Machine Learning - Day 2

Data modelling

Mariachiara Fortuna

Libraries

```
library(dplyr)
library(ggplot2)
library(here)
```

DATA IMPORT

```
dc <-  file.path(here(), "data", "dc-wikia-data.csv") %>%
  read.csv(na.strings = "")

max_year <- max(dc$YEAR, na.rm =T)

dc <- dc %>%
  mutate(active_years = max_year - YEAR)
```

DATA MODELING

LINEAR REGRESSION

Simple linear regression

Easy ex

```
ggplot(data = cars, aes(x = dist, y = speed)) +
  geom_point() +
  geom_smooth(method='lm',formula=y~x)

m1 <- lm(data = cars, speed ~ dist)

summary(m1)

residuals(m1)
predict(m1)

plot(m1)
```

Comics example

```r
dc_small <- dc %>%
  filter(APPEARANCES >20)

dc %>%
  ggplot(aes(x = active_years, y = APPEARANCES)) +
  geom_point()

dc_small %>%
  ggplot(aes(x = active_years, y = APPEARANCES, col = ID)) +
  geom_point()
```

```r
m1 <- lm(data=dc_small, APPEARANCES ~ active_years)

summary(m1)

plot(m1)
```

Multiple linear regression

```r
m3 <- lm(data=dc, APPEARANCES ~ active_years + ALIGN)

summary(m3)

plot(m3)
```

```r
m4 <- lm(data=dc, APPEARANCES ~ active_years + ALIGN + active_years*ALIGN)

summary(m4)

plot(m4)
```

Log-level regression

http://www.cazaar.com/ta/econ113/interpreting-beta

```r
dc %>%
  ggplot(aes(x = APPEARANCES)) +
  geom_density()
```

```r
dc %>%
  ggplot(aes(x = log(APPEARANCES))) +
  geom_density()
```

```r
dc %>%
  ggplot(aes(x = active_years, y = log(APPEARANCES), col = ID)) +
  geom_point()
```

```r
m2_l <- lm(data=dc, log(APPEARANCES) ~ active_years)

summary(m2_l)

plot(m2_l)
```

Se aggiungiamo un altro anno di attività, ci aspettiamo che il numero di apparizioni cresca del 3%

```
m3 <- lm(data=dc, log(APPEARANCES) ~ active_years + ALIGN)

summary(m3)

plot(m3)
```

Se aggiungiamo un altro anno di attività, ci aspettiamo che il numero di apparizioni cresca del 3%. Se il personaggio è cattivo, però, ci aspettiamo un decremento del numero di apparizioni del 17%, mentre se è buono un incremento del 47%.

https://www.youtube.com/watch?v=wXC2kViEGz8

```
m4 <- lm(data=dc, log(APPEARANCES) ~ active_years + ALIGN + active_years*ALIGN)

summary(m4)

plot(m4)
```

LOGISTIC REGRESSION

https://datascienceplus.com/perform-logistic-regression-in-r/

```
dc_class <- dc %>%
  filter((ALIGN == "Bad Characters" | ALIGN == "Good Characters") &
           (SEX == "Female Characters" | SEX == "Male Characters")) %>%
  select(name, ALIGN, SEX, APPEARANCES, active_years) %>%
  na.omit

l1 <- glm(data = dc_class, ALIGN ~ SEX, family = "binomial")

summary(l1)

post_l1 <- predict(l1, type = "response")

ALIGN_pred <- ifelse(post_l1>0.5,  "Good Characters", "Bad  Characters")

misClasificError <- mean(ALIGN_pred != dc_class$ALIGN)
print(paste('Accuracy',1-misClasificError))

table(dc_class$ALIGN)
table(ALIGN_pred)

table(dc_class$ALIGN, ALIGN_pred)
```

```
ggp <- ggplot(data = dc_class, mapping = aes(x = active_years, y = ALIGN)) +
  geom_point(colour="blue") +
  #geom_line(mapping = aes(x = active_years, y = ALIGN), colour="red") +
  facet_wrap(facets = ~SEX)
print(ggp)

l2 <- glm(data = dc_class, ALIGN ~ SEX + active_years, family = "binomial")
```

```r
summary(l2)

post_l2 <- predict(l2, type = "response")

ALIGN_pred_l2 <- ifelse(post_l2>0.5,  "Good Characters", "Bad  Characters")

misClasificError <- mean(ALIGN_pred_l2 != dc_class$ALIGN)
print(paste('Accuracy',1-misClasificError))

table(dc_class$ALIGN, ALIGN_pred_l2)

anova(l2, test="Chisq")
```

```r
l3 <- glm(data = dc_class, ALIGN ~ SEX + active_years + APPEARANCES, family = "binomial")

summary(l3)

post_l3 <- predict(l3, type = "response")

ALIGN_pred_l3 <- ifelse(post_l3>0.5,  "Good Characters", "Bad  Characters")

misClasificError <- mean(ALIGN_pred_l3 != dc_class$ALIGN)
print(paste('Accuracy',1-misClasificError))

table(dc_class$ALIGN, ALIGN_pred_l3)

anova(l3, test="Chisq")
```

DECISION TREE

https://gormanalysis.com/decision-trees-in-r-using-rpart/

```r
library(rpart)

table(iris$Species)

iris_tree <- rpart(Species ~ ., method = "class", data = iris)

print(iris_tree)

summary(iris_tree)

plot(iris_tree, compress = T, margin = 0.2, branch = 0.3)
text(iris_tree, use.n = T, digits = 3, cex = 0.8)

printcp(iris_tree)

iris_pred <- predict(iris_tree, type = "class")

table(iris_pred, iris$Species)

misClasificError <- mean(iris_pred != iris$Species)
```

```r
print(paste('Accuracy',1-misClasificError))
```

```r
align_tree <- rpart(ALIGN ~ SEX + active_years + APPEARANCES, method = "class", data = dc_class)

summary(align_tree)

plot(align_tree, uniform = T, compress = T, margin = 0.2, branch = 0.3)
text(align_tree, use.n = T, digits = 3, cex = 0.6)

align_pred_tree <- predict(align_tree, type = "class")

table(align_pred_tree, dc_class$ALIGN)

misClasificError <- mean(align_pred_tree != dc_class$ALIGN)
print(paste('Accuracy',1-misClasificError))
```

Training and test dataset

```r
train <- sample(nrow(dc_class), 4800)
dc_train <- dc_class[train,]
dc_test <- dc_class[-train,]
```

CLUSTER

```r
dc_full <- dc %>%
  select(name, ALIGN, EYE, HAIR, SEX, ALIVE, APPEARANCES, active_years, YEAR) %>%
  na.omit()

dc_cluster <-dc_full %>%
  select(active_years, APPEARANCES) %>%
  scale() %>%
  as.data.frame()


# Finding cluster number through Within groups sum of squares
wss <- (nrow(dc_cluster)-1)*sum(apply(dc_cluster,2,var))
for (i in 2:15){
  wss[i] <- sum(kmeans(dc_cluster, centers=i)$withinss)
  }
plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")
# Let's try 4

set.seed(1234)

# K-Means Cluster Analysis
fit <- kmeans(dc_cluster, 4) # 4 cluster solution
# append cluster assignment
dc_full <- data.frame(dc_full, cluster = as.factor(fit$cluster))
```

```r
# CLUSTERS MEANS

# Stats
cluster_stats <- dc_full %>%
  group_by(cluster) %>%
  summarise(count = n(),
            perc = paste0(round(n()/nrow(dc_full)*100,2),"%"),
            avg_appear = mean(APPEARANCES),
            avg_year = mean(active_years))

### LABELS MAY BE DIFFERENT!!

dc_full <- dc_full %>%
  mutate(label = case_when(
    cluster == 1 ~ "Stabili",
    cluster == 2 ~ "Star",
    cluster == 3 ~ "Comparse",
    cluster == 4 ~ "Secondari"
  ))

ggplot(data = dc_full, aes(x = active_years, y = APPEARANCES, col = label)) +
  geom_point()

star <- dc_full %>%
  filter(label == "Star")
```