420sn1_projet

November 6, 2024

Projet : Analyse de données scientifiques

0.0.1 Contexte

Les maladies cardiovasculaires (MCV) sont responsables de millions de décès chaque année. Dans ce projet, vous avez été mandatés par une équipe de recherche pour utiliser vos compétences en Python et les bibliothèques pandas, matplotlib et numpy afin de créer un programme capable de prédire les risques de maladies cardiaques en analysant certaines caractéristiques clés, comme l'âge, le sexe, et le taux de cholestérol.

0.0.2 Objectif

L'objectif de ce projet est de démontrer comment les techniques de programmation et d'analyse de données peuvent être appliquées pour résoudre des problèmes de santé publique.

Ce projet mettra en lumière l'importance de l'interdisciplinarité entre les sciences de la nature et l'informatique, en montrant comment les compétences en programmation peuvent être utilisées pour des applications pratiques et bénéfiques dans le domaine de la santé.

0.0.3 Parties du projet

1. **Partie 1** : Importer et préparer les données, extraire des informations clés pour effectuer des analyses pertinentes. (Pandas)
2. **Partie 2** : Visualiser les données sous forme de graphiques, pour en faciliter l'interprétation. (Matplotlib)
3. **Partie 3** : Analyser les résultats et en tirer des conclusions. (NumPy)

0.0.4 Présentation des données

Les données de l'expérience sont dans le fichier prediction_cardiaque.csv. Voici la description des données s'y trouvant. Ce fichier contient les données de patients à travers le monde.

- **Âge du patient** : en années
- **Sexe du patient** : M : Masculin, F : Féminin
- **Type de douleur thoracique**: TA : Angine Typique, ATA : Angine Atypique, NAP : Douleur Non Angineuse, ASY : Asymptomatique
- **Pression artérielle au repos** : pression artérielle au repos (en mm Hg)
- **Cholestérol** : cholestérol sérique (en mg/dl)
- **Glycémie à jeun** : 1 : si Glycémie à jeun $>$ 120 mg/dl, 0 : sinon

- **Résultats de l'ECG au repos** : Normal : Normal, ST : anomalie de l'onde ST-T (inversions de l'onde T et/ou élévation ou dépression du segment ST de > 0,05 mV), LVH : hypertrophie ventriculaire gauche probable ou certaine selon les critères d'Estes
- **Fréquence cardiaque maximale atteinte**: valeur numérique entre 60 et 202
- **Angine induite par l'exercice** : Y : Oui, N : Non
- **Oldpeak** : ST valeur numérique mesurée en dépression
- **Pente du segment ST au pic de l'exercice** : Up : ascendant, Flat : plat, Down : descendant
- **Maladie cardiaque** : classe de sortie 1 : maladie cardiaque, 0 : Normal

0.0.5 Livrables

- Vous devrez remettre un seul fichier Jupyter Notebook (**PrenomNom_projet.ipynb**) contenant tout le code, les analyses et les visualisations et le fichier de données (**.csv**).
- **À la fin de chaque séance**, vous devez remettre votre travail (fichiers .ipynb et .csv) dans la boîte de remise prévue à cet effet sur Moodle.
- Vous pourrez continuer à travailler sur votre projet entre chaque séance, mais la boite de remise sera fermée.
- Ces remises sont des points de contrôle.

0.0.6 Consignes et informations de départ pour le projet

1. De Moodle, télécharger sur votre ordinateur, dans le dossier réservé au projet, les deux fichiers :
   - Le fichier de données : insuffisance_cardiaque.csv
   - Le fichier de code : 420sn1_projet.ipynb
2. Renommer le fichier .ipynb avec votre prénom et nom de famille (ex: **NathalieDesmangle_projet.ipynb**)
3. Dans le fichier de départ (ipynb) que vous allez utiliser pour ce projet, les cellules sont déjà créées et organisées dans un ordre précis. Il est essentiel que vous respectiez cet ordre et le contenu de chaque cellule. Votre tâche consiste à écrire le code dans les cellules désignées, en suivant les instructions fournies.
4. Faites attention à bien répondre aux bons endroits, le code python dans une cellule de CODE et les réponses textuelles dans une cellule de MARCAGE (***MARKDOWN***)
5. Assurez-vous que tout votre bloc note Jupyter (notebook) s'exécute correctement en une seule fois avec le bouton **Exécuter Tout** (***Run All***) de VS Code. Les cellules de code suivent un ordre, certaines reprennent le résultat d'une précédente. Faites attention à ne pas altérer les données entre deux cellules de code.
6. Pour vous faciliter cette tâche, utilisez régulièrement le bouton **Exécuter Tout** (***Run All***) pour vérifier que l'exécution arrive correctement là où vous êtes, n'attendez pas d'avoir tout codé pour utiliser ce bouton.

0.0.7 PARTIE 1: Importation des données et extraction d'informations (Pandas)

Importer les bibliothèques qui seront nécessaires au projet. Écrire votre code dans la cellule suivante.

[ ]:
```
# Complétez le code
import
import
import
```

Écrire le code pour charger les données dans le dataframe df_prediction. Écrire votre code dans la cellule suivante.

[ ]:
```
# Complétez le code
df_prediction =
```

Afficher les noms des colonnes.Écrire le code dans la cellule suivante.

[ ]:
```
# Complétez le code
df_prediction.
```

```
<b>Quel est le nom de la 5e colonne ?</b><br>
```

Écrire votre réponse dans cette cellule : (Double-cliquez dessus pour écrire votre réponse)

Modifier les noms de toutes les colonnes pour les traduire en français, tel qu'indiqué ci-dessous. Écrire le code dans la cellule suivante. **ATTENTION**: Vous devez obligatoirement utiliser les deux listes fournies dans le code: 'noms_actuels' et 'nouveaux_noms' et aussi une boucle.

- **Age** : Âge
- **Sex** : Sexe
- **ChestPainType** : Type de douleur thoracique
- **RestingBP** : Pression artérielle au repos
- **Cholesterol** : Cholestérol
- **FastingBS** : Glycémie à jeun
- **RestingECG** : ECG au repos
- **MaxHR** : Fréquence cardiaque maximale
- **ExerciseAngina** : Angine induite par l'exercice
- **ST_Slope** : Pente du segment ST
- **HeartDisease** : Maladie cardiaque

[ ]:
```
# Complétez le code

# Liste des noms actuels des colonnes
noms_actuels = ['Age', 'Sex', 'ChestPainType', 'RestingBP', 'Cholesterol',␣
 ↪'FastingBS', 'RestingECG', 'MaxHR', 'ExerciseAngina', 'ST_Slope',␣
 ↪'HeartDisease']

# Liste des nouveaux noms des colonnes
nouveaux_noms = ['Âge', 'Sexe', 'Type de douleur thoracique', 'Pression␣
 ↪artérielle au repos', 'Cholestérol', 'Glycémie à jeun', 'ECG au repos',␣
 ↪'Fréquence cardiaque maximale', 'Angine induite par l\'exercice', 'Pente du␣
 ↪segment ST', 'Maladie cardiaque']
```

```
# Utilisation d'une boucle for pour renommer les colonnes
for i in range(len(noms_actuels)):
    df_prediction =
```

Afficher juste les 2 premières lignes du DataFrame, pour confirmer le changement des noms des colonnes. Écrire le code dans la cellule suivante.

[ ]:
```
# Complétez le code
df_prediction.
```

Quels sont les types des colonnes “Fréquence cardiaque maximale” et “Oldpeak”?Écrire le code permettant d’obtenir les réponses dans la cellule suivante.Écrire vos réponses dans la cellule après celle du code.

[ ]:
```
# Complétez le code

df_prediction.
```

Écrire votre réponse dans cette cellule : (Double-cliquez dessus pour écrire votre réponse)
Type de ‘Fréquence cardiaque maximale’:
Type de ‘Oldpeak’:

Combien il y a t-il de données manquantes dans chaque colonne dans le dataFrame ?Écrire une fonction permettant d’obtenir les réponses dans la cellule suivante.Écrire vos réponses dans la fonction ‘print’ dans la cellule ci-dessous.

[ ]:
```
# Compléter le code

def compter_nb_valeurs_manquantes(df):
    # Compter le nombre de valeurs manquantes dans chaque colonne
    nombre_valeurs_manquantes =

    # Afficher le nombre de valeurs manquantes
    print(f"Il manque :\n{nombre_valeurs_manquantes}")

# Appel de la fonction
compter_nb_valeurs_manquantes(df_prediction)
```

- Définissez une fonction qui prend un dataframe en paramètre et qui retourne le dataframe avec aucune donnée manquante.
- Utilisez une boucle pour effectuer la recherche de valeurs manquantes.
- Utilisez cette fonction avec le dataframe pour remplacer les valeurs valeurs manquantes par 0.
- Utilisez la fonction ‘compter_nb_valeurs_manquantes()’ pour confirmer l’absence de valeurs manquantes.

[ ]:
```
# Complétez le code

# Définition de la fonction
```

```
def nettoyage_df(df_sale):
    # Convertir le DataFrame en liste
    liste_df =

    # Remplacer les valeurs manquantes par 0


    # Reconstruction du dataframe
    df_propre =

    # Retour du dtaframe nettoyé
    return df_propre

# Appel de la fonction
df_prediction = nettoyage_df(df_prediction)

# Confirmation qu'il n'y a plus de données manquantes
compter_nb_valeurs_manquantes()
```

Quelle est la moyenne d'âge des patients ?Écrire le code permettant d'obtenir les réponses dans la cellule suivante.Écrire vos réponses dans la cellule après celle du code.

[ ]:
```
# Complétez le code

df_prediction.
```

Écrire votre réponse dans cette cellule : (Double-cliquez dessus pour écrire votre réponse) Âge moyen des patients:

Sélection de la population **atteinte** de maladies cardiaques

Filtrez les personnes atteintes de maladie cardiaque. Nous voulons un dataframe avec seulement les patients (Hommes et Femmes) atteints de maladies cardiaques. Indices: - Filtrer la colonne 'Maladie cardiaque' - Un patient est atteint de maladies cardiaques si la valeur de la colonne 'Maladie cardiaque' est égale à 1

Combien il y a t-il de personnes atteintes de maladies cardiaques ?Écrire le code permettant d'obtenir les réponses dans la cellule suivante.Écrire la réponse avec la fonction 'print' écrite dans la cellule code.

[ ]:
```
# Complétez le code

df_cardiaques = df_prediction[(df_prediction[''] == )]
nombre_de_lignes = len()

print(f"Il y a {nombre_de_lignes} personnes atteintes de maladies cardiaques")
```

Combien il y a t-il de personnes non atteintes de maladies cardiaques ?Écrire le code permettant d'obtenir les réponses dans la cellule suivante.Écrire la réponse avec la fonction 'print' écrite dans la cellule code.

```
[ ]: # Complétez le code ci-dessous
df_non_cardiaques = df_prediction[(df_prediction[''] == )]
nombre_de_lignes = len()

print(f"Il y a {nombre_de_lignes} personnes non atteintes de maladies␣
  ↪cardiaques")
```

Sélection de la population féminine **atteinte** de maladies cardiaques

Filtrez les personnes féminines atteintes de maladie cardiaque. Nous voulons un dataframe avec seulement les patientes de sexe féminin et cardiaques. Indices: - Filtrer les colonnes 'Sexe' et 'Maladie cardiaque' - Un patient est atteint de maladies cardiaques si la valeur de la colonne 'Maladie cardiaque' est égale à 1

Combien il y a t-il de femmes atteintes de maladies cardiaques ?Écrire le code permettant d'obtenir les réponses dans la cellule suivante.Écrire la réponse avec la fonction 'print' écrite dans la cellule code.

```
[ ]: # Complétez le code ci-dessous
df_F_cardiaques = df_prediction[(df_prediction[''] == '') & (df_prediction['']␣
  ↪== )]
nombre_de_lignes = len()

print(f"Il y a {nombre_de_lignes} femmes atteintes de maladies cardiaques")
```

Sélection de la population féminine **non atteinte** de maladies cardiaques

Filtrez les personnes féminines non atteintes de maladie cardiaque. Nous voulons un dataframe avec seulement les patientes de sexe féminin et non cardiaques. Indices: - Filtrer les colonnes 'Sexe' et 'Maladie cardiaque' - Une patiente n'est pas atteinte de maladies cardiaques si la valeur de la colonne 'Maladie cardiaque' est égale à 0

Combien il y a t-il de femmes non atteintes de maladies cardiaques ?Écrire le code permettant d'obtenir les réponses dans la cellule suivante.Écrire la réponse avec la fonction 'print' écrite dans la cellule code.

```
[ ]: # Complétez le code ci-dessous
df_F_non_cardiaques = df_prediction[(df_prediction[''] == '') &␣
  ↪(df_prediction[''] == )]
nombre_de_lignes = len()

print(f"Il y a {nombre_de_lignes} femmes non atteintes de maladies cardiaques")
```

Que pouvez-vous conclure par rapport au nombre de femmes atteintes versus celles qui ne sont pas atteintes de maladies cardiaques ?Écrire la réponse dans la cellule suivante.

Écrire votre réponse dans cette cellule : (Double-cliquez dessus pour écrire votre réponse) Conclusion (F atteintes vs non atteintes):

Les patientes atteintes de maladies cardiaques sont moins nombreuses que celles sans maladies cardiaques.

Quelle est la moyenne du cholestérol pour les personnes (Hommes et Femmes) atteintes de maladies cardiaques ?Écrire le code permettant d'obtenir la réponse dans la cellule suivante. Pour ce faire: - Définissez une fonction qui prend un dataframe en paramètre et qui retourne la moyenne du cholestérol pour ce dataframe. - Utilisez une boucle pour effectuer votre calcul dans la fonction. - Utilisez cette fonction avec le dataframe contenant seulement les personnes atteintes de maladies cardiaques et affichez le résultat avec un `print`.

[ ]:
```
# Complétez le code de la fonction
def calcul_moyenne_cholesterol(donnees):



    return somme / len(donnees)

# Appel de la fonction
moyenne_cholesterol = calcul_moyenne_cholesterol(df_cardiaques)
print(f"La moyenne du cholestérol des patients atteints de maladies cardiaques␣
 ↪est de {round(moyenne_cholesterol,2)} mg/dl")
```

Vérifiez votre résultat avec la fonction 'describe()'. Écrire votre code dans la cellule suivante.

[ ]:
```
# Votre code ici
```

0.0.8 Partie 2. Visualiser graphiquement les données des patients (Matplotlib)

Entre les hommes et les femmes quel genre est plus nombreux parmis les patients ?Écrire le code permettant de créer un un diagramme à barres pour montrer le nombre de patients masculins et féminins., dans la cellule suivante.

[ ]:
```
# Complétez le code du graphique
# Graphique 1: Diagramme à barres du nombre de patients par sexe

# Compter le nombre de patients par sexe
nombre_patients = df_prediction[''].value_counts()

# Créer le diagramme à barres
plt.figure(figsize=(10, 6))
plt.bar(nombre_patients.index,     , color=['blue', 'pink'])
plt.xlabel('Sexe')
plt.ylabel('Nombre de patients')
plt.title('Nombre de patients par sexe')
plt.grid(True)
plt.show()
```

Quel est la tranche d'âges de la majorité des patients ?Écrire le code permettant de créer un histogramme pour visualiser la répartition des âges des patients, dans la cellule suivante.

[ ]:
```
# Complétez le code du graphique
# Graphique 2: Histogramme de la répartition des âges

```

```
plt.figure(figsize=(10, 6))
# Les âges
plt.hist(df_prediction[''], bins=10, edgecolor='black')
plt.xlabel('Âge')
plt.ylabel('Nombre de patients')
plt.title('Répartition des âges des patients')
plt.grid(True)
plt.show()
```

Quelle est la distribution des âges, entre les patients atteints et ceux pas atteints de maladies cardiaque ?Écrire le code permettant de créer un un histogramme pour montrer la distribution des patients atteints vs non atteints., dans la cellule suivante.

[ ]:
```
# Complétez le code du graphique
# Graphique 3: Histogramme de la distribution des âges selon qu'ils sont␣
 ↪atteints ou non

# Créer une liste contenant l'age des patients atteints
liste_age_atteints = df_cardiaques[''].values.tolist()

# Créer une liste contenant l'age des patients non atteints
liste_age_non_atteints = ['Âge'].values.tolist()

# Créer l'histogramme
plt.hist([  , ], bins=10, color=['skyblue', 'salmon'], edgecolor='black',␣
 ↪label=['Atteints', 'Non atteints'])

# Ajouter des titres et des labels
plt.title('Distribution de l\'âge')
plt.xlabel('Âge')
plt.ylabel('Nombre de patients')
plt.legend()

# Afficher le graphique
plt.show()
```

Quel est le type de douleur thoracique le plus fréquent parmis les patients ?Écrire le code permettant de créer un un diagramme en secteurs pour visualiser la répartition des types de douleur thoracique parmi les patients., dans la cellule suivante.

[ ]:
```
# Complétez le code du graphique
# Graphique 4: Diagramme en secteurs du type de douleur thoracique

# Compter le nombre de patients par type de douleur thoracique
nb_patients = df_prediction[''].value_counts()
secteur = (0, 0, 0, 0.2) # Écarte le 4e secteur
# Créer le diagramme en secteurs
plt.figure(figsize=(10, 6))
```

```
plt.pie(  .values, explode=secteur, labels=  .index, autopct='%1.1f%%',␣
 ↪startangle=140)
plt.title('Répartition des types de douleur thoracique')
plt.axis('equal')  # Assurer que le diagramme est circulaire
plt.show()
```

Quel est la tranche d'âge des paptients qui ont la fréquence cardiaque maximale parmis les patients ?Écrire le code permettant de créer un un Nuage de points de la fréquence cardiaque maximale en fonction de l'âge., dans la cellule suivante.

[ ]:
```
# Complétez le code du graphique
# Graphique 5: # Créer le nuage de points

plt.figure(figsize=(10, 6))
plt.scatter(df_prediction[''], df_prediction[''], alpha=0.7)
plt.xlabel('Âge')
plt.ylabel('Fréquence cardiaque maximale')
plt.title('Relation entre l\'âge et la fréquence cardiaque maximale')
plt.grid(True)
plt.show()
```

Quel est la tranche d'âge des patients qui ont la un taux de cholestérol élevé parmis les patients ?Écrire le code permettant de créer un nuage de points pour visualiser la relation entre l'âge et le cholestérol chez les patients atteints., dans la cellule suivante.

[ ]:
```
# Complétez le code du graphique
# Graphique 6: Nuage de points relation âge et taux de cholestérol

# Créer une liste contenant le taux de cholestérol des patients atteints
liste_cholesterol_atteints = df_cardiaques[''].values.tolist()

# Créer le nuage de points
plt.scatter(liste_age_atteints, , color='red')

# Ajouter des titres et des labels
plt.title('Âge vs Cholestérol')
plt.xlabel('Âge')
plt.ylabel('Cholestérol')

# Afficher le graphique
plt.show()
```

0.0.9 Partie 3. Analyse statistiques (NumPy)

Calcul de la moyenne et de l'écart-type de l'âge des patientsÉcrire le code NumPy dans la cellule suivante.

[ ]:
```
# Complétez le code

# Extraire les âges des patients
ages = df_prediction[''].values

# Calculer la moyenne et l'écart-type sur les âges
moyenne_age =
ecart_type_age =

print(f"Moyenne de l'âge des patients : {round(moyenne_age,2)}")
print(f"Écart-type de l'âge des patients : {round(ecart_type_age,2)}")
```

Calcul de la pression artérielle moyenne au repos par sexeÉcrire le code NumPy dans la cellule suivante.

[ ]:
```
# Complétez le code

# Extraire les pressions artérielles au repos par sexe
pression_homme = df_prediction[df_prediction['Sexe'] == ''][['Pression␣
 ↪artérielle au repos'].values
pression_femme = df_prediction[df_prediction['Sexe'] == ''][['Pression␣
 ↪artérielle au repos'].values

# Calculer la moyenne pour chaque sexe
moyenne_pression_homme =
moyenne_pression_femme =

print(f"Pression artérielle moyenne au repos (Hommes) :␣
 ↪{round(moyenne_pression_homme,2)}")
print(f"Pression artérielle moyenne au repos (Femmes) :␣
 ↪{round(moyenne_pression_femme,2)}")
```

Calcul de la médiane du cholestérolÉcrire le code NumPy dans la cellule suivante.

[ ]:
```
# Complétez le code

# Extraire les valeurs de cholestérol
cholesterol = df_prediction[''].values

# Calculer la médiane
medianne_cholesterol =

print(f"Médiane du cholestérol des patients : {medianne_cholesterol}")
```

Calcul de la fréquence cardiaque maximale moyenne pour les patients avec et sans maladie cardiaque.Écrire le code NumPy dans la cellule suivante.

[ ]:
```
# Complétez le code

# Extraire les fréquences cardiaques maximales pour chaque groupe
hr_max_avec_maladie = df_prediction[df_prediction['Maladie cardiaque'] ==␣
 ↪]['Fréquence cardiaque maximale'].values
hr_max_sans_maladie = df_prediction[df_prediction['Maladie cardiaque'] ==␣
 ↪]['Fréquence cardiaque maximale'].values

# Calculer la moyenne pour chaque groupe
moyenne_hr_max_avec_maladie =
moyenne_hr_max_sans_maladie =

print(f"Fréquence cardiaque maximale moyenne (avec maladie cardiaque) :␣
 ↪{round(moyenne_hr_max_avec_maladie,2)}")
print(f"Fréquence cardiaque maximale moyenne (sans maladie cardiaque) :␣
 ↪{round(moyenne_hr_max_sans_maladie,2)}")
```

Calcul de la proportion de patients ayant une angine induite par l'exercice.Écrire le code NumPy dans la cellule suivante.

[ ]:
```
# Complétez le code

# Extraire les valeurs d'angine induite par l'exercice
angine_par_exercice = df_prediction[''].values

# Calculer la proportion
proportion_angine = np.  (angine_par_exercice == 'Y') / len(angine_par_exercice)

print(f"Proportion de patients ayant une angine induite par l'exercice :␣
 ↪{round(proportion_angine,2)}")
```

Calcul de la Régression linéaire qui relie l'âge et le taux de cholestérol pour les personnes de 70 ans et plusÉcrire le code NumPy dans la cellule suivante.

[ ]:
```
# Complétez le code

# Filtrer les âges de 70 ans et plus
df_45_65 = df_prediction[(df_prediction['Âge'] )]

# Extraction des âges et des taux de cholestérol dans des tableaux NumPy
df_ages = df_45_65['Âge'].
df_cholesterol = df_45_65['Cholestérol'].

# Calcul de la droite de régression linéaire
coefficients = np.polyfit(df_ages, df_cholesterol, 1)
polynomial = np.poly1d(coefficients)
regression_line = polynomial(df_ages)
```

```
# Tracer les points de données et la droite de régression
plt.scatter(df_ages, df_cholesterol, color='blue', label='Données')
plt.plot(df_ages, regression_line, color='red', label='Droite de régression')

# Ajouter des labels et un titre
plt.xlabel("Âge")
plt.ylabel("Cholestérol")
plt.title("Relation entre l'âge et le cholestérol (70 ans et plus)")
plt.legend()

# Afficher le graphique
plt.show()
```

Calcul du coefficient de corrélation entre l'âge et le taux de cholestérol pour les personnes de 70 ans et plus Écrire le code NumPy dans la cellule suivante, ensuite interprétez le coefficient de corrélation et écrire en 1-2 phrases la conclusion que vous en tirez.

[ ]:
```
# Complétez le code
x =
y =
matrice_correlation =
print(matrice_correlation)
```

Écrire votre réponse dans cette cellule : (Double-cliquez dessus pour écrire votre réponse) Interprétation du coefficient de correlation: