A summary of our seven day study

01 Machine Learning

- What is the Machine Learning

What is the Machine Learning

Machine learning is a computer theory developed by a group of computer scientists to make computers think like people

Place of application:Google Now, Google Photos, Baidu image recognition, etc.

Types of machine learning algorithms

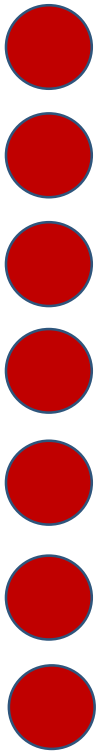
supervised learning

Unsupervised learning

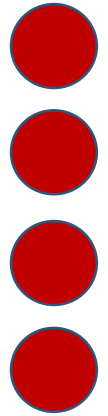
semi-supervised learning

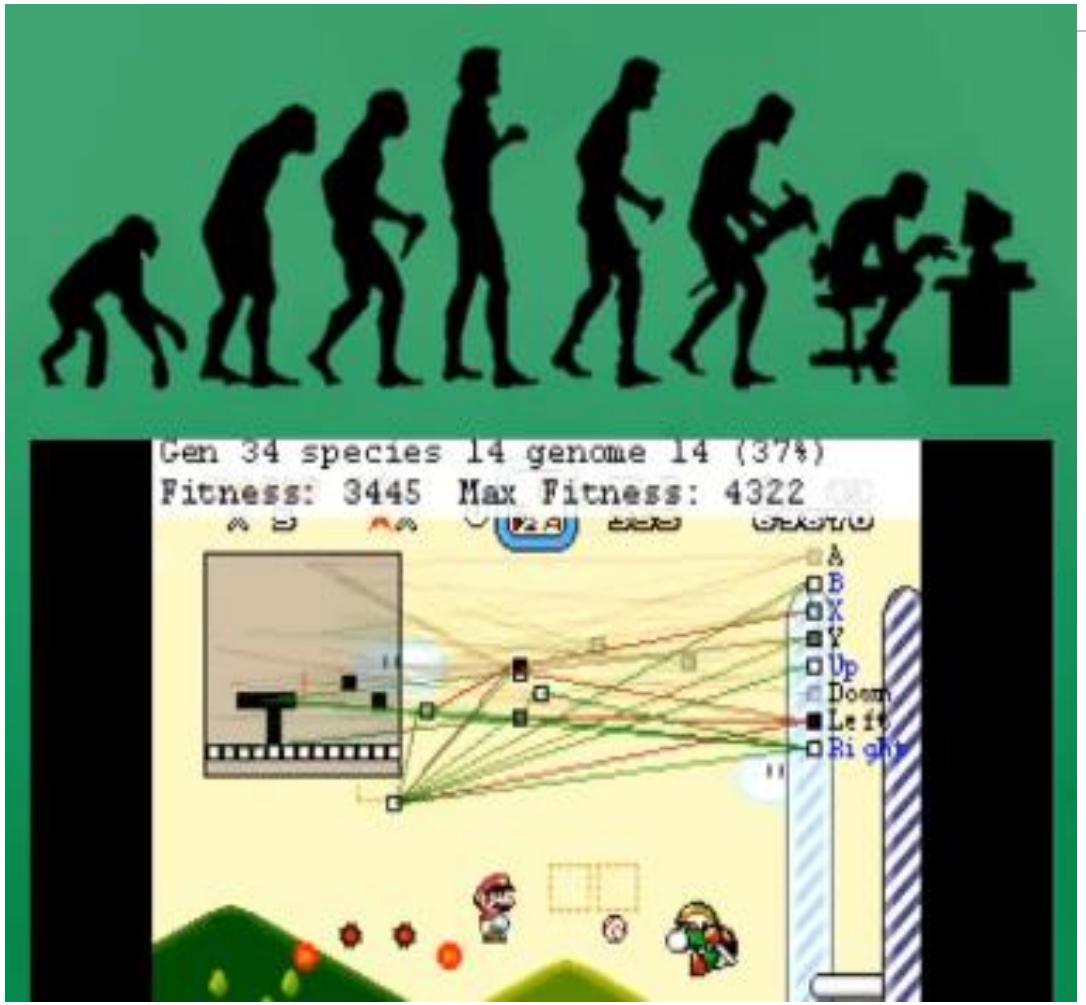
reinforcement learning

hit



miss





genetic algorithm

Simulating the familiar evolutionary theory, the survival of the fittest, and selecting the best design or model through such elimination mechanism

02

Decision tree

- The knowledge of the Decision tree

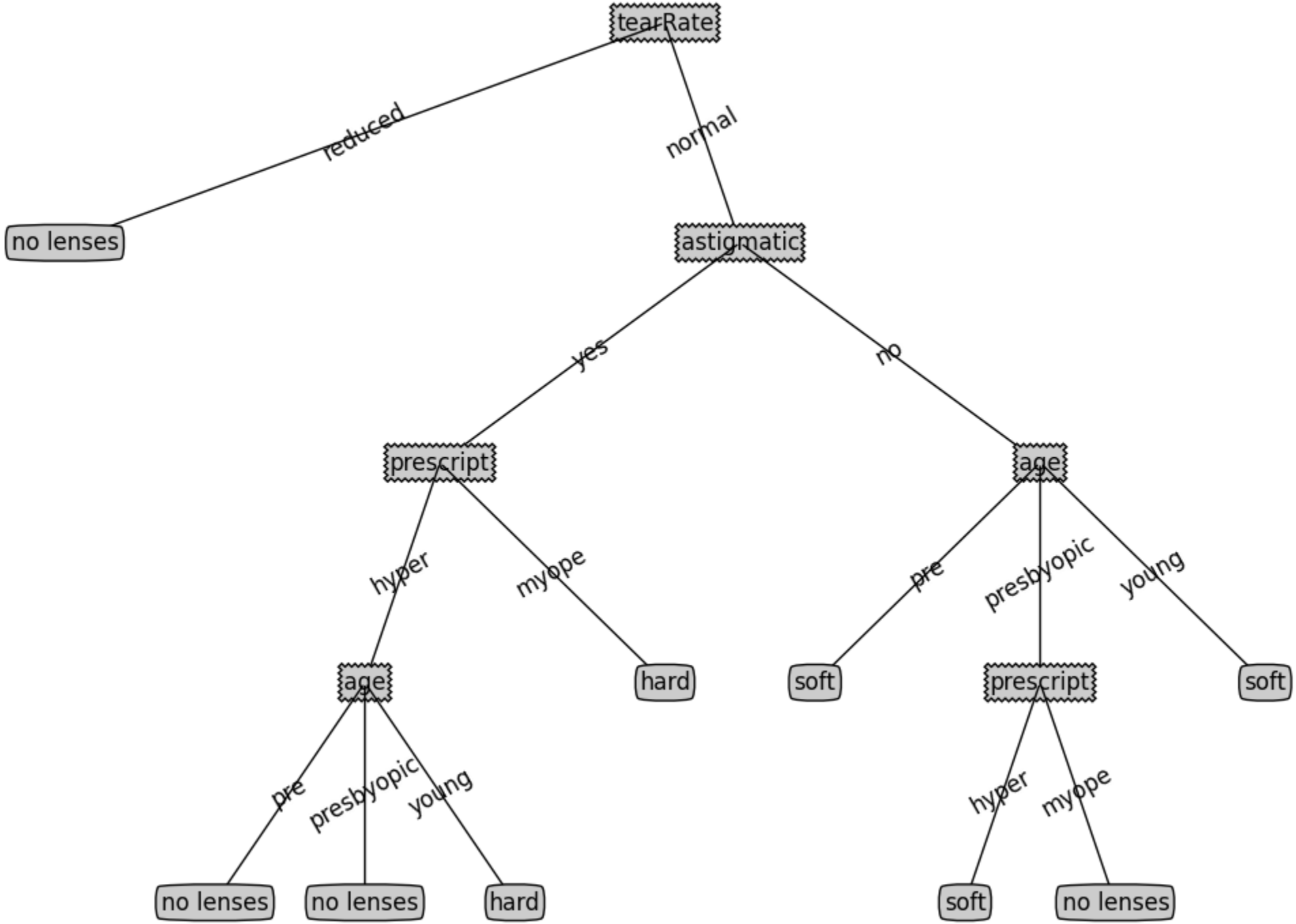
Decision tree

A graphical method of probabilistic analysis. Because this decision branch is graphically shaped like a branch of a tree

The Decision Tree is based on the possible attributes to divide the graphical tree

The decision tree is a tree structure in which represents a test on an attribute, each branch represents a test output, and each leaf node represents a category

Classification tree is a very common classification method. It is a kind of supervised learning, a supervised learning is given a bunch of samples, each sample has a set of attributes and a category, the category is determined in advance, then get a classifier by learning, the classifier to a new object of the correct classification is given. Such machine learning is called supervised learning.


tearRate
reduced
normal
no lenses
astigmatic
yes
no
prescript
age
hyper
myope
pre
presbyopic
young
hard
soft
age
prescript
soft
pre
presbyopic
young
hyper
myope
no lenses
no lenses
hard
soft
no lenses

How do you divide the decision tree

decision tree Use attribute selection metrics to select attributes that best divide tuples into different classes

The key step in constructing a decision tree is to split the attributes. The so-called split attribute refers to constructing different branches at a node according to different divisions of a characteristic attribute, and its goal is to make each split subset as "pure" as possible. To be as "pure" as possible is to try to make a subset of the items to be classified into the same category.

ID3 algorithm

example

ID3 Is the information gain measurement attribute selection, after the selection of the information gain the largest attribute split

The information entropy

$$\mathrm{Ent}(\mathrm{D}) = -\sum_{k=1}^{|y|} p_k \log_2 p_k$$

The smaller the value of Ent(D), the higher the purity of attributes

Information gain

Expect to gain

$$Ent(a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

Assume that the discrete attribute a has V possible values {a1, a2,... .,av}, V branch nodes will be generated

information gain

$$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

continuous attribute division methods

If attribute is a continuous value. At this point, a value is determined as the split point split_point

and two branches are generated according to >split_point and <=split_point.

03

A small example （iris）

- Data classification of iris using DecisionTree

It is mainly divided into the following steps

```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.decomposition import PCA
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
```

Import the required Library

```
path = "./iris.data"
data = pd.read_csv(path,header=None)
iris_feature_E = "sepal length","sepal width","petal length","petal width"
iris_feature_C = u"花萼长度",u"花萼宽度",u"花瓣长度",u"花瓣宽度"
iris_class = "Iris-setosa","Iris-versicolor","Iris-virginica"
```

Import data

It is mainly divided into the following steps

```
x = data[np.arange(0,4)]
y = pd.Categorical(data[4]).codes

x_train1, x_test1, y_train1, y_test1 = train_test_split(x,y,test_size=0.2,random_state=14)
x_train, x_test, y_train, y_test = x_train1, x_test1, y_train1, y_test1

ss = MinMaxScaler()
x_train = ss.fit_transform(x_train,y_train)
x_test = ss.transform(x_test)
```

Defining X, Y variables and standardization

Using SelectKBest to select 3 attributes
that affect the target in the four original feature attributes.

```
ch2 = SelectKBest(chi2,k=3)
x_train = ch2.fit_transform(x_train,y_train)
x_test = ch2.transform(x_test)
select_name_index = ch2.get_support(indices=True)
```

It is mainly divided into the following steps

```
pca = PCA(n_components=2)

x_train = pca.fit_transform(x_train)
x_test = pca.transform(x_test)

clf = DecisionTreeClassifier(criterion="entropy",random_state=0)

clf.fit(x_train,y_train)
y_test_hat = clf.predict(x_test)
```

Building Decision tree and predict the result

It is mainly divided into the following steps

```
from sklearn.externals.six import StringIO
with open("iris.dot",'w') as f:
    f = tree.export_graphviz(clf,out_file=f)
import os
os.unlink('iris.dot')
import pydotplus
dot_data = tree.export_graphviz(clf, out_file = None,
                        filled=True, rounded=True,
                        special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data)
graph.write_pdf("iris.pdf")
```

Output Decision tree

```
print("预测结果",clf.predict(x_test))
print("实际结果",y_test)
print("Score:",clf.score(x_test,y_test))
```

Output results and accuracy

It is mainly divided into the following steps

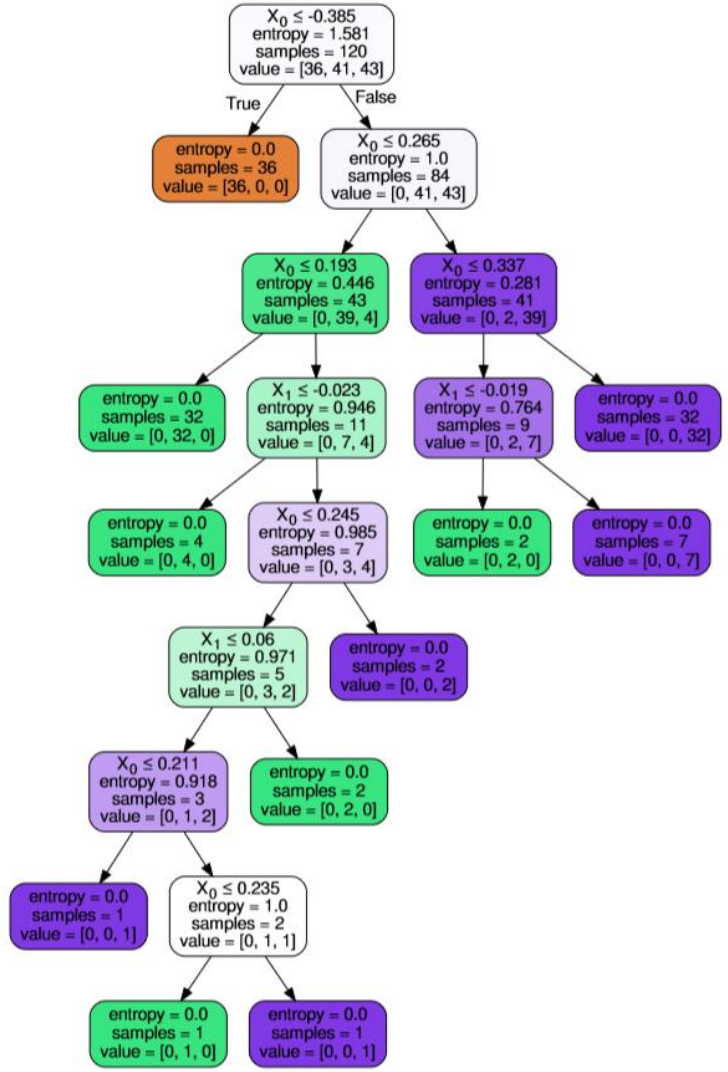
->Running result

```
-bash-4.1$ python project.py

预测结果 [0 0 0 1 2 1 0 1 0 1 1 0 2 2 0 1 0
         2 2 1 0 0 0 1 0 2 0 1 1 0]

实际结果 [0 0 0 1 2 1 0 1 0 1 2 0 2 2 0 1 0
         2 2 1 0 0 0 1 0 2 0 1 1 0]

Score: 0.9666666666666667
```



04

Another example(Signal data)

- Problems encountered in operation and Solutions

Problems encountered in operation and Solutions

Problem 1:

```
Traceback (most recent call last):
  File “root.py”, line 1, in <module>
    import ROOT
  File "/cefs/higgs/marui/root/install/lib/ROOT.py", line 24, in <module>
    import cppyy
  File "/cefs/higgs/marui/root/install/lib/cppyy.py", line 61, in <module>
    import libPyROOT as _backend
ImportError: libpython2.7.so.1.0: cannot open shared object file: No such file or directory
```

Resolvent :

```
#export PATH="/cefs/higgs/marui/anaconda/bin:$PATH“       // python 3.6.5
export PATH=/cefs/higgs/marui/python2.77/bin:$PATH        // python 2.7
```

Problem 2:

```
File "project.py", line 1
SyntaxError: Non-ASCII character '\xe9' in file project.py on line 1, but no encoding
declared; see http://python.org/dev/peps/pep-0263/ for details
```

Resolvent :

Add to the file header: `#-*- coding: UTF-8 -*-`

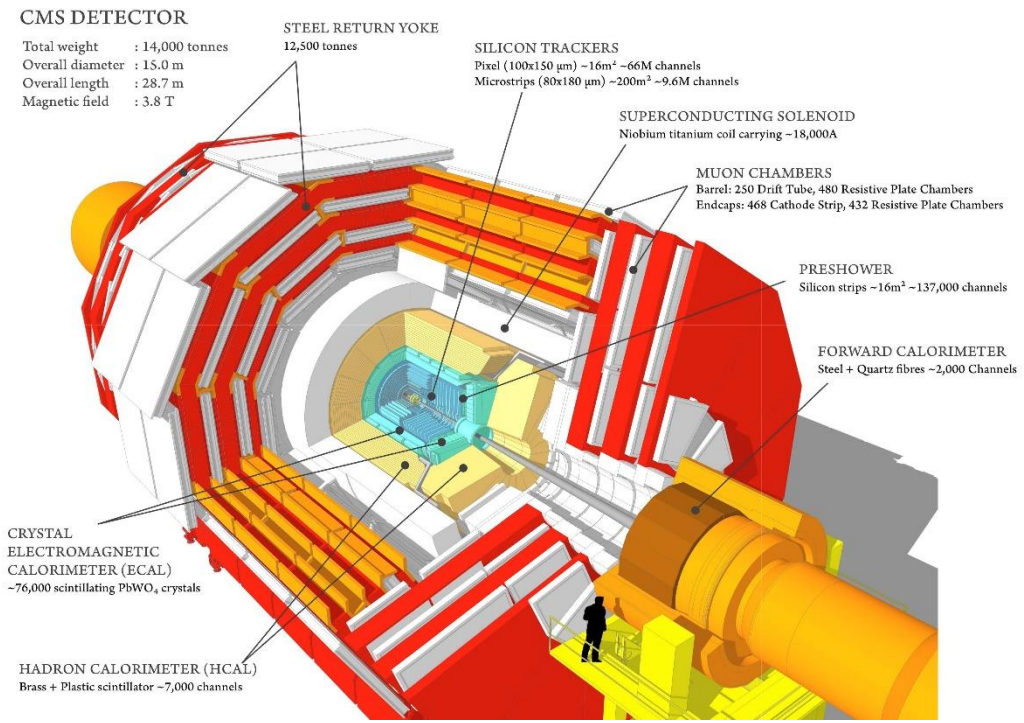
Problems encountered in operation and Solutions

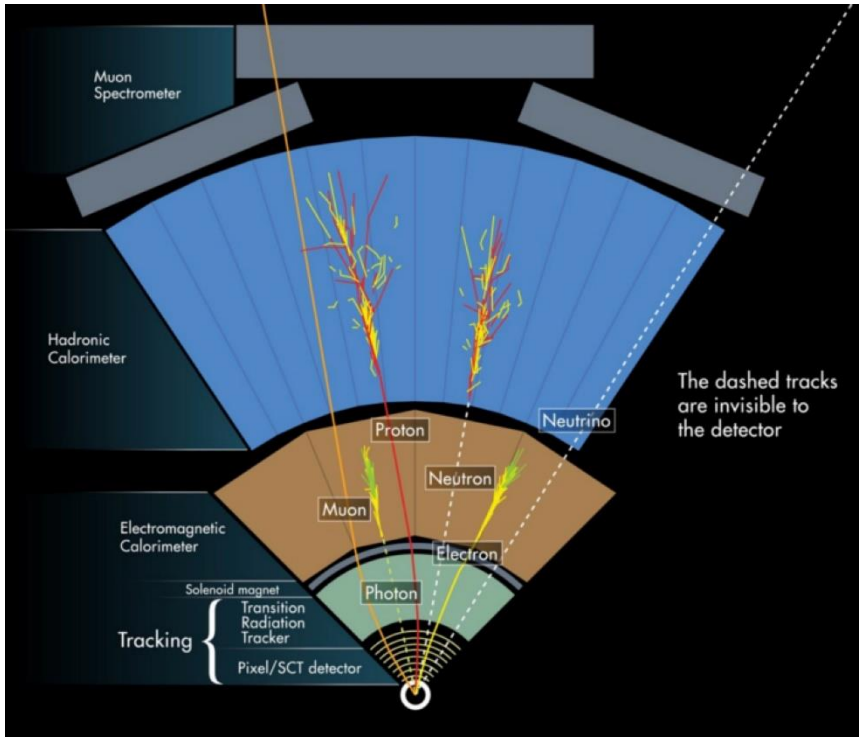
Problem 3:

The foundation is weak, Read the data in the .root file to the python

Resolvent :

Learn some basic knowledge and Search for sentence usage in ROOT official network,and read the data in the .root file to the python is still in progress





Thanks.