HOUSING RENTAL PRICE PREDICTION USING MACHINE LEARNING

PROJECT REPORT

Gunjan Shah

Zinat Saneeinia

JULY 29, 2019

UNIVERSITY OF WATERLOO
200 University Eve West, Waterloo, ON, N2L3G1

Abstract

In this project, we aim at developing a machine learning application that identifies opportunities in predicting the student housing rental price. This program can be useful for both investors interested in the student housing market and international students who are in the search of accommodation. We have focused in a use case considering real houses and apartments located in the Waterloo, Ontario, Canada. We created a questionnaire for the students of University of Waterloo and did several web scraping to collect valid and reliable data. Then we performed a feature engineering stage in order to discover relevant features that allows for attaining a high predictive performance. Several machine learning algorithms have been tested, including multiple linear regression, decision tree regression, clustering and apriori algorithms and random forest regression along with identifying advantage and handicap of each of them.

Table of Contents

List of Figures

List of Tables

1. Introduction

It is common that the new students find it difficult to choose the suitable house or apartment in the beginning of their studies especially if the city or area is unknown to them and renting a house is undoubtedly one of the most important decisions they make in their education life. They usually spend a lot of time to find affordable place since the price of house may depend on a wide variety factors. In other words, higher variability can be found when looking at specific assets. Most important factors driving the value of a student house or apartment are the size and the location, but there are many other variables that are often taken into account when determining its value: number of bedrooms, proximity to some form of public transport (buses, trains, etc.), number and quality of malls/stores in the area, availability of 24/7 security (especially in apartments with several units for in each floor), availability of terrace or car parking, etc. However, certainly, the main force ultimately determining the value of houses is demand.

The housing market is also one crucial elements of the national economy. According to 2018 Canadian Rental Housing Index developed by the BC Non-Profit Housing Association and Vancity Credit Union [1], between 2011 and 2016 the number of renter increased 7,395 to 47,460 in Kitchener and Waterloo as the number of Waterloo students and alumni have raised significantly [2], [3]. During that same period, average income increased by 15 per cent and 12 per cent respectively, while average rent went up by about 20 per cent [2]. Based on the result of public consultation, some low to middle income households are renting places for more they can afford due to the lack of affordable rental units, especially in the core areas. In addition, the existing affordable housing for low income households is often poorly maintained and lacks adequate accessibility to services and amenities [4]. This shows why a strong housing rental price predictor is absolutely necessary not only for renters but also for agents and economic professionals. Fortunately, the improvement of machine learning methods and availability of big data have paved the way for building housing price predictor.

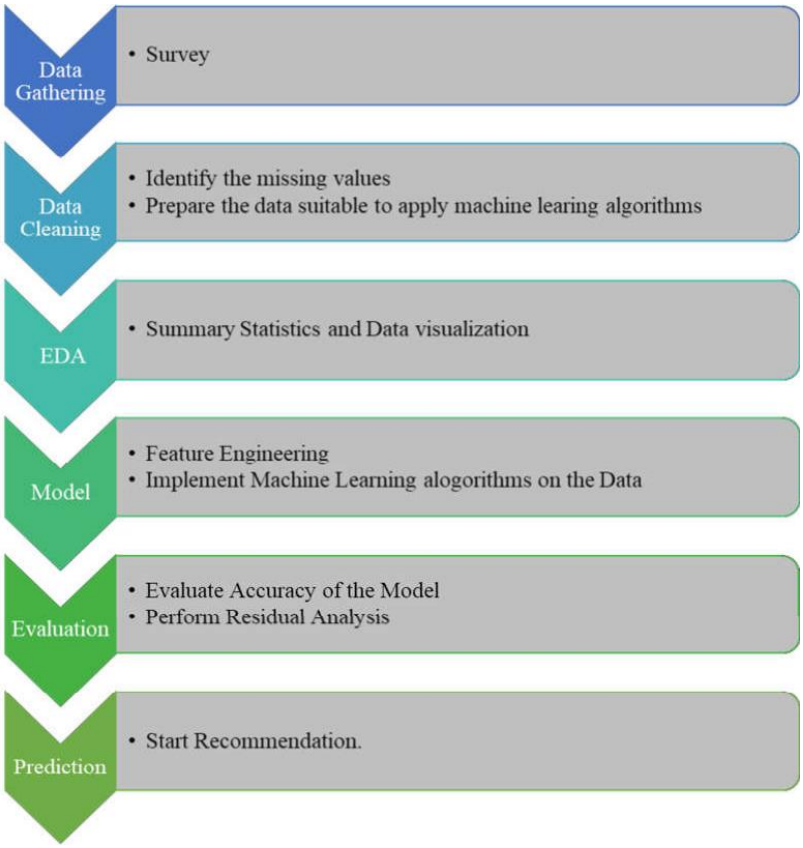
The report is arranged as follows. Section 2 consists of literature review and related work. Section 3 presents the process data gathering, exploratory data analysis, handling missing values in data and data visualization. Model development and Results including insights are discussed in Section 4. Lastly, Conclusion and proposed work is presented in Section 5.

2. Related work

The most popular house price valuation studies focus on the estimation of house values are the United States [5], [6], [7], [8], [9], [10]; Europe [11], [12], [13]; as well as China [14], [15]; and Taiwan [16]. However, research on the housing market by applying data analytics with machine learning algorithms in Canada is rare, or elusive to find. These studies seek useful models to predict the house price given characteristics like location, land size, and the number of rooms. Support Vector Machine (SVM) and its combination with other techniques have been commonly adopted for house value prediction. For instance, [15] integrates SVM with Genetic Algorithm to improve accuracy, while [16] combines SVM and Stepwise to effectively project house prices. Furthermore, other methods such as Neural Network and Partial Least Squares (PLS) are also employed for house values prediction [2].

The goal of this study is through analyzing a real dataset to derive valuable insight into the housing market in Waterloo city. It seeks useful models to predict the value of a house given a set of its characteristics using machine learning algorithms. Effective models could allow home renters, or real estate agents to make better decisions. Moreover, it could benefit the projection of future house prices and the policymaking process for the real estate market. The following diagram describes the flow of our project.

Project Flow



3. Exploratory Data Analysis

3.1 Data Set

One questionnaire created using Google form website and has about 13 questions which take approximately 5-10mins. Collection of data was done by 3 methods:

1. Web-link sent to students by mails individually. The web-link of questionnaire is given below.

Questionnaire Google form:

https://docs.google.com/forms/d/e/1FAIpQLSf3PmhHBhbYMo1mviR8uc3wcUx7PkKW84IbbAZjlOnwJR6tyg/viewform?vc=0&c=0&w=1

2. Web-link was sent to graduate students personally through social media applications.

3. Word of mouth (meeting grad students in the department and requesting for response)

Around 100 responses have been received from these 3 methods. Due to shortage of data, we extended our dataset by web scraping through Waterloo rental agencies websites:

1. https://www.kw4rent.com/
2. https://www.accommod8u.com/
3. http://www.schembripm.com/
4. https://atlasproperty.com/
5. https://www.kijiji.ca/h-kitchener-waterloo/1700212

So, our data set include 200 cases.

3.2 Data Visualization

We have 18 explanatory variables and 1 class variable for our project (Table 1 and Table 2).

Table1. Explanatory Variables

No.	Explanatory Variable	Possible Values	Data Type
1	Enrollment Term	Fall, Winter, Spring	Categorical
2	Building Type	Apartment, House	Categorical
3	Room Type	Master Room, Regular Room, Basement, Other	Categorical
4	Furnished?	Yes, No	Binary
5	Shared or Private?	Shared, Private	Categorical
6	Number of bedrooms?	1,2,3,4,5,6,7	Numeric
7	Internet	Yes, No	Binary
8	Hydro	Yes, No	Binary
9	Air Conditioning	Yes, No	Binary
10	Laundry	Yes, No	Binary
11	Car Parking	Yes, No	Binary
12	24/7 Security	Yes, No	Binary
13	Terrace	Yes, No	Binary
14	Location Ward	Southwest, Northwest, Lakeshore, Northeast,	Categorical
15	House Age	New, Middle, Old	Categorical
16	Number of Bathroom	1,2,3,4,5,6,7	Numeric
17	Market commute (Minutes)	5,10,15,20,25,35,50	Numeric
18	Bus Stop Commute (Minutes)	3,8,13,18,23	Numeric

Table2. Class Variable

No.	Explanatory Variable	Possible Values	Data Type
1	Rent Price (CAD)	200,275,300, 350,375, 400, 450,500, 550, 600, 650, 700, 800, 900, 1100, 1300, 1500, 1700, 1900	Numeric

In this Section we want to understand the data set by looking at statistical information related to each of our variables, visualizing the data to look at obvious trends and analyzing the variables and their relationships to each other. In order to introduce our findings, the data set will first be introduced to the reader, followed by visualization of the data and representations of the features and their relation to each other. In order to visualize the data, a plot of all variables related to one another was made. Figure below represents each feature on the y-axis in relation to itself and other features on the x-axis. After carefully observing the relationships, a few were singled out for further analysis.

The figure below shows that the highest number of houses available for rent from our dataset is located in Central columbia ward. As Waterloo is a student driven city, this actually makes sense. Central columbia ward is located such that all the universities in the area are in vicinity. Students prefer the houses or apartments that are near to their university so that it becomes easy for them to commute.

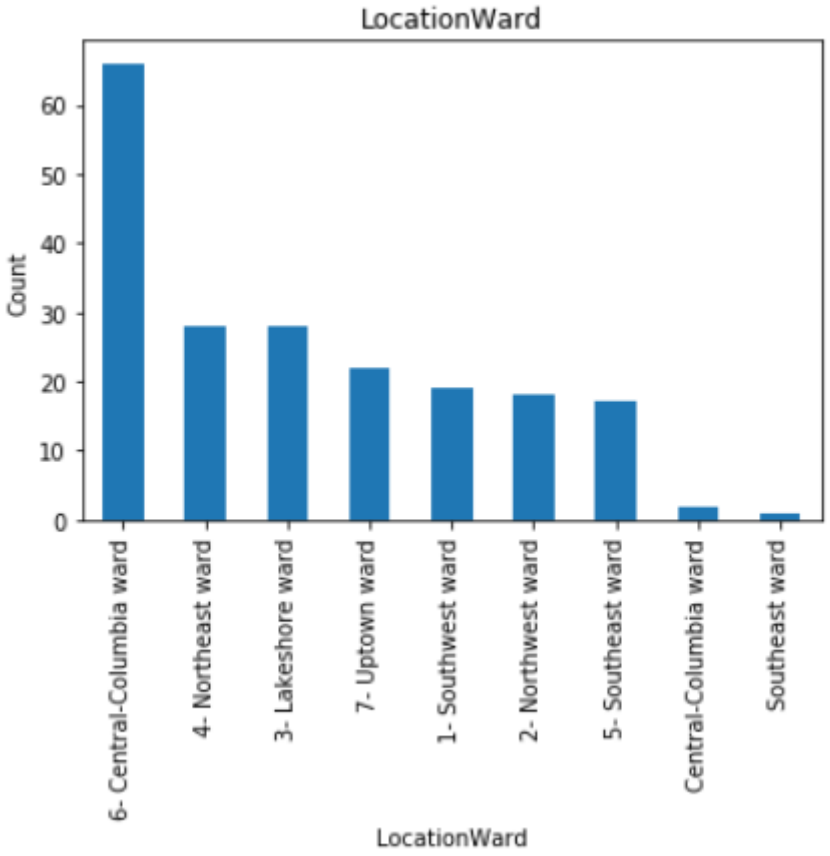


Figure (1) – Number of houses versus Location Ward

As we know, different features and facilities play vital role in houses or apartments rental price. So. The next step is to check the count of different facilities that are available in our data set filled with 200 cases. For instance, Figure 2 illustrate that the number of houses which are near bus stops is more than 120 out of 200 or the number of houses with air conditioning is more than 150 out of 200. Most of the houses has no 24/7 security option and the most interesting thing is the rent price which is around $600 to $700 that paid in more than 80 cases.

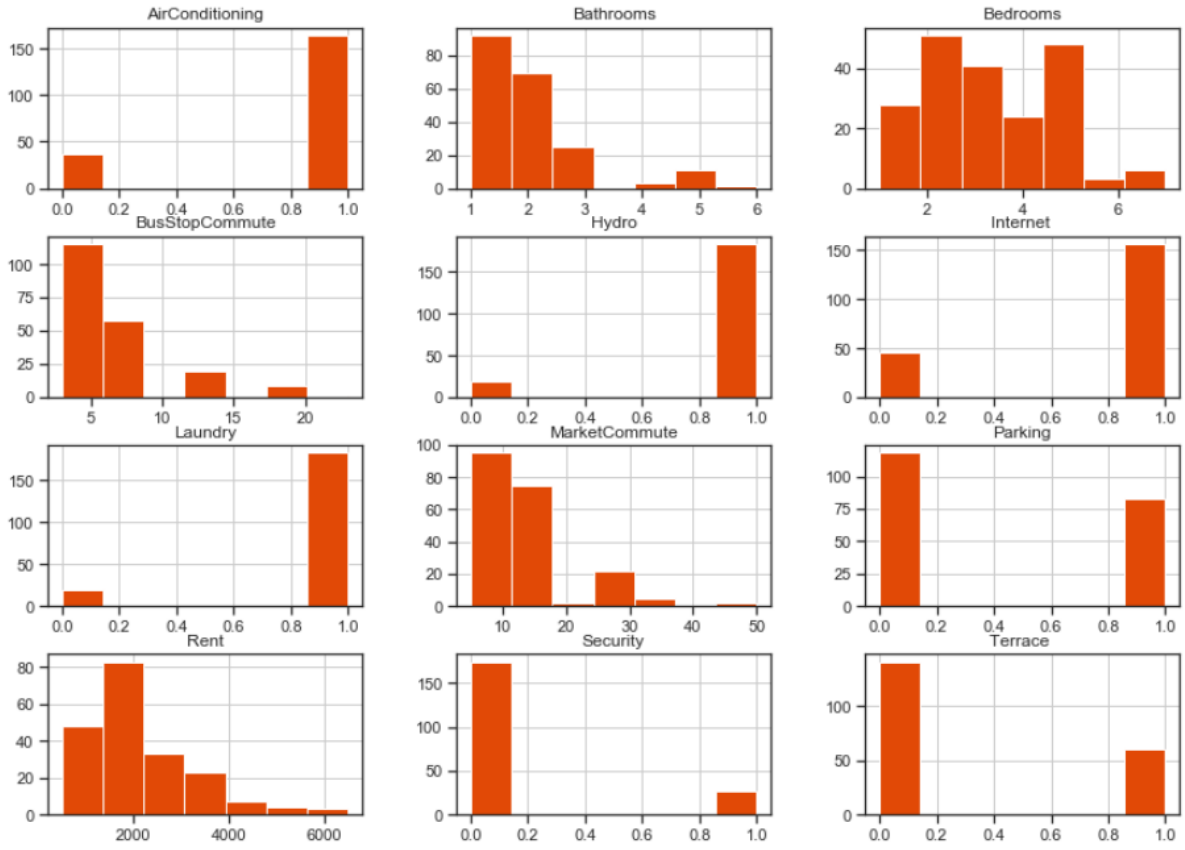


Figure (2) – Number of houses versus different facilities

As we are predicting rent of the house, Q-Q plot and histogram plot will reveal the nature of this dependent variable. The plot reveals fairly normal distribution.

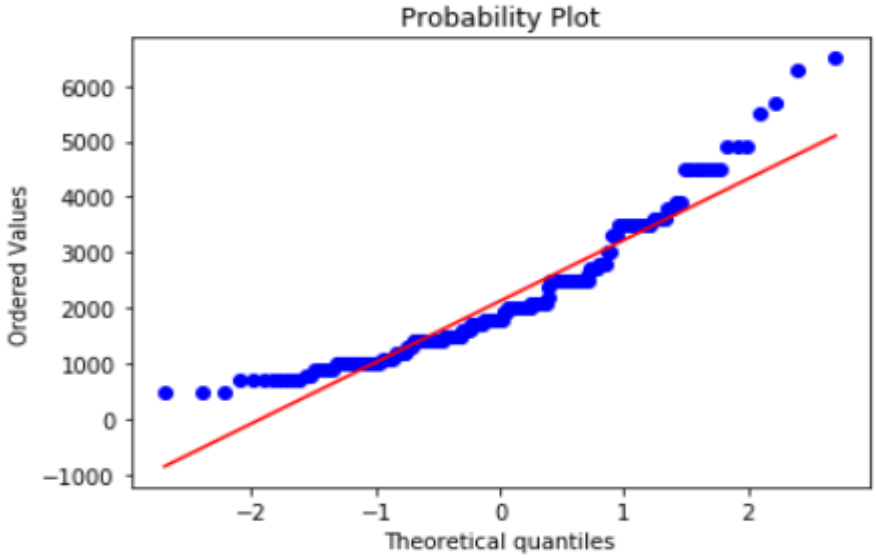


Figure (3) – QQ plot of outcome variable (housing rental price)

Moreover, boxplot reveals the values that can be potential outliers. It does make sense since there are always expensive houses with higher rent value in real world.

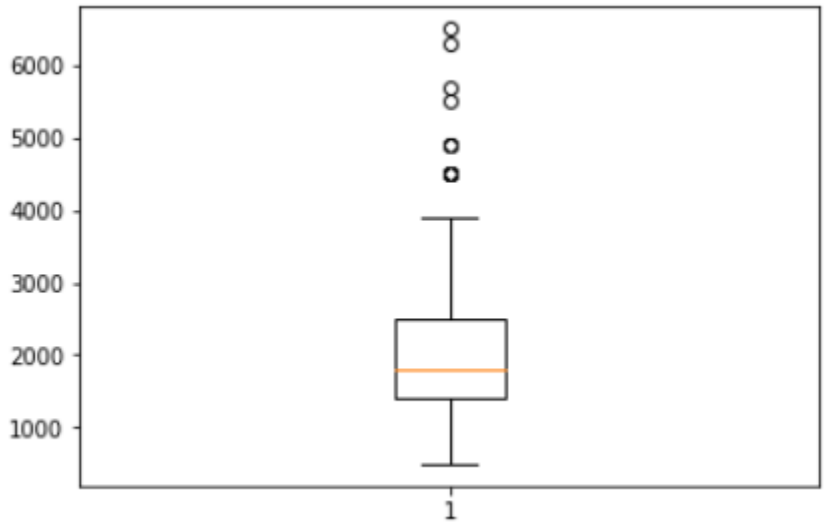


Figure (4) – Boxplot of outcome variable (housing rental price)

3.3 Data Cleaning

The most important step before starting to apply any process on the data is to clean the data. The first step was to check the missing values. Fortunately, as all the questions on questionnaire were mandatory and other data was collected using web scraping, there was no missing data in the formed csv.

3.4 Correlation and variable selection

After analyzing the data set for each variable separately, we decided to use the data to predict our target variable as rent paid by the tenants. Keeping that in mind, a correlation matrix was constructed to see if there are any strong direct linear relationships between our target variable and the other features. According to figure 3, from first inspection as can be seen in the correlation heat map, there does seem to be high correlation between one variable and the target variable. Number of bedrooms and bathrooms and the type of bedroom shows high positive correlation with rent. However, correlation between 24/7 security or having laundry with rent value is not significant which is less than 0.1.

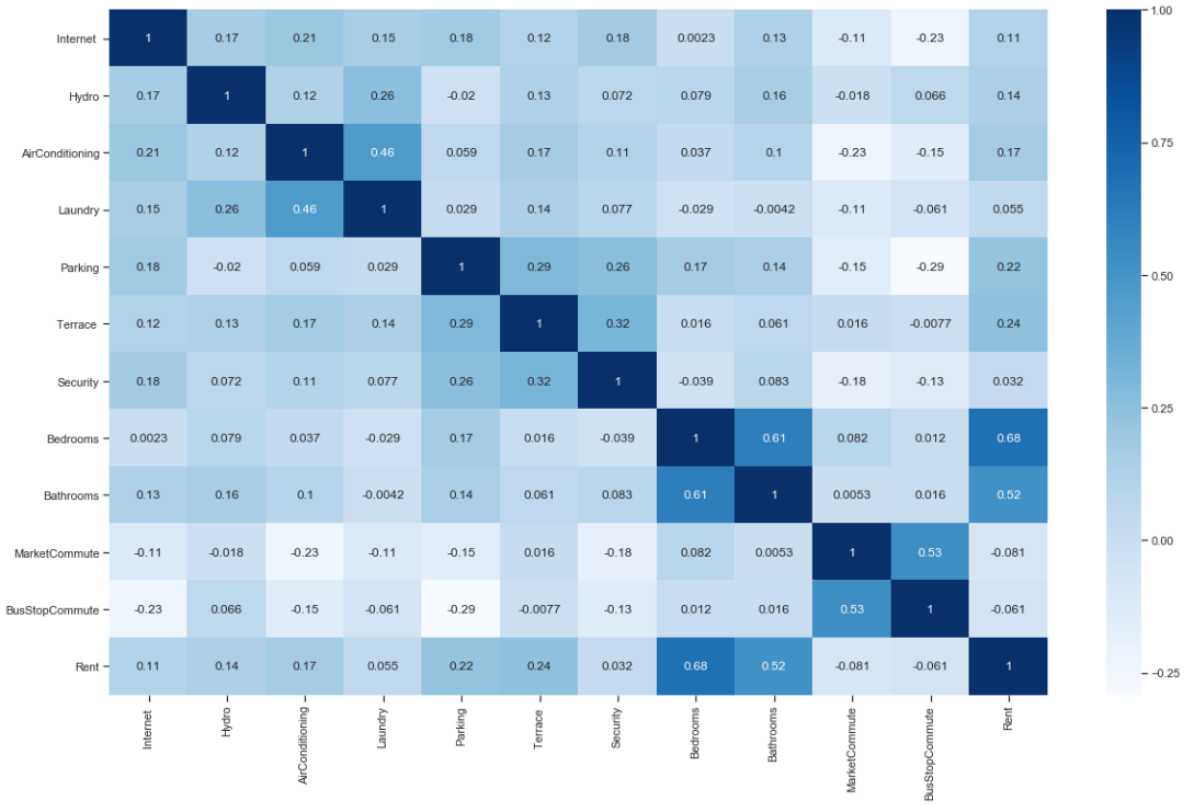


Figure (5) – Correlation Matrix between features and housing rental price

As rent is a numeric variable, we decided to apply regression techniques for further analysis. So, selecting the predictor variables that correlate with each other will introduce multicollinearity. Based on that, correlation using Pearson's R was done on the features that revealed that number of bedrooms and number of bathrooms are highly correlated. Therefore, as number of bedrooms have relatively high correlation with rent, we decided to drop the number of bathrooms variable and continued with other features. In order to run histogram, Q-Q plot and Pearson's R to visualize the data, we checked all assumption such as normality, homoscedasticity and independency of variables. To conclude, bar graph is used for visualize number of houses in each ward and histograms for number of houses with different features. Also, to check correlation between variables Pearson's R was implemented.

3.5 Clustering Algorithm:

Before doing clustering, we need to find out the number of clusters that we need for our data set. We found out optimum numbers of clusters using the Elbow method. So, based on the figure 6, we decided to use three clusters.

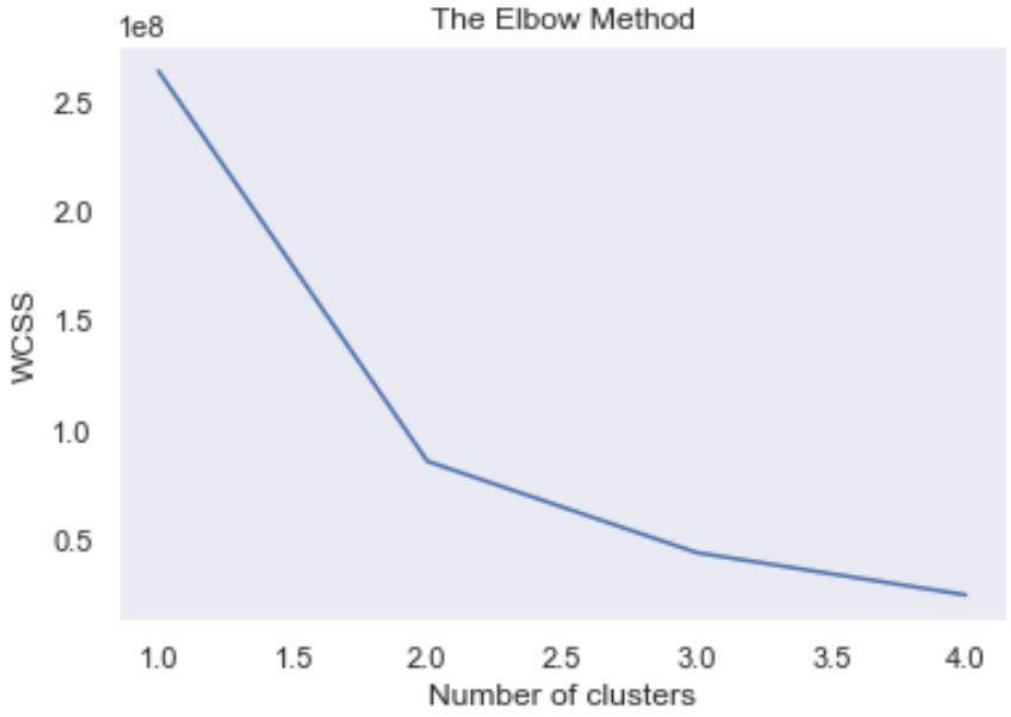


Figure (6) – Elbow method

The output of the clustering algorithm is shown in figure 7. The blue dots represent the centroid of three groups. Y axis shows the rent value whereas X axis is the number of data points. As we see, when there are more bedrooms in the house the rent would be higher and market commute time do not affect the rent value as much as number of bedrooms (According to the correlation matrix in figure 5).

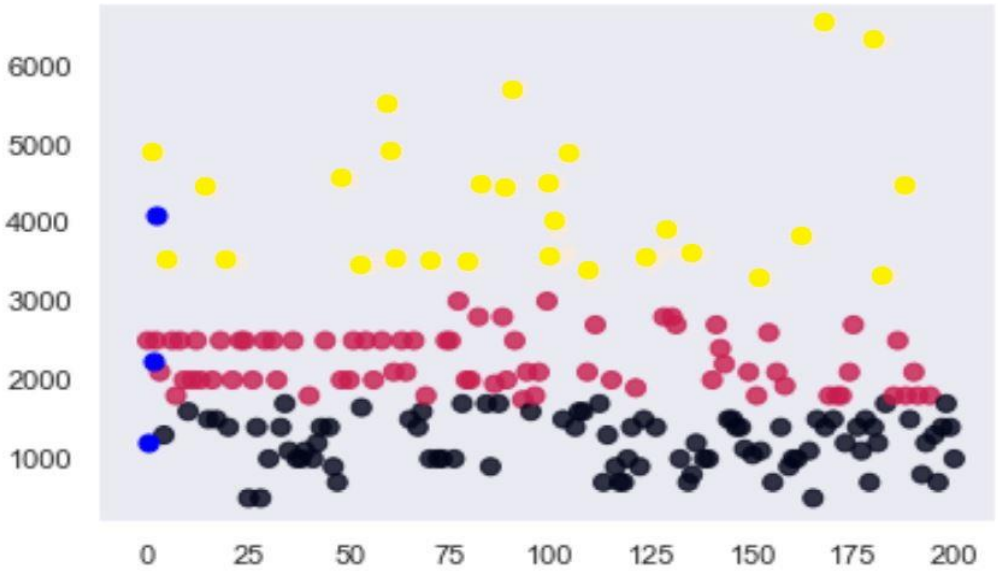


Figure (7) – Clusters

3.6 Apriori Algorithm:

To find the association between different features we decided to find out the support confidence and left using apriori algorithm. According to our result, features like Hydro, laundry, furnished_Yes has support higher than 0.9 which suggest that these are very common options in each house. Picking the appropriate values for support and confidence can be difficult, as it is very much and unsupervised process. With few try and error, we got reasonable good value of support and confidence.

	support	itemsets
0	0.512438	(EnrolmentTerm_Fall)
1	0.283582	(EnrolmentTerm_Spring)
2	0.203980	(EnrolmentTerm_Winter)
3	0.696517	(BuildingType_Apartment)
4	0.303483	(BuildingType_House)
5	0.213930	(RoomType_Master Room)
6	0.671642	(RoomType_Regular Room)
7	0.184080	(Furnished_No)
8	0.815920	(Furnished_Yes)
9	0.273632	(SharedorPrivate_Private)
10	0.726368	(SharedorPrivate_Shared)
11	0.412935	(HouseAge_Middle)
12	0.343284	(HouseAge_New)
13	0.243781	(HouseAge_Old)
14	0.139303	(LocationWard_3- Lakeshore ward)
15	0.139303	(LocationWard_4- Northeast ward)
16	0.328358	(LocationWard_6- Central-Columbia ward)
17	0.109453	(LocationWard_7- Uptown ward)
18	0.776119	(Internet)
19	0.910448	(Hydro)
20	0.815920	(AirConditioning)
21	0.905473	(Laundry)
22	0.412935	(Parking)
23	0.298507	(Terrace)
24	0.134328	(Security)
25	0.338308	(EnrolmentTerm_Fall, BuildingType_Apartment)
26	0.174129	(EnrolmentTerm_Fall, BuildingType_House)
27	0.368159	(RoomType_Regular Room, EnrolmentTerm_Fall)
28	0.412935	(EnrolmentTerm_Fall, Furnished_Yes)

Figure (8) – Apriori algorithm

	antecedents	consequents	antecedent support	consequent support	support	confi
0	(EnrolmentTerm_Fall)	(RoomType_Regular Room)	0.512438	0.671642	0.368159	0.7
1	(EnrolmentTerm_Fall)	(Furnished_Yes)	0.512438	0.815920	0.412935	0.8
2	(EnrolmentTerm_Fall)	(SharedorPrivate_Shared)	0.512438	0.726368	0.368159	0.7
3	(EnrolmentTerm_Fall)	(Internet)	0.512438	0.776119	0.388060	0.7
4	(EnrolmentTerm_Fall)	(Hydro)	0.512438	0.910448	0.447761	0.8
5	(EnrolmentTerm_Fall)	(AirConditioning)	0.512438	0.815920	0.412935	0.8
6	(EnrolmentTerm_Fall)	(Laundry)	0.512438	0.905473	0.452736	0.8
7	(EnrolmentTerm_Spring)	(BuildingType_Apartment)	0.283582	0.696517	0.223881	0.7
8	(EnrolmentTerm_Spring)	(Furnished_Yes)	0.283582	0.815920	0.233831	0.8
9	(EnrolmentTerm_Spring)	(SharedorPrivate_Shared)	0.283582	0.726368	0.228856	0.8
10	(EnrolmentTerm_Spring)	(Internet)	0.283582	0.776119	0.233831	0.8
11	(EnrolmentTerm_Spring)	(Hydro)	0.283582	0.910448	0.278607	0.9
12	(EnrolmentTerm_Spring)	(AirConditioning)	0.283582	0.815920	0.243781	0.8
13	(EnrolmentTerm_Spring)	(Laundry)	0.283582	0.905473	0.273632	0.9
14	(EnrolmentTerm_Winter)	(Furnished_Yes)	0.203980	0.815920	0.169154	0.8
15	(EnrolmentTerm_Winter)	(Internet)	0.203980	0.776119	0.154229	0.7
16	(EnrolmentTerm_Winter)	(Hydro)	0.203980	0.910448	0.184080	0.9
17	(EnrolmentTerm_Winter)	(AirConditioning)	0.203980	0.815920	0.159204	0.7
18	(EnrolmentTerm_Winter)	(Laundry)	0.203980	0.905473	0.179104	0.8
19	(RoomType_Regular Room)	(BuildingType_Apartment)	0.671642	0.696517	0.507463	0.7
20	(BuildingType_Apartment)	(RoomType_Regular Room)	0.696517	0.671642	0.507463	0.7
21	(Furnished_Yes)	(BuildingType_Apartment)	0.815920	0.696517	0.572139	0.7
22	(BuildingType_Apartment)	(Furnished_Yes)	0.696517	0.815920	0.572139	0.8
23	(SharedorPrivate_Shared)	(BuildingType_Apartment)	0.726368	0.696517	0.537313	0.7
24	(BuildingType_Apartment)	(SharedorPrivate_Shared)	0.696517	0.726368	0.537313	0.7
25	(HouseAge_New)	(BuildingType_Apartment)	0.343284	0.696517	0.303483	0.8
26	(LocationWard_4- Northeast ward)	(BuildingType_Apartment)	0.139303	0.696517	0.109453	0.7
27	(LocationWard_6- Central-Columbia ward)	(BuildingType_Apartment)	0.328358	0.696517	0.248756	0.7
28	(Internet)	(BuildingType_Apartment)	0.776119	0.696517	0.547264	0.7
29	(BuildingType_Apartment)	(Internet)	0.696517	0.776119	0.547264	0.7

Figure (9) – Apriori algorithm 2

We find out new houses with Internet, laundry, furnished_Yes and hydro has confidence greater than 0.85. This will help new property builder to include different facilities in order to increase the sales or rent value.

4. Results

4.1 Encoding

Label Encoding is used to convert the categorical data to numeric data. In label encoding, each possible value of the feature is encoded into an integer value.

The resulting data of Label encoding is used for multiple linear regression, decision tree regression and random forest regression.

We also checked for all the assumptions that apply to prediction algorithm used.

4.2 Cross validation

In order to have unbiased training and testing data set, we applied k fold cross validation method. Moreover, we checked for stratified k fold and leave one out k fold method. We decided to use 6 splits to properly divide our data in training and testing set.

4.3 Approaches and comparison

First, we did **multiple linear regressions**. The selection of features was done based on correlation matrix that we saw in figure 5. The accuracy that we got was about 62.2% without using k fold. After applying the k fold, we got an accuracy of 65%. By applying cross_val_score function the accuracy of 69% is achieved.

The second method we applied is **decision tree regression**. We checked the accuracy here by obtaining mean absolute error, mean square error and root mean square error. Both MAE and RMSE express average model prediction error in units of variable of interest. They are negatively oriented scores, which mean lower values are better. We got MAE equals to 35%, MSE equals to 22% and RMSE equals to 47%. The lower values of error suggest that model performed well.

The third method we applied is **random forest regression.** We also checked the accuracy here by obtaining mean absolute error, mean square error and root mean square error. Both MAE and RMSE illustrate average model prediction error in units of variable of interest. They are negatively oriented scores, which mean lower values are better. We got MAE equals to 33%, MSE equals to 19.5% and RMSE equals to 44.2%.

By comparing all three models, we found out that random forest regression outperforms the other two models. Decision tree are prone to overfitting, especially when a tree is particularly deep. Random forest is able to discover more complex relation at the cost of time.

5. Conclusion and Future Implementation

In summary, this paper seeks useful models for house price prediction. It also provides insights into the Waterloo housing Market. Firstly, the original data is prepared and transformed into a cleaned dataset ready for analysis. Data reduction and transformation are then applied by using regression techniques. Then we applied apriori and clustering method. Different methods are then implemented and evaluated to achieve an optimal solution. The evaluation phase indicates that the random forest regression will give us prediction housing rental values more precisely than decision tree and multiple linear regression models. So, this model can be used for further deployment. This research can also be applied for transactional datasets of the housing market from different locations across Canada.

For further investigation, it is suggested to deploy two models: Stepwise - SVM, and Polynomial regression to predict observations with no outcome values. Polynomial regression can act as a new baseline for comparing prediction results. This implementation should be rigorously tested on historical datasets from different cities in Canada. The results could help to improve the performance and accuracy of these models.

6. References

[1] www.rentalhousingindex.ca

[2] www.kitchenerpost.ca/news-story/8608328-rental-housing-becoming-more-unaffordable-census-data-shows/

[3] engineerthefuture.ca/get-know-us/waterloo-engineering-fast-facts/

[4] uwspace.uwaterloo.ca/bitstream/handle/10012/12431/Xinyue_Pi.pdf?sequence=5&isAllowed=y

[5] Gupta, R., Kabundi, A., & Miller, S. M. (2011). Forecasting the US real house price index: Structural and non-structural models with and without fundamentals. Economic Modelling, 28(4), 2013-2021.

[6] Mu, J., Wu, F. & Zhang, A., 2014. Housing Value Forecasting Based on Machine Learning Methods. Abstract and Applied Analysis, 2014(2014), p.7.

[7] Bork L. & Moller S., 2015. Forecasting house prices in the 50 state using Dynamic Model Averaging and Dynamic Model Selection.

International Journal of Forecasting, 31(1), pp.63–78

[8] Balcilar, M., Gupta, R., & Miller, S. M. (2015). The out-of-sample forecasting performance of nonlinear models of regional housing prices

in the US. Applied Economics, 47(22), 2259-2277.

[9] Park, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Systems with Applications, 42(6), 2928-2934.

[10] Plakandaras, V., Gupta, R., Gogas, P., & Papadimitriou, T. (2015). Forecasting the US real house price index. Economic Modelling, 45,

259-267

[11] Ng, A., & Deisenroth, M. (2015). Machine learning for a London housing price prediction mobile application. Technical Report, June

2015, Imperial College, London, UK.

[12] Rahal, C. (2015). House Price Forecasts with Factor Combinations (No.15-05).

[13] Risse M. & Kern M., 2016. Forecasting house-price growth in the Euro area with dynamic model averaging. North American Journal of

Economics and Finance, 38, pp.70–85.

[14] Jie, T. J. Z. (2005). What pushes up the house price: Evidence from Shanghai [J]. World Economy, 5, 005.

[15] Gu J., Zhu M. & Jiang L., 2011. Housing price forecasting based on genetic algorithm and support vector machine. Expert Systems with

Applications, 38(4), pp.3383–3386.

[16] Chen, J.-H. et al., 2017. Forecasting spatial dynamics of the housing market using Support Vector Machine. International Journal of Strategic Property Management, 21(3), pp.273–28.

Appendix

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import mpl_toolkits
from sklearn.model_selection import train_test_split
import statsmodels.api as sm
from sklearn.preprocessing import OneHotEncoder,LabelEncoder
from sklearn.model_selection import KFold
from sklearn.linear_model import LinearRegression
import scipy.stats as stats
import warnings
warnings.filterwarnings("ignore")
import sys
import warnings
warnings.filterwarnings("ignore")
import seaborn as sns
sns.set(style="dark",color_codes=True)
import sklearn
from sklearn.cluster import KMeans

#Getting the dataset and checking the variables
data = pd.read_csv('final.csv')
data = data.iloc[:,:19]
# Printing the shape of the dataset
print('Total number of rows and columns are -> ', data.shape)
# Printing the column names of the dataset
print('Headings of the columns of the dataset are -> ',data.columns)
print('Checking the dataset: ')
print(data.head())

# Fetching only catagorical features
catagorical_feature_df = data.select_dtypes(include=['object']).copy()
print(catagorical_feature_df.head())
# Fetching only numerical features
Numerical_feature_df = data.select_dtypes(include=['int64','float64']).copy()
print(Numerical_feature_df.head())

# Plots and Graphs
for i in catagorical_feature_df:
data[i].value_counts().plot(kind='bar')
plt.title(i)
plt.xlabel(i)
plt.ylabel('Count')
plt.show()
```

```
46. sns.despine
47.
48. # Let's see how the numeric data is distributed.
49. data.hist(bins=7, figsize=(14,10), color='#E14906')
50. plt.show()
51.
52. data['Rent'].hist(bins=10, figsize=(14,10), color='#E14906')
53. xlabel = ('Rent')
54. ylabel = ('Count')
55. plt.show()
56. stats.probplot(data['Rent'], dist="norm", plot=plt)
57. plt.show()
58. plt.boxplot(data['Rent'])
59.
60. # Initial Plots for dataset
61. sns.set(style="ticks")
62. sns.pairplot(data, palette="Set1")
63.
64. # Plotting the catagorical features for dataset
65. for i in catagorical_feature_df:
66. fig = plt.figure(figsize = (20,8))
67. sns.countplot(x=i, data=catagorical_feature_df)
68. plt.show()
69.
70. # Correlation Plot
71. #using Pearson correlation (https://towardsdatascience.com/feature-selection-w
72. ith-pandas-e3690ad8504b)
73. plt.figure(figsize=(20,12))
74. # print(data.head())
75. cor = data.corr()
76. sns.heatmap(cor, annot=True, cmap=plt.cm.Blues)
77. #plt.show()
78.
79. #Correlation with output variable
80. cor_target = abs(cor["Rent"])
81. #Selecting highly correlated features
82. relevant_features = cor_target[cor_target>0.25]
83. print(relevant_features)
84. print(cor_target)
85.
86. #Creating Dummy Variables for categorical features
87. l = ["EnrolmentTerm","BuildingType","RoomType","Furnished","SharedorPrivate",
88. "HouseAge","LocationWard"]
89. one_hot = pd.get_dummies(data[l])
90. data = data.drop(l,axis = 1)
91. data = one_hot.join(data)
92. print(data.head())
```

```
print(data.columns)

data.head()
# Everything looks proper now, encoded and all dummy variables have been created

"""
Variables within a dataset can be related for lots of reasons.
        For example:
        One variable could cause or depend on the values of another variable.
        One variable could be lightly associated with another variable.
        Two variables could depend on a third unknown variable.
        The Pearson correlation coefficient (named for Karl Pearson) can be used to s
    u
        mmarize the strength of the linear relationship between two data samples.
        The Pearson's correlation coefficient is calculated as the covariance of the t
        wo variables divided by the product of the standard deviation of each data sa
    m
        ple.
        It is the normalization of the covariance between the two variables to give an

        interpretable score.
        '''

        from scipy.stats import pearsonr
        corrData = []
        for i in data.columns:
            for j in data.columns:
                if i =="Rent" or j == 'Rent':
                    break
                corr, _ = pearsonr(data[i],data[j])
                corrData.append(corr)
                if corr >0.5 and corr<1:
                    print(i,j,corr)

        #using Pearson correlation (https://towardsdatascience.com/feature-
    selection-w
        ith-pandas-e3690ad8504b)
        plt.figure(figsize=(20,12))
        # print(data.head())
        cor = data.corr()
        sns.heatmap(cor, annot=True, cmap=plt.cm.Blues)
        #plt.show()

        #Correlation with output variable
        cor_target = abs(cor["Rent"])
        #Selecting highly correlated features
```

```
relevant_features = cor_target[cor_target>0.25]
print(relevant_features)
print(cor_target)

#Setting up the dependent and independent variables
data1 = data[['Bedrooms','RoomType_Master Room', 'Rent']]
X = data1.drop('Rent', axis=1)
y = data.iloc[:, 2:3].values

# Let's begin prediction

from sklearn.model_selection import train_test_split
X_Train, X_Test, y_Train, y_Test = train_test_split(X, y, test_size = 0.2, ran
dom_state = 0)
from sklearn.metrics import mean_squared_error

# 1 - Multiple Linear regression
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_Train, y_Train)
# Predicting the test result
y_pred = regressor.predict(X_Test)
print(y_pred)
regressor.score(X_Test,y_Test)

# Defining a general funciton
def get_score(model, X_Train, X_Test, y_Train, y_Test):
model.fit(X_Train, y_Train)
return model.score(X_Test, y_Test)
get_score(regressor,X_Train, X_Test, y_Train, y_Test)

# Improvising the training and testing dataset by applying KFold cross validati
on
from sklearn.model_selection import KFold
folds = KFold(n_splits=6)
scores_linear = []
for train_index, test_index in folds.split(X,y):
    X_Train, X_Test = X.iloc[train_index], X.iloc[test_index]
    y_Train, y_Test = y[train_index], y[test_index]
    scores_linear.append(get_score(regressor, X_Train, X_Test, y_Train, y_Test
))
scores_linear
np.max(scores_linear)

# So all the training and testing data is now selected using KFold cross Validati
on.

```

```
from sklearn.model_selection import cross_val_score
print(max(cross_val_score(regressor, X, y,cv=10)))

# 2 - Decision Tree Regression
from sklearn.tree import DecisionTreeRegressor
Dregressor = DecisionTreeRegressor(random_state = 0)
Dregressor.fit(X,y)

y_pred = Dregressor.predict(X_Test)
df=pd.DataFrame({'Actual':y_Test, 'Predicted':y_pred})
df

# Checking the accuracy

from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_Test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_Test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_Te
    st, y
    _pred)))

from sklearn.model_selection import train_test_split
X_Train, X_Test, y_Train, y_Test = train_test_split(X, y, test_size = 0.2, ran
dom_state = 0)
from sklearn.metrics import mean_squared_error

# 3 - Random Forest Regression
from sklearn.ensemble import RandomForestRegressor
RFregressor = RandomForestRegressor(n_estimators = 100, random_state = 0
    )
RFregressor.fit(X,y)

y_pred1 = RFregressor.predict(X_Test)

# Checking the accuracy

from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_Test, y_pred1))

print('Mean Squared Error:', metrics.mean_squared_error(y_Test, y_pred1))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_Te
    st, y
    _pred1)))

# This shows that Random forest regressor gives comparatively less mean squ
    are
error than Decision tree regressor.
```

```
221.
222.
223.        # Before clustering, let's find out the number of clusters that we need to appl
    y
224.
225.        # Finding optimum number of clusters using The Elbow Method
226.        from sklearn.cluster import KMeans
227.        clust = dataa.iloc[:, 35:37].values
228.        wcss = []
229.        for i in range(1,5):
230.            kmeans = KMeans(n_clusters = i, init = 'k-
    means++', max_iter = 300, n_init= 10, random_state = 0)
231.            kmeans.fit(clust)
232.        wcss.append(kmeans.inertia_)
233.        plt.plot(range(1,5),wcss)
234.        plt.title('The Elbow Method')
235.        plt.xlabel('Number of clusters')
236.        plt.ylabel('WCSS')
237.        plt.show()
238.
239.        #Clustering -
240.        dataa = pd.read_csv("final.csv")
241.        dataa = pd.DataFrame(dataa)
242.        #print(dataa.head())
243.        Data = pd.DataFrame(dataa,columns = ['Rent'])
244.        import sklearn
245.        from sklearn.cluster import KMeans
246.        #print(Data)
247.        # 3 clusters
248.        kmeans = KMeans(n_clusters=3).fit(Data)
249.        centroids = kmeans.cluster_centers_
250.        print(centroids)
251.        centroids = pd.DataFrame(centroids)
252.        X= Data['Rent']
253.        cluster_map = pd.DataFrame()
254.        cluster_map['data_index'] = Data.index.values
255.        cluster_map['cluster'] = kmeans.labels_
256.        print(cluster_map[cluster_map.cluster == 0])
257.        print(cluster_map[cluster_map.cluster == 1])
258.        print(cluster_map[cluster_map.cluster == 2])
259.        #plt.scatter(Data.index, X, c= kmeans.labels_.astype(float), s=50, alpha=0.5)
260.        #plt.scatter(centroids.index,centroids, c='red', s=50)
261.        plt.scatter(Data.index, X, c= kmeans.labels_.astype(float), s=50, alpha = 0.8)
262.        plt.scatter(centroids.index,centroids, c='blue', s=50)
263.
264.        #Creating Dummy Variables for categorical features
```

```
l = ["EnrolmentTerm","BuildingType","RoomType","Furnished","SharedorPrivate",
"HouseAge","LocationWard"]
one_hot = pd.get_dummies(dataa[l])
dataa = dataa.drop(l,axis = 1)
dataa = one_hot.join(dataa)
print(dataa.head())
print(dataa.columns)

dataa.head()

new_data = dataa.iloc[:, 0:32]
new_data.head()

from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori

fme = apriori(new_data, min_support=0.1, use_colnames=True)
fme

from mlxtend.frequent_patterns import association_rules
association_rules(fme, metric="confidence", min_threshold=0.7)

# End of this file
```