CASS
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON
EST 1894

Group Coursework Submission Form

Specialist Masters Programme

Please list all names of group members: (Surname, first name) 1. Bian, Yu 2. Huang, Ruiqi 3. Liu, Ching-Wei	4. Liu, Mengjie 5. Wang, Zhaofei 6. 7. **GROUP NUMBER:** 5

MSc in:
Business Analytics

Module Code:
SMM636

Module Title:
Machine Learning

Lecturer:
Rui Zhu

Submission Date:
21/3/2019

Declaration:

By submitting this work, we declare that this work is entirely our own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the coursework instructions and any other relevant programme and module documentation. In submitting this work we acknowledge that we have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. We also acknowledge that this work will be subject to a variety of checks for academic misconduct.

We acknowledge that work submitted late without a granted extension will be subject to penalties, as outlined in the Programme Handbook. Penalties will be applied for a maximum of five days lateness, after which a mark of zero will be awarded.

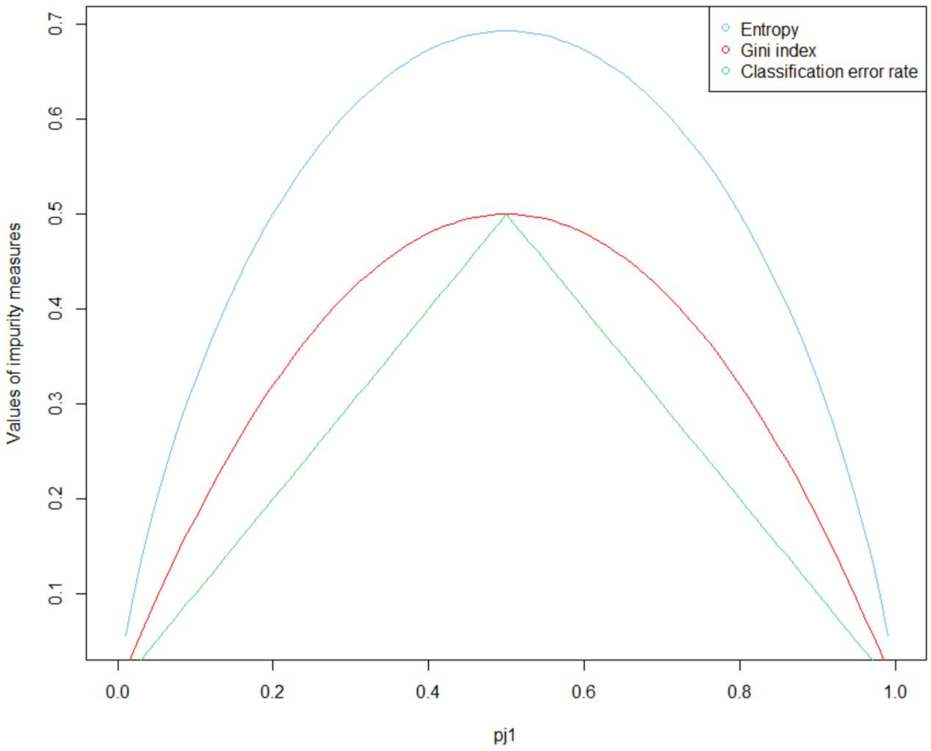
Marker's Comments (if not being marked on-line):

Deduction for Late Submission:

Final Mark: %

Question 1

(1)



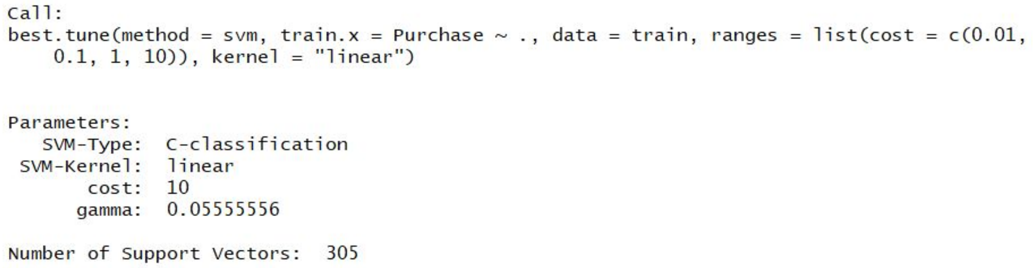
(2)

Overall, all three measures attain their maximum value when *pj1*=0.5, which means the binary classes in the node have the equal probability, the minimum values for the measures are attained when when *pj1* equals to 0 or 1, which means all the records belong to the same class in this case. In addition, entropy shows the highest impurity value all the time, followed by Gini index and classification error rate, because entropy can be higher than 0.5, while Gini index and classification error rate always keep no more than 0.5.

Question 2

(1)

Best model:

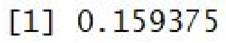


```
Call:
best.tune(method = svm, train.x = Purchase ~ ., data = train, ranges = list(cost = c(0.01, 
    0.1, 1, 10)), kernel = "linear")


Parameters:
   SVM-Type:  C-classification 
 SVM-Kernel:  linear 
       cost:  10 
      gamma:  0.05555556 

Number of Support Vectors:  305
```

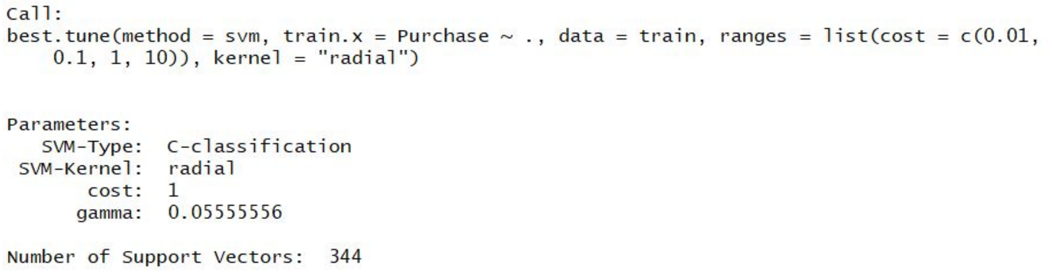
Test error rate:



```
[1] 0.159375
```

(2)

Best model:

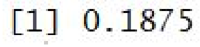


```
Call:
best.tune(method = svm, train.x = Purchase ~ ., data = train, ranges = list(cost = c(0.01,
    0.1, 1, 10)), kernel = "radial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.05555556

Number of Support Vectors:  344
```

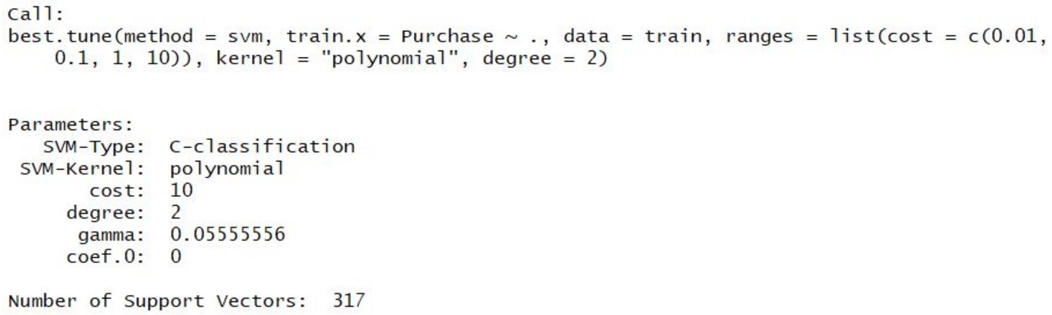
Test error rate:



```
[1] 0.1875
```

(3)

Best model:

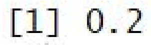


```
Call:
best.tune(method = svm, train.x = Purchase ~ ., data = train, ranges = list(cost = c(0.01,
    0.1, 1, 10)), kernel = "polynomial", degree = 2)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  10
     degree:  2
      gamma:  0.05555556
     coef.0:  0

Number of Support Vectors:  317
```

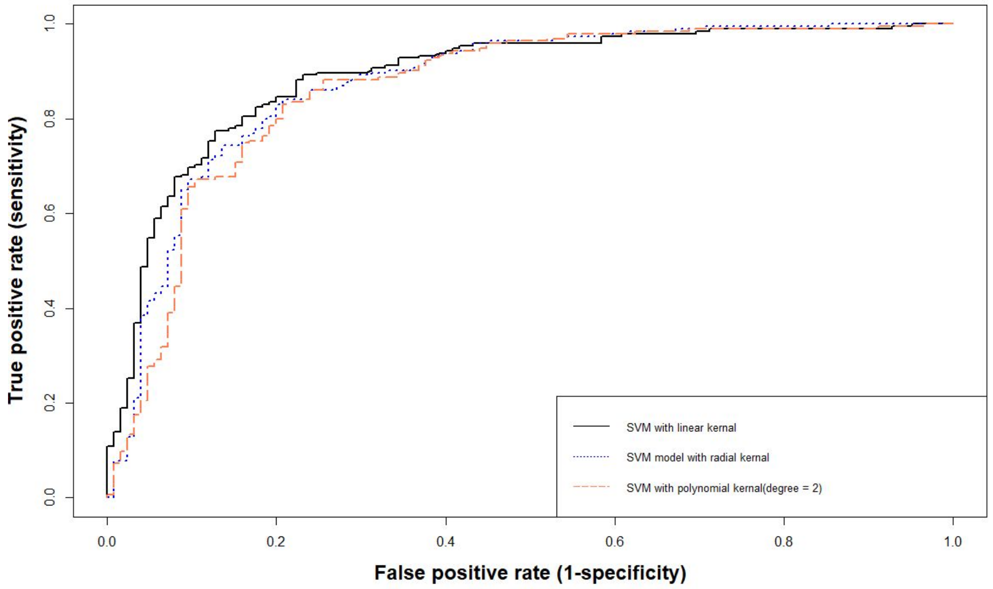
Test error rate:



```
[1] 0.2
```

(4)

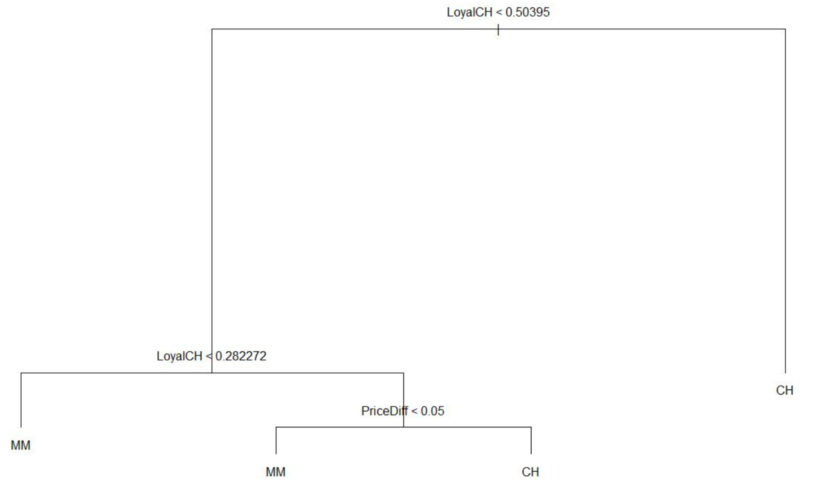
The plot with three ROC curves for the test predictions in (1), (2) and (3):



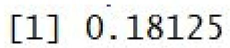
Out of the three classification methods, linear kernel provides the largest AUC (area under the ROC curve), followed by radial kernel and polynomial kernel. AUC measures the prediction accuracy, the larger the AUC, the better the prediction accuracy, therefore in this case linear kernel model has the best prediction power and polynomial kernel has the least prediction power. The findings are in line with test error generated before, with linear kernel has the smallest test error of 0.1594. The reason could be that the classification boundary are more strictly linear than other shapes, therefore the linear model fits the data best.

(5)

The plot of the pruned tree (with the best size of 4 terminal nodes):



Test error rate:

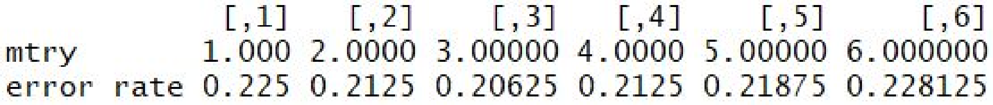


```
[1] 0.18125
```

First, we use the cv.tree() function over the training data to find an optimal tree size determined by cross-validation, then we can see the result showing that when the size is 4, the dev has the smallest value of 140, which means that if we prune the tree with 4 terminal nodes, we can get a classifier with the smallest error rate. When looking in this pruned tree in detail, it indicates that the variable "LoyalCH < 0.50395" in the highest branch gives the first classification, followed by "LoyalCH < 0.282272", and "PriceDiff < 0.05". Finally, with this decision tree, we can apply to the test data and then make a prediction of the outcome. The test error rate of 0.181 shows that the decision tree performs well in predicting the test label.

(6)

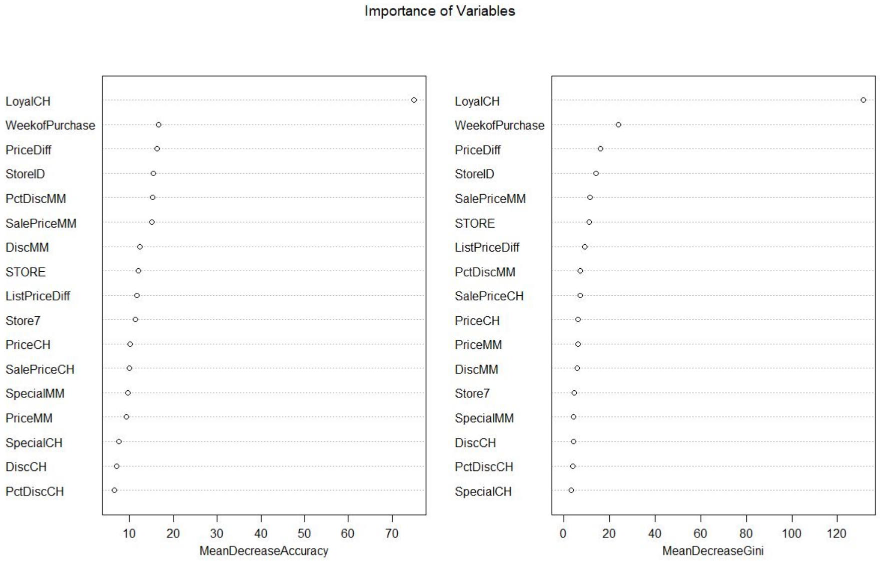
Test error rates of the 6 mtry values:



```
            [,1]   [,2]    [,3]   [,4]    [,5]     [,6]
mtry       1.000 2.0000 3.00000 4.0000 5.00000 6.000000
error rate 0.225 0.2125 0.20625 0.2125 0.21875 0.228125
```

To find the best random forest model, firstly we apply the random forest algorithm over the training data, with changing the "mtry" argument from 1 to 6, meaning at each split, there are 1 to 6 variables randomly sampled as candidates. After that, we use these 6 trained random forest on the testing data to make predictions, then the test error rate after prediction can be calculated, as shown in the table. In this result, we can see that when mtry = 3, we can obtain the lowest error rate, indicating that when 3 variables randomly sampled as candidates at each split, the random forest model is the best one.

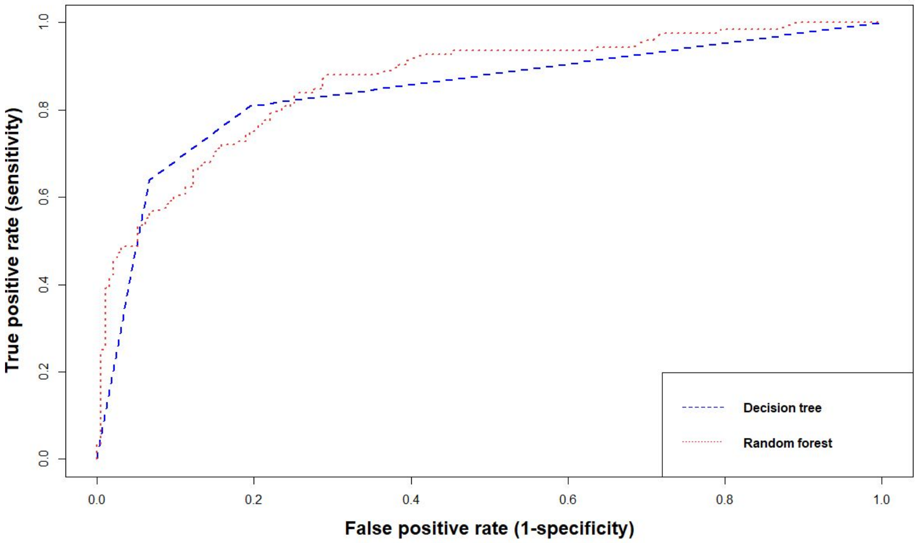
The plot of the variable importance for the model with the best test error (mtry = 3):



The two plots show the variable importance for the random forest model with mtry = 3. The variable importance measures the mean Accuracy and Gini index are decreased by splits over a given predictor. As we can see in both of the plots, variable “LoyalCH” plays the most important role in classification, followed by “WeekofPurchase”, “PriceDiff” and so on.

(7)

The plot with two ROC curves for the test predictions in (5) and (6):



This plot shows that the area under the ROC curve (AUC) for the random forest, which is 0.865, is larger than that for the decision tree, which is 0.842. This indicates that the model of the random forest has a better prediction accuracy. However, if we compare this result with the error rates of these two methods, we can find that the error rate of the random forest is 0.206, which is higher than that of the decision tree. This contradiction may be because of the number of the two classes in the data is imbalanced. Therefore, the measure of the ROC curve, using true positive rate and false positive rate gives a better prediction considering the sensitivity and the specificity. Under this reason, we can conclude that with different classification thresholds, the prediction power of the random forest is better, since it can more precisely measure the sensitivity and the specificity.

(8)

The ROC curve shows the tradeoff between sensitivity and specificity, and the corresponding AUC value measures the accuracy of the classifying model. The AUC values for SVMs with linear, radial and polynomial kernels are 0.891, 0.875, 0.862, respectively; the AUC value for random forest is 0.865 and for decision tree is 0.842. We can see that the SVM with linear kernel has the highest AUC, meaning it has the highest predicting power out of all five models. One potential explanation for this is that the OJ dataset is reasonably clean and has few outliers, so the SVMs generally works better than decision trees. Also it has more of a linear classification boundary, which explains why the linear SVM outruns all other models.

Overall, if we have knowledge of the boundary shape and can apply a supervised learning approach, it is often better than decision tree in terms of prediction accuracy. Decision trees are also very non-rubust with various training input. However, in other cases, decision trees are more interpretable and are more suitable for qualitative features.