Final Degree Project

Bachelor's degree in Industrial Technology Engineering

Application of data mining technology to analyze and predict academic performance

REPORT

Author: Núria Armengol i Escolà
Director: Lluís Talavera

Call: 04 2020

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

Synopsis

This project attempts to predict the performance of students in their third semester (the first semester of their second year) of the bachelor's degree in Industrial Technology Engineering based on their marks on their first year. To do so two models will be used, decision tree and random forest, as well as several evaluation metrics.

The four evaluation metrics that will be used are accuracy, recall, precision and F1. From all of them the more important will be F1 because is the one that provides a more balanced explanation of how our model is performing.

The objective will be to evaluate how each model performs and compare them, as well as to study how the parameters of each model affect them and in which way. The decision tree will use two parameters, Max_Depth and Min_Sample_Split, while the random forest will use four, the former two as well as N_Estimators and Max_Features. It will be interesting to see how the two decision tree parameters affect the random forest.

Finally this project will find the best combination of parameters for each model achieving an optimized F1. Those results will be compared and they will establish if either model would be worth implementing by the teachers of the third semester subjects.

Sumari

1. Foreword

1.1. Project's Origen

This project was one of the many proposed by teachers and companies to the students of industrial engineering. The instigators of this particular project were the department of Computer Science Department.

1.2. Motivation

When choosing which project to do one of the key elements that were important for me were that it had to be a project where I could achieve a result not only theoretical but practical. I wanted to focus on doing something that could be of use to somebody in a tangible and direct manner. This project picked my interest since me and my peers were the subject of the study and our achievement could, once properly assembled and studied, help pave an easier way for our underclassmates.

Another thing that attracted me towards this study was the programing side which has interested me since my freshmen year when I had my first class on the subject and I'd been hoping for a change to deepen my knowledge and understanding of it.

1.3. Previous Requirements

In order to realize this project some previous knowledge of both statistics and programing were needed. However this knowledge could be achieve by cursing the corresponding subjects in previous years, so I did not have to undertake any further course

2. Introduction

This project's objective is to study the accuracy of the Random Forest prediction technic when applied to the academical results of Industrial Engineering students of ETSEIB. A rigorous methodology of data mining, Pandas libraries and sklearn python will be used to properly achieve this task.

2.1. Data mining

In their book 'Advanced Data Mining Techniques', Dr. Olson and Dr. Delen[1] describe data mining as 'the analysis of the large quantities of data that are stored in computers'. More specifically it's the use of statistical methods to explore huge amounts of information in search of patterns or anomalies. This facilitates the understanding of the data as well as serving as a good basis for predicting how it will continue to evolve and what to do to modify it according to our interests.

Data mining learning can be classified either as supervised or unsupervised depending on whether the data is already labelled with the correct answer or not.[2] **Supervised** learning uses training data labelled with the correct output to learn, building the prediction algorithm using past experience. Since we already have the correct answer for past data it's easy to compare it to the result given by the algorithm using metric evaluators to test it. This is the type of learning used to study, for example, the seasonal flu. In that example the output would be how many people get sick and the algorithm would have data of years past and how many people got sick then which it would use to build a model to predict the new data. This project will also use supervised learning in order to predict which students will fail and which will pass their first semester of second year.

However, we don't always possess previous data. When we are dealing with unlabeled data and letting the algorithm learn on its own we are using **unsupervised** learning. Since this model is blind to start off it helps to find unknown patterns in the data. This is the kind of learning that would be used to explore data in real time when dealing with a new kind of

[1] OLSON David and DELEN Dursun, *Advanced Data Mining Techniques*, 2008

[2] GURU99, *Supervised Vs Unsupervised Learning: Key Differences*, 2020, [https://www.guru99.com/supervised-vs-unsupervised-learning.html]

RECUERO DE LOS SANTOS Paloma, *Tipos de aprendizaje en Machine Learning: supervisado y no supervisado*, 2017, [https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje]

disease for example.

Data mining has become an integral part of many business. For example, supermarkets process thousands of transactions each day and accumulate data from each and everyone of them. Such a large amount of data would be useless if we didn't have a way to explore it and analyses it. Data mining allows us to find and study data about a particular product to determinate selling trends and implement strategies on how to better market it towards the clients.

In order to systematically analyze the data, data mining uses several standard processes, in this project we will study the Cross-Industry Standard Process for Data Mining, called CRISP-DM for short.

2.2. CRISP-DM

This model includes six different phases[3] and depending on the results you achieve there is transactions between phases. The six phases contained are: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment. The phases and the way they interact with each other according the cyclical process can be seen on the Figure 1.

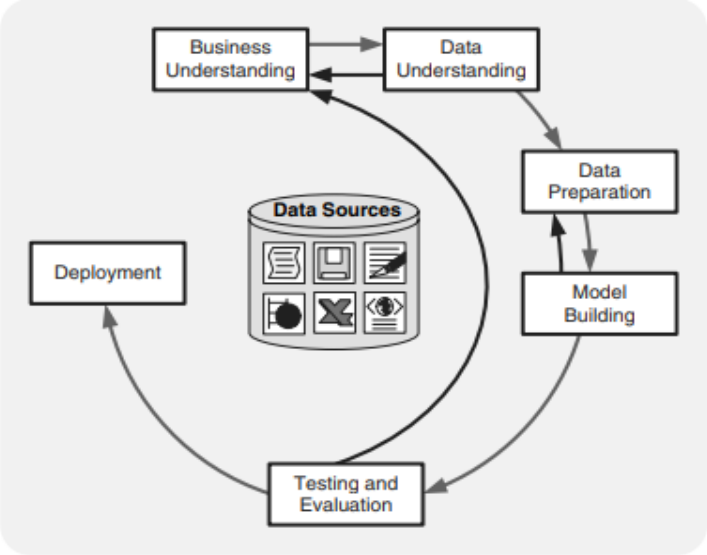


Figure 1: Schematic representation of CRISP-DM

[3] OLSON David and DELEN Dursun, *Advanced Data Mining Techniques*, 2008

RODRIGUES Israel, *CRISP-DM methodology leader in data mining and big data*, 2020 [https://towardsdatascience.com/crisp-dm-methodology-leader-in-data-mining-and-big-data-467efd3d3781]

2.2.1. Business Understanding

First it is important to determinate what the business' objectives are regarding this project, what information is needed to extract from it and how it can be used to its benefit. To do so it's essential to have an in-depth knowledge of the business and to have already some idea on how to improve it using the possible results of the project.

2.2.2. Data Understanding

After establishing which are the objectives of the project we center our attention towards the data we have. In this phase we must select the relevant data from all the available. This is only an initial selection, it's very well possible that not all the data chosen will be useful to us and future processes may filter it out. However it's important to limit the data we subject to these processes to optimize them since otherwise it will be too costly in terms of time and power used.

Overall data can be categorized as either quantitative or qualitative. **Quantitative** data is numeric in nature, be it discreet or continuous, while **qualitative** data expresses a finite number of attributes. Both types are valid in our study even if qualitative data will have to be transformed in the next phase since this project uses sklearn.

It's also important to verify the quality of the data in this step since unreliable or incomplete data may tank the study and must be filtered out. It may help performing cluster analysis to find patterns within the data and identify redundancies and superfluous information.

2.2.3. Data Preparation

The data selected may come in different formats and lengths and it's necessary to homogenize it before starting to study it. In this phase we filter out null values and tidy the data so it's easier to work with. We may also use more in-depth analysis of the data to find outliers and, if needed, remove them from our set of data.

There are several processes that can be used the more common of which are clean, transform and merge. Clean is used to eliminate everything useless or wrong from the data, for example the rows with null values in them. Transform gives the table the desired form by modifying its rows and columns. Finally, merge joins different tables of results into one, allowing a singe dataframe to include all the information wanted.

Qualitative data must also be transformed into numeric values, either binary or not, so they can later be treated with the rest using the algorithms chosen. To do so we equalize a quality with a number (usually 1 and 0 if we only have two options).

2.2.4. Modeling

In this phase data mining software is used to generate results for various situations which helps us get a better understanding of our data. In this project we will be using tree classifier and random forest.

2.2.5. Evaluation

Once we obtain the results from the modeling they need to be contextualize with the objectives fixed for the project. When the results are analyzed some indicators may show the need of reevaluating some of the previous steps to optimize the analyses.

2.2.6. Deployment

Once we've acquired the results and have gained more in-depth knowledge of the business we are ready to implement whatever changes found to be necessary to improve it.

3. Pre-processing

As stated previously, before being able to apply any algorithm to the data, this must be selected and assembled in a tidy fashion.

This project started off three groups of data: 'initial phase', 'non-initial phase' and 'pre-inscription phase'. Each containing data of several thousand students.

3.1. Initial Phase

On the "initial phase" file there's information about the performance of the students in their first year as seen in Figure 2. For each student, it was recorded their marks on a subject in a specific year and call, the degree the student was coursing, the class group they attended, the credits of the subject, the mark of the subject given by the teacher, the mark obtained using the evaluation formula and the definitive mark. All this information was distributed on different rows.

Index	CODI_PROGRAMA	CODI_EXPEDIENT	CODI_UPC_UD	CREDITS	CURS	QUAD	SUPERA	NOTA_PROF	NOTA_NUM_AVAL	NOTA_NUM_DEF	GRUP_CLASSE
0	752	229928	240015	6	2010	1	S	nan	nan	nan	CONV
1	752	229036	240013	6	2010	1	S	nan	nan	nan	CONV
2	753	230399	240011	6	2010	1	S	nan	nan	nan	CONV
3	752	232924	240014	6	2010	1	S	nan	nan	nan	CONV
4	753	232121	240214	6	2010	1	S	nan	nan	nan	CONV
5	752	232295	240014	6	2010	1	S	6.5	6.5	6.5	CONV
6	752	239838	240012	6	2010	2	S	4.5	4.5	5	nan
7	752	230520	240011	6	2010	2	S	6.2	6.2	6.2	nan
8	752	239833	240025	7.5	2010	2	N	1.4	1.4	1.4	52

Figure 2: Initial Phase Data before Treatment.

However, not all the information provided on the "initial phase" data was included on the analysis, because some of the information contained there was found not to be of interest or it added unnecessary variability.

Given the fact that this project only focused on Engineering in Industrial Technologies, the other degrees were purged. Therefore, the column referring to which degree the student belonged became unnecessary and it was dropped.

There were students that had changed degrees and did not course all the subjects because some of them had been recognized from their previous degree. Taken under consideration that those students accounted only for a 4,964% of the overall it was decided that it was a small enough percentage to exclude them from the study. This ensured that the variability

due to studying in other universities or with other curriculums was avoided.

The class group could have been an interesting variable to analyze if there was the certainty that each year had the same teacher imparting that group, or that the same hours were scheduled. Given the fact that those varied from year to year it was decided that this row would be exclude it too.

The subjects were referred by codes which were changed by their respective names to make it easier to interpret the data.

Regarding the number of hours dedicated to each subject it was observe that it didn't necessarily correlate with the difficulty level of the subject. It was moreover information already inferable from the variability owed to each different subject, so this column was also dropped.

As stated before, the data gave us three different marks for student in any given subject. Out of the three marks, the one gotten through the numeric formula was the only one kept since it was the one that only relied on the student performance and avoided being rounded or modified according to the marks of other students.

The first complication was that some students had coursed the same subject more than once. Since it was impossible to determinate with certainty which out of all the marks ranked in this subject was the best suited for this study, three different variables were kept, the average, the first try and the last try. A new variable called 'call' which informed of the amount of tries it took someone to pass a subject was used to select the first and last try and kept since it was deemed possible that it could be relevant when predicting the future performance of the students. These marks were accompanied by their binary counterpart which appeared as a 1 if it was a five or above and 0 otherwise.

Since there hadn't been too many changes on the curriculum of the subjects studied and those were not documented in the data available to us, the information of the year and term were also deemed unimportant and dropped.

Therefore the information that was found to be relevant for the study included on the "initial phase" file was the average performance of a student in every subject, the performance of a student in their first call of each subject, the performance of the student in the last call of each subject and the number of calls they had cursed for each subject.

3.2. Pre-inscription Phase

In the pre-inscription file there was personal data about the students from before they started university, this can be seen in Figure 3. Out of this data only the access mark, the sex and the Postal Code (either of their family or education center) seemed relevant and not reductant with the information already gotten from the Initial Phase.

Index	CODI_EXPEDIENT	SEXE	CP_FAMILIAR	ANY_ACCES	TIPUS_ACCES	NOTA_ACCES	CP_CENTRE_SEC
0	274511	H	08640	2013	1	12.99	8640
1	275156	H	43002	2013	1	13.018	43002
2	259794	D	08006	2012	1	12.65	8021
3	262031	D	08017	2012	1	10.74	8017
4	261879	D	08504	2012	1	11.696	8500
5	258115	H	08907	2012	1	10.346	8907

Figure 3: Pre-Inscription Data before Treatment

However, the Postal Code could take on too many values and while some referred to a town, others were a just a section of a neighborhood which made it quite difficult to work with and unlikely to be relevant.

Also, given the disparity between the number of men and women in engineering and similar degrees and the outreach programs done with the objective of decreasing this problem the data provided by the sex of the student could be subject to too many variables and thus not very reliable to base predictions on, so it was also decided to leave it out.

Consequently, the only data kept for this study from the pre-inscription file was the access mark to the university which is calculated using the students' performance in the last two years of high school and their marks on a national exam and it is a way to determine the level of knowledge they had prior starting the degree. Figure 4 shows the resulting data.

Index	NOTA_ACCES
226378	11.044
226410	12.507
226431	11.796
226441	9.852
226455	10.642
226463	10.004
226464	10.916

Figure 4: Pre-Inscription Data once Treated

3.3. Non-Initial Phase

Figure 5 shows that this last set we had the information we wanted to predict which is to say the marks of the students in their first semester of second year. Out of all this information, only the first semester will be studied, and we will only consider the students that have completed it so as to have regular sets of data and exclude any variables like students cursing subjects of other semesters as well as the ones we are taking into account.

Index	CODI_PROGRAMA	CODI_EXPEDIENT	CODI_UPC_UD	CREDITS	CURS	QUAD	SUPERA	NOTA_PROF	NOTA_NUM_AVAL	NOTA_NUM_DEF	GRUP_CLASSE
0	752	289160	280A006	3	2016	1	S	nan	nan	nan	nan
1	752	230843	240041	6	2011	2	S	nan	nan	nan	CONV
2	752	229928	240131	6	2011	1	S	8.2	8.2	8.2	10
3	752	256156	240033	4.5	2011	2	S	5	5	5	nan
4	752	227207	240032	4.5	2011	1	S	7.4	7.4	7.4	10
5	752	228015	240032	4.5	2011	1	S	6.3	6.3	6.3	10
6	752	230666	240032	4.5	2011	1	S	7.6	7.6	7.6	10
7	752	227765	240032	4.5	2011	1	S	7.6	7.6	7.6	10
8	752	262674	240052	6	2012	1	S	5.6	5.6	5.6	nan

Figure 5: Non-Initial Data before treatment

As we've done with the initial phase data, we select only those students of Engineering in Industrial Technologies and drop the column that tells us which degree they are taking once we've done so.

The prediction processes used can only make a binary prediction, in this case whether the student will pass or fail. As this data is the one that we will use to base our predictions on and to check whether they are acceptable or not we also need it to be binary. For this reason, we only keep the information regarding which subject we are talking about and the column that tells us if the student has approved or not dropping all the others. Once that is done all that's left is exchange the numeric codes for the subjects they represent, switch 'passed' for 1 and 'failed' for 0 and rearrange the columns so at the end of each row you can see how each student has fared in each subject. The result can be seen in Figure 6.

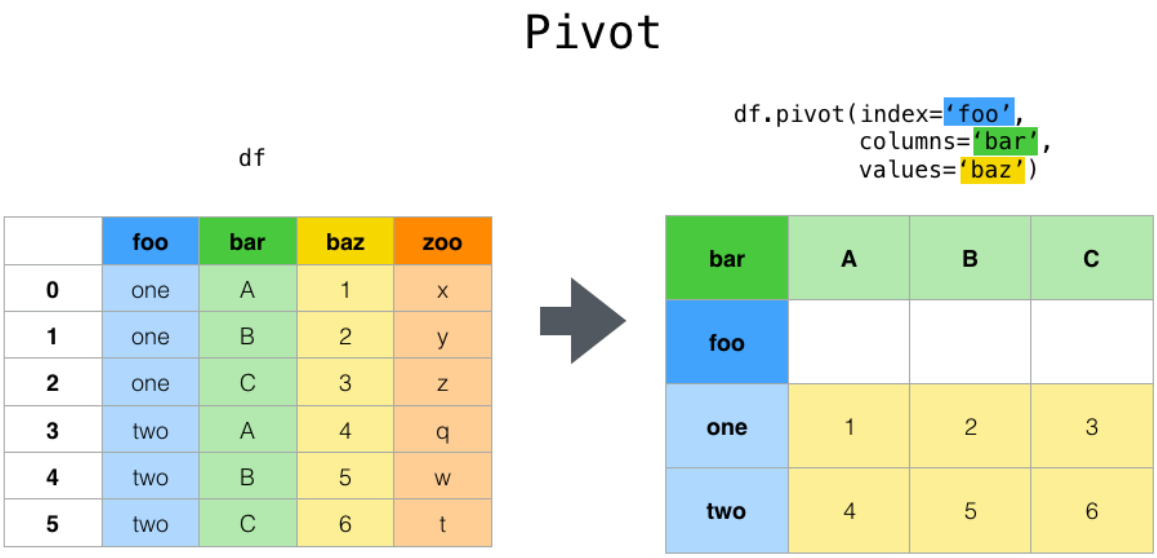
Index	RA', 'Electromagne	', 'Equacions Dife	JPERA', 'Informati	UPERA', 'Material	UPERA', 'Mecanic	RA', 'Metodes Nur
226410	1	1	1	1	1	1
226431	1	1	1	1	0	1
226455	1	1	1	1	1	1
226464	1	0	1	1	1	1
226467	0	1	1	1	0	1

Figure 6: Non-Initial data after treatment

3.4. Final Table

Once we've gotten the previous three data frames filtered accordingly we need to create a single set of data that can be easily studied. To do so we first need to transform the data frames we have.

The Pre-inscription set had two columns for each index, one was the student's number and the other their access mark. Using the method 'pivot_table' we selected the first column containing the student's number as the index thus leaving the column with the access marks as the only one in the data frame. An example of 'pivot_table' can be seen in Figure 7[4] which shows how this method works. It takes the values of the column and transforms them into columns of their own, the values of those columns can be assigned, like in the example where there were in the 'baz' column, can be a combination of several columns or even be new values.



	foo	bar	baz	zoo
0	one	A	1	x
1	one	B	2	y
2	one	C	3	z
3	two	A	4	q
4	two	B	5	w
5	two	C	6	t

bar	A	B	C
foo			
one	1	2	3
two	4	5	6

Figure 7: Example of 'pivot_table'

With the Initial Phase and Non-Initial Phase data frames it was slightly more complicated since they had a row for every call in every subject a student had taken the course. First it was necessary to leave the data frame with only one call for student in the Non-Initial Phase

[4] MOFFITT Chris, *Pandas Pivot Table Explained*, 2014, [https://pbpython.com/pandas-pivot-table-explained.html]

set we only conserved the first call while in the Initial Phase we allowed three options: first call, last call and average

Now that all subjects cursed by a student only had one row we used the method 'pivot_table' to select as each row's index their student's number and subject. This left us with a first index, the student's number, which had several indexes, the subjects, each of which had a row. The method 'unstack' allowed us to transform that into rows whose only indexes were the student's number and which had their columns labelled according to each subject.

This gave us several data sets with the student's number as indexes and we used the method 'merge' to create a single data frame, joining data according to the indexes. Once that was done we realized that we had rows with incomplete data (for example of students that had not passed their third semester and consequently didn't have a row in the Non-Initial Phase set) as we had already decided that we'd only select students that had a complete set of data we dropped those rows.

The code used can be found in Annex 1 and the end result can be seen in Figure 8.

Index	'Geo	ORIA', 'Mecanica	/OCATORIA', 'Qui	CATORIA', 'Qui\xa	.IA', 'Termodinami	MITJANA', 'Algebr	TA_MITJANA', 'Cal	A_MITJANA', 'Calc	ITJANA', 'Expressi	ANA', 'Fonaments	_MITJANA', 'Geor	ANA', 'Mecanica F
226410		1	2	1	1	6.6	7.6	7	5.1	8	7.6	6.3	
226431		2	1	1	2	5.1	6	6	8	5.6	5	5.6	
226455		1	1	1	1	8.1	7.2	7.3	5.4	9.2	6.6	5.8	
226463		2	4	2	2	3.73333	5.2	5	6.7	3.23333	4.4	3.55	
226464		2	1	1	2	4.5	5	5	7.2	5	5	5.8	
226467		3	1	3	2	4.86667	4.5	3.6	5.8	3.7	5.75	4.8	
226472		1	1	1	1	7.8	7.9	7.8	6.5	5.05	7.3	5.7	
226494		1	1	1	1	5	6.6	6.1	5.2	3.43333	6.5	5.7	
226495		1	1	2	1	6.2	5.1	5.8	5.5	3.93333	6.1	5	
226499		1	1	1	1	7.3	6.6	7.1	7	6.5	7.1	5	
226515		1	1	1	1	7.8	6.1	7.4	8.7	7.3	6.3	7.4	
226543		1	1	1	2	7.4	6	6.25	5.9	6.9	6	6.6	
226635		1	1	1	1	5.3	5.5	5.9	6.7	5	6.3	5	
226648		1	1	1	3	5.6	6.6	5.4	5.9	4.8	5.4	5.5	
226679		1	1	2	3	5.5	6.5	5.3	5.4	7.6	4.7	6.3	
226680		1	1	1	1	6.8	7.6	8.2	6.9	7.2	8	6.1	
226684		1	1	2	1	6	6.1	4.95	2.7	5.1	5.8	5.5	
226685		1	1	1	1	5.7	5.4	4.8	6.8	5.3	4.4	7.5	
226687		1	2	1	2	4.53333	5	6.1	3.5	2.46667	5.3	5.6	

Figure 8: Final Data

4. Validation

4.1. Estimation Method

As stated previously, both Decision Tree Classifier and Random Forest are supervised learning algorithms. To train them data labelled correctly with the output must be provided. [5] It is also important to test them later with data that has not been used to teach the model to see whether the algorithm does appropriate predictions. There are two main methods to split the data: hold out and cross-validation.

4.1.1. Hold Out

Hold out is the simpler option where you split the data into only two groups: training set and testing set. The training set will be used to create the model while the testing set will be used to evaluate it. Our model learns through the training set data so it cannot be used to test it, that's why the second set is needed.

Using the method 'train_test_split'[6] we can select which percentage of our data is set aside for testing. This ratio is important because with a smaller training set the parameter estimates have greater variability but on the other hand if the testing set is not big enough the model statistic performance suffers. The less parameters you want to tune in the algorithm the smaller your training set needs to be. At the same time, the more data you have the smaller your training set needs to be. For a project with a size like ours it's recommended a 70-30 ratio because it gives both sets enough data to function correctly.

'Train_test_split' also has another parameter that's important to correctly fix to obtain a good result. This parameter is 'stratify' and it evens the proportion of true and false values in each set. In our project this means that both the training set and testing set will have the same ratio of passed-failed marks. An easy way to show why that's important is to ask ourselves what would happen if our testing set had all the passing marks and the testing one all the failing ones. If that were the case our model would fail any evaluation we did since it wouldn't be able to predict failing marks.

---

[5] ALLIBHAI Eijaz, *Hold-out vs. Cross-validation in Machine Learning*, 2018, [https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f]

[6] SCIKIT-LEARN, *sklearn.model_selection.train_test_split*, 2019, [https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html]

Lastly, 'random_state' is a parameter which guaranties that our model makes the same choice every time that the random number generator is called. Thanks to that it always divides the data in the same why which allows us to compared result since they are all achieved using the same sets of data.

4.1.2. Cross-Validation

When using cross-validation instead of dividing the data into two groups you divide it into a number (k) of them. One of this groups is used as the 'test' while k-1 others are used as the 'train'. After the model is trained the process is repeated with each of the other k-1 groups as the 'test'. Figure 9 shows an example of cross-validation where k=5.

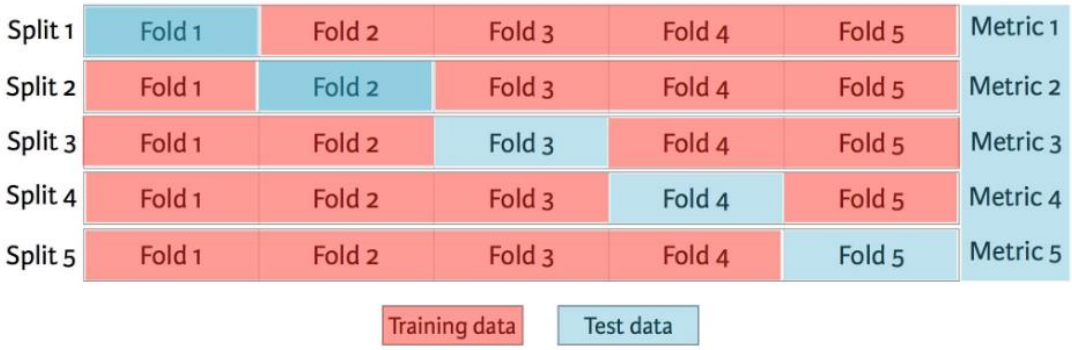


Figure 9: Example of 5-fold cross validation

Given that cross-validation allows you to test several training sets, it's the best method to predict how your model will predict new data. It also reduces your model dependence on the train-set ratio from the hold out method. On the other hand, cross-validation is a very expensive method which requires a lot of time and computational power since it has to run through every combination of sets.

Considering the expense of cross-validation, our project will be done using hold out to decide the training and test set. However cross-validation will be employed internally by a method used to test different parameter combinations.

4.2. Evaluation Metrics

There are several metrics[7] that can be used to evaluate whether our predictions are good or not. In this project, we will make use of the following five: the confusion matrix, accuracy, recall, precision and F1. They all are focused on predicting the 'positive' so in this project, we will select 'failed' as positive and 'passed' as negative given that our main concern is to identify which students will need more help.

4.2.1. Accuracy

This is the most widely known model but it may not be the best in all cases. It measures the number of items that have been correctly classified. To do so we follow this formula shown in Figure 10.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Figure 10: Accuracy formula

The principal problem of accuracy is that even though it may seem that this gives us an objective idea of how well our predictions are it only works when there are a similar number of positive and negative numbers. If there is too big of a difference the system will default into predicting all future data as the category in majority and it will have an excellent accuracy even when those predictions are bad.

For example, if a hundred people visit a place and only five of these people return sick a 95% accurate prediction would be that everyone that returns is healthy. That could be a potentially dangerous error even if it's only 5% inaccurate.

In our case, if nine students failed and one passed, predicting that all the students would fail would be 90% accurate.

---

[7]DRAKOS George, *How to select the right evaluation metric for machine learning models*, 2019 [https://medium.com/@george.drakos62/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-3-classification-3eac420ec991]

PING SHUNG Koo, *Accuracy, Precision, Recall or F1?*, 2018, [https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9]

4.2.2. Confusion Matrix

This evaluation metric will be used mainly as a visualizing tool or as a base for other metrics. It consists on a table with the same number of rows and columns which relate to the number of datasets we are comparing. In our case we will only need four quadrants.

Top left is the **True Positive** cell which tells us how many of the values predicted as 1 are truly 1. Top right is the **False Positive**, the number of values predicted as 1 that are 0. Bottom left is the **False Negative**, the number of values predicted as 0 when they are 1. Finally bottom right is the **True Negative**¸ the number of all the values that are correctly predicted as 0.

As Figure 11 shows, the first row shows the values predicted as 1 and the second the ones predicted as 0 while the first column show the values actually 1 and the second the ones that are actually 2. For this reason, in an ideal situation there would only be values in the diagonal.

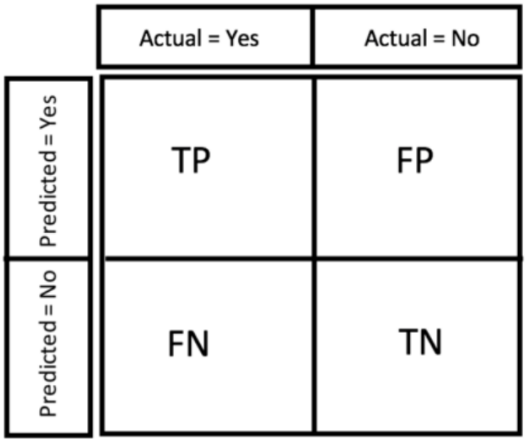


Figure 11: Confusion Matrix

4.2.3. Recall

The recall, also called the True Positive Rate measures the number of values we have correctly classified as positive in front of all the values that are positive. The formula we follow can be observed in the Figure 12.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Figure 12: Recall formula

Recall punishes the False Negative, the items classified as negative that should be positive, so in the example used previously for accuracy the recall would be 0% (in this case positive would be sick and negative healthy).

If our predictions were used as a guide for teachers to give some extra help to the students that needed it most, recall would show which percentage of the students that would have benefited from it were reached.

4.2.4. Precision

Precision refers to how many of the positive values are classified as positive by using the formula in Figure 13.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Figure 13: Precision formula

In the example given when explaining the drawbacks of accuracy, it would also be a 0% which tell us that a 95% accuracy means very little if we also have a 0% recall and 0% precision. But while recall would measure how many sick people we've classified as sick, precision measures how many of the people we've classified as sick are truly sick. Ergo recall is needed so we don't leave out anyone that needs medical help while precision is needed so we don't waste resources on someone that is fine.

On our project precision informs us about the percentage of people that truly needed help, out of all those that did received it.

4.2.5. F1 Score

As we have seen both recall and precision are very important, and it may be difficult to choose which one to use. When in a position where only a single metric value must be used, a mix of the two is needed. That is what F1 score is. Unlike accuracy it's a good fit when there is disparity between the amount of positive and negative and uses the formula seen in the Figure 14 to seek a balance between precision and recall.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Figure 14: F1 formula

5. Decision Tree Classifier

A decision tree classifier is a supervised learning algorithm works by splitting the data according to different parameters[8]. The classifier tree splits the data using binary questions. It is a great tool to visualize and represent decision making. It is drawn upside down, each node representing a condition that the tree uses to split into two different branches depending on whether the condition is met or not. When the branch can no longer be split we arrive at a final node also named leaf. Figure 15 show these three components in an example.

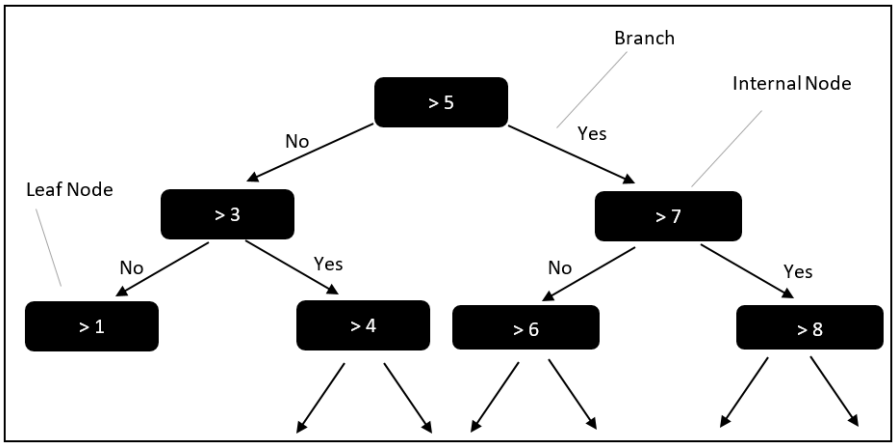


Figure 15: Example of a Decision Tree Classifier

While a decision tree allows us to quickly classify unknown data and is easy to interpret, it also has its drawbacks. The main one is that a decision tree is easy to overfit. Overfitting happens when our model, be it a decision tree, random forest or any other, is so specific that while it predicts the training data perfectly it has difficulties predicting any data that slightly differs. For example, if our overfitted decision tree decides that students that pass are those that have cleared all the subjects, did so in the first try and had a good access mark if a student fails to clear even one of those questions they won't be predicted as passed regardless of how well he did on the rest.[9]

---

[8] GUPTA Prashant, *Decision Trees in Machine Learning*, 2017, [https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052]

[9] ELITEDATASCIENCE, *Overfitting in machine learning*, 2019,

https://elitedatascience.com/overfitting-in-machine-learning]

To avoid over or under fitting a tree and get the best results, it's important to correctly tune decisions trees classifiers' parameters[10].

5.1. Parameters

5.1.1. Max_Depth

Max_Depth marks the maximum depth we allow the tree to develop. The highest this depth is the more splits and information we can acquire but also the highest the chance we will overfit our model. On the other hand, if it is a very low value we will not let our tree grow enough to give us useful information. If left at default, the tree will expand until all their leaves contain less than min_sample_split samples. The value we give this parameter will be decided later during the tuning.

5.1.2. Min_Samples_Split

Min_Sample_Split is the minimum samples an internal node must have to be split. This parameter is usually used to avoid overfitting because it stops the tree to grow too specific to a concrete set of data. By not allowing the tree to split small nodes we avoid it learning hyper specific relations that may only apply to the training set being studied. Like with max_depth, this parameter will be further tuned.

5.1.3. Random_State

Random_State allows to fix the seed of the random generation, in order to get consistent results. Basically, every time the tree is rerun it will give the same one random result. This will be useful when there is the need to compare how a different value in a parameter affects the same tree. The number given it is irrelevant if it is kept constant.

[10] BEN FRAJ Mohtadi, *InDepth: Parameter tuning for Decision Tree*, 2017, [https://medium.com/@mohtedibf/indepth-parameter-tuning-for-decision-tree-6753118a03c3]

5.2. Parameter Tuning

First, we fit the data with a 7-3 ratio. As stated above we shall be using hold out stratified to determine the train and test sets. Once this is done we fit the tree leaving all parameters of in default except for the Max_Depth and Min_Sample_Split.

In order to choose the best values for both parameters the best way would be to do a massive 'GridSearchCV'[11] which would allow us to evaluate the decision tree using every combination of the all values for each parameter. 'GridSearchCV' would tell us which the best values are but to do so we would have to specify which metric to use. However, this method is too expensive since it would have to run through every single value so we will select smaller groups of values for each parameter to test them. Figure 16 shows a representation of the procedure.

To do so, we first make a search to see what tendency they have, since we will be using cross-validation we are not looking for concrete values, but intervals based on the observed tendency. Once we have selected a group of the best values for each parameter we do a cross-validation using the method 'GridSearchCV'.

Our objective is to predict as many failing marks as possible which is information given by Recall but at the same time, avoid mistakenly classifying too many passing marks, therefore Precision must not be too low. To have a more global view of the data we will choose F1 since it is a balance between the two metrics.

Finally, we will evaluate the accuracy of the best tree chosen according F1 to check that there's no need to sacrifice it and that it's high values don't necessarily correlate with the best performances. We will also record its F1, recall and precision to compare it with the result we will obtain using random forest.

The code used can be found in Annex 2.

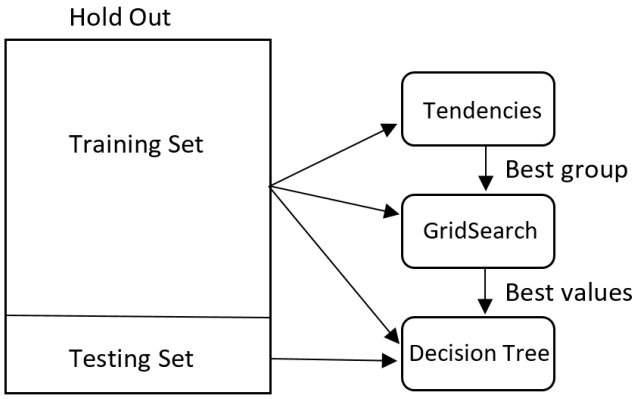


Figure 16: Parameter Tuning

[11] SCIKIT-LEARN, *sklearn.model_selection.GridSearchCV*, 2019 [https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html]

5.3. Results

5.3.1. Max_Depth Tendency

The tendency of Max_Depth as shown in Figure 17 is to increase the performance of the model until it arrives to a maximum value after which it achieves stability and does not increase. It can also be observed that the precision is very prone to overfitting since in most subjects it ends up decreasing once the maximum value has been surpassed. This is consistent with our knowledge of max depth: the more we let our trees grow the more information we get from them, but it comes to a point where there is not more to learn (hence the stability) or that we have overfitted the model (and therefore performances decrease).

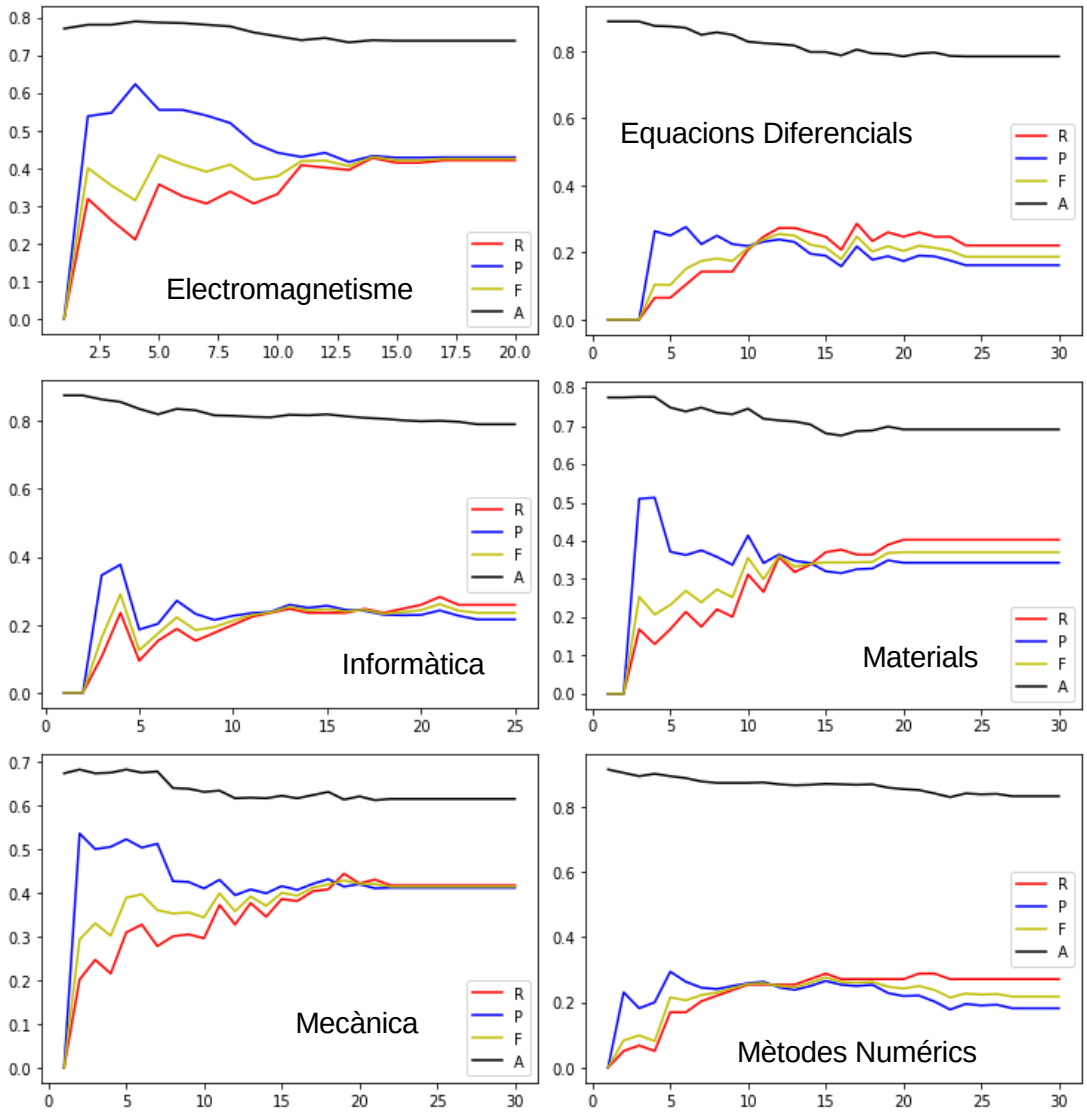


Figure 17: Max_Depth tendency.

It is also worth noting that the subjects being analyzed can be classified into two groups depending on which value of F1 (and usually recall and precision too) they arrive at. Electromagnatisme, Mecànica and Materials can achieve an F1 of 40% while Mètodes Numérics, Informàtica and Equacions Diferencials can only manage an F1 from around 20%.

This separation seems to depend on the rate of failing grades of the subjects. The three more difficult ones where the number of passing and failing marks are similar appear to be more easily predictable. The three subjects were most people pass give our model more problems.

Regarding the intervals we need to select to do the cross validation we will seek the ones around the maxim value as long as that value isn't a peak. If this interval is too long (for example if it reaches the stable part) we will select the lower values to avoid overfitting our tree. Figure 18 chose the ones we have chosen for each subject.

SUBJECT	INTERVAL
Electromagnetisme	[10,11,12,13,14,15]
Equacions Diferencials	[10,11,12,13,14,15]
Informàtica	[20,21,22,23,24,25]
Materials	[15,16,17,18,19,20]
Mecànica	[18,19,20,21,22,23]
Mètodes Numèrics	[13,14,15,16, 17, 18]

Figure 18: Table with the interval of Max_Depth for each subject

5.3.2. Min_Sample_Split Tendency

Figure 19 shows us that the highest the Min_Sample_Split is the worse our model performs. If the number is high enough the model cannot predict anything at all since the tree can't grow. The lowest the value we chose is the highest the F1 and recall but the precision may increase along with the value which again points that it's the value more easily overfitted and that if the Min_Sample_Split is too low that causes the model to be too specific to the training set and consequently to be bad at predicting other data.

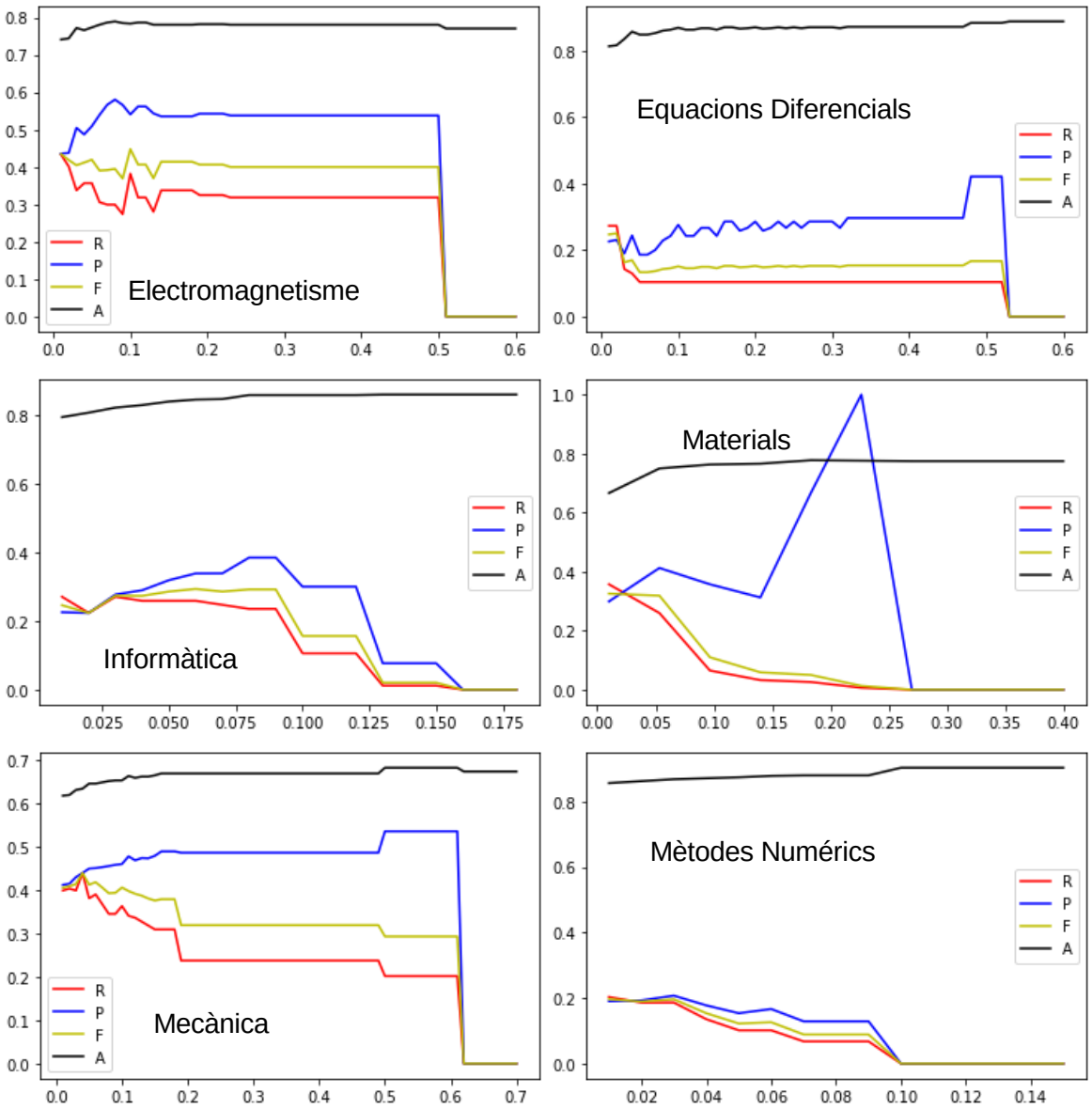


Figure 19: Min_Sample_Split tendency

Like with Max_Depth we can observe the two groups of subjects which arrive either to around to 40% or 20%. Since it's happening also with this parameter it appears that this separation is not a coincidence.

We select several intervals of values from around the highest F1 values ignoring peaks and attempting to get choose them as high as possible as to avoid overfitting. Figure 20 shows them.

SUBJECT	INTERVAL
Electromagnetisme	[0.02,0.04,0.06,0.08,0.1]
Equacions Diferencials	[0.01,0.02,0.03,0.04,0.05]
Informàtica	[0.03,0.04,0.05,0.06,0.07,0.08]
Materials	[0.01,0.02,0.03,0.04,0.05,0.06]
Mecànica	[0.05,0.06,0.07,0.08,0.09,0.1]
Mètodes Numèrics	[00.01,0.015,0.02,0.025,0.03]

Figure 20: Min_Sample_Split intervals chosen for each subject

5.3.3. Evaluation

As previously explained, Gridsearch allows us to run the tree with each pair of values for the intervals selected and it returns the best combination. In Figure 20 we can see that while those values tend to be the lowest of the interval for Max_Depth and the highest for Min_Sample_Split (like we had theorized) that is not always the case.

Now that we have tuned the tree and we're dealing with values and not tendencies, it seems that instead of two groups, it forms three. The first, formed by Mecànica and Electromagnetisme which are the two subjects with more failing marks, has a F1 of around 40%. The second F1 is around 30% for the subjects Informàtica and Materials. Finally, the two easiest subjects, Equacions Diferencials and Mètodes Numèrics, have a F1 of around 20%. If this correlation keeps happening when we study the data using random forest it will be interesting to compare the number of failing marks with this tendency.

This values of F1 (and their recall, precision and accuracy that can be seen in Figure 21) are the ones that we will use as reference to compare the results obtained by using random forest.

	Max_Depth Interval	Min_Sample_Split Interval	Best Parameters	F1	Accuracy	Precision	Recall
Electromagnetisme	[10,15]	[0.02,0.1]	MD= 12 MS = 0.08	0.4	0.79	0.58	0.3
Equacions Diferencials	[10,15]	[0.01,0.05]	MD= 10 MS = 0.04	0.21	0.88	0.37	0.14
Informàtica	[20,25]	[0.03, 0.08]	MD= 20 MS = 0.08	0.29	0.86	0.39	0.24
Materials	[15, 20]	[0.01, 0.06]	MD= 15 MS = 0.06	0.3	0.75	0.41	0.24
Mecànica	[18, 23]	[0.05, 0.1]	MD= 19 MS = 0.09	0.39	0.65	0.46	0.35
Mètodes Numèrics	[13, 18]	[00.01,0.03]	MD= 13 MS = 0.03	0.19	0.87	0.21	0.17

Figure 21: Best Values for the Decision Tree

6. Random Forest

Random forest consists of several decision trees all of which put forward a classification, the result with more votes is the one accepted as true. For example, if three trees decide a student will fail but seven say he will pass the, random forest will defend that they will pass. Another example can be seen in Figure 22.[12]

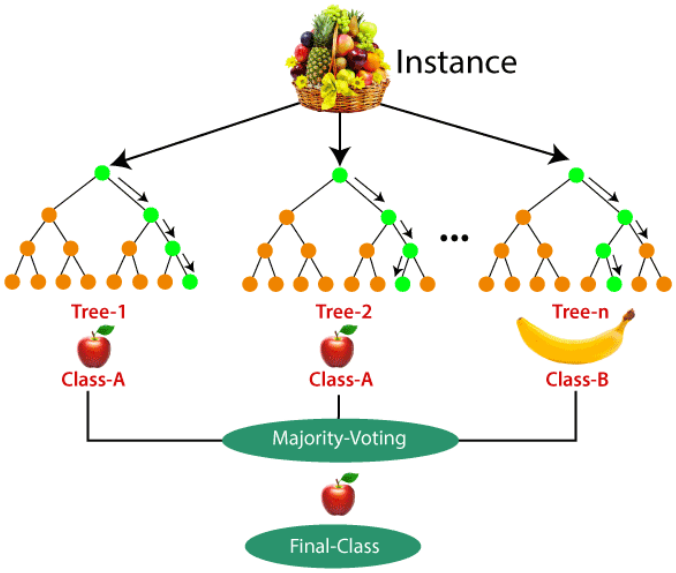


Figure 22: Example of random forest's voting

It's a type of assemble learning method which obtain better performances by using multiple learning algorithms (in this case decision trees). As stated above, the way all the information from the various trees is summarized is by listening to the majority. This voting systems allows the model to reduce its variability since it arrives to the result by several unrelated random paths. It also helps weed out the individual errors since it's highly unlikely two trees will make the same mistake. In case there's a tie, the random forest will break it randomly.

So, the random forest bases its philosophy on the wisdom of the majority and uses the collective to protect our predictions from singular trees that misbehave. For this reason, the trees inside the forest must have little correlation between one another so the model can achieve optimal performance.

---

[12] YIU Tony, *Understanding Random Forest*, 2019, [https://towardsdatascience.com/understanding-random-forest-58381e0602d2]

Another feature of the random forest is bootstrapping, a method it uses to sample the data. Basically, some samples will be used multiple times in a single tree. By doing so your tree will have a higher variance depending on the training set you give him but as a result the random forest variance will be lessened. Also for each tree the random forest selects a random number of variables to take into account. This also increases the variances of the trees.

Compared to decision tree, random forest provides better predictions and is able to handle missing data. It also has more power so it is able to treat bigger amounts of data. Finally, the voting system reduces greatly the chances of overfitting our model.

This model is used in a lot of different sectors both public and private. For example, it can use someone's medical records to identify their illness or help decide what products to recommend to a costumer based on their browser history.

Since a random forest contains several decision tree's it also allows you to modify its parameters, for example Max_Depth and Min_Sample_Split, but it also has its own parameters that have to be tuned for optimal results[13].

6.1. Parameters

6.1.1. N_Estimators

N_Estimators tells how many trees are in our forest. Its default is ten but in theory the more trees there are the better performance it gives since more votes are casted and less individual errors should pass. The main drawback of having too many estimators is that it slows the process since it has to run more every tree that it contains so to achieve an optimal result, one has to find the maximum value for each random forest.

6.1.2. Max_Features

Max number of features considered for splitting a node. The higher it is the more options the model will consider and highest the chance to get a better split but the less variability it will have. Considering that a random forest performance depends on its variability it's capital that one be careful not to choose a number too which would cause the trees to be too similar and add correlations between them, consequently rendering the vote biased.

[13] BEN FRAJ Mohtadi, *In Depth: Parameter tuning for Random Forest*, 2017, [https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d]

6.2. Parameter Tuning

With random forest, we follow a similar procedure than with Decision Tree. We are still looking to maximize the recall without letting the precision suffer so we're focusing on optimizing the F1. The train-test ratio is still 7-3 and stratified and we first consider the tendencies the trees parameters. For each Max_Depth and Min_Sample_Split we fix all parameters of the random forest in default (including Max_Depth when looking into Min_Sample_Split and vice versa) and vary their value to plot how they affect our model performances.

Once we've seen their tendencies we select the best values for each, which are those that achieve the highest F1. We evaluate our random forest for each combination of Max_Depth and Min_Sample_Split values leaving the other parameters in default and select the best pairs, those are the only ones we will take into account on the proceeding experiments.

Then we focus on the parameters used to define our random forest: MaxFeatures and N_Estimators. Like we've done for the parameters of the decision trees, we plot their tendencies leaving the other values in default.

We evaluate our random forest for each value of each parameter with each pair of max_depth and min_sample_split values and we obtain the best combination with both Max_Features and N_Estimators. Afterwards we evaluate the best Max_Features combinations with the best N_Estimators combinations to find the max F1 for our random forest.

The code used can be found in Annex 3.

6.3. Results

6.3.1. Max_Depth Tendency

As seen in Figure 23 Max_Depth seems to affect the random forest the same way it affected the decision trees on their own. The more we allow the trees to grow the better the recall and F1 are. Precision seems to achieve its optimal value faster and then get slightly worse probably due to overfitting. Finally, accuracy doesn't seem to vary too much regardless of the Max_Depth we select.

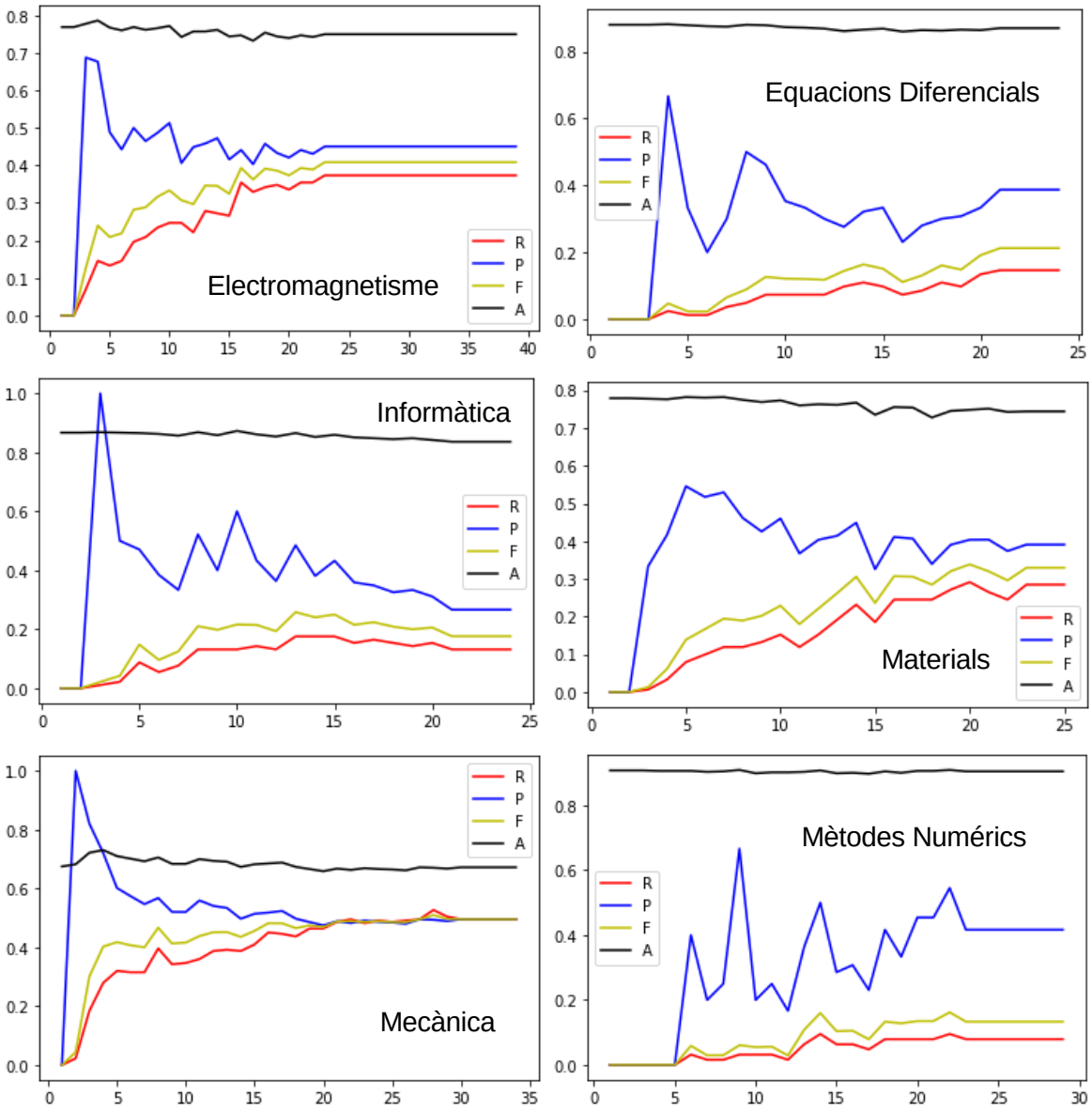


Figure 23: Max_Depth tendencies

Also like before, there seem to be groups of subjects that can achieve better predictions than the others. In this case ‘Mecànica’ and ‘Electromagnetisme’ seems to be the more easily predicted achieving up to around 40% in their F1 followed by ‘Materials’ which goes around a 30%. ‘Informàtica’ and ‘Equacions differencials’ both have an F1 of around 20% and ‘Mètodes Numèrics’ comes in last with around 10%.

In Figure 24 there are both the value to achieve the highest F1 and the group of values that while may not get a performance as high, still give us good results. Like before we’ve selected the lowest Max_Depth that complied with that since we want to avoid overfitting and consequently they are from before the model achieve stability and stops improving when Max_Depth grows.

SUBJECT	Best Value	Values to try
Electromagnetisme	23	16, 18, 21, 23
Equacions Diferencials	21	19, 20, 21, 22, 23, 24
Informàtica	13	13, 14, 15
Materials	20	19, 20, 21, 23
Mecànica	28	22, 24, 27, 28, 29
Mètodes Numèrics	22	14, 22

Figure 24: Best Values Max_Depth in Default

6.3.2. Min_Sample_Split Tendency

Since we've seen that Max_Depth in random forest follows the same tendencies that in Decision Tree it should not come as a surprise that Min_Sample_Split does the same. Which is to say that the more we allow our tree to branch (lower Min_Sample_Split values) the best performance it gives us as long as we don't end up overfitting our tree.

In Figure 25 we can observe that overfitting not only in precision but also in F1 and recall in the case of Informàtica and Mètodes Numèrics where you can see that at the beginning increasing the Min_Sample_Split improve our performance. In those cases a lower value causes the tree to be too defined and thus gives us a worse performance.

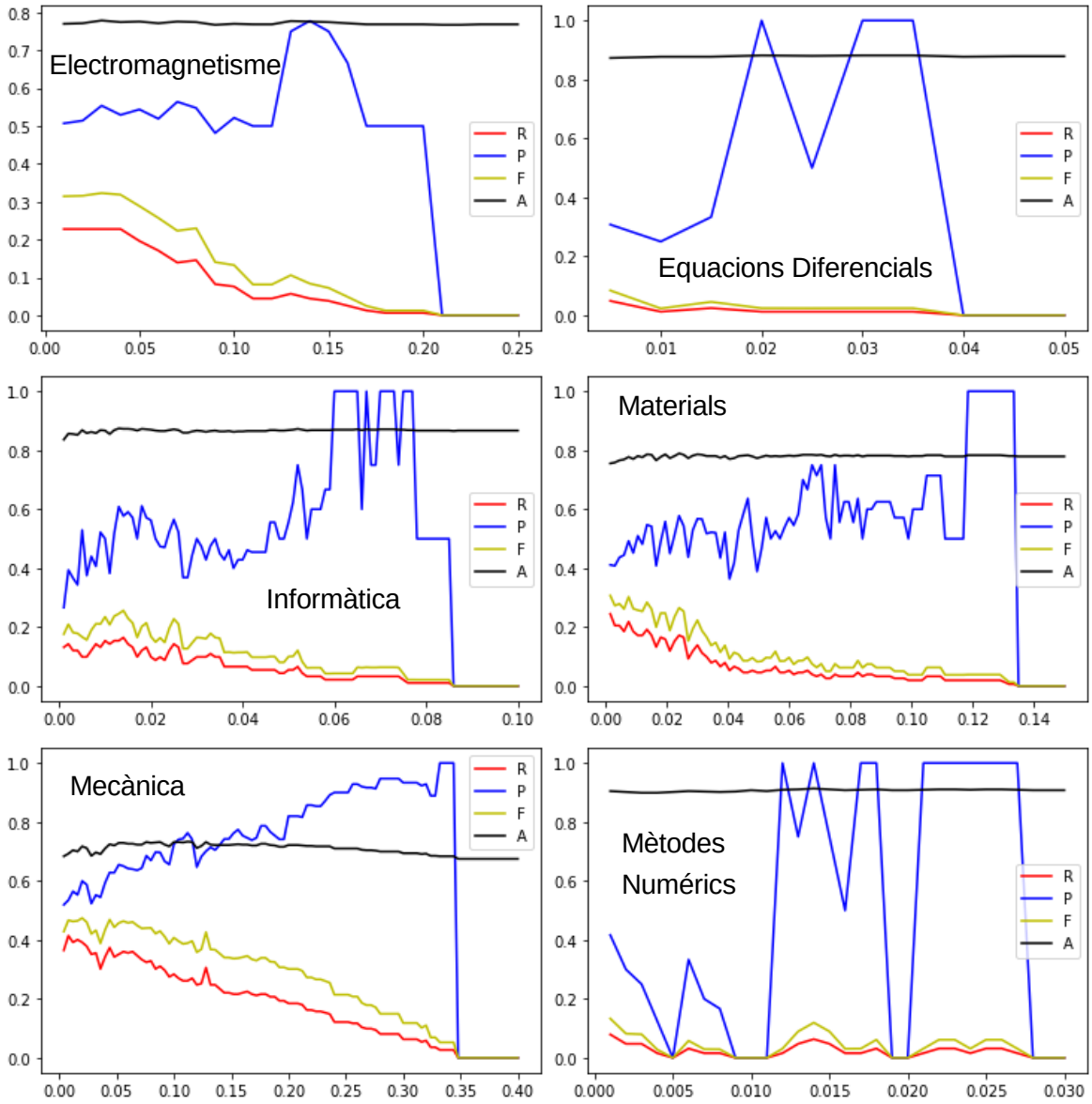


Figure 25: Min_Sample_Split tendencies

The subjects are separated in the same groups that we've seen in Max_Depth which indicates that it is not a coincidence. And the accuracy seems little effected by this parameter also.

Overfitting seems once again to affect precision over the other metrics, the best example being 'Mecànica' where instead of a tendency to decrease, it tends to increase right until it drops to zero when the Min_Sample_Split is too big for the model to split any node.

When selecting the values of Min_Sample_Split we try and get not only those with higher F1 but also those high enough to avoid as much overfitting as possible. The chosen ones can be seen in Figure 26.

SUBJECT	Best Value	Values to try
Electromagnetisme	0.03	0.01, 0.02, 0.03, 0.04
Equacions Diferencials	0.005	0.005, 0.006, 0.007
Informàtica	0.014	0.008, 0.01, 0.012, 0.014, 0.016
Materials	0.0015	0.0015, 0.0075, 0.0135
Mecànica	0.02	0.008, 0.02, 0.044
Mètodes Numèrics	0.001	0.001, 0.014

Figure 26: Best Values of Min_Sample_Split in Default

6.3.3. Decision Tree Tuning

Electromagnetisme

Figure 27 shows the best results obtained by combining diferent values of Max_Depth and Min_Sample_Split. Min_Sample_Split seem to be the more important parameter when influencing the F1 of our random forest since only half of the values tried managed to achieve this result while most Max_Depth do so when combined with them. The two values of Min_Sample_Split that remain are the highest and lowest of the interval chosen.

F1	Max_Depth	Min_Sample_Split
0.37	18, 21, 23	0.01
	21, 23	0.04

Figure 27: Electromagnetisme Evatuation with Decision Tree Parameters

Equacions Diferencials

While one of the F1 achieved is higher and the other, it's only so by a 2% so both results are shown here while the other combinations are discarded. Figure 28 shows that Max_Depth plays a bigger role in this subject since the F1 gotten with 0.05 Min_Sample_Split changes depending of it.

F1	Max_Depth	Min_Sample_Split
0.1	20	0.006
0.08	20, 24	0.005
	22	0.007

Figure 28: Equacions Diferencials Evaluation with Decision Tree Parameters

Informàtica

Similar to Electromagnetisme, Informàtica's F1 performance seems to depend more heavily in Min_Sample_Split than in Max_Depht since even when Max_Depth change the value of F1 (which happens at low Min_Sample_Split values) the difference is tiny.

In this case we'll also study a group of combinations that arrive at the highest precision even if it produces a slightly lower F1 than the others. The two F1 achieved along the combination of parameters used to get them can be seen in Figure 29.

F1	Max_Depth	Min_Sample_Split
0.26	13, 15	0.012
0.25	15	0.012
	14, 15	0.014

Figure 29: Informàtica's Evaluation with Decision Tree Parameters

Materials

In Materials' case, there is an F1 value higher than the others and two combinations that lead to it. These combinations have different Max_Depth and Min_Sample_Split. The Min_Sample_Split values seem to tend towards the lowest region of the parameter given while the Max_Depth values tend towards the higher one. The chosen values can be seen in Figure 30.

F1	Max_Depth	Min_Sample_Split
0.31	20	0.0075
	23	0.0015

Figure 30: Materials' Evaluation with Decision Tree Parameters

Mecànica

This case is the clearest example that Min_Sample_Split carries most of the weight when evaluating the F1 only using parameters of the Decision Tree. As shown in Figure 31 for any of the Max_Depth values chosen before the F1 will be the same provided we don't change the Min_Sample_Split. Also worth noting that the smaller the Min_Sample_Split the worse our model performs due to overfitting. However, the difference between F1 depending on the Min_Sample_Split is so minuscule that there has had to be a decimal added to appreciate it. So while Min_Sample_Split seems to be more important than Max_Depth it's a very relative importance.

F1	Max_Depth	Min_Sample_Split
0.474	Any of the evaluated	0.02
0.469	Any of the evaluated	0.044
0.467	Any of the evaluated	0.008

Figure 31: Mecànica's Evaluation with Decision Tree Parameters

Mètodes Numèrics

Mètodes numèrics has the best F1 performance with the Min_Sample_Split 0.001 as seen in Figure 32. This Min_Sample_Split can be paired with a couple of values of Max_Depht to achieve the F1 in question. It's interesting to notice that this subject has the worst recall which is of 9,5% and it drags the F1 down, future experiments should try to improve on it.

F1	Max_Depth	Min_Sample_Split
0.16	14	0.001
	22	0.001

Figure 32: Mètodes Numèrics' Evaluation with Decision Tree Parameters

6.3.4. Decision Tree parameters in Random Forest

Using random forest only tuning the parameters of decision tree (Max_Depth and Min_Sample_Split) and using random forest parameters in default shows improvement over decision tree on two subjects (Mecànica and Materials) and very little difference on three others (Electromagnetisme, Informàtica and Mètodes Numèrics). However, one subject performs significantly worse (Equacions Diferencials). This can be observed in Figure 33.

The results aren't regular enough among the subjects to get a clear conclusion. Furthermore, the improvement is too small and happens in too few subjects to justify the random forest model over the decision tree one. This may change once the other random forest parameters are tuned.

	F1, Decision Tree	F1 Random Forest Default
Electromagnetisme	0.4	0.37
Equacions Diferencials	0.21	0.1
Informàtica	0.29	0.26
Materials	0.3	0.31
Mecànica	0.39	0.47
Mètodes Numèrics	0.19	0.16

Figure 33: Comparation between Decision Tree and Random Forest only tuning Decision Tree Parameters

Max_Feature Tendency

Figure 34 shows us that the performance of our model has the tendency to very slightly increase along Max_Feature. We can also see various pics were the performance is significantly higher and that out of all the parameters this is the one that keeps all the metrics the closest together. This is especially clear in the case of Mecànica where the recall, precision and F1 have very similar values for each Max_Feature.

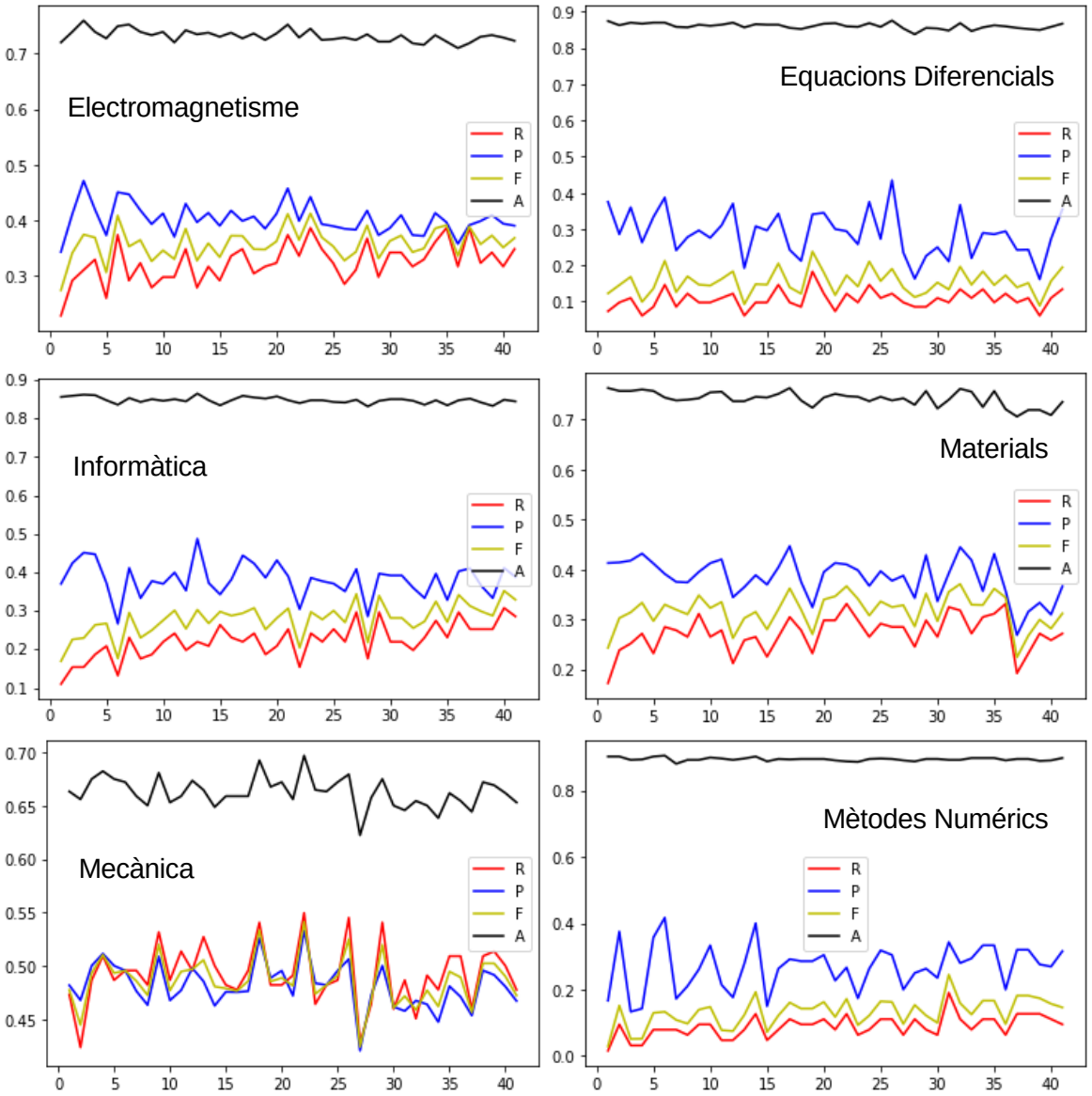


Figure 34: Max_Features tendencies

This is also the parameter that alters the accuracy the most but it still remains constant enough that we can claim that it does not significantly worsens depending on our choices. The best values and the ones we'll try on the combinations can be found in Figure 35.

SUBJECT	Best Value	Values to try
Electromagnetisme	23	6, 21, 23
Equacions Diferencials	19	15, 16, 17, 18, 19, 20, 21
Informàtica	40	27,36,40
Materials	32	22, 32, 35
Mecànica	22	18, 22, 26, 29
Mètodes Numèrics	31	14, 31, 37

Figure 35: Best Values of Max_Features in Default

6.3.5. N_Estimator Tendency

Figure 36 shows that in the very beginning the tendency of the F1 and recall is to increase alongside the N_Estimators but it quickly decreases. This surprised me since N_Estimator is the number of trees the random forest has and in theory the more trees the better performance. Accuracy and Precision do follow the tendency expected by improving the more trees they are given until they achieve a maximum value after which they stay stable. Since F1 is a balance between Recall and Precision, its odd behavior can be attributed to the Recall.

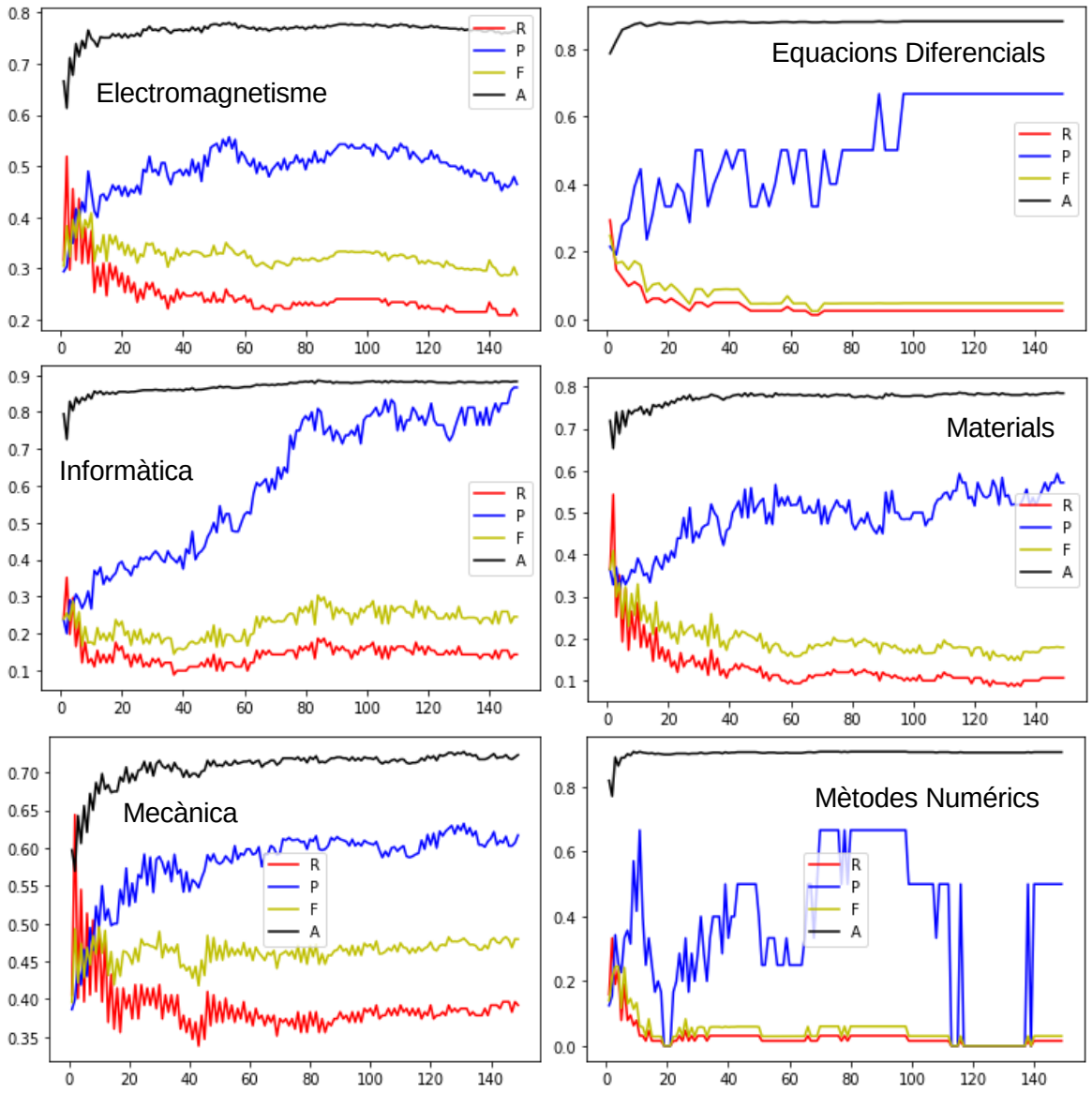


Figure 36: N_Estimators tendencies

In the cases of Informàtica, Electromagnetisme and Mecànica the F1 seems to behave as we would expect with the exception of a peak at the beginning. It's interesting to see that this contains the two subjects that are usually easier and a third that's usually in the middle group when classified by max F1. In fact, those three subjects arrive at higher F1 in the tendency graphics than the other three.

We choose the values with higher F1 trying to select those with more estimators since otherwise our Precision suffers heavily. Figure 37 shows those values.

SUBJECT	Best Value	Values to try
Electromagnetisme	6	6, 21, 23
Equacions Diferencials	2	1, 2, 3
Informàtica	84	82, 84, 86
Materials	2	1, 2, 4, 6, 10
Mecànica	10	8, 10, 12, 30, 46, 48
Mètodes Numèrics	4	3, 4, 6

Figure 37: Best Values of N_Estimators in Default

6.3.6. Evaluation

Electromagnetisme

When combining the Max_Depth and Min_Sample_Split pairs with the values of Max_Feature (Figure 38) that have been selected, the best value for Max_Features is 23 which allows us to achieve an F1 of 38%. Regardless of the value though, it seems that the combination of Min_Sample_Split and Max_Feature does not depend too much of the Max_Depth value when deciding the F1.

F1	Max_Depth	Min_Sample_Split	Max_Features
0.38	21, 23	0.04	23
0.37	18, 21, 23	0.01	6
	21, 23	0.04	

Figure 38: Combinations between Decision Tree parameters and Max_Feature values for Electromagnetisme

Max_Depth also doesn't seem to weight much when calculating F1 with Decision Tree parameters combined with N_Estimators. Figure 39 shows us that while 8 estimators seems to give a better result 6 estimators is a very close second so both will be combined with the Max_Feature value selected.

F1	Max_Depth	Min_Sample_Split	N_Estimators
0.39	21, 23	0.04	8
0.38	18, 21, 23	0.01	6

Figure 39: Combinations between Decision Tree parameters and N_Estimators values for Electromagnetisme

When combining the best values for all the parameters we're studying (Figure 40) we find that the one the parameters of Decision Tree seem to affect the F1 less than those only present in Random Forest. It's also interesting that the best value of F1 (40%) seem to be when N_Estimators and Max_Features are highest.

F1	Max_Depth	Min_Sample_Split	Max_Features	N_Estimators
0.4	21, 23	0.04	23	8
0.39	21, 23	0.04	6	8
0.38	18, 21, 23	0.01	6	6

Figure 40: Combinations of all the parameters for Electromagnetisme

Equacions Diferencials

As seen in Figure 41 and Figure 42 there is one value for both N_Estimators and Max_Features that maximizes F1 and out of the two parameters of the Decision Trees, Min_Sample_Split seems to still be the most important.

F1	Max_Depth	Min_Sample_Split	N_Estimators
0.24	20	0.006	1
0.23	22, 24	0.005	1

Figure 41: Combinations between Decision Tree parameters and N_Estimators values for Equacions Diferencials

F1	Max_Depth	Min_Sample_Split	Max_Features
0.19	20	0.007	21
0.14	22, 24	0.005	21

Figure 42: Combinations between Decision Tree parameters and Max_Features values for Equacions Diferencials

Curiously when combining all the parameters a Min_Sample_Split that gave a worse F1 when one of the parameters was on default is now the one giving the best F1. With this we can prove the hypothesis that the combination of the best values of each parameter when the rest are in default doesn't have to be the best one. Figure 43 shows us the results.

F1	Max_Depth	Min_Sample_Split	Max_Features	N_Estimators
0.25	22, 24	0.005	21	1
0.24	20	0.007	21	1

Figure 43: Combinations of all the parameters for Equacions Diferencials

Informàtica

Informàtica is an interesting case since it has one parameter (N_Estimators) that will it doesn't improve the F1 too much it does help achieve much higher precision. In Figure 44 the Max_Features are being tuned and because of that the N_Estimators is left in default the value of which is 10. In those cases, the precision barely raises to the 55%. In Figure 45 however, the default parameter is Max_Feature and we can see that varying the N_Estimators we can easily achieve up to 85% precision.

This can be explained by the fact that in this particular subject the N_Estimators values chosen are the highest out of all the cases. As seen before, Precision is the parameter that most improve the more trees you allow in the random forest.

F1	P	Max_Depth	Min_Sample_Split	Max_Features
0.29	0.45	13	0.012	36
0.28	0.55	15	0.014	27
0.28	0.53	14	0.014	27

Figure 44: Combinations between Decision Tree parameters and Max_Features values for Informàtica

F1	P	Max_Depth	Min_Sample_Split	N_Estimators
0.22	0.71	15	0.012	84
0.22	0.67	15	0.012	86
0.21	0.85	14	0.014	82,84

Figure 45: Combinations between Decision Tree parameters and N_Estimators values for Informàtica

When we combine all the parameters as the Figure 46 shows, the best F1 obtained has the same value as the best F1 found with N_Estimators in default when tuning Max_Depth. However even if both have the same F1, when we consider the N_Estimators we manage to double our Precision. This improvement comes with the cost of a worse Recall though so it will come down to a decision of whether we consider that having a lower chance to wrongly classifying some students as likely to fail when in reality they will pass is worth having a lower chance to correctly classify which students will fail.

F1	P	Max_Depth	Min_Sample_Split	Max_Features	N_Estimators
0.29	0.61	13	0.014	36	86
0.28	0.64	13	0.014	36	82
0.27	0.59	15	0.012	36	86
		14	0.014	36	82,84

Figure 46: Combinations of all the parameters for Informàtica

Materials

Unlike most of the cases already studied when it comes to Materials it seems that Min_Sample_Split plays a lesser role in determining the F1 than the combination between Max_Depth and Min_Sample_Split. Figure 47 shows that out of all the combinations that have been tried the best results by a large margin are both due to the same Max_Depth and Min_Sample_Split pair.

F1	Max_Depth	Min_Sample_Split	Max_Features
0.36	23	0.0015	22
0.34	23	0.0015	35
0.30	20	0.0075	22

Figure 47: Combinations between Decision Tree parameters and Max_Features values for Materials

Figure 48 shows how important the N_Estimators values is since it being optimal seems to be more important than the pair Max_Depth and Min_Sample_Split. The same pair of decision tree parameters give a far better F1 when combined with 2 N_Estimators than with 6 and another pair combined with 2 give almost the same value of F1. In Figure 49 can be observed that that is also the case for the value of Max_Feature which seems to hold less weigh in the result of F1 than N_Estimators does.

F1	Max_Depth	Min_Sample_Split	N_Estimators
0.39	23	0.0015	2
0.38	20	0.0075	2
0.37	23	0.0015	6

Figure 48: Combinations between Decision Tree parameters and N_Estimators values for Materials

F1	Max_Depth	Min_Sample_Split	Max_Features	N_Estimators
0.42	23	0. 0015	35	2
0.41	23	0.0015	22	2
0.4	20	0.0075	35	2

Figure 49: Combinations of all the parameters for Materials

Mecànica

Out of all the subjects, Mecànica is the one that most clearly shows that Max_Depht is the parameter with less weight when deciding the F1. Figure 50 shows that each combination of Min_Sample_Split and N_Estimators gives the same F1 regardless of the value attributed to Max_Depth. It also shows that out of all the values of N_Estimators that has been tried 46 and 48 achieve the best results when paired with any Min_Sample_Split. Figure 51 shows the same behavior when paired with Max_Features.

F1	Max_Depth	Min_Sample_Split	N_Estimators
0.49	Any of the evaluated	0.008	46, 48
0.48	Any of the evaluated	0.02	8, 46
0.47	Any of the evaluated	0.02	10

Figure 50: Combinations between Decision Tree parameters and N_Estimators values for Mecànica

F1	Max_Depth	Min_Sample_Split	Max_Features
0.5	Any of the evaluated	0.008, 0.02	29
0.49	Any of the evaluated	0.02	22
0.48	Any of the evaluated	0.044, 0.02	26

Figure 51: Combinations between Decision Tree parameters and Max_Features

values for Mecànica

An interesting tendency can be seen in Figure 52. When paired with low Min_Sample_Split, while higher Max_Depth values don't seem to affect our result the lowest value tried does. This is explained with overfitting. As stated before overfitting happens with low values of Min_Sample_Split and high values of Max_Depth. Then it follows that when selecting low Min_Sample_Split we are risking our model overfitting so lower values of Max_Depth will help avoid it and thus perform better.

F1	Max_Depth	Min_Sample_Split	N_Estimators	Max_Features
0.5	22	0.008	46	29
0.49	Any of the evaluated	0.008	48	29
0.49	24,27,28,29	0.008	46	29
0.49	Any of the evaluated	0.02	48	29
0.48	Any of the evaluated	0.02	46	29

Figure 52: Combinations of all the parameters for Mecànica

Mètodes Numèrics

Studying the results of Mètodes Numèrics we see that even if one value of Max_Features is better when leaving N_Estimators in default (Figure 53) that doesn't mean that will be the best value when we combine all the parameters (Figure 55).

F1	Max_Depth	Min_Sample_Split	Max_Features
0.24	14, 22	0.001	31

Figure 53: Combinations between Decision Tree parameters and Max_Features values for Mètodes Numèrics

Likewise, even if 4 is the best N_Estimator when Max_Features is in default (Figure 54), when Max_Features is also given a value (Figure 55) it performs far better when the value for N_Estimators being 6.

Random Forest is a complex model whose parameters are all interconnected and the optimal number of trees will differ according to how much we let those trees grow, how many nodes are allowed to split and how many questions we ask before splitting a node. That's why finding the best values for each parameters when the rest are left in default doesn't guarantee that a combination of those values will give the best result and it's necessary to try different combinations in order to best tune the random forest.

F1	Max_Depth	Min_Sample_Split	N_Estimators
0.27	22	0.001	3
0.26	22	0.001	4
0.26	22	0.001	6

Figure 54: Combinations between Decision Tree parameters and N_Estimators values for Mètodes Numèrics

F1	Max_Depth	Min_Sample_Split	Max_Features	N_Estimators
0.28	14	0.001	37	6
0.27	14	0.001	37	4
0.25	22	0. 001	31	4

Figure 55: Combinations of all the parameters for Mètodes Numèrics

7. Comparation between Decision Tree and Random Forest

Our objective was to maximize the F1 of our model in order to get a good Recall without sacrificing the Precision. While we weren't focused on Accuracy we recorded its values to ensure it didn't drop to much when trying to achieve our objective. The results obtained using decision tree and random forest can be seen in Figure 56.

	% 'Fail'	DECISION TREE				RANDOM FOREST			
		A	F1	P	R	A	F1	P	R
Electromagnetisme	23,2	0.79	0.4	0.58	0.3	0.76	0.4	0.48	0.34
Equacions Diferencials	12	0.88	0.21	0.37	0.14	0.79	0.25	0.22	0.29
Informàtica	13,3	0.86	0.3	0.39	0.24	0.88	0.29	0.61	0.19
Materials	22,1	0.75	0.3	0.41	0.24	0.67	0.42	0.34	0.55
Mecànica	32,6	0.65	0.39	0.46	0.35	0.72	0.5	0.60	0.42
Mètodes Numèrics	9,3	0.87	0.19	0.21	0.17	0.9	0.28	0.38	0.22

Figure 56: Results of Decision Tree and Random Forest.

Out of all the subjects the only one which F1 dropped slightly after the random forest was tuned is Informàtica where we decided to sacrifice a small percentage of it (0.5% compared with the decision tree) to almost double the Precision. Consequently, our Recall dropped a 5%. The rest of the subjects improved their F1 up to a 12% (Materials) when using the random forest model.

Mecànica and Mètodes Numèrics are the two subjects that benefited the most out of using Random Forest since they not only improved their F1 but also their Precision, Recall and Accuracy.

Electromagnetimse, Equacions Diferencials and Materials however had to sacrifice a bit of Precision in order to improve their Recall. Since their F1 improved though, the gain of Recall is more significative than the loss of Precision.

The more even ratio of 'fail' and 'pass' marks the easier our model can make predictions. Mecànica having a 30% of failing marks manage to achieve an F1 of 50% while Electromagnetisme and Materials which 20% of the students fail only arrive at 40% F1. Finally, the three subjects that only 10% of people fail, stop improving at around 20% F1.

The accuracy improves in some cases and drops in others, but the changes are not significative enough to claim that by improving the F1 we are sacrificing it.

Figure 57 shows that the trees in random forest tend to grow more both in depth and breadth. There's also a need for a high number of features to make the best decisions. The number of trees for each forest is irregular but the most interesting cases are Equacions Diferencials and Materials which have the fewer trees. Equacions Diferencials is not truly a forest but a single decision tree where the Max_Feature value has been adjusted too. Materials are only two trees and when they disagree the tie is broken randomly.

	%	DECISION TREE		RANDOM FOREST			
	'Fail'	MD	MS	MD	MS	MF	NE
Electromagnetisme	23,2	12	0.08	21, 23	0.04	23	8
Equacions Diferencials	12	10	0.04	22, 24	0.005	21	1
Informàtica	13,3	20	0.08	13	0.014	36	86
Materials	22,1	15	0.06	23	0. 0015	35	2
Mecànica	32,6	19	0.09	22	0.008	29	46
Mètodes Numèrics	9,3	13	0.03	14	0.001	37	6

Figure 57: Results of Decision Tree and Random Forest.

An ideal random forest is one with a high number of tree's in order to reduce the instability of the individual trees and with a small number of features in order to have as much variability within those trees as possible. As seen above, in these experiments we have achieved the opposite result.

This means the forest will have a small number of trees and that those trees will be similar to one another. Consequently, the model used takes little advantage to the strong points of random forest. This may be why the performance improvement is not as drastic as expected in most subjects.

Conclusions

This project's objectives were to achieve the best performance possible using the models decision tree and random forest, as well as to compare the results obtained by each. To do so an evaluation metric had to be chosen and the models' parameters had to be tuned. The project has followed the methodology CRISP-DM has been used.

To obtain the best parameters for each model, those parameters have been explored leaving the rest in default to observe the tendency of the evaluation metrics when they change. In the case of the decision tree model, the comparation between the different groups of parameters has been done using 'Gridsearch' in order to accomplish the best results possible.

First, it has been decided that F1 is the best evaluation metric when looking to correctly classify future data since it allows to correctly guess as many failing marks as possible (high Recall) without mistakenly predicting too many passing marks as failing ones (high precision). It has also been observed that out of all the evaluation metrics used, Precision is the one more susceptible to overfitting.

It has been proven that the best decision trees are those that we allow to grow as much as possible (high Max_Depth and low Min_Sample_Split). However, we must be careful we don't let the tree grow too much or it will overfit and consequently perform worse.

Also, only modifying the parameters of decision tree in a random forest does not improve the results, it is necessary to tune the other parameters too.

The results also show that the random forest model allows for better predictions that the decision tree model, especially when it comes to Recall. While the decision tree parameters found in random forest (Min_Sample_Split and Max_Depth) are important to tune correctly, the main weight of the predictions falls into the parameters more typical from the random forest (N_Estimator and Max_Features).

The best random forests gotten are those with depth trees and small nodes which are allowed to use a high number of features when making a decision. Regarding the number of trees in the forest it depends on the subject but considering that in some the number is very small random forest, whose main advantage is that it allows for a big number of them, may not be the best model to make predict this data. The same can be said about the high number of features needed for every forest which makes the trees in them too similar to each other and thus worsen our results. This contradictions between the theory and the practical results can be due to the limitation of the decision trees to predict this data.

The subjects that have a more regular distribution of failing and passing marks are easier for our model to predict. However, our best F1 is 50% and at most the random forest has gained a 10% compared to the tree classifier. Because of this, this model probably wouldn't be useful as it stands since too many struggling students would be left out.

In order to get better predictions, one could attempt to balance the number of failing and passing grades by oversampling (adding artificial data) or try other prediction models that could be a better fit.

Acknowledgements

The circumstances we found ourselves into, increased the challenge of realizing this project which is why I would like to thank several people that have helped me overcome the hurdles the current pandemic has placed on my way.

First, I'd like to thank my project director, Lluís Talavera, for taking the time to coach me and meet with me both in real live and online during a time where he was particularly busy with rearranging the subject he teaches and manning other projects.

I would also like to thank my family for supporting me and encouraging me, especially in this last few weeks when we have been kept inside, in close proximity from each other every hour of every day.

Out of all the members of my family, a special thank goes to my sisters. One of which agreed to read my project in order to check whether it was compressible for people who did not know anything about the subject and the other which listened to me talk about the code, helping me organize my ideas.

Bibliography

[1] OLSON David and DELEN Dursun, *Advanced Data Mining Techniques*, 2008

[2] GURU99, *Supervised Vs Unsupervised Learning: Key Differences*, 2020, [https://www.guru99.com/supervised-vs-unsupervised-learning.html]

RECUERO DE LOS SANTOS Paloma, *Tipos de aprendizaje en Machine Learning: supervisado y no supervisado*, 2017, [https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje]

[3] OLSON David and DELEN Dursun, *Advanced Data Mining Techniques*, 2008

RODRIGUES Israel, *CRISP-DM methodology leader in data mining and big data*, 2020 [https://towardsdatascience.com/crisp-dm-methodology-leader-in-data-mining-and-big-data-467efd3d3781]

[4] MOFFITT Chris, *Pandas Pivot Table Explained*, 2014, [https://pbpython.com/pandas-pivot-table-explained.html]

[5] ALLIBHAI Eijaz, *Hold-out vs. Cross-validation in Machine Learning*, 2018, [https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f]

[6] SCIKIT-LEARN, *sklearn.model_selection.train_test_split*, 2019, [https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html]

[7] DRAKOS George, *How to select the right evaluation metric for machine learning models*, 2019 [https://medium.com/@george.drakos62/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-3-classification-3eac420ec991]

PING SHUNG Koo, *Accuracy, Precision, Recall or F1?*, 2018, [https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9]

[8] GUPTA Prashant, *Decision Trees in Machine Learning*, 2017, [https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052]

[9] ELITEDATASCIENCE, *Overfitting in machine learning*, 2019, [https://elitedatascience.com/overfitting-in-machine-learning]

[10] BEN FRAJ Mohtadi, *InDepth: Parameter tuning for Decision Tree*, 2017, [https://medium.com/@mohtedibf/indepth-parameter-tuning-for-decision-tree-

6753118a03c3]

[11] SCIKIT-LEARN, *sklearn.model_selection.GridSearchCV*, 2019 [https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html]

[12] YIU Tony, *Understanding Random Forest*, 2019, [https://towardsdatascience.com/understanding-random-forest-58381e0602d2]

[13] BEN FRAJ Mohtadi, *In Depth: Parameter tuning for Random Forest*, 2017, [https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d]