Comparing Dress Classifiers

Supriya Shingade
sshingad

Due Wed, Apr 28, at 8:00PM (Pittsburgh time)

Contents

```
set.seed(151)
library("knitr")
library("kableExtra")
library("pander")
library("readr")
library("magrittr")
library("car")
library("MASS")
library("klaR")
library("tree")
library("rpart")
library("rpart.plot")
```

Introduction

The constant changing environment of fashion and the large retail industry built around clothes, it is important to find what factors are associated with positive results and successful sales of clothes. If retailers are able to find out what qualities in a dress make it a successful sell, they will be able to employ the same techniques or look for pieces of garments that have these qualities to help sell the clothing. This way they will also able able to increase overall profits and get ahead of their competition. Thus, exploring what factors are associated with dresses that are sold successfully is worthwhile.

Exploratory Data Analysis

A clothing store dress sales have not been very successful. Although some styles sold out very fast, others were still there after sales events. The store wants to know which dresses they should order for next year so they have more successful sales. In the file dresses_train.csv, you have available data from last year's sales to train your classifiers, and you should use the data available in dresses_test.csv to verify your classifiers' accuracy and pick the best model to be used next year to decide if a dress with pre-determined characteristics should be included in next year's collection. *Background and Variables* The sample of inforamtion was from a clothing store to help determine how to increase the success of dress sales for all items overall. The information that was collected were about the following variables:

- `Style`: dress style (cute, work, casual, fashion, party)

- `Price`: price range (low, average, high)
- `Rating`: average customer rating from dress factory market survey (average of stars, 0-5)
- 'Season: which season is the dress appropriate for (summer, fall, winter, spring)
- `NeckLine`: type of neckline (O-neck, V-neck, other)
- `Material`: if it is a cotton dress or not
- `Decoration`: if it has any decoration or not
- `Pattern`: if the fabric has a pattern (yes) or of it's a solid color (no)
- `Sleeve`: if the dress has a sleeve
- `Waistline`: type of waistline (other, empire, natural)

The response variable to be predicted:

- `Recommendation`: binary outcome if the dress sells well (success) (1) or not (failure) (0).

```
dress_train <- readr::read_csv("http://stat.cmu.edu/~gordonw/dress_train.csv")
dress_test <- readr::read_csv("http://stat.cmu.edu/~gordonw/dress_test.csv")
```

Summary of the Response Labels in the Training Dataset

There are a total of 347 observations with 189 (54.47%) observations of dresses that were not successfully sold (ie failures) and 158 (45.53%) there were successfully sold.

```
table(dress_train$Recommendation)
```

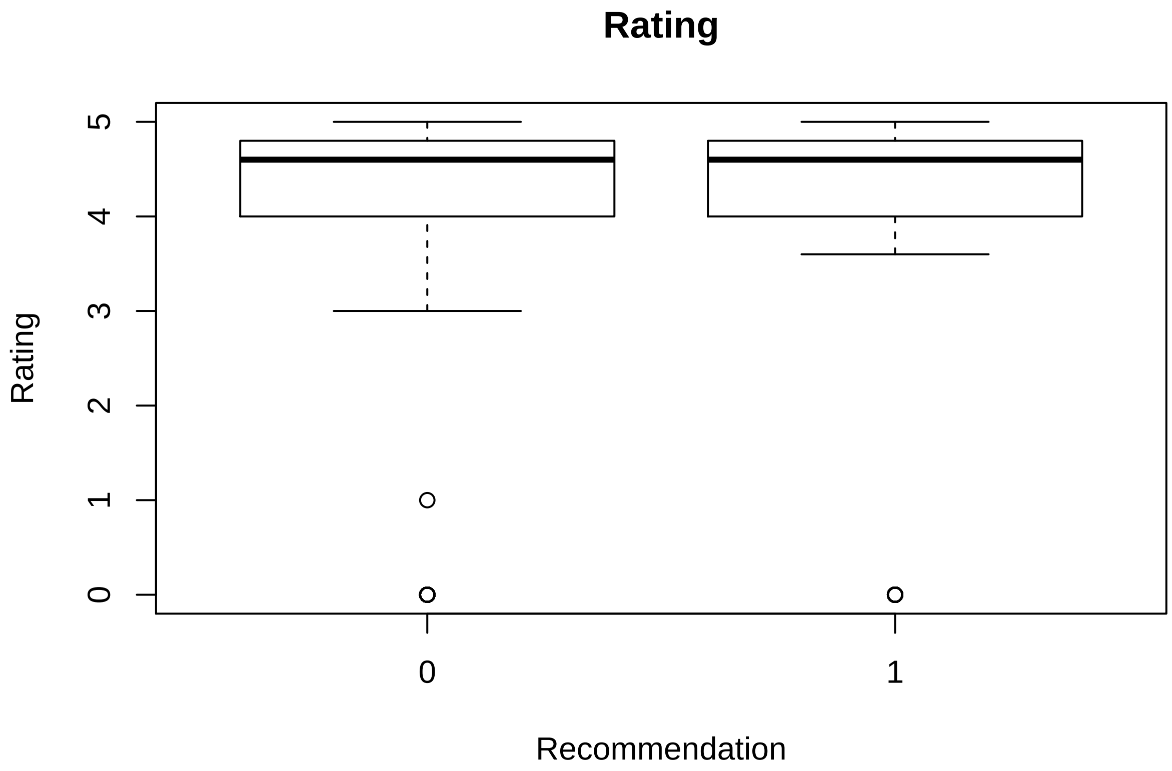
```
##
##   0   1
## 189 158
```

```
prop.table(table(dress_train$Recommendation))
```

```
##
##         0         1
## 0.5446686 0.4553314
```

EDA of the quantitative predictors The only quantitative predictor we have in our variable set is the Ratings Variable We then move toward visualizing the relationship between the response (type) and the various predictors (the physiochemical tests and the wine quality). For visually exploring whether we expect the quantitative predictors to be useful in helping to classify the wine type, we show boxplots, which appear as follows:

```
boxplot(Rating ~ Recommendation,
  main="Rating",
  data = dress_train)
```



```
prop.table(
table(dress_train$Recommendation, dress_train$Style),
margin = 2)
```

```
##
##        casual      cute   fashion     party      work
##   0 0.5757576 0.5048544 0.6363636 0.3030303 0.7500000
##   1 0.4242424 0.4951456 0.3636364 0.6969697 0.2500000
```

```
prop.table(
table(dress_train$Recommendation, dress_train$Price),
margin = 2)
```

```
##
##       average      high       low
##   0 0.5729730 0.3750000 0.5461538
##   1 0.4270270 0.6250000 0.4538462
```

```
prop.table(
table(dress_train$Recommendation, dress_train$Season),
margin = 2)
```

```
##
##          fall    spring    summer    winter
##   0 0.6818182 0.3483146 0.6194690 0.5742574
##   1 0.3181818 0.6516854 0.3805310 0.4257426
```

```r
prop.table(
table(dress_train$Recommendation, dress_train$NeckLine),
margin = 2)
```

```
##
##         oneck     other     vneck
##   0 0.5537634 0.5443038 0.5243902
##   1 0.4462366 0.4556962 0.4756098
```

```r
prop.table(
table(dress_train$Recommendation, dress_train$Material),
margin = 2)
```

```
##
##        cotton     other
##   0 0.5504587 0.5420168
##   1 0.4495413 0.4579832
```

```r
prop.table(
table(dress_train$Recommendation, dress_train$Decoration),
margin = 2)
```

```
##
##            no       yes
##   0 0.5562500 0.5347594
##   1 0.4437500 0.4652406
```

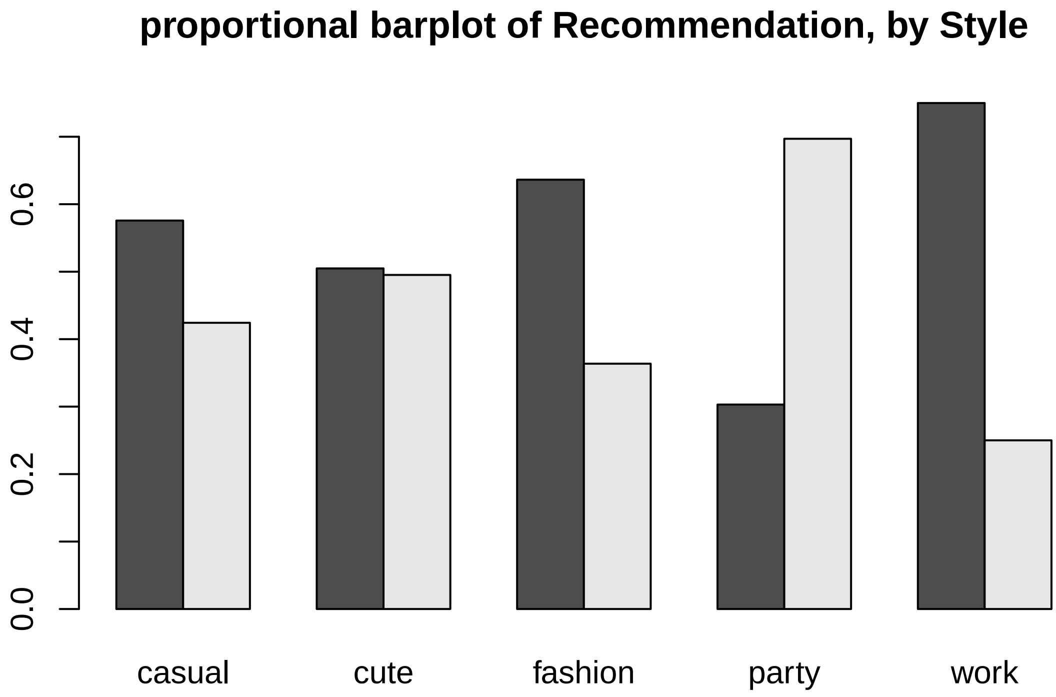
```r
prop.table(
table(dress_train$Recommendation, dress_train$Pattern),
margin = 2)
```

```
##
##            no       yes
##   0 0.5229358 0.5813953
##   1 0.4770642 0.4186047
```

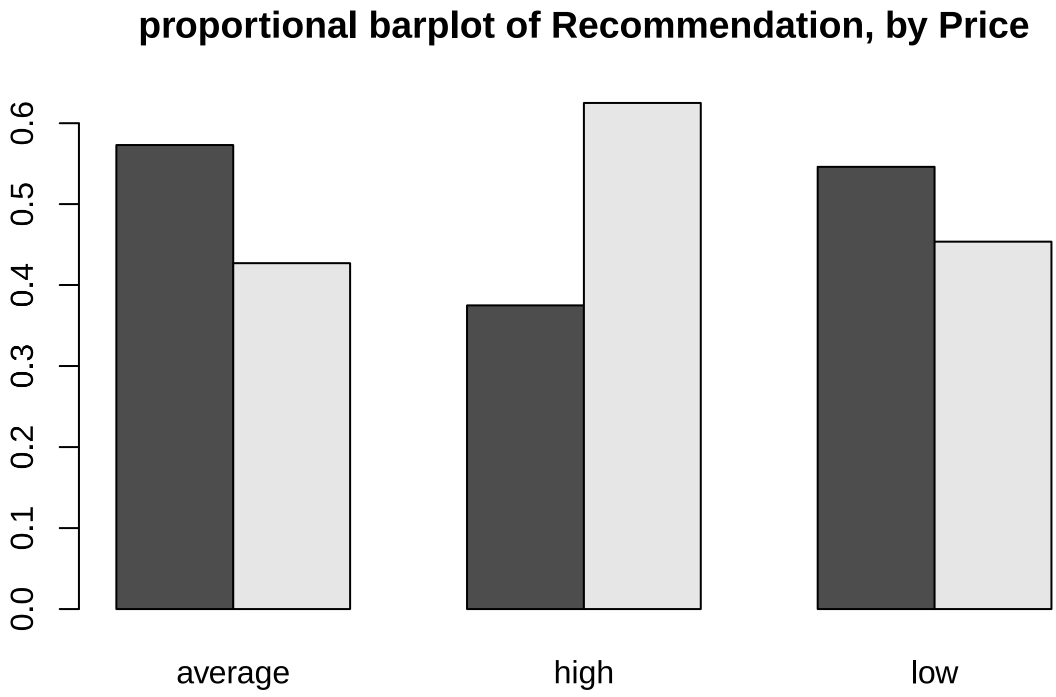
```r
prop.table(
table(dress_train$Recommendation, dress_train$Waistline),
margin = 2)
```

```
##
##        empire   natural     other
##   0 0.4722222 0.5769231 0.5223881
##   1 0.5277778 0.4230769 0.4776119
```

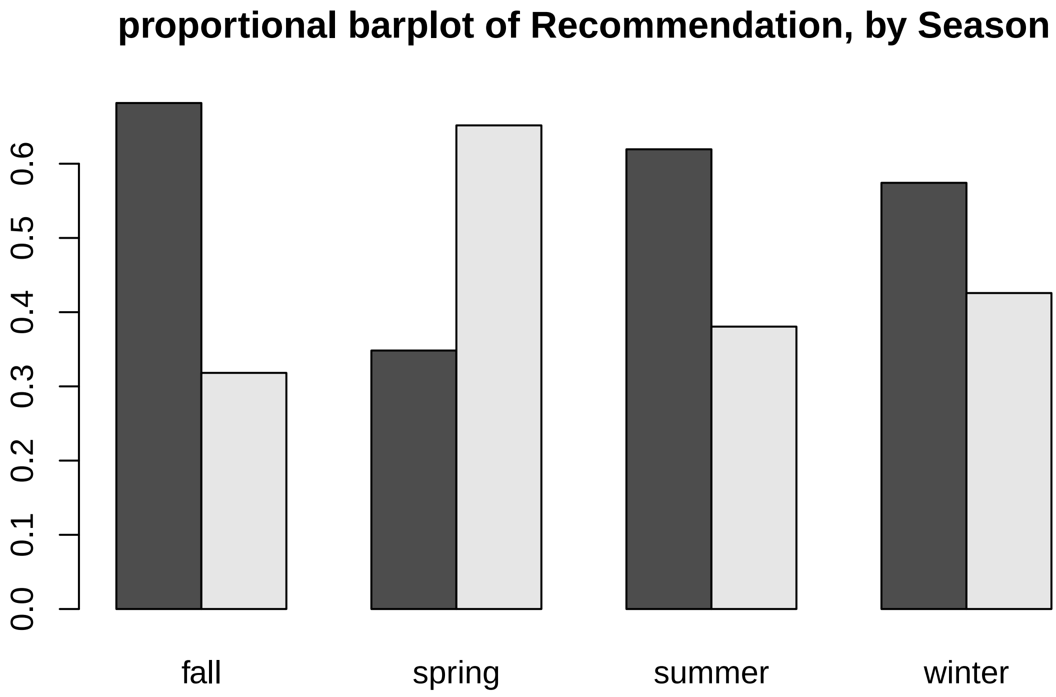
```r
barplot(
prop.table(
table(dress_train$Recommendation, dress_train$Style),
margin = 2)
, beside = TRUE,
main = "proportional barplot of Recommendation, by Style")
```



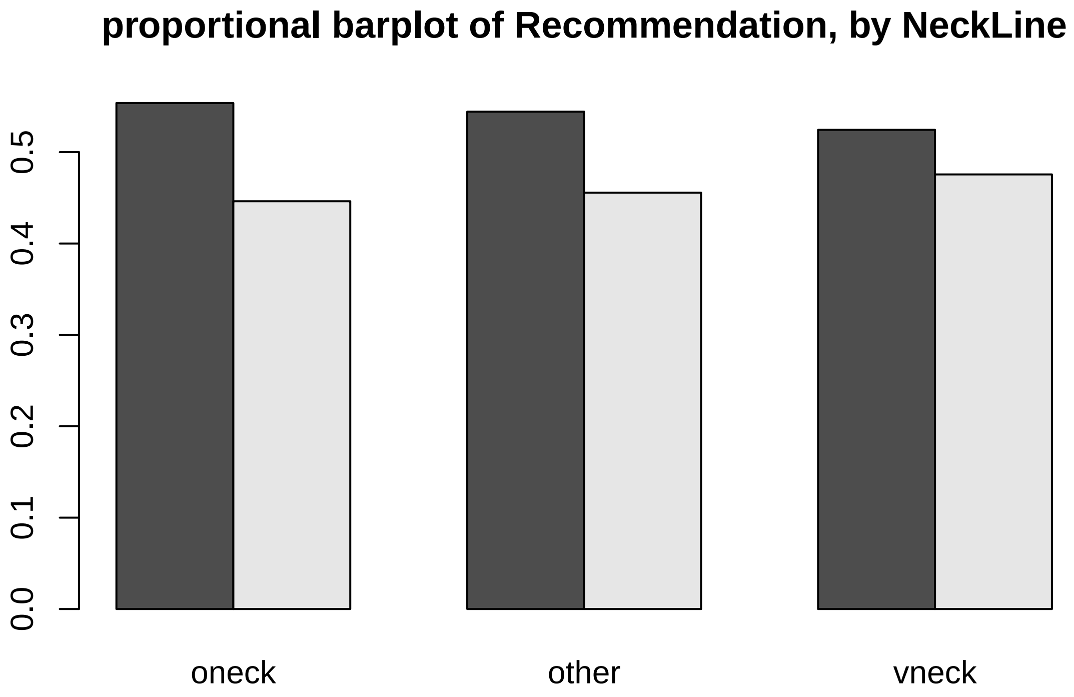
```
barplot(
prop.table(
table(dress_train$Recommendation, dress_train$Price),
margin = 2)
, beside = TRUE,
main = "proportional barplot of Recommendation, by Price")
```



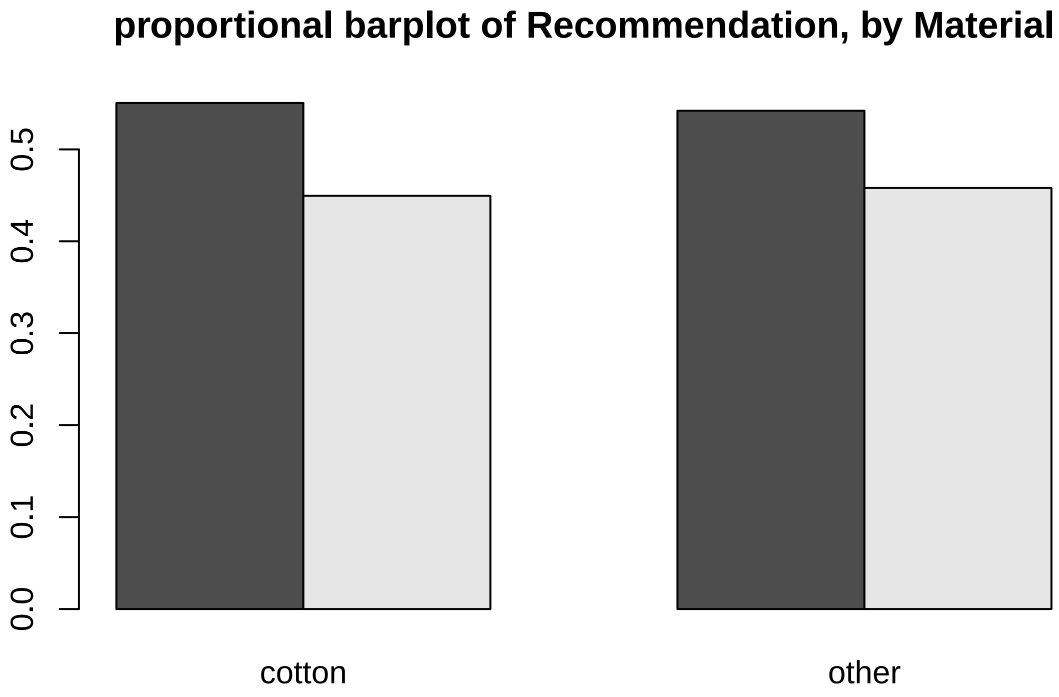
```
barplot(
prop.table(
table(dress_train$Recommendation, dress_train$Season),
margin = 2)
, beside = TRUE,
main = "proportional barplot of Recommendation, by Season")
```



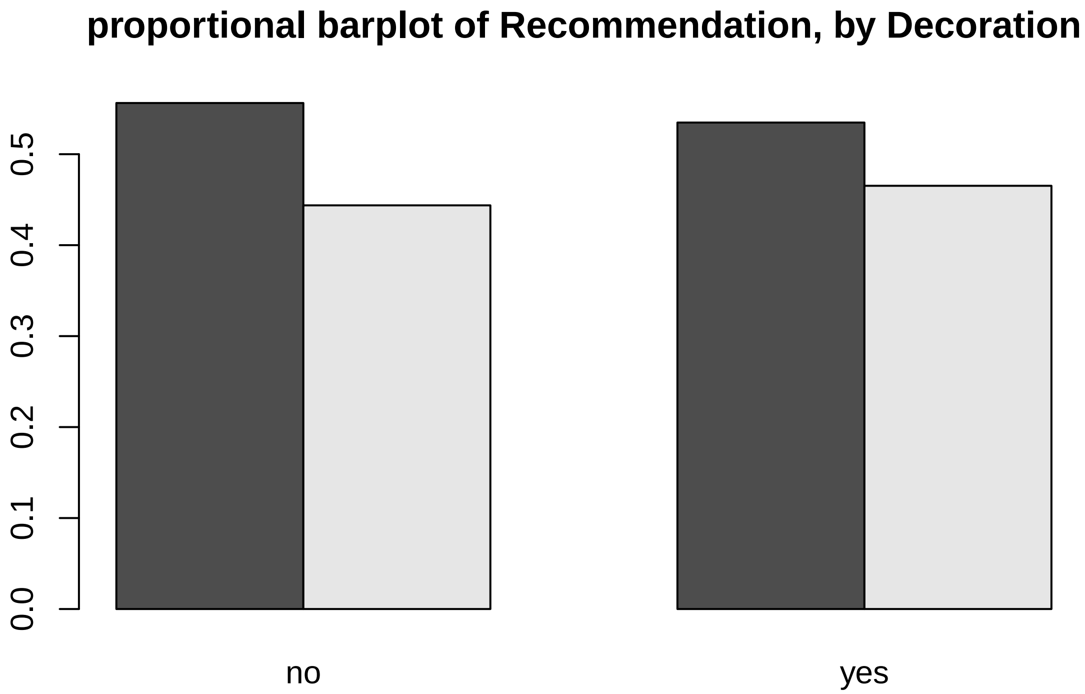
```
barplot(
prop.table(
table(dress_train$Recommendation, dress_train$NeckLine),
margin = 2)
, beside = TRUE,
main = "proportional barplot of Recommendation, by NeckLine")
```



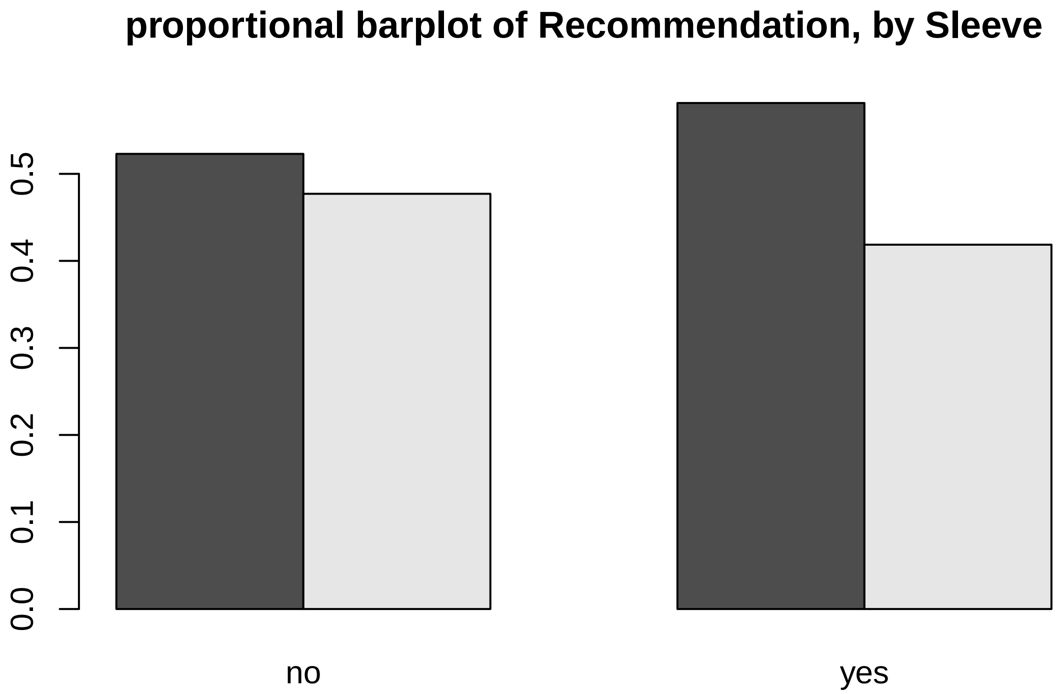
```
barplot(
prop.table(
table(dress_train$Recommendation, dress_train$Material),
margin = 2)
, beside = TRUE,
main = "proportional barplot of Recommendation, by Material")
```



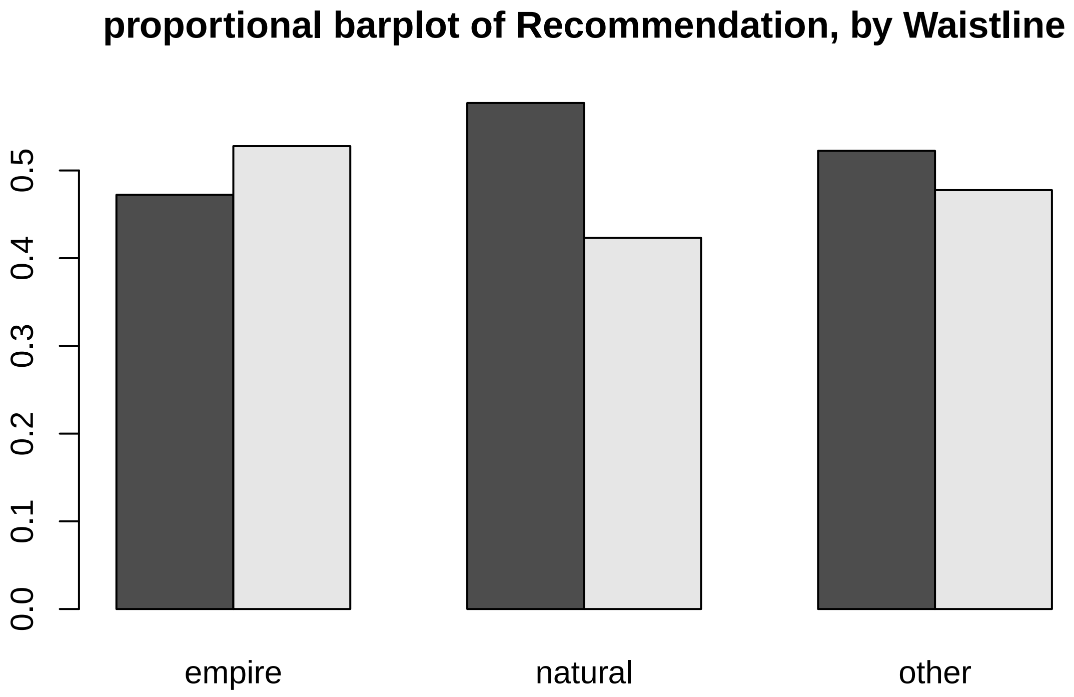
```
barplot(
prop.table(
table(dress_train$Recommendation, dress_train$Decoration),
margin = 2)
, beside = TRUE,
main = "proportional barplot of Recommendation, by Decoration")
```



```
barplot(
prop.table(
table(dress_train$Recommendation, dress_train$Pattern),
margin = 2)
, beside = TRUE,
main = "proportional barplot of Recommendation, by Sleeve")
```



```r
barplot(
prop.table(
table(dress_train$Recommendation, dress_train$Waistline),
margin = 2)
, beside = TRUE,
main = "proportional barplot of Recommendation, by Waistline")
```



The variable Style seems to have some separation of successes and failures depending on the category: The party (if in party category most likely a success) and work category (if in work category most likely a failure). The rest of the categories dont have that much os a distinction. The variable Season seems to have some separating of successes and failures in the fall and spring categories where fall tends to have more failures and spring tends to have more succeses. The variable Price, NeckLine, Material, Decoration, Sleeve, and Waistline do not seem to have that much of a distinction across their categories.

Modeling

We will build our classifier models that will help us predict if the dress will be a success or failure. We will be using the following 4 models: Linear Discriminant Analysis (lda), Quadratic Discriminant Analysis (qda), Classification Trees, and Binary Logistic Regression.

To make sure that we avoid overfitting we will have a set of training data and a set of testing data, which will be randomly split from the main data into the two groups. We will train the classifier models with the training split of the data and test the classifier models on the testing split of the data.

Summary of the Response Labels in the Testing Dataset

There are a total of 149 observations with 100 (67.11%) observations of dresses that were not successfully sold (ie failures) and 49 (32.89%) there were successfully sold.

```
table(dress_test$Recommendation)
```

```
##
##   0   1
## 100  49
```

```
prop.table(table(dress_test$Recommendation))
```

```
##
##         0         1
## 0.6711409 0.3288591
```

LDA Model

The first model we can try is the linear discrimanant analysis. This model only accounts for the quantitative variables, but the only variable we will be using to help make decisions is the Ratings variables. However since Ratings do not seems to differ by the success or failure of the dress, as indicated by out EDA, these models will probably be error prone since the Ratings variable is not very helpful as pointing out a distiction.

We will build the LDA model now and see how it performs with the data

```
dress.lda <- lda(Recommendation ~ Rating,
                 data = dress_train)
dress.lda.pred <- predict(dress.lda,
  as.data.frame(dress_test))
table(dress.lda.pred$class, dress_test$Recommendation)
```

```
##
##       0   1
##   0 100  49
##   1   0   0
```

The overall error rate for this model is (0+49)/149 = 32.89% which seems low. The error rate for failures is 0%, however the error rate for successes is 49/49 = 100% so this does not seem to give us accurate predictions.

QDA Model

We will build the QDA model now and see how it performs with the data

```
dress.qda <- qda(Recommendation ~ Rating,
                 data = dress_train)

dress.qda.pred <- predict(dress.qda,
  as.data.frame(dress_test))

table(dress.qda.pred$class, dress_test$Recommendation)
```

```
##
##       0   1
##   0 100  49
##   1   0   0
```

This model also has similar results the last as the overall error rate for this model is 32.89%, the error rate for true failures is 0% and the error rate for true successes is 100% so this modle also does not seem to give us accurate predictions.

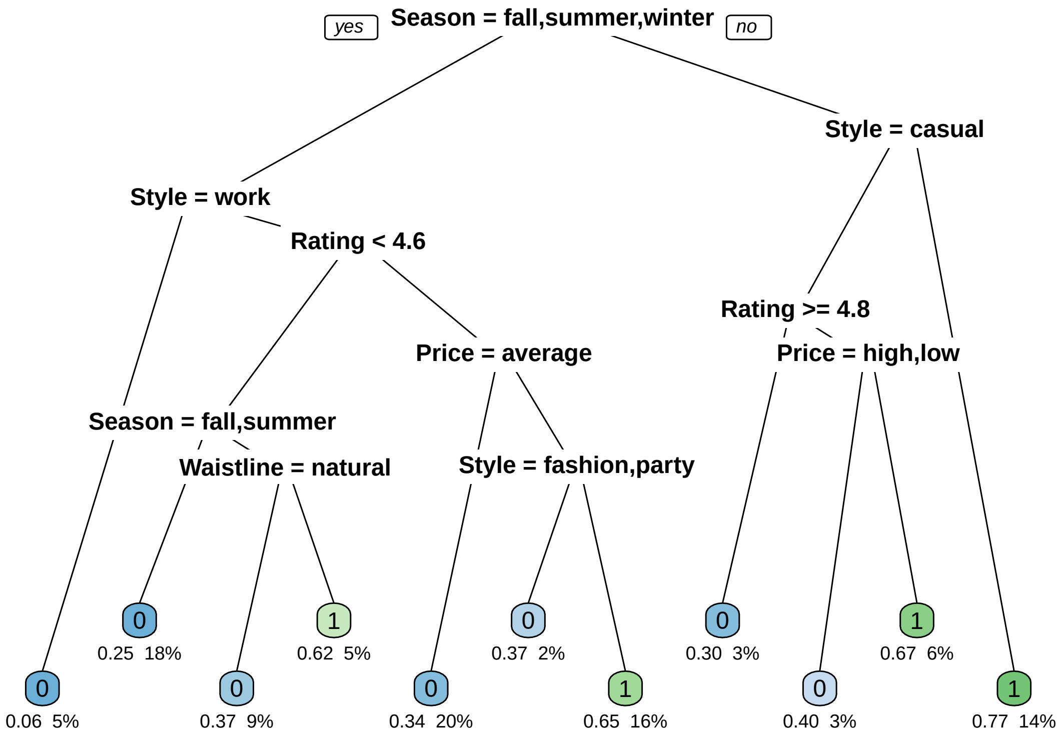
Both the LDA and QDA models do well for the true failures, but they are not able to distinguish them from the true successes. This is is probably a result of the fact that only Ratings is being taken into account to make these predictor models. When doing the EDA we saw that the Ratings do not differ based on whether the dress is a success (sells well) or a failure (it does not sell well). Thus, these predictor models were expected not to be as useful and have high error rates.

Tree Model

Now we will try the Tree Model to predict the sucesses and failures. This model will use all the predictors given and we expect the error rates to be lower than the QDA and LDA as more data will help inform the prediction decision. However trees are prone to over fitting the data, but this model could still end up with asignificant amount of errors. We will build the tree model now and see how it performs with the data

```
dress.tree <- rpart(Recommendation ~ Rating + Price + Style +
  Season + NeckLine + Material + Decoration + Pattern +
  Sleeve + Waistline,
  data=dress_train,
  method="class")

rpart.plot(dress.tree,
type = 0,
clip.right.labs = FALSE,
branch = 0.1,
under = TRUE)
```



```
dress.tree.pred <- predict(dress.tree,
  as.data.frame(dress_test),
  type="class")
table(dress.tree.pred, dress_test$Recommendation)
```

```
##
## dress.tree.pred  0  1
##               0 71 31
##               1 29 18
```

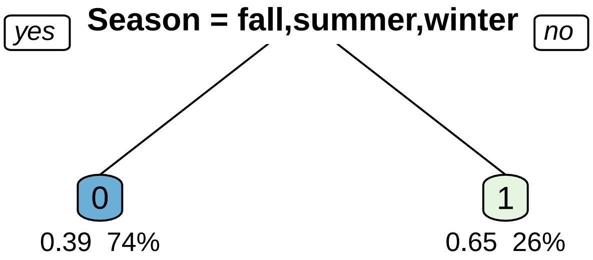
The overall error rate is (30+34)/149 = 40.27%. The error rate for true failures is 29% and error rate for true successes is 63.27%. These numbers are reallly high which means that there could be some overfitting with the tree, so we need to prune it.

Simplier Pruned Tree

Now we prune the tree, giving it a max depth of 1 to simplify the branches and see how it performs with the data

```
dress.tree2 <- rpart(Recommendation ~ Rating + Price + Style +
  Season + NeckLine + Material + Decoration + Pattern +
  Sleeve + Waistline,
  data=dress_train,
  method="class",
  control = rpart.control(maxdepth=1))

rpart.plot(dress.tree2,
type = 0,
clip.right.labs = FALSE,
branch = 0.1,
under = TRUE)
```



```
dress.tree2.pred <- predict(dress.tree2,
  as.data.frame(dress_test),
  type="class")
table(dress.tree2.pred, dress_test$Recommendation)
```

```
##
## dress.tree2.pred  0  1
##                0 82 32
##                1 18 17
```

The overall error rate is (18+32)/149 = 33.56%. The error rate for true failures is 18% and error rate for true successes is 65.31%. This model would still be better to use because the overall error rate for this model is much lower than the LDA, QDA, and more complex tree. This model also has less chance of overfitting future data, as there are less predictors and the tree is much simplier.

Binary Logistic Regression Model

Now we can try a Binary Logistic Regreesion model to predict successes and failures and see how it performs with the data.

```
dress.logit <- glm(factor(Recommendation) ~ Rating + factor(Price) + factor(Style)+
  factor(Season) + factor(NeckLine) + factor(Material) + factor(Decoration) + factor(Pattern) +
  factor(Sleeve) + factor(Waistline),
    data = dress_train,
    family = binomial(link = "logit"))

dress.logit.prob <- predict(dress.logit,
  as.data.frame(dress_test),
  type = "response")

levels(factor(dress_test$Recommendation))
```

```
## [1] "0" "1"
```

```
dress.logit.pred <-ifelse(dress.logit.prob > 0.5,"0","1")
table(dress.logit.pred, dress_test$Recommendation)
```

```
##
## dress.logit.pred  0  1
##                0 23 22
##                1 77 27
```

The overall error rate is (77+22)/149 = 66.44%. The error rate for true failures is 77% and error rate for true successes is 44.90%. The over all error rate for this model is the highest out of all the models, however it seems to have the lowest error rate for true successes. This model would not be a good choice regardless because the overall error rate is quite high.

Final Recommendation

The model to use best would the the simplier tree model to predict the successes and failures of a dress selling well. This is because the model has the lowest error rate overall, 33.65%, and the lowest error rate for true successes, 65.31%. The LDA and QDA models gave us 100% error for true successes of the dress selling well, which means that these models were very error prone. The Binary Logistic Model gave an overall error rate of 66.44%, which was much higher than the either of the tree models. The Tree Model (not Pruned) was better than the latter ones however it still had an overall error rate of 44.90% which was still high. However it is important to note that the Tree Model (not Pruned) did better with the true successes error rate as the error rate for true successes was 63.27%.

Since the overall error rate of the Simplier Pruned Tree model is lower than for the Tree Model (not Pruned), we recommend going with the Simplier Pruned Tree mdoel as it has the lowest overall error rate and the second lowest error rate for true successes.

Discussion

As expected the classifier models that took into account categorical variables performed better than those that didn't as the rating variable didn't differ across successes or failures. Thus it was expected that the Tree and Binary Logistic Regression models would do better than the LDA and QDA models. The Simplier

Pruned Tree Model ended up doing better than the Tree Model in terms of the overall error rate. However it is important to keep in mind that the Tree Model (not Pruned) had a lower error rate for true successes (63.27%). However overall the Simplier Pruned Tree Model would be the better model as it has a lower overall error rate and since it is simplier it is also less prone to over fitting. However a limitation of this model is that the error rate for true successes is quite high, which means you are more likely to find true successes that were accidently classified as errors To overcome these limitations and make the classifier models more accurate, it would be helpful to take into account more observations variables such as the time of selling, the Location placed in the store, types of advertisements, and similarity to currently trending styles.