Question 5

(a). The supervised methods I decided to classify the reviews based on the numerical features are logistic regression, bagging, random forest and support vector machine classifier (Gaussian Radial Basis Kernel). Logistic regression is a model that derives the probability of an observation being classified positive in the form of a mathematical equation of predictor variables with its corresponding coefficients that measures the effect of predictor variable classifying an observation of the class of interest. However, classification accuracy depends on that one model for logistic regression unless bagging is implemented in the logistic regression. Unlike logistic regression, bagging and random forest are ensemble methods that compute the overall classification accuracy by averaging the accuracy of multiple fitted classification trees, reducing the variance in the predictions. In addition, random forest increases the independency of trees unlike bagging by randomly splitting the trees such that a subset of variables is considered for each split. Support Vector Machine classifiers are supervised classifiers that deal with complex linear and non-linear boundaries between classes. Many kernel functions such as Polynomial kernel Gaussian Radial Basis Function kernel and Laplace Radial Basis Function kernel can be implemented to the data such that hyperparameters can be used to tune a model and most importantly, improve classification accuracy.

(b). I decided to split the training-validation and test data by 70% and 30% respectively and performed a 3 folds cross validation (66.67% training set and 33.33% validation set) on each of the tuned supervised models and then obtained the average accuracy of 5 replications of the procedure ran repeatedly for each tuned model. Since the results of running a supervised algorithm are arbitrary for each replication, it would be better to run a large number of replications (>10) in order to achieve a more accurate mean classification accuracy of all replications considered. However, I used 5 replications due to slow computation on my laptop for each tuned supervised model.

Support Vector Machine Classifier

The reason I decided to use the Gaussian Radial Basis Function Kernel is because it seems to be more flexible and usually a better fit to the data than the polynomial kernel. Also, Laplace Radial Basis Function kernel is similar to the Gaussian Radial Basis Function kernel as they both depend on the Euclidean distance (continuous predicator variables lies in the Euclidean space) between observations so either method can be chosen.

The values of the cost hyperparameters I have used are 5,50 and 100. The cost hyperparameter controls the complexity of the model, allowing a certain amount of overlapping of observations between classes as cost gets larger. The sigma hyperparameter values I used are 0.033,0.066 and 0.1 which control the flexibility of the model. Therefore, 9 tuned support vector machine classifiers are considered to compare with other supervised models. It would be better to include more hyperparameters and compare more tuned models but due to the slow computation of my laptop, I decided to tune 9 support vector machine classifiers.

The results I have obtained from running 3 folds cross validation and 5 replications on each of the tuned support vector machine models are shown below:

Cost	Sigma	Validation Accuracy
5	0.033	0.6514440
5	0.066	0.6342002
5	0.1	0.5619368
50	0.033	0.6509454
50	0.066	0.6348234
50	0.1	0.5619784
100	0.033	0.6511948
100	0.066	0.6355716
100	0.1	0.5620200

The validation accuracy for each tuned model does not seem to be extremely high but still is sufficiently good at classifying validation data. The best tuned support vector machine model is the model with cost=5 and sigma=0.033. However, these results will be compared with other tuned supervised models.

Advantages of SVM classifiers

Unlike logistic regression, bagging and random forest, this family of classifier has the ability to manage data with a non-linear boundary between classes as well as offering the choice of using different kernel functions, allowing more model flexibility compared to other supervised methods. However, it depends on the structure of the data to find the best supervised method for that particular dataset.

Bagging and Random Forest

The hyperparameter for random forest is the number of classification trees splits known as mtry in R. The values of hyperparameter I considered are 5,20,40 and 79. 79 classification tree splits are equivalent to running the bagging algorithm on the data such that number of variables – 1=79. The higher the number of variables considered for a split, the more complex the model will be. It would be better to include more hyperparameters values, clearly more than 4 and compare more tuned models to improve the accuracy of the model. Due to the slow computation of my laptop, I decided to tune 4 random forest classifiers.

The results I have obtained from running 3 folds cross validation and 5 replications on each of the tuned bagging and random forest models are shown below:

Number of splits (mtry)	Validation Accuracy
5	0.7048839
20	0.7094546
40	0.7099951
79	0.7060057

The validation classification accuracy for the tuned bagging and random forest classifiers are higher than the validation classification accuracy for the tuned support vector machine classifier. The best tuned random forest classifier is mtry=40 with 70.99% validation accuracy.

Advantages of Bagging and Random Forest

The advantages for both bagging and random forest are that they are both ensemble methods that uses multiple weak classification trees of the same data, resampled with replacement (bootstrapping). The classification accuracy of bagging and random forest is the average of the classification accuracy of each classification tree, and this reduce the variance in the prediction as classification trees causes high variance in predictions. Reducing the variance in the predictions will improve the classification accuracy, compared to only considering one classification tree and a logistic regression model. In addition to random forest, random forest decreases the correlation between trees by considering each split of the fitted classification trees, observing a random subset of variables for each split. Random forest also observes the importance of predictor variables considered related to the classification accuracy which is statistically interesting to examine and see what predictor variable contributes the most to classifying an observation. Unlike support vector machine classifiers, bagging and random forest can be applied to classifying an observation such that there are more than two classes.

Logistic regression

The hyperparameter for logistic regression is lambda which is a penalisation and complexity parameter that penalises any additional unnecessary parameters that do not contribute to classifying the observations to a class. This concept is known as Lasso which shrinks the coefficients to zero that have a magnitude close to zero or do not explain the classification of an observation. The smaller the lambda, more variables will be

used in the model and the larger the lambda, less variables will be used in the model. The values of lambda I have used is 0.5, 5, 35, and 50. It would be better to include more hyperparameters values, clearly more than 4 and compare more tuned models to improve the accuracy of the model. Due to the slow computation of my laptop, I decided to tune 4 logistic regression models.

The results I have obtained from running 3 folds cross validation and 5 replications on each of the tuned bagging and random forest models are shown below:

Lambda	**Validation Accuracy**
0.5	0.7006851
5	0.6054846
35	0.5242052
50	0.5239975

The validation accuracy for the tuned logistic regression models with lambda equal to 5,35,50 is lower than the validation accuracy for the tuned support vector machine classifier and random forest classifier. However, the validation accuracy for the tuned logistic regression models with lambda equal to 0.5 is higher than the validation accuracy for the tuned support vector machine classifier but lower than the bagging and random forest classifier. The best tuned logistic regression model is the model with lambda 0.5 of approximately 70% validation accuracy. This indicates that a larger number of variables should be included in the model compared to models with lambda values 5,35 and 50.

Advantages of Logistic Regression

Logistic regression is easy to interpret such that the coefficient of the predictor variable is the size of the effect on the classification of an observation and can examine the importance of variable by observing the size of the coefficients. It is more computationally fast and efficient compared to support vector machines classifiers and random forest classifiers which require more computation time. Similar to bagging and random forest (classification trees), logistic regression can be extended to multinomial regression such that the model can predicts the class of an observations where there are more than two classes whereas support vector machine classifier can only consider a binary variable with only two classes.

(c). The best supervised method to predict the class of a review is random forest with the highest validation accuracy of approximately 70%. This can be easily compared by the summary statistics of mean validation accuracies of 5 replications for each supervised method below:

	Minimum	1st Quartile	Median	3rd Quartile	Maximum	Mean
Support Vector Machines	0.6446384	0.6468204	0.6533666	0.6552369	0.6602244	0.6514440
Random Forest	0.6965732	0.7048922	0.7090343	0.7160224	0.7238155	0.7099951
Logistic Regression	0.6853583	0.6892145	0.7028037	0.7108125	0.7171340	0.7006851

The predictive performance of the best model for classifying unseen data has a classification accuracy of 70.6% which seems to classify unseen observations sufficiently well. Also, I observed the ability of the best model for predicting correctly negative and positive reviews by computing the specificity (true negative rate) and sensitivity (true positive rate) (given by the output of confusionMatrix()). The specificity of the model is 0.6972, indicating that 69.72% of the observations is classified as negative correctly. The model seems to correctly classify reviews as negative quite well but not as well as correctly classify reviews as positive since the sensitivity of the model is 0.7139, indicating that the model correctly classifies 71.39% of the reviews as positive.

Code

```
data=read.csv("data_rotten_tomatoes_review.csv")
str(data) #check the class of each variable
```

```
library(caret)
data$class=factor(data$class)#convert class from character to a factor
range(data[,-c(1,81)]) #variables are measured in different magnitudes so need to
standardise the data
data[,-c(1,81)]=scale(data[,-c(1,81)])#standardise data
data=data[,-1] #interested in classifying the reviews as positive or negative depending
on the numerical features
library(doParallel)
cl <- makeCluster(6,setup_strategy = "sequential") #speed up computations and my laptop
(macbook air) has 8 cores so kept 2 free as shown in lab 9
registerDoParallel(cl) #enable pararell computing

#split data into training and test set, 70% and 30% respectively
train_val=createDataPartition(data$class,p=0.7,list=FALSE)
data_train=data[train_val,]
data_test=data[-train_val,]

#3 folds cross validation is applied when tuning models for 5 replications for each
supervised method (used a low number of replications due to slow replication
kfoldscvtrain=trainControl(method="repeatedcv",number=3,repeats=5)

#Used 3 cost parameters and sigma parameters since computational time was slow on my
laptop
set.seed(4573)

#compute the hyperparameters
tune_grid_for_svm=expand.grid(C=c(5,50,100),
                              sigma=c(0.033,0.066,0.1))

svm_grbf=train(class~.,data=data_train,method="svmRadial",trControl=kfoldscvtrain,
               tuneGrid=tune_grid_for_svm)
svm_grbf

#Used 4 values of classification tree splits to tune random forest models since
computational time was slow on my laptop
set.seed(6473)
#compute the hyperparameters
tune_grid_for_random_forest=expand.grid(mtry=c(5,20,40,79))

random_forest=train(class~.,data=data_train,method="rf",trControl=kfoldscvtrain,
                    tuneGrid=tune_grid_for_random_forest)
random_forest


#Used 4 values of lambda to tune logistic regression models since computational time
was slow on my laptop
set.seed(6476)
#compute the hyperparameters
tune_grid_for_logistic_reg=expand.grid(alpha=0,lambda=c(0.5,5,35,50))

logistic_reg=train(class~.,data=data_train,method="glmnet",trControl=kfoldscvtrain,
                    tuneGrid=tune_grid_for_logistic_reg)
logistic_reg

stopCluster(cl) #Turn of pararell computing

#Obtain summary statistics of the accuracy of the tuned models of all supervised
methods for 5 replications of each supervised method. The best supervised method for
this data is random forest with the highest accuracy out of the other supervised
methods of approximately 70%
set.seed(4352)
comp=resamples(list(svm_radial= svm_grbf,glmnet=logistic_reg,rf=random_forest))
summary(comp)

#Predictive performance of the best model (gives overall classification accuracy, the
sensitivity and specificity) (random forest)
set.seed(5352)
```

```
class_hat=predict(random_forest,newdata=data_test)
confusionMatrix(class_hat,data_test$class,positive = 'positive')
```