Data Warehousing and Data Mining
Assignment - 4

CLASSIFICATION
HOMICIDE DATA

Jayitha .C
20171401

Thapaswini
20161066

DATA SET DESCRIPTION

Year	Month	Day	Race	Age	Sex	City	State	Latitude	Longitude	Disposition
2007	1	1	Asian	10	Female	Albuquerque	AL	00.00	000.00	Closed by Arrest
2008	2	2	Black	20	Male	Atlanta	AZ	25.73	-122.51	Closed without Arrest
2009	3	3	Hispanic	30	S_Unknown	Baltimore	CA	25.74	-122.5	Open/ No Arrest
2010	4	.	White	40		.	CO	25.75	-122.49	
2011	5	.	Other	50		.	DC	.	.	
2012	6	.	R_Unknown	60		.	.	.	.	
2013	7	.		70		.	.	.	.	
2014	8	.		80		.	.	.	.	
2015	9	.		90		Stockton	.	45.03	.	
2016	10	29		100		Tampa	TX	45.04	-71.05	
2017	11	30		100		Tulsa	VA	45.05	-71.04	
	12	31		-500		Washington	WI	45.06	-71.02	

```
     Year           Month            Date          victim_race
Min.   :2007   Min.   : 1.00   Min.   :  1.00   Asian   :  685
1st Qu.:2010   1st Qu.: 4.00   1st Qu.:  8.00   Black   :33361
Median :2012   Median : 7.00   Median : 16.00   Hispanic: 6901
Mean   :2012   Mean   : 6.67   Mean   : 15.83   Other   :  700
3rd Qu.:2015   3rd Qu.: 9.00   3rd Qu.: 23.00   Unknown : 4199
Max.   :2017   Max.   :12.00   Max.   :105.00   White   : 6333

  victim_age         victim_sex              city              state
Min.   :-500.000   Female : 7209   Chicago     : 5535   CA     : 6288
1st Qu.:  21.000   Male   :40739   Philadelphia: 3037   TX     : 5891
Median :  27.000   Unknown: 4231   Houston     : 2942   IL     : 5535
Mean   :   1.236                   Baltimore   : 2827   PA     : 3668
3rd Qu.:  39.000                   Detroit     : 2519   MO     : 2867
Max.   : 102.000                   Los Angeles : 2257   MD     : 2827
                                   (Other)     :33062   (Other):25103
              disposition
Closed by arrest     :25674
Closed without arrest: 2922
Open/No arrest       :23583
```

The 2 images perfectly summarise the homicide data set

DATA PREPROCESSING

We have removed the Latitude and Longitude columns from the data set. This should be taken care of by the city and state attributes. We aren't going to a granular level to perform the classifications. We have also removed the names as we do not look for classification based on names. To enable the treating age as an integer, we replaces "Unknown" by -300

CLASSIFICATION TARGETS

We have tried to perform classification to predict the following

- Disposition - To build a classifier that is capable of predicting if given a set of variables, if the case is going to close or remain opened
- Race - Build a classifier to see if the race can be detected, i.e. to see if there is some kind of inherent racial bias to homicides

Listed below are the other parameters and also reasons stating why they haven't been chosen

- Date - Primarily because homicides are rare events within years and also this is usually a measure parameter
- Age - Again this is a measured parameter and going by the dataset the number of samples is too far and few to be able to produce results of any kind. Also we might want to consider a Regressive approach instead of a classification for this task
- City - Again data provided is simply insufficient
- State - This is a redundant parameter when the city is known, it's all now a matter of mapping the city to the state

CLASSIFICATION ALGORITHMS

DECISION TREES

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

A decision tree consists of three types of nodes:

1. Decision nodes
2. Chance nodes
3. End nodes – Targets

We decided to use decision trees for this assignment specifically because decision trees are very good at handling categorical data which is mostly what we have in this data set

```
1.compute the entropy for data-set

2.for every attribute/feature:
       1.calculate entropy for all categorical values
       2.take average information entropy for the current
attribute
       3.calculate gain for the current attribute

3. pick the highest gain attribute.
4. Repeat until we get the tree we desired.
```

Splitting measures:

- Information gain

$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

- Gini

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

In this project we have used both splitting measures, and we have only presented results for whichever one performed better

RANDOM FORESTS

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.Random decision forests correct for decision trees' habit of overfitting to their training set.

Algorithm 1 Random Forest

Precondition: A training set $S := (x_1, y_1), \ldots, (x_n, y_n)$, features F, and number of trees in forest B.

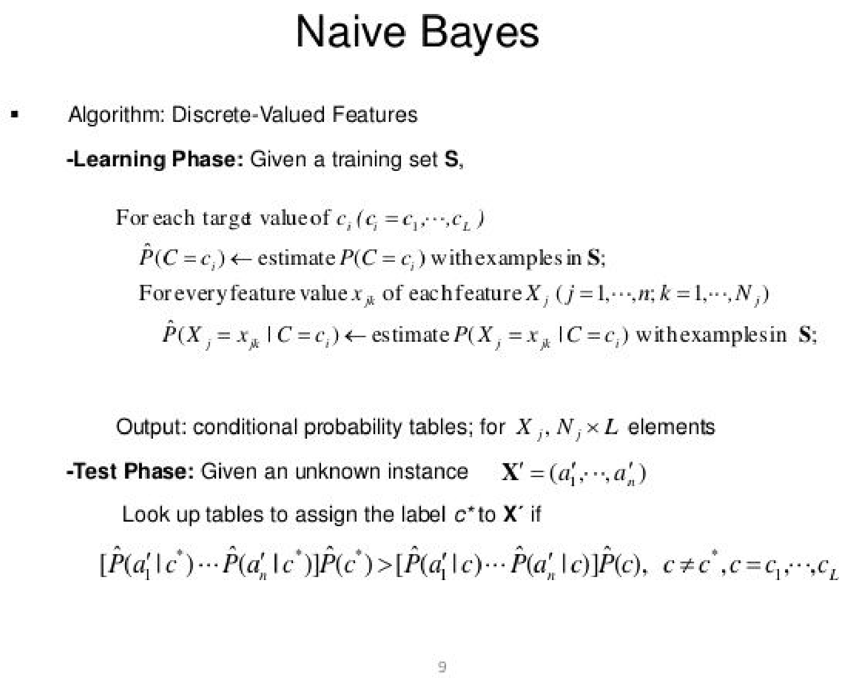
function RandomForest(S, F)
    $H \leftarrow \emptyset$
    for $i \in 1, \ldots, B$ **do**
        $S^{(i)} \leftarrow$ A bootstrap sample from S
        $h_i \leftarrow$ RandomizedTreeLearn($S^{(i)}, F$)
        $H \leftarrow H \cup \{h_i\}$
    end for
    return H
end function
function RandomizedTreeLearn(S, F)
    At each node:
        $f \leftarrow$ very small subset of F
        Split on best feature in f
    return The learned tree
end function

NAIVE BAYES CLASSIFIER

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

We have also chosen to perform naive bayes on this data set as the data set is such that there seem to be an uneven number of samples and the conditional probability for a specific class that is less in number might be caught by the naive bayes and not by any other classification algorithm

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$



DATA PREPARATION

We split the data into training and testing data, out of the 52180, we used 80% (41,000) of the samples to train and 20% to test. Unless and until specified the parameters set are the default parameter used by the package.

WHY WE CHOSE THESE ALGORITHMS ?

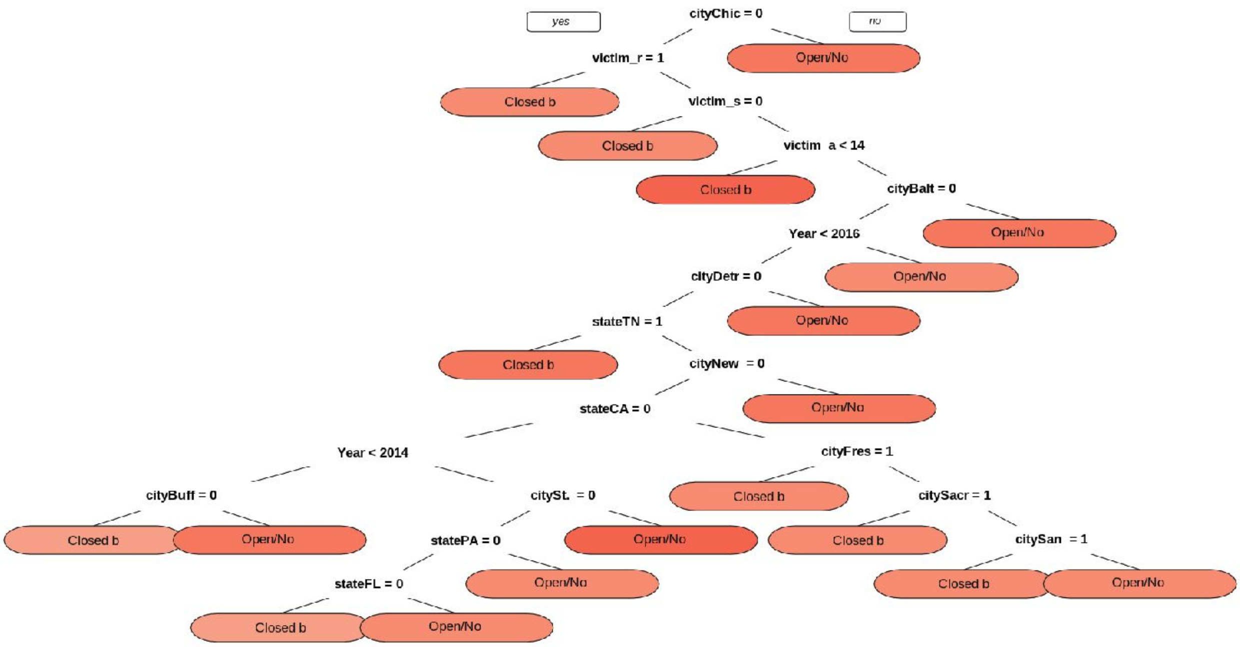
We primarily chose decision trees because they are very good at handling categorical data. We chose to improve those results using random forests because they prevent overfitting of the classifier. And we chose Naive Bayes because from previous assignments we noticed that the dataset itself was biased in some way and that some classes might be hard to find due to insufficient data samples
And so naive bayes might possibly catch them (conditional probabilities) although an MLE approach might work better

TOOLS USED

We used R to run these tests. R packages such as `carat` and `randomForests` have implementations of decision tree algorithm and random forests algorithm. We used another tool `weka` to perform naive bayes on the data. In the following sections we proceed to present the results we got from each algorithm

RESULTS

CLASSIFICATION ALGORITHM - DECISION TREES
CLASSIFICATION TARGET - DISPOSITION



The decision tree algorithm generates the decision tree shown above, the accuracy for the training set is as shown below

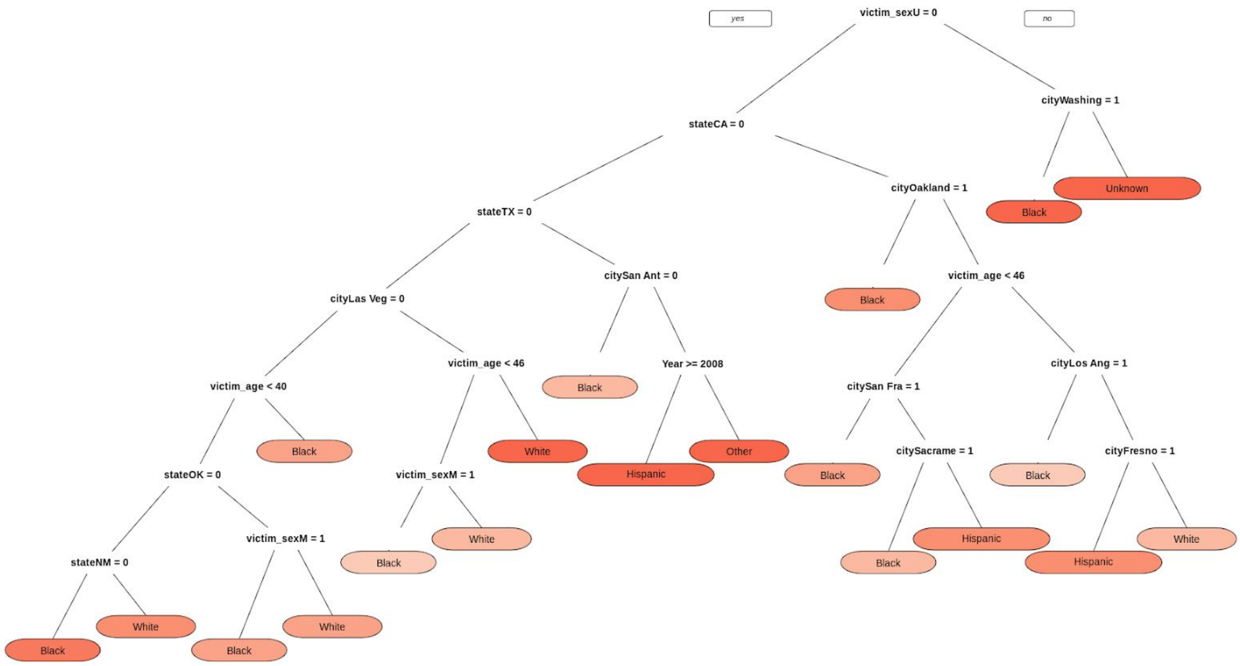
```
CART

46962 samples
    8 predictor
    3 classes: 'Closed by arrest', 'Closed without arrest', 'Open/No arrest'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 42267, 42265, 42265, 42266, 42266, 42266, ...
Resampling results across tuning parameters:

  cp           Accuracy   Kappa
  0.001509118  0.5864673  0.21296093
  0.001760637  0.5843806  0.20779430
  0.001991197  0.5831598  0.20610576
  0.002179837  0.5814777  0.20253828
  0.002473276  0.5799162  0.19925885
  0.002766716  0.5786243  0.19497989
  0.003605114  0.5768213  0.19114303
  0.004066233  0.5749262  0.18843553
  0.012198700  0.5495577  0.12932791
  0.083420667  0.5157639  0.05134657

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.001509118.
```

The **accuracy is around 58%** and these results aren't very bad but again one of the classes has an insufficient number of samples in it, this makes it al the more harder to be able to classify this data set. Below is a summarisation of the results when we ran the classifier on the test data

```
Confusion Matrix and Statistics

                       Reference
Prediction              Closed by arrest Closed without arrest Open/No arrest
  Closed by arrest                  1845                   221           1117
  Closed without arrest                0                     0              0
  Open/No arrest                     722                    71           1241

Overall Statistics

               Accuracy : 0.5915
                 95% CI : (0.578, 0.6049)
    No Information Rate : 0.492
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2198
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Closed by arrest Class: Closed without arrest
Sensitivity                           0.7187                      0.00000
Specificity                           0.4951                      1.00000
Pos Pred Value                        0.5796                          NaN
Neg Pred Value                        0.6450                      0.94403
Prevalence                            0.4920                      0.05597
Detection Rate                        0.3537                      0.00000
Detection Prevalence                  0.6101                      0.00000
Balanced Accuracy                     0.6069                      0.50000
                     Class: Open/No arrest
Sensitivity                         0.5263
Specificity                         0.7226
Pos Pred Value                      0.6101
Neg Pred Value                      0.6491
Prevalence                          0.4520
Detection Rate                      0.2379
Detection Prevalence                0.3899
Balanced Accuracy                   0.6245
```

As can be seen, **the accuracy is around 59%** and also that `Closed Without Arrest` is neither there in the decision tree nor in the test data set provided, so we do not have sufficient information to be able to predict if a given case is going to be closed without arrest

CLASSIFICATION ALGORITHM - DECISION TREES
CLASSIFICATION TARGET - RACE



Immediately we notice that, `Asians` are never predicted by the decision tree, lack of samples is always a huge problem, we could correct this by possibly letting the tree get deeper and maybe try to fit the data more, but based on the data shown below this is the most ***practically feasible and trainable tree for this data set***

```
41745 samples
    8 predictor
    6 classes: 'Asian', 'Black', 'Hispanic', 'Other', 'Unknown', 'White'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 37570, 37571, 37571, 37571, 37570, 37570, ...
Resampling results across tuning parameters:

  cp            Accuracy   Kappa
  0.0006907545  0.7391385  0.4337346
  0.0010626993  0.7388910  0.4328423
  0.0013062345  0.7380046  0.4288823
  0.0019925611  0.7373898  0.4270390
  0.0028560043  0.7361921  0.4201047
  0.0039851222  0.7345071  0.4131185
  0.0041843783  0.7338923  0.4118500
  0.0047489373  0.7322155  0.4087108
  0.0052692171  0.7228887  0.3526076
  0.2193809777  0.6786684  0.1528829

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.0006907545.
```

From the image we see that we get an **accuracy of about 73%,** and clearly this means that there is some racial bias among the data, i.e. there are some inherent properties that are specific to a given race and we can extract these patterns directly from the decision trees. Here’s how it performed on the test data:

```
Confusion Matrix and Statistics

          Reference
Prediction Asian Black Hispanic Other Unknown White
  Asian        0     0        0     0       0     0
  Black       83  6270      905    90       5  1050
  Hispanic    28   311      422    16       2    85
  Other        0     8        1    16       0     6
  Unknown      1     0        0     3     831     1
  White       25    83       52    15       1   124

Overall Statistics

               Accuracy : 0.7344
                 95% CI : (0.7258, 0.7429)
    No Information Rate : 0.6394
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4278
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Asian Class: Black Class: Hispanic Class: Other
Sensitivity               0.00000       0.9397         0.30580     0.114286
Specificity               1.00000       0.4330         0.95118     0.998543
Pos Pred Value                NaN       0.7462         0.48843     0.516129
Neg Pred Value            0.98687       0.8021         0.89990     0.988080
Prevalence                0.01313       0.6394         0.13226     0.013418
Detection Rate            0.00000       0.6009         0.04044     0.001533
Detection Prevalence      0.00000       0.8053         0.08281     0.002971
Balanced Accuracy         0.50000       0.6864         0.62849     0.556414
                     Class: Unknown Class: White
Sensitivity                 0.99046      0.09795
Specificity                 0.99948      0.98080
Pos Pred Value              0.99402      0.41333
Neg Pred Value              0.99917      0.88731
Prevalence                  0.08041      0.12133
Detection Rate              0.07964      0.01188
Detection Prevalence        0.08012      0.02875
Balanced Accuracy           0.99497      0.53937
```

We were able to achieve a **good accuracy of around 73%**. The decision tree predicting vicin race is particularly fascinating as the first deciding factor itself is if the sex is `unknown` and if so and if the victim is from `Washington` then the person is predicted to be `Black`

CLASSIFICATION ALGORITHM - RANDOM FORESTS
CLASSIFICATION TARGET - DISPOSITION

The results from running Random Forest algorithms are summarised below

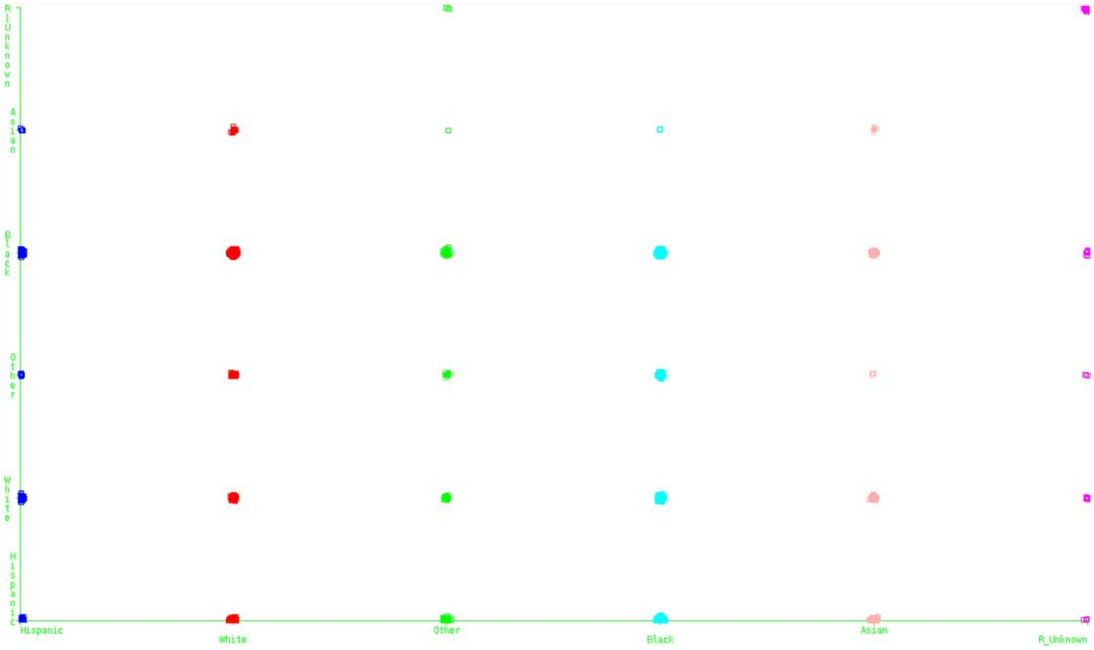
```
 randomForest(formula = disposition ~ ., data = training)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 39.04%
Confusion matrix:
                      Closed by arrest Closed without arrest Open/No arrest
Closed by arrest                 14187                    29           6324
Closed without arrest             1483                   116            739
Open/No arrest                    7708                    14          11145
                      class.error
Closed by arrest        0.3092989
Closed without arrest   0.9503849
Open/No arrest          0.4092861
            MeanDecreaseGini
Year               1837.1905
Month              1926.4787
Date               2804.1534
victim_race         797.0467
victim_age         3140.4042
victim_sex          406.2508
city               1807.1669
state              1383.0889
```

Above we also see the importance of each attribute wrt the Gini Index of that attribute. Again we notice the same pattern, the inability to be able to classify under `Closed Without Arrest`

CLASSIFICATION ALGORITHM - NAIVE BAYES CLASSIFIER
CLASSIFICATION TARGET - RACE

When we ran Naive Bayes with disposition as target we got bad results and hence we haven't presented said results in the report

Below is the confusion matrix when we used weka to apply naive bayes to classify race



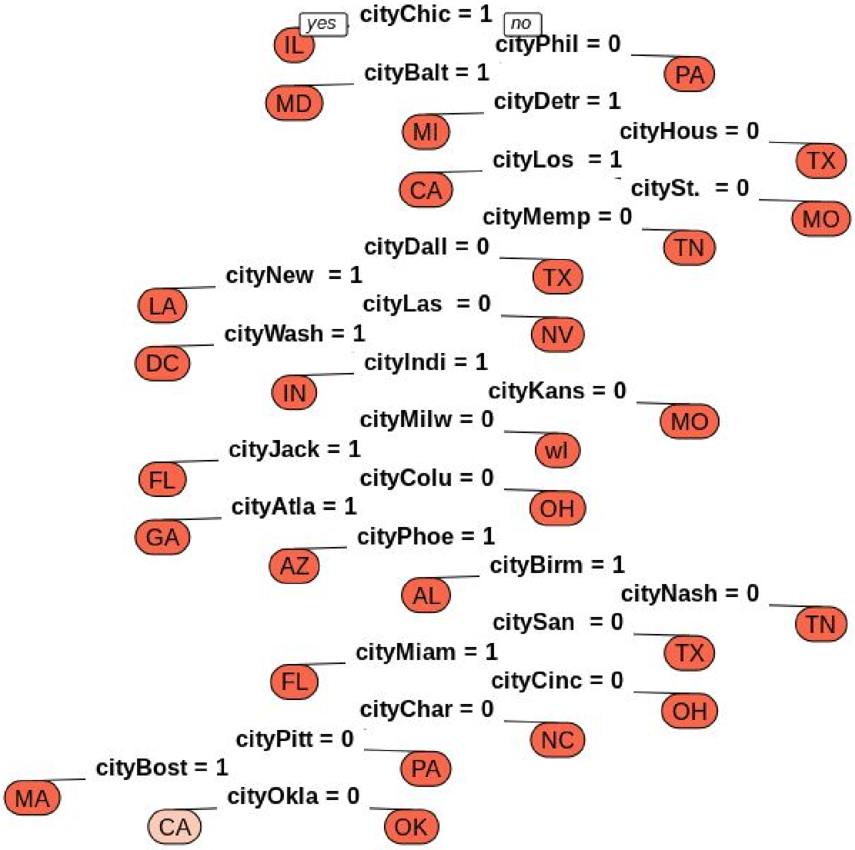
Below are the results summarised for the testing data

```
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances       14121               79.5953 %
Incorrectly Classified Instances      3620               20.4047 %
Kappa statistic                          0.6266
Mean absolute error                      0.0815
Root mean squared error                  0.2272
Relative absolute error                 44.3233 %
Root relative squared error             75.1262 %
Total Number of Instances            17741

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.794     0.089      0.576     0.794     0.668      0.939    Hispanic
                 0.373     0.046      0.522     0.373     0.435      0.84     White
                 0.194     0.007      0.263     0.194     0.223      0.864    Other
                 0.878     0.219      0.878     0.878     0.878      0.913    Black
                 0.073     0.001      0.474     0.073     0.126      0.88     Asian
                 0.985     0          0.998     0.985     0.992      0.999    R_Unknown
Weighted Avg.    0.796     0.158      0.792     0.796     0.788      0.914

=== Confusion Matrix ===

     a     b     c     d     e     f   <-- classified as
  1865   152     6   322     4     0 |     a = Hispanic
   317   786    33   958    12     0 |     b = White
    86    37    41    43     1     3 |     c = Other
   848   474    71 10006     3     0 |     d = Black
   116    51     2    60    18     0 |     e = Asian
     4     6     3     8     0  1405 |     f = R_Unknown
```

Naive Bayes does fairly well on the test set, we get an **accuracy of around 80%** when we performed it on Race. Clearly this data set favours Naive Bayes Algorithm

FOR FUN

As a fun experiment we decided to try to predict the state given the city using the decision tree algorithms and it turns out it produces the right results with high ***accuracy of about 83%***



CONCLUSION

In conclusion we were able to generate classifiers to predict both the disposition (`Closed without Arrest`, `Closed With Arrest`, `Open`) and Race (`Asian`, …) with reasonable accuracies (highest found through our experiments were 59% and 80% respectively. We have established that `Closed Without Arrest` is hard to classify and also that there is possibly a lot of racial bias in the data i.e. Race plays a very huge role in deciding any factor about the data. This is also reflected in the decision trees generated

We feel we could possibly get better results using **ADA BOOSTING ON DECISION TREES.** Decision trees are weak learners and using ADA BOOST on them should collectively make them a strong learner, we were unable to do this in time but hope to do so for the next submission (if there is one)

Here provided is the link to all the codes and images used in this report for clarity - https://github.com/Jayitha/Data-Warehousing-and-Data-Mining-/