Otto Product Classification

Mariem BOUHAHA Eden BELOUADAH

26 décembre 2017

1 Introduction

Product classification into meaningful categories helps marketers decide which strategy will help promote their business. This allows businesses, for example, design separate marketing campaigns for each category of product they offer.

In this project, we will be working on the Otto group product classification challenge where the objective is to build a predictive model which is able to distinguish between the main product categories of the Group.

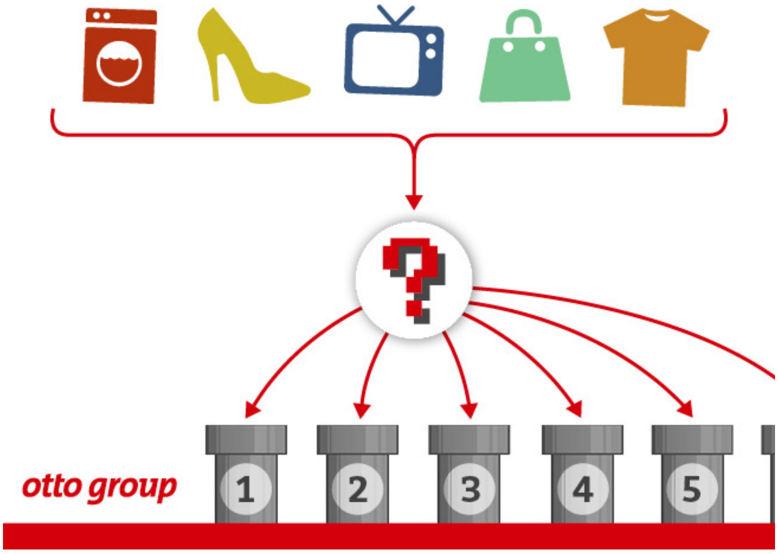
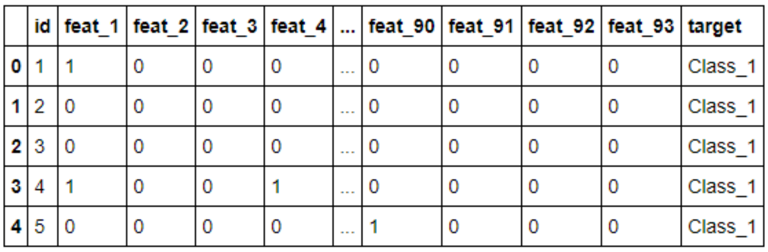


Figure 1. Otto product classification

2 Data Analysis

2.1 Presenting the data

Tha dataset provided by Otto has 61878 different products. Each sample is described by 93 categorical features. Values of the target variable can be one out of the 9 classes that the group identified.



	id	feat_1	feat_2	feat_3	feat_4	...	feat_90	feat_91	feat_92	feat_93	target
0	1	1	0	0	0	...	0	0	0	0	Class_1
1	2	0	0	0	0	...	0	0	0	0	Class_1
2	3	0	0	0	0	...	0	0	0	0	Class_1
3	4	1	0	0	1	...	0	0	0	0	Class_1
4	5	0	0	0	0	...	1	0	0	0	Class_1

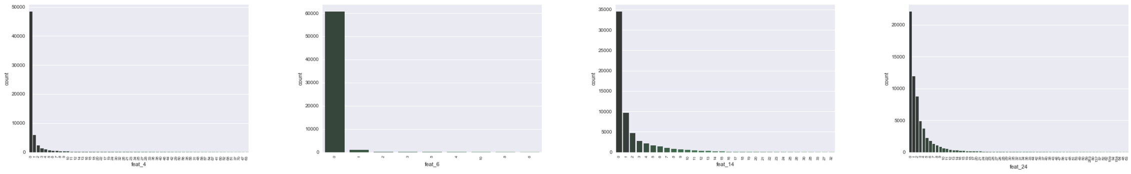
*Figure 2. Data presentation (61878 samples * 93 features * 9 classes)*

2.2 Exploratory Data Analysis

Before tackling the classification problem, a good understanding of our data is needed. Thus, we will started by providing some statistics and features caracteristics .

2.2.1 Features Distributions

We plotted the distribution of all 93 features to see how they differ across products. The first thing to notice in the figures below is the dominance of level 0 for all 93 fetures. Except for 2 or 3 features, the 75th percentile of the features distribution is 0. Since we have no idea about the sens of features in our data, we won't be able to figure out what a zero value means. Is might be either a missing value or a categorical level in itself.



(a) Feature 4 *(b) Feature 6* *(c) Feature 14* *(d) Feature 24*

Figure 3. Some features distributions

2.2.2 Feature Correlation

In figure 4, we show the correlation of the 93 features. As one may notice, a relatively high correlatio exis between a subset of features. For instance, top 10 highest correlation coefficients are shown in figure 5. Features like feat_39 and feat_45 are positively and highly correlated, with a coefficient of 0.82. Thus, we may think of a way to reduce dimensionality of the feature space, albeit no significant information loss occurs.

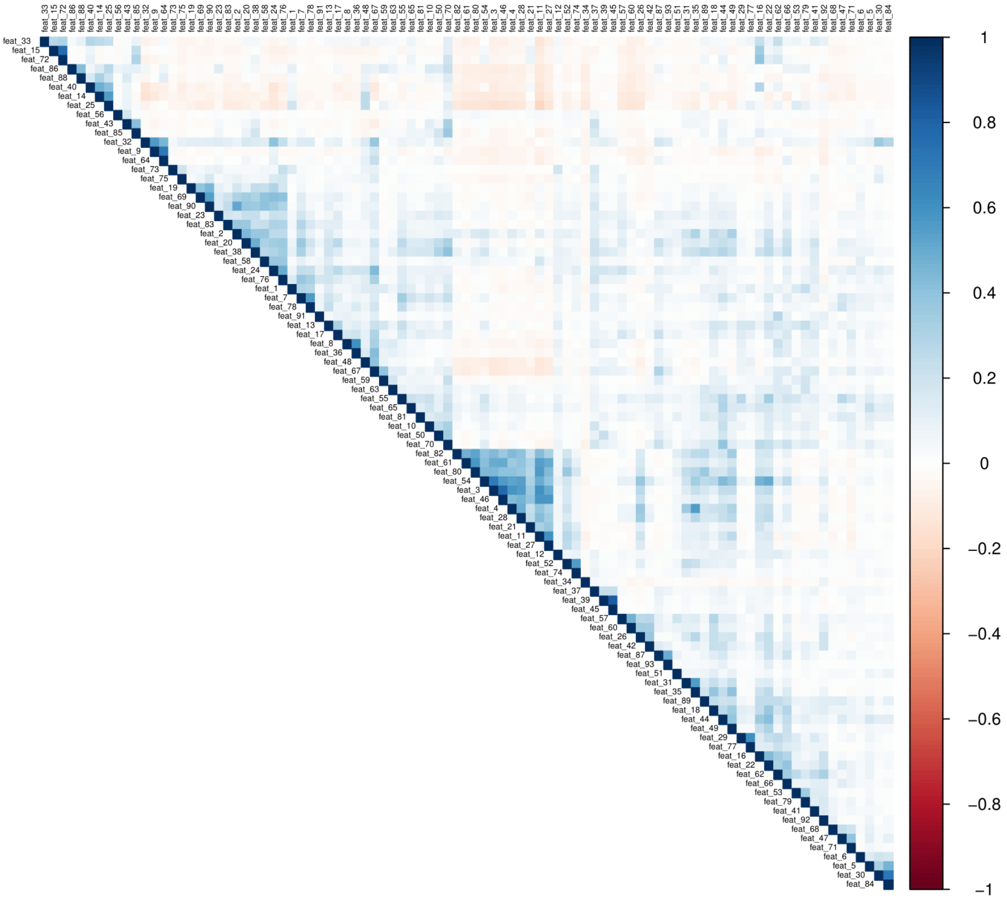
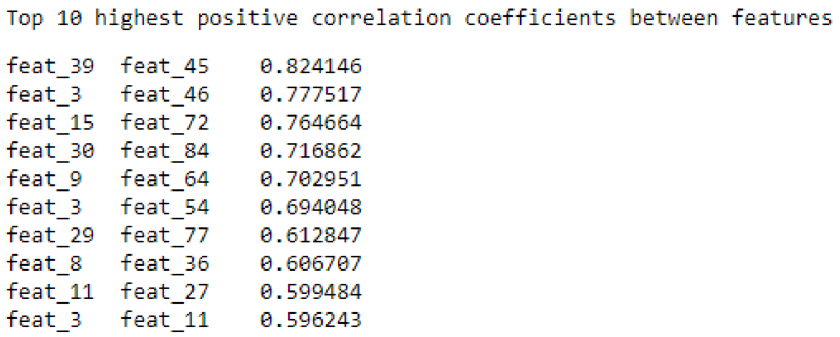


Figure 4. Correlation between features



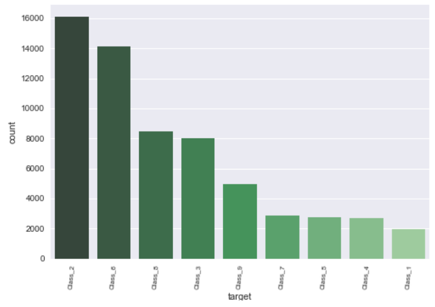
```
Top 10 highest positive correlation coefficients between features

feat_39  feat_45    0.824146
feat_3   feat_46    0.777517
feat_15  feat_72    0.764664
feat_30  feat_84    0.716862
feat_9   feat_64    0.702951
feat_3   feat_54    0.694048
feat_29  feat_77    0.612847
feat_8   feat_36    0.606707
feat_11  feat_27    0.599484
feat_3   feat_11    0.596243
```

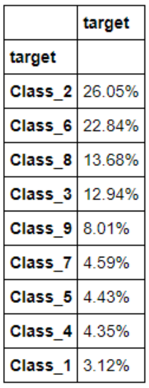
Figure 5. Top 10 highest correlation coefficients

2.2.3 Class Distribution

In figures 6a and 6b we show the distribution of the target variable and the proportion of each class in the data, respectively. We clearly see that classes are very imbalanced, indicating that we should be careful later on when we move to classification.



(a) Class distribution in Otto dataset



	target
target	
Class_2	26.05%
Class_6	22.84%
Class_8	13.68%
Class_3	12.94%
Class_9	8.01%
Class_7	4.59%
Class_5	4.43%
Class_4	4.35%
Class_1	3.12%

(b) Class proportions

Figure 6. Classes distribution

3 Predictive Modeling

In order to predict the class for each product, we tried several Scikit-Learn classification algorithms and compared their performances. Since it's a multiclass classification problem, we used a One-Vs-Rest classification schema using the following models :

- Linear Support Vector Machines
- Random Forests
- Logistic Regression
- MLP Classifier
- Gaussian Naive Bayes

Since we are facing a class-imbalance problem, it is not wise to use simple accuracy to evaluate the model. Instead, we used the ROC AUC as a performance evaluation metric for our classifiers. Quoting Scikit-Learn documentation : "ROC curves typically feature true positive rate on the Y axis, and false positive rate on the X axis. This means that the top left corner of the plot is the “ideal” point - a false positive rate of zero, and a true positive rate of one. This is not very realistic, but it does mean that a larger area under the curve (AUC) is usually better. Although ROC curves are typically used in binary classification to study the output of a classifier, we may extend ROC curve and ROC area to multi-class or multi-label classification, but it is necessary to binarize the output first. One ROC curve can be drawn per label, but we can also draw a ROC curve by considering each element of the label indicator matrix as a binary prediction (micro-averaging).

Another evaluation measure for multi-class classification is macro-averaging, which gives equal weight to the classification of each label.

The “steepness” of ROC curves is also important, since it is ideal to maximize the true positive rate while minimizing the false positive rate".

We also checked the performace of some models when they are applied to PCA-transformed data.

Finally, we used bagging of different classiers and evaluated its performance. Voting is one of the simplest ways of combining the predictions from multiple machine learning algorithms.It works by first creating two or more standalone models from the training dataset. A Voting Classifier can then be used to wrap the models and average the predictions of the sub-models when asked to make predictions for new data.

3.1 Results

In the figure below, we show the ROC curves resulting from the predictions of the models used.

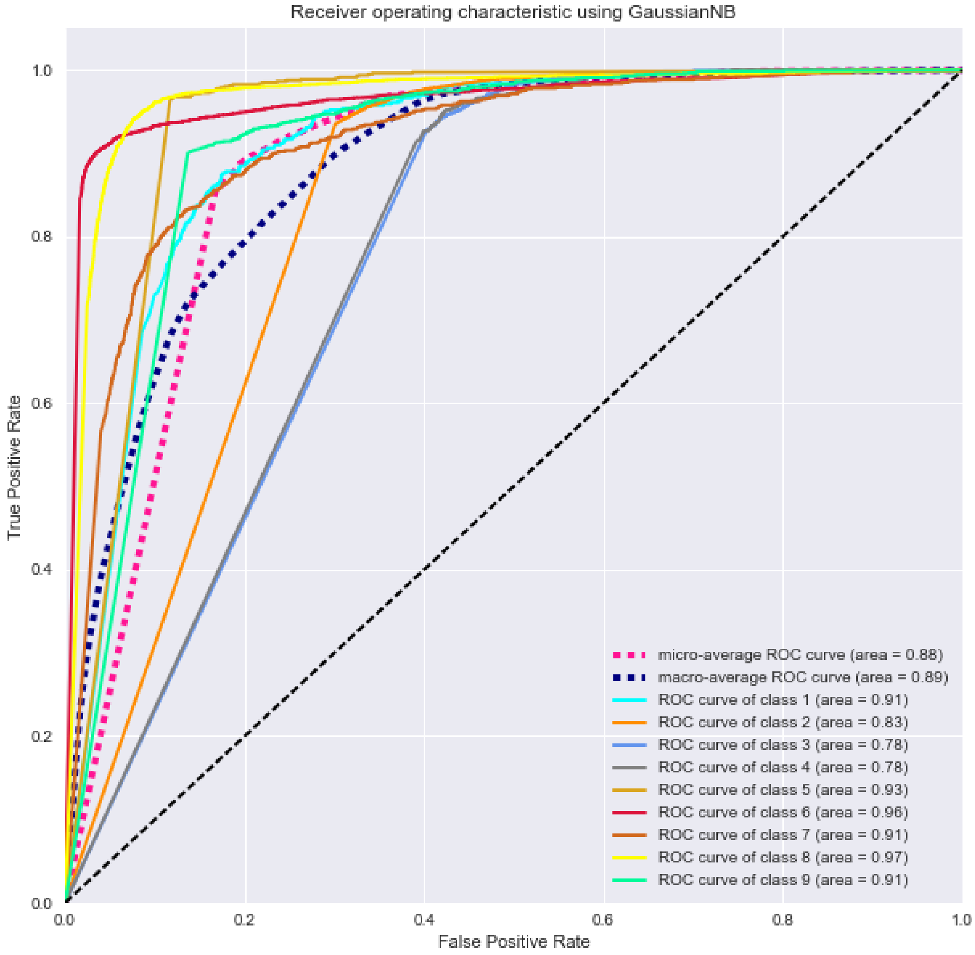


Figure 7. ROC for GaussianNB

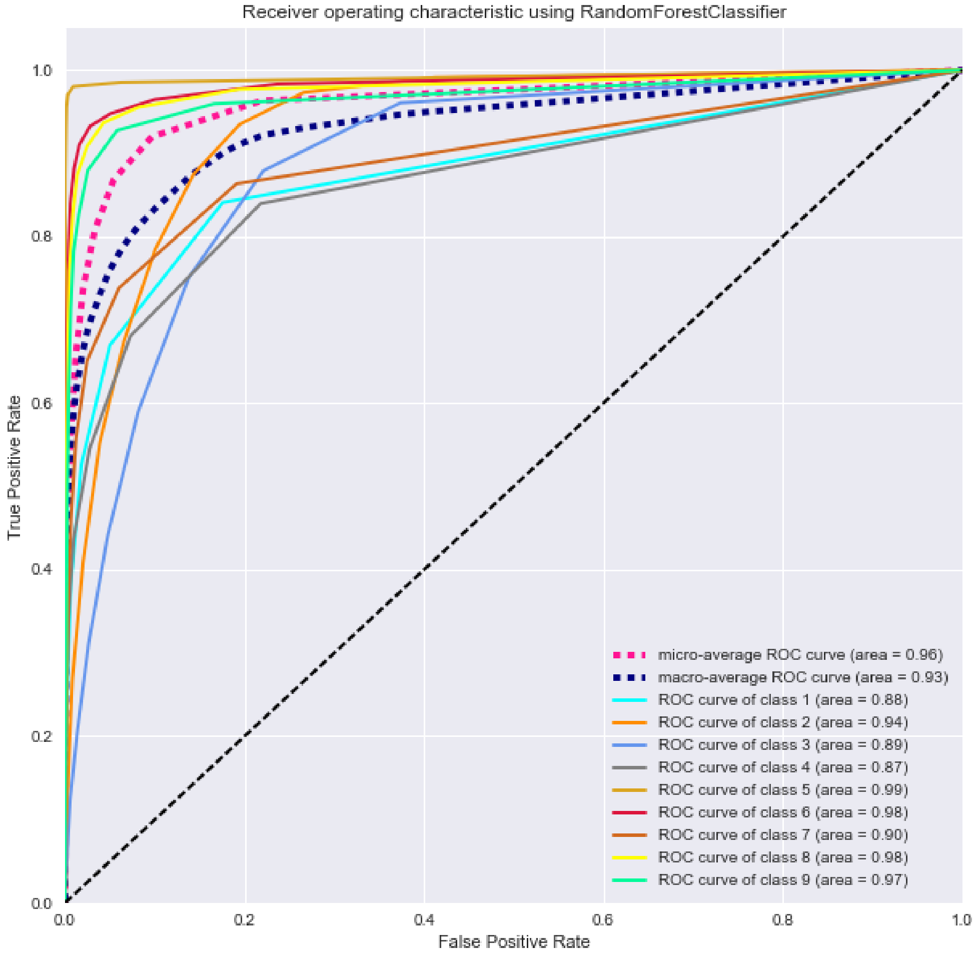


Figure 8. ROC for Random Forests

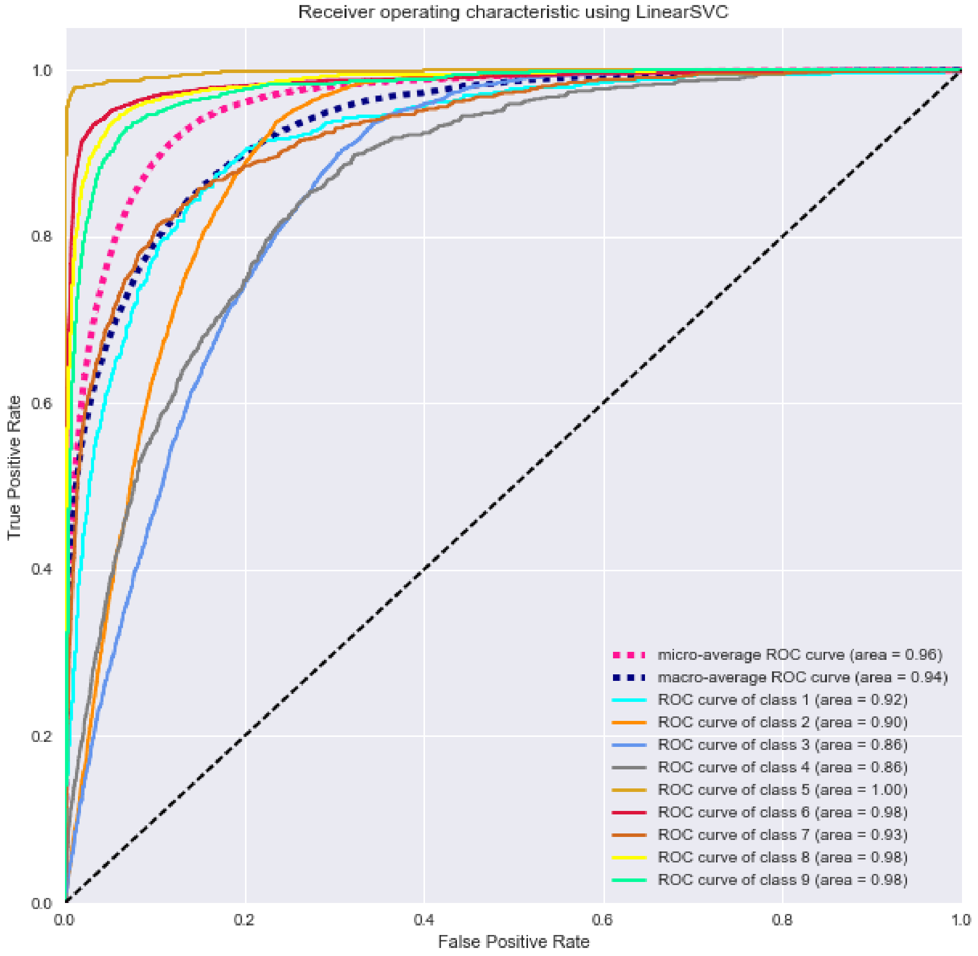


Figure 9. ROC for Linear SVC

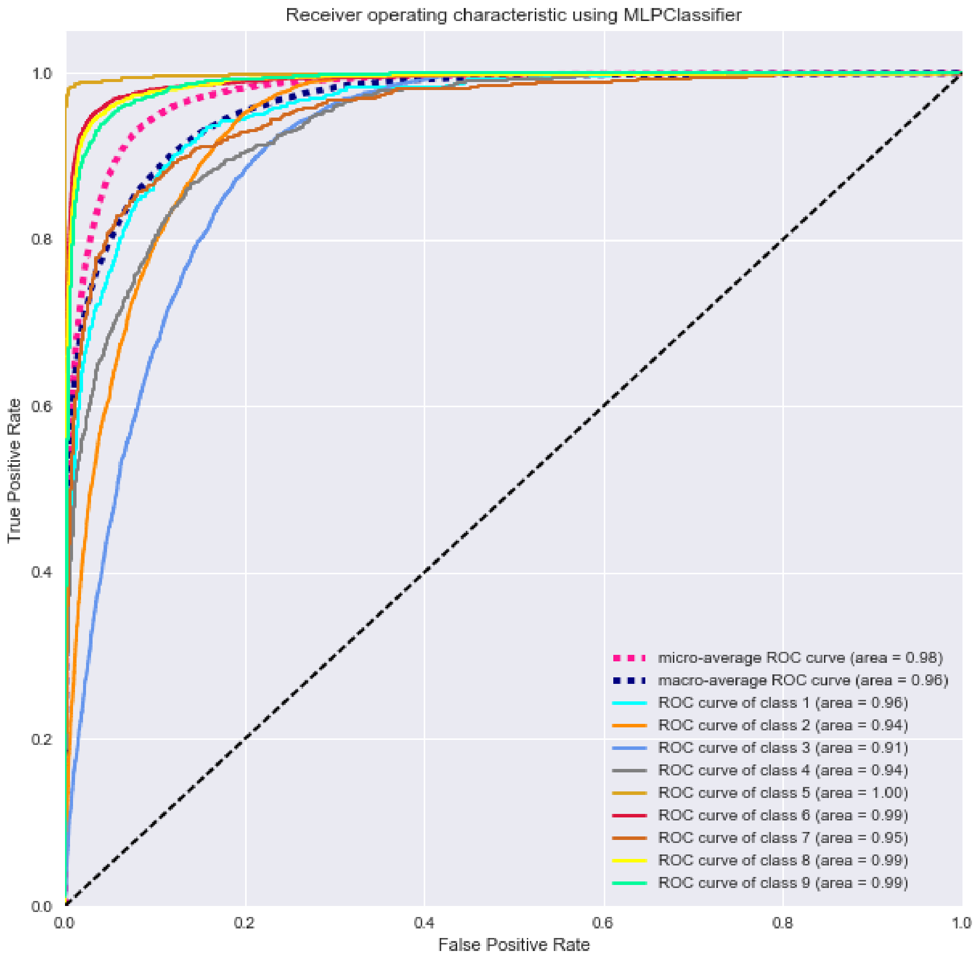


Figure 10. ROC for MLP

Clearly, the GaussianNB does not perform well on our data. However, Random Forests, SVM and MLP seem to have better AUCs. Also, their corresponding curves are steep, indication a good trade-off between True positives and False positives. We also notice that the curves differ from one class to another, reflecting the degree of difficulty to distinguish it from the other classes.

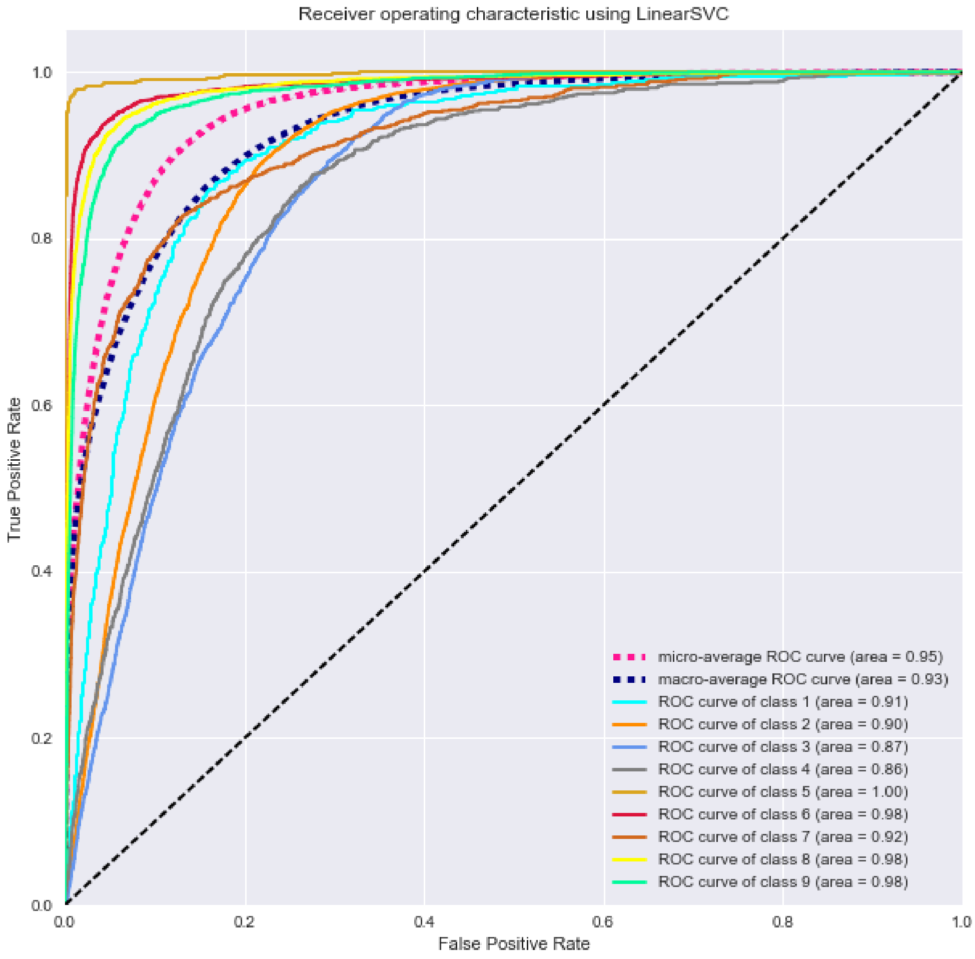


Figure 11. ROC for SVM using class weights for the PCA-transformed data

We can notice that using the PCA-transformed feature space, we don't loose too much information when we use LinearSVC with balanced class weights as a classifier. Thus, for the sake of computation time optimization, one would prefer to use the reduced data rather than the initial one.

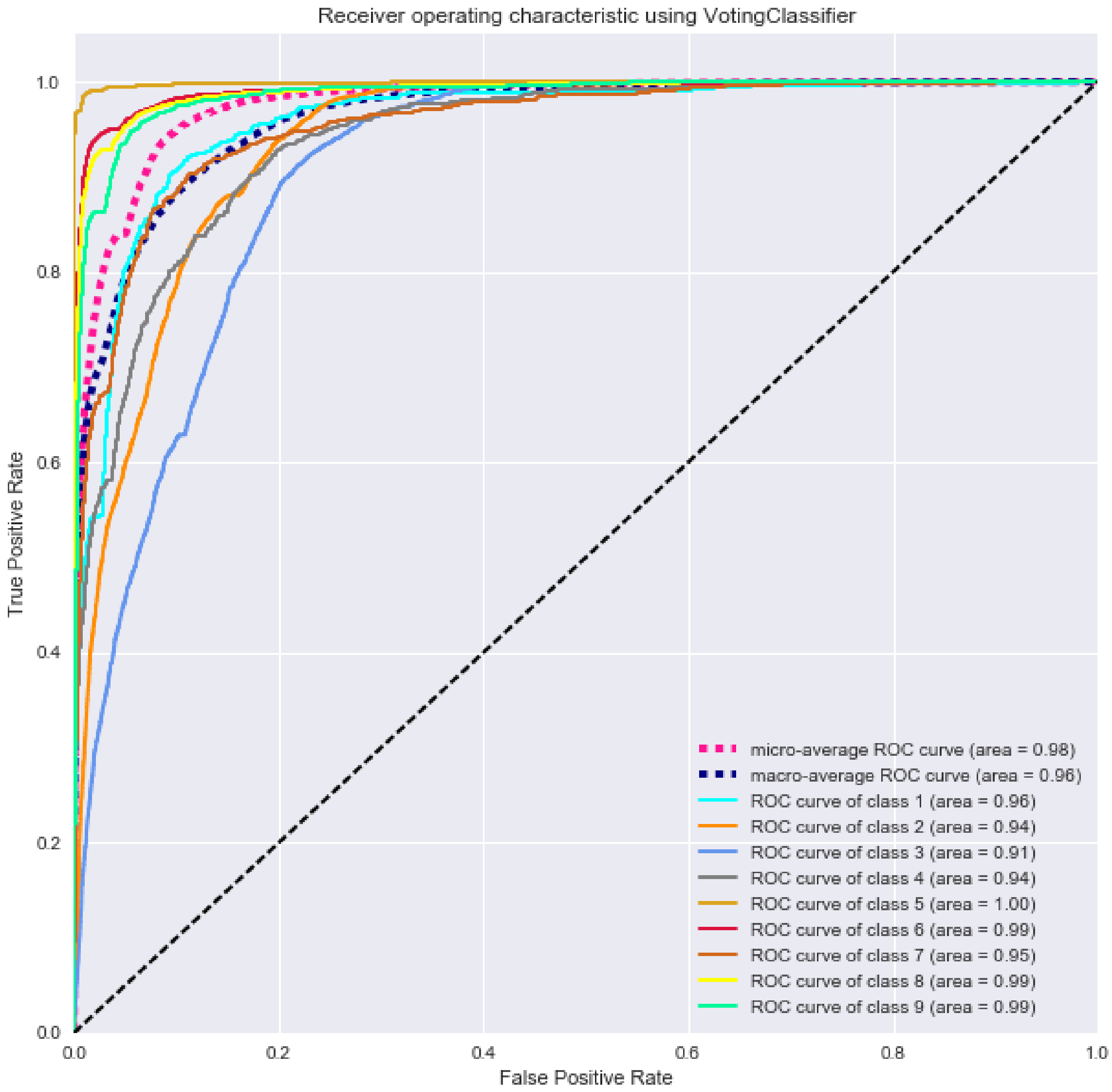


Figure 12. ROC for the Voting Classifier

As for the Voting Classifier, overall performance is now clearly enhanced, which is already expected.