High-Dimensional Analysis on LSVT Voice Rehab Data

Justine Huynh jgh75@georgetown.edu | Zixun Hao zh210@georgetown.edu

ANLY 565 Adv Analytics Streaming HighD, Spring2022 | Prof. Nathaniel Strawn

Summary

Our analysis uses the Lee Silverman Voice Treatment (LSVT) voice rehabilitation dataset from the UCI archive. LSVT is a treatment to improve vocal function in patients with Parkinson's disease. There are 126 speech samples from 14 Parkinson Disease patients and 310 features. Each column represents an application of a particular speech signal detecting and processing algorithm. These voice signal algorithms include: standard perturbation analysis methods, wavelet-based features, fundamental frequency-based features, etc. (1. UCI, 2016) The response variable is whether the speech sample is considered phonetically 'acceptable' or 'unacceptable' by a clinician.

The goal of the project is to perform dimensionality reduction, model selection, and descriptive and predictive analysis on what algorithms (features) can best predict whether the patients' speech was deemed phonetically 'acceptable' (labeled 1) or 'unacceptable' (labeled 0) (2. Ramig et al, 2001).

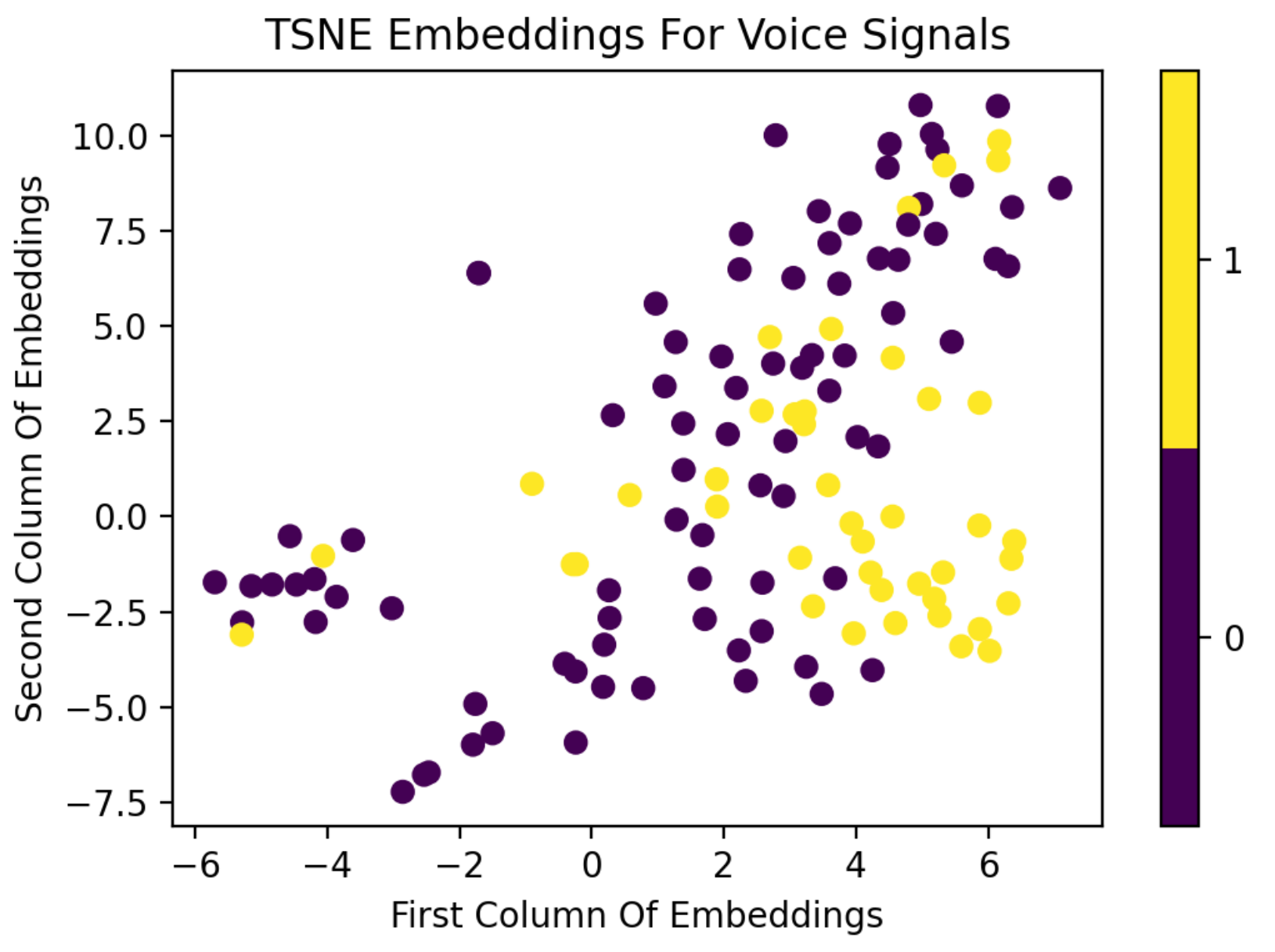
Clustering & Projecting to Low Dimensions

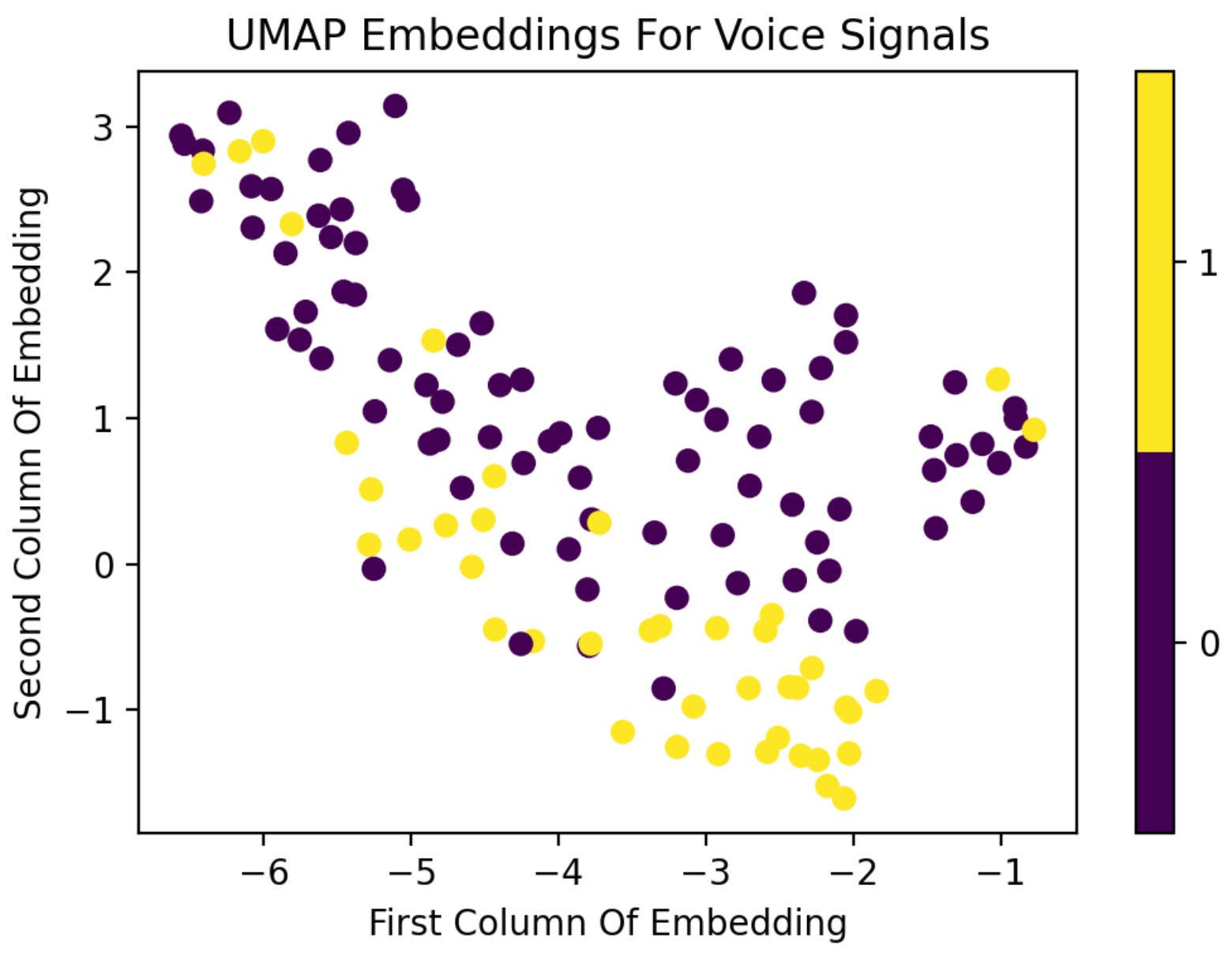
We used PCA, spectral methods (t-SNE, UMAP), and Andrews Curve to project high-dimensional data to lower dimensions. We standardized the explanatory variables and encoded the "acceptable" phonation as 1 and "unacceptable" phonation as 0. Because of the large number of features, we opted to perform dimension reduction via Principal Component Analysis (PCA). PCA is an unsupervised method to understand the underlying structure of the data. It does so by performing singular value decomposition of the design matrix, to find eigenvectors that explain the most of the data's variance. Our PCA results were mediocre, with around 59.52% of the variance explained with the first three principal components (PC 1: 32.29%, PC 2: 19.87%, PC 3: 7.36%). This shows that our dimensional data has an underlying low-dimensional structure to a certain degree. According to the scree plot, the "elbow" appears to be PC 3, adding any more principal components would have contributed to much smaller additions to the variance explained – about 6% or less with each additional principal component.

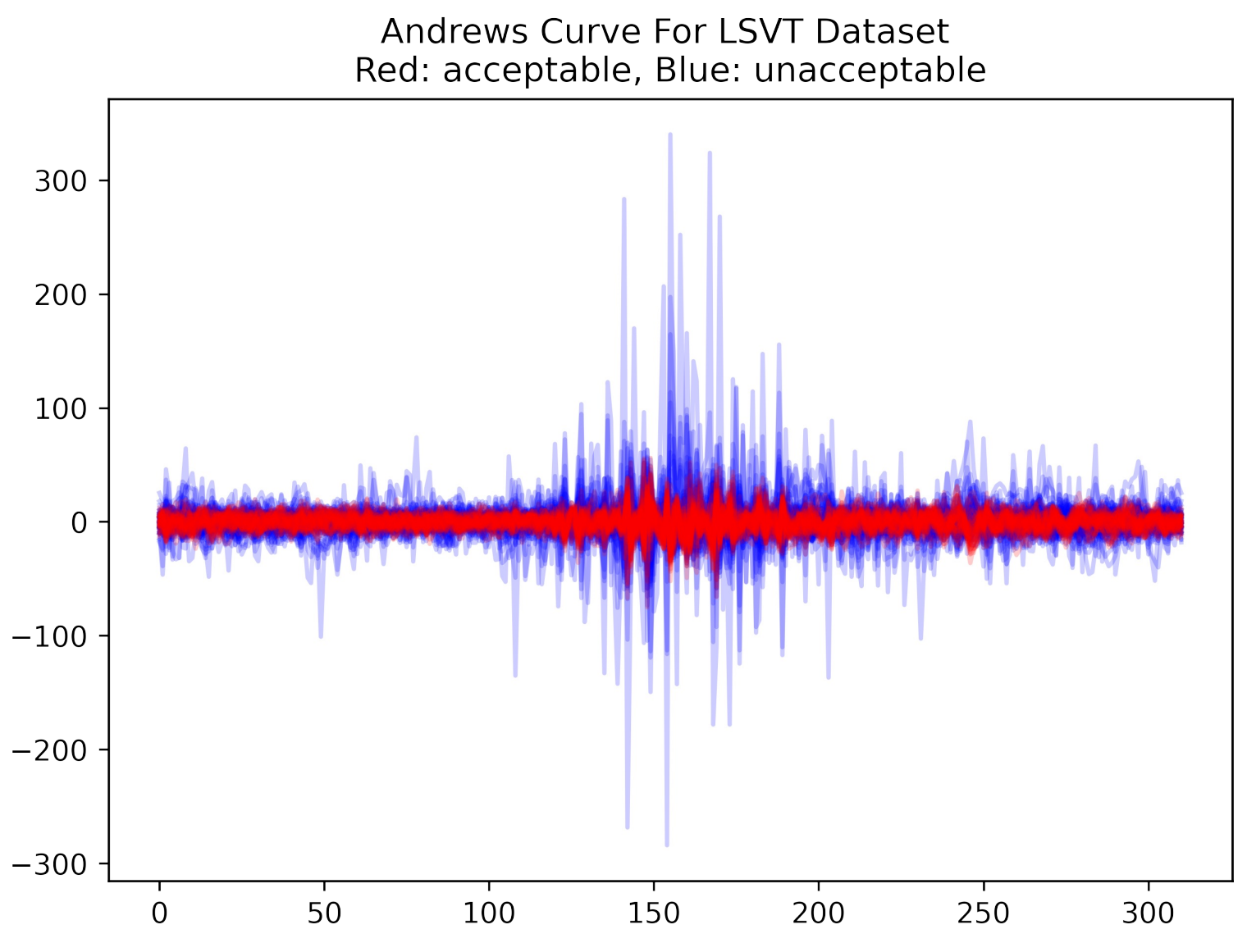
Our other method, t-SNE, is similar to PCA in that it performs dimension reduction. Unlike PCA, t-SNE does so by calculating the distance between points instead of explaining variance. T-SNE can display nonlinear relationships on the data. Both t-SNE and Uniform Manifold Approximation and Projection (UMAP) attempt to cluster the data by first initializing a low-dimensional graph of the data. T-SNE always randomly initializes the low-dimensional graph regardless of whether it performs on the same dataset. (4. Kobak, 2019) In other words, every t-SNE run will spawn a different low-dimensional graph. In contrast, UMAP does spectral embedding that keeps the initialized low-dimensional graph constant regarding a specific dataset.

We also performed dimension reduction with UMAP. UMAP is a manifold learning and nonlinear dimension reduction technique used to project data to lower dimensions. We clustered our explanatory data matrix via UMAP's similarity scores and spectral embedding to initialize low dimensional graph and to find latent subgroups. While UMAP can be used for supervised learning, it is used in an unsupervised circumstance here. It outperformed t-SNE in run time performance. Also, UMAP embeddings are deterministic (3. Becht et al, 2018); t-SNE is not because of the randomized initialization - we set random seed for reproducibility of the code.

T-SNE was able to form a small cluster of no more than 25 data points who were classified as 1, or voices that were deemed acceptable. UMAP was able to form a ragged boundary between a group of acceptable phonations and unacceptable phonations, although this ragged boundary existed for only the middle portion of the embedding graph. UMAP appears to separate the two groups marginally better than t-SNE. Because PCA had mediocre performance in explaining the total data variance and our clustering methods UMAP and t-SNE both struggled to cluster the data well, our voice data latent structure is not inherently low-dimensional.







We also used Andrews Curve to group and cluster the observations and visualize them on 2-dimensional space. Andrews curve is a method to visualize high-dimensional data and spot trends in high-dimensional signal interactions. It has the ability to accurately display statistical groupings and distances by grouping signals with similar distributions (mean, variance) together, while preserving inter-distribution distance measures, which is proportional to the Euclidean distance between corresponding points. (5. Atkins, et al 2016) Andrews curve displays multivariate data by transforming each observation into a curve, generated by a series of sine and cosine functions.

For our case, Andrews Curve is suited for clustering the voice data. Andrews curve was able to project the unacceptable and acceptable voices onto 2-dimensional spaces, with each "spike" on the graph representing a feature/dimension. Though there is no clear partitions between the two classes, we can observe that acceptable voices are clustered in the middle part with lower variability while unacceptable voices are further away from each other. Unacceptable observations have much bumpier curves with bigger variability. This projection makes it possible to spot the clustering of signals and outlier observations, which may not be directly visible in higher dimensions.

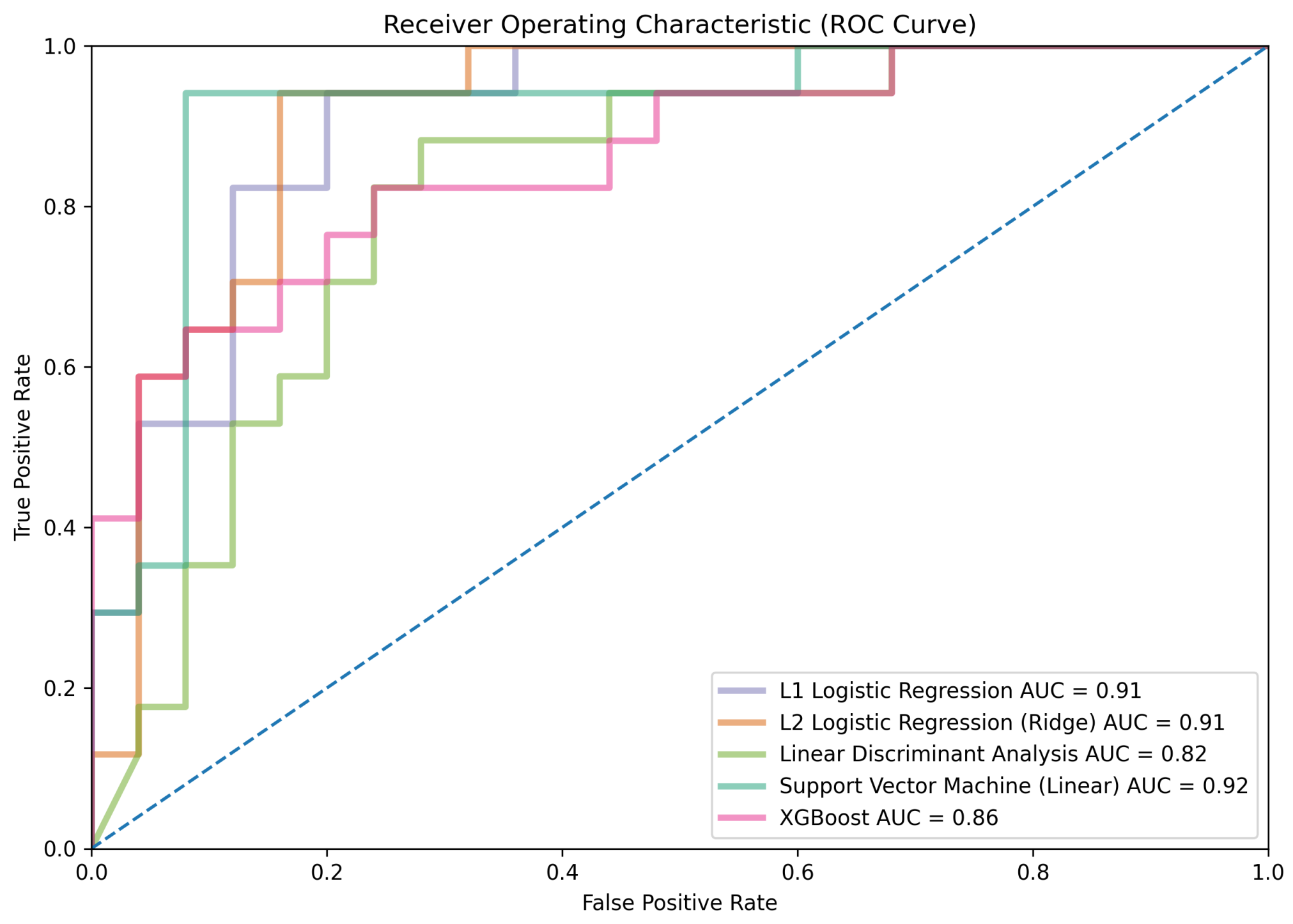
Predictive Analysis

We analyzed the dataset to identity if these features can accurately predict if a voice signal is "acceptable", according to voice therapy standards, and which features are more important for this prediction. We used Lasso (L1), Ridge (L2) logistic regression, Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), and XGBoost. Lasso regression relies on a penalized linear regression method to avoid overfitting. It uses the penalization term: L1 penalization, which is the absolute value of the coefficients' magnitude. Lasso shrinks feature coefficients to 0 to grab the most important features. When using the L1 logistic regression (Lasso) with regularization parameter C = 1 on our data, 46 out of 310 features got non-zero feature coefficients while the rest were shrunken to zero. Ridge regression performs like Lasso regression except Ridge uses a different penalization term: L2 penalization, which is the square of the coefficients' magnitude.

SVM draws boundaries among the data to separate the data into their respective classes. LDA, similar to SVM, draws boundaries among the data points. Nevertheless, LDA uses all data points while SVM focuses on points who are difficult to classify, such as data points who lie close to the boundary. We used the linear kernel for SVM to make it more analogous to the other linear models we are utilizing. XGBoost is a nonparametric, ensemble tree algorithm that can perform both regression and classification. XGBoost uses boosting, which combines many different weak decision tree learners to improve predictive accuracy. XGBoost grows many little weak regression and classification decision trees (CART), analyzes different parameters, and gradually creates a more powerful, accurate model with low misclassification error based on these analyses.

All of these methods had an area under curve (AUC) value of at least 82%, indicating decent performances on identifying true positives. Interestingly, both Lasso and Ridge algorithms had the same AUC value, though their ROC curves were not identical. Out of all these models, SVM had the highest area under the curve and best accuracy, precision and F-1 scores. After optimizing parameters with GridSearchCV, XGBoost still had the highest number of false negatives (7 false negatives; predicted unacceptable, but voice was truly acceptable) and consequentially the lowest recall score (the classifier's ability to correctly identify all the true positive cases on the data) of 0.588.

	L1	L2	LDA	SVM	XGB
Accuracy	0.786	0.833	0.762	0.857	0.810
Precision	0.786	0.778	0.706	0.867	0.909
Recall	0.647	0.824	0.706	0.765	0.588
F-1	0.710	0.800	0.706	0.813	0.714



Selective Inference

We also applied the fsInf function from the selectiveInference R package to perform selective inference for forward stepwise inference. We first fit a forward stepwise regression and then use fsInf() to compute selective p-values and confidence intervals for the regression estimates and get conditional inference. It's a sequential hypothesis problem. The result prints: "Sequential testing results with alpha = 0.100", meaning that the procedure stops when the average p value exceeds alpha, so it guarantees false discovery rate (FDR) control at level alpha.

The selective inference methods allow researchers suppress false positives when selecting models and generating inferences. The result of the procedure shows that only 5 features (feature index 84, 89, 241, 93, 57) have p-values smaller than 0.05, indicating less significant results and a loss of power in the conditional inference step than a regular forward stepwise regression. Notably, 4 out of 5 of these features got non-zero coefficients from the L1 Logistic Regression and bootstrapping steps too, which will be discussed later, indicating that these features are important when predicting the response variable. The estimated stopping point from forward stepwise is 125 using sequential stopping rules, which means the forward stepwise procedure selects 125 predictors for the results.

Cross Validation & Boostrapping

We also performed 25-time bootstrapping trials with L1 penalized logistic regression solver using majorization-minimization (MM) coordinate descent (credit to Prof. Nathaniel Strawn). Bootstrapping is the process of resampling data with replacement to simulate additional data that represents the actual population well. It is especially useful because 1) our data had only 126 rows and 2) bootstrapping is a great method to avoid underfitting. In the end, though this bootstrapping test has a relatively high training error (25.14% mean error rate in the bootstrapping process), it preserved 67 non-zero betas and "picked up" parameter index 72 (feature name: "VFER->SNR_SEO"), 41 ("Shimmer->Ampl_abs0th_perturb"), 79 ("IMF->NSR_SEO"), and 83("MFCC_0th coef") in 10 bootstrapping trials or more. These parameters appear to be the most important features in predicting the response. By comparison, the top 10 most picked up parameters in the bootstrapping process all appear to have non-zero coefficients in the L1 logistic regression that we performed, showing the consistency of in features' importance. This bootstrapping test with L1 logistic regression is an effective regularization and feature selection tool that identifies the most useful parameters.

10-fold cross validation is applied to each of the aforementioned methods to generate an accuracy score for each fold. We also performed cross validation with LassoCV from sklearn whose results/predictions are continuous. We converted the continuous results to discrete ones with the threshold 0.5. The predictions on the test data have high false negative rate and low recall score, indicating that the model is hyperactive in labeling a patient as having "unacceptable" phonation because of the class imbalance.

Conclusion & Discussion

The results from dimensionality reduction with PCA, UMAP, t-SNE, and Andrews Curve show that the data is not intrinsically low dimensional, though we can observe some partitions between the two response variable classes. Among the 5 models, SVM, Lasso, and Ridge performed well in terms of ROC-AUC. SVM also outperformed the other models in accuracy, precision and F-1 scores. The statistically significant results from the forward stepwise selective inference are consistent with the results from L1-MM logistic regression bootstrapping.

The limitation of the current project is our lack of knowledge of the specific voice detecting algorithms included in the features and how they interact. To further study the LVST data, researchers may want to improve the predictive results, and study the effect of the LVST treatment on the phonation performances of the patients by including the treatment-control variable.

References:

1. UCI Machine Learning Repository: LSVT Voice Rehabilitation Data Set. archiveicsuciedu. [accessed 2022 May 1]. https://archive.ics.uci.edu/ml/datasets/LSVT+Voice+Rehabilitation.
2. Ramig L.O, Sapir S, Countryman S, *et al.* Intensive voice treatment (LSVT®) for patients with Parkinson's disease: a 2 year follow up. *Journal of Neurology, Neurosurgery & Psychiatry* 2001;**71:**493-498.
3. Becht, E., McInnes, L., Healy, J. *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nat Biotechnol* **37,** 38–44 (2019). https://doi-org.proxy.library.georgetown.edu/10.1038/nbt.4314
4. Kobak D, Berens P. 2019. The art of using t-SNE for single-cell transcriptomics. Nature Communications. 10(1). doi:10.1038/s41467-019-13056-x.
5. Atkins J, Sharma DP. 2015. Visualization of Babble–Speech Interactions Using Andrews Curves. Circuits, Systems, and Signal Processing. 35(4):1313–1331. doi:10.1007/s00034-015-0123-4.