Predicting housing prices in London

Done by: Heng Jian Shun

Abstract/summary

In this report, I conduct several machine learning methods to identify the combination of model and parameters that produces the best out-of-sample predictions for housing prices in London. I first run a benchmark regression and tree model, followed by other more sophisticated models. Finally, I used an ensemble stacking method to aggregate the results from the top performing models. The final outcome showed that the ensemble tree methods significantly outperformed the base regression and tree models. Additionally, the ensemble stacking model was the top performer; it returned the lowest validation RMSE. This model was then used to select a portfolio of 200 properties that would yield the highest expected returns.

Introduction

London's housing prices has been on a steady climb over the past 2 decades. With the wealth of data available to the masses, it is wise and prudent for potential investors and buyers to employ machine learning techniques in predicting housing prices.

Dataset

The dataset that I used is a composite dataset from multiple publicly available sources. The property sales data stem from the UK Land Registry and only involves transactions made within 2019. This data was then merged with specific data on each property, drawn from Energy Performance Certificate's datasets. The dataset also contains public transportation information pertaining to the properties, as well as postcode and district name data.

To begin with, I isolated a list of variables, depicted in *table 1*, that I felt would be essential to property price determination and list them alongside a description for each variable (referenced from the data dictionary provided). I have briefly split the features into four distinct categories; namely generic property information that could help reveal hidden price mechanisms, geographical factors that reveal clustering through property buyers' tastes and preferences, convenience factors that logically motivate heightened demand from property buyers, and household affluence factors that should be positively correlated with property buyers' willingness and abilities to pay for properties. These variables will be the ones that I will use to form the basis of my machine learning models.

Column Name	Descriptions
GENERIC PROPERTY INFORMATION	
property_type	Whether the property was classified as detached, semi-detached, terraced, flats or others
whether_old_or_new	Records whether the property is a newly built building or pre-existing established one
freehold_or_leasehold	Whether the property was freehold or leasehold
number_habitable_rooms	Documents how many rooms within the property are habitable
tenure	Whether the property is owner-occupied, rented out privately or rented out for social reasons
GEOGRAPHHICAL ASPECTS	
district	The district in which the property is located
latitude / longitude	Longitudes and Latitudes of the centroids of the properties' postcodes
CONVENIENCE ENJOYED BY PROPERTY	
num_tube_lines/num_rail_lines/num_light_rail_lines	The corresponding number of tube/rail/light rail lines within the closest geographical train station.
distance_to_station	Geographical proximity to train station
water_company	Documents the water company that is responsible for the postcode
AFFLUENCE OF HOUSEHOLD	
average_income	The average household income of the postcode
co2_emissions_current	Records the current CO2 emissions by the property
windows_energy_eff	Records the energy efficiency rating of the property on a 5-point Likert scale

Table 1: Description of important variables that I used for analysis

Exploratory Data Analysis

Before beginning my analysis, I wanted to first get a good understanding about the London Housing price dataset. I start off by getting a feel of property price distribution within London; *figure 1* below illustrates this. The density plot has a heavily skewed right-tail, suggesting that certain housing properties with London are extremely highly-priced anomalies within the city. Despite this, I opt to keep these properties within my dataset since no fixed price threshold cut-off point was provided for this exercise.

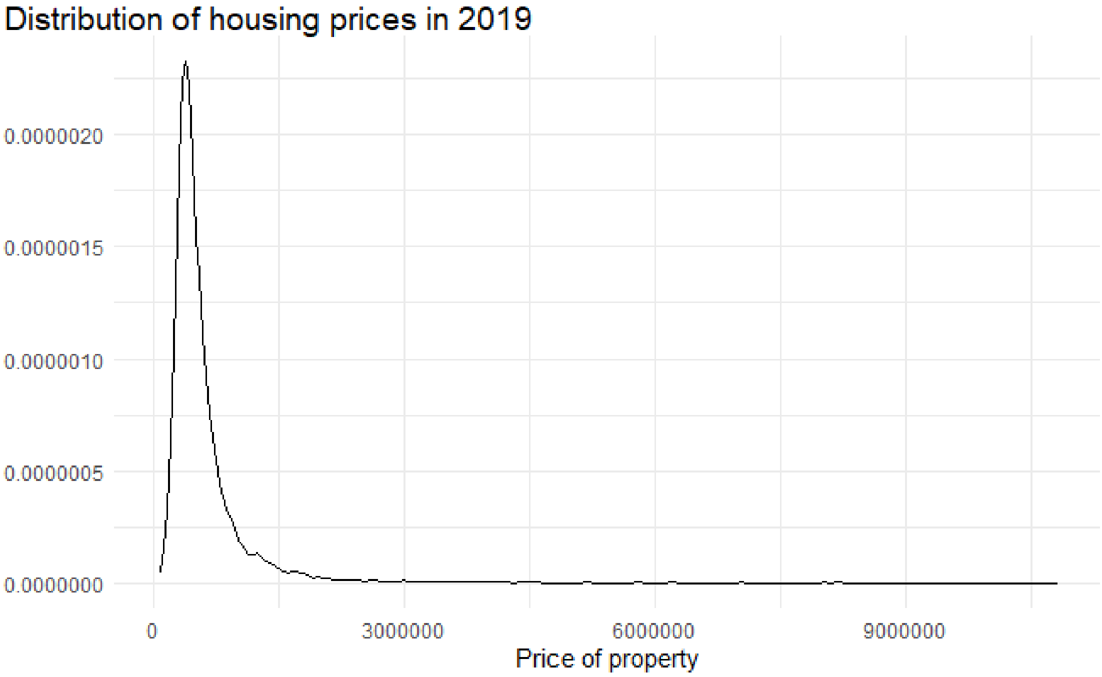

Figure 1: Distribution of London housing prices

Next, I explore how geographical locations within London affect housing prices. *Figure 2* showcases the top 10 most expensive and least expensive districts in terms of average housing prices. From *figure 2*, we see that the top 10 most expensive districts are clustered around Central London while the least expensive districts are at the outskirts of London; this suggests that geographical factors heavily influence housing prices within London.

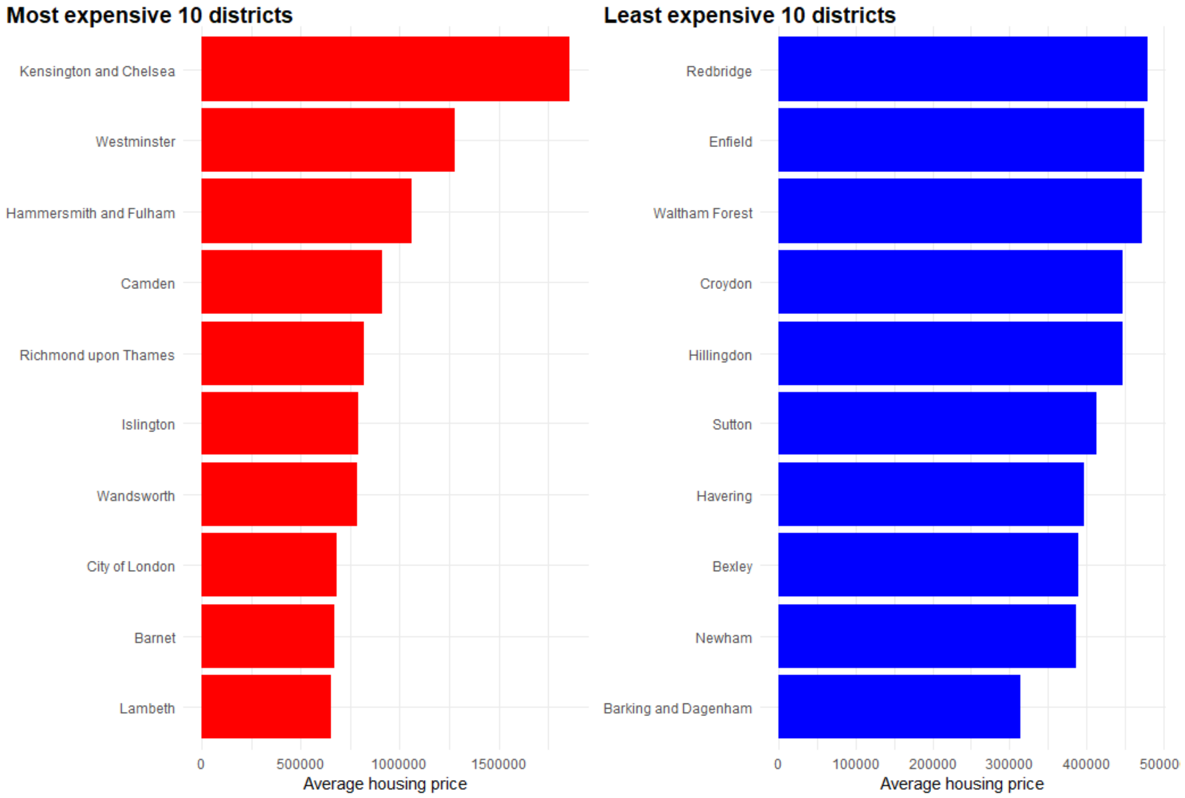

Figure 2: 10 most expensive and 10 least expensive districts in terms of average housing price

In addition, I performed a mandatory correlation check for the variables that I had selected in the section above, with the results observed in *figure 3* below.

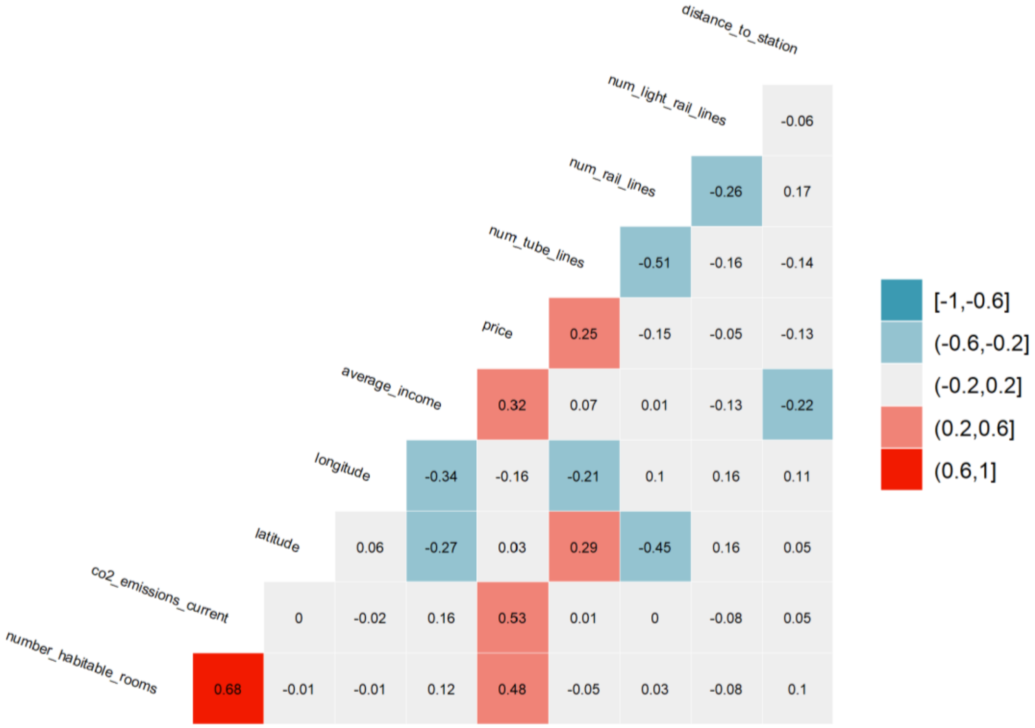

Figure 3: Correlation matrix generated on selected variables

From this correlation matrix, I observe that there neither exists perfect or imperfect multicollinearity between the selected numerical variables. There are also noteworthy correlations between the price of properties and metrics that measure household affluence, as well as distinct correlations between the number of tube lines and property prices, suggesting that the convenience gained from easy access to tube travel may be a strong determinant of property prices.

Methodology

In this report, I seek to tease out the most optimal machine learning or parametric model that can be used to predict housing prices within London in 2019. The models that I have decided upon for this analysis are documented in *table 2*.

Number	Model Name
1	Linear Regression (base)
2	Tuned Tree Model
3	LASSO Model
4	K-Nearest-Neighbour Model
5	Random Forest Model

6	Gradient Boosting Model (R “gbm” package)
7	Boosting xgboost Model

Table 2: Models to be used within analysis

To compare the performances between different models, I will be using squared loss metrics in the form of Root Mean Squared Error (RMSE); models that produce lower validation/test RMSE will be deemed as superior and as better choices for housing price prediction. I will also be performing a common ensemble average method known as stacking to see if even better results can be achieved via aggregation.

Finally, the goal of the analysis is to isolate the best method/model that will allow me to select a portfolio of 200 houses that I intend to invest in for the highest returns. I will be running my selected model on the predictor variables of interest – my housing price predictions will then be used in calculating returns via the given asking price on each property. Properties that yield the largest value of

$$return = \frac{predicted\ price - asking\ price}{asking\ price}$$

will be selected within my portfolio.

Linear Regression (base model)

The linear regression model is the most basic model and also forms the benchmark against which I will subsequently compare all further models against. The linear regression model outcome reveals that the top 20 variables of importance, seen in *figure 4* below, contain specific districts, co2_emissions_current, number_habitable_rooms, average_income, windows_energy_efficiency, property_type and num_tube_lines. This can be seen in *figure 2* below. Within these variables, the 5 variables with variable importance scores above 50 paint a narrative that household affluence in terms of current co2 emissions and average income (both of which are proxies for the standard of living of residents within the property) and specific districts within London are associated with distinct clusters of housing prices. These variables are intuitively important as determinants of housing prices.

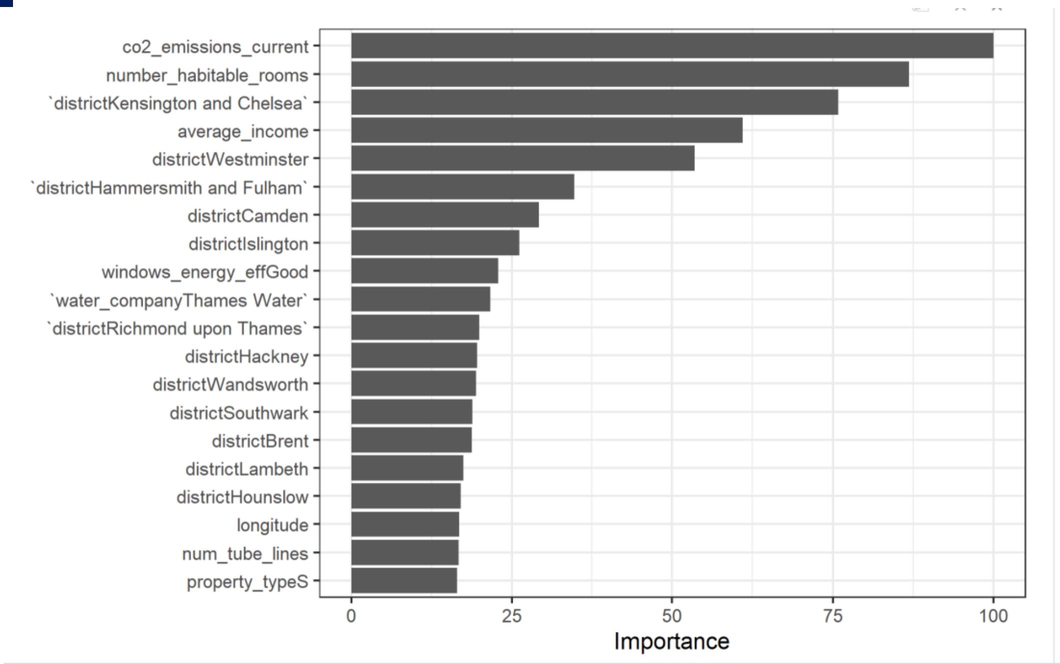

Figure 4: Variable Importance Plot for the top 20 most important variables

Table 3 below documents the RMSE for the benchmark Linear Regression Model.

Linear Regression Model	
RMSE	333122.5
R^2	0.619

Table 3: Test-sample RMSE and R^2 tabulation for the linear regression model (benchmark)

Tree Model

I then carried out tree-based modelling. As can be seen from *table 4* below, the benchmark tree model (with no parameters tuned) performed worse than the base linear regression model. This seems to be the case because the chosen model was extremely simplistic with only 10 leaf nodes.

Tree model (benchmark)	
RMSE	379837.7
R^2	0.508
Leaf nodes	10

Table 4: Test-sample RMSE and R^2 tabulation for the base tree model (benchmark)

To build a more optimal tree model, I went about tuning the complexity parameter value. The final optimal complexity parameter was 0 (completely un-pruned) and this model yielded a different set of most important variables, seen in *figure 5* below. In this model, the variable importance was concentrated across a smaller number of predictor variables, namely geographical latitude/longitude, average_income, co2_emissions_current and distance_to_station.

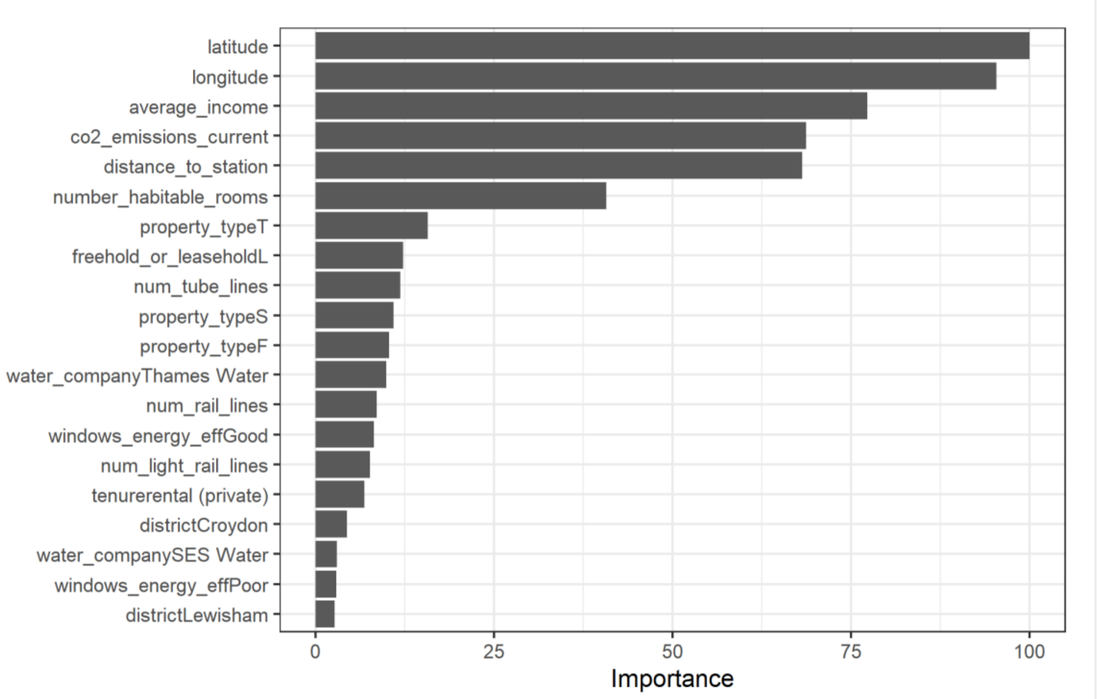

Figure 5: Variable Importance Plot for the top 20 most important variables (Tree model)

Table 5 below documents the RMSE for the optimal Tree Model.

Tree model (benchmark)	
RMSE	312083.5
R^2	0.676
Complexity parameter	0

Table 5: Test-sample RMSE and R^2 tabulation for the optimal tree model

LASSO Model

At this juncture, I thought it would be interesting to shrink the model complexity to obtain a more parsimonious model; LASSO achieves this by shrinking coefficients to zero.

Table 6 below documents the RMSE for the LASSO Model. Surprisingly, the LASSO model has an almost equivalent performance to the base linear regression model. This may also be because this LASSO model failed to shrink any coefficient to exactly zero; essentially this means that the model functions almost equivalently to an OLS regression.

Tree model (benchmark)	
RMSE	333198.1
R^2	0.619
Number of non-zero coefficients	57
Number of zero coefficients	0

Table 6: Test-sample RMSE and R^2 tabulation for the LASSO Model

K-Nearest-Neighbour Model

I then turned to the K-Nearest-Neighbour (KNN) Model for cross referencing. To do so, I first tuned the number of neighbours to the optimal number. *Figure 6* below reveals that the optimal number of neighbours is 11.

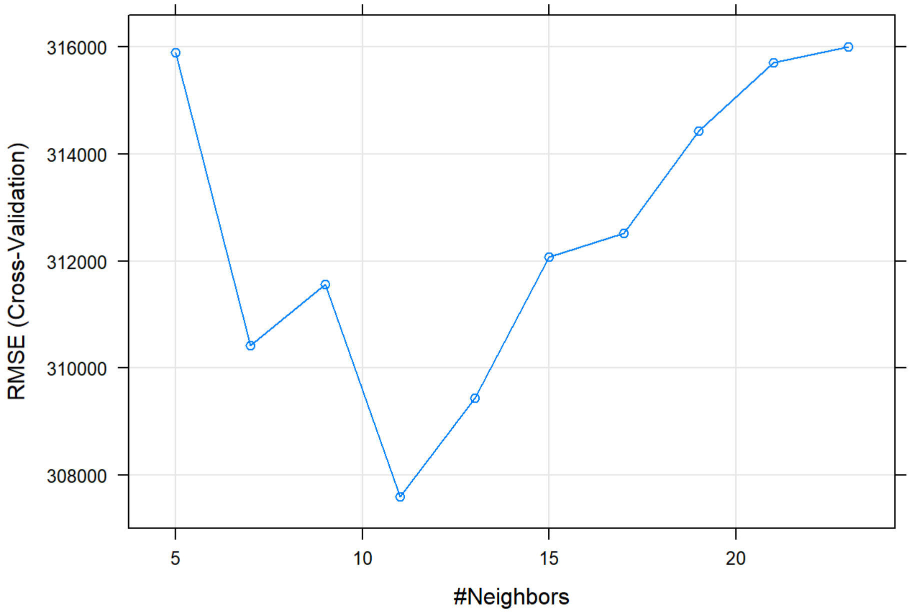

Figure 6: Tuning for number of neighbours in the KNN Model

Table 7 below documents the RMSE for the KNN Model. The KNN Model performs better than regression and basic tree models. However, it runs the risk of overfitting if tuning is sub-optimal.

K-Nearest-Neighbour Model	
RMSE	301321.8
R^2	0.713
k	11

Table 7: Test-sample RMSE and R^2 tabulation for the KNN model

Ensemble tree methods

Random Forest Model

Since the basic tree models did not effectively reduce the RMSE, I decided to run ensemble tree methods in hopes of achieving lower RMSE through iterative/bagged processes. The first ensemble tree method I used was the Random Forest model.

Table 8 below documents the RMSE for the Random Forest Model alongside the optimal mtry and minimum node size parameters that I obtained during tuning. This model returns the lowest RMSE by far, confirming my initial believes that the ensemble methods would perform significantly better.

Random Forest	
RMSE	258544.3
R^2	0.790
mtry	7
Splitrule	variance
Minimum node size	5

Table 8: Test-sample RMSE and R^2 tabulation for the Random Forest model

Gradient Boosting Model

To cross-check the outcomes from the Random Forest Model, I ran a gradient boosting algorithm with the gbm package on the chosen features and tuned the interaction depth, number of trees, shrinkage and minimum in-node observations accordingly. *Table 9* below documents the RMSE for the Gradient Boosting (gbm) Model, and returns an outcome that is comparable to that of the Random Forest.

Gradient Boosting Model	
RMSE	253037.0
R^2	0.780
Interaction depth	7
Number of trees	250
Shrinkage	0.075
Minimum observations in node	10

Table 9: Test-sample RMSE and R^2 tabulation for the Gradient Boosting (gbm) model

Xboost Model

Out of curiosity, I decided to attempt another boosting model that operates on a different algorithm. The model selected here is the xgboost model, and has the advantage of running at a much faster speed due to parallel computation. *Table 10* below documents the RMSE for the xgboost model; it does not perform as well as both the Random Forest and Gradient Boosting (gbm) models.

Xgboost Model	
RMSE	288282.4
R^2	0.728
Max depth	7
Number of iterated rounds	99

Table 10: Test-sample RMSE and R^2 tabulation for the xgboost model

Overall model selection

Table 11 below showcases the RMSE comparison between all aforementioned models. As stated previously, both the ensemble tree methods – random forest and gradient boosting (gbm) – perform significantly better than regression models.

Model Name	**RMSE**
Linear Regression	333122.5
Tree (tuned)	312083.5
LASSO	333198.1
KNN	301321.8
Random Forest	258544.3
Gradient Boosting	253037.0

Xgboost	288282.4

Table 11: RMSE result tabulation across all models

Ensemble Method - Stacking

Finally, I decided to perform yet another ensemble method – this time, it is one that is not a pure ensemble tree method. I ran a stacking model on my 4 top-performing models with the lowest RMSE. These models were the Tuned Tree, KNN, Random Forest, and Gradient Boosting (gbm) models. One caveat of the stacking model was that it was unable to accommodate the xgboost package which returned the 3rd best results. However, this is a minor issue. The stacking model utilises a logistic regression to aggregate and average results from each of the individual models above. *Table 12* below showcases the outcomes of the stacking model; indeed, it yielded the lowest RMSE amongst all of the individual models.

RMSE	251102.4
R^2	0.787
Models selected for stacking	Tuned Tree, KNN, Random Forest, Gradient Boosting

Table 12: Test-sample RMSE and R^2 tabulation for the ensemble stacking model

With this in mind, I then proceed to use this stacking model on the given dataset for out-of-sample predictions to create my portfolio of 200 housing properties The objective will be to select the properties that yield the highest returns according to the formula:

$$return = \frac{predicted\ price - asking\ price}{asking\ price}$$

Crossrail Elizabeth line analysis

With the addition of the Elizabeth Line to London's public transportation network, one would expect properties closer in proximity to stations that house the crossrail line to retail at higher prices; this is under the assumption that the Elizabeth Line helps bridge areas of London which may have been previously less accessible. The commute time that residents in these regions are able to save can be loosely translated into monetary gains through heightened productivity and convenience – thus, it is important to examine the impact of the crossrail line on housing prices. The simplest way to go about doing this is to include an additional feature, the distance of a property from the nearest crossrail station, into the subset of features that are to be selected for housing price prediction.

Additionally, I could also add a binary independent variable (I will name this variable "is_crossrail" for simplicity) denoting whether or not a crossrail station lies within an arbitrarily selected threshold walking distance, for example 1 kilometre, from a property. With this "is_crossrail" analysis, it is imperative to jointly consider whether other public transportation options such as tube and rail lines are also available within this threshold distance (I will name this binary variable "is_publictransport"). Properties that have both the crossrail line and at least one other public transportation line option can be defined as "interchanges". I believe that the strength of the correlation between property price and proximity to these interchanges could go in either direction when compared to the relationship between property price and stations which only house the crossrail line. Two opposing effects are at play here. On the one hand, these interchanges could make property prices even more expensive since they allow for convenient transfers between public transportation options. On the other hand, overlaps between the crossrail line route and existing train/rail routes will diminish the attractiveness of the crossrail line between those regions; the extent of property price increases in those regions as a result of the inauguration of the crossrail line will thus be weaker. To determine and isolate the true effect of the crossrail line on housing prices, I would create a binary interaction variable between the dummy "is_crossrail" variable and "is_publictransport" variable and run my analysis on all relevant features.

Conclusion

While I have tried to tune each model above with a unique combination of hyperparameters, in actual practice, parameter and hyperparameter tuning can be extremely complicated, with tens of thousands of iterations run on dozens of models, each of which containing different subsets of features. Hence, this analysis that I conducted is comparatively simplistic; the hyperparameter tuning processes that I conducted on each of my selected models were not as elaborate and extensive. In this report, I created a stacking model made out of the individual Tree, KNN, Random Forest and Gradient Boosting (gbm) models and used it to select a list of 200 properties that would yield the highest potential returns. The analysis also revealed that certain predictor variables were consistently more important than others and these were unique to each model; the top 5 most important variables can be seen in *table 13* below.

Linear Regression	Tuned Tree	KNN	LASSO	Random Forest
Co2 emissions current	Latitude	Co2 emissions current	Kensington and Chelsea district	Number habitable rooms
Number habitable rooms	Longitude	Number habitable rooms	Co2 emissions current	Co2 emissions current
Kensington and Chelsea district	Average income	Average income	Number habitable rooms	Kensington and Chelsea district
Average income	Co2 emissions current	Longitude	Westminster district	Freehold or leasehold
Westminster district	Distance to station	Num tube lines	Average income	Num tube lines

Table 13: Top 5 most important predictor variables in variable importance plots

Loosely speaking, the most important variables that predict a property's price seem to be a combination of geographical (latitude, longitude, Westminster district, and Kensington and Chelsea district), convenience (distance to station, and num tube lines), property (number habitable rooms, freehold or leasehold), and household affluence (average income, and co2 emissions current) factors.

A more comprehensive list of predictors for future analysis could include information about the neighbourhood around the property. One could likely theorise that property buyers are more family-centric than property renters; these are individuals who probably have higher socioeconomic status since they most likely have been in the workforce for a longer period of time and are looking to settle down. In this case, factors such as the number of available schools in the area and the proximity to family-friendly shopping malls and amenities may influence the prices of certain types of properties more than others.

APPENDIX

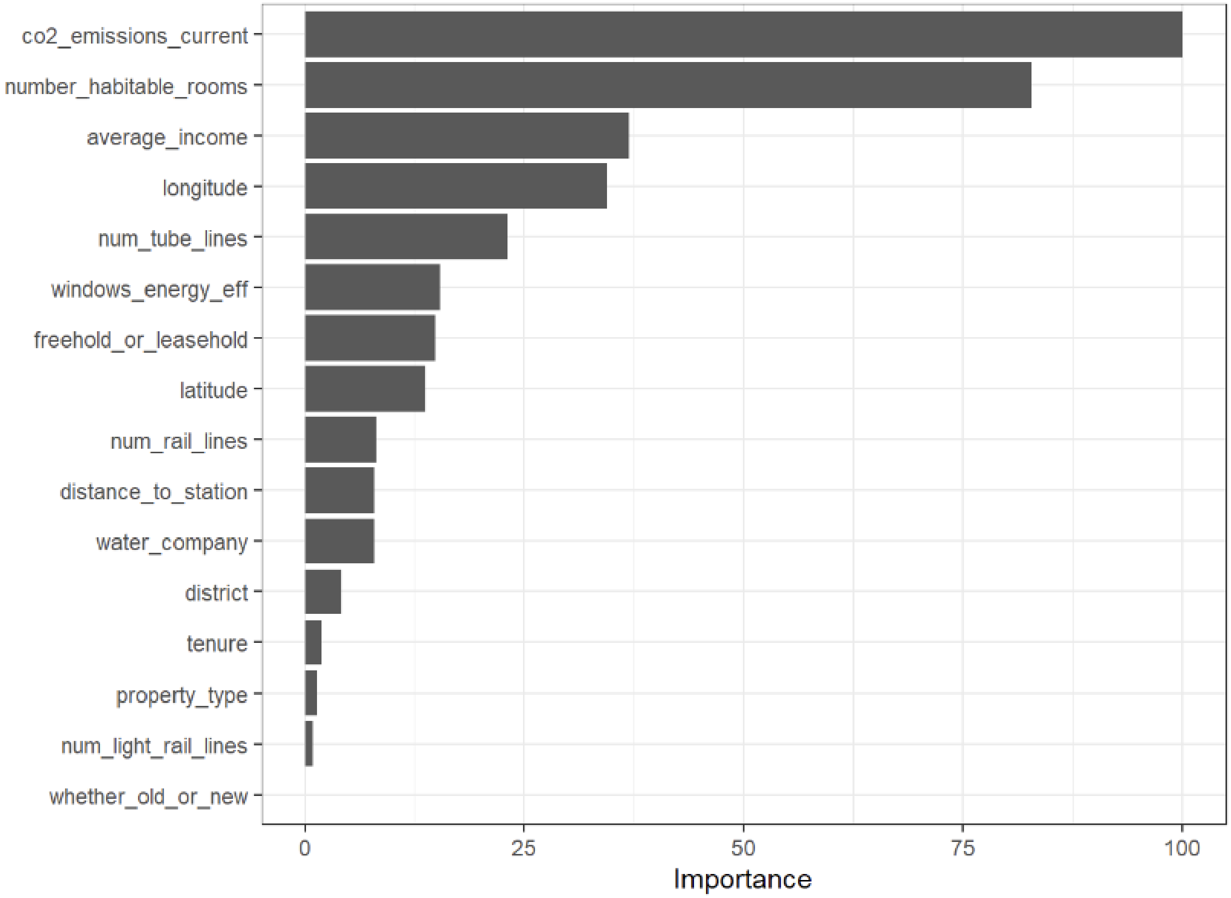

Figure A1: Variable Importance Plot for the top 20 most important variables (KNN Model)

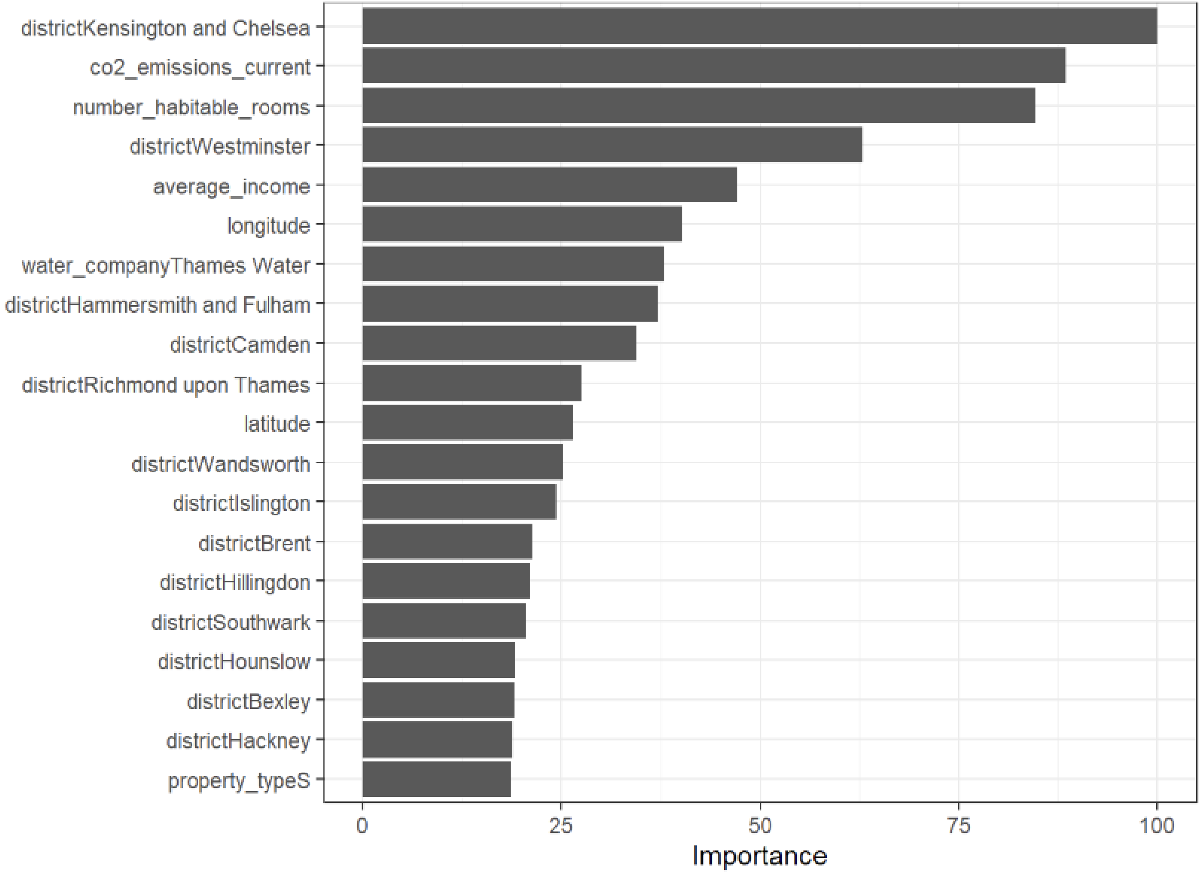

Figure A2: Variable Importance Plot for the top 20 most important variables (LASSO Model)

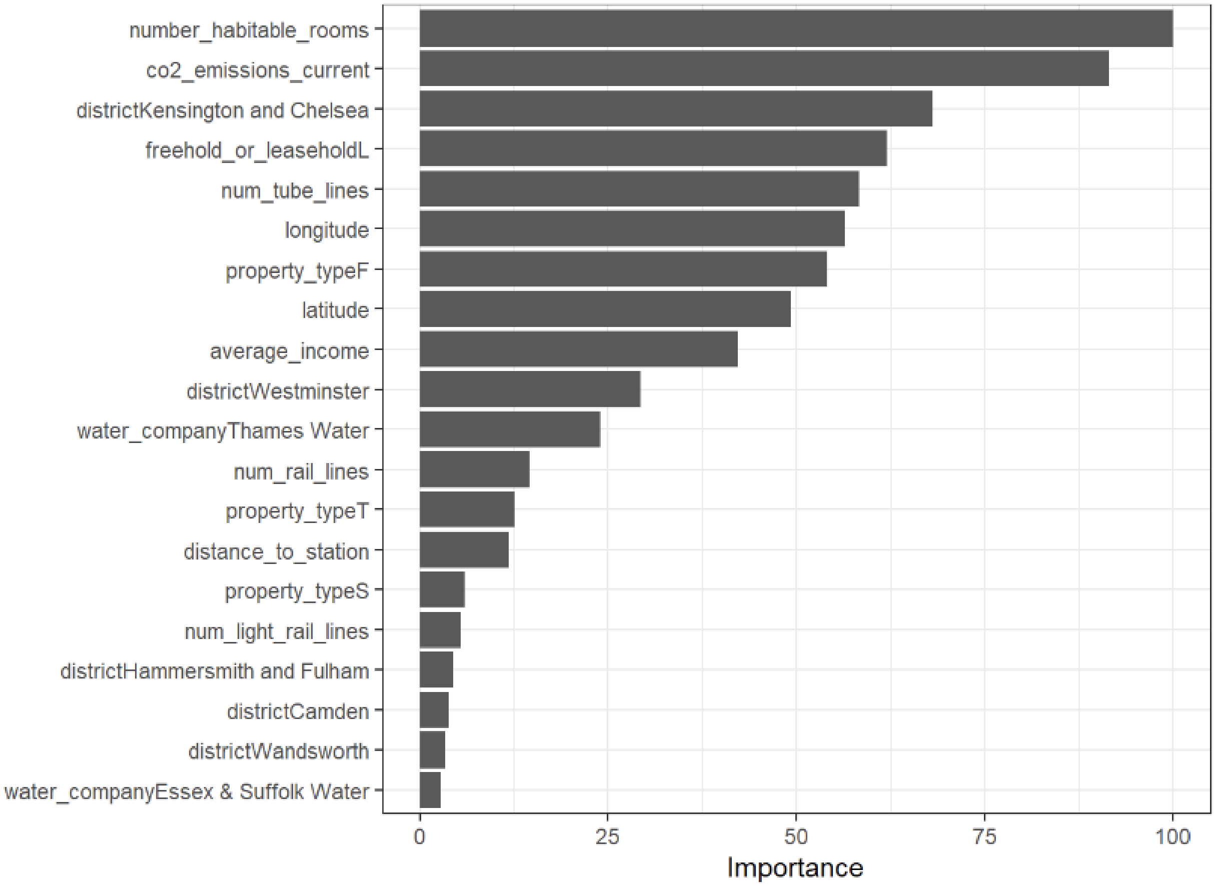

Figure A3: Variable Importance Plot for the top 20 most important variables (Random Forest Model)