



Laboratoire pratique : IA générative pour l'analyse et l'exploration de données

Effort estimé : 30 minutes

Introduction

L'une des dernières tâches effectuées par un ingénieur de données consiste à analyser les données finales, à en tirer des enseignements et à utiliser des stratégies d'exploration de données pour extraire des modèles cachés dans la distribution des données. Dans ce laboratoire, vous apprendrez à utiliser l'IA générative pour créer des codes Python capables d'effectuer les stratégies d'analyse et d'exploration de données requises.

Scénario

En tant qu'ingénieur de données senior pour une entreprise de soins de santé, vous êtes chargé d'effectuer des analyses et des explorations de données sur les dossiers médicaux des patients indiquant si le patient a été diagnostiqué ou non avec une maladie du foie. D'autres équipes ont enregistré et nettoyé les données prêtes à être analysées.

Objectifs

Dans ce laboratoire, vous apprendrez à utiliser l'IA générative pour :

1. Effectuer une analyse exploratoire des données sur un ensemble de données donné.
2. Effectuer l'exploration de données sur l'ensemble de données donné et tirer des enseignements des données.

Ensemble de données

Dans le cadre de ce laboratoire, nous utilisons l' [ensemble de données des patients indiens souffrant de troubles hépatiques](#) , accessible au public sous la licence [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](#) . Vous pouvez vous référer à la page Web de l'ensemble de données pour plus de détails sur les attributs.

L'ensemble de données est disponible pour une utilisation dans ce laboratoire à l'URL suivante :

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

Interface de test

Vous trouverez une interface de test distincte à la fin de la leçon sur la page du cours. Veuillez garder cette interface de test ouverte en tant que laboratoire séparé sur le côté et suivre les étapes initiales pour être prêt avec la configuration.

Analyse exploratoire des données

Grâce à l'analyse exploratoire des données, vous pouvez identifier le type de distribution des données et la manière dont les différentes variables s'influencent mutuellement. Dans cette tâche, vous utiliserez l'IA générative pour créer un code qui effectue les opérations suivantes :

1. Créez un histogramme des données en fonction des attributs « Âge », « Sexe » et « Sélecteur ».
2. Créez une carte thermique de corrélation pour l'ensemble de données.
3. Sélectionnez les 5 principales fonctionnalités avec la valeur de corrélation la plus élevée avec l'attribut « Sélecteur ».

Remarque : enregistrez les graphiques générés sur la machine locale pour les visualiser. Vous ne pouvez pas ouvrir d'images sur cette plateforme.

Vous pouvez utiliser la classe GenAI avec l'invite suivante :

```
For a given data set available in a CSV format, write a python code that does the following:  
1. Load the CSV data to a pandas data frame assuming the first row of the CSV contains the headers of the data.  
2. Save an image to disk, containing histograms of the data distribution with respect to a continuous attribute 'Age' and discrete  
3. Save an image to disk containing the correlation heatmap of the data set.  
4. Identify the top 5 features with highest absolute correlation with the attribute 'Selector'.
```

Vous pouvez vous attendre à ce que le code généré soit similaire à celui partagé ci-dessous.

```
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
# Load the CSV data into a pandas data frame  
data = pd.read_csv('data.csv')  
# Save histograms of data distribution for 'Age', 'Gender', and 'Selector'  
plt.figure()  
data['Age'].hist()  
plt.savefig('age_histogram.png')  
plt.figure()  
data['Gender'].hist()  
plt.savefig('gender_histogram.png')  
plt.figure()  
data['Selector'].hist()  
plt.savefig('selector_histogram.png')  
# Save correlation heatmap of the data set
```

```
plt.figure()
corr = data.corr()
sns.heatmap(corr, annot=True)
plt.savefig('correlation_heatmap.png')
# Identify top 5 features with highest absolute correlation with 'Selector'
correlation_with_selector = corr['Selector'].abs().sort_values(ascending=False)
top_5_features = correlation_with_selector[1:6]
# Exclude 'Selector' itself
print(top_5_features)
```

Vous pouvez modifier l'URL, ajuster quelques lignes de code pour rendre la sortie un peu plus utilisable, et le code final devrait ressembler à celui indiqué ci-dessous.

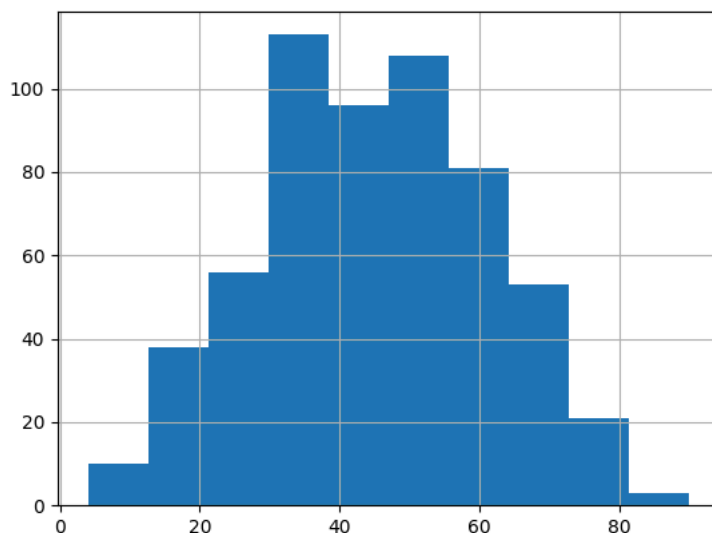
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Load the CSV data into a pandas data frame
data = pd.read_csv('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-AI0273EN-SkillsNetwork/labs/
# Save histograms of data distribution for 'Age', 'Gender', and 'Selector'
plt.figure()
data['Age'].hist()
plt.savefig('age_histogram.png')
plt.figure()
data['Gender'].hist()
plt.savefig('gender_histogram.png')
plt.figure()
data['Selector'].hist()
plt.savefig('selector_histogram.png')
# Save correlation heatmap of the data set
plt.figure(figsize=(12,8))
corr = data.corr()
sns.heatmap(abs(corr), annot=True)
plt.savefig('correlation_heatmap.png', bbox_inches='tight')
# Identify top 5 features with highest absolute correlation with 'Selector'
correlation_with_selector = corr['Selector'].abs().sort_values(ascending=False)
top_5_features = correlation_with_selector[1:6]
# Exclude 'Selector' itself
print(top_5_features)
```

Les résultats requis seraient les suivants :

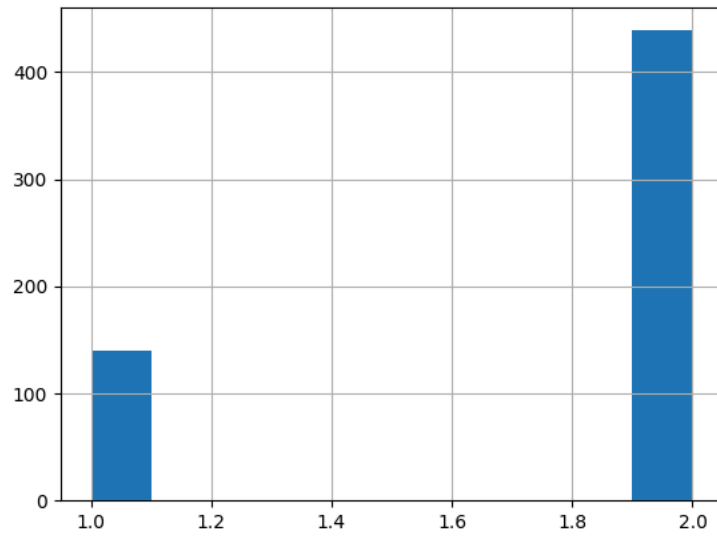
1. Sortie du terminal

```
theia@theia-abhishek1:/home/project$ python3 test_file.py
Direct_Bilirubin      0.246273
Total_Bilirubin       0.220218
Alkaline_Phosphotase  0.183363
Albumin and Globulin Ratio  0.163131
Alamine_Aminotransferase 0.163117
Name: Selector, dtype: float64
```

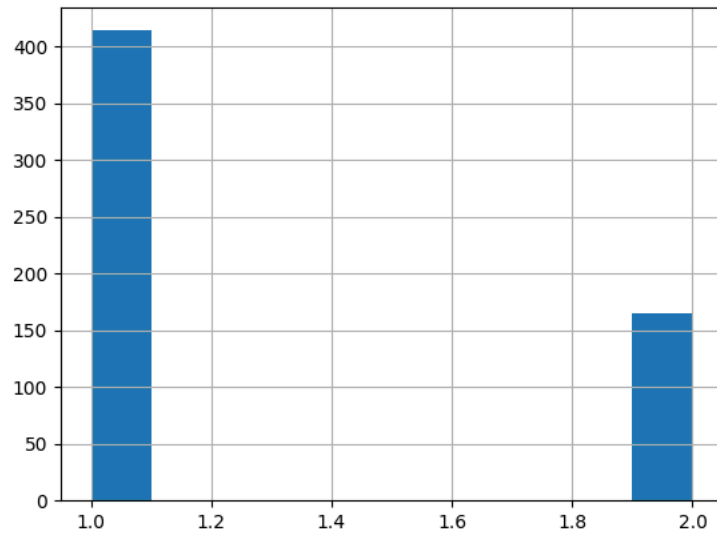
2. Histogramme de l'âge



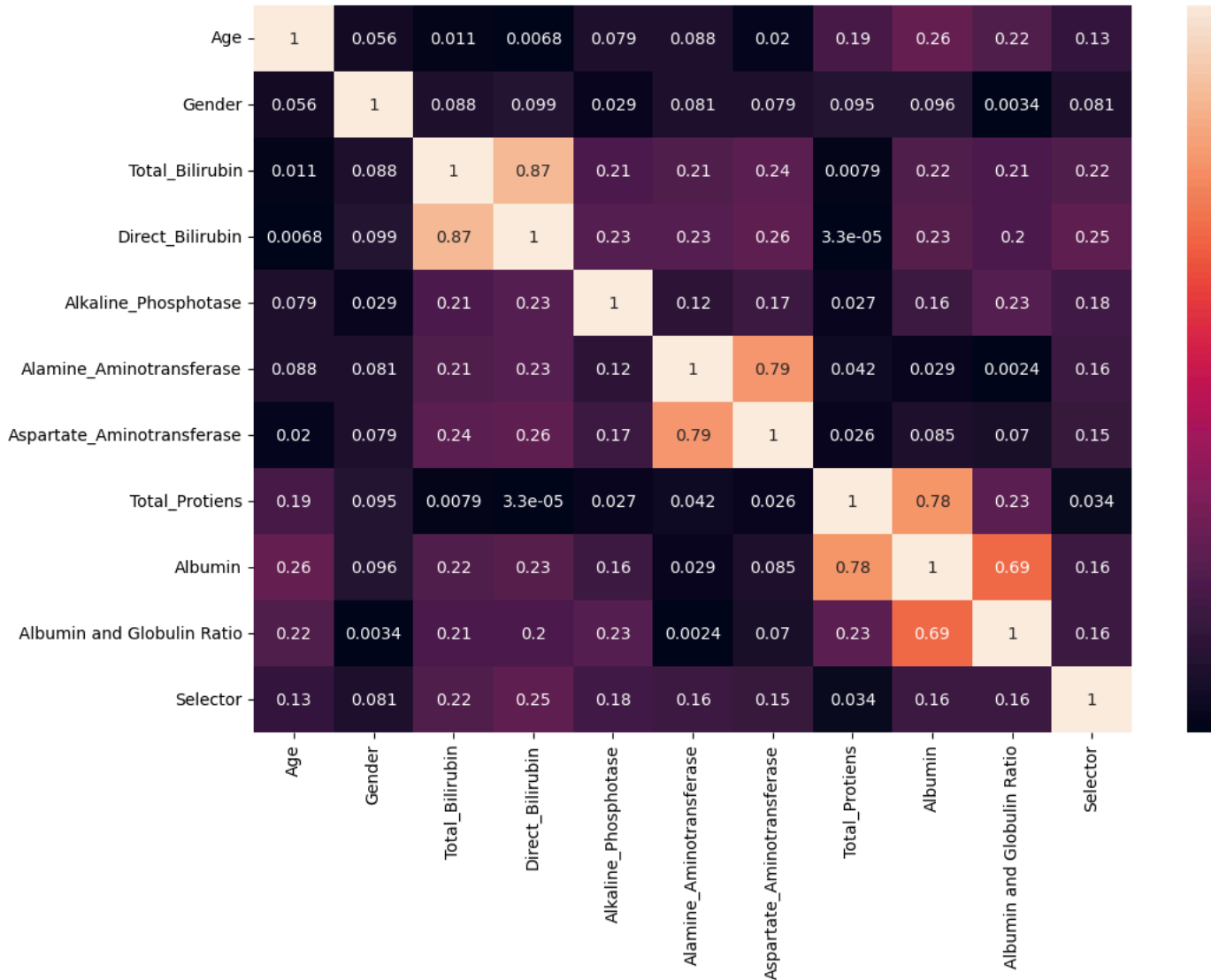
3. Histogramme de genre



4. Histogramme du sélecteur



5. Carte thermique de corrélation



Stratégies d'exploration de données

L'exploration de données consiste à identifier des modèles dans la distribution des données. De la création de classificateurs à l'identification d'associations entre différents attributs, les stratégies d'exploration de données peuvent être très vastes. Dans cette tâche, vous utiliserez l'IA générative pour créer un classificateur qui prendra les 5 meilleures caractéristiques identifiées et créera un classificateur de détection des maladies du foie. Pour illustrer davantage la puissance des outils d'IA générative, vous pouvez inclure les aspects suivants dans le code.

1. Effectuer une mise à l'échelle standard sur les attributs.
2. Comparez la précision de classification de plusieurs classificateurs sur les données pour identifier le meilleur. Ceux qui devraient être utilisés sont :
 - a. Régression logistique
 - b. KNN
 - c. Bayes naïf
 - d. Arbres de décision
 - e. Forêts aléatoires
 - f. Perceptron multicouche

Vous pouvez créer ce code simplement en utilisant l'invite suivante.

Write a python code that can perform the following tasks:

1. Perform standard scaling operation on the top 5 attributes identified in the previous code.
2. Train and test the following classifiers on the data and print a comparative table for their accuracy performance:
 - a. Logistic Regression
 - b. KNN
 - c. Naive Bayes
 - d. Decision Trees
 - e. Random Forests
 - f. Multi layer perceptron

Vous pouvez vous attendre à ce qu'un code soit généré, similaire à celui illustré ci-dessous.

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
```

```

import pandas as pd
# Standard scaling on the top 5 attributes
scaler = StandardScaler()
top_5_attributes = data[top_5_features.index]
scaled_data = scaler.fit_transform(top_5_attributes)
scaled_data = pd.DataFrame(scaled_data, columns=top_5_attributes.columns)
# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(scaled_data, data['Selector'], test_size=0.2, random_state=42)
# Train and test classifiers
classifiers = {
    'Logistic Regression': LogisticRegression(),
    'KNN': KNeighborsClassifier(),
    'Naive Bayes': GaussianNB(),
    'Decision Trees': DecisionTreeClassifier(),
    'Random Forests': RandomForestClassifier(),
    'Multi layer perceptron': MLPClassifier()
}
results = {}
for name, clf in classifiers.items():
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    results[name] = accuracy
# Create DataFrame for the results
results_df = pd.DataFrame(list(results.items()), columns=['Classifier', 'Accuracy'])
print(results_df)

```

Vous pouvez utiliser le code ci-dessus et modifier votre fichier existant pour y ajouter les aspects d'exploration de données. Le code résultant devrait ressembler à celui indiqué ci-dessous.

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
# Load the CSV data into a pandas data frame
data = pd.read_csv('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-AI0273EN-SkillsNetwork/labs/')
# Save histograms of data distribution for 'Age', 'Gender', and 'Selector'
plt.figure()
data['Age'].hist()
plt.savefig('age_histogram.png')
plt.figure()
data['Gender'].hist()
plt.savefig('gender_histogram.png')
plt.figure()
data['Selector'].hist()
plt.savefig('selector_histogram.png')
# Save correlation heatmap of the data set
plt.figure(figsize=(12,8))
corr = data.corr()
sns.heatmap(abs(corr), annot=True)
plt.savefig('correlation_heatmap.png', bbox_inches='tight')
# Identify top 5 features with highest absolute correlation with 'Selector'
correlation_with_selector = corr['Selector'].abs().sort_values(ascending=False)
top_5_features = correlation_with_selector[1:6] # Exclude 'Selector' itself
#print(top_5_features)
# Standard scaling on the top 5 attributes
scaler = StandardScaler()
top_5_attributes = data[top_5_features.index]
scaled_data = scaler.fit_transform(top_5_attributes)
scaled_data = pd.DataFrame(scaled_data, columns=top_5_attributes.columns)
# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(scaled_data, data['Selector'], test_size=0.2, random_state=42)
# Train and test classifiers
classifiers = {
    'Logistic Regression': LogisticRegression(),
    'KNN': KNeighborsClassifier(),
    'Naive Bayes': GaussianNB(),
    'Decision Trees': DecisionTreeClassifier(),
    'Random Forests': RandomForestClassifier(),
    'Multi layer perceptron': MLPClassifier()
}
results = {}
for name, clf in classifiers.items():
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    results[name] = accuracy
# Create DataFrame for the results
results_df = pd.DataFrame(list(results.items()), columns=['Classifier', 'Accuracy'])
print(results_df)

```

En tant que résultat final, vous pouvez vous attendre à ce que la trame de données suivante soit imprimée.

	Classifier	Accuracy
0	Logistic Regression	0.620690
1	KNN	0.646552
2	Naive Bayes	0.577586
3	Decision Trees	0.603448
4	Random Forests	0.620690
5	Multi layer perceptron	0.620690

Conclusion

Félicitations pour avoir terminé ce laboratoire !

Vous savez désormais comment utiliser l'IA générative pour :

1. Effectuer une analyse exploratoire des données
2. Mettre en œuvre des stratégies d'exploration de données

Auteur(s)

[Abhishek Gagneja](#)

© IBM Corporation. Tous droits réservés.