

PROJECT : IMAGE CLASSIFICATION

Data Load from Drive

```
In [ ]: #Drive Mount
        from google.colab import drive
        drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Import Required Libraries

```
In [ ]: import pandas as pd
        import numpy as np
        from sklearn.svm import SVC
        import seaborn as sns
        import matplotlib.pyplot as plt
        import os
        from skimage.io import imread
        from sklearn.model_selection import train_test_split
        from skimage.transform import resize
```

```
In [ ]: os.listdir('/content/drive/MyDrive/Data-20231204T072415Z-001/Data')
```

```
Out[ ]: ['Dog', 'Cat']
```

```
In [ ]: len(os.listdir('/content/drive/MyDrive/Data-20231204T072415Z-001/Data/Cat'))
```

```
Out[ ]: 85
```

```
In [ ]: len(os.listdir('/content/drive/MyDrive/Data-20231204T072415Z-001/Data/Dog'))
```

```
Out[ ]: 68
```

```
In [ ]: #Cat
        #Data .join Cat
        #Dog
        #or
        #catpath=os.path('/content/drive/MyDrive/Data-20231204T072415Z-001/Data/Cat')
        catpath=os.path.join('/content/drive/MyDrive/Data-20231204T072415Z-001/Data','Cat')
        for img in os.listdir(catpath):
            print(img)
```

cat.110.jpg
cat.106.jpg
cat.102.jpg
cat.111.jpg
cat.101.jpg
cat.107.jpg
cat.100.jpg
cat.105.jpg
cat.103.jpg
cat.104.jpg
cat.109.jpg
cat.108.jpg
cat.119.jpg
cat.114.jpg
cat.112.jpg
cat.113.jpg
cat.118.jpg
cat.117.jpg
cat.115.jpg
cat.116.jpg
cat.39.jpg
cat.36.jpg
cat.50.jpg
cat.61.jpg
cat.43.jpg
cat.49.jpg
cat.54.jpg
cat.51.jpg
cat.35.jpg
cat.53.jpg
cat.56.jpg
cat.38.jpg
cat.58.jpg
cat.41.jpg
cat.44.jpg
cat.40.jpg
cat.47.jpg
cat.55.jpg
cat.63.jpg
cat.62.jpg
cat.45.jpg

cat.46.jpg
cat.42.jpg
cat.60.jpg
cat.37.jpg
cat.59.jpg
cat.52.jpg
cat.48.jpg
cat.57.jpg
cat.81.jpg
cat.66.jpg
cat.86.jpg
cat.91.jpg
cat.84.jpg
cat.92.jpg
cat.70.jpg
cat.82.jpg
cat.68.jpg
cat.99.jpg
cat.77.jpg
cat.94.jpg
cat.78.jpg
cat.69.jpg
cat.75.jpg
cat.64.jpg
cat.95.jpg
cat.89.jpg
cat.80.jpg
cat.98.jpg
cat.87.jpg
cat.74.jpg
cat.67.jpg
cat.76.jpg
cat.97.jpg
cat.88.jpg
cat.72.jpg
cat.65.jpg
cat.73.jpg
cat.85.jpg
cat.93.jpg
cat.90.jpg
cat.96.jpg

cat.83.jpg
cat.71.jpg
cat.79.jpg

```
In [ ]: dogpath=os.path.join('/content/drive/MyDrive/Data-20231204T072415Z-001/Data', 'Dog')  
        for img in os.listdir(dogpath):  
            print(img)
```

dog.12.jpg
dog.11.jpg
dog.10.jpg
dog.13.jpg
dog.14.jpg
dog.1.jpg
dog.15.jpg
dog.22.jpg
dog.21.jpg
dog.23.jpg
dog.20.jpg
dog.2.jpg
dog.18.jpg
dog.16.jpg
dog.19.jpg
dog.17.jpg
dog.24.jpg
dog.30.jpg
dog.31.jpg
dog.29.jpg
dog.32.jpg
dog.33.jpg
dog.27.jpg
dog.3.jpg¹
dog.25.jpg
dog.26.jpg
dog.28.jpg
dog.38.jpg
dog.52.jpg
dog.51.jpg
dog.53.jpg
dog.36.jpg
dog.49.jpg
dog.4.jpg²
dog.60.jpg
dog.44.jpg
dog.50.jpg
dog.35.jpg
dog.56.jpg
dog.46.jpg
dog.59.jpg

dog.48.jpg
dog.43.jpg
dog.42.jpg
dog.47.jpg
dog.40.jpg
dog.37.jpg
dog.34.jpg
dog.57.jpg
dog.5.jpg
dog.58.jpg
dog.45.jpg
dog.41.jpg
dog.54.jpg
dog.55.jpg
dog.39.jpg
dog.6.jpg
dog.9.jpg
dog.65.jpg
dog.62.jpg
dog.7.jpg
dog.61.jpg
dog.63.jpg
dog.68.jpg
dog.66.jpg
dog.64.jpg
dog.8.jpg
dog.67.jpg

```
In [ ]: flat_data_arr=[] #collecting input variable eg:cat features
        target_arr=[] #collecting output variable
        categories=['Cat', 'Dog']
        datadir='/content/drive/MyDrive/Data-20231204T072415Z-001/Data'
```

```
In [ ]: for i in categories:
        print("Loading categories...")
        path=os.path.join(datadir,i)
        for img in os.listdir(path):
            img_arr=imread(os.path.join(path,img))
            img_resize=resize(img_arr,(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

```
In [ ]: flat_data=np.array(flat_data_arr)
target_data=np.array(target_arr)
df=pd.DataFrame(flat_data)
df
```

```
Out[ ]:
```

	0	1	2	3	4	5	6	7	8	9 ...	67490	67491	67
0	0.654589	0.092548	0.203156	0.643521	0.093398	0.199501	0.633144	0.093794	0.195976	0.627333 ...	0.196421	0.644119	0.166
1	0.252673	0.225222	0.200269	0.253675	0.226224	0.201271	0.269715	0.242264	0.217310	0.277829 ...	0.718447	0.671737	0.695
2	0.567769	0.512867	0.461886	0.573767	0.518865	0.467885	0.545760	0.490858	0.439877	0.557858 ...	0.521950	0.728231	0.660
3	0.362646	0.316581	0.213245	0.338924	0.300045	0.192860	0.331955	0.296877	0.183513	0.282675 ...	0.749911	0.991918	0.968
4	0.281699	0.995425	0.999346	0.281699	0.995425	0.999346	0.281699	0.995425	0.999346	0.281569 ...	0.975382	0.677381	0.997
...
148	0.339050	0.367841	0.423704	0.496663	0.514714	0.548421	0.471958	0.488760	0.506873	0.425741 ...	0.305655	0.616179	0.447
149	0.379472	0.528492	0.194307	0.377842	0.527678	0.186123	0.376462	0.529329	0.174595	0.376284 ...	0.074985	0.350342	0.554
150	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922 ...	0.003922	0.003922	0.003
151	0.342631	0.439104	0.279701	0.275549	0.388488	0.238657	0.286709	0.418971	0.200387	0.280261 ...	0.972995	0.987786	0.964
152	0.567795	0.743022	0.536629	0.576316	0.754119	0.562781	0.532521	0.711565	0.525503	0.521763 ...	0.192060	0.292546	0.214

153 rows × 67500 columns

```
In [ ]: df.shape
```

```
Out[ ]: (153, 67500)
```



```
In [ ]: df['target']=target_data
df
```

```
Out[ ]:
```

	0	1	2	3	4	5	6	7	8	9	...	67491	67492	67
0	0.654589	0.092548	0.203156	0.643521	0.093398	0.199501	0.633144	0.093794	0.195976	0.627333	...	0.644119	0.166650	0.215
1	0.252673	0.225222	0.200269	0.253675	0.226224	0.201271	0.269715	0.242264	0.217310	0.277829	...	0.671737	0.695783	0.694
2	0.567769	0.512867	0.461886	0.573767	0.518865	0.467885	0.545760	0.490858	0.439877	0.557858	...	0.728231	0.660557	0.600
3	0.362646	0.316581	0.213245	0.338924	0.300045	0.192860	0.331955	0.296877	0.183513	0.282675	...	0.991918	0.968385	0.740
4	0.281699	0.995425	0.999346	0.281699	0.995425	0.999346	0.281699	0.995425	0.999346	0.281569	...	0.677381	0.997971	0.993
...
148	0.339050	0.367841	0.423704	0.496663	0.514714	0.548421	0.471958	0.488760	0.506873	0.425741	...	0.616179	0.447552	0.310
149	0.379472	0.528492	0.194307	0.377842	0.527678	0.186123	0.376462	0.529329	0.174595	0.376284	...	0.350342	0.554265	0.122
150	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	...	0.003922	0.003922	0.003
151	0.342631	0.439104	0.279701	0.275549	0.388488	0.238657	0.286709	0.418971	0.200387	0.280261	...	0.987786	0.964669	0.972
152	0.567795	0.743022	0.536629	0.576316	0.754119	0.562781	0.532521	0.711565	0.525503	0.521763	...	0.292546	0.214115	0.210

153 rows × 67501 columns

```
In [ ]: df.shape
```

```
Out[ ]: (153, 67501)
```

```
In [ ]: x=df.iloc[:, :-1]
y=df.iloc[:, -1]
```

```
In [ ]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

```
Out[ ]:
```

	0	1	2	3	4	5	6	7	8	9	...	67490	67491	67
108	0.859031	0.902251	0.878887	0.862030	0.907689	0.889164	0.864652	0.913549	0.896876	0.866896	...	0.677147	0.630222	0.673
90	0.768130	0.691507	0.619205	0.562160	0.443307	0.340889	0.623372	0.472422	0.342538	0.606955	...	0.908014	0.953688	0.958
104	0.685722	0.646506	0.368075	0.699454	0.660238	0.381807	0.698172	0.658956	0.380525	0.702169	...	0.255720	0.279402	0.255
16	0.133865	0.169159	0.227983	0.139061	0.174355	0.233178	0.146083	0.181377	0.240201	0.154350	...	0.220043	0.393147	0.302
111	0.320218	0.355512	0.375120	0.316788	0.352083	0.371690	0.314014	0.349308	0.368916	0.315008	...	0.617914	0.486669	0.521
...
71	0.746801	0.664834	0.570610	0.768159	0.669622	0.570784	0.766071	0.672205	0.569096	0.722603	...	0.414763	0.622804	0.523
106	0.699010	0.691167	0.695088	0.695338	0.687495	0.691417	0.699813	0.691969	0.695891	0.710722	...	0.397026	0.195983	0.274
14	0.082382	0.082468	0.075251	0.094962	0.097496	0.081870	0.107494	0.110449	0.093546	0.108454	...	0.551303	0.535894	0.535
92	0.162867	0.161325	0.192789	0.088525	0.085070	0.095780	0.138907	0.127494	0.119424	0.106875	...	0.491688	0.642948	0.537
102	0.623609	0.767230	0.750716	0.629402	0.761934	0.747243	0.630935	0.743134	0.725714	0.554050	...	0.246763	0.399577	0.337

107 rows × 67500 columns

```
In [ ]: from sklearn.svm import SVC
s=SVC()
s.fit(x_train,y_train)
y_pred=s.predict(x_test)
y_pred
```

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

```
In [ ]: from sklearn.metrics import confusion_matrix,accuracy_score
matr=confusion_matrix(y_test,y_pred)
print(matr)
score=accuracy_score(y_test,y_pred)
score
```

```
[[12 17]
 [ 8  9]]
```

Out[]: 0.45652173913043476

```
In [ ]: path1='/content/drive/MyDrive/Data-20231204T072415Z-001/Data/Cat/cat.100.jpg'
img=imread(path1)
img=resize(img,(150,150,3)).flatten().reshape(1,-1)#dimension corrct
s.predict(img)
```

Out[]: array([0])