Basics in ML with Python

For th two folling exercises, one may rely on the notebook `Introduction to the Pandas library`

The RMS Titanic was a British passenger liner that sank in the North Atlantic Ocean in the early morning hours of 15 April 1912, after it collided with an iceberg during its maiden voyage from Southampton to New York City. There were an estimated 2,224 passengers and crew aboard the ship, and more than 1,500 died, making it one of the deadliest commercial peacetime maritime disasters in modern history.

Women and children first? The aim is to understand how survivors of Titanic were selected...

The Titanic dataset may be downloaded on
`https://sites.google.com/site/marianneclausel/home/enseignements-20-21/cirm?authuser=0`

Exercise 1 : Importation of the data and description of the dataset

1. In this first practical session, we shall work on the dataset `titanic.csv` on the survival of the passengers of Titanic. Download this dataset as a data frame
2. Describe the dataset `titanic` : features, nature of the features, number of observations
3. Basic statistics : mean of each variable, quartiles
4. Percentage of missing values for each column. Sort by descending values

Exercise 2 : Basic graphic analysis

We want to understand what features could contribute to a high survival rate. It would make sense if everything except 'PassengerId', 'Ticket' and 'Name' would be correlated with a high survival rate.

1. Get rid off the features 'PassengerId', 'Ticket' and 'Name' which seem irrelevant to analyse the data
2. We focus on the features 'Age' and 'Sex'.
   (a) Separate the dataset into men and women
   (b) Display the distribution of the age survivors and non survivors according to the sex. Comment
3. At first glance is there some link between 'Embarked' and 'Survival'.
4. At first glance is there some link between 'Pclass' and 'Survival'.

For the two next sections, one may rely on the notebook `Introduction to Principal Component Analysis`

Exercise 3 : Analysis of Financial Time series

We want to understand the information contained in financial time series. The dataset is `rs.csv` and can be downloaded on Arche.

1. Import the data into a dataframe `rs`. this dataframe may contain missing values. Imputation of missing values can be done using the function `fillna` : `https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html`
2. To process the data in order to perform PCA, convert this dataframe into a numpy array using `to_numpy()`
3. Perform a PCA with two components. Transform the data projecting on the two first principal components
4. Plot the two first components
5. Visualize the data in the first two-component space
6. Compare with t-SNE

Exercise 4 : the Olivetti faces dataset

We want to perform PCA on a classical dataset : the Olivetti faces dataset.

1. Import the dataset using the function `fetch_olivetti_faces`
2. Center the faces : `faces_centered = faces - faces.mean(axis=0)`
3. We now plot the faces

```
import numpy as np
import matplotlib.pyplot as plt
n_row, n_col = 5, 7
n_components = n_row * n_col
image_shape = (64, 64)

def plot_gallery(title, images):
    plt.figure(figsize=(2. * n_col, 2.26 * n_row))
    plt.suptitle(title, size=16)
    for i, comp in enumerate(images):
        plt.subplot(n_row, n_col, i + 1)
        comp = comp.reshape(image_shape)
        vmax = comp.max()
        vmin = comp.min()
        plt.imshow(comp, cmap=plt.cm.gray, vmax=vmax, vmin=vmin)
        plt.xticks(())
        plt.yticks(())

    plt.subplots_adjust(0.01, 0.05, 0.99, 0.93, 0.04, 0.)


# Plot a sample of the input data
plot_gallery("First centered Olivetti faces", faces_centered[:n_components])
```

4. Perform PCA with 20 components. Plot the 20 first components

For the two next exercises, one may rely on the notebook `Introduction to Classical ML models`

Exercise 5 : Classification on the Breast Cancer dataset

The Breast Cancer Wisconsin (Diagnostic) Data Set is a classical dataset widely used in classification. The task consists in predicting if a tumor is malignant (M) or benign (B) depending on certain features, that can be downloaded from the UCI Machine Learning repository. More details here
`https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`
We want to compare the performance of Logistic Regression, SVM and Random Forest on this classification task. The confusion matrix is the matrix

$$M = \begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}$$

where TP : True Positive, TN : True Negative, FP : False Positive, FN : False Negative. We recall the notion of precision and recall

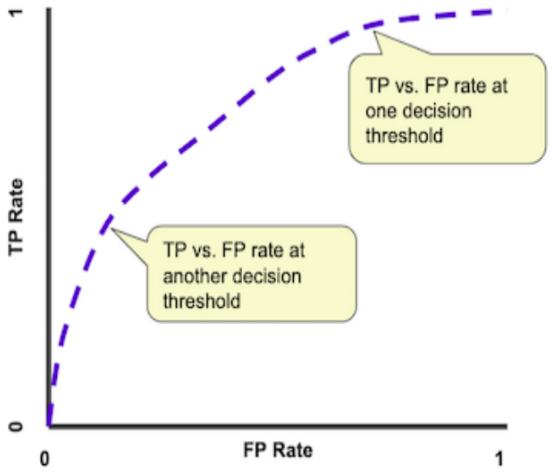
$$\text{Precision} = \frac{TP}{TP+FP}, \text{ Recall} = \frac{TP}{TP+FN}$$

The accuracy is then defined as

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Accuracy is a very useful metric when all the classes are equally important. But this might not be the case if we are predicting if a patient has cancer. In this example, we can probably tolerate FPs but not FNs. This metric could also be not so relevant when considering imbalanced data.

ROC curve : A ROC curve (receiver operating characteristic curve) graph shows the performance of a classification model at all classification thresholds.



AUC : AUC stands for Area under the ROC Curve. It provides an aggregate measure of performance across all possible classification thresholds.

The higher the area under the ROC curve (AUC), the better the classifier. A perfect classifier would have an AUC of 1. Usually, if your model behaves well, you obtain a good classifier by selecting the value of the threshold that gives TPR close to 1 while keeping FPR near 0.

1. Import the dataset `breast_cancer.csv`. Drop the `unnamed` and `id columns`.
2. In each case, for each classification method (Logistic Regression, SVM and Random Forest)
   (a) Give the confusion matrix
   (b) Evaluate the accuracy of the model
   (c) Plot the ROC curve

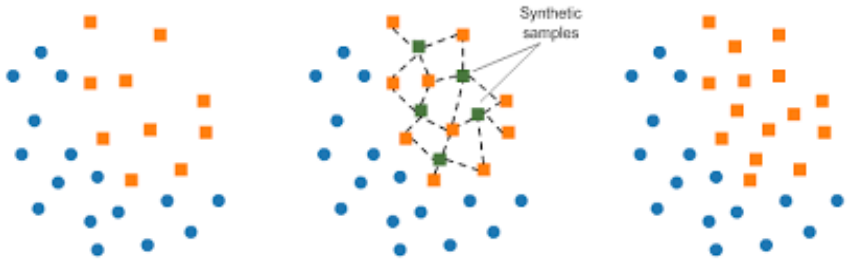
Exercise 6 : Deal with imbalanced dataset

We now consider a dataset aout fraudulent credit card transaction
`https://www.kaggle.com/mlg-ulb/creditcardfraud`

1. Import the dataset `creditcardfraud.csv`. What is the distribution of the target variable? Explain teh specificity of this dataset
2. A first strategy consists in oversampling the minority class using the so-called SMOTE method which is implemented in Python in the library `imblearn`
   `https://imbalanced-learn.org/stable/generated/imblearn.over_sampling.SMOTE.html`
   The idea is to synthesize new examples from the minority class. SMOTE first selects a minority class instance a at random and finds its k nearest minority class neighbors.

   The synthetic instance is then created by choosing one of the k nearest neighbors b at random and connecting a and b to form a line segment in the feature space.

   The synthetic instances are generated as a convex combination of the two chosen instances a and b.



   After oversampling, for each classification method (Logistic Regression, SVM and Random Forest)

(a) Give the confusion matrix

(b) Evaluate the accuracy of the model. Do you think this metric is relevant for imbalanced classification? An alternative consists in defining Sensitivity-Specificity Metrics One defines

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity is the complement to sensitivity, or the true negative rate, and summarises how well the negative class was predicted.

$$Specificity = \frac{TN}{FP + TN}$$

For imbalanced classification, the sensitivity might be more interesting than the specificity. One then defines the G–mean as

$$G - Mean = \sqrt{(Sensitivity \times Specificity)}$$

(c) We now define the following metric

$$F_\beta = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{(\beta^2 \cdot Precision + Recall)}$$

When will be the F_1 (resp $F_{0.5}$, F_2 mesure relevant)? What happens in our case

3. We now consider a second strategy consisting in considering another loss function. The idea is that the costs caused by different kinds of errors are not assumed to be equal and the objective is to minimize the expected costs instead of the usual loss. For e.g. it means that we replace the usual logistic loss

$$-\sum_i y_i \log \widehat{y}_i - \sum_i (1 - y_i) \log(1 - \widehat{y}_i$$

with

$$\frac{1}{N} \sum_i y_i \left(\widehat{y}_i \, C_{TP_i} + (1 - \widehat{y}_i) \, C_{FN_i}\right) + \frac{1}{N} \sum_i (1 - y_i) \left(\widehat{y}_i \, C_{FP_i} + (1 - \widehat{y}_i) \, C_{TN_i}\right)$$

where C_{TP}, C_{FN}, C_{FP} and C_{TN} are the cost assoicated to TP, FN, FP and TN.

Test the cost sensitive loss for logistic regression and random forest using the package `costcla` available at `https://pypi.org/project/costcla/`