Fetal Health Classification

Machine Learning and Artificial Intelligence

Academic year 2020/2021

Filippo Grotto VR460638

August 31, 2021

Contents

1 Motivation and rationale

This project is heavily inspired by a kaggle task [2] about fetal health classification.

> Reduction of child mortality is reflected in several of the United Nations' Sustainable Development Goals and is a key indicator of human progress. The UN expects that by 2030, countries end preventable deaths of newborns and children under 5 years of age, with all countries aiming to reduce under-5 mortality to at least as low as 25 per 1,000 live births.
>
> Parallel to notion of child mortality is of course maternal mortality, which accounts for 295 000 deaths during and following pregnancy and childbirth (as of 2017). The vast majority of these deaths (94%) occurred in low-resource settings, and most could have been prevented.
>
> In light of what was mentioned above, Cardiotocograms (CTGs) are a simple and cost accessible option to assess fetal health, allowing healthcare professionals to take action in order to prevent child and maternal mortality. The equipment itself works by sending ultrasound pulses and reading its response, thus shedding light on fetal heart rate (FHR), fetal movements, uterine contractions and more.

In this context this project is not only an application of machine learning techniques but it also have some meaningful application into real world scenarios thanks to the real data provided.

2 Problem definition and Dataset

The dataset comes from UCI Machine Learning Repository [3] and it is composed by

> 2126 fetal cardiotocograms (CTGs) automatically processed with the respective diagnostic features measured. The CTGs are were classified by three expert obstetricians and a consensus classification label assigned to each of them. Classification was both with respect to a morphologic pattern (A, B, C. ...) and to a fetal state (N=normal; S=suspect; P=pathologic). Therefore the dataset can be used either for 10-class or 3-class experiments.

We will address the 3-class classification problem so we will try to classify the data according to normal, suspect or pathologic.

3 Methodology

The classification problem is addressed using the following techniques:

- Naive Bayes [4]
- SVM (with different kernels linear and RBF) [4]
- KNN (with different k based on preliminary analysis) [4]
- ANN [5]

Moreover, PCA and Fisher (LDA) dimensionality reduction techniques are considered as well as the effects of data scaling and/or normalization. We don't expect to see improvements of the state of the art where other techniques (XGBoost, Random forests, Decision tree models) are used.

4 Experiments and Results

4.1 Dataset exploration

The dataset is composed by 2126 entries with 22 features (21 feature + fetal health which is our target for the classification). From a first analysis there are no null or empty values but there are 13 duplications which we dropped. A brief description of the dataset is reported in Fig 1.

	count	mean	std	min	25%	50%	75%	max
baseline value	2126.0	133.303857	9.840844	106.0	126.000	133.000	140.000	160.000
accelerations	2126.0	0.003178	0.003866	0.0	0.000	0.002	0.006	0.019
fetal_movement	2126.0	0.009481	0.046666	0.0	0.000	0.000	0.003	0.481
uterine_contractions	2126.0	0.004366	0.002946	0.0	0.002	0.004	0.007	0.015
light_decelerations	2126.0	0.001889	0.002960	0.0	0.000	0.000	0.003	0.015
severe_decelerations	2126.0	0.000003	0.000057	0.0	0.000	0.000	0.000	0.001
prolongued_decelerations	2126.0	0.000159	0.000590	0.0	0.000	0.000	0.000	0.005
abnormal_short_term_variability	2126.0	46.990122	17.192814	12.0	32.000	49.000	61.000	87.000
mean_value_of_short_term_variability	2126.0	1.332785	0.883241	0.2	0.700	1.200	1.700	7.000
percentage_of_time_with_abnormal_long_term_variability	2126.0	9.846660	18.396880	0.0	0.000	0.000	11.000	91.000
mean_value_of_long_term_variability	2126.0	8.187629	5.628247	0.0	4.600	7.400	10.800	50.700
histogram_width	2126.0	70.445908	38.955693	3.0	37.000	67.500	100.000	180.000
histogram_min	2126.0	93.579492	29.560212	50.0	67.000	93.000	120.000	159.000
histogram_max	2126.0	164.025400	17.944183	122.0	152.000	162.000	174.000	238.000
histogram_number_of_peaks	2126.0	4.068203	2.949386	0.0	2.000	3.000	6.000	18.000
histogram_number_of_zeroes	2126.0	0.323612	0.706059	0.0	0.000	0.000	0.000	10.000
histogram_mode	2126.0	137.452023	16.381289	60.0	129.000	139.000	148.000	187.000
histogram_mean	2126.0	134.610536	15.593596	73.0	125.000	136.000	145.000	182.000
histogram_median	2126.0	138.090310	14.466589	77.0	129.000	139.000	148.000	186.000
histogram_variance	2126.0	18.808090	28.977636	0.0	2.000	7.000	24.000	269.000
histogram_tendency	2126.0	0.320320	0.610829	-1.0	0.000	0.000	1.000	1.000
fetal_health	2126.0	1.304327	0.614377	1.0	1.000	1.000	1.000	3.000

Figure 1: Description of all feature dataset

Another observation to make is the presence of high class imbalance that comes from the real dataset. In Fig 2 it is visible that **Normal** represents around the 57% of the entire dataset.

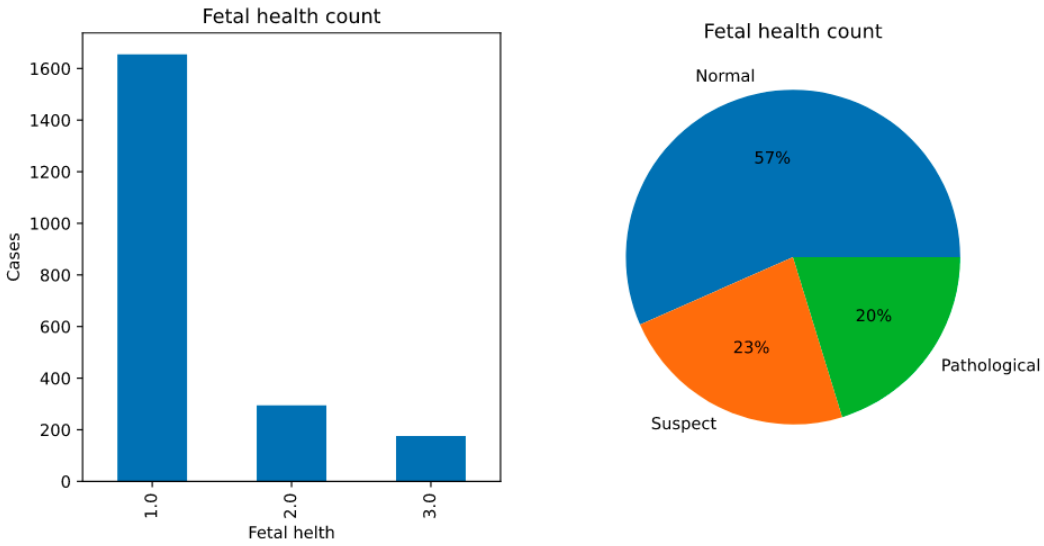

Figure 2: Class imbalance in the dataset, around 57% is classified as normal fetal health

Having verified the presence of class imbalance we will use different performance measures for our models in order to avoid misleading considerations:

- **Precision**
- **Accuracy**
- **Recall**
- **Confusion Matrix**
- **F1 Score**

Finally we can analyse the correlation between features of the dataset in order to evaluate a possible subset to improve our performances. In Fig 3 a heatmap is reported with the correlation between our features. In table 1 the features with more than 30% correlation with **fetal health** are reported

Feature Name	Correlation
accelerations	-0.364066
prolongued decelerations	0.484859
abnormal short term variability	0.471191
percentage of time with abnormal long term variability	0.426146

Table 1: Feature with correlation higher than 30%

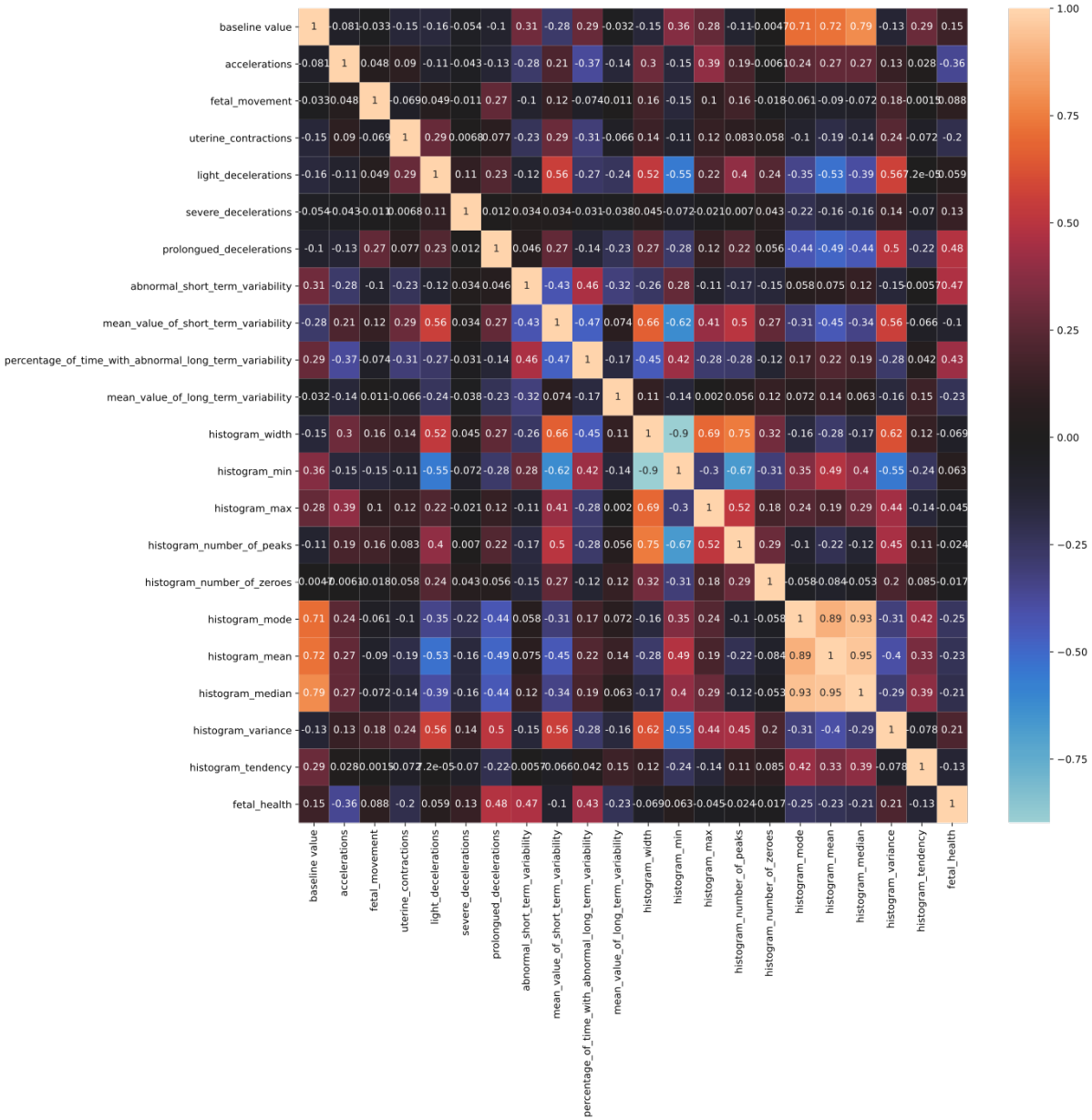

Figure 3: Correlation between features

As a result the features with high correlation are **accelerations**, **prolongued decelerations**,**abnormal short term variability** and **percentage of time with abnormal long term variability**. They will be considered in a further analysis as a reasonable subset of the dataset to exploit our models.

4.2 Data scaling

In Fig 4 the original data is reported. It's visible that the data requires a proper scaling to avoid the presence of dominant features that might affect our analysis. In Fig 5 the scaled data is reported where **StandardScaler** was used. Data normalization is not actually reported since from experiments it doesn't provide much more benefits and it's irrelevant for the sake of this report.

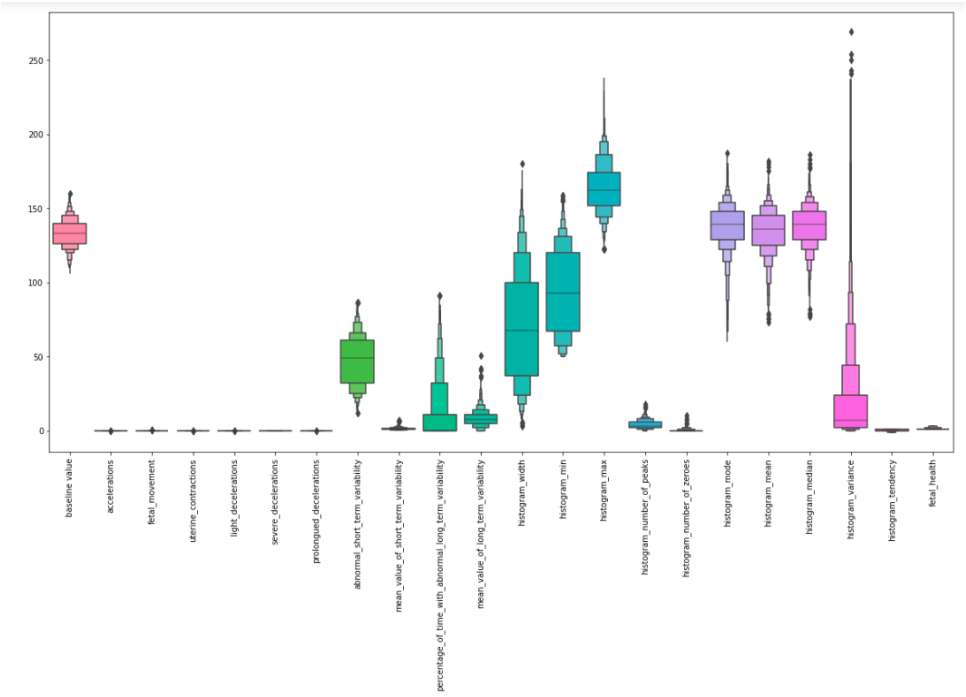

Figure 4: Plot of the original data without scaling

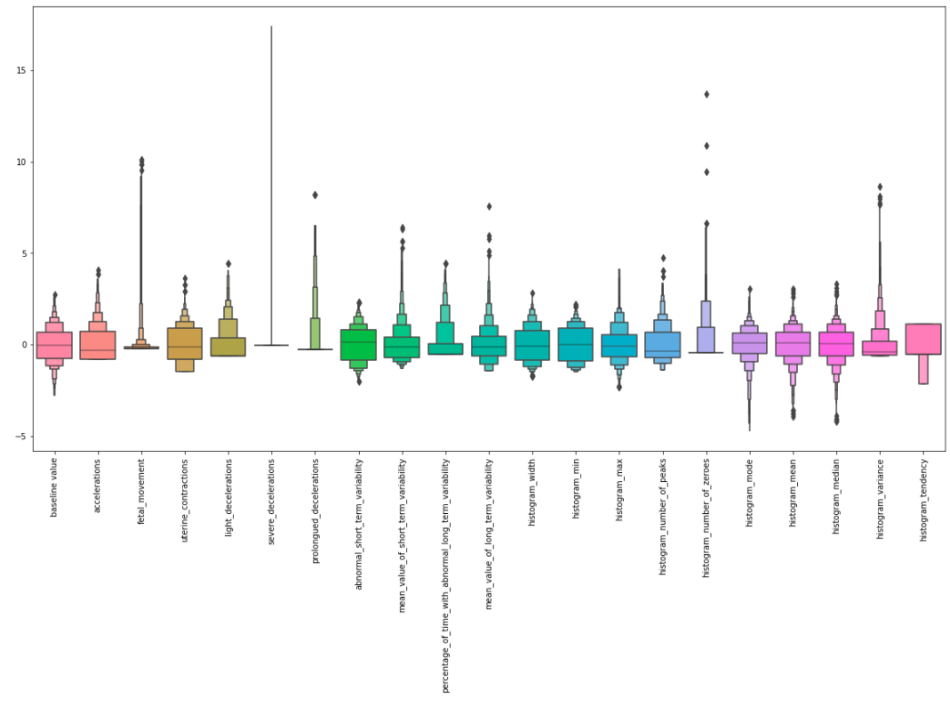

Figure 5: Plot of the scaled data

4.3 PCA dimensionality reduction

In Fig 6 the analysis of PCA dimensionality reduction is considered. In particular the graph rappresents the cumulative explained variation ratio with respect of all the possible components we might want to consider.

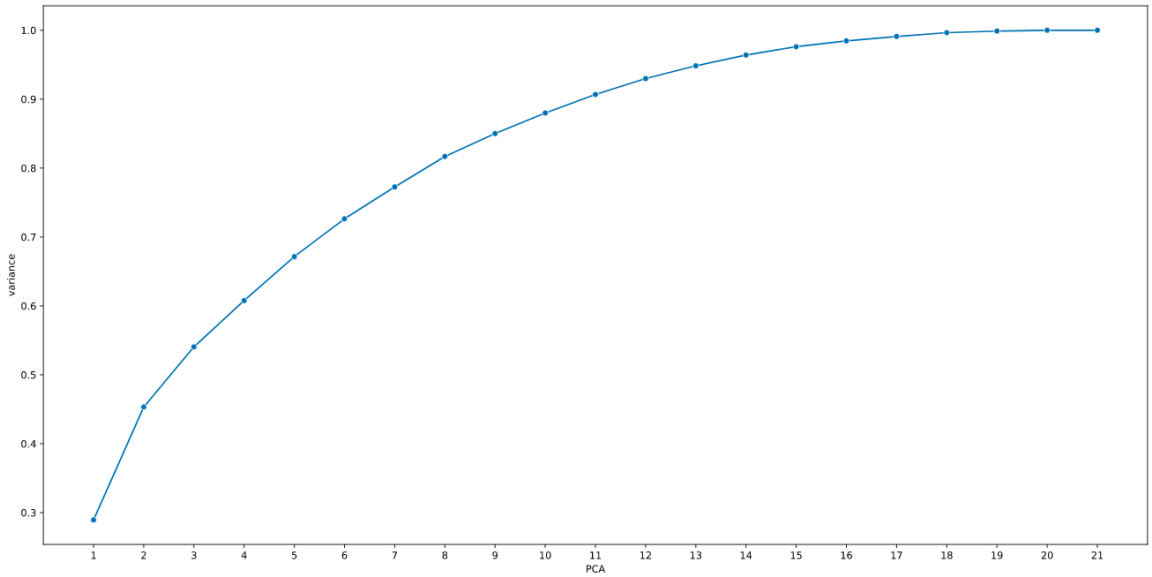

Figure 6: PCA cumulative explained variance ratio wrt of components

In particular the last five components don't provide much more information (close to zero) and with 8 components we have a representation of around 80% of our dataset. In table 2 the results obtained by our models over the 8 components is provided.

Model	accuracy	precision	recall	f1 score
SVM linear	0.899687	0.898669	0.899687	0.899022
SVM poly	0.891850	0.885607	0.891850	0.885699
SVM rbf	0.905956	0.902835	0.905956	0.903968
SVM sigmoid	0.717868	0.765628	0.717868	0.737608
Gaussian Naive	0.862069	0.865387	0.862069	0.861556
Logistic Regression	0.899687	0.897470	0.899687	0.898194

Table 2: Evaluation of our models over 8 PCA components

4.4 LDA dimensionality reduction

In this section a brief analysis of Linear Discriminant Analysis is reported in Fig 7 where we can easily see that the cumulative variance ratio accumulated with 1 component rappresents more than 80% of our dataset Moreover, both the case with 1 and 2 components are reported in the two tables 3 and 6.

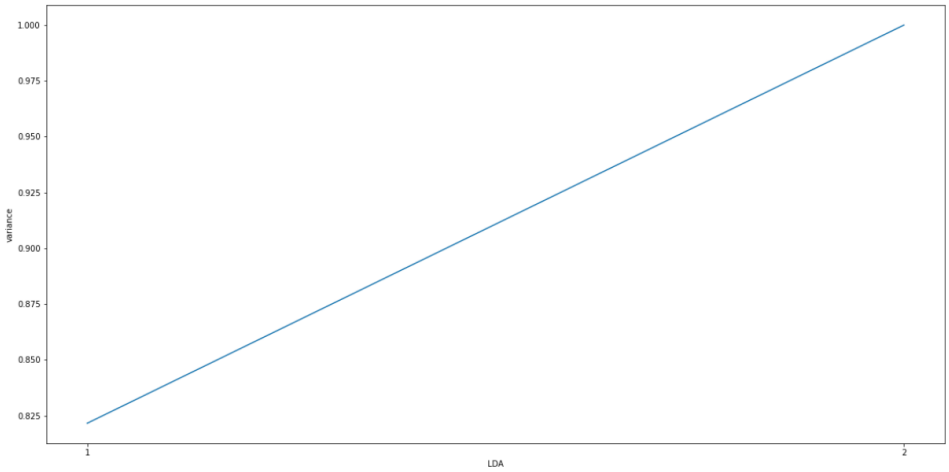

Figure 7: LDA cumulative explained variance ratio wrt of components

Model	accuracy	precision	recall	f1 score
SVM linear	0.847962	0.829295	0.847962	0.828071
SVM poly	0.841693	0.820448	0.841693	0.802800
SVM rbf	0.846395	0.832120	0.846395	0.836413
SVM sigmoid	0.766458	0.776691	0.766458	0.771115
Gaussian Naive	0.854232	0.847738	0.854232	0.850451
Logistic Regression	0.844828	0.828893	0.844828	0.832487

Table 3: Evaluation of our models with 1 LDA component

Model	accuracy	precision	recall	f1 score
SVM linear	0.898119	0.898106	0.898119	0.898103
SVM poly	0.899687	0.895180	0.899687	0.894518
SVM rbf	0.905956	0.907072	0.905956	0.906281
SVM sigmoid	0.689655	0.725761	0.689655	0.704979
Gaussian Naive	0.873041	0.885894	0.873041	0.877463
Logistic Regression	0.902821	0.898863	0.902821	0.899858

Table 4: Evaluation of our models with 2 LDA components

In general in case of uniformly distributed data, LDA almost always performs better than PCA. However if the data is highly skewed (irregularly distributed) then it is advised to use PCA since LDA can be biased towards the majority class.

4.5 Model evaluations with the full dataset

In this section the evaluations of several models are presented. For all of the the dataset has been divided using *train_test_split* function from sklearn using around 30% of the entire dataset for the testing set. The results of our models are reported in Fig 8. It's visible that SVM with rbf kernel produces the best results over our dataset. This is coherent with the analysis in [4] and what the literature provides us about Support Vector Machines Kernels. Let's consider:

- 1.0 is Normal class
- 2.0 is Suspicious class
- 3.0 is Pathologic class

Gaussian NB

	precision	recall	f1-score	support
1.0	0.99	0.66	0.79	496
2.0	0.37	0.94	0.53	101
3.0	0.47	0.61	0.53	41
accuracy			0.70	638
macro avg	0.61	0.74	0.62	638
weighted avg	0.86	0.70	0.73	638

Logistic Regression

	precision	recall	f1-score	support
1.0	0.94	0.94	0.94	496
2.0	0.68	0.62	0.65	101
3.0	0.65	0.83	0.73	41
accuracy			0.88	638
macro avg	0.76	0.80	0.77	638
weighted avg	0.88	0.88	0.88	638

SVM linear

	precision	recall	f1-score	support
1.0	0.96	0.94	0.95	496
2.0	0.70	0.69	0.70	101
3.0	0.73	0.85	0.79	41
accuracy			0.90	638
macro avg	0.79	0.83	0.81	638
weighted avg	0.90	0.90	0.90	638

SVM poly

	precision	recall	f1-score	support
1.0	0.92	0.97	0.94	496
2.0	0.76	0.56	0.65	101
3.0	0.78	0.78	0.78	41
accuracy			0.89	638
macro avg	0.82	0.77	0.79	638
weighted avg	0.89	0.89	0.89	638

SVM rbf

	precision	recall	f1-score	support
1.0	0.95	0.97	0.96	496
2.0	0.79	0.73	0.76	101
3.0	0.84	0.78	0.81	41
accuracy			0.92	638
macro avg	0.86	0.83	0.84	638
weighted avg	0.91	0.92	0.92	638

SVM sigmoid

	precision	recall	f1-score	support
1.0	0.88	0.85	0.87	496
2.0	0.63	0.50	0.56	101
3.0	0.20	0.39	0.27	41
accuracy			0.77	638
macro avg	0.57	0.58	0.57	638
weighted avg	0.80	0.77	0.78	638

Figure 8: Results of the selected models over our dataset

From a brief analysis we can clearly see that SVM in general works pretty well over our dataset, Naive Bayes, Logistic Regression and SVM with sigmoid kernel are pretty bad in terms of precision for **suspicious** and **pathologic**. In general the class **Normal** has better performances due to the class imbalance (more than 50%). Moreover in Fig 9 the confusion matrix for the svm with rbf kernel is reported and in Fig 10. What is clearly visible is the fact that for SVM with rbf there are close to zero false positive and false negative values which is a pretty good result. For Naive Bayes we have a false negative value which is even higher than the true value which is pretty bad for our classification problem.

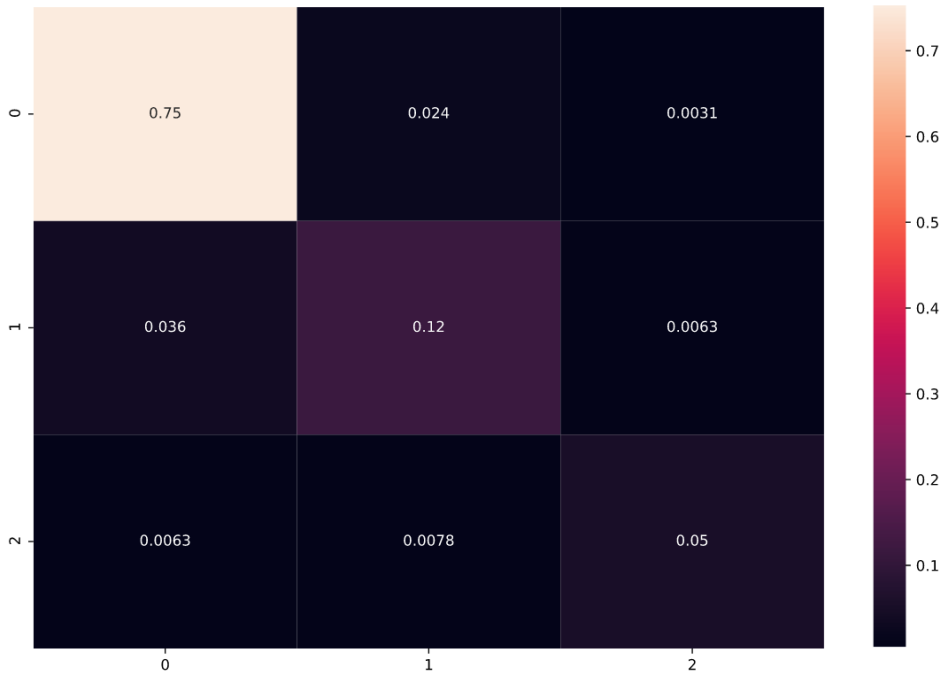

Figure 9: Confusion matrix of SVM with rbf kernel

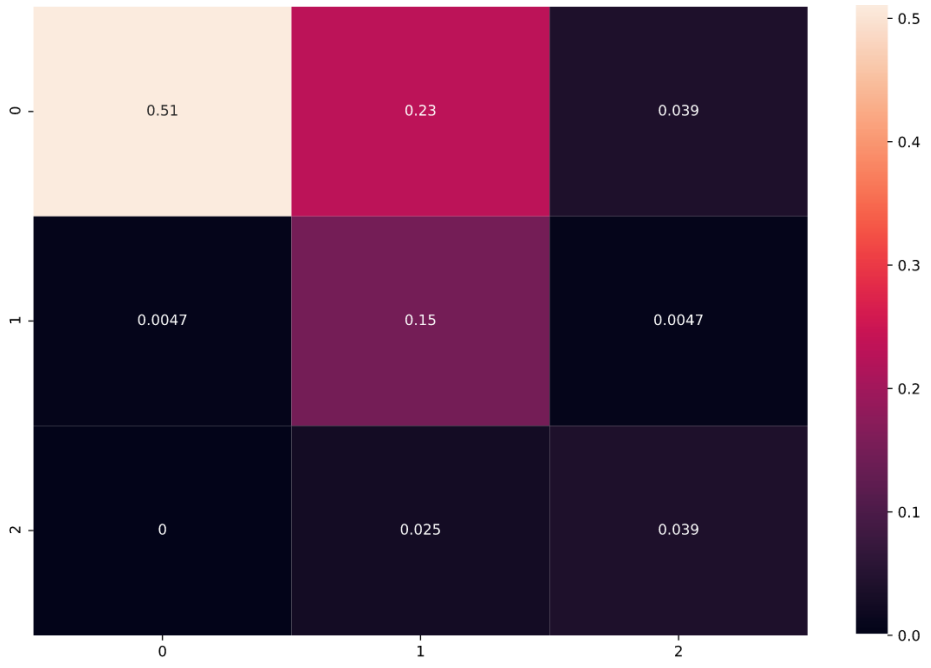

Figure 10: Confusion matrix of Naive Bayes approach

4.5.1 K-Nearest Neighbors

In this section we will consider the use of K-NN (with an euclidean distance applied) to our dataset. In Fig 11 a brief analysis of several k is provided with the related metrics to get an idea of which are the most promising values of K to use.

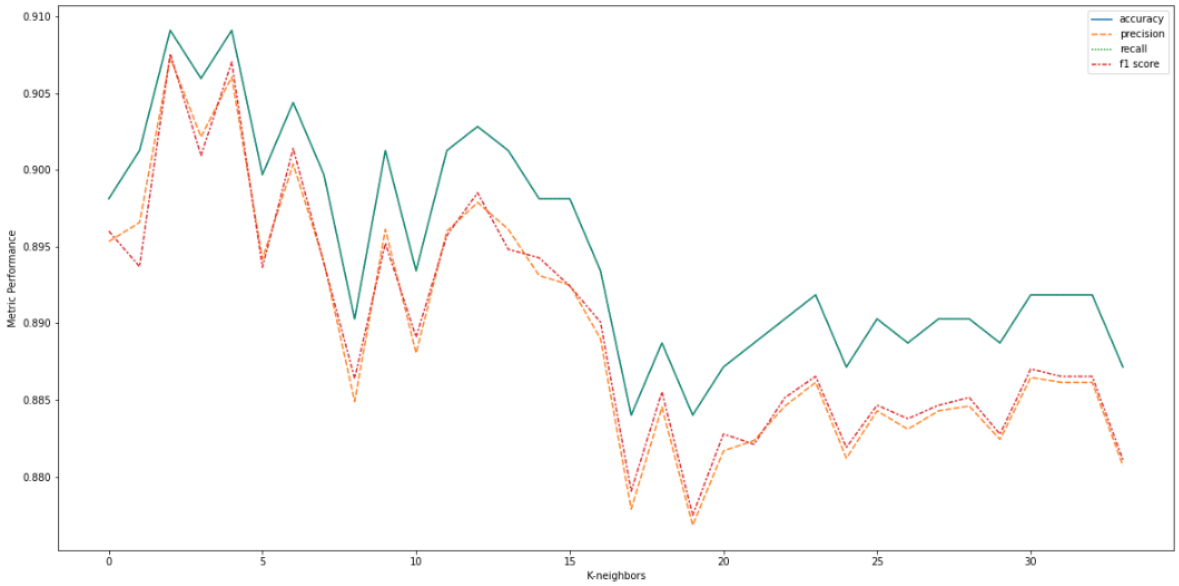

Figure 11: K-NN metrics with respect of the value K-neighbors

As a result the values K = 2 and K = 4 are the best choices which is very close to what the literature provides use which is $\sqrt{N}$ where N is the number of features. In this case $\sqrt{21} = 4.6$ which is pretty close to what we get from the practical analysis. In Table 5 the two best cases are considered and analysed.

Model	accuracy	precision	recall	f1 score
K-NN with k=2	0.90	0.90	0.90	0.89
K-NN with k=4	0.91	0.91	0.91	0.91

Table 5: Evaluation of K-NN models

As a conclusion K-NN with k=4 or k=2 are great models for our analysis

4.5.2 Artificial Neural Networks MLP

In this section we will brief provide some results obtained by using Artificial Neural Networks to solve the classification problem. In particular we created a multilayer perceptron (MLP) with the following parameters

```
input_size = 21
hidden_size = 100
num_classes = 3
num_epochs = 100
batch_size = 50
learning_rate = 0.001

# Fully connected neural network with one hidden layer
class NeuralNet(nn.Module):
  def __init__(self, input_size, hidden_size, num_classes):
      super(NeuralNet, self).__init__()
      self.fc1 = nn.Linear(input_size, hidden_size)
      self.relu = nn.ReLU()
      self.fc2 = nn.Linear(hidden_size, num_classes)

  def forward(self, x):
      out = self.fc1(x)
      out = self.relu(out)
      out = self.fc2(out)
      return out
```

The MLP uses one fully connected layer, a ReLU as activation function and another fully connected layer (Linear). This models proved to have an accuracy around 91% which makes it a good model for the classification task. In Fig 12 the train accuracies and losses are reported.

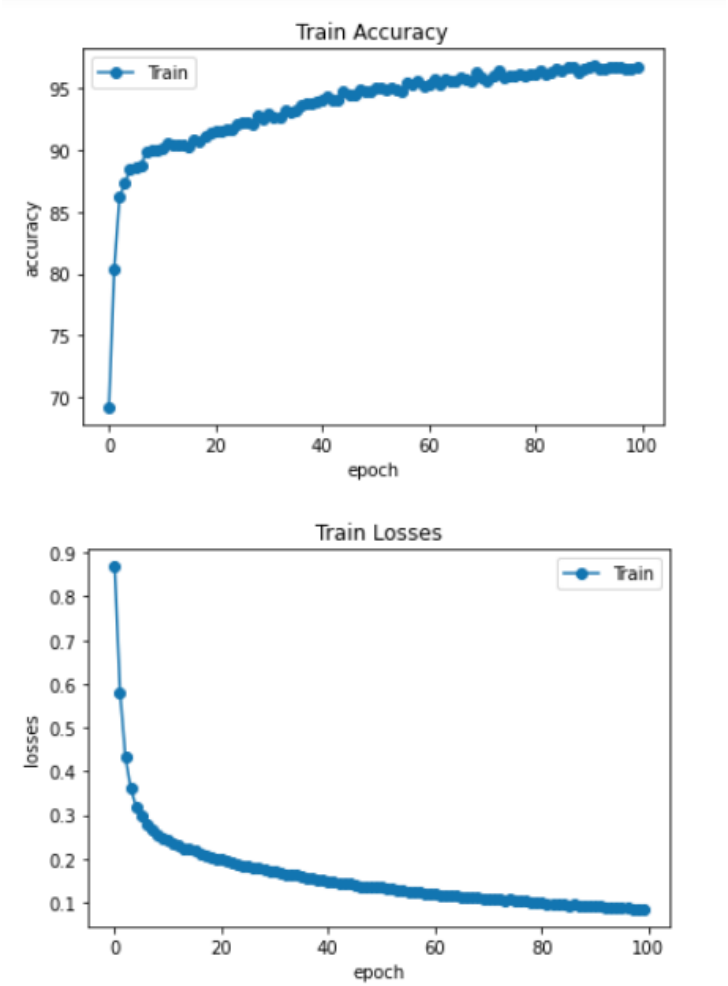

Figure 12: Accuracy and losses of the MLP

4.5.3 Convolutional Neural Network CNN

In this section the use of CNN for solving the classification problem is brief introduced

4.6 Other Exploratory Tasks

4.6.1 Models evaluation using a subset of features

In this section an evaluation of our models over a subset of features defined in the previous section is presented. Only features with more than 30% correlation with the target **fetal health** are used. They are reported in Table 1. In Table **??** the evaluation of our models is reported according to the defined metrics

Model	accuracy	precision	recall	f1 score
SVM linear	0.85	0.85	0.84	0.85
SVM poly	0.90	0.90	0.90	0.90
SVM rbf	0.88	0.88	0.88	0.88
SVM sigmoid	0.82	0.82	0.82	0.82
Gaussian Naive	0.84	0.84	0.86	0.84
Logistic Regression	0.88	0.88	0.88	0.88

Table 6: Evaluation of our models over a subset of features

4.6.2 Semisupervised Problem

5 Conclusion

In this report the study of the classification health of the fetal has been analysed using a real dataset. Several different models have been analysed using different metrics and different dimensionality reduction methodology. It is showed how using SVM with rbf kernel and 17 PCA components produces the best results. Finally, several other classification methods has been exploited and compared using different metrics.

References

[1] Ayres de Campos et al. (2000) SisPorto 2.0 A Program for Automated Analysis of Cardiotocograms

[2] Fetal Health Classification https://www.kaggle.com/andrewmvd/fetal-health-classification

[3] Cardiocotography dataset https://archive.ics.uci.edu/ml/datasets/cardiotocography

[4] Hoodbhoy, Zahra et al. "Use of Machine Learning Algorithms for Prediction of Fetal Risk using Cardiotocographic Data."

[5] Yılmaz, E. Fetal State Assessment from Cardiotocogram Data Using Artificial Neural Networks.