London Housing Price Prediction

Helen Lai

London Housing Price Prediction

A review of literature has shown that housing prices are often modeled using regression models, whereby structural and locational attributes are used to explain variances in housing prices. This is justified on the basis that houses often share similar structural features with their nearby properties (Basu & Thibodeau, 1998). In addition, poverty index has also been used to identify at-risk housing (Margulis, 1998). Previous research on housing price prediction has taken an econometric approach, ordinary least squares (OLS). However, recent research on house prediction in Santiago has shown that machine learning algorithms, especially Random Forest have yielded superior performance than the traditional approach. The current research aims to extend the literature by examining whether compared to other classification algorithms, Random Forest will perform better in explaining housing price variances in London. The study also aims to explore the predictive power of a series of independent variables: deprivation index, quality rating, distance to station, the venue categories and the geographical coordinates of the nearby properties.

Method

Data[1]

The house price and geographical coordinate datasets are directly downloaded online. The merged dataset covers 527 ward and 273567 house price records. The merged dataset contains the dependent variable: London housing price and the candidate independent variables: index of multiple deprivation, quality rating, distance to station. The other two candidate independent variables: nearby venue category and geographical coordinates of nearby venue are fetched using Foursqaure API. Except nearby venue category data, all data was obtained in its numerical form,

Procedure.

Data Wangling

After importing the two downloaded datasets, initial data cleaning is carried out to select the relevant columns and drop any missing values in London housing price dataset.
When grouping the dataframe by ward code, it was shown that 267 entries existed for each ward code. It is necessary to reduce it to one datapoint per area code, as the ward code column is used as the primary key for later merging which must contain unique values. A closer inspection of single area code shows that each ward has three outcome measures (medium, mean, sales) and each of them were measured at different timepoints. Only the most recent data and its corresponding mean value is selected. The dataframe is grouped by area code again to check if each ward code contains only one entry. Since the algorithmic calculations will be carried out on the value column for its conversion to categorical variables, it is important to make sure that at this stage, the price values exist in its numerical form. However, an examination of the column datatypes shows that the housing price values exist in the form of string. The comma symbol of each price value is removed as data type conversion will not proceed if the cells contain any symbol. After converting the price values into numerical datatype, further data cleaning is carried out to drop any missing value and check for unwanted duplicates. The resulting housing price dataset is then merged with the London geographical coordinate dataset by ward code. The merged dataframe contains 168210 rows and 6 columns. It was found that values in latitude and longitude columns are very precise, which to some extent, is advantageous, as it narrows down to very specific areas. However, when the areas are too specific, it might result in duplications and significant computation time when fetching the corresponding nearby venue data. Therefore, we

shall try shortening it further by rounding up to 2 decimal places as this is also the standard format used in Foursquare API. The resulting dataframe is grouped by longitude and latitude. If each unique set of geographical coordinates contains multiple price values, the average of which is then calculated. The shortened dataframe contains 1809 rows

Exploratory Data Analysis

Prior to fetching nearby venue data through Foursquare API, exploratory data analysis is carried out for two main purposes: first, to modify the frequency distribution of the feature variable (housing price) if necessary and second, to perform first round of feature selection by examining the relevance of three candidate feature variables: distance to station, deprivation index and quality rating. Such an order is preferred because dropping rows containing certain outliers will alter the number of geographical coordinates used for API calls.

With regard to the first aim, the distribution plot of housing price value is produced alongside its skewness and kurtosis. Log-transformation and outlier removals are performed to reduce the skewness if necessary. Outlier removal is carried out by first normalizing the data and then removing data points that have z score greater than three. Regarding to the second aim, the relevance of each existing candidate features is examined by plotting it against the price values. The strength of association and its corresponding p-value is also calculated using the linregress package. Features that are significantly associated with housing prices are selected for modelling.

Data Sourcing and Preparation for Machine Learning

After the first round of feature selection, the remaining geographical coordinates are fed into Foursquare API to source the nearby venue category and its corresponding geographical coordinates. The resulting venue categories are converted into 465 dummy variables through one-hot encoding. The whole dataset is split into training and testing set (75/25). An alternative

to such approach is to split the data into training, validation and testing sets. Such an approach will drastically reduce the sample size that is used for model building. This might negatively influence the model performance of the non-parametric models (i.e. KNN and RandomForest) at the modelling stage. However, without the validation set, hyper-parmeters tuning is needed to be conducted on the testing set which runs the risk of overfitting. Such limitation is overcome by the use of 5-fold cross-validation where training set is split into five smaller sets. Four of which will be used as the training data and the resulting model is validated on the remaining fold. Such process is iterated five times and average accuracy score is computed for selecting the best hyper-parameters. The rationale of using accuracy score as the evaluation criteria for model training is provided in the next section. At this stage, only training set is converted into numpy array as such conversion will remove the column names that are necessary for feature selection on the testing set. Prior to fitting any classification model, the baseline accuracy is first established by calculating the null accuracy, which is the predictive accuracy that could be achieved by always predicting the most frequent class (baseline accuracy = 33.83%).

Feature Selection

The second round of feature selection is carried out by comparing the relative contribution of each predictor using Random Forest's inbuilt feature importance measure. The importance value of each predictor is the total amount that Gini index is decreased by splitting over the predictor, averaged over all the trees. A larger value indicates a higher feature importance. After fitting the Random Forest model using the training set, the feature importance value for all features are calculated using 5-fold cross-validation. Since there is no built-in cross-validation method for feature importance measure, the feature importance calculation is iterated through manually splitting the data into 5 folds. A dataframe that contains a list of feature names

and its corresponding average importance value is then generated for feature selection. An arbitrary cut-off threshold (importance value >0.005) was set to remove the redundant variables.

An alternative to such a Tree-based approach is Recursive Feature Elimination which eliminates worst performing features on a model one after one until the best subset of features are known. In sklearn such an approach is combined with cross-validation which also computes the best number of selected features. Such an approach seems superior to the arbitrary cut-off point adopted in the current study. However, since it is computationally very expensive, Recursive Feature Elimination is not suitable for large number of features which is the case of the current study. Therefore, the use of tree-based feature importance measure is justified. Apart from the Random Forest algorithm, the Decision Tree algorithm is also considered a type of tree-based approach that could also be used to generate feature importance measure. However, although results Decision Tree algorithm are more interpretable, they are prone to overfitting and tend to have lower predictive accuracy. Random Forest overcome this problem by splitting the bootstrapped training sample using random subset of predictors as split candidates. Therefore, the use of Random Forest is considered more appropriate for the current study.

Model Building

After the second round of feature selection, Random Forest is applied again to test the accuracy improvement using 5-fold cross validation. If feature selection yielded reasonable amount of improvement in predictive accuracy, the rest three classification models are then built upon the selected features. Since the current dataset contains relatively balanced classes (low:5932, medium: 5941, high: 5898), the accuracy score is used as the scoring method for cross-validation at the model building stage as it works well on balanced data. ROC curve is not provided as it is most suitable for binary classification problems.

For KNN model, GridSearchCV with 5-fold cross-validation is carried out on the training set to select the best number of K. The K parameter represents the number of neighbor points that the model considers when assigning a label to a test observation through majority voting. The entire training data is used to fit the KNN model with the identified best K value, which will be used for later model evaluation. To build the best Support Vector Machine model, the best C parameter was identified using the same procedure, i.e. GrdiSearchCV with 5-fold cross-validation. C is the penalty parameter which represents error term that tells the SVM how much error is bearable. It controls the trade-off between the small classification errors and selecting the hyperplane with largest minimum margin. As a result, different C might lead to different model performance. Therefore, it is important to select the best C parameter by cross-validating the model on different samples at the model building stage. Unlike Random Forest and KNN, SVM is intrinsically two-class. There are two ways in which SVM could be modified for multi-class prediction: the SVC 'one-against-one' approach and the LinearSVC 'one-vs-the rest' approach. Both two versions of the model are implemented with C parameter tuning. The highest accuracy score for the two models are recorded. The approach that yielded a higher accuracy score is selected for later model comparison. For logistic regression, the parameter tuning procedure is slightly different, as in sklearn there is an option to directly search for the best hyper-parameters using the logistic regression model with built-in cross-validation function (LogisticRegressionCV). The accuracy score of the tuned model is computed through 5-fold cross-validation.

Model Comparison

After all classification models are tuned and fitted, the generalizability of the predictive accuracy of each model is examined by applying the four models to the unseen dataset, the

testing set. Both accuracy score and classification report for each model are generated as both of which are used as the evaluation criteria for selecting the best classification model.

Results

Exploratory Data Analysis

Frequency Distribution of the Housing Price Value

For the target variable, London housing price, the mean (M=576056) was about 70000 higher than its medium (Medium=5045090), which is about one fifth of the standard deviation (SD=330092.6). Suh a deviation implies that the distribution of the price values is positively skewed. This is shown more clearly in the distribution plot below. The calculated skewness and

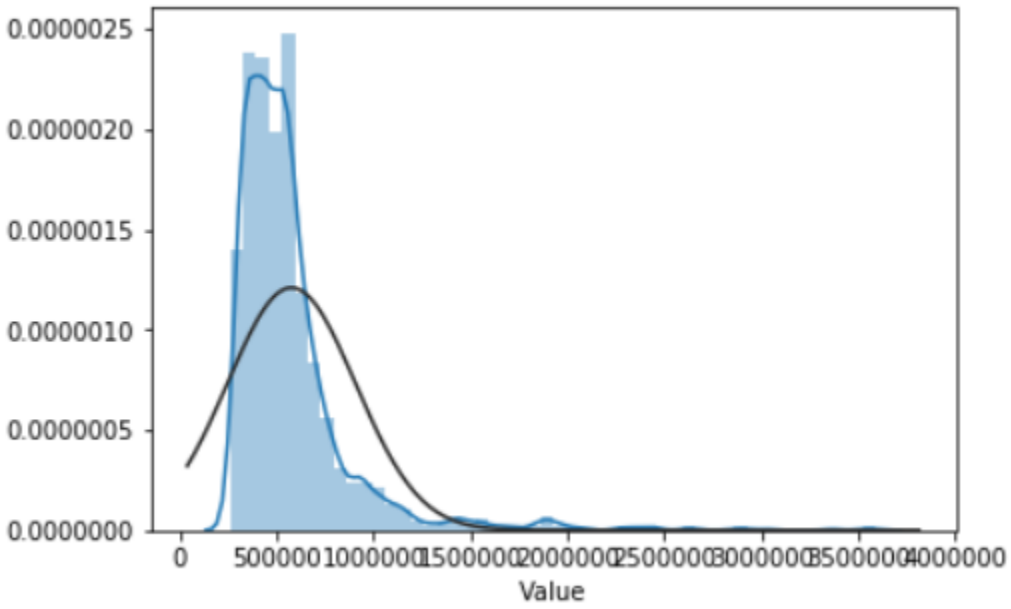

Figure 1.0

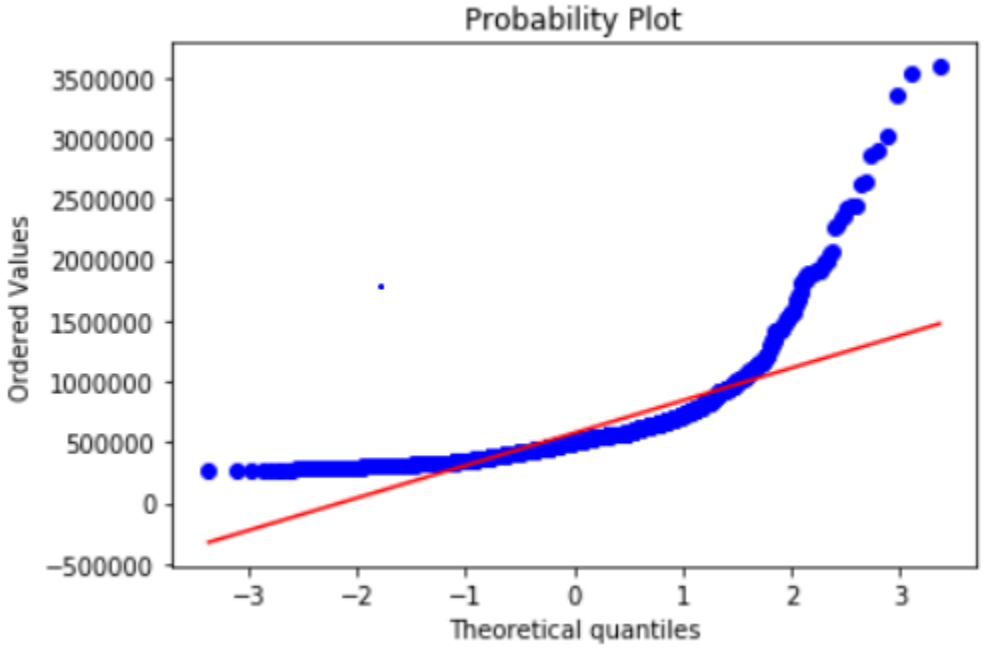

Figure 1.1

Kurtosis was 3.79 and 20.99 respectively. As shown in the probability plot, this might be attributed to the few extremely high values at the outer range. 3.79 is considered a relatively high skewness which will negatively influence the prediction accuracy. The skewness was reduced to 1.203 through log-transformation. To reduce the skewness further, data points that have z-scores greater than three are dropped as outliers. As a result, the skewness was reduced to 0.6744 and

the frequency distribution is reasonably approximated to the normal distribution.

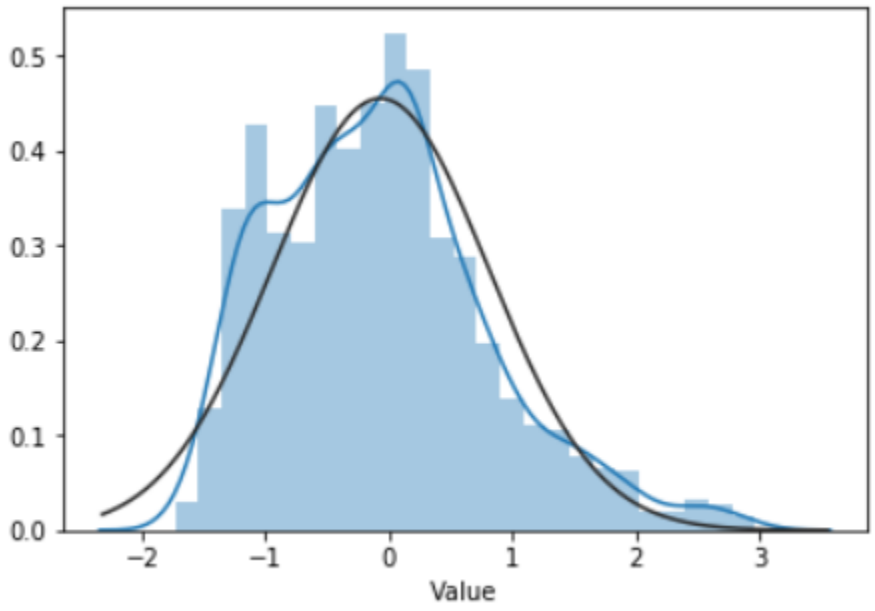

Figure 2.0

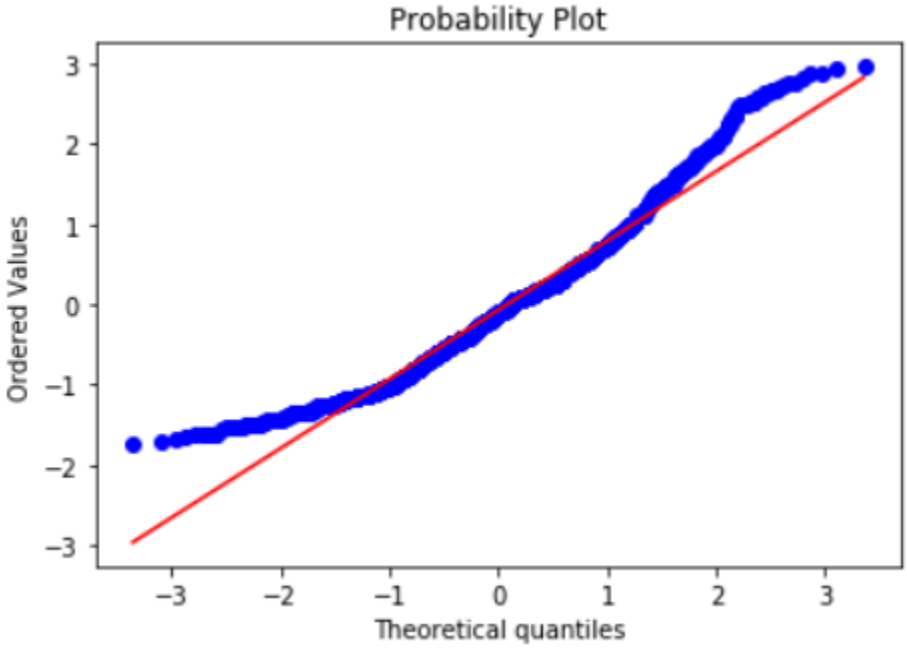

Figure 2.1

First Round of Feature Selection

Results of the Pearson correlation indicated that there was a weak negative yet statistically significant association between log-transformed Distance to Station and the modified price values ($r = -.0174$, $p < .005$). Therefore, the Distance to Station variable is included as one of the feature variables at the modelling stage. Similarly, the association between the second candidate variable, Index of Multiple Deprivation and price values also reached statistical significance. However, unlike the Distance to Station variable, the direction of association was positive, and strength was slightly stronger ($r = 0.2546$, $p < 0.05$). Unlike the previous two variables, the third candidate variable, Quality Rating, was not associated with the price values. Therefore, it will not be included at the modeling stage. The scatter plots of the association patterns of the three candidate feature variables are shown below.

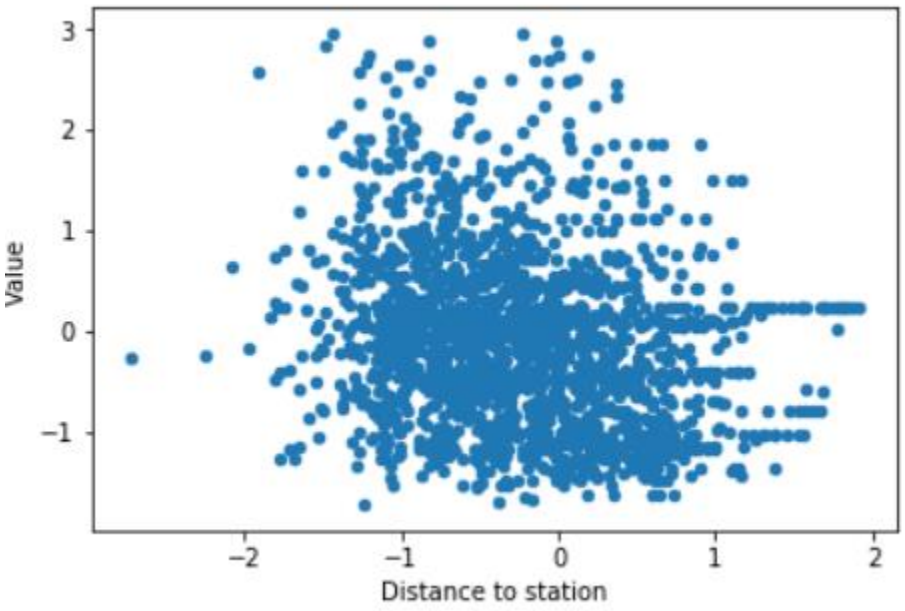

Figure 3.0 The relationship between modified price value and distance to station (log-transformed)

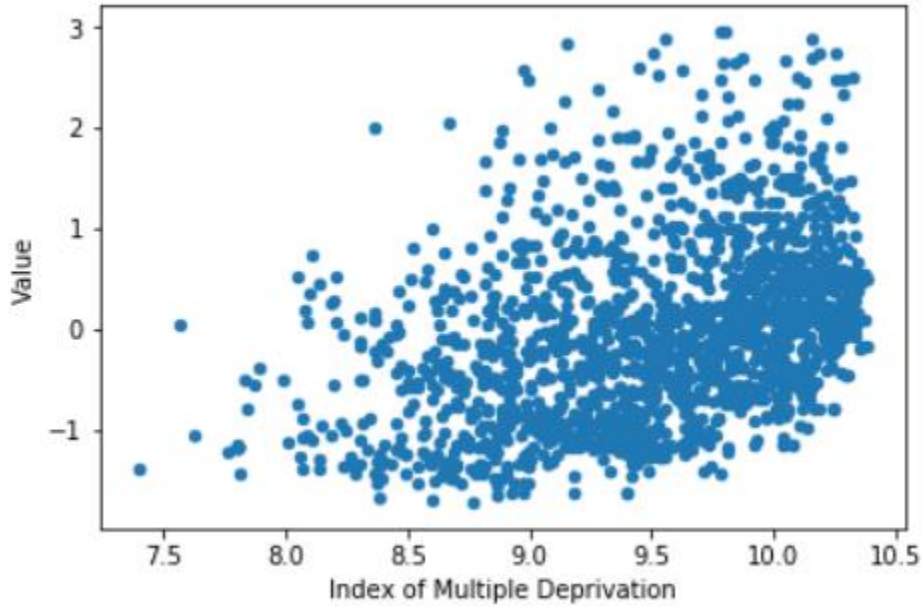

Figure 3.1 The relationship between modified price value and Index of Multiple Deprivation (log-transformed)

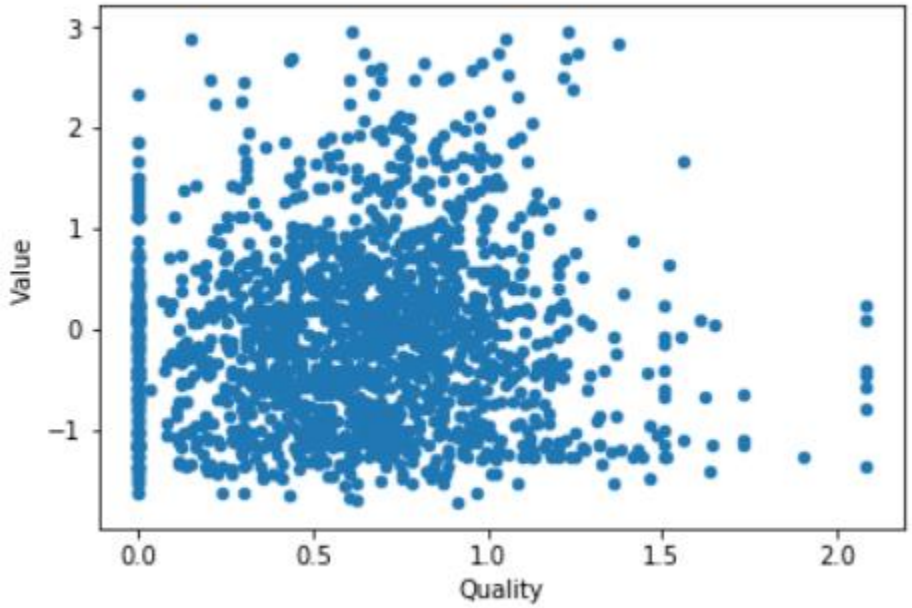

Figure 3.2 The relationship between modified price value and Quality rating (log-transformed)

Model Building

As mentioned in the procedure section, Random Forest model is first applied to the entire training set for feature selection. Without any modifications, results of 5-fold cross-validation suggest 96.84% accuracy score, which is considered relatively high. However, there are 454 venue categories variables, it is likely that not all variables are relevant in predicting housing price categories. As shown in the graph below, the mean importance score dropped significantly after the first four variables. Additionally, a few features are assigned with a zero-importance value. After dropping features with values below the cut-off threshold that was mentioned in the method section (<0.0015), the cross-validation results of the Random Forest model improved to 99.06%. With regard to the KNN model, when fitted with the selected features, results of the GridSearchCV indicate that the best accuracy score (84.05%) was yielded when K=1.The implication of such findings will be elaborated in more details in the discussion section. Regarding the third classification model, SVM, the one-against-one approach yielded 59.75% accuracy score which is higher than its one-against-the rest counterpart (47.12%). Therefore, only the former is included in the model comparison stage. In addition, the identified best C parameter for the two approaches are 100 and 0.1 respectively. With respect to the Logistic Regression model, the cross-validated accuracy score is 49.07%, which is also considered quite low. Potential explanations for such a low accuracy score are offered in the discussion section.

Model Comparison

Table 1.0

Classification Results for the Four models

	Accuracy Score	F1-Score	Precision	Recall
RF	99%	99%	99%	99%
KNN	88%	88%	88%	88%
SVM	61%	61%	61%	63%

	Accuracy Score	F1-Score	Precision	Recall
LG	49%	48%	49%	49%

As shown in the table above, the model performance rank is the same as the one derived from the model building stage. Random Forest model yielded the best overall performance and, therefore, considered as the best classification model. Surprisingly, the KNN model yielded better prediction performance on the unseen testing data than training data.

Discussion

With regard to the first aim of the study, the prediction performance of all the above models exceeded the null acuracy on both cross-validation set and the test set, with Random Forest model yielded the highest accuracy and F1 score. This is consistent with findings of the international literature that examined housing prices in Onondaga County (Yoo, Jungho & Wagner, 2012), Saint-Petersburg (Antipov & Pokryshevskaya, 2012) and Santiago (Masías et al., 2016). The results of this case study strengthened the evidence for the superior predictive performance of RF for explaining variances in housing price. However, the high accuracy score is at the expense of its interpretability, as the only way to inspect the predictive power of each feature variable is through the variable importance measure. In addition to RF model, KNN model also performed optimally. However, the model yielded the best accuracy score when K=1, which is often considered as a low bias but very high variance model, which means it might be sensitive to outliers which consequently lead to overfitting. However, in our case, the outliers are largely removed, which might potentially explain the higher accuracy and f1 score generated from the test set. It is also worth noticing that the linear SVM and logistic regression yielded significantly lower accuracy and f1 score on both training and testing sets. Indeed, the accuracy scores were only about 15% higher than the null accuracy. A possible explanation is that such

drastic differences among models are a result of model assumption violations. Both Linear SVM and logistic regression are linear classifiers, it is possible that the decision boundary of the current dataset is non-linear as the both non-linear classifiers (KNN and RF) performed optimally. Such explanation was tested by running the SVM with non-linear kernel. However, due to the relatively large size of data and the nature of SVM, the algorithm training did not finish within 3 hours and the notebook crashed as a result. This indeed has been considered as one of the major drawbacks of SVM with non-linear kernels. No further investigation attempts were made as its computation time is not justified by the reasonably good performance that is already obtained using random forest and KNN models. Therefore, the above non-linearity explanation is only probable.

Regarding the second aim of the study, the housing price of a given set of London geographical coordinates could be predicted based on its distance to the station, deprivation index, and the venue categories and geographical coordinates of its nearby properties. However, the predictors differ in relative importance. The latitudes and longitudes of nearby venues are the top two variables that are most discerning between classes, which are then followed by the deprivation index and distance to station. Venue categories data played a relatively insignificant role in predicting housing price categories. There are two possible explanations for this result. Firstly, this could be attributed to weak relationship between housing price categories and its nearby venue categories. Secondly, the venue category data is limited to its functional categories, which provide little information about its associated expenditure level, which could potentially be a better predictor of housing price categories. This information could be sourced using the 'get venue details' query, however, due to the relatively large data size, the number of API calls will

exceed the quota of the current standard Foursquare developer account. Therefore, future research with the additional focus on nearby venue categories that reflect household expenditure level is encouraged.

References

Basu, S., & Thibodeau, T. G. (1998). Analysis of spatial autocorrelation in house prices. The Journal of Real Estate Finance and Economics, 17(1), 61-85.

Islam, K. S., & Asami, Y. (2009, July). Housing market segmentation: a review. In Review of Urban & Regional Development Studies: Journal of the Applied Regional Science Conference (Vol. 21, No. 2‐3, pp. 93-109). Melbourne, Australia: Blackwell Publishing Asia.

Margulis, H. L. (1998). Predicting the growth and filtering of at-risk housing: Structure ageing, poverty and redlining. Urban Studies, 35(8), 1231-1259..