K Means Clustering

January 13, 2018

1 Importing the libraries

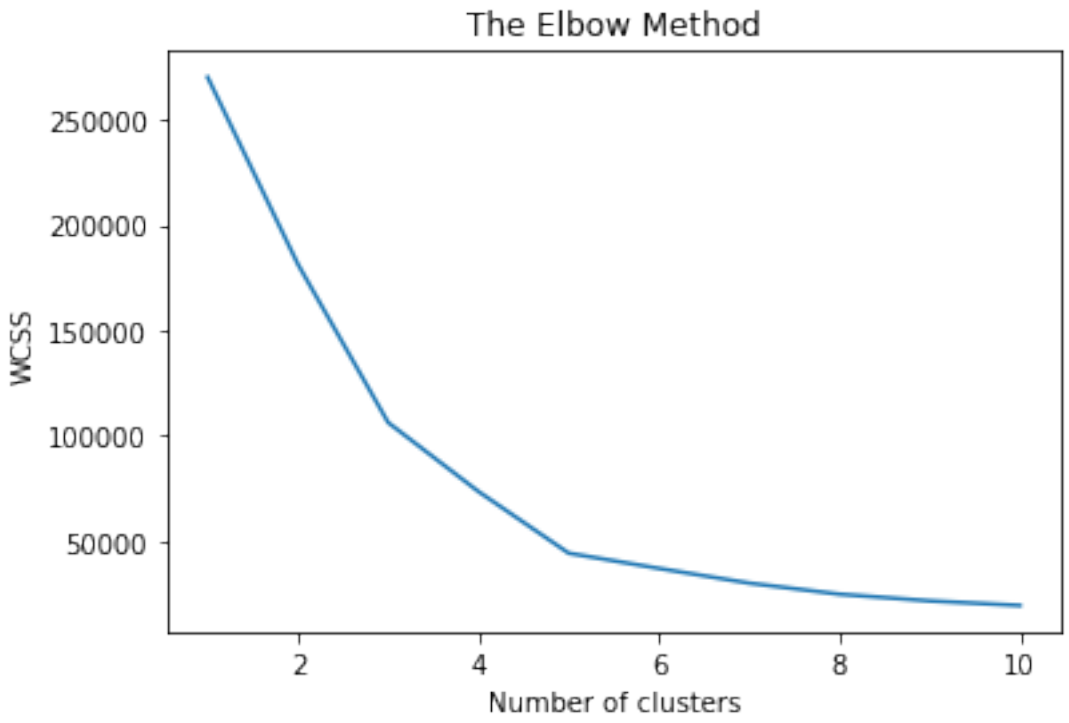
```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
```

2 Importing the dataset

```
In [2]: dataset = pd.read_csv('Mall_Customers.csv')
        X = dataset.iloc[:, [3, 4]].values
```

3 Using the elbow method to find the optimal number of clusters

```
In [3]: from sklearn.cluster import KMeans
        wcss = []
        for i in range(1, 11):
            kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
            kmeans.fit(X)
            wcss.append(kmeans.inertia_)
        plt.plot(range(1, 11), wcss)
        plt.title('The Elbow Method')
        plt.xlabel('Number of clusters')
        plt.ylabel('WCSS')
        plt.show()
```



3.1 We choose five.

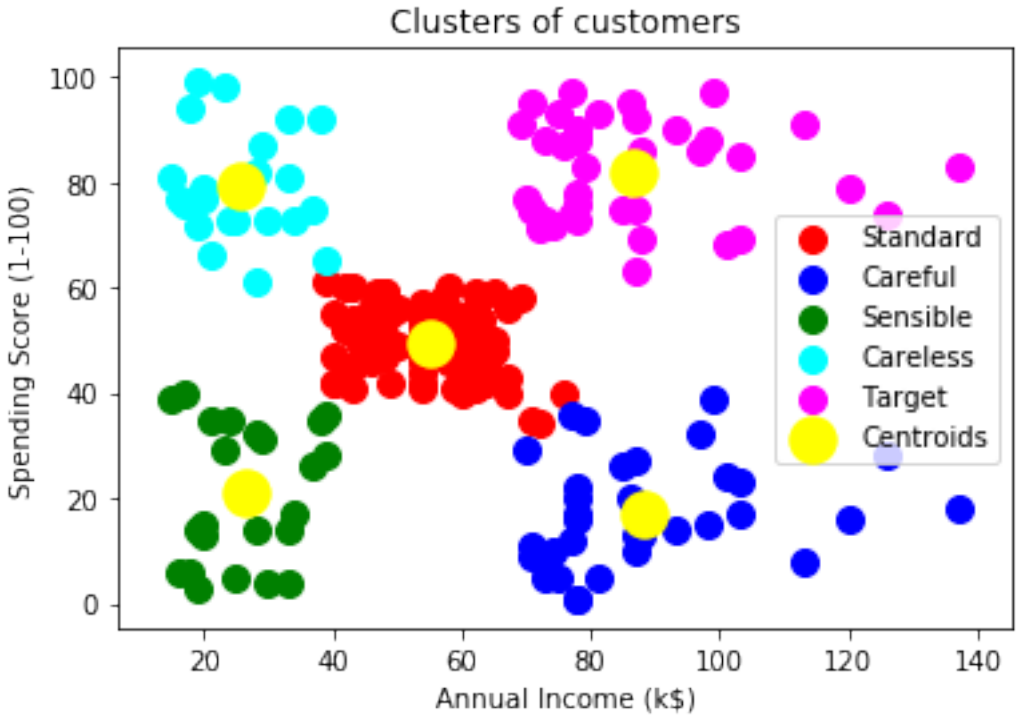
4 Fitting K-Means to the dataset

```
In [4]: kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
        y_kmeans = kmeans.fit_predict(X)
```

5 Visualising the clusters

```
In [5]: plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1],
                    s = 100, c = 'red', label = 'Standard')
        plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1],
                    s = 100, c = 'blue', label = 'Careful')
        plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1],
                    s = 100, c = 'green', label = 'Sensible')
        plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1],
                    s = 100, c = 'cyan', label = 'Careless')
        plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1],
                    s = 100, c = 'magenta', label = 'Target')
        plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
                    s = 300, c = 'yellow', label = 'Centroids')
        plt.title('Clusters of customers')
        plt.xlabel('Annual Income (k$)')
```

```
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



6 Save the predictions for visualization in Tableau

```
In [21]: z = []
         for i in range(len(X)):
             z.append(kmeans.predict([X[i]]))

In [23]: import numpy as np
         np.savetxt("classifications.csv",z,delimiter = ",")
```