William Ansehl
Intro to Machine Learning Pset 1 Essay

Technologies significant to the development of the next decade include autonomous cars, 5G, cloud computing, cybersecurity, blockchain and artificial intelligence among others. In particular, AI is central to the development of many innovative technologies that will drive innovation and disrupt industries. Yet the general premise of AI is not as impossible to understand as many might believe. In fact, the algorithms behind AI systems can mostly be broken down into one of two subgroups - supervised or unsupervised machine learning. AI simply incorporates the two in order to create more complex systems. Yet for the purposes of better understanding machine learning at its base, it is important to better understand the subgroups independently.

Supervised machine learning is akin to the creator of the algorithm guiding how the algorithm analyzes data - like a teacher and a student. Supervised machine learning involves knowing the input variables X and the output variables Y before an algorithm is run. The algorithm ultimately decides the mapping function that best maps X to Y. This sequence is essentially $Y = f(X)$. The goal of the algorithm is to define a best fit mapping function that can accurately predict the outcome of inputting new data not contained within the originally processed dataset. This notion contains the idea of a training data set that trains the algorithm versus a testing data set which helps the creator understand the accuracy of the model when new data is introduced. A key feature of supervised machine learning is the creator's ability to correct the model such that it makes more accurate predictions.

One example of supervised machine learning is predicting the selling price of a house based on the number of bedrooms, number of bathrooms, square footage, number of garage spaces and so on. This might be considered a regression problem. The algorithm is trained with input data (organized by features) and output data. When a new house is listed in the dataset with a set number of bedrooms, bathrooms etc, the algorithm can accurately predict the property's selling price based on the model fitted to prior data. Classification is another common type of supervised machine learning. One such example is an algorithm that identifies email spam. In this case, the algorithm is trained with a dataset containing features of different emails and whether or not they are spam. From this dataset, the algorithm can then filter out spam emails when it identifies new emails as such. Regression problems typically output variables as values such as price while classification problems output categorical data such as spam or not spam.

Unsupervised machine learning allows the model to discover its own relationships within the data set without predetermined input variables from the creator of the algorithm. In this manner, Unsupervised ML algorithms can work with unlabeled data and can be more complex than supervised machine learning. Due to this tradeoff, supervised machine learning is often much more easily interpretable than unsupervised machine learning. As a benefit, unsupervised machine learning can use unlabeled data from a computer rather than manually processed data. These types of algorithms can find unknown relationships in the data unbeknownst to the creator of the algorithm themselves. Unlike supervised machine learning, unsupervised machine learning does not connect input data to output data. Instead, learning takes place in real time as opposed to all at once as is the case in supervised ML.

Unsupervised ML can be broken down into clustering and association problems. Clustering algorithms are used when trying to identify groups based on behaviour - such as how

google clusters articles and hyperlinks based on similarities to what the user inputs in the search bar. Association algorithms are used to discover relationships between data objects in large databases. One example is associating groups of online shoppers to certain products they might like, based on the shopper's browser history. An unsupervised ML algorithm's main goal is to find the underlying distribution and structure the database.

Supervised learning produces an output from previous experience, or "well-labeled" training data. Unsupervised learning discovers various types of unknown patterns in unlabeled data in real time. Yet they both have their own drawbacks. Supervised learning struggles with processing large data sets while unsupervised learning lacks ease of interpretability. Understanding the problem is crucial to deciding which type of machine learning to utilize in order to provide key insights and solve the challenge.

Intro to ML Pset 1

William Ansehl

1/16/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
## Linear Regression
# part (a)

model <- lm(mpg ~ cyl, data = mtcars)
summary(model)
```

```
##
## 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
```

```
# Beta_0 = 37.8846
# Beta_1 = -2.8758
# p-value for cyl implies we reject the null that claims it is insignificant, implying that
#cyl is significant in regressing for mpg
plot(mtcars$cyl, mtcars$mpg)
abline(model)

# part (b)
# mpg = 37.8846 + (-2.8758)*cyl

# part (c)
```

```r
model_2 <- lm(mpg ~ cyl + wt, data = mtcars)
summary(model_2)
```

```
##
## 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
```

```r
# First off, the adjusted r^2 increased from .7171 to .8185 which indicates a substantial
#increase in the amount of variance of the data explained by the model.
# the beta value for "cyl" also increased substantially from -2.8785 to -1.5078, especially in
#comparison to its original standerd error (0.3224). This indicates some interaction or
#colinearity between the two regressors.
# in either model, we are still rejecting the notion that the cyl regressor is insignificant,
#impying that it is significant in predicting mpg. We also reject the null for wt, indicating
#that wt is also a significant regressor.
# note: I discussed adjusted r^2 because it accounted for the change in degrees of freedom when
#adding an additional regressor to the model.

# part (d)
model_3 <- lm(mpg ~ cyl + wt + cyl*wt, data = mtcars)
summary(model_3)
```

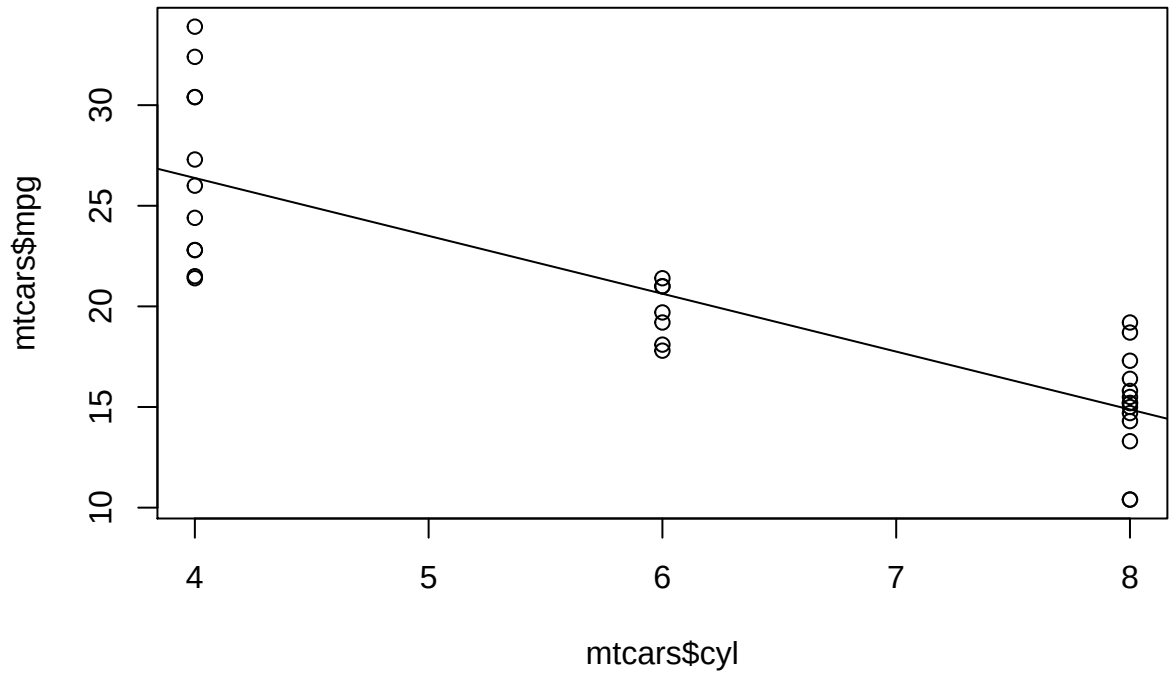
```
##
## Call:
## lm(formula = mpg ~ cyl + wt + cyl * wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2288 -1.3495 -0.5042  1.4647  5.2344
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  54.3068     6.1275   8.863 1.29e-09 ***
## 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
```

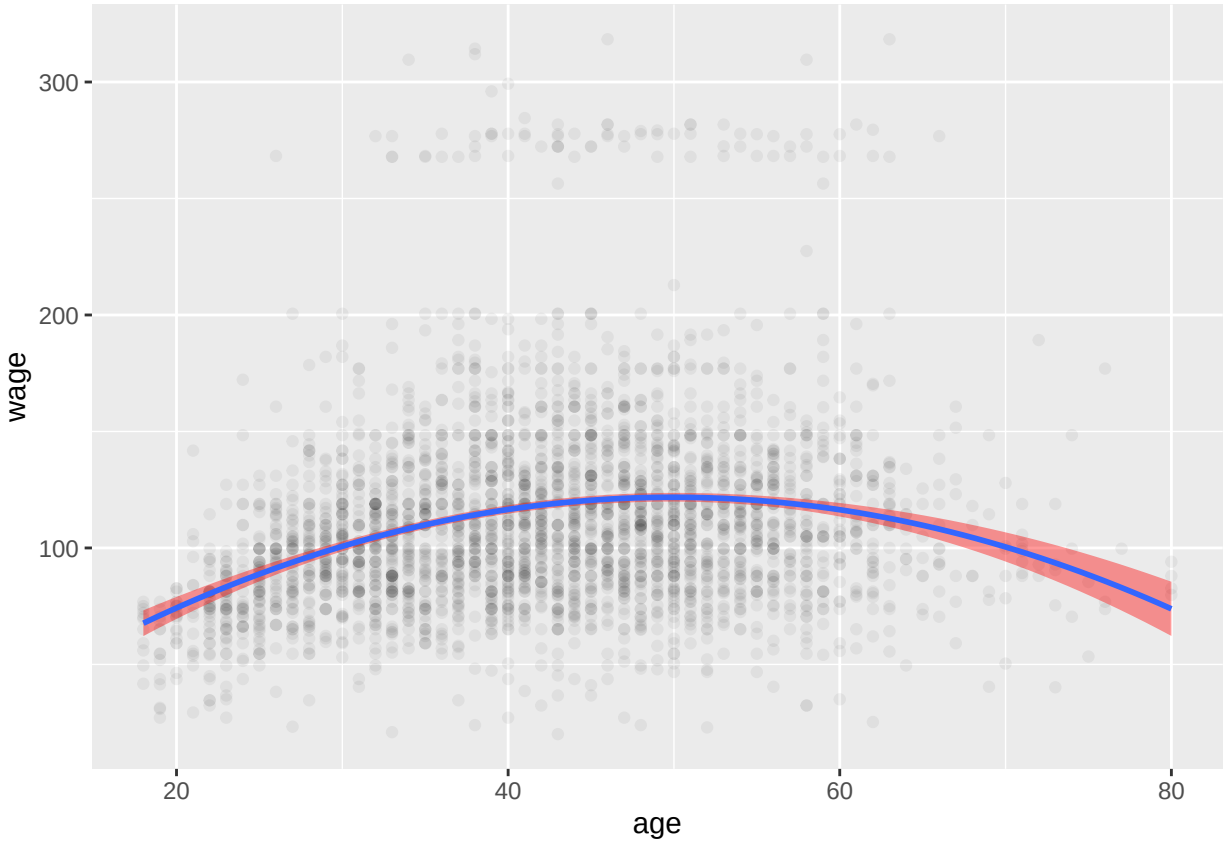
```
# the interaction term implies that there is a relationship between cyl and wt. In other words,
#the effect of cyl on mpg changes depending on the value of wt and vice versa.
# We see another substantial change in r^2 from 0.8185 to 0.8457. This implies that a the
#interaction term helps explain away a bit more of the variance in the mpg data.
# We are still rejecting the null for all the regressors, including the interaction term. This
#implies that they are all significant.
# both beta_cyl and beta_wt change drastically again in comparison to their standard eror. This
#implies there might be some colinearity between them.

## Non-Linear Regression
wage_data <- read.csv('wage_data.csv')

# part (a)
poly_model <- lm(wage ~ age+I(age^2), data = wage_data)
summary(poly_model)
```

```
##
## Call:
## lm(formula = wage ~ age + I(age^2), data = wage_data)
##
## 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
## age           5.294030   0.388689  13.620   <2e-16 ***
## I(age^2)     -0.053005   0.004432 -11.960   <2e-16 ***
## ---
## 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^2 doesn't mean anything anymore because this is no longer a linear model
# Beta_0 = -10.425224
# Beat_1 = 5.294030
# Beta_2 = -0.053005
# wage = -10.425224 + 5.294030*age - 0.053005*age^2
# according to the p-values for age and age^2, we reject the null hypotheses that age and age^2
#are insignificant, implying that they both have bearing on wage


# part (b)
library(ggplot2)
```




```
ggplot(wage_data, aes(y=wage, x=age)) +
  geom_point(alpha = .05) +
  stat_smooth(method = "lm",fill = 'red', formula = y ~ poly(x,2))
```




```
# part (c)
# We see that the 95% confidence interval is pretty small around the polynomial line.
```

```
# We can interpret this as there being a lot of data points, not easily seen, centered at the
#polynomial line.
# In fact, if we look at the number of rows in the data set, there are 3000.
# by fitting a polynomial regression to the data, we are asserting that the plotted data does
#not represent a linear relationship
# and the distribution of the data is more complex.
# According to the bias-variance tradeoff, a linear model would underfit non-linear data,
#resulting in high bias.
# As such, it might be better to fit a polynomial model to better fit the data (barring
#over-fitting)

# part (d)
# a linear regression model must follow the following assumptions:
# 1. Linear relationship
# 2. Multivariate normality
# 3. No or little multicollinearity
# 4. No auto-correlation
# 5. Homoscedasticity (uniform variance)
# if the data violates one or more of these assumptions, transformations can help reshape the
#data for better predictability.
# Strictly speaking, polynomial regression is still linear regression due to its linearity in
#the regression coefficients:
# B_0, B_1, B_2, ...
# That said, there are still clear differences between the two regressions. A plot of the
#residuals of the data in a linear model, for instance,
# can indicate the existence of a non-linear relationship in the data. To account for this
#non-linear relationship,
# a polynomial regression of order n can be used to reduce bias and better fit the data.
# non-linear models are more flexible than linear models due to it's lack of need to follow set
#rules or assumptions.
# Consequently, non-linear models have the possibility of overfitting the data and curving too
#much.
# In evaluating non-linear models, r^2 and p-values can no longer be used. Other metrics such
#as MSE are used
# to evaluate the model's fit to the data.
```