Supervised Learning Techniques for Gender Recognition by Voice and Speech Analysis

José Andrés Carvajal Bautista
Department of Mathematical Sciences
Mathematical Engineering
Universidad EAFIT
Medellín, Colombia
jacarvajab@eafit.edu.co

David Andrés Romero Millán
Department of Mathematical Sciences
Mathematical Engineering
Universidad EAFIT
Medellín, Colombia
dromero1@eafit.edu.co

***Abstract*—This work applies five different supervised learning techniques (Decision Tree, Linear SVM, Polynomial SVM, RBF SVM, and Multilayer Perceptron) to a *Voice Gender* dataset that was built to distinguish a voice as male or female, based upon their speech's audile properties. Case in point, the workflow operates within two spaces, a high-dimensional space of 20 acoustic features and a 2-dimensional space using Barnes Hut t-SNE algorithm, to assess which learner renders the most suitable performance metrics in each scenario. Results show that, in the high-dimensional space, the Decision Tree and the $\mathcal{L} = [20, 9, 9, 10, 2]$ with $\eta = 0.9$ MLP setup tied in the first place as the most competent learning machines with accuracy figures above the 0.98 mark. In addition, no sufficient evidence was found to point out that the complexity of the architecture has a strong relationship with the obtained metrics in that scenario. Contrastingly, the embedding procedure furnished no meaningful benefits within this study, as neither trained models performed decently in that context.**

I. Introduction

Detecting the gender (male or female) of a voice signal poses a challenging task to computerized systems. In contrast to humans, computer programs do not have the inherent capacity to extract distinguishable features from voice samples to derive personal attributes such as gender, nationality, dialect, emotion, age, or language fluency. Undoubtedly, there is a significant social need to classify gender accurately even in conditions of deprived or less than impeccable sensible input. In fact, gender classification is, in many cases, a trivial problem for the human brain as people learn over time to classify males and females from peculiarities like pitch, timbre, frequency, and breathiness [1].

Gender plays a significant role in the day-to-day lives of the members of society. Given the rise of voice recognition applications, gender classification has emerged as a notable subject of examination as it increases the interpretability of voice signals [2]. Being that the case, gender identification has various potential use-cases. In surveillance, voice gender perception might help enhance security systems' ability to unmask criminals hiding their identity [3]. Such a technique could be used to reduce search efforts and speed up critical investigations. Additionally, in digital marketing, gender classification can provide indicative data to afford customized services to the users of a platform [4].

This work provides new insights regarding how various classification techniques can be used to identify the speaker's gender from a voice sample. The idea is to apply five distinct supervised learning machines, both in high-dimensional and low-dimensional feature spaces, to determine which learner procures the most solid performance metrics.

The present work is organized as follows: Section II introduces some relevant literature that examines how machine learning methods have been used to tackle gender recognition in speech samples. Section III describes the proposed workflow and its details. Later, Section IV reveals the learners' results, in the two feature spaces, to determine which one derives the finest quality attributes. Finally, Section V presents the concluding remarks.

II. State of the Art

To date, several publications have investigated the application of artificial intelligence techniques for gender classification in voice samples [2], [5], [6]. For instance, the work in [2] proposes a stacked gender classification algorithm using the acoustic parameters of a set of voice samples. In fact, this publication introduces a scheme that mixes Classification Trees (CTs), Support-Vector Machines (SVMs), and Neural Networks (NNs) and ensembles them together to derive sounder classification metrics. Verily, the scholars secure a 96.74% accuracy score. However, the authors admit that stacking an SVM did not procure notable benefits.

A similar approach was followed in [7] to achieve the same goal. In this case, 3000 voice samples in `.WAV` format are pre-processed by extracting 22 acoustic properties of each signal. The authors implement a Multilayer Perceptron Network (MLP) with one input layer, four hidden layers, and one output layer. For validation purposes, a 5-fold cross-validation setup was implemented to obtain an average classification score. The scholars use a softmax activation function and a 0.25 dropout at each hidden layer. Positively, the authors obtain a 97.64% accuracy score. Nonetheless, the researchers claim that a larger dataset is needed to minimize incorrect classifications.

Lastly, a comparable strategy is implemented in [5] with a different methodology. In this instance, 3168 American English voice samples are used to develop a Deeper Long

Short Term Memory (LSTM) network that is able to achieve a 98.4% accuracy rate. Likewise, 20 features are extracted from each sample to build a high-dimensional space. Remarkably, the authors apply several relief-based methods to obtain the top 10 features of the dataset. In addition, the scholars admit that more data is needed to achieve even better results.

Overall, these studies highlight the interest in developing new methods to increase gender classification accuracy in audio samples. These investigations indicate that many machine learning algorithms have been proven effective in many related scenarios. Despite that, these papers suggest that the problem is far from being fully resolved. Given all that has been mentioned so far, some common flaws were detected, mainly concern with the exclusive use of euclidean norms.

III. Methods

As mentioned earlier, five classification techniques were employed to build gender-detection learners using the acoustic properties of voice samples. The following subsections present the task's specifications in greater detail.

A. Dataset

The *Voice Gender* dataset was built to distinguish a voice as male or female, based upon their speech's acoustic properties. In particular, the dataset consists of $3,168$ recorded voice samples collected from male and female talkers with no further details regarding age, language, or any other distinctive features [8]. Moreover, its author pre-processed the before-mentioned instances using acoustic analysis in the human vocal range (0 Hz - 280 Hz). Besides, very few details are given of the pre-processing phase. Notwithstanding, the scholar mentions she measured 20 acoustic parameters in each sample using the `WarbleR` R package.

B. Features

The following 20 acoustic properties of each voice sample are available for analysis:

- *meanfreq*: mean frequency (in kHz)
- *sd*: standard deviation of frequency
- *median*: median frequency (in kHz)
- *Q25*: first quantile (in kHz)
- *Q75*: third quantile (in kHz)
- *IQR*: interquantile range (in kHz)
- *skew*: skewness
- *kurt*: kurtosis
- *sp.ent*: spectral entropy
- *sfm*: spectral flatness
- *mode*: mode frequency
- *centroid*: frequency centroid
- *peakf*: peak frequency (frequency with highest energy)
- *meanfun*: average of fundamental frequency measured across acoustic signal
- *minfun*: minimum fundamental frequency measured across acoustic signal
- *maxfun*: maximum fundamental frequency measured across acoustic signal
- *meandom*: average of dominant frequency measured across acoustic signal
- *mindom*: minimum of dominant frequency measured across acoustic signal
- *maxdom*: maximum of dominant frequency measured across acoustic signal
- *dfrange*: range of dominant frequency measured across acoustic signal

C. Workflow

The proposed pipeline for the classification task includes the following sequence of steps. Firstly, the dataset is pre-processed using conventional scaling and cleaning techniques. Such procedure procures a 20-dimensional feature space in the unit hypercube. Verily, normalization is deemed particularly useful in many machine learning algorithms [9]. Indeed, it reduces the chance of a particular feature governing over the others in the context of the objective function [10]. Secondly, the *Barnes Hut t-SNE* algorithm is applied to reduce the original space's dimension and explore how the selected supervised methods operate in such a scenario. The aforementioned approach is credited as an exceptional procedure for visualizing high-dimensional data [11].

Thirdly, once the two spaces are built, a "Probably Approximately Correct" (PAC) analysis is carried out to resolve the optimal sample size to use in order to guarantee with probability greater than $1-\delta$ that for every hypothesis $h \in H$ with training error close to zero, h also has a true error bounded by some constant ϵ, that can be made arbitrarily small. Certainly, PAC theory provides a framework to determine how many training instances are required to converge to a successful hypothesis with a high probability. In this case, given fixed pairs of error bounds (δ, ϵ), a set of adequate sample sizes is procured for each learner.

In the fourth step, each dataset is split with the standard $60-20-20$ rule for the training, validation, and test set, respectively. Later, in the fifth step, the training subset is employed to build the following learning machines: *Decision Tree*, *Linear SVM*, *Polynomial SVM*, *Radial basis function SVM*, and *Multilayer Perceptron*. Next, in the sixth step, each classifier is evaluated in its generalization capability following the computation of several quality metrics in the validation and test sets. Finally, a performance assessment is carried out to settle on the most suitable learner for each space.

D. Pre-processing

As mentioned earlier, pre-processing is made by scaling each feature to the $[0, 1]$ interval. Such a state is achieved by applying Equation 1.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad \forall i = 1, \ldots, n \tag{1}$$

where, $x = (x_1, \ldots, x_n)$ is the feature vector and z_i is the i^{th} normalized data point. No supplementary cleaning techniques are used since the dataset was found in excellent conditions.

E. Probably Approximately Correct Learning

As mentioned above, PAC learning is a scheme that provides a set of mathematical tools to study the convergence of supervised learning machines [12]. The main question is, given a classifier with a low training error, What can be said about the generalization error? To dissect this problem, one has to assume a probability distribution P over the space X, and a training set S being randomly sampled from X under P, following that S is an adequate representation of future instances. In fact, the goal is to find a hypothesis $h \in H$ that agrees with the target class over S and renders low true errors. However, X being partially unknown makes this quandary even trickier.

In simple terms, PAC learnability means that if one possesses a large enough training set concerning some property of the hypothesis class H, then with high probability, every hypothesis $h \in H$ has a training error close to its true error [13]. Indeed, formalizing this notion, we have that if S of size N is taken such that,

$$N \geq \frac{1}{\epsilon}\left(\ln|H| + \ln\frac{1}{\delta}\right) \tag{2}$$

then, the probability of procuring a high-performant learner is defined by,

$$P\Big(|error_{true}(h) - error_S(h)| \leq \epsilon\Big) \geq 1 - \delta \tag{3}$$

where, ϵ and δ are the true and training error bounds, respectively. Nevertheless, it is clear from Equation 2 that H is assumed to be a finite set. If such a requirement is not met, then the *Vapnik-Chervonenkis Dimension* (VC-dimension) is used instead. The VC-dimension of a hypothesis class H is the cardinality of the largest set S that H can shatter, i.e., the size of the biggest collection S that a hypothesis $h \in H$ can split in all $2^{|S|}$ ways [14]. It happens that the size of the training set S in terms of the VC-dimension must be,

$$N \geq \mathcal{O}\left(VC_{dim}\left(H\right)\ln\frac{1}{\epsilon} + \ln\frac{1}{\delta}\right) \tag{4}$$

Having said that, for Decision Trees with a fixed depth k, its optimal sample size can be computed as follows,

$$N \geq \frac{\ln 2}{2\epsilon^2}\Big(\left(2^k - 1\right)\left(1 + \log_2 m\right) + 1 + \ln\frac{1}{\delta}\Big) \tag{5}$$

where, m is the number of features in the dataset. In contrast, for Support-Vector machines, a more general expression is employed to acknowledge the VC-dimension of the hypothesis class under a specified kernel. Veritably, in this case, the minimum sample size is calculated as shown below,

$$N \geq \max\left\{\frac{4}{\epsilon}\ln\frac{2}{\delta}, \frac{8VC_{dim}\left(H\right)}{\epsilon}\ln\frac{13}{\epsilon}\right\} \tag{6}$$

where, $VC_{dim}\left(H\right)$ is the VC-dimension of the SVM under the selected kernel. As a matter of fact, of the examined kernels, only the linear and r-degree polynomial ones hold a finite VC-dimension. That being the case, the VC-dimensions for those two are $m + 1$ and $\binom{m+r}{r}$, respectively.

F. Neural Networks

A fully-connected Multilayer Perceptron (MLP) module was implemented from scratch according to the guidelines presented in [15]. However, since the network architecture generally influences the learning outcomes, all permutations up to three hidden layers are trained with a different number of neurons in each layer. In particular, for the 20-d space, the neurons at i-th hidden layer (l_i) are taken between five and ten. In contrast, for the 2-d space, l_i is selected from one to three. The setup above renders a total of 297 different architectures. Notwithstanding, such a scheme is replicated for learning rates as $\eta = 0.2, 0.5, 0.9$, yielding 891 Neural Networks to train.

The mentioned MLPs are trained sequentially, fixing a maximum of 50 epochs and a tolerance of 0.01. In this case, all layers employ the sigmoid activation function with bias. Since the number of trained networks is relatively high, only the two best and worst configurations, with regards to their accuracy, are presented in this paper. For each setup, three sets of curves are analyzed: the training curve of average error energy (ξ_{av}), the progression of Euclidean norms of the local gradients per layer, and the Receiver Operating Characteristic (ROC) curve.

The average error energy, ξ_{av}, is the cost function of the MLP optimization problem. Indeed, such a figure is the network's error measured in a particular epoch. In essence, it is computed by averaging the instantaneous energy errors, $\mathscr{E}(n)$, for all the training instances [15]. By the way, the latter are proxies of the neurons' lack of effectiveness. Adjectitiously, the $\mathscr{E}(n)$ values can be determined as presented in Equation 7, where e_j is the error of the j-th neuron in the output layer L.

$$\mathscr{E}(n) = \frac{1}{2}\sum_{j\in L}^{N} e_j^2(n) \tag{7}$$

The progression of the average Euclidean norms of the local gradients per layer constitutes a simplified representation of the inner workings of the Backpropagation algorithm. Such a depiction is consistent with the algorithm's operation since the norm of the local gradients converges to zero as the gradients move closer to the same spot. Keep in mind that gradients' convergence is a necessary condition for optimality; therefore, one might expect that high-performant learners exhibit the asserted behavior.

The ROC curve is a popular visualization tool that illustrates the diagnostic capacity of a binary classifier as its discrimination outset is varied. Specifically, such a graphical representation is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. On the other hand, the Area Under the Curve (AUC) is an indicator of the ability of a classifier to discriminate between classes and is used as a summary of the ROC curve. Explicitly, as AUC approaches 1, the classifier converges to being perfectly able to distinguish between all the positive and negative instances.

G. Performance Metrics

By means of comparing the classifiers' performance, the models are graded according to the following metrics: accuracy, sensitivity, and specificity. In fact, the sensitivity is the proportion of actual positive cases that got predict as positive, while the specificity is the proportion of actual negatives that got predicted as negatives. Having that in mind, the figures above are computed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{Specificity} = \frac{TN}{TN + TP} \tag{10}$$

where, TP stands for true positives, FN for false negatives, TN for true negatives, and FP for false positives. As a matter of fact, the designated components are taken from the classifiers' confusion matrix.

IV. Results

This section presents the results of the proposed workflow in the *Voice Gender* dataset.

A. Visualization

The fundamental question is, Do the acoustic properties of a voice sample provide enough information to develop a satisfactory gender classifier? Surprisingly, Figure 1 sheds some light into this topic. For instance, without performing any intricate modeling task, one can easily recognize that projecting the fundamental frequencies and the mean frequencies of the available samples in a 2-dimensional plane enables a rough distinction between males and females. However, only a small set of features within the dataset beget the same response. Remarkably, evidence shows that developing gender classifiers might actually be possible.

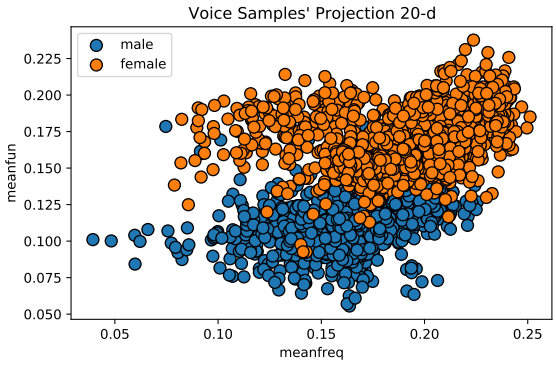


Figure 1: Voice samples' projection for 20-d

Still, the application of the embedding technique affords a different story. Namely, Figure 2 showcases how t-SNE made the samples even harder to set apart. Consequently, one might expect to obtain substandard results in such a scenario. It is well-known that the embedding procedure is beneficial for early visualization geared at understanding the degree of data separability. Nonetheless, consider not entirely relying on t-SNE to appraise how easy data will be classifiable. Out of the box, that method holds a few hyperparameters, the main one being perplexity, which might require further examination.

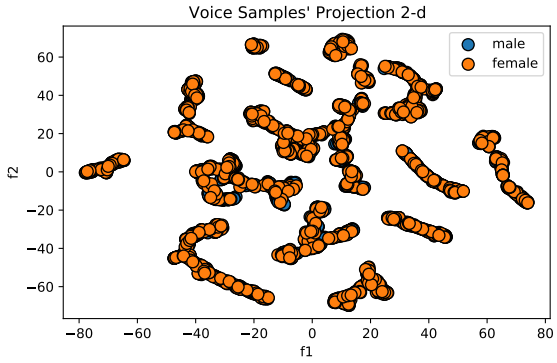


Figure 2: Voice samples' projection for 2-d

B. Probably Approximately Correct Learning

Recall the main goal is to develop a high-performant learner that affords a reasonable true error bound. In a previous section, it was discussed that given a fixed pair of error bounds (δ, ϵ), one could compute the minimum sample size to employ in order the assure with probability greater than $1 - \delta$ that the gap between the training and true error is bounded by ϵ. Thus, Tables I and II display the optimal sample size to use, in both spaces, for each learner under three reasonable setups.

Table I: Optimal samples sizes for 20-d

Classifier	δ	ϵ	N^*
Decision Tree $k = 8$	0.020	0.050	188814
	0.070	0.100	47160
	0.100	0.150	20955
Linear SVM	0.020	0.050	18684
	0.070	0.100	8178
	0.100	0.150	4998
Polynomial SVM $r = 3$	0.020	0.050	1575675
	0.070	0.100	689633
	0.100	0.150	421458

For all intents and purposes, the authors only possess 1900 instances for the training set, as this is 60% of the available voice samples. Unexpectedly, as revealed in Table I, not even one learner in the 20-d space concurs with the concrete circumstances. Accordingly, one has to accept that the true error might be bounded even higher than anticipated. In particular, as Figure 3 shows, 1900 instances could be used to furnish a Decision Tree of depth 8 with a true error bounded over 0.50 with a 0.70 probability. Similarly, the same number of samples are able to come up with a Linear SVM bounded over 0.32 with a 0.88 probability. In contrast, the 3-degree Polynomial SVM is significantly distant from a reasonable true error bound with a substantial likelihood.

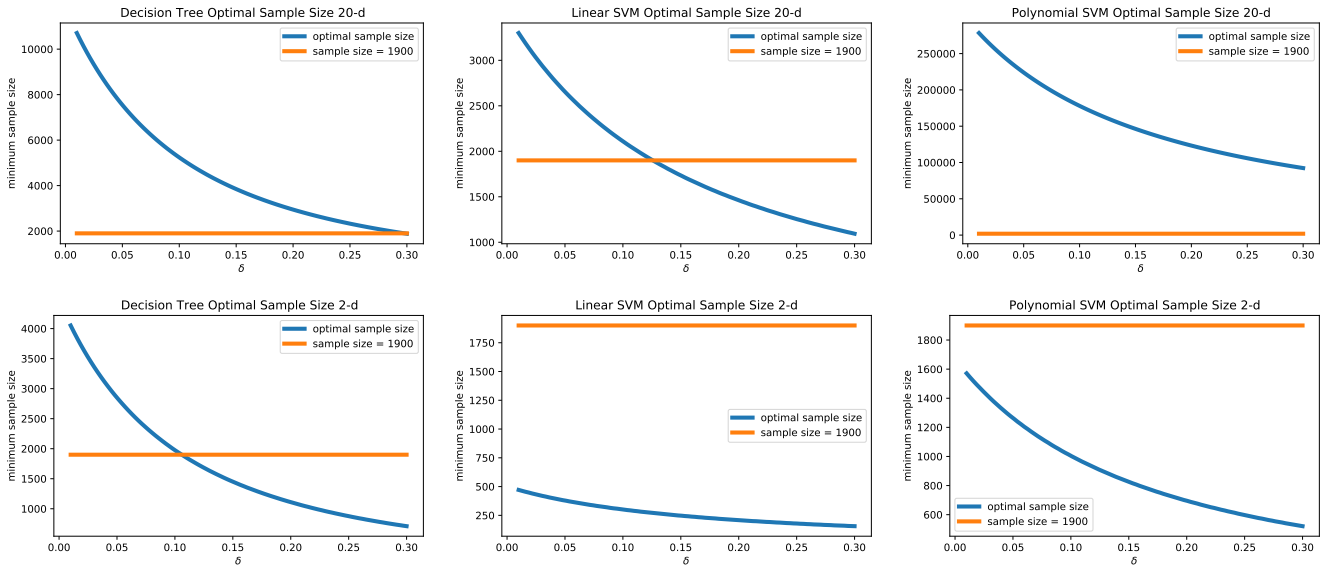


Figure 3: Optimal sample size comparison for 20-d and 2-d

As it happens, evidence shows that the Polynomial SVM demands a more extensive training set than the other two examined learning machines; leaving the Linear SVM as the one with the fewest requirements. Despite that, it is essential to clarify that the PAC conditions do not guarantee the dataset will be effortlessly learnable. Now, turning to the embedded space, Table II conveys that more modest sample sizes are required if the dimensions are reduced. Expressly, a Linear SVM with a true error bounded over 0.10 could be attainable with a 0.93 probability, which is actually quite exceptional.

Table II: Optimal samples sizes for 2-d

Classifier	δ	ϵ	N^*
Decision Tree $k = 8$	0.020	0.050	71382
	0.070	0.100	17802
	0.100	0.150	7907
Linear SVM	0.020	0.050	2669
	0.070	0.100	1168
	0.100	0.150	713
Polynomial SVM $r = 3$	0.020	0.050	8897
	0.070	0.100	3894
	0.100	0.150	2379

The PAC analysis showed that the number of training instances available for this study is far below the optimal sample size for each examined learner in the original 20-dimensional space. So, one might be tempted to train each machine and assert the test set's metrics are fair proxies for the generalization error. Although, at this point, you must notice that such reasoning is flawed under the convergence theorem. Ergo, even if the estimated accuracy of each classifier is ideal, that does not mean such models can perform well under a new set of previously unknown voice samples.

C. Decision Trees and Support-Vector Machines

Table III presents the quality metrics for the Decision Tree and the Support-Vector Machines in the 20-dimensional space. What stands out in the table is that all the models performed superbly, yielding accuracy, sensitivity, and specificity figures over the 0.97 mark. Closer inspection of the records reveals that the Decision Tree holds non-dominated accuracy values in the validation and test sets, making this particular learner a strong candidate for the title of the best learner in the original space. On the flip side, the Linear SVM exhibits comparably the worst performance. Though, by no means the classifier above is ill-suited since it keeps top grades.

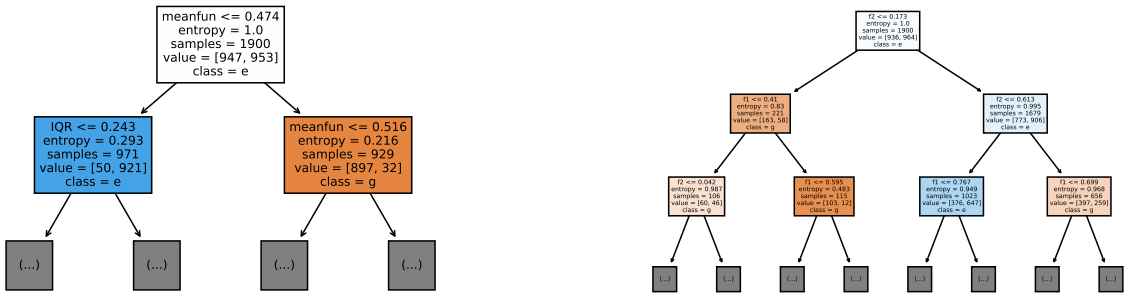
Table III: Classifiers' metrics in 20-d

Classifier	Set	Accuracy	Sensitivity	Specificity
Decision Tree	Val.	0.985	0.984	0.986
	Test	0.987	0.991	0.983
Linear SVM	Val.	0.975	0.979	0.971
	Test	0.971	0.979	0.963
Poly. SVM	Val.	0.981	0.977	0.985
	Test	0.981	0.981	0.981
RBF SVM	Val.	0.979	0.982	0.975
	Test	0.977	0.982	0.972

What is striking about the figures in Table III is that a linear model performed exceptionally. Intuitively, such a fact signals that the dataset holds linear separability to a certain degree. In that fashion, one might argue that the gender classification problem could be tackled with simpler learning machines. Along these lines, one could avoid complex learners who achieve almost the same performance as a multi-dimensional linear model.

Turning the analysis to Figure 4, one notices that the feature at the top of the Decision Tree for the 20-d space is the mean fundamental frequency of the voice sample. Plainly, that means that the stated attribute boasts the largest entropy of the feature set, making it a significant factor within the classification procedure.

A completely different situation is evinced in Table IV,



(a) Decision Tree for 20-d

(b) Decision Tree for 2-d

Figure 4: Decision Trees for 20-d and 2-d

where the results for the embedded space are presented. As speculated before, the embedding procedure ended in a rather tricky feature space to split between male and female instances. As a consequence, the performance metrics in that table reveal four weak classifiers. In this case, the Support-Vector Machines' accuracy, in both sets, is so close to the 0.5 threshold that they may not be more reliable than a dummy classifier. Again, the Decision Tree takes the lead with most measurements over the 0.7 mark. Another interesting fact is that the SVMs exhibit an apparent gap between sensitivity and specificity, which exposes that such models are more likely to classify one class more precisely than the other.

The analysis above can be summarized in Figure 5, where the ROC curve of each learner is displayed. Unsurprisingly, the curves for the models in the 20-dimensional space show an almost perfect shape with AUC scores close to 0.99. In contrast, the machines in the embedded space report a poor performance with AUC ratings below the 0.80 mark. In spite of everything, evidence shows that the Decision Tree is undoubtedly the runner-up for the most competent learner in both scenarios. Regardless, recall from the PAC examination that the ascertained metrics are not reliable concerning the generalization error, since the sample size is far below the determined lower bounds. That means that although the performance in training, validation, and test seems to match the modelers' desires, an overfitting problem could be around the corner.

Table IV: Classifiers' metrics in 2-d

Classifier	Set	Accuracy	Sensitivity	Specificity
Decision Tree	Val.	0.713	0.742	0.687
	Test	0.734	0.733	0.734
Linear SVM	Val.	0.552	0.629	0.480
	Test	0.569	0.655	0.478
Poly. SVM	Val.	0.580	0.704	0.463
	Test	0.614	0.738	0.482
RBF SVM	Val.	0.645	0.595	0.693
	Test	0.658	0.591	0.729

D. Multilayer Perceptrons

As mentioned above, numerous neural networks, in a Multilayer Perceptron scheme, were built while varying the number of layers and number of neurons in each layer. For the original space, 774 networks were trained sequentially following the implementation of an MLP module. Since a myriad of performance figures cannot be adequately enfolded in a research paper, only the two best and worst setups are showcased. Namely, Table V presents the quality metrics for the top and bottom two configurations in the 20-dimensional scenario.

Table V: MLPs' metrics in 20-d

Architecture	η	Set	Accuracy	Sensitivity	Specificity
$[20, 9, 9, 10, 2]$	0.9	Val.	0.988	0.987	0.989
		Test	0.980	0.972	0.987
$[20, 10, 7, 10, 2]$	0.9	Val.	0.987	0.986	0.989
		Test	0.980	0.974	0.986
$[20, 7, 5, 8, 2]$	0.5	Val.	0.919	0.842	0.995
		Test	0.911	0.827	0.995
$[20, 7, 7, 8, 2]$	0.5	Val.	0.912	0.828	0.995
		Test	0.910	0.823	0.996

It is apparent from Table V that astonishing results are obtained with $\mathcal{L}_1 = [20, 9, 9, 10, 2]$ and $\mathcal{L}_2 = [20, 10, 7, 10, 2]$. In both cases, the accuracy, sensitivity, and specificity figures, in validation and test, are over the 0.97 mark. Strikingly, these two setups hold a large number of parameters to estimate. However, a deeper analysis revealed that in this particular scenario, the number of layers and neurons in each layer are not indicative of the network's performance. In many cases, simpler networks performed almost as well as deeper ones, as seen in Figure 8.

Another interesting aspect of the MLPs' results in the original space is that $\eta = 0.9$ seems to be, on average, the learning rate setting that renders the most favorable outcomes. In addition, one could argue that although the lousiest architectures are significantly inferior to the best ones, these are

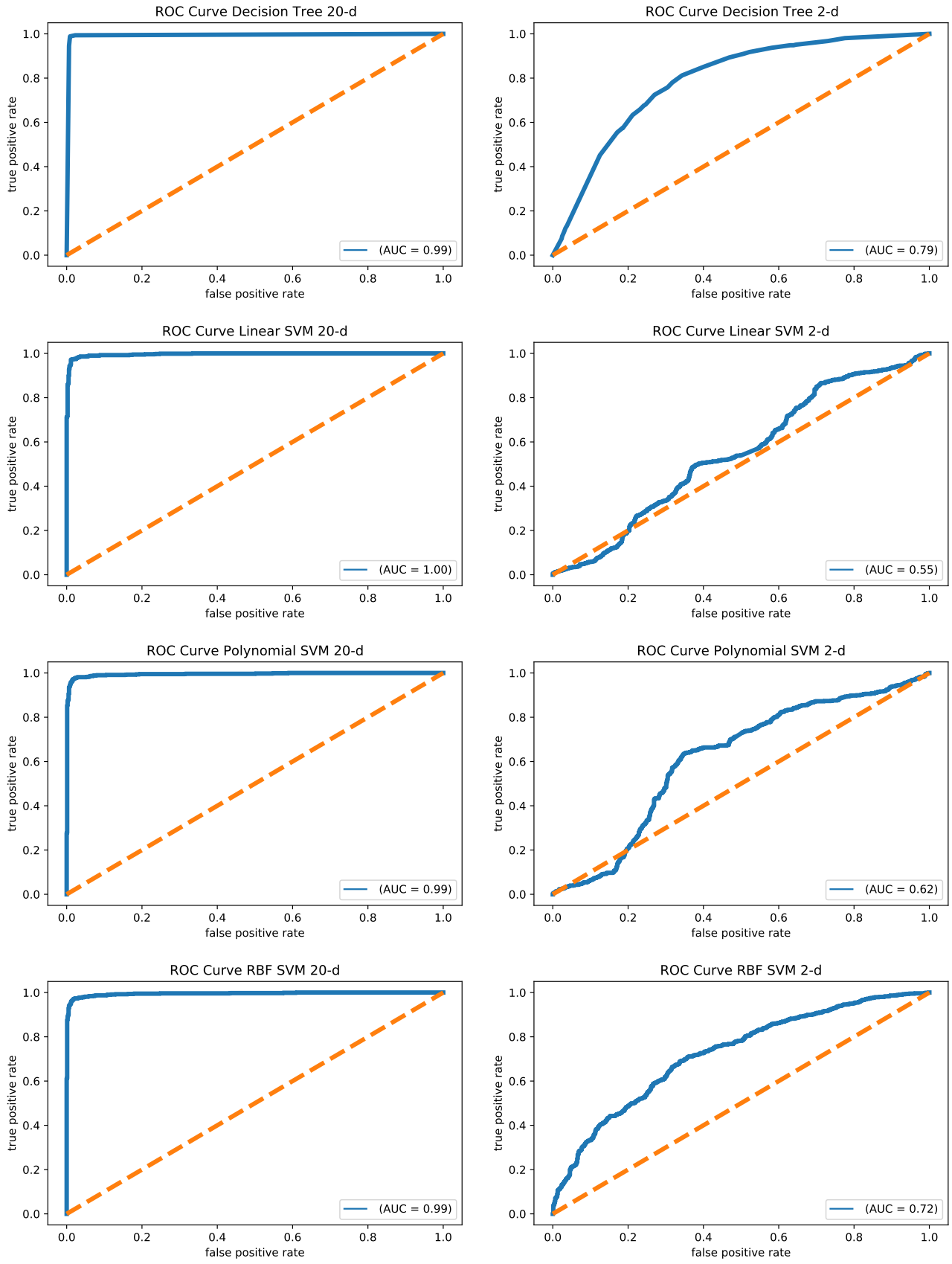


Figure 5: Classifiers' ROC curves in 20-d and 2-d

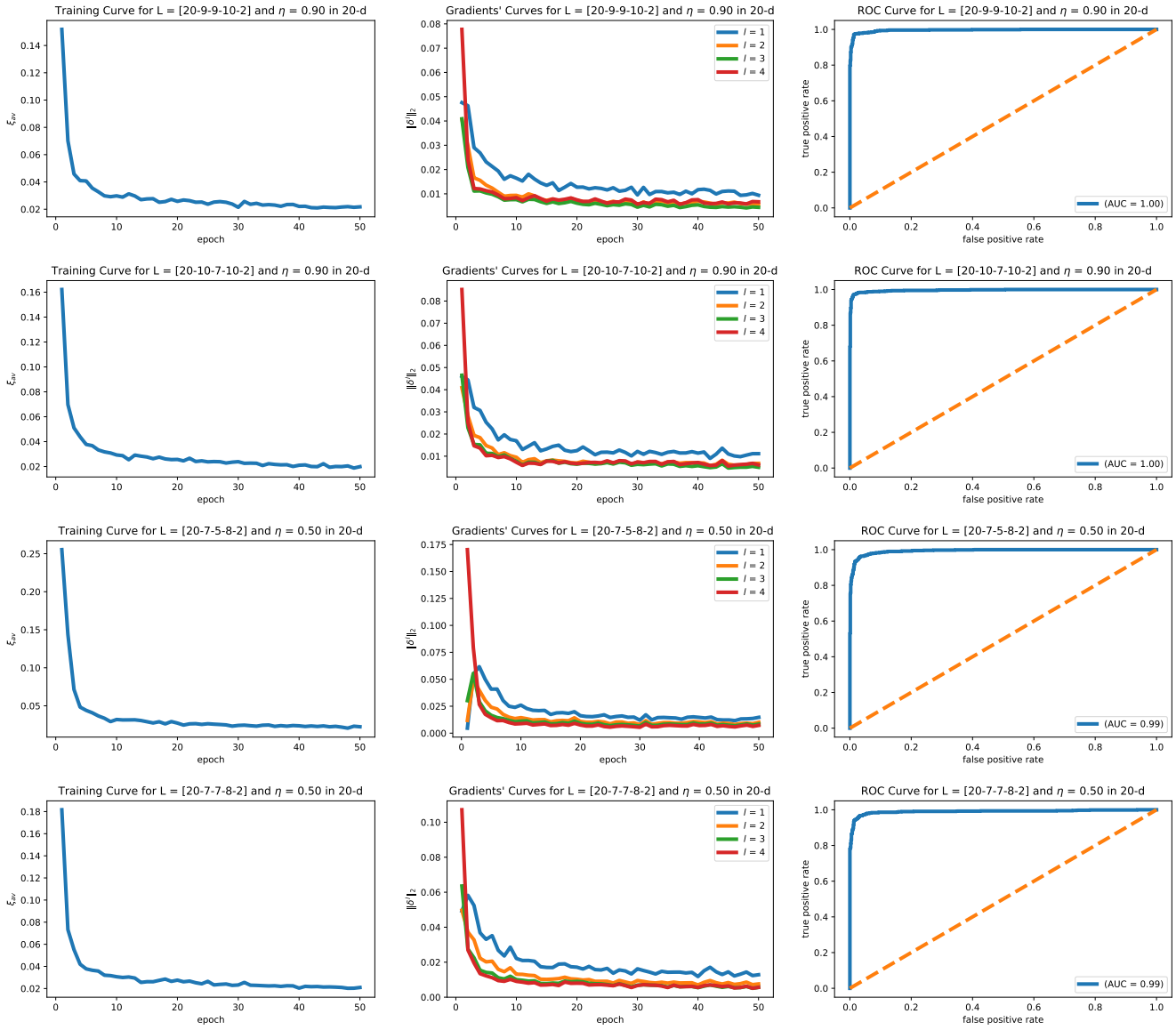


Figure 6: Two best and worst Multilayer Perceptrons in 20-d

bounded by a 0.90 accuracy, which by the way, makes them not as faulty as foreseen.

Figure 6 displays the training, gradients, and ROC curves for the examined setups in the 20-dimensional space. As anticipated, considering the satisfactory outcomes in Table V, the curves reflect the inner workings and the effects of successful Backpropagation. For instance, all the training curves show how the average error energy, ξ_{av}, reduces appreciably in the first ten epochs and progressively decreases to reach an average network error close to 0.02. Besides, the gradients' curves per layer bring to view how the local gradients converge from one iteration to the next. Furthermore, the ROC curves confirm that, on average, the trained networks perform adequately in the test instances.

For the embedded space, 117 networks were trained in a similar fashion. As predicted, the MLPs' metrics in the 2-dimensional space, unveiled in Table VI, leave much to be desired. In this case, the two best setups render accuracy values, both in validation and test, below the 0.65 mark. While the two foulest ones yield null specificity figures, signaling that no single woman was classified correctly. What is conspicuous

Table VI: MLPs' metrics in 2-d

Architecture	η	Set	Accuracy	Sensitivity	Specificity
$[2, 3, 2]$	0.9	Val.	0.625	0.561	0.688
		Test	0.642	0.589	0.696
$[2, 2, 2]$	0.9	Val.	0.622	0.618	0.627
		Test	0.635	0.623	0.648
$[2, 1, 2, 1, 2]$	0.5	Val.	0.494	1.000	0.000
		Test	0.494	1.000	0.000
$[2, 3, 3, 3, 2]$	0.9	Val.	0.494	1.000	0.000
		Test	0.494	1.000	0.000

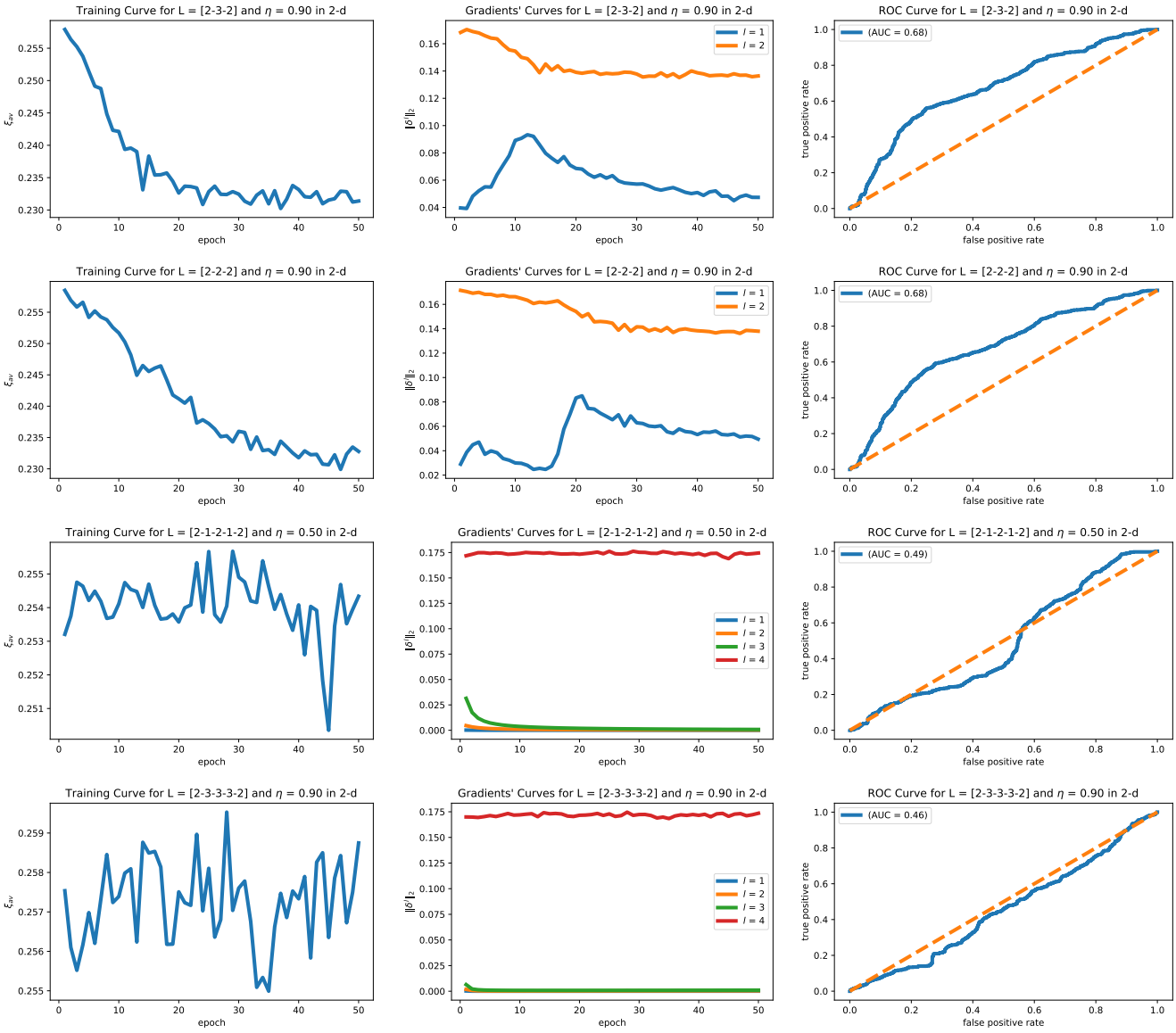


Figure 7: Two best and worst Multilayer Perceptrons in 2-d

about this scenario is that the complete result set disclosed that the simplest architectures perform, on average, better than the more intricate ones.

Figure 7 displays the training, gradients, and ROC curves for the examined setups in the 2-dimensional space. It is safe to say that such plots showcase what happens when the Backpropagation algorithm is unable to work properly. As it happens, the training curves exhibit insignificant reductions or even erratic behavior in the average error energy. Similarly, the local gradients are incapable of verging and appear to wabble close to their initial positions. Moreover, the ROC curves expose a substandard performance in the test instances.

E. Model Comparison

Interestingly enough, previous results show that building high-performant gender classifiers is plausible. As disclosed above, the original space keeps a set of features that makes it unusually effortless to learn to distinguish male and female voice samples. On that path, if one selects the finest classifier for that space based on dominance relations, then we have a tie between the Decision Tree and the $\mathcal{L}_1 = [20, 9, 9, 10, 2]$ with $\eta = 0.9$ MLP setup, as neither one is strictly superior to the other while analyzing the accuracy figures in the validation and test sets.

As manifested above, the MLPs in the 20-dimensional space seem to converge too fast. Those results indicate that regularly, no more than 20 epochs are needed to obtain a decently-trained learner. As suspected, a momentum heuristic is required to counterbalance the speedy error loss and the gradients' vanishment. Notwithstanding, if a client commands to pick just one model, the Linear SVM would be supplied, as

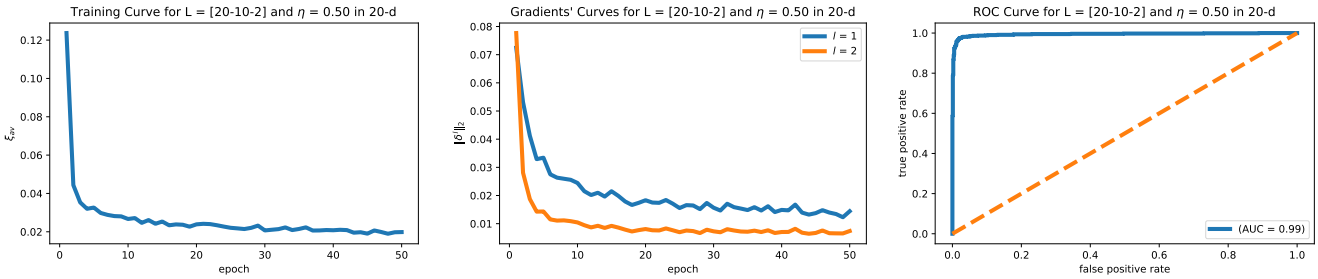


Figure 8: Results for a Simple Three-layer Multilayer Perceptron in 20-d

this learning machine procures stellar effects with the fewest data requirements.

On the other hand, while examining the embedded space results, it is evident that the Decision Tree blazes a trail, with accuracy figures over the 0.70 grade. However, bear in mind that not a single classifier performed decently in the 2-dimensional space. Thus, you should ask yourself whether embedding your features with t-SNE is a good or bad idea in your specific scenario.

Although the preceding assessment is based on the collected results, a more in-depth analysis is required to develop robust estimates of each learner's performance. In particular, a cross-validation procedure should be effected to derive statistically significant metrics that characterize each models' ability to fulfill the classification task.

V. Conclusions

This study set out to assess the performance of five different learning machines in gender recognition by voice and speech analysis. Unquestionably, the dataset possesses a rich set of features to build proper classifiers. This study has identified that procuring a high-performant learner is achievable. In particular, the collected results reveal that rendering classifiers with accuracy ratings above the 0.98 mark is relatively straightforward, as no intricate calibration procedure was necessary to achieve such magnificent outcomes.

As mentioned above, in the high-dimensional space, the Decision Tree and the $\mathcal{L}_1 = [20, 9, 9, 10, 2]$ with $\eta = 0.9$ MLP setup tied in the first place as the most accurate learners. In that respect, several exciting facts were revealed. For instance, the mean of the fundamental frequency of the voice sample seems to be the most indicative attribute for a tree-based classification. Besides, after a thorough examination of the NNs' results, there is no sufficient evidence to point out that the complexity of the architecture has a strong relationship with the obtained metrics in that scenario, as many simple ones were up to par with further intricate ones. Alongside, the outstanding results in 20-d are backed up by the training curves that showcase a monotonically decreasing average error energy and the progressive confluence of local gradients.

Contrastingly, the models in the embedded space show no promising outcomes, as neither one of them performed decently. The main reason being that the t-SNE procedure procured a highly unlearnable feature space. Still, the Decision Tree was able to stand out from the crowd with accuracy figures over the 0.70 grade. Evidence suggests that applying the Barnes Hut t-SNE furnished no meaningful benefits within this study. Equivalent to that, the related training curves flaunt the behavior of faulty Backpropagation, whereas the average error energy is unable to plummet and the local gradients keep floundering around.

The conclusions above are made with extreme caution, given that the sample size is rather small and relatively distant from the optimal sample sizes computed for Decision Trees and Support-Vector Machines. Subsequently, a more considerable amount of training instances are needed to comply with PAC's conditions, in view of the fact that overfitting could be in the offing. As for future work, the authors must work on increasing the sample size to obtain a rational generalization error's upper bound, and also on applying cross-validation to derive sharper metrics of each learner's performance.

References

[1] S. Watson, "The unheard female voice: Women are more likely to be talked over and unheeded. but slps can help them speak up and be heard.," 2019.

[2] P. Gupta, S. Goel, and A. Purwar, "A stacked technique for gender recognition through voice," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, IEEE, 2018, pp. 1–3.

[3] M. Demirkus, K. Garg, and S. Guler, "Automated person categorization for video surveillance using soft biometrics," in *Biometric Technology for Human Identification VII*, International Society for Optics and Photonics, vol. 7667, 2010, 76670P.

[4] M. Paluchamy, D. Suganya, and S. Ellammal, "Human gait based gender classification using various transformation techniques," *IJRCCT*, vol. 2, no. 11, pp. 1315–1321, 2013.

[5] F. Ertam, "An effective gender recognition approach using voice data via deeper lstm networks," *Applied Acoustics*, vol. 156, pp. 351–358, 2019.

[6] S. Lakra, J. Singh, and A. K. Singh, "Automated pitch-based gender recognition using an adaptive neuro-fuzzy inference system," in *2013 International Conference on*

Intelligent Systems and Signal Processing (ISSP), IEEE, 2013, pp. 82–86.

[7] M. Buyukyilmaz and A. O. Cibikdiken, "Voice gender recognition using deep learning," in *2016 International Conference on Modeling, Simulation and Optimization Technologies and Applications (MSOTA2016)*, Atlantis Press, 2016, pp. 409–411.

[8] K. Becker, *Identifying the gender of a voice using machine learning*, https : / / www . kaggle . com / primaryobjects/voicegender, 2016.

[9] A. B. Graf and S. Borer, "Normalization in support vector machines," in *Joint pattern recognition symposium*, Springer, 2001, pp. 277–282.

[10] J.-M. Jo, "Effectiveness of normalization pre-processing of big data to the machine learning performance," *The Journal of the Korea institute of electronic communication sciences*, vol. 14, no. 3, pp. 547–552, 2019.

[11] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[12] F. Denis, "Pac learning from positive statistical queries," in *International Conference on Algorithmic Learning Theory*, Springer, 1998, pp. 112–126.

[13] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[14] V. Vapnik, "On the uniform convergence of relative frequencies of events to their probabilities," in *Doklady Akademii Nauk USSR*, vol. 181, 1968, pp. 781–787.

[15] S. Haykin and N. Network, "A comprehensive foundation," *Neural networks*, vol. 2, no. 2004, p. 41, 2004.

APPENDIX
DECISION TREES

Figures 9 and 10 present wider shots of the trained Decision Trees for the high and low-dimensional spaces, respectively.

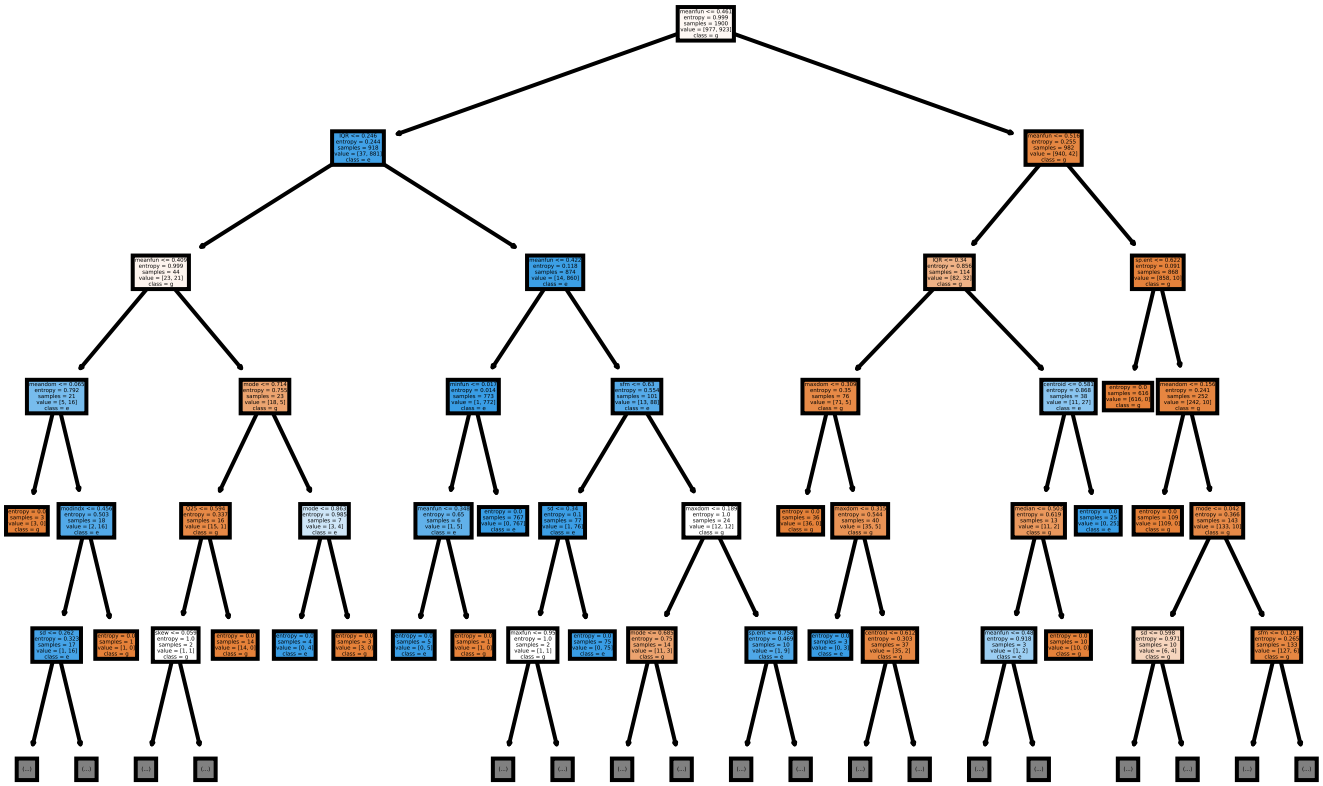


Figure 9: Decision Tree for 20-d

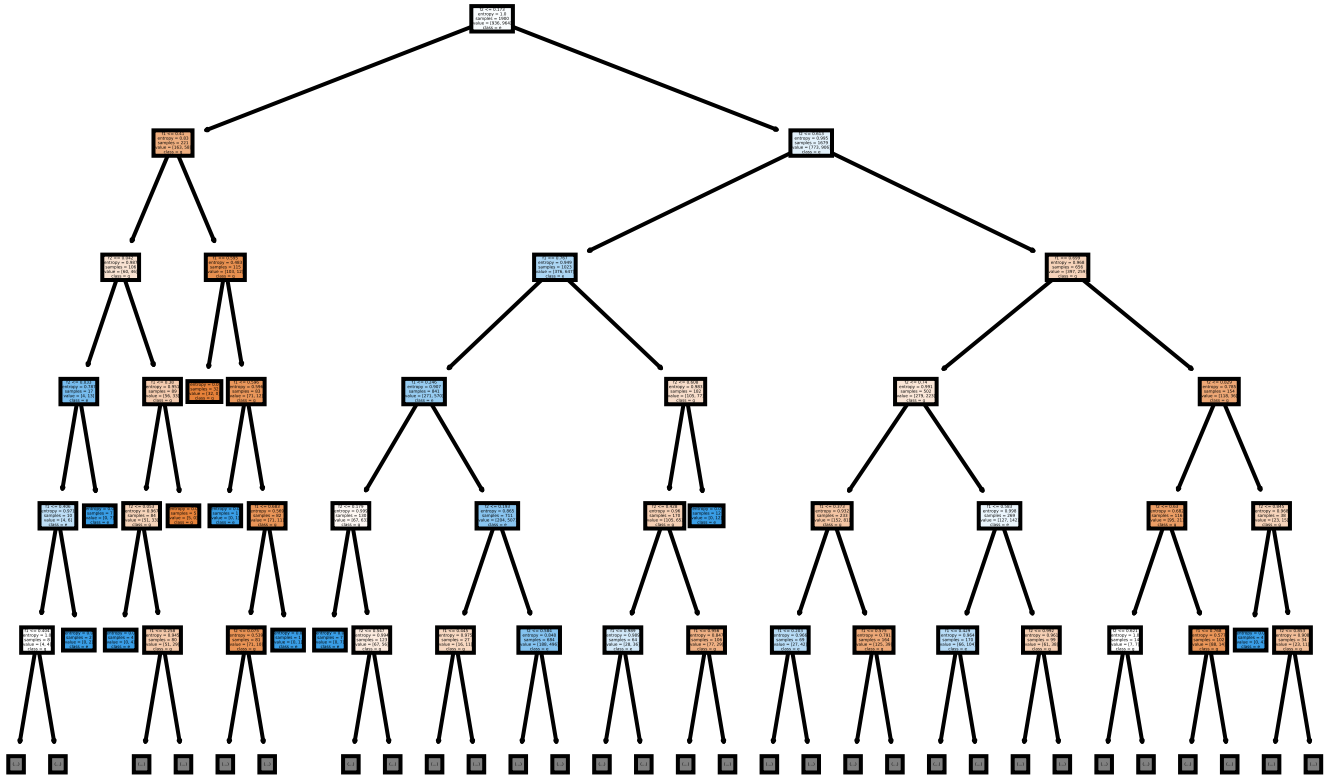


Figure 10: Decision Tree for 2-d