Attached file: hw1.ipynb

I open the data file data.txt as a data frame from pandas. According to the given information, this set has 4480 data rows, each of which has 535 values. According to the labels.txt file, the title of each column is specified. The last column is the data label, which has a value between 1 and 4 and is equal to the class number of that data.

1-neutral, 2-emotional, 3-mental and 4-physical

The first column is the number of people who have been tested, which are 40 people in total. This column is just an index and I don't use it in the training steps. So the data space that is to be classified has 532 dimensions (features) and the problem is a 4-class classification. The first step is to separate the test and training data. First, I shuffle the data and then divide it in the ratio of 33 to 66. This is done with train_test_split of sklearn library. All the training steps are done with x_train and y_train data and all the reports for each model are classified based on the x_test set with the built model and the results of this y_predict classification are compared with the actual y_test values and confusion matrix. and accuracy, precision, recall and f-measure are made. In order to have an understanding of the performance of the model on the training data, I did an evaluation on the training. These reports are produced with the metrics collection from the sklearn library.

Note: The data I am working with had a value in column 417 that was stored as a wrong data type, and this entire column is stored as a string even though it has a continuous numeric data type. I corrected this before proceeding.

Decission Tree

The first model built on the clf1 data is a decision tree with the information gain criterion, which builds the tree in the direction of the best branching to reduce entropy. This model is built with sklearn functions. It has a depth of 17 It is fully fitted on the training data and has no classification errors. Each leaf contains at least 1 data and each node has at least 2 data for branching, and tree pruning criteria are set to a minimum. I group the test data with clf1. The following results were obtained:

```
              precision    recall  f1-score   support

           1       0.96      0.94      0.95       372
           2       0.87      0.87      0.87       363
           3       0.86      0.85      0.85       376
           4       0.97      0.99      0.98       368

    accuracy                           0.91      1479
   macro avg       0.91      0.91      0.91      1479
weighted avg       0.91      0.91      0.91      1479

[[351   4  17   0]
 [  4 316  36   7]
 [ 11  43 319   3]
 [  1   0   1 366]]
```

According to the confusion matrix, the most error of the model is related to classifications of class 2 and 3 instead of each other. The main diameter of the correct classification matrix is the largest number on each row and column, and as a result, precision, recall, and f-measure measures are obtained with acceptable sizes for each class. The accuracy of the model is equal to 91, which is a good result.

Random Forest

The first model built on the clf2 data is a random forest in which there are 100 trees with information gain criteria as categories, and each one is built by bootstrap sampling on the data and on the features for the best branching to reduce entropy. have became. Tree construction is done using sklearn. By evaluating the test data label with clf2, the following results are obtained.

```
              precision    recall  f1-score   support

           1       1.00      0.99      0.99       372
           2       0.98      0.99      0.98       363
           3       0.98      0.98      0.98       376
           4       1.00      1.00      1.00       368

    accuracy                           0.99      1479
   macro avg       0.99      0.99      0.99      1479
weighted avg       0.99      0.99      0.99      1479

[[368   0   4   0]
 [  0 360   3   0]
 [  0   9 367   0]
 [  0   0   0 368]]
```

The accuracy obtained is the best. Especially from the comparison with clf1 was a single tree, social from 100 randomly constructed trees, the mistakes were reduced significantly. The clf2 model is fully fitted on the training data and without errors, and it is almost the same on the test data, and the classification is done with 99% accuracy.

XGBoost

The third model or clf3 is an XGBoost model built with the xgboost library. Default parameters are used for this model, the number of trees is 100, the learning rate is 0.300000012, and the maximum depth of each tree is 6. Each time, each tree is built iteratively from the previous tree so as to improve the previous tree. The results are as follows:

```
              precision    recall  f1-score   support

           1       1.00      0.99      1.00       372
           2       0.99      0.99      0.99       363
           3       0.98      0.99      0.99       376
           4       1.00      1.00      1.00       368

    accuracy                           0.99      1479
   macro avg       0.99      0.99      0.99      1479
weighted avg       0.99      0.99      0.99      1479

[[370   0   2   0]
 [  0 358   4   1]
 [  0   3 373   0]
 [  0   0   0 368]]
```

As expected, clf3 is better than any clf2. This model misclassified only 9 data on 1479 test data. This improved model is the same random forest as it has its positive features, including that it uses the consensus vote of several categories that are randomly built on samples of features and data, and it has this advantage. which includes only trees that complete each other in the constructed samples. This model is fully fitted on both the training and test sets.

SVM

The clf4 model is a support vector machine. With the regularization factor value of 10, radial basis function is used as kernel to transfer the data to the new space. The effort is to make the data linearly separable in the kernel space. The results of the model are as follows:

	precision	recall	f1-score	support
1	0.32	0.81	0.46	372
2	0.31	0.05	0.08	363
3	0.36	0.07	0.12	376
4	0.48	0.52	0.50	368
accuracy			0.36	1479
macro avg	0.37	0.36	0.29	1479
weighted avg	0.37	0.36	0.29	1479

```
[[301   8  11  52]
 [255  17  17  74]
 [250  14  27  85]
 [140  16  19 193]]
```

And on the training data we have:

	precision	recall	f1-score	support
1	0.32	0.82	0.46	748
2	0.34	0.05	0.09	757
3	0.42	0.09	0.14	744
4	0.50	0.57	0.53	752
accuracy			0.38	3001
macro avg	0.40	0.38	0.31	3001
weighted avg	0.40	0.38	0.31	3001

The model error is very high on both test and training data and is not comparable with previous models in terms of performance. The data of the problem is far from the initial assumption of SVM and it is clear that the complexity of a linear classifier is not suitable for separating these data. The classification accuracy of clf4 is equal to 36%. If we refer to the confusion matrix, the model tends to place most of the data in category 1, which causes the recall and f-measure values on classes 2 and 3 to be very low. This model is not reliable at all for finding data with labels 2 and 3. This feature causes low precision on class 1, which lowers the accuracy of voting for data labeled 1.

5 Fold Cross Validation Results

Among the 4 built models, the best results were related to the XGBoodt model, which was built under the name of clf3. For the final report of the performance of this model on these data, I use the k-fold cross validation method with cv=5. The following results are the average performance of 5 models built on x_train, y_train data:

```
fit_time
11.308441638946533

score_time
0.030693578720092773

test_accuracy
0.9830027731558513

test_recall_macro
0.9829835115004638

test_precision_macro
0.9832222462418407

test_f1_macro
0.9830225731153739

test_recall_weighted
0.9830027731558513

test_precision_weighted
0.9832524804465852

test_f1_weighted
0.9830472746431346
```

Also, if prediction is made on these 5 models, the result of confusion_matrix will be as follows.

```
[[732   3  13   0]
 [  0 743  14   0]
 [  4  15 724   1]
 [  0   0   1 751]]
```

For the returned accuracy value, I use the 5-fold cross validation method of the model that has the best result as the final model, and the parameters of this final model (clf model):

```
{'objective': 'multi:softprob',
 'use_label_encoder': True,
 'base_score': 0.5,
 'booster': 'gbtree',
 'colsample_bylevel': 1,
 'colsample_bynode': 1,
 'colsample_bytree': 1,
 'enable_categorical': False,
 'gamma': 0,
 'gpu_id': -1,
 'importance_type': None,
 'interaction_constraints': '',
 'learning_rate': 0.300000012,
 'max_delta_step': 0,
 'max_depth': 6,
 'min_child_weight': 1,
 'missing': nan,
 'monotone_constraints': '()',
 'n_estimators': 100,
 'n_jobs': 8,
 'num_parallel_tree': 1,
 'predictor': 'auto',
 'random_state': 0,
 'reg_alpha': 0,
 'reg_lambda': 1,
 'scale_pos_weight': None,
 'subsample': 1,
 'tree_method': 'exact',
 'validate_parameters': 1,
 'verbosity': None}
```

Comparison of the obtained results with the results of the article

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6960825/

In this article, instead of building a model with all features, several models are built, each of which works by selecting a number of features equal to Nmax, and after building and testing the models, a model with a combination of (ECG+TEB+EDA) signals is created. With a maximum of m features from this category, it is introduced as the best model with a probability of 22.2% error and selected as a category. In Table 6, the confusion matrix of this model is reported, according to which the values of precision, recall and accuracy are calculated for this model. The results of this model are written in the last row of the table. The number of features that I made the models for this exercise was 533, and my best result was 98.4 percent correct, and in this model, the accuracy criterion was 77 percent with only 40 features.

Method	Accuracy	Precision	Recall	f-measure

Decision tree	0.91	0.91	0.91	0.91
Random Forest	0.99	0.99	0.99	0.99
XGBoost	0.99 *Cv :* 0.983	0.99 *Cv :* 0.983	0.99 *Cv:* 0.983	0.99 *Cv :* 0.983
SVM	0.36	0.37	0.36	0.29
	0.7705	0.77085	0.77052	