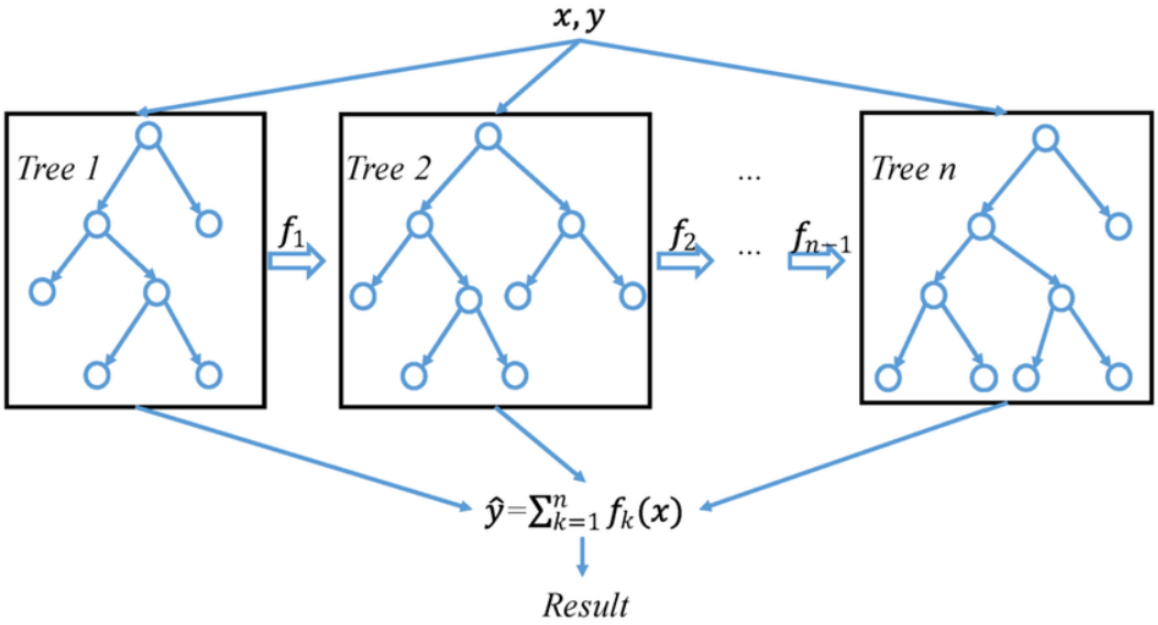
◆ XGBoost Implementation

In this practice, you are going to build an XGBoost model that predicts which passengers survived the Titanic shipwreck. The dataset has been split into two groups, which one is for training and the other is for testing. There are ten different features and one label. Follow the hints below to finish it.



▫ Practice：

■ Analysis datasets：

The dataset consists of two files：train.csv, test.csv

The train.csv file contains the 891 training examples and labels. The test.csv contains 418 test examples and labels. Each row consists of 12 values: the first value is the passenger ID and the second value is the label which mean the person is alive, and remaining 10 values are the training features. You need to notice that some features are characters, and you have to change these features to numbers(One-Hot Encoding or other skills).

Hint 01：Import library you needed.

For example：

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import xgboost as xgb

import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RandomizedSearchCV
```

Hint 02：Use pandas to read data , then fix or drop features if they are not complete.

For example：***The number of NAN in each features***

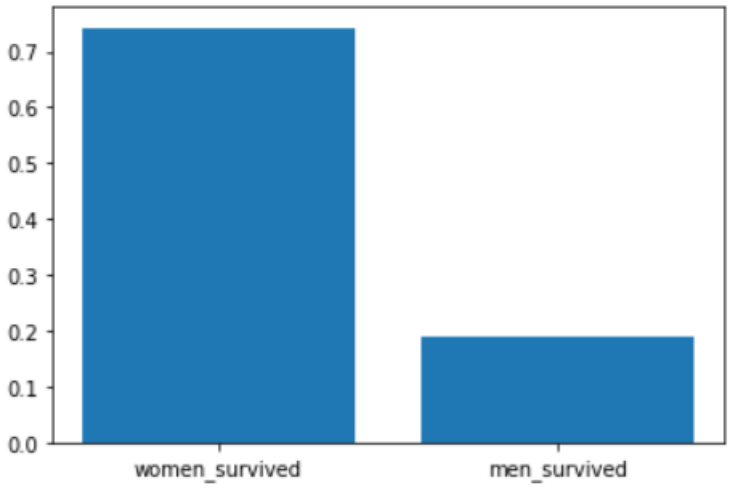
```
train_data.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
```

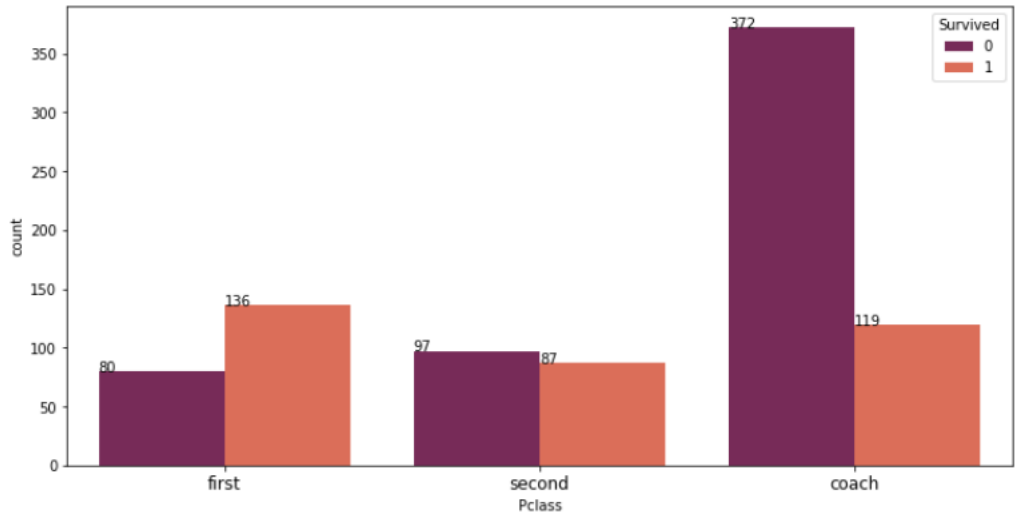
Hint 03：Visualize the dataset, you can choose the features you like to analyze

For example：

```
% of women who survived: 0.7420382165605095
% of men who survived: 0.18890814558058924
```



```
Pclass  Survived
1       1           136
        0            80
2       0            97
        1            87
3       0           372
        1           119
Name: Survived, dtype: int64
```



Hint 04：Doing **One-Hot encoding** or other skills to convert features into numerical form

For example：*Feature "Embark"*

Embarked
2
0
2
2
2
...
2
2
2
0
1

Hint 05：Define a list of parameters, and then use RandomizedSearchCV or GridSerachCV to find a best parameters combination. The parameter of cv is at least 3.

For example：

```
random_grid = {'n_estimators': n_estimators,
               'max_depth': max_depth,
               'learning_rate': learning_rate,
               'colsample_bytree': colsample_bytree}
random_grid
```

```
{'n_estimators': [100, 200, 300],
 'max_depth': [5, 10, 15],
 'learning_rate': [0.01, 0.06, 0.1],
 'colsample_bytree': [0.2, 0.5, 0.8]}
```

```
xgbm = xgb.XGBClassifier(min_child_weight = 1,
                         tree_method='approx',
                         sampling_method = "uniform",
                         eval_metric ="error")

xgbm_random = RandomizedSearchCV(estimator = xgbm,
                                 param_distributions=random_grid,
                                 n_iter=100,
                                 cv=5,
                                 verbose=2,
                                 random_state=np.random.randint(256),
                                 n_jobs=-1)
```

Hint 06：After finding the best parameters, reconstruct a XGBoost model, then use cross_val_score in sklearn to evaluate your model.

For example：

```
xgbm = xgb.XGBClassifier(n_estimators = 200,
                         min_child_weight = 1,
                         tree_method='approx',
                         max_depth = 10,
                         learning_rate = 0.01,
                         sampling_method = "uniform",
                         eval_metric ="error",
                         colsample_bytree=0.8)
```

Hint 07：After training, please show the cross validation score by using cross_val_score in sklearn, then show the testing result by using testing dataset. You should make Average Cross Validation score over 0.8, and the parameter of cv is at least 3.

```
Cross Validation Scores are [0.79329609 0.80446927 0.85955056 0.83707865 0.8700565 ]
Average Cross Validation score :0.8328902147573558
```