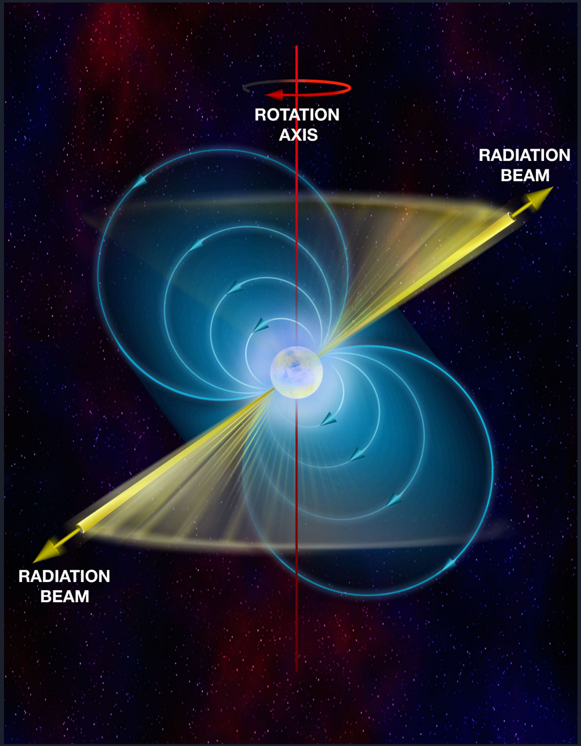
Benchmarking Models in Classifying Pulsar Stars

By: Anthony Cavallero, Tong Liang, Nicholas Kashani Motlagh, Rohit Rajendran

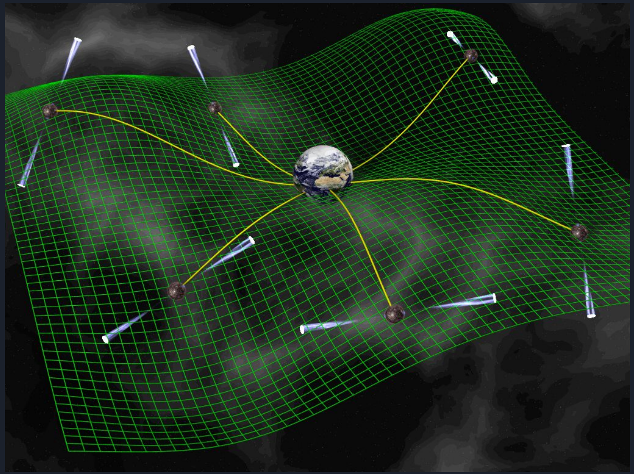
Introduction: What is a Pulsar?



- Neutron stars are formed from the crushed core of a massive star that cannot support fusion. When the star is between 1 to 3 solar masses the crushing of the core bind protons and electrons into a solid ball of neutrons, thus a Neutron Star remains. This process condenses masses from millions of times the earth down to a sphere only a few miles across

- Pulsars are an easily observable subset of Neutron stars. Due to their high rotational speed they generate a strong magnetic field that sweeps jets of radiation across our sensors at an extremely consistent period.

Introduction: Why are Pulsars Important?



Pulsars have a number of useful applications in the scientific world:

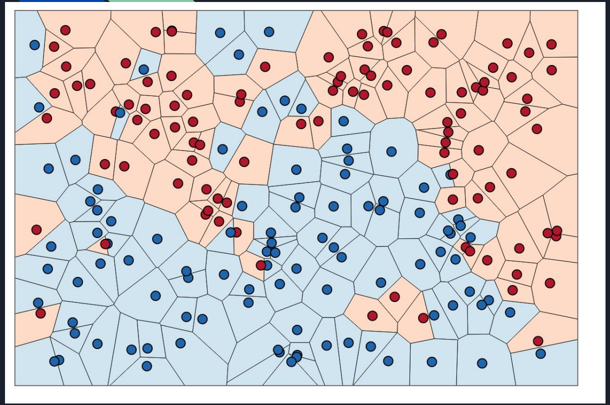
PTA - Pulsar Timing Arrays exploit the near perfect timing of millisecond period Pulsars being observed by radio telescopes across the globe. Roughly 100 of these Pulsars are observed with the goal of detecting fluctuations attributed to gravitational waves interrupting the matter traversing the pulse. This data is used to support and study general relativity in the universe

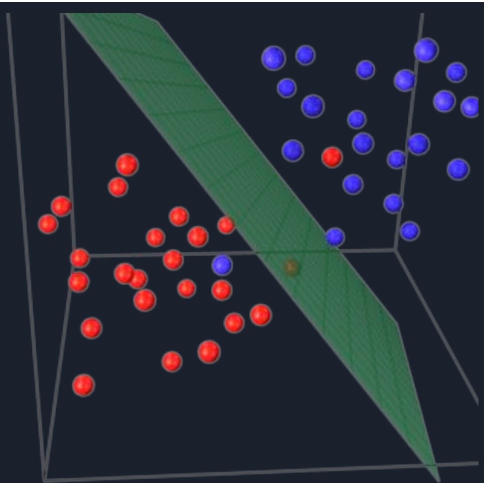
Dark Matter Detection - Similar to the PTA example, we can use the fluctuations of consistent pulses to detect halos, or collections, of dark matter traversing the line of sight between the pulsar and Earth where no visible matter is detected to disrupt the pulse (Theoretical concept)

Set-Up

- Binary Classification was performed with a plethora of Machine Learning models on the High Time Resolution Universe Survey (HTRU), which is the collection of survey data from the Parkes Radio Telescope, which searches for candidate pulses that could be generated from a Pulsar.
- Our data encompases 4 statistics across 2 main features of a pulsar: the Mean, Standard Deviation, Skewness, and Kurtosis of the Integrated Pulse Profile(Averaged Modal Graph of Pulses over a given time) and Dispersion Measure (How much the pulses have been distorted while traversing interstellar material like gas and dust)
- The data used consists of 17898 total samples, 1639 Real Pulsars and 16,259 fake examples from noise and other sources

Method: K-Nearest Neighbors and SVM, Principal Component Analysis





- K-NN: Simple K-NN model finding the closest k points and classifying based on the majority of the labels.

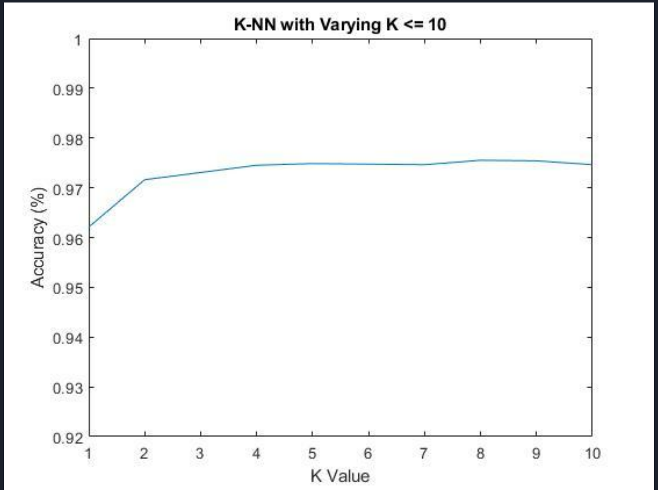
- SVM: Calculate optimal hyperplane separating data based on margin with slack parameter c.
- Small c value => Larger margin at expense of higher error.
- Large c value => Higher accuracy at expense of possible overfitting.

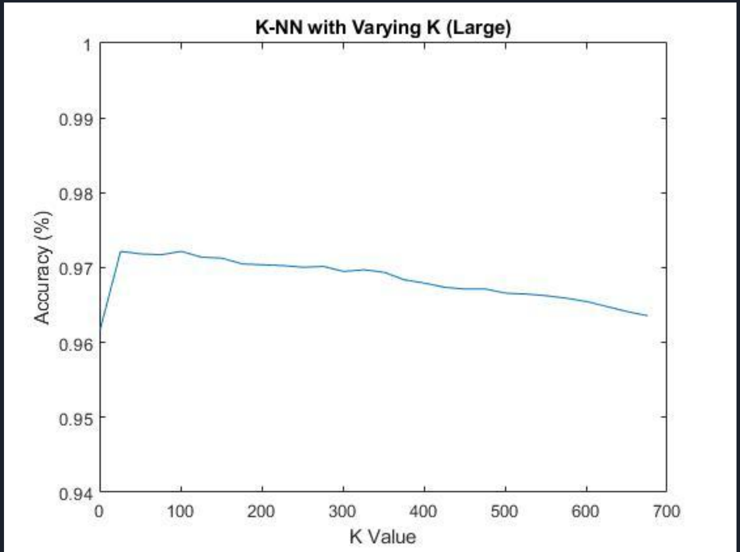
- PCA analysis: deconstruct feature set to eigenvectors to find most informative dimensions for feature dimension reduction.

Results: K-NN

K-NN:

- K-value between 5-100 optimal.
- Accuracy of 97.2% compared to 90.8% accuracy using majority classifier.
- Precision of 92.3%.





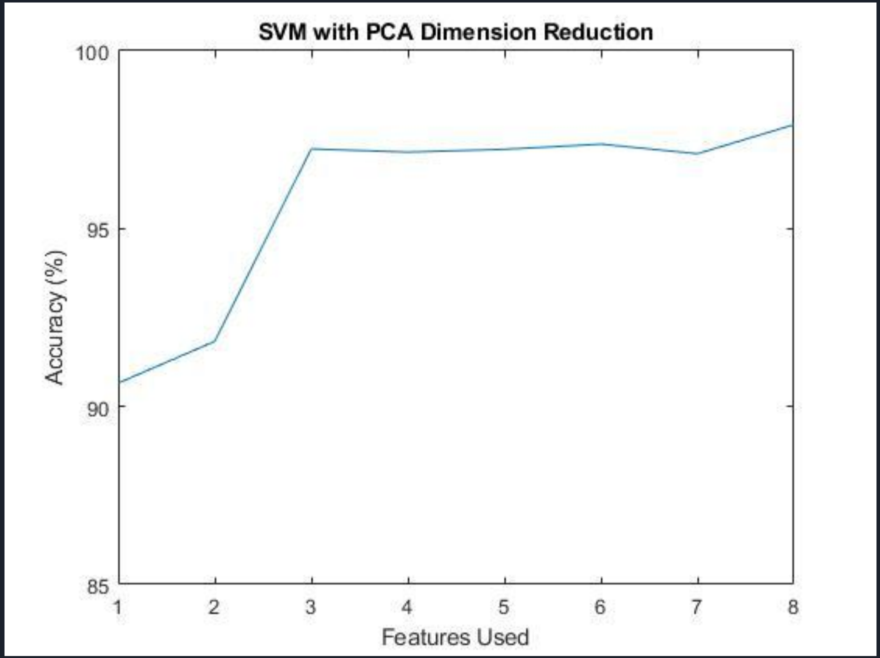
Results: SVM, PCA

SVM classifier (with c value of 1)

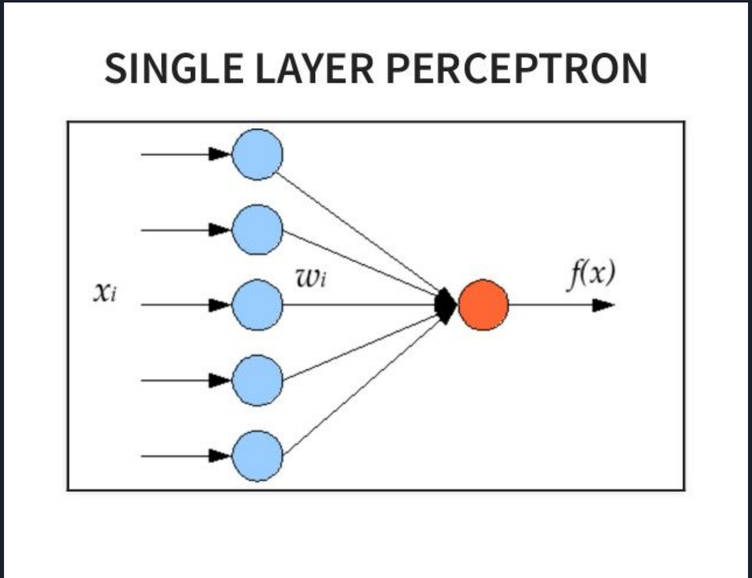
- 97.8% accuracy, 95.2% precision

PCA Dimension Reduction:

- Using just the 3 most informative features yielded accuracy and precision within .6% of using all 8 features



Method: Single Layer Perceptron

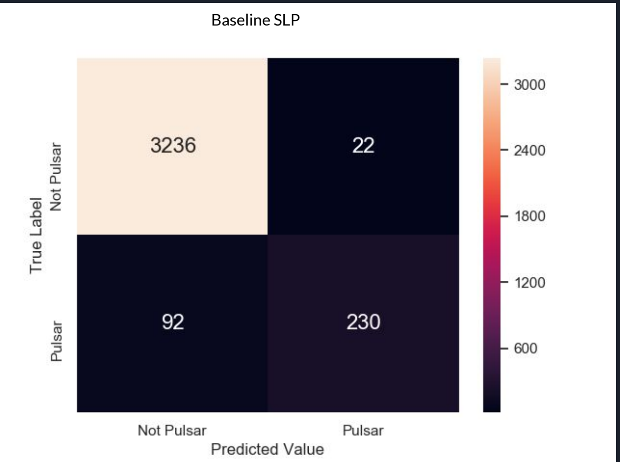


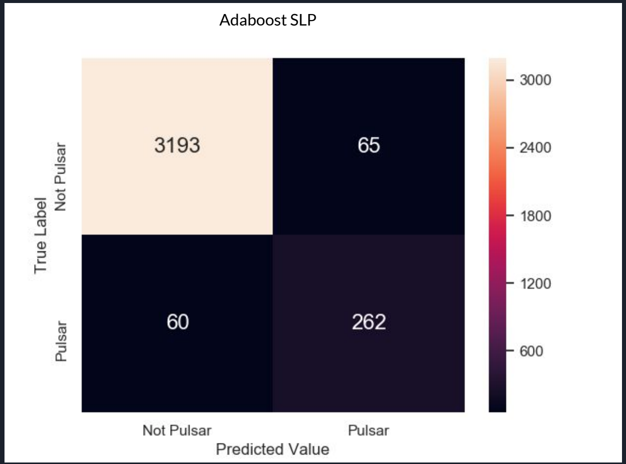
This Perceptron model is the same as what was discussed in class. We input our 8 features and a bias term multiplied by an updated weight parameter for each term and bias.

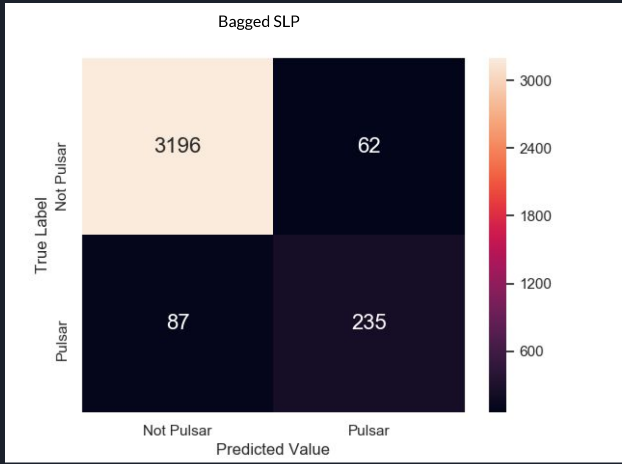
Once the sum of all the features and bias multiplied by their weights exceeds the threshold for our activation function, the Perceptron activates and classifies the example as a Pulsar, else it is not a pulsar.

A learning rate of .0001, and Maximum Iteration of 100 combined with an L2 loss function was determined through Grid Search to be the most optimal hyper parameters as the baseline, Adaboost was applied with 50 model iterations on the baseline model, and Bagging was applied with 15 voter models also based on the baseline

Results: Single Layer Perceptron Confusion Matricies







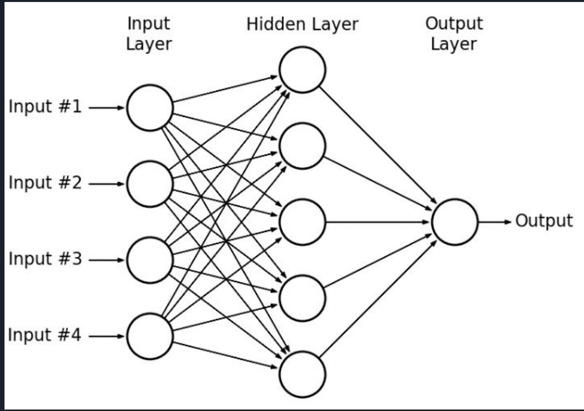
Test Accuracy: 96.8%
Test Precision of Pulsars: 71.4%

Test Accuracy: 96.5%
Test Precision of Pulsars: 81.4%

Test Accuracy: 95.8%
Test Precision of Pulsars: 73%

Baseline is the most accurate, but Adaboost sacrifices a little accuracy for a 10% increase in Precision

Method: MultiLayer Perceptron and Bagging



Multilayer Perceptron Architecture:

- Uses many perceptrons in layers.
- Minimizing loss function over a given training set.
- Multiple hyper parameters including activation function, hidden layer sizes, number of epochs, learning rate, and optimizer.
    - Hidden layer sizes included (4,4,3,2), (4, 4, 4, 3), (5, 4, 3, 2), (5, 2)

Bagging:

- 15 estimators are trained with the same randomly sampled data
- Majority vote is taken during classification

Experimentation:

- Train many models with different hyperparameter permutations
- Compare testing accuracy
- Use the tuned model as a baseline estimator for a bagging ensemble classifier

Method: MultiLayer Perceptron

Baseline Model:

- Activation function of Relu
- Default learning rate of .0001
- Default max iterations of 200

Optimally Tuned Model:

- Activation function of Tanh
- Learning rate of .01
- Max iterations of 400
- Hidden layer architecture of (5, 4, 3, 2)

Results: MultiLayer Perceptron Confusion Matrices

Test Accuracy: 98.11%
Test Precision of Pulsars: 83.21%

Test Accuracy: 98.54%
Test Precision of Pulsars: 79.93%

Test Accuracy: 97.89%
Test Precision of Pulsars: 74.82%

AdaBoost SVM

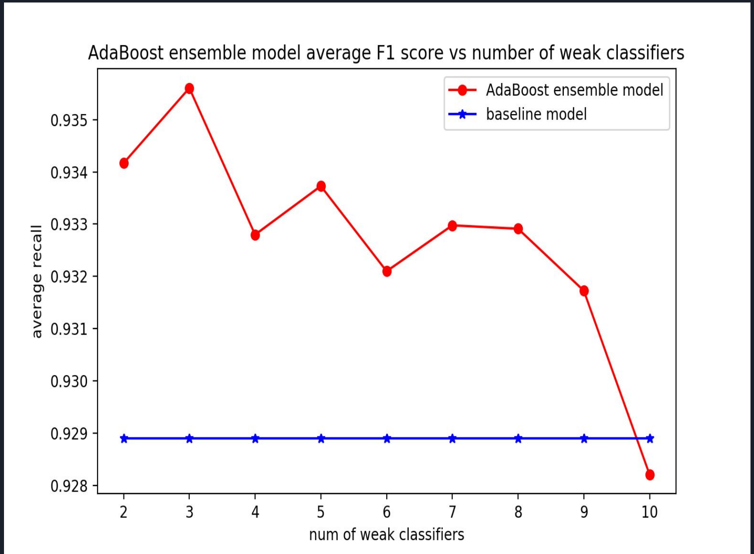
Baseline:
Combining 3 baseline classifiers, SVM, nearest neighbour and MLP with a voting classifier.

Tuned ensemble model:

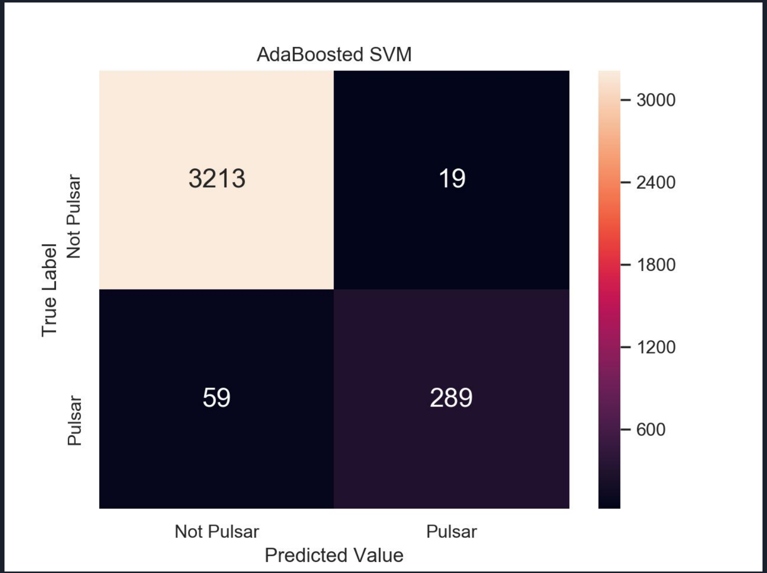
Using PCA estimated from training set to decorrelate and whiten the training and testing data.

Using SVM with radial basis function (RBF) kernel as the base estimators (weak classifier) for AdaBoost.

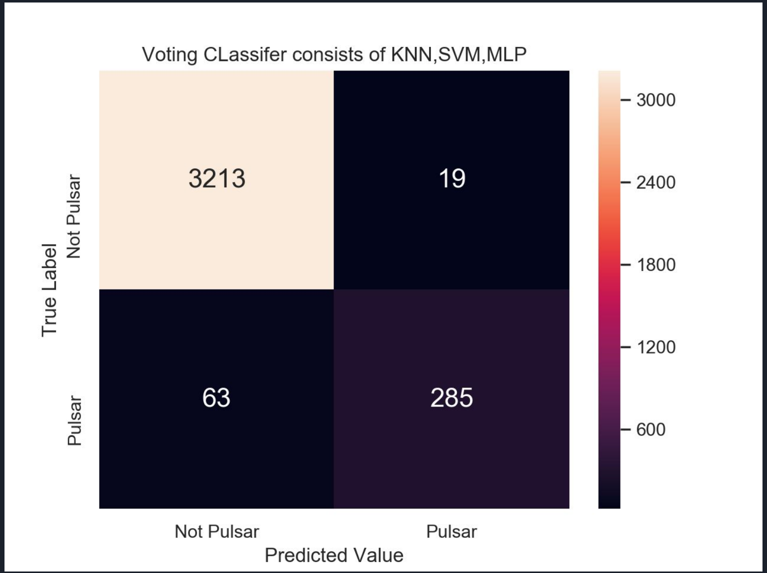
Three SVM base estimators in AdaBoost showed the best evaluation results in terms of average F1 scores.



AdaBoost SVM



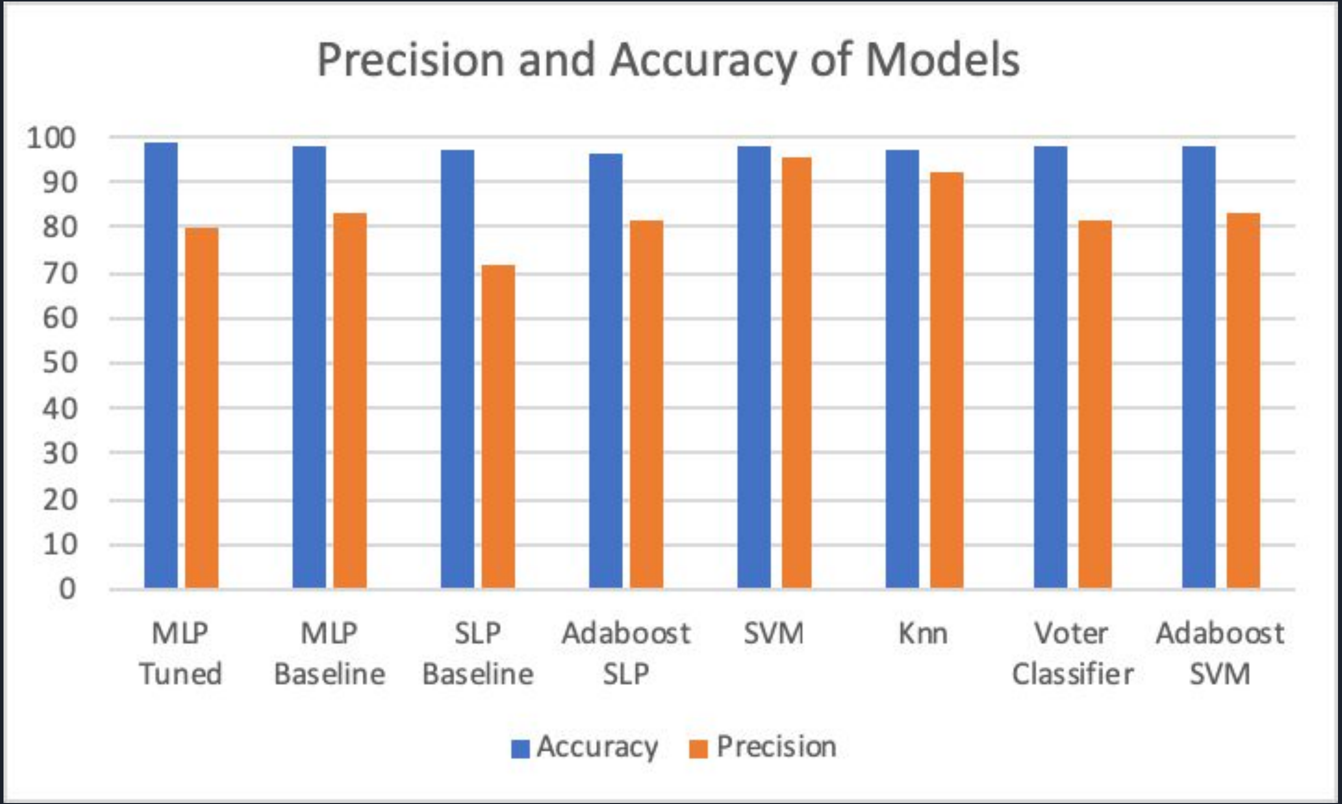
Test accuracy: 97.82%
Test precision of Pulsar star: 83.05%



Test accuracy: 97.71%
Test precision of Pulsar star: 81.89

Results

We can look at models with the best accuracy and precision from experiments



Conclusion

Optimal results

- The most accurate classifier was the tuned multilayer perceptron
  - Accuracy: 98.54%
  - Precision: 79.93%
- The most precise classifier was the SVM
  - Accuracy: 97.80%
  - Precision: 95.20%

Takeaways

- SVM is the most optimal solution for this problem, contrary to our initial findings detailed in our paper, yet it is inefficient to train as it relies heavily on matrix multiplication which is inefficient.
- Multilayer perceptron began overfitting, but in a general case, it would be sufficient to use Single Perceptron with Adaboost or Multilayer Perceptrons as a efficient solution to the classification problem.

Conclusion

Takeaways Continued:

- SVM's stellar performance allows us to attribute a linearly separable nature to our data, however the failure of MLP and SP to achieve a high precision suggests that outliers and skewness of data lead to overfitting on the Non-Pulsar Label

- Another solution to the precision problem is using a K-NN classifier, it trains quickly due to the nature of our dataset being mostly separable with some outliers and achieves similar outcomes as the SVM classifier.

Future Work

- Gather a larger dataset helps us combat overfitting in our models, in addition gather more pulsar examples to combat skewness in data and provide a stronger class definition.
- Construct an Adaboost ensemble compatible with MLP (SkLearn does not support this). This would allow us to conduct a more thorough analysis of model performance and enable us to see if MLP gets a similar boost in precision that SP gained.
- Use our models with new unseen data from HTRU to see if the models hold up in practice compared to in theory.