MA540	**Crime Rate Classification Model Evaluation**		**SUBMISSION 001**
Group Name:			
Instructor:	Dr. Hong Liu	Date:	05/04/2020

Abstract:

This report pertains to evaluate the process of extracting, visualizing, transforming, and loading historical crime data within Volusia County (VC) with the purpose of building a classification model to predict whether a block group would have a high or low crime rate based on its unique features. In addition, the report addresses the process used to source, cleanse, and clean the data as to provide insightful information about each data attribute.

In total, four models were created: Naïve Bayes, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (K-NN) models were designed in Python to predict the class type of a block group. This report also evaluates the design, tuning, and quality of these classification models.

Additional Related Documents

Doc Number	Description	Date	Author
MA540_Final_Project.pdf			
MA540_Crime_Rate_Classification.pptx			

Revisions

Revision	Description	Date	Author
A	Initial release	05/04/2021	All

Member Name	Model
Ross Dickinson	Naïve Bayes
Diogo Cobos	Random Forest
Alan Hall	Visualizations
Dennis Moreno	Support Vector Machine
Juan Leon	K-Nearest Neighbors

General Terms

AUC	Area Under Curve
BG	Block Group
DUI	Driving Under the Influence
FRP	False Positive Rate
K-NN	K-Nearest Neighbors
ROC	Receiver Operating Company
SVM	Support Vector Machine
TPR	True Positive Rate
UCR	Uniform Crime Reporting
VC	Volusia County

1. Introduction

The purpose of this project was to collect real-world crime data to predict and classify whether a parcel (house/building), is located in a low or high crime rate area based on its distinctive combination of attribute values. For this project, Volusia County, Florida, was selected as the region of investigation. Volusia county has in total 29 zip codes which are divided further into 114 *Tracts*. However, tracts are still relatively large regions that contains different demographics which increases noise in the data, preventing patterns to be found. It is crucial that the areas in the dataset contain parcels which are similar in the chosen attributes. For this reason, the group decided to use data collected by the United States Census Bureau, which defines even smaller geographic areas, called *Block Groups*, within each tract. This allows for more detailed and specific analysis in comparison to zip codes and tracts themselves, increasing the chance of success of the classifier. *Figure 1.1* below depicts the exact definition for each division within a county. As previously stated, the group decided to focus on Block Groups (BG), which could still be further divided into *Blocks* if needed.

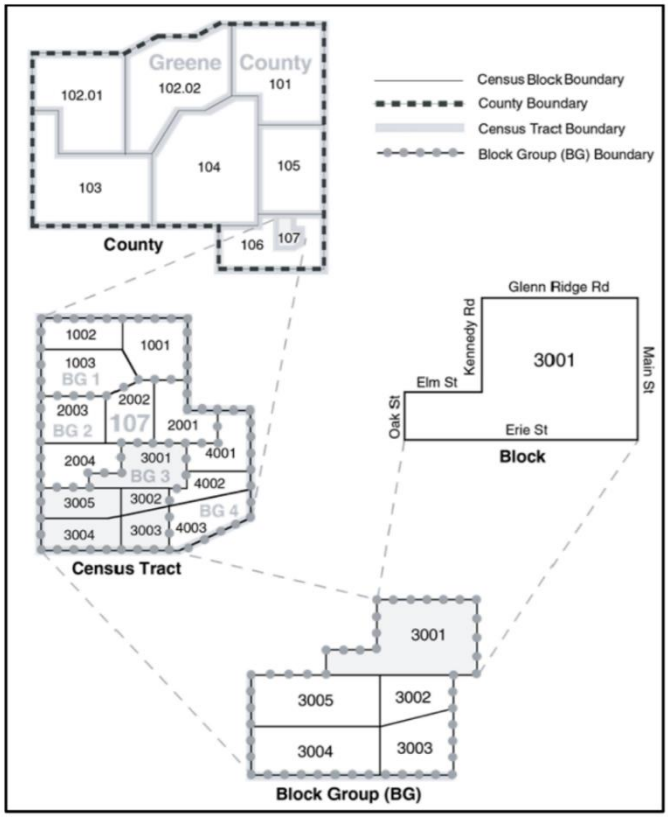


Figure 1-1: Tract and Block group characterization as defined by the US Census Bureau

Tracts and Block Groups for Volusia county can be seen in the images below where tracts are delimited by black lines while Block Groups borders are shown through purple lines. The United State Census Bureau provides different data for different geographic levels. Therefore, attributes given at Block Group level such as *Household Income*, were assumed to be the same for all parcels within that BG, while the population, which was given at Block level, was simply calculated by adding the population of each individual Block contained in a BG.

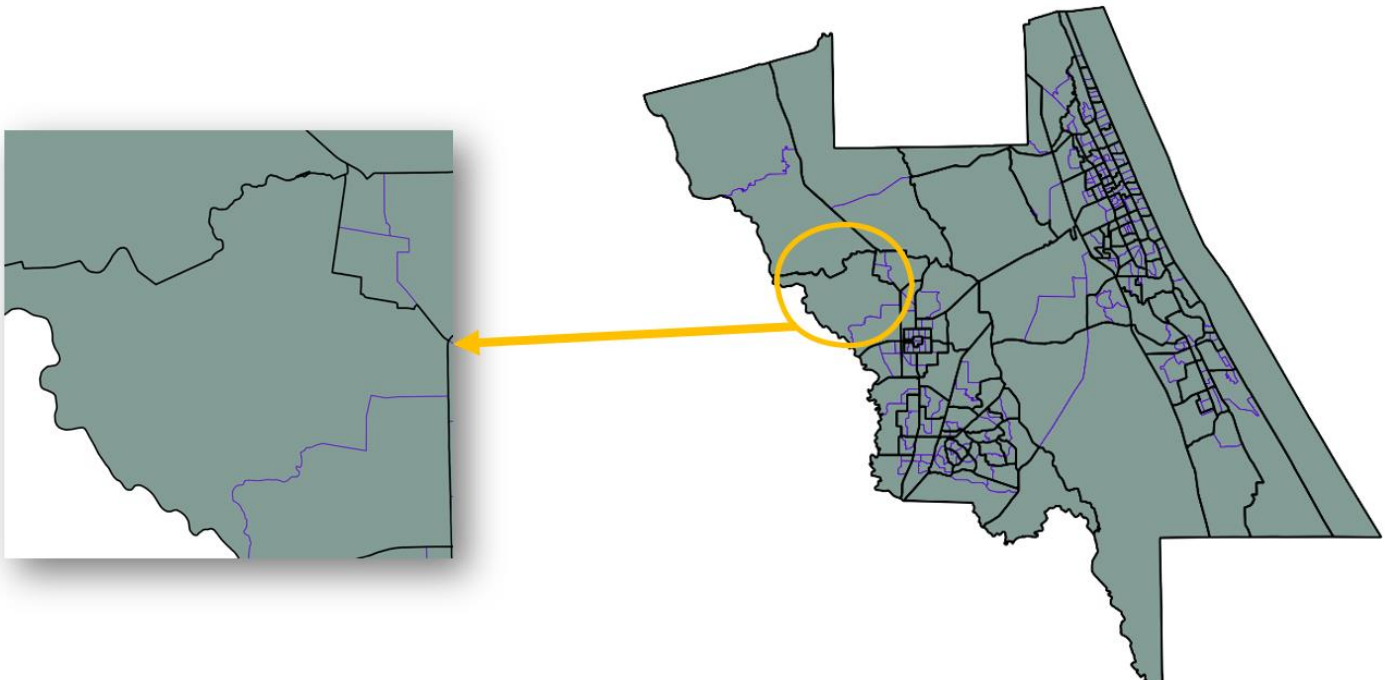


Figure 1-2: Volusia county geographic divisions for Tract and Block groups

2. Dataset

2.1. Collection

In order to establish high quality prediction models, it was imperative that informative data was selected as attributes. As suggested by the Uniform Crime Reporting (UCR) program, population density and degree of urbanization were major factors considered due to their high correlation with criminality rates. These were the first two "broad" attributes to be selected which were then split into simpler attributes that could easily be quantified such as *total population, average household income,* and *number of housing units* for a block group. Demographic data of Volusia county was acquired through the US Census Bureau which provides complete reports on the population very 10 years. As this is a government regulated data source, it can be trusted to be well monitored and managed. For this project, data from 2010 was utilized. Additional information deemed crucial to predict crime rates were factors related to the parcels themselves such as *living area*, *aprtot, number of parcels,* and *number of bedrooms*. This data was acquired through the already ongoing real estate research conducted by Dr. Steven Lehr, from Embry-Riddle Aeronautical University,

and through the Volusia County Property Appraisal database which constantly updates details about all parcels within the county.

Supplementarily, the objective of the project was to predict the crime rate based on the attributes mentioned previously, hence, it was essential to gather accurate and realistic data with respect to law violations in Volusia county. For that reason, six months' worth of crime data was collected from the Volusia Sheriff's department. In acquiring the data, however, it was observed that the types of crimes vary from driving under the influence (DUI) to homicide. For the purpose of this project, it was necessary to remove trivial crimes such as DUI since these are not strong indicators of unsafe areas as they occur everywhere. Crimes such as homicide, burglary, and assaults provide much more insight as to behavior and danger presented in a block group and are more often committed by locals.

2.2. Cleansing

Six months of data were acquired and resulted in 8068 crime instances, each containing the type of crime, code, address, department which attended to it, and date. After removing crimes such as DUI, domestic disturbance and trespassing, around 6000 occurrences were left in the dataset.
In order to map these crimes, a Python script was developed to translate addresses and police departments into geographical coordinates for longitude and latitude. This process was successful in 80%> of the cases, however, crimes that occurred in regions defined as "Volusia County Sheriff" could not have their exact location found considering that only the street name was provided and not the city. Regions defined as "Volusia County Sheriff" are usually lowly populated regions in between cities. For this reason, all block groups that had a total count of *zero* crimes had to be dropped from the analysis since they became noisy data. It is important to reinforce that all these regions *zero* crimes due to a software and information limitation, however, it does not mean no crimes happened within their borders. In further studies, collecting further details on these regions might help improve classification.

In regards to the attributes of the dataset, much of the cleansing was conducted alongside the collection of the data. Although most building are considered parcels, this project focused on parcels related to housing such as *Single-Family* housing and *Condominiums*. Furthermore, due to the small number of attributes chosen for this project, each attribute largely contributes to the model's classification process. For this reason, any rows that contained *null* values were automatically dropped. While only few instances matched such criteria, this step was crucial in maintaining the pattern with the data, allowing for a better predictor. At the end of the cleansing procedure there were around 200 instances, one for each block group in Volusia county, and all of them had a non-zero number of crime incidents to prevent any misrepresentation of real factors in play.

3. Data Exploration

3.1. Attributes and Class Definition

Many of the attributes used in this project have already been discussed. However, it is also crucial to understand their individual characteristics and their relationship with one another as the first step to an accurate prediction is a full comprehension of the dataset. All features utilized were numerical and they all represent either averages or total counts for an entire block group of Volusia county.

Attribute	Data Type	Abbreviation
Average Square-footage of living area	Numeric	*avg_sfla*
Average appraised total value	Numeric	*avg_aprtot*
Average selling price	Numeric	*avg_price*
Average number of bedrooms	Numeric	*avg_rmbed*
Total number of parcel units	Numeric	*nbr_parcels*
Total number of housing units	Numeric	*nbr_housing*
Total population	Numeric	*pop*
Average household income	Numeric	*avg_hh_income*

Table 3-1: Attribute definitions

In order to define the predicted labels (classes), the group's solution was to use the median of the data. This is a simple approach that allowed for a very balanced set, preventing bias in the model. The total count and class split information are shown in *Table 2* and *Figure 2*. In total, the dataset contained 103 instances where crime rate was labeled "HIGH" and 98 instances labeled as "LOW".

After instances with *zero* crimes were removed, the dataset was divided almost perfectly in half at 19 crimes, meaning the half of the Block Groups had a total count of crime occurrences equal or larger to 19, and the other half of the instances possessed a total crime count less than. For this reason, 19 was selected as the threshold for class labels.

Class	Count	Range
HIGH	103	>= 19 Crimes
LOW	98	< 19 Crimes

Table 3-2: Class count and threshold

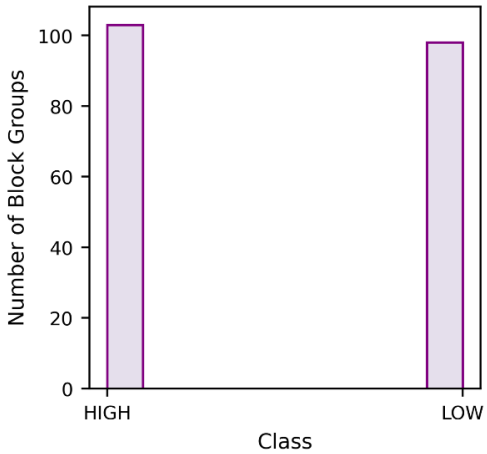


Figure 3-1: Graphical representation of balance between classes

3.2. Crime Visualization

An important part of feature selection is visualization, which allows for quick interpretations of the dataset which are simple to understand, providing good overall relationship between each attribute. After collecting the crime data from the sheriff's department and translating addresses into coordinates, the first step taken by the group was to plot points in such coordinates and analyze the outcome. As expected, regions more developed and therefore more populated had a much higher criminal manifestation as *Figure 3.2* demonstrates.

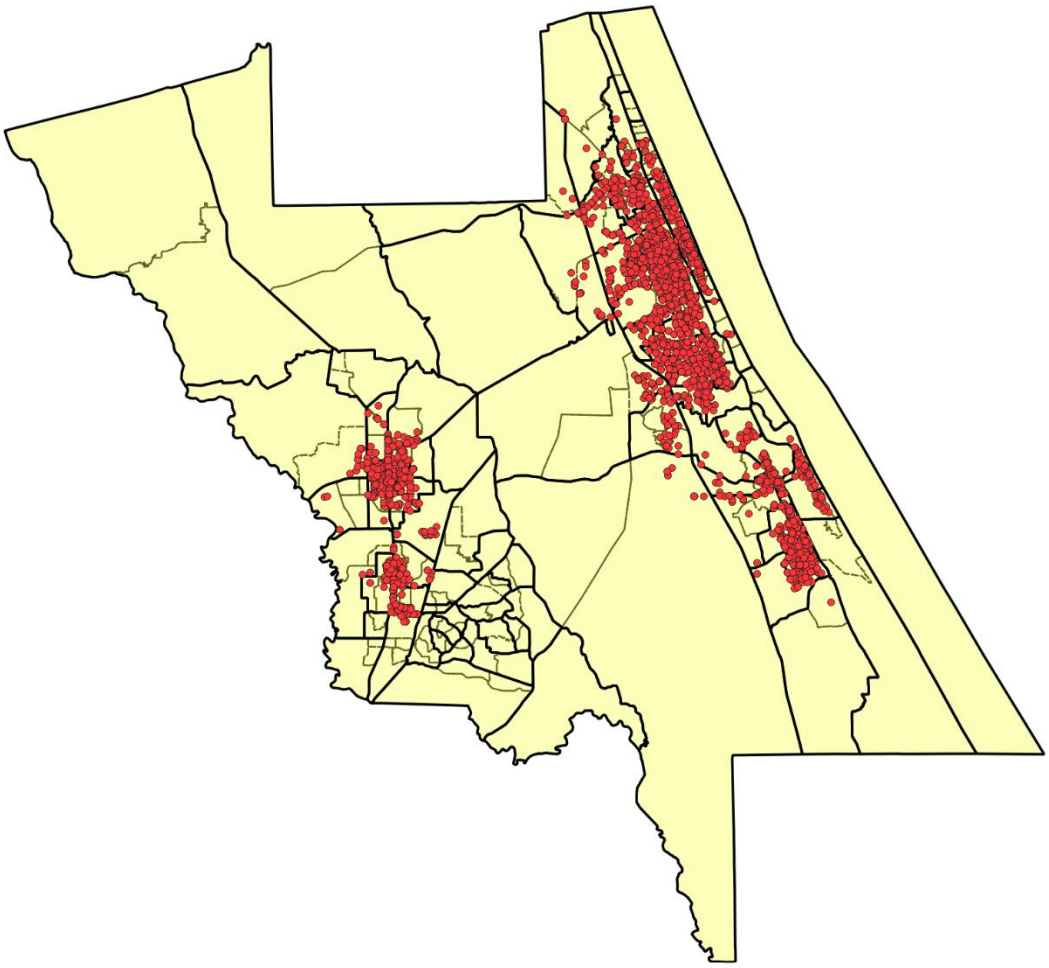


Figure 3-2: Crime occurrences in Volusia county from April 9, 2021 to October 13, 2020.

The image above represents around 5500 crimes that occurred in Volusia county during the period of October 13, 2020 to April 9, 2021. As explained previously, crimes which were reported by the Volusia county sheriff's departments could not have their exact location determined due to software limitations, resulting in areas absent of crimes. However, this was not deemed an issue since areas with larger population contain more housing units and are therefore, more relevant to the project.

3.3. Data visualization

The following visualizations is what was used to decide on model types. Then utilized the hypotheses we gained from the visualizations to train our models. The visualizations were created using Tableau® and RStudio®.

Figure 3-3: Urban/Suburbia comparison of crime distribution.

The above illustration is the first clue into what the data is trying to tell us. Once zoomed into the clustered areas from Figure 3-2 we can see smaller clusters forming. Notice the higher crime density in the small block shaped urban areas versus the curvy roads of the suburbs.

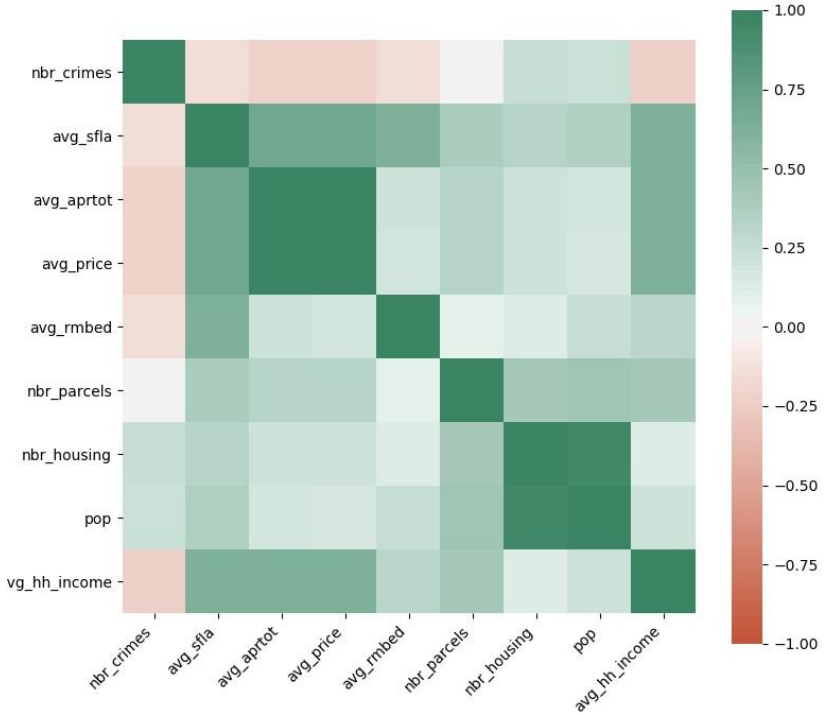


Figure 3-4: Correlation between attributes. Nbr_crimes was used to define the class of the instance (High or Low)

Figure 3-4 shows possible correlations to exploit with our models. This allows for a further investigation of each attribute compared to the number of crimes.

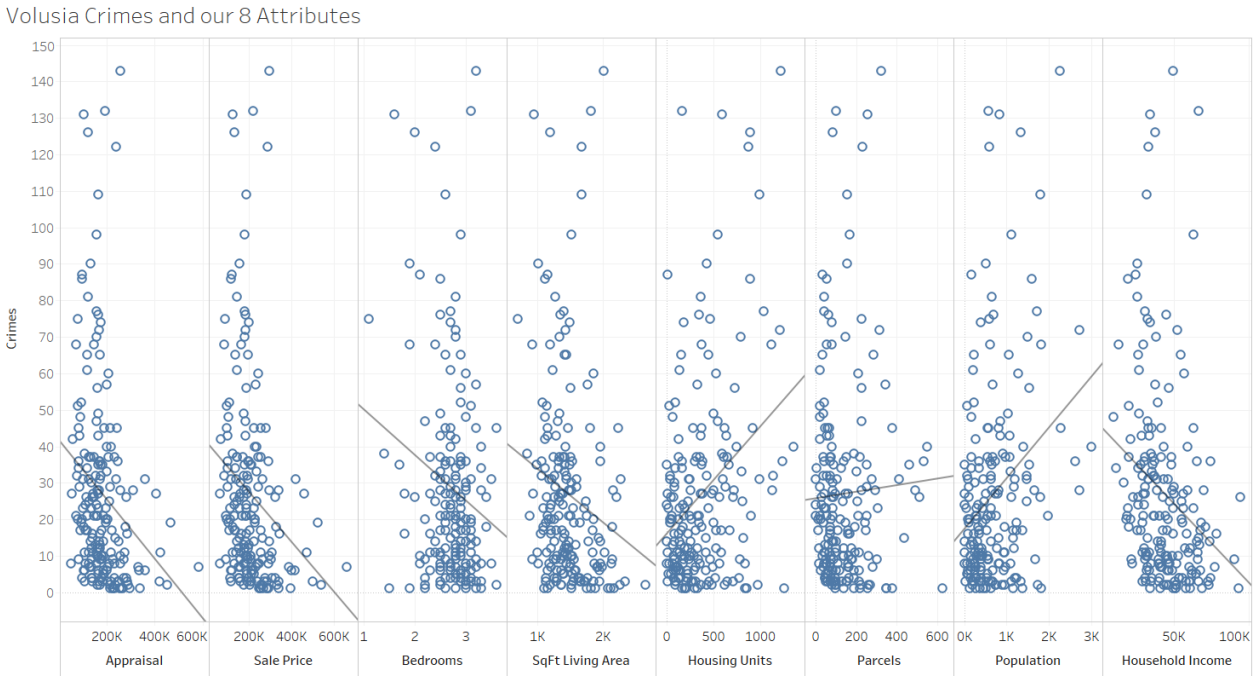


Figure 3-5: Regression of the 8 attributes and the number of crimes.

The above image shows all the attributes have some correlation to the number of crimes. All 8 have a P-value of less than 5%, except number of parcels. With all but number of bedrooms and square foot living area having less than 1% P-value. The strongest correlation was found in housing units and population. Note: these two attributes are also strongly correlated to each other.

Figure 3-6: Histogram of each attribute.

Figure 3-6 shows the distribution of High/Low crime among each attribute. Some strong indicators of crime rate can be seen among the attributes. However, most are heavily skewed. To better see the distributions, we can use a log base 10 transformation on our data set.

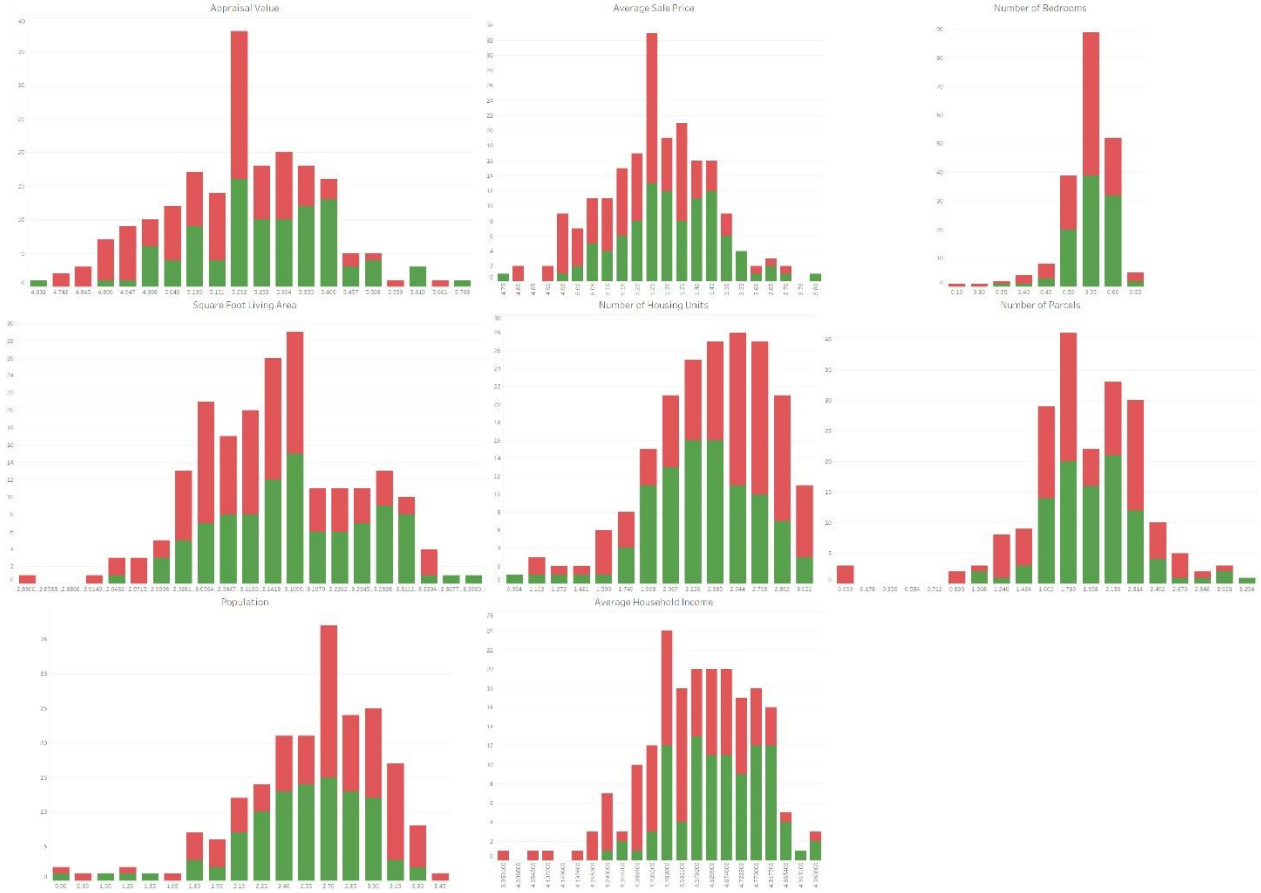


Figure 3-7: Log base 10 transformation histogram of all 8 attributes.

After the log transformation the disparities in crime rate among the attributes can better be seen. This gives a great indication as to where to begin our models. Different peak distributions can clearly be seen in number of homes. This implies number of homes may be the strongest decision node.

4. Model Selection and Development

The main objective of this project is to allow people to predict crime rate in the region of a specified parcel as a method of assistance when selecting a property to purchase. Although obtaining an accuracy of 100% is always the ideal, the amount of complexity combined with the small dataset were two obstacles which restrained the model from achieving such results. Hence, the group decided to focus on other metrics to determine the success of the models.

Firstly, instead of focusing on the total accuracy of the model, the recall was closely looked at. Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds. It is more important to detect areas of high danger than to detect low risk areas. In other words, when purchasing a property, if the model erroneously classifies a region of low crime rate as high, the consequences are not as severe as the opposite happening.

In addition to recall scores, Receiver Operating Characteristic Curves (ROC Curve) summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds. ROC curves are appropriate when the observations are balanced between each class, whereas precision-recall curves are also appropriate for imbalanced datasets.

Another possible metric to be used to ascertain the success of a model is the Cohen's Kappa score. This metric measures the level of agreement between two different raters, basically comparing confusion matrices and measuring their agreement. The Kappa score is an important benchmark to determine the improvement of a model over randomly assigned classes.

4.1. Naïve Bayes Classifier

A Naïve Bayes classifier uses posterior probability to make predictions. It assumes that the attributes within the dataset are independent. However, it is virtually impossible to attain true attribute independence in real-world data, Naïve Bayes has been proven to be an extremely effective and efficient prediction method despite its simplicity when compared to more sophisticated classifiers.

The main benefits of using this type of model were its ease of implementation and processing speed. Where independence among data attributes is true, Naïve Bayes classifiers generally outperform other models such as logistic regression. Another benefit of this model is that it does not require ample data to get excellent results. Lastly, it handles categorical data types well assuming that they are independent of one another.

It was important to avoid having any categorical data types in the test data set but not featured in the training data. If this were to occur, then the model would not have a posterior probability for that particular type of category. In this case, with the crime data frame, there were only numeric

data types and such there was no need to address this problem. However, if there were cases of "Zero Frequency" categorical types then the data would need to be smoothed. The most substantial flaw of the Naïve Bayes method is that it requires independence about the data attributes. This is almost impossible to find such data sets in real life. The data frame was split using a 80/20 ratio for the training data and testing data, respectively. The model was then fit using the training data. The model was then used to predict the class type using the test attribute values.

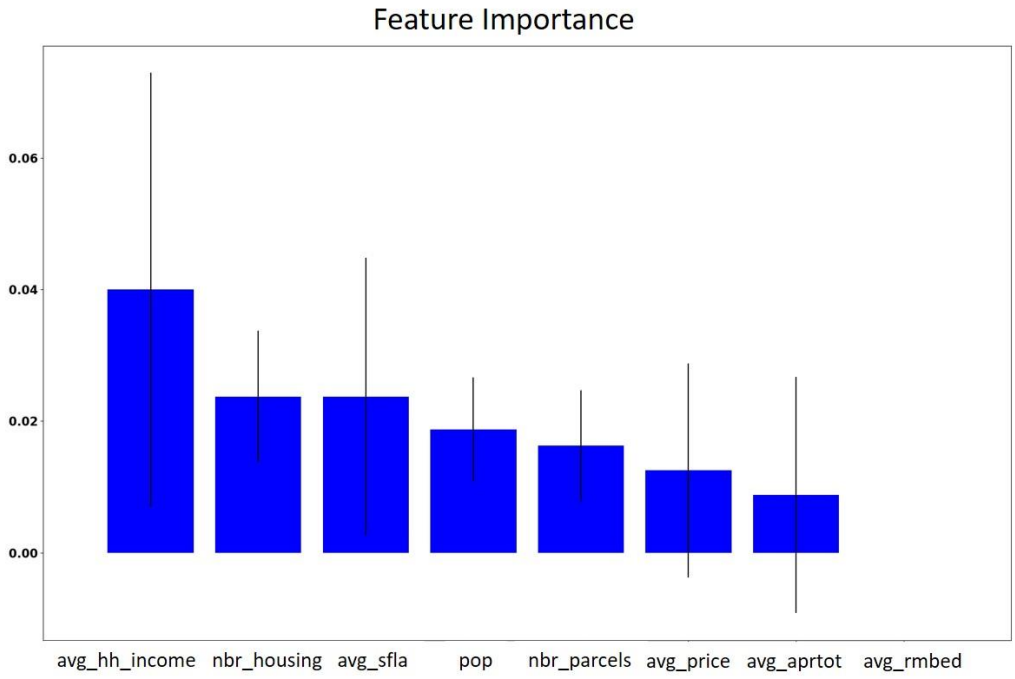


Figure 4-1: Naïve Bayes permutation feature importance

Overall, the model was reasonably accurate and surpassed all other classifiers used. It achieved a recall of 0.84 and 0.64 for the high and low classes, respectively. Recall measures the proportion of actual positives that were predicted properly. For the case of crime rate classification this is one of the most important metrics to measure because home buyers will want to ensure that they are purchasing a home in a "safe" location.

	Precision	**Recall**	**F1-Score**
High	0.64	0.84	0.72
Low	0.73	0.47	0.57
Accuracy			0.66
Macro average	0.68	0.66	0.65
Weighted average	0.68	0.66	0.65

Table 4-1: Metrics scores for Naive Bayes model.

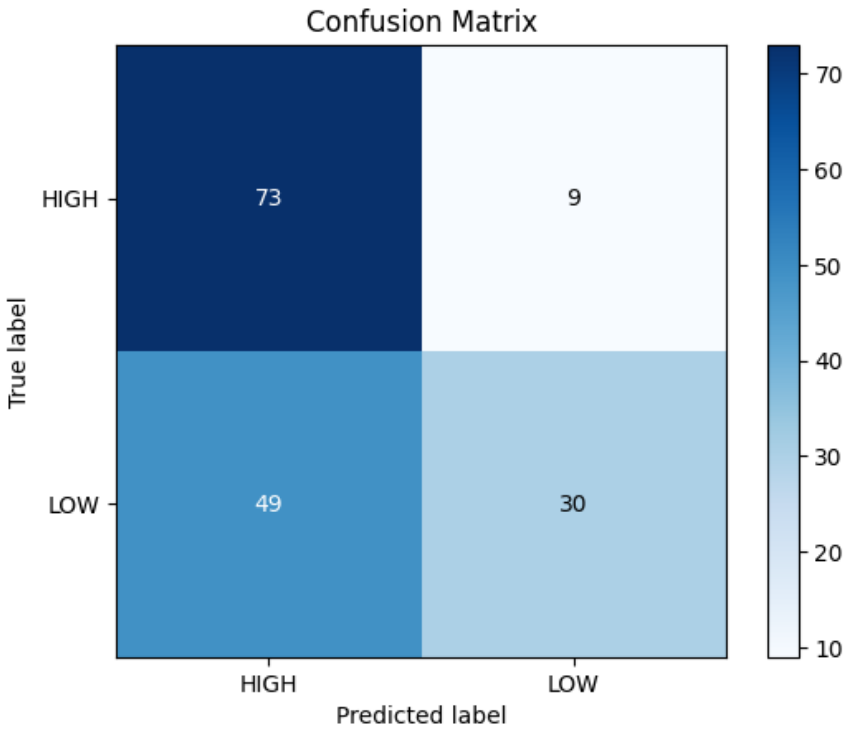


Figure 4-2: Confusion matric for Naive Bayes model.

Another important measurement was the area under the ROC curve, otherwise known as the AUC. The ROC curve demonstrates the ability of a binary classifier as its discrimination threshold is varied [1]. The AUC is a measure of the number of predictions that were made correctly by the model and is a valuable comparison of models because it is scale-invariant which evaluates how predictions are ranked rather than the absolute values of those predictions. The ROC AUC score for the Naïve Bayes model was 0.7205 and plot can be seen below. The diagonal white line represents the baseline for the classification.

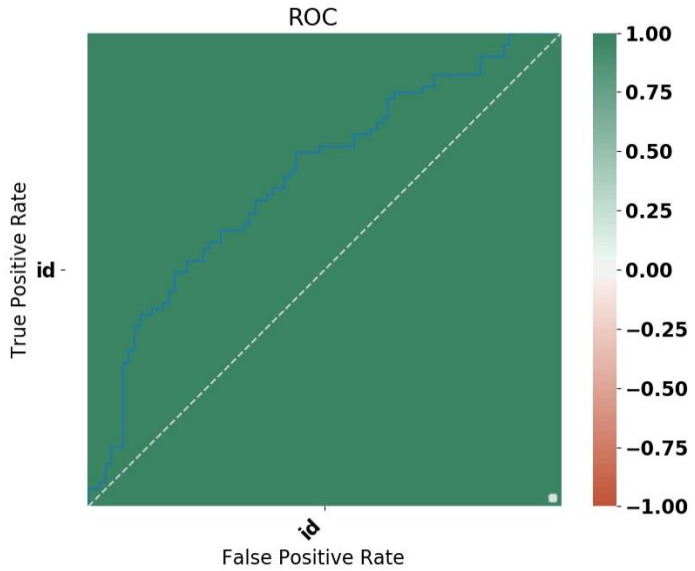


Figure 4-3: Naive Bayes ROC AUC plot.

4.2. Random Forest Classifier

Similarly to decision trees, random forests are a comprehensible supervised learning method, however, it is also considered an ensemble algorithm and thus, possesses a few extra steps which help increase classification accuracy. Random forests rely on several small decision trees which generate predictions based on a subsample of attributes that were selected. In a binary classification such as the one for this project, the label with more "votes" is the one chosen as the final prediction. Votes are nothing but the prediction generated by each subtree. One of the most important hyper-parameters of random forests is the maximum depth of each subtree.

This hyper-parameter is essential because one of the advantages of random forests over simple decision tree algorithms is that it has a low risk of overfitting. However, increasing the maximum depth of each subtree can increase that risk, and therefore, tuning such hyper-parameters is crucial in order to develop the best model possible. Apart from overfitting robustness, random forest was chosen as one of the algorithms for this project also due to it not being easily influenced by outliers and due to its versatility to work well with not only categorical data, but also numerical. Additionally, random forests perform well with liner or non-linear relationships and since many attributes in the dataset do not have a simple linear correlation with crime rate, this was an important trait.

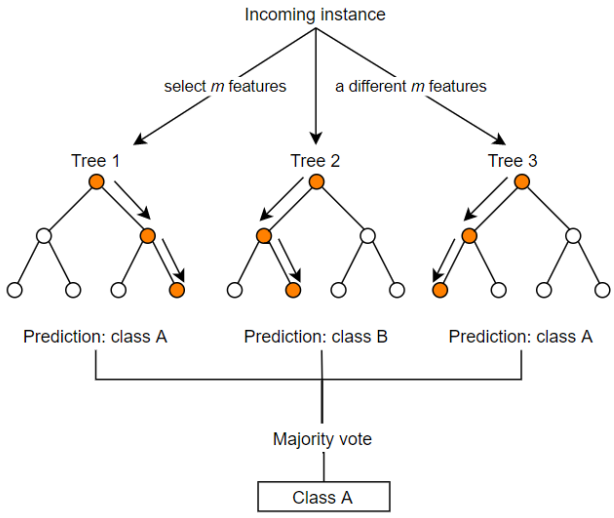
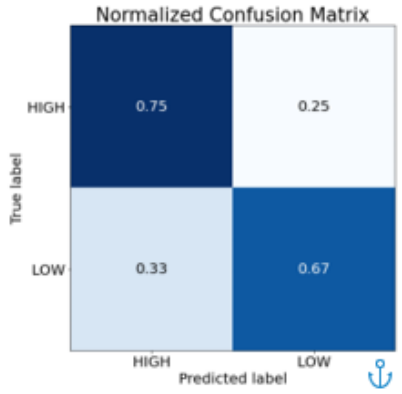


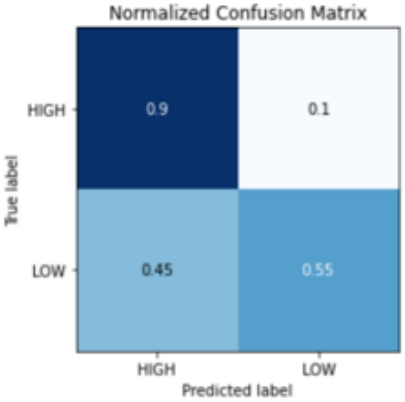
Figure 4-4: Basic visualization of a Random Forest model [3]

Perhaps the most important idea of random forest classifiers is its ability to perform *bootsrapping*. During training, each subtree created by the algorithm only learns from a random sample of the data points, however, these sample are drawn with a replacement. This means that one instance might be used several times in a subtree which in return decreases the overall variance of the model without increasing bias. All these traits mentioned are important when dealing with a particularly small dataset such as the one for this project.

The random forest model was trained on 90% of the dataset and tested on the remaining portion. Initial results were poor in both accuracy and recall for the "high" class and for this reason, cross-validation and hyper-parameter search was conducted through scikit-learn's *GridSearchCV* module. After a series of hyper-parameter tuning, the best parameters had some variation depending since not all runs of the model training will result in the same end model. In general, however, the model's recall for the "high" label remained close to 0.75, even reaching 0.90 in one of the training cycles, while accuracy stay closed to 0.70. Results for precision and recall of two random forest models generated can be seen in the following figures and tables.



	Precision	Recall
High	0.75	0.75
Low	0.67	0.67
Accuracy	0.71	
Macro average	0.71	0.71
Weighted average	0.71	0.71



	Precision	Recall
High	0.64	0.90
Low	0.86	0.55
Accuracy	0.71	
Macro average	0.75	0.72
Weighted average	0.76	0.71

Figure 2-5: Precision, recall and accuracy for two different trained Random Forest models

The values for recall and precision, especially for the "high" class, were impressive and better than expected for this particular algorithm. Once more, to further analyze the success of the model, an ROC curve was built (*figure 4-5*), to compare the False Positive Rate (FPR) and the True Positive Rate (TPR) of the model. The computed area under the curve (AUC) was 0.861.

Similarly, to recall, the results for the AUC proved the model to be quite valuable as the best possible result is an AUC score of 1.0, where the False Positive Rate is *zero* and the True Positive Rate is 1, meaning that all instances have been correctly classified. For this model, which had and a ROC AUC score of 0.861 as already mentioned, the optimal threshold value would be 0.4594 as it provides the best trade-off between FPR (0.25) and TPR (1.0).

Furthermore, the model was evaluated based on Cohen's Kappa statistic, a performance metric which ranges from -1 to 1, as 1 being the best result indicating a perfect model.

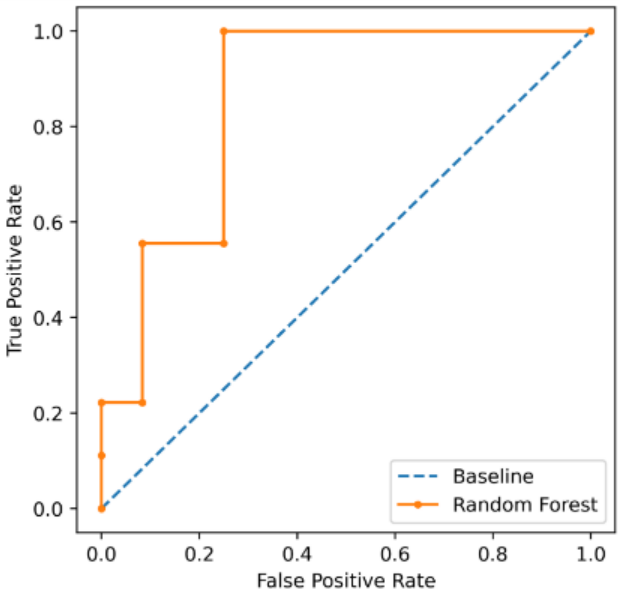


Figure 4-6: ROC AUC for Random Forest model.

Although the kappa statistic is important in imbalanced datasets in order to prevent bias when determining the performance of a model, it is still a useful metric since it provides an insight in the improvement of the classification when compared to the overall accuracy of random guesses. The random forest model resulted in a kappa score of 0.416, which according to Anthony J. Viera [4] is considered a *fair* to *moderate* value, meaning that the random forest model generated decent improvement over random classification.

Hyper-Parameter	Value
Max_depth	110
Max_features	2
Min_samples_leaf	4
Min_samples_split	9
N_estimators	30

Table 4-2: Hyper-parameters that resulted in ROC AUC depicted in figure 4-5.

4.3. Support Vector Machine Classifier

Support Vector Machine, SVM, is a machine learning algorithm which works well for small and/or medium size datasets. It is very useful to perform non-linear or linear classification, regression, and it can also identify outliers. The scope of the SVM is to classify two or more classes by drawing a line between those classes. Since the line is primarily use to separate classes, it means that is also allows prediction of future data based on where the new point lies. When drawing the line for SVM, each margin should be touching a point which represent each class and the idea is to add a line which contains the widest margins (soft margins), meaning that the classification has been optimized for that specific dataset. As shown on *Figure 4-7* the points that lie on the two margins that are away from the center line based on a distance d2 are known as support vectors and that solid or center line is also known as hyperplane because on SVM we usually work with more than two dimensions.

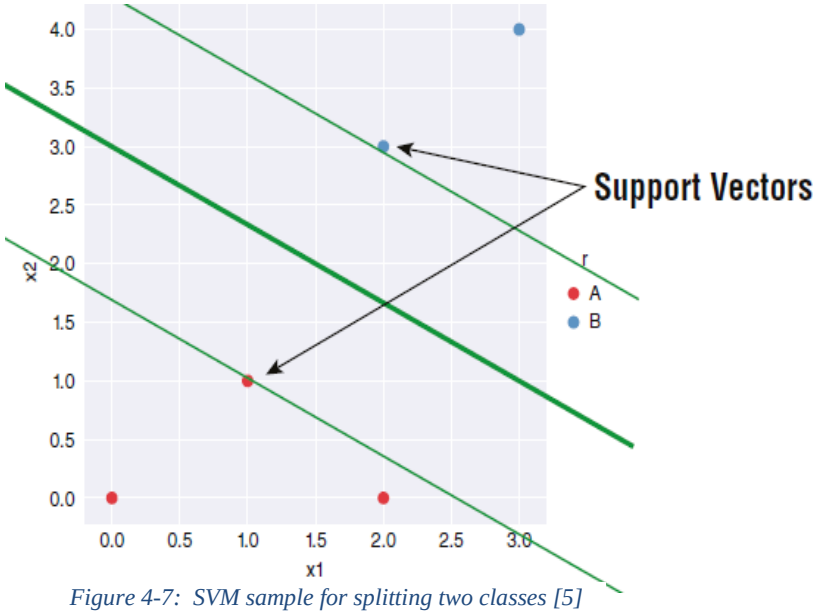


Figure 4-7: SVM sample for splitting two classes [5]

For this model, all attributes shown in figure 3-3 were used to help achieve the best classification accuracy as possible and a linear kernel was used to train the SVM algorithm. Furthermore, the hyper-parameter C is used as a penalty factor when building and SVM line or hyperplane and is,

therefore, a crucial aspect of the model. In simple terms, this hyper-parameter provides a "punishment" for each misclassification where large values of C will result in smaller softer margins to avoid misclassification.

	Precision	Recall	F1-Score
High	0.69	0.86	0.77
Low	0.80	0.60	0.69
Accuracy			0.73
Macro average	0.75	0.73	0.72
Weighted average	0.74	0.73	0.73

Table 4-3: Metrics classification scores for Support Vector Machine

As seen on *table 4-3* and *Figure 4-8* the accuracy of the SVM model is 0.73 and the recall for HIGH crime is 0.7 and for LOW crime is 0.79 meaning that "LOW" labels were correctly classified 79% of the time while "HIGH" label only 70%. To achieve these results the 80% of the dataset was used in training and the other 20% was used as the test dataset. It is important to note that different hyper-parameters were used until the optimal solution was met.

The ROC curve on figure 4-9 shows the performance of the SVM model. The ROC AUC score for the Support Vector Machine model was 0.828 which is a measure of correct predictions that were made. The diagonal dotted blue line represents the baseline for the classification.

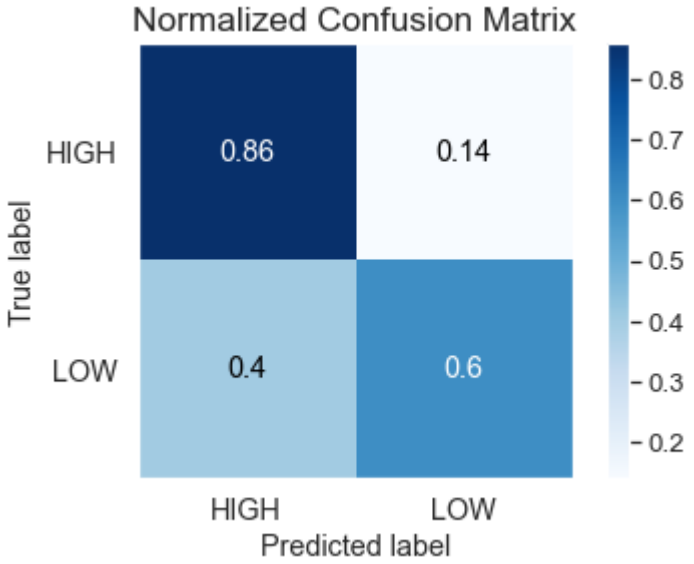


Figure 4-8: SVM confusion matrix

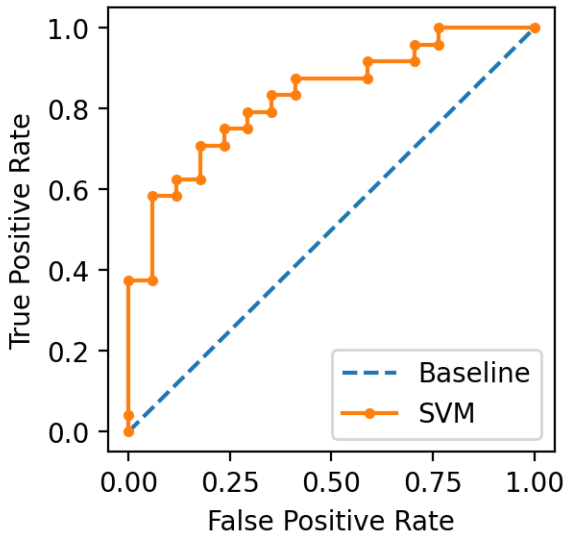


Figure 3-9: ROC AUC plot for SVM

4.4. K Nearest Neighbors

By now, it is proved that Machine Learning models make predictions by learning from the past data available. But there is a Machine Learning model built on known inputs and they are used to create a predicted output. For this model, all the attributes used will differ one to the other based on the characteristics.

K Nearest Neighbors (KNN) is one of the simplest Supervised Machine Learning algorithm mostly used for classification. It classifies a data point on how its neighbors are classified. KNN stores all available cases and classifies new based on a similarity measure. *k* in KNN is a parameter that refers to the number of nearest neighbors to include the majority voting process. A data point was classified by majority votes from its *k* nearest neighbors. Because KNN is based on feature similarity, it is possible to do classification using KNN classifier.

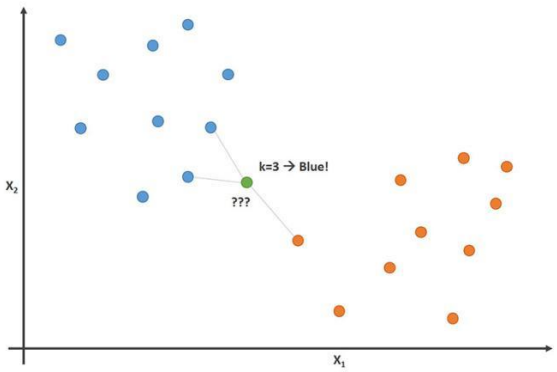


Figure 4-10: Basic visualization of a K-NN model

KNN Algorithm is based on feature similarity: Choosing the right value of *k* is a process called parameter tuning and is important for better accuracy. It is important to recall that *k* changes depending on where that point is that drastically changes your answer. The *k* number where choose by taking the square root of the total number of data points, and an odd value for *k* was selected to avoid confusion between two classes of data. For the dataset given the *k* parameter obtained is 13.

On the bases of the given dataset, the objective was to classify the obtained set as LOW or HIGH using KNN. To find the nearest neighbor the Euclidian distance was calculated, this is the distance between two points in the plane with coordinates (x, y) and (a, b) is given by:

$$Euclidian = \sqrt{(x-a)^2 - (y-b)^2}$$

The Euclidian distance of unknow data point from all points in the dataset. Since, the data set is small, the program did not take long to calculate each distance and accuracy.

	Precision	Recall	F1-Score
High	0.86	0.89	0.87
Low	0.77	0.71	0.74
Accuracy		0.83	
Macro average	0.81	0.80	0.81
Weighted average	0.83	0.83	0.83

Table 3-4: Metrics classification scores for KNN

Table 4-4 and Figure 4-11 summarize the recall obtained from the K-NN model. It achieved 89% for High classes and 71% for Low classes. Those results can tell the buyers how "safe" the location is where the house is located. KNN is a fundamental place to start in machine learning because it is easy to understand and incorporated in other forms of machine learning. It works great with small datasets and one example of that is represented on the ROC curve computed with an AUC of 0.796

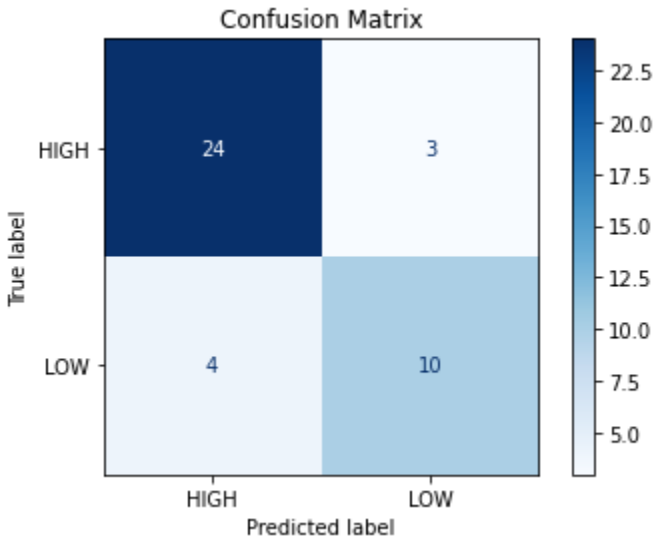


Figure 4-11: Confusion matrix for KNN

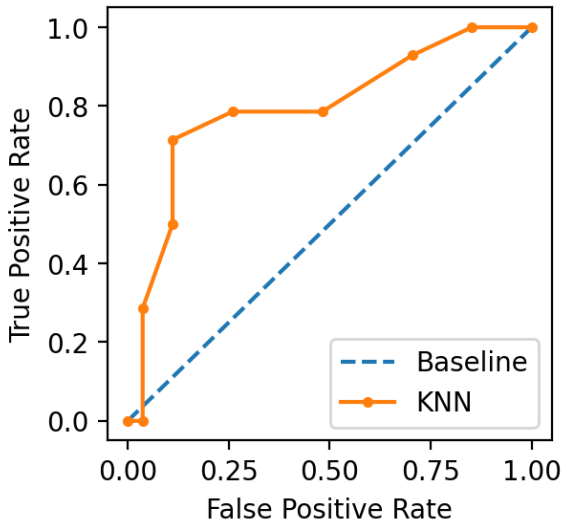


Figure 4-12: ROC Curve for K-NN

5. Results & Conclusion

Overall, the project object of collecting real-world crime data to predict and classify whether a parcel is situated in a “High” or “Low” crime rate area was achieved. It was observed that the aggregated best performing model was the KNN model. If the number of attributes in the data set was too large, then KNN would not have been as effective because of the curse of dimensionality.

This is considering that the AUC and recall scores are the most relevant measures for the crime data set. Recall was valued as one of the most important model measurements because, in the interest of public safety, there is a greater cost of mislabeling a “High” crime rate area as a “Low” crime rate area than vice versa. The AUC score was selected as a model performance parameter because it provides an aggregated comparison between all of the models and is irrespective of varying threshold values.

MODEL	ACCURACY	RECALL (HIGH)	AUC
Naïve Bayes	0.68	0.84	0.721
Random Forest	0.71	0.75	0.861
Support Vector Machine	0.73	0.86	0.828
K-Nearest Neighbors	0.83	0.89	0.796

Table 5-1: Classification Models Comparison

As shown by the recall and AUC scores in Table 5-1, the models can be ranked in order of model performance as: KNN, SVM, Random Forest, and Naïve Bayes. KNN performed well on this data set because it handles well data with few instances when compared to the number of attributes. In addition, the limited number of attributes in the data set benefited the KNN algorithm because it was not negatively affected by the curse of dimensionality.

Although some models were able to better classify new instances, the overall expectation of the project was met. One of the hypotheses proposed by the group was that an increase in population density result in a higher crime rate. Alternatively, based on the data and personal experiences, the group also expected a decrease of crime rate as the average household income went up. Both of this expectations were clearly seen in the classified regions where those factor play an important role in classification. For future research, the best way to improve on the accuracy of the model is without a doubt to collect more and better data and possibly assigning weights to each crime based on how serious they are, as in this project, a simple robbery had the same importance as occurrences such as homicide.

References

[1] - *Classification: ROC Curve and AUC | Machine Learning Crash Course*. Developers Google https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc. Date retrieved 04/30/2021.

[2] - U.S. Census Bureau. (2020). *TIGER/Line Shapefiles (machinereadable data files)*. https://www2.census.gov/geo/pdfs/maps-data/data/tiger/tgrshp2020/TGRSHP2020_TechDoc.pdf

[3] – DeepAI. (2020, September). *Random Forest*. https://deepai.org/machine-learning-glossary-and-terms/random-forest. Date retrieved 05/03/2021.

[4] – Garret, J & Viera, A. (2005, May). *Understanding Interobserver Agreement: The Kappa Statistic.* http://www1.cs.columbia.edu/~julia/courses/CS6998/Interrater_agreement.Kappa_statistic.pdf. Date retrieved 05/03/2021.

[5] – Wei-Meng, L. (2019). *Python Machine Learning*. Wiley. http://103.159.250.162:81/fdScript/RootOfEBooks/E%20Book%20Collection%202021/CSE/[Wei-Meng_Lee]_Python_Machine_Learning.pdf Date retrieved 05/03/2021.