Loan prediction based on customer behavior

Code

How does linear models for classification work?

Classification is a form of qualitative data where we are looking at categorical data for linear models. This is a great model to use when wanting to understand data in a simple and straight forward form. The downside to using a linear model is that it assumes the data is in linear form, which is not how real world data sets work. One great thing about the model is that the model can get updated very easily with new data but unless the data set has linear relationships, this is not the best model to use.

a. Divide into 80/20 train/set

Hide

```
set.seed(1234)
i <- sample(1:nrow(loan2.data), nrow(loan2.data)*0.80, replace=FALSE)
train2 <- loan2.data[i,]
test2 <- loan2.data[-i,]
```

It is important for the larger chunk to be the training data so when we predict new values, we can see how accurate the computer was with the test data since it has not seen this set yet.

b. The 5 R functions I used for data exploration on the training data was sum, range, correlation, standard deviation, and covariance

Hide

```
sum(train2$Experience)
range(train2$House_Ownership)
cov(train2$Income, train2$Age)
cor(train2$Income, train2$Age)
sd(train2$Income)
```

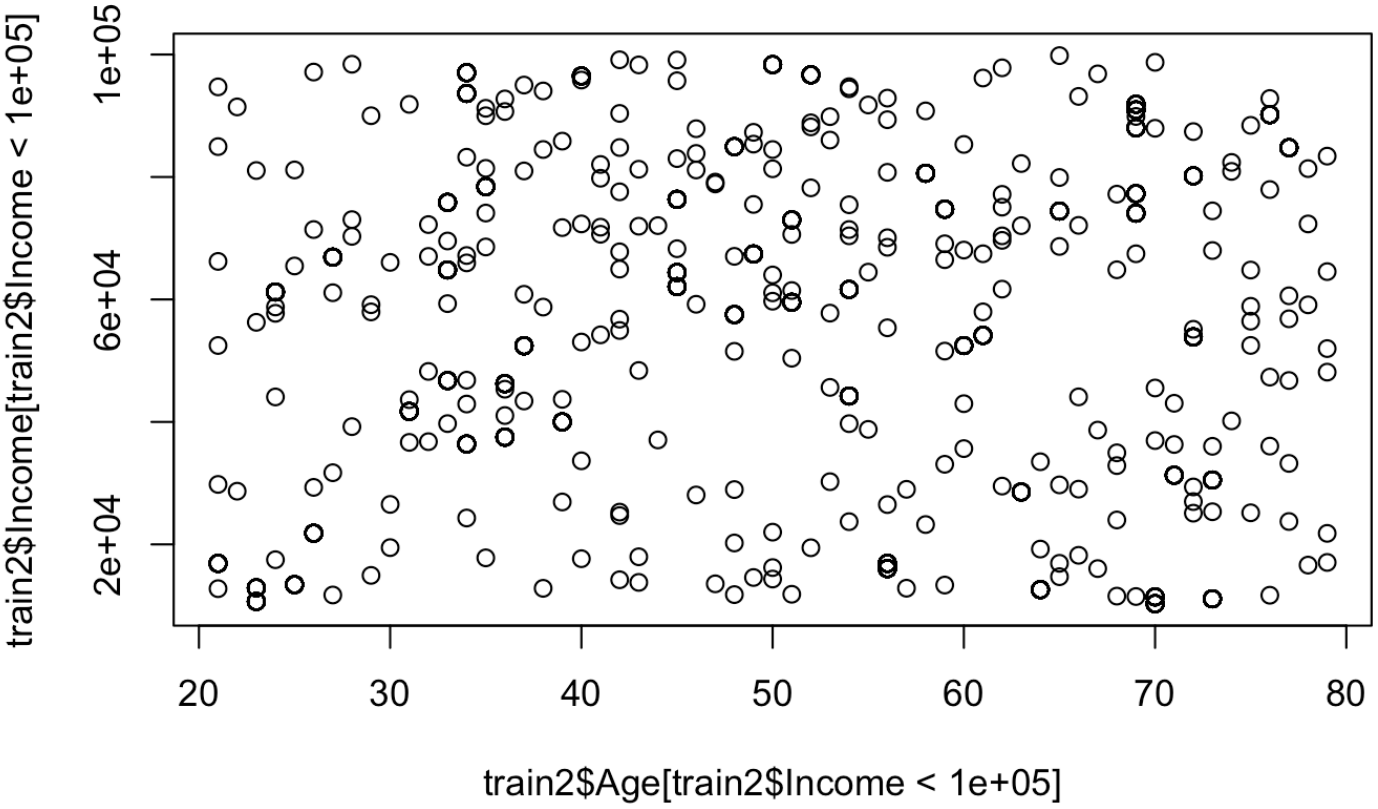
c. 2 informative graphs using the training data

Hide

```
hist(train2$Age)
```

Hide

```
plot(train2$Age[train2$Income < 100000] ,
     train2$Income[train2$Income < 100000])
```



I created a histogram with the age of the customers and how frequently it was seen in the dataset. The next graph I built was a plot diagram with the Age and Income of customers. Because my dataset was so large, I chose a chunk of the data to plot on the graph otherwise the data points were not able to be seen as easily.

d. Simple logistic regression model

Hide

```
glm1 <- glm(loan2.data$Income~loan2.data$Experience, data=train2)
summary(glm1)
```

The following model summary gives us a good chunk of information, including the glm call, the residual distribution, the coefficients, and model metrics. The deviance residual gives us information on the loss function and what specific points are contributing to that occurrence. For my formula, our deviance residuals are given in the forms of Min, 1Q, Median, 3Q, and Max. We also get information on the null deviance and residual deviance. The null deviance is responsible for measuring the lack of fit within our model with only the intercept and the residual deviance measures it with the entire model. A good indicator is when the residual deviance is lower than the null deviance which is seen in our data set. And lastly the AIC is the Akaike Information Criterion which tells us the likelihood of future values based on predictions from the data set.

e. naïve Bayes model

Hide

```
library(e1071)
nb1 <- naiveBayes(train2$Risk_Flag~., data=train2)
nb1
```

The Bayes model works the best with discrete values and gives us a summary on likelihood data with certain conditional probabilities. The A-priori for whether or not a person is a risk for a loan or not is 0.876 and our likelihood data is known as the conditional probabilities shown in the output. We also notice that the variables like Id, Income, Age, etc. all have 2 discrete and each row will sum up to one, for example, the Age variable has 50.08650 + 48.96175 = 99.048 which is very close to 100. This model gives us a better understanding of the classification data by using the predictor variables and how they are independent of each other.

f. Predict and evaluate test data

Hide

```
#predicting test data
pred <- predict(glm1, newdata=test2)

cor <- cor(pred, loan2.data$Risk_Flag)
cor

mse <- mean((pred - loan2.data$Risk_Flag)^2)
mse

rmse <- sqrt(mse)
rmse

#test data
glm2 <- glm(loan2.data$Income~loan2.data$Experience, data=test2)
summary(glm2)
```

For the following code block, we predicted some new test data to see how accurate the algorithm could estimate new values based on data it has not seen yet. I then used the built in R correlation function on the predicted data set and the loan2.data Risk_Flag data set. This variable tells us whether or not a customer is a risk for a credit/loan application. I also took the mse of those two variables and the rmse. The following three functions were done using the predicted values and I then ran a logistic regression on the test data because the prior regression model was done using the training data. I got very similar results using the predicted values and the test values as well. The console output gave me the deviance residuals, the coefficients, and null/residual deviance of the following values.

g. Naïve Bayes and Logistic Regression

Naïve Bayes is a probability algorithm that finds the maximum likelihood of an event occurring. Similar to this, logistic regression models the probability of an outcome using binary prediction based on previous data sets. Although both models are great for interpreting data, one will be more suitable depending on the situation and they both come with their own disadvantages. For Naïve Bayes theorem, the model assumes that all the

predictors are independent when that is not always the case in real world data sets. However, some of the advantages with Bayes is how easy it is to implement, it does not require a lot of training data, and it handles both continuous and discrete data. Logistic regression, similarly, is easy to implement as well, and provides a measure of how appropriate a predictor is as well as whether the direction of assosciation is positive and negative. Some drawbacks with this model are that it constructs linear boundaries, so it can only solve linear problems which is not accurate for real world data sets. Another weakness is that it can only be used to predict discrete functions whereas Bayes can do discrete and continuous.

h. Benefits and drawbacks with the classification metrics used

Two classification models I used were the mean squared error (mse) and root mean squared error (rmse) which was very useful when I was comparing two models that were built on the same training data. The mse was also very beneficial because I was able to confirm that the trained model did not have any outliar predictions with any potential errors. A drawback to using the mse is if by any change there are bad predictions, the metric will increase the magnitude of the error due to the squaring within the formula. I also used correlation to compare the predicted values to the test values as well. This was beneficial for me because I was able to see how similar the values were and whether or not the model did a good job at predicting the values based on the training set. If the correlation value was low, that means the values were not dependent on one another and the predictions were not accurate.