6685 project proposal

Zac Rios Yingying Deng

Stellar Classification

1.The goal. What problem are you trying to solve? Provide some context and motivation for what you're trying to accomplish. Identify whether your problem is supervised (and if so, classification or regression), unsupervised, or something else.

In the field of astronomy, stellar classification is a fundamental process for categorizing stars according to their distinctive spectral characteristics. This classification system is essential not only for stars but also for galaxies and quasars. The early efforts to catalog stars and map their positions in the night sky played a crucial role in realizing that these stars collectively form our Milky Way galaxy. Furthermore, as astronomical observations progressed and it was established that Andromeda is a distinct galaxy separate from our own, the exploration of numerous other galaxies became possible with the advent of increasingly powerful telescopes.
In this project, we will aim to classify stars, galaxies and quasars based on their spectral characteristics. Our problem is supervised classification.

2.The data. What data will you use to solve this problem? How much data do you think you will end up with? Will data augmentation methods be necessary?

The dataset we will use for this project is a dataset on stellar classification from the kaggle competition.This datasat aims to classificate stars, galaxies, and quasars based on their spectral characteristics.It consists of 100,000 observations of space taken by the SDSS (Sloan Digital Sky Survey). Every observation is described by 17 feature columns and 1 class column which identifies it to be either a star, galaxy or quasar.The link to the data is shown below.Stellar Classification Dataset - SDSS17 (kaggle.com). Data augmentation is likely unnecessary for this particular problem, as we have a high number of observations relative to our features.

3.The methods. What neural network architectures do you plan to use to solve the problem? Also, what other machine learning algorithms (i.e. non-neural network approaches) will you compare to for a baseline? Note that comparisons to non-neural network methods (that you carry out) are required otherwise you will lose points in the final report and (possibly) the presentation.

When it comes to neural network architectures, we will likely be primarily using Artificial Neural Networks. We will work to find an optimal learning rate, regularization parameter, batch size, number of hidden layers, and hidden layer sizes. There is a possibility that other types of neural networks would be helpful, but our data is already quite simple and well-represented.
Non-Neural Network Approaches (Baseline Comparisons):
Decision Trees and Random Forests: These methods can handle both classification and regression tasks, making them versatile for baseline comparisons.

Support Vector Machines (SVM): SVMs are effective for classification tasks and can work well for complex datasets.
k-Nearest Neighbors (k-NN): k-NN algorithms are simple yet effective for classification tasks, making them useful for initial comparisons.
Gradient Boosting Machines (e.g., XGBoost, LightGBM): Gradient boosting algorithms can capture complex relationships in data and often perform well in classification tasks.

4.Potential issues. What are some potential problems you may run into? What are some alternate strategies you could employ if these issues arise?

Class Imbalance: If our classes are imbalanced, it might skew the learning of our model. We could use techniques like oversampling, undersampling, or using class weights during training that can help mitigate this issue.
Overfitting: Deep learning models, especially with a large number of parameters, can easily overfit the data. We will primarily use L1/L2 regularization to overcome this issue. We could also implement a stopping criterion with our data.
Interpretable Models: Neural networks, especially complex ones, are often seen as "black boxes". If interpretability is important, we could emphasize simpler models like decision trees for comparison.
Algorithm Selection: Experiment with different algorithms and model architectures. It's often hard to predict which one will perform best for a specific problem, so empirical testing and validation are crucial. Consider techniques like cross-validation to get a better estimate of our models' performance. It’s very possible that neural networks will not be able to outperform random forests or gradient boosting.

5.Project management and planned contributions of each member. How do you plan to communicate and work together as a group?

In order to complete this project, we will meet weekly at 3:00pm on Friday. Each meeting will consist of establishing baselines (both in terms of other ML methods and our old ANN), and continuing to update our ANN.
10/23 - Proposal due
- Get the data
- complete the proposal

10/27 Group meeting
- Data engineering (Zac and Yingying)

11/3 Group Meeting
- Machine Learning Algorithms
- Decision Trees/Random Forest (Zac)
- KNN (Yingying)

11/10 - Group Meeting
- Multinomial Logistic Regression(Zac)
- Multilayer Perceptron (Yingying)

11/17 Group Meeting
- Random Forest (Yingying)
- Support Vector Machines (Zac)

- Soft Voting Classifier(Yingying)

11/24 Final Group Meeting
- Write report
- Presentation

12/? - Project Presentations