ML Project : Classification of Stars and Quasars

Raunuk Sengupta, Dev Bhartra, Yash Pradhan
PES1201700072, PES1201700186, PES1201700272

Section 5G, PES University

1 Abstract

Through this project, we aim to apply the principles of Machine Learning in the field of Astronomy, and come up with a classification algorithm for differentiating between Stars and Quasars - two bright and distant heavenly bodies in the night sky. Just based on photo metric data, these can be hard to distinguish between, and this is where Machine Learning comes into the picture. We show how we decided on a suitable algorithm for this task (k-Nearest Neighbours in our case), and how we made some improvements to tweak it for our requirements. We were also able to incorporate multi-core parallelism in our project, cutting the run time of the program by a significant amount in some scenarios. Finally, we compare our results and accuracy with other models and variations, and come up with some interesting conclusions.

2 Problem Statement

About Quasars
Quasars are the brightest and most distant objects in the known universe - initially known as radio stars, because they were noticed to be a strong source of radio waves. Although not exactly know, most evidence points to the possibility that quasars are produced by super massive black holes consuming matter in an acceleration disk. As the matter spins faster and faster, it heats up. The friction between all of the particles would give off enormous amounts of light other forms of radiation such as x-rays.

Identifying Quasars
Since its discovery in the year 1960, more than 2000 quasars have been identified. Nasa's Hubble Space telescope has been a key tool in the search for quasars. Quasars are primarily identified by their extremely high redshift.
If an object is identified to have an extremely high redshift and is producing vasts amounts

of energy, it is thought to be a prime candidate for being a Quasar.

Challenges in identifying Quasars
It is challenging to distinguish between a star and a quasar just based on photo metric surveys - they both have a compact optical morphology. Hence, we generally require spectroscopic data such as their optical variability or their optical colors to make the distinction. Some other methods for distinguishing between the two include using optical classification schemes that use the optical fluxes of source. Here, using a larger wavelength range such as near-infrared or UV is found to increase efficiency.

Use of Machine Learning
Machine learning is becoming an important technique for characterizing sources in huge astronomy data sets. The ideal method is to just use the spectroscopic data so that the red-shift of quasars (derived from their optical spectra), can be used to differentiate between the two. However, spectroscopic data is not always available, and here is where the real challenge lies.

3 Solution

A note on the Data Set

We are using only optical photo metric data and UV data, which was observed using the e Sloan Digital Sky Survey (SDSS) and the Galaxy Evolution Explorer (GALEX) telescope respectively.

The data is from two regions,
North Galactic Region - From the region greater than 75 deg of galactic latitude
Equatorial Region - From around the equatorial region 30 deg to 30 deg.

In the end, we have 4 catalogs to choose from,
Catalog 1: North Galactic Region Only
Selected only samples that have fuv

Catalog 2: Equatorial Region Only
Selected only samples that have fuv

Catalog 3: North Galactic Region and Equatorial Region Combined
Selected only samples that have fuv

Catalog 4: Removed fuv and fuv-related features

Our approach

We chose to use the k-Nearest Neighbours (kNN) algorithm. kNN is a distance based algorithm which is used to solve both classification and regression problem. It is a non-parametric, lazy learning algorithm and is based on feature similarity object being assigned to the class most common among its k nearest neighbors. kNN is a supervised learning algorithm, which means that it analyzes the training data and produces an inferred function, which can be used for mapping new examples.

We started by performing Principle Component Analysis (PCA) on the data. This reduced the number of dimensions to 2, and we were hence able to understand the effect of bias and variance on different values of k. Based on this, we were able to settle on k=5 as the most optimum value of k.

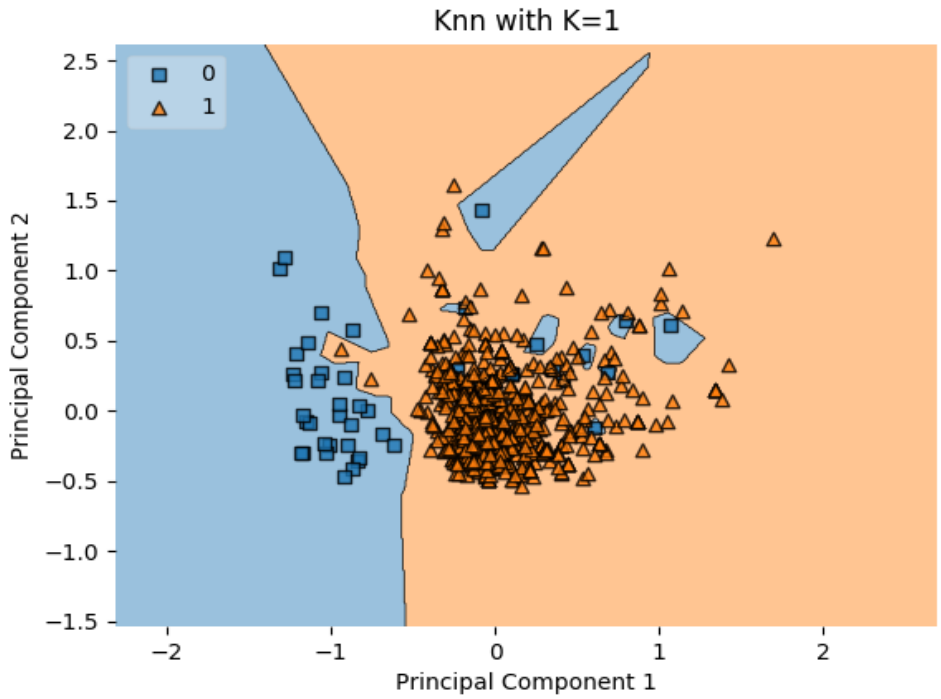

Figure 1: Scatterplot for kNN with k=1, an example of overfitting

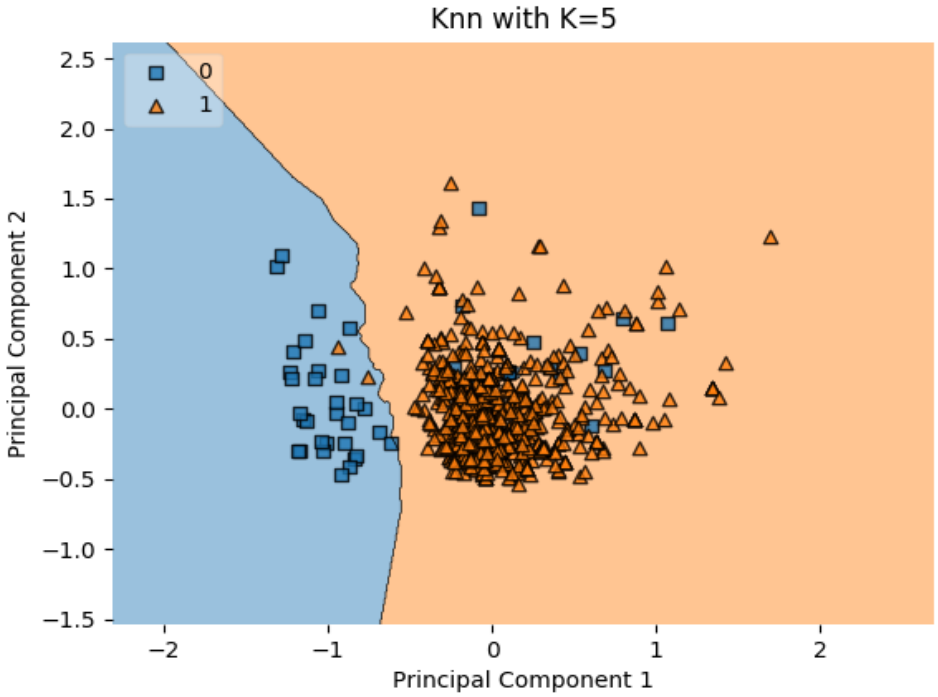

Figure 2: Scatterplot for kNN with k=5, this is the optimal value of k

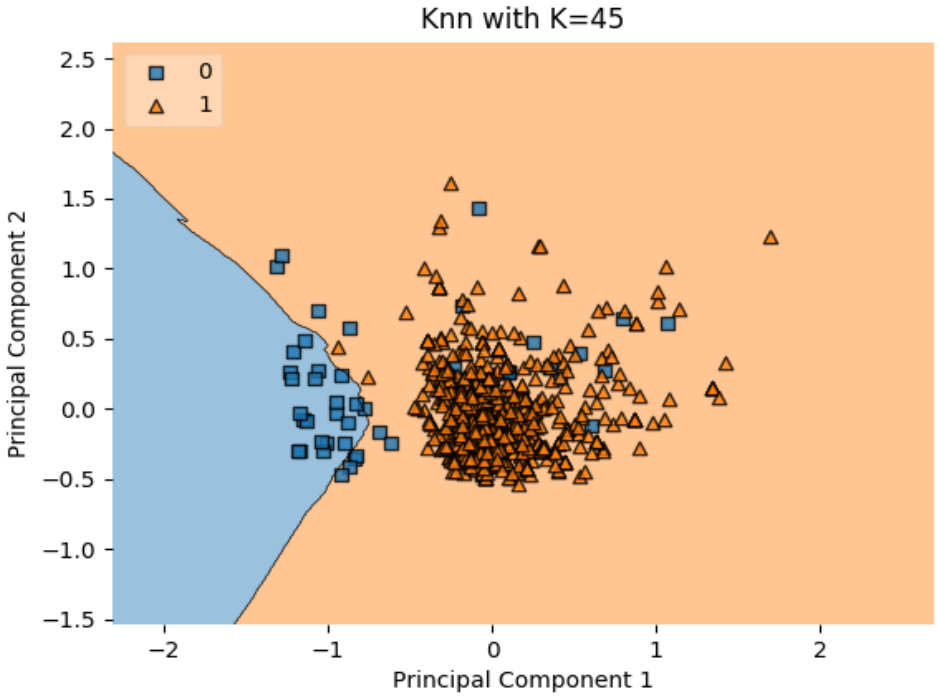

Figure 3: Scatterplot for kNN with k=45, an example of underfitting

4 Unique Selling Proposition : Parallelism in kNN

We implemented multiprocessing for each file in kNN and multi-threading for calculating distance. This was useful in speeding up the execution time of the algorithm. There are some drawbacks of this approach.

- This is a CPU intensive task, thus context switch results in a large overhead.
- Multithreading is good only for I/O bound tasks.

Multiprocessing for both the files and the distance calculation resulted in a deadlock. We ran multiple simulations and tests.

Some of our observations with this parallelism were as follows,

- The speedup offered in serial execution of the file with multithreading in the distance calculation is weighed down by the context switching of the threads.
- With experimental analysis and results, processing the files on which the model runs in parallel gave the maximum efficiency.

This approach has helped us fix two of the main cons of KNN, which are slow speed and expensive computation.

We discuss our exact observations and the speedup offered in the next section.

5 Observations

Through our multiple tests and analysis, we made a number of interesting observations. We show them through the graphs below.

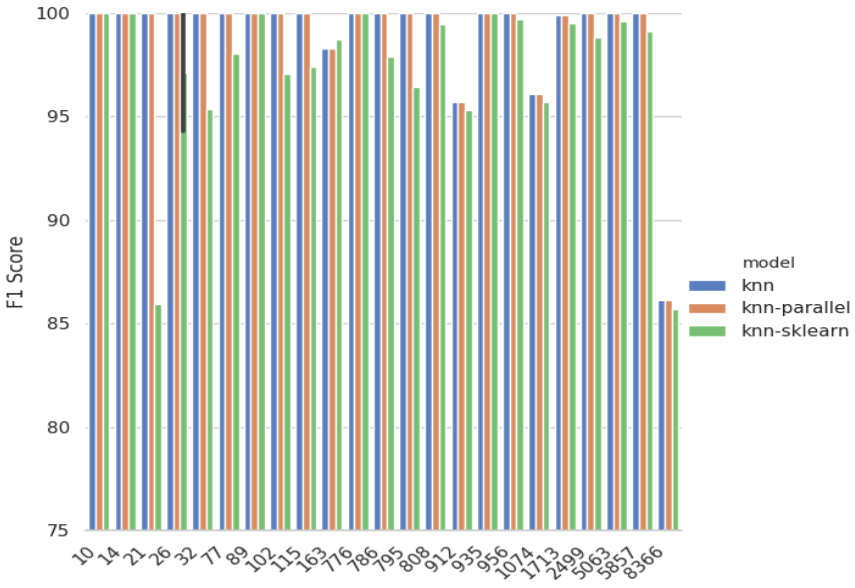

Figure 4: F1 Score for different KNN implementations

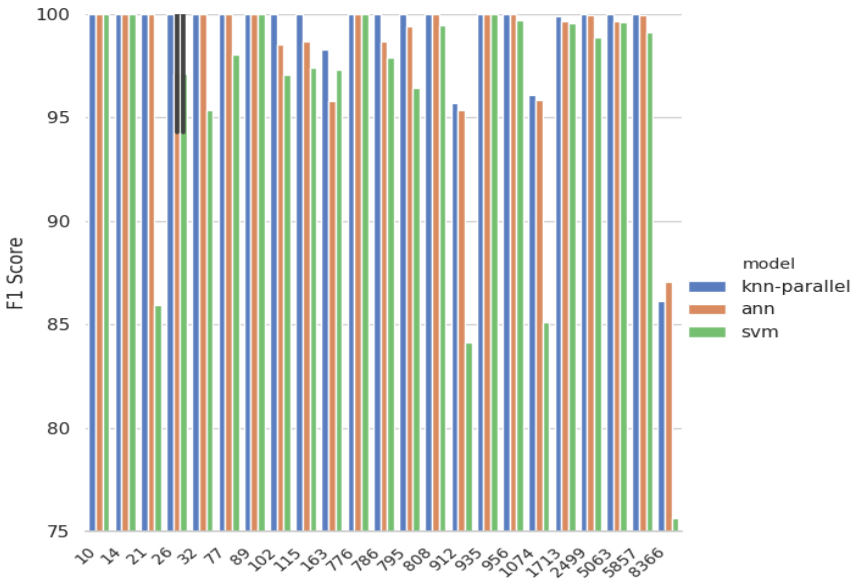

Figure 5: F1 Score across different algorithms

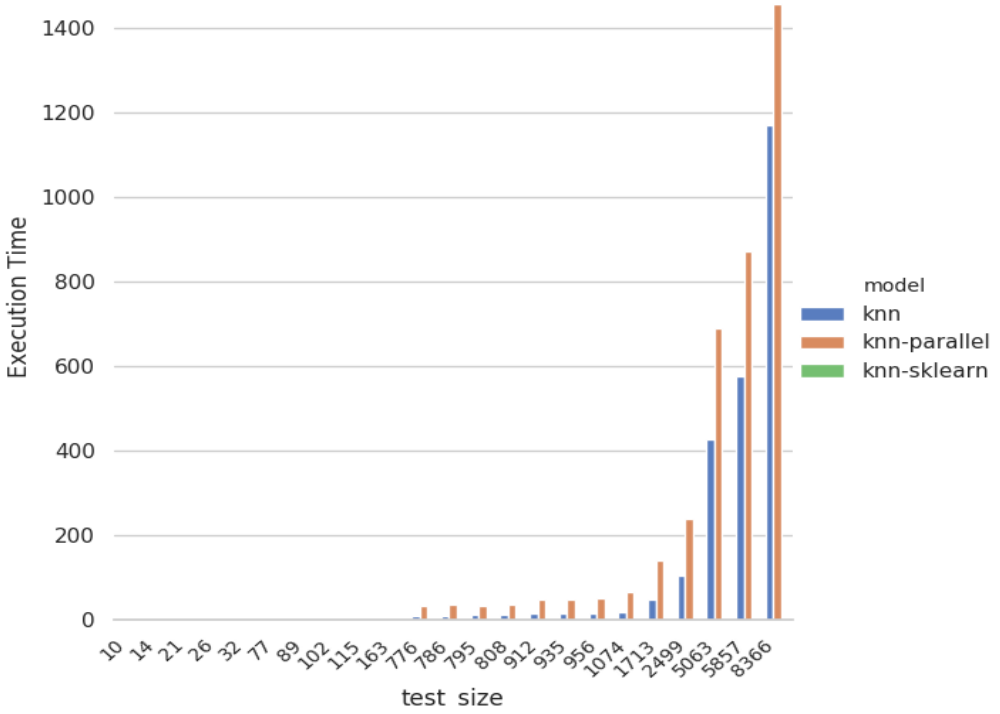

Figure 6: Execution times for differnet KNN implementations

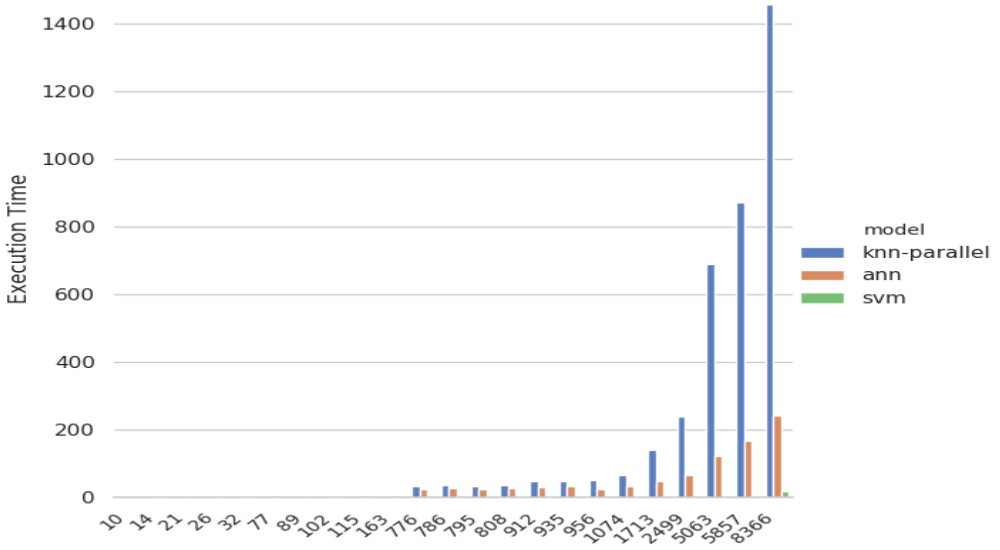

Figure 7: KNN (with parallelis) execution time vs other algorithms

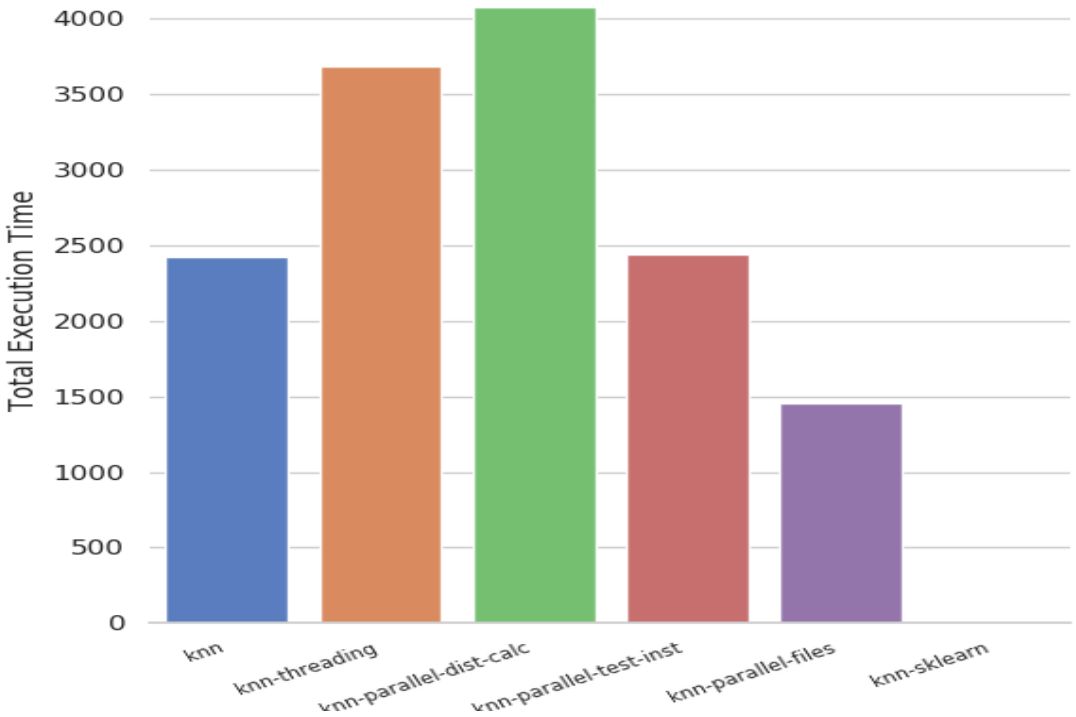

Figure 8: Total execution times for running knn with different files

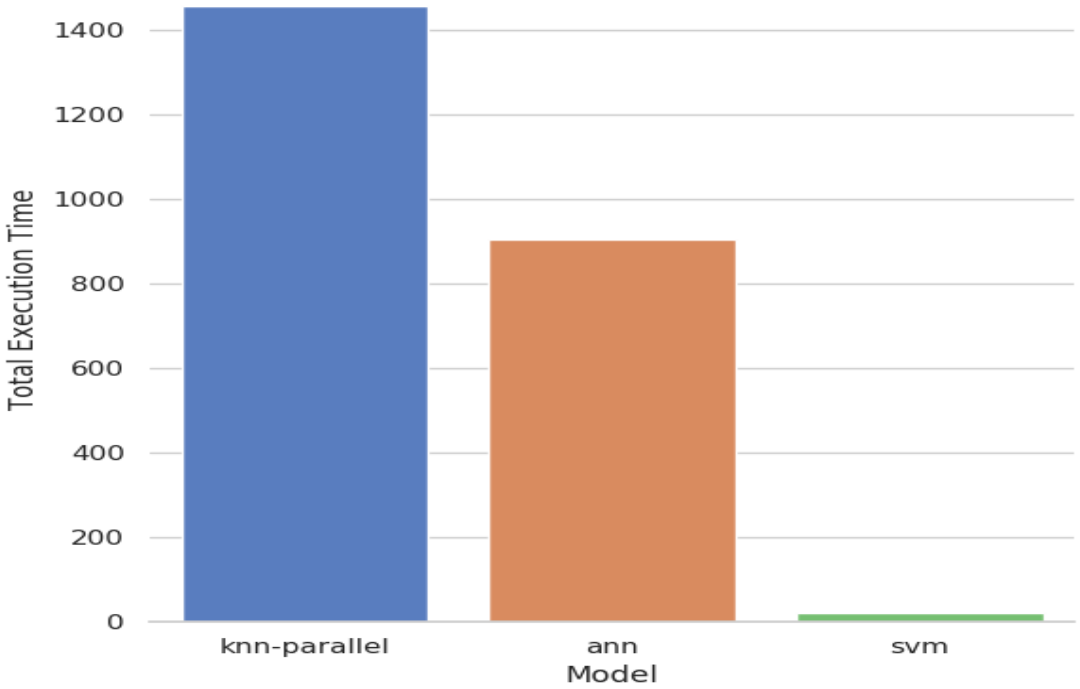

Figure 9: Total execution times for different algorithms

6 Conclusion

This project has provided us with a insight into the power of Machine Learning in fields such as astronomy. We had to extensively research and carry out various analysis and learnt a lot in this process. A few conclusions that we learned as we progressed through this project:

1. Same object is labelled as star and quasar in different catalogs, inferring class labels aren't very reliable.
2. As the class labels may not be reliable, photometric - redshift is a better parameter to use for labelling the data (quasars have a high redshift)
3. The general knn algorithm performs well on linearly inseparable data.
4. KNN can generate a highly convoluted decision boundary as it is driven by the raw training data itself.
5. Inter-body distance is a good measure of classifying photometric data of celestial objects.
6. The drawback of using KNN is that KNN doesnt perform very well on skewed data.
7. KNN requires the training data samples to be a close estimate of the test data that the algorithm encounters.