THỰC HÀNH 3 : HỌC GIÁM SÁT – MÔ HÌNH PHÂN LỚP VỚI K-NEAREST NEIGHBOURS

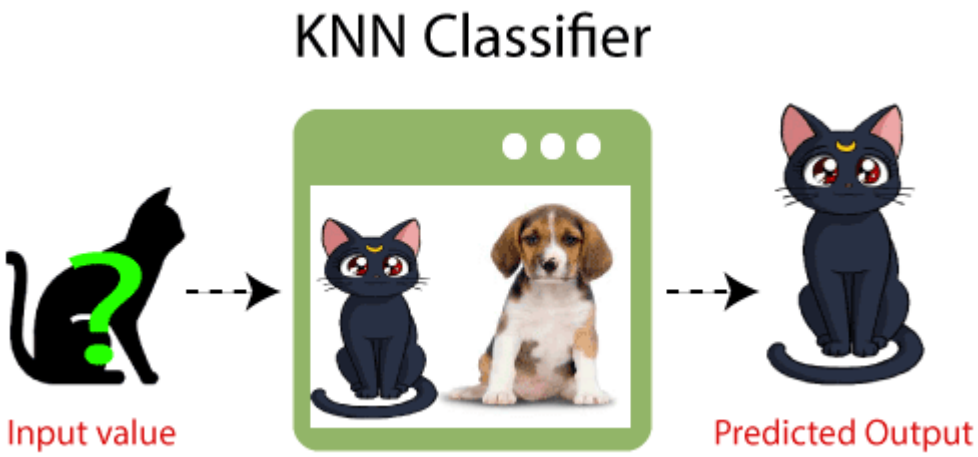
Sau bài thực hành này, sinh viên có khả năng:

- Cài đặt được giải thuật K-Nearest Neighbours với ngôn ngữ Python
- Sử dụng thư viện Pandas để xử lý, tính toán dữ liệu

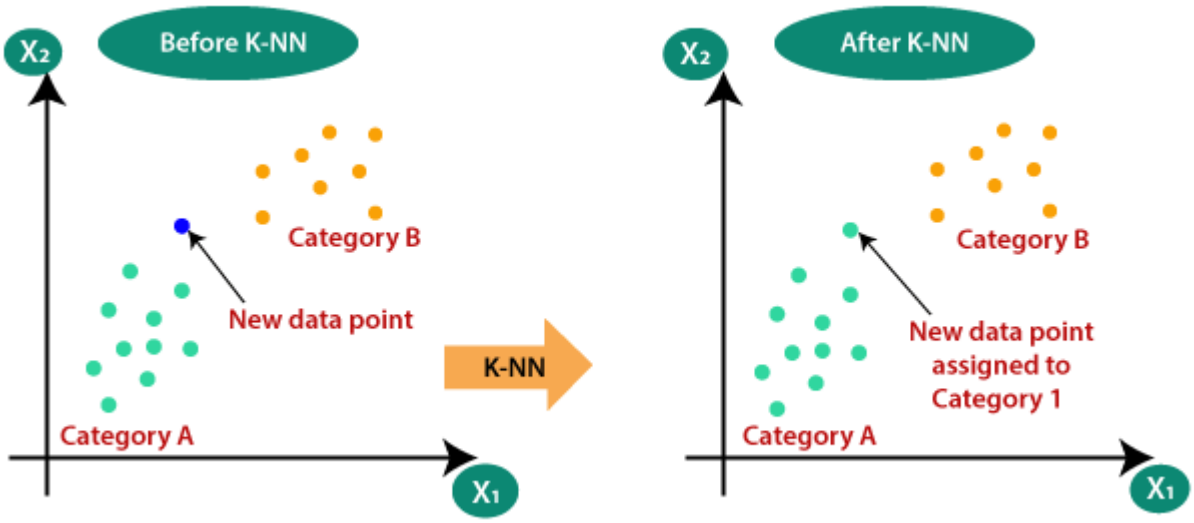
1. GIỚI THIỆU GIẢI THUẬT K-NEAREST NEIGHBOURS

Là giải thuật học giám sát và thường áp dụng cho bài toán nhận dạng mẫu, khai phá dữ liệu và phát hiện xâm nhập.

Giải thuật KNN giả định các đối tượng tương tự thì sẽ ở gần nhau. KNN còn được gọi là lazy learner algorithm vì nó không huấn luyện tập dữ liệu mà chỉ lưu trữ tất cả các dữ liệu trong dataset. Khi có dữ liệu mới, KNN phân lớp vào nhóm tương tự.



Vấn đề đặt ra là làm thế nào phân lớp một đối tượng mới vào đúng nhóm có đặc trưng tương tự với nó

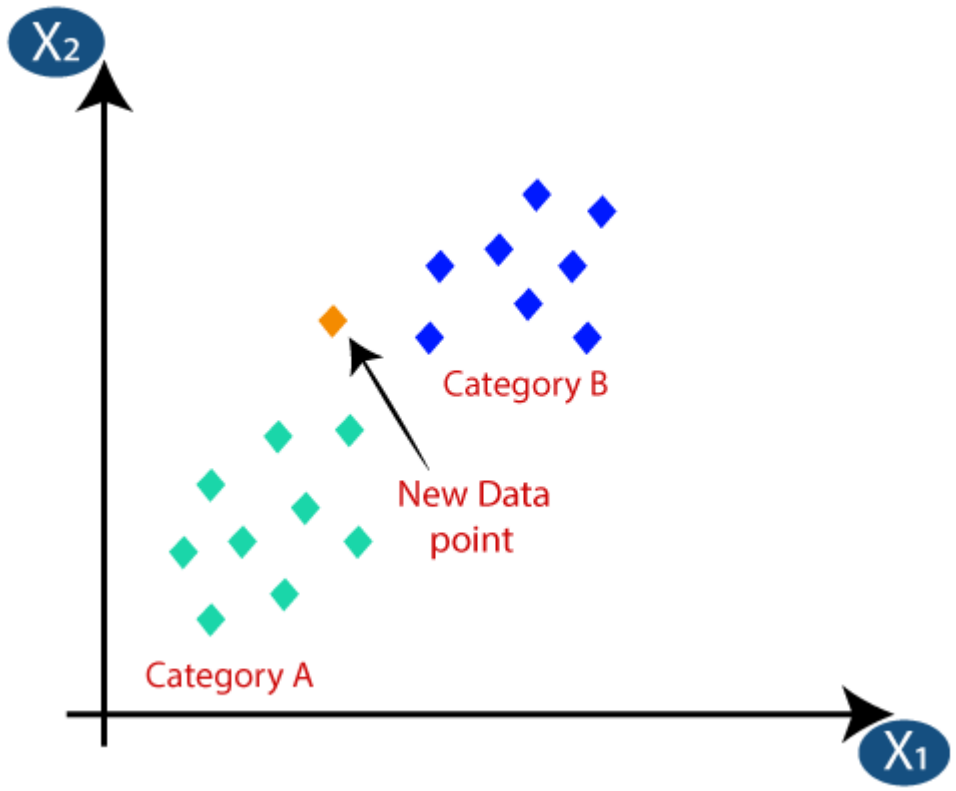


2. CÁC BƯỚC ÁP DỤNG GIẢI THUẬT K-NN

- Chọn số k phần tử kề

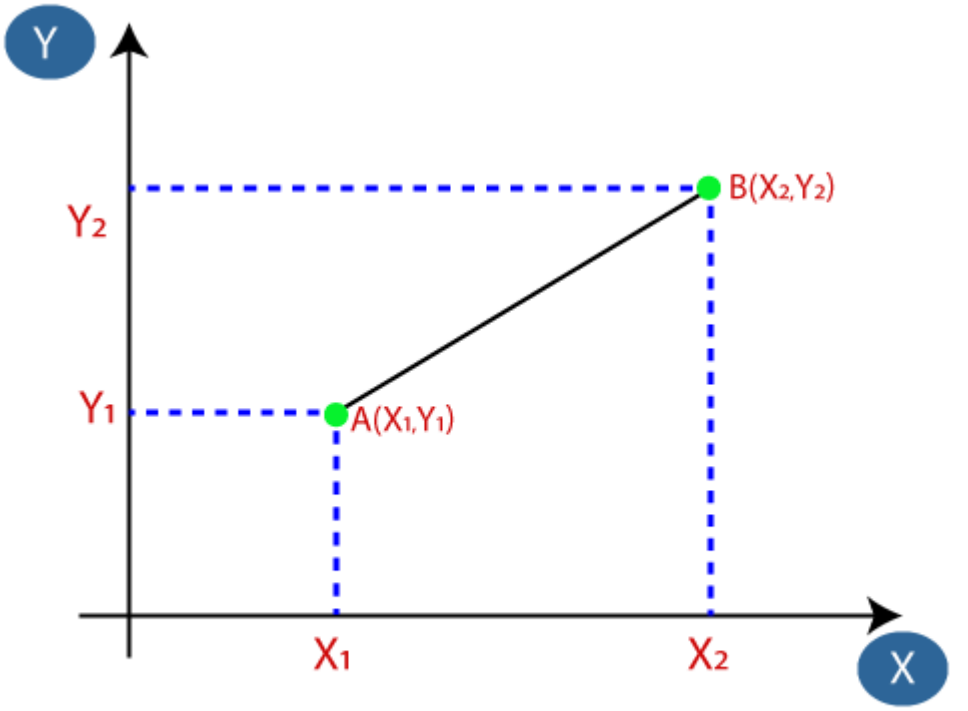
- Tính khoảng cách Euclide của k phần tử kề
- Chọn k phần tử kề gần nhất của mỗi bước tính khoảng cách
- Trong số k phần tử kề, đếm số data của mỗi lớp
- Gán data mới cho lớp có số phần tử kề nhiều nhất.

Ví dụ

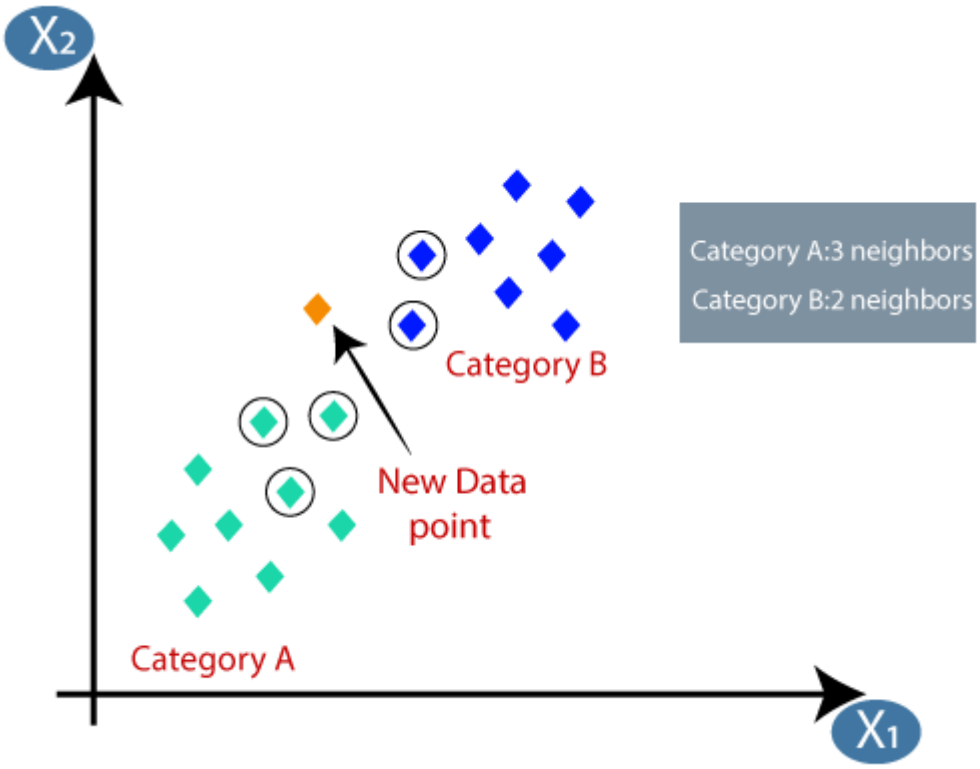


Chọn k = 5 phần tử kề

Dùng công thức Euclide để tính khoảng cách giữa các data points



Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$



3. CÀI ĐẶT GIẢI THUẬT K-NN

Ta sử dụng lại bộ dữ liệu User_Data.csv đã dùng trong LAB 2

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0

3.1. Nạp thư viện

```
[1]  import numpy as nm
     import matplotlib.pyplot as mtp
     import pandas as pd
```

3.2. Nạp dữ liệu

```
#importing datasets
data_set= pd.read_csv('/content/sample_data/User_Data.csv')
print(data_set.head())

#Extracting Independent and dependent Variable
x= data_set.iloc[:, [2,3]].values
y= data_set.iloc[:, 4].values

# Splitting the dataset into training and test set.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.25, random_state=0)
print(x_train.shape)
print(y_train.shape)
```

```
    User ID  Gender  Age  EstimatedSalary  Purchased
0  15624510    Male   19            19000          0
1  15810944    Male   35            20000          0
2  15668575  Female   26            43000          0
3  15603246  Female   27            57000          0
4  15804002    Male   19            76000          0
(300, 2)
(300,)
```

3.3. Feature Scaling

```
[ ]  #feature Scaling
     from sklearn.preprocessing import StandardScaler
     st_x= StandardScaler()
     x_train= st_x.fit_transform(x_train)
     x_test= st_x.transform(x_test)
```

3.4. Huấn luyện phân lớp K-NN

```
[ ]  #Fitting K-NN classifier to the training set
     # n_neighbors: To define the required neighbors of the algorithm. Usually, it takes 5.
     # metric='minkowski': This is the default parameter and it decides the distance between the points.
     # p=2: It is equivalent to the standard Euclidean metric.
     from sklearn.neighbors import KNeighborsClassifier
     classifier= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
     classifier.fit(x_train, y_train)
```

3.5. Dự báo data mới với K-NN

```
[ ]  #Predicting the test set result
     y_pred= classifier.predict(x_test)
```

3.6. Tạo Confusion matrix

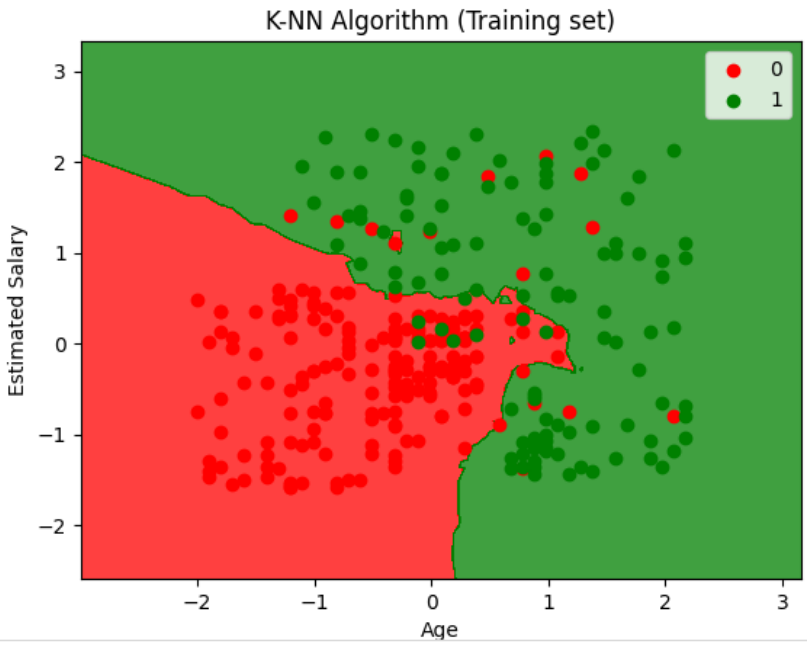
```
#Creating the Confusion matrix
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, y_pred)
cm
```

```
array([[64,  4],
       [ 3, 29]])
```

3.7. Trực quan hóa K-NN trên bộ train

```
from matplotlib.colors import ListedColormap
x_set, y_set = x_train, y_train
x1, x2 = nm.meshgrid(nm.arange(start = x_set[:, 0].min() - 1, stop = x_set[:, 0].max() + 1, step  =0.01),
nm.arange(start = x_set[:, 1].min() - 1, stop = x_set[:, 1].max() + 1, step = 0.01))
mtp.contourf(x1, x2, classifier.predict(nm.array([x1.ravel(), x2.ravel()]).T).reshape(x1.shape),
alpha = 0.75, cmap = ListedColormap(('red','green' )))
mtp.xlim(x1.min(), x1.max())
mtp.ylim(x2.min(), x2.max())
for i, j in enumerate(nm.unique(y_set)):
    mtp.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
        c = ListedColormap(('red', 'green'))(i), label = j)
mtp.title('K-NN Algorithm (Training set)')
mtp.xlabel('Age')
mtp.ylabel('Estimated Salary')
mtp.legend()
mtp.show()
```

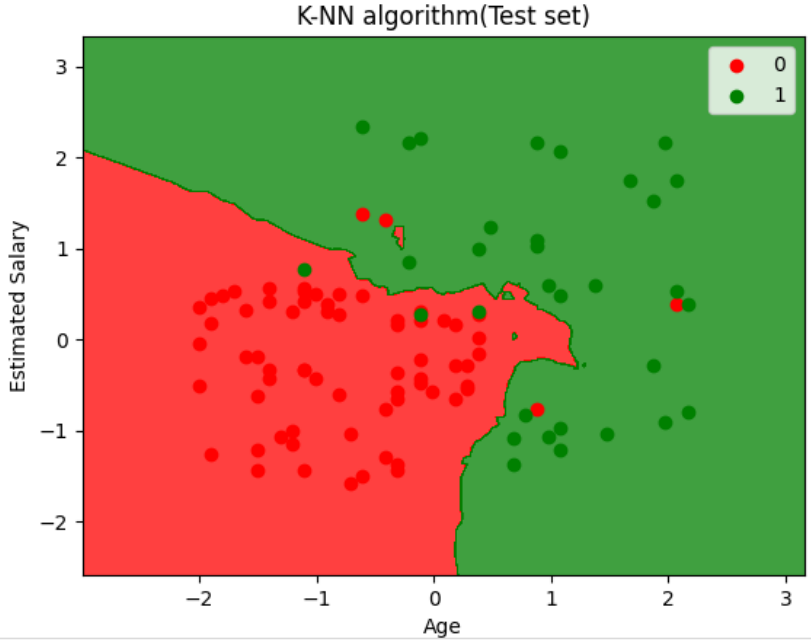
```
<ipython-input-12-f363d1875db7>:11: UserWarning: *c* argument looks like a single numeric RGB or RGBA sequence,
  mtp.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
```



3.8. Trực quan hóa K-NN trên bộ test

```
from matplotlib.colors import ListedColormap
x_set, y_set = x_test, y_test
x1, x2 = nm.meshgrid(nm.arange(start = x_set[:, 0].min() - 1, stop = x_set[:, 0].max() + 1, step  =0.01),
nm.arange(start = x_set[:, 1].min() - 1, stop = x_set[:, 1].max() + 1, step = 0.01))
mtp.contourf(x1, x2, classifier.predict(nm.array([x1.ravel(), x2.ravel()]).T).reshape(x1.shape),
alpha = 0.75, cmap = ListedColormap(('red','green' )))
mtp.xlim(x1.min(), x1.max())
mtp.ylim(x2.min(), x2.max())
for i, j in enumerate(nm.unique(y_set)):
    mtp.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
        c = ListedColormap(('red', 'green'))(i), label = j)
mtp.title('K-NN algorithm(Test set)')
mtp.xlabel('Age')
mtp.ylabel('Estimated Salary')
mtp.legend()
mtp.show()
```

```
<ipython-input-13-1eca06fa12c2>:11: UserWarning: *c* argument looks like a single numeric RGB or RGBA sequence
  mtp.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
```



4. BÀI TẬP K-NN

1. Cho bảng dữ liệu sau

```
Height, Weight, Age, Class
1.70, 65, 20, Programmer
1.90, 85, 33, Builder
1.78, 76, 31, Builder
1.73, 74, 24, Programmer
1.81, 75, 35, Builder
1.73, 70, 75, Scientist
1.80, 71, 63, Scientist
1.75, 69, 25, Programmer
```

a) Tạo file “data.txt” có nội dung như bản trên

b) Viết hàm đọc nội dung từ "data.txt"
c) Cài đặt giải thuật K-NN cho bộ dữ liệu "data.txt"

2. Cài đặt giải thuật phân lớp giống hoa theo biến **"Class"** trong bộ dữ liệu "iris_data.txt" do giảng viên cung cấp.

3. Cài đặt giải thuật phân loại khách hàng sử dụng dịch vụ viễn thông theo biến **"custcate"** trong bộ dữ liệu "teleCust1000t.csv" do giảng viên cung cấp.

4. Cài đặt giải thuật phân loại bệnh nhân bị ung thư thuộc nhóm "maligant" hay "begin" theo biến **"diagnosis"** trong bộ dữ liệu "breast_cancer.csv" do giảng viên cung cấp.

5. Triển khai các câu 1-5 trên nền tảng Web