```
#Drive mount
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
import pandas as pd
from sklearn.svm import SVC
import numpy as np
import matplotlib.pyplot as plt
import os                           #files and dictionary path define
from skimage.io import imread        #io == input output
from sklearn.model_selection import train_test_split
from skimage.transform import resize
```

```
os.listdir('/content/drive/MyDrive/Data')
```

```
['Cat', 'Dog', 'withmask1.jpeg']
```

```
#length of the cat data
len(os.listdir('/content/drive/MyDrive/Data/Cat'))
```

```
121
```

```
#length of the dog data
len(os.listdir('/content/drive/MyDrive/Data/Dog'))
```

```
129
```

```
#loading cat data
catpath=os.path.join('/content/drive/MyDrive/Data','Cat')
```

```
for i in os.listdir(catpath):
  print(i)
```

```
cat.4077.jpg
cat.4063.jpg
cat.4117.jpg
cat.4103.jpg
cat.4113.jpg
cat.4107.jpg
cat.4098.jpg
cat.4073.jpg
cat.4067.jpg
cat.4099.jpg
cat.4066.jpg
cat.4072.jpg
cat.4112.jpg
cat.4106.jpg
cat.4110.jpg
cat.4104.jpg
cat.4064.jpg
cat.4070.jpg
cat.4058.jpg
cat.4059.jpg
cat.4071.jpg
cat.4065.jpg
cat.4105.jpg
cat.4111.jpg
cat.4002.jpg
cat.4016.jpg
cat.4003.jpg
cat.4017.jpg
cat.4001.jpg
cat.4029.jpg
cat.4015.jpg
cat.4014.jpg
cat.4028.jpg
cat.4038.jpg
cat.4010.jpg
cat.4004.jpg
cat.4039.jpg
cat.4005.jpg
cat.4011.jpg
cat.4007.jpg
cat.4013.jpg
cat.4012.jpg
cat.4006.jpg
1.jpeg
2.jpeg
3.jpeg
```

```
#loading dog data
dogpath=os.path.join('/content/drive/MyDrive/Data','Dog')
```

```
for i in os.listdir(dogpath):
  print(i)
```

```
    dog.4092.jpg
    dog.4045.jpg
    dog.4051.jpg
    dog.4079.jpg
    dog.4125.jpg
    dog.4119.jpg
    dog.4109.jpg
    dog.4121.jpg
    dog.4041.jpg
    dog.4055.jpg
    dog.4096.jpg
    dog.4069.jpg
    dog.4082.jpg
    dog.4083.jpg
    dog.4068.jpg
    dog.4097.jpg
    dog.4040.jpg
    dog.4054.jpg
    dog.4108.jpg
    dog.4120.jpg
    dog.4122.jpg
    dog.4095.jpg
    dog.4042.jpg
    dog.4081.jpg
    dog.4056.jpg
    dog.4094.jpg
    dog.4080.jpg
    dog.4057.jpg
    dog.4043.jpg
    dog.4123.jpg
    dog.4024.jpg
    dog.4030.jpg
    dog.4018.jpg
    dog.4031.jpg
    dog.4019.jpg
    dog.4025.jpg
    dog.4033.jpg
    dog.4027.jpg
    dog.4026.jpg
    dog.4032.jpg
    dog.4036.jpg
    dog.4022.jpg
    dog.4023.jpg
    dog.4037.jpg
    dog.4021.jpg
    dog.4035.jpg
    dog.4009.jpg
    dog.4034.jpg
    dog.4020.jpg
    dog.4008.jpg
    download.jpeg
    download (1).jpeg
    download (2).jpeg
```

```
flat_data_arr=[]          #used for collecting input variables
target_arr=[]             #used for collecting output variables
categories=['Cat','Dog'] #cat ===> 0 dog ===>1
datadir='/content/drive/MyDrive/Data'
```

```
for i in categories:
  print('Loading categories....')    #cat
  path=os.path.join(datadir,i)     #/content/drive/MyDrive/Data,Cat
  for img in os.listdir(path):
    img_array=imread(os.path.join(path,img))   #/content/drive/MyDrive/Data,cat,eac
    img_resize=resize(img_array,(150,150,3))
    flat_data_arr.append(img_resize.flatten())
    target_arr.append(categories.index(i))
  print('Loaded completely...',i)
```

```
Loading categories....
Loaded completely... Cat
Loading categories....
Loaded completely... Dog
```

```
target_arr
```

```
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1,
      1]
```

```
flat_data_arr=np.array(flat_data_arr)
```

```
target_arr=np.array(target_arr)
```

```
print(flat_data_arr)
```

```
[[0.17549737 0.1833405  0.16373266 ... 0.09063287 0.40080126 0.72296581]
 [0.39658875 0.4022493  0.34369725 ... 0.7595291  0.86936811 0.78441249]
 [0.67607008 0.64709053 0.62452062 ... 0.23391951 0.54159827 0.18205878]
 ...
 [0.53803513 0.52627042 0.56940768 ... 0.23727256 0.40582654 0.13545867]
 [0.21096731 0.17261854 0.12599221 ... 0.75294118 0.75294118 0.74509804]
 [0.66690223 0.64441999 0.66325052 ... 0.57114853 0.75938383 0.22997206]]
```

```
#creating data frame
df=pd.DataFrame(flat_data_arr)
```

```
df
```

	0	1	2	3	4	5	6	7	
0	0.175497	0.183341	0.163733	0.587230	0.595073	0.575465	0.679054	0.686897	0.66
1	0.396589	0.402249	0.343697	0.437244	0.445418	0.369242	0.446242	0.458015	0.35
2	0.676070	0.647091	0.624521	0.667982	0.655925	0.629289	0.646135	0.647337	0.62
3	0.482721	0.529710	0.584640	0.501765	0.529838	0.592293	0.509485	0.526949	0.59
4	0.169435	0.125899	0.058132	0.204059	0.150855	0.102604	0.139838	0.074551	0.03
...	...	...	...	...	...	...	...	...	
245	0.661749	0.642142	0.618612	0.658396	0.638788	0.615259	0.641410	0.621802	0.59
246	0.690510	0.834792	0.845626	0.674535	0.810311	0.824553	0.293848	0.415151	0.43
247	0.538035	0.526270	0.569408	0.534300	0.522536	0.565673	0.530813	0.519049	0.56
248	0.210967	0.172619	0.125992	0.190351	0.162230	0.117551	0.180270	0.159213	0.11
249	0.666902	0.644420	0.663251	0.676728	0.626942	0.624616	0.712589	0.638977	0.61

250 rows × 67500 columns

```
df['Target']=target_arr
```

```
df
```

	0	1	2	3	4	5	6	7	
0	0.175497	0.183341	0.163733	0.587230	0.595073	0.575465	0.679054	0.686897	0.66
1	0.396589	0.402249	0.343697	0.437244	0.445418	0.369242	0.446242	0.458015	0.35
2	0.676070	0.647091	0.624521	0.667982	0.655925	0.629289	0.646135	0.647337	0.62
3	0.482721	0.529710	0.584640	0.501765	0.529838	0.592293	0.509485	0.526949	0.59
4	0.169435	0.125899	0.058132	0.204059	0.150855	0.102604	0.139838	0.074551	0.03
...	...	...	...	...	...	...	...	...	
245	0.661749	0.642142	0.618612	0.658396	0.638788	0.615259	0.641410	0.621802	0.59
246	0.690510	0.834792	0.845626	0.674535	0.810311	0.824553	0.293848	0.415151	0.43
247	0.538035	0.526270	0.569408	0.534300	0.522536	0.565673	0.530813	0.519049	0.56
248	0.210967	0.172619	0.125992	0.190351	0.162230	0.117551	0.180270	0.159213	0.11
249	0.666902	0.644420	0.663251	0.676728	0.626942	0.624616	0.712589	0.638977	0.61

250 rows × 67501 columns

```
df.shape
```

```
(250, 67501)
```

```
#seperatong imput varibales
x=df.iloc[:,:-1]
```

```
x
```

	0	1	2	3	4	5	6	7	
0	0.175497	0.183341	0.163733	0.587230	0.595073	0.575465	0.679054	0.686897	0.66
1	0.396589	0.402249	0.343697	0.437244	0.445418	0.369242	0.446242	0.458015	0.35
2	0.676070	0.647091	0.624521	0.667982	0.655925	0.629289	0.646135	0.647337	0.62
3	0.482721	0.529710	0.584640	0.501765	0.529838	0.592293	0.509485	0.526949	0.59
4	0.169435	0.125899	0.058132	0.204059	0.150855	0.102604	0.139838	0.074551	0.03
...	...	...	...	...	...	...	...	...	
245	0.661749	0.642142	0.618612	0.658396	0.638788	0.615259	0.641410	0.621802	0.59
246	0.690510	0.834792	0.845626	0.674535	0.810311	0.824553	0.293848	0.415151	0.43
247	0.538035	0.526270	0.569408	0.534300	0.522536	0.565673	0.530813	0.519049	0.56
248	0.210967	0.172619	0.125992	0.190351	0.162230	0.117551	0.180270	0.159213	0.11
249	0.666902	0.644420	0.663251	0.676728	0.626942	0.624616	0.712589	0.638977	0.61

250 rows × 67500 columns

```
#seperating output variable
y=df.iloc[:,-1]
```

```
#seperating traing and testing data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1,random_state=1)
```

```
y
```

```
0      0
1      0
2      0
3      0
4      0
      ..
245    1
246    1
247    1
248    1
249    1
Name: Target, Length: 250, dtype: int64
```

```
#model creation
model=SVC()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
```

```
array([1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 0])
```

```
#checking accuracy
from sklearn.metrics import accuracy_score,confusion_matrix,ConfusionMatrixDisplay,
score=accuracy_score(y_test,y_pred)
print('accuracy score = ',score)
```

```
accuracy score =  0.56
```

```
mse = mean_squared_error(y_test,y_pred)
print('mean squared error = ',mse)
```

```
mean squared error =  0.44
```

```
matrix=confusion_matrix(y_test,y_pred)
print(matrix)
```

```
[[ 4  8]
 [ 3 10]]
```

```
#predicting a test image
path1='/content/drive/MyDrive/dog1.jpeg'
img=imread(path1)
img=resize(img,(150,150,3)).flatten().reshape(1,-1)
model.predict(img)
```

```
array([0])
```