Memorization Method - part 1

SIT220009: Data Science

Presented by Hyebong Choi

Classification and Regression

- Classification is a task that predicts discrete event(class)
  - is a e-mail spam or not (binary)
  - does a patient have breast cancer or not (binary)
  - predict letter grade a student expected to get for this class (multi-class, A, B, C, D, F)
- Regression is a task that predicts continuous value(score)
  - expected housing price
  - expected GPA

Memorization Method

- The simplest methods that generate answers of
  - **a majority category** (in the case of classification)
  - **a average value** (in the case of scoring)
- single variable models that use one variable to make answer
- multi-variable models that use more than one variables
  - includes **decision trees**, **k nearest neighbor** and **Naive Bayes methods**.
- intuitive and straightforward

Sample Dataset

Data originally extracted from 1994 Census database. Prediction task is to determine whether a person makes over 50K a year.

Variables:

age: continuous.

workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

education-num: the number of year each person get educated

marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

sex: Female, Male.

capital-gain: continuous.

capital-loss: continuous.

hours-per-week: continuous. working hours per week.

native-country: United-States, Cambodia, England, Puerto-Rico, …

income_mt_50k: Indicating if the person’s yearly income is more than 50,000 USD. Target Variable

Data Exploration

```
str(adult)

## 'data.frame':    32561 obs. of  14 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass     : Factor w/ 9 levels " ?"," Federal-gov",..: 8 7 5 5 5 5 5 7 5 5 ...
##  $ education     : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13
10 ...
##  $ education_num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital-status: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3
3 4 3 5 3 ...
##  $ occupation    : Factor w/ 15 levels " ?"," Adm-clerical",..: 2 5 7 7 11 5 9 5 11
5 ...
##  $ relationship  : Factor w/ 6 levels " Husband"," Not-in-family",..: 2 1 2 1 6 6 2 1
2 1 ...
##  $ race          : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5
5 ...
##  $ sex           : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
##  $ capital-gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital-loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours-per-week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ native-country: Factor w/ 42 levels " ?"," Cambodia",..: 40 40 40 40 6 40 24 40 40
40 ...
##  $ income_mt_50k : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

Classification with Single Variable Model

- Given a **single input variable**, we predict if **person's yearly income** is more than **50k USD.**
- We can choose predictor (input variable) from age, education, workclass, …

Data Preparation

```r
load(url('https://github.com/hbchoi/SampleData/raw/master/adult.RData'))


set.seed(2020)
n_sample <- nrow(adult)
rgroup <- runif(n_sample)

adult.train <- subset(adult, rgroup <= 0.8)
adult.test <- subset(adult, rgroup > 0.8)


dim(adult.train)

## [1] 26040    14

dim(adult.test)

## [1] 6521   14
```

- We partition the dataset into two groups with ratio of 8:2
  - train.df for building prediction model
  - test.df is to evaluate our model

Data Preparation

```
table(adult.train$income_mt_50k)

##
## FALSE  TRUE
## 19740  6300

prop.table(table(adult.train$income_mt_50k))

##
##     FALSE      TRUE
## 0.7580645 0.2419355

prop.table(table(adult.test$income_mt_50k))

##
##     FALSE      TRUE
## 0.7636866 0.2363134
```

Building a Single Variable Model

we first choose "**occupation**" variable as predictor

```
tble <- table(adult.train$occupation,
adult.train$income_mt_50k)
tble
```

```
##
##                       FALSE TRUE
##     ?                  1325  148
##     Adm-clerical       2560  415
##     Armed-Forces          8    1
##     Craft-repair       2539  748
##     Exec-managerial    1708 1578
##     Farming-fishing     721   94
##     Handlers-cleaners  1010   72
##     Machine-op-inspct  1401  193
##     Other-service      2530  107
##     Priv-house-serv     119    0
##     Prof-specialty     1830 1490
##     Protective-serv     341  169
##     Sales              2127  790
##     Tech-support        504  228
##     Transport-moving   1017  267
```

```
prop.table(tble, margin = 1)
```

```
##
##                            FALSE       TRUE
##     ?                 0.89952478 0.10047522
##     Adm-clerical      0.86050420 0.13949580
##     Armed-Forces      0.88888889 0.11111111
##     Craft-repair      0.77243687 0.22756313
##     Exec-managerial   0.51978089 0.48021911
##     Farming-fishing   0.88466258 0.11533742
##     Handlers-cleaners 0.93345656 0.06654344
##     Machine-op-inspct 0.87892095 0.12107905
##     Other-service     0.95942359 0.04057641
##     Priv-house-serv   1.00000000 0.00000000
##     Prof-specialty    0.55120482 0.44879518
##     Protective-serv   0.66862745 0.33137255
##     Sales             0.72917381 0.27082619
##     Tech-support      0.68852459 0.31147541
##     Transport-moving  0.79205607 0.20794393
```

Building a Single Variable Model

```
sv_model_job <- prop.table(tble, margin = 1)[,2]
sort(sv_model_job, decreasing = T)
```

```
##     Exec-managerial      Prof-specialty     Protective-serv
##          0.48021911          0.44879518          0.33137255
##        Tech-support               Sales        Craft-repair
##          0.31147541          0.27082619          0.22756313
##    Transport-moving        Adm-clerical   Machine-op-inspct
##          0.20794393          0.13949580          0.12107905
##     Farming-fishing        Armed-Forces                   ?
##          0.11533742          0.11111111          0.10047522
##   Handlers-cleaners       Other-service     Priv-house-serv
##          0.06654344          0.04057641          0.00000000
```

48% of executive-managers earn more than 50k yearly

none of private house servant earn more than 50k yearly

Prediction on Training Dataset

```
adult.train$est_prob <- sv_model_job[adult.train$occupation]

head(adult.train[, c('occupation','est_prob','income_mt_50k')], 10)
```

```
##          occupation   est_prob income_mt_50k
## 1      Adm-clerical 0.13949580         FALSE
## 2   Exec-managerial 0.48021911         FALSE
## 3 Handlers-cleaners 0.06654344         FALSE
## 4 Handlers-cleaners 0.06654344         FALSE
## 5    Prof-specialty 0.44879518         FALSE
## 6   Exec-managerial 0.48021911         FALSE
## 7     Other-service 0.04057641         FALSE
## 8   Exec-managerial 0.48021911          TRUE
## 9    Prof-specialty 0.44879518          TRUE
## 10  Exec-managerial 0.48021911          TRUE
```

Making a Decision based on Prob.

```
# threshold setting
threshold <- 0.4
adult.train$prediction <- adult.train$est_prob > threshold
head(adult.train[, c('occupation','est_prob','prediction', 'income_mt_50k')], 10)
```

```
##          occupation   est_prob prediction income_mt_50k
## 1      Adm-clerical 0.13949580      FALSE         FALSE
## 2   Exec-managerial 0.48021911       TRUE         FALSE
## 3 Handlers-cleaners 0.06654344      FALSE         FALSE
## 4 Handlers-cleaners 0.06654344      FALSE         FALSE
## 5    Prof-specialty 0.44879518       TRUE         FALSE
## 6   Exec-managerial 0.48021911       TRUE         FALSE
## 7     Other-service 0.04057641      FALSE         FALSE
## 8   Exec-managerial 0.48021911       TRUE          TRUE
## 9    Prof-specialty 0.44879518       TRUE          TRUE
## 10  Exec-managerial 0.48021911       TRUE          TRUE
```

- We classify the group of people earning more than 50k, if their estimated probability is greater than threshold (0.4 here)

Accuracy

- Now we have predicted answers for training set
- Let's see how accurate it is
- accuracy = # of correct predictions / # of all examples

```
conf.table <- table(pred = adult.train$prediction,
actual = adult.train$income_mt_50k)
conf.table
```

```
##          actual
## pred     FALSE  TRUE
##    FALSE 16202  3232
##    TRUE   3538  3068
```

```
accuracy <- sum(diag(conf.table)) / sum(conf.table)
accuracy
```

```
## [1] 0.7400154
```

Prediction on Test Data

Working well in the training dataset not necessarily guarantees it works well in real world

Since it can memorize training examples to make accurate prediction - **Overfitting**

We need a prediction model that can be generalized

To see the generalized performance, we use test set which is unseen during the model training

We simulate the future data with the test data

Prediction on Test Data

```
adult.test$est_prob <- sv_model_job[adult.test$occupation]
adult.test$prediction <- adult.test$est_prob > threshold

head(adult.test[, c('occupation','est_prob','prediction',
'income_mt_50k')], 10)
```

```
##            occupation  est_prob prediction income_mt_50k
## 13       Adm-clerical 0.1394958      FALSE         FALSE
## 17    Farming-fishing 0.1153374      FALSE         FALSE
## 23    Farming-fishing 0.1153374      FALSE         FALSE
## 24   Transport-moving 0.2079439      FALSE         FALSE
## 25       Tech-support 0.3114754      FALSE         FALSE
## 31    Protective-serv 0.3313725      FALSE         FALSE
## 33    Exec-managerial 0.4802191       TRUE         FALSE
## 34       Adm-clerical 0.1394958      FALSE         FALSE
## 38       Adm-clerical 0.1394958      FALSE         FALSE
## 41  Machine-op-inspct 0.1210790      FALSE         FALSE
```

Prediction on Test Data

```
conf.table <- table(pred = adult.test$prediction, actual =
adult.test$income_mt_50k)
conf.table
```

```
##          actual
## pred     FALSE TRUE
##   FALSE  4139  782
##   TRUE    841  759
```

```
accuracy <- sum(diag(conf.table)) / sum(conf.table)
accuracy
```

```
## [1] 0.7511118
```

Two Questions

Acc. of 0.740 on adult.train is quite similar 0.751 on adult.test

So our model does not over-fit the problem

- Is “Accuracy” good enough to measure our prediction model?
- Is 0.751 good enough? Can we do it better?
  - Try different threshold or predictor to build a prediction model

Changing Threshold

prediction with threshold 0.3

```
get_accuracy <- function(pred, actual){
  tble <- table(pred , actual)
  return( round(sum(diag(tble))  / sum(tble), 3) )
  }
```

```
threshold <- 0.3
adult.train$prediction <- adult.train$est_prob > threshold

print(paste("accuracy on training set",
            get_accuracy(adult.train$prediction, adult.train$income_mt_50k)))

## [1] "accuracy on training set 0.723"


adult.test$prediction <- adult.test$est_prob > threshold

print(paste("accuracy on test set",
            get_accuracy(adult.test$prediction, adult.test$income_mt_50k)))

## [1] "accuracy on test set 0.729"
```

Exercise

- Try a different input variable “education” to build a single variable prediction model
- Set the threshold 0.5 and Find out the accuracy on adult.train and adult.test
- Change the threshold to 0.6 and 0.4, is Accuracy different?
- Is “**education**” variable more predictive than “**occupation**” variable?

Confusion Matrix

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

Precision and Recall

```
conf.table

##          actual
## pred     FALSE  TRUE
##    FALSE  4139   782
##    TRUE    841   759

precision <- conf.table[2,2] / sum(conf.table[2,])
recall <- conf.table[2,2] / sum(conf.table[,2])


precision

## [1] 0.474375

recall

## [1] 0.4925373
```

- precision:
  - true positive / (true positive + false positive)
- Recall
  - true positive / (true positive + false negative)

Questions

- In which cases, precision is more important that recall?
- When recall is more important then?

Question

- Change threshold of occupation model into 0.25 and 0.45
- How does the **accuracy** change?
- How do **precision** and **recall** change?
- Having higher threshold value, does it increase or decrease **precision**?
- What about **recall**?
- Explain why they are so.

ROC curve

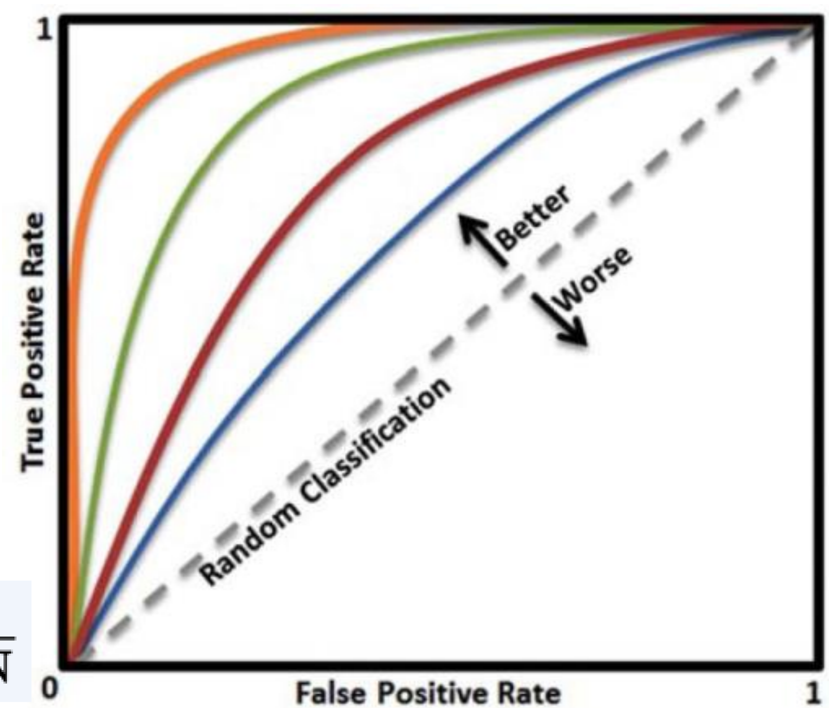
threshold를 높이면 precision이 올라가고
threshold를 낮추면 recall이 올라간다.

- We can change our stance from conservative to optimistic by changing threshold
- Accordingly accuracy, precision, and recall changes as well
- Then how can we compare performance of two different models
- We use Receiver Operating Characteristic (ROC) curve and area under the curve (AUC)

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$$



ROC curve

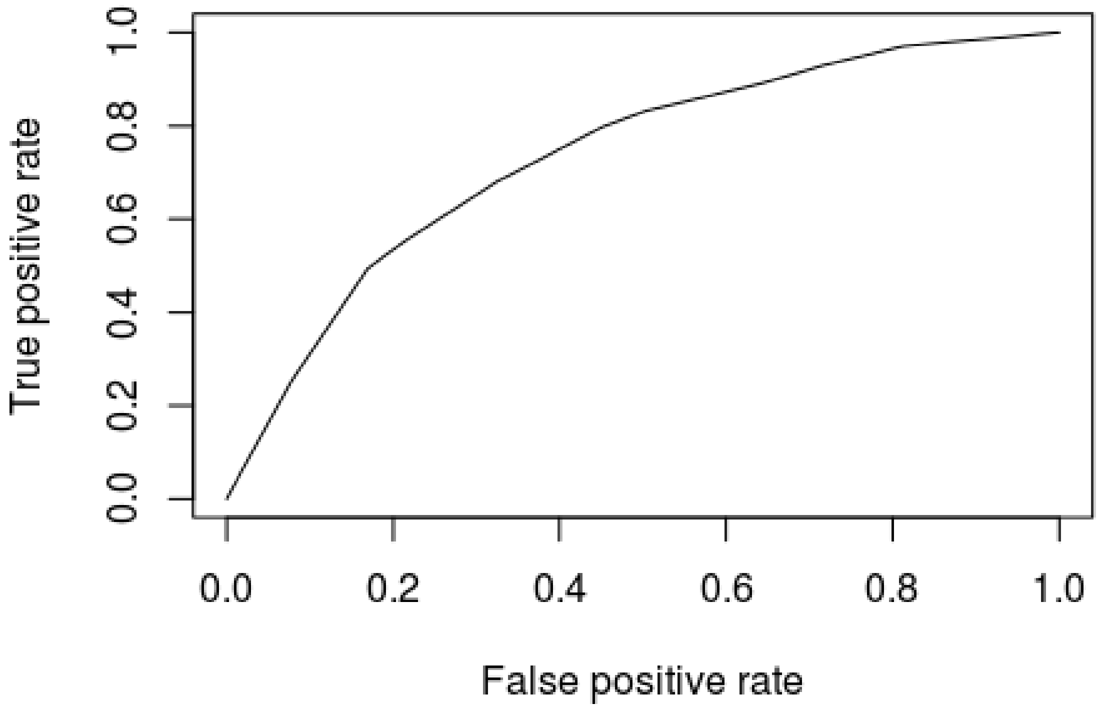
$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

ROC curve for occupation model

```
library(ROCR)

plot(performance(prediction(adult.test$est_prob, adult.test$income_mt_50k),
'tpr', 'fpr'))
```



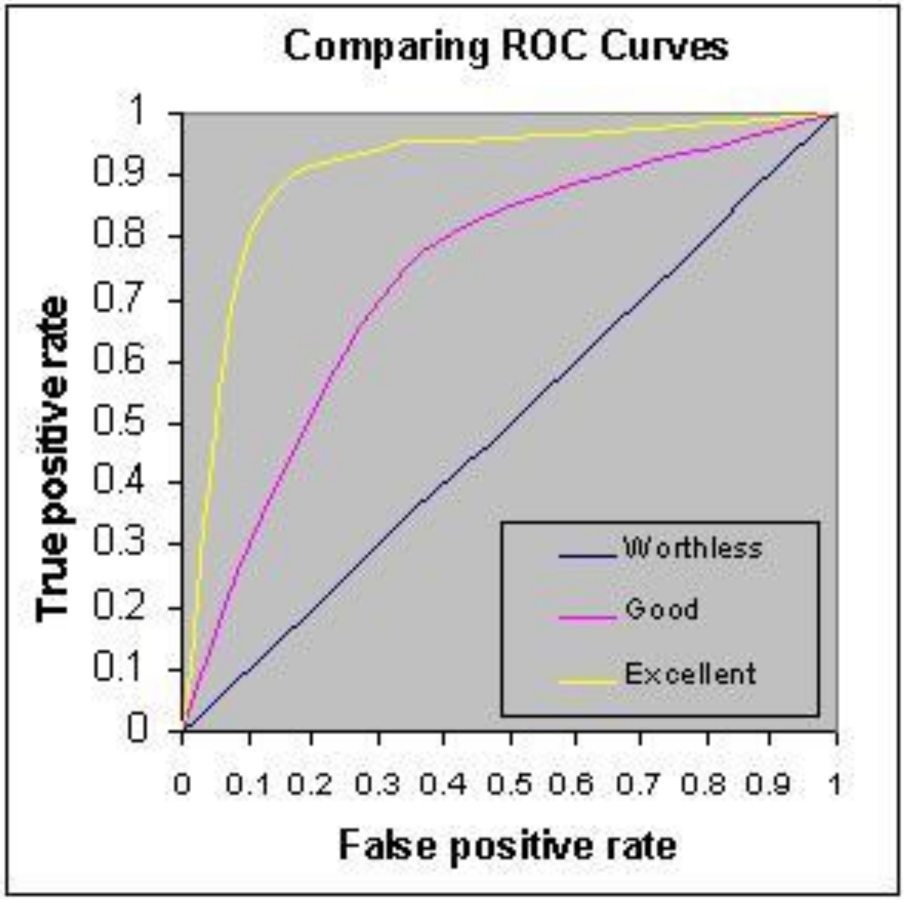
AUC for occupation model

```
calAUC <- function(predCol, targetCol){
  perf <- performance(prediction(predCol, targetCol), 'auc')
  as.numeric(perf@y.values)
}

calAUC(adult.train$est_prob, adult.train$income_mt_50k)

## [1] 0.7299324

calAUC(adult.test$est_prob, adult.test$income_mt_50k)

## [1] 0.7347861
```

Finding best model

- we use area under curve (AUC) to find the best model

- .90-1 = excellent (A)
- .80-.90 = good (B)
- .70-.80 = fair (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)



Question

- What is AUC for education model?
- Does education model outperform the occupation model?
- Why do you think so?

Using continuous variable as input variable

Now we take a continuous variable "age" as predictor(input variable) to make prediction

To use age variable for prediction, we convert it into range variable age_group, which contains under20, 20s, 30s, 40s, 50s, over60

```
summary(adult$age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   28.00   37.00   38.58   48.00   90.00

adult.train$age_group <- cut(adult.train$age, breaks = c(0,20,30,40,50,60, Inf),
                             labels = c('under20', '20s', '30s', '40s', '50s', 'over60'),
                             right = F)
table(adult.train$age_group)

##
## under20     20s     30s     40s     50s  over60
##    1332    6401    6920    5732    3571    2084
```

Using contiguous variable as input variable

```
tble <- table(adult.train$age_group, adult.train$income_mt_50k)
tble
```

```
##
##            FALSE TRUE
##   under20  1330     2
##   20s      5986   415
##   30s      5064  1856
##   40s      3592  2140
##   50s      2186  1385
##   over60   1582   502
```

```
sv_model_age <- prop.table(tble, margin = 1)[,2]
sort(sv_model_age, decreasing = T)
```

```
##         50s         40s         30s      over60         20s     under20
## 0.387846542 0.373342638 0.268208092 0.240882917 0.064833620 0.001501502
```

- We find that older people are more likely to make more money than younger people and the retired

Accuracy with threshold (0.3)

```r
get_accuracy <- function(pred, actual){
  tble <- table(pred , actual)
  return( round(sum(diag(tble))  / sum(tble), 3) )
  }

threshold <- 0.3

adult.train$est_prob <- sv_model_age[adult.train$age_group]
adult.train$prediction <- adult.train$est_prob > threshold

print(paste("accuracy on training set",
            get_accuracy(adult.train$prediction, adult.train$income_mt_50k)))

## [1] "accuracy on training set 0.672"

adult.test$age_group <- cut(adult.test$age, breaks = c(0,20,30,40,50,60, Inf),
                             labels = c('under20', '20s', '30s', '40s', '50s', 'over60'),
                             right = F)

adult.test$est_prob <- sv_model_age[adult.test$age_group]
adult.test$prediction <- adult.test$est_prob > threshold

print(paste("accuracy on test set",
            get_accuracy(adult.test$prediction, adult.test$income_mt_50k)))

## [1] "accuracy on test set 0.671"
```

Regression - Sample Dataset

```
load(url('https://github.com/hbchoi/SampleData/raw/master/insurance.RData'))
```

- `age`: This is an integer indicating the age of the primary beneficiary (excluding those above 64 years, since they are generally covered by the government).
- `sex`: This is the policy holder's gender, either `male` or `female`.
- `bmi`: This is the **body mass index** (**BMI**), which provides a sense of how over or under-weight a person is relative to their height. BMI is equal to weight (in kilograms) divided by height (in meters) squared. An ideal BMI is within the range of 18.5 to 24.9.
- `children`: This is an integer indicating the number of children / dependents covered by the insurance plan.
- `smoker`: This is `yes` or `no` depending on whether the insured regularly smokes tobacco.
- `region`: This is the beneficiary's place of residence in the U.S., divided into four geographic regions: `northeast`, `southeast`, `southwest`, or `northwest`.

Data Exploration

```
str(insurance)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
```

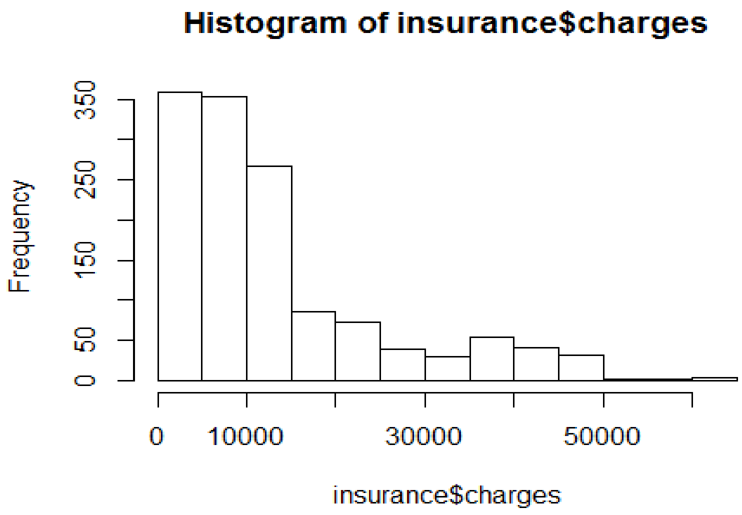
```
summary(insurance)
```

```
##       age            sex            bmi           children       smoker
##  Min.   :18.00   female:662   Min.   :15.96   Min.   :0.000   no :1064
##  1st Qu.:27.00   male  :676   1st Qu.:26.30   1st Qu.:0.000   yes: 274
##  Median :39.00                Median :30.40   Median :1.000
##  Mean   :39.21                Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                Max.   :53.13   Max.   :5.000
##        region        charges
##  northeast:324   Min.   : 1122
##  northwest:325   1st Qu.: 4740
##  southeast:364   Median : 9382
##  southwest:325   Mean   :13270
##                  3rd Qu.:16640
##                  Max.   :63770
```

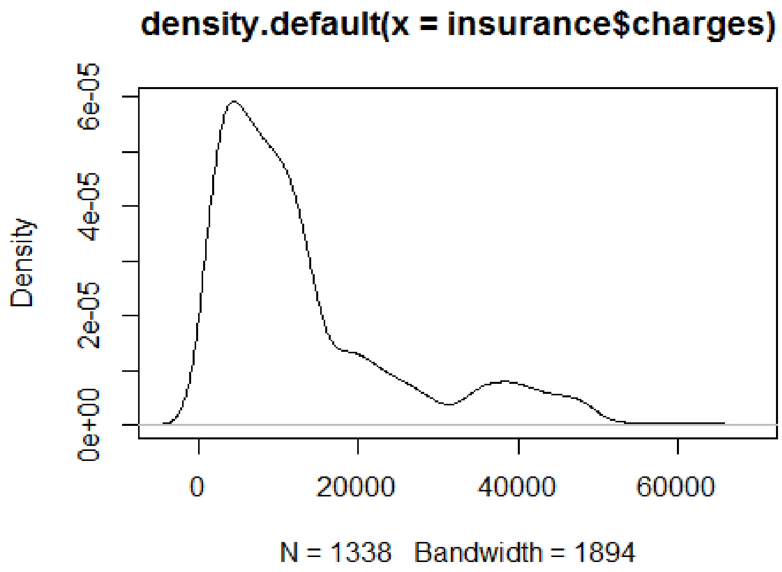
Data Exploration

- charges
  - amount of medical expenses charged by the customer – continuous value
  - regression

```
hist(insurance$charges)
```




```
plot(density(insurance$charges))
```

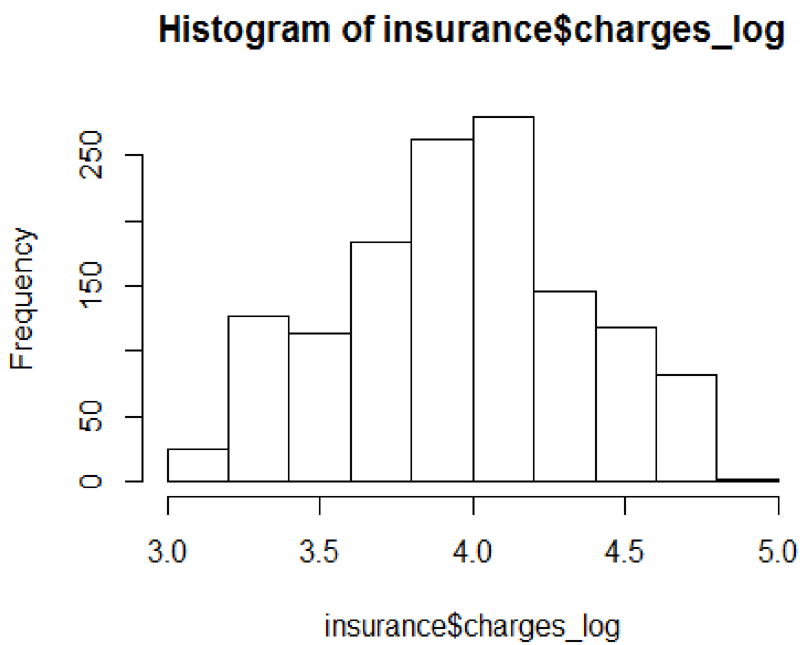


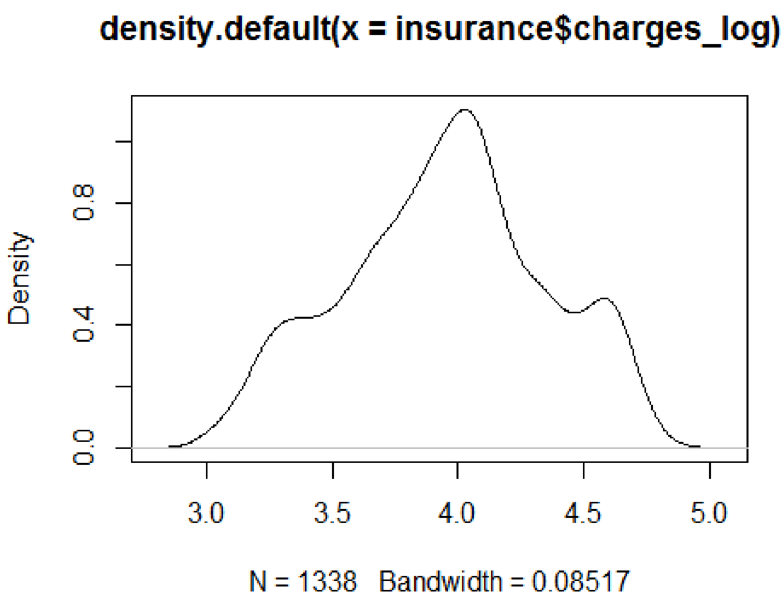
Transforming Response

```
insurance$charges_log <- log10(insurance$charges)
```

```
hist(insurance$charges_log)
```

```
plot(density(insurance$charges_log))
```





Data Preparation

```
set.seed(2018)
ncustomer <- nrow(insurance)
rgroup <- runif(ncustomer)

# data partition to learn a prediction model
train.df <- subset(insurance, rgroup <= 0.8)

# hold-out data for testing
test.df <- subset(insurance, rgroup > 0.8)

dim(train.df)

## [1] 1088    9

dim(test.df)

## [1] 250   9
```

- We partition the dataset into two groups with ratio of 8:2
  - train.df for building prediction model
  - test.df is to evaluate our model

Single Variable Regression Model

- We model to predict **charge_log** with a single input variable
- Let us choose **smoker** variable for the first time
- We take average value for given value of **smoker**

```
sv_reg_smoker <- tapply(train.df$charges_log, train.df$smoker, mean)
sv_reg_smoker
```

```
##       no      yes
## 3.815283 4.473136
```

```
# make a prediction on train dataset
train.df$pred_charges_log <- sv_reg_smoker[train.df$smoker]


head(train.df[, c('smoker','pred_charges_log',  'charges_log', 'charges')])
```

```
##   smoker pred_charges_log charges_log   charges
## 1    yes         4.473136    4.227499 16884.924
## 2     no         3.815283    3.236928  1725.552
## 3     no         3.815283    3.648308  4449.462
## 4     no         3.815283    4.342116 21984.471
## 5     no         3.815283    3.587358  3866.855
## 6     no         3.815283    3.574797  3756.622
```

Errors

To know how closely our model can predict, take a look at errors

```
train.df$error <- train.df$charges_log - train.df$pred_charges_log

head(train.df[, c('smoker','pred_charges_log',  'charges_log', 'error')])

##   smoker pred_charges_log charges_log      error
## 1    yes         4.473136    4.227499 -0.2456373
## 2     no         3.815283    3.236928 -0.5783553
## 3     no         3.815283    3.648308 -0.1669760
## 4     no         3.815283    4.342116  0.5268326
## 5     no         3.815283    3.587358 -0.2279255
## 6     no         3.815283    3.574797 -0.2404860

summary(train.df$error)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.76530 -0.19350  0.05517  0.00000  0.20930  0.74800

hist(train.df$error)
```

We measure the **closeness** form predicted value to actual label

To avoid positive and negative errors canceling out each other, we take squared error instead.

MSE is average squared error

RMSE is square rooted **MSE**. (to have same unit measurement)

```
MSE_train <- mean(train.df$error ** 2)
MSE_train
```

```
## [1] 0.08899245
```

```
RMSE_train <- sqrt(MSE_train)
RMSE_train
```

```
## [1] 0.298316
```

Our predicted values are typically 0.29 away from actual **charges_log**

i.e. $10^{0.29}$ = 1.95 times bigger or lower

RMSE on Test data

```
test.df$pred_charges_log <- sv_reg_smoker[test.df$smoker]

RMSE_test <- sqrt(mean((test.df$charges_log - test.df$pred_charges_log) ** 2))
RMSE_test

## [1] 0.2964113
```

RMSE on test data is 0.296 which is quite similar to train data

Comparing with Standard Deviation

```
RMSE_train
## [1] 0.298316
sd(train.df$charges_log)
## [1] 0.3998689
RMSE_test
## [1] 0.2964113
sd(test.df$charges_log)
## [1] 0.3978409
```

R^2

A measure of how well the model fits or explains the data

A value between 0-1

- ✓ near 1: model fits well
- ✓ near 0: no better than guessing the average value

Calculating R^2

R^2 is the variance explained by the model.

$$R^2 = 1 - \frac{RSS}{SS_{Tot}}$$

Where

$$RSS = \sum (y - prediction)^2$$

Residual sum of squares (variance from model)

$$SS_{Tot} = \sum (y - \bar{y})^2$$

Total sum of squares (variance of data)

R^2 for our S.V. regression model

```
RSS = sum(train.df$error ** 2)
RSS
```

```
## [1] 96.82378
```

```
SStot = sum((train.df$charges_log - mean(train.df$charges_log)) ** 2)
SStot
```

```
## [1] 173.806
```

```
Rsq = 1- RSS/SStot
Rsq
```

```
## [1] 0.4429204
```

R^2 for our S.V. regression model

```
# Rsq on Test set
test.df$error <- test.df$charges_log -
test.df$pred_charges_log
RSS = sum(test.df$error ** 2)
RSS

## [1] 21.96491

SStot = sum((test.df$charges_log -  mean(test.df$charges_log))
** 2)
SStot

## [1] 39.41108

Rsq = 1- RSS/SStot
Rsq

## [1] 0.4426717
```

Exercise

- Try the variable **region** as a input variable for regression to predict **charges_log**
  - What is RMSE and R^2 for the model

- Try the variable **age** as a input variable for regression to predict **charges_log**
  - What is RMSE and R^2 for the model

References

- Practical Data Science with R, by Nina Zumel and John Mount
- R을 이용한 데이터 분석 실무, 서민구, 길벗
- [DBGUIDE 연재] ggplot2를 이용한 R 시각화
  - http://freesearch.pe.kr/archives/3134