DATA MINING PROJECT - MATTEO GIACCONE

Predict the course of Covid-19 disease: Algorithms comparison and selection

Machine Learning techniques applied to Covid-19 Data analysis

Dataset's choice

PATIENTS INDEXED DATASET - ESTADO DO ESPIRITO SANTO (BRAZIL)

Comorbidities	Gender, Age, Education
Symtomps	Diagnosis Date
Clinical State	Position

Python environment and libraries

Matplotlib: Visualization

Jupyter Notebook

Pandas, Numpy: Data interaction

SkLearn: ML models and metrics

SMOTE: Data recalibration

XGBoost

Espírito Santo Brazilian State Covid Dataset

WEB INTERFACE

Data Cleanup

The dataset is only available in Portuguese, consequently the first step for the analysis was to translate it into English and eliminate some unnecessary columns for the analysis.

Science papers

WHO - World Health Organization

COVID-19 in Brazilian cities: Impact of social determinants (https://doi.org/10.1371/journal.pone.0257347)

Nature and Pubmed researches

Dataframe output (from first step)

	Diagnosis Date	Closing Date	Classification	Evolution	Confirmation Criteria	Status Notification	Position	Age range	Age	Gender	Education Level	Pregnancy	Fever	Respiratory Difficulty
0	2021-04-23	2021-05-05	Confirmed	Cure	Laboratorial	Closed	(SERRA, JARDIM ATLÂNTICO)	60-69	64	M	Complete elementary school	No	Yes	No
1	2021-12-24	2021-12-20	Discarded	Unknown	Laboratorial	Closed	(PIUMA, JARDIM MAILY)	70-79	74	F	Incomplete high school	No	Yes	No
2	2021-12-21	NaN	Confirmed	Unknown	Laboratorial	Open	(GUARAPARI, ITAPEBUSSU)	40-49	41	F	Complete higher education	No	Yes	No
3	2021-12-23	2020-12-19	Suspect	Unknown	Laboratorial	Closed	(PIUMA, CENTRO)	20-29	22	M	Complete high school	No	No	No
4	2021-12-24	NaN	Suspect	Unknown	Clinic	Open	(VITORIA, BONFIM)	40-49	42	F	Complete high school	No	No	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2208172	2020-02-15	2020-04-16	Discarded	Unknown	Laboratorial	Closed	(VILA VELHA, JOCKEY DE ITAPARICA)	30-39	39	F	Unknown	No	Yes	Yes
2208173	2020-02-15	2020-04-16	Discarded	Unknown	Laboratorial	Closed	(VILA VELHA, JOCKEY DE ITAPARICA)	0-4	4	F	Unknown	No	Yes	No
2208174	2020-02-25	2020-04-15	Discarded	Unknown	Laboratorial	Closed	(ARACRUZ, JEQUITIBÁ)	20-29	25	M	Unknown	No	Yes	No
2208175	2020-02-13	2020-04-16	Discarded	Death	Laboratorial	Closed	(SERRA, BAIRRO DE FÁTIMA)	50-59	54	M	Unknown	No	Yes	Yes
2208176	2020-01-20	2020-10-29	Discarded	Unknown	Laboratorial	Closed	(AGUIA BRANCA, CENTRO)	40-49	46	F	Unknown	No	Yes	No

Data Preprocessing

The choices on the columns to keep were made in accordance with the current scientific literature. In some classes the order is relevant (Age, Pregnancy status, number of comorbidities / symptoms ...), while in others a simple pandas.dummify has been applied

Data import and filters

Pandas

Import translated dataset

Columns selection

Pandas

Select only useful columns

Closed cases selection

Pandas

Get closed cases

String to int classification

Pandas

transform string data into integer classes

Standard Scaler

Sklearn

Standard scaler on "over 1" classes (ex: Age, Pregnancy, Age range...)

SMOTE: Synthetic Minority Oversampling Technique

IMBALANCED CLASSIFICATION

Imbalanced classification involves developing predictive models on classification datasets that have a severe class imbalance. One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don’t add any new information to the model. Instead, new examples can be synthesized from the existing examples.
In our original dataset, we had 600242 closed cases, where:

- 587354 are "cured cases" (97.85%)
- 12888 are "death cases" (2.15%)

After resampling, we had:

- 587354 are "cured cases"
- 293677 are "death cases"

Models selected

(N.B. extra algorithms are XGBoost, SVC and KN)

Logistic Regression with SAG solver: Statistical model that in its basic form uses a logistic function to model a binary dependent variable

Linear Discriminant Analysis (LDA): Generalization of Fisher's linear discriminant, a method used in statistics and other fields, to find a linear combination of features that characterizes or separates two or more classes of objects or events.

Gaussian Naive Bayes (NB): Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features (see Bayes classifier).

Decision Tree Classifier (DTC): It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called classification trees

Random Forest Classifier: Consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction

Metrics

R2 Score

Measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model

MAE

Mean Absolute Error, a measure of errors between paired observations expressing the same phenomenon.

Accuracy

Simply a ratio of correctly predicted observation to the total observations.

Precision

Ratio of correctly predicted positive observations to the total predicted positive observations.

Recall

Fraction of relevant instances that were retrieved.

F1 Score

Weighted average of Precision and Recall

Kappa Score

is used to measure inter-rater reliability (and also intra-rater reliability) for qualitative (categorical) items.

ROC AuC

Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.

PR AuC

Area Under the Precision Recall Curve (ROC AUC) from prediction scores.

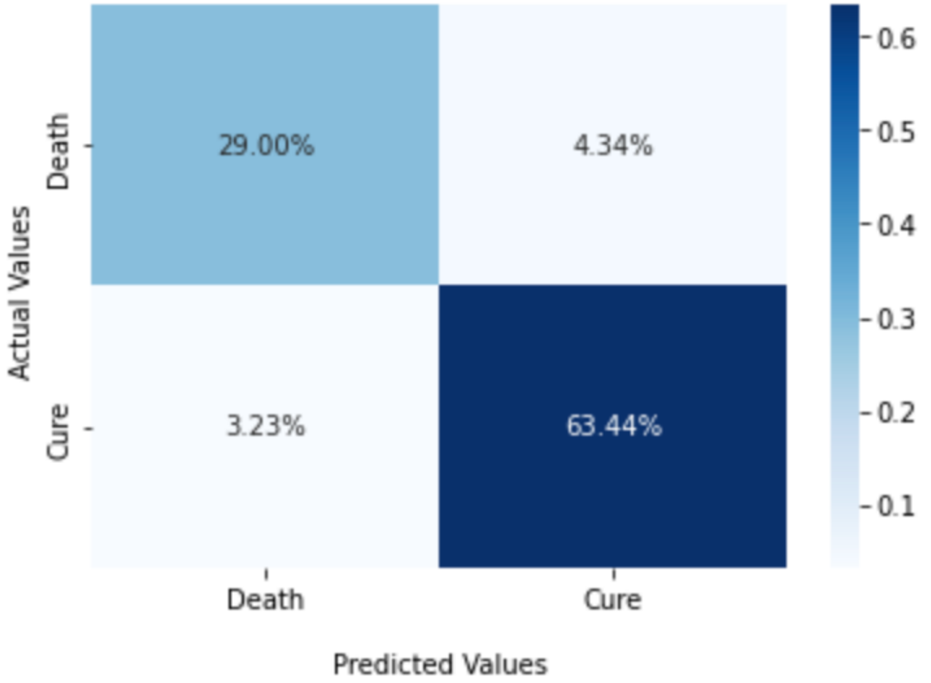
Experiment Function

10-FOLD-CROSS-VALIDATION, AGGREGATED VALUES AND DATA STRCTURE

10-fold cross validation would perform the fitting procedure a total of ten times, with each fit being performed on a training set consisting of 90% of the total training set selected at random, with the remaining 10% used as a hold out set for validation. Every model is trained and tested and his output and aggregated values is entered into a python dictionary

Logistic Regression

Bad Predictions: 66663 over 881031 (7.57%)

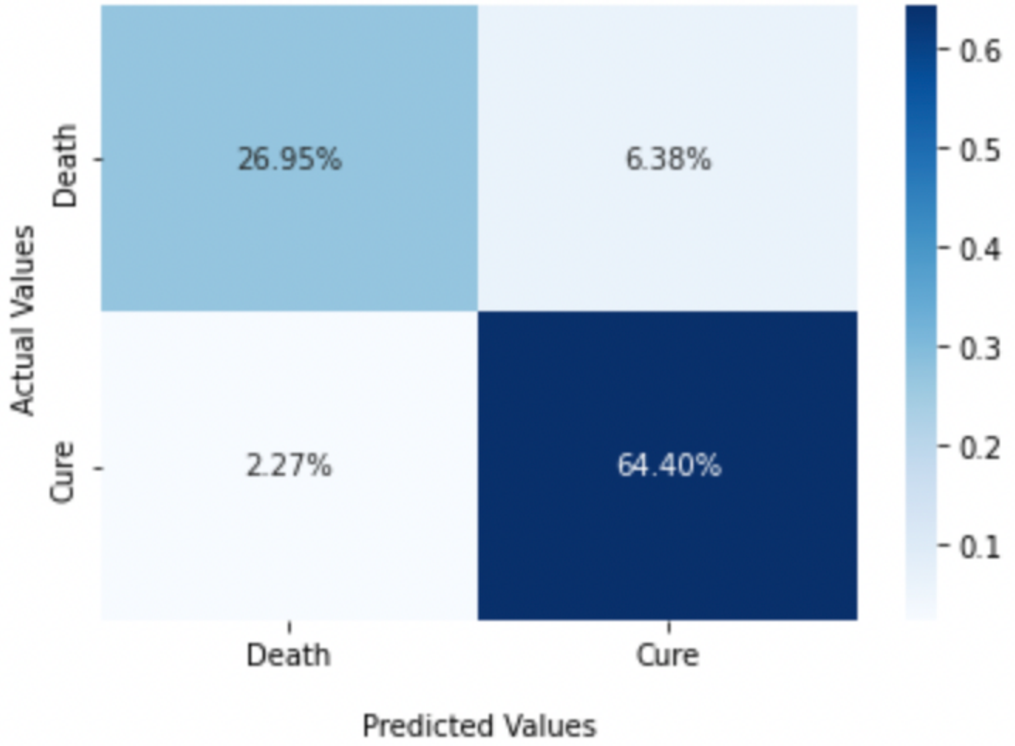


	fit_time	score_time	test_r2	train_r2	test_mae	train_mae
mean	206.603830	0.143427	0.659509	0.701185	-0.075665	-0.066403
std	57.305816	0.009138	0.135849	0.004597	0.030189	0.001022
min	180.787356	0.133075	0.295760	0.697491	-0.156499	-0.067224
max	369.281039	0.164371	0.730266	0.713126	-0.059941	-0.063750

	test_accuracy	train_accuracy	test_precision	train_precision	test_recall	train_recall	test_f1	train_f1	test_kappa	train_kappa	test_roc_auc	train_roc_auc	test_pr_auc	train_pr_auc
Mean	0.924335	0.933597	0.936159	0.937405	0.951535	0.964822	0.943267	0.950916	0.829372	0.848375	0.971489	0.980239	0.985561	0.989334
Std	0.030189	0.001022	0.013350	0.001023	0.044695	0.000482	0.024872	0.000741	0.062228	0.002380	0.027088	0.000648	0.011480	0.000406
Min	0.843501	0.932776	0.898973	0.936655	0.824755	0.964347	0.875416	0.950315	0.667142	0.846479	0.896677	0.979782	0.955361	0.989078
Max	0.940059	0.936250	0.942301	0.940158	0.971414	0.965852	0.955768	0.952832	0.862889	0.854576	0.984061	0.982015	0.991499	0.990458

Linear Discriminant Analysis

Bad Predictions: 76264 over 881031 (8.66%)

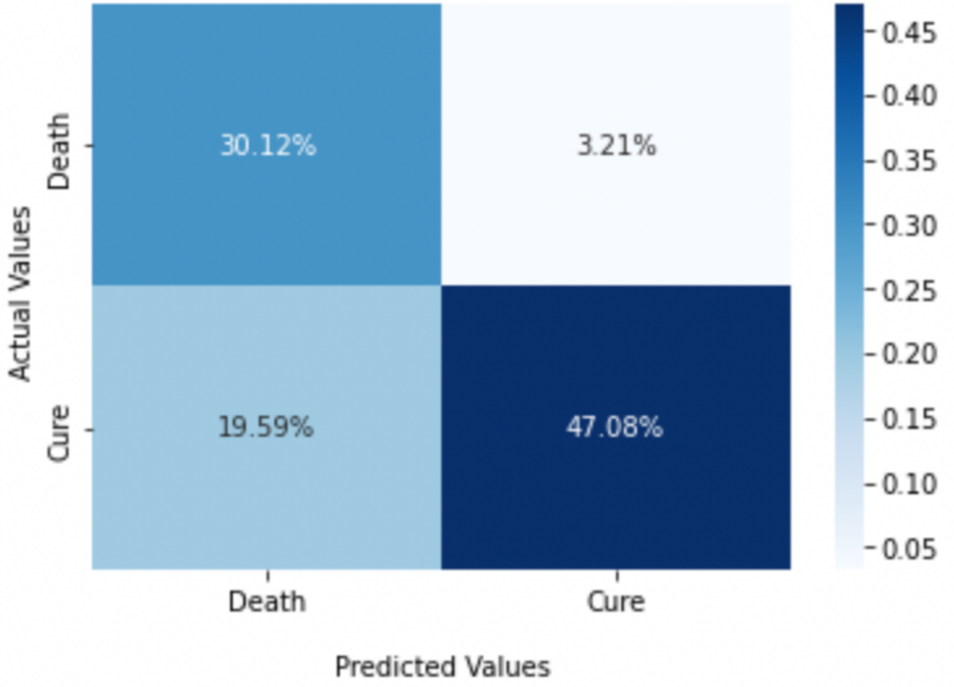


	fit_time	score_time	test_r2	train_r2	test_mae	train_mae
mean	4.912799	0.141828	0.610470	0.648422	-0.086562	-0.078129
std	0.145746	0.009227	0.131396	0.004110	0.029199	0.000913
min	4.717790	0.128438	0.256840	0.644479	-0.165148	-0.079005
max	5.100709	0.158465	0.678475	0.657571	-0.071450	-0.076095

	test_accuracy	train_accuracy	test_precision	train_precision	test_recall	train_recall	test_f1	train_f1	test_kappa	train_kappa	test_roc_auc	train_roc_auc	test_pr_auc	train_pr_auc
Mean	0.913438	0.921871	0.909858	0.911054	0.965925	0.978321	0.936612	0.943490	0.800050	0.817472	0.970684	0.977638	0.985117	0.988151
Std	0.029199	0.000913	0.012129	0.000870	0.043376	0.000464	0.023624	0.000646	0.061020	0.002187	0.023603	0.000622	0.010368	0.000389
Min	0.834852	0.920995	0.877054	0.910067	0.842785	0.977791	0.871865	0.942866	0.640413	0.815390	0.904907	0.977143	0.957066	0.987876
Max	0.928550	0.923905	0.916166	0.912944	0.983996	0.979235	0.948353	0.944928	0.832960	0.822335	0.981958	0.979284	0.990533	0.989202

Gaussian Naive Bayes

Bad Predictions: 200854 over 881031 (22.80%)

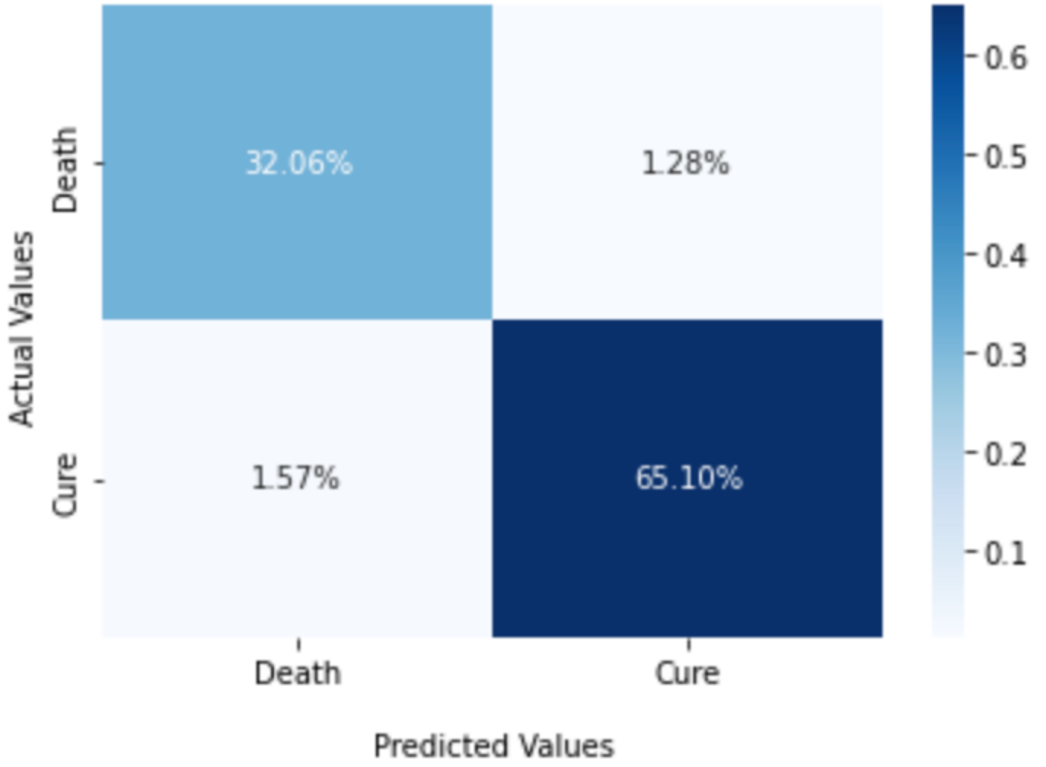


	fit_time	score_time	test_r2	train_r2	test_mae	train_mae
mean	0.457834	0.137112	-0.025890	0.032540	-0.227976	-0.214991
std	0.015623	0.003872	0.500863	0.496868	0.111303	0.110415
min	0.429189	0.132112	-1.392247	-1.375243	-0.531610	-0.527832
max	0.479883	0.146485	0.269239	0.290411	-0.162389	-0.157686

	test_accuracy	train_accuracy	test_precision	train_precision	test_recall	train_recall	test_f1	train_f1	test_kappa	train_kappa	test_roc_auc	train_roc_auc	test_pr_auc	train_pr_auc
Mean	0.772024	0.785009	0.940313	0.944649	0.706200	0.725796	0.790331	0.801773	0.552978	0.577256	0.906965	0.922134	0.945905	0.953500
Std	0.111303	0.110415	0.045672	0.037941	0.174702	0.188537	0.157985	0.159769	0.163504	0.152089	0.051505	0.001065	0.018890	0.001293
Min	0.468390	0.472168	0.811611	0.840401	0.210059	0.212049	0.345056	0.348809	0.140243	0.146871	0.762033	0.920901	0.893208	0.952472
Max	0.837611	0.842314	0.965641	0.982410	0.792053	0.928746	0.866727	0.882368	0.663604	0.668231	0.934157	0.924294	0.958277	0.957011

Decision Tree Classifier

Bad Predictions: 25049 over 881031 (2.84%)

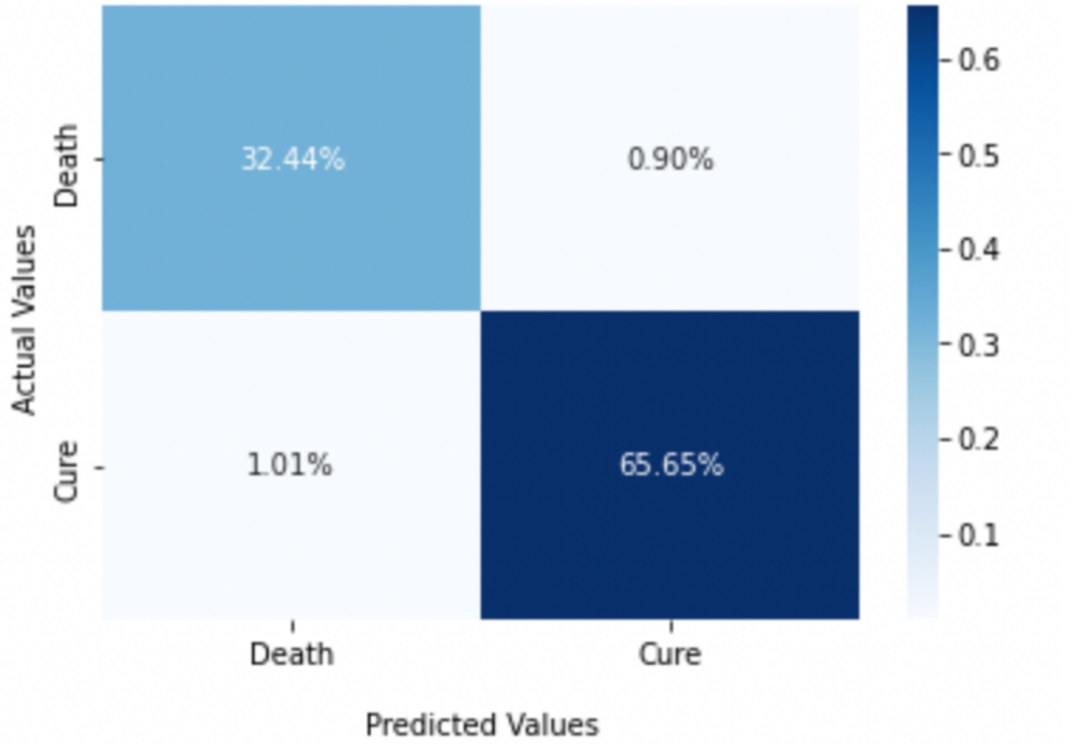


	fit_time	score_time	test_r2	train_r2	test_mae	train_mae
mean	4.794570	0.097814	0.872268	0.983069	-0.028385	-0.003763
std	0.132927	0.001912	0.144271	0.003560	0.032060	0.000791
min	4.495614	0.095266	0.507219	0.981788	-0.109507	-0.004047
max	4.990692	0.101816	0.942436	0.993195	-0.012792	-0.001512

	test_accuracy	train_accuracy	test_precision	train_precision	test_recall	train_recall	test_f1	train_f1	test_kappa	train_kappa	test_roc_auc	train_roc_auc	test_pr_auc	train_pr_auc
Mean	0.971615	0.996237	0.982856	0.996825	0.976619	0.997533	0.979039	0.997179	0.934807	0.991531	0.972317	0.999947	0.977886	0.999971
Std	0.032060	0.000791	0.042444	0.000621	0.022969	0.000572	0.022632	0.000593	0.077526	0.001781	0.047469	0.000016	0.041016	0.000009
Min	0.890493	0.995953	0.862059	0.996547	0.912505	0.997229	0.923747	0.996966	0.733033	0.990890	0.840890	0.999940	0.862990	0.999968
Max	0.987208	0.998488	0.996775	0.998578	0.994943	0.999154	0.990344	0.998866	0.971401	0.996597	0.992206	0.999992	0.994248	0.999996

Random Forest Classifier

Bad Predictions: 16844 over 881031 (1.91%)



	fit_time	score_time	test_r2	train_r2	test_mae	train_mae
mean	26.568622	1.016203	0.914141	0.983059	-0.019080	-0.003765
std	1.382666	0.055393	0.065116	0.003558	0.014470	0.000791
min	24.404291	0.929742	0.733741	0.981777	-0.059169	-0.004050
max	28.389063	1.081947	0.945451	0.993178	-0.012122	-0.001516

	test_accuracy	train_accuracy	test_precision	train_precision	test_recall	train_recall	test_f1	train_f1	test_kappa	train_kappa	test_roc_auc	train_roc_auc	test_pr_auc	train_pr_auc
Mean	0.980920	0.996235	0.987036	0.996137	0.984806	0.998225	0.985786	0.997180	0.956749	0.991520	0.996912	0.999851	0.997832	0.999924
Std	0.014470	0.000791	0.021474	0.000851	0.005373	0.000342	0.010387	0.000592	0.034061	0.001782	0.005137	0.000041	0.004446	0.000021
Min	0.940831	0.995950	0.925928	0.995767	0.970954	0.998002	0.957118	0.996966	0.862062	0.990878	0.982334	0.999835	0.985192	0.999916
Max	0.987878	0.998484	0.994521	0.998544	0.990483	0.999183	0.990884	0.998863	0.972798	0.996588	0.998845	0.999966	0.999391	0.999983

From the analysis above, we can clearly see that the best models for our purpose are the RandomForestClassifier and the DecisionTreeClassifier.
Since the second model is nothing more than a combination of 100 DecisionTrees, considering an execution time per single cycle of 138 seconds (2.3 minutes), we will examine, at the moment, only these two.
For our purpose, **the choice falls on the RandomForest model**, given the relatively short execution times and accuracy in the predictions.

Random Forest

Original set dimension: 600242 x 52
Resampled (SMOTE) set dimension:

- 587354 "cured cases"
- 293677 "death cases"

SMOTE Train set dimension: 616721 x 62
SMOTE Test set dimension: 254310 x 62

	Pred_Evolution	Evolution	Age range	Age	Gender	Education Level	Pregnancy	sum_comorb	sum_sym
77	Death	Cure	70-79	74	F	Unknown	No	1	1
115	Death	Cure	50-59	53	M	Incomplete 5th to 8th grade of elementary school	No	0	2
160	Cure	Death	50-59	57	F	Unknown	No	0	4
205	Death	Cure	60-69	65	F	Incomplete 1st to 4th grade of elementary school	No	2	2
284	Cure	Death	70-79	73	F	Unknown	No	0	3
...	...	...	...	...	...	...	...	...	...
264040	Death	Cure	60-69	61	F	Complete elementary school	No	1	3
264138	Death	Cure	60-69	64	F	Unknown	No	1	2
264168	Cure	Death	60-69	62	M	Complete higher education	No	2	4
264253	Death	Cure	50-59	56	F	Complete high school	No	0	4
264257	Cure	Death	30-39	37	F	Incomplete 5th to 8th grade of elementary school	No	0	0

R2 Score and Mean Absolute Error

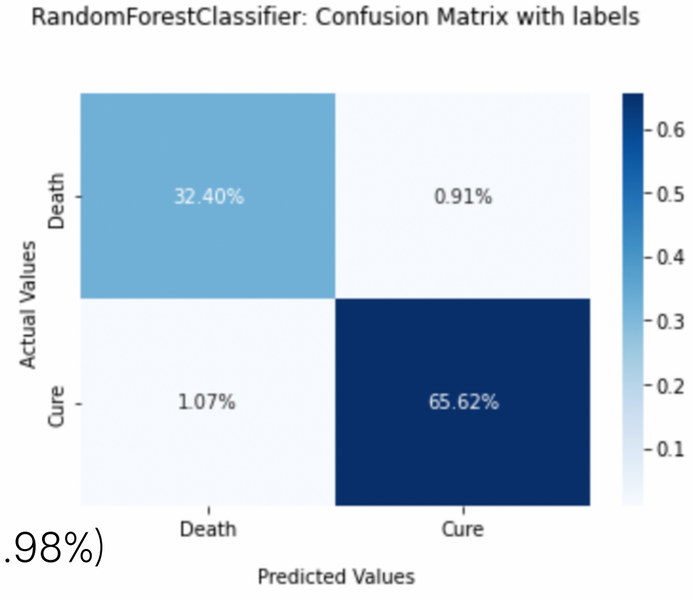
- R2 Score: 0.911
- MAE: 0.020

Other Metrics

- Accuracy: 0.980
- Precision: 0.986
- Recall: 0.984
- F1-Score: 0.985
- Kappa-Score: 0.955

AUC

- ROC AuC score: 0.996
- PR AuC score: 0.997



Bad Predictions: 5235 over 264310 (1.98%)

Extra: Construct Graph for contact tracing

Data Selection and preprocessing

Import cleaned data and select only closed cases.
Encode labels as in the previous case (for more information, see the associated jupyter notebook)

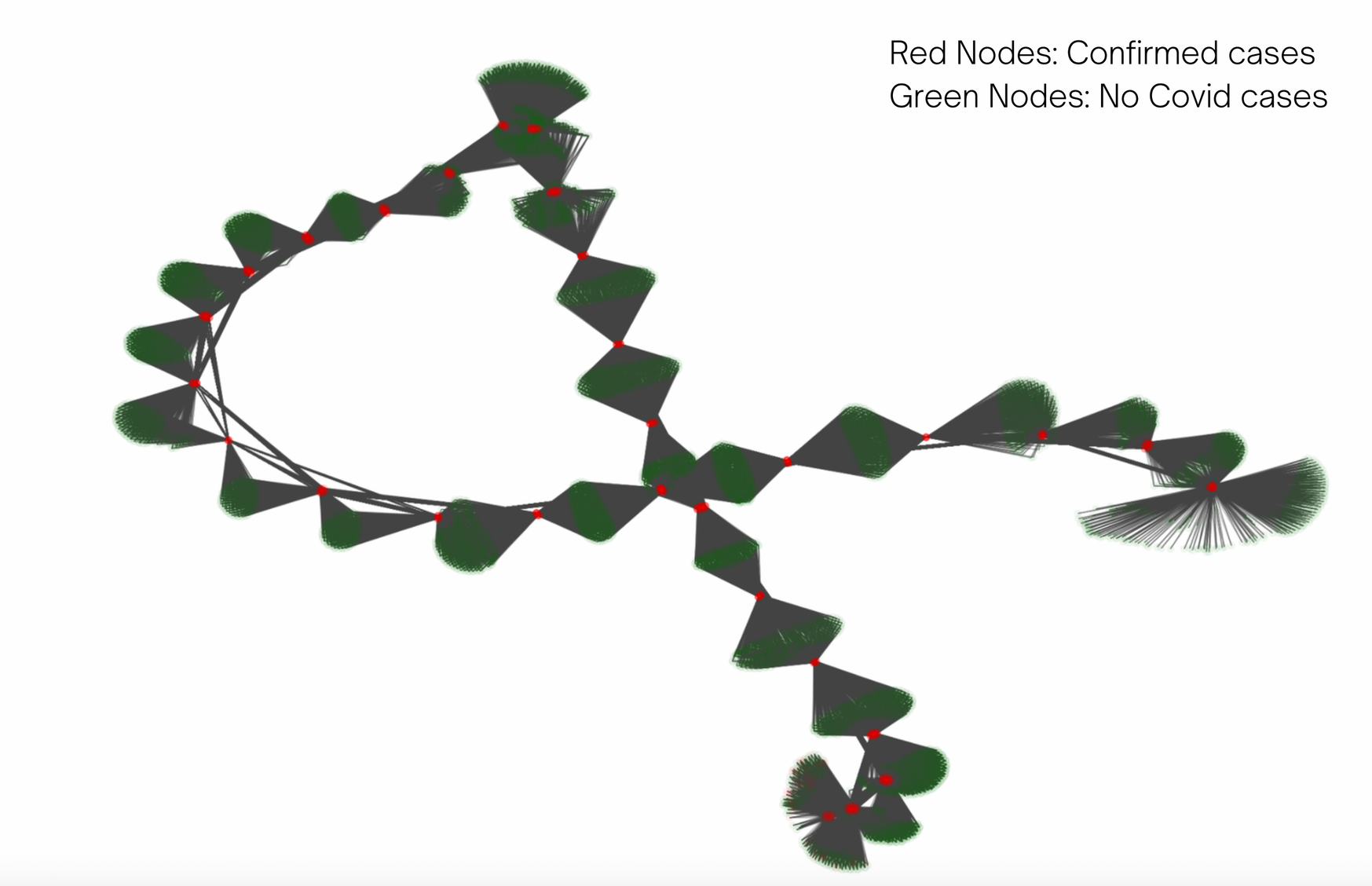
Construct Graph from edgelist

Create edgelist and construct graph with NetworkX library

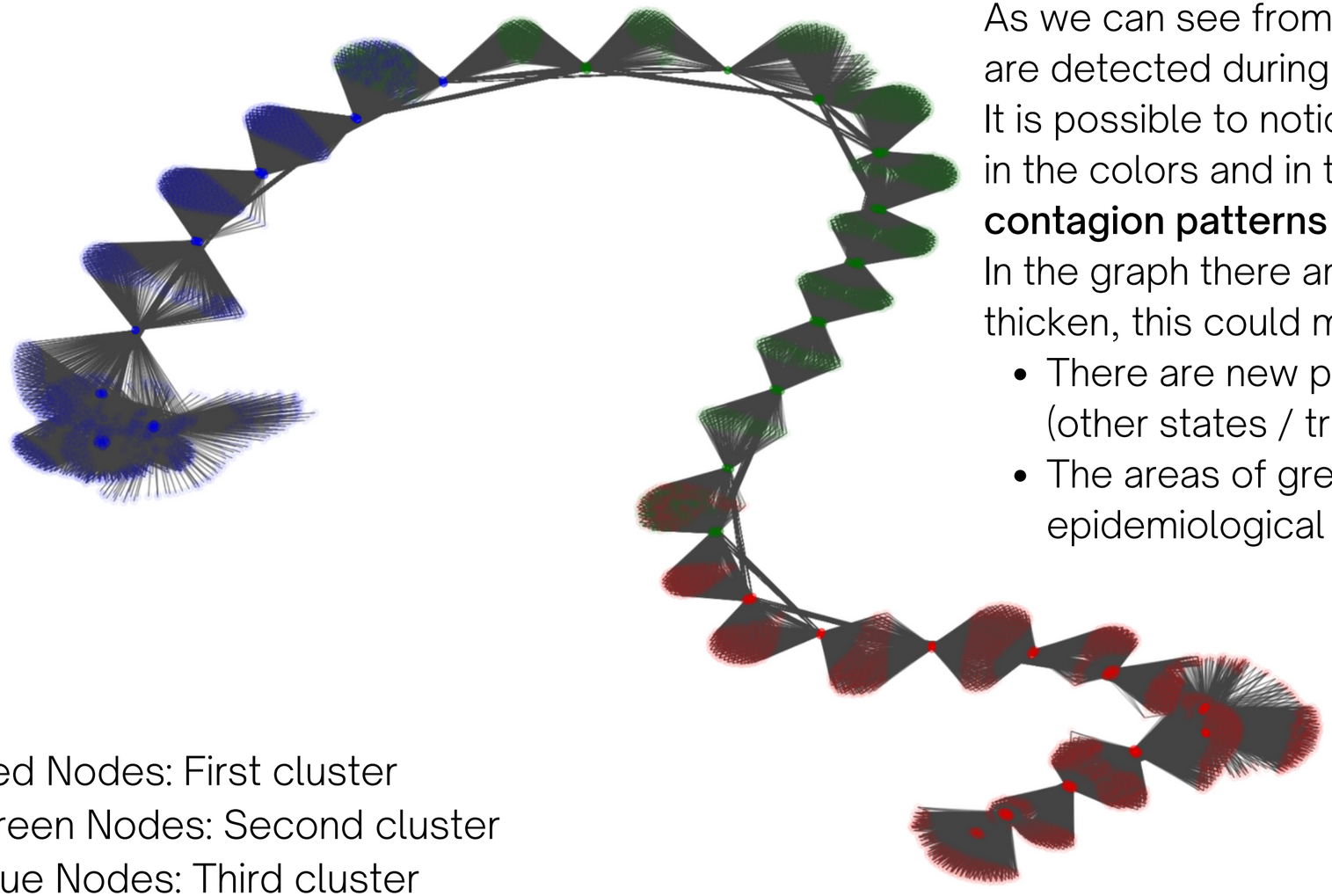
Add node's labels ("cured" or "death") and node's features

Add edge weights, computed as product of:
- distance (in term of days and position)
- number of confirmed cases over total ones

NetworkX Graph Drawing



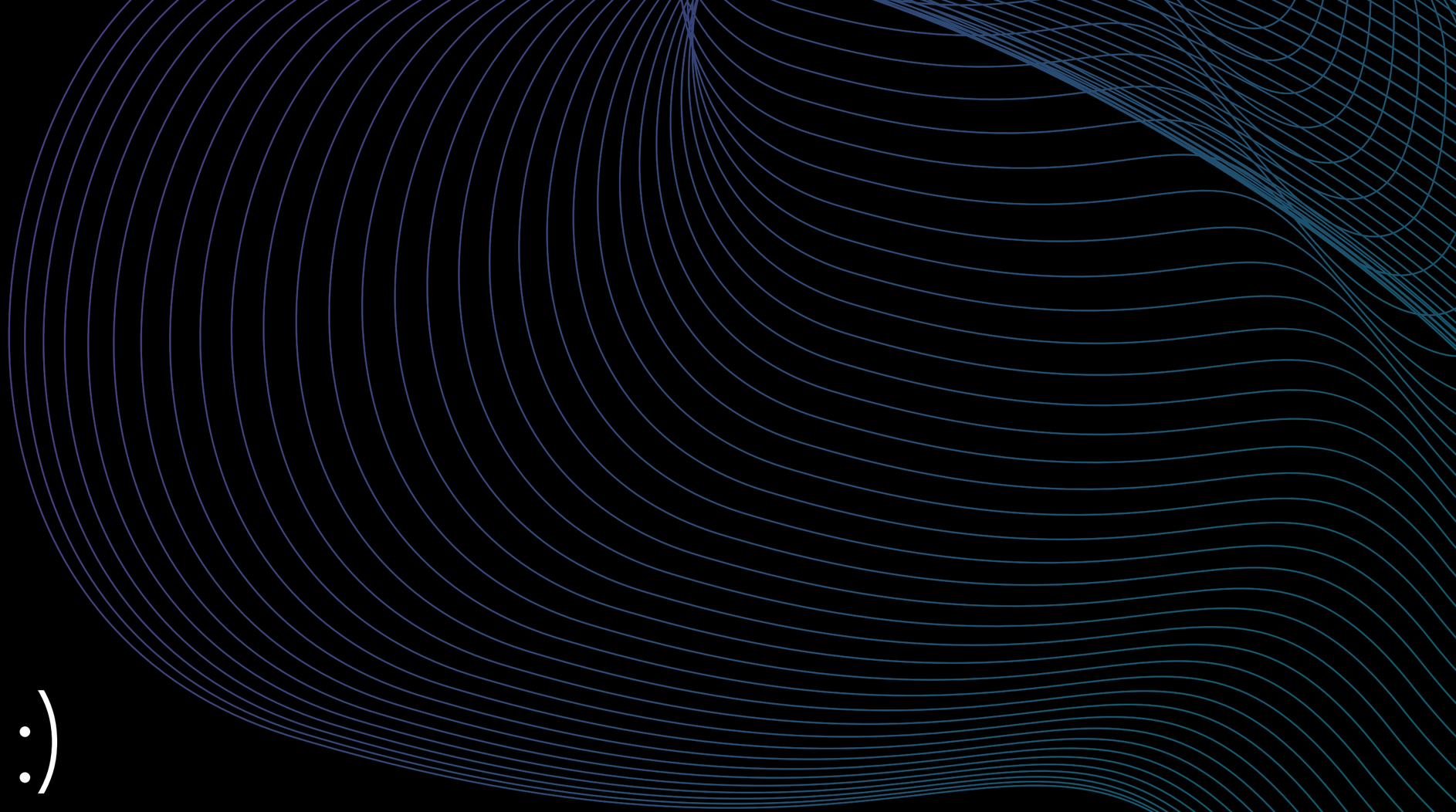
Communities Detection Graph Drawing



As we can see from the graph drawn above, mainly 3 periods are detected during our testing month.
It is possible to notice how the **pattern is very repetitive** also in the colors and in the shape, making us understand that the **contagion patterns are quite repetitive**.
In the graph there are areas where the clusters tend to mix or thicken, this could mean two things:

- There are new positivity coming into play from the outside (other states / trips ...)
- The areas of greatest density are related to larger epidemiological clusters

Red Nodes: First cluster
Green Nodes: Second cluster
Blue Nodes: Third cluster



Thanks :)

Matteo Giaccone - mat. 1745151