Stellar Classification

Bailee Hodge, Levi Homer, Spencer Neu, Elijah Ryan, and Garrett Suggs
Department of Computer Science
Brigham Young University

Abstract

Stellar classification is a critical task in astrophysics, involving the categorization of stars based on a variety of properties. With the advent of large-scale astronomical datasets, there is a growing need for automated systems to efficiently classify stars. This paper explores the use of machine learning models for classifying stars into spectral classes based on their observed features. We focused on two models: a Multi-Layer Perceptron (MLP) and a K-Nearest Neighbor (KNN) classifier, both trained on the Simbad database. We conducted extensive feature selection to identify the most relevant stellar properties for classification, including B-V, temperature, and luminosity. Our results show that the KNN model outperforms the MLP in terms of accuracy, but both have the potential to be reliable astronomical tools.

1 Introduction

1.1 Background Information

Stars are generally categorized based on their temperature and their spectral characteristics. The most widely used system for classifying stars is the Harvard spectral classification, which divides stars into seven main classes: O, B, A, F, G, K, and M. These classes are arranged in order of decreasing temperature, with O-type stars being the hottest and M-type stars being the coolest. Each of these main classes is further subdivided into 9 subclasses (e.g., B0, B1, B2...), which provide more specific information about the star's characteristics.

Most stars, however, do not fall randomly across the classification spectrum but instead primarily reside along a region known as the "main sequence" of the Hertzsprung-Russell (H-R) diagram, which is a plot of luminosity versus temperature. This main sequence represents a stable phase in a star's lifecycle, where stars spend the majority of their time fusing hydrogen into helium in their cores. The position of a star on the main sequence is determined by its mass, with more massive stars being hotter and more luminous, found toward the upper-left of the diagram, while less massive stars are cooler and less luminous, located toward the lower-right. As seen in figure 1, this main sequence forms a diagonal band stretching from the upper-left to the lower-right of the H-R diagram, which is a key feature of stellar evolution. Stars that are not on the main sequence are in different stages of their lifecycle, such as red giants or white dwarfs.

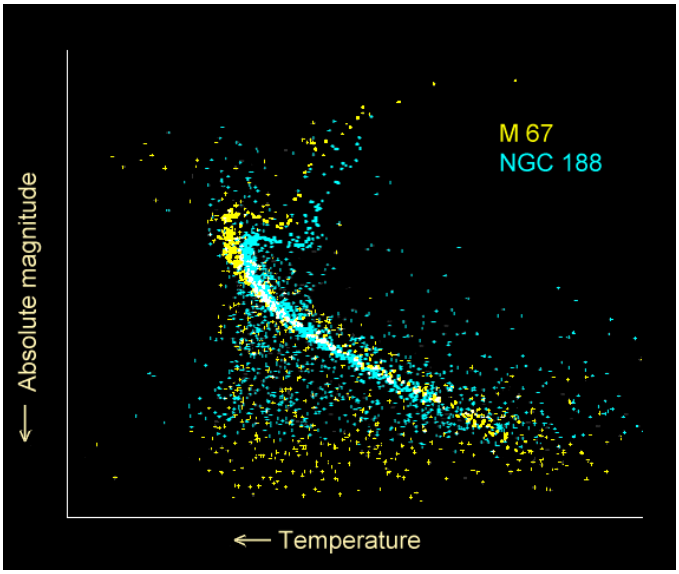

Figure 1: Spectal Class Graph

The classification system is based not only on the temperature but also on the absorption lines in the star's spectrum, which are caused by the elements and molecules present in the star's atmosphere. As a result, stars within the same class tend to exhibit similar colors, with hotter stars appearing blue and cooler ones appearing red. This classification system has been fundamental in stellar astronomy, allowing scientists to understand the physical properties and evolution of stars. Although there are several other factors, temperature is the primary classification metric.

1.2 Main Goal of the Project

The primary goal of this project was to explore whether the classification of stars can be determined en masse by a machine learning model without individual evaluation. This is important because of the sheer number of stars that are observed, recorded, and studied by astronomers. By building a model that can accurately and automatically classify stars, we

can streamline the process of stellar classification, especially for large-scale datasets where spectral class is unrecorded or undetermined. All of the stellar datasets that our team looked into had data points with missing features. This model could mean that spectral class won't be one of them.

2 Methods

2.1 Data Source

We used the Simbad (Set of Identifications, Measurements, and Bibliography for Astronomical Data) dataset for our project. The data in the Simbad database comes from a variety of astronomical surveys, observations, and scientific literature. The data is collected, curated, and organized by the Centre de Données astronomiques de Strasbourg (CDS), a French institution that provides public access to a variety of astronomical resources.

This dataset has millions of total data entries. Approximately 50,000 of the entries are stars labeled with the spectral classification; approximately 12,000 are unlabeled stars. The set has both spectral data for and the actual classification of each star. It also includes other key stellar properties such as distance, luminosity, and motion. Researchers worldwide use Simbad as a reliable source of astronomical information. [Database, 2024]

We considered other datasets, but decided against them for the sake of retrieval time. The Gaia dataset has approximately 1 billion total instances. Although more data is usually better, pulling the data from the database would have taken approximately 100 hours of computing time. The SDSS (Sloan Digital Sky Survey) dataset presented a similar problem: it would have taken about 400 hours to pull all of the data. Pulling only part of the data in either case would have taken less time and given us enough to work with, but it would have been difficult to guarantee that the subset was randomly selected. Additionally, the Simbad dataset had the aforementioned key stellar properties, as well as features and organization better suited to our task.

Using multiple datasets was considered, but the idea was rejected because of the potential for variance between astronomical instruments.

2.2 Data Preparation and Feature Selection

Features related to spectral measurements and classification are most relevant to our research question. We examined the list manually and threw out irrelevant features. This left us with 9 features: B, V, B-V, Temperature, radius, luminosity, metallicity, surface gravity, and mass. To eliminate some members of this subset, we trained a Decision Tree on the data and kept only the most significant features as determined by prominence in the tree. One of these features was B-V.

B-V, was manufactured by us from two of the existing features. As the moniker suggests, it is obtained by subtracting the Violet value from the Blue value. This combination produces a feature with an approximation of the spectral curve of the star; this combined feature is more powerful than either of the features on their own.

2.3 Model selection

As stated, we used a Decision Tree model to reduce the number of features to work with. For classification, we chose to use an MLP (Multi-Layer Perceptron) and a KNN (K-Nearest Neighbor) classifier. We chose the MLP because it is generally good at learning tasks. The downside to this model is that, due to the nature of the model, it will not yield any insights into which features were the most important. We chose to use it anyway; knowing the importance of each feature is less critical because of the work of this nature that we did previously with the Decision Tree. We chose the KNN because it fits our use case well. The KNN is particularly useful for classifying data with blurred lines between groups. Because stellar classification is based on multiple factors, our dataset has this attribute. Additionally, stellar classification is based on selection of clusters, and so is the KNN algorithm.

3 Initial Outcomes

3.1 MLP

We first trained our MLP model with the following parameters: 256 hidden nodes, ReLU as the activation function, a learning rate of 0.005, a momentum value of 0.9, Nesterov's momentum set to false, 50 as the maximum number of iterations with no hange, 2000 as the maximum iterations under any circumstances, early stopping set to false, a validation fraction value of 0.15, and with shuffle enabled. These parameters were chosen based on the known attributes of the dataset and the results of a run with multiple default parameters. Figure 2 shows the loss curve for this model, and table 1 shows the accuracy. Under these parameters, the MLP ran until the maximum number of epochs without reaching convergence. Additionally, the loss value never truly dipped below a value of 1.0.

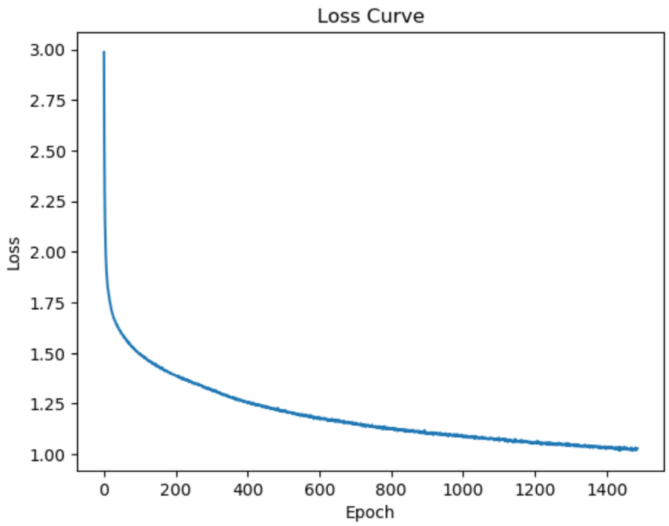

Figure 2: Initial MLP Loss Curve

Training Accuracy	Test Accuracy	Epochs to Converge
68.8%	67.3%	2000

Table 1: Initial MLP Accuracy

3.2 KNN

We first trained our KNN model with all default parameters, including a k value of 5. Surprisingly, this produced favorable results: specifically, a test accuracy of approximately 96% (see table 2). Limited testing of different k values resulted in a slightly increased accuracy when k is equal to 3.

Training Accuracy	Test Accuracy	k-value
98.1%	96.2%	5

Table 2: Final KNN Accuracy

4 Model and Feature Improvement

4.1 Measuring Improvement

Although training accuracy is an interesting metric, throughout our model refinement process, we relied on testing accuracy as our primary indicator of the success of a model. Secondary factors included runtime, number of epoch until conversion (in the case of the MLP), and graphs of loss curves and classifications. The most significant and relevant graphs and metrics are included in this report.

4.2 Changes to Parameters

The MLP parameters were studied through a series of training runs of the model with various combinations. The most successful of these combinations is described in section 5.1. We ran a grid search to determine the most efficient KNN parameters for our purposes. The grid search took approximately 4 hours to run. The chosen parameters are detailed in section 5.2.

5 Final Results

5.1 MLP

Our final MLP used the following parameters: 4 layers of hidden nodes distributed thus [256, 256, 128, 64], ReLU as the activation function, a learning rate of 0.005, a momentum value of 0.1, Nesterov's momentum set to true, 50 as the maximum number of iterations with no change, 5000 as the maximum iterations under any circumstances, early stopping set to false, a validation fraction value of 0.15, and with shuffle enabled. These parameters were the result of many tests of many different parameter combinations.

Figure 3 shows the loss curve for this model, and table 3 shows the accuracy. Under these parameters, the MLP converged after just 499 epochs, a number well below that of our initial MLP model. Additionally, the loss was significantly lower; while previously it never dipped below 1, for this model, it stayed below 0.5 after approximately 100 epochs had taken place.

These results are accurate enough that this model would be a useful and reliable tool for classifying stars into spectral classes.

Training Accuracy	Test Accuracy	Epochs to Converge
96.0%	95.3%	499

Table 3: Final MLP Accuracy

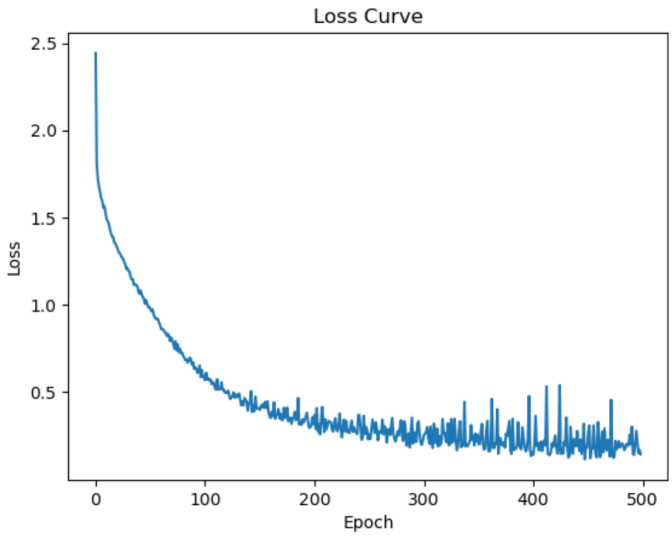

Figure 3: Final MLP Loss Curve

5.2 KNN

Our grid search revealed a set of parameters that gives us approximately 99% accuracy on labeled data. This combination of parameters included setting the algorithm to auto, setting the leaf size to 10, using the manhattan method as the distance metric, using 15 as the k number of neighbors, and using distance weighting. This level of accuracy means that the model does exactly what we trained it to do, and it does so very well.

The KNN algorithm's success could be due to its reliance on distance-based metrics. The KNN uses similarities between data points to form classifications. Stars with similar temperature and luminosity will likely be part of the same (or a similar) spectral class.

Training Accuracy	Test Accuracy	k-value
100%	99.1%	5

Table 4: Final KNN Accuracy

Figure 4 shows a graph of the distribution of the labeled data in the set. As you can see, the stars are distributed along the main curve in the same way that spectral classes are expected to be distributed (compare figure 4 to figure 1).

Figure 5 is a graph of the KNN's predictions on unlabeled data. It is clearly consistent with the labeled data of the set as shown in figure 4. Several classes seem to be underrepresented, but this is consistent with the data's actual distribution. The number of unlabeled instances in our set is far smaller than the number of labeled instances, and the classes with larger mass tend to be fewer in number in both our dataset and in the real world.

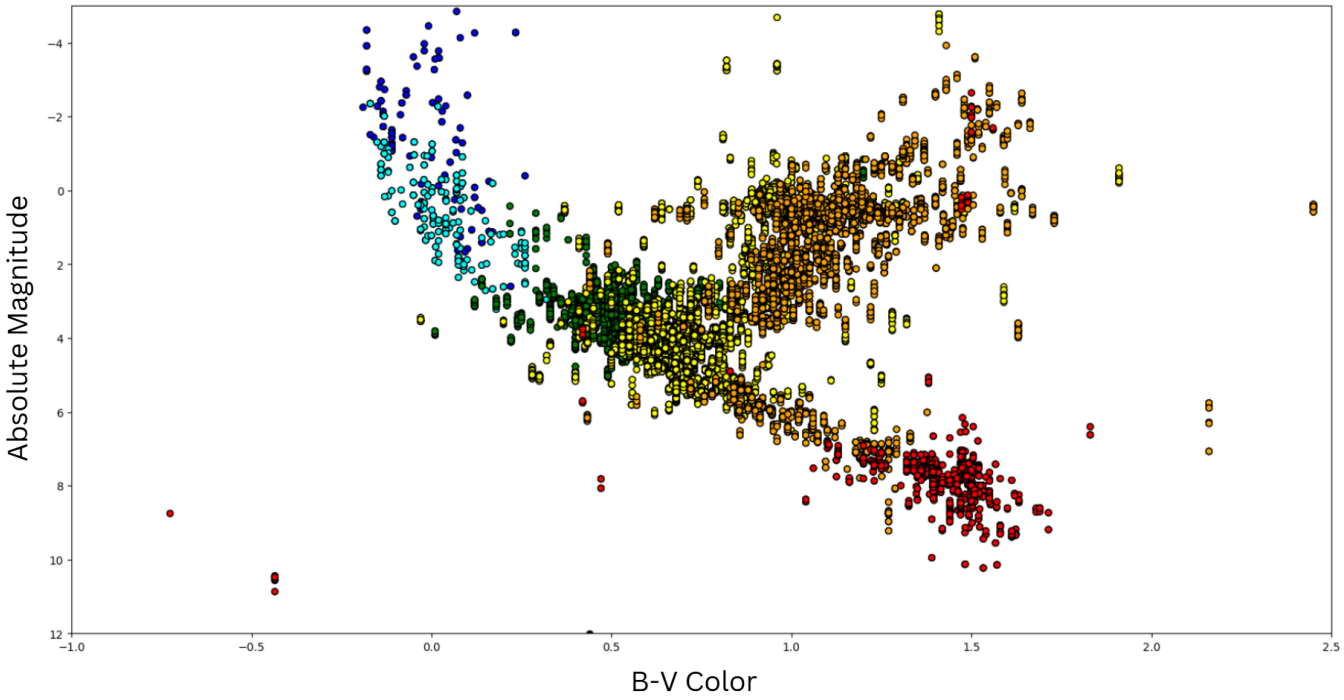

Figure 4: Labeled Data Points

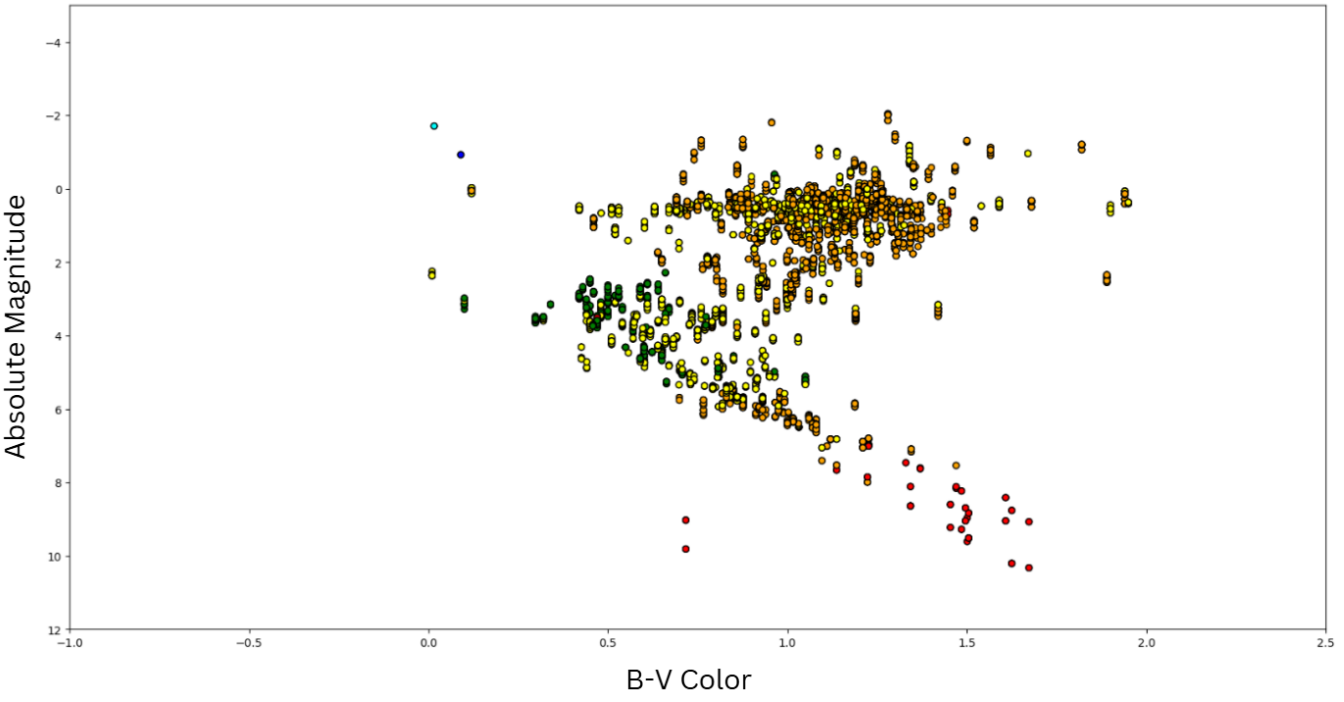

Figure 5: KNN Classification of Unlabeled Data

6 Discussion and Conclusion

6.1 Comparative Results

Although our final MLP model yields incredibly respectable results, our final KNN model is superior. Our KNN model was more accurate, as seen by the accuracy scores of each model shown in the tables. Additionally, it took us longer to train the MLP than the KNN. Granted, the grid search for the KNN contributed to its accuracy, and we did not perform a grid search on the MLP. Preliminary testing of such a method indicated that the runtime of the grid search made it infeasible for the scope of this project.

6.2 Implications

We consider our project to be a success. We trained 2 effective models to classify stars into their spectral classes. Both models produced impressive accuracy. Predictions of our KNN model on unlabeled data (figure 5) are consistent with expected outcomes. Either of these models, but especially our KNN model, could be used in real world situations to fill gaps in astronomical databases.

The integration of machine learning models into automated astronomical pipelines is a natural next step. Currently, several large-scale astronomical surveys use traditional data analysis methods, which often require significant manual intervention and human oversight. Our models, especially the KNN classifier, could be integrated into these pipelines to automatically classify stars and other astronomical objects as data is collected. Such integration would provide near-instantaneous classification results, enabling researchers to quickly identify and prioritize interesting objects for further study. This could lead to faster scientific discovery and the efficient allocation of resources in observational astronomy.

7 Future Work

7.1 Further Exploration

A myriad of paths for further exploration of the topic could be found by altering any of the decisions made by our team during the planning and experimenting process. A notable option is expanding the capability of the model to include classification into the subclasses of the 7 spectral classes. There are over 600 potential classes that a star can be ultimately categorized into. Training the model to handle all of these classes would require a re-evaluation of which features to give the model, at the very least. Given more time, a grid search could be performed to optimize the features used to train as MLP on this data.

Other machine learning algorithms could be explored to solve this case. Existing literature indicates that Random Forest models have been trained for this type of problem [Sharma *et al.*, 2019]. Additionally, we could look into using a Decision Tree model for the actual classification instead of just feature reduction.

7.2 Real-World Application

Spectral class is an incredibly powerful feature to have in a dataset because it combines several other aspects of the star into one. It is necessary for any study with a scope limited to fewer than all of the classes. It is standard for estimating the age of a star.

In datasets with billions of stars, there are bound to be missing values. If one of those missing values is spectral class (as we saw often in our examination of astronomical datasets) then an accurate classifier would allow anyone, regardless of astronomical prowess, to fill in those missing values with confidence in their veracity.

Acknowledgments

This research has made use of the SIMBAD database, operated at CDS, Strasbourg, France [de Données astronomiques de Strasbourg, 2024].

References

[Database, 2024] SIMBAD Astronomical Database. Simbad basic identifier for m33. Available at CDS, Strasbourg, France, 2024. Accessed: 2024-12-08.

[de Données astronomiques de Strasbourg, 2024] Centre de Données astronomiques de Strasbourg. Simbad astronomical database. Available at CDS, Strasbourg, France, 2024. Accessed: 2024-12-08.

[Sharma *et al.*, 2019] Kaushal Sharma, Ajit Kembhavi, Aniruddha Kembhavi, T Sivarani, Sheelu Abraham, and Kaustubh Vaghmare. Application of convolutional neural networks for stellar spectral classification. *Monthly Notices of the Royal Astronomical Society*, 491(2):2280–2300, 11 2019.