Applications of Supervised Learning to Parkinson's Disease Discrimination from Vocal Dysphonia Measurements

Michael Walton
Department of Computer Science, Georgia Institute of Technology

September 20, 2015

Abstract

In this work we explore the application of supervised Machine Learning algorithms to Parkinson's Disease (PD) speech pathology datasets. Our objectives include the development of models capable of discriminating unseen Parkinsonian voice samples from normative controls and the classification of known PD+ samples into one of five distinct stages of disease progression. Generalization performance is estimated using both cross-validation and a disjoint test set. We conclude with an analysis of the suitability of each algorithm as a potentially clinically significant diagnostic tool.

1 Introduction

Parkinson's disease (PD) is a neurodegeneritive central nervous system disorder characterized by progressive loss in motor control, speech and cognitive function. Common symptoms include tremor, difficulty speaking (dysphonia) and dementia. Many treatments for PD are effective only in the early motor stages of the disease which makes early detection crucial for improving patients' quality of life and long-term prognosis.

Today, the majority of PD cases are diagnosed using the Unified Parkinson's Disease Rating Scale (UPDRS). This behavioral test panel assigns a numeric value which encodes the progression of the disease. These observational tests must be administered by a qualified neurologist making PD diagnosis a relatively time consumptive and costly process. The desire for a rapid, low-cost pre-screening diagnostic has motivated a number of studies which attempt to distinguish early and difficult to detect PD symptoms such as dysphonia quantified by subtle aperiodic fluctuations in voice recordings. Although a thorough description of these metrics is beyond the scope of this study,

it suffices to understand that clinical speech pathology literature has identified vocal acoustic irregularities such as turbulent non-Gaussian noise in early stage PD patients. Determining if variation in these phonations will yield a robust PD diagnostic remains a subject of active research.

The supervised learning algorithms considered in the present study are k-Nearest Neighbor, Support Vector Machines, Artificial Neural Networks, Decision Trees and Boosting. The python modules `scipy, numpy,` and `scikit-learn` [2] [3] are used extensively throughout our implementation; the GPU-accelerated math compiler Theano [1] is used in conjunction with convenience wrappers `lasagne` and `nolearn` for neural networks.

2 Methods

In the sections that follow, we will provide a brief overview of the dysphonia datasets used in this study and the assumptions made in working with these data. In addition, we will briefly discuss the implementation details for the considered algorithms and highlight any model-specific data transformations or optimization procedures.

2.1 Data

The two datasets considered in this study originate from Parkinson's Disease clinical diagnostic literature from a research domain interested in correlating vocal dysphonia metrics with the disease. In one study, Sakar et al.[4] collected voice recordings from 20 persons with Parkinson's (PWP) and 20 otherwise healthy adults. Multiple types of sound recordings were collected from each participant (including sustained vowels, words, numbers and short sentences). The raw audio samples were then processed into a set of 26 features using various linear and time-frequency based metrics such as jitter, shimmer and pitch. The input data is $X \in \mathbb{R}^{n \times m}$ where n is the number of audio samples ($n = 1040$) and m is the dimensionality of the feature space ($m = 26$); target labels are represented by a binary vector $\boldsymbol{y} \in \mathbb{R}^n$. We therefore define our instance space as the 26 dimensional space of vocal dysphonia features sampled in [4]. Our objective is to derive a hypothesis $h \in H, h : x \in \mathbb{R}^m \mapsto y \in \{+, -\}$ approximating the target concept $c(x)$. We define the preferred hypothesis as the h which minimizes $error_{\mathbb{D}}(h)$ on the actual distribution $\mathbb{D}$ by learning on the training set $S \subset D$ where D is the available dataset of (instance, label) pairs $\{(x_i, y_i)\}$ The objective is some tbd. 'best' function which partitions the instance space by mapping each example $x \in X$ to a corresponding binary classification label $y \in \boldsymbol{y}$ denoting Parkinson's Disease (PD+) or cognitively normative (PD-).

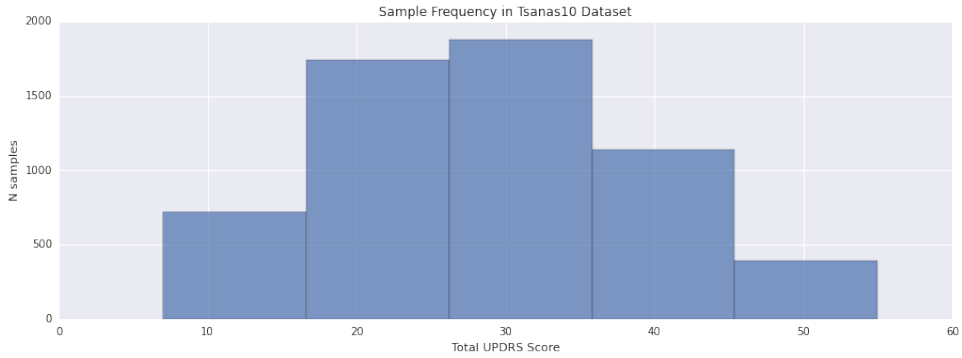


Figure 1: Tsanas Dataset: Class label Distribution

The second dataset, collected by Tsanas et al.[5] contains similar dysphonia features to Sakar, however instances are associated with continuous, real-valued UPDRS labels. These data may be readily modeled as a regression problem; in order to render the data suitable for classification, we generate proxy labels from the original distribution of UPDRS scores which represent the continuous scale as discrete 'stages' (the number of stages may be arbitrary, 5 have been selected so as to maintain consistency with the Parkinson's literature). In the associated study, researchers gathered a total of 5,875 voice recordings from 42 PWP. The vocal dysphonia metrics used are largely overlapping with [4], however fewer metrics are considered. We define our target hypothesis as a function $h : x \in \mathbb{R}^m \mapsto y \in [1, 5]$ minimizing $error_{\mathbb{D}}(h)$ where $x, y, m, \mathbb{D}$ are defined similarly to the previous study ($m = 16$).

Although the two datasets are very similar in their methods of data collection and overall approach, there are several key differences worth bearing in mind. The shape of these data is interesting in that: the Sakar study possesses less samples and a higher dimensional instance space when compared to Tsanas; it is possible that insufficient data and the 'curse of dimensionality' play a significant factor in the generalization error of algorithms applied to Sakar. However, the sample distribution over classes in Sakar is perfectly uniform (20 in both PD+ and PD- classes), as opposed to the roughly gaussian distribution of UPDRS scores in Tsanas leading to imbalanced classes after partitioning the samples.

Taking these observations further, given the low occurrence of Parkinsons, the real-world performance of a model trained on uniformally distributed samples will be far lower than the estimated generalization error during training and validation. This is due to the fact that the a-priori distribution is only artificially uniform, when in fact the true probability of a randomly selected individual having Parkinson's $P_{y \sim \mathbb{D}}(y = PD+) = 0.00013$. In the sample distribution $\mathbb{S}$ used in[4]

$P_{y\sim\mathbb{S}}(y = PD+) = 0.5$. In the Tsanas study, however, participants are assumed to be PD+ and the learning task is simply the prediction of the progression of the disease from voice samples. In this context, it is much more likely that the prior distribution of the collected samples ($n = 5,875$) more closely match the true distribution of PWP.

2.2 Training

To derive maximum likelihood hypotheses given the aforementioned datasets, we apply the following preprocessing and training procedures which are roughly the same for all algorithms used:

- Separate the raw-data feature columns X from the label column $\boldsymbol{y}$
- Randomly shuffle and split the data into distinct training and testing sets (1/3 held out for testing)
- Scale the data to zero mean and unit variance using `StandardScaler` (the transformation is computed only from the training set, then applied to all data)
- To reduce the chance of overfitting, train each algorithm using stratified k-folds cross validation. In this method, the algorithm is iteratively evaluated on one of each of the k folds, after using the other $k-1$ folds for training. ($k = 5$ in Sakar experiments, $k = 3$ for Tsanas)[1]

For several algorithms we also applied an cross-validated parameter search over a finite, discrete range using `GridSearchCV`. In this method, every combination of the specified parameter range is iteratively trained and evaluated using the cross-validation method previously described. The optimal parameter set is determined by the model that performs best over the average of associated cross-validation runs.

2.3 Algorithms and Evaluation

In general, the performance measure we will assume is the F1 score. This metric defines accuracy as the harmonic mean of precision[2] and recall[3]. In the case of binary classification, given precision p,

[1] A smaller k was used on Tsanas data to ensure that each fold could sufficiently preserve the label distribution of the complete training set

[2] ratio of true positives to all predicted positives

[3] ratio of true positives to all actual positives

recall r and true positive, false positive and false negative rates denoted tp, fp and fn respectively; F1 is defined as

$$F1 = 2\frac{pr}{p+r} \text{ where } p = \frac{tp}{tp+fp},\ r = \frac{tp}{tp+fn} \tag{1}$$

This definition of accuracy was chosen because an algorithm that optimizes f1 simultaneously maximizes precision and recall; that is, the learners' ability to correctly label the data is maximized while the occurrence of false positives is minimized. In the case of multi-class classification, as we have defined for the Tsanas dataset, individual scores are computed for each class using a one-vs-rest method and the total F1 score is defined by the average of the individual scores weighted by the frequency of true instances of each label.

Unless otherwise specified, we have attempted to keep the learning algorithm implementation as consistent with the course material as possible; though some liberties have been taken with certain modular components that are particular to our application or the `scikit-learn` library. In the implementation of k-NN, we used the Minkowski distance defined over arbitrary dimensional vector spaces which is the generalization of both Euclidian and Manhattan distance. Specifically, given points $X = (x_1, x_2, ...x_n),\ Y = (y_1, y_2, ...y_n) \in \mathbb{R}^n$

$$\delta_m(X, Y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}} \tag{2}$$

We chose to use the Minkowski distance measure to allow straightforward manipulation of the power parameter p[4]. In the implementation of decision trees and the boosted version using the AdaBoost algorithm, the attribute selection criterion normally defined by entropy and corresponding information gain measures was replaced with the GINI impurity[5]; given data D divided into n subsets d_i by an attribute A we define:

$$gini_A(D) = \sum_{i=1}^{n} \frac{|d_i|}{|D|}(1 - \sum_{j=1}^{m} p(c_j|d_i)^2) \tag{3}$$

Which is the sum of the relative probabilities p of label c_j given the subset d_i taken over m classes, weighted by size of the subset in proportion to the total size of the data.

In our neural network implementation, we elected to use the tangent hyperbolic function $tanh$ in place of the standard sigmoid transfer function. This was based on its comparable non-linearity and less compute intensive calculation allowing larger networks and faster updates. Weight updates were

[4] p may take on any positive non-zero value, when $p = 1$, the measure is equivalent to the Manhattan distance, when $p = 2$ the measure is equivalent to the Euclidean distance

[5] In practice, we observed little to no difference between GINI impurity and information gain with regards to attribute selection and overall performance.

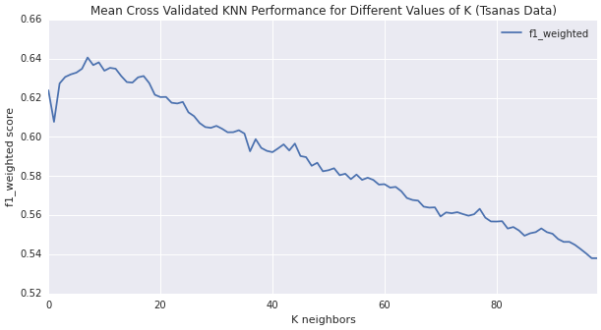


Figure 2: k-NN Performance Over k

performed with stochastic gradient descent using the rule:

$$\omega_i^n \leftarrow \omega_i^{n-1} - \eta \frac{\partial \epsilon_n(\omega)}{\partial \omega_i} = \omega_i^{n-1} - \eta \frac{\partial y_\omega(x_n)}{\partial \omega_i}(y_\omega(x_n) - t_n) \tag{4}$$

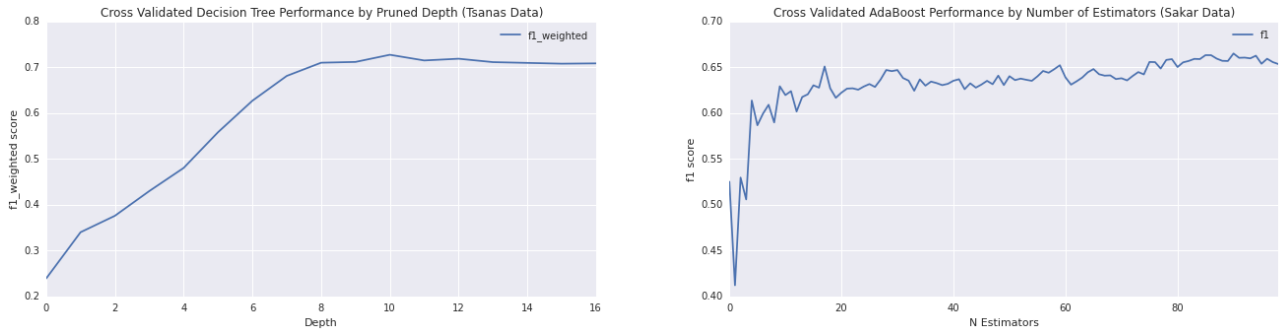
We update the weight vector ω of layer i, on the n^{th} iteration, given a randomly selected instance x_n, t_n by computing the old ω_i^{n-1} less the approximated error ϵ gradient of weight parameters ω with respect to ω_i scaled by some learning constant η. The second equation is an expansion where the error gradient is explicitly denoted in terms of the input vector x_n, target output t_n and network transformation $y_\omega(x_n)$ using parameters ω.

3 Results

3.1 Model Evaluation

The k-NN models trained on these datasets were tested over a range of values for k, with cross-validation applied to each unique value. In the Tsanas dataset, we observed a strong correlation between classification performance and k; specifically, for $k > 8$, classifier f1 scores decrease approximately linearly in k. In the Sakar dataset, however, we observed no such trend. f1 variance across cross validation folds, however, was much higher ($\sigma^2 = 0.0004$) averaged over all k vs the Tsanas dataset ($\sigma^2 = 0.0001$). In these experiments we observe the expected behavior that beyond a particular optimum, as increasingly distant elements in feature space are incorporated into the prediction, classification performance decreases.

The development of decision trees proceeded similarly, cross validated f1 scores were measured for classifiers with different pruning depths. It was observed in both the Tsanas and and Sakar datasets that performance first increased as a function of maximum depth, then remained unchanged. After



(a) Decision Tree `max_depth`

(b) AdaBoost `n_estimators`

Figure 3: Decision Tree and Boosted Trees CV Parameter Search

the tree reaches some optimum depth, decreasing the aggressiveness of pruning has no effect as the algorithm will continue to generate the same optimized tree irrespective of the depth limitation.

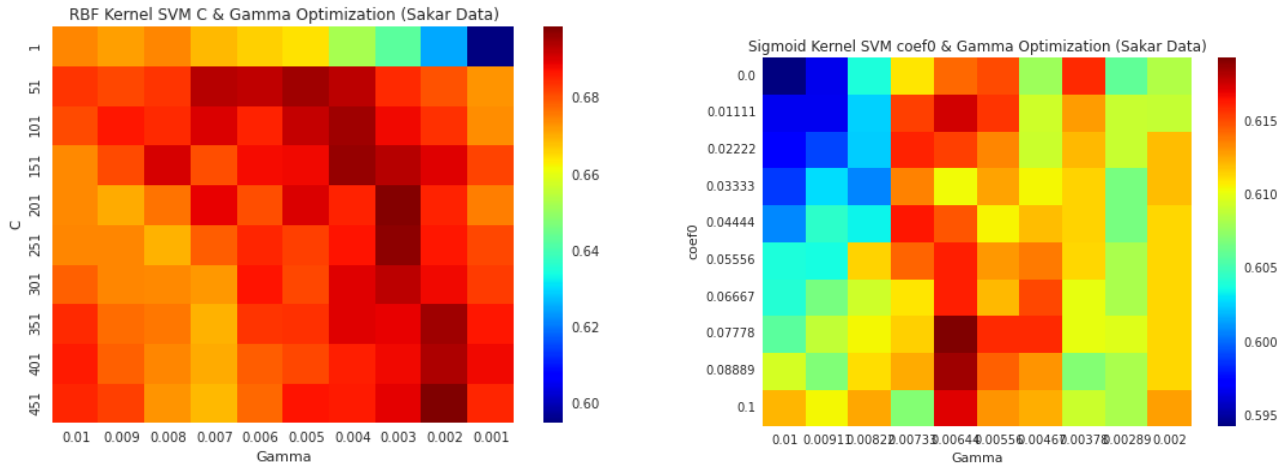
Boosting optimization was conducted by testing over a range of values for the maximum number of sub-estimators that could be generated by AdaBoost. On both the Tsanas and Sakar datasets, around 5 weak learners was sufficient to achieve performance better than uniform random guessing and 20-30 was enough to achieve comparable performance to a classifier with double the number of estimators.

Two Support vector machines were implemented: one with a sigmoid kernel and another with a Radial Basis Function (RBF) kernel. Cross validated grid search was performed over hyperparameters C and either γ or x_0 for RBF and sigmoid kernels respectively.[6] Where C is the misclassification penalty term defined such that large values of C allow smaller margin hyperplanes to be selected if more training points are classified correctly. Conversely, for small values of C, the optimizer will prefer higher-margin separating hyperplanes at the cost of increased training error. In general, the RBF kernel performed better across most observed values of γ and C on both datasets.

3.2 Performance Comparison

Performance on the Sakar dataset was similar for all the assessed algorithms. Surprisingly, generalization performance did not improve much as the size of the training set increased (though there is a clear upward trend in both training and testing f1 scores). That is, most models converged to a level of performance that did not vary much after the training set size exceeded 200 examples. The exceptions to this observation were the sigmoid SVM and the decision tree, which exhibited

[6]Kernel coefficients γ = the kernel coefficient of the radial basis function. x_0 = the constant term of the sigmoid function



(a) RBF Kernel

(b) Sigmoid Kernel

Figure 4: SVM Performance Over γ, C, x_0

relatively erratic performance as a function of training set size. The majority of algorithms trained on the Tsanas dataset exhibit the characteristic logarithmic increase in performance as a function of training set size. The variation in the types of trends we observe however is worth noting. In the case of AdaBoost and the sigmoid SVM, there is tight agreement and overall poor performance on both the training and testing sets. This may indicate that the algorithms are failing to learn a good approximation to the target hypothesis regardless of the size of the training set. This may be indicative of an incompatibility of the algorithms' representational bias with respect to the data. That is, the target hypothesis h_t does not exist in the hypothesis space of the algorithm $h_t \notin H$. This may be the case for the AdaBoost models; we draw this conclusion by investigating the structure of a single decision tree trained on the Tsanas dataset. The attributes closest to the root of the decision tree separate the data based primarily on patient metadata such as age and gender. It may be that only after these assumptions do the vocal dysphonia measures carry much statistical significance with respect to the labels. Therefore a collection of weak learners correlating simple rule sets to the labels may not be sufficient to aggregate into a coherent hypothesis. There may be some inductive bias of the algorithm impacting performance as well. For example, kernel function selection often relies on some a-priori knowledge of the data; judging from its behavior it is likely that the sigmoid SVM suffers from poor kernel selection.

A more expressive means of visualizing multi-class performance on the Tsanas dataset is in the form of Confusion Matrices. Each column of each matrix represents the instances in the predicted stage label, each row indicates the actual stage label. Interestingly, from this arrangement we can

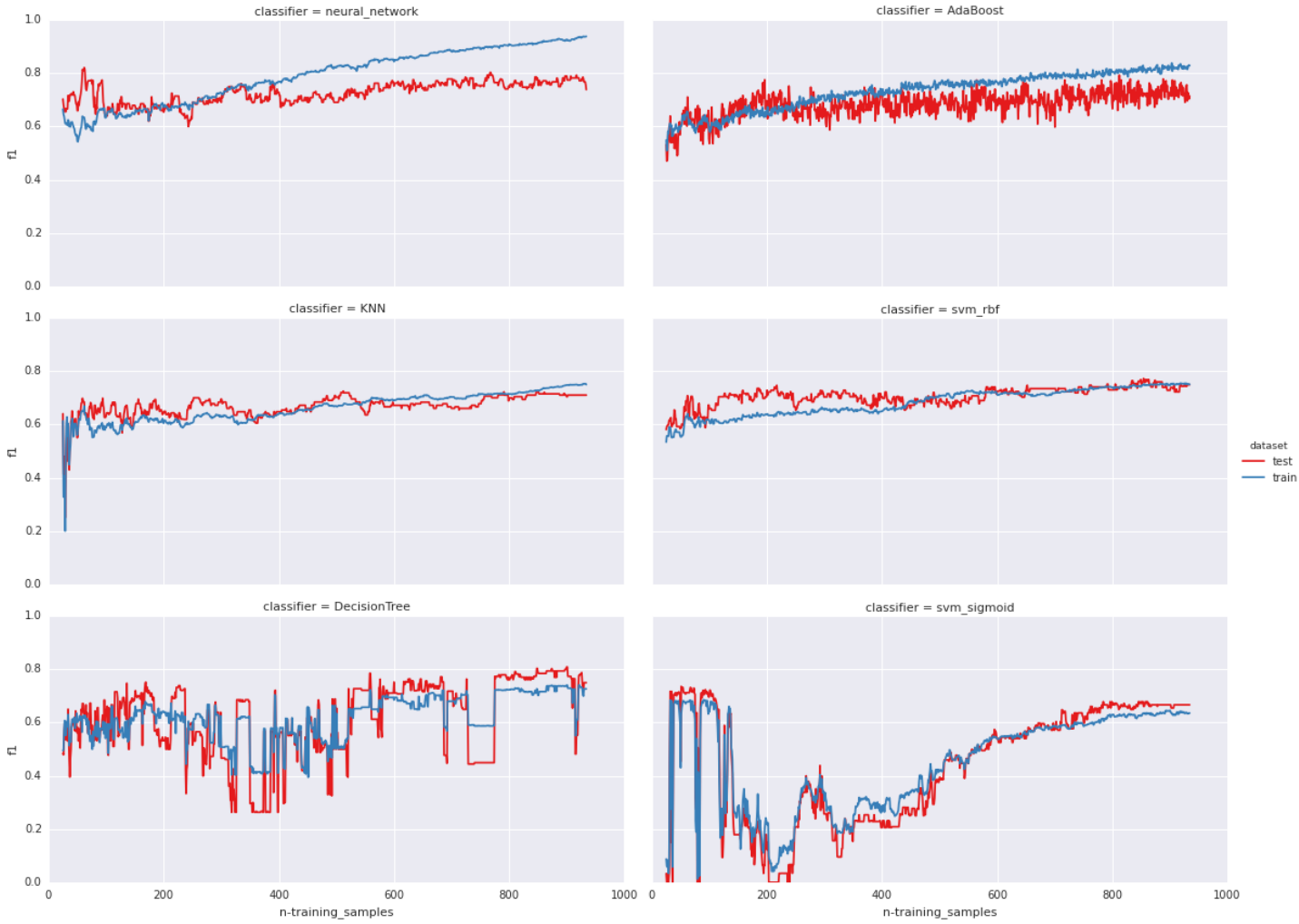


Figure 5: Sakar Dataset Learning Curves

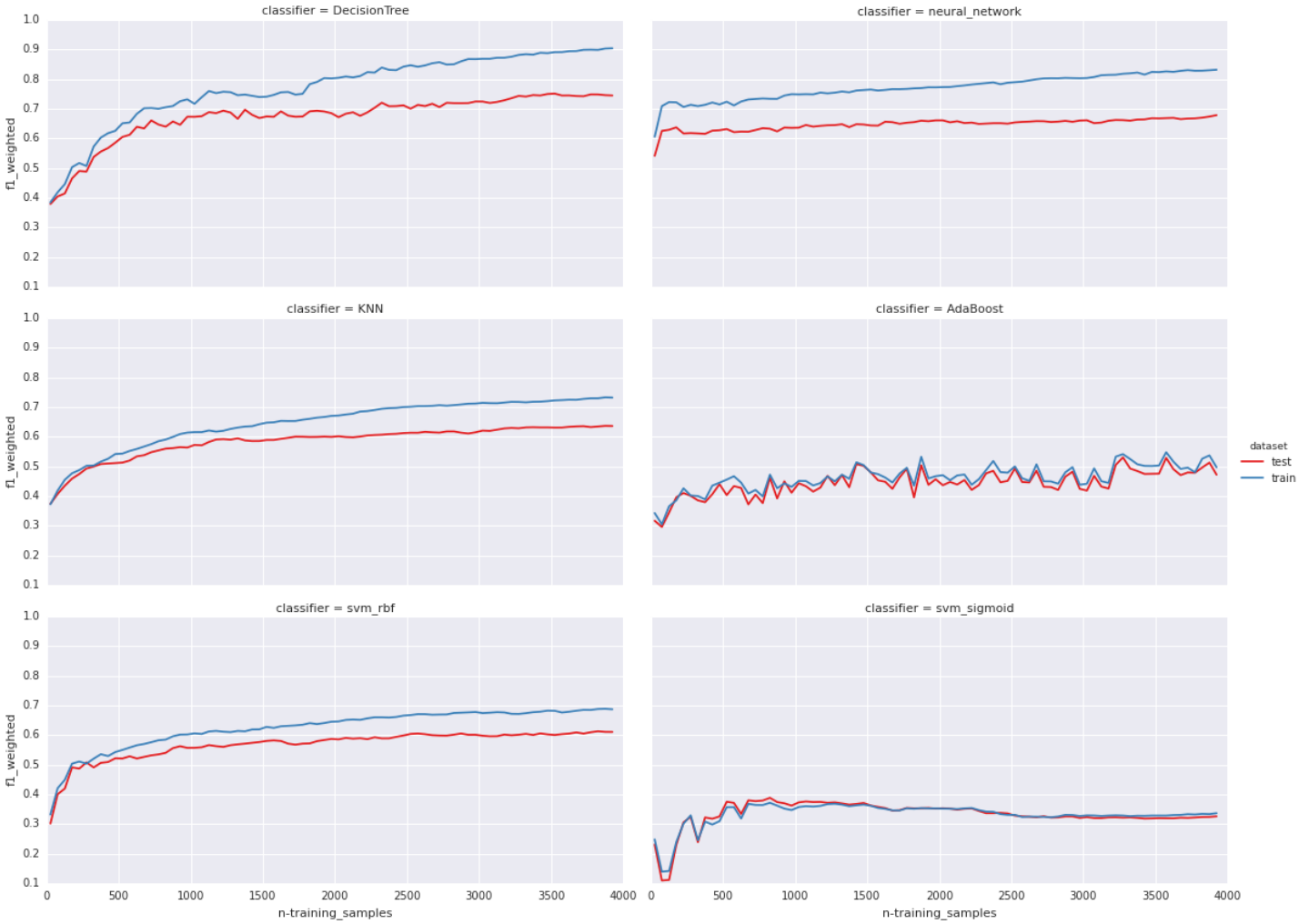
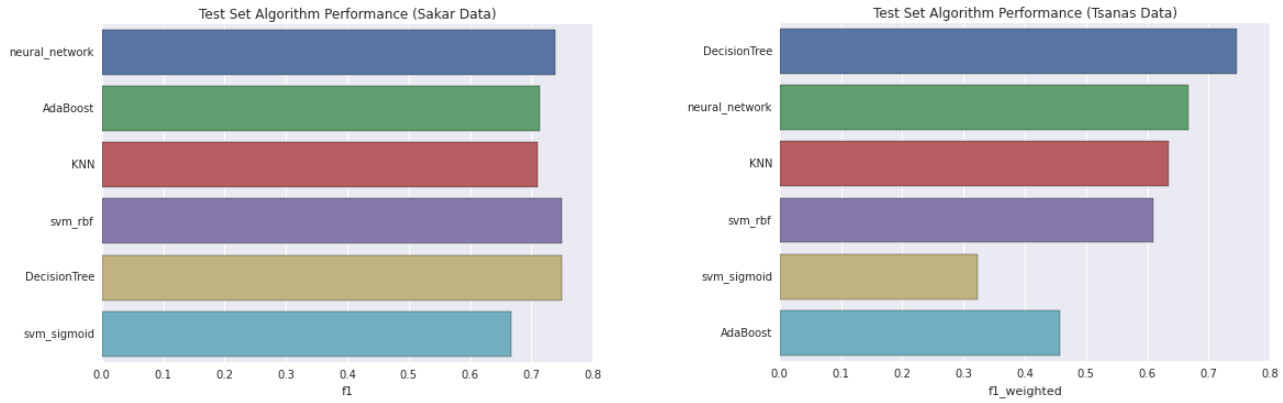


Figure 6: Tsanas Dataset Learning Curves



(a) Sakar Dataset

(b) Tsanas Dataset

Figure 7: Best Model Algorithm Comparison

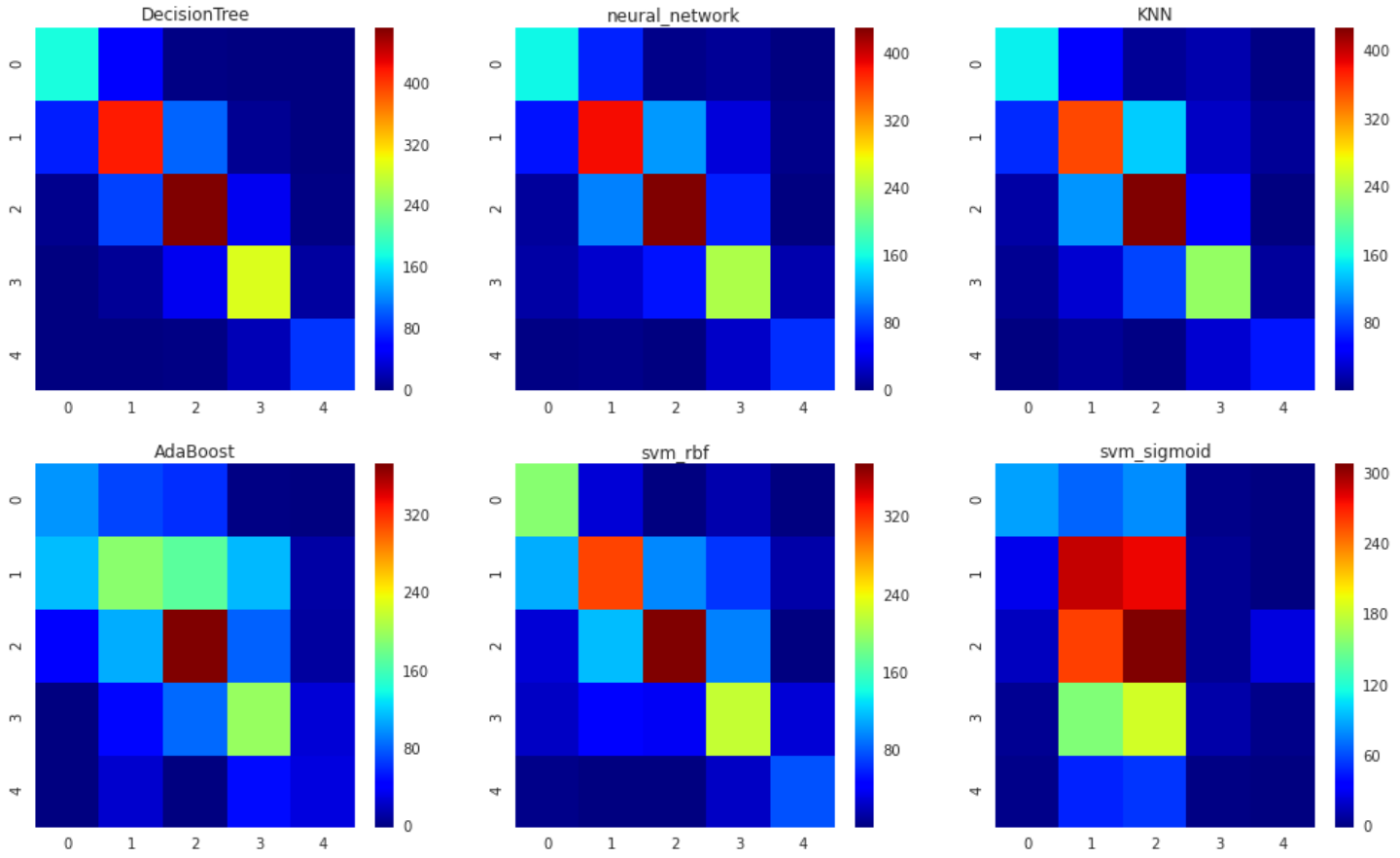


Figure 8: Tsanas Dataset Confusion Matrices

directly observe that the learners that fit the data have a distribution along the diagonal that roughly follows the actual distribution of labels. This follows from the fact that the algorithms all seek to maximize the weighted f1 score, which assigns importance to instances proportional to the probability of their corresponding label.

4 Conclusion

In this study we have explored the value and utility of Supervised Learning algorithms in the development of predictive models for Parkinson's Disease. We have explored these data from the perspective of two unique classification paradigms (binary and multi-class). In the former, we applied five algorithms to voice dysphonia measurements to discriminate PD+ from PD- voice recordings. In the latter, we applied related methods with appropriate modifications to classify the severity of Parkinson's Disease using labels derived from a discretization of the Unified Parkinson's Disease Rating Scale. From our analysis, we have observed that decision trees achieve the best f1 score in most

contexts. The performance of sigmoid kernel SVMs and boosted trees indicates that these methods may be unsuited to the multi-class PD stage prediction scenario as we have defined it. Further, we have gained some insight into the nature of the PD voice diagnostic problem which suggests that augmenting dysphonia measurements with patient metadata such as age and gender may improve classification performance. Overall, we have quantified and compared the utility of supervised learning algorithms in the context of Parkinson's Disease speech pathology.

References

[1] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[2] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2015-09-15].

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] B. E. Sakar, M.E. Isenkul, C. O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin, and O. Kursun. Collection and analysis of a parkinson speech dataset with multiple types of sound recordings. *IEEE Journal of Biomedical and Health Informatics*, 17(4):828–834, jul 2013.

[5] A. Tsanas, M.A. Little, P.E. McSharry, and L.O. Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57(4):884–893, apr 2010.