```python
# -*- coding: utf-8 -*-
"""retail.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1ZiKpChcS6oS_QfRlCWDe28ZxGadlb4rX
"""

import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt

from sklearn.cluster import KMeans

df=pd.read_csv("/content/store.csv",sep=",", header=0)

df.head(10)

# Pre-processing
df.dropna(inplace=True)
df = pd.get_dummies(df, columns=['region'])

df.head(2)

# Split the data into features and target
X = df.drop('revenue', axis=1)

df = df.dropna() # Remove missing values

df

df = df[['qty', 'revenue']]

df

# Scale the features
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df)

# Fit the k-means model
kmeans = KMeans(n_clusters=5)
kmeans.fit(df_scaled)

# Make predictions on the data
y_kmeans = kmeans.predict(df_scaled)

# Add the cluster labels to the data
df['Cluster'] = y_kmeans

df



# Group the data by cluster
clustered_data = df.groupby('Cluster').agg({
    'revenue': 'mean'
})

print(clustered_data)



"""*metrics that can be used to evaluate the performance of a K-Means clustering algorithm*"""

from sklearn.metrics import silhouette_score, calinski_harabasz_score

# Generate some sample data
np.random.seed(0)
X = np.random.rand(100, 2)

# Train the K-Means model with different number of clusters
for n_clusters in range(2, 11):
    kmeans = KMeans(n_clusters=n_clusters, random_state=0).fit(X)

# Evaluate the performance with Silhouette Score
silhouette = silhouette_score(X, kmeans.labels_)
print(f"Silhouette Score for {n_clusters} clusters: {silhouette:.3f}")

# Evaluate the performance with Calinski-Harabasz Index
calinski = calinski_harabasz_score(X, kmeans.labels_)
print(f"Calinski-Harabasz Index for {n_clusters} clusters: {calinski:.3f}")

# Plot the clusters
plt.scatter(X[:, 0], X[:, 1], c=kmeans.labels_)
plt.title(f"{n_clusters} Clusters")
plt.show()
```