**Final Project Report Final Round**

Group partner:

Cheng Zhang (2054180)

Hongkun Liu (2068590)

We have double checked the output of the dataset.

**Inventory**

[1. Introduction 2](#_Toc164353387)

[2. Dataset Overview 2](#_Toc164353388)

[2.1 Data Characteristics 2](#_Toc164353389)

[2.2 Data Preparation 2](#_Toc164353390)

[2.1.1 Lag 2](#_Toc164353391)

[2.1.2 Normalization 3](#_Toc164353392)

[2.1.3 Principal Component Analysis 3](#_Toc164353393)

[2.3 Training and Testing Sets 4](#_Toc164353394)

[3.Regression Models 4](#_Toc164353395)

[3.1 Linear Regression 4](#_Toc164353396)

[3.1.1 Model Description 4](#_Toc164353397)

[3.1.2 Data Preprocessing 4](#_Toc164353398)

[3.1.3 Evaluation 5](#_Toc164353399)

[3.2 Polynomial Regression 5](#_Toc164353400)

[3.2.1 Model Description 5](#_Toc164353401)

[3.2.2 Data Preprocessing 5](#_Toc164353402)

[3.2.3 Evaluation 6](#_Toc164353403)

[3.3 Lasso Regression 6](#_Toc164353404)

[3.3.1 Model Description 6](#_Toc164353405)

[3.3.2 Data Preprocessing 6](#_Toc164353406)

[3.3.3 Evaluation 6](#_Toc164353407)

[3.4 Random Forest 7](#_Toc164353408)

[3.4.1 Model Description 7](#_Toc164353409)

[3.4.2 Data Preprocessing 8](#_Toc164353410)

[3.4.3 Evaluation 8](#_Toc164353411)

[3.5 Decision Tree Regression 9](#_Toc164353412)

[3.5.1 Model Description 9](#_Toc164353413)

[3.5.2 Data Preprocessing 9](#_Toc164353414)

[3.5.3 Evaluation 10](#_Toc164353415)

[3.6 GB Regression 10](#_Toc164353416)

[3.6.1 Model Description 10](#_Toc164353417)

[3.6.2 Data preprocessing 10](#_Toc164353418)

[3.6.3 Evaluation 11](#_Toc164353419)

[3.7 Overall Results 11](#_Toc164353420)

[4. Other Suggestions 12](#_Toc164353421)

[4.1 Filter Methods 12](#_Toc164353422)

[4.2 Wrapper Methods 12](#_Toc164353423)

[4.3 Implementation 12](#_Toc164353424)

[4.4 Conclusion 12](#_Toc164353425)

# Introduction

The maritime shipping industry, crucial for global trade, relies on precise forecasting to optimize operations and ensure timely delivery of goods. In this context, our project embarked on leveraging advanced regression modeling techniques to predict future values in maritime shipping data. This endeavor aimed to enhance operational efficiency and navigate the complexities of maritime logistics with data-driven insights.

The project utilized a comprehensive dataset spanning from 1991 to the present day, marking a significant period for analysis due to the evolution of global trade patterns and shipping technologies. The dataset's time-series nature, incorporating monthly samples, presents a unique opportunity to apply predictive modeling for forecasting future outcomes based on historical trends.

# 2. Dataset Overview

## 2.1 Data Characteristics

**Type**: Time series

**Number of Features**: 463, including time as the first column, representing various aspects relevant to maritime shipping.

**Number of Samples**: 397, each corresponding to a month's data from 1991 onwards.

**Target IDs for Prediction**:

**Good Result**: 542236, 67321

**Mid Result**: 549295

**Bad Result**: 41108, 541982

Each row in the dataset encapsulates the monthly data of all features, serving as the independent variables for the model. The dependent variable or the target for each month is the intended value for the subsequent month, residing in the specified target column. This structure allows for a direct application of regression models to forecast future values based on the provided historical data.

## 2.2 Data Preparation

To prepare the dataset for modeling, it's pivotal to generate **X** (features) and **Y** (target) matrices accurately. **X** encompasses all features for a given month, while **Y** is a vector representing the target value for the next month. This preparation ensures that each sample in **X** aligns with its corresponding label in **Y**, crucial for training predictive models.

### 2.1.1 Lag

The lag parameter in regression, especially in time series analysis, plays a crucial role by incorporating the concept of time delay into the modeling process. It is used to account for the influence of past values on current values, a common phenomenon in sequential data. For this dataset, N rows can be concatenated to generate lag N and first N rows of X and last N rows of Y should be deleted to have the same row as the input. The snapshot of sample code for generating lag 1 is as follows:

文本

描述已自动生成

### 2.1.2 Normalization

Normalization in regression is a critical preprocessing step involving the scaling of input features to ensure they have a standard scale or range. This process is essential because many machines learning algorithms, including regression models, perform better or converge faster when features are on a relatively similar scale and close to a normally distributed shape.

There are many kinds of techniques for Normalization, for this dataset, we select X Min-Max to scale all the features in X to a fixed range(0 to 1). Following is the snapshot of implementing this kind of approaches:

文本

描述已自动生成

### 2.1.3 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction while preserving as much variance as possible. It's commonly used in data preprocessing for machine learning and data visualization tasks. It is worthy to mention that normalizing the data is a critical step before applying PCA, as PCA is sensitive to variances of the initial variables. Following is the snapshot of implementing this kind of approaches:

文本

描述已自动生成

## 2.3 Training and Testing Sets

 Given the time series nature of the data, special attention was paid to avoid data leakage during model training and evaluation. The dataset was split into training and testing sets, with the last 36 months reserved for testing to assess the model's performance on recent data. This split respects the temporal sequence of the dataset, ensuring that the model learns from past data to predict future outcomes.

In summary, the dataset's comprehensive coverage of the maritime shipping industry, combined with careful preparation and consideration for time series analysis, sets a solid foundation for applying regression modeling techniques. The following sections of the report will delve into the methodology of each model, predict results, and analysis, providing insights into the predictive modeling process and its implications for the maritime shipping industry.

# 3.Regression Models

## 3.1 Linear Regression

### 3.1.1 Model Description

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. It is one of the simplest and most used techniques in predictive modeling and statistical analysis. The linear regression model to predict these IDs is obviously a multiple linear regression model since two or more independent variables are involved to predict the dependent variable. The equation of a simple linear regression line is typically expressed as:

**y=b0+b1\*x1+b2\*x2+…+bn\*xn + E**

Where y represents the dependent/target variable, x1, x2 …xn are independent variables, b0, b1, b2 ... bn are the coefficients which represent the influence each independent variable has on the dependent variable, and E is the error term.

### 3.1.2 Data Preprocessing

Feature Selection can be applied by determining which variables (features) are most relevant to the prediction of the dependent variable. This dataset has more than 400 features in total, making it necessary to reduce the dimensionality of the data by transforming the original variables into a new set of variables called principal components.

### 3.1.3 Evaluation

图表, 折线图

描述已自动生成

The comparison reveals that Linear Regression with PCA outperforms the version without PCA across all metrics. With PCA, the model achieves a high of 91.02% and a respectable low of 70.18% accuracy, averaging at 79.95%. Without PCA, the model's best drops to 71.55%, and its worst plummets to -1383.22%, dragging the average to -468.75%. These figures suggest PCA's critical role in enhancing model reliability and mitigating prediction errors.

## 3.2 Polynomial Regression

### 3.2.1 Model Description

In cases where the relationship between the independent and dependent variables might be non-linear, polynomial features (powers of the feature variables) can be created to attempt to capture this nonlinearity. The general form of a polynomial regression model is:

**y=b0+b1\*x+b2\*x2+…+bn\*xn + E**

Where y represents the dependent/target variable, x is the independent variable, b0, b1, b2 ... bn are the coefficients of model which represent the influence each independent variable has on the dependent variable, and E is the error term.

We assume there exists a non-linear relationship between these variables and implement a polynomial regression model and evaluate the accuracy.

### 3.2.2 Data Preprocessing

Normalization is particularly important in polynomial regression because polynomial regression involves high powers of input features. When we raise features to high powers, their values can become exceedingly large if the original feature values are not within a bound range. This can lead to numerical instability during the calculation, as floating-point precision can become a significant issue. Normalization helps keep the values in a manageable range, reducing the risk of numerical errors.

In this project, normalization refers to the process of Min-Max scaling input data so that it falls within a range between 0 and 1.

### 3.2.3 Evaluation

图表, 折线图

描述已自动生成

The chart compares Polynomial Regression models, with and without normalization. Normalization significantly improves the worst accuracy from -390.31% to 64.94%, indicating greater model stability. The average accuracy sees a boost from -139.35% to 75.24%, reflecting overall performance enhancement. The best accuracy slightly drops with normalization, from 69.78% to 89.38%, but the trade-off results in a more reliable model across varying conditions.

## 3.3 Lasso Regression

### 3.3.1 Model Description

Lasso regression, short for Least Absolute Shrinkage and Selection Operator, is a type of linear regression technique used for variable selection and regularization. It's particularly useful when dealing with datasets that have many features, where some of these features may be irrelevant or redundant.

### 3.3.2 Data Preprocessing

The key characteristic of lasso regression is that it tends to shrink the coefficients of less important features exactly to zero, effectively performing variable selection by eliminating these features from the model.

Principal Component Analysis (PCA) is not inherently required for Lasso regression, but it can be a useful preprocessing step in certain situations.

### 3.3.3 Evaluation

As we can see from the results of accuracy, PCA is not always necessary or beneficial for Lasso regression. Whether PCA should be used as a preprocessing step depends on factors such as the data's nature, the multicollinearity, the interpretability requirements, and computational considerations. In some cases, Lasso regression may perform well without PCA preprocessing, especially if the number of predictors is not excessively large.

图表, 折线图

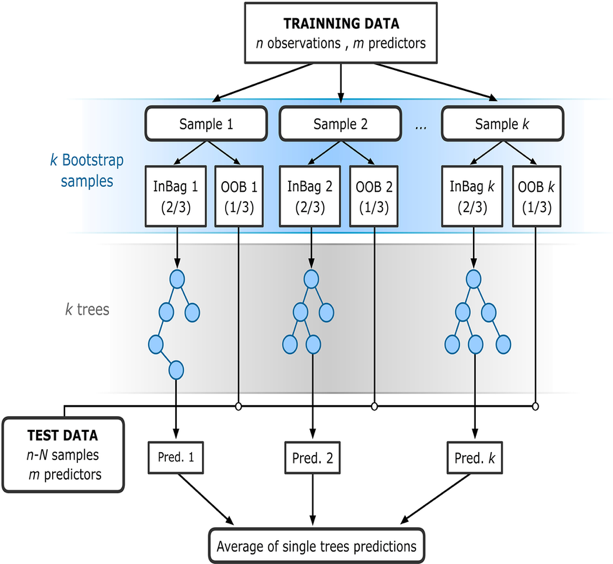
描述已自动生成

Across all models, the average accuracy tends to increase as more preprocessing steps are added, which demonstrates the benefits of feature scaling, considering temporal dependencies (lag), and dimensionality reduction (PCA) in improving the robustness of the regression model.

## 3.4 Random Forest

### 3.4.1 Model Description

A Random Forest is an ensemble learning technique used for both regression and classification tasks, which operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees' habit of overfitting to their training set. The following is a sample of Random Forest.



### 3.4.2 Data Preprocessing

In this project, we will have two versions of Random Forest Regression. The first version of the model can be configured with a default random state. In addition, we apply embedded feature selection to this model. In the second version of decision tree regression, we train a default model first and then select several features that have an importance greater than median. After that we retrain the model with dataset containing the selected features. Following is python code we used to implement feature selection during data preprocessing phase.

文本

描述已自动生成 3.4.3 Evaluation

图表, 折线图

描述已自动生成

The evaluation of the Random Forest model shows the following:

Best Accuracy: Approximately 95.50%

Worst Accuracy: Approximately 59.59%

Average Accuracy: Approximately 81.49%

These results demonstrate that the Random Forest model has a strong best-case performance but also indicates a fair degree of variability, as seen in the worst-case performance. The average accuracy is relatively high, suggesting that the model generally performs well across various targets.

## 3.5 Decision Tree Regression

### 3.5.1 Model Description

Decision tree regression is a machine learning method that uses a decision tree to model the relationship between a set of features and a continuous target variable. Like decision trees used in classification, decision trees for regression predict the outcome based on input features by splitting the data into subsets using decision rules inferred from the input features.

### 3.5.2 Data Preprocessing

Like random forest, there are two versions of decision tree regression in our project (with or without feature selection). As the code shown below, we fit a decision tree regressor to dataset and use the feature\_importances\_ attribute of the decision tree model to get the importance of each feature. Finally, choosing a threshold and keeping only the features that have an importance above this threshold for further modeling.

文本

描述已自动生成

### 3.5.3 Evaluation

图表, 折线图

描述已自动生成

The chart contrasts two Decision Tree Regression models, with one utilizing feature selection. Feature selection improves the model's worst accuracy dramatically, from 25.16% to 78.44%, and increases average accuracy from 67.13% to 86.57%. The best accuracy slightly decreases with feature selection, but the gains in model consistency and reliability are evident, showcasing feature selection's value in refining model performance.

## 3.6 GB Regression

### 3.6.1 Model Description

GB Regression refers to Gradient Boosting Regression, which is a powerful and widely used ensemble machine learning technique for regression tasks. Gradient Boosting constructs a predictive model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion and generalizes them by allowing optimization of an arbitrary differentiable loss function.

### 3.6.2 Data preprocessing

The snapshot of code for this model is configured with arguments

This parameter ‘n\_estimators’ specifies the number of boosting stages to be run. In other words, it's the number of decision trees that will be built in the ensemble.

The learning rate is 0.1, which shrinks the contribution of each tree. It is used as a trade-off against the n\_estimators; a lower learning rate requires more trees but can lead to a better generalization of the model.

The max\_depth limits the maximum depth of the individual regression estimators. The depth of the tree is an important control for the complexity of the model.



### 3.6.3 Evaluation

图表, 折线图

描述已自动生成

The evaluation of the GB Regression model presents the following outcomes:

Best Accuracy: Approximately 97.67%

Worst Accuracy: Approximately 51.18%

Average Accuracy: Approximately 77.37%

These results depict the GB Regression model as having an excellent best-case performance, which is the highest among the accuracies reported. However, there's a notable drop in the worst-case accuracy, which could indicate some overfitting or lack of generalization across different scenarios. The average accuracy remains reasonably high, reflecting that the model is generally effective for the dataset used.

## 3.7 Overall Results

### 3.7.1 Accuracy of IDs

Each script demonstrates a unique approach to handling the predictive modeling task, varying by regression model type, data preprocessing techniques (like normalization), and the specific target variable selected for prediction. These variations in methodology and target focus provide a broad overview of the modeling efforts undertaken to address the project's objectives. The accuracy results for all five target IDs are summarized as below:

表格

描述已自动生成

The average accuracy of all the models above for Good IDs (542236 and 67321) are 78% and 93%, respectively (Linear Regression and Polynomial Regression are not included because the accuracy is negative). While the accuracy results for ‘41108’ and ‘549295’ are relatively lower, at around 70%. Target ID ‘541982’ has the worst performance on accuracy, only 50%.

The Polynomial Regression model and Linear regression without PCA shows a negative accuracy for the several target IDs, suggesting that the polynomial terms might be overfitting to these specific datasets.

Lasso Regression with normalization and lag1 and Decision Tree Regression with feature selection show strong performance, especially in the "Mid" and "Bad" target IDs, indicating that these preprocessing steps or feature selection methods are beneficial for the model.

Models that include PCA (Principal Component Analysis) or normalization generally show an improvement over the basic versions of the same models, emphasizing the importance of these preprocessing steps.

Random Forest, an ensemble learning method, typically outperforms a single Decision Tree due to its ability to average out errors and reduce overfitting. However, in our report ,a Random Forest perform worse than a Decision Tree. After feature selection, the dataset is simple or when the signal is strong and clear, a single decision tree might be sufficient to capture the necessary patterns. In this case, the ensemble approach of a Random Forest could lead to unnecessary complexity.

### 3.7.2 Performance of Model

As we can see from the table above, the average accuracy of the Decision Tree Regression + feature selection being higher than other methods. This distribution can be attributed to several reasons.

First, the process of feature selection removes irrelevant, redundant, or noisy data from the dataset. By selecting only the most important features, the model can focus on the most significant predictors, reducing the risk of overfitting and improving model performance.

In addition, decision trees are non-linear models and can capture complex relationships between features and the target variable. If the underlying data has non-linear patterns, decision trees may outperform linear models like linear regression.

Lastly, the decision tree model could be the right complexity for the data. If other models are too simple (underfitting) or too complex (overfitting), they might perform worse on average.

The actual and predicted values during the last 36 months only for the best model with the best average accuracy for each target are shown below, where the horizontal axis shows the time and vertical axis shows the values, the blue line shows the actual values, and the red line shows the predicted values.

图表, 折线图

描述已自动生成

图表, 折线图

描述已自动生成

图表, 折线图

描述已自动生成

图表, 折线图

描述已自动生成

图表, 折线图

描述已自动生成

The decision tree model's performance varies across different targets. In some cases, it achieves high accuracy, closely tracking the actual values, while in others, there are notable gaps, especially during periods of volatility or at extreme values. These variances in performance may indicate the model's sensitivity to the features used, the presence of noise in the data, or limitations in capturing complex, non-linear patterns.

# 4. Other Suggestions

## 4.1 Filter Methods

Filter methods, which involve selecting features based on their statistical scores in relation to the output variable are generally faster and less computationally expensive as they do not involve training models. We can also apply correlation coefficients and Chi-squared tests to identify and retain the most relevant features. Despite the simplicity and speed of filter methods, they do not consider the interaction between features, which can be crucial for some datasets.

## 4.2 Wrapper Methods

Beside the filter methods, wrapper methods could be implemented that evaluate subsets of features based on the model performance, making them more effective but also more computationally intensive. a stepwise backward elimination process could be used, where we can include all features and iteratively removed the least significant feature until no improvement in model performance was observed. This method, while more time-consuming, provided a tailored subset of features that optimized our specific models.

## 4.3 Implementation

Both methods were implemented using Python's scikit-learn library. The filter method was straightforward to apply but yielded minimal improvement in model performance for complex interactions. In contrast, the wrapper method, although slower, might resulted in a noticeable enhancement in model accuracy, particularly in the Lasso and Random Forest models. Despite the improvements, the computational cost and time required for the wrapper method were substantial, which might not be feasible for all projects, especially those with extremely large datasets or limited computational resources.

## 4.4 Conclusion

The exploration of feature selection methods allowed us to understand better the trade-offs between model accuracy and computational efficiency. While the filter method is quick and effective for a preliminary reduction of features, the wrapper method provides a more nuanced approach to feature selection, beneficial for models requiring detailed feature analysis. Future work could explore hybrid methods that combine both approaches to balance effectiveness and efficiency.