Deliverable 2

Project Idea:

I would like to create a web app that the astronomers can use to predict the existence of a pulsar star given the 8 features. They could use the prediction to further investigate and confirm the discovery of the star.
Link to the coding of the project:
https://colab.research.google.com/drive/1x6hevsBIp60kVpYxgwDZx2ZVssrSCaiQ

Data Preprocessing:

Dataset: UCI Machine Learning Repo - https://archive.ics.uci.edu/ml/datasets/HTRU2[1]
Reasons for choosing this:

- It contains 17,898 instances and 1,639 are real pulsar examples. They have been checked by human annotators.
- The data contains 9 variables (8 features and 1 label).
- The variables are:
    1. Mean of the integrated profile
    2. Standard deviation of the integrated profile
    3. Excess kurtosis of the integrated profile
    4. Skewness of the integrated profile
    5. Mean of the DM-SNR curve
    6. Standard deviation of the DM-SNR curve
    7. Excess kurtosis of the DM-SNR curve
    8. Skewness of the DM-SNR curve
    9. Class
- The values had to be converted to their respected data types (floats and integers) before inputting them to the model
- I searched for missing values and tried to fix them using sklearn's impute module

Machine Learning Model:

- I used the scikit-learn framework for all of the models I tried out.
- I used model-selection to split the dataset into train and test, and for cross validation grid search.
- I used metrics module to get the accuracy score, confusion matrix and classification report.
- I had decided to do a train:test split of 90:10 so that the model gets a good amount of data to learn from.

[1] R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, J. D. Knowles, Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach, Monthly Notices of the Royal Astronomical Society 459 (1), 1104-1123, DOI: 10.1093/mnras/stw656

[1] R. J. Lyon, HTRU2, DOI: 10.6084/m9.figshare.3080389.v1.

- Since the dataset is small, it gives me the flexibility to try out various classification models to find the model which suits best based on the classification report produced on the test set. I plan to test out KNN, Logistic Regression, Random Forest Classifier, Support Vector Machines classifier and Decision Tree classifier
- Again, as the dataset is small, I have decided to do a cross validation through grid search and try various hyperparameters to fine-tune the models to get the best results. (I have cross validated and found the best hyperparameters for some of the models and plan to do the rest soon, as it is time consuming)
- The models seem to be overfitting as it is visible through the confusion matrix, where there are more incorrect predictions for sample points which are stars.
- Finding the best hyperparameters is quite the struggle, and I am still working on finding them. I plan on picking random hyperparameters using RandomizedSearchCV.

Preliminary results:

- It seems like RandomForestClassifier performs the best amongst the models as seen from the 97.9% accuracy on the test cases and a relatively low false predictions of not a star cases, as seen when run the code.

Next steps:

- My current approach is hugely time consuming, mainly because I have to fine-tune each model before choosing the best model to work with. However, this gives the most reliable model to work with, which is a huge bonus.
- I plan on getting started on a web app as soon as the hyperparameter tuning is done.