

MASARYK UNIVERSITY
DEPARTMENT OF THEORETICAL PHYSICS AND ASTROPHYSICS



MASTER'S THESIS
VIRTUAL OBSERVATORY AND DATA MINING
SUPERVISOR: RNDr. PETR ŠKODA, CSc.

JAROSLAV VÁŽNÝ

BRNO 2011

--

Declaration

I declare that I wrote my diploma thesis independently and exclusively using sources cited. I agree with borrowing the work and its publishing.

In Brno:

What is it that makes us human? It's not something you can program. You can't put it into a chip. It's the strength of the human heart. The difference between us and machines.

Marcus Wright

Acknowledgements

I would like to thank Filip Hroch and Petr Škoda for their remarkable support and patience not only during this project. I greatly appreciated comments and help from following friends: Tereza Jeřábková, Josef Pacula and Petr Šafařík.

Funding for the SDSS and SDSS-II has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, the U.S. Department of Energy, the National Aeronautics and Space Administration, the Japanese Monbukagakusho, the Max Planck Society, and the Higher Education Funding Council for England. The SDSS Web Site is <http://www.sdss.org/>.

This publication makes use of data products from the Two Micron All Sky Survey, which is a joint project of the University of Massachusetts and the Infrared Processing and Analysis Center/California Institute of Technology, funded by the National Aeronautics and Space Administration and the National Science Foundation.

This research has made use of data obtained from the High Energy Astrophysics Science Archive Research Center (HEASARC), provided by NASA's Goddard Space Flight Center.

This research has made use of the SIMBAD database, operated at CDS, Strasbourg, France.

This research has made use of The WEKA Data Mining Software [[Hall et al., 2009](#)].

This research has made use of Free software [[Free Software Foundation, 2007](#)].

Abstract

Modern astrophysics and natural sciences in general become extensively penetrated by Computer Science. Petabyte scale databases, GRID Computing and Data Mining become routine part of scientific work. Skills related to information technologies are now essential. The new concept of data infrastructure named Virtual Observatory naturally emerged from digitalized surveys. Based on proven standards offers ideal basis for dealing with distributed heterogeneous data. This thesis is a case study of using Virtual Observatory and Data Mining technologies to proceed automatics classification of Be stars. Photometric and spectra classification were done on large scale sample of almost 200 000 spectra from SDSS Segue survey.

Many byproduct originated during the work on the thesis. Wiki pages, Virtual Observatory and Data Mining documentation and about dozen of programs for data manipulation, spectra fitting and result publishing. Everything is given free to the public on the web of the project.

http://physics.muni.cz/~vazny/wiki/index.php/Diploma_work



Contents

Contents	v
List of Figures	vii
Nomenclature	viii
1 Virtual Observatory (VO)	3
1.1 Data avalanche: Opportunity or disaster?	3
1.2 International Virtual Observatory Alliance (IVOA)	4
1.3 Architecture	4
1.4 VOResources	5
1.5 Data Access Protocols	7
1.5.1 Cone Search Protocol	7
1.5.2 Simple Image Access Protocol	8
1.5.3 Simple Spectra Access Protocol	9
1.6 Data Formats	10
1.6.1 VOTable	10
1.6.2 FITS	11
1.7 Tools & Libraries	12
2 Data Mining	13
2.1 Supervised Methods	13
2.1.1 Decision Tree (DT)	13
2.1.1.1 Cross-validation	14
2.1.1.2 Example: Classifying Galaxies Stars and QSO using color indices	15
2.2 Data Mining tools	17
2.2.1 Weka	17
2.2.2 SVM lib	17
2.2.3 DAME	17
3 Be candidates	19
3.1 Be stars	19
3.2 Photometric Data Mining	22
3.2.1 Data preprocessing	22
3.2.2 Classification	23

3.3	Spectral Data Mining	24
3.3.1	Testing Data	24
3.3.2	Training Data	24
3.3.2.1	Spectra Reduction	26
3.3.3	Spectra Lines Characteristics	27
3.3.3.1	Normalization	29
3.3.3.2	The hight of the H α line	29
3.3.3.3	The noise level of the spectrum	29
3.3.3.4	The width of the H α line	29
3.3.4	Data Mining	30
3.3.5	Results	33
3.3.6	Experiment	34
4	Conclusion	41
Appendix1:	Spectra of result objects	43
Appendix2:	Spectra of Be stars from Ondřejov	49
References		55

List of Figures

1	Astroinformatics in the context of astronomy [Ball and Schade, 2010]	1
2	Thesis structure	2
1.1	Chapter structure	3
1.2	IVOA members	4
1.3	VO Architecture	5
1.4	UML diagram of VOResource	6
2.1	Chapter structure	13
2.2	Color Diagram of the problem. It shows that individual object classes occupies different regions in the diagram	16
3.1	Chapter structure	19
3.2	Model of a typical Be star. Emission lines coming from an equatorial disk is added to the photo-spheric absorption spectrum. Central B star emits UV (Lyman continuum) and ionizes the disk, which in turn re-emits at high wavelength such as visible domain. [Hirata and Kogure, 1984]	20
3.3	Example of spectra of Be stars based on view angle [Slettebak, 1988]	20
3.4	Schematic diagram of the photometric Data Mining process. The lists of confirmed Be stars consisted of Hipparcos IDs, this was correlated with Hipparcos catalog to obtain right ascension and declination of the objects and subsequently cross-matched with 2MASS catalog to get photometric data. The second set of B stars were acquired in similar manner but using SQL the condition was set to get B type stars different from the list of Be stars.	21
3.5	Color diagram of confirmed Be stars Vs B stars	22
3.6	Color diagram of confirmed Be stars, B stars with errors	23
3.7	Schematic diagram of the spectral Data Mining process. Using SSA protocol the spectra from Ondejov server was acquired based on the list from photometric study. Convolution with profile of SDSS slit had to be performed to ensure compatibility with SDSS. Afterwards desired features were extracted automatically from the spectra after the continuum and H α line were fitted by appropriate functions. The same was done for spectra from SDSS except the convolution process.	25

3.8 Reduction of Ondejov's spectra of the Be star 4 Hercules. The top figure shows Gaussian function used for convolution with the spectrum, followed by the original spectrum then there is a spectrum after convolution with the Gaussian function. The last is the final spectrum after reduction.	28
3.9 Normalized spectrum of Be star 60 Cyg. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line correspond to the maximum value in the region of 50 Å. The Gaussian fit is in red. Although the fit is almost perfect, this approach fails to get characteristic "double peak" of the emission line.	31
3.10 Normalized spectrum of Be star HR 8682. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line correspond to the maximum value in the region of 50 Å. The Gaussian fit is in red.	32
3.11 Example 1: SDSS J035747.16-063850.7	35
3.12 Example 2: SDSS J094325.89+520128.6	35
3.13 Example 3: SDSS J120729.12+003659.8	36
3.14 Example 4: SDSS J120908.18+194035.8	36
3.15 Spectrum of 4 Her. Be star	37
3.16 Spectrum of HR 7418 (Albireo B). A fast-rotating Be star, with an equatorial rotational velocity of at least 250 kilometers per second. Its surface temperature has been spectroscopically estimated to be about 13.200 K	37
3.17 Spectrum of 6 Cepheus. Be star	38
3.18 Spectrum of Gamma Cassiopeiae. Be Star	38



Introduction

From the dawn of its existence astronomy has been starving for data but in the last few decades the situation has changed and now we are facing data deluge of biblical proportions. The data are not just increasing in size but also in complexity and dimensionality [Ball and Schade, 2010]. Astroinformatics is the new field of science which has emerged from this technology driven progress. Virtual Observatory, Machine Learning, Data Mining, Grid Computing are just few examples of new tools available to scientists.

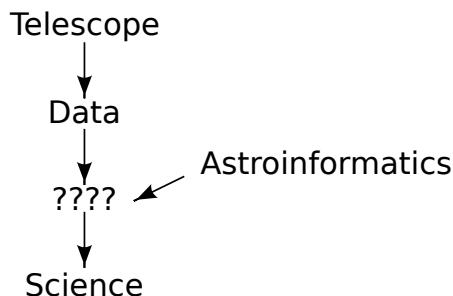


Figure 1: Astroinformatics in the context of astronomy [Ball and Schade, 2010]

The astronomers are not alone and particle physics, biology and other sciences are also in the vanguard of the data intensive science. This is great opportunity for interdisciplinary collaboration.

This work deals with the problem of semi-automatic procedures for finding Be stars [Porter and Rivinius, 2003] candidates in the astronomy surveys. More than straight forward process it's trial and error approach probing new possibilities.

The aim of this work is to be introductory to the technologies of Virtual Observatory and massive data processing in general.

Chapter one is an introduction to the technologies related to Virtual Observatory. The motivation behind the concept is given without paying too much attention to historical details. Main principles and protocols are discussed and explained. Important aspect are demonstrated on numerous examples. Chapter two is an introduction to Machine Learning and Data Mining in the context of astrophysics. Only methods used in practical part of this work are described in detail: Decision Trees and Support Vector Machines. Examples of several classifications are demonstrated. Third chapter introduces issues of Be stars. Chapter Four is practical application of previously described technologies and methods. Training data of confirmed Be

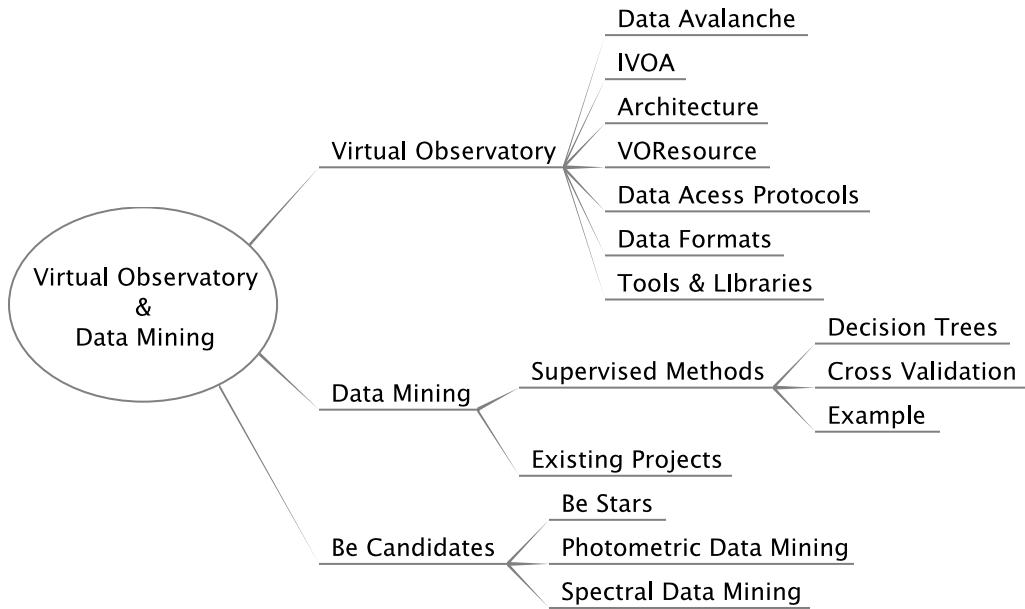


Figure 2: Thesis structure

stars from Ondrejov are correlated with others catalogues to obtain color indexes and spectra. Results are processed by Data Mining algorithms using several libraries and tools. In the last chapter achieved results are critically discussed.

Many scripts were written to achieve individual goals. In the text there are numerous commented snippets of codes. Their purpose is to demonstrate the concept and they are therefore short and without auxiliary technicalities such are error handling etc. They are mostly Python and shell scripts. Any interested person can obtain the full source codes (including thesis itself) from GIT repository ¹.

Name	Description
analyse	Check the wavelength range, rename according to target name
getSpectraList	Create SSA Compliant list from
getSpectra	Get spectra links from SSA Server
madmax	Extract features from spectra
convolve	Reduction of Ondejov's spectra
pf	Print Fits. Shows the spectrum
dm	Perform classification
makeHTML	Creates HTML pages of results

Table 1: Scripts developed within the scope of the thesis.

Activities related to this work went beyond this text. Wiki pages ² were created to present the results and discuss related topic with supervisor as well as with others scientist around the world. Source codes were maintained by GIT version system allowing easy sharing. All software used and produced are open source.

¹[git://github.com/astar/diplomaWork](https://github.com/astar/diplomaWork)

²http://physics.muni.cz/~vazny/wiki/index.php/Diploma_work

Virtual Observatory (VO)

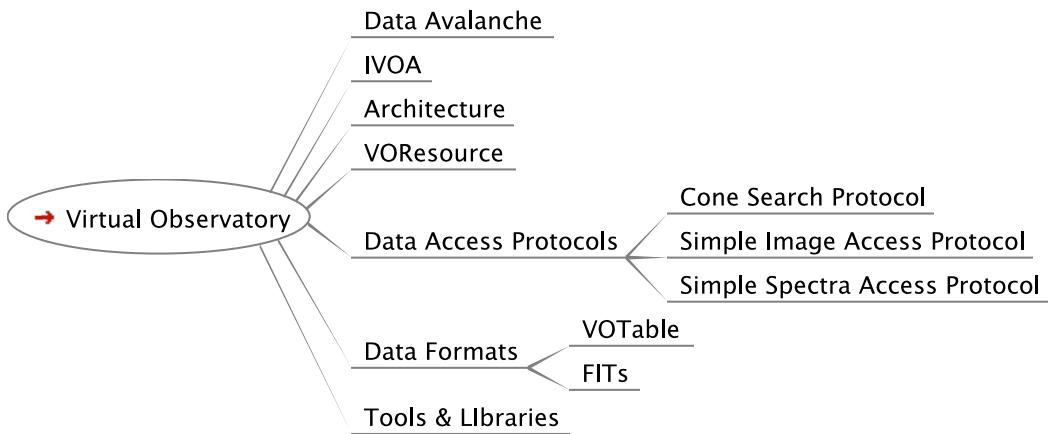


Figure 1.1: Chapter structure

1.1 Data avalanche: Opportunity or disaster?

There are two important trends in current astronomy surveys:

- Size: The cumulative compressed data holdings of the ESO archive will reach 1 Petabyte by 2012 [Hanisch and Quinn, 2010]. Projects like Large Synoptic Survey Telescope (LSST) will produce about 30 TB per night, leading to a total database over the ten years of operations of 60 PB for the raw data [Becla et al., 2006].
- Complexity: Modern surveys will cover the sky in different wave-bands, from gamma- and X-rays, optical, infrared to radio. The ability to cross correlate these observations together may lead to new understanding of physical phenomena. [Hanisch and Quinn, 2010]

Such an amount of data is not possible to transfer over the network. Data resources are heterogeneous, distributed and decentralized in nature.

There is an interesting analogy with the problem (and the solution) which scientists discovered during LEP project at CERN. Their problem was too many documents

in different formats. Tim Berners-Lee¹ designed set of protocols (URIs, HTTP and HTML) which allowed link and share documents [Berners-Lee and Cailliau, 1990]. This was recognized as generally useful and World Wide Web was born. An important role plays the World Wide Web Consortium (W3C) in developing Web standards².

1.2 International Virtual Observatory Alliance (IVOA)

What is necessary is sets of standards and protocols to deal with heterogeneous distributed data and authority which encourages their implementation. Such an authority is the International Virtual Observatory Alliance (IVOA). It comprises 19 VO programs from Argentina, Armenia, Australia, Brazil, Canada, China, Europe, France, Germany, Hungary, India, Italy, Japan, Russia, Spain, the United Kingdom, and the United States and inter-governmental organizations (ESA and ESO) [Hanisch and Quinn, 2010].

Standards and specifications produced by IVOA can be obtained at <http://www.ivoa.net/>.



Figure 1.2: IVOA members

1.3 Architecture

The Architecture is depicted on the figure 1.3. The level of abstraction goes from top to bottom. Starting with interfaces, used by people or applications to discover resources. Next level is the service layer implemented by standard protocols, followed by the hardware level where actual data are stored. This onion like structure hides the complexity of the lower layer and provide data and meta-data to the higher layer. This concept is similar to TCP/IP³ protocol.

The VO architecture is serviced oriented. Each service is autonomous with well defined defined boundaries. Very important aspect of VO implementation is the adoption of formats and protocols used in astronomy (FITS) and Computer Science

¹ Sir Timothy John "Tim" Berners-Lee. British engineer and computer scientist and MIT professor credited with inventing the World Wide Web.

²Prior to its creation, incompatible versions of HTML were offered by different vendors, increasing the potential for inconsistency between web pages.

³TCP/IP (Transmission Control Protocol/Internet Protocol). The basic communication language or protocol of the Internet.

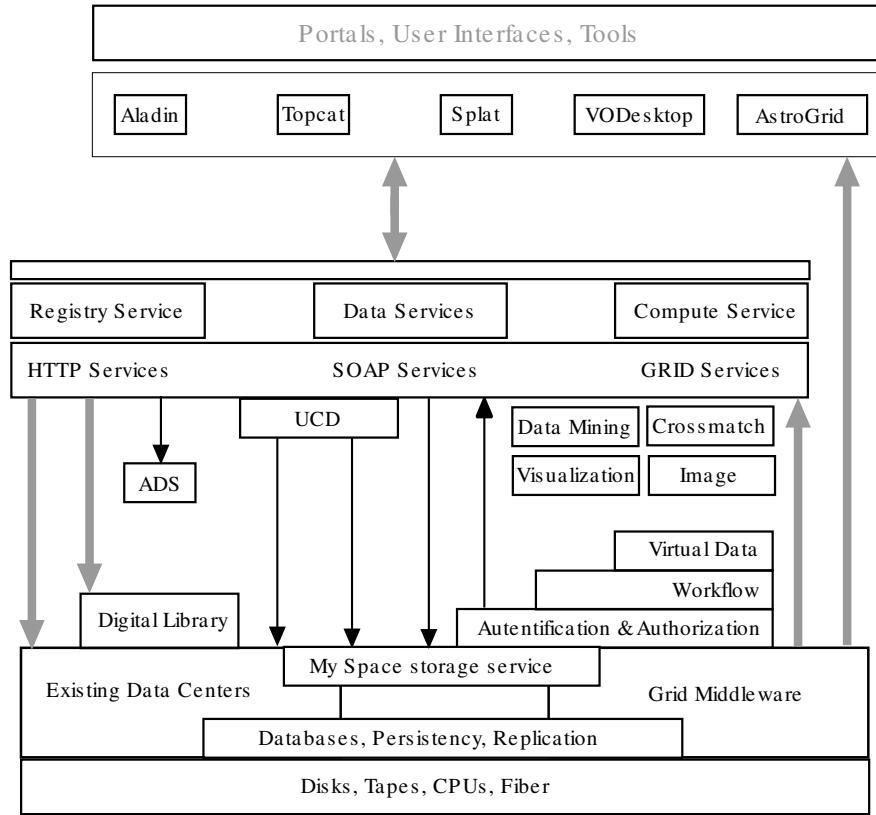


Figure 1.3: VO Architecture

(XML¹, Web service² SOAP³) for many years. In other words VO does not try to reinvent the wheel but it stands on the shoulders of giants.

1.4 VOResources

A resource is a general term referring to a VO element that can be described in terms of who curates or maintains it and which can be given a name and a unique identifier. Just about anything can be a resource: it can be an abstract idea, such as sky coverage or an instrumental setup, or it can be fairly concrete, like an organization or a data collection. [Benson et al., 2009]

UML⁴ diagram of the resource is on the figure 1.4. Next paragraph is an attempt to explain this diagram to non-programmers. Full arrow means generalization, Resource can be a generalization of organization, data collection, application or service. Single arrow means association. Organization can be linked (associated) together with other organization (multiplicity is represented by number 1, 0..). The same is true for data collection. Organization is a generalization of and/or provider

¹Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form.

²method of communication between two electronic devices over a network.

³Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks.

⁴Unified Modeling Language. Standardized general-purpose modeling language in the field of object-oriented software engineering.

which can own zero to N services. Diamond means the aggregation. Publisher can have any resources.

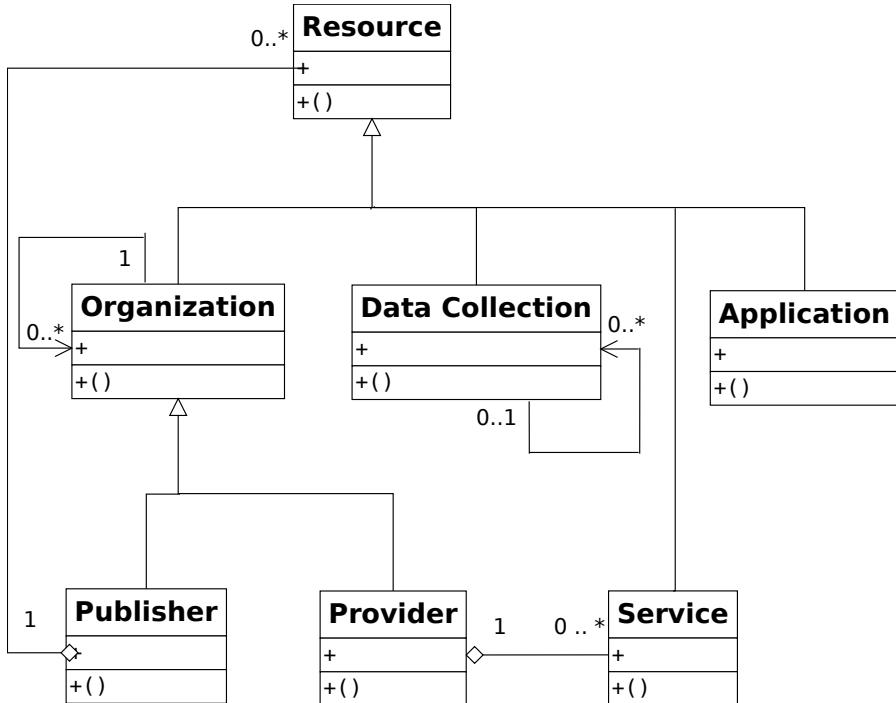


Figure 1.4: UML diagram of VOResource

Following example uses program stilts¹ to query registry with parameter shortName equal to 'AIASCR'². This returns VOTable (see section 1.6.1) containing meta-data about the resource.

```

1 stilts regquery query="shortName like 'AIASCR'"
2 regurl=http://registry.euro-vo.org/services/RegistrySearch
3 ofmt=votable-tabledata > resourceExample.vot
  
```

Rows 1–4 define XML and VOTable schema with adequate locations (xmlns³) followed by informations about the actual resource. The listing is abbreviated.

```

1 <?xml version='1.0'?>
2 <VOTABLE version="1.1"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns="http://www.ivoa.net/xml/VOTable/v1.1">
5 .
6 <DATA>
7 <TABLEDATA>
8   <TR>
9     <TD>ivo://asu.cas.cz</TD>
10    <TD>AIASCR</TD>
11    <TD>Astronomical Institute of the Academy of Sciences of the Czech
       Republic Naming Authority</TD>
  
```

¹STIL Tool Set. Set of command-line tools based on STIL, the Starlink Tables Infrastructure Library.

²Astronomical Institute of the Academy of Sciences of the Czech Republic

³XML namespaces. Provide uniquely named elements and attributes in an XML document.

```
12 <TD>http://stelweb.asu.cas.cz/web/index/index-en.php</TD>
13 <TD>Petr Skoda &lt;skoda@sunstel.asu.cas.cz&gt;</TD>
```

1.5 Data Access Protocols

Protocols are very important part of Virtual Observatory. Their understanding is key to comprehend the concepts behind VO. They allow to discover resource and obtain desirable data. All of them are based on existing web standards and are designed to be simple and therefore easy to implement on existing astronomical archives. The main idea is simple and universal: HTTP GET request with parameter is sent to the resource and structured document (VOTable) is sent back.

1.5.1 Cone Search Protocol

Cone Search was the first standard protocol of Virtual Observatory. It enables to retrieve records from an astronomical catalog. The input is the query which describes sky position and the radius on the sky. The output is a list of objects whose positions lie in the defined vicinity. The output is formatted as a VOTable. Service compliant with Cone Search Protocol is called Cone Search Service. Only the request and response is specified not the implementation or data storage.

The requirements are:

1. Respond to a HTTP GET request represented by a URL

```
1 http://<server-address>/<path>? \[<extra-GET-arg> & \[...\] \]"
```

The constraints are expressed as a list of ampersand-delimited GET arguments. For example:

```
1 http://simbad.u-strasbg.fr/simbad-conesearch.pl?RA=24.5&DEC=-57.2&SR=0.1
```

Where RA is right-ascension, DEC declination and SR the radius of the cone in the ICRS coordinate system in decimal degrees. These parameters are required others are optional.

2. Return an XML document in the VOTable format.

There are several requirements on the contents of the table:

- UCD fields "ID_MAIN", "POS_EQ_RA_MAIN", "POS_EQ_DEC_MAIN" must be present.
- Return VOTable with single PARAM element name="Error" in the case of error.

Cone Search is implemented in many software packages. Besides standard VO tools like TOPCAT or STILTS also in MUNIPACK and many others. Following example shows simple query to SIMBAD [Wenger et al., 2000] catalog using method *urlopen* of Python library *urllib2*.

```

1 import urllib2
2 response = urllib2.urlopen('http://simbad.u-strasbg.fr/simbad-conesearch.
    p1?RA=24.5&DEC=-57&SR=0.1')
3 print response.read()

```

The same result can be obtained using program like wget¹ or Web browser.

1.5.2 Simple Image Access Protocol

The key idea behind the SIA Protocol is to allow users and programs to retrieve images created by an image service on-the-fly. From technical point of view it is designed in a similar way as Cone Search Protocol (see section 1.5.1), specifically as name-value HTTP GET requests and the VOTable XML format output. The user specifies ideal image coverage (position and the size) he wants to receive and the image service produces a list of images it can return in the VOTable format. The user then could issue getImage request to retrieve desirable images.

There are following requirements for compliance. To be a SIA service To be a SIA service, it must support:

- Image Query web method,
- Image Retrieval (getImage) web method.

Furthermore the image service should be registered to be able to locate optimal service. There are several types of image services:

- Image Cutout Service.
Provides rectangular regions of large images.
- Image Mosaicing Service.
Size, scale and projection could be specified.
- Atlas Image Archive
Pre-computed atlas of images.
- Pointed Image Archive.
Images are not part of a sky survey but rather focused on specific source

To get a list of images query has to send via HTTP GET method. The first part is base URL. The second part are parameters specifying image properties such as position (POS), size (SIZE), etc.

```

1 http://<server-address>/<path>?[<extra GET arg>&[...]]

```

There are two examples of using SIA protocol to obtain image. First one from SDSS, second from Hubble Space Telescope archive.

```

1 http://skyview.gsfc.nasa.gov/cgi-bin/vo/sia.pl?SURVEY=SDSS&POS
    =18.87667,-0.86083&SIZE=1
2 http://hubblesite.org/cgi-bin/sia/hst_pr_sia.pl?POS=83.6,22.0&SIZE=1.0

```

¹program for non-interactive download of files from the Web

There is more complex example using Astrogrid framework to show how to discover SIA service and obtain an image. First registry method searchSiap is used to find SIA service for SDSS, this is then used in SiapSearch method to obtain result in VOTable format.

```

1 In [1]: from astrogrid import Registry, ConeSearch
2 In [2]: list = reg.searchSiap('SDSS')
3 In [3]: print [p['id'] for p in list]
4 -----> print([p['id'] for p in list])
5 ['ivo://nasa.heasarc/skyview/sdss']
6
7 In [4]: siap = SiapSearch('ivo://nasa.heasarc/skyview/sdss')
8 In [5]: result = siap.execute(18.8, -0.8, 1.0)

```

1.5.3 Simple Spectra Access Protocol

SSA Protocol allows to discover and obtain 1-D spectra from VO Service. It shares many similarities to the previously discussed SIA Protocol. It defines a uniform interface to remotely discover and access simple 1-D spectra.

The process to obtain a spectrum compose of following steps:

- Query the resource registry.
- Data discovery to selected service to get available resources in VOTable format.
- Download selected spectra using URL.

The spectra could be one of the following types:

- Pre-computed
- Computed on the fly

To be a SSA-compliant, the service must provide:

1. HTTP GET interface, returning the query response encoded as a VOTable document, with at least parameters POS, SIZE, TIME, BAND, and FORMAT.
2. GetData method returning data in at least one of the SSA-compliant data formats (VOTable, FITS)
3. FORMAT=METADATA metadata query feature

Following example show how to discover resources with SSA capability using STILTS program.

```

1 stilts regquery query="shortName like 'ESO' capability/@standardID =
2   'ivo://ivoa.net/std/SSA'" ocmd="keepcols 'ShortName accessUrl'"
3   ofmt=ascii

```

With information of service URL, one can specify a query to obtain a list with available spectra in VOTable format. This can be used in Web browser or via programs such *wget* or *curl*.

```

1 http://archive.eso.org/apps/ssaserver/EsoProxySsap?REQUEST=queryData&POS
   =83.63,22&SIZE=1

```

1.6 Data Formats

Astronomy has always been, by its nature, on vanguard of image producing and processing. This is especially true for the era of digitalization. The situation with data formats in astronomy is unique. There are just few very good standards with variety of implementation in many programming languages. Virtual Observatory takes advantage of this heritage and implement these formats in sensible way into its own standards.

1.6.1 VOTable

Motivation

VOTable is flexible storage and exchange format fundamentally interconnected with Virtual Observatory. It has features for big-data and Grid computing. Data can be stored in different ways in dependence on the character and size. Small tables can be stored in pure XML¹, while large-scale data can be referenced with the URL² syntax protocol://location. It combines web standards (it is based on XML) and astronomy tradition in storing data (it is FITS compatible). Expiration and authentication are also supported.

Structure

Following example of VOTable was created from SDSS FITS file used in this work. First there is an information about XML and VOTable versions and references to corresponding XML Schema³. TABLE tag encapsulating tabular data. FIELD tag describe identification (ID), type and precision of columns. DATA tag contains data (here) in TABLEDATA format (other types are FITS and BINARY)

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!-- Produced with vo.table version 0.6
3      http://www.stsci.edu/trac/ssb/astrolib
4      Author: Michael Droettboom <support@stsci.edu> -->
5  <VOTABLE version="1.0"
6      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7      xsi:noNamespaceSchemaLocation="http://www.ivoa.net/xml/VOTable/v1.0"
8      xmlns="http://www.ivoa.net/xml/VOTable/v1.0">
9      <RESOURCE type="results" >
10     <TABLE >
11         <FIELD ID="col0" name="wave" datatype="float" unit=""
12             precision="F9"/>
13         <DATA>
14             <TABLEDATA>
15                 <TR>
16                     <TD>4012.50757</TD>
```

¹Extensible Markup Language. W3C standard. Set of rules for encoding documents in machine-readable form

²Uniform Resource Locator. Uniform Resource Identifier (URI) that specifies where an identified resource is available and the mechanism for retrieving it.

³Define the legal building blocks of an XML document. Note: XML schema can be described by XML Schema, Document Type Definition (DTD) or RELAX NG

```

17   </TR>
18 </TABLEDATA>
19   </DATA>
20 </TABLE>
21 </RESOURCE>
22 </VOTABLE>
```

Examples

All examples were created using ATPy¹

Following example shows transformation FITS into VOTable.

```

1 In [1]: import atpy
2 In [2]: tbl = atpy.Table('spSpec-53401-2052-458.fit', hdu=1)
3 Auto-detected input type: fits
4 In [3]: tbl.write('votableExample.xml')
5 Auto-detected input type: vo
```

1.6.2 FITS

Motivation

“An archival format must be utterly portable and self-describing, on the assumption that, apart from the transcription device, neither the software nor the hardware that wrote the data will be available when the data are read.” [Council, 1995]

FITS (Flexible Image Transport System) was originally created for data exchange between WSRT² and the VLA³ [Schlesinger, 1997]. It is now used as a file format to store, transmit, and manipulate scientific data and it is (thanks to its revolutionary design) de facto standard in astronomy.

Structure

One file can contain several HDUs (Header Data Units). The first part of each HDU is the header, composed of ASCII card images containing keyword=value statements that describe the size, format, and structure of the data that follow.

- Primary header and data unit (HDU).
- Conforming Extensions (optional).
- Other special records (optional, restricted).

Standards and documents related to FITS are maintained by IAU-FWG⁴ and available at <http://fits.gsfc.nasa.gov>.

¹High-level Python package providing a way to manipulate tables of astronomical data in a uniform way.

²Westerbork Synthesis Radio Telescope

³Very Large Array

⁴International Astronomical Union FITS

Examples

There are many libraries for working with FITS files. The official list is available at http://fits.gsfc.nasa.gov/fits_libraries.html. PyFITS, library for Python programming language was used for following examples. PyFITS is a development project of the Science Software Branch at the Space Telescope Science Institute http://www.stsci.edu/resources/software_hardware/pyfits.

Reading FITS headers.

```
1 In [1]: import pyfits
2 In [2]: hdulist = pyfits.open('spSpec-53237-1886-248.fit')
3 In [3]: hdulist.info()
4
5 No.    Name        Type      Cards   Dimensions Format
6 0    PRIMARY    PrimaryHDU     213   (3874, 5)   float32
7 1          BinTableHDU     54    6R x 23C   [1E, 1E, ...
8 2          BinTableHDU     54   44R x 23C   [1E, 1E, ...
9 3          BinTableHDU     18    1R x 5C    [1E, 1E, ...
```

Printing primary HDU.

```
1 In [4]: print hdulist[0].header
2 -----> print(hdulist[0].header)
3 DATE-OBS= '2004-08-20'           / 1st row - TAI date
4 TAIHMS = '10:36:18.11'          / 1st row - TAI time (HH:MM:SS.SS) (TAI-UT =
5           appr
6 TAI-BEG =      4599713999.00 / Exposure Start Time
7 TAI-END =      4599717089.00 / Exposure End Time
8 MJD      =          53237 / MJD of observation
9 MJDLIST = '53237'           /
10 VERSION = 'v3_140_0'          / version of IOP
11 TELESCOP= 'SDSS 2.5-M'       / Sloan Digital Sky Survey
```

Updating FITS file.

```
1 In [1]: prihdr = hdulist[0].header
2 In [2]: prihdr.update('observer', 'Astar')
3 In [3]: prihdr.add_history('I updated this file 3/27/11')
```

1.7 Tools & Libraries

There are many programs and libraries allowing user to interact with VO services. Such application is called VO Enabled. Thanks to openness and standardization anyone can develop his own application or enable existing¹ application to interact with VO Services . Libraries are also available for many programming languages enabling advanced users to interact with VO from scripts and programs. Such diversity is healthy and probably the only possible way to ensure natural evolution of Virtual Observatory.

¹For example Astroweka or Mirage.

Chapter 2

Data Mining

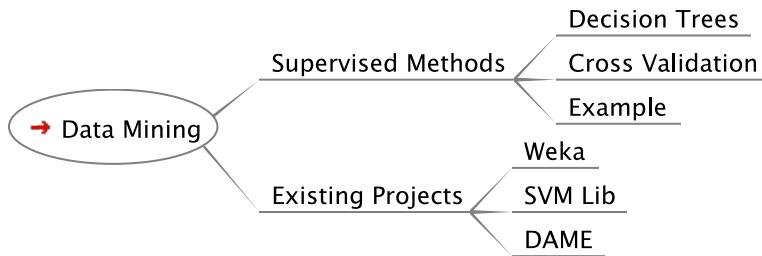


Figure 2.1: Chapter structure

Virtual Observatory may be seen as data infrastructure. It enables astronomers to get data more easily in a uniform way. But there is another and even bigger problem now. How to deal with huge amount of data? Can we change the problem to opportunity? Can we discover new phenomena, new types of objects or exploit natural groups in the data? Data Mining and related techniques are created exactly for such purposes. Used correctly, it can be powerful approach, promising scientific advance. On the other side this field is very complex with dozens of different methods and algorithms. This forms needs and opportunity for interdisciplinary cooperation with Data Mining experts. This can be very beneficially for both fields, providing astronomers with interesting methods for data analysis and computer scientist with large amount of quality data.

2.1 Supervised Methods

These methods are also known as predictive[Ball et al., 2010]. They rely on training set with known target property. This set must be representative. The selected method is trained on that set and the result is then used on data for which the target property is not known. Among supervised method are classification, regression, anomaly detection and others.

2.1.1 Decision Tree (DT)

Is an example of supervised classification. Based on final number of data $(x^{(1)}, \dots, x^{(p)})$ with known class C_1, \dots, C_m classifier is created, i.e. image f classifying any

$x \in \mathcal{X}, f : \mathcal{X} \rightarrow \mathcal{Y}$, where \mathcal{X} is a set of possible input vectors and \mathcal{Y} is a set which values represent classes C_1, \dots, C_m (for example $\mathcal{Y} = 1, \dots, m$). The model is constructed based on training set as a tree structure, where leaves represent classifications and branches conjunctions of features that lead to those classifications. The main advantages of DT are:

- Simple to understand and interpret.
- Able to handle both numerical and categorical data.
- Uses a white box model.
- Perform well with large data in a short time.

In pseudo-code, the general algorithm for building decision trees is [Kotsiantis et al., 2007]:

1. Check for base cases
2. For each attribute A
 - Find the normalized information gain from splitting on A
3. Let "A best" be the attribute with the highest normalized information gain
4. Create a decision node that splits on "A best"
5. Recur on the sublists obtained by splitting on "A best", and add those nodes as children of node

Furthermore algorithms C4.5 is described for several reasons: Its code is available and free implementations exist (J48 in Weka), is de-facto standard in classification using DT, is used in practical part of this work. The key question of DT algorithm is how to choose attribute for splitting the tree. C4.5 Uses measures based on information entropy:

$$H = - \sum_{i=1}^T p_t \log_2 p_t, \quad (2.1)$$

where $p(x_i)$ is probability of occurrence of class t and T is the number of classes [Berka, 2003].

After the tree is created it is optimized by pruning, which prevents over-fitting.

2.1.1.1 Cross-validation

The quality of the training set is crucial to good results. The amount of data for testing is always limited. In general, one cannot be sure whether a sample is representative. If for example certain group is missing, one could not expect a classifier learned from such data to perform well on the examples s of that class. One of the technique used here is cross-validation.

The data is divided into fixed number of partitions (folds) and each in turn is used for testing and the reminder is used for training. Finally, the number of partitions error estimates are averaged to yield an overall error. The standard is to used 10-fold cross-validation. This number is a result of tests on numerous data sets [Witten and Frank, 2005]

2.1.1.2 Example: Classifying Galaxies Stars and QSO using color indices

There is an example of classifying Galaxies Stars and QSO based on photometric properties using Decision Tree algorithm J48 (C4.5 in Weka). The data come from SDSS (Sloan Digital Sky Survey) DR7. 298 Objects were used (100 Stars, 99 Galaxies, 99 QSO). SDSS Filters u,g,r,i were used as parameters. Data were obtained using SQL query from SDSS CAS.

Filter	Effective wavelength [Å]
Ultraviolet (u)	3543
Green (g)	4770
Red (r)	6231
Near Infrared (i)	7625
Infrared (z)	9134

Table 2.1: SDSS photometric system

```

1 SELECT TOP 100 u-g,g-r,r-i,s.specClass
2 FROM PhotoPrimary p join SpecPhotoAll s on p.objid=s.objid
3 WHERE s.specClass in (1)
4 AND u between 18 and 19
5 UNION all
6 SELECT top 100 u-g,g-r,r-i,s.specClass
7 FROM PhotoPrimary p join SpecPhotoAll s on p.objid=s.objid
8 WHERE s.specClass in (2)
9 AND u between 18 and 19
10 UNION all
11 SELECT top 100 u-g,g-r,r-i,s.specClass
12 FROM PhotoPrimary p join SpecPhotoAll s on p.objid=s.objid
13 WHERE s.specClass in (3)
14 AND u between 18 and 19

```

The following listing shows the result of classification. The classifier was able to distinguish 95% of the processed objects.

1	Correctly Classified Instances	96	95.0495 %
2	Incorrectly Classified Instances	5	4.9505 %
3	Kappa statistic	0.9257	
4	Mean absolute error	0.0669	
5	Root mean squared error	0.1778	
6	Relative absolute error	15.0587 %	
7	Root relative squared error	37.6973 %	
8	Total Number of Instances	101	

The big advantage of Decision Trees over black box algorithms (such as Neural Network) is that one could understand the classification process. The decision tree generated for this example is following:

```

1 ug <= 0.663668
2 | gr <= -0.191208: 1 (7.0)
3 | gr > -0.191208: 3 (104.0/5.0)
4 ug > 0.663668

```

```

5 |   ri <= 0.285854: 1 (88.0/5.0)
6 |   ri > 0.285854
7 |   |   ri <= 0.314657
8 |   |   |   gr <= 0.692108: 2 (6.0)
9 |   |   |   gr > 0.692108: 1 (3.0)
10|   |   ri > 0.314657: 2 (90.0/2.0)

```

Useful tool for understanding how classifier was successful on individual classes is the confusion matrix. Columns show how the object was classified and the row what is his actual class. In this example QSO were classified correctly in 100% cases. Distinction between stars and galaxies are a bit worse and the algorithm classify 2 galaxies incorrectly as stars and two stars were confused with galaxies. One stars was incorrectly classified as QSO.

```

1 s g q <-- classified as
2 30 2 1 | s
3 2 33 0 | g
4 0 0 33 | q

```

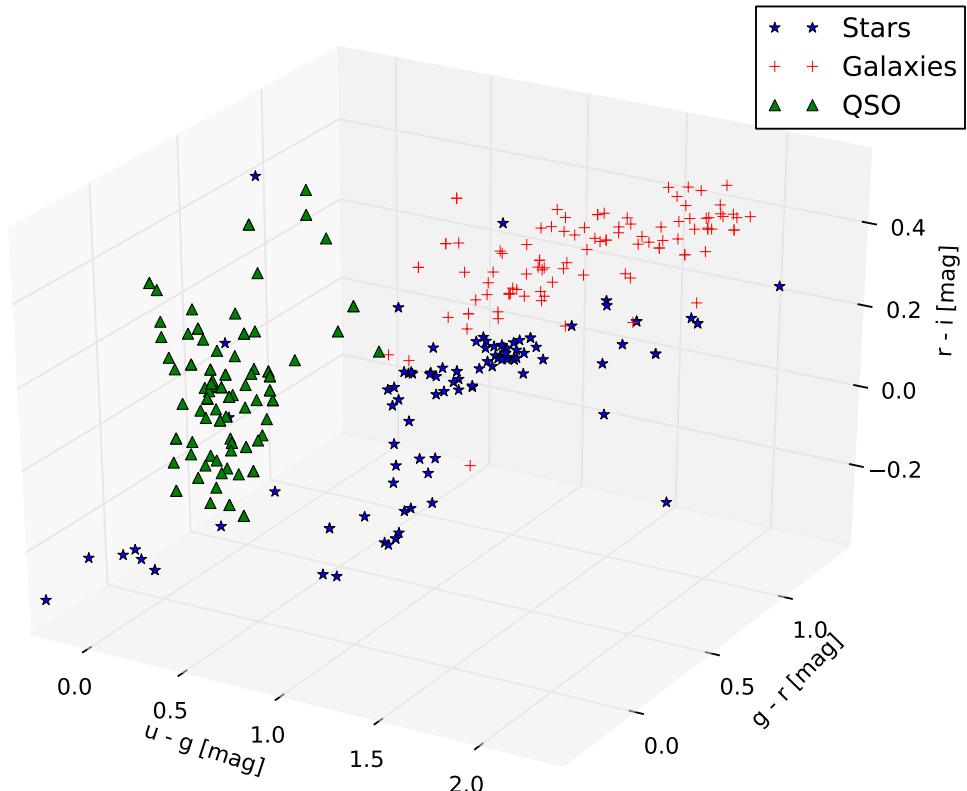


Figure 2.2: Color Diagram of the problem. It shows that individual object classes occupies different regions in the diagram

2.2 Data Mining tools

There are many open source projects related to Machine Learning and Data Mining. I would like to mention those which were used during experiments related to this work.

2.2.1 Weka

Weka is a collection of machine learning algorithms developed at University of Waikato, New Zealand. It includes functions for preprocessing, clustering, classification, regression, visualization, and feature selection. Originally designed as a tool for analyzing data from agricultural domains become extremely popular in data mining community because of its quality, openness, perfect documentation, and multi-platform implementation. Weka can be obtained at <http://www.cs.waikato.ac.nz/~ml/weka/>

2.2.2 SVM lib

Is library implementing Support Vector Machine with following properties

- Different SVM formulations
- Efficient multi-class classification
- Cross validation for model selection
- Probability estimates
- Various kernels (including precomputed kernel matrix)
- Weighted SVM for unbalanced data
- Both C++ and Java sources
- GUI demonstrating SVM classification and regression
- Python, R, MATLAB, Perl, Ruby, Weka, Common LISP, CLISP, Haskell, and LabVIEW, interfaces. C# .NET code and CUDA extension is available. It's also included in some data mining environments: RapidMiner and PCP.
- Automatic model selection which can generate contour of cross validation accuracy.

The project is hosted on <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>. The part of this project is useful and well written practical guide to SVM classification.

2.2.3 DAME

Is great example of full understanding of the paradigm shift in astronomy. The project implements Neural Networks and Support Vector Machines algorithms and it is VO-compatible. The documentation includes scientific use cases, masters and PhD thesis and lectures. As it is typical in these projects it exceeds its original domain of astronomical data into general platform. The projects is available at <http://dame.dsfa.unina.it/>

Chapter 3

Be candidates

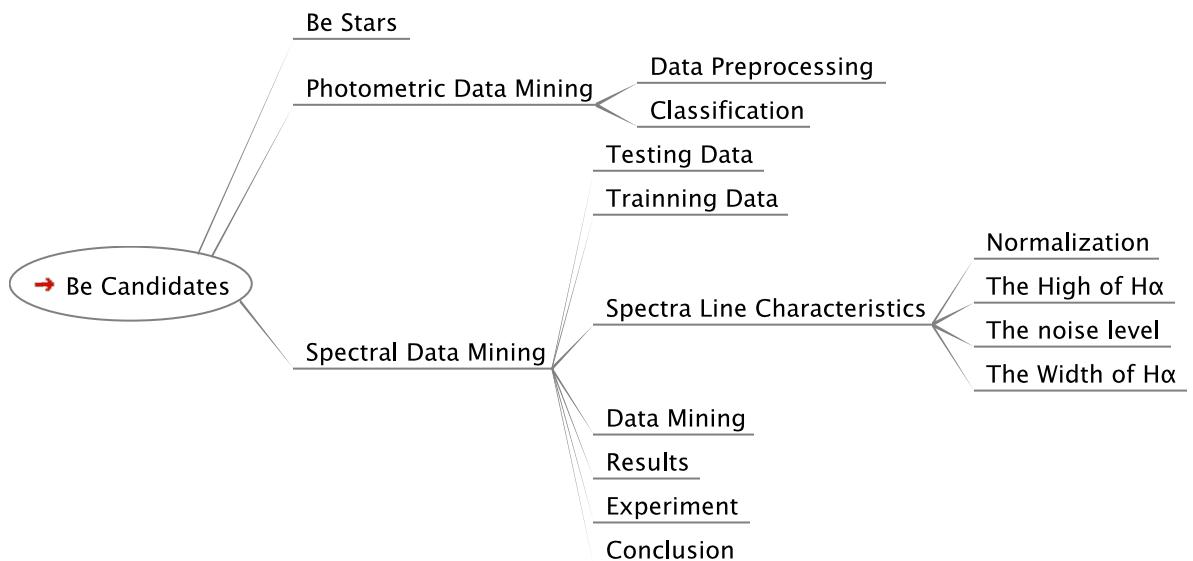


Figure 3.1: Chapter structure

Astronomical objects used in this work to demonstrate some of the discussed technologies and methods were Be stars. The goal was to develop a process of finding new candidates in the available data. Several approaches were considered and two of them are discussed in the rest of this text. First one utilizes photometric properties of Be stars, second uses spectra characteristics.

3.1 Be stars

The first example of Be star was reported by Padre Angelo Secchi in his letter to the Astronomische Nachrichten in 1866.

Classical Be stars are non-supergiant B-type stars whose spectrum has or had at some time, one or more Balmer lines in emission. The current accepted explanation of this phenomena is circumstellar gaseous component in the form of equatorial disk. Rapidly rotating central star is important feature of these objects, which may be important contributor of the circumstellar medium [Porter and Rivinius, 2003].

The important characteristic used later in the work is the H α emission. The explanation of its origin is on following pictures.

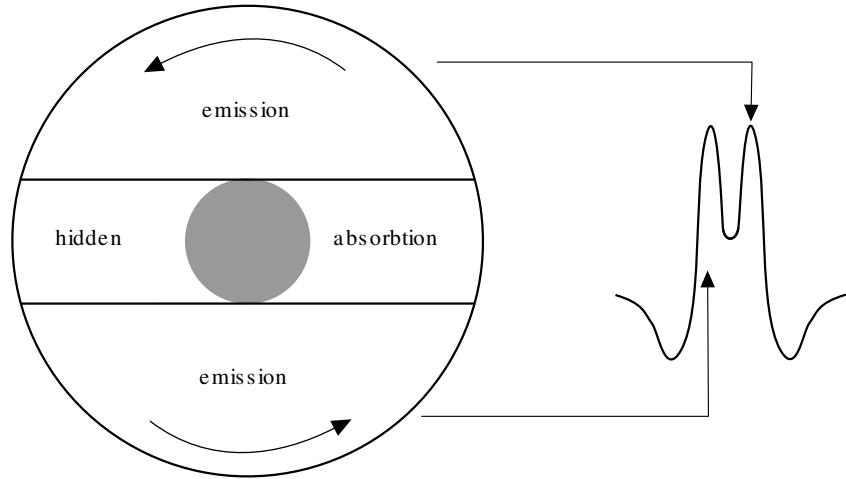


Figure 3.2: Model of a typical Be star. Emission lines coming from an equatorial disk is added to the photo-spheric absorption spectrum. Central B star emits UV (Lyman continuum) and ionizes the disk, which in turn re-emits at high wavelength such as visible domain. [Hirata and Kogure, 1984]

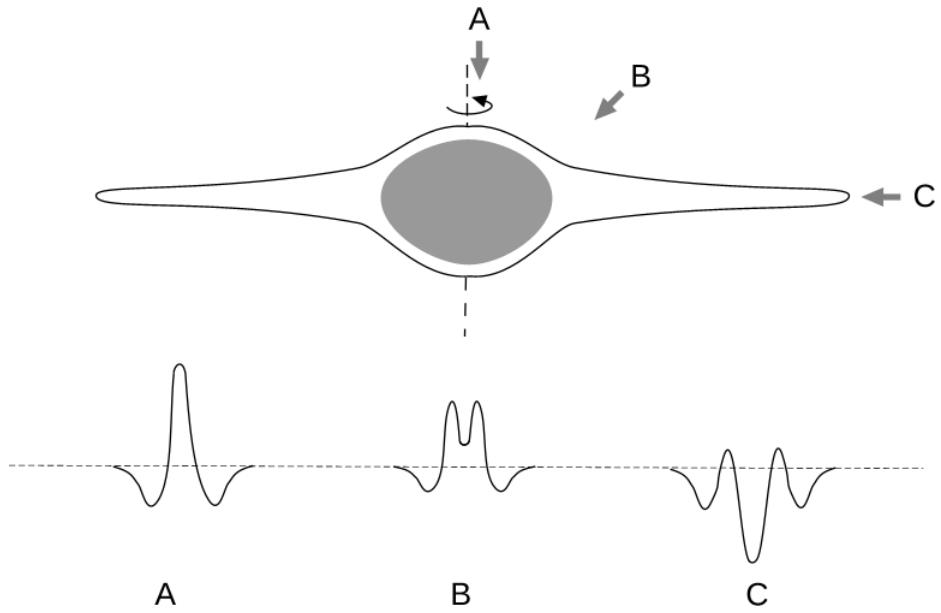


Figure 3.3: Example of spectra of Be stars based on view angle [Slettebak, 1988]

There are still many open questions related to their rotation, evolutionary status, presence and origin of the magnetic fields, mass and angular momentum transfer and others, therefore the process for automatic discoveries of Be phenomena in the digitalized surveys and obtaining new candidates could help answer these questions.

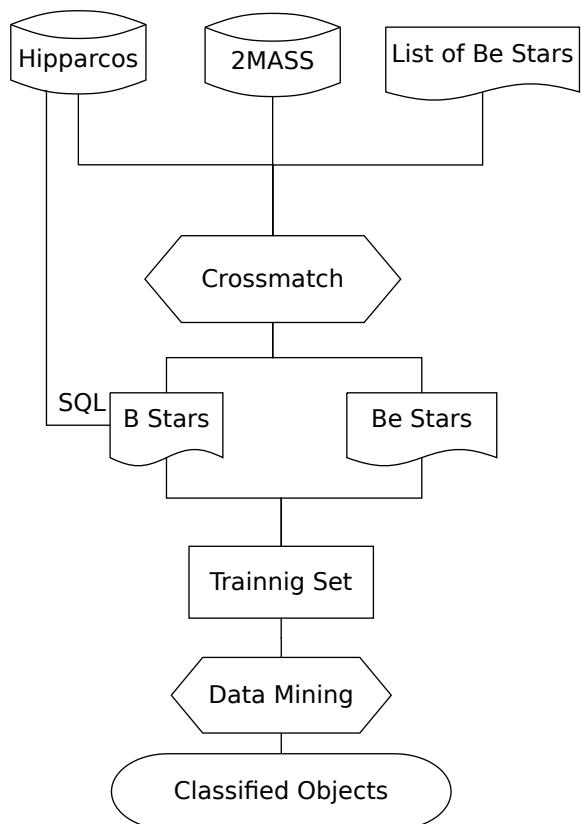


Figure 3.4: Schematic diagram of the photometric Data Mining process. The lists of confirmed Be stars consisted of Hipparcos IDs, this was correlated with Hipparcos catalog to obtain right ascension and declination of the objects and subsequently cross-matched with 2MASS catalog to get photometric data. The second set of B stars were acquired in similar manner but using SQL the condition was set to get B type stars different from the list of Be stars.

3.2 Photometric Data Mining

Classification based on photometric properties is very attractive from several points of view. There are much more available photometric than spectral data and they are easier accessible. Because they are easier to gain the disproportion between photometric and spectral data will probably increase in the future as well. The distinction between Be and other types of stars also should be theoretically possible since the Be stars exhibits infrared excess correlated to the H α emission [Van Kerkwijk et al., 1995].

3.2.1 Data preprocessing

I was provided with a list of confirmed Be stars from Academy of Science Ondejov. This list consist of 625 manually chosen objects. Data were correlated with Hipparcos [Perryman et al., 1997] catalog to obtain RA, DEC and then with 2MASS [Skrutskie et al., 2006] catalog to obtain J,H,K Colors using method of multi-cone search in Virtual Observatory. The second set was acquired from Hipparcos catalog using following SQL query:

```
1 Select *
2 From maincat as m, hipva1 as h
3 Where (m.HIP=h.HIP )
4 And h.SpType Like 'B%'
```

The result was cross-correlated with 2MASS catalog to obtain the same colors as for the confirmed Be stars. Color diagram of this two sets are on the figure 3.5

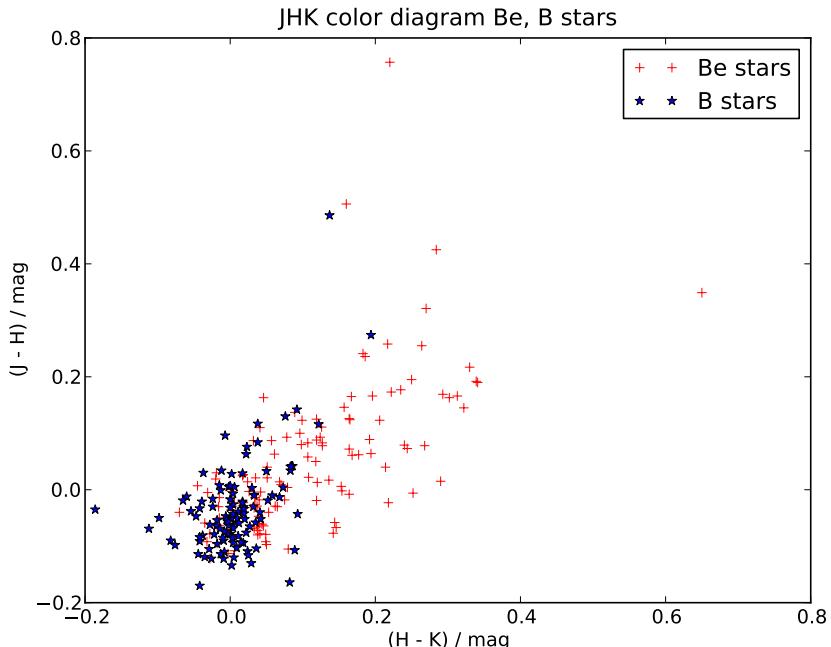


Figure 3.5: Color diagram of confirmed Be stars Vs B stars

The uncertainties were computed for each object using propagation of error. These errors are depicted on the figure 3.6. Although the uncertainties are significant certain trends are presented.

$$\delta_{(m_j - m_h)} = \sqrt{\left(\frac{\partial(m_j - m_h)}{\partial m_j}\right)^2 \delta_{m_j}^2 + \left(\frac{\partial(m_j - m_h)}{\partial m_h}\right)^2 \delta_{m_h}^2}$$

$$\frac{\partial(m_j - m_h)}{\partial m_j} = 1, \frac{\partial(m_j - m_h)}{\partial m_h} = -1$$

$$\delta_{(m_j - m_h)} = \sqrt{\delta_{m_j}^2 + \delta_{m_h}^2}$$

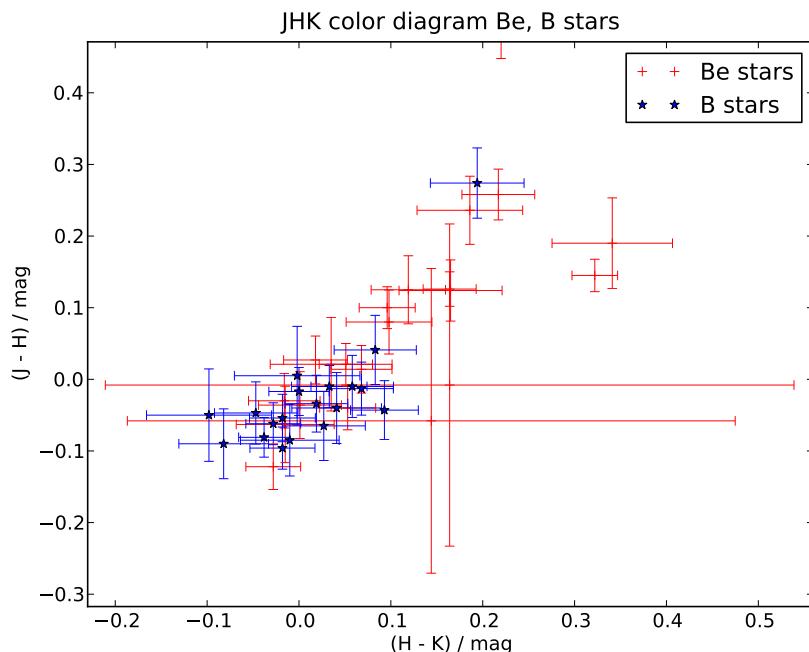


Figure 3.6: Color diagram of confirmed Be stars, B stars with errors

3.2.2 Classification

Data were transformed from original VOTable obtained from Virtual Observatory tools to arff¹format used in Weka Data Mining system. Algorithm C4.5 (J48) was used to perform actual classification with following result:

1	Correctly Classified Instances	769	73.0989 %
2	Incorrectly Classified Instances	283	26.9011 %
3	Kappa statistic	0.4496	
4	Mean absolute error	0.3843	
5	Root mean squared error	0.4383	
6	Relative absolute error	79.4985 %	

¹Attribute-Relation File Format. Developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato.

7	Root relative squared error	89.1648 %
8	Total Number of Instances	1052

As seen on the first row 73 % from 1052 objects were classified correctly. More details can be obtained from confusion matrix below.

1	B	Be	<-- classified as
2	304	126	B
3	157	465	Be

304 of B and 456 of Be stars were classified correctly but 126 of B and 157 of Be stars were classified incorrectly. In virtue of these results one should be sceptical if the distinction based only on photometric properties is significant enough to find relevant new candidates of Be stars. For this reason more sophisticated (and much more complicated) approach using spectra analysis was tested.

3.3 Spectral Data Mining

Spectra provide much wider scientific informations over photometric properties. Spectral lines exhibits many distinguish features and astronomers have long tradition of analysing their properties. On the other side its much complicated to handle them because of different characteristics (resolution, calibration, wavelength range, etc.). This is especially true for massive automated processing.

3.3.1 Testing Data

As testing sample the project SEGUE of SDSS were selected. This contains 178315 spectra in DR7. Following SQL query was used to generate the list of URL links for individual FITS files. These files were then downloaded to local sever using wget command.

```

1 SELECT objid, dbo.fGetUrlFitsSpectrum(s.specObjID)
2 INTO mydb.segue_1
3 FROM SpecPhotoAll s, plateX p
4 WHERE s.specObjID is not null
5 AND s.plateid = p.plateid
6 AND p.programname LIKE 'segue%'
7 AND specClass = 1

```

3.3.2 Training Data

The spectra from Ondejov Observatory were used as a training sample. Files were downloaded using SSA protocol. The SSA server is not publicly available, therefore SSH tunneling was used. Two scripts for this process were created. First to construct the list of SSA compliant addresses, the second to analyse acquired response in VOTable format. Then the spectra were downloaded using wget command. The function for constructing the links based on list of the RA, DEC which were obtained from Hipparcos catalog using the specification of IDs from Ondejov's index.

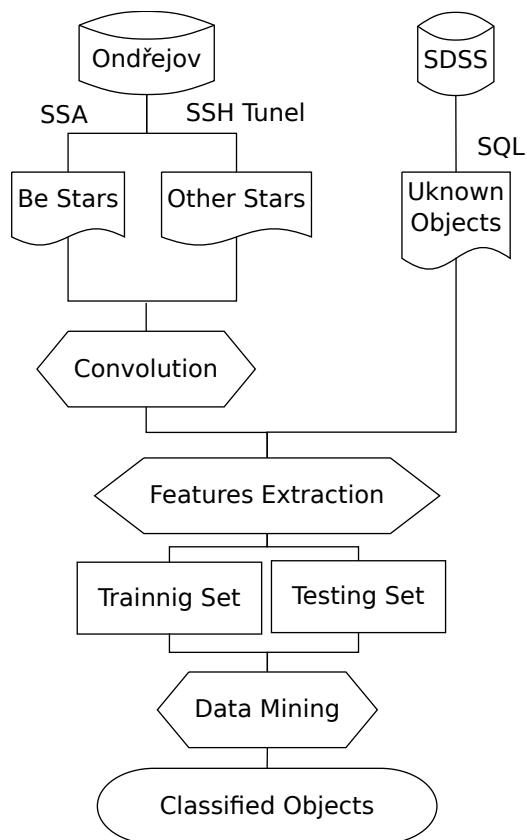


Figure 3.7: Schematic diagram of the spectral Data Mining process. Using SSA protocol the spectra from Ondejov server was acquired based on the list from photometric study. Convolution with profile of SDSS slit had to be performed to ensure compatibility with SDSS. Afterwards desired features were extracted automatically from the spectra after the continuum and H α line were fitted by appropriate functions. The same was done for spectra from SDSS except the convolution process.

```

1 def createQuery(data):
2     """ From raw data construct ra, dec """
3     """ Convert to degrees """
4     for line in data:
5         ra = ac.AngularCoordinate(line[0:10]).degrees # convert ra to
6             degrees
7         dec = ac.AngularCoordinate(line[-13:-1]).degrees # convert dec to
8             degrees
9         ra = line[0]
10        dec = line[1]
11        ssaTemp = 'http://tvoserver/coude/coude.cgi?c=ssac&n=coude_ssa&
12            REQUEST=queryData&POS=<ra>,<dec>&SIZE=1'
13        ssaTemp = ssaTemp.replace('<ra>', "%0.3f" % ra)
14        ssaTemp = ssaTemp.replace('<dec>', "%0.3f" % dec)
15        ssa.append(ssaTemp)
16    return ssa

```

The script generate the following output. The same process were used later for obtaining the sample of non Be stars.

```

1 http://tvoserver/coude/..._ssa&REQUEST=queryData&POS=83.113,-65.582&SIZE
2 =60
3 http://tvoserver/coude/..._ssa&REQUEST=queryData&POS=162.537,148.333&SIZE
4 =60
5 http://tvoserver/coude/..._ssa&REQUEST=queryData&POS=19.907,-73.502&SIZE
6 =60

```

3.3.2.1 Spectra Reduction

Because spectra from SDSS and Ondejov Observatory had different resolution, reduction was needed. Wavelength for spectra from Ondejov Observatory is computed using formula:

$$\lambda = \lambda_0 + x\Delta\lambda_{OND} \quad (3.1)$$

and for SDSS spectra by equation:

$$\lambda = 10^{\lambda_0 + x\Delta\lambda_{SDSS}}, \quad (3.2)$$

where λ_0 is value of parameter *CRVAL1* and $\Delta\lambda$ of *CD1_1*. Obtaining them from FITS file is easy using PyFITS¹ module:

```

1 In [1] import pyfits as pf
2 In [2]: hdu = pf.open('sdss_test.fit')
3 In [3]: hdu[0].header['CD1_1']
4 Out[3]: 0.0001 # SDSS spectrum
5 Out[3]: 0.2567 # Ondejov spectrum

```

¹PyFITS is a product of the Space Telescope Science Institute, which is operated by AURA for NASA.

From equations 3.1 and 3.2 the ration was computed:

$$\frac{\Delta\lambda_{SDSS}}{\Delta\lambda_{OND}} \doteq 3.87 \quad (3.3)$$

Based on this computation 4 pixels of Ondejov's spectra were reduced into one. There is the critical part of the reduction program:

```

1 def convolution(f, g):
2     """ Convolve two functions"""
3     fg = np.convolve(g,f,'same')
4     return fg
5 def reduce(x,y,bin):
6     """ Reduce bin pixel into 1"""
7     size = x.size/bin
8     l = 0
9     xx = x[:x.size-1:bin]
10    yy = list()
11    for i in range(0,size):
12        s = 0
13        for j in range(0,bin):
14            s = s + y[l]
15            l+=1
16        yy.append(s/bin)
17    return xx, yy

```

Prior to binning pixels convolution with function representing the slit in SDSS spectrograph was performed. As good approximation Gaussian function was used for representing the slit:

$$f(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.4)$$

Convolution is defined:

$$(f * g)(\lambda) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(\lambda - \tau) d\tau \quad (3.5)$$

Here it was used in it's discrete form:

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m] g[n-m] \quad (3.6)$$

The figure 3.8 shows the result.

3.3.3 Spectra Lines Characteristics

As parameters for Data Mining process characteristic values of H α line were extracted from the spectra. Many possible characteristics from fitting functions through Wavelets Coefficients and Eigenvalues Values were discussed with experts. Three parameters were finally selected. The hight and the width of the H α emission line and median absolute deviation as a characterization of the noise level in the spectrum.

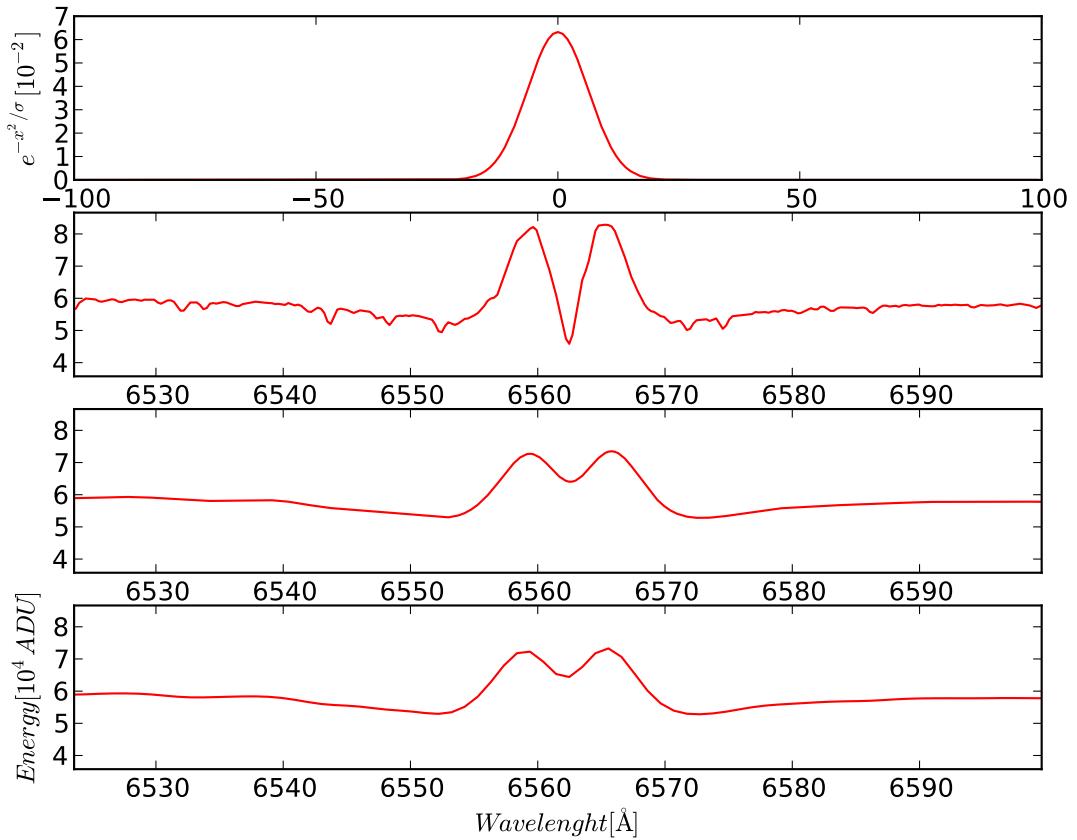


Figure 3.8: Reduction of Ondejov's spectra of the Be star 4 Hercules. The top figure shows Gaussian function used for convolution with the spectrum, followed by the original spectrum then there is a spectrum after convolution with the Gaussian function. The last is the final spectrum after reduction.

3.3.3.1 Normalization

Spectra from SDSS are normalized but the spectra from Ondejov are not. The spectra were divided by it's continuum fit function. This process ensures the compatibility when comparing different spectra. Function polyfit from numpy package was used to perform the fit. The solution minimizes the squared error:

$$\frac{\partial}{\partial q_j} \sum_{i=1}^n [y_i - f(q_j | x_i)]^2 = 0, \quad j = 0, 1 \quad (3.7)$$

where

$$f(x) = q_1 x + q_0 \quad (3.8)$$

3.3.3.2 The hight of the H α line

The maximum value in the region of 50 Å around H α were extracted from the spectrum.

```

1 def getMax(x,y,line,range):
2     """ Return maximum value of range in the spectrum"""
3     xrange = x[(x < line + range) & (x > line - range)]
4     yrange = y[(x < line + range) & (x > line - range)] - 1
5     maximum = yrange.max()
6     minimum = yrange.min()
7     if abs(maximum) > abs(minimum):
8         extrem = maximum
9     else:
10        extrem = minimum
11    return xrange, extrem, sgn

```

3.3.3.3 The noise level of the spectrum

The noise in the spectrum contributes to the characteristics of the spectral lines. As an estimator of the noise level the median absolute deviation was used. It is defined as:

$$\text{mad} = \text{median}_i(|X_i - \text{median}_j(X_j)|) \quad (3.9)$$

3.3.3.4 The width of the H α line

The Gaussian function:

$$f(x) = 1 + e^{-\frac{(x-x_0)^2}{S^2}} \quad (3.10)$$

was fitted to the profile of H α spectral line. First the robust estimators were computed and used as input parameters for leastsq¹ method from scipy.optimize module, which minimize the sum of squares.

¹”leastsq” is a wrapper around MINPACK’s lmdif and lmder algorithms.

$$x_0 = \frac{\text{median}(w_j x_j)}{\sum w_i}, \quad (3.11)$$

$$S = \frac{\text{mad}(x_i - x_0)}{\sum w_i}. \quad (3.12)$$

[Launer, 1979]

Part of the script implementing fitting the Gaussian function

```

1 x0 = np.median(sum(w*x))/sum(w)
2 S = sum(w*mad((x - x0)))/sum(w)
3 params = np.array([1, maximum, x0, S], dtype=float)
4 fit, flag = opt.leastsq(residuals, params, args=(yrange, xrange))
5 gauss = model(xrange, fit) + 1
6
7 def model(t, coeffs):
8     return coeffs[0] + coeffs[1] * np.exp( - ((t-coeffs[2])/coeffs[3])**2
9     )
10 def residuals(coeffs, y, t):
11     return y - model(t, coeffs)

```

The final result is on the figure 3.9 and 3.10. The script was adjust to work with SDSS and Ondejov's spectra. The whole procedure was performed on all of the cca 200 000 SDSS spectra and few dozens Ondejov's spectra resulting with the ASCII files with the characteristic values used later in Data Mining process.

3.3.4 Data Mining

Classification was performed using Weka software with algorithm J48 described in the chapter 2. Training set had 173 and testing set 178314 items. The excerpt from these files follows.

```

1 @RELATION STAR-B-BE
2 @ATTRIBUTE name STRING
3 @ATTRIBUTE alpha NUMERIC
4 @ATTRIBUTE grp {be,o}
5 @DATA
6 10_cas,-0.822196556626,be
7 11_cyg,1.68689566629,be

```

```

1 @RELATION STAR-B-BE
2 @ATTRIBUTE name STRING
3 @ATTRIBUTE alpha NUMERIC
4 @ATTRIBUTE grp {be,o}
5 @DATA
6 spSpec-53228-1884-001 -0.584628294569 ?
7 spSpec-53228-1884-002 -0.877184482566 ?

```

The attribute grp is known for the training set but unknown for testing set. The classification process fills this information based on decision tree created during learning phase. To automate the process command line version of Weka software was used.

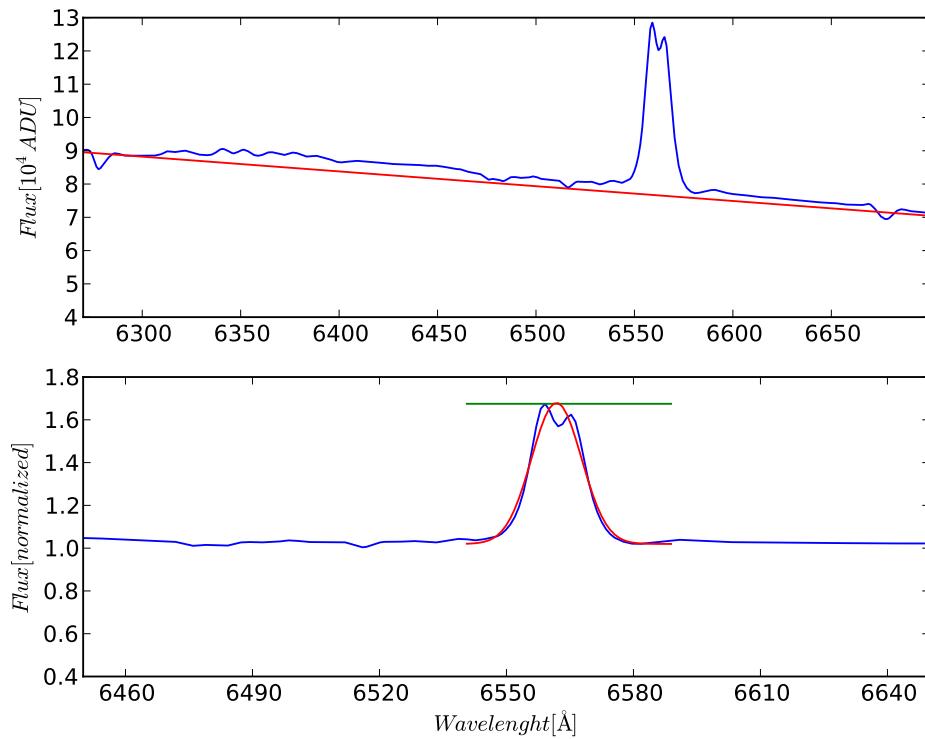


Figure 3.9: Normalized spectrum of Be star 60 Cyg. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line correspond to the maximum value in the region of 50 Å. The Gaussian fit is in red. Although the fit is almost perfect, this approach fails to get characteristic "double peak" of the emission line.

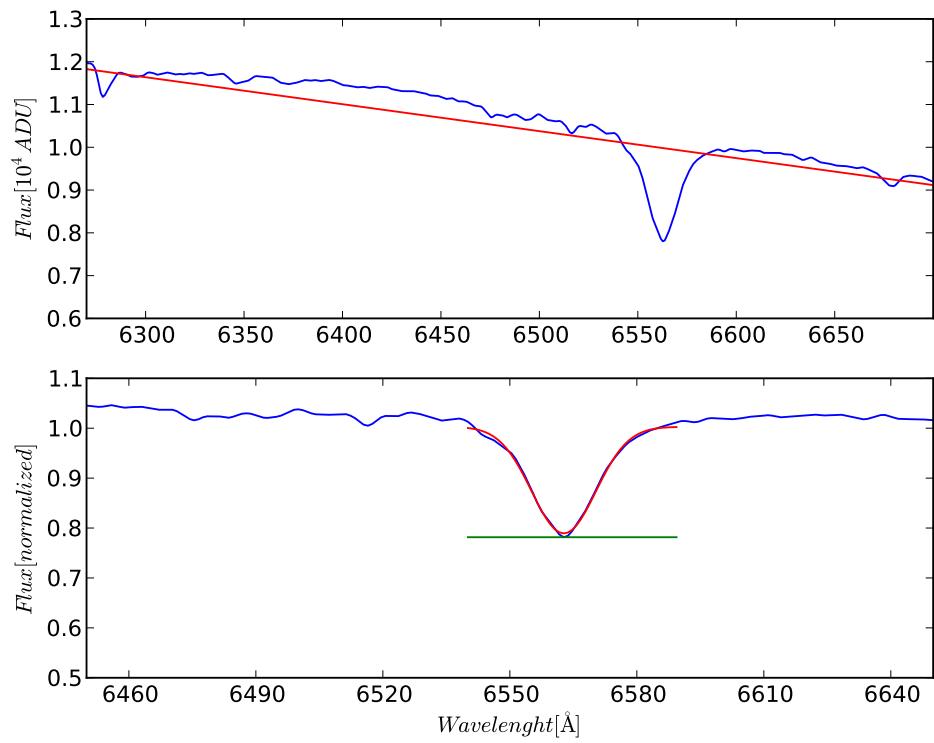


Figure 3.10: Normalized spectrum of Be star HR 8682. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line correspond to the maximum value in the region of 50 Å. The Gaussian fit is in red.

```

1  java -classpath weka.jar
2  weka.classifiers.meta.FilteredClassifier -F
3  weka.filters.unsupervised.attribute.RemoveType -W
4  weka.classifiers.trees.J48 -t $1 -T $2 -p 1

```

3.3.5 Results

The overall fruitfulness of the classification process is almost 84%. 10 folds cross-validation was used to compute the error rate.

```

1  === Summary ===
2  Correctly Classified Instances      145          83.815 %
3  Incorrectly Classified Instances   28           16.185 %
4  Kappa statistic                   0.6529
5  Mean absolute error              0.1849
6  Root mean squared error          0.3652
7  Relative absolute error          39.8819 %
8  Root relative squared error     75.8919 %
9  Total Number of Instances        173

```

The classification tree is relatively complicated. But still we can learn few things. It is using all of the parameters putted in so they are choose correctly (if they were irrelevant classifier would not used them). The most important parameter was max which determines the hight of the line above the continuum. This was expected as H α emission is dominated feature of Be stars. The second important parameter was the noise of the spectrum expressed in parameter mad. The less important (at least in this example) was the width of the line. It need to be emphasized that the parameter max does not really measure the hight, mad the noise or width the width of the line but there are some simplified (and certainly buggy) versions of real parameters. Though it can give us some physical insight of the studied phenomenas. Decision trees are therefore very powerful compared to black box approaches such are Neural Networks where the classification process is beyond human understanding.

```

1  J48 pruned tree
2  -----
3  max <= -0.18843
4  |  max <= -0.324763: o (46.0/5.0)
5  |  max > -0.324763
6  |  |  max <= -0.255475
7  |  |  |  mad <= 0.004133: o (2.0)
8  |  |  |  mad > 0.004133: be (13.0/1.0)
9  |  |  max > -0.255475
10 |  |  |  mad <= 0.009862: o (10.0)
11 |  |  |  mad > 0.009862
12 |  |  |  |  width <= 7.621593: o (3.0/1.0)
13 |  |  |  |  width > 7.621593: be (2.0)
14 max > -0.18843
15 |  mad <= 0.030316
16 |  |  max <= -0.091726
17 |  |  |  width <= 5.286489

```

```

18 |   |   |   | max <= -0.170022: be (2.0)
19 |   |   |   | max > -0.170022: o (3.0)
20 |   |   | width > 5.286489: be (9.0)
21 |   |   max > -0.091726: be (76.0)
22 |   mad > 0.030316
23 |   |   max <= 6.917615: o (4.0)
24 |   |   max > 6.917615: be (3.0)

```

```

1 === Confusion Matrix ===
2 Be Others <-- classified as
3 95 15 | Be
4 13 50 | Others

```

The Confusion Matrix evince that the classifier is more successful in assigning Be stars (95/15) than in the case of others types stars where 13/50 were associated with wrong class.

Spectra of some of the objects classified as Be stars are presented here. These samples represent tiny fraction of complete result. The program for generating web pages with thumbnails were created and full result is available on the Wiki pages of this project.

#	SDSS name	RA	DEC	u	g	r	i
1	SDSS J035747.16-063850.7	59.44	-6.64	19.83	19.99	19.73	19.86
2	SDSS J094325.89+520128.6	145.86	52.02	16.57	16.42	16.55	16.70
3	SDSS J120729.12+003659.8	181.87	0.62	17.57	15.28	14.30	13.96
4	SDSS J120908.18+194035.8	182.3	19.7	17.87	16.26	15.52	15.19

Table 3.1: Examples of the result

#	link
1	http://cas.sdss.org/dr7/en/tools/explore/obj.asp?sid=583165493179842560
2	http://cas.sdss.org/dr7/en/tools/explore/obj.asp?sid=671267254834298880
3	http://cas.sdss.org/dr7/en/tools/explore/obj.asp?sid=814259934286839808
4	http://cas.sdss.org/dr7/en/tools/explore/obj.asp?sid=814541407275450368

Table 3.2: Links to objects on SDSS Skyserver

For comparison there are spectra of know Be stars. It is clear that the profile of the H α line is complex and just one parameter cannot possibly express it's characteristic. More advanced description such as Wavelets coefficients or theoretical models of the line is needed if we want to create reliable process for identifying Be stars.

3.3.6 Experiment

One could be interested what would happened if we have choose different parameters, used other algorithm, different training set etc. These are perfectly valid questions and it is actually the purpose and essence of Data Mining and computers in general:

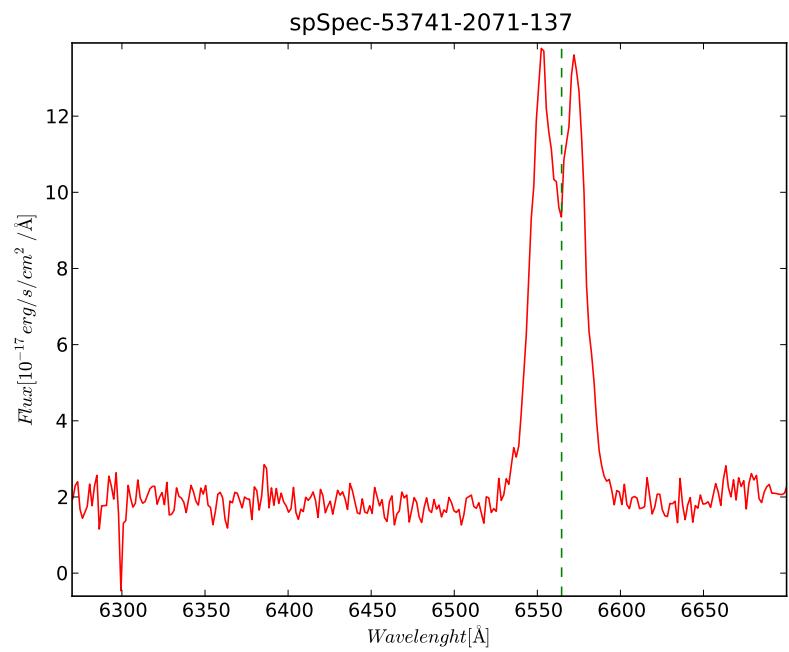


Figure 3.11: Example 1: SDSS J035747.16-063850.7

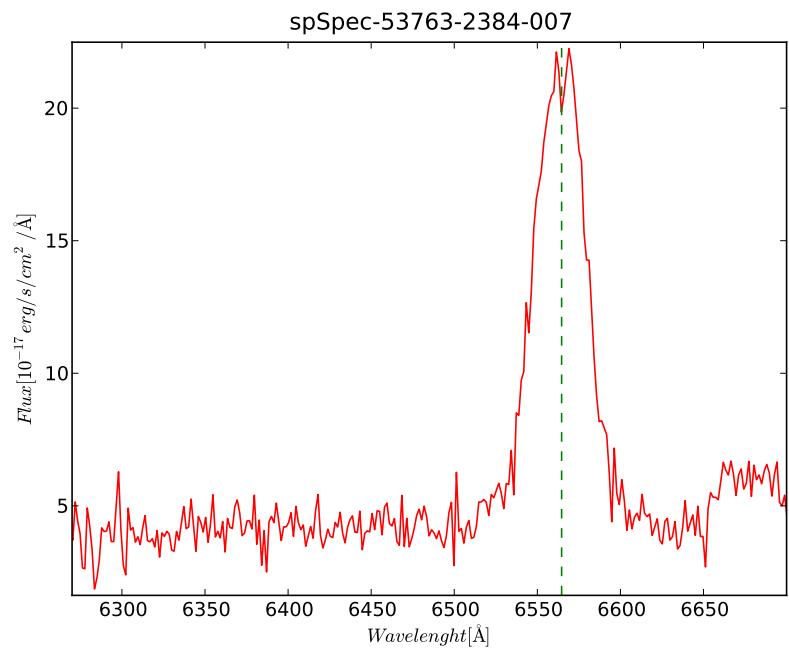


Figure 3.12: Example 2: SDSS J094325.89+520128.6

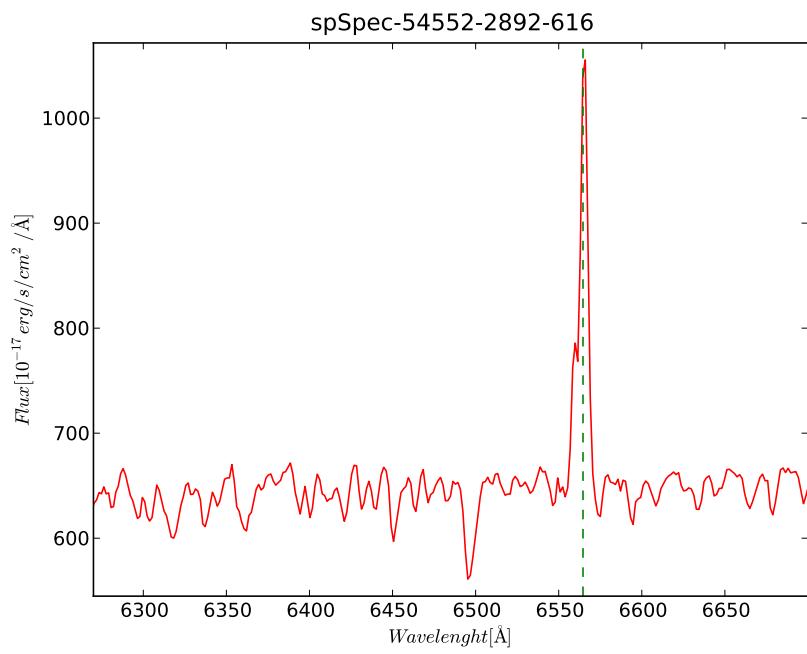


Figure 3.13: Example 3: SDSS J120729.12+003659.8

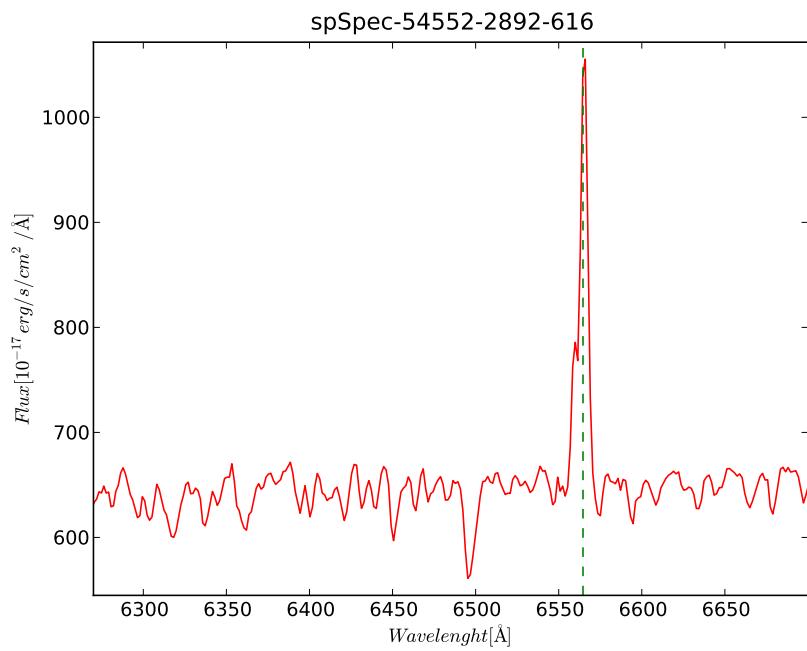


Figure 3.14: Example 4: SDSS J120908.18+194035.8

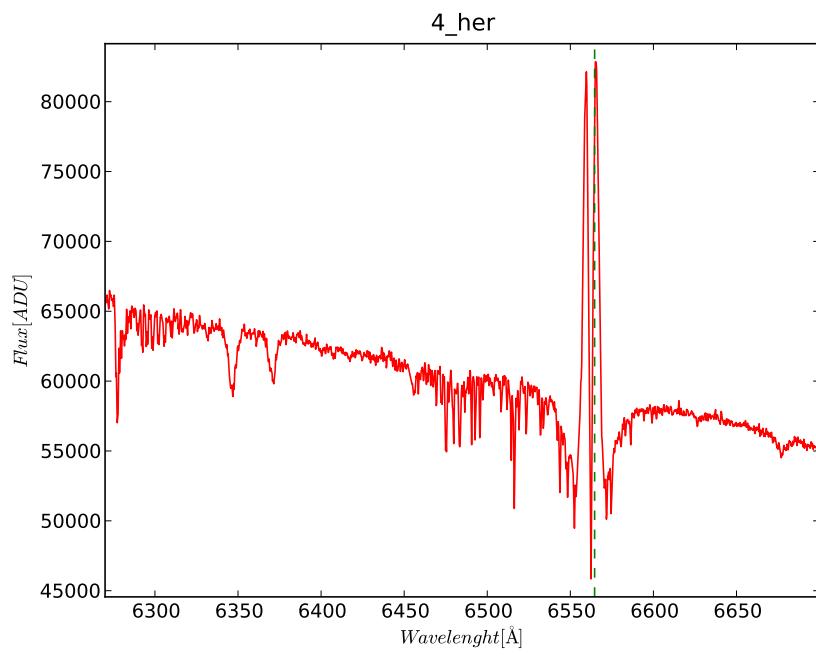


Figure 3.15: Spectrum of 4 Her. Be star

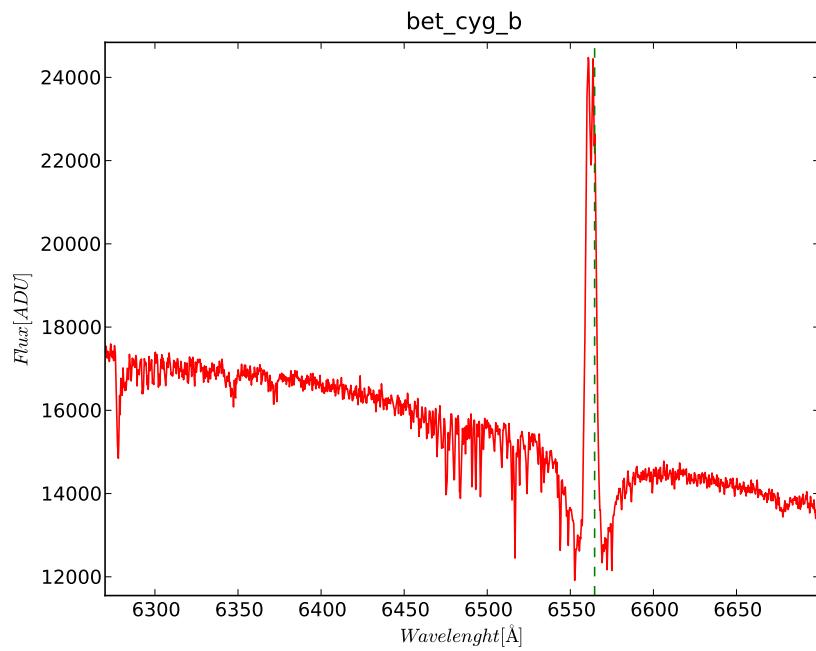


Figure 3.16: Spectrum of HR 7418 (Albireo B). A fast-rotating Be star, with an equatorial rotational velocity of at least 250 kilometers per second. Its surface temperature has been spectroscopically estimated to be about 13.200 K

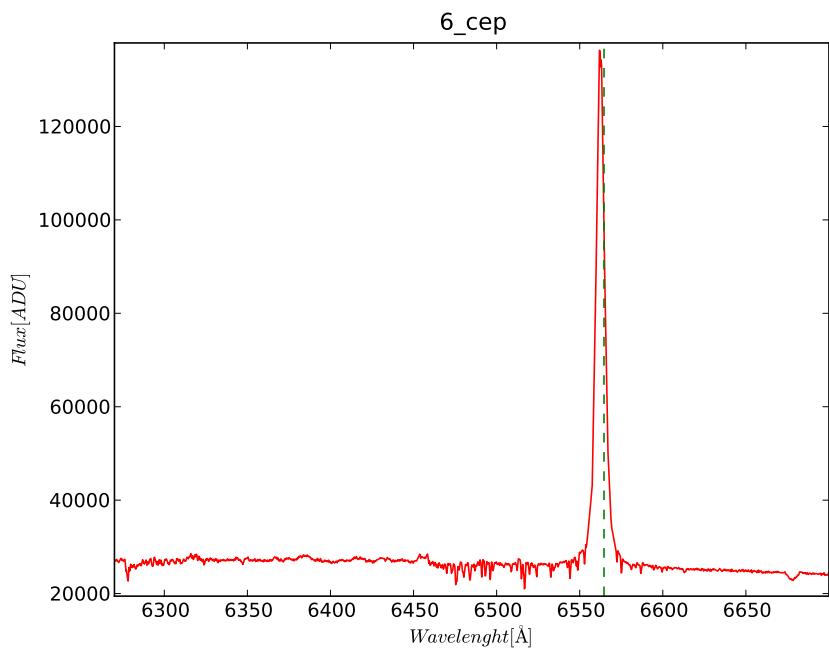


Figure 3.17: Spectrum of 6 Cepheus. Be star

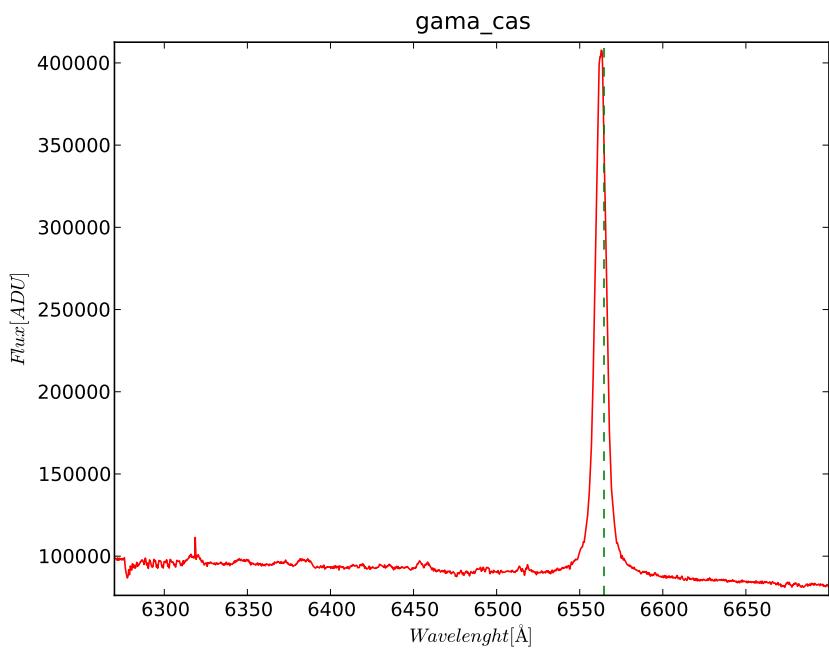


Figure 3.18: Spectrum of Gamma Cassiopeiae. Be Star

perform similar task over and over again. With some automation in mind such experiments are easy to do. Here is one of the test I have performed.

Natural idea one could have is using just emission line hight and write a program to check the spectra for condition if emission \geq threshold. Then we do not need "expensive" Data Mining algorithms to get some interesting results¹. Lets try to use Classification using just emission line parameter. Here is consequent tree.

```
1 J48 pruned tree
2 -----
3 alpha <= -0.464633
4 |   alpha <= -0.676474: be (45.0/18.0)
5 |   alpha > -0.676474: o (46.0/5.0)
6 alpha > -0.464633: be (92.0/16.0)
```

We can see that it is not that simple. In the training set there are some Be stars with extreme hight in the negative direction (absorption line). We have some statistics from classifier. In this example the effectiveness were 77.6%. Also there is Confusion Matrix

```
1 === Confusion Matrix ===
2 Be    0  <-> classified as
3 102    6 |  Be
4  35   40 |  Others
```

which indicates how the classifier will fail to perform in individual cases. Here it shows that it is almost exact when the object is Be star but it confuses others type of stars with Be stars.

This experiment proves that using Data Mining has a sense even in simple cases and can provide non-trivial insight on tested data.

¹This was actually done in the early stage of this project.

Chapter 4

Conclusion

The harvesting of large-scale astronomical data is challenging but solvable problem. The technology of Virtual Observatory offers solid background for data discovery and retrieval. The whole process can be automated using UN*X like approach of small and single purpose scripts or programs. The last stage of choosing the right characteristics and Data Mining method is even more complex task requiring deep understanding of the researched phenomena and Machine Learning theory and technology. The possible solution can be based on cooperation between experts in scientific and computer science field. Without such collaboration we are missing lots of opportunities.

It is evident that dealing with spectroscopic data is much more complicated but also more fruitful. One could extract many characteristic features which fit to the actual problem. The FITs standard is real godsend and makes work with spectra from different sources possible. The results obtained from the Data Mimicing process are reasonable. During the work it was also "discovered" how humans are good at visual judgment: when thumbnails of result spectra were created it provided much better understanding if something went wrong then statistics and numbers. This is one example how machines and humans could work together when we utilize ours and theirs natural abilities.

There are many aspect which could be done better, some of the considered but not implemented subjects are discussed here.

- Spectral Characteristics

The spectrum was characterized with few, very simple parameters, which can be similar in different types of objects. We have discussed ¹ many advanced possibilities such are wavelets eigen values etc. This could be subject of further investigation.

- Continuum fit

The simple linear function is too rough to capture true continuum features. There is an interesting and effective algorithm discussed in the paper: Advanced fit technique for astrophysical spectra by S. Bukvić et. al. from University of Belgrade [Bukvić et al., 2008] which seems ideal for this purpose.

¹Petr Skoda initiated rich and interesting email conversation with leading experts regarding this topic.

- More Data Mining algorithms

Originally more advanced approaches such as Support Vector Machines were considered.

- Larger training sample

To obtain large enough meaningful training sample of confirmed Be stars was a real problem and many surveys were consider (e.g. IPHAS) but without success.

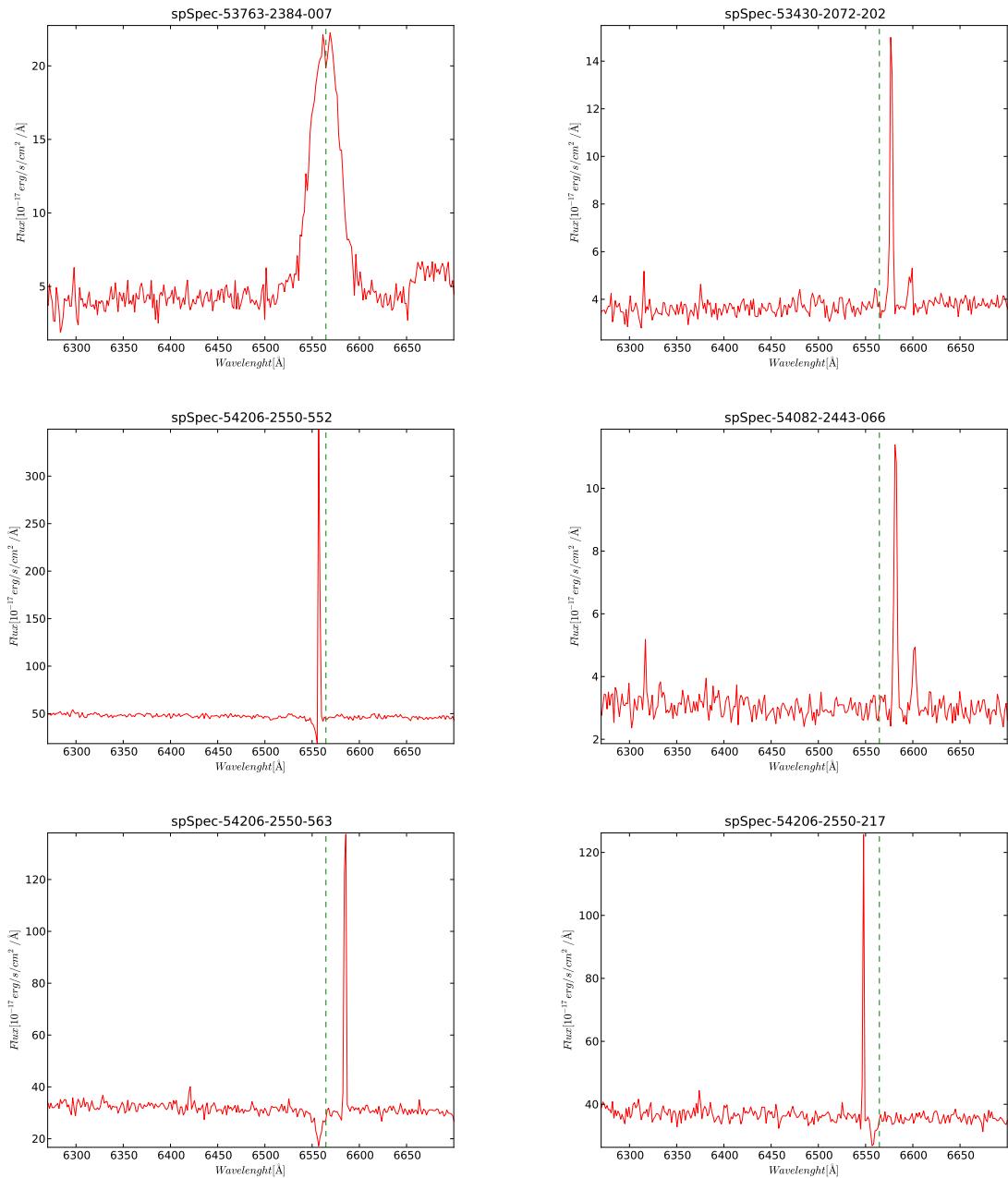
- Usage of Light curves

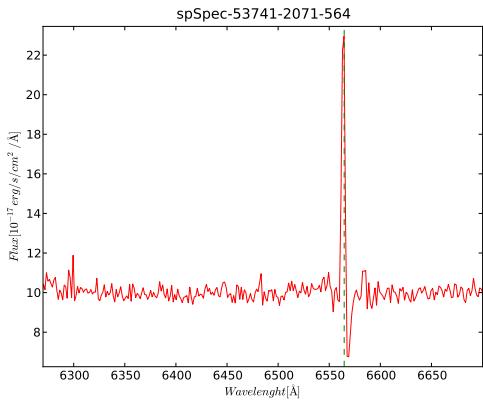
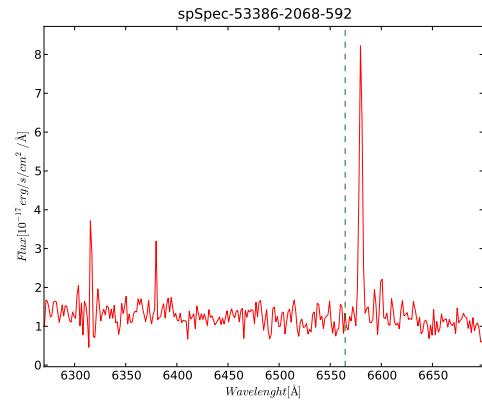
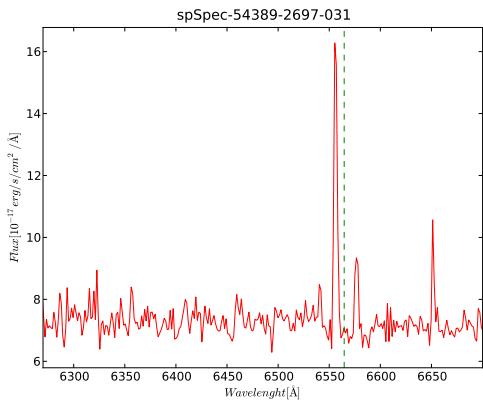
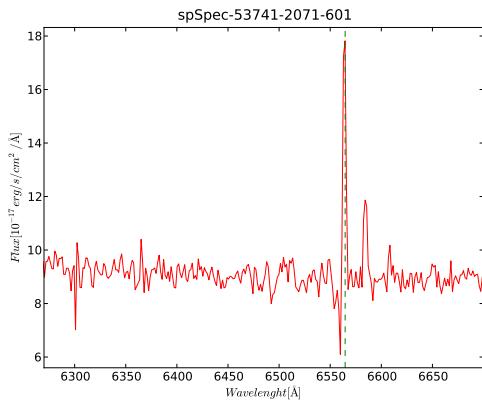
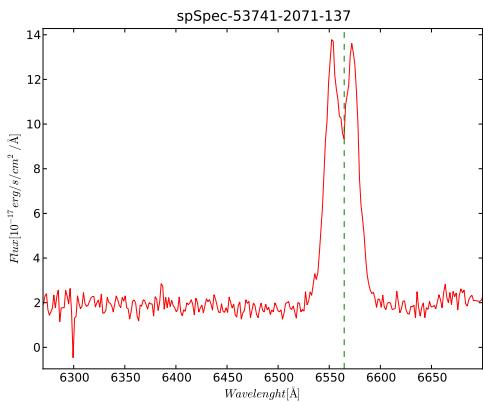
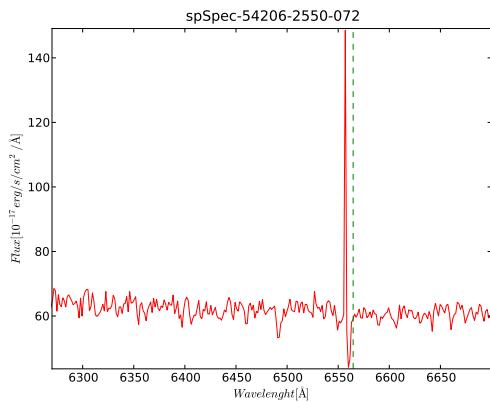
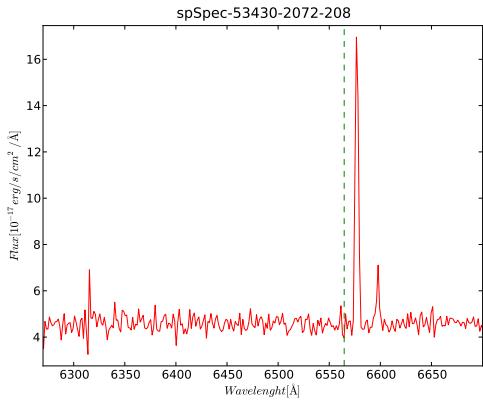
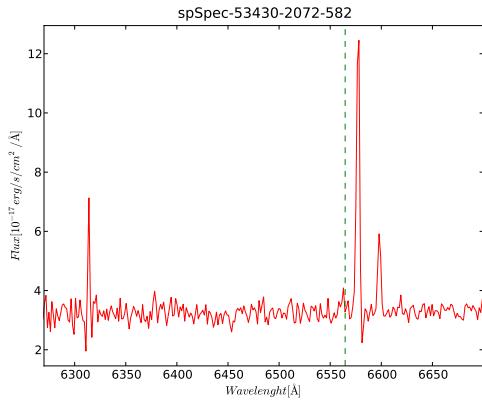
The whole process static but the "Be phenomena" is dynamic in nature. Using light curves could significantly improve the efficiency.

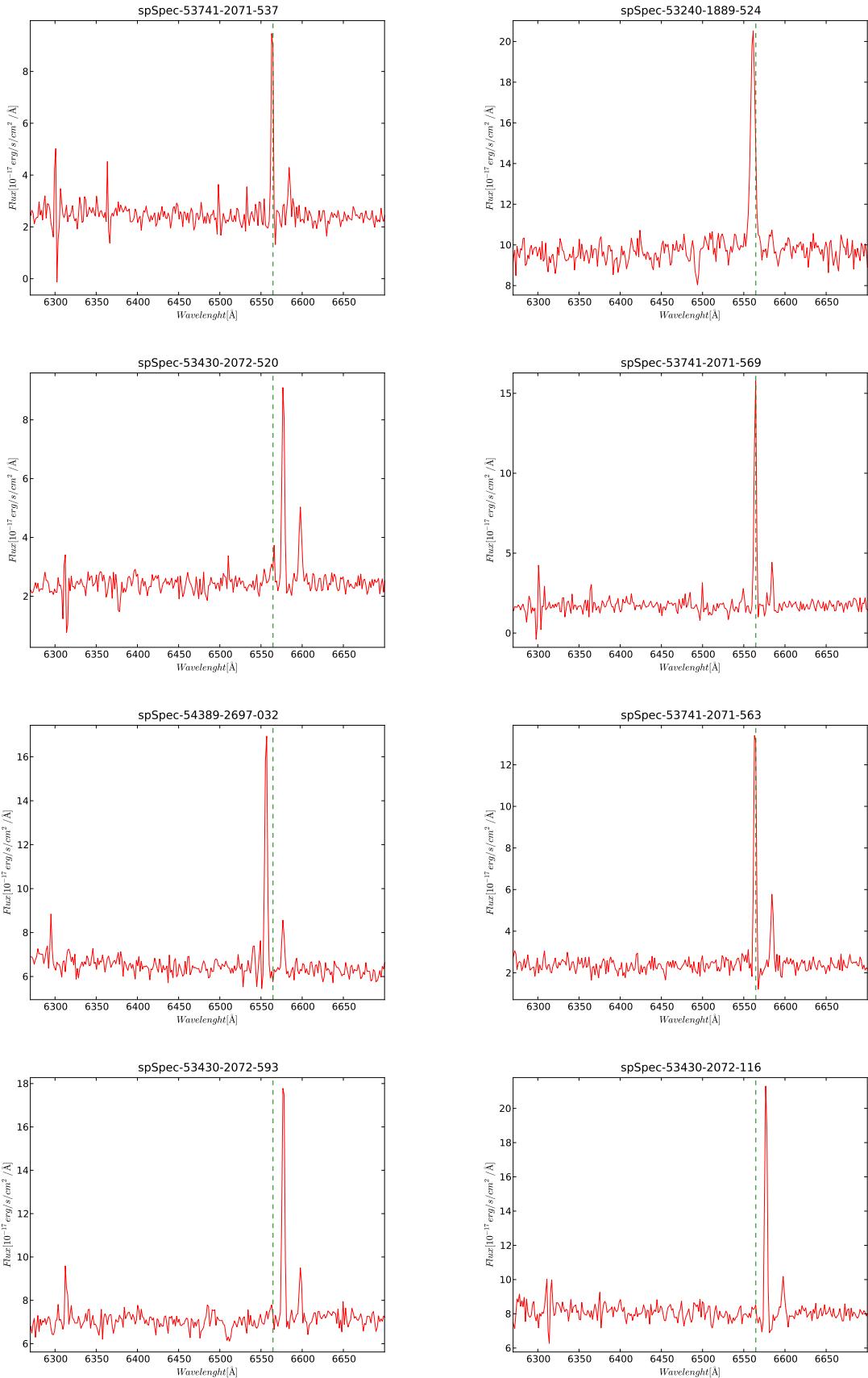
- Unknown errors

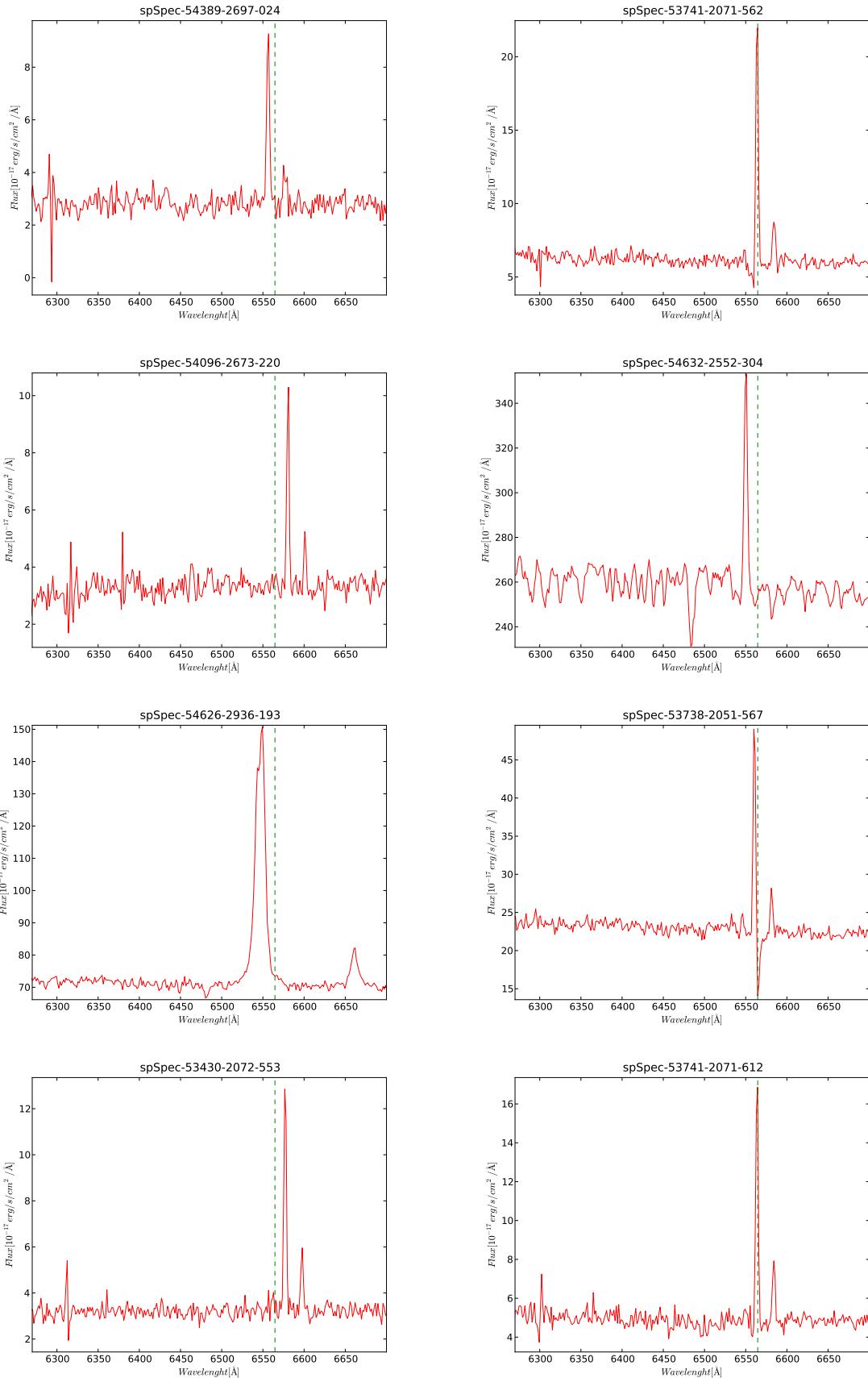
There are things we do not know we don't know. The overall process were very complicated and involved hundreds lines of codes. Any overlooked mistake could effect the results. The absence of evidence is not evidence of absence.

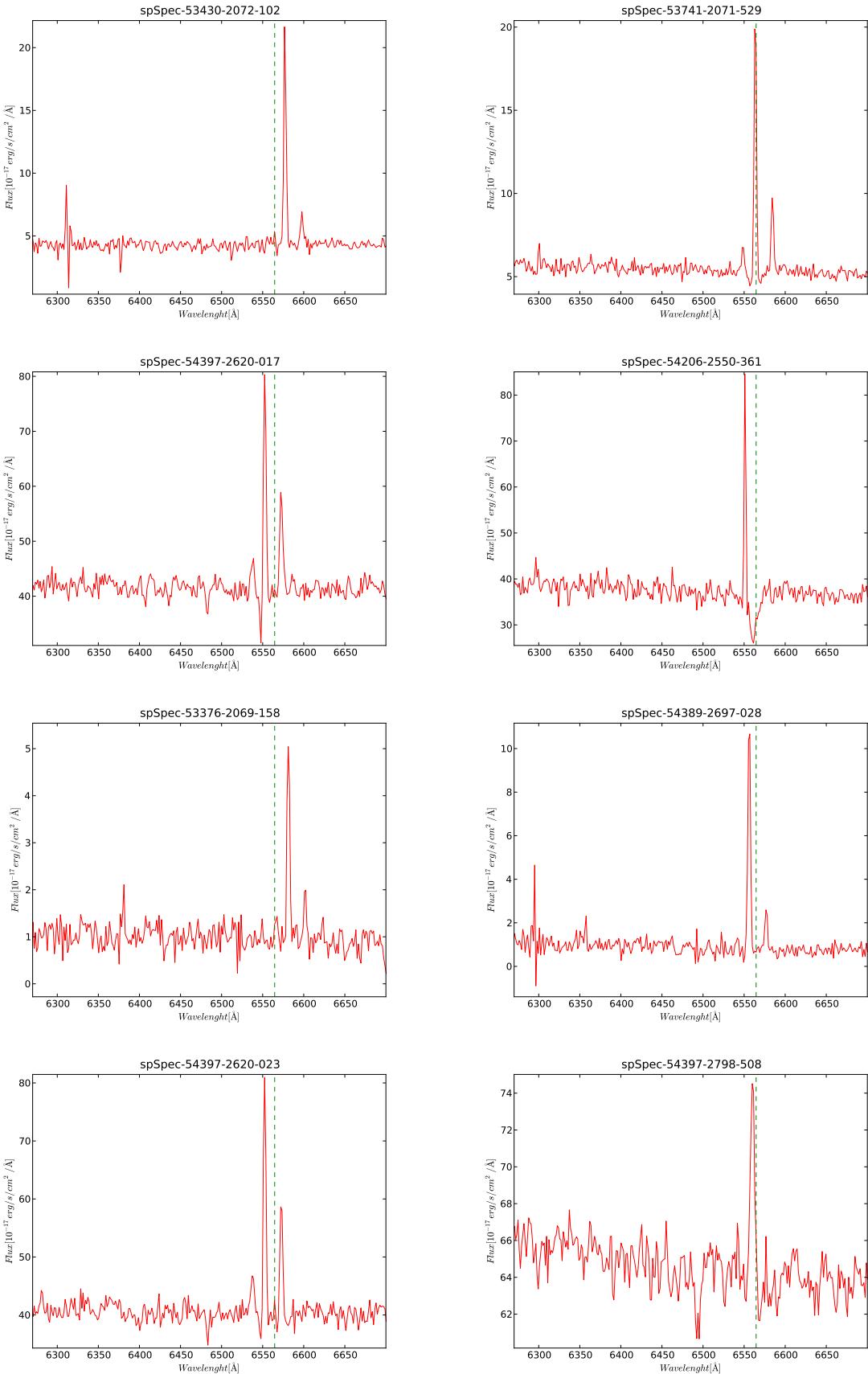
Appendix1: Spectra of result objects

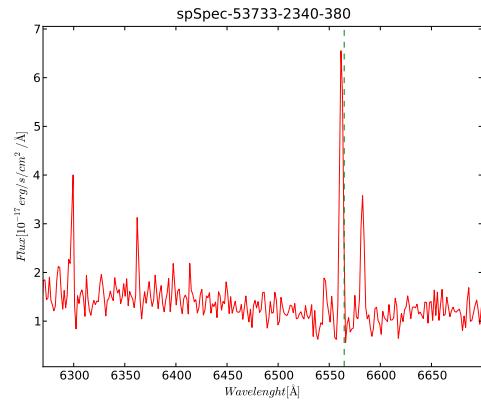
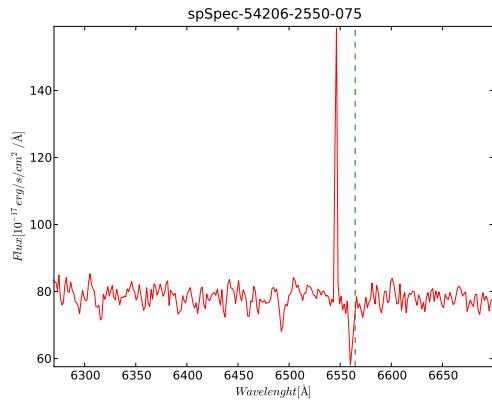
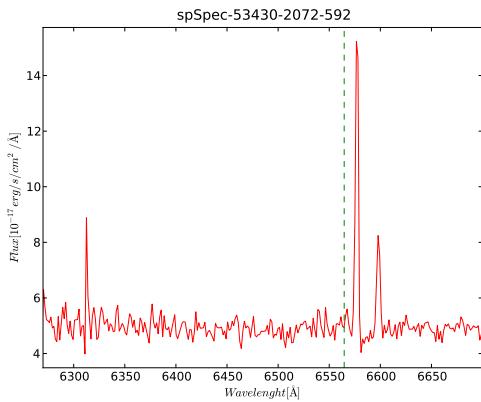
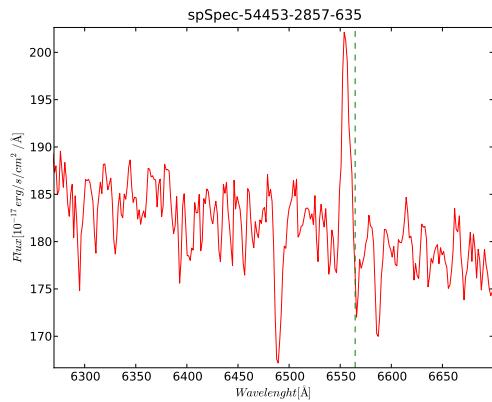
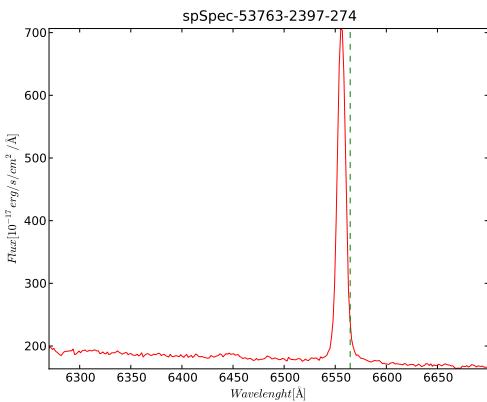
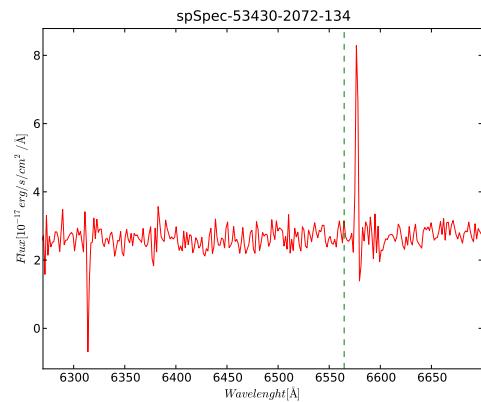
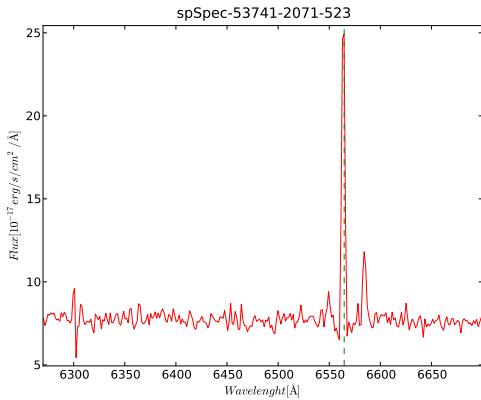
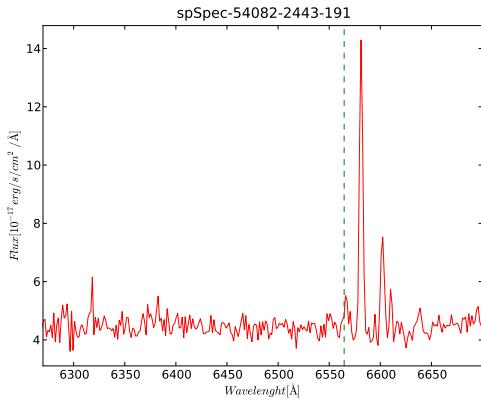




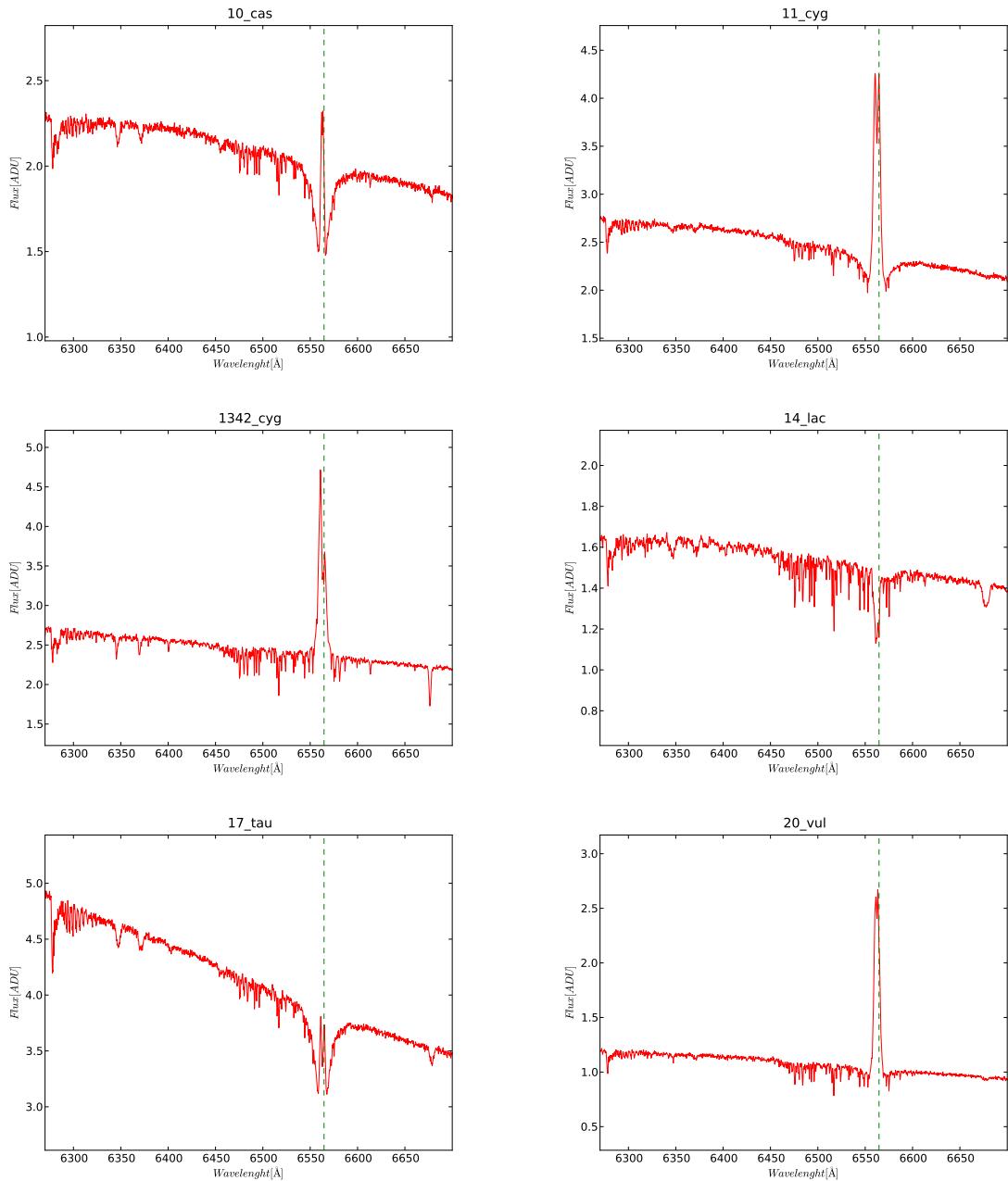


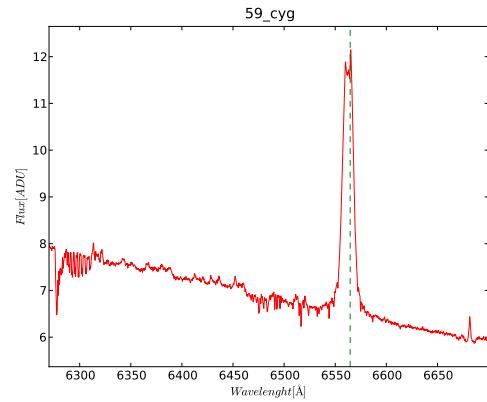
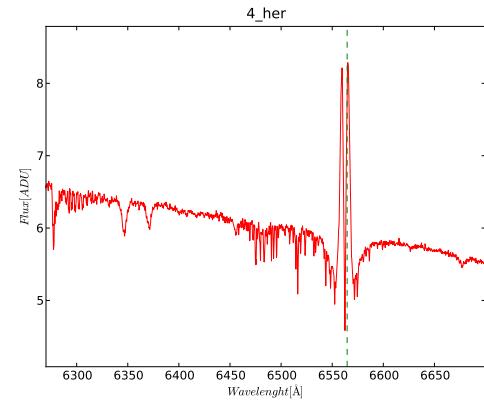
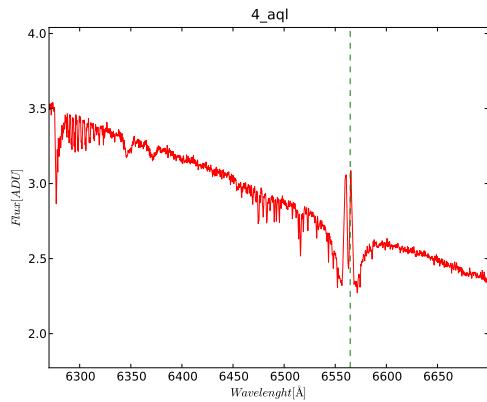
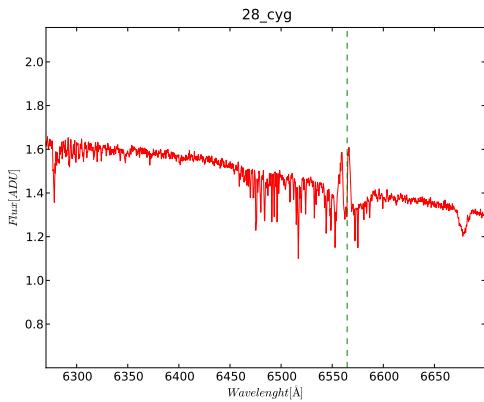
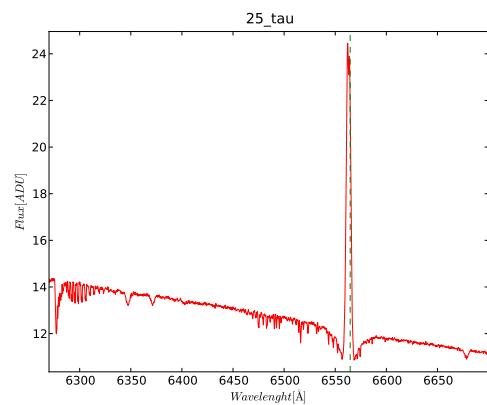
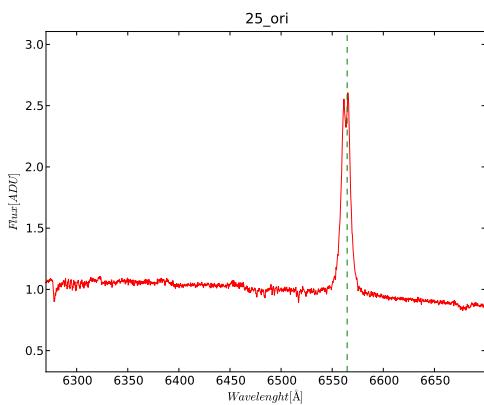
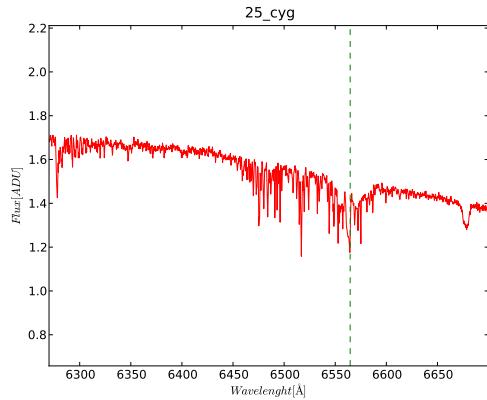
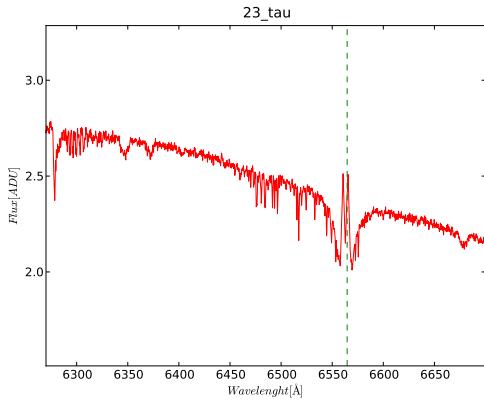


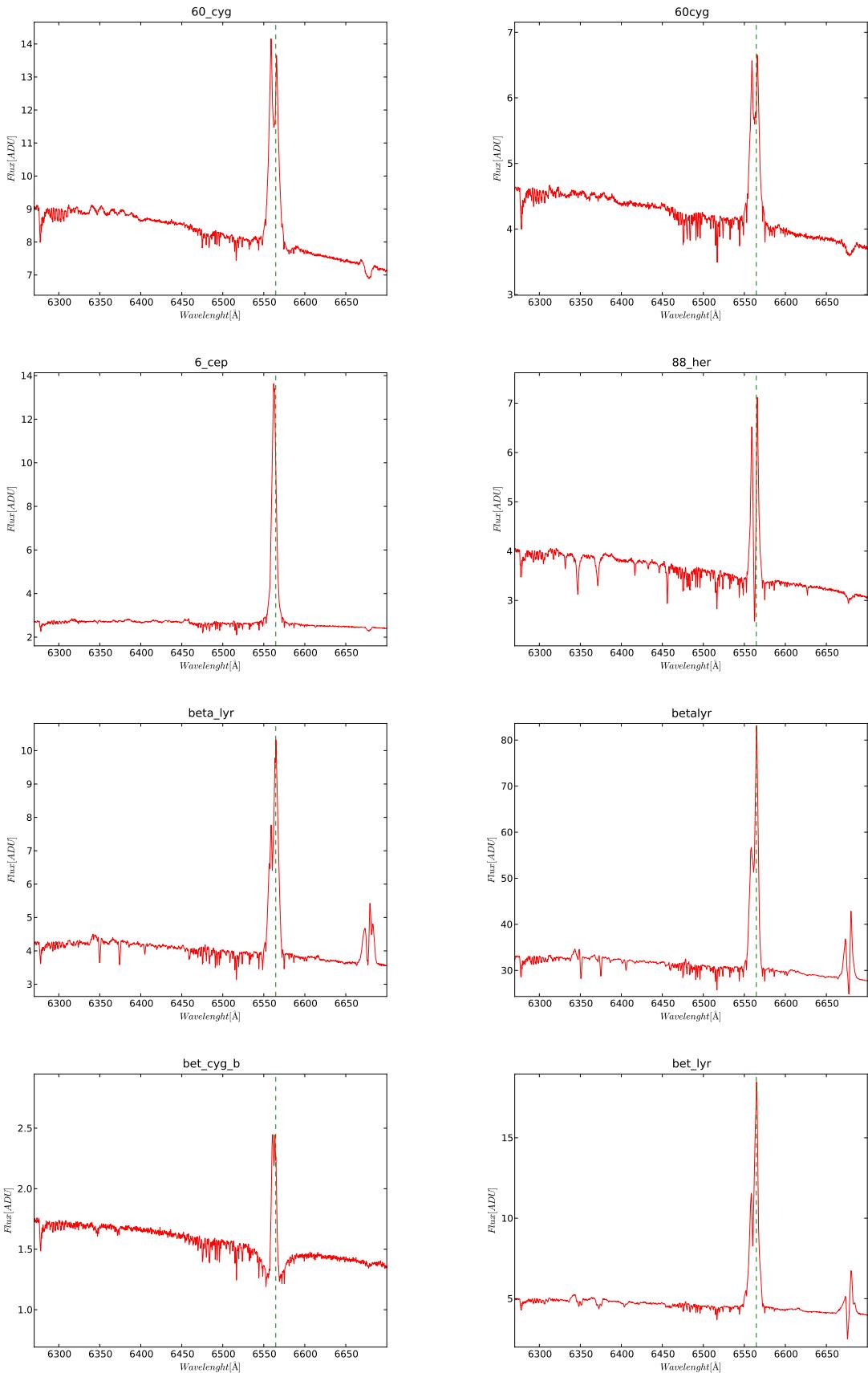


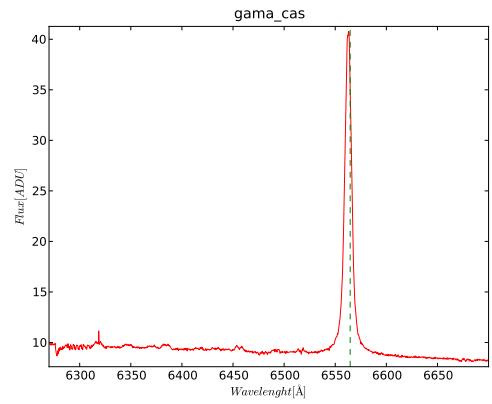
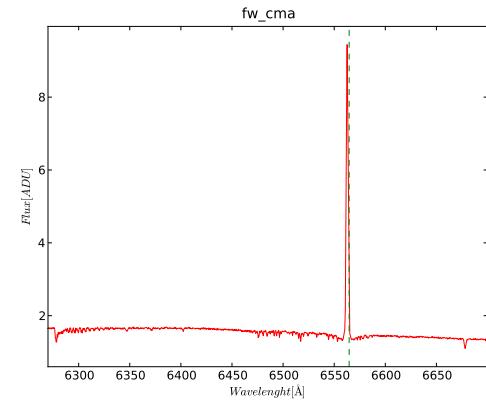
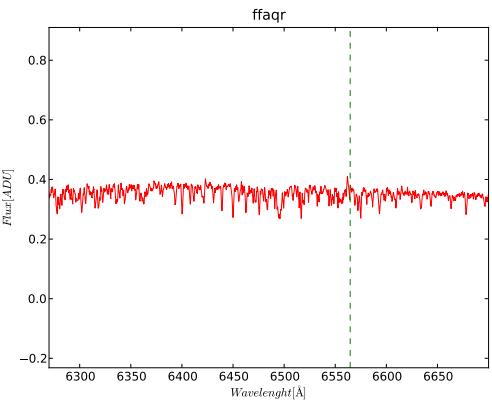
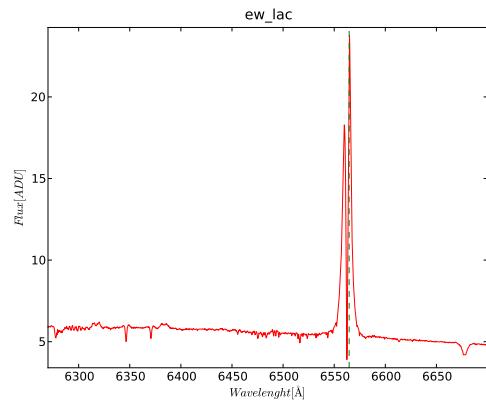
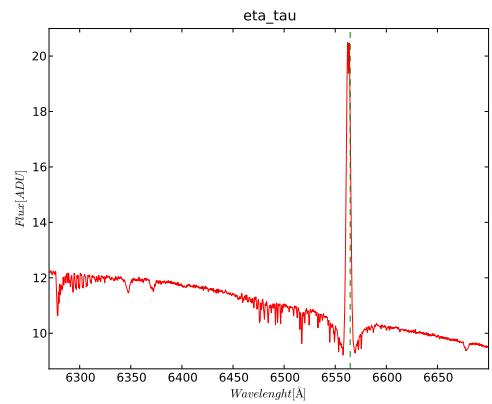
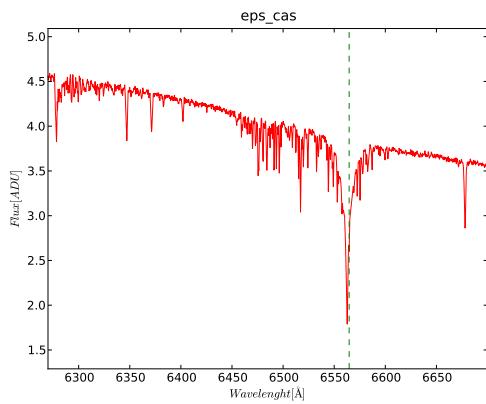
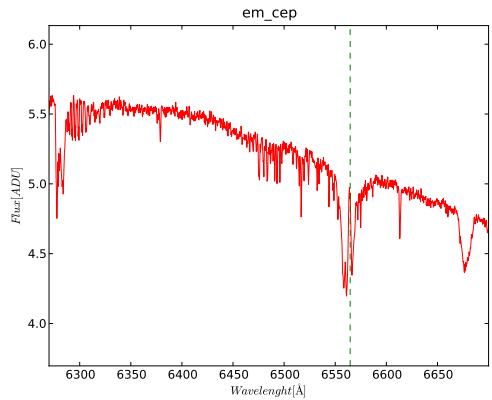
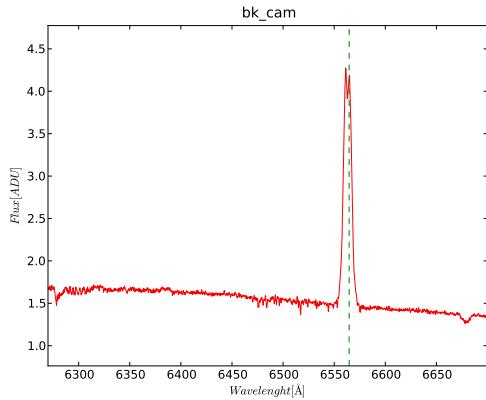


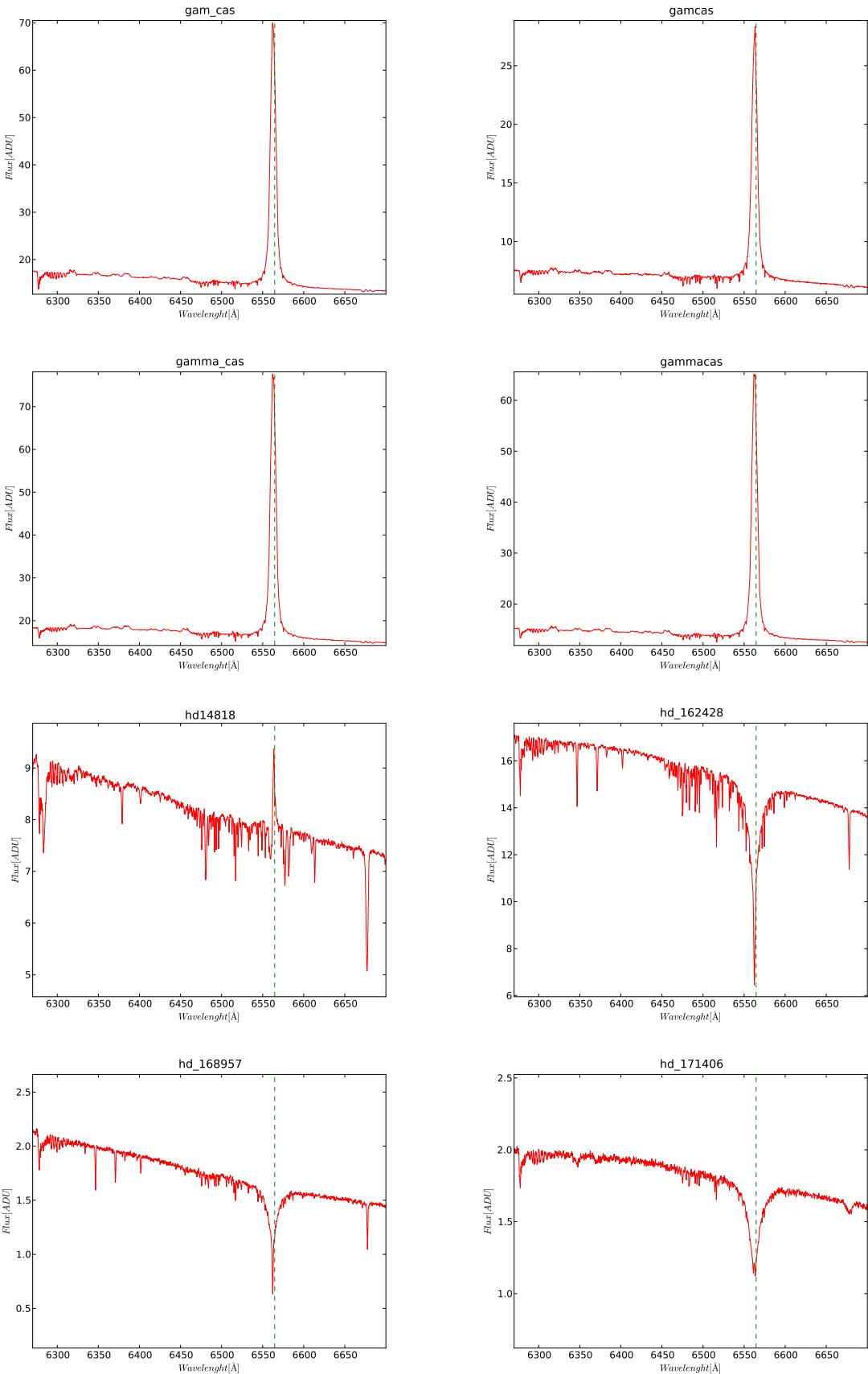
Appendix2: Spectra of Be stars from Ondřejov

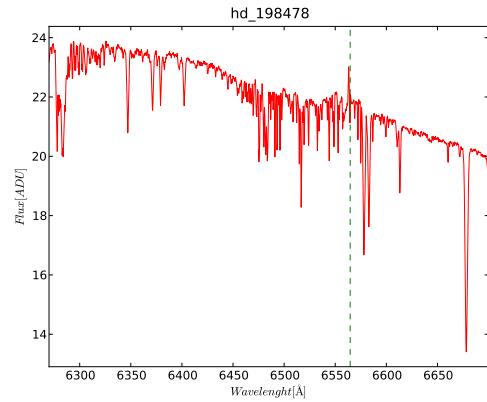
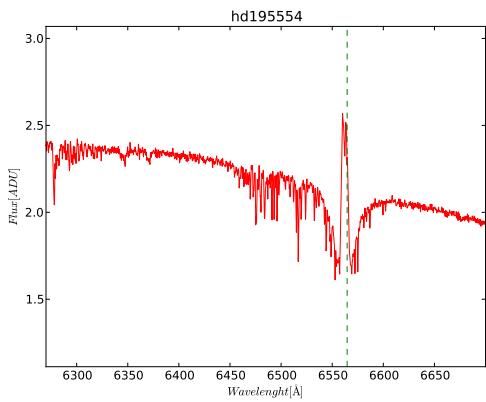
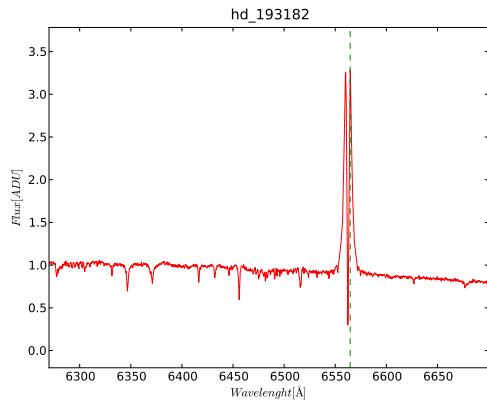
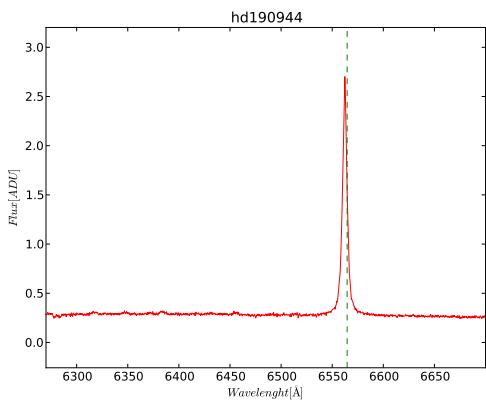
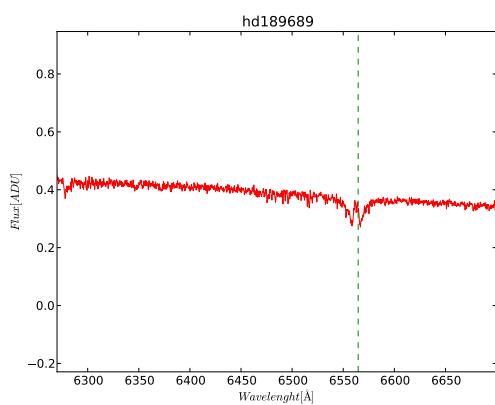
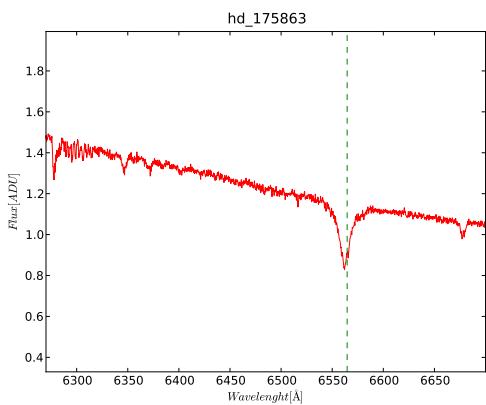
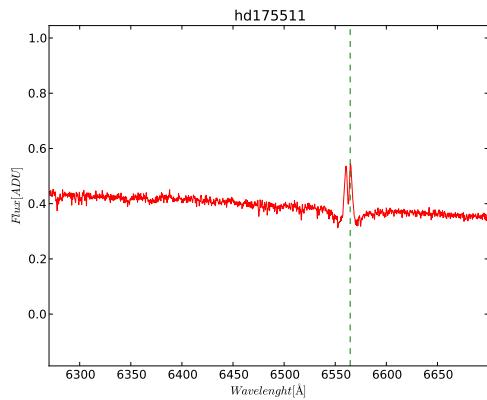
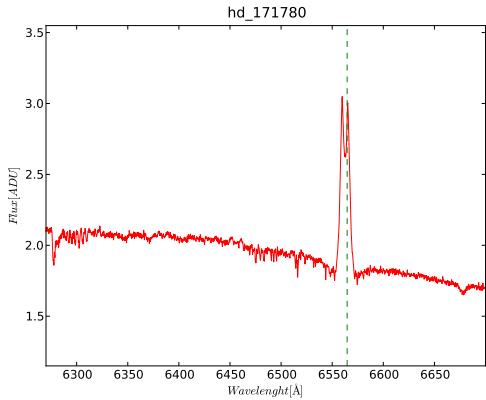












References

- N.M. Ball and D. Schade. The Long Range Plan for Canadian Astronomy: 2010-2020. 2010. [vii](#), [1](#)
- N.M. Ball, R.J. Brunner, and R. Gregory. Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19(7):1049–1106, 2010. ISSN 0218-2718. [13](#)
- J. Becla, A. Hanushevsky, S. Nikolaev, G. Abdulla, A. Szalay, M. Nieto-Santisteban, A. Thakar, and J. Gray. Designing a multi-petabyte database for LSST. *Arxiv preprint cs/0604112*, 2006. [3](#)
- K. Benson, R. Plante, E. Auden, et al. IVOA Registry Interfaces. *IVOA Working Draft*, 2009. [5](#)
- P. Berka. *Dobývání znalostí z databází*. Academia, 2003. ISBN 8020010629. [14](#)
- T. Berners-Lee and R. Cailliau. WorldWideWeb: Proposal for a HyperText project. *European Particle Physics Laboratory (CERN)*, 1990. [4](#)
- S. Bukvić, D. Spasojević, and V. Žigman. Advanced fit technique for astrophysical spectra. *Astronomy and Astrophysics*, 477(3):967–977, 2008. ISSN 0004-6361. [41](#)
- National Research Council. Preserving Scientific Data on our Physical Universe. 1995. [11](#)
- Free Software Foundation. GNU General Public License, 2007. URL <http://www.gnu.org/copyleft/gpl.html>. [iii](#)
- M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009. ISSN 1931-0145. [iii](#)
- RJ Hanisch and PJ Quinn. The international virtual observatory. *Retrieved from http://www.ivoa.net/pub/info/TheIVOA.pdf on*, 24, 2010. [3](#), [4](#)
- R. Hirata and T. Kogure. The Be star phenomena. II-Spectral formation and structure of envelopes. *Bulletin of the Astronomical Society of India*, 12:109–151, 1984. ISSN 0304-9523. [vii](#), [20](#)
- SB Kotsiantis, ID Zaharakis, and PE Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3, 2007. [14](#)

R.L. Launer. *Robustness in statistics: proceedings of a workshop*. Academic Pr, 1979.

[30](#)

MAC Perryman, L. Lindegren, J. Kovalevsky, E. Hoeg, U. Bastian, PL Bernacca, M. Crézé, F. Donati, M. Grenon, F. van Leeuwen, et al. The HIPPARCOS catalogue. *Astronomy and Astrophysics*, 323:L49–L52, 1997. ISSN 0004-6361. [22](#)

J.M. Porter and T. Rivinius. Classical Be Stars. *Publications of the Astronomical Society of the Pacific*, 115(812):1153–1170, 2003. ISSN 0004-6280. [1](#), [19](#)

B. Schlesinger. A Users Guide for the Flexible Image Transport System. 1997. [11](#)

MF Skrutskie, RM Cutri, R. Stiening, MD Weinberg, S. Schneider, JM Carpenter, C. Beichman, R. Capps, T. Chester, J. Elias, et al. The two micron all sky survey (2MASS). *The Astronomical Journal*, 131:1163, 2006. [22](#)

A. Slettebak. The Be stars. *Publications of the Astronomical Society of the Pacific*, 100:770–784, 1988. ISSN 0004-6280. [vii](#), [20](#)

MH Van Kerkwijk, L. Waters, and JM Marlborough. Halpha emission and infrared excess in Be stars: probing the circumstellar disc. *Astronomy and Astrophysics*, 300:259, 1995. ISSN 0004-6361. [22](#)

M. Wenger, F. Ochsenbein, D. Egret, P. Dubois, F. Bonnarel, S. Borde, F. Genova, G. Jasniewicz, S. Lalo ”e, S. Lesteven, et al. The SIMBAD astronomical database. *Arxiv preprint astro-ph/0002110*, 2000. [7](#)

I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann Pub, 2005. ISBN 0120884070. [14](#)