United States Crime Predictions Using Machine Learning

MSBA 273
Fall 2020

Group #9
Chengwu Weng
Mira Daya
Misha Khan
Yue Fang

TABLE OF CONTENTS

I. Overview

Crime is widespread throughout the entire United States. Such occurrences can happen daily and due to lack of resources, police officers' efforts are hindered. With our project, we examined the data collected throughout the nation and evaluated certain attributes to determine violent crime rates in different communities across the United States.

A. Business Implications

Our objective for this project is to predict whether a city is considered to have higher rates of violent crimes by considering attributes like neighborhood, income level, education level, ethnicity, and more. We hope that by developing a model that classifies whether a city is high in violent crimes or not, it can help with properly allocating police resources and better police enforcement. Additionally, citizens can also review our findings to help make decisions on whether or not to move to a certain city.

II. Data

A. Data Acquisition

We gathered our crime and communities data set from UCI Machine Learning Repository. The dataset combines socio-economic data from 1990 US Census, 1990 US Law Enforcement, and 1995 FBI crime data. It consists of 128 attributes, including 122 predictive attributes and five non-predictive as well as one goal attribute. With our project, we predict if a city is considered a high violent crime area by using the "ViolentCrimePerPop" variable and discretizing it into a binary variable to perform classification modeling.

Table 1: Variables Overview

Data	**Variable**
non-predictive attributes	state
	county
	community
	communityname
	fold
predictive attributes	demographic(70)
	income(20)
	education(3)
	employment(6)
	police-related(21)
	crime-related(2)
goal attributes	ViolentCrimesPerPop

B. Data Cleaning

First, we noticed our dataset had several “?” so we converted it into NaN to make our process easier for analysis. For each variable that had NaN observations, we found the percent of how many observations in that column were NaN and used a threshold of 50% to determine which variables to eliminate. Our dataset is fairly large so we also decided to eliminate 22 columns that had NaN values greater than 50%.

Table 2: Columns name with the percentage of NA greater than 50%

LemasSwFTPerPop 0.84	PctPolicHisp 0.84
LemasSwFTFieldOps 0.84	PctPolicAsian 0.84
LemasSwFTFieldPerPop 0.84	PctPolicMinor 0.84

LemasTotalReq 0.84	OfficAssgnDrugUnits 0.84
LemasTotReqPerPop 0.84	NumKindsDrugsSeiz 0.84
PolicReqPerOffic 0.84	PolicAveOTWorked 0.84
PolicPerPop 0.84	PolicCars 0.84
RacialMatchCommPol 0.84	PolicOperBudg 0.84
PctPolicWhite 0.84	LemasPctPolicOnPatr 0.84
PctPolicBlack 0.84	LemasGangUnitDeploy 0.84
LemasSwFTPerPop 0.84	PolicBudgPerPop 0.84

C. Data Preprocessing

1. *Creating a Binary Response Variable*

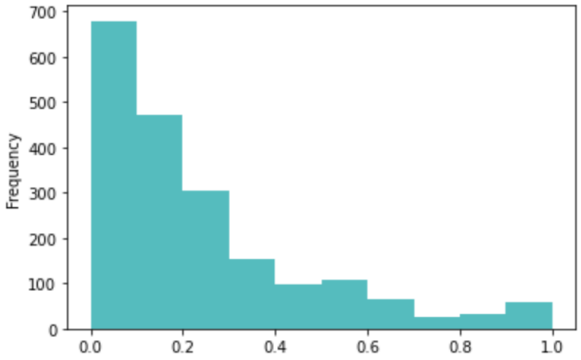
The response variable we are interested in predicting is whether or not a location is high in violent crimes. Violent crimes are classified as murder, rape, robbery, and assault. We decided to create a binary variable called HighCrime by discretizing "ViolentCrimesPerPop".

For "ViolentCrimesPerPop", we calculated the summary statistics in order to determine the best cutoff value to create binary classification.

Table 3: "ViolentCrimesPerPop" Summary Statistics

Statistics	**Values**
Average	0.2380
Max	1.0000
Min	0.0000
Median	**0.1500**
Q1 quantile	0.07
Q2 quantile	0.15
Q3 quantile	0.33
100th quantile	0.03

Initially, we considered setting the mean, 0.238, as our cutoff value to determine binary classification for "ViolentCrimesPerPop" but saw that the mean is heavily influenced by the outliers shown in Figure 1 and Figure 2.



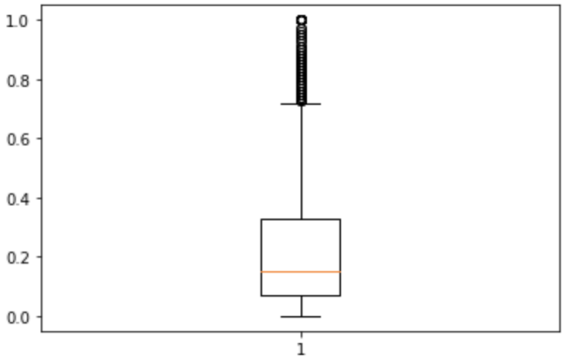


Figure 1: Distribution of "ViolentCrimesPerPop" Figure 2: Boxplot of "ViolentCrimesPerPop"

Since the median is less distorted by outliers, we decided to use the median, as our cutoff value. We selected median = 0.15 as our best cutoff value. Therefore, we can get the following equation:

$$HighCrime = \begin{cases} 1, & if\ ViolentCrimesPrerPop \geq 0.15 \\ 0, & Otherwise \end{cases}$$

After setting the threshold to be 0.15, the distribution of the new transformed variable is approximately 50/50. The percentage of high crime changes depending on our threshold value for "ViolentCrimesPerPop". There are 49.79% of regions in the United States that are considered a high crime area and 50.20% are classified as a low crime area.

```
pos = crime[(crime['highCrime'] == 1)]
pos_perc = len(pos) / len(crime)
neg_perc = 1 - pos_perc
print('Positive instance percentage is ',pos_perc)
print('Negative instance percentage is ',neg_perc)
```

```
Positive instance percentage is  0.4979939819458375
Negative instance percentage is  0.5020060180541626
```

Figure 3: Positive/Negative Instance Percentage

2. *Principal Component Analysis*

We utilized the Principal Component Analysis (PCA) package in Python. PCA can be considered as a projection method which projects observations from a p-dimensional space with p-variable to a k dimensional space (where K< P) so as to conserve the maximum amount of information (information is measured here through the total variance) from the initial dimensions.

PCA is useful to reduce the number of variables from a large dataset and transforms a set of variables into a new set of uncorrelated variables. Since there are many variables in our datasets, PCA is an optimal way for us to examine fewer relationships between specific variables and possibly avoid overfitting the model. Multicollinearity causes small changes in input data to make large changes in a model and it's predicted parameters. The principal components only take effect on the "HighCrime" variables. As mentioned, the problem with data redundancy is overfitting models so via PCA, we can alleviate the overfitting problem to some extent. The principal components will be classified as "new" variables that are independent of one another.

We conducted PCA on the dataset with 10-fold cross validation and training/ testing subsets to get both cumulative variance ratio and explained variance ratio. The results are consistent even though we use different training, testing and validation dataset.

	Cumulative Variance Ratio	Explained Variance Ratio
15	0.857003	0.008921
16	0.865629	0.008626
17	0.873105	0.007477
18	0.880060	0.006955
19	0.886483	0.006423
20	0.892731	0.006248
21	0.898692	0.005961
22	0.904052	0.005360
23	0.909252	0.005200
24	0.914118	0.004866
25	0.918932	0.004814

	Cumulative Variance Ratio	Explained Variance Ratio
15	0.856756	0.008927
16	0.865351	0.008595
17	0.872908	0.007557
18	0.879904	0.006996
19	0.886329	0.006425
20	0.892726	0.006397
21	0.898621	0.005895
22	0.904044	0.005423
23	0.909303	0.005259
24	0.914201	0.004898
25	0.918897	0.004695

	Cumulative Variance Ratio	Explained Variance Ratio
15	0.856756	0.010006
16	0.865351	0.008391
17	0.872908	0.007570
18	0.879904	0.007090
19	0.886329	0.006597
20	0.892726	0.006130
21	0.898621	0.005721
22	0.904044	0.005465
23	0.909303	0.005321
24	0.914201	0.005043
25	0.918897	0.004595

Figure 4: Cumulative and Explained Variance Ratio of
CV (left), Training (middle) and Testing data (right)

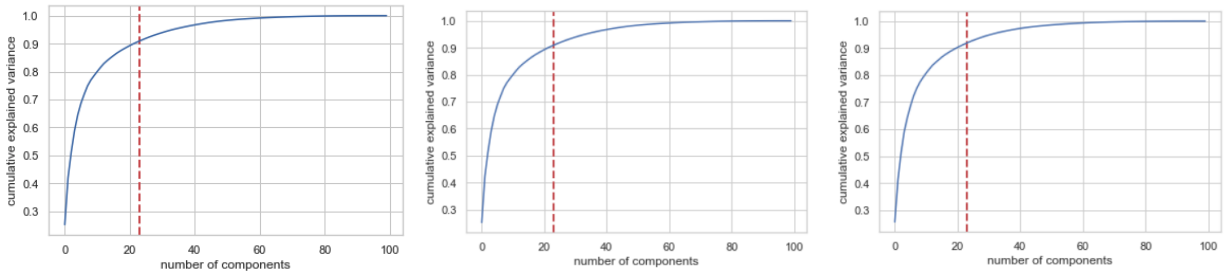


Figure 5: Cumulative Explained Variance Curve of
CV (left), Training (middle) and Testing data (right)

The curve quantifies how much of the total n-dimensional variance is contained within the first N components. The first 23 components contain 90% of the variances. We then take the 23 variables that explain the 90% of the variance out of the original 103 variables and then transform the dataset. In Section IV, parts B and E, we will use these 23 principal components to create Naive Bayes and Random Forest Models.

III. Exploratory Data Analysis

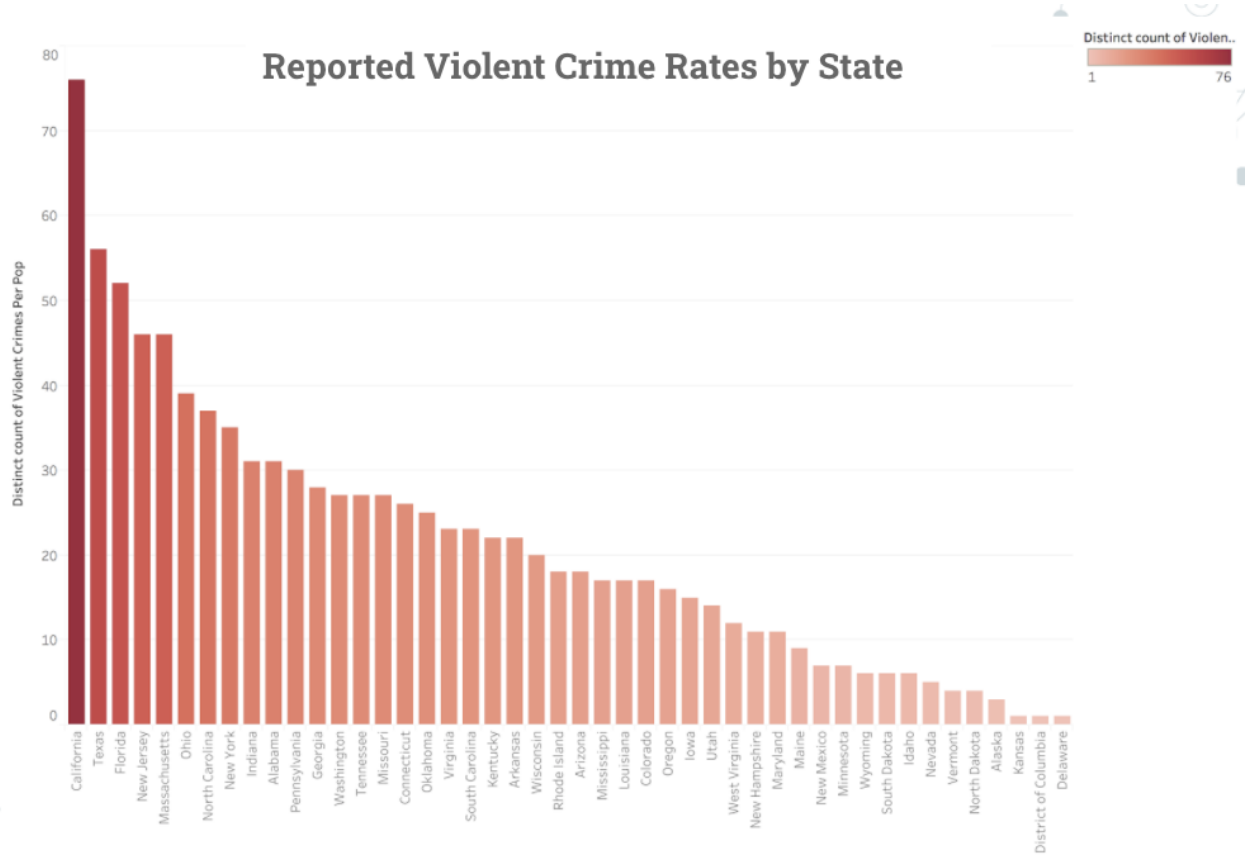


Figure 6: Number of Crime per Population

This graph shows us the number of cities that are reporting violent crime rates per state. California shows the highest value count and has over 70 instances of communities who reported violent crime rates meaning our dataset includes more cities in California reporting their rates followed by Texas, Florida, New Jersey, and so on.

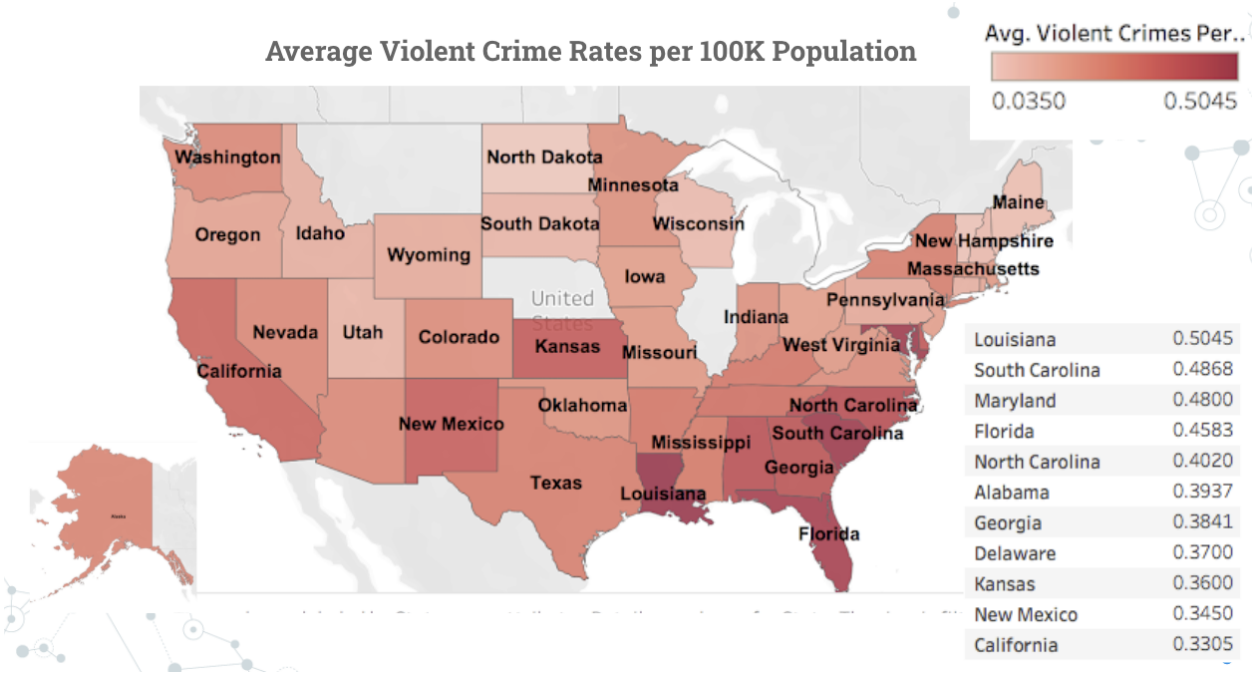


Figure 7: Average Violent Crime Rate per 100k Population

This map shows the average crime rate by state considering all the cities present. Notice that the South-Eastern states are much darker shades than all others and then states like California, Kansas, New Mexico also show higher rates of violent crime. This means that the average violent crime rates are high in these states.

The box to below shows a glimpse at the numbers for the high crime states represented by darker colored shading in the map.

State	City	Violent Crimes Per Pop
Alabama	Gadsdencity	0.9
Arkansas	Blythevillecity	0.9
California	EastPaloAltocity	0.97
	Comptoncity	0.9
	SantaFeSpringscity	0.86
	Inglewoodcity	0.86
Connecticut	Hartfordtown	0.94
	NewHaventown	0.88
Florida	Orlandocity	0.95
	LakeCitycity	0.87
	DaytonaBeachcity	0.86
Georgia	Brunswickcity	0.86
Indiana	Garycity	0.89
Maryland	Salisburycity	0.91
Massachusetts	Lawrencecity	0.88
Mississippi	Grenadacity	0.96
New Jersey	Bridgetoncity	0.93
	Trentoncity	0.85
	AsburyParkcity	0.85
New York	NewYorkcity	0.87
North Carolina	NewBerncity	0.91
	Fayettevillecity	0.86
Ohio	Limacity	0.97
	Youngstowncity	0.95

Figure 8: Violent Crimes per Population of Cities

These are some of the top violent crimes per pop cities by state. California, Mississippi, Florida, and Ohio all include cities with violent crime per population greater than 0.95. This relates back to the map, California has many cities with high rates which drags the average crime rate higher and therefore shows a higher density on the map.

IV. Modeling

We are interested in predicting the binary variable ‘highCrime’. We used the follow models which all allow for binary classification:

1. Naive Bayes
2. Decision Tree
3. Random Forest

The dataset was split into training (70%) and test (30%). Additionally, we used 10-fold cross-validation to build ten different models with our data set. Using cross-validation allows us to make predictions on all of our data rather than just one training and testing split. Cross-validation allows us to be more confident in our algorithm performance because we have ten sets of metrics. If all ten sets of metrics are similar, that means the algorithm is consistent and performing accurately. In this way, compared with training & testing splitting, we will get consistent and robust results since we train and test our model ten times and get average evaluation metrics (like accuracy, precision, recall).

A. Naive Bayes - Benchmark

We are using the Guassian Naive Bayes method as a benchmark first. In Gaussian Naive Bayes, features that are continuous are assumed to be distributed according to Gaussian Normal Distribution where the mean is 0 and standard deviation is 1. We want to compare various ways of running Naive Bayes Classification to determine which methods (training/testing or cross validation) give the best accuracy scores. We began by comparing Naive Bayes classification with training and testing data versus 10-fold cross validation.

Table 4: Accuracy/ Precision/ Recall Scores

	Naive Bayes	**Cross Validation**	**Training/Testing**
Average accuracy	0.79990	0.80004	0.79449
Average precision	0.87032	0.87056	0.83750
Average recall	0.70292	0.70597	0.70526

The accuracy of the Naive Bayes simple model is 79.99% for simple models, 80.04% for cross-validation models, and 79.44% for models using training and testing sets. For the simple model, the confusion matrix shows 99% of communities that are low in violent crime were predicted correctly, and 40% of high violent crime communities were predicted correctly. Alternatively, 1.2% of actual low violent crime areas were predicted as high violent crime areas and 60% of actual high violent crime areas were predicted as low.

For the training and testing set model, the confusion matrix shows 98% of communities that are low in violent crime were predicted correctly, and 40% of high violent crime communities were predicted correctly. Alternatively, 1.9% of actual low violent crime areas were predicted as high violent crime areas and 60% of actual high violent crime areas were predicted as low.

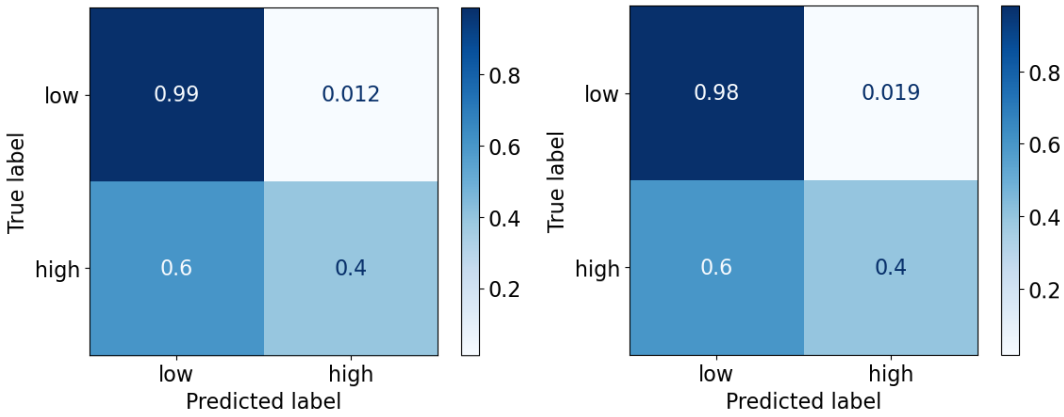


Figure 9: Confusion Matrix of Simple NB(left), NB with Train/Test(right)

B. PCA & Naive Bayes

We applied the 23 principal components found through PCA to Naive Bayes using two methods: split data into training and testing and split data based on 10-fold cross validation. Since we were experiencing issues with collinear independent variables, the benefit of PCA is that all variables are now independent. Additionally, multicollinearity causes redundancy that leads to overfitting.

Table 5: Accuracy/ Precision/ Recall Scores of Simple NB, CV, Training/Testing

	Simple NB	**Cross Validation**	**Training/Testing**
Average accuracy	0.79990	0.7427488	0.716792
Average precision	0.87032	0.7347291	0.720000
Average recall	0.70292	0.7321659	0.663158

The accuracy of the Naive Bayes simple model is 79.99% for simple models, 74.27% for cross-validation models, and 71.7% for models using training and testing sets. For the simple model, the confusion matrix shows 67% of communities that are low in violent crime were predicted correctly, and 76% of high violent crime communities were predicted correctly. Alternatively, 33% of actual low violent crime areas were predicted as high violent crime areas and 24% of actual high violent crime areas were predicted as low.

For the training and testing set model, the confusion matrix shows 76% of communities that are low in violent crime were predicted correctly, and 65% of high violent crime communities were predicted correctly. Alternatively, 24% of actual low violent crime areas were predicted as high violent crime areas and 35% of actual high violent crime areas were predicted as low.

Compared to the benchmark value, the accuracy rate dropped from 80% to 74%. However, based on the confusion matrix, Naive Bayes model with PCA performs better on predicting a higher crime rate correctly. The preprocessing method PCA did help us to improve our model.

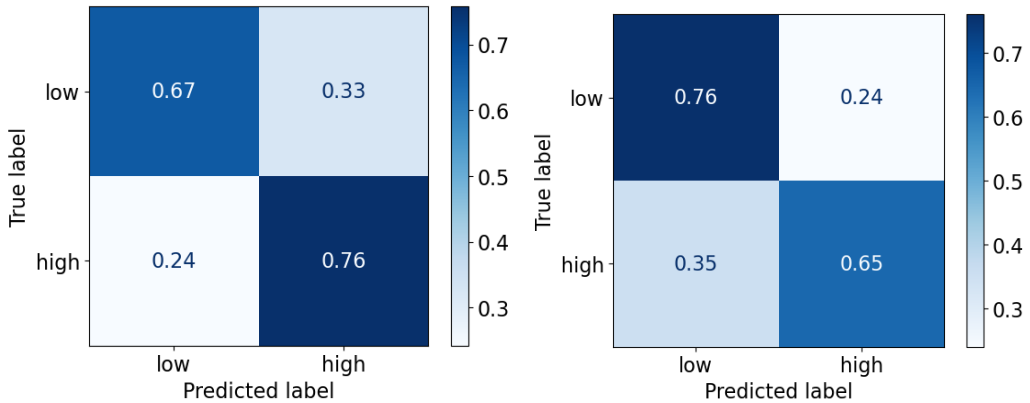


Figure 10: Confusion Matrix of Simple Model(left) and Training/Testing(right)

C. Decision Tree - Benchmark

Before we can model a benchmark model: decision tree, we use cross validation and "for" loops to determine the depth of the tree. Looking at Figure 11, the x-axis represents the number of tree depths, and y-axis is the average accuracy score of 10-fold cross validation. We can find that the max accuracy score appeared at the position of depth = 4 and therefore we can conclude that our tree will have a length of four nodes.

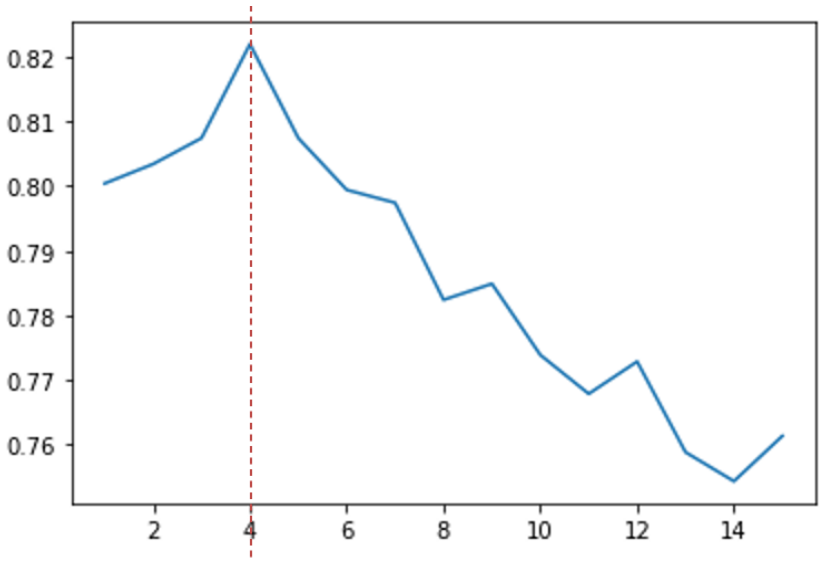


Figure 11: Accuracy Score with Different Tree Depths

Similar to Naive Bayes, we adopt two methods of splitting data: split data into training and testing and split data based on 10-fold cross validation. For Method 1, we first modeled a decision tree on the dataset using 10-fold cross validation.

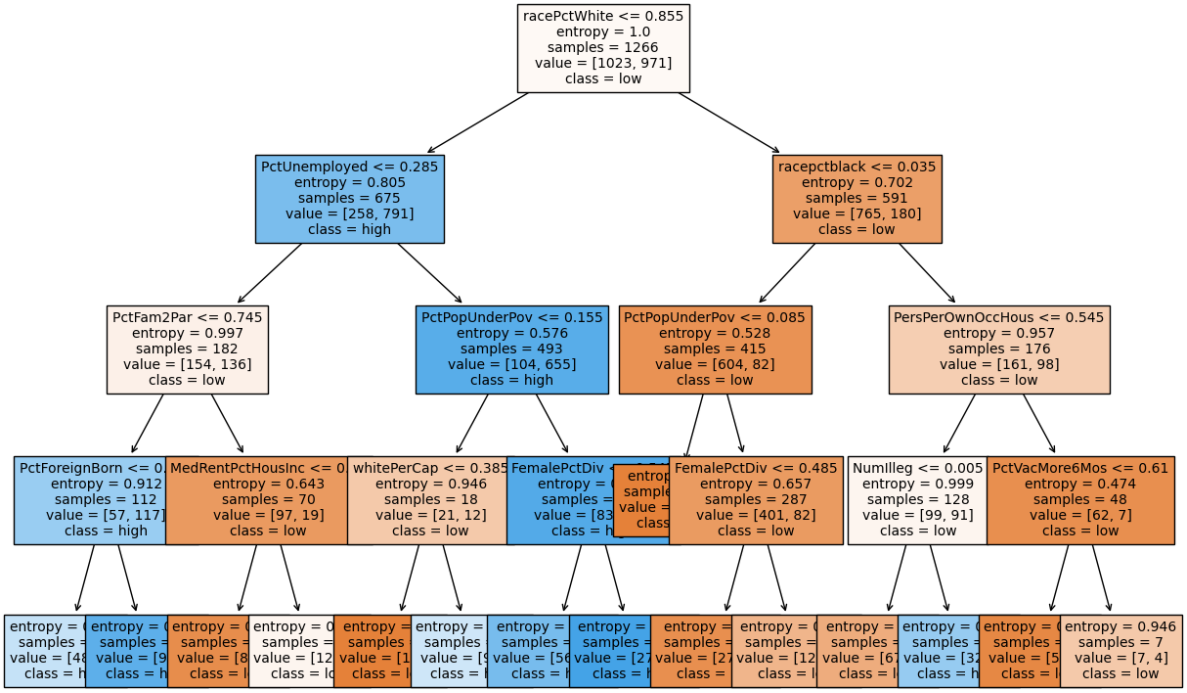


Figure 12: Decision tree of Method 1

Table 6: Accuracy/ Precision/ Recall Scores of Method 1

	Value
Average accuracy	0.7958955
Average precision	0.8344216
Average recall	0.7463581

The accuracy of the Random Forest is 83.04% for simple models, 79.84% for cross-validation models, and 76.7% for models using training and testing sets. For the simple model, the confusion matrix shows 93% of communities that are low in violent crime were predicted correctly, and 74% of high violent crime communities were predicted correctly. Alternatively, 6.7% of actual low violent crime areas were predicted as high violent crime areas and 26% of actual high violent crime areas were predicted as low.

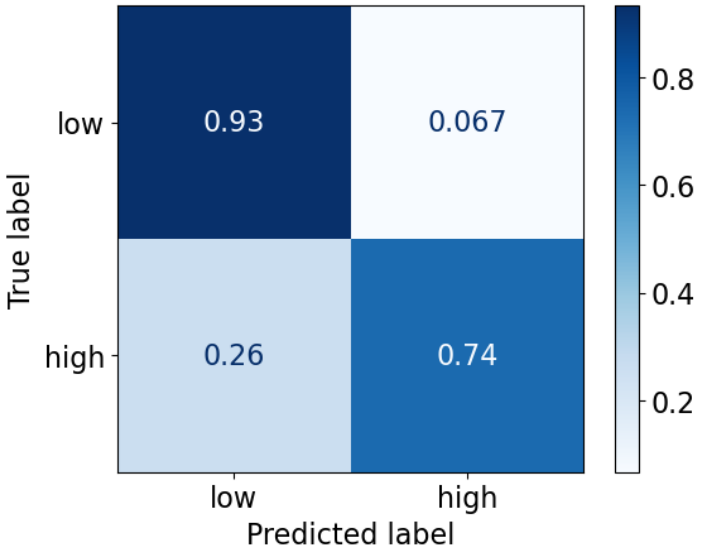


Figure 13: Confusion Matrix of Method 1

For Method 2, we modeled a decision tree with a length of four nodes on splitting data into training and testing subsets.

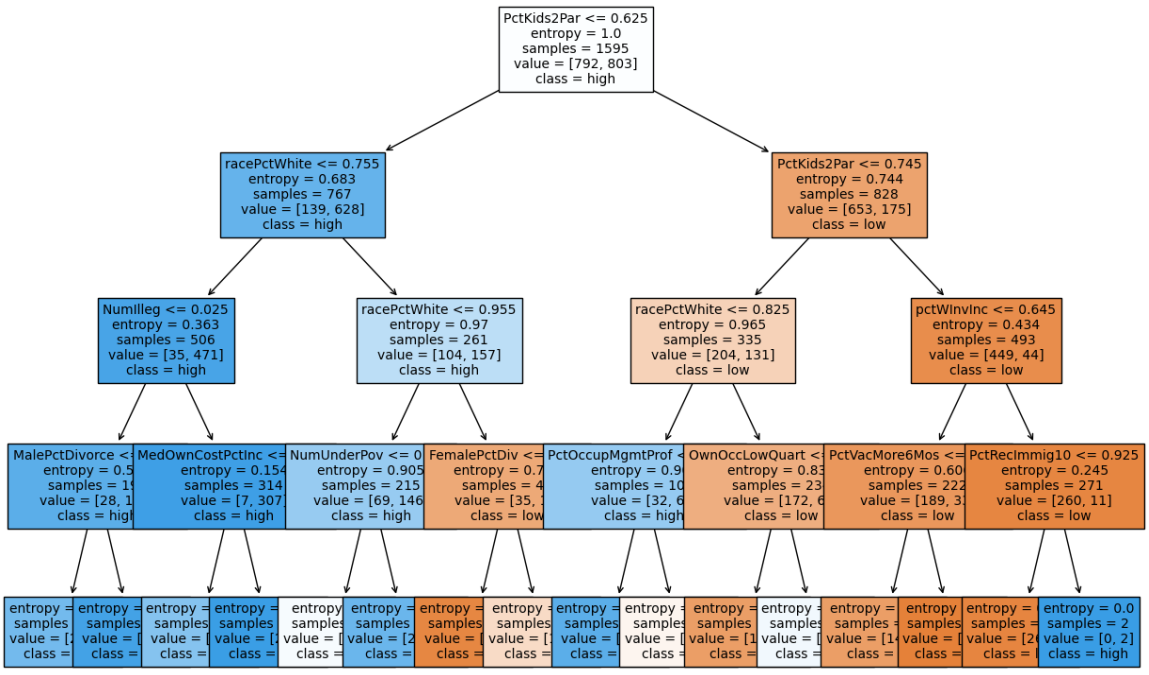


Figure 14: Decision Tree of Method 2

Table 7: Accuracy/ Precision/ Recall Scores of Method 2

	Value
Average accuracy	0.8270677
Average precision	0.8071066
Average recall	0.8368421

With Method 2, the accuracy of the decision tree is 82.71%. The confusion matrix shows 82% of communities that are low in violent crime were predicted correctly, and 7984 of high violent crime communities were predicted correctly. Alternatively, 18% of actual low violent crime areas were predicted as high violent crime areas and 16% of actual high violent crime areas were predicted as low.

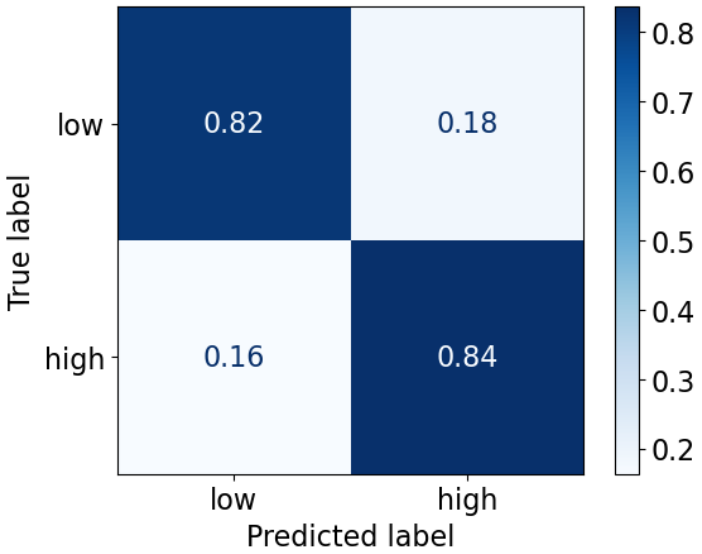


Figure 15: Decision Tree of Method 2

D. Random Forest

We conducted a Random Forest model since it is an improved model of decision tree. As we have 103 attributes in our dataset, the estimator is set to 10 by calculating the square root of total attributes. There are 10 Decision Trees models and the result of prediction will be determined by majority rules.

For Method 1, we created a Random Forest on the dataset using 10-fold cross validation.

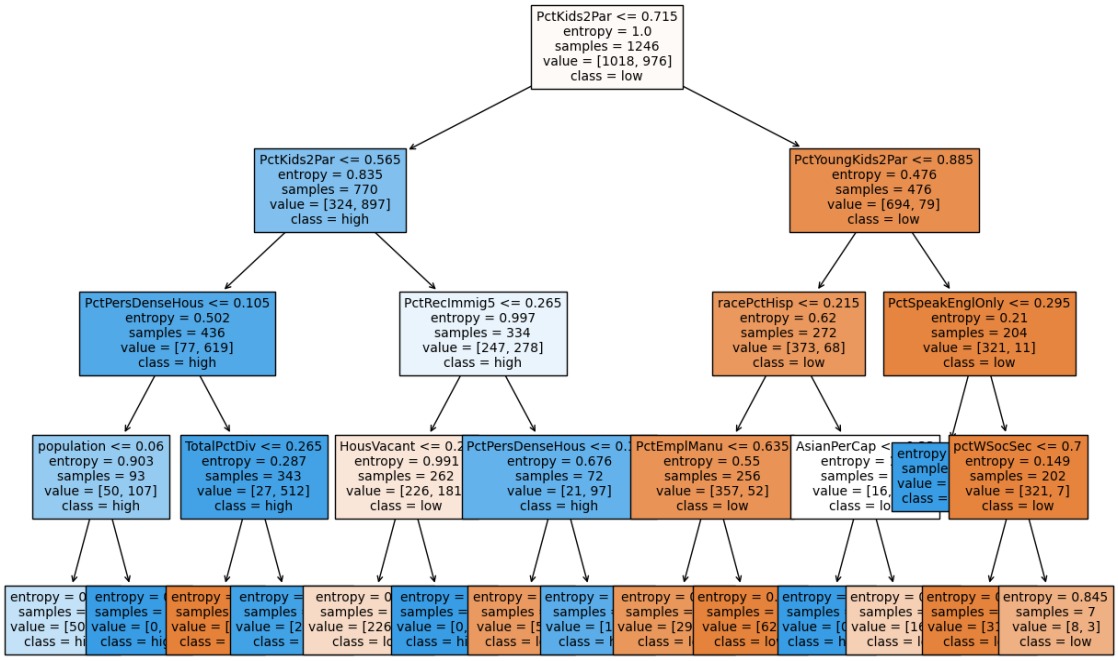


Figure 16: One of Random Forest Model of Method 1

Table 8: Accuracy/ Precision/ Recall Scores of Method 1

	Value
Average accuracy	0.8304933
Average precision	0.8363264
Average recall	0.8217552

With Method 1, the accuracy of the decision tree is 83.25%. The confusion matrix shows 88% of communities that are low in violent crime were predicted correctly, and 85% of high violent crime communities were predicted correctly. Alternatively, 12% of actual low violent crime areas were predicted as high violent crime areas and 15% of actual high violent crime areas were predicted as low.

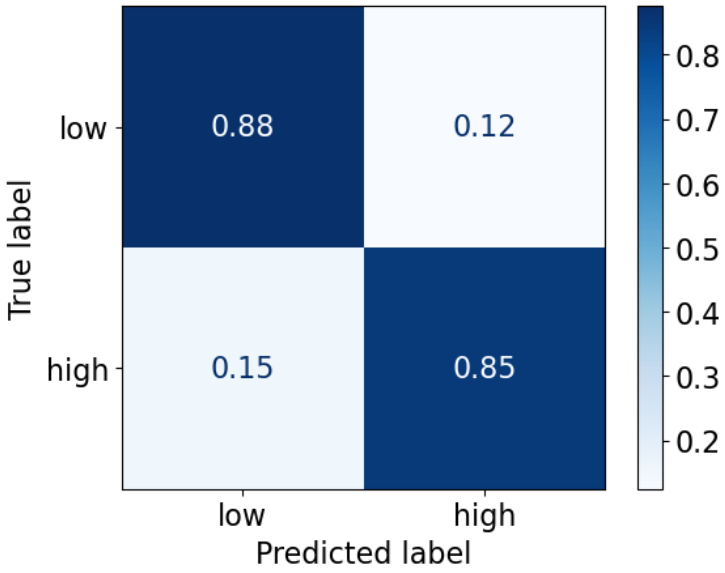


Figure 17: Confusion Matrix of Method 1

For Method 2, we then conducted a Random Forest model on splitting data into training and testing subsets.

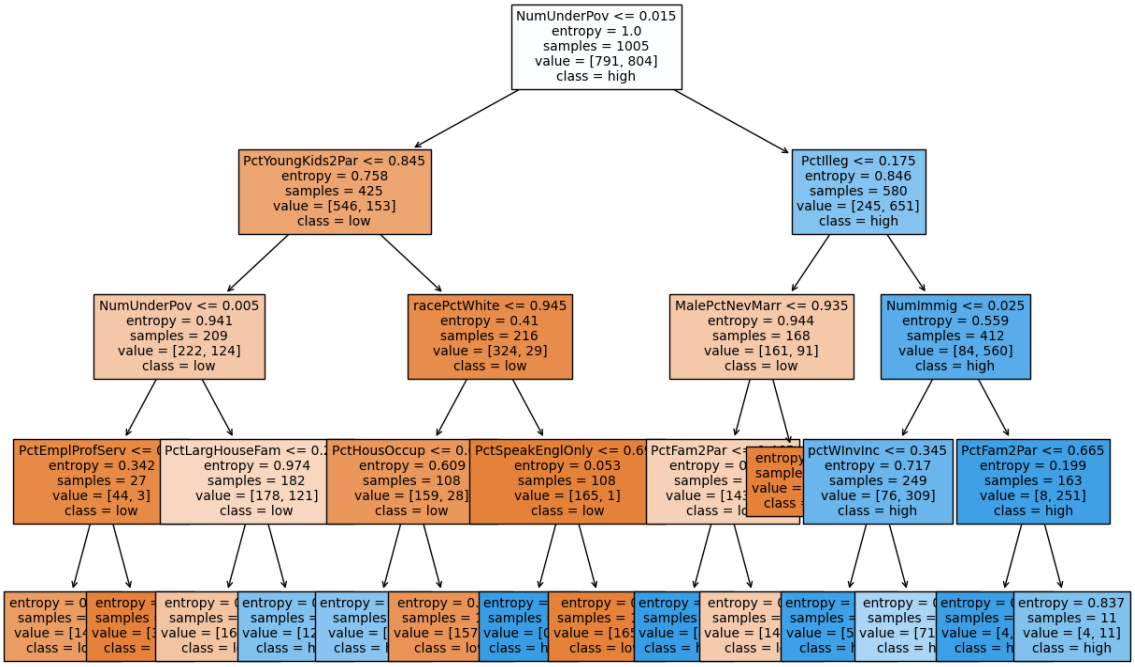


Figure 18: One of Random Forest Model of Method 2

Table 9: Accuracy/ Precision/ Recall Scores of Method 2

	Value
Average accuracy	0.8471178
Average precision	0.8307692
Average recall	0.8526316

With Method 2, the accuracy of the decision tree is 82.70%. The confusion matrix correctly predicted 85% of communities low violent crime communities and 85% of communities to be high violent crime communities. It also predicted that 15% communities are predicted to be low but were actually high violent crime areas and 15% of communities to be high but were actually low violent crime areas.

The confusion matrix shows 85% of communities that are low in violent crime were predicted correctly, and 85% of high violent crime communities were predicted correctly. Alternatively, 15% of actual low violent crime areas were predicted as high violent crime areas and 15% of actual high violent crime areas were predicted as low.

Compared to the benchmark, the Random Forest model has a higher accuracy score. Based on the confusion matrix, the percentages of both predicting a low crime and a high crime are slightly improved as well. Increasing a diversity in random forest improves a decision tree model performance.

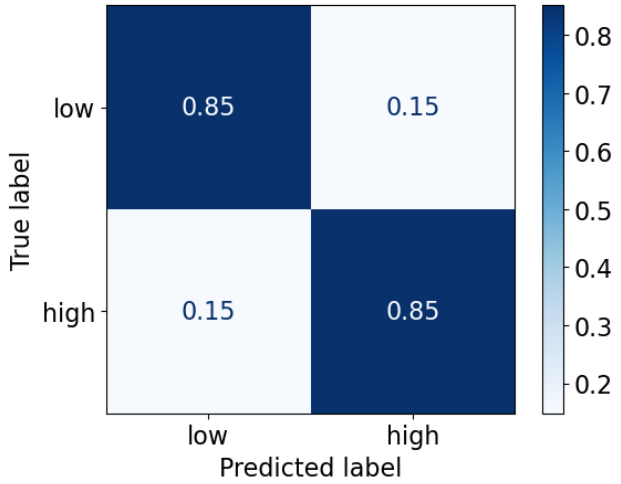


Figure 19: Confusion Matrix of Method 2

E. PCA & Random Forest

We train the Random Forest with the dataset after PCA and select the top 23 informative components. Additionally, we split the new data into training and testing and split data based on 10-fold cross validation. Similar to Section B, PCA & Naive Bayes, the evaluation metrics (accuracy, precision and recall) decreases compared to the result of Random Forest, that is, the problem of overfitting can be to some extent alleviated.

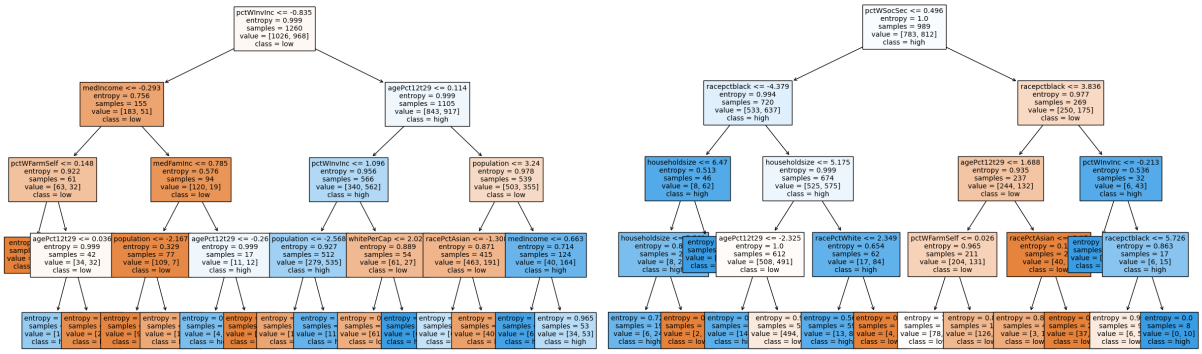


Figure 20: One of Random Forest Model of CV (left) and Training/Testing (right)

Table 10: Accuracy/ Precision/ Recall Scores of Random Forest, CV, Training/Testing

	Random Forest	**Cross Validation**	**Training/Testing**
Average accuracy	0.8304933	0.7974031	0.766917
Average precision	0.8363264	0.8030297	0.753927
Average recall	0.8217552	0.7401959	0.757895

The accuracy of the Random Forest is 83.04% for simple models, 79.84% for cross-validation models, and 76.7% for models using training and testing sets. For the simple model, the confusion matrix shows 81% of communities that are low in violent crime were predicted correctly, and 58% of high violent crime communities were predicted correctly. Alternatively, 19% of actual low violent crime areas were predicted as high violent crime areas and 42% of actual high violent crime areas were predicted as low.

For the training and testing set model, the confusion matrix shows 70% of communities that are low in violent crime were predicted correctly, and 79% of high violent crime

communities were predicted correctly. Alternatively, 30% of actual low violent crime areas were predicted as high violent crime areas and 21% of actual high violent crime areas were predicted as low.

Based on the accuracy score and confusion matrix, this model does not perform as well as Decision Tree and Random Forest. PCA reduces variables to help with overfitting problems, but Random Forest also has mechanisms to control overfitting through bagging. We are not surprised to see that PCA does not improve Random Forest performance.

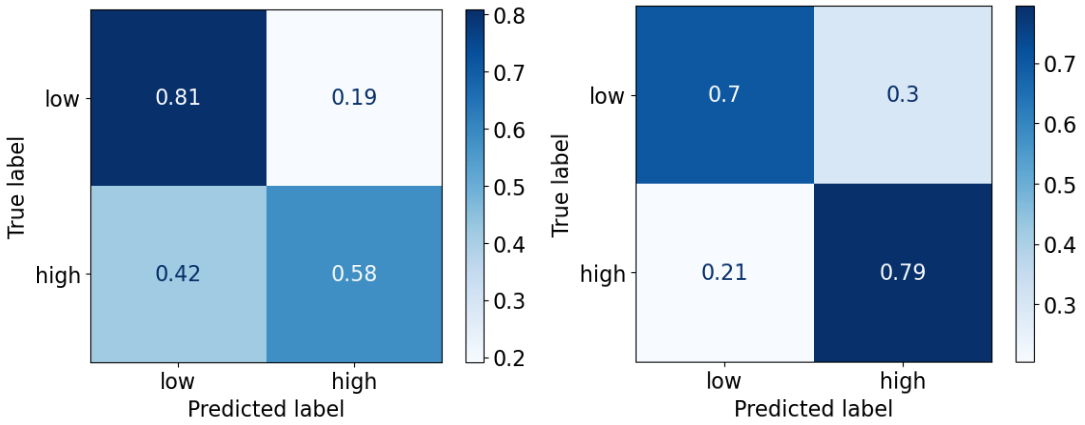


Figure 21: Confusion matrix of Simple Model (left) and Training/Testing(right)

V. Model Evaluation

These are the ROC curves we created for both Naive Bayes and Random Forest Models.

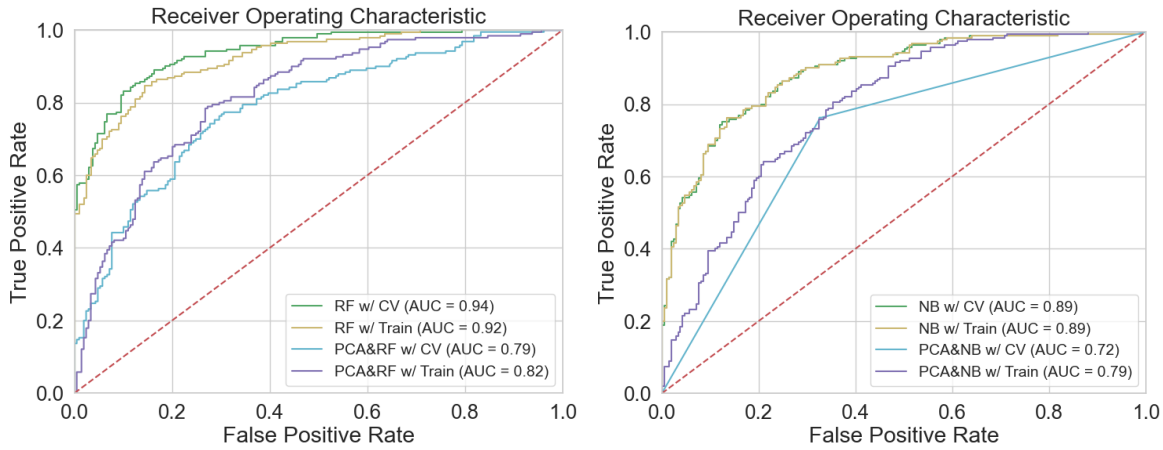


Figure 22: ROC curve of Random Forest (left) and Naive Bayes (right)

On the left, we can see that the RF model with Cross Validation gives us the most AUC value and is closest to the top left corner indicating a better performance than the other methods we used for RF. For Naive Bayes, the NB with training and testing and NV with CV has the highest ROC curve, therefore telling us that the TPR are higher for that model and the AUC is the largest as well. AUC is used as a general measure of predictive accuracy, and measured as the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance.

The true positive rates for these models are higher than the comparison models. The AUC values reinforce that the predictive accuracy is highest for Naive Bayes with Training and testing data and highest for Random Forest with cross validation.

VI. Key Takeaways

Accuracy measures across the board range from a minimum of 71.69% to a maximum of 84.71%. For Naive Bayes, our highest accuracy was seen when taking the average from conducting 10-fold accuracy. Note that the accuracy scores are only different by less than 1% from each model before using PCA. After using PCA, the accuracy decreases more dramatically which can be due to eliminating multicollinearity and overfitting issues that came from independent variable realtionships.

For Decision Trees, accuracy scores increase with use of training and testing sets versus using 10-fold cross validation. This may be due to the overfitting problem: we only use training dataset to train the model one time, and this may result in lack of information from testing sets. If we select the testing dataset fortunately fit with the model, the accuracy score will be higher. In cross validation context, we use all data observations and train and test the model ten times so we can alleviate the overfitting problem compared to training/testing one time.

Random Forest models follow similar accuracy patterns as Naive Bayes models where models with cross validation and training and testing data show less than 1% difference in accuracy. When PCA is applied to the data set and we feed components into the model, the accuracy scores decrease more dramatically. Similar to Naive Bayes, this can be due to

elimination of multicollinearity and overfitting problems that could be causing simple Random Forest models to have higher accuracy.

	Accuracy	Precision	Recall
Simple NB	0.799900	0.870324	0.702920
NB CV	0.800406	0.870564	0.705969
NB w/ Train/Test	0.794486	0.837500	0.705263
PCA NB CV	0.742749	0.734729	0.732166
PCA NB Train/Test	0.716792	0.720000	0.663158
DT CV	0.795896	0.834422	0.746358
DT w/ Train/Test	0.827068	0.807107	0.836842
RF CV	0.830493	0.836326	0.821755
RF w/ Train/Test	0.847118	0.830769	0.852632
PCA RF CV	0.797403	0.803030	0.740196
PCA RF w/ Train/Test	0.766917	0.753927	0.757895

Figure 23: Accuracy, Precision and Recall Among Different Models

Random Forest's accuracy scores are much higher no matter using Cross Validation or Training/Testing dataset. Hence, we select the most important features based on the entropy from Random Forest. From Figure 24, the features of population, household size, age, race play a significant role in predicting crime.

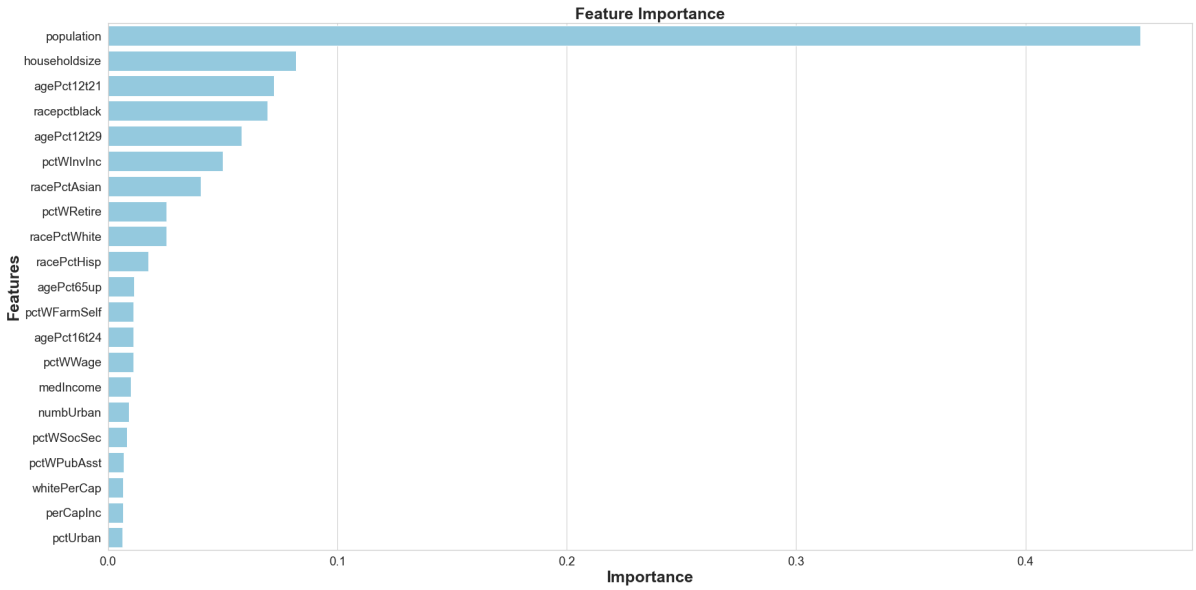


Figure 24: Features Importance of Random Forest

However, crime patterns cannot be static since patterns change over time as well as the crime factors change over time. Since we are considering only some limited factors years ago, full accuracy cannot be achieved. For getting better results in prediction we have to find more crime attributes of places instead of fixing certain attributes. Till now, we trained our models using certain attributes but we are planning to include more factors to improve accuracy.

VII. Conclusion

Our goal is to help with resource management for the allocation of police departments for better police enforcement, the percentage of accuracy rate for correctly predicting the high violent crime communities is an important feature. We want to ensure that we are accurately predicting what communities will have high violent crime rates.

Overall, the Random Forest model performed with the highest accuracy 83%. This model provides us with the most trustworthy model to make the best predictions so far. Some things to consider for the future would be to work on improving the model and examining which specific attributes contribute to a higher crime rate to help police departments with properly allocate their resources.

Appendix

Zoomed In Decision Trees

Figure 12 Decision Tree of Method 1

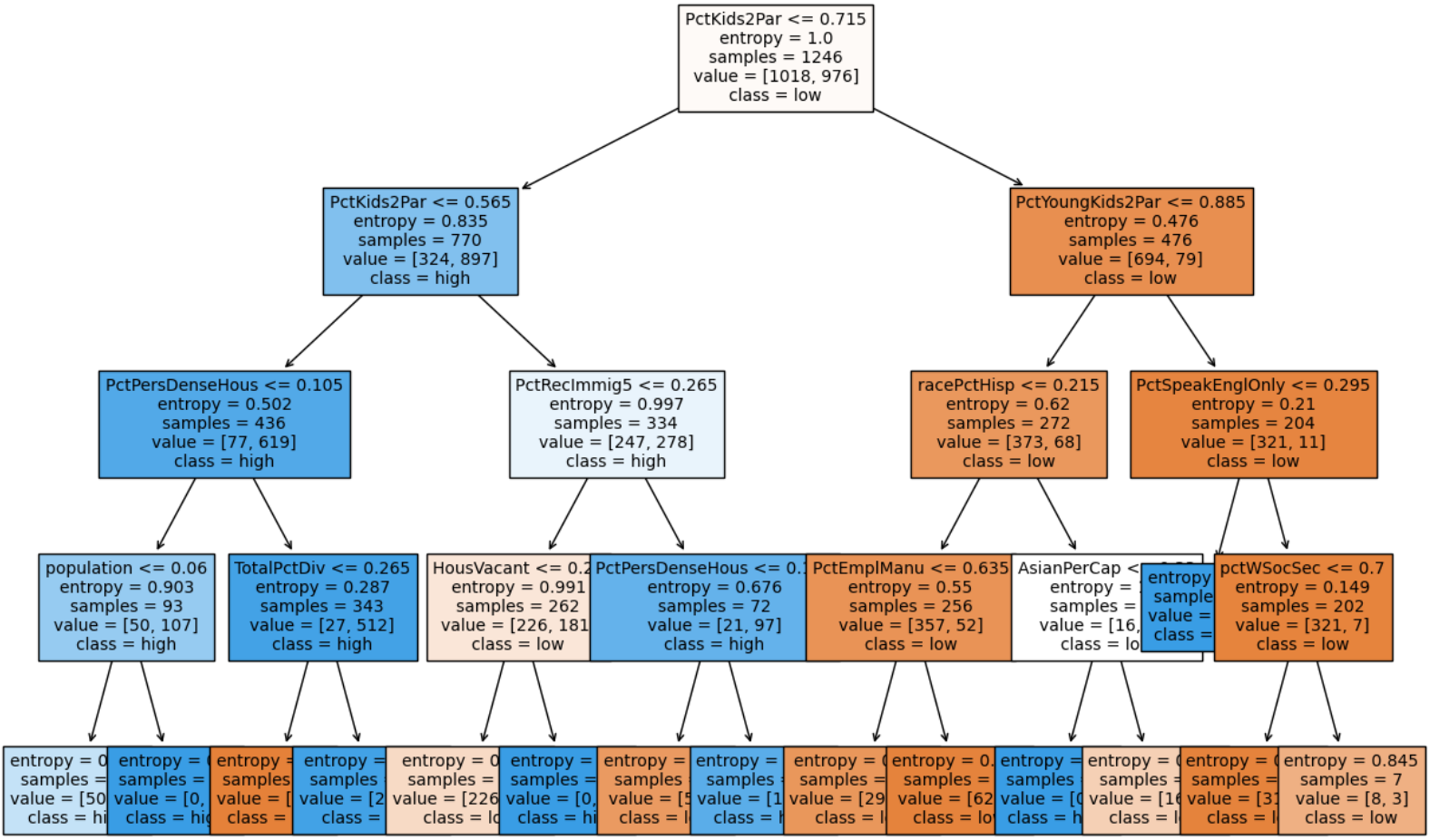


Figure 14 Decision Tree of Method 2

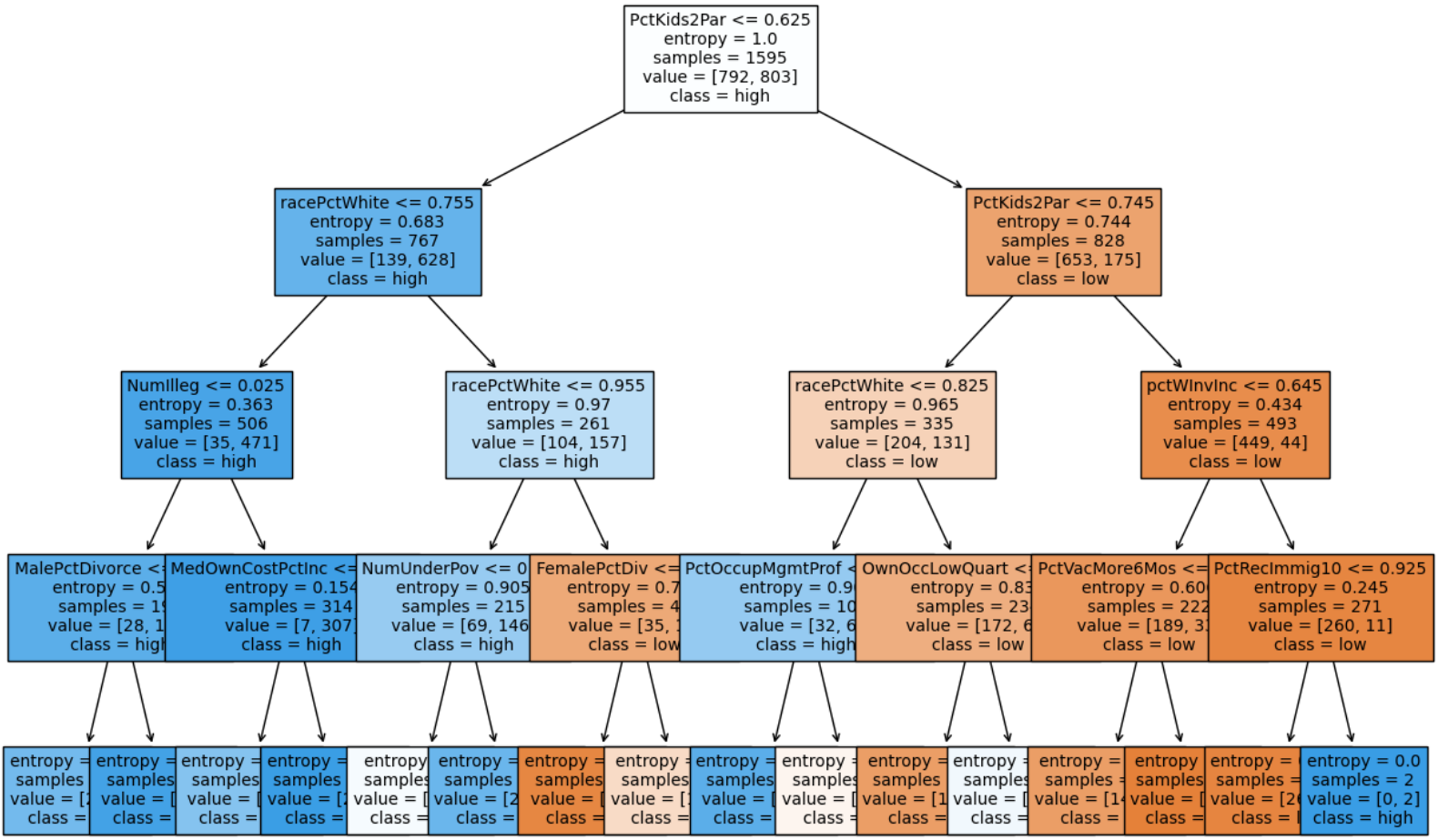


Figure 16 Random Forest of Method 1

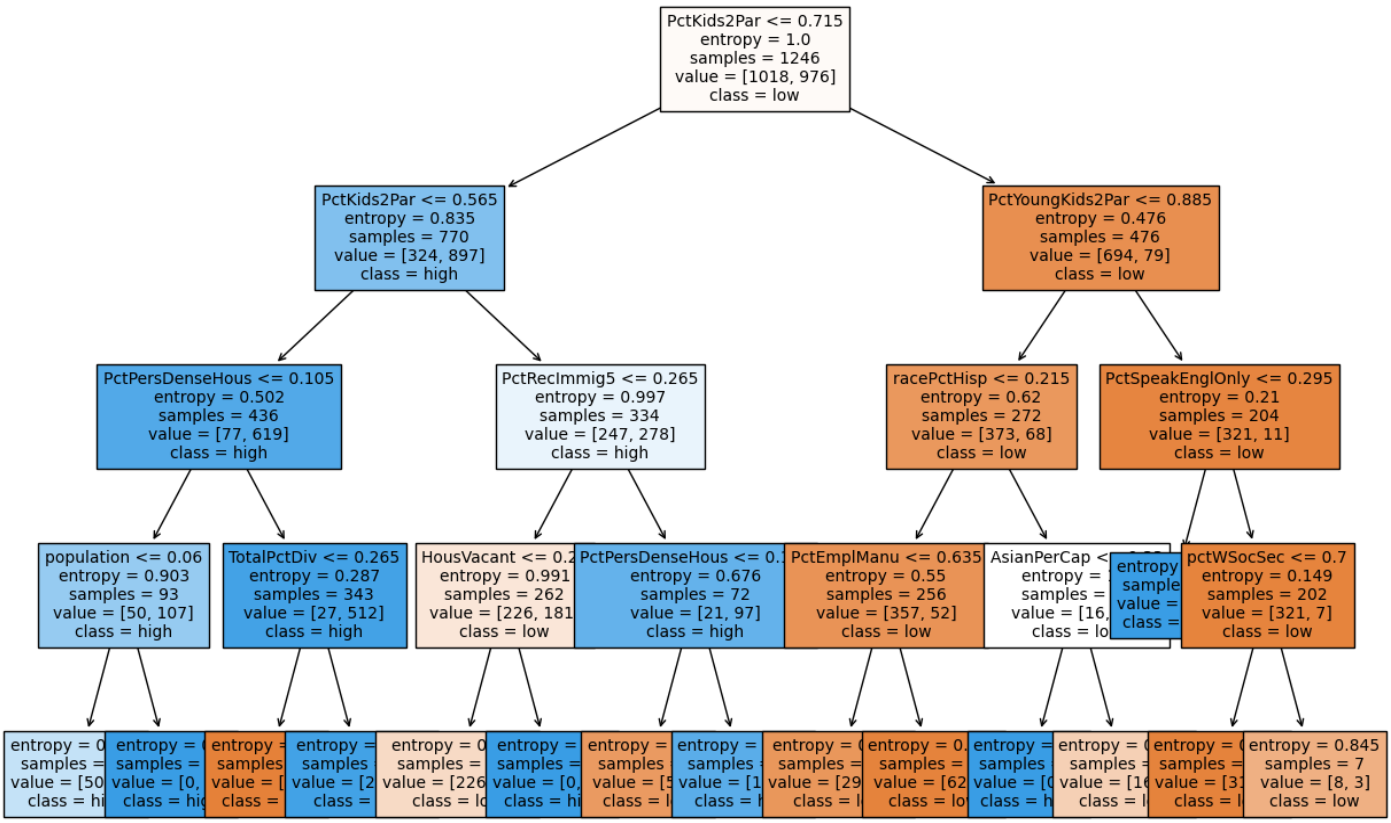


Figure 18 Random Forest of Method 2

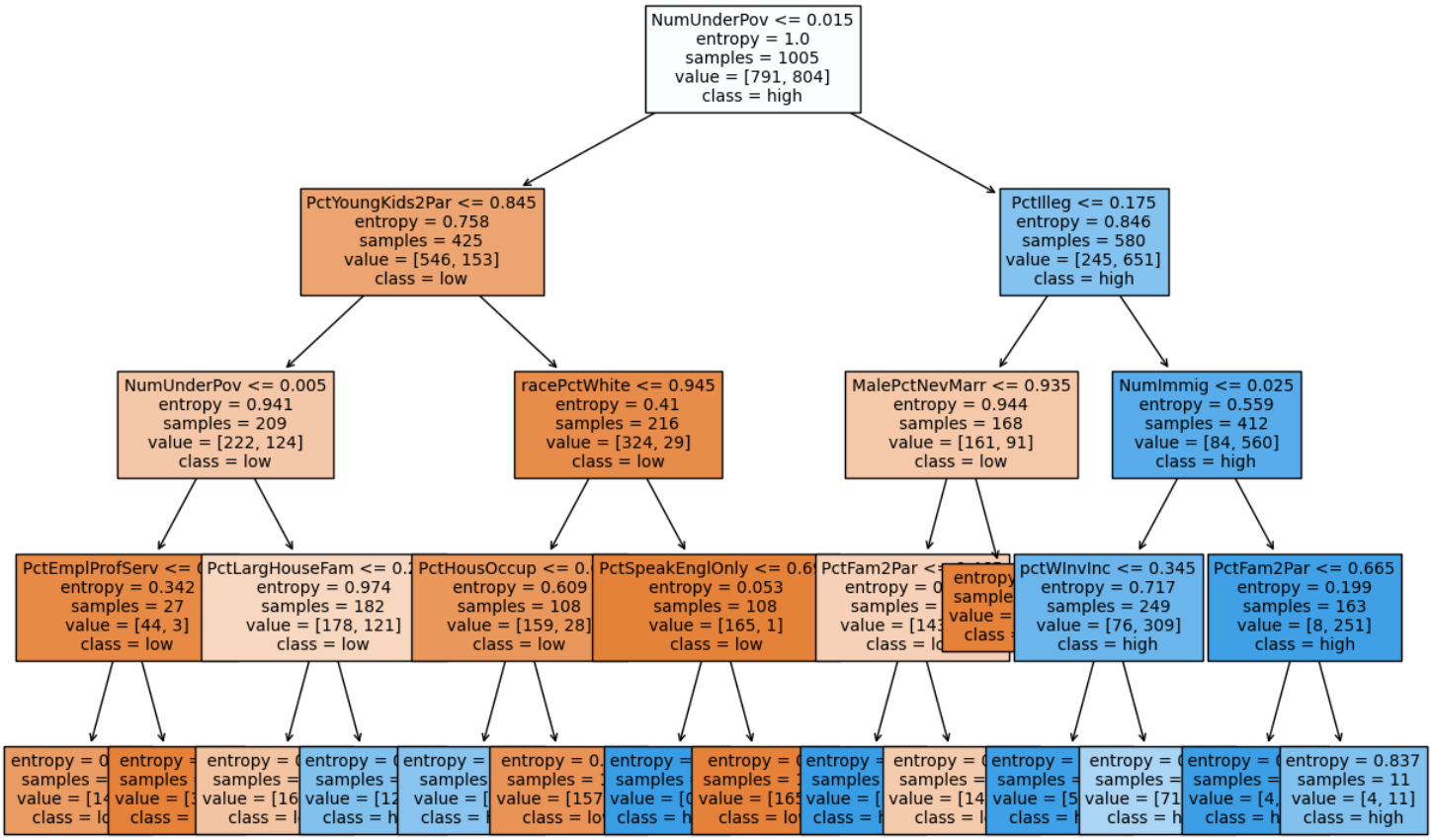


Figure 20 Random Forest of CV (left)

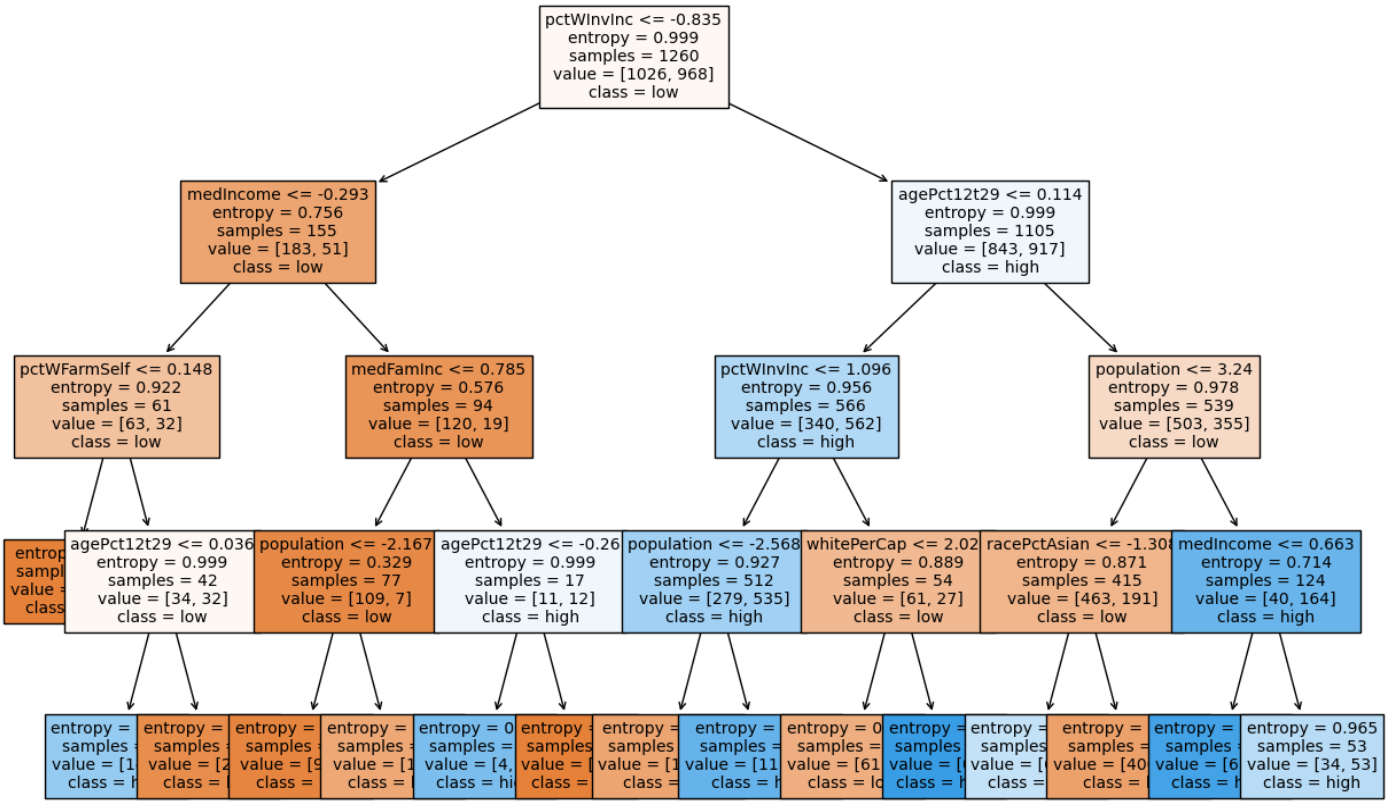


Figure 20 Training/Testing (right)