Silhouette Analysis

I tried several times in jupyter notebook but as notebook kept crashing, I used Colab. But even when Colab crashed, I standardized the data and took two variables i.e., 'Request Balance' and 'FICO Score'. I found below silhouette analysis,values and graphs. Highest silhouette score is for cluster 3 and 4. From below we can understand the clustering pattern too.

In [13]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [14]:
```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving loan.csv to loan (1).csv
```

In [15]:
```
import pandas as pd
import io

df = pd.read_csv(io.BytesIO(uploaded['loan.csv']))
df.head()
```

Out[15]:

	Approval	Debt-to-Income Ratio	FICO Score	Request Amount	Interest
0	F	0.0	397	1000	450
1	F	0.0	403	500	225
2	F	0.0	408	1000	450
3	F	0.0	408	2000	900
4	F	0.0	411	5000	2250

In [16]:
```
X=df.iloc[:,2:4].values
```

In [17]:
```
from sklearn import preprocessing
# Get column names first
scaler = preprocessing.StandardScaler()
# Fit your data on the scaler object
scaled_df = scaler.fit_transform(X)
```

In [7]:
```
import matplotlib.pyplot as plt
from sklearn.datasets.samples_generator import make_blobs
from sklearn.cluster import Birch
# Creating the BIRCH clustering model
model = Birch( n_clusters = 5, threshold = 0.1)
```

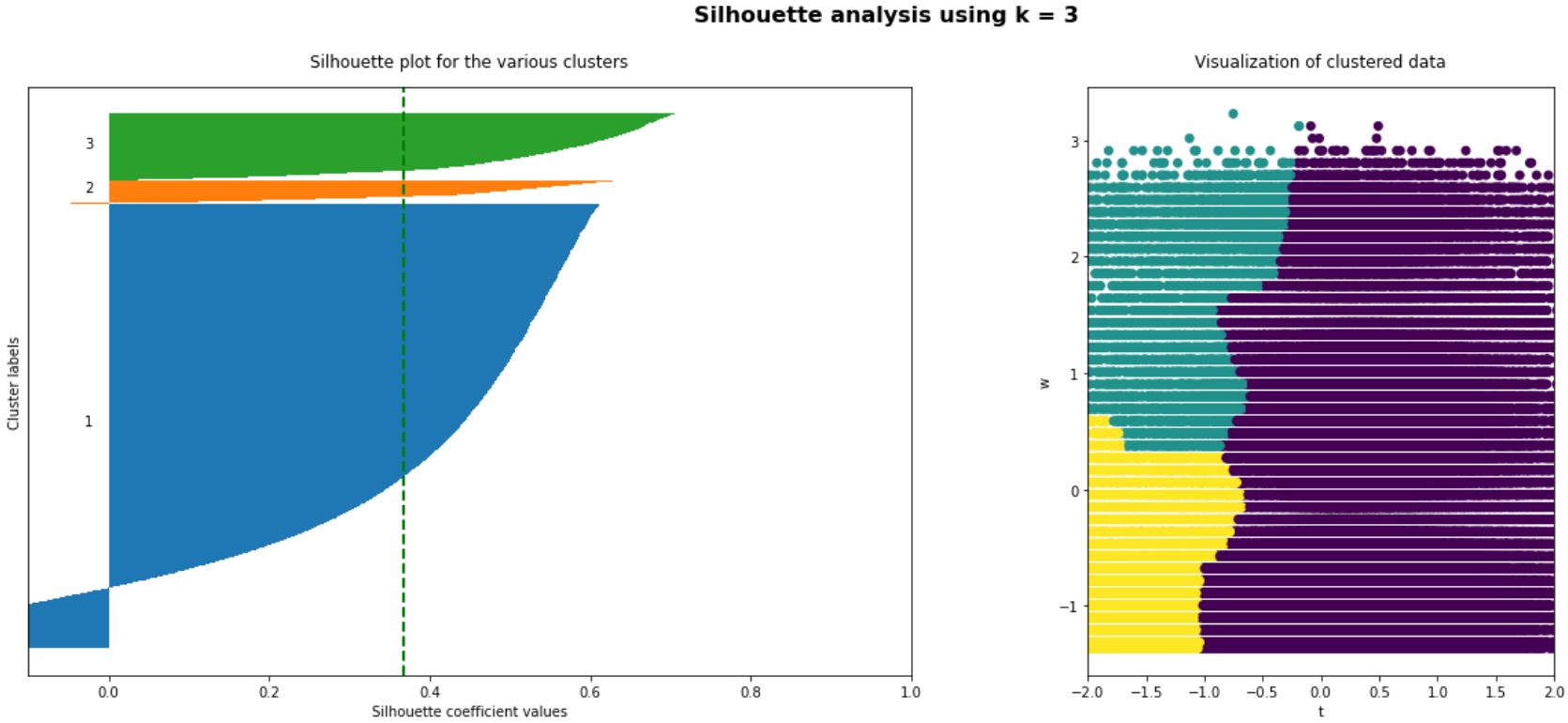
```
/usr/local/lib/python3.6/dist-packages/sklearn/utils/deprecation.py:144: FutureWarning: The sklearn.datasets.
samples_generator module is  deprecated in version 0.22 and will be removed in version 0.24. The correspondin
g classes / functions should instead be imported from sklearn.datasets. Anything that cannot be imported from
sklearn.datasets is now part of the private API.
  warnings.warn(message, FutureWarning)
```

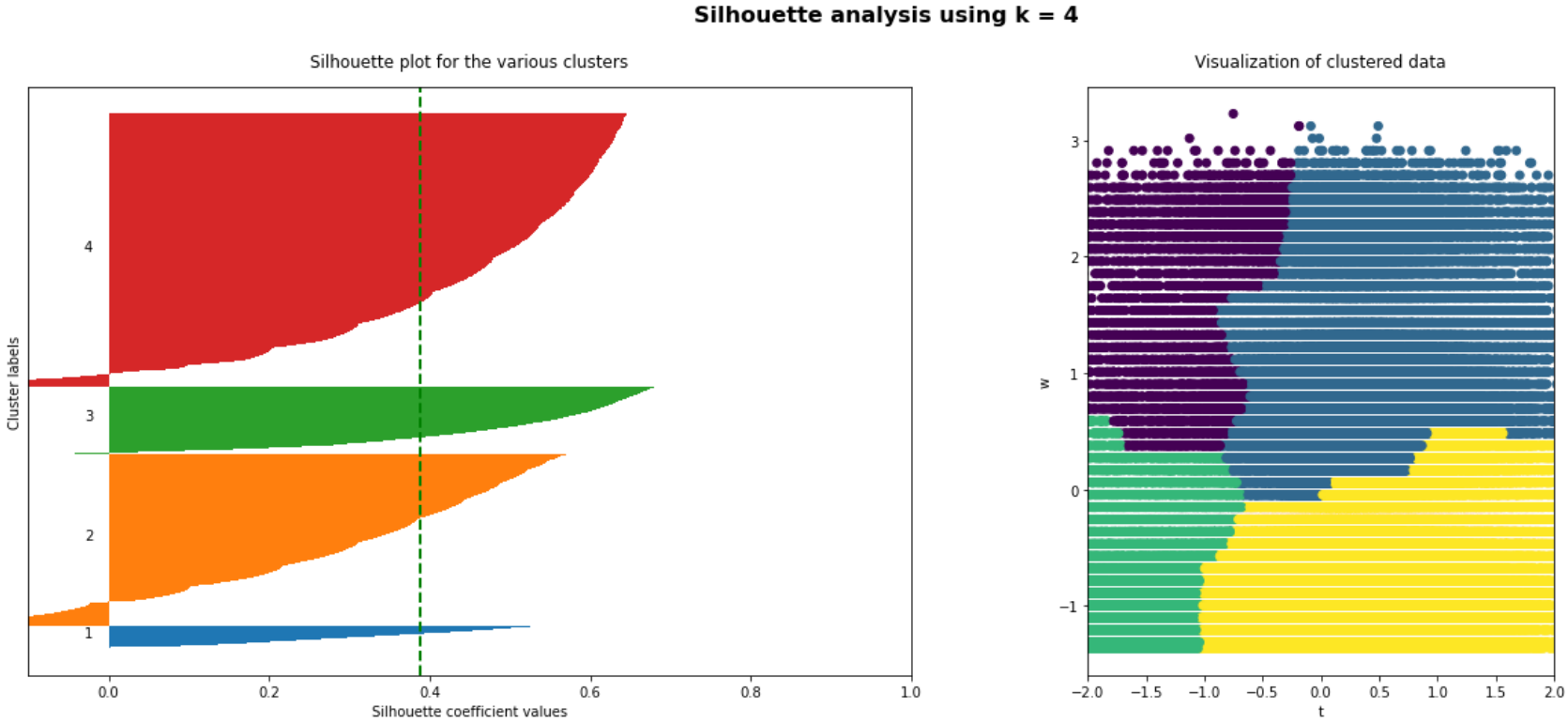
In [9]:
```
# Fit the data (Training)
model.fit(scaled_df)

# Predict the same data
pred = model.predict(scaled_df)
```

In [19]:

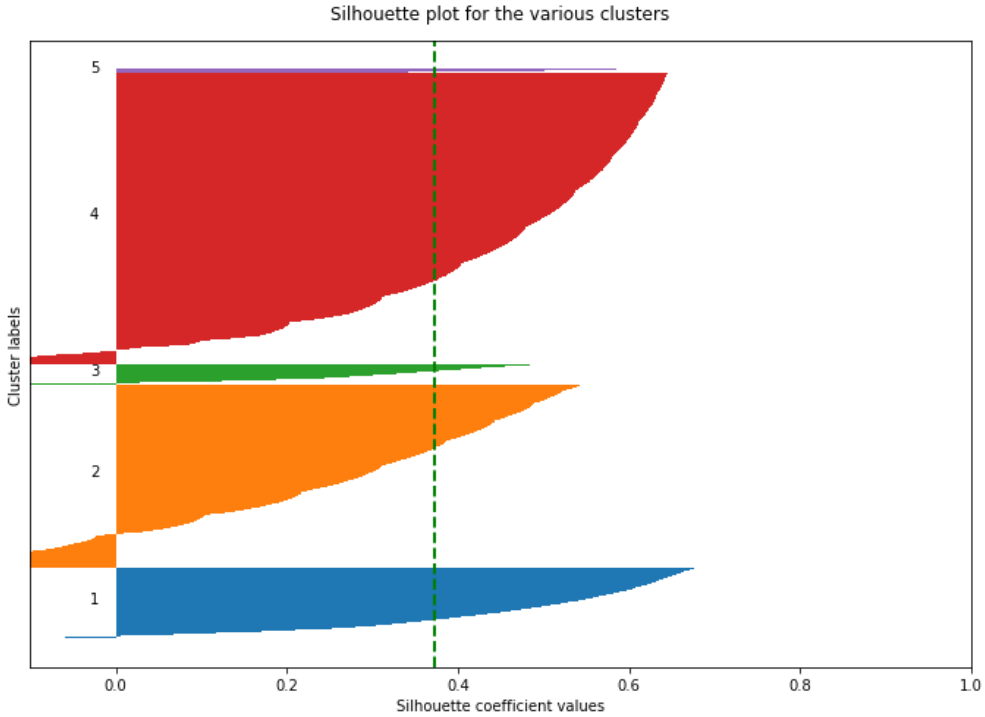
```
from sklearn.cluster import KMeans, SpectralClustering
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_samples, silhouette_score
import numpy as np
import matplotlib.pyplot as plt
for i, k in enumerate([ 3, 4, 5,6]):
    fig, (ax1, ax2) = plt.subplots(1, 2)
    fig.set_size_inches(18, 7)

    # Run the Kmeans algorithm
    model = Birch( n_clusters = k, threshold = 0.5)
    labels = model.fit_predict(scaled_df)
    centroids = model.n_clusters
    # Get silhouette samples
    silhouette_vals = silhouette_samples(scaled_df, labels)

    # Silhouette plot
    y_ticks = []
    y_lower, y_upper = 0, 0
    for i, cluster in enumerate(np.unique(labels)):
        cluster_silhouette_vals = silhouette_vals[labels == cluster]
        cluster_silhouette_vals.sort()
        y_upper += len(cluster_silhouette_vals)
        ax1.barh(range(y_lower, y_upper), cluster_silhouette_vals, edgecolor='none', height=1)
        ax1.text(-0.03, (y_lower + y_upper) / 2, str(i + 1))
        y_lower += len(cluster_silhouette_vals)

    # Get the average silhouette score and plot it
    avg_score = np.mean(silhouette_vals)
    ax1.axvline(avg_score, linestyle='--', linewidth=2, color='green')
    ax1.set_yticks([])
    ax1.set_xlim([-0.1, 1])
    ax1.set_xlabel('Silhouette coefficient values')
    ax1.set_ylabel('Cluster labels')
    ax1.set_title('Silhouette plot for the various clusters', y=1.02);

    # Scatter plot of data colored with labels
    ax2.scatter(scaled_df[:, 0], scaled_df[:, 1], c=labels)
    #ax2.scatter(centroids[:, 0], centroids[:, 1], marker='*', c='r', s=250)
    ax2.set_xlim([-2, 2])
    ax2.set_xlim([-2, 2])
    ax2.set_xlabel('t')
```

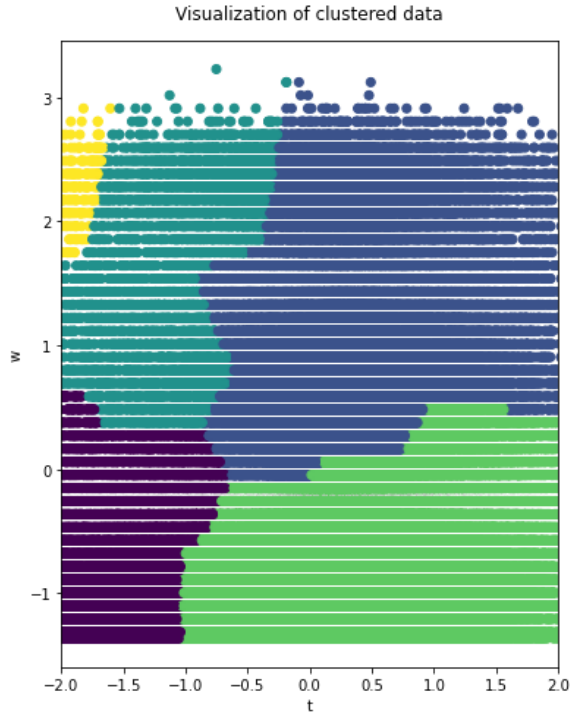
```
    ax2.set_ylabel('w')
    ax2.set_title('Visualization of clustered data', y=1.02)
    ax2.set_aspect('equal')
    plt.tight_layout()
    plt.suptitle(f'Silhouette analysis using k = {k}',
                 fontsize=16, fontweight='semibold', y=1.05);
```

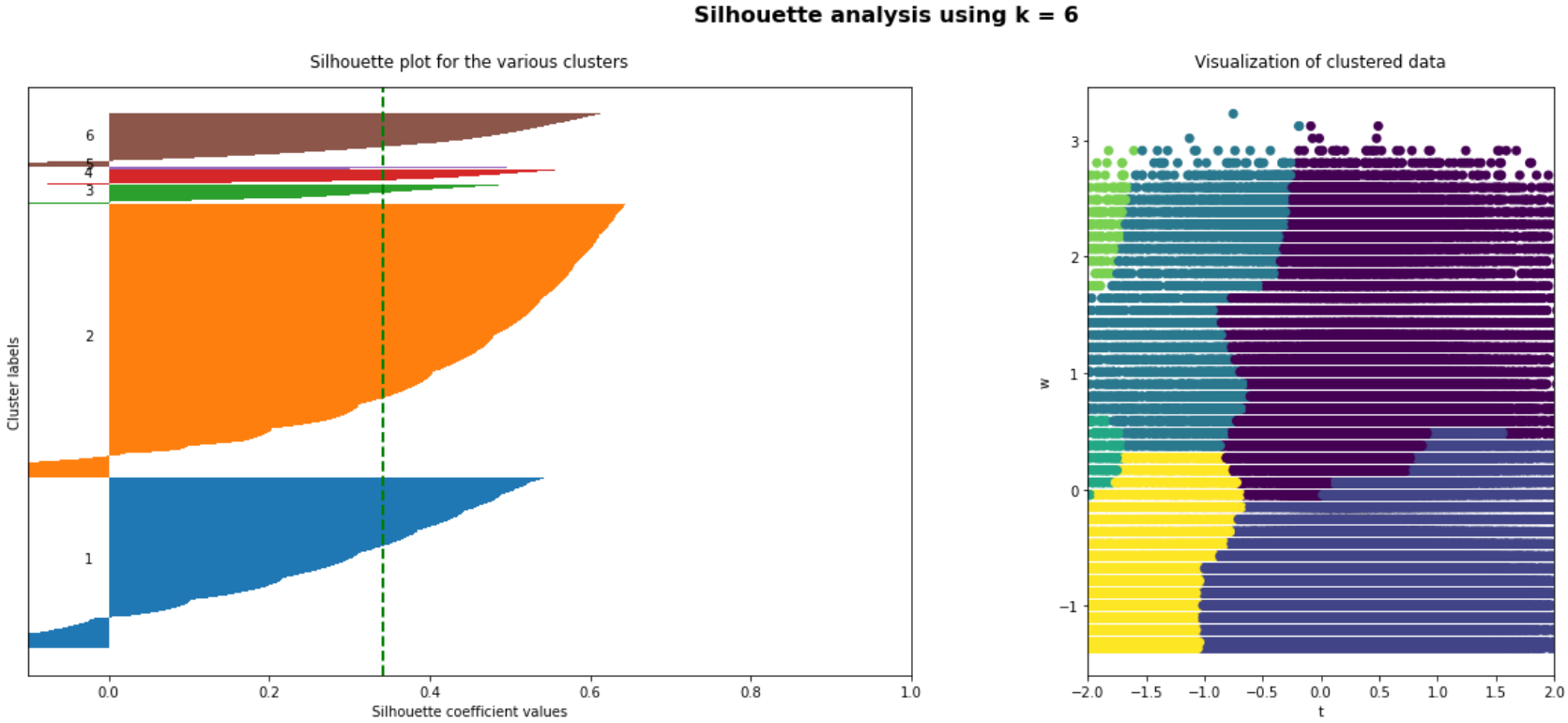

Silhouette analysis using k = 3
Silhouette plot for the various clusters
Cluster labels
3
2
1
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette coefficient values
Visualization of clustered data
w
3
2
1
0
−1
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
t


Silhouette analysis using k = 4
Silhouette plot for the various clusters
Cluster labels
4
3
2
1
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette coefficient values
Visualization of clustered data
w
3
2
1
0
−1
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
t

Silhouette analysis using k = 5







In [23]:
```
silhouette_vals
```

Out[23]:
```
array([0.41417284, 0.41333077, 0.42881247, ..., 0.38447377, 0.52000486,
       0.43791861])
```

In [ ]: