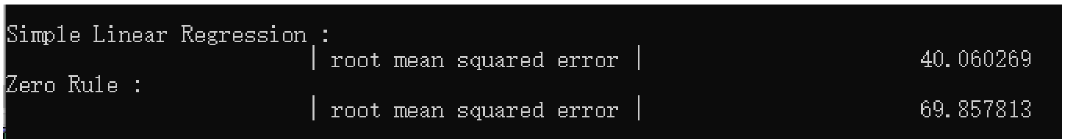
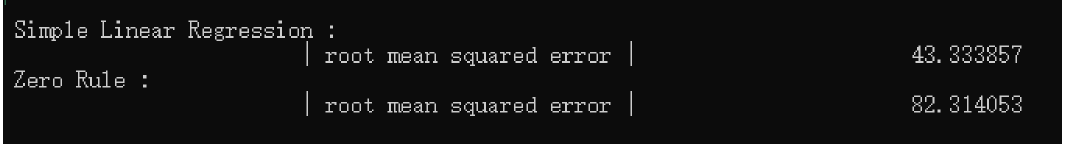
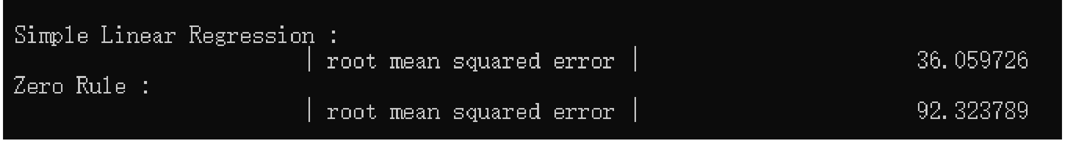
C语言重现机器学习算法---测试说明

1. Simple Linear Regression (20)

- **参数说明**
  - **数据集：Swedish Auto Insurance Dataset**
  - **数据集分割：train-test split， 0.6 : 0.4**
  - **评估指标: root mean squared error**
  - **基准模型: zero rule algorithm**
- **测试结果**
  - **分别用不同的seed(1,39,97),生成不同的训练集-数据集分割多次评估**
  - 

    ```
    Simple Linear Regression :
                                 | root mean squared error |                  40.060269
    Zero Rule :
                                 | root mean squared error |                  69.857813
    ```
  - 

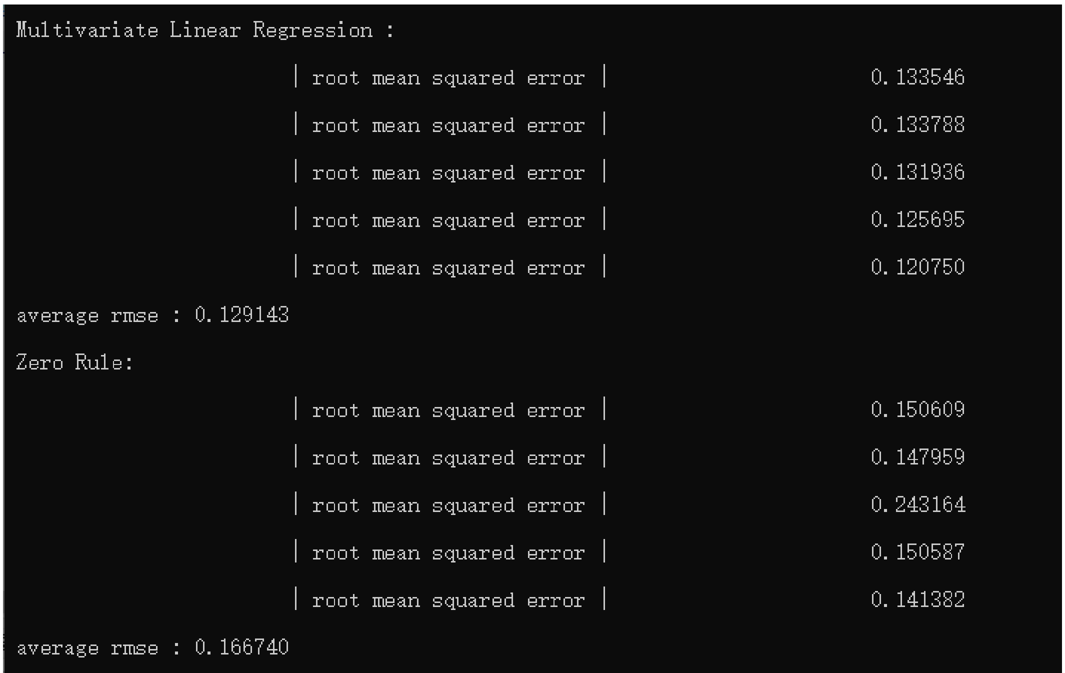
    ```
    Simple Linear Regression :
                                 | root mean squared error |                  43.333857
    Zero Rule :
                                 | root mean squared error |                  82.314053
    ```
  - 

    ```
    Simple Linear Regression :
                                 | root mean squared error |                  36.059726
    Zero Rule :
                                 | root mean squared error |                  92.323789
    ```
- **main.c相关代码**

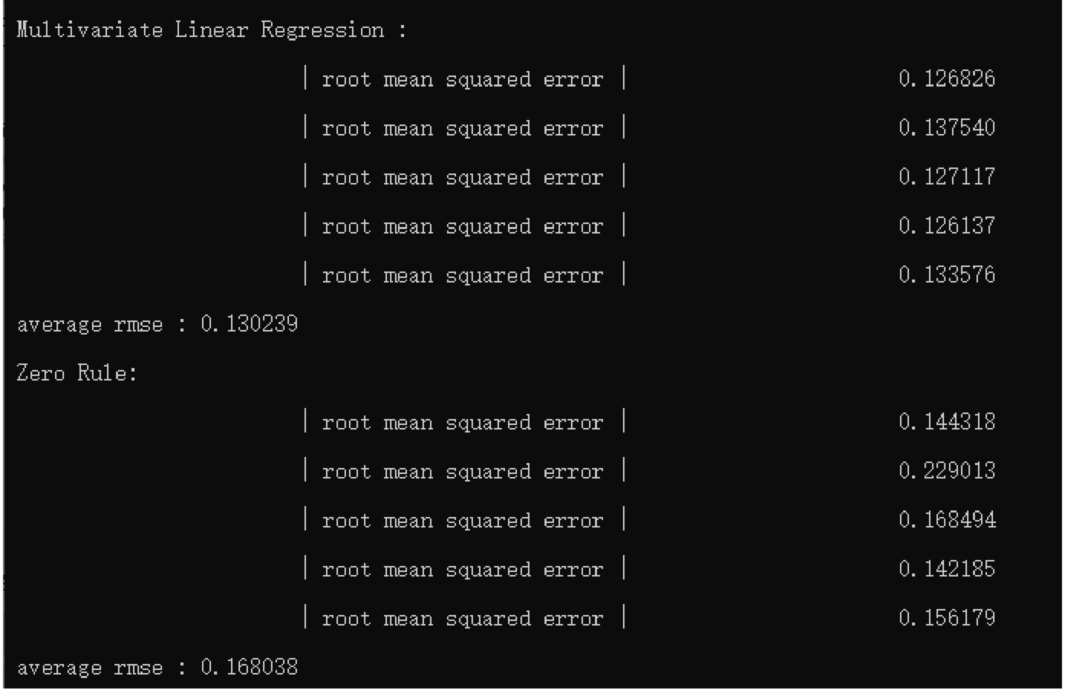
  -
    ```
    train_size = s_rate * row_num;
        rand_index(row_num, 2020, index);

        linear_pre(data, 0, col_num,  index, train_size);
        printf("\n Simple Linear Regression : ");
        rmse_metric( data, col_num, index, train_size);

        printf(" Zero Rule : ");
        zero_rule(data,col_num,0,index,train_size);
        rmse_metric( data, col_num, index, train_size);
    ```

2. Multivariate Linear Regression (25)

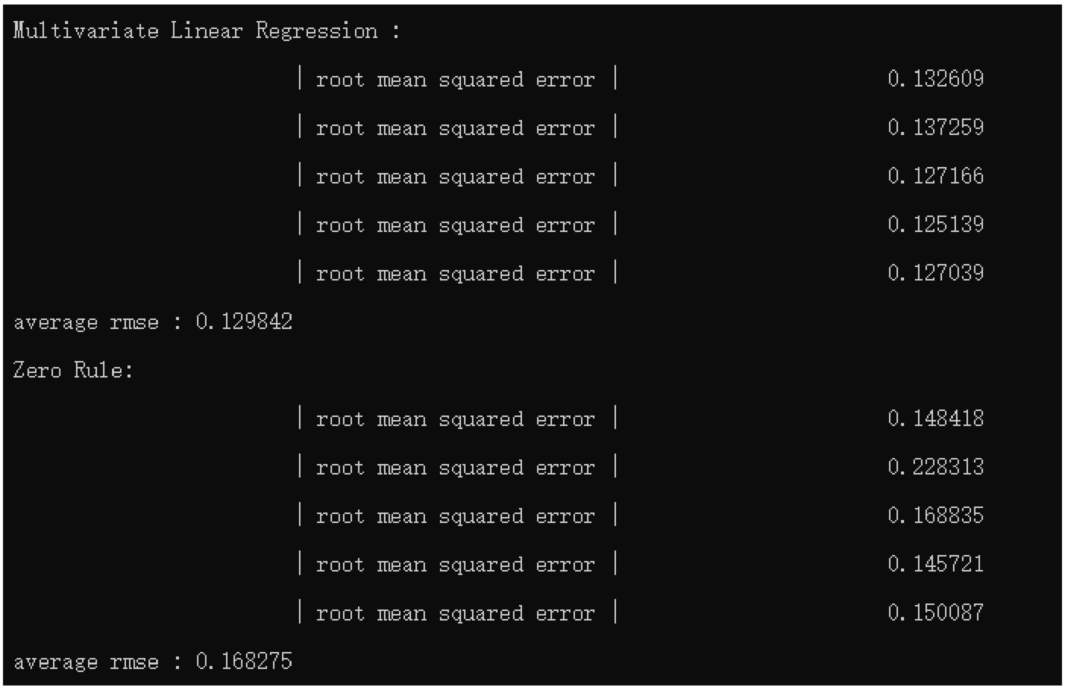
- **参数说明**
  - **数据集：Wine Quality Dataset**
  - **数据集分割：cross validation split， 5-fold**
  - **每组分割训练次数：100**
  - **评估指标: root mean squared error**
  - **基准模型: zero rule algorithm**

- **测试结果**
  - **分别用不同的seed(1,39,97),生成不同的训练集-数据集分割多次评估**
  - 

```
Multivariate Linear Regression :
                        | root mean squared error |                     0.133546
                        | root mean squared error |                     0.133788
                        | root mean squared error |                     0.131936
                        | root mean squared error |                     0.125695
                        | root mean squared error |                     0.120750
average rmse : 0.129143
Zero Rule:
                        | root mean squared error |                     0.150609
                        | root mean squared error |                     0.147959
                        | root mean squared error |                     0.243164
                        | root mean squared error |                     0.150587
                        | root mean squared error |                     0.141382
average rmse : 0.166740
```

  - 

```
Multivariate Linear Regression :
                        | root mean squared error |                     0.126826
                        | root mean squared error |                     0.137540
                        | root mean squared error |                     0.127117
                        | root mean squared error |                     0.126137
                        | root mean squared error |                     0.133576
average rmse : 0.130239
Zero Rule:
                        | root mean squared error |                     0.144318
                        | root mean squared error |                     0.229013
                        | root mean squared error |                     0.168494
                        | root mean squared error |                     0.142185
                        | root mean squared error |                     0.156179
average rmse : 0.168038
```

- 

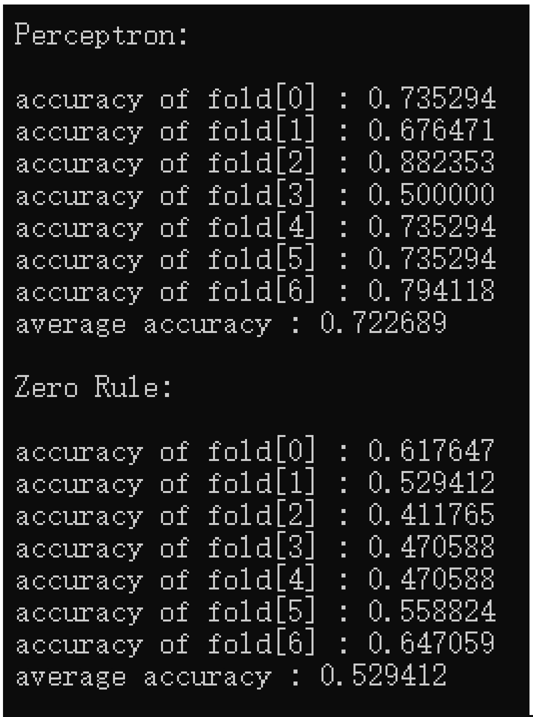
```
Multivariate Linear Regression :
                         | root mean squared error |                    0.132609
                         | root mean squared error |                    0.137259
                         | root mean squared error |                    0.127166
                         | root mean squared error |                    0.125139
                         | root mean squared error |                    0.127039
average rmse : 0.129842
Zero Rule:
                         | root mean squared error |                    0.148418
                         | root mean squared error |                    0.228313
                         | root mean squared error |                    0.168835
                         | root mean squared error |                    0.145721
                         | root mean squared error |                    0.150087
average rmse : 0.168275
```

- **main.c相关代码**

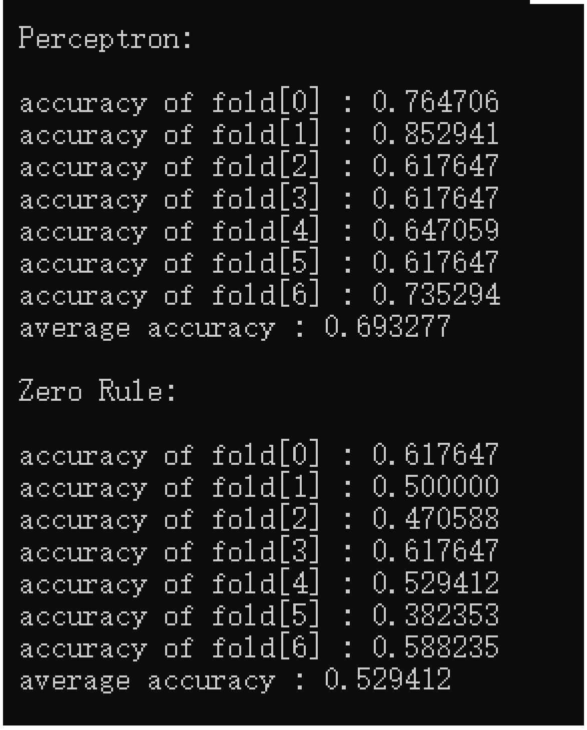
  -
    ```
    normalize_data(data,col_num);

    printf(" \n\n Multivariate Linear Regression : \n");
         mlr_eva(l_rate, epoch, data, col_num, k);

        //生成第一个随机索引数组
        rand_index(row_num, 1, index);
        fold = row_num/k;
        train_size = fold*(k-1);
        printf(" \nZero Rule: \n");
        temp = 0;
        //交叉检验
        for(i=0;i<k;i++)
        {
            zero_rule(data,col_num,0,index, train_size,count);

            temp += rmse_metric(data, col_num, index, train_size);
            next_fold(fold, row_num, index);
        }
        temp /= k;
        printf("\n average rmse : %f \n",temp);
    ```

3. Perceptron (25)

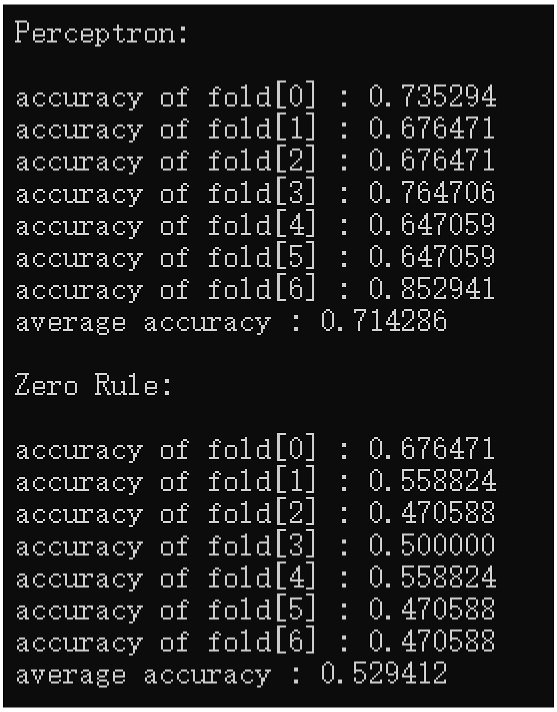
- **参数说明**
  - **数据集：Sonar Dataset**
  - **数据集分割：cross validation split， 7-fold**
  - **学习速率: 0.01**
  - **评估指标: accuracy**
  - **基准模型: zero rule algorithm**

- **测试结果**
    - **分别用不同的seed(1,39,97),生成不同的训练集-数据集分割多次评估**
    - 

```
Perceptron:

accuracy of fold[0] : 0.735294
accuracy of fold[1] : 0.676471
accuracy of fold[2] : 0.882353
accuracy of fold[3] : 0.500000
accuracy of fold[4] : 0.735294
accuracy of fold[5] : 0.735294
accuracy of fold[6] : 0.794118
average accuracy : 0.722689

Zero Rule:

accuracy of fold[0] : 0.617647
accuracy of fold[1] : 0.529412
accuracy of fold[2] : 0.411765
accuracy of fold[3] : 0.470588
accuracy of fold[4] : 0.470588
accuracy of fold[5] : 0.558824
accuracy of fold[6] : 0.647059
average accuracy : 0.529412
```

    - 

```
Perceptron:

accuracy of fold[0] : 0.764706
accuracy of fold[1] : 0.852941
accuracy of fold[2] : 0.617647
accuracy of fold[3] : 0.617647
accuracy of fold[4] : 0.647059
accuracy of fold[5] : 0.617647
accuracy of fold[6] : 0.735294
average accuracy : 0.693277

Zero Rule:

accuracy of fold[0] : 0.617647
accuracy of fold[1] : 0.500000
accuracy of fold[2] : 0.470588
accuracy of fold[3] : 0.617647
accuracy of fold[4] : 0.529412
accuracy of fold[5] : 0.382353
accuracy of fold[6] : 0.588235
average accuracy : 0.529412
```



```
Perceptron:

accuracy of fold[0] : 0.735294
accuracy of fold[1] : 0.676471
accuracy of fold[2] : 0.676471
accuracy of fold[3] : 0.764706
accuracy of fold[4] : 0.647059
accuracy of fold[5] : 0.647059
accuracy of fold[6] : 0.852941
average accuracy : 0.714286

Zero Rule:

accuracy of fold[0] : 0.676471
accuracy of fold[1] : 0.558824
accuracy of fold[2] : 0.470588
accuracy of fold[3] : 0.500000
accuracy of fold[4] : 0.558824
accuracy of fold[5] : 0.470588
accuracy of fold[6] : 0.470588
average accuracy : 0.529412
```

- **main.c相关代码**

```
pcep_eva(l_rate, epoch, data, col_num,k);

    //生成第一个随机索引数组
    rand_index(row_num, 1, index);
    fold = row_num/k;
    train_size = fold*(k-1);
    printf("\n Zero Rule: \n");
    //交叉检验
    for(i=0;i<k;i++)
    {
        zero_rule(data,col_num,1,index, train_size,count);

        acc[i] = accuracy(data,col_num,index,train_size);

        next_fold(fold, row_num, index);
    }

    for(i=0;i<k;i++)
    {
        temp += acc[i];
        printf("\n accuracy of fold[%d] : %f",i,acc[i]);
    }
    temp /= k;
    printf("\n average accuracy : %f \n",temp);
```

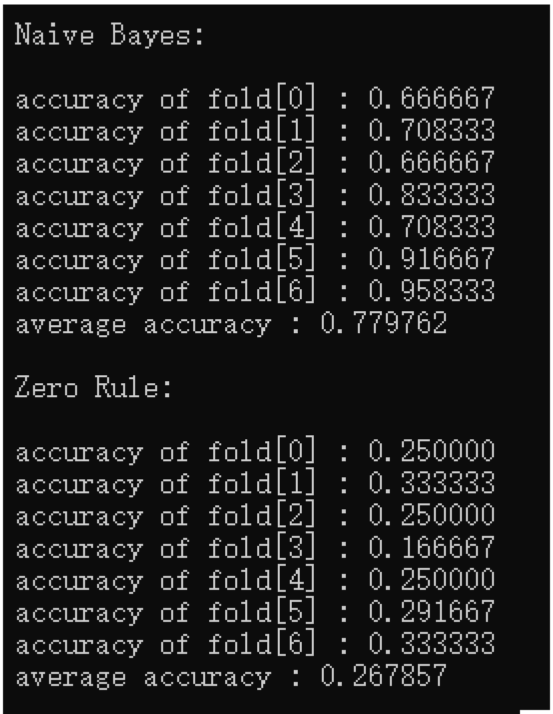
4. Naive Bayes (30)

- **参数说明**
    - **数据集：Iris Flower Species Dataset**
    - **数据集分割：cross validation split， 7-fold**
    - **评估指标: accuracy**

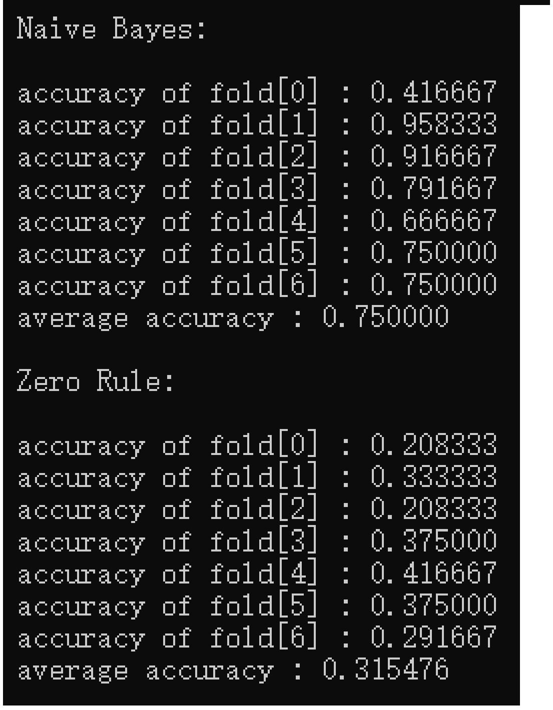
  - **基准模型: zero rule algorithm**

- **测试结果**

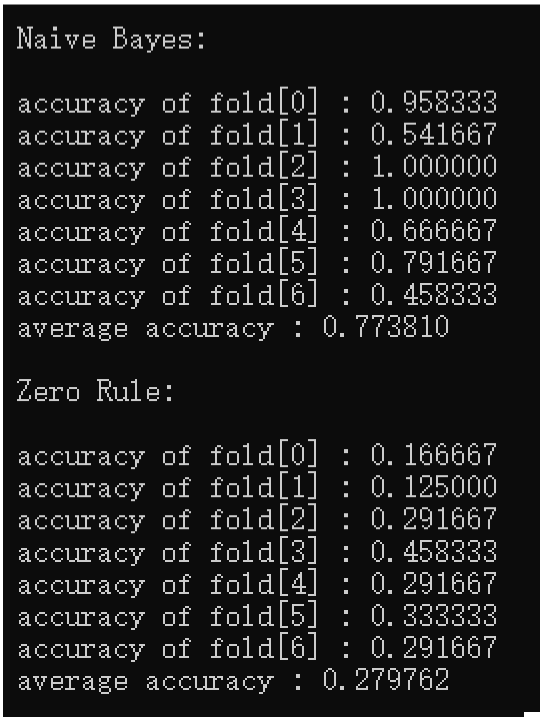
  - **分别用不同的seed(1,39,97,2021),生成不同的训练集-数据集分割多次评估**

  - 

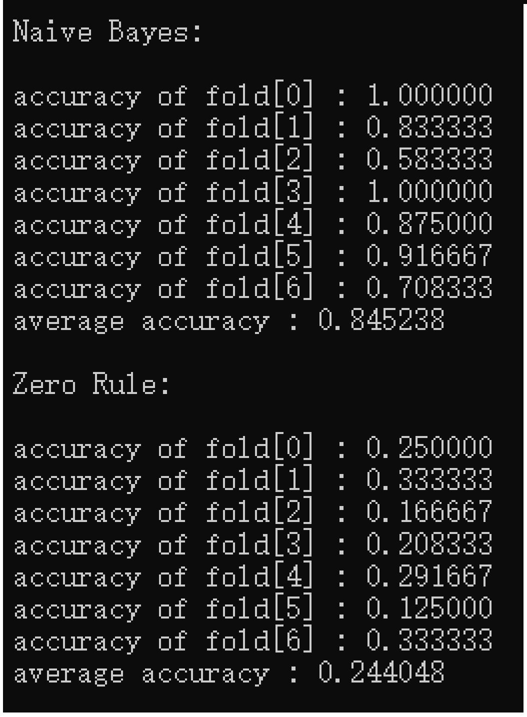
```
Naive Bayes:

accuracy of fold[0] : 0.666667
accuracy of fold[1] : 0.708333
accuracy of fold[2] : 0.666667
accuracy of fold[3] : 0.833333
accuracy of fold[4] : 0.708333
accuracy of fold[5] : 0.916667
accuracy of fold[6] : 0.958333
average accuracy : 0.779762

Zero Rule:

accuracy of fold[0] : 0.250000
accuracy of fold[1] : 0.333333
accuracy of fold[2] : 0.250000
accuracy of fold[3] : 0.166667
accuracy of fold[4] : 0.250000
accuracy of fold[5] : 0.291667
accuracy of fold[6] : 0.333333
average accuracy : 0.267857
```

  - 

```
Naive Bayes:

accuracy of fold[0] : 0.416667
accuracy of fold[1] : 0.958333
accuracy of fold[2] : 0.916667
accuracy of fold[3] : 0.791667
accuracy of fold[4] : 0.666667
accuracy of fold[5] : 0.750000
accuracy of fold[6] : 0.750000
average accuracy : 0.750000

Zero Rule:

accuracy of fold[0] : 0.208333
accuracy of fold[1] : 0.333333
accuracy of fold[2] : 0.208333
accuracy of fold[3] : 0.375000
accuracy of fold[4] : 0.416667
accuracy of fold[5] : 0.375000
accuracy of fold[6] : 0.291667
average accuracy : 0.315476
```

- 

  ```
  Naive Bayes:

  accuracy of fold[0] : 0.958333
  accuracy of fold[1] : 0.541667
  accuracy of fold[2] : 1.000000
  accuracy of fold[3] : 1.000000
  accuracy of fold[4] : 0.666667
  accuracy of fold[5] : 0.791667
  accuracy of fold[6] : 0.458333
  average accuracy : 0.773810

  Zero Rule:

  accuracy of fold[0] : 0.166667
  accuracy of fold[1] : 0.125000
  accuracy of fold[2] : 0.291667
  accuracy of fold[3] : 0.458333
  accuracy of fold[4] : 0.291667
  accuracy of fold[5] : 0.333333
  accuracy of fold[6] : 0.291667
  average accuracy : 0.279762
  ```

- 

  ```
  Naive Bayes:

  accuracy of fold[0] : 1.000000
  accuracy of fold[1] : 0.833333
  accuracy of fold[2] : 0.583333
  accuracy of fold[3] : 1.000000
  accuracy of fold[4] : 0.875000
  accuracy of fold[5] : 0.916667
  accuracy of fold[6] : 0.708333
  average accuracy : 0.845238

  Zero Rule:

  accuracy of fold[0] : 0.250000
  accuracy of fold[1] : 0.333333
  accuracy of fold[2] : 0.166667
  accuracy of fold[3] : 0.208333
  accuracy of fold[4] : 0.291667
  accuracy of fold[5] : 0.125000
  accuracy of fold[6] : 0.333333
  average accuracy : 0.244048
  ```

- **main.c相关代码**

  -
    ```
    bayes_top(data,col_num,k);

        //生成第一个随机索引数组
        rand_index(row_num, 2021, index);
        fold = row_num/k;
        train_size = fold*(k-1);
        printf("\n Zero Rule: \n");
        //交叉检验
        for(i=0;i<k;i++)
        {
            zero_rule(data,col_num,1,index, train_size,count);

            acc[i] = accuracy(data,col_num,index,train_size);

            next_fold(fold, row_num, index);
        }
    ```

```
    for(i=0;i<k;i++)
    {
        temp += acc[i];
        printf("\n accuracy of fold[%d] : %f",i,acc[i]);
    }
    temp /= k;
    printf("\n average accuracy : %f \n",temp);
```