Project report

Rithy SOCHET
Marc KASPAR

Contents

1 Introduction

As computer science students specialised in data science and with one of the member of the group who did one year of study in medicine, we wanted to ally both fields of studies. So we came up with a subject of medicine where machine learning could be used : electrocardiograms, or ECG for short.

An ECG is a graph of voltage over time, it measures the electric change due to the activity of the heart. To get an ECG, we place multiple electrodes at different angle around the heart on the skin of the patient. We then get what is called a 12-lead ECG (12 angles) which is multiple signals. The ECG is then given to the doctor who will have to read it and look for potential anomalies.

Before going over the dataset we used, we will briefly give some knowledge on what to look for in an ECG. Here is a figure showing 2 successive QRS complex.

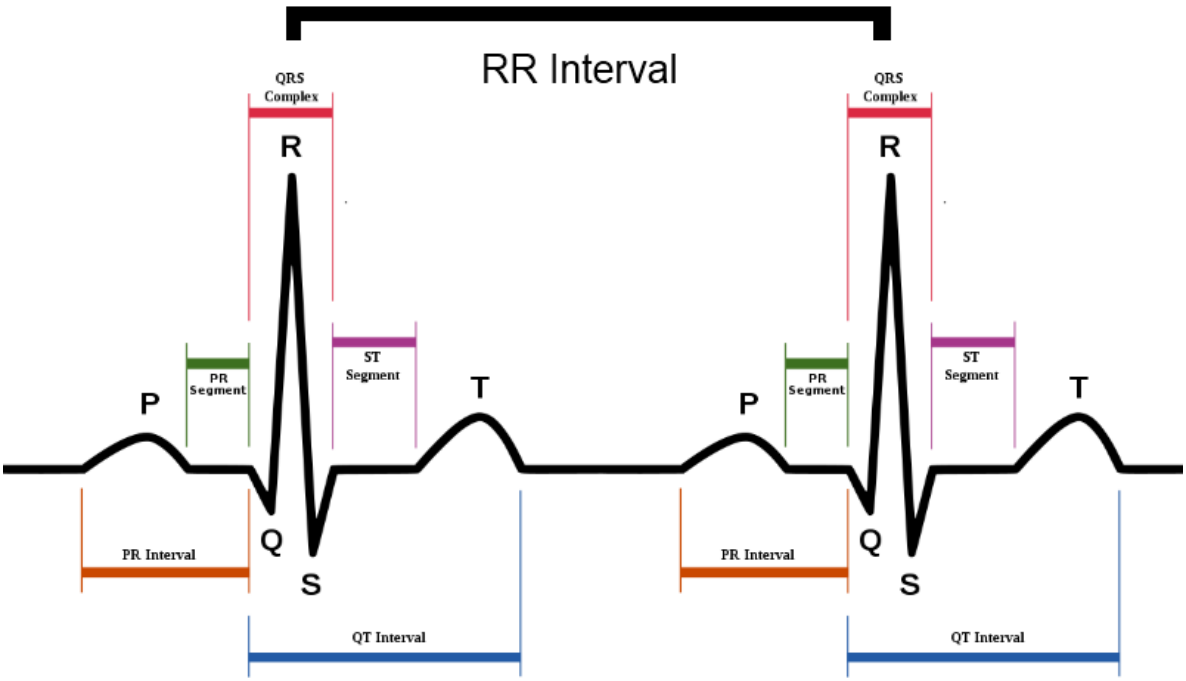

Figure 1: 2 successive QRS complex of a normal patient

In this figure you can observe different "wave" of polarity, for example the wave P, the wave T, and the QRS complex composed of the waves Q, R and S.
Each of these waves corresponds to the polarisation or depolarisation of a certain part of the heart.
You can also observe the interval of time between two waves. Finally we can observe the RR interval which is the amount of time between two peak of two successive QRS complex.
To find an anomaly, the doctor will have to read multiple signals and find potential irregularity.
For example, an interval too long between two waves, a wave that has an abnormal duration, a signal with an abnormal form, a peak that is too high or too low and so on...

2 Tasks

The objective of this project is to find out if machine learning methods can be used to detect an anomaly in the ECG and identify the problem. This is a multiclass classification problem.
Getting the ECG signals and reading it to find a potential anomaly can take time and the doctor may make a mistake. So the goal of the project is to find out if we could automatise this process and if it is possible to use machine learning in hospitals with this aim.
If it is possible, then which model should be used ? How efficient is it ? What kind of work should be done on the data for it to be used efficiently ? Can it identify a problem quickly enough ?

To answer these questions we trained multiple models that we learned about during our studies. We make use of pipelines to find the best hyperparameters and then we compare the end results.
We will also explore the data to visualise the data and find out if there are results that can be explained.

3 Dataset description

3.1 Why did we choose this dataset

The dataset we chose for this project is a dataset from Kaggle named ECG of Cardiac Ailments Dataset made by Alekhya Lanka (15). This dataset uses multiple dataset available publicly on Physionet.org, the features were obtained from signals using MODWPT technique.

While true ECG signals are not already given and processed into usable data, for this project we decided to use this dataset as we lack the knowledge to transform raw signals into usable data. Moreover we wished to be able to understand and explain the features that will be used in the models and their possible influence on the result.
Indeed, if we only considered the voltage measured at each point in time it would be difficult to find meaning of each feature. This would only complicate the interpretation of the results obtained, whether they are good or not.
Was the transformation of the signal done correctly ? Is the model not good for this type of data ? Was there some kind of data leakage due to a manipulation during the transformation ?
Thus, we chose this dataset instead of another one.

3.2 Exploration of the dataset

In this part, we will show multiple figures, they can all be obtained by running the notebook available in annex of this report. We used the libraries matplotlib and seaborn for data visualisation and we used pandas to use dataframes and display the data.

The dataset we used contains 1200 record each of one of the following classes:

- "NSR" - Normal Sinus Rhythm
- "ARR" - Arrhythmia
- "AFF" - Atrial Fibrillation
- "CHF" - Congestive heart failure

The first class "NSR" correspond to a patient with no cardiac disease, the 3 others correspond to a different kind of cardiac disease.

A quick scan of the dataset shows it contains 300 records of each class as shown in the following plot:

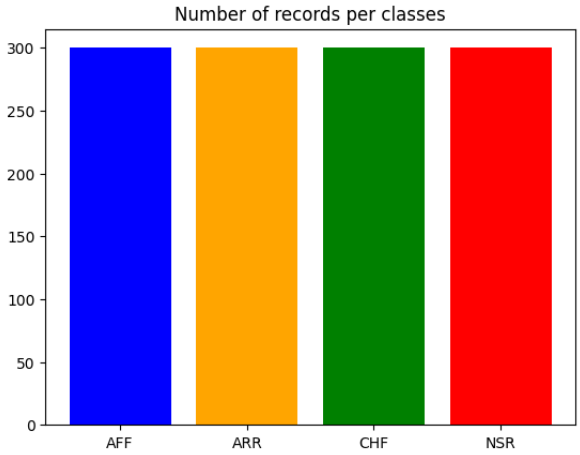

Figure 2: Distribution of each class of the dataset

This shows us that the dataset is balanced concerning the distribution of classes.
However, we still have to note, that the amount of records with people with a diseases is 3 times higher than for those with no disease. This may cause some problems when we calculate the median or the mean for each features, since these values might be skewed towards an abnormal value.
This could be the cause of error when we try to classify a patient with no cardiac disease if we have to use some of those values for imputation.

The dataset contains 55 features:

- The first feature is the number of the record. This feature will be dropped when we will train our models as it doesn't bring any information.
- The second one is the heartbeat per minute. We will use this feature for visualisation.
- The 3rd to the 16th correspond to the duration of an interval. We will use the "RRmean" feature for visualisation. It corresponds to the mean time between two successive R peak.
- The next 25 features correspond to an euclidian distance or an angle between two points of the signals
- The next 7 features correspond to values of time between 2 heartbeats.
- The 47th feature ("QRSarea") correspond to the area under the QRS complex in the graph. We will use this feature for visualisation.
- The 48th feature ("QRSperi") correspond to the sum of the distance between QR, RS and QS points. We will use this feature for visualisation.
- The next 4 features correspond to slope between different points of the ECG.
- The last 2 features correspond to a number of successive RR intervals respecting some conditions.

All these features are float numbers and we can note that there are missing values.
In particular half of the dataset is lacking a value for certain angles or duration between two waves or for the slope. This could be explained by the fact that these records originally come from different datasets and that getting those information was impossible.

We now want to see how these different features interact with each other. To do that we will calculate the correlation matrix. We use the pandas function corr(), which uses the Pearson correlation coefficient:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Where X and Y are variables (here our features) and σ_X and σ_Y are their standard deviation. The correlation matrix's coefficient (i, j) is the correlation coefficient of the feature i and j. The value will be a float between -1 and 1.
A value near -1 means that the two features are highly negatively correlated, i.e. if X increases then Y should decrease.
A value around 1 means that they are highly positively correlated, i.e. if X increases then Y should increase.
A value near 0 means that the features are not correlated, i.e the value of X does not influence the value of Y. Note that this matrix is symmetric.

An efficient way to quickly to visualise these values is to use a heatmap.

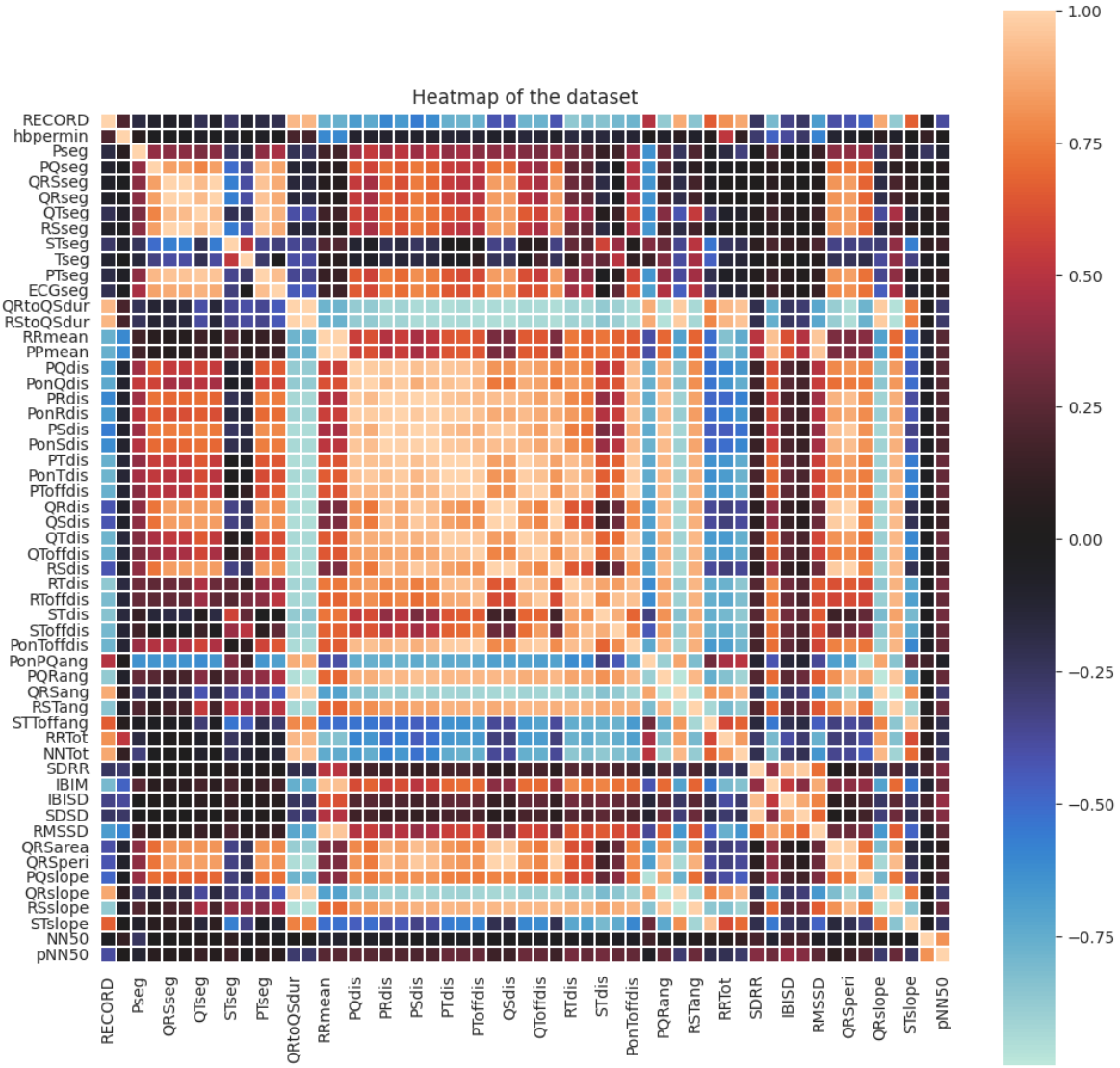

Figure 3: Heatmap of the dataset

With this figure, you can see more easily the correlation between the features by using the color with the legends on the right. For example:

- The heartbeat per minute does not have an influence on the duration of the different intervals but it is negatively correlated with the RRmean feature. This is consistent with the fact that if the heartbeat per minute increases then the intervals between two R should decrease.
- All the features concerning the distance between two points are highly positively correlated with the duration of the different intervals.
- The distance features are highly positively correlated with the features on the angle and slopes.

Let's now visualise the distribution of some features with respect to their classes.
We will use box plots. In a box plot, a box will represent the different values taken, the bottom of the box represents the first quartile, the line inside the box represents the median and the top of the box represents the 3rd quartile. The line under the box represent the minimum value while the line over the box represent the maximum value.

We decide to visualise the following features "hbpermin", "QRSarea","QRSperi","RRmean" as they have important medical impact:

- "hbpermin" : The heartbeat per min is an easy and efficient way to see if a patient might be suffering from a cardiac problem.
- "QRSarea": The area under the QRS complex is also an efficient way to see if a patient suffers from some cardiac anomaly. Indeed a value of this feature that is different from the norm means that some wave of the QRS complex has a different value from what it should be.
- "QRSperi" : This feature is highly correlated with more than half of the features of the dataset. This is consistent with the fact that this feature is the sum of distances between different points of the ECG. We could say that this feature summarises the different features about distances.
- "RRmean": This feature is also highly correlated with more than half of the features of the dataset. It is also negatively correlated with the heartbeat per minute. So we will see the influence this feature has with "hbpermin".

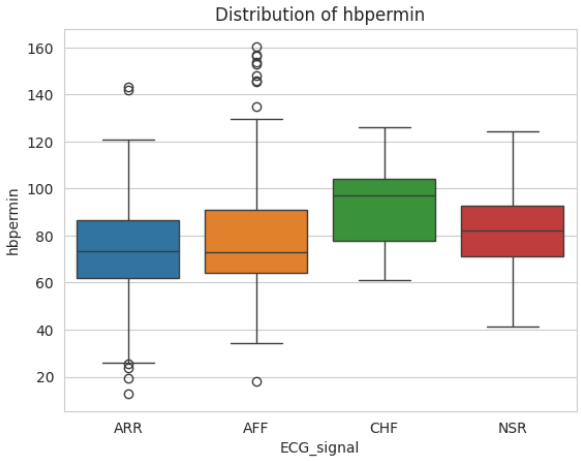

Figure 4: Box plot of the feature hbpermin

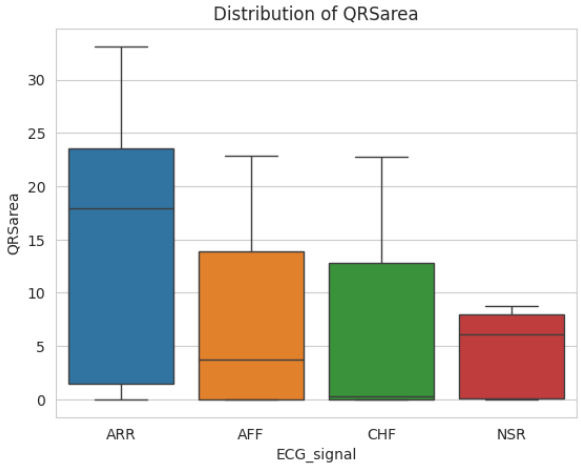

Figure 5: Box plot of the feature QRSarea

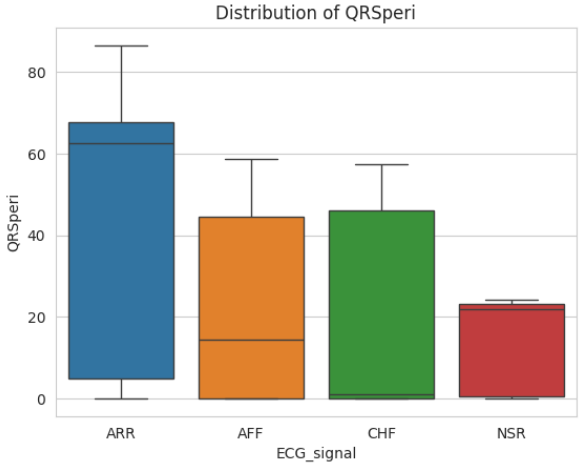

Figure 6: Box plot of the feature QRSperi

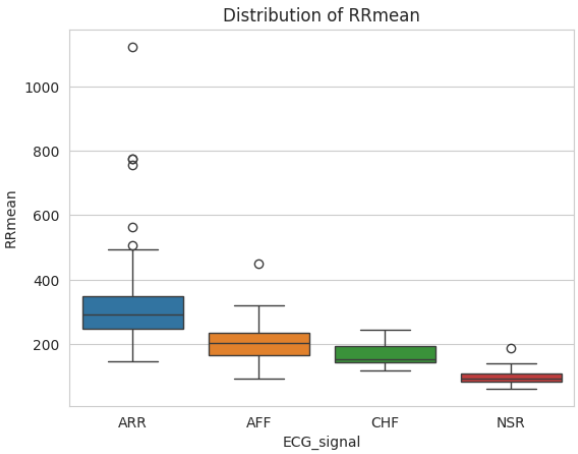

Figure 7: Box plot of the feature RRmean

Let's analyse these different plots.

Figure 4 does not allow us to classify efficiently a record from the class ARR, AFF or NSR as records of these three classes have similar values for this feature. Indeed the boxes show that records of these three classes have values between 60 and 90, with around 80 being the median value for patient not suffering from disease (class NSR).
However at least half of the records of the class CHF have a value near 100 and 25% of the records of this class have value at least above 100. So this feature will be useful to identify a record of the class CHF.

Figure 5 gives us a lot of information about the feature's distribution:

- The normal values for a patient should be between 0 and 8 as shown by the NSR box with median value around 6.
- For records of the class ARR, the variance is high (it has the biggest box) and the median value is around 18. 25% of the records of this class has values above 23 which the other 3 classes are far from reaching.
- For records of the class CHF, we can note that the median value is almost 0 which is a really strange value for this feature from a medical outlook.
- For records of the class AFF, the values are similar to that of the class CHF but the median value is around 4 which is still lower than the value of NSR i.e. the normal value.

We can conclude from this box plot that the feature "QRSarea" can efficiently help us classify a record. Indeed any value above 8 cannot be from the class NSR. If the value is above 23, we can already classify it as ARR. If the value is near 0 there is a high chance that it is from the class CHF.

Figure 6 gives a really similar box plot as the previous figure. Unfortunately it does not gives us more information since we reach the same conclusion as the previous plot. This is consistent with the fact that the features "QRSarea" and "QRSperi" are highly positively correlated (according to the heatmap there is a correlation of 1 between these two features).

Figure 7 is the most conclusive. Indeed, contrary to the three other plots, almost all four boxes do not intersect.
It means that records of different classes have differents values for this feature. 75% of the records of the class NSR have value less than 100, while all the records of the ARR have value greater than 100. Only the boxes of the classes AFF and CHF intersect.

Figure 8: Pairwise scatter plots

We visualised these four features individually, now let's study them together, pairwise:
Remark: We can see this figure as a matrix where each coefficient is a plot. This matrix is symmetric if we consider that we just need to reverse the axis, so we can only consider the diagonal and the plots under or over it.
The diagonal shows plots of the distribution of each feature for each class.

Let's start by analysing the plots of the diagonal. Like the box plots we previously saw, we can see how the values for each features are distributed depending on their class. We can get similar conclusion but unlike the box plots, we can get a better visualisation on how the values are distributed:

- For example, for the features "QRSarea" and "QRSperi", all the classes have two peaks with one of the peak happening at the same point in abscissa, though the height of the peak is different for each class.
- Like with box plots, we can easily see that the plots for "QRSarea" and "QRSperi" are really similar, while the classes NSR and ARR can easily be distinguished, the classes AFF and CHF

have a lot of intersecting values.

- Concerning the plot for hbpermin, like before, we cannot give too many conclusion beside the fact that the class CHF seem to behave differently.
- Finally, for "RRmean", we can easily distinguish the classes NSR and ARR though AFF and CHF do share some values with each other.

Another thing that we can observe with these plots is that "hbpermin" and "RRmean" seem to be following a gaussian distribution, however "QRSarea" and "QRSperi" do not as they have two peaks where the values are concentrated. This information will be useful when we will talk about some of our models.

Let's now analyse the pair plots (we will consider the ones under the diagonal).

- We can note that plots with "QRSarea" or "QRSperi" and another feature have the same shape (like the plot of "hbpermin" and "QRSarea" and the plot of "hbpermin" and "QRSperi"), this is consistent with the fact that both follows the same distribution.
- In the plot of "hbpermin" and "RRmean" we ca, easily distinguish three curves. One is of the class NSR, another one if of the class ARR, and finally the last one where both class CHF and AFF intersect.
- In the plot of "hbpermin" with "QRSarea", we can also distinguish the classes NSR and ARR, while the classes AFF and CHF seem to concentrate in the same area.
- In the plot of "QRSarea" and "QRSperi", we can see that the points seem to follow a line, each class seem to concentrate in some part of the line. NSR concentrate at the bottom left while ARR concentrate at the top right of this line. AFF and CHF concentrate at the same place between the two other class.

What can we conclude from these observations ?

From these observations, we can see that it should be easy to classify a record of class NSR or ARR. However the classes AFF and CHF seem to share a lot of similarities, so if the models make mistakes, it should be for these two classes. In fact, these two diseases are commonly encountered together and share risk factors.
Records of the class NSR (so the ECG of a patient not suffering from heart disease) have unique values that will allow us to easily distinguish if a patient has a disease or not (if we consider the classes normal ECG and abnormal ECG).
"QRSarea" and "QRSperi" are highly positively correlated and share almost the same distribution, we can then wonder if keeping both of these features is useful when getting new data.

These conclusions will guide us for the different tests we will do in the next parts.

4 Methodology

As a reminder, the objective of this project is to find out if it is possible to use machine learning techniques to detect cardiac problem in ECG, and if it is, how efficient are they ?

To answer these questions we will be using three different models we learned about during our studies: Logistic regression, Linear Discriminant Analysis (LDA) and Decision trees. We will also try to use the MultiLayer Perceptron neural network (MLP) of scikit-learn.

- Logistic regression is a linear model that uses a sigmoïd function to determine the class of a record in a two class settings. In a multiclass problem the scikit-learn implementation uses the multinomial logistic regression which is a generalisation of the logistic regression for multiclass problems (according to the documentation).

- LDA is a linear model that assumes that each class follows a Gaussian distribution and all have the same covariance matrix.
- A decision tree is a tree that splits at each node depending on a condition on a feature (in our case this condition is a threshold since all the features are floats). When we classify a record, we traverse the tree starting from the root until we reach a leaf where a label will be given.
- MLP is a modern feed forward neural network.

To store our data we will use Numpy, for the preprocessing of the data, the training of our models and the models themselves, we will use scikit-learn.

We will use pipelines and a cross validation grid search to select the best hyperparameters for each models. In the pipelines we will also see if using a standardisation is better or not and which imputer between mean and median gives the best performance.

We will evaluate our models using the accuracy score.

We will train our models in three different settings:

1. First setting: we consider only two classes, AFF and CHF. During our data analysis we saw that these two classes were often focused in the same area (at least for the features we visualised). The goal here is to have a first approach of the dataset concerning the training of our model.
2. Second setting: we consider two classes, normal ECG (class NSR in the dataset) and abnormal ECG (the three other classes). The goal here is to see how efficient our models will be to detect an anomaly.
3. Third setting: we will consider the four classes, contrary to the first two settings, here we have a multiclass problem. The goal is to try to answer the main objective.

In each setting we will separate our dataset into a training set representing 80% of the dataset and a test set which will get the remaining 20%. This separation is stratified and we make ten folds for the cross validation.

After the training we will see if reducing the number of feature with the Principal Components Analysis (PCA) still allow us to keep a good performance.

5 Results

We were expecting good results from our models, considering that doctors can already do the work quickly and efficiently, but we were surprised to see that the results we obtained were extremely good. Here are tables of results of the different settings:

Model	Accuracy
Logistic regression	0.958
LDA	0.958
Decision trees	0.95

Table 1: Results of setting 1

Model	Accuracy
Logistic regression	1.0
LDA	1.0
Decision trees	1.0

Table 2: Results of setting 2

Model	Accuracy
Logistic regression	0.9667
LDA	0.9667
Decision trees	0.95
MLP	0.9876

Table 3: Results of setting 3

As a first approach to the dataset we decided to only consider records of the class AFF and CHF as they seem to bear the most resemblance according to our data analysis. Table 1 shows that the three models gives excellent results as they all have an accuracy of 0.95 or above. We can note that logistic regression and LDA are slightly better than decision trees, however this might change if we change the hyperparameters of the decision tree, though we might risk overfitting.
From this result we can already start to see that machine learning models will be effective to classify ECGs.

Table 2 shows the results when we consider the classes normal ECG and abnormal ECG. Considering the different plots we saw earlier and the result of Table 1, the results of Table 2 are not really surprising. We can still wonder if the fact that we have two unbalanced classes (300 normal against 900 abnormal) had an effect on these results.

Finally Table 3 shows the results of the original problem with the four classes. Like in Table 1, we have a really high accuracy. For the first three models the results are similar to those of table 1. We also tried using the MLP neural network, the result shows that it has slightly better accuracy than the linear models.
Using confusion matrices we could observe that for every models, the errors are from record of the class AFF that are classified as CHF or vice versa. So the models never makes mistakes when classifying ECG of the class NSR or ARR.

After all these tests we were wondering if the dataset might be providing too many features, making the problem "too easy". Thus, we tried applying PCA on the dataset with logistic regression add MLP. We made use of a pipeline to decide how many features will allow us to keep the best accuracy for setting 3.

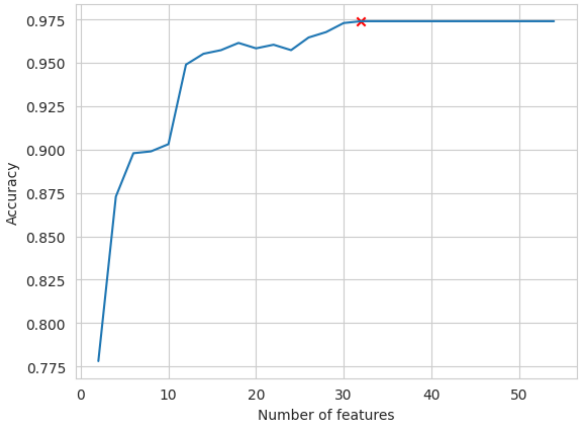

Figure 9: Number of features to keep maximum accuracy with logistic regression

Figure 9 shows us that with 32 features, logistic regression gets the best accuracy. More features do not increase it.

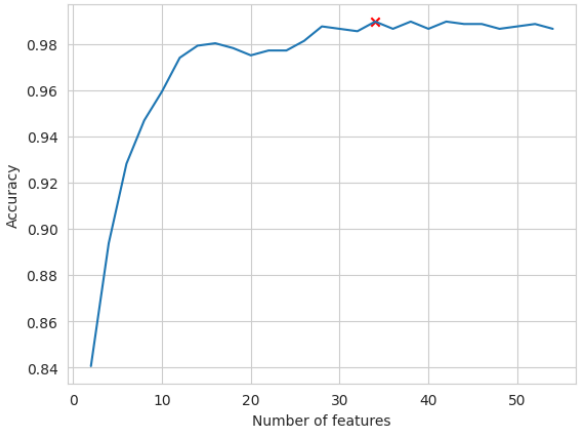

Figure 10: Number of features to get maximum accuracy with MLP neural network

Figure 10 shows us that with 34 features can let us get maximum accuracy and that more features may decrease the accuracy. We can remark that running the pipeline multiple times gives different results, this is due to the variability starting from 26 features.

Finally, we used the PCA to calculate the cumulative variance ratio, it shows us how many features allow us to keep a certain amount of information. Our results show that with 7 features we can keep 99% of the information of the dataset. We then tried to run the MLP with only 7 features. All the results can be found in the following table:

Models	Number of features	Accuracy
Logistic regression	32	0.983
MLP	34	0.992
MLP	7	0.958

Table 4: Accuracy results after PCA

We can remark that the accuracy of the logistic regression and MLP increased compared to the result of table 3. This could be explained by the fact that some features being noises in the dataset.
We can also note that with only 7 features, the MLP model still gives good results and with 34 it is almost perfect.
We can conclude that reducing the number of feature is useful for this dataset. Indeed some features can be deducted from others, for example "hbpermin" can be deducted from "RRmean". "QRSperi" is a sum of the different interval of time between waves, this information can be obtained from the different "segment" features. (This can be seen in the heatmap of figure) Another thing to recall is that half of this dataset is missing values for some features like features on "angles", "duration" and "slopes".

6 Conclusion

6.1 Our results and conclusion

The objective of this project was to find out if machine learning models could be used to detect anomalies in ECG and classify these anomalies and how efficiently this task could be done.

The results shown in the previous part tell us that machine learning techniques are really promising and efficient for this task. Linear models can already classify multiple cardiac diseases, and if we simplify the problem to only find out if a patient has a disease or not, then our results show a perfect accuracy. Of course neural network should also get results as good and even better than linear models (by a few percents). In the context of healthcare, even a few percents can mean the life or death of a patient, therefore training and using neural network is not a wasted effort.

However it should be noted that the dataset used for this project uses relevant information extracted from raw ECG signals.
While the process used to extract these information could be automatised, we can still wonder if the results from using raw signal would be as good as the ones shown in this report. Moreover, the dataset only considered fours classes, it is possible that there are diseases that are difficult to distinguish, lowering our models' performance.
We can also wonder if getting all these 54 features was really necessary as shown by the results of the PCA.

6.2 What did we get from the project

From this project we had our first approach in using machine learning models and techniques in a concrete project using real life data. While the results did not really allow us to manipulate the data or the models to try to improve performance (from the start the accuracy was above 0.9), we had a first approach in data visualisation.
Indeed we learned how to use different kinds of plots and figures like box plots and heatmap. To do that we used the library seaborn with which we also did pairwise plot to better visualise the relation between multiple features.
Finally, to write this report we also used Overleaf and LaTeX for the first time.

6.3 What can we improve for next time

We overestimated the scope of the project and had to change the objective of the project from trying to find out if machine learning techniques can be used for ECGs to data visualising our data to explain the results we obtained.
What could be interesting to do for next time is to work directly on the raw signals, we could then try to use the MODWPT technique used by the dataset's author to extract relevant information.
If we want to stay in the medical field, we could also have a project concerning biology and genomes using LLM.

Another thing that could be improve is the knowledge related to the use of Overleaf and LaTeX as there are problems in the position of figures and tables that takes more places than we would have liked.

7 Annex and References

You can find with this report a notebook called "Project_ML_ECG.ipynb" and the dataset in csv format "ECGCvdata.csv".

Kaggle dataset : ECG of Cardiac Ailments Dataset
Citations and acknowledgement of the dataset:

- Alekhya, L., and P. Rajesh Kumar, "A new approach to detect cardiovascular diseases using ECG scalograms and ML-based CNN algorithm." Mar 20, 2023. International Journal of Computational Vision and Robotics/Inderscience publishers.
  DOI: 10.1504/IJCVR.2022.10051429
  Link to article

- Alekhya, L., and P. Rajesh Kumar. "A Novel Application for Autonomous Detection of Cardiac Ailments using ECG
  Scalograms with Alex Net Convolution Neural Network." Design Engineering (2021): 13176-13189.
  Link to article

- Autonomous Detection of Cardiac Ailments using Long-short term Memory Model based on Electrocardiogram signals, L. Alekhya, P. Rajesh Kumar, A. Venkata Sriram
  DOI: 10.14704/nq.2022.20.7.NQ33431. Pages: 3509 - 3518.
  Link to article

- Autonomous Detection of Cardia Ailments diagnosed by Electrocardiogram using various Supervised Machine Learning AlgorithmsAutonomous Detection of Cardia Ailments diagnosed by Electrocardiogram using various Supervised Machine Learning Algorithms
  AMA, Agricultural Mechanization in Asia, Africa and Latin America (ISSN: 00845841) · Sep 18, 2021.
  Link to article

- L Alekhya, P Rajesh Kumar, "Maximal Overlap Discrete Wavelet Packet Transform Based Characteristic waves detection in Electrocardiogram of Cardiovascular Diseases", INTERNATIONAL JOURNAL OF SPECIAL EDUCATION, vol 36 (1), pp 51-61, 2021.

Resource used to understand how to use heatmaps: Link to article

Resource used to understand how to use box plots: Link to article