READ ME

1. Environment

Here is the information about my laptop that all these experiments run on. The system may not have influence.

My running system is macOS Mojave, version 10.14.2. My laptop is MacBook Pro. My Graphics is Intel Iris Plus Graphics 650 1536 MB. As in the picture, you can also see my CPU is Intel Core i5-7276U CPU @ 3.10GHz. All code is written in Python language.

You can see the Python version is 3.6.7 .

2. Install a software and set the environment in this software

Step 1:

A software called Anaconda-Navigator to manage all required packages need to be installed. You can download the software on this website: https://www.anaconda.com/distribution/#download-section. Python 3.7 version is what I am using. Install it.

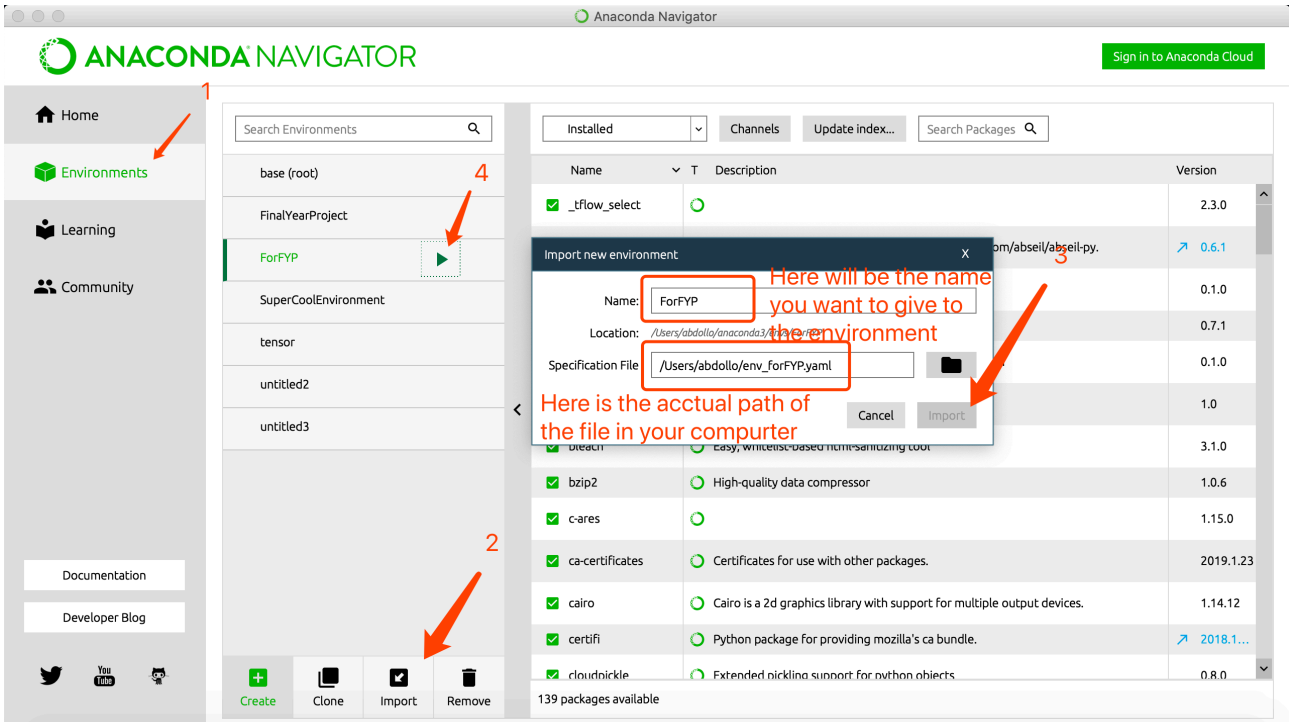
Step 2:

1.Open the anaconda navigator, you can click the Environments on the bar at the left side.

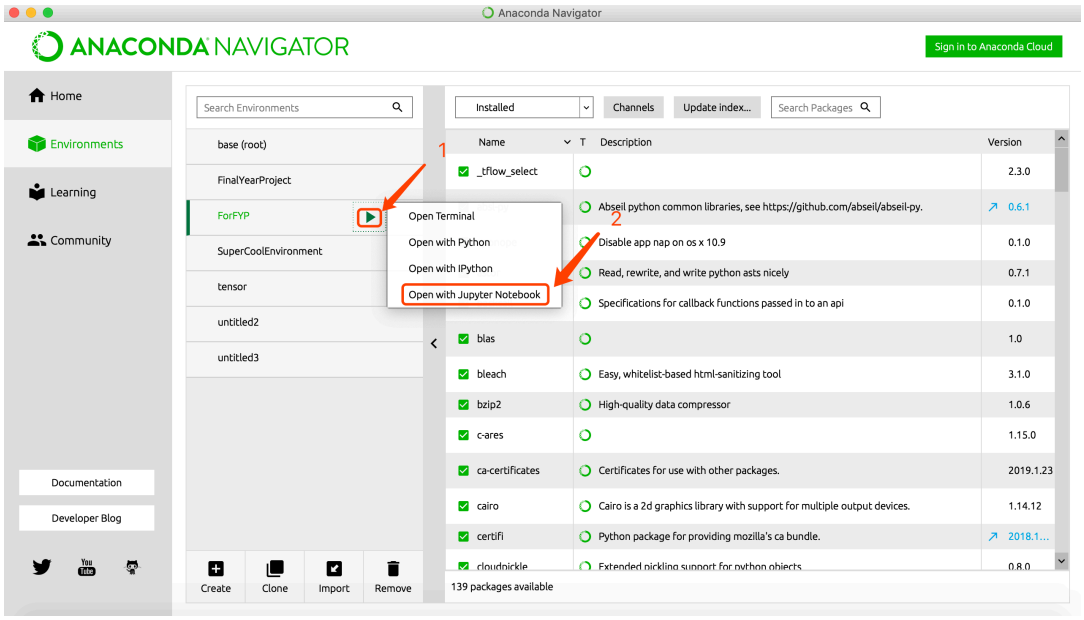
2.You can see the Import bottom and click it.

3.Then you can choose the file named " env_forFYP.yaml" and use any environment name you like. You may spend the time to install all these packages.

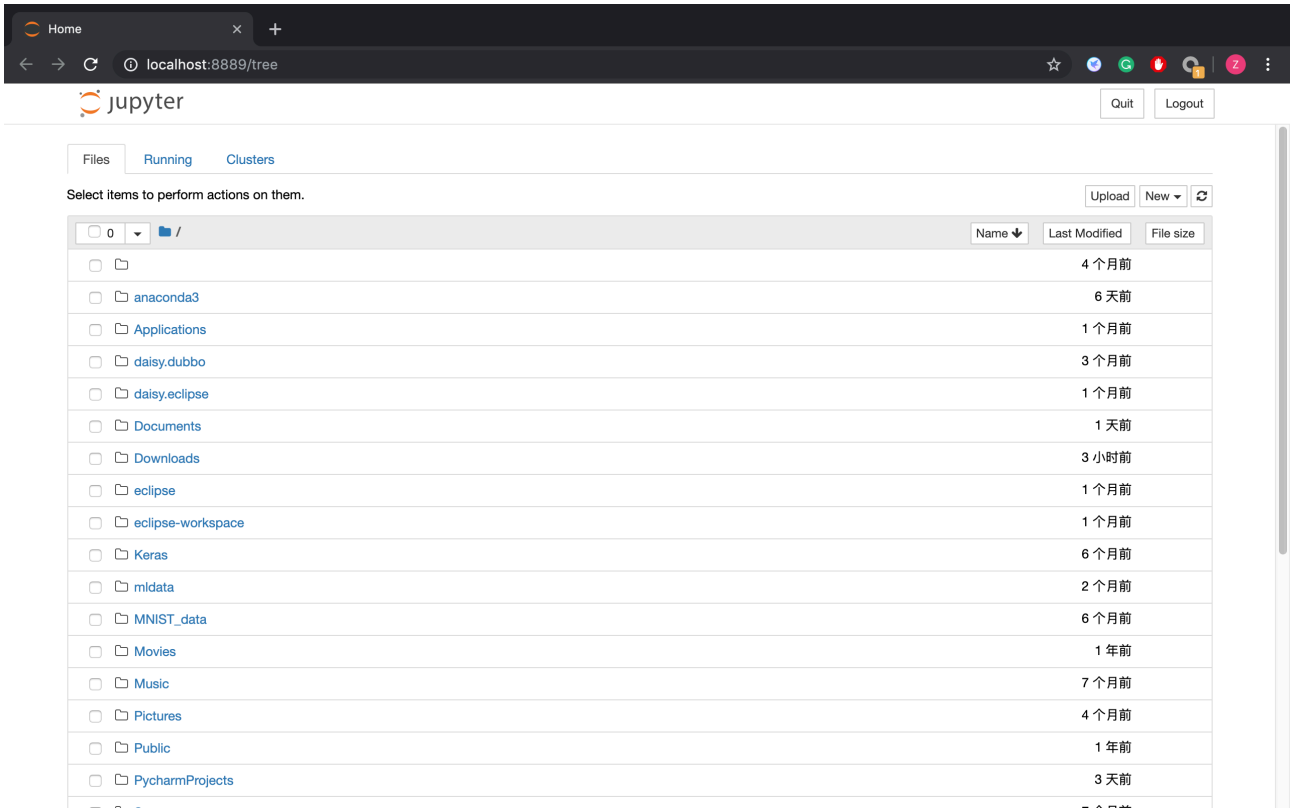
4.Then you can run the environments using Jupiter Notebook. Then in a new web page opened automatically, choose the file and run.



3. Run the experiments
Open the anaconda. Fellow the steps in the picture. Wait and a new web page will be created.

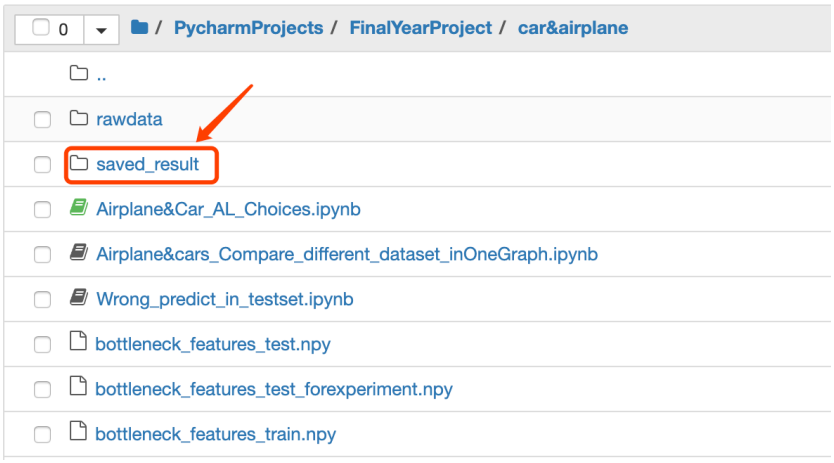


This is an example of the page maybe opened. Here should be the root directory of the computer. Single click can get in the directory. Choose the directory until you find the code.

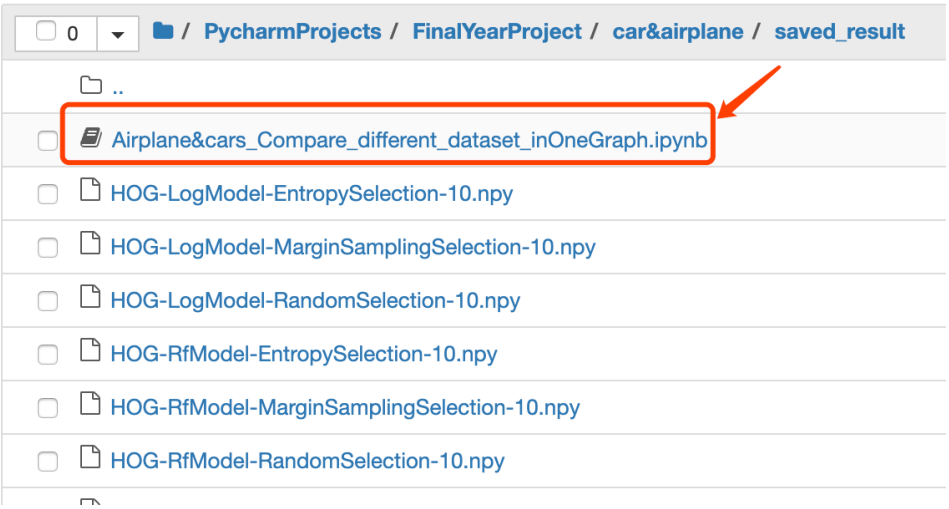


For me the code is in these two directory. (because of the upload file size limit so I only upload these two)

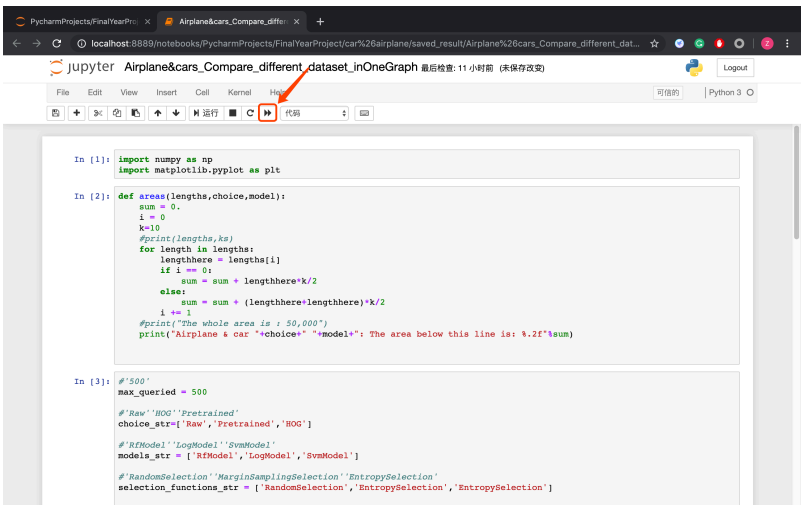
Choose the airplane and car. In this directory, choose the saved_result.



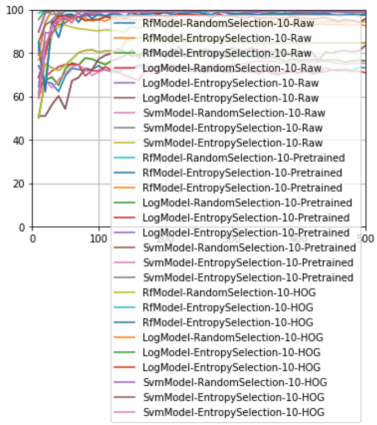
The file end with .ipynb is the code that can be run. Click it.



A new web page will be opened. Click the bottom shown in the picture. It will run the whole file in of this code.



At the end of the code. You can see the result like this. This code is to read the file that record the result of experiments and show them.



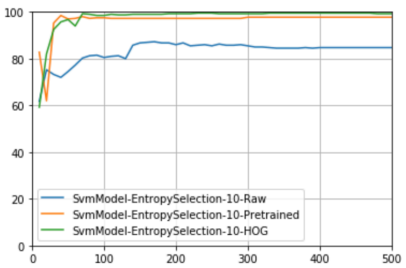
It looks a little bit mess. You can change some code in block 3. The choices have been provided after the #.

```
#'500'
max_queried = 500

#'Raw''HOG''Pretrained'
choice_str=['Raw','Pretrained','HOG']

#'RfModel''LogModel''SvmModel'
models_str = ['RfModel','LogModel','SvmModel']

#'RandomSelection''MarginSamplingSelection''EntropySelection'
selection_functions_str = ['RandomSelection','EntropySelection','EntropySelection']
```

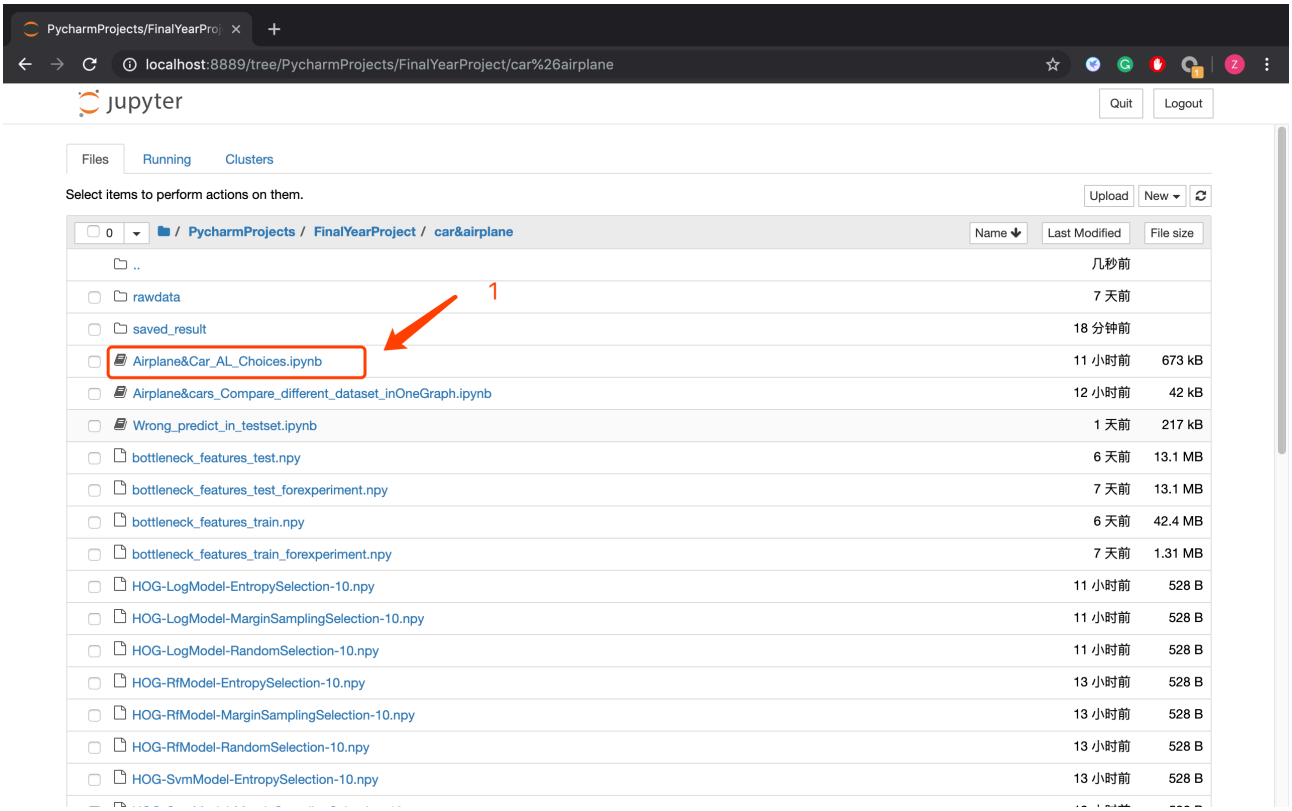
You can choose less to make the picture easier to compare different model. For example, you can change it to this. Run the whole file. And the result become easier to understand.

```
#'500'
max_queried = 500

#'Raw''HOG''Pretrained'
choice_str=['Raw','Pretrained','HOG']

#'RfModel''LogModel''SvmModel'
models_str = ['SvmModel']

#'RandomSelection''MarginSamplingSelection''EntropySelection'
selection_functions_str = ['EntropySelection']
```

```
Airplane & car Raw SvmModel: The area below this line is: 41281.25
Airplane & car Pretrained SvmModel: The area below this line is: 47821.25
Airplane & car HOG SvmModel: The area below this line is: 48578.75
```



The main code is the one end with Choices.ipynb.
Click it, a new page will be opened.



From the last serval blocks, you can find this block. It is used to generate the result with different models you want to choose. After run it, all the experiments with different models will be run. Each experiments from choose Ks pictures to train until it reach the max_queried. For example, here it will choose 10 pictures to train the logic regression with randomselection, record the accuracy, and choose 10 more(20 in total needed to be trained), then 10 more and 10 more until it choose 500 pictures to train. Now a file named HOG-LogModel-RandomSelection-10.npy will be created to record. For different models, selection functions and choices of dataset, they will run individually and automatically and be record in the file automatically. You can choose more than one choices, models and selection functions each time.

If you want to know what is the accuracy of the full dataset. You can change the max_queried and Ks to 1294 for this dataset, and it will get the accuracy. If you do not want to save this data(because it will cover the old file), you can set the save_file to False.

```
#'500''1294'
max_queried = 500
Ks = [10]

#True means will save, False means will not save
save_file=True

#'Raw''HOG''Pretrained'
choices=['HOG']

#'RfModel''LogModel''SvmModel'
models = [LogModel]

#'RandomSelection''MarginSamplingSelection''EntropySelection'
selection_functions = [RandomSelection, MarginSamplingSelection,EntropySelection]
```

After you run it, it will show the picture of the data at the end of the code. Also it will generate(cover if exist) a file to record.

You can run Airplane&cars_Compare_different_dataset_inOneGraph.ipynb to compare different data.

In Wrong_predict_in_testset.ipynb it will generate the pictures that the model predicts wrong. This is fixed mode: 40 pictures with pre-trained model and SVM.

For the other dataset, everything is similar.