HW 10.3

Yu Fung David Wang

Data Attributes

Attribute 1: (qualitative)
Status of existing checking account A11 : ... < 0 DM A12 : 0 <= ... < 200 DM A13 : ... >= 200 DM / salary assignments for at least 1 year A14 : no checking account

Attribute 2: (numerical) Duration in month

Attribute 3: (qualitative) Credit history A30 : no credits taken/ all credits paid back duly A31 : all credits at this bank paid back duly A32 : existing credits paid back duly till now A33 : delay in paying off in the past A34 : critical account/ other credits existing (not at this bank)

Attribute 4: (qualitative) Purpose A40 : car (new) A41 : car (used) A42 : furniture/equipment A43 : radio/television A44 : domestic appliances A45 : repairs A46 : education A47 : (vacation - does not exist?) A48 : retraining A49 : business A410 : others

Attribute 5: (numerical) Credit amount

Attibute 6: (qualitative) Savings account/bonds A61 : ... < 100 DM A62 : 100 <= ... < 500 DM A63 : 500 <= ... < 1000 DM A64 : .. >= 1000 DM A65 : unknown/ no savings account

Attribute 7: (qualitative) Present employment since A71 : unemployed A72 : ... < 1 year A73 : 1 <= ... < 4 years
A74 : 4 <= ... < 7 years A75 : .. >= 7 years

Attribute 8: (numerical) Installment rate in percentage of disposable income

Attribute 9: (qualitative) Personal status and sex A91 : male : divorced/separated A92 : female : divorced/separated/married A93 : male : single A94 : male : married/widowed A95 : female : single

Attribute 10: (qualitative) Other debtors / guarantors A101 : none A102 : co-applicant A103 : guarantor

Attribute 11: (numerical) Present residence since

Attribute 12: (qualitative) Property A121 : real estate A122 : if not A121 : building society savings agreement/ life insurance A123 : if not A121/A122 : car or other, not in attribute 6 A124 : unknown / no property

Attribute 13: (numerical) Age in years

Attribute 14: (qualitative) Other installment plans A141 : bank A142 : stores A143 : none

Attribute 15: (qualitative) Housing A151 : rent A152 : own A153 : for free

Attribute 16: (numerical) Number of existing credits at this bank

Attribute 17: (qualitative) Job A171 : unemployed/ unskilled - non-resident A172 : unskilled - resident A173 : skilled employee / official A174 : management/ self-employed/ highly qualified employee/ officer

Attribute 18: (numerical) Number of people being liable to provide maintenance for

Attribute 19: (qualitative) Telephone A191 : none A192 : yes, registered under the customers name

Attribute 20: (qualitative) foreign worker A201 : yes A202 : no

Read in Data

```
creditdata <- read.table("germancredit.txt", stringsAsFactors=T)
head(creditdata)
```

```
##    V1 V2  V3  V4   V5  V6  V7 V8  V9  V10 V11  V12 V13  V14  V15 V16  V17 V18
## 1 A11  6 A34 A43 1169 A65 A75  4 A93 A101   4 A121  67 A143 A152   2 A173   1
## 2 A12 48 A32 A43 5951 A61 A73  2 A92 A101   2 A121  22 A143 A152   1 A173   1
## 3 A14 12 A34 A46 2096 A61 A74  2 A93 A101   3 A121  49 A143 A152   1 A172   2
## 4 A11 42 A32 A42 7882 A61 A74  2 A93 A103   4 A122  45 A143 A153   1 A173   2
## 5 A11 24 A33 A40 4870 A61 A73  3 A93 A101   4 A124  53 A143 A153   2 A173   2
## 6 A14 36 A32 A46 9055 A65 A73  2 A93 A101   4 A124  35 A143 A153   1 A172   2
##    V19  V20 V21
## 1 A192 A201   1
## 2 A191 A201   2
## 3 A191 A201   1
## 4 A191 A201   1
## 5 A191 A201   2
## 6 A192 A201   1
```

```
set.seed(1234)
```

Split Data into Train, Validate, Testing

```
splitSample <- sample(1:3, size = nrow(creditdata), prob = c(0.7, 0.15, 0.15), replace = TRUE)

creditdata.train <- creditdata[splitSample == 1,]
creditdata.val <- creditdata[splitSample == 2,]
creditdata.test <- creditdata[splitSample ==3,]
```

Fit Logistic Regression Models on Train Data to Find a Good Predictive Model for whether Credit Applicants are Good Credit Risks or Not **(Question 10.3(1))**

```
# Essentially, we want to predict V21 (Binary Status of Credit Account where 1 = Good, 2 = Bad)

# Model 1: Prediction based on Credit History (V3) and # of Existing Credits at Bank (V16)

model1 <- glm(as.factor(V21)~V3+V16, family = binomial(link = "logit"), data = creditdata.train)

# Model 2: Prediction based on Employment which includes present years of employment (V7), installment

model2 <- glm(as.factor(V21)~V7+V8+V17, family = binomial(link = "logit"), data = creditdata.train)

# Model 3: Prediction based on identity such as age (V13), foreign worker (V20)

model3 <- glm(as.factor(V21)~V13+V20, family = binomial(link = "logit"), data = creditdata.train)


# Model Baseline: Prediction based on all features

modelbase <- glm(as.factor(V21)~., family = binomial(link = "logit"), data = creditdata.train)
```

See How the Models Performed on the Validation Data

- Have to Determine a Good Threshold Probability Based on Model **(Question 10.3(2))**
- Since we are dealing with if applications pose a credit risk, being a credit risk is quite a big financial loss and issue for businesses/groups that require good credits such as banks. Hence, it would not make sense

to set this threshold to be too low in the case we get a false positive. In the case of a false positive, it would cause lots of issues and threats, whereas a false negative may not pose too big of a issue, as those individuals can be invited to reapply. However, this threshold may differ between different models, so each model will have a different threshold depending on the features used.

```
# Actual Data: (Binary Status of Credit Account where 1 = Good, 2 = Bad)

# Accuracy = sum(all the correctly identified cases) divided by all cases

# For model 1, predictions were made based on historical and existing records of credits. These are two

model1.results <- predict(model1, newdata = creditdata.val, type = "response")
model1.data <- ifelse(model1.results > 0.9, 1, 2)

model1.acc <- sum(1*(model1.data == creditdata.val$V21))/length(creditdata.val$V21)

paste("Model 1: Historical/Existing Credit Records", model1.acc)
```

```
## [1] "Model 1: Historical/Existing Credit Records 0.352201257861635"
```

```
# For model 2, predictions were made based on employment details. Employment is typically a good indica

model2.results <- predict(model2, newdata = creditdata.val, type = "response")
model2.data <- ifelse(model2.results > 0.2, 1, 2)

model2.acc <- sum(1*(model2.data == creditdata.val$V21))/length(creditdata.val$V21)

paste("Model 2: Employment Records", model2.acc)
```

```
## [1] "Model 2: Employment Records 0.471698113207547"
```

```
# For model 3, predictions were made based on identity. Specifically, this includes age and alien statu

model3.results <- predict(model3, newdata = creditdata.val, type = "response")
model3.data <- ifelse(model3.results > 0.7, 1, 2)

model3.acc <- sum(1*(model3.data == creditdata.val$V21))/length(creditdata.val$V21)

paste("Model 3: Identity Records", model3.acc)
```

```
## [1] "Model 3: Identity Records 0.352201257861635"
```

```
# Baseline model with all features so will just give it halfway of 0.5

modelbase.results <- predict(modelbase, newdata = creditdata.val, type = "response")
modelbase.data <- ifelse(modelbase.results > 0.5, 1, 2)

modelbase.acc <- sum(1*(modelbase.data == creditdata.val$V21))/length(creditdata.val$V21)

paste("Baseline Model", modelbase.acc)
```

```
## [1] "Baseline Model 0.238993710691824"
```

Given the points made earlier, we first see that all models outperformed the baseline model with all features. This means that feature selection was worthwhile and sensible in understanding the credit risks of applications. Then, we see that model 1 and model 3 had the same result despite different threshold values. Finally, Model 2 had the best result and it had the lowest threshold value due to the factor of employment not having the greatest direct influence towards credit score.

However, despite model 2 having the highest accuracy, these features may not make the most sense as features that tend to not have a high influence will not be utilized the best in real-world situations. As such, we should still use either model 1 or model 3 as the two of them gives better indicators. Model 1 makes more sense as it directly utilizes previous history and current credit states to predict credit risk.

```
highacc_model <- predict(model2, newdata = creditdata.test, type = "response")
highacc_model.data <- ifelse(highacc_model > 0.2, 1, 2)

highacc_model.acc <- sum(1*(highacc_model.data == creditdata.test$V21))/length(creditdata.test$V21)

paste("(High Accuracy) Logistic Model Accuracy on Application Credit Risk with Threshold = 0.2:", higha
```

```
## [1] "(High Accuracy) Logistic Model Accuracy on Application Credit Risk with Threshold = 0.2: 0.5734
```

```
real_model <- predict(model1, newdata = creditdata.test, type = "response")
real_model.data <- ifelse(real_model > 0.2, 1, 2)

real_model.acc <- sum(1*(real_model.data == creditdata.test$V21))/length(creditdata.test$V21)

paste("(Real World) Logistic Model Accuracy on Application Credit Risk with Threshold = 0.9:", real_mod
```

```
## [1] "(Real World) Logistic Model Accuracy on Application Credit Risk with Threshold = 0.9: 0.4615384
```