

[← Go Back to Unsupervised Learning](#)

## Course Content

# Hands-on Quiz

Type	:	Practice Quiz
Attempts	:	1/1
Questions	:	10
Time	:	2h
Your Marks	:	20/20

## Instructions

Please download and import the dataset [Nutrient Composition Dataset.csv](#) into Jupyter Notebook to attempt this quiz.

### Problem Statment:

HealthifyUs is a US-based startup company that gives nutrition advice to its customers to help them stay healthy and fit. They have collected data about various food items sold in the market along with their nutrient composition. The data contains information about the amount of the following nutrients in food items - Protein, Fat, Vitamin C, and Fibre. The food items can be segmented based on their nutrient composition so that suggestions can be provided based on the customer's nutrition requirements.

### Attribute information:

1. Protein: protein content in the food products
2. Fat: fat content in the food products
3. vitaminC: vitamin C content in the food products
4. Fibre: Fibre content in the food products

## 5. Product: Name of the food products

Kindly go through these guidelines before you attempt the quiz.

1. Use random\_state=1 wherever this parameter can be used.
2. Ensure there is a proper internet connection while taking up the quiz. Any breakup in the connection will automatically submit your quiz.
3. Only attempt the quiz when you are prepared and have enough time on your hands to finish it. Please ensure you attempt the quiz well before the due date. No extension will be provided for any quiz once the deadline is passed.
4. The quiz once opened, must be completed within the time frame provided. You CANNOT start the quiz, leave it unattended for an extended period of time and come back later to finish.
5. No re-attempts will be provided if the quiz gets submitted for any of the above-mentioned reasons.
6. If you face any other technical issues on Olympus, you should share the screenshot with your Program Manager so that the team can understand and resolve it on priority.

## Attempt History

Attempt #1

Feb 25, 12:14 PM

Marks: 20



Q No: 1

Correct Answer

Marks: 2/2

**Import the dataset and drop the S.No column from the data. What is the Interquartile Range (IQR) of the variable Protein?**

☐ 0.435☒ 0.859

You Selected

☐ 1.295☐ 0.224

# importing the data

df = pd.read\_csv("Nutrient Composition Dataset.csv")

# dropping S.No column

df.drop("S.No", axis=1, inplace=True)

# IQR of Protein column

df.Protein.quantile(0.75) - df.Protein.quantile(0.25)

Q No: 2

Correct Answer

Marks: 2/2

The variable Fat has a left skewed distribution.

☐ True☒ False

You Selected

sns.histplot(data=df, x="Fat")

Q No: 3

Correct Answer

Marks: 2/2

Which of the following pair of variables have the highest correlation by magnitude?

☐ Fat and Fibre☒ Fat and Protein

You Selected

☐ Fibre and VitaminC☐ Protein and Fibre

```
df.corr()
```

(or)

```
plt.figure(figsize=(15, 7))
```

```
sns.heatmap(df.corr(), annot=True, vmin=-1, vmax=1, fmt=".2f", cmap="Spectral")
```

```
plt.show()
```

Q No: 4

Correct Answer

Marks: 2/2

Scale the numerical variables using z-score scaling. Apply the K-means algorithm on the dataset, draw the elbow curve for different values of k (ranging from 1 to 15), and find the silhouette coefficients for each k.

What is the appropriate value for k from the elbow curve?

**Note:** Do not treat outliers in the data.

☒ 2 or 3

You Selected

☐ 5 or 6☐ 8 or 9☐ 11 or 12

```
# scaling the data before clustering
scaler = StandardScaler()
subset_scaled_df = df.iloc[:, :-1].copy()
subset_scaled_df.iloc[:, :] = scaler.fit_transform(subset_scaled_df.iloc[:, :])
subset_scaled_df.head()

# creating a copy of the scaled dataframe
k_means_df = subset_scaled_df.copy()

# elbow plot

clusters = range(1, 16)
meanDistortions = []

for k in clusters:
    model = KMeans(n_clusters=k, random_state=1)
    model.fit(subset_scaled_df)
    prediction = model.predict(k_means_df)
    distortion = (
        sum(np.min(cdist(k_means_df, model.cluster_centers_, "euclidean"), axis=1))
        / k_means_df.shape[0]
    )

    meanDistortions.append(distortion)

print("Number of Clusters:", k, "\tAverage Distortion:", distortion)
```

```
plt.plot(clusters, meanDistortions, "bx-")  
plt.xlabel("k")  
plt.ylabel("Average Distortion")  
plt.title("Selecting k with the Elbow Method", fontsize=20)  
plt.show()
```

**Q No: 5****Correct Answer**

Marks: 2/2

For which of the following values of k is the silhouette score highest?

☒ 3**You Selected**☐ 4☐ 6☐ 9

```
sil_score = []  
cluster_list = range(2, 16)  
for n_clusters in cluster_list:  
    clusterer = KMeans(n_clusters=n_clusters, random_state=1)  
    preds = clusterer.fit_predict((subset_scaled_df))  
    score = silhouette_score(k_means_df, preds)  
    sil_score.append(score)  
    print("For n_clusters = {}, the silhouette score is {}".format(n_clusters, score))  
  
plt.plot(cluster_list, sil_score)  
plt.show()
```

**Q No: 6****Correct Answer**

Marks: 2/2

When K-means clustering is applied on the scaled data with random state=1 for 3 clusters, how many observations are there in the 2nd cluster?

☐ less than 100

☐ between 200 and 300

☒ between 400 and 500

You Selected

☐ greater than 600

# fitting K-means model

```
kmeans = KMeans(n_clusters=3, random_state=1)
kmeans.fit(k_means_df)
```

# adding kmeans cluster labels to the original and scaled dataframes

```
k_means_df["KM_segments"] = kmeans.labels_
```

```
df2 = df.copy()
df2["KM_segments"] = kmeans.labels_
```

# cluster profiling

```
km_cluster_profile = df2.groupby("KM_segments").mean()
```

```
km_cluster_profile["count_in_each_segment"] = (
df2.groupby("KM_segments")["Protein"].count().values
)
```

```
km_cluster_profile.style.highlight_max(color="lightgreen", axis=0)
```

Q No: 7

Correct Answer

Marks: 2/2

On applying Hierarchical clustering with Euclidean distance on the scaled dataset, which of the following linkage methods gives the highest cophenetic correlation?

☒ Single

You Selected

☐ Complete

☐ Ward

☐ Centroid

```
# creating a copy of the scaled dataset
hc_df = subset_scaled_df.copy()

# list of distance metrics
distance_metrics = ["euclidean"]

# list of linkage methods
linkage_methods = ["single", "complete", "average", "weighted", "ward", "centroid"]

high_cophenet_corr = 0
high_dm_lm = [0, 0]

for dm in distance_metrics:
    for lm in linkage_methods:
        Z = linkage(hc_df, metric=dm, method=lm)
        c, coph_dists = cophenet(Z, pdist(hc_df))
        print(
            "Cophenetic correlation for {} distance and {} linkage is {}".format(
                dm.capitalize(), lm, c
            )
        )
        if high_cophenet_corr < c:
            high_cophenet_corr = c
            high_dm_lm[0] = dm
            high_dm_lm[1] = lm
```



```
# printing the combination of distance metric and linkage method with the highest
cophenetic correlation
print("*" * 120)
print(
    "Highest cophenetic correlation is {}, which is obtained with {} linkage.".format(
        high_cophenet_corr, high_dm_lm[1]
    )
)
```

**Q No: 8****Correct Answer**

Marks: 2/2

On applying Hierarchical clustering on the scaled dataset with Euclidean distance and complete linkage method, we get 2 clusters for dendrogram height between 9 to 10.

☒ True
**You Selected**
☐ False

```
# list of linkage methods
linkage_methods = ["complete"]

# lists to save results of cophenetic correlation calculation
compare_cols = ["Linkage", "Cophenetic Coefficient"]
compare = []

# to create a subplot image
fig, axs = plt.subplots(len(linkage_methods), 1, figsize=(15, 7))

# We will enumerate through the list of linkage methods above
# For each linkage method, we will plot the dendrogram and calculate the cophenetic
correlation
for i, method in enumerate(linkage_methods):
    Z = linkage(hc_df, metric="euclidean", method=method)

    dendrogram(Z, ax=axs)
    axs.set_title(f"Dendrogram ({method.capitalize()} Linkage)")

    coph_corr, coph_dist = cophenet(Z, pdist(hc_df))
    axs.annotate(
```

```
f"Cophenetic\nCorrelation\n{coph_corr:0.2f}",  
(0.80, 0.80),  
xycoords="axes fraction",  
)  
  
compare.append([method, coph_corr])
```

Q No: 9

Correct Answer

Marks: 2/2

Based on dendrogram height, what is the appropriate number of clusters from the dendrogram obtained with Euclidean distance and Ward linkage?

☐ 9☒ 3

You Selected

☐ 5☐ 8

```
# list of linkage methods  
linkage_methods = ["ward"]  
  
# lists to save results of cophenetic correlation calculation  
compare_cols = ["Linkage", "Cophenetic Coefficient"]  
compare = []  
  
# to create a subplot image  
fig, axs = plt.subplots(len(linkage_methods), 1, figsize=(15, 7))  
  
# We will enumerate through the list of linkage methods above  
# For each linkage method, we will plot the dendrogram and calculate the cophenetic  
# correlation  
for i, method in enumerate(linkage_methods):  
    Z = linkage(hc_df, metric="euclidean", method=method)  
  
    dendrogram(Z, ax=axs)  
    axs.set_title(f"Dendrogram ({method.capitalize()} Linkage)")
```

```
coph_corr, coph_dist = cophenet(Z, pdist(hc_df))
axs.annotate(
    f"Cophenetic\nCorrelation\n{coph_corr:0.2f}",
    (0.80, 0.80),
    xycoords="axes fraction",
)

compare.append([method, coph_corr])
```

Q No: 10

Correct Answer

Marks: 2/2

For 3 clusters with Euclidean distance and Ward linkage, which of the following is true for the cluster having food items with high Protein?

☒ Fat is high

You Selected

☐ Vitamin C is high☐ Fibre is high☐ Has the lowest number of food items

# model fitting

```
HCmodel = AgglomerativeClustering(n_clusters=3, affinity="euclidean",
linkage="ward")
HCmodel.fit(hc_df)
```

&lt; Previous

Next &gt;

Proprietary content.©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

© 2024 All rights reserved

[Privacy](#) [Terms of service](#) [Help](#)