The Insurance Company - Base Model

Atanu Choudhury

2021-05-19

Contents

1 Introduction

In the previous section of Data Exploration there are quite a few things that have been observed: * Prediction of the class involves classification * The target class has imbalance, observed in both training and test * Quite a few predictors have high correlation, while some are skewed or heavy tailed with outliers

In this section, we look at approaches to build our model. As the problem of prediction of two classes it is a simple classification problem, we will implement the logistic regression learning algorithm. But in order to do so we have to initially look at the predictors to make our model simpler as only accuracy is not the desired attribute of our model, simplicity of the model is also important as it would be implemented in practice. Therefore, we make a trade off with accuracy and simplicity in the model. In order to achieve a more explanable model it's very important to focus on the predictors which when targeted as part of marketing would have a customer buying the policy. In order to achieve the best result we look at different ways of selecting the correct features, as most classification learning algorithms need the right set of predictors for a lower mis-classification rate.

```
source("read_data.R")
source("prediction_summary.R")
```

2 Logistic Regression model with all the predictors

2.1 Fitting the model

```
glm_model_0 <- glm(V86~.,
                   data = caravan_data,
                   family = binomial)
summary(glm_model_0)
```

```
##
## Call:
## glm(formula = V86 ~ ., family = binomial, data = caravan_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7047  -0.3711  -0.2450  -0.1588   3.2916
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.542e+02  1.116e+04   0.023  0.98183
## V1           6.580e-02  4.624e-02   1.423  0.15468
## V2          -1.832e-01  1.927e-01  -0.951  0.34157
## V3          -2.696e-02  1.399e-01  -0.193  0.84723
## V4           2.096e-01  1.016e-01   2.063  0.03911 *
## V5          -2.767e-01  2.076e-01  -1.333  0.18247
## V6          -1.142e-01  1.069e-01  -1.068  0.28535
## V7          -1.910e-02  1.177e-01  -0.162  0.87112
## V8          -1.618e-02  1.055e-01  -0.153  0.87818
## V9          -6.817e-02  1.113e-01  -0.612  0.54024
## V10          2.310e-01  1.566e-01   1.475  0.14031
## V11          8.509e-02  1.466e-01   0.580  0.56169
## V12          1.467e-01  1.562e-01   0.939  0.34759
## V13         -8.291e-02  1.311e-01  -0.633  0.52702
## V14         -1.154e-01  1.337e-01  -0.863  0.38813
## V15         -8.140e-02  1.417e-01  -0.575  0.56561
## V16          9.717e-04  1.311e-01   0.007  0.99408
## V17         -9.077e-02  1.365e-01  -0.665  0.50605
## V18         -1.994e-01  1.376e-01  -1.449  0.14740
## V19          8.883e-02  9.349e-02   0.950  0.34204
## V20          3.918e-02  9.897e-02   0.396  0.69219
## V21         -1.169e-01  1.104e-01  -1.059  0.28951
## V22          1.353e-01  9.191e-02   1.472  0.14106
## V23          3.976e-02  9.067e-02   0.438  0.66104
## V24          9.954e-02  9.143e-02   1.089  0.27628
## V25          2.690e-02  1.035e-01   0.260  0.79502
## V26         -8.801e-03  1.011e-01  -0.087  0.93064
## V27          1.200e-02  9.081e-02   0.132  0.89485
## V28          9.016e-02  9.958e-02   0.905  0.36527
## V29         -2.468e-02  9.724e-02  -0.254  0.79967
## V30         -1.472e+01  8.140e+02  -0.018  0.98557
## V31         -1.469e+01  8.140e+02  -0.018  0.98561
## V32          1.819e-01  1.514e-01   1.202  0.22953
## V33          1.507e-01  1.371e-01   1.099  0.27162
## V34          9.325e-02  1.436e-01   0.649  0.51603
## V35         -1.445e+01  9.359e+02  -0.015  0.98768
## V36         -1.451e+01  9.359e+02  -0.016  0.98763
## V37          1.181e-01  1.006e-01   1.174  0.24039
## V38          1.366e-01  9.650e-02   1.415  0.15694
## V39          1.009e-01  9.667e-02   1.043  0.29678
## V40          1.144e-01  1.027e-01   1.114  0.26513
## V41         -1.607e-01  1.449e-01  -1.109  0.26738
## V42          9.214e-02  9.945e-02   0.927  0.35417
```

```
## V43           6.856e-02  4.642e-02   1.477  0.13966
## V44           5.954e-01  3.901e-01   1.526  0.12693
## V45          -2.757e-01  4.635e-01  -0.595  0.55196
## V46          -4.405e-01  1.035e+00  -0.425  0.67052
## V47           2.306e-01  4.199e-02   5.491 4.01e-08 ***
## V48           1.215e+01  4.029e+02   0.030  0.97595
## V49          -8.101e-02  1.147e-01  -0.706  0.48006
## V50          -2.106e+00  2.557e+03  -0.001  0.99934
## V51           1.014e+00  9.371e-01   1.082  0.27917
## V52           7.229e-01  4.278e-01   1.690  0.09107 .
## V53          -5.525e+00  4.805e+03  -0.001  0.99908
## V54           2.170e-01  4.865e-01   0.446  0.65559
## V55          -2.382e-01  1.170e-01  -2.036  0.04173 *
## V56          -4.523e-01  2.094e+00  -0.216  0.82901
## V57           1.444e+00  1.029e+00   1.404  0.16033
## V58           8.239e-01  5.943e-01   1.386  0.16565
## V59           2.401e-01  7.714e-02   3.113  0.00185 **
## V60          -8.658e+00  3.261e+03  -0.003  0.99788
## V61          -1.886e-01  3.259e-01  -0.579  0.56289
## V62           3.664e-01  8.325e-01   0.440  0.65985
## V63          -1.068e+00  8.764e-01  -1.219  0.22301
## V64          -1.676e-01  3.321e-01  -0.505  0.61373
## V65          -9.293e-01  7.802e-01  -1.191  0.23364
## V66           4.197e-01  1.082e+00   0.388  0.69824
## V67           2.762e-01  3.528e+00   0.078  0.93758
## V68          -3.902e-02  1.772e-01  -0.220  0.82566
## V69          -7.298e+01  2.417e+03  -0.030  0.97591
## V70           2.418e-01  3.772e-01   0.641  0.52142
## V71          -4.490e+00  1.078e+04   0.000  0.99967
## V72          -1.351e+00  1.687e+00  -0.801  0.42322
## V73          -2.376e+00  1.524e+00  -1.559  0.11899
## V74          -8.749e-01  9.682e+03   0.000  0.99993
## V75          -1.060e+00  1.549e+00  -0.684  0.49367
## V76           4.789e-01  2.245e-01   2.133  0.03291 *
## V77           3.997e-01  4.329e+00   0.092  0.92644
## V78          -3.163e+00  2.706e+00  -1.169  0.24247
## V79          -3.212e+00  3.433e+00  -0.936  0.34939
## V80          -4.118e-01  2.787e-01  -1.477  0.13956
## V81           1.047e+01  3.261e+03   0.003  0.99744
## V82           2.516e+00  1.010e+00   2.490  0.01276 *
## V83           2.318e-01  5.699e-01   0.407  0.68420
## V84           1.947e+00  1.412e+00   1.378  0.16812
## V85           1.078e+00  1.103e+00   0.977  0.32870
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2635.5  on 5821  degrees of freedom
## Residual deviance: 2243.5  on 5736  degrees of freedom
## AIC: 2415.5
##
## Number of Fisher Scoring iterations: 17
```

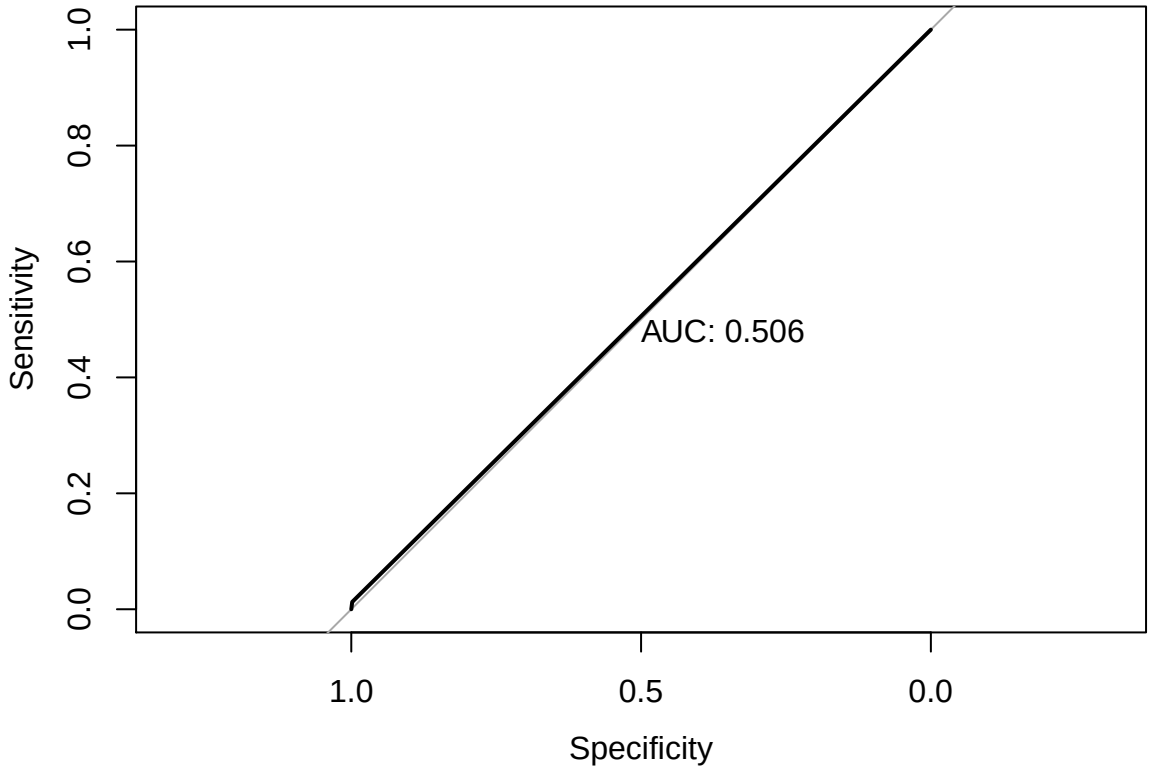
2.2 Interpreting the model

* The deviances have a higher number which indicates that the model is a bad fit for the data, as it uses the ordinary least squares by fitting a model based on its maximum likelihood.
* The significance level of the predictors shows v47 as most significant, and a few more as comparatively less significant than 47 but more than most (V4,V52,V55,V59,V76,V82) based on their p-values
* The Z-value for the variables do not show a distinctively higher magnitude to indicate the importance of a variable
* The coeffecient estimates gives how much change in the coeffecient of a predictor is need to make an unit change in the target. In a logistic regression the coeffecient is based on the log odds of the outcome for a unit change in the variable.
* The higher AIC value is evident as it penalises a more complex model, which is the model in our case
* This model took 17 iterations to converge as per the Fisher scoring.

2.3 Evaluating model

The accuracy of a classifier is based on the number of classifications. It is important to note the accuracy of the models being built.

```
pred_sum_0 <- prediction_summary(glm_model_0,caravan_test, 0.5, "V86")
```




```
pred_sum_0$auc_value
```

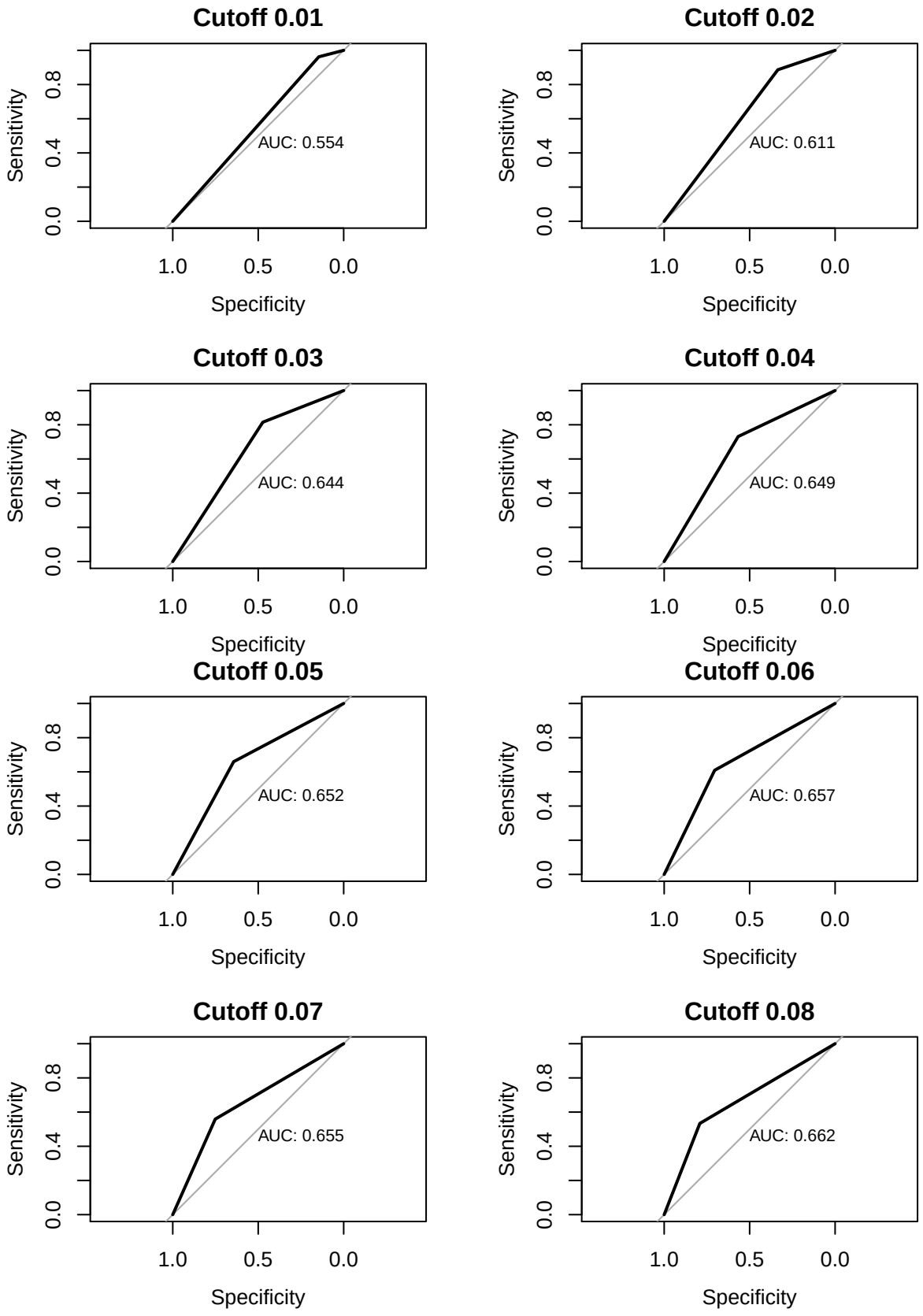
```
## Area under the curve: 0.5055
```

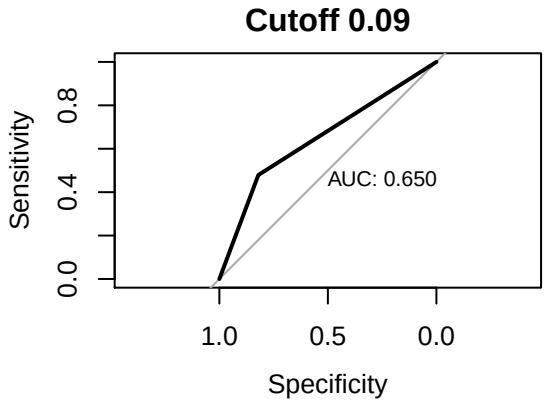
2.3.1 Interpreting the prediction results:

```
* There is a high sensitivity and a low specificity, which imply that the
   True positive rate is very high as the model was able to determine the
   non-caravan insurance buyers, but was considerably poor in identifiying
    the ones that would have bought the policy
* The accuracy gave us almost 94% as there are more 0s in the data than 1s
   . But for our assignment it is important to identify the customers who
   are more likley to buy the policy.
* As we set the cutoff to about 50%, we assume that the likelihood of the
   classes are even. The prior probablity from the data is around 5-6%.
   This is a thing to be considered.
* Looking at the AUC value which stands at 0.505, the model is not much
   effective in terms of prediction. As AUC imples the area under the
   curve for a classifier, it would be the criteria of judgin our model. A
    higher AUC value would mean a better classifier model.
```

2.4 Find the cutoff for the maximum AUC

```
par(mfrow = c(2, 2))
for (cutoff in c(0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09)){
    glm_prob <- predict.glm(glm_model_0,caravan_test[,-86],type="response"
       )
    glm_predict <- rep(0,nrow(caravan_test))
    glm_predict[glm_prob>cutoff] <- 1
    roc_obj <- roc(caravan_test$V86,glm_predict)
    plot(roc_obj, print.auc=TRUE,main=paste("Cutoff",cutoff))
}
```


Cutoff 0.01
AUC: 0.554
Cutoff 0.02
AUC: 0.611
Cutoff 0.03
AUC: 0.644
Cutoff 0.04
AUC: 0.649
Cutoff 0.05
AUC: 0.652
Cutoff 0.06
AUC: 0.657
Cutoff 0.07
AUC: 0.655
Cutoff 0.08
AUC: 0.662
Sensitivity
Specificity
0.0
0.4
0.8
1.0
0.5
0.0

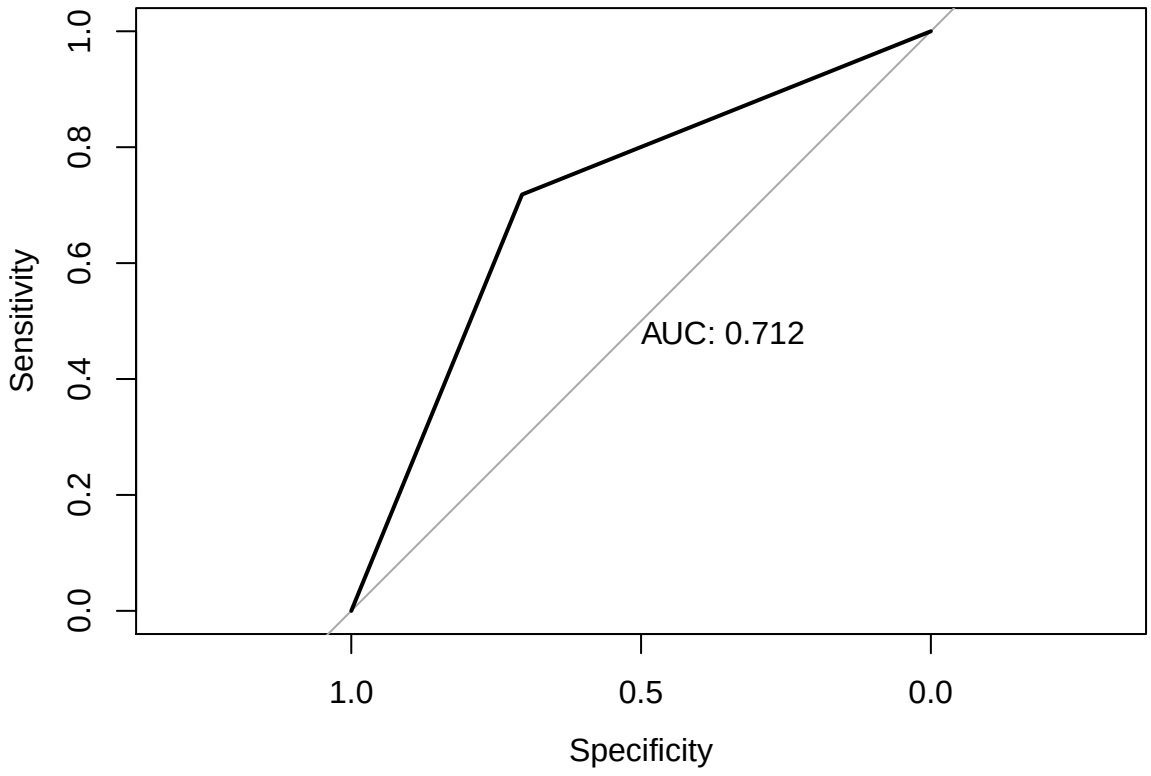


2.4.1 Interpreting the above graphs:

```
* We look at the different AUC curves for different cutoff boundaries for
   the clas probabilities. This helps us to determine that the highest
   area under curve lies for the cutoff of 0.06 which is surprisingly
   close to the prior probability value
* This may seem unusual but its important to understand the importance of
   the boundary in our dataset. As the number of caravan policy buyers are
    very low, it can be thought of the number of people having cancer in
   the whole population. To predict the probable persons having cancer the
    true prior probability affects our knowledge of predicting the actual
   cancer patients, in our case the customers buying caravan policy.
```

2.4.1.1 Training prediction summary

```
pred_sum_0 <- prediction_summary(glm_model_0,caravan_data, .06, "V86")
```

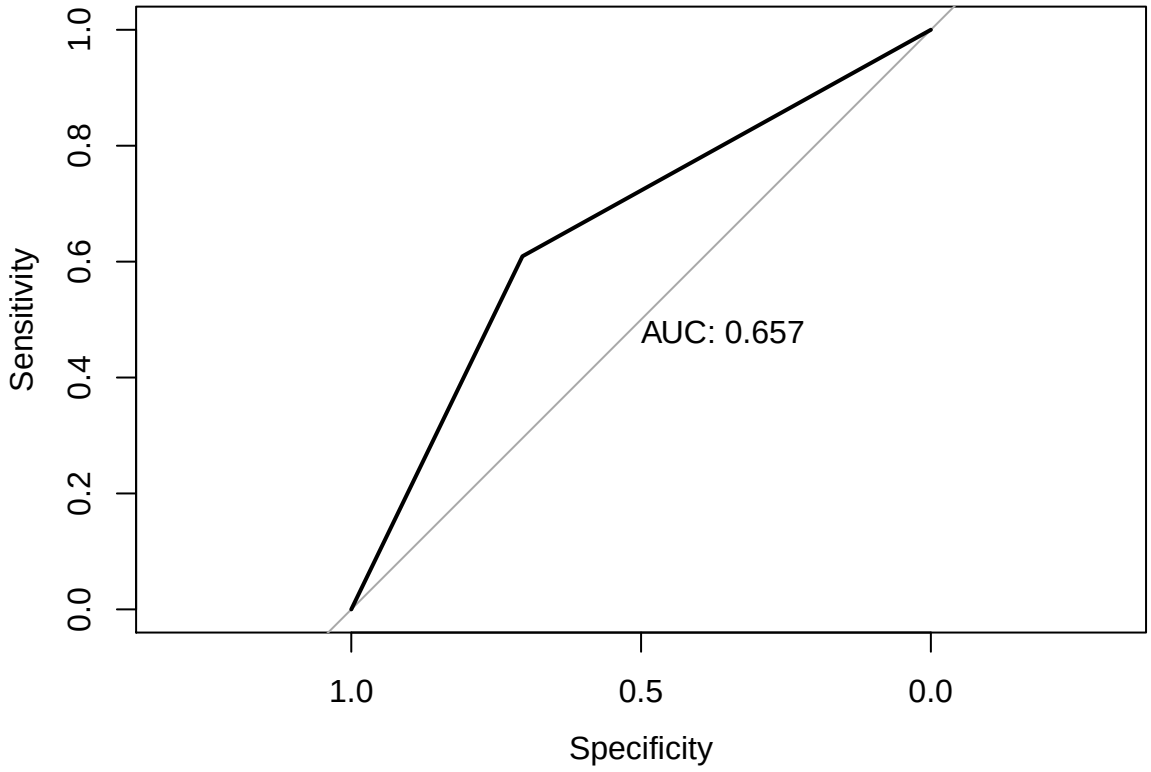


```
pred_sum_0$auc_value
```

```
## Area under the curve: 0.7119
```

2.4.1.2 Test data prediction summary

```
pred_sum_0 <- prediction_summary(glm_model_0,caravan_test, .06, "V86")
```




```
pred_sum_0$auc_value
```

```
## Area under the curve: 0.657
```

We see above that the specifity and sensitivity have similar values but the accuracy has drastically dropped as it was previously biased towards the more prevalent 0s and now it is more balanced towards the 1s. Also to note we have a higher AUC value than before, which determines that this classifier is better at classification using all the predictors and choosing the correct cutoff boundary.