BattishaPset1

#Statistical and Machine Learning

A key difference between supervised and unsupervised ML is in their relationship between the X's and Y's. While in supervised learning there is an associated and measured response Y for each predictor X, in unsupervised learning there is no associated response Y for each predictor X. What this means in practice is that supervised ML algorithms have pre-labeled, corresponding input and output data pairs that can be used to train the algorithm to learn the function describing the relationship between the inputs and outputs. Meanwhile, in unsupervised ML algorithms, there is no pre-existing output data, data structure or labeling, which means that the input variables have no given corresponding output variables. Instead, unsupervised ML algorithms must first discover the structure for the input variables, and then proceed to predict the outcome based on the patterns it identified within the input data.

In simpler terms, ML algorithms have access to previous similar cases where the "right" outputs to the given inputs are known. All it must do from there, is utilize these previous cases to identify the "right" outputs for inputs it hasn't seen yet, in a case similar to its training cases. Meanwhile, unsupervised learning algorithms don't have access to the "right" outputs for its given inputs; it's seeing these inputs for the first time. It must therefore make sense of the given inputs without guidance, and then determine identifying characteristics for the given inputs so that it can replicate the process with future inputs.

Because of the distinct nature of the two ML types, the aims we have when using each type also differ. Because supervised ML algorithms already have labels and structure for their data, their primary goal is to effectively map each X predictor to its proper Y response. As such, supervised ML algorithms are generally used for Regression and Classification problems, where each X is either classified into its proper Y bin (where bins are discrete), or correlated to a certain Y value corresponding to a fitted regression line (where the Y values on the regression line are continuous). On the other hand, because unsupervised ML algorithms don't have labeled or structured data, its primary objective to make out some structure from the input data. It generally does this through Clustering–where the input data is clustered into distinct groups based on certain common features–or Association–where the algorithm seeks to map out the relationships between the various input data parts.

The different nature of the two ML types has consequential ramifications for data collection and data generation strategies. Because supervised ML requires data to be pre-labeled, structured, and have outputs, data generation for supervised ML must be rigorous, comprehensive and well-structured. For example, if one was running a supervised ML algorithm to accurately classify whether a tumor was malignant, they would need to generate a dataset containing the input (the tumors themselves), as well as the output (whether they were malignant or not), and accurately label and structure the dataset so that it the algorithm can be effectively trained on it. On the other hand, because unsupervised ML deals with unstructured and unlabeled data, and primarily seeks to structure the data, the data generation process doesn't require as much structuring beforehand. For example, if one was running a unsupervised ML algorithm that sought to distinguish fruits from one other, they wouldn't need to label their fruits beforehand, as that would be the task of the unsupervised algorithm.

```
library(dplyr)
```

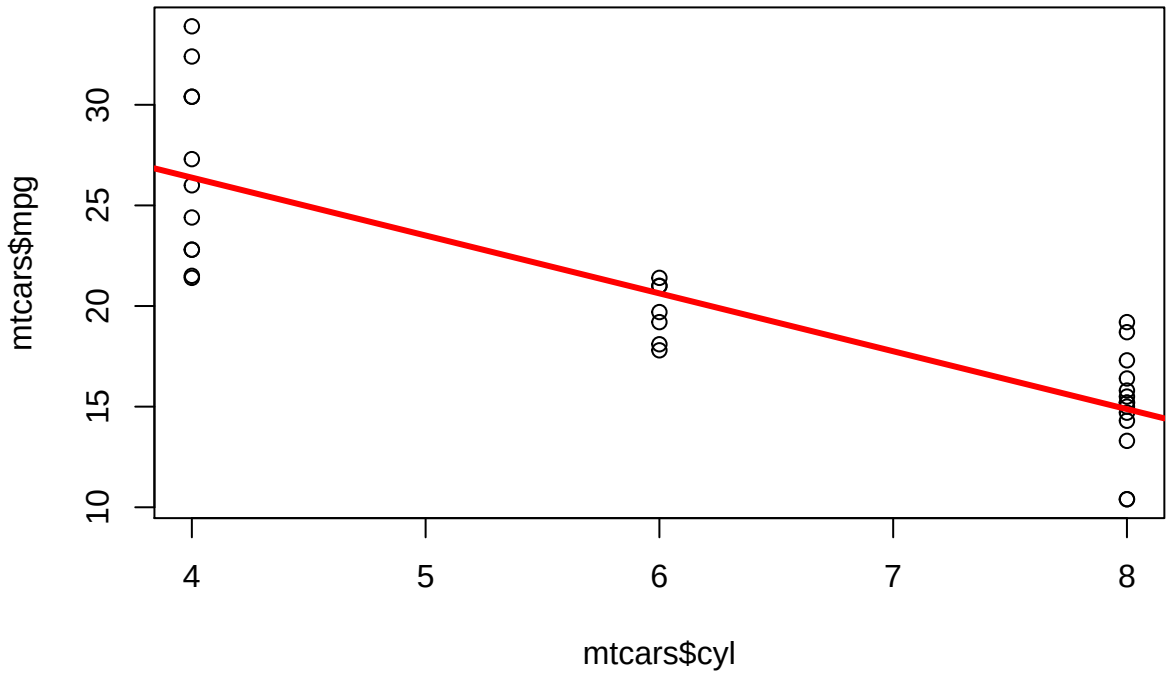
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

#Linear Regression Regression

1a.Linear prediction model for miles per gallon as a function of cylinders shown below:

```
#Plot Coordinates of Data
plot(mtcars$cyl,mtcars$mpg)

#Plot LS Regression Line
abline(lm(mpg ~ cyl, data=mtcars), col="red", lwd=3)
```




```
#Obtain Summary Regression Data
reg <- lm(mpg ~ cyl, data=mtcars)
summary(reg)
```

```
##
## Call:
## lm(formula = mpg ~ cyl, data = mtcars)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9814 -2.1185  0.2217  1.0717  7.5186
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.8846     2.0738   18.27  < 2e-16 ***
## cyl          -2.8758     0.3224   -8.92 6.11e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.206 on 30 degrees of freedom
## Multiple R-squared:  0.7262, Adjusted R-squared:  0.7171
## F-statistic: 79.56 on 1 and 30 DF,  p-value: 6.113e-10
```

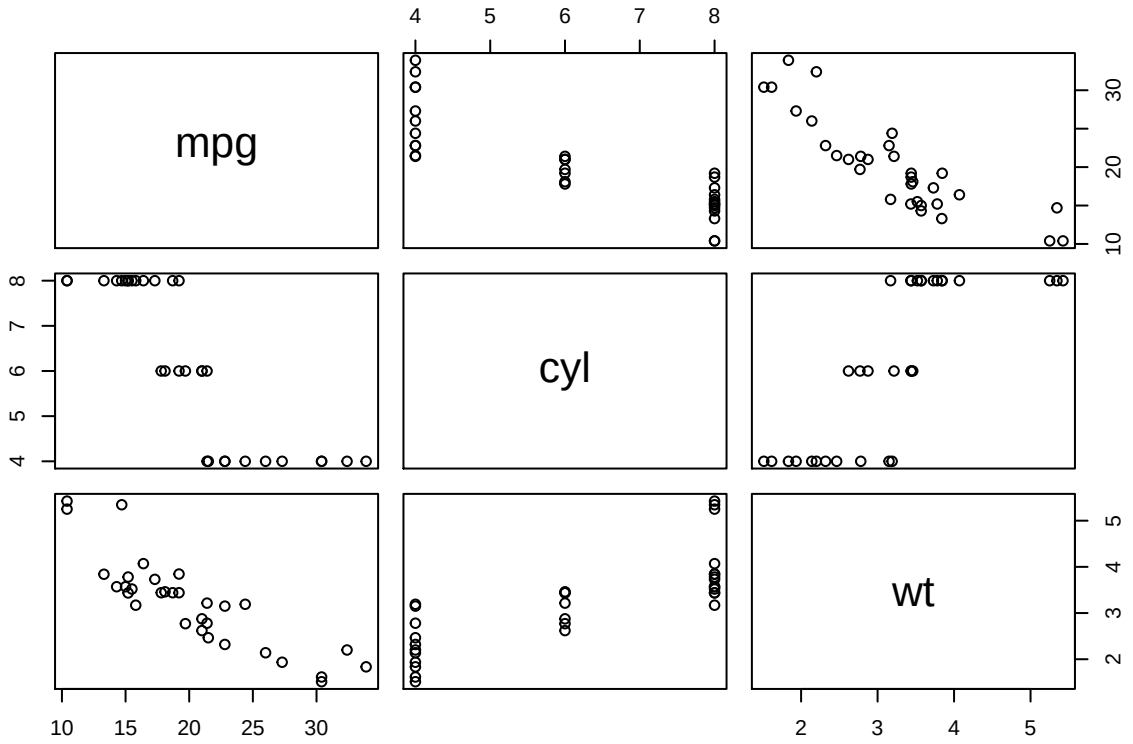
1a (continued). Assuming the linear regression line plotted above follows the form Yi=B0+B1*Xi+Ei, where Y is miles per gallon, X is cylinders, and E is the random error term, the parameters are B0 (in this case, the Y-Intercept) and B1 (in this case, the slope). The value for B0 is 37.885, and the value for B1 is -2.876. The output for this model will be the dependent variable, which is miles per gallon, given a specific input of the independent variable, which is cylinders.

The graph of the linear regression line shows that an increase in cylinders maps on to a decrease in mpg. Furthermore, because cars typically have either 4, 6 or 8 cylinders, all the points lie on one of those three x values.

1b. Given the form Yi=B0+B1*Xi+Ei, the statistical form of this simple model will be Yi=37.885+(-2.876)Xi+Ei, where Yi is miles per gallon, Xi is cylinders and Ei is irreducible error.

1c.Multiple Regression Model shown below:

```
#Plot Relevant Variables
plot(select(mtcars,mpg,cyl,wt))
```




```
#Perform Multiple Linear Regression and Obtain Output
multireg <- lm(mpg ~ cyl+wt,data=mtcars)
summary(multireg)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2893 -1.5512 -0.4684  1.5743  6.1004
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.6863     1.7150  23.141  < 2e-16 ***
## cyl          -1.5078     0.4147  -3.636 0.001064 **
## wt           -3.1910     0.7569  -4.216 0.000222 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.568 on 29 degrees of freedom
## Multiple R-squared:  0.8302, Adjusted R-squared:  0.8185
## F-statistic: 70.91 on 2 and 29 DF,  p-value: 6.809e-12
```

1c(continued). The multiple linear regression relying on both vehicle weight and cylinders predicted the miles per gallon of the sample vehicles better than the simple linear regression relying only on cylinders.

This is evident in the R-squared values; while the simple linear regression had an R-squared value of 0.7262, the multiple linear regression had a higher R-squared value of 0.8302, showing that the second model was more strongly correlated to the data. Furthermore, the very low p-value of 6.809e-12 (significant to level 0), illustrates that the correlation was very statistically significant, and that it is unlikely that this correlation occurred by random chance.

There was also a clear adjustment in coefficient size between the two models. While in the initial linear model, a car was predicted to lose 2.8758 mpg for every additional cylinder that it had, in the multiple linear regression, a car was predicted to lose only 1.5078 mpg for every additional cylinder that it had, making the size of the cylinder coefficient smaller. On the other hand, while the initial linear model didn't take weight into consideration (the size of the weight coefficient was 0), in the multiple linear model, a car was predicted to lose 3.1910 mpg for every additional 1000 lbs in its weight, making the size of the weight coefficient larger.

1d.Interacted Weight and Cylinder Regression results shown below:

```
#Obtain Results for Interacted Weight and Cylinder
interact.multireg <- lm(mpg ~ cyl+wt+cyl*wt,data=mtcars)
summary(interact.multireg)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + wt + cyl * wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2288 -1.3495 -0.5042  1.4647  5.2344
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  54.3068     6.1275   8.863 1.29e-09 ***
## cyl          -3.8032     1.0050  -3.784 0.000747 ***
## wt           -8.6556     2.3201  -3.731 0.000861 ***
## cyl:wt        0.8084     0.3273   2.470 0.019882 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.368 on 28 degrees of freedom
## Multiple R-squared:  0.8606, Adjusted R-squared:  0.8457
## F-statistic: 57.62 on 3 and 28 DF,  p-value: 4.231e-12
```

1d(continued). One change in these results compared to the previous results were the coefficients for (the non-interacted) cylinder and weight variables–both increase in size. Though the R-squared value does increase in the interacted test from the previous test(from 0.8302 to 0.8606), it doesn't increase by a great amount (roughly 0.03), showing that this test is better correlated to our data but not by much. While the interaction term is very statistically significant (to a level 0.01), it is not as statistically significant as the weight and cylinder terms (which are both significant to a level 0.00).

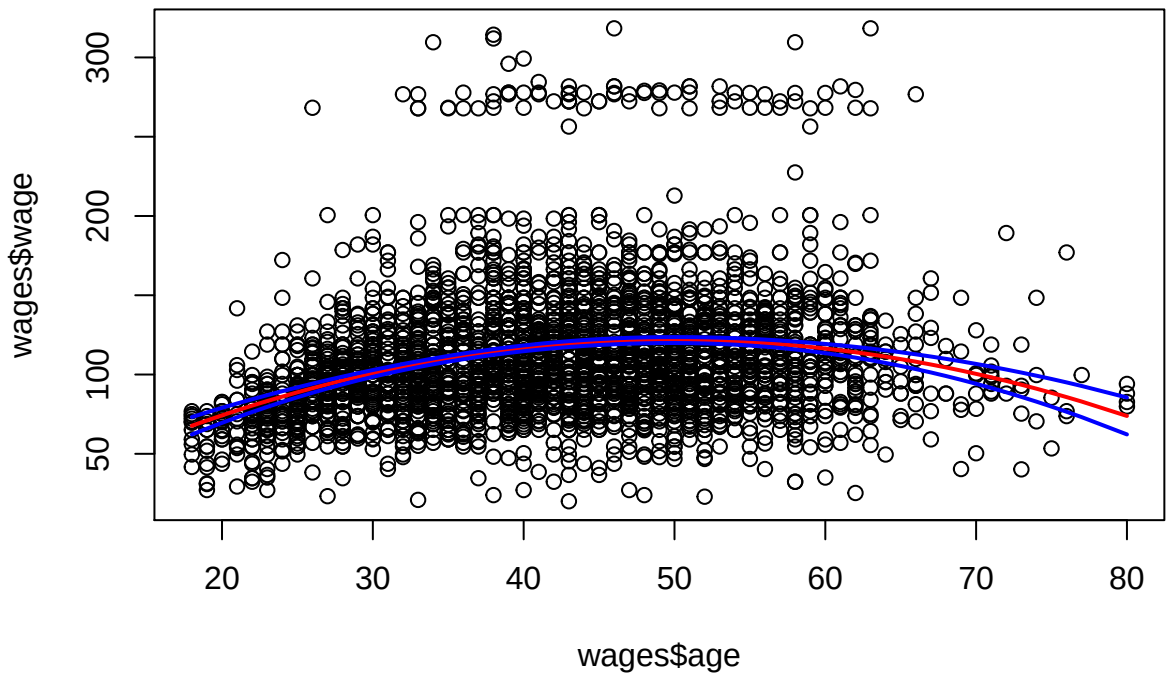
By including a multiplicative interaction in the term we are theoretically asserting that the two independent variables (in this case, cylinders and weight) interact with each other in a way that affects our resulting outcome (miles per gallon). Since the impact of the independent variables on each other cannot be taken into account in a solely additive formula, we must multiply them together to take their impact on each other into account.

#Non Linear Regression

```r
#Import and Plot Wages Data
wages <- read.csv("wage_data.csv")
plot(wages$age,wages$wage)

#Create Polynomial Regression
polyreg <- lm(wages$wage ~ poly(wages$age, degree=2, raw=T))
summary(polyreg)
```

```
##
## Call:
## lm(formula = wages$wage ~ poly(wages$age, degree = 2, raw = T))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.126 -24.309  -5.017  15.494 205.621
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         -10.425224   8.189780  -1.273    0.203
## poly(wages$age, degree = 2, raw = T)1   5.294030   0.388689  13.620   <2e-16
## poly(wages$age, degree = 2, raw = T)2  -0.053005   0.004432 -11.960   <2e-16
##
## (Intercept)
## poly(wages$age, degree = 2, raw = T)1 ***
## poly(wages$age, degree = 2, raw = T)2 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.99 on 2997 degrees of freedom
## Multiple R-squared:  0.08209,    Adjusted R-squared:  0.08147
## F-statistic:   134 on 2 and 2997 DF,  p-value: < 2.2e-16
```

```r
wagesreg <- lm(wage ~ age, data=wages)


#Graph Polynomial Regression Line in Red
lines(smooth.spline(wages$age, predict(polyreg)), col="red", lwd=2)

#Graph Confidence Interval Lines in Blue
confidence<-predict(polyreg, interval="confidence", level=.95)
lines(smooth.spline(wages$age, confidence[,"lwr"]), col="blue", lwd=2)
lines(smooth.spline(wages$age, confidence[,"upr"]), col="blue", lwd=2)
```



1a. Fitted Polynomial regression is shown above. The results reveal a second-order polynomial regression of the statistical form: Yi=10.425224+(5.294030)Xi+(-0.053005)Xi^2. The test is very statistically significant with a p-value of 2.2e-16. The correlation of the regression to the test values is also high, with a R-squared value of 0.8209.

1b. The function is plotted above in red and its 95% confidence interval values are plotted above in blue.

1c. Substantively, I see that the pattern of one's wage increasing with their age holds true till the age of 50. After the age of 50, our data points become less dense (likely as a result of death or retirement), and as age enters the 60s and 70s only the data points receiving lower wages continue to appear (probably because only those with lower wages were not able to save enough to retire and had to continue working). I also see another, relatively linear pattern in the datapoints at the top. There seems to be a large gap between those earning less than $200 an hour and those earning more than $250 an hour, revealing that very few people earn between $200 and $250 an hour. With the group earning above $250 an hour, wages seem to be relatively constant throughout the lifetime, and retirement seems to occur in the ages of 60-65.Finally the confidence intervals show that there is much more uncertainty regarding wages for the ages of 60-80 than for the rest of the line, showing greater variance in the end of the spectrum.

By plotting a polynomial regression, we are asserting that the relationship between age and wages is not perfectly linear; rather, its derivative (specifically, its rate of increase or decrease) either increases or decreases over time. In this particular case, the polynomial regression shows that on average in our dataset, wages increase till the age of 50 and then decrease afterwards.

1d. From a statistical perspective, a polynomial regression differs from a linear regression in 2 key ways. First, a polynomial regression will provide a more accurate approximation of of the relationship between the input and the output in the dataset. Rarely are correlations perfectly represented by a linear model; because non-linear models allow for curvature in their regression lines, they can more accurately describe the relationship. Furthermore, in the off-chance that a dataset is represented by a perfectly linear model,

a polynomial regression can attach a 0 coefficient to its exponentialized term and act as a linear model. However, a polynomial regression is also more sensitive to outliers. Because non-linear regressions are more flexible than linear regressions, they can curve to adhere to specific outliers in the data set, which can lead to overfitting.

From a substantive perspective, a non-linear model can better represent a wider variety of correlations that are found in the real world. For fields like population and economics, where most growth or decrease happens at an exponential rate, a linear regression will serve as an inaccurate representor of the underlying phenomenon. Furthermore, for many fields, the change in rate of growth is an important metric to identify. Since the derivative of any linear regression will be constant, non-linear or polynomial regressions can be much more useful in these scenarios.