Laboratoire pratique : IA générative pour la maintenance et l'administration des référentiels de données

Effort de l'apprenant : 30 minutes

Introduction

La gestion et l'administration d'un référentiel de données constituent un aspect crucial de l'ingénierie des données. Elle comprend tous les aspects qui garantissent le maintien de la qualité, de l'intégrité et de l'exhaustivité des données. Dans ce laboratoire, vous explorerez les différents aspects de l'administration et de la gestion d'un référentiel de données dans le contexte d'un ensemble de données du monde réel.

Objectifs

Dans ce laboratoire, vous apprendrez à utiliser l'IA générative pour générer des codes qui effectueront les tâches suivantes :

1. Gérer les entrées manquantes dans les données
2. Supprimer les entrées en double des données
3. Détection de valeurs aberrantes dans les données

Interface de test

L'interface de test à utiliser pour ce laboratoire est disponible sur la page du cours à la fin de cette leçon. Veuillez la garder ouverte dans un navigateur sur le côté et suivre les étapes mentionnées pour la configurer.

Ensemble de données

L'ensemble de données utilisé pour ce laboratoire est un sous-ensemble modifié qui cartographie le prix des ordinateurs portables avec différents attributs. L'ensemble de données est une version filtrée et modifiée de l'ensemble de données « Prédiction du prix des ordinateurs portables à l'aide des spécifications », disponible sous la licence Database Contents License (DbCL) v1.0 sur le site Web de Kaggle.

L'ensemble de données a été rendu disponible à l'URL suivante.

```
URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-AI0273EN-SkillsNetwork/labs/v1/m2/data/la
```

Gérer les valeurs manquantes

La première chose à faire avec n'importe quelle donnée est de gérer les données manquantes. Les données manquantes peuvent être traitées de plusieurs manières.

1. Si l'attribut pour lequel la valeur est manquante est catégoriel ou discret, remplacez-le par la valeur la plus fréquente.
2. Si l'attribut pour lequel la valeur est manquante est continu, remplacez-le par la valeur moyenne ou médiane.
3. Si l'attribut pour lequel la valeur est manquante est l'attribut cible, il est préférable de supprimer toute cette ligne de données.

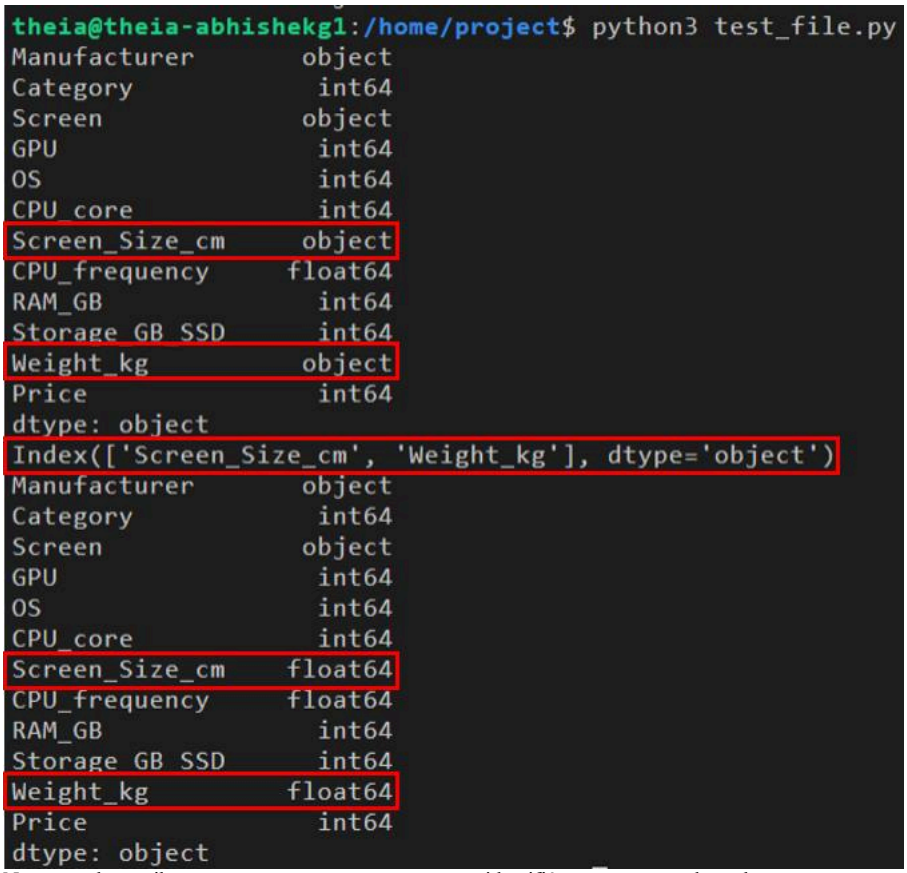
Dans ce cas, les points de données manquants se trouvent dans des variables continues et les données manquantes sont représentées par le caractère `?`. Vous pouvez utiliser le modèle Generative AI pour écrire un code qui gère ces valeurs manquantes à votre place.

```
Write a Python code that reads a csv record from a URL and does the following.
1. Identify the columns which have values as "?" in them.
2. Replace these with the mean value of the respective attribute.
2. Modify the data type of the attribute to float after replacement.
```

Vous pouvez vous attendre à un code, similaire à celui indiqué ci-dessous, en guise de réponse.

```
import pandas as pd
# Read CSV record from URL
url = 'https://example.com/data.csv'
data = pd.read_csv(url)
print(data.dtypes)
# Identify columns with "?" values
cols_with_question_mark = data.columns[data.isin(['?']).any()]
print(cols_with_question_mark)
# Replace "?" values with the mean value of the respective attribute
for col in cols_with_question_mark:
    mean_val = pd.to_numeric(data[col], errors='coerce').mean()
    data[col] = pd.to_numeric(data[col].replace('?', mean_val))
# Modify the data type of the attribute to float after replacement
data[cols_with_question_mark] = data[cols_with_question_mark].astype(float)
# Print the modified data
print(data.dtypes)
```

Vous pouvez mettre à jour ce code avec l'URL de l'ensemble de données tel que partagé auparavant. Le résultat de l'exécution de ce code devrait être tel qu'indiqué dans l'image ci-dessous.



Notez que les attributs `Screen Size`et `Weight`sont ceux identifiés comme ayant des valeurs manquantes et, par conséquent, leur type de données était précédemment `object`. Après modification et remplissage avec les valeurs moyennes des attributs, le type de données a été modifié en `float64`.

Supprimer les doublons

Ensuite, vous devez supprimer les entrées en double de l'ensemble de données. Vous pouvez créer le code Python requis pour cela à l'aide de l'IA générative avec l'invite suivante.

```
For a data in a python dataframe, write a code that removes all duplicate entries from the data. Print the total number of rows in t
```

Vous pouvez vous attendre à voir le code suivant en réponse.

```
import pandas as pd
# Assuming 'data' is the pandas DataFrame containing the data
# Print the total number of rows before removal
print("Total number of rows before removal:", len(data))
# Remove duplicate entries
data.drop_duplicates(inplace=True)
# Print the total number of rows after removal
print("Total number of rows after removal:", len(data))
# Print the modified data
print(data)
```

Cet extrait de code supprimera toutes les entrées en double des « données » du DataFrame et imprimera le nombre total de lignes avant et après la suppression pour démontrer l'efficacité du code.

Vous pouvez ajouter la partie pertinente de ce code à votre fichier existant sur l'interface de test. Assurez-vous de commenter les instructions d'impression de l'étape précédente afin d'avoir une visibilité claire du résultat de celle-ci.

Le contenu de votre code de test devrait maintenant ressembler à peu près à celui indiqué ci-dessous.

```
import pandas as pd
# Read CSV record from URL
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-AI0273EN-SkillsNetwork/labs/v1/m2/data/la
data = pd.read_csv(url)
#print(data.dtypes)
# Identify columns with "?" values
cols_with_question_mark = data.columns[data.isin(['?']).any()]
#print(cols_with_question_mark)
# Replace "?" values with the mean value of the respective attribute
for col in cols_with_question_mark:
    mean_val = pd.to_numeric(data[col], errors='coerce').mean()
    data[col] = pd.to_numeric(data[col].replace('?', mean_val))
# Modify the data type of the attribute to float after replacement
data[cols_with_question_mark] = data[cols_with_question_mark].astype(float)
# Print the modified data
#print(data.dtypes)
print("Total number of rows before removal:", len(data))
# Remove duplicate entries
data.drop_duplicates(inplace=True)
# Print the total number of rows after removal
print("Total number of rows after removal:", len(data))
```

Notez que les instructions d'impression précédentes ont été commentées.
Le résultat attendu est

```
theia@theia-abhishekg1:/home/project$ python3 test_file.py
Total number of rows before removal: 243
Total number of rows after removal: 238
```

Cela montre qu'il y avait 5 entrées en double dans l'ensemble de données, qui ont maintenant été supprimées.

Identification des valeurs aberrantes

Vous devez maintenant tester la qualité des données et identifier s'il existe des valeurs aberrantes anormales dans l'attribut cible « Prix ». Vous pouvez simplement utiliser l'invite suivante sur la plateforme d'IA générative pour créer un code qui extraira les entrées requises.

```
Write a python code to extract entries of a dataframe where the attribute 'Price' has outliers that might be anomalies in comparisor
```

Vous pouvez vous attendre au code suivant en réponse.

```
import pandas as pd
import numpy as np
# Assuming 'data' is the pandas DataFrame containing the data
# Calculate the mean and standard deviation of the prices
mean_price = data['Price'].mean()
std_price = data['Price'].std()
# Define a threshold for outliers (e.g., 3 standard deviations away from the mean)
outlier_threshold = 3
# Identify outliers using the threshold
outliers = data[(np.abs(data['Price'] - mean_price) > outlier_threshold * std_price)]
# Extract entries with price outliers
entries_with_outliers = data[data['Price'].isin(outliers['Price'])]
# Print the entries with price outliers
print("Entries with price outliers:")
print(entries_with_outliers)
```

Cet extrait de code calcule la moyenne et l'écart type des prix dans l'ensemble de données, identifie les valeurs aberrantes en fonction d'un seuil spécifié, puis extrait les entrées où l'attribut « Prix » présente des valeurs aberrantes. Cela vous aidera à isoler et à analyser les entrées avec des valeurs de prix anormales par rapport au reste des données.

Vous pouvez utiliser la partie pertinente de ce code et l'ajouter au fichier existant sur l'interface de test. Assurez-vous de commenter les instructions d'impression précédentes pour voir correctement la sortie de cet extrait de code.

Le code mis à jour sur l'interface de test doit être comme indiqué ci-dessous.

```
import pandas as pd
# Read CSV record from URL
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-AI0273EN-SkillsNetwork/labs/v1/m2/data/la
data = pd.read_csv(url)
#print(data.dtypes)
# Identify columns with "?" values
cols_with_question_mark = data.columns[data.isin(['?']).any()]
#print(cols_with_question_mark)
# Replace "?" values with the mean value of the respective attribute
for col in cols_with_question_mark:
    mean_val = pd.to_numeric(data[col], errors='coerce').mean()
    data[col] = pd.to_numeric(data[col].replace('?', mean_val))
# Modify the data type of the attribute to float after replacement
data[cols_with_question_mark] = data[cols_with_question_mark].astype(float)
# Print the modified data
#print(data.dtypes)
#print("Total number of rows before removal:", len(data))
# Remove duplicate entries
data.drop_duplicates(inplace=True)
# Print the total number of rows after removal
#print("Total number of rows after removal:", len(data))
import numpy as np
# Assuming 'data' is the pandas DataFrame containing the data
# Calculate the mean and standard deviation of the prices
mean_price = data['Price'].mean()
std_price = data['Price'].std()
# Define a threshold for outliers (e.g., 3 standard deviations away from the mean)
outlier_threshold = 3
# Identify outliers using the threshold
outliers = data[(np.abs(data['Price'] - mean_price) > outlier_threshold * std_price)]
# Extract entries with price outliers
entries_with_outliers = data[data['Price'].isin(outliers['Price'])]
# Print the entries with price outliers
print("Entries with price outliers:")
print(entries_with_outliers)
```

La sortie de ce code sera

```
theia@theia-abhishekg1:/home/project$ python3 test_file.py
Entries with price outliers:
    Manufacturer  Category     Screen  GPU  OS  ...  CPU_frequency  RAM_GB  Storage_GB_SSD  Weight_kg  Pric
64          Asus         1    Full HD    3   1  ...            2.9      16             256       3.60   381
77          Dell         5    Full HD    3   1  ...            2.9      16             256       3.42   366
144       Lenovo         3  IPS Panel    3   1  ...            2.8       8             256       3.40   381
159        Razer         1    Full HD    3   1  ...            2.8      16             256       1.95   330

[4 rows x 12 columns]
```

Comme on peut le constater, ces données sont identifiées comme ayant des valeurs de prix anormales.

Conclusion

Félicitations pour avoir terminé ce laboratoire !

Vous avez maintenant appris à utiliser l'IA générative pour :

1. Gérer les valeurs manquantes dans les données
2. Supprimer les valeurs en double des données
3. Identifier les données aberrantes

Auteur(s)

Abhishek Gagneja