**Inference for patients with suspected COVID**

# **Prezentarea sistemului**

The project aims to predict whether or not a COVID test is positive or negative, based on the data we provide (gender, age, date when the symptoms debuted, symptoms declared, date of hospitalization, symptoms reported in the hospitalization day, diagnostic and hospitalization signs, traveling history, means of transport, confirmation of contact with another infected person, date of the test result).

# **Implementation**

For the implementation, we used

* [Python](https://docs.python.org/3/library/) (programming language)
* [Scikit](https://scikit-learn.org/stable/) (library for machine learning)
* [Pandas](https://pandas.pydata.org/) (open-source data analysis and manipulation tool)
* [Pickle](https://github.com/python/cpython/blob/3.9/Lib/pickle.py) (module which implements binary protocols for serializing and de-serializing Python object structures)

# **Results**

We used a data set of 5851 entries to train and test the model. We split the entries the following way:

* ⅔ of the entries for training
* ⅓ of the entries for testing

The metrics resulted are:

- accuracy: 0.906

- AUC: 0.639

- confusion\_matrix: [1707, 37

145, 62]

- f1\_score: 0.94

- precision: 0.97

- recall: 0.92

Recall, precision , f1\_score is for patients who are negative.

# **SVM Algorithm**

We encoded the results of the tests:

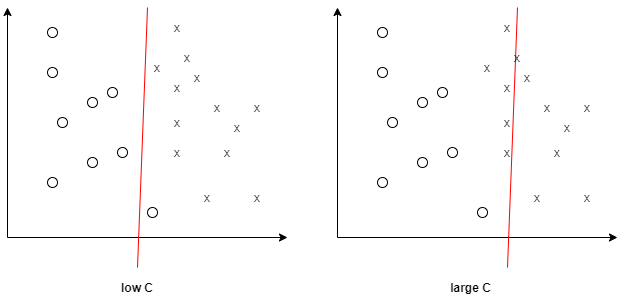
* 0 for NEGATIVE
* 1 for POSITIVE

In order to get good accuracy, the data set used for training should have

a number of positive cases approximately equal to the number of negative cases.

We used the SVM algorithm to separate the negative results from the positive results into two sets, in order to train our model on a balanced data set. This is how the SVM algorithm works:

The zeros and ones are situated in a hyperplane. The first hyperparameter (C) is used to control the error (the lower the hyperparameter, the better the separation).



The other hyperparameter is used for the Gaussian RBF kernel (Gamma). Gamma decides how much curvature we want in a decision boundary (the higher the hyperparameter, the larger the curvature).

