**Plan de Projet : Prédiction de l'Absentéisme des Etudiants**

**1. Introduction**

Objectif : Développer un modèle prédictif pour classifier l'absentéisme des Etudiants en se basant sur diverses caractéristiques.

**2. Collecte et Préparation des Données**

Source des Données : Ensemble de données sur l'absentéisme des etudiants.

Inspection Initiale des Données :

Charger l'ensemble de données.

Afficher les premières lignes pour comprendre sa structure et son contenu.

**3. Nettoyage et Prétraitement des Données**

Gestion des Valeurs Manquantes :

Vérifier les valeurs manquantes dans l'ensemble de données.

Décider des méthodes d'imputation (moyenne, médiane, mode) ou supprimer des lignes/colonnes si nécessaire.

Sélection des Caractéristiques :

Identifier les caractéristiques numériques et catégorielles.

Convertir les caractéristiques catégorielles en valeurs numériques en utilisant des techniques de codage.

Ingénierie des Caractéristiques :

Créer de nouvelles caractéristiques si nécessaire.

Standardiser ou normaliser les caractéristiques numériques pour qu'elles soient sur une échelle similaire.

**4. Analyse Exploratoire des Données (EDA)**

Visualisation :

Tracer des histogrammes, des diagrammes en barres et des boîtes à moustaches pour comprendre la distribution des caractéristiques.

Utiliser des diagrammes de dispersion pour identifier les relations entre les caractéristiques numériques et la variable cible (temps d'absentéisme en heures).

Analyse Statistique :

Calculer les corrélations entre les caractéristiques numériques et la variable cible.

Identifier la skewness et la kurtosis pour comprendre la distribution du temps d'absentéisme en heures.

**5. Sélection et Transformation des Caractéristiques**

Test du Chi-Deux :

Utiliser le test du Chi-Deux pour sélectionner les caractéristiques les plus significatives pour prédire l'absentéisme.

Transformer la variable cible en un problème de classification binaire.

**6. Séparation des Données**

Division Entraînement-Test :

Diviser les données en ensembles d'entraînement et de test (par exemple, 80 % entraînement, 20 % test).

Assurer une division aléatoire et maintenir la distribution de la variable cible.

**7. Sélection et Entraînement des Modèles**

Initialisation des Modèles :

Initialiser divers modèles : Régression Logistique, Machine à Vecteurs de Support (SVM), K-Plus Proches Voisins (KNN), Arbre de Décision et Réseau de Neurones.

Tuning des Hyperparamètres :

Utiliser GridSearchCV pour trouver les meilleurs hyperparamètres pour chaque modèle.

Évaluer les modèles en utilisant la validation croisée pour assurer leur robustesse.

**8. Évaluation des Modèles**

Métriques de Performance :

Calculer la précision de validation et la précision de test pour chaque modèle.

Comparer les modèles en fonction de leurs métriques de performance et sélectionner le meilleur.

**9. Sérialisation des Modèles**

Sauvegarde des Modèles :

Sérialiser les modèles entraînés en utilisant joblib pour une utilisation future.

Sauvegarder les paramètres et hyperparamètres des modèles.

**10. Modèle de Réseau de Neurones**

Construction du Réseau de Neurones :

Définir l'architecture du réseau de neurones en utilisant Keras.

Compiler et entraîner le modèle.

Évaluer le modèle en utilisant la validation croisée et les ensembles de test.

**11. Résultats et Discussion**

Analyse Comparative :

Comparer la performance de tous les modèles.

Discuter des forces et des faiblesses de chaque modèle.

Mettre en évidence le modèle le plus performant et justifier ce choix.

**12. Conclusion**

Résumé des Résultats :

Résumer les principaux résultats de l'EDA, de l'entraînement des modèles et de l'évaluation.

Discuter des implications des résultats pour la prédiction de l'absentéisme.

Travaux Futurs :

Suggérer des améliorations potentielles et des directions de recherche futures.

Discuter d'autres caractéristiques ou sources de données qui pourraient améliorer la performance du modèle.

**Étapes Détaillées avec Références de Code**

**\*Chargement et Inspection Initiale des Données :**

data = pd.read\_csv("Absenteeism\_at\_work.csv")

print(data.head())

-Gestion des Valeurs Manquantes :

data.fillna(data.median(), inplace=True)

**\*Sélection et Codage des Caractéristiques :**

categorical\_features = data.select\_dtypes(include=['object']).columns

data = pd.get\_dummies(data, columns=categorical\_features)

**\*EDA et Visualisation :**

sns.histplot(data['Absenteeism time in hours'])

plt.show()

**\*Ingénierie et Transformation des Caractéristiques :**

data['absenteeism'] = data['Absenteeism time in hours'].apply(lambda x: 1 if x > data['Absenteeism time in hours'].median() else 0)

**\*Division Entraînement-Test :**

from sklearn.model\_selection import train\_test\_split

x = data.drop(columns=['Absenteeism time in hours', 'absenteeism'])

y = data['absenteeism']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=123)

**\*Entraînement et Évaluation des Modèles :**

from sklearn.linear\_model import LogisticRegression

logistic\_model = LogisticRegression(random\_state=123)

logistic\_model.fit(x\_train, y\_train)

**\*Modèle de Réseau de Neurones :**

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, BatchNormalization, Input

from sklearn.model\_selection import cross\_val\_score

def nn\_model():

model = Sequential()

model.add(Input(shape=(x.shape[1],)))

model.add(BatchNormalization())

model.add(Dense(units=len(data.index), activation='tanh'))

model.add(BatchNormalization())

model.add(Dense(units=len(data.index) // 5, activation='tanh'))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

return model