Prachin J Thapa

Final Project Report

HTRU2 Pulsar Report

Introduction

- *Pulsars*

Pulsars are a rare type of neuron stars that emit electromagnetic wave radiation when they spin in the form of beams. They are remnants of massive stars that have gone supernova explosions with extremely strong magnetic fields. These beams are detectable here on Earth in the form of radio signals when the beams point toward the Earth. Pulsars were first discovered in 1967 by Jocelyn Bell Burnell and Antony Hewish while conducting radio astronomy observations using a telescope at the Mullard Radio Astronomy Observatory in England. Pulsars play a significant role in astrophysics and cosmology.

The rapid rotating pattern of pulsars involves people looking for period radio signals with large radio telescopes.The emission pattern of each pulsar is slightly distinct and varies with each spin as well. Hence, a 'candidate' star is averaged over several pulsar rotations based on the duration of observation. However, pulsar signals are often buried in overwhelming background noise because of radio frequency interference and noise, making signals hard to detect. Human-based analysis is subjective and time consuming.

- *Machine in Pulsar Learning*

High Time Resolution Universe Survey was conducted to search for pulsars. It enables the radio transmissions at a frequency of 1400 MHz. They are all-sky survey for pulsars and radio transients
Machine learning algorithms can learn the pattern of pulsars with feature extraction techniques. It offers faster and accurate prediction on the pulsar detection. My primary objective of the project is to develop a machine learning project which can effectively predict a potential pulsar and non-pulsar candidates in a given dataset. Building a robust and classification model.

Introduction to datasets
The data set shared here contains 16,259 examples caused by RFI/noise, and 1,639 real pulsar examples. These examples have all been checked by human annotators with each row listing the variables first, and the class label as the final entry. The class labels used are 0(negative) and 1(positive). Each candidate is described by 8 continuous variables and a single class variable. The first four are simple statistics obtained from the integrated pulse profile. This is an array of continuous variables that describe a longitude-resolved version of the signal that has been averaged in both time and frequency. The remaining four variables are similarly obtained

from the DM-SNR curve. All these features assist identify and classify the pulsar signals from other types of signals and noise based on their unique characteristics.

Features

1. Mean of the integrated profile: an average of pulsar's emission over one rotation period.

2. Standard deviation of the integrated profile: spread of values of integrated pulse profile

3. Excess kurtosis of the integrated profile: a statistical measure to describe the shape of the distribution which measures the tailedness of the distribution compared to a normal distribution

4. Skewness of the integrated profile: the asymmetric distribution of the values in the integrated profile.

5. Mean of the DM-SNR curve: average value of the dispersion measure versus signal to noise ratio curve of a pulsar where dispersion measure is the amount of free electrons in line of sight of the pulsar

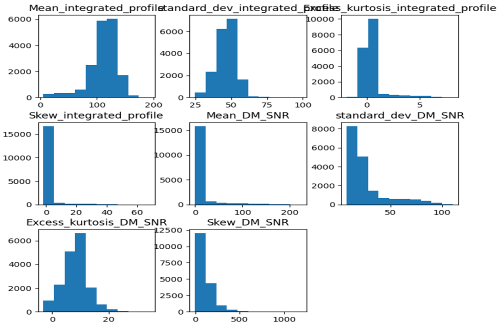
6. Standard deviation of the DM-SNR curve: spread of the DM-SMR curve.

7. Excess kurtosis of the DM-SNR curve: measures tailedness distributed value in the DM-SNR curve compared to a normal distribution

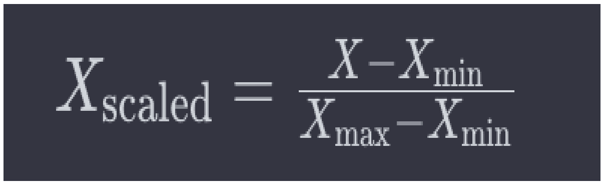
8. Skewness of the DM-SNR curve: the asymmetry distribution of values in the DM-SNR curve.

Analysis

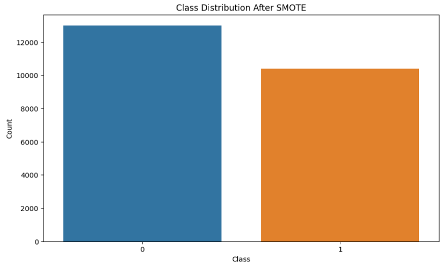
- *Feature engineering and Model Selection*

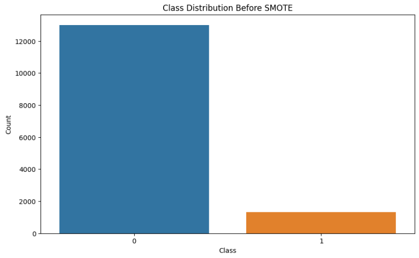


Above histogram gave the basic idea about the model with some features having the negative values with different ranges. So, I chose the MinMax Scaling for normalization which scales the data to a specific range and maps the minimum value to 0 and maximum value to 1. The formula for normalization is:

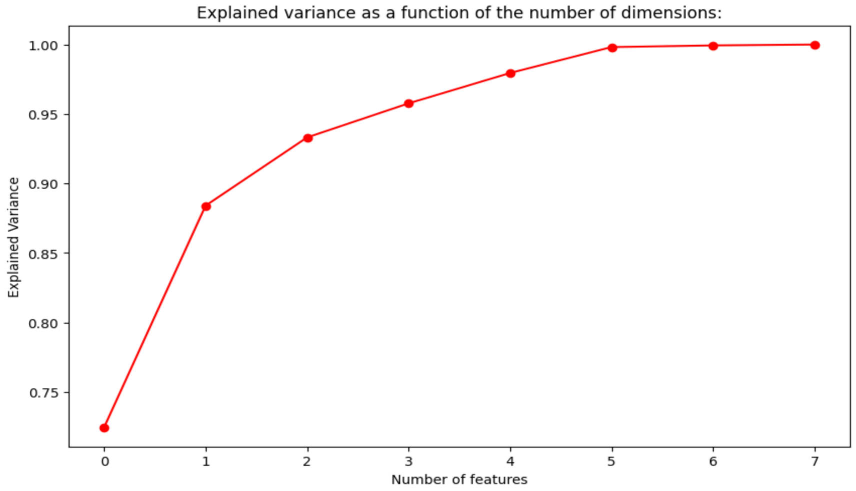


$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$





There was a huge imbalance between two class distributions 0 and 1 with 16259 and 1639 respectively. So, I generated synthetic examples of the minority class samples to 80% of majority class samples using SMOTE. The difference is the bar diagram above pre-SMOTE(right) and post-SMOTE(left). There were 23400 datasets including the synthetic classes.



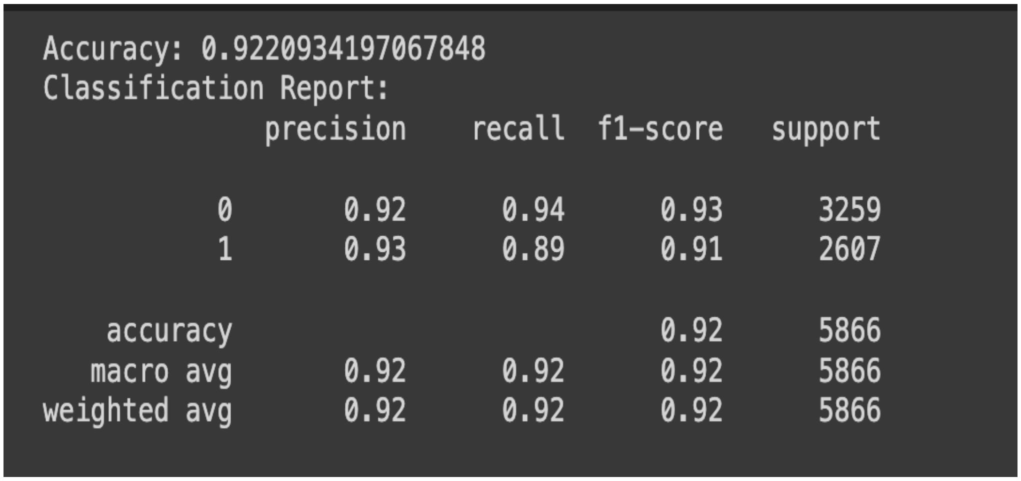
PCA was used for reducing the dimensionality of the datasets while preserving the important information to visualize the data and improve the efficiency of models. I chose an explained variance of 0.9. Hence, it decided on 3 components to keep which are 'Mean integrated profile', ‘standard deviation integrated profile’ 'Excess kurtosis integrated profile'.

After these steps of feature engineering, the datasets went through the pipeline to ensure that data is scaled and dimensionally reduced appropriately for subsequent machine learning tasks.

- *Testing Model with scikit learning*

Since, its binary classification, and to start with some easier models, utilize the GridSearchCV for hyperparameter tuning in machine learning which performs an exhaustive search over a specified parameter grid, evaluating each combination of hyperparameters using cross validation.

  - *Random Forest Classifier*

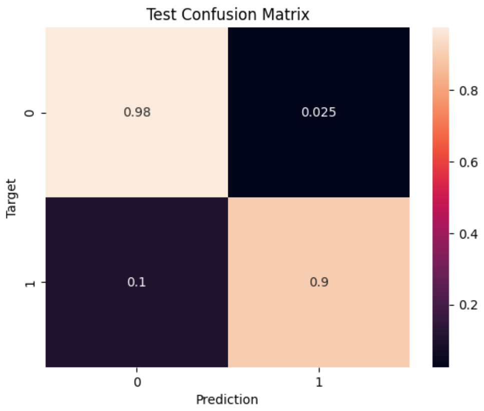


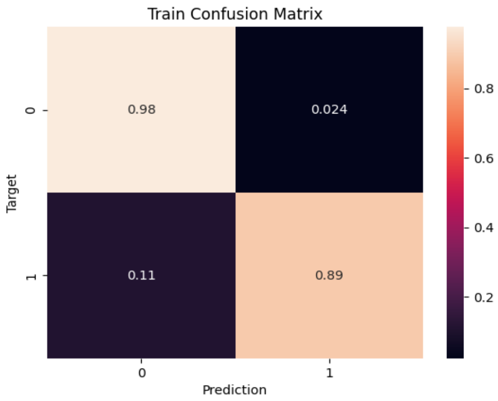
```
Accuracy: 0.9220934197067848
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.94      0.93      3259
           1       0.93      0.89      0.91      2607

    accuracy                           0.92      5866
   macro avg       0.92      0.92      0.92      5866
weighted avg       0.92      0.92      0.92      5866
```

Random forest classifier had an accuracy of 92.2% with precision of 0.92 for class 0 and 0.93 for class 1, the recall for class 0 is 0.94 and for class 1 is 0.89. The model shows high precision, recall, and F1-score indicating strong performance for classification.Good F1 score indicates a good balance between precision and recall.
`Best Parameters: {'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}`

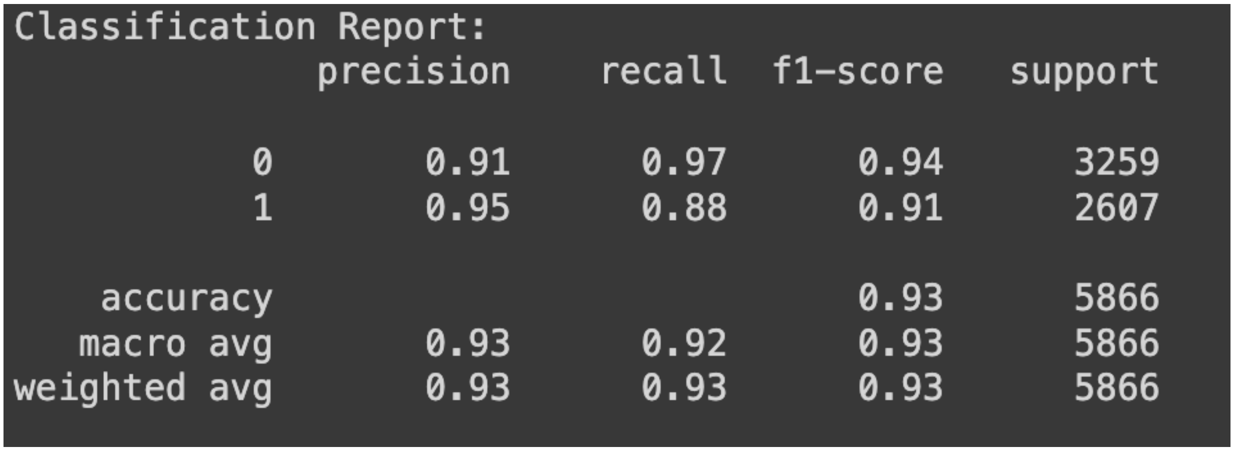
  - *SVM*

SVM also had an overall accuracy score 94.12%. The test and train confusion matrix shows overall correctness of the model is high with 0.96 true negative and 0.9 true positive for true negative. The training model also shows it has a similar score implying the data were perfectly fitted and the model is performing nicely.
The best model for hyperparameter parameter is `Best Parameters: {'C': 100, 'gamma': 1, 'kernel': 'rbf'}`



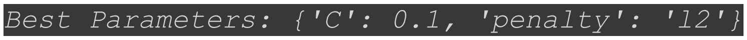


- *Logistic regression*

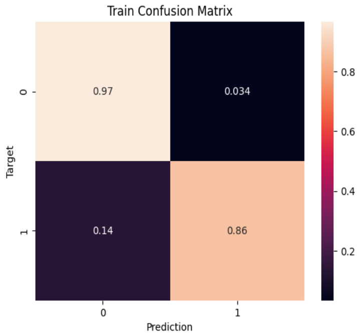


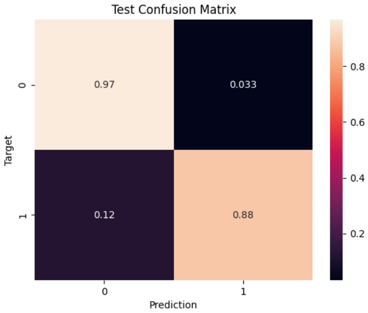
```
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      3259
           1       0.95      0.88      0.91      2607

    accuracy                           0.93      5866
   macro avg       0.93      0.92      0.93      5866
weighted avg       0.93      0.93      0.93      5866
```

The accuracy of 93% suggests that the model is effective in making correct predictions. The precision score for class 0 is 0.91 and for class 1 is 0.95. The recall score was 0.97 for class 0 and 0.88 for class 1. The F1 score score was also high implying the performances were consistent and performing well in the classification.
The confusion matrix diagram below also helps satisfy the point and make sure that data aren't overfitting.

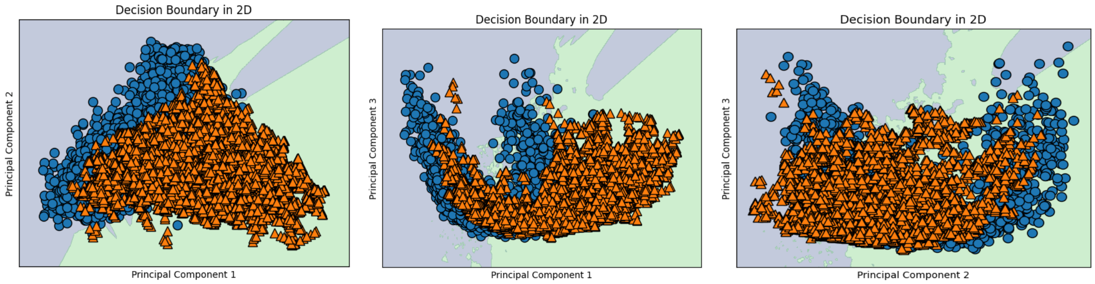


```
Best Parameters: {'C': 0.1, 'penalty': 'l2'}
```

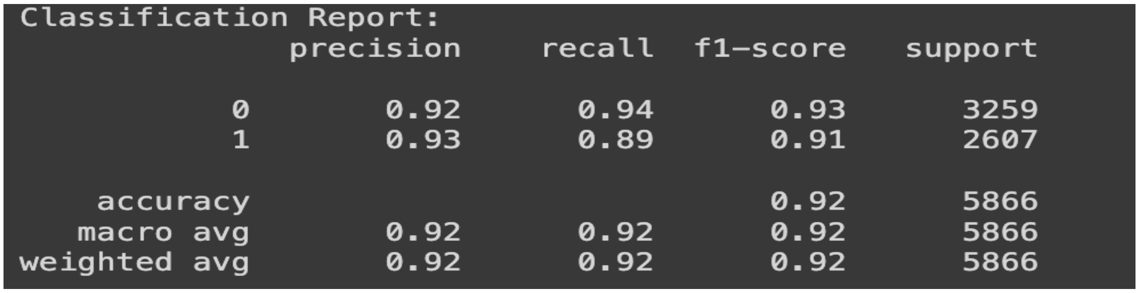




- ➢ *KNN*



Above diagram shows the decision boundary for k= 3 neighbors between three selected principal components. Except for components 2 and 3(rightmost), the plots are scattered and have a nice decision boundary.



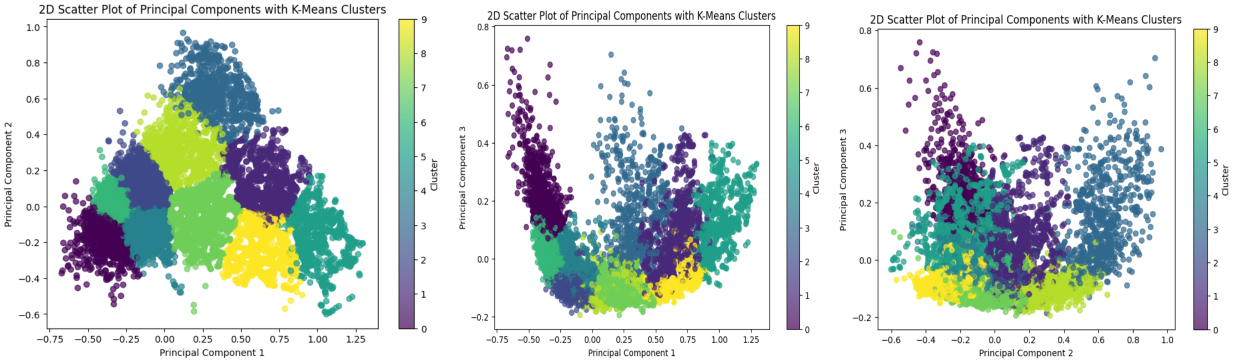
```
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.94      0.93      3259
           1       0.93      0.89      0.91      2607

    accuracy                           0.92      5866
   macro avg       0.92      0.92      0.92      5866
weighted avg       0.92      0.92      0.92      5866
```

Based on the classification report, the precision score and recall for class 0 was 0.92 and 0.94 respectively whereas for class 1 was 0.93 and 0.89. The F1-scores are also high indicating a good balance between precision and recall for both classes. This model also showing consistent performance.

- ❖ *Classifier*

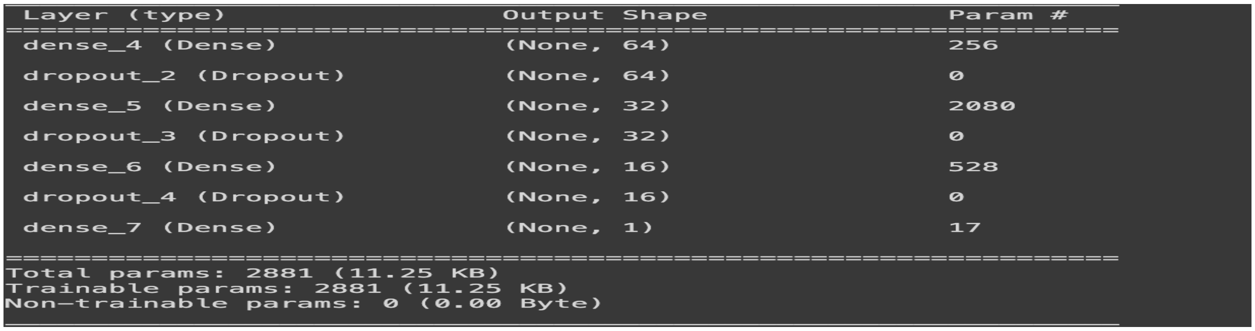
- ➢ *K-Means*

Kmeans had the silhouette score of 0.67 which measures how similar one cluster is to its own. This score is close to 1, which means that clusters are well-defined and objects within are similar. Hence, implying a positive indication of the quality of the cluster. To determine the optimal number of clusters in a dataset, the elbow method was used. We obtained the optimal value k =2, a point where the rate of sum of squared distance between data points and their centroids is minimum. Since I got the positive result, I didn't go through 'DBSCAN' for classification.



Above are 2d diagrams of the cluster between 3 principal components.

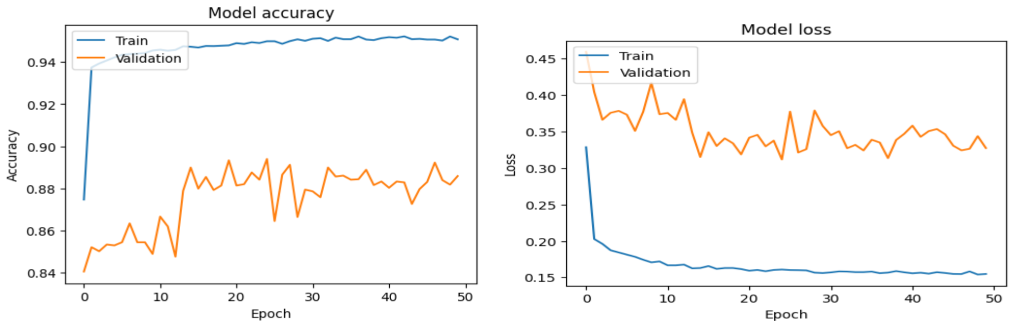
- *Testing Model Neural Network*
  - *Feed forward Neural Network*



```
 Layer (type)                Output Shape              Param #
=================================================================
 dense_4 (Dense)             (None, 64)                256

 dropout_2 (Dropout)         (None, 64)                0

 dense_5 (Dense)             (None, 32)                2080

 dropout_3 (Dropout)         (None, 32)                0

 dense_6 (Dense)             (None, 16)                528

 dropout_4 (Dropout)         (None, 16)                0

 dense_7 (Dense)             (None, 1)                 17

=================================================================
Total params: 2881 (11.25 KB)
Trainable params: 2881 (11.25 KB)
Non-trainable params: 0 (0.00 Byte)
```

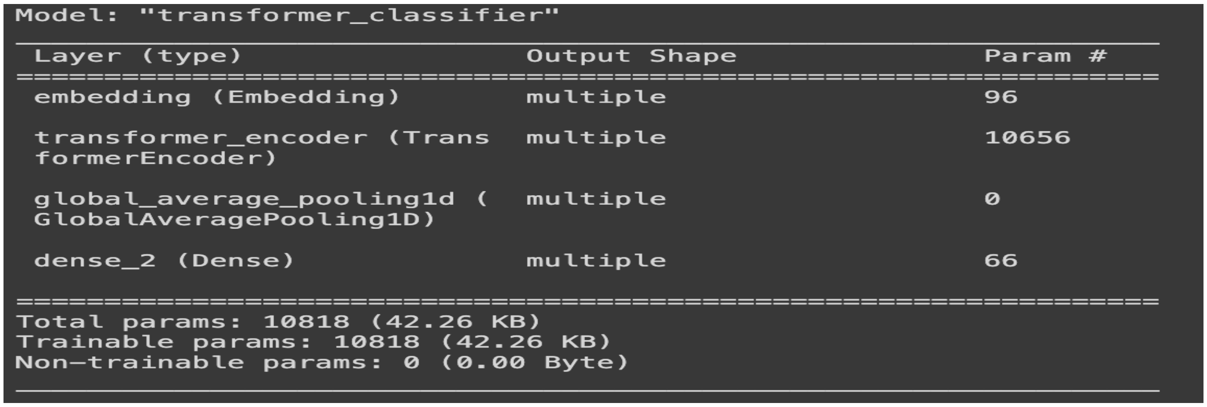
Above is the model summary of a sequential neural network with multiple dense layers with dropout layers for regularization and a final dense layer with a sigmoid activation function for binary classification. The accuracy of 94% indicating the data was performing well on the test set. The precision for class 0 was 0.92 and precision for class 1 was 0.97. The recall for class 0 was 0.98 and class 1 was 0.9. High precision and recall indicates good performance in distinguishing between positive and negative instances.

The F1-score for class 0 was 0.95 and class 1 was 0.93 which was the harmonic mean of precision and recall. F1-score is high indicating a good balance between precision and recall.



Above diagram provides the information about validation and train accuracy along 50 epochs

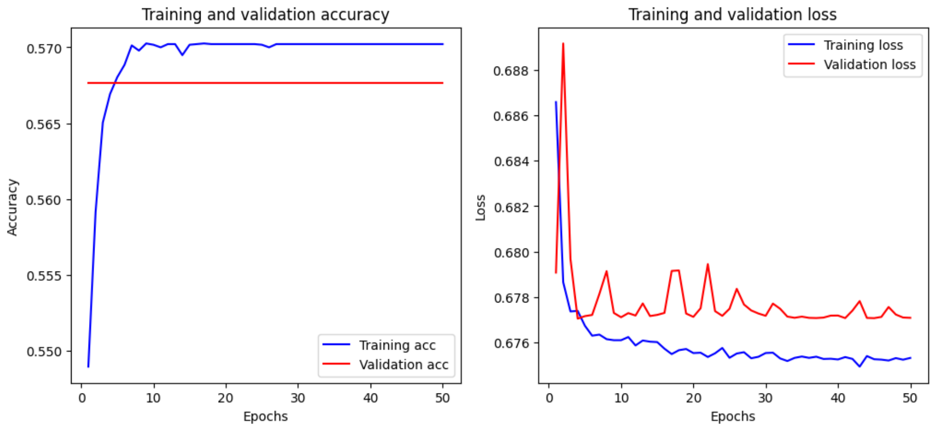
- ➢ *Transformer*



```
Model: "transformer_classifier"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       multiple                  96

 transformer_encoder (Trans  multiple                  10656
 formerEncoder)

 global_average_pooling1d (  multiple                  0
 GlobalAveragePooling1D)

 dense_2 (Dense)             multiple                  66

=================================================================
Total params: 10818 (42.26 KB)
Trainable params: 10818 (42.26 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

All the above 5 models gave good scores, but I was curious how the transformer would react to the pulsars data. The results were surprising where I got an accuracy score of just 0.5656.

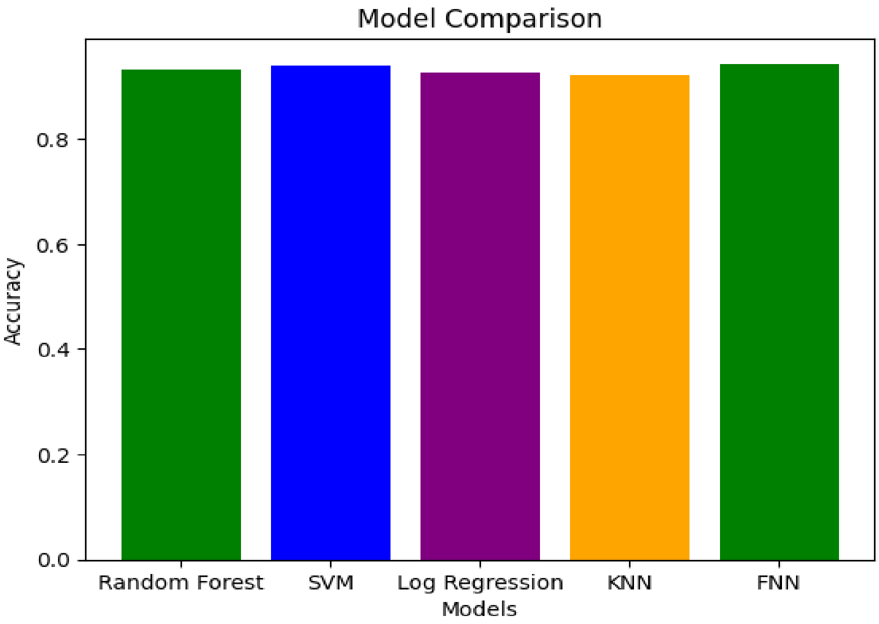
The precision for class 0 was 0.56 while for class 1 was 1.0. The recall score for recall was 1.0 and recall score was 0.02. For class 0, the model shows decent precision, recall, and F1-score which indicates that it is effective in identifying instances of the class. However, recall and f1-score was very low, suggesting that model misses a significant number of positive instances and has problems in correctly classifying them. The overall accuracy of 57% shows suboptimal overall model performance. Hyperparameter tuning and class imbalance might have affected the transformer based classifier.



The diagram shows that the custom model ran for 50 epochs with the average of 0.6833 validation accuracy. Training accuracy score was also similar to the validation score just around 0.6877. Hence, less likely of underfitting the model.
Below is the summary of the transformer model

Conclusion

In conclusion, the analysis of the pulsar dataset has provided valuable insights into the task of identifying pulsars with a comprehensive approach of data preprocessing, feature engineering and evaluating different models. Feed Forward Neural network was the top performer with an accuracy score of 94%, other model also demonstrated competitive performance of 92 and 93.



The project addressed the issue of class imbalance using SMOTE but an alternative approach technique could be considered for improving the performance. Explorations of ensemble techniques and experimenting with more advanced neural network architectures can enhance *model interpretability and performance. Hence, the insights and the methodologies employed provide a roadmap for future enhancements and explorations.*

Sources

Lyon,Robert.(2017).HTRU2.UCI Machine LearningRepository.https://doi.org/10.24432/C5DK6R.

[2]

2 Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MITPress.

[3]

3 Encyclopædia Britannica, inc. (2023, September 1). Pulsar. Encyclopædia Britannica. https://www.britannica.com/science/pulsar

[4]

4 Htru. HTRU — Max Planck Institute for Radio Astronomy. (n.d.). https://www.mpifr-bonn.mpg.de/research/fundamental/htru