28462000 - Bourez Nicolas

LINFO2262 - Machine Learning competition

1 Preprocessing

At first, I analyse the data. I've looked at the distribution of the features and it does not required a log transformation. I've also changed the features `event`, `node` and `grade` to categorical data. Finally, I've changed the label encoding of the target according to :

label	class number
ER+/HER2-	0
ER-/HER2-	1
HER2+	2

Table 1: Caption

2 Feature selection

For feature selection, I've performed two methods :

1. Select most important feature according to correlation with the target variable, standardize the resulting dataset, apply PCA to reduce the number of features furthermore. The number of kept principal components becomes a parameter for the model selection.

2. Select most important feature according to an XGBoost classifier training on the initial dataset. To do this, I've selected the feature with a score different than zero. Then, standardize the resulting dataset.

3 Model Selection

I've tried the three different models from the sklearn library which seem to perform the best in the literature:

- Logistic Regression
- Linear SVM classifier
- RandomForest
- XGboost classifier

I've decided not to use deep learning methods because the size of the dataset seems to be a little bit too small. But I might have use since I've heard some people reached good people with it.

Here is approximately the different result I get with the different models :

model	mean 10-CV accuracy	mean 10-CV bcr accuracy
LogisticRegression	0.874	0.866
SVC	0.871	0.855
RandomForestClassifier	0.81	0.77
XGBClassifier	0.875	0.855

Table 2: Caption

The best model according to both feature selection was the LogisticRegression with the given parameters:

```
model = LogisticRegression(C=0.007,
penalty='l2',class_weight='balanced',multi_class='multinomial',max_iter=2000)
```

My selection strategy was based on the balanced accuracy of the model performing 3 times a 10-CV with different splitting and take the mean of the result. I've also add penalization in my model to avoid overfitting.

4 Dealing with unbalanced classes

To give more weight to the less represented classes, i.e. 1 and 2, I have decided to train two binary classifiers :

- One with class 1 against the two others
- One with class 2 against the two others

These two classfiers achieved better result than my original one. I thus decided to combine the three classifiers in my final prediction with the following schema :

Algorithm 1: Probability selection

for i ***in*** *range(len(data))* **do**
  $y_{\text{new\_prob}}[i][0] \leftarrow$ probability of the first model for class 0;
  $y_{\text{new\_prob}}[i][1] \leftarrow$ (weighted) probability of the second model for class 1, if $>$ the one from the first model;
  $y_{\text{new\_prob}}[i][2] \leftarrow$ (weighted) probability of the third model for class 2, if $>$ the one from the first model;
end
$y_{\text{pred}} \leftarrow$ Select the max probability for each sample i;

Thanks to this selection, I was able to increase my accuracy by 1 on average.