Pattern Recognition –Lab#1

About the Assignment

The main aim of the assignment is to make you familiar with a traditional classifier by using KNN. Contributions of this lab are;

- Learning the K-NN classification method.
- Understanding the idea behind the classification task.

Submit the Assignment

Send your code and pdf document as zipped and rar.

Ex: No_Name_Surname_LAB1.zip

Hint

No hint.

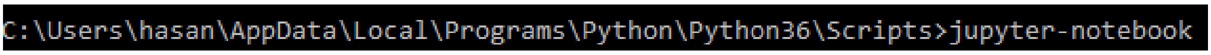
Step0:

Create a folder Lab1Pattern in your desktop.

Open jupyter-notebook in the following place.

C:\Users\hasan\AppData\Local\Programs\Python\Python36\Scripts

In here, open cmd.exe, and type the jupyter-notebook. Like below:



```
C:\Users\hasan\AppData\Local\Programs\Python\Python36\Scripts>jupyter-notebook
```

Once it launched, just press new button, and select python3. A new file would be opened in your web browser. Later pursuing the following steps.

Step0:

Change directory with this snipped code.

```
import os
os.chdir('C:/Users/hasan/Desktop/HW2')
```

Step2:

Download the Cifar-10 dataset python version by using the following commands

from keras.datasets import cifar10

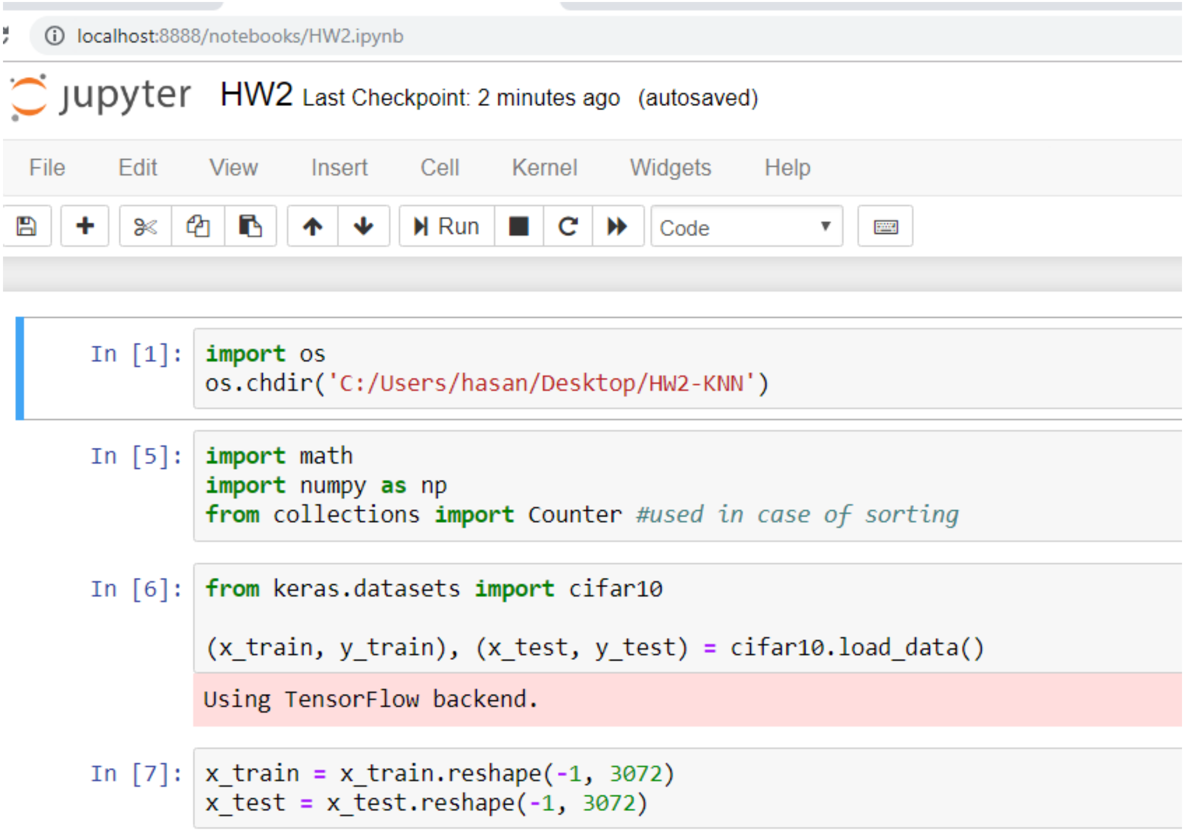
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

Step3:

Convert images to vector format by writing this snipped code. It means that an image with 32x32x3 channels converted as 1x3072 vector. There 50,000 train vectors and 10,000 test vectors.

x_train = x_train.reshape(-1, 3072)

x_test = x_test.reshape(-1, 3072)


localhost:8888/notebooks/HW2.ipynb
jupyter
HW2 Last Checkpoint: 2 minutes ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
In [1]: import os
        os.chdir('C:/Users/hasan/Desktop/HW2-KNN')

In [5]: import math
        import numpy as np
        from collections import Counter #used in case of sorting

In [6]: from keras.datasets import cifar10

        (x_train, y_train), (x_test, y_test) = cifar10.load_data()

        Using TensorFlow backend.

In [7]: x_train = x_train.reshape(-1, 3072)
        x_test = x_test.reshape(-1, 3072)
```

Step4:

Write a function send the parameters of ***x_train, y_train, sample_test, k*** as input, then return the most similar class name for sample_test. In case of computing the similarity, you are expected to use L2 distance, which is explained in lecture notes by Teacher in class. You can use any code.

```
sample_test = x_test[1,:];
```

The test code look like this.

```
sample_test = x_test[1,:];
k=5
similar_class_name = knnClassifier(x_train, y_train, sample_test, k )
print(similar_class_name)
```

I have uploaded my jupyter-notebook codes as HW2.ipynb.
You can open it, in your jupyter-notebook and fulfill the required tasks.
Besides, the required python libraries are imported to use when it is needed.