Machine Learning Engineer Nanodegree

Capstone Project

Santiago Giraldo
July 31, 2017

I. Definition

Project Overview

Perhaps one of the trendiest topics in the world right now is machine learning and big data, which have become recurring topics on main street. The application of these in different fields including branches of knowledge that range from astronomy to manufacturing, but perhaps the field of medicine is one the most interesting of them all[1]. Current advances in this field include disease diagnosis, image analysis, new drug developments and pandemic forecasts among others. These progress poses new challenges and opportunities to improve the quality of treatments and the life expectancy of human beings.

Being able to participate in something that contributes to human well-being motivated me to research and apply machine learning in an area of medicine to the capstone project.

From personal experience, a recurring problem that affects people at certain ages is pneumonia, which in some cases is usually fatal especially when it refers to children or older adults. Knowing this and the availability of records in Intensive Care Unit MIMIC-III database, I decided to select pneumonia deaths as a theme to develop the capstone project, that in addition, could be useful for the prognosis of deaths in Intensive Care Units, and with further investigations be the outset for development of tools that doctors and nurses could use to improve their work.

Problem Statement

As mentioned above, the pneumonia can be fatal in some cases. The hypothesis is that one can predict the death probability of said disease by analyzing microbiological variables coming from tests. This leads us to a problem that can be represented binarily, and that can be related to physicochemical variables obtained from microbiological tests. These relationships allow the modeling of the pneumonia death problem by means of a supervised learning model such as the Support Vector Machines, Decision Trees, Logistic Regression and Ada Boost ensemble. The following is a condensed plan of work:

- The first step is ETL (Extract Transform and Load), this part is key in the process, including defining the variables to be extracted from PostgreSQL, convert categorical variables to dummies values, transform matrix data in a pivot table that allows the application of models that will be evaluated.
- The second step is DE (Data Exploration), which goes hand in hand with the ETL process. Here we analyze graphically and statistically the characteristics of the population we want to model, and understand the behavior among its various variables
- The third step is modeling, here the different models are applied to the sample (SVM, DT, LR, ABE) the different metrics are evaluated and the different results are compared.

[1] https://www.aspenideas.org/session/can-artificial-intelligence-revolutionize-medicine?imm_mid=0f48bd&cmp=em-business-na-na-newsltr_econ_20170721

- The fourth step is the conclusion, here the results are analyzed and concluded from the results which is the best model.

Metrics

The metric used to validate the model results is accuracy[2] and it will be defined as:

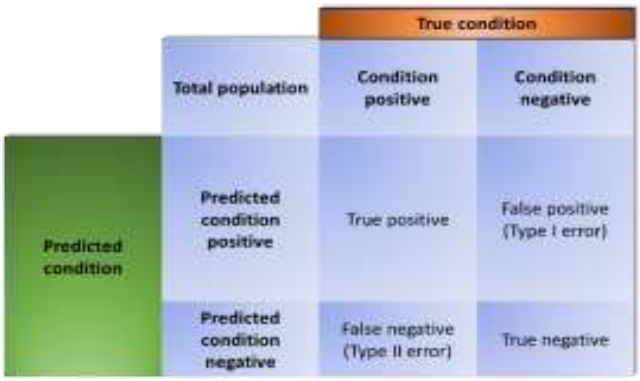
$$Accuracy = \frac{\sum True\ Positive + \sum True\ Negative}{\sum Total\ Population}$$

This measure is used because it is the one used in the benchmark study, but it is convenient to evaluate if the data is balanced, if it is not, it is important to consider other measures such

Another measurement that is used to validate the models is the F1 score, this is recommended when the data are unbalanced[3], this is defined as.

$$F_1 = 2\,\frac{Precision \times Recall}{Precision + Recall}$$

In addition, the confusion matrix[4] will be constructed for each of the models to be analyzed, complementing the results of the accuracy measurement and F_1 score.



II. Analysis

Data Exploration

The data source for this analysis is database MIMIC-III[5] (Medical Information Mart for Intensive Care III) developed by MIT. The database also provides different kind of records of approximately 49,000 patients admitted to Beth Israel Deaconess Medical Center Intensive Care Unit (ICU) recorded between 2001 and 2012. The records from this database contain demographic characteristics such as age, gender, religion, language and ethnicity. In the database

[2] https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers

[3] http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

[4] https://en.wikipedia.org/wiki/Confusion_matrix

[5] Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. Scientific Data (2016). DOI: 10.1038/sdata.2016.35. Available at: http://www.nature.com/articles/sdata201635

are also the records of diagnoses, events and test results, originated by patients during their stay in the ICU.

The database consists of a collection of csv files, which can be imported into PostgreSQL. Once imported to PostgreSQL, it is possible to use libraries from python to analyze the different data contained in it, and make the necessary transformations to implement the desired forecast model. The input variables will be defined based in different tables and seeking relationship between independent binary variable (life or death) of subjects and age, sex, chart events, results of microbiological events (test results) and severity of these results.

Given the limitations of the hardware, I was forced to restrict the data for the analysis (the detail of how the data selection was made will be explained below).

The intensive care unit is a department of the hospitals where people treat life-threatening and require continuous monitoring of doctors, nurses and specialized equipment to monitor your condition and try to prevent his death[6].

The MIMIC III database used for this study contains the historical records of 58,976 admissions to the ICU of the Beth Israel Deaconess Medical Center between 2001 and 2012. There were 46,520 people[7] admitted in total to this ICU during this period. From this people, there are 1,424 who were admitted several times, but lately they died as you can see in table 2.

No. Patients	Alive	Deceased	Total	Marginal
Men	23,797	3,123	26,920	**56.15%**
Women	18,334	2,690	21,024	**43.85%**
Total	**42,131**	**5,813**	**47,944**	**100.00%**
Total [%]	**87.88%**	**12.12%**	**100.00%**	

Table 1. MIMIC III population who were admitted to the ICU

Sex	Patients	Patients who remained alive in all their income to the ICU	Patients admitted to the ICU several times but died her last admission to the ICU
Men	26,121	22,998	799
Women	20,399	17,709	625
TOTAL	**46,520**	**40,707**	**1,424**

Table 2. MIMIC III population who were admitted to the ICU that remained alive after their stay and those who died in their last income

Of this sample 26,121 are men and 20,399 are women, and 40,707 people left the ICU alive and 5,813 people died during their stay there.

[6] https://en.wikipedia.org/wiki/Intensive_care_unit

[7] https://mimic.physionet.org/mimictables/patients/

No. Patients	Alive	Deceased	Total	Marginal
Men	22,998	3,123	26,121	**100.00%**
Women	17,709	2,690	20,399	**43.85%**
Total	**40,707**	**5,813**	**46,520**	**100.00%**
Total [%]	**84.91%**	**12.12%**	**100.00%**	

Table 3. MIMIC III population by gender and survival summary

As I mentioned earlier, this study focuses on analyzing the relationship between the microbiological events of the pneumonia and the survival of patients entering the ICU. From the MIMIC III database, 2,257 patients were diagnosed for some type of pneumonia (21 types of pneumonia diagnoses Fig. 1)[8], and the total number of diagnoses for this population is 2,572, which means that some patients had one or more diagnoses in admissions to the ICU.

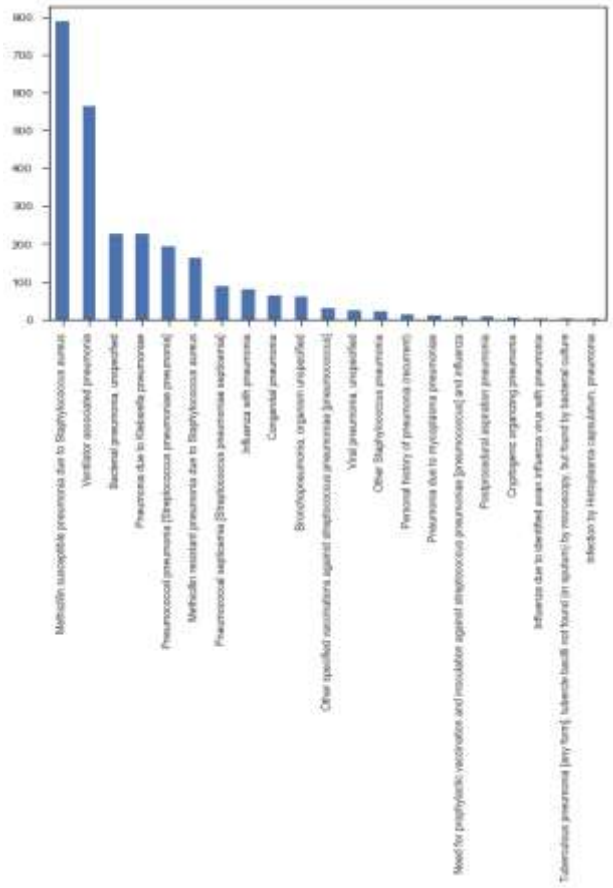


Figure 1. Pneumonia diagnosis.

According to the records of the database, anyone who has pneumonia and is admitted to an ICU has an 18.74% chance of dying while receiving treatment. And the deadliest types of pneumonia, or rather, with which there is a greater risk or probability of dying, are: Methicillin susceptible pneumonia due to Staphylococcus aureus (5.76%) and Ventilator associated pneumonia (4.82%).

[8] On this website, you can find a better explanation of pneumonia, its origins, types and other details of the illness: http://www.healthline.com/health/pneumonia#types-and-causes3

Patients who have had these diagnoses, 18.76% and 21.99% respectively, have died during their treatment at the ICU.

Diagnosis	Alive	Deceased	Marginal	Alive[%] | Long_Title	Deceased [%] | Long_Title	Alive[%]	Deceased [%]	Marginal [%]
Methicillin susceptible pneumonia due to Staphylococcus aureus	641	148	789	81.24%	18.76%	24.92%	5.75%	30.68%
Ventilator associated pneumonia	440	124	564	78.01%	21.99%	17.11%	4.82%	21.93%
Pneumonia due to Klebsiella pneumoniae	177	49	226	78.32%	21.68%	6.88%	1.91%	8.79%
Methicillin resistant pneumonia due to Staphylococcus aureus	131	31	162	80.86%	19.14%	5.09%	1.21%	6.30%
Bacterial pneumonia, unspecified	198	29	227	87.22%	12.78%	7.70%	1.13%	8.83%
Bronchopneumonia, organism unspecified	33	26	59	55.93%	44.07%	1.28%	1.01%	2.29%
Pneumococcal pneumonia [Streptococcus pneumoniae pneumonia]	171	23	194	88.14%	11.86%	6.65%	0.89%	7.54%
Influenza with pneumonia	59	21	80	73.75%	26.25%	2.29%	0.82%	3.11%
Pneumococcal septicemia [Streptococcus pneumoniae septicemia]	72	16	88	81.82%	18.18%	2.80%	0.62%	3.42%
Other Staphylococcus pneumonia	17	4	21	80.95%	19.05%	0.66%	0.16%	0.82%
Personal history of pneumonia (recurrent)	10	3	13	76.92%	23.08%	0.39%	0.12%	0.51%
Cryptogenic organizing pneumonia	2	2	4	50.00%	50.00%	0.08%	0.08%	0.16%
Influenza due to identified avian influenza virus with pneumonia	2	1	3	66.67%	33.33%	0.08%	0.04%	0.12%
Need for prophylactic vaccination and inoculation against streptococcus pneumoniae [pneumococcus] and influenza	7	1	8	87.50%	12.50%	0.27%	0.04%	0.31%
Postprocedural aspiration pneumonia	7	1	8	87.50%	12.50%	0.27%	0.04%	0.31%
Pneumonia due to mycoplasma pneumoniae	8	1	9	88.89%	11.11%	0.31%	0.04%	0.35%
Viral pneumonia, unspecified	22	1	23	95.65%	4.35%	0.86%	0.04%	0.89%
Congenital pneumonia	62	1	63	98.41%	1.59%	2.41%	0.04%	2.45%
Infection by Histoplasma capsulatum, pneumonia	1	0	1	100.00%	0.00%	0.04%	0.00%	0.04%
Other specified vaccinations against streptococcus pneumoniae [pneumococcus]	29	0	29	100.00%	0.00%	1.13%	0.00%	1.13%
Tuberculous pneumonia [any form], tubercle bacilli not found (in sputum) by microscopy, but found by bacterial culture	1	0	1	100.00%	0.00%	0.04%	0.00%	0.04%
TOTAL	**2,090**	**482**	**2,572**	**81.26%**	**18.74%**	**81.26%**	**18.74%**	**100.00%**

Table 4. Pneumonia patients.

The sample is reduced from 2.257 to 829 subjects when filtering the microbiological events that only have interpretation (non-null), categorized (non-null) laboratory results and the dates of admission to the ICU are the last recorded in the database, in order to have the most complete records data for the analysis.

For each subject, I categorized the population in five types of groups according to the age recorded at the time of admission to the ICU, which are neonates [0,1], middle (1, 14), adults (14, 65), Older adults [65, 85] and older elderly people (85, 91.4].

Reading this, you will wonder why the last age is 91.4, the reason for this is because those who created the database they replaced the age of all of those who were 89 years or older by 300 years in order to protect their privacy. In the Website[9] they explain that the average age of these patients is 91.4 years, reason why I decided that if I want have some consistent data from this segment of the population I should replace it at least for its average value.

Group	Patients
Neonate	3
Middle	0
Adult	386
Elderly	356
Oldest old	84
Total	**829**

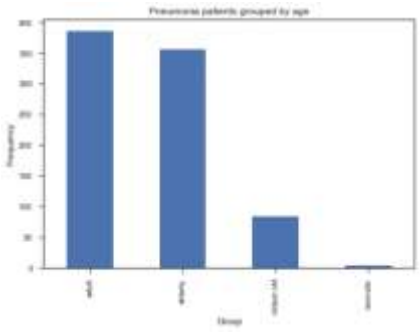


Figure 2. Pneumonia patients grouped by age.

As you can see, there is no one in the middle group[10] because in general the data base doesn't contain data from pediatric patients, even more, the minimum age in Adult segment is 19 years old and the main concentration in this sample concentration are in Adult and Elderly persons whit pneumonia.

Gender	Age group	Number of Patients			Percentage of Patients		
		Alive	Deceased	Total	Alive [%]	Deceased [%]	Total [%]
Feminine	adult	104	31	**135**	12.55%	3.74%	**16.28%**
	elderly	109	54	**163**	13.15%	6.51%	**19.66%**
	neonate	1	0	**1**	0.12%	0.00%	**0.12%**
	oldest old	34	13	**47**	4.10%	1.57%	**5.67%**
Masculine	adult	197	54	**251**	23.76%	6.51%	**30.28%**
	elderly	134	59	**193**	16.16%	7.12%	**23.28%**
	neonate	2	0	**2**	0.24%	0.00%	**0.24%**
	oldest old	22	15	**37**	2.65%	1.81%	**4.46%**
TOTAL		**603**	**226**	**829**	**72.74%**	**27.26%**	**100.00%**

Table 5. Population features.

[9] https://mimic.physionet.org/mimictables/patients/
[10] https://mimic.physionet.org/tutorials/intro-to-mimic-iii/

The table above, show the population composition by gender, age and deceases. As you can see, the adult population is 46.6% or 386 persons, followed by elderly group which has 42.9% or 356 and oldest old group with 10.1% or 84 persons, and as I said above, the middle group doesn't have any person, and the neonate population is represented by 0.4% or 3 persons. The sample consists of 42% women and 58% men, it is 346 and 483 persons respectively. The people is concentrated adult and elderly groups, as you can see in next figures.

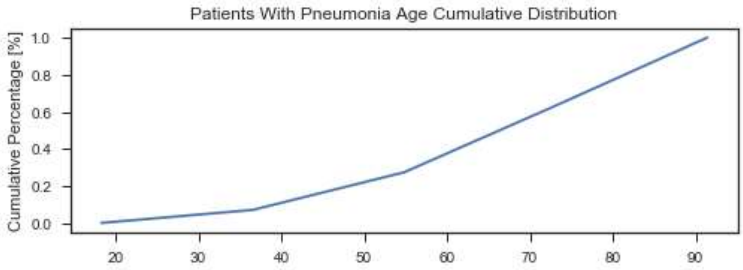


Figure 3. Cumulative Distribution by age.

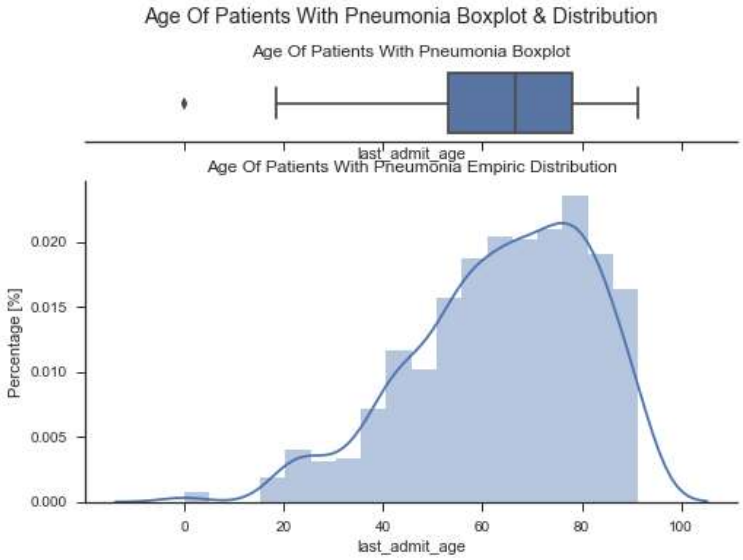


Figure 4. Cumulative Distribution by age.

The sample has a mean of 64.20 years and a standard deviation of 17.74 years, where the 75% has more than 53 years, this is showing more vulnerable population with pneumonia in ICU is a middle age person or older. The above empirical distribution is showing a negative asymmetry with shape skewed to left side.

Patients	mean	std	min	25%	50%	75%	max
829	64.20	17.74	0	53.11	66.55	78.21	91.4

Table 6. Sample statistical summary.

The next graphs confirm this evidence when you see the bivariate distributions between age (last_admit_age) and gender, and age (last_admit_age) and decease (hospital_expire_flag). The left graph, show the gender distribution (Masculine = 0, Feminine = 1) vs. age. This graph shows a higher dispersion in men, and higher concentration in women, it is interesting, is that men are more prone to acquire acute pneumonia being admitted the ICU. The following graph shows the relationship between age and deaths (Alive = 0, Deceased = 1), showing a similar distribution, but in this case the people most likely to die in the ICU are those who are in the elderly group.

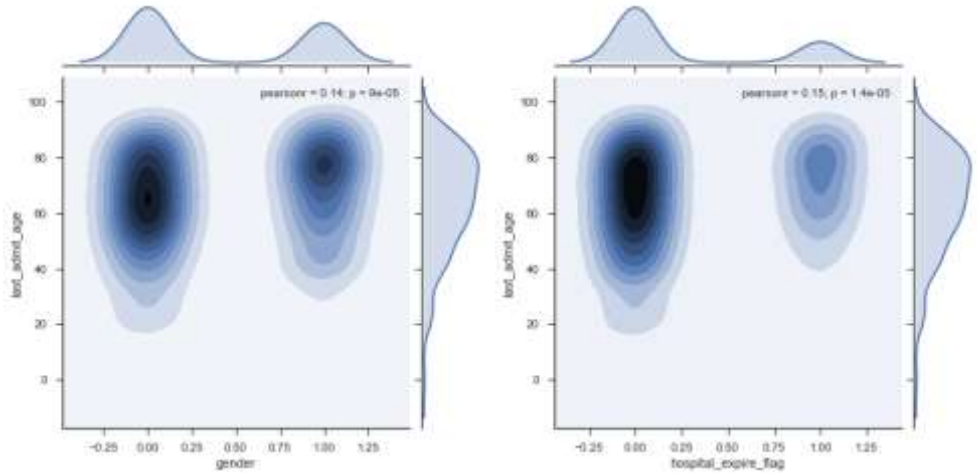


Figure 5. Bivariate Distribution age vs gender and age vs. deceases.

The neonate group of patients has no deaths in this sample, all of deaths in this sample are more related to adult people, with higher concentration in people that are 60 years or older. This relationship must be collected by the model given the assumptions formulated by the hypothesis.

Algorithms and Techniques

The problem here is a classification problem, in this vein, a recommended solution is applying a supervised learning technique such as vector support machines. This algorithm has the advantage of being effective to treat problems with high dimensional spaces, it is efficient in memory when dividing the data into subsets called support vectors, and it is also versatile because it allows to use several kernels in the decision functions[11]. Besides these methods, I want explore other methods more classics in statistics, like logistic, decision trees and random forests classifiers algorithms and others like Extremely Randomized Trees and ada boost classifier. As all models fit among classifier types, I think this could be a good opportunity to compare their performances with SVM classifiers. Next chart outlines the pipeline used in this study.

Figure 6. Study pipeline.

[11] http://scikit-learn.org/stable/modules/svm.html

Benchmark

As benchmark paper published by Youn-Jung Son et al (2010)[12] is used, in which a support vector machine in predicting medication adherence in patients with heart failure is applied. The model was validated using the accuracy metric.

The importance of this work as a benchmark for this study is that they used a supervised learning model (SVM) to predict the way in which patients who had some type of heart failure follow their medication treatment. Although the idea of the proposed study is not to repeat this same experiment, this study serves as background for the application of machine learning to medicine.

III. Methodology

Data Preprocessing

The MIMIC III database consists of a collection of csv files, which can be imported into PostgreSQL. Once imported to PostgreSQL, is possible to use libraries from python to analyze the different data contained in it, make the necessary transformations to implement the desired forecast model. The input variables will be defined from different tables and seeking relate the independent binary variable (life or death) of subjects with age, sex, results of microbiological events (test results) and severity of these results.

Before you can download the data, you must complete the CITI "Data or Specimens Only Research" course[13]. Once you accomplish the course you can download the data from https://physionet.org/works/MIMICIIIClinicalDatabase/. Then, the first step was to understand the structure of the database. This consists of a collection of 26 csv files, which can be imported into PostgreSQL[14]. These files contain medical, economic, demographic and death of patients admitted information for several years at the ICU of Beth Israel Deaconess Medical Center, as the data is sensitive, some records were changed like date of admittance and date of birth, in order to avoid the identification of the patients from these records, and this information will be misused in the future.

As I said earlier, the objective of this study is to model through supervised learning classification models the relationship of patient deaths in ICU due to pneumonia. So, I must find the data that let me construct the right frame to the analysis. The first step was creating four tables to facilitate consult the required data for the analysis, these tables are:

- last_event: This table born from a join of patients and admissions tables. In this, was selected the fields subject_id, dob, and gender. The age is computed for all patients, the last admission column is created and all age are classified by age groups as categorical variable.
- age: Is a join between last_event and admission tables. In this, I selected the subject_id, last_admit_age, gender, last_admit_time, but the records are limited to last patient admission (there are records for several admissions for some patients, so is important filter the last one to related the records with deaths when these occur) computed in last_event table.
- valuenum_avg: In a first instance, I have grouped the 14 tables that have the data records of graphical events. As a group, it is the largest table in the database and it contains

[12] Youn-Jung Son et al. Application of Support Vector Machine for Prediction of Medication Adherence in Heart Failure Patients. Healthcare Informatics Research. 2010 December;16(4):253-259. doi: .4258/hir.2010.16.4.253. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3092139

[13] https://mimic.physionet.org/gettingstarted/access/

[14] https://mimic.physionet.org/tutorials/install-mimic-locally-ubuntu/

330,712,483 records. Given the size of the data, hardware constraints, I considered a strong assumption, and is that the records in this table where the numerical value (valuenum) of these graphic events are measured, can be averaged (huge assumption) can serve as a numerical dependent variable within the models to be studied. It is a huge assumption because I have no evidence from other studies, at least as far as I know, the results average can be done. But on the other hand, you can think by experience (as patient because I'm not physician), the results from exams are a good estimation, and the issue, at least for me, is if this data could be averaged as I did and if it could be a good proxy regressor. For this table, I take this data: subject_id, hadm_id, itemid, and compute valuenum_avg.

- pneumonia: It is the most important table for this study because I group the relevant data from others tables like microbiology events, charted events, and demographic data. The specific fields grouped here are: hospital_expire_flag, subject_id, hadm_id, gender, last_admittime, last_admit_age, age_group, itemid, label, category, valuenum_avg, icd9_code, short_title, spec_type_desc, org_name, ab_name, interpretation. And this data where filtered by pneumonia word in long_title diagnosis field, values not null in interpretation in microbiology events, values not null in category laboratory items and admittime is equal to last_admittime. The objective here is assuring me that data is complete (not null records), is related with pneumonia diagnosis and the records selected where from the last admission.

I must emphasize something here when I talk about selecting pneumonia from the diagnoses of patients, is when I filter by a word I am skewing the sample, because a patient may have other diagnoses besides the one I mentioned earlier and that filter omits. The reason for doing this on purpose and not adding the data associated with other diseases that patients suffer in addition to pneumonia, is due to the restriction of capacity in the hardware that forces me to try to mix the data in such a way that the algorithms could run on my laptop.

The final result of these process is a sql query which filter and transform the data in a matrix with this columns fields: hospital_expire_flag, subject_id, gender, last_admit_age, age_group, category, label, valuenum_avg, org_name, ab_name, interpretation, and 2,430,640 records to 829 patients with some diagnosis related with pneumonia panda data frame. Each variable is explained next:

Hospital_expire_flag has the binary dependent variable to the model, 0 when the patient goes out from ICU alive, and 1 when the patient has deceased while stay in ICU. Subject_id is the key value which relates the respective record with an acute patient in ICU. gender give the patient sex of the subject. Last_admit_age (is a computed field) has the age when the patient is admitted in ICU. Age_group (is a computed field) serves to categorize the sample by age. Category is employed to categorize the charted events, the main categories of this data column are Labs, Respiratory, Alarms, Routine Vital Signs, Chemistry which gathering the 82% of the records which are present in this query. Label is the detail of the category, and is represented in 578 labels, where the most important are: Hemoglobin Arterial Base Excess, Phosphorous, WBC, Creatinine, Magnesium, PTT, INR, ALT, AST, Lactic Acid. And the largest amount (Hemoglobin Arterial Base Excess) represents 0.94% of the sample and the lowest (Lactic Acid) 0.82% of the sample. Valuenum_avg is the average number for valuenum of this respective label measure. Org_name contains the names of the microorganisms (bacteria) related to pneumonia, where the main ones are staph aureus coag +, klebsiella pneumoniae, escherichia coli, pseudomonas aeruginosa, staphylococcus, coagulase negative, klebsiella oxytoca, enterococcus sp, which represent 80% of the sample. Ab_name indicates which antibiotic is sensitive the microorganism, this field together with the interpretation indicates if the microorganism the degree of resistance of this one to the antibiotic., the main antibiotics evaluated are gentamicin, trimethoprim/sulfa, levofloxacin, ceftazidime, tobramycin, cefepime, ciprofloxacin, meropenem, erythromycin, oxacillin, vancomycin, ceftriaxone, tetracycline, clindamycin, piperacillin/tazo, which represent 80% of the

sample. Finally, there is interpretation indicates the results of the test, "S" when the antibiotic is sensitive, "R" when is resistant, "I" when the antibiotic is intermediate, and "P" when is pending.

Implementation

The last query must be transformed in a pivot table, to associate the alive/death event by pneumonia with all information in a single line. The panda data frame has 2,430,640 rows by 10 columns so, I need reduce the rows from 2,430,640 to 829 and transpose the rest of the data into columns like dummy or categorical variables.

To transform the matrix in a pivot table, the first step is transform some categorical variables as dummy variables. The chosen variables are gender, age_group, category, label, org_name, ab_name, and interpretation. This operation was done with pandas get_dummies command. The result of this transformation is a panda data frame with shape 2,430,640 rows by 716 columns, these new columns are binaries variables and only take the number 1 once the categorical effect happened.

The next step, is to transform the matrix into a PivotTable, the purpose of this transformation is to be able to have the medical data in individual lines per subject and numerical form.

To do that, I employed pandas pivot_table command, and using as indexes subject_id and hospital_expire_flag. With this transformation, the resulting panda data frame has 829 rows by 724 columns. The data transformed in this form allow apply the classifier models to this data.

As I´m mentioned above, I´ve chosen work with different classifier methodologies. For every methodology applied to the data, I used grid search together with cross validation capabilities from scikit-learn library, to find the best solution for a set of parameters previously defined and get consistency at the time of evaluating the results of the different models.

The models applied to the data are:

- Support Vector Machine: Machine Support Vector (SVM) is a classification method that separates a sample points in different hyperplanes in multidimensional spaces, which are separated by different labels. The algorithm seeks to group the data (classify) by optimal search (minimum distance) between the hyperplanes, the resulting vectors are called support vectors[15]. The optimization is made over kernels (mathematical functions), in this analysis I used different methods: linear, radial basis function (rbf) and sigmoid. I purposely avoided using the polynomial kernels, more because of parameterization problems that did not allow me to run the data with this algorithm.

  All kernels were run with grid search function and parameters used in this function are:

  C: [1e-3, 1e-2, 1, 1e3, 5e3, 1e4, 5e4, 1e5]
  gamma: [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1, 1]
  coef0: [-1,0,1]

  C were used with linear function. C and gamma parameters were used with rbf kernel, and shortened vector of C (1e3, 1e4, 1e5) and coef0 were used with sigmoid kernel.

  C value is a penalty parameter of the error, the value of this parameter is too high could overfit the model, but it is too low it could underfit the training model[16]. Gamma is a factor $\gamma = \frac{1}{2\sigma^2}$ which affect the squared Euclidian distance and define the distance that a

[15] http://www.statsoft.com/textbook/support-vector-machines
[16] http://www.svms.org/parameters/

training set can reach, high value is close and low value is far[17], this parameter only is applied in polynomial, rbf and sigmoid kernels. While coef0 is an "*independent term in kernel function*[18]"

You must note, that grid search is useful because allowing permute different combinations to run in these models and find the best solution among its results. This give flexibility in code execution. The same methodology is applied in other methods used in this study.

- Decision Tree Classifier is a no parametric method that learns through binary decisions that when deployed are forming a decision tree. A decision tree classifier is a sequential methodology in which from several (many or many) attributes, it evaluates each situation according to the values of the attributes at each decision point (node), which in the case of this study would be the result of whether a person remains alive or not, according to whether the person is Male or Female, or if this person is Old or newborn or if the bacteria that the person has is resistant to antibiotics or not, just to name a few attributes[19]. The process is recursive and uses algorithm that optimize local decisions. As the decisions are taken, and if they were put on paper, the form that the graph acquires is that of a tree that grows organically with a given decision. The depth of the steps or nodes in which they make the decisions serve to find the local optimum. The grid search applied to this model were max depth with this range [2, 3, 4, 5, 6, 7].

- Ensemble methods like Random Forest, Extremely Tree and Ada Boost Classifiers. These methods "*combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator*[20]". The random forest is a combination of decision trees from random sample sub-setting such that each tree depends on the values of a random vector independently tested and with the same distribution for each of these[21], the decision trees are averaged to reduce the volatility in the results. The extremely random forest is like the random forest, but this one differentiates those in the sense that in addition to randomizing the sample's subset, the point where one tree is cut-off and starts to grow the other is completely random[22] (classical random trees have definite depths), which makes it completely random. For both algorithms, the estimators are the numbers of trees in the forest and the max_depth is the maximum depth allowed to the tree.

  Both methods are averaging methods, where independent estimators are used over random samples and resulting predictions are averaged, getting as result a lower variance than a single estimator. The parameters used for these models in grid search are:

  n_estimators: [3,5,7,10]
  max_depth: [2, 3, 4, 5, 6,7]

  The next model is part of boosting methods[23]. These methods start from a base estimator which sequentially built and one of these estimators try to reduce the bias of combined estimators. The adaboost or adaptative[24] boost method is a method that takes a sample of the training set, with weak relationships and assigns weights to the elements of the set. For the available characteristics of the set, the algorithm trains a classifier using only one

[17] https://globaljournals.org/GJCST_Volume13/3-Performance-Evaluation.pdf
[18] http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[19] ttp://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/lguo/decisionTree.html
[20] http://scikit-learn.org/stable/modules/ensemble.html#forest
[21] https://link.springer.com/article/10.1023%2FA%3A1010933404324
[22] http://www.montefiore.ulg.ac.be/~ernst/uploads/news/id63/extremely-randomized-trees.pdf
[23] http://scikit-learn.org/stable/modules/ensemble.html#forest
[24] https://en.wikipedia.org/wiki/AdaBoost

characteristic and evaluates the formation error, the one with the smallest error is chosen, once this is recalculated the weights. It increases the weight if the classifier erroneously classifies the dependent variable and decreases if it does it correctly. This process is iterative until finally obtaining a robust classifier. The parameters used in this case are:

n_estimators: [3,5,10]
learning_rate: [0.01]

The learning rate scale the gradient descent step length used in optimization process.

- Logistic Regression Classifier is the most traditional method applied to classification problems. Here a logistic probability function (Sigmoid Function) is applied to data. This function always produces a result between [0,1], the result obtained is a probability of occurrence of the binary categorical variable[25], in this case the state alive (0) or the state dead (1). The optimization problem can be adjusted with a gradient descent algorithm. The variables used to grid search are:

  C: [1e-3, 1e-2, 1, 1e3, 5e3, 1e4, 5e4, 1e5],

Is important say when it is possible I parametrize the "class_weight" parameter as "balanced", because the classes are imbalanced, these are not homogeneous.

As I mentioned above, the metrics used in this study are accuracy and f1 score. At the beginning, I tried to set as score measure f1 in cross validation score function, but I find in some cases the result is indeterminate (divided by zero), so I left the score function parameters as default. Instead I used classification score report as alternative to get the scores (Accuracy and f1) when was possible.

IV. Results

Model Evaluation and Validation

What follows are the best models obtained for each methodology applied to the data. In all models, the variable dependent is survival state (Alive / Deceased). In order to sub-setting the data I work with a test size of 25% of the sample, I chose this value after some essays, a higher percentage could lower the computer velocity, and a higher value could make the results will be spurious or meaningless.

The cross-validation score is getting from training data, and as I mentioned above, the metric used with this function was the default value accuracy. The metric score report was applied to test data.

SVM whit a kernel RBF best estimator found by grid search:

- LinearSVC(C=0.001, class_weight=None, dual=True, fit_intercept=True, intercept_scaling=1, loss='hinge', max_iter=1000, multi_class='ovr', penalty='l2', random_state=None, tol=0.0001, verbose=0)

  Cross validation score: [0.71794872, 0.72258065, 0.72258065, 0.72258065]

  Accuracy: 0.72 (+/- 0.00)

[25] https://en.wikipedia.org/wiki/Logistic_regression

Linear Support Vector Classification - Hingue loss classifier best estimator found by grid search:

- LinearSVC(C=0.001, class_weight=None, dual=True, fit_intercept=True, intercept_scaling=1, loss='hinge', max_iter=1000, multi_class='ovr', penalty='l2', random_state=None, tol=0.0001, verbose=0)

  Cross validation score: [0.72435897, 0.72258065, 0.72258065, 0.72258065]

  Accuracy: 0.7230 (+/- 0.0015)

SVM whit a kernel Sigmoid classifier b est estimator found by grid search:

- SVC(C=1000.0, cache_size=200, class_weight='balanced', coef0=1, decision_function_shape=None, degree=3, gamma=0.001, kernel='sigmoid', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

  Cross validation score: [0.62179487, 0.67096774, 0.57419355, 0.60000000]

  Accuracy: 0.6167 (+/- 0.0711)

Decision Tree Classifier best estimator found by grid search:

- DecisionTreeClassifier(class_weight='balanced', criterion='entropy', max_depth=7, max_features=None, max_leaf_nodes=None, min_impurity_split=1e-07, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort=False, random_state=123, splitter='best')

  Cross validation score: [0.60897436, 0.6516129, 0.61935484, 0.70322581]

  Accuracy: 0.6458 (+/- 0.0734)

Random Forest Classifier best estimator found by grid search:

- RandomForestClassifier(bootstrap=True, class_weight='balanced', criterion='gini', max_depth=7, max_features='auto', max_leaf_nodes=None, min_impurity_split=1e-07, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1, oob_score=False, random_state=123, verbose=0, warm_start=False)

  Cross validation score: [0.69230769, 0.67096774, 0.66451613, 0.61935484]

  Accuracy: 0.6618 (+/- 0.0531)

Extremely Tree Classifier best estimator found by grid search:

- ExtraTreesClassifier(bootstrap=False, class_weight='balanced', criterion='gini', max_depth=5, max_features='auto', max_leaf_nodes=None, min_impurity_split=1e-07, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1, oob_score=False, random_state=123, verbose=0, warm_start=False)

  Cross validation score: [0.72435897, 0.67741935, 0.63870968, 0.64516129]

Accuracy: 0.6714 (+/- 0.0678)

Ada Boost Classifier best estimator found by grid search:

- AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None, learning_rate=0.01, n_estimators=3, random_state=123)

  Cross validation score: [0.72435897, 0.72258065, 0.72258065, 0.72258065]

  Accuracy: 0.7230 (+/- 0.0015)

Logistic Regression Classifier best estimator found by grid search:

- LogisticRegression(C=0.001, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1, penalty='l2', random_state=123, solver='liblinear', tol=0.0001, verbose=0, warm_start=False)

  Cross validation score: [0.72435897, 0.72258065, 0.72258065, 0.72258065]

  Accuracy: 0.7230 (+/- 0.0015)

The following table summarizes the cross-validation scores from all models. The summary shows interesting results. The best model is SVM - kernel Radial Basis Function classifier, but some can argue that other methods performance better scores, but here the results are showing weird values, all are the same to Linear Support Vector Classification - Hingue loss, Ada Boost Classifier, Logistic Regression Classifier. They have the same average, standard deviation and same scores, this let me conclude theses implementations are untrusted because it is not normal have the same results.

Model	μ	σ	Cross-validation score
SVM - kernel Radial Basis Function classifier	0.72	+/- 0.00	[0.71794872 0.72258065 0.72258065 0.72258065]
Linear Support Vector Classification - Hingue loss	0.723	+/- 0.0015	[0.72435897 0.72258065 0.72258065 0.72258065]
SVM whit a kernel Sigmoid classifier	0.6167	+/- 0.0711	[0.62179487 0.67096774 0.57419355 0.60000000]
Decision Tree Classifier	0.6458	+/- 0.0734	[0.60897436 0.6516129 0.61935484 0.70322581]
Random Forest Classifier	0.6618	+/- 0.0531	[0.69230769 0.67096774 0.66451613 0.61935484]
Extremely Tree Classifier	0.6714	+/- 0.0678	[0.72435897 0.67741935 0.63870968 0.64516129]
Ada Boost Classifier	0.723	+/- 0.0015	[0.72435897 0.72258065 0.72258065 0.72258065]
Logistic Regression Classifier	0.723	+/- 0.0015	[0.72435897 0.72258065 0.72258065 0.72258065]

Table 7. Cross-validation score summary.

The following table show the results from classification_report, summarizing the main score metrics applied to models.

Model	Variable	precision	recall	f1-score	support
SVM - kernel Radial Basis Function classifier	**0**	0.75	0.99	0.85	154
	1	0.50	0.04	0.07	54
	avg / total	**0.63**	**0.52**	**0.46**	**208**
Linear Support Vector Classification - Hingue loss	**0**	0.74	1.00	0.85	154
	1	0.00	0.00	0.00	54
	avg / total	**0.37**	**0.50**	**0.43**	**208**
SVM whit a kernel Sigmoid classifier	**0**	0.72	0.68	0.70	154
	1	0.22	0.26	0.24	54
	avg / total	**0.47**	**0.47**	**0.47**	**208**
Decision Tree Classifier	**0**	0.84	0.64	0.73	154
	1	0.39	0.65	0.49	54
	avg / total	**0.62**	**0.65**	**0.61**	**208**
Random Forest Classifier	**0**	0.79	0.75	0.77	154
	1	0.38	0.44	0.41	54
	avg / total	**0.59**	**0.60**	**0.59**	**208**
Extremely Tree Classifier	**0**	0.83	0.81	0.82	154
	1	0.48	0.52	0.50	54
	avg / total	**0.66**	**0.67**	**0.66**	**208**
Ada Boost Classifier	**0**	0.74	1.00	0.85	154
	1	0.00	0.00	0.00	54
	avg / total	**0.37**	**0.50**	**0.43**	**208**
Logistic Regression Classifier	**0**	0.74	1.00	0.85	154
	1	0.00	0.00	0.00	54
	avg / total	**0.37**	**0.50**	**0.43**	**208**

0: The patient stays alive,
1: The patient is deceased

Table 8. Classification report.

This table shows different conclusions that obtained previously, here the best model are Decision Tree and Extremely Tree Classifier, they have the best score performance (f1 score included). It is important to note that the same three models produce the same result. I think this is no coincidence, and that they cannot be taken into account in the analyzes. It is possible[26] to verify

[26] I think it is not worth showing these results because they do not contribute to the analysis, this statement can be checked using the command confusion_matrix (y_test, y__model_predicted)

that these three models do not predict deaths (neither true nor false), which makes these models useless when using them.

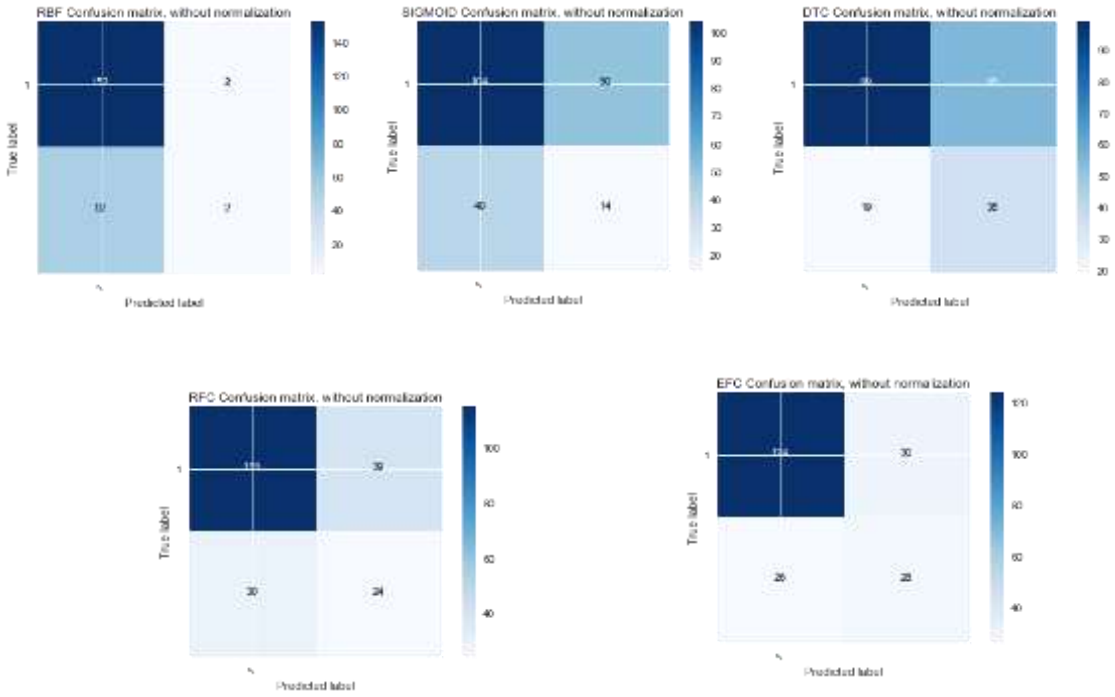


Figure 7. Confusion Matrix. Left to right: SVM rbf, SVM sigmoid, Decision Tree, Random Forest and Extremely Tree models.

Next tables show the confusion matrix without normalization and with normalization respectively.

	RBF		SIGMOID		DTC		RF		ER	
	P	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**
P	152	2	104	50	99	55	115	39	124	30
N	52	2	40	14	19	35	30	24	26	28

Table 9. Confusion matrix, without normalization.

	RBF		SIGMOID		DTC		RF		ER	
	P	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**
P	99%	1%	68%	32%	64%	36%	75%	25%	81%	19%
N	96%	4%	74%	26%	35%	65%	56%	44%	48%	52%

Table 10. Normalized confusion matrix

The confusion matrix results explain better the classification report (this makes sense since precision, recall and f1 score are in function of the results of this matrix), The best performance is achieved by the decision tree and the extremely tree models.

Another approach is using an ensemble voting classifier, aggregating the models used in this study. The first ensemble model aggregates the five models used above. The second aggregates

decision tree and extremely tree models. The following table shows the cross-validation results for the voting models. The metric accuracy score of both sets of models are better than those obtained in previous analyzes. Is important note that the performance of both models is practically the same.

Model	μ	σ	Cross-validation score
Voting 5 models	0.7149	+/- 0.0564	[0.74, 0.74, 0.67, 0.71]
Voting 2 best models	0.7133	+/- 0.0529	[0.74, 0.72, 0.67, 0.72]

Table 11. Voting cross-validation score summary.

The classification report and confusion matrix, are consequent with the above results, the ensemble models improve on true negatives (deceases) forecasting. And the average metrics results perform better than individual models result too.

	Variable	precision	recall	f1-score	support
Voting 5 models	0	0.81	0.88	0.84	154
	1	0.55	0.43	0.48	54
	avg / total	**0.74**	**0.76**	**0.75**	**208**
Voting 2 best models	0	0.82	0.86	0.84	154
	1	0.54	0.48	0.51	54
	avg / total	**0.75**	**0.76**	**0.75**	**208**

Table 12. Voting classification report

	Voting 5 models		Voting 2 best models	
	P	**N**	**P**	**N**
P	135	19	132	22
N	31	23	28	26

Table 13. Voting confusion matrix, without normalization

Predicted Class

	Voting 5 models		Voting 2 best models	
	P	**N**	**P**	**N**
P	88%	12%	86%	14%
N	57%	43%	52%	48%

Table 14. Voting confusion matrix, without normalization

V. Conclusion

The reference study employed a support vector machine model for Prediction of Medication Adherence in Heart Failure Patients, the subject matter is pretty different that studied here, for that reason compare results are not reasonable option, not only because the subject of analysis was different, but as far as I know this study is unique. The benchmark is made over a small sample, they using almost the same metrics, and the study only explore SVM methods with kernels.

The best model found here is the ensemble model with the decision tree and the extremely tree, even though the ensemble model with five aggregate methods shows a slightly better score, applying the principle of Occam's razor make the simplest better. At this time, the resulting model can accurately predict the deaths given a series of medical examinations, as proposed in the hypothesis. While accuracy is not the best (76%), I think it may be a good start for future investigations of interdisciplinary teams in ICU forecasting diseases.

From the forest model is possible to find how features weights in the results, such weights are called importance. From this analysis, only 129 features are important to the model, the rest has no weights, on table 15 and figure 8 are show the main features of the analysis. As you could expect, the antibiotic sensitivity is the most important feature (together weights add 57% of importance) and AMIKACIN antibiotic is the most important feature of the sample. Every feature from age group weight 0.97% in average, followed by category feature which everyone weights less than 1%.

Idx	Feature	Importance	Idx	Feature	Importance
0	ab_name_AMIKACIN	5.39%	17	ab_name_MEROPENEM	1.49%
1	ab_name_AMPICILLIN	3.45%	18	ab_name_NITROFURANTOIN	1.47%
2	ab_name_AMPICILLIN/SULBACTAM	3.02%	19	ab_name_OXACILLIN	1.45%
3	ab_name_CEFAZOLIN	2.95%	20	ab_name_PENICILLIN	1.42%
4	ab_name_CEFEPIME	2.93%	21	ab_name_PENICILLIN G	1.41%
5	ab_name_CEFTAZIDIME	2.78%	22	ab_name_PIPERACILLIN	1.34%
6	ab_name_CEFTRIAXONE	2.53%	23	ab_name_PIPERACILLIN/TAZO	1.33%
7	ab_name_CEFUROXIME	2.49%	24	ab_name_RIFAMPIN	1.24%
8	ab_name_CHLORAMPHENICOL	2.39%	25	ab_name_TETRACYCLINE	1.16%
9	ab_name_CIPROFLOXACIN	2.03%	26	ab_name_TOBRAMYCIN	1.11%
10	ab_name_CLINDAMYCIN	1.97%	27	ab_name_TRIMETHOPRIM/SULFA	1.10%
11	ab_name_DAPTOMYCIN	1.94%	28	ab_name_VANCOMYCIN	1.04%
12	ab_name_ERYTHROMYCIN	1.62%	29	age_group_adult	1.02%
13	ab_name_GENTAMICIN	1.59%	30	age_group_elderly	0.98%
14	ab_name_IMIPENEM	1.58%	31	age_group_neonate	0.95%
15	ab_name_LEVOFLOXACIN	1.54%	32	age_group_oldest old	0.93%
16	ab_name_LINEZOLID	1.51%	33	category_ABG	0.89%

Table 15. Extremely Tree Classifier main importance rank.

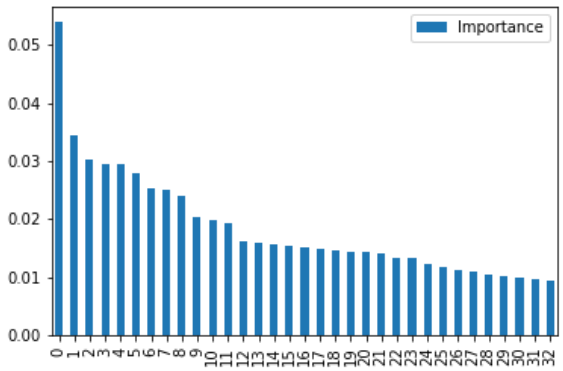


Figure 8. Extremely Tree Classifier main importance rank.

Reflection

The real data application is very difficult, compared with course examples. All transformation preprocessing was quite challenging. The use of SQL was a learning curve with high slope for me. I should have understood different relationships presented in the data, and take decisions about filters and selection in order to find a manageable problem that would allow me to make use of a large sample but could be run on my laptop and such data were consistent with the hypotheses raised.

Besides, I think this kind of studies must be done together with physicians' researchers, or at least an interdisciplinary team with people of different backgrounds in order to find better approaches to variables selections etc., and hypothesis validation and be another way to improve this kind of studies. For example, the variable selection could be made with more scientific rigor, and give to resulting models more trust (better metrics) to be applied in real life solutions.

For the future, could be valuable use a neural network approach, for example whit use of sequential neural network kind models applied to electrocardiograms or other kind of chartered time series, or extend the data sample selection to another acute illness.

From the importance results appear two important facts. First, could bet a high multicollinearity risk, it is the features can be representing same phenomenon, the vector solution can imply multiples solutions, so could be important restrict the data to antibiotic, to reduce dimensions, make the model simplest and avoid this risk. Second, there is an issue from feature database, there some raw data that are not standardized in their names, e.g. ABG, ABG'S, and ABG's are all the same category, but the program takes them as different features, so a semantic analysis could help improve the database and consequently the results of the models.

In literature, the SVM almost all the time overperform other kind of classifiers, but in this case, decision trees are more suitable to this data, but the examples normally are applied to Minst data set (which is numerical) and it doesn't have many categorical variables as I've chosen from Mimic's data base.