CS-GY 6923 Machine Learning

Professor: Dr. Raman Kannan

HW2: Working with Individual Classifiers

Anil Poonai

Due 5th November 2021

Contents

For homework assignment two, I did two versions of each classifier: One without the variable J and one with. The reason for that variable being the one is because in the variable importance algorithm done earlier, J was highly correlated to the classification set and didn't want that to have bias, so the actual results is the one without the variable J, while I left the version with J in to see the difference in results.

1 NaiveBayes

After the EDA was completed I started working with individual classifiers on the dataset. The first used was NaiveBayes. This follows the formula:

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

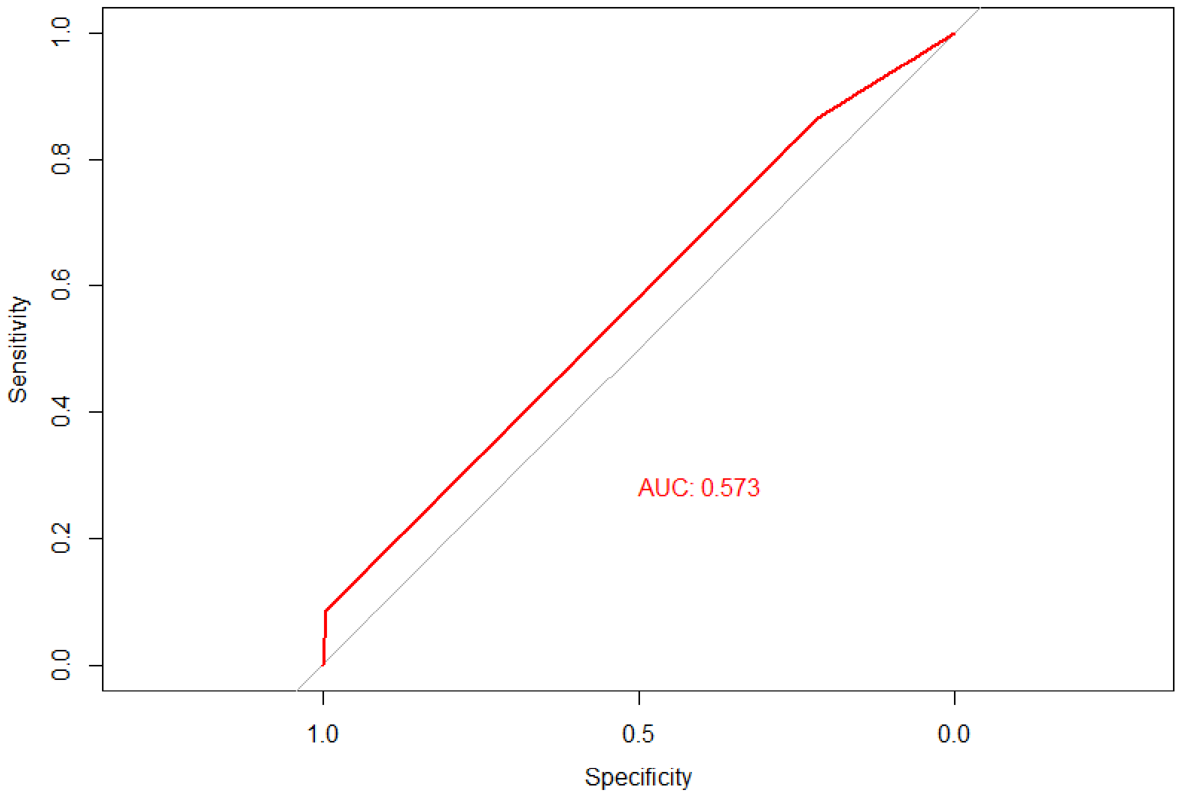
The code I used is as follow:

```
#NaiveBayes

library(e1071)
set.seed(222)
nbDefault = naiveBayes(Dependent~., data=train, prob = TRUE)
nbDefault
nbDefault_pred = predict(nbDefault, test, type="class", prob = TRUE)
nbDefault_pred
confusionMatrix(nbDefault_pred, test$Dependent)
plot(nbDefault_pred)
roc_nb_test <- roc(response = test$Dependent, predictor =as.numeric(nbDefault_pred))
plot(roc_nb_test)
plot(roc_nb_test,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#NaiveBayes (No J)
library(e1071)
set.seed(222)
nbtrain = train
nbtest = test
nbtrain$J = NULL
nbtest$J = NULL
nbDefault2 = naiveBayes(Dependent~., data=nbtrain, prob = TRUE)
nbDefault2
nbDefault_pred2 = predict(nbDefault2, nbtest, type="class", prob = TRUE)
nbDefault_pred2
confusionMatrix(nbDefault_pred2, nbtest$Dependent)
plot(nbDefault_pred2)
roc_nb_test2 <- roc(response = nbtest$Dependent, predictor =as.numeric(nbDefault_pred))
plot(roc_nb_test2)
plot(roc_nb_test2,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)
```

Which means the algorithm will classify based on the highest correlated probabilities between independent variables and the dependent variable. The confusion matrix results are below for the inclusion of variable J:

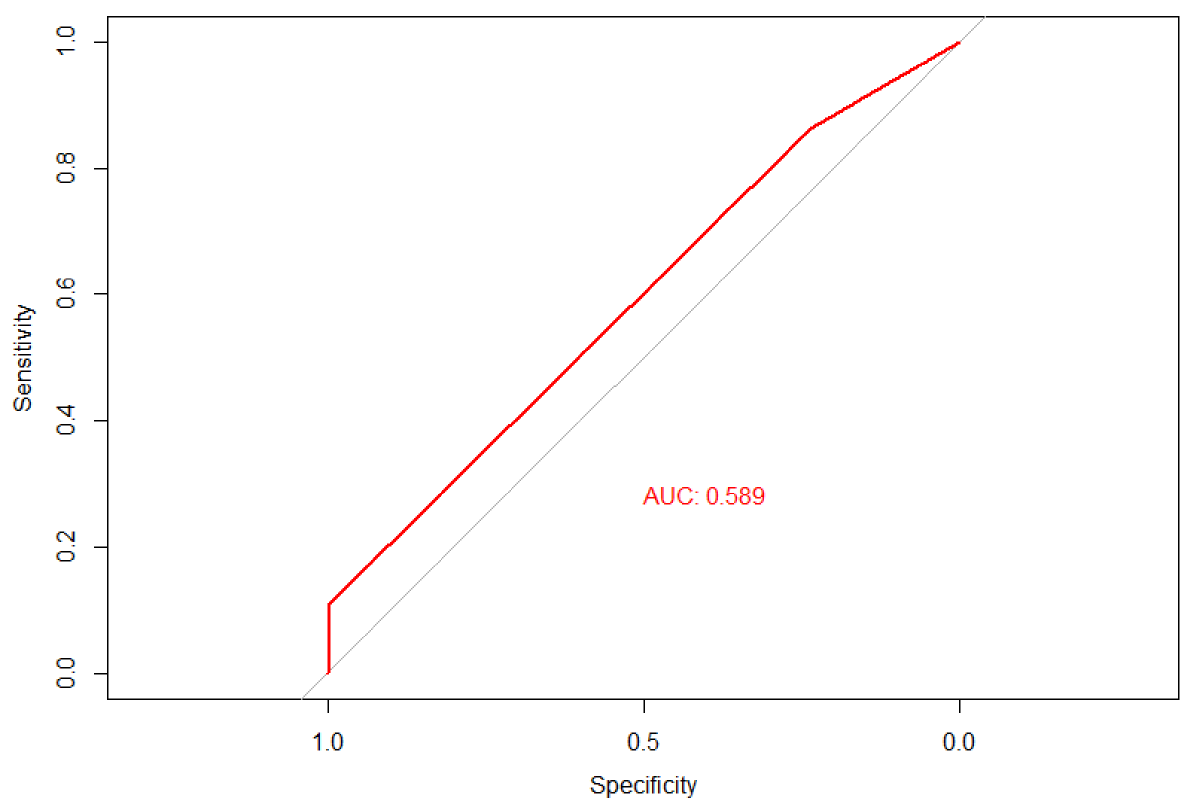
```
> confusionMatrix(nbDefault_pred, test$Dependent)
Confusion Matrix and Statistics

             Reference
Prediction    FELONY MISDEMEANOR VIOLATION
  FELONY         100         101        28
  MISDEMEANOR  19464       28534        68
  VIOLATION       99        2847     10348

Overall Statistics

               Accuracy : 0.6329
                 95% CI : (0.6291, 0.6367)
    No Information Rate : 0.5112
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3484

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

Statistics by Class:

                     Class: FELONY Class: MISDEMEANOR Class: VIOLATION
Sensitivity               0.005086             0.9064           0.9908
Specificity               0.996923             0.3512           0.9424
Pos Pred Value            0.436681             0.5936           0.7784
Neg Pred Value            0.681177             0.7820           0.9980
Prevalence                0.319262             0.5112           0.1696
Detection Rate            0.001624             0.4633           0.1680
Detection Prevalence      0.003718             0.7804           0.2159
Balanced Accuracy         0.501004             0.6288           0.9666
```

I then had the ROC curve and AUC value created for it:



And this is the results without variable J:

```
> confusionMatrix(nbDefault_pred2, nbtest$Dependent)
Confusion Matrix and Statistics

             Reference
Prediction    FELONY MISDEMEANOR VIOLATION
  FELONY        2195        2124       468
  MISDEMEANOR   5900       10750      2114
  VIOLATION     4321        7492      6436

Overall Statistics

               Accuracy : 0.4637
                 95% CI : (0.4589, 0.4685)
    No Information Rate : 0.4872
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1788

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: FELONY Class: MISDEMEANOR Class: VIOLATION
Sensitivity                0.17679             0.5278           0.7137
Specificity                0.91179             0.6261           0.6396
Pos Pred Value             0.45853             0.5729           0.3527
Neg Pred Value             0.72385             0.5826           0.8904
Prevalence                 0.29703             0.4872           0.2157
Detection Rate             0.05251             0.2572           0.1540
Detection Prevalence       0.11452             0.4489           0.4366
Balanced Accuracy          0.54429             0.5770           0.6767
>
```



2 KNN

The next algorithm used was k-nearest neighbors, which is a classification following this formula:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

This works by having all of the observations plotted and seeing which are close together, then it'll group them together based on the dependent variable and try to find common attributes to query in order to see to which group or neighborhood the observations belong.

The code I used is as follow:

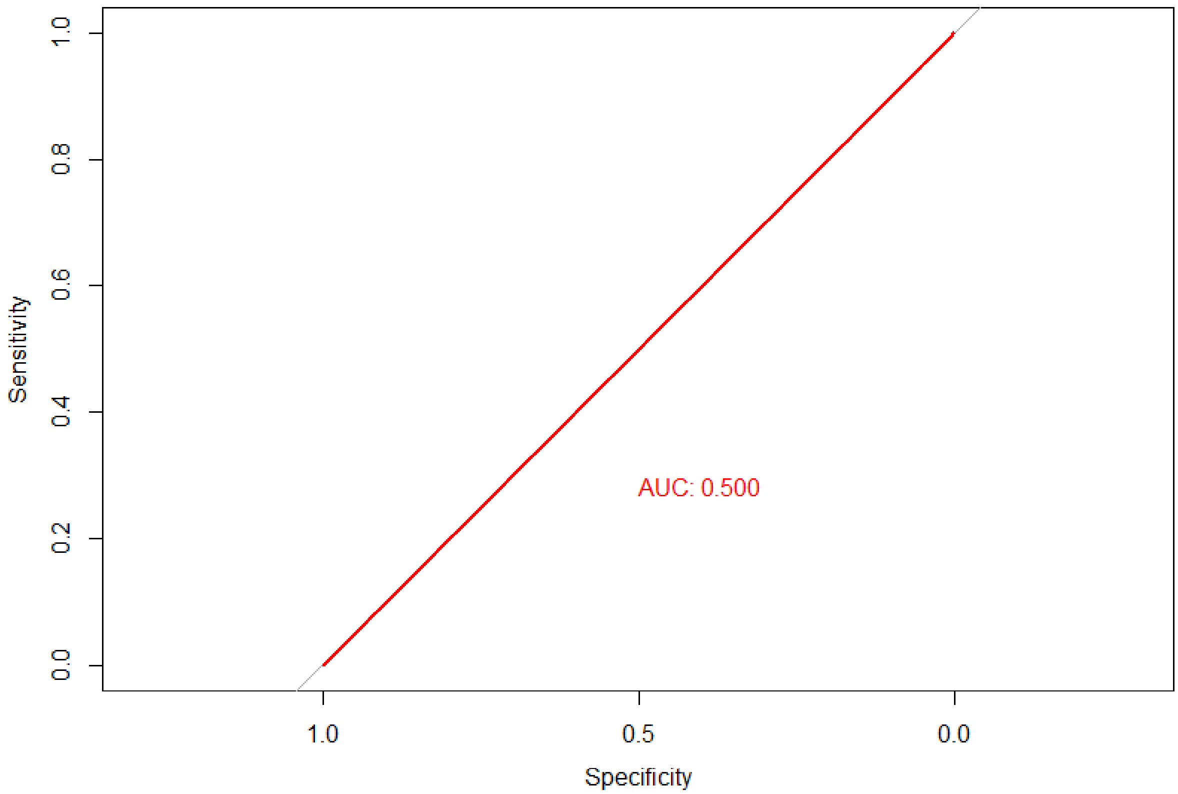
```
#KNN
library(class)
knndf = df
str(knndf)
knndf$B = as.numeric(knndf$B)
knndf$E = as.numeric(knndf$E)
knndf$G = as.numeric(knndf$G)
knndf$H = as.numeric(knndf$H)
knndf$I = as.numeric(knndf$I)
knndf$M = as.numeric(knndf$M)
knndf$N = as.numeric(knndf$N)
knndf$O = as.numeric(knndf$O)
knndf$P = as.numeric(knndf$P)
knndf$Q = as.numeric(knndf$Q)
knndf$R = as.numeric(knndf$R)
knntrain = knndf[index==1,]
knntest = knndf[index==2,]
train_knn = na.omit(knntrain)
test_knn = na.omit(knntest)
train_knn_dep = train_knn
test_knn_dep = test_knn
train_knn$Dependent=NULL
test_knn$Dependent=NULL

knnModel = knn(train = train_knn, test = test_knn, cl = train_knn_dep$Dependent, prob = TRUE)
knnModel
table(knnModel, test_knn_dep$Dependent,dnn=c("Prediction","Actual"))
sum(knnModel == test_knn_dep$Dependent)/nrow(test_knn_dep)
confusionMatrix(knnModel, test_knn_dep$Dependent)
knnroc=roc(test_knn_dep$Dependent, attributes(knnModel)$prob)
plot(knnroc,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)
```

These are the results for the variables including J:

The confusion matrix generated has these properties:

```
> confusionMatrix(knnModel, test_knn_dep$Dependent)
Confusion Matrix and Statistics

             Reference
Prediction    FELONY MISDEMEANOR VIOLATION
  FELONY        7027        4361       891
  MISDEMEANOR   4556       13940      1759
  VIOLATION      833        2065      6368

Overall Statistics

               Accuracy : 0.6539
                 95% CI : (0.6494, 0.6585)
    No Information Rate : 0.4872
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4497

 Mcnemar's Test P-Value : 9.822e-07

Statistics by Class:

                     Class: FELONY Class: MISDEMEANOR Class: VIOLATION
Sensitivity                 0.5660             0.6845           0.7061
Specificity                 0.8213             0.7054           0.9116
Pos Pred Value              0.5723             0.6882           0.6872
Neg Pred Value              0.8175             0.7017           0.9185
Prevalence                  0.2970             0.4872           0.2157
Detection Rate              0.1681             0.3335           0.1523
Detection Prevalence        0.2938             0.4846           0.2217
Balanced Accuracy           0.6936             0.6949           0.8089
```

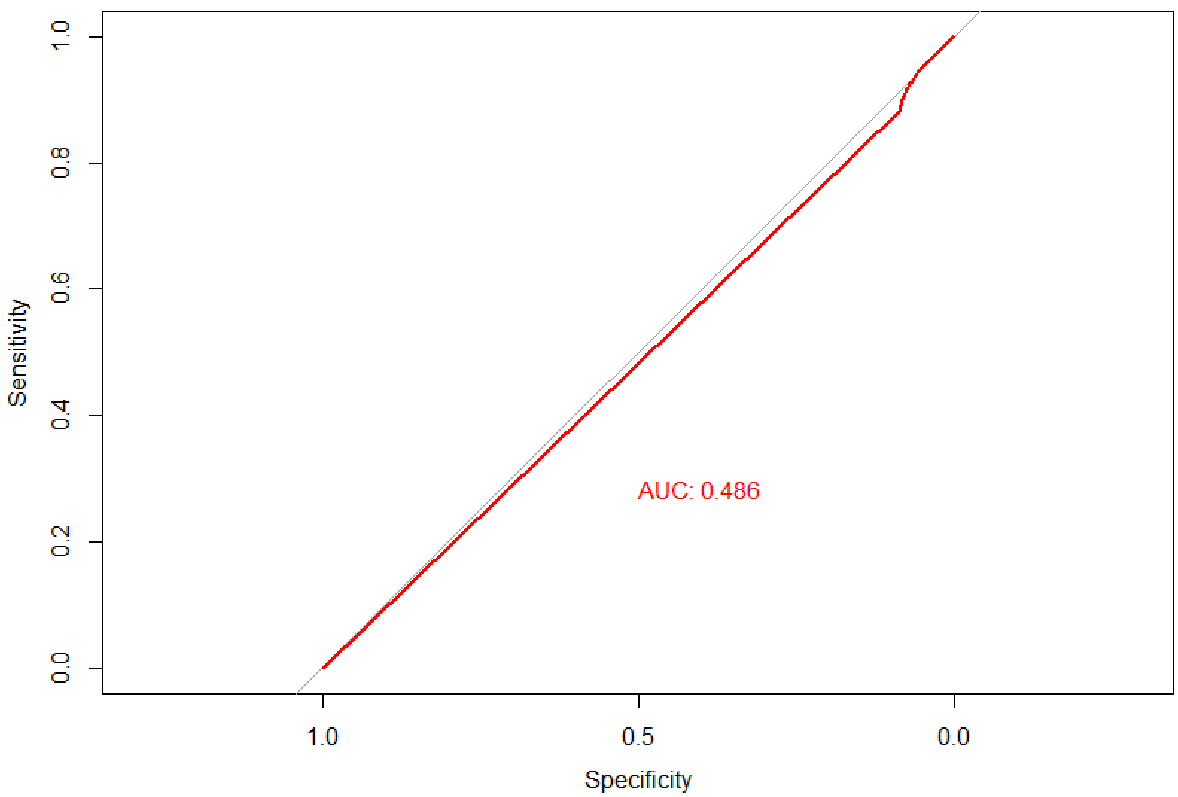
And the ROC curve and AUC value:



And these are the values without J:

```
#KNN (No J)
library(class)
knndf = df
str(knndf)
knndf$B = as.numeric(knndf$B)
knndf$E = as.numeric(knndf$E)
knndf$G = as.numeric(knndf$G)
knndf$H = as.numeric(knndf$H)
knndf$I = as.numeric(knndf$I)
knndf$M = as.numeric(knndf$M)
knndf$N = as.numeric(knndf$N)
knndf$O = as.numeric(knndf$O)
knndf$P = as.numeric(knndf$P)
knndf$Q = as.numeric(knndf$Q)
knndf$R = as.numeric(knndf$R)
knntrain2 = knndf[index==1,]
knntest2 = knndf[index==2,]
knntrain2$J = NULL
knntest2$J = NULL
train_knn2 = na.omit(knntrain2)
test_knn2 = na.omit(knntest2)
train_knn_dep2 = train_knn2
test_knn_dep2 = test_knn2
train_knn2$Dependent=NULL
test_knn2$Dependent=NULL

knnModel2 = knn(train = train_knn2, test = test_knn2, cl = train_knn_dep2$Dependent, prob = TRUE)
knnModel2
confusionMatrix(knnModel2, test_knn_dep2$Dependent)
knnroc2=roc(test_knn_dep2$Dependent, attributes(knnModel2)$prob)
plot(knnroc2)
plot(knnroc2,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#Decision Tree
```

```
> confusionMatrix(knnModel2, test_knn_dep2$Dependent)
Confusion Matrix and Statistics

             Reference
Prediction     FELONY MISDEMEANOR VIOLATION
  FELONY         4626        5177      2204
  MISDEMEANOR    5494       11546      3687
  VIOLATION      2296        3643      3127

Overall Statistics

               Accuracy : 0.4617
                 95% CI : (0.4569, 0.4665)
    No Information Rate : 0.4872
    P-Value [Acc > NIR] : 1.000000

                  Kappa : 0.1405

 Mcnemar's Test P-Value : 0.009044

Statistics by Class:

                     Class: FELONY Class: MISDEMEANOR Class: VIOLATION
Sensitivity                 0.3726             0.5669          0.34675
Specificity                 0.7488             0.5717          0.81883
Pos Pred Value              0.3853             0.5571          0.34492
Neg Pred Value              0.7385             0.5815          0.82003
Prevalence                  0.2970             0.4872          0.21574
Detection Rate              0.1107             0.2762          0.07481
Detection Prevalence        0.2872             0.4959          0.21689
Balanced Accuracy           0.5607             0.5693          0.58279
```



3 Decision Trees

This algorithm divides the data into nodes and will use Boolean logic for attributes to create new nodes. It doesn't really have a formula but it does have a sequential order to find the results.

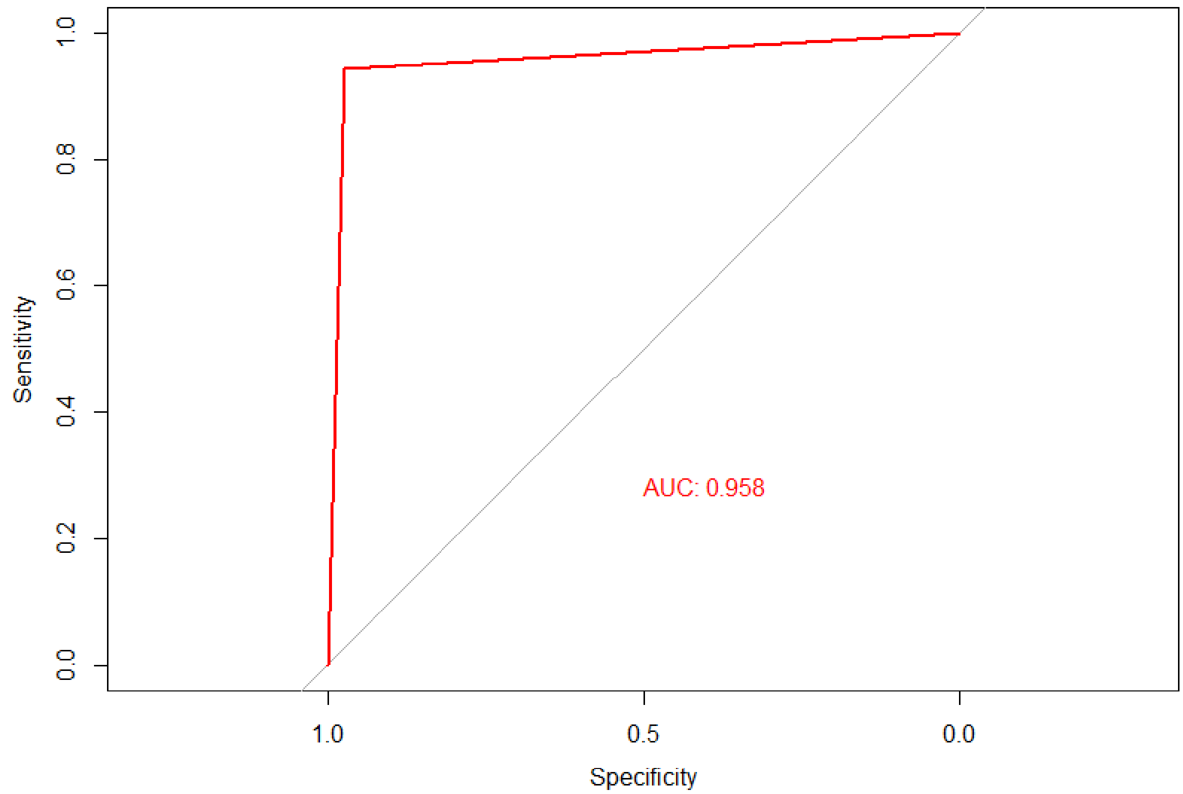
Below is the results with all of the variables:

```
#Decision Tree
library(rpart)
library(rpart.plot)
dt=rpart(Dependent~., data=train, method='class')
rpart.plot(dt)
dt
printcp(dt)
dtpred = predict(dt,test,type = 'class')
dtpred
confusionMatrix(dtpred, test$Dependent)
dtroc=roc(test$Dependent, as.numeric(dtpred))
plot(dtroc)
plot(dtroc,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#Decision Tree (No J)
library(rpart)
library(rpart.plot)
dttrain = train
dttrain$J= NULL
dttest = test
dttest$J= NULL
dt2=rpart(Dependent~., data=dttrain, method='class')
rpart.plot(dt2,box.palette = "blue")
dt2
printcp(dt2)
dtpred2 = predict(dt2,dttest,type = 'class')
dtpred2
confusionMatrix(dtpred2, test$Dependent)
dtroc2=roc(test$Dependent, as.numeric(dtpred2))
plot(dtroc2)
plot(dtroc2,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#SVM
```

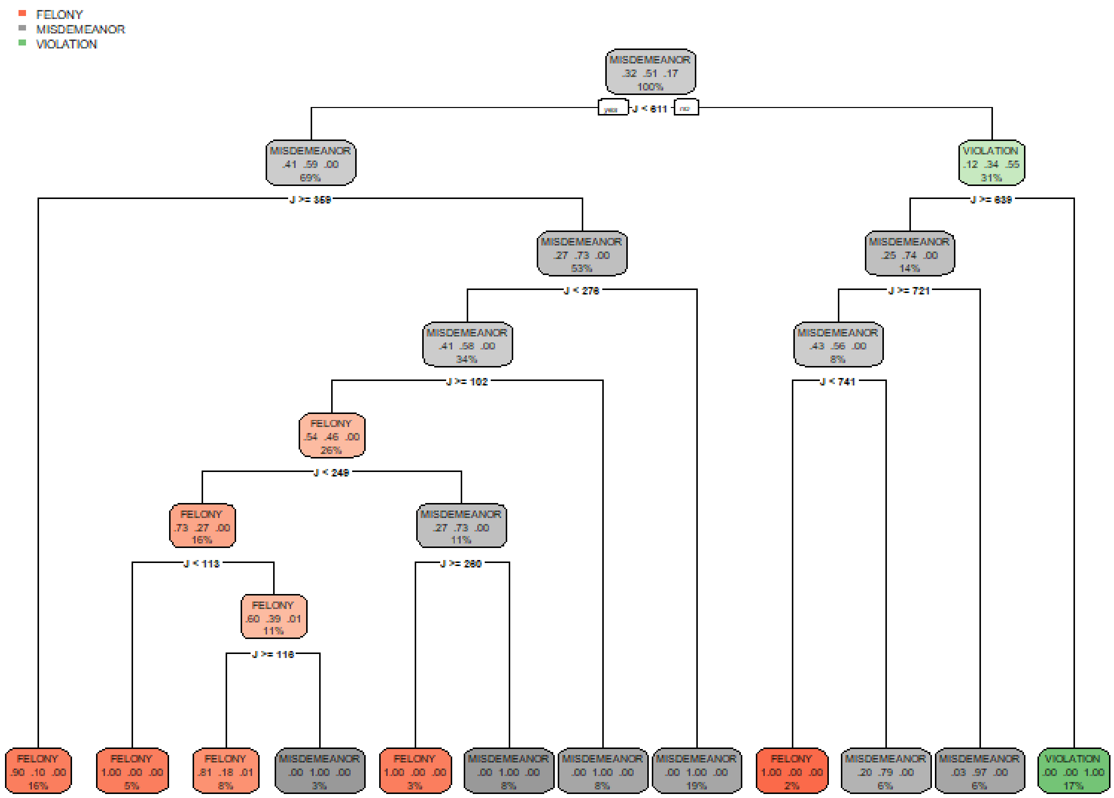
This is the confusion matrix value:

```
> confusionMatrix(dtpred, test$Dependent)
Confusion Matrix and Statistics

             Reference
Prediction    FELONY MISDEMEANOR VIOLATION
  FELONY       12094        1185        40
  MISDEMEANOR    322       19180         4
  VIOLATION        0           1      8974

Overall Statistics

               Accuracy : 0.9629
                 95% CI : (0.961, 0.9647)
    No Information Rate : 0.4872
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9412

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

Statistics by Class:

                     Class: FELONY Class: MISDEMEANOR Class: VIOLATION
Sensitivity                 0.9741             0.9418           0.9951
Specificity                 0.9583             0.9848           1.0000
Pos Pred Value              0.9080             0.9833           0.9999
Neg Pred Value              0.9887             0.9468           0.9987
Prevalence                  0.2970             0.4872           0.2157
Detection Rate              0.2893             0.4589           0.2147
Detection Prevalence        0.3186             0.4667           0.2147
Balanced Accuracy           0.9662             0.9633           0.9975
```

This is the ROC curve with the AUC value:

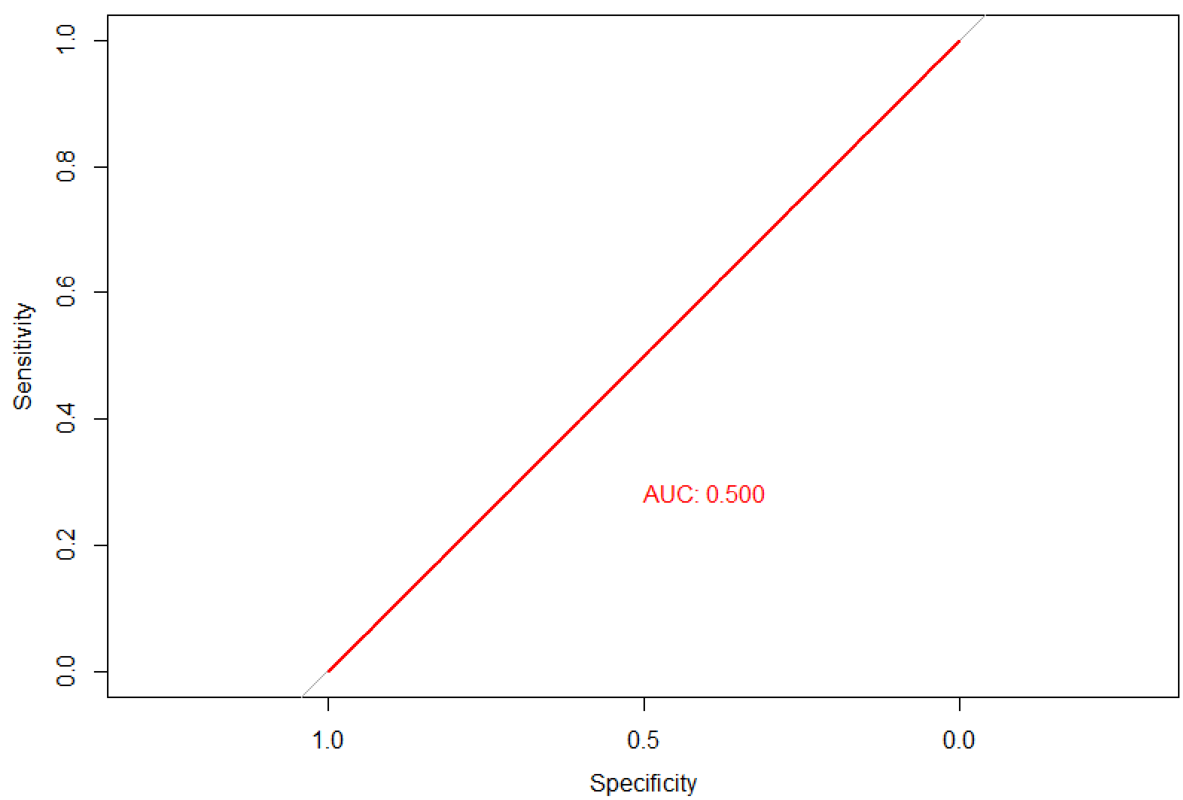


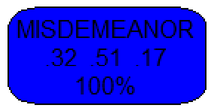
Map of the Tree:



This is the value without the J variable:

```
> confusionMatrix(dtpred2, test$Dependent)
Confusion Matrix and Statistics

             Reference
Prediction     FELONY MISDEMEANOR VIOLATION
  FELONY            0           0         0
  MISDEMEANOR   12416       20366      9018
  VIOLATION         0           0         0

Overall Statistics

               Accuracy : 0.4872
                 95% CI : (0.4824, 0.492)
    No Information Rate : 0.4872
    P-Value [Acc > NIR] : 0.5019

                  Kappa : 0

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: FELONY Class: MISDEMEANOR Class: VIOLATION
Sensitivity                  0.000             1.0000           0.0000
Specificity                  1.000             0.0000           1.0000
Pos Pred Value                 NaN             0.4872              NaN
Neg Pred Value               0.703                NaN           0.7843
Prevalence                   0.297             0.4872           0.2157
Detection Rate               0.000             0.4872           0.0000
Detection Prevalence         0.000             1.0000           0.0000
Balanced Accuracy            0.500             0.5000           0.5000
>
```




MISDEMEANOR
.32 .51 17
100%

4 SVM

The last one used was the support vector machine, which is taking each observation as data points but in vector form. We then have the support vectors which are the points closest to the hyperplane, which is the line or plane that best separates the data points. The hyperplane that best separates will be the one with the greatest margin between the hyperplane and any data point. SVM uses a kernel trick to increase the dimension of the space. So it ends up as a calculation of the dot product of x transpose and y transpose. The kernel function used in my analysis is radial and follows the formula:

$$K(x,y) = exp(-\gamma \sum_{j=1}^{p}(x_{ij} - y_{ij})^2)$$

I used this function as this is the best kernel used for a prediction variable with more than two classes.

Below is the results with all of the variables:

```
#SVM
str(train)
train$B = NULL
train$E = NULL
train$G = NULL
train$H = NULL
train$I = NULL
train$M = NULL
train$N = NULL
train$O = NULL
train$P = NULL
train$Q = NULL
train$R = NULL
test_svm= na.omit(test)

svmfit = svm(Dependent ~ ., data = train, probability = TRUE, type = "C-classification", kernal = "radial", gamma
svmpred <- predict(svmfit, test, decision.values = TRUE, probability = TRUE, type ='class')
svmfit
svmpred
confusionMatrix(svmpred, test_svm$Dependent)

roc_svm_test <- roc(response = test_svm$Dependent, predictor =as.numeric(svmpred))
plot(roc_svm_test,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)
```

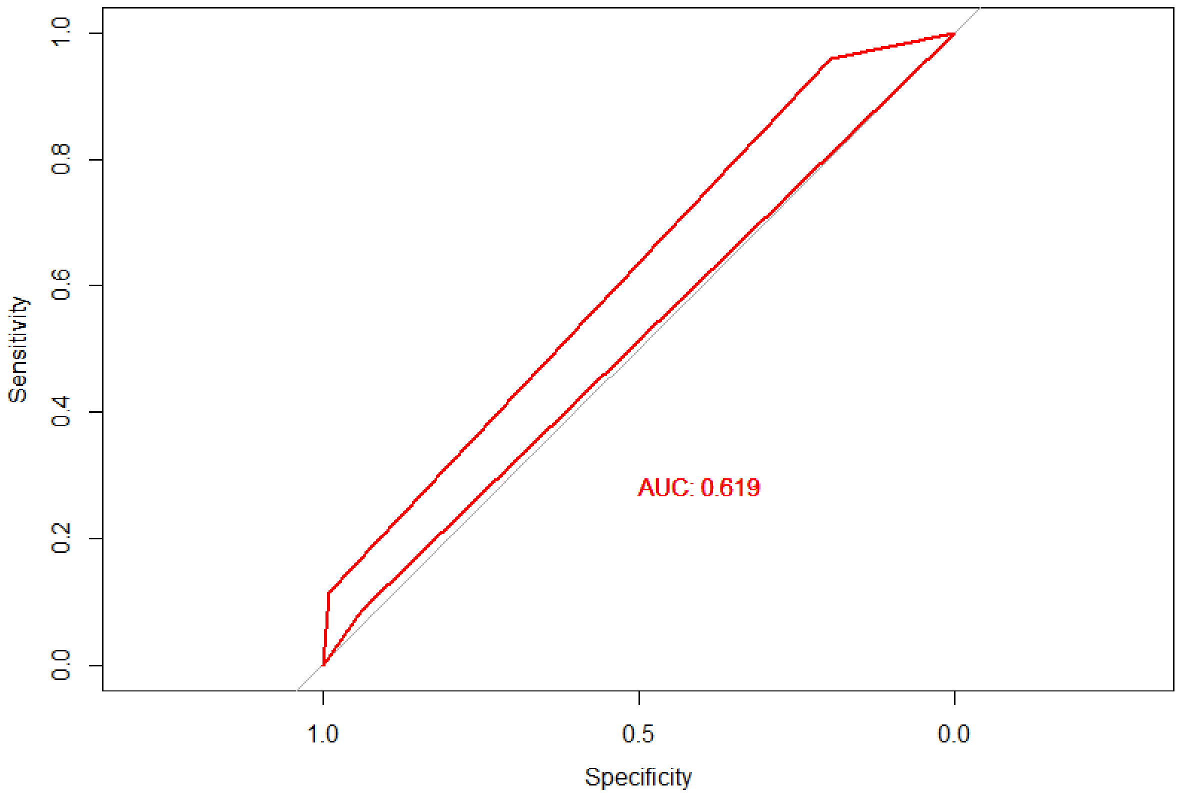
The SVM generated the following parameters from the training data:

```
Call:
svm(formula = Dependent ~ ., data = train, probability = TRUE, type = "C-classification", kernal = "radial",
    gamma = 0.1, cost = 1)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  103944
```

After applying this model to the predict function on the testing data and making a confusion matrix:

```
Confusion Matrix and Statistics

             Reference
Prediction    FELONY MISDEMEANOR VIOLATION
  FELONY        2433         840         3
  MISDEMEANOR   9863       17162        51
  VIOLATION      120        2364      8964

Overall Statistics

               Accuracy : 0.6832
                 95% CI : (0.6787, 0.6877)
    No Information Rate : 0.4872
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4738

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

Statistics by Class:

                     Class: FELONY Class: MISDEMEANOR Class: VIOLATION
Sensitivity                0.19596             0.8427           0.9940
Specificity                0.97131             0.5375           0.9242
Pos Pred Value             0.74267             0.6338           0.7830
Neg Pred Value             0.74086             0.7824           0.9982
Prevalence                 0.29703             0.4872           0.2157
Detection Rate             0.05821             0.4106           0.2144
Detection Prevalence       0.07837             0.6478           0.2739
Balanced Accuracy          0.58363             0.6901           0.9591
```

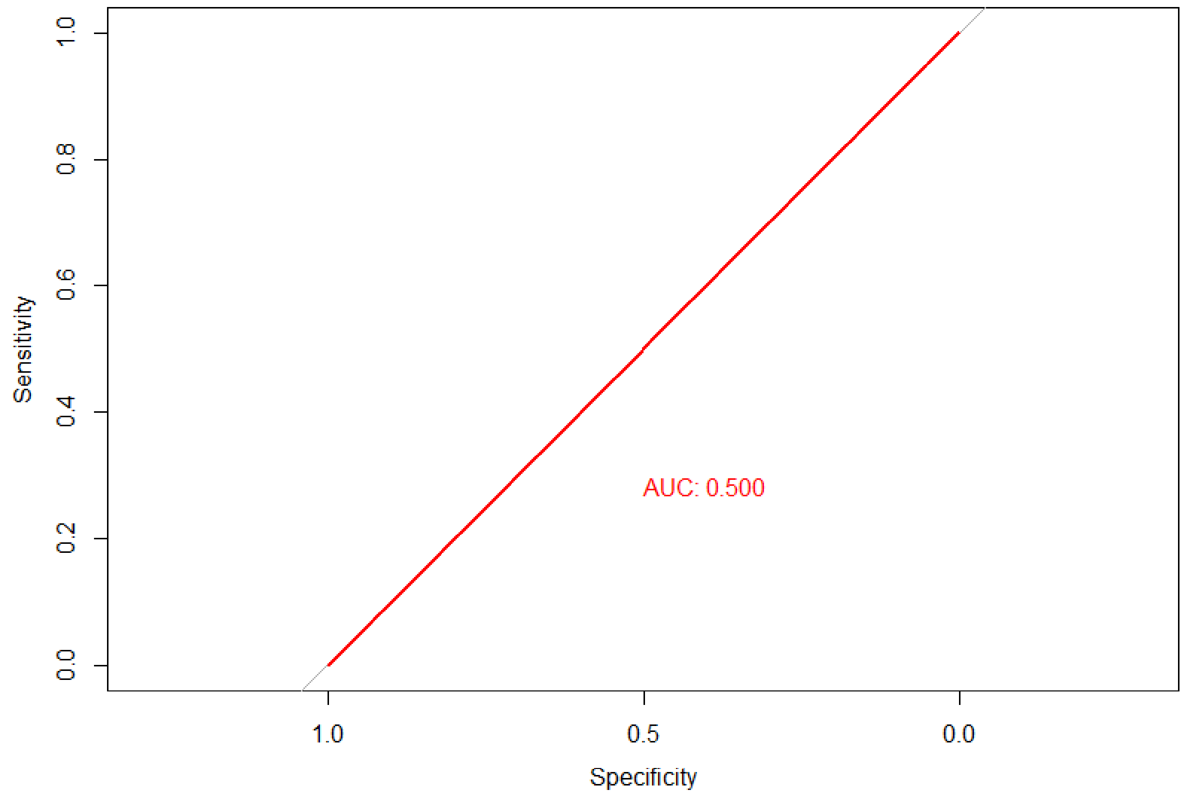
I then for the ROC curve and AUC value:



Below is the results excluding the J variable:

```
#SVM (No J)
str(train)
svmtrain2 = train
svmtest2 = test
svmtrain2$B = NULL
svmtrain2$E = NULL
svmtrain2$G = NULL
svmtrain2$H = NULL
svmtrain2$I = NULL
svmtrain2$M = NULL
svmtrain2$N = NULL
svmtrain2$O = NULL
svmtrain2$P = NULL
svmtrain2$Q = NULL
svmtrain2$R = NULL
svmtest2$B = NULL
svmtest2$E = NULL
svmtest2$G = NULL
svmtest2$H = NULL
svmtest2$I = NULL
svmtest2$M = NULL
svmtest2$N = NULL
svmtest2$O = NULL
svmtest2$P = NULL
svmtest2$Q = NULL
svmtest2$R = NULL
svmtrain2$J = NULL
svmtest2$J = NULL
svmfit2 = svm(Dependent ~ ., data = svmtrain2, probability = TRUE, type = "C-classification", kernal = "radial", g
svmpred2 <- predict(svmfit2, svmtest2, decision.values = TRUE, probability = TRUE, type ='class')
svmfit2
svmpred2
confusionMatrix(svmpred2, svmtest2$Dependent)

roc_svm_test2 <- roc(response = svmtest2$Dependent, predictor =as.numeric(svmpred2))
plot(roc_svm_test2)
plot(roc_svm_test2,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)


```

```
> confusionMatrix(svmpred2, svmtest2$Dependent)
Confusion Matrix and Statistics

             Reference
Prediction    FELONY MISDEMEANOR VIOLATION
  FELONY          16          21         1
  MISDEMEANOR  12400       20345      9017
  VIOLATION        0           0         0

Overall Statistics

               Accuracy : 0.4871
                 95% CI : (0.4823, 0.4919)
    No Information Rate : 0.4872
    P-Value [Acc > NIR] : 0.5214

                  Kappa : 1e-04

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: FELONY Class: MISDEMEANOR Class: VIOLATION
Sensitivity              0.0012887          0.9989689           0.0000
Specificity              0.9992513          0.0007931           1.0000
Pos Pred Value           0.4210526          0.4871654              NaN
Neg Pred Value           0.7030794          0.4473684           0.7843
Prevalence               0.2970335          0.4872249           0.2157
Detection Rate           0.0003828          0.4867225           0.0000
Detection Prevalence     0.0009091          0.9990909           0.0000
Balanced Accuracy        0.5002700          0.4998810           0.5000
```


1.0
0.8
0.6
0.4
0.2
0.0
Sensitivity
AUC: 0.500
1.0
0.5
0.0
Specificity

5 Classification Table

Here is a list of the first 248 observations and the algorithms predictions:

```
#Classification Table
ct = data.frame(dttest$Dependent,nbDefault_pred2,knnModel2,dtpred2,svmpred2)
view(ct)
colnames(ct) <- c("Given", "NaiveBayes","Knn","DecisionTree","SVM")
view(ct)
write.csv(ct,"C:\\Users\\poona\\Desktop\\ct.csv")
```

	Given	NaiveBayes	Knn	DecisionTree	SVM
1	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
2	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
3	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
4	FELONY	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
5	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
6	FELONY	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
7	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
8	FELONY	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
9	FELONY	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
10	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
11	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
12	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
13	FELONY	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
14	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
15	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
16	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
17	FELONY	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
18	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
19	VIOLATION	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
20	FELONY	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
21	FELONY	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR
22	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
23	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
24	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
25	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
26	VIOLATION	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
27	VIOLATION	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
28	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
29	FELONY	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
30	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
31	MISDEMEANOR	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR
32	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
33	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR

34	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
35	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
36	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
37	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
38	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
39	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
40	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
41	FELONY	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
42	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
43	MISDEMEANOR	FELONY	FELONY	MISDEMEANOR	MISDEMEANOR
44	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
45	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
46	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	FELONY
47	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
48	FELONY	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
49	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
50	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
51	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
52	FELONY	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR
53	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
54	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
55	VIOLATION	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
56	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
57	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
58	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
59	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
60	MISDEMEANOR	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
61	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
62	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
63	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
64	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
65	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
66	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
67	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
68	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
69	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
70	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
71	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
72	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
73	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
74	FELONY	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
75	FELONY	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
76	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR

77	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
78	VIOLATION	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
79	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
80	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
81	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
82	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
83	MISDEMEANOR	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
84	FELONY	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR
85	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
86	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
87	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
88	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
89	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
90	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
91	MISDEMEANOR	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR
92	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
93	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
94	VIOLATION	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
95	FELONY	FELONY	FELONY	MISDEMEANOR	MISDEMEANOR
96	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
97	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
98	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
99	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
100	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
101	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
102	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
103	FELONY	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
104	FELONY	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
105	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
106	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
107	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
108	VIOLATION	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
109	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
110	MISDEMEANOR	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR
111	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
112	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
113	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
114	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
115	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
116	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
117	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
118	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
119	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR

120	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
121	VIOLATION	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
122	VIOLATION	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
123	MISDEMEANOR	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
124	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
125	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
126	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
127	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
128	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
129	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
130	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
131	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
132	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
133	FELONY	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
134	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
135	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
136	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
137	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
138	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
139	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
140	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
141	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
142	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
143	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
144	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
145	FELONY	FELONY	FELONY	MISDEMEANOR	MISDEMEANOR
146	VIOLATION	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
147	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
148	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
149	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
150	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
151	MISDEMEANOR	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR
152	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
153	FELONY	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
154	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
155	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
156	VIOLATION	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
157	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
158	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
159	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
160	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
161	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
162	MISDEMEANOR	FELONY	FELONY	MISDEMEANOR	MISDEMEANOR

163	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
164	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
165	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
166	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
167	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
168	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
169	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
170	FELONY	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
171	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
172	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
173	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
174	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
175	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
176	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
177	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
178	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
179	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
180	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
181	MISDEMEANOR	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR
182	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
183	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
184	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
185	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
186	VIOLATION	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
187	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
188	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
189	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
190	FELONY	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
191	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
192	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
193	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
194	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
195	FELONY	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
196	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
197	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
198	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
199	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
200	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
201	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
202	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
203	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
204	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
205	MISDEMEANOR	FELONY	FELONY	MISDEMEANOR	MISDEMEANOR

206	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
207	FELONY	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
208	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
209	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
210	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
211	VIOLATION	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
212	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
213	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
214	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
215	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
216	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
217	MISDEMEANOR	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
218	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
219	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
220	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
221	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
222	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
223	FELONY	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
224	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
225	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
226	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
227	MISDEMEANOR	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
228	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
229	FELONY	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
230	MISDEMEANOR	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
231	FELONY	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
232	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
233	FELONY	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
234	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
235	FELONY	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
236	MISDEMEANOR	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR
237	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
238	VIOLATION	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
239	VIOLATION	VIOLATION	FELONY	MISDEMEANOR	MISDEMEANOR
240	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
241	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
242	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
243	FELONY	FELONY	VIOLATION	MISDEMEANOR	MISDEMEANOR
244	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
245	MISDEMEANOR	VIOLATION	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR
246	MISDEMEANOR	MISDEMEANOR	FELONY	MISDEMEANOR	MISDEMEANOR
247	MISDEMEANOR	VIOLATION	VIOLATION	MISDEMEANOR	MISDEMEANOR
248	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR	MISDEMEANOR

6 Results/Recommendation

If you noticed in the decision trees, in the confusion matrix for all of the variables including J, there was a 96% accuracy meaning that column is correlated to the results very highly that it was the only column it used in that model. In the other models involving variable j, the accuracy is about 60-69%, I believe this is because column J is a categorical variable represented numerically on the original dataset. So, this has led me to disregarding all data using that column as that seems to be redundant for the results. With that said, the confusion matrix for all the other models without J have an accuracy of less than 50%.

The sensitivity of each model is the proportion of observations of a class correctly identified while specificity is the proportion of observations outside of the class correctly identified. Precision is the number of correct observations of that class over the number of predicted observations of that class, which says how likely it is that an observation with that predicted class is correct.

The accuracy ranges from 46.17(Knn) to 48.72(Decision Tree) %. I'm going to disqualify the decision tree model however, as it just claimed everything was a "Misdemeanor". Same with the SVM model, I believe this is due to most cases being "Misdemeanor", in a rerun of the algorithms I would go back and change the proportion of each category I used. The naive Bayes model had better predictions for the "Violation" class by a sensitivity value of .4 but the Knn model has better sensitivity values in the other two classes. The specificity of shows the same thing just mentioned, that Knn was better for classes "Felony" and Misdemeanor" but not for "Violation". The precision however shows that the Naïve Bayes algorithm is generally better at predicting the correct observation for each class. I would go with the Naïve bayes algorithm for this reason.

Naïve Bayes is high bias low variance while Decision trees, Knn, and SVM are low bias and high variance algorithms. Bias is the difference between the average prediction and the correct value, this is due to oversimplifying the model and not using the training data enough. Variance has a high importance on training data and measures the variability of a prediction with fluctuating training data.

7 Code

```
#Load the data

library(readr)

library(plyr)

library(dplyr)

library(tidyverse)

library(usethis)

library(devtools)

#install_github("vqv/ggbiplot", force=TRUE)

library(grid)

library(ggbiplot)

library(ggplot2)

library(lattice)

library(caret)

#From IBM Terminal

#df <-
read_csv("/home/2021/nyu/fall/ap5254/hw01/NYPD_Complaint_Data_Current__Year_To_Date_.csv")

#If you have the file

df <- read_csv("C:/Users/poona/Downloads/NYPD_Complaint_Data_Current__Year_To_Date_.csv")

#If you have the link

#df = read.csv(url("https://data.cityofnewyork.us/api/views/5uac-
w243/rows.csv?accessType=DOWNLOAD"))

#Check dimensions

dim(df)


#Change names to numbers to help reduce bias

names(df) = c(1:36)

names(df)
```

```
#Label the dependent variable
names(df)[14] = 'Dependent'
names(df)

head(df)

#Imbalance check
sum(df$Dependent=='FELONY')/nrow(df)
sum(df$Dependent=='MISDEMEANOR')/nrow(df)
sum(df$Dependent=='VIOLATION')/nrow(df)

ggplot(data = df) +
  geom_bar(mapping = aes(Dependent))

#Get rid of Identifier
df = df[-c(1)]

#Data Types of each column
str(df)

#Removing columns that have missing values summing at least half of the total amount of observations
colSums(is.na(df))
nrow(df)/2
df = df[-c(5,6,8,9,16,22,26)]

#Removing columns that are a description of another column
df = df[-c(11,14)]
str(df)
```

```
#Principle Component Analysis
pca<- prcomp(df[,c(1,21,22,23,24)], center = TRUE,scale. = TRUE)
str(pca)
#ggbiplot(pca,labels = df$`13`, ellipse=TRUE, groups=df$Dependent)
#91.2

pca<- prcomp(df[,c(1,21,22)], center = TRUE,scale. = TRUE)
str(pca)
#ggbiplot(pca,labels = df$`13`, ellipse=TRUE, groups=df$Dependent)
#89.5

pca<- prcomp(df[,c(1,23,24)], center = TRUE,scale. = TRUE)
str(pca)
#ggbiplot(pca,labels = df$`13`, ellipse=TRUE, groups=df$Dependent)
#89.5

pca<- prcomp(df[,c(1,8)], center = TRUE,scale. = TRUE)
str(pca)
#ggbiplot(pca,labels = df$`13`, ellipse=TRUE, groups=df$Dependent)
#Removing column 8 as that seems to be a 100% correlated to the dependent variable
df = df[-c(8)]
df = df[-c(24,25)]

#randomForest and Variable Importance
library(randomForest)
df$Dependent = as.factor(df$Dependent)
table(df$Dependent)
colnames(df) = c('A','B','C','D','E','F','G','Dependent','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V')
```

```
df$B = as.factor(df$B)
df$C = as.factor(df$C)
df$E = as.factor(df$E)
df$G = as.factor(df$G)
df$H = as.factor(df$H)
df$I = as.factor(df$I)
df$K = as.factor(df$K)
df$L = as.factor(df$L)
df$M = as.factor(df$M)
df$N = as.factor(df$N)
df$O = as.factor(df$O)
df$P = as.factor(df$P)
df$Q = as.factor(df$Q)
df$R = as.factor(df$R)

df$C = NULL
df$D = NULL
df$K = NULL
df$L = NULL

set.seed(222)
index = sample(2, nrow(df), replace = TRUE, prob = c(0.7,0.3))
train = df[index==1,]
test = df[index==2,]
str(train)
summary(train)

rf = randomForest(Dependent~., train, na.action=na.omit)
rf
```

```
VI =varImp(rf)
varImpPlot(rf,main="Variable Importance")

str(df)
#VIF
library(car)
m = lm(A~F+J+S+T+U+V, data=df)
vif(m)

#Individual Classifiers
library(pROC)



test= na.omit(test)
#NaiveBayes


library(e1071)
set.seed(222)
nbDefault = naiveBayes(Dependent~., data=train, prob = TRUE)
nbDefault
nbDefault_pred = predict(nbDefault, test, type="class", prob = TRUE)
nbDefault_pred
confusionMatrix(nbDefault_pred, test$Dependent)
plot(nbDefault_pred)
roc_nb_test <- roc(response = test$Dependent, predictor =as.numeric(nbDefault_pred))
plot(roc_nb_test)
plot(roc_nb_test,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)
```

```
#NaiveBayes (No J)
library(e1071)
set.seed(222)
nbtrain = train
nbtest = test
nbtrain$J = NULL
nbtest$J = NULL
nbDefault2 = naiveBayes(Dependent~., data=nbtrain, prob = TRUE)
nbDefault2
nbDefault_pred2 = predict(nbDefault2, nbtest, type="class", prob = TRUE)
nbDefault_pred2
confusionMatrix(nbDefault_pred2, nbtest$Dependent)
plot(nbDefault_pred2)
roc_nb_test2 <- roc(response = nbtest$Dependent, predictor =as.numeric(nbDefault_pred))
plot(roc_nb_test2)
plot(roc_nb_test2,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#KNN
library(class)
knndf = df
str(knndf)
knndf$B = as.numeric(knndf$B)
knndf$E = as.numeric(knndf$E)
knndf$G = as.numeric(knndf$G)
knndf$H = as.numeric(knndf$H)
knndf$I = as.numeric(knndf$I)
knndf$M = as.numeric(knndf$M)
knndf$N = as.numeric(knndf$N)
knndf$O = as.numeric(knndf$O)
```

```
knndf$P = as.numeric(knndf$P)
knndf$Q = as.numeric(knndf$Q)
knndf$R = as.numeric(knndf$R)
knntrain = knndf[index==1,]
knntest = knndf[index==2,]
train_knn = na.omit(knntrain)
test_knn = na.omit(knntest)
train_knn_dep = train_knn
test_knn_dep = test_knn
train_knn$Dependent=NULL
test_knn$Dependent=NULL

knnModel = knn(train = train_knn, test = test_knn, cl = train_knn_dep$Dependent, prob = TRUE)
knnModel
confusionMatrix(knnModel, test_knn_dep$Dependent)
knnroc=roc(test_knn_dep$Dependent, attributes(knnModel)$prob)
plot(knnroc)
plot(knnroc,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#KNN (No J)
library(class)
knndf = df
str(knndf)
knndf$B = as.numeric(knndf$B)
knndf$E = as.numeric(knndf$E)
knndf$G = as.numeric(knndf$G)
knndf$H = as.numeric(knndf$H)
knndf$I = as.numeric(knndf$I)
knndf$M = as.numeric(knndf$M)
```

```
knndf$N = as.numeric(knndf$N)
knndf$O = as.numeric(knndf$O)
knndf$P = as.numeric(knndf$P)
knndf$Q = as.numeric(knndf$Q)
knndf$R = as.numeric(knndf$R)
knntrain2 = knndf[index==1,]
knntest2 = knndf[index==2,]
knntrain2$J = NULL
knntest2$J = NULL
train_knn2 = na.omit(knntrain2)
test_knn2 = na.omit(knntest2)
train_knn_dep2 = train_knn2
test_knn_dep2 = test_knn2
train_knn2$Dependent=NULL
test_knn2$Dependent=NULL

knnModel2 = knn(train = train_knn2, test = test_knn2, cl = train_knn_dep2$Dependent, prob = TRUE)
knnModel2
confusionMatrix(knnModel2, test_knn_dep2$Dependent)
knnroc2=roc(test_knn_dep2$Dependent, attributes(knnModel2)$prob)
plot(knnroc2)
plot(knnroc2,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#Decision Tree
library(rpart)
library(rpart.plot)
dt=rpart(Dependent~., data=train, method='class')
rpart.plot(dt)
dt
```

```
printcp(dt)
dtpred = predict(dt,test,type = 'class')
dtpred
confusionMatrix(dtpred, test$Dependent)
dtroc=roc(test$Dependent, as.numeric(dtpred))
plot(dtroc)
plot(dtroc,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#Decision Tree (No J)
library(rpart)
library(rpart.plot)
dttrain = train
dttrain$J= NULL
dttest = test
dttest$J= NULL
dt2=rpart(Dependent~., data=dttrain, method='class')
rpart.plot(dt2,box.palette = "blue")
dt2
printcp(dt2)
dtpred2 = predict(dt2,dttest,type = 'class')
dtpred2
confusionMatrix(dtpred2, test$Dependent)
dtroc2=roc(test$Dependent, as.numeric(dtpred2))
plot(dtroc2)
plot(dtroc2,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#SVM
str(train)
svmtrain = train
```

```
svmtest = test
svmtrain$B = NULL
svmtrain$E = NULL
svmtrain$G = NULL
svmtrain$H = NULL
svmtrain$I = NULL
svmtrain$M = NULL
svmtrain$N = NULL
svmtrain$O = NULL
svmtrain$P = NULL
svmtrain$Q = NULL
svmtrain$R = NULL
svmtest$B = NULL
svmtest$E = NULL
svmtest$G = NULL
svmtest$H = NULL
svmtest$I = NULL
svmtest$M = NULL
svmtest$N = NULL
svmtest$O = NULL
svmtest$P = NULL
svmtest$Q = NULL
svmtest$R = NULL

svmfit = svm(Dependent ~ ., data = svmtrain, probability = TRUE, type = "C-classification", kernal = "radial", gamma = 0.1, cost = 1)
svmpred <- predict(svmfit, svmtest, decision.values = TRUE, probability = TRUE, type ='class')
svmfit
svmpred
```

```
confusionMatrix(svmpred, svmtest$Dependent)

roc_svm_test <- roc(response = svmtest$Dependent, predictor =as.numeric(svmpred))
plot(roc_svm_test)
plot(roc_svm_test,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)

#SVM (No J)
str(train)
svmtrain2 = train
svmtest2 = test
svmtrain2$B = NULL
svmtrain2$E = NULL
svmtrain2$G = NULL
svmtrain2$H = NULL
svmtrain2$I = NULL
svmtrain2$M = NULL
svmtrain2$N = NULL
svmtrain2$O = NULL
svmtrain2$P = NULL
svmtrain2$Q = NULL
svmtrain2$R = NULL
svmtest2$B = NULL
svmtest2$E = NULL
svmtest2$G = NULL
svmtest2$H = NULL
svmtest2$I = NULL
svmtest2$M = NULL
svmtest2$N = NULL
svmtest2$O = NULL
```

```
svmtest2$P = NULL

svmtest2$Q = NULL

svmtest2$R = NULL

svmtrain2$J = NULL

svmtest2$J = NULL

svmfit2 = svm(Dependent ~ ., data = svmtrain2, probability = TRUE, type = "C-classification", kernal =
"radial", gamma = 0.1, cost = 1)

svmpred2 <- predict(svmfit2, svmtest2, decision.values = TRUE, probability = TRUE, type ='class')

svmfit2

svmpred2

confusionMatrix(svmpred2, svmtest2$Dependent)


roc_svm_test2 <- roc(response = svmtest2$Dependent, predictor =as.numeric(svmpred2))

plot(roc_svm_test2)

plot(roc_svm_test2,add = TRUE,col = "red", print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)


#Classification Table

ct = data.frame(dttest$Dependent,nbDefault_pred2,knnModel2,dtpred2,svmpred2)

view(ct)

colnames(ct) <- c("Given", "NaiveBayes","Knn","DecisionTree","SVM")

view(ct)

write.csv(ct,"C:\\Users\\poona\\Desktop\\ct.csv")
```