← Go Back to Capstone Project

Course Content

FAQs - Marketing Campaign Customer Segmentation

1. Getting an error when importing the SilhouetteVisualizer from the yellow brick library. How to install the Yellowbrick library?

```
# To perform K-means clustering and compute Silhouette scores
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# To visualize the elbow curve and Silhouette scores
from yellowbrick.cluster import SilhouetteVisualizer

# Importing PCA
from sklearn.decomposition import PCA

# To encode the variable
from sklearn.preprocessing import LabelEncoder

# Importing TSNE
from sklearn.manifold import TSNE

# To perform hierarchical clustering, compute cophenetic correlation, and create dendrograms
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage, cophenet

# To compute distances
from scipy.spatial.distance import pdist

# To import K-Medoids
from sklearn_extra.cluster import KMedoids

# To import Gaussian Mixture
from sklearn.mixture import GaussianMixture

# To supress warnings
import warnings

warnings.filterwarnings("ignore")
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
T:\Temp/ipykernel_22428/2173424247.py in <module>
     18
     19 # To visualize the elbow curve and Silhouette scores
---> 20 from yellowbrick.cluster import SilhouetteVisualizer
     21
     22 # Importing PCA

ModuleNotFoundError: No module named 'yellowbrick'
```

We have listed three possible solutions to resolve the above error. Please try the same:

Solution 1:
Run the below command in your **jupyter notebook**:

```
!pip install yellowbrick
```

After running the above command successfully, please restart the notebook and import the library, it will be imported now.

Solution 2:

Run the command in your anaconda prompt (for Windows) or Terminal (for Mac):

```
conda install yellowbrick
```

Note: If there is an issue either with the internet or firewall during the installation, please run the below 2 commands step-by-step:

.

```
conda config --set ssl_verify no
```

After running the above command successfully, please run the below command:
2.

```
conda install -c districtdatalabs yellowbrick
```

Solution 3:

As an alternate and quick solution, please switch to Google Colab. The code runs smoothly without any installation errors.

2. What was your criteria to indicate random_state = 0 and not some other value of random_state. Is this decision also random, or do you have a preference, and why?

The random state values make difference in the randomness of the environment, let's say random_state = 10, it will generate the results with the randomness of 10, likewise, it applies to other random state values. We cannot interpret what change it will make on the results as it is more like an experiment. As a good practice, please stick to the one fixed random state and try to optimize the model in that random state.

If we change the random state then the results will also vary since the results depend upon the randomness of the environment as well. The random_state is used to get reproducible results, there is no straightforward rule here to choose the value of the random_state, please stick to the constant random state throughout the notebook and work on model improvement.

3. For this code data_model["DBSCAN"].hist(bins=5, grid = False), I'm getting the below error, how to resolve this?

```
TypeError: '<=' not supported between instances of 'method' and 'method'
```

The error that you are getting is due to the fact that DBSCAN and hist() both are methods in Python. It is not possible to compare the two methods. You must take labels from DBSCAN to plot the histogram in the notebook. The below code will help to resolve the error:

```
data_model["DBSLabels"].hist(bins=3, grid = False) data_model["DBSLabels"].value_counts()
```

4. Getting the below error while calculating the silhouette_score for each of the combinations using the DBSCAN algorithm. How do resolve the error?

ValueError: The number of labels is 1. Valid values are 2 to n_samples - 1 (inclusive)

You might receive this error when the DBSCAN model predicts all the data points into a single cluster, however, to calculate the silhouette_score, we need at least 2 labels. To resolve this error, please try to alter the hyperparameter values that are going to play a crucial role through the **min_samples** parameter of the DBSCAN model.

5. Can you help me to fix the below error while implementing the Agglomerative clustering algorithm?

```
In [73]: # Add Agglomerative Clustering cluster labels to data_pca

         data_pca["HClabels"]=HCmodel.label_

         # Add Agglomerative Clustering cluster labels to the whole data

         data["HClabels"]=HCmodel.label_

         # Add Agglomerative Clustering cluster labels to data_model

         data_model["HClabels"]=HCmodel.label_
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_11176/3216948708.py in <module>
      1 # Add Agglomerative Clustering cluster labels to data_pca
      2
----> 3 data_pca["HClabels"]=HCmodel.label_
      4
```

You are getting this attribute error because there is no attribute called the '**label_**' for the Agglomerative Clustering algorithm. To fix the error, please use '**labels_**' instead of '**label_**' in the code. So, the correct syntax is **HCmodel.labels_**

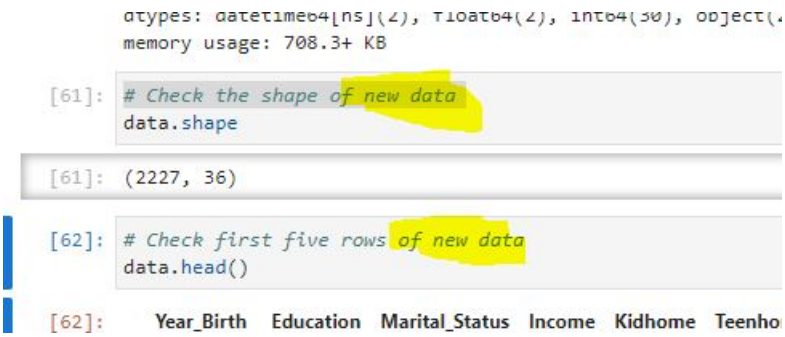
6. How to resolve the below error occurred while finding predictions using the K-Medoids algorithm?

```
In [116]: # Predicting on data_pca and ddding K-Medoids cluster labels to the whole data
          data['K_medoids_segments_5']= kmedo.predict(data_pca)
          # Predicting on data_pca and ddding K-Medoids cluster labels to data_model
          data_model['K_medoids_segments_5']= kmedo.predict(data_pca)
          # Predicting on data_pca and ddding K-Medoids cluster labels to data_pca
          data_pca['K_medoids_segments_5']= kmedo.predict(data_pca)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
C:\Users\MIKEYJ~1\AppData\Local\Temp/ipykernel_15880/3027786728.py in <module>
      1 # Predicting on data_pca and ddding K-Medoids cluster labels to the whole data
----> 2 data['K_medoids_segments_5']= kmedo.predict(data_pca)
      3 # Predicting on data_pca and ddding K-Medoids cluster labels to data_model
      4 data_model['K_medoids_segments_5']= kmedo.predict(data_pca)
      5 # Predicting on data_pca and ddding K-Medoids cluster labels to data_pca

~\anaconda3\lib\site-packages\sklearn_extra\cluster\_k_medoids.py in predict(self, X)
    351             # Return data points to clusters based on which cluster assignment
    352             # yields the smallest distance
--> 353             return pairwise_distances_argmin(
    354                 X, Y=self.cluster_centers_, metric=self.metric
    355             )

~\anaconda3\lib\site-packages\sklearn\metrics\pairwise.py in pairwise_distances_argmin(X, Y, axis, metric, metric_kwargs)
    789             metric_kwargs = {}
    790
--> 791         indices = PairwiseDistancesArgKmin.compute(
    792             X=X,
    793             Y=Y,

sklearn\metrics\_pairwise_distances_reduction.pyx in sklearn.metrics._pairwise_distances_reduction.PairwiseDistancesArgKmin.com
pute()

sklearn\metrics\_pairwise_distances_reduction.pyx in sklearn.metrics._pairwise_distances_reduction.FastEuclideanPairwiseDistanc
esArgKmin.__init__()

sklearn\metrics\_dist_metrics.pyx in sklearn.metrics._dist_metrics.DatasetsPair.get_for()

sklearn\metrics\_dist_metrics.pyx in sklearn.metrics._dist_metrics.DenseDenseDatasetsPair.__init__()

~\anaconda3\lib\site-packages\sklearn\metrics\_dist_metrics.cp39-win_amd64.pyd in View.MemoryView.memoryview_cwrapper()

~\anaconda3\lib\site-packages\sklearn\metrics\_dist_metrics.cp39-win_amd64.pyd in View.MemoryView.memoryview.__cinit__()

ValueError: ndarray is not C-contiguous
```

To resolve this error, please use the **fit_predict()** method instead of **predict()** method to find predictions using the K-Medoids algorithm.

7. In the Low Code reference notebook, in the 'Preparing Data for Segmentation' section, all the irrelevant columns are dropped and stored in a dataframe data_model from data. There are many references in the next bunch of lines that state to use new data. What is new data in this case? Is it data df or data_model df? Please clarify.



The new_data means newly created dataFrame at the start of the 'Preparing Data for Segmentation' section. So here, the new data means **data_model,** not data.

8. DBSCAN Error ValueError: The number of labels is 1. Valid values are 2 to n_samples - 1 (inclusive)

This error message is indicating that there is an issue with the number of labels in a classification problem. In sci-kit learn (a popular machine learning library in Python), the number of labels in a classification problem should be greater than 1 and less than the number of samples (observations).

The error message states that the number of labels is 1, which is not a valid value. You should ensure that you have at least 2 unique labels in your data and that the number of labels is less than the number of samples. Re-examine your data and pre-processing steps to resolve the error.

9. I got different silhouette scores for the same data. Why is that?

The random_state argument in the KMeans function sets the random seed for the algorithm, which determines the initial centroids and the way the algorithm splits the data into clusters. Since the final clusters depend on the initial centroids, different random seeds can result in different silhouette scores.

10. ValueError: Input X contains NaN.

There could be several reasons why the error message is appearing even though you can't see any NaN values in your dataset. Some possible causes include:

Hidden NaN values: There could be NaN values that are not immediately visible, such as those encoded as blanks, zeros, or other values that are not recognized as NaN.

Previous

sues: Make sure that the data type of each column is compatible with PCA and does not contain any NaN values.

Next

Preprocessing steps: Check if any preprocessing steps, such as normalization or scaling, introduced NaN values into the dataset.