UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI **MATEMATICA E INFORMATICA**

Criminals' recidivity

A Machine Learning model suitable for predicting cases of criminal recidivism

Student	Student number
Giorgio Andronico	227815
Claudio Lucisano	231871
Chiara Passarelli	223971

Contents

1 Introduction

2 Business understanding

2.1 Background

This project focuses on analyzing and predicting the COMPAS recidivism dataset, that is a dataset made by the use of the Correctional Offender Management Profiling for Alternative Sanctions, a decision support tool developed by Equivant. This very dataset was utilized in a research of ProPublica, who did an investigation to evaluate the recidivity scores given by the COMPAS algorithm. It was made taking in account criminal histories from the Broward County; it contains information about those criminal records, such as the committed crime, the ammount of juvenile crimes or the COMPAS risk scores. This one is a peculiar attribute that can help to determine if a person is more likely to commit another crime in the near future. Specifically, the ProPublica study focused on finding a racial bias in how the scores were given by the COMPAS algorithm.

2.2 Business objectives

The main objective of this study is to create a Machine Learning model capable to analyze and predict whether a given person is likely to commit another crime after the one that was screened for.

2.3 Inventory of resources

In this project we've used the following tools:

Python is an high level programming language, primarily used for scripting, given mostly its readability; alongside a functional programming approach, it can be used with an object oriented one and it's largely used to manage and analyze data thanks to its large number of modules.

Pandas is one of the best Python library to manipulate data. It is a free-software that offers data structures and functions to operate on tables and datasets.

Jupyter, or better the Project Jupyter, is an open-source software for interactive computing across various languages, most importantly Python.

Figure 1: Python logo

Figure 2: Pandas logo

Figure 3: Jupyter logo

Figure 4: Colab logo

Colab, on the other end, is a free Jupyter notebook enviroment developed by Google that permits to access a said notebook enviroment via cloud.

Scikit is a Python library that provides multiple classification, regression and clustering algorithms, leaning on the **NumPy** library.

Matplotlib is a Python plot library based on the mathematics extension **NumPy** used to create mathematical plot, used also to provide an integration for other GUI toolkit.

Seaborn is a Python plot library based on the **Matplotlib**, which we used mainly to make prettier plots.

2.4 Data Mining goals

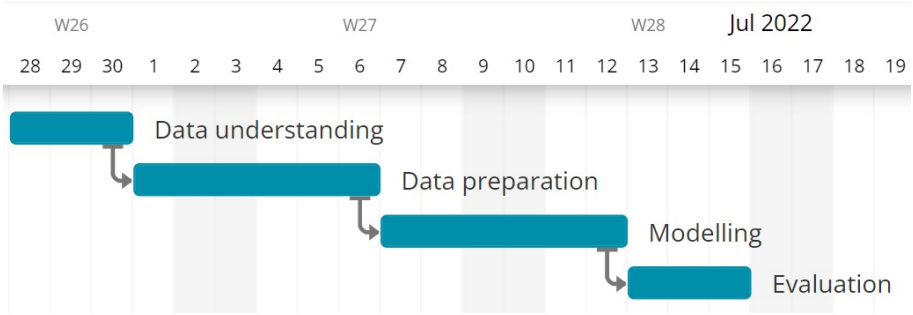
The goal of this project is to successfully train a model to recognize when a criminal will likely reoffend or not. That is, given new unseen data of a criminal as input, the model should be able to create as output a reasonably accurate prediction on whether that person will reoffend or not.

2.5 Success Criteria

Our success criteria is to reach an accuracy value not lower than 65% and F-measure not lower than 60%.

2.6 Project plan

Below there are the Gantt diagrams and task schedule.



Search tasks...	START	DUE	%
Data understanding	28/Jun	30/Jun	100%
Data preparation	01/Jul	06/Jul	100%
Modelling	07/Jul	12/Jul	100%
Evaluation	13/Jul	15/Jul	100%

3 Data understanding

3.1 Initial Data Collection Report

The dataset is taken from ProPublica, a non-profit organization which set to analyze the correctness of automated computer tools used in the U.S. justice system. It was sourced from the Sheriff's Office of Broward County, FL, USA. In particular, they obtained two years' worth of scores and crime data for all the people screened in 2013 and 2014. COMPAS is a computer software that takes in account many parameters of the criminal, and outputs a decile score which details how likely is a criminal to reoffend. 1: very low probability, 9: very high probability. Broward County primarily uses the score to determine whether to release or detain a defendant before his or her trial. Thus, ProPublica discarded scores that were assessed at parole, probation or other stages in the criminal justice system. That left 11,757 people who were assessed at the pretrial stage. We will detail this further in the next sections, but in general each row of the dataset contains information about both the crime and the criminal. Information about crime are time spent in jail, degree, case number and such. Information about criminals are race, sex, age, name, juvenile felony count etc. Specifically, the **is-recid** column is fundamental. This details if the criminal detailed on a certain row actually has reoffended or not.

3.2 Data Description

The selected dataset is composed by 11757 rows (or instances) and 47 attributes, for a total of 552579 entries. Here we are going to explain each attribute, its type and its meaning.

Column	Type	Description	Null percentage
id	Numerical	Unique identifier of the criminal case	0%
name	Categorical	Full name of the outlaw	0%
first	Categorical	First name of the outlaw	0%
last	Categorical	Last name of the outlaw	0%
compas_screening_date	Date	Date of the screening for the current case	0%
sex	Categorical	Sex of the outlaw	0%
dob	Date	Date Of Birth of the outlaw	0%
age	Numerical	Numeric age of the outlaw	0%
age_cat	Categorical	Categorization of the outlaw's age	0%
race	Categorical	Race of the outlaw	0%
juv_fel_count	Numerical	Amount of juvenile fellony committed by the current criminal	0%
decile_score	Numerical	1 to 10 number, assigned according to the COMPAS score, that determines the likelihood of recidivism by the current criminal	0%
juv_misd_count	Numerical	Amount of juvenile misdemeanors committed by the current criminal	0%
juv_other_count	Numerical	Amount of other type of juvenile crimes committed by the current criminal	0%
priors_count	Numerical	Amount of criminal records	0%
days_b_screening_arrest	Numerical	Number of days between arrest and screening	10.037%
c_jail_in	Date	Date of entry into jail	10.037%
c_jail_out	Date	Date of release from jail	10.037%
c_case_number	Categorical	Case identification number	6.311%
c_offense_date	Date	Date of the committed crime	22.114%
c_arrest_date	Date	Date of the arrest	84.197%

c_days_from_compas	Numerical	Number of days between COMPAS screening and sentencing	6.311%
c_charge_degree	Categorical	Type of crime committed. It goes from F for fellony, M for misdemeanor and O for other	0%
c_charge_desc	Categorical	More accurate description of the crime committed	6.371%
is_recid	Numerical	Number representing if the criminal has committed a crime again. 0 means that he did not re-committed a crime, 1 if he did. -1 is used as an indicative of unknown	0%
num_r_cases	Numerical	Amount of recidive cases	100%
r_case_number	Categorical	Case number of the recidive case	68.504%
r_charge_degree	Categorical	Type of the recidive case. It goes from F for fellony, M for misdemeanor and O for other	0%
r_days_from_arrest	Numerical	Number of days between the second arrest and the previous one	79.076%
r_offense_date	Date	Date of the committed recidive case	68.504%
r_charge_desc	Categorical	More accurate description of the committed recidive case	69.014%
r_jail_in	Date	Date of entry into jail, in regard of the recidive case	79.076%
r_jail_out	Date	Date of release from jail, in regard of the recidive case	79.076%
is_violent_recid	Numerical	Binary representation of whether the crime was violent or not. 0 represents it was not violent, 1 if it was	0%

num_vr_cases	Numerical	Amount of violent recidive cases	100%
vr_case_number	Categorical	Case number of the violent recidive case	92.498%
vr_charge_degree	Categorical	Type of the violent recidive case committed	92.498%
vr_offense_date	Date	Date of the recidivist violent crime that occurred	92.498%
vr_charge_desc	Categorical	More accurate description of the recidivist violent crime committed	92.498%
v_type_of_assessment	Categorical	Type of risk attribuitable to the criminal given by COMPAS, regarding the violent case	0%
v_decile_score	Numerical	1 to 10 number, assigned according to the COMPAS score, that determines the likelihood of violent recidivism by the current criminal	0%
v_score_text	Categorical	Textual classification of the probability that the criminal is violent. If the v_decile_score is between 1-4 is classified as "Low", 5-7 as "Medium", 8-10 as "High"	0.043%
v_screening_date	Date	Date of the screening of the violent crime	0%
type_of_assessment	Categorical	Type of risk attribuitable to the criminal given by COMPAS, regarding the recidivist case	0%

decile_score.1	Numerical	1 to 10 number, assigned according to the COMPAS score, that determines the likelihood of recidivism by the current criminal. Redundant copy of the decile_score attribute	0%
score_text	Categorical	Textual classification of the COMPAS score. If the decile_score is between 1-4 is classified as "Low", 5-7 as "Medium", 8-10 as "High"	0.128%

Table 1: Description of all the attributes of the dataset

As can be seen above, for each row there can be either one or two crimes detailed. In particular, for certain rows the columns beginning with $v\_$, $vr\_$ or $r\_$ are not null. In that case, those columns are detailing the later, second crime committed by that same defendant.

3.3 Data Exploration

In this phase of the study we've concentrated on understanding in more detail every attribute. So we analyzed which attribute could be more valuable to take in consideration and which is possible to discard, due to too many *null* values or simply for lack of analytical relevance. Later, we focused on separating numerical attributes, categorical ones and date ones into individual sets, as you can see in the table 1.

3.3.1 Statistics

We, then, found out the statistical values of each numerical column, as it's visible in the following table.

	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
id	11757.0	5879.0	3394.098	1.0	2940.0	5879.0	8818.0	11757.0
age	11757.0	35.143	12.023	18.0	25.0	32.0	43.0	96.0
juv_fel_count	11757.0	0.062	0.445	0.0	0.0	0.0	0.0	20.0
decile_score	11757.0	4.371	2.878	-1.0	2.0	4.0	7.0	10.0
juv_misd_count	11757.0	0.076	0.450	0.0	0.0	0.0	0.0	13.0
juv_other_count	11757.0	0.094	0.472	0.0	0.0	0.0	0.0	17.0
priors_count	11757.0	3.082	4.687	0.0	0.0	1.0	4.0	43.0
days_b_screening_arrest	10577.0	-0.878	72.889	-597.0	-1.0	-1.0	-1.0	1057.0
c_days_from_compas	11015.0	63.588	341.900	0.0	1.0	1.0	2.0	9485.0
is_recid	11757.0	0.254	0.558	-1.0	0.0	0.0	1.0	1.0
num_r_cases	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
r_days_from_arrest	2460.0	20.411	74.355	-1.0	0.0	0.0	1.0	993.0
is_violent_recid	11757.0	0.075	0.263	0.0	0.0	0.0	0.0	1.0
num_vr_cases	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
v_decile_score	11757.0	3.571	2.500	-1.0	1.0	3.0	5.0	10.0
decile_score.1	11757.0	4.371	2.878	-1.0	2.0	4.0	7.0	10.0

Table 2: Main statistical values of the numerical attributes

3.3.2 Numerical Histograms

As the table of the numerical attributes shows, there are a lot of attributes that have a somewhat skewed distribuition of values. This is much more clear when looking at the histograms. Also, some values such as *id*, *c_days_from_compas*, *r_days_from_arrest* have a high standard deviation, a clear indication that these values are not very representative for the modeling phase. Columns such as *num_r_cases* or *num_vr_cases* are completely null. We take note of all of this and plan to remove the columns in the Data Preparation phase. Instead, some columns are showing low standard deviation, and these could be very useful for the model. To get a better look at the distribution of such columns, we tabulated values and their relative occurrences. In particular, we did it for *juv_misd_count*, *juv_fel_count*, *priors_count*, and the results can be seen in table 3.

value	juv_misd_count	juv_fel_count	priors_count
0	0.953	0.964	0.356
1	0.033	0.024	0.187
2	0.009	0.007	0.109
3	0.003	0.003	0.073
4	0.001	0.001	0.050

Table 3: Table detailing values of the columns *juv_misd_count*, *juv_fel_count*, *priors_count*

We chose only some of the columns for our histograms and exclude others. In particular, of the numeric columns, we left out some of those with no statistical significance such *id* (which is a column with no logical relation to the others and usually not really useful for modeling, which is our ultimate goal), those with too little values such as *is_recid*, duplicates such as *decile_score.1*, and columns with very high null percentages such as *num_r_cases*. To have a better view of the data, the attributes that exceed the 60% of *null* values were not rendered in the histograms. Refer to table 1 for more details.

Commenting on the histograms, we can see that the columns *days_b_screening_arrest*, *c_days_from_compas*, and the *juv_* columns are strongly centered around one value. Meaning that most people did not com-

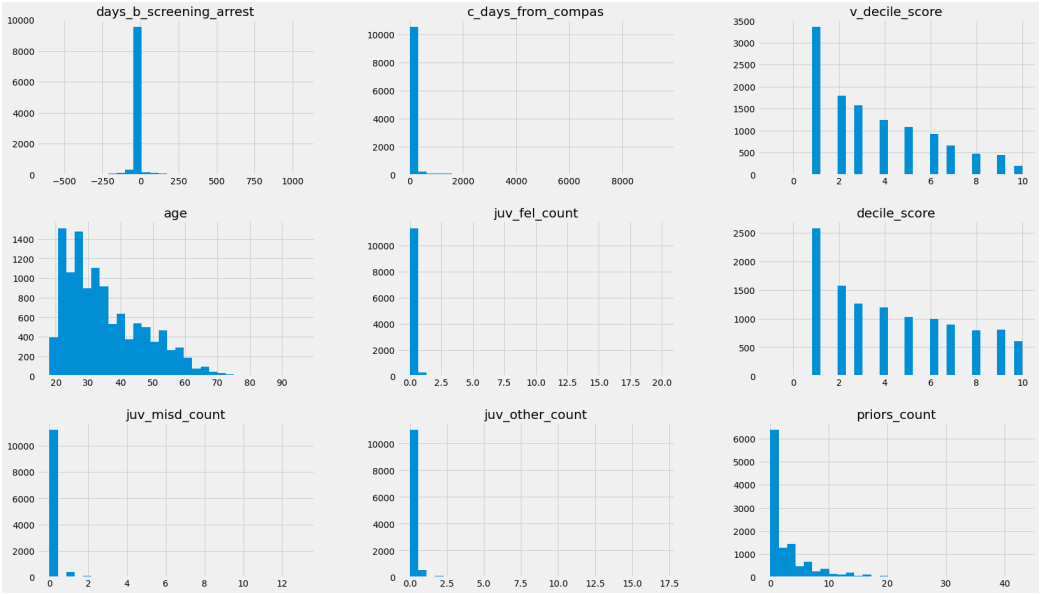


Figure 5: Histograms of the numeric attributes

mit felonies or other crimes when they were underage, and for most people not many days pass between screening and arrest.

Instead, for the *decile_score* columns, as the distribution decreases towards the right, we can infer that most people are deemed at a low risk of recidivism, while people with high scores generally represent the minority. Specifically, people with a score of 10 do not surpass the 500 mark in the case of *decile_score* and the 100 mark in the case of *v_decile_score*. Which, considering we have upwards of 10000 instances, is really a very small minority.

priors_count has a left-tailed distribution. This means that a great majority (upwards of 6000) people has not committed prior crimes, and again considering we have about 10000 cases, we can say that a very large majority of people have no priors. *age* is slightly skewed towards the left, but that is expected. Indeed, it is way less likely to find criminals of old age.
The boxplot just confirms what we have seen with the histograms. For example, *days_b_screening_arrest* has high dispersion as already seen, while *priors_count* has a very strong centering towards zero.

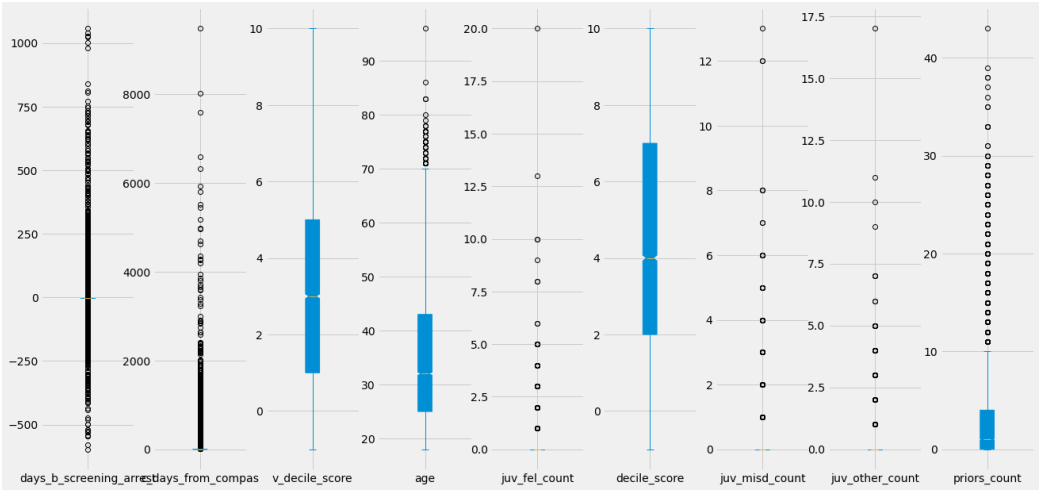


Figure 6: Boxplots of numerical attributes

3.3.3 Categorical histograms

These histograms are certainly less insightful than the numerical ones, but still useful to glance at the data and infer trends. Only some of these are reported here for brevity. Looking at the figure, we can see that the vast majority of defendants are male. Also, most people commit crimes in the 25-45 age range. African americans are the most registered race committing crimes, but caucasians are not far off (there's a delta of about 1500 occurrences). Most crimes are of the felony type, and half of that are murder types. Most recidive crimes are of type other. And generally, most people are valued at a low risk of recidivism, both in violent and non violent cases.

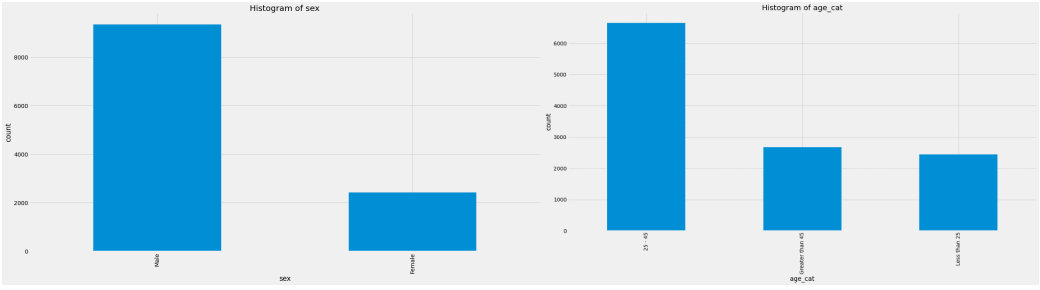


Figure 7: *sex* and *age* histograms

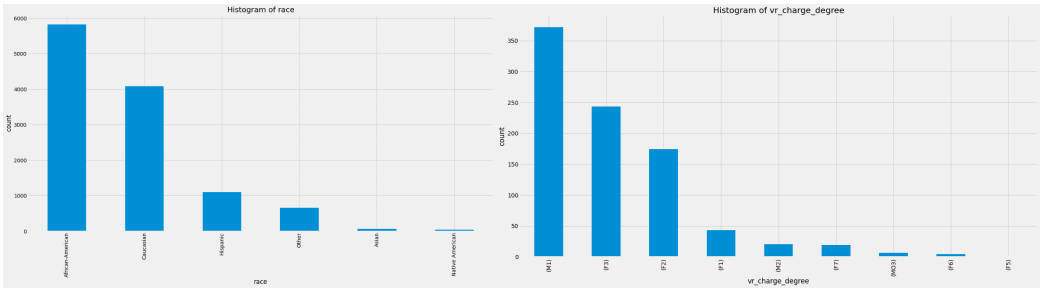


Figure 8: *race* and *vr_charge_degree* histograms

It must be said that the values of *vr_charge_degree* are values that reflects the degree of misdemeanors or felonies crimes in Florida. There are more explanation on these links for felonies and misdemeanors.

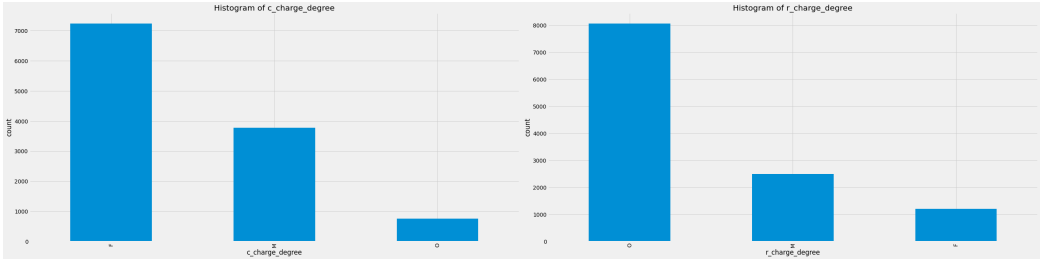


Figure 9: *c_charge_degree* and *r_charge_degree* histograms

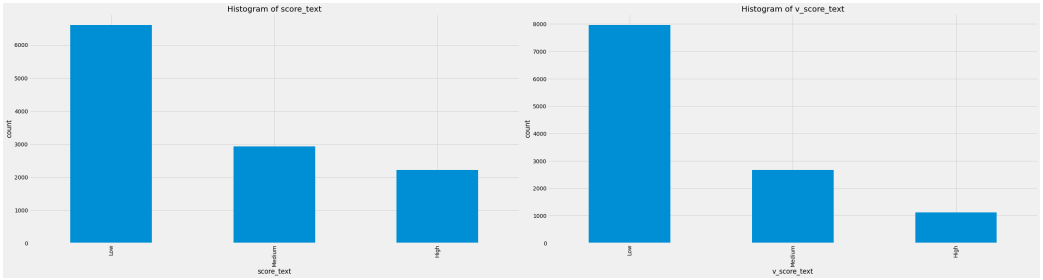


Figure 10: *score_text* and *v_score_text* histograms

3.3.4 Numerical plots by class label

Now we plot the both categorical and numerical columns (the ones we deemed most important respecting the rules defined above) with respect to *is_recid*, our class label. This is to locate variables which we think could be good predictors for the model.

We didn't take in exam the value of -1 for the flag *is_recid* because that value was used to indicate, as ProPublica explain in his jupyter notebook, a missing COMPAS case. Figure 11 shows a stark difference in priors count w.r.t.

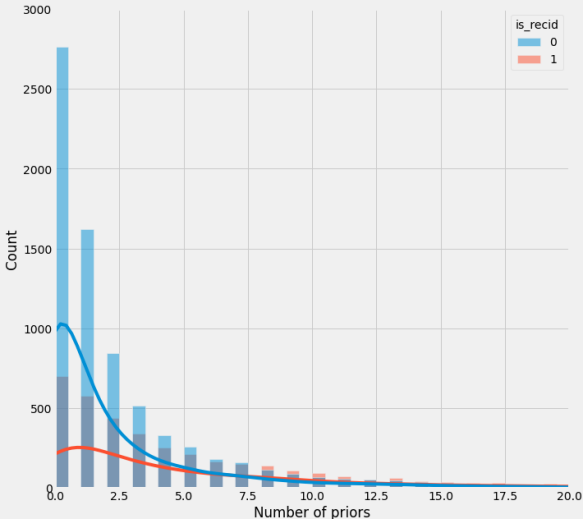


Figure 11: *priors_count* w.r.t. *is_recid*

the color-coded class label. Indeed, there is a difference in the distribution according to the class label. A lot more non-recidive people have from 0 to 2 number of priors, with respect to the recidive people. This means that this column is most likely a very good predictor. Same goes for *juv_fel_count* and *juv_misd_count* (Figure 12, Figure 13) we can see that non recidive people with 0 juvenile felonies are 5000, compared to the 3000 c.a. that are recidive. The exact same goes for this graph here.

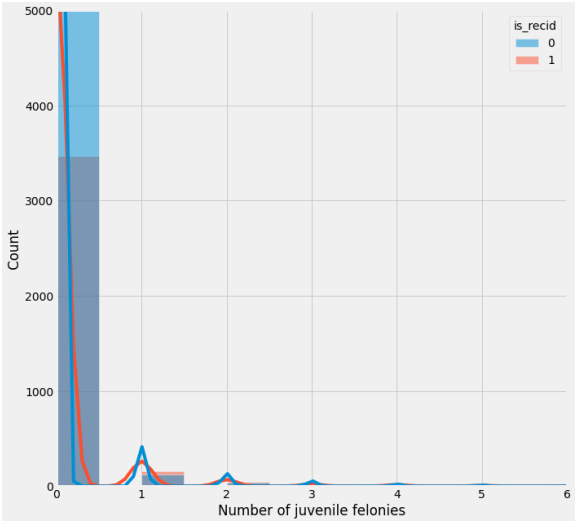


Figure 12: *juv_fel_count* w.r.t. *is_recid*

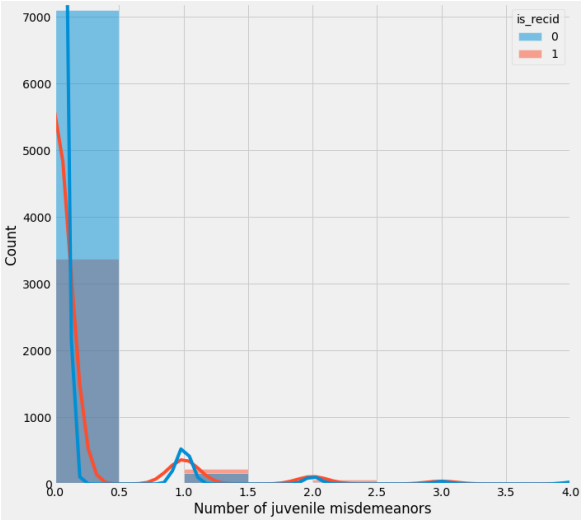


Figure 13: *juv_misd_count* w.r.t. *is_recid*

Looking at both the scatterplots and the correlation matrix, it is possible to see that there is no couple of numerical columns that is strongly correlated. We consider a score as strong correlation if the value is greater than 0.8, and while the couple *(decile_score, v_decile_score) comes close*, it is not very relevant to our analysis anyway.

Figure 14: Scatterplots representing attributes' values in relation to each other and to the value of *is_recid*, {0,1}.

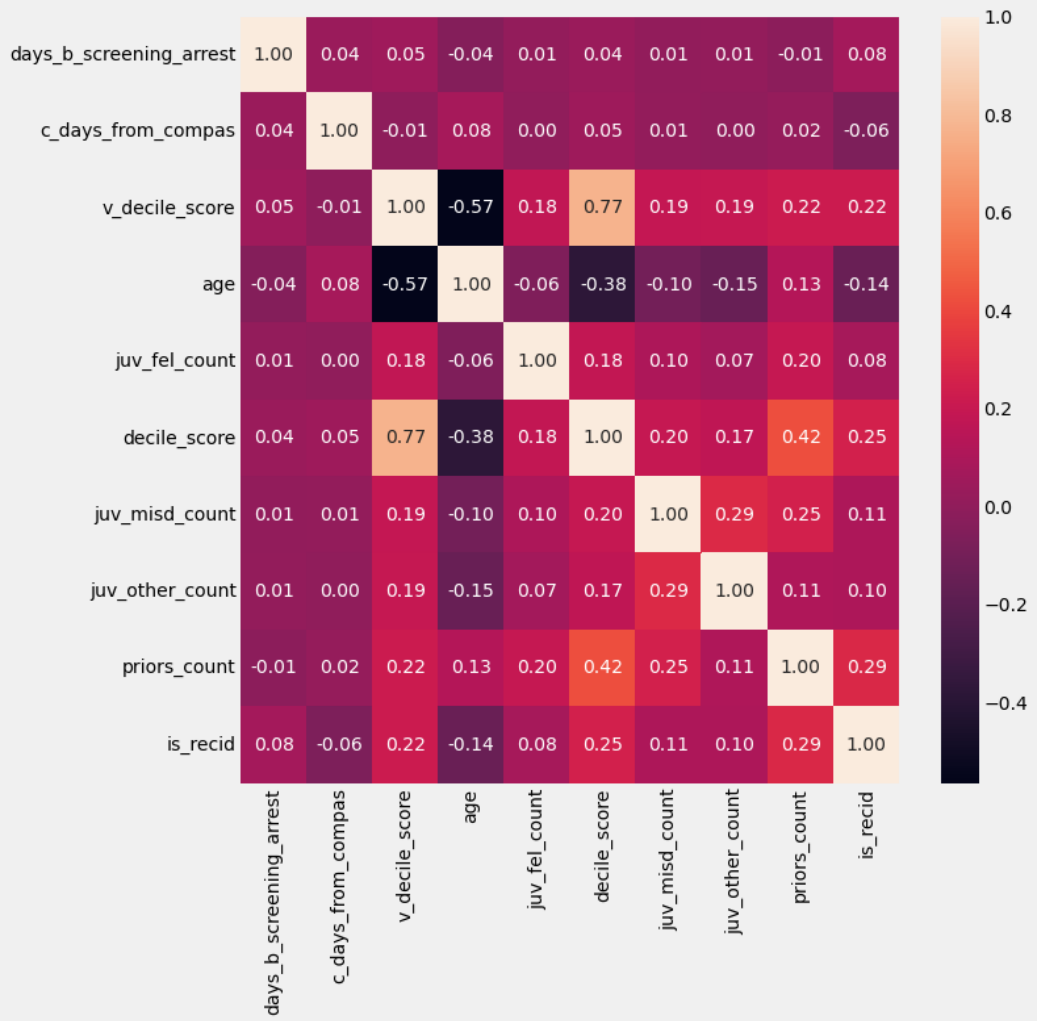


Figure 15: Correlation matrix of the attributes' values.

3.3.5 Categorical plots by class label

The categorical histograms do not show very relevant information, basically every histogram tells us that there are more non recidives than recidives. The only wo useful plots might be the ones of *age_range* and *race*. In the race plot, we can see that the delta between recidives and non recidives is big in caucasians, while it is smaller in african americans, indicating that

african americans might be more recidive in general. Thus, race might be a good predictor. Also, women appear to be generally less recidive than men, as we can see that the ratio recidive over non recidive varies between men and women. Thus, sex might also be a good predictor.

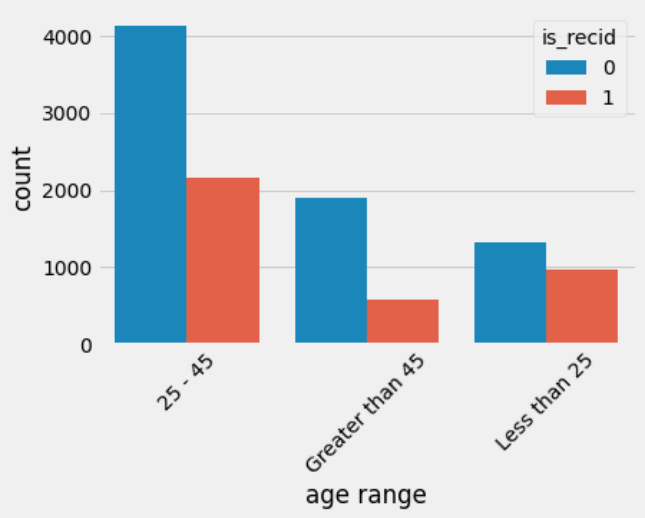


Figure 16: *Age_range* with reference to *is_recid*

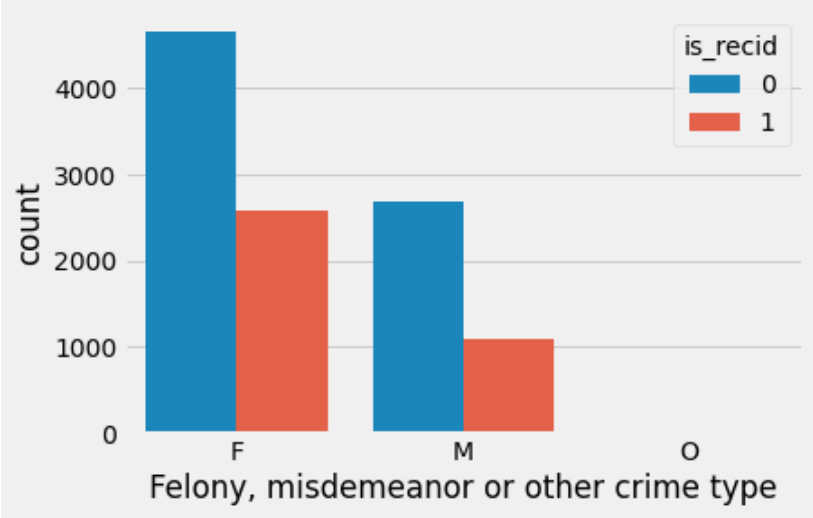


Figure 17: *Crime_type* with reference to *is_recid*

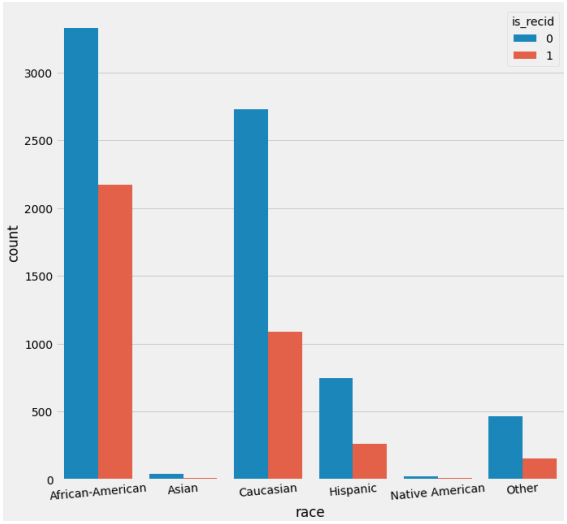


Figure 18: *Race* with reference to *is_recid*

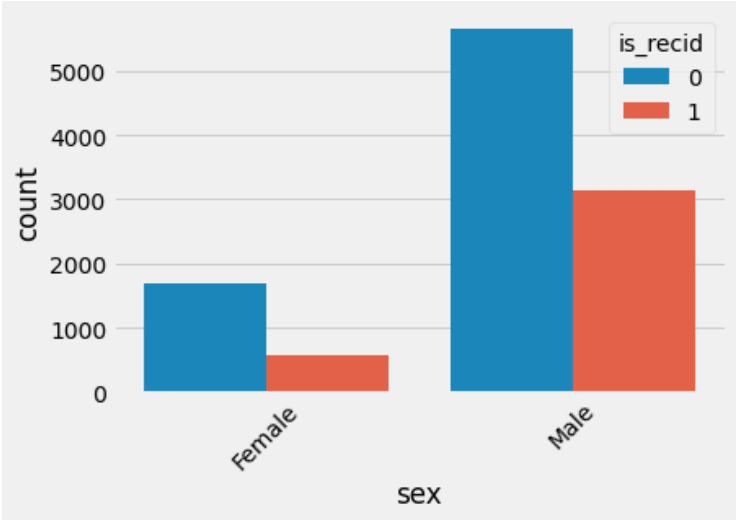


Figure 19: *Sex* with reference to *is_recid*

3.4 Data Quality

For the data quality phase we looked for inconsistencies in data.
Fist, since the ProPublica article suggested a discrepancy between the name column and first and last column combined, we checked for it but didn't find such discrepancies, as can be seen in the Jupyter notebook.

According to ProPublica, in the dataset it's not always clear which criminal case was associated with an individual's COMPAS score. Thus, to match COMPAS scores with accompanying cases, we considered cases with arrest dates or charge dates within 30 days of a COMPAS assessment being conducted. We found that 10% of the rows were "out of bounds".

Then, we checked the c_jail_in and c_jail_out dates to see if all rows had the release date chronologically subsequent to the imprisonment one, and resulted that the 2.56% of the rows didn't respect that. Thus, we considered those rows as "dirty".

At this point we looked for duplicate columns in the data set, and found that compas_screening_date is a duplicate of screening_date and decile_score is a duplicate of decile_score.1. Finally, we checked for the null value percentages of the columns and also plotted the percentages as follows. We planned to remove the columns where the null value percentage exceeded 60%.

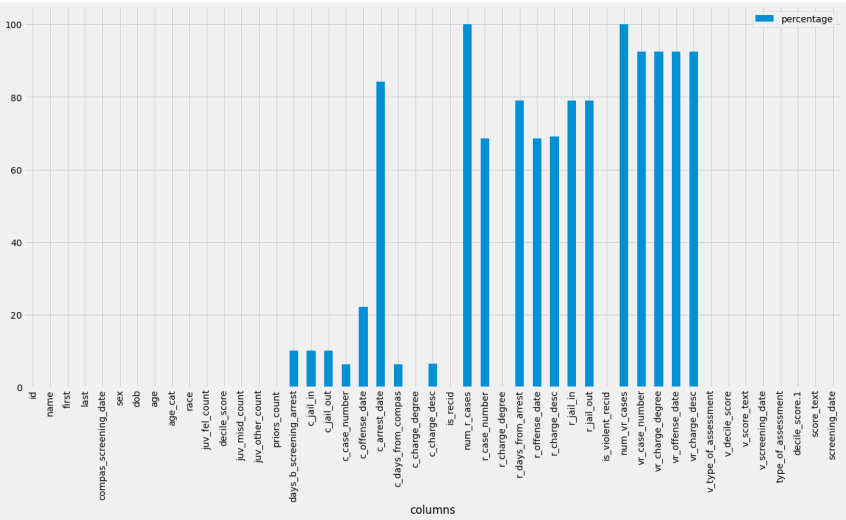


Figure 20: Null values percentages

We also checked the null percentage of the class label, and resulted that the 6.11% of the is_recid column was null.

4 Data preparation

In this section we've modified the dataset to have a better training capability for our model. The dataset output by this phase is what we feed directly to our model. For example, we have dropped some columns that were almost or totally *null*, we reunited some "similar" values or even drop some columns with a too high standard deviation. Here we'll go in details about each transformation.

4.1 Dropping Columns

4.1.1 Dropping duplicate and high *null* value percentage

We dropped all the column that were duplicate and the one with a *null* percentage that exceed the 60%. In particular, the duplicate columns were *decile_score.1* (of *decile_score*), *screening_date* (of *compas_screening_date*) and *v_screening_date* (of compas_screening_date). The attributes that contained mostly *null* values were the following (60%).

- *num_vr_cases*
- *num_r_cases*
- *vr_charge_desc*
- *vr_offense_date*
- *vr_charge_degree*
- *vr_case_number*
- *c_arrest_date*
- *r_jail_in*
- *r_jail_out*
- *r_days_from_arrest*
- *r_charge_desc*
- *r_offense_date*
- *r_case_number*

4.1.2 Dropping Univariate Columns

Here, we dropped columns which contained exactly only one value. These are not useful for modeling, for obvious reasons. These columns do not change value according to the values of other columns, thus it provides no useful insight for the model.

- *compas_screening_date*
- *name*

4.1.3 Dropping excessively discriminant columns

There are some columns which have a very strict correlation with the class label. In particular, all of the columns beginning with *v_*, *vr_* and *r_* are detailing a possible second crime committed by the defendant. i.e., if for a given row these columns are not null, then the class label will equal 1 100% of the time. This is obvious: if these columns are not null, that means that they are detailing another crime committed after the first, and that's why the defendant is classified as recidive in this case. We remove these columns because they are too discriminant, they would help the model too much and we could risk overfitting it.

4.1.4 Dropping IDs and useless columns

We also dropped the IDs, that is *id* and *c_case_number*, because IDs are notoriously useless for modeling as they have no logical correlation with the other columns. They are generated progressively. Moreover, it was necessary to drop the column *is_violent_recid* because it is not our class label. We also removed the following categorical columns because they had too much levels and would not help the model learn anything useful.

- *compas_screening_date*
- *name*
- *first*
- *last*
- *c_offense_date*

- *c_days_from_compas*
- *score_text*

4.1.5 Dropping redundancies

It was also useful to drop some redundancies attributes, that is *dob* and *age_cat*, or rather columns that simply repeat the age of the subject.

4.1.6 Dropping dirty data

The values of *days_b_screening_arrest* where limited only within the interval $[-30, 30]$, because as explained by ProPublica in its Jupyter notebook:
"If the charge date of a defendants Compas scored crime was not within 30 days from when the person was arrested, we assume that because of data quality reasons, that we do not have the right offense". For those reasons, it was necessary to remove the rows that had a value of *day_b_screening_arrest* not included between the interval $[-30, 30]$. It was also necessary to remove the rows that contain *is_recid* equals to -1, for the same reasons explained beforehand in section 3.3.4. Moreover the dataset was cleared from the rows that had at least one *null* value.

4.2 Transformations

4.2.1 New columns

It is obvious that *c_jail_in* and *c_jail_out* could be annexed in a single column *days_in_prison* (detailing the number of days spent in prison), so we created said column and dropped the previous two because of the created redundancy. We dropped, for data quality reasons, rows where the *jail_in* date was later than the *jail_out* date.

4.2.2 Unifying columns

The *c_charge_desc* column had way too many values to be useful for the modeling, so we decided to categorize all the crimes into 7 macro categories:

- Assoult
- Robbery

- Possession of drugs
- Battery
- Burglary
- Drug delivery
- DUI (Driving Under Influence)
- Other

4.2.3 Columns binarization

We binarized some attributes that had values strongly skewed thowards one single value.

- *priors_count* with values "*less than 10*" or "*10 or more*"
- *juv_misd_count* with values 0 or > 0
- *juv_other_count* with values 0 or > 0
- *juv_fel_count* with values 0 or > 0

5 Modeling

As we already described in the Data Preparation chapter, we did some transformations on the data. Therefore, we needed to see how they worked. So, we ran some models on the dataset to see whether they performed well or not. To be more specific we used the following classifiers:

- Decision Tree AdaBoost
- Naive Bayes
- Random Forest

First we encoded categories as numbers (as needed for the models), then after the analysis of the dataset labels, we immediately noticed the presence of an imbalance problem in the dataset in favor of the "0" class (the non recidive ones), so we had two options to balance data at this point:

- Undersampling
- Oversampling

We chose to do both oversampling, (using SMOTE oversampler), and undersampling (manually) to see which strategy yielded better results. Furthermore, to get the optimal hyperparameters to use for our model, we utilized the Grid Search method. For each model, we gave the search function a set of hyperparameter values, then launched the search, and then picked the best ones. Once we had those, we were able to create six instances of models: three classifiers to fit on the undersampled dataset, and three other classifiers to fit on the oversampled dataset. For the Random Forest classifier, we specified these hyperparameter values:

- Max depth (3, 5 or 10)
- Min samples split (2, 5 or 10)
- A criterion (gini or entropy)
- Max leaf nodes (5, 10, or 15)

Secondly, we applied it at the Naive Bayes classifier (where we assigned a range of very small values to the *var_smoothing* hyperparameter). Finally, we did the same for the Decision Tree AdaBoost classifier. We reduced the number of values relative to the decision tree itself (at least, compared to the Random Forest parameter choice) because then the combinations would be too much and the fitting process would take a very long time.

- Max depth (5 or 10)
- Min samples split (2 or 5)
- A criterion (gini or entropy)
- Max leaf nodes (5 or 10)
- Number of estimators (10, 20 or 50)
- Learning Rate (0.01, 0.1)

Then, we fit the models on the data itself. We used a 10-fold cross-validation technique with a split or 80% - 20%. In the following section results are detailed.

6 Evaluation

6.1 Choosing the best models

In our analyses, we discovered that the oversampling dataset yielded the best results. We inferred this from seeing that all of the metrics in the undersampled dataset case were (in mean) worse than the oversampled case. So, we decided to discard all of the models fit on the undersampled dataset.

Of the various classifiers fit on oversampled data, the AdaBoost with Decision Tree is the best one, with a F-measure of .68, followed by the Random Forest classifier, with a F-score of .66. We decided to pick these two models and exclude the Naive Bayes, because even though it had the same F-measure as the Random Forest, it had a lower accuracy.

	F1	Accuracy	Recall	Precision
AdaBoost	0.68	0.68	0.67	0.69
Random Forest	0.66	0.65	0.68	0.67
Naive Bayes	0.66	0.61	0.76	0.59
AdaBoost	0.65	0.64	0.69	0.61
Random Forest	0.64	0.65	0.63	0.63
Naive Bayes	0.64	0.62	0.71	0.58

Table 4: Modeling metrics table for both the datasets. Blue: oversampling dataset - Red: undersampling dataset

6.2 Comparing the best models chosen

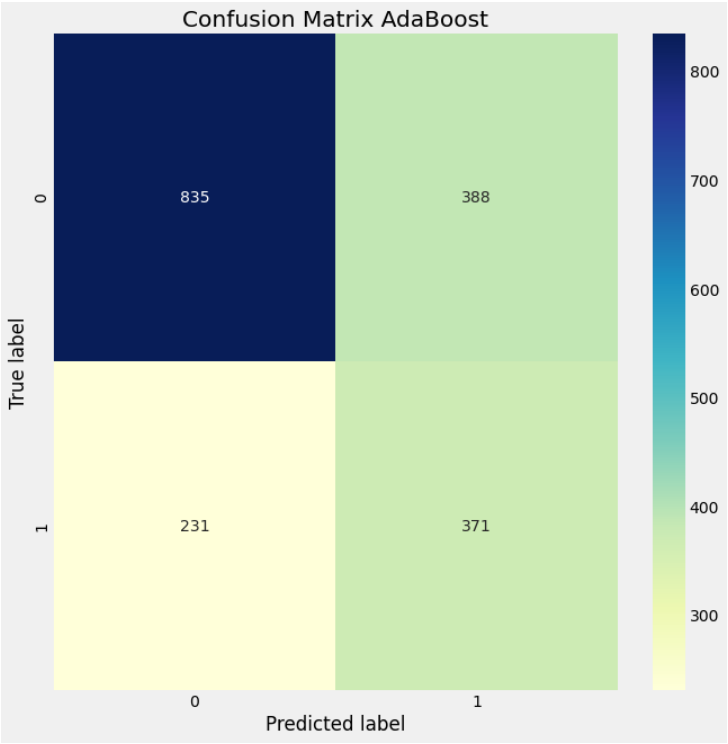


Figure 21: Confusion matrix Adaboost

In this confusion matrix, is possible to see the effective accuracy, recall and precision of the Adaboost classifier with a more graphical representation of the metrics resulted.

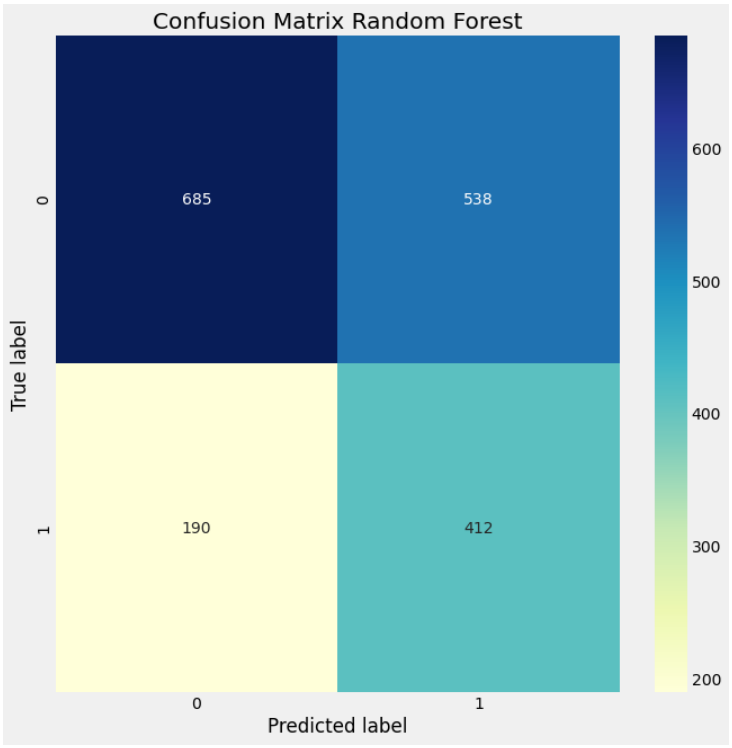


Figure 22: Confusion matrix Random Forest

Here is it possible to see the confusion matrix of the Random Forest classifier, which had the same F-score of the Gaussian Naive Bayes but a better accuracy.
As we can see from the ROC Curve, the best model in both classes is the DT AdaBoost, as it has a slightly bigger area than Random Forest.

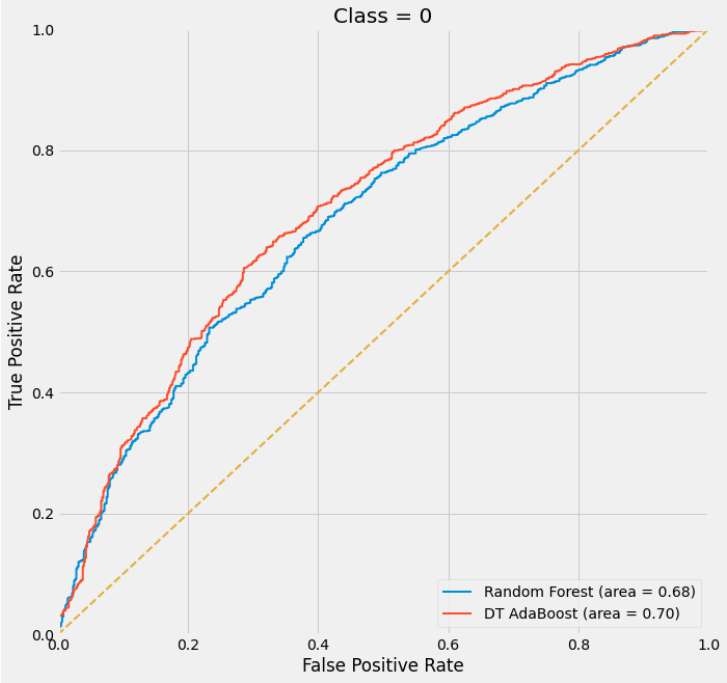


Figure 23: ROC Curve, negative class

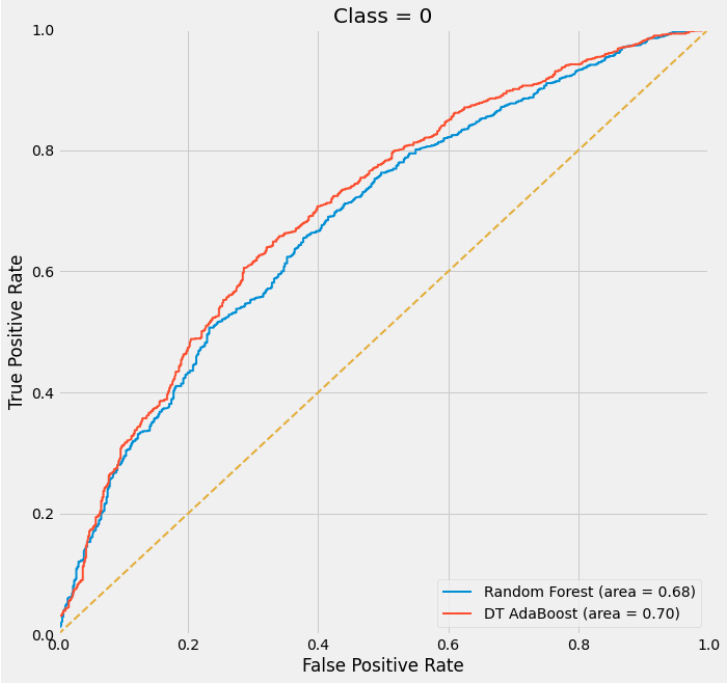


Figure 24: ROC Curve, positive class

7 Conclusions

All the choices that we made had a underlying logic with the goal of improving metrics. We matched our Data Mining goals, but as we learned during the Business Understanding Phase, the dataset was quite "dirty" and the premises formulated by ProPublica were not the best. Considering how dirty and unbalanced the dataset is, the conclusion we've reached is that with this dataset is impossible to obtain metrics balanced and all higher than 60% or 70%.