NCSR Demokritos and University of Piraeus

Machine Learning

MSc in Artificial Intelligence

Heart Murmur Detection using Machine Learning

Authors
Andreas Sideras, MTN2214
Tatiana Boura, MTN2210

Supervisor
Theodoros
Giannakopoulos

February 14, 2023

Contents

1 Introduction

Cardiac auscultation via stethoscope is the most common diagnostic screening tool that can help to identify patients with heart murmurs. However, this technology has limited diagnostic sensitivity and accuracy and requires skilled doctors to interpret its results. Lately, the traditional cardiac auscultation is being replaced by digital phonocardiography, where algorithmic methods are used for heart sound analysis and diagnosis.

Heart sounds are acoustic signals, that are mainly generated by the blood flow turbulence within the arteries, as well as the vibrations of cardiac valves as they open and close during the cardiac cycle. The phonocardiogram (PCG) captures the fundamental heart sounds during a normal cardiac cycle. The first heart sound (*S1*) is produced by the closure of the *mitral* (MV) and *tricuspid* (TV) valves at the beginning of the systole. The second heart sound (*S2*) is produced by the closure of the *aortic* (AV) and *pulmonary* (PV) valves at the beginning of the diastole. The interval between S1 and S2 is called the systolic phase and the interval between the S2 and the S1 is called the diastolic phase. During the systolic and diastolic phases, turbulent blood flow may create enough vibrations to make audible heart sounds and abnormal waveforms in the PCG. Cardiac murmurs are the direct result of this blood flow turbulence. These murmurs can be separated into different categories after analyzing their location, their timing, their pitch etc. So, an expert that performs the auscultation is able to analyze these murmurs and identify cardiovascular physiologies or pathologies.

It is reasonable for the scientific community to aim to convert this expert-centered process to an automated one and use computer-assisted decision systems based on auscultation, to support physicians in their decisions. Until now, such systems do not provide satisfactory results, mainly due to the lack of large datasets, where a more detailed description abnormal waves is included. In order to cure this inadequacy scientists created *The CirCor DigiScope Dataset* [5], that is, at this moment, the largest pediatric heart sound dataset. It includes 5282 recordings from the four main auscultation locations of 1568 patients and each cardiac murmur has been manually annotated by an expert annotator according to its characteristics. The dataset provides the recordings, some demographic and clinical information (patient's sex, age, height, weight) and information about the murmur itself i.e. its existence, type, intensity etc. Based on this dataset, the George B. Moody PhysioNet Challenge 2022 invited teams to develop automated approaches for detecting abnormal heart function from multi-location PCG recordings of heart sounds. The results of the challenge can be seen in (https://moody-challenge.physionet.org/2022/results/).

In this project, we applied Machine Learning methods to the aforementioned dataset to identify if a given recording belongs to a patient with heart murmur. In order to solve this classification problem, we used the PCG's waveform and none of the other information provided by the dataset. We used the raw audio to extract sound and domain specific features that will be thoroughly explained in Section 2. On these features, we performed feature selection methods, as seen in Section 3.1. After, we proceeded to select the classification model by performing the parameter tuning in Section 3.2, that led us to the results mentioned in Section 3.3.

2 Feature Extraction

In this section we will provide an in-depth overview of the features we extracted, alongside with the choices that accompany the feature extraction procedure. Since we chose to only use the recordings provided by the dataset, we utilized Audio Signal Processing techniques in order to extract useful features suitable for the nature of our problem.

2.1 Data preprocessing

The dataset that was provided by the challenge included the recordings of 942 patients. For each patient, the dataset provides at most four recordings (.wav) of the patient's heart, each one from a different cardiac auscultation spot : AV, MV, PV and TV. Every waveform's duration varies from approximatelly 5 to 30 seconds.

Given this information, firstly, we disposed of the patients' records that did not include all four recordings, as we chose to use all four heart areas to predict murmur's existence, since that is the proccess that a doctor would follow. This led us to a dataset that includes 574 patients, where 117 were identified suffering from heart murmur and 457 were not. Note that the dataset is quite imbalanced, but that reflects on the distribution of people who suffer from heart murmur in real life.

Our second step, was to make sure that all waveform samples were of the same length. This was essential to the process, as many of the extracted features depend on the sample's size. So, we trimmed our samples to be 5 seconds. Also, in order to make our dataset more balanced, we performed *data augmentation*, so that our algorithms are trained on more positive examples. Specifically, the samples that lasted 15 or more seconds and were labelled as positive ones, were split into three 5-second samples and all of them were added to the dataset. This procedure, resulted to the partitioning of the dataset being: 457 patients with no murmur and 301 patients with murmur.

2.2 Features

In this project a total of 128 features from *time domain*, *frequency domain*, *cepstrum domain* and *statistical features* were extracted that have the potential to discriminate among the normal and murmur signals. These features were extracted from all four auscultation areas. They are the following and are explained below.

1. *Amplitude Envelope* (mean, median, standard deviation, 75% percentile)
2. *Total Energy*
3. *Root-Mean Square Energy* (mean, median, standard deviation, 75% percentile)
4. *Zero-Crossing Rate*
5. *Skewness*
6. *Kurtosis*
7. *Peak Frequency*
8. *Onset Detection*
9. *Band Energy Ratio*
10. *Autocorrelation*

11. *Spectral Centroid*
12. *Spectral Bandwidth*
13. *MFCCs*

2.2.1 Amplitude Envelope

The time-domain feature Amplitude Envelope (AE), provides the maximum amplitude value of all samples in a frame. It gives a rough idea of the signal's loudness, but is sensitive to outliers. According to [6], higher peak amplitude values are expected for murmur signals as compared to normal signals. We implemented a function that computes AE for all four heart regions. This feature alone is not representative of the signal's behavior and for that reason we computed the mean, the median, the standard deviation and the 75% percentile of the maximum values for all the frames in each audio signal.

2.2.2 Total Energy

The energy, or total magnitude of a signal, measures how loud an audio signal is. The murmur is a higher amplitude signal and hence is expected to have a higher value for this feature. We computed the energy of the signal by summing the squares: $E_x = \sum_n \mid x[n] \mid^2$.

2.2.3 Root-Mean Square Energy

Root Mean Squared of the Energy (RMSE), is a time-domain feature that indicates the loudness of a signal, but is less sensitive to outliers than AE. We computed, again, the mean, the median, the standard deviation and the 75% percentile of the RMSE in all frames. It is computed that way: $RMSE = \sqrt{\frac{1}{N} \sum_n \mid x[n] \mid^2}$.

2.2.4 Zero-Crossing Rate

The Zero-Crossing Rate (ZCR) indicates how many times a signal crosses the horizontal axis i.e. the rate at which the signal changes from positive to negative or back. A larger value of ZCR is expected for murmur signals. We compute the ZCR value using Python's library librosa [3]. This feature was selected by the feature selection method *Recursive Feature Elimination (RFE)* (see Section 3.1) as one of the most important features that provides a good separation of our classes. As seen in Figure 1, indeed the signal where heart murmur exists has larger ZCR values than the signal with no heart murmur.

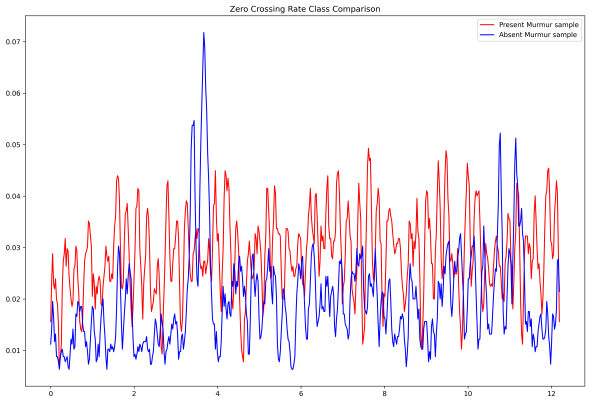


Figure 1: ZCR of absent and present murmur

2.2.5 Skewness

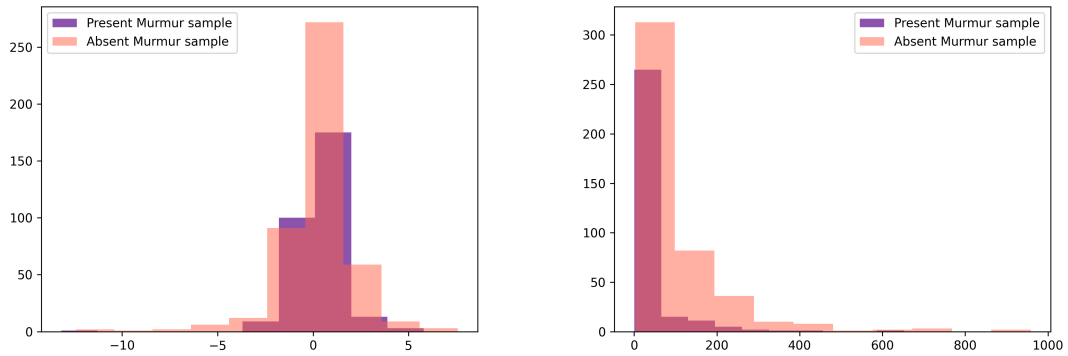
Skewness is a statistical measure of asymmetry of the data around the sample mean. In our case, skewness measures the signal's asymmetrical spread about its mean value. A distribution, or dataset, is symmetric if it looks the same to the left and right of the center point. A zero value indicates that the values are relatively evenly distributed on both sides of the mean, but not necessarily implying a symmetric distribution. If it is positive, then the data are spread out more to left of the mean than to the right and vice versa. The skewness of a distribution is defined as $Y = \frac{\mathbb{E}[x-\mu]^3}{\sigma^3}$.

We decided to use this feature, because heart signals without murmurs are repeated overtime and have no abnormal behaviors, resulting to skewness value around zero. Skewness of all four heart areas was a feature that was selected by RFE and the Figure 2a illustrates the histogram of the distribution of skewness' values for our problem classes for the PV area.

2.2.6 Kurtosis

Kurtosis is a measure of how outlier-prone a distribution, i.e. our signal, is. Distributions that are more outlier-prone than the normal distribution have kurtosis greater than 3, whereas distributions that are less outlier-prone have kurtosis less than 3. Here, we view kurtosis as a measure of the "peakedness" of the probability distribution of the heart signal. Higher kurtosis means more of the variance is the result of infrequent extreme deviations, as opposed to frequent modestly sized deviations.

The occurrence of a normal heart sound produces abrupt changes in the PCG signal. As a result, transient impulses corresponding to heart sounds occur cyclically in the signal. These transient impulses lead to instantaneous energy fluctuations at the heart sound locations. Hence, kurtosis of a normal signal contains peaks at these fluctuations and, for that reason, has a high value. On the other hand, a murmur is not an instantaneous peak, but rather a noise that lasts some amount of time. Thus, heart signals that represent murmur have frequent modestly sized deviations, resulting to a smaller kurtosis value. Figure 2b illustrates the distribution of kurtosis' values for the PV area for both signals with present and absent murmur. This feature was, also, selected by RFE, but only for two corresponding heart areas.



(a) Skewness of signals at PV

(b) Kurtosis of signals at PV

Figure 2: Skewness and Kurtosis of signals at PV with present and absent heart murmurs.

The following features are *frequency domain* features. This means that, for their computation the signal is transferred from time domain to the frequency domain using the *Fourier Transform (FFT)*. For some frequency features we want to preserve the time axis of each frequency's occurrence and for these we apply the *Short-Time Fourier Transform (STFFT)* in order to extract the so called *Spectrogram* that combines both time and frequency domains.

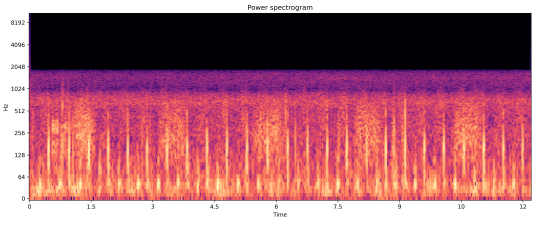
2.2.7 Peak Frequency

Peak frequency shows the frequency at which the peak amplitude occurs. We thought of this feature, as the murmurs and normal signals vary in amplitude and frequency. Thus, we implemented a function that computes its value via FFT.

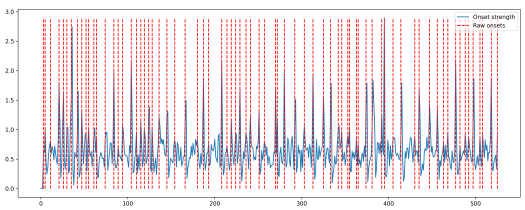
2.2.8 Onset Detection

As mentioned in Section 1, S1 and S2 are the he fundamental heart sounds during a cardiac cycle. S1 is audible on the chest wall, formed by the mitral and tricuspid components and S2 is also formed by two components, the aortic component and the pulmonary component. Since some types of murmurs (diastolic and systolic) occur during the diastolic and systolic periods and affect their duration, some useful features to compute would be the duration of S1, duration of S2, systole duration and diastole duration. However, since S1 can be heard at MV, TV and S2 at AV, PV and we have recordings from those specific areas we cannot compute the systole and diastole duration. Also, the durations of S1 and S2 respectivelly cannot be computed due to the noise caused by murmurs, snaps or the auscultation process itself. However, this behaviour led us to the idea of *onset detection*.

Generally, an onset marks the position at which the beginning of the transient part of a sound, or the earliest moment at which a transient can be reliably detected. In our case, onset detection helps us identify the start of S1, S2 and murmurs' peaks. So, for a fixed time the number of onsets of a heart signal with murmur is higher than the number of onsets of a signal with no murmur. Figures 3, 4 illustrate this exact behavior. Note that this feature, at all four heart areas, was one of the top picks of our feature selection method.

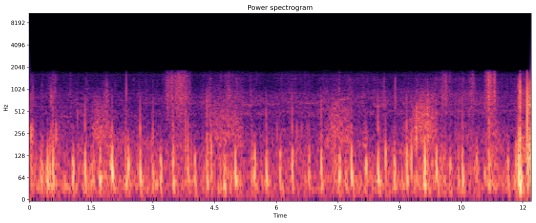


(a) Power spectogram of signal with murmur

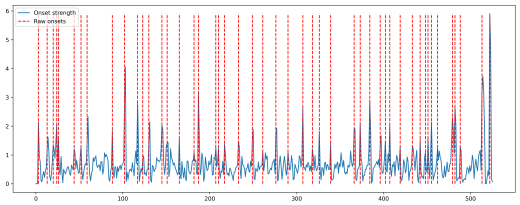


(b) Audio and onsets of signal with murmur

Figure 3: Onset detection of signals with murmur



(a) Power spectogram of signal without murmur



(b) Audio and onsets of signal without murmur

Figure 4: Onset detection of signals without murmur

2.2.9 Band Energy Ratio

This next feature, Band Energy Ratio (BER), does a comparison between the lower and higher frequency bands. It is a measure of how dominant low frequency bands are.

We computed the mean and the standard deviation of the BER of each frame, by applying to each frame the following formula,

$$BER_t = \frac{\sum_{n=1}^{F-1} m_t(n)^2}{\sum_{n=F}^{N} m_t(n)^2}$$

where F is the split frequency (a threshold) and $m_t(n)$ is the signal's power at n and t. This feature was selected by RFE as, as seen in Figure 5, signals without murmurs have higher BER values.

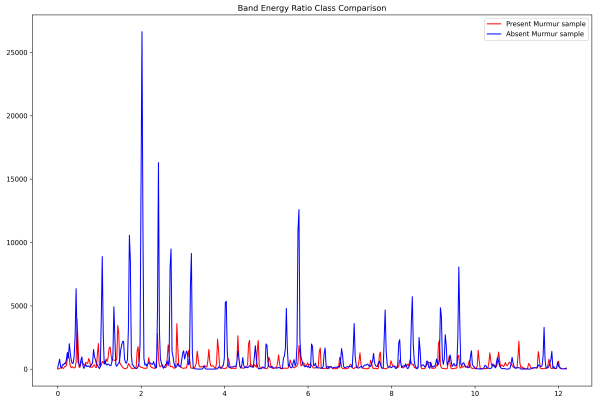


Figure 5: BER of absent and present murmur

2.2.10 Autocorrelation

Autocorrelation is the correlation of a function with itself. This feature is useful for finding repeating patterns in a signal, such as determining the presence of a periodic signal which has been buried under noise, or identifying the missing fundamental frequency in a signal implied by its harmonic frequencies. In our problem this feature could be proven useful as it identifies the periodicity of a heart cycle. We used librosa's function to compute autocorrelation in all four auscultation areas.

2.2.11 Spectral Centroid

Spectral Centroid (SC) is a weighted method to calculate the central frequency of a signal. It is a frequency band where most of the energy is concentrated. We computed the mean of the SC in each frame by applying to it the following formula,

$$SC_t = \frac{\sum_{n=1}^{N} m_t(n) \cdot n}{\sum_{n=1}^{N} m_t(n)}$$

where $m_t(n)$ is the signal's power at n and t. It was selected by our feature selection method. As seen in Figure 6, signals with murmurs have higher SC values. We used librosa's function to compute the SC in all four auscultation areas.

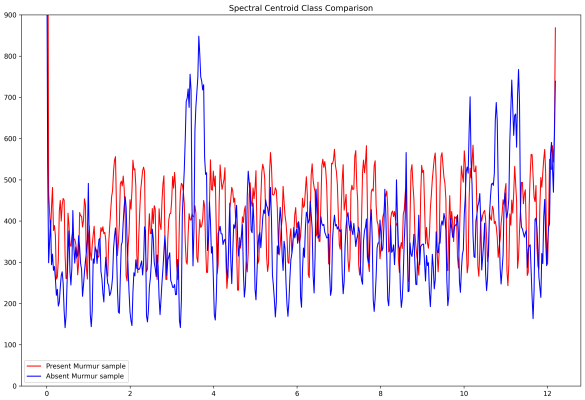


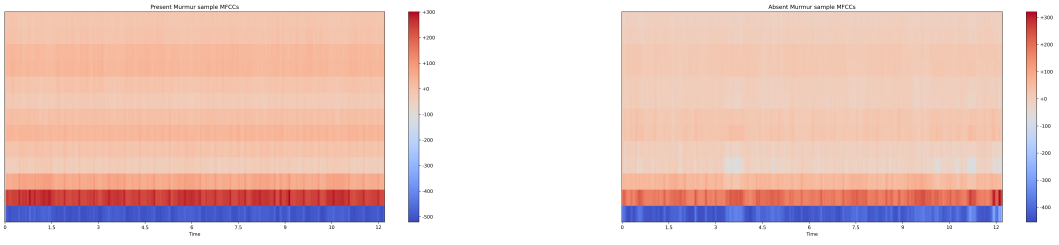
Figure 6: SC of absent and present murmur

2.2.12 Spectral Bandwidth

The Spectral Bandwidth (SB), or spectral spread, is a feature derived from the SC. It formulates the spectral range around the SC. More intuitively, it is the difference between the upper and lower frequencies in a continuous set of frequencies. As murmurs are high in frequency, a higher bandwidth is expected in case of murmur signals. For AV and TV, the SB is considered an important feature. Again, we computed it with librosa.

2.2.13 MFCCs

The Mel-Frequency Coefficients (MFCCs) of a signal are features that describe the overall shape of a spectral envelope using the Mel scale. The Mel scale represents pitch in a logarithmic manner. It is a *cepstrum* domain feature. Using librosa, we computed the 13 first MFCCs for each of the four heart regions. They proved to be important features as can be distinguished in Figure 7.



(a) MFCCs of signal with murmur

(b) MFCCs of signal without murmur

Figure 7: MFCCs of absent and present murmur

Note that we also computed the first and second derivatives the MFCCs for our signals, but we removed them as they weakened the performance of our algorithms.

3 Heart Murmur Classification

As mentioned in the introduction, we will perform heart murmur detection using machine learning. At this stage **we are using the audios for all the 4 regions AV, PV, MV and TV**. Given that we have labelled samples, we will use algorithms from the *supervised learning* family. The goal of supervised learning algorithms is learning a function (or *hypothesis*) that maps feature vectors to labels, based on input-output pairs. If these labels are discrete values, we call the above process *classification*. There are mainly three approaches for classification tasks. The first one is to construct a *Discriminant Function*, using the training dataset, that directly assigns each input vector $\mathbf{x}$ to a specific *class* (label). The rest two approaches involve the computation of the posterior probabilities $P(C_k|\mathbf{x})$, where C_k means the k-ith class. In *Probabilistic Discriminative Models*, we compute $P(C_k|\mathbf{x})$ directly, by representing them as parametric models and then optimize their parameters by performing *Maximum Likelihood Estimation* on the training dataset. Alternatively, in *Probabilistic Generative Models*, we compute the class-conditional densities given by $P(\mathbf{x}|C_k)$, together with the prior probabilities $P(C_k)$ for the classes, and then we compute the required posterior probabilities using Bayes' theorem. In our assignment we performed experiments using machine learning algorithms from all these three approaches.

In Section 2, we demonstrated our available features that describe each data sample $\mathbf{x}$. It is possible, some of the features that a machine learning model uses, to be redundant or even worse to limit the performance of the model. In the first case, that means that these features don't provide to the model with any extra information and in the second case, these features may add a significant amount of noise to our dataset. Also, the time complexity of training and inference of some algorithms, increases even exponentially with the number of features used. For these reasons, we may need to select a subset of the available features. In section 3.1 we present the the methods we used.

The process of defining the most suitable hypothesis for our classification problem is called *Model Selection*. We define a *hypotheses space* the moment we consider a specific choice of the hypothesis representation [4]. For example we define a hypotheses space, at the moment we decide to use a linear model that defines a hyperplane as a decision boundary between our classes. All the possible hypotheses (hyperplanes) that live in this hypotheses space, are defined by the parameters of these models. So, we would like to come up with a hypothesis that fits perfectly our training examples and does not fail on previously unseen data. In 3.2 we describe the steps we followed in order to select the best hypothesis for our problem. Finally, in 3.3 we report the final performance evaluation of the hypothesis we selected for our problem.

In order to carry out the aforementioned procedures, we need to divide our dataset into three subsets. The first one is called *training set*. We use the training set in order to fit our model, to specify our hypothesis. Our model will use the input features $\mathbf{x}$ and the corresponding labels $\mathbf{y}$ and will try to tune its parameters in order to optimize some criterion. The second one is called *validation set* and will be used in the model selection. The *testing set* will be used only for the final evaluation of our model and we didn't take any choices regarding our model or the features we selected, based on this set. As we mentioned in 2.1, we augmented the positive samples in our dataset. We should make sure that samples coming from the same patient are distributed in the same subset (training, validation or testing set). We are doing so, because we would like to validate and test our model in previously unseen data. A sample from a specific patient's audio, has a pattern that our model can partially capture, during the training phase using the training set. If we add a sample from the same audio in the testing or validation set, our model would face

something quite familiar and that couldn't be an unbiased approximation of our model's performance. After the data augmentation, our training set consists of 510 samples (67.28 %) and has a positive examples rate of 0.47. The validation set consists of 124 samples (16.36 %) and has 0.27 positive examples rate and the testing set has 124 samples (16.36 %) and 0.24 positive examples rate.

3.1 Feature Selection

As mentioned earlier, some features may limit the performance of a machine learning model, if for example they provide some information quite irrelevant than the problem we are trying to solve. We experimented with three feature selection methods. The first one is the *Lasso Logistic Regression*, where we leveraged the ability of this method to eliminate some features during the optimization of its parameters. Then we used the *ANOVA F-value* method, which is a statistical based measure and then an iterative schema, the *Recursive Feature Elimination*. Each of these methods produced a .txt file with the suggested features in ascending order, based on their importance. Then, the model selection module of our project was able to select only the features proposed from these methods, fit the models properly and evaluate.

3.1.1 Lasso Logistic Regression

In the first feature selection method, we fitted a *regularized* logistic regression model and then we took the coefficients of this model in order to sort our features by importance. This version of logistic regression is a linear model that uses this cost function in order to find its parameters a_j:

$$\frac{1}{2N_{training}} \sum_{i=1}^{N_{training}} (y_{real}^{(i)} - y_{pred}^{(i)})^2 + a \sum_{j=1}^{n} |a_j|$$

The above regularized cost function has the side effect that may ground some coefficients a_j to zero. The corresponding values of the feature vector x to these coefficients will play no role to the prediction made. So, what we did is to fit a model like this to our training and validation sets combined and discard the features that got zero coefficients. We did so, by performing Grid Search, where we used the Cross Validation technique in order to maximize the f1 score and select the most appropriate regularization coefficient a. You can see a sample of the first most important features below:

3.1.2 ANOVA F-value

ANOVA [2] is an acronym for "analysis of variance" and is a parametric statistical hypothesis test for determining whether the means from two or more samples of data (often three or more) come from the same distribution or not. An F-statistic, or F-test, is a class of statistical tests that calculate the ratio between variances values, such as the variance from two different samples or the explained and unexplained variance by a statistical test, like ANOVA. The ANOVA method is a type of F-statistic referred to here as an ANOVA f-test. Importantly, ANOVA is used when one variable is numeric and one is categorical, such as numerical input variables and a classification target variable in a classification task. The results of this test can be used for feature selection where those features that are independent of the target variable can be removed from the dataset.

3.1.3 Recursive Feature Elimination

RFE [1] is an efficient approach for eliminating features from a training dataset for feature selection. RFE is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is given and used in the core of the method, is wrapped by RFE, and used to help select features. This is in contrast to filter-based feature selections that score each feature and select those features with the largest (or smallest) score. Technically, RFE is a wrapper-style feature selection algorithm that also uses filter-based feature selection internally. RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains. Again here we fitted logistic regression models and we set a minimum amount of features to keep of 20.

3.2 Model Selection

This particular module of our project loads the training, validation and testing sets, among with the features that have been selected given the various techniques we experimented with and performs model selection. We used various models, defining different hypotheses spaces, from the three main categories described in the Section 3.2. For each classifier, we followed a process called *hyperparameter tuning* in order to find their best settings, that not only fit our training dataset, but also generalize well. These hyperparameters tune the complexity of our model.

If we give to our model much freedom and expressive power it will be able to fit perfectly our training set. But such complex models may be hard to generalize to unseen data. This phenomenon is called *overfitting*. The opposite to it is called *underfitting* and that states that our model doesn't have much flexibility in order to fit our training set properly. So, there is a trade-off between overfitting and underfitting and we should find the parameters and hyperparameters that balance this trade-off. A way to test for the underfitting case is to test our classifier on the training set, right after the training on it ends and report the *training error* . If we get high training error, that means that our model was unable to fit the training set. This should drive us to give more flexibility to our model via its hyperparameters or consider a different hypotheses space (a different classification algorithm). On the other hand, if we get a relatively low training error but when we test our classifier against the validation set we get a much higher error (*validation error*), we have evidence that our model suffers from overfitting. In this case we should restrict a bit our model, again using its hyperparameters, in order to increase its ability to generalize. The final evaluation of the model selected must be reported on the testing set which we have put aside until then.

For the algorithms that have hyperparameters we performed Grid Search. This is a process where we defined some possible values for each hyperparameter and in a loop we tried out every combination of them. Inside this loop, after fitting the model, we report the training error. We used the f1 metric because it combines both precision and recall. Then we test the model on the validation set and report the validation error. After creating these models we manually selected the model which has the higher f1 value on the validation error but the difference between training and validation error is the minimum. This is the model with the best performance that seems to generalize well. Note that on top of that, we used the features that our three feature selection methods suggested and

again maximized the f1 metric. At the end, we compute the baseline performance using a Dummy classifier that uses the probability distribution of the positive and negative samples (of the training set) in order to make a prediction. Our selected model got a significantly higher performance. However, a Dummy classifier doesn't always reflect the real difficulty of the problem and it is rather domain oriented. For example, in our case we augmented the positive samples, but in the real world the percentage of people who suffer from heart murmur is much less than 47%, which is what our training set consists of. The corresponding percentage in our test set is 24%, which is a bit more representative according to the real world case. Now, let's examine the parameters that each of our models selected and their final performance report based on the testing set.

Model	Grid Search Parameters	Model Selected	Training f1 Score	Validation f1 Score	Testing f1 Score
Logistic Regression	penalty : [l1, l2, none, elasticnet] C : [0.01, 0.1, 1, 10, 100, 1000]	C=0.01, penalty = elasticnet	0.745	0.66	0.704
SVM	C : [0.01, 0.1, 1, 10, 100, 1000] kernel : [linear, poly, rbf, sigmoid]	C=0.1, kernel=sigmoid	0.69	0.685	0.716
Naive Bayes			0.7		0.5
KNN	n_neighbors = 3, 5	n_neighbors = 5	0.84		0.51
Decision Tree	criterion : [gini, entropy, log_loss] max_depth : [10,50,100,200,500] min_samples_split: [2,5,10,20,40] min_samples_leaf': [1,5,10,20,50]	criterion=entropy, max_depth=50, min_samples_leaf=5 min_samples_split=20, splitter=random	0.8	0.68	0.49
LDA			0.82		0.51
QDA			0.97		0.3
ADABOOST	n_estimators : [10, 50, 100] learning_rate' : [0.1, 0.5, 0.8, 1.0] algorithm : [SAMME,SAMME.R]	algorithm=SAMME, $n_estimators = 100$	0.93	0.68	0.66

The SVM model had the greatest performance based on the testing set. The Logistic Regression was quite close also. We can see that the probabilistic generative models (Naive Bayes, LDA and QDA) were the most overfitted models. We should point out that the performance between training, validation and test set is very close in the models were we did our grid search. This validates the procedure we followed in order to avoid underfitting and overfitting. We performed model selection using the features suggested for all three feature selection methods plus also the case where all the features were included. The results presented in the current report use the 84 features that Recursive Feature Elimination produced. We would like just to note here that when we used all the available features we got a f1 value about 10% less than our final model with RFE features.

We experimented also with classifiers that use only one of the four examination regions (AV, PV, MV, TV). Although, the results couldn't be comparable, because there are many types of murmurs and each type can be noticed better from one of these regions. However, we would like to see if any of them had better performance on our dataset. We saw that the best classifier trained only on features of AV area had a noticeable increase in its performance, rather than the others. However, we should consider all the four regions in our initial case study, because we would like to perform general murmur detection and do not detect the exact type of murmur. In that case, we perhaps should follow a different approach regarding the 4 regions.

3.3 Evaluation

As mentioned before our final model is a SVM with C = 0.1 and a sigmoid kernel. Based on our testing set, our model has a Precision equal to 0.648, Recall equal to 0.8, a f1 score

of 0.716 and accuracy 0.846. You can see the confusion matrix in Figure 8.

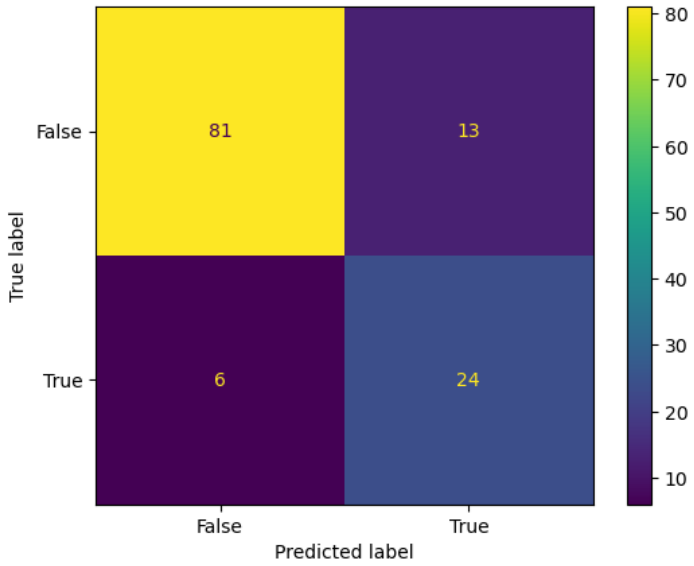
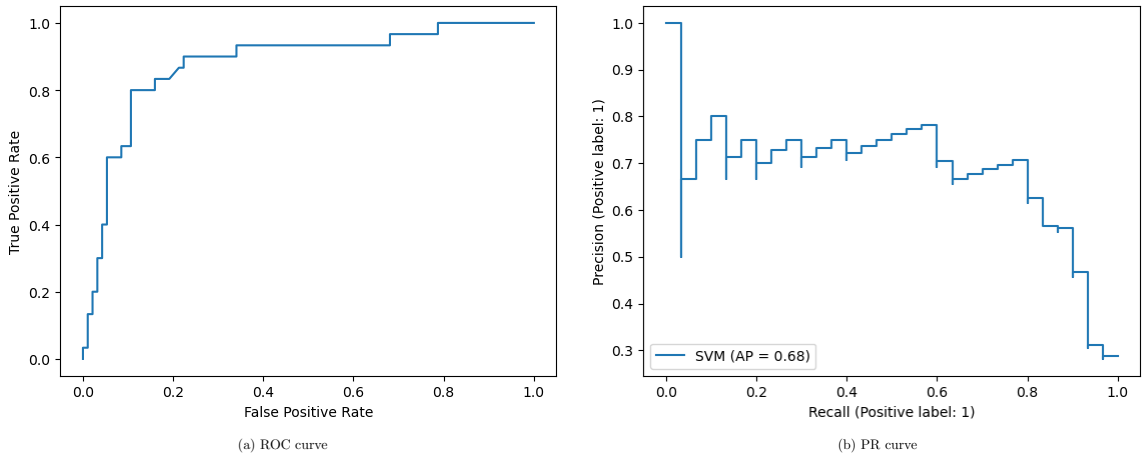


Figure 8: Confusion matrix of our final model

Given that the SVM above is the best model we got, we would like to see its probabilistic interpretation also and plot the corresponding roc curve. SVM does not have probabilities as output. It uses the sign() on the decision boundary to make predictions. What sklearn (the ML library we used) does is that it gives as an input the linear part of the SVM hypothesis to a sigmoid function, as a linear combination of two parameters A and B, that are defined using Maximum Likelihood Estimation during the training. AUC score is 0.879 and you can see the corresponding ROC and PR curves in Figure 9.



(a) ROC curve

(b) PR curve

Figure 9: ROC and PR curves

References

[1] Kjell Johnson Max Kuhn. *Applied Predictive Modeling*, volume 1. Springer, 2013.

[2] Kjell Johnson Max Kuhn. *Feature Engineering and Selection: A Practical Approach for Predictive Models*, volume 1. Chapman and Hall/CRC, 2019.

[3] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015.

[4] Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[5] Jorge Oliveira, Francesco Renna, Paulo Dias Costa, Marcelo Nogueira, Cristina Oliveira, Carlos Ferreira, Alipio Jorge, Sandra Mattos, Thamine Hatem, Thiago Tavares, Andoni Elola, Ali Bahrami Rad, Reza Sameni, Gari D. Clifford, and Miguel T. Coimbra. The CirCor DigiScope dataset: From murmur detection to murmur classification. *IEEE Journal of Biomedical and Health Informatics*, 26(6):2524–2535, jun 2022.

[6] Mandeep Singh and Amandeep Cheema. Heart sounds classification using feature extraction of phonocardiography signal. *International Journal of Computer Applications*, 77:13–17, 09 2013.