109 Data Mining Project 02

I. Project objective:

The theme of Project 01 is "data preprocessing" and Project 02 is "model building".
The classifier of Project 02 can only choose "Decision Tree" or "Logistic Regression".
Please choose one and build it by yourself.
Do not copy the source code online or using sklearn model otherwise you will get zero points.

II. Language:

You can only use Python, C, C++ or Java to finish this project.

III. Dataset:

Project 02 will continue the theme of Project 01. We will use the same data type but different data set.
「Info」: 1 row represents a patient.
共病症_Comorbidities」: A number indicates that the patient has a comorbid condition, and the label "0" means that there is no comorbid condition. If one patient has (1,2,4) in this feature, it means this patient has comorbidity 1, comorbidity 2 and comorbidity 4 at the same time.
「TPR」: Time series data of "temperature, pulse, breathing rate, systolic blood pressure, diastolic blood pressure". The same "No" means the data of the same patient.

IV. Hint for Experience Step:

Considering that some students didn't get a good grade in Project 01, we decided to provide a relatively appropriate experiment flow to ensure that Project 02 will not be too much affected by Project 01.
However, if you have confidence in your data preprocessing and experiment design, you can just ignore this part. But if your validation result is far from the testing data result, please refer to the following tips.

1. Combine TPR sheet and Info sheet:

Since the submission format is based on the patient as a unit, you may do some preprocessing to make TPR merge with Info. You can use sampling or statistics (mean、min、max…) to transform TPR data into a fixed shape. Then you can merge TPR and Info by "No".

2. Data Transformed:

TPR is more important than Info, so you should put more effort on handling TPR sheet.
As for Info sheet, you can apply "one-hot encoding" for some columns, and do feature selection to reduce dimension.

3. Validation:

Model evaluation is also an important part of the experiment. A good validation method can roughly estimate the real score on testing data, and let you tune your model and select features.
First, you must ensure that your data is transformed into the shape based on the patient as a unit.
Then you can do shuffling and K-fold CV to evaluate your performance. We suggest K=5. Because if K is too large, the validation set will be too small. When K is too small, your training set will be not enough.
Special reminder here, in Project 01, some students shuffle the entire TPR sheet. This action will disperse the measurement values of the same patient on different days in the Training data and Validation data, which will lead to overestimation due to data leakage. If your Project 01 validation F1-Score> 0.8 but the testing result is very low, there is a very high chance that you have made this mistake.

Last but not least, this dataset is imbalanced (about 1:3), so only focusing on Accuracy is meaningless.
And we have shown that F1-score is used as a basis for grading. Hence, you should validate by F1-score at least. A normal validation F1-score in Project1 should be between 0.4~0.7.
If you get more than 0.8 (F1-score) on validation, it must be something wrong!

V. Model Implementation:

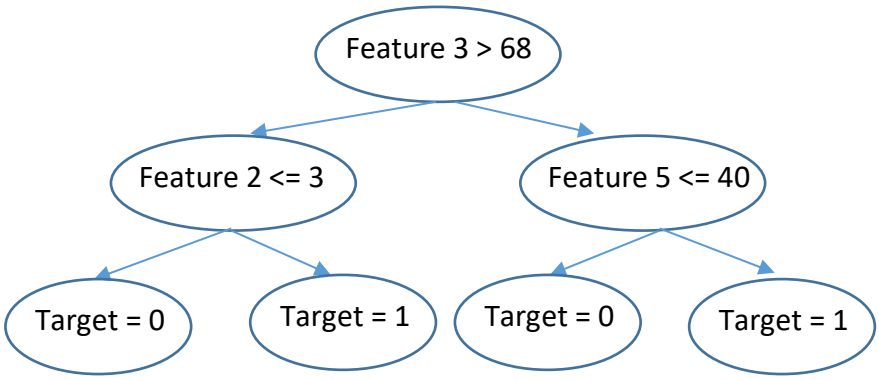
The classifier of Project 02 can only choose Decision Tree or Logistic Regression, please choose one and build by yourself. If you do both, we will only take the highest score as the result of Project 02. Do not copy the source code online or using sklearn model otherwise you will get zero points.

*The scores marked below are all semester total scores; (7%) = 7 points of semester total score.

- Decision Tree（7%）：

Basic：

1. fit (1%): Using training data to generate a tree model.
   You can use gini/entropy/others as the impurity metrics.
   In addition, there must be at least one hyperparameter "max_depth" to limit the depth of your tree model.
   Input: training data, hyperparameter **Output:** model's parameters
2. Predict (2%): Basic classification function, which can predict the label of testing data.
   Input: model, training data **Output:** predict label, either 1 or 0.
3. Predict_probability (2%)：Advanced classification function, which can predict the probability of each class for each testing data.
   Input: model, testing data **Output:** the probability of (target=1).
4. Visualize (2%)：Print out your tree model. It must contain "Feature name, cutting point, leaf " as shown in figure below.
   How to draw circles and arrows is not the focus of this project. You can first use the program to generate the content of each node, and then use PPT, word or other auxiliary tools to complete the drawing.
   Remember to attach the results to the Report.



Bonus：

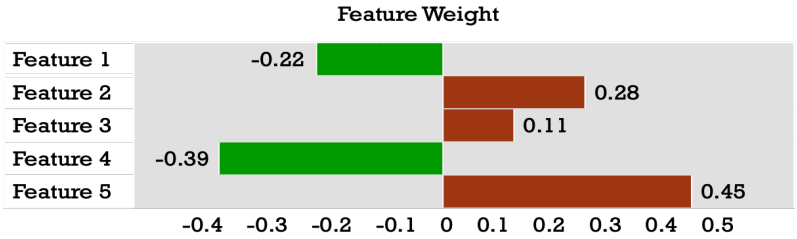
1. Pruning (1%):
   If you use "Pre-pruning" you need to provide a hyperparameter that specifies the "condition" and annotate the hyperparameter name and the visual comparison figure before and after pruning in the Report. If you use Post-pruning, please provide a visual comparison figure before and after pruning in the Report.
2. Dealing with missing values (1%)：
   Do not directly fill in the average or mode, you must use the DT's method to handle missing values.
3. Random Forest (1%):
   Must have a hyperparameter that specifies the "number of trees", and the hyperparameter name should be noted in the Report.

- Logistic Regression (7%)：

Basic:

1. fit (1%): Using training data to generate a model.
   Input: training data **Output:** model's parameters
2. Predict (2%): basic classification function, which can predict the label of testing data.
   Input: model,training data **Output**: predict label either 1 or 0.
3. Predict_probability (2%)：advanced classification function, which can predict the probability of each class for each testing data.
   Input: model, testing data **Output**: the probability of (target=1).

4. Visualize (2%)：Print out your tree model. The weight is equivalent to taking Exponential to the equation coefficient found by LR. The positive and negative values of the Feature weight indicate that the change of the Feature will make the result tend to P(y=1) or P(y=0). The picture should be marked with the coefficient and the name of the feature, as shown in the example below, but it is not necessary to color it.



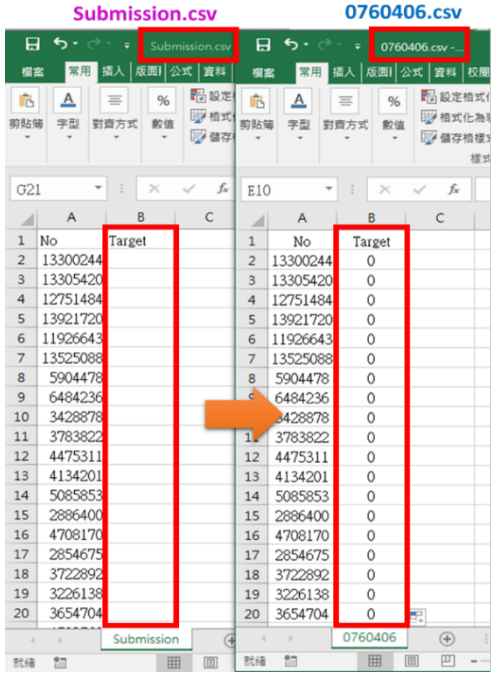
Bonus:

1. Regularization (1%): Adding regularization on cost function.
   Please provide a visual comparison figure before and after pruning in the Report.
2. Ensemble (1%): Must have a hyperparameter that specifies the "number of models", and the hyperparameter name should be noted in the Report.

VI. Submitted file name and format:

1. <your-student ID>.csv

- This file is your predict result for test data.
- Please rename your "Submission.csv" to "<your-student ID>.csv"
  EX: 0760406.csv
- 「Target」column only can be 0 or 1.
- The entire CSV file can only have two columns.



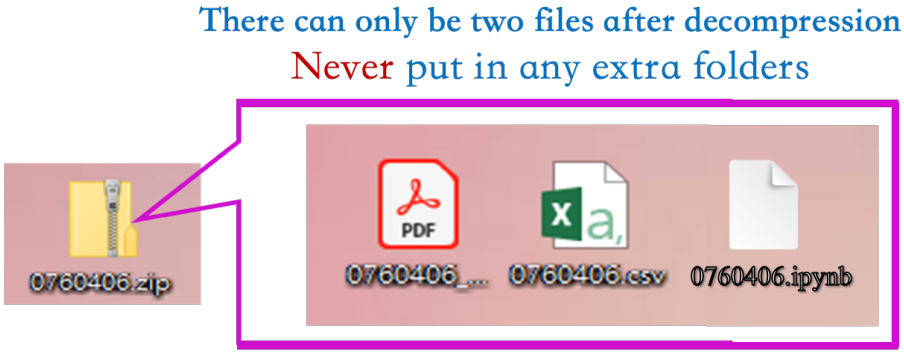
2. <your-student ID>_Report.pdf

- File name: <your-student ID>_Report.pdf
- EX: 0760406_Report.pdf
- Size: 12
- Font：Times New Roman
- Include: Write "ReadMe" first, explaining how to execute the program and configure the parameters. Other content must include "Data preprocessing", "Formula", "Validation method". Can be increased if necessary.

3. < **your-student ID** >.****
   This is your source code file.
   Please note that it cannot be an exe, it must be a file format where the code can be seen.
   After the TA runs the program, the output needs to be able to see at least one validation result that is the same as in the report.

VII. Submission:

- Please upload the zip to e-Campus system "Project_01".



VIII. Rule

Items that will lose points	Project Score
Any wrong file name	-20%
Any format error	-20%
Late project per day	-20%
Use classifiers other than LR or DT or Bonus	-100%
Classifier Call package or copy the source code online	-100%

Item	Total score for the entire semester
F1_score (base on Target=1)	7%
Model	7%
Project Report	7%

$$*F1-score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} = \frac{2TP}{(2TP+FP+FN)}$$

$TP: True\ Positive \quad TN: True\ Negative \quad FP: False\ Positive$

IX. Deadline

- 2021/01/08 (Fri.) AM 12:00 TA will upload Test file
- 2021/01/10 (Sun.) PM 11:50 upload deadline.