Assignment_10.2_HillZach

Zach Hill

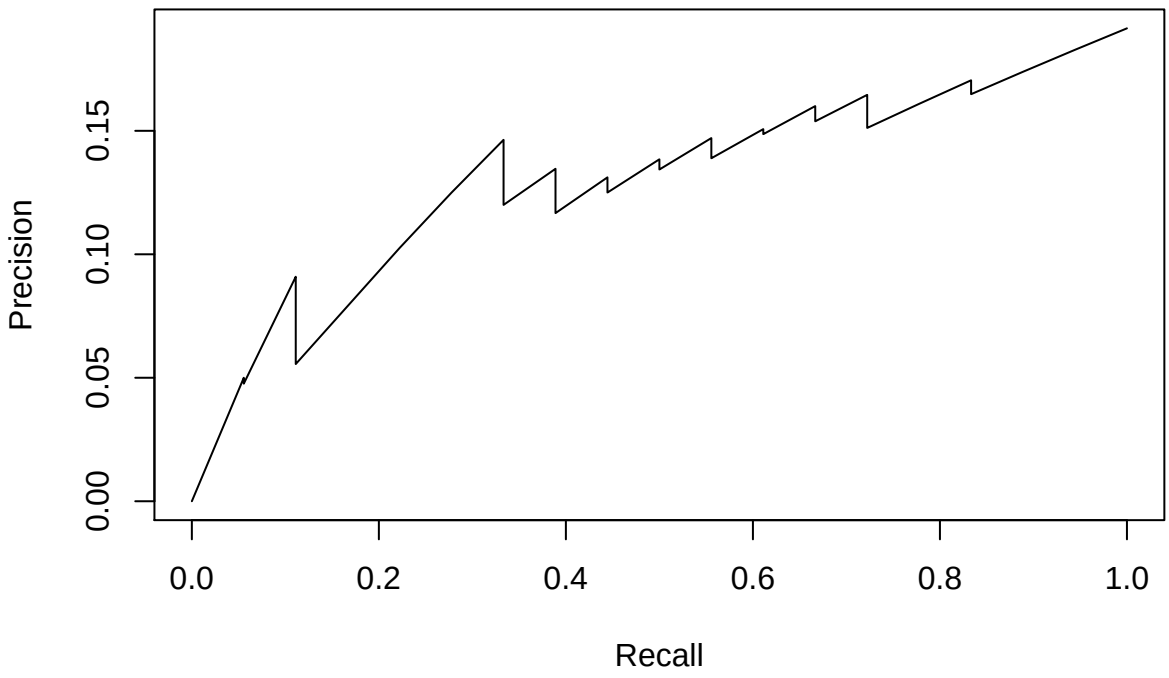
May 19, 2019

Creating a training and testing dataset

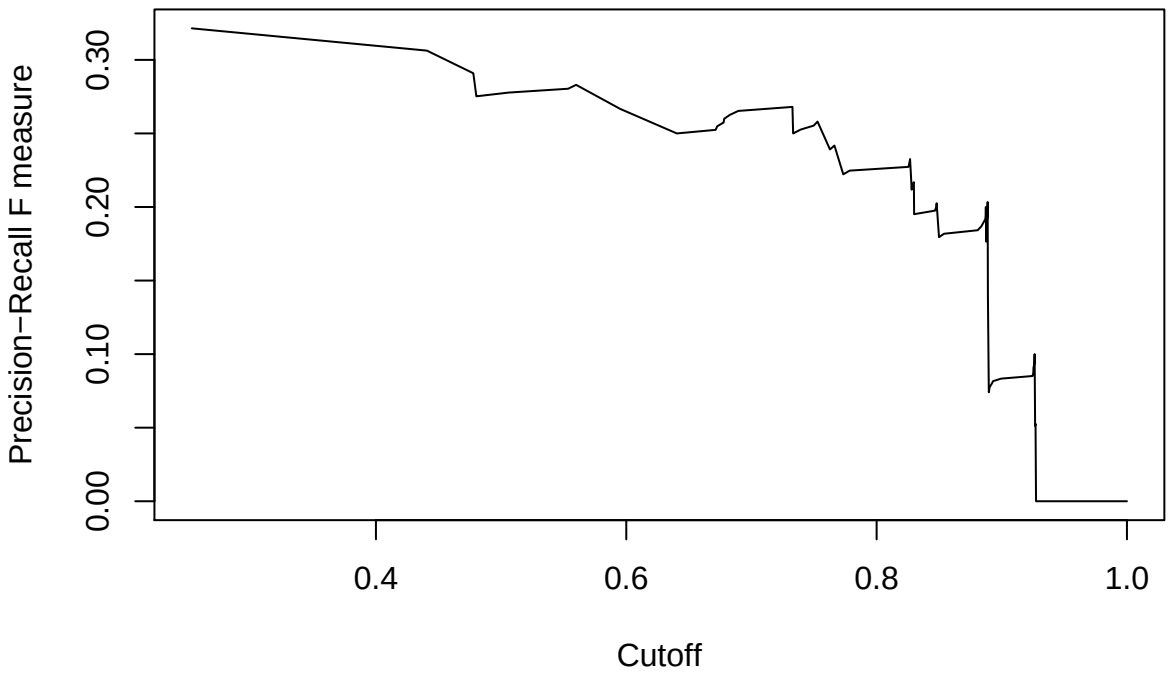
```
data_sample <- sample(seq_len(nrow(data)), size = floor(.8 * nrow(data)))
data_train <- data[data_sample, 1:17]
data_test <- data[-data_sample, 1:17]

# glm_train <- glm(Risk1Yr ~ ., data = data_train, family = binomial)
# glm_test <- glm(Risk1Yr ~ ., data = data_test, family = binomial)

glm_data <- glm(formula = Risk1Yr ~ DGN + PRE5 + PRE9 + PRE11 + PRE14 + PRE17 + PRE30, data = data, fami
glm_train <- glm(formula = Risk1Yr ~ DGN + PRE5 + PRE9 + PRE11 + PRE14 + PRE17 + PRE30, data = data_trai
glm_test <- glm(formula = Risk1Yr ~ DGN + PRE5 + PRE9 + PRE11 + PRE14 + PRE17 + PRE30, data = data_test

data_train$predicted.probabilities <- fitted(glm_train)
data_train$standardized.residuals <- rstandard(glm_train)
data_train$studentized.residuals <- rstudent(glm_train)
data_train$dfbeta <- dfbeta(glm_train)
data_train$dffit <- dffits(glm_train)
data_train$leverage <- hatvalues(glm_train)

data_test$predicted.probabilities <- fitted(glm_test)
data_test$standardized.residuals <- rstandard(glm_test)
data_test$studentized.residuals <- rstudent(glm_test)
data_test$dfbeta <- dfbeta(glm_test)
data_test$dffit <- dffits(glm_test)
data_test$leverage <- hatvalues(glm_test)
```

A: Precision, recall and F1 scores

```
predictions <- predict(glm_data, data_test, type = "response")
pred <- prediction(predictions, data_test$Risk1Yr)

PR.perf <- performance(pred, "prec", "rec")
plot(PR.perf)
```

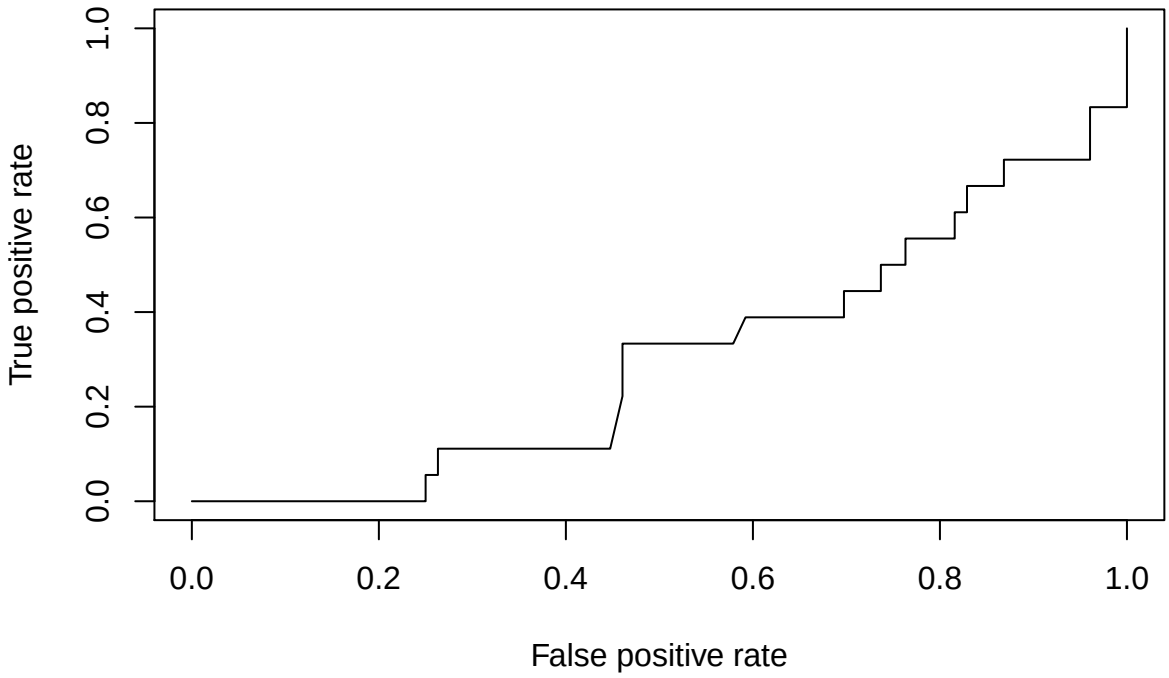



```
F1.perf <- performance(pred, "f")
plot(F1.perf)
```



B: ROC and AUC

```
ROC.perf <- performance(pred, "tpr", "fpr")
plot(ROC.perf)
```



The Area under the curve:

```
AUC.perf <- performance(pred, "auc")
AUC.perf <- as.numeric(AUC.perf@y.values)
AUC.perf
```

```
## [1] 0.3022661
```

C: An accurate model?

Horton described a model such as this as being accurate without any applicability. Basically it looks good on paper but has no function in the real world. This could be caused by situations where a feature always reports negative and is right due to the population having a very low probability of testing true for the feature in question. The AUC would tell us the model doesn't fit however as 53% is close to having no test for the feature in the first place.