Machine Learning Workshop in R

Azka Javaid, Caleb Ki & Muling Si

March 20, 2017

Machine Learning Overview

- Ability to process raw data like speech, text and images
- Develop pattern-recognition, image identification and reinforcement learning models
- Facilitate speech transcription through natural language processing (NLP)
- Identify data anomalies and create user-targetted recommendation systems

Supervised Learning

- Modeling a response variable as a function of explanatory variables
- Data contains measurements of outcome variables (whether or not someone has diabetes)
    - Regression
    - Decision Trees
    - Random Forests
    - Nearest Neighbor
    - Naive Bayes
    - Artificial Neural Networks
    - Ensemble Learning Models

Unsupervised Learning

- Finding patterns or groupings with the absence of a clear response variable
- Unmeasured outcome (assembling DNA data into an evolutionary tree)
- Clustering (k-means, hierarchical clustering)
- Anomaly Detection
- Hidden Markov Models

Supervised Learning Workflow

- Partition data in training and test Sets
- Fit model (regression, decision trees, ensemble models)
- Assess model predictions through accuracy, ROC curves and k-fold cross-validation

Predict High Earners (>$50,000)

```
census <- read.csv(
  "http://archive.ics.uci.edu/ml/machine-learning-databases/ac
)
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")
```

Data Glimpse

```r
glimpse(census)
```

```
Observations: 32,561
Variables: 15
$ age            <int> 39, 50, 38, 53, 28, 37, 49, 52, 31, 42, 37, 30, 23, 32, 40, 34, 25, 32,...
$ workclass      <fctr> State-gov, Self-emp-not-inc, Private, Private, Private, Private,...
$ fnlwgt         <int> 77516, 83311, 215646, 234721, 338409, 284582, 160187, 209642, 45781, 15...
$ education      <fctr> Bachelors, Bachelors, HS-grad, 11th, Bachelors, Masters, 9th, ...
$ education.num  <int> 13, 13, 9, 7, 13, 14, 5, 9, 14, 13, 10, 13, 13, 12, 11, 4, 9, 9, 7, 14,...
$ marital.status <fctr> Never-married, Married-civ-spouse, Divorced, Married-civ-spouse, ...
$ occupation     <fctr> Adm-clerical, Exec-managerial, Handlers-cleaners, Handlers-cleaner...
$ relationship   <fctr> Not-in-family, Husband, Not-in-family, Husband, Wife, Wife, Not...
$ race           <fctr> White, White, White, Black, Black, White, Black, White, White...
$ sex            <fctr> Male, Male, Male, Male, Female, Female, Female, Male, Female,...
$ capital.gain   <int> 2174, 0, 0, 0, 0, 0, 0, 0, 14084, 5178, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
$ capital.loss   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20...
$ hours.per.week <int> 40, 13, 40, 40, 40, 40, 16, 45, 50, 40, 80, 40, 30, 50, 40, 45, 35, 40,...
$ native.country <fctr> United-States, United-States, United-States, United-States, Cuba,...
$ income         <fctr> <=50K, <=50K, <=50K, <=50K, <=50K, <=50K, <=50K, >50K, >50K, ...
```

Figure 1:

Partitioning Training and Test Sets

```r
set.seed(164)
n <- nrow(census)
test <- sample.int(n, size = round(0.2 * n))
train <- census[-test, ]
test <- census[test, ]
tally(~income, data = train, format = "percent")
```

```
## income
##  <=50K   >50K
##   75.8   24.2
```

Logistic Regression

```
logmod <- glm(income ~  age + workclass + education +
                marital.status + occupation +
                relationship + race + sex +
                capital.loss + hours.per.week,
                data = train, family=binomial(link='logit'))
```

Variable Importance

```
varImp(logmod)
```

	Overall
	<dbl>
age	16.40375910
workclass Federal-gov	5.50505253
workclass Local-gov	2.25059283
workclass Never-worked	0.03781848
workclass Private	3.88254277
workclass Self-emp-inc	4.83279840
workclass Self-emp-not-inc	0.36856340
workclass State-gov	0.56331601
workclass Without-pay	0.04394140
education 11th	0.47913244

Confusion Matrix

```
pred = predict(logmod, newdata=test)
accuracy <- table(pred, test[,"income"])
sum(diag(accuracy))/sum(accuracy)
```

```
## [1] 0.000154
```

K-Nearest Neighbors (KNN)

- "Lazy Learners"
- Predict outcomes without constructing a model

The idea

- A dataset with p attributes (explanatory variables)
- Use Euclidean distance as the metric
- Observations that are *close* to each other probably have similar outcomes

Steps

- Find the k observations in the training set closest to x^*
- Aggregate function f, calculate $y^* = f(y)$ using the k observations. y^* is the predicted value (that comes directly from the k observations in the training set)
- No need to process the training data before making new classifications!

Example

knn() in package class

```
trainX <- train %>%
  select(age, education.num,capital.gain, capital.loss, hours.
trainY <- train$income
incomeknn <- knn(trainX, test=trainX, cl=trainY, k = 10)
confusion <- tally(incomeknn~trainY, format="count")
confusion
```

```
##           trainY
## incomeknn  <=50K  >50K
##     <=50K  18845  3005
##     >50K     897  3302
```

Calculating accuracy

```r
sum(diag(confusion))/nrow(train)
```

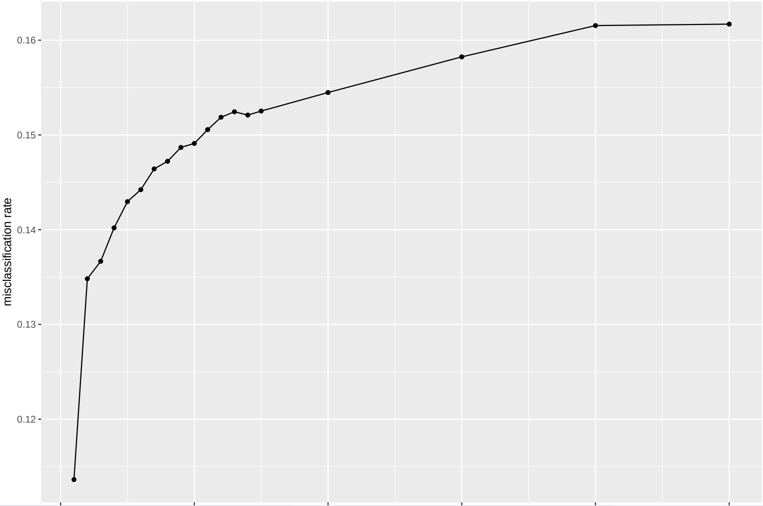
```
## [1] 0.85
```

Observing changes with different k

```
knn_error_rate <- function(x,y, numNeighbors, z=x){
  y_hat <- knn(train =x, test=z, cl=y, k=numNeighbors)
  return(sum(y_hat!=y)/nrow(x))
}
ks <- c(1:15,20,30,40,50)
train_rate <- sapply(ks, FUN=knn_error_rate, x=trainX, y=trai
knn_error_rates <- data.frame(k=ks, train_rates=train_rate)
```

Plotting results

```
ggplot(data=knn_error_rates, aes(x=k,y=train_rate))+geom_point
```



Decision Trees

- Assigns class labels to individual observations where each branch of tree separates data records in more pure class labels through recursive partitioning
- Use Gini coefficient and information gain as the purity criteria

Decision Tree Model

```r
mod_treeCap <- rpart(income ~ capital.gain, data = train)
```

```
n= 26049

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 26049 6307  <=50K (0.75787938 0.24212062)
  2) capital.gain< 5119 24774 5102  <=50K (0.79405829 0.20594171) *
  3) capital.gain>=5119 1275   70  >50K (0.05490196 0.94509804) *
```

Figure 2:

Decision Tree Model Cont.

```
form <- as.formula("income ~ age + workclass +
                    education + marital.status +
                    occupation + relationship +
                    race + sex + capital.gain +
                    capital.loss + hours.per.week")
mod_tree <- rpart(form, data = train)
```

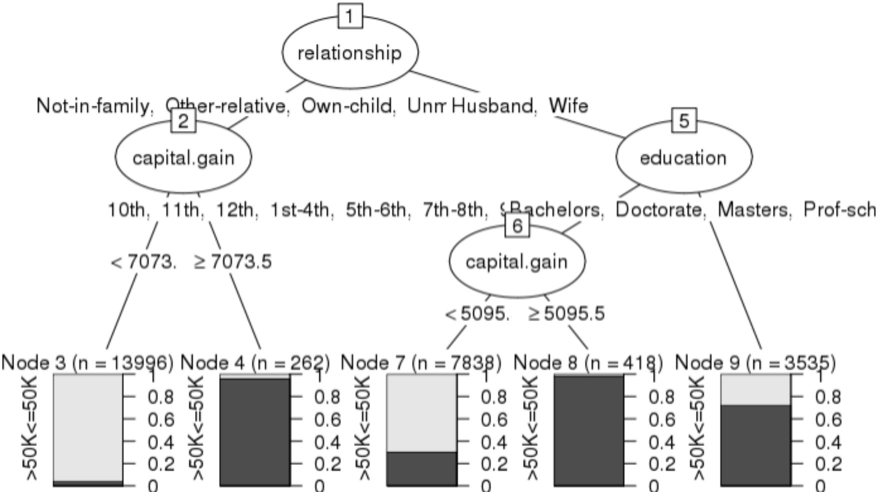
```
n= 26049

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 26049 6307  <=50K (0.75787938 0.24212062)
   2) relationship= Not-in-family, Other-relative, Own-child, Unmarried 14258  957  <=50K (0.93287979 0.06712021)
     4) capital.gain< 7073.5 13996  706  <=50K (0.94955702 0.05044298) *
     5) capital.gain>=7073.5 262   11  >50K (0.04198473 0.95801527) *
   3) relationship= Husband, Wife 11791 5350  <=50K (0.54626410 0.45373590)
     6) education= 10th, 11th, 12th, 1st-4th, 5th-6th, 7th-8th, 9th, Assoc-acdm, Assoc-voc, HS-grad, Preschool, Some-
college 8256 2778  <=50K (0.66351744 0.33648256)
      12) capital.gain< 5095.5 7838 2367  <=50K (0.69800970 0.30199030) *
      13) capital.gain>=5095.5 418    7  >50K (0.01674641 0.98325359) *
     7) education= Bachelors, Doctorate, Masters, Prof-school 3535  963  >50K (0.27241867 0.72758133) *
```

Plotting Decision Tree

```r
plot(as.party(mod_tree))
```



Variable Importance

```
varImp(mod_tree)
```

	Overall
	<dbl>
age	338.23753
capital.gain	2719.31982
capital.loss	307.15574
education	2073.88315
hours.per.week	95.29972
marital.status	1901.00569
occupation	1879.58651
relationship	1929.33273
workclass	0.00000
race	0.00000

Calculating Model Accuracy

```
pred <- predict(mod_tree, test, type = "class")
conf <- table(test$income, pred)
sum(diag(conf))/sum(conf) #accuracy
```

```
## [1] 0.845
```

ROC Curve

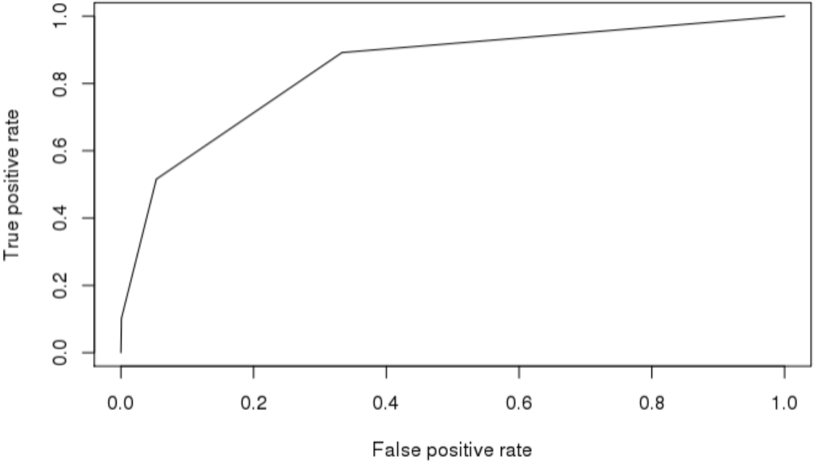
- Receiver Operating Curve (ROC) considers all possible thresholds to predict the number of high-earners ($>$50K)
- Shows trade-off between sensitivity (true positive rate: TPR) and specificity (true negative rate: TNR)

Plotting the ROC Curve

```
income_prob <- predict(mod_tree, newdata=test, type="prob")
perf <- prediction(income_prob[, 2], test$income)
perf <- performance(perf, measure = "tpr",
                    x.measure = "fpr")
```

ROC Curve

```
plot(perf)
```



Random Forest

- Collection of aggregated decision trees
- Constructed process entails:
  - Choosing the number of decision trees (ntree) and number of variables to consider in each tree (mtry)
  - Randomly select data rows with replacement
  - Randomly select mtry variables
  - Build decision tree on resulting data
  - Repeat process ntree times

Random Forest Model

```
mod_forest <- randomForest(formula = form, data = train,
                           ntrain = 201, mtry = 3)
```

```
Call:
 randomForest(formula = form, data = train, ntrain = 201, mtry = 3)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 13.4%
Confusion matrix:
        <=50K  >50K class.error
 <=50K  18423  1319      0.0668
 >50K    2184  4123      0.3463
```

Model Accuracy

```
sum(diag(mod_forest$confusion))/nrow(train)
```

```
## [1] 0.865
```

Variable Importance

```
importance(mod_forest) %>%
  as.data.frame() %>%
  tibble::rownames_to_column() %>%
  arrange(desc(MeanDecreaseGini))
```

rowname <chr>	MeanDecreaseGini <dbl>
relationship	1063
age	1054
capital.gain	1048
education	1015
occupation	859
marital.status	786
hours.per.week	634
capital.loss	333
workclass	329
race	145

Variable Importance Plot

```
varImpPlot(mod_forest, type = 2)
```