SORBONNE UNIVERSITÉ

Projet Python

ISUP – Parcours ISDS

Septembre-Novembre 2024

Présentation générale

Le projet vise à appliquer les outils vus pendant le cours.

Plusieurs jeux de données seront fournis pour l'application des méthodes implémentées et la visualisation des résultats.

Le projet est individuel.

Date de rendu

Le projet doit être déposé sur Moodle au plus tard le **jeudi 19 décembre 2024 à 23:00**. Aucun projet ne sera accepté passé cette date.

Contraintes imposées

Les outils devront être implémentés de bout en bout, c'est-à-dire sans avoir recours à des librairies existantes, à l'exception de `NumPy, Pandas, Matplotlib` et `Seaborn`. Les librairies `sklearn` et `statsmodels` sont proscrites.

L'implémentation de tests unitaires sera apprécié. Utiliser la librairie `pytest`.

La qualité du code devra être évaluée avec `pylint` avec si possible une note au moins supérieure à 7/10.

L'utilisation des outils vus dans le cours sera très appréciée. L'organisation du code devra être réfléchie en amont selon les recommandations évoquées dans le cours (packaging, qualité de code, etc.).

Un exemple de construction de projet est donné à ce lien, chaque partie devra être construite sur ce modèle.

Rendu

Pour chaque partie, le code source devra être compilé dans une archive `tar.gz` qui puisse être installé avec `pip`.

Il contiendra notamment un fichier `README.md` (langage Markdown) comprenant la documentation du package et l'explication de ses fonctionnalités.

Enfin, le package sera accompagné d'un programme principal `main.py` pour l'exécution du code.

Partie 1 : analyse statistique et modèle linéaire

L'objectif est de créer un package `linearmodel` pour l'analyse d'un jeu de données et la construction d'un modèle linéaire.

1.1 Jeu de données

L'objectif est d'analyser les émissions de CO2 de différents véhicules en fonction de certaines de leurs caractéristiques. Les données proviennent du gouvernement du Canada : elle enregistrent les émissions de CO2 pour chaque véhicule ainsi que les détails de ces véhicules.

Les données sont enregistrées au format CSV sont décrites dans le document joint :

- Marque et type de modèle
- Détails sur les caractéristiques techniques du véhicule
- Type de carburant
- Consommation au 100km
- Estimation des emission de CO2 en g/km

1.2 Analyse statistique et visualisation

Implémenter des fonctions dans un module `statistics.py` et `visualization.py` pour analyser le jeu de données. En particulier :

- statistiques descriptives pour analyser et comparer les différents véhicules
- analyse de corrélation entre les différentes variables
- visualization des données
- etc.

1.3 Moindres carrés ordinaires

L'objectif est de construire un modèle linéaire pour expliquer le **taux d'emission de CO2** en fonction de certaines covariables du jeu de données.

Pour cela, implémenter la méthode des moindres carrés ordinaires pour le modèle linéaire multiple :

$$y = X\beta + \varepsilon,$$

où y est un vecteur de dimension n, X est une matrice de dimension $n \times d$, β est un vecteur de paramètres inconnus de dimention d et ε est le vecteur des erreurs de dimension n.

L'estimateur des moindres carrés est donné par

$$\hat{\beta} = (X^T X)^{-1} X^T y.$$

Pour cela, il faudra implémenter une classe `OrdinaryLeastSquares` dont les **méthodes** sont :

- `fit` : prend des données X et y en entrée et calcule l'estimateur des moindres carrés $\hat{\beta}$
- `predict` : prend des données de test X_t et renvoie les prédictions associées

- `get_coeffs` : retourne les valeurs des coefficients estimés
- `determination_coefficient` : calcule et renvoie le coefficient de détermination R^2

Le contructeur devra prendre en entrée un argument booléen `intercept` qui visera à ajouter ou non une covariable constante au modèle.

D'autres méthodes, fonctions ou attributs pourront être implémentés si besoin. Par exemple :

- Visualisation des résultats
- Vérification des hypothèses du modèle
- Intervalles de confiance
- etc.

Utilisez ensuite cette classe que vous avez implémenté afin de construire un modèle linéaire ajusté sur les données (ou une partie des donnés) du dataset.

Partie 2 : calculateur d'empreinte carbone

L'objectif est d'implémenter un outil permettant à un restaurateur d'évaluer l'empreinte carbone de son activité dans un package `carboncalc`.

2.1 Jeu de données

Le jeu de données `Base_Carbone_V23.4.csv` est issu de la base carbone de l'ADEME[1]. Il contient les quantités d'emission de CO_2 pour une grande quantité de postes de consommation (transport, énergie, alimentation, etc.). Ces quantités sont exprimées en équivalent CO_2[2].

Ce jeu de données est à l'état brut, la première étape sera de l'ouvrir et de le transformer afin de pouvoir l'utiliser comme base de données pour les différents postes d'émissions de CO_2.

Comme cette première étape peut être bloquante, 3 extractions du jeu de données ont été faites `aliments.csv`, `energie.csv` et `equipements.csv`. Des points seront accordés à ceux qui feront eux mêmes l'extraction des données, `mais surtout ne restez pas bloqués sur cette partie !`

2.2 Objectifs

Implémentation du calculateur

L'objectif est de calculer précisément l'empreinte carbone annuelle, mensuelle ou hebdomadaire en tenant compte des différents postes de consommation et des données fournies.

Le programme principal demande à l'utilisateur les quantités des différents postes de consommation (aliments utilisés, matériel acheté, électricité, etc.) et calcule son empreinte carbone totale en tonnes.

Le programme récolte les réponses et fait un bilan des émissions annuelles :

- affichage du total annuel et du détail par poste comme donné dans l'exemple `calculator_example.py` ;
- visualisation des résultats, voir par exemple l'étude publiée par le cabinet Carbone4[3].
- etc.

[1] `https://base-empreinte.ademe.fr/`
[2] `https://en.wikipedia.org/wiki/Global_warming_potential#Carbon_dioxide_equivalent`
[3] `https://www.carbone4.com/myco2-empreinte-moyenne-evolution-methodo`

La manière de demander les informations à l'utilisateur devra se faire par une **méthode passant par python**. Par exemple il n'est pas permis d'écrire un programme qui prendrait comme entrée un fichier `.csv` qui recenserait les postes de consommation de l'utilisateur.

Dans tous les cas, la manière de faire fonctionner le programme devra être explicitement décrite dans le fichier `README.md` du dossier.

Remarque : Les objectifs indiqués ci-dessus sont des minimums à accomplir, toute idée supplémentaire est la bienvenue ! Sentez-vous libre de vous emparer du sujet et d'implémenter les différentes fonctionalités que vous jugerez pertinentes pour un restaurateur souhaitant controler l'empreinte carbone de son activité.