Early Prediction of Parkinson's Detection

ML 695 Group Project – Presentation

Matthew Benvenuto George Marfo Anthony Mascaro

Parkinson's Disease Background and Early Screening

- \- Chronic neurologic condition first named in 1817 by Dr. James Parkinson
  - A slow progressive disease that causes nerve cell loss in brain leading to disfunction of motor symptoms
    - Loss of neurotransmitter dopamine
    - Resting Tremor, Bradykinesia(overall slowness), and Cogwheel Rigidity (limb stiffness)
    - Impact on speech: Dysarthria (sound articulation), Hypophonia (lowered volume), and Monotonicity (reduced pitch range)
    - Risk of Dementia is increased
    - Actor Michael J. Fox, prominent in the 1980's and 90's, has battled this for decades
  - No definitive lab test so detection can be challenging
  - Impacted speech can be an initial sign so patient audio tests are vital
  - Early detection can be improved by using Machine Learning to analyze audio data
    - Audio testing is incredibly non-invasive
    - Early detection can lead to faster treatment

Problem Statement and Objective

- Use a variety of Machine Learning Algorithms to detect Parkinson's Disease as a classification problem

- Existing research exists using both legacy audio features as well as more complex measurements
  - There is debate if existing audio analysis is sufficient with only legacy measurements
  - Existing audio tools also may not be sufficient to capture more recent complex measurements

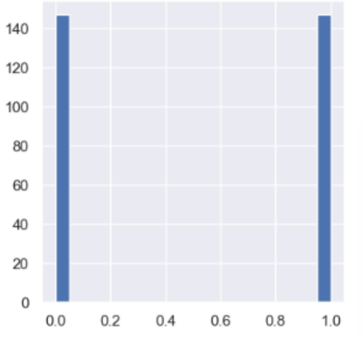
- We will set out to combine both sets of features into our algorithms to achieve the best predictive model possible for patients

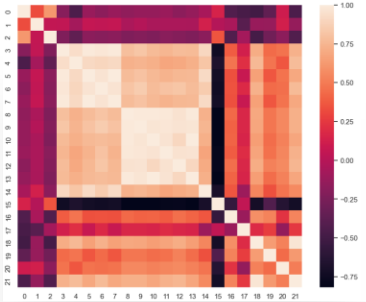
- Early screening and detection is vital to begin treatment as well as improve a very non-invasive audio test

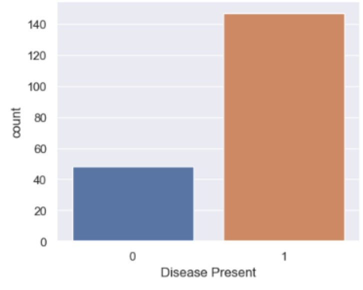
- We will utilize the following ML techniques
  - Principal Component Analysis, Logistic Regression, Support Vector Machines, and a Random Forest

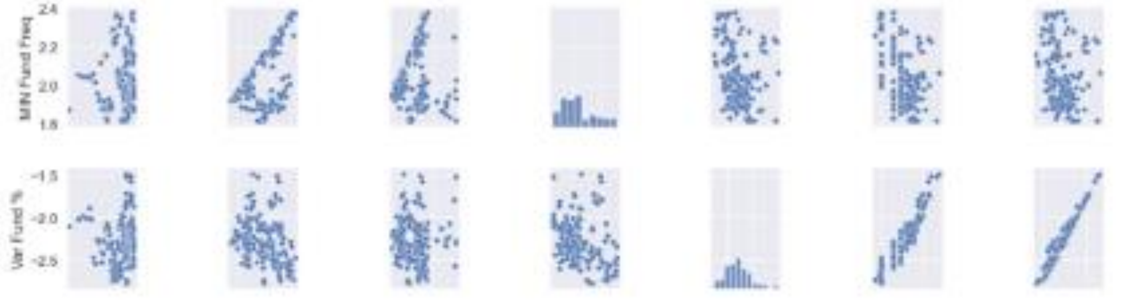
Description of Data

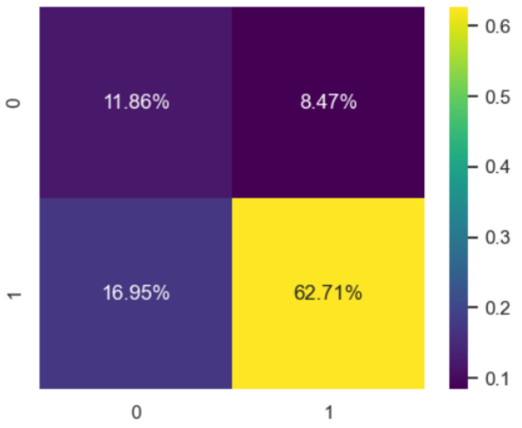
- 195 patient audio recordings with 23 features
- Input variables: 22 features of disease bio-indicators
    - Legacy audio measures and more complex non-linear measures
- Predictor variables: 1 feature (presence of disease in patient)
- Pre-Processing and EDA
    - Created dictionary of features to more easily understand and discuss
    - Investigation of missing values (verification of no null values in data)
    - EDA led to balancing data given minority class of patient’s without disease
    - Scaled features in order for model’s to run
    - We reviewed pairplots of features for distribution and the correlations between them

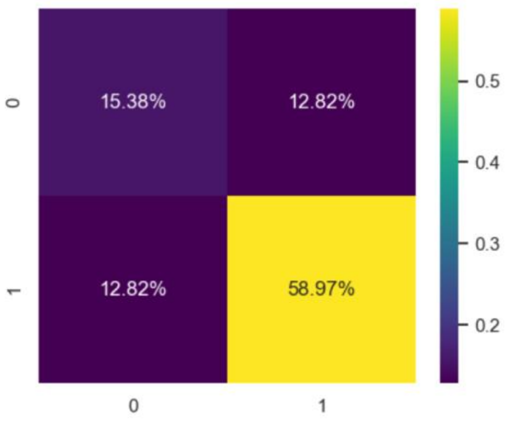












	Accuracy	Precision	Recall
20%	0.7435	0.82	0.82
30%	0.7457	0.88	0.79

Logistic Regression

- Normal scale of features to limit model iterations
  - Model could not perform with initial values
- Various logistic regressions were performed to achieve best results
- Initial regressions: 20% vs. 30% test data
  - 30% split showed slightly better results
  - Nearly identical accuracy with a significant improvement to precision but a slight decrease in recall

Logistic Regression Tuning

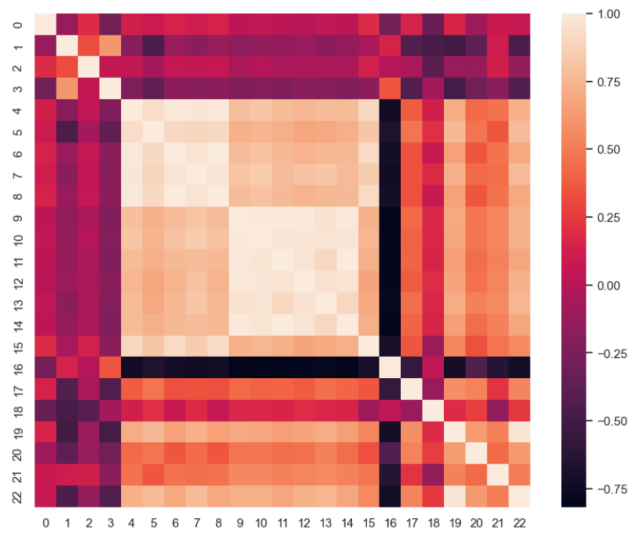
- Further regressions completed with different solver techniques and penalties
  - Newton-CG, Liblinear, and lbfgs at 30% test split
  - L2 Norm penalty used for regularization
    - Liblinear had a slightly better Precision than others
  - Liblinear using L1 vs. L2 for regularization conducted
    - Identical results with adjusting to L1 penalty
  - Ultimately the 30% split, Liblinear solver, and L1/L2 penalty performed the best
  - A 20% split was used for our ensemble algorithm in order to preserve model consistency

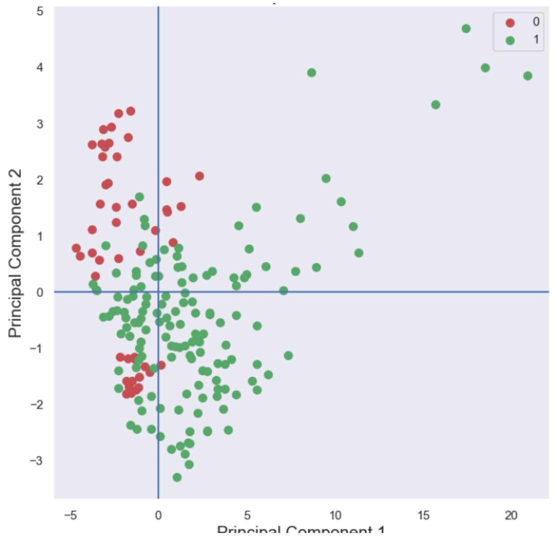
30%, L2	Precision	Recall
Newton-CG	0.88	0.79
Liblinear	0.90	0.79
Lbfgs	0.88	0.79

30%, Liblinear	Precision	Recall
L1	0.9	0.79
L2	0.90	0.79

Principal Component Analysis

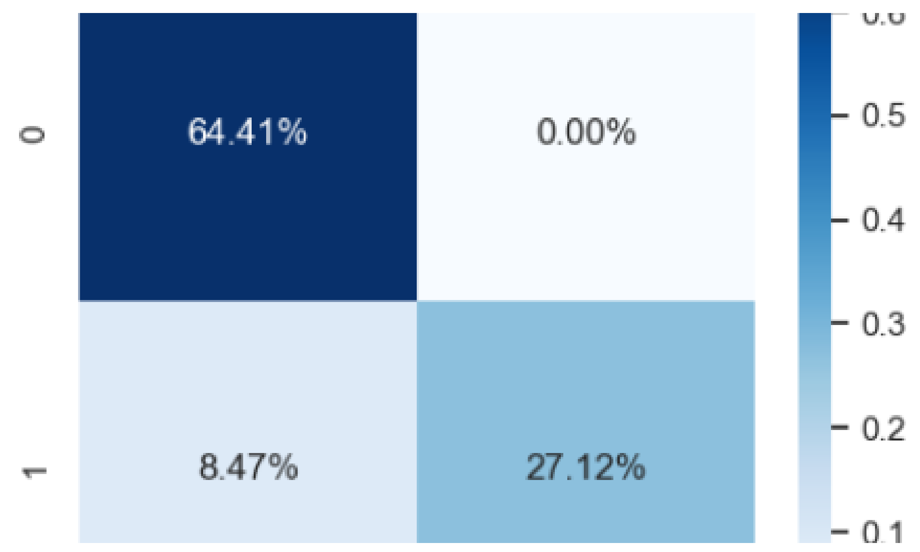
- A data dimensionality reduction technique that aims to maximize the variance in the data while minimizing the error in its principal components
- Correlation seemed to indicate redundancy among variables
- Leverage PCA to reduce the feature space accordingly

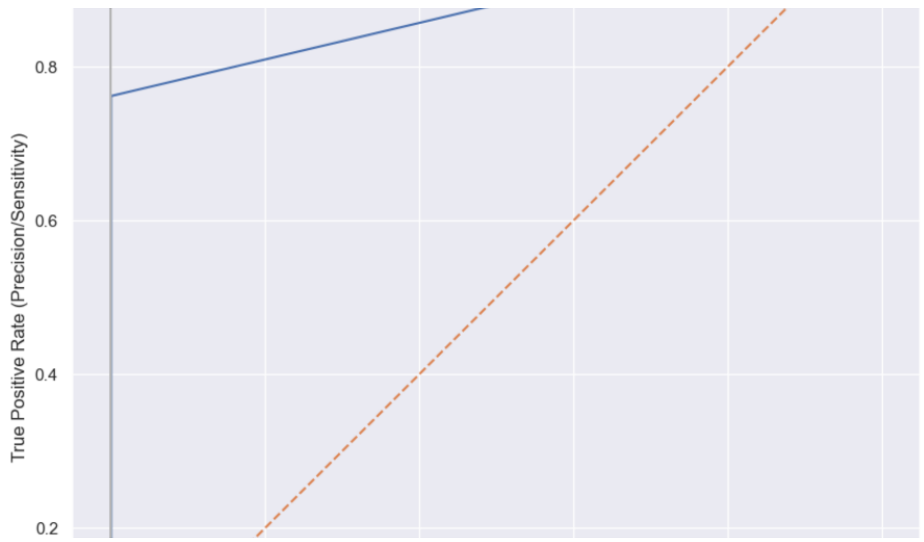




Support Vector Machine

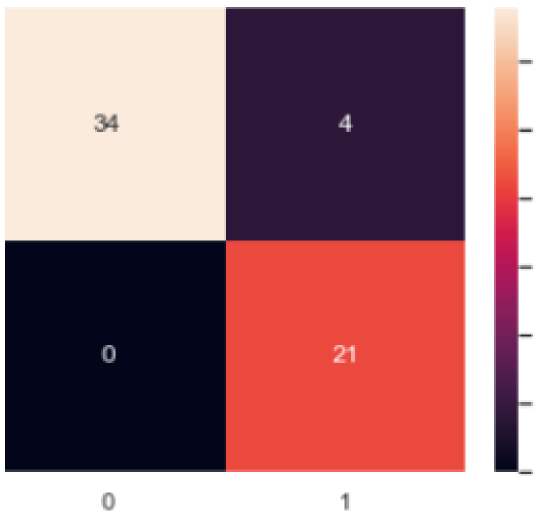
- Supervised Machine learning model, mapping data to a high dimensional feature space for classification
- Accuracy: 0.91
- F1_score: 0.86
- AUC: 0.88

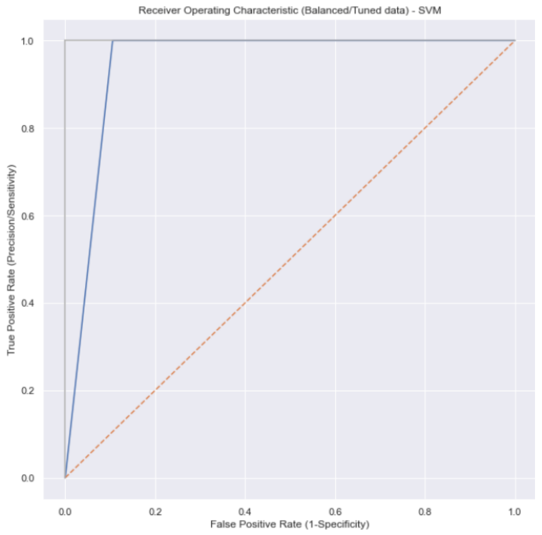




Support Vector Machine Tuning

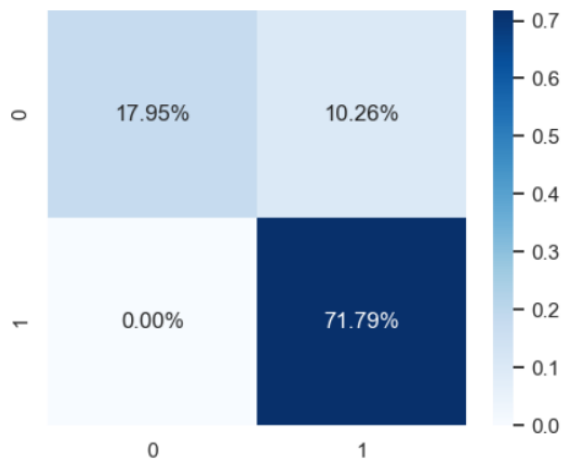
- Gamma = 1 ; C = 1, Kernel = ‘rbf’
- Gamma: Controls distance of influence points
- C: Defines the margin of the hyperplane
- Kernel: Perform the mapping
- Accuracy: 0.93
- F1 Score: 0.91
- AUC: 0.94

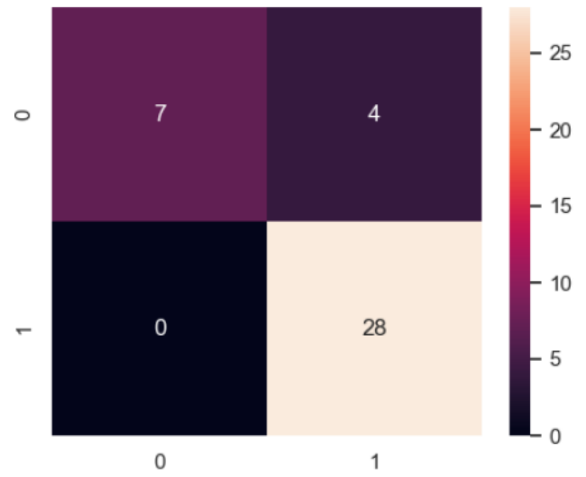




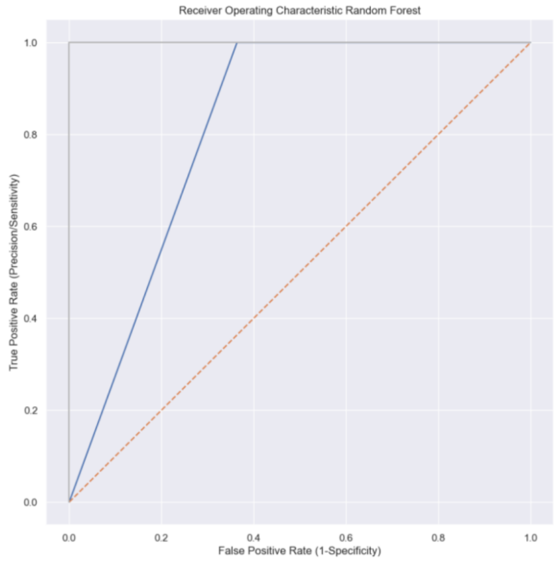
Random Forest

- Supervised Machine learning model, It is an ensemble method, meaning that a random forest model is made up of a large number of small decision trees, called estimators, which each produce their own predictions. The random forest model combines the predictions of the estimators to produce a more accurate prediction.
- Accuracy: .897
- F1_score: 0.933





Random Forest - Tuning



- HyperTuning Parameters
    - criterion: 'entropy'
    - max_depth: 7
    - max_features: 'auto'
    - n_estimators: 50

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'criterion': ['mse', 'entropy'],
                         'max_depth': [4, 5, 6, 7, 8],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'n_estimators': [50, 100, 200, 500]})
```

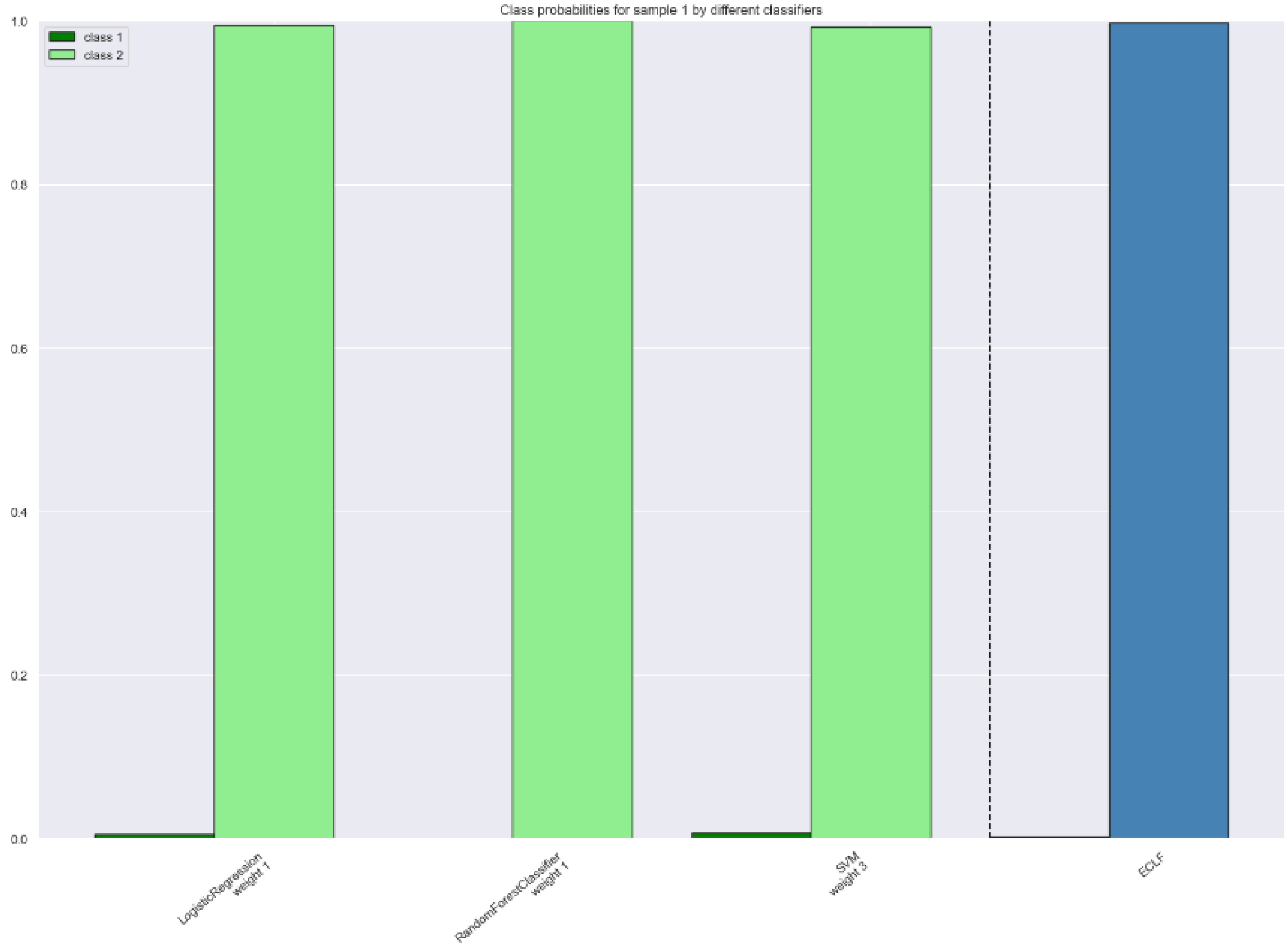
Ensemble Learning: Voting Classifiers

- Logistic Regression Weight : 2
- RandomForestClassifier Weight : 2
- SVM Weight : .5

After GridSearching

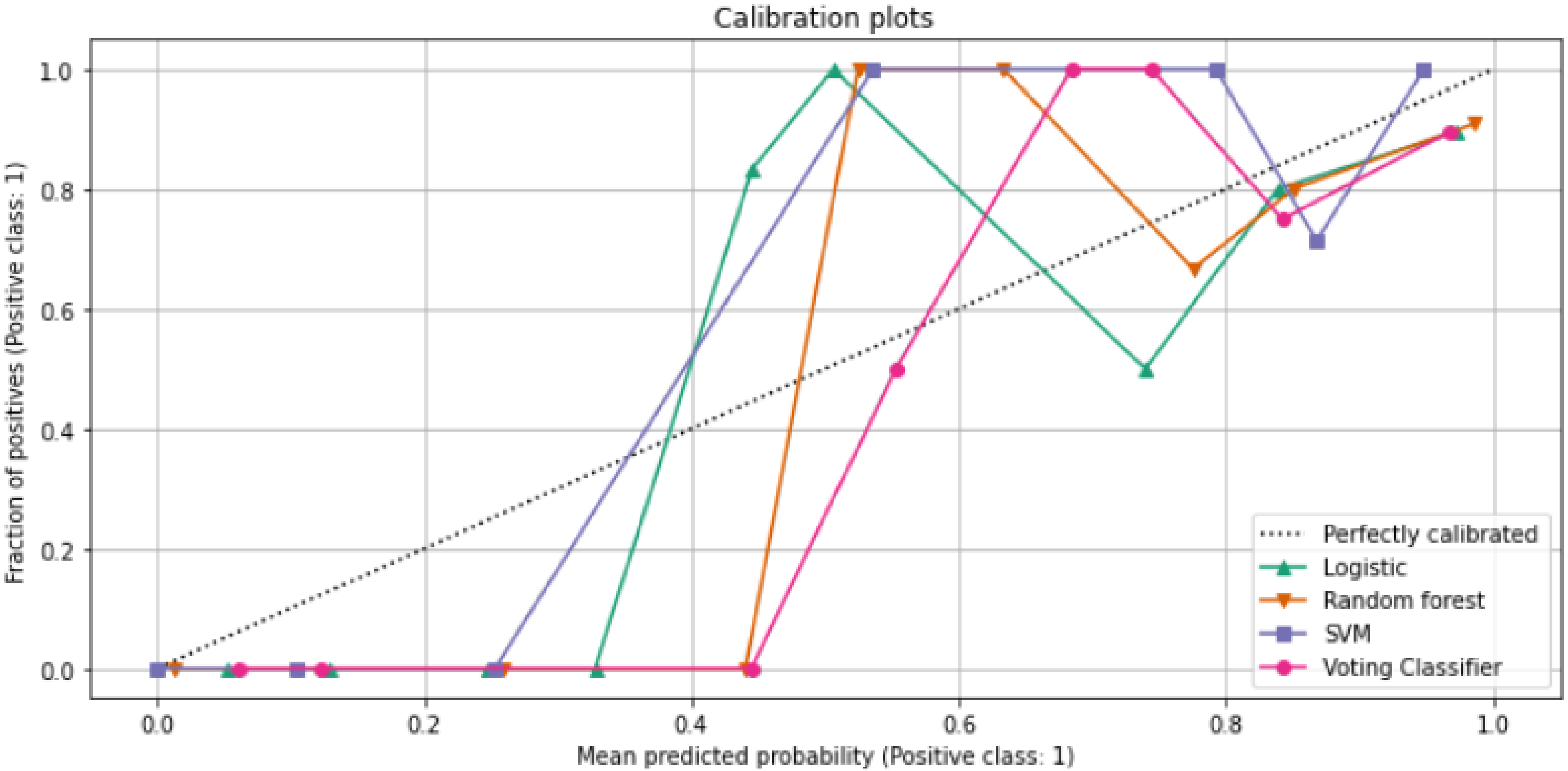
- Hyper Tuning Parameters:
- voting: 'soft'
- weights': (1, 1, 1)

Cross Val Score: .9416

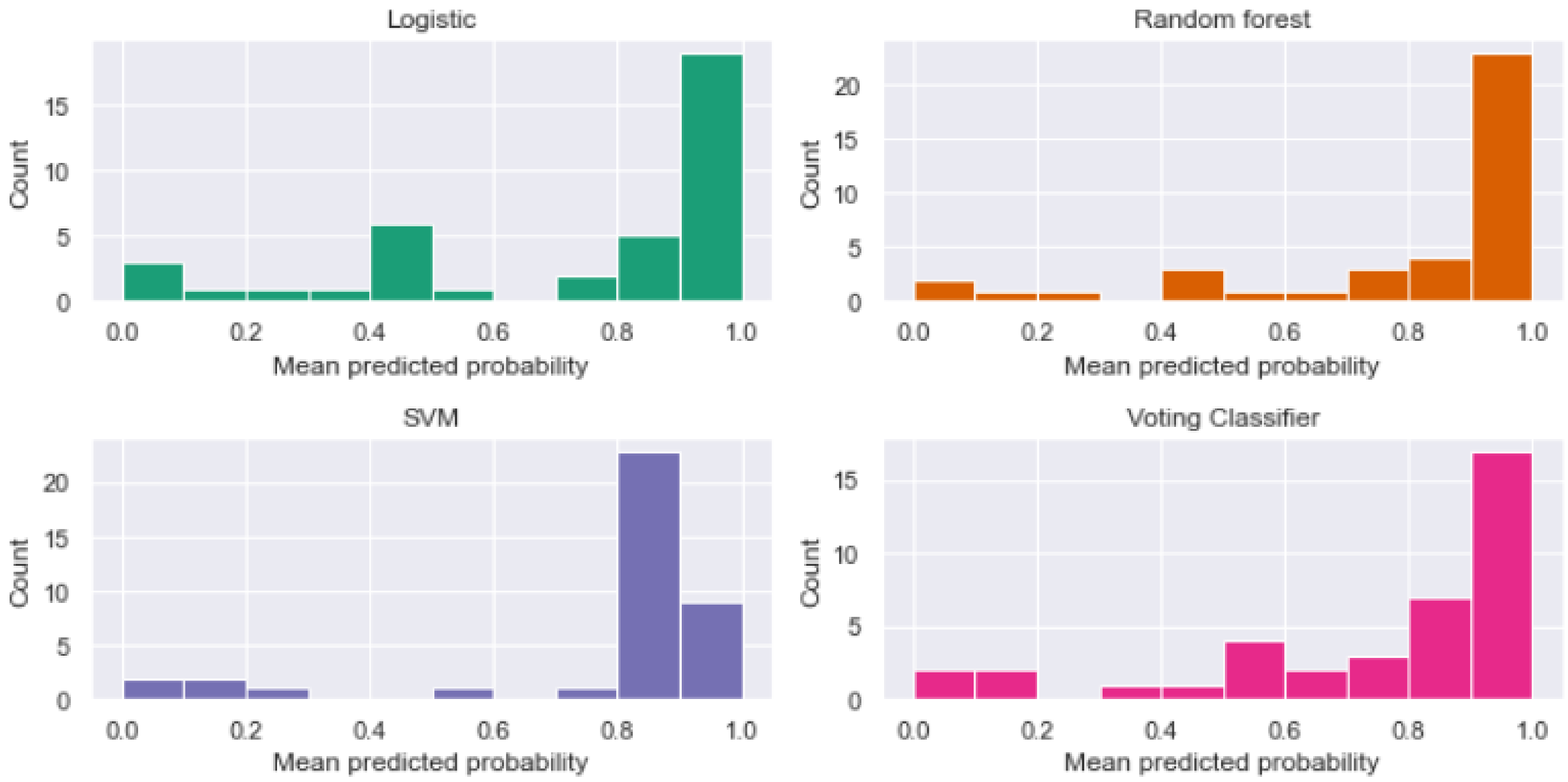


Comparison of Techniques

- From the plot, we can see that there is room to calibrate our models further.



Comparison of Techniques Continued



Conclusion

```
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
```