Chapter 5: Dataset

Part A: Data Set

A dataset is any collection of large data. It can be anything from an array to files, to images to a complete database.

In Machine Learning it is must to work with very large dataset.

Example of a database:

Car Name	Color	Age	Speed	Auto Pass
BMW	red	5	99	Y
Volvo	black	7	86	Y
VW	gray	8	87	N
VW	white	7	88	Y
Ford	white	2	111	Y
VW	white	17	86	Y
Tesla	red	2	103	Y
BMW	black	9	87	Y
Volvo	gray	4	94	N
Ford	white	11	78	N
Toyota	gray	12	77	N
VW	white	9	85	N
Toyota	blue	6	86	Y

If we could predict if a car had an "Auto Pass", just by looking at the other values? That is what Machine Learning is for: **Analyzing data and predicting or classifying the outcome**

1- Data Types

To analyze data, it is important to know what type of data we are dealing with.

We can split the data types into three main categories:

Numerical / Categorical / Ordinal

a) **Numerical** data are numbers, and can be split into two numerical categories:

- *Discrete Data*: counted data that are limited to integers. Example: The number of cars passing by.
- *Continuous Data*: measured data that can be any number. Example: The price of an item, or the size of an item.

b) **Categorical** data are values that cannot be measured up against each other. Example: a color value, or any yes/no values.

c) **Ordinal** data are like categorical data, but can be measured up against each other. Example: school grades where A is better than B and so on.

By knowing the data type of your data source, you will be able to know what technique to use when analyzing them.

2- Mean, Median, and Mode

In Machine Learning there are often three values that interests us:

- **Mean** – The average value
- **Median** – The mid-point value
- **Mode** – The most common value

We have registered the speed of 13 cars:

```
speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]
```

a) **Mean:** (average): the sum of all values divided by the number of values using NumPy method called `mean()`

```
(99+86+87+88+111+86+103+87+94+78+77+85+86) / 13 = 89.77
```

```
import numpy as np
speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]
x = np.mean(speed)
print(x)
```

b) **Median:** value in the middle using a Numpy method called `median()`

```
77, 78, 85, 86, 86, 86, 87, 87, 88, 94, 99, 103, 111
```

It is important that the numbers are sorted before you can find the median.

```
Import numpy as np
speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]
x = np.median(speed)
print(x)
```

If there are two numbers in the middle, divide the sum of those numbers by two.

```
77, 78, 85, 86, 86, 86, 87, 87, 94, 98, 99, 103

(86 + 87) / 2 = 86.5
```

c) **Mode:** value that appears the most number of times

```
99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86 = 86
```

The SciPy module has a method for this: `mode()`

```
from scipy import stats
speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]
x = stats.mode(speed)
print(x)
```

3- Standard Deviation (σ)

Standard deviation is a number that describes how spread out the values are.

- A low standard deviation number means that most of the numbers are close to the mean (average) value.
- A high standard deviation means that the values are spread out over a wider range.

We have registered the speed of 7 cars: `speed = [86, 87, 88, 86, 87, 85, 86]`

The standard deviation is: `0.9` Meaning that most of the values are close to the mean value, which is 86.4.

```
speed = [32, 111, 138, 28, 59, 77, 97]
```

The standard deviation is: `37.85` Meaning that most of the values are away from the mean value, which is 77.4.

The NumPy module has a method to calculate the standard deviation: `std()`

```
import numpy
speed = [86, 87, 88, 86, 87, 85, 86]
x = numpy.std(speed)
print(x)
```

4- **Variance** (σ^2)

Variance is another number that indicates how spread out the values are.

- variance = standard deviation * standard deviation

To calculate the variance you have to do as follows:

a) Find the mean:

```
(32+111+138+28+59+77+97) / 7 = 77.4
```

b) For each value: find the difference from the mean:

```
 32 - 77.4 = -45.4
111 - 77.4 = 33.6
138 - 77.4 = 60.6
 28 - 77.4 = -49.4
 59 - 77.4 = -18.4
 77 - 77.4 = - 0.4
 97 - 77.4 = 19.6
```

c) For each difference: find the square value:

```
(-45.4)² = 2061.16
 (33.6)² = 1128.96
 (60.6)² = 3672.36
(-49.4)² = 2440.36
(-18.4)² = 338.56
(- 0.4)² = 0.16
 (19.6)² = 384.16
```

d) The variance is the average number of these squared differences:

```
(2061.16+1128.96+3672.36+2440.36+338.56+0.16+384.16) / 7 = 1432.2
```

NumPy has a method to calculate the variance: `var()`

```
import numpy
speed = [32, 111, 138, 28, 59, 77, 97]
x = numpy.var(speed)
print(x)
```

Standard deviation is the square root of the variance: √1432.25 = 37.85

5- **Percentiles**

Percentiles are used in statistics to give you a number that describes the value that a given percent of the values are lower than.

Example: the ages of all the people that live in a street.

```
Ages = [5,31,43,48,50,41,7,11,15,39,80,82,32,2,8,6,25,36,27,61,31]
```

What is the 75. Percentile? The answer is 43, meaning that 75% of the people are 43 or younger.

Meaning, what is the age that 75% of the people are younger than? = 43

The NumPy module has a method for finding the percentile: `percentile()`

```
import numpy
ages = [5,31,43,48,50,41,7,11,15,39,80,82,32,2,8,6,25,36,27,61,31]
x = numpy.percentile(ages, 75)
print(x)
```

Part B: Data Distribution

1- Get Big DataSets:

To create big datasets for testing, NumPy comes with methods to create random datasets.

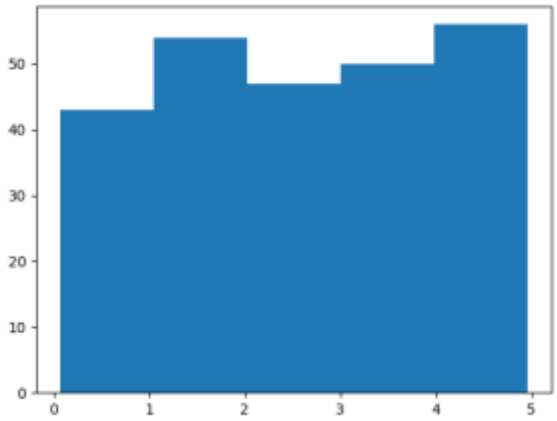
Create an array containing 250 random floats between 0 and 5:

```
import numpy as np
x = np.random.uniform(0.0, 5.0, 250)
print(x)
```

2- Histogram:

To visualize the dataset we can draw a histogram with the 5 bars of array data

```
import sys
import numpy as np
import matplotlib.pyplot as plt
x = np.random.uniform(0, 5, 250)
plt.hist(x, 5) # 5 bars
plt.show()
```



- The first bar represents how many values in the array are between 0 and 1.
- The second bar represents how many values are between 1 and 2. Etc.

Which gives us this result:

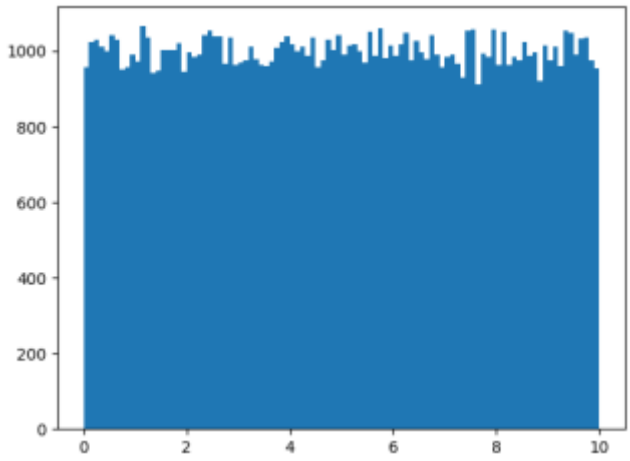
- 42 values are between 0 and 1
- 54 values are between 1 and 2
- 45 values are between 2 and 3
- 49 values are between 3 and 4
- 60 values are between 4 and 5

Note: The array values are random numbers and will not show the exact same result on next run.

3- Big Data Distributions

Create an array with 100,000 random numbers, and display them using a histogram with 100 bars:

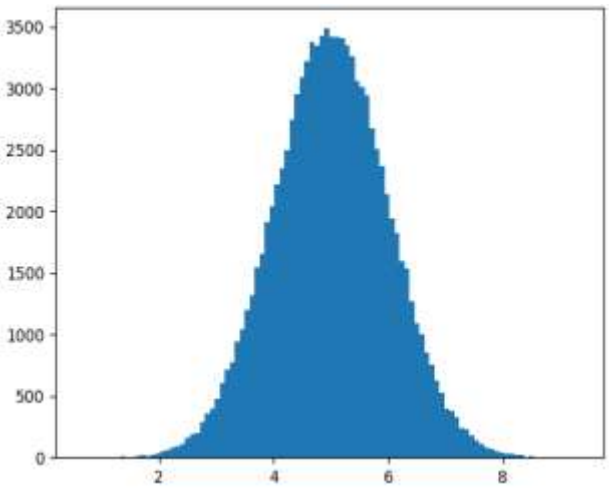
```
import numpy as np
import matplotlib.pyplot as plt
x = np.random.uniform(0, 5, 100000)
plt.hist(x, 100)
plt.show()
```



4- Normal Data Distribution

To create an array where the values are concentrated around a given value known as *normal data distribution* (*Gaussian distribution*).

```
import numpy as np
import matplotlib.pyplot as plt
x = np.random.normal(5, 1, 100000)
plt.hist(x, 100) #100 bars
plt.show()
```



We specify the mean value is 5.0, and the standard deviation is 1.0. Meaning that the values should be concentrated around 5.0, and rarely further away from the mean.

Example: the histogram of the height of 250 people

For simplicity, we use NumPy to randomly generate an array with 250 values, where the values will concentrate around 170, and the standard deviation is 10.

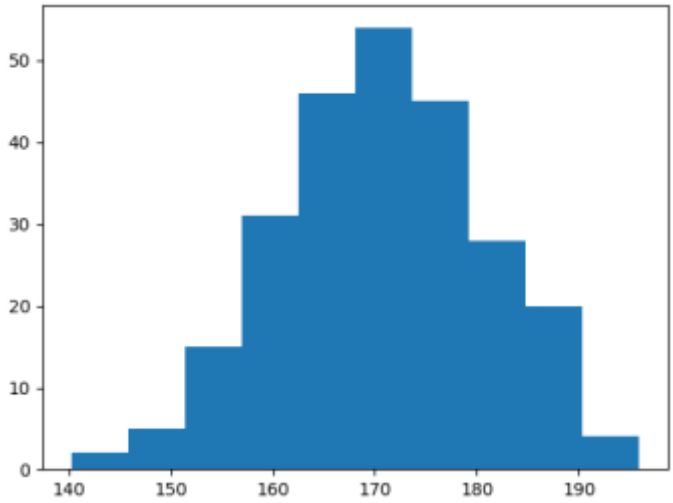
```
import matplotlib.pyplot as plt
import numpy as np

x = np.random.normal(170, 10, 250)

plt.hist(x)
plt.show()
```

You can read from the histogram that there are approximately:

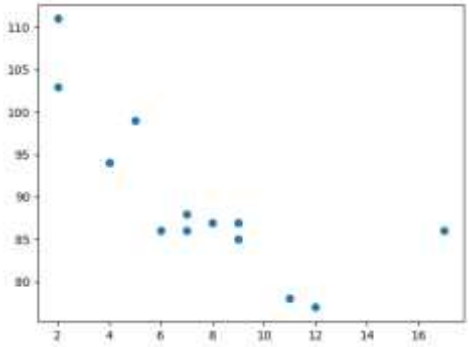
2 people from 140 to 145cm
5 people from 145 to 150cm
15 people from 151 to 156cm
31 people from 157 to 162cm
46 people from 163 to 168cm
53 people from 168 to 173cm
45 people from 173 to 178cm
28 people from 179 to 184cm
21 people from 185 to 190cm
4 people from 190 to 195cm



5- Scatter Plot

A scatter plot is a diagram where each value in the data set is represented by a dot. The x array represents the age of each car. The y array represents the speed of each car.

```
import matplotlib.pyplot as plt
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,
     94,78,77,85,86]
plt.scatter(x, y)
plt.show()
```



What we can read from the diagram is that the two fastest cars were both 2 years old, and the slowest car was 12 years old.

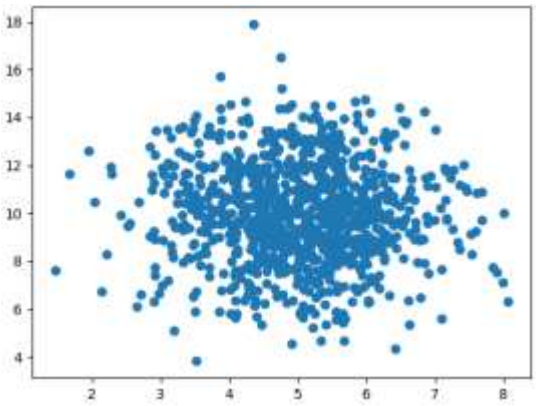
6- Random Data Distributions

In Machine Learning the datasets can contain thousands or even millions of values. You might have real world data when you are testing an algorithm or you might have to use randomly generated values.

```
import numpy
import matplotlib.pyplot as plt

x = numpy.random.normal(5, 1, 1000)
y = numpy.random.normal(10, 2, 1000)

plt.scatter(x, y)
plt.show()
```



The dots are concentrated around the value 5 on the x-axis, and 10 on the y-axis.
We can also see that the spread is wider on the y-axis than on the x-axis.