# Numerical project 2

Machine Learning Course
by Morten Hjorth-Jensen at GANIL

# Machine learning on the pulsar data base

Paul GUIBOURG, Iftikhar SAFI, Tristan LE CORNU
Master 2 NAC Physique - UFR Sciences CAEN

February 21, 2020

## Abstract

*This document presents the code and the analysis on the subject of the project 2, on the pulsar data base. Using a variety of methods, we seeks to determine the best way to find the parameter wich describe data. We applied classification algorithms and neural networks using Python librairies such as Scikit-Learn or TensorFlow. We have encouraging results for our neural networks, we will have to continue and do a more detailed analysis afterwards.*

## Introduction

The aims of this work are to be able to evaluate the best method to analys data according to the set of data. We choose the pulsar data base available on the web site **Kaggle**.
In the first part, we will use several logistic regressions, to etablish a model which describe the data set. In the second part, we draw the results from the neural network we will use.
We analysed the result with a critical evaluation of the pros and the cons. This document will be finish wenn we conclude about the best method to analys the data according to the perform model.

## Classification problem

### A - Exhibition part. : the data

The data data that we use are extract from **Kaggle** web-site. That contains eight continous variables and the return value class i.e. if the sample is a pulsar, that give back 1, else zero. The eight values are :

Mean of the integrated profile.

Standard deviation of the integrated profile.

Excess kurtosis of the integrated profile.

Skewness of the integrated profile.

Mean of the DM-SNR curve.

Standard deviation of the DM-SNR curve.

Excess kurtosis of the DM-SNR curve.

Skewness of the DM-SNR curve.

The last column is the class (zero or one). The author precize that the sample contains 17,898 example and approximately ten pourcent are a pulsar. Data base comme from the site [1].

The file that we have, are a '.csv' format. So, we need to use read it and keep it somewhere.

### B - Piece of code

We can read the file with the csv librairy in python. We put it into a pikkle format trought a pandas DataFrame. The data will be more easily accessible.

```
import csv
import pickle
import pandas as pd
```

[1] https://archive.ics.uci.edu/ml/datasets/HTRU2

```
FILE_REP = './pulsar_dataBase/'
FILE_NAME = 'pulsar_stars.csv'

with open(FILE_REP+FILE_NAME , newline='')
                   as csvfile :
    CSVreader = csv.reader(csvfile, delimiter=',')
    data_base = pd.DataFrame(CSVreader)
pickle.dump( data_base, open( "save_data_pulsar.p",
                             "wb" ) )
```

Python 3.7

### C - Analysis

We failed to implement a logistic regression on our data indeed our accuracy is 0.05 which is really bad.

## Neural network

### A - Code

This is the code for the neural network :

```
def DNN(lmbd, eta):
dnn = Sequential([
Dense(n_hidden_neurons, input_shape=(n_features,),
activation='sigmoid', kernel_regularizer=l2(lmbd)),
Dense(n_categories, activation='sigmoid'),
])
adam = Adam(learning_rate=eta)
dnn.compile(loss='mean_squared_error',
optimizer=adam, metrics=['acc'])

history = dnn.fit(X_train_scaled, Y_train, epochs=epochs, batch_size=32,  validation_split=0.25
EarlyStopping(monitor='val_loss', min_delta=1e-5, patience=10,
verbose=0, mode='auto', baseline=None, restore_best_weights=True)
])

# Plot training & validation accuracy values
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

score = dnn.evaluate(X_test, Y_test, batch_size=32)
print(score)
```

Python 3.7

We use a classification method as in project 1.

## B - Analysis

Here are the results we got for the neural network :

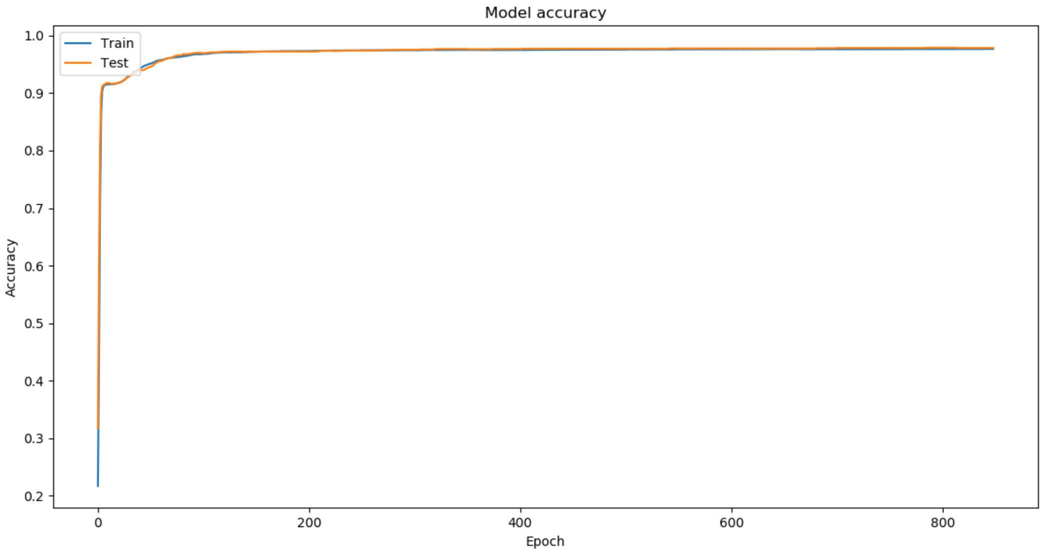

Figure 1: Accuracy of train data and test data in function of the epoch with the model train by neural network

We can see that we need a lot of epoch before reaching overtraining (around 820). We have an accuracy of 0.9786 with a mse of 0.0187 which is good.

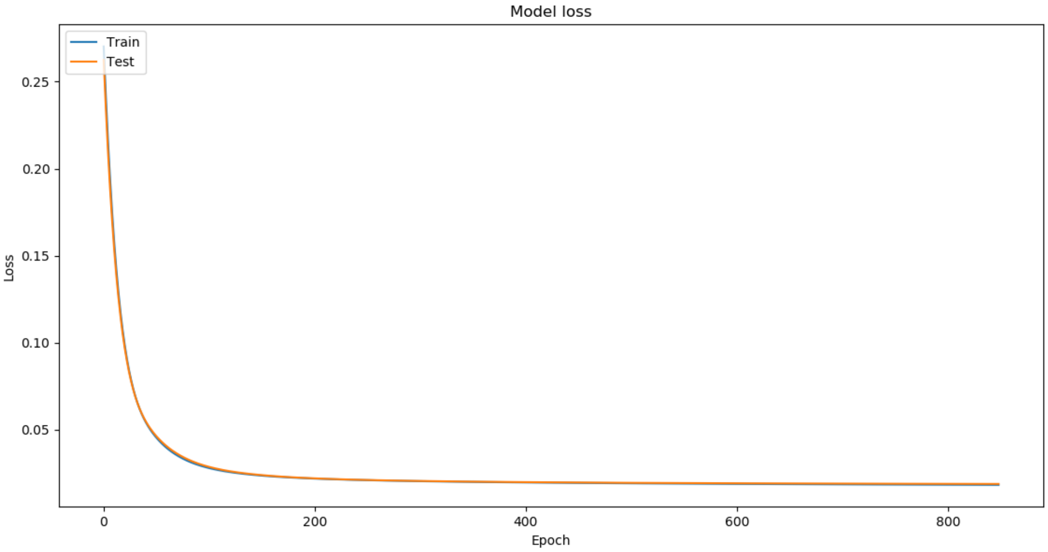

Figure 2: MSE of train data and test data in function of the epoch with the model train by neural network

## Conclusion

We have encouraging results for the neural network but we can surely do better in our analysis such as changing the weight factors, looking at the parameters that seem to be correlated, etc... Unfortunately with the start of the internship and the distance, we did not have much time to work together on this project.