Use.All in KNN Probabilities

June 1, 2020

1 Packages

[2]:
```
library(mdsr)
library(class)
library(ROCR)
library(gridExtra)
```

2 Init

[3]:
```
census <- read.csv(
  "http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data",
  header = FALSE)
names(census) <- c("age", "workclass", "fnlwgt", "education",
                   "education.num", "marital.status", "occupation",␣
 ↪"relationship",
                   "race", "sex", "capital.gain", "capital.loss", "hours.per.
 ↪week",
                   "native.country", "income")
set.seed(364)
n <- nrow(census)
test_idx <- sample.int(n, size = round(0.2 * n))
train <- census[-test_idx,]
test <- census[test_idx,]

form <- as.formula("income ~ age + workclass + education + marital.status +
  occupation + relationship + race + sex + capital.gain + capital.loss +
  hours.per.week")

train_q <- train %>%
  select(age, education.num, capital.gain, capital.loss, hours.per.week)
test_q <- test %>%
  select(age, education.num, capital.gain, capital.loss, hours.per.week)
```

3 Use.All = TRUE

`use.all` is used to determine how to deal with ties.

`knn` documentation on `use.all = TRUE`: "If true, all distances equal to the kth largest are included." Thus, fractions other than n/k will be present.

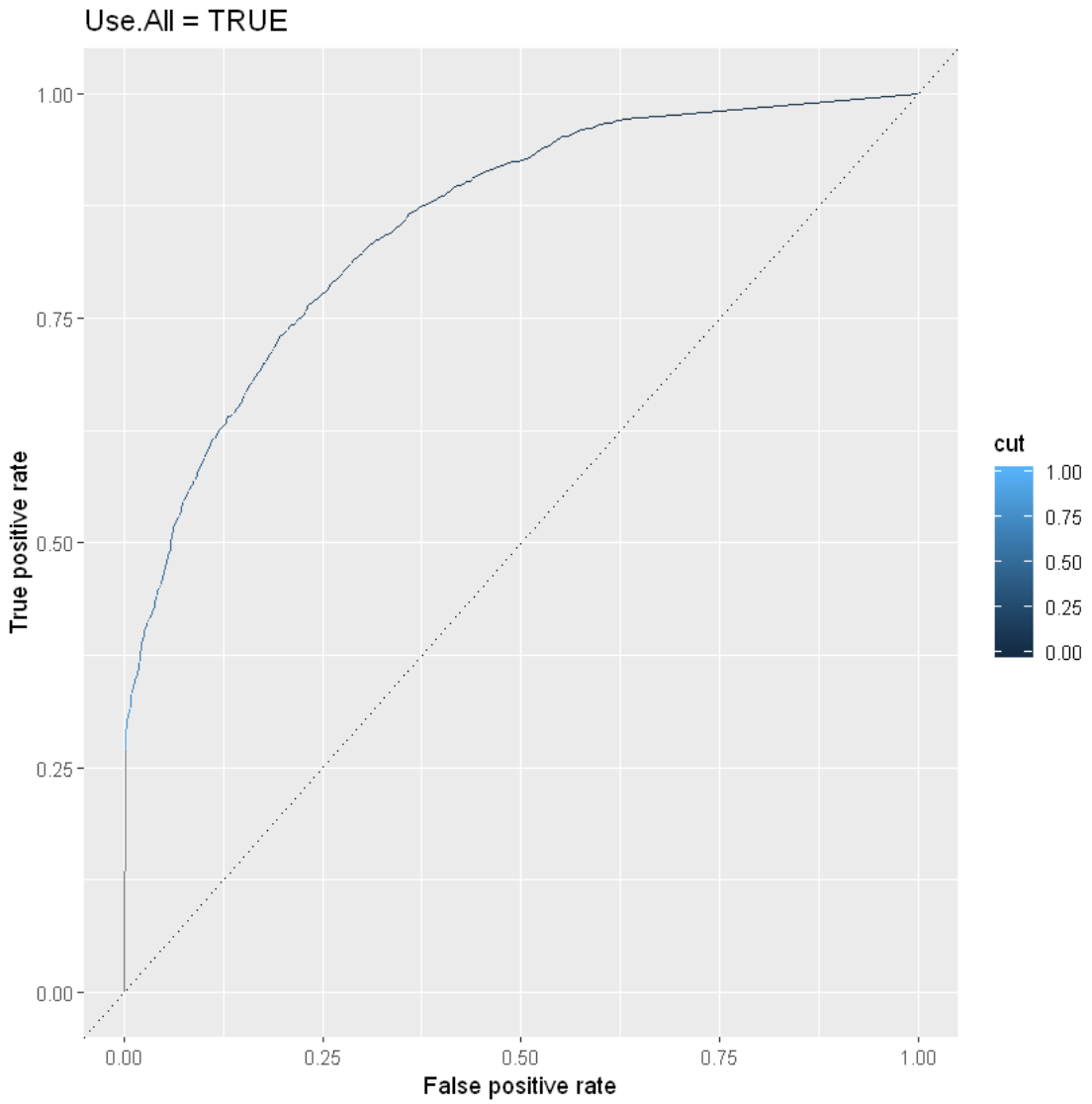
[5]:
```
income_true <- knn(train_q, test = test_q, cl = train$income, k = 10, prob =␣
↪TRUE, use.all = TRUE)

income_true_probs <- matrix(nrow = length(income_true), ncol = 1)
for(i in 1:length(income_true)) {
  p = attr(income_true, 'prob')[i]
  income_true_probs[i, 1] <- ifelse(income_true[i] == ' >50K', p, 1 - p)
}
income_true_probs <- income_true_probs %>% as.data.frame()
names(income_true_probs) <- c(' >50K')

income_true_probs %>% head(15)
```

>50K
0.00000000
0.30000000
0.00000000
0.00000000
0.00000000
0.00000000
0.05882353
0.30000000
0.18823529
0.05128205
0.27272727
0.28571429
0.27272727
0.00000000
1.00000000

[7]:
```
pred_true <- ROCR::prediction(income_true_probs$` >50K`, test$income)
perf_true <- ROCR::performance(pred_true, 'tpr', 'fpr')
perf_true_df <- data.frame(perf_true@x.values, perf_true@y.values,␣
↪perf_true@alpha.values)
names(perf_true_df) <- c("fpr", "tpr", "cut")
perf_true_df %>% head(15)
```

fpr	tpr	cut
0.000000000	0.0000000	Inf
0.001435309	0.2691131	1.0000000
0.001435309	0.2697248	0.9333333
0.001435309	0.2709480	0.9285714
0.001640353	0.2727829	0.9166667
0.002050441	0.2770642	0.9090909
0.002665573	0.2917431	0.9000000
0.002665573	0.2923547	0.8666667
0.003075661	0.2929664	0.8571429
0.003075661	0.2941896	0.8461538
0.003075661	0.2948012	0.8387097
0.003075661	0.2960245	0.8333333
0.003485749	0.2990826	0.8181818
0.003485749	0.2996942	0.8148148
0.003485749	0.3009174	0.8125000

[8]:
```
roc_true <- perf_true_df %>% ggplot(aes(x = fpr, y = tpr, color = cut)) +
  geom_line() + geom_abline(intercept = 0, slope = 1, lty = 3) +
  ylab(perf_true@y.name) + xlab(perf_true@x.name) + ggtitle("Use.All = TRUE")
roc_true
```



4 Use.All = FALSE

`knn` documentation on `use.all = FALSE`: "If false, a random selection of distances equal to the kth is chosen to use exactly k neighbours."

Thus, only fractions in the form n/k will be present.

[9]:
```
income_false <- knn(train_q, test = test_q, cl = train$income, k = 10, prob =␣
↪TRUE, use.all = FALSE)

income_false_probs <- matrix(nrow = length(income_false), ncol = 1)
for(i in 1:length(income_false)) {
  p = attr(income_false, 'prob')[i]
```

```
  income_false_probs[i, 1] <- ifelse(income_false[i] == ' >50K', p, 1 - p)
}
income_false_probs <- income_false_probs %>% as.data.frame()
names(income_false_probs) <- c(' >50K')

income_false_probs %>% head(15)
```

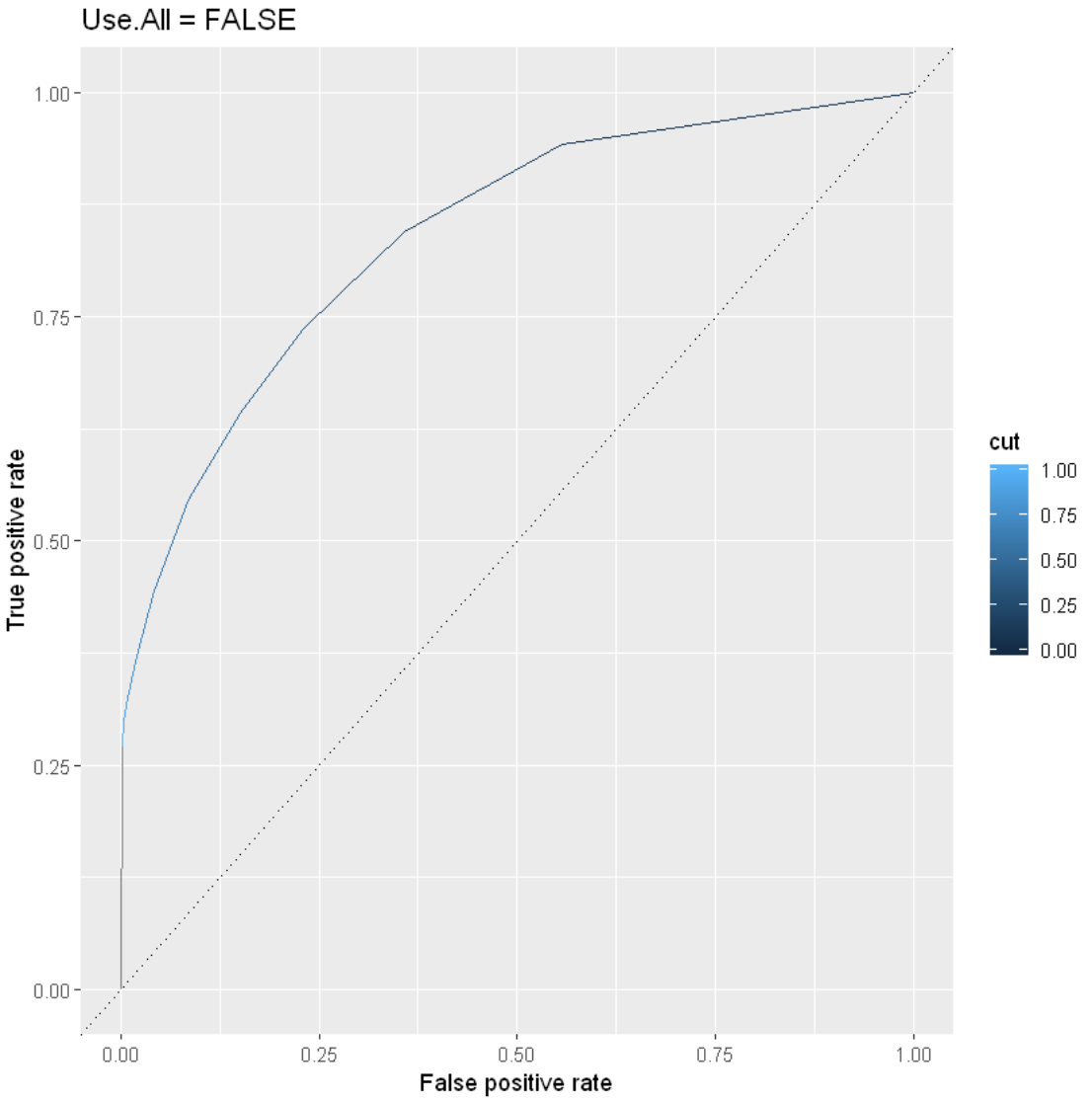
>50K
0.0
0.3
0.0
0.0
0.0
0.0
0.1
0.3
0.1
0.1
0.2
0.2
0.3
0.0
1.0

[10]:
```
pred_false <- ROCR::prediction(income_false_probs$` >50K`, test$income)
perf_false <- ROCR::performance(pred_false, 'tpr', 'fpr')
perf_false_df <- data.frame(perf_false@x.values, perf_false@y.values,␣
 ↪perf_false@alpha.values)
names(perf_false_df) <- c("fpr", "tpr", "cut")
perf_false_df %>% head(15)
```

fpr	tpr	cut
0.000000000	0.0000000	Inf
0.001640353	0.2697248	1.0
0.003485749	0.2990826	0.9
0.008816896	0.3253823	0.8
0.021324585	0.3718654	0.7
0.042239081	0.4440367	0.6
0.084888251	0.5467890	0.5
0.151322534	0.6428135	0.4
0.231084683	0.7370031	0.3
0.358417060	0.8446483	0.2
0.556899733	0.9418960	0.1
1.000000000	1.0000000	0.0

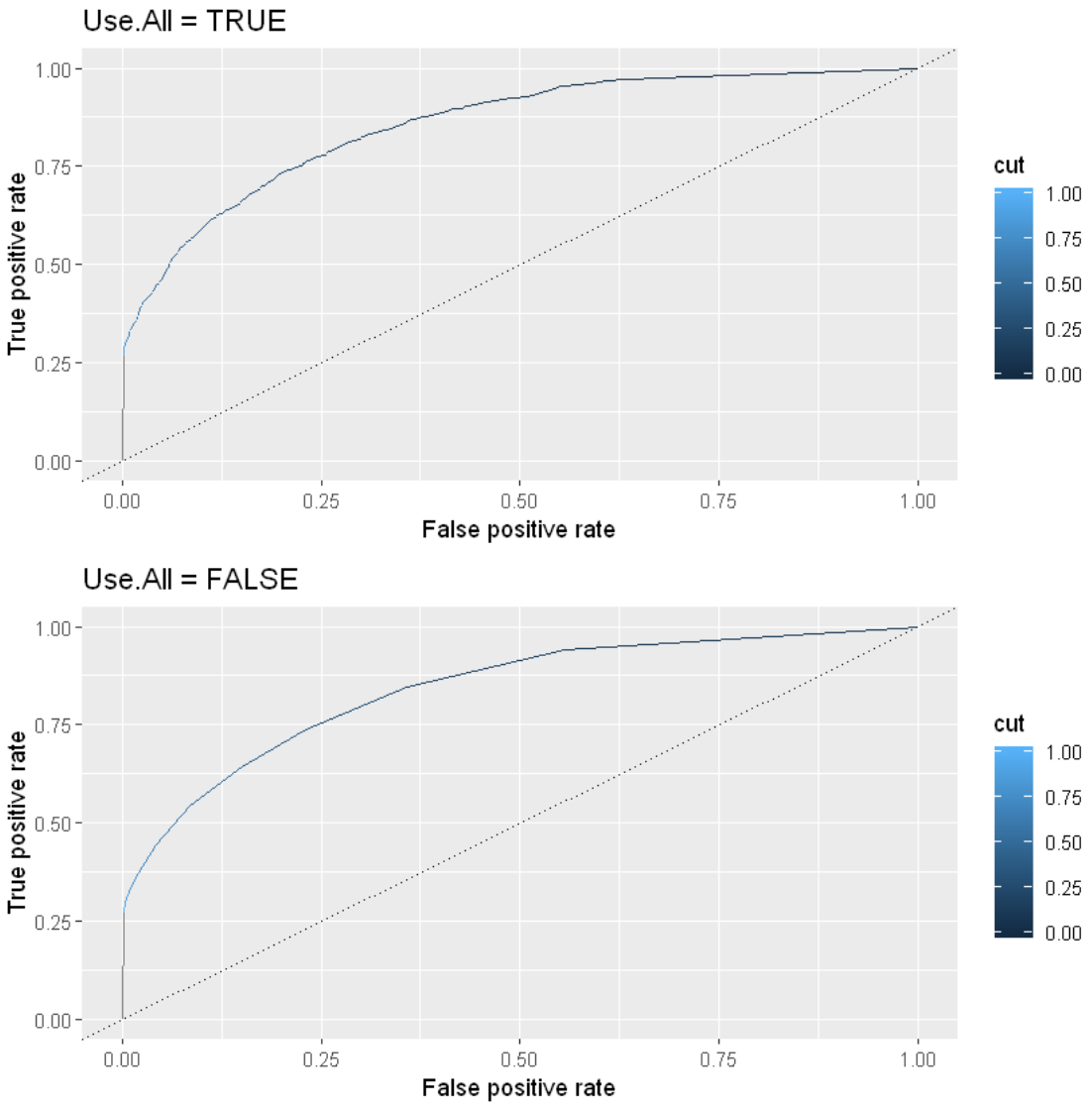
[11]:
```
roc_false <- perf_false_df %>% ggplot(aes(x = fpr, y = tpr, color = cut)) +
  geom_line() + geom_abline(intercept = 0, slope = 1, lty = 3) +
  ylab(perf_false@y.name) + xlab(perf_false@x.name) + ggtitle("Use.All = FALSE")
```

```
roc_false
```

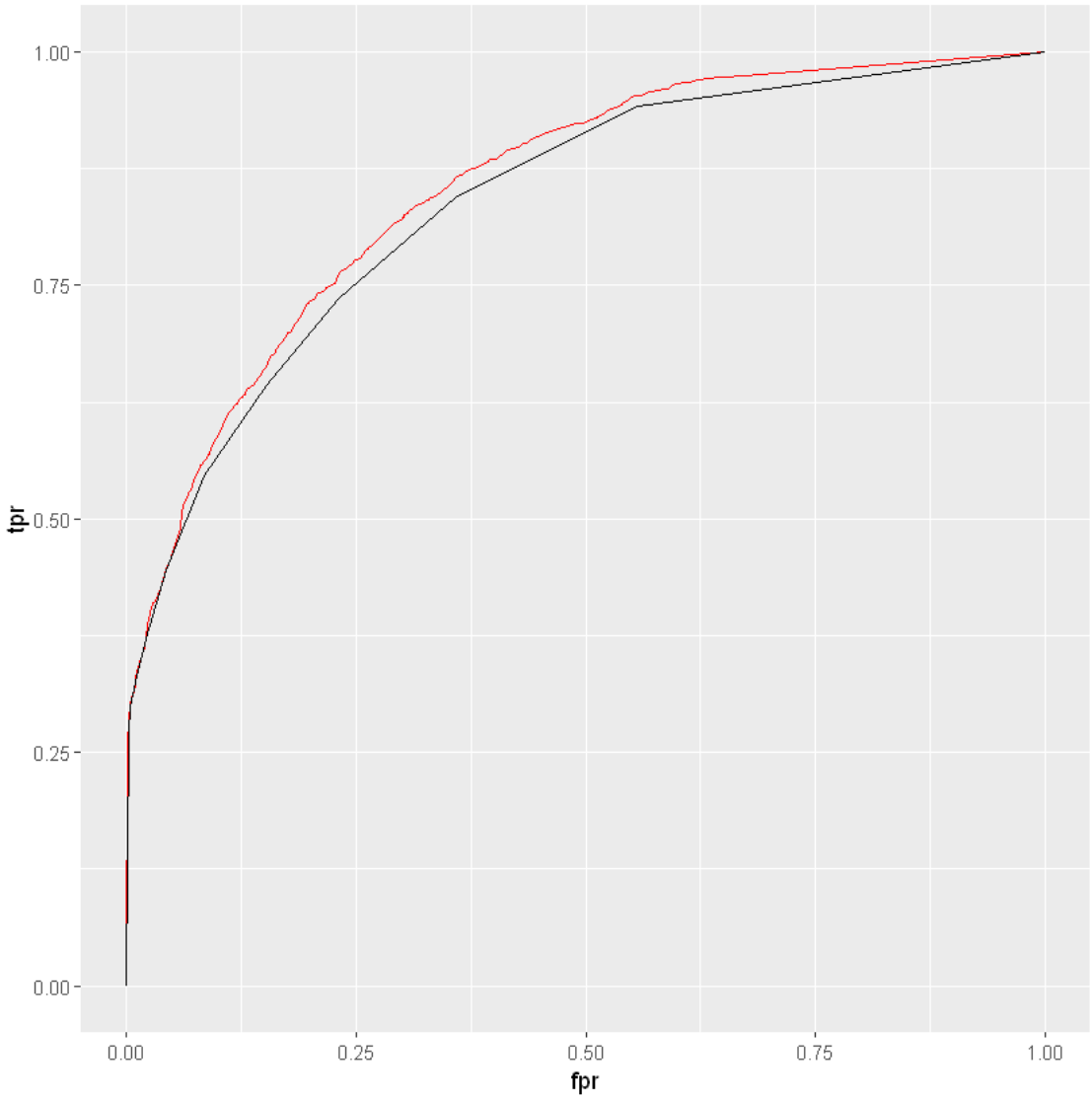


5 Graphing Both Models

[12]:
```
grid.arrange(roc_true, roc_false)
```



[13]:
```
ggplot() +
  geom_line(data = perf_true_df, aes(x = fpr, y = tpr), color = "red") +
  geom_line(data = perf_false_df, aes(x = fpr, y = tpr), color = "black")
```



Red = `use.all = TRUE`
Black = `use.all = FALSE`