PROJECT 2

AI 2020 Naïve Bayesian Project

Le Van Hung (黎文雄)

Student ID: 0860831

1. Data Preprocessing

Reading data by using pandas.read_excel(data). Because "No" is not one of the features, so drop column "No", after that replace value 2 in column "Gender" by value 0.

```
def import_data(file_name):

    print("import data name: ", file_name)
    data = pd.read_excel(file_name)

    #drop column No
    data.drop('No', axis=1, inplace=True)

    # replace 2 in 'Gender' to 0
    for row_th, data_th in data.iterrows():
        if data_th['Gender'] == 2:
            data.loc[row_th,'Gender'] = 0

    print('data.shape :', data.shape)
    return data
```

With training set, we have 300 data points totally. The original class distribution is 0/1 = 23/7. I will perform Stratified 10-fold Cross validation, in which each fold follows the ratio 23/7. So, in each fold, we have 23 data with class 0 and 7 data with class 1. Function reshape_data(data) will do this job.

```
def reshape_data(data):
    num_class_all = [0 , 0]
    new_data = data
    new_data_0 = []
    new_data_1 = []
    fold_size = len(data)//10
    num_0_per_fold = 23
    num_1_per_fold = 7
    for row_th, data_th in data.iterrows():
        num_class_all[data_th['Target']] += 1

    for row_th, data_th in data.iterrows():
        if data_th['Target'] == 0:
            new_data_0.append(data_th)
        else:
            new_data_1.append(data_th)

    i = 0
    while (i < 10):
        new_data.loc[0+30*i:22+30*i,:] = new_data_0[0+23*i:23+23*i]
        new_data.loc[23+30*i:29+30*i,:] = new_data_1[0+7*i:7+7*i]
        i +=1
    return new_data
```

Because this data set include NaN value only in the numerical feature, so I will replace this NaN value with the mean of this feature by using function get_mean_feature(data, feature)

```
def get_mean_feature(data,feature):
    sum = 0
    count = 0
    for row_th, data_th in data.iterrows():
        value = data_th[feature]
        if np.isnan(value):
            continue
        sum += value
        count +=1
    return sum/count
```

2. Formula

a. Introduction Naïve Bayes' Theorem

In this project, we use Naïve Bayes' Theorem, in simple terms, this theorem will assume that independence among the features in class.

Navie Bayes' Theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c) like this equation below:

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Where:

- P(c|x) is the posterior probability of class (c is target) given predictor (x is attributes or features)
- P(c) is the prior probability of class
- P(x|c) is the likelihood which is the probability of predictor given class. We can easy to calculate the likelihood if we suppose that each feature in class is independent.
- P(x) is the prior probability of predictor.

b. Apply to project

In this project, class c (presented at column 'Target') can receive 2 value: 0 or 1. We have 36 features totally (19 categorical features and 17 numerical features). So, the likelihood P(x|c) can present as:

$$P(x \mid c) = P(x_1 \mid c)P(x_2 \mid c)...P(x_{36} \mid c)$$

With each data in testing set, we need to calculate P(x|c) and P(c) at c =0 and c=1 and compare the result. The predict class can be received by:

$$c = \arg \max_{c} P(x \mid c)P(c)$$

Because, we have lagre features (36 features), so to avoid the error, I use log function to get the prediction.

$$c = \arg \max_{c \in \{0,1\}} \log(P(c)) + \sum_{i=1}^{36} \log(P(x_i \mid c))$$

- **Calculate P(c)**

$$P(c=0)=\frac{number\ of\ class\ 0}{number\ of\ all\ data}\quad ;\quad P(c=1)=\frac{number\ of\ class\ 1}{number\ of\ all\ data}$$

- **Calculate P(x_i|c)**

If xi is value of numerical feature, I use **Gaussian Naive Bayes** to calculate this probability. With **i**th-data x_i and class c (0 or 1), x_i will base on **gaussian distribution** with mean μ_c and variance σ^2_c. P(x_i|c) can be calculated by the below equation:

$$P(x_i \mid c)=\frac{1}{\sqrt{2\pi\sigma^2_c}}\exp(-\frac{(x_i-\mu_c)^2}{2\sigma^2_c})$$

Where:

$$\mu_c=\frac{\sum value\ x_i\ in\ class\ c}{number\ of\ class\ c};\ \sigma^2_c=\frac{\sum(value\ x_i\ in\ class\ c-\mu_c)^2}{number\ of\ class\ c-1}$$

If x_i is value of categorical feature, I use **Multinomial Naïve Bayes** to calculate this probability. I use min_category variable as **minimum of number of x_i in class c** to avoid zero when using log function, fortunately, no zero value appears in this project.

$$P(x_i \mid c)=\frac{number\ of\ x_i\ in\ class\ c}{number\ of\ class\ c}$$

3. Verification method

With training set, we have 300 data points totally. The original class distribution is 0/1 = 23/7. To validate the training set, I perform Stratified 10-fold Cross validation, in which each fold follows the ratio 23/7. So, in each fold, we have 23 data will class 0 and 7 data with class 1.

In each time for training (9 folds for training and 1 fold for validation), Call function **model.reset_model()** to reset model and **model.train_model()** to calculate mean, variance. After that, call function **model.validate_model()** to estimate the accuracy in this validation set. If the accuracy is higher than last time training, call function **model.save_model()** to save this training and testing set.

Do this process in 10-fold CV, we will find the best model and use this for testing.

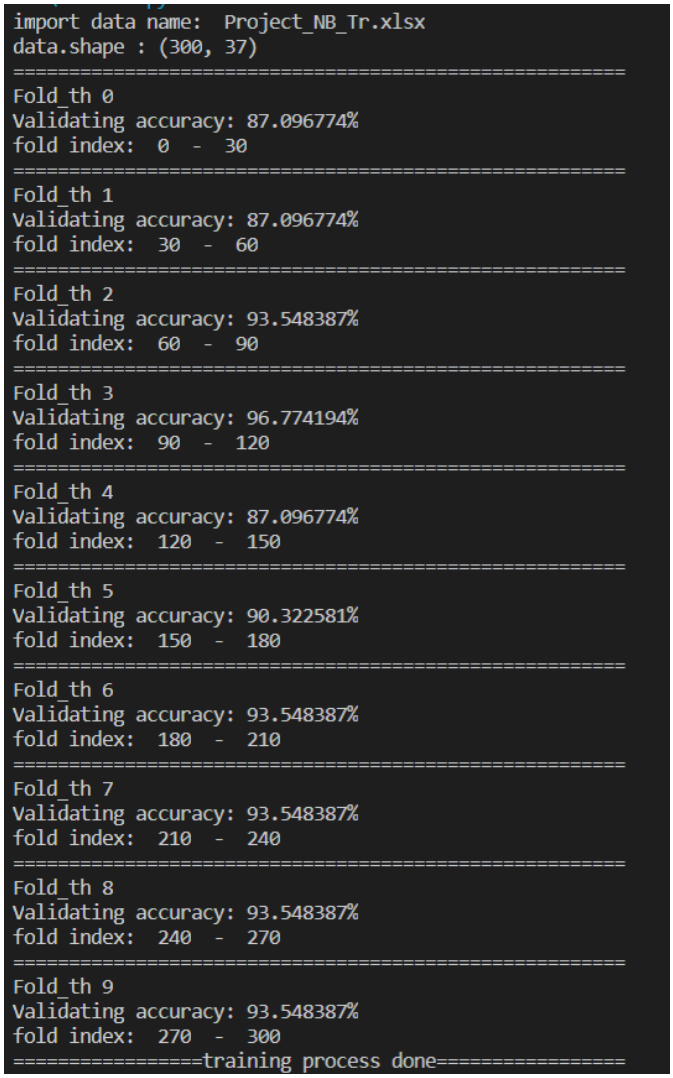
```
model = K_fold_NB()
#check acc in each fold: 10 fold, 30 data each fold
#save model with the best acc
for start_test in range(0, len(training_set)-fold_size+1, fold_size):
    print('==================================================')
    print('Fold_th %d' % (start_test // fold_size))
    model.reset_model()
    model.train_model(training_set, start_test, start_test + fold_size)
    val_acc = model.validate_model(training_set, start_test, start_test + fold_size)

    if best_val_acc < val_acc:
        model.save_model()
        best_val_acc = val_acc

    print("fold index: ", start_test, " - " , start_test+fold_size)

print("================training process done================")
testing_set = import_data(TESTING_FILE)
output_test = model.test_model(testing_set)
for i in range(len(output_test)):
    print('index: ',output_test[i][0]+1, '  predic class: ', output_test[i][1])
```

Result of training and validation process



4. How to run code

My Environment to run:

- Window 10 64 bit
- IDE: Visual Studio Code (VS Code)
- Python 3.7.8 with numpy 1.19.0, pandas 1.0.1 and xlrd 1.2.0 (only use pandas for read file)

Run code:

- 1st step: copy file **Project_NB_Tr.xlsx** and **Project_NB Ts.xlsx** to the same directory with 0860831.py file.
- python 0860831.py or Ctrl+F5 to run
- F5 to debug

P/S: If you do have any problem with my report, don't hesitate to contact me via ***hungle0804@gmail.com***