

“Image Classifier”

OBJECTIVE:

This Image Classifier is created by the help of Machine Learning and Python. The features of the Image Classifier are Predict the new Image and show its category name as well as show the probability of all category

TECH STACK USED:

Python

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming

Machine Learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

PROJECT DESCRIPTION:

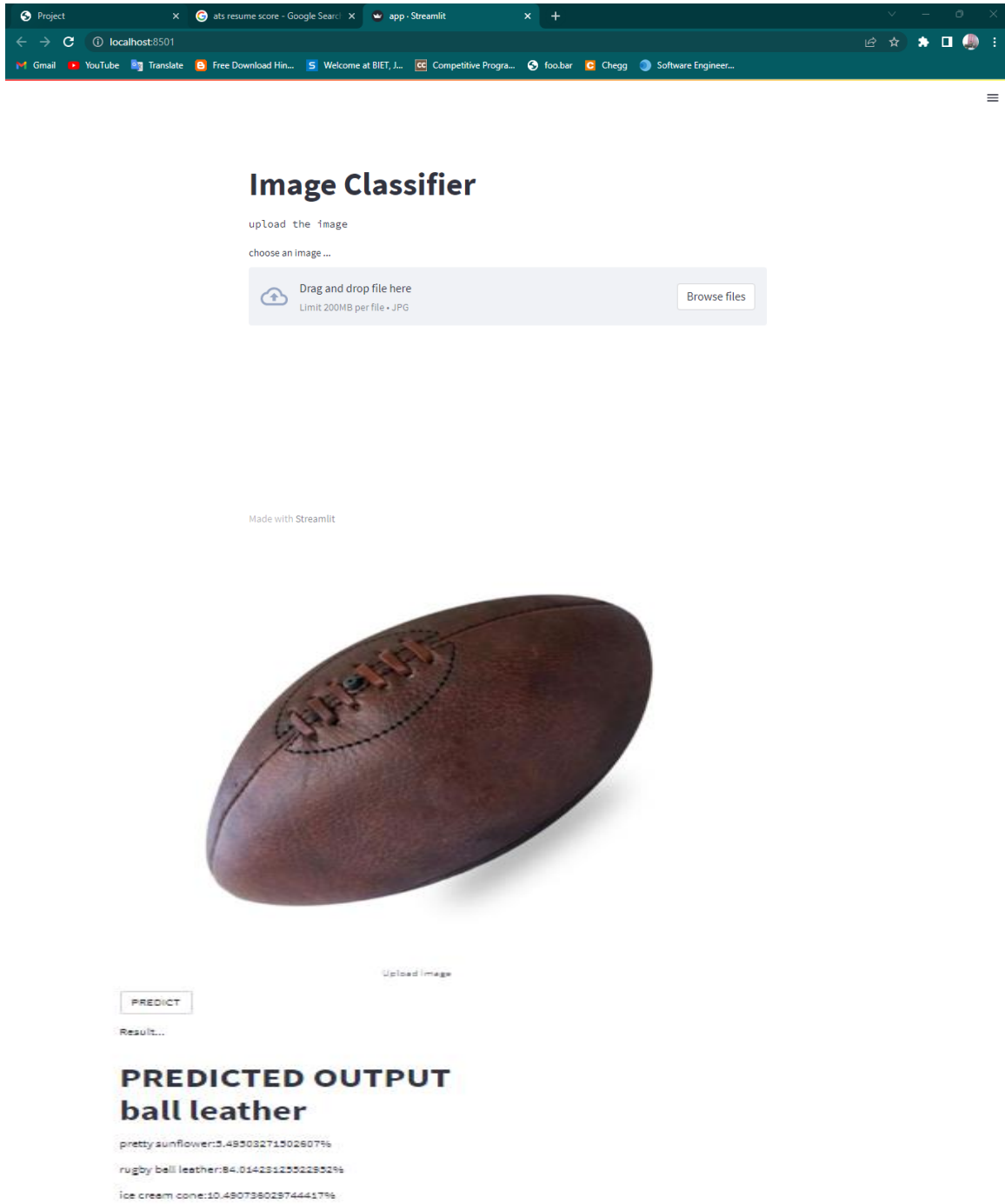
The Image Classifier is built using Machine Learning Support Vector Machines Algorithm.

It takes the input an Image and predict it show its category name as well as show the probability of all category

PROJECT GITHUB LINK:

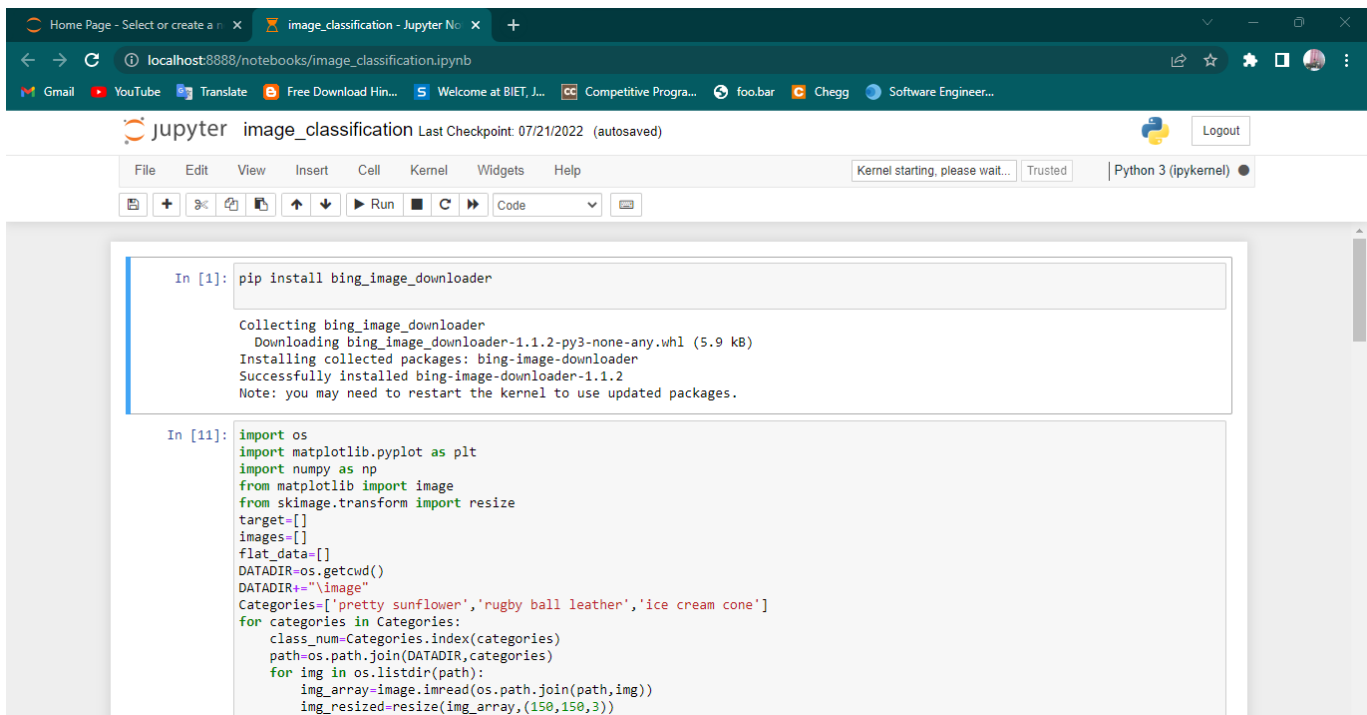
<https://github.com/Dhananjay17728/image-classification-using-machine-learning>

Screenshot





```
1 import streamlit as st
2 import numpy as np
3 from skimage.transform import resize
4 import pickle
5 from PIL import Image
6 st.title("Image Classifier")
7 st.text("upload the image")
8 model=pickle.load(open('img_model.pkl','rb'))
9 upload_file=st.file_uploader('choose an image ...', type='jpg')
10 if upload_file is not None:
11     img=Image.open(upload_file)
12     st.image(img,caption='Upload Image')
13     if st.button('PREDICT'):
14         flat_data=[]
15         Categories=['pretty sunflower','rugby ball leather','ice cream cone']
16         st.write('Result...')
17         img=np.array(img)
18         img_resized=resize(img,(150,150,3))
19         flat_data.append(img_resized.flatten())
20         flat_data=np.array(flat_data)
21         print(img.shape)
22         y_out=model.predict(flat_data)
23         y_out=Categories[y_out[0]]
24         st.title(f'PREDICTED OUTPUT:{y_out}')
25         q=model.predict_proba(flat_data)
26         for index,item in enumerate(Categories):
27             st.write(f'{index}:{q[index][0]*100}%')
```



```
In [1]: pip install bing_image_downloader

Collecting bing_image_downloader
  Downloading bing_image_downloader-1.1.2-py3-none-any.whl (5.9 kB)
Installing collected packages: bing-image-downloader
Successfully installed bing-image-downloader-1.1.2
Note: you may need to restart the kernel to use updated packages.

In [11]: import os
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import image
from skimage.transform import resize
target=[]
images=[]
flat_data=[]
DATADIR=os.getcwd()
DATADIR+="image"
Categories=['pretty sunflower','rugby ball leather','ice cream cone']
for categories in Categories:
    class_num=Categories.index(categories)
    path=os.path.join(DATADIR,categories)
    for img in os.listdir(path):
        img_array=image.imread(os.path.join(path,img))
        img_resized=resize(img_array,(150,150,3))
```