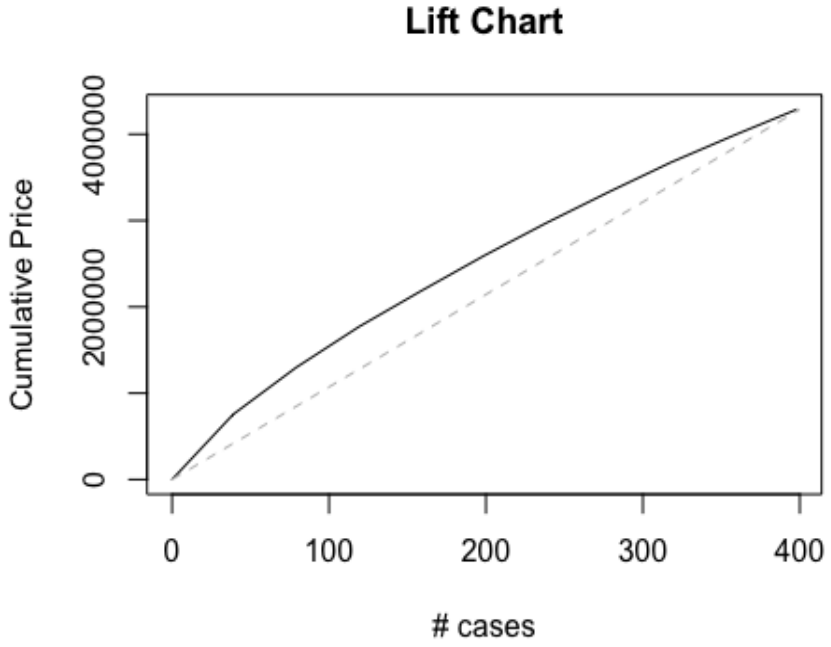
Predicting Prices of Used Cars Linear Regression

Lucky

```
### Predicting Prices of Used Cars Linear Regression

# Evaluate Performance before building the model
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

# load file
toyota.corolla.df <- read.csv("ToyotaCorolla.csv")

# randomly generate training and validation sets
training <- sample(toyota.corolla.df$Id, 600)
validation <- sample(setdiff(toyota.corolla.df$Id, training), 400)

# run linear regression model
reg <- lm(Price~., data=toyota.corolla.df[,-c(1,2,8,11)], subset=training,
          na.action=na.exclude)
pred_t <- predict(reg, na.action=na.pass)
pred_v <- predict(reg, newdata=toyota.corolla.df[validation,-c(1,2,8,11)],
                  na.action=na.pass)

## Warning in predict.lm(reg, newdata = toyota.corolla.df[validation, -c(1, :
## prediction from a rank-deficient fit may be misleading

## evaluate performance
# training
accuracy(pred_t, toyota.corolla.df[training,]$Price)

##                      ME     RMSE      MAE        MPE     MAPE
## Test set -6.38504e-11 1042.883 774.8431 -0.8174744 7.694502

# validation
accuracy(pred_v, toyota.corolla.df[validation,]$Price)

##                  ME     RMSE      MAE       MPE     MAPE
## Test set -129.9249 1209.017 885.0014 -2.586886 9.039274

### Build Model

# remove missing Price data
toyota.corolla.df <- toyota.corolla.df[!is.na(toyota.corolla.df[validation,]$
Price),]
```

```r
# regression model based on all numerical predictors
reg <- lm(Price~., data = toyota.corolla.df[,-c(1,2,8,11)], subset = training
)

# predictions
pred_v <- predict(reg, newdata = toyota.corolla.df[validation,-c(1,2,8,11)])

## Warning in predict.lm(reg, newdata = toyota.corolla.df[validation, -c(1, :
## prediction from a rank-deficient fit may be misleading

# load package gains, compute gains (we will use package caret for categorica
l y later)
library(gains)
gain <- gains(toyota.corolla.df[validation,]$Price[!is.na(pred_v)], pred_v[!i
s.na(pred_v)])

# cumulative lift chart
options(scipen=999) # avoid scientific notation
# we will compute the gain relative to price
price <- toyota.corolla.df[validation,]$Price[!is.na(toyota.corolla.df[valida
tion,]$Price)]
plot(c(0,gain$cume.pct.of.total*sum(price))~c(0,gain$cume.obs),
     xlab="# cases", ylab="Cumulative Price", main="Lift Chart", type="l")

# baseline
lines(c(0,sum(price))~c(0,dim(toyota.corolla.df[validation,])[1]), col="gray"
, lty=2)
```



```r
# Decile-wise lift chart
barplot(gain$mean.resp/mean(price), names.arg = gain$depth,
        xlab = "Percentile", ylab = "Mean Response", main = "Decile-wise lift
chart")
```