P03 Results and Operationalization

Kris Walker

12/10/2019

Introduction

Our final goal is to build a model that can predict whether the income of a random American adult is less than or greater than 50K a year based on given features such as age, education, occupation, gender, race, etc.

```
library(ggplot2)
library(scales)
library(plyr)
library(vcd)
library(ggthemes)
library(caret)
library(GoodmanKruskal)
library(VIF)
library(ResourceSelection)
library(randomForest)
library(e1071)
library(nnet)
library(DMwR)
library(here)
```

Reading the Preprocessed Data

We read the train and test data into the `adult_train` and `adult_test` dataframes, respectively:

```
setwd("/home/taudin/MiscFiles/Fall19/CSCI385/DSProject/CensusData")
adult_train <- read.csv("adult_df.csv")
adult_test <- read.csv("test_df.csv")
```

Logistic Regression

Since we are interested in predicting the values of the variable `income`, `income` will be the response variable. It assumes only two values - less than 50K per year or more than 50K per year. We are considering a classification problem. Let Y_i be the random variable "income of the ith subject". Let also $Y_i = 1$ if income is greater than 50K and $Y_i = 0$ if income is less than or equal to 50K. Then, $Y_i(i = 1, 2, \ldots, n,)$ where $n = 30162$ is the count of observations in the data frame which follows the binomial distribution. Since we have a binary response variable, we use a logistic regression model.

Fitting the Logistic Regression Model

We start with a list of explanatory variables that consist of all variables except `fnlweight`, `hours_per_week`, `native_country`, `capital_gain`, and `capital_loss`.

```
names(adult_train)
```

```
##  [1] "age"            "workclass"      "fnlwgt"         "education"
##  [5] "education_num"  "marital_status" "occupation"     "relationship"
##  [9] "race"           "sex"            "capital_gain"   "capital_loss"
## [13] "hours_per_week" "native_country" "income"         "hours_worked"
## [17] "native_region"  "cap_gain"       "cap_loss"
```

```
covariates <- paste("age", "workclass", "education", "education_num", "marital_status",
"occupation",
                    "relationship", "race", "sex", "native_region", "hours_worked", "cap
_gain", "cap_loss",
                    sep = "+")
form <- as.formula(paste("income ~", covariates))

glm_model <- glm(formula = form, data = adult_train, family = binomial(link = "logit"),
 x = TRUE)
```

Collinearity of Predictor Variables

Let’s investigate if there are collinear predictor variables beginning with a summary of the fit logistic model:

```
summary(glm_model)$coefficients[, 1:2]
```

```
##                                          Estimate   Std. Error
## (Intercept)                            0.91201267 6.771106e-01
## age                                    0.02644527 1.703476e-03
## workclass Local-gov                   -0.63979788 1.122291e-01
## workclass Private                     -0.45296679 9.310283e-02
## workclass Self-emp-inc                -0.23579597 1.233487e-01
## workclass Self-emp-not-inc            -0.87237123 1.099155e-01
## workclass State-gov                   -0.75407759 1.250335e-01
## workclass Without-pay                -13.17870374 1.993486e+02
## education 11th                         0.10327139 2.136984e-01
## education 12th                         0.44357605 2.736652e-01
## education 1st-4th                     -0.46351454 4.798060e-01
## education 5th-6th                     -0.42436451 3.520322e-01
## education 7th-8th                     -0.50459676 2.420517e-01
## education 9th                         -0.27989325 2.691258e-01
## education Assoc-acdm                   1.28649157 1.802168e-01
## education Assoc-voc                    1.26237026 1.731703e-01
## education Bachelors                    1.90332005 1.612117e-01
## education Doctorate                    2.96804170 2.231656e-01
## education HS-grad                      0.76439229 1.567952e-01
## education Masters                      2.26099269 1.721540e-01
## education Preschool                  -12.51222254 9.782617e+01
## education Prof-school                  2.90128621 2.067497e-01
## education Some-college                 1.12458256 1.590538e-01
## marital_status Married-AF-spouse       2.90690753 5.786968e-01
## marital_status Married-civ-spouse      2.16467057 2.735431e-01
## marital_status Married-spouse-absent   0.00945387 2.365227e-01
## marital_status Never-married          -0.44259615 8.766392e-02
## marital_status Separated              -0.07290986 1.640836e-01
## marital_status Widowed                 0.21333400 1.567969e-01
## occupation Armed-Forces               -1.38136623 1.593366e+00
## occupation Craft-repair                0.03695684 8.042066e-02
## occupation Exec-managerial             0.76891302 7.746007e-02
## occupation Farming-fishing            -0.86754594 1.380989e-01
## occupation Handlers-cleaners          -0.71716202 1.442101e-01
## occupation Machine-op-inspct          -0.31501476 1.027861e-01
## occupation Other-service              -0.79072460 1.181541e-01
## occupation Priv-house-serv            -3.19476178 1.316902e+00
## occupation Prof-specialty              0.50249174 8.199658e-02
## occupation Protective-serv             0.59361727 1.255850e-01
## occupation Sales                       0.25797933 8.301024e-02
## occupation Tech-support                0.65900377 1.114930e-01
## occupation Transport-moving           -0.07079422 9.946665e-02
## relationship Not-in-family             0.58318919 2.704634e-01
## relationship Other-relative           -0.28234326 2.452131e-01
## relationship Own-child                -0.61474339 2.697575e-01
## relationship Unmarried                 0.43955664 2.859820e-01
## relationship Wife                      1.38045241 1.050326e-01
## race Asian-Pac-Islander                0.72794887 2.747593e-01
## race Black                             0.52825495 2.408549e-01
## race Other                             0.18381759 3.703321e-01
## race White                             0.63160176 2.303950e-01
## sex Male                               0.83936192 7.951229e-02
```

```
## native_region Central-Asia          -0.06180058 2.892526e-01
## native_region East-Asia              0.05345392 2.625967e-01
## native_region Europe-East            0.35955691 3.357858e-01
## native_region Europe-West            0.56504172 1.941417e-01
## native_region Outlying-US            0.28042604 2.233229e-01
## native_region South-America         -0.99100862 4.696179e-01
## native_region United-States          0.41107875 1.356727e-01
## hours_worked between_45_and_60       0.43829975 4.369557e-02
## hours_worked between_60_and_80       0.41123185 9.821219e-02
## hours_worked less_than_40           -0.80461744 6.216447e-02
## hours_worked more_than_80            0.27399981 1.935998e-01
## cap_gainLow                         -6.65655806 5.091866e-01
## cap_gainMedium                      -4.77939873 5.138280e-01
## cap_lossLow                         -0.79021189 1.489679e-01
## cap_lossMedium                       0.80733753 1.791349e-01
```

From the output of `summary(glm.model)` we notice the following warning message: "Coefficients: (1 not defined because of singularities)" and in the coefficients table, the coefficient for the variable `education_num` is NA. This is an indication that the covariate `education_num` is collinear with some other predictor. We have to exclude it from the list of predictor variables and fit the model again. We will use the R function `findLinearCombos()` from the package `caret` to test whether the covariate `education_num` is collinear with some of the other predictors. `findLInearCombos()` returns a list that enumerates these dependencies and a vector of column positions that can be removed to eliminate the linear dependencies:

```
findLinearCombos(glm_model$x)
```

```
## $linearCombos
## $linearCombos[[1]]
##  [1] 24  1  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
##
##
## $remove
## [1] 24
```

```
findLinearCombos(glm_model$x)$remove
```

```
## [1] 24
```

R found linear dependencies between the covariates and recommends to remove column 24 from the design matrix. Below we identify which predictor corresponds to column 24:

```
colnames(glm_model$x)[findLinearCombos(glm_model$x)$remove]
```

```
## [1] "education_num"
```

The problematic predictor is `education_num`. The variable `education_num` provides redundant information, same as what's contained in `education`. From the content of `education_num` and `education` we can also see that the two variables are linearly dependent, i.e. collinear. Below we list the unique combinations of values for the variables `education` and `education_num`. The variable `education` has a total of 16 factor levels and we see that each level of `education` corresponds to a number from the variable `education_num`:

```
unique_combinations <- unique(adult_train[,c("education", "education_num")])

unique_combinations[order(unique_combinations$education_num),]
```

```
##         education education_num
## 209     Preschool             1
## 387       1st-4th             2
## 53        5th-6th             3
## 15        7th-8th             4
## 7             9th             5
## 205          10th             6
## 4            11th             7
## 386          12th             8
## 3         HS-grad             9
## 11   Some-college            10
## 46      Assoc-voc            11
## 14     Assoc-acdm            12
## 1       Bachelors            13
## 6         Masters            14
## 49    Prof-school            15
## 20      Doctorate            16
```

We remove the covariate `education_num` and fit the new model `glm_model_wld`, resolving the problem with linearly dependent predictors:

```
new_covariates <- paste("age", "workclass", "education","marital_status","occupation",
"relationship", "race",                         "sex", "native_region", "hours_worked",
"cap_gain", "cap_loss", sep = "+")

new_form <- as.formula(paste("income ~", new_covariates))

glm_model_wld <- glm(formula = new_form,
                     data = adult_train,
                     family = binomial(link = "logit"),
                     x = TRUE,
                     y = TRUE)
```

There aren’t linear dependencies between the covariates anymore:

```
findLinearCombos(glm_model_wld$x)
```

```
## $linearCombos
## list()
##
## $remove
## NULL
```

Other Collinearity Detection Diagnostics

Another way to see if there are correlations between covariates is to calculate the Goodman and Kruskal's tau measure for all pairs of covariates. The Goodman and Kruskal's tau measures the strength of association between categorical variables, but for discrete numerical variables, the Goodman Kruskal's tau measure treats each value as a separate level of a factor variable. Although not applicable to categorical variables, the standard measure is the Pearson's correlation coefficient.

The Goodman and Kruskal's tau measure will give us the strength of association between predictors two by two. This coefficient will not help us identify any association if there are 3 or more mutually dependent predictors. The tau measure ranges from 0 to 1; values closer to zero indicate weak association, whereas values closer to 1 indicate strong association. The tau measure is not symmetric. This means that, if A and B are two categorical variables, then

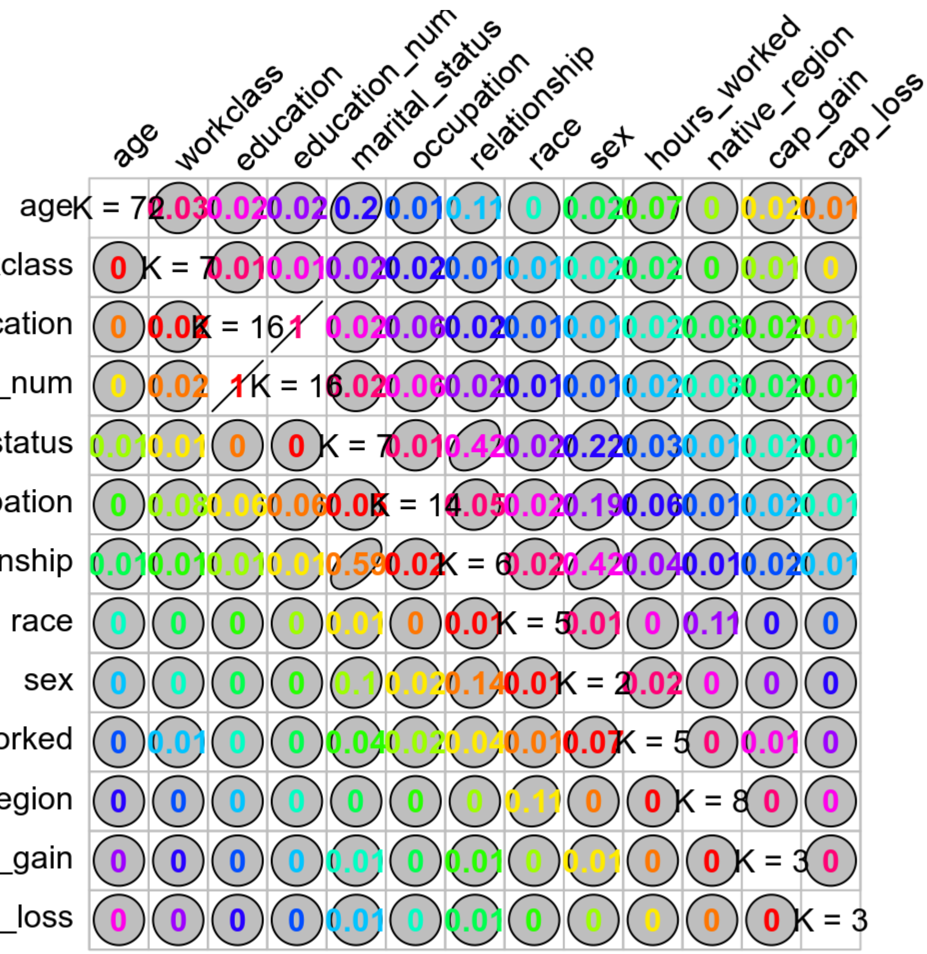
$$\tau(A, B) \neq \tau(B, A)$$

.

The function `GKtauDataframe()` returns an object of class `GKtauMatrix` and the plotting can be applied for this object class, allowing us to visualize all pairs of predictors. The plot is in the form of a matrix. Numbers on the diagonal are equal to the number of levels for each categorical variable, while the off-diagonal numbers display the Goodman-Kruskal tau values. Each tau measure is represented by an ellipse, which is a circle for $\tau = 0$ and degenerates into a straight line for $\tau = 1$.

```
GK_matrix <- GKtauDataframe(adult_train[, c("age", "workclass", "education", "education_
num", "marital_status",                                             "occupation", "relat
ionship", "race", "sex", "hours_worked",
                                            "native_region", "cap_gain", "cap_loss")])

plot(GK_matrix)
```



We see that `education` is predictable from `education_num` and vice versa. This confirms our conclusion that the two variables are collinear. All other τ values are close to zero, except for τ(relationship, marital_status), τ(relationship, sex), and τ(marital_status, relationship). This makes sense, because `relationship` can be predicted by `marital_status` and vice versa. However, the association between `relationship` and `sex` isn't really obvious. The tau value of 0.42 suggests that being a female or male can determine the type of relationship that an individual is in. Looking at the percentage of women and men belonging to each category of the factor variable `relationship`, we see that 36% of women are Not-in-family compared to 20% of men, and 25% of women are Unmarried in contrast to only 4% of men.

```
tab <- xtabs(~ sex + relationship, data = adult_train)

ptab <- prop.table(tab, 1)

print(format(round(ptab, 2), scientific = FALSE))
```

```
##          relationship
## sex        Husband  Not-in-family  Other-relative  Own-child  Unmarried
##    Female "0.00"   "0.36"         "0.04"          "0.20"     "0.25"
##    Male   "0.61"   "0.20"         "0.02"          "0.12"     "0.04"
##          relationship
## sex        Wife
##    Female "0.14"
##    Male   "0.00"
```

We compute the Cramer’s V value for the above pairs of variables. The Cramer’s V is symmetric, i.e. V(relationship, marital_status) = V(marital_status, relationship) so it follows that we need to compute only one of these values. Values we obtained for Cramer’s V indicate strength of association, similar to the those predicted by the Goodman and Kruskal’s tau:

```
assocstats(tab)$cramer
```

```
## [1] 0.6502624
```

```
tab1 <- xtabs(~ marital_status + relationship, data = adult_train)
assocstats(tab1)$cramer
```

```
## [1] 0.4871943
```

Goodness of Fit of the Model

Likelihood Ratio Test

Below we display the deviance of the intercept-only model and the deviance of the model `glm_model_wld` :

```
summary(glm_model_wld)$null.deviance
```

```
## [1] 33850.71
```

```
summary(glm_model_wld)$deviance
```

```
## [1] 19643.42
```

The null deviance shows how well the response is predicted by a model with nothing but an intercept compared to the saturated model. The deviance shows how well the observed response is predicted by a model with a given set of predictors compared to the saturated model.

We apply the likelihood ratio test to compare the goodness of fit of the intercept only model and the fitted model with explanatory variables - `glm_model_wld` . We test the null hypothesis that the null model fits the observed data well and explains about the same amount of variation in the response variable as the model `glm.model.wld` at the 0.05 significance level:

```
k <- length(glm_model_wld$coefficients)
D_M <- glm_model_wld$deviance
D_0 <- glm_model_wld$null.deviance

1 - pchisq(q = D_0 - D_M, df = k - 1)
```

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

The p-value is smaller than 0.05 meaning that we reject the null hypothesis that there is no significant difference between the intercept-only model and the model `glm.model.wld` (model with predictors). This means that at the 5% significance level the null model does not fit the observed data better than the multivariate model `glm.model.wld`.

Hosmer-Lemeshow Test

The Hosmer-Lemeshow test is a goodness of fit test for logistic regression models with ungrouped (individual) binary data. The idea of the Hosmer-Lemeshow test is to divide the data into subgroups based on the predicted probabilities π_i instead on the values of the explanatory variables.

We first extract the fitted probabilities.

```
head(glm_model_wld$fitted.values)
```

```
##          1          2          3          4          5          6
## 0.08854642 0.45746896 0.03016798 0.09477747 0.55772478 0.83319749
```

```
predicted_probs <- predict(glm_model_wld, type = "response")
head(predicted_probs)
```

```
##          1          2          3          4          5          6
## 0.08854642 0.45746896 0.03016798 0.09477747 0.55772478 0.83319749
```

Next, we need to transform the vector of predicted probabilities $\hat{\pi} = (\hat{\pi}_1, \hat{\pi}_2, \dots, \hat{\pi}_n)(n = 30162)$ into a binary vector of predicted responses (predicted income). We denote this binary vector with $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$. We construct the vector $\hat{y}$ in the following way:

$$\hat{y}_i = 1, \quad if \quad \hat{\pi}_i > 0.5,$$

and

$$\hat{y}_i = 0, \quad if \quad \hat{\pi}_i \leq 0.5,$$

where a value of 1 is equivalent to an yearly income of more than 50K, and a value of 0 means an income of less than 50K.

We take the vector of observed responses (which is a factor variable with two levels - " >50K" and " <=50K") and create a binary vector:

```
observed_values <- ifelse(adult_train$income  == " >50K", 1, 0)
```

Next we generate the vector of predicted probabilities – `predicted_probs`, and then we use it to create the binary vector `predicted_response`:

```
predicted_probs <- predict(glm_model_wld, type = "response")
predicted_response <- ifelse(predicted_probs > 0.5, 1, 0)
head(predicted_response, 20)
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  0  0  0  0  1  1  0  0  1  1  1  0  0  0  0  0  0  0  0  1
```

```
head(observed_values, 20)
```

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

We test the logistic model's accuracy on the training dataset, that is, we calculate the percentage of correctly predicted response values:

```
mean(observed_values == predicted_response)
```

```
## [1] 0.8488827
```

There is a 84.89% match between observed and predicted values of the dependent variable. In order to test the prediction accuracy of the fitted model, we need to test it on a new test data set.

Finally, we proceed with the Hosmer-Lemeshow test. We run the test with different number of groups. We take $g = 10, 20, 50, 100, 200, 300$ and 400. For small number of groups, we obtain very small p-values, meaning a poor fit of the model, whereas for bigger values of g, we obtain larger p-values indicating a good fit of the model. The Hosmer-Lemeshow test has some serious drawbacks, such as the demonstrated dependence of the results on the choice of groups, so we have to interpret the outcome of the test with caution. The final goal we would like to achieve determines the adequacy of the model, such as whether we want the constructed model to match the observed data as close as possible or predict new observations with high accuracy.Ddespite the lack of fit according to the Hosmer-Lemeshow test (in the case of relatively small number of groups), the fitted model `glm_model_wld` predicts new obervations with high accuracy.

```
hoslem.test(observed_values, predicted_response, g = 10)
```

```
## 
##  Hosmer and Lemeshow goodness of fit (GOF) test
## 
## data:  observed_values, predicted_response
## X-squared = 403.62, df = 8, p-value < 2.2e-16
```

```
hoslem.test(observed_values, predicted_response, g = 20)
```

```
## 
##  Hosmer and Lemeshow goodness of fit (GOF) test
## 
## data:  observed_values, predicted_response
## X-squared = 403.62, df = 18, p-value < 2.2e-16
```

```
hoslem.test(observed_values, predicted_response, g = 50)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  observed_values, predicted_response
## X-squared = 403.62, df = 48, p-value < 2.2e-16
```

```
hoslem.test(observed_values, predicted_response, g = 100)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  observed_values, predicted_response
## X-squared = 403.62, df = 98, p-value < 2.2e-16
```

```
hoslem.test(observed_values, predicted_response, g = 200)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  observed_values, predicted_response
## X-squared = 403.62, df = 198, p-value = 3.331e-16
```

```
hoslem.test(observed_values, predicted_response, g = 300)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  observed_values, predicted_response
## X-squared = 403.62, df = 298, p-value = 4.3e-05
```

```
hoslem.test(observed_values, predicted_response, g = 400)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  observed_values, predicted_response
## X-squared = 403.62, df = 398, p-value = 0.4122
```

Explanatory Variable Significance in the Model

Categorical Variable Significance

We will perform likelihood ratio tests. When we run `anova(glm.model.wld, test=”LRT“)`, the function sequentially compares nested models with increasing complexity against the full model, adding one predictor at a time. The comparisons are done with the help of a likelihood ratio test. The p-values of the tests are calculated using the chi-squared distribution:

```
anova(glm_model_wld, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: income
##
## Terms added sequentially (first to last)
##
##
##                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           30161      33851
## age             1   1738.5     30160      32112 < 2.2e-16 ***
## workclass       6    426.1     30154      31686 < 2.2e-16 ***
## education      15   3570.2     30139      28116 < 2.2e-16 ***
## marital_status  6   5091.4     30133      23024 < 2.2e-16 ***
## occupation     13    765.5     30120      22259 < 2.2e-16 ***
## relationship    5    199.5     30115      22059 < 2.2e-16 ***
## race            4     21.3     30111      22038 0.0002802 ***
## sex             1    165.7     30110      21872 < 2.2e-16 ***
## native_region   7     39.1     30103      21833 1.909e-06 ***
## hours_worked    4    418.6     30099      21415 < 2.2e-16 ***
## cap_gain        2   1473.3     30097      19941 < 2.2e-16 ***
## cap_loss        2    297.9     30095      19643 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All explanatory variables are significant and we should definitely keep all of the considered predictors in the model.

Estimated Model Parameter Significance

Let’s look at the significance of each level of the categorical predictors in the fitted model. We have a total of 67 model parameters:

```
length(glm_model_wld$coefficients)
```

```
## [1] 67
```

We have 12 predictors, most of which are categorical, and dummy variables are created to account for the factor levels of each categorical covariate making the number of model parameters greater than the number of predictors. For each categorical variabe with l levels, $l-1$ dummy variables are created and one level is chosen as the so-called “base” level.

```
summary(glm_model_wld)
```

```
##
## Call:
## glm(formula = new_form, family = binomial(link = "logit"), data = adult_train,
##     x = TRUE, y = TRUE)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6865  -0.5220  -0.1929  -0.0004   3.7388
##
## Coefficients:
##                                       Estimate Std. Error z value
## (Intercept)                           0.912013   0.677111   1.347
## age                                   0.026445   0.001703  15.524
## workclass Local-gov                  -0.639798   0.112229  -5.701
## workclass Private                    -0.452967   0.093103  -4.865
## workclass Self-emp-inc               -0.235796   0.123349  -1.912
## workclass Self-emp-not-inc           -0.872371   0.109915  -7.937
## workclass State-gov                  -0.754078   0.125033  -6.031
## workclass Without-pay               -13.178704 199.348575  -0.066
## education 11th                        0.103271   0.213698   0.483
## education 12th                        0.443576   0.273665   1.621
## education 1st-4th                    -0.463515   0.479806  -0.966
## education 5th-6th                    -0.424365   0.352032  -1.205
## education 7th-8th                    -0.504597   0.242052  -2.085
## education 9th                        -0.279893   0.269126  -1.040
## education Assoc-acdm                  1.286492   0.180217   7.139
## education Assoc-voc                   1.262370   0.173170   7.290
## education Bachelors                   1.903320   0.161212  11.806
## education Doctorate                   2.968042   0.223166  13.300
## education HS-grad                     0.764392   0.156795   4.875
## education Masters                     2.260993   0.172154  13.134
## education Preschool                 -12.512223  97.826168  -0.128
## education Prof-school                 2.901286   0.206750  14.033
## education Some-college                1.124583   0.159054   7.070
## marital_status Married-AF-spouse      2.906908   0.578697   5.023
## marital_status Married-civ-spouse     2.164671   0.273543   7.913
## marital_status Married-spouse-absent  0.009454   0.236523   0.040
## marital_status Never-married         -0.442596   0.087664  -5.049
## marital_status Separated             -0.072910   0.164084  -0.444
## marital_status Widowed                0.213334   0.156797   1.361
## occupation Armed-Forces              -1.381366   1.593366  -0.867
## occupation Craft-repair               0.036957   0.080421   0.460
## occupation Exec-managerial            0.768913   0.077460   9.927
## occupation Farming-fishing           -0.867546   0.138099  -6.282
## occupation Handlers-cleaners         -0.717162   0.144210  -4.973
## occupation Machine-op-inspct         -0.315015   0.102786  -3.065
## occupation Other-service             -0.790725   0.118154  -6.692
## occupation Priv-house-serv           -3.194762   1.316902  -2.426
## occupation Prof-specialty             0.502492   0.081997   6.128
## occupation Protective-serv            0.593617   0.125585   4.727
## occupation Sales                      0.257979   0.083010   3.108
## occupation Tech-support               0.659004   0.111493   5.911
## occupation Transport-moving          -0.070794   0.099467  -0.712
```

```
## relationship Not-in-family           0.583189   0.270463   2.156
## relationship Other-relative         -0.282343   0.245213  -1.151
## relationship Own-child              -0.614743   0.269757  -2.279
## relationship Unmarried               0.439557   0.285982   1.537
## relationship Wife                    1.380452   0.105033  13.143
## race Asian-Pac-Islander              0.727949   0.274759   2.649
## race Black                           0.528255   0.240855   2.193
## race Other                           0.183818   0.370332   0.496
## race White                           0.631602   0.230395   2.741
## sex Male                             0.839362   0.079512  10.556
## native_region Central-Asia          -0.061801   0.289253  -0.214
## native_region East-Asia              0.053454   0.262597   0.204
## native_region Europe-East            0.359557   0.335786   1.071
## native_region Europe-West            0.565042   0.194142   2.910
## native_region Outlying-US            0.280426   0.223323   1.256
## native_region South-America         -0.991009   0.469618  -2.110
## native_region United-States          0.411079   0.135673   3.030
## hours_worked between_45_and_60       0.438300   0.043696  10.031
## hours_worked between_60_and_80       0.411232   0.098212   4.187
## hours_worked less_than_40           -0.804617   0.062164 -12.943
## hours_worked more_than_80            0.274000   0.193600   1.415
## cap_gainLow                         -6.656558   0.509187 -13.073
## cap_gainMedium                      -4.779399   0.513828  -9.302
## cap_lossLow                         -0.790212   0.148968  -5.305
## cap_lossMedium                       0.807338   0.179135   4.507
##                                     Pr(>|z|)
## (Intercept)                          0.17801
## age                                  < 2e-16 ***
## workclass Local-gov                 1.19e-08 ***
## workclass Private                   1.14e-06 ***
## workclass Self-emp-inc               0.05592 .
## workclass Self-emp-not-inc          2.08e-15 ***
## workclass State-gov                 1.63e-09 ***
## workclass Without-pay                0.94729
## education 11th                       0.62891
## education 12th                       0.10505
## education 1st-4th                    0.33402
## education 5th-6th                    0.22802
## education 7th-8th                    0.03710 *
## education 9th                        0.29834
## education Assoc-acdm                9.43e-13 ***
## education Assoc-voc                 3.11e-13 ***
## education Bachelors                  < 2e-16 ***
## education Doctorate                  < 2e-16 ***
## education HS-grad                   1.09e-06 ***
## education Masters                    < 2e-16 ***
## education Preschool                  0.89823
## education Prof-school                < 2e-16 ***
## education Some-college              1.54e-12 ***
## marital_status Married-AF-spouse    5.08e-07 ***
## marital_status Married-civ-spouse   2.50e-15 ***
## marital_status Married-spouse-absent  0.96812
## marital_status Never-married        4.45e-07 ***
## marital_status Separated             0.65679
```

```
## marital_status Widowed               0.17365
## occupation Armed-Forces              0.38597
## occupation Craft-repair              0.64584
## occupation Exec-managerial            < 2e-16 ***
## occupation Farming-fishing           3.34e-10 ***
## occupation Handlers-cleaners         6.59e-07 ***
## occupation Machine-op-inspct          0.00218 **
## occupation Other-service             2.20e-11 ***
## occupation Priv-house-serv            0.01527 *
## occupation Prof-specialty            8.89e-10 ***
## occupation Protective-serv           2.28e-06 ***
## occupation Sales                      0.00188 **
## occupation Tech-support              3.41e-09 ***
## occupation Transport-moving           0.47663
## relationship Not-in-family            0.03106 *
## relationship Other-relative           0.24956
## relationship Own-child                0.02267 *
## relationship Unmarried                0.12429
## relationship Wife                     < 2e-16 ***
## race Asian-Pac-Islander               0.00806 **
## race Black                            0.02829 *
## race Other                            0.61964
## race White                            0.00612 **
## sex Male                              < 2e-16 ***
## native_region Central-Asia            0.83082
## native_region East-Asia               0.83870
## native_region Europe-East             0.28426
## native_region Europe-West             0.00361 **
## native_region Outlying-US             0.20923
## native_region South-America           0.03484 *
## native_region United-States           0.00245 **
## hours_worked between_45_and_60        < 2e-16 ***
## hours_worked between_60_and_80       2.82e-05 ***
## hours_worked less_than_40             < 2e-16 ***
## hours_worked more_than_80             0.15698
## cap_gainLow                           < 2e-16 ***
## cap_gainMedium                        < 2e-16 ***
## cap_lossLow                          1.13e-07 ***
## cap_lossMedium                       6.58e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33851  on 30161  degrees of freedom
## Residual deviance: 19643  on 30095  degrees of freedom
## AIC: 19777
##
## Number of Fisher Scoring iterations: 13
```

We can see the significant covariates and levels of categorical covariates for the log odds model based on the corresponding p-values. These p-values are obtained using the Wald test statistic to test the following hypotheses for the model coefficients:

$$H_0 : \hat{\beta}_j = 0$$

vs.

$$H_1 : \hat{\beta}_j \neq 0$$

for all $j = 0, 1, 2, \ldots, k$.

The Wald test is used to evaluate the statistical significance of each coefficient in the fitted logistic model, that is, the test checks the hypothesis that each individual coefficient is zero. If the coefficient of a category is not statistically significant, this does not imply that the whole categorical predictor is unimportant and should be removed from the model. The overall effect of the factor variable is tested by performing a likelihood ratio test as we showed earlier.

In the regression output above, the reported coefficients for each category of a factor variable measure the differences from the base level.

Consider the independent variable `education` . We notice that it is 18 times more likely for an individual to have an income of more than 50K per year if they have a doctorate degree compared to having only a 10th grade diploma. Below we list the levels of education:

```
levels(adult_train$education)
```

```
##  [1] " 10th"         " 11th"         " 12th"         " 1st-4th"
##  [5] " 5th-6th"      " 7th-8th"      " 9th"          " Assoc-acdm"
##  [9] " Assoc-voc"    " Bachelors"    " Doctorate"    " HS-grad"
## [13] " Masters"      " Preschool"    " Prof-school"  " Some-college"
```

```
summary(adult_train$education)
```

```
##          10th          11th          12th       1st-4th       5th-6th
##           820          1048           377           151           288
##       7th-8th           9th    Assoc-acdm     Assoc-voc     Bachelors
##           557           455          1008          1307          5044
##     Doctorate       HS-grad       Masters     Preschool   Prof-school
##           375          9840          1627            45           542
##  Some-college
##          6678
```

It is much more likely to earn more than 50K if one has a Bachelor or Masters degree (6.5 times and 9.3 times more likely, respectively) relative to the baseline 10th grade education. The same can be said for Prof-school, Assoc-acdm and Assoc-voc; the odds of having an income of more than 50K are 17 times, 3.5 times and again 3.5 times greater, respectively, compared to the reference category. Furthermore, people with college degree are 3 times more likely to earn more than 50K compared to people with 10th grade education. If an individual has a 1st-4th, 5th-6th, 7th-8th, or 9th grade education, their odds of being paid more than 50K a year are 1.75, 1.5, 1.64 and 1.3 times lower, respectively, than if they had 10th grade education. The result for Preschool is very extreme - the odds of earning more than 50K a year are 1/2.5×10−6=400000 times lower relative to the base level. This number makes sense but is also due to the fact that there are very few people in this category - only 45 people out of the

30162 in the sample. This can be seen also from the insignificant p-value for Preschool, which indicates that the covariate (dummy variable in this case) is not significant to the model at the 5% level. From the p-values we also notice that 1st-4th, 5th-6th, 9th, 11th and 12th are not significant at the 5% level.

Below we show the 95% confidence intervals for all estimated model parameters, along with the number of people belonging to each category of `workclass` :

```
summary(adult_train$workclass)
```

```
##      Federal-gov         Local-gov           Private      Self-emp-inc
##              943              2067             22286              1074
##  Self-emp-not-inc         State-gov       Without-pay
##             2499              1279                14
```

```
confint.default(glm_model_wld)
```

```
##                                                 2.5 %        97.5 %
## (Intercept)                               -0.41509979   2.239125127
## age                                        0.02310652   0.029784024
## workclass Local-gov                       -0.85976296  -0.419832799
## workclass Private                         -0.63544499  -0.270488585
## workclass Self-emp-inc                    -0.47755503   0.005963082
## workclass Self-emp-not-inc                -1.08780155  -0.656940906
## workclass State-gov                       -0.99913868  -0.509016492
## workclass Without-pay                   -403.89473116 377.537323679
## education 11th                            -0.31556969   0.522112466
## education 12th                            -0.09279792   0.979950016
## education 1st-4th                         -1.40391705   0.476887966
## education 5th-6th                         -1.11433504   0.265606020
## education 7th-8th                         -0.97900945  -0.030184060
## education 9th                             -0.80737004   0.247583531
## education Assoc-acdm                       0.93327309   1.639710055
## education Assoc-voc                        0.92296277   1.601777752
## education Bachelors                        1.58735101   2.219289091
## education Doctorate                        2.53064525   3.405438138
## education HS-grad                          0.45707934   1.071705232
## education Masters                          1.92357705   2.598408331
## education Preschool                     -204.24798949 179.223544403
## education Prof-school                      2.49606431   3.306508110
## education Some-college                     0.81284291   1.436322199
## marital_status Married-AF-spouse           1.77268264   4.041132412
## marital_status Married-civ-spouse          1.62853597   2.700805161
## marital_status Married-spouse-absent      -0.45412211   0.473029847
## marital_status Never-married              -0.61441429  -0.270778017
## marital_status Separated                  -0.39450774   0.248688007
## marital_status Widowed                    -0.09398235   0.520650336
## occupation Armed-Forces                   -4.50430629   1.741573819
## occupation Craft-repair                   -0.12066475   0.194578432
## occupation Exec-managerial                 0.61709407   0.920731967
## occupation Farming-fishing                -1.13821473  -0.596877145
## occupation Handlers-cleaners              -0.99980861  -0.434515416
## occupation Machine-op-inspct              -0.51647173  -0.113557794
## occupation Other-service                  -1.02230233  -0.559146876
## occupation Priv-house-serv                -5.77584236  -0.613681196
## occupation Prof-specialty                  0.34178139   0.663202094
## occupation Protective-serv                 0.34747512   0.839759424
## occupation Sales                           0.09528224   0.420676414
## occupation Tech-support                    0.44048151   0.877526032
## occupation Transport-moving               -0.26574528   0.124156841
## relationship Not-in-family                 0.05309058   1.113287804
## relationship Other-relative               -0.76295213   0.198265602
## relationship Own-child                    -1.14345833  -0.086028454
## relationship Unmarried                    -0.12095772   1.000070993
## relationship Wife                          1.17459236   1.586312455
## race Asian-Pac-Islander                    0.18943057   1.266467167
## race Black                                 0.05618808   1.000321822
## race Other                                -0.54202001   0.909655184
## race White                                 0.18003582   1.083167702
## sex Male                                   0.68352070   0.995203144
```

```
## native_region Central-Asia          -0.62872520   0.505124030
## native_region East-Asia             -0.46122618   0.568134019
## native_region Europe-East           -0.29857121   1.017685032
## native_region Europe-West            0.18453089   0.945552556
## native_region Outlying-US           -0.15727886   0.718130940
## native_region South-America         -1.91144283  -0.070574405
## native_region United-States          0.14516508   0.676992412
## hours_worked between_45_and_60       0.35265801   0.523941491
## hours_worked between_60_and_80       0.21873951   0.603724203
## hours_worked less_than_40           -0.92645756  -0.682777319
## hours_worked more_than_80           -0.10544879   0.653448407
## cap_gainLow                         -7.65454544  -5.658570680
## cap_gainMedium                      -5.78648311  -3.772314351
## cap_lossLow                         -1.08218361  -0.498240181
## cap_lossMedium                       0.45623954   1.158435515
```

The 95% confidence interval for the odds ratio comparing Without-pay versus Federal-gov ranges from $exp(-401.8) \rightarrow -\infty \quad to \quad exp(374.7) \rightarrow \infty$. This anomaly is due to the fact that there are a very small number of people - only 14, who belong to the category Without-pay, so this association should be interpreted with a lot of caution. The same can be said for the category (dummy variable) Preschool to which belong only 45 people from the study:

```
summary(adult_train$education)
```

```
##         10th         11th         12th      1st-4th      5th-6th
##          820         1048          377          151          288
##      7th-8th          9th   Assoc-acdm    Assoc-voc    Bachelors
##          557          455         1008         1307         5044
##    Doctorate      HS-grad      Masters    Preschool  Prof-school
##          375         9840         1627           45          542
##  Some-college
##         6678
```

Fitted Model Performance

Training Data Performance

We calculate the percentage of accurately guessed response variables using the training dataset `adult_train`. Given the vector of predicted probabilities $\hat{\boldsymbol{\pi}}$, we calculate a character vector $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$ such that

$$\hat{y}_i = ">50K", \quad if \quad \hat{\pi}_i > 0.5,$$

and

$$\hat{y}_i = "<=50K", \quad if \quad \hat{\pi}_i \leq 0.5$$

Since R codes the factor variables as numbers, the binary response variable is also being coded. Therefore when estimating a logistic regression model we need to know how the binary response variable is being modeled. By default R orders the factor levels alphabetically and the response level modeled in the logistic regression is the highest level. In our case the highest level in the `income` variable is " >50K“:

```
attributes(adult_train$income)
```

```
## $levels
## [1] " <=50K" " >50K"
##
## $class
## [1] "factor"
```

The response level being modeled is >50K, that is, when fitting the logistic regression model, the probability of the income being greater than 50K is calculated. We denote the vector with the predicted income values as `predicted_income_train` :

```
predicted_income_train <- ifelse(predicted_probs > 0.5, " >50K", " <=50K")
 predicted_income_train <- as.factor(predicted_income_train)
```

There is an 84.89% match between observed and predicted values of `income` :

```
mean(predicted_income_train == adult_train$income)
```

```
## [1] 0.8488827
```

We show the confusion matrix:

```
stat_log_train <- confusionMatrix(data = predicted_income_train, reference = adult_train
$income,
                                  positive = levels(adult_train$income)[2])
stat_log_train
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  <=50K  >50K
##      <=50K  21076  2980
##      >50K    1578  4528
##
##                Accuracy : 0.8489
##                  95% CI : (0.8448, 0.8529)
##     No Information Rate : 0.7511
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5689
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.6031
##             Specificity : 0.9303
##          Pos Pred Value : 0.7416
##          Neg Pred Value : 0.8761
##              Prevalence : 0.2489
##          Detection Rate : 0.1501
##    Detection Prevalence : 0.2024
##       Balanced Accuracy : 0.7667
##
##        'Positive' Class :  >50K
##
```

The sensitivity is the proportion of income values equal to >50K that are accurately identified, and the specificity is the proportion of income values equal to <=50K that are accurately identified. We see that the sensitivity is 60.31% and the specificity is 93.03%.

New Observations

We will test how well the fitted model predicts new observations. We will use the provided test dataset and, hence, the corresponding test data frame that we created, `adult_test` .

```
predicted_income_test <- predict(glm_model_wld, newdata = adult_test, type = "response")
predicted_income_test <- ifelse(predicted_income_test > 0.5, " >50K", " <=50K")
predicted_income_test <- as.factor(predicted_income_test)
```

Below we show the respective confusion matrix:

```
stat_log_test <- confusionMatrix(data = predicted_income_test, reference = adult_test$in
come,
                                 positive = levels(adult_test$income)[2])
stat_log_test
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  <=50K  >50K
##      <=50K  10559  1496
##      >50K     801  2204
##
##                Accuracy : 0.8475
##                  95% CI : (0.8416, 0.8532)
##     No Information Rate : 0.7543
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5607
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5957
##             Specificity : 0.9295
##          Pos Pred Value : 0.7334
##          Neg Pred Value : 0.8759
##              Prevalence : 0.2457
##          Detection Rate : 0.1463
##    Detection Prevalence : 0.1995
##       Balanced Accuracy : 0.7626
##
##        'Positive' Class :  >50K
##
```

The model predicts correctly 84.75% of the values of the dependent variable. We consider this a good predictive rate. On the test dataset the sensitivity is 59.57% and the specificity is 92.95%.

Below, the p-values indicate that all predictors in the model are significant and should be retained.

```
drop1(glm_model_wld, trace = TRUE, test = "LRT")
```

```
## Single term deletions
##
## Model:
## income ~ age + workclass + education + marital_status + occupation +
##     relationship + race + sex + native_region + hours_worked +
##     cap_gain + cap_loss
##                Df Deviance   AIC     LRT  Pr(>Chi)
## <none>               19643 19777
## age             1    19887 20019  243.31 < 2.2e-16 ***
## workclass       6    19746 19868  102.60 < 2.2e-16 ***
## education      15    20638 20742  995.09 < 2.2e-16 ***
## marital_status  6    19752 19874  108.32 < 2.2e-16 ***
## occupation     13    20181 20289  537.22 < 2.2e-16 ***
## relationship    5    19921 20045  277.62 < 2.2e-16 ***
## race            4    19656 19782   12.91 0.0117441 *
## sex             1    19759 19891  115.50 < 2.2e-16 ***
## native_region   7    19670 19790   26.94 0.0003414 ***
## hours_worked    4    20013 20139  369.30 < 2.2e-16 ***
## cap_gain        2    21196 21326 1552.37 < 2.2e-16 ***
## cap_loss        2    19941 20071  297.91 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```