Travaux pratiques sur Jupyter Notebook, Machine Learning

Tp1/ Partie1

Régression linéaire multiple

L'objectif de ce TP est d'appliquer un modèle de **régression linéaire multiple** afin de prédire le **profit d'une entreprise** à partir de plusieurs variables explicatives : les dépenses en **recherche et développement (R&D)**, les dépenses en **administration**, le budget alloué au **marketing**, ainsi que **la ville** dans lequel l'entreprise exerce ses activités.

Ce travail pratique vise non seulement à construire et entraîner un modèle prédictif, mais également à analyser l'influence relative de chaque facteur sur la variable cible (le profit), et à évaluer la performance du modèle sur des données réelles.

1. **Chargement et exploration des données**

   - Importer les bibliothèques nécessaires : numpy, pandas, matplotlib, sklearn.

   - Charger le fichier CSV.

   - Afficher les premières lignes et vérifier les types de données.

   - Visualiser les corrélations entre les variables.

2. **Prétraitement des données**

   - Gérer les données manquantes (s'il y en a).

   - Encoder la variable catégorielle State avec OneHotEncoder.

   - Séparer les variables explicatives (X) et la variable cible (y).

   - Diviser le dataset en train (80%) et test (20%) avec train_test_split.

3. **Ajustement du modèle de régression linéaire multiple**

   - Importer et entraîner le modèle LinearRegression de sklearn.linear_model.

   - Ajuster le modèle sur l'ensemble d'apprentissage.

4. **Prédiction des résultats**

   - Prédire les profits de l'ensemble de test.

   - Comparer les prédictions aux valeurs réelles.

5. **Évaluation du modèle**

   - Calculer la précision avec le **R^2 score**.

   - Évaluer l'erreur moyenne absolue (MAE) et l'erreur quadratique moyenne (RMSE).

Ensemble de données

R&D Spend	Administration	Marketing Spend	State	Profit
165349.2	136897.8	471784.1	New York	192261.83
162597.7	151377.59	443898.53	California	191792.06
153441.51	101145.55	407934.54	Florida	191050.39
144372.41	118671.85	383199.62	New York	182901.99
142107.34	91391.77	366168.42	Florida	166187.94
131876.9	99814.71	362861.36	New York	156991.12

Régression linéaire simple

L'objectif de ce TP est de modéliser la relation entre la superficie d'une maison (Area) et son prix (Price) à l'aide d'une régression linéaire simple.

Ce modèle permettra de prédire le prix d'un bien immobilier en fonction de sa surface.

On dispose d'un fichier CSV nommé homeprices.csv contenant deux colonnes :

Area : surface du logement (en m^2)

Price : prix du logement (en dollars)

1. Importer les bibliothèques nécessaires : pandas, numpy, matplotlib, et sklearn.linear_model.
2. Charger les données à partir du fichier CSV dans un DataFrame pandas.
3. Visualiser la relation entre Area et Price à l'aide d'un nuage de points (scatter plot).
4. Appliquer la régression linéaire simple en utilisant LinearRegression() de sklearn.
5. Afficher la droite de régression sur le graphique.
6. Afficher les paramètres du modèle :

   - L'ordonnée à l'origine (a)
   - Le coefficient directeur (b)
   - L'équation de la droite : Price= a+b×Area

8. Évaluer la performance du modèle avec le coefficient de détermination R^2.
9. Prédire le prix d'une maison pour une surface donnée (exemple : 230 m^2).

	Area	Price
0	2600	550000
1	3000	565000
2	3200	610000
3	3600	680000
4	4000	725000

Régression logistique

L'objectif de ce TP est de mettre en œuvre un modèle de régression logistique afin de prédire si un individu achète (ou non) un produit, en fonction de son âge et de son salaire estimé.

Ce TP permettra :

1. Comprendre la différence entre régression linéaire et régression logistique.

2. Appliquer la régression logistique à un problème de classification binaire.
3. Visualiser les résultats du modèle et interpréter les probabilités.
4. Évaluer la performance du modèle à l'aide de métriques adaptées.

Le tableau contient les variables suivantes :

- User ID : identifiant de l'utilisateur (non pertinent pour la prédiction).
- Gender : sexe de l'utilisateur (Male/Female).
- Age : âge de l'utilisateur.
- EstimatedSalary : salaire estimé.
- Purchased : variable cible (0 = n'a pas acheté, 1 = a acheté).

Ensemble de données

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0

1. Chargement et exploration des données

- Importer les bibliothèques nécessaires (pandas, numpy, matplotlib, seaborn, sklearn).
- Charger le fichier CSV.
- Vérifier la structure des données (head(), info(), describe()).
- Visualiser la distribution de la variable cible (Purchased).

2. Prétraitement des données

- Supprimer la colonne User ID (non informative).
- Encoder la variable Gender (Male/Female → 0/1).
- Définir les variables explicatives (X) : Age, EstimatedSalary, Gender.
- Définir la variable cible (y) : Purchased.
- Diviser les données en train (80%) et test (20%) avec train_test_split.

3. Construction et entraînement du modèle

- Importer la classe LogisticRegression de sklearn.linear_model.
- Entraîner le modèle sur l'ensemble d'apprentissage.

4. Évaluation du modèle

- Construire une matrice de confusion.
- Calculer l'accuracy, la précision, le rappel et le F1-score.
- Tracer la courbe ROC et calculer l'AUC.