

## **Lab #3 : kNN**

**Objectives** : learn how to build and train a machine learning classifier in python using a k nearest neighbors (kNN) algorithm

### **Exercise #1**

Train a kNN model that can predict the class for the dataset “data.csv”. Follow these steps:

#### **1) Import libraries**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn import svm
```

#### **2) Load data:**

Load the csv file “data.csv”.

Store the result into a variable called data.

#### **3) Clean the dataset**

Print the size of the dataset

Drop rows containing NULL values. You can use the `dropna` function.

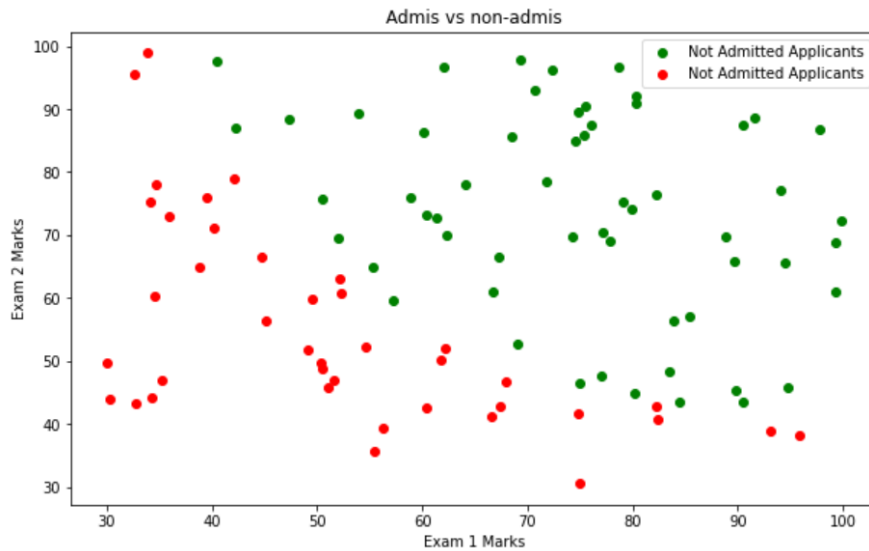
Print the size of the dataset

#### **4) Data visualization :**

Visualize the data using scatter plot (represent in red when `data['Admission status'] == 1` and in blue otherwise).

```
data_admitted = data[data['Admission status'] == 1]
plt.figure(figsize=(10,6))
#TODO: ajouter les élèves admis
plt.scatter(df_notadmitted['Exam 1 marks'], df_notadmitted['Exam 2 marks'],
color= 'red', label= 'Not Admitted Applicants')
#TODO ajouter un label pour les élèves admis
plt.ylabel('Exam 2 Marks')
#TODO ajouter un titre à la figure
plt.title("Admis vs non-admis")
plt.legend()
#TODO afficher la figure
```

This is what you should get



### 5) Data separation and data split:

Separate features from target

Split training set from test set (20% for test)

### 6) Create and train kNN models

Create three kNN models (k=1, k=5, k=9)

train the three models

### 7) Which model is better?

### Exercise #2

Consider the Iris dataset. Given an observation of unknown class, you need to predict its class based on the class of its neighbors. Write a python program that predicts the class of the following observations : [ 0.75, 0.75, 0.75, 0.75] [ 1, 1, 1, 1]