Queen's MASTER OF MANAGEMENT ANALYTICS

MMA 869

Machine Learning and AI

Dr. Stephen W. Thomas

Individual Assignment 1

December 16, 2018

Shivaki

1. RUSPINI

Enrique H. Ruspini is a famous researcher who published a dataset (known simply as the rupini dataset) that has become a classic for students to experiment with various machine learning algorithms.

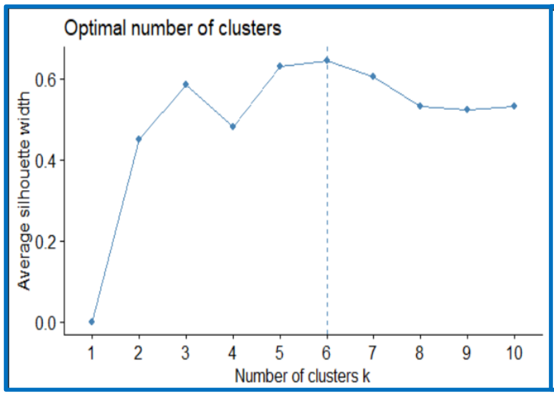
Join the club. Use the `ruspini` dataset provided with the `cluster` package in R. Perform a K-means analysis. Document your findings and justify your choice of k. Hint: use `data(ruspini)` to load the dataset into the R workspace.

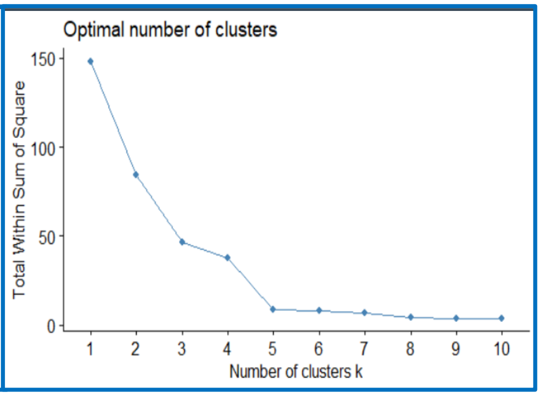
Answer:

Data Preparation: Data was loaded to the R studio and the structure and summary for dataset was observed. The data has 70 instances and 2 features. There are no missing values but there are some outliers. We don't eliminate the outliers and the algorithm clustering to handle them unsupervised. Since we do not the clusters to be biased by the scale of any particular feature, I started by scaling/standardizing the dataset.

Distance Measure: Since both variables are numeric, I chose to go ahead with the default distant measure i.e. Euclidian.

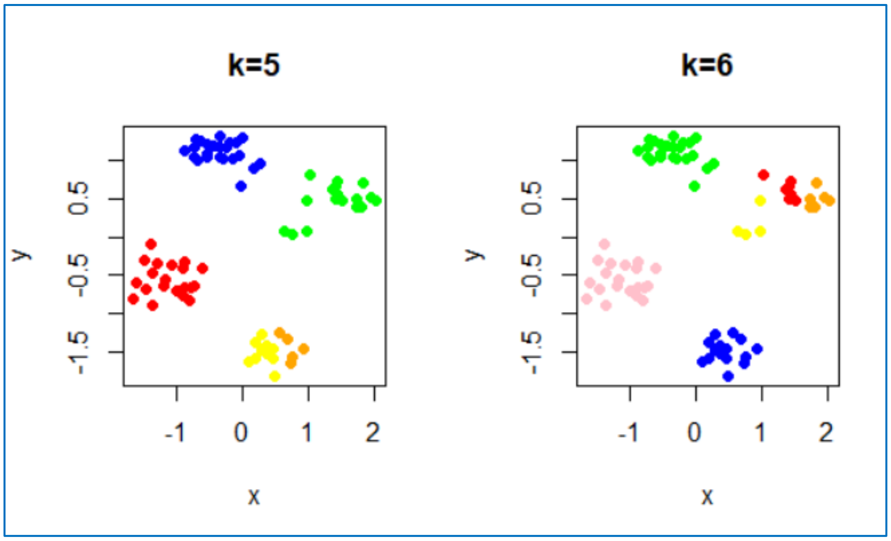
Optimum Number of Clusters: I used wss elbow point and average silhouette method to determine the optimum number of clusters.





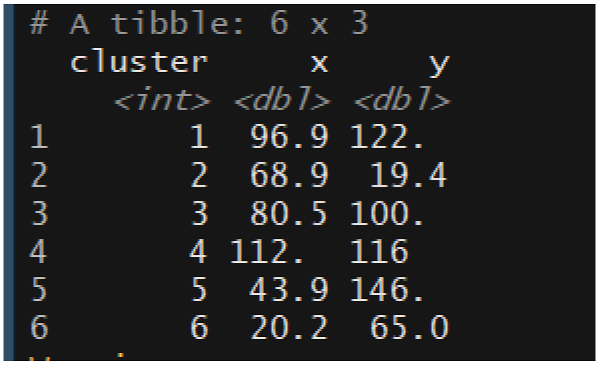
Both of these approaches suggest 5 or 6 as the optimum number of clusters. Hence, we would explore both and finally choose one.

Results: I performed the k means clustering using both 5 and 6 clusters. Both the clusters are plotted below:



Interpretation: For the six clusters, three clusters are very small. Hence for such a small dataset it would be more logical to go ahead with 5 clusters.

Analyzing the five clusters based on the means of the original variables



```
# A tibble: 6 x 3
  cluster     x     y
    <int> <dbl> <dbl>
1       1  96.9 122.
2       2  68.9  19.4
3       3  80.5 100.
4       4 112.  116
5       5  43.9 146.
6       6  20.2  65.0
```

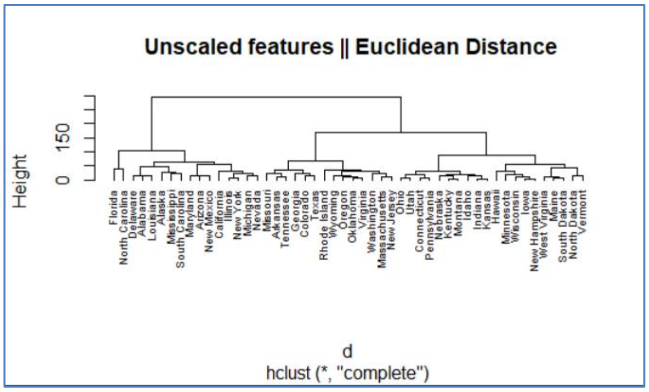
2. ARREST THAT MAN!

You work in Washington, DC. You help politicians do… whatever politicians do. Like most political analysts, you would like to make a strong case about crime in America. In particular, you would like to analyze arrest rates in each state, and determine which states are most similar to each other in that regard. You find the `USArrests` dataset in the `datasets` package. You will now perform clustering on the states to determine groups of similar states.

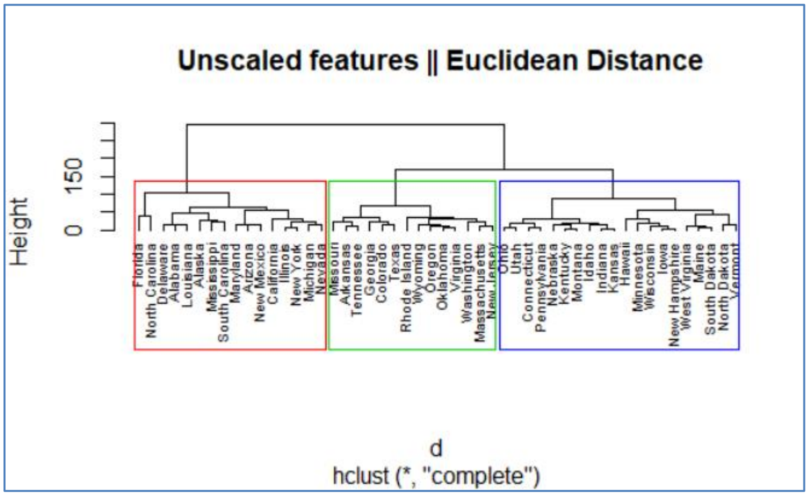
a) Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states. Hint: use the `hclust` package in R.
b) Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters? Provide a description/interpretation of each cluster.
c) Hierarchically cluster the states again, except this time, first scale all of the features to have standard deviation one. Hint: use the `scale()` function.

Answer:

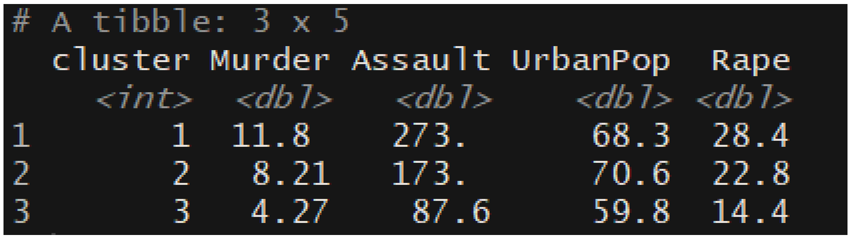
a) Agglomerative HC using the Euclidean distance as the dissimilarity matrix and complete linkage.



b) Cutting tree into 3 clusters:



Analyzing the cluster means for describing the clusters:



```
# A tibble: 3 x 5
  cluster Murder Assault UrbanPop  Rape
    <int>  <dbl>   <dbl>    <dbl> <dbl>
1       1  11.8    273.      68.3  28.4
2       2   8.21   173.      70.6  22.8
3       3   4.27    87.6     59.8  14.4
```

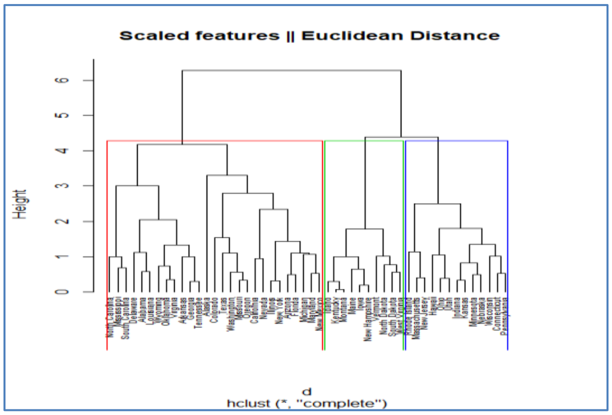
Based on these, I came up with the following explanations for the 3 clusters:

Cluster 1 – Most Dangerous (with very high crime rates across all categories)

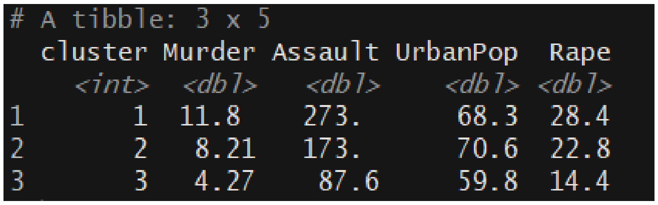
Cluster 2 – The Middle

Cluster 3 – Least Dangerous (with lowest crime rates across all categories)

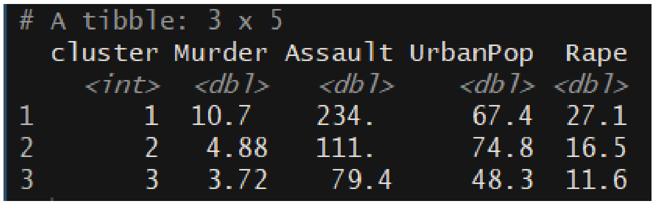
c) Cutting tree into 3 clusters:



<u>Comparing the clusters obtained from the two approaches: </u>The clusters through the two approaches are mostly similar but the clusters obtained with the scaled features are less impacted with the values of the “Assault” feature which has high magnitude compared to other features.



```
# A tibble: 3 x 5
  cluster Murder Assault UrbanPop  Rape
    <int>  <dbl>   <dbl>    <dbl> <dbl>
1       1  11.8    273.      68.3  28.4
2       2   8.21   173.      70.6  22.8
3       3   4.27    87.6     59.8  14.4
```



```
# A tibble: 3 x 5
  cluster Murder Assault UrbanPop  Rape
    <int>  <dbl>   <dbl>    <dbl> <dbl>
1       1  10.7    234.      67.4  27.1
2       2   4.88   111.      74.8  16.5
3       3   3.72    79.4     48.3  11.6
```

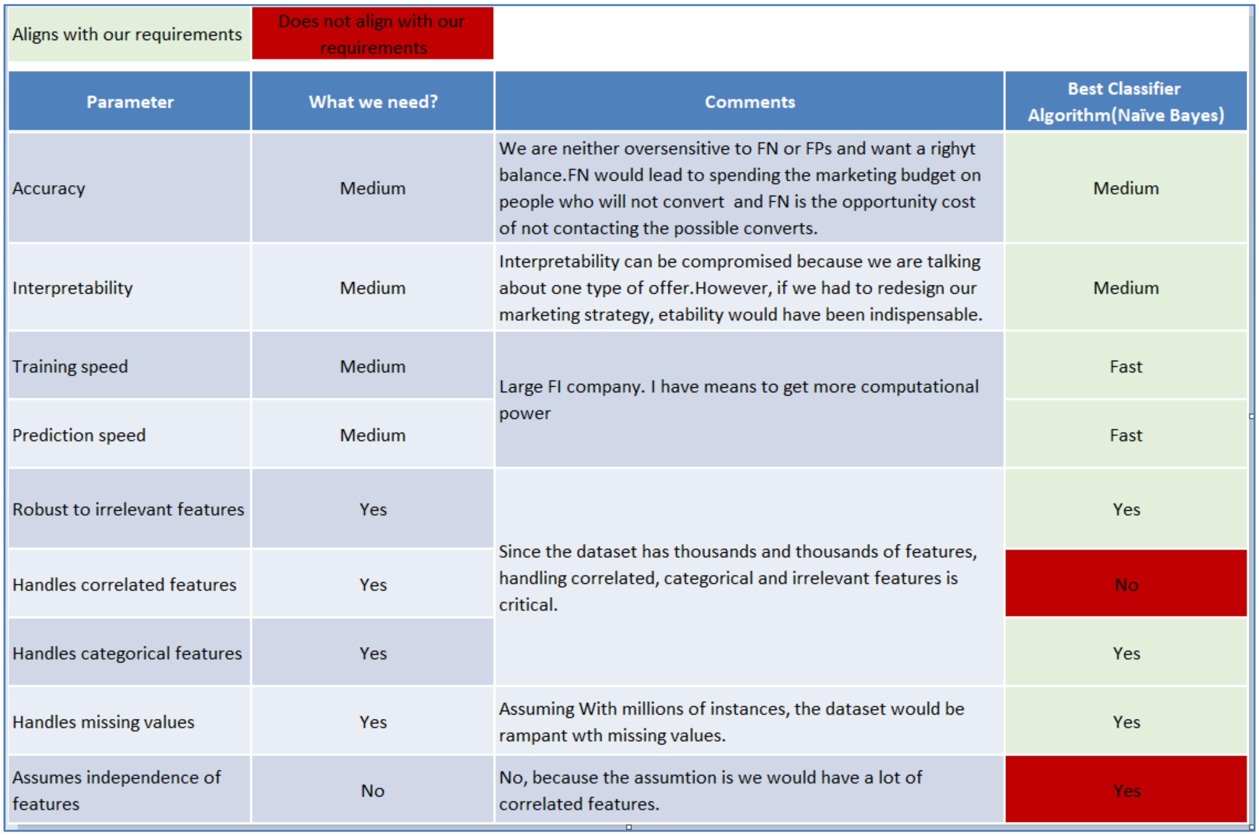
3. WHO'S THE BEST?

You work at a large financial services company. The marketing department presents your analytics team with a customer dataset, which has thousands and thousands of features, and millions of instances. Your goal is to classify whether customers will likely respond to a given offer, but first you must choose a classifier algorithm to use. Of the five classifier algorithms we've discussed in class (i.e., Decision Trees, Naïve Bayes, KNN, SVM, and Neural Networks), which algorithm would you choose? Why? State any additional assumptions you are making.

Answer:

To understand the best classifier algorithm, I compared the parameter that would are critical for us based on the dataset and then find the best match from the given models. In my opinion, Naïve Bayes would be good model. There is almost a tie between Decision Tree and Naïve Bayes, but Naïve Bayes is better at handling irrelevant features which is critical here since we have thousands of features.

However, Naïve Bayes is not good at handling correlated features and does assume independence of features. This is a significant limitation given the size especially width of our dataset. So, we will make sure that the correlations are handled carefully during the data preparation phase.



🟩 Aligns with our requirements	🟥 Does not align with our requirements

Parameter	What we need?	Comments	Best Classifier Algorithm(Naïve Bayes)
Accuracy	Medium	We are neither oversensitive to FN or FPs and want a righyt balance.FN would lead to spending the marketing budget on people who will not convert and FN is the opportunity cost of not contacting the possible converts.	Medium
Interpretability	Medium	Interpretability can be compromised because we are talking about one type of offer.However, if we had to redesign our marketing strategy, etability would have been indispensable.	Medium
Training speed	Medium	Large FI company. I have means to get more computational power	Fast
Prediction speed	Medium		Fast
Robust to irrelevant features	Yes	Since the dataset has thousands and thousands of features, handling correlated, categorical and irrelevant features is critical.	Yes
Handles correlated features	Yes		No
Handles categorical features	Yes		Yes
Handles missing values	Yes	Assuming With millions of instances, the dataset would be rampant wth missing values.	Yes
Assumes independence of features	No	No, because the assumtion is we would have a lot of correlated features.	Yes

4. Catching Recidivists Before They Strike

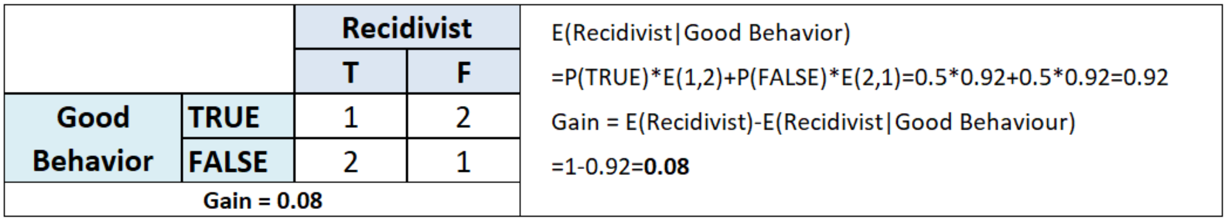
A ***recidivist*** is a criminal that was released from prison, but commits another crime. You are a warden at a maximum-security prison in Kingston, and you want to determine which prisoners will likely become recidivists. Luckily, you have a Queen's degree, so you are going to take a data-driven approach. You have collected some historical training data that include some basic metadata, and whether the prisoner ended up becoming a recidivist or not.

Given the training data below, use the ID3 algorithm and entropy-based information gain to construct a decision tree by hand to predict which prisoners will become recidivists. Show all the steps. Use the resulting decision tree to predict the class of the following prisoner: Good Behavior = false, Age < 30 = false, Drug dependent = true.

Id	Good Behavior	Age < 30	Drug Dependent	Recidivist
1	False	True	False	True
2	False	False	False	False
3	False	True	False	True
4	True	False	False	False
5	True	False	True	True
6	True	False	False	false

Answer:

1. **Entropy:**
   We need to calculate the entropy at the output variable level fist. The output consists of 6 instances with two labels: 3 True and 3 False values.
   Entropy (Recidivist) = - p(True).$\log_2$p(True) – p(False).$\log_2$p(False) = $-0.5\log_2 0.5 - 0.5\log_2 0.5 = 1$
2. **All three factors Gain calculation:**



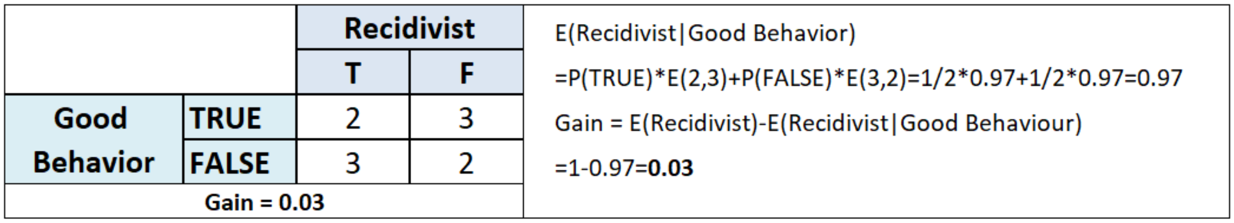
		Recidivist	
		T	F
Good Behavior	TRUE	1	2
	FALSE	2	1
Gain = 0.08			

E(Recidivist|Good Behavior)

=P(TRUE)*E(1,2)+P(FALSE)*E(2,1)=0.5*0.92+0.5*0.92=0.92

Gain = E(Recidivist)-E(Recidivist|Good Behaviour)

=1-0.92=**0.08**



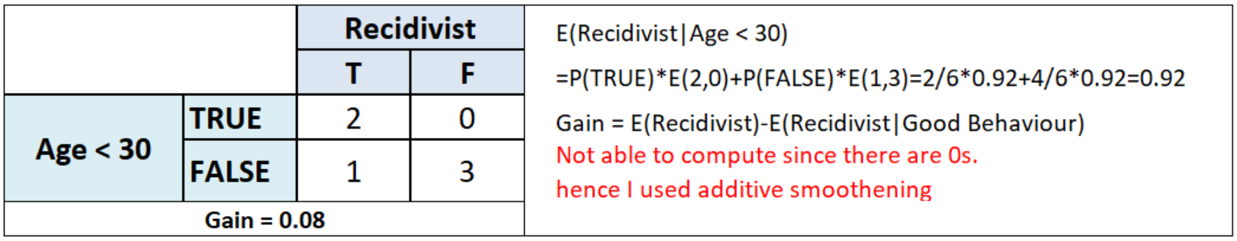
		Recidivist	
		T	F
Good Behavior	TRUE	2	3
	FALSE	3	2
Gain = 0.03			

E(Recidivist|Good Behavior)

=P(TRUE)*E(2,3)+P(FALSE)*E(3,2)=1/2*0.97+1/2*0.97=0.97

Gain = E(Recidivist)-E(Recidivist|Good Behaviour)

=1-0.97=**0.03**



		Recidivist	
		T	F
Age < 30	TRUE	2	0
	FALSE	1	3
Gain = 0.08			

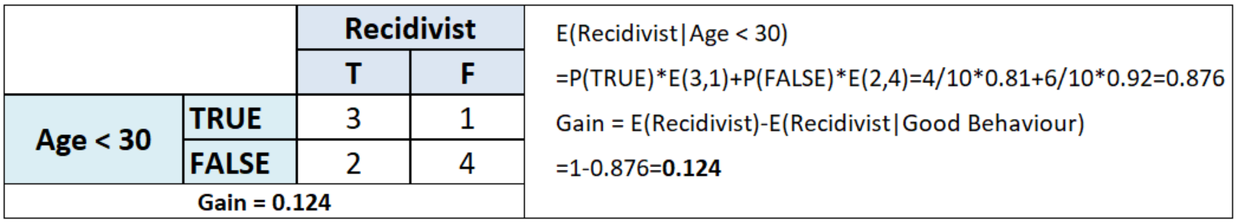
E(Recidivist|Age < 30)

=P(TRUE)*E(2,0)+P(FALSE)*E(1,3)=2/6*0.92+4/6*0.92=0.92

Gain = E(Recidivist)-E(Recidivist|Good Behaviour)

Not able to compute since there are 0s.

hence I used additive smoothening



		Recidivist	
		T	F
Age < 30	TRUE	3	1
	FALSE	2	4
Gain = 0.124			

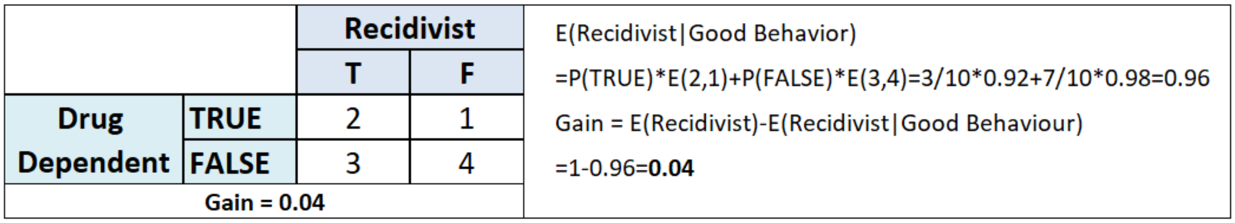
E(Recidivist|Age < 30)

=P(TRUE)*E(3,1)+P(FALSE)*E(2,4)=4/10*0.81+6/10*0.92=0.876

Gain = E(Recidivist)-E(Recidivist|Good Behaviour)

=1-0.876=**0.124**

We need to recompute the values with additive smoothening for Good Behavior again for apple to apple comparison. (Please find the recalculated table above).



		Recidivist	
		T	F
Drug Dependent	TRUE	2	1
	FALSE	3	4
Gain = 0.04			

E(Recidivist|Good Behavior)

=P(TRUE)*E(2,1)+P(FALSE)*E(3,4)=3/10*0.92+7/10*0.98=0.96

Gain = E(Recidivist)-E(Recidivist|Good Behaviour)

=1-0.96=**0.04**

As seen, Age < 30 factor produces the highest gain. Hence, this feature will appear in the root node of the decision tree.

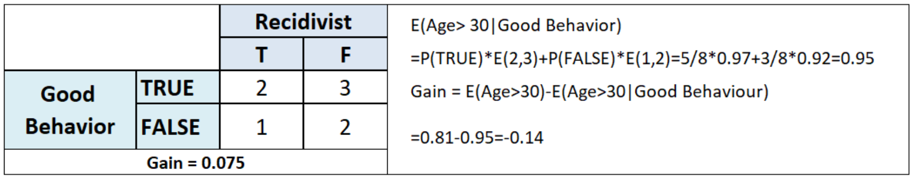
In the resultant tree, all prisoners with Age < 30 years are recidivists, hence we get a perfect stopping point. But for Age > 30 years, we still have mixed results and hence we do the iterations again.

3. **<u>Calculating Gains for Age > 30 years:</u>**
4. Doing the iteration again for Age > 30 years: There are 4 instances where the Age > 30 years, one if True and 3 are False.

Entropy (Recidivist |Age>30) = - p(True).log₂p(True) – p(False).log₂p(False) =

$$-0.25\log_2 0.25 - 0.75\log_2 0.75 = 0.81$$

Again, we calculate the gain and use additive smoothening since we have 0 in the final numbers.



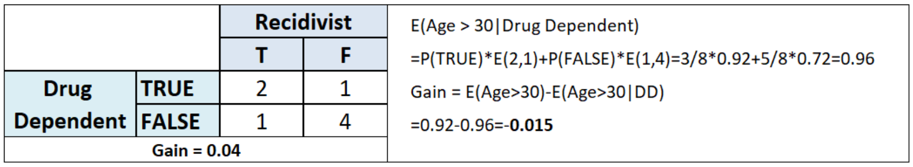
		Recidivist	
		T	F
Good Behavior	TRUE	2	3
	FALSE	1	2
Gain = 0.075			

E(Age> 30|Good Behavior)

=P(TRUE)*E(2,3)+P(FALSE)*E(1,2)=5/8*0.97+3/8*0.92=0.95

Gain = E(Age>30)-E(Age>30|Good Behaviour)

=0.81-0.95=-0.14



		Recidivist	
		T	F
Drug Dependent	TRUE	2	1
	FALSE	1	4
Gain = 0.04			

E(Age > 30|Drug Dependent)

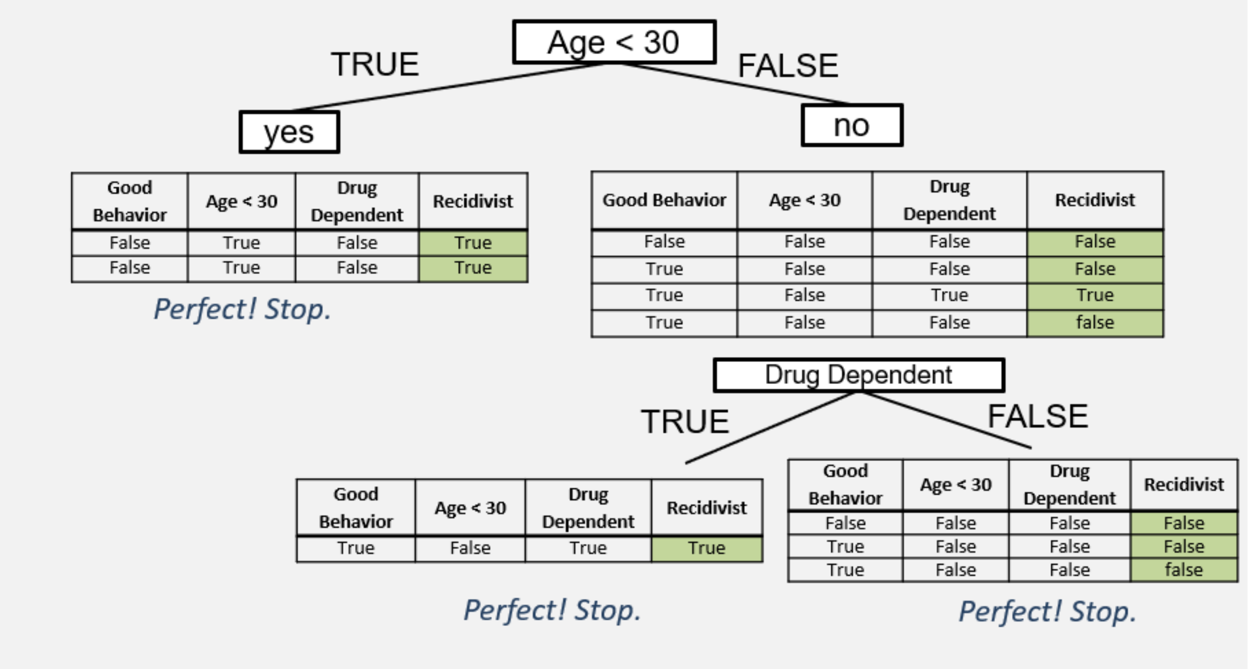
=P(TRUE)*E(2,1)+P(FALSE)*E(1,4)=3/8*0.92+5/8*0.72=0.96

Gain = E(Age>30)-E(Age>30|DD)

=0.92-0.96=-**0.015**

As seen, Drug Dependent factor produces the highest gain. Hence, this feature will appear in the next decision node.

After, this spilt we get perfect Trues and Falses and hence we will stop at this iteration. The final decision tree looks as follows:



Good Behavior	Age < 30	Drug Dependent	Recidivist
False	True	False	True
False	True	False	True

Good Behavior	Age < 30	Drug Dependent	Recidivist
False	False	False	False
True	False	False	False
True	False	True	True
True	False	False	false

Good Behavior	Age < 30	Drug Dependent	Recidivist
True	False	True	True

Good Behavior	Age < 30	Drug Dependent	Recidivist
False	False	False	False
True	False	False	False
True	False	False	false

Use this tree, we can predict the class of the new prisoner Good Behavior = false, Age < 30 = false, Drug dependent = true.

Age < 30 Years (False) -> Drug Dependent (True) -> Recidivist (True)

Hence, the prisoner will belong to Recidivist (TRUE) class.

5. More Classification Measures

In class, we talked about several classification measures, such as accuracy, precision, recall, AUC, etc. There are many other measures out there, each with its own pros and cons.

Do a bit of research of your own to find at least two classification measures that we did not discuss in class. For each measure, describe what it is, how it is calculated, and in which scenarios it is best used.

Answer:

1. **<u>Geometric Mean:</u>** This G-Mean is a metric that finds a balance between classification performance on both False positives and False negatives.

$$G - Mean = \sqrt{Sensitivity \times Specificity}$$

A low mean indicates poor performance in the positives i.e. high false positives even when the negatives are correctly classified. This is an important measure to avoid overfitting the negatives and underfitting the positives.

Applications: This is used where data imbalance is an issue and where False positives and False negatives are equally evil. One application can be disease detection. Traditionally, it has also been used to evaluate the diagnostic abilities of tests.

2. **Discriminant Power:** This is another measure that is a summary of sensitivity and specificity. The formula is given by:

$$DP = \frac{\sqrt{3}}{\pi}(\log X + \log Y)$$

Where X = sensitivity/(1-sensitivty) and Y = specificity/(1-specificty). This measure assesses how well a classifies differentiates between positive and negative cases. It is considered poor if value is <1, limited if <2, fair if <3 and good otherwise.

Applications: As the name suggests, it evaluates how the algorithm distinguishes between positive and negative values has mostly been used in feature selection example by Li & Sleep (2004).

6. THE INTERN - TOMORROW

You are an in-demand, world-traveling, work-all-night consultant who specializes in designing supervised machine learning solutions for clients in a wide-range of industries. You have seen it all and you know what to do. To help you get more done in less time, you have hired an intern from Ivey, who, unfortunately, needs some handholding. Your intern does not understand when to use which classification measure. Your intern keeps getting it wrong. To help your intern learn from your experience, you have decided to look at some previous projects and describe which measure you used, and more importantly, why.

For each project below, describe which measure(s) are best, and why. Also, give an example of a measure which would be horrible to use, and why. List any assumptions you are making, about the dataset, problem, or business priorities that were involved in the project.

a) The fraud department at a bank wanted to predict which transactions were fraudulent. The training dataset had 100K credit card transactions, of which 97K are legit and 3K are fraud.
Answer:
This is a very disbalanced dataset with only 3% of instances as frauds (or positives). So for us the metrics of choice should be Recall. Recall is about capturing cases that have “fraud” with the answer as “fraud”. If we are able to capture all the 3k cases then we would say recall is 100%. So higher recall rate is better in this scenario. A disaster in this case would be to use Accuracy as a measure. You can have a scenario where you have 90% accuracy without identifying a single fraud case.

b) A hospital wanted to predict whether an MRI scan contained cancer.
Answer:
I am assuming that an MRI is a late stage test for Cancer. This information is important because in such scenarios both False Positives (as in detecting cancer when patient do not have) and False Negatives (not detecting cancer when patient actually had) are equally bad. However, since it's a late stage test, other measure has already ruled that the patient had. So, in such scenario, we would be more sensitive to False Positives and would try to minimize that. So, Precision and Recall are good measures for this.
Accuracy would again be bad in this scenario however not as much because the population going through MRI would not be as disbalanced as the first scenario.

c) An IT team wanted to filter spam from email inboxes.
Answer:
Let's say, you are awaiting an important mail from a client and it didn't reach you because IT tagged them as spam. However, if the system is not able to detect a spam, it's annoying but you can just go ahead and delete it. So, here minimizing false positives is more important than False negatives. In this scenario, the best performance measure should be specificity.
A bad measure in this scenario would be to use recall where we are able to identify 100% spam correctly but it will not serve our purpose if it comes at the cost of classifying some important mails as spam.

d) A sports analytics department wants to predict which team will win the match.
Answer:
What is important for this team – predict win when the team wins and predict loose when it loses. Neither of the false positives and false negatives are lethal to this because the purpose of the prediction is actually to understand the variables which would be impacting the win/loose. Accuracy is a good enough measure in this. Using F1 score would be a disaster to use here given it would make things unnecessarily complex for the leadership to understand for a pretty straightforward problem.

e) A city government wanted to build a system to monitor Twitter to see if any local residents were tweeting about emergencies that needed quick response from the police department. They don't trust Twitter that much; they only want to send police in true emergencies.

Answer:

The team wants to respond to only true cases. The assumption here is that it would be very disbalanced data since one multiple people will respond to the same emergency and two people will also raise false alarms. The govt is also trying to minimize the false negatives and hence recall is a good metric here. Again because of unbalanced data accuracy would be a disaster and higher accuracy would not mean that we are detecting the true emergencies correctly.

f) ***Predicting the reoccurrences of breast cancer***

False negatives are probably worse than False positives for this problem. A test can probably clear the false positives but false negatives can lose the follow-up evaluation.
In this scenario, accuracy can be very misleading. We want to select a measure that would reflect better on predicting the problem. Both precision and recall are good measures for this, but we want a balance between these measures. Hence, on of the best measures for this would be F1 score. It conveys the optimum balance between the precision and recall.

7. UNCLE STEVE’S GROCERY STORE

Uncle Steve runs a small, local grocery store. Looking for some customer insights, he has hired you to do some data science. He has given you a few years’ worth of customer transactions, i.e., sets of items that customers have purchased. You have applied an association rules learning algorithm to the data, and the algorithm has generated a large set of association rules. For each of the following scenarios, provide an example of one of the discovered association rules that satisfies the following conditions. (Just make up the rule, using your human experience and intuition!) Also, describe whether and why each rule would be considered subjectively interesting or uninteresting.

a) A rule that has high support and high confidence.

Eggs -> Bread.

Eggs is very common breakfast item and people frequently buy this from local grocery store. Eggs usually go with bread which again is frequently bought from local grocer and not hoarded in bulk from the supermarkets. This signifies that bread would have high support. Since, this combination is often bought together – this rule would have a high confidence and high support.

b) A rule that has reasonably high support but low confidence.

Coffee -> Milk

People often buy fresh milk from the local grocer and hence this is a frequent purchase with high support. People would sometimes buy coffee with milk but chances that they buy it together are not very high. This can be either because people buy coffee packets from big supermarkets or because of the frequency of purchase is very different for the two products. Milk is bought and consumed much more frequently than coffee. This is in contrast to the first example where eggs and bread have similar consumption patter. Hence, this rule would have high support and low confidence though logically this sounds like a good combination.

c) A rule that has low support and low confidence.

Frozen Foods -> Healthy Snacks
Though the sales of healthy stuff are generally on a rise, it is only a small percentage of overall sales at a grocery at the moment. Healthy snacking is a relatively new trend and hence would have a low support. On top of it, people who buy healthy snacks often do not indulge in frozen foods and hence makes this rule a low support and low confidence.

d) A rule that has low support and high confidence.

Bread and cake - > Party Supplies

Party supplies is a a less frequent sales item in any grocery store (generally – not during the festival season). Hence, this would be low support. But we would find high confidence in this rule because people usually throw parties on special occasions which always call for a yummy cake 😊.

7. Association Rules, Two Ways

Consider the following table of customer transactions.

Customer ID	Transaction ID	Items Bought
1	0001	{a, d, e}
1	0024	{a, b, c, e}
2	0012	{a, b, d, e}
2	0031	{a, c, d, e}
3	0015	{b, c, e}
3	0022	{b, d, e}
4	0029	{c, d}
4	0040	{a, b, c}
5	0033	{a, d, e}
5	0038	{a, b, e}

a) Compute the support for itemsets {e}, {b,d}, and {b,d,e} by treating each transaction ID as a market basket (i.e., the normal way).

Support {e} = 8/10 = 0.8

Support {b,d} = 2/10 = 0.2

Support {b,d,e} = 2/10 = 0.2

b) Use the results in part (a) to compute the confidence for the association rules {b,d} → {e} and {e} → {b,d}. Is confidence a symmetric measure?

Confidence {b,d} → {e} = S(b,d,e)/S(b,d) = 0.2/0.2 = 1

Confidence {e} → {b,d} = S(b,d,e)/S(e) = 0.2/0.8 = 0.25

No confidence is not a symmetrical measure. This is because it is dependent on the support of the LHS item. In this case support and frequency {e} is high compared to the support of {b,d} and hence the confidence of {e} → {b,d} is low compared to the reverse.

c) Repeat part (a) by treating each customer ID as a market basket. Each item should be treated as a binary feature (i.e., 1 if an item appears in at least one transaction bought by the customer, and 0 otherwise).

The table will look like follows if the transactions IDs are combined together:

Cust ID	Items Bought
1	{a, b, c, d, e}
2	{a, b, c, d, e}
3	{b, c, d, e}
4	{a, b, c, d}
5	{a, b, d, e}

Support {e} = 4/5 = 0.8

Support {b,d} = 4/5 = 0.8

Support {b,d,e} = 3/5 = 0.6

8. Viva la Vino

Some Smith faculty have started a wine club. At each meeting, members of the club perform blind taste tests of different wine varietals. Members indicate how much they enjoy each varietal, using an integer scale of 1 (worst) to 7 (best). After the most recent meeting, here are the ratings.

	Zin	Pinot Noir	Chard	Merlot	Cab	Pinot Gris
Yuri	7	6	7	4	5	4
Steve		7	6	4	3	4
Gary	3	3	3	1	1	5
Qurat	2	2	1	3	7	4
Brigid	5	6	7	2	3	3

Unfortunately, the club ran out of Zin before Steve had a chance to try it. Luckily, the club has you, a data-driven, clever, and charming Queen's student. Use your skills to predict what Steve would rate Zin. Use user-based collaborative filtering with cosine distance. To predict the rating, find the two nearest neighbors, and take a weighted average of their scores of the item in question.

Hints:

- Recall that the cosine distance is calculated as $\boldsymbol{dist_{cos}(A,B) = 1 - \frac{\sum_{i=0}^{n} A_i B_i}{\sqrt{\sum_{i=0}^{n} A_i^2}\sqrt{\sum_{i=0}^{n} B_i^2}}}$
- When comparing two users, only include items that both users have rated. For example, to compare Yuri and Steve, the calculation would ignore Zin (since Steve hasn't rated it) and would be $1 - \frac{\boldsymbol{7*6+6*7+4*4+5*3+4*4}}{\boldsymbol{\sqrt{7^2+6^2+4^2+5^2+4^2}\sqrt{6^2+7^2+4^2+3^2+4^2}}} = 0.021$.
- To find the weighted average of two ratings R1 and R2, which have a distances D1 and D2 respectively, use the formula: Weighted_Average = ((1-D1)*R1 + (1-D2)*R2) / (2-D1 - D2).
- Round the predicted rating to the nearest integer.

Answer:

I calculated the cosine distances of Steve with other four members to find the two nearest neibours:

Name	Cosine Distance	Rank
Yuri	0.021	1
Gary	0.123	3
Qurat	0.308	4
Brigid	0.027	2

As per this, the distance between Steve and Yuri, Brigid is the least and hence, these are Steve's nearest neighbors.

Calculating the final rating of Zin for Steve = ((1-D1)*R1 + (1-D2)*R2) / (2-D1 - D2)

= ((1-0.021)*7+(1-0.027)*5)/(2-0.021-0.027)

= 6.0003 = Rounding off 6

The final rating of Steve for Zin would be 6.

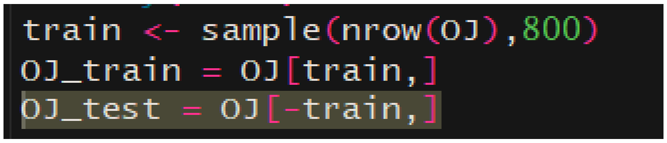
9. YUM, ORANGE JUICE!

One cup of fresh orange juice has 124 mg of vitamin C, which is 200% of the recommended daily intake of vitamin C for an adult. With this as (completely unrelated) motivation, consider the `OJ` data set, which is part of the `ISLR` package.

a) Create a training set containing a random sample of 800 instances, and a test set containing the remaining instances.
b) Using the `svm()` function from the `e1071` package, fit a support vector machine classifier to the training data using cost=0.01, with `Purchase` as the target. Use the `summary()` function to produce summary statistics, and describe the results obtained.
c) What are the training and test error rates?
d) Use the `tune()` function to select an optimal value for cost. Consider values in the range 0.01 to 10.
e) Compute the training and test error rates using this new value for cost.
f) Repeat parts (b) through (e) using a support vector machine with a radial kernel. Use the default value for gamma.
g) Overall, which approach seems to give the best results on this data?

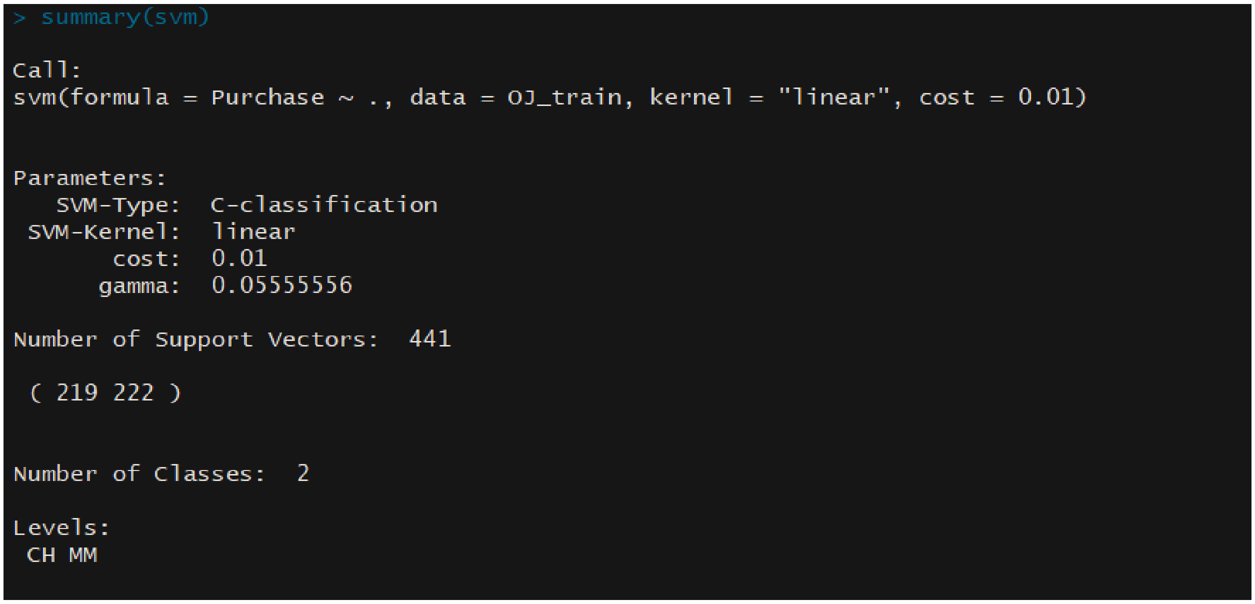
Answer:

a)



```
train <- sample(nrow(OJ),800)
OJ_train = OJ[train,]
OJ_test = OJ[-train,]
```

b)



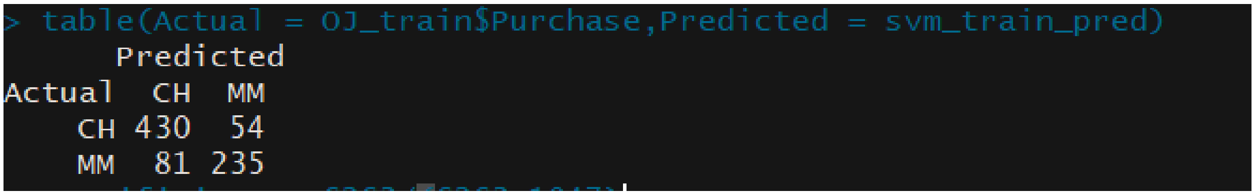
```
> summary(svm)

Call:
svm(formula = Purchase ~ ., data = OJ_train, kernel = "linear", cost = 0.01)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  0.01
      gamma:  0.05555556

Number of Support Vectors:  441

 ( 219 222 )


Number of Classes:  2

Levels:
 CH MM
```

Description of results:

The model selects 441 out of 800 instances as support points. Otherwise, the summary doesn't tell us much except for that there are two classes – CH and MM.

c)

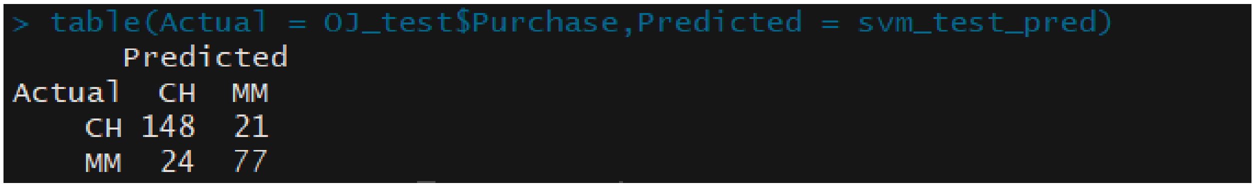
Train confusion matrix



```
> table(Actual = OJ_train$Purchase,Predicted = svm_train_pred)
      Predicted
Actual  CH  MM
    CH 430  54
    MM  81 235
```

Train error rate = 16.9%

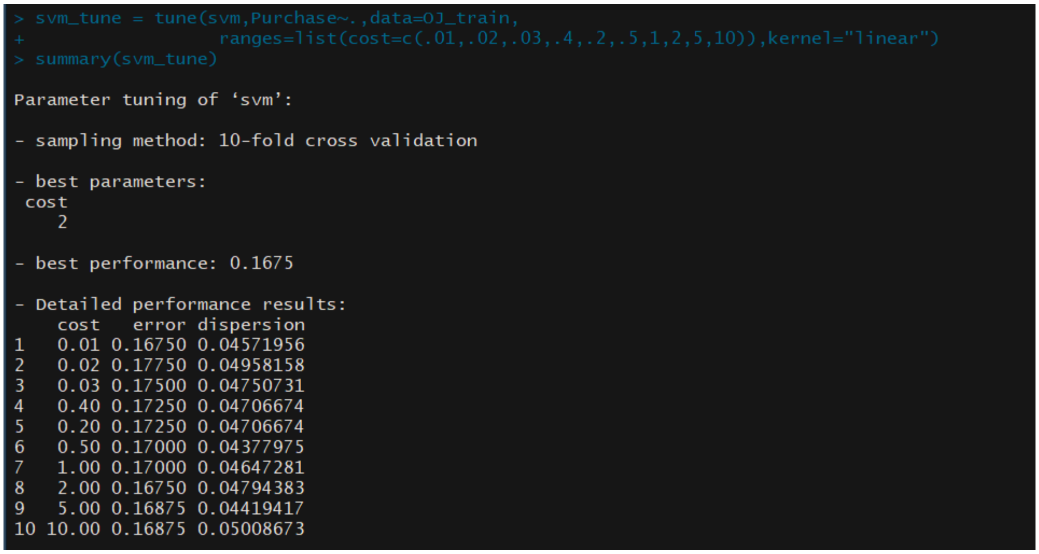
Test confusion matrix



```
> table(Actual = OJ_test$Purchase,Predicted = svm_test_pred)
      Predicted
Actual  CH  MM
    CH 148  21
    MM  24  77
```

Test error rate = 16.7%

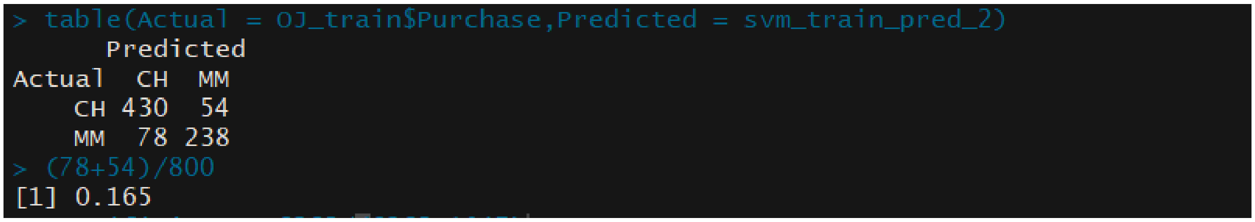
d)



```
> svm_tune = tune(svm,Purchase~.,data=OJ_train,
+                 ranges=list(cost=c(.01,.02,.03,.4,.2,.5,1,2,5,10)),kernel="linear")
> summary(svm_tune)

Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost
    2

- best performance: 0.1675

- Detailed performance results:
    cost   error dispersion
1   0.01 0.16750 0.04571956
2   0.02 0.17750 0.04958158
3   0.03 0.17500 0.04750731
4   0.40 0.17250 0.04706674
5   0.20 0.17250 0.04706674
6   0.50 0.17000 0.04377975
7   1.00 0.17000 0.04647281
8   2.00 0.16750 0.04794383
9   5.00 0.16875 0.04419417
10 10.00 0.16875 0.05008673
```

e)

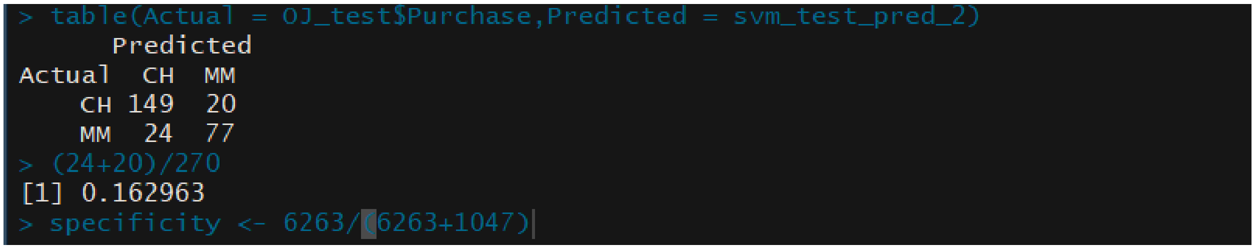
Train confusion matrix



```
> table(Actual = OJ_train$Purchase,Predicted = svm_train_pred_2)
      Predicted
Actual  CH  MM
    CH 430  54
    MM  78 238
> (78+54)/800
[1] 0.165
```

Train error rate = 16.5%

Test confusion matrix



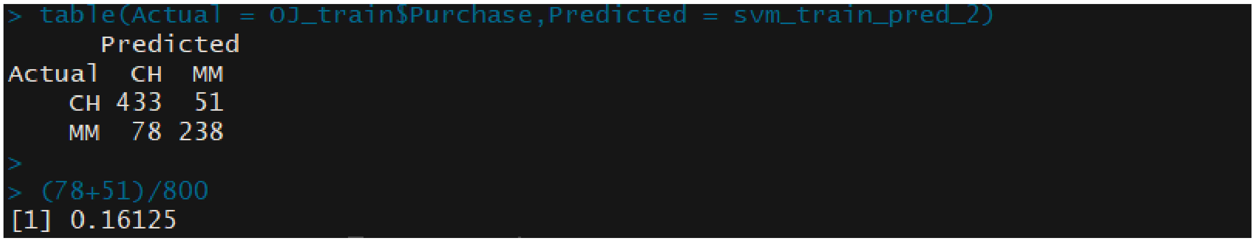
```
> table(Actual = OJ_test$Purchase,Predicted = svm_test_pred_2)
      Predicted
Actual  CH  MM
    CH 149  20
    MM  24  77
> (24+20)/270
[1] 0.162963
> specificity <- 6263/(6263+1047)
```

Test error rate = 16.2%

f)

I tuned the svm model on radial kernel with costs ranging from 0.01 to 10. The best error rate is 0.176 at the cost value of 0.5.

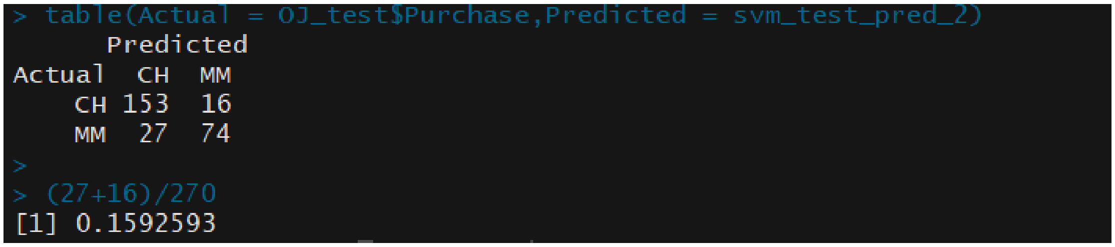
Train confusion matrix



```
> table(Actual = OJ_train$Purchase,Predicted = svm_train_pred_2)
      Predicted
Actual  CH  MM
    CH 433  51
    MM  78 238
>
> (78+51)/800
[1] 0.16125
```

Train error rate = 16.1%

Test confusion matrix



```
> table(Actual = OJ_test$Purchase,Predicted = svm_test_pred_2)
      Predicted
Actual  CH  MM
    CH 153  16
    MM  27  74
>
> (27+16)/270
[1] 0.1592593
```

Test error rate = 15.9%

g)

Comparison of models:

Model	Train Error	Test Error
Linear Kernel cost - 0.01	16.9%	16.7%
Linear Kernel cost – 2	16.5%	16.2%
Radial Kernel cost - 0.5	16.1%	15.9%

Among these, the best model with the least error on both i.e. test and train data is Radial Kernel with cost parameter of 0.5.

REFERENCES

https://pdfs.semanticscholar.org/ff0b/d954443338708279f97feb05d6b29e41382c.pdf

https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/

https://users.ece.cmu.edu/~fcondess/preprints/rejection_measures.pdf

https://support.sas.com/resources/papers/proceedings17/0942-2017.pdf