Spring 2019

Classification of Stars from Redshifted Stellar Spectra utilizing Machine Learning

Michael J. Brice
Central Washington University, michael.brice@cwu.edu

CLASSIFICATION OF STARS FROM REDSHIFTED STELLAR SPECTRA UTILIZING MACHINE LEARNING

A Thesis

Presented to

The Graduate Faculty

Central Washington University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Computational Science

by

Michael James Brice

June 2019

CENTRAL WASHINGTON UNIVERSITY

Graduate Studies

We hereby approve the thesis of

Michael James Brice

Candidate for the degree of Master of Science

APPROVED FOR THE GRADUATE FACULTY

_____ _____
Dr. Răzvan Andonie, Committee Chair

_____ _____
Dr. Szilárd VAJDA

_____ _____
Dr. Boris Kovalerchuk

_____ _____
Dean of Graduate Studies

ABSTRACT

CLASSIFICATION OF STARS FROM REDSHIFTED STELLAR SPECTRA UTILIZING MACHINE LEARNING

by

Michael James Brice

June 2019

The classification of stellar spectra is a fundamental task in stellar astrophysics. There have been many explorations into the automated classification of stellar spectra but few that involve the Sloan Digital Sky Survey (SDSS). Stellar spectra from the SDSS are applied to standard classification methods such as K Nearest Neighbors, Random Forest, and Support Vector Machine to automatically classify the spectra. Stellar spectra are high dimensional data and the dimensionality is reduced using standard Feature Selection methods such as Chi-Squared and Fisher score and with domain-specific astronomical knowledge because classifiers work in low dimensional space. These methods are utilized to classify the stellar spectra into the two standard star classification schemes, the Harvard Spectral Classification and the Morgan Keenan Luminosity Classes. If a star is classified into both of these schemes, many stellar properties can be approximated with ease, whereas the direct approaches can take up to months of observations. A physical phenomenon known as redshift causes machine learning complications through the feature matrix when classifying stellar spectra. However, classifiers utilizing redshifted stellar spectra performed with high accuracy. An approach for extracting redshift using predictions from the classification models is explored.

ACKNOWLEDGEMENTS

I would like to thank Dr. Răzvan Andonie and Dr. Szilárd VAJDA for helping me secure an account for the CWU High-Performance Machine (Turing) and for assisting me with its operations.

I would like to thank the Sloan Digital Sky Survey for providing me with their database and Benjamin Alan Weaver for providing me with assistance regarding the Sloan Digital Sky Survey.

Dr. Boris Kovalerchuk and Dr. Szilárd VAJDA who provided feedback and advice.

I would like to thank my colleagues, friends, and family for listening to me talk about my research and for providing me with feedback.

I am extremely grateful and thankful that Dr. Răzvan Andonie accepted to be my thesis adviser and allowed me to work on a thesis that is related to astronomy and for providing advice and ideas to achieve my goals.

AUTHOR COMMENTS

This thesis utilizes hyperlinks. If reading in PDF form, every reference to a Chapter, Section, Subsection, Equation, Table, Figure, or Reference is clickable and will automatically transition to the appropriate page.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF FIGURES (CONTINUED)

Figure Page

CHAPTER I

INTRODUCTION

Stellar classification is a fundamental task in stellar astrophysics. Traditionally, stellar spectra are classified by determining the wavelengths of absorption lines using wavelet transformations, statistical analysis, and using references to the Harvard Spectral and Morgan Keenan Luminosity Classification scheme [1] or they are classified by comparing the best fit of the spectra to that of templates using statistical tests [2]. The traditional classification schemes require complex data transformations and analysis to identify the class of a star based on its spectrum.

The amount of astronomical data and dimensionality of said data is growing rapidly through more and more ambitious astronomical surveys. The Sloan Digital Sky Survey (SDSS) is an example of an ambitious astronomical survey with high quantity and dimensional data.

Presently, SDSS is creating the most detailed three-dimensional maps of the Universe ever made, with deep multi-color images of one-third of the sky, and spectra for more than three million astronomical objects [3]. The SDSS provides stellar spectra with redshift wavelengths. This thesis will classify stars using SDSS data runs 12, 13, and 14 optical spectra datasets. These data runs were chosen because they are the most recent data runs at the time of this thesis and that each data run used the exact same spectrographs, data reduction pipelines, and the same wavelength resolution.

The SDSS and other large astronomical surveys create challenging problems for a thorough and speedy analysis. As such, automated classification methods are explored. However, some classification algorithms are limited to low dimensional data, making the use of feature selection and feature extraction essential.

In previous work related to machine learning classification of stellar spectra, Xing and Guo [4], Zhang *et al.* [5], and Yi and Pan [6] performed automated classification of stellar spectra using sources such as Pickles [7] and Jacoby [8]. Bazarghan and Gupta [9] also used the stellar spectra sources Jacoby and SDSS [10]. There are few results of automated classification of stellar spectra using SDSS data.

Redshift is a physical phenomenon that creates complications for the automated classification of stellar spectra. The process for determining redshift found within a spectrum produces the classification of a stellar spectrum as a by-product. The Human approach to identifying the redshift in a spectrum is as follows [11]:

1. Obtain the spectrum of a star which shows Absorption Lines [12] (defined in Chapter III).

2. From the pattern of lines, identify which line corresponds to which atom, ion, or molecule.

    - This pattern of lines are used to classify into the Harvard Spectral and Morgan Keenan Luminosity Classes.

3. Measure the shift of any one of those lines with respect to its expected wavelength, as measured in a laboratory on Earth.

4. Apply a formula that relates the observed shift to velocity along the line-of-sight (eq. (3.1)).

The automated process for identifying redshift that the Sloan Digital Sky Survey uses is as follows (Bolton *et al.* [2] and SkyServer: Redshifts, Classifications, and Velocity Dispersions [13]) :

1. Redshift and classification templates for galaxy, quasar, and CV star classes are constructed by performing a rest-frame principal-component analysis (PCA) of training samples of known redshift.

2. The combination of redshift and template class that yields the overall best fit (in terms of lowest reduced chi-squared) is adopted as the pipeline measurement of the redshift and classification of the spectrum.

3. The most common warning flag is set to indicate that the change in reduced chi-squared between the best and next-best redshift/classification is either less than 0.01 in an absolute sense, or less than 1% of the best model reduced chi-squared, which indicates a poorly determined redshift.

This makes redshift correcting spectra redundant when classifying stellar spectra. However, identifying the redshift is still an important aspect of stellar astrophysics, and as such, cannot be ignored.

This thesis proposes two approaches to stellar classification and, as an intentional by-product, an approach for redshift extraction. These approaches are characterized by the following:

- Avoid complex transformation and statistical analysis of the spectra space.
- Use spectra without redshift corrections.
- Use Machine Learning Classifiers.
- Uses Standard Feature Selection Methods to reduce the number of flux measurements.
- Uses astronomical knowledge to perform feature selection.

- Extracts redshift using predictions from a machine learning model.
- Creates a basic confidence metric for model prediction of stellar classification.

However, the accuracies of the classifications in this thesis are only as accurate as the classifications from the SDSS. Spectra from the SDSS are labeled with a redshift warning which is used to help determine if a sample is poorly classified. The majority of SDSS spectra have a low, if not 0, redshift warning.

The *Classification into the Harvard Spectral Classification Scheme* method uses Machine Learning classifiers to classify stellar spectra only into the Harvard Spectral Classification Scheme. Since the dimensionality of the spectra is large, feature selection is used. Feature selection may destroy the shape and the structure of each stellar spectrum by only using the most relevant flux measurements. This is in contrast to [4], where PCA and wavelet reduction are used to reduce the number of flux measurements while maintaining the shape and structure of each spectrum.

The *Single Classification into both Harvard and MK Classification Schemes* method uses Machine Learning classifiers to classify stellar spectra into both the Harvard Spectral and MK Classification Schemes using a single classifier method. Astronomical knowledge is used to reduce the number of flux measurements. This results in key aspects of the spectra being preserved for classification which allows simultaneous classification to be possible.

The motivation for this work is as follows. If a star can be accurately classified into both the Harvard Spectral and Morgan Keenan Luminosity schemes then stellar properties can be easily obtained. More importantly, cataloging stars based on their classification makes it easier for future research. For example, if a particular type of stellar physics occurs in G type stars, it is easier to observe and research G type stars if you already know

which stars are G types stars and which ones are not. Most importantly, the use of machine learning allows for a computationally fast classification of stars.

The research described in Chapter V has been disseminated as ”Classification of Stars using Stellar Spectra collected by the Sloan Digital Sky Survey” [14] in the Proceedings of the International Joint Conference on Neural Networks (IJCNN). This work will be presented at the IJCNN in July 2019. The research described in Chapter VI has been disseminated as a full paper for a peer review journal.

The research from Chapter V and Chapter VI was adapted for a general audience and was presented at Central Washington University's Symposium Of University Research and Creative Expression (SOURCE) during May 2019.

The structure of this thesis is as follows. Chapter II provides previous results on stellar spectra classification. Chapter III provides background information on astronomy. Chapter IV provides background information on Machine Learning. Chapter V describes the first classification experimental setup, analysis, and results of the Classification into the Harvard Spectral Classification Scheme method. Chapter VI describes the experimental setup, analysis, and results of the Single Classification into both Harvard and MK Classification Schemes method. Chapter VII describes redshift extraction from spectra and Machine Learning results. Chapter VIII presents the conclusions.

CHAPTER II

PREVIOUS RESULTS ON STELLAR SPECTRA CLASSIFICATION

Yi and Pan [6] utilized Random Forest to classify stellar spectra. The authors also compared Random Forest to Neural Networks (Multi-layer Perceptron). The spectra used in their experiments are taken from Pickles [7]. Random Forest is implement with mtry = 400 and mtree = 1000. The Multi-layer Perceptron has N feature nodes in the input layer and 5 nodes in the first hidden layer, 1 node in the second hidden layer, and 1 node in the output layer. The authors find that Random Forest performed better than the Multi-layer Perceptron with Root Mean Square Error (RMSE) of 1.04 and 1.36 respectively.

Xing and Guo [4] utilizes a Support Vector Machine for stellar spectra classification. The authors utilize Principle Component Analysis (PCA) to reduce the dimensions of the spectra and wavelet transformations to reduce noise in the spectra. The stellar spectra used in their experiments are selected from Jacoby [8]. Xing and Guo [4] reports 81.66% accuracy for just SVM with no data reduction, 93.26% for wavelet + SVM and 81.30% for PCA + SVM.

Zhang, Luo, and Tu [5] separated the classification into two classifiers. For the Harvard Spectral Classification scheme, the authors used a non-parameter regression method. For the Morgan Keenan Luminosity Classification scheme, the authors removed or normalized the continuum of the spectra and used a partial least-squared regression method. The authors use three spectra data sources, Silva [15], Pickles [7], and Jacoby [8]. They achieved a standard deviation for the Harvard Spectral Classification scheme of $\sigma = 0.7994$ and for the Morgan Keenan Luminosity Classification scheme of $\sigma = 0.58159$.

Weaver and Torres-Dodgen make the following claim: "The results of this series of papers and of the preceding section make it apparent that a complete classification system

is probably beyond the capabilities of a single network. However, we have demonstrated that [Artificial Neural Network (ANN)] components of a system of ANNs can successfully perform a complete classification." [16]. This claim is important because this thesis is able to classify into both the Harvard Spectral Classification and Morgan Keenan Luminosity Classification scheme using a single classifier (Chapter VI).....

Bailer-Jones, Irwin, and Hippel [17] use spectra from the Michigan Spectral Survey [18] with a wavelength range of 3,800 - 5,200 $\mathring{A}$. The authors utilize a committee of identical neural network models, for their paper, 10 neural networks are used. The authors use a Multi-layer Perceptron (MLP) with gradient descent and backpropagation for classification into the Harvard Spectral Classification scheme. The authors then utilized a MLP in probabilistic mode for classification into the Morgan Keenan Luminosity Classification scheme. Bailer-Jones, Irwin, and Hippel [17] utilize Principal Component Analysis to reduce the dimensionality. For classification into the Harvard Spectral Classification scheme, the neural network had 50 nodes in the input layer, 5 nodes in the hidden layer and a single output node. The average classification error is 1.07 SpT. SpT stands for spectral type. For classification into the Morgan Keenan Luminosity Classification scheme, they report an error of 1.53 SpT. For luminosity classes of V and III they achieve over 95% accuracy but for IV it was considerably worse. The authors make the following statement: "Owing to the spectral type–luminosity class correlation in the data set, the network may be unable to adequately separate out luminosity effects from temperature ones. This is not helped by the weakness of the luminosity [distinguishing] features in this wavelength region." [17]. This is important because this thesis uses a similar wavelength range and is able to classify luminosity classes with high accuracy (Chapter VI).

Bazarghan and Gupta [9] utilize a Probabilistic Neural Network implemented in MATLAB. They use SDSS data from data run 2 for their testing sets. They use Jacoby dataset for spectra for their training sets. They state "[The spectra] must be all uniform, having [the] same wavelength scale, the starting and end wavelengths must be [the] same for all the spectra and [the spectra] must also have closely match [wavelength] resolution. These must be valid [for] both the training and test dataset." [9]. This is extremely important for any automated classification approach for stellar spectra. The dimensionality of the spectra was not reduced. However, the authors re-binned the SDSS spectra to have the same resolution as the Jacoby spectra. One could argue that this is feature selection because they are converting and reducing the number of measurements, but a more accurate description would be that they are simply spectra fitting and not significantly reducing the number of dimensions. The authors used a χ^2 value to determine classification accuracy. They make the assumption that a χ^2 value of 0.002 or lower is considered classified correctly, then they achieved a success rate of about 88% in only a few seconds.

However, Schierscher and Paunzen [19] claim that they were surprised that Bazarghan and Gupta [9] were able to classify a significant number of hot O and B type stars. Schierscher and Paunzen [19] took a deeper look at the O type stars in Bazarghan and Gupta [9] samples. Schierscher and Paunzen [19] concluded that the majority of samples labeled as O and B type stars are really A type stars. Therefore, Schierscher and Paunzen [19] claims that the PNN that Bazarghan and Gupta [9] utilize fails to classify the SDSS spectra on a reliable basis. This appears to be the result of Bazarghan and Gupta [9] utilizing other spectral sources to train their model. Schierscher and Paunzen make this statement: "From our experience and other results from the literature (Bailer-Jones, Irwin & von Hippel [17]), it is quite problematic not to use spectra from the target sample for the training phase." [19]. In other words, it becomes problematic when the machine learning model is trained using

stellar spectra from a different data source as the testing data. This is obvious because each astronomical survey has a different set of precision, image focus, and average viewing conditions, not to mention different atmospheric conditions.

Schierscher and Paunzen [19] utilize an ANN. The ANN utilizes a back-propagation algorithm and is implemented using the Stuttgart Neural Network Simulator (SNNS). The authors utilized two approaches, continuous mapping with one neuron in the output layer and discontinuous mapping with one neuron per class in the output layer. The authors used SDSS for their Stellar Spectra source. However, their SDSS source was from the data run 7 dataset and were classified using SEGUE Stellar Parameter Pipeline (SSPP) [20], while this thesis is data run 12, 13, and 14 and are classified using the approach described in Bolton *et al.* [2]. In total there were 29,182 samples for a total of 26 different classes. These classes are generated from effective temperature of the star and not from its Harvard class. However, effective temperature and the Harvard class are related through the Hertzsprung-Russell diagram (Chapter III: Importance of Stellar Classes).

Schierscher and Paunzen [19] reduce the spectra dimensions by using interesting areas in the spectra. These areas are the Hydrogen lines (H_α to H_δ), the Ca H + K lines and the G band. They resulted in 2,400 dimensions being reduced to 435. This thesis uses a related approach in Chapter VI but with only two lines and reducing 4,617 dimensions to 34. The authors are able to achieve an 85% match in comparison with SSPP. However. the authors never address the fact that their data is imbalanced. The authors acknowledge that the majority of samples were classified as the majority classes but they do not take into account the bias the model has towards the majority class. What is more interesting is how well the model handles the minority classes, but this is not explored in their paper.

Schierscher and Paunzen [19] make the claim: ”Nevertheless, some inconsistencies of the classification and the pipeline values turned up. In most cases, these differences could

be traced back to incorrect pipeline values which are clearly visible by [inspecting] the corresponding spectra by eye." [19]. The authors are correct in stating that the pipeline does make incorrect classifications on occasions, but they are also assuming that the Human eye (more importantly their eyes) are absolutely correct, especially at the boundaries between classes. For example, if the goal is to classify colors into Red or Orange classes and the sample is a red-orange color, some would classify it as red, while others would classify it as orange. This does not take away from their conclusion, but it does raise the question, were the samples that the authors claim were misclassified in the SSPP on the boundaries between classes or were they clearly within the scope of a different class. It is reasonable to argue that if the samples were on the boundaries then the authors claim of pipleline values being incorrect is subjective to the authors' interpretation of the data. Even though the authors state "clearly visible", this is still subjective because to one person red-orange can be "clearly visible" as red and to someone else "clearly visible" as orange.

Gray and Corbally [21] use an expert computer program (MKCLASS) to classify stellar spectra. MKCLASS is accompanied by a preprocessing program called MKPRELIM. MKPRELIM determines the redshift of the spectrum and transforms it into its rest frame. As stated before, this is redundant because identifying the redshift before using an automated classifier results in the classification. The authors could argue that MKCLASS is faster than using a lookup table for both Harvard and MK standards, but they do not address this in their paper.

Almeida and Prieto [22] use K means clustering to classify SDSS stellar spectra. Classification consists of three points [22]:

1. Find the number of clusters and their centers.

2. Assign each spectrum to one of these centers.

3. Estimate the probability that the choice is correct. This is used to quantify the goodness of the classification.

The authors experiment with spectra with continuum intact and spectra with the continuum removed. Both of these experiments has a problem, the authors used K means in a 3,849-dimensional space. Moreover, the authors ran their experiments for about two hours. The experiments utilizing spectra with its continuum seem to generate 16 major classes and 10 minor classes where 99% of the spectra lie in the major classes [22]. For spectra with its continuum removed generates 13 major classes and 1 minor class [22]. Not only does this author not generate classes with a one to one ratio between cluster and MK classes making their results hard to compare, they do not reduce the number of dimensions nor use a computationally fast tool.

CHAPTER III

ASTRONOMY BACKGROUND

This chapter describes the astronomical information required to better understand this thesis. Stellar spectra, classification schemes, importance of classifying stars, and redshift are described. Additional information regarding stellar spectra and their classes can be found in "Stellar Spectra Classification" [23] and "Advances in Spectral Classification" [24].

Stellar Spectra

A stellar *spectrum* (plural *spectra*) is the distribution of starlight with respect to wavelength [25]. Incandescent light bulbs, when viewed with the unaided eye, seem to emit white light. When viewed through a prism or a diffraction grating, the bulb actually is emitting a rainbow or spectrum of light. Stars also emit a spectrum of light. Stellar spectra are collected by a spectrograph, which utilizes a fine-tuned diffraction grating to collect the spectrum of light emitted by a star.

The stellar spectra from the SDSS dataset contain wavelength measurements and flux measurements. The *wavelengths* are discrete values that represent a range of wavelengths of light. For example, a wavelength value can be 6,000$\pm$5 $\mathring{A}ngstr\ddot{o}ms$ ($\mathring{A}$). *Flux* is the number of photons that pass through an area per second per particular measured wavelength. The shape and features of stellar spectra are defined by the flux measurements at particular wavelength values. Figure 1 shows an example spectrum of a flux scaled G2 (Sun-like) star. The downward spikes are absorption lines. Figures 1 and 2 show how the shape of the spectra differ with class.

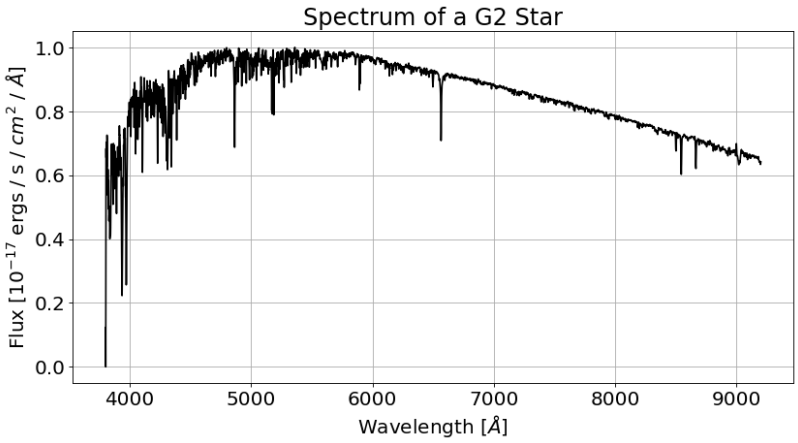


FIGURE 1: Example: Stellar Spectrum of a flux scaled G2 star collected by the SDSS.

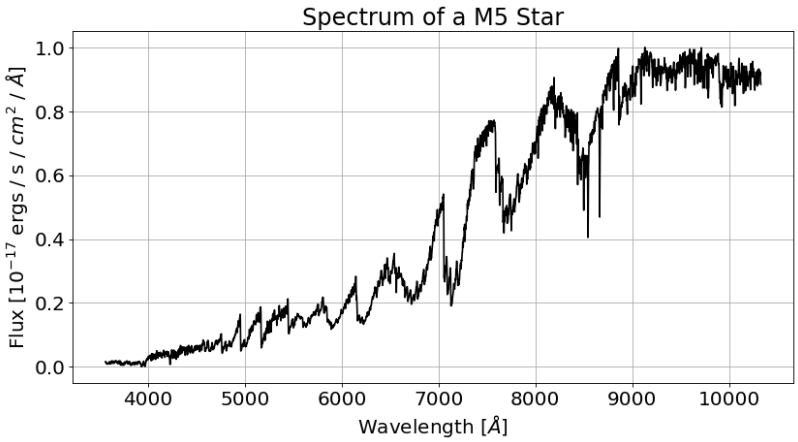


FIGURE 2: Example: Stellar Spectrum of a flux scaled M5 star collected by the SDSS.

Stellar spectra have many interesting properties. One of the most prominent are Absorption lines. Absorption lines indicate the types of elements, ions, and molecules present in a star such as hydrogen, helium, and heavy metals and are key to classifying stars. Most elements, ions, and molecules have more than one Absorption line. For example, some of the common Hydrogen (H) lines are known as H_α, H_β, H_δ, and H_γ.

Absorption lines are formed when an element, ion, or molecule absorbs a photon (light). The electrons in an atom have to be in quantized energy levels (orbitals), so when a photon with the exact right energy hits the electron, the electron jumps to a higher orbital

[26, 27]. This absorbs the photon. Emission lines work the same way, except the electron drops an energy level and releases a photon [26, 27].

Additional information regarding absorption lines can be found here: ”SkyServer: Absorption and Emission Lines” [12].

Stellar Classification Types

There are two primary stellar classification schemes in use today: The Harvard Spectral classification (Harvard) and the Morgan - Keenan Luminosity Classes (MK) [28]. The Harvard spectral classification is a one-dimensional surface temperature classification that uses absorption lines found in the spectra from stars to categorize into groups labeled O, B, A, F, G, K, and M. These groups are then divided into 0 - 9. For example, A0 - A9 where A0 is an early A type star and A9 as a late A star [28]. A star’s Harvard class will change over time. For example, the Sun in approximately 4.5 billion years will turn into a red supergiant, the Harvard class will change from G to a K or M type star.

The MK Classes extend on the Harvard Spectral classification by adding a luminosity class. The MK Classes are based on the widths of the absorption lines, which is related to a star’s luminosity, and by comparing to the MK standard stars [23]. The MK adds the identifiers of I to VI, which represents Supergiants, Bright Giants, Giants, Sub-Giants, Main Sequence (Dwarfs), and Sub-Dwarfs [28]. For example, Betelgeuse is a red supergiant and a stellar class of M1 (Harvard) Ia (MK) star and Proxima Centauri is a main sequence red dwarf and a stellar class of M6 V star. A star’s MK class will change over time as well. Using the same Sun example, when the Sun turns to a red supergiant, its MK class will turn from V to I. For more information on the MK Classes, read ”An atlas of stellar spectra, with an outline of spectral classification” [1].

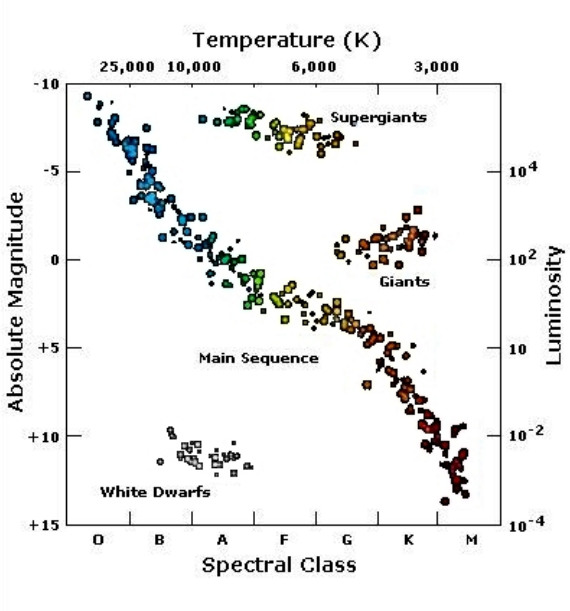


FIGURE 3: Hertzsprung Russell Diagram [29].

It is important to note that the Harvard and MK schemes are really one scheme, where the MK is an extension of Harvard. However, in common published literature, they are described as two separate schemes and this thesis will do the same.

Importance of Stellar Classes

The spectrum of a star reveals a wide range of information to astronomers. This includes the chemical composition of the star's atmosphere through the absorption lines or Harvard Spectral Classification, an approximate luminosity of the star through the Morgan Keenan Classification, the star's radial velocity through redshift, and more. A completely classified spectrum of a star places the star on the Hertzsprung-Russell (H-R) diagram, shown in Fig. 3. The H-R diagram is a plot of stars where the horizontal axis is the spectral class or surface temperature and the vertical axis is the luminosity or absolute magnitude.

The location of a star on the H-R diagram tells astronomers information about that star. For example, stellar evolution can be mapped out on the H-R diagram [28]. The

location of a star on the H-R diagram defines whether it is in the main sequence, a giant, a super giant, or whether the star has stopped fusion of Hydrogen and is fusing Helium, or if it has stopped fusing Helium and so on [28]. More information regarding the H-R diagram can be found in sources "To Measure The Sky: An Introduction to Observational Astronomy" [25], "An Introduction to Modern Astrophysics" [28], and "SkyServer: The Hertzsprung-Russell Diagram"[30].

Redshift

Redshift is caused by the relative motion of a star with respect to an observer through the Doppler effect on light. When a star moves away from an observer, the wavelengths of light appear to be longer, or, in terms of the visible spectrum, redder. Redshift causes the flux measurements to be shifted compared to what would have been observed at rest [25, 28]. Additional information regarding redshift can be found in: "SkyServer: Redshifts [11].

All stars have different redshift values. Redshift is defined by eq (3.1), where Z is the redshift value, $\lambda_{observed}$ is the observed (redshifted) wavelength and λ_{rest} is the rest wavelength [25, 28]. Equation (3.2) defines how the flux density, f, changes due to redshift (see [25]). Redshift for stars from the SDSS database have $Z \ll 1$, therefore eq. (3.3) is a sufficient approximation and is used in this thesis. This is because the stars that the SDSS database has collected are found within the Milky Way Galaxy:

$$Z = \frac{\lambda_{observed} - \lambda_{rest}}{\lambda_{rest}} \tag{3.1}$$

$$f(\lambda_{rest}) = (1 + Z)^2 f(\lambda_{observed}) \tag{3.2}$$

$$f(\lambda_{rest}) \approx f(\lambda_{observed}) \tag{3.3}$$

Figure 4 shows a simplified example of redshift. The center spectrum is at rest. The top spectrum is identical to the center spectrum except it is redshifted (positive redshift). The bottom spectrum is identical to the center spectrum except it is blueshifted (negative redshift). The center spectrum is what would be observed if the star was not moving relative to the telescope. The solid black lines are absorption lines. Notice how the black lines are shifted in the top and bottom spectra. They appear in the exact same pattern as the center spectrum but shifted. The absorption lines that appear in the center spectrum are the exact same absorption lines that appear in the top and bottom spectra, but they are found at different wavelengths due to redshift. The absorption lines found in the center spectrum, in terms of this thesis, are referred to as *rest wavelength* absorption lines, where the absorption lines found in the top or bottom spectra are referred to as *redshift wavelength* absorption lines. Regardless if an absorption line is at rest wavelength or redshift wavelength, the absorption lines are for the same elements/molecules.

Figure 5 shows a plot of a spectrum with rest and redshift wavelengths presented in the format used in this thesis. Notice that the two spectra are identical, but shifted. The dashed line represents spectra with redshift and the solid line represents the same spectra but with redshift corrections applied. Redshift correction shifts the wavelengths from their observed values to the values that would have been observed if the star was at rest relative to the observer.

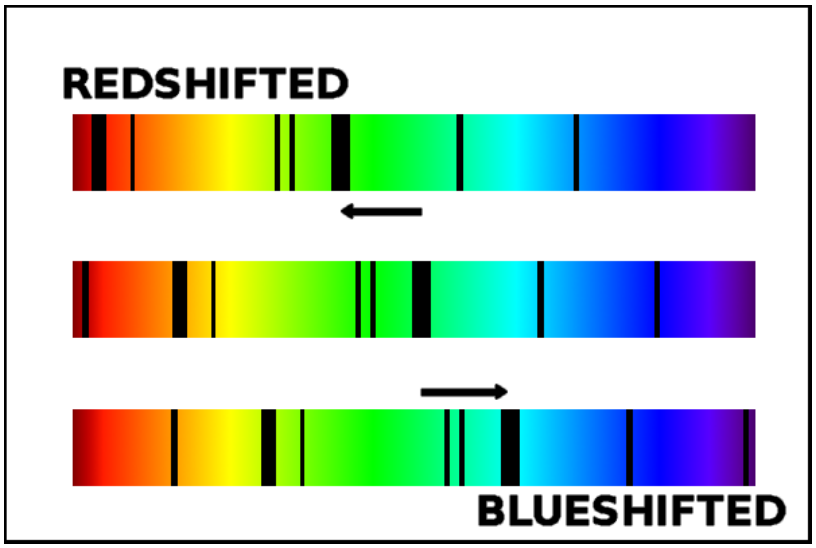


FIGURE 4: Simplified example of redshift [31].

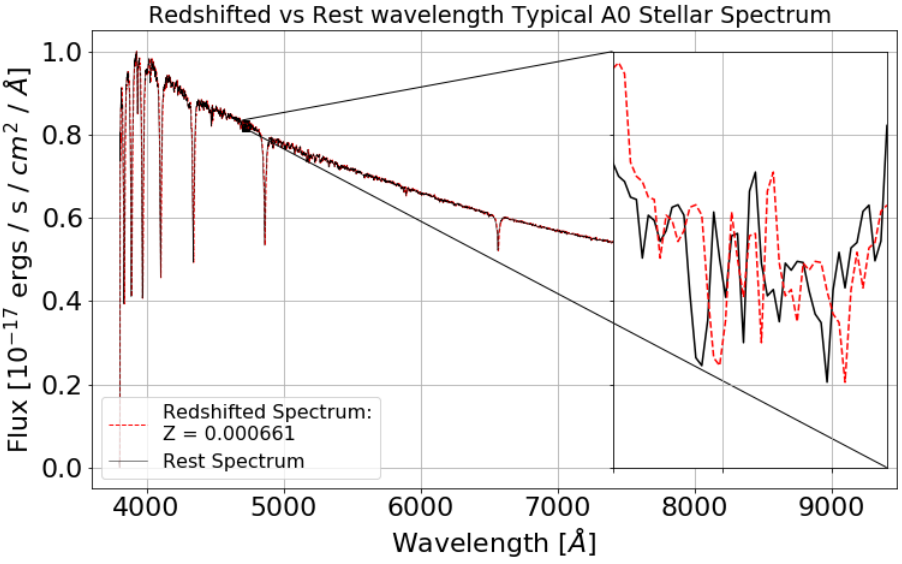


FIGURE 5: Example of redshift.

CHAPTER IV

MACHINE LEARNING BACKGROUND

This chapter describes background information of the Machine Learning used in this thesis. Classifier and feature selection methods are described. Feature selection is used to reduce the high dimensional data from SDSS so that the stellar spectra found in SDSS can be automatically classified. Additional information regarding classifier and feature selection methods can be found in "Machine Learning: An Algorithmic Perspective" [32], "Statistics, Data Mining, and Machine Learning in Astronomy" [33], and "Feature Selection for High-Dimensional Data" [34].

Classifier Methods

The classifier methods used in this thesis are described and the reasoning for using them is provided. However, there is truly no for sure way to know which methods will work best on any given dataset until it is tried empirically [33].

K-Nearest Neighbors

K-Nearest Neighbors (KNN) classifies using the K closest data points. Lets say someone is in a nightclub and they want to dance. However, they do not know how to dance to the particular song playing. The dancer would look around and see what the nearest people are doing and mimic their dance [32]. This is how KNN works. The sample to be classified "asks" its k-nearest neighbors what their class is and based on popular vote, decides that the sample must be of that class [32]. This is similar to how the dancer decides that the appropriate dance to that song has to be the dance that the nearest people are dancing [32].

However, there can be inherent flaws with this approach. The first is the assumption that the other people are dancing the correct dance or that the nearest neighbors are in fact the correct class. The second is whether the nearest neighbor is actually close by. For the example of the dancer, the nearest dancer could be in the nightclub across the street and as such could be dancing to a different song entirely. To overcome the second flaw, numerical weights can be applied to the vote in the form of distance, the closer the neighbor, the more influence it has. Knowledge of the particular dataset in question warrants whether weights are required or not.

To identify the nearest neighbors, a distance metric is used. For this thesis, Euclidean distance is used:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + \dots} \tag{4.1}$$

The distance metric inherently makes KNN a computationally expensive method for high dimensional data. However, the computational time is reasonable when the data is in the low dimensions. Not only does KNN works best with low dimensional data, it also works best when there are a large number of samples to train with [32, 33].

KNN was chosen for this thesis because of its fast computational time in low dimensions (the spectra's dimensions are reduced in this thesis) and for its ability to separate spectra of classes that have extremely similar features. This is apparent when samples in KNN training space are represented as a N-dimensional point, where minor changes in any of the N dimensions can significantly change its position in the N-dimensional space. KNN was also chosen because of the large number of samples available for training.

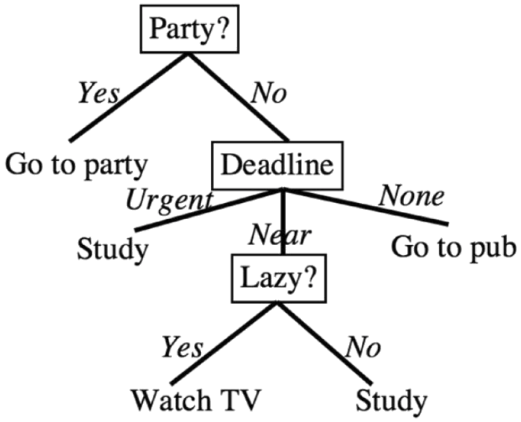
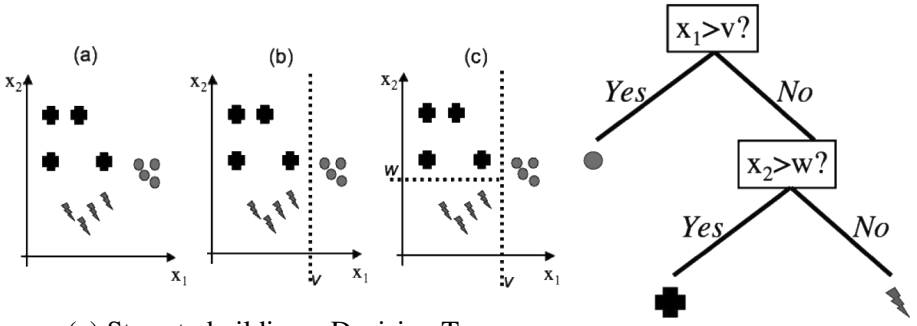


FIGURE 6: Example of a Decision Tree [32].

Random Forest

The idea of Random Forests (RF) is that if one decision tree is good, then many trees (forest) should be better [32]. The most frequent name attached to RF is Breiman [32]. The most interesting aspect about RF is that it creates randomness from a standard dataset. This is accomplished in two ways. First the individual trees are trained using slightly different sets of features from the data and second each tree uses a subset from its set of the features from the data. The RF classifier is an ensemble method. The output of the forest is by majority vote where the output of each individual tree is cast as a vote.

The individual trees are decision trees. An example of a decision tree can be seen in Fig. 6. For this example, the decision that needs to be made is whether a student should go to a party, study, go to a pub, or watch TV. To make this decision, there are three features, is there a party, is there a deadline, and is the student feeling lazy. A decision tree is read from the top (root) down. If the party feature is yes, then the decision is go to party, but if the party feature is no, then the outcome depends on deadline. If the deadline feature is set to urgent, then the decision is to study and so on.



(a) Steps to building a Decision Tree.

(b) Resulting Decision Tree.

FIGURE 7: Example of building a Decision Tree [32].

To build a decision tree, a feature is selected to be the root of the tree. In Fig. 7 X_1 is used as the root. Then a splitting value is chosen, in the case of Fig. 7 that value is V. This can be seen in Fig. 7a (b). Since all samples where X_1 is greater than V are of a single class, that branch stops. For all samples where the value of X_1 is less than V, there are two classes, so another split is needed. The feature used for this split is X_2. For all samples whose value of X_2 is greater than W is a single class and all samples whose value of X_2 is less than W is a different single class. This is seen in Fig. 7a (c). This results in a decision tree seen in Fig. 7b.

Essentially a decision tree tries to separate the classes by splitting the tree at the different features. Each split in the tree has to be evaluated to see if it is the best possible split. For this thesis, the Gini impurity is used. The Gini impurity is calculated as follows [32]:

$$I_G = 1 - \sum_{i=1}^{J} p_i^2 \tag{4.2}$$

where I_G is the Gini impurity, J is the number of classes, p_I is the fraction of items that have the class label of class i.

Random Forest was chosen for this thesis because of its simplistic manner. Similarly to KNN, the training space can be represented as an N-dimensional space, where minor changes in any of the N dimensions can significantly change the position of a N-dimensional point in the N dimensional space. Random Forest was also chosen because authors who used similar data reported excellent results [6].

Support Vector Machine

Support Vector Machine (SVM) was introduced by Vapnik in 1992 and has taken off radically since [32]. In the Binary Class problem, SVM attempts to linearly separate the two classes, as seen in Fig. 8. However, it is unreasonable to expect similar points on the testing dataset. There could be points that are closer to, if not on the wrong side of the classification line. Now imagine if there was a ”no mans land” around the classification line (Fig. 9), such that any point within that ”no mans land” would be declared to close to be accurately classified [32]. The lines used to create ”no mans land” are called Support Vectors. These Support Vectors are generated at the closest data point from each class to the classification line and it is parallel to the classification line. Therefore the best classifier is the one that goes through the middle of ”no mans land” [32]. There are two claims made: first is that ”no mans land” should be as large as possible and second that the Support Vectors are the most important data points because they are the ones that might get classified wrong [32].

One major limitation of SVM is that it is limited to linear decision boundaries [33]. However, the use of kernels is a simple and powerful way to take an SVM and make it nonlinear. The basic idea is to take a kernel which transforms an occurrence of $(x_i, x_{i'})$ to $K(x_i, x_{i'})$ where K has certain properties which allow the SVM to operate in a higher

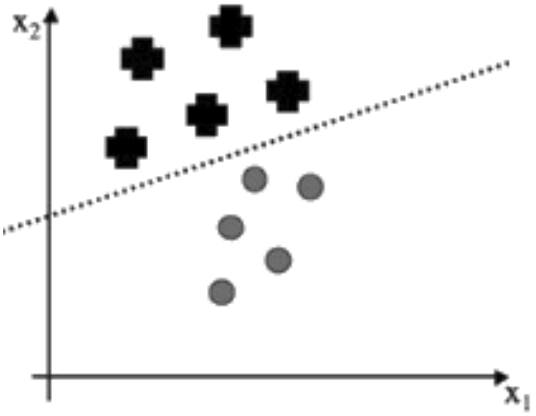


FIGURE 8: Example of linearly separating two classes [32].

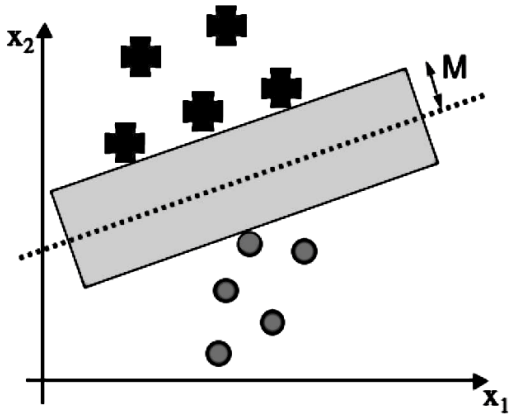


FIGURE 9: Example of Support Vector Machine [32].

dimensional space where the data is linearly separable [33]. This thesis uses the Radial Bias Function or Gaussian kernel. The Gaussian kernel is as follows:

$$K(x_i, x_{i'}) = e^{-\gamma||x_i - x_{i'}||^2} \tag{4.3}$$

However, this thesis is concerned with the Multi-class problem. Multi-class SVM uses N Binary Class SVMs to classify N classes. Each Binary Class SVM classifies into either $Class_i$ or $Class_{Remaining}$. The classification is based on the strongest SVM classifier, the one where the classification is the furthest into the $Class_i$ region. [32].

SVM was chosen for this thesis because of how it separates data and that similar research found it to be an effective classifier for stellar spectra [4]. However, SVMs do not work well with extremely large datasets due to being computationally expensive [32], which is why SVM was only used for Undersampled balanced datasets discussed in Chapter V.

Feature Selection Methods

There are three categorical types of standard feature selection methods: Filter, Embedded, and Wrapper. Filter feature selection methods are based on performance evaluation metrics that are calculated directly from the data [34]. Embedded methods do not separate the learning from the feature selection [34]. The weights from a classifier method are used to determine which features are most important to classification. Wrapper methods use learning algorithms to determine how accurate a particular subset of features are and compares to other subsets to find the optimal set of features [34]. This thesis utilizes Filter methods because they are computationally less expensive than Embedded and Wrapper methods. A review of feature selection methods for high dimensional data in astronomy can be found in [35].

Chi Squared

Chi-Squared feature selection is an univariate filter based on the χ^2 statistic. Chi-Squared feature selection evaluates each feature independently with respect to the classes. The higher the Chi-Squared value, the more relevant is the feature with respect to the class [34]. Given a number of intervals V, the number of classes B, and the total number of instance N, the value of Chi-Squared for a feature is calculated as:

$$\chi^2 = \sum_{i=1}^{V} \sum_{j=1}^{R} \frac{[A_{ij} - \frac{R_i * R_j}{N}]^2}{\frac{R_i * R_j}{N}} \tag{4.4}$$

where R_i is the number of instances in the range i-th. B_j is the number of instances in class j-th. A_{ij} is the number of instance in the range ith and class j-th [34]. The top K best features can be selected by the features with K largest Chi-Squared values.

Chi Squared was chosen because of its simplicity and low computational time. Chi-Squared is a commonly used feature selection method that is reliable.

Fisher Score

Fisher score is a supervised filter feature selection method. Fisher score selects features whose values of samples within the same class are similar while feature values of samples from different classes are dissimilar [36]. The Fisher Score of each feature f_i is calculated as:

$$Fisher\_score(f_i) = \frac{\sum_{j=1}^{c} n_j(\mu_{ij} - \mu_i)^2}{\sum_{j=1}^{c} n_j \sigma_{ij}^2} \tag{4.5}$$

where n_i indicates the number of samples in class j, μ_i is the mean value of feature f_i, μ_{ij} is the mean value of feature f_i for samples in class j, and σ_{ij}^2 is the variance value of feature

f_i for samples in class j [36]. The top K best features can be selected by the features with K largest Fisher Scores.

Fisher Score was chosen because of its simplicity and because it performed well on other large astronomical datasets [35].

CHAPTER V

CLASSIFICATION INTO THE HARVARD SPECTRAL CLASSIFICATION SCHEME

This chapter describes the experiments to classify the stellar spectra into only the Harvard classification scheme using a single classifier. The contents of this chapter are as follows. Section Approach to Classification describes the data, data pre-processing, the feature matrix, and the classifier methods used. Section Experiments explains the implementation of the feature selection and classifier methods and the experimental framework and results. Section Discussion analyzes the results.

Approach to Classification

In this section, the data, the pre-processing steps, and the feature selection and classification techniques for stellar classification is described.

Data

The stellar spectra collected by the SDSS are pre-processed by SDSS scientists through the methods presented by Dawson *et al.* [37] and Stoughton *et al.* [38]. There are two spectrographs used by the SDSS to collect the stellar spectra: the Baryon Oscillation Spectroscopic Survey (BOSS) spectrograph and the SDSS spectrograph. The data used in this paper was collected using the SDSS spectrograph. The SDSS spectrograph is identical to the BOSS spectrograph except the BOSS spectrograph has upgraded Charged Coupled Device (CCD) cameras, a larger wavelength sensitivity, and a larger number of astronomical objects that can be simultaneously observed [39]. Therefore, due to the limited literature of the SDSS spectrograph, the more in depth descriptions of the BOSS spectrograph will suffice and are presented by Smee *et al.* [40].

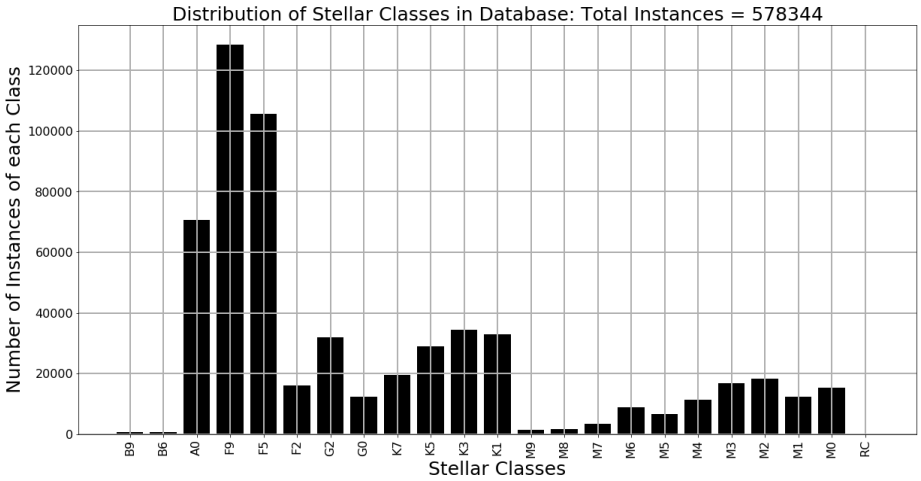


FIGURE 10: Distribution of classes in the dataset. RC = Remaining Classes of 0 instances

The dataset used in this chapter comes from SDSS data run 14, which collected 600,967 stellar spectra. Some of the stellar spectra were rejected from this study because they were not able to be classified (e.g. the SDSS stellar class is not O, B, A, F, G, K, and M with sub-classes of 0 - 9). Other spectra were rejected because large portions of flux measurements were missing due to CCD or other instrument failures. Samples of class G5 were rejected due to there being a total of 2 samples of that class. The usable dataset contains 578,346 stellar spectra covering 22 of the 70 Harvard classes. The distribution of the classes from this dataset is highly imbalanced, as seen in Fig. 10. The dataset is balanced using: *i)* undersampling (removing samples); and *ii)* hybrid sampling, consisting in both undersampling and oversampling (duplicating samples) [41]. Undersampling results in each class having 572 samples for a total of 12,584 samples. Hybrid sampling results in each class having 16,682 samples for a total of 367,004 samples. Data balancing and data pre-processing (Chapter V: Data pre-processing) do not change the 22 output classes. These spectra are used as the input vectors to the classifier methods, with each spectrum having approximately 3,800 to 4,000 features. These features are the flux measurements.

Data pre-processing

The flux intensity for a given class of star varies for a variety of reasons, including proximity to the spectrograph and the star's luminosity. Another reason is the interstellar medium that lies between the telescope and the star. The interstellar medium also absorbs and emits light which can have minor effects on a spectrum. This creates a problem for classification because samples of similar classes do not have approximately the same measurements. To resolve this issue, the flux is scaled from 0 to 1 using eq. (5.1), where f_i is the ith flux measurement, f_{max} and f_{min} are the maximum and minimum flux measurements respectively, and $f_{i,scaled}$ is the resulting scaled flux:

$$f_{i,scaled} = \frac{f_i - f_{min}}{f_{max} - f_{min}} \tag{5.1}$$

Each stellar spectrum in the dataset collected different amounts of flux measurements. This creates a problem when building the feature matrix, which will be described in Section Feature Matrix from this chapter, because there has to be a set number of features (flux measurements). To overcome this problem, an average number of flux measurements in the first 5,000 spectra was computed. The resulting average number of flux measurements is 3,834. This average is then used to fit each spectrum to a standardized number of flux measurements.

The next pre-processing phase deals with redshift. Redshift causes the flux measurements to shift with respect to wavelength. This causes problems with the feature matrix. The redshift provided in the SDSS dataset is combined with eq. (3.1) to create artificial at rest spectra.

When the wavelengths are represented in logarithmic space, the difference between two adjacent wavelength values is 0.0001 (see [42]). Therefore, the margin for each

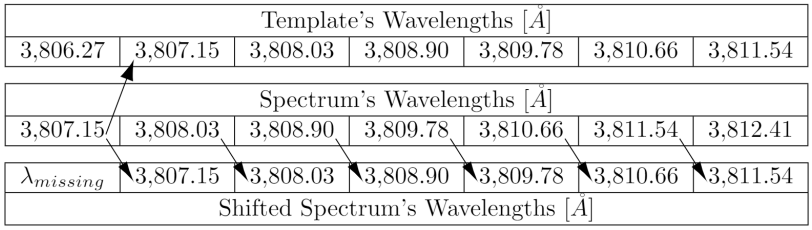


FIGURE 11: Example of fitting a spectrum's wavelengths to the template wavelengths: Right Shifting.

wavelength value is $\lambda \pm 0.00005$. Using the wavelengths in logarithmic space, the artificial rest wavelengths are fitted to the nearest λ, within the margin of 0.00005.

The purpose of the wavelength template is to fit each stellar spectrum to identical wavelength values. This is done because each stellar spectrum stars at different wavelength values [42]. This is to ensure the feature matrix starts and ends with the same wavelength values for every sample. The first spectrum in the database with 3,834 flux measurements and wavelength values was chosen as the template for all spectra wavelength fitting. The wavelengths of this spectrum are then extracted into the template and used to fit each spectrum to a uniform set of wavelength values.

The process of fitting takes the spectrum's first wavelength value and locates it in the template's wavelength array (Fig. 11). For example, if the first wavelength value (index 0) in the spectrum's wavelength array is located at index 1 in the template's wavelength array, then the spectrum's wavelength array is "shifted" to the right. This causes index 0 of the spectrum's wavelength array to be shifted to index 1. If index 0 of the spectrum's wavelength array does not appear in the template's wavelength array, then the fitting process locates the template's index 0 in the spectrum's wavelength array (Fig. 12).

Notice in Figs. 11 and 12 that the shifted spectrum's wavelengths appear in the same column as the template's wavelengths. The shifted spectrum's wavelength array is now fitted to the template. Since each flux measurement is directly associated with a wavelength

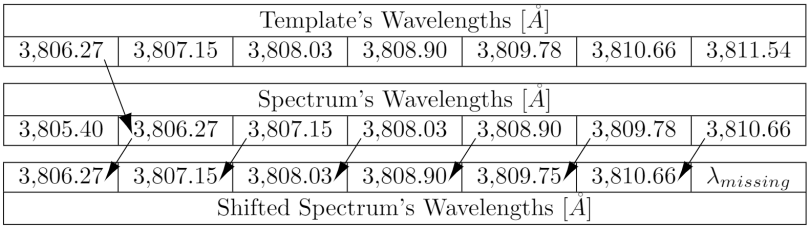


FIGURE 12: Example of fitting a spectrum's wavelengths to the template wavelengths: Left Shifting.

$w_{missing}$	3,807.15	3,808.03	3,808.90	3,809.78	3,810.66	3,811.54
Shifted Spectrum's Wavelengths [$\AA$]						
$f_{missing}$	0.09193	0.73104	0.80613	0.73711	0.80371	0.77967
Shifted Spectrum's Flux [$ergs\ s^{-1}cm^{-2}$]						

FIGURE 13: Example of a spectrum's wavelengths and flux arrays after fitting to a template.

value, the flux array is fitted in parallel to the wavelength arrays. When the wavelength array is shifted to the left or right, the flux array is shifted to the left or right by the same amount, which is apparent in Fig. 13. $w_{missing}$ and $f_{missing}$ are missing values that are caused by the wavelength fitting.

Feature Matrix

Each stellar spectra has two arrays, a flux array that stores flux measurements and a wavelength array that stores the wavelengths that correspond to the flux measurements. The feature matrix is constructed from these arrays with the rows as stellar spectra and the columns as the flux measurements (see Table 1). The wavelength array is used as column headers in the feature matrix.

TABLE 1: Example of the feature matrix

	Wavelength: 3,807.15 $\AA$	Wavelength: 3,808.03 $\AA$	Wavelength: 3,808.90 $\AA$	Wavelength: 3,810.66 $\AA$
Spectrum 1	Flux	Flux	Flux	Flux
Spectrum 2	Flux	Flux	Flux	Flux

As stated in Section Data pre-processing from this chapter, each stellar spectra starts at different wavelength values. To ensure that each column of the feature matrix represents flux measured at the same wavelength, each wavelength array is fitted to the wavelength array template.

When the wavelength array is being fitted to the wavelength template array, it can create missing values (represented by $\lambda_{missing}$), as seen in Figs. 11 and 12. When a wavelength array is right shifted, missing values are guaranteed to form at the beginning. When a wavelength array is left shifted, a missing value forms if the last wavelength value in the spectrum is smaller than the last wavelength value in the template. After shifting the spectrum's wavelength and flux arrays, if the array length is larger than 3,834, then the spectrum's wavelength and flux arrays are cut off at index 3,833 to ensure all spectra have the same number of flux measurements.

Missing wavelength values are replaced with the values from the template's wavelength array. Missing flux values (represented by $f_{missing}$) are replaced using the average of the next or last *K* flux measurements. A moving average is used to ensure the continuation of the trend and that no artificial absorption lines are created. Artificial absorption lines can lead to misclassification.

When the missing flux measurement is at the beginning of the flux array, the next *K* flux measurements are averaged using eq. (5.2), where *j* is the index of $f_{missing}$ being replaced. The sequence of missing values at the beginning of the flux array is as follows: [$\bar{f}_0$, $\bar{f}_1$, ..., $\bar{f}_w$], where *w* is the index of the last missing value. The missing values are replaced from index *w* to index 0. The computed moving average includes any estimated missing values.

$$\bar{f}_j = \frac{1}{K} \sum_{i=j+1}^{K+j+1} flux_i \tag{5.2}$$

TABLE 2: Example of the feature matrix that has redshifted data

Star Class	Wavelength: 6,560.8 $\mathring{A}$	Wavelength: 6,561.8 $\mathring{A}$	Wavelength: 6,562.8 $\mathring{A}$	Wavelength: 6,563.8 $\mathring{A}$	Wavelength: 6,564.8 $\mathring{A}$
A0			Hα		
A0		Hα			
A0				Hα	
A0					Hα
A0	Hα				

When the missing flux measurement is at the end of the flux array, the last *K* flux measurements are averaged using eq. (5.3), where *j* is the index of $f_{missing}$ being replaced and $j > K$. The sequence of missing values at the end of the flux array is as follows: [$\bar{f}_v$, $\bar{f}_{v+1}$, ..., $\bar{f}_{3,833}$], where *v* is the index of the first missing value. The missing values are replaced starting at index *v* to index 3,833. The moving average when computed includes any estimated missing values.

$$\bar{f}_j = \frac{1}{K} \sum_{i=K-j-1}^{j-1} flux_i \tag{5.3}$$

A problem arises with the feature matrix because redshift causes the flux measurements to be shifted in wavelength. This causes the columns of the feature matrix to have the same flux measurement at different wavelengths for the same class. Certain flux measurements have real physical meaning such as the absorption lines. For example, the Hydrogen Alpha (Hα) absorption line is observed at rest with wavelength at 6,562.8 $\mathring{A}$ in air, but due to redshift, the Hα line can be observed at other wavelengths such as 6,563.8 $\mathring{A}$ or 6,561.8 $\mathring{A}$. An example of how redshift changes the meaning of each column is seen in Table 2. Notice how Hα appears in different columns of the feature matrix due to redshift for the same stellar class. This is comparable to taking the petal width of the iris dataset's [43] feature matrix and putting it in the petal length column.

As described in Chapter III: Stellar Spectra and Stellar Classification Types, absorption lines are a key parameter in classifying stars with the Harvard spectral classification system. The classification of a star is associated with the expected wavelength at which certain absorption lines occur. However, to redshift correct a spectrum, the redshift is computed using eq. (3.1), where the wavelengths of known absorption lines are used. By definition, knowing the absorption lines to perform redshift corrections also means that the class of the star is known. The other approach to determine the redshift of a spectrum utilizes statistical analysis to compare the spectra to templates, which as a by-product the class of the star is known.

Machine learning approaches to classifying stars with redshift corrected data are redundant because the by-product of redshift correcting a spectrum is the star's class.

The redshift of stars in the SDSS dataset is small, Z is on the order of magnitude of 0.0001. As such, absorption lines, key features of the dataset, have similar wavelengths for samples of similar class. Classification with a large number of stellar spectra of each class with a large range of redshift values overcome the redshift problem.

Feature Selection and Classification Methods

Chi-Squared [44] and Fisher [36] filter feature selection algorithms are used. For a large dataset as SDSS, the Chi-Squared and Fisher methods performed equally.

RF and SVM classifiers are chosen because other works using these two classifiers using stellar spectra collected from different sources reported good results. For instance, Xing and Guo [4] used a SVM to classify stellar spectra and got 93.26%. Yi and Pan [6] applied RF to the classification of stellar spectra and got a RMSE of 1.04.

Experiments

In the following, the results of the Chi-Squared and Fisher feature selection algorithm in conjunction with the RF and SVM classifiers are presented. The resulting best feature selection and classifier algorithm pair is then expanded on with its precision, recall, and F1 score statistics.

The results are then compared for redshifted spectra and artificial rest spectra. Artificial rest spectra are used as a baseline for comparison for the redshifted spectra.

Misclassification costs are not explicitly considered. However, misclassification costs are not expected to be more consequential than misclassification costs from Human classified spectra.

Implementation

The experiments presented in this chapter are implemented using Python and scikit-learn [44]. Due to the size of the imbalanced raw dataset (35.4 GB per dataset), the Python NumPy memmap [45, 46] module was utilized to read very large arrays from storage rather than RAM. The experiments are performed on an IBM S822LC cluster with IBM POWER 8 nodes, NVLink and NVidia Tesla P100 GPUs [47].

RF and SVM are implemented with scikit-learn with default parameters [44]. SVM is implemented using a Gaussian kernel. Precision, recall, and F1 score are computed using functions implemented by the scikit-learn sklearn.metrics package [44]. The Chi-Squared feature selection algorithm is implemented using scikit-learn [44]. The Fisher feature selection algorithm is implemented using scikit-feature [36].

The first step of the experiments is to perform data pre-processing, as described in Section Data pre-processing from this chapter. The flux is scaled to ensure that similar classes have similar flux measurements using eq. (5.1). There are two datasets used in this chapter, redshifted spectra and artificial rest spectra. Both datasets are created from the SDSS dataset. For artificial rest spectra, the wavelengths are adjusted using eq. (3.1) through the process described in Section Data pre-processing. For redshifted spectra, this process is forgone because all of the spectra in the SDSS database are redshifted. The remaining parts of the data pre-processing are applied to both datasets.

The wavelengths are fitted to a template to ensure that each spectrum's wavelength coverage is identical, which is supported by the work presented by Bazarghan and Gupta [9]. This process is described in Section 5.1 from this chapter. Since each flux measurement is associated with a wavelength value when the wavelengths are fitted to the template, the flux array gets transformed in parallel. The process of fitting the wavelengths to the template and the flux array transformation causes missing values to appear. These missing values are approximated using a moving average, as described in Section Feature Matrix, and by eq. (5.2) and eq. (5.3).

Due to the number of samples obtained after hybrid sampling (367,004 samples), a random balanced subset of 110,000 samples is used to compute the feature rankings for Chi-Squared and Fisher feature selection methods. After the feature selection phase, the RF and SVM classifiers are applied to the resulted reduced feature matrix. Due to the large execution time for SVM (approximately 21 hours for the first experiment) and the higher accuracy of RF, SVM is not computed for hybrid sampling. The accuracy, precision, recall, and F1 score are computed. In each experiment 10-fold cross validation is used to split the dataset into test and training sets.

Discussion

Figures 14, 15, 16 and 17 show that classification using the redshift spectra has essentially the same accuracy as the artificial rest spectra, despite the fact that redshift has introduced problematic data into the feature matrix. Figs. 14 and 16 also demonstrate that RF is the best classifier for both redshifted spectra and artificial rest spectra.

Tables 3 and 5 show that the Fisher method performs better than Chi-Squared method for both redshifted spectra and artificial rest spectra when using the RF classifier. From Tables 4 and 6 observe that with the Fisher feature selection method obtains roughly the same accuracy as with the Chi-Squared method. However, Table 8 demonstrates that Chi-Squared has a significantly shorter execution times than Fisher.

Tables 3, 4, 5, and 6 demonstrate that hybrid sampling outperforms undersampling. Tables 3 and 4 show that accuracy increases with more samples. Supplementing SDSS data run 14 with samples from SDSS data run 12 and 13 further may improve accuracy.

Figure 18 shows that the precision, recall, and F1 scores for redshifted spectra are lower than for artificial rest spectra. However, Table 7 show that the precision for redshifted spectra is only lower by 0.0045, recall is only lower by 0.0045 and the F1 score is lower by 0.0045 (for 500 features). This means that classification using redshifted spectra is just as accurate as rest spectra.

Redshifted spectra achieved an accuracy of 96.87% using the Fisher feature selection with hybrid sampling using the RF classifier applied to 500 selected flux measurements. Using the same work flow, artificial rest spectra obtained 97.32% accuracy. These results are superior by more than three percent than the ones achieved by Xing *et al.* [4], since they reported 93.26% accuracy using wavelet reduction and SVMs for Jacoby [8] spectra.

This approach takes considerably fewer steps than the one in Bolton *et al.* [2] and produces excellent results. The execution times and the obtained accuracy demonstrate

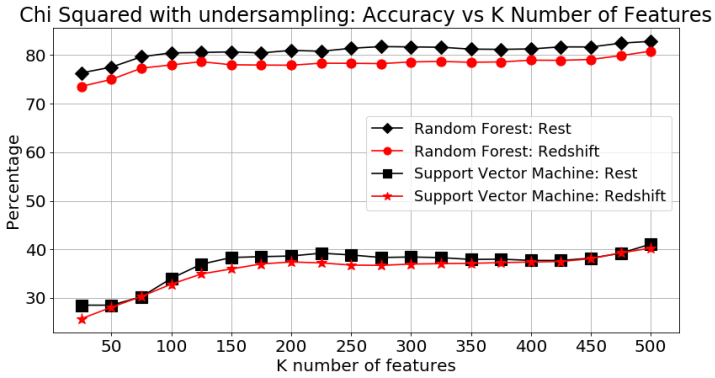


FIGURE 14: 10-Fold cross validation results for Chi-Squared feature selection with undersampling for redshifted spectra and artificial rest spectra.

TABLE 3: 10-Fold cross validation results for Chi-Squared feature selection with undersampling (12,584 samples) for redshifted spectra and artificial rest spectra.

Classifier	Accuracy (%) for Chi-Squared for K number of features									
	50.0	100.0	150.0	200.0	250.0	300.0	350.0	400.0	450.0	500.0
redshifted spectra										
RF	76.60	80.81	80.66	80.88	80.92	81.64	81.99	81.79	82.13	83.59
SVM	28.33	33.60	37.64	37.95	38.73	39.01	39.02	38.97	39.13	40.41
artificial rest spectra										
RF	81.81	86.17	86.03	86.13	86.72	86.85	86.82	86.89	86.92	88.24
SVM	29.13	35.69	38.63	39.13	39.59	39.82	40.27	39.87	39.68	41.27

that, for a real application of this work, the automated classification of redshifted stellar spectra into the Harvard classification scheme using a single classifier not only achieves a high accuracy but is also fast.

However, the approach presented in this chapter only classifies into the Harvard classification scheme, and as stated in Chapter III: Importance of Stellar Classes, it would be better to classify into both the Harvard and MK classification schemes. The next chapter deals with just that.

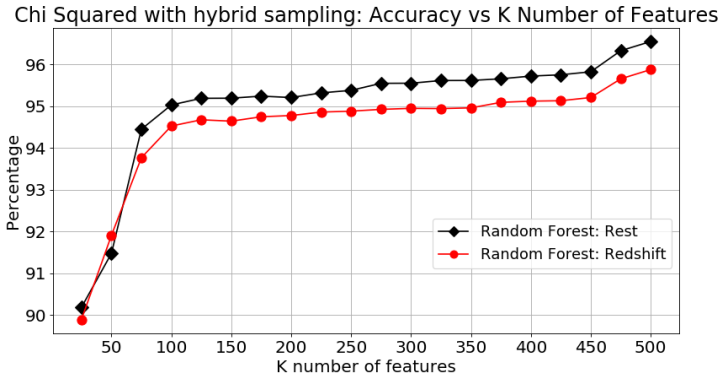


FIGURE 15: 10-Fold cross validation results for Chi-Squared feature selection with hybrid sampling for redshifted spectra and artificial rest spectra spectra.

TABLE 4: 10-Fold cross validation results for Chi-Squared feature selection with hybrid sampling (367,004 samples) for redshifted spectra and artificial rest spectra using Random Forest.

Accuracy (%) for Chi-Squared for K number of features									
50.0	100.0	150.0	200.0	250.0	300.0	350.0	400.0	450.0	500.0
redshifted spectra									
91.24	95.11	95.25	95.24	95.46	95.62	95.65	95.76	95.90	96.55
artificial rest spectra									
94.34	97.02	97.03	97.14	97.31	97.45	97.43	97.54	97.65	98.21

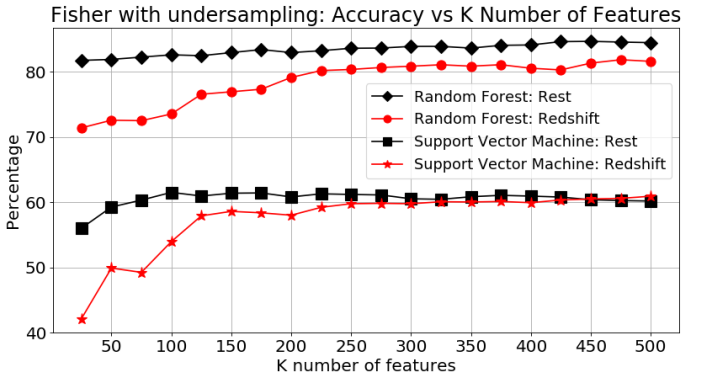


FIGURE 16: 10-Fold cross validation results for Fisher feature selection with undersampling for redshifted spectra and artificial rest spectra.

TABLE 5: 10-Fold Cross Validation Results for Fisher feature selection with undersampling (12,584 samples) for redshifted spectra and artificial rest spectra.

Classifier	Accuracy (%) for Fisher for K number of features									
	50.0	100.0	150.0	200.0	250.0	300.0	350.0	400.0	450.0	500.0
redshifted spectra										
RF	81.177	82.00	82.79	83.51	83.45	83.73	83.26	83.90	84.52	84.95
SVM	60.43	66.51	65.62	64.83	64.44	64.48	65.21	65.33	64.94	64.66
artificial rest spectra										
RF	83.64	83.46	85.29	86.29	86.27	86.19	86.43	87.16	87.19	87.40
SVM	55.63	56.06	57.29	58.73	61.02	62.23	62.59	63.21	63.63	63.81

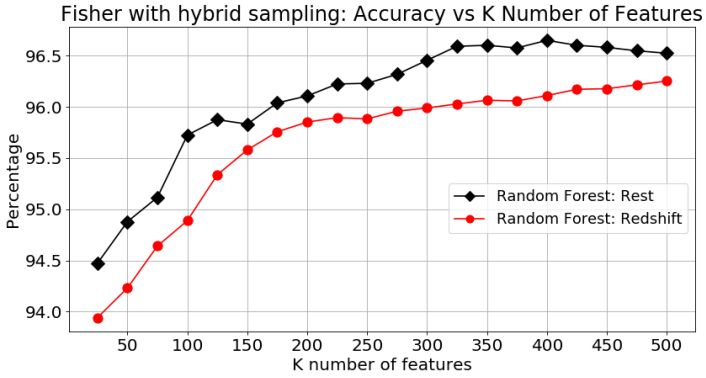


FIGURE 17: 10-Fold cross validation results for Fisher feature selection with hybrid sampling for redshifted spectra and artificial rest spectra.

TABLE 6: 10-Fold cross validation results for Fisher feature selection with hybrid sampling (367,004 samples) for redshifted spectra and artificial rest spectra using Random Forest.

Accuracy (%) for Fisher for K number of features									
50.0	100.0	150.0	200.0	250.0	300.0	350.0	400.0	450.0	500.0
redshifted spectra									
94.65	95.89	96.22	96.38	96.49	96.59	96.62	96.74	96.79	96.87
artificial rest spectra									
96.30	97.22	97.51	97.59	97.58	97.44	97.37	97.36	97.32	97.32

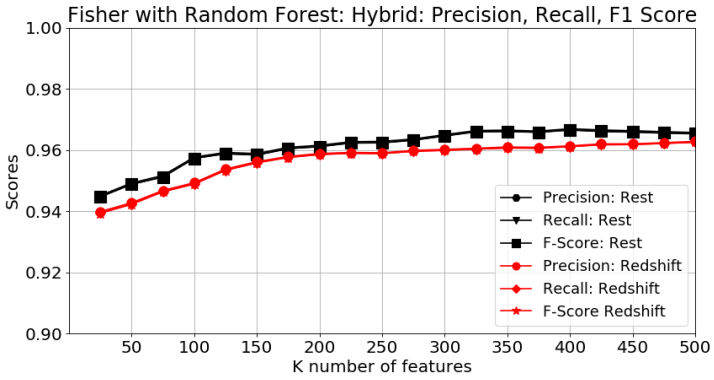


FIGURE 18: 10-Fold cross validation results for Fisher feature selection for random forest.

TABLE 7: Precision, Recall, and F1 Score for Fisher feature selection with Hybrid for Random Forest.

	Results for Fisher + Random Forest for K number of features									
	50.0	100.0	150.0	200.0	250.0	300.0	350.0	400.0	450.0	500.0
redshifted spectra										
Precision	0.9474	0.9593	0.9626	0.9642	0.9652	0.9662	0.9665	0.9677	0.9682	0.9690
Recall	0.9465	0.9589	0.9622	0.9638	0.9649	0.9659	0.9662	0.9674	0.9679	0.9687
F1 Score	0.9467	0.9589	0.9622	0.9639	0.9649	0.9659	0.9662	0.9674	0.9680	0.9687
artificial rest spectra										
Precision	0.9634	0.9724	0.9753	0.9761	0.9760	0.9746	0.9740	0.9739	0.9735	0.9735
Recall	0.9630	0.9722	0.9751	0.9759	0.9758	0.9744	0.9737	0.9736	0.9732	0.9732
F1 Score	0.9630	0.9722	0.9751	0.9759	0.9758	0.9744	0.9738	0.9736	0.9732	0.9732

TABLE 8: Execution time for hybrid sampling for random forest with artificial rest spectra.

	Execution times (Seconds) for K number of features									
	50.0	100.0	150.0	200.0	250.0	300.0	350.0	400.0	450.0	500.0
Fisher + artificial rest spectra										
Feature Selection	7.94×10^5	7.94×10^5	7.94×10^5	7.94×10^5	7.94×10^5	7.94×10^5	7.94×10^5	7.94×10^5	7.94×10^5	7.94×10^5
Train	39.83	55.49	66.23	76.77	82.10	93.07	99.46	110.21	115.85	121.24
Test	0.65	0.58	0.61	0.62	0.63	0.68	0.69	0.70	0.75	0.79
Chi-Squared + artificial rest spectra										
Feature Selection	6.84	7.98	9.12	10.41	11.21	12.40	13.24	14.72	16.13	17.48
Train	37.11	50.08	60.03	69.79	75.07	84.80	90.28	100.28	105.45	109.06
Test	0.57	0.55	0.57	0.60	0.62	0.64	0.68	0.70	0.73	0.74

CHAPTER VI

SINGLE CLASSIFICATION INTO BOTH HARVARD AND MK CLASSIFICATION SCHEMES

In contrast to Chapter V where stellar spectra are classified only into the Harvard classification scheme, this chapter describes the experiments to classify the stellar spectra into both the Harvard and MK classification schemes using a single classifier method. The contents of this chapter are as follows. Section Approach to Classification describes the data, data pre-processing, feature selection, the feature matrix, and the classifier methods used. Section Experiments explains the implementation of the feature selection and classifier methods and the experimental framework and results. Section Discussion analyzes the results.

Approach to Classification

Data

The dataset used in these experiments comes from SDSS data run 12, 13, and 14, which was collected using the BOSS spectrograph [39]. The data runs collected a combined total of 892,614 spectra. The data was rejected for similar reasons described in Chapter V. Similarly the data was also pre-processed by SDSS scientists through the methods presented by Dawson et al [37] and Stoughton et al [38].

The usable dataset contains 453,378 stellar spectra and 46 of the 420 Harvard + MK class combinations. However, it is important to note that the majority of stars have a MK class of V and that not every combination is common. It is also important to note that this is real collected data and not simulated data. The distribution of the classes from the dataset

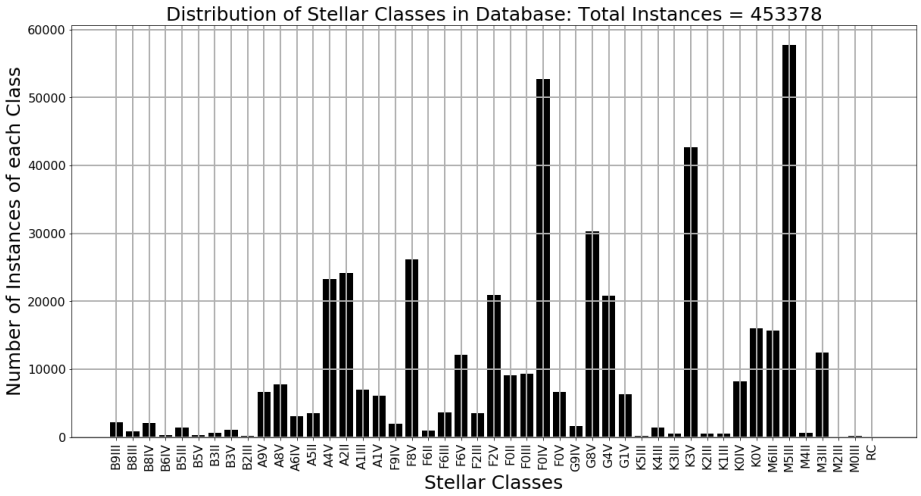


FIGURE 19: Distribution of classes in the dataset. RC = Remaining Classes of 0 instances

is imbalanced, as seen in Fig. 19. The dataset is balanced using an *Undersampling* method [41] and a *Hybrid* method. An *Oversampling* method was used to improve accuracy. This technique is supported by the findings presented in Chapter V. The Undersampling method results in each class having 55 samples for a total of 2,530 samples. The Oversampling method results in each class having 57,771 samples for a total of 2,657,466 samples. The Hybrid method results in each class having 20,813 samples for a total of 957,398 samples. For Hybrid and Oversampling methods, when a sample is duplicated the duplicate gets a randomly generated new redshift, which makes it a unique sample.

Data pre-processing

The only data pre-processing required for this approach is flux scaling using eq. (5.1). If a sample is known to have missing or corrupt flux measurements around the absorption lines used for Feature Selection (see Chapter VI: Approach to Classification: Feature Selection), then an Imputer method is required to fill in missing values in the feature matrix . None of the samples in the dataset used in this analysis required an Imputer.

Feature Selection

Domain knowledge is used to perform Feature Selection (Algorithm 1). As stated in Chapter III: Stellar Spectra, key features of stellar spectra are the absorption lines. Using the standard source "The Classification of Stars" [48], an analysis was performed to identify the absorption lines present in each class of the Harvard Spectral Classification Scheme.

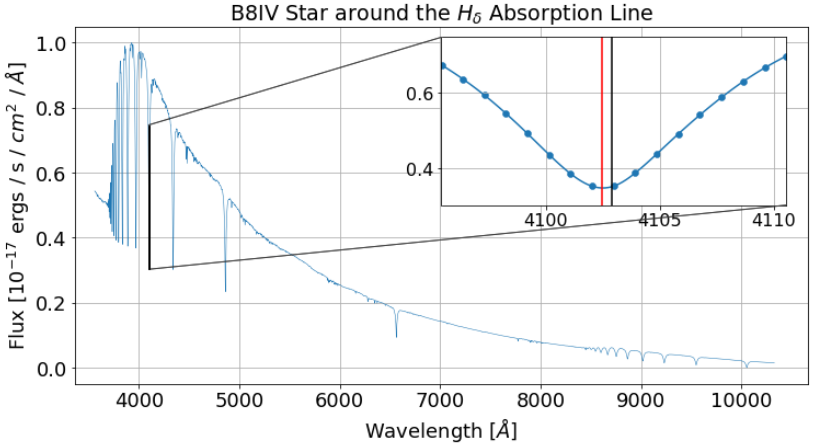
Redshift makes it impossible to identify the wavelength of any absorption line that is in any given sample. However, the rest wavelength of any absorption line is known. The known rest wavelength of an absorption line can be used to create a set of features (flux measurements) for classification. These set of features are generated by using flux measurements surrounding the rest wavelengths of known absorption lines. It is important to note that the absorption line does not have to be present in every sample/class, it is its rest wavelength that is important. The absorption lines used in this study are the H_δ (4102 $\mathring{A}$) and Ca I (4227 $\mathring{A}$) lines. Using these two absorption lines, the variability in the intensity and shape of the flux measurements in these regions can separate the samples in the Harvard classes and the widths of the line that is present in one or both regions create separability in the samples for the MK classes.

Figures 20, 21, 24, 25, 22, 23, and 26 demonstrate how using only the H_δ and Ca I lines preserve variability in the intensity, shape, redshift, and widths of the absorption lines needed to classify into the Harvard and MK luminosity classes. The red line is the absorption line at redshift wavelengths, black is at rest wavelengths, and no line means there is no absorption line. Notice in Figs. 20 and 21 (B and A stars) that only the H_δ absorption line is present, whereas Figs. 22 and 23 (K and M stars) only the Ca I absorption line is present, and Figs. 24 and 25 (F and G stars) both are present. Figure 27 shows how redshift is preserved because the absorption line is still found within the search space. Figures 20, 21, 24, 25, 22, and 23 show that combining these two sets of flux measurements around

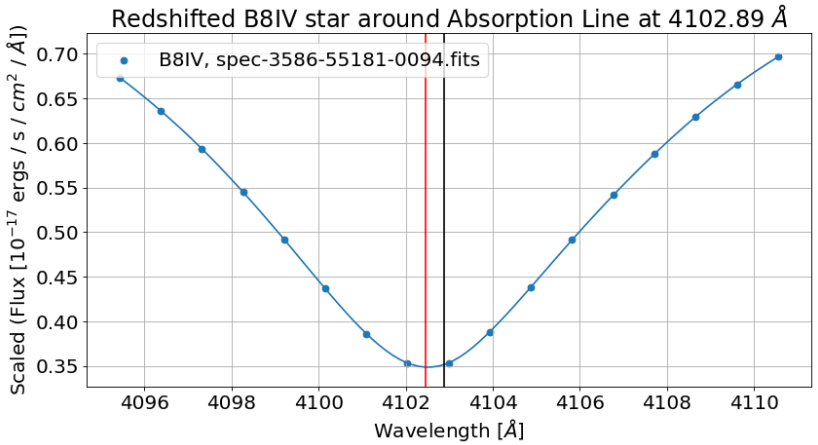
these absorption lines creates separable samples for the Harvard classification. This is achieved because every Harvard major class has one or both of these absorption lines.

Figure 26 shows that for two samples with the same Harvard class, the width of the absorption line is preserved. As stated in Chapter III: Stellar Classification Types, the widths of the absorption lines determine the MK classification.

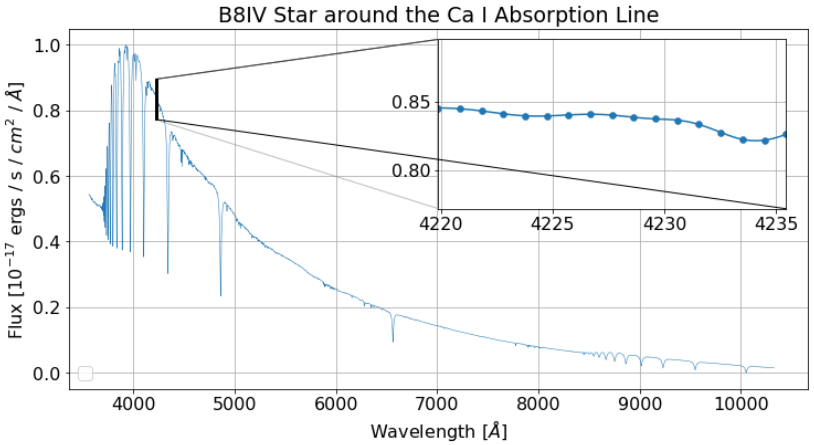
It is important to note that Algorithm 1 is embarrassingly parallel and was implemented in parallel for this chapter. Using these two absorption lines results in a feature matrix as seen in Table 9.



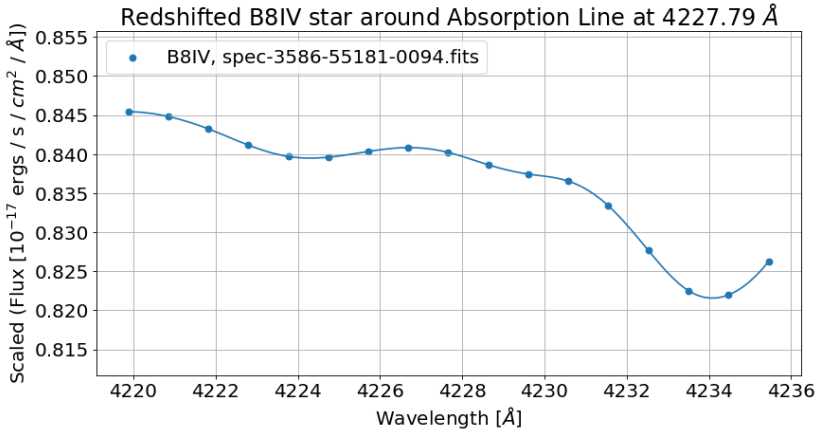
(a) Around the H_δ absorption line.



(b) Subplot of (a).

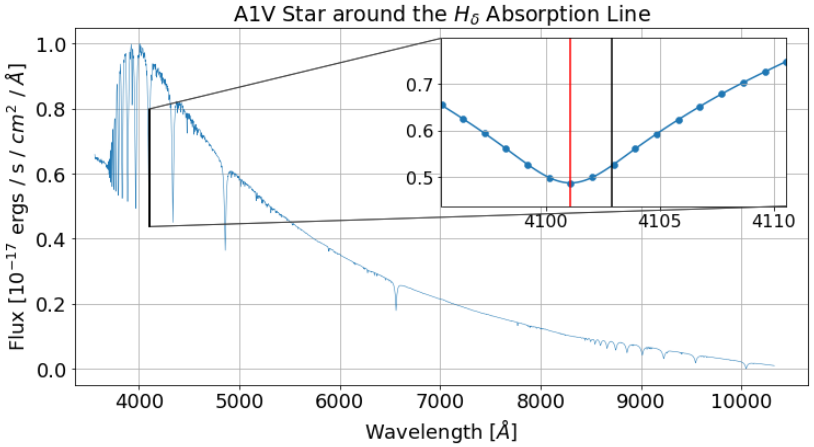


(c) Around the Ca I absorption line.

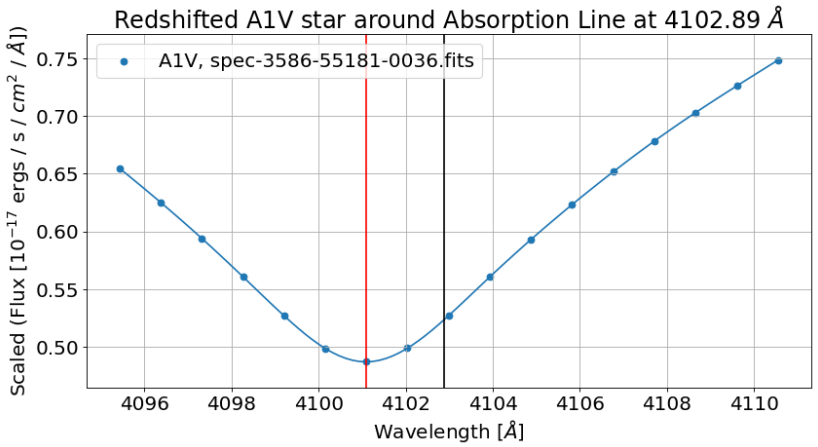


(d) Subplot of (c).

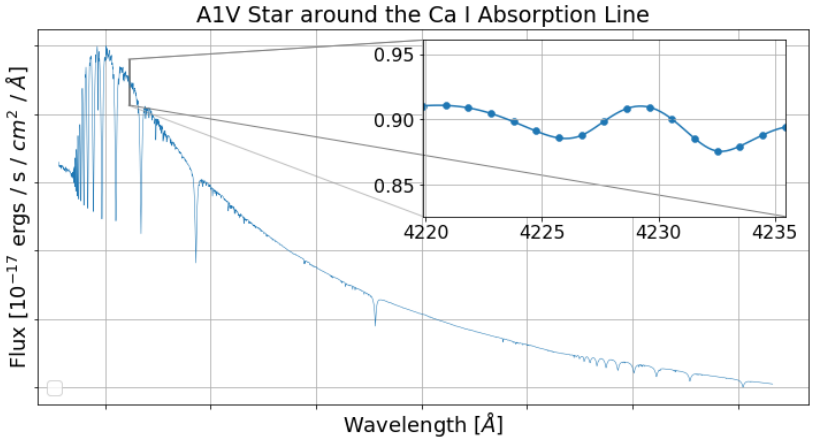
FIGURE 20: Example of a B Type Star focusing on wavelengths near H_δ and Ca I absorption lines.



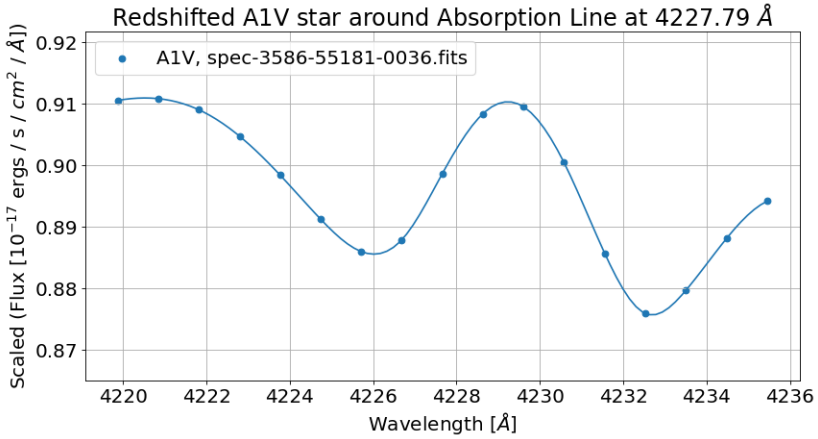
(a) Around the H_δ absorption line.



(b) Subplot of (a).

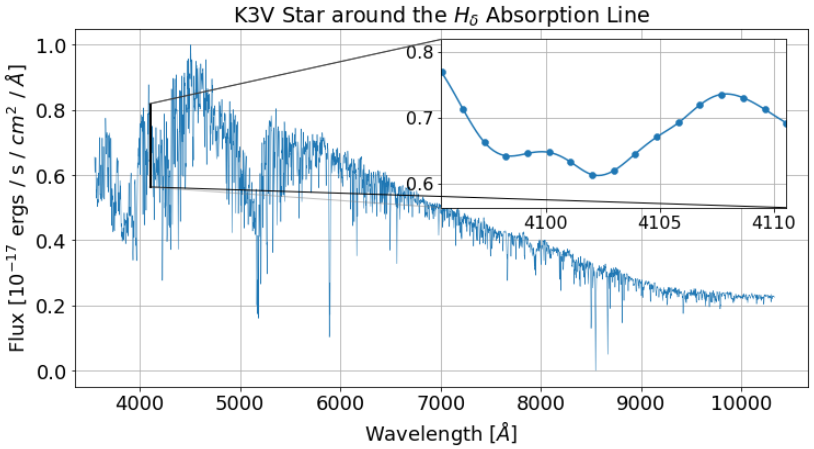


(c) Around the Ca I absorption line.

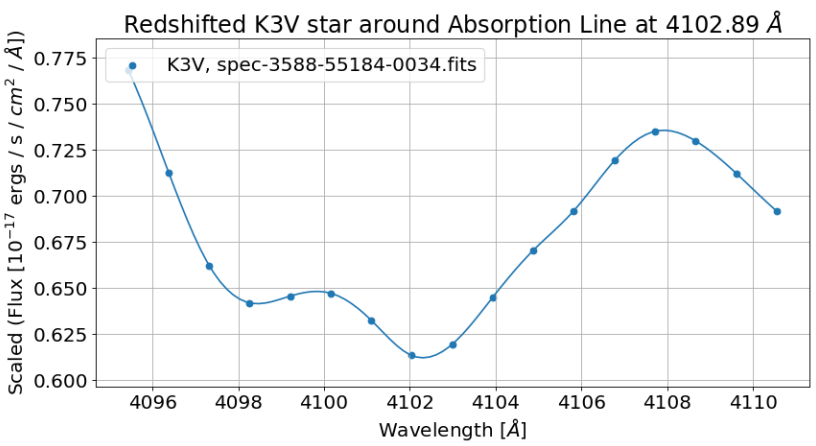


(d) Subplot of (c).

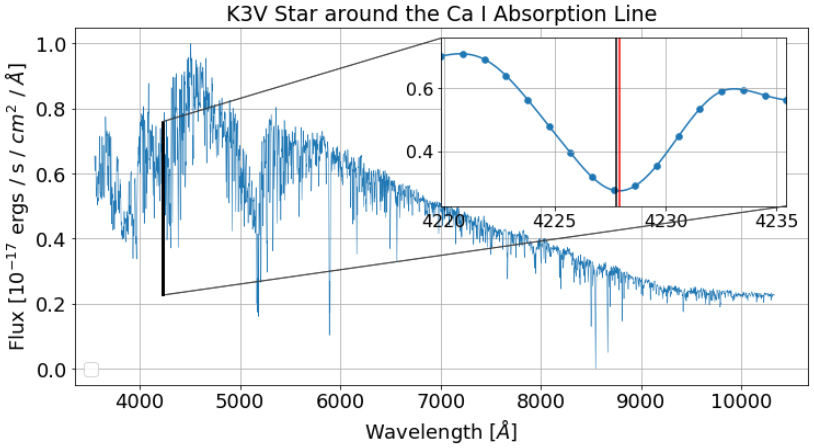
FIGURE 21: Example of an A Type Star focusing on wavelengths near H_δ and Ca I absorption lines.



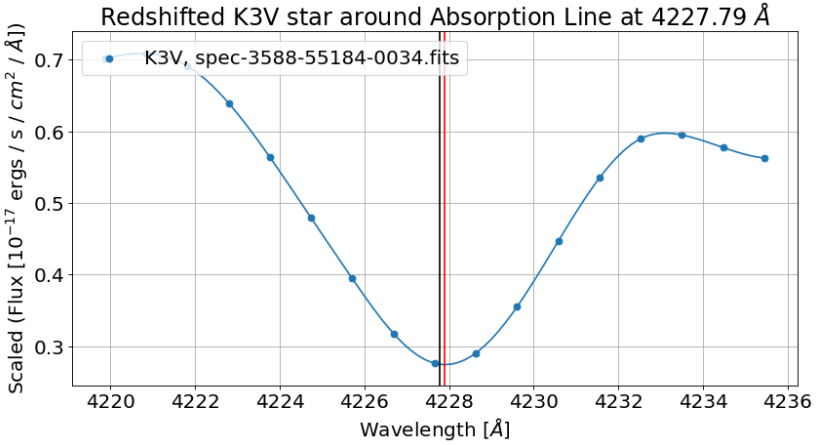
(a) Around the H_δ absorption line.



(b) Subplot of (a).

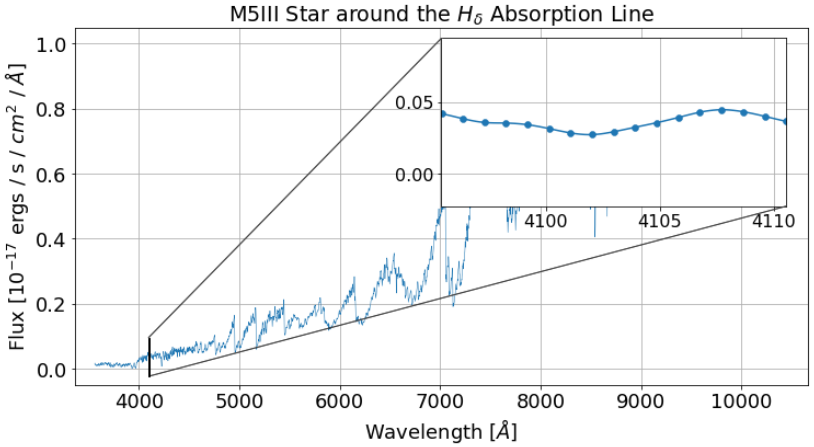


(c) Around the Ca I absorption line.

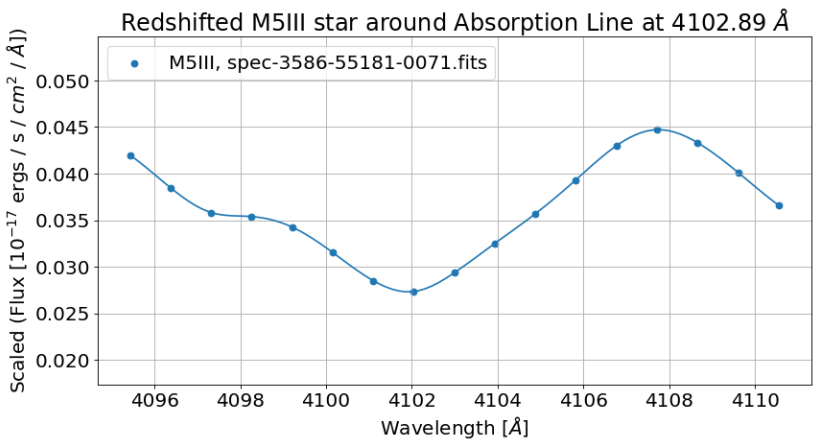


(d) Subplot of (c).

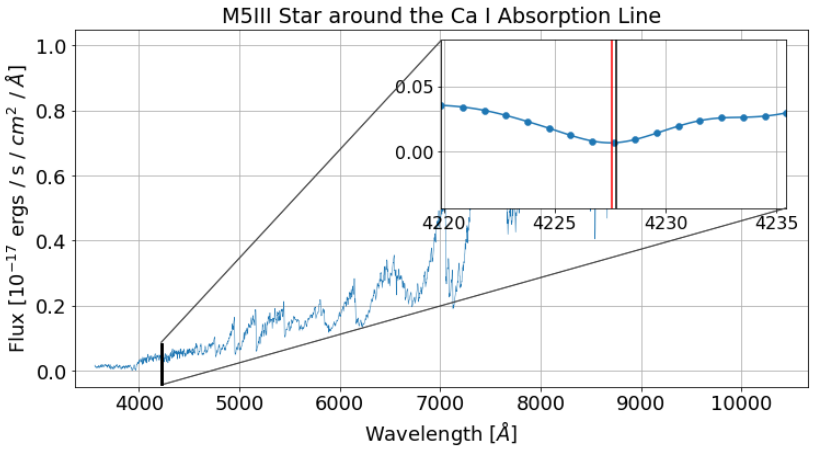
FIGURE 22: Example of a K Type Star focusing on wavelengths near H_δ and Ca I absorption lines.



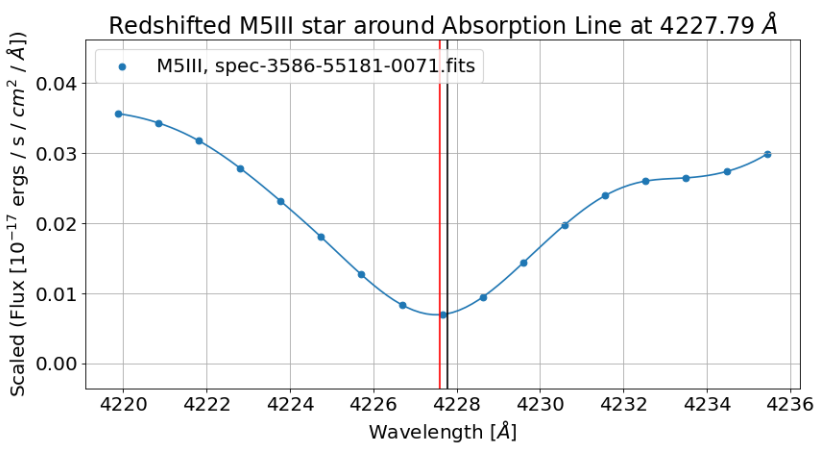
(a) Around the H_δ absorption line.



(b) Subplot of (c).

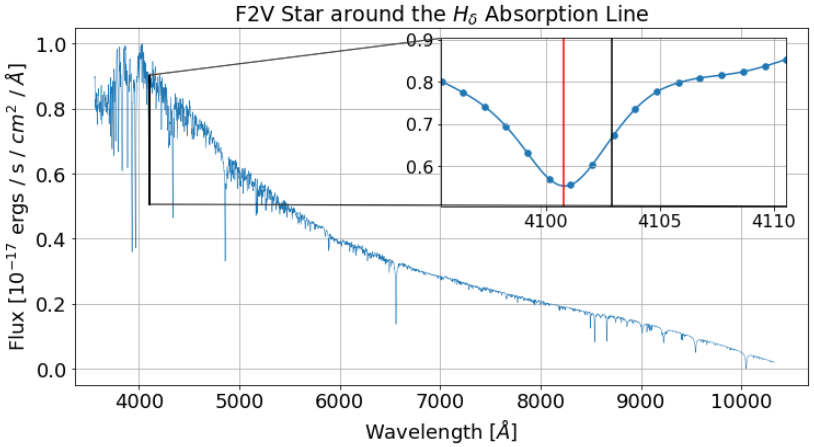


(c) Around the Ca I absorption line.

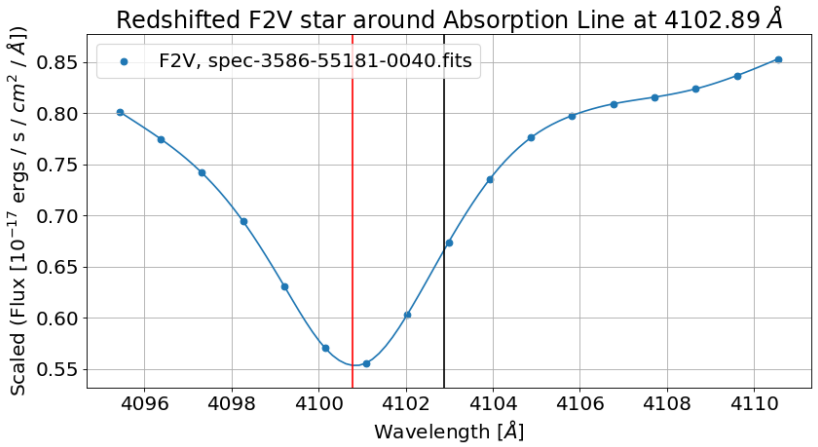


(d) Subplot of (c).

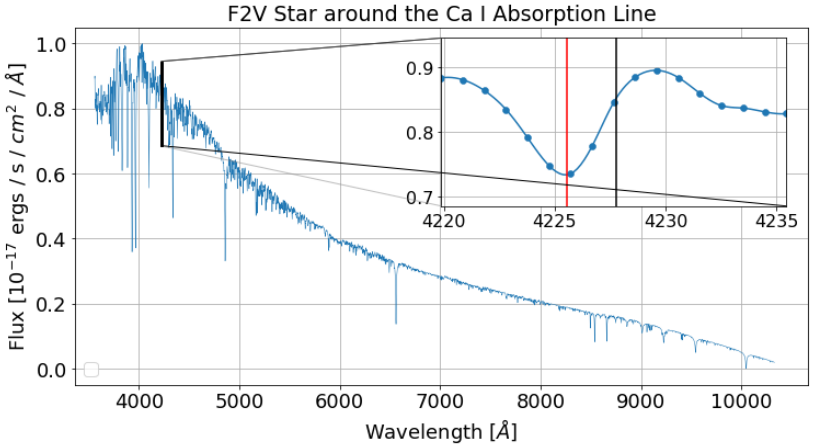
FIGURE 23: Example of a M Type Star focusing on wavelengths near H_δ and Ca I absorption lines.



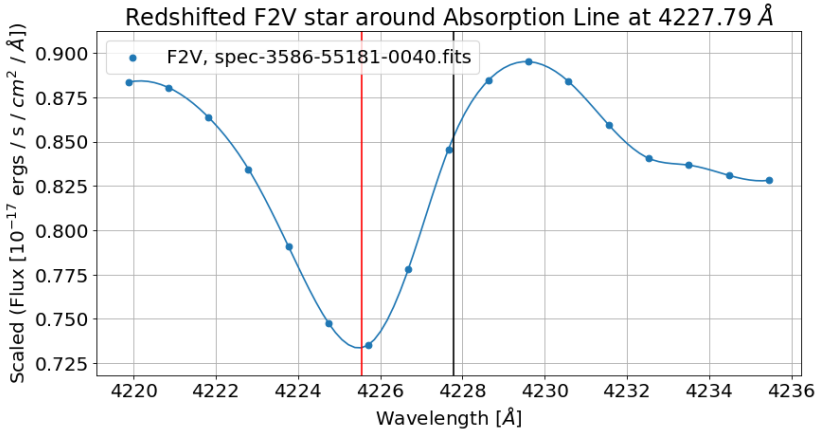
(a) Around the H_δ absorption line.



(b) Subplot of (a).

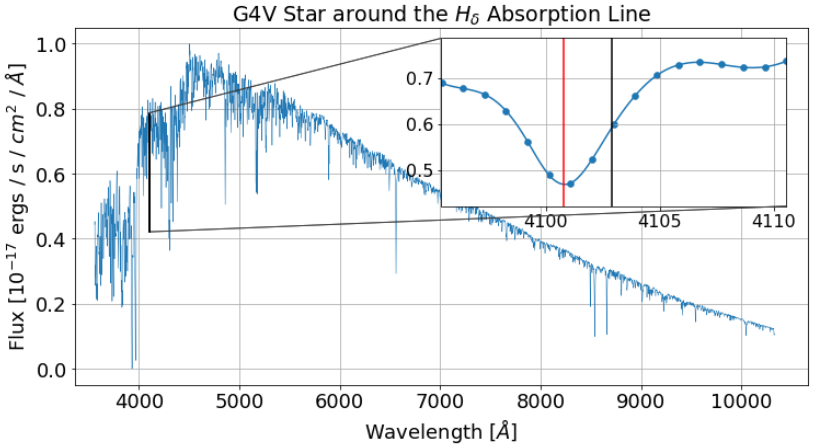


(c) Around the Ca I absorption line.

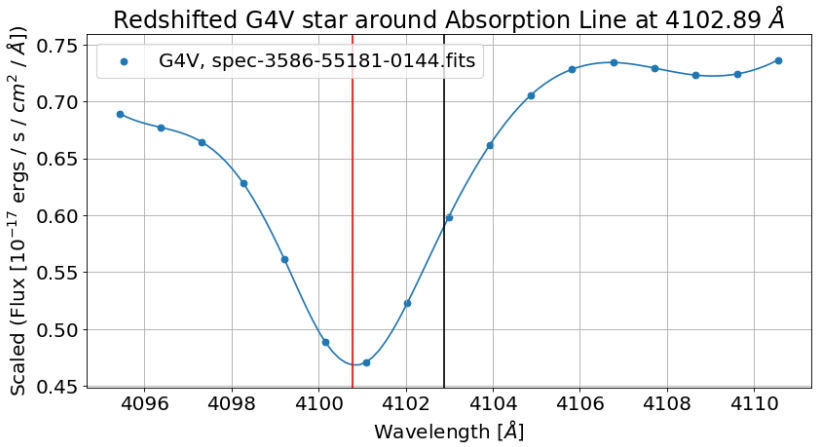


(d) Subplot of (c).

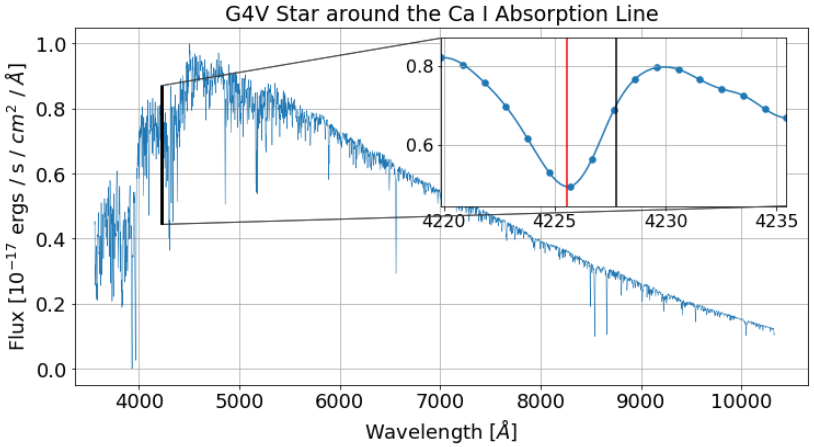
FIGURE 24: Example of a F Type Star focusing on wavelengths near H_δ and Ca I absorption lines.



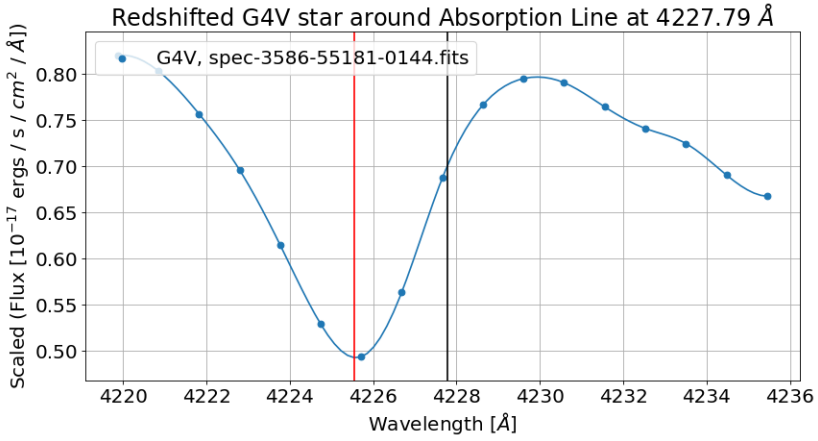
(a) Around the H_δ absorption line.



(b) Subplot of (a).



(c) Around the Ca I absorption line.



(d) Subplot of (c).

FIGURE 25: Example of a G Type Star focusing on wavelengths near H_δ and Ca I absorption lines.

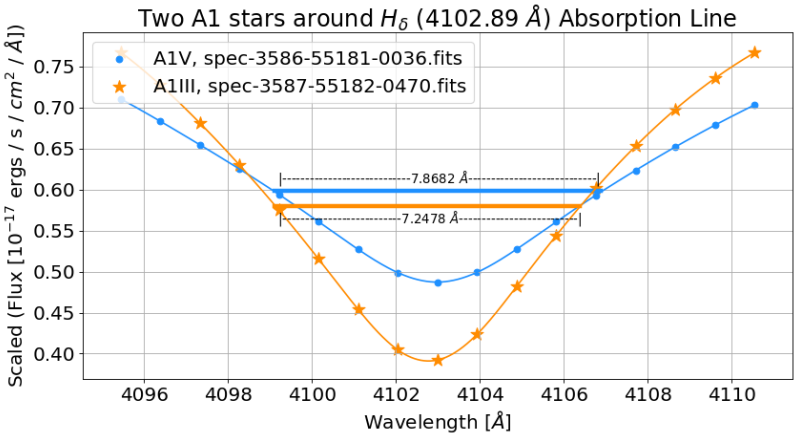


FIGURE 26: Example of the same Harvard class with different wavelength width (Full Width Half Max) for the same absorption line for different MK classes.

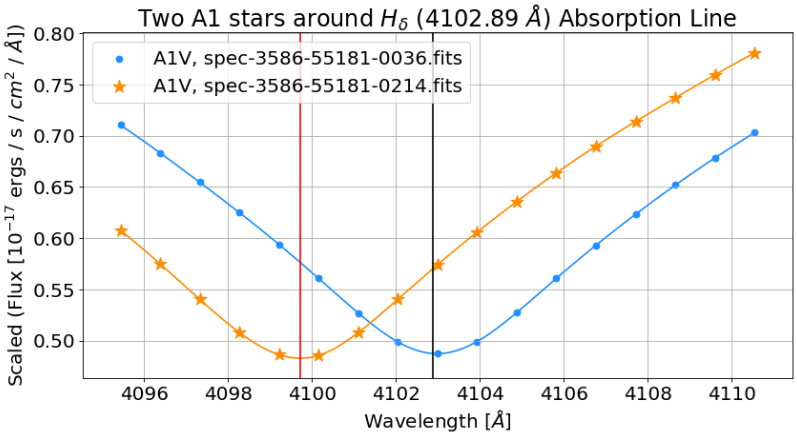


FIGURE 27: Example of how Redshift is preserved.

Algorithm 1 Flux Feature Selection

```
function FEATURE_SELECTION(flux_Arr, wavelength_Arr, absorption_Line)
    bounds = 8 // range of flux measurements before and after the absorption line
    // This can be implemented in parallel because each row (sample) in
     flux_Arr wavelength_Arr do not depend on each other. Each thread/process can be a sample
    for 0 < i < flux_Arr_Length do
        index = Find_Nearest(wavelength_Arr[i], absorption_Line)
        for index − bounds < j < index + bounds do
            new_flux_Arr[i][j] = flux_Arr[i][j]
            new_wavelength_Arr[i][j] = wavelength_Arr[i][j]
        end for
    end for
    return new_flux_Arr, new_wavelength_Arr
end function
function FIND_NEAREST(array, value)
    min = ∞
    index = −1
    for 0 < i < array_Length do
        if array[i] − value < min then
            index = i
        end if
    end for
    return index
end function
procedure MAIN
    // flux_Arr and wavelength_Arr are both n x m arrays where n is the
    number of samples and m is the number of dimensions
    flux_Arr_1, wavelength_Arr_1 = Feature_Selection(flux_Arr,
                                                     wavelength_Arr, 4102.89)
    flux_Arr_2, wavelength_Arr_2 = Feature_Selection(flux_Arr,
                                                     wavelength_Arr, 4227.79)
    flux_Arr = Append(flux_Arr_1, flux_Arr_2)
    wavelength_Arr = Append(wavelength_Arr_1, wavelength_Arr_2)
end procedure
```

TABLE 9: Example of the feature matrix using two sets of wavelengths around two absorption lines for a total of 34 features

	Wavelength: 4,095.43 $\mathring{A}$	...	Wavelength: 4,110.55 $\mathring{A}$	Wavelength: 4,219.88 $\mathring{A}$	...	Wavelength: 4,235.45 $\mathring{A}$
Spectrum 1	Flux	...	Flux	Flux	...	Flux
Spectrum 2	Flux	...	Flux	Flux	...	Flux

Feature Matrix

Each stellar spectra has a wavelength array and an array of corresponding flux measurements. The feature matrix is constructed using 17 flux measurements around the H_δ (4102 $\mathring{A}$) and 17 flux measurements around the Ca I (4227 $\mathring{A}$) absorption line (see Table 9).

A problem arises with the feature matrix because redshift causes the flux measurements to be shifted in wavelength. This is illustrated in Table 2.

Classification Methods

This chapter utilizes KNN and RF classifier methods.

KNN was chosen for these experiments because of its fast computational time in low dimensions (the spectra's dimensions are reduced in this thesis) and for its ability to separate spectra of classes that have extremely similar features. This is apparent when samples in the training space are represented as a N-dimensional point, where minor changes in any of the N dimensions can significantly change its position in the N-dimensional space. KNN was also chosen because of the large number of samples available for training.

Random Forest was chosen for these experiments because of its simplistic manner and for the same reasons as KNN with the training space. Random Forest was also chosen because authors who used similar data reported excellent results [6].

Experiments

In the following experiments, the results of the KNN and RF classifiers are presented. This chapter does not take misclassification costs into consideration.

Implementation

The experiments presented in this chapter are implemented using Python and scikit-learn [44]. Due to the size of the datasets (35.4 GB for hybrid balancing and 98.2 GB for oversampled balancing), the Python NumPy memmap [45, 46] module was utilized to read very large arrays from storage rather than RAM. The experiments are performed on a personal computer with the following relevant specifications: AMD Ryzen 7 1800x 16 logical core CPU, 16 GB RAM, and 1 TB Samsung 860 EVO Solid State Drive.

KNN and RF are implemented with scikit-learn with default parameters [44]. Precision, recall, and F1 score are computed using functions implemented by the scikit-learn sklearn.metrics package [44]. Feature selection is implemented using Algorithm 1 in Python and utilizes Python multiprocessing package [49] for parallelization.

Experimental Framework and Results

The first step is to scale the flux to ensure that similar classes have similar flux measurements using eq. (5.1). There are three datasets used in this chapter, Undersampled, Hybrid, and Oversampled balanced. All three datasets are created from the combined SDSS dataset described in Chapter VI: Approach to Classification: Data.

The datasets go through feature selection described in Chapter VI: Approach to Classification: Feature Selection. After the feature selection phase, the KNN and RF classifiers are applied to the resulting reduced feature matrix. The accuracy, precision,

recall, and F1 score are computed. In each experiment, 10-fold cross validation is used to split the dataset into test and training sets.

Discussion

Figures 28 and 30 and Tables 10 and 13 show that classification using KNN has essentially the same accuracy as RF. These Figs. and Tables demonstrate that using KNN and RF alongside Algorithm 1 for feature selection are viable options for the automated classification of stellar spectra because of the high accuracy achieved.

Figure 28 and Table 10 demonstrate that using three neighbors for classification performs the best. Figure 30 and Table 13 shows that changing the number of trees used in RF does not significantly change the classification accuracy. Figures 28 and 30 show that Oversampling balancing performs best. However, Tables 10 and 13 shows that Oversampling balancing (2,657,466 samples) only outperforms Hybrid balancing (957,398 samples) by roughly one percent. Figures 29 and 31 and Tables 11 and 14 show that the Precision, Recall, and F1 Score are all roughly the same for each experiment.

Tables 12 and 15 show the execution times for each experiment. These tables show that KNN performs much faster than RF. KNN has a faster train time than RF. but RF has a faster test time than KNN. For both KNN and RF, feature selection takes the same amount of time, which is expected since they both use the same feature selection.

This approach takes considerably fewer steps than the one in Bolton *et al.* [2] and produces excellent results. The execution times and the obtained accuracy demonstrate that, for a real application of this work, the automated classification of redshifted stellar spectra into both the Harvard and MK classification schemes using a single classifier not only achieves a high accuracy but is also fast.

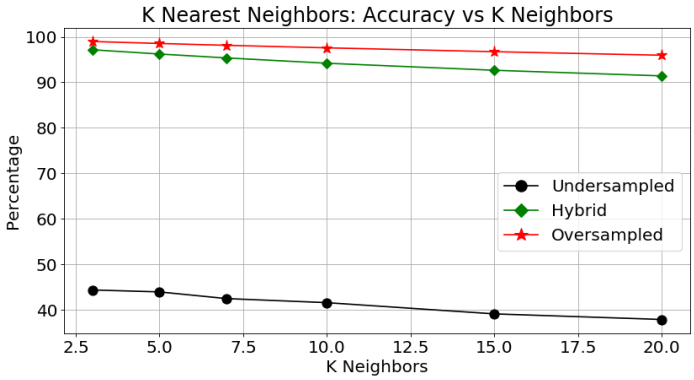


FIGURE 28: 10-Fold cross validation results for K Nearest Neighbors using Undersampling (2,530 samples), Hybrid (957,398 samples), and Oversampling (2,657,466 samples).

TABLE 10: 10-Fold cross validation results for KNN.

Balance Method	Accuracy (%) for K Neighbors					
	3.0	5.0	7.0	10.0	15.0	20.0
Undersampled	44.36	43.93	42.47	41.57	39.11	37.88
Hybrid	97.15	96.22	95.36	94.19	92.65	91.41
Oversampled	98.97	98.54	98.13	97.58	96.73	95.95

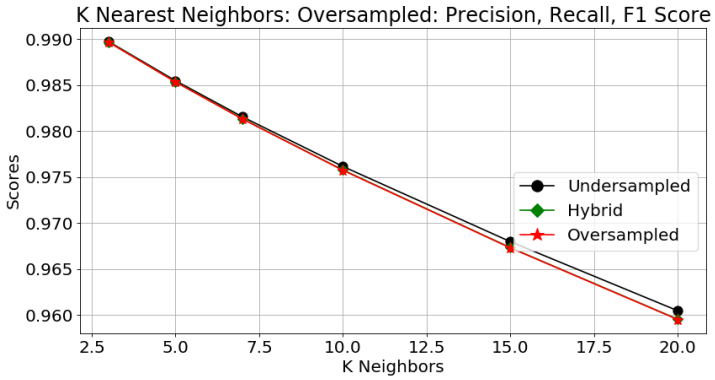


FIGURE 29: Precision, Recall, and F1 Score for Oversampling with K Nearest Neighbors

TABLE 11: 10-Fold cross validation Precision, Recall, and F1 Score for KNN using Oversampling.

	K Neighbors					
	3.0	5.0	7.0	10.0	15.0	20.0
Precision	0.989757	0.985492	0.981557	0.976155	0.967970	0.960448
Recall	0.989707	0.985373	0.981348	0.975772	0.967320	0.959524
F1 Score	0.989692	0.985351	0.981322	0.975739	0.967290	0.959495

TABLE 12: 10-Fold cross validation Execution Times for KNN using Oversampling.

	Time in seconds for K Neighbors					
	3.0	5.0	7.0	10.0	15.0	20.0
Feature Selection	1388.96	1388.96	1388.96	1388.96	1388.96	1388.96
Train	10.81	10.72	10.74	10.68	10.67	10.76
Test	67.89	82.63	95.85	111.24	134.73	156.39

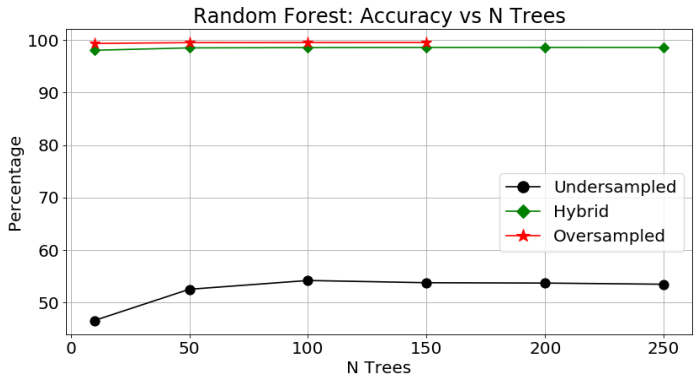


FIGURE 30: 10-Fold cross validation results for Random Forest using Undersampling (2,530 samples), Hybrid (957,398 samples), and Oversampling (2,657,466 samples).

TABLE 13: 10-Fold cross validation results for RF.

Balance Method	Accuracy (%) for N Trees					
	10.0	50.0	100.0	150.0	200.0	250.0
Undersampled	46.61	52.55	54.22	53.80	53.73	53.51
Hybrid	98.06	98.52	98.57	98.60	98.61	98.60
Oversampled	99.36	99.51	99.53	99.54	N/A	N/A

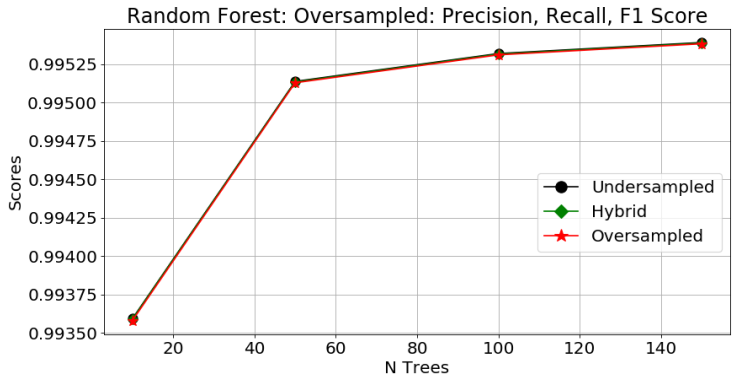


FIGURE 31: Precision, Recall, and F1 Score for Oversampling with Random Forest

TABLE 14: 10-Fold cross validation Precision, Recall, and F1 Score for RF using Oversampling.

	N Trees					
	10.0	50.0	100.0	150.0	200.0	250.0
Precision	0.993592	0.995137	0.995318	0.995390	N/A	N/A
Recall	0.993589	0.995133	0.995315	0.995387	N/A	N/A
F1 Score	0.993580	0.995128	0.995310	0.995382	N/A	N/A

TABLE 15: 10-Fold cross validation Execution Times for RF using Oversampling.

	Time in seconds for N Trees					
	10.0	50.0	100.0	150.0	200.0	250.0
Feature Selection	1388.96	1388.96	1388.96	1388.96	N/A	N/A
Train	204.08	1013.98	2039.38	3068.80	N/A	N/A
Test	5.77	22.31	42.26	61.65	N/A	N/A

As described above in Section Approach to Classification: Data, this chapter deals with data collected by a real astronomical survey. As such, when an astronomical survey points their telescopes into the sky, they get the samples (classes) that they get. This chapter deals with a subset of all possible class combinations. It is important to note that not all possible class combinations (O, B, A, F, G, K, and M with subclasses of 0 - 9 combined with I, II, III, IV, V, VII) are common or even found in nature. Therefore, even though this approach yielded great results, there cannot be a claim that this approach will guarantee work for all stellar classes. There is, however, some theoretical validity to this approach.

Recalling back to Chapter III: Stellar Classification Types, the Harvard classes are based on absorption lines and the MK classes are based on the widths of those absorption lines. Recalling back to Section Feature Selection in this chapter, the shape, intensity, and width of the absorption lines are preserved (seen in Figs. 20, 21, 24, 25, 22, and 23). Referencing Fig. 32, O type stars also contain the H_{δ} absorption line. This strengthens the theoretical validity because the missing Harvard major class would also be represented using this approach.

As stated in Section Feature Selection in this chapter, the widths of the absorption lines are also preserved. This provides strength to the theoretical validity because every Harvard major class would have at least one absorption line preserved and in turn, the widths preserved. This means that this approach should work with the missing Harvard and MK class combinations, assuming the model was retrained.

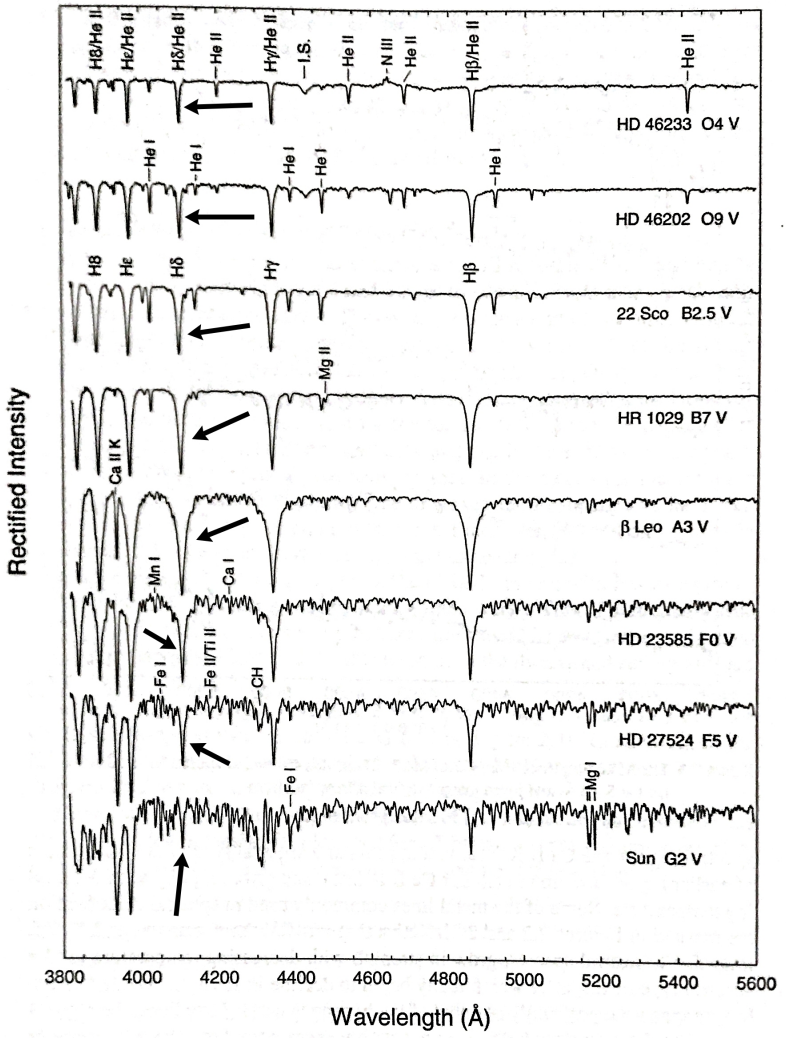


FIGURE 32: Sample of continuum normalized spectrums from O - G type stars, [23]. The arrows point to the H_δ absorption line.

CHAPTER VII

AN ALTERNATIVE METHOD FOR REDSHIFT EXTRACTION FOR FUTURE WORK

A new approach to extracting the redshift using the results from a Machine Learning model has been developed and is presented in this chapter.

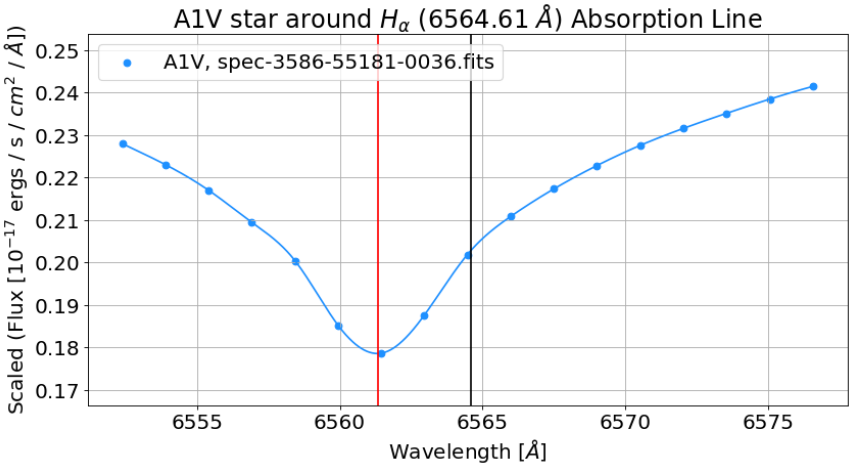
To extract redshift from stellar spectra, absorption lines at redshift wavelengths have to be identified, compared to their rest wavelength counterparts, and compared against each other. Then the most frequent redshift numbers are extracted and averaged to produce the redshift. During this process, these absorption lines are then used to identify the Harvard and MK class. The other approach is to compare the spectra to templates, which as a by-product produces the Harvard and MK class.

It is redundant to extract the redshift after the automatic classification of stellar spectra using the standard approaches or to automatically classify stellar spectra that have been redshift corrected because in both cases, classification is done twice. However, extracting the redshift is still an important step in analyzing stellar spectra and can provide information about a star. Any automatic classification scheme for classifying stellar spectra that does not address how to extract the redshift is an incomplete process.

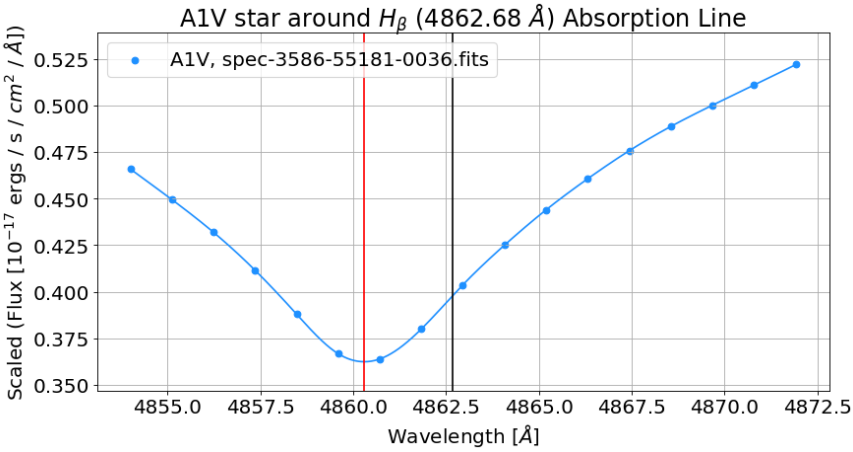
Approach to Redshift Extraction and Results

Instead of brute force trying to identify the absorption lines found within a spectrum by search through the entire spectrum for candidate absorption lines, the fact that the Harvard class is known (Machine Learning model prediction) is an important tool. As stated in Chapter III: Stellar Classification Types, the Harvard classes are determined from the absorption lines within the spectrum. Which means that for any sample, the

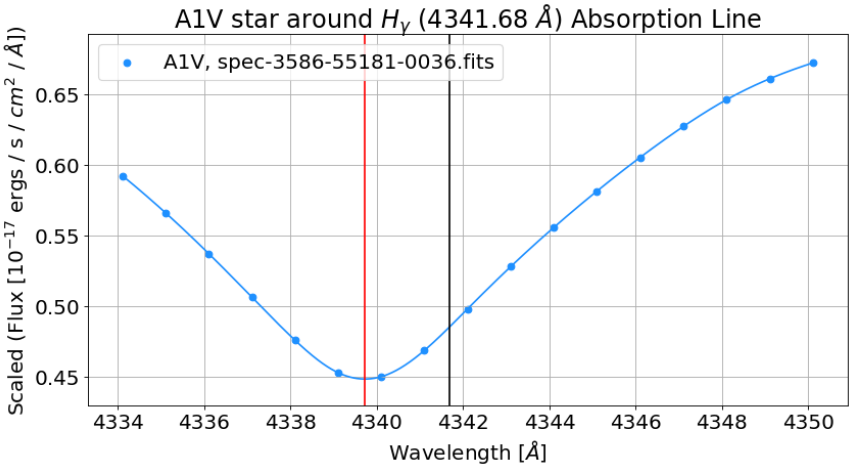
rest wavelengths of the absorption lines are immediately known due to knowing the Harvard class. Knowing the rest wavelengths of the absorption lines is useful because their redshift counterparts that are actually in the sample must be nearby. This is apparent in Fig. 33, where the black line represents the rest wavelength absorption line and the red line represents the same absorption line at its redshift wavelength. The redshift is computed using Algorithm 2. For the example found in Fig. 33, the actual redshift is -0.000489279$\pm$5.97$\times$ 10^{-5} and the extracted redshift is -0.000523273, which is within the bounds. The more absorption lines used, the better the result. This is because there are more redshift values to average.



(a) Redshift = -0.00047929



(b) Redshift = -0.00054568



(c) Redshift = -0.00054622

FIGURE 33: Redshift Extraction.

Algorithm 2 Redshift Extraction

```
function REDSHIFT_EXTRACTION(flux_Arr, wavelength_Arr, absorption_Line)
    bounds = 8 // range of flux measurements before and after the absorption line
    index, value = Find_Nearest(wavelength_Arr, absorption_Line)
    f = interpolate(x = wavelength_Arr[index - bounds — index + bounds],
                    y = flux_Arr[index - bounds — index + bounds])
    xp = linspace(min = wavelength_Arr[index - bounds],
                  max = wavelength_Arr[index + bounds], steps = 1,000)
    idx = argmin(f(xp))
    absorption_line = xp(idx)
    return absorption_line
end function
function FIND_NEAREST(array, value)
    min = ∞
    index = −1
    for 0 < i < array_Length do
        if array[i] − value < min then
            index = i
        end if
    end for
    return index, array[index]
end function
procedure MAIN
    absorption_Line_Arr ← Array of rest wavelength absorption lines in sample
    Z_Arr ← Array of 0 that is the same length as absorption_Line_Arr
    N = absorption_Line_Arr_Length
    for 0 < i < N do
        Z_Arr[i] = redshift_Extraction(flux_Arr, wavelength_Arr, absorption_Line_Arr[i])
    end for
    Z_Extracted = Sum(Z_Arr) ÷ N
end procedure
```

Discussion

Table 16 show a sample of the results for Harvard B, A, F, G, K, and M stars. It is important to note that the SDSS computed the redshift by comparing to templates [2] and minimizing a least squares problem. Therefore this thesis makes the assumption that if the extracted redshift is within double the SDSS error bounds to the redshift found

by the SDSS, then it is accurate. This is justifiable because the SDSS redshift is an approximation from the closest fit. However, samples with extremely small redshift seem to be approximated poorly with respect to the SDSS redshift. However, it is possible that the SDSS redshift was also approximated poorly.

TABLE 16: Sample of redshift extraction results for A type stars.

	Redshift at Wavelength: 6,564.61 $\mathring{A}$	Redshift at Wavelength: 4,862.68 $\mathring{A}$	Redshift at Wavelength: 4,341.68 $\mathring{A}$	Redshift at Wavelength: 4,102.89 $\mathring{A}$	Redshift at Wavelength: 3,971.2 $\mathring{A}$	Average Redshift	Standard Deviation	SDSS Redshift
A1V	-0.000465	-0.000539	-0.00539	-0.000465	-0.000465	-0.0004948	3.644×10^{-5}	-0.000489279$\pm$ 5.97×10^{-5}
A4V	-0.000762	-0.000874	-0.000911	-0.000799	-0.000911	-0.000851	6.041×10^{-5}	-0.000820341$\pm$ 1.502×10^{-4}
A2II	-0.000613	-0.000651	-0.000725	-0.000651	-0.000688	-0.000665	3.792×10^{-5}	-0.0005807$\pm$ 7.31×10^{-6}

Notice in Table 16 that the standard deviation for all three samples is small. This can be used as a basic confidence metric. Overall, each Harvard major class has different absorption lines present; therefore, if the Machine Learning model predicts the spectrum to be an A type star and the Standard Deviation is small, then the prediction is most probably correct. However, if the model predicts the spectrum to be an A type star and the Standard Deviation is large, then the prediction is most probably wrong. This would be the result of the software computing redshift in the spectrum where absorption lines do not exist.

Even though this approach relies on the assumption that the model's prediction is correct, there is a confidence metric produced. As a result, this approach has significantly fewer steps to produce the redshift. This creates an alternative approach to finding the redshift of a spectrum and makes using Machine Learning to automatically classify stellar spectra a realistic and viable approach.

In future work, this approach will be used to provide a classification confidence metric and as a step in a stellar spectra analysis pipeline.

CHAPTER VIII

CONCLUSIONS

Accurate stellar classifications can be obtained using a combination of domain-specific data pre-processing, feature selection and classification techniques. Compared to the previous work of other authors, there are two interesting conclusions:

1. The entire spectrum is not necessary to obtain this high level of accuracy.

2. Aside from wavelength fitting and flux scaling, any additional spectrum pre-processing after the processes presented by Dawson *et al.* [37] and Stoughton *et al.* [38] is unnecessary.

Therefore, when a star can be accurately classified prior to redshift correction, many stellar properties can be easily attained without the complex data transformations and statistical analyses used by other authors.

Even though redshift results in an undesired feature matrix because of flux discrepancies, high accuracy was still achieved. This can be accounted for by redshift values for stars being small and use of a large sample set of redshift values to train the classifier. However, the findings for the approach from Chapter V are not as important as the findings from Chapter VI and Chapter VII because Chapter VI classifies into both the Harvard and MK classes and Chapter VII and extracts the redshift.

The results from Chapter VI: Single Classification into both Harvard and
MK Classification Schemes and Chapter VII: An Alternative Method for Redshift Extraction
for Future Work support that accurate automatic stellar classification can be obtained using

domain-specific feature selection and the redshift can be easily extracted. Compared to the previous work of other authors, there are five interesting conclusions:

1. A high level of accuracy (99.54%) can be obtained by considering only flux measurements at wavelengths near the H_δ and Ca I absorption lines.

2. Harvard and MK classes can be identified with a high level of accuracy.

3. Aside from flux scaling, any additional spectrum pre-processing after the processes presented by Dawson *et al.* [37] and Stoughton *et al.* [38] is unnecessary.

4. The redshift can be easily extracted using domain knowledge.

Therefore, when a star can be accurately classified into both classification schemes, most stellar properties can be easily attained. When paired with the approach proposed for redshift extraction, the new approaches presented in this thesis for the automatic classification of stellar spectra are feasible, useful, and accurate.

REFERENCES CITED

[1] W. W. Morgan, P. C. Keenan, and E. Kellman, *An atlas of stellar spectra, with an outline of spectral classification*. Chicago, III., The University of Chicago Press, 1943.

[2] A. S. Bolton *et al.*, "Spectral classification and redshift measurement for the SDSS-III baryon oscillation spectroscopic survey," *The Astronomical Journal*, vol. 144, no. 144, pp. 1–20, 2012.

[3] Sloan Digital Sky Survey, "Sloan Digital Sky Survey." `https://www.sdss.org/`.

[4] F. Xing and P. Guo, "Classification of stellar spectral data using svm," in *Advances in Neural Networks (ISNN 2004)* (F.-L. Yin, J. Wang, and C. Guo, eds.), (Berlin, Heidelberg), pp. 616–621, Springer Berlin Heidelberg, 2004.

[5] J. N. Zhang, A. L. Luo, and L. P. Tu, "A stratified approach for automatic stellar spectra classification," in *2008 International Congress on Image and Signal Processing (CISP 2008)*, pp. 249–252, 2008.

[6] Z. Yi and J. Pan, "Application of random forest to stellar spectra classification," *978-1-4244-6516-3/10/26.00 2010 IEEE*, pp. 3129–3132, 2010.

[7] A. J. Pickles, "A Stellar Spectral Flux Library: 1150-25000 Å," *pasp*, vol. 110, pp. 863–878, July 1998.

[8] G. H. Jacoby, D. A. Hunter, and C. A. Christian, "A library of stellar spectra," *apjs*, vol. 56, pp. 257–281, October 1984.

[9] M. Bazarghan and R. Gupta, "Automated classification of sloan digital sky survey (SDSS) stellar spectra using artificial neural networks," *Astrophysics and Space Science*, vol. 315, p. 201, May 2008.

[10] D. G. York *et al.*, "The Sloan Digital Sky Survey: Technical Summary," *The Astronimical Journal*, vol. 120, pp. 1579–1587, 2000.

[11] "Redshifts." `https://skyserver.sdss.org/dr12/en/proj/advanced/hubble/redshifts.aspx`.

[12] "Absorption and Emission Lines." `http://skyserver.sdss.org/dr14/en/proj/basic/spectraltypes/absorption.aspx`.

[13] "Redshifts, classifications and velocity dispersions." `https://www.sdss.org/dr12/algorithms/redshifts/`.

[14] M. Brice and R. Andonie, "Classification of Stars using Stellar Spectra collected by the Sloan Digital Sky Survey," in *Proceedings of the International Joint Conference on Neural Networks*, IEEE, July 2019.

[15] D. R. Silva and M. E. Cornell, "A new library of stellar optical spectra," *Astrophysical Journal Supplement Series*, vol. 81, pp. 865–881, Aug. 1992.

[16] W. B. Weaver and A. V. Torres-Dodgen, "Accurate two-dimensional classification of stellar spectra with artificial neural networks," *The Astrophysical Journal*, vol. 487, no. 2, p. 847, 1997.

[17] C. A. L. Bailer-Jones, M. Irwin, and T. von Hippel, "Automated classification of stellar spectra - II. Two-dimensional classification with neural networks and principal components analysis," *Monthly Notices of the Royal Astronomical Society*, vol. 298, pp. 361–377, aug 1998.

[18] N. Houk and M. Smith-Moore, *Michigan Catalogue of Two-dimensional Spectral Types for the HD Stars. Volume 4*. 1988.

[19] F. Schierscher and E. Paunzen, "An artificial neural network approach to classify sdss stellar spectra," *Astronomische Nachrichten*, vol. 332, no. 6, pp. 597–601, 2011.

[20] K. N. Abazajian, J. K. Adelman-McCarthy, M. A. Agüeros, S. S. Allam, C. Allende Prieto, D. An, K. S. J. Anderson, S. F. Anderson, J. Annis, N. A. Bahcall, and et al., "The Seventh Data Release of the Sloan Digital Sky Survey," *Astrophysical Journal Supplement Series*, vol. 182, pp. 543–558, June 2009.

[21] R. O. Gray and C. J. Corbally, "An expert computer program for classifying stars on the mk spectral classification system," *The Astronomical Journal*, vol. 147, no. 4, p. 80, 2014.

[22] J. S. Almeida and C. A. Prieto, "Automated unsupervised classification of the Sloan Digital Sky Survey Stellar Spectra using k-Means clustering," *The Astrophysical Journal*, vol. 763, no. 1, p. 50, 2013.

[23] R. O. Gray and C. Corbally, J., *Stellar Spectral Classification*. Princeton University Press, 2009.

[24] S. Giridhar, "Advances in spectral classification," *Bulletin of the Astronomical Society of India*, vol. 38, pp. 1–33, Mar. 2010.

[25] F. R. Chromey, *To Measure The Sky: An Introduction to Observational Astronomy*. Cambridge University Press, 2010.

[26] D. J. Griffiths, *Introduction to Quantum Mechanics*. Pearson Prentice Hall, 2nd ed., 2005.

[27] D. J. Griffiths, *Introduction to Electrodynamics*. Pearson Education, 4th ed., 2014.

[28] B. W. Carroll and D. A. Ostlie, *An Introduction to Modern Astrophysics*. Cambridge University Press, 2nd ed., 2017.

[29] University of Iowa Department of Physics and Astronomy, "Imaging the universe." http://astro.physics.uiowa.edu/ITU/labs/foundational-labs/exploring-hertzsprung-russe/part-1-the-hr-diagram.html.

[30] "The Hertzsprung-Russell Diagram." http://skyserver.sdss.org/dr12/en/proj/advanced/hr/hrhome.aspx.

[31] Samihahplanet, "One shift, two shift, redshift, blueshift." https://samihahplanet.wordpress.com/2016/02/17/one-shift-two-shift-redshift-blueshift/. Redshift Image Reference.

[32] S. Marsland, *Machine Learning: An Algorithmic Perspective*. CRC Press, 2nd ed., 2015.

[33] Z. Ivezic, A. Connolly, J. VanderPlas, and A. Gray, *Statistics, Data Mining, and Machine Learning in Astronomy: A Practice Python Guide for the Analysis of Survey Data*. Princeton University Press, 2014.

[34] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, *Feature Selection for High-Dimensional Data*. Springer International Publishing, 2015.

[35] H. Zheng and Y. Zhang, "Feature selection for high-dimensional data in astronomy," *Advances in Space Research*, vol. 41, pp. 1960–1964, 09 2007.

[36] J. Li, K. Cheng, S. Wang, F. Morstatter, T. Robert, J. Tang, and H. Liu, "Feature selection: A data perspective," *arXiv:1601.07996*, 2016.

[37] K. S. Dawson *et al.*, "The baryon oscillation spectroscopic survey of SDSS-III," *The Astronomical Journal*, vol. 145, no. 10, pp. 1–41, 2013.

[38] C. Stoughton *et al.*, "Sloan digital sky survey: Early data release," *The Astronomical Journal*, vol. 123, pp. 485–548, 2002.

[39] Sloan Digital Sky Survey, "BOSS spectrograph." https://www.sdss.org/instruments/boss_spectrograph/.

[40] S. A. Smee *et al.*, "The multi-object, fiber-fed spectrographs for SDSS and the baryon oscillation spectroscopic survey," *The Astronomical Journal*, vol. 146, no. 32, pp. 1–40, 2013.

[41] N. Japkowicz, "Learning from imbalanced data sets: A comparison of various strategies," in *Papers from the AAAI Workshop Technical Report WS-00-05*, pp. 10–15, AAAI Press, 2000.

[42] Sloan Digital Sky Survey, "Understanding the optical data." https://www.sdss.org/dr14/spectro/spectro_basics/.

[43] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[45] T. E. Oliphant, *A guide to NumPy*. Published by Continuum Press, a division of Continuum Analytics, Inc., 2015.

[46] Numpy, "Memmap." https://docs.scipy.org/doc/numpy-1.14.0/reference/generated/numpy.memmap.html.

[47] "Turing: The cwu supercomputer." http://www.cwu.edu/faculty/turing-cwu-supercomputer.

[48] C. Jaschek and M. Jaschek, *The Classification of Stars*. July 1990.

[49] "Multiprocessing - process-based parallelism." https://docs.python.org/3.7/library/multiprocessing.html#module-multiprocessing.