Math 437 - Midterm Exam 2

Nicholas De Santos

Due April 26, 2022 at 11:59 PM

```
# Packages you definitely need
library(ISLR2)
library(ggplot2)
library(class)
library(e1071)
library(car)
library(MASS)
library(matlib)
library(glmnet)
library(insuranceData)
library(dplyr)
library(yardstick)
library(rsample)
library(tidyverse)
library(broom)
library(leaps)
library(caret)
library(lattice)
```

This exam is worth a total of **105** points, but is graded out of **100** points.

Conceptual Problem (16 pts total)

Suppose that an observation has prior probability 1/3 of coming from class 1 and prior probability 2/3 of coming from class 2. We can observe a single predictor variable x, and it is known that $x \sim N(-1, 2)$ in class 1 and $x \sim N(1, 2)$ in class 2.

Part a (Computation: 3 pts)

Use Bayes's Rule to find the posterior probability of an observation coming from class 2, given an observed value of x.

Part b (Computation and Explanation: 3 pts)

Suppose we fit a linear discriminant analysis (LDA) model on data from this population. Find the (true, not estimated) discriminant functions $\delta_1(x)$ and $\delta_2(x)$, and use them to find the Bayes decision boundary.

Part c (Computation and Explanation: 3 pts)

Suppose we fit a logistic regression model on data from these populations. Using your answer to part (a), find the coefficients β_0 and β_1 for the true population model relating x to the log-odds of being in population 2.

Part d (Computation and Explanation: 2 pts)

Show that the Bayes decision boundary for the logistic regression model is the same as for the LDA model.

Part e (Code/Computation and Explanation: 3 pts)

Suppose that a true positive occurs when the model correctly predicts an observation to come from population 2, and a true negative occurs when the model correctly predicts an observation to come from population 1. Find the true sensitivity and specificity of the model, using the Bayes decision boundary. Hint: you can use the `pnorm` function to obtain probabilities of being in the left tail of a normal distribution.

Part f (Code/Computation and Explanation: 2 pts)

Using the prior probabilities, find the probability that an unknown observation is a true positive, false negative, false positive, and true negative. Using these probabilities, obtain the (theoretical) Bayes error rate.

Applied Problem 1 (39 points total)

The `OChousing` dataset on Canvas contains data from the 2019 American Community Survey, aggregated by census tract. Each row represents one of the 583 ce

We would like to predict to median monthly rent in a tract, `med_rent` , using a combination of other variables.

Note that, like `med` in the `Boston` data set, `med_rent` is right-censored: there are 13 tracts in which the median rent was over $\$3500$, and all have a recorded `med`

Realistically, there is also some spatial correlation in the residuals that we would need to worry about when fitting a model, but since we haven't covered how to de

```
r #Setting directory and reading in setwd("E:/Academics/SPRING 2022/Midterm 2 Stuff") housing <- read.csv("OChousing.csv")
```

Part a (Code: 10 pts; Explanation: 10 pts)

Immediately split your (remaining) data into a training set (75-80% of the rows) and validation set (remaining 20-25%).

```r #Splitting into training and testing data sets set.seed(222) housing_split <- initial_split(housing, prop = 0.75)

# training and holdout set split train_housing <- training(housing_split) validation_housing <- testing(housing_split) ```

Perform a partial exploratory data analysis on the training set:

* Provide a thorough univariate analysis of the response variable, `med_rent` . You should be able to describe the center of the distribution, the variability of the distr

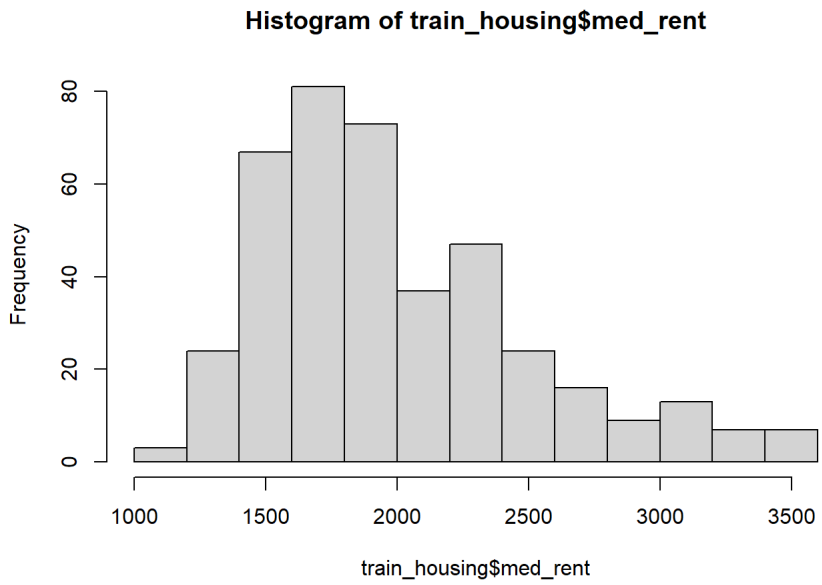
Explanation:

The variable “med_rent” is the response variable we wish to predict. It represents the median gross monthly rent paid by renters in the tract. The center of the distri

```r #center of med_rent variable center.med.rent <- mean(train_housing$med_rent)

#variability of med_rent variable var.med.rent <- var(train_housing$med_rent)

#shape of med_rent hist(train_housing$med_rent) ```




* Consider which possible predictor variables in the dataset *you* suspect influence the price of rent in a neighborhood. For three of these variables, provide a univar

The variables that I believe influence the price of rent the most are the average number of people in a household (ave_household), the median age of housing units

```r ## Variable 1: ave_household_size #center of med_rent variable center.med.rent <- mean(train_housing$avg_household_size)

#variability of med_rent variable var.med.rent <- var(train_housing$avg_household_size)

#shape of med_rent hist(train_housing$avg_household_size) ```


The first variable I chose was the average number of
```

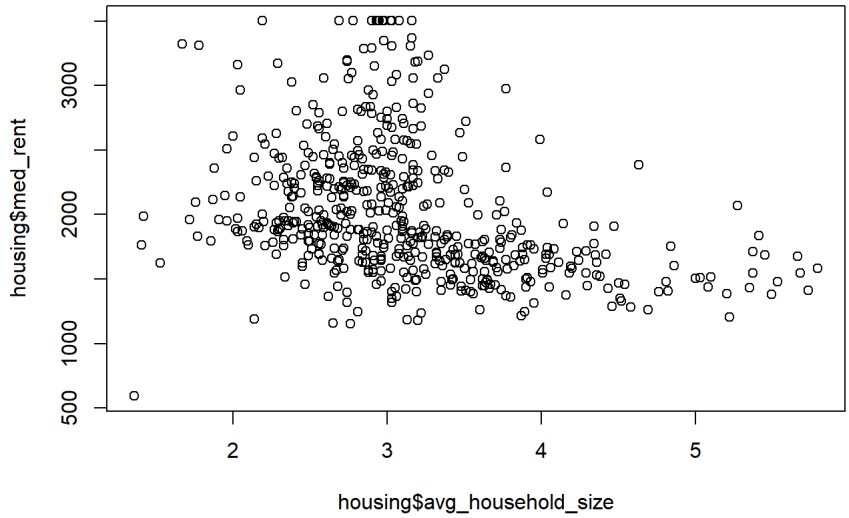
```r ## Variable 2: med_rooms #center of med_rooms variable center.med.rent <- mean(train_housing$med_rooms)

#variability of med_rooms variable var.med.rent <- var(train_housing$med_rooms)

#shape of med_rooms hist(train_housing$med_rooms) ```


The second variable that I chose was the median nun

```r ## Variable 3: med_income #center of med_income variable center.med.rent <- mean(train_housing$med_income)

#variability of med_income variable var.med.rent <- var(train_housing$med_income)

#shape of med_income hist(train_housing$med_income) ```


Lastly, the thirdd variable that I chose was the median income of residents in each tract. The center of this distribution is 93806.52. The variability of this distributior

* For these predictors, provide bivariate analyses of their relationships with `med_rent` and each other. You should be able to describe the direction, form, and stren

```r ## Variable 1: ave_household_size plot(housing$avg_household_size, housing$med_rent)```
```

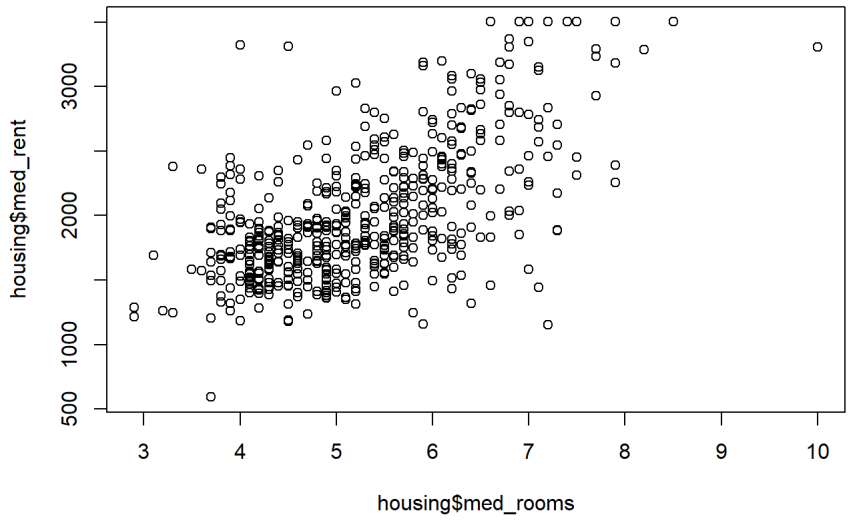



```
r cor(housing$avg_household_size, housing$med_rent)
```

```
## [1] -0.3467674
```

The relationship between the average number of people in a household and the median amount of rent appears to be a negative relationship l

```
r ## Variable 2: med_rooms plot(housing$med_rooms, housing$med_rent)
```

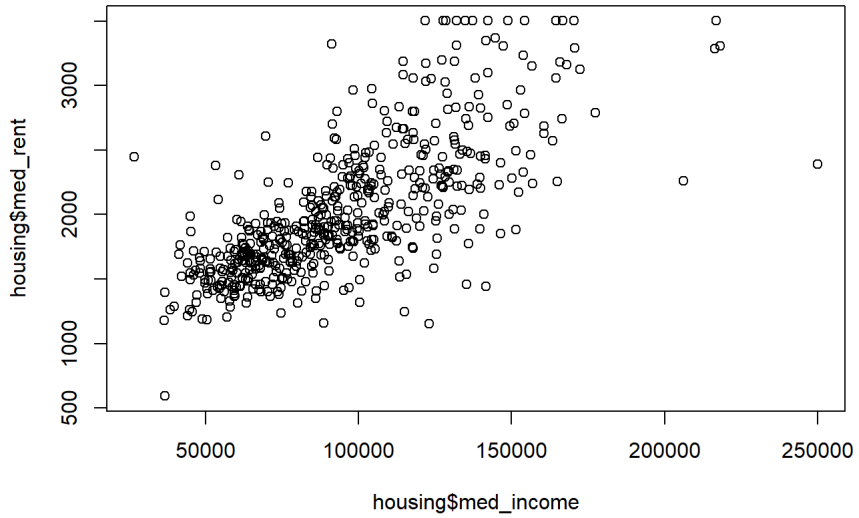



```
r cor(housing$med_rooms, housing$med_rent)
```

```
## [1] 0.622554
```

The relationship between the median amount of rooms in a household and the median amount of rent actually seems to be a positive one with relatively strong cor

```
r ## Variable 3: med_income plot(housing$med_income, housing$med_rent)
```




```
r cor(housing$med_income, housing$med_rent)
```

```
## [1] 0.7289554
```

Lastly, the relationship between median household income and the median amount of rent is highly correlated. There appears to be a clear linear relationship betw

* Each analysis (4 univariate, 6 bivariate) is worth **1 point for producing appropriate graphical and numerical summaries** and **1 point for documenting your t**

Part b (Code: 6 pts)

Use any model selection algorithm (best subset selection, forward selection, or backward selection), with Mallows' Cp as the selection criterion, to determine the be

```
r back.h <- regsubsets(med_rent~avg_household_size + med_age + med_rooms + pct_no_internet + poverty_rate + med_income, data = train_housing, nvm
```

```
## Subset selection object ## Call: regsubsets.formula(med_rent ~ avg_household_size + med_age + ##     med_rooms + pct_no_internet + poverty_rat
```

```
r back.sum$cp
```

```
## [1] 41.697127 19.865222 14.624635  5.293732  5.550041  7.000000
```

I chose to use the backwards selection as my model selection algorithm. After running

You may use `nvmax = 14` , or you may choose to ignore some predictors entirely. If you ignore a predictor, explain why you are ignoring it.

Part c (Code and Explanation: 9 pts)

Fit three of the following four models on the training set:

* A multiple linear regression model with a first-order (two-way) interaction term between two predictors of your choice. Justify your choice of interaction based on y

For the regular multiple regression models, you should use at least 2 predictors (but need not include more). For the penalized regression models, you should use

```
r ##linear regression model with a first order (two way) interaction term between med_rooms and avg_household_size lrm <- lm(med_rent ~ avg_house
```

```r ##ridge regression model using cv for optimal lambda # First standardize the numerical predictors

set.seed(222) train_housing[,!c(3,4,5,6,7,10,13,14)] <- scale(train_housing[,!c(3,4,5,6,7,10,13,14)])

# now create the model matrix and remove the intercept x.train <- model.matrix(med_rent~avg_household_size + med_age + med_rooms + pct_no_internet + pove

# finally do the cross-validation cv_ridge <- cv.glmnet(x = x.train, y = y.train, family = “gaussian”, alpha = 0) #alpha = 0 for ridge regression

#choosing lambda using 1 standard error rule l <- cv_ridge$lambda.1se #1 standard error rule lambda ridge.mod <- glmnet(x = x.train, y = y.train, alpha = 0, lambd
```

```r ##lasso model using cv for optimal lambda # now create the model matrix and remove the intercept x.train <- model.matrix(med_rent~avg_household_size + m

# finally do the cross-validation cv_lasso <- cv.glmnet(x = x.train, y = y.train, family = “gaussian”, alpha = 1) #alpha = 0 for ridge regression

#choosing lambda using 1 standard error rule lam <- cv_ridge$lambda.1se #1 standard error rule lambda lasso.mod <- glmnet(x = x.train, y = y.train, alpha = 1, lam
```

Part d (Code and Explanation: 4 pts)

Determine which of your four models (your “best” model in part b and your three models in part c) has the lowest MSE on the validation set. Then, fit that model usi

```
r #MSE for First Model (Backward Selection) back.h <- lm(med_rent~avg_household_size + med_age + med_rooms + med_income, data = train_housing) fi
```

```
## [1] 117642.7
```

```
r #MSE for Multiple Linear Regression Model mlr.sum <- summary(lrm) mse.mlr <- mean((mlr.sum$residuals)^2) mse.mlr
```

```
## [1] 146791.9
```

```
r #MSE for Ridge Regression Model ridge.sum <- summary(ridge.mod) newX <- model.matrix(med_rent~avg_household_size + med_age + med_rooms + pct_nc
```

```
## [1] 151758.5
```

```
r #MSE for LASSO Model lasso.sum <- summary(lasso.mod) lasoo.pred <- predict(lasso.mod, s = lam, newx = newX, type = "response") mse.lasso <- mea
```

```
## [1] 151758.5
```

Based on the test MSE, the model we would choose would be the first model (backward selection). So now we will fit a model with the entire dat

```
r back.h <- lm(med_rent~avg_household_size + med_age + med_rooms + med_income, data = housing) summary(back.h)
```

```
## ## Call: ## lm(formula = med_rent ~ avg_household_size + med_age + med_rooms + ##     med_income, data = housing) ## ## Residuals: ##      Mir
```

med_rent = 1.354e+03 + avg_household_size(-7.385e+01) + med_age(-8.420e+00) + med_rooms(1.225e+02) + med_income(6.729e-03)

Applied Problem 2 (40 points total)

Public health officials are concerned about the high consumption of sugar-sweetened beverages (sodas, energy drinks, sweet tea, etc.). Researchers would like to find effective alternatives to local laws prohibiting the sale of these beverages. One promising alternative, adapted from strategies to combat tobacco use, is to include a warning label on the beverages.

At the UNC Mini-Mart in Chapel Hill, North Carolina, eligible customers filled out a brief survey about their attitude toward sugary drinks and then selected one drink for their child as part of their survey compensation. Unknown to them, researchers had added either a graphic warning label about the dangers of sugary drinks or an unusually large bar code to all of the sugar-sweetened beverages for sale.

We would like to predict which type of beverage a parent will buy (`purchase`), based on the parent's attitude toward sugar-sweetened beverages and the type of label the parent sees. Please see the `ssb_dictionary` file on Canvas for an explanation of each of the variables.

```
#Setting directory and reading in
setwd("E:/Academics/SPRING 2022/Midterm 2 Stuff")
ssb <- read.csv("ssb.csv")
```

Part a (Code: 8 pts; Explanation: 12 pts)

Immediately split your (remaining) data into a training set (75-80% of the rows) and validation set (remaining 20-25%).

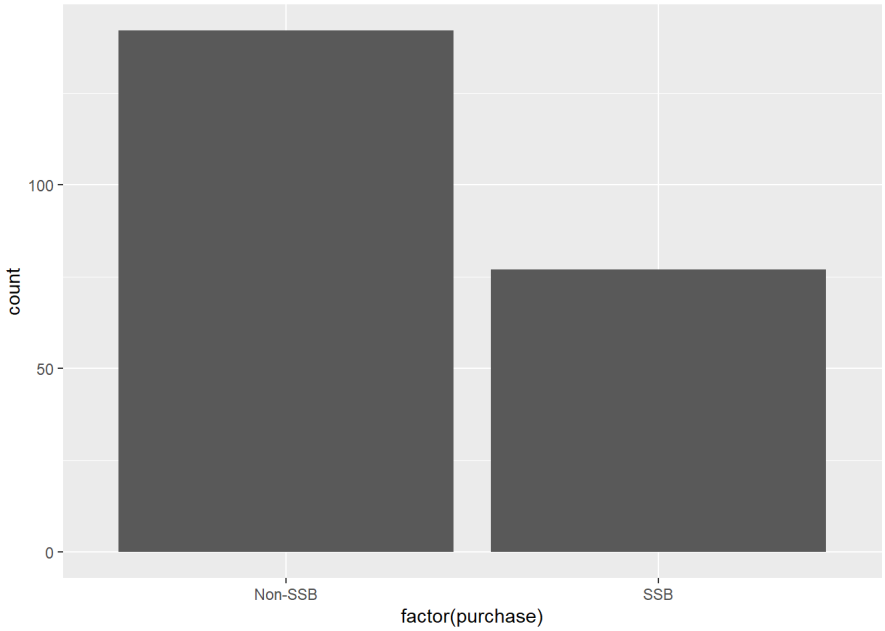
```
#Splitting into training and testing data sets
set.seed(222)
ssb_split <- initial_split(ssb, prop = 0.75)

# training and holdout set split
train_ssb <- training(ssb_split)
validation_ssb<- testing(ssb_split)
```

Perform a partial exploratory data analysis on the training set:

- Provide a univariate analysis of the response variable, `purchase` , and the primary predictor of interest, `picture_label` .

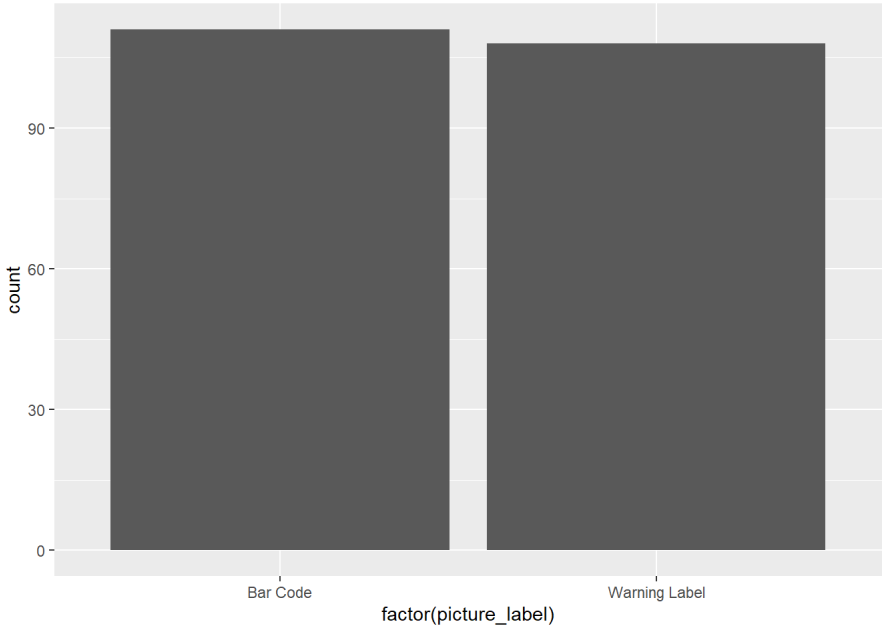
The purchase variable is a categorical response variable which represents whether or not a parent decided to purchase the sugar-sweetened beverage (SSB) or not (non-SSB). The picture_label variable is also a categorical varaible but it is our main predictor of interest. It represents whether the parent saw a Warning Label or a Bar Code picture on the sugar-sweetened beverage bottles which contains one of two outputs. Either the beverage had the warning label (Warning Label) or just a bar code without a warning label (Bar Code).

```
#For Purchase Variable
ggplot(train_ssb, aes(x = factor(purchase))) +
    geom_bar()
```



As you can see from the bar plot above for the purchase variable, we see that for the majority of the time (in the training set of the data) the parents ultimately decided to not purchase the sugar-sweetened beverage.

```
#For picture_label Variable
ggplot(train_ssb, aes(x = factor(picture_label))) +
    geom_bar()
```

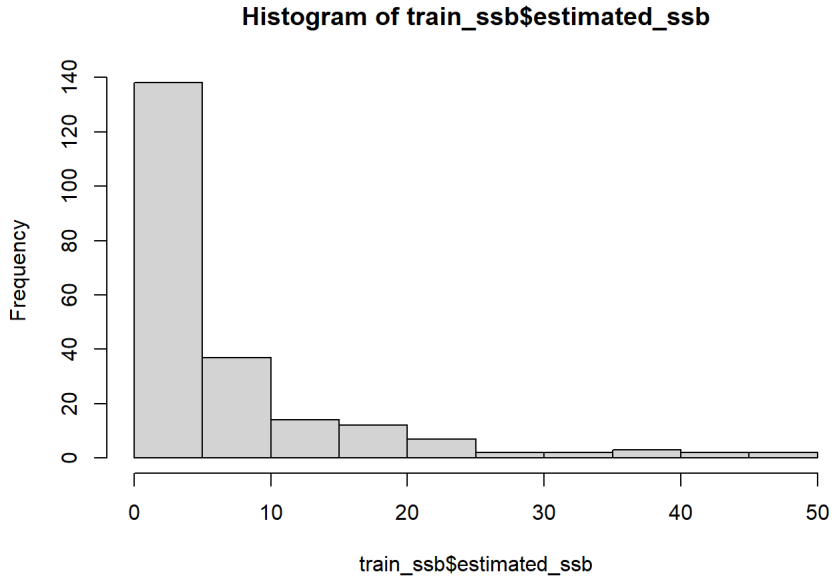


Looking at the bar plot above for the purchase_label varaible, we see that the about half the time the beverage included a warning level for the amount of sugar in the beverage.

- Consider which other possible predictor variables in the dataset might influence a parent's decision to purchase a sugar-sweetened beverage (or not) for their child. For three of these variables, provide a univariate analysis of the variable.

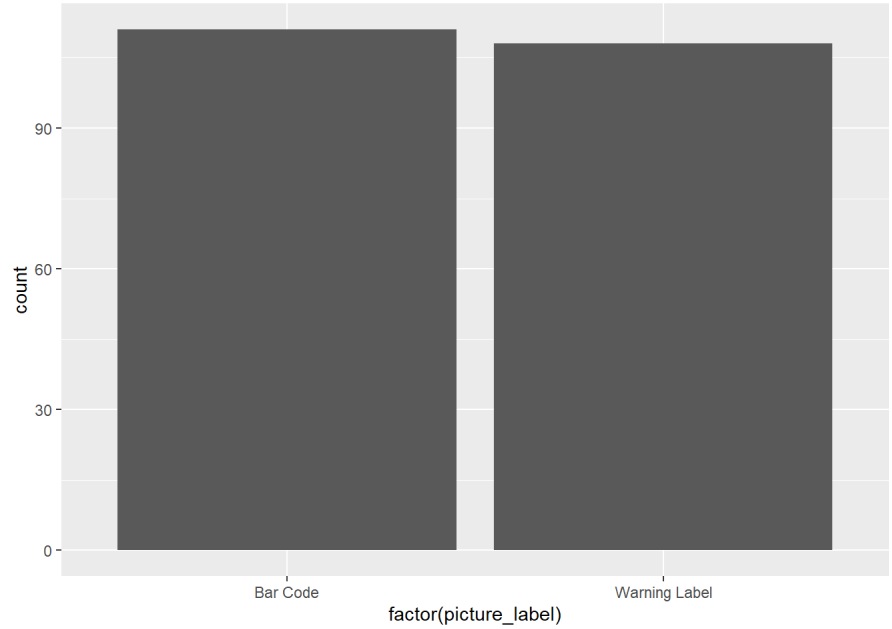
The main variables that I believe have the greatest influence on whether or not a parent decides to purchase a sugar sweetened beverage is the parent-estimated number of sugar-sweetened beverages consumed by the child in a typical week (estimated_ssb), the child's age (child_age), whether or not the beverage contains a warning label for the amount of sugar (picture_label), strongly the parent believed that daily intake of added sugar would increase the child's risk of heart disease, diabetes, or other health problems where 1 = not at all and 5 = a lot (increased_risk) and lastly, how strongly the parent believed that sugar-sweetened beverages should have a warning label where 1 = not at all and 5 = a lot (supporting_label).

```
#Variable 1: estimated_ssb
#center of med_rent variable
center.estimated.ssb <- mean(train_ssb$estimated_ssb)

#variability of med_rent variable
var.estimated.ssb <- var(train_ssb$estimated_ssb)

#shape of med_rent
hist(train_ssb$estimated_ssb)
```



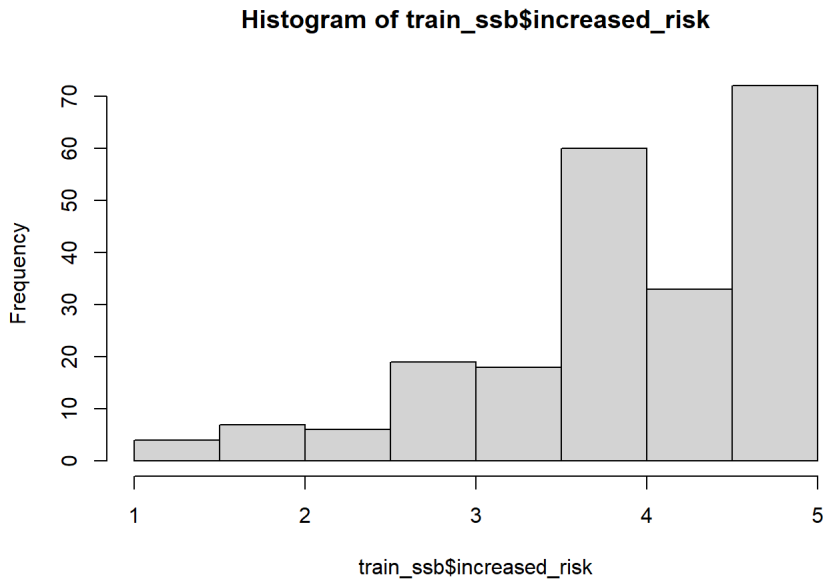
For the estimated_ssb variable, the center of its distribution is 4.004566. The variability of the distribution is 0.8353511. And lastly, looking at the shape of the distribution we can see that there is a significant skew to the right, leading us to believe that there is likely outliers on the high end of the dataset.

```
#Variable 2: picture_label
ggplot(train_ssb, aes(x = factor(picture_label))) +
    geom_bar()
```



We have already looked at the graph for the picture_label variable. We still see that it seems evenly split between the amount of beverages that included the warning label and those that didn't.

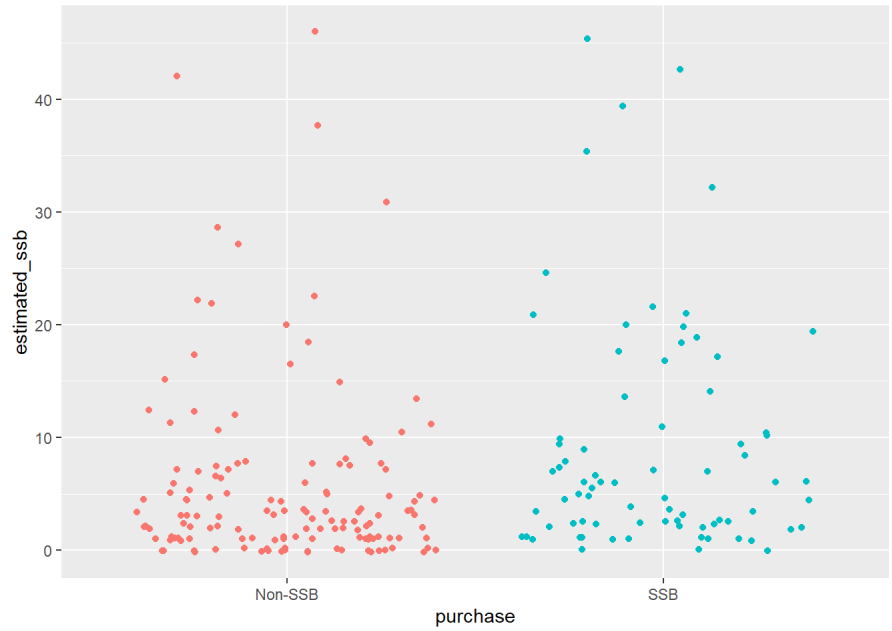
```
#Variable 3: increased_risk
#center of increased_risk variable
center.estimated.ssb <- mean(train_ssb$increased_risk)

#variability of increased_risk variable
var.estimated.ssb <- var(train_ssb$increased_risk)

#shape of increased_risk
hist(train_ssb$increased_risk)
```



Lastly, for the increased_risk variable, we have that the center of the distribution was 4.004566. As for the variability, the distribution had variability of 0.8353511. Lastly, looking at the shape of the distribution, we see that the data seems to be heavily skewed to the left. This leaves the great possibility of outliers towards the left side of the data.

- For each of the four predictors (`picture_label` and three of your choice), provide bivariate analyses of their relationships with `purchase` .

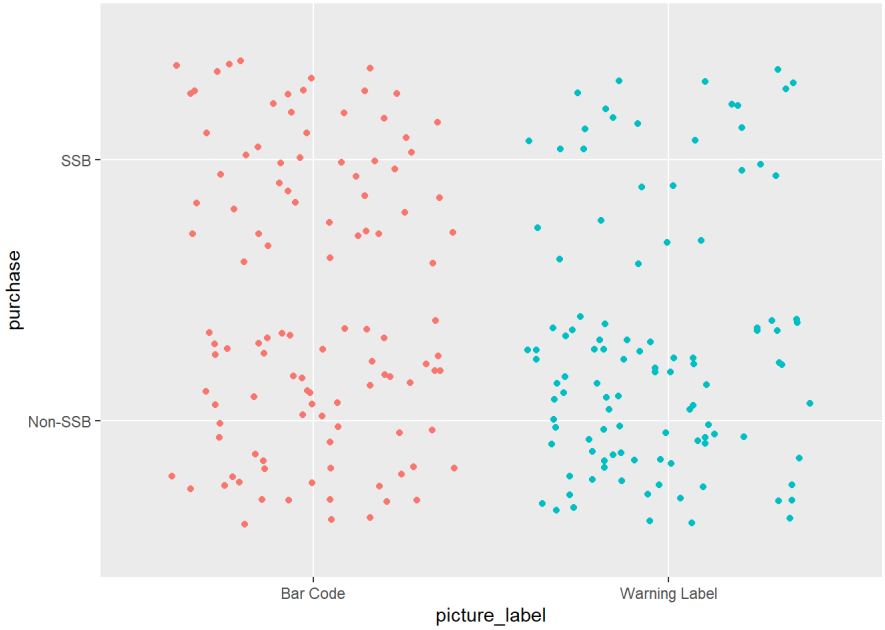
```
#Variable 1: estimated_ssb
ggplot(data = train_ssb) +
  aes(x = purchase, y = estimated_ssb, color = purchase) +
  geom_jitter() +
  theme(legend.position = "none")
```



In the graph above, we are looking at the relationship between the estimated_ssb variable and our response variable purchase. We see the pattern that the less amount of estimated_ssb we have, the less likely a parent is to purchase a sugar_sweetened beverage. This conclusion is supported by the fact that the

bottom left of the graph appears to be more cluttered than the bottom right of the graph above. This means that when the parent-estimated number of sugar-sweetened beverages consumed by the child in a typical week is small, the chance of a parent purchasing a sugar_sweetened beverage for their child is small as well.

```
#Variable 2: picture_label
ggplot(data = train_ssb) +
  aes(x = picture_label, y = purchase, color = picture_label) +
  geom_jitter() +
  theme(legend.position = "none")
```



In the second graph, we are looking at the relationship between picture_label and purchase. After looking at the graph above, there is a possible pattern between the variables. It seems that if a beverage contains the warning label, more often than not, the parent decided not to purchase the sugar_sweetened beverage. Similarly, when the product only contained the bar code and not the warning label, parents seemed to be more willing to purchase the sugar-sweetened beverage. This conclusion is supported by the fact that the bottom right seems more cluttered than the top right and the top left is also more cluttered than the top right in the graph. This means that when the beverage contains a warning label for the amount of sugar in a beverage, the less likely a parent is to purchase the sugar_sweetened beverage for their child.

```
#Variable 3: increased_risk
ggplot(data = train_ssb) +
  aes(x = purchase, y = increased_risk, color = purchase) +
  geom_jitter() +
  theme(legend.position = "none")
```

Looking at the relationship between increased_risk and purchase in the graph above, there seems to be a pattern. When the increased_risk was higher, parents were less likely to purchase the sugar_sweetened beverage. We see this by the fact that the top left is more cluttered than the top right of the graph. This means that

the higher of a risk the parent believes that daily intake of added sugar would increase the child's risk of heart disease, diabetes, or other health problems the less likely a parent is to purchase the sugar_sweetened beverage.

- For each of your nine analyses (5 univariate, 4 bivariate), make sure to **produce appropriate numerical and graphical summaries** and **document your thought process as you choose your predictors and interpret your results**.

Part b (Code: 12 pts; Explanation: 4 pts)

Fit four of the following five generative models for classification on the training set. For each model you choose, briefly explain the assumptions you are making about the predictor variables and/or their relationship with the response.

- A logistic regression model using the `glm` function. Use 10-fold cross-validation to choose from at least two different subsets of predictors.
- A logistic regression model using the `glmnet` function (still including the `family = "binomial"` argument) to incorporate ridge regression, LASSO, or elastic net. Use 10-fold cross-validation to decide what value of λ to use.
- A naive Bayes model. Use 10-fold cross-validation to choose from among at least two different subsets of predictors.
- A linear discriminant analysis (LDA) model. Use 10-fold cross-validation to choose from among at least two different subsets of predictors.
- A quadratic discriminant analysis (QDA) model. Use 10-fold cross-validation to choose from among at least two different subsets of predictors.

```
#Logistic Regression Model using glm() function:

ctrl <- trainControl(method = "cv", number = 10)
glm_fit1 <- train(purchase ~ estimated_ssb + child_age + picture_label + increased_risk + support_labeling, data = train_ss
b, method = "glm", trControl=ctrl, tuneLength = 0)
glm_fit1
```

```
## Generalized Linear Model
##
## 219 samples
##   5 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 197, 198, 197, 197, 197, 196, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6390646  0.08438218
```

```
glm.acc1 <- 0.6301713

glm_fit2 <- train(purchase ~ estimated_ssb + child_age + picture_label + increased_risk, data = train_ssb, method = "glm", t
rControl=ctrl, tuneLength = 0)
glm_fit2
```

```
## Generalized Linear Model
##
## 219 samples
##   4 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 198, 198, 197, 197, 197, 198, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6525127  0.09590152
```

```
glm.acc2 <- 0.647939


glm_fit3 <- train(purchase ~ estimated_ssb + child_age + picture_label, data = train_ssb, method = "glm", trControl=ctrl, tu
neLength = 0)
glm_fit3
```

```
## Generalized Linear Model
##
## 219 samples
##   3 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 198, 197, 197, 198, 197, 196, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6533126  0.08229018
```

```
glm.acc3 <- 0.662667 #best model subset
```

```
#Naive Bayes
nb_fit1 <- train(purchase ~ estimated_ssb + child_age + picture_label + increased_risk + support_labeling, data = train_ssb,
method = "naive_bayes", trControl=ctrl, tuneLength = 0)
nb_fit1
```

```
## Naive Bayes
##
## 219 samples
##   5 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 197, 198, 196, 197, 196, 197, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.6262846  0.08661243
##    TRUE      0.6268963  0.07137383
##
## Tuning parameter 'laplace' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were laplace = 0, usekernel = TRUE
##  and adjust = 1.
```

```
acc1t <- 0.6163467
acc1f <- 0.5964897

nb_fit2 <- train(purchase ~ estimated_ssb + child_age + picture_label + increased_risk, data = train_ssb, method = "naive_ba
yes", trControl=ctrl, tuneLength = 0)
nb_fit2
```

```
## Naive Bayes
##
## 219 samples
##   4 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 197, 196, 197, 197, 197, 198, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.6578581  0.13290381
##    TRUE      0.6351120  0.09743944
##
## Tuning parameter 'laplace' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were laplace = 0, usekernel = FALSE
##  and adjust = 1.
```

```
acc2t <- 0.6573123
acc2f <- 0.6434594 #best model subset


nb_fit3 <- train(purchase ~ estimated_ssb + child_age + picture_label, data = train_ssb, method = "naive_bayes", trControl=c
trl, tuneLength = 0)
nb_fit3
```

```
## Naive Bayes
##
## 219 samples
##   3 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 198, 197, 198, 196, 197, 196, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy  Kappa
##   FALSE      0.648419  0.07676647
##    TRUE      0.620911  0.06024533
##
## Tuning parameter 'laplace' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were laplace = 0, usekernel = FALSE
##  and adjust = 1.
```

```
acc3t <- 0.6030585
acc3f <- 0.6481178
```

```
#LDA
lda_fit1 <- train(purchase ~ estimated_ssb + child_age + picture_label + increased_risk + support_labeling, data = train_ss
b, method = "lda", trControl=ctrl, tuneLength = 0)
lda_fit1
```

```
## Linear Discriminant Analysis
##
## 219 samples
##   5 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 198, 197, 197, 197, 197, 196, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6345379  0.08085921
```

```
lda.acc1 <- 0.640288

lda_fit2 <- train(purchase ~ estimated_ssb + child_age + picture_label + increased_risk, data = train_ssb, method = "lda", t
rControl=ctrl, tuneLength = 0)
lda_fit2
```

```
## Linear Discriminant Analysis
##
## 219 samples
##   4 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 198, 196, 197, 196, 197, 198, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6262658  0.02252307
```

```
lda.acc2 <- 0.6255411


lda_fit3 <- train(purchase ~ estimated_ssb + child_age + picture_label, data = train_ssb, method = "lda", trControl=ctrl, tu
neLength = 0)
lda_fit3
```

```
## Linear Discriminant Analysis
##
## 219 samples
##   3 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 197, 198, 197, 197, 196, 198, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6517222  0.08594724
```

```
lda.acc3 <- 0.6492283 #best model subset
```

```
#QDA
qda_fit1 <- train(purchase ~ estimated_ssb + child_age + picture_label + increased_risk + support_labeling, data = train_ss
b, method = "qda", trControl=ctrl, tuneLength = 0)
qda_fit1
```

```
## Quadratic Discriminant Analysis
##
## 219 samples
##   5 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 197, 197, 197, 198, 197, 198, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.603275  0.01449177
```

```
qda.acc1 <- 0.6262658

qda_fit2 <- train(purchase ~ estimated_ssb + child_age + picture_label + increased_risk, data = train_ssb, method = "qda", t
rControl=ctrl, tuneLength = 0)
qda_fit2
```

```
## Quadratic Discriminant Analysis
##
## 219 samples
##   4 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 198, 197, 197, 196, 197, 197, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6124318  0.005775853
```

```
qda.acc2 <- 0.6171278


qda_fit3 <- train(purchase ~ estimated_ssb + child_age + picture_label, data = train_ssb, method = "qda", trControl=ctrl, tu
neLength = 0)
qda_fit3
```

```
## Quadratic Discriminant Analysis
##
## 219 samples
##   3 predictor
##   2 classes: 'Non-SSB', 'SSB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 197, 197, 197, 197, 196, 197, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6481555  0.06710741
```

```
qda.acc3 <- 0.6492471 #best model subset
```

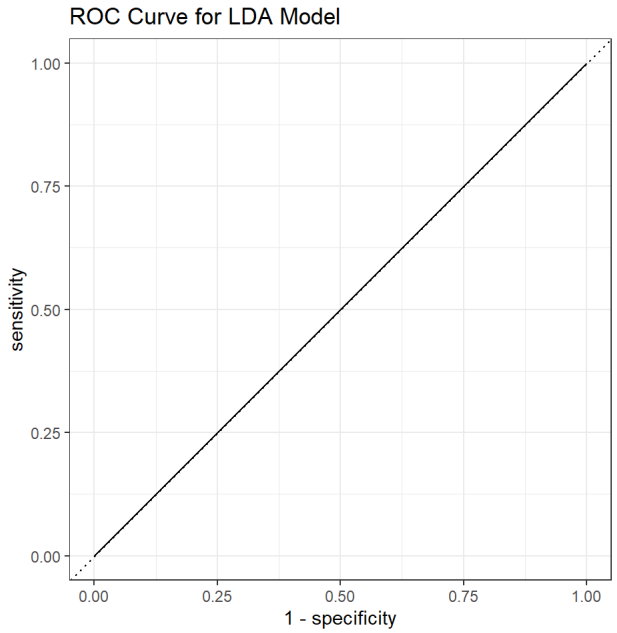
Part c (Code and Explanation: 4 pts)

Which of the four models from part (b) would you select as the “best” model for predicting new observations? Use one of the following accuracy metrics, evaluated on the validation set, to justify your decision:

- Matthews Correlation Coefficient (mcc) at the estimated Bayes decision boundary
- Area Under Curve (AUC) for the Receiver Operating Characteristic (ROC) Curve
- Brier Score

For this section we will be evaluating the models using the AUC as out accuracy metric.

```
#Logistic Regression model using glm() function:
glm.mod <- glm(as.factor(purchase) ~ estimated_ssb + child_age + picture_label, data = train_ssb, family = "binomial")
validation_predictions <- predict(glm.mod, validation_ssb, type = "response")

validation_all <- tibble(predicted = as.numeric(validation_predictions[[1]]), actual = validation_ssb$purchase)

roc_tibble <- roc_curve(validation_all, truth = as.factor(actual), predicted, event_level = "second")

autoplot(roc_tibble) + labs(title = "ROC Curve for LDA Model")
```

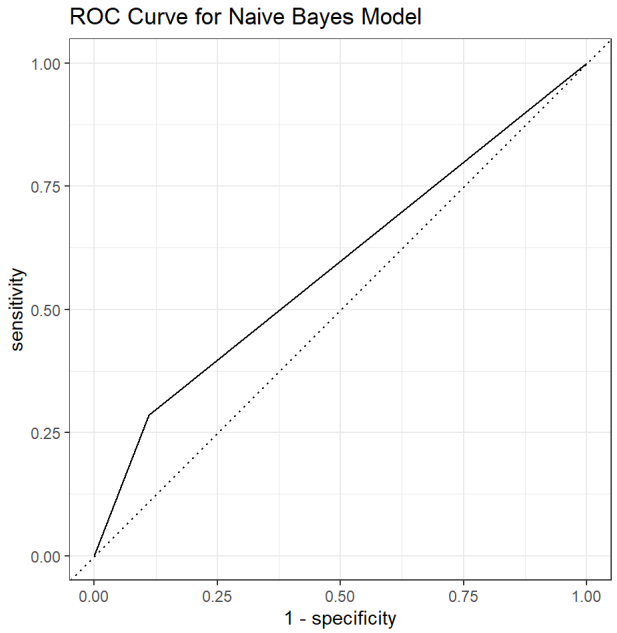



```
roc_auc(validation_all, truth = as.factor(actual), predicted, event_level = "second")
```

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
roc_auc	binary	0.5

1 row

```
#Naive Bayes:
nb.mod <- naiveBayes(purchase ~ estimated_ssb + child_age + picture_label + increased_risk, data = train_ssb)
nb.pred <- predict(nb.mod,validation_ssb, type = "class")

validation_all <- tibble(predicted = as.numeric(nb.pred), actual = validation_ssb$purchase)

roc_tibble <- roc_curve(validation_all, truth = as.factor(actual), predicted, event_level = "second")

autoplot(roc_tibble) + labs(title = "ROC Curve for Naive Bayes Model")
```

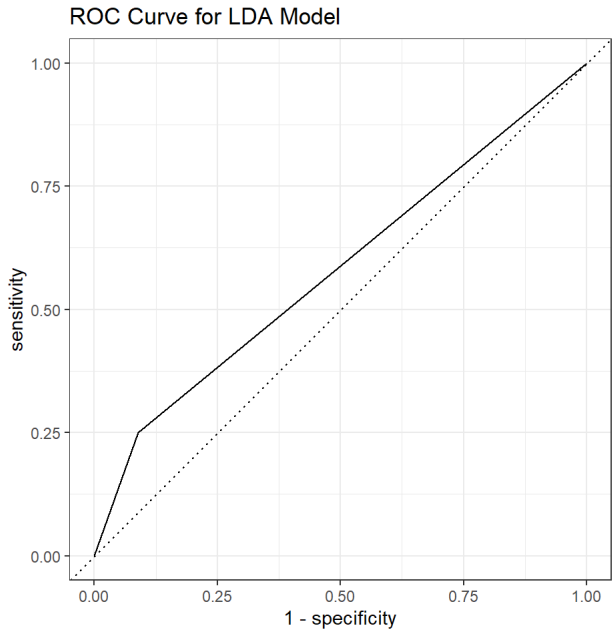



```
roc_auc(validation_all, truth = as.factor(actual), predicted, event_level = "second")
```

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
roc_auc	binary	0.5873016

1 row

```
#LDA:
lda.mod <- lda(purchase ~ estimated_ssb + child_age + picture_label, data = train_ssb)
validation_predictions <- predict(lda.mod, validation_ssb, type = "class")

validation_all <- tibble(predicted = as.numeric(validation_predictions[[1]]), actual = validation_ssb$purchase)

roc_tibble <- roc_curve(validation_all, truth = as.factor(actual), predicted, event_level = "second")

autoplot(roc_tibble) + labs(title = "ROC Curve for LDA Model")
```

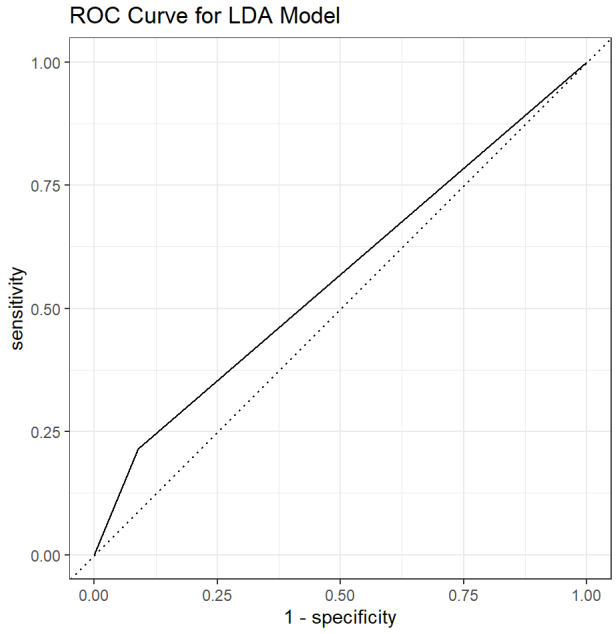



```
roc_auc(validation_all, truth = as.factor(actual), predicted, event_level = "second")
```

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
roc_auc	binary	0.5805556

1 row

```
#QDA:
qda.mod <- qda(purchase ~ estimated_ssb + child_age + picture_label, data = train_ssb)
validation_predictions <- predict(qda.mod, validation_ssb, type = "class")

validation_all <- tibble(predicted = as.numeric(validation_predictions[[1]]), actual = validation_ssb$purchase)

roc_tibble <- roc_curve(validation_all, truth = as.factor(actual), predicted, event_level = "second")

autoplot(roc_tibble) + labs(title = "ROC Curve for LDA Model")
```




```
roc_auc(validation_all, truth = as.factor(actual), predicted, event_level = "second")
```

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
roc_auc	binary	0.5626984

1 row

The auc for the logistic regression model was 0.5. The auc for the naive bayes model was 0.5873016. The auc for the LDA model was 0.5805556. The auc for the QDA model was 0.5626984.

The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. Therefore the model that performed the best based on the AUC of the ROC was the Niave Bayes model.