Florent Rossignol
Pol-antoine Loiseau
Constantin Lenglos

MLII_Unsupervised_Learning_and_Agents

Exo 1 :

Objective:
The objective of this exercise is to demonstrate the convergence of the empirical average of random samples
drawn from two normal distributions to the expected values of those distributions.
We generate random samples representing heights (X) and weights (Y) and observe how the empirical average approaches
the expected values over increasing sample sizes.

Code Overview:

Random Data Generation:

Two normal distributions are generated using np.random.normal to simulate heights (X) and weights (Y).
Parameters for X: mean (mu_x) = 170, standard deviation (sigma_x) = 10.
Parameters for Y: mean (mu_y) = 70, standard deviation (sigma_y) = 15.
Sample size (n) is set to 1000.
Seed for reproducibility is set to 42.
Scatter Plot:

A scatter plot is created to visualize the random samples in a 2D space, where X represents height and Y represents weight.

Convergence Analysis:
Empirical averages are computed for increasing sample sizes using np.cumsum and then dividing by the number of samples.
Expected values are defined based on the distribution parameters.
Euclidean distances between the empirical averages and expected values are calculated.

Convergence Plot:
A plot is created to illustrate how the Euclidean distance between the empirical
average and the expected value changes with an increasing number of samples.

Conclusion:
The scatter plot visually represents random samples of heights and weights, while the convergence plot demonstrates
how the empirical average converges to the expected values as the sample size increases.
This exercise provides a practical illustration of the law of large numbers,
showing how sample averages become more accurate estimators of population means with larger sample sizes.

Exo 2 :

Objective:
The goal of this exercise is to compare the effectiveness of two popular dimensionality reduction techniques, Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), on a given dataset. The dataset is loaded, standardized, and then reduced to 2D and 3D using both PCA and t-SNE. The visualizations are plotted for a qualitative assessment of how well the reduced dimensions separate the data points based on their labels.

Code Overview:

Data Loading and Preprocessing:

Data and labels are loaded from 'data.npy' and 'labels.npy', respectively.
The data is standardized using StandardScaler from scikit-learn.

PCA Dimensionality Reduction:

PCA is applied to reduce the data to 2D and 3D.

Visualization:
Subplots are created to display the results side by side.
Scatter plots are used for both 2D and 3D representations, with different colors indicating different labels.

Conclusion:

The visualizations provide insights into how well PCA and t-SNE capture the underlying structure of the data.
PCA tends to preserve global structure, whereas t-SNE is effective in capturing local relationships.
The choice between PCA and t-SNE depends on the specific characteristics of the data and the goals of the analysis.

Exo 3:

Rapport d'Analyse des Métriques et des Graphiques - Clustering

Objectif de l'Analyse :

L'objectif de cette analyse est d'évaluer la performance relative de deux méthodes de clustering, à savoir le K-Means et le Hierarchical Clustering, sur un ensemble de données clients.
Nous avons utilisé différentes métriques et visualisations pour parvenir à des conclusions significatives.

Résultats :

K-Means :

Silhouette Score : 0.49
Un score élevé indique une bonne séparation des clusters.

Davies-Bouldin Index : 0.86
Une valeur relativement basse indique des clusters compacts et bien séparés.

Hierarchical Clustering :

Calinski-Harabasz Index : 1103.44
Un index élevé suggère des clusters bien définis et séparés.

Adjusted Rand Index : 0.22

Un indice ajusté proche de 0.22 indique une correspondance modérée avec les vraies étiquettes.

Analyse des Métriques :

K-Means :

Performances solides avec un Silhouette Score élevé et un Davies-Bouldin Index relativement bas.
Les clusters semblent bien séparés et compacts.

Hierarchical Clustering :
Un bon Calinski-Harabasz Index indiquant des clusters bien définis.

L'Adjusted Rand Index montre une correspondance modérée avec les vraies étiquettes.

Visualisation des Résultats :

Les graphiques PCA montrent la répartition des clusters dans l'espace réduit pour les deux méthodes.
Ils indiquent visuellement la séparation des clusters et peuvent aider à évaluer la performance.

Stabilité des Clusters (K-Means) :

En évaluant la stabilité des clusters avec différentes initialisations pour le K-Means, nous avons observé une stabilité rapide avec un Silhouette Score constant autour de 0.49-0.50.
Cela suggère que le K-Means converge rapidement vers une solution stable, indépendamment de l'initialisation.
La stabilité rapide est une caractéristique positive qui renforce la confiance dans les clusters identifiés.

Conclusion :

Le K-Means semble avoir des performances solides avec une stabilité rapide et des métriques indiquant une bonne séparation des clusters.
Le Hierarchical Clustering montre également de bonnes performances, mais l'Adjusted Rand Index indique une correspondance modérée avec les vraies étiquettes.

Exo 4 :

To carry out Ex-4:

First we need to understand how the simulation is created and how the agent knows the positions of the last rewards it obtains.

In order to increase the rewards that the agent obtains in the default policy I first documented the different algorithms that I can use
The multi-armed bandit algorithms and in particular the E-greedy algorithm allows to choose the best known action, but with a small probability, it chooses a random action too.

In my policy:

I first added a variable that I initialized at 0.25 this value represents the exploration rate (it's the probability that the agent explores new positions)
I then added an if which performs a random action like going left or right if the agent explores (the agent has a 0.25 chance of exploring).
Then I added an else which if the agent doesn't explore it will choose according to the positions of the rewards obtained previously a move. If the agent knows that a previous reward was on the left it will move to the left for example.

Thanks to this I obtained an average reward around 20. However I was looking for a way to improve my average rewards even more.

To do this

I added an if that will increase the exploration rate if the agent has no previous rewards position by multiplying the exploration rate by 1.5.

So if the agent hasn't found any rewards, he'll decide to explore more. This allows him to maximize the average number of rewards, which now exceeds 20.

Exo 5 :

Objective:
This analysis aims to explore and gain insights from the Olympic dataset. Various steps are performed,
including data cleaning, basic statistical analysis, visualization of medal distribution, and K-means clustering on selected features.

1. Data Loading and Overview:

The Olympic dataset is loaded from the "dataset_olympics.csv" file.
Basic information about the dataset is displayed using info().
The first few rows of the dataset are printed to provide an initial look at the data.

2. General Analysis:

Descriptive statistics and the count of missing values are calculated to understand the overall characteristics of the dataset.

3. Distribution of Medals by Country:

The number of medals is aggregated by country, and the top 10 countries with the highest medal counts are visualized using a bar plot.

4. Preprocessing:

Missing values are imputed using the most frequent strategy.
The 'Sex' column is label-encoded for further analysis.

5. Visualization of Medals Over the Years:

The distribution of medals over the years is visualized for the top 5 teams with the highest cumulative medal counts.

6. K-means Clustering:

Relevant features ('Age' and 'Year') are selected for clustering.
Numeric data is standardized using StandardScaler.
K-means clustering is applied with a specified number of clusters (in this case, 2), and clusters are visualized.

Conclusion:

The analysis provides a comprehensive exploration of the Olympic dataset, including descriptive statistics,
visualization of medal distribution, and K-means clustering.
The distribution of medals over the years for top teams offers insights into historical performance trends.
K-means clustering on age and year features visually separates data points into clusters,
potentially revealing underlying patterns or trends in the dataset.
Adjusting the number of clusters may yield different insights.