logit_demo

dan le

1/4/2019

toy training data with numerical predictors

```
# requirements:
# library(dplyr)
# library(onehot)
# library(pscl)

# generate some training data
pred.train = c(0,1,1,0,0)
res.train = c('A','B','B','A','B')
df = data.frame(pred = pred.train, res = res.train)

# train model
mod = glm(res ~ pred, family=binomial(link='logit'),data=df)
# summary(mod) # uncomment to see model summary

# calculate McFadden's pseudo R2 (ref: https://www.r-bloggers.com/evaluating-logistic-regression-models/
## A measure of model accuracy very roughly analogous to variance explained (R2) in least squares regres
pR2(mod)[4]
```

```
##  McFadden
## 0.4325381
```

toy training data with categorical predictors

```
# convert numeric pred to categorical pred
pred.train.cat = sapply(pred.train, as.character)

# one-hot encode pred
encoder = onehot(data.frame(pred = pred.train.cat))
pred.train.cat.mat = predict(encoder, data.frame(pred = pred.train.cat))
pred.train.cat.df = data.frame(pred.train.cat.mat)

# append response
df.onehot = pred.train.cat.df %>% mutate(res=as.factor(res.train))

# train model
mod = glm(res ~ ., family=binomial(link='logit'),data=df.onehot)
# summary(mod) # uncomment to see model summary

# calculate McFadden's pseudo R2
pR2(mod)[4]
```

```
##  McFadden
## 0.4325381
```

As you can see, the one-hot encoded categorical training predictors returns the same pseudo R2 value as when the predictors are numerical.