Ben-Gurion University of the Negev

Faculty of Engineering Sciences

The Department of Software and Information Systems Engineering

Introduction to Clinical Data Science

Final Assignment

COVID-19 prediction

Submitted by:

Alex Romanovski

ID: ***

E-mail: romaleks@post.bgu.ac.il

Login on Synapse: Romaleks

ABSTRACT

Testing for Covid-19 is still a major concern in 2020 year. Misuse of testing resources can lead to misdiagnoses for a patient and even to the collapse of the health care system.

The goal of the project is to develop a machine-learning model for predicting the probabilities of patient's positive Covid-19. The developed model in this project was trained on based on clinical information available from patient medical records, namely: *patient age*, *body mass index, diastolic blood pressure, systolic blood pressure, patient height, heart rate, oxygen saturation, sex, non fasting blood glucose levels (mg/dL), patient temperature, patient weight, neutrophils, leukocytes, platelets, fibrin D-dimer FEU, natriuretic peptide B, creatine kinase, lactate dehydrogenase, oxygen , cholesterol* and evaluated using both synthetic data and real EHR data from UW Medicine as part of the "COVID-19 DREAM Challenge" competition [1]. Real data represent clinical records from 9500 patients over the last 10 years collected from multiple large hospitals in the University of Washington Medical System including Harborview Medical Center, UW Medical Center, and Northwest Hospital and Medical Center as well as hundreds of regional clinics across Washington.

The developed model was evaluated using a ROC AUC and a PR AUC metrics and got the following results: ROC AUC=0.6546 and PR AUC=0.0679.

EXPERIMENT

1. Local learning

At the first stage, to design a machine-learning model, I used a dataset [2] with the following features:

Variable	**Description**
`Age`	Patient age
`BMI`	Body mass index
`BP_dias`	Diastolic blood pressure
`BP_diff`	Difference between “*BP_dias*” and “*BP_sys*”
`BP_sys`	Systolic blood pressure
`Height`	Patient height
`HR`	Heart rate
`O2_Saturation`	Oxygen saturation
`SexCode`	Sex: 0-Male, 1-Female
`Sugar_inBlood`	Non fasting blood glucose levels (mg/dL)
`Temp`	Patient temperature
`Weight`	Patient weight

The task is a binary classification, to predict one if the patient has a positive Covid-19 and zero otherwise. The problem was in class imbalance (Fig.1). Even if the model will always predict zero, it will get a high score for accuracy.

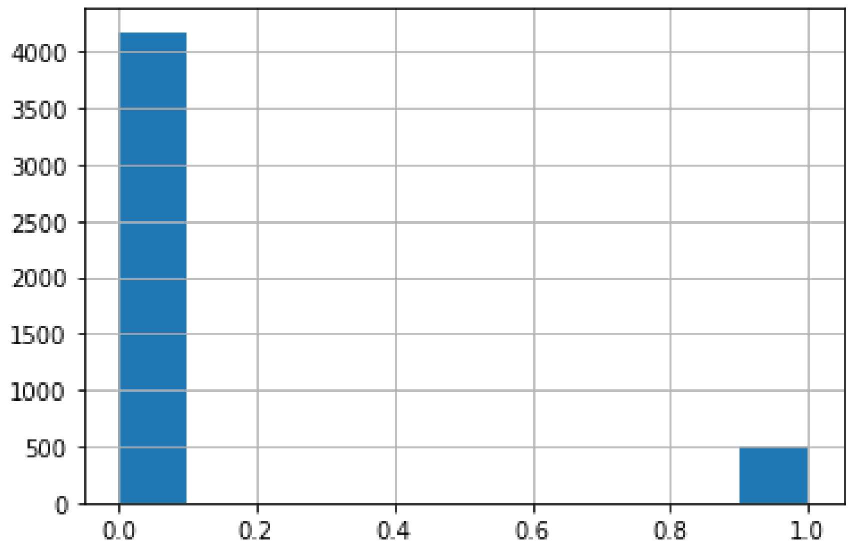


Fig.1 Class distribution.

As a base learner, I choose BalancedRandomForestClassifier [3] from the library “imblearn” [4], which showed better results than XGboost [5], [6] and Catboost [7]. BalancedRandomForestClassifier does not work with missing values (NaN), as well as categorical features, so preprocessing was necessary. Therefore, label encoder was applied for all categorical features and replaced Nan with -999. Dataset was splitted by 66% for training and 33% for validation with preservation of distribution target value (Fig. 2).

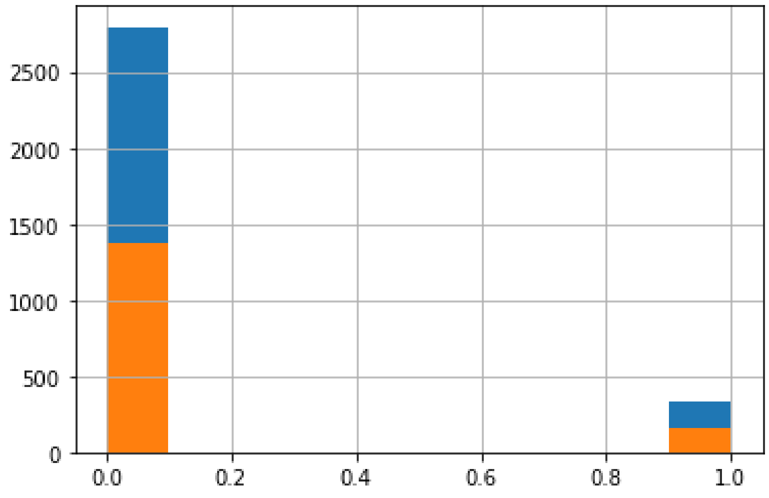


Fig.2 Target distribution in both splits.

The following results were obtained during validation:

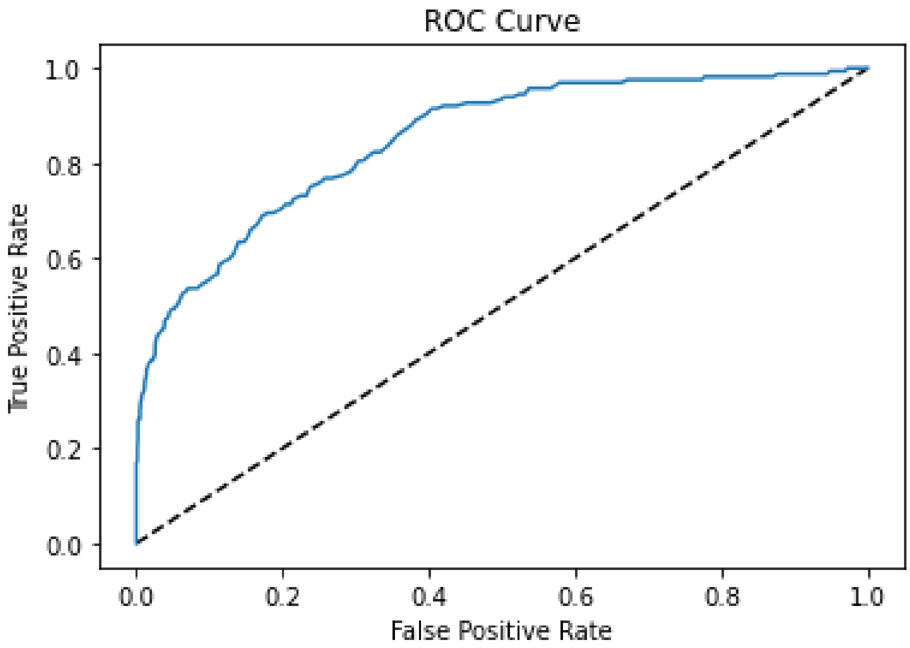
a. Confusion matrix

	precision	recall	F1-score	support
0	0.96	0.74	0.84	1378
1	0.26	0.77	0.39	164
Accuracy			0.74	1542
Macro avg	0.61	0.75	0.61	1542
Weighted avg	0.89	0.74	0.79	1542

Table 1. Confusion matrix

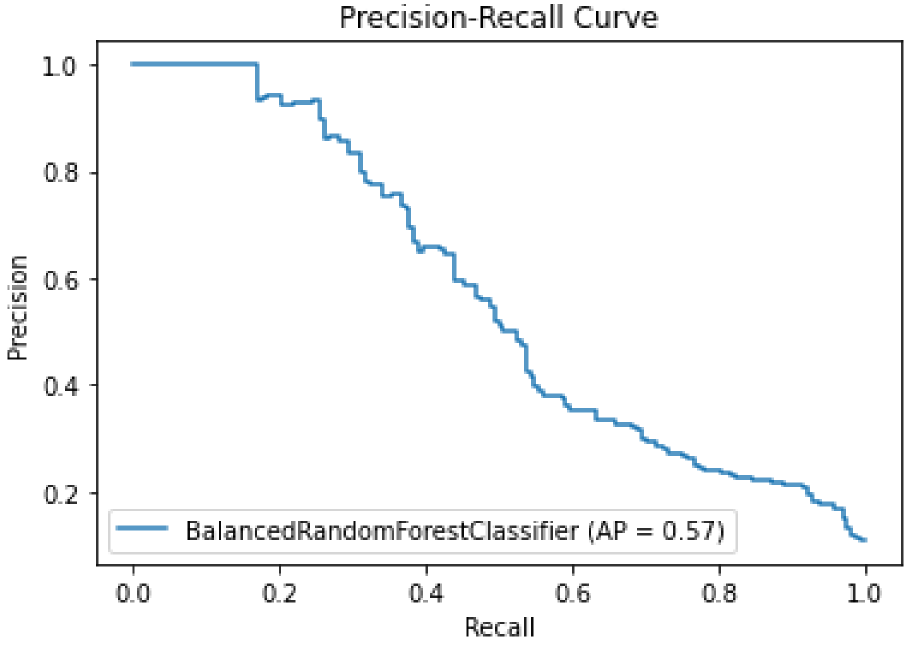
I got a good recall (Table 1) compared to other models. It is important for me to identify all patients positive Covid-19, so I give preference to this metric.

b. ROC AUC



- **ROC AUC**: 0.85

c. PR AUC



- **AUC PR**: 0.57

2. Model validation on Synapse data.

At this stage, I submit a pre-trained model from previous stage for evaluation on Synapse Unseen data. For this purpose, I created a function "*data extractor*" that transforms the data on the Synapse server into a dataset suitable for predicting my model.

For the data extractor to work correctly, it was necessary to find the location of the current features from our dataset in the Synapse data. Table 2 shows the location of the features in Synapse data.

Variable	**Description**	**Location in Synapse data: table “*measurement*” (concept_id)**
`age`	Patient age	No
`BMI`	Body mass index	No
`BP_dias`	Diastolic blood pressure	3012888
`BP_diff`	Difference between “*BP_dias*” and “*BP_sys*”	No
`BP_sys`	Systolic blood pressure	3004249
`Height`	Patient height	3036277
`HR`	Heart rate	3027018
`O2_Saturation`	Oxygen saturation	3016502
`SexCode`	Sex: 0-Male, 1-Female	No
`Sugar_inBlood`	Non fasting blood glucose levels (mg/dL)	3004501
`Temp`	Patient temperature	3020891
`Weight`	Patient weight	3025315

Table 2. Location in Synapse data: table “*measurement*” (concept_id)

Most of the features have been found in the table “*measurement*”. The following features required additional processing:

- **Age -** To find the age, I used the difference between the date of birth from the table “*person*” column “*year_of_birth*” and the current date.
- **BMI -** To calculate this value, I used the following formula: $BMI = \frac{Weight}{Height^2}$
- **BP_diff -** To find this parameter, I used the difference between the “*BP_dias*” and “*BP_sys*”.

- **SexCode -** I found this value in the table "*person*", column "*gender_source_value*"

It is also worth focusing on the processing of repeated values (for example, for the same patient, several values of the same feature). In this case, I consider only the most recent value of all (the most recent date).

After submission (**submission id 9707026**), my model got the following evaluation:

- **PR AUC**: 0.036
- **ROC AUC**: 0.4506

3. Model training and validation on Synapse data.

At this stage, my model will not only be evaluated on the Synapse data, but also trained using previous features (**submission id 9707335**).

- **PR AUC**: 0.056
- **ROC AUC**: 0.6345

Training on synapse data allowed me to improve the model results.

4. Model with additional features training and validation on Synapse data.

In this part, I wanted to improve on the previous model by adding additional features. To design the model, I relied on synthetic data also provided by the synapse.

First, the model was trained on measurement data from synthetic data using three-fold cross-validation and a list of feature importances was compiled according to their weight in the model. As additional features, I choose following top-9 features from the list:

Rank	Measurement value	Concept id
1	Neutrophils	3013650
2	Leukocytes	3000905
3	Platelets	3024929
4	Fibrin D-dimer FEU	42870366
5	Natriuretic peptide B	3011960
6	Creatine kinase	3007220
7	Lactate dehydrogenase	3022250
8	Oxygen	3027801
9	Cholesterol	3027114

Training on the Synapse data and adding new features to the previous ones allowed to significantly increase the model evaluation. After submission (**submission id 9707028**), my model got the following evaluation:

- **PR AUC**: 0.0679
- **ROC AUC**: 0.6546

Additional features allowed me to significantly increase the PR AUC from 0.056 to 0.0679.

DISCUSSION

In the course of the work, several models were trained and their quality was evaluated by two metrics: ROC AUC and PR AUC. The first model was trained on real but own data and, when evaluated on real Synapse data, showed the worst result (submission id 9707026, PR AUC: 0.036 and ROC AUC: 0.4506). Training on synapse data has already allowed me to significantly improve the model (submission id 9707335, PR AUC: 0.056 and ROC AUC: 0.6345). The use of the additional features also affected the quality of the model, despite the fact that the ROC AUC did not increase significantly, I significantly increased the PR AUC (submission id 9707028, PR AUC: 0.0679 and ROC AUC: 0.6546). This model shows good results, but nevertheless, at this stage, its use alone cannot replace other tests.

As a future work, I would like to note the opportunity to work with other types of real data. In this project, the emphasis was on measurement data, but perhaps adding symptom data will increase the results.

REFFERENCE

1. EHR DREAM Challenge COVID-19. https://www.synapse.org/#!Synapse:syn21849255/wiki/601865
2. Data.csv (available in the folder with the project)
3. Chen, Chao, Andy Liaw, and Leo Breiman. “Using random forest to learn imbalanced data.” University of California, Berkeley 110 (2004): 1-12.

4. https://imbalanced-learn.org/stable/generated/imblearn.ensemble.BalancedRandomForestClassifier.html#imblearn.ensemble.BalancedRandomForestClassifier
5. Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. InProceedings of the 22Nd ACM SIGKDD InternationalConference on Knowledge Discovery and Data Mining, pages 785–794. ACM, 2016.
6. https://xgboost.ai/
7. https://catboost.ai/