London Real Estate Asset Management

Advait Jayant

# Abstract

This report analyses the London Housing Prices dataset from 2019 to build an investment portfolio of 200 houses such that the profit margins are maximised. In order to do this, I first analyse the dataset, explore the distribution of the prices, and visualise the relationships of the prices with other parameters. Machine Learning algorithms are statistical methods that allow computers to identify patterns and relationships in data. I build several machine learning models to predict the relationship between the dependent variables and the house price. Through stacking together my four best data learning models, I constructed my final model which performs with the highest accuracy to determine the ideal property choices. An investment of ~£100 million on my housing portfolio can yield predicted returns of ~£60 million. Upon creating the best model, I structure a methodical approach to analysing the impact of the introduction of the Elizabeth line to the London Tube System.

# Introduction

“Data beats emotions.” - Sean Rad, Founder and CEO of Tinder.

Over the past few decades, the average house prices in London have increased at a pace far greater than the increase in average incomes making London an extraordinarily lucrative opportunity for potential investors. As the Founder and CEO of Tinder, Sean Rad, stated *“Data beats emotions”.* An investor that enters the market today has access to concrete data sources which can be used to analyse and gauge the potential impact of each of their investments - even in the sphere of real estate management. This document is constructed to empower an investor with the tools to use the precious resource, data, to maximise returns from investments.

The dataset that I use to analyse the price fluctuations in the London housing market is a subset of the government Prices Paid dataset that contains information about the house sales made in 2019. The dataset contains information about the area of the property, the date of the sale, leasing information, postcode, property layout, energy rating and consumption, tenure of property, population statistics of the area, tube rail proximity, and zone. These are the parameters whose influence I intend to analyse and determine which factors impact the prices to the maximum extent.

The algorithms that I have used in pursuit of the best returns are *Linear Regression*, *Decision Trees*, *Random Forests*, *GBM*, and *Lasso Regression*. Based on the results of these algorithms, I construct a proposal with the top 200 properties to invest in.

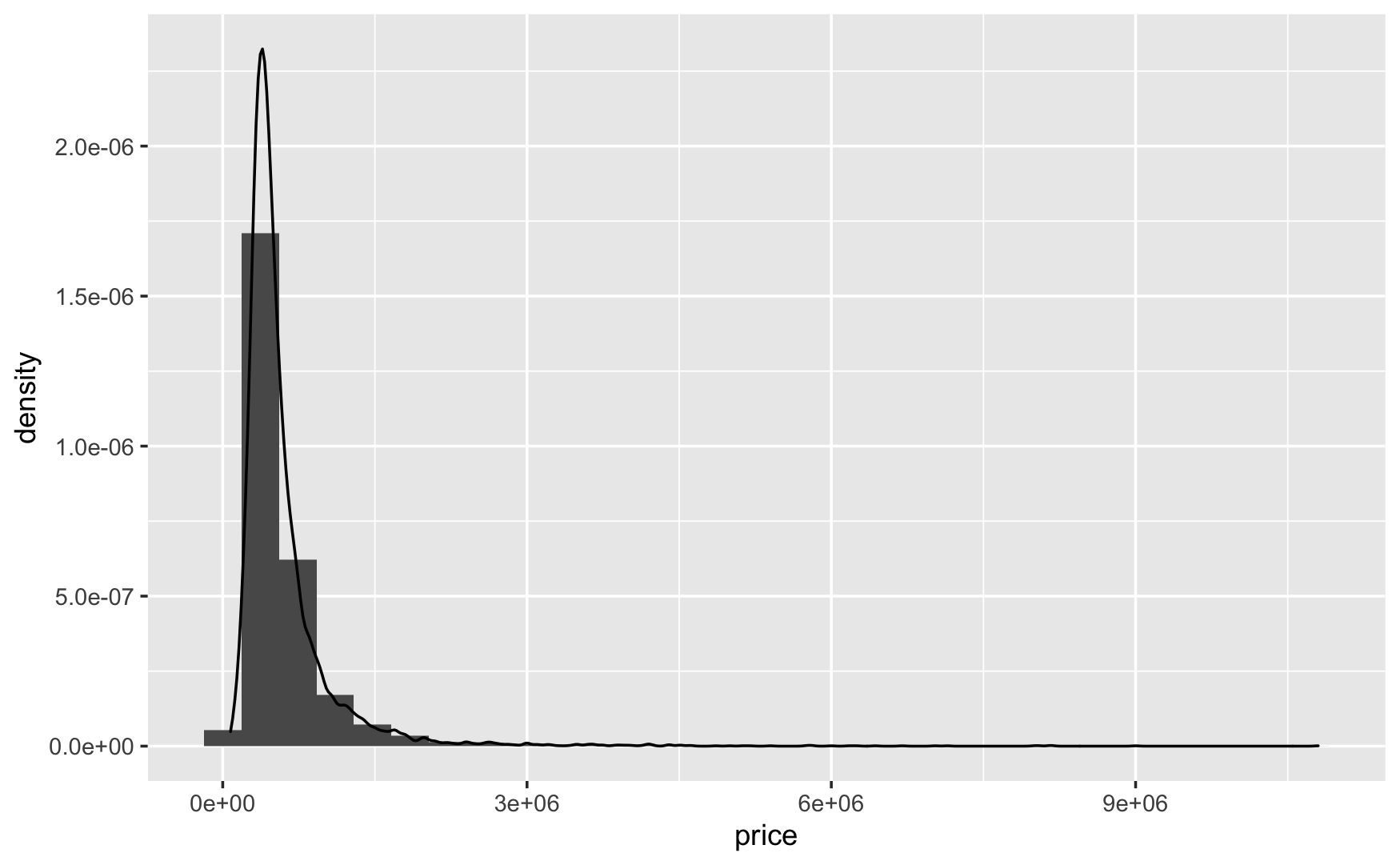
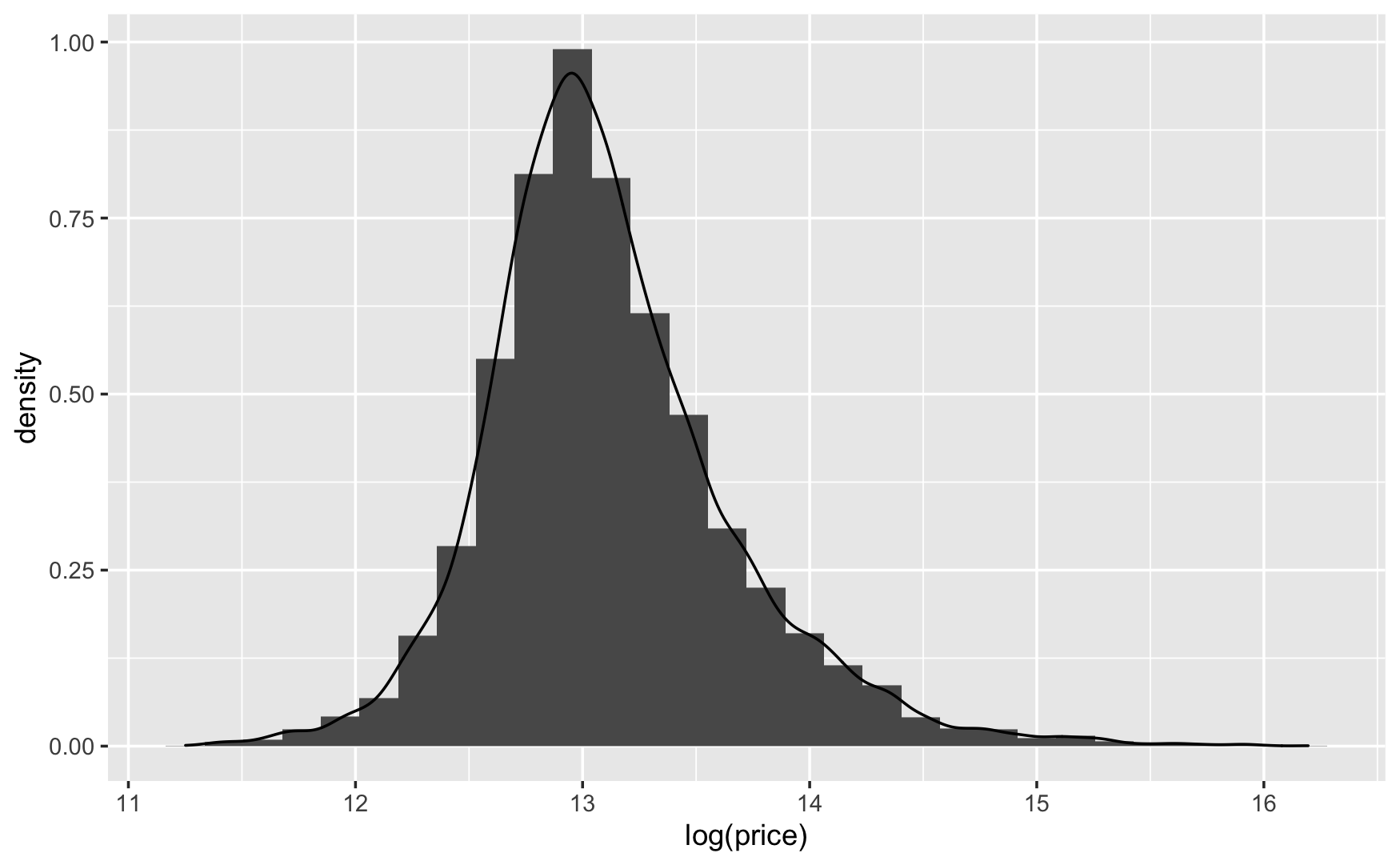
After deciding the most suitable investments, I look at the impact of a potential new tube line, the Elizabeth line, which traverses London from West to East, on the prices of properties in the areas closest to the line.

# 2. Exploratory Data Analysis

“Data scientists, according to interviews and expert estimates, spend 50 percent to 80 percent of their time mired in the mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.” - Steve Lohr, The New York Times.

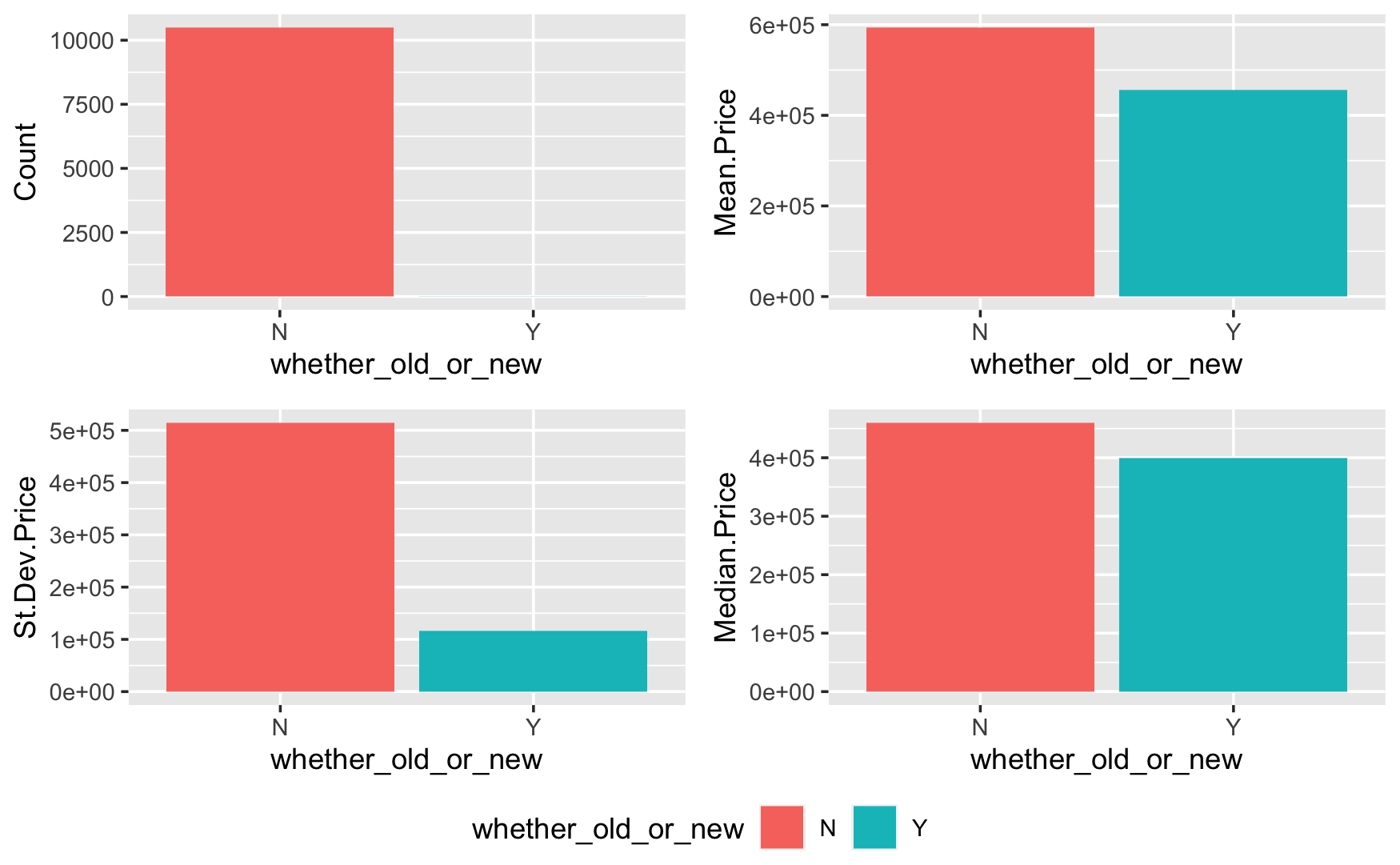
The most important part of conducting any analysis is ensuring that the dataset used to structure hypotheses and arrive at conclusions is the most optimal dataset for the problem statement. I begin the exploratory data analysis by performing several data cleaning processes such as converting character variables into factors and ensuring that the date variable is in the date datatype.

Upon confirming that the dataset is suitable, I set a seed to facilitate easier duplication of results, followed by splitting the dataset into a training set and a testing set. I use the training set to develop a model that can evaluate predictions correctly on unknown sample points, i.e, the testing set.

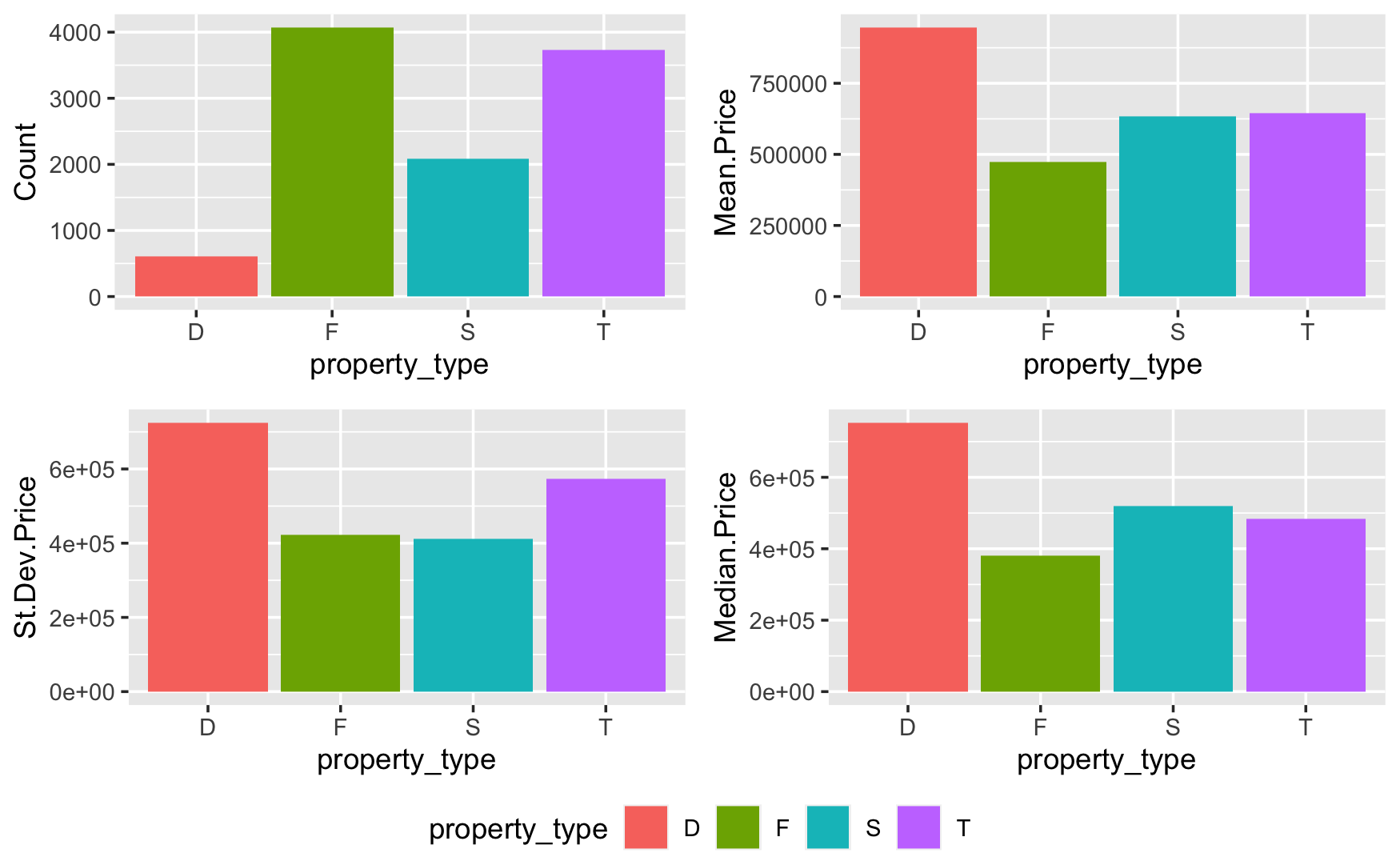
Before building any models, it is critical to analyse the distribution of the price of each house and its relationship with the several factors that possess the potential to impact its fluctuations. I find that price follows a right-skewed fat-tailed distribution due to the presence of a large number of exorbitantly expensive houses in London. The price distribution can clearly be observed to follow the 80:20 law as 80% of the investments stem from houses at the lower end and only 20% of the investments stem from houses priced greater than £600,000.

*Fig 2.1 Density Distribution of Prices (Left)*

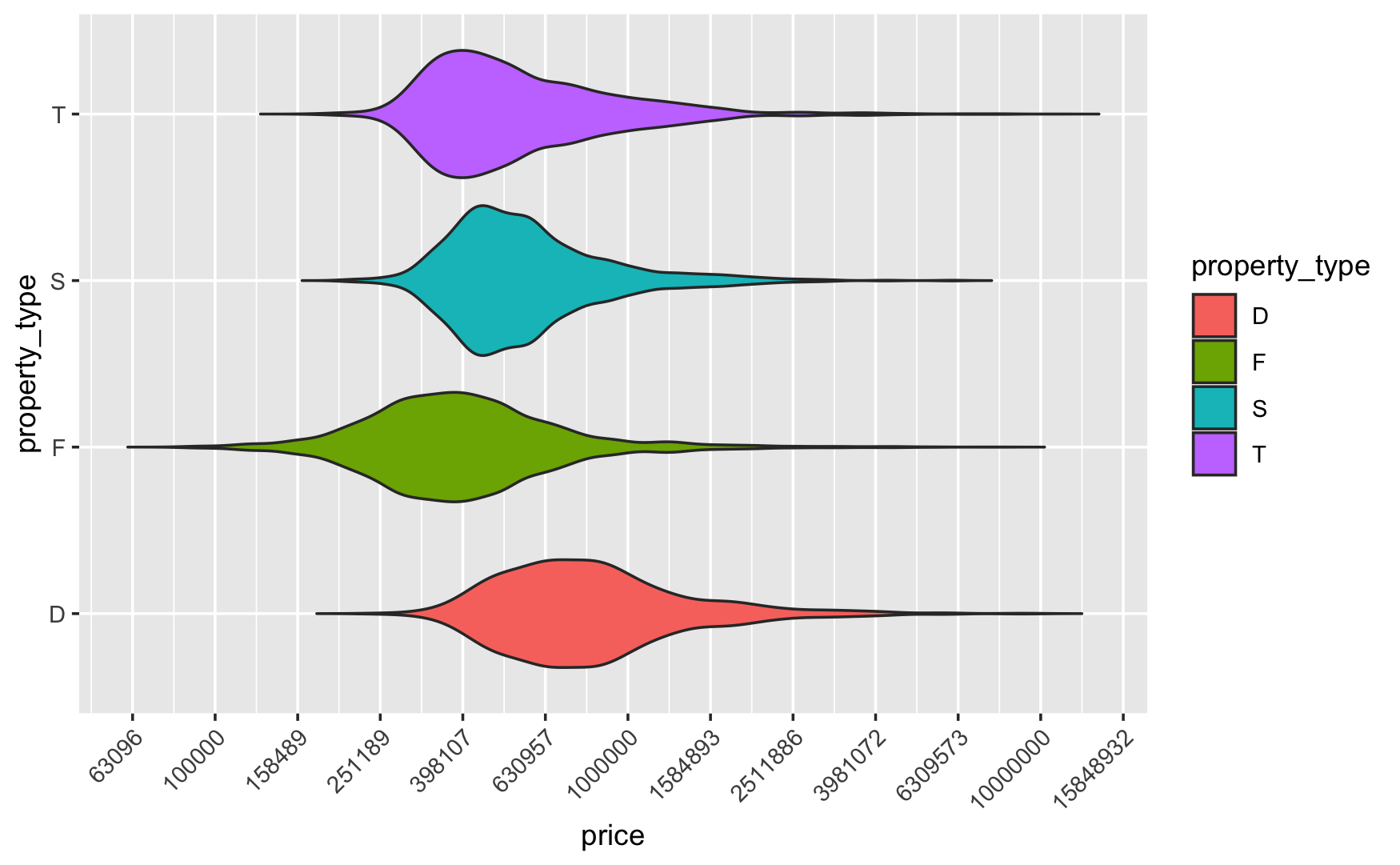
*Fig 2.2 Density Distribution of Logarithm of Prices (Right)*

Upon plotting, we find the logarithm of price to be a normally distributed variable. Predicting the logarithm of the price using linear regressions could prove to be potentially useful but different algorithms react differently to logarithmic variables and stacking these models would prove to be an arduous task. Therefore, I will be predicting the prices and not the logarithms of the prices.

*Fig 2.3 Grouping on whether\_old\_or\_new to find House Counts, Mean Prices, Standard Deviation in Prices, and Median Prices*

The *whether\_old\_or\_new* variable in the dataset describes whether a house is old or new. I construct visualisations after grouping properties based on this variable to determine the relationship of this variable and the price. Through plotting the price distributions for the two groups (*fig A.1)* , we observe that there exists a median price difference between new and old houses. Comparing the houses in these categories, we can structure the hypothesis that there exists a significant difference between newly built homes and older built homes. However, since there are only 7 new houses in this dataset - we don’t have enough data to prove our hypothesis.

*Fig 2.4 Counts, Mean Prices, Standard Deviation in Prices, and Median Prices per Property Type*

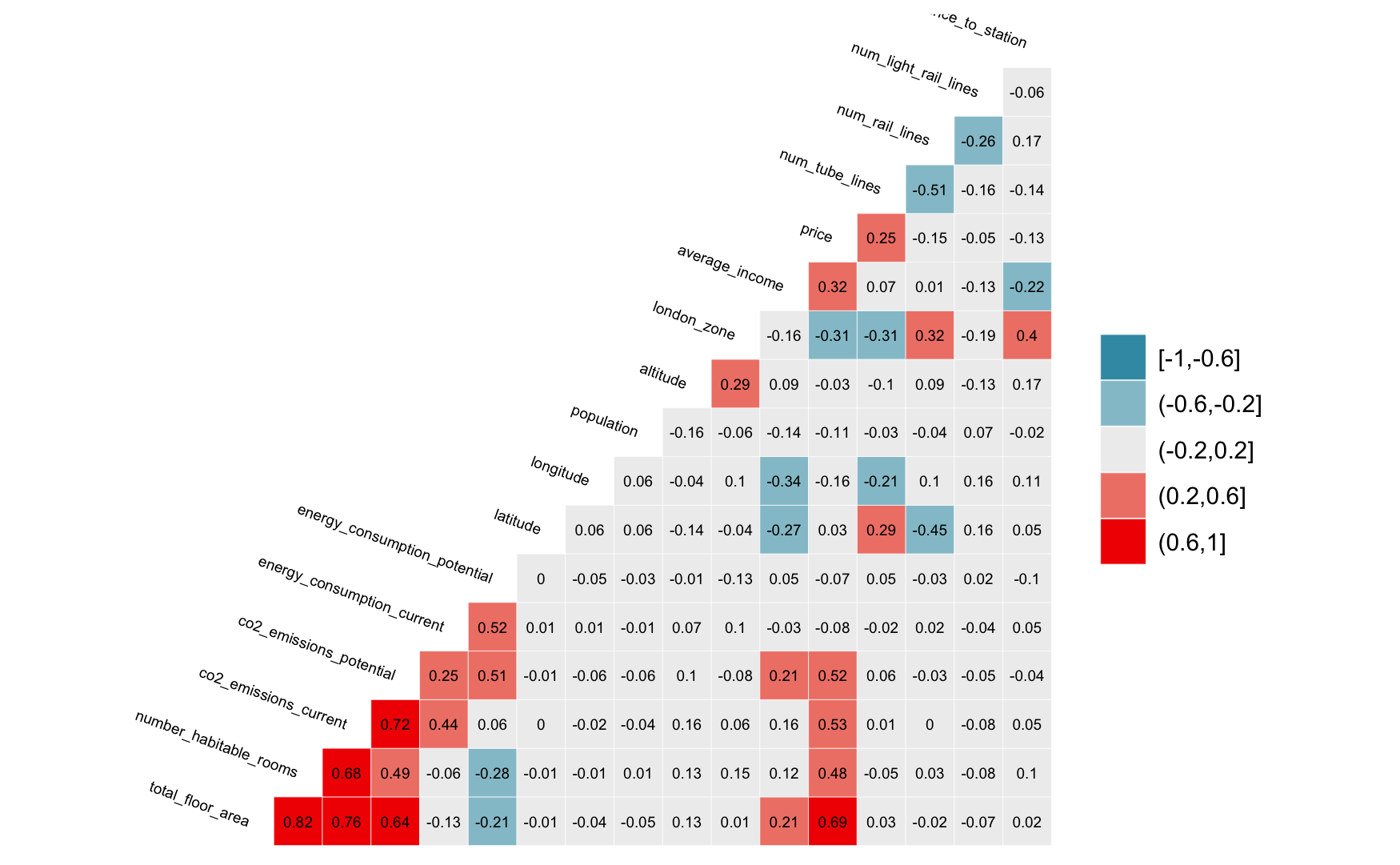
The *property\_type variable* represents the type of the property: Terraced Houses (T), Detached Houses (D), Semi-Detached Houses (S), or Flats/Maisonettes (F). Through the counts of the houses, we clearly find that type the most purchased houses in London were Flats, followed by Terrace Houses. Whereas the number of sales was considerably lower for Semi-Detached and Detached properties. From the standard deviations, we can see that Detached and Terraced houses vary considerably in price, followed by Flats, whereas Semi-Detached house prices are comparatively less spread out.

*Fig 2.4 Price Distributions per Property Type*

From the plot envisioned, we can see that the distributions for Terraced and Semi-Detached Houses are quite similar. Whereas, Flats are priced lower than the previous two and detached houses significantly higher than the others.

I conducted a similar analysis (*fig A.2* and *fig A.3*) on the *freehold\_or\_leasehold* variable to find that there were 33% more freehold properties sold in 2019 than leasehold properties - which is surprising because freehold prices are on average £200,000 more expensive. This difference in sales may be due to the fact that most freehold houses are in Central London - and there exists a growing demand for purchase of houses in Central London particularly.

*Fig 2.5 Correlation Matrix*

From *Fig 2.5,* the price has a high positive correlation with the total floor area, number of inhabitable rooms, current CO2 emissions, potential CO2 emissions, and average income. These are the variables that I need to be careful of while including in my models due to the problem of Multicollinearity. Multicollinearity happens when one predictor variable in a multiple regression model can be linearly predicted from the others with a high degree of accuracy - this can lead to skewed results. Luckily, decision trees and boosted trees algorithms are immune to multicollinearity by nature.

# 3. Machine Learning Methods

i. Linear Regression

I build four linear regression models to fit the best prediction line between the explanatory variable and dependent variables. I analysed the significance of the variables using the p-value (<0.05) to find the model with the lowest value of *Root Mean Square Error (RMSE)* and highest *R-squared*. The greater the R-squared of a model - the better it performs. The variables that I use in my best linear regression model are *distance\_to\_station, num\_tube\_lines, property\_type, altitude, latitude, longitude, water\_company, postcode\_short,* and an interaction term between *total\_floor\_area* and *london\_zone* as a factor. I obtain an adjusted R-Squared of 83.53% on the training set and 84.27% on the testing set.

ii. Decision Trees

Next, I build a decision tree model for my training dataset in order to compare its performance to that of the linear regression model. I use an initial set of parameters to build a model with a tune length of 10. The adjusted R-square starts off at 46.16%. I then use pruning to tune the hyperparameters to make the most efficient model by selecting appropriate values of the complexity parameter. The complexity parameter determines the size of the decision tree and aids in choosing the optimal value of the size of the tree. The expand grid function now determines the range of the complexity parameter and find the best R-squared. The decision tree yields an adjusted R-Square of 82.57% on the testing set. The linear regression model that I built performs better because there is a large number of features in the dataset and low noise. Decision trees are also better suited at predicting categorical independent variables whereas their performance is compromised while predicting discrete independent variables such as price.

iii. Gradient Boosting

Next, I build a model using the Gradient Boosting Machine by adding new variables. In order to ensure the best performance of this model, I added features which were significantly increasing R-Squared values in the linear regression and decision tree models. There exist three key elements to gradient boosting: optimising a loss function, building weak learners to make predictions, and finally building an additive model that combines all the weak learners to increase prediction power. In order to optimise the performance of my GBM I use pruning again to tune the hyper parameters to make the most efficient model by selecting the ideal values of the complexity parameter. I tuned the *n.trees, interaction.dept,* and *shrinkage* variables specifically after tuning the model with the best fit parameters. Through Gradient Boosting, I create a Machine Learning model with an adjusted R-Squared of 82.57% on the training dataset and an adjusted R-Squared of 86.57% on the testing dataset.

iv. LASSO Regression

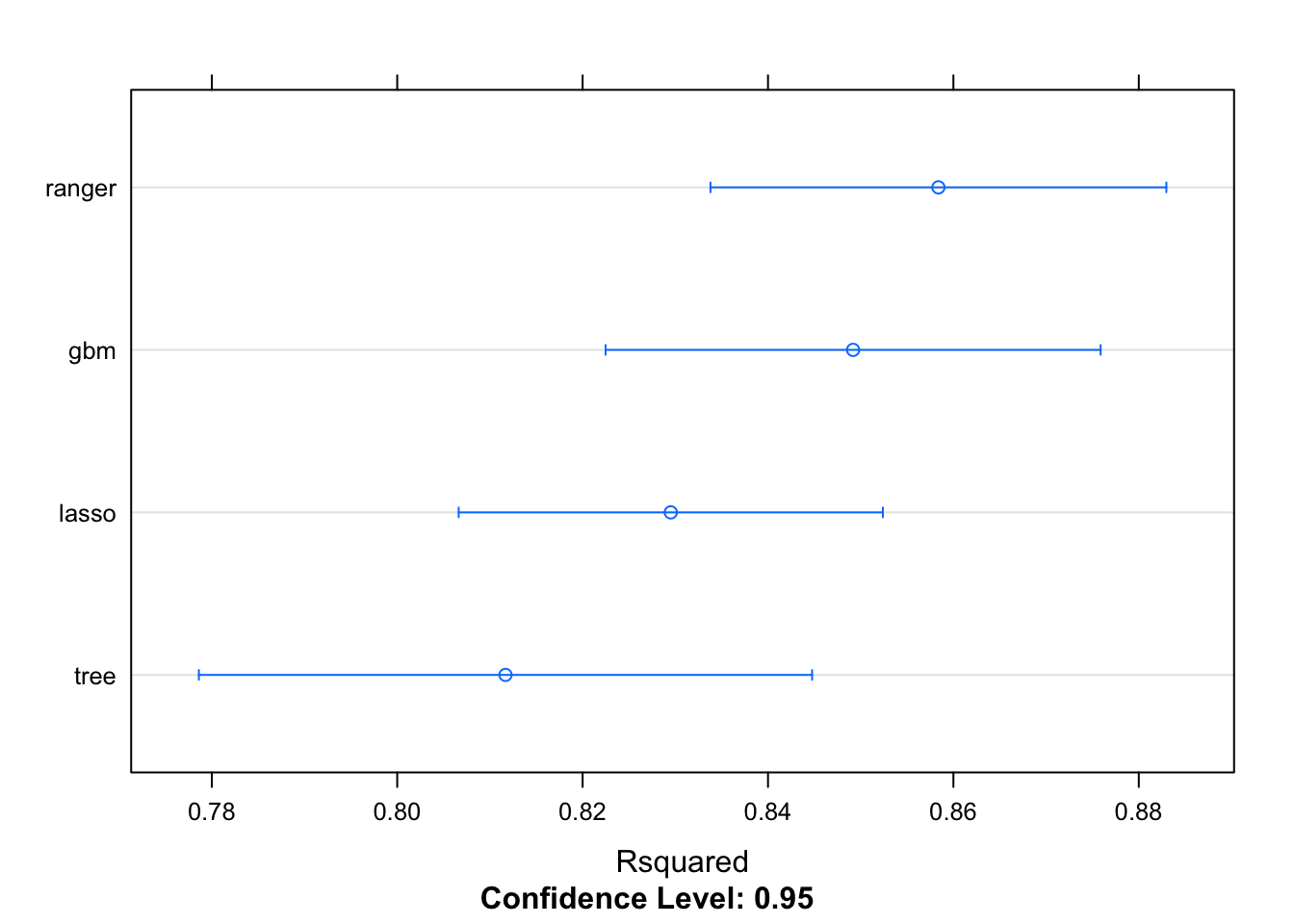
Next, I build a model using LASSO Regression to find the relationship between the price and the dependent variables. Lasso Regression is an L1 regularisation method that uses the concept of penalisation to shrink the regression coefficients towards zero by penalising the regression model using the L1 Norm. Through the use of a process of feature selection, Lasso Regression to reduce the model complexity and prevent overfitting. In order to find the best value of the regularisation parameter, lambda, I create a sequence of length 1000 ranging from 0 to 5000. I then build my model using the *distance\_to\_station, water\_company, property\_type, latitude, longitude, postcode\_short,* and *total\_floor\_area\*(as.factor(london\_zone))* variables to obtain an adjusted R-Squared of 84.22% on the testing set.

*Fig 3.1 Fluctuations in RMSE with Reguralization Parameter, lambda*

v. Random Forest

I finally use the Random Forest Machine Learning method to create an ensemble model aggregating multiple decision trees. I train the *mtry* parameter using expand.grid with values between 5 and 10. The best mtry I yield is 8 and the corresponding R-squared is 83.77%. The parameter *mtry* was trained using expand.grid with values between 10 and 15. The best mtry after tuning was 9 and the Rquare corresponding to this finalized model was 85.48%.

vi. Stacking

I now combine all of the best models that I created together to build a model that will be able to evaluate more precise predictions by *stacking* them together - this is an ensemble method that combines heterogeneous weak learners to create a powerful meta-model. Since the Lasso Regression and Linear Regression are highly correlated, I build my stacked learner from the Decision Tree, the Lasso Regression, the Random Forest, and the GBM. I perform 15-fold cross-validation to confirm the performance of the stacked model and find an adjusted R-squared of 84.68% on the training set and 88.03% on the testing set.

*Fig 3.2: Visualisation of Models Being Stacked*

# 4. Choosing the Top 200 Investments

Now that I have obtained the best model, I set the seed and compute the asking price and predicted price in the unknown sample investment points and use the formula

*profitMargin=(predict-asking\_price)/(asking\_price)*

to compute the profit margins, sort the entries in decreasing order of margins, and choose the top 200 rows as the most attractive choices and set the value of the *buy* variable to 1. The mean of my final profit margins is around 70.22%.

# 5. Elizabeth line

The Elizabeth Line is a new addition to the tube rail system that entails an investment of £20 billion which is intended to reduce the commute times to Central London considerably. The creation of this tube line will allow people from the outskirts of London or those in zones from 3 to 7 to access Central London quicker. My initial hypothesis is that the introduction of the Elizabeth line will increase the prices of the properties in Zones 3 to 7 considerably with a smaller impact on Zone 2 house prices and a negligible impact on Zone 1 house prices. Under the assumption that a *distance\_to\_elizabeth\_line* variable can be created in order to store the distance of each property from the Elizabeth Line, I would like to explore the correlations between the *distance\_to\_elizabeth\_line* variable, the *prices* variable, and the *london\_zone* variable.

In order to further prove this hypothesis, it is essential to build a few crucial models - the first one to determine how the house prices would increase over the span of the next few years using historical data if the Elizabeth line were not implemented. To build an accurate indicator of how the change of house prices after establishing a new can be observed I will use the historical data from that of the Jubilee Line. I will build two prediction models: the first performs price prediction assuming that the concept of the Elizabeth line never existed and the second learns the price fluctuations that happened after the Jubilee line was implemented - taking inflation into account in all prices. After training my machine learning models on the historical data from the Jubilee line, I will perform house price predictions for what will happen if the Elizabeth line is implemented taking the time value of money into account and using the *distance\_to\_elizabeth\_line* variable instead of the *distance\_to\_jubilee\_line* variable. Once I obtain the prices for both cases - what would happen if Elizabeth line never existed and what would happen if it did - I would calculate the percentage change in the house prices to determine if there still exists an opportunity to profit from the implementation of the Elizabeth line. I would also analyse and find which properties will receive the maximum appreciation due to the Elizabeth line to determine the most lucrative investment opportunities. For analysing neighbourhoods instead of houses in particular, I would consider the mean house prices in the neighbourhoods (grouped together by districts or postcodes - based on the investor’s interest) to analyse the most lucrative neighbourhoods.

# Conclusions

In this project, I implemented several machine learning models to perform price predictions for houses in London. I evaluated the performance of Linear Regressors, Decision Trees, Random Forests, GBMs, and Lasso Regressors to evaluate the best performing model. I finally chose to stack together the Decision Tree, GBM, Lasso Regressor, and Random Forest to create an ensemble method that yielded an adjusted R-Squared of 0.8802. Using the final model, on an investment of ~£100 million the net profit that can be made is ~£60 million.

One of the limitations in this analysis is that the dataset is too small - it only contains 10,000 records and therefore it cannot prove to be very accurate while looking at new unknown sample points. Another limitation is that the dataset used for this project doesn’t contain data for more than 2019 and therefore will not be able to perform well while using this model to perform predictions in times of crises (such as 2020). To improve the explanatory power of my stacked ensemble learner, I can train the model on the entire Price Paid dataset from 2008 to 2020.

# Appendix

# *Fig A.1 Price Distributions for Old and New*

# *Fig A.2 Price Distributions for Freehold or Leas*

# *Fi**g A.3 Price Distributions for Freehold and Leasehold*