به نام خدا

تمرین سری 4

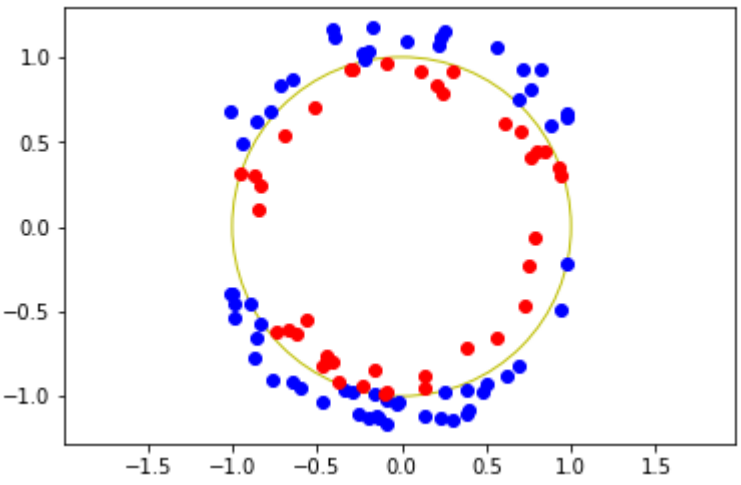
یادگیری ماشین

سوال 1

ابتدا اطلاعات را تخمین زده و وارد مسئله میکنیم

با acc میزان پخش شدن ، seed نوع عدد رندم ، n تعداد عدد عدد رندم

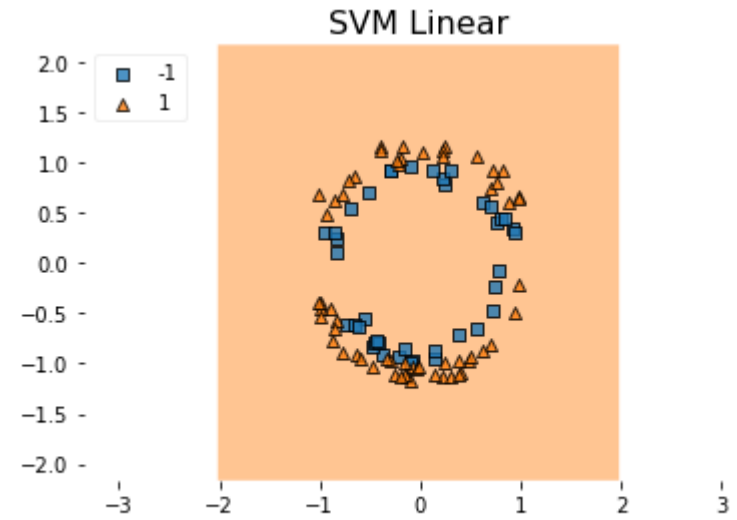
```python
#prepare test train
#train_DATA
n=100 #number of train
xp,yp,xm,ym=get_list(n,0.2,4)
xx=xp+xm
yy=yp+ym
Xl=list()
for i in range(n):
    Xl.append([xx[i],yy[i]])
X = np.asarray(Xl, dtype=np.float32)
y1=[1 for x in range(len(xp))]
y2=[-1 for x in range(len(xm))]
y= np.asarray(y1+y2, dtype=np.integer)
#train_DATA
T=10 #number of train
xp,yp,xm,ym=get_list(n,0.2,4)
xx=xp+xm
yy=yp+ym
Xl=list()
for i in range(n):
    Xl.append([xx[i],yy[i]])
X_test = np.asarray(Xl, dtype=np.float32)
y1=[1 for x in range(len(xp))]
y2=[-1 for x in range(len(xm))]
y_test= np.asarray(y1+y2, dtype=np.integer)
```




```python
def get_list(n,acc,seeds):
    ''' data generation function
    n is number of sample
    acc is as like as variance
    seeds is the random pack seed'''
    from random import random
    from random import seed
    from math import sqrt
    seed(seeds)
    X=list()
    Y=list()
    xm=list()
    ym=list()
    xp=list()
    yp=list()
    for i in range(0,n):
        x=random()
        y=sqrt( 1 - (x**2) )
        signx=int( random()*2 )
        x=(-1)*x if signx==1 else x
        signy=int( random()*2 )
        y=(-1)*y if signy==1 else y
        sign_acc=int( random()*2 )
        A=(-1)*acc if sign_acc==1 else acc
        X.append(x+random()*A)
        sign_acc=int( random()*2 )
        A=(-1)*acc if sign_acc==1 else acc
        Y.append(y+random()*A)
        if (X[i]**2+Y[i]**2)<1 :
            xm.append(X[i])
            ym.append(Y[i])
        else:
            xp.append(X[i])
            yp.append(Y[i])
    return xp,yp,xm,ym
```

الف) در بخش اول سوال svm خطی بدون ، خطا را میخواد ، پر واضح است که نمیتوان این کار را انجام داد که با تغییر ، پارامتر ها نتیجه تغییری نخواهد کرد ، با این حال cros_validation انجام شد و دقت محاسبه شد.

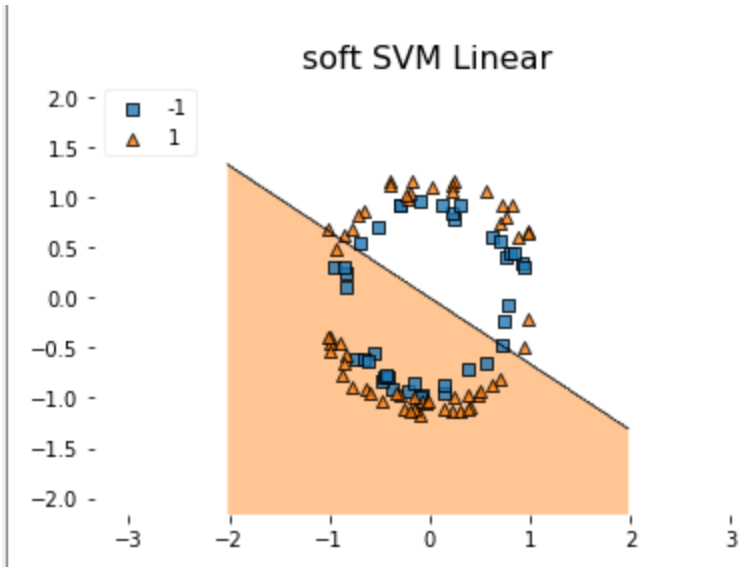
نتیجه:




```
Accuracy: 0.46 (+/- 0.02)
```

```python
#1-a
clf1 = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto',
               coef0=0.0, shrinking=True, probability=False, tol=0.001,
               cache_size=200, class_weight=None, verbose=False, max_iter=-1,
               decision_function_shape='ovr', random_state=None)
clf1.fit(X,y)
scores = cross_val_score(clf1, X , y , cv=3)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
plot_decision_regions(X=X, y=y, clf=clf1, legend=2)
plt.title('SVM Linear', size=16)
plt.axis("equal")
plt.show()
```

ب) در این بخش ، سوال از ما svm خطی به صورت soft را میخواهد به این معنی که خطا را میتوان پذیرش کرد ، برای یافتن بهرین پارامتر از روی داده یادگیری ، kFold_cross validation با k=5 انجام شد و بهترین نتیجه 62 درصد حاصل شد.



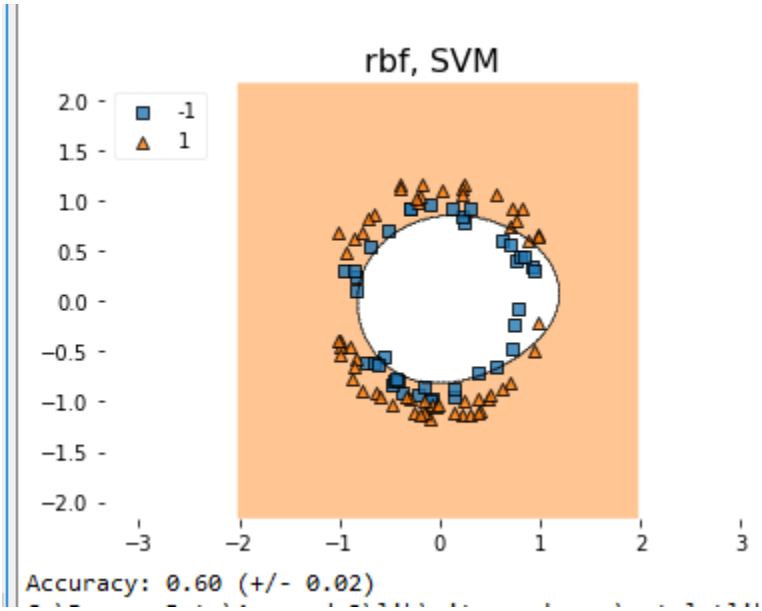
```
#1-b
clf2 = svm.LinearSVC(penalty='l2', loss='squared_hinge',
                     dual=True, tol=0.00001, C=1.0, multi_class='crammer_singer',
                     fit_intercept=False, intercept_scaling=2,
                     class_weight=None, verbose=0, random_state=0,
                     max_iter=2000)

clf2.fit(X,y)
scores = cross_val_score(clf2, X , y , cv=3)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
plot_decision_regions(X=X, y=y, clf=clf2, legend=2)
plt.title('soft SVM Linear', size=16)
plt.axis("equal")
plt.show()
```

ج) در ادامه سوال پاسخ را برای تابع کرنل rbf ,poly درجه 2 و 3 میخواهد که به شرح زیر است:

```
#1-c
clf3 = svm.SVC(C=1.0, kernel='rbf', degree=3, gamma='auto',
               coef0=0.0, shrinking=True, probability=False, tol=0.001,
               cache_size=200, class_weight=None, verbose=False, max_iter=-1,
               decision_function_shape='ovr', random_state=None)

clf3.fit(X,y)
scores = cross_val_score(clf3, X , y , cv=3)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
plot_decision_regions(X=X, y=y, clf=clf3, legend=2)
plt.title('rbf, SVM ', size=16)
plt.axis("equal")
plt.show()
```

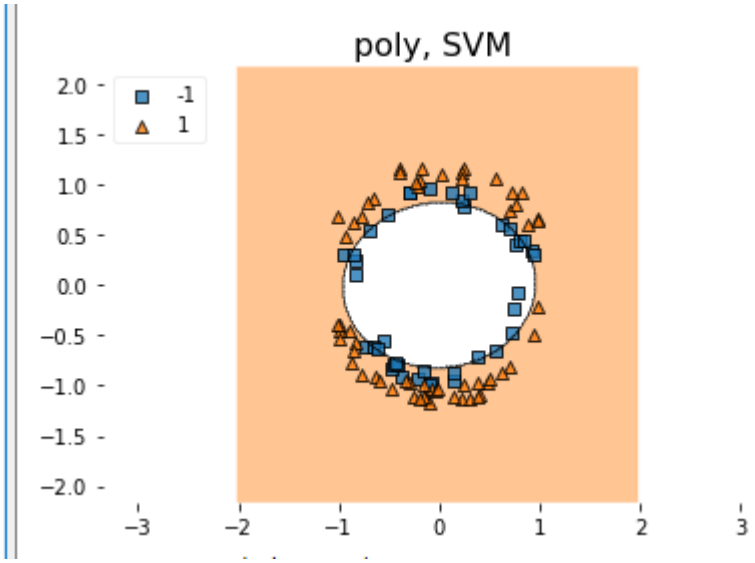



```
Accuracy: 0.60 (+/- 0.02)
```

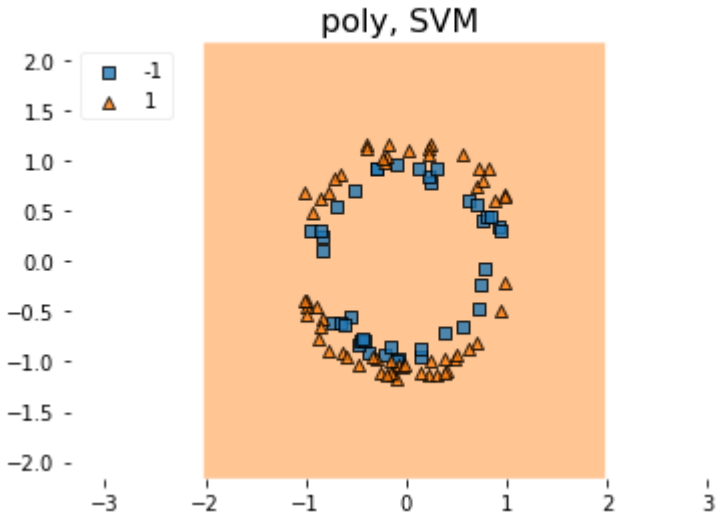
کرنل درجه 2

```
#1-d
clf4 = svm.SVC(C=1.0, kernel='poly', degree=2, gamma='auto',
               coef0=0.0, shrinking=True, probability=False, tol=0.001,
               cache_size=200, class_weight=None, verbose=False, max_iter=-1,
               decision_function_shape='ovr', random_state=None)

clf4.fit(X,y)
scores = cross_val_score(clf4, X , y , cv=3)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
plot_decision_regions(X=X, y=y, clf=clf4, legend=2)
plt.title('poly, SVM ', size=16)
plt.axis("equal")
```



کرنل درجه 3



مشاهده میشود که نمیتواند محسابه کند چرا که ماهیت دایره تابعی درجه 2 است و با افزایش درجه خراب میشود.

سوال2

در تمرین شماره 2 نحوه تبدیل اطلاعات به csv توضیح داده شد. از همان دیتا فریم ساخته شده پانداس استفاده شد.

برای سوال 2 ابتدا به دلیل ماهیت categorical بودن اطلاعات باید توسط تابعی ، به اعداد ترجمه شوند تا بتوان از کتابخانه sklearn استفاده شود. برای این کار از مجموعه توابع preprocessing و توسط label enconding ، و تابع زیر ، نوع دیتا را عوض میکنیم

```
def transformer (dataframe):
    import numpy as np
    from sklearn import preprocessing
    le = preprocessing.LabelEncoder()
    le.fit(list(ascii_lowercase))
    a=X_train.values.tolist()
    x=list()
    for i in range(len(a)):
        x.append(le.transform(a[i]))
        X=np.asarray(x, dtype=np.integer)
    return X
```

در بخش اول سوال 2 از ما استفاده از svm خطی با تخمین soft است. که با warning همگرا نشدن الگوریتم مواجه میشویم. که با افزایش iterations نیز مشکل حل نمیشود

```
#start learning part
clf = svm.LinearSVC(penalty='l2', loss='squared_hinge',
                    dual=True, tol=0.00001, C=1.0 , fit_intercept=False,
                    intercept_scaling=2,class_weight=None, verbose=0,
                    random_state=0,max_iter=10000000)

clf.fit(Xt_train,y_train)
```

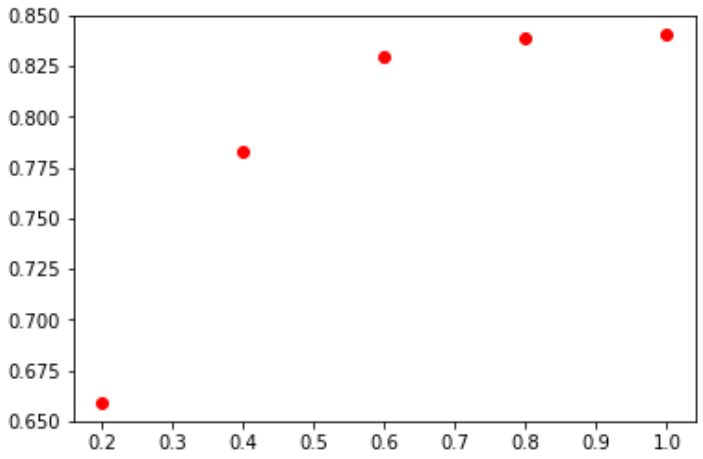
```
In [20]: runfile('C:/Users/Dell/Desktop/HW4-2.py', wdir='C:/Users/Dell/Desktop')
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:931: ConvergenceWarning: Liblinear
failed to converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
```

در بخش دوم میزان تغیرات c را در آموزش مد نظر دارد:

```
result=list()
for i in range(5):
    clf = svm.SVC(C=1.0-i/5, kernel='rbf', degree=3, gamma='auto',
                  coef0=0.0, shrinking=True, probability=False, tol=0.0001,
                  cache_size=200, class_weight=None, verbose=False, max_iter=-1,
                  decision_function_shape='ovr', random_state=None)
    clf.fit(Xt_train,y_train)

    result.append(test(Xt_valid,y_valid,clf))
import matplotlib.pyplot as plt
c=[0.2,0.4,0.6,0.8,1]
plt.plot(c,result,'ro')
```

```
def test(X,Y,clf):
    A=list()
    x=list(clf.predict(X))
    y=list(Y)
    for i in range(len(X)):
        if(x[i]==y[i]):
            A.append(1)
        else:
            A.append(0)
    return sum(A)/len(A)
```



که با نزدیک شدن c به 1 افزایش میابد و میزان دقت روی داده validation به ازای c=1 ، 0.8405197873597164, می باشد و به ازای بررسی روی داده تست : 0.8524203069657615, می باشد.

در بخش سوم ، تفاوت نتیجه به ازای چند جمله ای و rbf را میپرسد که دقت rbf روی داده validation برابر :

```
#start learning part
clf = svm.SVC(C=1.0, kernel='rbf', degree=3, gamma='auto',
                  coef0=0.0, shrinking=True, probability=False, tol=0.0001,
                  cache_size=200, class_weight=None, verbose=False, max_iter=-1,
                  decision_function_shape='ovr', random_state=None)
clf.fit(Xt_train,y_train)

print('the acc is :' , test(Xt_valid,y_valid,clf))
```

```
In [33]: runfile('C:/Users/Dell/Desktop/HW4-2-1.py', wdir='C:/Users/Dell/Desktop')
the acc is : 0.8405197873597164
```

و برای چند جمله ای در داده validation ، به طور کلی الگوریتم همگرا نمیشود و باید ittration را محدود کنیم ، با بررسی چند حالت ، بهترین نتیجه را میتوان از ، درجه4 با max_iter= 5000گرفت که مشاهده میشود:

```
#start learning part
clf = svm.SVC(C=1.0, kernel='poly', degree=4, gamma='auto',
                  coef0=0.0, shrinking=True, probability=False, tol=0.00001,
                  cache_size=100000, class_weight=None, verbose=False, max_iter=5000,
                  decision_function_shape='ovr', random_state=None)
clf.fit(Xt_train,y_train)

print('the acc is :' , test(Xt_valid,y_valid,clf))
```

```
In [10]: runfile('C:/Users/Dell/Desktop/HW4-2-1.py', wdir='C:/Users/Dell/Desktop')
the acc is : 0.6916715888954519
C:\Anaconda3\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver
terminated early (max_iter=5000).  Consider pre-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

نتیجه بهتر روی داده validation برای rbf است که نتیجه روی داده تست میدهد:

```
In [14]: runfile('C:/Users/Dell/Desktop/HW4-2-1.py', wdir='C:/Users/Dell/Desktop')
the acc is : 0.8524203069657615
```

```
#start learning part
clf = svm.SVC(C=1.0, kernel='rbf', degree=4, gamma='auto',
                  coef0=0.0, shrinking=True, probability=False, tol=0.00001,
                  cache_size=100000, class_weight=None, verbose=False, max_iter=-1,
                  decision_function_shape='ovr', random_state=None)
clf.fit(Xt_train,y_train)

print('the acc is :' , test(Xt_test,y_test,clf))
```

می باشد.

نکته جالب این است که این درصد ها در scale درصد های حاصل شده از درخت پس از یادگیری و حرص کردن است.