Housing & Residential Safety: Exploring the Impact of Crime on Property Value in Pierce County, Washington

August 13, 2019

Preslav Angelov, Huy Le, and Evin Tolentino

**Table of Contents**

Executive Summary 3

Problem Statement 4

Dataset Description 4

Literature Review 4

Data Preprocessing 6

Data Mining Models & Evaluations 9

Discussion & Evaluation 12

Conclusion 13

References 14

Figures & Tables 15

Appendix 47

**Executive Summary**

Evaluating and predicting property values has been a hot research topic in recent years given the 2008 Great Recession and the current booming housing markets in many US cities. Although it might sound like a relatively easy question to answer, there are many factors that influence the sale price of a private property. In this research paper, we focus on how different type of crimes affects the residential property sale price in Pierce County, Washington. We worked with two sets of data. The first data source includes all physical attributes of a property, such as the square feet, quality, year built and remodeled. The second data source includes crime data for parts of Pierce county from July 2018 – July 2019. Additionally, we divided our full, integrated data into three clusters. Both the full and clustered datasets were used to build two different types of models–one with crime attributes, and the other without crime attributes. We built our models using three methods of prediction–decision trees, neural networks, and random forests–yielding 32 models to run and evaluate given the initial dataset parameters. Out of all the models, the random forest models reported the lowest values of RMSE and MAE, with little difference between with crime and without crime model variations. Accordingly, we concluded that crime does have a statistical significance in predicting the sale price of residential properties in Pierce County, Washington. However, limitations to our research should be addressed, such as the decision to use sale price versus market value of a residential property as the predicted variable. In addition, the perception of crime and crime that actually occurred needs to be further analyzed to determine the true impact on actual crime on residential property values in Pierce County, Washington.

**Problem Statement**

Usually, the value of a house that home buyers use to indicate its quality and benefit are based on the house’s attributes. Examples of these attributes include the total square footage, number of bedrooms, number of bathrooms, year built, location, etc. However, from the buyers’ and sellers’ perspective, there are other factors that are also considered when making these investment decisions.

One factor that is often considered when investing in a home is residential/community safety. However, crime is a complex factor to consider. There are many types of crimes that occur, such as homicide, burglary, and drug-related, as well as varying degrees of crime severity. Accordingly, we are interested in exploring whether crime does have an effect on housing value in Pierce County, Washington?

**Dataset Description**

Our group primarily worked with two datasets within our database. One dataset we are worked with involved property value and tax information for Pierce County in Washington state. This dataset contained the following tables, appraisal account, improvement, improvement built as, improvement details, land attribute, sale, seg merge, tax account, and tax information. The second dataset we worked with is crime information for Pierce County in Washington state. This dataset contained the following information: CaseNo (A unique ID given to each crime case), City, District, LocCode (General description of the location of the crime), NAT\_Name (Name of the Neighborhood Action Team (NAT) area where the crime occurred. A "None" value indicates that the crime did not occurred in a NAT area), OBJECTID (Internal feature number), OccurredOn (Date of occurrence), Public\_Nam (General description of the type of crime), XCoord (X-coordinates of location), YCoord (Y-coordinates of location), latitude, and longitude. Additionally, we included another dataset called Address Point that contained ZIP code information for Pierce County, Washington.

**Literature Review**

The following are summaries of various articles that pertained to our project, helped guide our process, and better informed our analysis of property value and crime in Pierce County, Washington.

*The Impact of Crime on Residential Property Value – On the Example of Szczecin (Foryś Iwona, & Putek-Szeląg Ewa., 2017):*

The motivation behind this article is to determine if there is a relationship crime rate, or rather the sense of security for a given area surrounding Szczecin, Poland, and housing prices. The authors indicate that this analysis of crime rates on housing prices also incorporates elements of criminology and social sciences in addition to their statistical methods. Their hope was to gain a more comprehensive understanding if there is any relationship between crime rates and housing prices. The authors first analyzed crime data provided by the Regional Police Headquarters in Szczecin. This information allowed the authors to understand the population statistics given a region and further filter out the types of crime that occurs most in those areas. Moreover, the authors looked into the property market of Szczecin to determine which regions in Szczecin, as defined by the Regional Police Headquarters in Szczecin data. As a result, the authors were able to segment the descriptive statistics for housing prices, given the police station districts of Szczecin.

From this exploratory analysis, the authors were able to perform a pattern analysis to further investigate crime rates in a given police station district and compare those findings against the change in housing prices in Szczecin. As a result, the authors determined that areas with low-priced apartments connected with their location (transaction price) coincide with areas of increased crime of a non-pecuniary nature (Foryś Iwona, 2017). Additionally, the authors identified that a rise in housing prices is associated with subsequent increase in property crime (e.g. dwelling theft), which is mostly the result of the increased number of new households with higher incomes (the effect of gentrification) (Foryś Iwona, 2017). Overall, the authors concluded that an increase in housing prices for a given location corresponds with a decrease in the number crime, but simultaneously is susceptible to a higher number of property crime.

*Determining the Impact of Residential Neighborhood Crime on Housing Investment Using Logistic Regression (Olajide, S., 2016):*

The author begins the paper by explaining that there are various findings of how exactly crime affects the property values of a residential area. The paper also explains that different types of levels of residential crimes, such as burglary, street crime, vandalism, can have various effects on property values. Therefore, the author’s purpose of the article is to predictively determine the impact of residential crime and its sub-variables (i.e. type of crime) on the residential property values. The results can be useful for local governments as a guide on which crimes to focus on.

The author used logistic regression analysis to predict the degree of impact of the various forms of residential neighborhood crime on residential property values. The data used for the model comes from multiple surveys collected from residential neighborhoods within Southwestern Nigeria. Crime was divided into multiple independent variables: street crime, vandalism, robbery, and violent crime. All of the predictors showed a direct influence on property values; however, the biggest contributor that negatively affected residential values was violent crime with an odd ratio of 71.1252. Furthermore, the author defends the statistical significance of the logit model by discussing the validity of its p-value, R-squared, and the ROC curve. The author moves on to explain that violent crimes have the highest effect on property values because such crimes attract more fear in residential communities. The author concludes that the findings of the logit model are in line with other published papers; that is, the model supports the hypothesis that residential neighborhood crime negatively influences the residential property values. This negative relationship can deter housing investments, which in turn causes residential neighborhood decline and a reduction in government revenue through property tax that can be used to fight residential crimes.

*The Impact on Community Safety on House Ranking (Yao, Z., & Fu, Y., 2017):*

One factor that often in consideration when purchasing a home is the degree of community safety where that house is located. Therefore, there is a demand for a tool to compare the house values which taking community safety degrees into account.

All the data of houses and crimes are collected from Denver Open Data Catalog. For the housing dataset, authors decided to only choose single-family detached home which is the major type in US. They selected 3000 houses evenly spread in a major residential region of north Denver which include North Park Hill, South Park Hill, Hale, Montclair, and Hilltop. All the appraised valued was in 2015. For crime data, the authors collected the residential forcible burglaries happened in Denver in five years before the appraisal implemented (2009-2014). For the house profiles, it was the neighborhood data from demographic data of 2010 and 2000 the US census.

This paper focuses on solving two research challenges relate to analyzing the impact of community safety degree on house value ranking. Challenge 1: what crime analysis can be done to generate an in-depth understanding of community safety. In other words, how can we assess the community safety degrees of an area by analyzing historical crime data? Challenge 2: How to systematically model the impacts of community safety on house values without the effects of other aspects, such as neighborhood income level and rating of a nearby school. This means building a model which can separately evaluate the impact of community safety degree on house value.

Yao, Z., & Fu, Y.(2017) presented a systematically study on house ranking. For the first challenge, they came up with the extract of community crimes evidence in crime severity and crime temporal correlation. To recognize which crime account to which house they calculated the distance between the house and the location where the crime committed. The crime will be counted to a house if it happens within a specific range from the house. To build the model, they adopted RankLib as baseline algorithm implementation.

**Data Preprocessing**

Null Values

The null value ratio for the columns that we chose to integrate is extremely low, as shown in the data dictionary in the Appendix. However, we took further steps to ensure we eliminated any possible null values in some of the more critical columns that we ended up choosing. For example, any null values in the AppraisalAccountType column are eliminated by filtering for only rows having ‘Residential’ values with our WHERE clause. The same principle was applied to the PropertyType and AccountType columns. Furthermore, we chose to focus our research only on specific zip codes due to the availability of our crime data. This naturally took out any null values from our ZipCode column because we used a WHERE clause for a range of given zip codes. The full SQL query can be found in the Appendix.

Data Integration

The first step to integrate the data was to see which tables from the original dataset will actually be useful for our research. We looked at each table and picked out the columns that would be most relevant to our problem statement. The table below summarizes all the tables that we integrated, along with the column names that we took from each of them.

*Table 1.1 Tables integrated and their respective columns*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Appraisal Account** | **Improvement** | **Improvement Builtas** | **Tax Account** | **Address Point** | **Crime** | **Sale** |
| ParcelNumber | BuildingID | PhysicalAge | AccountType | ZipCode | ObjectID | SalePrice |
| AppraisalAccountType | PropertyType | YearBuilt | TaxableValuePriorYear |  | OccurredOn | SaleDate |
| Buildings | SquareFeet | YearRemodeled | TaxableValueCurrentYear |  | Public\_NAM |  |
| LandGrossSquareFeet | PercentComplete |  |  |  | Latitude |  |
| LandNetSquareFeet | Condition |  |  |  | Longitude |  |
| AppraisalDate | Quality |  |  |  |  |  |
| Latitude |  |  |  |  |  |  |

Since ParcelNumber was the unique identification of each property in the majority of the tables, we used it as a key to join each all table together to create the new Property Attributes Tables. However, CrimeData does not have a ParcelNumber; in fact, that dataset has no property identification number. It has ObjectID, the latitude and longitude to identify the crime and where it occurred. To integrate Crime data into Property data for evaluating community safety features of houses, our methodology is considering the distance between the location where crimes occurred and a specific property. If a crime occurred within 1-mile radius area from the property, this crime will be count as crime attribute of the property. The advantage of this method is the number of different types of crime that occurred around a given property can reflect more accurate the public safety than crime rate of a wider area such as city or zip code. However, this method requires too many calculations. We have around 20,000 rows of properties after filtering the data and 27,000 rows of crime, this would lead to a half-billion calculations to obtain the entire data set of properties and crime in Pierce County. Due to the constraints of time and resources, we reduced our crime data to a 10% sample of the crime dataset.

The purpose of this project is to clarify the effect of crime on property value in Pierce County. We used the sale price to represent the current house value. Therefore, we use the most current sale price whose sale date is from 01/01/2018 until the present day. In addition, we only selected the most recent sale price in case the property has more than 1 sale price within this period.

The original crime data divides crime into 14 categories. We decided to regroup these categories into 5 new categories which crime type has similar attributes: Personal Crime, Property Crime, Drug Crime, Homicide, and Other crime. The following table illustrates how we recategorized the original 14 crime categories.

*Table 1.2 New crime categories*

|  |  |
| --- | --- |
| **Original Name** | **Category** |
| Arson - Non-residential | Property |
| Arson - Residential | Property |
| Assault - Aggravated | Personal |
| Assault - Simple | Personal |
| Burglary - Non-residential | Property |
| Burglary - Residential | Property |
| Criminal Traffic | Other |
| Drug Possession (Methamphetamine) | Drug |
| Drug Possession (Other) | Drug |
| Drug Sale/Manufacture (Methamphetamine) | Drug |
| Drug Sale/Manufacture (Other) | Drug |
| Fraud or Forgery | Property |
| Homicide | Homicide |
| Intimidation | Personal |
| Liquor Law Violations | Other |
| Motor Vehicle Theft | Property |
| Possession of Stolen Property | Property |
| Robbery - Business | Property |
| Robbery - Other | Property |
| Robbery - Residential | Property |
| Robbery - Street | Property |
| Telephone Harassment | Personal |
| Theft - Gas Station Runout | Property |
| Theft - Mail | Property |
| Theft - Other | Property |
| Theft - Vehicle Prowl | Property |
| Theft -Shoplifing | Property |
| Trafficking in Stolen Property | Property |
| Vandalism - Non-residential | Personal |
| Vandalism - Residential | Personal |
| Warrant Arrests | Other |

The final dataset includes 13 house attributes and 5 crime attributes, which is present in table below

*Table 1.3 Final dataset attributes*

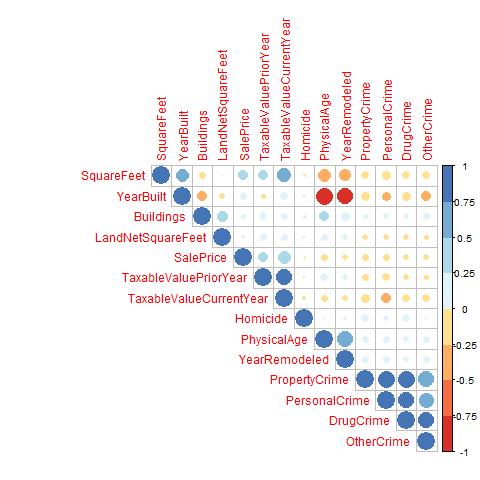
|  |  |
| --- | --- |
| **Property Attributes** | **Crime Attributes** |
| ParcelNumber | DrugCrime |
| Buildings | Homicide |
| LandNetSquareFeet | OtherCrime |
| SquareFeet | PersonalCrime |
| Condition | Other Crime |
| Quality |  |
| PhysicalAge |  |
| YearBuilt |  |
| YearRemodeled |  |
| TaxableValuePriorYear |  |
| TaxableValueCurrentYear |  |
| Sale Price |  |
| Sale Date |  |

Correlation Analysis

The results show that Condition and Quality have a high correlation coefficient. Among the House’s attribute only PhysicalAge, YearBuilt and YearRemodeled are highly correlated with each other. Another interesting finding is that type of crimes has a high correlation coefficient among themselves. SalePrice has a negative correlation with all type of crimes even the value of coefficients are not too large.

Correlation Matrix

*Figure 1.1 Correlation matrix for the variables used in our prediction models*



**Data Mining Models and Evaluations**

We began our analysis using the two datasets, one with crime attributes and one without crime attributes. With these two full datasets, we proceeded to use the following data mining models, decision tree, neural network, and random forest, in order to explore whether or not crime does have an effect on property value in Pierce County, Washington. Likewise, we implemented a clustering approach to both datasets in order to further investigate and compare the possible effects of crime on property value in Pierce County, Washington.

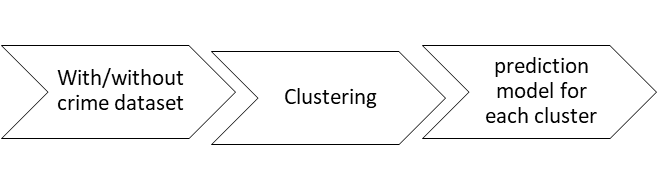
Clustering

One might expect that the effect of different type of crime varies by house attributes and local areas since the severity of crimes are different. Inspection of data cannot prove this hypothesis. Thus, we considered segmenting the data set into different groups which share similar characteristics in order to assess the different effects of crime on different groups of property.

To evaluate this effect, we implemented the clustering model with two types of datasets, one with and one without crime attributes. Then we used the clustered data and fed the prediction models with each cluster separately. We applied clustering model with 3 clusters and EM clustering method. The summary statistics of each cluster are included in the Figures & Tables section.

The following graphic illustrates our process for modeling the dataset after clustering.

*Figure 2.1 Graphic representation of the analytical processing of the dataset*



Decision Tree

The first method we used to build prediction model is decision tree model. We decided to use decision tree because this is a popular machine learning tool which is easy to interpret the result, handle categorical and numeric variables. Decision tree does not require scaled data and is also able to capture linear and non-linear relationship. In this study, we trained the models with all dataset and then separate models for each cluster. Both kind of model will be compared between with and without crime attributes. To obtain the best model, we apply two split method binary and combined binary and complete. The tree is split by choosing the best split set from a set of possible splits base on Bayesian Dirichlet Equivalent split score.

The result from model with the full dataset proved that number of occurred crime near by the house has statistically significant effects on houses value. Our assumption is more consolidated since the result from each clusters’ model still return similar results. In cluster model, crime attribute present on the first 2 level of the tree. Moreover, the effects of crimes on house’s value are varied in different clusters.

Neural Network

The second method we used to build a prediction model is neural network. A neural network is a set of connected input and output units where each connection has a weight associated with it. One of the advantages of this method is that the neural network works with patterns that have not been seen in the training data. Furthermore, multiple hidden layers can be used in a neural network. For example, a two-layer neural network uses the outputs of the first layer as the input of the second layer. The number of nodes within each network can also be adjusted to get the most statistically reliable model; however, as the number of nodes increase, the processing time also increase, so we need to be careful about choosing an optimum amount of nodes that results in quick processing times while it does not compromise the statistical significance of the model. It is important to note that the full dataset neural network did not produce better RMSE and MAE values when compared to the clustered dataset neural networks.

Random Forest

The third method used to build a prediction model is random forest. Random forest takes the building block of a decision tree and constructs many uncorrelated decision tree iterations in order to build a predictive model. Furthermore, the random forest model is designed to handle large volumes of data and correct the overfitting habit of decision trees. Similar to the decision tree model, we trained the random forest model for each cluster with all house and crime attributes. Finally, we used the random forest to analyze the importance of the house and crime variables in relation to the value of a home, in particular, the home’s sale price.

When comparing the full dataset against the clustered dataset, similar results are reported as mentioned with the decision tree model where the dataset with crime attributes are statistically significant on the sales price of a home.

Model Evaluation

We used Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) in order to evaluate and compare the performance of each model on a standardized level. The following are tables of the RMSE and MAE values from each model:

*Table 2.1 Decision Tree RMSE & MAE results*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Decision Tree** | | **With Crime** | | | | **Without Crime** | | | |
| Full | Cluster 1 | Cluster 2 | Cluster 3 | Full | Cluster 1 | Cluster 2 | Cluster 3 |
| RMSE | Binary Split | 802,791 | 760,370 | 151,577 | 1,090,287 | 1,191,100 | 822,770 | 103,862 | 2,283,599 |
| MAE |  | 140,862 | 139,524 | 87,607 | 246,482 | 157,959 | 139,941 | 61,992 | 593,919 |
| RMSE | Combined Split | 808,843 | 716,590 | 153,184 | 1,313,918 | 1,217,505 | 844,002 | 102,695 | 2,771,435 |
| MAE |  | 146,382 | 155,479 | 88,808 | 278,539 | 177,539 | 158,148 | 61,560 | 788,747 |

*Table 2.2 Neural Network RMSE & MAE results*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Neural Network** | | **With Crime** | | | | **Without Crime** | | | |
| Full | Cluster 1 | Cluster 2 | Cluster 3 | Full | Cluster 1 | Cluster 2 | Cluster 3 |
| RMSE | 10 Nodes | 942,523 | 1,009,950 | 382,911 | 1,310,449 | 446,063 | 1,013,235 | 307,147 | 1,524,744 |
| MAE |  | 159,348 | 232,421 | 334,709 | 194,418 | 427,282 | 174,120 | 283,337 | 342,764 |
| RMSE | 10 and 15 Nodes (hidden layer) | 942,523 | 1,009,950 | 382,939 | 1,086,080 | 446,550 | 1,010,770 | 307,165 | 1,641,107 |
| MAE |  | 140,083 | 201,409 | 334,562 | 162,899 | 429,833 | 174,433 | 283,255 | 348,074 |

*Table 2.3 Random Forest RMSE & MAE results*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Random Forest** | **With Crime** | | | | **Without Crime** | | | |
| Full | Cluster 1 | Cluster 2 | Cluster 3 | Full | Cluster 1 | Cluster 2 | Cluster 3 |
| RMSE | 383,215 | 935,123 | 85,499 | 1,039,295 | 707,119 | 872,753 | 88,143 | 1,298,972 |
| MAE | 75,139 | 124,167 | 51,584 | 130,690 | 102,167 | 126,689 | 47,404 | 203,945 |
| % Var Explained | 68.64 | 21.45 | 78.41 | 67.41 | 49.92 | 16.52 | 57.88 | 55.74 |

At first glance, it appears that there is a small margin of difference between using the dataset with the crime attributes and the dataset without the crime attributes. However, between all three models, it appears that Random Forest provided the smallest RMSE and MAE values against Decision Tree and Neural Network. Further, it appears that Cluster 2, even when compared to the full dataset, in both cases of the data having crime attributes and not having crime attributes, provided the smallest RMSE and MAE overall.

**Discussion and Evaluations**

As discussed in the previous section, Random Forest gave us the most statistically significant results. In modeling the datasets with and without crime, we noticed that Cluster 2 reported the lowest RMSE and MAE values, even in comparison to the full dataset. Both models only had minor differences in the RMSE and MAE values, meaning that crime does have a significant impact on property values. Furthermore, attention should be paid to Cluster 3 because its RMSE and MAE values on the model With Crime are much lower than those of Without Crime. This is important because Cluster 3 had a lot more crime observations, as explained in previous sections, and the decrease in the RMSE and MAE values could be explained by the addition of the crime attributes to the With Crime model. In addition, the Percent Variable Explained is much higher in all clusters for the With Crime model than the Without Crime model.

The Appendix shows the information gain values for Cluster 2 of the Random Forest model with crime attributes. It can be seen that, out of all crime attributes, property crime had the largest information gain value. This result can help local government and law enforcement agencies to target such crimes and attempt to reduce their rates. The reduction in property crimes can in turn raise the property values and property taxes, which can then be used for investment back into the community. The information gain values chart also shows that most internal attributes to a house (quality, age, square feet, etc.) are much larger than the crime attributes. One relatively quick explanation is that usually home buyers initially consider the physical attributes of a house rather than the crime rates of the neighborhood. This can be demonstrated by the lack of crime information provided in most real estate platforms today, such as Redfin and Zillow. When looking at houses for sale, these platforms list all various physical house attributes and other external factors, such as walkability and school ratings, but crime rating are not represented in the listing of a house for sale. As a result, unless a buyer thinks of gathering more research on the neighborhood crime rates, most residential property buyers will quickly overlook factoring in crime rates into their decision to buy the house.

It is also important to talk about our predicted variable, Sale Price. Although sale price is the closest comparable variable to the market value of residential properties, it does not represent the true economic conditions. The sale price variable is a historic snapshot that could easily be discredited due to economic volatility in the past few months, especially in the housing market. The housing market has seen a significant halt in the past few months, so taking a year worth of historical sale data could potentially be inaccurate in determining the current sale price of residential properties. Consequently, a more reliable model could be produced by using the market value of a property as the predicted variable and re-running all models with the same attributes used in our research and analysis.

**Conclusion**

Property values and correct predicting their value is a popular research question. Although, there has been a lot of research done on this topic in other areas of the country, there have been few such examples for one of the biggest counties in Washington State, Pierce County. Through the various modeling methods that we implemented, we can firmly conclude that crime does have a significant effect on the property values in Pierce County.

So how are our results important to the Pierce County community? Local governments can use these findings in order to invest more resources into stopping the types of crimes that have a significantly negative impact on the property values. Furthermore, investors can use the results to make important and informed decisions on housing development locations that would maximize their future profits.

Although our research gives statistically significant results, it does come with limitations. For example, the difference between actual and perceived crime has been widely discussed in previous literature. Pierce County, and especially Tacoma area, has been widely looked down upon in terms of safe neighborhoods. However, with the boom in population and economy in the Sea-Tac area, Tacoma and its suburbs have been widely gentrified. Therefore, we suggest further research to distinguish how perceived crime rates affect the property values in Pierce County.

**References**

Foryś Iwona, & Putek-Szeląg Ewa. (2017). THE IMPACT OF CRIME ON RESIDENTIAL PROPERTY VALUE - ON THE EXAMPLE OF SZCZECIN. Real Estate Management and Valuation, 25(3), 51-61.

Hellman, D., & Naroff, J. (1979). The Impact of Crime on Urban Residential Property Values. Urban Studies, 16(1), 105-112.

Tita, G., Petras, T., & Greenbaum, R. (2006). Crime and Residential Choice: A Neighborhood Level Analysis of the Impact of Crime on Housing Prices. Journal of Quantitative Criminology, 22(4), 299-317.

Wentland, S., Waller, B., & Brastow, R. (2014). Estimating the Effect of Crime Risk on Property Values and Time on Market: Evidence from Megan's Law in Virginia. Real Estate Economics, 42(1), 223-251.

Naroff, J. L., Hellman, D., & Skinner, D. (1980). The Boston Experience: Estimated of the Impact of Crime on Property Values. Growth and Change. Retrieved July 14, 2019.

Olijade, S. E., & Lizam, M. (2016). Determining the Impact of Residential Neighborhood Crime on Housing Investment Using Logistic Regression. Path of Science, 2(12). Doi: 10.22178/pos. 17-13

Yao, Z., Fu, Y., Liu, B., & Xiong, H. (2016). The Impact of Community Safety on House Ranking. Proceedings of the 2016 SIAM International Conference on Data Mining. doi:10.1137/1.9781611974348.52

Lynch, A., Rasmussen, D. (2001). Measuring the Impact of Crime on House Prices. Applied Economics. 33, 1981-1989

**Figures and Tables**

*Table 3.1 Summary statistics for house and crime attributes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Overall** | **Min** | **Max** | **Mean** | **SD** |
| Buildings | 0 | 10 | 1.5 | 1.28373 |
| LandNetSquareFeet | 1500 | 4316796 | 31158 | 88788.37 |
| SquareFeet | 192 | 7230 | 1889 | 802.3615 |
| PhysicalAge | 0 | 119 | 21.88 | 15.86617 |
| TaxableValueCurrentYear | 0 | 1979600 | 358418 | 165613.3 |
| TaxableValuePriorYear | 0 | 1882900 | 286620 | 171529.3 |
| DrugCrime | 0 | 51 | 5.342 | 9.572653 |
| Homicide | 0 | 2 | 0.01917 | 0.1633227 |
| PropertyCrime | 0 | 459 | 38.85 | 55.35086 |
| PersonalCrime | 0 | 108 | 14.15 | 21.8497 |
| OtherCrime | 0 | 393 | 29 | 61.01812 |
| SalePrice | 500 | 21700000 | 484224 | 1106847 |

*Table 3.2 Summary statistics for cluster 1 with crime attributes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cluster 1** | **Min** | **Max** | **Mean** | **SD** |
| Buildings | 0 | 3 | 1 | 0.270195 |
| LandNetSquareFeet | 0 | 1282406 | 18921.03 | 52462 |
| SquareFeet | 0 | 6894 | 2353.699 | 679.0965 |
| PhysicalAge | 0 | 44 | 9.603129 | 7.565222 |
| TaxableValueCurrentYear | 0 | 1979600 | 373669.6 | 177036.9 |
| TaxableValuePriorYear | 0 | 1882900 | 264042.2 | 196002.9 |
| DrugCrime | 0 | 12 | 1.619496 | 1.968218 |
| Homicide | 0 | 0 | 0 | 0 |
| PropertyCrime | 0 | 84 | 16.02462 | 13.56211 |
| PersonalCrime | 0 | 30 | 5.535052 | 5.307078 |
| OtherCrime | 0 | 81 | 6.950319 | 8.346022 |
| SalePrice | 500 | 21700000 | 499132.1 | 905757.1 |

*Table 3.3 Summary statistics for cluster 2 with crime attributes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cluster 2** | **Min** | **Max** | **Mean** | **SD** |
| Buildings | 1 | 4 | 1.264653 | 0.5392625 |
| LandNetSquareFeet | 1500 | 871200 | 41066.87 | 78270.44 |
| SquareFeet | 192 | 7230 | 1704.541 | 673.2939 |
| PhysicalAge | 1 | 87 | 30.07559 | 10.68368 |
| TaxableValueCurrentYear | 0 | 1800900 | 356373.9 | 167416.8 |
| TaxableValuePriorYear | 6300 | 1661200 | 315437 | 156293 |
| DrugCrime | 0 | 9 | 1.406719 | 1.754265 |
| Homicide | 0 | 0 | 0 | 0 |
| PropertyCrime | 0 | 72 | 16.1317 | 13.1777 |
| PersonalCrime | 0 | 26 | 5.56862 | 5.148634 |
| OtherCrime | 0 | 57 | 6.974446 | 7.764558 |
| SalePrice | 500 | 2500000 | 385781.2 | 212411.6 |

*Table 3.4 Summary statistics for cluster 2 with crime attributes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cluster 3** | **Min** | **Max** | **Mean** | **SD** |
| Buildings | 1 | 10 | 2.668346 | 2.175665 |
| LandNetSquareFeet | 1885 | 4316796 | 38856.59 | 139313.3 |
| SquareFeet | 200 | 3850 | 1313.305 | 695.4345 |
| PhysicalAge | 0 | 119 | 32.20546 | 18.16045 |
| TaxableValueCurrentYear | 2842 | 695000 | 333532.9 | 135404.5 |
| TaxableValuePriorYear | 0 | 613900 | 285193.4 | 134956.5 |
| DrugCrime | 0 | 51 | 17.99576 | 13.23702 |
| Homicide | 0 | 2 | 0.08271474 | 0.3314157 |
| PropertyCrime | 0 | 459 | 114.3505 | 72.134 |
| PersonalCrime | 0 | 108 | 42.6474 | 30.1919 |
| OtherCrime | 0 | 393 | 102.4753 | 94.00959 |
| SalePrice | 500 | 21700000 | 602984.7 | 1920534 |

*Table 3.5 Summary statistics for cluster 1 without crime attributes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cluster 1** | **Min** | **Max** | **Mean** | **SD** |
| Buildings | 0 | 3 | 1.016065 | 0.270195 |
| LandNetSquareFeet | 1885 | 445619 | 13571.1 | 52462 |
| SquareFeet | 408 | 5024 | 2279.152 | 679.0965 |
| PhysicalAge | 0 | 34 | 9.623827 | 7.565222 |
| TaxableValueCurrentYear | 0 | 725400 | 338385.8 | 177036.9 |
| TaxableValuePriorYear | 0 | 660200 | 230937.5 | 135322.2 |
| SalePrice | 500 | 21700000 | 459135 | 905757.1 |

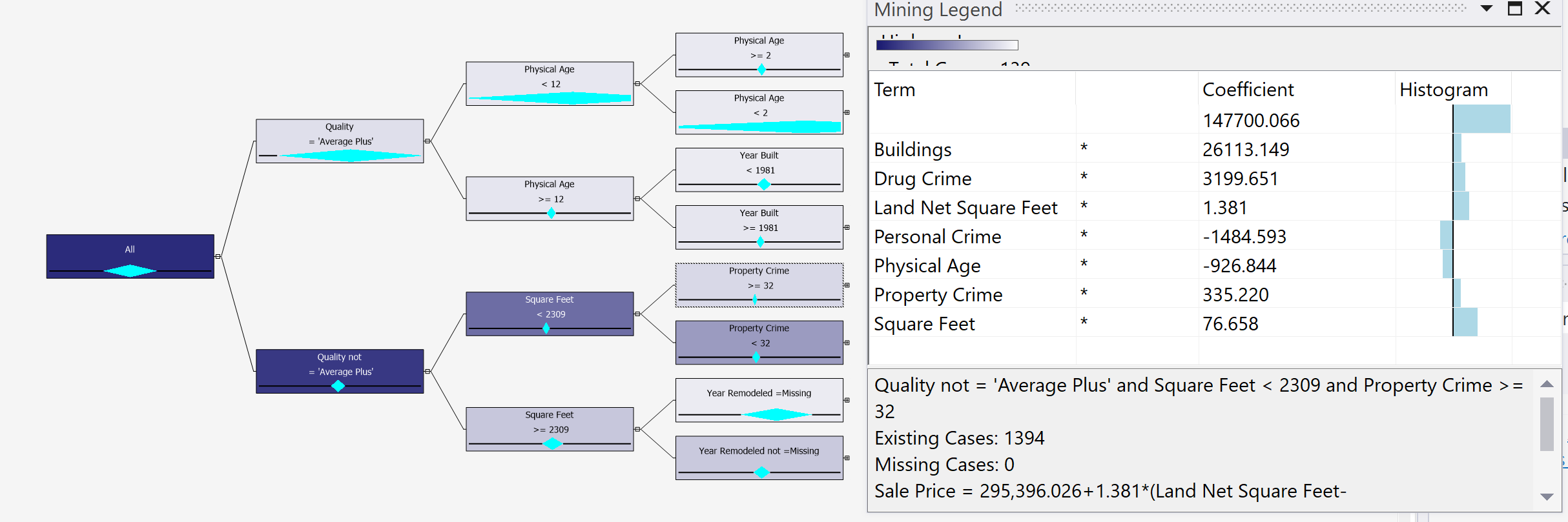
*Table 3.6 Summary statistics for cluster 2 without crime attributes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cluster 2** | **Min** | **Max** | **Mean** | **SD** |
| Buildings | 1 | 4 | 1.246728 | 0.5220631 |
| LandNetSquareFeet | 1500 | 665597 | 30611.14 | 55293.02 |
| SquareFeet | 192 | 3680 | 1543.026 | 515.9087 |
| PhysicalAge | 2 | 113 | 30.9991 | 11.77407 |
| TaxableValueCurrentYear | 0 | 643000 | 298456.7 | 84995.16 |
| TaxableValuePriorYear | 0 | 559200 | 262277.6 | 156293 |
| SalePrice | 500 | 2500000 | 321347 | 118164.3 |

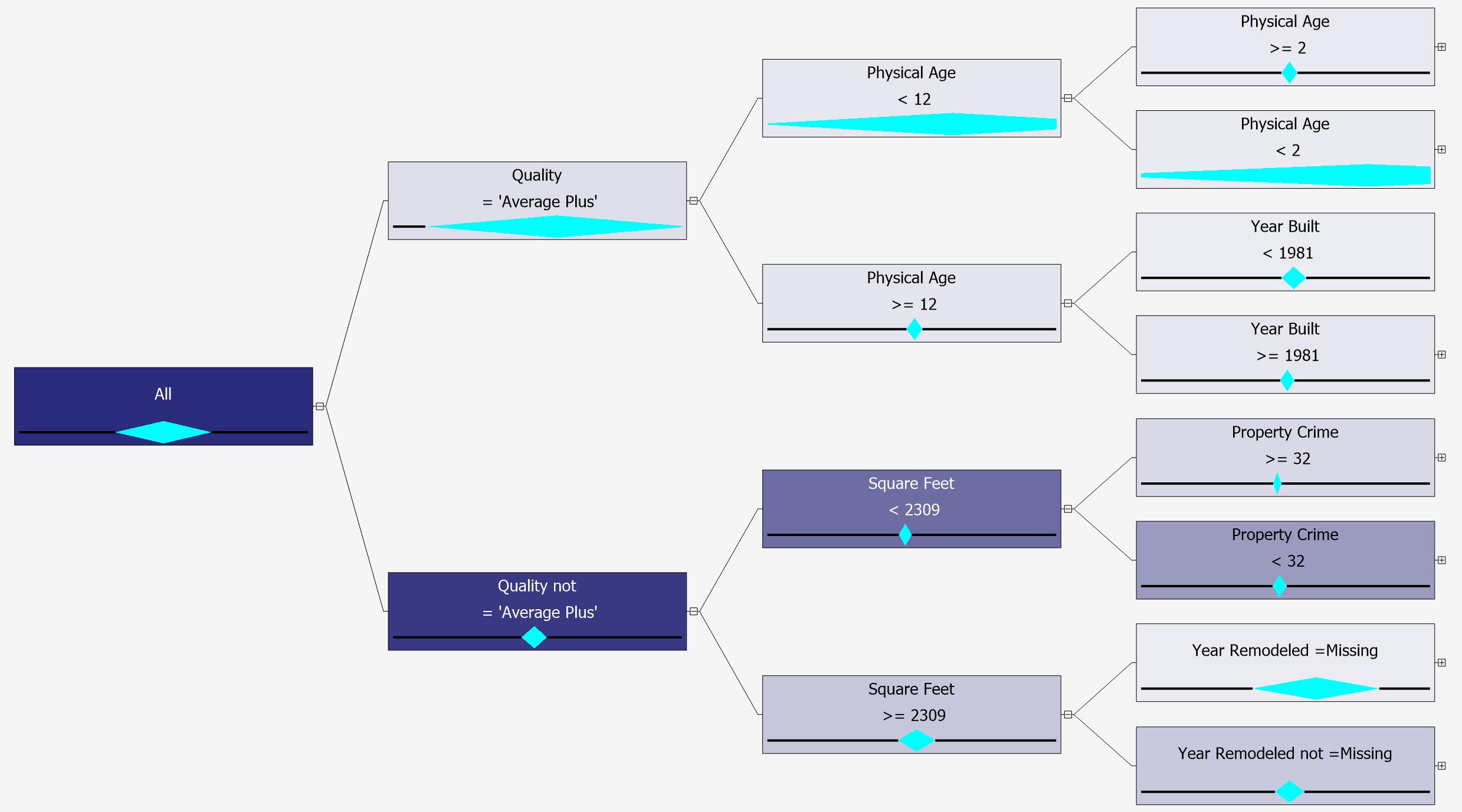
*Table 3.7 Summary statistics for cluster 2 without crime attributes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cluster 3** | **Min** | **Max** | **Mean** | **SD** |
| Buildings | 1 | 10 | 3.467387 | 2.22195 |
| LandNetSquareFeet | 3672 | 4316796 | 80307.47 | 187982.2 |
| SquareFeet | 200 | 7230 | 1719.135 | 1280.959 |
| PhysicalAge | 0 | 119 | 31.77808 | 18.6681 |
| TaxableValueCurrentYear | 23500 | 1979600 | 566623.6 | 255287.3 |
| TaxableValuePriorYear | 8000 | 1882900 | 501256.4 | 250848 |
| SalePrice | 1000 | 21700000 | 970184.7 | 2284014 |

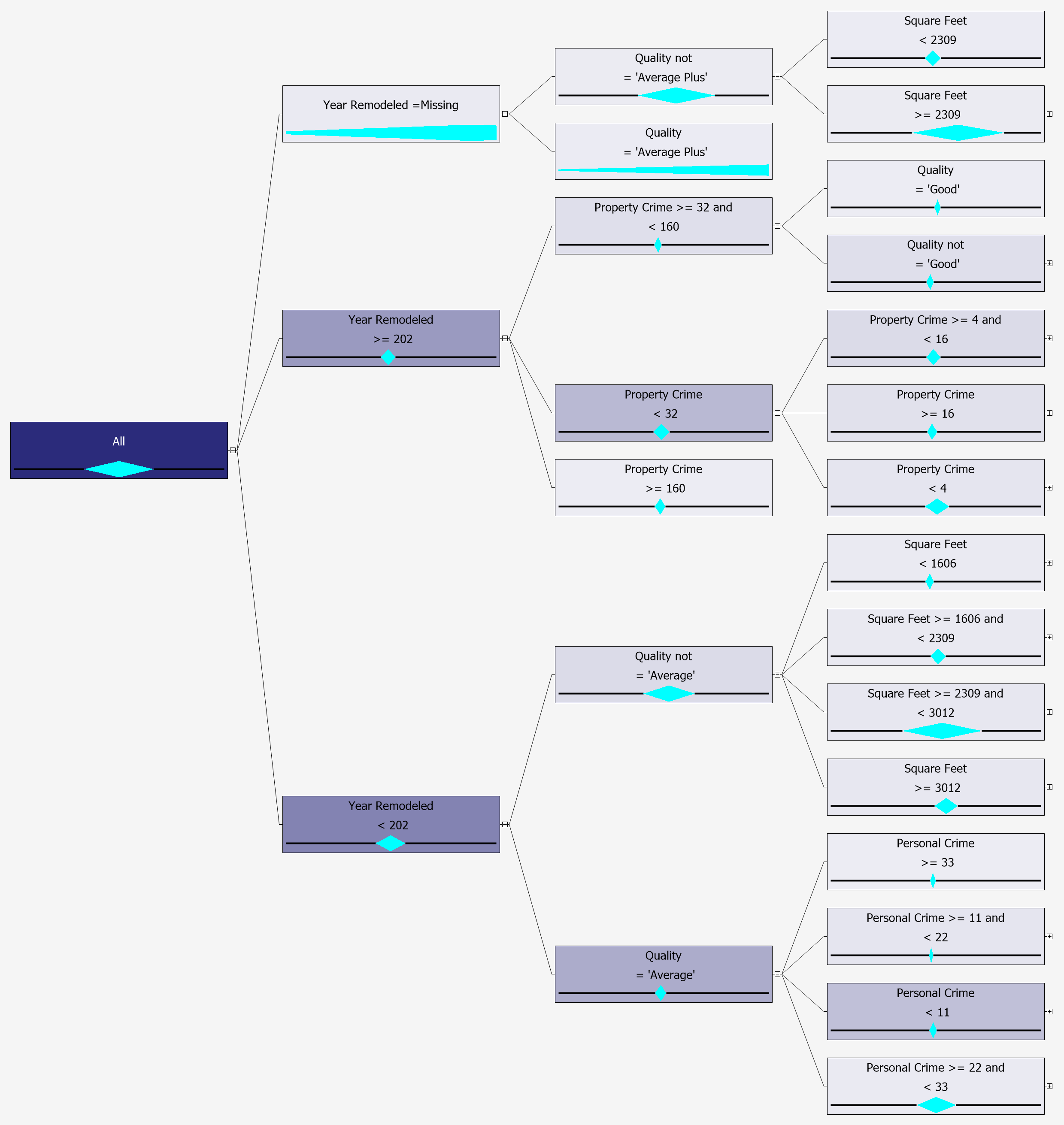
*Figure 3.1 Example from decision tree model with crime data binary split with mining legend*



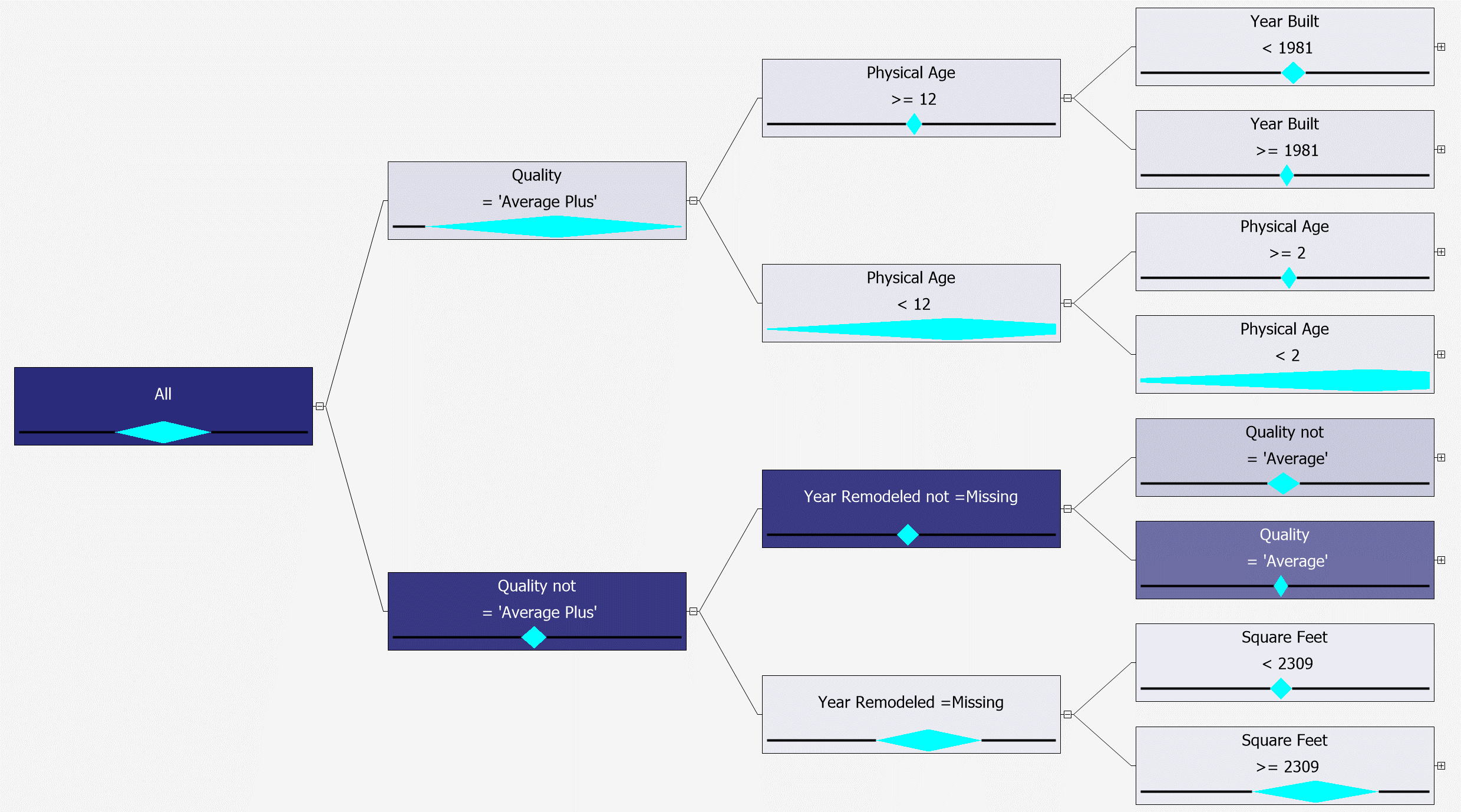
*Figure 3.2 Decision tree of full dataset binary split with crime attributes*



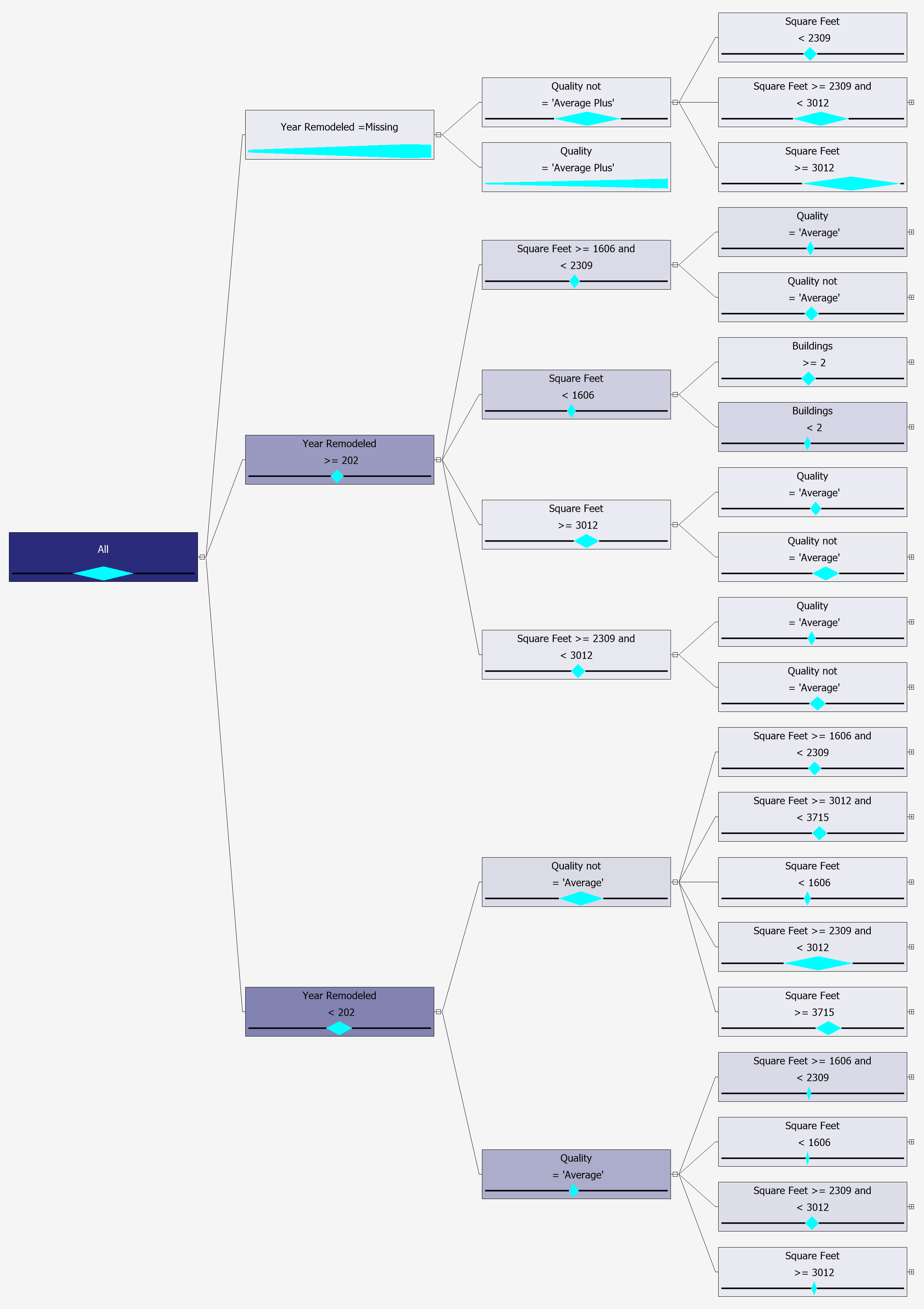
*Figure 3.3 Decision tree full dataset combined split with crime attributes*



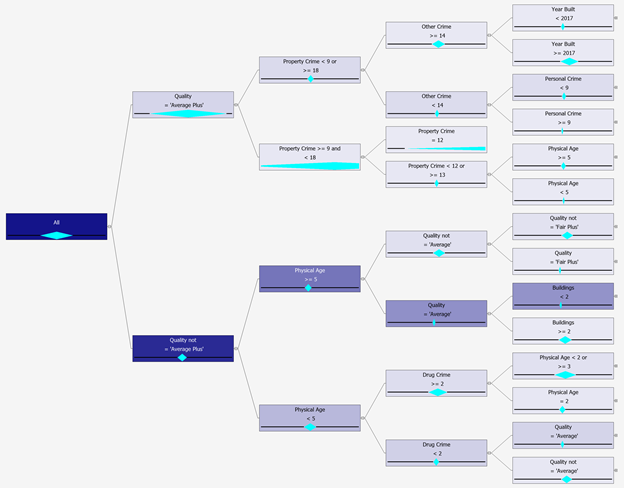
*Figure 3.4 Decision tree final dataset binary split without crime attributes*



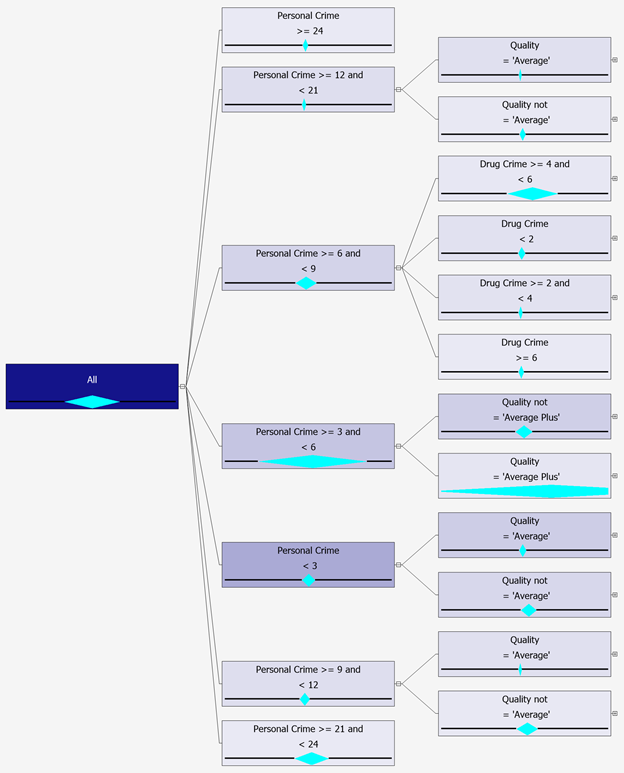
*Figure 3.5 Decision tree full dataset combined split without crime attributes*



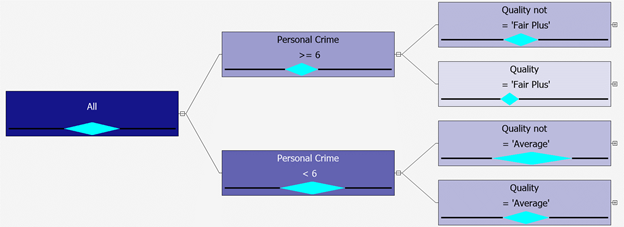
*Figure 3.6 Decision tree cluster 1 binary split with crime attributes*



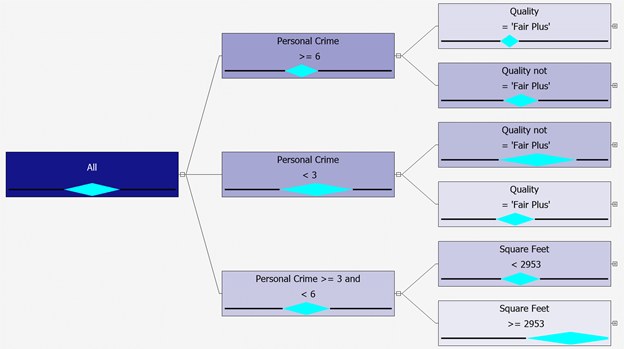
*Figure 3.7 Decision tree cluster 1 combine split with crime attributes*



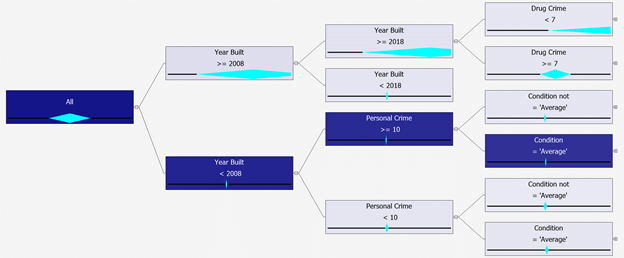
*Figure 3.8 Decision tree cluster 2 binary split with crime attributes*



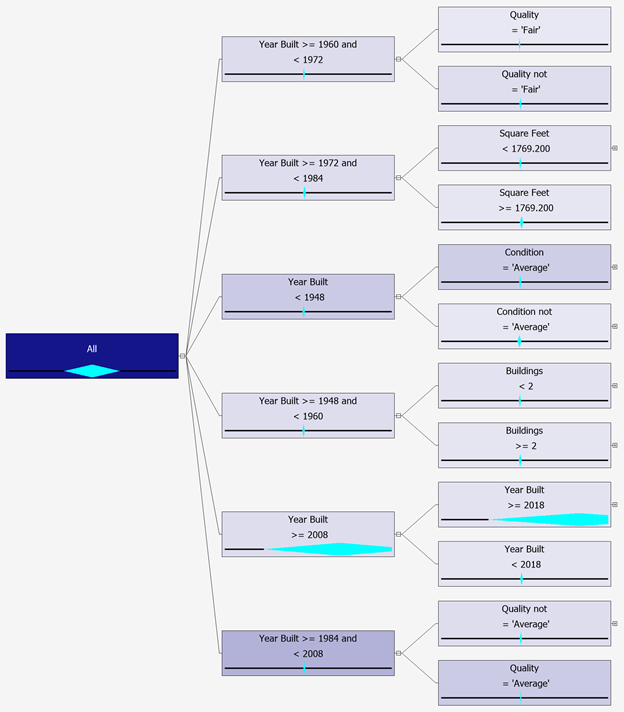
*Figure 3.9 Decision tree cluster 2 combine split with crime attributes*



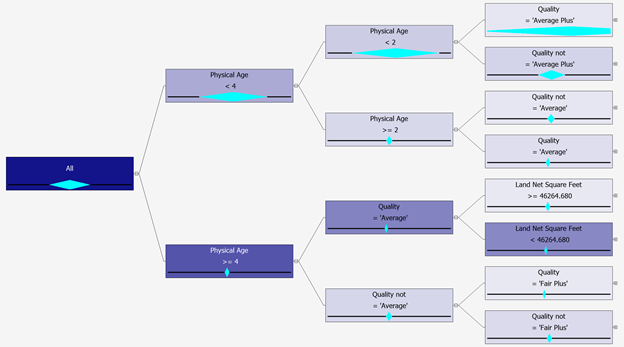
*Figure 3.10 Decision tree cluster 3 binary split with crime attributes*



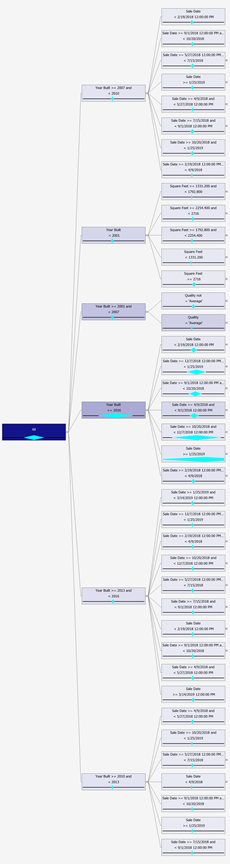
*Figure 3.11 Decision tree cluster 3 combine split with crime attributes*



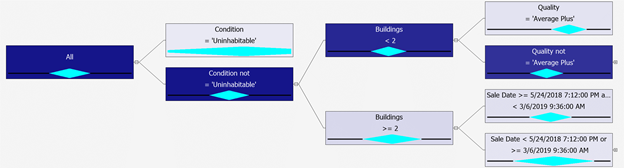
*Figure 3.12 Decision tree cluster 1 binary split without crime attributes*



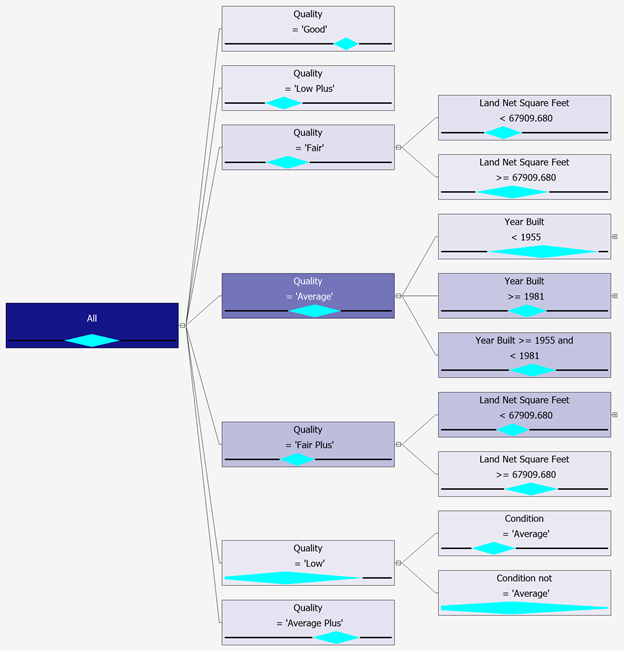
*Figure 3.13 Decision tree cluster 1 combine split without crime attributes*



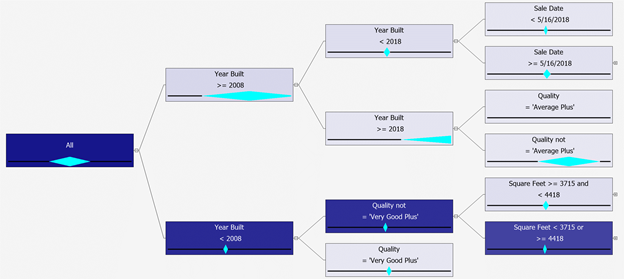
*Figure 3.14 Decision tree cluster 2 binary split without crime attributes*



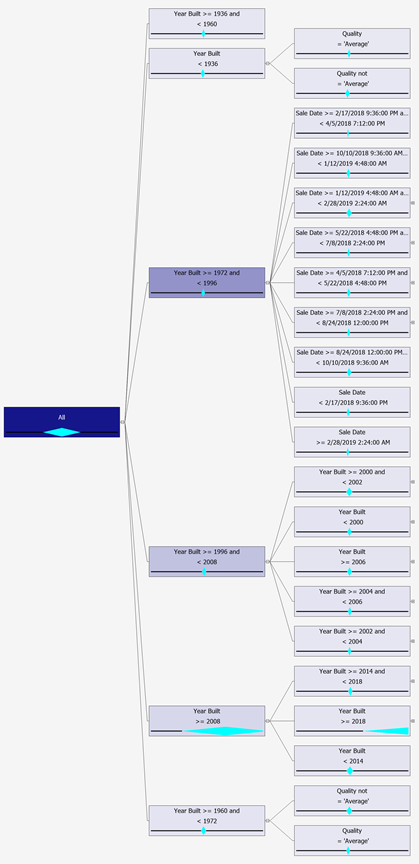
*Figure 3.15 Decision tree cluster 2 combine split without crime attributes*



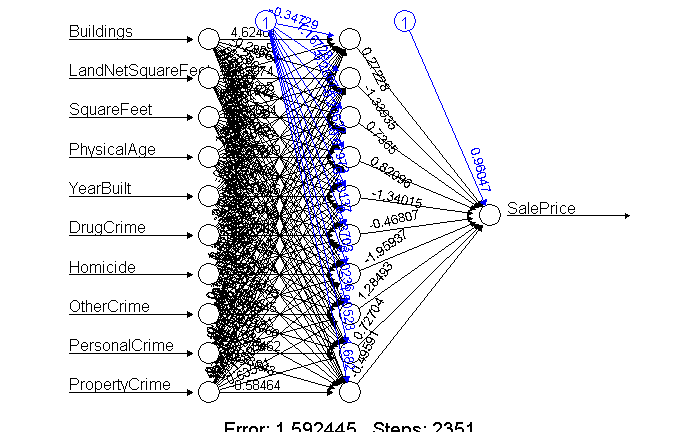
*Figure 3.16 Decision tree cluster 3 binary split without crime attributes*



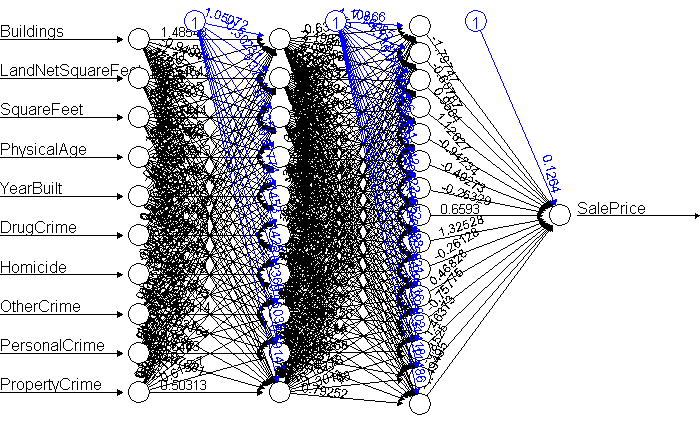
*Figure 3.17 Decision tree cluster 3 combine split without crime attributes*



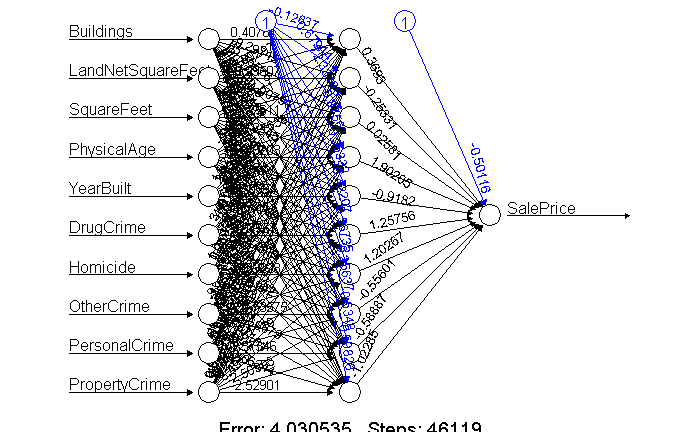
*Figure 4.1 Neural network with 1 hidden layer and 10 nodes for full dataset with crime attributes*

**

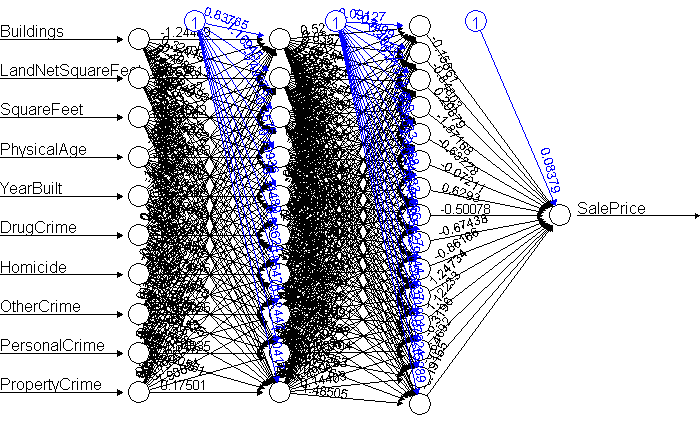
*Figure 4.2 Neural network with 2 hidden layers (10 and 15 nodes) for full dataset with crime attributes*

**

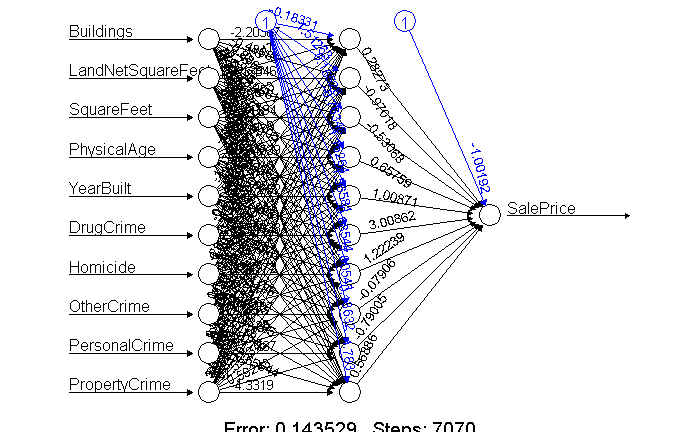
*Figure 4.3 Neural network 1 hidden layer (10 nodes) for cluster 1 with crime attributes*



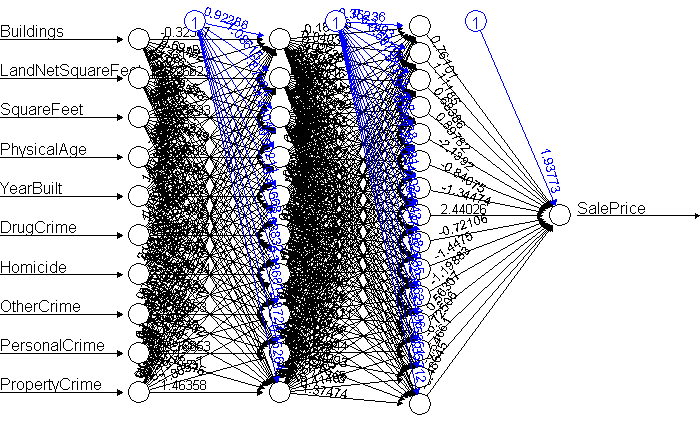
*Figure 4.4 Neural network with 2 hidden layers (10 and 15 nodes) for cluster 1 with crime attributes*



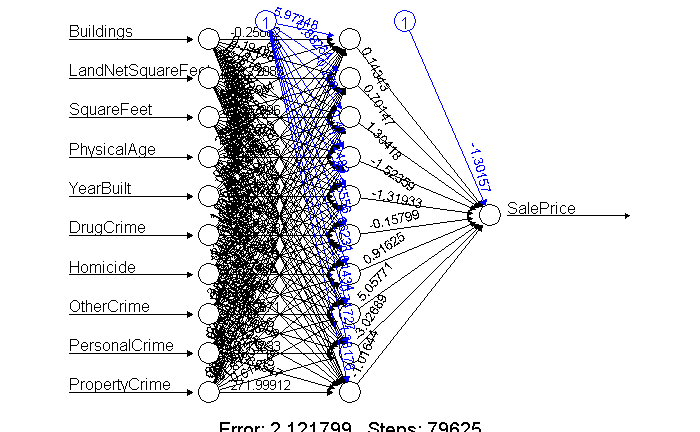
*Figure 4.5 Neural network with 1 hidden layer and 10 nodes for cluster 2 with crime attributes*



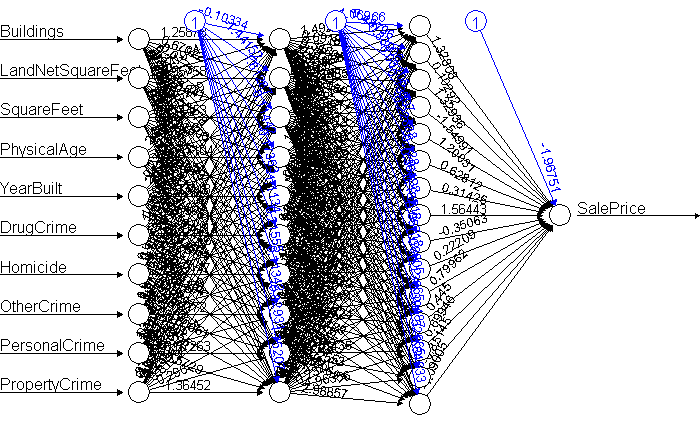
*Figure 4.6 Neural network with 2 hidden layers (10 and 15 nodes) for cluster 2 with crime attributes*



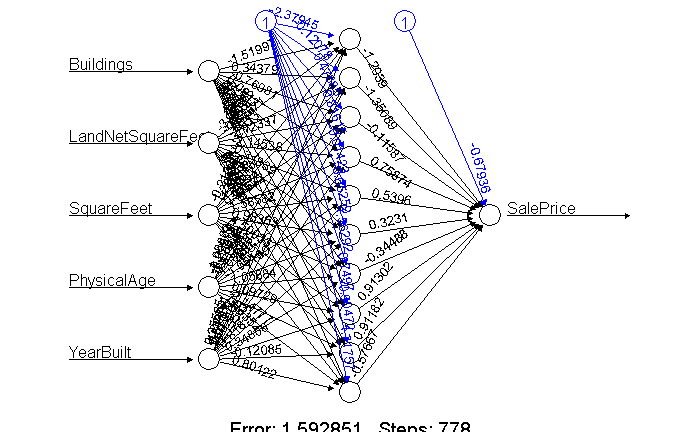
*Figure 4.7 Neural network with 1 hidden layer and 10 nodes for cluster 3 with crime attributes*



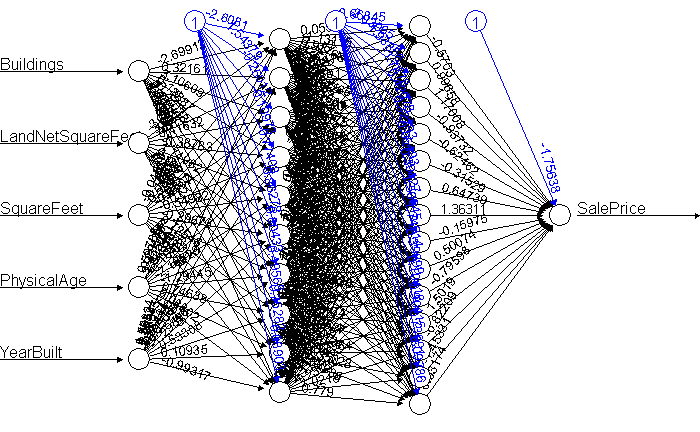
*Figure 4.8 Neural network with 2 hidden layers (10 and 15 nodes) for cluster 3 with crime attributes*



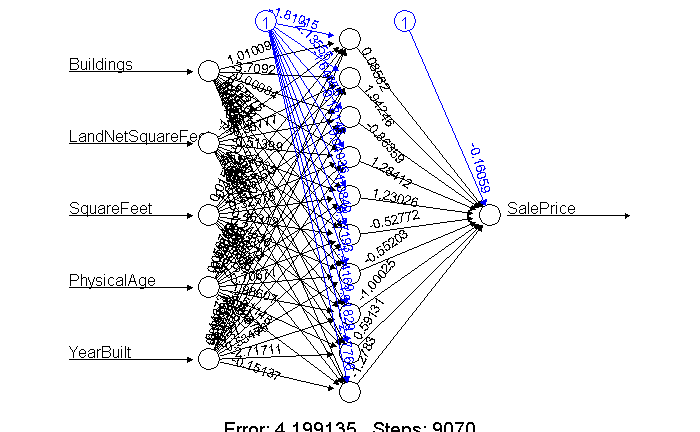
*Figure 4.9 Neural network with 1 hidden layer and 10 nodes for full dataset without crime attributes*

**

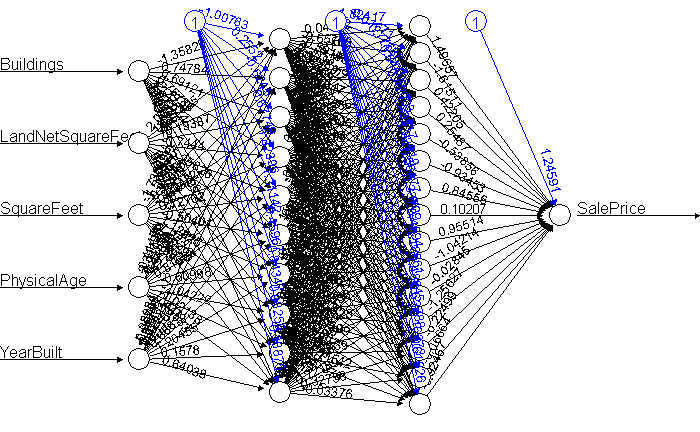
*Figure 4.10 Neural network with 2 hidden layers (10 and 15 nodes) for full dataset without crime attributes*

**

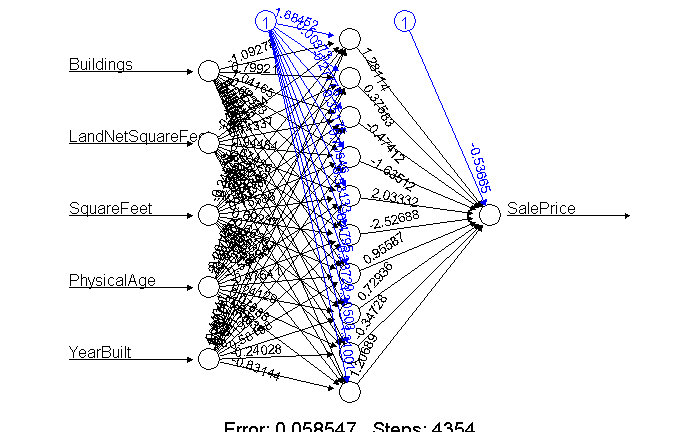
*Figure 4.11 Neural network with 1 hidden layer and 10 nodes for cluster 1 without crime attributes*

**

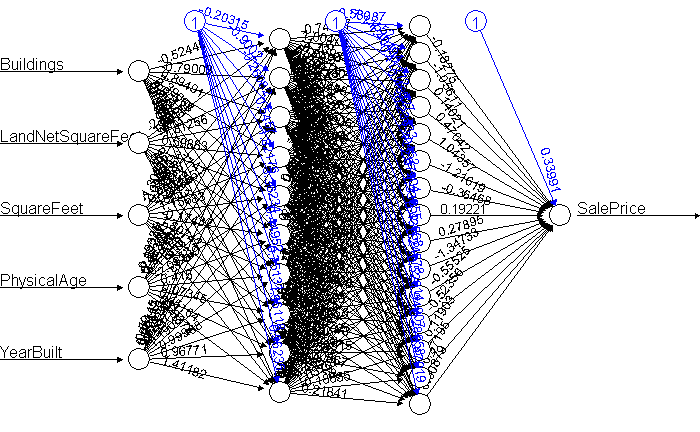
*Figure 4.12 Neural network with 2 hidden layers (10 and 15 nodes) for cluster 1 without crime attributes*

**

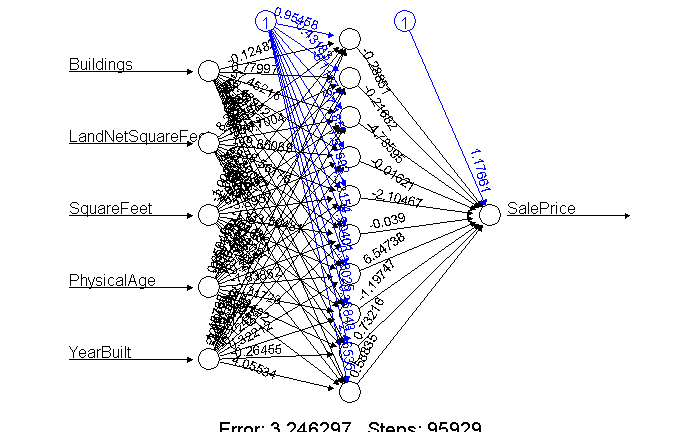
*Figure 4.13 Neural network with 1 hidden layer and 10 nodes for cluster 2 without crime attributes*

**

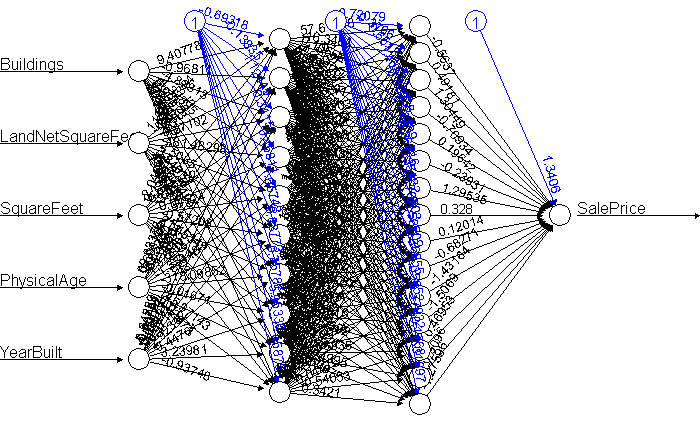
*Figure 4.14 Neural network with 2 hidden layers (10 and 15 nodes) for cluster 2 without crime attributes*

**

*Figure 4.15 Neural network with 1 hidden layer and 10 nodes for cluster 3 without crime attributes*

**

*Figure 4.16 Neural network with 2 hidden layers (10 and 15 nodes) for cluster 3 without crime attributes*

**

*Figure 5.1 Information gain for random forest model from full dataset with crime attributes*

A screenshot of a cell phone

Description automatically generated

*Table 4.1 Information gain values for random forest model from full dataset with crime attributes*

|  |  |
| --- | --- |
|  | IncNodePurity |
| Buildings | 5.340989e+12 |
| LandNetSquareFeet | 8.141879e+14 |
| SquareFeet | 4.385042e+14 |
| Condition | 1.392937e+12 |
| Quality | 7.220826e+14 |
| PhysicalAge | 3.400183e+14 |
| YearBuilt | 2.580768e+14 |
| YearRemodeled | 3.378380e+12 |
| TaxableValueCurrentYear | 7.426023e+14 |
| TaxableValuePriorYear | 9.038564e+14 |
| DrugCrime | 6.507751e+13 |
| Homicide | 1.123723e+11 |
| PropertyCrime | 6.115721e+14 |
| PersonalCrime | 5.879450e+14 |
| OtherCrime | 4.578828e+14 |

*Figure 5.2 Information gain for random forest model from cluster 1 with crime attributes*

**A screenshot of a cell phone

Description automatically generated**

*Table 4.2 Information gain values for random forest model from cluster 1 with crime attributes*

|  |  |
| --- | --- |
|  | IncNodePurity |
| Buildings | 8.696612e+12 |
| LandNetSquareFeet | 5.713952e+14 |
| SquareFeet | 3.785349e+14 |
| Condition | 3.494440e+11 |
| Quality | 2.573262e+14 |
| PhysicalAge | 7.813821e+13 |
| YearBuilt | 7.142558e+13 |
| YearRemodeled | 5.651236e+10 |
| TaxableValueCurrentYear | 7.040547e+14 |
| TaxableValuePriorYear | 3.805204e+14 |
| DrugCrime | 3.411225e+13 |
| Homicide | 0.000000e+00 |
| PropertyCrime | 1.445744e+14 |
| PersonalCrime | 8.904546e+13 |
| OtherCrime | 1.806005e+14 |

*Figure 5.3 Information gain for random forest model from cluster 2 with crime attributes*

*A screenshot of a cell phone

Description automatically generated*

*Table 4.3 Information gain values for random forest model from cluster 2 with crime attributes*

|  |  |
| --- | --- |
|  | IncNodePurity |
| Buildings | 1.520997e+12 |
| LandNetSquareFeet | 6.107343e+12 |
| SquareFeet | 1.297560e+13 |
| Condition | 1.731202e+12 |
| Quality | 1.985896e+13 |
| PhysicalAge | 5.248040e+12 |
| YearBuilt | 3.974364e+12 |
| YearRemodeled | 4.397659e+12 |
| TaxableValueCurrentYear | 8.186202e+13 |
| TaxableValuePriorYear | 5.635709e+13 |
| DrugCrime | 8.868254e+11 |
| Homicide | 0.000000e+00 |
| PropertyCrime | 3.606841e+12 |
| PersonalCrime | 2.346453e+12 |
| OtherCrime | 2.548712e+12 |

*Figure 5.4 Information gain for random forest model from cluster 3 with crime attributes*

*A screenshot of text

Description automatically generated*

*Table 4.4 Information gain values for random forest model from cluster 3 with crime attributes*

|  |  |
| --- | --- |
|  | IncNodePurity |
| Buildings | 2.001466e+13 |
| LandNetSquareFeet | 6.594608e+14 |
| SquareFeet | 2.781837e+14 |
| Condition | 4.589452e+12 |
| Quality | 5.145611e+14 |
| PhysicalAge | 7.521254e+14 |
| YearBuilt | 1.181167e+15 |
| YearRemodeled | 5.749126e+13 |
| TaxableValueCurrentYear | 1.156248e+15 |
| TaxableValuePriorYear | 4.862232e+14 |
| DrugCrime | 8.344999e+14 |
| Homicide | 1.332887e+12 |
| PropertyCrime | 1.582083e+15 |
| PersonalCrime | 8.247136e+14 |
| OtherCrime | 1.791367e+15 |

*Figure 5.5 Information gain for random forest model from full dataset without crime attributes*

A screenshot of a cell phone

Description automatically generated

*Table 4.5 Information gain values for random forest model from full dataset without crime attributes*

|  |  |
| --- | --- |
|  | IncNodePurity |
| Buildings | 7.157626e+12 |
| LandNetSquareFeet | 1.151621e+15 |
| SquareFeet | 6.565455e+14 |
| Condition | 1.607062e+12 |
| Quality | 6.113299e+14 |
| PhysicalAge | 3.066288e+14 |
| YearBuilt | 2.199684e+14 |
| YearRemodeled | 7.778978e+12 |
| TaxableValueCurrentYear | 1.080728e+15 |
| TaxableValuePriorYear | 1.105347e+15 |

*Figure 5.6 Information gain for random forest model from cluster 1 without crime attributes*

**A screenshot of a cell phone

Description automatically generated**

*Table 4.6 Information gain values for random forest model from cluster 1 without crime attributes*

|  |  |
| --- | --- |
|  | IncNodePurity |
| Buildings | 4.160050e+11 |
| LandNetSquareFeet | 6.992626e+14 |
| SquareFeet | 3.837899e+14 |
| Condition | 1.280691e+11 |
| Quality | 3.113038e+14 |
| PhysicalAge | 8.828755e+13 |
| YearBuilt | 9.994246e+13 |
| YearRemodeled | 2.429492e+09 |
| TaxableValueCurrentYear | 6.073323e+14 |
| TaxableValuePriorYear | 4.236776e+14 |

*Figure 5.7 Information gain for random forest model from cluster 2 without crime attributes*

*A screenshot of a cell phone

Description automatically generated*

*Table 4.7 Information gain values for random forest model from cluster 2 without crime attributes*

|  |  |
| --- | --- |
|  | IncNodePurity |
| Buildings | 8.055342e+11 |
| LandNetSquareFeet | 5.077312e+12 |
| SquareFeet | 5.835284e+12 |
| Condition | 8.570300e+11 |
| Quality | 3.692681e+12 |
| PhysicalAge | 2.747523e+12 |
| YearBuilt | 2.988945e+12 |
| YearRemodeled | 2.796651e+12 |
| TaxableValueCurrentYear | 2.392984e+13 |
| TaxableValuePriorYear | 1.708902e+13 |

*Figure 5.8 Information gain for random forest model from cluster 3 without crime attributes*

*A screenshot of a cell phone

Description automatically generated*

*Table 4.8 Information gain values for random forest model from cluster 3 without crime attributes*

|  |  |
| --- | --- |
|  | IncNodePurity |
| Buildings | 1.690680e+13 |
| LandNetSquareFeet | 1.704885e+15 |
| SquareFeet | 3.864085e+14 |
| Condition | 3.303242e+12 |
| Quality | 6.028949e+14 |
| PhysicalAge | 1.352002e+15 |
| YearBuilt | 8.647813e+14 |
| YearRemodeled | 2.757214e+13 |
| TaxableValueCurrentYear | 1.159912e+15 |
| TaxableValuePriorYear | 9.581543e+14 |

**Appendix**

Exploratory Data Analysis

> #eda of appraisal account table

> summary(appraisal)

ParcelNumber AppraisalAccountType

Min. :1.901e+07 Residential:286531

1st Qu.:6.153e+08 Commercial : 16252

Median :5.000e+09 Mobile Home: 12512

Mean :4.313e+09 Condominium: 5287

3rd Qu.:6.440e+09 Reference : 3155

Max. :9.900e+09 Industrial : 3133

(Other) : 4472

BusinessName ValueAreaID

SANS SOUCI MHP : 253 PI0: 71

LIPOMA FIRS NORTH PDD PH 2 : 213 PI1:40022

BOATHOUSE TACOMA YACHT CLUB BLDG ONLY: 206 PI2:67341

CRYSTAL POINTE CONDO : 206 PI3:58201

NORTH TAPPS ESTATES : 202 PI4:58902

(Other) : 40761 PI5:47707

NA's :289501 PI6:59098

LandEconomicArea Buildings GroupAccountNumber

070201 : 6403 Min. : 0.000 Min. : 2

120605 : 5867 1st Qu.: 1.000 1st Qu.: 3941

070101 : 5070 Median : 1.000 Median : 32600

090601 : 4709 Mean : 1.045 Mean : 8247579

060701 : 3985 3rd Qu.: 1.000 3rd Qu.: 64589

(Other):305298 Max. :110.000 Max. :830209752

NA's : 10 NA's :2171 NA's :296854

LandGrossAcres LandNetAcres LandGrossSquareFeet

Min. : 0.0 Min. : 0.000 Min. :0.000e+00

1st Qu.: 0.1 1st Qu.: 0.147 1st Qu.:6.020e+03

Median : 0.2 Median : 0.239 Median :1.029e+04

Mean : 9.5 Mean : 2.333 Mean :4.148e+05

3rd Qu.: 0.6 3rd Qu.: 0.530 3rd Qu.:2.469e+04

Max. :2242517.0 Max. :3033.400 Max. :9.768e+10

NA's :16966

LandNetSquareFeet LandGrossFrontFeet LandWidth

Min. : 0 Min. : 0.00 Min. : 0.000

1st Qu.: 6404 1st Qu.: 0.00 1st Qu.: 0.000

Median : 10405 Median : 0.00 Median : 0.000

Mean : 101615 Mean : 28.06 Mean : 5.438

3rd Qu.: 23087 3rd Qu.: 0.00 3rd Qu.: 0.000

Max. :132134904 Max. :144330.00 Max. :23232.000

NA's :16966 NA's :1642

LandDepth SubmergedAreaSqrFeet AppraisalDate

Min. : 0.000 Min. : 0 Min. :2001-09-12

1st Qu.: 0.000 1st Qu.: 0 1st Qu.:2015-04-02

Median : 0.000 Median : 0 Median :2017-03-20

Mean : 8.254 Mean : 1258 Mean :2016-10-09

3rd Qu.: 0.000 3rd Qu.: 0 3rd Qu.:2018-05-23

Max. :3200.000 Max. :10545254 Max. :3016-10-11

NA's :275920 NA's :1296

WaterfrontType ViewQuality UtilityElectric

WF Lake : 4682 View Lim : 5204 POWER AVAILABLE : 27556

WF River : 1128 View Lim -: 3997 POWER INSTALLED :300932

WF Salt : 5257 View Avg : 3966 POWER NO - COMMENT: 2854

WF Stream/Creek: 3207 View Good : 2541

NA's :317068 View Lim +: 2484

(Other) : 2280

NA's :310870

UtilitySewer UtilityWater

SEWER/SEPTIC AVAIL : 7247 WATER AVAILABLE: 19234

SEWER/SEPTIC INSTALLED:301889 WATER INSTALLED:299673

SEWER/SEPTIC NO : 21898 WATER NO : 12435

SEWER/SEPTIC NO PERC : 308

StreetType Latitude Longtitude

PAVED :307896 Min. :46.74 Min. :-122.8

STREET NO ROAD: 2527 1st Qu.:47.12 1st Qu.:-122.5

STREET UNPAVED: 20919 Median :47.18 Median :-122.4

Mean :47.18 Mean :-122.4

3rd Qu.:47.24 3rd Qu.:-122.3

Max. :47.40 Max. :-121.5

NA's :1414 NA's :1414

>

> #eda of improvement table

> summary(improvement)

ParcelNumber BuildingID PropertyType

Min. :1.912e+07 Min. : 0.000 Residential :450850

1st Qu.:5.194e+08 1st Qu.: 1.000 Out Building: 98626

Median :4.720e+09 Median : 1.000 Commercial : 57614

Mean :4.262e+09 Mean : 1.387 Mobile Home : 46200

3rd Qu.:6.490e+09 3rd Qu.: 1.000 Townhouse : 16718

Max. :9.900e+09 Max. :822.000 Duplex : 10498

(Other) : 7388

Neighborhood NeighborhoodExtension SquareFeet

070201 : 13158 0 :607822 Min. : 1

120605 : 11516 862 : 7330 1st Qu.: 1046

070101 : 11220 P2 : 6524 Median : 1540

090601 : 10358 PL1 : 6160 Mean : 2266

160702 : 9916 852 : 5392 3rd Qu.: 2130

(Other):631724 (Other): 54664 Max. :1078155

NA's : 2 NA's : 2

NetSquareFeet PercentComplete Condition

Min. : 0.0 Min. :0.0000 Average :642742

1st Qu.: 0.0 1st Qu.:1.0000 Fair : 29298

Median : 0.0 Median :1.0000 Poor : 8188

Mean : 843.2 Mean :0.9978 Very Poor : 3352

3rd Qu.: 0.0 3rd Qu.:1.0000 Extra Poor: 2300

Max. :1523580.0 Max. :1.0040 (Other) : 1372

NA's :6 NA's : 642

Quality PrimaryOccupancyCode

Average :329520 Min. : 26.0

Fair Plus :112090 1st Qu.: 100.0

Fair : 97712 Median : 100.0

Average Plus: 59246 Mean : 174.7

Low : 33408 3rd Qu.: 100.0

(Other) : 55286 Max. :9997.0

NA's : 632 NA's :4194

PrimaryOccupancyDescription MobileHomeSerialNumber

Single Family Residential :446730 0 : 1524

Detached Garage : 69286 FO : 10

Mobile or Manufactured Home: 46212 \* : 6

Storage - Material : 13990 0213 : 6

Townhouse/Condo : 10960 0340 : 6

(Other) : 96522 (Other): 44064

NA's : 4194 NA's :642278

MobileHomeTotalLength MobileHomeMake AtticFinishedSquareFeet

Min. : 0.0 FLTWD : 9068 Min. : 36

1st Qu.:48.0 MODUL : 5028 1st Qu.: 280

Median :56.0 SKY : 3414 Median : 362

Mean :55.6 LIBER : 3412 Mean : 395

3rd Qu.:64.0 REDMN : 2874 3rd Qu.: 472

Max. :84.0 (Other): 21696 Max. :3200

NA's :642368 NA's :642402 NA's :652014

BasementSquareFeet BasementFinishedSquareFeet CarportSquareFeet

Min. : 49 Min. : 1 Min. : 56

1st Qu.: 720 1st Qu.: 656 1st Qu.: 276

Median : 1008 Median : 970 Median : 391

Mean : 1068 Mean : 1160 Mean : 414

3rd Qu.: 1336 3rd Qu.: 1327 3rd Qu.: 510

Max. :38230 Max. :77597 Max. :1224

NA's :602104 NA's :634126 NA's :686492

BalconySquareFeet PorchSquareFeet AttachedGarageSquareFeet

Min. : 30.0 Min. : 0.0 Min. : 1

1st Qu.: 78.0 1st Qu.: 160.0 1st Qu.: 420

Median : 96.0 Median : 342.0 Median : 484

Mean : 267.4 Mean : 477.4 Mean : 532

3rd Qu.: 226.0 3rd Qu.: 621.0 3rd Qu.: 610

Max. :17609.0 Max. :13468.0 Max. :6896

NA's :685044 NA's :263164 NA's :371498

DetachedGarageSquareFeet Fireplaces BasementGarageDoor

Min. : 112.0 Min. :0.00 Min. :-502.0

1st Qu.: 420.0 1st Qu.:1.00 1st Qu.: 1.0

Median : 576.0 Median :1.00 Median : 1.0

Mean : 642.5 Mean :1.14 Mean : 1.5

3rd Qu.: 768.0 3rd Qu.:1.00 3rd Qu.: 2.0

Max. :4608.0 Max. :8.00 Max. : 760.0

NA's :650802 NA's :293072 NA's :671340

>

> #eda of improvement builtas table

> summary(improvement\_builtas)

ParcelNumber BuildingID BuiltAsNumber BuiltAsID

Min. :1.912e+07 Min. : 0.00 Min. :1.000 Min. : 1.00

1st Qu.:5.193e+08 1st Qu.: 1.00 1st Qu.:1.000 1st Qu.: 1.00

Median :4.715e+09 Median : 1.00 Median :1.000 Median : 8.00

Mean :4.251e+09 Mean : 1.39 Mean :1.009 Mean : 70.18

3rd Qu.:6.460e+09 3rd Qu.: 1.00 3rd Qu.:1.000 3rd Qu.: 68.00

Max. :9.900e+09 Max. :822.00 Max. :7.000 Max. :1499.00

BuiltAsDescription BuiltAsSquareFeet HVAC

1 Story :221900 Min. : 0 Min. : NA

2 Story :149918 1st Qu.: 1047 1st Qu.: NA

Detached Garage: 69418 Median : 1542 Median : NA

1 1/2 Story Fin: 33882 Mean : 2248 Mean :NaN

Double Wide : 32548 3rd Qu.: 2136 3rd Qu.: NA

Split Entry : 26514 Max. :1016109 Max. : NA

(Other) :159164 NA's :693344

HVACDescription Exterior

Forced Air :352818 Frame Siding :475420

None :111796 Frame Vinyl : 53918

Electric Baseboard : 92564 Hardboard Sheet : 27454

Heat Pump : 66428 Ribbed Aluminum : 11248

Warm and Cool Air Zone: 23026 Masonry Common Brick: 8074

(Other) : 46070 (Other) : 21344

NA's : 642 NA's : 95886

Interior Stories StoryHeight

Drywall :506608 Min. : 0.000 Min. : 0.000

Paneling: 21474 1st Qu.: 1.000 1st Qu.: 8.000

NA's :165262 Median : 1.000 Median : 8.000

Mean : 1.296 Mean : 8.412

3rd Qu.: 2.000 3rd Qu.: 8.000

Max. :135.000 Max. :5010.000

NA's :14 NA's :4

SprinklerSquareFeet RoofCover Bedrooms

Min. : 0.0 Composition Shingle:437354 Min. : 0.000

1st Qu.: 0.0 Shingle Comp : 35072 1st Qu.: 1.000

Median : 0.0 Concrete Tile : 12420 Median : 3.000

Mean : 470.3 Wood Shake : 11568 Mean : 2.384

3rd Qu.: 0.0 Metal Ribbed : 11142 3rd Qu.: 3.000

Max. :1029110.0 (Other) : 19098 Max. :21.000

NA's :166690 NA's :874

Bathrooms Units ClassCode

Min. : 0.000 Min. : 0.000 A : 1204

1st Qu.: 1.000 1st Qu.: 1.000 B : 1048

Median : 1.750 Median : 1.000 C : 14850

Mean : 1.526 Mean : 1.265 D : 48154

3rd Qu.: 2.500 3rd Qu.: 1.000 P : 17088

Max. :2075.000 Max. :2319.000 S : 5192

NA's :866 NA's :12 NA's:605808

ClassDescription YearBuilt YearRemodeled

Fireproof Steel : 1204 Min. : 0 Min. : 0

Masonry : 14850 1st Qu.:1960 1st Qu.: 0

Metal Frame : 5192 Median :1982 Median :1976

Pole : 17088 Mean :1976 Mean :1237

Reinforced Concrete: 1048 3rd Qu.:1999 3rd Qu.:1991

Wood Frame : 48154 Max. :2019 Max. :2019

NA's :605808 NA's :1686

AdjustedYearBuilt PhysicalAge BuiltAsLength BuiltAsWidth

Min. : 0 Min. : 0.00 Min. : 0.000 Min. : 0.00

1st Qu.:1981 1st Qu.: 19.00 1st Qu.: 0.000 1st Qu.: 0.00

Median :1991 Median : 28.00 Median : 0.000 Median : 0.00

Mean :1987 Mean : 29.17 Mean : 3.713 Mean : 1.54

3rd Qu.:2000 3rd Qu.: 38.00 3rd Qu.: 0.000 3rd Qu.: 0.00

Max. :2019 Max. :134.00 Max. :100.000 Max. :80.00

NA's :2 NA's :2 NA's :1706 NA's :1706

MobileHomeModel

ARDMORE : 1404

FLEETWOOD : 982

GREEN HILL: 966

SKYLINE : 876

BARRINGTON: 864

(Other) : 41114

NA's :647138

>

> #eda of improvement detail table

> summary(improvement\_detail)

ParcelNumber BuildingID DetailType

Min. :1.912e+07 Min. : 0.000 Fixture :762467

1st Qu.:2.206e+09 1st Qu.: 1.000 Appliance:464828

Median :5.001e+09 Median : 1.000 Porch :351133

Mean :4.457e+09 Mean : 1.066 Rough In :261717

3rd Qu.:6.490e+09 3rd Qu.: 1.000 Garage :186091

Max. :9.900e+09 Max. :819.000 Add On :134209

(Other) :126001

DetailDescription Units

Allowance :524127 Min. : -512

Laundry Facility:261164 1st Qu.: 1

Bath 3 Fixture :250950 Median : 1

PreFab/Stoves :121668 Mean : 325

Wood Deck :111742 3rd Qu.: 162

Attached :111254 Max. :5000000

(Other) :905541

>

> #eda of land attribute table

> summary(land\_attribute)

ParcelNumber Attribute

Min. :1.901e+07 R AMENITIES :118740

1st Qu.:4.192e+08 R FUNCTIONAL:106061

Median :4.002e+09 R UTILITIES : 83239

Mean :3.760e+09 R ECONOMIC : 39782

3rd Qu.:6.026e+09 R STREETS : 31812

Max. :9.900e+09 R WATERFRONT: 28028

(Other) :145581

AttributeDescription

PLATTED AMENITIES : 62481

PLATTED AMENITIES LIMITED: 35205

POWER AVAILABLE : 27556

SEWER/SEPTIC NO : 21898

STREET UNPAVED : 20931

WATER AVAILABLE : 19234

(Other) :365938

>

> #eda of sale table

> summary(sale)

ENT ParcelCount ParcelNumber

4256595 : 591 Min. : 1.000 Min. :1.901e+07

4246057 : 568 1st Qu.: 1.000 1st Qu.:2.745e+09

1054878 : 423 Median : 1.000 Median :5.004e+09

4246056 : 401 Mean : 6.973 Mean :4.680e+09

4435443 : 309 3rd Qu.: 1.000 3rd Qu.:6.825e+09

4102507C: 273 Max. :591.000 Max. :9.900e+09

(Other) :521664

SaleDate SalePrice

Min. :1997-01-01 Min. : 0

1st Qu.:2003-04-02 1st Qu.: 130000

Median :2007-05-24 Median : 207000

Mean :2008-05-15 Mean : 833401

3rd Qu.:2014-02-07 3rd Qu.: 325000

Max. :2019-06-04 Max. :368586669

DeedType

Statutory Warranty Deed :402204

Trustee Deed (Foreclosure) : 38488

Bargain & Sale Deed : 18576

Special Warranty Deed : 18184

Quit Claim Deed : 13222

Mobile Home Excise Affidavit: 13165

(Other) : 20390

Grantor

NORTHWEST TRUSTEE SERVICES INC: 11872

FEDERAL NATIONAL MORTGAGE ASSO: 4838

QUALITY LOAN SERVICE CORP OF W: 4761

SSHI LLC : 4229

SECRETARY OF HOUSING & URBAN D: 2754

(Other) :481830

NA's : 13945

Grantee ValidInvalid

FEDERAL NATIONAL MORTGAGE ASSOCIATION : 4574 Min. :0.0000

SSHI LLC : 3422 1st Qu.:0.0000

SECRETARY OF HOUSING & URBAN DEVELOPMENT: 2544 Median :1.0000

FEDERAL HOME LOAN MORTGAGE CORPORATION : 1885 Mean :0.6157

WELLS FARGO BANK : 1674 3rd Qu.:1.0000

(Other) :496546 Max. :1.0000

NA's : 13584

ConfirmedUnconfirmed ExcludeReason

Min. :0.0000 MultiParcel Sale one time only: 33579

1st Qu.:0.0000 Improved after sale : 19427

Median :0.0000 Foreclosure Sale : 18120

Mean :0.4739 Estate sale : 11908

3rd Qu.:1.0000 Exempt for taxatn Gov nonprof : 11036

Max. :1.0000 (Other) : 53489

NA's :376670

ImprovedVacant

Min. :0.0000

1st Qu.:1.0000

Median :1.0000

Mean :0.8413

3rd Qu.:1.0000

Max. :1.0000

>

> #eda of seg merge table

> summary(segmerge)

SegMergeNumber ParentChildIndicator ParcelNumber

2012-0331: 423 C:126610 Min. :1.013e+07

2013-0001: 422 P: 43829 1st Qu.:5.192e+08

2018-0012: 324 Median :5.003e+09

2005-0719: 320 Mean :4.360e+09

2006-1194: 319 3rd Qu.:6.470e+09

2005-1191: 303 Max. :1.000e+10

(Other) :168328

ContinuedIndicator CompletedDate TaxYear

N: 42202 Min. :1991-07-17 Min. :1992

Y:128237 1st Qu.:1996-09-16 1st Qu.:1997

Median :2003-03-28 Median :2003

Mean :2003-02-17 Mean :2003

3rd Qu.:2007-11-09 3rd Qu.:2008

Max. :2019-06-12 Max. :2019

NA's :13

>

> #eda of tax account table

> summary(tax\_account)

ParcelNumber AccountType PropertyType

Min. :1.850e+05 MOBIL: 12512 ASIMP: 5

1st Qu.:6.192e+08 PERS : 11114 LNDIM:317506

Median :4.695e+09 REAL :317634 MBLHM: 12512

Mean :4.214e+09 STRUC: 2379 SAPP : 1000

3rd Qu.:6.385e+09 SARP : 123

Max. :9.995e+09 STRUC: 1379

NA's : 11114

SiteAddress UseCode

XXX Undetermined Situs: 1916 Min. : 0

TRACTS : 1154 1st Qu.:1101

OP PROP : 1123 Median :1101

Undetermined Situs : 1006 Mean :2391

REFERENCE : 809 3rd Qu.:1305

XXX ORVILLE RD E : 172 Max. :9900

(Other) :337459 NA's :886

UseDescription TaxYearPrior TaxCodeAreaPrior

SINGLE FAMILY DWELLING :220028 Min. :2018 Min. : 5.0

VACANT LAND UNDEVELOPED: 23805 1st Qu.:2018 1st Qu.: 45.0

MOBILE/MFG HOME : 15482 Median :2018 Median :250.0

MH TITLE ELIM : 9349 Mean :2018 Mean :325.1

DUPLEX 2 UNITS : 5608 3rd Qu.:2018 3rd Qu.:578.0

(Other) : 68481 Max. :2018 Max. :999.0

NA's : 886 NA's :3518

ExemptionTypePriorYear

Senior/Disabled A : 7715

Municipal Corp and Misc Taxing Districts: 4249

Reference Parcels : 3114

Senior/Disabled B : 1789

County Owned Property : 1631

(Other) : 10052

NA's :315089

CurrentUseCodePriorYear LandValuePriorYear ImprovementValuePriorYear

AGRI : 1260 Min. : 0 Min. : 0

FORDG: 2885 1st Qu.: 57600 1st Qu.: 76800

OPBRS: 681 Median : 83900 Median : 155800

OPEN : 216 Mean : 123919 Mean : 218756

NA's :338597 3rd Qu.: 117000 3rd Qu.: 217800

Max. :61605900 Max. :200520800

NA's :14881 NA's :13955

TotalMarketValuePriorYear TaxableValuePriorYear TaxYearCurrent

Min. : 0 Min. : 0 Min. :2019

1st Qu.: 145800 1st Qu.: 112000 1st Qu.:2019

Median : 240000 Median : 231800 Median :2019

Mean : 337970 Mean : 295969 Mean :2019

3rd Qu.: 327300 3rd Qu.: 319600 3rd Qu.:2019

Max. :220839400 Max. :220839400 Max. :2019

NA's :3551 NA's :3551

TaxCodeAreaCurrentYear

Min. : 5.0

1st Qu.: 45.0

Median :250.0

Mean :324.9

3rd Qu.:578.0

Max. :999.0

NA's :463

ExemptionTypeCurrentYear

Senior/Disabled A : 7862

Municipal Corp and Misc Taxing Districts: 4271

Reference Parcels : 3157

Senior/Disabled B : 1848

County Owned Property : 1646

(Other) : 9798

NA's :315057

CurrentUseCodeCurrentYear LandValueCurrentYear

AGRI : 1274 Min. : 0

FORDG: 2905 1st Qu.: 72500

OPBRS: 697 Median : 101100

OPEN : 210 Mean : 147202

NA's :338553 3rd Qu.: 139900

Max. :67142800

NA's :12343

ImprovementValueCurrentYear TotalMarketValueCurrentYear

Min. : 0 Min. : 0

1st Qu.: 87900 1st Qu.: 173300

Median : 170500 Median : 272100

Mean : 236934 Mean : 378027

3rd Qu.: 236800 3rd Qu.: 367000

Max. :209019000 Max. :222953800

NA's :11219 NA's :535

TaxableValueCurrentYear Range Township

Min. : 0 Min. : 0.000 Min. : 0.00

1st Qu.: 133700 1st Qu.: 2.000 1st Qu.:19.00

Median : 263700 Median : 3.000 Median :20.00

Mean : 332097 Mean : 3.123 Mean :19.58

3rd Qu.: 358400 3rd Qu.: 4.000 3rd Qu.:20.00

Max. :222953800 Max. :11.000 Max. :22.00

NA's :535 NA's :24782 NA's :24782

Section QuarterSection SubdivisionName

05 : 13361 22 : 21102 2ND SCHOOL LD ADD: 2205

28 : 12506 41 : 21033 TAC LAND CO 6TH : 1738

03 : 11974 33 : 20884 1ST SCHOOL LD ADD: 1538

32 : 11402 44 : 20709 INDIAN ADD : 1190

04 : 11246 11 : 20487 LAKE PARK : 971

(Other):258369 (Other):214642 (Other) :219011

NA's : 24781 NA's : 24782 NA's :116986

LocatedOnParcel

| :320721

0419104001|: 240

0419203047|: 182

0320312076|: 168

0420223061|: 159

0319281056|: 143

(Other) : 22026

>

> #eda of tax description table

> summary(tax\_description)

ParcelNumber LineNumber

Min. :1.901e+07 Min. : 1.000

1st Qu.:4.151e+08 1st Qu.: 1.000

Median :3.375e+09 Median : 3.000

Mean :3.500e+09 Mean : 4.623

3rd Qu.:6.024e+09 3rd Qu.: 6.000

Max. :9.995e+09 Max. :102.000

TaxDescriptionLine

EASE OF RECORD| : 6245

TOG/W EASE & RESTRICTIONS OF REC| : 2419

NEW TACOMA| : 1780

TAC LAND CO 6TH| : 1273

RECORD| : 1262

EASE & RESTRICTIONS OF REC OUT OF|: 1236

(Other) :1179349

>

> #eda of complied data table

> summary(compiled)

ParcelNumber AppraisalAccountType Buildings

Min. :2.101e+07 Residential:127232 Min. : 0.000

1st Qu.:5.202e+08 1st Qu.: 1.000

Median :5.003e+09 Median : 1.000

Mean :4.334e+09 Mean : 1.199

3rd Qu.:6.026e+09 3rd Qu.: 1.000

Max. :9.900e+09 Max. :10.000

LandGrossSquareFeet LandNetSquareFeet AppraisalDate

Min. : 0 Min. : 510 Min. :2006-04-19

1st Qu.: 7778 1st Qu.: 7781 1st Qu.:2014-12-29

Median : 12000 Median : 12000 Median :2015-11-18

Mean : 37781 Mean : 37809 Mean :2016-04-24

3rd Qu.: 22576 3rd Qu.: 22575 3rd Qu.:2018-04-26

Max. :13939200 Max. :13939200 Max. :2019-06-25

NA's :2

Latitude Longtitude BuildingID PropertyType

Min. :46.77 Min. :-122.8 Min. : 1.000 Residential:127232

1st Qu.:47.11 1st Qu.:-122.5 1st Qu.: 1.000

Median :47.16 Median :-122.3 Median : 1.000

Mean :47.17 Mean :-122.4 Mean : 1.045

3rd Qu.:47.22 3rd Qu.:-122.3 3rd Qu.: 1.000

Max. :47.40 Max. :-122.1 Max. :11.000

SquareFeet PercentComplete Condition

Min. : 1 Min. :0.0000 Average :123188

1st Qu.: 1419 1st Qu.:1.0000 Extra Poor : 167

Median : 1822 Median :1.0000 Fair : 2807

Mean : 1920 Mean :0.9976 Good : 19

3rd Qu.: 2320 3rd Qu.:1.0000 Poor : 656

Max. :13004 Max. :1.0000 Uninhabitable: 173

Very Poor : 222

Quality PhysicalAge YearBuilt YearRemodeled

Average :70678 Min. : 0.0 Min. :1872 Min. : 0

Fair Plus :20038 1st Qu.: 16.0 1st Qu.:1971 1st Qu.: 0

Average Plus:16041 Median : 25.0 Median :1990 Median :1980

Fair : 7932 Mean : 25.4 Mean :1984 Mean :1207

Good : 6197 3rd Qu.: 33.0 3rd Qu.:2002 3rd Qu.:1992

Good Plus : 2606 Max. :119.0 Max. :2019 Max. :2019

(Other) : 3740 NA's :368

AccountType TaxableValuePriorYear TaxableValueCurrentYear

REAL:127232 Min. : 0 Min. : 0

1st Qu.: 229100 1st Qu.: 261000

Median : 282800 Median : 319300

Mean : 322957 Mean : 364307

3rd Qu.: 368500 3rd Qu.: 413900

Max. :4182900 Max. :4600100

NA's :202

Zipcode

Min. :98328

1st Qu.:98371

Median :98375

Mean :98386

3rd Qu.:98391

Max. :98467

Correlation Analysis

# Nominal data:

dfnominal<- df %>% select(Condition, Quality)

# Builds a contingency table

nominaltable <- table(dfnorminal$Condition,dfnorminal$Quality)

# conduct Chi-square test

chisq.test(nominaltable)

Chi-squared approximation may be incorrect

Pearson's Chi-squared test

data: nominaltable

X-squared = 1823.2, df = 60, p-value < 2.2e-16

## P-VALUE < 2.2e-16, We reject the null hypothesis. We can say that Condition and Quality have a high correlation

# numerical data:

# Numerical data

dfnumerical <- df %>% select("Buildings","LandGrossSquareFeet","LandNetSquareFeet","SquareFeet","PercentComplete","PhysicalAge","YearBuilt","YearRemodeled","TaxableValuePriorYear","TaxableValueCurrentYear","Arrests","Arson","Assault","Burglary","Drug","Fraud","Harassment","Homicide","Intimidation","LiquorLaw","Robbery","Theft","Traffic","Vandalism")

#correlation matrix

cor(dfnumerical,use ='complete.obs')

# Overall Parcel attribute do not have much correlation with each others, except, Year Build, Year remodel and Physical Age

# Number of type of crime occured have strong correlation wtich each other

# House value has low correlation coefficient with crime variable

# function to test correlation between variable in the dataset.

funccortest <- function(listx, df){

#y = enquo(yaxis)

for(i in 1:length(df)){

for(j in 2:length(df)){

cortest<- cor.test(df[,i], df[,j])

p <- cortest$p.value

result <- ifelse(p<= 0.05, "are highly correlated", "are independent")

print(paste('p-value =',p,',' ,listx[i],'and',listx[j],result, sep =' '))

}

}

}

# testing correlation all numerical variable

funccortest(colnames(dfnumerical),dfnumerical)

Correlation Analysis R Code:

Creating sql server connection

===============================

```{r}

library(tidyverse)

library(RODBC)

library(stringr)

library(psych)

conn <- odbcConnect("oitap22")

```

Loading data from sql server

===============================

```{r}

# get data sample from sql server

data <- sqlQuery(conn,"select \* from [19su5510\_lehuy].[dbo].[FinalData0809]")

data <- data[complete.cases(data), ]

summary(data)

```

Correlation analysis

==================================

```{r}

# Prepare Nominal data

dfnominal<- data %>% select(Condition, Quality)

# Builds a contingency table

nominaltable <- table(dfnorminal$Condition,dfnorminal$Quality)

chisq.test(nominaltable)

```

- X-squared =1871.8

- P-VALUE < 2.2e-16 => Condition and Quality have a strong correlation

```{r}

colnames(data)

**# Prepare Numerical data**

dfnumerical <- data %>% select("Buildings","LandNetSquareFeet","SquareFeet",

"PhysicalAge","YearBuilt","YearRemodeled",

"TaxableValuePriorYear","TaxableValueCurrentYear",

"DrugCrime","Homicide","PropertyCrime","PersonalCrime",

"OtherCrime","SalePrice") %>% distinct()

**#correlation matrix**

library(corrplot)

library(RColorBrewer)

M <-cor(dfnumerical,use ='complete.obs')

jpeg("corrplot.jpg")

corrplot(M, type="upper", order="hclust",

col=brewer.pal(n=8, name="RdYlBu"))

dev.off()

# Overall Parcel attribute do not have much correlation with each others, except, Year Build, Year remodel and Physical Age

# Number of type of crime occured have strong correlation wtich each other

# House value has low correlation coefficient with crime variable

**# function to test correlation between variable in the dataset.**

funccortest <- function(listx, df){

#y = enquo(yaxis)

for(i in 1:length(df)){

for(j in 2:length(df)){

cortest<- cor.test(df[,i], df[,j])

p <- cortest$p.value

result <- ifelse(p<= 0.05, "are highly correlated", "are independent")

print(paste('p-value =',p,',' ,listx[i],'and',listx[j],result, sep =' '))

}

}

}

# testing correlation all numerical variable

funccortest(colnames(dfnumerical),dfnumerical)

```

Coordinate Distance Function

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[getDistanceSTDistanctMile] Script Date: 8/12/2019 4:00:19 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE function [dbo].[getDistanceSTDistanceMile] (@lat1 as real,

@lon1 as real,

@lat2 as real,

@lon2 as real)

returns real as

begin

declare @d as real

declare @NWI geography, @EDI geography

SET @NWI = geography::Point( @lat1,@lon1, 4326)

SET @EDI = geography::Point( @lat2,@lon2, 4326)

SELECT @d = @NWI.STDistance(@EDI) / 1609.344

return @d

end;

GO

Data Integration

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object: View [dbo].[CompiledDataVersion1] Script Date: 8/12/2019 4:09:24 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create view [dbo].[CompiledDataVersion1]

as

select

a.ParcelNumber,

a.AppraisalAccountType,

a.Buildings,

a.LandGrossSquareFeet,

a.LandNetSquareFeet,

a.AppraisalDate,

a.Latitude,

a.Longtitude,

b.BuildingID,

b.PropertyType,

b.SquareFeet,

b.PercentComplete,

b.Condition,

b.Quality,

c.PhysicalAge,

c.YearBuilt,

c.YearRemodeled,

d.AccountType,

d.TaxableValuePriorYear,

d.TaxableValueCurrentYear,

e.Zipcode

from dbo.AppraisalAccount a

left join dbo.Improvement b ON b.ParcelNumber = a.ParcelNumber

left join dbo.ImprovementBuiltas c ON c.ParcelNumber = b.ParcelNumber AND c.BuildingID = b.BuildingID

left join dbo.TaxAccount d ON d.ParcelNumber = c.ParcelNumber

left join dbo.AddressPoint e ON e.ParcelNumber = d.ParcelNumber

WHERE a.AppraisalAccountType = 'Residential' AND b.PropertyType = 'Residential' AND d.AccountType = 'REAL'

AND e.Zipcode in ('98328','98338','98375','98387','98385','98448','98374','98373','98446','98445','98444','98467','98464','98371','98466','98391','98372','98335','98335','98394','98332','98329')

GO

Sample Crime Data & Distance Calculation

insert into [dbo].[LandCrimeDistanceSample2]

select ta.ParcelNumber, d.OBJECTID,

ta.Latitude,ta.Longtitude, d.lattitude, d.longtitude,

[dbo].[getDistanceSTDistanceMile](ta.Latitude,ta.Longtitude, d.lattitude, d.longtitude) as Distance

from

(select a.ParcelNumber, a.Latitude as Latitude, a.Longtitude as Longtitude, ap.ZipCode

from AppraisalAccount a inner join Sale s on a.ParcelNumber= s.ParcelNumber

left join AddressPoint ap on a.ParcelNumber= ap.ParcelNumber

where s.SaleDate >='2018-01-01' and

ap.ZipCode in ('98328','98338','98375','98387',

'98385','98448','98374','98373','98446',

'98445','98444','98467','98464','98371',

'98466','98391','98372','98335','98335',

'98394','98332','98329')

) ta ,

(select \* from CrimeData TABLESAMPLE (10 PERCENT)) d

Views

The following view integrates house attribute data from CompiledDataVersion1 view, table LandCrimeDistanceSample2 and table Crime Data. Then It aggregated and grouped crime by new 5 categories as mentioned in the paper.

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object: View [dbo].[groupedsample] Script Date: 8/12/2019 4:17:27 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE view [dbo].[groupedsample]

as

select d.\*, e.CrimeCategory, e.NoofCrime

from CompiledDataVersion1 d right join

(select tb1.ParcelNumber, tb1.CrimeCategory, count( tb1.CrimeCategory) as NoofCrime

from

(select a.ParcelNumber, c.Public\_Nam,case

when Public\_Nam in ('Arson - Non-residential',

'Arson - Residential',

'Burglary - Non-residential',

'Burglary - Residential','Fraud or Forgery','Motor Vehicle Theft','Possession of Stolen Property',

'Robbery - Business','Robbery - Residential','Robbery - Street','Robbery - Other',

'Theft - Gas Station Runout','Theft - Mail','Theft - Other','Theft - Vehicle Prowl','Theft -Shoplifing','Trafficking in Stolen Property'

) then 'PropertyCrime'

when Public\_Nam like '%Drug%' then 'DrugCrime'

when Public\_Nam = 'Homicide' then 'Homicide'

when Public\_Nam in ('Assault - Aggravated','Assault - Simple','Intimidation','Telephone Harassment','Vandalism - Non-residential','Vandalism - Residential') then 'PersonalCrime'

else 'OtherCrime'

end as CrimeCategory

from LandCrimeDistanceSample2 a left join CrimeData c on a.ObjectID = c.OBJECTID

where a.Distance <=1) tb1

group by tb1.ParcelNumber, tb1.CrimeCategory) e on d.ParcelNumber = e.ParcelNumber

GO

The following view spreads crime data in groupedsample view to column. This action is to tidy data table by using pivot function, make every single row is correspond with only one property.

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object: View [dbo].[groupeddatatull] Script Date: 8/12/2019 4:23:55 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE view [dbo].[groupeddatafull]

as

select a.ParcelNumber, a.AppraisalAccountType,

a.Buildings, a.LandGrossSquareFeet,

a.LandNetSquareFeet, a.AppraisalDate,

a.Latitude, a.Longtitude, a.BuildingID, a.PropertyType,

a.SquareFeet, z.PercentComplete, a.Condition, a.Quality, a.PhysicalAge,

a.YearBuilt, a.YearRemodeled,

a.AccountType, a.TaxableValuePriorYear,

a.TaxableValueCurrentYear, a.Zipcode, b. [DrugCrime], b.[Homicide], b.[OtherCrime], b.[PersonalCrime], b.[PropertyCrime]

from [dbo].[CompiledDataVersion1] a left join

(select ParcelNumber,[DrugCrime],[Homicide],[OtherCrime],[PersonalCrime],[PropertyCrime]

from

(select ParcelNumber, CrimeCategory, NoofCrime

from [dbo].[groupedsample] ) as a

pivot (max(a.NoofCrime) for a.CrimeCategory in ([DrugCrime],[Homicide],[OtherCrime],[PersonalCrime],[PropertyCrime]))as p ) b

on a.ParcelNumber = b.ParcelNumber

WHERE

--Zipcode in ('98371', '98372', '98373','98374','98375') and

a.ParcelNumber in( select l.ParcelNumber from LandCrimeDistanceSample2 l)

GO

The following view is the data in groupeddatafull integrated with sale data. We only select those most recent sale data from 01-01-2018 until now.

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object: View [dbo].[GroupedDatawithSale] Script Date: 8/12/2019 4:32:09 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

/\*\*\*\*\*\* Script for SelectTopNRows command from SSMS \*\*\*\*\*\*/

CREATE view [dbo].[GroupedDatawithSale]

as

select a.ParcelNumber,

a.Buildings,

a.LandNetSquareFeet,

a.SquareFeet,

a.Condition,

a.Quality,

a.PhysicalAge,

a.YearBuilt,

a.YearRemodeled,

a.TaxableValueCurrentYear,

a.TaxableValuePriorYear,

isnull(a.DrugCrime,0) as DrugCrime,

isnull(a.Homicide,0) as Homicide,

isnull(a.PropertyCrime,0)as PropertyCrime,

isnull(a.PersonalCrime,0) as PersonalCrime,

isnull(a.OtherCrime,0) as OtherCrime, s.SalePrice, s.SaleDate

from groupeddatatull a inner join

(select ParcelNumber, SalePrice, SaleDate

from(

select Row\_number()

OVER (

partition BY (Sale.ParcelNumber)

ORDER BY Sale.SaleDate Desc) SaleOrder, ParcelNumber, SaleDate,SalePrice

from Sale) so

where SaleOrder=1) s on a.ParcelNumber = s.ParcelNumber

where s.SaleDate >='2018-01-01'

GO

After having a view with final data, we insert those data into table ***FinalData0809***. To keep this dataset constant

The following view is a normalized version of final dataset. We used this view for clustering section.

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object: View [dbo].[FinalNormalizedData] Script Date: 8/12/2019 4:34:52 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE view [dbo].[FinalNormalizedData]

as

select ParcelNumber, (cast(f.Buildings as float) - 0)/(10-0) as Buildings

, (f.LandNetSquareFeet - 1500)/(4316796-1500) as LandNetSquareFeet

, (f.SquareFeet - 192)/(7230-192) as SquareFeet

, (cast(f.PhysicalAge as float) - 0)/(119-0) as PhysicalAge

, (cast(f.YearBuilt as float) - 1890)/(2019-1890) as YearBuilt

, (cast(f.YearRemodeled as float)- 0)/(2019-0) as YearRemodeled

, (f.TaxableValueCurrentYear - 0)/(1979600-0) as TaxableValueCurrentYear

, (f.TaxableValuePriorYear - 0)/(1882900-0) as TaxableValuePriorYear

, (cast(f.DrugCrime as float)- 0)/(51-0) as DrugCrime

, (cast(f.Homicide as float) - 0)/(2-0) as Homicide

, (cast(f.PropertyCrime as float)- 0)/(459-0) as PropertyCrime

, (cast(f.PersonalCrime as float)- 0)/(108-0) as PersonalCrime

, (cast(f.OtherCrime as float)- 0)/(393-0) as OtherCrime

, (f.SalePrice - 500)/(21700000-500) as SalePrice

, f.SaleDate, f.Condition, f.Quality

from FinalData0809 f

GO

Clustering

Finally, we used visual studio SQL data tool to cluster final dataset, then create data with cluster column.

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object:  View [dbo].[FinalDataClusterWithCrime]    Script Date: 8/13/2019 3:06:41 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create view [dbo].[FinalDataClusterWithCrime]

as

select f.\*, c.[$CLUSTER] as Cluster

from FinalData0809 f inner join ClusteredDataWithCrime c on f.ParcelNumber = c.ParcelNumber

GO

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object:  View [dbo].[FinalDataClusterWithoutCrime]    Script Date: 8/13/2019 3:10:30 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create view [dbo].[FinalDataClusterWithoutCrime]

as

select f.\*, c.[$CLUSTER] as Cluster

from FinalData0809 f inner join ClusteredDataWithoutCrime c on f.ParcelNumber = c.ParcelNumber

GO

These two views below are created for building models.

FinalDataClusterWithCrime view

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object: View [dbo].[FinalDataClusterWithCrime] Script Date: 8/13/2019 3:06:41 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create view [dbo].[FinalDataClusterWithCrime]

as

select f.\*, c.[$CLUSTER] as Cluster

from FinalData0809 f inner join ClusteredDataWithCrime c on f.ParcelNumber = c.ParcelNumber

GO

FinalDataClusterWithoutCrime view

USE [19su5510\_lehuy]

GO

/\*\*\*\*\*\* Object: View [dbo].[FinalDataClusterWithoutCrime] Script Date: 8/13/2019 3:10:30 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create view [dbo].[FinalDataClusterWithoutCrime]

as

select f.\*, c.[$CLUSTER] as Cluster

from FinalData0809 f inner join ClusteredDataWithoutCrime c on f.ParcelNumber = c.ParcelNumber

GO

Decision Tree

Decision tree models were created in Microsoft Visual Studio.

Neural Network

The following is the R code for the neural network model using the clustered data.

library(tidyverse)

library(RODBC)

library(stringr)

library(psych)

install.packages("neuralnet")

library(neuralnet)

conn <- odbcConnect("oitap22")

#===================================

# model with clustered data with crime

#===================================

dfmn <- sqlQuery(conn,"select \* from [19su5510\_angelovp].[dbo].[FinalDataClusterWithCrime]")

apply(dfmn,2,range)

fullds <- dfmn %>% select(-Condition, -Quality,-SaleDate) %>%

mutate(cluster = as.numeric(substr(`Cluster`,8,9))) %>%

select(-`Cluster`)

##### Normalize data

maxVals <- apply(fullds, 2,max)

minVals <- apply(fullds, 2,min)

fulldsCluster1 <- fullds %>% filter(cluster ==1)

fulldsCluster2 <- fullds %>% filter(cluster ==2)

fulldsCluster3 <- fullds %>% filter(cluster ==3)

fullds <- fullds %>% select(-cluster)

#===================================

# full data

dfmnNfulldata <- as.data.frame(scale(fullds , center = minVals, scale = (maxVals-minVals)))

#Partition the dataset for training and testing

set.seed(1234) #A seed can be any number you like.

rs <- sample(nrow(dfmnNfulldata), .3\*nrow(dfmnNfulldata))

Training <-fullds[rs,]

Testing <- fullds[-rs,]

TrainingN <- dfmnNfulldata[rs,]

TestingN <- dfmnNfulldata[-rs,]

attach(TrainingN)

colnames(TrainingN)

TrainingN[is.na(TrainingN)] <- 0

TestingN[is.na(TestingN)] <- 0

# full data

#A model with 10 nodes on 1 hidden layer

nm1 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt+DrugCrime+Homicide+OtherCrime+PersonalCrime+PropertyCrime, hidden = c(10), data=TrainingN)

plot(nm1)

#testing

testPred1 <- compute(nm1,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt','DrugCrime','Homicide','OtherCrime','PersonalCrime','PropertyCrime')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse1 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse1

mae1 <- sum(abs(testPred1$net.result- TestingN$SalePrice))/nrow(TestingN)

mae1

################################

actualTestPred1 <- testPred1$net.result \* (max(fullds[,'SalePrice'])-min(fullds[,'SalePrice'])) + min(fullds[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual1 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual1

sqrt (mseActual1)

maeActual1 <- sum(abs(actualTestPred1- Testing [,'SalePrice']))/nrow(Testing)

maeActual1

# full data

#A model with 10,15 nodes on 2 hidden layers

nm42 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt+DrugCrime+Homicide+OtherCrime+PersonalCrime+PropertyCrime, hidden = c(10,15), data=TrainingN)

plot(nm42)

#testing

testPred42 <- compute(nm42,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt','DrugCrime','Homicide','OtherCrime','PersonalCrime','PropertyCrime')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse42 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse42

mae42 <- sum(abs(testPred42$net.result- TestingN$SalePrice))/nrow(TestingN)

mae42

################################

actualTestPred42 <- testPred42$net.result \* (max(fullds[,'SalePrice'])-min(fullds[,'SalePrice'])) + min(fullds[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual42 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual42

sqrt (mseActual42)

maeActual42 <- sum(abs(actualTestPred42- Testing [,'SalePrice']))/nrow(Testing)

maeActual42

#===================================

# cluster 1

dfmnNCluster1 <- as.data.frame(scale(fulldsCluster1 , center = minVals, scale = (maxVals-minVals)))

#Partition the dataset for training and testing

set.seed(1234) #A seed can be any number you like.

rs <- sample(nrow(dfmnNCluster1), .8\*nrow(dfmnNCluster1))

Training <-fulldsCluster1[rs,]

Testing <- fulldsCluster1[-rs,]

TrainingN <- dfmnNCluster1[rs,]

TestingN <- dfmnNCluster1[-rs,]

attach(TrainingN)

colnames(TrainingN)

TrainingN[is.na(TrainingN)] <- 0

TestingN[is.na(TestingN)] <- 0

# cluster 1

#A model with 10 nodes on 1 hidden layer

nm1 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt+DrugCrime+Homicide+OtherCrime+PersonalCrime+PropertyCrime, hidden = c(10), data=TrainingN)

plot(nm1)

#testing

testPred1 <- compute(nm1,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt','DrugCrime','Homicide','OtherCrime','PersonalCrime','PropertyCrime')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse1 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse1

mae1 <- sum(abs(testPred1$net.result- TestingN$SalePrice))/nrow(TestingN)

mae1

################################

actualTestPred1 <- testPred1$net.result \* (max(fulldsCluster1[,'SalePrice'])-min(fulldsCluster1[,'SalePrice'])) + min(fulldsCluster1[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual1 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual1

sqrt (mseActual1)

maeActual1 <- sum(abs(actualTestPred1- Testing [,'SalePrice']))/nrow(Testing)

maeActual1

# cluster 1

#A model with 10,15 nodes on 2 hidden layers

nm42 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt+DrugCrime+Homicide+OtherCrime+PersonalCrime+PropertyCrime, hidden = c(10,15), data=TrainingN)

plot(nm42)

#testing

testPred42 <- compute(nm42,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt','DrugCrime','Homicide','OtherCrime','PersonalCrime','PropertyCrime')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse42 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse42

mae42 <- sum(abs(testPred42$net.result- TestingN$SalePrice))/nrow(TestingN)

mae42

################################

actualTestPred42 <- testPred42$net.result \* (max(fulldsCluster1[,'SalePrice'])-min(fulldsCluster1[,'SalePrice'])) + min(fulldsCluster1[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual42 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual42

sqrt (mseActual42)

maeActual42 <- sum(abs(actualTestPred42- Testing [,'SalePrice']))/nrow(Testing)

maeActual42

#===================================

# cluster 2

rs <- sample(nrow(dfmnNCluster2), .8\*nrow(dfmnNCluster2))

Training <-fulldsCluster2[rs,]

Testing <- fulldsCluster2[-rs,]

TrainingN <- dfmnNCluster2[rs,]

TestingN <- dfmnNCluster2[-rs,]

attach(TrainingN)

colnames(TrainingN)

TrainingN[is.na(TrainingN)] <- 0

TestingN[is.na(TestingN)] <- 0

# cluster 2

#A model with 10 nodes on 1 hidden layer

nm1 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt+DrugCrime+Homicide+OtherCrime+PersonalCrime+PropertyCrime, hidden = c(10), data=TrainingN)

plot(nm1)

#testing

testPred1 <- compute(nm1,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt','DrugCrime','Homicide','OtherCrime','PersonalCrime','PropertyCrime')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse1 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse1

mae1 <- sum(abs(testPred1$net.result- TestingN$SalePrice))/nrow(TestingN)

mae1

################################

actualTestPred1 <- testPred1$net.result \* (max(fulldsCluster2[,'SalePrice'])-min(fulldsCluster2[,'SalePrice'])) + min(fulldsCluster2[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual1 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual1

sqrt (mseActual1)

maeActual1 <- sum(abs(actualTestPred1- Testing [,'SalePrice']))/nrow(Testing)

maeActual1

# cluster 2

#A model with 10,15 nodes on 2 hidden layers

nm42 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt+DrugCrime+Homicide+OtherCrime+PersonalCrime+PropertyCrime, hidden = c(10,15), data=TrainingN)

plot(nm42)

#testing

testPred42 <- compute(nm42,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt','DrugCrime','Homicide','OtherCrime','PersonalCrime','PropertyCrime')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse42 <- sum((testPred42$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse42

mae42 <- sum(abs(testPred42$net.result- TestingN$SalePrice))/nrow(TestingN)

mae42

################################

actualTestPred42 <- testPred42$net.result \* (max(fulldsCluster2[,'SalePrice'])-min(fulldsCluster2[,'SalePrice'])) + min(fulldsCluster2[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual42 <- sum((actualTestPred42- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual42

sqrt (mseActual42)

maeActual42 <- sum(abs(actualTestPred42- Testing [,'SalePrice']))/nrow(Testing)

maeActual42

#===================================

# cluster 3

rs <- sample(nrow(dfmnNCluster3), .8\*nrow(dfmnNCluster3))

Training <-fulldsCluster3[rs,]

Testing <- fulldsCluster3[-rs,]

TrainingN <- dfmnNCluster3[rs,]

TestingN <- dfmnNCluster3[-rs,]

attach(TrainingN)

colnames(TrainingN)

TrainingN[is.na(TrainingN)] <- 0

TestingN[is.na(TestingN)] <- 0

# cluster 3

#A model with 10 nodes on 1 hidden layer

nm1 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt+DrugCrime+Homicide+OtherCrime+PersonalCrime+PropertyCrime, hidden = c(10), data=TrainingN)

plot(nm1)

#testing

testPred1 <- compute(nm1,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt','DrugCrime','Homicide','OtherCrime','PersonalCrime','PropertyCrime')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse1 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse1

mae1 <- sum(abs(testPred1$net.result- TestingN$SalePrice))/nrow(TestingN)

mae1

################################

actualTestPred1 <- testPred1$net.result \* (max(fulldsCluster3[,'SalePrice'])-min(fulldsCluster3[,'SalePrice'])) + min(fulldsCluster3[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual1 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual1

sqrt (mseActual1)

maeActual1 <- sum(abs(actualTestPred1- Testing [,'SalePrice']))/nrow(Testing)

maeActual1

# cluster 3

#A model with 10,15 nodes on 2 hidden layers

nm42 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt+DrugCrime+Homicide+OtherCrime+PersonalCrime+PropertyCrime, hidden = c(10,15), data=TrainingN)

plot(nm42)

#testing

testPred42 <- compute(nm42,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt','DrugCrime','Homicide','OtherCrime','PersonalCrime','PropertyCrime')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse42 <- sum((testPred42$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse42

mae42 <- sum(abs(testPred42$net.result- TestingN$SalePrice))/nrow(TestingN)

mae42

################################

actualTestPred42 <- testPred42$net.result \* (max(fulldsCluster3[,'SalePrice'])-min(fulldsCluster3[,'SalePrice'])) + min(fulldsCluster3[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual42 <- sum((actualTestPred42- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual42

sqrt (mseActual42)

maeActual42 <- sum(abs(actualTestPred42- Testing [,'SalePrice']))/nrow(Testing)

maeActual42

#===================================

# model with clustered data without crime

#===================================

dfmn <- sqlQuery(conn,"select \* from [19su5510\_angelovp].[dbo].[FinalDataClusterWithoutCrime]")

apply(dfmn,2,range)

fullds <- dfmn %>% select(-Condition, -Quality,-SaleDate, -PersonalCrime, -DrugCrime, -OtherCrime, -Homicide, -PropertyCrime) %>%

mutate(cluster = as.numeric(substr(`Cluster`,8,9))) %>%

select(-`Cluster`)

##### Normalize data

maxVals <- apply(fullds, 2,max)

minVals <- apply(fullds, 2,min)

fulldsCluster1 <- fullds %>% filter(cluster ==1)

fulldsCluster2 <- fullds %>% filter(cluster ==2)

fulldsCluster3 <- fullds %>% filter(cluster ==3)

fullds <- fullds %>% select(-cluster)

#===================================

# full data

dfmnNfulldata <- as.data.frame(scale(fullds , center = minVals, scale = (maxVals-minVals)))

#Partition the dataset for training and testing

set.seed(1234) #A seed can be any number you like.

# full data

rs <- sample(nrow(dfmnNfulldata), .3\*nrow(dfmnNfulldata))

Training <-dfmnNfulldata[rs,]

Testing <- dfmnNfulldata[-rs,]

TrainingN <- dfmnNfulldata[rs,]

TestingN <- dfmnNfulldata[-rs,]

attach(TrainingN)

# full data

#A model with 10 nodes on 1 hidden layer

nm1 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt, hidden = c(10), data=TrainingN)

plot(nm1)

#testing

testPred1 <- compute(nm1,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse1 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse1

mae1 <- sum(abs(testPred1$net.result- TestingN$SalePrice))/nrow(TestingN)

mae1

################################

actualTestPred1 <- testPred1$net.result \* (max(fullds[,'SalePrice'])-min(fullds[,'SalePrice'])) + min(fullds[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual1 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual1

sqrt (mseActual1)

maeActual1 <- sum(abs(actualTestPred1- Testing [,'SalePrice']))/nrow(Testing)

maeActual1

# full model

#A model with 4 nodes on 2 hidden layers

nm42 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt, hidden = c(10,15), data=TrainingN)

plot(nm42)

#testing

testPred42 <- compute(nm42,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse42 <- sum((testPred42$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse42

mae42 <- sum(abs(testPred42$net.result- TestingN$SalePrice))/nrow(TestingN)

mae42

################################

actualTestPred42 <- testPred42$net.result \* (max(fullds[,'SalePrice'])-min(fullds[,'SalePrice'])) + min(fullds[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual42 <- sum((actualTestPred42- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual42

sqrt (mseActual42)

maeActual42 <- sum(abs(actualTestPred42- Testing [,'SalePrice']))/nrow(Testing)

maeActual42

#===================================

# cluster 1

dfmnNCluster1 <- as.data.frame(scale(fulldsCluster1 , center = minVals, scale = (maxVals-minVals)))

#Partition the dataset for training and testing

set.seed(1234) #A seed can be any number you like.

rs <- sample(nrow(dfmnNCluster1), .8\*nrow(dfmnNCluster1))

Training <-fulldsCluster1[rs,]

Testing <- fulldsCluster1[-rs,]

TrainingN <- dfmnNCluster1[rs,]

TestingN <- dfmnNCluster1[-rs,]

attach(TrainingN)

colnames(TrainingN)

TrainingN[is.na(TrainingN)] <- 0

TestingN[is.na(TestingN)] <- 0

# cluster 1

#A model with 10 nodes on 1 hidden layer

nm1 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt, hidden = c(10), data=TrainingN)

plot(nm1)

#testing

testPred1 <- compute(nm1,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse1 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse1

mae1 <- sum(abs(testPred1$net.result- TestingN$SalePrice))/nrow(TestingN)

mae1

################################

actualTestPred1 <- testPred1$net.result \* (max(fulldsCluster1[,'SalePrice'])-min(fulldsCluster1[,'SalePrice'])) + min(fulldsCluster1[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual1 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual1

sqrt (mseActual1)

maeActual1 <- sum(abs(actualTestPred1- Testing [,'SalePrice']))/nrow(Testing)

maeActual1

# cluster 1

#A model with 10 & 15 nodes on 2 hidden layers

nm42 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt, hidden = c(10,15), data=TrainingN)

plot(nm42)

#testing

testPred42 <- compute(nm42,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse42 <- sum((testPred42$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse42

mae42 <- sum(abs(testPred42$net.result- TestingN$SalePrice))/nrow(TestingN)

mae42

################################

actualTestPred42 <- testPred42$net.result \* (max(fulldsCluster1[,'SalePrice'])-min(fulldsCluster1[,'SalePrice'])) + min(fulldsCluster1[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual42 <- sum((actualTestPred42- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual42

sqrt (mseActual42)

maeActual42 <- sum(abs(actualTestPred42- Testing [,'SalePrice']))/nrow(Testing)

maeActual42

#===================================

# cluster 2

dfmnNCluster2 <- as.data.frame(scale(fulldsCluster2 , center = minVals, scale = (maxVals-minVals)))

#Partition the dataset for training and testing

set.seed(1234) #A seed can be any number you like.

rs <- sample(nrow(dfmnNCluster2), .8\*nrow(dfmnNCluster2))

Training <-fulldsCluster2[rs,]

Testing <- fulldsCluster2[-rs,]

TrainingN <- dfmnNCluster2[rs,]

TestingN <- dfmnNCluster2[-rs,]

attach(TrainingN)

colnames(TrainingN)

TrainingN[is.na(TrainingN)] <- 0

TestingN[is.na(TestingN)] <- 0

# cluster 2

#A model with 10 nodes on 1 hidden layer

nm1 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt, hidden = c(10), data=TrainingN)

plot(nm1)

#testing

testPred1 <- compute(nm1,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse1 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse1

mae1 <- sum(abs(testPred1$net.result- TestingN$SalePrice))/nrow(TestingN)

mae1

################################

actualTestPred1 <- testPred1$net.result \* (max(fulldsCluster2[,'SalePrice'])-min(fulldsCluster2[,'SalePrice'])) + min(fulldsCluster2[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual1 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual1

sqrt (mseActual1)

maeActual1 <- sum(abs(actualTestPred1- Testing [,'SalePrice']))/nrow(Testing)

maeActual1

# cluster 2

#A model with 10,15 nodes on 2 hidden layers

nm42 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt, hidden = c(10,15), data=TrainingN)

plot(nm42)

#testing

testPred42 <- compute(nm42,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse42 <- sum((testPred42$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse42

mae42 <- sum(abs(testPred42$net.result- TestingN$SalePrice))/nrow(TestingN)

mae42

[1] 0.00334919

################################

actualTestPred42 <- testPred42$net.result \* (max(fulldsCluster2[,'SalePrice'])-min(fulldsCluster2[,'SalePrice'])) + min(fulldsCluster2[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual42 <- sum((actualTestPred42- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual42

sqrt (mseActual42)

maeActual42 <- sum(abs(actualTestPred42- Testing [,'SalePrice']))/nrow(Testing)

maeActual42

#===================================

# cluster 3

dfmnNCluster3 <- as.data.frame(scale(fulldsCluster3 , center = minVals, scale = (maxVals-minVals)))

#Partition the dataset for training and testing

set.seed(1234) #A seed can be any number you like.

rs <- sample(nrow(dfmnNCluster3), .8\*nrow(dfmnNCluster3))

Training <-fulldsCluster3[rs,]

Testing <- fulldsCluster3[-rs,]

TrainingN <- dfmnNCluster3[rs,]

TestingN <- dfmnNCluster3[-rs,]

attach(TrainingN)

colnames(TrainingN)

TrainingN[is.na(TrainingN)] <- 0

TestingN[is.na(TestingN)] <- 0

# cluster 3

#A model with 10 nodes on 1 hidden layer

nm1 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt, hidden = c(10), data=TrainingN)

plot(nm1)

#testing

testPred1 <- compute(nm1,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse1 <- sum((testPred1$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse1

mae1 <- sum(abs(testPred1$net.result- TestingN$SalePrice))/nrow(TestingN)

mae1

################################

actualTestPred1 <- testPred1$net.result \* (max(fulldsCluster3[,'SalePrice'])-min(fulldsCluster3[,'SalePrice'])) + min(fulldsCluster3[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual1 <- sum((actualTestPred1- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual1

sqrt (mseActual1)

maeActual1 <- sum(abs(actualTestPred1- Testing [,'SalePrice']))/nrow(Testing)

maeActual1

# cluster 3

#A model with 10,15 nodes on 2 hidden layers

nm42 <- neuralnet(SalePrice~Buildings+LandNetSquareFeet+SquareFeet+PhysicalAge+YearBuilt, hidden = c(10,15), data=TrainingN)

plot(nm42)

#testing

testPred42 <- compute(nm42,TestingN[,c('Buildings','LandNetSquareFeet','SquareFeet','PhysicalAge','YearBuilt')])

#Calcuate prediction errors using MSE - Mean Squared Error

mse42 <- sum((testPred42$net.result- TestingN$SalePrice)^2)/nrow(TestingN)

mse42

mae42 <- sum(abs(testPred42$net.result- TestingN$SalePrice))/nrow(TestingN)

mae42

################################

actualTestPred42 <- testPred42$net.result \* (max(fulldsCluster3[,'SalePrice'])-min(fulldsCluster3[,'SalePrice'])) + min(fulldsCluster3[,'SalePrice'])

#Calculate the MSE using actual (original) values

mseActual42 <- sum((actualTestPred42- Testing [,'SalePrice'])^2)/nrow(Testing) #Calculation of mse using the original scale

mseActual42

sqrt (mseActual42)

maeActual42 <- sum(abs(actualTestPred42- Testing [,'SalePrice']))/nrow(Testing)

maeActual42

Random Forest

The following is the R code for the random forest model using the clustered data.

install.packages("randomForest")

library(randomForest)

install.packages("RODBC")

library(RODBC)

install.packages("tidyverse")

library(tidyverse)

#===================================

# model with clustered data with crime

#===================================

conn <- odbcConnect("oitap22")

dfmn <- sqlQuery(conn,"select \* from [19su5510\_tolenti4].[dbo].[FinalDataClusterWithCrime]")

apply(dfmn,2,range)

fullds <- dfmn %>% mutate(cluster = as.numeric(substr(`Cluster`,8,9))) %>%

select(-`Cluster`)

fulldsCluster1 <- fullds %>% filter(cluster ==1)

fulldsCluster2 <- fullds %>% filter(cluster ==2)

fulldsCluster3 <- fullds %>% filter(cluster ==3)

# Partition the dataset for training and testing

set.seed(1234) #A seed can be any number you like

#===================================

# Before Clustering

rs <- sample(nrow(fullds), .8\*nrow(fullds))

Training <-fullds[rs,]

Testing <- fullds[-rs,]

attach(Training)

colnames(Training)

Training[is.na(Training)] <- 0

Testing[is.na(Testing)] <- 0

dim(Training)

dim(Testing)

# Build a random forest model before clustering

rfm <- randomForest(SalePrice ~Buildings+LandNetSquareFeet+SquareFeet+Condition+Quality+PhysicalAge+YearBuilt+YearRemodeled+TaxableValueCurrentYear+TaxableValuePriorYear+DrugCrime+Homicide+PropertyCrime+PersonalCrime+OtherCrime, data = Training)

print(rfm)

# Evaluate random forest model before clustering

mse <- sum((rfm$predicted - Training$SalePrice)^2)/nrow(Training)

mse

sqrt(mse) #for comparison betwwen models (RMSE)

mae <- sum(abs((rfm$predicted - Training$SalePrice)))/nrow(Training)

mae #for comparison between models

# Evaluate random forest model before clustering (Testing)

p <- predict(rfm, Testing[,-17])

mseTest <- sum((p - Testing$SalePrice)^2)/nrow(Testing)

mseTest

sqrt(mseTest) #for comparison between models

maeTest <- sum(abs((p - Testing$SalePrice)))/nrow(Testing)

maeTest #for comparison between models

# Variable importance - Information Gain

varImpPlot(rfm)

importance(rfm)

#===================================

# Cluster 1 – with crime

rs1 <- sample(nrow(fulldsCluster1), .8\*nrow(fulldsCluster1))

Training1 <-fulldsCluster1[rs1,]

Testing1 <- fulldsCluster1[-rs1,]

attach(Training1)

colnames(Training1)

Training1[is.na(Training1)] <- 0

Testing1[is.na(Testing1)] <- 0

dim(Training1)

dim(Testing1)

# Build a random forest model with cluster 1

rfm1 <- randomForest(SalePrice ~Buildings+LandNetSquareFeet+SquareFeet+Condition+Quality+PhysicalAge+YearBuilt+YearRemodeled+TaxableValueCurrentYear+TaxableValuePriorYear+DrugCrime+Homicide+PropertyCrime+PersonalCrime+OtherCrime, data = Training1)

print(rfm1)

# Evaluate random forest model with cluster 1

mse1 <- sum((rfm1$predicted - Training1$SalePrice)^2)/nrow(Training1)

mse1

sqrt(mse1) #for comparison betwwen models (RMSE)

mae1 <- sum(abs((rfm1$predicted - Training1$SalePrice)))/nrow(Training1)

mae1 #for comparison between models

# Evaluate random forest model with cluster 1 (Testing)

p1 <- predict(rfm1, Testing1[,-17])

mseTest1 <- sum((p1 - Testing1$SalePrice)^2)/nrow(Testing1)

mseTest1

sqrt(mseTest1) #for comparison between models

maeTest1 <- sum(abs((p1 - Testing1$SalePrice)))/nrow(Testing1)

maeTest1 #for comparison between models

# Variable importance - Information Gain

varImpPlot(rfm1)

importance(rfm1)

#===================================

# cluster 2 – with crime

rs2 <- sample(nrow(fulldsCluster2), .8\*nrow(fulldsCluster2))

Training2 <-fulldsCluster2[rs2,]

Testing2 <- fulldsCluster2[-rs2,]

attach(Training2)

colnames(Training2)

Training2[is.na(Training2)] <- 0

Testing2[is.na(Testing2)] <- 0

dim(Training2)

dim(Testing2)

# Build a random forest model with cluster 2

rfm2 <- randomForest(SalePrice ~Buildings+LandNetSquareFeet+SquareFeet+Condition+Quality+PhysicalAge+YearBuilt+YearRemodeled+TaxableValueCurrentYear+TaxableValuePriorYear+DrugCrime+Homicide+PropertyCrime+PersonalCrime+OtherCrime, data = Training2)

print(rfm2)

# Evaluate random forest model with cluster 2

mse2 <- sum((rfm2$predicted - Training2$SalePrice)^2)/nrow(Training2)

mse2

sqrt(mse2) #for comparison betwwen models (RMSE)

mae2 <- sum(abs((rfm2$predicted - Training2$SalePrice)))/nrow(Training2)

mae2 #for comparison between models

# Evaluate random forest model with cluster 2 (Testing)

p2 <- predict(rfm2, Testing2[,-17])

mseTest2 <- sum((p2 - Testing2$SalePrice)^2)/nrow(Testing2)

mseTest2

sqrt(mseTest2)

maeTest2 <- sum(abs((p2 - Testing2$SalePrice)))/nrow(Testing2)

maeTest2 #for comparison between models

#Variable importance - Information Gain

varImpPlot(rfm2)

importance(rfm2)

#===================================

# Cluster 3 – with crime

rs3 <- sample(nrow(fulldsCluster3), .8\*nrow(fulldsCluster3))

Training3 <-fulldsCluster3[rs3,]

Testing3 <- fulldsCluster3[-rs3,]

attach(Training3)

colnames(Training3)

Training3[is.na(Training3)] <- 0

Testing3[is.na(Testing3)] <- 0

dim(Training3)

dim(Testing3)

# Build a random forest model with cluster 3

rfm3 <- randomForest(SalePrice ~Buildings+LandNetSquareFeet+SquareFeet+Condition+Quality+PhysicalAge+YearBuilt+YearRemodeled+TaxableValueCurrentYear+TaxableValuePriorYear+DrugCrime+Homicide+PropertyCrime+PersonalCrime+OtherCrime, data = Training3)

print(rfm3)

# Evaluate random forest model with cluster 3

mse3 <- sum((rfm3$predicted - Training3$SalePrice)^2)/nrow(Training3)

mse3

sqrt(mse3) #for comparison betwwen models (RMSE)

mae3 <- sum(abs((rfm3$predicted - Training3$SalePrice)))/nrow(Training3)

mae3 #for comparison between models

# Evaluate random forest model with cluster 3 (Testing)

p3 <- predict(rfm3, Testing3[,-17])

mseTest3 <- sum((p3 - Testing3$SalePrice)^2)/nrow(Testing3)

mseTest3

sqrt(mseTest3)

maeTest3 <- sum(abs((p3 - Testing3$SalePrice)))/nrow(Testing3)

maeTest3 #for comparison between models

#Variable importance - Information Gain

varImpPlot(rfm3)

importance(rfm3)

odbcClose(conn)

#===================================

# model with clustered data without crime

#===================================

conn <- odbcConnect("oitap22")

dfmn\_wo <- sqlQuery(conn,"select \* from [19su5510\_tolenti4].[dbo].[FinalDataClusterWithoutCrime]")

apply(dfmn\_wo,2,range)

fullds <- dfmn\_wo %>% select(-PersonalCrime, -DrugCrime, -OtherCrime, -Homicide, -PropertyCrime) %>%

mutate(cluster = as.numeric(substr(`Cluster`,8,9))) %>%

select(-`Cluster`)

fulldsCluster1 <- fullds %>% filter(cluster ==1)

fulldsCluster2 <- fullds %>% filter(cluster ==2)

fulldsCluster3 <- fullds %>% filter(cluster ==3)

# Partition the dataset for training and testing

set.seed(1234) #A seed can be any number you like

#===================================

# Before Clustering

rs <- sample(nrow(fullds), .8\*nrow(fullds))

Training <-fullds[rs,]

Testing <- fullds[-rs,]

attach(Training)

colnames(Training)

Training[is.na(Training)] <- 0

Testing[is.na(Testing)] <- 0

dim(Training)

dim(Testing)

# Build a random forest model before clustering

rfm <- randomForest(SalePrice ~Buildings+LandNetSquareFeet+SquareFeet+Condition+Quality+PhysicalAge+YearBuilt+YearRemodeled+TaxableValueCurrentYear+TaxableValuePriorYear, data = Training)

print(rfm)

# Evaluate random forest model before clustering

mse <- sum((rfm$predicted - Training$SalePrice)^2)/nrow(Training)

mse

sqrt(mse) #for comparison betwwen models (RMSE)

mae <- sum(abs((rfm$predicted - Training$SalePrice)))/nrow(Training)

mae #for comparison between models

# Evaluate random forest model before clustering (Testing)

p <- predict(rfm, Testing[,-12])

mseTest <- sum((p - Testing$SalePrice)^2)/nrow(Testing)

mseTest

sqrt(mseTest) #for comparison between models

maeTest <- sum(abs((p - Testing$SalePrice)))/nrow(Testing)

maeTest #for comparison between models

# Variable importance - Information Gain

varImpPlot(rfm)

importance(rfm)

#===================================

# Cluster 1 – without crime

rs1 <- sample(nrow(fulldsCluster1), .8\*nrow(fulldsCluster1))

Training1 <-fulldsCluster1[rs1,]

Testing1 <- fulldsCluster1[-rs1,]

attach(Training1)

colnames(Training1)

Training1[is.na(Training1)] <- 0

Testing1[is.na(Testing1)] <- 0

dim(Training1)

dim(Testing1)

# Build a random forest model with cluster 1

rfm1 <- randomForest(SalePrice ~Buildings+LandNetSquareFeet+SquareFeet+Condition+Quality+PhysicalAge+YearBuilt+YearRemodeled+TaxableValueCurrentYear+TaxableValuePriorYear, data = Training1)

print(rfm1)

# Evaluate random forest model with cluster 1

mse1 <- sum((rfm1$predicted - Training1$SalePrice)^2)/nrow(Training1)

mse1

sqrt(mse1) #for comparison betwwen models (RMSE)

mae1 <- sum(abs((rfm1$predicted - Training1$SalePrice)))/nrow(Training1)

mae1 #for comparison between models

# Evaluate random forest model with cluster 1 (Testing)

p1 <- predict(rfm1, Testing1[,-12])

mseTest1 <- sum((p1 - Testing1$SalePrice)^2)/nrow(Testing1)

mseTest1

sqrt(mseTest1)

maeTest1 <- sum(abs((p1 - Testing1$SalePrice)))/nrow(Testing1)

maeTest1 #for comparison between models

# Variable importance - Information Gain

varImpPlot(rfm1)

importance(rfm1)

#===================================

# Cluster 2 – without crime

rs2 <- sample(nrow(fulldsCluster2), .8\*nrow(fulldsCluster2))

Training2 <-fulldsCluster2[rs2,]

Testing2 <- fulldsCluster2[-rs2,]

attach(Training2)

colnames(Training2)

Training2[is.na(Training2)] <- 0

Testing2[is.na(Testing2)] <- 0

dim(Training2)

dim(Testing2)

# Build a random forest model with cluster 2

rfm2 <- randomForest(SalePrice ~Buildings+LandNetSquareFeet+SquareFeet+Condition+Quality+PhysicalAge+YearBuilt+YearRemodeled+TaxableValueCurrentYear+TaxableValuePriorYear, data = Training2)

print(rfm2)

# Evaluate random forest model with cluster 2

mse2 <- sum((rfm2$predicted - Training2$SalePrice)^2)/nrow(Training2)

mse2

sqrt(mse2)

mae2 <- sum(abs((rfm2$predicted - Training2$SalePrice)))/nrow(Training2)

mae2 #for comparison between models

# Evaluate random forest model with cluster 2 (Testing)

p2 <- predict(rfm2, Testing2[,-12])

mseTest2 <- sum((p2 - Testing2$SalePrice)^2)/nrow(Testing2)

mseTest2

sqrt(mseTest2)

maeTest2 <- sum(abs((p2 - Testing2$SalePrice)))/nrow(Testing2)

maeTest2 #for comparison between models

# Variable importance - Information Gain

varImpPlot(rfm2)

importance(rfm2)

#===================================

# Cluster 3 – without crime

rs3 <- sample(nrow(fulldsCluster3), .8\*nrow(fulldsCluster3))

Training3 <-fulldsCluster3[rs3,]

Testing3 <- fulldsCluster3[-rs3,]

attach(Training3)

colnames(Training3)

Training3[is.na(Training3)] <- 0

Testing3[is.na(Testing3)] <- 0

dim(Training3)

dim(Testing3)

# Build a random forest model with cluster 3

rfm3 <- randomForest(SalePrice ~Buildings+LandNetSquareFeet+SquareFeet+Condition+Quality+PhysicalAge+YearBuilt+YearRemodeled+TaxableValueCurrentYear+TaxableValuePriorYear, data = Training3)

print(rfm3)

# Evaluate random forest model with cluster 3

mse3 <- sum((rfm3$predicted - Training3$SalePrice)^2)/nrow(Training3)

mse3

sqrt(mse3)

mae3 <- sum(abs((rfm3$predicted - Training3$SalePrice)))/nrow(Training3)

mae3 #for comparison between models

# Evaluate random forest model with cluster 3 (Testing)

p3 <- predict(rfm3, Testing3[,-12])

mseTest3 <- sum((p3 - Testing3$SalePrice)^2)/nrow(Testing3)

mseTest3

sqrt(mseTest3)

maeTest3 <- sum(abs((p3 - Testing3$SalePrice)))/nrow(Testing3)

maeTest3 #for comparison between models

# Variable importance - Information Gain

varImpPlot(rfm3)

importance(rfm3)

odbcClose(conn)