Question 1

No R coding is required for this question. Figure 1 below shows the classification decision boundaries obtained by training four different models on a 2D dataset (X1,X2,Y) where Y is a 2-class categorical response variable. Indicate, in the table provided in your answer document, which of the scenarios represented in Figure 1 the models proposed in column A correspond to. (Indicate N/A if a model proposed in column A does not match any of the scenarios of Figure 1.) In column C, provide a <u>brief</u> explanation for your answer about each of the proposed models.

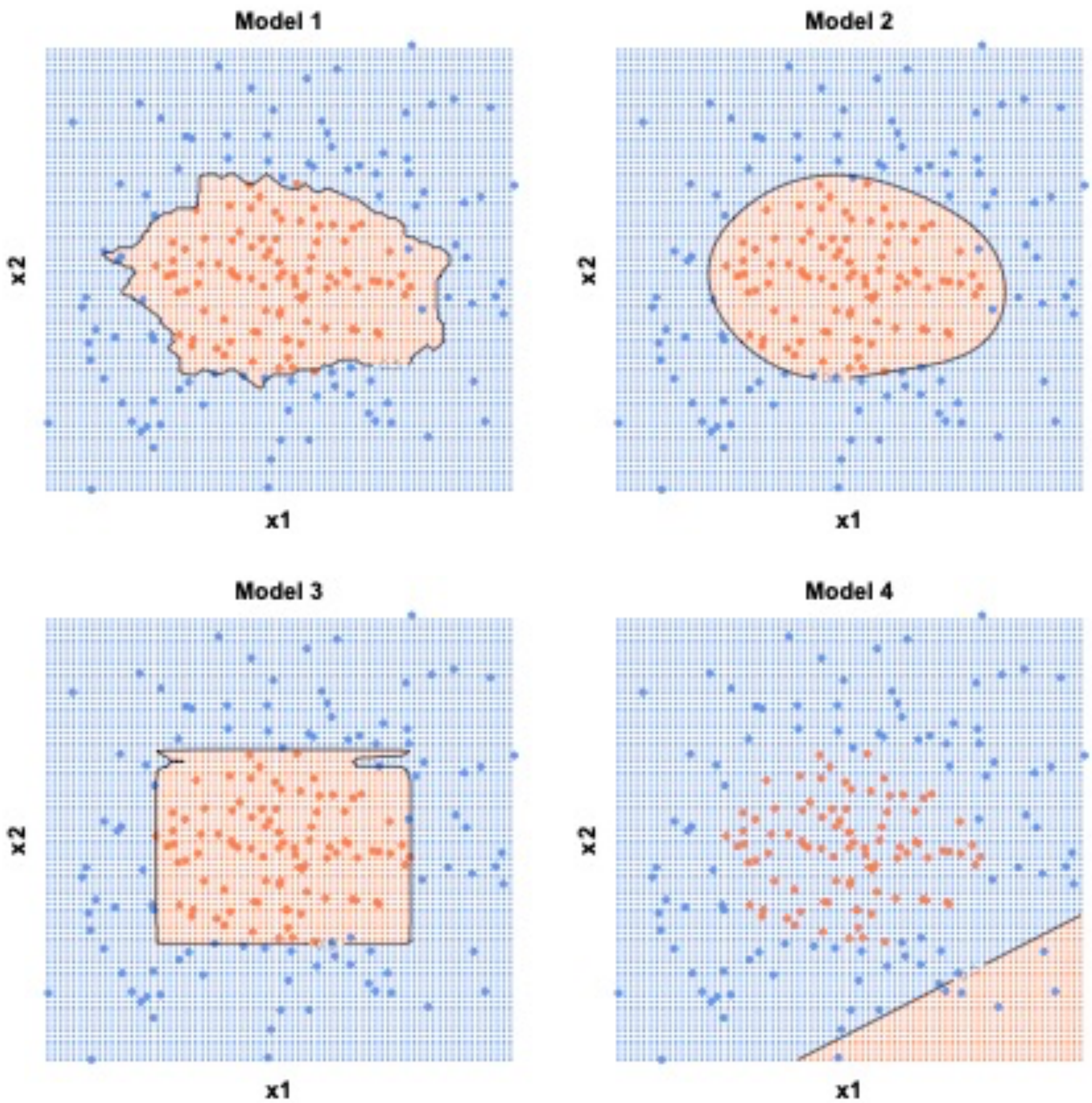


Figure 1 - Classification boundaries from 4 models on the same 2-class dataset (X1, X2, Y); blue and red dots depict the labelled data, and the areas correspond to the decision boundaries of each classifier.

Answers to Question 1

Table 1 - Match the models proposed in Column A to the models depicted in Figure 1 of Question 1. Indicate N/A when a model is not represented in Figure 1. Provide a brief explanation for each of your answers in column C.

(A) Proposed model	(B) Model of Figure 1	(C) Explanation for your answer in column B
A random forest	**Model 3**	**Clear sequence of horizontal and vertical decisions in the branching pattern, typically associated with a tree-based decision process**
A logistic regression model	**Model 4**	**Linear boundary (+ associated with poor classification for this nonlinear problem)**
A Quadratic Discriminant Analysis with 3 Gaussian components	**N/A**	***3 components would be used for a 3D dataset, whereas this one is 2D***
A Support Vector Machine using a radial basis function	**Model 2**	**Clear radial kernel pattern**
A lasso classifier with an extremely large shrinkage parameter	**N/A**	***This model would remove one of the two predictors; this would yield the boundary to be either a horizontal or vertical line***
A kNN classifier (with k=5)	**Model 1**	**A non-regular decision region contour resulting from low value of k; the only plausible model for the top-left scenario in Fig 1**

Question 2

Run the following R instructions to load the required dataset and libraries:

```
library(randomForest)
library(gbm)
dat = read.csv(file="CA2_2021-22.dat", stringsAsFactors=TRUE)
```

Here the response variable of interest is `Sale_Price`, which corresponds to the selling price of a sample of US dwellings, in $1,000's. All other variables in the dataset are used as potential predictors. Do **not** perform any action on the predictors unless instructed to do so. Do **not** use any other package (such as `caret` or `tidymodels`) for this question. Provide your answers in the table below.

(1) Name which numerical features in the dataset have Pearson correlation of over 90% in absolute value, if any.

(2) Implement a simple (i.e. not repeated) 10-fold cross-validation framework, to train and test 3 models, namely a GLM, a random forest and a gradient boosting model, using all variables to predict sale prices. Set the random seed to 4061 before running the cross-validation code.

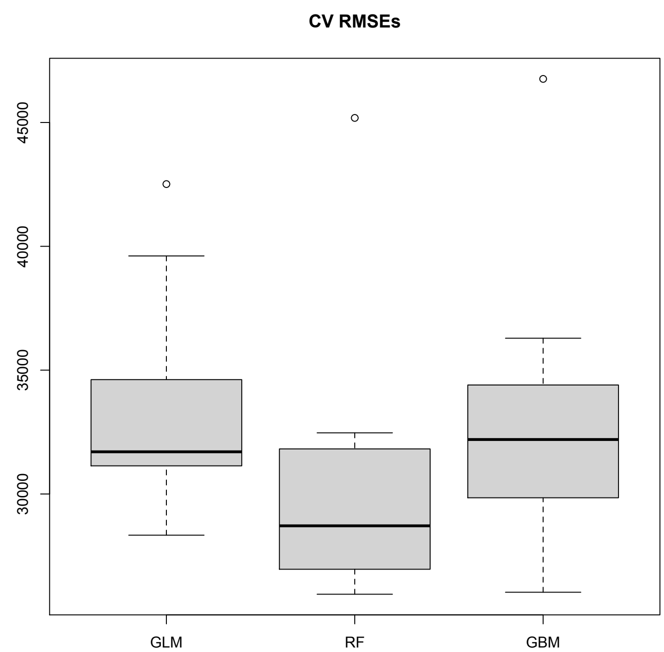
  (a) Report the mean cross-validated RMSEs for the 3 models.
  (b) Provide a boxplot of test-set RMSEs for the 3 models (within one figure).

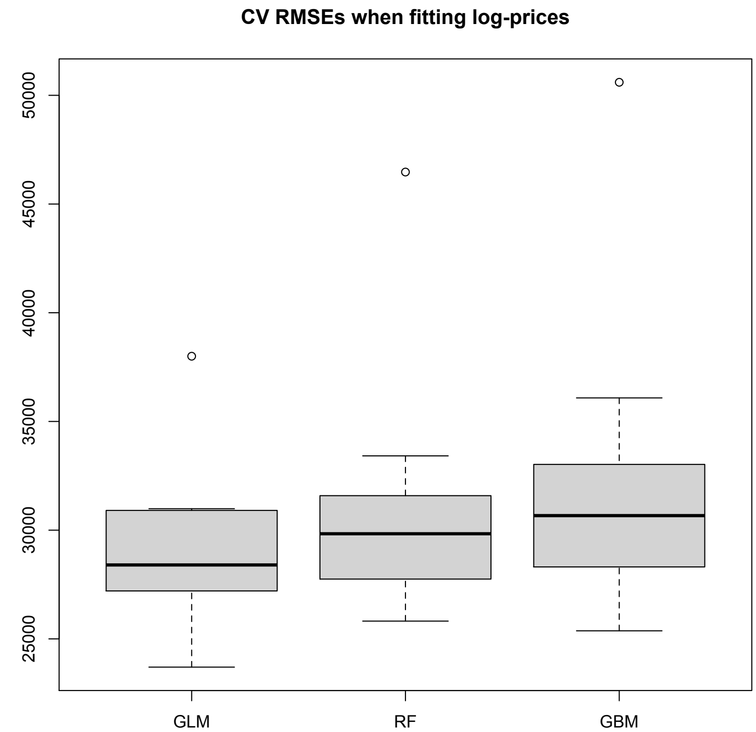
(3) Perform the same cross-validation as in (2) but training the models on `log(Sale_Price)`.

  (a) Report the mean cross-validated RMSEs for the 3 models, in the scale of the original `Sale_Price` variable.
  (b) Provide a boxplot of test-set RMSEs for the 3 models (within one figure) , in the scale of the original `Sale_Price` variable.

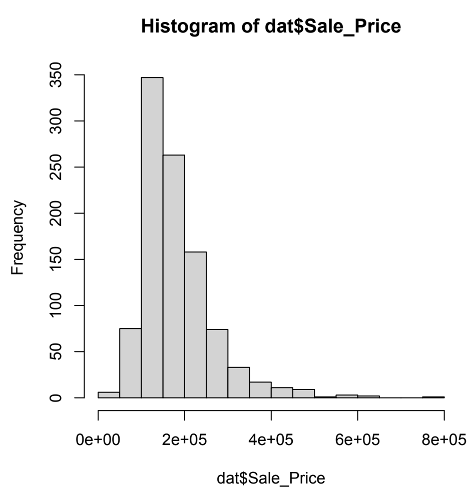
(4) Explain any differences you may find between your results in (2) and (3). If you could not obtain a complete set of results in previous steps, describe what you would have expected to see.

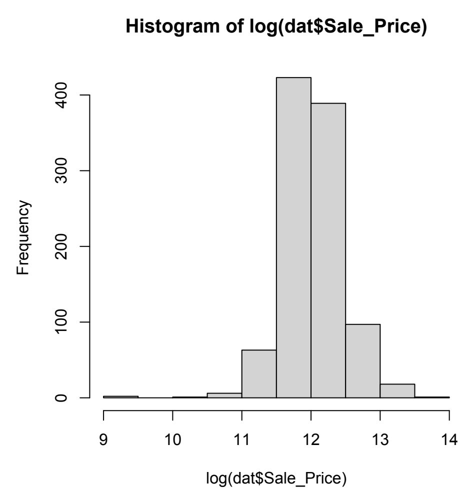
Answers to Question 2

Question	Your answer
1	There are no correlations greater than 89.5% in absolute value in this dataset.
2(a)	GLM RMSE: 33251.27 RF RMSE: 30270.95 GBM RMSE: 33006.23
2(b)	
3(a)	GLM RMSE: 29117.23 RF RMSE: 31071.46 GBM RMSE: 32334.20

<table>
<tr>
<td>3(b)</td>
<td>


</td>
</tr>
<tr>
<td>4</td>
<td>
The distributions of CV RMSEs for RF and GBM have not changed greatly after log-transforming the dependent variable Y, unlike for GLM. Relatively speaking, RF “does” slightly better than GBM in both frameworks.
However we notice a significant improvement in performance of the GLM, which becomes the best-performing model once we use log-transformed prices. This is due to the fact that the original data has significant right-skewness, which is known to affect the performance of GLMs since they assume Gaussian-distributed observations:





(We could also try using a different distribution when fitting the GLM.)
</td>
</tr>
</table>

R code for Question 2:

```
M = round(cor(dat),3)
diag(M) = 0
max(abs(M))

# uncomment this line for question (3):
# dat$Sale_Price = log(dat$Sale_Price)
n = nrow(dat)
K = 10
folds = cut(1:n, K, labels=FALSE)
rmse.glm = rmse.gbm = rmse.rf = numeric(K)
rmse.glm.os = rmse.gbm.os = rmse.rf.os = numeric(K)
set.seed(4061)
for(k in 1:K){
  itrain = which(folds!=k)
  dtrain = dat[itrain,]
  dtest = dat[-itrain,]
  # GLM
  glmo = glm(Sale_Price~., data=dtrain)
  glmo.p = predict(glmo, newdata=dtest, type="response")
  rmse.glm[k] = sqrt(mean((glmo.p-dtest$Sale_Price)^2))
  rmse.glm.os[k] = sqrt(mean((exp(glmo.p)-exp(dtest$Sale_Price))^2))
  # RF
  rfo = randomForest(Sale_Price~., data=dtrain)
  rfo.p = predict(rfo, newdata=dtest)
  rmse.rf[k] = sqrt(mean((rfo.p-dtest$Sale_Price)^2))
  rmse.rf.os[k] = sqrt(mean((exp(rfo.p)-exp(dtest$Sale_Price))^2))
  # GBM
  gbmo = gbm(Sale_Price~., data=dtrain, distribution="gaussian")
  gbmo.p = predict(gbmo, newdata=dtest, n.trees=100)
  rmse.gbm[k] = sqrt(mean((gbmo.p-dtest$Sale_Price)^2))
  rmse.gbm.os[k] = sqrt(mean((exp(gbmo.p)-exp(dtest$Sale_Price))^2))
}

boxplot(rmse.glm,rmse.rf,rmse.gbm)
c(mean(rmse.glm),mean(rmse.rf),mean(rmse.gbm))

boxplot(rmse.glm.os,rmse.rf.os,rmse.gbm.os)
c(mean(rmse.glm.os),mean(rmse.rf.os),mean(rmse.gbm.os))

io = order(importance(rfo),decreasing = T)
cbind(importance(rfo)[io,])
summary(gbmo)
```