Topic	**Clustering**	
Class Description	**Students learn about Clustering or Cluster analysis using the K- means algorithm.**	
Class	**C118**	
Class time	**45 mins**	
Goal	● Explore clustering or cluster analysis in machine learning. ● Perform the cluster analysis using the K-means.	
Resources Required	● Teacher Resources ○ Google Colab notebook ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ● Student Resources ○ Google Colab notebook ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm Up** **Teacher-led Activity** **Student-led Activity** **Wrap up**	**5 mins** **15 min** **15 min** **5 min**

CONTEXT

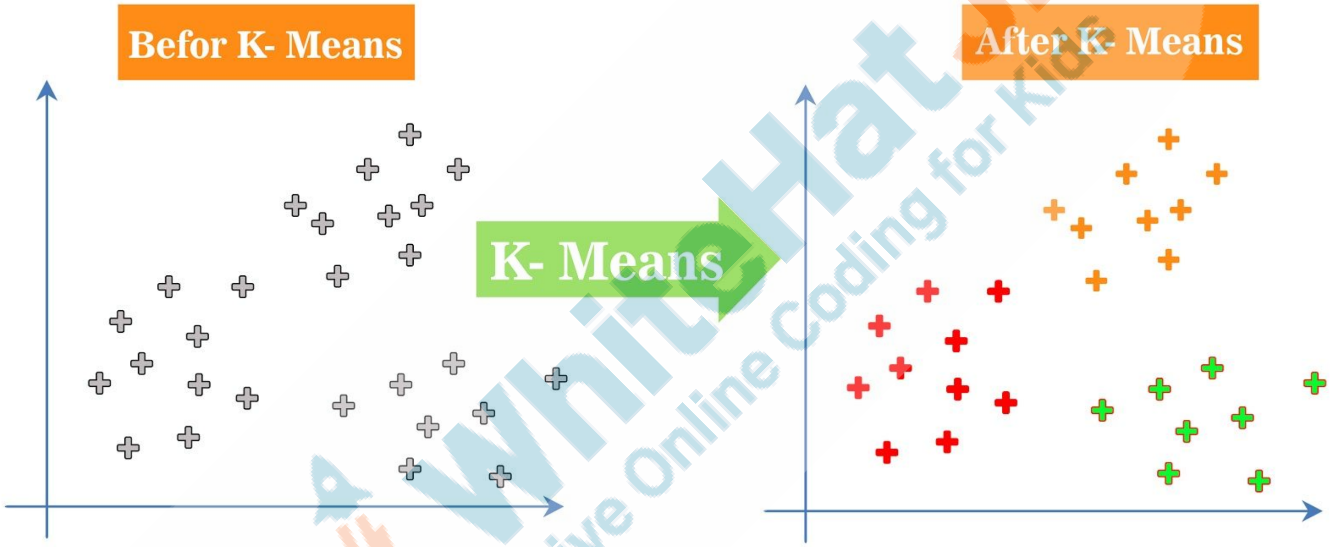
- **Introduce clustering or cluster analysis**

Class Steps	**Teacher Action**	**Student Action**
Step 1: **Warm Up** **(5 mins)**	Hi <Student Name>. How are you doing today? Let's quickly revise what we did in last class.	**ESR:** We learned about the confusion matrix. Using the confusion matrix we

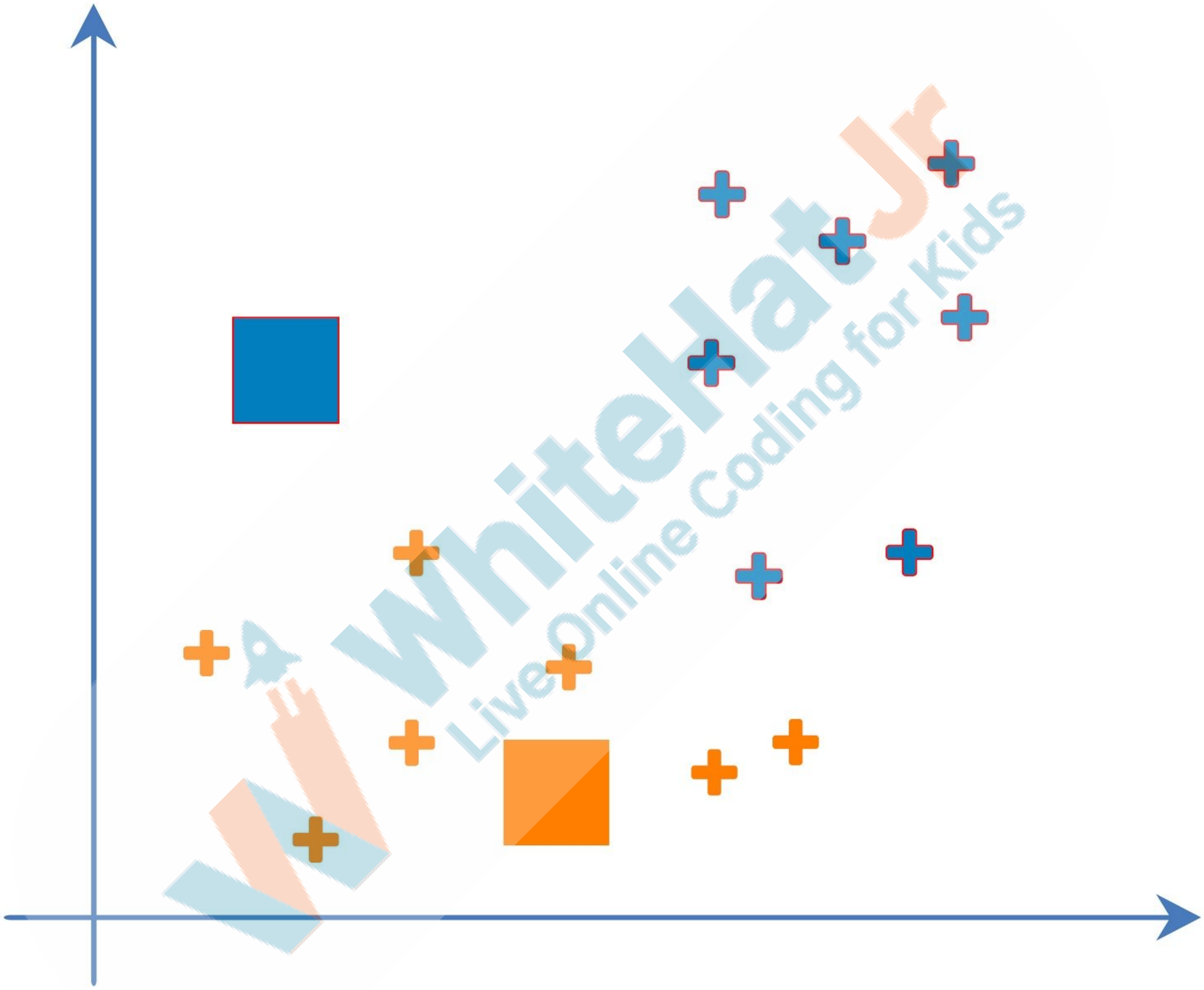
		checked the precision of the prediction or classification model. We also calculated the accuracy of the prediction model that we built.
	Very good! Let's say you have some red, blue, green and yellow coloured pins scattered on the floor. And now you have to seperate them and make groups of them so what will you do?	**ESR:** I'll separate them on the basis of the color and make a group of them. For eg:- i'll separate red colored pins and put them together then separate blue colored pins and put them together, then separate the green colored pins and put them together and finally take the yellow colored pins and put them together.
	Awesome. So here you grouped them on the basis of their color the same way we can also group our data and this group of data is called cluster or clusters for many. Clustering or cluster analysis is an unsupervised learning problem. It is often used as a data analysis technique for discovering interesting patterns in data, such as groups of customers based on their behavior. There are many clustering algorithms to choose from and no single best clustering algorithm for all cases. Today we'll explore one such algorithm. Are you excited for it?	**ESR:** Yes!

<table>
<tr><td></td><td>Let's get started then.</td><td></td></tr>
<tr><td colspan="3"><b>Teacher Initiates Screen Share</b></td></tr>
<tr><td colspan="3"><b><u>CHALLENGE</u></b><ul><li><b>Explore the steps to perform the Cluster analysis.</b></li><li><b>Explore the Elbow method</b></li></ul></td></tr>
<tr><td><b>Step 2:
Teacher-led Activity
(15 min)</b></td><td>Let's understand about clustering in more detail.
Let's say when you are looking for things such as music, you might want to look for meaningful groups. It could be from a particular artist, a particular genre, a particular language or a particular decade. How you group items gives you more insights about it.

In Machine learning we often group the examples to understand more about the data.</td><td><i>Student listens and asks questions.</i></td></tr>
<tr><td></td><td>Clustering or cluster analysis has a wide use of activities. Based on the example above, it can be used in the field of biology to differentiate species from each other, or it can be used to identify different images / audio. It can also be used to group behaviours, or detecting abnormal behaviour.
There are many algorithms which are used for clustering . One such widely used algorithm is the K-means algorithm.</td><td><i>The student observes and learns.</i></td></tr>
</table>

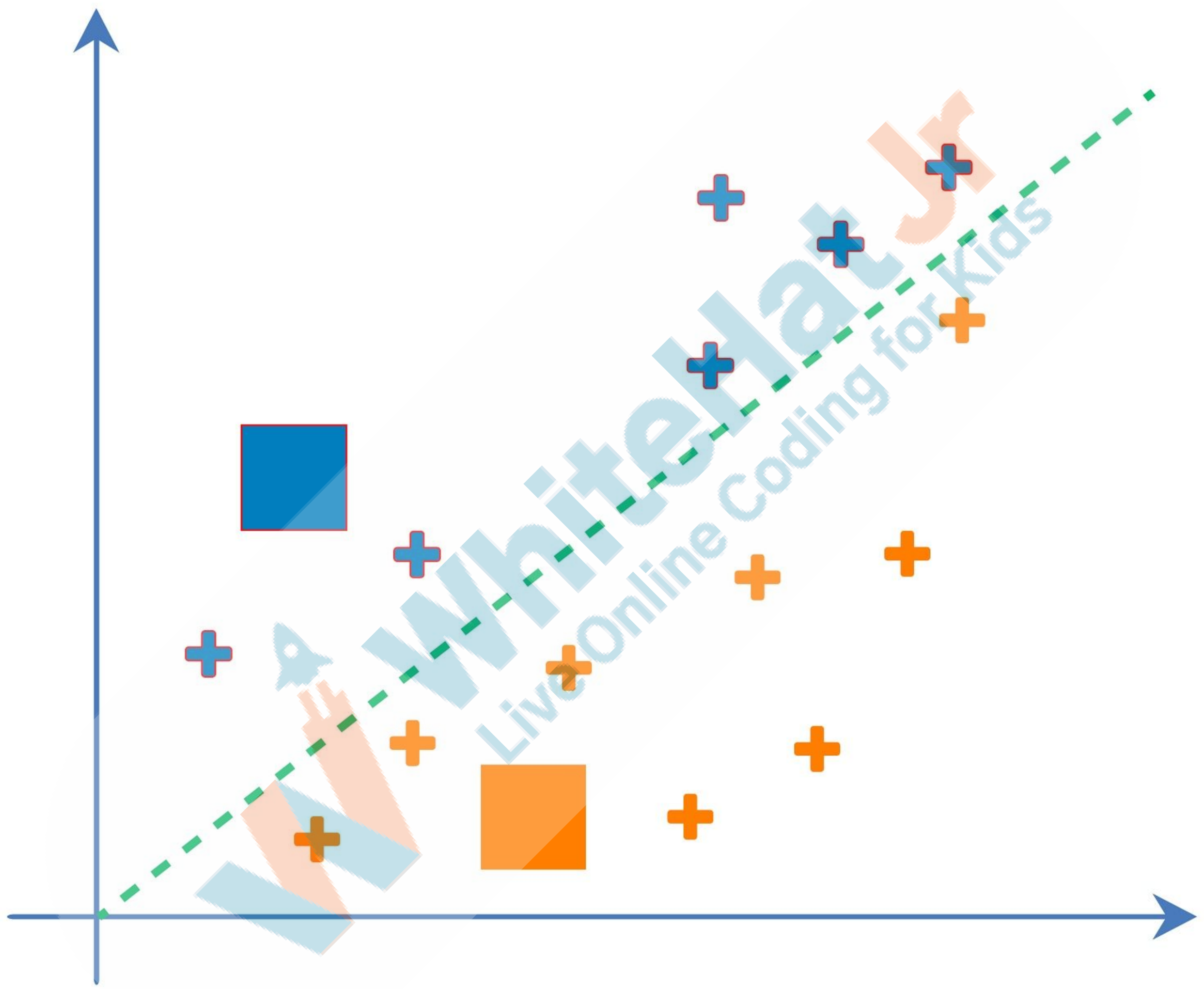
	<Teacher opens image from Teacher Activity 1 and shows it to the student > And today we are going to learn the k mean algorithm.	
		
	The first step to perform here is to decide the number of clusters. The K signifies the number of clusters that the algorithm would find in the dataset. Choosing the right K is very important. Sometimes, it is clearly visible from the dataset when it is visualized, however, most of the time, this is not the case.	*The student observes and learns.*

	Steps to perform the K-means algorithm: Step 1 **Choose the number K of clusters** Step 2 **Select randomly the center points (centroids) for the K clusters (2 in this case)** *<Teacher opens the link and shows the image* *https://drive.google.com/uc?export=view&id=1PQ28Olk1DQzXC0HnSEudZSUx4N8LKWfA*>	*Student listens and asks questions.*

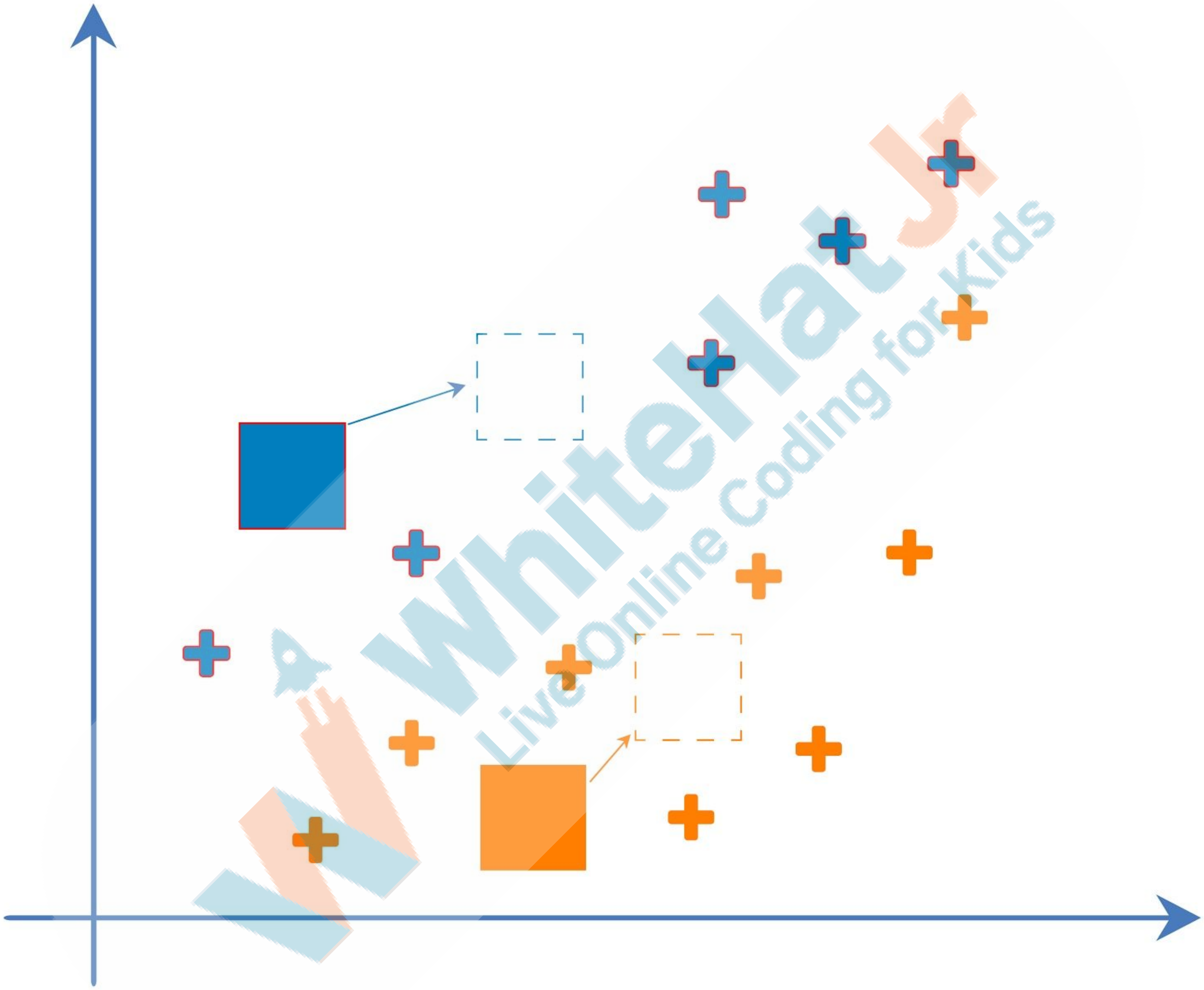


	Step 3 **Assign each data point to the closest centroid** *<Teacher opens the link and shows*	

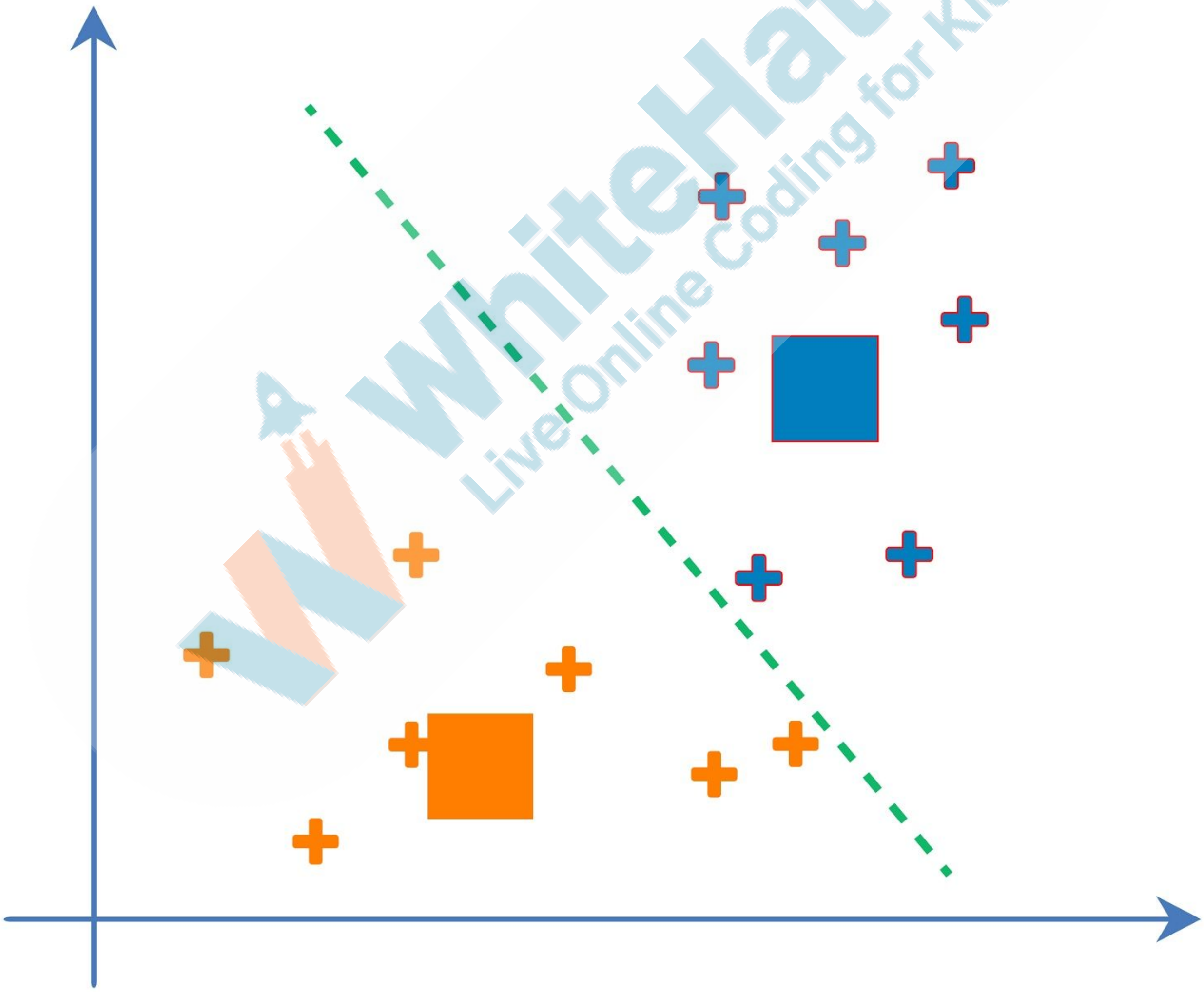
	the image https://drive.google.com/uc?export=view&id=10AeUS7ARbR_EN9sNhtl7F8cuOGDNhjOV>	

<table>
<tr><td colspan="3"></td></tr>
<tr><td></td><td>Step 4
<b>Shift the centroids a little for all the clusters</b>
<i>&lt;Teacher opens the link and shows the image</i></td><td></td></tr>
</table>

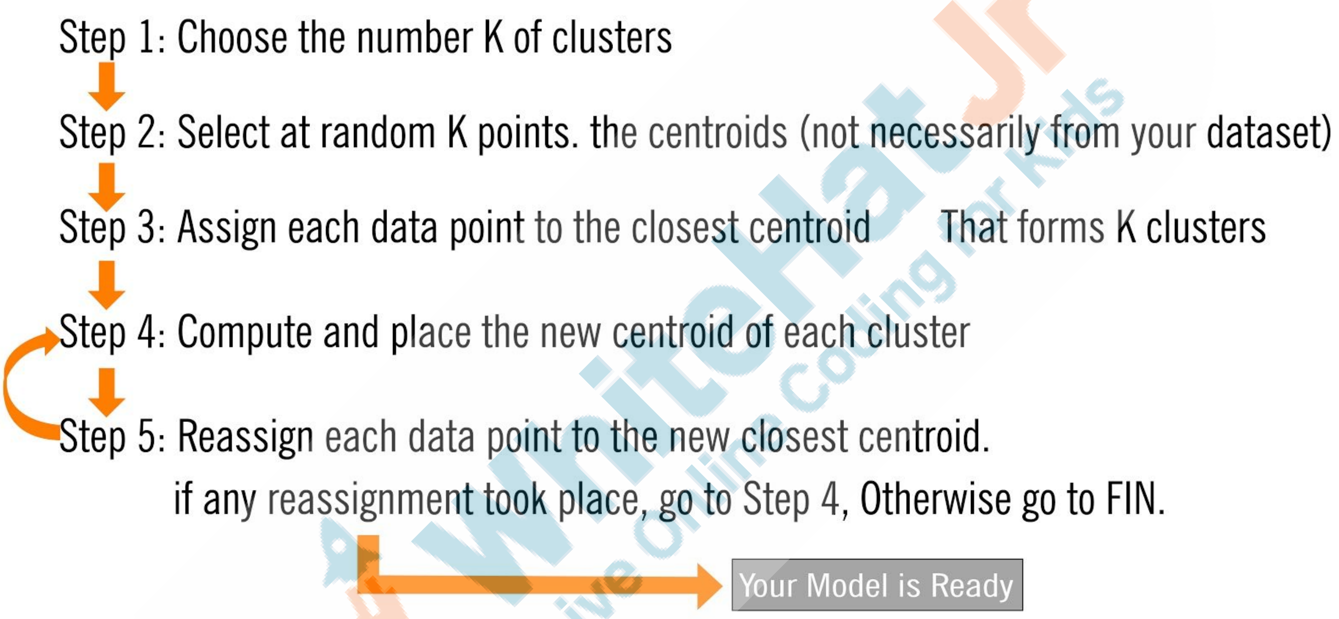
	https://drive.google.com/uc?export=view&id=1F7IfWlqj5JT8zqUVetHbZOvYyDUvcBvU>	



	Step 5 **Re-assign each data point to the new closest centroid. If any points got reassigned, repeat `Step 4` again otherwise the model is ready.**	

	<Teacher opens the link and shows the image https://drive.google.com/uc?export=view&id=1sJx_QRvVDFXE1Otm-uApU0FvTMF4600t>	
		

	So to summarize. *<Teacher opens the image and summarizes the steps >* https://drive.google.com/uc?export=view&id=1ZOG4uYODnOBpJwfpjx5E3TVN8qIx0EJw	-



Step 1: Choose the number K of clusters

Step 2: Select at random K points. the centroids (not necessarily from your dataset)

Step 3: Assign each data point to the closest centroid That forms K clusters

Step 4: Compute and place the new centroid of each cluster

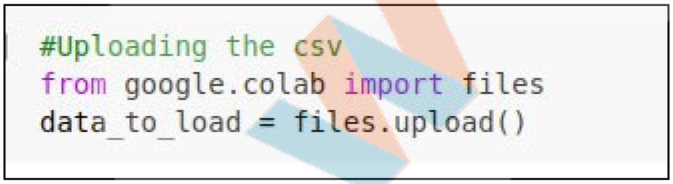
Step 5: Reassign each data point to the new closest centroid.
if any reassignment took place, go to Step 4, Otherwise go to FIN.

Your Model is Ready

	Alright now let's learn how it works as we write code for it. You'll be writing the code and I'll be helping you with it . Sounds good?	**ESR:** Yes!
	Let’s get started then.	

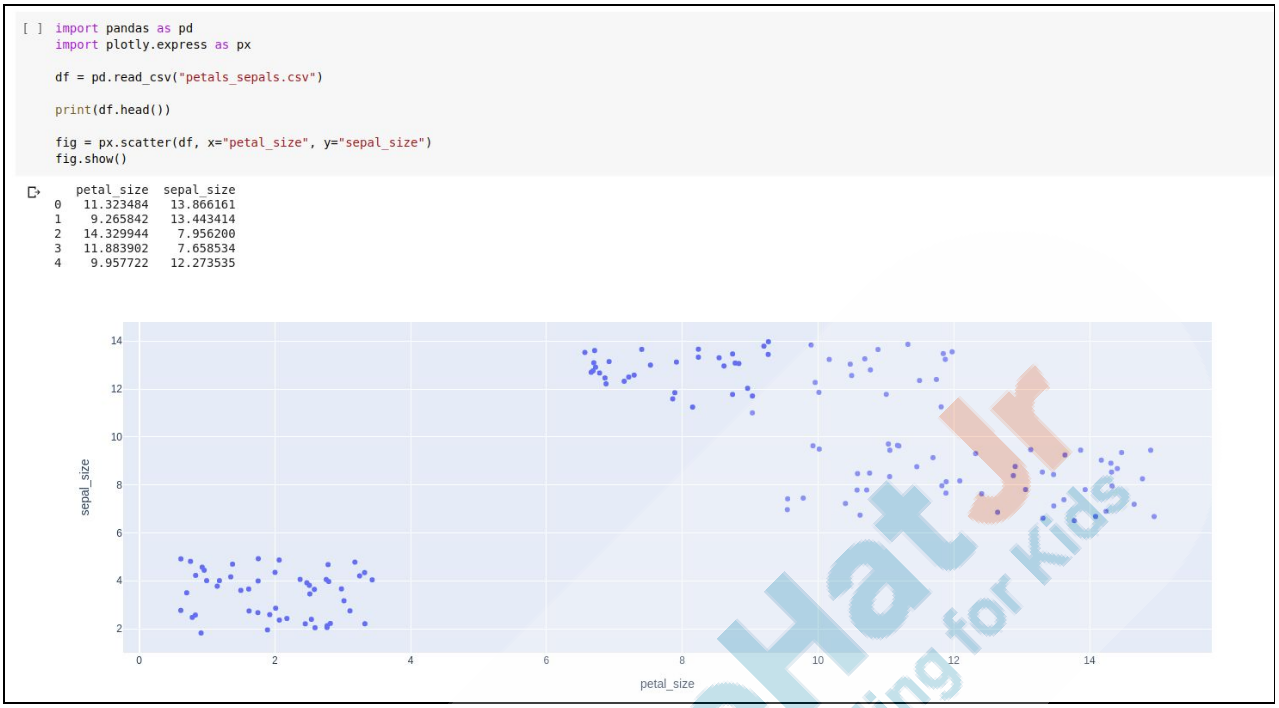
Teacher Stops Screen Share

	Now it's your turn. Please share your	

	screen with me.	
● Ask Student to press ESC key to come back to panel **● Guide Student to start Screen Share** **● Teacher gets into Fullscreen**		
ACTIVITY **● Student codes to perform the cluster analysis.** **● Conclude the findings from the analysis.**		
Step 3: Student-Led Activity (15 min)	*Teacher helps the student to download data from Student activity 1 and open a new Colab notebook from Student Activity 2.*	*Student downloads the data from Student Activity 1 and opens the Colab notebook from Student Activity 2.*
	Teacher helps the student to upload the data in the Colab notebook.	*The Student uploads data in the Colab notebook.*
		

```
#Uploading the csv
from google.colab import files
data_to_load = files.upload()
```

	Here we have some data of different kinds of petals and sepals of flowers. Let's plot it and see how it looks. *<Teacher helps student to plot the data in the scatter plot>* Code:- # here we plot the data on the normal scatter plot. **import pandas as pd** **import plotly.express as px** **df = pd.read_csv("petals_sepals.csv")** **print(df.head())** **fig = px.scatter(df, x="petal_size", y="sepal_size")** **fig.show()** So what do you see?	*Student codes to plot the data in the scatter plot.* **ESR:** We can see the dots have formed groups or clusters.



```
[ ] import pandas as pd
    import plotly.express as px

    df = pd.read_csv("petals_sepals.csv")

    print(df.head())

    fig = px.scatter(df, x="petal_size", y="sepal_size")
    fig.show()
```

```
      petal_size  sepal_size
   0   11.323484   13.866161
   1    9.265842   13.443414
   2   14.329944    7.956200
   3   11.883902    7.658534
   4    9.957722   12.273535
```

	Yes. Now let's see how to choose the right K. To do this we are going to use the WCSS perimeter to evaluate the choice of K. WCSS stands for **Within Cluster Sum of Squares**. What this means is that we are going to choose a center point for a cluster, from where all the points falling inside that cluster will be closest. Then, we will calculate the distance of all the points from the center, add up all the distances and then note the value. We will then take 2 centre points and do the same. We will choose the value of K to be the one which has the minimum sum of all the distances.	*Student codes to find the WCSS of the clusters.*

Then we'll use the elbow method to choose the best value for K. Let's see how it works!

<Teacher helps student with the code.>

Code :-

# here we are coding to find the best possible values of k using WCSS perimeter

```
from sklearn.cluster import KMeans


#Pandas provide a unique method to retrieve rows from a Data frame. Dataframe.iloc[] method is used when the index label of a data frame is something other than numeric series of 0, 1, 2, 3….n or in case the user doesn’t know the index label.

X = df.iloc[:, [0, 1]].values

print(X)

wcss = []

#Here the range is taken till 11 because we just need 10 cluster points.
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state = 42)
    kmeans.fit(X)

    # inertia method returns wcss for that model
    wcss.append(kmeans.inertia_)
```

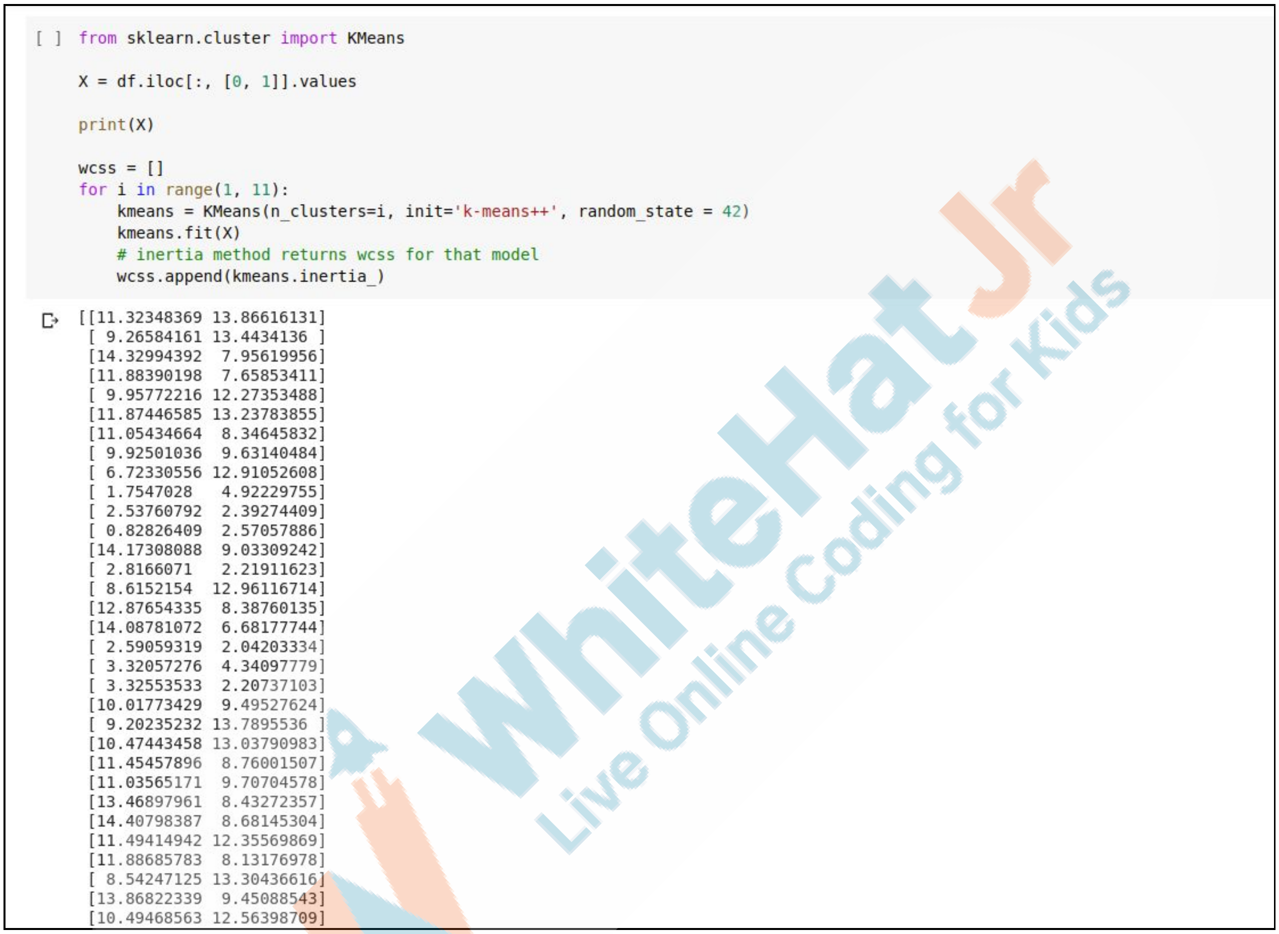
Here, we are first using the iloc[] method to get the list. Inside the [] of the iloc method, we are saying that we want all the values (:) in the form of a list ([0, 1]) containing the 0th and the first elements of the rows.

We are then creating an empty list to store our WCSS values.

Finally, since there are usually less than 10 clusters for most of the cases, we are iterating in the range(1, 11) and we are using the KMeans() classifier. In the classifier, we are passing the number of clusters we want to use for the classifier (i), the initialisation method (k-means++ since it is one of the best algorithms to find the kmeans value) and a random state (required tell the classifier where it should start from. It helps in saving time).

Once our classifier is ready, we are fitting our list of lists we created and finding the inertia of the classifier (which is also the WCSS) value and appending this value into our empty list.

	random_state has value 42 because we need to start with some random value. It can be any value.	



```
[ ] from sklearn.cluster import KMeans

    X = df.iloc[:, [0, 1]].values

    print(X)

    wcss = []
    for i in range(1, 11):
        kmeans = KMeans(n_clusters=i, init='k-means++', random_state = 42)
        kmeans.fit(X)
        # inertia method returns wcss for that model
        wcss.append(kmeans.inertia_)
```

```
[[11.32348369 13.86616131]
 [ 9.26584161 13.4434136 ]
 [14.32994392  7.95619956]
 [11.88390198  7.65853411]
 [ 9.95772216 12.27353488]
 [11.87446585 13.23783855]
 [11.05434664  8.34645832]
 [ 9.92501036  9.63140484]
 [ 6.72330556 12.91052608]
 [ 1.7547028   4.92229755]
 [ 2.53760792  2.39274409]
 [ 0.82826409  2.57057886]
 [14.17308088  9.03309242]
 [ 2.8166071   2.21911623]
 [ 8.6152154  12.96116714]
 [12.87654335  8.38760135]
 [14.08781072  6.68177744]
 [ 2.59059319  2.04203334]
 [ 3.32057276  4.34097779]
 [ 3.32553533  2.20737103]
 [10.01773429  9.49527624]
 [ 9.20235232 13.7895536 ]
 [10.47443458 13.03790983]
 [11.45457896  8.76001507]
 [11.03565171  9.70704578]
 [13.46897961  8.43272357]
 [14.40798387  8.68145304]
 [11.49414942 12.35569869]
 [11.88685783  8.13176978]
 [ 8.54247125 13.30436616]
 [13.86822339  9.45088543]
 [10.49468563 12.56398709]
```

	Now let's plot the data in a normal line plot. *<Teacher helps student plot the line plot>* Code:- #importing pyplot and sns **import matplotlib.pyplot as plt** **import seaborn as sns**	*Student codes to plot the WCSS in a line plot.*

#plotting a figure to show an elbow like structure in the graph

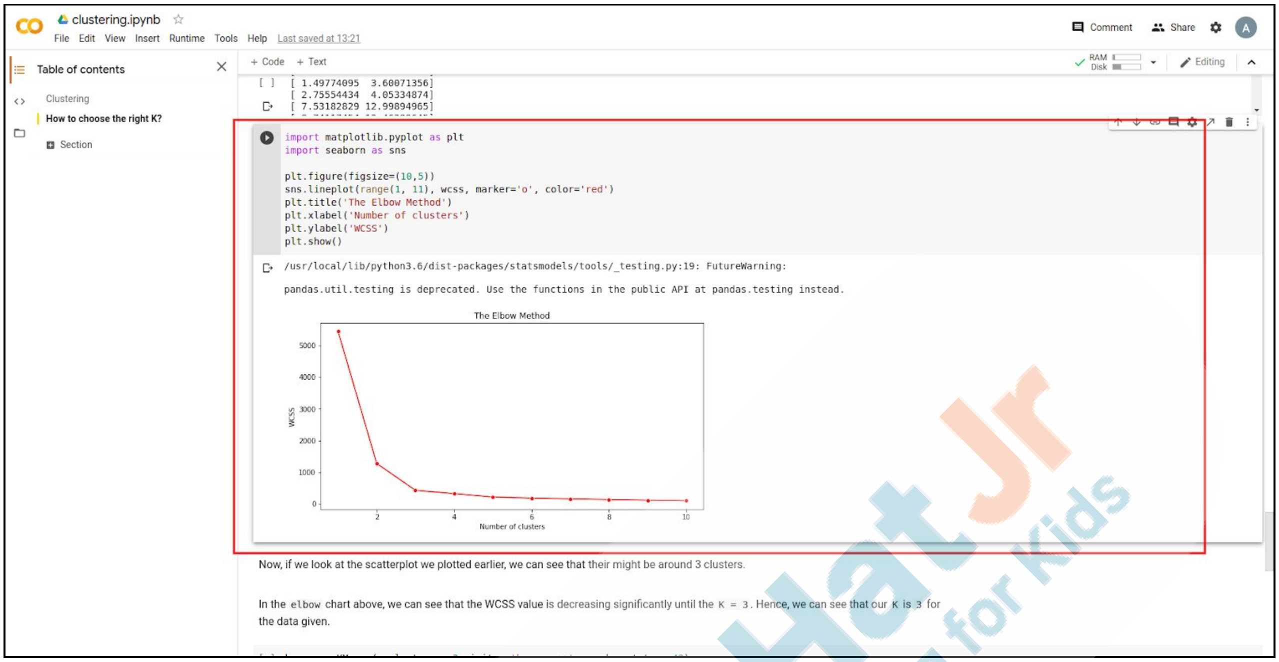
```
plt.figure(figsize=(10,5))
sns.lineplot(range(1, 11), wcss,
marker='o', color='red')
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

Here, are using the pyplot and the seaborn libraries to create a chart for k-means.

We are first specifying the size of our chart (10 units in width and 5 units in height).

We are then using the sns to create a lineplot (from 1, 11 since it was our range) and we are passing our list of wcss values we created earlier. We are also specifying the marker to be “o” or a dot and the color of the line (red).

We are finally adding the titles and the labels and displaying the chart.



```
[ ]  [ 1.49774095  3.60071356]
     [ 2.75554434  4.05334874]
     [ 7.53182829 12.99894965]
```

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10,5))
sns.lineplot(range(1, 11), wcss, marker='o', color='red')
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

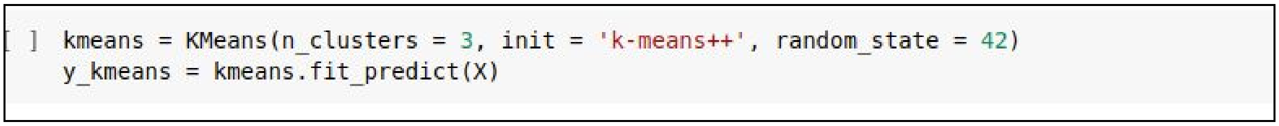
```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:

pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
```

Now, if we look at the scatterplot we plotted earlier, we can see that their might be around 3 clusters.

In the `elbow` chart above, we can see that the WCSS value is decreasing significantly until the `K = 3`. Hence, we can see that our `K` is `3` for the data given.

	What do we see here? Perfect! Now using the K means function we'll find the proper cluster points. *<Teacher helps student to find the proper cluster points.>* Code: **kmeans = KMeans(n_clusters = 3, init = 'k-means++', random_state = 42)** **y_kmeans = kmeans.fit_predict(X)** Now, since we know the K should be 3, we are again creating a classifier with a number of clusters as 3 and we are fitting our X (list of lists of petal and sepal sizes) into this classifier.	**ESR:** In the elbow chart above, we can see that the WCSS value is decreasing significantly until the K = 3. Hence, we can see that our K is 3 for the data given. *Student codes to find the proper cluster points.*



```
[ ]  kmeans = KMeans(n_clusters = 3, init = 'k-means++', random_state = 42)
     y_kmeans = kmeans.fit_predict(X)
```

y_means is now a list of values containing 0, 1 and 2. This value is based on the cluster where the corresponding element in X should be.

What we can do next is that we can separate out all the data-points for cluster 0, cluster 1 and cluster 2 and plot these points in different colors.

Now let's plot these cluster points on the scatter plot.

Teacher helps the student with the code to plot the cluster points on the plot.

Code:-

#here we are going to create a scatter plot with the groups and different colors of center points for each group.

```
plt.figure(figsize=(15,7))
sns.scatterplot(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], color = 'yellow', label = 'Cluster 1')
sns.scatterplot(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], color = 'blue', label = 'Cluster 2')
sns.scatterplot(X[y_kmeans == 2,
```

Student codes to plot the scatter plot on the graph.

```
0], X[y_kmeans == 2, 1], color =
'green', label = 'Cluster 3')
sns.scatterplot(kmeans.cluster_centers_[:, 0],
kmeans.cluster_centers_[:, 1],
color = 'red', label =
'Centroids',s=100,marker=',')
plt.grid(False)
plt.title('Clusters of Flowers')
plt.xlabel('Petal Size')
plt.ylabel('Sepal Size')
plt.legend()
plt.show()
```

Here, we are first specifying the size of the plot.

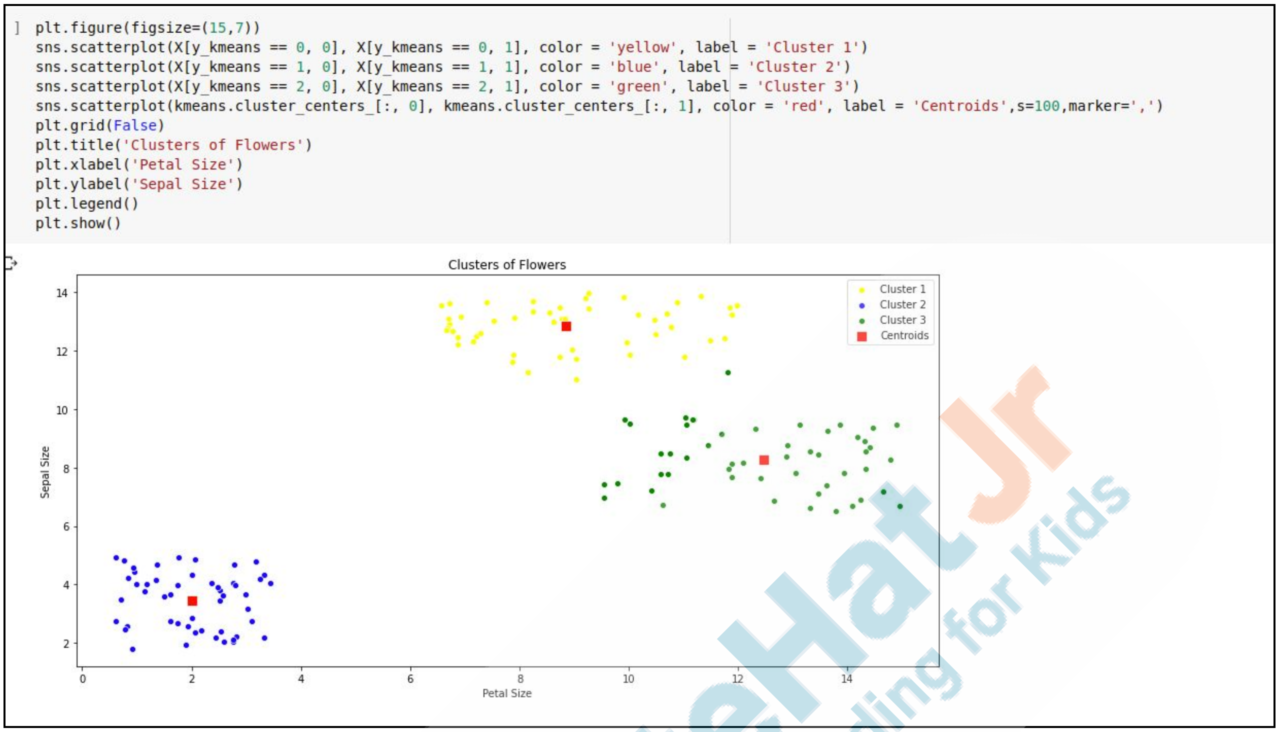
We are then creating 3 scatterplots. Let’s take one example -

```
sns.scatterplot(X[y_kmeans == 0,
0], X[y_kmeans == 0, 1], color =
'yellow', label = 'Cluster 1')
```

Here, the first thing that we are doing is that we are taking the 0th element from X (X contains petal and sepal sizes (2 elements)) based on if this pair of petal and sepal sizes belongs to the 0th cluster (conditioned as y_means == 0). Similarly, we are taking the 1st element as well.

We are then giving it color “Yellow” and labeling it as “Cluster 1”

	We repeat this process for y_means 0, 1 and 2. We are then creating the centroids of these clusters with - **sns.scatterplot(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], color = 'red', label = 'Centroids',s=100,marker=',')** Here, we are first passing the center values for all the clusters with kmeans.cluster_centers_[:, 0] where it will take all the 0th elements from the entire list of cluser_centers_ for the X-Coordinate and repeating the same to take all the 1st elements from the entire list (Y-Coordinate). We are then coloring the centroids as red, giving them a label, sizing them as 100 (so they appear a bit bigger) and giving them a marker. We are finally just adding the titles and labels and displaying the chart.	



```
plt.figure(figsize=(15,7))
sns.scatterplot(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], color = 'yellow', label = 'Cluster 1')
sns.scatterplot(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], color = 'blue', label = 'Cluster 2')
sns.scatterplot(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], color = 'green', label = 'Cluster 3')
sns.scatterplot(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], color = 'red', label = 'Centroids',s=100,marker=',')
plt.grid(False)
plt.title('Clusters of Flowers')
plt.xlabel('Petal Size')
plt.ylabel('Sepal Size')
plt.legend()
plt.show()
```

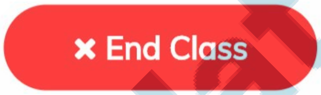
	What do we see here?	**ESR:** We see the 3 centroids in the clusters.
	We can see that our model has identified 3 clusters, which means that we had data for 3 different species of flowers.	

Teacher Guides Student to Stop Screen Share

FEEDBACK

- **Appreciate the student for their efforts**
- **Identify 2 strengths and 1 area of progress for the student**

Step 4: **Wrap-Up** **(5 min)**	So let's quickly review what we did today.	**ESR:** - We learned about clustering. - We saw the k- means algorithm. - We saw the elbow method to find the clusters.
	Amazing! You are making good progress. In next class we'll be learning more about machine learning. Are you excited for it?	**ESR:** Yes!
	Alright then see you in next class.	
Project Overview	Clustering **Goal of the Project:** In this project you will apply what you learned in the class and perform an clustering analysis using the k means clustering. **Story:** You are responsible for the operations of a telescope, so you have to find out how to use the device in optimal way so that you can scan as many stars as possible, what you have is huge data of stars, their size and how much light they	

	emit, use the knowledge of clustering algorithm to find out the clusters of the stars in the sky. I am very excited to see your project solution and I know you will do really well. Bye Bye!	
	Teacher Clicks 	
Additional Activities	*Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: ● What happened today? - Describe what happened - Code I wrote ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	*The student uses the markdown editor to write her/his reflection in a reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	K-means image.	https://i.imgur.com/rwkQNbv.png
Teacher Activity 2	REFERENCE LINK	https://colab.research.google.com/dr

		ive/13z4_8yahIREYaFxF1VtEqyMM2SkRKH-s?usp=sharing
Student Activity 1	data link	https://raw.githubusercontent.com/whitehatjr/datasets/master/C118/petals_sepals.csv
Student Activity 2	Google Colab Notebook	https://colab.research.google.com/notebooks/intro.ipynb#recent=true