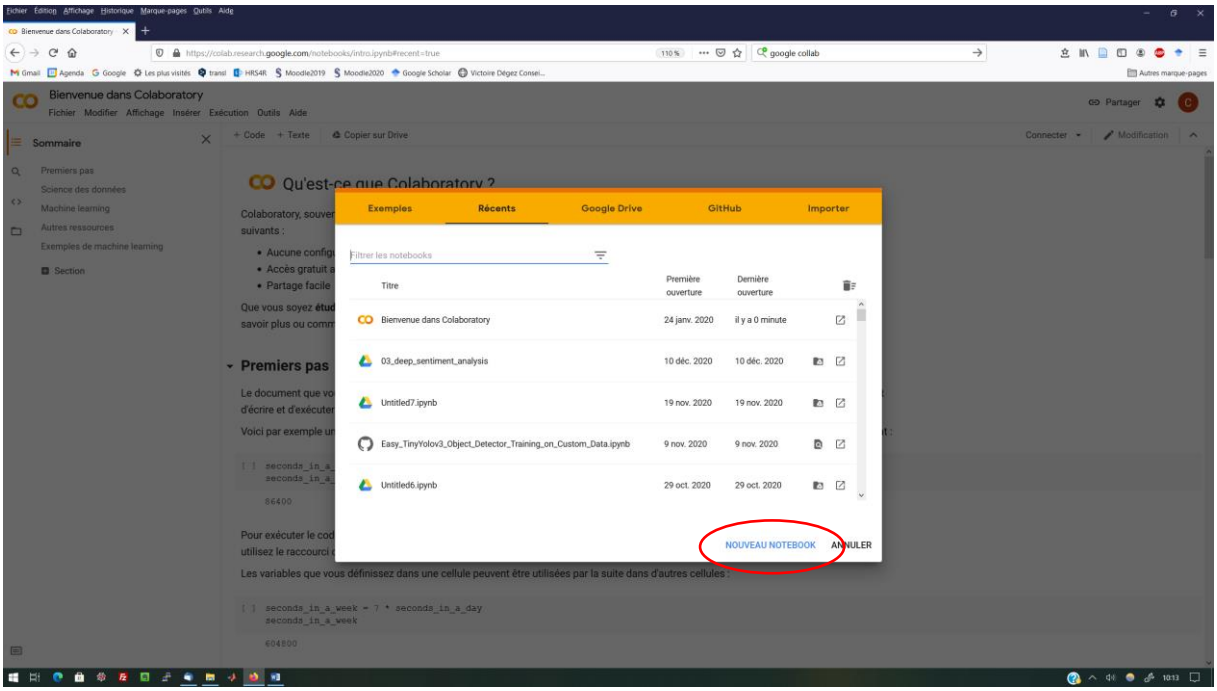
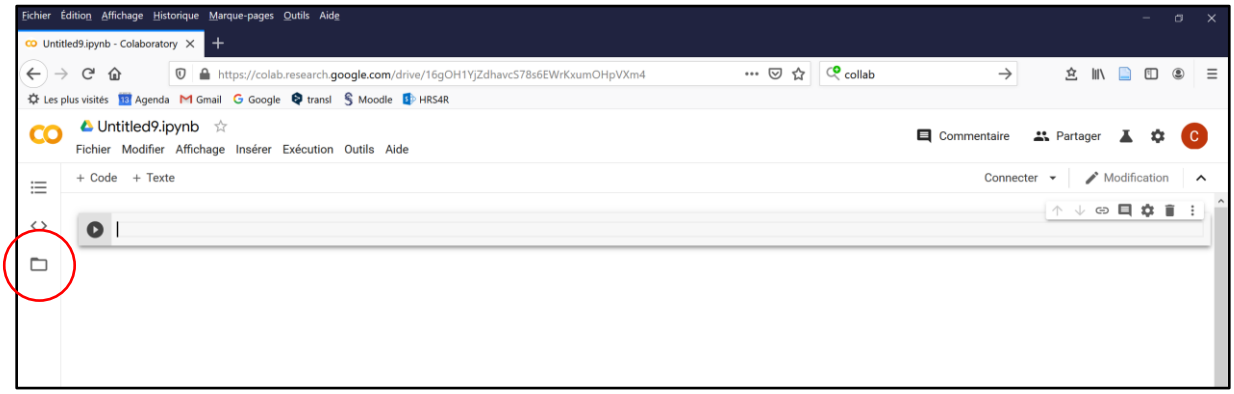
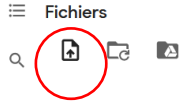
TP1 – Classification par kppv

Nous utiliserons google colab pour travailler. Pour cela, aller dans google colab (https://colab.research.google.com) et ouvrer un nouveau notebook en python 3.



Appuyer sur l'icône répertoire et attendre quelques instants jusqu'à l'apparition du menu importer pour télécharger les données sur lesquelles nous travaillerons (TP1.npy).





Dans ce TP, nous allons utiliser une partie de la base de visages "Labeled Faces in the Wild" provenant de http://vis-www.cs.umass.edu/lfw/. Cette base contient 5749 personnes et 13233 images de taille 62 x 47 pixels. Certaines personnes ne sont représentées qu'une seule fois tandis que d'autres sont représentées très souvent (plus de 80 fois). Nous utiliserons ici seulement 7 personnes représentées 1288 fois.

I. Chargement des données

a. Charger les données

Charger les données, puis afficher les en utilisant les fonctions plotHistoClasses() et plot_gallery() fournies.

```
from display import plotHistoClasses, plotGallery
[X, y, name]=np.load("TP1.npy",allow_pickle=True )
plotGallery(imgs)
plotHistoClasses(lbls)
```

Question

Sachant que X représente les features, y les labels et name le nom des classes, déterminer la taille des images, le nombre d'images et le nombre de classes.

Retrouver l'identité des 12 personnes affichées. Est-ce que les classes sont équiprobables ? Retrouver le nombre d'exemples par classe.

b. Partitionnement de la base

Partitionner la base en une base d'apprentissage, une base de validation et une base de test en mettant 15% des données en test et 15% en validation (fonction train_test_split()) pour obtenir les variables X_train, X_val, X_test, y_train, y_val et y_test.

Question

Combien y a-t-il d'images en apprentissage, validation et en test ? Quelles sont les dimensions des variables X_train, X_val, X_test, y_train, y_val et y_test ?

II. Prétraitement des données

a. Redimensionnement des données

Pour réaliser une classification par kppv, on utilise un codage rétinien. Chaque image est donc représentée par un vecteur de caractéristique de dimension $n = 2914$. Redimensionner X_train, X_val et X_test de façon à ce qu'ils aient pour dimension N x n (np.reshape()) où N est le nombre d'exemples.

b. Mise en forme des données pour la classification

Mettre en forme les données (train val et test) en utilisant la classe StandardScaler. On estimera la moyenne et l'écart-type de chaque dimension sur les données d'apprentissage, puis on transformera les données (train val et test) en utilisant ces valeurs. Aller sur la documentation en ligne de StandardScaler pour voir quelle méthode de cette classe utiliser.

Question

A quoi consiste la mise en forme des données ? Comment sont-elles transformées ?

III. Classification par les KPPV

a. Classifieur 1PPV

Définir le classifieur 1PPV en utilisant la classe KNeighborsClassifier(). On souhaite utiliser la distance euclidienne et le 1PPV.

Réaliser la classification des exemples de validation en utilisant la méthode predict().
Afficher la matrice de confusion (fonction confusion_matrix()) et estimer le taux de reconnaissance à partir des éléments de cette matrice. Vérifier que le taux est identique à celui renvoyé par la fonction accuracy_score(). Afficher également le rapport de classification (fonction classification_report())

Questions
Que représente la matrice de confusion ? Que vaut sa somme ? Est-ce que les classes sont équilibrées ?

Que représente le rapport de classification ? Retrouver chacun de ses éléments à partir de la matrice de confusion.

b. Classifieur KPPV

Faire varier le K des KPPV et tracer l'évolution du taux de reconnaissance en fonction de K.

Questions
Conclusion ? Interpréter l'évolution des résultats en fonction de K

c. Classifieur KPPV et distance de Manhattan

Réaliser les mêmes tests avec la distance de Manhattan.

Questions
Conclusion ? Interpréter l'évolution des résultats en fonction de K.

d. Performance du classifieur

En utilisant les résultats précédents, mettez au point le classifieur final et évaluez ses performances sur la base de test.