CMPEDA Project: Lending Club loan data prediction

A Finance Predictive Model based on DNN and XGboost

Manuel Luci, Andrea Dell'Abate

1 Introducion

LendingClub is a US peer-to-peer lending company which specialises in lending various types of loans to urban customers.
When the company receives a loan application, we have to make a decision based on the applicant's profile.
Two types of risks are associated with the bank's decision :

- if the applicant is likely to repay the loan, then not approving it results in a loss of business to the company ;
- if the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving it may lead to a financial loss for the company.

Our aim is to develop the best supervised machine learning model capable to identify which borrowers will payoff their loans.
We'll try to do this binary classification task using Deep Neural Network and X-G boost models.
Link dataset :`https://www.kaggle.com/code/faressayah/lending-club-loan-defaulters-predictio`
`data?select=lending_club_loan_two.csv`

2 Preprocessing (module : data_analysis)

We divided the preprocessing part into two phases :

- In the first one we developed the statistical part of the dataset analyzing how all the features were distributed and how they were correlated with each other.
- In the second part we modified the dataset for our purposes : the missing values were treated (eliminating or replacing them as appropriate), we eliminated the variables that were highly correlated with each other and we chose those that were most useful for our machine learning task.
  Finally we replaced all categorical variables with numeric ones and we performed the one hot encoder by means of the dummy module (dummy data).

Below we report some usefull graphs for our analysis :

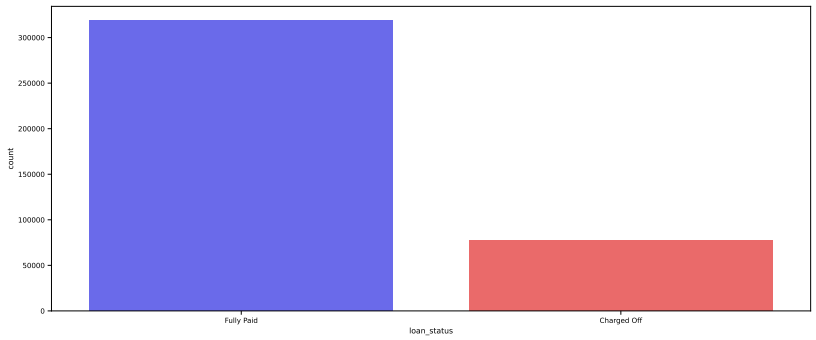

FIGURE 1 – loan_status distribution, we can see that our database is strictly unbalanced

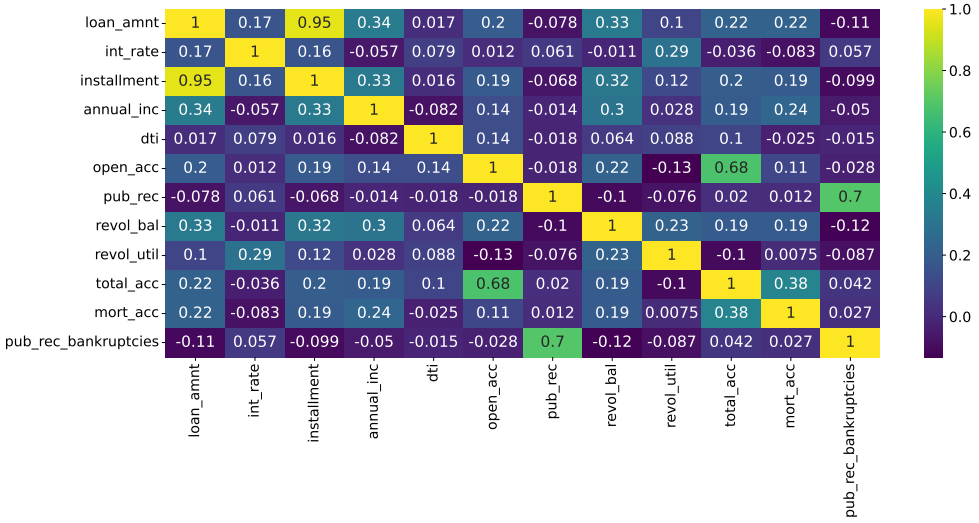

FIGURE 2 – Correlation matrix

3 Modelling [module : machine_learning]

We tried to use two different models, an algorithm such as XGBoost and a neural network model.
Regarding the optimization of these models, we saw as a first instance that the class to be predicted was highly unbalanced, with a ratio of 4 :1.
We decided therefore to adopt a technique to rebalance the dataset called SMOTE (Synthetic Minority Oversampling).
This technique relies on resampling the data in such a way as to create new instances through k neighbors on imaginary lines connecting minority class values.

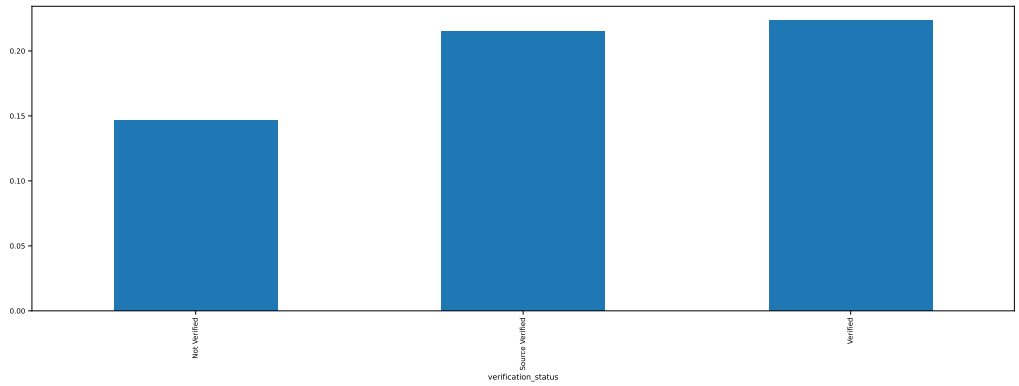

FIGURE 3 – barplot verification_status distribution

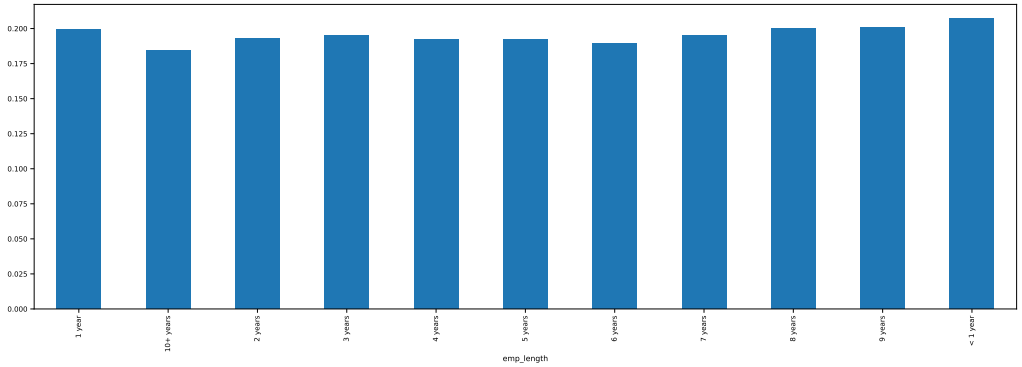

FIGURE 4 – barplot emp_lenght_bar distribution

3.1 X-G Boost

As a first step, we built a preliminary model, instead of determining the best number of trees with cross validation we implemented a 'Early Stopping' callback to stop training when the model doesn't learn anymore.
To evaluate the XGBoost model we used AUC value. Once the first preliminary model is trained we have verified how well it works plotting a confusion matrix.
It can predict very well the majority class but not as well the minority.
For instance we tried to optimaze the hyperparameters also doing Cross Validation, with Grid Search.

3.2 Deep Neural Network

For this model we proceed in several steps.
We built an "inverted pyramid" model and also here to avoid overfitting we used a validation set and we implemented a 'Early Stopping' callback.

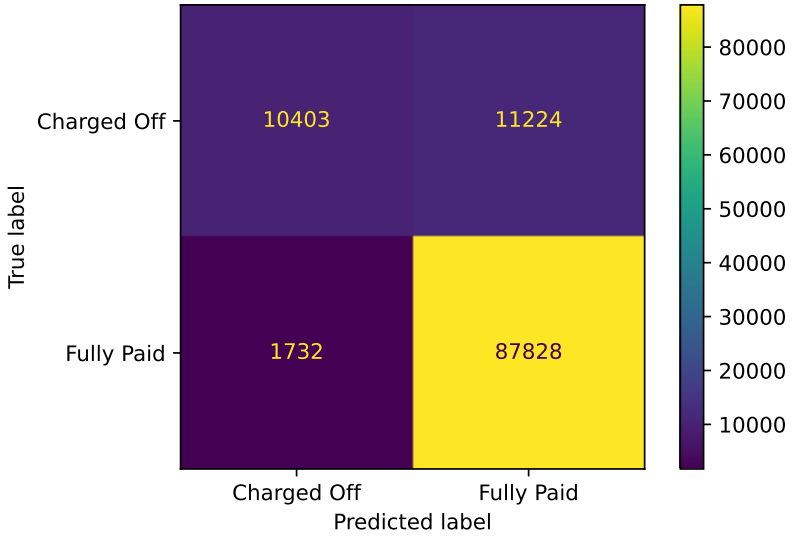

FIGURE 5 – Confusion matrix XG-Boost without hyperparameters

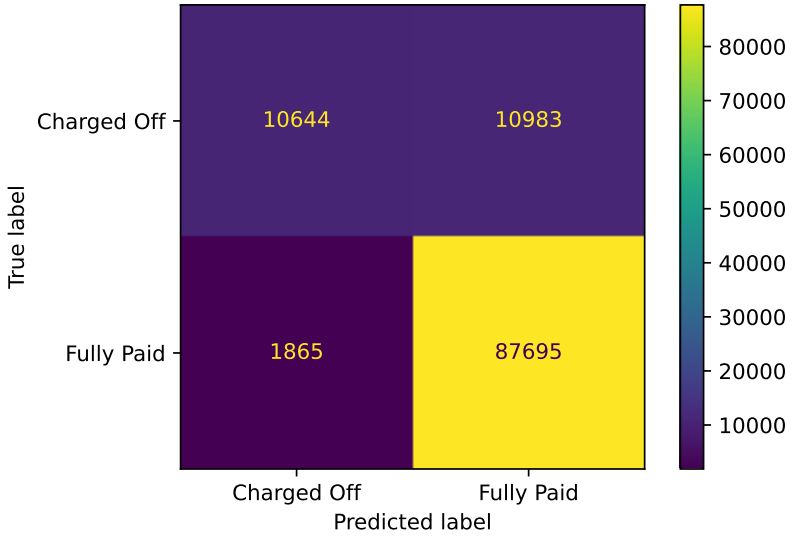

FIGURE 6 – Confusion matrix XG-Boost with hyperparameters

We added other layers using Dropout regolarization and compared the two models.
The regolarization gave us the best results so we procedeed always considering Dropout.
After this we tried to train another model called "diamond" and compare to the precedents one.
All the models we trained and tested give us results very similar but at the end the best one has been the "inverted pyramid" model with Dropout regolarization.

Also in this case we tried to optimize the hyperparamenters, doing Cross Validation, with Random Search.

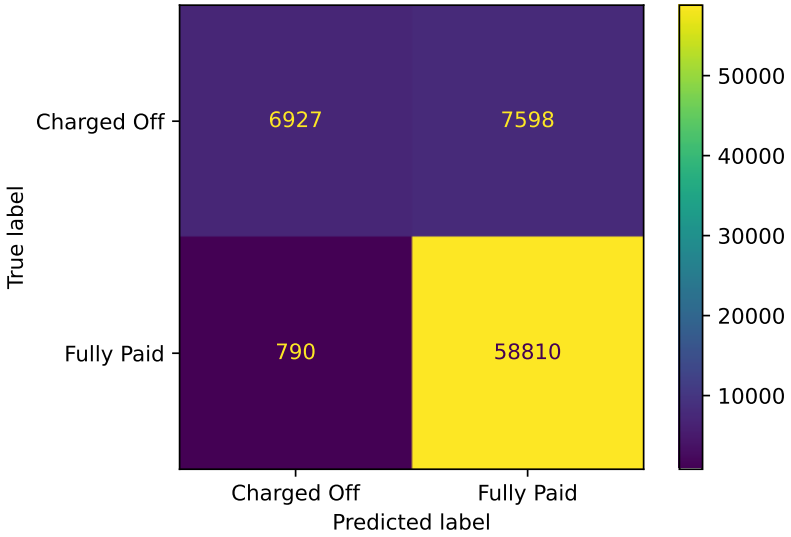

FIGURE 7 – Confusion matrix DNN without dropout

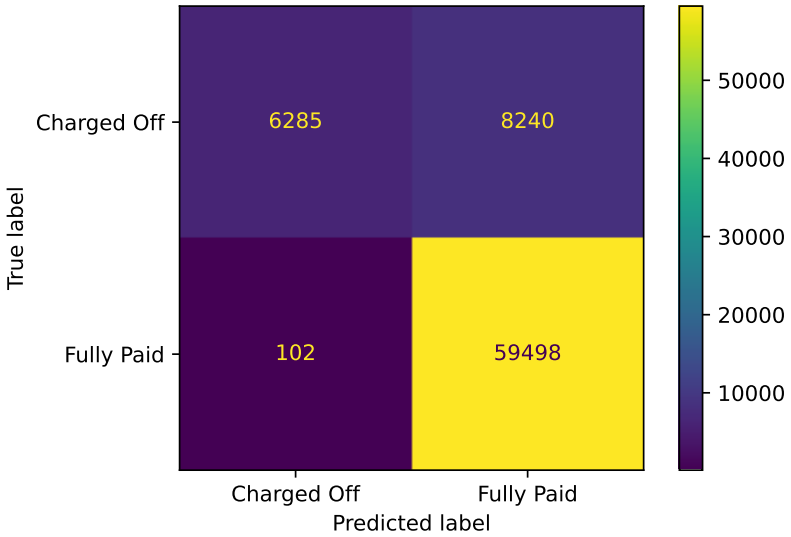

FIGURE 8 – Confusion matrix DNN with dropout

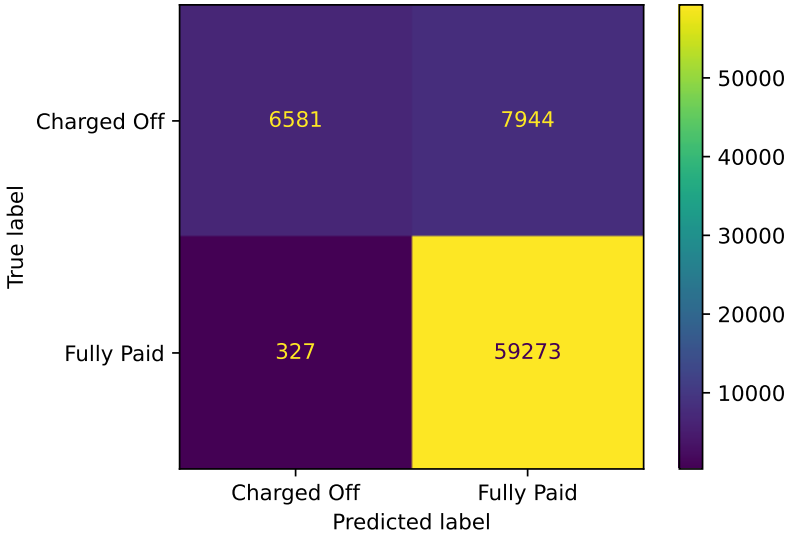

FIGURE 9 – Confusion matrix diamond DNN with dropout

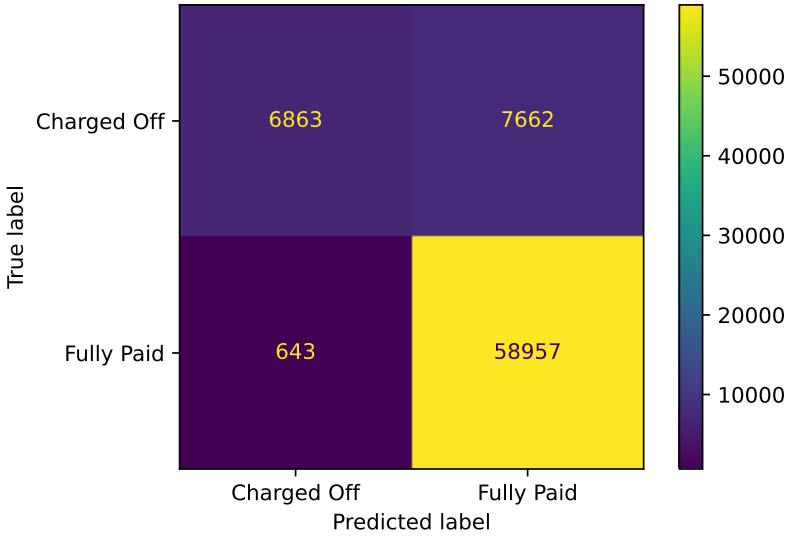

FIGURE 10 – Confusion matrix DNN with hyperparameters

4 Conclusion

In this table we summarize the results of our model :

model	Accuracy	Precision	Recall	F1 Score
Basic XGBoost	0.8835	0.8867	0.9807	0.9313
Hyperparameters XGBoost	0.8844	0.8887	0.979	0.9317
Basic DNN	0.8543	0.9086	0.9104	0.9095
Dropout DNN	0.8874	0.8783	0.9982	0.9344
Diamond DNN	0.8544	0.9086	0.9104	0.9095
Hyperprameter DNN	0.8880	0.8850	0.9892	0.9342

As we can see we have obtained great results in each metrics but f1 score.
On confusion matrix we have obtained a great prediction of the majority class $\sim$ 90 % in every model but a worse one on minority $\sim$ 54 %. This not so optimal classification is due to the beginning unbalanced of our dataset.
From our model, DNN predicts better than XGBoost.