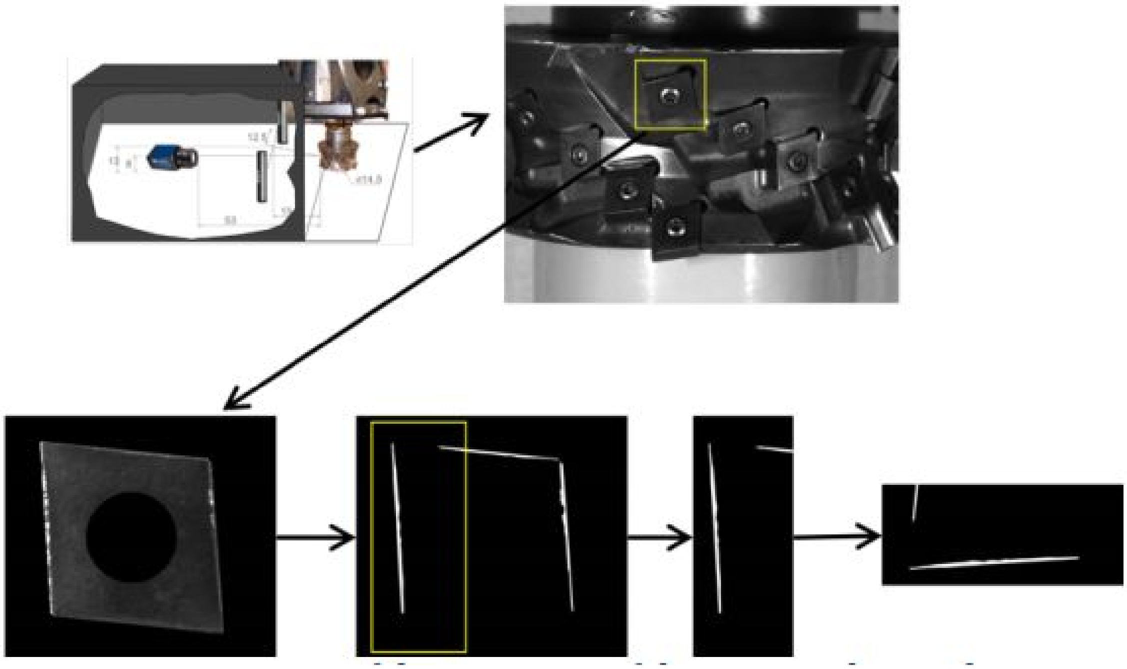


Machine learning to improve tool wear monitoring in milling processes

10.01.2020

EL HAKOUNI Adil

Ecole des Mines de St Etienne

TB3:Image and pattern recognition

Introduction

When it comes to heavy industries, changing the tools is a costly procedure in terms of time and money. In this project, we are going to perform a Machine Learning based solution to solve the tool changing problem.

Goal

building a classier that predicts the state of the cutting tool and help in decision-making

(it is time to change the tool (class:1) or not yet (class : 0)) before the tool is totally damaged.

Characteristics

This is a supervised machine learning problem (the data is already labeled) where the duty is to binary classify an input image (an image of the tool describing its cracks using some geometrical features).

The amount of data that we have is not big enough to use a big neural network (deep learning) to handle the task. A classical machine learning algorithm is, so far, enough to build an accurate classifier.

The raw data (images of the tool) is not directly implemented in the algorithm, but we need an important preprocessing phase; image preprocessing and segmentation, features selection and extraction, and then we apply the model to the features we already have extracted.

Outline

I. Data description

- a- Image data.
- b- Features extraction.

II. Classification' models and comparison

- a- Logistic regression
- b- Support Vector Machine
- c-Neural Networks

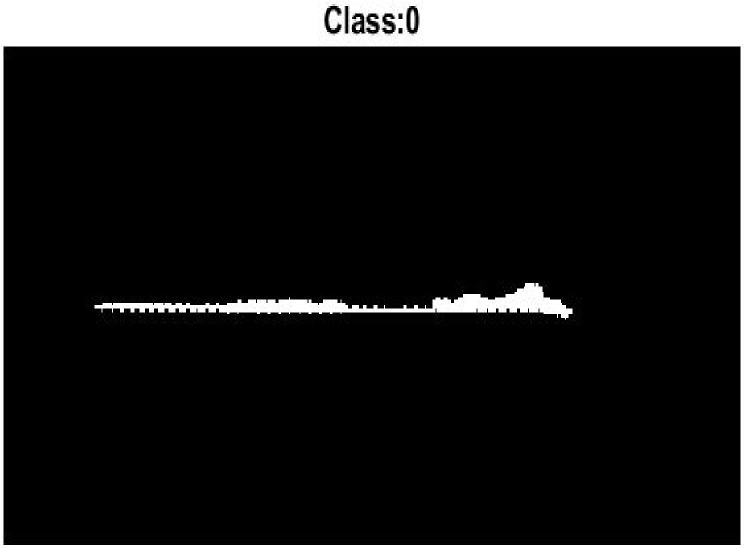
III. Principal components' analysis

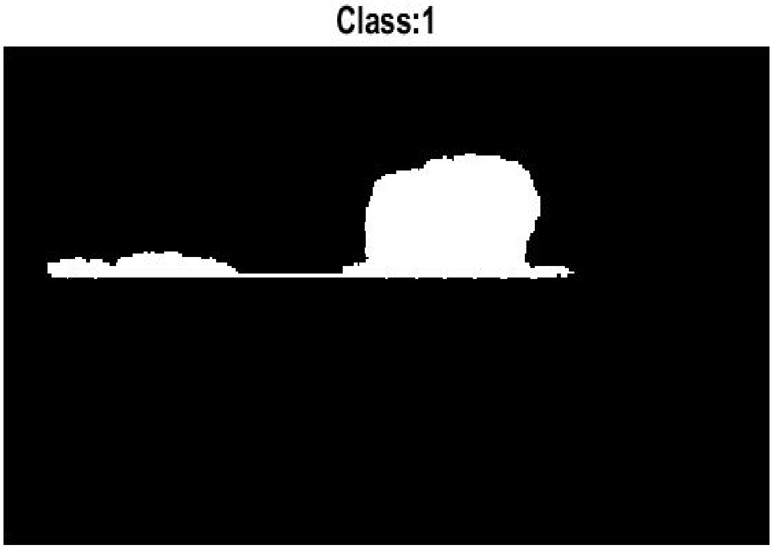
IV. Conclusion

I. Data description

a- Image data

we have a set of 202 binary images showing the state of the tool and already labeled by experts.





b- Descriptors: geometrical features.

to describe our images we use the following geometrical features

· Convex Area: Number of pixels of the smallest convex polygon that contains the region.

· Eccentricity: Scalar that specifies the eccentricity of the ellipse that has the same second central moments as the region.

· Perimeter: Number of points in the contour of the region.

· Equivalent Diameter: Scalar that specifies the diameter of a circle with the same area as the region.

· Extent: Scalar that specifies the ratio of the pixels of the region to the pixels in the bounding box around the region.

· Filled Area: Number of pixels belonging to the region after filling its possible holes.

· Minor Axis Length: Length of the minor axis of the ellipse that has the same second central moments as the region.

· Major Axis Length: Length of the major axis of the ellipse that has the same second central moments as the region.

· R: Ratio between the major and minor axis of the ellipse that has the same second central moments as the region.

· Solidity: Ratio between the area of the region and the Convex Area of the region.

II. Classification: models and comparison

During this project, we built the Logistic regression classier from scratch with no built-in functions.

In this section, we will focus more on the Logistic regression classifier, and in the end, we will see a comparison of three classifiers regarding the same problem.

a- Logistic regression

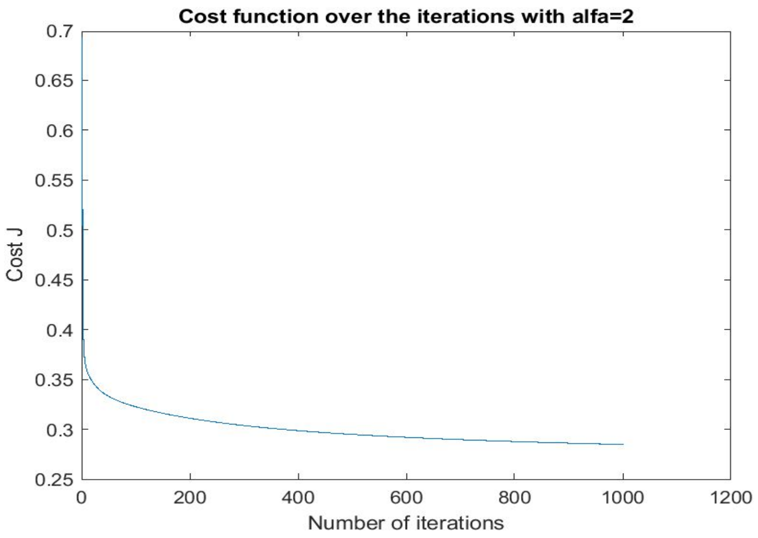
% Data splitting

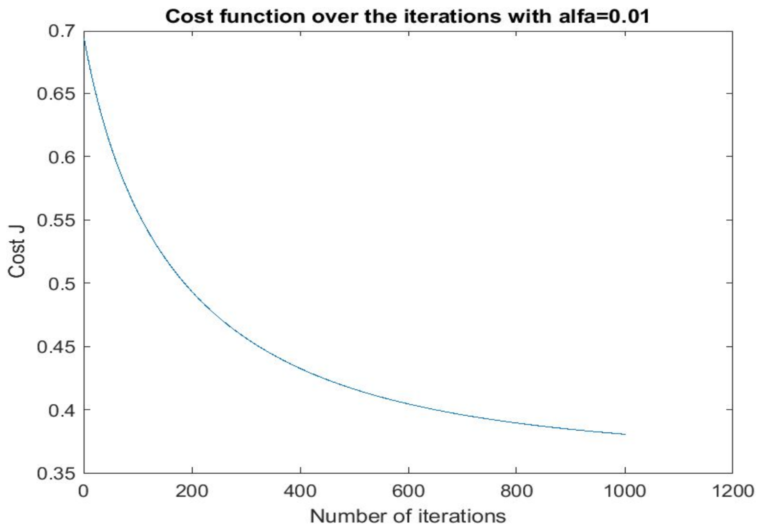
We used the 80-20 rule to split the data (80% for the training and 20% for the test), the two sets have been chosen randomly among the data.

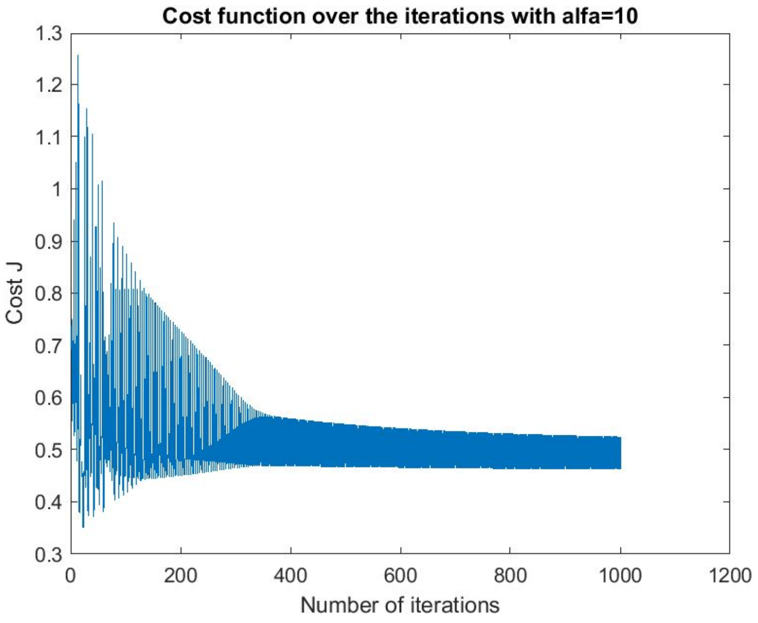
% Parameters tuning

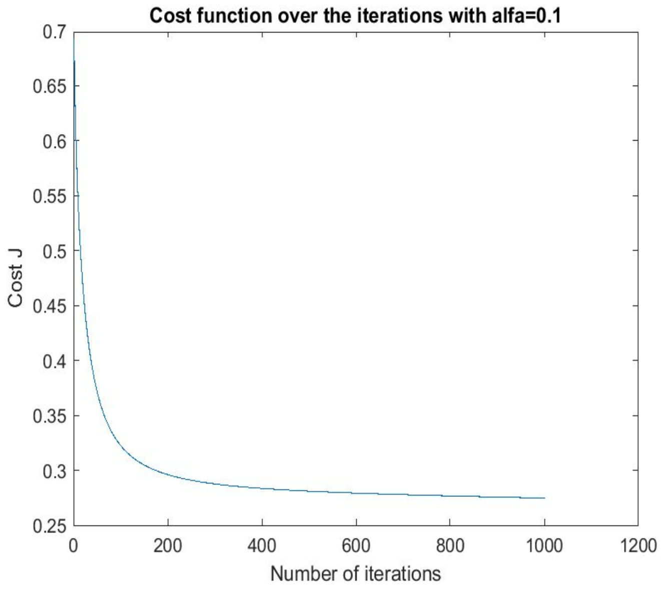
Like many machine learning algorithms, the logistic regression has its own hyper parameters that influence significantly the performance of the model, in this case, let's take a look at the learning rate alpha and the learning cost (the number of iterations iter_max). To assign the best values to those parameters we fix one and then we study the effect of the other one, where the objective is to have a well-fitting model at a minimum cost.

The impact of alpha on the cost function



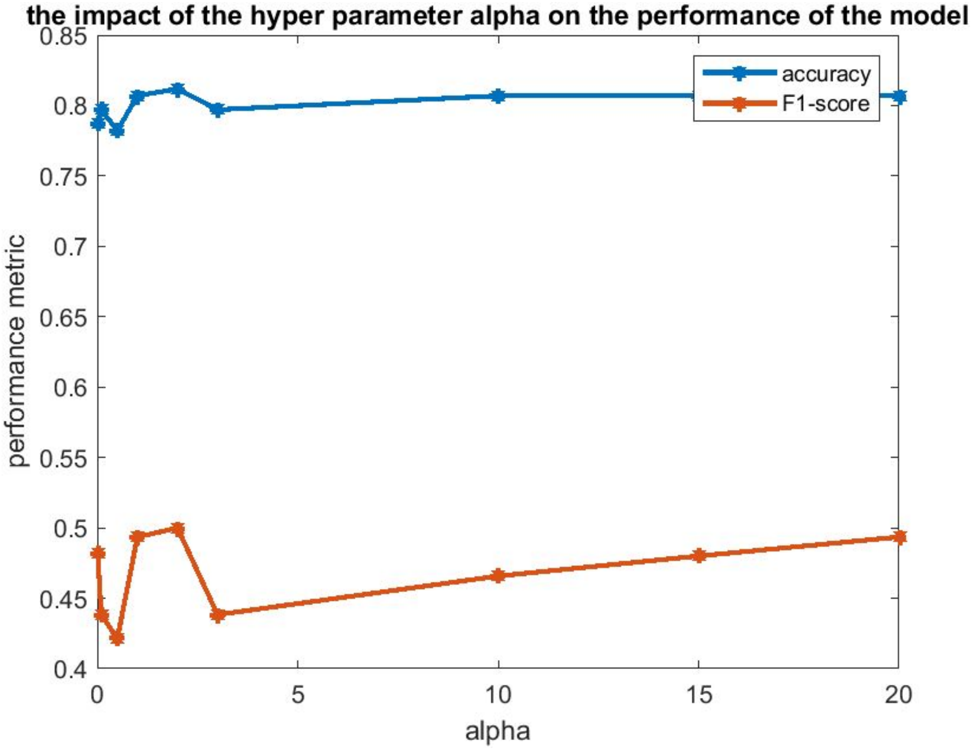






Conclusions

- When alpha is too small the cost function takes more iterations (time) to converge; for alpha =0.01 even after max_iter =1000 the cost function still showing a decrease.
- When alpha is too big the cost function is not showing a smooth decrease, we see the presence of fluctuations.

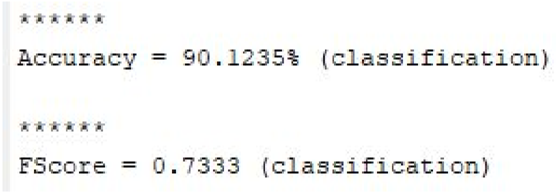


From the graph above we can say that alpha=2 is the best learning rate for max_iter=1000, that why the model’s hyper parameters are set to alpha=2 and max_iter=1000.(In the graph above we used the LOO-cross validation for the performance metric)

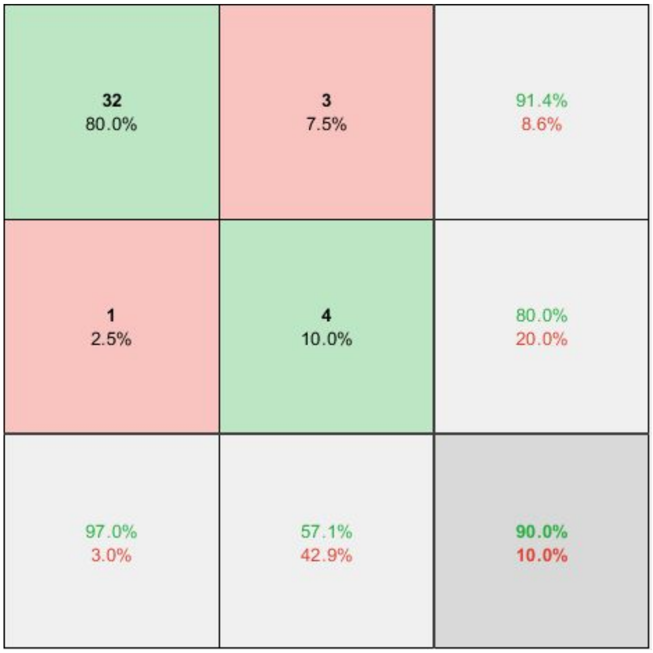
F1-score vs accuracy

It is time now to see how good our model fitted the data, to measure the performance of our model the accuracy is a widely used measure, but this problem has its own specificities and the F1 score is more fair as quality measure for two reasons; first, the data is not balanced (42 of positive answers out of 202 observations (only 21% of the data) !! If we use a model with 0 as a fixed output its accuracy may reach 70%!!). Second, the decision is not symmetric (not the same benefit/cost for a positive answer as a negative one if the answer is 1 then we need to change the tool immediately if not keep the machine working).

Those are the performance measure and the ROC curve for a given train/test data

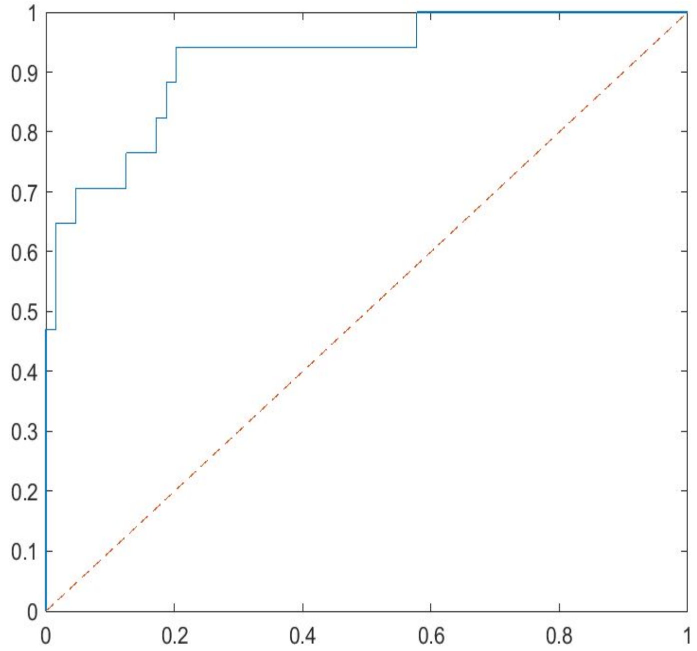


```
******
Accuracy = 90.1235% (classification)

******
FScore = 0.7333 (classification)
```



32 80.0%	3 7.5%	91.4% 8.6%
1 2.5%	4 10.0%	80.0% 20.0%
97.0% 3.0%	57.1% 42.9%	90.0% 10.0%

Confusion matrix



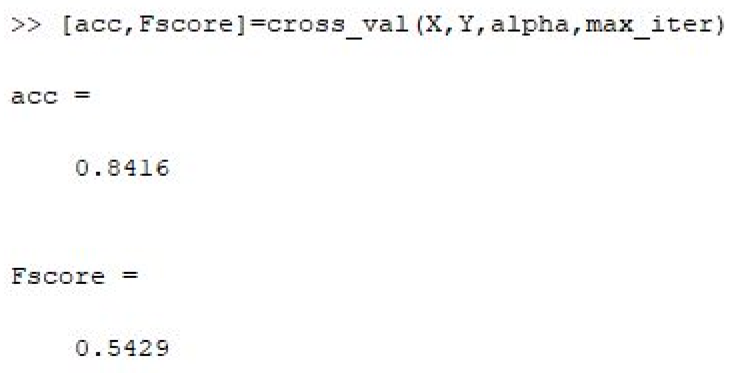
ROC curve

The cross-validation

Since the train data and test data are not the same in each time we run the code (because we randomly select the elements of each set in each time we run the code) the accuracy/F-score above is linked to a given data split rather than the accuracy/F-score of the model. That why cross-validation is convenient to assign a performance measure(accuracy/F-score) to our model. The matlab file cross_val contains the function that does the leave one out cross-validation (LOOCV)

! This function was built for this model and it may not work otherwise!

The result after cross-validation:



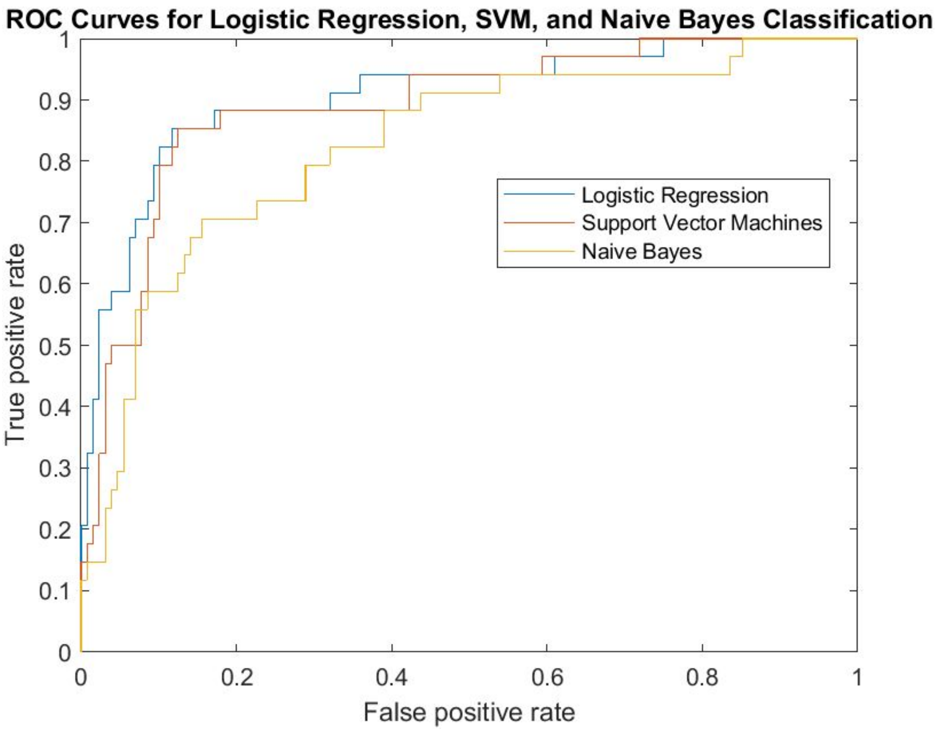
```
>> [acc,Fscore]=cross_val(X,Y,alpha,max_iter)

acc =

    0.8416


Fscore =

    0.5429
```

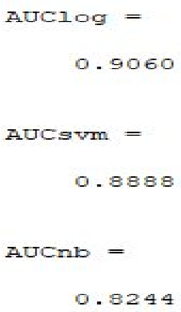
 The accuracy of the model logistic regression after the cross-validation is about 84 % where the F1-score is about 54%.

B-models comparison

We compared the SVM (support vector machine), the NB (Naive Baise) and the logistic regression classifiers, for the two classifiers (SVM and NB) we used the built-in matlab’s functions fitcsvm() and fitcnb().

One of the best ways to compare models is the ROC curve; as a reminder, big the area under the curve (AUC) is, the best model is.





```
AUClog =

    0.9060

AUCsvm =

    0.8888

AUCnb =

    0.8244
```

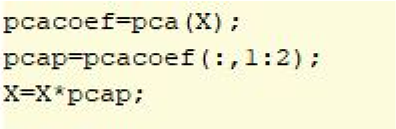
Areas under the curve values

conclusion

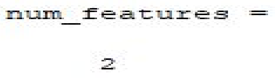
It is clear from the graph and the AUC values that the logistic regression is the best classier followed by svm and finally the NB .

III- Dimensionality reduction using Principal Components Analysis

The principal component analysis is a technique for *feature extraction* — so it combines our input variables in a specific way, then we can drop the “least important” variables while still retaining the most valuable parts of all the variables, in this case, we have 10 variables, and we will only save two variables after we put our data in the new space.



```
pcacoef=pca(X);
pcap=pcacoef(:,1:2);
X=X*pcap;
```



```
num_features =

     2
```

Even if the number of variables is only 2 (two) the accuracy is about 80% and the FScore is “good” as well which is interesting because we have a good classifier with only 2 variables instead of 10 (we got rid of a lot of calculations)

This technique is more interesting when we have a lot of data. It reduces considerably the cost in terms of memory and time without loosing as much in terms of performance .

Conclusion

This project was an end to end machine learning project starting from the raw data the data preprocessing then building the logistic regression classifier from scratch, the training and finally validation of the model, when we decide which is the best model we retrain it using all the labeled data that we have (train set and test set) to tackle the real world images that had never been examined.