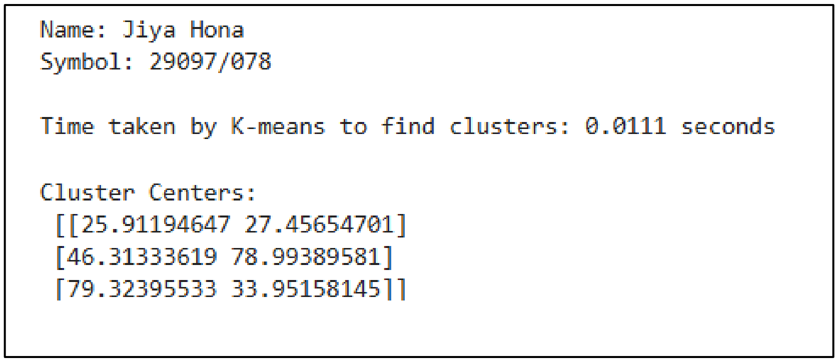
Lab 2

1. **Write a python program to implement K-means Clustering algorithm**
    - **Generate 1000 2-D data points in the range 0-100 randomly.**
    - **Divide data points into 3 clusters.**

Program:

```
import numpy as np

import time

from sklearn.cluster import KMeans

print("Name: Jiya Hona")

print("Symbol: 29097/078\n")

# Generate 1000 2D data points in range 0–100

data = np.random.rand(1000, 2) * 100

# Initialize K-means with 3 clusters

kmeans = KMeans(n_clusters=3, random_state=42)

# Measure execution time

start_time = time.time()

kmeans.fit(data)

end_time = time.time()

# Calculate and print results

time_taken = end_time - start_time

print(f"Time taken by K-means to find clusters: {time_taken:.4f} seconds\n")

print("Cluster Centers:\n", kmeans.cluster_centers_)
```

Output:



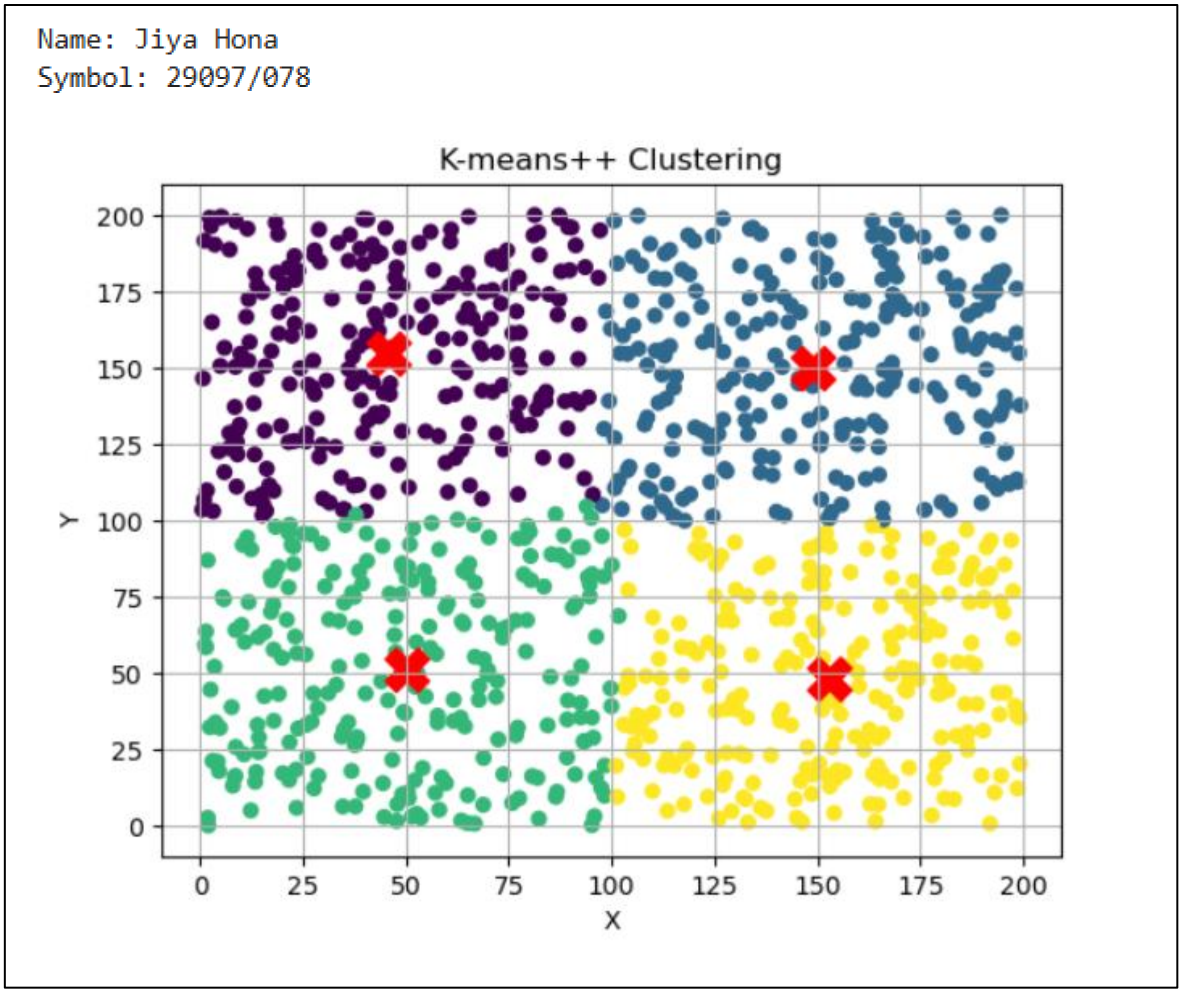
```
Name: Jiya Hona
Symbol: 29097/078

Time taken by K-means to find clusters: 0.0111 seconds

Cluster Centers:
 [[25.91194647 27.45654701]
 [46.31333619 78.99389581]
 [79.32395533 33.95158145]]
```

2. **Write a python program to implement K-means++ Clustering algorithm.**
    - **Generate 1000 2-D data points in the range 0-200 randomly.**
    - **Divide data points into 4 clusters.**

Program:

```
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

print("Name: Jiya Hona")
print("Symbol: 29097/078\n")

# Generate 1000 2-D data points in the range 0-200
data = np.random.rand(1000, 2) * 200

# Initialize K-means++ algorithm with 4 clusters
kmeans_plus = KMeans(n_clusters=4, init='k-means++', random_state=42)

# Fit the model
kmeans_plus.fit(data)

# Plot the data points and cluster centers
plt.scatter(data[:, 0], data[:, 1], c=kmeans_plus.labels_, cmap='viridis', s=30)
plt.scatter(kmeans_plus.cluster_centers_[:, 0], kmeans_plus.cluster_centers_[:, 1], s=300, c='red', marker='X')
plt.title("K-means++ Clustering")
plt.xlabel("X")
plt.ylabel("Y")
plt.grid(True)
plt.show()
```

Output:



```
Name: Jiya Hona
Symbol: 29097/078
```