

## 1 Support Vector Machine

- one-versus-the-rest (one-against-all)：訓練時，依序將其中一個類別的 data 歸為一類，其他的数据 歸為另一類。如此一來，若有  $k$  個類別，就會有  $k$  個 classifier。分類時，將 data 帶入  $k$  個 classifier，並將該 data 分類為具有最大分類函數值的類別。
- one-versus-one classifier: 在  $k$  個類別中，任選兩類得到一個 classifier。因此，如果有  $k$  個類別，就會得到  $k(k-1)/2$  個 classifier。分類時，將 data 分類為得票最多的類別。

1. 透過 PCA 將 Data 降成 2 維

2. 計算 kernel

- Linear kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$$

- Polynomial (homogeneous) kernel of degree 2:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j)^2$$

$$\phi(\mathbf{x}) = [x_1^2, \sqrt{2}x_1x_2, x_2^2]$$

$$\mathbf{x} = [x_1, x_2]$$

3. 使用 scikit-learn 套件的 SVC 計算 support vector

4. 計算出 Lagrange multiplier 及 target

5. SVM 使用 one-versus-the-rest 來分類，總共 3 次，所以會有 3 個 classifier。首先計算出 weight 與 bias，接著就可算出  $y$

$$w = \sum_{n=1}^N a_n t_n \phi(x_n)$$

$$b = \frac{1}{N} \sum_{n \in S} \left( t_n - \sum_{m \in S} a_m t_m k(x_n, x_m) \right)$$

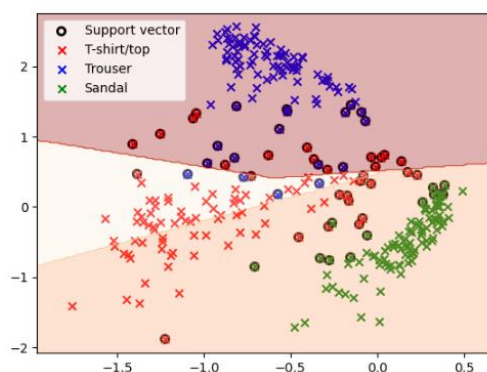
$$y(x) = w^\top \phi(x) + b$$

總共求出 3 條 decision boundary

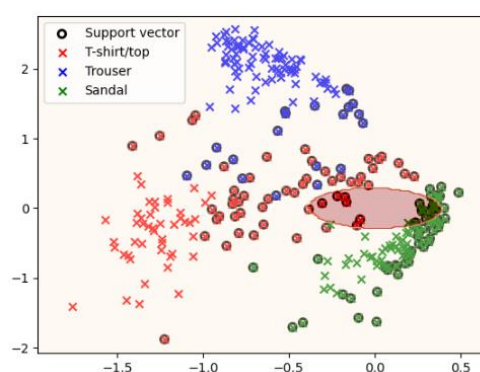
### 1.2

我使用 one-versus-one 作為我的 decision strategy，因為分類多個類別的資料(multiclass)，假設有  $n$  個類別，可以一次找出其中一個類別與其他類別的邊界，如此一來就可以一次訓練好模型，在預測的時候比較方便。

### 1.3 & Bonus



Linear Kernel



Polynomial Kernel (degree = 2)

## 2 Gaussian Mixture Model

在 dataset 中隨機找 K 筆資料作為初始群集的中心，並計算每一筆 data 到 K 個群集中心的距離，並將該筆 data 分類給與之最接近的群集中心，並產生一個群集邊界，在計算每個群集新的中心，在利用新群集中心計算距離，透過這樣的迭代方式，直到群集成員不再變動。

Expectation Maximization(EM) algorithm 可以用來解決存在隱藏資訊最佳化問題，將 K-means 計算出 mean、variance 與 mixing coefficient 等參數，在經由 EM algorithm 最佳化。

### 2.1~2.3

---- K-means ----

K=2

	R	G	B
0	177	165	147
1	35	68	37

K=3

	R	G	B
0	24	54	23
1	84	119	90
2	201	177	163

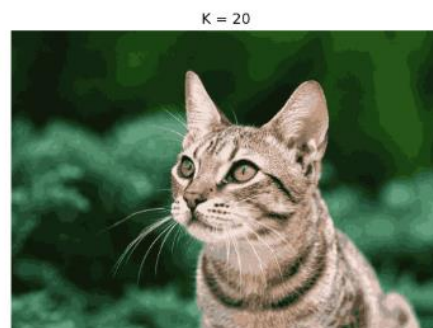
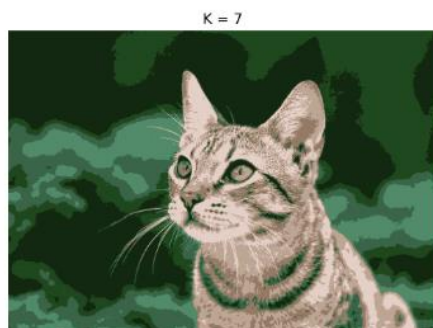
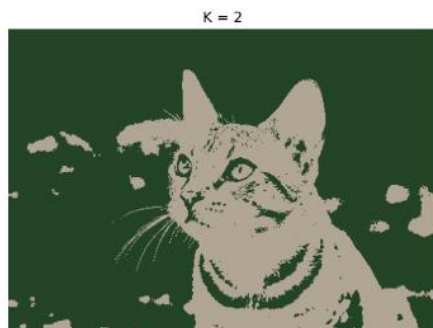
K=7

	R	G	B
0	59	101	72
1	190	166	149
2	149	123	102
3	18	40	16
4	81	139	105
5	229	208	199
6	32	72	30

K=20

	R	G	B
0	21	51	17
1	55	109	77
2	100	157	125
3	68	124	91
4	34	76	49
5	38	85	24
6	90	76	56
7	192	154	129
8	17	38	17
9	44	94	64
10	26	67	19
11	159	131	109
12	202	177	163
13	81	141	105
14	28	57	34
15	243	225	218
16	222	199	189
17	15	26	12
18	126	106	85
19	166	158	149

Kmeans



----- GMM -----

K=2

	R	G	B
0	34	72	40
1	173	152	136

K=3

	R	G	B
0	172	151	135
1	49	96	66
2	22	51	16

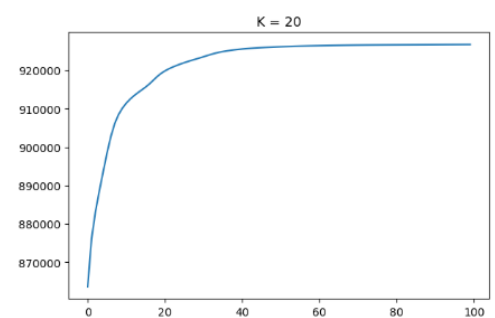
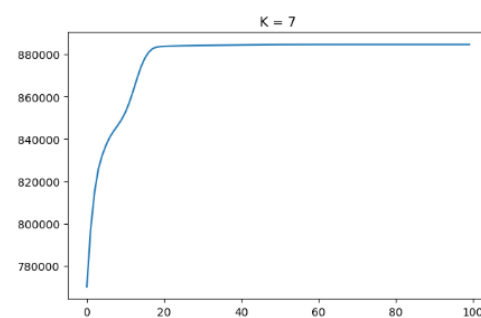
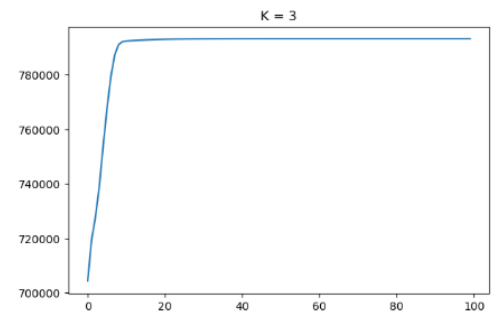
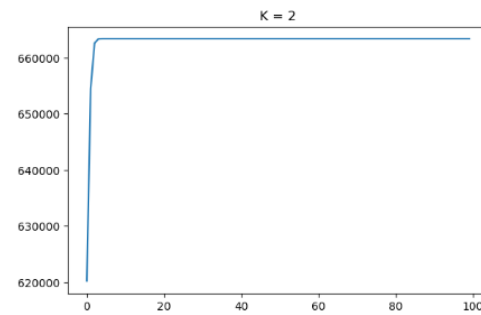
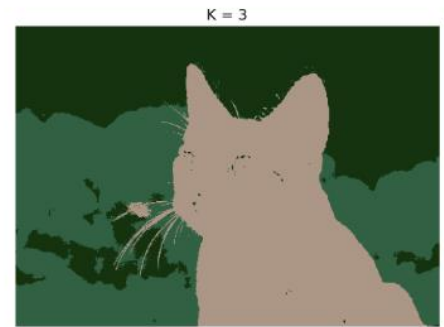
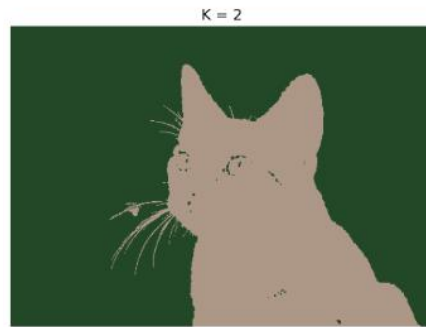
K=7

	R	G	B
0	134	123	109
1	19	41	23
2	60	115	81
3	20	48	15
4	210	195	191
5	34	79	22
6	174	150	128

K=20

	R	G	B
0	43	90	62
1	127	107	83
2	56	111	79
3	81	141	100
4	71	127	89
5	172	151	135
6	18	39	18
7	73	136	100
8	139	128	115
9	32	52	31
10	15	31	19
11	15	30	11
12	242	221	213
13	204	164	148
14	207	183	165
15	26	63	38
16	35	80	22
17	23	60	16
18	203	190	188
19	18	45	15

GMM



## 2.4

當 K 值變大，分割的類別變多，圖像變的更清晰。由實驗結果觀察，K-means 可以更好保留原圖的資訊，細節保留的比較完整(例如:貓毛)。GMM 則比較容易看出各個群集的颜色(例如:輪廓)。