Introduction to Machine Learning

Contents

Chapter 1: What is Machine Learning

Introduction

Machine Learnng:

- Explores the construciton and usage of algorithms.
- Improves performance as it receives **more** infomrtion.
- Experience comes from observations on how particular problems have been previously solved.

No matter what algorithm used, the primary concept for Machine Learning is **imput knowledge** or **Data**. Typically this data is a **dataset** containing a number of observations, each having a number of well defined variables (often called features) :

size	edge	color
small	dotted	green
big	striped	yellow
medium	normal	green

From the figure above, we see that each square (row) and its corresponding color is an **observations**. The **features** in this case are the `size` and `edge` and the `color` is the **label**. In a **R**, the `data.frame()` function is used to depict the dataset above.

```
squares <- data.frame(size = c("small", "big", "medium"),
                      edge = c("dotted", "stripped", "normal"),
                      color = c("green", "yellow", "green"))
print(xtable(squares, caption = "The dataset as a data.rame"), comment = FALSE)
```

	size	edge	color
1	small	dotted	green
2	big	stripped	yellow
3	medium	normal	green

Table 1: The dataset as a data.rame

The **observations** correspond to the rows and the columns correspond to the **variables**.

So the goal of Machine Learning, based on data shown in the example, is to build a **Model of Prediction**. Build a model that can help make predictions about the data for future instances of similar problems. But before the model can be built, one firstly has to acquaint themselves with the Data. The following Excercises deomstrate this process.

Exercise 1: Getting acquanted with data

As a first step, we will find out some properties of the dataset with which we will be working. More specifically, we want to know mre about the dataset's number of observations and variables. To do this, we will explore the `iris` dataset.

Instructions:

- Use the two ways presented in the video to find out the number of observations and variables of the iris data set: `str()` and `dim()`.
- Call `head()` and `tail()` on `iris` to reveal the first and last observations in the iris dataset.
- Finally, call the `summary()` function to generate a summary of the dataset. What does the printout reveal?

```
# The iris is available from the datasets package and is loaded by default
# Reveal number of observations and variables by looking at the structure
str(iris)
```

Results:

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
# Reveal number of observations and variables by looking at the dimensions
dim(iris)
```

```
[1] 150   5
```

```
# Show first and last observations in the iris data set
head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```
tail(iris)
```

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
```

```
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

```r
# Summarize the iris data set
summary(iris)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

Exercise 2: Basic Prediction Model

To examine a first take a using Machine Learning to make a prediction, we will be using the `Wage` dataset. This dataset contains the wage and some general informaiton for workers in the mid-Atlanic regions of the United States and there could be some relationship between the `age` of a worker and his/her `wage`. Older workers tend to earn more on average than their younger counterparts, hence one could expect an increasing trend in wage as workers age. So we build a linear regression model for you, using the `lm()` function to model the wage of a worker based on his/her age.

With a linear model `lm_wage`, that is built with previous observations, one can predict the wage of new observations. For example, suppose we want to predict the wage of a 60 year old worker. We can use the `predict()` function for this. This generic function takes a model as the first argument. The second argument should be some unseen observations as a data frame. The `predict()` function is then able to predict outcomes for these observations.

Instructions:

- Build a Linear Model called `lm_wage`, that models the `wage` by the `age` variable.
- Create a single colum data frame called `unseen`, with a single column called `age`, containing a single value of `60`.
- Predict the average wage at age 60 using the `predict()` function.

```r
# The Wage dataset is already loaded from the outset.
# Build a Linear Model called lm_wage
lm_wage <- lm(wage ~ age, data = Wage)
```

```r
# Create a data frame for an unseen age (60)
unseen <- data.frame(age = 60)

# Predict the wage of a 60 year old worker
result <- predict(lm_wage, unseen)
```

Results: The Average wage of a 60 year old worker is **124.14 USD** per day.

Classification, Regression and Clustering

In the majority of Machine Learning problems involve **Classification**, **Regression** and **Clustering**.

- A classification problem involves predicting whether a given observation belongs to a certain catagory. What's important to remember regaridng Classification problems, is that the output is **Qualitative** and the possible classes to which a new observation can belong, are known beforehand (Predefined Classes).
- A Regression problem involves predicting a continuous, quantitative value based on previous information. The input value are refered to as **Predictors** and the outout is the **Response**. Regression is somehat simmilar to Classification in that it tries to estimate a funciton that maps the input to the output based on earlier observations, except that the estimate is an actual value. See Appendix A: Regression Coefficients.
-

Appendix A: Regression Coefficients