一、 下載資料集

```
from sklearn.datasets import load_wine
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np

data = load_wine()
data
```

二、 選擇特徵欄位與預測目標欄位

```
feature = pd.DataFrame(data['data'], columns = data['feature_names'])
target = pd.DataFrame(data['target'], columns = ['class'])

df = pd.concat([feature[['alcohol','malic_acid']], target], axis = 1)
df
```

executed in 32ms, finished 00:27:54 2020-12-07

	alcohol	malic_acid	class
0	14.23	1.71	0
1	13.20	1.78	0
2	13.16	2.36	0
3	14.37	1.95	0
4	13.24	2.59	0
...	...	...	...
173	13.71	5.65	2
174	13.40	3.91	2
175	13.27	4.28	2
176	13.17	2.59	2
177	14.13	4.10	2

178 rows × 3 columns

三、 切割訓練&測試資料集，並將資料及標準化

→ train_test_split(X,y,test_size = 0.2, random_state= 50, stratify=y)

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X = df.iloc[:,:2].values
y = df.iloc[:,2].values

# X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.2, random_state= 50)
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.2, random_state= 50, stratify=y)

sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
```

四、 模型設定與結果

1. kernel = ‘linear’

✓ **模型與參數設定：**

```
svm = SVC(kernel = 'linear')
svm.fit(X_train_std,y_train)
```

executed in 9ms, finished 00:27:56 2020-12-07

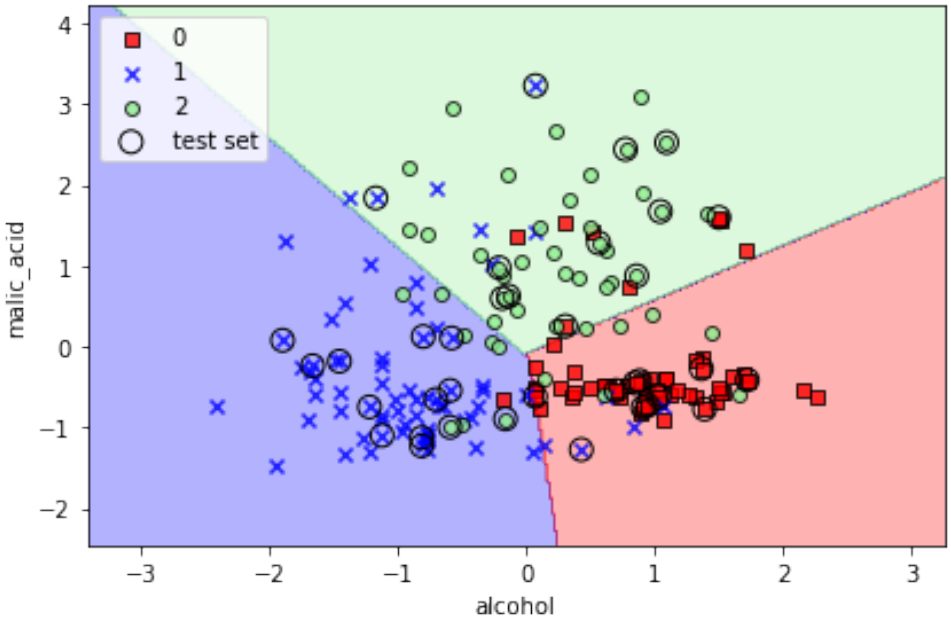
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

✓ **準確度：**

trian: 0.8028169014084507

test: 0.8055555555555556

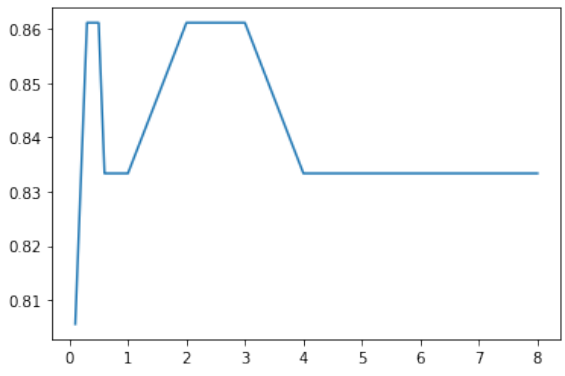
✓ **決策區域圖：**



2. kernel = ‘ rbf ‘

✓ **調參情況 (gamma):**

→ gamma = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,3,4,5,6,7,8]




```
[0.8055555555555556,
 0.8333333333333334,
 0.8611111111111112,
 0.8611111111111112,
 0.8611111111111112,
 0.8333333333333334,
 0.8333333333333334,
 0.8333333333333334,
 0.8333333333333334,
 0.8333333333333334,
 0.8611111111111112,
 0.8611111111111112,
 0.8333333333333334,
 0.8333333333333334,
 0.8333333333333334,
 0.8333333333333334,
 0.8333333333333334]
```

✓ **模型與參數設定：(gamma = 0.3)**

```
1  svm_rbf = SVC(kernel = 'rbf', gamma = 0.3)
2  svm_rbf.fit(X_train_std,y_train)
```

executed in 10ms, finished 00:50:11 2020-12-07

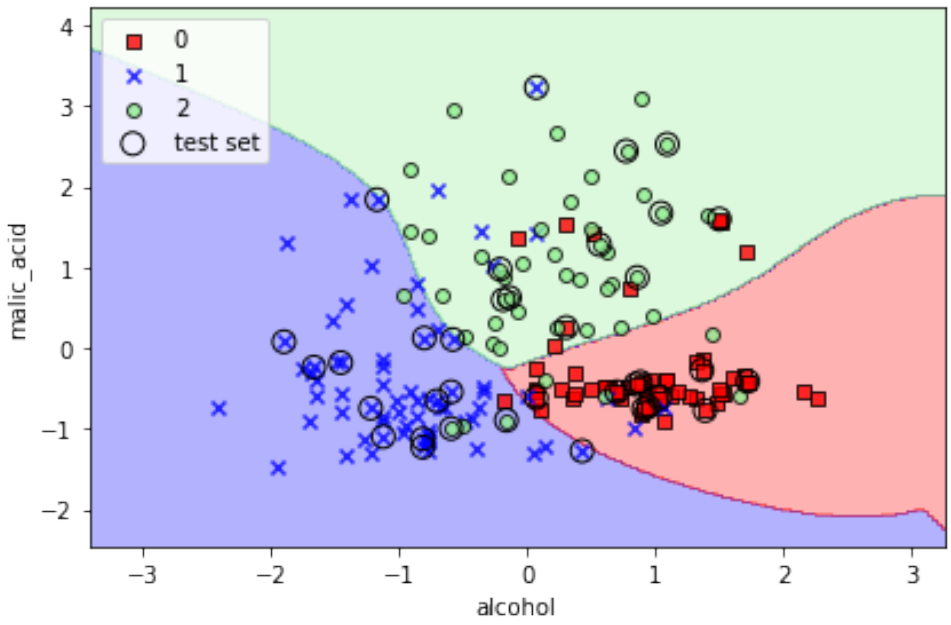
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.3, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

✓ **準確度：**

trian: 0.8450704225352113

test: 0.8611111111111112

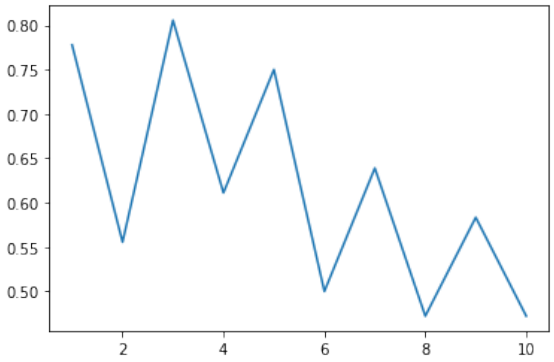
✓ **決策區域圖：**



3. kernel = ‘ poly ‘

✓ **調參情況 (degree):**

→ degree = [1,2,3,4,5,6,7,8,9,10]




```
[0.7777777777777778,
 0.5555555555555556,
 0.8055555555555556,
 0.6111111111111112,
 0.75,
 0.5,
 0.6388888888888888,
 0.4722222222222222,
 0.5833333333333334,
 0.4722222222222222]
```

✓ **模型與參數設定：(degree = 3)**

```
1  svm_poly = SVC(kernel = 'poly', degree = 3)
2  svm_poly.fit(X_train_std,y_train)
```

executed in 12ms, finished 00:52:07 2020-12-07

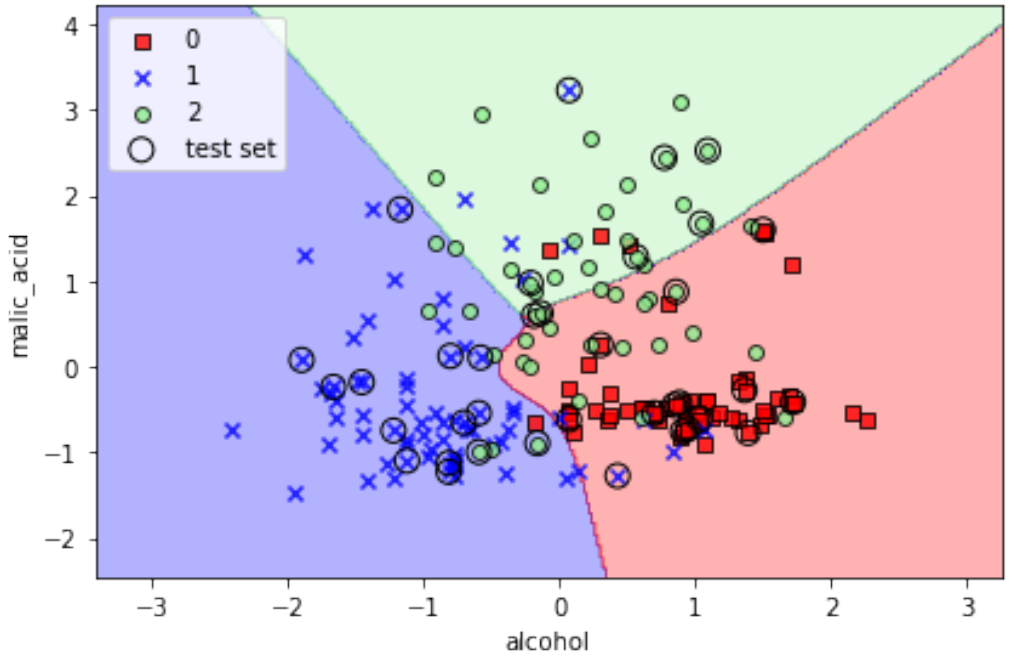
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='poly', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

✓ **準確度：**

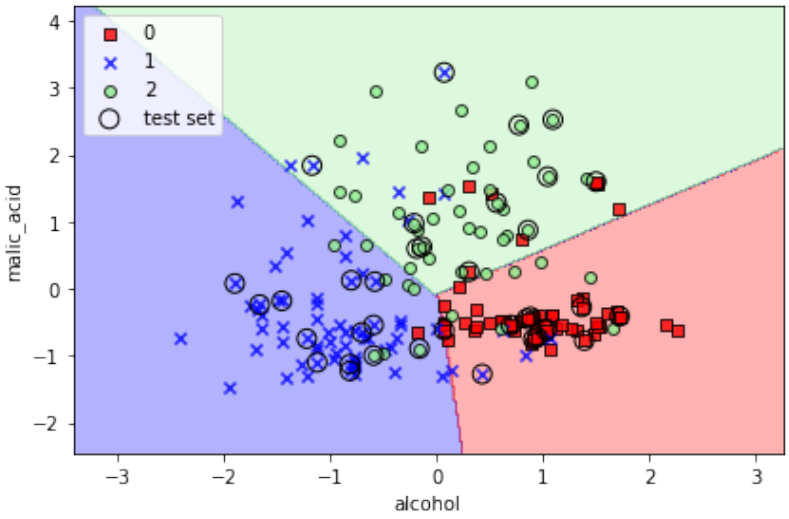
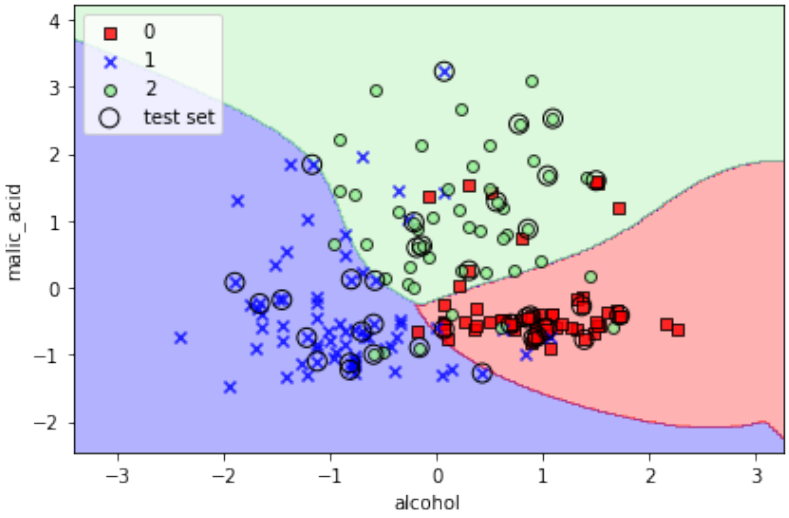
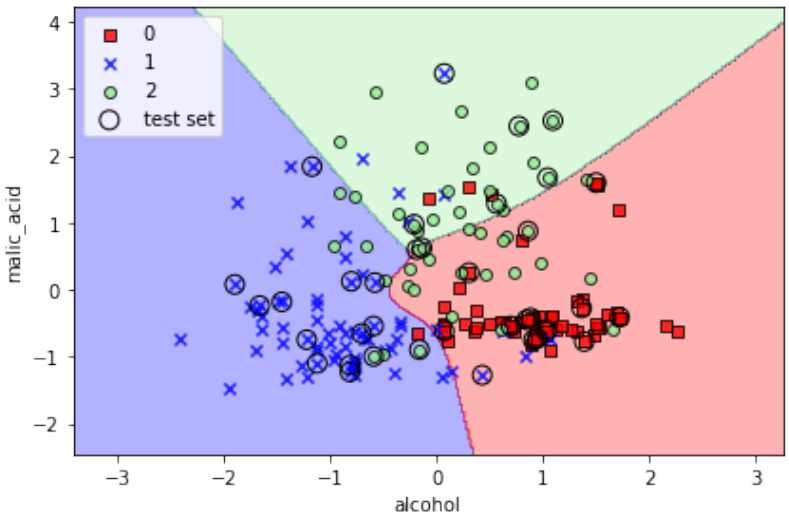
trian: 0.7464788732394366

test: 0.8055555555555556

✓ **決策區域圖：**



五、 分析與比較

核函數	參數設定	Train accuracy	Test accuracy	決策圖
linear	預設	0.8028	0.8056	
rbf	gamma = 0.3	0.8451	0.8611	
poly	degree = 3	0.7465	0.8056	

→分析：

Gamma 是選擇核函數 kernel 為 RBF 時，對模型影響性極大的一個重要參數，他決定了數據映射到新特徵空間後的分佈。當 Gamma 越大，支持向量越少，從高斯分佈來看，會長得又高又瘦，會造成模型只學習支持向量附近的樣本，對於未知的樣本分類效果很差，而發生 Train Accuracy 很高、Test Accuracy 很低的過擬合情況；而 Gamma 越小，支持向量越多，從高斯分佈來看，會使分佈過於平滑，無法在訓練時提高準確率，進而影響測試時的準確度。而 Gamma 值所控制的支持向量的個數，則會影響模型訓練與預測的執行速度。

Degree 參數則是在選用 kernel 為 poly（多項式核函數）時，決定模型之多項式的最高次冪。

而從數學理論與實驗分析結果來看，核函數 rbf 是最有機會也最可能接近線性 linear 的情況的，決策區域圖也顯示出兩者訓練與預測時非常相似的情況，因 gamma 僅設置為 0.3，與預設情況相差不多，模型在訓練時微調了決策邊界，以至於學習（訓練）狀況與預測狀況都較線性核函數稍微好一些。而 poly 核函數的表現則為三者中最差的，雖然模型預測的效果與線性核函數差不多，但在訓練時的表現卻遜色了些。

→ 結論：

在本次實驗中可以看出，若只調整 rbf 核函數模型中的 gamma 參數，與 poly 核函數模型中的 degree 參數，模型的表現為：rbf > linear > poly，且核函數為 rbf 與 linear 的模型，在訓練與測試時的準確度相近，模型表現較趨於穩定良好的狀態。