Lab 09 - LogisticRegression

Pouria

3/28/2022

Prompt

This dataset contain the data related to the passengers of the Titanic. The goal is to see if we are able to build a model that can predict whether or not a passenger would survive. The `Survived` column of this dataset reflects if they really survived or not (`1`: survived, and `0`: didn't survive).

1. Load the attached csv file into your Markdown accounting for the missing values.
2. Preprocess the data.

- Check for the missing values for each variable.
- Use `missmap()` to visualize the missing values.
- Reasonably ignore variables that probably do not affect the predictions.
- Impute the missing values in the numeric (continuous variables)
- Remove any of the rows with `NA` in their categorical variables (any categorical variable that is left in your data after leaving the unnecessary ones out).

1. Import/read the data.

Load the attached csv file into your Markdown accounting for the missing values.

Import the csv file

Let's import the training data here:

```
training.data.raw <- read.csv('train.csv', na.strings=c("")

head(training.data.raw)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                  Name
## 1                             Braund, Mr. Owen Harris
```

2. Preprocess the data.

Missing values

- Check for the missing values for each variable.

Now we need to check for missing values and look how many unique values there are for each variable using the `sapply()` function.

```
sapply(training.data.raw,function(x) sum(is.na(x)))
```

```
## PassengerId    Survived      Pclass        Name
##           0           0           0           0
##       SibSp       Parch      Ticket        Fare          Ca
##           0           0           0           0
```

```
sapply(training.data.raw, function(x) length(unique(x)))
```

```
## PassengerId    Survived      Pclass        Name
##         891           2           3         891
```

3. Split the data

Split the data up into the training and test set. (`set.seed = 1`).

Split the data

- Sample 100 rows from the data and use as your test set. Use the rest as your training set.

```
set.seed(1)
test.ID <- sample(dim(data)[1], 100)
train <- data[-test.ID,]
test <- data[test.ID,]
```

4. Fit a logistic regression model onto the training set

Fit and summarize

- Fit and summarize the model to predict Survived as a function of all other remaining variables.

```
model <- glm(Survived ~ ., family=binomial(link="logit"), c
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "lo
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6224  -0.5968  -0.4434   0.6336   2.4269
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
```

5. Evaluate your model on the test set.

Misclassification error

- Use your model on the test set to make predictions of `Survived`.
- Calculate the misclassification error on your test set.

```
pred.results <- predict(model, newdata=subset(test, select=

pred.results <- ifelse(pred.results > .5, 1, 0)
```

```
misClassificationError <- mean(pred.results!= test$Survived
print(paste("Accuracy = ", 1-misClassificationError))
```

```
## [1] "Accuracy =  0.83"
```

Calculate the CV error

- the accuracy result your obtained depends on the test set – so, now re-estimate the accuracy with a 10-fold CV

6. Fit a new model

- Redo the last two steps with only `Pclass` and `Sex`.

Implement a new fit with `Pclass` and `Sex`

Do the fit again with only Pclass and Sex;

```
model2 <- glm(Survived ~ Pclass + Sex, family=binomial(link
summary(model2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex, family = binomial
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1696  -0.7104  -0.4680   0.6804   2.1288
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```