MAT394 Report - Prediction of housing Prices

Group 9

Aravindan T S 1910110080

Adhitya Swaminathan 1910110025

Akshay Jayakumar 1810110017

Jhananii Y 1810110093

ABSTRACT

This model predicts prices of houses based on features like the square footage of the house, the number of bedrooms, the number of floors, etc. We have used three models. A multiple linear regression model, a support vector machine model and a random forest model to predict the prices. We then compare the models to see which one yields the better result. The dataset was obtained from Kaggle.

INTRODUCTION

The aim of our model is to predict housing prices based on the number of bedrooms, number of bathrooms, square footage of the house, number of floors, condition, view, square footage of the house excluding the basement, etc. We use three models to predict the housing prices - a multiple linear regression model, a random forest model and a support vector machine model.

(UPDATE :The SVM model was added later on)

The dataset is given below:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	date	price	bedrooms	bathroom	sqft_living	sqft_lot	floors	waterfron	view	condition	sqft_abov	sqft_base	yr_built	yr_renova	street	city	statezip	country
2	########	313000	3	1.5	1340	7912	1.5	0	0	3	1340	0	1955	2005	18810 Der	Shoreline	WA 98133	USA
3	########	2384000	5	2.5	3650	9050	2	0	4	5	3370	280	1921	0	709 W Blai	Seattle	WA 98119	USA
4	########	342000	3	2	1930	11947	1	0	0	4	1930	0	1966	0	26206-262	Kent	WA 98042	USA
5	########	420000	3	2.25	2000	8030	1	0	0	4	1000	1000	1963	0	857 170th	Bellevue	WA 98008	USA
6	########	550000	4	2.5	1940	10500	1	0	0	4	1140	800	1976	1992	9105 170th	Redmond	WA 98052	USA
7	########	490000	2	1	880	6380	1	0	0	3	880	0	1938	1994	522 NE 88t	Seattle	WA 98115	USA
8	########	335000	2	2	1350	2560	1	0	0	3	1350	0	1976	0	2616 174th	Redmond	WA 98052	USA
9	########	482000	4	2.5	2710	35868	2	0	0	3	2710	0	1989	0	23762 SE 2	Maple Val	WA 98038	USA
10	########	452500	3	2.5	2430	88426	1	0	0	4	1570	860	1985	0	46611-466	North Ber	WA 98045	USA
11	########	640000	4	2	1520	6200	1.5	0	0	3	1520	0	1945	2010	6811 55th	Seattle	WA 98115	USA
12	########	463000	3	1.75	1710	7320	1	0	0	3	1710	0	1948	1994	Burke-Gilr	Lake Fore:	WA 98155	USA
13	########	1400000	4	2.5	2920	4000	1.5	0	0	5	1910	1010	1909	1988	3838-4098	Seattle	WA 98105	USA
14	########	588500	3	1.75	2330	14892	1	0	0	3	1970	360	1980	0	1833 220th	Sammami	WA 98074	USA
15	########	365000	3	1	1090	6435	1	0	0	4	1090	0	1955	2009	2504 SW P	Seattle	WA 98106	USA

(over 4500 data entries of properties located in different cities in the state of Washington, US)

METHODOLOGY

We start by importing the following libraries:

1. Dplyr - Used for manipulating the data
2. Ggplot2 - Used for data visualization

3. Catools - Used to split the data into training set and test set
4. Corrgram - Used to make a correlation matrix plot
5. randomForest - Used to make a random forest model.
6. relaimpo - Used to plot variable importance.
7. ggcorrplot - Used to plot the correlation matrix.
8. e1071 - Used to make the SVM model.

We then import the dataset and start checking for any missing values which in our case were none. We then proceed to feature scale the data. We remove parameters like date, waterfront etc which have very negligible effect on the prices. We find these parameters using the forward and backward feature selection method. In the forward method, the software looks at all the predictor variables you selected and picks the one that predicts the most on the dependent measure. That variable is added to the model. This process is repeated until the best model is obtained. In the backward method, all the predictor variables you choose are added into the model. Then, the variables that do not (significantly) predict anything on the dependent measure are removed from the model one by one.

Then we use the corrgram library to make a correlation plot to see which are the parameters affecting the price which we are supposed to predict.Our data is now cleaned and we can proceed to the next step which is splitting the dataset and creating the necessary models.

```
#forward and backward step to determine base features
base.mod <- lm(price ~ 1 , data=df)
all.mod <- lm(price ~ . , data= df)
stepMod <- step(base.mod, scope = list(lower = base.mod, upper = all.mod), direction = "both", trace = 0, steps = 1000)
shortlistedVars <- names(unlist(stepMod[[1]]))
shortlistedVars <- shortlistedVars[!shortlistedVars %in% "(Intercept)"]
print(shortlistedVars)

#removing the rest of features
df = subset(df, select = -c(sqft_basement, sqft_lot))
head(df)

#correlation plot
corr <- round(cor(df), 1)
ggcorrplot(corr)
```

We use the caTools library to create a 80-20 split for our dataset. This library randomizes the dataset and splits it into 80 percent which we use to train our data and the rest 20 percent is used to check how accurate our model is. We then proceed to create a multiple regression model. Multiple linear regression is a technique that uses several explanatory (independent) variables to predict the outcome of a response (dependent) variable. This method is an extension of single regression which uses a single explanatory variable.

The following is the formula used for multiple line regression:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon$$

y_i = dependent variable
x_i = explanatory variables
β_0 = y-intercept (constant term)
β_p = slope coefficients for each explanatory variable
ϵ = the model's error term (also known as the residuals)

```
#training linear reg model
model <- lm(formula = price ~ ., data = trainset)
summary(model)
```

We then test the model on the test set, plot the result and find the rmse value.

We then proceed to create the random forest model. A random forest is a supervised-learning algorithm, random forest creates ensembles of decision trees to obtain more accurate predictions. The result of the random forest is the mean of all predictions of the decision trees.

```
#training random forest
rf.forest <- randomForest(price ~ .,mtry = 1,data = trainset,importance=TRUE)
```

We create a variable importance plot using the random forest model. Then we test the model on the test set, plot the result and find the rmse value.

Now we create the **support vector machine model**. The idea of SVR is to consider the points within the decision boundaries. The line of best fit in this case is a hyperplane which contains the maximum number of points. Here the error parameter epsilon represents the decision boundaries.

Let the hyperplane equation be Y = mx + b. Then the decision boundary equations are

mx + b = a and mx + b = -a.

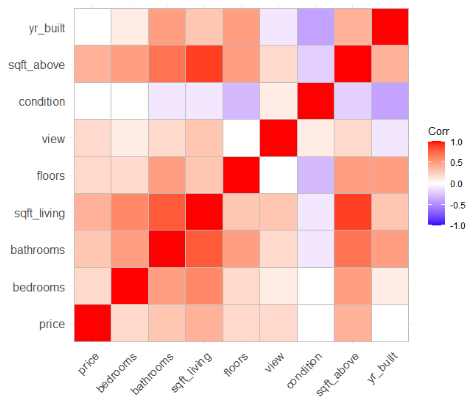
Our goal here is to decision boundary at 'a' distance from the original hyperplane such that all the data points are closest to the hyperplane.

```
#creating model using support vector machines
model_svm <- svm(price ~ ., trainset)
```

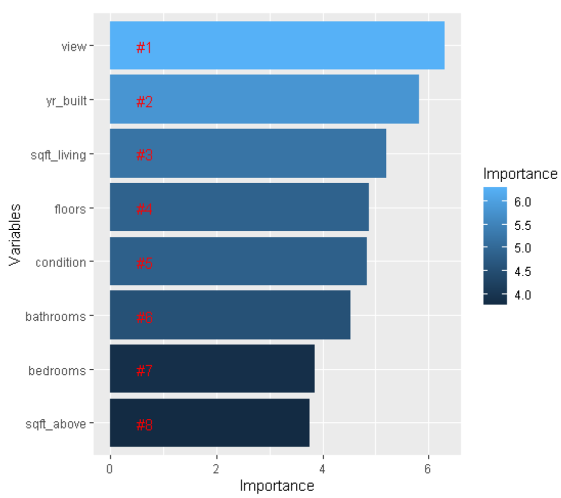
Then we test the model on the test set, plot the result and find the rmse value.

RESULTS

Correlation plot(multiple regression)

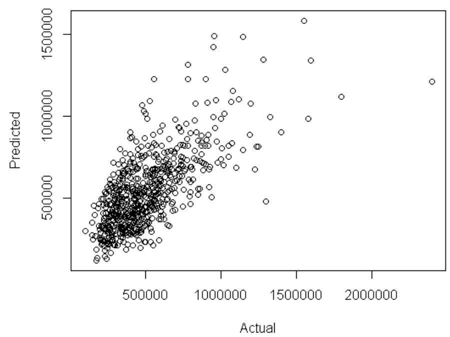


Variable importance(random forest)

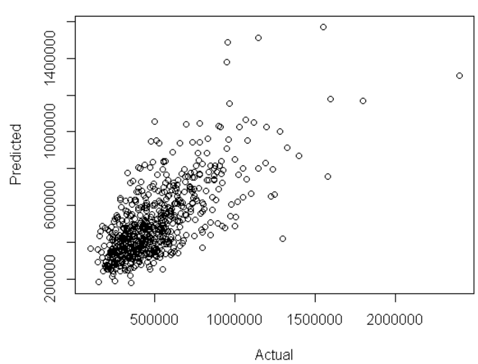


The graphs from the three models are attached below.

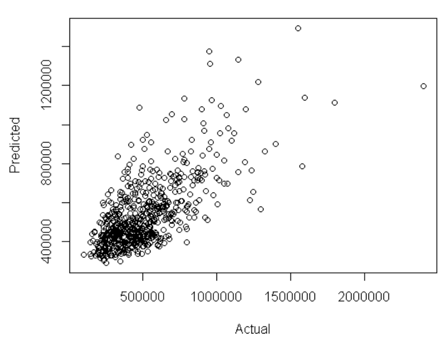
1) **Multiple regression**



2) Random forest



3) Support vector machines(SVM)



The rmse value for our random forest model was 184260.9 and 192458.4 for the regression model.

UPDATE: SVM was added, and the rmse value was 181512.5

CONCLUSION

We have found the random forest model to be more efficient in the prediction of housing prices than the multiple regression model.

UPDATE: Upon adding the SVM model, we found that it was the most efficient of the three models used.

REFERENCES

1. Abdul Qureshi, *Multiple Linear Regression using R to predict housing prices,* https://medium.com/@aqureshi/multiple-linear-regression-using-r-to-predict-housing-prices-c1ba7fe1674a (accessed April 25, 2021).
2. *R - Multiple Regression,* https://www.tutorialspoint.com/r/r_multiple_regression.htm (accessed April 25, 2021).
3. *Random Forest in R | Random Forest Algorithm | Random Forest Tutorial | Machine Learning | Simplilearn,* https://www.youtube.com/watch?v=HeTT73WxKIc (accessed April 25, 2021).
4. Building Regression Models in R using Support Vector Regression (accessed April 29, 2021)