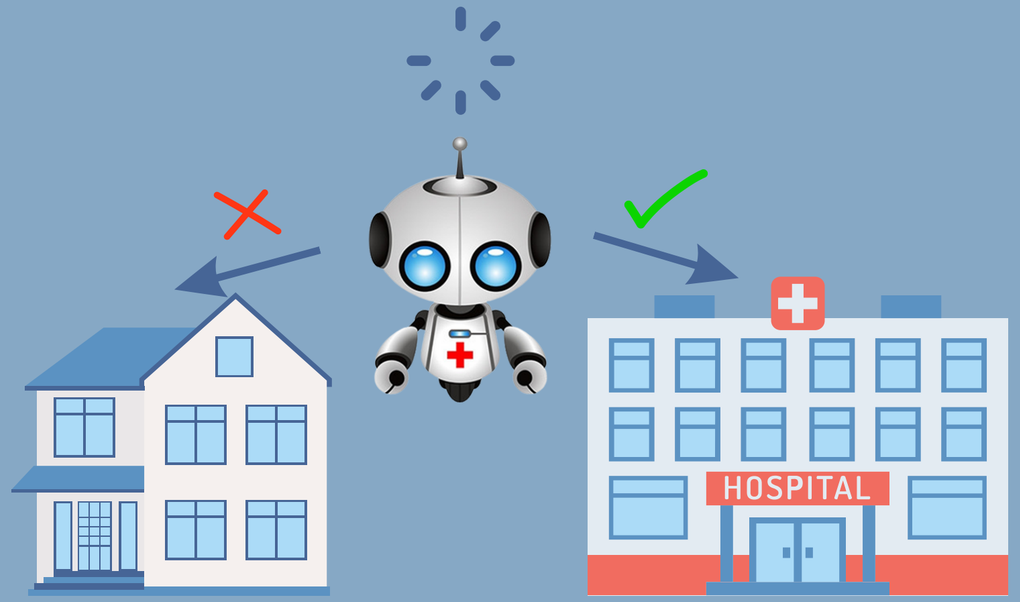
Applying Machine Learning to Intelligent Chatbot for Preventive Care

Sofia Malpique



Supervisor: Prof. Eva Maia

Co-Supervisor: Prof. Rita Ribeiro

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

iCare4NextG

gecad

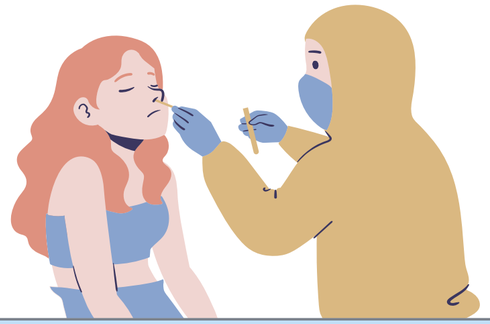
Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development

ISEP
P.PORTO

Project iCare4NextG

- Develop a healthcare chatbot for preventive care.
- Assign intelligence, so it remotely monitors COVID-19 patients.
- Reduce the healthcare professionals' workload.

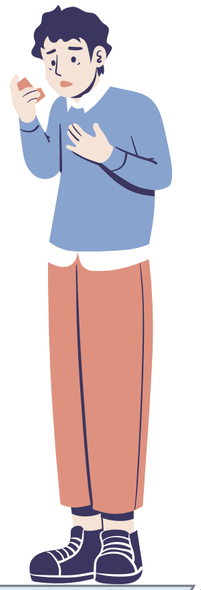
How it started?



COVID-19 became a pandemic

Overloaded Health Facilities and Professionals

COVID-19 Patients sent Home



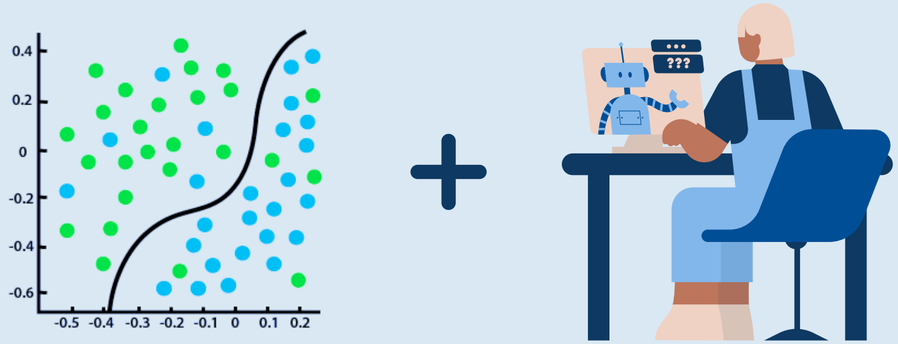
Who will follow them?

Goals

The idea was to develop a tool to support remote patient monitoring.

1. Exploratory Data Analysis

2. Apply ML to Chatbot



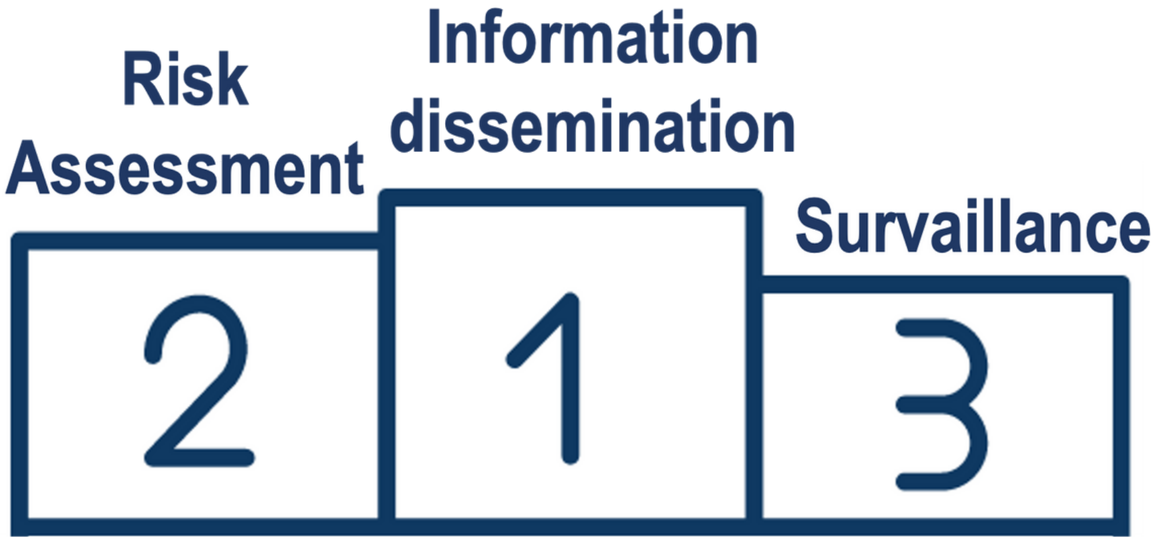
Related work

Most explored ML Applications for COVID-19

Purpose	ML Technique	Data Type
Diagnose Classify Predict	CNN	CT-Scans X-Rays

Related work

Top 3 use cases for COVID-19 Chatbots



Noteworthy datasets

DS4C

- Classification of patient status
- Prediction of the number of days to recover
- Prediction of survival rate

Dec. 2019 - Feb. 2020

nCov2019

- Prediction of survival rate
- Assess risk factor for mortality

Jan. 2020 - Jun. 2020

TriCovB

- Severity assessment with fuzzy systems

Jan. 2020 - Jul. 2021

Looking at the data...

before

DS4C	5,165 x 14
nCov2019	18,527 x 32
TriCovB	1,679,329 x 45

Filtering the data

	before		after
DS4C	5,165 x 14	**Step 1:** • ID columns • High % of Missing Values • Same value	1,633 x 6
nCov2019	18,527 x 32	**Step 2:** • Entries with few values or missing values	39 x 9
TriCovB	1,679,329 x 45		188,383 x 24

Selecting a dataset

	before	after
DS4C ⚠	5,165 x 14	1,633 x 6
nCov2019	18,527 x 32	39 x 9 ⚠
TriCovB	1,679,329 x 45	188,383 x 24

Step 1:

- ID columns
- High % of Missing Values
- Same value

Step 2:

- Entries with few values or missing values

DISCLAIMER: the following results are for informational purposes only, as they might not have been validated from a medical perspective!

Get to know TriCovB better (EDA)

Demographics

Extra Patient Information

Relevant Dates

Comorbidities

TriCovB

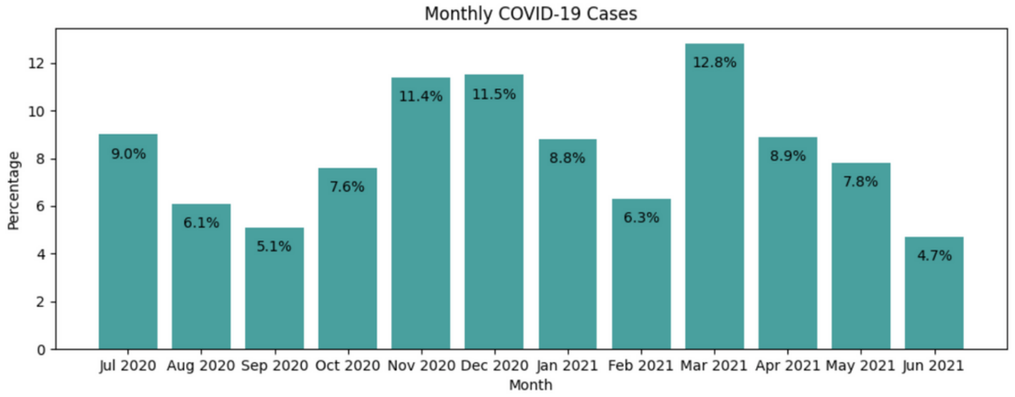
Patient Outcome

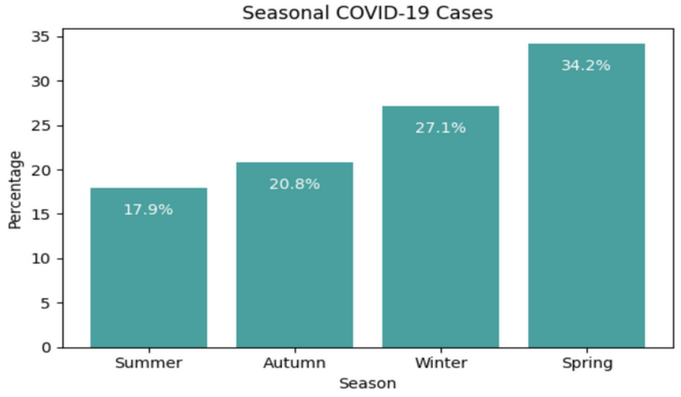
Symptoms

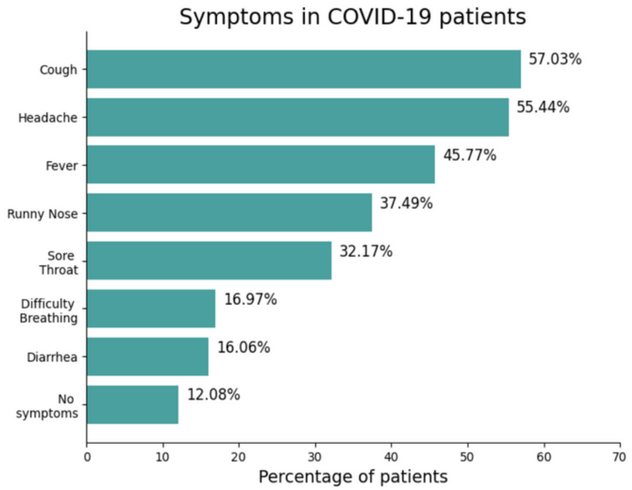
Patient Hospitalization

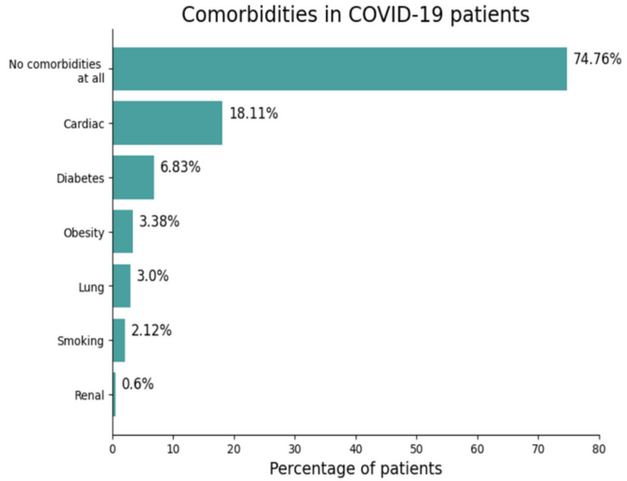
Quarantine Durantion

Univariate Analysis - Highlights





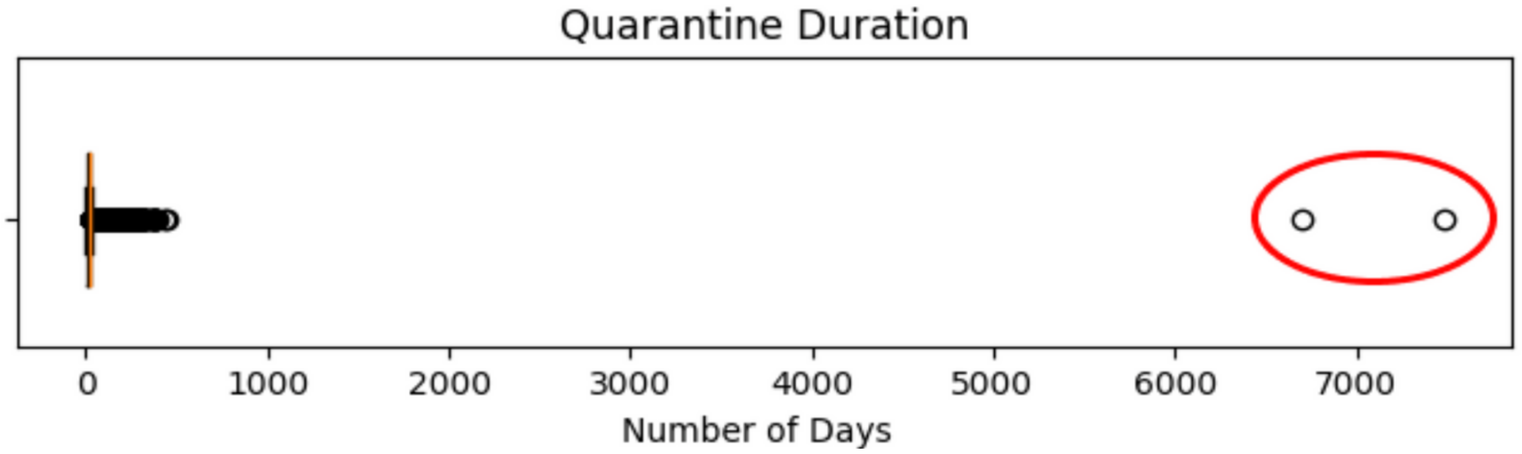


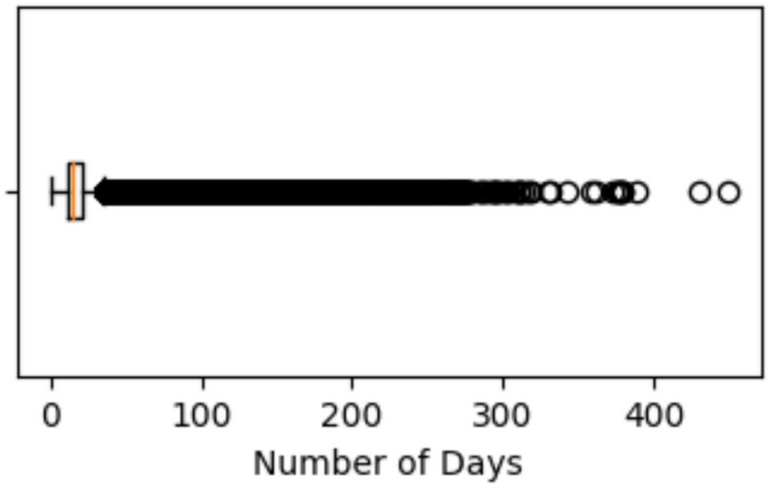


Univariate Analysis - Highlights

Some outliers were detected! And handled...

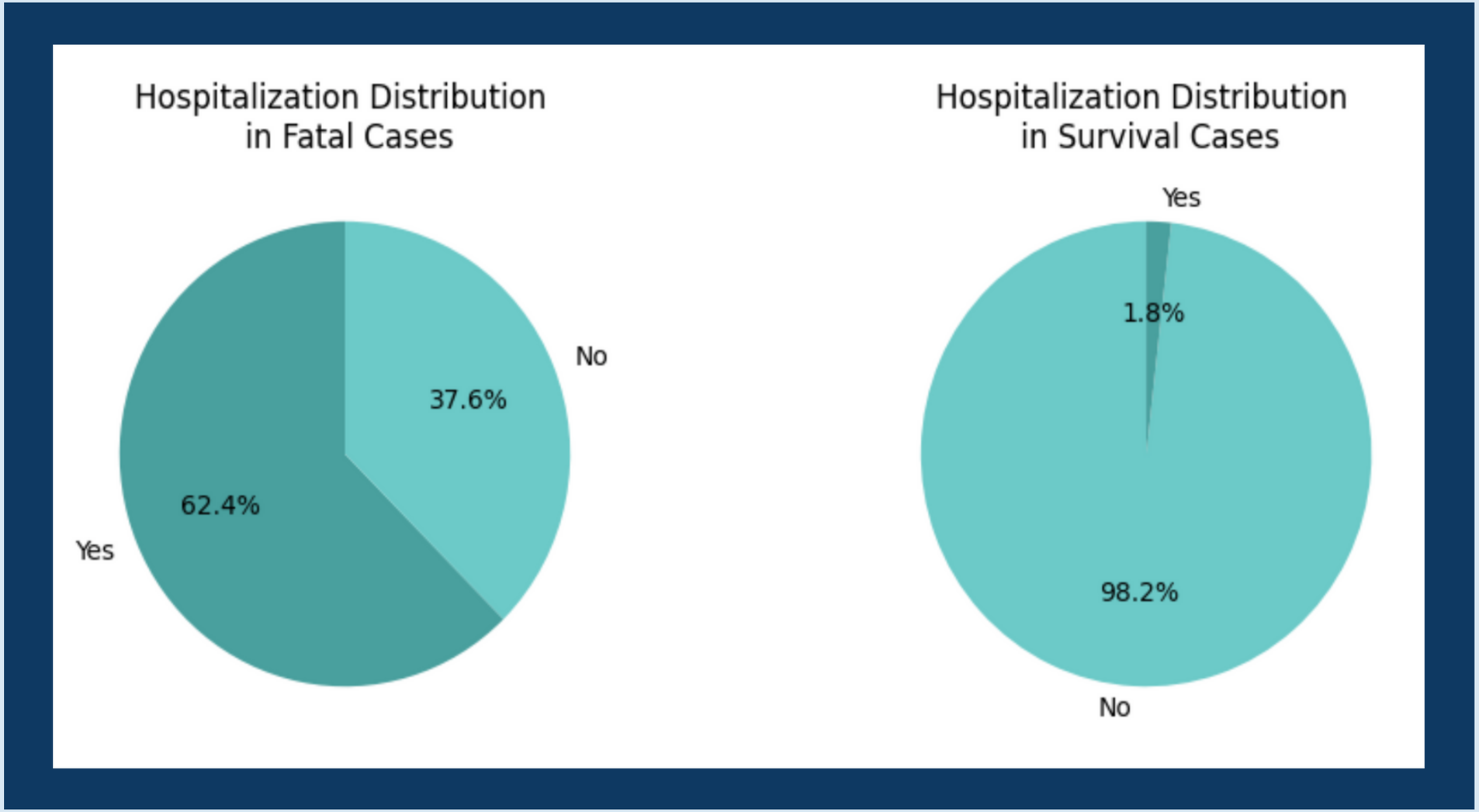
before



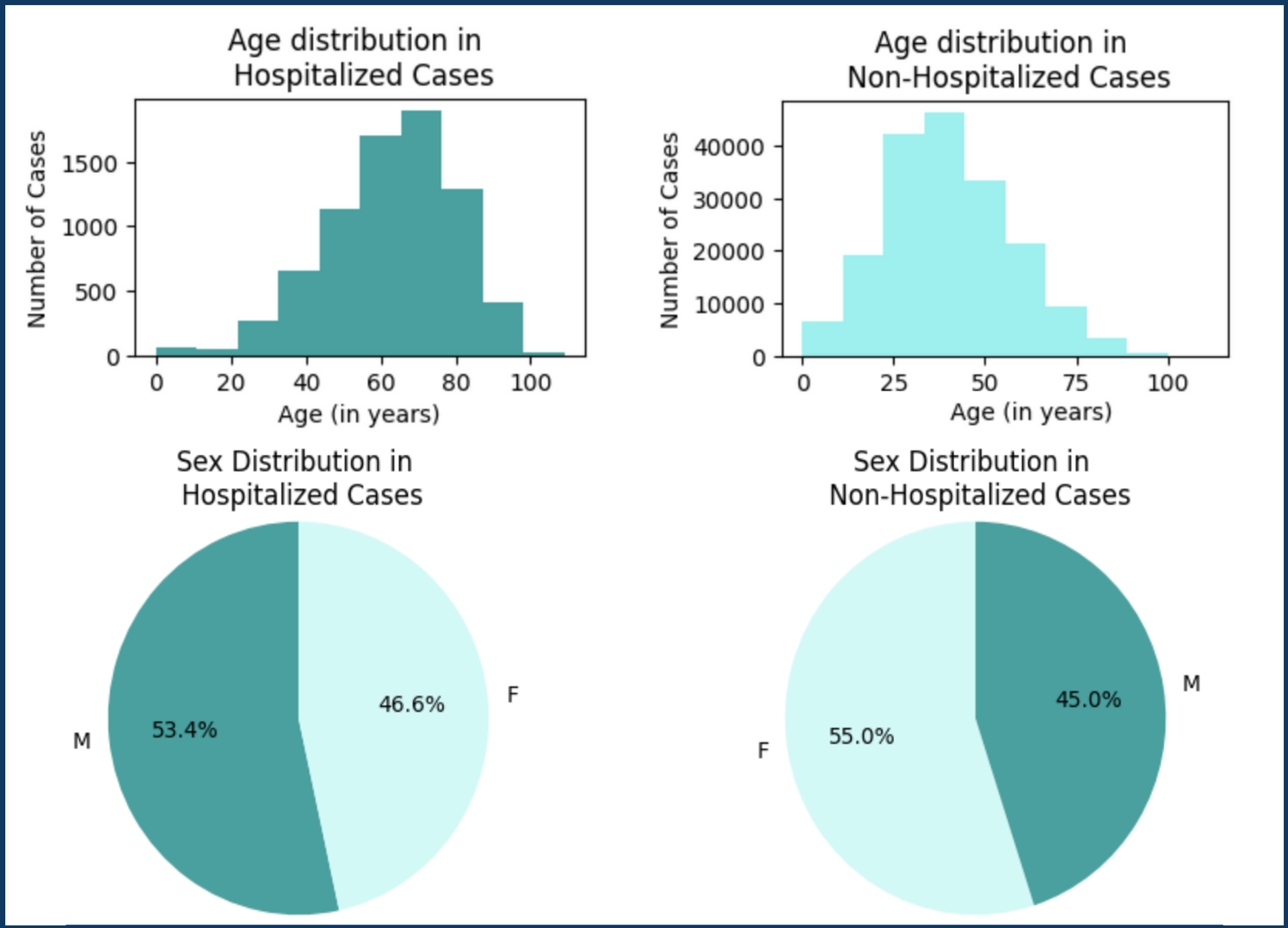


after

Multivariate Analysis - Death & Hospitals



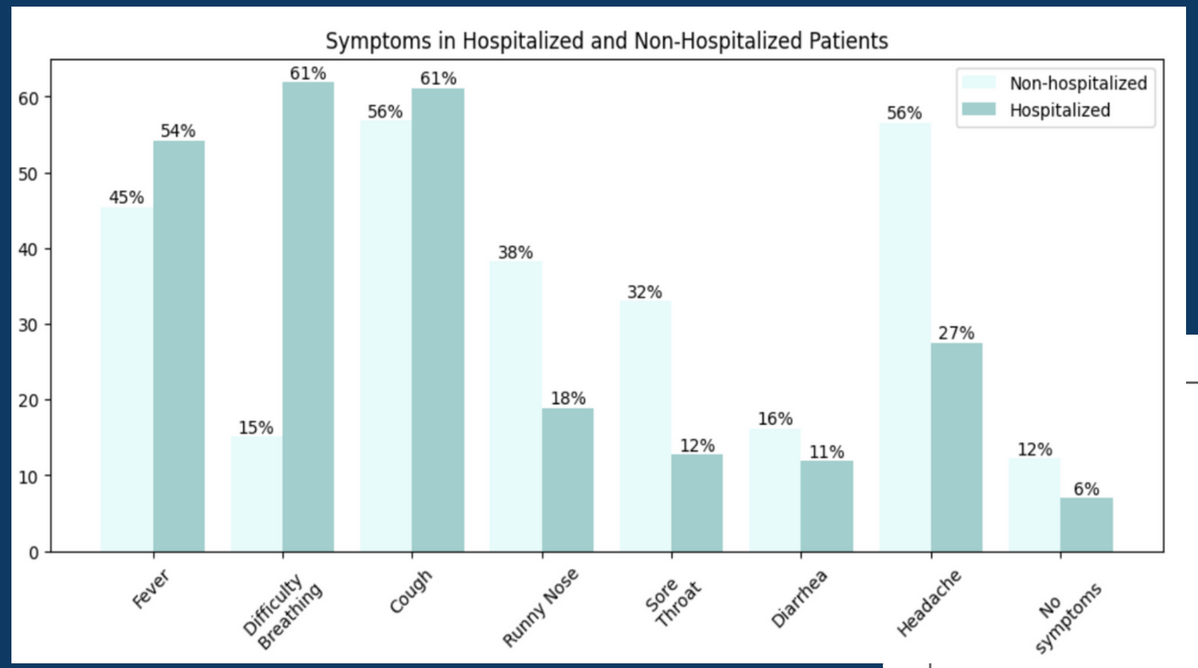
Hospitalized VS Non-hospitalized



Age might be indicative of needing hospitalization.

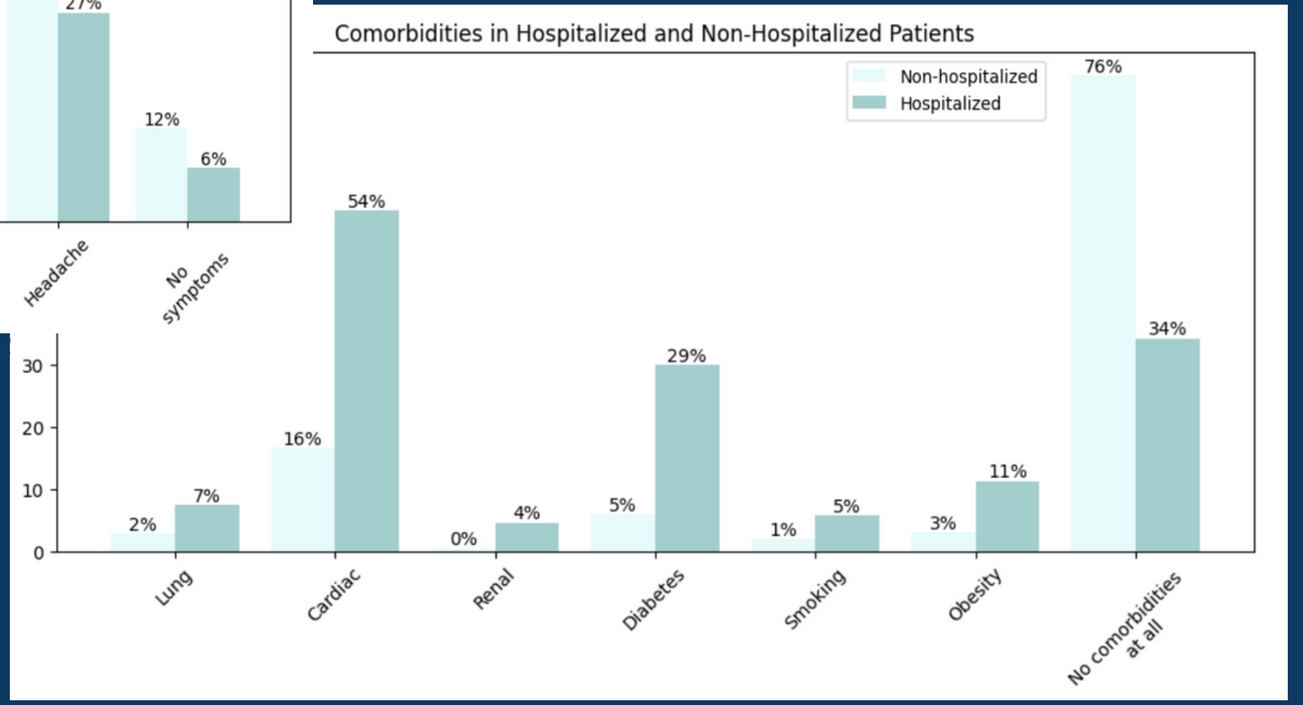
Sex does not seem to be an indicator of hospitalization.

Hospitalized VS Non-hospitalized



Non-hospitalized:

- Runny Nose
- Sore Throat
- Headache
- No comorbidities



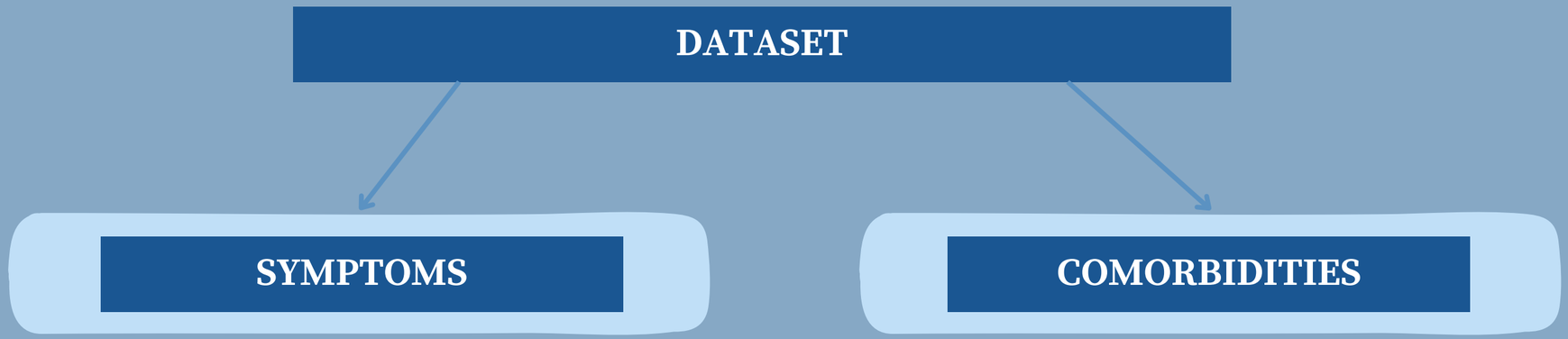
Hospitalized:

- Difficulty Breathing
- Cardiac Problems
- Diabetes
- Obesity

Clustering Analysis

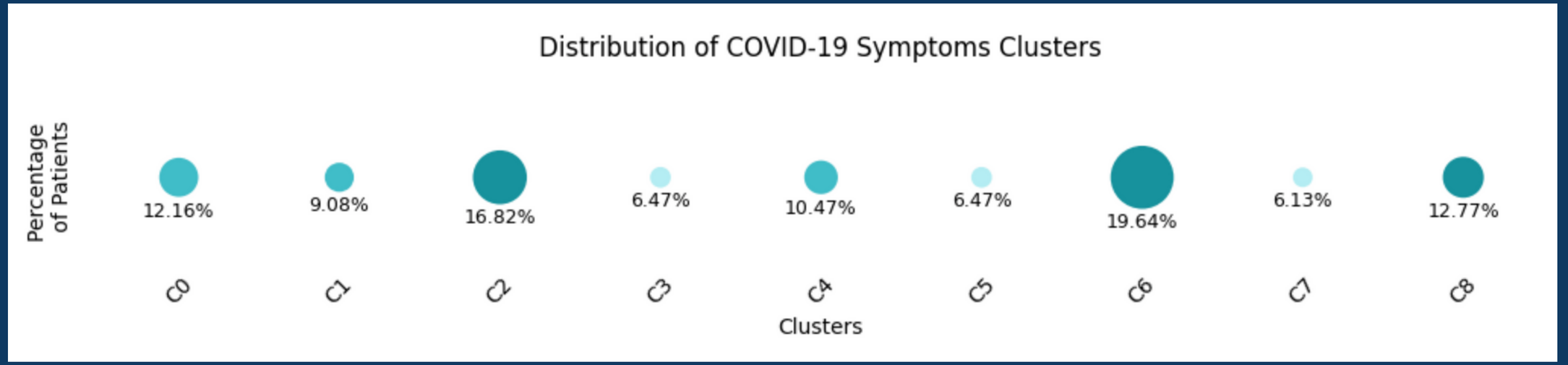
We performed a clustering analysis, using the K-means algorithm, **to find patterns in** both **symptoms** and **comorbidities**.

We split the data into two sub-datasets, one about symptoms and another about comorbidities.



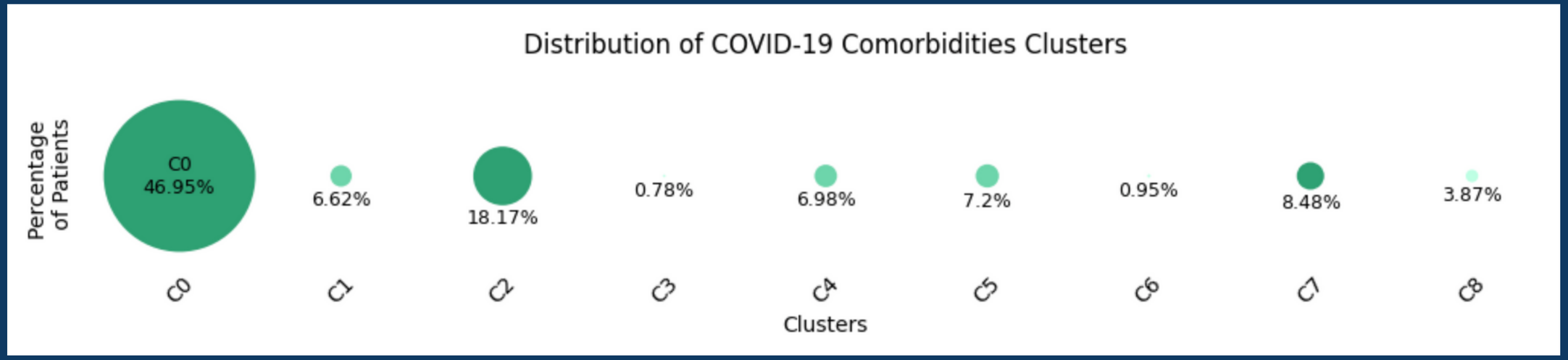
Note: patients without comorbidities were excluded.

Clustering Analysis – Symptoms



Top 3	Present	Absent	Bottom 3	Present	Absent
C6	headaches	runny nose, sore throat	C3	runny nose, sore throat	fever
C2	-	runny nose, sore throat	C5	cough	runny nose, fever, headaches
C8	runny nose	sore throat	C7	runny nose difficulty breathing	-

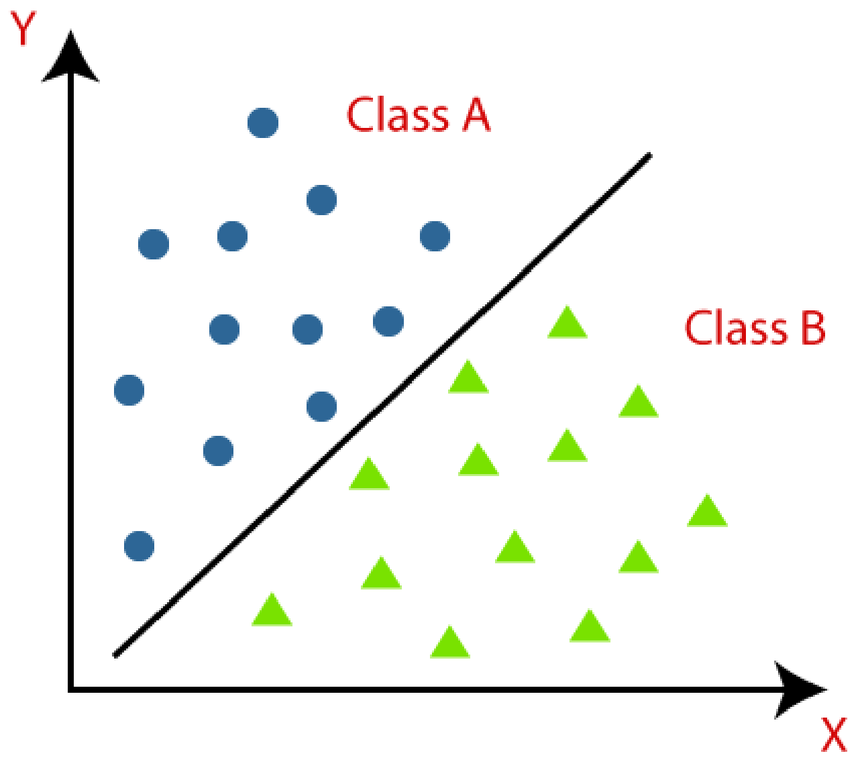
Clustering Analysis – Comorbidities



Top 3	Present	Absent	Bottom 3	Present	Absent
C0	diabetes	cardiovascular, lung, obesity	C8	lung	diabetes
C2	cardiovascular	renal, obesity	C6	renal, diabetes	cardiovascular, lung
C7	obesity	renal	C3	lung, obesity	-

Hospitalization: Classification Task

The Journey



What's the task exactly?

To predict whether a patient will require hospitalization based solely on patient characteristics.

TriCovB ready to train stage

~~Quarantine Duration~~

~~Patient Outcome~~

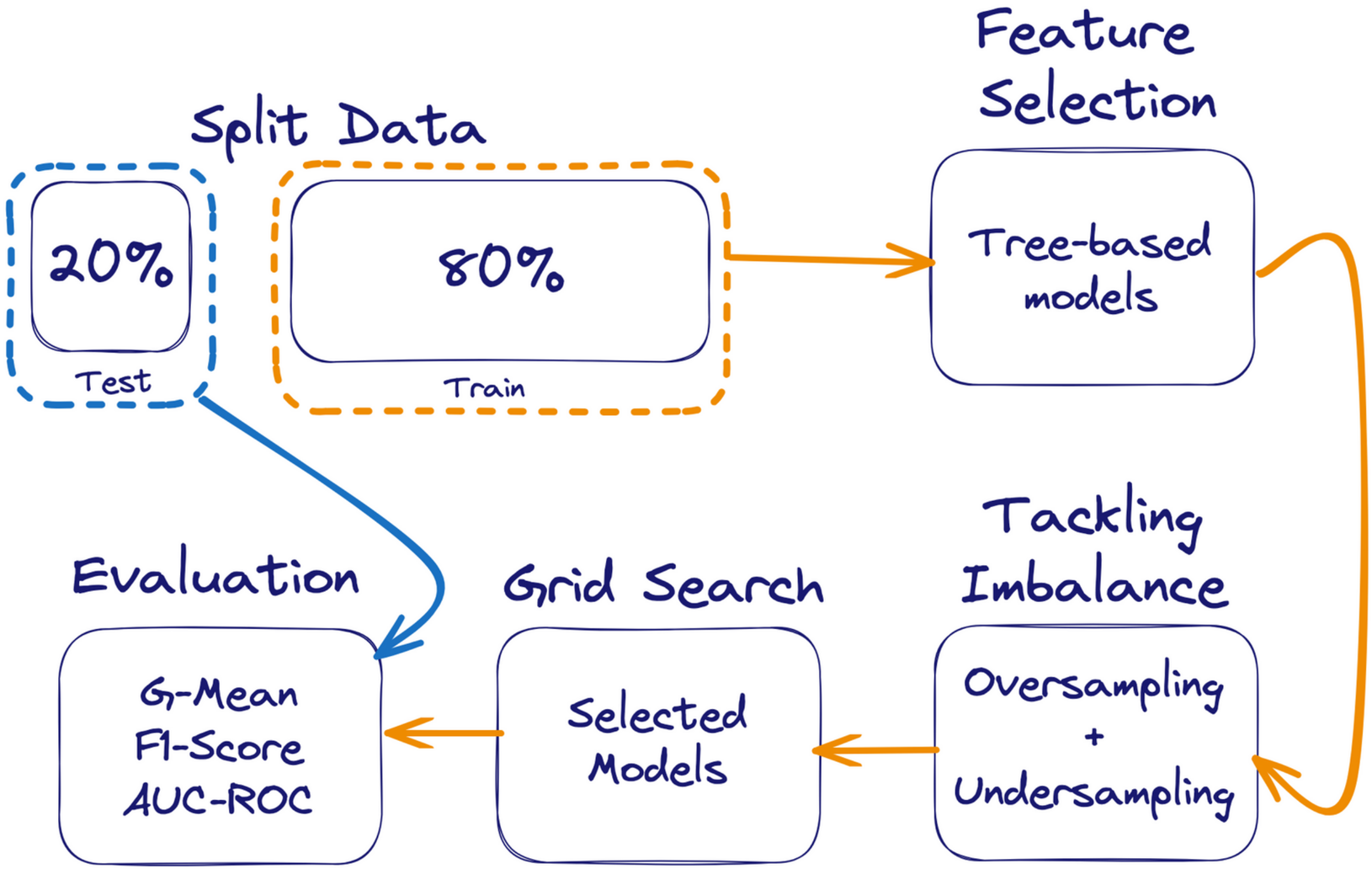
22 Binary Features + 1 Numerical Feature

Independent Variables

Patient Hospitalization
(Binary)

Target Variable

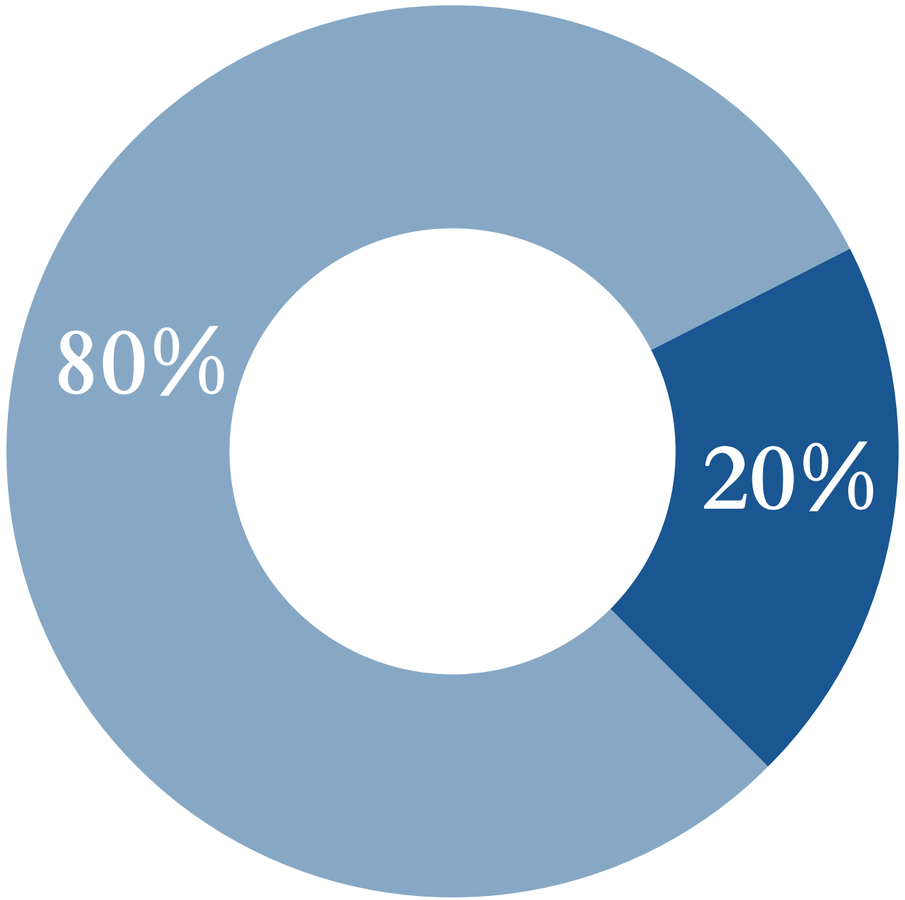
Machine Learning Pipeline



Data Split

Training

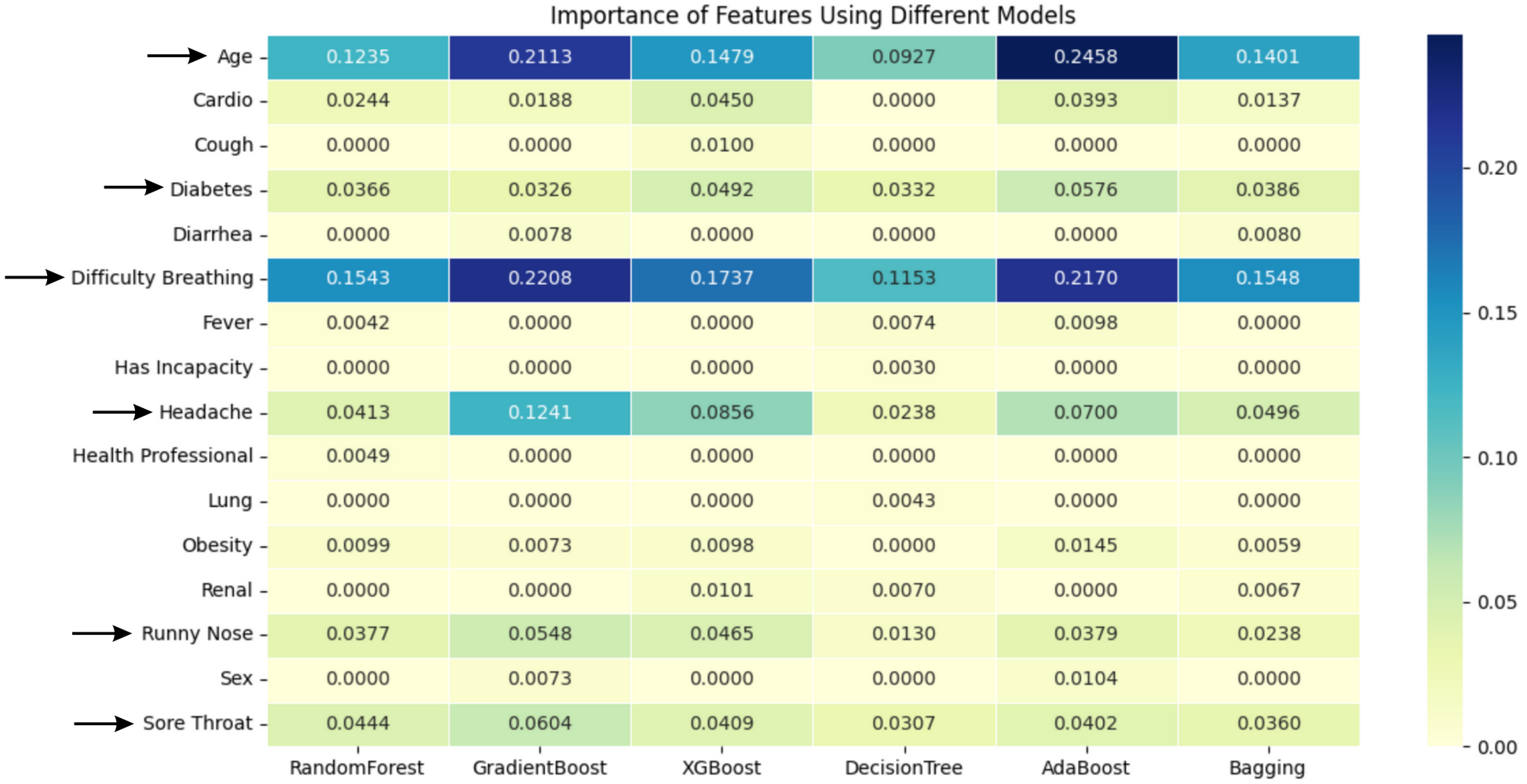
147,920 entries



Test

36,981 entries

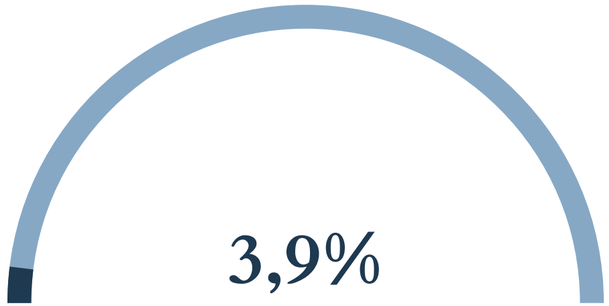
Feature Selection



Now the working dataset only has 16 features.

Challenge: Class imbalance

Priority: Accurately identify patients requiring hospitalization

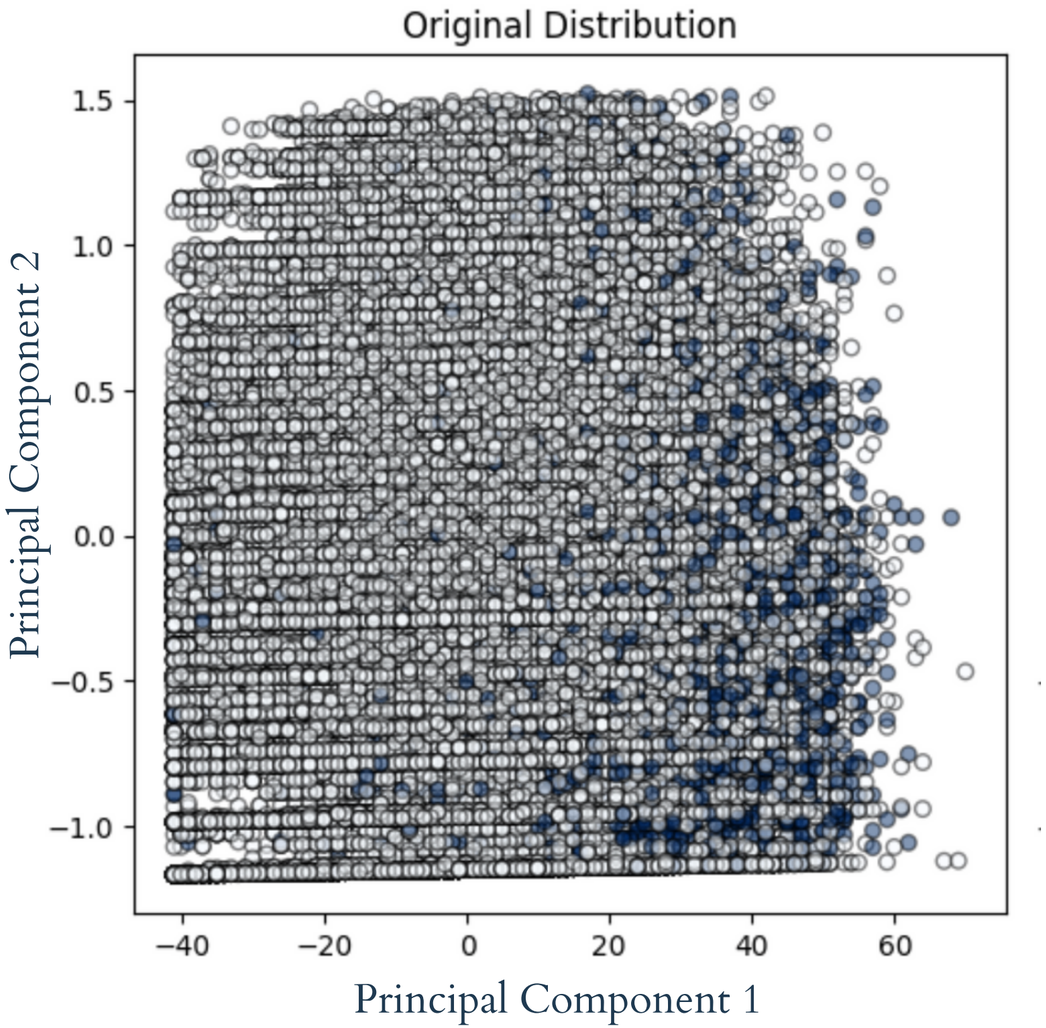


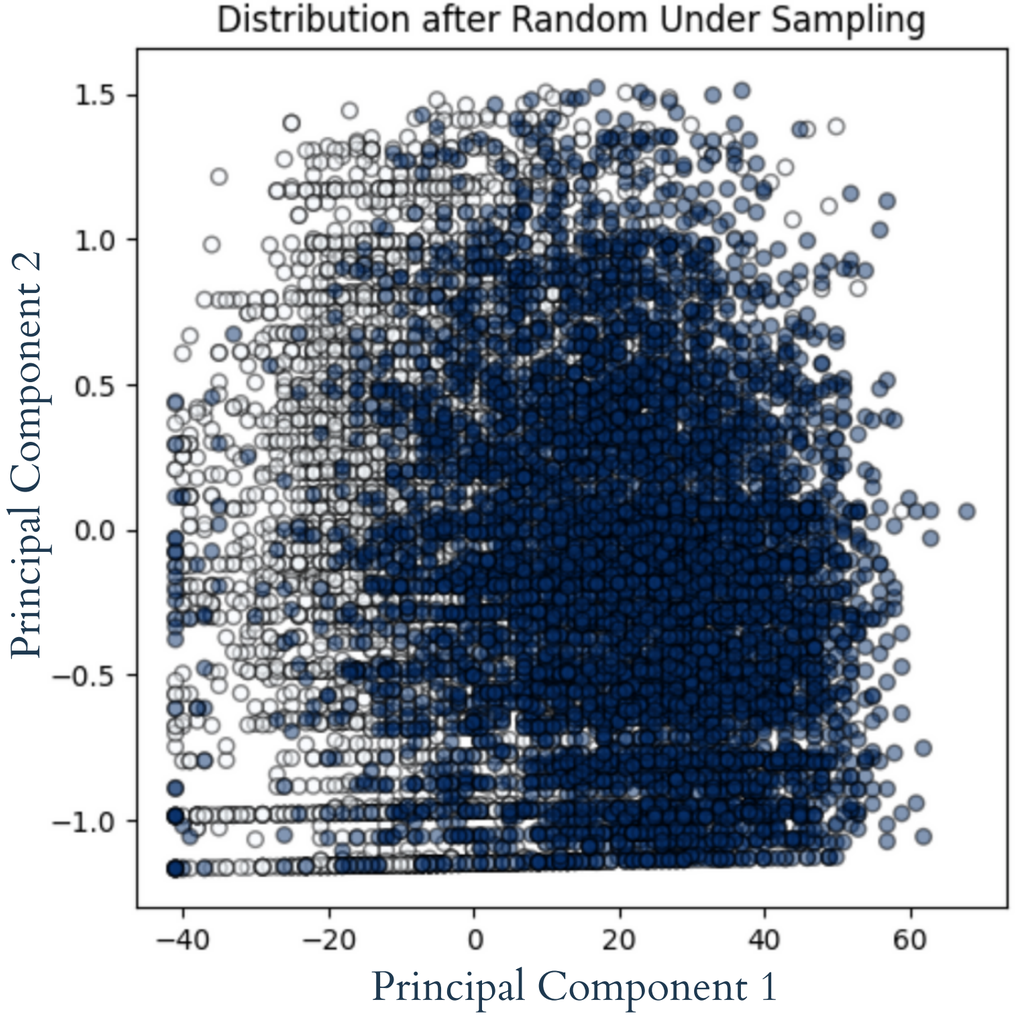
Only 3.9% of entries
representing Positive Class

What will be our approaches?

- Undersampling: **RUS**
  Random Undersampling
- Oversampling: **SMOTE**
  Synthetic Minority Oversampling Technique
- Oversampling + Undersampling: **SMOTEENN**
  Synthetic Minority
  Oversampling Technique +
  Edited Nearest Neighbors

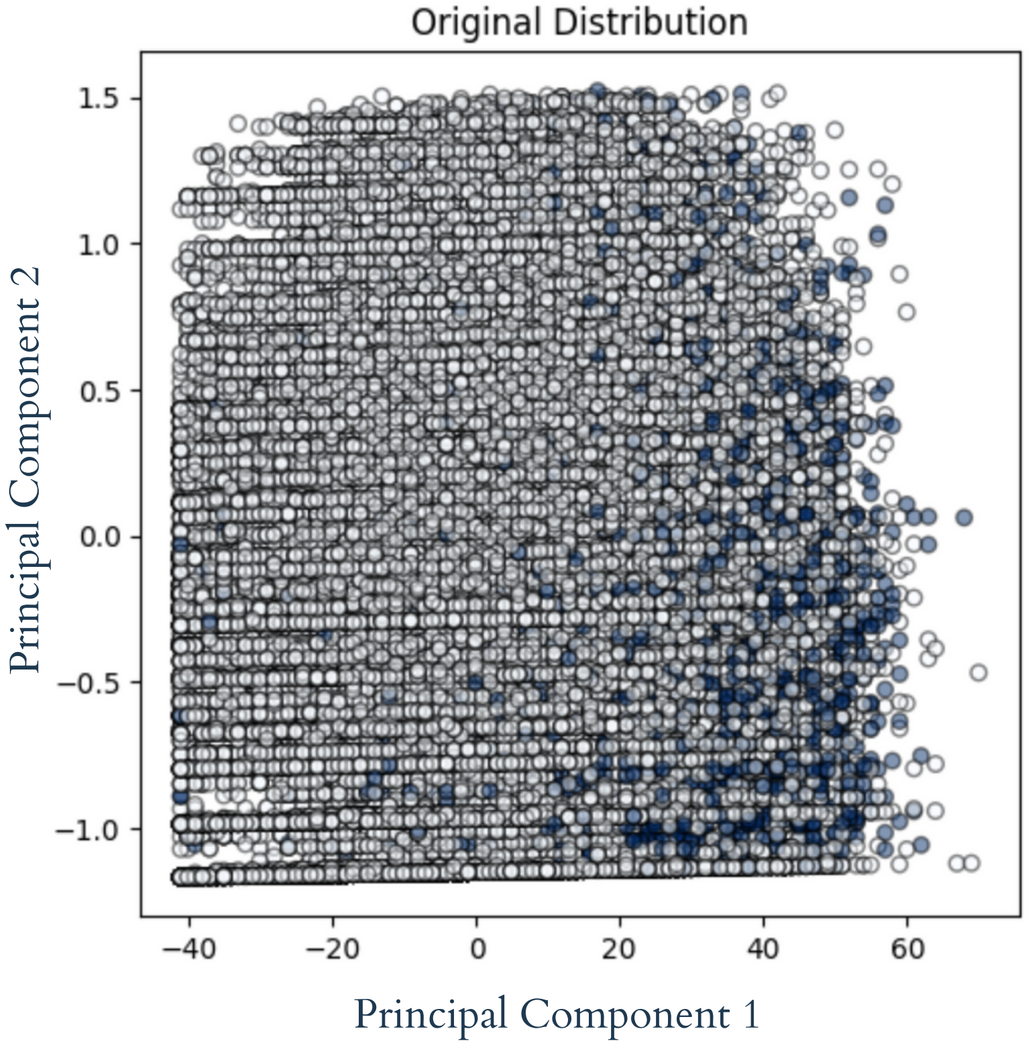
RUS
Random UnserSampling

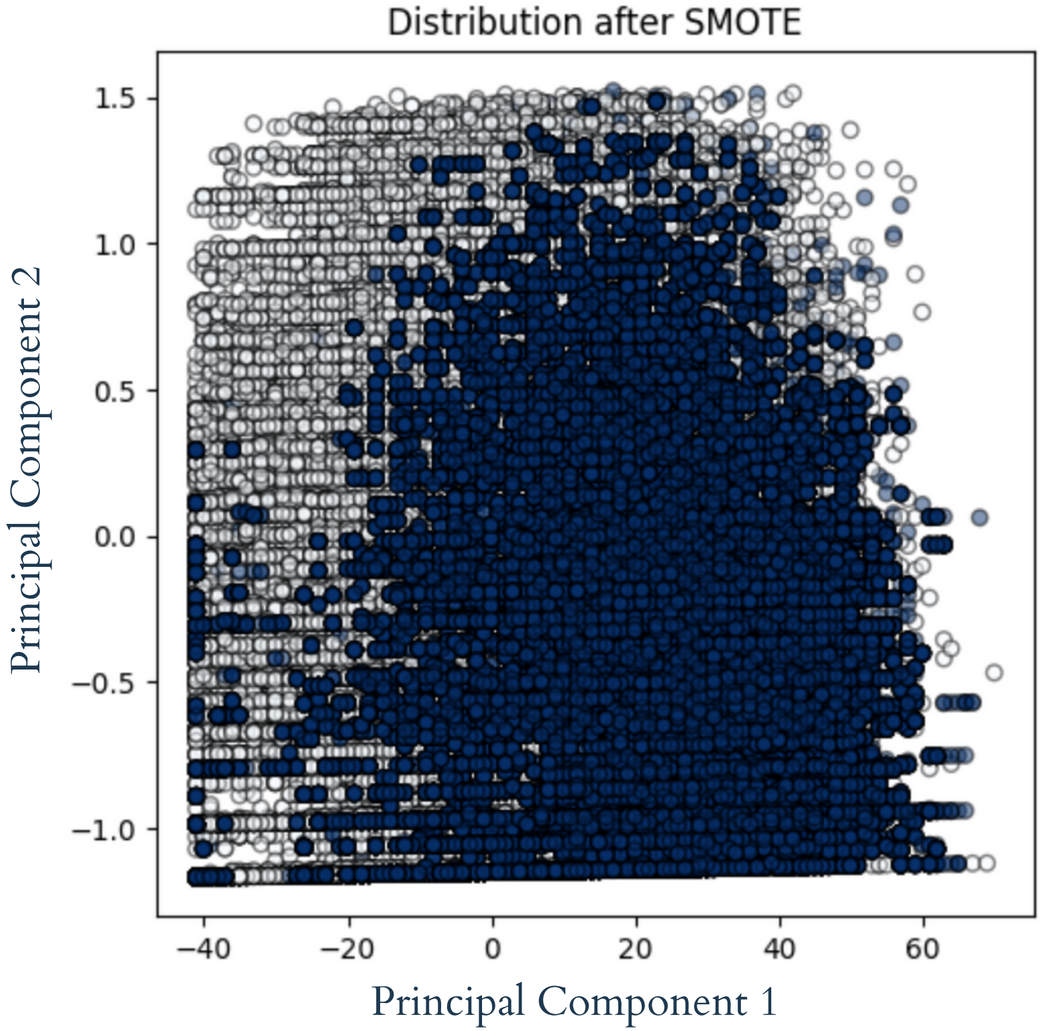




SMOTE

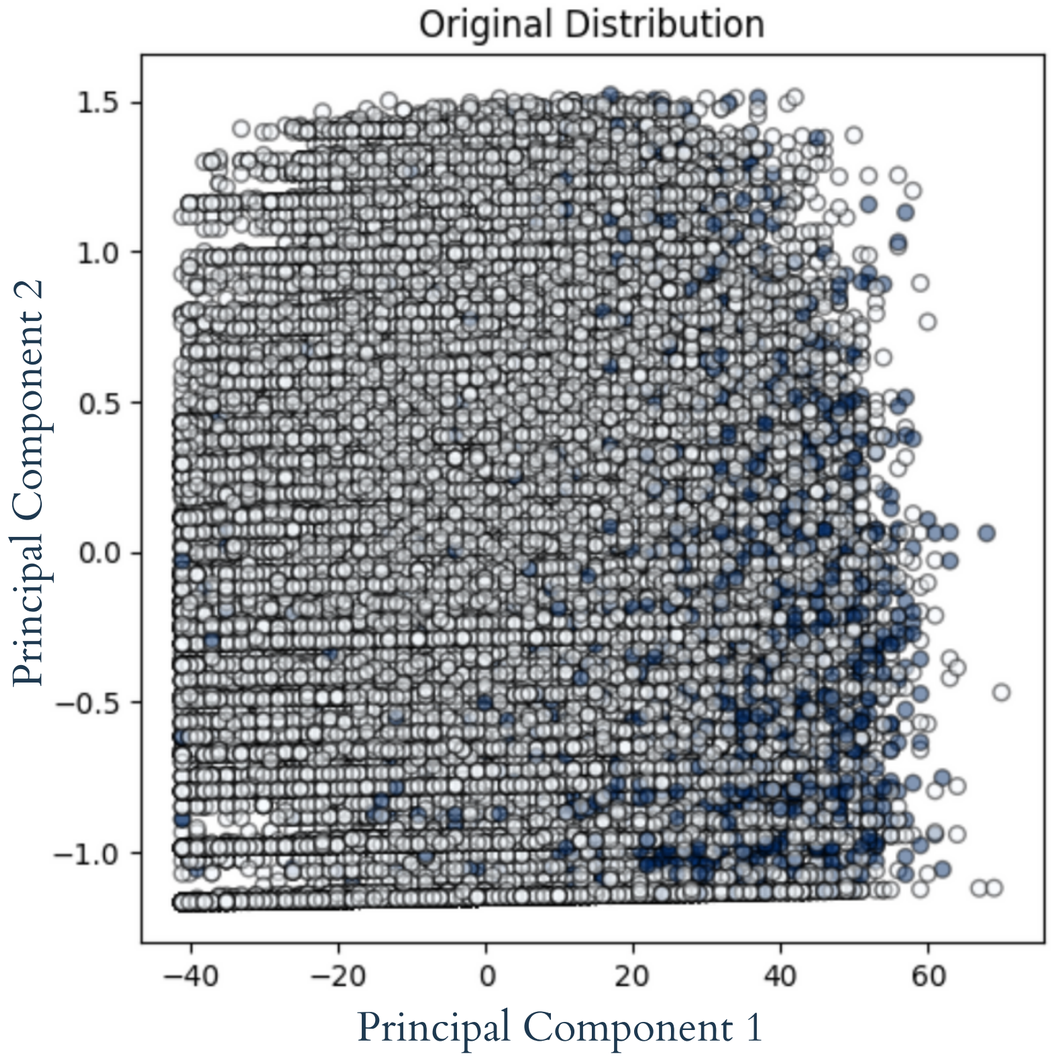
Synthetic Minority Oversampling Tecnhique

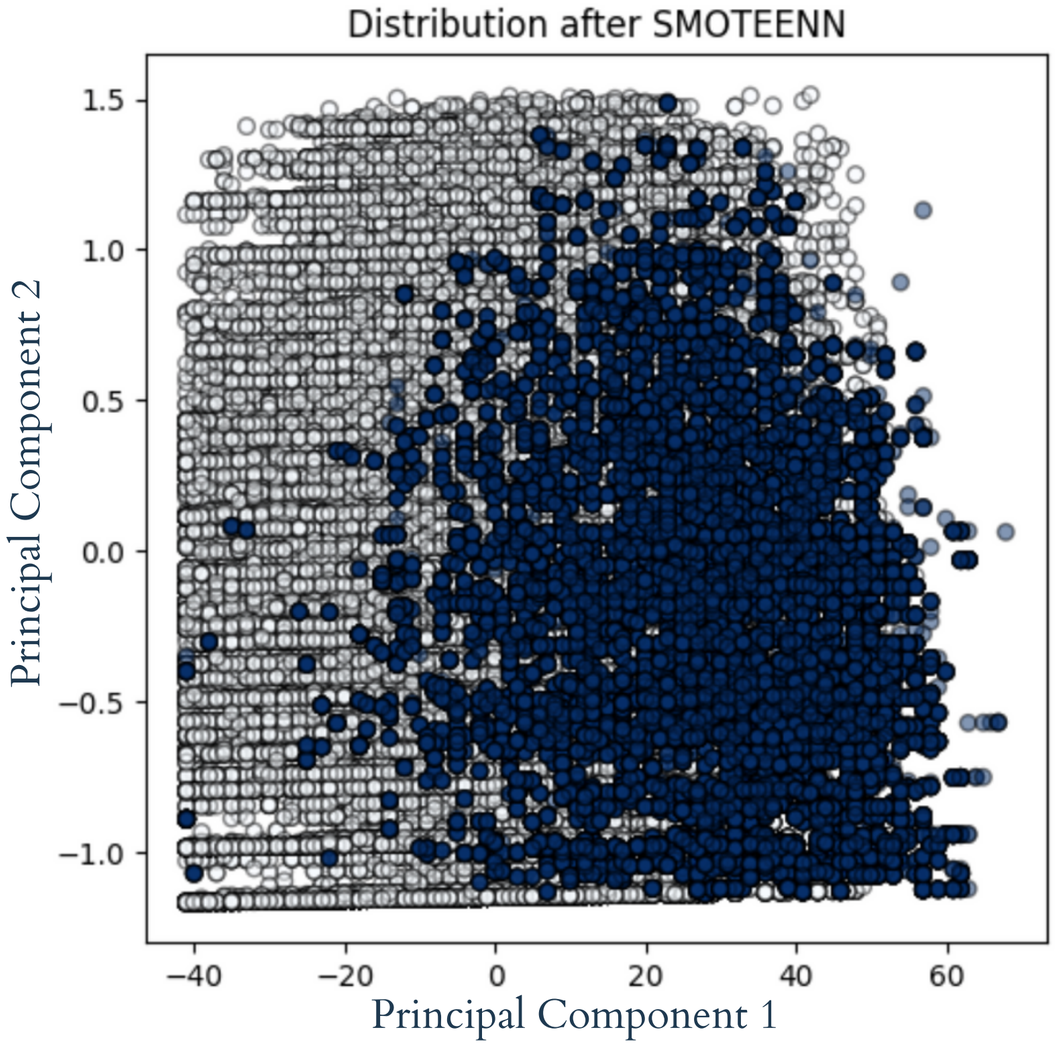




SMOTEENN

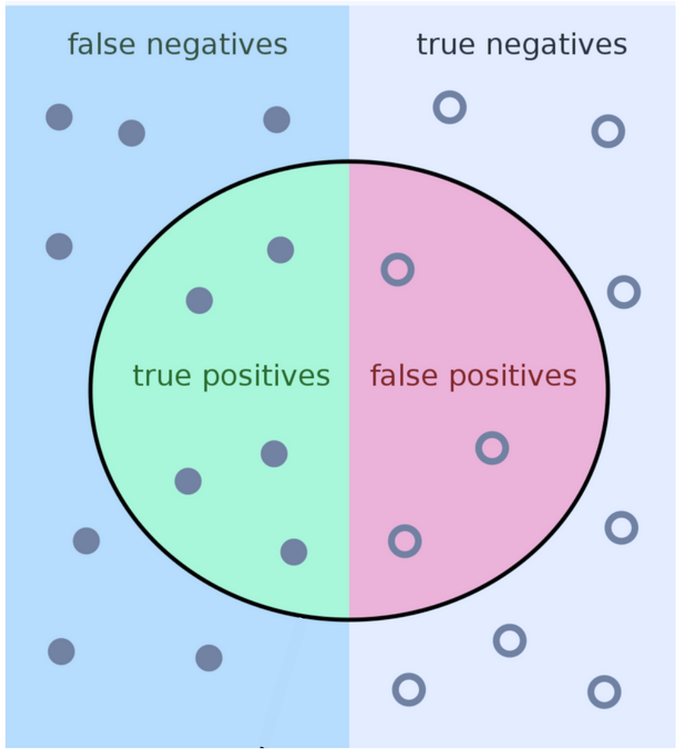
Synthetic Minority Oversampling Tecnhique + Edited Nearest Neighbor





Grid Search with the metric G-Mean

Why the metric G-Mean?



- It evaluates how well a model works for both classes.
- It makes sure the model doesn't just focus on the majority class.

Grid Search

What models are being considered?

Tree-based Models

- Decision Tree
- Gradient Boosting
- Random Forrest
- XGBoost

Emsemble Methods

- Bagging
- AdaBoost

Linear Model

- Logistic Regression

Nearest Neighbors

- K-Nearest Neighbors

What happens with the **original** version of the dataset?

What would happen if we only worked with the original (imbalanced) dataset?

	Decision Tree	K-Nearest Neighbors	Logistic Regression	AdaBoost	Bagging	Gradient Boosting	Random Forest	XGBoost
Best Hyperparameters	max depth = None	num neighbors = 3	C = 1	learning rate = 0.1, num estimators = 100	num estimators = 20	learning rate = 0.1, num estimators = 100	max depth = None, num estimators = 200	learning rate = 0.1, num estimators = 100
G-Mean Score	0.4607	0.4458	0.4379	0.3048	0.4743	0.4696	0.4678	0.4459

Results on the Training Data

RUS

	Decision Tree	K-Nearest Neighbors	Logistic Regression	AdaBoost	Bagging	Gradient Boosting	Random Forest	XGBoost
Best Hyperparameters	max depth = 5	num neighbors = 7	C = 0.1	learning rate = 0.1, num estimators = 100	num estimators = 20	learning rate = 0.1, num estimators = 100	max depth = 10, num estimators = 200	learning rate = 0.1, num estimators = 100
G-Mean Score	0.8109	0.7859	0.8256	0.8234	0.7827	0.8331	0.8226	0.8268

SMOTE

	Decision Tree	K-Nearest Neighbors	Logistic Regression	AdaBoost	Bagging	Gradient Boosting	Random Forest	XGBoost
Best Hyperparameters	max depth = None	num neighbors = 7	C = 10	learning rate = 0.1, num estimators = 100	num estimators = 20	learning rate = 0.1, num estimators = 200	max depth = None, num estimators = 100	learning rate = 0.1, num estimators = 200
G-Mean Score	0.9011	0.7961	0.8332	0.8248	0.9030	0.8411	0.9029	0.8560

SMOTEENN

	Decision Tree	K-Nearest Neighbors	Logistic Regression	AdaBoost	Bagging	Gradient Boosting	Random Forest	XGBoost
Best Hyperparameters	max depth = None	num neighbors = 3	C = 0.01	learning rate = 0.1, num estimators = 100	num estimators = 20	learning rate = 0.1, num estimators = 200	max depth = None, num estimators = 200	learning rate = 0.1, num estimators = 200
G-Mean Score	0.9954	0.9974	0.9246	0.9133	0.9959	0.9328	0.9964	0.9599

Models Evaluation ★★★★☆

After the models have trained with each version and validated on the unseen test dataset!

TABLE 5.11: Best Scores for Each Metric on Different Dataset Versions

Metric	Original	RUS	SMOTE	SMOTEENN
G-Mean	0.4764 (GB)	0.8382 (RF)	0.8216 (AdaBoost)	0.8306 (AdaBoost)
F1-Score	0.3241 (GB)	0.2984 (RF)	0.2955 (GB)	0.3241 (KNN)
AUC-ROC	0.9129 (XGBoost)	0.9132 (GB)	0.8961 (AdaBoost)	0.9035 (GB)

Contributions

COVID-19
Datasets
Presentation

Build a tool to help monitor COVID-19 patients remotely

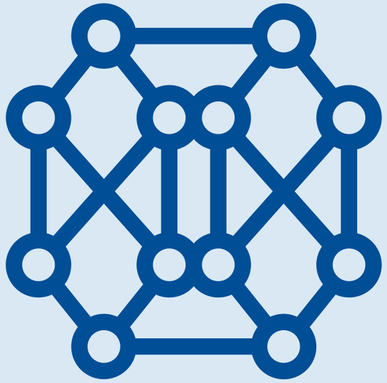
Participation at 5rRACS symposium

5ª Reunião Internacional da Rede Académica das Ciências da Saúde

Future work

Use of Advanced Resampling Techniques

Integration into an Intelligent Chatbot



Experimentation with Different Model Architectures

Thank you!