Clinical Data Analysis to assess ICU Admission (COVID-19)

Kimia Noorbakhsh
Department of Computer Engineering
Sharif University of Technology
knoorbakhsh@ce.sharif.edu

I. Introduction

The hospitals worldwide have been facing the urgent consequences of Covid-19 transmissions and requested ICU beds above the usual capacity. This project is developed to advise the Hospital team to be prepared and obtain an accurate prediction of future ICU admissions for confirmed COVID-19 cases.

II. Dataset

The dataset can be downloaded from [1]. The data has four parts:

1) Patient demographic information (count: 3)
2) Patient previous grouped diseases (count: 9)
3) Blood results (count: 36)
4) Vital signs (count: 6)

There are 54 features, expanded when pertinent to the mean, median, max, min, diff, and relative diff. According to Min Max Scaler, data has been cleaned and scaled by column to fit between -1 and 1 by the Sirio-Libanes Hospital team.

There are also 2 study variables:

1) ICU admission -> Target (Yes | No) - (0,1)
2) Time Window - WINDOW - ['0-2', '2-4', '6-12', 'Above-12')

As stated in data description section, a model with only the "0-2 window" is more clinically relevant. Therefore, we will use the data reported in this period for our predictions. Moreover, as warned by the authors as well, we should be aware when the target variable (ICU) is 1, since we cannot be sure about the events' order (see figure 1).

III. Exploratory Data Analysis and Feature Engineering

We do this task in 3 different stages:

1) Data Cleaning and Early Prediction Scenario
2) Visualization
3) Feature Engineering

A. Data Cleaning and Early Prediction Scenario

First of all, a column named 'tags' at the end of the dataset shows different kinds of diseases that each patient has been suffering in the past or even now. We encode this column into six distinct columns for the studied disease to better represent them in our data.

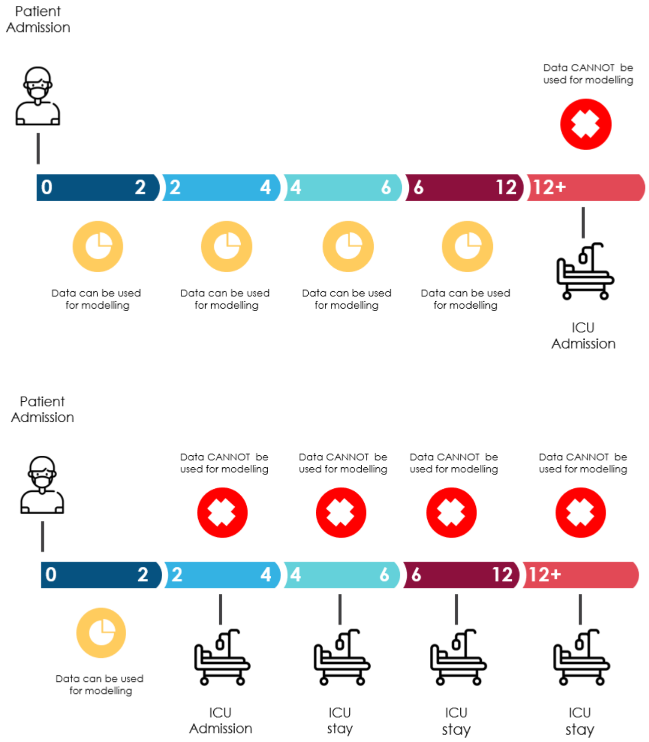
Patient Admission
Data CANNOT be used for modelling
0
2
2
4
4
6
6
12
12+
Data can be used for modelling
Data can be used for modelling
Data can be used for modelling
Data can be used for modelling
ICU Admission
Patient Admission
Data CANNOT be used for modelling
Data CANNOT be used for modelling
Data CANNOT be used for modelling
Data CANNOT be used for modelling
0
2
2
4
4
6
6
12
12+
Data can be used for modelling
ICU Admission
ICU stay
ICU stay
ICU stay

Fig. 1. The order of the events matters. [1].

Also, because the dataset provider mentioned that it is better to fill missing values with the previous or next entry of the same patient, we do the same thing to fill out the Nan values in the data.

OUR EARLY PREDICTION SCENARIO: As mentioned in the previous section, our data consists of many different features, and the raw dataset has 227 different columns representing these features. As the data provider claim, a model with only the "0-2 window" is more clinically relevant. Therefore, we remove the rows corresponding to other periods from our data in the data cleaning section. Therefore, to have more accurate data and satisfy our early prediction goal, we only use the data FROM THE FIRST ROW of each patient to predict whether he/she will need to go to the ICU at the end (this may happen in the last window) or not. Therefore, from now on, all of our implemented models are capable of predicting the ICU admission by JUST HAVING THE INFORMATION OF THE FIRST WINDOW (hours 0-2).

B. Visualization

Different kinds of visualization have been done to have a better understanding of the data.

1) ICU Admission Distribution: As you can see in figure 2, 163 entries have been admitted to ICU and 190 entries that have not been admitted to ICU. Therefore, we have almost the same number of cases for both labels, a fair distribution for our study.

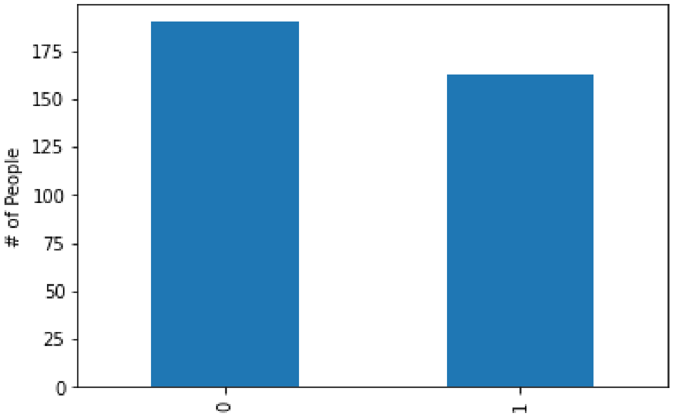

Fig. 2. The distribution of the people in the dataset with respect to ICU admission. 1 states that the person is admitted to ICU and 0 states otherwise.

2) Age Distribution: In this section, we are checking the age distribution for ICU admission to see how they are related. As we can see in figure 3, it is clear that ages above 65 increase the chance of ICU admission. For a better comparison, please refer to figure 4.

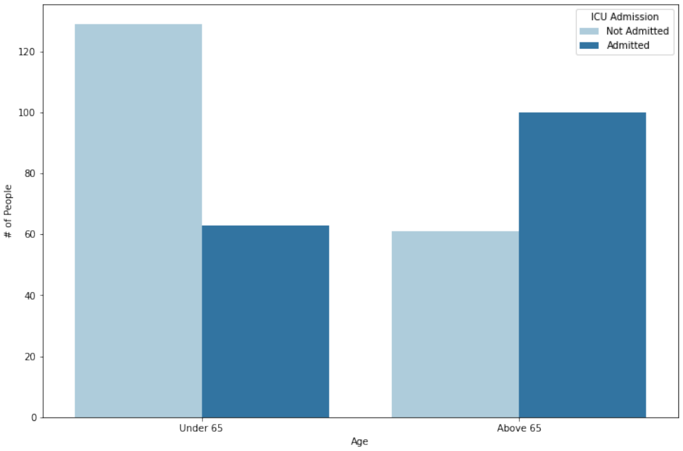

Fig. 3. The age distribution with respect to ICU admission.

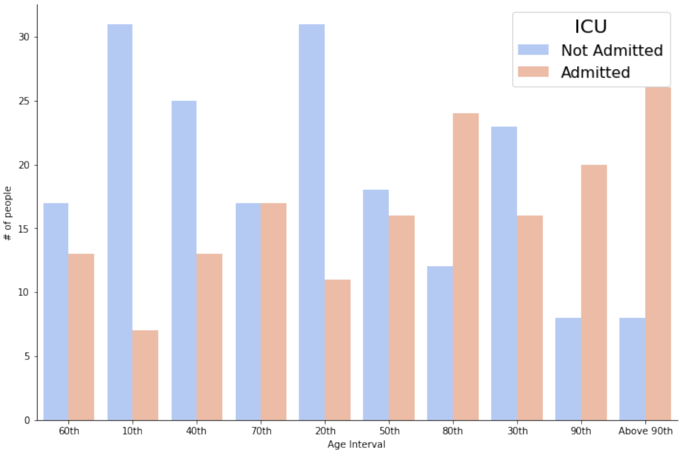

Fig. 4. A comparison of different age periods with respect to ICU Admission.

3) Gender Distribution: Here, we wish to see how gender distribution looks like for ICU Admission. Unfortunately, it is not stated in the dataset that which number represents which gender, but as you can see from figure 5, it sounds like more people were admitted for ICU in gender '0' than '1'. And also, the majority of gender '1' was not admitted to ICU.

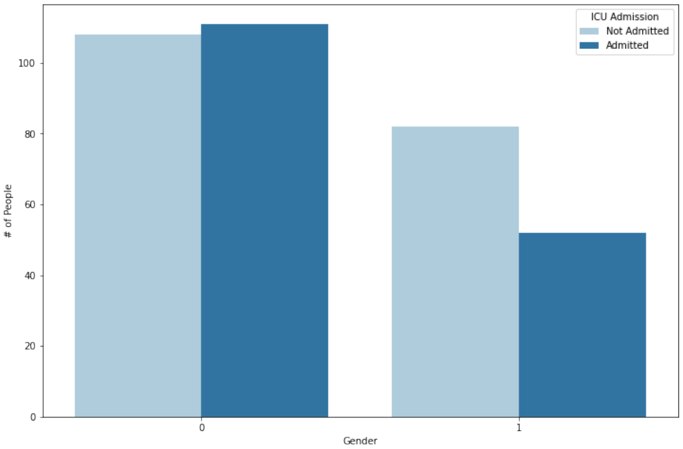

Fig. 5. Gender distribution for ICU Admission.

4) Disease Distribution: If you look at the dataset carefully, you will find a column called 'tags' at the end. This column represents different kinds of diseases that the patient may have had in the past. These diseases include Motor Neurone Disease, Smoking, Lung cancer, asthma, Kidney disease, and heart disease. Also, some other columns in the data show some disorders that one may be interested in studying its effect on ICU Admission, such as HTN and immunocompromised. You can see what is the distribution of these kinds of disease for the ICU Admission in figure 6.

5) Vital Signs Distributions: Various vital signs have been reported in this dataset. The critical symptoms in

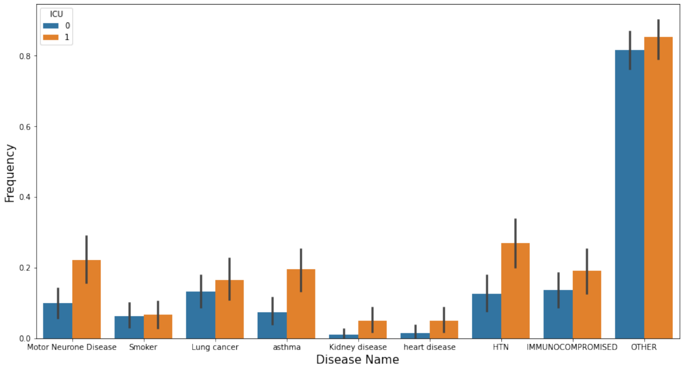

Fig. 6. Disease distribution for ICU Admission.

the dataset include blood pressure diastolic blood pressure systolic, heart rate, respiratory rate, temperature, and oxygen saturation. We have the information about their mean value, median, min, max, diff, and rel values in the dataset. To have a general idea of how they might affect ICU Admission, you can refer to figure 7.

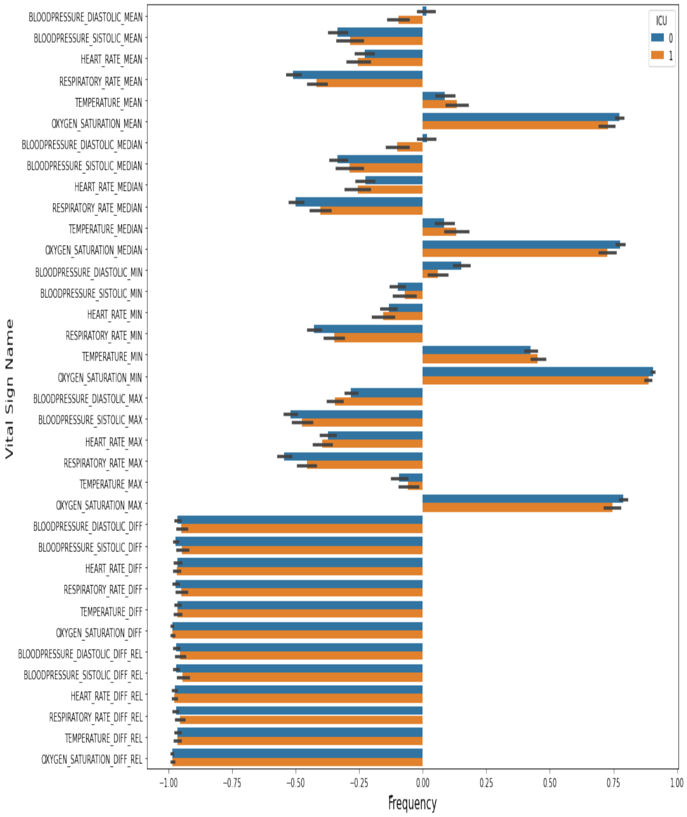

Fig. 7. Vital Signs distribution for ICU Admission.

C. Feature Engineering

We explained some of our feature engineering approaches in the Data Cleaning section. There are many other medical tests that we have their results in our dataset. Also, we have the min, max, median, mean, diff values for each of them. We should decide whether we want to keep them for modeling or not. If you look closely at the data, you will notice that the 'DIFF' columns are all the same! Therefore, they will have a variance of 0, and we should omit them from our data.

Moreover, after removing the 'DIFF' columns from our data, as you can also see in figure 8, we have a correlation of 1 between the min, max, mean, and the median of these medical tests' columns too, which means that they all have the same value. Therefore we will keep one of these four columns for each feature (for example, the mean column).

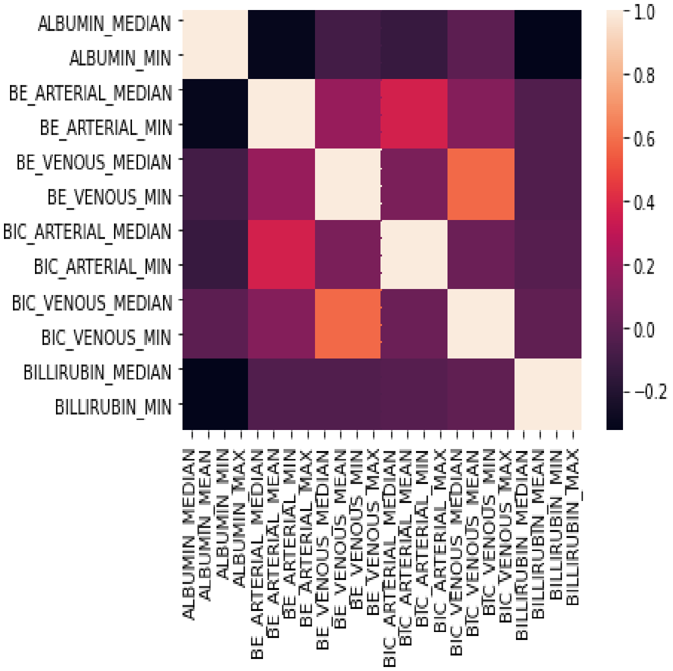

Fig. 8. Heat map representing the correlation between some of the medical test results. The darker cells show low correlation, the brighter one show high correlation between features.

There was also a feature called 'AGE_PERCENTIL' that was categorical and showed the age period of the patient. We converted this column into one-hot encoding columns of numerical data.

After all of these steps, our final dataset contains 353 rows and 94 columns.

IV. Experiments and Results

In this section, we want to try different machine learning models that we have learned through this course to answer this question of "Does this patient need to go to the ICU?".

It is good to note that we use the 80-20 proportion for our train-test split.

A. Metrics

We use three Metrics that we think are useful for the evaluation of our models. The most important one is the F1-score. We report ROC-AUC and Accuracy as well.

1) F1-Score: In statistical analysis of binary classification, the F-score or F-measure is a measure of a test's accuracy. It is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive [2].

$$\mathrm{F1} = 2.\frac{\mathrm{precision.recall}}{\mathrm{precision} + \mathrm{recall}} \tag{1}$$

2) ROC-AUC: A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative') [3].

B. Models

We have implemented five different models for our dataset, Logistic Regression, Random Forest, Extra Tree Classifier, Bayes Net, and Support Vector Machine (SVM) using the built-in functions in the SKLEARN library in python [4].

1) Logistic Regression: Logistic regression is a supervised learning algorithm which is mostly used to solve binary "classification" tasks although it contains the word "regression". The basis of logistic regression is the logistic function, also called the sigmoid function, which takes in any real valued number and maps it to a value between 0 and 1 [5].

$$\mathrm{Sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{2}$$

Assume that $f(x)$ is a function consisting our features (x_j) and their corresponding weights/coefficients (β_j) in a linear form shown below.

$$f(x) = x_0 + x_1\beta_1 + \cdots + x_k\beta_k + \epsilon \tag{3}$$

where $x, \beta, f(x) \in R^k$ and ϵ is representing the random error process noise inevitably happening in the data generating process [6]. Now, if we assume that,

$$P(Y|X) = \mathrm{sigmoid}(f(x)) \tag{4}$$

then we can rewrite the estimation function $f(x)$ in the form of 'posterior probability' as shown below.

$$\log[\frac{P(Y|X)}{1-P(Y|X)}] = x_0 + x_1\beta_1 + \cdots + x_k\beta_k + \epsilon. \tag{5}$$

Logistic Regression has an 'objective function' which tries to maximize 'likelihood function' of the experiment. This approach is known as 'Maximum Likelihood Estimation — MLE' and can be written mathematically as follows.

$$\mathrm{argmax}_\beta : \log\{\prod_{i=1}^{n} P(y_i|x_i)^{y_i}(1-P(y_i|x_i))^{1-y_i}\} \tag{6}$$

Now, we should optimize this objective function with respect to β [6].

For prediction of the ICU admission, we train our data with Logistic Regression model with the help of 10-fold Cross Validation technique. We reached an accuracy of 74%, AUC of 78%, and the F1-Score of 71%.

2) Random Forrest Classifier: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees. However, data characteristics can affect their performance [7].

For prediction of the ICU admission, we train our data with Random Forrest Classifier model with the help of Grid Search Cross Validation (Hyper-parameter Tuning step). The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid [8]. We reached an accuracy of 73%, AUC of 73%, and the F1-Score of 70%.

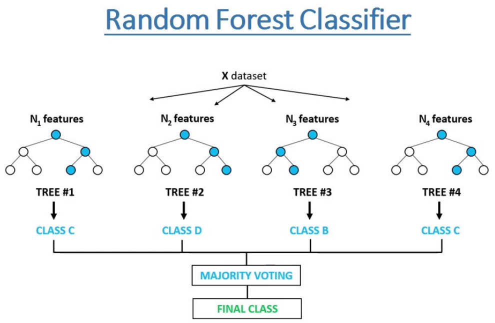

Fig. 9. Random Forrest Classifier.

3) Extra Tree Classifier: Extremely Randomized Trees Classifier(Extra Trees Classifier) is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a "forest" to output it's classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.

Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, Each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees [9].

For prediction of the ICU admission, we train our data with Extra Tree Classifier model with the help of Grid Search Cross Validation (Hyper-parameter Tuning step). We reached an accuracy of 72%, AUC of 71%, and the F1-Score of 69%.

4) Naive Bayes: Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location. Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence [10]. Naive Bayes Classifier is relying on one simple rule, Bayes Rule,

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{7}$$

where $P(h)$ is the probability of hypothesis h being true (regardless of the data). This is known as the prior probability of h. $P(D)$ is the probability of the data (regardless of the hypothesis). This is known as the prior probability. $P(h|D)$is the probability of hypothesis h given the data D. This is known as posterior probability. $P(D|h)$ is the probability of data d given that the hypothesis h was true. This is known as posterior probability.

Naive Bayes Classifier works as follows:

1) calculate prior probability for given class labels.
2) calculate conditional probability with each attribute for each class.
3) multiply same class conditional probability.
4) multiply prior probability with the above step probability.
5) see which class has higher probability, higher probability class belongs to given input step.

For prediction of the ICU admission, we train our data with Gaussian Naive Bayes model with the help of 10-fold Cross Validation technique. We reached an accuracy of 58%, AUC of 74%, and the F1-Score of 25%.

5) Support Vector Machine (SVM): A support vector machine (SVM) is a supervised learning algorithm used for many classification and regression problems , including signal processing medical applications, natural language processing, and speech and image recognition.

The objective of the SVM algorithm is to find a hyperplane that, to the best degree possible, separates data points of one class from those of another class. "Best" is defined as the hyperplane with the largest margin between the two classes, represented by plus versus minus in the figure 10. Margin means the maximal width of the slab parallel to the hyperplane that has no interior data points. Only for linearly separable problems can the algorithm find such a hyperplane, for most practical problems the algorithm maximizes the soft margin allowing a small number of misclassifications.

Support vectors refer to a subset of the training observations that identify the location of the separating hyperplane. The standard SVM algorithm is formulated for binary classification problems, and multiclass problems are typically reduced to a series of binary ones [11].

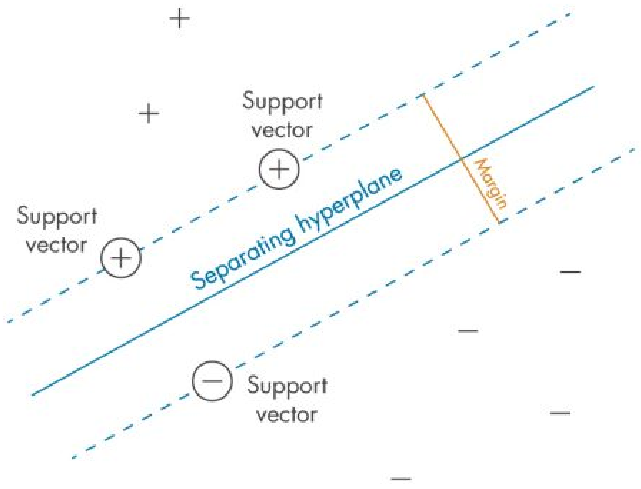

Fig. 10. Learn optimal hyperplanes as decision boundaries in SVM.

For prediction of the ICU admission, we train our data with SVM model with the help of 10-fold Cross Validation technique. We reached an accuracy of 76%, AUC of 76%, and the F1-Score of 75%.

C. Finding the Best Model

Now, let us summarize the results from the previous section. As we mentioned before, F1-score is the most crucial feature we with to maximize, and figure 11 shows that the SVM Classifier reaches the most in terms of F1-score. The results are also numerically reported in table 1.

Model	F1-Score	Accuracy	AUC
Logistic Regression	71%	74%	78%
Random Forrest Classifier	70%	73%	73%
Extra Tree Classifier	69%	72%	71%
Gaussian Naive Bayes Classifier	25%	58%	74%
SVM	75%	76%	76%

TABLE I
Comparison between different models with respect to F1-Score, Accuracy, and AUC.

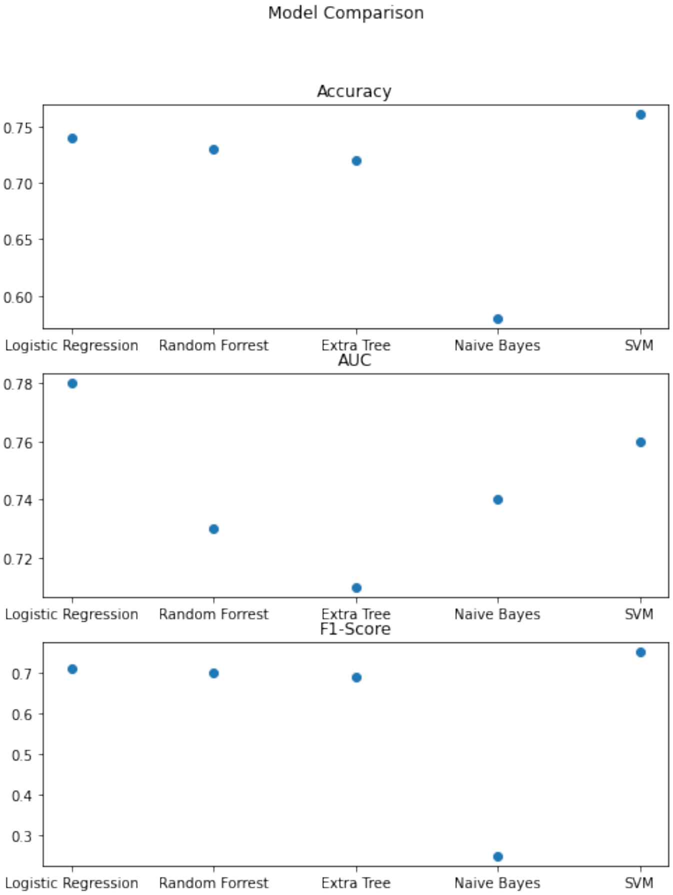

Fig. 11. Comparison of the models with respect to the metrics.

D. Feature Importance

Now that we have found out that SVM Classifier have the best results, we want to see which features most affect its prediction. You can see the top 15 important features in the SVM model in figure 12.

We can see this from figure that the SVM model is paying attention to the PCR test results the most, something we all know that is common in most countries to confirm Covid-19 cases. Also, it is paying attention to the people above age 90, which is also reasonable based on our thoughts. Other interesting features include Lung Cancer, Linfocitos, Blood Pressure and so on.

V. Conclusion

In this report, we analyzed the data received from a hospital in Italy to find a model that can early predict whether a patient needs to be admitted to ICU or not. Our final model, an SVM Classifier, can predict ICU Admission of a patient with an F1-score of 75% only by using the data collected in the first two hours from when the patient has gone to the hospital.

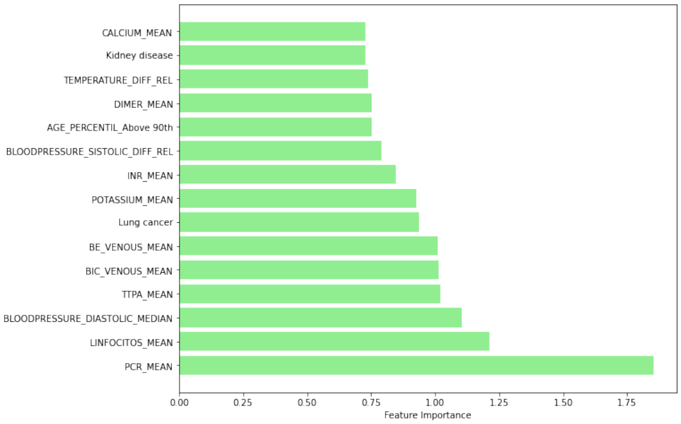

Fig. 12. The top 15 most important feature in the SVM Model.

Acknowledgment

This is the Machine Learning Course (CE 717) final project (phase 1) report, Sharif University of Technology.

References

[1] COVID-19 - Clinical Data to assess diagnosis, Sírio-Libanês data for AI and Analytics by Data Intelligence Team, {https://www.kaggle.com/S%C3%ADrio-Libanes/covid19}.

[2] Wikipedia contributors. (2021, May 17). F-score. In Wikipedia, The Free Encyclopedia. Retrieved 15:49, May 22, 2021.

[3] Wikipedia contributors. (2021, May 16). Receiver operating characteristic. In Wikipedia, The Free Encyclopedia. Retrieved 16:00, May 22, 2021.

[4] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[5] How is Logistic Regression Used as A Classification Algorithm?, Towardsdatascience, Soner yildirim, 2020.

[6] LOGISTIC REGRESSION CLASSIFIER, Towardsdatascience, Caglar Subasi, 2019.

[7] Wikipedia contributors. (2021, May 6). Random forest. In Wikipedia, The Free Encyclopedia. Retrieved 06:07, May 26, 2021.

[8] sklearn.model_selection.GridSearchCV.

[9] ML | Extra Tree Classifier for Feature Selection,GeeksforGeeks, Alind Gupta, 2020.

[10] Naive Bayes Classification using Scikit-learn, datacamp.com, Avinash Nalvani, 2018.

[11] Machine Learning With Matlab, Support Vector Machine(SVM).

[12] https://github.com/AndreisSirlene/Bootcamp_datascience/tree/main/Final%20Project%20-%20ICU%20Prediction%20Sirio%20Libanes.