Exercise 5

for Advanced Methods for Regression and Classification

Dzhamilia Kulikieva

20.11.2024

```
library(ROCit)

data(Loan)

# Convert Status to a numeric variable (0 and 1)
Loan$Status <- as.numeric(Loan$Status)
str(Loan)
```

```
## 'data.frame':    900 obs. of  9 variables:
##  $ Amount : num  67.6 23 54 24.3 43.2 ...
##  $ Term   : int  36 36 36 36 36 36 36 36 36 36 ...
##  $ IntRate: num  0.184 0.12 0.117 0.173 0.172 ...
##  $ ILR    : num  0.035 0.032 0.032 0.034 0.034 0.033 0.035 0.03 0.031 0.034 ...
##  $ EmpLen : Factor w/ 5 levels "A","B","C","D",..: 4 4 4 1 1 2 4 4 2 4 ...
##  $ Home   : Factor w/ 3 levels "MORTGAGE","OWN",..: 3 3 1 3 1 3 3 1 1 1 ...
##  $ Income : num  126400 30900 111900 66000 71900 ...
##  $ Status : num  1 1 2 2 1 2 2 2 2 2 ...
##  $ Score  : num  201 180 162 197 203 ...
```

```
head(Loan)
```

```
##   Amount Term IntRate   ILR EmpLen     Home Income Status    Score
## 1  67.57   36  0.1838 0.035      D     RENT 126400      1 200.9581
## 2  22.97   36  0.1198 0.032      D     RENT  30900      1 179.6058
## 3  54.05   36  0.1166 0.032      D MORTGAGE 111900      2 161.7622
## 4  24.32   36  0.1733 0.034      A     RENT  66000      2 196.6619
## 5  43.24   36  0.1723 0.034      A MORTGAGE  71900      1 203.4912
## 6  16.22   36  0.1355 0.033      B     RENT  27614      2 186.3070
```

1. Building linear regression model for the binary variable Status

```
# Split the data
set.seed(1234)
train_indices <- sample(1:nrow(Loan), size = 2/3*nrow(Loan))
train_data <- Loan[train_indices, ]
test_data <- Loan[-train_indices, ]

# Fit a linear regression model on the training data
lm_model <- lm(Status ~., data = train_data)
summary(lm_model)
```

```
##
## Call:
## lm(formula = Status ~ ., data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97297  0.02554  0.11413  0.18598  0.39224
##
## Coefficients: (2 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.402e-01  1.268e+00  -0.426  0.67019
## Amount      -1.498e-03  7.413e-04  -2.021  0.04369 *
## Term                NA         NA      NA       NA
## IntRate     -6.570e+00  2.217e+00  -2.964  0.00316 **
## ILR          1.022e+02  4.770e+01   2.143  0.03256 *
## EmpLenB      4.859e-03  4.212e-02   0.115  0.90819
## EmpLenC     -3.084e-02  4.565e-02  -0.676  0.49951
## EmpLenD      1.213e-03  3.911e-02   0.031  0.97526
## EmpLenU     -2.867e-02  6.673e-02  -0.430  0.66760
## HomeOWN      2.755e-02  4.727e-02   0.583  0.56015
## HomeRENT    -4.476e-02  3.099e-02  -1.444  0.14920
## Income       5.883e-07  3.406e-07   1.727  0.08468 .
## Score               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.341 on 589 degrees of freedom
## Multiple R-squared:  0.07046,    Adjusted R-squared:  0.05468
## F-statistic: 4.465 on 10 and 589 DF,  p-value: 4.347e-06
```

2. Inspecting the outcome of summary()

"Term" and "Score" have NA values for both coefficients and standard errors. This indicates that these variables are singular or highly collinear with other variables, meaning they do not provide unique information for the model.

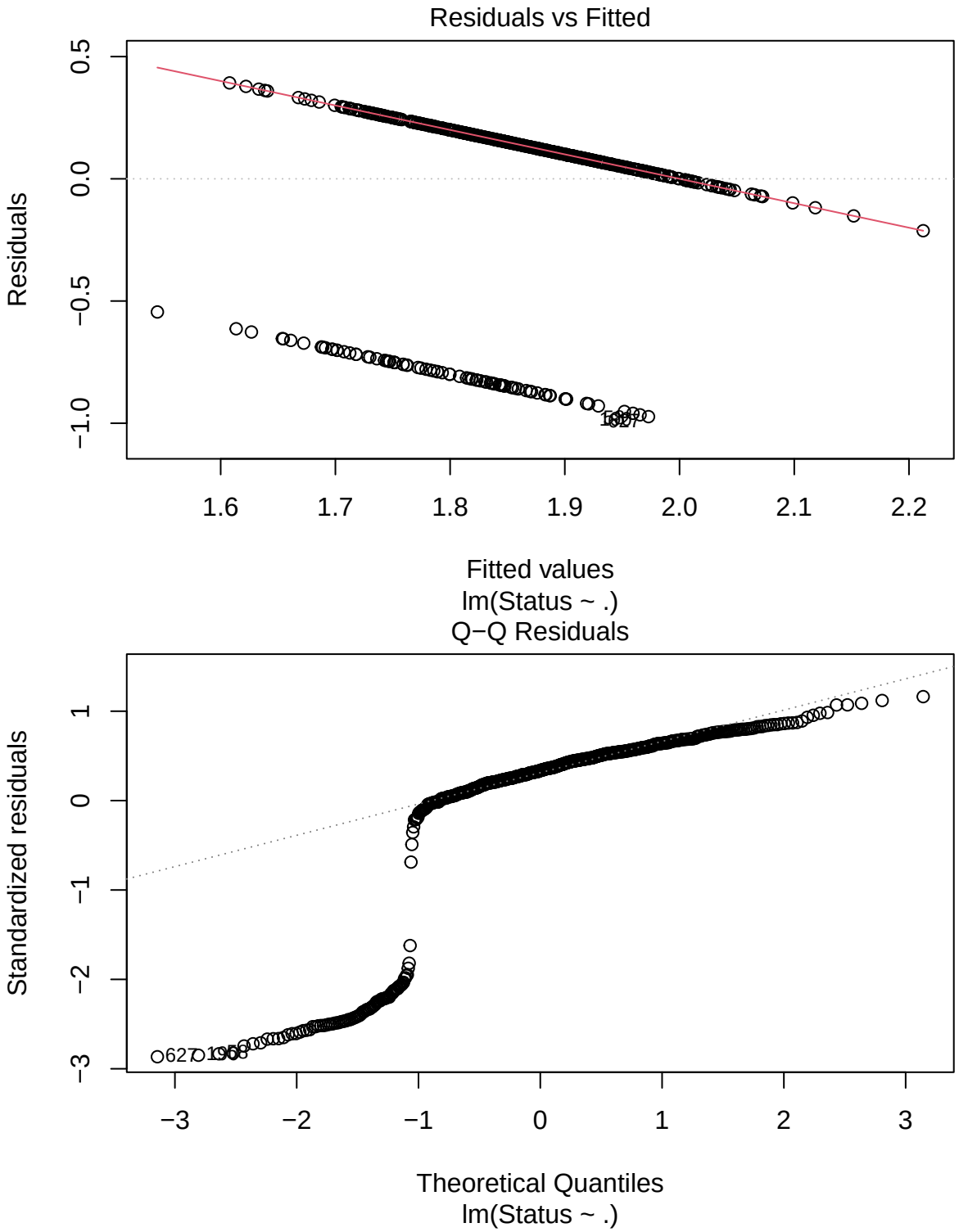
Predictors like EmpLenB, EmpLenC, EmpLenD, EmpLenU, and HomeOWN have high p-values, meaning they do not significantly contribute to the model.

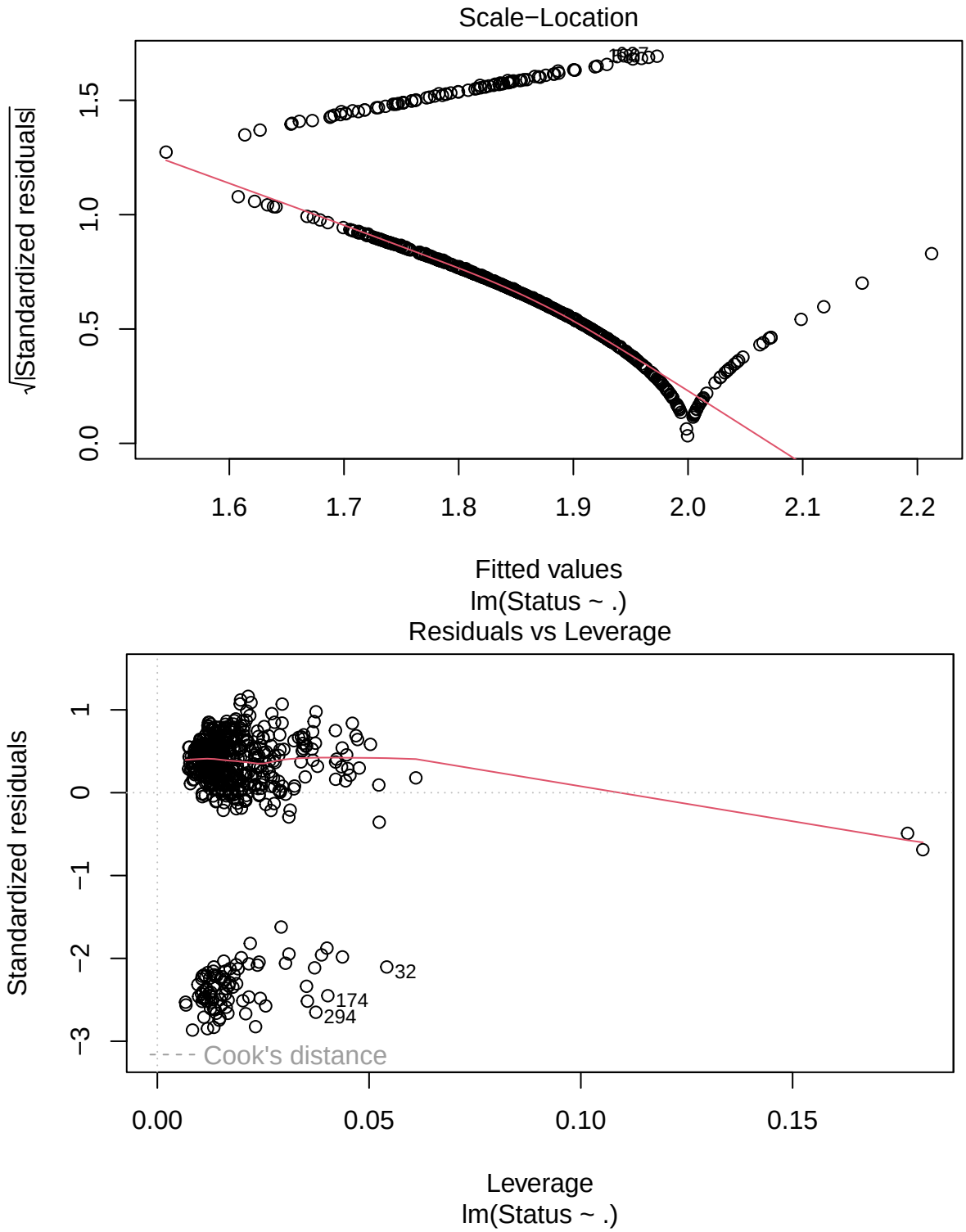
The model has a low R-squared value of 0.07046, meaning that only about 7% of the variance in the target variable Status is explained by the model. This suggests that the model is not performing well and could be improved.

Predictores with small p-value like Amount, Income, and ILR show whether the variables are significant for the model. Although they may have very different scales, that can negatively impact the performance of some algorithms.

3. Inspecting the plot() of LM

```
plot(lm_model)
```


Residuals vs Fitted
Residuals
0.5
0.0
−0.5
−1.0
1.6
1.7
1.8
1.9
2.0
2.1
2.2
Fitted values
lm(Status ~ .)
Q−Q Residuals
Standardized residuals
1
0
−1
−2
−3
627
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(Status ~ .)



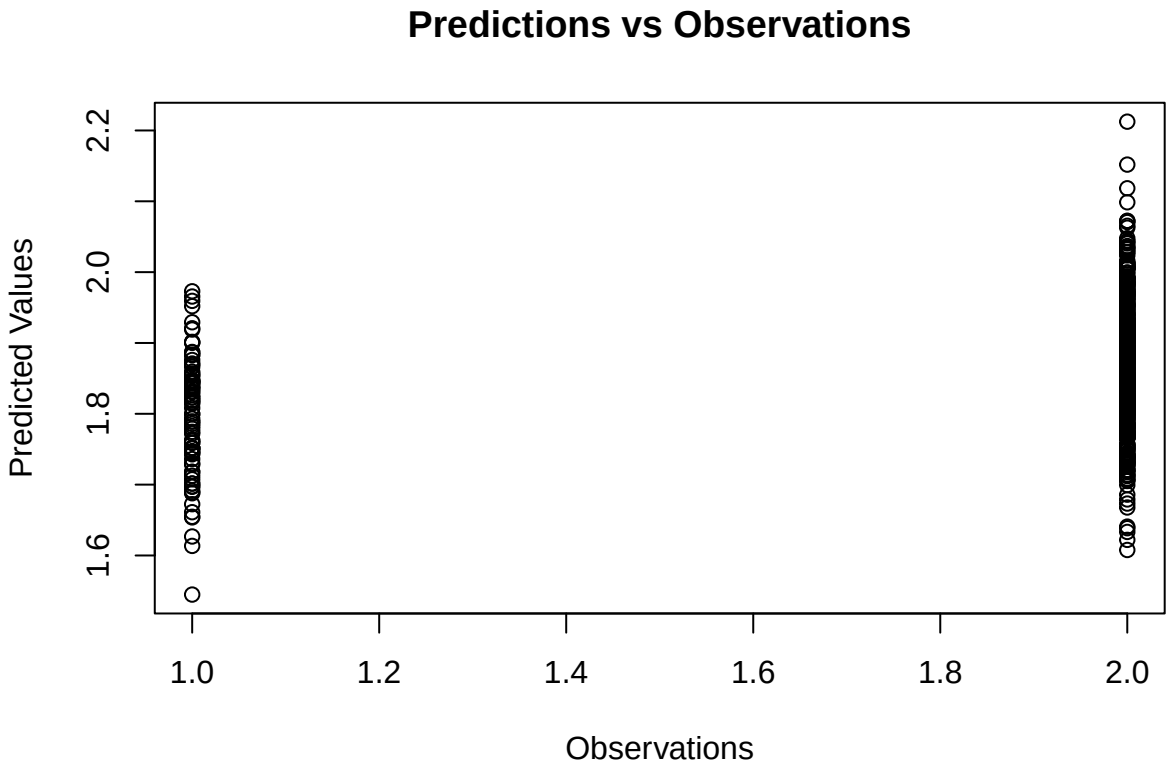
In each plot we see two parallel lines of points (one below zero), that indicates an issue with using linear regression for a binary target variable. Linear regression assumes a continuous dependent variable. For binary classification tasks, LR tries to fit the data as if the target were continuous, resulting in residuals clustering into two distinct parallel lines(Residuals vs Fitted) around values close to 0 and 1.

The patterns in the other diagnostic plots also all point to the same conclusion: LR is not the right tool for a binary target variable like Status. The model fails to meet critical assumptions (normality, homoscedasticity,

and linearity), and the unusual plot shapes are direct consequences of trying to apply a continuous model to categorical data.

4. Prediction of the responce value for the train_data

```
train_predictions <- predict(lm_model, newdata = train_data)
plot(train_data$Status, train_predictions, main = "Predictions vs Observations", xlab = "Observations",
```



Status is a binary variable that takes two values (1 and 2). These values are represented as categories, so all points with the same Status values line up in vertical lines on the graph.

For cutoff values, I would try setting a threshold (e.g., 0.5) and see how it affects the predictions.

5. Confusion Matrix for the Training Set.

```
# Apply cutoff (e.g., 0.5)
predicted_class <- ifelse(train_predictions > 0.5, 1, 0)

# Confusion matrix
table(Actual = train_data$Status, Predicted = predicted_class)
```

```
##       Predicted
## Actual   1
##      1  86
##      2 514
```

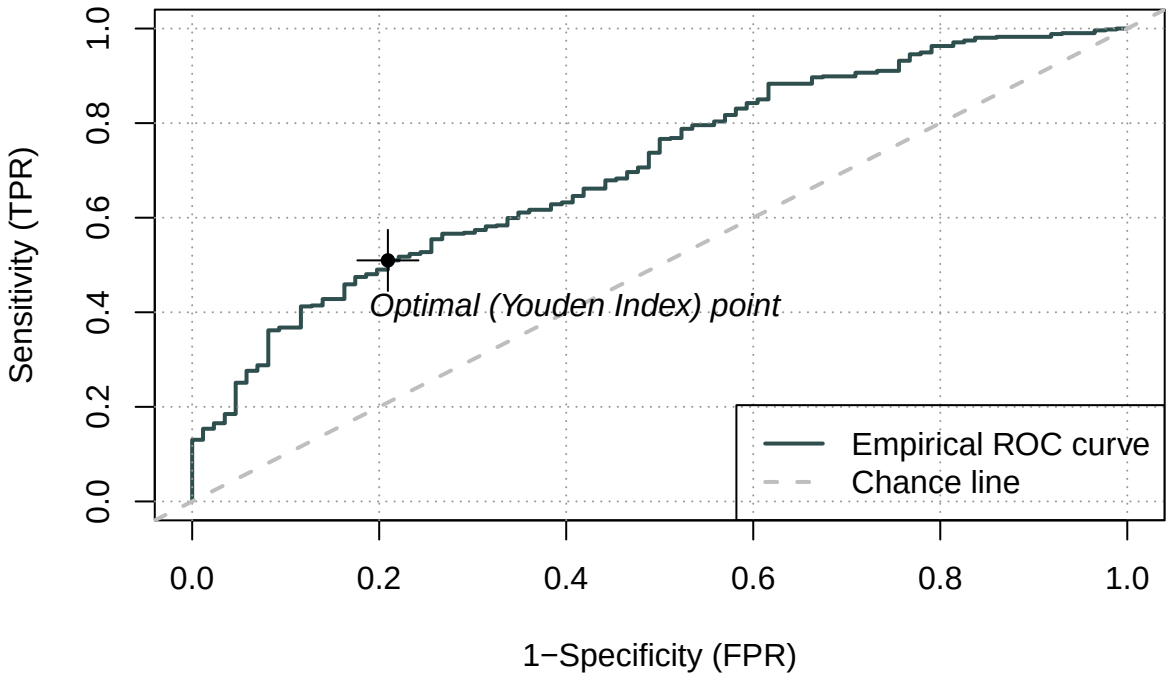
Actual 1, Predicted 1 (86): TP: The model correctly predicted 86 instances as belonging to class 1 when they truly belong to class 1. Actual 2, Predicted 1 (514): FP: The model incorrectly predicted 514 instances as belonging to class 1 when they actually belong to class 2.

6. Evaluating classifier and plotting the result

```
roc_result <- rocit(score = train_predictions, class = train_data$Status)
summary(roc_result)
```

```
##
##  Method used: empirical
##  Number of positive(s): 514
##  Number of negative(s): 86
##  Area under curve: 0.7061
```

```
plot(roc_result)
```



Area Under the Curve (AUC) value is a key indicator of classifier quality. AUC = 0.7061 suggests that the model has moderate performance. It's better than random guessing (since it's above 0.5), but it is not performing excellently.

ROC curve on the graph is above the diagonal line, meaning the model is performing better than random guessing. With x ~ 0.2 and y~ 0.5 is the point that is farthest from the diagonal and corresponds to the highest value of the Youden Index on the ROC curve, then this is the optimal cutoff for classification.