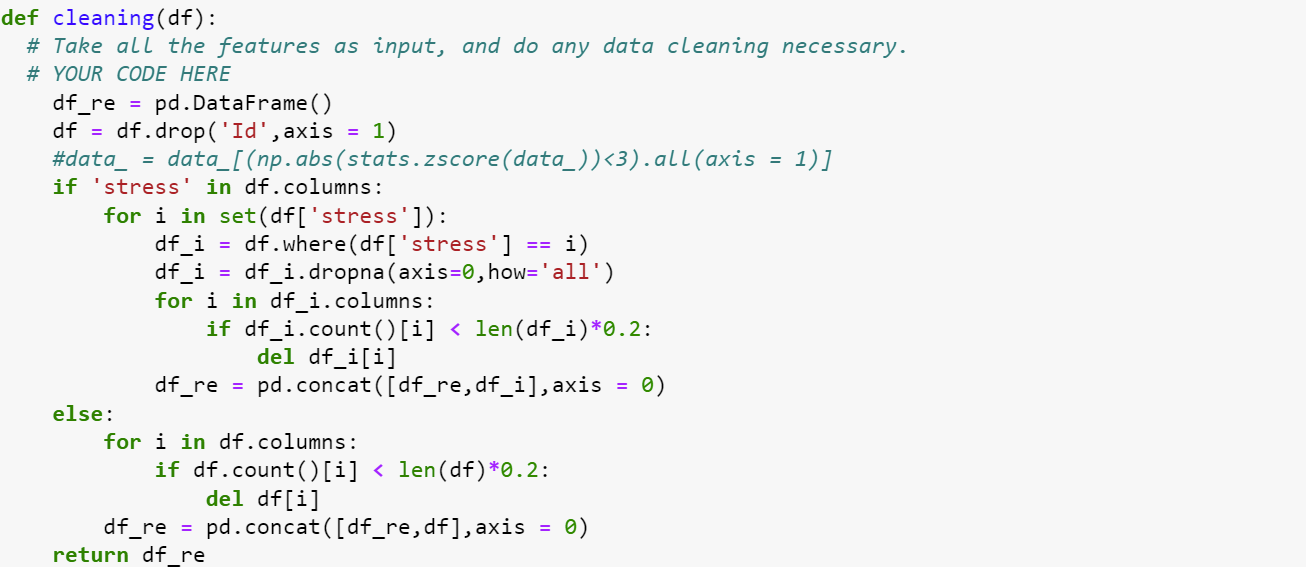
**Report**

**Programming Assignment 2**

**Xiangyu Gao**

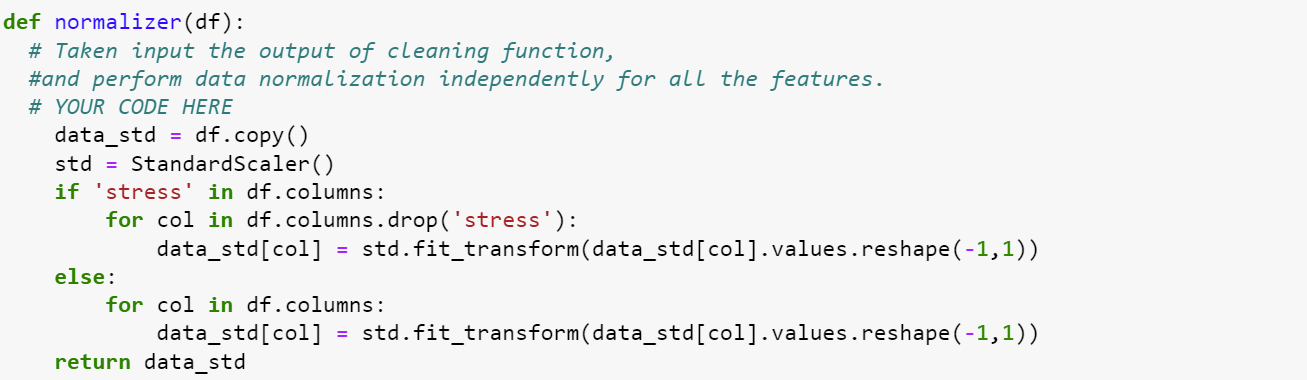
**Answer 1) (20 points)**

* 1. **Cleaning**



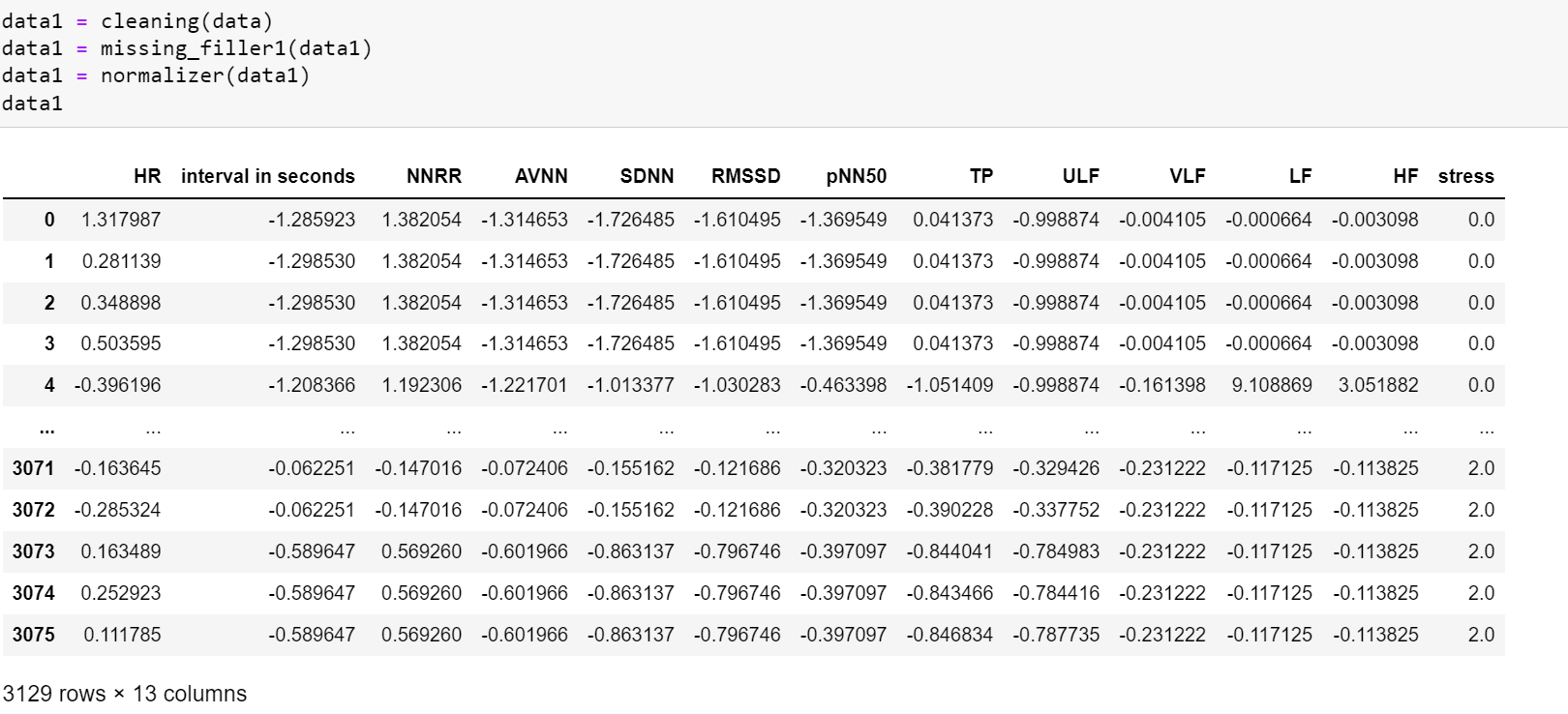
In this part, I dropped Ids in the data, which are the unique features of the samples. I suppose that this kind of feature is useless and unpractical. Furthermore, I used a for-loop in the function, to fill the missing data by groups. So that the samples that have the same label will have the features in the same format. Then, in each group, I removed the columns in which the number of samples has less than 20% of the total.

* 1. **Normalizer**



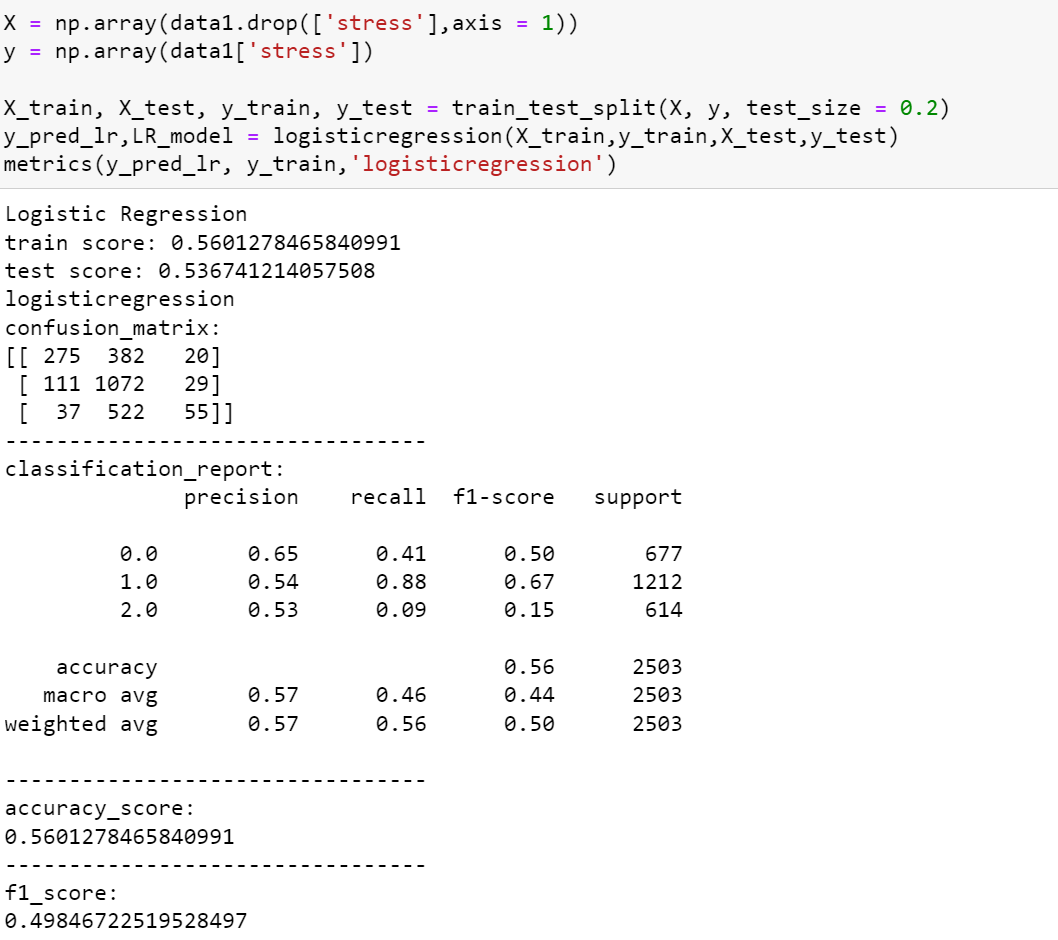
I used the StandardScaler function to normalize.

* 1. **Discussion**

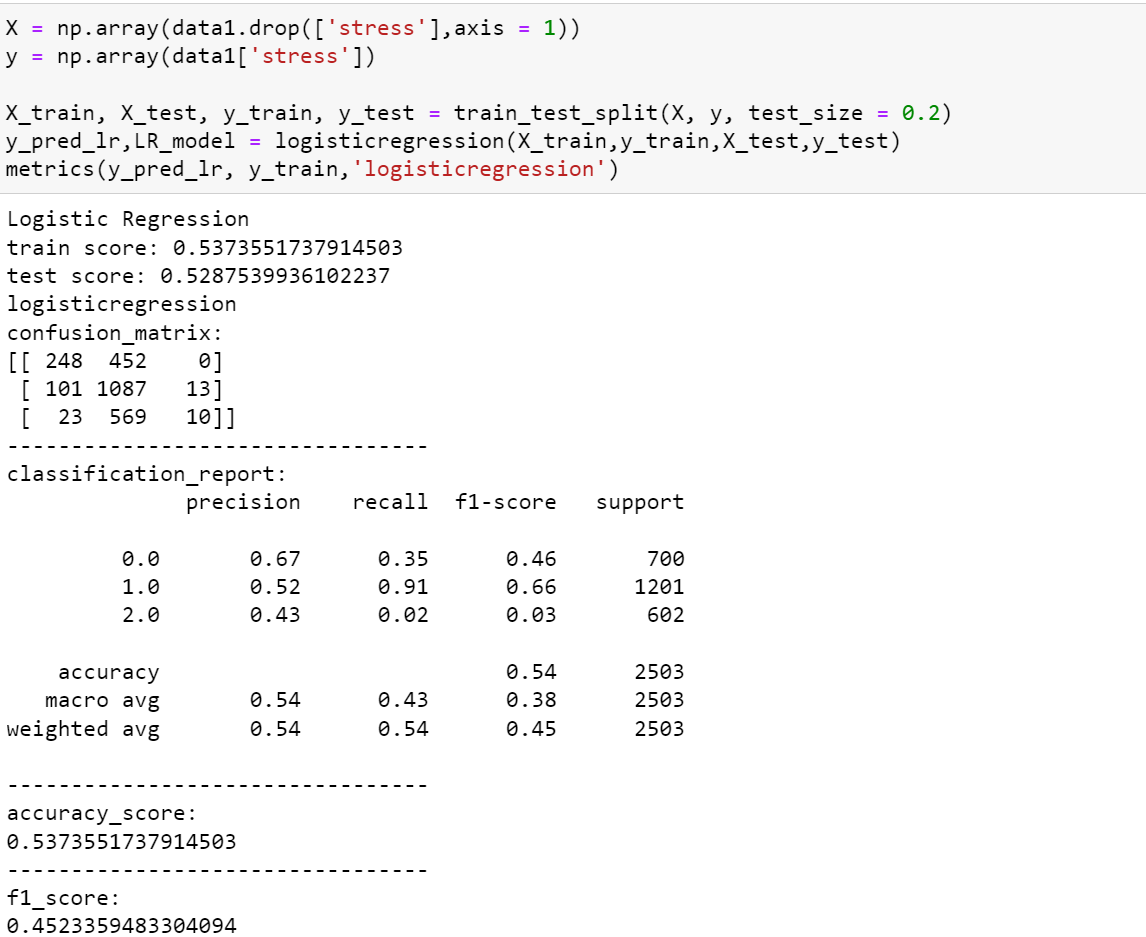


The dataset has been cleaned, filled, and normalized. The feature LF\_HF has been removed since it has less than 20% data of the total.

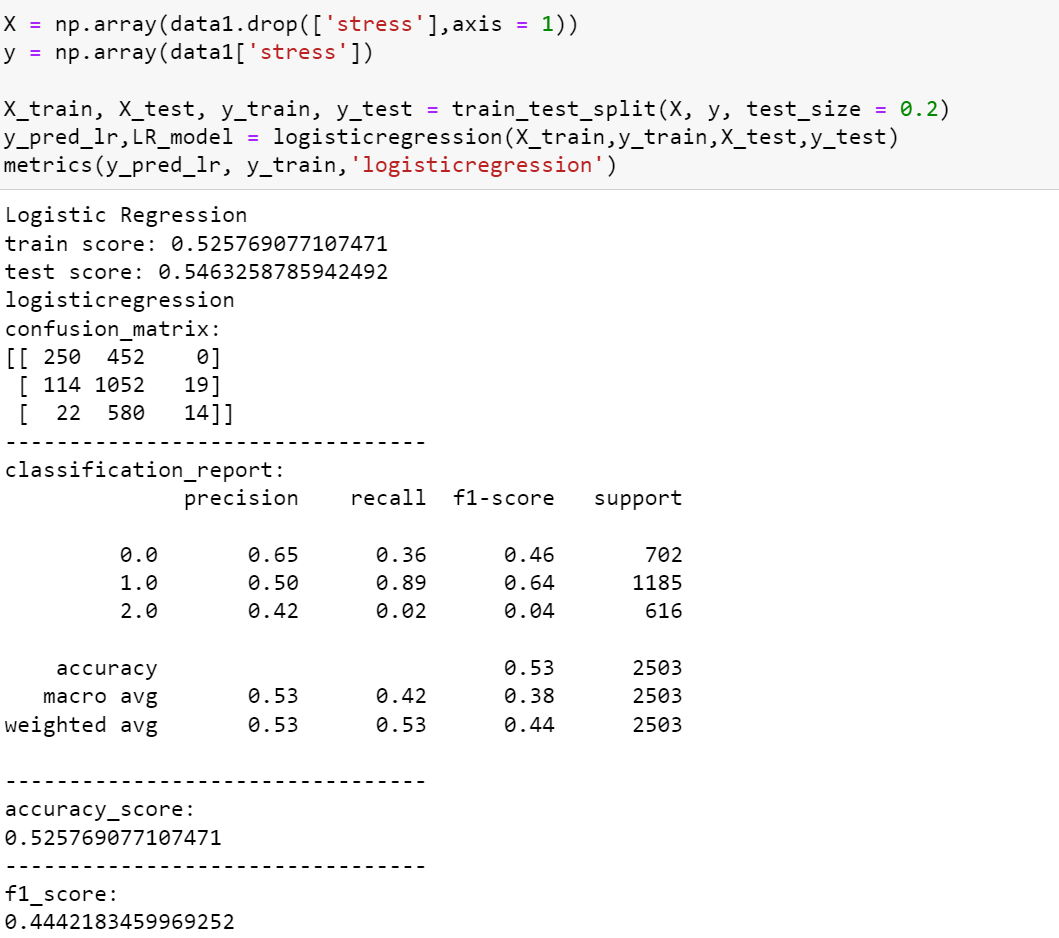
The F1 score is 0.49 after doing these things.



If we do not normalize it, the f1 score will be lower than before: 0.45



If we do not clean it, the f1 score will be 0.44



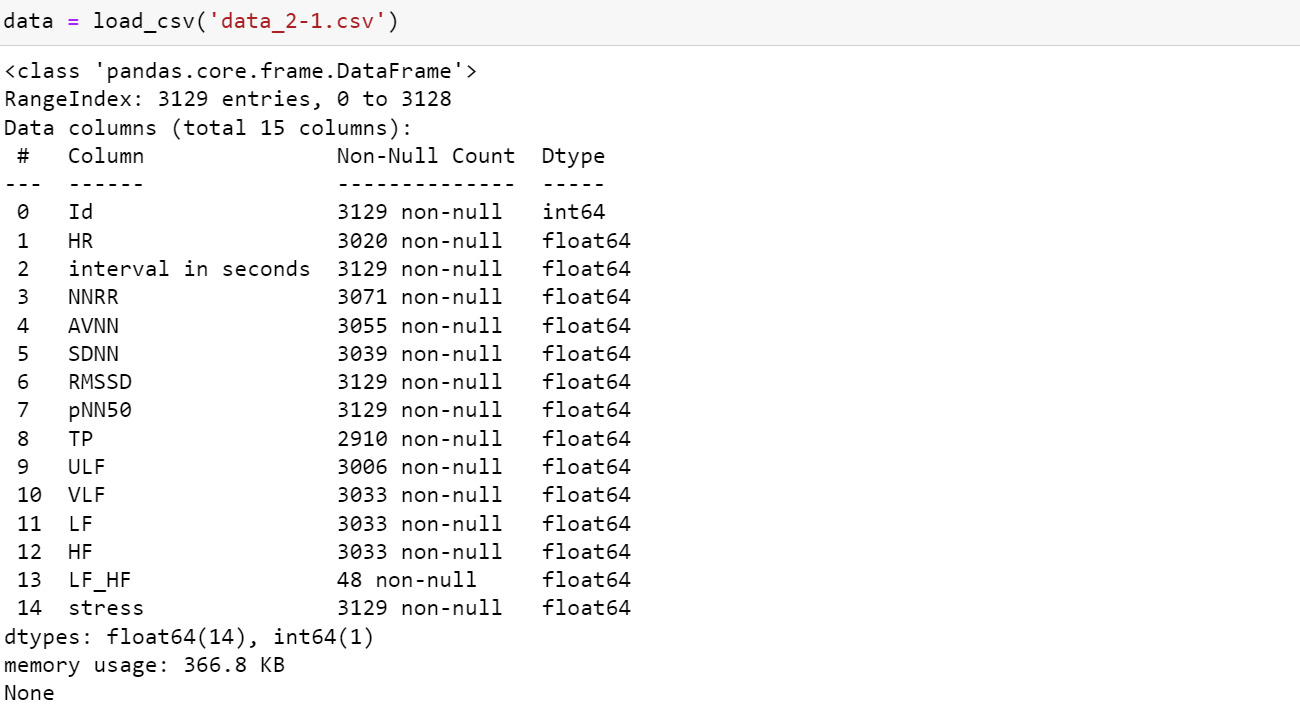
**Answer 2) (15 points)**

# Briefly describe how much data is missing in the dataset provided.

# How did you handle the missing data? Discuss at least 2 different methods used over here,

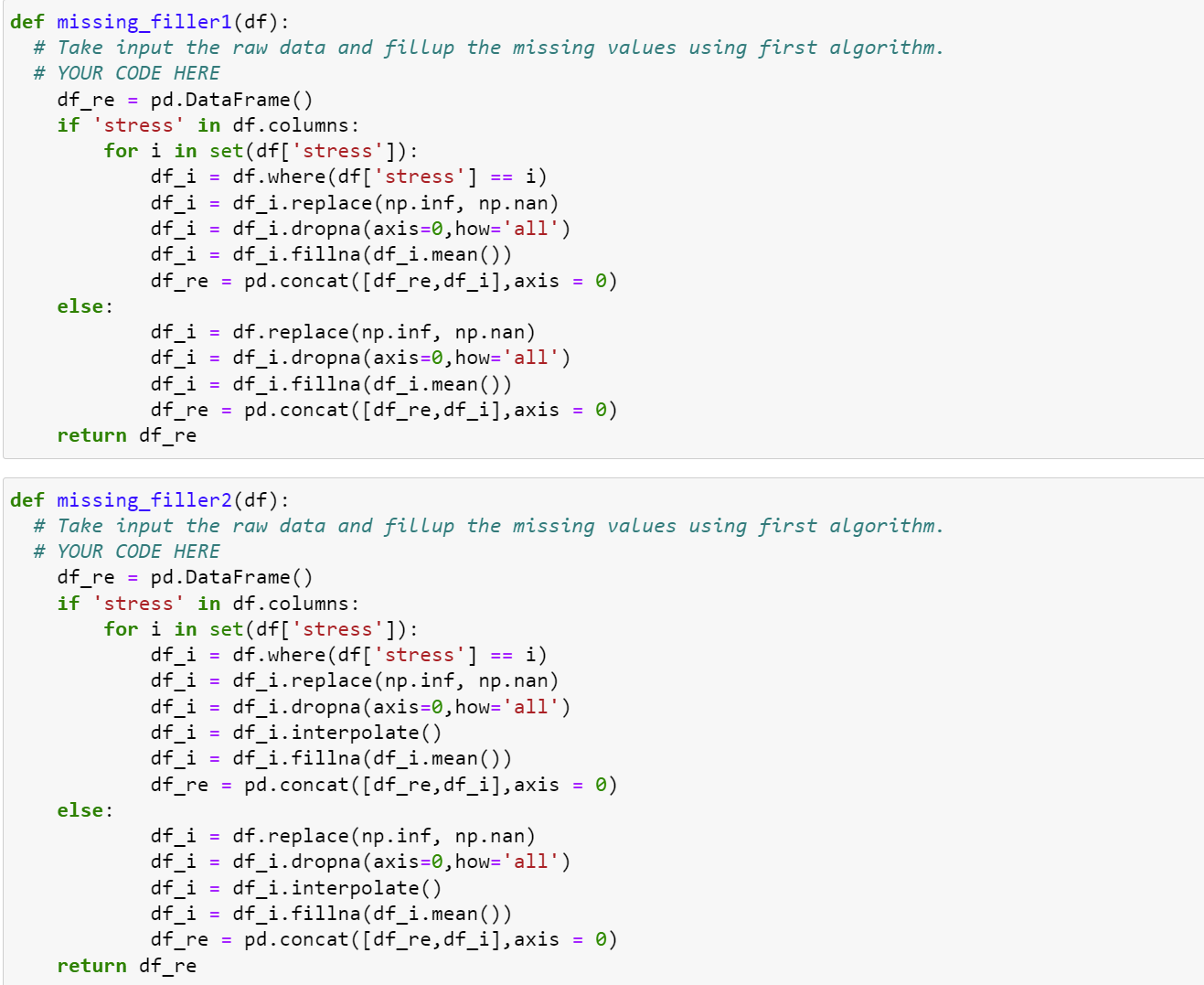
and which one gives better increment in metrics and why?

**2.1 dataset**

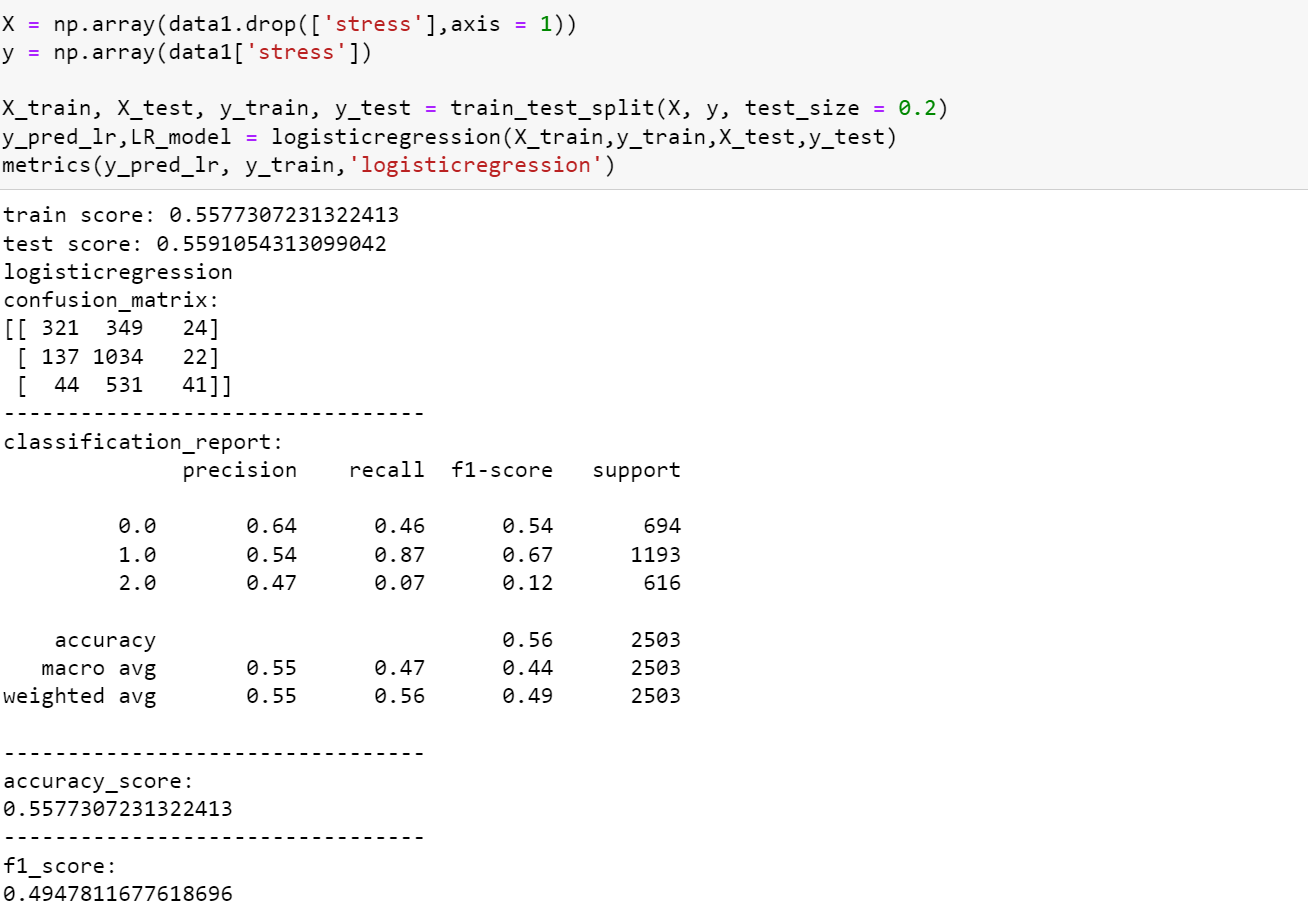


This dataset, like the picture above, has 14 kinds of features: Id, HR, the interval in seconds, NNRR, AVNN, SDNN, RMSSD, pNN50, TP, ULF, VLF, LF, HF, and LF\_HF. The label is stress. As we can see, some features are intact: Id, the interval in seconds, RMSSD, and pNN50, while some are incomplete up to 7% lost, like HR, NNRR, AVNN, SDNN, TP, ULF, VLF, LF, and HF. What is worth to be noticed is that LF\_HF only has 1.5% of the total: 48 samples, which is too small to be an effective feature.

**2.2 Missing Filler 1**

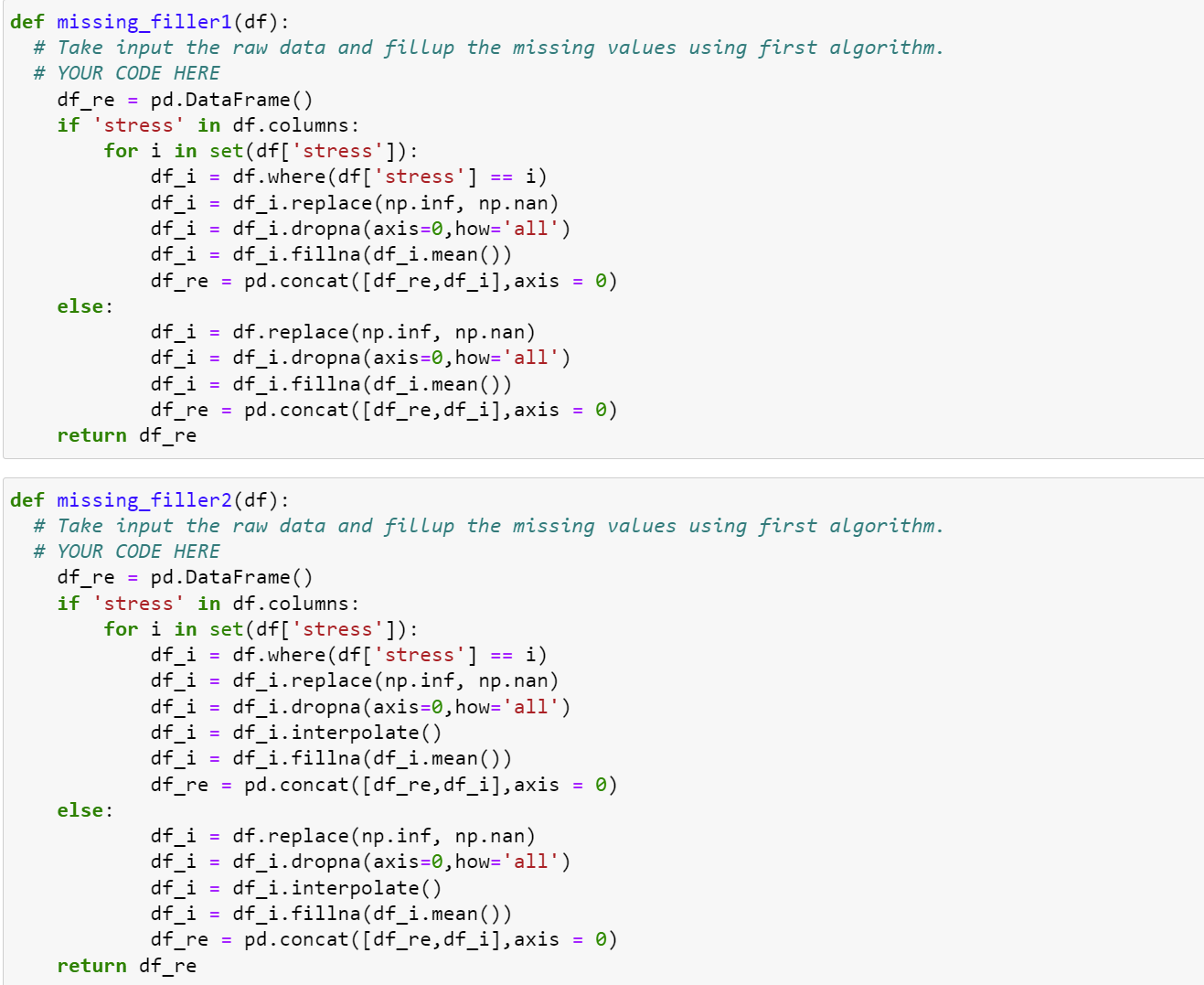


For method 1, I replaced all Inf with NaN, and fill NaN to be the mean value of the data in each label group.

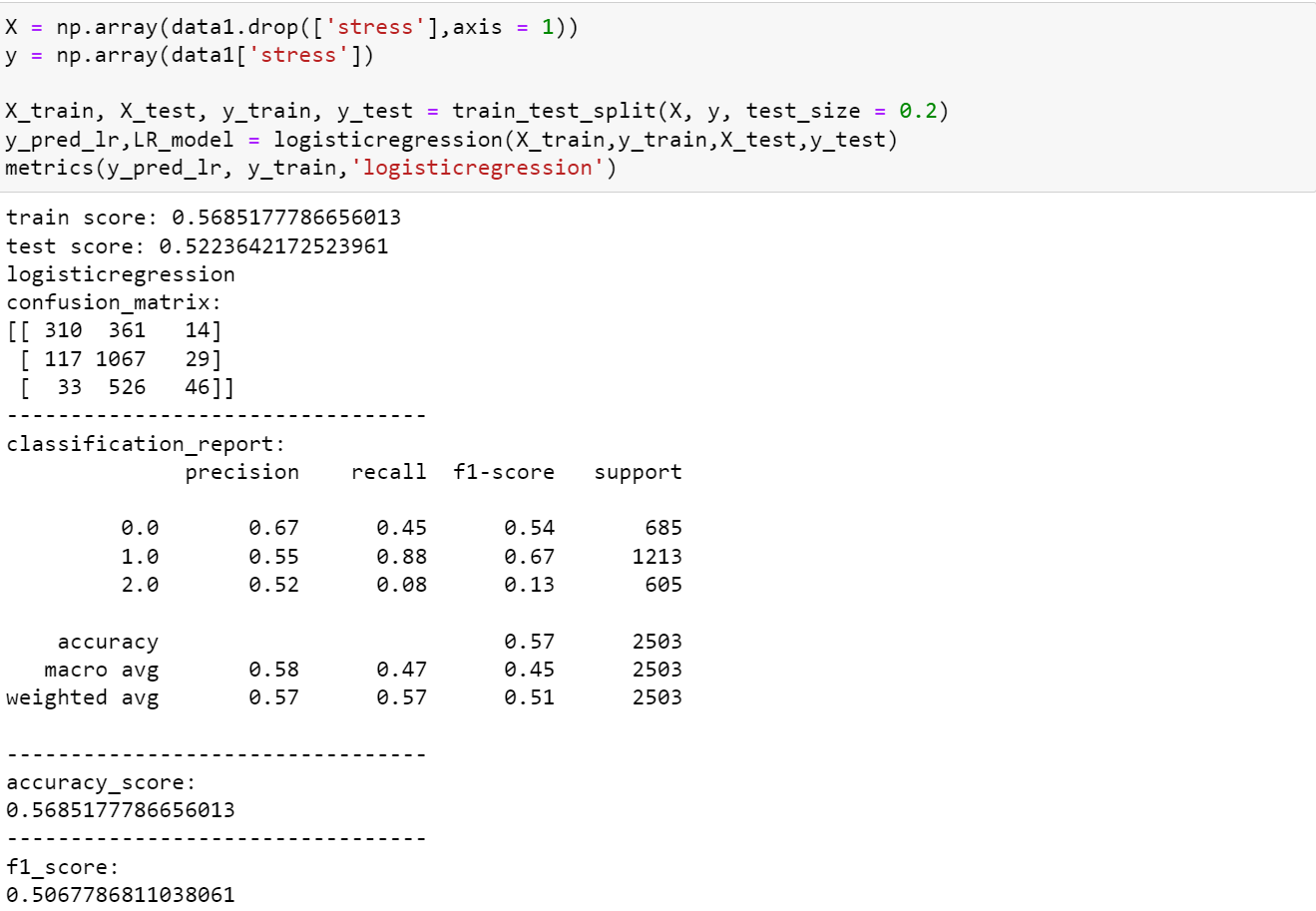


The f1 score is 0.49 and the accuracy is 0.55.

**2.3 Missing Filler 2**



In method 2, similarly, I replaced Inf with NaN, while I filled it with interpolation and fill the rest of the values with mean.



The f1 score is 0.50and the accuracy is 0.56. Obviously, interpolation is better than mean.

**2.4 Discussion**

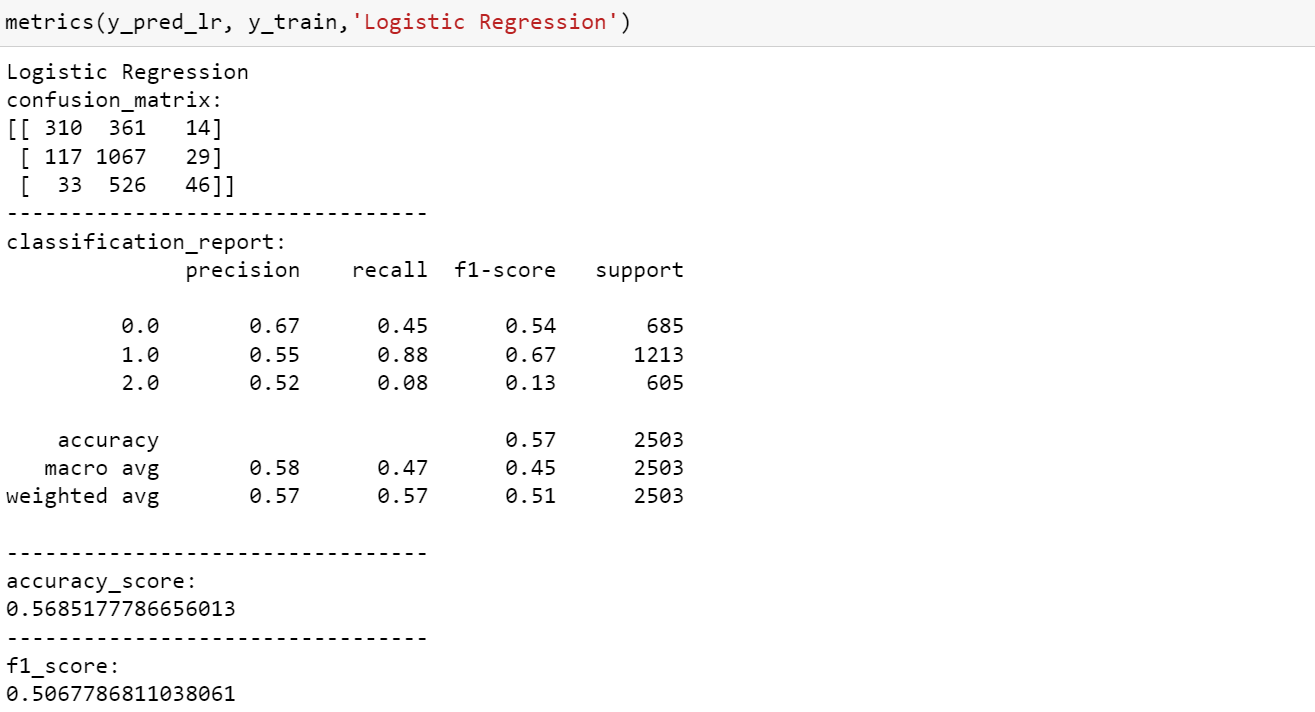
Method 2 has a better performance. Linear interpolation is an approximate calculation method that uses the proportional relationship to find other values of the unknown function according to a set of known values of the independent variables of the unknown function and its corresponding function values. method. It finds the unknown point according to the linear formula of the known point, and the accuracy is higher.

**Answer 3) (25 points)**

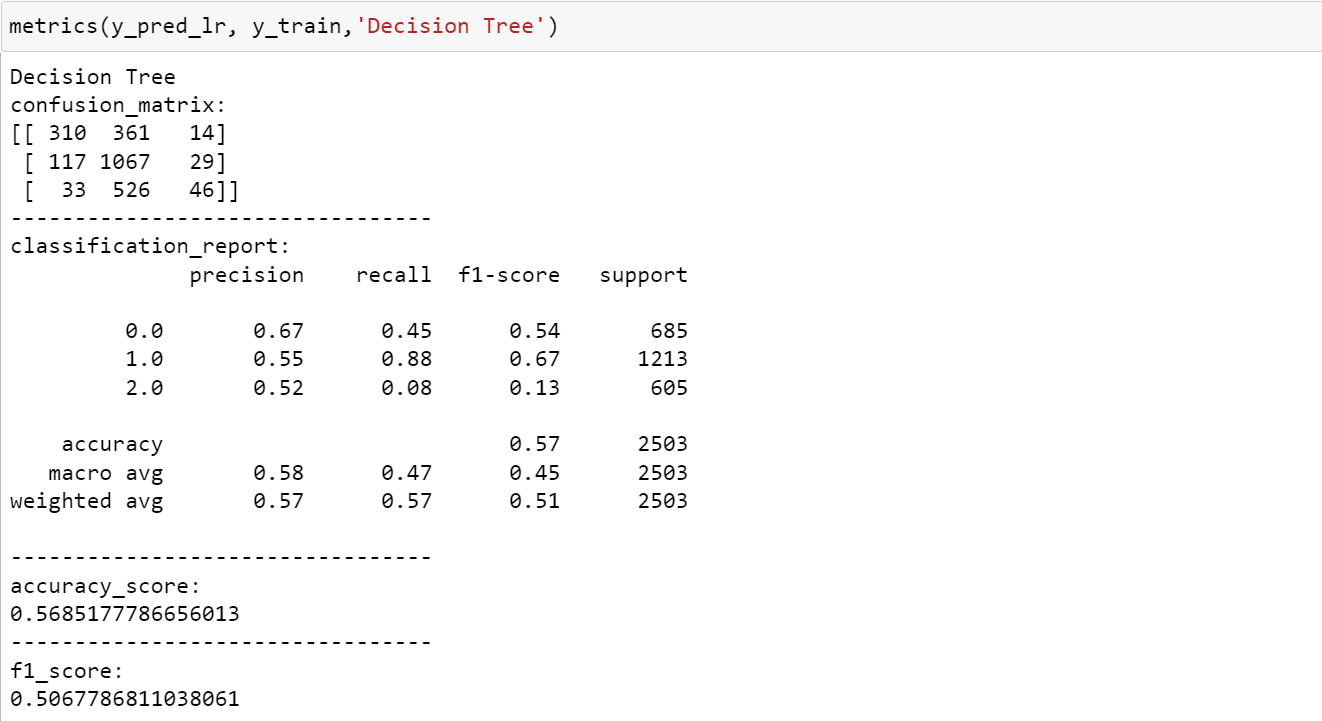
# Share results (F1score, precision, recall, and accuracy score) from at least 2 models in your

report.

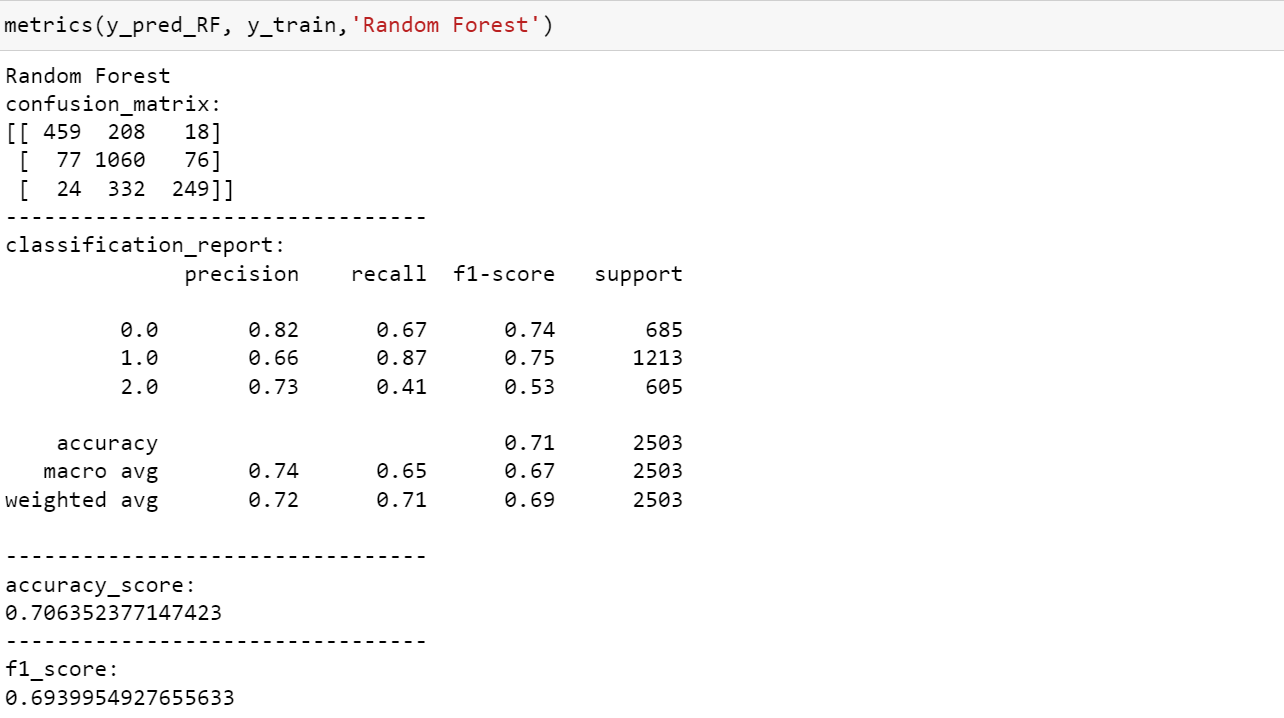
**3.1 Logistic Regression**



**3.2 Decision Tree**

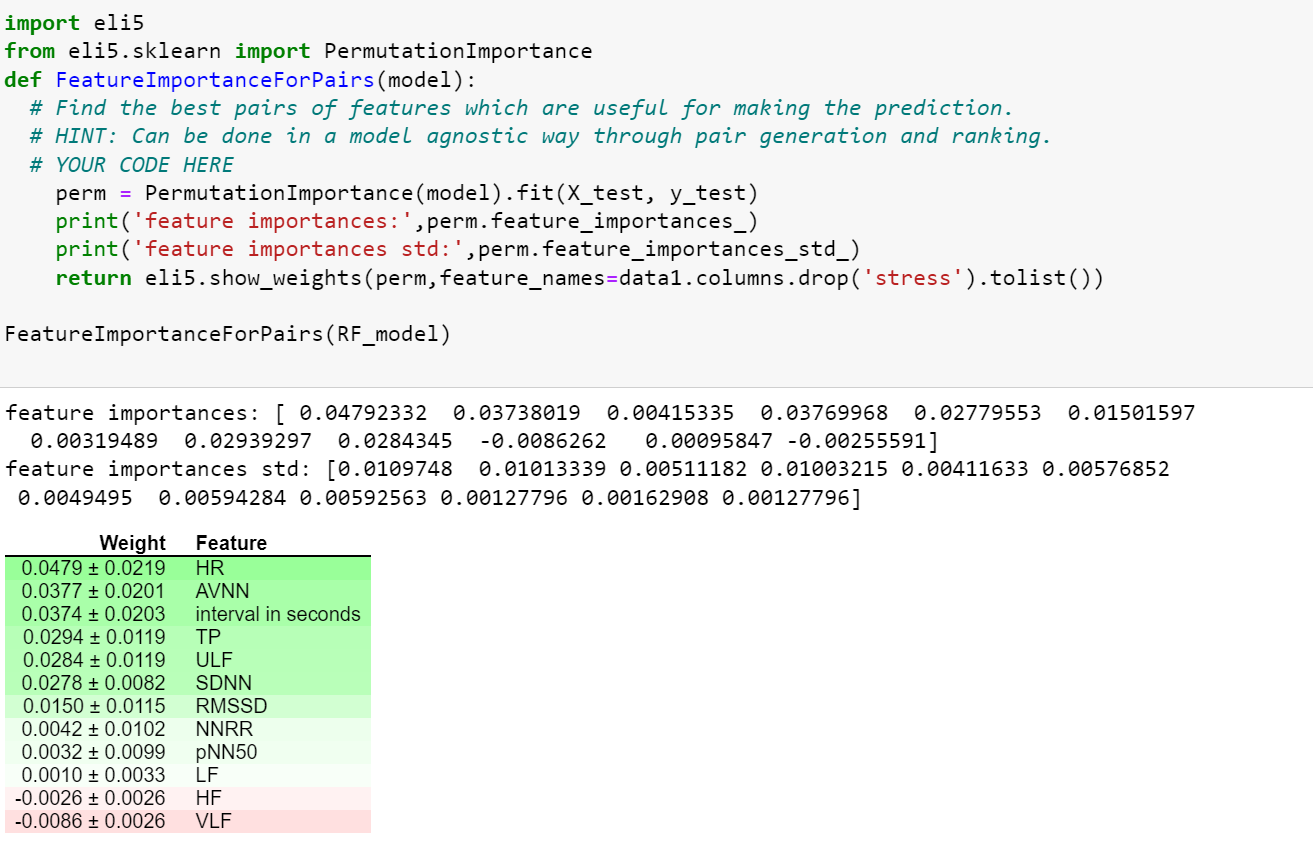


**3.3 Random Forest**



**Answer 4) (20 points)**

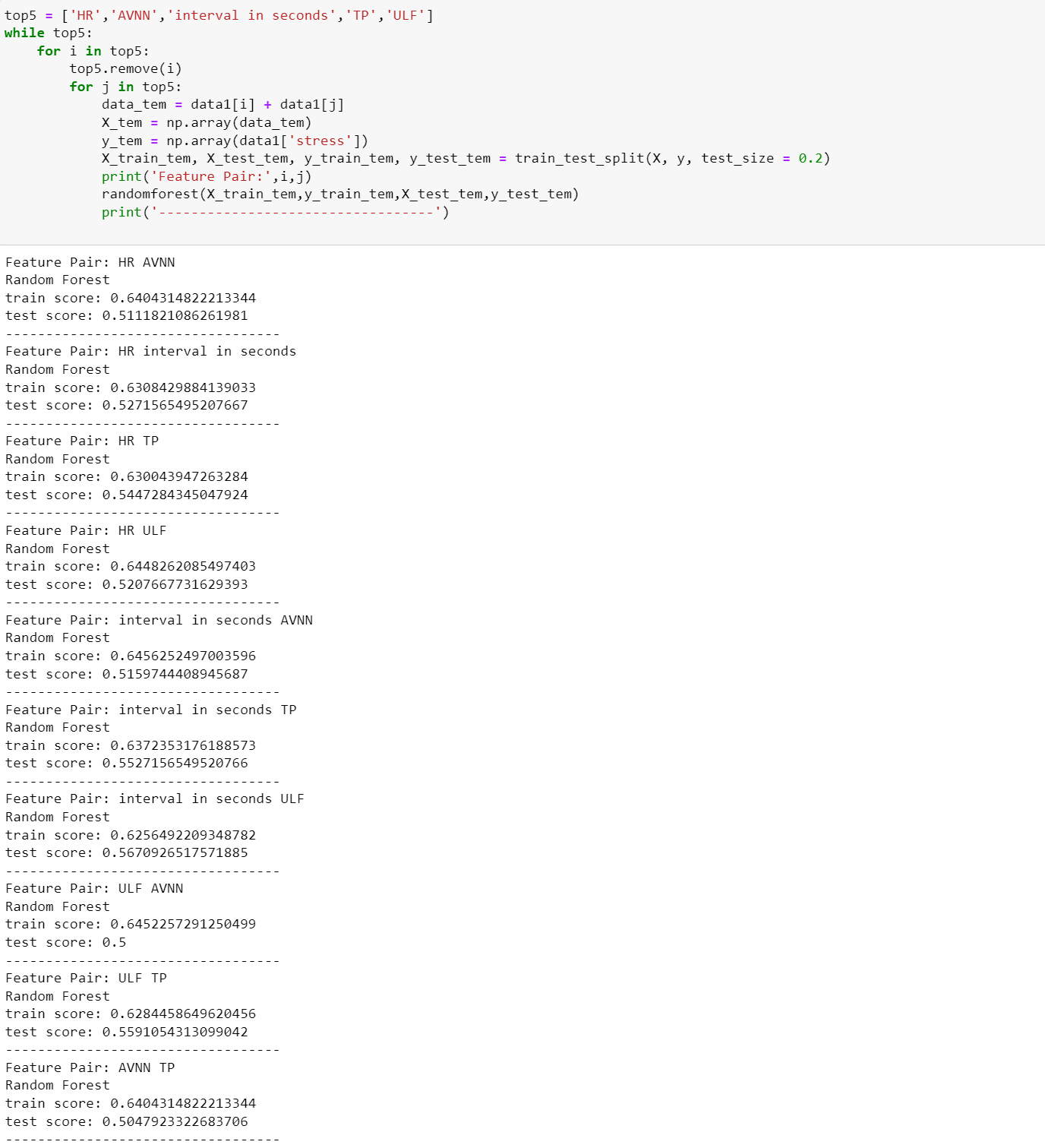
**4.1 Top 5 Features**



I processed features by PermutationImportance in eli5. In this function, the features will be instead by random values, so that the features can be checked for the importance of the prediction. If nothing changes after altering its value, that feature is insignificant.

So, by this way, I chose the top five features that influence the result of the random forest: HR, AVNN, the interval in seconds, TP, and ULF.

**4.2 Most relevant pair of features**



So, the most significant pair of features is ‘interval in seconds’ and ‘ULF’, which have an accuracy of 0.62 in the training set and an accuracy of 0.56 in the testing score.

In the measurement of mental stress, the greater the pressure, the more nervous the mind, the faster the heartbeat, and the more obvious the ultra-low frequency signal of the heartbeat. In this way, it is possible to predict whether an individual has greater mental stress by measuring these data, such as HR, AVNN, ULF and other data. It matches the result of the code.

**Answer 5) (10 points)**

A random forest is a classifier that is built in a random fashion and consists of multiple decision trees. The category of its output is determined by the mode of the category output by each tree.

Randomness is mainly reflected in two aspects:(1) When training each tree, select a dataset with the same size as N from all training samples (the number of samples is N) for training (that is, bootstrap sampling);

(2) At each node, a subset of all features is randomly selected to calculate the optimal segmentation method.

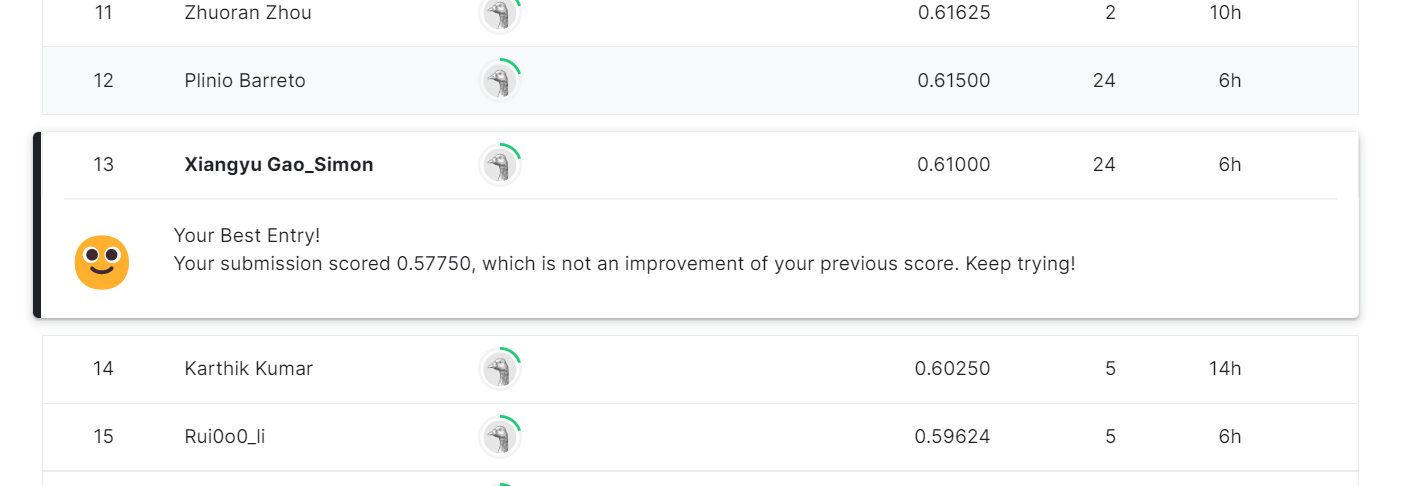
It has good performance and great advantages compared with other algorithms. Random forests can handle very high-dimensional data without feature selection. After training, the random forest can give which features are more important. The training speed is fast, and it is easy to make a parallel method. During the training process, the influence between features can be detected. For imbalanced datasets, random forests can balance the error. When there is a classification imbalance, random forests can provide an effective way to balance the errors in the dataset. The random forest has a strong anti-overfitting ability. So, it has the best performance in this classification task.

Logistic regression is widely used in regression problems and works well, but it also has some shortcomings, such as not being able to solve nonlinear problems, being sensitive to multicollinear data, and being difficult to deal with data imbalance problems, etc.

Decision trees work by finding interactions between variables. But each split of the tree reduces the dataset. Deliberately creating divisions will potentially introduce bias. Applying the greedy strategy to the decision tree variance, finding the correct starting point of the tree affects the final result. That is to say, small changes in the early stage will have a big impact later. Also, perturb a little, change a little value, and the decision tree changes. We want our classifier to be robust to noise, so this is not what we expect.

**Answer 6) (5 points)**

# Describe the model that gets the best Kaggle metrics accuracy for you.



My best f1 score is 0.61 by using the random forest with the 54 estimators, 47 max\_depth, and 0.005 min\_weight\_fraction\_leaf.