ballons_SVC

April 24, 2018

```
In [69]: import warnings
         warnings.filterwarnings('ignore')
         import seaborn as sns
         sns.set(style="whitegrid", color_codes=True)
         import pandas as pd
         import numpy as np
         from sklearn import svm #not decision tree
         from sklearn.model_selection import train_test_split
         from sklearn.preprocessing import LabelEncoder
```

```
In [3]: name = ['color','size','act','age','inflated']
```

```
In [58]: df1 = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/balloons/a
         df2 = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/balloons/a
         df3 = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/balloons/y
         df4 = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/balloons/y

         frames = [df1,df2,df3,df4]
         df = pd.concat(frames)
```

```
In [59]: df.describe()
```

```
Out[59]:          color   size  act    age inflated
         count       76     76   76     76       76
         unique       2      2    2      2        2
         top     YELLOW  SMALL  DIP  CHILD        F
         freq        40     40   38     38       41
```

```
In [60]: df.tail()
```

```
Out[60]:      color   size      act    age inflated
         11  PURPLE  SMALL      DIP  CHILD        F
         12  PURPLE  LARGE  STRETCH  ADULT        T
         13  PURPLE  LARGE  STRETCH  CHILD        F
         14  PURPLE  LARGE      DIP  ADULT        F
         15  PURPLE  LARGE      DIP  CHILD        F
```

```
In [61]: y = df['inflated']
         x = df
         x= x.drop('inflated',axis=1)
         x.head()
```

```
Out[61]:     color   size      act    age
         0  YELLOW  SMALL  STRETCH  ADULT
         1  YELLOW  SMALL  STRETCH  ADULT
         2  YELLOW  SMALL  STRETCH  CHILD
         3  YELLOW  SMALL      DIP  ADULT
         4  YELLOW  SMALL      DIP  CHILD
```

```
In [62]: x = pd.get_dummies(x)
         x.head()
```

```
Out[62]:    color_PURPLE  color_YELLOW  size_LARGE  size_SMALL  act_DIP  act_STRETCH  \
         0             0             1           0           1        0            1
         1             0             1           0           1        0            1
         2             0             1           0           1        0            1
         3             0             1           0           1        1            0
         4             0             1           0           1        1            0

            age_ADULT  age_CHILD
         0          1          0
         1          1          0
         2          0          1
         3          1          0
         4          0          1
```

```
In [63]: le  = LabelEncoder()
         le.fit(y)
         y = le.transform(y)
         y
```

```
Out[63]: array([1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1,
                1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,
                1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0,
                0, 0, 1, 0, 0, 0, 1, 0, 0, 0])
```

```
In [64]: x_train,x_test,y_train,y_test=train_test_split(x,y, random_state=7)
```

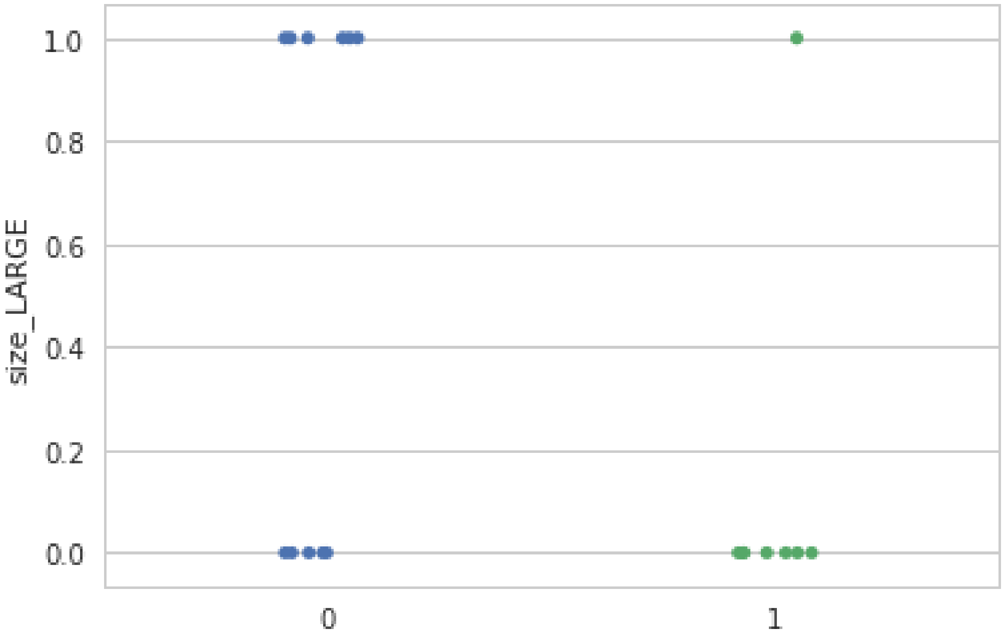
0.1 Score varies with diffrent samples but with tunig paramter C=0.2 and gamma=0.3 with default kernel('rbf') better score got then without any paramter. And with kernel('poly') same happens.

0.2 sometimes C=0.3 and gamme=0.8 gives same as default and diifferent values gives lower/variation in result. Things is need to check different values.

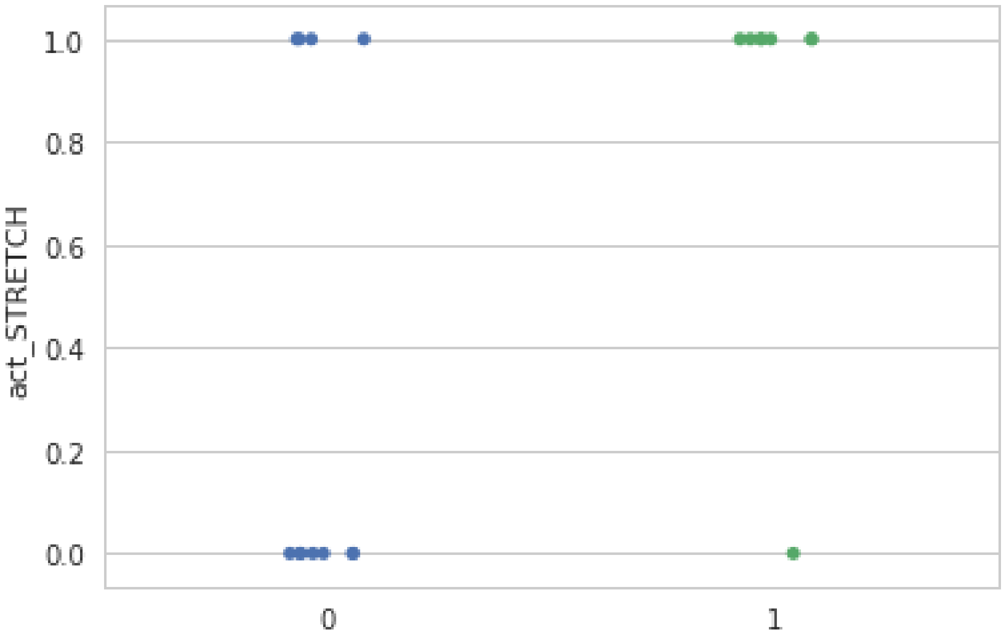
```
In [75]: # svc = svm.SVC()
         svc = svm.SVC(C=0.2,gamma=0.3)
         svc.fit(x_train,y_train)
```

```
Out[75]: SVC(C=0.2, 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)
```

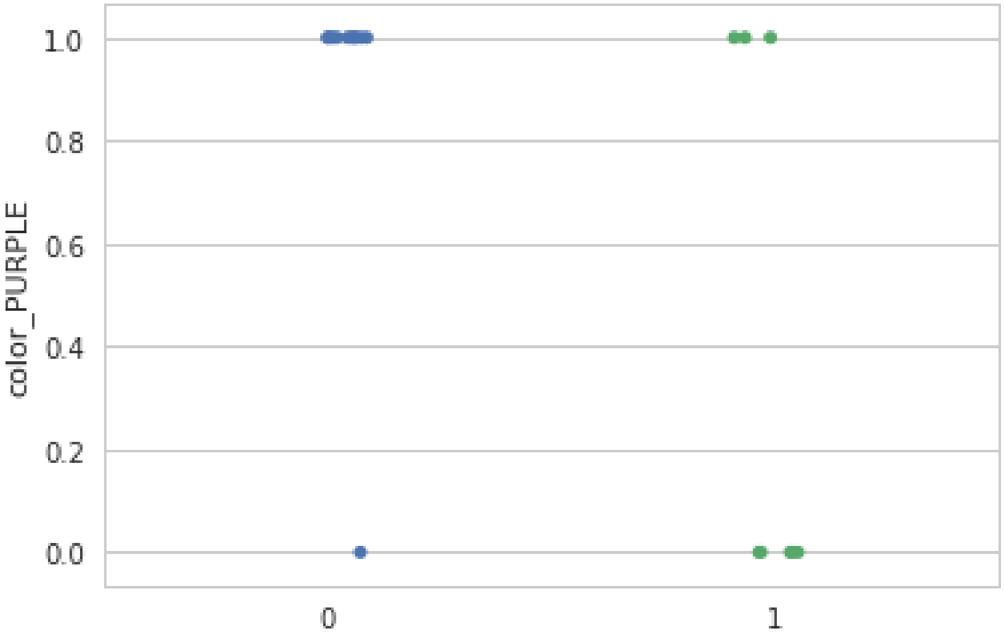
```
In [76]: svc.score(x_test,y_test)

Out[76]: 0.7368421052631579

In [77]: predicted = svc.predict(x_test)
         predicted

Out[77]: array([0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1])

In [78]: mse = np.mean((predicted-y_test)**2)
         mse

Out[78]: 0.2631578947368421

In [96]: # size and inflated relation
         sns.stripplot(x=predicted,y=x_test['size_LARGE'],jitter=True)

Out[96]: <matplotlib.axes._subplots.AxesSubplot at 0x7fabd64b7240>
```



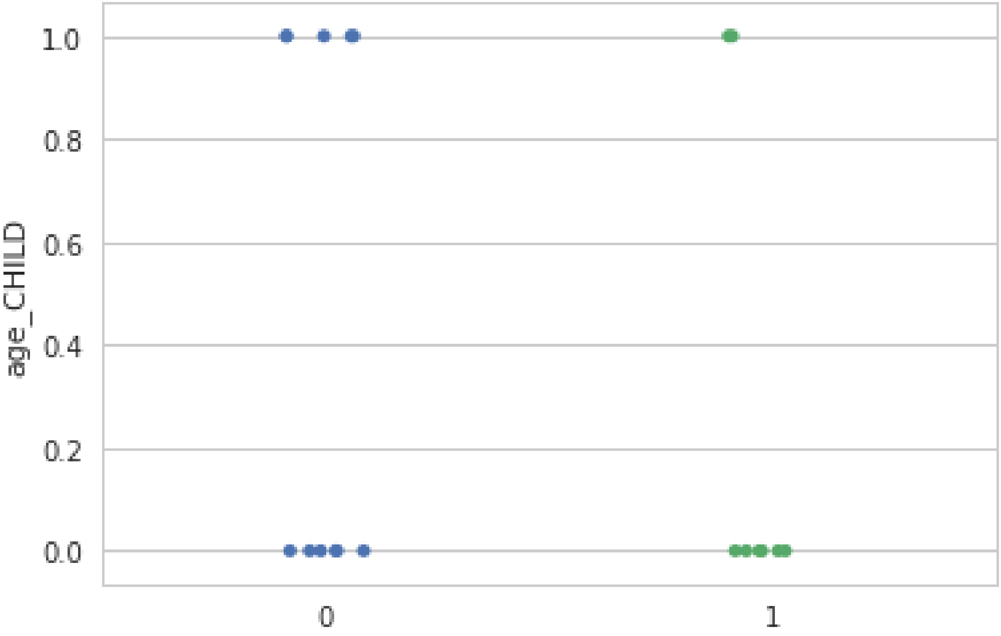
```
In [82]: # act(dim/stretch) and inflated relation
         sns.stripplot(x=predicted,y=x_test['act_STRETCH'],jitter=True)

Out[82]: <matplotlib.axes._subplots.AxesSubplot at 0x7fabd68c0128>
```



```
In [87]: # color(purple/yellow) and inflated relation
         sns.stripplot(x=predicted,y=x_test['color_PURPLE'],jitter=True)

Out[87]: <matplotlib.axes._subplots.AxesSubplot at 0x7fabd6737d30>
```



```
In [90]: # age and inflated relation
         sns.stripplot(x=predicted,y=x_test['age_CHILD'],jitter=True)

Out[90]: <matplotlib.axes._subplots.AxesSubplot at 0x7fabd6660080>
```


age_CHILD