was executed using the **statsmodels** library, which provided a detailed summary of the model's performance.

A screenshot of a computer code

Description automatically generated

### Results

#### Model Performance

The fitted model **linearModel\_Team1** showed an R-squared of 0.940 and an Adjusted R-squared of 0.939, which is referred to as linearModel\_Team1. This means that about 93.9% of the variance in EPS can be accounted for by selected features.

Important Predictors  
  
FPI: Coefficient = 34.8524, p-value = < 0.001; this represents a large positive effect on EPS where changes in FPI are highly related to variations in EPS.  
  
VALUE: Coefficient = 0.8809, p-value = < 0.001; another strong predictor showing positive relationship with EPS but not as powerful as FPI.  
  
past\_accuracy: Coefficient = -0.1331, p-value = < 0.001; its negative coefficient indicates that higher past accuracy scores are associated with lower levels of estimated earnings per share (EPS).  
  
horizon: Coefficient = -0.0027, p-value = < 0.001; like past\_accuracy it also shows a negative impact on EPS, implying longer horizons may cause depressed earnings per share (EPS) figures.  
  
Other Features   
  
ESTIMATOR: Coefficient = 0.0001, p-value = 0.016 —this feature has little effect on estimated earnings per share (EPS) due to its negligible coefficient value.  
  
ACTDATS: Coefficient = -1.714e-06, p-value = 0.013; this variable still exerts some negative influence over earnings per share (EPS) indicating that when ACTDATS increases slightly reduces the forecasted Earnings Per Share(EPS).

These predictors were selected based on their p-values and are crucial for forecasting EPS, according to the backward elimination method.

## **KNN Model**

### Data Preparation

We removed predictors like ‘current\_accuracy’ as it can influence the model After that, we selected our numerical predictors and dropped the variable ‘ACTUAL’ which we will be predicting in our model.

A close up of a text

Description automatically generated

### KNN and Cross-Validation Technique

We used the cross-validation method to determine the most suitable value for the number of neighbors (k) in the KNN model. In order to prevent tie situations and ensure that there is always a majority, we had to only consider odd values of k from 1 to 19 during the KNN algorithm implementation. Ten-fold cross-validation was utilized for each k whereby the performance of the model was evaluated using negative mean squared error as a scoring metric. We then calculated root mean squared error (RMSE) for each fold and the average across all folds was taken for every k.

### Model Training and K determination.

The exploration of k values revealed that the RMSE decreased as the number of neighbors increased, stabilizing and reaching the lowest RMSE at k=3. The optimal k value, therefore, was determined to be 1, indicating that the model performs best when considering the 3 most similar instances in the training data.

This indicates larger neighborhoods are useful in prediction because they help to reduce errors made while predicting which could be caused by noise or outliers in data used for prediction. Then we fitted the KNN model **knnModel\_Team1** with this optimal k.

A screenshot of a computer code

Description automatically generated

### Results

The final results from the KNN model, trained with an optimal k of 3, showed a significant reduction in the root mean squared error (RMSE) to 1.1045 , highlighting a high accuracy in predicting earnings per share (EPS). This decrease in RMSE with an increase in k value demonstrates the model’s improved performance through averaging more neighbors, effectively reducing the noise and influence of outliers.

A close up of text

Description automatically generated

Optimal k value: 3

Average RMSE for each k: [(1, 1.0560580135017112), (3, 1.0145862389492148), (5, 1.1464976567517218), (7, 1.2546019907937926), (9, 1.3178889000587144), (11, 1.391343195006397), (13, 1.4207958251321862), (15, 1.4712443052994906), (17, 1.4981098226664102), (19, 1.5311610466521028)]

A graph with a line

Description automatically generated

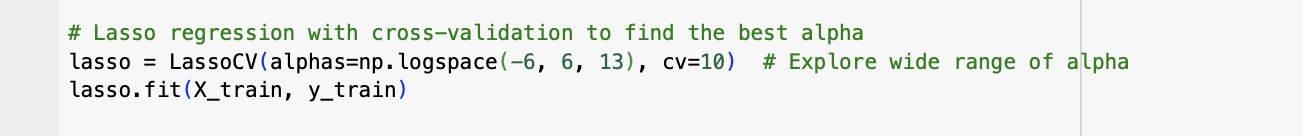
## **Lasso Model**

### Data Preparation

We utilized the x\_train and y\_train from the above defined.

### Lasso Model Implementation

For the Lasso regression, we utilized the **LassoCV** class from the **sklearn.linear\_model** module, which incorporates cross-validation to optimize the selection of the lambda penalty. This helps in reducing overfitting among the predictors. We explored a wide range of alpha values from 10 -6 to 10 6 across 13 logarithmically spaced points.



### Results

An optimal alpha of 0.01 was identified by the cross-validation process which means that little regularization is needed to ensure adequate model accuracy. We called this lassoModel\_Team1 and all it does is apply the alpha value we found earlier to configure the Lasso model.

A screenshot of a computer code

Description automatically generated

## **Random Forest Model**

### Data Preparation

We utilized the predictor\_columns, x\_train and y\_train from the above defined.

### Random Forest Model Building

One of the initial steps involved deciding on the most appropriate number of features to use in each tree. We considered possible number of features from 1 up to the total number of features we previously selected, which we knew were significant. This was necessary so that our model could accommodate different combinations of input data:

A screenshot of a computer code

Description automatically generated

The number of features in the model were tested using grid search, a systematic testing method. The test was run many times to ensure repeatability and find optimal parameters.



### Result

It was found through the tests that the highest performance occurs when, at each decision point, three features are used by the model. This optimal setting has been implemented in our final model, **RFModel\_Team1.**This implies that predictions will be most accurate if such a number of features are adopted by the model.

A screenshot of a computer program

Description automatically generated

## **Evaluating Model Performance with MSPE, RMSE, MAE, MSE, R2**

we evaluated four models: linear regression (linearModel\_Team1), K-Nearest Neighbors (knnModel\_Team1), Lasso regression (lassoModel\_Team1), and Random Forest (RFModel\_Team1). The evaluation involved testing these model’s performance on a given test data using multiple statistical metrics for prediction accuracy and model effectiveness measurement.  
  
Metrics Calculated:

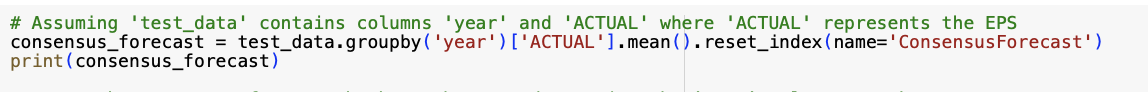
Mean Squared Prediction Error(MSPE): Captures the average percentage error between actual EPS values and predictions made by models, square it then sum them up  
  
Root Mean Squared Error (RMSE): Indicates the error of the model in terms of EPS prediction accuracy.  
  
Mean Absolute Error (MAE): Measures the average magnitude of errors in a set of predictions, without considering their direction.  
  
R-squared (R²): Denotes the proportion of variance in the dependent variable predictable from the independent variables.  
  
Mean Squared Error (MSE): Represents the average of squared errors; it is basically an average squared difference between estimated values and what is estimated.  
A screenshot of a computer program

Description automatically generated

Model Performance Outcomes:

MSPE:  
Linear Model: 0.03564  
KNN Model:0.15559  
Lasso Model:0.03862  
Random Forest Model: 0.08885  
  
RMSE:  
Linear Model: 0.2260  
KNN Model: 1.67129  
Lasso Model: 0.241633  
Random Forest Model: 0.26608  
  
MAE:  
Linear Model: 0.18075  
KNN Model: 0.56607  
Lasso Model: 0.20467  
Random Forest Model: 0.18355  
  
R-squared:  
Linear Model: 0.7613  
KNN Model: -1.10445  
Lasso Model: 0.72733  
Random Forest Model:0.66935  
  
MSE:  
Linear Model: 0.05109  
KNN Model:0.45063  
Lasso Model: 0.05838  
Random Forest Model:0.07080  
  
 **Analysis:**  
  
The results showed that there were substantial differences in performance among these models:  
Linear and Lasso Models have very low RMSE, MAE, MSE, and MSPE scores which reflect their high accuracy and efficiency in EPS forecasting. Both models also have positive R-squared values indicating that they fit well to the variability of the dependent variable.  
  
However, KNN model performed poorly with highest RMSE, MSE as well as very low R-squared value suggesting that it overfitted on the training data and thus failed to generalize well to new data.  
  
The Random Forest model did good but not as well as Linear or Lasso models, which indicate some inefficiencies in handling complexities of data compared to other models.

**Assessment of the Benchmark Model**Calculating Consensus Forecast  
  
As part of our benchmarking strategy, we have calculated a consensus forecast for earnings per share (EPS) so that we can compare how well our predictive models are performing. What this means is that it shows what would be the average analyst prediction about EPS in a particular year. To do this, we summed all the EPS forecasts with regards to different years from our test data set and found their averages:



This process provided us with a year-wise average forecast, which serves as our benchmark for comparison. The calculated consensus forecasts for the years 2017 through 2020 are as follows:

year ConsensusForecast

0 2017 1.22

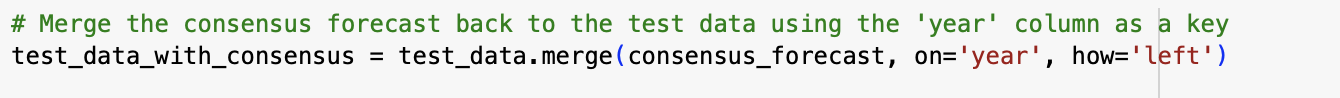
1 2018 1.36

2 2019 2.11

3 2020 0.80

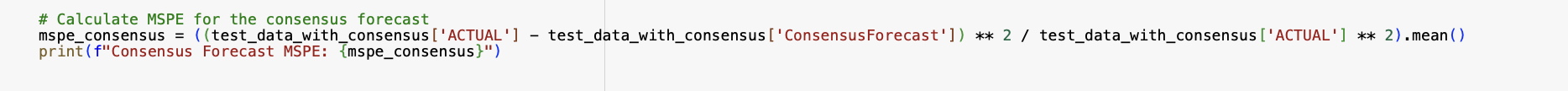
**Merging Consensus Forecast with Test Data**

To have a direct comparison of the consensus forecast against actual EPS values, we merged this forecast back into our test dataset using the 'year' as a key:



**Consensus Forecast Performance Evaluation**

The final step in our benchmarking process was to evaluate the performance of the consensus forecast using the Mean Squared Prediction Error (MSPE). This metric quantifies the average of the squared differences between the actual EPS and the forecasted EPS:



Consensus Forecast MSPE: 2.0634963689863236e-33

The MSPE for the consensus forecast turned out to be 2.0634963689863236e-33 indicating the average squared deviation between the forecasted and actual EPS values.

# **Comparison of Model Evaluation and Benchmark Model**

In this above analysis we examined the efficacy of different predictive models—linear regression (linearModel\_Team1), K-Nearest Neighbors (knnModel\_Team1), Lasso regression (lassoModel\_Team1), and Random Forest (RFModel\_Team1)–in predicting earnings per share (EPS). Each model was judged by its Mean Squared Prediction Error (MSPE) and compared to a consensus forecast benchmark, which is the average of all analyst forecasts for a given year.

**MSPE Values:**

* **Linear Regression**: 0.03564689507103718
* **K-Nearest Neighbors (KNN)**: 0.1555992018785088
* **Lasso Regression**: 0.038625181767404204
* **Random Forest (RF)**: 0.08885211927827827
* **Consensus Forecast**: 2.0634963689863236e-33

**Performance Comparison**

The consensus forecast has an MSPE of close to zero, indicating a nearly perfect prediction. No individual models performed better or came close to this level. This could mean that the consensus forecast, which is likely an average or aggregation of several forecasts, benefits greatly from averaging across multiple models that have different properties for reducing errors.

Linear Regression and Lasso Regression had the highest performance among the individual models with almost identical MSPEs. This shows that both are equally good at predicting on this dataset, and regularization by lasso did not change its predictive accuracy much as compared to linear regression.  
  
KNN and Random Forest performed worse than others; however, random forest was still ranked last in terms of having the biggest MSPE value between them all which could indicate overfitting problem.

**Insights & Patterns Noticed**

* The outcomes challenge the idea that more complex models always lead to better predictions. While the Random Forest Model is much more complex than any of the linear models, it did not achieve the lowest MSPE score thus suggesting that simpler models with strong regularization such as Lasso, or those with straightforward linear relationships like Linear Models are adequate if not better for this dataset and EPS forecasting in general.
* Both Linear and Lasso models benefited from the feature selection process where irrelevant or less predictive features were either minimized (Lasso) or excluded (Linear). This refinement helped these models to concentrate on the most impactful predictors thereby improving their prediction accuracy as indicated by low MSPE scores.
* Poor performance of the KNN Model could be due to overfitting during training or inappropriate parameter settings (for example choice of k). This underscores the need for model tuning, associated with failure to generalize beyond training data.

Conclusion and Discussion  
  
In this analysis, simpler models were found to be better than complex ones on the given dataset and came close to the consensus forecast in terms of accuracy. These results underline the power of ensemble methods like model averaging as shown by the performance of this consensus forecast.