5. Categorical data clustering

(a) One- hot encoding for categorical data clustering.

```
 import pandas as pd
from sklearn.preprocessing import OneHotEncoder
from sklearn.cluster import KMeans

data = pd.DataFrame({
   "weather": ["sunny", "cloudy", "rainy", "sunny", "rainy", "cloudy", "sunny"]
})

print("Original Data:")
print(data)

encoder = OneHotEncoder(sparse_output=False)
onehot = encoder.fit_transform(data[["weather"]])

print("\nOne-Hot Encoded Data:")
print(onehot)

print("\nCategories:", encoder.categories_)

kmeans = KMeans(n_clusters=2, random_state=0)
labels = kmeans.fit_predict(onehot)

print("\nCluster Labels:")
print(labels)

data["cluster"] = labels
print("\nFinal Data with Cluster Assignments:")
print(data)
```

output:

Original Data:

```
  weather
0   sunny
1  cloudy
2   rainy
3   sunny
4   rainy
5  cloudy
6   sunny
```

One-Hot Encoded Data:

```
[[0. 0. 1.]
 [1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]
 [0. 1. 0.]
 [1. 0. 0.]
 [0. 0. 1.]]
```

Categories: [array(['cloudy', 'rainy', 'sunny'], dtype=object)]

Cluster Labels:

[0 0 1 0 1 0 0]

Final Data with Cluster Assignments:

```
  weather  cluster
0   sunny        0
1  cloudy        0
2   rainy        1
3   sunny        0
```

4	rainy	1
5	cloudy	0
6	sunny	0

(b) Dummy encoding for categorical data clustering.

```
import pandas as pd
from sklearn.cluster import KMeans

data = pd.DataFrame({
   "weather": ["sunny", "cloudy", "rainy", "sunny", "rainy", "cloudy", "sunny"]
})

print("Original Data:")
print(data)

dummy_encoded = pd.get_dummies(data, columns=["weather"], drop_first=True)

print("\nDummy Encoded Data (drop_first=True):")
print(dummy_encoded)

kmeans = KMeans(n_clusters=2, random_state=0)
labels = kmeans.fit_predict(dummy_encoded)

print("\nCluster Labels:")
print(labels)

data["cluster"] = labels

print("\nFinal Data with Cluster Assignments:")
print(data)
```

output:

```
Original Data:
  weather
0   sunny
1  cloudy
```

2 rainy

3 sunny

4 rainy

5 cloudy

6 sunny

Dummy Encoded Data (drop_first=True):

  weather_rainy weather_sunny

0   False   True

1   False   False

2   True   False

3   False   True

4   True   False

5   False   False

6   False   True

Cluster Labels:

[0 1 1 0 1 1 0]

Final Data with Cluster Assignments:

 weather cluster

0 sunny   0

1 cloudy   1

2 rainy   1

3 sunny   0

4 rainy   1

5 cloudy   1

6 sunny   0

(c) Effective encoding for categorical data clustering.

```
import pandas as pd
from sklearn.cluster import KMeans
import numpy as np

data = pd.DataFrame({
   "weather": ["sunny", "cloudy", "rainy", "sunny", "rainy", "sunny", "cloudy"]
})

print("Original Data:")
print(data)

freq_map = data["weather"].value_counts(normalize=True).to_dict()
data["weather_freq"] = data["weather"].map(freq_map)

print("\nFrequency Encoded Data:")
print(data)

kmeans = KMeans(n_clusters=2, random_state=0)
labels = kmeans.fit_predict(data[["weather_freq"]])

data["cluster"] = labels

print("\nFinal Data With Cluster Assignments:")
print(data)
```

output:

Original Data:

	weather
0	sunny
1	cloudy
2	rainy
3	sunny
4	rainy
5	sunny
6	cloudy

Frequency Encoded Data:

	weather	weather_freq
0	sunny	0.428571
1	cloudy	0.285714
2	rainy	0.285714
3	sunny	0.428571
4	rainy	0.285714
5	sunny	0.428571
6	cloudy	0.285714

Final Data With Cluster Assignments:

	weather	weather_freq	cluster
0	sunny	0.428571	0
1	cloudy	0.285714	1
2	rainy	0.285714	1
3	sunny	0.428571	0
4	rainy	0.285714	1
5	sunny	0.428571	0
6	cloudy	0.285714	1