UQAC

Travail pratique 2

8PRO408

N°	Nom des Étudiants
01.	AUDREY LAVOIE (LAVA24539302)
02.	AIDA SALL (SALA31550002)
03.	GABRIEL SENCE (SENG17120405)
04.	HICHAM LAKHAL (LAKH30109204)
05.	ANDRE ALANO (ALAA17010009)
06.	JEANNOT MIMBO KINGOLO (MIMJ11099709)
07.	ANDRIAMITIA TOLOTRINIAINA HARIMAHEFA (ANDT11059907)

Travail remis à
Julien Maitre

Le 29 novembre 2022

Sommaire

1. Description de l'ensemble de données

1.1. Lien du Dataset :

https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+#

1.2. Objectif de la collecte des données :

Données expérimentales utilisées pour la classification binaire (occupation des pièces) à partir de la température, de l'humidité, de la lumière et du CO2.

1.3. Comment ont été collectées les données :

Les données ont été obtenues à partir de photos horodatées prises toutes les minutes.

1.4. Nombres d'Instances :

Dans notre Dataset il y a 20560 d'instances.

1.5. Nombres d'Attributs :

Notre jeu des données contient au moins 7 attributs.

1.6. Noms des variables

Les différents noms des variables dans l'ensembles des données qui sont :

- date time year-month-day hour:minute:second
- Temperature, in Celsius
- Relative Humidity, %
- Light, in Lux
- CO2, in ppm
- Humidity Ratio, Derived quantity from temperature and relative humidity, in kgwater-vapor/kg-air
- Occupancy, 0 or 1, 0 for not occupied, 1 for occupied status

1.7. Données manquantes :

Il n'y a aucune donnée manquante(NaN) sur l'ensemble des Données (Dataset).

2. Étapes pour la création de l’ensemble de données

L’algorithme de création est basé sur cela fait dans les sessions de live coding, avec quelques modifications pertinentes.

L'algorithme fait l'extraction des données des plusieurs fichiers en utilisant une fenêtre temporelle.

L'algorithme prend les valeurs minimum, maximum, la moyenne, l’écart-type, le skew et le kurtosis à chaque itération.

Les modifications les plus importantes sont le changement des donnés ver type float et la lecture des fichiers en question.

La seule colonne qui n’est pas présente dans le nouveau fichier est la date, à raison que le format n’est pas compatible avec les autres données.

3. Explication de l’algorithme de réduction de dimensionnalité

La réduction de dimensionnalité en « machine learning » est utilisée principalement pour combattre le surapprentissage *(overfitting).* Celui-ci est défini par le manque de fiabilité dans la prédiction des futures données dû à une classification trop précise de celles-ci. On souhaite donc généraliser pour améliorer le modèle. Il existe plusieurs algorithmes afin de faire la réduction de dimensionnalité, on s’intéresse ici à l'algorithme « ExtraTreesClassifier » de la librairie scikit-learn de Python. Cet algorithme crée des arbres de décisions aléatoires sur plusieurs sous-échantillons du dataset et fait une moyenne pour améliorer la précision de la prédictibilité.

La première étape est d’appeler le constructeur de classe ExtraTreesClassifier avec les paramètres choisis. On peut par exemple définir le nombre d’arbres en spécifiant le paramètre « n_estimators ». Il existe plusieurs autres paramètres qui peuvent être définis ou non (criterion, max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, max_features et plusieurs autres).

La méthode « fit() » est ensuite appelée pour y insérer nos données. Cette méthode construit la forêt avec notre jeu de données. La méthode prend comme paramètre X (les échantillons des données d’apprentissage), y (les valeurs cibles) et sample_weight qui, par défaut, sera équilibré.

On place ensuite dans une variable la propriété « feature_importances » qui donne un tableau avec des valeurs entre 0 et 1. Plus le nombre est près de 1, plus la caractéristique est importante.

La méthode argsort() de Numpy va ensuite faire le tri de ce tableau afin de placer les plus importantes caractéristiques en premier.« x.columns » va faire la sélection des colonnes les plus importantes par rapport à notre tableau trier et du nombre d’attributs que l’on avait préalablement choisi. Le nouveau dataset sera ce produit final.

4. La Présentation des Résultats obtenus dans la Réduction du nombre de colonnes (caractéristiques) de votre ensemble de données

Dans ce point nous allons baser sur les résultats obtenus lors de la création d'une nouvelle Dataset et la comparaison entre les deux résultats de la réduction pour les deux ensembles de données.

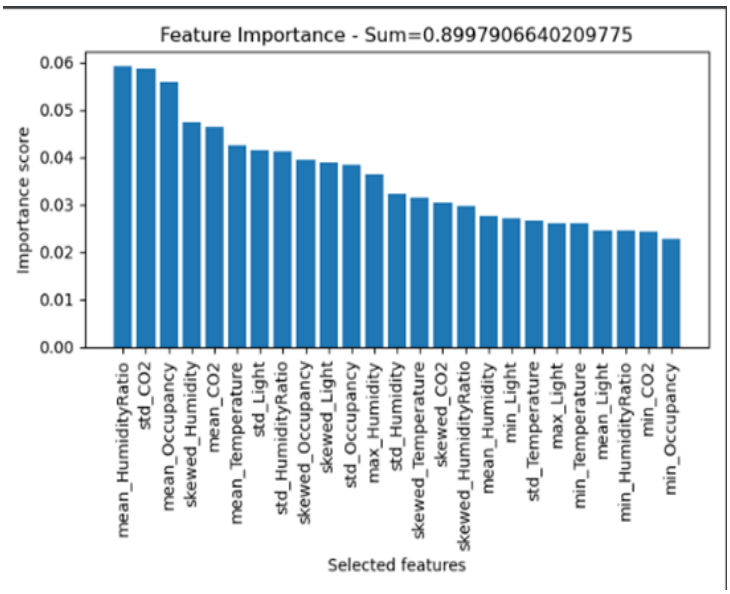
a. La taille du Dataset avant la réduction de dimensionnalité

```
..            ...                    ...
349     -1.434224              -0.618663
350     -1.420711              -1.160360
351     -1.422142              -0.655194
352     -0.603889              -0.242489
353     -0.627128              -0.250206

[354 rows x 37 columns]
```

b. La taille du Dataset après la réduction de dimensionnalité

```
352  610.983333           0.004302  937
353  614.283333           0.004302  937

[354 rows x 25 columns]

Process finished with exit code 0
```

c. Le Premier résultat du Dataset on définir la largeur de fenêtre temporelle glissante est de 5.



- **La liste des colonnes avant la réduction de dimensionnalité du Dataset avec (37 colonnes) :**

index min_Temperature min_Humidity min_Light min_CO2 min_HumidityRatio min_Occupancy max_Temperature max_Humidity max_Light max_CO2 max_HumidityRatio max_Occupancy mean_Temperature mean_Humidity mean_Light mean_CO2 mean_HumidityRatio skewed_Light skewed_CO2 skewed_HumidityRatio skewed_Occupancy mean_Occupancy skewed_Temperature skewed_Humidity std_Temperature std_Humidity std_Light std_CO2 std_HumidityRatio std_Occupancy kurtosis_Temperature kurtosis_Humidity kurtosis_Light kurtosis_CO2 kurtosis_HumidityRatio kurtosis_Occupancy.

- **La liste des colonnes après réduction de dimensionnalité du Dataset avec (25 colonnes) :**

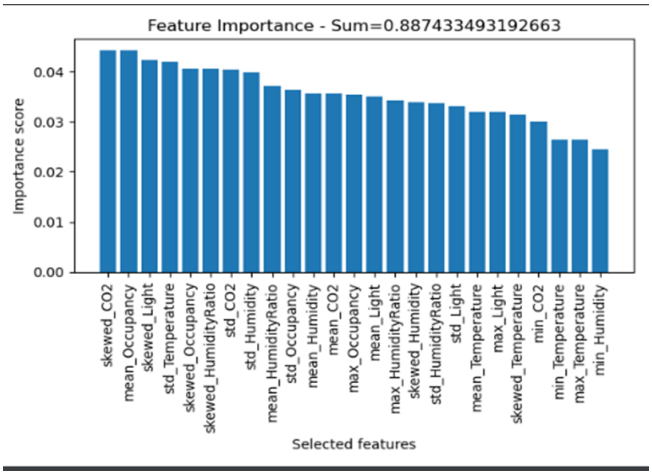
skewed_CO2 mean_Occupancy skewed_Light std_Temperature skewed_Occupancy skewed_HumidityRatio std_CO2 std_Humidity mean_HumidityRatio std_Occupancy mean_Humidity mean_CO2 max_Occupancy mean_Light max_HumidityRatio skewed_Humidity std_HumidityRatio skewed_Temperature min_CO2 min_Temperature max_Temperaturestd_Light mean_Temperature max_Light min_Humidity.

```
Time to extract: 13.696996212005615

Total size: 15900X36

Number of labels: 6
```

```
[1380 rows x 25 columns]
```

d. Deuxième résultat Dataset avec une nouvelle largeur de fenetre temporelle glissante

Dans le deuxième ensemble de données on définir la nouvelle largeur de fenêtre temporelle glissante est de 15 voici le résultat obtenu après la glissante d'une fenêtre temporelle dans cette nouvelle Dataset.



5. Conclusion (résumé des informations essentielles)

Dans cet travailler pratique a pour but collecter des données « signaux » et de formaliser un traitement à partir d'un algorithme. Ces données collectées sont utilisées pour la classification binaires à partir de certaines caractéristiques (voir tp).

L'algorithme utilisé fait l'extraction des données de plusieurs fichiers. Les modifications les plus importantes dans cet algorithme sont le changement des donnés ver type float et la lecture des fichiers en question.

Pour la réduction de dimensionnalité, nous avons choisi d'utiliser l'algorithme « ExtraTreesClassifier » de la librairie scikit-learn de Python qui crée des arbres de décisions aléatoires sur plusieurs sous-échantillons du dataset et fait une moyenne pour améliorer la précision de la prédictibilité.

Nous avons expliqué le fonctionnement ainsi que les étapes de l'algorithme de réduction de dimensionnalité qui sont d'appeler le constructeur de classe ExtraTreesClassifier avec les paramètres choisis, appeler la méthode « fit() » pour y insérer nos données, appeler la méthode argsort() de Numpy pour faire le tri de ce tableau afin de placer les plus importantes caractéristiques en premier.

6. Conclusion générale (ce que nous avons apprécié, appris, moins apprécié par rapport aux données etc.)

Dans le cadre de ce deuxième travail pratique, nous avons décidé d'analyser des données expérimentales utilisées pour la classification binaire (occupation des pièces) à partir de la température, de l'humidité, de la lumière et du CO2.

Le dataset ne présentait pas de donnée manquante (NaN).

Puisqu'il s'agissait d'un dataset de taille considérable, il nous a fallu beaucoup de temps pour le comprendre.

Grâce à cet travail pratique, nous avons appris comment collecter et analyser les données exploitées et les étapes d'extraction des caractéristiques que nous avons appliqué pour la création de l'ensemble de données.

En général, le travail a été bien cadré cela dû à la répartition des tâches et la cohérence des membres de groupes.