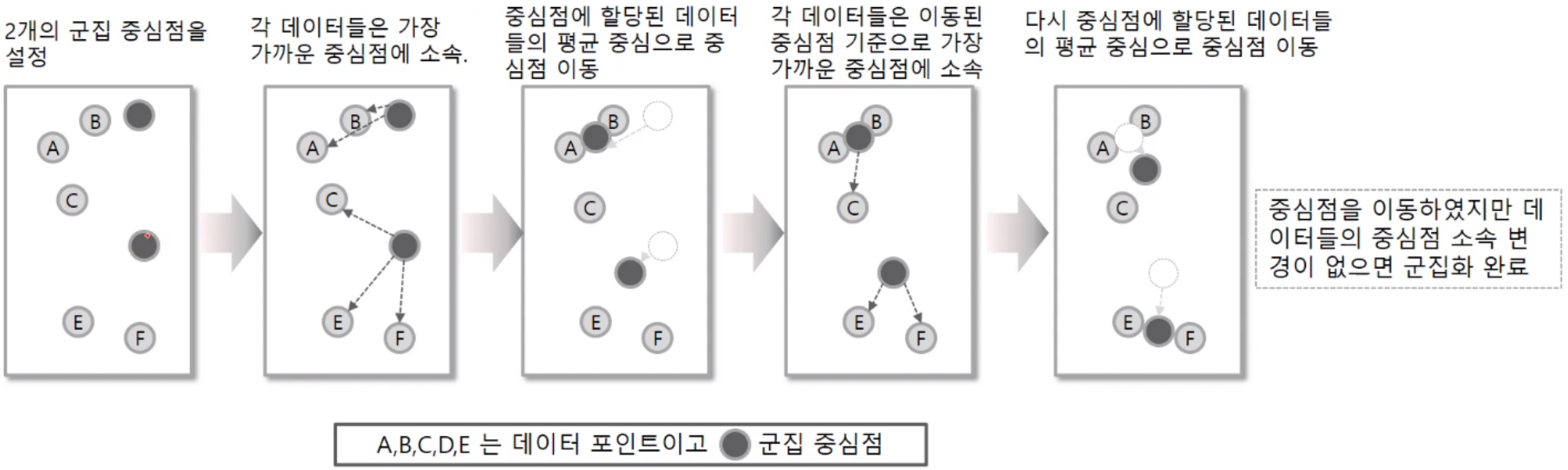
[비지도 학습 : 군집화]

비지도학습 : 군집화(Clustering)

- K- 평균(K-means) 군집화
  - 군집화에서 가장 일반적으로 사용하는 알고리즘
  - 군집 중심점(centroid)라는 특정함 임의의 지점을 선택해 해당 중심에 가장 가까운 포인트들을 선택하는 군집화 기법

- K-평균 중심점 선택 프로세스
  1) 선택된 포인트의 평균 지점으로 이동
  2) 이동된 중심점에서 다시 가까운 포인트를 선택
  3) 다시 중심점을 평균 지점으로 이동
  4) 더 이상 중심점의 이동이 없을 경우 반복을 멈추고 해당 중심점에 속하는 데이터 포인트들을 군집화



비지도학습 : 군집화(Clustering)

- **K- 평균(K-means) 군집화의 장점**
  - 일반적인 군집화에서 가장 많이 활용되는 알고리즘
  - 알고리즘이 쉽고 간결하다.

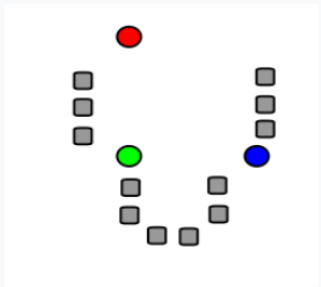
- **K- 평균(K-means) 군집화의 단점**
  - 거리 기반 알고리즘으로 속성의 개수가 많을 경우 군집화 정확도가 떨어진다.
    (PCA로 차원 감소한 후 적용하는 것이 바람직)
  - 반복을 수행하는데 반복횟수가 많을 수록 수행시간이 매우 느려진다.
  - 처음에 몇 개의 군집(cluster)을 선택할지 가이드가 어렵다.

- **군집 평가 - 실루엣 분석(silhouette analysis)**
  - 각 군집 간의 거리가 얼마나 효율적으로 분리되어 있는지를 나타냄
  - 효율적으로 잘 분리되었다는 것은 다른 군집과의 거리는 떨어져 있고 동일 군집끼리는 서로 가깝게 잘 뭉쳐 있다는 것을 의미
  - 군집화가 잘 될 수록 개별 군집은 비슷한 정도의 여유공간을 가지고 떨어져 있다.

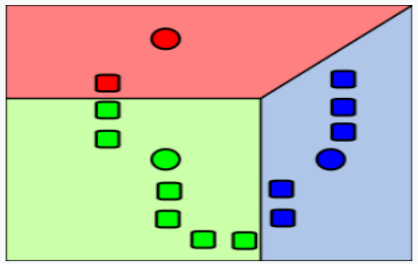
- **실루엣 계수(silhouette coefficient)**
  - 개별 데이터가 가지는 군집화 지표
  - 개별 데이터가 해당 데이터가 같은 군집 내의 데이터와 얼마나 가깝게 군집화되어 있고, 다른 군집에 있는 데이터와는 얼마나 멀리 분리되어 있는지를 나타내는 지표

비지도학습 : 군집화(Clustering)

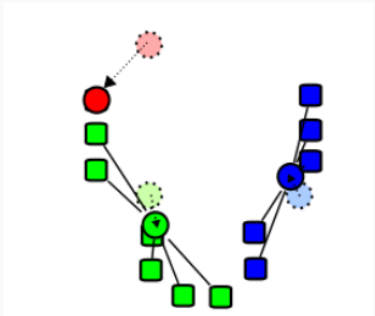
- K- 평균(K-means) 군집화



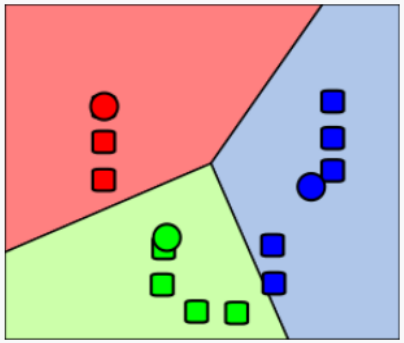
1) 초기 k "평균값" (위의 경우 k=3) 은 데이터 오브젝트 중에서 무작위로 뽑힌다. (색칠된 동그라미로 표시됨).



2) k 각 데이터 오브젝트들은 가장 가까이 있는 평균값을 기준으로 묶인다. 평균값을 기준으로 분할된 영역은 보로노이 다이어그램 으로 표시된다..



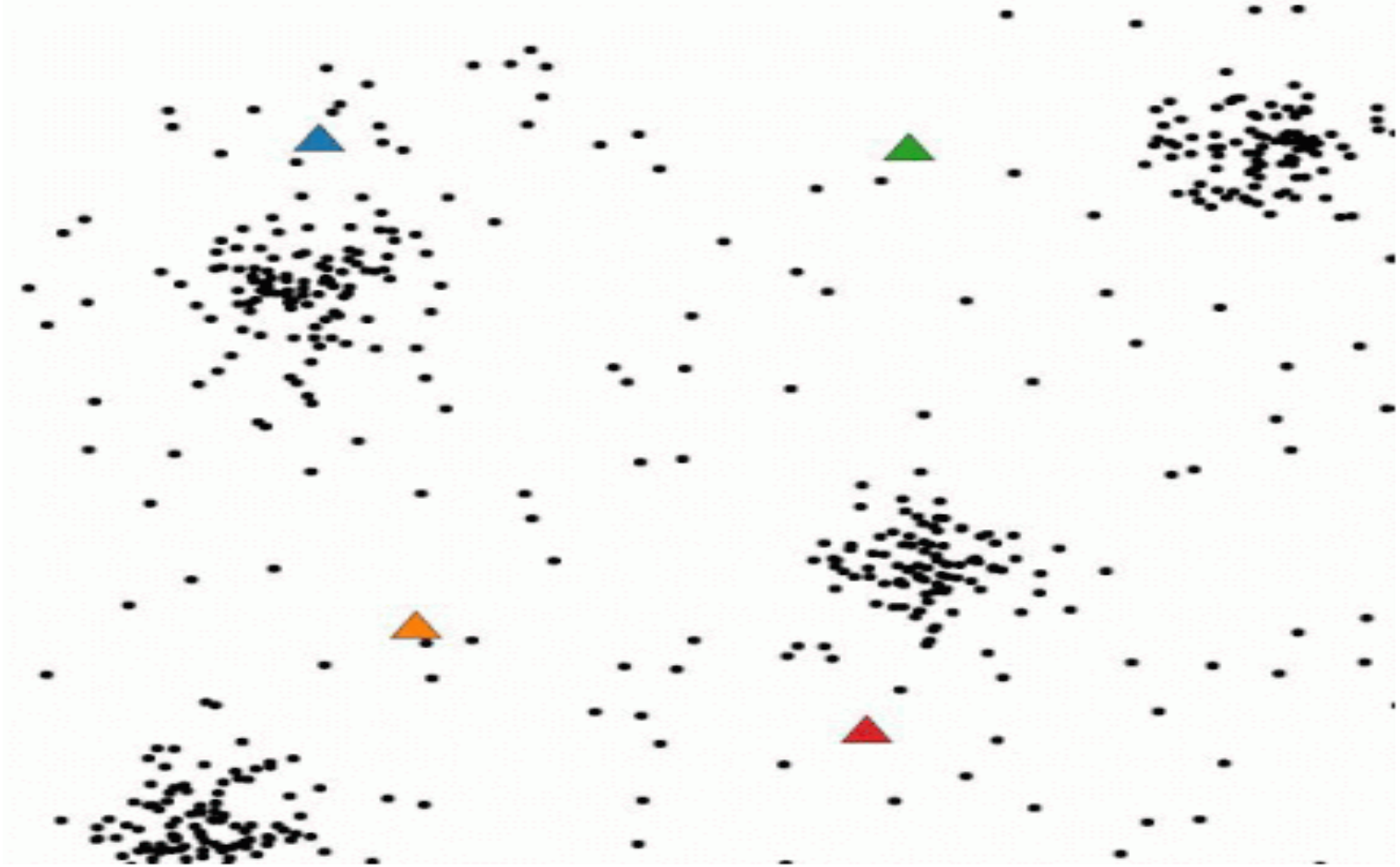
3) k개의 클러스터의 중심점을 기준으로 평균값이 재조정된다.



4) 수렴할 때 까지 2), 3) 과정을 반복한다.

비지도학습 : 군집화

- K- 평균(K-means) 군집화 - https://ko.wikipedia.org/wiki/K-평균_알고리즘



비지도학습 : 군집화

- 실습 예제 : wine dataset

```
1  import os
2  from os.path import join
3  import copy
4  import warnings
5  warnings.filterwarnings('ignore')
6
7  import numpy as np
8  import pandas as pd
9
10 import sklearn
11
12 import matplotlib.pyplot as plt
```

```
1  from sklearn.datasets import load_wine
2  wine = load_wine()
```

이번 군집화 실습을 위해 sklearn 내장 데이터인 와인 데이터를 불러오겠습니다.
와인 데이터셋은 알콜, 말산, 페놀 등 13개의 변수를 가지고 있으며, 1,2,3 와인 등급을 라벨 데이터로 가지고 있습니다.

```
1  print(wine.DESCR)
```

```
.. _wine_dataset:

Wine recognition dataset
```

비지도학습 : 군집화

- 실습 예제 : wine dataset

```
data = wine.data #컬럼 가져오기
label = wine.target #label 가져오기
columns = wine.feature_names
```

```
data = pd.DataFrame(data, columns = columns)
data.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nc
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	

```
data.shape
```

```
(178, 13)
```

```
data.describe()
```

비지도학습 : 군집화

- 실습 예제 : wine dataset

1. K-means Clustering

pca - 스켈링을 먼저 해보고 진행하여야 한다.

```
# PCA
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data = scaler.fit_transform(data)
```

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
data = pca.fit_transform(data)
```

```
data.shape
```

```
(178, 2)
```

1) 모델 불러오기 및 정의하기

클러스터링은 비지도학습이므로 클러스터의 수는 라벨의 수와 관계 없지만, 3개의 군집을 형성하도록 해보겠습니다.
k-means 클러스터링은 sklearn의 cluster 패키지에 있습니다.

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3)
```

비지도학습 : 군집화

- 실습 예제 : wine dataset

1. K-means Clustering

2) 모델 학습하기 (클러스터링을 통한 중심점 찾기)

```
1  kmeans.fit(data)
```

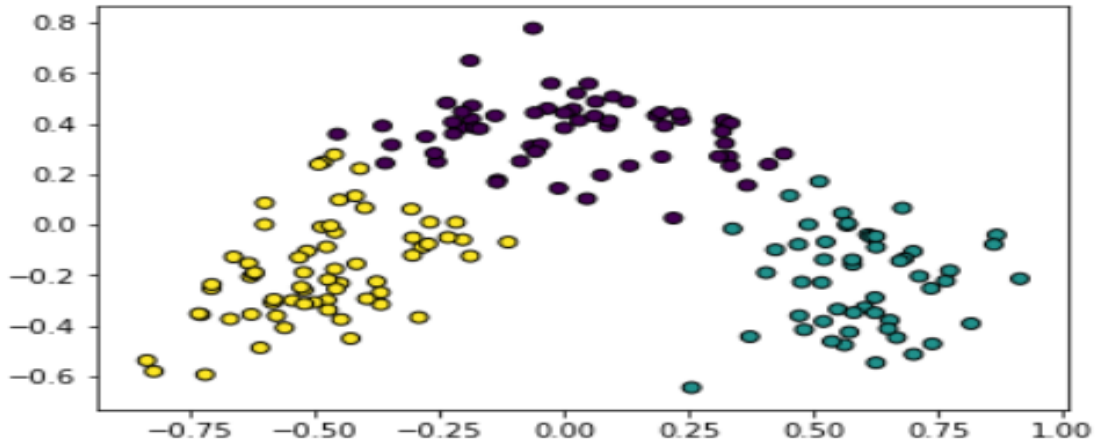
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

3) 클러스터 할당

```
1  cluster = kmeans.predict(data)
```

4) 결과 살펴보기

```
1  plt.scatter(data[:, 0], data[:, 1], c=cluster, linewidth=1, edgecolor='black')
2  plt.show()
```



비지도학습 : 군집화

- 실습 예제 : wine dataset

군집분석의 모델평가(Evaluation) : 실루엣(Silhouette)

- 실루엣 값은 한 클러스터 안의 데이터들이 다른 클러스터와 비교해서 얼마나 비슷한가를 나타냄
- 같은 클러스터 내의 점들간 거리는 가깝고, 서로 다른 클러스터 간의 거리는 멀수록 높은 값을 얻을 수 있다.
- 일반적으로 실루엣 값이 0.5보다 크다면 데이터가 잘 클러스터링 되었다는 것을 나타냄

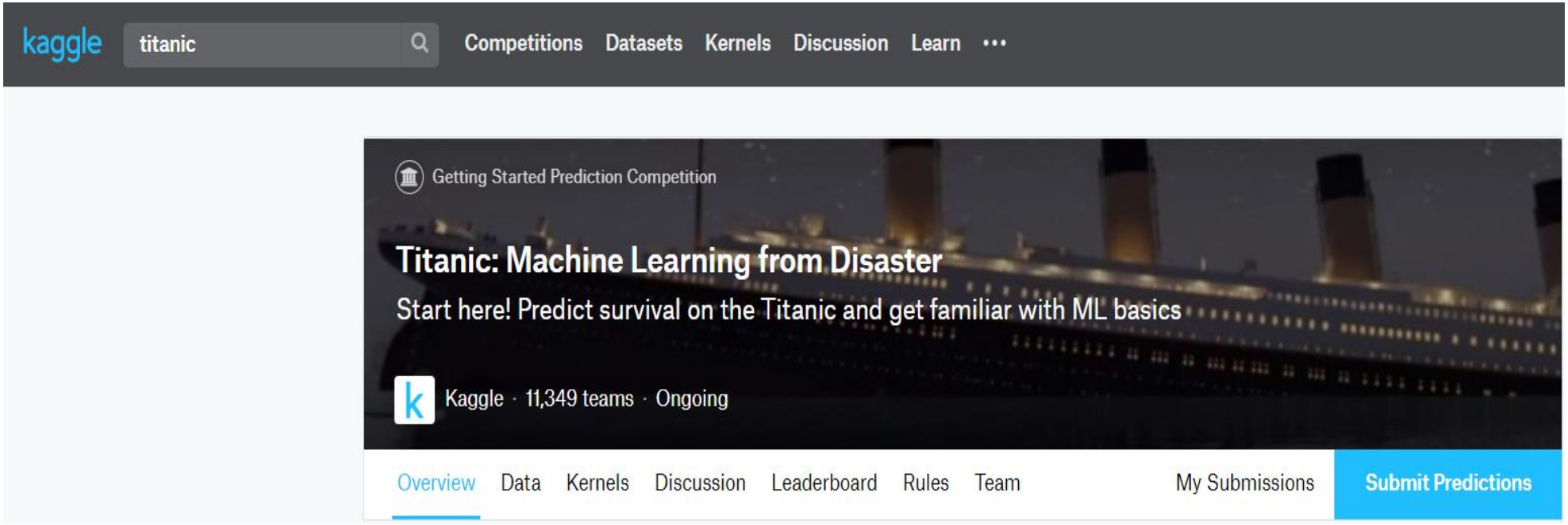
k-means의 실루엣

```
1  from sklearn.metrics import silhouette_score
2
3  best_n = 1
4  best_score = -1
5
6  for n_cluster in range(2, 11):
7      kmeans = KMeans(n_clusters=n_cluster)
8      kmeans.fit(data)
9      cluster = kmeans.predict(data)
10     score = silhouette_score(data, cluster)
11
12     print('클러스터의 수 : {}, 실루엣 점수 : {:.2f}'.format(n_cluster, score))
13     if score > best_score :
14         best_n = n_cluster
15         best_score = score
16
17 print('가장 높은 실루엣 점수를 가진 클러스터 수 : {}, 실루엣 점수 : {:.2f}'.format(best_n, best_score))
18
```

```
클러스터의 수 : 2, 실루엣 점수 : 0.49
클러스터의 수 : 3, 실루엣 점수 : 0.47
클러스터의 수 : 4, 실루엣 점수 : 0.37
클러스터의 수 : 5, 실루엣 점수 : 0.36
클러스터의 수 : 6, 실루엣 점수 : 0.35
클러스터의 수 : 7, 실루엣 점수 : 0.37
클러스터의 수 : 8, 실루엣 점수 : 0.37
클러스터의 수 : 9, 실루엣 점수 : 0.35
클러스터의 수 : 10, 실루엣 점수 : 0.36
가장 높은 실루엣 점수를 가진 클러스터 수 : 2, 실루엣 점수 : 0.49
```

캐글 사이트 소개

- 캐글 사이트 - https://www.Kaggle.com
  - 2010년에 설립된 머신러닝 경진 대회를 여는 사이트
  - 기업과 연계해서 주최
  - 해결하려는 문제와 데이타셋을 제공하면 누구나 참여하여 문재를 해결
  - 막대한 상금과 명예를 위해 다양한 분야의 전문가들이 참여



캐글 사이트 소개

- 캐글 사이트에서 제공하는 서비스

- 캐글을 참여하면서 얻는 것들
  - 다양한 데이타셋과 실제 문제들을 접할 수 있는 환경
  - 캐글에서 성공은 성공적인 데이터 분석 전문가의 길
  - 다양한 디스커션을 통한 커뮤니티
  - 부수입

THANK YOU

마소캠퍼스: masocampus.com
이메일 문의: biz@masocampus.com
전화 문의: 02-6080-2022