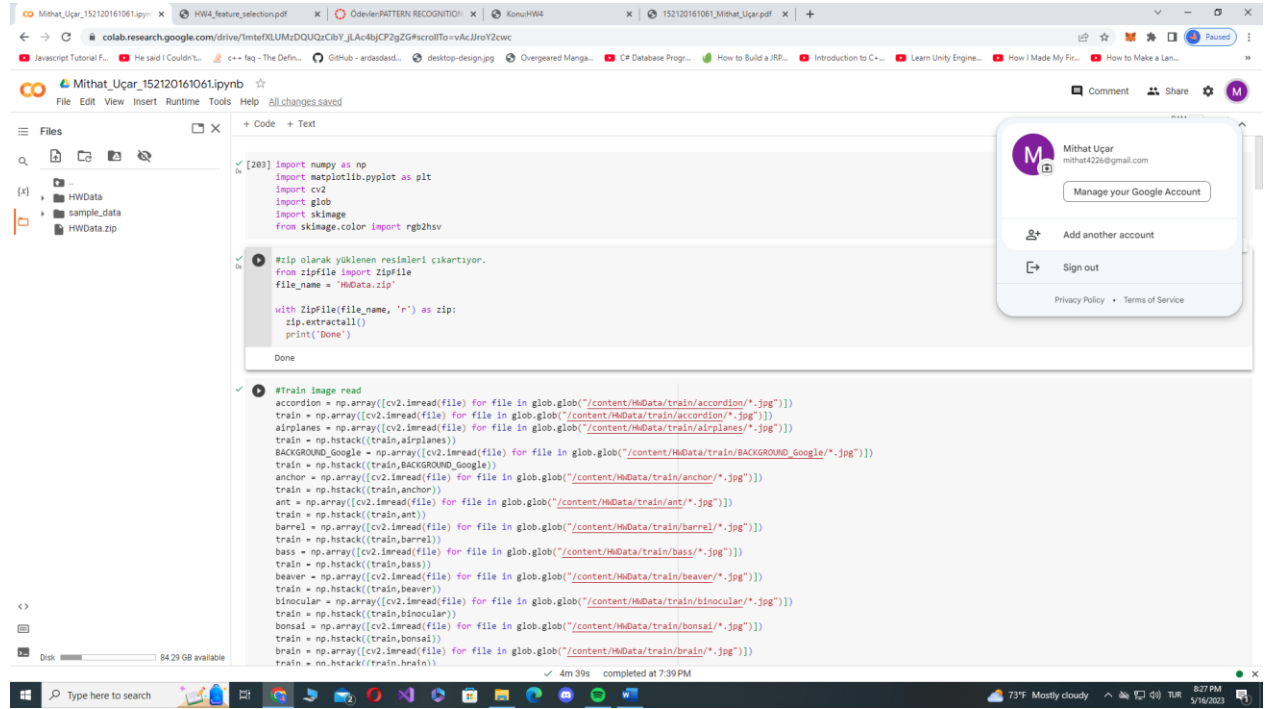


Pattern Recognition HW4



The screenshot shows a Google Colab notebook with the following code:

```
[203] import numpy as np
import matplotlib.pyplot as plt
import cv2
import glob
import skimage
from skimage.color import rgb2hsv

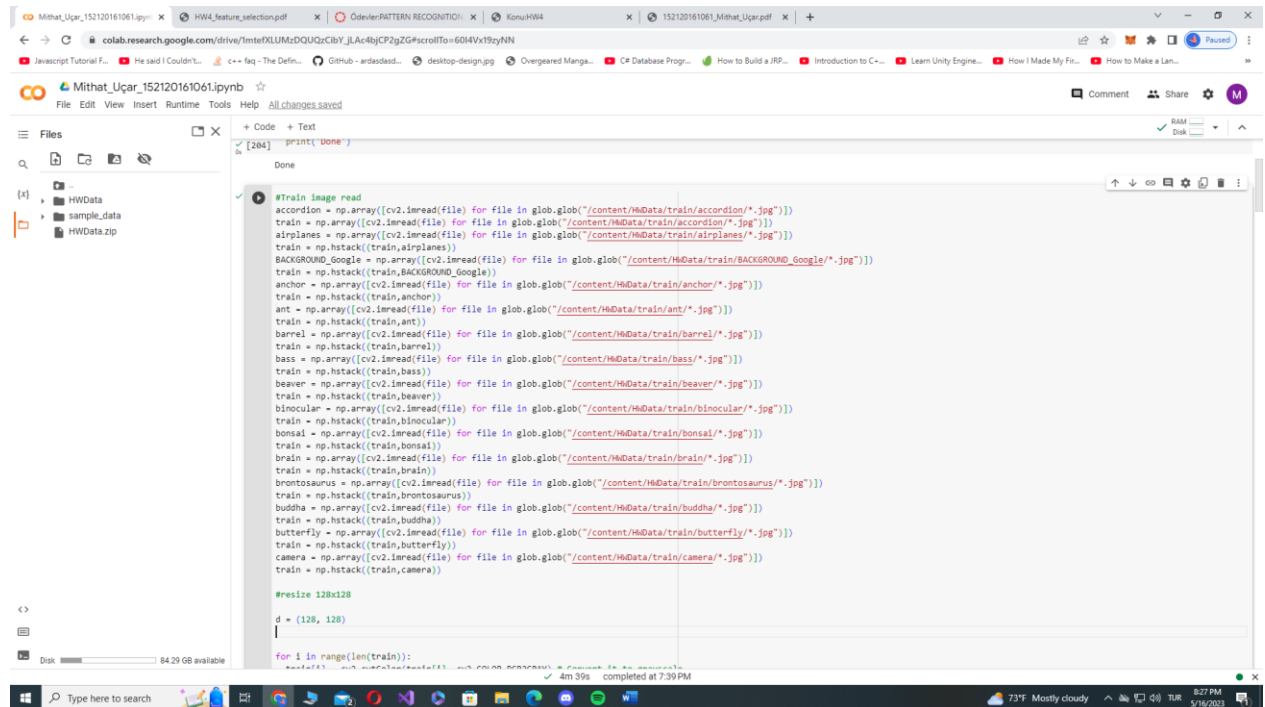
#zip olarak yüklenen resimleri çıkartıyor.
from zipfile import ZipFile
file_name = 'HwData.zip'

with ZipFile(file_name, 'r') as zip:
    zip.extractall()
    print('Done')

Done

#train image read
accordion = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/accordion/*.jpg")])
train = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/accordion/*.jpg")])
airplanes = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/airplanes/*.jpg")])
train = np.hstack((train,airplanes))
BACKGROUND_Google = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/BACKGROUND_Google/*.jpg")])
train = np.hstack((train,BACKGROUND_Google))
anchor = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/anchor/*.jpg")])
train = np.hstack((train,anchor))
ant = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/ant/*.jpg")])
train = np.hstack((train,ant))
barrel = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/barrel/*.jpg")])
train = np.hstack((train,barrel))
bass = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/bass/*.jpg")])
train = np.hstack((train,bass))
beaver = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/beaver/*.jpg")])
train = np.hstack((train,beaver))
binocular = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/binocular/*.jpg")])
train = np.hstack((train,binocular))
bonsai = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/bonsai/*.jpg")])
train = np.hstack((train,bonsai))
brain = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/brain/*.jpg")])
train = np.hstack((train,brain))
```

Resimleri colab'a zip olarak yükledim, zip dosyasını kod ile çıkartarak colab'da kullandım.



The screenshot shows the continuation of the code in the Google Colab notebook:

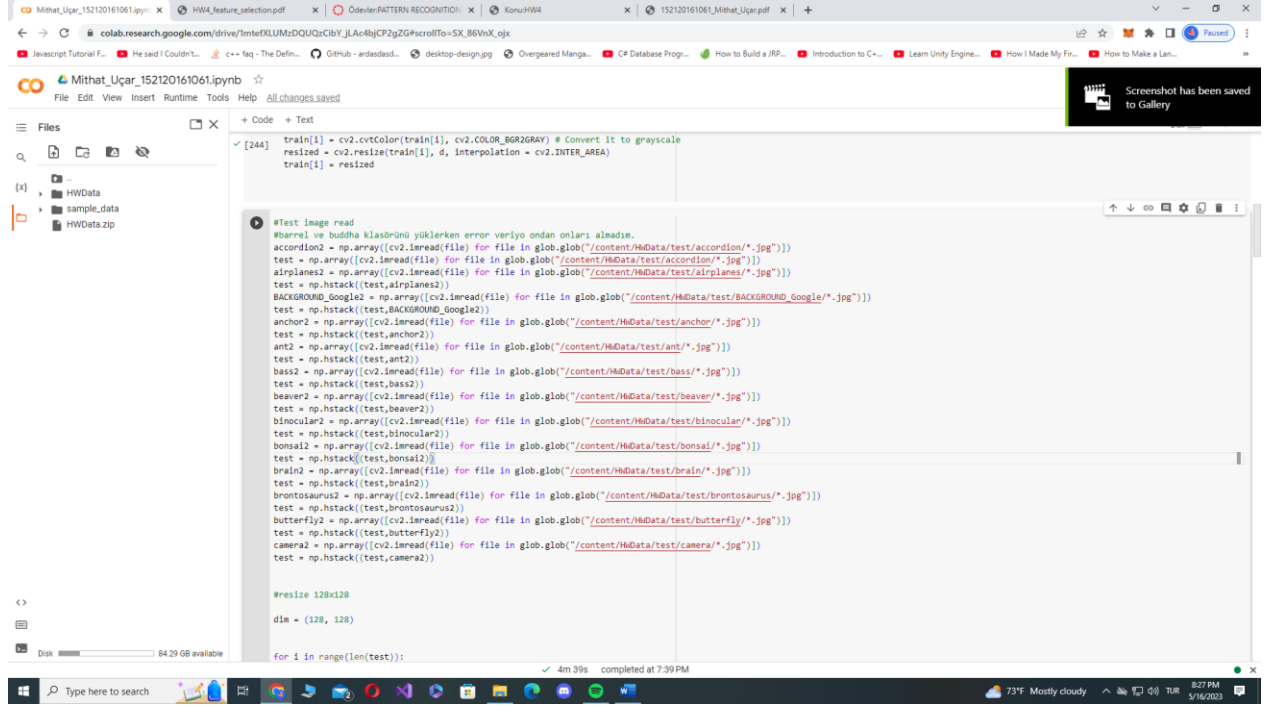
```
[204] print('Done')
Done

#train image read
accordion = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/accordion/*.jpg")])
train = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/accordion/*.jpg")])
airplanes = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/airplanes/*.jpg")])
train = np.hstack((train,airplanes))
BACKGROUND_Google = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/BACKGROUND_Google/*.jpg")])
train = np.hstack((train,BACKGROUND_Google))
anchor = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/anchor/*.jpg")])
train = np.hstack((train,anchor))
ant = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/ant/*.jpg")])
train = np.hstack((train,ant))
barrel = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/barrel/*.jpg")])
train = np.hstack((train,barrel))
bass = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/bass/*.jpg")])
train = np.hstack((train,bass))
beaver = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/beaver/*.jpg")])
train = np.hstack((train,beaver))
binocular = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/binocular/*.jpg")])
train = np.hstack((train,binocular))
bonsai = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/bonsai/*.jpg")])
train = np.hstack((train,bonsai))
brain = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/brain/*.jpg")])
train = np.hstack((train,brain))
brontosaurus = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/brontosaurus/*.jpg")])
train = np.hstack((train,brontosaurus))
buddha = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/buddha/*.jpg")])
train = np.hstack((train,buddha))
butterfly = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/butterfly/*.jpg")])
train = np.hstack((train,butterfly))
camera = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/camera/*.jpg")])
train = np.hstack((train,camera))

#resize 128x128
d = (128, 128)

for i in range(len(train)):
    train[i] = cv2.resize(train[i], d)
```

Resimleri 128x128 piksele dönüştürdüm.



The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The code in the notebook is as follows:

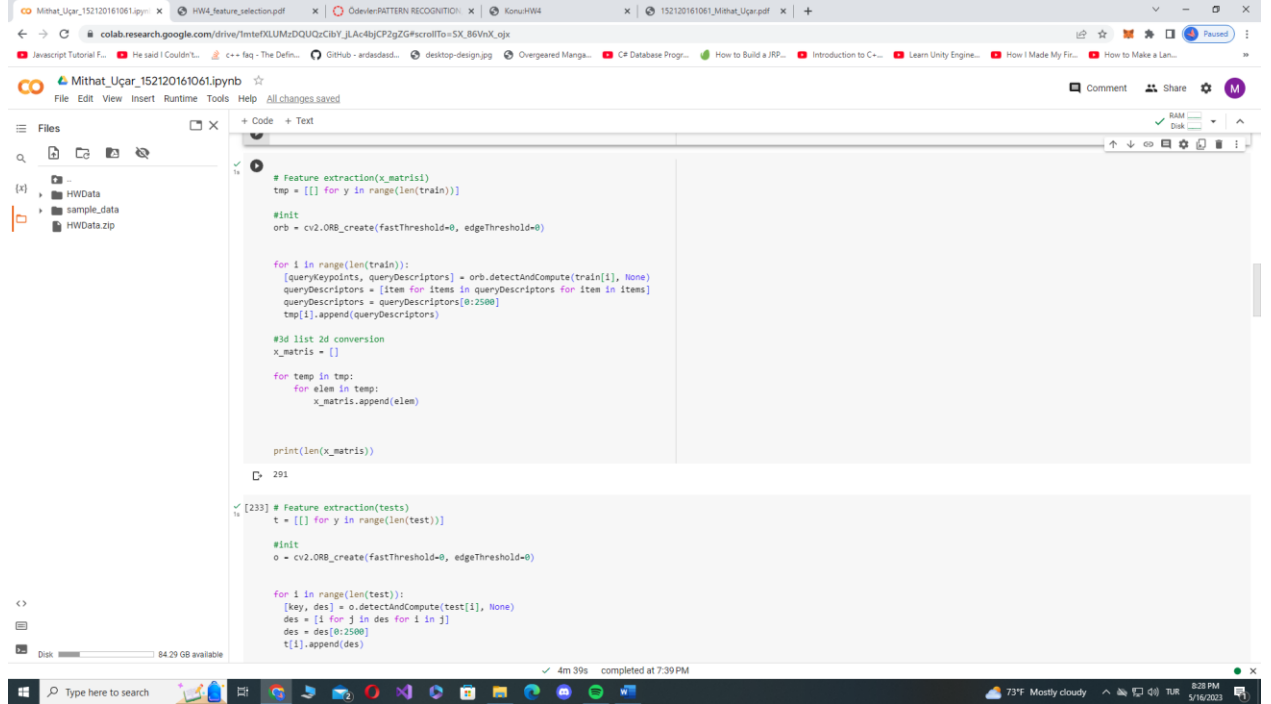
```
train1 = cv2.cvtColor(train1, cv2.COLOR_BGR2GRAY) # Convert it to grayscale
resized = cv2.resize(train1, d, interpolation = cv2.INTER_AREA)
train1 = resized

#Test image read
#barrel ve buda klasörünü yüklerken error veriyö ondan onları almadım.
accordion = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/accordion/*.jpg")])
test = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/accordion/*.jpg")])
airplanes = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/airplanes/*.jpg")])
test = np.hstack((test,airplanes2))
BACKGROUND_Google = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/BACKGROUND_Google/*.jpg")])
test = np.hstack((test,BACKGROUND_Google2))
anchor = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/anchor/*.jpg")])
test = np.hstack((test,anchor2))
ant = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/ant/*.jpg")])
test = np.hstack((test,ant2))
bass = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/bass/*.jpg")])
test = np.hstack((test,bass2))
beaver = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/beaver/*.jpg")])
test = np.hstack((test,beaver2))
binocular = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/binocular/*.jpg")])
test = np.hstack((test,binocular2))
bonsai = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/bonsai/*.jpg")])
test = np.hstack((test,bonsai2))
brain = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/brain/*.jpg")])
test = np.hstack((test,brain2))
brontosaurus = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/brontosaurus/*.jpg")])
test = np.hstack((test,brontosaurus2))
butterfly = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/butterfly/*.jpg")])
test = np.hstack((test,butterfly2))
camera = np.array([cv2.imread(file) for file in glob.glob("/content/H4Data/test/camera/*.jpg")])
test = np.hstack((test,camera2))

#resize 128x128
dim = (128, 128)

for i in range(len(test)):
```

Feature extraction işlemini yapmak için orb yöntemini kullandım.Feature matrisini kullanabilmek için 2 boyuta dönüştürdüm.



The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The code in the notebook is as follows:

```
# Feature extraction(x_train1)
tmp = [[] for y in range(len(train))]

#init
orb = cv2_Orb_create(fastThreshold=0, edgeThreshold=0)

for i in range(len(train)):
    [keypoints, queryDescriptors] = orb.detectandCompute(train[i], None)
    queryDescriptors = [item for items in queryDescriptors for item in items]
    queryDescriptors = queryDescriptors[0:2500]
    tmp[i].append(queryDescriptors)

#3d list 2d conversion
x_train1 = []

for temp in tmp:
    for elem in temp:
        x_train1.append(elem)

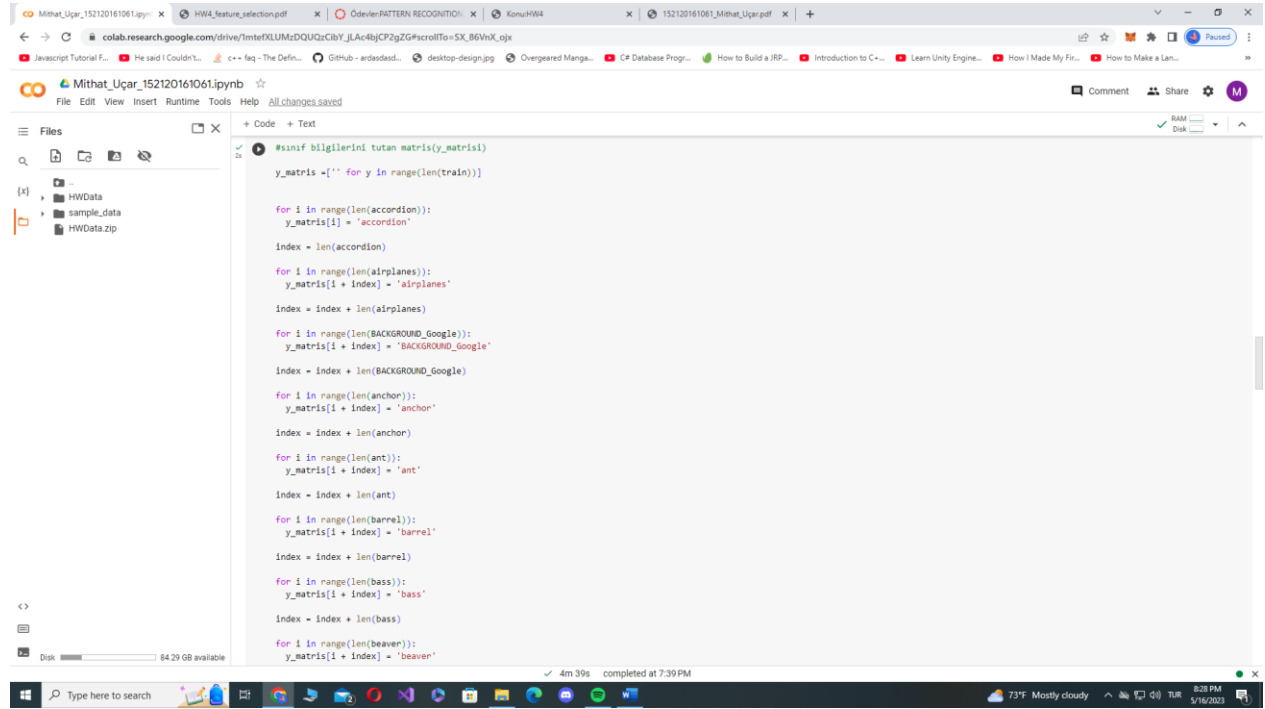
print(len(x_train1))

# Feature extraction(tests)
t = [[] for y in range(len(test))]

#init
o = cv2_Orb_create(fastThreshold=0, edgeThreshold=0)

for i in range(len(test)):
    [key, des] = o.detectandCompute(test[i], None)
    des = [i for j in des for i in j]
    des = des[0:2500]
    t[i].append(des)
```

Labelların olduđu y matrisi



The screenshot shows a Google Colab notebook with the following code in the first cell:

```
#sınıf bilgilerini tutan matris(y_matrisi)

y_matris = ['' for y in range(len(train))]

for i in range(len(accordion)):
    y_matris[i] = 'accordion'

index = len(accordion)

for i in range(len(airplanes)):
    y_matris[i + index] = 'airplanes'

index = index + len(airplanes)

for i in range(len(BACKGROUND_Google)):
    y_matris[i + index] = 'BACKGROUND_Google'

index = index + len(BACKGROUND_Google)

for i in range(len(anchor)):
    y_matris[i + index] = 'anchor'

index = index + len(anchor)

for i in range(len(ant)):
    y_matris[i + index] = 'ant'

index = index + len(ant)

for i in range(len(barrel)):
    y_matris[i + index] = 'barrel'

index = index + len(barrel)

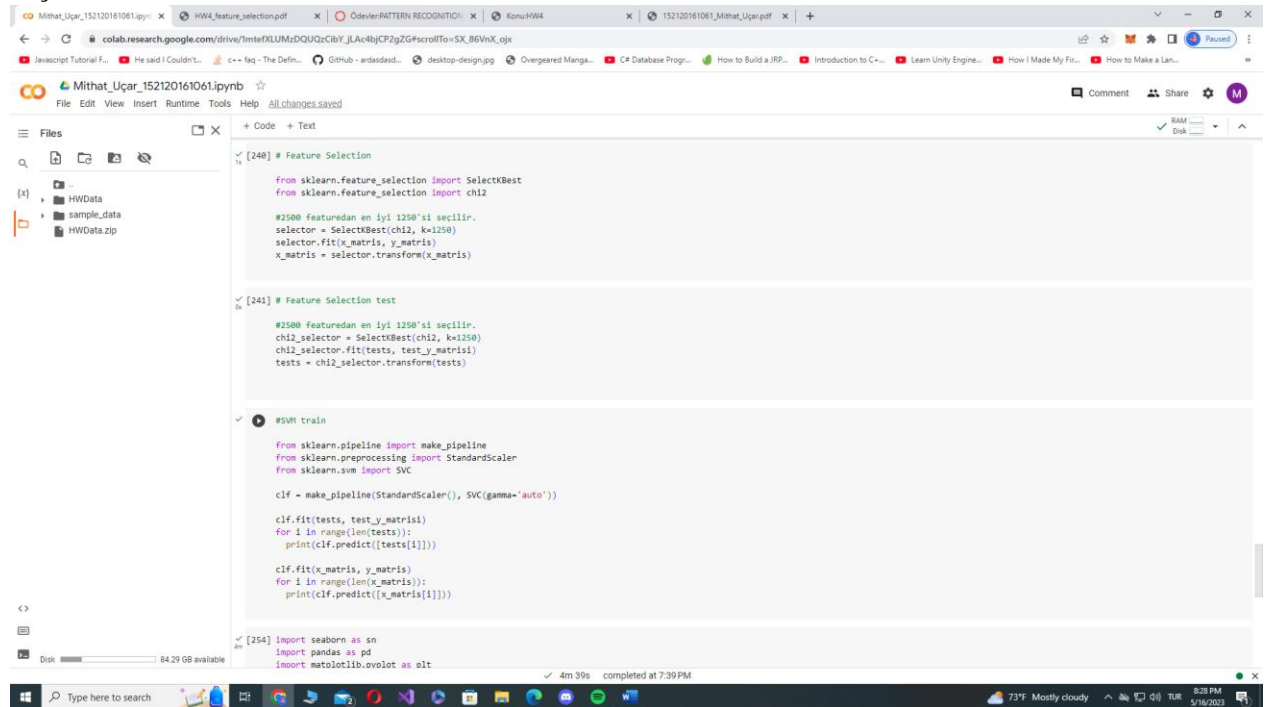
for i in range(len(bass)):
    y_matris[i + index] = 'bass'

index = index + len(bass)

for i in range(len(beaver)):
    y_matris[i + index] = 'beaver'
```

The notebook interface shows the file explorer on the left with folders 'HWDData', 'sample_data', and 'HWDData.zip'. The bottom status bar indicates the code was completed at 7:39 PM.

Feature reduction yaparak 2500 featuredan 1250 feature seçilir.Bunun amacı en zengin değeri seçmektir.



The screenshot shows a Google Colab notebook with the following code in the second cell:

```
[248] # Feature Selection

from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

#2500 featuredan en iyi 1250'si seçilir.
selector = SelectKBest(chi2, k=1250)
selector.fit(x_matris, y_matris)
x_matris = selector.transform(x_matris)

[241] # Feature Selection test

#2500 featuredan en iyi 1250'si seçilir.
chi2_selector = SelectKBest(chi2, k=1250)
chi2_selector.fit(tests, test_y_matrisi)
tests = chi2_selector.transform(tests)

#SVM train

from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC

clf = make_pipeline(StandardScaler(), SVC(gamma='auto'))

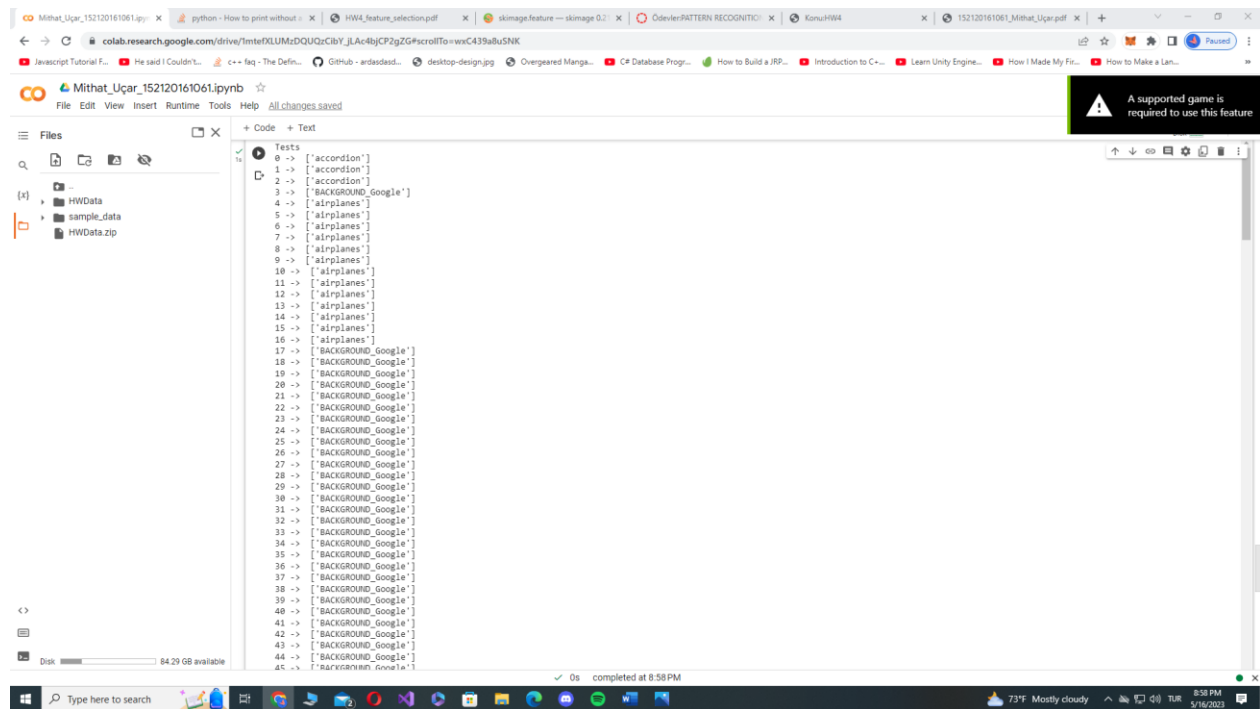
clf.fit(tests, test_y_matrisi)
for i in range(len(tests)):
    print(clf.predict([tests[i]]))

clf.fit(x_matris, y_matris)
for i in range(len(x_matris)):
    print(clf.predict([x_matris[i]]))

[254] import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt
```

The notebook interface shows the file explorer on the left with folders 'HWDData', 'sample_data', and 'HWDData.zip'. The bottom status bar indicates the code was completed at 7:39 PM.

Printler



The screenshot shows a Jupyter Notebook with a file explorer on the left displaying 'HWData', 'sample_data', and 'HWData.zip'. The main code cell contains a list of labels for 45 tests, alternating between 'airplanes' and 'BACKGROUND_Google'.

```
Tests
0 -> ['accordion']
1 -> ['accordion']
2 -> ['accordion']
3 -> ['BACKGROUND_Google']
4 -> ['airplanes']
5 -> ['airplanes']
6 -> ['airplanes']
7 -> ['airplanes']
8 -> ['airplanes']
9 -> ['airplanes']
10 -> ['airplanes']
11 -> ['airplanes']
12 -> ['airplanes']
13 -> ['airplanes']
14 -> ['airplanes']
15 -> ['airplanes']
16 -> ['airplanes']
17 -> ['BACKGROUND_Google']
18 -> ['BACKGROUND_Google']
19 -> ['BACKGROUND_Google']
20 -> ['BACKGROUND_Google']
21 -> ['BACKGROUND_Google']
22 -> ['BACKGROUND_Google']
23 -> ['BACKGROUND_Google']
24 -> ['BACKGROUND_Google']
25 -> ['BACKGROUND_Google']
26 -> ['BACKGROUND_Google']
27 -> ['BACKGROUND_Google']
28 -> ['BACKGROUND_Google']
29 -> ['BACKGROUND_Google']
30 -> ['BACKGROUND_Google']
31 -> ['BACKGROUND_Google']
32 -> ['BACKGROUND_Google']
33 -> ['BACKGROUND_Google']
34 -> ['BACKGROUND_Google']
35 -> ['BACKGROUND_Google']
36 -> ['BACKGROUND_Google']
37 -> ['BACKGROUND_Google']
38 -> ['BACKGROUND_Google']
39 -> ['BACKGROUND_Google']
40 -> ['BACKGROUND_Google']
41 -> ['BACKGROUND_Google']
42 -> ['BACKGROUND_Google']
43 -> ['BACKGROUND_Google']
44 -> ['BACKGROUND_Google']
45 -> ['BACKGROUND_Google']
```

The bottom status bar indicates the notebook was completed at 8:58 PM.

Confusion Matrix:

