DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2019

Emergency Department Triage Prediction of Emergency Severity Index using Machine Learning Models

Zohib Sekandari & Shahin Saleh

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2019

Akutmottagningens Förutsägelse av Emergency Severity Index med hjälp av Maskininlärningsmodeller

Zohib Sekandari & Shahin Saleh

Abstract

Study Objective: The emergency department (ED) in the United States strongly rely on subjective assessment of patients. This study seeks to evaluate an electronic triage system based on machine learning models that can predict the patients emergency severity index (ESI).

Methods: A dataset containing 560 486 patients triage data was investigated. Three different machine learning models was tested and evaluated. A cross validation table and a confusion matrix was conducted from each of the models. The precision rate ,recall rate and f1-score were calculated and reported.

Result: The Gradient Boosting model returned an accuracy rate of 68%. The random forest model returned an accuracy rate of 66%. The Gaussian Naive Bayes model returned an accuracy rate of 25%.

Conclusion: The model that best predicted the ESI-level is the Gradient Boosting model. Further testing is needed with better computational power since we could not train our model with the whole dataset.

Keywords

Machine Learning, Triage, ESI, Classification

Abstract

Syfte: Akutmottagningen i USA förlitar sig kraftigt på en subjektiv värdering av patienter. Denna studie söker efter att evaluera ett elektronisk triage system baserad på maskininlärningsmodeller som kan förutse patienters ESI.

Metod: Ett data set som innehåller 560 486 patienters triage data har undersökts. Tre olika maskininlärningsmodeller har testats och evaluerats. En cross validation tabell och en confusion matrix har skapats för varje modell. Precision, recall och f1 värde har kalkylerats och rapporterats.

Resultat: Gradient Boosting modellen har returnerat ett accuracy värde av 68%. Random Forest modellen har returnerat ett accuracy värde av 66%. Gaussian Naive Bayes modellen har returnerat ett accuracy värde av 25%.

Slutsats: Modellen som har bäst förutsett ESI nivåerna är Gradient Boosting modellen. Flera tester behövs med starkare beräkningskraft då vi inte kunde träna vår modell med hela datasetet.

Nyckelord

Maskininlärning, ESI, Klassificering

Authors

Shahin Saleh <ssaleh@kth.se>
Zohib Sekandari <zohib@kth.se>
Information and Communication Technology
KTH Royal Institute of Technology

Place for Project

Stockholm, Sweden

Examiner

Örjan Ekeberg
KTH Royal Institute of Technology

Supervisor

Pawel Herman
KTH Royal Institute of Technology

Contents

1 Introduction

During recent years the demand for medical care have grown faster than the growth rate for hospital resources. This has led to several problems at Emergency Departments (ED). One of these are that waiting time has increased, hence leading to overcrowded EDs[4]. In 2008 only 18% of the patients that visited the ED were treated within 15 minutes, leaving the majority of patients waiting in the waiting room[9]. This could lead to poor outcomes for patients and increase the length of stay[4]. In order to prevent overcrowding, there has been attempts to optimize patient flow in EDs by developing a Triage System. Triage is the assessment process that will sort patients prior to their medical condition. Thus patients with the most severe condition will be treated first. The severeness of patients medical condition is determined by an ED nurse that assigns each patient with a severity index. The level can be assessed by using different types of indexes. The most common index system is 3 and 5-level index classifications. The most used Triage system in the US is the Emergency Severity Index (ESI). The value will reflect patients medical condition.

ESI triaging relies heavily on the triaging nurses judgment which is subject to high variation[6]. It also poorly differentiates ESI index 3 patients. Hence, there is a problem with under and overtriaging[1]. Undertriaging leads to safety risks for patients that should be assigned index 1 or 2 but was assigned index 3. Overtriaging leads to insufficient allocation of ED resources due to patients with more severe condition gets more medical attention[15]. To prevent overcrowded EDs, and to better predict the ESI level, we will develop an electronic triage system (E-triage) based on machine learning algorithms that can support ED nurses in deciding ESI levels for patients.

Machine learning algorithms are used more often within the area of health care. Recent studies have shown that machine learning algorithms can be helpful for disease detection, patient care, and community services to name a few[2, 3, 17]. Since Machine learning models have been used earlier on other parts of health care it can also be used to perform the triaging process at the ED.

1.1 Purpose

This research project will compare different machine learning models and see how accurate they can predict the ESI level for patients arriving at the ED. A successful model would help ED nurses at the ED to take more accurate decisions when classifying patients. It would also help avoid overcrowding and shorten decision times during the triaging process.

1.2 Research Question

The objective for this study is to use machine learning models to develop an electronic triage support system (e-triage) that predicts the ESI for patients. And compare the different models to find the most optimal model that can predict ESI with the highest accuracy. The question for this paper is the following:
Which of the following machine learning algorithms predicts the ESI for patients with the highest accuracy. Random Forest, Gradient Boosting or Gaussian Naive Bayes?

1.3 Scope

Main focus will be to predict the ESI level and not any other triaging system. The research will also only focus on Random Forest, Gradient Boosting and Gaussian Naive Bayes machine learning algorithms. We are limited to a single data set since there isn't many other free datasets that have the necessary variables that we need to solve our triaging problem.

2 Background

2.1 Triage

Triage is a classification process that is important to effectively assign patients with the necessary resources[27]. Such a classification process becomes especially important when there is a mismatch in time or location between the medical needs of patients and the available resources[1]. Most triage systems are constructed in a way that benefits human life and efficient use of resources. Another important aspect of triaging is to care for human health with fairness. This is done by determining how urgent the medical condition is. If the condition is urgent the patient will be prioritized otherwise the patient will have to wait for its turn[5].

The triaging is traditionally taken care of by ED nurses. Nurses at the ED is responsible for organized queues where the patients with the most critical condition is placed first in the queue[6].

2.1.1 Emergency Severity Index

ESI is the most commonly used triage system[18]. ESI reliably predicts the severeness of patients medical condition using a 5-level triage system. Patients with urgent medical condition gets sorted as index 1 and the patients with the least urgent medical condition gets sorted as index 5, the stratification is based on acuity and how much resources are needed[25].

Patients that are in critical condition is easily assigned the most severe index, thus receives immediate care. Patients that are in non urgent condition is easily assigned the least severe index. The difficult part is to assign a index for patients that is not a part of the already mentioned extreme states. A majority of patients visiting the ED is assigned a index between 2 and 4. Figure 2.1 describes how a triage algorithm could be constructed. The triage nurses will use this to decide the ESI level for patients.

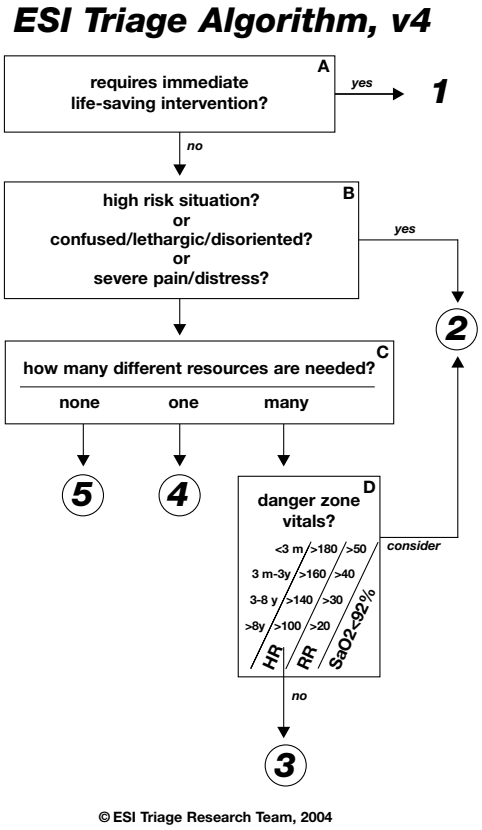

Figure 2.1: ESI triage process description [24]

2.2 Feature Selection

Feature is a variable that describes for an example an object. Some objects can have many features. To describe a rectangular box it would have three features height, depth and width. There are many factors that will contribute to the success of a machine learning model, one of the factors are the quality of the data. If there is to much irrelevant and unnecessary data in the data set it will slow the performance and make the training of the model difficult. Feature selection is the process that will remove those irrelevant and unnecessary features to improve the machine learning model[11].

2.3 Machine Learning

Machine learning is a set of computational methods that detects meaningful patterns in unseen data without being explicitly programmed for it. It relies on patterns and inference instead. In the past couple of decades it has become a common tool in tasks that requires decision making based on information extracted from large data sets[15, 23]. These methods offer advantages for predicting clinical applications because they can return stable predictions[21]. They are able to account for the most significant variables [13] and are proficient at identifying interactions in patient information, enabling them to classify patients to their subgroups with respect to the predicted outcome. The following sub chapters below are about some of the models that will be used.

2.3.1 Decision Tree Classifier

A Decision Tree Classifier (DTC) is a predictor that predicts the class associated with an instance x by travelling from a root node of a tree to a leaf. At each node of the path from the root node to a leaf, the successor child is chosen based on a splitting of the input space. In most cases the splitting is dependant on the feature of x or a set of predefined set of splitting rules. Each leaf belongs to a specific predefined class. Figure 2.2 gives an idea on how the decision tree classifies the instance x.

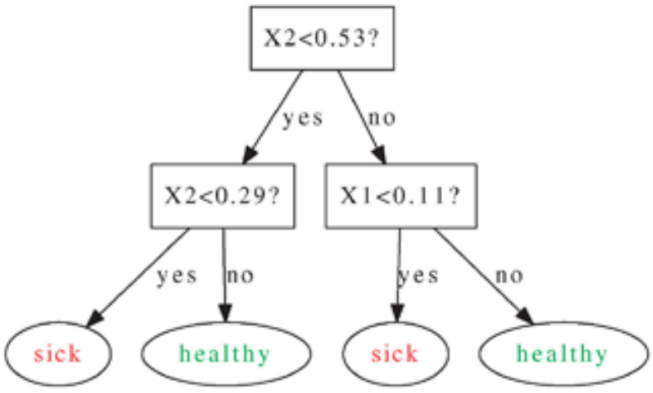

Figure 2.2: Decision tree example [8]

Each splitting of an internal node of the tree depends on thresholding the value of

a single feature of x. The threshold of each splits is determined during the training phase. The training phase target is to split the data set in a way such that all leaves only have instances from one of the classes. Hence, the splits with the greatest improvements ends up in the top of the tree. The training phase needs a stopping criteria or it would be possible to grow a DTC which each instance of the training set occupied its own node. This could result in overfitting. To avoid overfitting it is possible to set a maximum depth for DTCs. Another alternative is pruning. Pruning can further increase the performance of DTCs by removing branches that make use of feature with low importance. Hence, the complexity improves which may result in increased predictive power[10, 22, 23].

2.3.2 Random Forest

To further avoid overfitting of DTCs it is possible to construct an ensemble of DTCs. This is called Random Forest. Random Forest is a classifier consisting of a collection of DTCs, where each DTC is generated using a random vector sampled independently from the input vector, and each DTC casts an unit vote for the predicted class. The class with most votes will be the predicted result for Random Forest[16, 23]. Figure 2.3 gives an idea of how a Random Forest looks like.

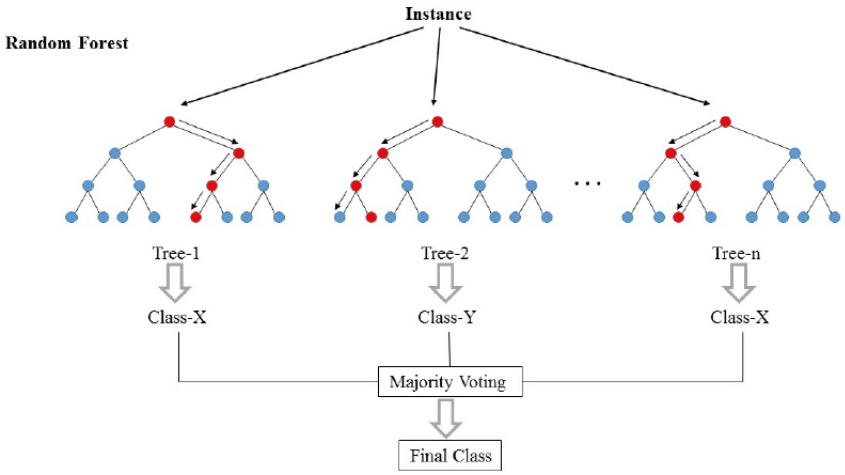

Figure 2.3: Random Forest Tree example [14]

When increasing the number of DTCs in the Random Forest model, the model generally produces more accurate predictions. However when increasing the number of DTCs over some threshold there will be no significant gain in accuracy,

there will only be a significant computational lost [19].

2.3.3 Gradient Boosting

The Gradient Boosting was proposed by Friedman et al. [7] in 2001. The Gradient Boosting algorithm is typically used with DTC and are trained in an additive manner. At each time step t, it creates another tree to minimize the loss function of the current model. The loss function l (see Equation 1) measures the difference between the label of the $i-th$ instance y_i and the prediction $\hat{y}^{t-1}$ at the last step plus the current tree output. We also have a regularization term $\Omega(f(t))$ that penalizes the complexity of the new tree.

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}^{t-1}) + f_t(x_i)) + \Omega(f(t)) \tag{1}$$

2.3.4 Gaussian Naive Bayes

Gaussian Naive Bayes classifier is a probabilistic classifier based on applying Bayes theorem with strong independence assumption. The following is Bayes rule:

$$P(Y|X_1, ..., X_n) = \frac{P(X_1, ..., X_n|Y)P(Y)}{P(X_1, ..., X_n)} \tag{2}$$

The posterior $P(Y|X_1, ..., X_n)$ is used to statistically determine the probability of a set of data point $X_1, ..., X_n$ belonging to a class Y given the observed data points. If there is n classes $Y_1, Y_2, ..., Y_n$ then the class that best fits the data point $X_1, X_2, ..., X_n$ is the class with the highest probability.

2.4 Related Work

Fitzgerald et al. (2010) discuss different types of solution to the triaging and overcrowded EDs problems[6]. The solution to not triage patients is criticized by Fitzgerald et al. since the purpose of triaging is to assign urgency to patients for clinical justice and system efficiency purposes. Fitzgerald et al. thinks that the question is not wether we should triage, the question is what to do with the

patients once they are triaged. Furthermore Fitzgerald et al. believes that the concept of having a fast-track at the ED requires further research.

Challenges with patient differentiating and overcrowded EDs motivated Levin et al. (2017) to develop an electronic triage system based on machine learning algorithms[15]. The electronic triage system demonstrated a better patient differentiation compared to ESI-level provided by the nurses.

Raita et al. (2019) predicted the clinical outcome using machine learning models[20]. The models used was Random Forest, Gradient Boosted Decision Tree and Deep Neural Network. Raita et al. received a AUC score of 0.85 for Random Forest, 0.85 for Gradient Boosted Decision Tree and 0.86 for Deep Neural Network.

3 Method

3.1 Dataset

The dataset used in this study consists of information about 560 486 patients visits to the ED. The data was gathered between March 2013 to July 2017 in three different EDs located in the US[12]. For each patient there is 972 different variables that has been collected from the triage evaluation performed by ED nurses.

Some of the variables are not of our interest in this study and are not collected during the triage process. These variables have been excluded from the dataset.

3.1.1 Variable Groups

To better interpret the dataset we have gathered the features in variable groups. The variable group consists of Demographic, Triage and Chief Complaints (CC) group. The Demographic group consists of the demographic variables such as race, religion and gender. The Triage group consist of triage test data and arrival data. An example of arrival data is how the patient arrived to the ED, by ambulance, car or walking. The CC group consists of 200 variables that describes the reason why the patient is visiting the ED, for example this could be a broken arm.

3.1.2 Cleaning the dataset

Some instances of the dataset contains undefined values such as NaN, Not a Number. All instances of the dataset that has an undefined value for the ESI variable has to be excluded from the dataset. That's because the ESI variable is essential to train our model. For those instances containing an undefined age variable it is set to -1. For those instances containing an undefined race variable it is set to 'other', as 'other' was already defined in the dataset. For all the CC

group that has the whole row containing NaN values it has been removed from the dataset. If there is rows that contains NaN values after the clean up it will be set to ”-1”. It’s because that the CC group only contains ”0” or ”1”, where ”0” indicates as false and ”1” as true. So ”-1” will represent a missing value.

3.1.3 Label encoding

The dataset contains some categorical variables that does not get properly interpreted by the machine learning model. For an example we have some Demographic and Triaging variables that have to be encoded into numerical values. Thus, we use sklearns labelencoder and onehotencoder to encode the categorical values into numerical values.

3.2 Implementation

The dataset is split into a training set consisting of 80% of the dataset and a test set consisting of the remaining 20%. The training set is used to fit our machine learning model. Thus making our model to do better predictions. The test set is used to provide an unbiased evaluation of the final model fit on the training set. The variable we are trying to predict is the ”ESI” variable. We will import and use the following packages for our ml models panda[1], numpy[2], matplotlib[3], seaborn and Sklearn[4]. All of the packages are open source. And we will use Jupyter as our desktop client.

The Gradient Boosting and Random Forest classifier have been chosen because they have shown great results on previous work that had a similar problem[12]. The Gaussian Naive Bayes model is used because of its speed and it should work when there is multiple classes in the classification problem.

[1]https://pandas.pydata.org/
[2]https://www.numpy.org/
[3]https://matplotlib.org/
[4]https://scikit-learn.org/stable/index.html

3.2.1 Random Forest

The Random Forest model is built upon *n* DTCs. For this study we will try to fit the random forest model on $10 \leq n \leq 100$ number of DTCs. Accordingly we will choose the *n* value that best predict the test set. We will also choose a maximum depth *k* of the DTC. We will try $10 \leq k \leq 50$ and choose *k* that gives best predictions on the test set. We will try a combination of this values to optimize our model. This will be done by hyperparameter Optimization see chapter 3.2.4. We will use scikits randomforestclassifier to train our model[5].

3.2.2 Gradient Boosting

We have imported sklearns GradientBoostingClassifier[6] and used it as our Gradient Boosting classifier. We will test our model with the same amount of trees and the same maximum depth level as our Random Forest classifier. The hyperparameters was set to the default values except the learning rate and warm_start they were set to 0.1 and True.

3.2.3 Gaussian Naive Bayes

For the Gaussian Naive Bayes model we have imported sklearns Gaussian Naive Bayes[7]. We are using the standard model without any special features. And we have used the fit and predict method to run the Gaussian Naive Bayes model. We have not tuned any hyper parameters, we have just used the standard model.

3.2.4 Hyperparameter Optimization

Parameters whose value is set before the learning process begins are refered to as hyperparameters[26]. Some key hyperparameters that are used is the number_of_trees in the forest and the maximum_depth of each tree.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[6]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

[7]https://scikit-learn.org/stable/modules/generated/sklearn.naive$_bayes.GaussianNB.html$

number_of_trees is how many DTCs see chapter 2.3.1 that we are going to use. These values are optimized by cross-validated grid-search over a parameter grid. This means that we have given some fixed number on number_of_trees and some fixed number on maximum_depth and then picked those numbers that has given us the highest accuracy.

3.3 Evaluation

To evaluate the performance of the different machine learning models we need to evaluate them. The evaluation method we will be using are precision rate, recall rate, f1-score, true vs false and positive vs negative. To calculate precision rate we use the following formula:

$$PrecisionRate = \frac{tp}{tp + fp} \tag{3}$$

The true positive (tp) are those patient that we successfully predicted the right ESI-level for. The false positives (fp) is a type 1 error and describes those patients we predicted a lower ESI-level than the actual outcome. The precision rate gives us an idea of how precise our model predicts the test set.

To calculate the recall rate we use the following formula:

$$RecallRate = \frac{tp}{tp + fn} \tag{4}$$

The false negative (fn) is a type 2 error and describes those patients that we thought had a less severe condition than they actually did. It is very important to not receive a high value for fn because this is the worst type of error. A high value for fn will result in a low recall rate. Hence, recall rate is a measure for fn.

The true vs false and positive vs negative evaluation method will be represented with a confusion matrix. A confusion matrix is a specific table that allows visualization of the performance of our model. Each row represents an instance in a predicted class and each column represents the instances in the actual class. The

confusion matrix is built based on the result from the test set. After receiving the confusion matrix the different kinds of errors will be visual. Based on those errors we can concentrate on those areas that needs improvement. The main target is to receive a high precision rate while trying to have as low type 1 and type 2 errors as possible. F1-score is a measurement of the type 1 and type 2 errors.

To compare the results from the different models we will calculate the f1-score from each models classification on the different ESI levels.

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \tag{5}$$

The F1 score is based on the precision and recall rate. The max value is 1, the closer the f1 score is to 1 the better the accuracy.
The final selection to decide which model have the best performance will be based on the accuracy score. It will be based on the total accuracy from all the classifications indexes.

4 Result

In this section the results from the tests on our models with the data set is presented. Every models performance is displayed with a confusion matrix and it will show the accuracy score for different parameters on our models. After the data have been cleaned, we changed our categorical variables to binary values with the onehotencoder we have a dataset containing 296 variables. We have chosen to modify the same parameters on our Random Forest model and our Gradient Boost model. For our Gaussian Naive bayes model we have not made any modifications. We will first present the model that we tested first and then move to the other models. Lastly we will make a comparison between them.

4.1 Results from Random Forest model

Our first model that we tested was the Random Forest classification model. The following are the results that we got when we modified our hyperparameters.

Results of Random Forest model with different hyperparameters			
Number of Trees	Max size of depth	Accuracy score	Time
10	10	0.555	0min 5s
25	10	0.0559	0min 11s
50	10	0.559	0min 21.4s
100	10	0.559	0min 42.5s
10	30	0.623	0min 13.3s
25	30	0.639	0min 30.8s
50	30	0.644	1min 10s
100	30	0.647	2min 00s
10	50	0.623	0min 18s
25	50	0.644	0min 53s
50	50	0.651	1min 16s
100	50	0.659	2min 48s

Table 4.1: Random Forest cross validation.

From table 4.1 we can see how the accuracy changes depending on the maximum depth and the number of trees. The best result comes from the last test where we had max_depth = 50 and number_of_trees = 100. At the lower amount of trees and sizes of depth there isn't any huge difference between the results. But as the size of the depth get's larger the better results we get. We can see when we have maximum amount of trees and minimum depth we have a accuracy score of 0.559. And when we use the maximum amount of depth and minimum amount of trees we get an accuracy score of 0.623. The Random Forest can be run concurrently on all available cores on the computer, for our model the tests have been run on four cores.

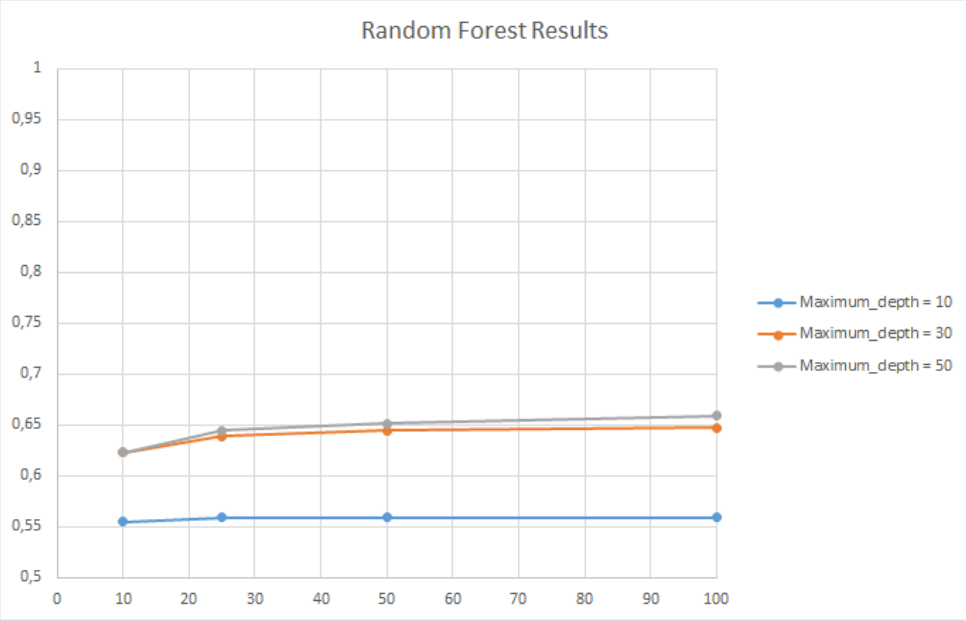

Figure 4.1: Cross validation graph for Random Forest.

From the results on table 4.1 we have figure 4.1. In the X axis we can see the number of trees that are used and on the Y axis we have the accuracy score. The different lines represent the variety of the depths for an example the blue line represents when the maximum_depth = 10. From the graph we can see that when the maximum_depth is modified it will increase the accuracy score, and it will further increase when the amount of trees get larger.

4.1.1 Confusion Matrix Random Forest

The confusion matrix that we have received from our test data. On the y-axis we can see the actual value and on the x-axis we can see the predicted value. For an example if the predicted value is 5 and the actual value is 5 we have a hit and if they aren't the same we will have miss.

Figure 4.2: Confusion matrix Random forest for tree = 100 and depth = 50.

Figure 4.2 describes the result received from the Random Forest model that had the best accuracy rate. We can see that it made the best predictions for index 3. And we can see that it wrongly classified many ESI level 2 patients as level 3.

4.1.2 Metrics classification report

Table 4.2 is the precision, recall and F1-score for each class and how many of each types we have in our data set. The report have a overview of the results and its not as detailed as the confusion matrix in figure 4.2.

Metrics classification report for Random Forest				
	Precision	Recall	F1-score	Support
index 1	0.662	0.185	0.289	940
index 2	0.714	0.611	0.658	29199
index 3	0.639	0.738	0.685	41634
index 4	0.640	0.649	0.644	22354
index 5	0.656	0.412	0.506	4964
micro avg	0.659	0.659	0.659	99091
macro avg	0.662	0.519	0.557	99091
weighted avg	0.662	0.659	0.655	99091

Table 4.2: Metrics classification results Random Forest.

On the left we can see the different indexes and the columns that follow are the precison, recall, f1 score and support. Support is how much data is of that specific type. As mentioned above we can see that we have most data for index 3 classification in our test data. The precision was highest for index 2. The recall and F1-score was highest for index 3. The recall and F1-score was very low for index 1 but it had the least amount of data as well.

4.1.3 Summary Random Forest

The results from all the different tests that we have done on the random forest model have shown that the accuracy increases as the number of trees and maximum depth increases. It does not take a lot of time to train the model. The biggest issue from the results is as mentioned above the recall rate on index 1.

4.2 Results for Gradient Boosting model

For the gradient boost we have chosen to optimize the same parameters as our Random Forest model. Since there is many features that could have been changed we wanted to tune only the features that they both had so that we could compare them on the same tuning settings.

Results of Gradient Boosting model			
Number of Tree's	Max size of depth	Accuracy score	Time
10	10	0.627	33min 50s
25	10	0.653	90min 42s
50	10	0.668	188min 4s
100	10	0.676	290min 50s
10	30	0.655	442min 0s
25	30	0.667	1112.5min 0s
50	30	0.669	1976min 27s
100	30	0.670	4063min 8s
10	50	0.658	724min 50s
25	50	x	x
50	50	x	x
100	50	x	x

Table 4.3: Gradient Boosting cross validation.

Table 4.3 shows that the gradient boosting model is different from our Random Forest model, it has higher accuracy and we can see how it improves after each time we tune our model. The data that is marked with x was too heavy for our computer that we have been using. We were not able to find a resource that have stronger computational power that could do the test. We can also see from the results when the depth increases the computation time will increase, the Gradient Boosting model can't run concurrently as the Random Forest model.

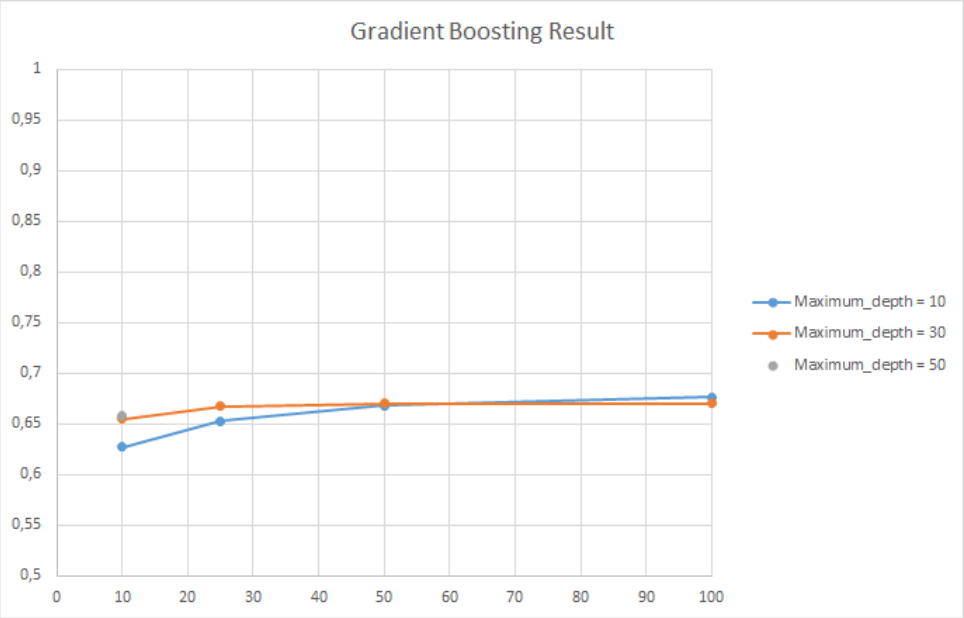

Figure 4.3: Cross validation graph for Gradient Boosting

In figure 4.3 we have the graph for the cross validation result. There is only one point for maximum_depth = 50 because as it was mentioned above we did only test it for 10 trees. As for maximum_depth = 30 we can see as the trees increase after 50 trees the accuracy drops. While for maximum_depth = 10 it will increase as the trees amount of trees increases.

4.2.1 Confusion Matrix Gradient Boosting

This is our results from the confusion matrix for the Gradient Boosting model, it has the same design as the confusion matrix for our Random Forest model.

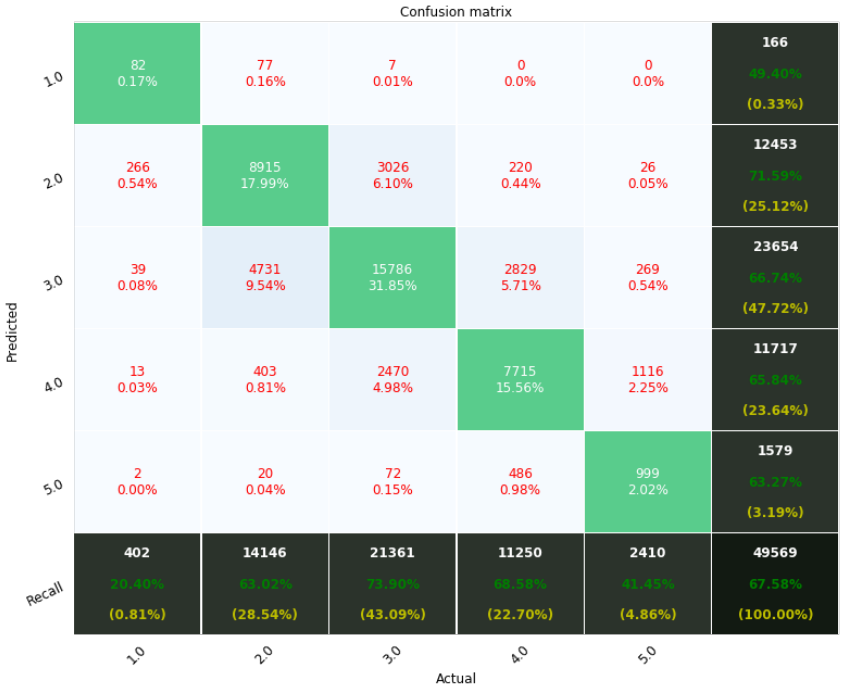

Figure 4.4: Confusion matrix Gradient Boosting for tree = 100 and depth = 10.

As we see in figure 4.4 we had most true predictions for index 3, This is because we had most cases for that specific index in our test set. Overall it had great classifications a little bit off on index 1 and 5 as did the Random Forest model.

4.2.2 Metrics classification report

The recall for index 1 was lower for our Gradient Boosting model than the Random Forest and Gaussian Naive Bayes model. However the training and testing data was not as large as the other models. This result is from when the maximum_depth is set to 10 and amount_of_trees is set to 100.

Metrics classification report for Gradient Boosting				
	Precision	Recall	F1-score	Support
index 1	0.494	0.204	0.289	402
index 2	0.716	0.630	0.670	14146
index 3	0.667	0.739	0.701	21361
index 4	0.658	0.686	0.672	11250
index 5	0.632	0.415	0.501	2410
micro avg	0.676	0.676	0.676	49569
macro avg	0.634	0.535	0.567	49569
weighted avg	0.676	0.676	0.673	49569

Table 4.4: Metrics classification results Gradient Boosting.

4.2.3 Summary Gradient Boosting

The Gradient Boosting model have the highest accuracy score compared to the other models but it is also the slowest one to train. We were not able to compute all the testes as mentioned in section 4.2. But from figure 4.3 we can see that as the amount of trees increased after 50 trees in maximum_depth = 30 the accuracy rate started to drop a little bit.

4.3 Important features

We have used Random Forest and Gradient Boostings feature_importance function to find the most important features for our models and how much they have affected our results. The function returns a array with all the features and a score that will measure its importance. The total value from all the features are summed up to 1. It does not show exactly which ESI index is affected by the different features from the list of features. This will give us a further knowledge on what data is important to gather. The following are the top five features that have the most effect on our models when making the classification.

- age
- triage_vital_hr
- triage_vital_dbp
- triage_vital_sbp
- triage_vital_temp

From the list we can see that age is the most important feature for the demographic variable group. triage_vital_hr is the most important for triage group. Lastly for the CC group we have the cc_abdominalpain as the leading variable.

4.4 Results for Gaussian Naive Bayes

As mentioned in the method chapter we have not used any hyperparameters for our Gaussian Naive Bayes model. So the results for this model is based on the standard model with the same data set that we have used for the other machine learning models. We have used the same amount of data for this model as the Random Forest. The training time is 7.5 seconds.

4.4.1 Confusion Matrix Gaussian Naive Bayes

As for our other models we have constructed a confusion matrix for this model to display how it performed in all its classification.

Figure 4.5: Confusion matrix Gaussian Naive Bayes

From figure 4.5 we can see that this model had issues when it was classifying index 2 and 3, but it did very good for index 1. But this model have a very low accuracy score.

4.4.2 Metrics classification report

From table 4.5 we can see that the recall was highest for index 1 compared to the rest of the indexes. The recall for this index was the highest one of all the tests we did compared to other models. the F1-score was highest for index 4 and precision was highest for index 3. The recall for index 2-4 was very low so we can state that our model have problems with the classification for these indexes.

Metrics classification report for Gaussian Naive Bayes				
	Precision	Recall	F1-score	Support
index 1	0.019	0.977	0.037	940
index 2	0.558	0.188	0.281	29199
index 3	0.726	0.176	0.283	41634
index 4	0.569	0.331	0.419	22354
index 5	0.189	0.650	0.293	4964
micro avg	0.246	0.246	0.246	99091
macro avg	0.411	0.464	0.263	99091
weighted avg	0.607	0.246	0.311	99091

Table 4.5: Metrics classification results Gaussian Naive Bayes.

4.4.3 Summary Gaussian Naive Bayes

The summary of this model is that it is very fast to train. It took only 7.5 seconds. But the accuracy is very low. This model have the lowest accuracy compared to the other models. It almost classified half of the data as index 1.

4.5 Model Comparison

After the models have been tested we want to see how they performed. The following figure will illustrate the F1-score from each model on each class. The Random Forest and Gradient Boosting models that have been chosen are the ones with the highest accuracy rate see chapter 4.1 and 4.2.

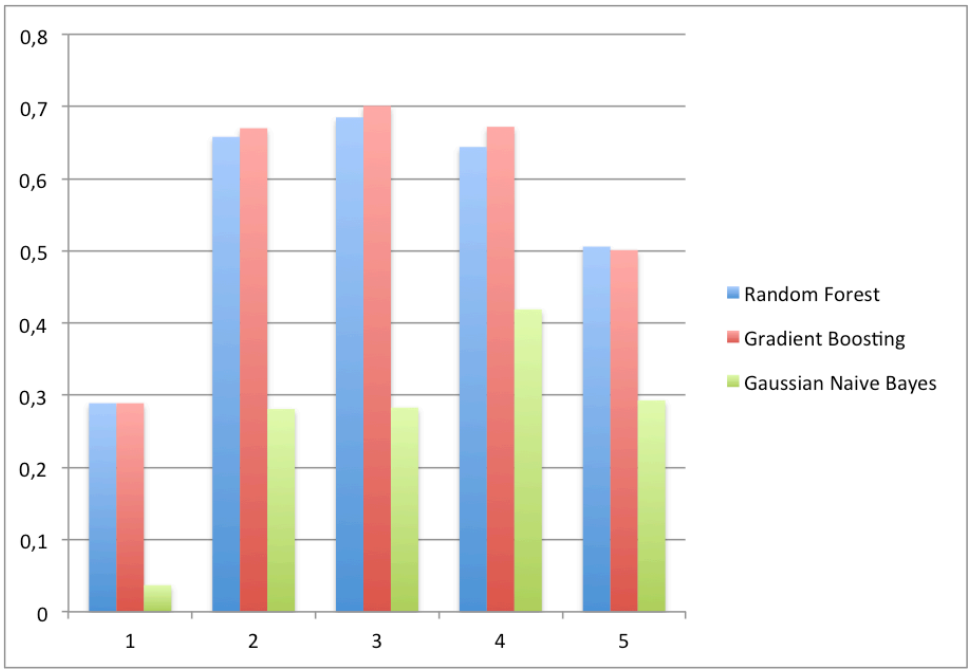

Figure 4.6: F1-score on all ESI levels

On the Y-axis we have the F1-score, on the X axis we have the different ESI levels.From figure 4.6 we can see that the Random Forest and Gradient Boosting model have similar F1-score. The Gradient Boosting model have a little bit better score on index 2-4 than the Random Forest model. The Gaussian Naive Bayes had the worst F1-score on all the classifications compared to the other models.

5 Discussion

In this section we will discuss the performance of the machine learning models used to predict the ESI level. We will also discuss whether these machine learning based triage systems could help to avoid overcrowded EDs and to make faster decisions when triaging patients.

5.1 Machine Learning Models: Key Findings

5.1.1 Random Forest

The Random Forest model was the second best model in terms of accuracy. This model performed somewhat faster than the Gradient Tree Boosting model, but the accuracy was not as good. We noticed that the accuracy increased with the number of trees. This effect is an expected outcome due to the fact that when we increase the number of trees we will have more predictors that predicts the class, thus resulting in a better prediction. This effect is confirmed by Oshiro et al. (2012), however, they also state that after increasing the number of trees over some threshold in the Random Forest model, there will be no significant gain in accuracy, but instead a significant computational lost. Oshiro et al. (2012) statement is confirmed when analyzing table 4.1. When increasing the number of trees from 10 to 25, with a maximum depth set to 10, we can observe a gain in accuracy by 0.4%. When further increasing the number of trees from 25 to 100, with a maximum depth set to 10, we observe no gain in accuracy. In this case there is only a computational loss.

As mentioned in the Evaluation section we want to avoid having many type 2 errors, which means that we want a high recall rate. The recall rate for ESI level 1 patients is 18.51%, which is a poor recall rate see figure 4.1. This could be due to only a small fraction, 0.95%, of our data consisting of ESI level 1 patients. The small fraction of level 1 patients makes it hard to train an accurate model. The highest recall rate, 73.82%, was achieved by the ESI level 3 patients. We can see that 43.09% of the data constitutes of ESI level 3 patients, hence the model has

more data to train on compared to the ESI level 1 patients.

5.1.2 Gradient Tree Boosting

The Gradient Tree Boosting model achieved an accuracy rate of 67.58%, which is the highest accuracy rate achieved by our models. However, the model was also the slowest to train. While tuning the hyperparameters, we noticed that the model became slower to train and that the gain diminished after some point when increasing the number of trees and the maximum depth of the tree. The behaviour is similar to what we observed in the the Random Forest model. This is further confirmed in table 4.3, where we can observe that when increasing the number of trees from 10 to 25 with a maximum depth set to 10 we get an additional 2.58% in accuracy. However, when increasing the number of trees from 50 to 100, with a maximum depth set to 10, we only get an additional 0.8% in accuracy. The computational loss was much greater when increasing the number of trees from 50 to 100, which further consolidates Okashi et al. (2012) statement about computational loss.

When analyzing figure 4.4, we see that for ESI level 1 patients we have a recall rate of 20.40%, a fact that is concerning. However, the recall rate for ESI level 3 patients is 73.90%, which is much better than the recall rate for ESI level 1 patients. If we compare these recall rates with the recall rates received by the Random Forest model we can see that they are similar. The recall rate differs with 0.08% for the ESI level 3 patients and 1.89% for the ESI level 1 patients. This difference could be a result of that Gradient Tree Boosting get better results when having less noisy data.

5.1.3 Gaussian Naive Bayes

The Gaussian Naive Bayes model produced the worst model in terms of accuracy. However, it was the fastest and therefore instantly trained the model. The accuracy rate is 24.58% see figure 4.5 which is close to random chance of 20%. Thus, this model performs nearly as bad as a random predictor. The reason we

have a recall rate of 97.66% for ESI level 1 patients is because the model has predicted that the patient have ESI level 1 about 49130 times. Since our model only contains around 1000 ESI level 1 patients we receive a precision rate of 1.87%.

5.2 Machine Learning Based Triage System in Practice

When it comes to human life, we believe that it is important to have a high accuracy rate and, especially, a high recall rate. Since the Gradient Tree Boosting model, our best model with respect to accuracy rate, achieved an accuracy rate of 67.58% we do not believe that the results are good enough for the model to be used in practice as this would mean that almost 32% of the patients would be incorrectly classified. These patients that have been undertriaged would experience an adverse effect since the medical condition is more acute than predicted. The recall rate for ESI level 1 patients measures the rate of those patients that has been undertriaged. Since the recall rate for ESI level 1 patient is 22.14% it means that nearly 78% of the ESI level 1 patient will be undertriaged. Therefore the model needs to be improved. Patients that are overtriaged results in insufficient use of hospital resources since patients that gets hospitalized may not necessarily need to be hospitalized.

5.3 Disagreement amongst Triage Nurses

A problem in the triaging process is the ED nurses subjective assessment of patients. This is a problem since it is hard to know the correct assessment of patient when there is a disagreement amongst ED nurses. To prevent our model from making subjective prediction we need to train our model based on the correct ESI levels. This could be accomplished through a group of triage experts stratifying patient together.

5.4 Machine Learning Perspectives

5.4.1 Ethical Perspective

A machine learning based triage system is in need of data. Since the data involves sensitive information about patients medical condition, questions about the patients integrity arises. The patients integrity must always be protected and the medical data must be stored in a secure and responsible way. The ED should also ask for the patients confirmation to use the patients medical data for research purposes. If none of the patients confirms then it will be impossible to train a machine learning based model however we believe that people would confirm.

5.4.2 Social Perspective

There is people that do not feel comfortable with automatizing processes that before involved humans. We believe that it becomes especially sensitive when it comes to healthcare, since the decision the machine learning based triage system makes directly affect the patients health. One reason for why people do not trust machines in fully could be that they are not used with these kind of systems. To increase faith in these systems we need to build a social trust by encourage the development of trustworthy machine learning based systems. Since we do not believe that we have come to the point where people fully trust these system we believe that the machine learning based triage system only should work as a support to the ED nurses.

5.4.3 Economical perspective

As previously mentioned patients that is overtriaged results in insufficient use of hospital resources. If our machine learning based triage system do less overtriaging errors than the ED nurses, we have a economic gain. Since the machine learning models stratifies the patients instantly we save time and therefore we can treat more patients more efficiently with respect to time.

5.5 Future Study

To achieve a higher recall rate we recommend collecting more ESI level 1 patients into the dataset since our dataset only consisted of 0.81% of ESI level 1 patients. We also recommend testing this dataset on a high performance computing platform because we could not train the whole dataset. Furthermore we recommend trying different machine learning models such as Artificial Neural Network because it has accomplished great results in related fields according to Raita et al. (2019). Even trying to incorporate several machine learning algorithms into one model could improve the accuracy rate.

An interesting idea is to study the ESI level predicted by the ED nurse and the correct ESI level. Thus we could compare the ED nurses accuracy to the accuracy achieved by the machine learning models.

6 Conclusions

The Gradient Boosting model returned the highest accuracy rate (68%) when asserting the number of trees to 100 and a maximum depth of 10. The Gaussian Naive Bayes model returned the lowest accuracy rate (23%), however it was the fastest one to train. The Random Forest model returned an accuracy rate of 66% when asserting the number of trees to 100 and a maximum depth of 50. Since we did not have computationally strong enough resources we could not train our models for the whole dataset and do all the testing that we wanted. We believe that if we could train our models for the whole dataset we would get a higher accuracy rate. We don't think that the models accuracy is good enough to be used in practice. Further improvements are needed before it could be used in practice.

References

[1] Aacharya, Ramesh P, Gastmans, Chris, and Denier, Yvonne. “Emergency department triage: an ethical analysis”. In: *BMC emergency medicine* 11.1 (2011), p. 16.

[2] Beam, Andrew L and Kohane, Isaac S. “Big data and machine learning in health care”. In: *Jama* 319.13 (2018), pp. 1317–1318.

[3] Chen, Min et al. “Disease prediction by machine learning over big data from healthcare communities”. In: *Ieee Access* 5 (2017), pp. 8869–8879.

[4] Di Somma, Salvatore et al. “Overcrowding in emergency department: an international issue”. In: *Internal and emergency medicine* 10.2 (2015), pp. 171–175.

[5] Farrokhnia, Nasim and Göransson, Katarina E. “Swedish emergency department triage and interventions for improved patient flows: a national update”. In: *Scandinavian journal of trauma, resuscitation and emergency medicine* 19.1 (2011), p. 72.

[6] FitzGerald, Gerard et al. “Emergency department triage revisited”. In: *Emergency Medicine Journal* 27.2 (2010), pp. 86–92.

[7] Friedman, Jerome H. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[8] Geurts, Pierre, Irrthum, Alexandre, and Wehenkel, Louis. “Supervised learning with decision tree-based methods in computational and systems biology”. In: *Molecular Biosystems* 5.12 (2009), pp. 1593–1605.

[9] Gilboy, N et al. “Emergency Severity Index (ESI): a triage tool for emergency department care, version 4”. In: *Implementation handbook* (2012), pp. 12–0014.

[10] Gupta, Prashant. “Decision trees in machine learning–towards data science”. In: *Towards Data Science* (2017).

[11] Hall, Mark A and Smith, Lloyd A. “Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper.” In: *FLAIRS conference*. Vol. 1999. 1999, pp. 235–239.

[12] Hong, Woo Suk, Haimovich, Adrian Daniel, and Taylor, R Andrew. “Predicting hospital admission at emergency department triage using machine learning”. In: *PloS one* 13.7 (2018), e0201016.

[13] James, Gareth et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[14] Koehrsen, Will. “Random Forest Simple Explanation”. In: *Medium* (2017).

[15] Levin, Scott et al. “Machine-learning-based electronic triage more accurately differentiates patients with respect to clinical outcomes compared with the emergency severity index”. In: *Annals of emergency medicine* 71.5 (2018), pp. 565–574.

[16] Liaw, Andy, Wiener, Matthew, et al. “Classification and regression by randomForest”. In: *R news* 2.3 (2002), pp. 18–22.

[17] Manogaran, Gunasekaran and Lopez, Daphne. “A survey of big data architectures and machine learning algorithms in healthcare”. In: *International Journal of Biomedical Engineering and Technology* 25.2-4 (2017), pp. 182–211.

[18] McHugh, Megan et al. “More patients are triaged using the Emergency Severity Index than any other triage acuity system in the United States”. In: *Academic Emergency Medicine* 19.1 (2012), pp. 106–109.

[19] Oshiro, Thais Mayumi, Perez, Pedro Santoro, and Baranauskas, José Augusto. “How many trees in a random forest?” In: *International workshop on machine learning and data mining in pattern recognition*. Springer. 2012, pp. 154–168.

[20] Raita, Yoshihiko et al. “Emergency department triage prediction of clinical outcomes using machine learning models”. In: *Critical Care* 23.1 (2019), p. 64.

[21] Rokach, Lior. “Ensemble-based classifiers”. In: *Artificial Intelligence Review* 33.1-2 (2010), pp. 1–39.

[22] Safavian, S Rasoul and Landgrebe, David. “A survey of decision tree classifier methodology”. In: *IEEE transactions on systems, man, and cybernetics* 21.3 (1991), pp. 660–674.

[23] Shalev-Shwartz, Shai and Ben-David, Shai. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[24] Tanabe, Paula et al. “Reliability and validity of scores on The Emergency Severity Index version 3”. In: *Academic emergency medicine* 11.1 (2004), pp. 59–65.

[25] Tanabe, Paula et al. “The Emergency Severity Index (version 3) 5-level triage system scores predict ED resource consumption”. In: *Journal of Emergency Nursing* 30.1 (2004), pp. 22–29.

[26] Thornton, Chris et al. “Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pp. 847–855.

[27] Travers, Debbie A et al. “Five-level triage system more effective than three-level in tertiary emergency department”. In: *Journal of Emergency Nursing* 28.5 (2002), pp. 395–400.

TRITA-EECS-EX-2019:341