**sAHANA CHAKRAVARTY**

**ECE 9603/9063b – Data Analytics Foundations**

**Table of Contents**

[1. Description of the selected forecasting problem 1](#_Toc85406781)

[2. Description of available data (attributes, context, quantity…). Clearly indicate what attributes and/or parts you have used. 1](#_Toc85406782)

[3. Short overview of the selected algorithms. 1](#_Toc85406783)

[4. Specifics about how algorithms were applied and the evaluation procedure. 2](#_Toc85406784)

[5. The Comparison of results obtained with different algorithms 3](#_Toc85406785)

[6. Reference 4](#_Toc85406786)

2. Description of the selected forecasting problem

The purpose of this report is to analyze and forecast the resale price of different types of apartments in Singapore using different models.

The selected forecasting problem is to predict resale price of apartments in Singapore with respect to town, flat\_type, street name, storey\_range, floor\_area\_sqm, flat\_model and lease\_commence\_date. This means that the column ‘resale price’ is the target variable and all the other columns except 'month' and 'block' are predictor variables.

The factors that contribute positively to the predictability of the resale price are that the above columns are relevant, consists of 45000 number of rows which is quite large and the forecasts do not affect the resale price of the apartments. Also, the forecasting problem is “normally assumed” and is expected to continue in the similar fashion in the future.

1. Description of available data (attributes, context, quantity…). Clearly indicate what attributes and/or parts you have used.

The source of the data is Singapore Housing and Development Board, December 8, 2016 (Singapore Open Data Licence).

The 10 attributes are month, town, flat\_type, block, street name, storey\_range, floor\_area\_sqm, flat\_model, lease\_commence\_date and resale\_price.

The context of this data holds that the resale apartment prices depend on the neighborhood, street and size of the flat. The dataset consists of a decade-long listing.

In the selected forecasting problem, the column ‘resale price’ is the target variable and all the other columns except 'month' and 'block' are predictor variables.

In the selected model, Outliers are removed with IQR method. Also, encoding of categorical values are done along with standard scaling. Data normalization was not required in the data set.

1. Short overview of the selected algorithms.

The selected models for the problem are

1. **Multivariate Linear Regression**

Multivariate linear regression is a supervised machine learning algorithm that involves multiple data variables for analysis. It consists of one dependent variable and multiple independent variables. The prediction of output is done on the basis of multiple independent variables. It tries to find out how factors in variables respond simultaneously to changes.

The generalized equation for the multivariate regression model is as follows:

y = β0 + β1.x1 + β2.x2 +….. + βn.xn

where n represents the number of independent variables, β0~ βn represents the coefficients and x1~xn, is the independent variable.

2. **Gradient Boosting Regressor**

Gradient Boosting is a machine learning technique used for both classification and regression and they predict models in the form of ensemble or grouping of weak prediction models, commonly decision trees.

Boosted trees grow sequentially by utilizing information from previously grown trees to improve performance. Each tree is fitted to the previous tree’s residuals in a sequence and then a new tree is allowed to focus on the previous tree’s mistakes.

First, we fit a decision tree to data F1(x)=y, then the next decision tree is fitted to the residuals of the previous : h1(x)=y−F1(x). After that a new tree is added to the existing algorithm : F2(x)=F1(x)+h1(x).

The next decision tree is fitted to the residuals of F2: h2(x)=y−F2(x) and then a new tree is added to the algorithm: F3(x)=F2(x)+h2(x). This process continues until cross validation asks to stop.

3.**Decision Tree Regressor**

Decision tree is a predictive modelling technique that uses a set of binary rules to calculate an item’s target value. They break down the data into smaller subsets of data while another associated decision tree is simultaneously developed. This results into decision nodes and leaf nodes. The highest decision node corresponding to the best predictor is called root node.

In decision tree regression, the features of an object is observed and then the model is trained in the similar way of a structure of a tree. Predicted data is in the form of continuous output which means that they are not represented by a discrete set of values.

Accuracy of a model is dependent on the decision of the strategic splits made. Decision trees regression normally use mean squared error (MSE) to decide whether to split a node into two or more sub nodes.

1. Specifics about how algorithms were applied and the evaluation procedure.

**Multivariate Linear Regression** predicts on the basis of one dependent variable and multiple independent variables. In the selected model, resale price is the dependent variable while town, flat\_type, street name, storey\_range, floor\_area\_sqm, flat\_model and lease\_commence\_date are the independent variables.

The generalized equation for the multivariate regression model is as follows:

y = β0 + β1.x1 + β2.x2 +….. + βn.xn

The multivariate regression model is formulated in the model is:

*Estimate****resale price****as a function of****town, flat\_type, street name, storey\_range, floor\_area\_sqm, flat\_model and lease\_commence\_date.***

***=> resale price*** ***= f(town, flat\_type, street name, storey\_range, floor\_area\_sqm, flat\_model, lease\_commence\_date)***

***=> price = β0 + β1. town + β2. flat\_type + β3. street name + β4. storey\_range + β5. floor\_area\_sqm + β6. flat\_model + β7. lease\_commence\_date***

The model was trained with samples of around 45000 and results were obtained from sklearn.linear\_model import LinearRegression.

While 80% of the total dataset was used for training, the rest was used for testing and these splitting of dataset is kept constant throughout the code during training and testing of other models.

For evaluation of linear regression model, evaluation metrics that are considered are R-squared or the coefficient of determination, or the coefficient of multiple determination for multiple regression. 0% represents the model does not explain any variance while 100% explains that the variation in response variable lies around its mean. The other metrics considered is Root Mean Squared Error.

**Gradient Boosting Regressor** predict models in the form of ensemble or grouping of weak prediction models, commonly decision trees. The results in the model is got by using sklearn.ensemble import GradientBoostingRegressor with 1000 regression trees of depth 7.

The various parameters used in this model are

1. n\_estimators: the number of boosting stages.

2.max\_depth : limits the number of nodes in the tree. The best value depends on the interaction of the input variables.

3.min\_samples\_split : the minimum number of samples required to split an internal node

4.learning\_rate : the amount of contribution of each tree that will shrink

For evaluation of **Gradient Boosting Regressor** model, evaluation metrics that are considered is Root Mean Squared Error and the evaluation process is similar to linear regression model.

**Decision Tree Regressor** breaks downdata into smaller subsets of data while another associated decision tree is simultaneously developed. Accuracy of a model is dependent on the decision of the strategic splits made. Decision trees regression normally use mean squared error (MSE) to decide whether to split a node into two or more sub nodes.

In the selected model, result is obtained from sklearn.tree import DecisionTreeRegressor with mean absolute error, random splitter and a max depth of 7.

The parameters used here are as follows:

1.criterion: ‘mae’ or mean absolute error, which minimizes the L1 loss using the median of each terminal node.

2.splitter: is a strategy to choose the split at each node. In this model, “random” is chosen to choose the best random split.

For evaluation of **Decision Tree Regressor**, evaluation metrics considered is Root Mean Squared Error and the evaluation process is similar to linear regression model.

1. The Comparison of results obtained with different algorithms

The Comparison of results obtained from the selected algorithms are

|  |  |  |
| --- | --- | --- |
| Algorithm | R2 | RMSE Test |
| Linear Regression | 0.4565 | 69517.048 |
| Gradient Boosting Regressor | NA | 72319.466 |
| Decision Tree Regressor | NA | 71304.649 |

From the above results we can see that the Linear regression model has outperformed in RMSE Test. Also, the R2 score for Linear regression is less than 1.

However, since we have not used tuning, it is difficult to say which is the best model.

1. Reference

|  |
| --- |
| https://scikit-learn.org/stable/auto\_examples/ensemble/plot\_gradient\_boosting\_regression.html |
| https://en.wikipedia.org/wiki/Gradient\_boosting |
| https://bradleyboehmke.github.io/HOML/gbm.html#gbm-gradient |
| https://en.wikipedia.org/wiki/Decision\_tree\_learning |
| https://gdcoder.com/decision-tree-regressor-explained-in-depth/ |
| https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html |