Option : Ml OIVM
Module : TP ATDN

Objectif :

Ce TP a pour objectif de :

- Premièrement (Partie 1) : Vous permettre de vous familiariser avec la régression et de mettre en exergue les différences entre régression linéaire et polynomiale (avec l'utilisation de la régularisation Ridge) en modélisant les erreurs avec la distribution gaussienne sur un ensemble de données fictifs bidimensionnelles.
- Deuxièmement (Partie 2) : Appliquer des techniques avancées de régression, d'effectuer la validation croisée et d'analyse statistique pour évaluer et comparer la performance des modèles sur un jeu de donnés spécifique : House Price.

Partie 1 : Régression avec Distribution Gaussienne sur des données générées

Données : dans cette partie les données utilisées seront générées de manière aléatoire :

- Avec 100 échantillons pour la variable indépendante X définis par $X=2\times np.random.rand(100,1)$
- Et la variable dépendante y est calculée selon l'équation $y=7+4X+$bruit gaussien

---

I- Régression Linéaire

1- Génération de données et visualisez les données
2- Appliquez une régression linéaire en utilisant la distribution gaussienne pour modéliser les erreurs et affichez les prédictions du modèle ajusté sur le graphique avec les données.
3- Calculez les résidus (différence entre les valeurs observées et prédites).
4- Visualisez la distribution des résidus. Utilisez un histogramme et un graphique de probabilité normale (Q-Q plot) pour évaluer les résidus par rapport à la distribution gaussienne. Interprétez les résultats obtenus.

II- Régression Polynomiale avec Distribution Gaussienne

5- En utilisant les mêmes données, appliquez une régression polynomiale d'un degré supérieur. Affichez les prédictions du modèle polynomiale ajusté sur le graphique avec les données.
6- Calculez les résidus. Visualisez la distribution des résidus. Utilisez un histogramme et un graphique de probabilité normale (Q-Q plot) pour évaluer les résidus par rapport à la distribution gaussienne. Interprétez les résultats obtenus.
7- Comparez les performances des deux modèles (linéaire et polynomial). Utilisez des métriques telles que le coefficient de détermination (R^2) et l'erreur quadratique moyenne (RMSE).
8- Interprétez les coefficients des modèles. Que signifient-ils dans le contexte du problème étudié ? Résumez vos observations et conclusions. Quel modèle semble mieux représenter les données ?

Partie II : Régression linéaire et polynomiale sur le jeu de données publiques ‘House Price’

Données : Utilisez un jeu de données public tel que celui sur les **prix des maisons** (dataset sur Kaggle) : https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data

---

I- Régression Linéaire et Polynomiale avec Visualisation et Analyse des Résidus

1. **Ajustement du Modèle Linéaire :**
   - Importez les données et visualisez la relation entre les variables explicatives et la variable cible.
   - Implémentez une régression linéaire et tracez les prédictions.
   - Calculez et visualisez les résidus avec un histogramme et un Q-Q plot.
2. **Ajustement du Modèle Polynomial :**
   - Implémentez une régression polynomiale (degré 2 ou 3) et visualisez les résultats.
   - Comparez la performance des modèles linéaire et polynomial à l'aide des métriques R^2 et RMSE.
3. **Régression Polynomiale avec Régularisation Ridge :**
   - Ajoutez une régularisation Ridge à la régression polynomiale.
   - Comparez les coefficients et les performances des modèles avec et sans régularisation.

II- Validation Croisée et Intervalle de Confiance

4. **Validation Croisée :**
   - Implémentez une validation croisée (k-fold) pour la régression linéaire et polynomiale.
   - Comparez les performances moyennes entre les modèles sur chaque pli et tracez la dispersion des scores.
5. **Calcul des Intervalles de Confiance :**
   - Calculez les intervalles de confiance des prédictions pour le modèle polynomial avec régularisation Ridge.
   - Expliquez l'interprétation de ces intervalles et leur importance dans le contexte de la régression.

III- Probabilités Jointes et Théorème de Bayes

6. **Exercice de Codage sur les Probabilités Jointes :**
    - Implémentez un programme qui calcule la probabilité conjointe de deux événements dans un dataset (par exemple, la probabilité qu'une maison ait plus de 3 chambres et soit située dans une région donnée).
    - Visualisez la probabilité conjointe sous forme de tableau croisé dynamique.
7. **Théorème de Bayes :**
    - Écrivez un programme Python qui calcule la probabilité a posteriori en utilisant le théorème de Bayes pour un problème simple (ex. : probabilité qu'une voiture soit économique sachant qu'elle a une faible consommation).
    - Expliquez l'importance du théorème de Bayes dans l'inférence statistique.

IV- Comparaison des Différentes Régressions

8. **Exercice de Comparaison :**
    - Comparez la régression linéaire, polynomiale simple, et polynomiale avec Ridge sur le même ensemble de données.
    - Analysez et discutez des différences en termes de complexité, de capacité à surmonter le surapprentissage et de performance.
    - Visualisez la courbe d'apprentissage de chaque modèle.

Partie III : Régression Logistique

Objectif :

Appliquer une régression logistique pour prédire si le prix d'une maison est au-dessus ou en dessous de la médiane des prix.

Données :

On va supposer qu'on dispose d'un dataset de prix de maisons, comprenant les colonnes suivantes :
Nous allons cettefois générer des données simulées pour notre exercice sur la régression logistique.
Soit n=100, le nombre d'observations (ajouter np.random.seed(42) pour rendre déterministe, semi aléatoire).

- **Surface** : la superficie de la maison en mètres carrés,
  - surface = np.random.randint(50, 200, n) # Surface en m2
- **Nb_pieces** : le nombre de pièces,
  - nb_pieces = np.random.randint(1, 6, n) # Nombre de pièces
- **Age** : l'âge de la maison,
  - age = np.random.randint(0, 50, n) # Âge de la maison en années
- **Distance_centre** : la distance de la maison au centre-ville (en km),
  - distance_centre = np.random.uniform(0, 20, n) # Distance au centre en km
- **Prix** : le prix de la maison (cible). Prix simulé avec une tendance (prix en milliers d'€)
  - prix = 30 + (surface * 0.8) + (nb_pieces * 15) - (age * 0.5) - (distance_centre * 2) + np.random.normal(0, 10, n)

I- Préparer les données

1- Charger les données.

(Respectez les consignes énoncées dans la partie données en amont)

2- Créer la variable cible binaire

On convertit les prix en 0 (moins chers) ou 1 (plus chers) en fonction de la médiane.

3- Diviser les données en ensemble d'entraînement et ensemble de test.

4- Standardiser les données

Standardisez les données pour améliorer la performance de la régression logistique.

II- Entraîner le modèle de régression logistique

5. Utiliser la régression logistique pour entraîner un modèle et évaluer sa performance.

III- Évaluation

6. Évaluer le modèle en utilisant la précision, le rappel, et la matrice de confusion.
7. Prédire la catégorie (plus chère ou moins chère) d'une nouvelle maison.
8. Créer une matrice de confusion et des distributions pour visualiser les performances du modèle.
9. Analyser la précision, le rappel, et la capacité du modèle à classer correctement les maisons dans chaque catégorie.

Instructions Techniques :

- Utilisez Python et installer les bibliothèques NumPy, Pandas, Matplotlib, scikit-learn, et Seaborn (vous pouvez configurer un environnement Conda avec ces bibliothèques).
- Lister toutes les bibliothèques que vous installez dans un fichier texte requirements.txt à fournir avec les codes (sur GitHub)
- Commentez le code pour expliquer chaque.
- Fournissez des graphiques soignés pour les visualisations et des interprétations claires des résultats. Présentez vos résultats de manière claire à l'aide de visualisations et de métriques.