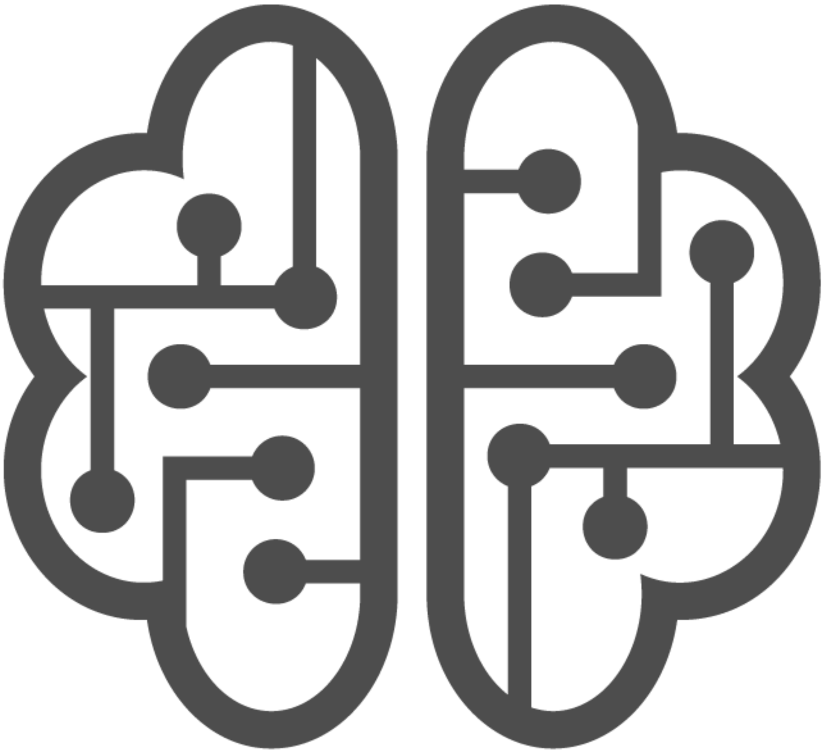


Data mining Application

January 2019

Contents

Introduction:

As machine learning and data mining are showing a lot of great result in different fields, the companies are using it in everywhere the data is available, so it was mandatory to use it in this project to help to predict or to make diagnosis on the university data, so this deliverable is dedicated to explain how we did the data mining and machine learning processes on the data warehouse, in this phase we did a hard work so we will mention the biggest and smallest things we did, in this document also we will explain the main difficulties we faced, the tools and the algorithm we use, and finally the result we obtained. Also here is the Github repository that holds the projects, datasets, code files and images that we did.

Tools we used:

Anaconda Package Manager:

We have used the anaconda package manager to manage our libraries, and software for the machine learning and the data mining process.

For download this software or for more information you can visit the link: https://www.anaconda.com/

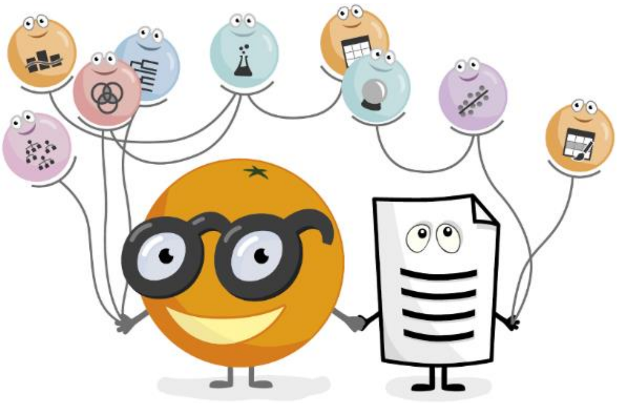
Spyder:

As we worked with Python, a need for python IDE is required, so we used Spyder as an IDE for Python Application Developments, it does come with the Anaconda.

For more information: https://www.spyder-ide.org/

Orange:

When time comes to data mining, the Orange data mining tools comes to play a great role in our process for getting insightful information from the data warehouse we establish,



To download, see features or screenshot visit this link:

https://orange.biolab.si/

Libraries:

As expected we use a lot of libraries to complete the tasks of making machine learning model, here is the list of the libraries:

1) Pandas for loading the dataset.
2) Numpy for numerical processing in python
3) Matplotlib for data visualization.
4) Sklearn: Implementation most of the machine learning algorithm and techniques.
5) Pytorch: Framework to build neural network.
6) Keras and TensorFlow: Framework to build neural network.
7) Other python file for Apriori and Eclat algorithms from Github.

Algorithms:

In the BI4SS project, the main goals are to extract insightful information we called knowledge, and this knowledge will come from the data itself by using some well-known algorithms, we used the up-to-date technique like Deep learning and machine learning in this project, also with some data science methods, here is a list of what we used:

- **Classification and Regression:**

1) Deep Neural Networks.
2) Support vector machine
3) K-Nearest neighbor.
4) Logistic regression.
5) Naïve Bayes.
6) Decision Tree.
7) Random Forest.
8) Linear Regression

- **Clustering**

1) Hierarchical Clustering
2) K-Means Clustering

❖ Association Rules

1) Apriori Association rule.
2) Eclat.

❖ Dimensionality Reduction

1) Principal Component analysis.

For detailed document here is some links: ML Algorithm, ML Algorithm 2, PCA, Association Rules, Apriori Algorithm.

Techniques:

Each Domain in computer science has its own technique and best practices, in the machine learning field there is also technique and best practices that a machine learning engineer should use it during this work, here is what we used as techniques and best practices:

1) Train/Test splitting: train the algorithm on part of the data and test it on unseen data before to get a realistic behavior result.

2) Grid search: to search the best hyperparameter for a given algorithm.

3) One hot encoding: it's a technique to encode categorical data in a way that don't guide the algorithm into a misconception.

4) Standardization and normalization: techniques to make the data in the same scale to prevent the algorithm to use features and ignore features

5) Evaluation metrics: to test how well our learning is going we used some metrics to evaluate it, we used the R2 score and confusion matrix.

6) K-Fold Cross-validation: is a resampling procedure used to evaluate machine learning models on a limited data sample.

7) L1/L2 Regularization for the linear regression model.

8) Dropout Regularization for the neural network model.

Problems:

In the following titles, we will talk about the problems we solved with machine learning and the knowledge we got data mining, a separate zip file need to be with a document which contain python code, images and datasets we did implements

• Problem #1:

In the first problem we did make a machine learning model to help the support staff for accepting new student into the university according to their profile, as a procedure, we did take all the student in their first year from our data warehouse apply some machine learning algorithms, it’s a classification problem in which the algorithm need to determine whether a student will be admitting or will be adjourning based on his socio-demographic profile, the next tables show the algorithms, their architecture, and hyperparameter and finally the result accuracy and the R2 score:

Algorithm	Hyper Parameters	R2 Score	Accuracy
Deep NN	Discuss Later	0.16	82%

KNN	Neighbors = 56
	Metric = Euclidian
		0.12	81%

Naïve Bayes	Gaussian Distribution	-0.45	68%
Logistic Regression	Default	0.11	80%

SVM	Kernel = RBF
	Gamma = 0.001
		0.147	81%

Decision Tree	Criterion = entropy	0.016	78%

Random Forest	# estimators = 50
	criterion =entropy
		0.047	79%

Table 1 Problem #1 Results

As we see from the table above, the neural network, KNN and SVM shows a great result despite the form of that the data we got, we will discuss the problem we got with the data in greater depth in the next few pages, for now put in mind that the data isn't that much sufficient for getting great model.

Neural Network Architecture:

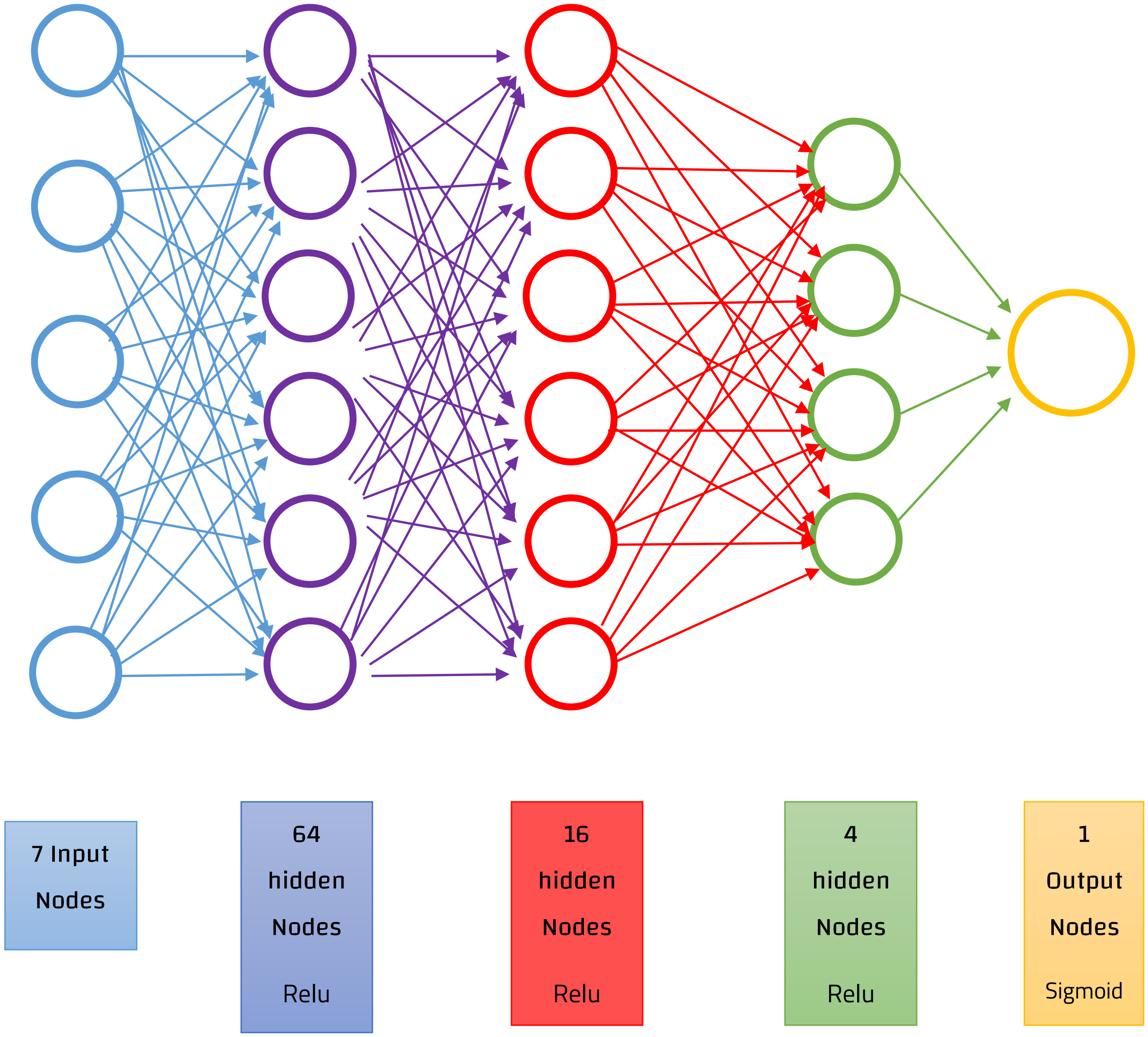


Figure 1 Problem #1 Deep NN Architecture

The neural network shows good scores in both accuracy and R2, isn't that good for real uses, but for this data, it's worked perfectly. Its architecture is one of the things that make it that good, with 3 hidden layers, the network was able to get depth insight to help to classify correctly 82% of the data.

The other part of why this was a good model, is the activations functions and this is another proof that the **ReLU** Activation function fit nicely to most of the networks, the tanh and the softmax activations function wasn't that good for this problem, and the sigmoid was placed at the output layer because the output is binary.

In the training we used a batch size with 20 samples for 100 epochs, alongside with an **Adam** optimizer after we tried the regular Gradient Decent and its variants like the SGD (stochastic gradient decent) with a learning rate of 0.01, all this goes with a metric of accuracy and with a loss function of type **Binary Cross Entropy**.

To prevent overfitting, we did add a Regularization technique in the deep neural network called **Dropout** to help generalize well for the data.

Finally, as this model show good results it will be deployed for the support staff to help them accept new student that can be successful.

For more information visit this links: Adam optimizer , More about Dropout

• Problem #2:

The second problem was about finding some association rule between the courses of the university, this will help a lot the support staff for making good decision about which subject goes with which others, perhaps a mix between the two courses in some projects and so on, the applications are limitless; We have used two known algorithms in finding the associations rule which are **Apriori** and **Eclat**, the two gave a good result, the next table shows the hyperparameters we used and some results:

Algorithm	Hyper Parameters	Results Example

Apriori	Support= 0.05
	Confidence = 0.7
	Lift = 3
		Component Approaches => Admin Networks & Prog Sys

Eclat	Support= 0.05	Advanced Database and Datamining => Logic For AI

Table 2 Problem #2 results

Note that the parameter we use here is specific to the dataset, because that each dataset will has its own support and confidence level, for example the dataset of the example above was the masters One results of the obtained subjects, for instance if we want to get the association rule of the first year LMD students, the hyper parameters will change effectively.

This results will be deployed in the database, as the user choose some subjects to analysis, the associated subjects with the user chosen subject will be displayed in order to get full analysis.

• Problem #3:

For the next one, it was a model for the students to help them in choosing the right branch for them in the master level, at the master level there are 3 branches: Software engineering, Computer Engineering and Networks, and telecommunication, in the first time we would use as features the average of each subject that a student passes in his first 3 years in the curriculum, then with ML techniques we find the relation that would predict the branch type if the student is successful in his current branch, unfortunately this isn't the case, the Software engineering branch was added in the year 2013, some student pass some subjects and some not, in short word, the data is messy and not consistent ! what we did instead is that we extract the averages of the 6 semesters in the curriculum of the students and based on that and their branch, we did establish a model, unfortunately we didn't get great results because of the data again, we tried some technique to find clusters with K-means and Hierarchical clustering, again no good results, an finally we had reduced the dimensionality of the data using the principal components analysis, it didn't give result also, but at least when we did plot the data we deduced from it that there are no clear boundaries to find in order to get a good model, the main issue is that the number of the samples is just 306 students, and this isn't enough number to work with in real world machine learning, despite the problems we did give it a try and the next tables describe the algorithms, parameters and the result we obtained.

- **Classification Algorithm:**

Algorithm	Hyper Parameters	R2 Score	Accuracy

KNN	Neighbors = 10
	Metric = Euclidian
		-0.45	60%

Naïve Bayes	Gaussian Distribution	-0.29	60%
Logistic Regression	Default	-0.73	54%

SVM	Kernel = RBF
	Gamma = 0.001
		-0.57	57%

Decision Tree	Criterion = entropy	-0.56	52%

Random Forest	# estimators = 60
	criterion =entropy
		-0.19	65%

Table 3 Problem #3 classification results

- **Clustering Algorithms:**

Algorithm	Hyper Parameters	Accuracy
K-Means	# clusters = 3	20%
Hierarchical clustering	# clusters = 3	51%

Table 4 Problem #3 clustering results

And as we see from the tables above, no algorithm could find a cluster or relationship in the data to determine the branch of the student based on his past experience at the university.

The following picture shows the 2 principal components of this data that capture the most of the variance.

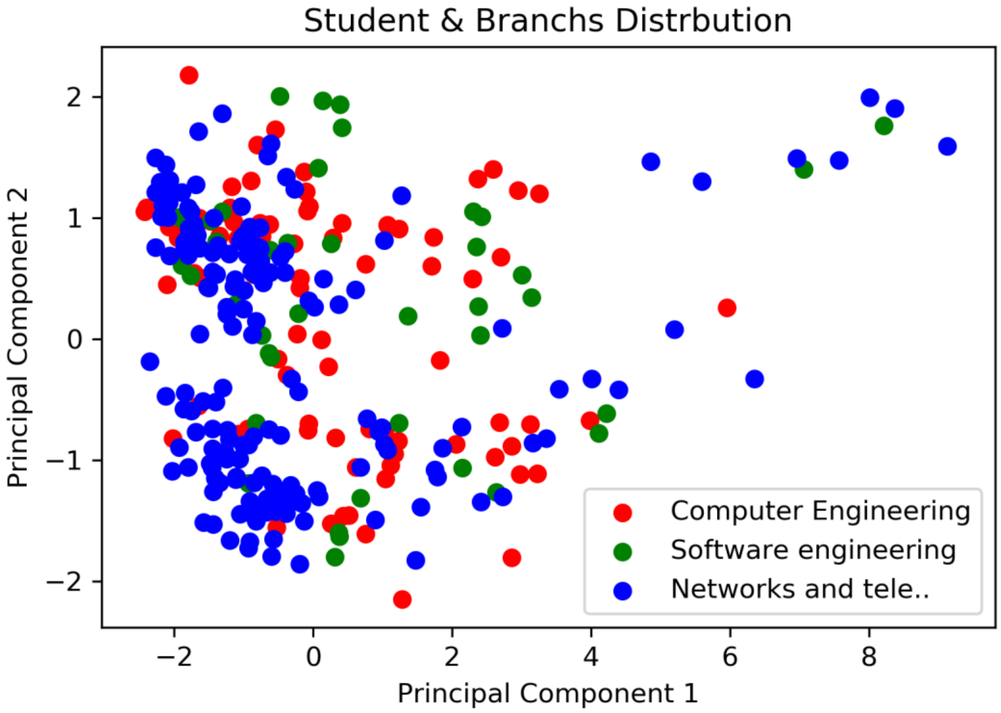


Figure 2 Student & Branches Distribution After PCA

As the above image shows, there is no clear boundary to determine how to do a classification or a clustering on this data, as a consequence no model will be deployed for production until we get a high accuracy in the future.

Problem #4:

Like the past one, students are our interest, so we built another model for the first year Student to help them in choosing the right branch in the second year which consists of Mathematics or computer science, and like the past one, the data isn't in our side. It was 100% against us!!

To make the model that predict based on you result of first-year which branch to choose in order to succeed we need the student in the second and third year with their marks and their results of success or failure, based on this data the algorithm will learn the type of student who can succeed in a given branch.

The problem we got is that in our data warehouse there is no mathematics student who succeeded in his curriculum, so if give this data to the algorithm, as a result, the algorithm will not learn what makes a student succeed in mathematics and he will think that math is very hard and as a consequence he will suggest always Go to Computer Science!

It was a great idea to help students make good choices for their success in the correct curriculum and to prevent failures, unfortunately, the data says no to our ideas and no for this problem to be solved intelligently and automatically, perhaps in the next years when the data will be available we will try again.

- ## Problem #5:

The last problem is a command from the client to prove that the student who gets good averages in the algorithmic subjects in the first year of their curriculum, has a great likelihood to succeed in the rest of it, with a simple visualization of the next scatter plot:

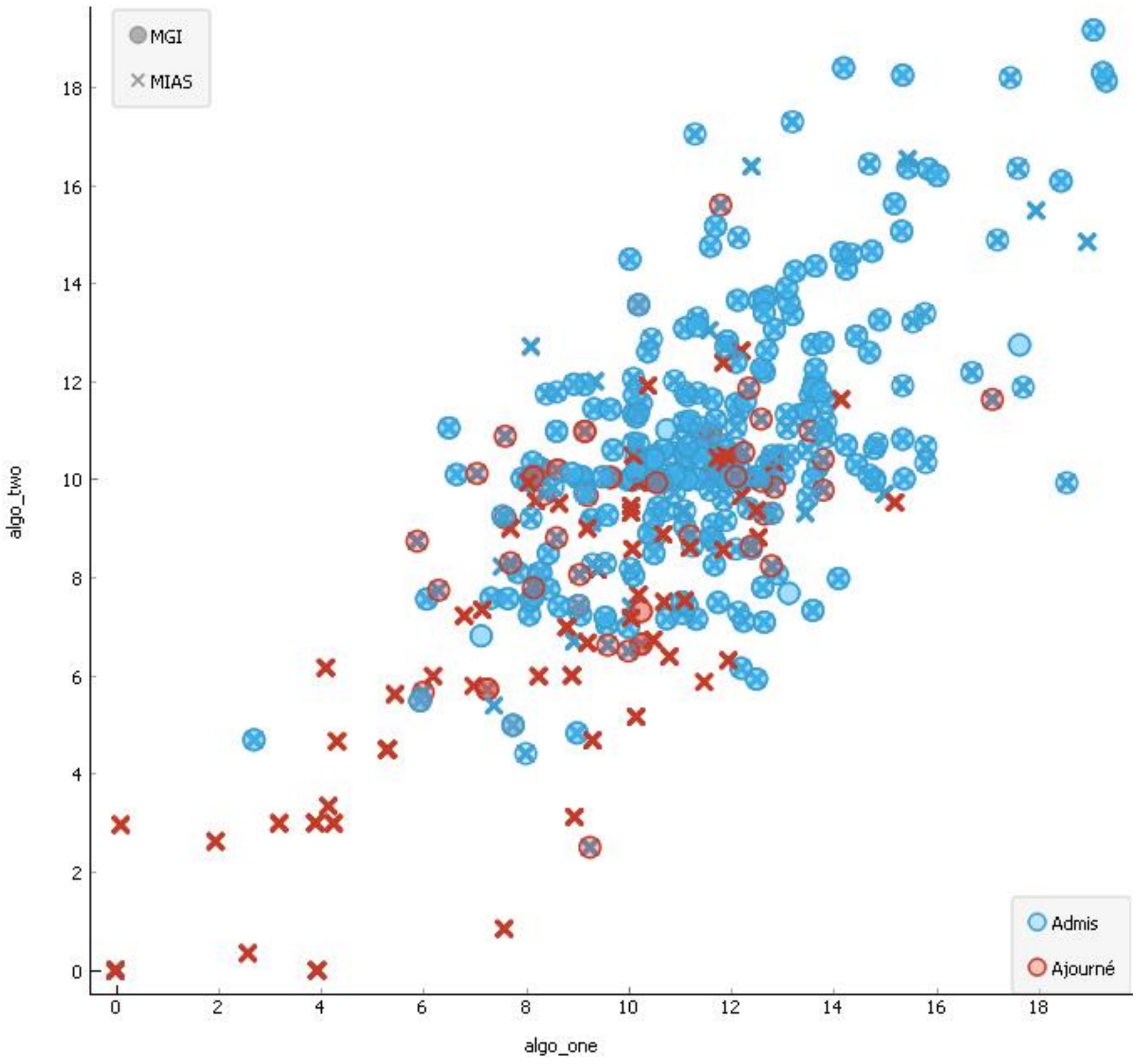


Figure 3 Students Scatter Plot according to algorithmic

Also to prove that relationship we built some ML models. And we got a model of ≈90% of accuracy!!, the associated document shows result of ml models and a pdf that show the distribution of the students according to the algorithmic subjects, here is one image of them as an example:

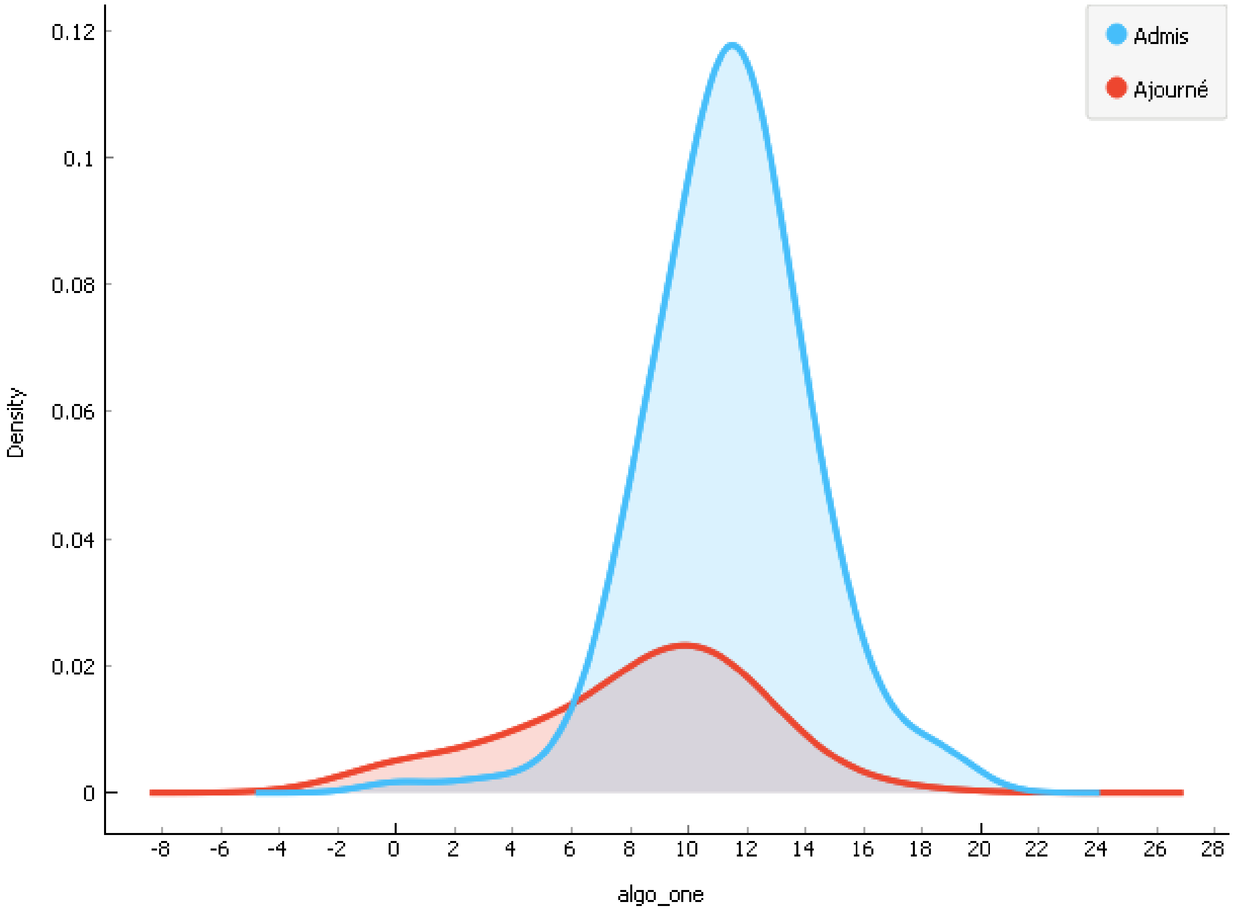


Figure 4 Students distribution according to algorithms

From this figure also we can see that the highest density of the student who admitted in the several are the students who get above the average in the algorithmic one in the first semester, and this also prove our hypothesis that algorithm is one of the main subjects that the student has to acquire to be successful during his curriculum.

Difficulties encountered:

Like every time, difficulties are around everything, and beating them does make us who we are, we encountered a lot of difficulties during this phase for getting some model and knowledge from the data, the first of them was the data itself, it wasn't that much consistent and ready for mining.

After some preprocessing and some ups and downs, and since we have a lot of ideas to work with we had extracted some knowledge like association rules and machine learning models, but not some of these ideas weren't born because of the data either it's sufficient, it requires special preprocessing or simply it's absent, of course, we don't have to blame the data that much, because at the end of the day it's just a data, maybe it's not the correct time to make such an idea or it's so earlier implemented these things.

Conclusion:

Machine learning and Datamining are exciting fields of study, experiment and be creative, we did our best to make, create, deploy good models and extract new knowledge from the data to help the support staff to choose successful students, and to help student by providing a predictive analysis to make them successful students, unfortunately the data wasn't on our side in this project, at least we give it a try and we obtain not that much higher result, but it can work in future versions of the project with good and consistent and well-organized data.