

S

Je veux utiliser ce data set :



[https://www.kaggle.com/datasets/shivamb/netflix-shows?](https://www.kaggle.com/datasets/shivamb/netflix-shows?resource=download&select=netflix_titles.csv)
[resource=download&select=netflix_titles.csv](https://www.kaggle.com/datasets/shivamb/netflix-shows?resource=download&select=netflix_titles.csv)

pour mon projet afin de prédire la popularité des films Netflix en utilisant des modèles d'apprentissage. Quel type d'apprentissage utiliser? comment dois je procéder? et quelles sont les spécifications de ce projet ?



Pour votre projet de prédiction de la popularité des films Netflix, vous pouvez utiliser des modèles d'apprentissage supervisé tels que la régression linéaire, les arbres de décision, les forêts aléatoires ou les réseaux de neurones.



Voici les étapes que vous pouvez suivre pour procéder à votre projet :

1. Importez le fichier CSV dans un environnement de développement Python tel que Jupyter Notebook ou Google Colab.
2. Explorez le jeu de données en utilisant des techniques d'analyse exploratoire des données (AED) pour comprendre les caractéristiques du jeu de données et comment il peut être utilisé pour la prédiction.
3. Préparez le jeu de données en effectuant des étapes de prétraitement telles que l'encodage des variables catégorielles, la gestion des valeurs manquantes, la normalisation des données, etc.
4. Divisez le jeu de données en ensembles de formation et de test pour évaluer les performances des modèles d'apprentissage.
5. Entraînez plusieurs modèles d'apprentissage supervisé sur l'ensemble de formation.
6. Évaluez les performances des modèles sur l'ensemble de test en utilisant des métriques telles que l'erreur quadratique moyenne (EQM), le coefficient de détermination (R^2), etc.
7. Sélectionnez le modèle qui a les meilleures performances sur l'ensemble de test.
8. Utilisez le modèle sélectionné pour prédire la popularité de nouveaux films Netflix.



quelles sont les étapes à suivre pour réaliser ce projet ?



Voici les étapes à suivre pour réaliser le projet de création d'un modèle de prédiction de la popularité des films Netflix en utilisant l'apprentissage supervisé :



1. Collecte des données :

- Télécharger les données sur Kaggle : https://www.kaggle.com/datasets/shivamb/netflix-shows?resource=download&select=netflix_titles.csv

2. Exploration des données :

- Charger les données dans un environnement de travail tel que Jupyter Notebook ou Google Colab.
- Explorer les données pour identifier les caractéristiques des variables et détecter des valeurs manquantes ou des incohérences dans les données.
- Visualiser les données pour avoir une compréhension plus profonde des variables et des relations entre elles.

3. Prétraitement des données :

- Nettoyer les données en supprimant les valeurs manquantes ou en les remplaçant par des valeurs appropriées.
- Convertir les variables textuelles en variables catégorielles ou numériques pour une utilisation dans un modèle d'apprentissage supervisé.
- Sélectionner les variables les plus pertinentes pour le modèle en utilisant des techniques d'analyse de corrélation ou d'analyse de variance.

4. Séparation des données :

- Diviser les données en un ensemble d'entraînement et un ensemble de test pour évaluer les performances du modèle.

5. Modélisation :

- Choisir un modèle approprié pour la tâche de prédiction de la popularité des films Netflix, tel que la régression linéaire, la régression logistique ou un modèle de forêt aléatoire.
- Entraîner le modèle sur l'ensemble d'entraînement et ajuster les hyperparamètres pour améliorer les performances.
- Évaluer les performances du modèle sur l'ensemble de test en utilisant des métriques telles que la RMSE, le R2, etc.

6. Amélioration du modèle :


- Effectuer des techniques de sélection de fonctionnalités ou de réduction de dimensionnalité pour améliorer les performances du modèle.
- Réajuster les hyperparamètres ou choisir un modèle plus sophistiqué pour améliorer les performances.

7. Livraison :

- Livrer le modèle documenté avec une explication des techniques utilisées, les performances obtenues et des recommandations pour l'amélioration du modèle.

Voici un exemple de code Python pour réaliser certaines de ces étapes :

python

 Copy code

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Collecte des données
data = pd.read_csv('netflix_titles.csv')

# Exploration des données
print(data.head())
print(data.describe())
print(data.info())

# Prétraitement des données
data = data.drop(['show_id', 'description'], axis=1)
data = pd.get_dummies(data, columns=['type', 'rating', 'country', 'listed_in'])
data = data.fillna(data.mean())

# Séparation des données
X = data.drop('popularity', axis=1)
y = data['popularity']
X_train, X_test, y_train, y_test = train_test_split(X, y, test
```



Pouvez vous rajouter des commentaires à ces lignes pour dire à quoi sert chaque importation :



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```



Bien sûr, voici des commentaires pour chaque ligne d'importation :



python

Copy code

```
import pandas as pd
```

Cette ligne permet d'importer la bibliothèque pandas, qui est utilisée pour la manipulation et l'analyse de données.

python

Copy code

```
from sklearn.model_selection import train_test_split
```

Cette ligne permet d'importer la fonction train_test_split de Scikit-learn, qui permet de diviser les données en un ensemble d'entraînement et un ensemble de test.

python

Copy code

```
from sklearn.linear_model import LinearRegression
```

Cette ligne permet d'importer le modèle de régression linéaire de Scikit-learn, qui sera utilisé pour prédire la popularité des films Netflix.

python

Copy code

```
from sklearn.metrics import mean_squared_error, r2_score
```

Cette ligne permet d'importer les métriques RMSE et R2 de Scikit-learn, qui seront utilisées pour évaluer les performances du modèle. RMSE mesure l'écart entre les valeurs prédites et les valeurs réelles, tandis que R2 mesure la qualité des prédictions par rapport aux valeurs moyennes.

S

En utilisant `data.info()` on remarque qu'il y a des valeurs à null sur quelques colonnes .
Comment procéder ?



Si ``data.info()`` indique qu'il y a des valeurs manquantes (c'est-à-dire des valeurs ``NaN``), il est important de les traiter avant de construire un modèle de prédiction. Les valeurs manquantes peuvent fausser les résultats des analyses et des prédictions.

Voici quelques étapes que vous pouvez suivre pour gérer les valeurs manquantes :

1. Identifier les colonnes avec des valeurs manquantes en utilisant la méthode ``isnull()`` de Pandas :

python

Copy code

```
import pandas as pd

# Charger le fichier CSV contenant les données Netflix
netflix_data = pd.read_csv('chemin/vers/le/fichier/netflix_titles.csv')


# Afficher le nombre de valeurs manquantes pour chaque colonne
print(netflix_data.isnull().sum())
```

2. Gérer les valeurs manquantes en fonction de leur nature et de la nature de la colonne.

Voici quelques exemples de techniques courantes :

- Supprimer les lignes qui contiennent des valeurs manquantes avec la méthode `dropna()` de Pandas :

python


 Copy code

```
# Supprimer les lignes avec des valeurs manquantes dans toutes les colonnes
netflix_data = netflix_data.dropna()

# Supprimer les lignes avec des valeurs manquantes dans une colonne spécifique
netflix_data = netflix_data.dropna(subset=['colonne_specifique'])
```

- Remplacer les valeurs manquantes par une valeur par défaut ou la moyenne/la médiane de la colonne avec la méthode `fillna()` de Pandas :

python

 Copy code

```
# Remplacer les valeurs manquantes par une valeur par défaut
netflix_data['colonne_specifique'] = netflix_data['colonne_specifique'].fillna(valeur_defaut)

# Remplacer les valeurs manquantes par la moyenne/la médiane de la colonne
netflix_data['colonne_specifique'] = netflix_data['colonne_specifique'].fillna(netflix_data['colonne_specifique'].mean())
```

S

dans le cas de notre projet, quelle méthode est la plus appropriée pour traiter les valeurs manquantes sachant que le résultat de `data.isnull().sum()` est :



```
show_id    0
type       0
title      0
director   2634
cast       825
country    831
date_added  10
release_year 0
rating     4
duration   3
listed_in  0
description 0
dtype: int64
```



Dans le cas de votre projet, vous pouvez choisir la méthode la plus appropriée en fonction des colonnes qui contiennent des valeurs manquantes et de la nature de ces valeurs manquantes.



Voici quelques suggestions pour gérer les valeurs manquantes dans les colonnes :

- ``director``, ``cast``, et ``country`` : ces colonnes contiennent des valeurs catégorielles qui sont difficiles à remplacer par des moyennes ou des valeurs par défaut. Une approche raisonnable consiste à supprimer les lignes avec des valeurs manquantes. Cependant, il est important de vérifier si ces colonnes sont essentielles à l'analyse ou à la prédiction. Si vous souhaitez utiliser ces colonnes dans votre modèle, vous pouvez envisager de remplacer les valeurs manquantes par une catégorie "inconnue" ou "non spécifiée" qui ne biaise pas les résultats.
- ``date_added``, ``rating`` et ``duration`` : ces colonnes contiennent des valeurs numériques ou des dates qui peuvent être remplacées par des moyennes ou des valeurs par défaut. Pour ``rating`` et ``duration``, vous pouvez remplacer les valeurs manquantes par la moyenne de la colonne. Pour ``date_added``, vous pouvez remplacer les valeurs manquantes par la date la plus fréquente ou la plus récente.



Voici le code pour effectuer les trois tâches mentionnées :



python

Copy code

```
import pandas as pd

# Charger le fichier CSV contenant les données Netflix
netflix_data = pd.read_csv('chemin/vers/le/fichier/netflix_titles.csv')

# Remplacer les valeurs manquantes par 'non spécifiée'
netflix_data['director'].fillna(value='non spécifiée', inplace=True)
netflix_data['cast'].fillna(value='non spécifiée', inplace=True)
netflix_data['country'].fillna(value='non spécifiée', inplace=True)

# Supprimer les lignes avec des valeurs manquantes dans les colonnes 'duration' et 'rating'
netflix_data.dropna(subset=['duration', 'rating'], inplace=True)

# Remplacer les valeurs manquantes dans la colonne 'date_added' par la date la plus fréquente
most_common_date = netflix_data['date_added'].mode()[0]
netflix_data['date_added'].fillna(value=most_common_date, inplace=True)
```

Concernant la dernière partie, remplacer les valeurs manquantes dans la colonne 'date_added' par la date la plus fréquente peut être une solution si le nombre de valeurs manquantes est faible par rapport à la taille du jeu de données. Cependant, cette solution peut introduire un biais dans l'analyse si la date la plus fréquente est différente de la distribution générale des dates. Il est donc important de comprendre la distribution des dates dans le jeu de données avant de prendre une décision sur la façon de traiter les valeurs manquantes dans la colonne 'date_added'.

S

dans le cas de notre projet, il y a 10 lignes où 'date_added' est null, le nombre total de lignes est 8807



Dans ce cas, remplacer les valeurs manquantes dans la colonne 'date_added' par la date la plus fréquente (ou moyenne) pourrait être une option raisonnable, étant donné que le nombre de lignes avec des valeurs manquantes est faible par rapport au nombre total de lignes dans le jeu de données. Cependant, il est important de visualiser la distribution des dates ajoutées pour s'assurer que la date la plus fréquente (ou moyenne) est représentative de la distribution globale des dates. Si la distribution des dates ajoutées est très dispersée, il peut être préférable de supprimer les lignes avec des valeurs manquantes plutôt que de remplacer ces valeurs par une valeur moyenne ou fréquente.





Après le traitement des valeurs manquantes, il est recommandé de réaliser une analyse exploratoire de données (EDA) afin de mieux comprendre notre jeu de données. Voici quelques idées de ce que vous pourriez faire dans le cadre de l'EDA :



- Afficher les statistiques descriptives pour chaque colonne numérique : cela permettra de mieux comprendre la distribution de chaque variable numérique et de détecter les outliers.
- Visualiser la distribution de chaque variable numérique : cela peut se faire à l'aide d'histogrammes ou de boxplots. Cela permettra de mieux comprendre la forme de la distribution et d'identifier les outliers.
- Afficher la distribution des variables catégorielles : cela peut se faire à l'aide de tableaux de fréquences ou de graphiques en barres. Cela permettra de mieux comprendre la répartition des différentes catégories dans chaque variable.
- Visualiser les corrélations entre les variables : cela peut se faire à l'aide d'une matrice de corrélation ou de graphiques de dispersion. Cela permettra de mieux comprendre les relations entre les variables et d'identifier les variables qui sont fortement corrélées.
- Visualiser la popularité des films : cela peut se faire à l'aide de graphiques de densité ou de graphiques en boîte en fonction de différentes variables. Cela permettra de mieux comprendre les distributions de popularité pour différents groupes de films en fonction des variables choisies.

L'EDA est une étape importante dans le processus de modélisation, car elle permet de mieux comprendre les caractéristiques des données et de détecter les éventuels problèmes qui pourraient affecter la qualité du modèle.