ML 695 Group Project - Final Report

Matthew Benvenuto George Marfo Anthony Mascaro

December 11th 2022

1 Introduction: Early Parkinson's Detection

Table 1: A visual of Parkinsons

Healthy
Parkinson's Disease before symptoms
Parkinson's Disease after symptoms

1.1 Overview of the Problem

There is no definitive laboratory test for Parkinson's and diagnosis can be difficult especially in early stages. Utilizing machine learning techniques using patient audio recording data can allow for earlier screening processes. This can increase early detection while acting as an incredibly non-invasive test. ML techniques may allow for faster treatment of the disease.

Understanding which traits observed on a patient's audio recording are vital to this screening process. Many tools and methods do not utilize key non-linear data which can capture important patient symptoms. We will include both legacy and more recent audio features to create the most predictive detection model.

1.2 Methodology

We begin our exploration in utilizing this data by treating it as a classification problem. As stated previously we will utilize both the legacy and newer more complex audio features. We will utilize PCA, Logistic Regression, SVM, Random Forest, and Ensemble Learning to achieve this.

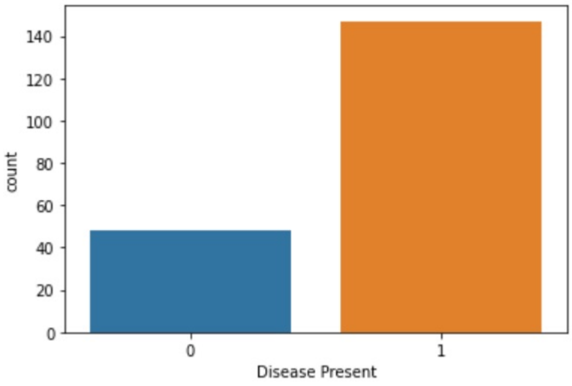
2 Related Work

2.1 Existing Techniques

It is believed that some audio tools and the models that analyze the data they provide are missing key bio-indicators of the disease. Vocal disorder is a key indicator of Parkinson's Disease. These disorders show both simple periodic imperfections as well as nonlinear and non-Gaussian random imperfections. Many of the tools employed today do not capture these important non-linear indicators.

Periodic vocal indicators such as fundamental frequency and amplitude have been used to attempt to detect Parkinson's. Fundamental frequency, which is a rate of vibration in vocal folds, has been shown to be increased in Parkinson patients. These results are in question and have been tested over time with varying results. (Reference 2)

Table 2: Presence of Disease (Imbalance in Data



Other and more powerful tools have been used to capture recurrence and fractal scaling in vocal speech. These measures pick up non-linearity and turbulent randomness in vocal speech which are key markers of Parkinson's. Using recurrence and fractal scaling a ”hoarseness” diagram is created. This is achieved using a bootstrapped classifier using these two features to verify vocal disorder. (Reference 3)

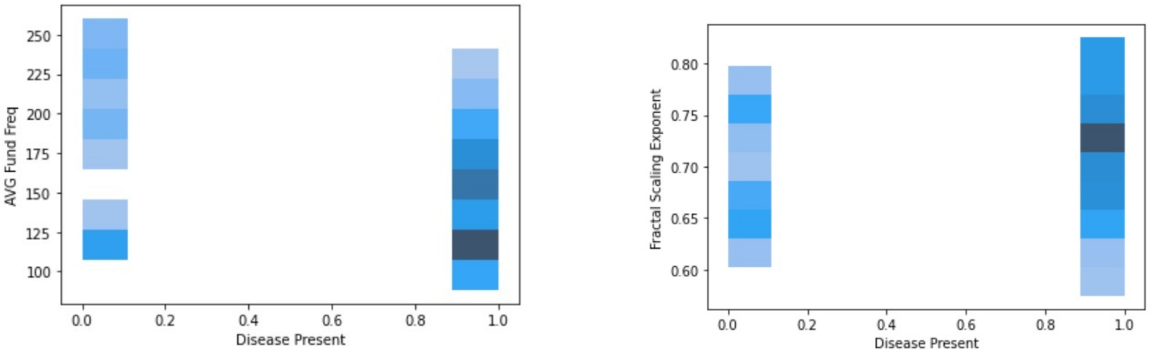
3 Our Solution

3.1 Description of Dataset

The data includes 195 patient audio recordings with 23 features. These in- clude the periodic audio measurements like fundamental frequency as well as the non-linear measurements such as fractal scaling. There is also a binary pre- dictor of Parkinson's for each patient. (Reference 1)

For pre-processing we created a dictionary of features to more easily understand and discuss them. We verified that there were no null values in our data. Our EDA revealed that the patients without Parkinson's was a minority class so we balanced our data. We reviewed distributions of each feature in our target variable. We ensured there were no null values in our data. We reviewed correlation between our features. (see figures below)

Table 3: Presence of Disease in Frequency Features vs. Fractal Feature



3.2 Machine Learning Algorithms

We have initially chosen to use a logistic regression given this is one of the simpler methods we have learned and that classification models have shown re sults in addressing this problem. It will be a good baseline to compare other classification models off of.

PCA is a powerful dimension reduction tool that aims re-orient the data on an axis that maximizes the variance thereby reducing the dimensions of the data.

We plan to use a support vector machine (SVM) because our sample space is small in addition to the algorithm working well with irregularly distributed data. This problem of identifying disease state is a classification task, and this model is fit for purpose and tends to not suffer from overfitting. In addition, outliers have less of an influence when executing this model

We plan on implementing the random forest algorithm, as our data is classification of disease. The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction is more accurate than that of any individual tree.

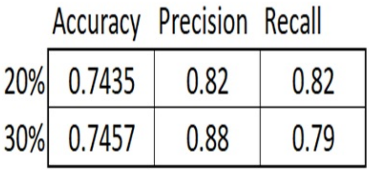
The Ensemble Model is a powerful way to integrate Machine Learning models into a larger model that interacts the decisions and applies a weight across each individual model to make a better prediction. Weighting the models that compose the ensemble is a tricky problem, that can ultimately be solved with Grid-Search via Hyper-Tuning to identify the appropriate weights to allocate to each classifier.

3.3 Implementation Details

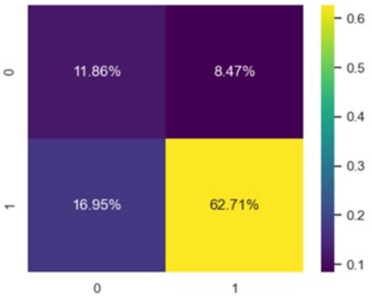
3.3.1 Logistic Regression

A number of logistic regressions were completed in order to arrive at the best set of parameters for this type of model. Initially the test size of the data was split at 20% as well as 30% . The regression with a split of 30 % for the test data provided better results. The accuracy's were nearly identical but with a much higher precision, although there was a slight decrease of recall.

Table 4: 20 vs. 30 split, and 30 % Error Matrix

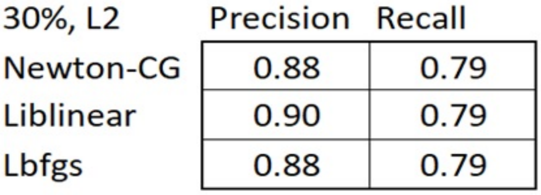


	Accuracy	Precision	Recall
20%	0.7435	0.82	0.82
30%	0.7457	0.88	0.79



For the tuning of the regression I investigated different solver techniques as well as regularization penalties. The Newton-CG, Liblinear, and lbfgs solver techniques were all used. The Recall was identical in these scenarios but the precision was slightly higher at 0.9 (vs. 0.88) for the liblinear technique. Adjusting the L1 and L2 regularization penalties had no additional impact on the metrics. For the ensemble learning technique (to be discussed later) a 20 % test split was used in order to preserve consistency between our models.

Table 5: Tuned LR Precision and Recall



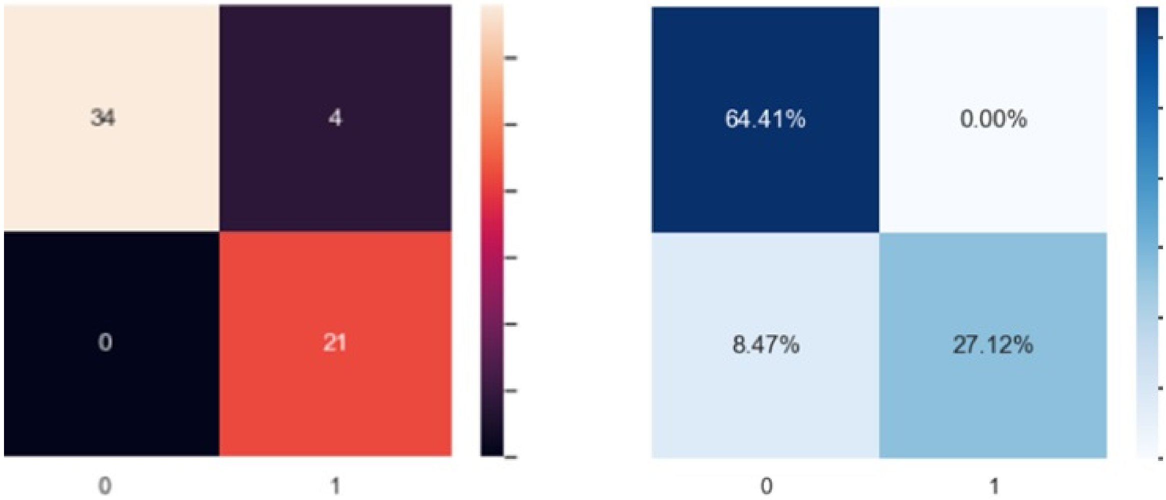
30%, L2	Precision	Recall
Newton-CG	0.88	0.79
Liblinear	0.90	0.79
Lbfgs	0.88	0.79

3.3.2 Support Vector Machines and Principal Component Analysis

Support Vector Machines (SVM), which creates non-linear boundaries with the largest margins (hyperplane) to classify the data points, were implemented. In inspecting the data there was a low training complexity given its size, which provided an appropriate space to leverage the model. In designing the feature matrix, I selected all the variables to provide a descriptive feature space for the classifier to learn from.

With some pre-processing and dimension reduction techniques (PCA) the data was partitioned into a training set consisting of 80 % of the data and a test set consisting of the remaining 20 % . The model was evaluated on the pre-processed and normalized data (to reduce magnitude of the values while preserving their ranges), on both the balanced and unbalanced states of the data's target column "Disease Present". Ultimately, the balanced data set produced the best scores in terms of accuracy and AUC. Taking this model as a benchmark I performed a "GridSearch" for hyperparameter tuning of the better model, which not surprisingly led to a model with better performance. This result led me to select the tuned model for integration into our prediction pipeline for Parkinson's disease.

Table 6: SVM Error Matrix, Tuned SVM Error Matrix



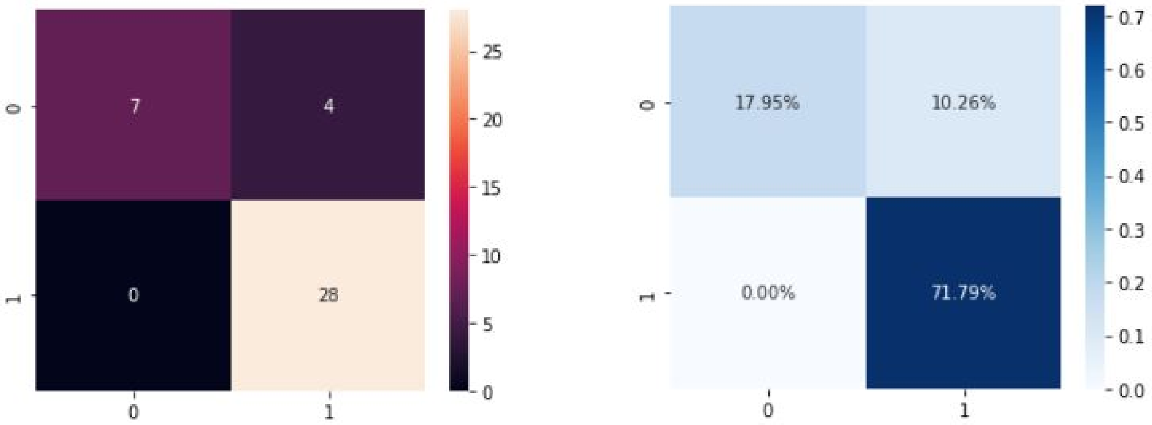
3.3.3 Random Forest

Random Forest is one of the most powerful algorithms in Machine Learning. The initialization under the hood of the sklearn RandomForestClassifier() package utilizes bagging to achieve the accurate results. While growing the trees, random forest, randomly samples a tree at each arbitrary level. When splitting at each node, this technique searches for the best feature among an already randomly generated subset of features as opposed to selecting the most important feature. Since the data we were working with was noise, there is apparently "noise" within the noise that needs to disambiguate. Random Forest is particularly good at handling noise within the data, so ultimately this technique was a strong candidate Machine Learning model to build.

The development of this model started with the initialization of the RandomForestClassifier() and then passing this initialization along with the parameter list mes,entropy, max depth from 4 to 8, max features of auto,sqrt,log2, number of estimators 50 to 500 to the GridSearchCV function. Upon tuning, the hyper-parameters selected were entropy, max depth of 7, max features of auto, and number of estimators of 50.

The f1 score reported was .933

Table 7: Random Forest Confusion Matrix represented as both numbers and percentages respectively

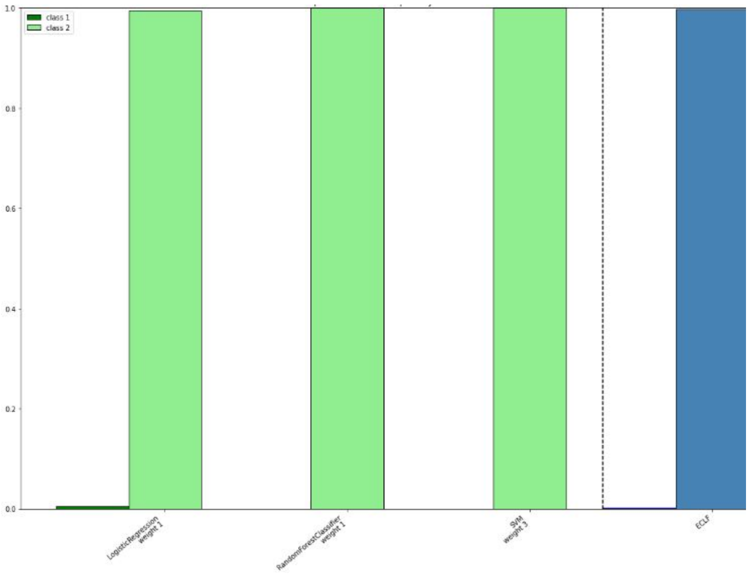


3.3.4 Ensemble Model

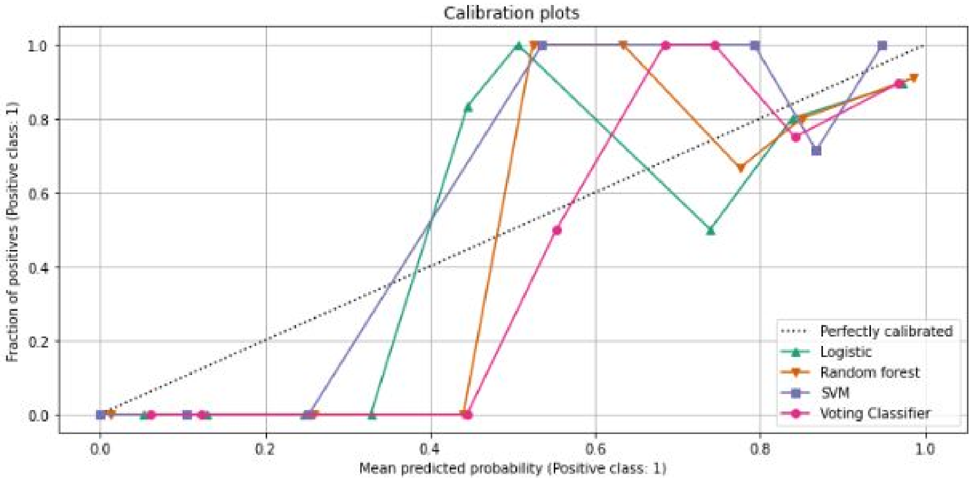
Code for building the ensemble model weighting the model

```
    #clf1,clf2,clf3 are the three ML models that compose the ensemble_model
    clf1 = LogisticRegression(penalty=’l1’,solver=’liblinear’,max_iter=200)
    clf2 = RandomForestClassifier(criterion=’entropy’,max_depth=5,max_features=’sqrt’,
    n_estimators=50)
    clf3 = SVC(C=1,gamma=1, probability=True)

    #The Ensemble Model
    ensemble_model = VotingClassifier(
        estimators=[("lr", clf1), ("rf", clf2), ("svm", clf3)],
        voting="soft",
        weights=[2,2,.5],
    )

    #Training and Testing the Ensemble Model
    ensemble_model.fit(X_train,y_train)
    ensemble_model_predicted = ensemble_model.predict(X_test)
```

Table 8: The breakdown of the Ensemble Model : The Class Probabilities for Sample 1 by Different ML Models



3.4 Comparing the Models



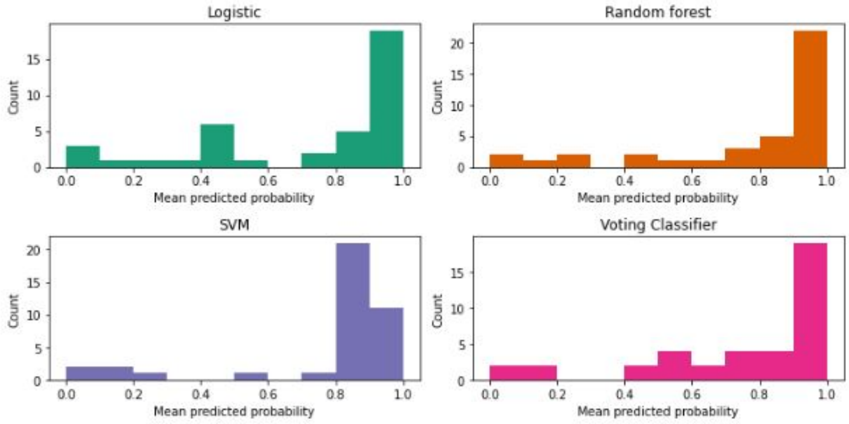
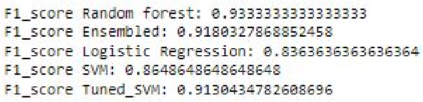


Table 9: Comparison Results: Measured Using F1-score



```
F1_score Random forest: 0.9333333333333333
F1_score Ensembled: 0.9180327868852458
F1_score Logistic Regression: 0.8363636363636364
F1_score SVM: 0.8648648648648648
F1_score Tuned_SVM: 0.9130434782608696
```

4 Future Directions

4.1 Type I vs. Type II Error

If given many more months we would complete research on what is a more negative outcome, a false positive or false negative. We can either misclassify a patient with Parkinson's as negative and therefore delay for an unknown period of time the beginning of treatment. We could also misclassify a patient without Parkinson's as positive and begin them on a treatment plan they do not need and potentially miss some other disease. In addition, what are possible side effects of medication that they could be unnecessarily prescribed? We could research these and better understand if we should focus these types of models to limit Type I or Type II errors.

4.2 Time Series Audio Data

Each of the features that are being processed through the Machine Learning models are time series data, frequencies, amplitudes, namely measurements of vocal signals that are intimately related. Whether one feature is a direct derivative of another, or whether one feature is capturing the maximum, another feature capturing the minimum, or another capturing the average, each feature that the ML models are training on are connected. With the abstraction of such signals, information on anomalies of the vocal chords is lost and the model is not maximizing the knowledge that is potentially there in the data. If our data was presented as a raw audio clip, there is potential for directly feeding the data points through the model resulting in a higher success rate.

5 Conclusion

Out of all of our ML Techniques the Random Forest performed the best (using F1-Score as a metric). The tuned SVM and Ensemble Learning Method were not far behind. The Logistic Regression had the poorest F1-Score. Combining the legacy and non-linear audio features did allow us to create a predictive model, with multiple techniques, for Parkinson's detection that could be helpful in the early screening process.

6 Sources

Reference 1: Parkinson Diseason Detection:https://www.kaggle.com/datasets/debasisdotcom/parkinson-disease-detection

Reference 2: Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection: https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-6-23

Reference 3: Effects of Parkinson's Disease on Fundamental Frequency Variability in Running Speech: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4380292