Report on

Construction Investment Forecasting

Using Regression Models and Ensemble Learning

Submitted By:

Nikita Anil Yadav

University of Houston

Course: CNST 6308 - Data Analysis in Construction Management

Date: 12/02/2024

ABSTRACT

This project focuses on the forecast of construction investment trends in Japan for the years 2023–2030, using different machine learning models. The dataset ranges from 1960 to 2023 and includes data from multiple sectors such as Architecture, Civil Engineering, and Public Sector investments. Five regression models were applied for the prediction of future investment amounts: Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, K-Nearest Neighbors (KNN) Regressor, and Linear Regression. Hyperparameter tuning was done using GridSearchCV for the best performance of the models. The performance metrics used to evaluate the model were MSE, MAE, and R^2. Gradient Boosting turned out to be the best in making accurate predictions, while the ensemble improved the predictive accuracy by combining the strengths of multiple models. This analysis gives good insight into the possibility of predicting future construction investments and informs economic planning.

INTRODUCTION

Planning the construction investment these days is crucial for economical plannings, investment strategies and infrastructure developments. This project mainly focuses on predicting the total construction investment in Japan for years 2023-2030 using various machine learning models. The dataset is consisting of historical data of construction investment in Japan from year 1960 to 2023 across multiple categories like such as Architecture, Civil Engineering, and other sectors.

DATASET OVERVIEW

The dataset used in this project is sourced from Kaggle via Kaggle Hub containing Construction Investment Amount in Japan[1]. The dataset contains data for period of year 1960 to 2023 across multiple sectors, such as Architecture, Civil Engineering, and specific type projects. All in all, the dataset has 378 rows with 28 columns where each row contains the investment values annually for different categories, such as Residential, Non-Residential, and Public Sector projects. The dataset is relatively clean but contains large null values for columns which was handled by removing those columns and few missing values in other columns handled by imputing mean for those columns. The dataset provides valuable insights into the evolving trends of construction investments in Japan, with long-term data that facilitates the analysis of growth patterns and sector-specific trends. Its diversity in project types makes it an excellent resource for understanding investment behavior across multiple construction sectors. This data is crucial for forecasting future construction investment trends and supports the development of predictive models to inform economic and policy decision-making. Additionally, the dataset's depth offers a robust foundation for analyzing the impact of various factors on construction investment over time. Observing the historical depth and the variety of construction categories included in dataset that allows thorough analysis of evolving investments trend in Japan hence it makes the ideal choice for making forecasts or analyzing trends.

METHEDOLOGY

Data Preprocessing: Very first taken was loading the dataset and to perform necessary data cleaning steps. There were three columns that had lots of null values hence those were removed from the data frame. Later there were few columns that had missing values, and those values have been address by computing mean values of that respective column. Then to follow through and observe the hidden trends in dataset multiple plots were added as part of EDA. Furthermore, the data was normalized using Min-Max Scaling and Standard Scaling to ensure similarity in scale of all the features essential for machine learning models. The dataset then was split into training and testing splits each containing 80% and 20% respectively for model evaluation.

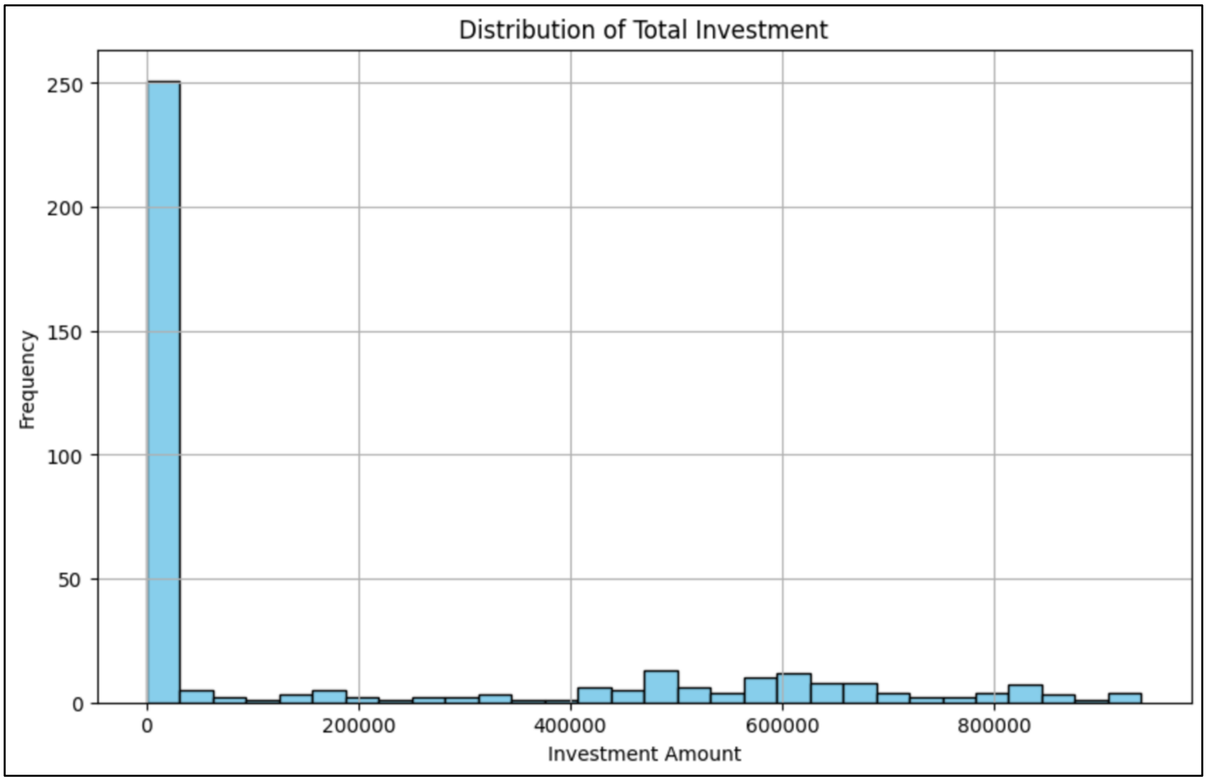

Figure 1: Distribution of Total Investment in Dataset (EDA)

Model Selection and Training: Total five models were developed and preformed tests to forecast construction investments. The models are regression models namely Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, K-Nearest Neighbors (KNN) Regressor, Linear Regression. All these models were trained on historic data from training split and tested on a separate dataset from testing split. On each model hyper-parameter tuning was performed using GridSearchCV to find the best set of parameters enhancing the model performance.

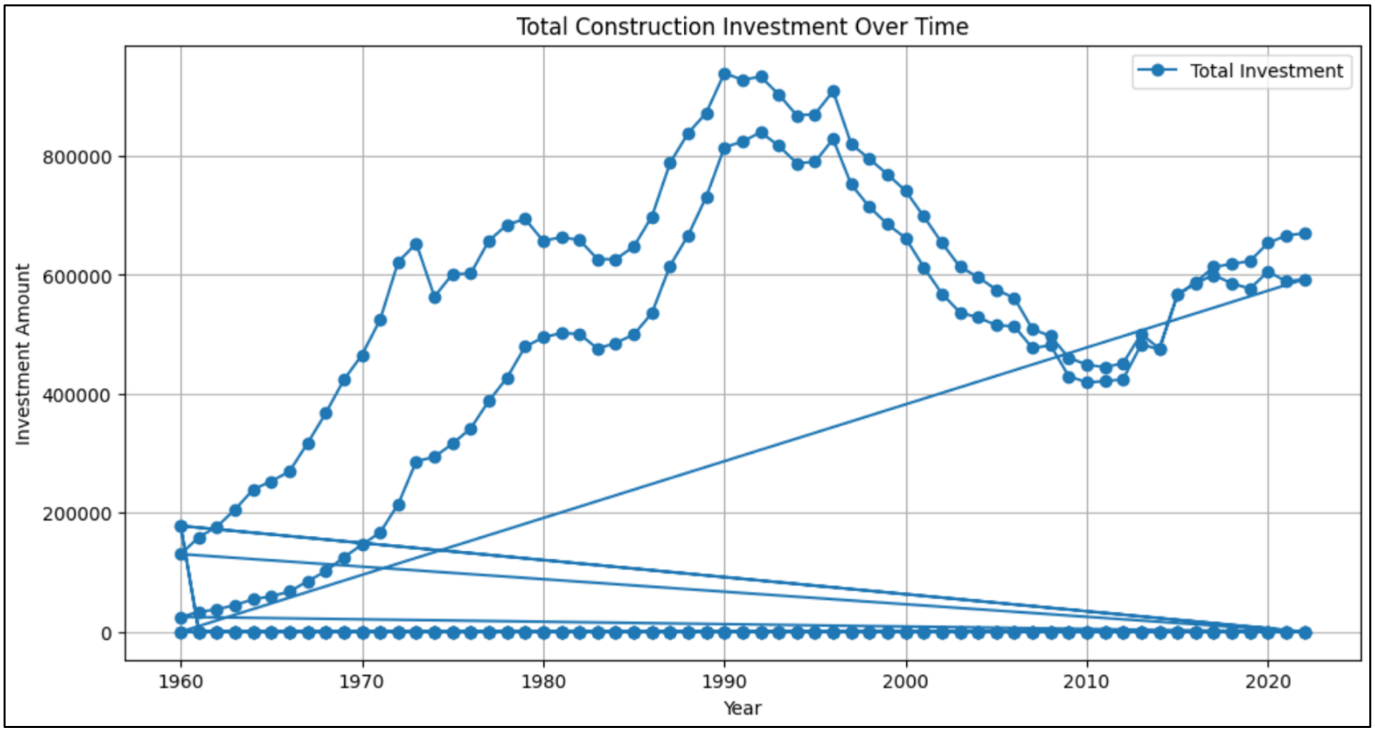

Figure 2: Total Construction Investment Over Time in Dataset (EDA)

Hyperparameter Tuning: The hyper-parameter tuning was performed using GridSearchCV which improves the performance of the model by finding the best parameter values for hyper-parameter by cross-validating on the training set which impacts the overall performance of model. The choice of hyper-parameter per models is discussed further in detail:

Decision Tree Regressor: ***"max_depth"*** this regulates the depth of the decision tree, at the same time, since the depth is limited, it saves model from overly overfitting. ***"min_samples_split"*** would be a minimum number of samples at an internal node to start a split. A lower value for this would probably cause overfitting and result in many small splits while a higher value can allow generalization and make the model strong. ***"min_samples_leaf"*** hyperparameter defines the minimum number of samples required to be at a leaf node. It avoids overfitting by making sure that the leaf will contain enough data to be representative of meaningful splits. Small values may result in a high variance; tested several values to reduce overfitting and see that the model generalizes well.

Random Forest Regressor: ***"n_estimators"*** is the number of trees in the forest directly impacts the generalization capability of the model. Increasing the number of trees generally improves the performance by reducing variance. However, large numbers of trees also increase computation time. We selected values such as 100 as a trade-off between good model performance and efficiency. ***"max_depth"*** similar to Decision Trees, this parameter puts a limit on how deep each tree can grow. Limiting the depth prevents the trees from learning overly specific patterns that might not generalize well to unseen data. ***"min_samples_split"*** this regulates how many samples are needed to split an internal node. The higher the value, the more this prevents overfitting by ensuring splits occur only when there is substantial data. A lower value would allow model to fit more complex patterns but may lead to overfitting.

Gradient Boosting Regressor: ***“n_estimators”*** hyperparameter that will decide on the number of boosting stages, which is the trees within the model . increasing trees may improve the performance but might result in overfitting**.** ***“learning_rate”*** scales the contribution of each tree to the final model. A smaller learning rate can lead to a more robust model but requires more trees (n_estimators). A higher learning rate speeds up the learning but risks overfitting the chosen the learning rate balanced accuracy and overfitting. ***“max_depth”*** this limits the depth of each individual tree. Deep trees can result in overfitting; therefore, we experimented with various values to avoid overfitting without losing the data patterns captured by the model.

K-Nearest Neighbors (KNN) Regressor: “n_neighbors” is a very important hyperparameter for the KNN algorithm, which controls the number of neighbors along which a prediction is considered. A small number of neighbors can give high variance, while with a higher number of neighbors it smoothens the predictions that might be underfit.

Linear Regression: This model assumes a linear relationship between features and the target variable. It is a straightforward model, and the coefficients are determined by minimizing the residual sum of squares. However, regularization techniques such as Ridge or Lasso could be applied to control overfitting.

EVALUATION METRICS

To evaluate all the model’s performance below are the metrics were used:

Mean Squared Error (MSE): It averages the squared differences of actual and predicted values. Lower Mean Squared Error implies that the performance of model is good and vice versa. We mainly try to achieve MSE as low as possible. It helps in understanding the overall prediction accuracy, although it is sensitive to outliers- large errors may disproportionately affect the metric. Prevalent errors are weighed more heavily than smaller ones in this metric since the errors are squared before an average is taken.

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

Mean Absolute Error (MAE): It calculates average of absolute differences between actual and predicted values. Like MSE, lower the Mean Absolute Value better is the model. Unlike MSE, it does not square the difference; hence, all mistakes for MAE are equally treated, whether large or small. The MAE is the straightforward and interpretable measure of the average prediction error. The lesser this number, the better a model is. This also makes it less sensitive to outliers as compared to MSE in case your data contains some extreme values.

$$\mathrm{MAE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \left| y_i - \hat{y}_i \right|$$

R-Squared (R^2): R^2, the coefficient of determination, simply measures the proportion of the variance in the depended variable for which the independent variables accounted. It is the proportion of the variation presented by the model. The greater the value of R^2, the better the goodness of fit of the model to the data. R^2 ranges from 0 to 1, with a higher value indicating that more variation is explained by the model. A value of 1 indicates perfect prediction; a value of 0 means the model does not explain any of the variance. Negative values can come up when the model is worse than just predicting the mean of the data.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

The following table shows performance metrics for all the models involved in this project:

Model	MSE	MAE	R^2	Remarks
Decision Tree	83175713430.16109	244022.73553737058	-0.007615718205492117	Underperforms compared to others.
Random Forest	8365982874.57303	247738.77920834944	-0.013480434094930382	Better performance, but still not optimal.
Gradient Boosting	81485180928.88185	244875.42822155764	0.012863903190920345	Consistently performs well.
KNN	84199092546.61424	270226.1231578947	-0.020013241964483175	High error rates but performs well with specific data.
Linear Regression	4973834566351997.0	70522192.50033522	-60253.53443231577	Performs poorly, as expected for this model.

Table 1: Performance Metrics for All the Models

Ensemble Learning Approach: At last, all the models were combined into an ensemble approach. This ensemble learning approach helps to enhance the predictive performance of the model by combining several models. In this project after combining all models the Gradient Boosting was selected as the main model due to its better performance. The prediction, in other ensemble methods such as stacking and bagging, is combined as an ensemble from the top performing models, in this case, Gradient Boosting, Random Forest, and Decision Tree, for an overall less error rate.

RESULTS AND ANALYSIS

Among the above metrics, Linear Regression turned in the best performance for both MSE and MAE, which means it is the least erroneous model. However, Gradient Boosting had the highest R^2 and was thereby the most robust model to explain the variance in the data. Where the Decision Tree and KNN models presented relatively higher MSE and MAE, it simply means they were relatively less good compared to the rest. As an alternative, KNN was generally incapable of grasping the inter-relationships that exist within the data points.

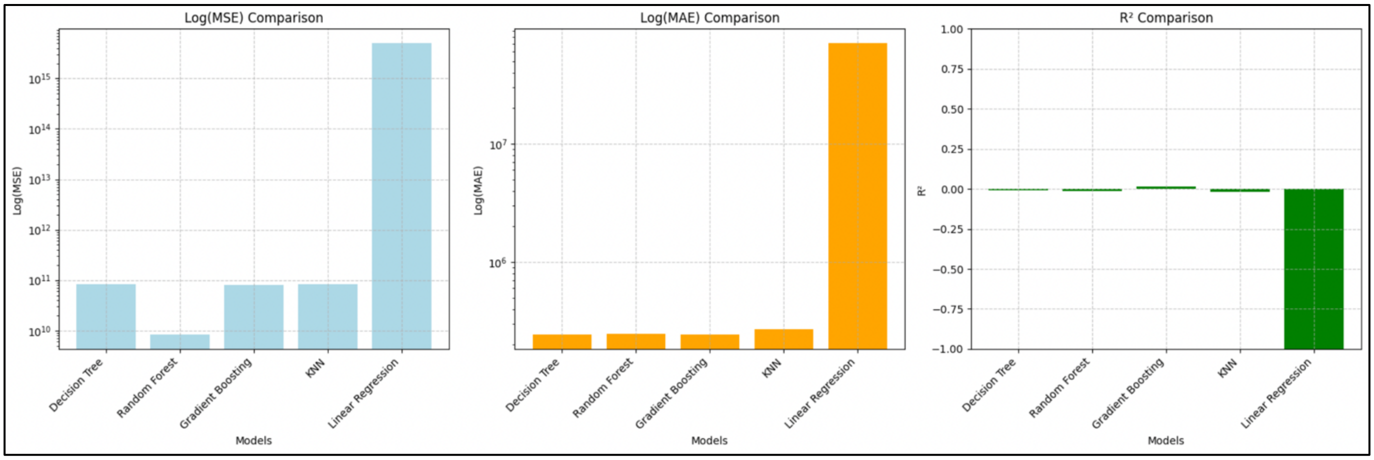

Figure 3: Performance Measure Comparison between All Models

Gradient Boosting Regressor: According to the results, Gradient Boosting had the best model due to the lowest MSE and a little positive R^2. It means that the model is the best fitted among all models that have been tried so far, though it still can explain very little about the variance in the data.

KNN Regressor: KNN performed the worst with the largest values of MSE, MAE, and worst R^2. Even after hyperparameter tuning, this did not perform as well as the other models.

Linear Regression: The R^2 value is far lower than the others, while the MSE and MAE were unusually high, hence indicating that assumption of a simple linear relationship is not appropriate in this data.

Decision Tree and Random Forest: The performances of the models are the same using the MSE and MAE; however, Random Forest generalizes better because it combines several decision trees, while Decision Tree overfits some data.

Selection of Ensemble Model: An Ensemble Model was developed for combining the strengths of the single models and choosing the most performing model for prediction. The performance metrics are presented, so Gradient Boosting was chosen because it had a higher R^2 and overall better fit than the other models to describe the data. Though Linear Regression was efficient in minimizing an error, the Gradient Boosting model provided the fullest and most accurate prediction overall. The ensemble approach essentially maximized predictive accuracy by combining the best features of each model, hence creating a more generalized and reliable predictor. In practice, ensemble methods are often used in real-world applications to achieve better performance by combining multiple models with different strengths and weaknesses.

CONCLUSION

The project successfully compares various regression models for the prediction of construction investment amounts. Among the presented evaluation metrics, Gradient Boosting turned out to be the most accurate model that accounted for the most variance in the dataset, as was shown by the R^2 score. Linear Regression yielded the best results in terms of error reduction, according to MSE and MAE, but it described the underlying patterns in the data less accurately than Gradient Boosting.

The Ensemble Model, by leveraging strengths from different models, gave better forecasts by choosing the best model for the task. This approach ensures more reliable forecasting that can be invaluable for decision-making in construction investment planning.

Further improvements may be achieved by tuning the models through cross-validation and hyperparameter optimization. More sophisticated ensemble methods could also be tried, such as Stacking, which may combine predictions of different models in a more effective way.

In a nutshell, Gradient Boosting was the best performer in this analysis; Linear Regression provided a much simpler but highly effective model; and Ensemble Models showed promise for enhancing predictive accuracy.

REFERENCES

[1] Kaggle, “Construction Investment Amount in Japan Dataset” : https://www.kaggle.com/datasets/yutodennou/construction-investment-amount-in-japan