



DATA ANALYSIS – FINAL ASSIGNMENT

Prediction of kidney failure

JASSEM THOMAS / HENRI DESQUESSES

thomas.jassem@alumnos.upm.es

henri.desquesSES@alumnos.upm.es

2017/2018

Table des matières

SCOPE	3
BUSINESS GOAL	3
WORK PLAN	3
Milestones	3
GANT.....	4
RISK PLAN	4
DATA UNDERSTANDING	4
Explanation of each feature in detail:	5
EXPLORING THE DATA	6
DATA PREPARATION	7
ISSUE.....	9
MODELING.....	9
VARIABLE REMOVING.....	9
MODELING TECHNIQUE	10
TEST DESIGN	10
KNIME GRAPH	10
EVALUATION.....	10
FEATURES IMPORTANCE	10
ROC CURVE	12
ACCURACY	12
BUSINESS GOAL OBJECTIVES	12

Table of illustrations

Figure 1 - Gant of the project	4
Figure 2 - Correlation.....	6
Figure 3 - Proportion of kidney failure	6
Figure 4 - Knime graph of the model.....	10
Figure 5 - Attribute statistics - Random Forest Learner	11
Figure 6 - Importance of the features (table).....	11
Figure 7 - Importance of each features (Graph).....	11
Figure 8 - ROC CURVE	12
Figure 9 - Results	12

SCOPE

Kidney failure is the last stage of chronic kidney disease. When your kidneys fail, it means they have stopped working well enough for you to survive without dialysis or a kidney transplant. In that sense it is important when you create a new drug to be sure that it cannot be taken when there is a metric that is too high or too slow or when another drug is incompatible.

There are two main goals in trying to apply Big Data / Data Science tools to identify this type of patients. The first goal is to help the physicians of the nephrology department to decide if a patient will have a kidney failure or not. The second is to help to classify which features scientists must in priority concentrate on.

BUSINESS GOAL

Business Goal	Description	Indicator of success
BG1	Decrease the number of exams to detect if a patient will have a kidney failure or not.	We decrease the number of exams by 20%.
BG2	Detect patient who will have a kidney failure	We detect 10% more than before.

Data Mining Goals	Description	Indicator	Maps to BG
DM1	Detect most important exams/drugs.	Reduce the number of features by 20%.	BG1
DM2	Create a boolean to know if someone will have kidney failure or not.	It works at 60% on test datasets.	BG2

WORK PLAN

Milestones

Phase	Due Date	Responsible	Risks
Business Understanding	10/12/2017	Thomas JASSEM	Some advance on the subject has been done.
Data Understanding	17/12/2017	Thomas JASSEM	Data problems
Data Preparation	31/12/2017	Henri DESQUESES	Data problems, technology problems
Modelling	7/01/2018	Thomas JASSEM	Inability to build adequate model
Evaluation	10/01/2018	Thomas JASSEM	Inability to implement results
Report	14/01/2018	Henri DESQUESES	Lack of time

GANT

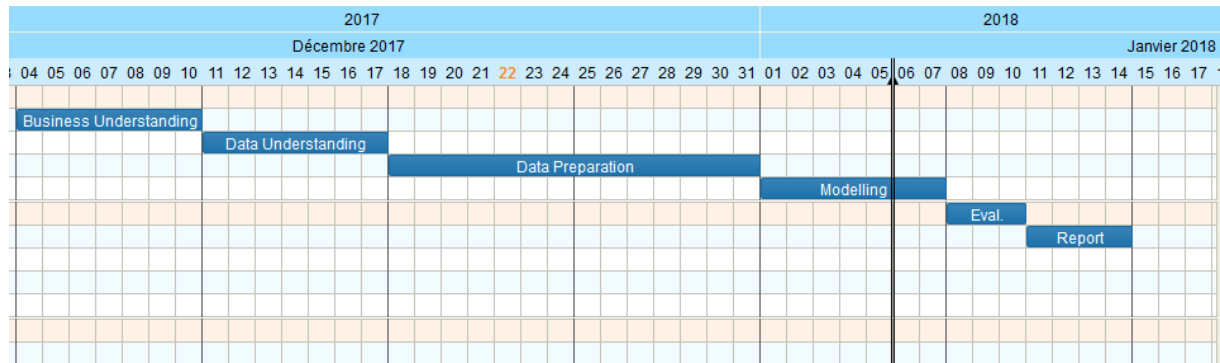


Figure 1 - Gant of the project

RISK PLAN

We should consider that the data comes from an hospital where they do not necessarily have a great system of information.

Risk	Probability	Plan
Blanc values.	100%	If the blank is for a continuous variable, we replace the null with the mean of this one.
Different names for a same thing.	75%	Try to score if two drugs are the same with their names.
Error in units	50%	Plot the different values of the exams to set

DATA UNDERSTANDING

Name	Number of attributes	Number of entries
kidney_fail_dataset.csv	20	957
drugs.csv	10	957

kidney_fail_dataset:

Name of the attribute	Values expected	Description	Type
patient_id	Numbers	Unique Identifier of an encounter	Nominal
height	Numbers	Height in inches	Numeric
weight	Numbers	Weight in pounds	Numeric
kidney_absortion_test	■	■	■
urea	Numbers	Urea level	Numeric
monocytes	Numbers	Monocytes level	Numeric
granulocytes	Numbers	Granulocytes level	Numeric
kidney_enzyme_test	■	■	■

eosinophils	Numbers	Number of eosinophils	Numeric
basophils	Numbers	Number of basophils	Numeric
kidney_suffering_test	■	■	■
platelets	Numbers	Platelets levels	Numeric
trgld	Numbers	?	Numeric
tflr	Numbers	?	Numeric
mean_platelet_volume	Numbers	Mean value of platelet volume	Numeric
leukocytes	Numbers	Leukocytes value	Numeric
glucose	Numbers	Glucose level	Numeric
Kidney failure	Boolean	The patient suffered a kidney failure.	Flag

drugs.csv:

patient_id	Numbers	Number of basophils	Numeric
drugX	Name	Name of the drug number X that the patient took.	Nominal

Explanation of each feature in detail:

Urea level: Urea is a waste product formed from the breakdown of proteins. It is eliminated from the body almost exclusively by the kidneys in urine.

Monocytes: Monocytes are a type of white blood cell. There are some articles that link Monocytes level and kidney diseases: <https://www.ncbi.nlm.nih.gov/pubmed/20649681>

Granulocytes: Granulocytes are a category of white blood cells. Some articles talk about the granulocyte colony-stimulating factor as a link with some kidney diseases.

Eosinophils: Eosinophils are a variety of white blood cells, we can find articles about the fact to have a rate high of eosinophils and kidney diseases: <https://www.ncbi.nlm.nih.gov/pubmed/21239387>

Basophils: Basophils are a type of white blood cells. Some article exists too: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3227736/>

Platelets: Platelets are a component of blood whose function (along with the coagulation factors) is to stop bleeding by clumping and clotting blood vessel injuries. They are linked with kidney diseases: <https://www.ncbi.nlm.nih.gov/pubmed/15497100>

Leukocytes: Most known as white blood cells.

So, we expect that the Leukocyte feature is correlated with Basophils, Eosinophils, Monocytes and Granulocytes.

EXPLORING THE DATA

Row ID	D height	D weight	D urea	D monocy...	D granulo...	D eosinop...	D basophils	D glucose	D platelets	D mean_...	D leukocy...	D trglid	D tfir	D nbDrugs
height	1	0.21	0.007	0.037	0.012	0.028	-0.017	0.045	-0.048	-0.039	0.048	0.045	0.002	-0.125
weight	0.21	1	-0.087	-0.004	0.066	-0.024	-0.035	0.121	-0.057	-0.017	0.085	0.121	0.002	0.023
urea	0.007	-0.087	1	0.004	0.107	0.066	-0.047	0.024	-0.089	0.085	0.01	0.024	0.163	0.181
monocytes	0.037	-0.004	0.004	1	-0.297	0.098	0.081	-0.076	-0.065	0.032	-0.16	-0.076	0.008	0.045
granulocytes	0.012	0.066	0.107	-0.297	1	-0.249	-0.21	0.088	0.078	-0.03	0.272	0.088	0.047	0.063
eosinophils	0.028	-0.024	0.066	0.098	-0.249	1	0.231	-0.046	-0.004	0.07	-0.005	-0.046	0.012	0.031
basophils	-0.017	-0.035	-0.047	0.081	-0.21	0.231	1	-0.062	0.062	0.069	-0.197	-0.062	0.061	-0.089
glucose	0.045	0.121	0.024	-0.076	0.088	-0.046	-0.062	1	-0.04	0.051	0.095	1	-0.04	0.122
platelets	-0.048	-0.057	-0.089	-0.065	0.078	-0.004	0.062	-0.04	1	-0.373	0.297	-0.04	0.012	0.05
mean_platele...	-0.039	-0.017	0.085	0.032	-0.03	0.07	0.069	0.051	-0.373	1	0.011	0.051	-0.067	0.004
leukocytes	0.048	0.085	0.01	-0.16	0.272	-0.005	-0.197	0.095	0.297	0.011	1	0.095	0.002	0.112
trglid	0.045	0.121	0.024	-0.076	0.088	-0.046	-0.062	1	-0.04	0.051	0.095	1	-0.04	0.122
tfir	0.002	0.002	0.163	0.008	0.047	0.012	0.061	-0.04	0.012	-0.067	0.002	-0.04	1	0.064
nbDrugs	-0.125	0.023	0.181	0.045	0.063	0.031	-0.089	0.122	0.05	0.004	0.112	0.122	0.064	1

Figure 2 - Correlation

In the figure above, we can see the correlation of the different features. We can deduce that, as expected, leukocytes are quite correlated to monocytes, granulocytes ...

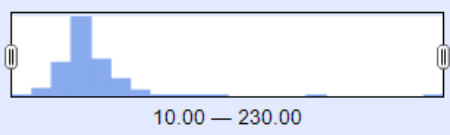
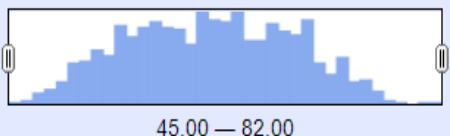
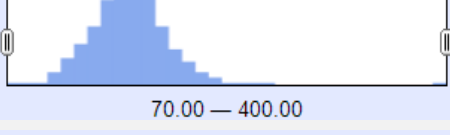
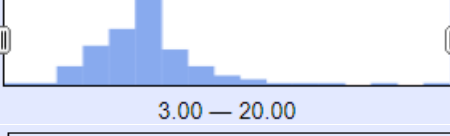
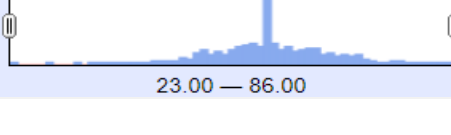
Granulocytes correspond to eosinophils and basophils and neutrophils so as expected they are correlated

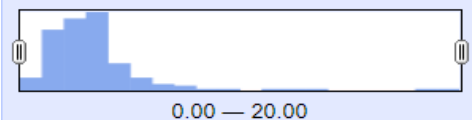
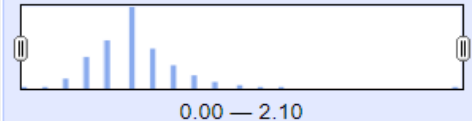
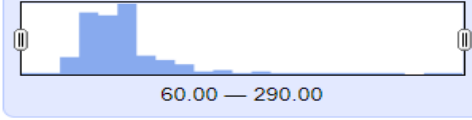
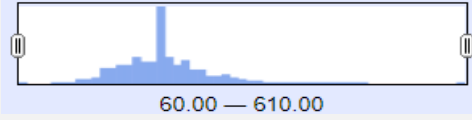
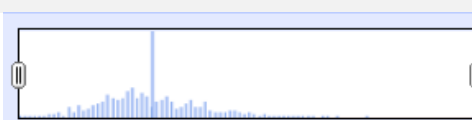
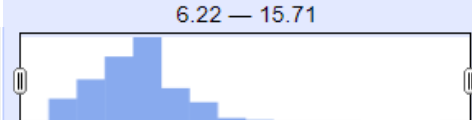
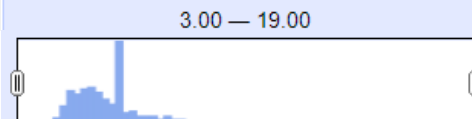
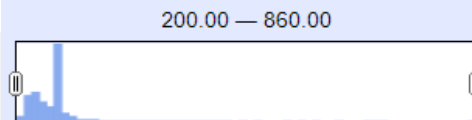
Row ID	count
0	497
1	460

Figure 3 - Proportion of kidney failure

We can see on the figure 2 above that the dataset is balanced with 48% of kidney failure.

Distribution of the different features:

FEATURE	GRAPH	COMMENT
UREA		Two values that seems to be aberrant. Because of that the feature seems unbalanced.
HEIGHT		Seems quite balanced.
WEIGHT		Seems quite unbalanced, there is one person who has a weight of 400 pounds and the next biggest one is 270 pounds.
MONOCYTES		Monocytes are quite balanced because variables are almost continuous until the maximum.
GRANULOCYTES		Granulocytes are balanced too.

EOSINOPHILS		Unbalanced, there are two values above 14 that makes it like this.
BASOPHILS		Unbalanced because of one value which is really high compared to the others.
GLUCOSE		Glucose graph is unbalanced but there is a continuity until the maximum.
PLATELETS		Platelets graph is a little bit unbalanced because of a value quite high: 610 vs 490 for the second max.
MEAN PLATELET VOLUME		Again, a value is high and makes the graph unbalances.
LEUKOCYTES		Same analysis as platelets.
TGRID		Here the value is continuous until the maximum.
TFLR		Again here, a value is really high but the graph will be unbalanced even without this value that we will change.

DATA PREPARATION

We chose to replace the maximums that seemed abberant by the second maximum to avoid distorting the data. Then, for numeric values that were not present we replaced them with the average. We used here OpenRefine to find the two maximums.

FEATURE	MAX	NEW MAX
UREA	227.7	160.6
WEIGHT	396.3	265.8
EOSINOPHILS	19.055	13.184
BASOPHILS	2.08	1.248
PLATELETS	600.08	486.72
MEAN PLATELET VOLUME	15.708	13.464
LEUKOCYTES	18.824	14.456
TFLR	6135.38	4939.12

Now that aberrant values have been changed we can replace blanks by the mean. I used spark to calculate the average:

```
import org.apache.spark.sql.Row

val inputDf =
spark.read.format("csv").option("delimiter",",").option("header","true").load("C:\\
Users\\tjass\\Documents\\Data_Analysis\\Projet\\merged_without_max.csv")

val avg_height = inputDf.select(avg($"height")).first.getDouble(0)
val avg_weight = inputDf.select(avg($"weight")).first.getDouble(0)
val avg_urea = inputDf.select(avg($"urea")).first.getDouble(0)
val avg_monocytes = inputDf.select(avg($"monocytes")).first.getDouble(0)
val avg_granulocytes = inputDf.select(avg($"granulocytes")).first.getDouble(0)
val avg_eosinophils = inputDf.select(avg($"eosinophils")).first.getDouble(0)
val avg_basophils = inputDf.select(avg($"basophils")).first.getDouble(0)
val avg_glucose = inputDf.select(avg($"glucose")).first.getDouble(0)
val avg_platelets = inputDf.select(avg($"platelets")).first.getDouble(0)
val avg_mean_platelet_volume =
inputDf.select(avg($"mean_platelet_volume")).first.getDouble(0)
val avg_leukocytes = inputDf.select(avg($"leukocytes")).first.getDouble(0)
val avg_trgld = inputDf.select(avg($"trgld")).first.getDouble(0)
val avg_tflr = inputDf.select(avg($"tflr")).first.getDouble(0)

val df = inputDf.na.fill(Map("urea"->avg_urea, "monocytes"->avg_monocytes,
"granulocytes"->avg_granulocytes, "eosinophils"->avg_eosinophils, "basophils"-
>avg_basophils, "glucose"->avg_glucose, "platelets"->avg_platelets,
"mean_platelet_volume"-> avg_mean_platelet_volume, "leukocytes"->avg_leukocytes,
"trgld"->avg_trgld, "tflr"->avg_tflr))

df.coalesce(1).write.format("com.databricks.spark.csv").option("header","true").sav
e("C:\\Users\\tjass\\Documents\\Data_Analysis\\Projet\\sample.csv")
```

With this code scala we used Spark to compute the average and replace blanks by this average in the dataset with the fill method.

The next step was to replace with OpenRefine the blanks in drug1, drug2, ... by “na” to process everything easier in Python.

Now the data is clean we merged the two “datasets drug.csv” and “kidney_fail_dataset.csv”:

```
a =
pa.read_csv("C:/Users/Max/Documents/UPM/Data_Analysis/Projet/src/Data/drugs.csv")
b =
pa.read_csv("C:/Users/Max/Documents/UPM/Data_Analysis/Projet/src/Data/kidney_fail_d
ataset.csv")
b = b.dropna(axis=1)
merged = a.merge(b, on='patient_id')
merged.to_csv("C:/Users/Max/Documents/UPM/Data_Analysis/Projet/src/Data/merged_data
set.csv", index=False)
```


We chose to use Pandas DataFrames in Python to process the data, so we put the merged csv into a panda dataframe:

```
df =  
pa.read_csv('C:/Users/Max/Documents/UPM/Data_Analysis/Projet/src/data/sparkFilled.csv');
```

We wanted to have the list of drugs that exists in the dataset so we decided to put the names of drugs of each column into a list. Then we used the function “get_close_matches” to group all the drug names that are similar with a probability of 40% and replace by the name of the group into the dataframe.

We wanted to help the algorithm to converge so we decided to create **new features**.

The first one is the number of drugs, so for each patient we added the number of drugs that it took.

Then we wanted to add the IMC or BMC (Body Mass Index) in English which is a variable that can be used to estimate a person's body size. This index is calculated as a function of height and body mass. We decided to put the IMC in categories:

IMC	INTERPRETATION
+40	morbid obesity
35 to 40	severe obesity
30 to 35	moderate obesity
25 to 30	overweight
18.5 to 25	normal corpulence
16.5 to 18.5	thinness
-16.5	famine

The last step is the normalization of the data, then we exported everything as csv named dataset2.0 that we will use in Knime.

ISSUE

We wanted to put each drug as a column to know exactly which drug is important or not but the problem is that we have only 900 records for a number of distinct drugs of approximately 1600. So, the model will not be good because the number of records is not big enough.

MODELING

VARIABLE REMOVING

As the IMC is really correlated to weight and height we decided to remove them.

Leukocytes which are white cells feature is correlated with other subcategories of white cells so we deleted it and then the granulocytes too because there are correlated with eosinophils and basophils.

Then we tried to put in knime all the features but the problem is that there are too much different drugs against the number of patient so at the end drugs are almost unique per patient. In fact, the

algorithm is warning us that the model cannot take in account columns drug1, drug2, drug3, ..., drug9.

MODELING TECHNIQUE

We have to determine a Boolean value and we have continuous and categorial features. The random forest seems for us to be the more adapted to our model because the first idea would be to use the logistic regression but we have categorial variables that would not be taken into account.

We tested different parameters for the depth of the tree, we arrived to the conclusion that 10 is good because it gives the best accuracy.

TEST DESIGN

We decided to use cross-validation to validate our model, we use 10 folds built randomly.

KNIME GRAPH

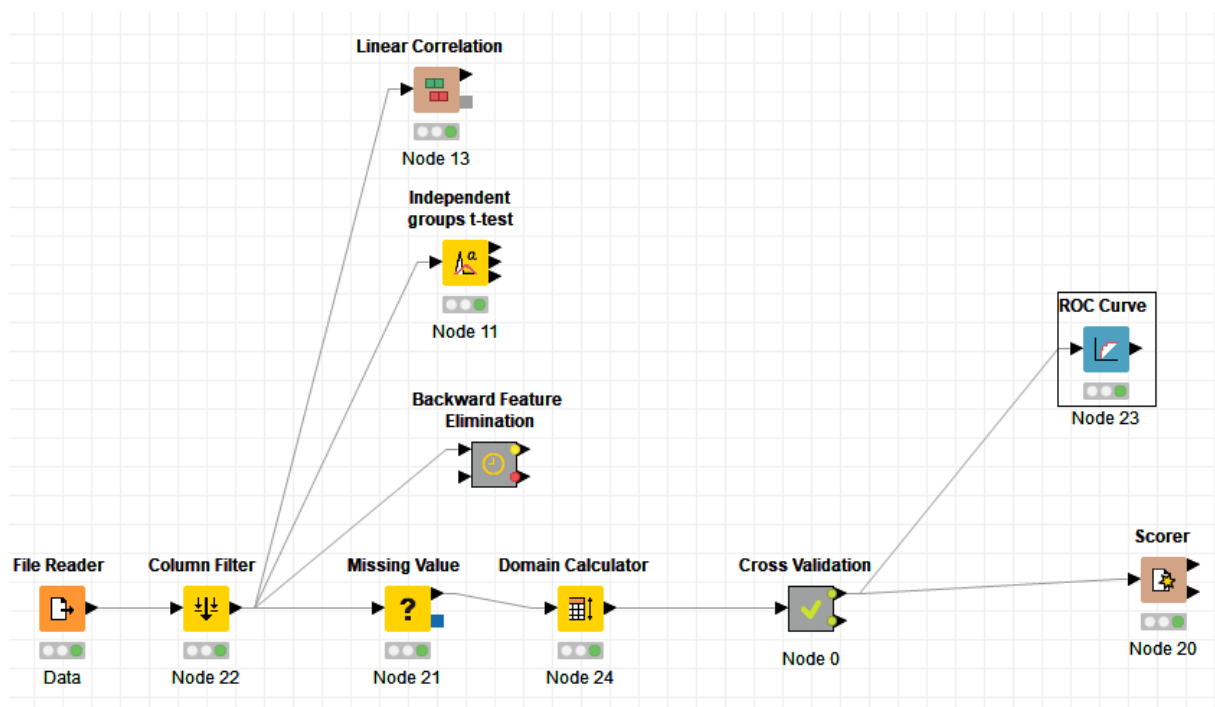


Figure 4 - Knime graph of the model

EVALUATION

FEATURES IMPORTANCE

As a measure for variable importance we took a look into the “attribute statistics” part of the random forest learner. Here we can see how often a variable was used for building a decision tree at the first second or third level. To rank the features, we divided the splits with its candidate and sum the three.

Attribute Statistics - 0:0:3 - Random Forest Learner

File Hilite Navigation View

Table "Tree Ensemble Column Statistic" - Rows: 11 Spec - Columns: 6 Properties Flow Variables						
Row ID	#splits ...	#splits ...	#splits ...	#candi...	#candi...	#candi...
urea	23	23	42	31	52	103
monocytes	5	18	22	31	62	100
eosinophils	3	11	32	27	57	101
basophils	3	8	22	20	55	109
glucose	6	20	25	30	55	91
platelets	7	12	31	25	55	105
mean_platele...	6	6	30	30	50	100
trgld	3	20	25	25	57	109
tfir	15	24	48	28	47	96
nbDrugs	23	33	43	24	55	127
IMC	6	16	25	29	55	105

Figure 5 - Attribute statistics - Random Forest Learner

Features	Importances	Relative Importance
nbDrugs	1,9	18,07%
urea	1,592	15,14%
tfir	1,546	14,70%
glucose	0,838	7,97%
platelets	0,793	7,54%
IMC	0,736	7,00%
trgld	0,7	6,66%
monocytes	0,672	6,39%
eosinophils	0,621	5,91%
mean_platelet_volume	0,62	5,90%
basophils	0,497	4,73%
Total général	10,515	100,00%

Figure 6 - Importance of the features (table)

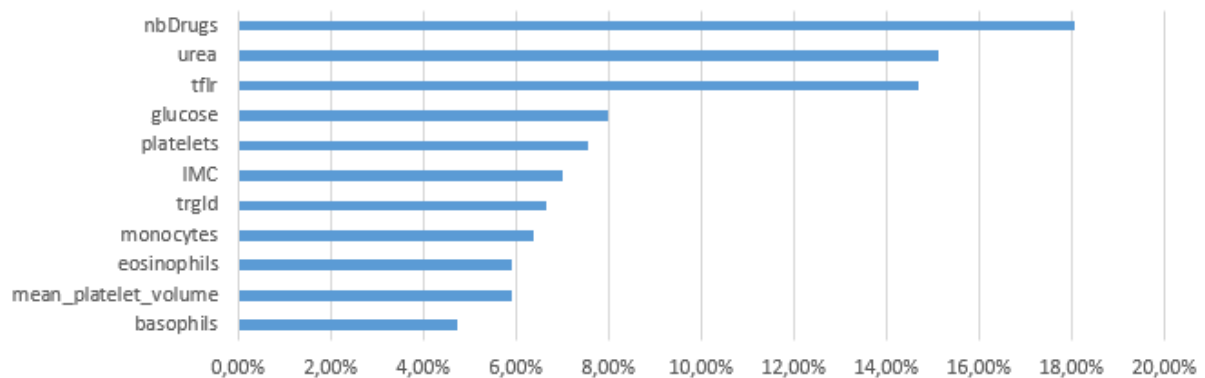


Figure 7 - Importance of each features (Graph)

ROC CURVE

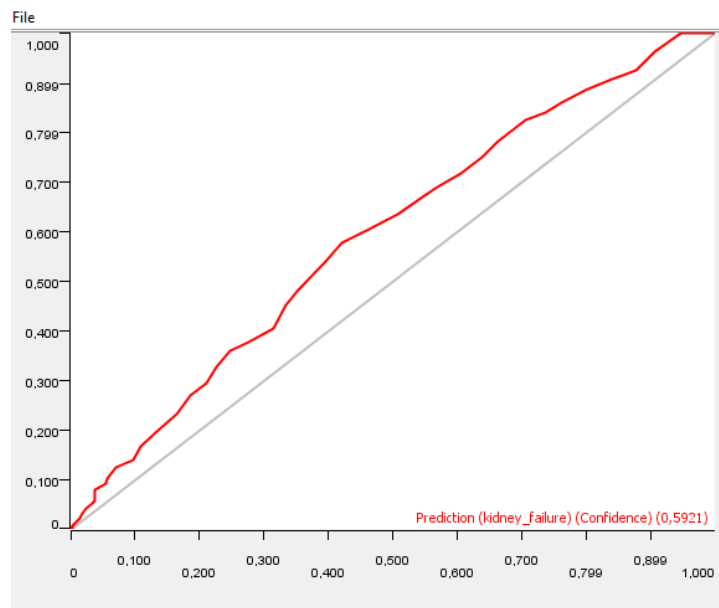


Figure 8 - ROC CURVE

The roc curve is a measure of the performance of a binary classifier, It gives the rate of true positives (fraction of positives that are actually detected) based on the rate of false positives (fraction of negatives that are incorrectly detected).

Here we can see than the curve (in red) the further the curve deviates from the random classifier line and approaches the elbow of the ideal classifier (which goes from (0, 0) to (0, 1) to (1, 1)). So we can conclude that our model is quite good.

ACCURACY

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	D Recall	D Precision	D Sensitivity	D Specificity	D F-meas...	D Accuracy	D Cohen...
0	291	176	284	206	0.586	0.623	0.586	0.617	0.604	?	?
1	284	206	291	176	0.617	0.58	0.617	0.586	0.598	?	?
Overall	?	?	?	?	?	?	?	?	?	0.601	0.202

Figure 9 - Results

We can see that we have an accuracy of 60,1%. It is not very high but considering that we only have 900 entries and that we cannot use the name of the drugs taken by the patient, it remains suitable.

We measured a **Shannon entropy** of 0.21 so the dataset is a predictable but not highly predictable.

BUSINESS GOAL OBJECTIVES

The Business Goal 1 was decrease the number of exams by 20%, we have deleted leukocytes and granulocytes tests and we have deleted all the drugs taken and replaced them by the number of drugs so we have deleted $2+9-1=10$ features over a number total of initial features of 23 features so 43% of the features. So, the Business Goal 1 is **validated**.

For the Business Goal 2, if we consider that before we had only 50% of chance (randomly) to find if a patient will have it or not we can say that yes with 62% of patient detected the business goal 2 is **validated**.