DETECTING POTENTIAL LENSED GALAXIES BEHIND FOREGROUND GALAXY TARGETS USING MACHINE LEARNING TECHNIQUES

by

Zahra Fahimfar

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

In

Data Management and Analysis

School of Computing

The University of Utah

May 2018

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of **Zahra Fahimfar**

has been approved by the following supervisory committee members:

Jeffrey Phillips	, Chair	**date approved** (Date Approved)
Bei Wang Phillips	, Member	**date approved** (Date Approved)
Vivek Sirkumar	, Member	**date approved** (Date Approved)

and by **Ross Whitaker**, Chair/Dean of

the Department/College/School of **School of Computing**

and by David B. Kieda, Dean of The Graduate School.

The University of Utah Graduate School

ABSTRACT

Detecting the background galaxies within the spectrum of the foreground galaxy is one of the most effective ways to identify strong lensing phenomenon. However, it is very hard and time consuming for astronomers to apply this search method manually (i.e., one by one) to huge cosmological data sets. This study attempts to predict the background galaxies and discover the potential lensed candidates by using classification methods. To achieve this, the most important step is to leverage cosmological data by extracting potentially useful features for the classification methods.

In this study, after extracting the potentially useful features from two different astronomy datasets, chi square weighting feature selection was applied on them to find the final set of the useful features. Then, various state-of-art classification methods were applied on the datasets to predict lens candidates. Classifier performance was measured in terms of accuracy, AUC, and F-measure. The results showed that 85 features chosen by chi square weighting are the most useful features. Logistic Regression outperformed all other classification methods for the prediction task. Finally, the prediction method using classifiers is significantly more efficient than manual inspection. The proposed method in this study is generalizable for detecting background galaxy and potential lenses in any cosmological data. This can significantly improve the efficiency for astronomers to apply their search methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Mass warps the space around it, of which gravitational lensing can be detected when light is bent in this warped space (i.e. lens) between the source and the viewer. Typically, the light is only bent a tiny bit such as 1/3600 of 1 degree. The path of the light from a source, for instance galaxy, can be bent significantly when it passes near a heavy mass, such as another galaxy. If the initial light path is near enough to a massive enough object(s), multiple images of the source can be bent towards the viewer, which is called strong gravitational lensing. In strong gravitational lensing, the lens (also known as the foreground object or deflector) produces either several stretched images of the source (also known as the background object) into the shape of an arc, or stretches the source into a ring around the lens. Since the background galaxy maintains its brightness, more light can be collected from the larger and magnified image(s) (1). From the lensing geometry, astronomers can compute the total mass enclosed within the strong lensing regime. This provides astronomers a powerful probe into detecting the contribution of dark matter within the enclosed lensing radius (i.e. the strong lensing regime).

An example of lensed features can be simulated by looking through the bottom of a wine glass at a lit candle (i.e. similar to the lens), and observe several arc-like images of the flame (i.e. similar to the source images observed). Most lens candidates have been

found by detecting gas emission lines from the source galaxy within the spectra of the lens galaxy. Since the source is farther away (i.e. higher cosmological redshift), its gas emission lines are observed at a redder bias than the spectra of the lens galaxy. Follow-up high-resolution imaging with the lens subtracted, can reveal the lensed features of the source, and thus confirm these candidates (2).

To stumble on a gravitational lens by observing just a photo is extremely rare, and many source galaxies can be faint relative to the flooding light of the lens galaxy, which can wash any sight of the source away. Many lensed features can only be seen after subtracting the bright lens galaxy from the image. Thus, the lens galaxy has to be modeled to extreme precision to prevent over/under subtraction features from affecting the quality of the observed lensed features. Lens light removal is a crucial step in gravitational lens modeling when the emission from the lens is high enough to hinder the correct interpretation of the lensed emission (3).

Although discovering the background galaxies within the spectrum of a foreground galaxy is very hard and time consuming, it is the most effective way to detect the faint background galaxy (4). Out of 250 lenses that have been discovered and examined by photo, 150 come from spectroscopic discovery using the SDSS data alone (5).

Many high yield surveys are using computational search methods to find potential background galaxies in huge datasets. Often this results in a large set of potential background galaxy spectra to manually inspect. For example, in the Spectroscopic Identification of Lensing Objects (SILO) survey, 1.5 million spectra were computationally scanned for high S/N emission lines from the background galaxy. They

report manually inspecting at least 11,421 spectra with good indications of emission lines from ~700 background galaxies(6). However, by using their domain knowledge, extracting related features and applying data mining methods the whole process can be automated with even a higher accuracy and time saving efficiency.

To the best of our knowledge, there has been no work on applying data mining methods on predicting and identifying the background galaxy. Therefore, in this paper we aim at predicting background galaxies by using state-of-art machine learning methods. This prediction could directly or indirectly result in finding potential lens candidates.

To achieve this, we use two dataset including the Extended Baryon Oscillation Spectroscopic Survey (eBOSS) (7), and the Mapping Nearby Galaxies at APO (MaNGA) (8) which are projects of the Sloan Digital Sky Survey (SDSS).

For eBOSS detections, detection of a background galaxy typically results in finding a lens by default since each fiber is focused on a distant galaxy, with a coverage as wide (about 2 arc seconds) as the typical strong lensing regime (i.e. you spot a background galaxy in eBOSS, it is likely strongly lensed). However, the spectra of MaNGA galaxies are recorded from bundles of fibers. Each fiber covers a 1 arc second radius, and is distributed within a field of view up to about 14 arc seconds in radius. However the strong lensing features are located within the first few arc seconds of the galaxy center. As a result, a detected background galaxy in MaNGA does not yet assure it is being strongly lensed until it can be shown that it is either near enough to the strong lensing regime, or there are multiple images observed from the source.

Therefore, for eBOSS, astronomers can use the machine learning method to isolate and increase assurance of potential gas emission lines of the background galaxy,

and then inspect them, of which good signals of background galaxies become strong lensing candidates. For MaNGA, they can also use the machine learning methods to isolate potential gas emission lines of the background galaxy, and inspect them. However, they need to compare the good signals of the background galaxies to see how close they are to the strong lensing regime derived from prior foreground galaxy information.

At the end, the results show that machine learning methods can make the prediction with a high accuracy of 94.66 and AUC of 97.50. The outcome of this paper can be used by astronomy researchers to facilitate their manual inspection and detection of background galaxies and strong gravitational lensing.

CHAPTER 2

BACKGROUND

2.1 Astronomy Background

Gravitational lensing occurs when a distribution of mass (i.e. Mass from one foreground star, one foreground galaxy, or multiple foreground galaxies in a cluster warping space) is between a distant light source and the observer that is capable of bending the light from the source onto a path that reaches the observer. This phenomenon is known as gravitational lensing, and the amount of bending is one of the predictions of Albert Einstein's general theory of relativity (9).

Normal lenses such as the ones in a magnifying glass work by bending light rays that pass through them in a process known as refraction, in order to focus the light somewhere such as in your eye. Strong galaxy-galaxy scale gravitational lensing happens when we have two galaxies aligned just right (i.e. about only arc seconds apart) on the sky, in which both their relative distances, and the mass of the foreground galaxy plays a huge role in the creating strong gravitational features. When detected, Astronomers often look at them with the Hubble space telescope (10). Consider that we have a massive elliptical galaxy and right behind it in a far distance there is a little galaxy. If the alignment is just right, we can have the situation where the light from the background object can bend around and refocus somewhere else and we can see multiple images or

distorted rings from the telescope. More massive foreground galaxies have a stronger gravitational lens and as a result will bend the passing light rays at a greater angle towards the lens.

There are three types of gravitational lensing including strong lensing, weak lensing and microlensing (11). In this paper, we focus on the first type of gravitational lensing. Strong lensing happens where there are easily visible distortions such as the formation of Einstein rings, arcs, and multiple images (12). It means that the strength of the gravitational potential is sufficient that an image passing on the opposite side of the foreground galaxy is bent enough to be seen as a counter image or contributes to the ring. Depending on the alignment of the observer on Earth with a distant background object such as a galaxy and a massive foreground object, which is often a galaxy or cluster of galaxies, all sorts of distorted images can be observed: rings, arcs, or even multiple images of the same background object. Figure 2.1 shows that how gravitational lensing works.

Strong gravitational lensing offers lots of research into the astrophysical distribution of dark matter such as measurements of foreground galaxy surface mass densities form lens models of multiple images (13). Strong lensing can also allow us to calculate the mass of the galaxy clusters which can give us intuition into the construction history of these massive galaxy clusters. This can also help to find objects far beyond the resolution or detection ability of earth and space telescopes, revealing more redshift samples about the expansion history of the universe.

Telescopes and instruments can only see details on objects up to a certain distance due to resolution limits (for example, not even the Hubble Space Telescope can observe

the NASA landing sites on the moon due to the resolution limit, a problem that rises due to the wave property of light, the diameter of the telescope, and the wavelength of the light).

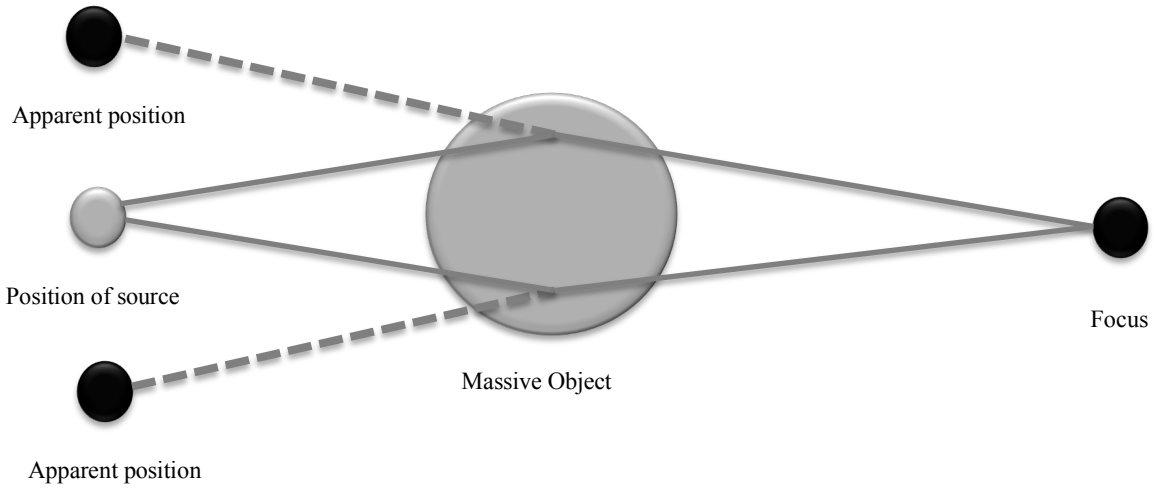


Figure 2.1 Gravity from a foreground object bends light from a more distant object

Also, the object might be too faint to see without gravitational lensing, and would require an excessive amount of exposure time to begin to see them. Strong gravitational lensing both stretches the background galaxy image, and thus effectively magnifies it, so we can see more features. Since the surface brightness density of the object stays the same, the amount of flux per magnified image is increased.

The most productive resource of detecting strong galaxy-galaxy lens candidates is spectroscopic discovery from different survey methods. This method provides evidence for background galaxy behind foreground galaxy along with accurate measurements of the lens and source red shift (14). Then high resolution images can confirm the lensing features and make precise measurements of the angular distance between the background

galaxy images.

It is important to note that the redshift is caused by the expansion of space on a cosmological scale. This cosmological expansion rate is known as 'Hubble's constant' that correlates to about 73.8 km/sec/Mpc (i.e. for every distance of 1 mega parsec from us, objects in any sky direction is moving away from us at 73.8kilometers per second (15). This means the wave pattern of the light reaching us is stretched out, similar to hearing a firetruck rush past you. Thus a 'cosmological distance' results in a redshift that makes it easier to see the background galaxy gas emission lines from the foreground galaxy gas emission lines, since the background galaxy is redshifted more than the foreground.

In this search method, the foreground galaxies work as a gravitational lens for any object behind it. Therefore, the spectra of the foreground galaxies should contain the emission features of background galaxies and so, such lensed objects can be discovered in the spectra of foreground galaxies. Spectroscopic discovery searches for these background galaxy gas emission features (4).

The Figure 2.2 shows the example plot of Spectroscopic discovery searches. The black line shows the observed emission lines plot as a flux and wavelengths. In fact it consists of both background and foreground galaxy emission lines and there is no way to completely separate the emission lines from background and foreground galaxies. The blue line shows the model fitted to the continuum of the foreground galaxy. For each spectra, astronomers constructed a best fit model to the galaxy continuum using a basis of 7 principle component analysis (PCA) eigenspectra. The red line known as Resflux shows the subtraction of two previous plots and contains the background emission lines. (16)

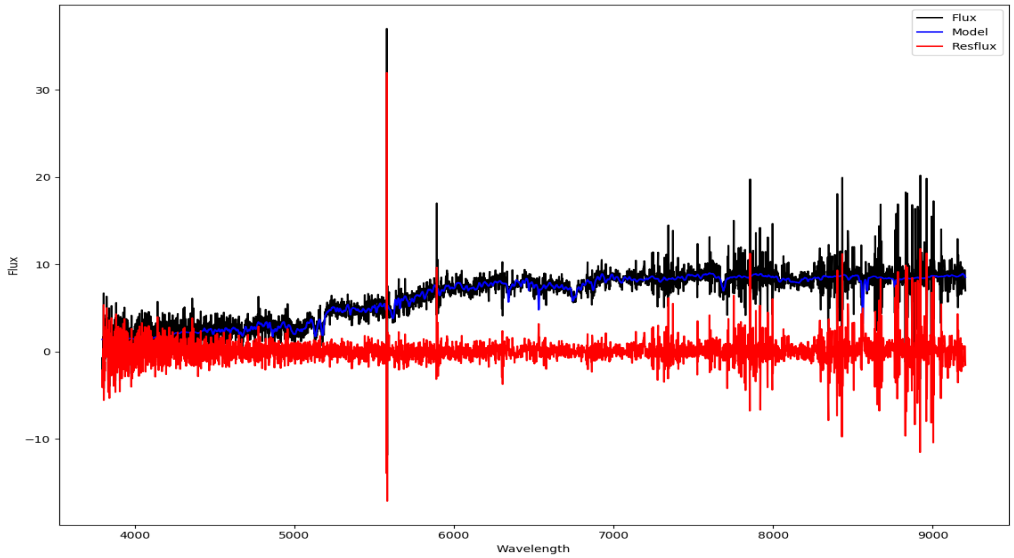


Figure 2.2 Observed emission line, best-fit model and background emission line plots

There are two different search methods for the emission lines. Oneline searches for the two OII emission lines that are close enough together that the OII doublet looks like a double peaked spike in the data. We search for potential OII doublets with Signal-to-Noise > 6. Multiline searches for 2 or more emission lines from a set of ten known emission line types with (signal-to-noise > 4). We include both search methods in the data set to help us predict the lenses.

The Figure 2.3 shows example plots of typical multiline and oneline detections in the ideal case when we do not have the noisy dataset. Right plot shows oneline search and left one shows the multiline search. The black solid-line shows the observed emission line, the blue dashed-line shows the model fitted to the continuum of the foreground galaxy, and the red vertical dashed-line shows the wavelength of the discovered background emission lines. The green dashed-dotted line shows the Gaussian fitted to the background emission lines.

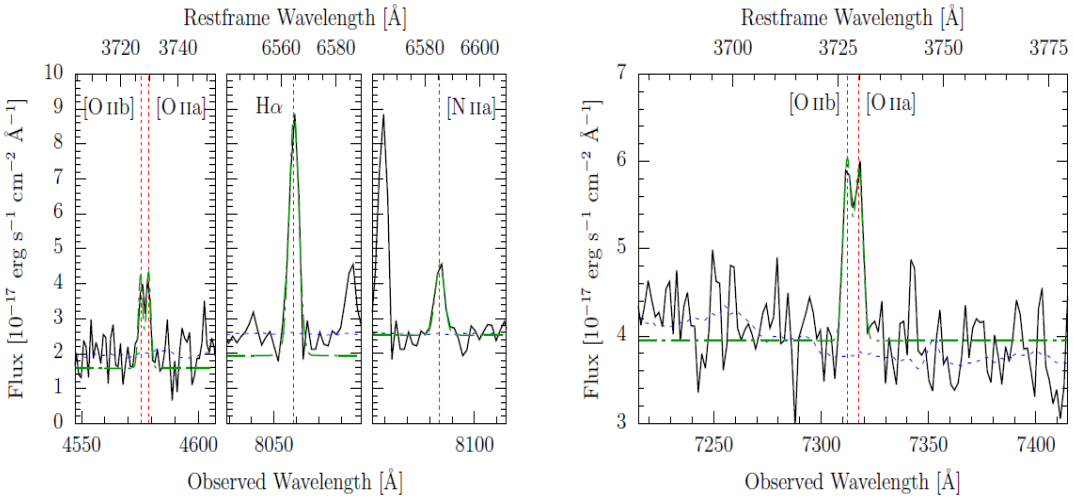


Figure 2.3 Example plots of typical multiline and oneline detections

When we have a noisy data set, it is very hard to inspect the emission line and perform the search methods. Figure 2.4 shows the example of oneline search when we found the hits. In order to perform the search method, we need to calculate the wavelength index of the desired emission line. The black rectangular shows the index for OII emission line. Figure 2.5 shows the picture of the OII emission line when we zoom into that specific index. As you can see in Figure 2.5, all requirements of the oneline search were satisfied and so, we can conclude that it is a hit.

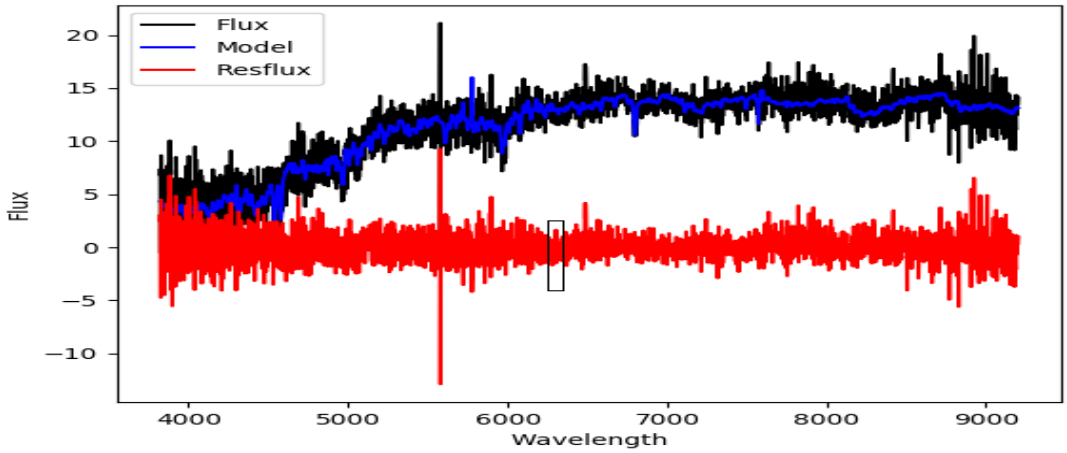


Figure 2.4 Example plot of oneline search when hits is found

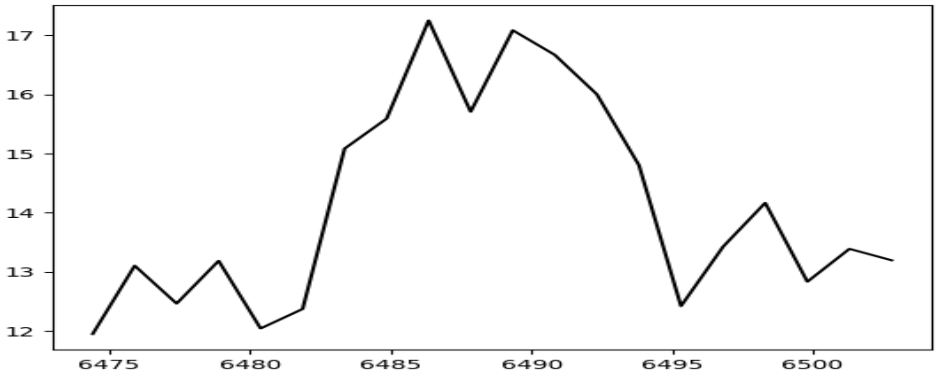


Figure 2.5 Zoomed picture of OII double emission when hit is found

Figure 2.6 shows the example of oneline search when we did not find the hits. We again calculate the wavelength index of the desired emission line for this example. The black rectangular shows the index for OII emission line and Figure 2.6 shows the picture of the OII emission line when we zoom into that specific index. As you can see in Figure 2.7, the requirements of the oneline search were not satisfied and so, we can conclude that it is a not hit.

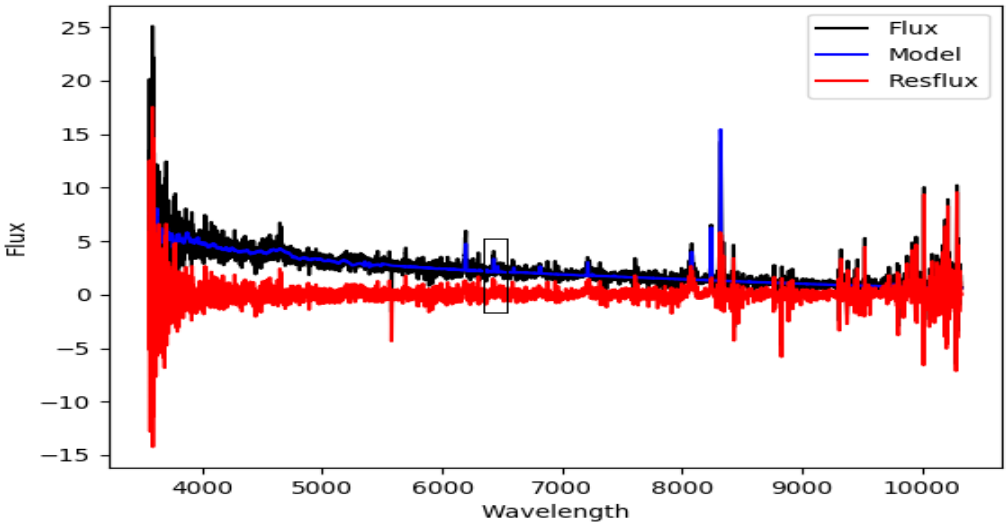


Figure 2.6 Example plot of oneline search when hits is not found

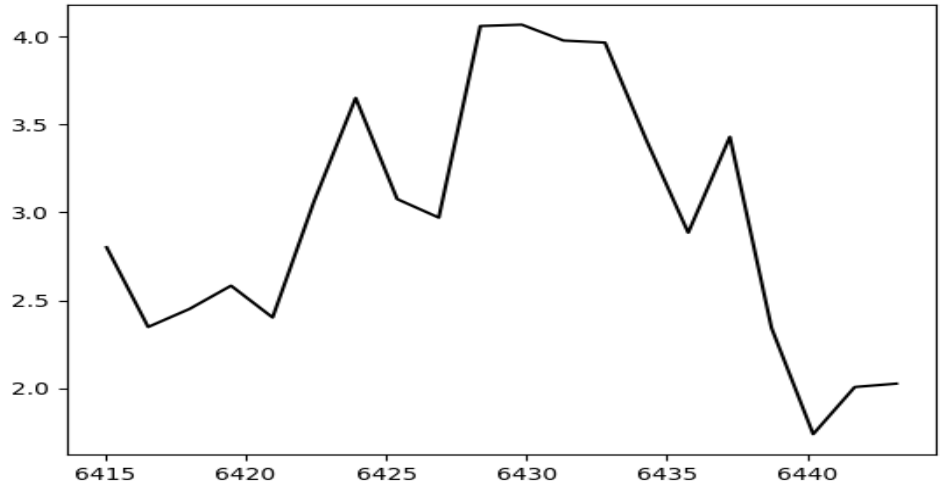


Figure 2.7 Zoomed picture of OII double emission when hit is found

2.2 Computer Science Background

2.2.1 Data mining

Over the past decade computational power of computer has significantly increased. Moreover, large amount of observed data have been recorded in datasets (17). As a result, extracting useful and valuable information from such datasets is becoming essential in a variety of areas such as astronomy with thousands of data record per day (18). To achieve this, data mining aims at discovering knowledge and finding important information, patterns and trends from data. More specifically, data mining analyzes large data sets in order to extract valuable information using methods in different fields such statistics, databases and data science (19).

Although, data mining experts are focused on the technical aspects of problem, they need to know the domain knowledge associated with that in order to better

understand the problem and propose a solution. Therefore, data mining of a problem (e.g., astronomy) needs a close collaboration between data mining experts and the scientists of the related field.

Data mining consists of several different tasks such as classification, clustering, association mining and etc. However, in this study we are focused on one the classification task because of the problem (20). Classification methods attempt to assign a generalized known structure (e.g., labels) to a new data. As an example, classification methods classify an e-mail as "spam" or "legitimate" according the similar previous email with known labels (21).

Classification methods get a set of features, as input to predict a feature as output for data instances (17). There are different names for these inputs and output in the literature. Input variables are also called independent features or predictors and output feature is also called dependent variable or target variable. The task of a classification method is to build a model on a specific partition of the input dataset (consisting of data instances) and apply that on the other partition to label (i.e., classify or predict) its data instances. The building model partition is called training dataset and the evaluation partition is called testing dataset (18). Some classification methods are designed for binary classification tasks where the dependent variable is binary or dichotomous and some method can handle categorical dependent variables as well (22). Binary classification means that there are only two values for output, "0" and "1", showing the target outcomes (e.g., pass or fail, alive or dead). Categorical (or nominal) classification means there are more than two values for dependent variables (23).

The internal process of various classification methods to build the data mining model is different. The rest of this section gives an overview of the internal process of the widely used classification methods.

2.2.2 Decision Tree

Using statistical analysis on the relationship between each input variable and the target variable decision tress predict the target variable (24). Decision tree is a popular classification method which can be explained as a combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data (25).

In the tree structures, at the first layer (i.e., top of the tree) there is a root node which contains all of the input variables describing data instances in the training set. In the next layers (branches) this tree is split into child nodes using the criterion that minimizes the classification error. This process repeats iteratively and stops when specific user defined criteria are reached. At the last layer class labels are represented by leaves of the tree and conjunction of the input features are represented by branches of the tree (25).

2.2.3 Logistic Regression

Logistic regression is one of the most popular methods for classification. It is most widely used where the dependent variable is binary (22). When there are more than two values for dependent variables multinomial logistic regression should be employed (23).

Binary logistic regression estimates the probability of a binary target variable based on the input variables. In other words, the goal of logistic regression is to find the best fitting model to describe the relationship between a binary target variable and a set of input variables (26). Logistic regression generates the coefficients and its standard errors and significance levels of a formula to predict a logit transformation of the probability of presence of the characteristic of interest (27).

2.2.4 K Nearest Neighbor

K Nearest Neighbor algorithm (k-NN) is one of simplest techniques to build a classification method. The basic idea is to classify an instance based on its similar neighbors (28). In other words, when there is an unlabeled data instances, the class label for that instance is determined by looking at the label of its neighbors. The underlying idea is that instances with similar input variables are most likely to belong to the same class and should be labeled with the same target label. Therefore, the classification of a instance is dependent on a target label of its neighboring instances (29).

Given a new sample, the method looks for the k instances in the training data that are the closest neighbor to this instance. Using voting over the labels of the k nearest neighbors the label of the new instance is assigned. As a result, a similarity measure is required to determine the closeness of different instances to the new instance. Variety of similarity measures such Manhattan Euclidean or Hamming distance function can be employed to fulfil the task (30).

2.2.5 Naïve Bayes

Naïve Bayes algorithm uses probabilistic approach for classification where the probabilities show the relationship between input variables and output variable (31). Given an input variable, the probability of each class is estimated and then the class with the high probability determined as a label of an unseen instance. This method is primarily based on applying Bayes' theorem (32) with independent assumption between input variables.

2.2.6 Artificial Neural Network

Artificial Neural Network (ANN) is a well-known classification method in various fields of study. ANN attempts to make computers model the brain and simulate the collection of neuron. This method comprised of a series of branching nodes that operate like the neuron in the body and then information is given to the nodes and transmits it across the entire complex. The network processes the information and generates the desire output (33).

ANN takes input features and maps them on to the output variable. When the network is trained, it can be used to label unseen test instances. It also uses an algorithm to minimize a cost function. (34)

2.2.7 Bayes Network

This method is a probabilistic graphical model that shows a set of input variables and their conditional dependencies via a directed acyclic graph. Bayes network tackle the problem of independency assumption of independence in Naïve Bayes method and improve the performance. Directed acyclic graph allow efficient representation of the join

probability distribution. Each vertex in the graph represents a random variable, and edges represent direct correlations between the variables (35).

Each input variable is independent of its non-descendants in the graph given the state of its parents. These independencies are then exploited to reduce the number of parameters needed to characterize a probability distribution, and to efficiently compute posterior probabilities given evidence. Probabilistic parameters are encoded in a set of tables, one for each input variable, in the form of local conditional distributions of a variable given its parents. Using the independence statements encoded in the network, the joint distribution is uniquely determined by these local conditional distributions. (36)

2.2.8 Support Vector Machine

Support Vector Machine (SVM) attempts to find the hyperplane that best splits two classes of data. This algorithm creates the decision boundary instead of creating a model of the data. The input data is considered as set of vectors and the data point (i.e., data instances) closes to the boundary are support vectors. Other than performing linear classification, SVM can achieve a non-linear classification using kernels. The input features are mapped into a higher dimensional space using a kernel in order to make the non-linear relationships in the data linear. (37)

2.2.9 Classification Evaluation

To evaluate classification methods various measures can be employed. This section briefly elaborates the measures used in this study as the most popular classification measures. More details about these measures can be found elsewhere (38).

Before starting with the classification performance measure, it is important to understand the confusion matrix. It is a table that is used to summarize and describe the performance of a classification model. Each column shows the instances in an actual class label and each row shows the instances in a predicted class label (or vice versa) (39).

Table 2.1 Confusion Matrix

	Actual +	Actual -
Predicted +	True Positive **(TP)**	False Positive **(FP)**
Predicted -	False Negative **(FN)**	True Negative **(TN)**

As shown in Table 2.1, confusion matrix consists of four values including True Positive (TP), True Negative (TN), False Positive (FP) and True Negative (TN). Here is the explanation of each value in the case that we predicted the presence of a disease. In this case there are two possible predicted classes: "yes" and "no".

- TP: The cases we predicted yes (they have the disease), and they do actually have the disease.
- TN: The cases we predicted no and they don't have the disease.
- FP: The cases we predicted yes, but they don't actually have the disease.
- FN: The cases we predicted no, but they actually have the disease.

Several standard performance measures have been defined from the confusion matrix. The most popular classification measures for binary classifiers are elaborated in the following.

- Accuracy: Accuracy is the number of instances predicted correctly divided

by total number of instances (in the test set).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FN}}$$

- Precision: Precision of a class label is the number of true positives (i.e. the number of instances correctly labeled as belonging to the positive class) divided by the total number of instances labeled as belonging to the positive class (i.e. the sum of true positives and false positives), which are instances incorrectly labeled as belonging to the class.

$$\text{Precision Yes} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Precision No} = \frac{\text{TN}}{\text{TN} + \text{FN}}$$

- Recall: Recall is the number of true positives divided by the total number of instances that actually belong to the positive class (i.e. the sum of true positives and false negatives), which are the instances which were not labeled as belonging to the positive class but should have been.

$$\text{Recall Yes} = \frac{\text{TP}}{\text{TP} + \text{TN}}$$

$$\text{Recall No} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

- F-measure: This is a weighted average of recall and precision where it reaches its best value at 1 and worst at 0.

$$\text{F} - \text{measure Yes} = 2 \times \frac{\text{precision yes} \times \text{recall yes}}{\text{Precision yes} + \text{recall yes}}$$

$$\text{F} - \text{measure No} = 2 \times \frac{\text{precision no} \times \text{recall no}}{\text{Precision no} + \text{recall no}}$$

- Area Under the Curve (AUC): AUC is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. AUC is calculated by finding the area under the curve of coordinate system of True Positive Rate (TPR) and False Positive Rate (FPR). Any binary classifier has a threshold for classifying an instance as "Yes" or "No". Changing the threshold, results in different values of FPR and TPR. Building a curve using these values, AUC is calculated as the area under that curve:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

CHAPTER 3

METHOD

As mentioned, this study attempts to predict and detect gravitational lens and background galaxy candidates. To achieve this, we apply the classification methods (described above) on the data of galaxies observed by Extended Baryon Oscillation Spectroscopic Survey (eBOSS) and the Mapping Nearby Galaxies at APO (MaNGA). To best of our knowledge, this is the first research to study the effect of data mining methods on prediction of lenses, which can be counted as the first contribution of this study.

The main obstacle in applying classification methods for the prediction of background galaxies is that the only available data from galaxies are the data gathered for astronomy purposes. This data manly consists of human manual inspection in the format of fits files which is not meaningful for classification methods. As a result, there is a need to find a decent set of features that can best describe the data and fed into the classification methods. This needs collaboration of both data mining and astronomy experts. As a result, the second contribution of this paper is to provide such features sets after long runs of collaborations with astronomy experts. These features can be used as a benchmark for the future data mining studies on the prediction of gravitational lens candidates and sequentially background galaxies.

3.1. Input Features Extraction

To find the best set of features from the dataset and extracting such features, we first tried to learn the manual inspection done by humans to examine the potential background galaxies in the astronomy field. Then, we selected different features from the manual inspection which are useful for classification methods. These features were extracted from the fits files and prepared for feeding into the classification methods. Table 3.1 shows the final set of extracted features:

Table 3.1 Input features

Feature	Description	Feature	Description
RedShift	Redshift describes how light shifts toward shorter or longer wavelengths as objects in space such as stars or galaxies move closer or farther away from us.	EMLINE	Required number of emission lines a signal-to-noise threshold in order to be recorded as a 'hit' (i.e. detection).
o2sn	These are the signal-to-noise of the background emission-lines identified by row aligned column to the right.	O IIB	These features are gas emission lines from the background galaxy caused by stars heating the gas nearby and causing it to glow at specific wavelengths characteristic to the elements atomic structure, abundance in the gas, and probability of emission.
emsn1		O IIA	
emsn2		Hδ	
emsn3		Hα	
emsn4		Hβ	
emsn5		O IIIB	
emsn6		O IIIA	
emsn7		N IIB	
emsn8		Hγ	
emsn9		N IIA	
emsn10		S IIB	
		S IIA	
HIT_PAR1	Gaussian fits wavelength position	HIT_PAR2	Initial model fitting base height
HIT_PAR3	Amplitude of gauss (i.e. how large it is)	HIT_PAR4	Sigma used (i.e. how wide is the gauss)
HIT_CHI2	Reduced Chi squared of the Gaussian model fit to the residual flux.	HIT_FWHM	Full Width at Half Maximum of the Gaussian model.
G_FAIL	The emission line feature can be too faint to fit a Gaussian model. Thus the header of the fits file specifies if a 1 or 0 means the model could or could not be fitted.		

3.2. Gaussian Model Fitting

To create the Gaussian model, python is used to fit either a single or double Gaussian model to the residual flux where detection is believed to be located. The residual flux is the flux – model continuum, which mostly leaves behind emission line flux or other false flux spikes. After the fit, the reduced chi square (i.e. goodness of fit) and the full width at half maximum (FWHM) is measured. Other information such as best fit model position, height, size and sigma (gauss width) is collected. This information can then be used to filter the more likely emission lines from the random flux spikes (for example, a double Gaussian model will fit a true OII doublet emission line feature better than a single Gaussian fit. The FWHM can be used to realize if the flux spike is skinnier than the doublet emission lines positions, which indicates this is a bad flux spike if the FWHM is skinnier than the emissions separation.

3.3. Parameter Tuning

To find the best classification algorithm, we compare different classification methods including Decision Tree, Logistic Regression, k-Nearest Neighbor, Bayesian Network, Naïve Bayes, Support Vector Machine and ANN. Optimizing each of these algorithms we tune different parameters for each algorithm. This parameter tuning includes depth of the tree, leaf size and confidence for Decision Tree, kernel type, kernel catch and maximum iteration for Logistic Regression, k (number of nearest neighbors) and measure type for k-Nearest Neighbor, learning rate and momentum for ANN, for Bayesian Network, minimum bandwidth and number of kernel for Naïve Bayes, kernel type, catch size, gamma and epsilon for Support Vector Machine. The best set of parameters for each algorithm is chosen for the final evaluation.

CHAPTER 4

EXPERIMENTS

4.1. Data

In this paper, we use two different datasets for our prediction task. The first one is from the Extended Baryon Oscillation Spectroscopic Survey (eBOSS) and the second one is from Mapping Nearby Galaxies at APO (MaNGA). Both data sets record the spectra from a galaxy and they have been collected from one or many fibers that transfer the light to a spectrograph. For example, imagine someone can point a transparent optical fiber towards a light source, and then connect a spectrograph on the other side of the fiber. The light will travel down the fiber and reach the spectrograph. The spectrograph then splits the light into a rainbow that is spread across a camera. The camera then records the intensity of light per part of the rainbow, which is the 'Flux' per wavelength (called spectra) that is in each fits file.

eBOSS places a single fiber on each galaxy to record overall the galaxy spectra. However, MaNGA uses an Integral-Field-Unit (which means they point many fibers bundled into a large cord) at one galaxy, and record many spectra all over the galaxy.

Therefore, the BOSS and SDSS-I galaxies were observed with one fiber yielding a single spectrum containing all of the light from the galaxy, each MaNGA galaxy was observed with a fiber bundle, in which each fiber yielded multiple spectra from multiple

exposures. This allows the candidate background emission-lines to be spatially correlated, increasing our confidence that the background emission-lines are real.

Then, the fiber spectra can be used to create a finely spaced grid of spectra over the galaxy. For example, 127 fibers could be used from all over the galaxy to create a 74 X 74 grid of special interpolated spectra.

eBoss maps the distribution of galaxies and quasars from when the Universe was 3 to 8 billion years old, a critical time when dark energy started to affect the expansion of the Universe. We use the sample of galaxies in eBoss to predict the strong lensing and background galaxies. This sample gives 2,670 plates where each of them contains several galaxies information. Out of all these galaxies, we only have 141 known galaxies labeled as either good hits or bad hits. Since, we have two distinct ways of measuring the hits (i.e. Oneline or Multiline), 282 records in total remained at the end.

MaNGA obtains spectra across the entire face of target galaxies using custom designed fiber bundles. Our sample of this dataset contains 192,650 fiber spectra including oneline and multiline search results. We have the target variable for 10,000 spectra and we know the hits type for them. After extracting the features from both oneline and multiline, there are 20,000 records in total for this sample.

For each spectra in both datasets, there are two corresponding .fits files. We need to process .fits files to filter out and extract the proper and necessary features. The first type of the fits file consist of the basic information of the potential background galaxy emission lines, the foreground galaxy spectra, and the foreground galaxy model of the spectra, while the second type include the information of the Gaussian model fits to the background emission lines.

After processing the data, we extract the variables shown in Table 4.1 form either oneline or multiline fits file for each spectra. Moreover, we extracted other features from the Gaussian model .fits files for each emission line of each galaxy shown in table 4.2. In total, we have 24 features from oneline or multiline fits file and 96 features from Gaussian model. More elaborations on the meaning of each feature in Table 4.1 and Table 4.2 can be found on Table 3.1.

As we mentioned above there are two types of fits files: oneline/multiline fits file and onelineGuess/multilineGuess fits file. The information of each record (i.e. galaxy spectra) in our data corresponds to two fits files. This is either multiline and multilineGuess or oneline and onelineGuess. The format of oneline is the same as multiline fits file. Similarly, the format of onelineGuess is the same as multilineGuess fits files. Here we show that how each format of fits file stores the information of the potential background emission lines and the corresponding foreground spectra they were detected in.

Figure 4.1 shows the info part of oneline/multiline fits files. It consists of the information of the data stores in this file. As seen in Figure 4.1, there are 10 types of different data in oneline/multiline fits files with different dimension, type and format. We did not need all these data to inspect the emission lines of the background galaxies and so, we just extracted those data that was useful to detect the target variable.

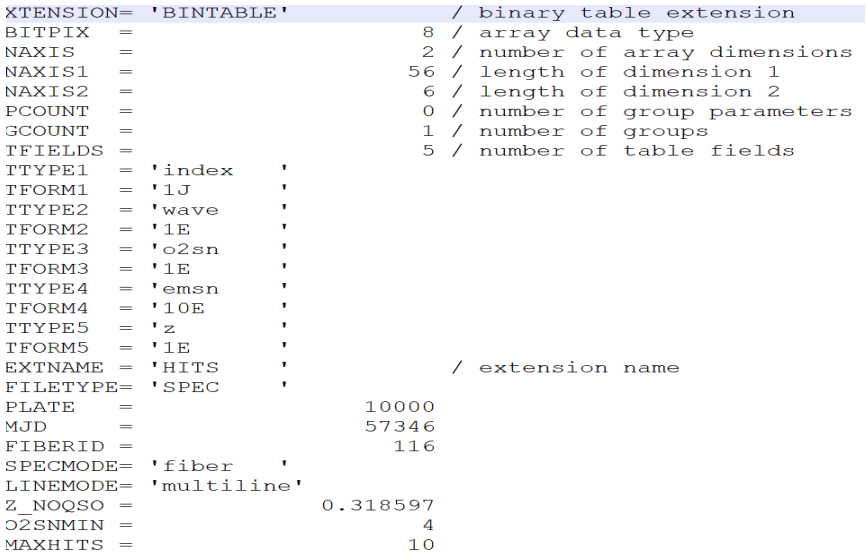
Figure 4.2 shows the info part of onelineGuess/multilineGuess fits file. There are several different types of data extracted from the Gaussian model filling and stored in these kinds of fits files. We also did not use all these information for our prediction and choose the necessary data.

No.	Name	Type	Cards	Dimensions	Format
0	PRIMARY	PrimaryHDU	37	()	
1	HITS	BinTableHDU	51	6R x 5C	[1J, 1E, 1E, 10E, 1E]
2	WAVE	ImageHDU	39	(4606,)	float32
3	FLUX	ImageHDU	39	(4606,)	float32
4	IVAR	ImageHDU	39	(4606,)	float32
5	IVAR_RESCALED	ImageHDU	39	(4606,)	float64
6	MODEL_GENERATED	ImageHDU	39	(4606,)	float64
7	RESFLUX	ImageHDU	39	(4606,)	float64
8	O2SN	ImageHDU	39	(4606,)	float64
9	EMSN	ImageHDU	40	(4606, 10)	float64
10	DATA_MODEL	ImageHDU	39	(4606,)	float32

Figure 4.1 Info part of oneline/multiline fits file

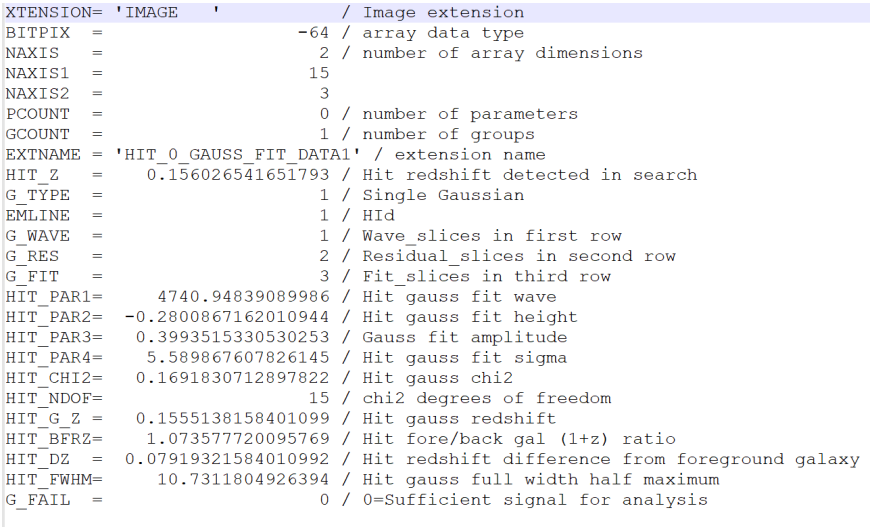
No.	Name	Type	Cards	Dimensions	Format	
0	HIT_0_GAUSS_FIT_DATA0	PrimaryHDU	24	(17, 3)		float64
1	HIT_0_GAUSS_FIT_DATA1	ImageHDU	25	(15, 3)		float64
2	HIT_0_GAUSS_FIT_DATA2	ImageHDU	25	(15, 3)		float64
3	HIT_0_GAUSS_FIT_DATA3	ImageHDU	25	(13, 3)		float64
4	HIT_0_GAUSS_FIT_DATA4	ImageHDU	25	(13, 3)		float64
5	HIT_0_GAUSS_FIT_DATA5	ImageHDU	25	(12, 3)		float64
6	HIT_0_GAUSS_FIT_DATA6	ImageHDU	25	(10, 3)		float64
7	HIT_0_GAUSS_FIT_DATA7	ImageHDU	25	(10, 3)		float64
8	HIT_0_GAUSS_FIT_DATA8	ImageHDU	25	(10, 3)		float64
9	HIT_0_GAUSS_FIT_DATA9	ImageHDU	25	(9, 3)		float64
10	HIT_0_GAUSS_FIT_DATA10	ImageHDU	25	(10, 3)		float64
11	HIT_0_GAUSS_FIT_DATA11	ImageHDU	25	(17, 3)		float64
12	HIT_1_GAUSS_FIT_DATA0	ImageHDU	25	(17, 3)		float64
13	HIT_1_GAUSS_FIT_DATA1	ImageHDU	25	(15, 3)		float64
14	HIT_1_GAUSS_FIT_DATA2	ImageHDU	25	(14, 3)		float64
15	HIT_1_GAUSS_FIT_DATA3	ImageHDU	25	(13, 3)		float64
16	HIT_1_GAUSS_FIT_DATA4	ImageHDU	25	(13, 3)		float64
17	HIT_1_GAUSS_FIT_DATA5	ImageHDU	25	(12, 3)		float64
18	HIT_1_GAUSS_FIT_DATA6	ImageHDU	25	(10, 3)		float64
19	HIT_1_GAUSS_FIT_DATA7	ImageHDU	25	(10, 3)		float64
20	HIT_1_GAUSS_FIT_DATA8	ImageHDU	25	(10, 3)		float64
21	HIT_1_GAUSS_FIT_DATA9	ImageHDU	25	(9, 3)		float64
22	HIT_1_GAUSS_FIT_DATA10	ImageHDU	25	(9, 3)		float64
23	HIT_1_GAUSS_FIT_DATA11	ImageHDU	25	(16, 3)		float64
24	HIT_2_GAUSS_FIT_DATA0	ImageHDU	25	(13, 3)		float64
25	HIT_2_GAUSS_FIT_DATA1	ImageHDU	25	(13, 3)		float64
26	HIT_2_GAUSS_FIT_DATA2	ImageHDU	25	(11, 3)		float64
27	HIT_2_GAUSS_FIT_DATA3	ImageHDU	25	(11, 3)		float64

Figure 4.2 Info part of onelineGuess/multilineGuess fits file



```
XTENSION= 'BINTABLE'                / binary table extension
BITPIX  =                    8 / array data type
NAXIS   =                    2 / number of array dimensions
NAXIS1  =                   56 / length of dimension 1
NAXIS2  =                    6 / length of dimension 2
PCOUNT  =                    0 / number of group parameters
GCOUNT  =                    1 / number of groups
TFIELDS =                    5 / number of table fields
TTYPE1  = 'index    '
TFORM1  = '1J       '
TTYPE2  = 'wave     '
TFORM2  = '1E       '
TTYPE3  = 'o2sn     '
TFORM3  = '1E       '
TTYPE4  = 'emsn     '
TFORM4  = '10E      '
TTYPE5  = 'z        '
TFORM5  = '1E       '
EXTNAME = 'HITS     '               / extension name
FILETYPE= 'SPEC     '
PLATE   =                10000
MJD     =                57346
FIBERID =                  116
SPECMODE= 'fiber    '
LINEMODE= 'multiline'
Z_NOQSO =             0.318597
O2SNMIN =                    4
MAXHITS =                   10
```

Figure 4.3 Header part of oneline/multiline fits file



```
XTENSION= 'IMAGE   '            / Image extension
BITPIX  =                  -64 / array data type
NAXIS   =                    2 / number of array dimensions
NAXIS1  =                   15
NAXIS2  =                    3
PCOUNT  =                    0 / number of parameters
GCOUNT  =                    1 / number of groups
EXTNAME = 'HIT_0_GAUSS_FIT_DATA1' / extension name
HIT_Z   =    0.156026541651793 / Hit redshift detected in search
G_TYPE  =                    1 / Single Gaussian
EMLINE  =                    1 / HId
G_WAVE  =                    1 / Wave_slices in first row
G_RES   =                    2 / Residual_slices in second row
G_FIT   =                    3 / Fit_slices in third row
HIT_PAR1=     4740.94839089986 / Hit gauss fit wave
HIT_PAR2=  -0.2800867162010944 / Hit gauss fit height
HIT_PAR3=   0.3993515330530253 / Gauss fit amplitude
HIT_PAR4=    5.589867607826145 / Hit gauss fit sigma
HIT_CHI2=   0.1691830712897822 / Hit gauss chi2
HIT_NDOF=                   15 / chi2 degrees of freedom
HIT_G_Z =   0.1555138158401099 / Hit gauss redshift
HIT_BFRZ=    1.073577720095769 / Hit fore/back gal (1+z) ratio
HIT_DZ  =  0.07919321584010992 / Hit redshift difference from foreground galaxy
HIT_FWHM=     10.7311804926394 / Hit gauss full width half maximum
G_FAIL  =                    0 / 0=Sufficient signal for analysis
```

Figure 4.4 Header part of onelineGuess/multilineGuess fits file

We should also comment that there are headers for all data in both types of fits file. The headers explain the data and it has some helpful structure of the data. Figure 4.3 and Figure 4.4 shows just two headers as an example for oneline/multiline fits file and onelineGuess/ multilineGuess fits file respectively.

We developed a python code and used the astropy.io.fits package to handle, read and access the data in the fits files. This library provides access to fits files. Fits (Flexible Image Transport System) is a portable file standard widely used in the astronomy community to store images and tables. Then we collected all data from fits files and created the proper dataset for data mining methods.

The target variable is a binary feature showing whether or not the record is a good hit. The problem we analyze in this study is the prediction of good hits, which can help inspect the background galaxies and strong lensing. Our sample of eBOSS has an imbalanced distribution of 25% bad hits and 75% good hits as shown in Table 4.3. The sample of MaNGA also has an imbalanced distribution of 25% good hits and 75% bad hits as shown in Table 4.4.

Table 4.1 Numeric features extracted from oneline and multiline fits files

RedShift	Emsn3	Emsn7	OIIB	HIB	HIA
o2sn	Emsn4	Emsn8	OIIA	OIIIB	NIIA
Emsn1	Emsn5	Emsn9	HID	OIIIA	SIIB
Emsn2	Emsn6	Emsn10	HIC	NIIB	SIIA

Table 4.2 Numeric features extracted from Gaussian model fits files

EMLINE	HIT_PAR3	HIT_CHI2	EMLINE	G_FAIL
HIT_PAR2	HIT_PAR5	HIT_FWHM	HIT_PAR2	

Table 4.3 Data distribution over the target variable (hit) for eBOSS sample

Nominal Value	**Absolute Count**	**Fraction**
Bad	68	25%
Good	213	75%

Table 4.4 Data distribution over the target variable (hit) for MaNGA sample

Nominal Value	**Absolute Count**	**Percentage**
Bad	15598	75%
Good	5194	25%

4.2. Manual Labeling

Every time the search code detects either a potential OII doublet of the background galaxy (with S/N > 6 or oneline hit), or at least 2 potential emission lines of the background galaxy (with S/N > 4 or multiline hit), a 'hit' is recorded to a database, and also saved in hit fits files, along with the Gauss fitting of the emission lines, and the spectra that the hit was found in.

Before manually inspect any of these hits, astronomers used the Gauss fitting parameters (such as FWHM, chi2, etc.) to identify fits that make more sense to be real emission lines of the background galaxy. They then manually inspect the hits with more sensible Gauss fits (or even multiple emission lines) to identify background emission line patterns. As an example they check if they can see a tall HIa and adjacent and smaller NIIa and NIIb emission lines and how well formed is the OII doublet (both spikes should be roughly the same height), and the OIIIb/OIIIa ratio should be 3:1. How well do these features stand out of the continuum, and how many of these emission lines can be seen at the expected redshift position, also help assure they are real. They also check if there are any signs that they may be bad, such as a nearby mask creating a false OII doublet, or a poorly subtracted part of the continuum results in an elevated residual flux region instead of a definitive emission line spike.

Hits with assuring emission line patterns of the background galaxy are manually labeled as 'good'. Hits identified as more likely to be random fake spikes or caused by affects such as masking are manually labeled as 'bad'.

'Good' hits are assuring emission lines of the background galaxy seen in the spectra recorded from a fiber. Since eBOSS fibers are foreground galaxy centered (one

fiber per galaxy), and their arc second coverage (about 2") is pretty much where the strong lensing regime is (eBOSS lenses have up to ~1" Einstein radius (i.e. strong lensing regime), but strong lensing happens when the main image is within twice the Einstein radius, or about 2" for eBOSS, i.e. its fiber coverage range), these automatically become lensing candidates. Good hits from MaNGA just assure they are emission lines from background galaxies. To determine if a MaNGA hit might be strongly lensed, astronomers use FIREFLY stellar density maps times a dark matter fraction to approximate the strong lensing regime of the foreground galaxy, and then see if the background galaxy is within twice the upper limit of our estimate of the strong lensing regime from the foreground galaxy center (if so, the background galaxy becomes a lens candidate if they can see its emission lines within twice this region).

4.3. Implementation

RapidMiner (40) is used to implement all experiments in this study. RapidMiner is data mining software which is capable of performing several different tasks such as data preparation, data analysis and reporting. RapidMiner is a java based open source software which has pre-built libraries for many data mining methods including the binary classification methods used in this study. Therefore, all the data manipulation (e.g., missing value imputation), model application and evaluation have been done using this powerful software.

Weka (41) is also used to implement some experiments that take more times to run. Weka is a machine learning software written in Java, and has the collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from the Java code.

4.4. Evaluation

The whole dataset is split to have 30 percent for parameter tuning and 70 percent for evaluation. For evaluation, 10-fold cross validation is used to evaluate the performance of different methods in terms of their Precision, Recall, Accuracy, AUC and F_measure. More specifically, after getting the evaluation results of each fold, the above measures are averaged over the 10 folds. The final 10 average values are reported in the results section.

4.5. Results

4.5.1. Base model application

In the first experiment we attempted to evaluate the effectiveness of different classification methods on the data we prepared from the eBOSS and MaNGA survey. Table 4.5 shows the results of this experiment for eBoss data and the results for MaNGA data is shown in table 4.6.

As seen in table 4.5 and table 4.6, for eBOSS data, Bayesian Network and Logistic Regression methods and for MaNGA data, Logistic Regression and ANN methods have the best performance outperforming all other methods in terms of accuracy, AUC, F_measure, precision and recall.

The reason for the poor performance of SVM is that they are not designed for imbalance dataset. Instead, they are more appropriate for balanced dataset. Moreover, the reason that Logistic Regression, Bayesian Network and ANN outperformed is that they can easily handle this problem for imbalance dataset.

Table 4.5 Performance of machine learning method on eBOSS dataset

Model	Precision (No)	Precision (Yes)	Recall (No)	Recall (Yes)	Accuracy	AUC	F_Measure
Bayes Net	83.82	94.84	83.82	94.84	92.17	98.10	94.53
Logistic Regression	90.48	94.95	83.82	97.18	93.93	97.30	95.89
Naïve Bayes	70.00	85.71	51.47	92.96	82.91	83.70	88.86
Decision Tree	80.36	89.78	66.18	94.84	87.91	79.40	92.19
ANN	81.97	91.82	73.53	94.84	89.67	79.40	93.00
k-NN	39.73	81.25	42.65	79.34	70.50	62.90	79.50
SVM	0.00	75.80	0.00	100.00	75.80	0.00	86.23

Table 4.6 Performance of machine learning method on MaNGA dataset

Model	Precision (No)	Precision (Yes)	Recall (No)	Recall (Yes)	Accuracy	AUC	F_Measure
Logistic Regression	86.4	83.00	96.30	54.44	85.82	88.10	83.00
ANN	88.30	72.70	92.10	63.40	84.90	86.4	84.60
Bayes Net	85	58.5	87.3	53.6	78.9	80.2	78.6
Naïve Bayes	89.27	31.25	36.37	86.87	48.99	74.7	46.09
Decision Tree	82.61	94.89	99.33	37.2	83.81	69.7	53.43
k-NN	68.57	24.73	3.39	95.34	26.36	41.5	39.26
SVM	0	24.2	0	100	24.19	0	38.94

4.5.2. Missing value imputation

The second experiment applied the missing value imputation by using the average value in order to analyze the effect of imputing the missing value. Table 4.7 and table 4.8 show the results of this experiment for eBOSS and MaNGA data set respectively.

As seen in Table 4.7 and Table 4.8, this imputation was not able to improve the performance of our best methods and decreased the performance measures of some methods. However, Bayesian Network and Logistic Regression methods still have the best performance for eBOSS and Logistic Regression and ANN have the best performance for MaNGA dataset.

This experiment shows that astronomy is different from many areas where such imputation could work. In other words, domain knowledge is required to impute the missing values and automated imputation may produce errors.

Table 4.7 Effectiveness of missing value imputation on eBOSS dataset

Model	Precision (No)	Precision (Yes)	Recall (No)	Recall (Yes)	Accuracy	AUC	F_Measure
Logistic Regression	91.80	94.55	82.35	97.65	93.94	97.10	95.93
Bayes Net	70.67	92.72	77.94	89.67	86.85	95.2	90.84
SVM	92.31	82.75	35.29	99.06	83.65	92.9	89.89
Naïve Bayes	65.45	85.84	52.94	91.08	81.83	84.2	88.21
ANN	81.54	93.06	77.94	94.37	90.39	81.8	93.58
Decision Tree	79.25	88.60	61.76	94.84	86.83	80.90	91.26
k-NN	71.19	88.29	61.76	92.02	84.7	74.7	89.97

Table 4.8 Effectiveness of missing value imputation on MaNGA dataset

Model	Precision (No)	Precision (Yes)	Recall (No)	Recall (Yes)	Accuracy	AUC	F_Measure
Logistic Regression	86.4	84	96.6	54.3	85.98	88.4	84.9
ANN	86.7	75.2	93.7	57	84.54	82.3	83.8
SVM	82.30	81.00	97.09	37.27	82.15	81.20	51.01
Bayes Net	85.6	57.4	86.1	56.5	78.67	76.6	78.6
k-NN	87.24	62.21	87.24	62.21	80.99	75.1	62.03
Naïve Bayes	88.51	38.72	59.53	76.8	63.85	74.6	51.69
Decision Tree	82.2	94.17	99.27	35.46	83.33	67.3	51.43

4.5.3. Feature Weighting Effect

The third experiment is designed to evaluate the effect of feature selection on both dataset. We used several feature weighting method including chi square, information gain, Gini index and correlation to select the proper features. Since chi square has the best performance, we report the result of this feature weighting method.

4.5.3.1. Chi Square weighting for eBOSS

Table 4.9 shows the results of this experiment for eBOSS dataset when we select top 85 features out of all features form chi square weights. As you can see in the table, same as the missing value imputation, feature selection does not have significant effect on the performance.

However, this selection of top 85 features shows that we can reduce the number of features to 85 and still have the same performance and so, all individual features is not important for the prediction task. Table 4.10 shows the top 5 features and bottom 5 features which removed for this experiment.

Table 4.9 Effectiveness of feature selection on eBoss dataset (top 85 features)

Model	Precision No	Precision Yes	Recall No	Recall Yes	Accuracy	AUC	F_Measure
Logistic Regression	90.48	94.95	83.82	97.18	93.93	98.10	97.04
Bayes Net	80.82	95.67	86.76	93.43	91.81	97.90	94.34
Naïve Bayes	68.42	82.72	38.24	94.37	80.79	82.10	87.78
ANN	75.00	90.78	70.59	92.49	87.17	80.60	91.44
Decision Tree	84.91	89.91	66.18	96.24	88.98	80.00	92.83
k-NN	40.28	81.34	42.65	79.81	70.86	63.10	79.88
SVM	0.00	75.80	0.00	100.00	75.83	0.00	85.94

Table 4.10 Weight of top 5 features and bottom 5 features

Top 5		Bottom 5	
Feature	**Normalized Weight**	**Feature**	**Normalized Weight**
HIT_PAR4_data11	1	G_FAIL_data0	0
HIT_PAR3_data11	0.95	G_FAIL_data2	0.0009
HIT_FWHM_data11	0.91	G_FAIL_data5	0.001
HIT_FWHM_data10	0.80	G_FAIL_data3	0.006
HIT_PAR2_data10	0.70	G_FAIL_data4	0.0096

We decided to show the effect of other feature selection methods to see the least number of features that are required in order to have a reasonable performance for eBOSS dataset. Table 4.11 and Table 4.12 show the performance for selection of the top 50 and the top 20 features respectively.

Table 4.11 Effectiveness of feature selection on eBoss dataset (top 50 features)

Model	**Precision No**	**Precision Yes**	**Recall No**	**Recall Yes**	**Accuracy**	**AUC**	**F_Measure**
Logistic Regression	87.30	94.04	80.88	96.24	92.5	97.50	94.93
Bayes Net	71.95	95.48	86.76	89.20	88.62	96.10	91.74
Naïve Bayes	76.19	90.83	70.59	92.96	87.55	91.40	91.52
Decision Tree	61.54	93.68	82.35	83.57	83.26	80.50	86.79
k-NN	70.59	78.79	17.65	97.65	78.33	66.90	97.65
ANN	86.89	93.18	77.94	96.24	91.81	50.30	94.42
SVM	0.00	75.80	0.00	100.00	75.83	0.00	85.94

Table 4.12 Effectiveness of feature selection on eBoss dataset (top 20 features)

Model	Precision No	Precision Yes	Recall No	Recall Yes	Accuracy	AUC	F_Measure
Logistic Regression	91.94	94.98	83.82	97.65	94.3	97.80	96.12
Bayes Net	74.39	96.48	89.71	90.14	90.02	96.1	92.85
Naïve Bayes	77.03	94.69	83.82	92.02	90.02	95.2	93.03
Decision Tree	76.47	92.49	76.47	92.49	88.58	85.3	92.23
ANN	93.1	93.72	79.41	98.12	93.6	61.1	95.68
k-NN	34.86	82.56	55.88	66.67	64.03	61.00	72.62
SVM	0.00	78.20	0.00	100.00	74.83	0.00	86.76

As seen in Table 4.11 and Table 4.12, we improved the performance of some methods including the best method by reducing the features to the top 20 features. It should be mentioned that the performance decreases significantly when there are less than 20 features in the data set. Therefore, 20 features are required to have a good performance.

4.5.3.2. Chi Square weighting for MaNGA

Table 4.13 shows the results of selecting features based on Chi Square weighting on MaNGA dataset when we select top 85 features out of all features. As shown in the table, we can have fewer features and still keep the same performance. Table 4.14 shows the top 5 features and bottom 5 features for this experiment.

Table 4.13 Effectiveness of feature selection on MaNGA dataset (top 85 features)

Model	Precision No	Precision Yes	Recall No	Recall Yes	Accuracy	AUC	F_Measure
Logistic Regression	85.90	83.90	96.70	52.20	85.56	87.50	84.30
ANN	88.30	75.30	93.10	62.90	85.57	86.40	85.10
Bayes Net	84.6	58.8	87.9	52	78.92	79.8	78.5
Decision Tree	82.73	93.14	99.07	37.91	78.92	79.80	78.50
Naïve Bayes	86.50	45.20	73.60	65.50	71.54	76.00	73.00
SVM	84.10	91.30	98.60	44.20	85.00	71.40	83.00
k-NN	69.7	24.8	3.50	95.40	26.46	38.30	14.80

Table 4.14 Weight of top 5 features and bottom 5 features

Top 5		**Bottom 5**	
Feature	**Normalized Weight**	**Feature**	**Normalized Weight**
emsn7	1	HIT_PAR4_data0	0
HIT_PAR1_data5	0.93368728	HIT_CHI2_data0	0
z	0.92430251	HIT_CHI2_data1	0
HIT_PAR1_data11	0.90424746	HIT_CHI2_data2	7.42E-05
emsn5	0.89018862	HIT_CHI2_data11	7.42E-05

4.5.4. Feature selection and missing value replacement effect

Experiment 4 shows the effect of feature selection and missing value imputation together on both dataset. Table 4.17 and Table 4.19 compares the performance of different data mining methods when we select top 85 features by chi square weighting and impute the missing values by average for those features in eBOSS and MaNGA respectively.

Table 4.15 Effectiveness of both feature selection and missing value imputation on eBoss dataset (top 85 features)

Model	Precision No	Precision Yes	Recall No	Recall Yes	Accuracy	AUC	F_Measure
Logistic Regression	92.06	95.41	85.29	97.65	94.66	97.50	96.48
Bayes Net	70.51	93.60	80.88	89.20	87.20	94.90	91.12
SVM	96.15	83.14	36.76	99.53	84.35	93.90	90.31
Naïve Bayes	68.52	86.34	54.41	92.02	82.92	83.70	88.81
Decision Tree	83.02	89.47	64.71	95.77	88.26	82.50	92.17
k-NN	72.41	88.34	61.76	92.49	85.06	74.60	90.36
ANN	81.67	91.40	72.06	94.84	89.31	68.50	93.04

Table 4.16 Weight of top 5 features and bottom 5 features for eBOSS

Top 5		**Bottom 5**	
Feature	**Normalized Weight**	**Feature**	**Normalized Weight**
HIT_PAR4_data11	1	G_FAIL_data0	0
HIT_PAR3_data11	0.93	G_FAIL_data2	0.0009
HIT_FWHM_data10	0.84	G_FAIL_data5	0.001
HIT_PAR2_data10	0.74	G_FAIL_data3	0.007
HIT_FWHM_data11	0.74	G_FAIL_data4	0.009

As seen in the table 4.17 the performance improved a little more in this experiment in comparison of when we apply feature selection and missing value replacement separately for eBOSS dataset. Table 4.18 shows the weight of top 5 and bottom 5 features.

Table 4.19 shows the effect of this experiment for MaNGA dataset. Table 4.20

shows the weight of top 5 and bottom 5 features for this experiment.

Table 4.17 Effectiveness of both feature selection and missing value imputation on MaNGA dataset (top 85 features)

Model	Precision No	Precision Yes	Recall No	Recall Yes	Accuracy	AUC	F_Measure
Logistic Regression	85.90	84.30	96.80	52.20	85.63	87.70	84.40
ANN	88.10	76.00	93.50	62.00	85.61	86.00	85.10
Decision Tree	89.60	75.10	92.50	67.60	86.30	82.10	86.10
Bayes Net	85.10	57.80	86.80	54.20	78.67	77.60	78.40
Naïve Bayes	84.42	60.21	88.83	50.75	79.32	74.80	55.06
k-NN	87.17	61.24	87.04	61.51	80.66	74.70	61.38
SVM	84.10	92.00	98.70	43.90	85.03	71.30	83.00

Table 4.18 Weight of top 5 features and bottom 5 features for MaNGA

Top 5		Bottom 5	
Feature	**Normalized Weight**	**Feature**	**Normalized Weight**
emsn7	1.0	HIT_PAR4_data3	0.0
HIT_PAR1_data11	0.92	HIT_PAR4_data0	5.942E-5
z	0.92	HIT_CHI2_data0	5.942E-5
emsn5	0.8934	HIT_CHI2_data1	5.942E-5
HIT_PAR1_data5	0.88	HIT_CHI2_data2	1.33E-4

4.5.5. Adding binary features for emission lines effect

For experiment 6, we decided to add a set of binary variables to the data using the domain knowledge. There are several zeros for the emission lines features. Some of those zeros are real and they measured as zero. Since some galaxies were very far from the earth, zero was used for the emission line of those galaxies and so they are not real. We have done the specific calculation to detect the fake zeros. Then, we added new binary

features for each emission line. If it is real, we make it as 1, otherwise it is 0. In the original emission lines variables, we consider those fake zeroes as missing values.

Table 4.21 shows the result of this experiment for eBOSS dataset and Table 4.22 shows the result for MaNGA dataset. As seen below, this experiment improved the performance measures a little for both data sets. The best methods still are Logistic Regression and Bays Net for eBOSS and are Logistic Regression and ANN for MaNGA dataset.

Table 4.19 Effectiveness of adding binary features on eBOSS dataset

Model	Precision No	Precision Yes	Recall No	Recall Yes	Accuracy	AUC	F_Measure
Logistic Regression	95.83	90.77	97.18	86.76	94.66	98.60	88.10
Bayes Net	93.87	79.71	93.43	80.88	90.38	97	80.02
k-NN	79.48	100.00	100.00	19.12	80.44	82.40	32.10
ANN	92.34	86.44	96.24	75.00	91.10	75.20	78.29
Naïve Bayes	85.44	78.95	97.78	33.33	84.97	68.80	46.88
Decision Tree	86.36	89.74	98.12	51.47	86.83	22.30	64.54
SVM	0.00	24.20	0.00	100.00	24.20	0.00	38.95

Table 4.20 Effectiveness of adding binary features on MaNGA dataset

Model	Precision No	Precision Yes	Recall No	Recall Yes	Accuracy	AUC	F_Measure
Logistic Regression	86.30	83.90	96.60	54.10	85.94	88.50	84.80
ANN	87.50	72.40	92.40	60.20	84.34	85.00	83.80
SVM	82.36	81.50	97.17	37.49	82.26	81.10	51.28
Bayes Net	85.2	58.6	87.2	54.4	78.99	79.5	42.6
Naïve Bayes	88.25	40.32	63.15	74.76	66.05	74.60	52.37
Decision Tree	82.68	94.93	99.33	37.50	83.89	69.80	53.71
k-NN	80.05	32.96	65.42	51.04	61.83	59.90	40.41

CHAPTER 5

CONCLUSION

This paper was an attempt to predict and detect gravitational lens candidates using data mining methods. The goal was to automate and replace this detection process performed by human. To achieve this, the first task was to find a decent set of features by collaborating with astronomy experts. The second task was to apply different classification methods on the extracted datasets. Our results show Logistic Regression has the highest accuracy for the prediction task for both dataset that we used. The third task was to evaluation the impact of the feature selection. Chi square weighting feature selection was applied to find the best set of the useful features. The results showed that 85 features chosen by chi square weighting are the most useful features.

CHAPTER 6

REFERENCES

1. Hoekstra H, Jain B. Weak gravitational lensing and its cosmological applications. Annual Review of Nuclear and Particle Science. 2008;58:99-123.
2. Treu T, Dutton AA, Auger MW, Marshall PJ, Bolton AS, Brewer BJ, et al. The SWELLS survey–I. A large spectroscopically selected sample of edge-on late-type lens galaxies. Monthly Notices of the Royal Astronomical Society. 2011;417(3):1601-20.
3. Shu Y, Bolton AS, Mao S, Kochanek CS, Pérez-Fournon I, Oguri M, et al. The BOSS emission-line lens survey. IV. Smooth lens models for the BELLS GALLERY sample. The Astrophysical Journal. 2016;833(2):264.
4. Bolton AS, Burles S, Schlegel DJ, Eisenstein DJ, Brinkmann J. Sloan Digital Sky Survey Spectroscopic Lens Search. I. Discovery of Intermediate-Redshift Star-forming Galaxies behind Foreground Luminous Red Galaxies. The Astronomical Journal. 2004;127(4):1860.
5. Bolton AS, Burles S, Koopmans LV, Treu T, Moustakas LA. The Sloan Lens ACS Survey. I. A Large Spectroscopically Selected Sample of Massive Early-Type Lens Galaxies. The Astrophysical Journal. 2006;638(2):703.
6. Bloom J, Richards J, Nugent P, Quimby R, Kasliwal M, Starr D, et al. Automating discovery and classification of transients and variable stars in the synoptic survey era. Publications of the Astronomical Society of the Pacific. 2012;124(921):1175.
7. Dawson KS, Kneib J-P, Percival WJ, Alam S, Albareti FD, Anderson SF, et al. The SDSS-IV extended baryon oscillation spectroscopic survey: Overview and early data. The Astronomical Journal. 2016;151(2):44.
8. Bundy K, Bershady MA, Law DR, Yan R, Drory N, MacDonald N, et al. Overview of the SDSS-IV MaNGA survey: mapping nearby galaxies at Apache Point observatory. The Astrophysical Journal. 2014;798(1):7.
9. Overbye D. Astronomers observe supernova and find they are watching reruns. New York Times, USA. 2015.
10. Negrello M, Hopwood R, De Zotti G, Cooray A, Verma A, Bock J, et al. The detection of a population of submillimeter-bright, strongly lensed galaxies. science. 2010;330(6005):800-4.
11. Bartelmann M. Gravitational lensing. Classical and Quantum Gravity. 2010;27(23):233001.
12. Treu T. Strong lensing by galaxies. Annual Review of Astronomy and Astrophysics. 2010;48:87-125.

13. Zitrin A, Broadhurst T, Barkana R, Rephaeli Y, Benítez N. Strong-lensing analysis of a complete sample of 12 MACS clusters at z> 0.5: mass models and Einstein radii. Monthly Notices of the Royal Astronomical Society. 2011;410(3):1939-56.
14. Hewett PC, Warren SJ, Willis JP, Bland-Hawthorn J, Lewis GF. High-redshift gravitationally lensed galaxies and tunable filter imaging. arXiv preprint astro-ph/9905316. 1999.
15. Peacock JA, Cole S, Norberg P, Baugh CM, Bland-Hawthorn J, Bridges T, et al. A measurement of the cosmological mass density from clustering in the 2dF Galaxy Redshift Survey. Nature. 2001;410(6825):169.
16. Talbot MS, Brownstein JR, Bolton AS, Bundy K, Andrews BH, Cherinka B, et al. SDSS-IV MaNGA: the spectroscopic discovery of strongly lensed galaxies. Monthly Notices of the Royal Astronomical Society. 2018;477(1):195-209.
17. Hand DJ. Principles of data mining. Drug safety. 2007;30(7):621-2.
18. Larose DT. Introduction to data mining: Wiley Online Library; 2005.
19. Witten IH, Frank E, Hall MA, Pal CJ. Data Mining: Practical machine learning tools and techniques: Morgan Kaufmann; 2016.
20. Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, et al. Top 10 algorithms in data mining. Knowledge and information systems. 2008;14(1):1-37.
21. Ma W, Tran D, Sharma D, editors. A novel spam email detection system based on negative selection. Computer Sciences and Convergence Information Technology, 2009 ICCIT'09 Fourth International Conference on; 2009: IEEE.
22. Hosmer Jr DW, Lemeshow S, Sturdivant RX. Applied logistic regression: John Wiley & Sons; 2013.
23. Starkweather J, Moske AK. Multinomial logistic regression. Consulted page at September 10th: http://www unt edu/rss/class/Jon/Benchmarks/MLR_JDS_Aug2011 pdf. 2011;29:2825-30.
24. Zhao Y, Zhang Y. Comparison of decision tree methods for finding active objects. Advances in Space Research. 2008;41(12):1955-9.
25. Banfield RE, Hall LO, Bowyer KW, Kegelmeyer WP. A comparison of decision tree ensemble creation techniques. IEEE transactions on pattern analysis and machine intelligence. 2007;29(1):173-80.
26. Menard S. Applied logistic regression analysis: Sage; 2002.
27. Cox DR. The regression analysis of binary sequences. Journal of the Royal Statistical Society Series B (Methodological). 1958:215-42.
28. Cover T, Hart P. Nearest neighbor pattern classification. IEEE transactions on information theory. 1967;13(1):21-7.
29. Peterson LE. K-nearest neighbor. Scholarpedia. 2009;4(2):1883.
30. Weinberger KQ, Blitzer J, Saul LK, editors. Distance metric learning for large margin nearest neighbor classification. Advances in neural information processing systems; 2006.
31. Lewis DD, editor Naive (Bayes) at forty: The independence assumption in information retrieval. European conference on machine learning; 1998: Springer.
32. Jeffreys H. Scientific inference: Cambridge University Press; 1973.
33. Wang S-C. Artificial neural network. Interdisciplinary computing in java programming: Springer; 2003. p. 81-100.

34. Ball NM, Brunner RJ. Data mining and machine learning in astronomy. International Journal of Modern Physics D. 2010;19(07):1049-106.
35. Singh M, Provan GM. Efficient learning of selective Bayesian network classifiers. 1995.
36. Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Machine learning. 1997;29(2-3):131-63.
37. Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods: Cambridge university press; 2000.
38. Gu Q, Zhu L, Cai Z, editors. Evaluation measures of the classification performance of imbalanced data sets. International Symposium on Intelligence Computation and Applications; 2009: Springer.
39. Powers DM. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 2011.
40. Hofmann M, Klinkenberg R. RapidMiner: Data mining use cases and business analytics applications: CRC Press; 2013.
41. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter. 2009;11(1):10-8.