```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

df = pd.read_csv('data2.csv', index_col = 'customer_name')

scaler = StandardScaler()

df[['1st_question_T','2nd_question_T','3rd_question_T','4th_question_T
','5th_question_T','6th_question_T','7th_question_T','8th_question_T',
'9th_question_T','10th_question_T','11th_question_T','12th_question_T'
]] =
scaler.fit_transform(df[['1st_question','2nd_question','3rd_question',
'4th_question','5th_question','6th_question','7th_question','8th_quest
ion','9th_question','10th_question','11th_question','12th_question']])

df

               1st_question  2nd_question  3rd_question  4th_question
\
customer_name

Name_1                0.162         0.108         0.587         0.953

Name_2                0.799         0.112         0.073         0.955

Name_3                0.218         0.172         0.340         0.908

Name_4                0.644         0.736         0.574         0.723

Name_5                0.054         0.226         0.842         0.533

...                     ...           ...           ...           ...

Name_213              0.795         0.879         0.841         0.220

Name_214              0.016         0.810         0.634         0.748

Name_215              0.928         0.940         0.451         0.106

Name_216              0.283         0.475         0.341         0.658

Name_217              0.937         0.010         0.060         0.935


               5th_question  6th_question  7th_question  8th_question
\
customer_name
```

Name_1	0.780	0.148	0.639	0.266
Name_2	0.891	0.211	0.157	0.858
Name_3	0.814	0.511	0.797	0.270
Name_4	0.769	0.023	0.966	0.234
Name_5	0.480	0.602	0.418	0.400
...	...	...	...	...
Name_213	0.848	0.630	0.008	0.394
Name_214	0.323	0.029	0.107	0.888
Name_215	0.408	0.695	0.468	0.513
Name_216	0.217	0.834	0.390	0.620
Name_217	0.883	0.852	0.961	0.527

customer_name	9th_question	10th_question	...	3rd_question_T
			...	
Name_1	0.935	0.406	...	0.201599
Name_2	0.317	0.514	...	-1.540686
Name_3	0.598	0.958	...	-0.635647
Name_4	0.395	0.949	...	0.157533
Name_5	0.509	0.596	...	1.065962
...	...	...	...	...
Name_213	0.470	0.691	...	1.062572
Name_214	0.429	0.567	...	0.360913
Name_215	0.606	0.461	...	-0.259395
Name_216	0.676	0.066	...	-0.632257
Name_217	0.301	0.593	...	-1.584751

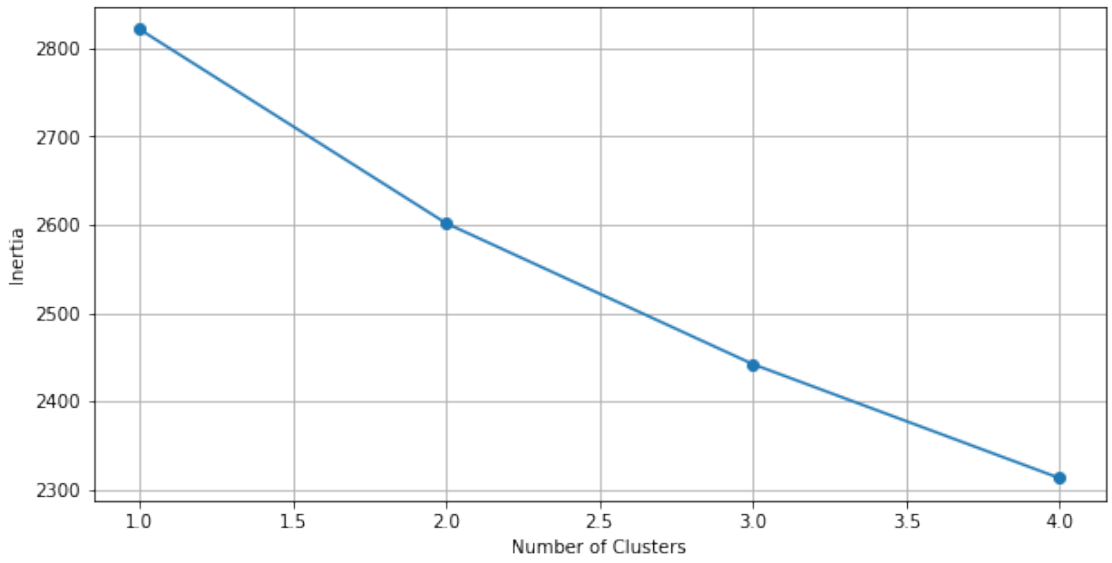
customer_name	4th_question_T	5th_question_T	6th_question_T	7th_question_T
Name_1	1.572849	0.928813	-1.204178	0.572117
Name_2	1.579901	1.322782	-0.988931	-1.091260
Name_3	1.414168	1.049488	0.036056	1.117373
Name_4	0.761814	0.889771	-1.631256	1.700590

```
Name_5               0.091828       -0.135968        0.346968       -
0.190552
...                       ...             ...             ...
...
Name_213            -1.011884        1.170164        0.442634       -
1.605457
Name_214             0.849970       -0.693204       -1.610757       -
1.263809
Name_215            -1.413876       -0.391516        0.664714       -
0.018002
Name_216             0.532608       -1.069427        1.139625       -
0.287180
Name_217             1.509377        1.294388        1.201124
1.683335

               8th_question_T  9th_question_T  10th_question_T  \
customer_name
Name_1               -0.894134        1.519849        -0.346281
Name_2                1.120459       -0.762528         0.034172
Name_3               -0.880522        0.275252         1.598255
Name_4               -1.003031       -0.474461         1.566550
Name_5               -0.438128       -0.053440         0.323034
...                        ...             ...              ...
Name_213             -0.458546       -0.197474         0.657691
Name_214              1.222550       -0.348894         0.220875
Name_215             -0.053586        0.304797        -0.152532
Name_216              0.310538        0.563319        -1.544002
Name_217             -0.005944       -0.821619         0.312466

               11th_question_T  12th_question_T
customer_name
Name_1                1.137741         1.407883
Name_2                1.036528        -0.418591
Name_3               -0.129166        -1.576595
Name_4               -0.520058         1.515919
Name_5               -0.935380        -1.644117
...                        ...              ...
Name_213             -0.879538         1.556432
Name_214             -1.078474        -1.360524
Name_215             -0.903969        -0.016834
Name_216              1.085390         1.239078
Name_217             -1.500777         1.158051

[217 rows x 24 columns]
```

```
# Create function to identify optimum number of clusters
def optimize_k_means(data, max_k):
    means = []
    inertias = []
```

```
    for k in range(1, max_k):
        kmeans = KMeans(n_clusters=k)
        kmeans.fit(df)

        means.append(k)
        inertias.append(kmeans.inertia_)

    #Generate the elbow
    fig = plt.subplots(figsize=(10,5))
    plt.plot(means, inertias, 'o-')
    plt.xlabel('Number of Clusters')
    plt.ylabel('Inertia')
    plt.grid(True)
    plt.show()

optimize_k_means(df[['2nd_question_T', '6th_question_T']], 5)
```

```
C:\Users\User\anaconda3\lib\site-packages\sklearn\cluster\
_kmeans.py:1039: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads.
You can avoid it by setting the environment variable
OMP_NUM_THREADS=1.
  warnings.warn(
```




```
kmeans = KMeans(n_clusters=3)

kmeans.fit(df[['6th_question_T', '2nd_question_T']])

KMeans(n_clusters=3)

df['kmeans_3'] = kmeans.labels_

df
```

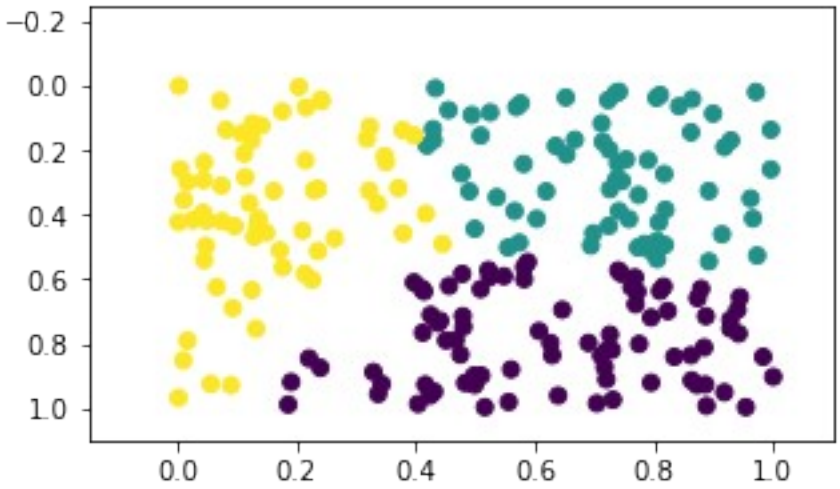
customer_name	1st_question	2nd_question	3rd_question	4th_question
Name_1	0.162	0.108	0.587	0.953
Name_2	0.799	0.112	0.073	0.955
Name_3	0.218	0.172	0.340	0.908
Name_4	0.644	0.736	0.574	0.723
Name_5	0.054	0.226	0.842	0.533
...	...	...	...	...
Name_213	0.795	0.879	0.841	0.220
Name_214	0.016	0.810	0.634	0.748
Name_215	0.928	0.940	0.451	0.106
Name_216	0.283	0.475	0.341	0.658
Name_217	0.937	0.010	0.060	0.935

customer_name	5th_question	6th_question	7th_question	8th_question
Name_1	0.780	0.148	0.639	0.266
Name_2	0.891	0.211	0.157	0.858
Name_3	0.814	0.511	0.797	0.270
Name_4	0.769	0.023	0.966	0.234
Name_5	0.480	0.602	0.418	0.400
...	...	...	...	...
Name_213	0.848	0.630	0.008	0.394
Name_214	0.323	0.029	0.107	0.888
Name_215	0.408	0.695	0.468	0.513

```
Name_216              0.217         0.834         0.390         0.620

Name_217              0.883         0.852         0.961         0.527


               9th_question  10th_question  ...  4th_question_T  \
customer_name                               ...
Name_1                0.935          0.406  ...        1.572849
Name_2                0.317          0.514  ...        1.579901
Name_3                0.598          0.958  ...        1.414168
Name_4                0.395          0.949  ...        0.761814
Name_5                0.509          0.596  ...        0.091828
...                     ...            ...  ...             ...
Name_213              0.470          0.691  ...       -1.011884
Name_214              0.429          0.567  ...        0.849970
Name_215              0.606          0.461  ...       -1.413876
Name_216              0.676          0.066  ...        0.532608
Name_217              0.301          0.593  ...        1.509377

               5th_question_T  6th_question_T  7th_question_T
8th_question_T  \
customer_name

Name_1               0.928813       -1.204178        0.572117       -
0.894134
Name_2               1.322782       -0.988931       -1.091260
1.120459
Name_3               1.049488        0.036056        1.117373       -
0.880522
Name_4               0.889771       -1.631256        1.700590       -
1.003031
Name_5              -0.135968        0.346968       -0.190552       -
0.438128
...                       ...             ...             ...
...
Name_213             1.170164        0.442634       -1.605457       -
0.458546
Name_214            -0.693204       -1.610757       -1.263809
1.222550
Name_215            -0.391516        0.664714       -0.018002       -
0.053586
Name_216            -1.069427        1.139625       -0.287180
0.310538
Name_217             1.294388        1.201124        1.683335       -
0.005944

               9th_question_T  10th_question_T  11th_question_T  \
customer_name
```

```
Name_1                1.519849        -0.346281         1.137741
Name_2               -0.762528         0.034172         1.036528
Name_3                0.275252         1.598255        -0.129166
Name_4               -0.474461         1.566550        -0.520058
Name_5               -0.053440         0.323034        -0.935380
...                        ...              ...              ...
Name_213             -0.197474         0.657691        -0.879538
Name_214             -0.348894         0.220875        -1.078474
Name_215              0.304797        -0.152532        -0.903969
Name_216              0.563319        -1.544002         1.085390
Name_217             -0.821619         0.312466        -1.500777

               12th_question_T  kmeans_3
customer_name
Name_1                1.407883         2
Name_2               -0.418591         2
Name_3               -1.576595         2
Name_4                1.515919         1
Name_5               -1.644117         2
...                        ...       ...
Name_213              1.556432         0
Name_214             -1.360524         1
Name_215             -0.016834         0
Name_216              1.239078         0
Name_217              1.158051         2

[217 rows x 25 columns]
```

```
plt.figure(figsize=(5, 3))
plt.scatter(x=df['2nd_question'], y=df['6th_question'],
c=df['kmeans_3'])
plt.xlim(-0.15, 1.1)
plt.ylim(1.1, -0.25)
plt.show()
```



```
df.to_csv('cust_classification_data.csv')
```