CAP5615 Justin Johnson
Summer 2018 Z23136514

Question 1

Use your own language to briefly explain the following concepts

Product Rule in Probability Theory

The product rule states that given two events, A and B, the probability of the two events occurring together is equal to the conditional probability of event A given event B, multiplied by the probability of event B. This allows us to calculate the probability of multiple events occurring in conjunction.

$$P(AB) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

If the two events are independent of each other, then this can be simplified to:

$$P(AB) = P(A)P(B)$$

Sum Rule in Probability Theory

The sum rule states that given two events, A and B, the probability of A or B occurring is equal to the probability of A occurring plus the probability of B occurring, minus the probability of A and B occurring. This allows us to calculate the probability of one event out of two or more occurring.

$$P(A + B) = P(A) + P(B) - P(AB)$$

If the two events are mutually exclusive, then this simplifies to:

$$P(A + B) = P(A) + P(B)$$

Bayes Rule (priori probability, likelihood, posteriori probability)

The conditional probability of event A given event B is equal to the conditional probability of event B given event A, multiplied by the probability of event A divided by the probability of event B. This is useful because we will often need conditional probability of event A given B, but we will only know the conditional probability of event B given A from experience (training data).

$$Bayes\ Rule: P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

P(A | B) = posteriori probability
P(B | A) = likelihood of B given A (observed in training data)
P(A) = prior probability of A (observed in training data)
P(B) = prior probability of training data

Maximum a Posteriori Estimation (MAP)

Bayesian learners seek to find the maximum posteriori probability, that is, the most probable hypothesis/model h given the training data D. Given the entire hypothesis space, which one is most probable given the training data. Searching the entire hypothesis space is very expensive.

$$h_{map} = maxP(h \mid D) = \ max_{h \epsilon H} P(D \mid h)P(h)$$

Conditional Independence
The notion that features are independent of each other, given the presence of a condition. In a different context, the features may be correlated, but under a specific condition, they are independent of each other. The example provided in lecture included features consisting of cold symptoms. If they are attributes of patients with the flu, then in this condition of patients having the flu, they are independent of each other. In a different context however, say where the condition is that the person is a student, then these features would be correlated. Conditional independence allows for simplification of Bayesian learners, to the Naïve Bayes learner.

Naïve Bayes Classification
A popular probabilistic supervised learning method that assumes attributes are independent of each other. It works well with high dimensional data, performs best with large training sets, and is very efficient when it comes to training (O(n) time complexity). Its qualities make it a good choice for text classification, where instances are text documents with thousands of features. It also makes for a great benchmark, as it is both easy to use and fast to compute.

$$v_{NB} = argmax_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j)$$

Question 2

Lab returns true positive of 95%, true negative of 95%, and 0.001 (0.1%) of entire population has disease. Use Bayes Rule to derive the probability of the patient having the disease given that his/her lab test is positive.

P(disease) = 0.001	P(no disease) = 0.999
P(positive test | disease) = .95	P(positive test | no disease) = .05
P(negative test | no disease) = .95	P(negative test | disease) = .05

P(disease | positive test) = P(positive test | disease) P(disease) / P(positive test)
P(disease | positive test) = (.95)(.001) / P(positive test)

P(positive test) = P(positive test | disease)P(disease) + P(positive test | no disease) P(no disease)
P(positive test) = (.95)(.001) + (.05)(.999).= 0.0509

P(disease | positive test) = (.95)(.001) / .0509 = 0.019

By Bayes Rule, there is 1.9% chance patient has disease given positive test result.

Question 3

Black box = 3A and 4O	P(black) = .5	
Red box = 5A and 1O	P(red) = .3	
Green box = 2A and 5O	P(green) = .2	
P(apple | black) = 3/7	P(apple | red) = 5/6	P(apple | green) = 2/7

What is overall chance that an Apple will be selected?
By the theorem of total probability:
P(apple) = P(apple | black)P(black) + P(apple | red)P(red) + P(apple | green)P(green)
P(apple) = (3/7)(.5) + (5/6)(.3) + (2/7)(.2) = **0.521 = P(apple)**

If the fruit selected is an Apple, what is the probability that the Apple was selected from the Green box?
By Bayes Rule:
P(green | apple) = P(apple | green) P(green) / P(apple)
P(green | apple) = (2/7)(.2) / (0.521) = **0.110 = P(green | apple)**

Question 4

Manually construct Naïve Bayes classifier from Table 1. The classifier is constructed by calculating the priori probabilities (P(yes) and P(no)) and all of the conditional probabilities. These calculations are outlined in following table:

Conditional Probabilities	
Outlook	
P(sunny | yes) = 2/9	P(sunny | no) = 3/6
P(rain | yes) = 4/9	P(rain | no) = 2/6
P(overcast | yes) = 3/9	P(overcast | no) = 1/6
Temperature	
P(hot | yes) = 2/9	P(hot | no) = 2/6
P(mild | yes) = 4/9	P(mild | no) = 3/6
P(cool | yes) = 3/9	P(cool | no) = 1/6
Humidity	
P(high | yes) = 4/9	P(high | no) = 4/6
P(normal | yes) = 5/9	P(normal | no) = 2/6
Wind	
P(strong | yes) = 4/9	P(strong | no) = 3/6
P(weak | yes) = 5/9	P(weak | no) = 3/6

Priori Probabilities
P(yes) = 9/15
P(no) =6/15

We can now classify new instances by comparing P(yes | <attributes>) vs P(no | <attributes>)

P(yes | <overcast, hot, normal, weak>) = (3/9)(2/9)(5/9)(5/9)(9/15) = 0.014
P(no | <overcast, hot, normal, weak>) = (1/6)(2/6)(2/6)(3/6)(6/15) = 0.004

The probability of Yes (play) given attributes <overcast, hot, normal, weak> is greater than the probability of No (don’t play). Therefore, based on experience, the person should play given these conditions.

Question 5

Manually construct Naïve Bayes classifier using m-estimate to calculate the conditional probabilities. Let m = 1 and p = 1 divided by total number of attribute values for each attribute.

$$P(A_i = a_{ij} \mid Y = y_k) = \frac{n_{ijk} + mp}{n_k + mp} = \frac{n_{ijk} + p}{n_k + p}$$

Attributes Outlook and Temperature both have 3 possible values, and so p = 1/3 for these conditional probability calculations. For Humidity and Wind, p = ½ because they only have two possible values per attribute.

The p value for each attribute, along with all conditional probabilities and priori probabilities are recorded in table below:

Conditional Probabilities	
Outlook (m = 1, p = 1/3)	
P(sunny | yes) = 1/4	P(sunny | no) = 10/19
P(rain | yes) = 13/28	P(rain | no) = 7/19
P(overcast | yes) = 5/14	P(overcast | no) = 4/19
Temperature (m = 1, p = 1/3)	
P(hot | yes) = 1/4	P(hot | no) = 7/19
P(mild | yes) = 13/28	P(mild | no) = 10/19
P(cool | yes) = 5/14	P(cool | no) = 4/19
Humidity (m = 1, p = 1/2)	
P(high | yes) = 9/19	P(high | no) = 9/13
P(normal | yes) = 11/19	P(normal | no) = 5/13
Wind (m = 1, p = 1/2)	
P(strong | yes) = 9/19	P(strong | no) = 7/13
P(weak | yes) = 11/19	P(weak | no) = 7/13

Priori Probabilities
P(yes) = 9/15
P(no) =6/15

Let new instance x = <overcast, hot, normal, weak>

P(yes | x) = (5/14)(1/4)(11/19)(11/19)(9/15) = 0.0180
P(no | x) = (4/19)(7/19)(5/13)(7/13)(6/15) = 0.0064

Given the conditions of instance x, the person should play tennis because the probability of Yes (play) is greater than the probability of No (don't play), based on past experience.

Question 6

Performing Naïve Bayes classification on mtcars categorical data set.

The following R script was used to complete all steps of Question 6:

```r
1  # define data path
2  dataDir = '../data'
3  fileName = 'mtcars.header.binary.categorical.txt'
4  dataPath = paste(dataDir, fileName, sep='/')
5
6  # load data
7  mtcars = read.table(dataPath, header=T, sep=',')
8
9  # train NB Classifier
10 require(e1071)
11 classifier = naiveBayes(factor(mpg)~., data = mtcars)
12
13 # view priori and conditional probabilities
14 # as generated by classifier
15 print(classifier)
16
17 # perform predictions on the training data
18 predictions = predict(classifier, newdata = mtcars)
19
20 # view confusion matrix
21 confMatrix = table(predictions, mtcars$mpg)
```

Once the data is loaded, all instances of data set are used to train Naïve Bayes classifier (line 11).

The classifier is then evaluated using the same training data (line 18) and a confusion matrix is generated (line 21) to view total number of correctly classified instances.

```
predictions  0  1
          0 16  1
          1  2 13
```

We can see that out of the 18 instances with MPG = 0, 16 were correctly classified and 2 were incorrectly labelled by the classifier as MPG = 1. We can also see that out of the 14 instances with MPG = 1, only 1 was incorrectly classified as MPG = 0, with the other 13 correctly classified.

$$Accuracy = \frac{total\ correctly\ classified\ instances}{total\ number\ of\ instances} = \frac{29}{32} = 0.906 = 90.6\%$$

The results of the classifier are printed to the console (line 15), to view the priori and conditional probabilities that were generated during the training process. The gear and cylinder conditional probabilities were cut from these results and are presented below:

```
   cyl
Y       eight      four       six
  0 0.6111111 0.1666667 0.2222222
  1 0.0000000 0.7857143 0.2142857
```

```
   gear
Y         five      four     three
  0 0.1666667 0.1111111 0.7222222
  1 0.1428571 0.7142857 0.1428571
```

These are the conditional probabilities that were calculated during the training process. These reflect the probability of an event A given the condition that event B has occurred.

Given an MPG value of label 0:
Probability of car having 8 cylinders = 61.1%
Probability of car having 4 cylinders = 16.7%
Probability of car having 6 cylinders = 22.2%
Probability of car having 5 gears = 16.7%
Probability of car having 4 gears = 11.1%
Probability of car having 3 gears = 72.2%

Given an MPG value of label 1:
Probability of car having 8 cylinders = 0.0%
Probability of car having 4 cylinders = 78.6%
Probability of car having 6 cylinders = 21.4%
Probability of car having 5 gears = 14.3%
Probability of car having 4 gears = 71.4%
Probability of car having 3 gears = 14.3%

Question 7

A Naïve Bayes classifier is trained using the housing binary data set. The below screenshot includes the R script that completes all tasks required of question 7:

Problem7.R

```r
# define data
dataDir = '../data'
fileName = 'housing.header.binary.txt'
filePath = paste(dataDir, fileName, sep = '/')

# load data
housing = read.table(filePath, header = T, sep = ',')

# partition data into 80/20 (train/test)
trainSize = floor(0.8 * nrow(housing))
set.seed(42)
trainIndices = sample(seq_len(nrow(housing)), size = trainSize)
trainData = housing[trainIndices, ]
testData = housing[-trainIndices, ]

# train classifier using the trainData set
require(e1071)
classifier = naiveBayes(factor(trainData$Medv)~., data = trainData)

# evaluate classifier using testData set
testPredictions = predict(classifier, newdata = testData)
confMatrix = table(testPredictions, testData$Medv)
print(confMatrix)

# create 1 synthetic instance and classify with NB classifier
synthInstance = data.frame('Crim'=0.03, 'Zn'=13, 'Indus'=3.5, 'Chas'=0.3, 'Nox'=0.58,
                           'Rm'=4.1, 'Age'=68, 'Dis'=4.98, 'Rad' =3, 'Tax'=225, 'Ptratio'=17,
                           'B'=396, 'Lstat'=7.56)
synthPrediction = predict(classifier, newdata = synthInstance, type='raw')
print(synthPrediction)

# create ROC curve using ROCR library
require(ROCR)
testPredictionProbOf1 = predict(classifier, testData, type = 'raw')
# creating matrix using columnbind method
# column 1 is probability that instance is classified as Medv = 1
# column 2 is the actual label of the instance
testPredictionProbOf1Labelled = cbind(testPredictionProbOf1[ , 2], testData$Medv)

pred = prediction(testPredictionProbOf1Labelled[ , 1], testPredictionProbOf1Labelled[ , 2])
perf = performance(pred, 'tpr', 'fpr')
plot(perf, col = 'red')
abline(0, 1, col = 'lightgray')

# calculate AUC
auc = performance(pred, 'auc')@y.values[[1]]
print(auc)
```

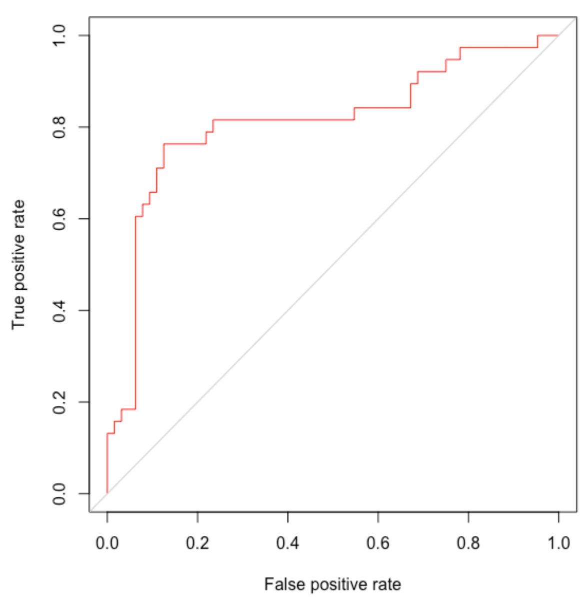
Report the confusion matrix and explain:

```
testPredictions  0  1
              0 38  7
              1 26 31
>
```

From the 38 instances with Medv = 1, 7 are incorrectly classified as Medv = 0 (FN), the remaining 31 instances are correctly classified as Medv = 1 (TP). From the 64 instances with a Medv = 0, 38 are correctly classified (TN) and 26 are incorrectly classified as Medv = 1 (FP).

True positive rate = TP / (TP + FN) = 31 / (31 + 7) = 0.816
True negative rate = TN / (TN + FP) = 38 / (38 + 26) = 0.594
False positive rate = FP / (FP + TN) = 26 / (38 + 26) = 0.406
False negative rate = FN / (FN + TP) = 7 / (7 + 31) = 0.184
Accuracy = (TP + TN) / (TP + TN + FP + FN) = 0.676 = 67.6%

The test data probabilities were used to construct ROC curve using ROCR R library. The screenshot listed below displays the ROC curve in red. The AUC was calculated as 0.810.



Create new instance <Crim=0.03, Zn=13, Indus=3.5, Chas=0.3, Nox=0.58, Rm=4.1, Age=68, Dis=4.98, Rad=3, Tax=225, Ptratio=17, B=396, Lstat=7.56> and predict the Medv of this new instance.

The synthetic instance was generated (line 26) and classified (line 29) as Medv = 1. Below are the posterior probabilities as calculated by the Naïve Bayes classifier:

```
> synthPrediction = predict(classifier, newdata = synthInstance, type='raw')
> print(synthPrediction)
               0         1
[1,] 0.001059318 0.9989407
```

The new instance was predicted to have Medv of 1 with 99.9% posterior probability.