```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split


file_x_test = pd.read_csv('x_test.csv', header=None)
file_y_test = pd.read_csv('y_test.csv', header=None)
file_x_train = pd.read_csv('x_train.csv', header=None)
file_y_train = pd.read_csv('y_train.csv', header=None)


column = file_x_test.values[0]
data_x_test = file_x_test.values[1:].astype('float')
data_y_test = file_y_test.values
data_x_train = file_x_train.values[1:].astype('float')
data_y_train = file_y_train.values

test_set_x = pd.DataFrame(data=data_x_test, columns=column, index=None)
test_set_y = pd.DataFrame(data=data_y_test, columns=['label'], index=None)
train_set = pd.DataFrame(data=data_x_train, columns=column, index=None)
train_set['label'] = data_y_train

train_err, test_err = [], []
for i in range(10):
    test_res, train_res = [], []
    for b in range(1, 31):
        bag, rest = train_test_split(train_set, train_size=1/3)
        model = RandomForestClassifier(n_estimators=b, bootstrap=True,
max_features=3)
        model.fit(train_set.drop(['label'], axis=1),train_set['label'])
        train_res.append(1-model.score(train_set.drop(['label'],
axis=1),train_set['label']))
        test_res.append(1-model.score(test_set_x, test_set_y))
    train_err.append(train_res)
    test_err.append(test_res)
train_err = pd.DataFrame(data=train_err, columns=range(1,31), index=None)
test_err = pd.DataFrame(data=test_err, columns=range(1,31), index=None)

mean_err_test, mean_err_train, std_err_test = [], [], []
```

```python
for i in range(1, 31):
    mean_err_test.append(np.mean(test_err[i]))
    mean_err_train.append(np.mean(train_err[i]))
    std_err_test.append(np.std(test_err[i], ddof=1))

plt.figure(1)
plt.title('Mean error rate on test and train set')
plt.plot(range(1, 31), mean_err_train, c='r', label='train')
plt.plot(range(1, 31), mean_err_test, c='b', label='test')
plt.xlabel('b')
plt.ylabel('error rate')
plt.legend()

plt.figure(2)
plt.title('Sample standard deviation of error rate on testing set')
plt.plot(range(1, 31), std_err_test)
plt.xlabel('b')
plt.ylabel('std of sample err_rate ')

plt.show()
```