

# ITEC 621 Week 5: Machine Learning & Variable Selection

# Training sample

```
library(ISLR)
# Get random sample
set.seed(1)
train=sample(nrow(Auto), nrow(Auto)/2)
train[1:5]
```

```
[1] 105 146 224 354 79
```

```
lm.fit=lm(mpg~horsepower,data=Auto[train,])
summary(lm.fit)
```

```
Call:
lm(formula = mpg ~ horsepower, data = Auto[train, ])
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-13.698	-3.085	-0.216	2.680	16.770

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	40.340377	1.002269	40.25	<2e-16 ***
horsepower	-0.161701	0.008809	-18.36	<2e-16 ***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 4.692 on 194 degrees of freedom
```

Multiple R-squared: 0.6346, Adjusted R-squared: 0.6327  
F-statistic: 336.9 on 1 and 194 DF, p-value: < 2.2e-16

```
#lm.fit=lm(mpg~horsepower,data=Auto,subset=train) # Same thing
```

# Mean Squared Error (MSE)

```
attach(Auto)
true_mpg = Auto$mpg
predicted_mpg = predict(lm.fit, Auto)
# MSE all data:
lm.fit.mse <- mean((true_mpg - predicted_mpg)^2)
lm.fit.mse
```

```
[1] 23.96564
```

```
# MSE training data:
lm.fit.mse.train <- mean((true_mpg - predicted_mpg)[train]^2)
lm.fit.mse.train
```

```
[1] 21.78987
```

```
# MSE test data:
lm.fit.mse.test <- mean((true_mpg - predicted_mpg)[-train]^2)
lm.fit.mse.test
```

```
[1] 26.14142
```

# MSE all is between MSE train and MSE test

```
(lm.fit.mse.train + lm.fit.mse.test)/2
```

```
[1] 23.96564
```

```
lm.fit.mse
```

```
[1] 23.96564
```

```
# This formula works since we did a 50/50 split. In general, this is a weighted sum based on the split.
```

# Cross-Validation (Leave One Out)

```
library(boot) # Has cv.glm()
library(ISLR)

glm.fit=glm(mpg~horsepower,data=Auto)

cv.loo <- cv.glm(Auto,glm.fit)
# Note: delta has 2 numbers and they should be almost identical. If not,
# see below. The first delta value is the actual raw cross-validation MSE. To
# list just the CV MSE:
cv.loo$delta[1]
```

```
[1] 24.23151
```

```
# Let's write a for loop to do 5 polinomials and storing results in a
# vector
# Make a blank array to store results:
cv.error=rep(0,5)
cv.error
```

```
[1] 0 0 0 0 0
```

```
for (i in 1:5){
  glm.fit=glm(mpg~poly(horsepower,i),data=Auto) # fit for polinomial i
  cv.error[i]=cv.glm(Auto,glm.fit)$delta[1] # cv.error for polinomial i
}
```

```
# Check out the vector with MSE values for each of the 5 polynomials  
cv.error
```

```
[1] 24.23151 19.24821 19.33498 19.42443 19.03321
```

# Cross-Validation (K-Folds)

```
cv.10K <- cv.glm(Auto,glm.fit,K=10)
cv.10K$delta[1]
```

```
[1] 18.7671
```

```
set.seed(17)
# Repeat
cv.error.10=rep(0,10)
for (i in 1:10){
  glm.fit=glm(mpg~poly(horsepower,i),data=Auto)
  cv.error.10[i]=cv.glm(Auto,glm.fit,K=10)$delta[1] # 10-Fold validation
}
cv.error.10
```

```
[1] 24.20520 19.18924 19.30662 19.33799 18.87911 19.02103 18.89609
[8] 19.71201 18.95140 19.50196
```



# Variable Selection 1

```
library(ISLR)
# Baseball data
head(Hitters)
```

```
      AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits
-Andy Allanson    293   66     1   30  29   14     1    293    66
-Alan Ashby       315   81     7   24  38   39   14   3449    835
-Alvin Davis      479  130    18   66  72   76    3   1624    457
-Andre Dawson    496  141    20   65  78   37   11   5628   1575
-Andres Galarraga 321   87    10   39  42   30    2    396   101
-Alfredo Griffin  594  169     4   74  51   35   11   4408   1133
      CHmRun CRuns CRBI CWalks League Division PutOuts Assists
-Andy Allanson     1    30   29    14     A         E     446     33
-Alan Ashby        69   321  414   375     N         W     632     43
-Alvin Davis       63   224  266   263     A         W     880     82
-Andre Dawson     225   828  838   354     N         E     200     11
-Andres Galarraga  12    48   46    33     N         E     805     40
-Alfredo Griffin   19   501  336   194     A         W     282    421
      Errors Salary NewLeague
-Andy Allanson    20     NA         A
-Alan Ashby       10  475.0         N
-Alvin Davis      14  480.0         A
-Andre Dawson     3   500.0         N
-Andres Galarraga 4    91.5         N
-Alfredo Griffin  25  750.0         A
```

```
# Get rid of missing values (na)
Hitters = na.omit(Hitters)
```

```
lm.reduced <- lm(Salary ~ AtBat + Hits + Walks, data=Hitters)
summary(lm.reduced)
```

Call:

```
lm(formula = Salary ~ AtBat + Hits + Walks, data = Hitters)
```

Residuals:

Min	1Q	Median	3Q	Max
-1007.5	-245.4	-70.8	165.4	2039.1

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	121.0590	73.2481	1.653	0.09960	.
AtBat	-2.0862	0.6324	-3.299	0.00111	**
Hits	8.9252	1.9922	4.480	1.12e-05	***
Walks	7.1645	1.4097	5.082	7.16e-07	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 386.1 on 259 degrees of freedom

Multiple R-squared: 0.2758, Adjusted R-squared: 0.2674

F-statistic: 32.88 on 3 and 259 DF, p-value: < 2.2e-16

```
lm.full <- lm(Salary ~ AtBat + Hits + Walks + Division + PutOuts,
data=Hitters)
summary(lm.full)
```

Call:

```
lm(formula = Salary ~ AtBat + Hits + Walks + Division + PutOuts,
    data = Hitters)
```

Residuals:

Min	1Q	Median	3Q	Max
-----	----	--------	----	-----

-907.58 -216.89 -62.37 169.87 1965.55

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	173.02126	75.85170	2.281	0.02336	*
AtBat	-2.01116	0.62016	-3.243	0.00134	**
Hits	8.28040	1.95398	4.238	3.15e-05	***
Walks	6.47059	1.38568	4.670	4.87e-06	***
DivisionW	-120.43396	46.83854	-2.571	0.01070	*
PutOuts	0.26542	0.08795	3.018	0.00280	**

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 376.3 on 257 degrees of freedom

Multiple R-squared: 0.3176, Adjusted R-squared: 0.3043

F-statistic: 23.92 on 5 and 257 DF, p-value: < 2.2e-16

```
anova(lm.reduced, lm.full)
```

Analysis of Variance Table

Model 1: Salary ~ AtBat + Hits + Walks

Model 2: Salary ~ AtBat + Hits + Walks + Division + PutOuts

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	259	38613147				
2	257	36385237	2	2227910	7.8682	0.0004824 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Variable Selection 2

```
lm.fuller <- lm(Salary ~ AtBat + Hits + Walks + Division + PutOuts +  
Errors, data=Hitters)  
summary(lm.fuller)
```

Call:

```
lm(formula = Salary ~ AtBat + Hits + Walks + Division + PutOuts +  
Errors, data = Hitters)
```

Residuals:

Min	1Q	Median	3Q	Max
-875.38	-208.93	-61.53	186.63	1980.81

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	183.51751	76.18550	2.409	0.01671	*
AtBat	-1.79757	0.64097	-2.804	0.00543	**
Hits	7.90088	1.97334	4.004	8.17e-05	***
Walks	6.15433	1.40529	4.379	1.74e-05	***
DivisionW	-120.78703	46.77802	-2.582	0.01038	*
PutOuts	0.26459	0.08784	3.012	0.00285	**
Errors	-4.93036	3.81042	-1.294	0.19686	

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 375.8 on 256 degrees of freedom

Multiple R-squared: 0.322, Adjusted R-squared: 0.3061

F-statistic: 20.27 on 6 and 256 DF, p-value: < 2.2e-16

```
anova(lm.full, lm.fuller)
```

### Analysis of Variance Table

Model 1: Salary ~ AtBat + Hits + Walks + Division + PutOuts

Model 2: Salary ~ AtBat + Hits + Walks + Division + PutOuts + Errors

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	257	36385237				
2	256	36148827	1	236409	1.6742	0.1969

# Variable Selection 3

```
library(leaps)
regfit.full=regsubsets(Salary~.,Hitters)
#options(width=140)
summary(regfit.full)
```

```
Subset selection object
Call: regsubsets.formula(Salary ~ ., Hitters)
19 Variables (and intercept)
      Forced in Forced out
AtBat      FALSE      FALSE
Hits       FALSE      FALSE
HmRun      FALSE      FALSE
Runs       FALSE      FALSE
RBI        FALSE      FALSE
Walks      FALSE      FALSE
Years      FALSE      FALSE
CAtBat     FALSE      FALSE
CHits      FALSE      FALSE
CHmRun     FALSE      FALSE
CRuns      FALSE      FALSE
CRBI       FALSE      FALSE
CWalks     FALSE      FALSE
LeagueN    FALSE      FALSE
DivisionW  FALSE      FALSE
PutOuts    FALSE      FALSE
Assists    FALSE      FALSE
Errors     FALSE      FALSE
NewLeagueN FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
```

		AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CA	AtBat	CHits	CHmRun	CRuns
1	( 1 )	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
2	( 1 )	" "	"*	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
3	( 1 )	" "	"*	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
4	( 1 )	" "	"*	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
5	( 1 )	"*	"*	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
6	( 1 )	"*	"*	" "	" "	" "	"*	" "	" "	" "	" "	" "	" "
7	( 1 )	" "	"*	" "	" "	" "	"*	" "	"*	"*	"*	" "	" "
8	( 1 )	"*	"*	" "	" "	" "	"*	" "	" "	" "	"*	"*	"*
		CRBI	CWalks	LeagueN	DivisionW	PutOuts	Assists	Errors	NewLeagueN				
1	( 1 )	"*	" "	" "	" "	" "	" "	" "	" "				
2	( 1 )	"*	" "	" "	" "	" "	" "	" "	" "				
3	( 1 )	"*	" "	" "	" "	"*	" "	" "	" "				
4	( 1 )	"*	" "	" "	"*	"*	" "	" "	" "				
5	( 1 )	"*	" "	" "	"*	"*	" "	" "	" "				
6	( 1 )	"*	" "	" "	"*	"*	" "	" "	" "				
7	( 1 )	" "	" "	" "	"*	"*	" "	" "	" "				
8	( 1 )	" "	"*	" "	"*	"*	" "	" "	" "				

```
# R-squared for each model:
summary(regfit.full)$rsq
```

```
[1] 0.3214501 0.4252237 0.4514294 0.4754067 0.4908036 0.5087146 0.5141227
[8] 0.5285569
```

# Variable Selection 4

```
regfit.full=regsubsets(Salary~.,data=Hitters,nvmax=19)
reg.summary <- summary(regfit.full) # Let's save the summary in an object
reg.summary
```

Subset selection object

Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19)

19 Variables (and intercept)

	Forced in	Forced out
AtBat	FALSE	FALSE
Hits	FALSE	FALSE
HmRun	FALSE	FALSE
Runs	FALSE	FALSE
RBI	FALSE	FALSE
Walks	FALSE	FALSE
Years	FALSE	FALSE
CAtBat	FALSE	FALSE
CHits	FALSE	FALSE
CHmRun	FALSE	FALSE
CRuns	FALSE	FALSE
CRBI	FALSE	FALSE
CWalks	FALSE	FALSE
LeagueN	FALSE	FALSE
DivisionW	FALSE	FALSE
PutOuts	FALSE	FALSE
Assists	FALSE	FALSE
Errors	FALSE	FALSE
NewLeagueN	FALSE	FALSE

1 subsets of each size up to 19

Selection Algorithm: exhaustive

AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns





```
18 ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*"
19 ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "
```

```
#options(width=80)
names(reg.summary) # Check out the subset data names
```

```
[1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"
```

```
reg.summary$rsq # Let's inspect R-squares
```

```
[1] 0.3214501 0.4252237 0.4514294 0.4754067 0.4908036 0.5087146 0.5141227
[8] 0.5285569 0.5346124 0.5404950 0.5426153 0.5436302 0.5444570 0.5452164
[15] 0.5454692 0.5457656 0.5459518 0.5460945 0.5461159
```

```
# Let's plot the subset stats
par(mfrow=c(2,2)) # 2 x 2 layout

# RSS
plot(reg.summary$rss,xlab="Number of Variables",ylab="RSS",type="l")

# Adjusted R-Square
plot(reg.summary$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",
type="l")

which.max(reg.summary$adjr2)
```

```
[1] 11
```

```
points(11,reg.summary$adjr2[11], col="red",cex=2,pch=20)

# Cp
```

```
#plot(reg.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')
```

```
# BIC
```

```
#plot(reg.summary$bic,xlab="Number of Variables",ylab="BIC",type='l')
```

