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: Božena Kováčová, Tereza Jeřábková, Josef Pacula, Petr Šafařík and Andrej Pančík.

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 the 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 digitized surveys. Based on proven standards it offers an 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 a 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

List of Figures

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 & 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.

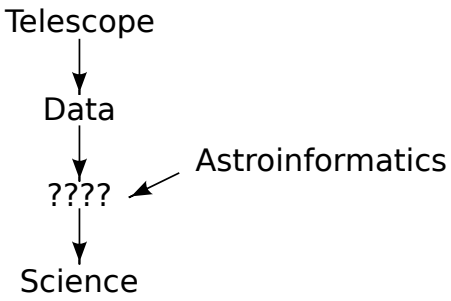


Figure 1: Astroinformatics in the context of astronomy [Ball & Schade, 2010]

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 & Rivinius, 2003] candidates in the astronomical surveys. More than straightforward process it is trial and error approach probing new possibilities.

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

The 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. The 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. The third chapter introduces issues of Be stars. The chapter Four is practical application of previously described technologies and methods. Training data of confirmed Be stars from Ondřejov are cross-matched with other catalogues to obtain color indexes and

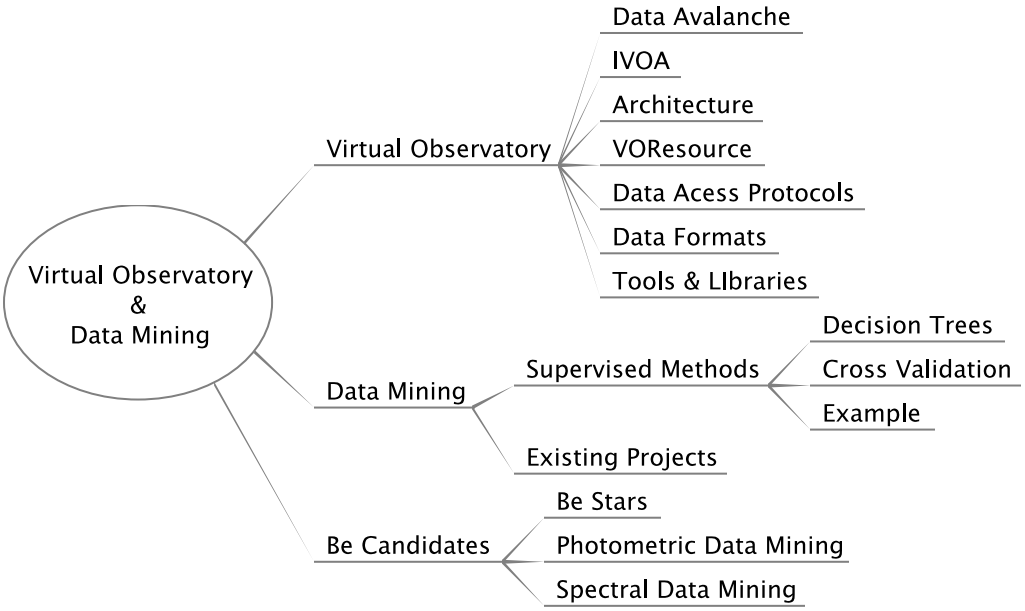


Figure 2: Thesis structure

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, that is why they are short and without auxiliary technicalities such as 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
getSpectra	Get spectra links from SSA Server
madmax	Extract features from spectra
convolve	Reduction of Ondřejov spectra
pf	Print spectrum from the FITS file
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 other scientist around the world. Source codes were maintained by GIT version system allowing easy sharing. All software used and produced is open source.

[1] git://github.com/astar/diplomaWork
[2] http://physics.muni.cz/~vazny/wiki/index.php/Diploma_work

Chapter 1

Virtual Observatory (VO)

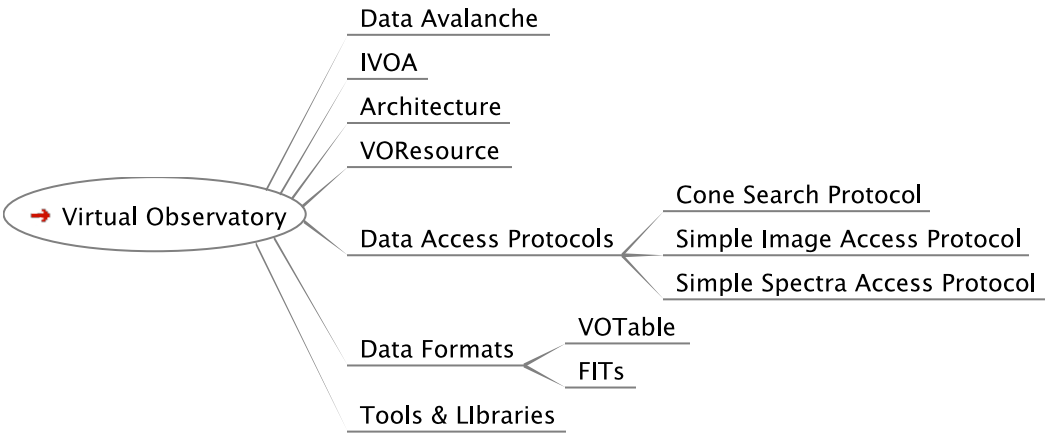


Figure 1.1: Chapter structure

1.1 Data avalanche: Opportunity or disaster?

There are two important trends in current astronomical surveys:

- Size: The cumulative compressed data holdings of the ESO archive will reach 1 Petabyte by 2012 [Hanisch & 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 phenomenas. [Hanisch & Quinn, 2010]

Such an amount of data is not possible to transfer over the network. Data resources are heterogeneous, distributed and decentralized in their 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-Lee[1] designed set of protocols (URIs, HTTP and HTML) which allowed to cross-link and share documents [Berners-Lee & Cailliau, 1990]. This was recognized as generally useful and World Wide Web was born. An important role in developing Web standards plays the World Wide Web Consortium (W3C) [2].

1.2 International Virtual Observatory Alliance (IVOA)

For handling of heterogeneous distributed data it is necessary to have the set of common standards and protocols as well as an authority encouraging their implementation. Such an authority is the International Virtual Observatory Alliance (IVOA). It currently 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 & 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. The 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/IP [3] protocol.

The VO architecture is serviced oriented. Each service is autonomous with well defined boundaries. The very important aspect of VO implementation is the adoption of formats and protocols used in astronomy (FITS) and Computer Science (XML [4]

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

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

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

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

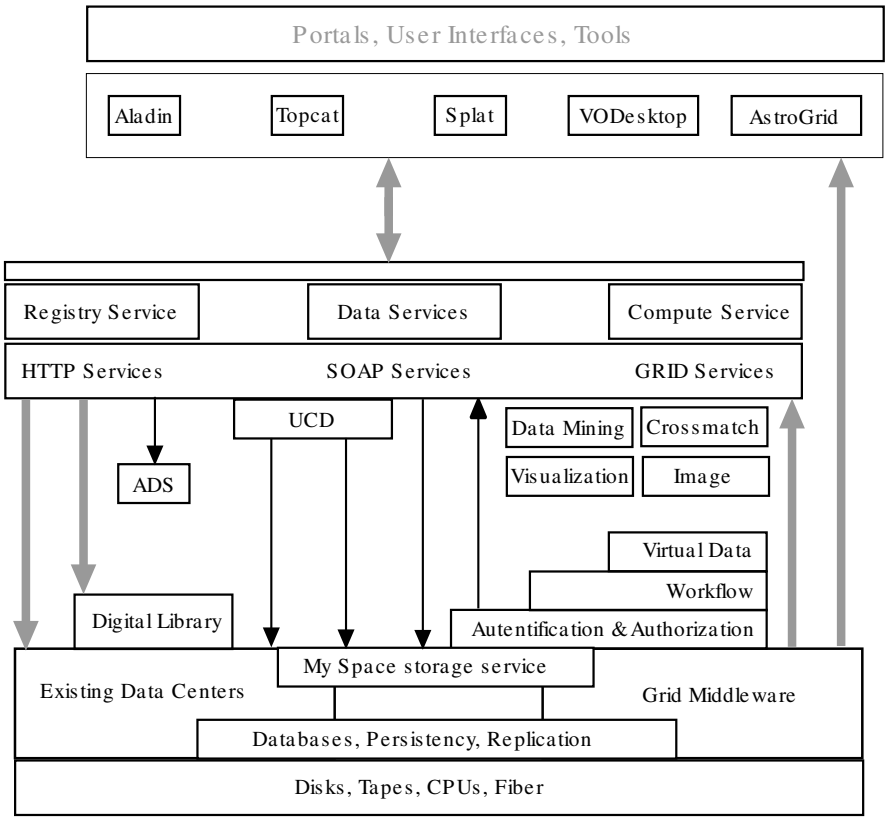


Figure 1.3: VO Architecture

, Web Service[1] SOAP[2],REST[3]) for many years. In other words VO does not try to reinvent the wheel but it stands on the shoulders of giants.

1.4 VO Registry

The VO Registry will allow an astronomer to be able to locate, get details of, and make use of, any resource located anywhere in the ivo ://space, i.e. in whole Virtual Observatory. The IVOA will define the protocols and standards whereby different registry services are able to inter-operate and thereby realize this goal.

1.5 VO Resources

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]

[1]method of communication between two electronic devices over a network.

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

[3]Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. Used in www.youtube.com

The UML[1] diagram of the resource is on the figure 1.4. The 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. An organization can be linked (associated) together with other organization (multiplicity is represented by number 0,1,...). The same is true for data collection. Organization is a generalization of and/or provider which can own zero to N services. The diamond means the aggregation. The publisher can have any resources.

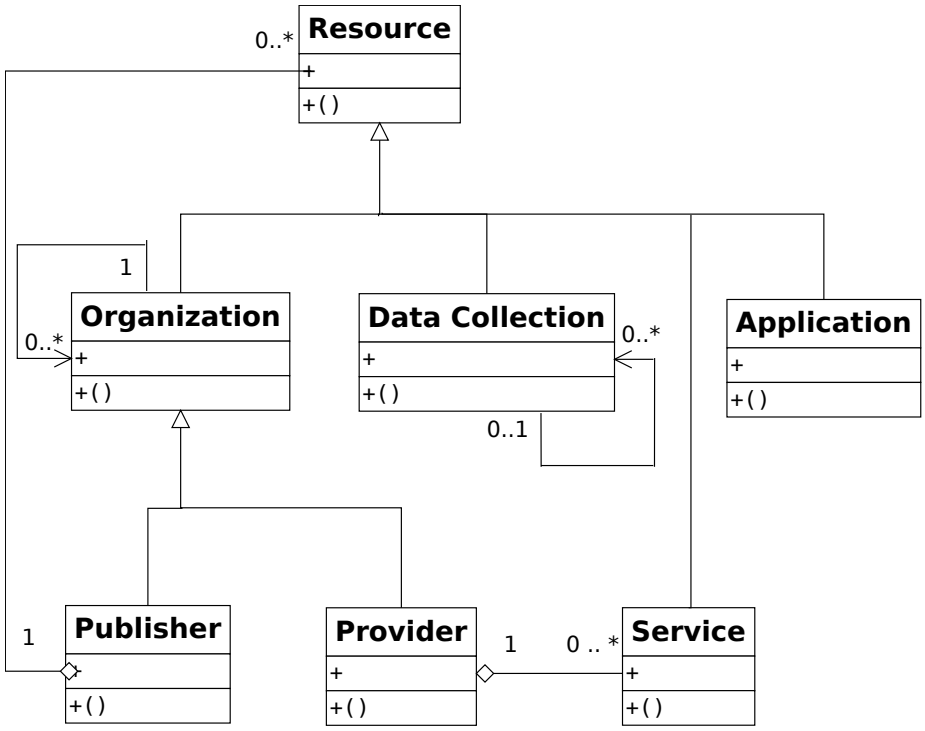


Figure 1.4: UML diagram of VOResource

Following example uses program stilts[2] to query registry with parameter shortName equal to 'AIASCR'[3]. This returns VOTable (see section 1.7.1) 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 (xmlns[4]) followed by information 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"
```

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

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

[3]Astronomical Institute of the Academy of Sciences of the Czech Republic

[4]XML namespaces. Provide uniquely named elements and attributes in an XML document.

```
 xmlns="http://www.ivoa.net/xml/VOTable/v1.1">
.
<DATA>
<TABLEDATA>
  <TR>
    <TD>ivo://asu.cas.cz</TD>
    <TD>AIASCR</TD>
    <TD>Astronomical Institute of the Academy of Sciences of the Czech
        Republic Naming Authority</TD>
    <TD>http://stelweb.asu.cas.cz/web/index/index-en.php</TD>
    <TD>Petr Skoda &lt;skoda@sunstel.asu.cas.cz&gt;</TD>
```

1.6 Data Access Protocols

Protocols are a very important part of Virtual Observatory. Their understanding is a key to comprehend the concepts behind VO. They allow to discover a resource and obtain desirable data. All of them are based on existing web standards and are designed to be simple and implement in existing astronomical archives. The main idea is simple and universal: HTTP GET request with parameters is sent to the resource and structured document (VOTable) is sent back. There are many Data Access Protocols like TAP (Table Access Protocol) or SLAP (Single Line Access Protocol). The most evolved and important for the practical part of this work are described below.

1.6.1 Cone Search Protocol

The 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 searched 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 The 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. A respond to a HTTP GET request represented by a URL

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

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

   ```
   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 searched radius of the cone in the ICRS coordinate system in decimal degrees. These parameters are required, others are optional.

2. A 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.
    pl?RA=24.5&DEC=-57&SR=0.1')
3 print response.read()
```

The same result can be obtained using program like wget[1] or Web browser.

1.6.2 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 1.6.1), 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. A SIA service must support:

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

Other optional services may be provided by servers:

- 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 sent via HTTP GET method. The first part is base URL. The second part are parameters specifying image properties such as position (POS), size of searched radius (SIZE), etc.

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

```
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.

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

There is more complex example using the 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.

```
In [1]: from astrogrid import Registry, ConeSearch
In [2]: list = reg.searchSiap('SDSS')
In [3]: print [p['id'] for p in list]
-------> print([p['id'] for p in list])
['ivo://nasa.heasarc/skyview/sdss']

In [4]: siap = SiapSearch('ivo://nasa.heasarc/skyview/sdss')
In [5]: result = siap.execute(18.8, -0.8, 1.0)
```

1.6.3 Simple Spectra Access Protocol

SSA Protocol allows to discover and obtain 1-D spectra from VO Service. It shares many similarities with 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 consists of following steps:

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

The spectra could be one of the following types:

- Observed spectra.
- Theoretical spectra.

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

1. HTTP GET interface, understanding at least parameters POS, SIZE, TIME (time and date of exposure), BAND (spectral range covered) and returning the query response encoded as a VOTable document.
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.

```
stilts regquery query="shortName like 'ESO' capability/@standardID =
'ivo://ivoa.net/std/SSA'" ocmd="keepcols 'ShortName accessUrl'"
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*.

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

1.7 Data Formats

Astronomy has always been in forefront of image producing and processing. This is especially true for the era of digitization. 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 implements these formats in a sensible way into its own standards.

1.7.1 VOTable

Motivation

VOTable is an flexible storage and exchange format fundamentally interconnected with Virtual Observatory. It has features for big-data and Grid computing. Data can be stored in the different ways in dependence of their nature and size. Small tables can be stored in pure XML [1], while large-scale data can be referenced with the URL [2] syntax protocol://location. It combines web standards (it is based on XML) and astronomy tradition in storing data (it is FITS compatible). The expiration and the 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 Schema [3]. TABLE tag encapsulates tabular data. FIELD tag describes identification (ID), type and precision of columns. DATA tag contains data (here) in TABLEDATA format (other types are FITS and BINARY).

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Produced with vo.table version 0.6
     http://www.stsci.edu/trac/ssb/astrolib
     Author: Michael Droettboom <support@stsci.edu> -->
```

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

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

[3] Define the legal building blocks of an XML document.

```
<VOTABLE version="1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="http://www.ivoa.net/xml/VOTable/v1.0"
 xmlns="http://www.ivoa.net/xml/VOTable/v1.0">
 <RESOURCE type="results" >
  <TABLE >
   <FIELD ID="col0" name="wave" datatype="float" unit=""
   precision="F9"/>
  <DATA>
    <TABLEDATA>
     <TR>
      <TD>4012.50757</TD>
     </TR>
 </TABLEDATA>
   </DATA>
  </TABLE>
 </RESOURCE>
</VOTABLE>
```

Examples

All examples were created using ATpy [1]. Following example shows transformation of FITS into VOTable.

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

1.7.2 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." [Thibodeau, 1995]

FITS (Flexible Image Transport System) was originally created for data exchange between WSRT [2] and the VLA [3] [Schlesinger, 1997]. It is now used as a file format to store, transmit, and manipulate scientific data and it is (thanks to it's revolutionary design) de facto standard in astronomy.

Structure

A FITS file can contain several HDUs (Header and Data Units). The first part of each HDU is the header, composed of ASCII card images containing keyword=value

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

[2] Westerbork Synthesis Radio Telescope

[3] Very Large Array

statements that describe the size, format and structure. A FITS file shall be composed of the following FITS structures, in the order listed:

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

Standards and documents related to FITS are maintained by IAU-FWG [1] and available at http://fits.gsfc.nasa.gov.

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.

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

Printing primary HDU.

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

Updating FITS file.

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

[1] International Astronomical Union FITS Working Group.

1.8 VO Tools & Libraries

There are many programs and libraries allowing user to interact with VO services. Such application is called to be VO-enabled. Thanks to the openness and the standardization anyone can develop his own application or enable existing[1] 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. A such diversity is healthy and probably the only possible way to ensure natural evolution of Virtual Observatory.

Every VO-enabled application can communicate with each other by sending messages and data (e.g. in VOTable format) by protocols PLASTIC and SAMP. It enables to create complex analysis systems by chaining individual components.

[1] For example Astroweka or Mirage.

Chapter 2

Data Mining

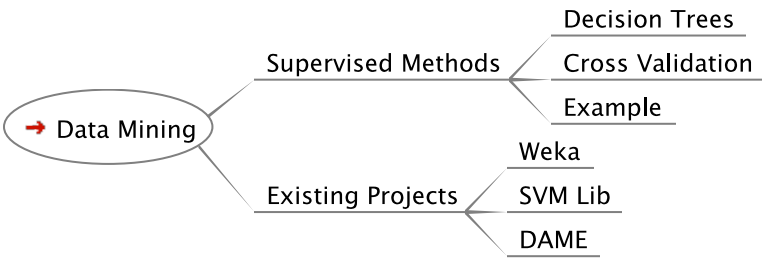


Figure 2.1: Chapter structure

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 hand this field is very complex with dozens of different methods and algorithms. This form needs and opportunity for interdisciplinary cooperation with Data Mining experts. This can be very beneficial for both fields, providing astronomers with interesting methods for data analysis and computer scientist with the large ammout 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)

DT Is an example of supervised classification. Based on final number of data $(x^{(1)}, \ldots, x^{(p)})$ with known class $C_1, \ldots, C_m$ classifier is created, i.e. mapping f clas-

sifying any $x \in \mathcal{X}, f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is a set of possible input vectors and $\mathcal{Y}$ is a set which values represent classes $C_1, \ldots, C_m$ (for example $\mathcal{Y} = 1, \ldots, m$). 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.
- Internal details are easily seen (white box model).
- Perform well with large data in a short time.

In pseudo-code, the general algorithm for building decision trees is demonstrated below [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

In practical part of this work algorithms C4.5 was used for several reasons: Its code is available and free implementations exist (J48 in Weka) and it is de-facto standard. The key question of DT algorithm is how to choose attribute for splitting the tree. C4.5 uses measure based on information entropy:

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

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 of that class. One of the technique used here is cross-validation.

The data are divided into fixed number of partitions (folds) where each in turn is used for testing while the reminder is used for training. Finally, the error estimates of partitions are averaged to yield an overall error. The standard way is to use 10-fold cross-validation. This number is a result of tests on numerous data sets [Witten & 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. Altogether 298 objects were used (100 stars, 99 galaxies, 99 QSOs). 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 [Fukugita et al., 1996]

```
SELECT TOP 100 u-g,g-r,r-i,s.specClass
FROM PhotoPrimary p join SpecPhotoAll s on p.objid=s.objid
WHERE s.specClass in (1)
AND u between 18 and 19
UNION all
SELECT top 100 u-g,g-r,r-i,s.specClass
FROM PhotoPrimary p join SpecPhotoAll s on p.objid=s.objid
WHERE s.specClass in (2)
AND u between 18 and 19
UNION all
SELECT top 100 u-g,g-r,r-i,s.specClass
FROM PhotoPrimary p join SpecPhotoAll s on p.objid=s.objid
WHERE s.specClass in (3)
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.

```
Correctly Classified Instances         277               92.953 %
Incorrectly Classified Instances        21                7.047 %
Kappa statistic                          0.8943
Mean absolute error                      0.0736
Root mean squared error                  0.2096
Relative absolute error                 16.5627 %
Root relative squared error             44.4707 %
Total Number of Instances              298
```

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 by Weka for this example is following:

```
ug <= 0.663668
|   gr <= -0.191208: 1 (7.0)
|   gr > -0.191208: 3 (104.0/5.0)
ug > 0.663668
|   ri <= 0.285854: 1 (88.0/5.0)
```

```
|   ri > 0.285854
|   |   ri <= 0.314657
|   |   |   gr <= 0.692108: 2 (6.0)
|   |   |   gr > 0.692108: 1 (3.0)
|   |   ri > 0.314657: 2 (90.0/2.0)
```

The 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 97 of 99 cases. Distinction between stars and galaxies are a bit worse and the algorithm classified 7 galaxies incorrectly as stars and 7 stars were confused with galaxies. Five stars were incorrectly classified as QSO.

```
  s  g  q   <-- classified as
 88  7  5 |  s
  7 92  0 |  g
  2  0 97 |  q
```

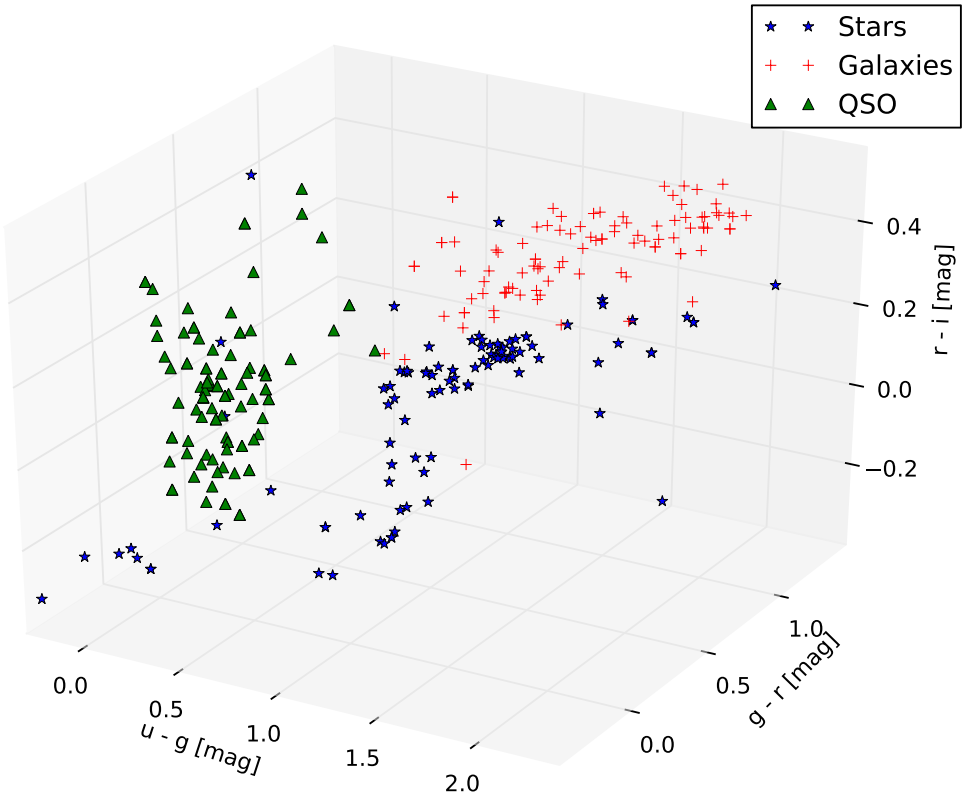


Figure 2.2: Color Diagram of the problem. It shows that individual object classes occupies different regions in the diagram

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 became 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

It is the 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 valiation 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

The project DAME (Data Mining & Exploration) is a 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 project is available at: http://dame.dsf.unina.it/

Chapter 3

Be candidates

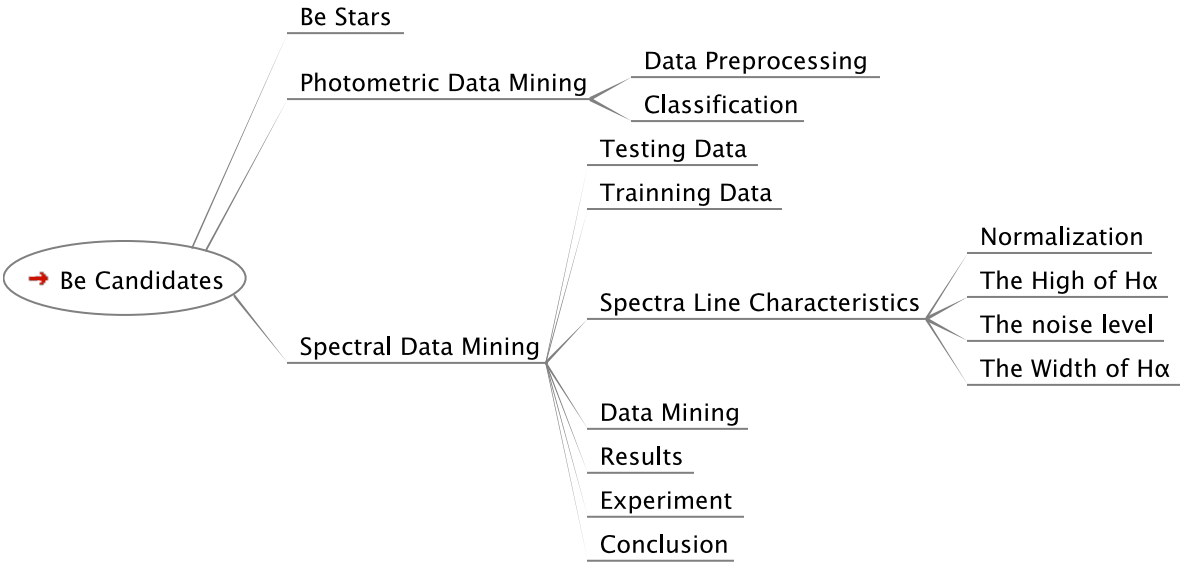


Figure 3.1: Chapter structure

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. The first one utilizes photometric properties of Be stars, the second one their spectra characteristics.

3.1 Be stars

The first example of Be star (γ Cas) 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. The rapidly rotating central star is important feature of these objects, which may be important contributor of the circumstellar medium. It is estimated that about 20 percent of B stars are in fact Be stars. As the definition suggests, the spectra

of Be stars can vary with time. Related to the Be stars are the shell stars: B stars with deep and narrow absorption lines in their spectra, superimposed on the normal broad absorption lines of hydrogen and helium which dominate their spectra. Stars can actually change from B to Be to B-shell and back to B again. Finally, there are variations on time scales of 0.3 to 2 days, which are due to non-radial pulsation, or perhaps rotation. These variations, which occur on or near the surface of the star, may be connected with the formation of the disc around the star [Porter & Rivinius, 2003].

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

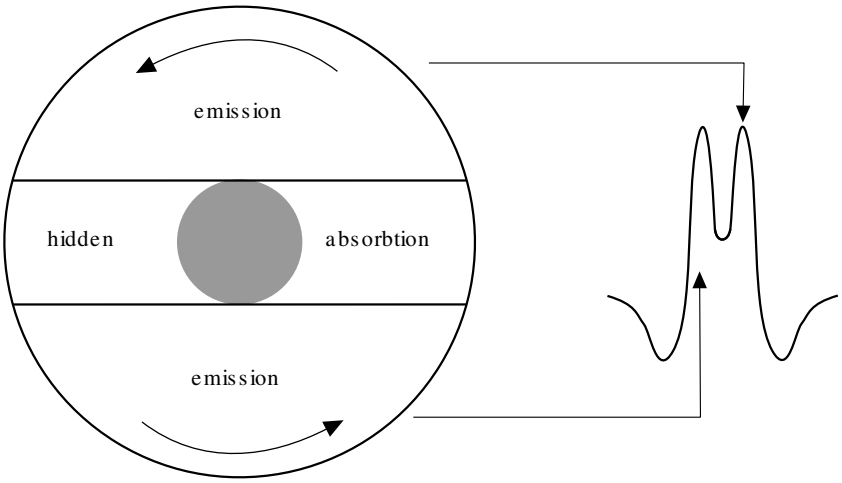


Figure 3.2: A model of a typical Be star. Emission lines coming from an equatorial disk is added to the photo-spheric absorption spectrum. Central B star emits UV (Lyman continuum) and ionizes the disk, which in turn re-emits at high wavelength such as visible domain [Hirata & Kogure, 1984]

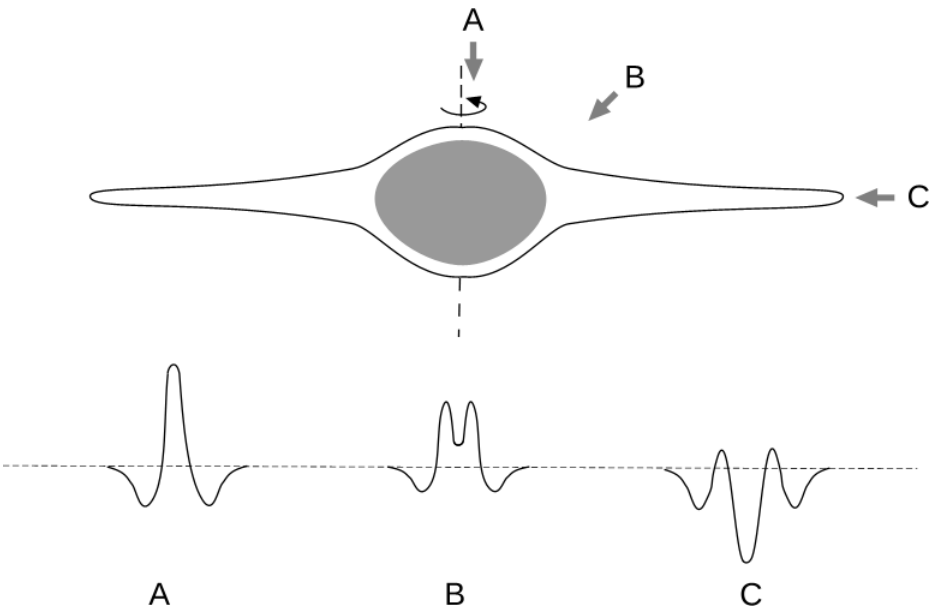


Figure 3.3: Example of spectra of Be stars based on view angle [Slettebak, 1988]

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 digitized surveys and obtaining new candidates could help to answer these questions.

3.2 Photometric Data Mining

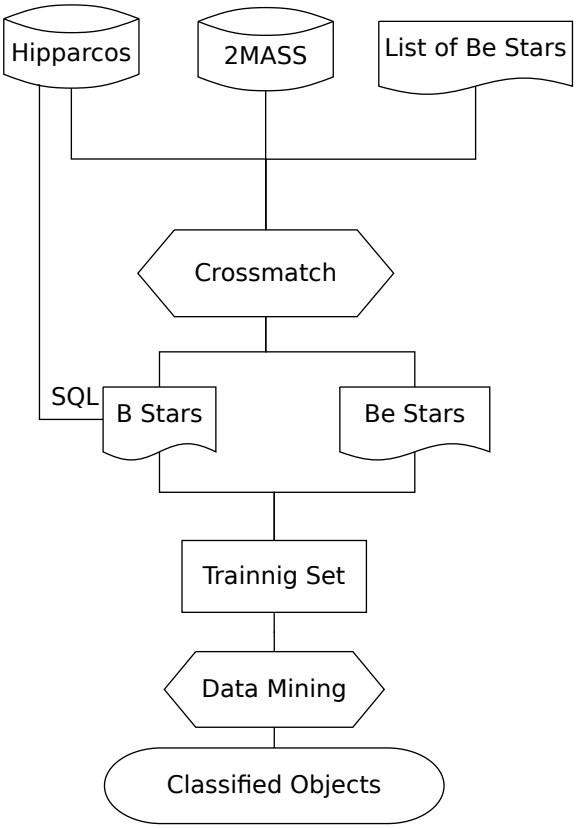


Figure 3.4: A schematic diagram of the photometric Data Mining process. The lists of confirmed Be stars consisted of Hipparcos IDs, this was correlated with Hipparcos catalog to obtain right ascension and declination of the objects and subsequently cross-matched with 2MASS catalog to get photometric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

The classification based on photometric properties is very attractive from several points of view. There are much more available photometric then spectral data and they are more accessible. Because they are easier to acquire 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 exhibit infrared excess correlated to the Hα emission [Van Kerkwijk et al., 1995].

Color indices were computed from 2MASS catalog which uses JHK photometric system. The filters JHK are an important extension of the Johnson system to near-infrared wavelengths.

Filter	Effective wavelength [μm]
J	1.235 ± 0.006
H	1.662 ± 0.009
K_s	2.159 ± 0.011

Table 3.1: JHK photometric system [Cohen et al., 2003]

3.2.1 Data preprocessing

The process is depicted in figure 3.7. The list of confirmed Be stars is from Astronomical Institute of the Academy of Science in Ondřejov. The list contains Hipparcos IDs [Perryman et al., 1997] of objects carefully checked in scientific papers. Correlation with Hipparcos was done using cross-matching function in STILTS application. The second part of the training set (stars different then Be) were obtained using SQL query against Hipparcos catalog. B stars were used as they are the most similar. J,H,K colors were obtained from 2MASS [Skrutskie et al., 2006] using method of multi-cone search protocol of Virtual Observatory.

```
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 digram of this two sets is in the figure 3.5

The uncertainties were computed for each object using propagation of errors. Equation 3.1 shows example for $m_j - m_h$. The color digram with these error bars is depicted in the figure 3.6. Although the uncertainties are significant certain trends are presented.

$$\delta_{(m_j - m_h)} = \sqrt{\left(\frac{\partial(m_j - m_h)}{\partial m_j}\right)^2 \delta^2_{m_j} + \left(\frac{\partial(m_j - m_h)}{\partial m_h}\right)^2 \delta^2_{m_h}} \tag{3.1}$$

$$\frac{\partial(m_j - m_h)}{\partial m_j} = 1, \frac{\partial(m_j - m_h)}{\partial m_h} = -1 \tag{3.2}$$

$$\delta_{(m_j - m_h)} = \sqrt{\delta^2_{m_j} + \delta^2_{m_h}} \tag{3.3}$$

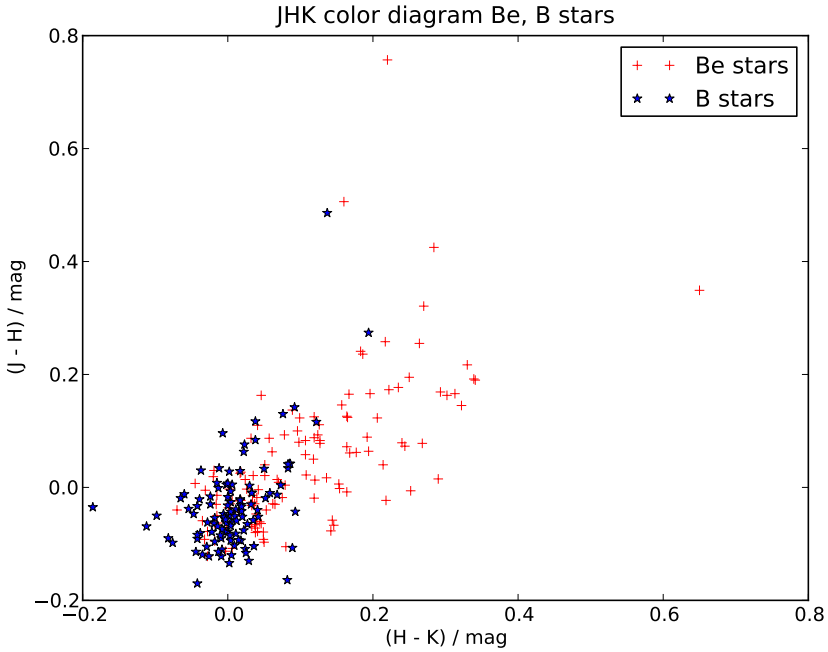


Figure 3.5: Color diagram of confirmed Be stars and B stars

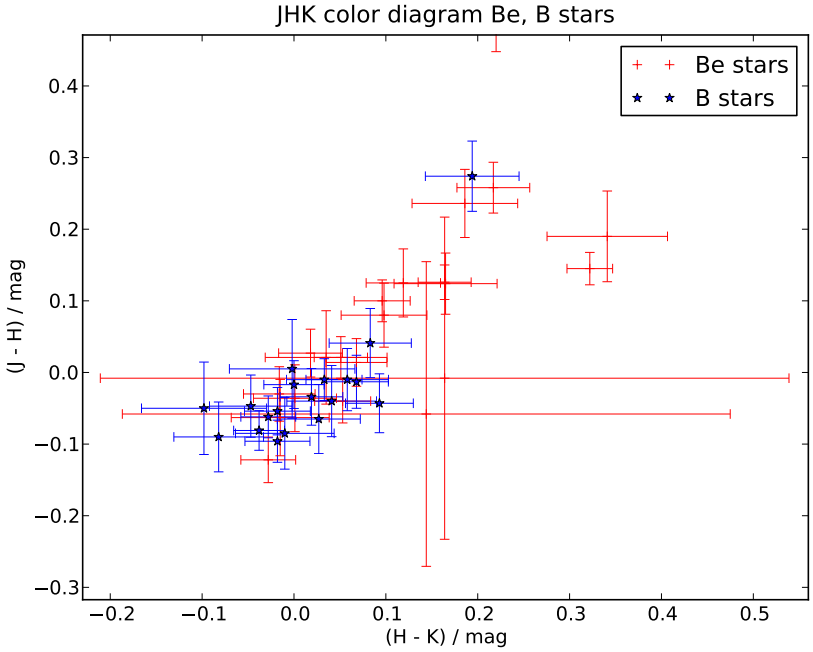


Figure 3.6: Color diagram of confirmed Be stars, B stars with errors

3.2.2 Classification

Data were transformed from original VOTable obtained from Virtual Observatory tools to arff[1] format used in Weka Data Mining system. Algorithm C4.5 (J48) was used to perform actual classification with following result:

```
Correctly Classified Instances     769              73.0989 %
Incorrectly Classified Instances   283              26.9011 %
Kappa statistic                      0.4496
Mean absolute error                  0.3843
Root mean squared error              0.4383
Relative absolute error             79.4985 %
Root relative squared error         89.1648 %
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.

```
  B   Be   <-- classified as
304 126 |  B
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. On the other side, as mentioned in 3.1, about 20 percent of B stars are estimated to be Be stars. For this and other reasons more sophisticated (and much more complicated) approach using spectra analysis was tested.

3.3 Spectral Data Mining

Spectra provide much wider scientific information over photometric properties. Spectral lines exhibit many distinguished features and astronomers have long tradition in analysing their properties. On the other hand it is much more 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 spectra from 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.

```
SELECT objid,dbo.fGetUrlFitsSpectrum(s.specObjID)
FROM SpecPhotoAll s, platex p
WHERE s.specObjID is not null
AND s.plateid = p.plateid
AND p.programname LIKE 'segue%'
AND specClass = 1
```

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

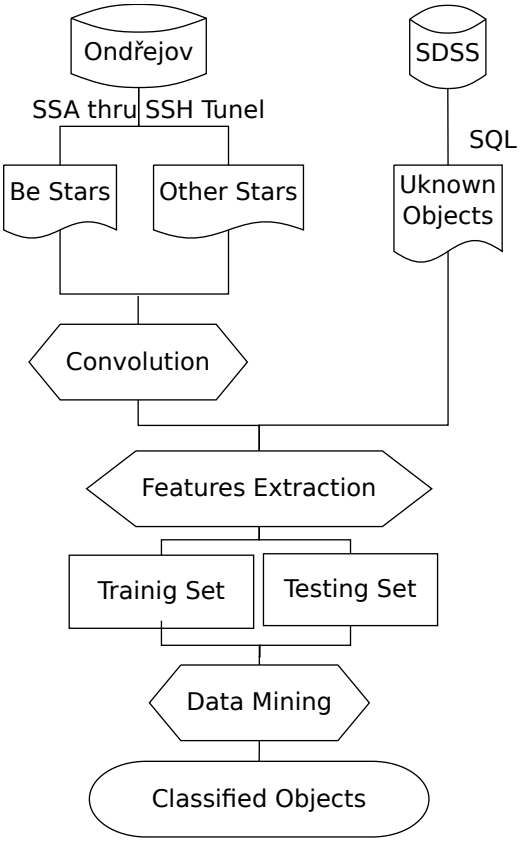


Figure 3.7: A schematic diagram of the spectral data mining process. Using SSA protocol the spectra from Ondřejov server were acquired based on the list from photometric study. Convolution with SDSS instrumental profile had to be performed to ensure compatibility with SDSS. Afterwards the desired features were extracted automatically from the spectra after the continuum normalization and Hα line was fitted by appropriate function. The same was done for spectra from SDSS except the convolution process

3.3.2 Training Data

The spectra obtained with coudè spectrograph of Ondřejov Observatory 2m telescope were used as a training sample. Files were downloaded using SSA protocol. The SSA server is not publicly accessible outside of the local network of Ondřejov observatory. That is why the SSH tunneling of HTTP protocol 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.

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

The script generates the following output. The same process was used later for obtaining the sample of non-Be stars.

```
http://tvoserver/coude/..._ssa&REQUEST=queryData&POS=83.113,-65.582&SIZE=60
http://tvoserver/coude/..._ssa&REQUEST=queryData&POS=162.537,148.333&SIZE=60
http://tvoserver/coude/..._ssa&REQUEST=queryData&POS=19.907,-73.502&SIZE=60
```

3.3.2.1 Degradation of Spectral Resolution

Because spectra from Ondřejov Observatory have higher spectral resolution than SDSS, the degradation of spectral resolution was applied on them. Wavelength for spectra from Ondřejov Observatory is computed using formula:

$$\lambda = \lambda_0 + x\Delta\lambda_{OND} \tag{3.4}$$

and for SDSS spectra by equation:

$$\lambda = 10^{\lambda_0 + x\Delta\lambda_{SDSS}}, \tag{3.5}$$

where λ_0 is value of parameter CRVAL1 and $\Delta\lambda$ of CD1_1. Obtaining them from FITS file is easy using PyFITS[1] module:

```
  In [1] import pyfits as pf
  In [2]: hdu = pf.open('sdss_test.fit')
  In [3]: hdu[0].header['CD1_1']
```

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

```
Out[3]: 0.0001 # SDSS spectrum
Out[3]: 0.2567 # Ondrejov spectrum
```

From equations 3.4 and 3.5 the ratio of spectral resolutions was computed:

$$\frac{\Delta\lambda_{SDSS}}{\Delta\lambda_{OND}} \doteq 3.87 \tag{3.6}$$

Based on this computation 4 pixels of Ondřejov spectra were binned into one to conserve the number of pixels in extracted spectral range. This is the critical part of the binning program:

```
def convolution(f, g):
   """ Convolve two functions"""
   fg = np.convolve(g,f,'same')
   return fg
def reduce(x,y,bin):
   """ Reduce bin pixel into 1"""
   size = x.size/bin
   l = 0
   xx = x[:x.size-1:bin]
   yy = list()
   for i in range(0,size):
       s = 0
       for j in range(0,bin):
           s = s + y[l]
           l+=1
       yy.append(s/bin)
   return xx, yy
```

Prior to binning pixels the convolution with function representing the instrumental profile of SDSS spectrograph was performed. As its first order approximation the Gaussian function with $\sigma = 4$ pixels was used:

$$f(x) = \mathrm{e}^{-\lambda^2/2\sigma^2} \tag{3.7}$$

Convolution is defined:

$$(f * g)(\lambda) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(\lambda - \tau)\, d\tau \tag{3.8}$$

Here it was used in it's discrete form:

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m]\, g[n - m] \tag{3.9}$$

The figure 3.8 shows the result.

3.3.3 Spectral Lines Characteristics

As parameters for data mining process characteristic values of Hα line were extracted from the spectra. Many possible characteristics from different fitting functions through wavelet coefficients to eigenvalues were discussed with experts. Three parameters were finally selected. The height and the width of the Hα emission line and median absolute deviation as a characterization of the noise level in the spectrum.

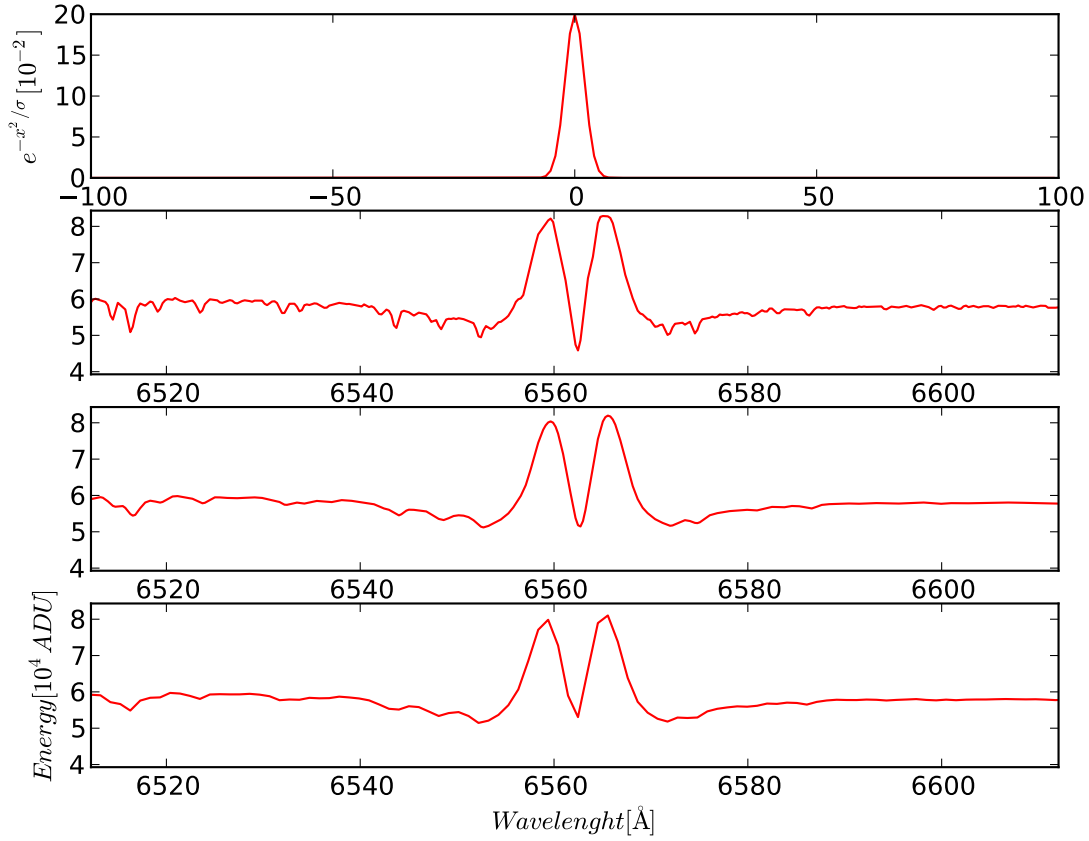


Figure 3.8: Reduction of spectral resolution of Ondřejov spectra of the Be star 4 Her. The top figure shows Gaussian function used for convolution with the spectrum, followed by the original spectrum then there is a spectrum after convolution with the Gaussian profile. The last is the final spectrum after binning

3.3.3.1 Continuum normalization

Spectra from SDSS are absolute flux calibrated, while the Ondřejov spectra state intensity only in ADUs. To be able to compare the size and shapes of spectral line profiles the so called rectification of spectra was performed. The spectra were divided by the linear function roughly representing its (pseudo)continuum. 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. To ensure the compatibility for data mining process only the spectral range covered by Ondřejov spectra ($6300 - 6800\mathring{A}$) was considered in SDSS spectra.

$$\frac{\partial}{\partial q_j} \sum_{i=1}^{n} [y_i - \mathrm{f}(q_j, x_i)]^2 = 0, \qquad j = 0, 1 \tag{3.10}$$

where

$$\mathrm{f}(x) = q_1 x + q_0 \tag{3.11}$$

3.3.3.2 The height of the Hα line

The maximum value in the region of 50 Å around Hα above the linear fit was extracted from the spectrum.

```
def getMax(x,y,line,range):
    """ Return maximum value of range in the spectrum"""
    xrange = x[(x < line + range) & (x > line - range)]
    yrange = y[(x < line + range) & (x > line - range)] - 1
    maximum = yrange.max()
    minimum = yrange.min()
    if abs(maximum) > abs(minimum):
        extrem = maximum
    else:
        extrem = minimum
    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:

$$\mathrm{mad} = \mathrm{median}_i \left(\, |X_i - \mathrm{median}_j(X_j)| \, \right) \tag{3.12}$$

3.3.3.4 The width of the Hα line

The Gaussian function:

$$f(x) = 1 + e^{-\frac{(x-x_0)^2}{S^2}} \tag{3.13}$$

was fitted to the profile of Hα spectral line. First the robust estimators [Launer, 1979] were computed and used as input parameters for leastsq[1] method from scipy.opt module, which minimize the sum of squares.

$$x_0 = \frac{\text{median}(w_j x_j)}{\sum w_i}, \tag{3.14}$$

$$S = \frac{\text{mad}(x_i - x_0)}{\sum w_i}. \tag{3.15}$$

Part of the script implementing fitting the Gaussian function

```
x0 = np.median(sum(w*x))/sum(w)
S = sum(w*mad((x - x0)))/sum(w)
params = np.array([1, maximum, x0, S], dtype=float)
fit, flag = opt.leastsq(residuals, params, args=(yrange, xrange))
gauss = model(xrange, fit) + 1

def model(t, coeffs):
    return coeffs[0] + coeffs[1] * np.exp( - ((t-coeffs[2])/coeffs[3])**2 )
def residuals(coeffs, y, t):
    return y - model(t, coeffs)
```

Final results are in the figure 3.9 and 3.10. The script was adjusted to work with SDSS and Ondřejov spectra. The whole procedure was performed on all of the cca 200 thousand SDSS spectra and almost 200 Ondřejov spectra resulting in the ASCII files with the characteristic values used later in data mining process. The training set contains 173 confirmed Be stars.

3.3.4 Data Mining

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

```
@RELATION STAR-B-BE
@ATTRIBUTE name STRING
@ATTRIBUTE alpha NUMERIC
@ATTRIBUTE grp {be,o}
@DATA
10_cas,-0.822196556626,be
11_cyg,1.68689566629,be
```

```
@RELATION STAR-B-BE
@ATTRIBUTE name STRING
@ATTRIBUTE alpha NUMERIC
@ATTRIBUTE grp {be,o}
@DATA
spSpec-53228-1884-001 -0.584628294569 ?
spSpec-53228-1884-002 -0.877184482566 ?
```

[1] "leastsqi"s a wrapper around MINPACKs lmdif and lmder algorithms.

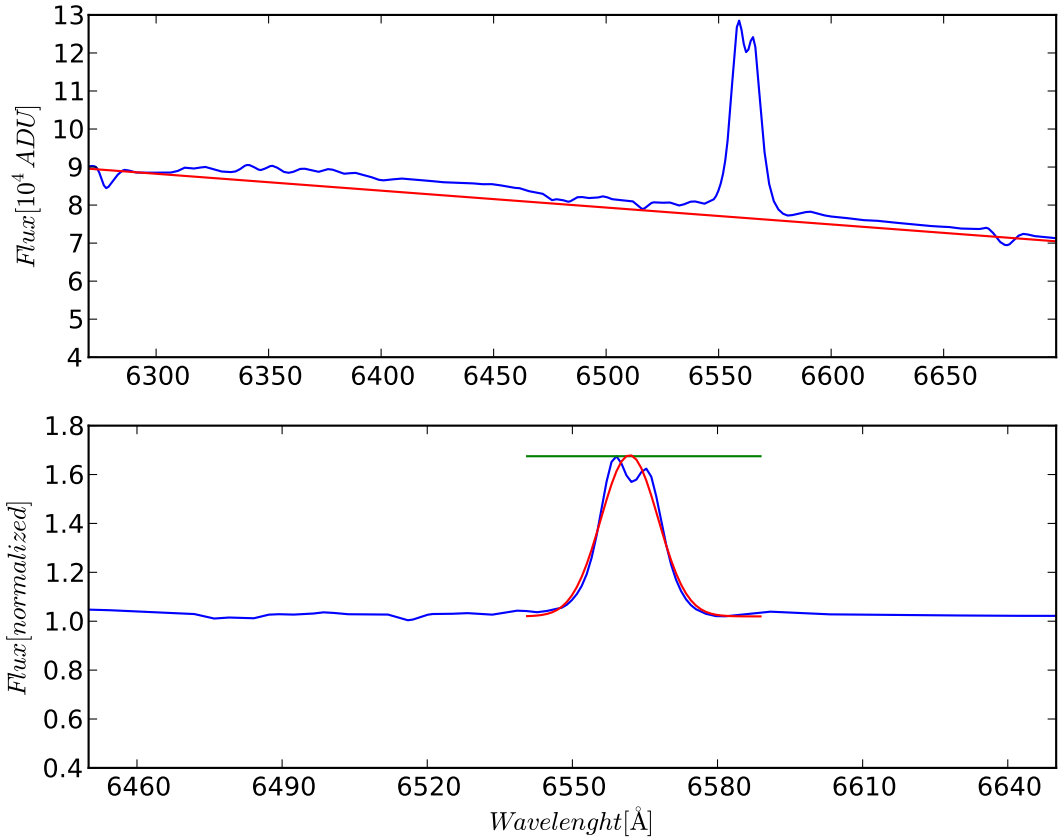


Figure 3.9: Normalized spectrum of Be star 60 Cyg. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line corresponds to the maximum value in the region of 50 Å. The Gaussian fit is in red. Although the fit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

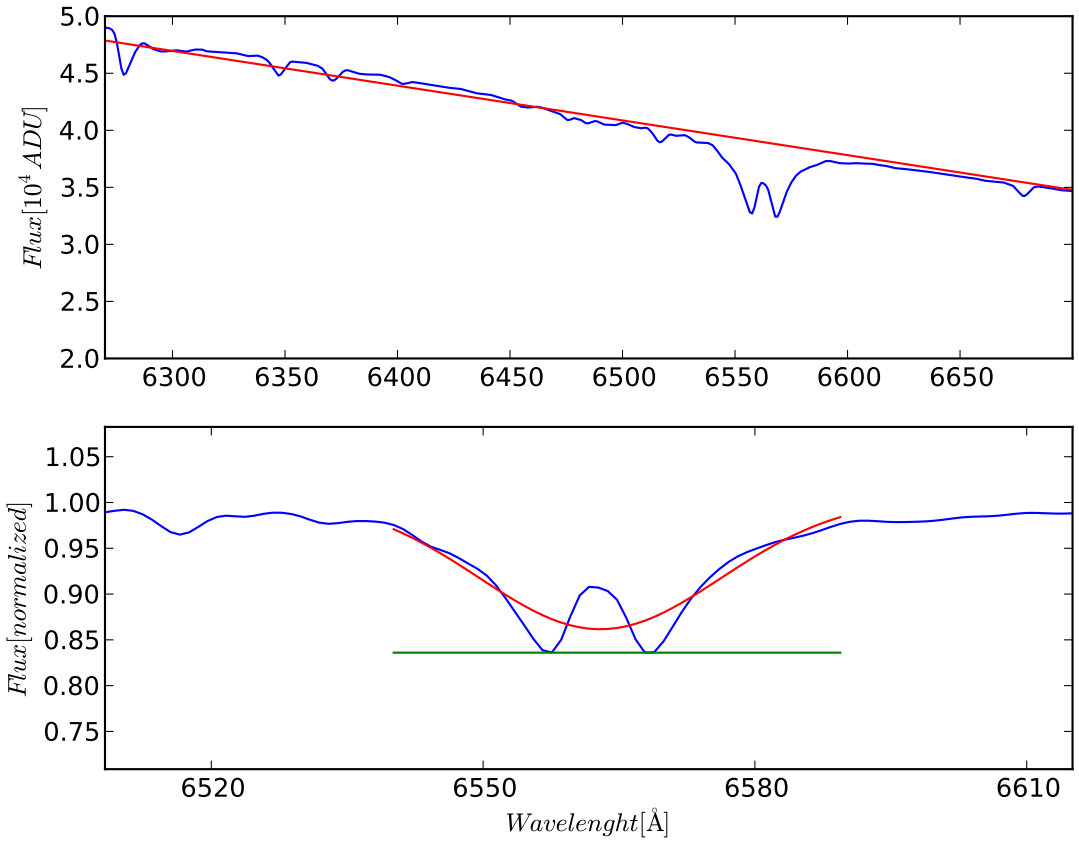


Figure 3.10: Normalized spectrum of Be star 17 Tau. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line corresponds to the maximum value in the region of 50 Å. The Gaussian fit is in red

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.

```
java -classpath weka.jar
weka.classifiers.meta.FilteredClassifier -F
weka.filters.unsupervised.attribute.RemoveType -W
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.

```
=== Summary ===
Correctly Classified Instances         145               83.815 %
Incorrectly Classified Instances        28               16.185 %
Kappa statistic                          0.6529
Mean absolute error                      0.1849
Root mean squared error                  0.3652
Relative absolute error                 39.8819 %
Root relative squared error             75.8919 %
Total Number of Instances              173
```

The classification tree shown below is relatively complicated. But still we can learn a few things. It is using all of the parameters put in so they are chosen correctly (if they were irrelevant, the classifier would not use them). The most important parameter was max which determines the height of the line above the continuum. This was expected as Hα emission is dominating 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 needs to be emphasized the above mentioned parameters are only simplified description of the real physical shape of the line profile, though they can give us some physical insight of the studied phenomena. Decision trees are therefore very powerful compared to black box approaches such as Neural Networks, where the classification process is beyond human understanding.

```
J48 pruned tree
------------------
max <= -0.18843
|   max <= -0.324763: o (46.0/5.0)
|   max > -0.324763
|   |   max <= -0.255475
|   |   |   mad <= 0.004133: o (2.0)
|   |   |   mad > 0.004133: be (13.0/1.0)
|   |   max > -0.255475
|   |   |   mad <= 0.009862: o (10.0)
|   |   |   mad > 0.009862
|   |   |   |   width <= 7.621593: o (3.0/1.0)
|   |   |   |   width > 7.621593: be (2.0)
max > -0.18843
|   mad <= 0.030316
|   |   max <= -0.091726
|   |   |   width <= 5.286489
```

```
|   |   |   |   max <= -0.170022: be (2.0)
|   |   |   |   max > -0.170022: o (3.0)
|   |   |   width > 5.286489: be (9.0)
|   |   max > -0.091726: be (76.0)
|   mad > 0.030316
|   |   max <= 6.917615: o (4.0)
|   |   max > 6.917615: be (3.0)
```

```
 === Confusion Matrix ===
Be Others  <-- classified as
95 15   | Be
13 50   | Others
```

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

Spectra of some of the SDSS objects classified as Be stars are presented in the figures 3.11,3.12,3.13 and 3.14. The vertical dashed line corresponds to laboratory wavelenght of Hα. These samples represent tiny fraction of the complete result. The program for generating web pages with thumbnails was created and full results 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 3.2: Examples of SDSS Be candidates

#	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 3.3: Links to Be candidates on SDSS Skyserver

For comparison there are spectra of known Be stars given in figures 3.15,3.16,3.17 and 3.18. It is clear that the profile of the Hα line is complex and just one parameter cannot possibly express it's characteristics. More advanced description such as wavelet 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 happen if we had chosen 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:

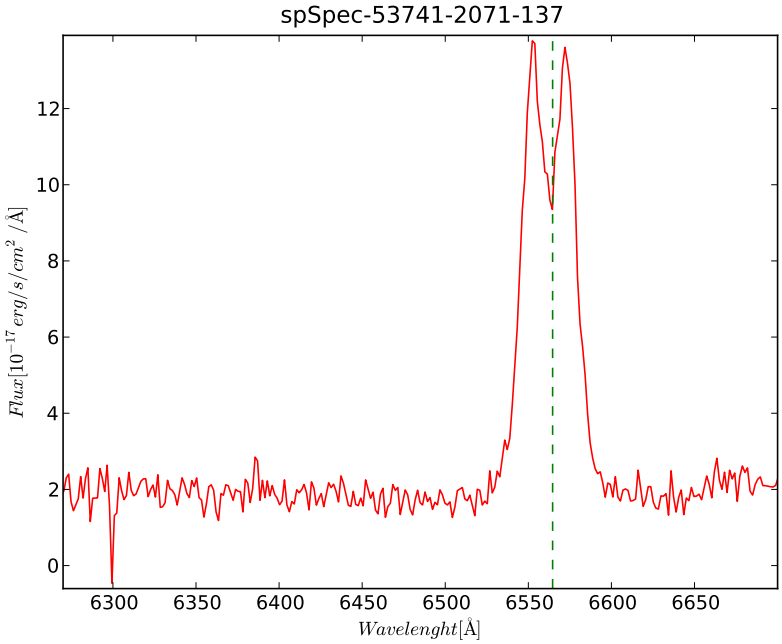


Figure 3.11: Example 1: SDSS J035747.16-063850.7

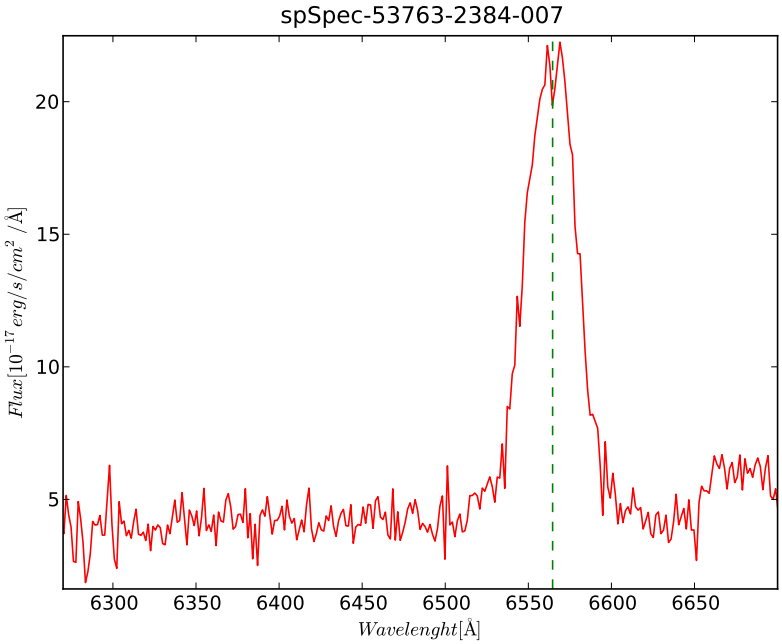


Figure 3.12: Example 2: SDSS J094325.89+520128.6

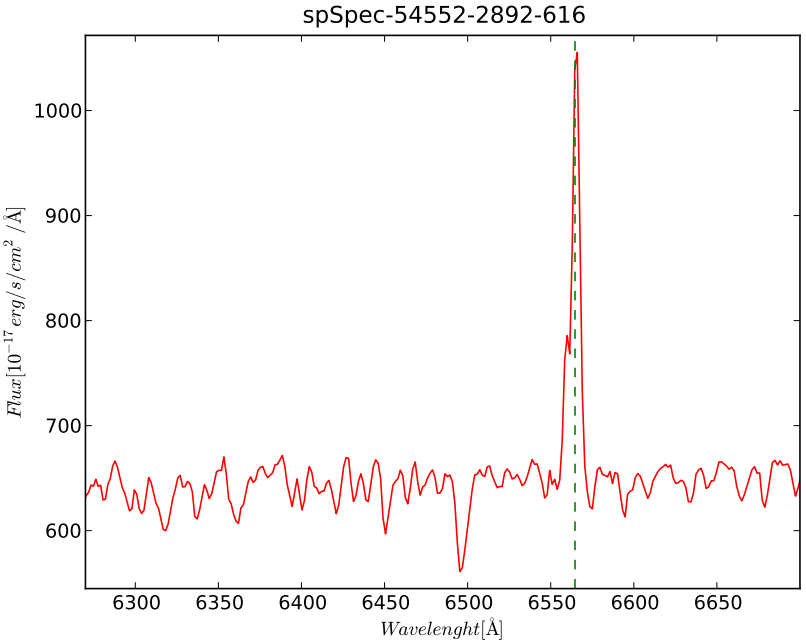


Figure 3.13: Example 3: SDSS J120729.12+003659.8

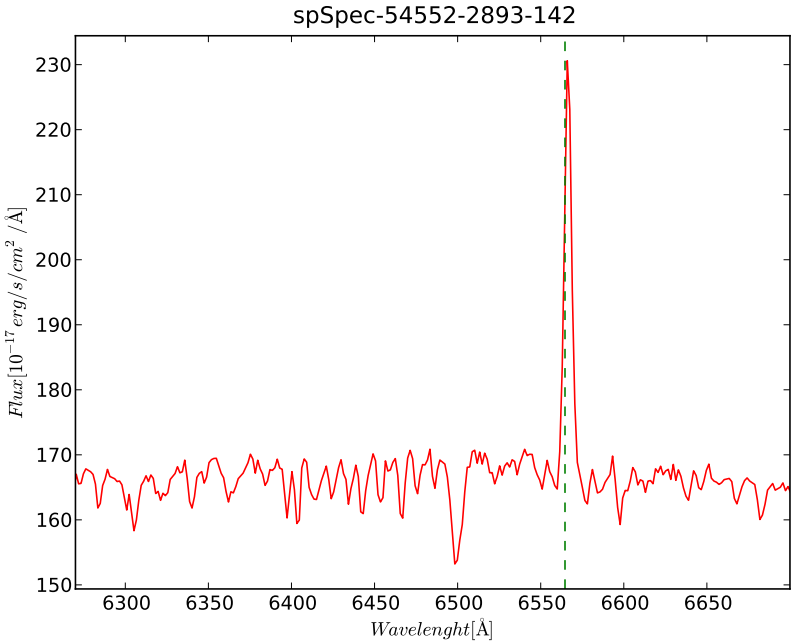


Figure 3.14: Example 4: SDSS J120908.18+194035.8

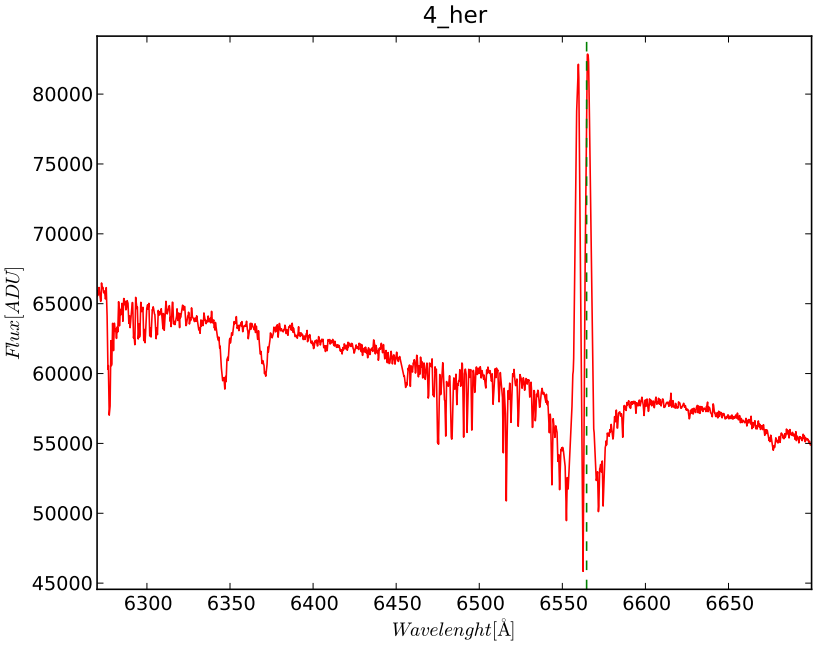


Figure 3.15: Spectrum of 4 Her. Be star

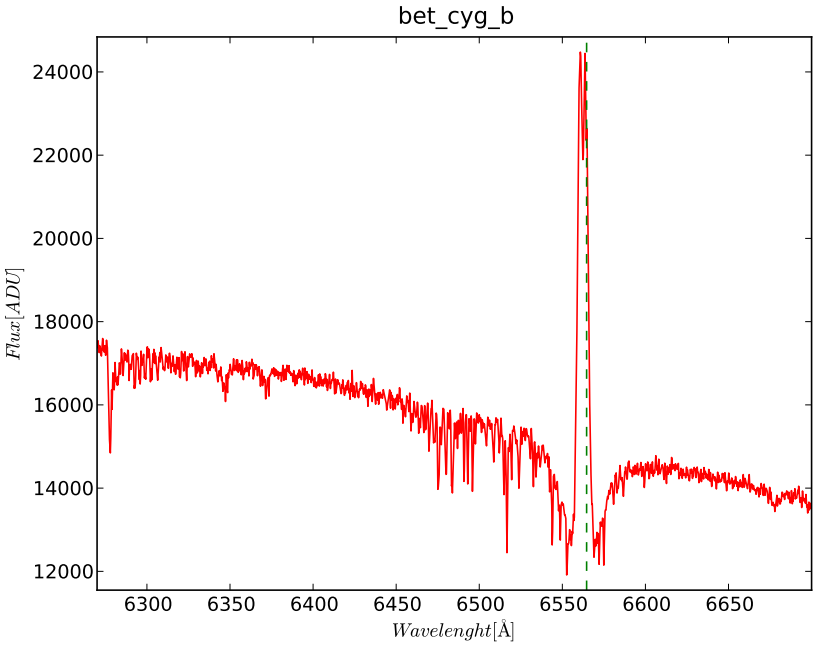


Figure 3.16: Spectrum of HR 7418 (Albireo B). A fast-rotating Be star, with an equatorial rotational velocity of at least 250 kilometers per second. Its surface temperature has been spectroscopically estimated to be about 13.200 K

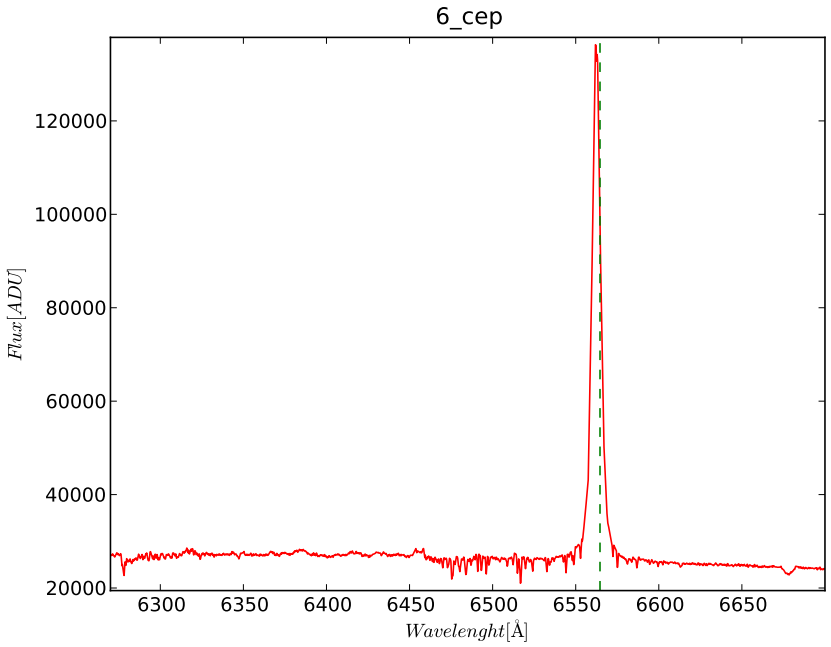


Figure 3.17: Spectrum of 6 Cep. Be star

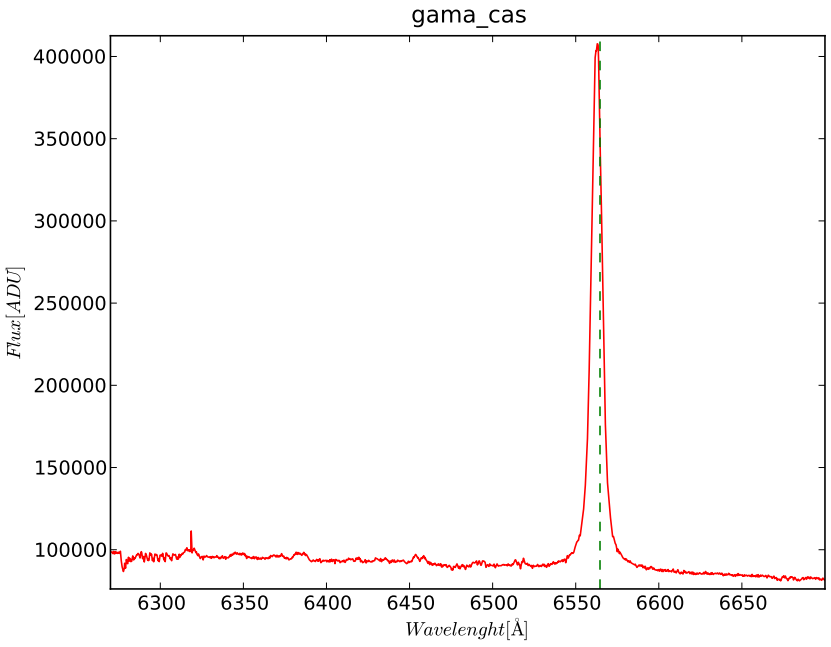


Figure 3.18: Spectrum of Gamma Cas. Be Star

to perform similar tasks 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 a height of emission line and write a program to check the spectra for condition if emission > threshold. Then we do not need "expensive" data mining algorithms to get some interesting results[1]. Lets try to use classification using just the height of the emission line (parameter alpha). Here is consequent tree:

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

We can see that it is not so straightforward. In the training set there are some Be stars with extreme height in the negative direction (the spectrum obtained out of Be phase shows Hα in absorption). We have some statistics from classifier. In this example the effectiveness was 77.6%. Also there is a confusion matrix, 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 other types of stars with Be stars.

```
  === Confusion Matrix ===
 Be     O   <-- classified as
 102    6 |   Be
  35   40 |   Others
```

This experiment proves that using data mining has a sense even in apparently simple cases and can provide non-trivial insight on data examined.

[1] This 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 investigated 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 mining process are reasonable. During the work it was also "discovered" how humans are good at visual judgment: when thumbnails of resulting spectra were created they 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 issues 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 as wavelets, eigenvalues 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.

- More data mining algorithms

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

Originally more advanced approaches such are 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 considered (e.g. IPHAS) but without success.

- Using the time information.

  The whole process presented is static but the "Be phenomena" is dynamic in nature. Using light curves or time series of spectra could significantly improve the efficiency.

- Unknown errors.

  There are things we do not know we do not know. The overall process was very complicated and involved hundreds lines of code. Any mistake overlooked could affect the results. The absence of evidence is not evidence of absence.

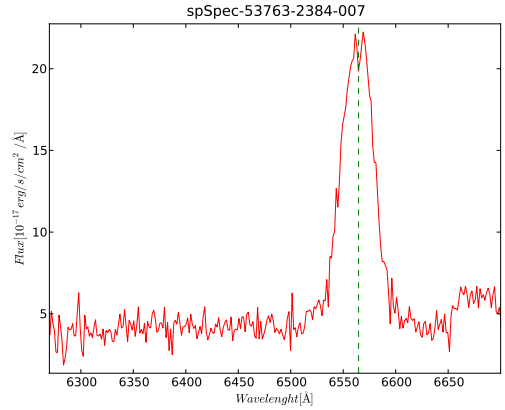
Nevertheless even with these simple parameters some interesting results were achieved:

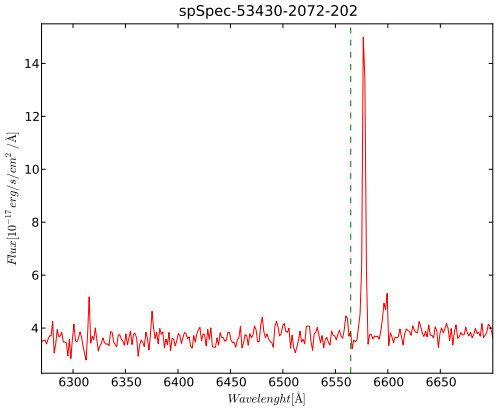
- Semi-automatic process from retrieving data through convolution up to data mining classification was implemented.

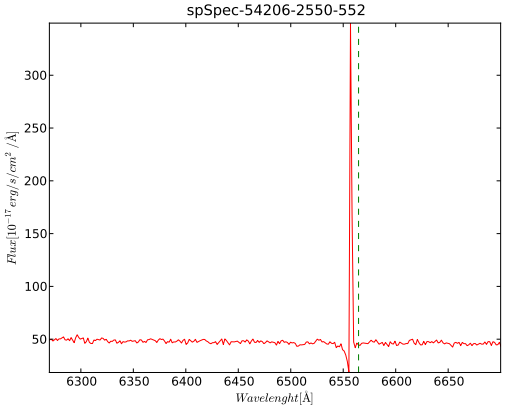
- From the 178314 analyzed spectra 1110 were classified as Be candidates (but there are many imperfect spectra in SDSS). A sample of 46 objects is given in the Appendix 1.

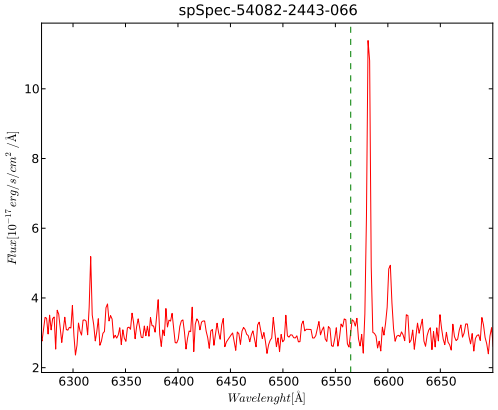
Appendix1: Spectra of Be candidates

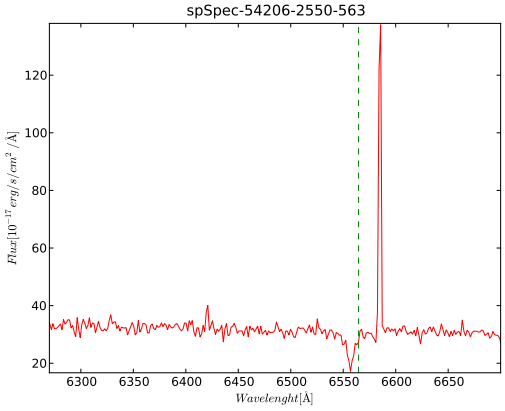
List of 46 objects from SDSS obtained by data mining process described in section 3.3.

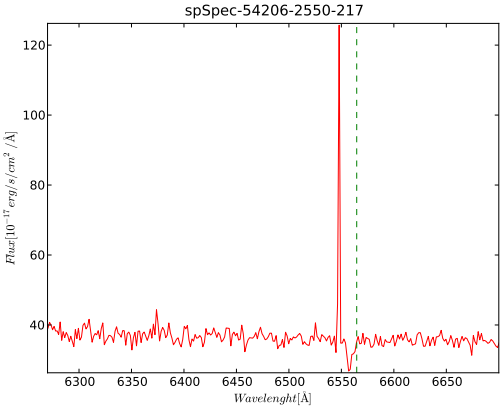




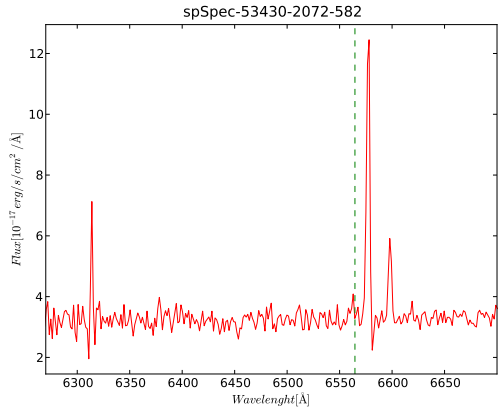


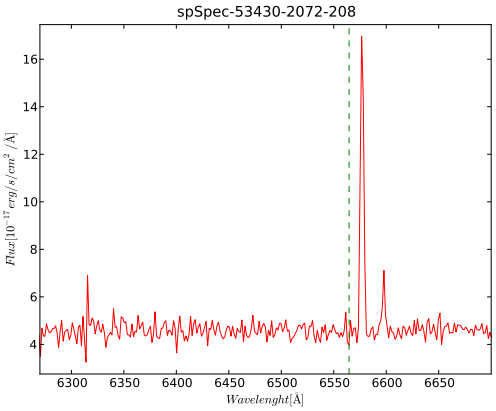


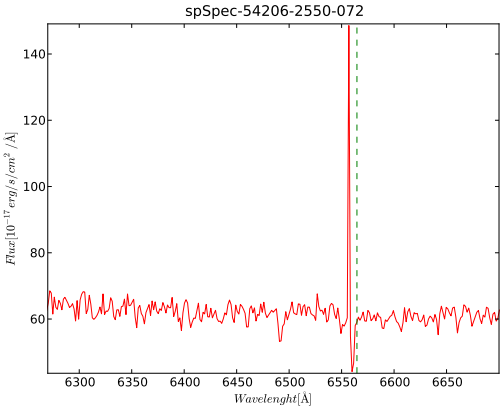


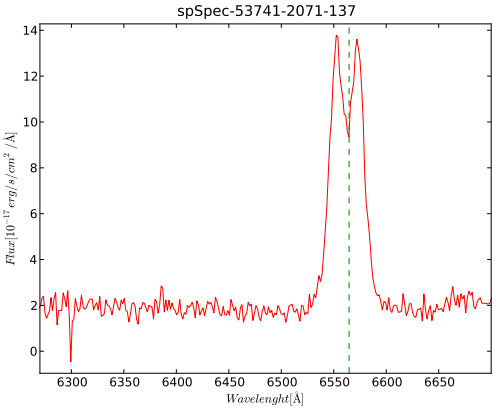


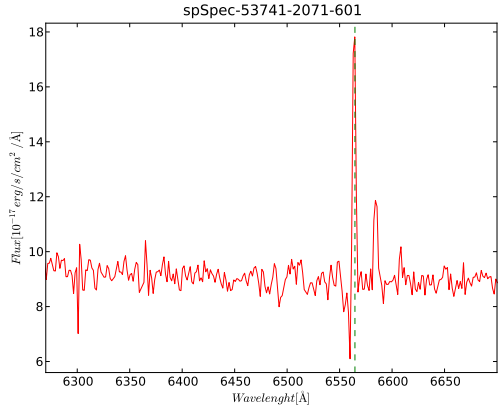
APPENDIX1: SPECTRA OF BE CANDIDATES

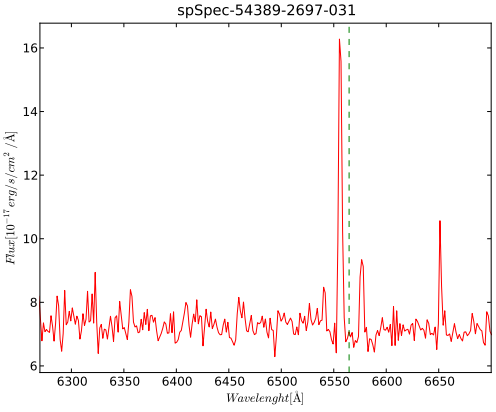


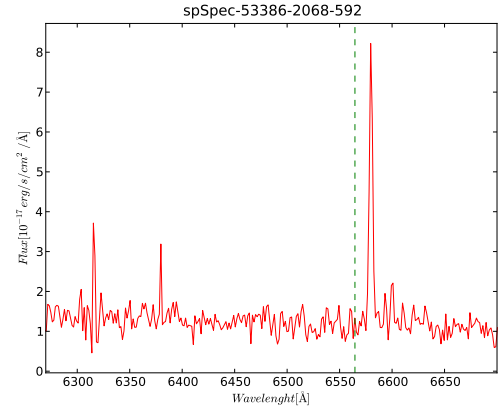


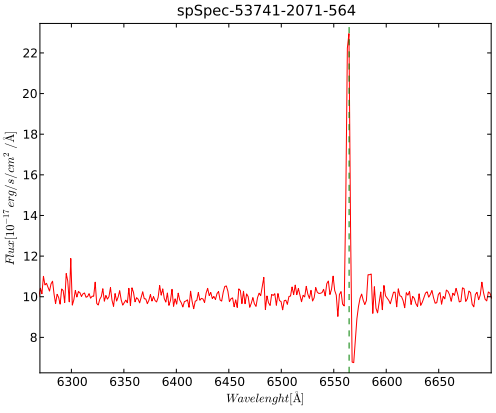


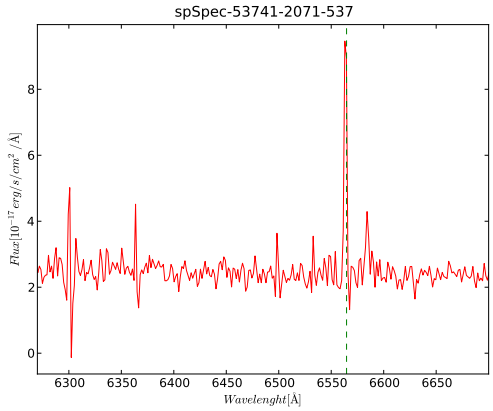


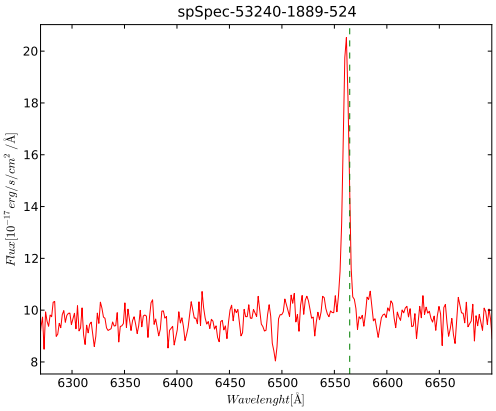


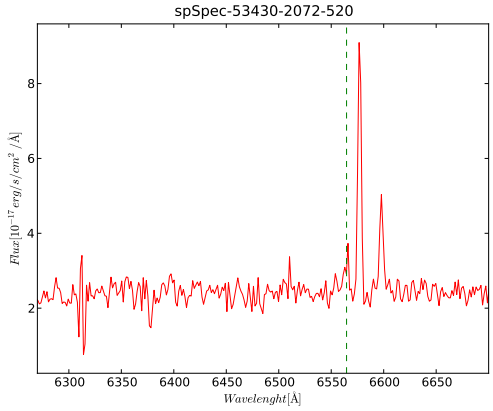


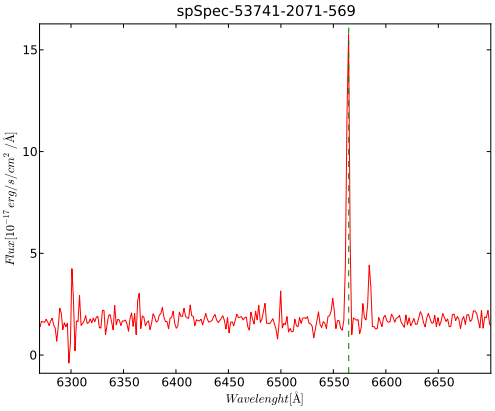


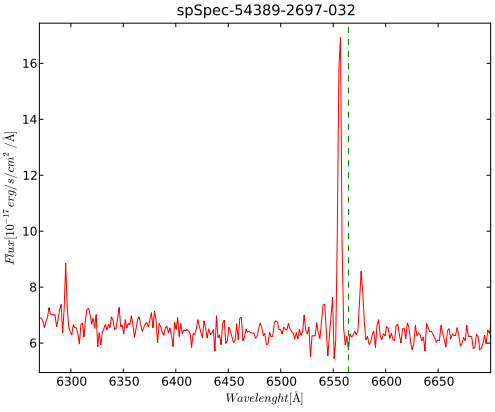


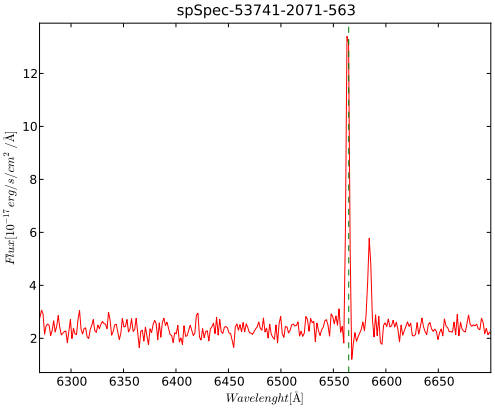

spSpec-53741-2071-537
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

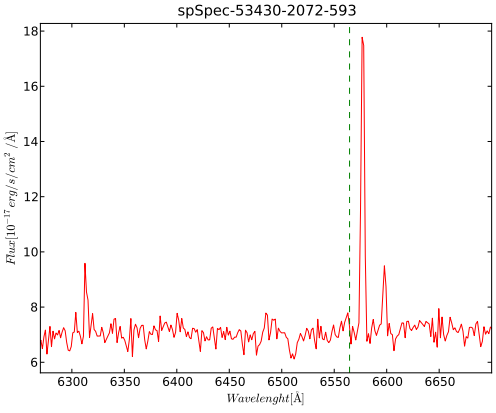

spSpec-53240-1889-524
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

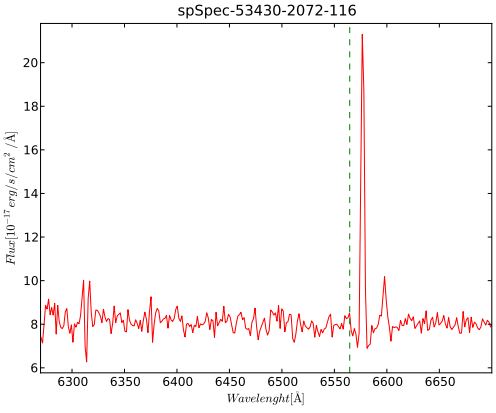

spSpec-53430-2072-520
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]


spSpec-53741-2071-569
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

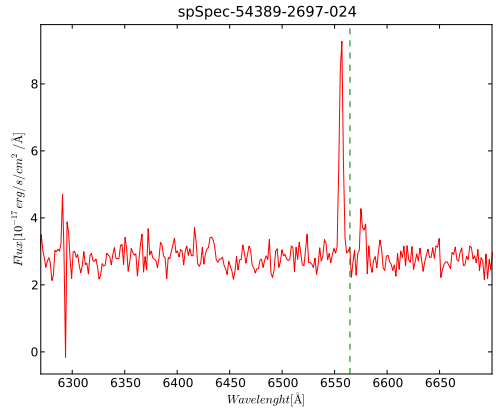

spSpec-54389-2697-032
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

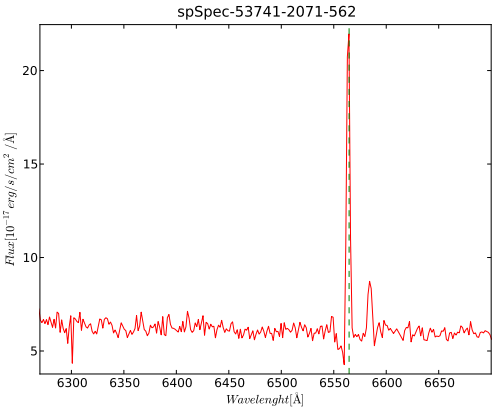

spSpec-53741-2071-563
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

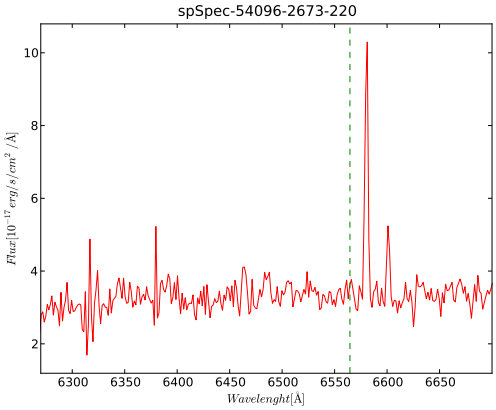

spSpec-53430-2072-593
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

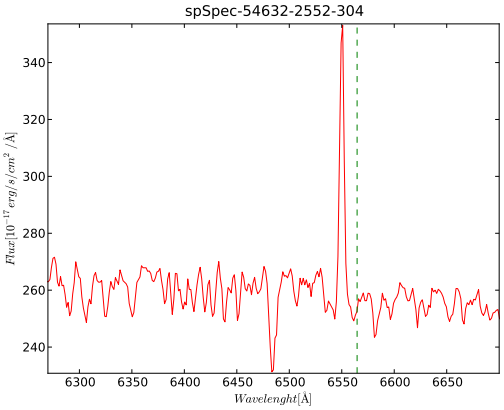

spSpec-53430-2072-116
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

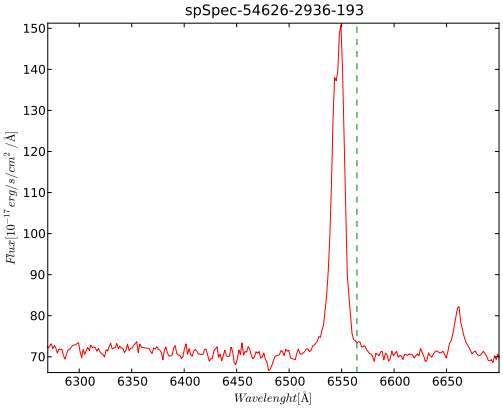
APPENDIX1: SPECTRA OF BE CANDIDATES

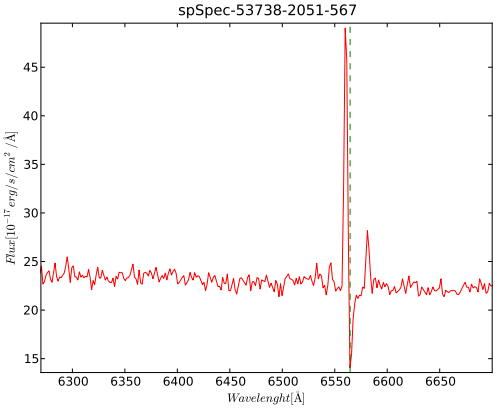


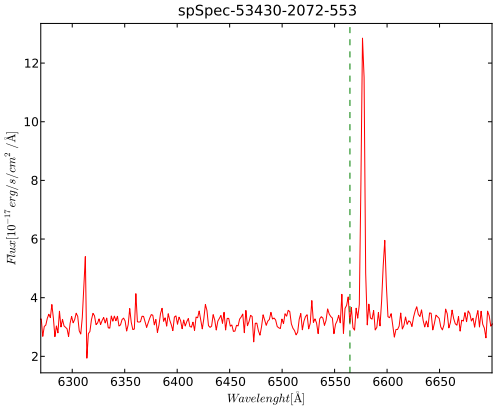


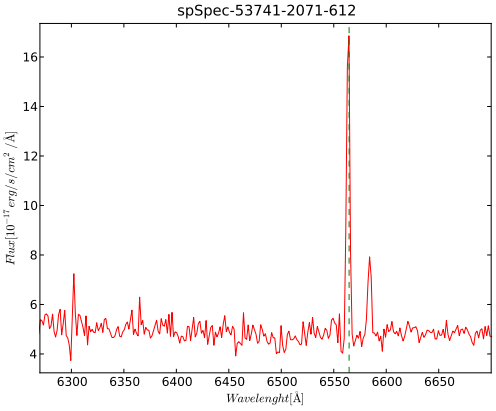


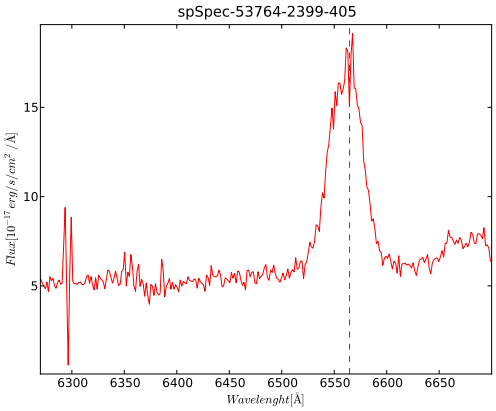


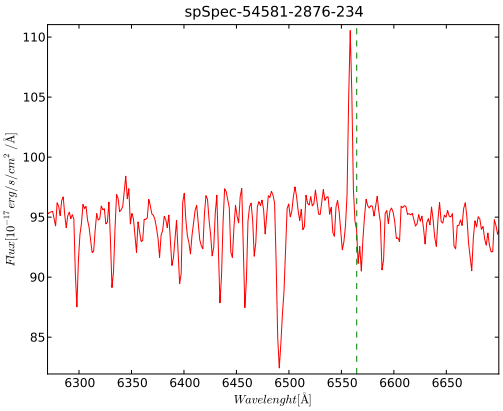


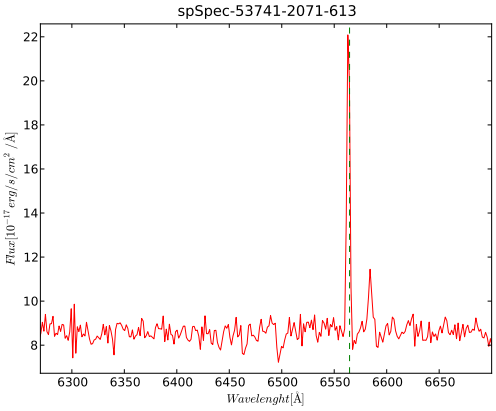


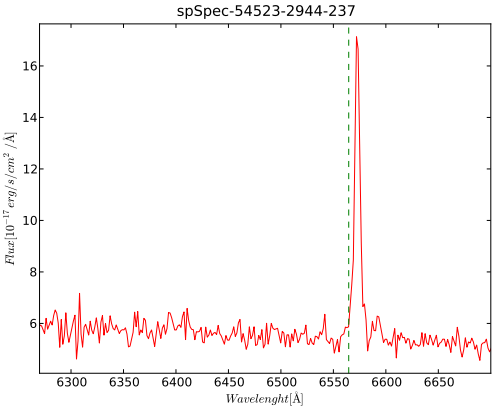


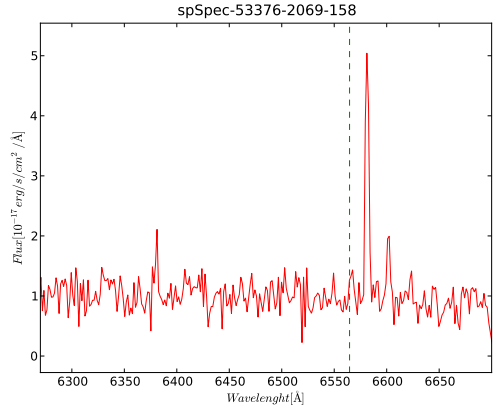


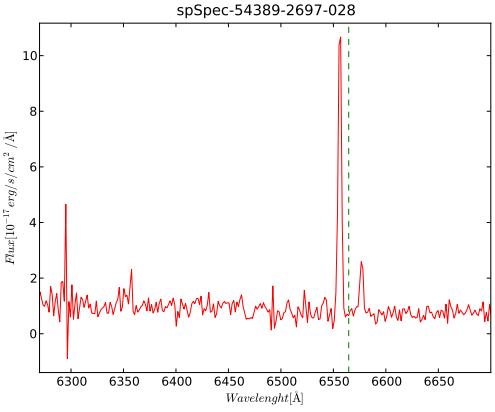

spSpec-53764-2399-405
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

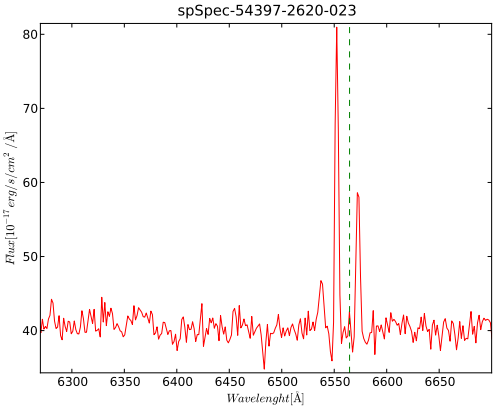

spSpec-54581-2876-234
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

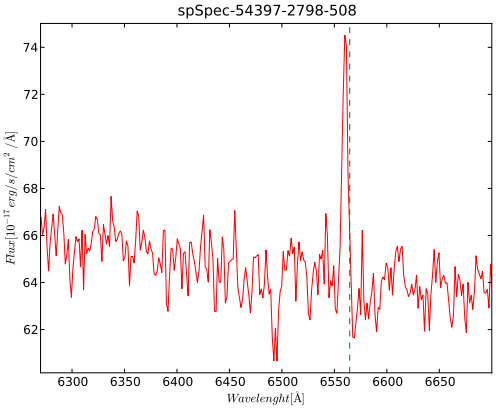

spSpec-53741-2071-613
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]


spSpec-54523-2944-237
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

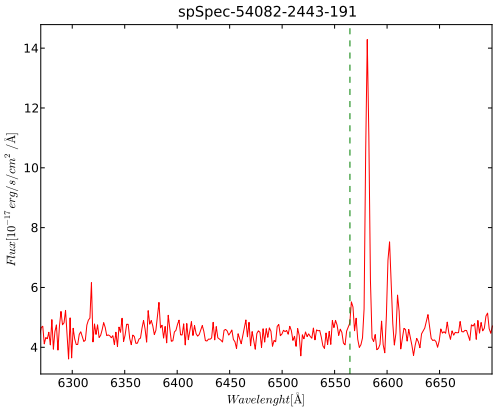

spSpec-53376-2069-158
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

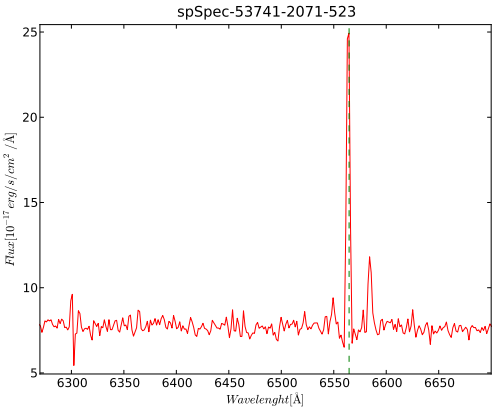

spSpec-54389-2697-028
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

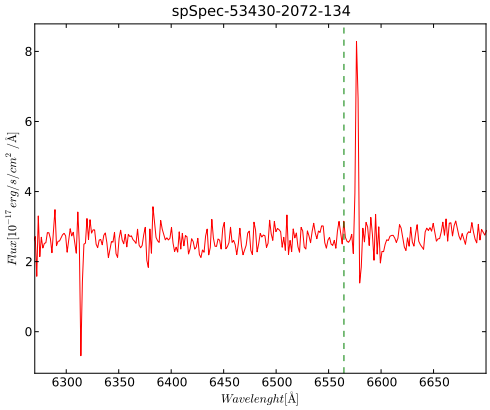

spSpec-54397-2620-023
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

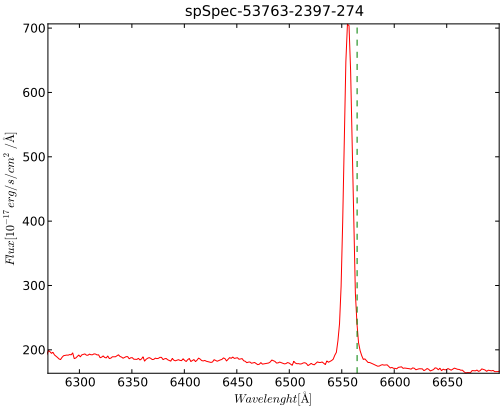

spSpec-54397-2798-508
Flux[10^-17 erg/s/cm^2/Å]
Wavelenght[Å]

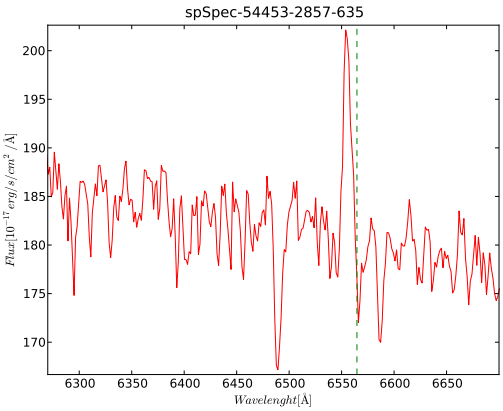
APPENDIX1: SPECTRA OF BE CANDIDATES

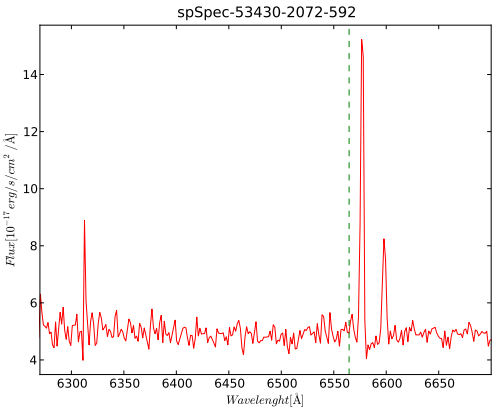


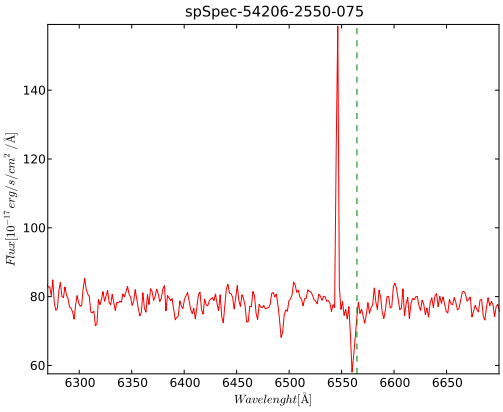


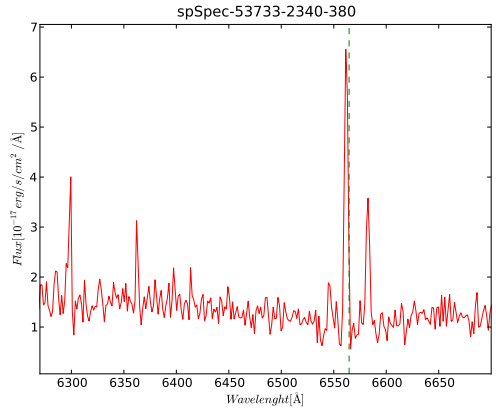






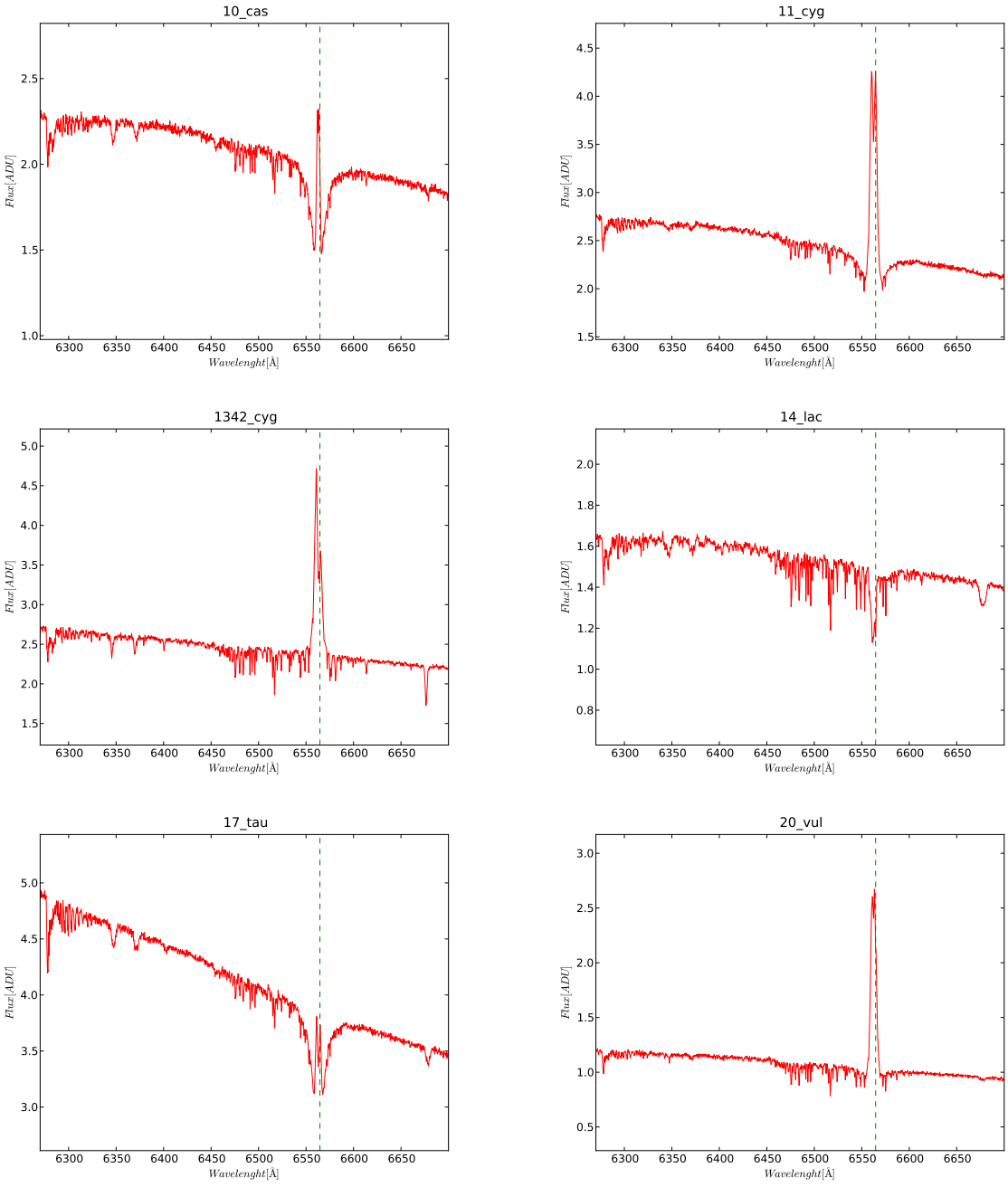




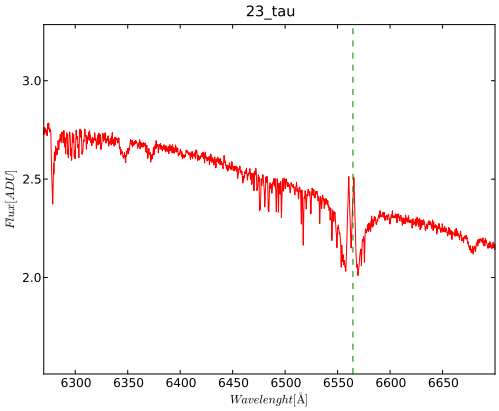


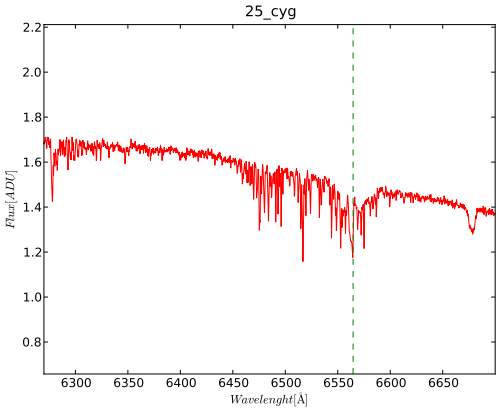
Appendix2: Be stars from Ondřejov

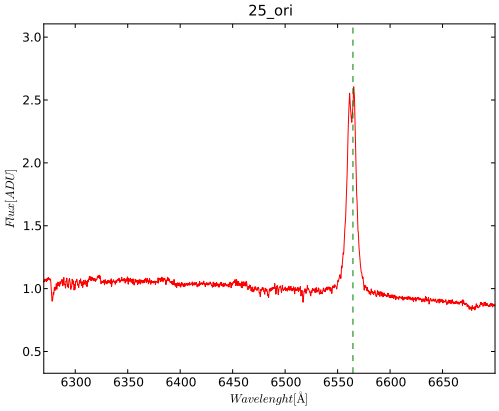
List of 46 objects from Ondřejov used as training set in data mining process described in section 3.3.

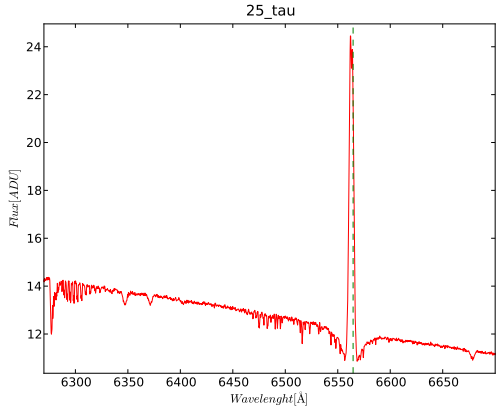


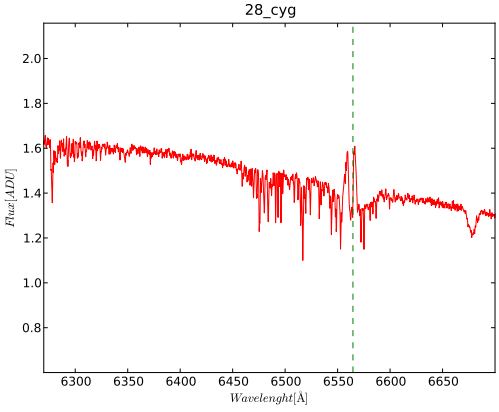
APPENDIX2: BE STARS FROM ONDŘEJOV

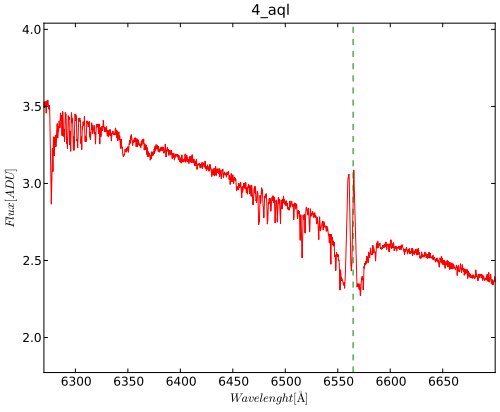


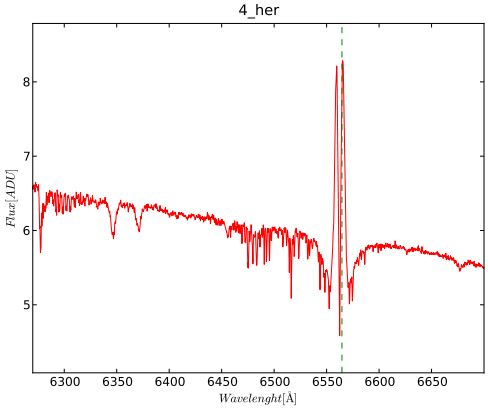


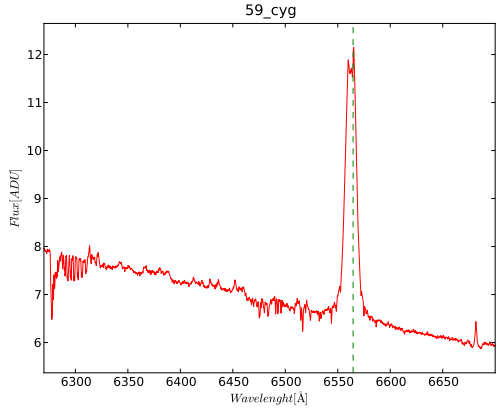


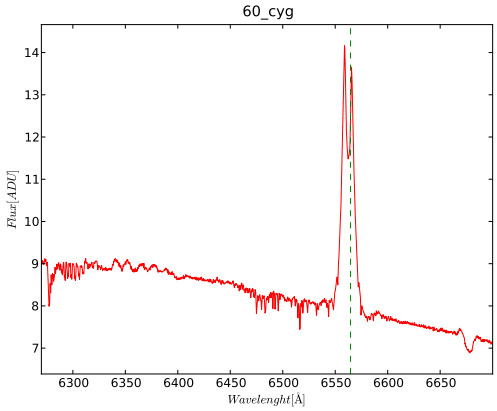


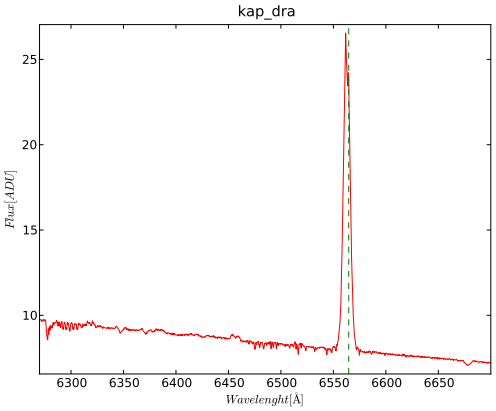


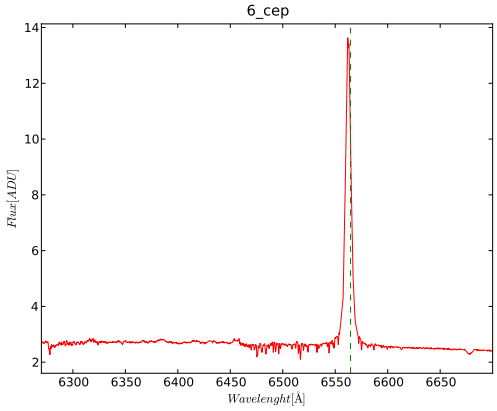


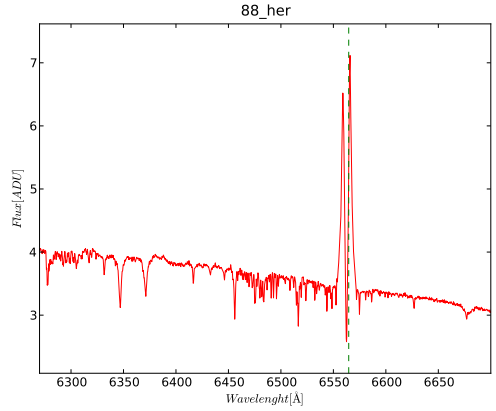


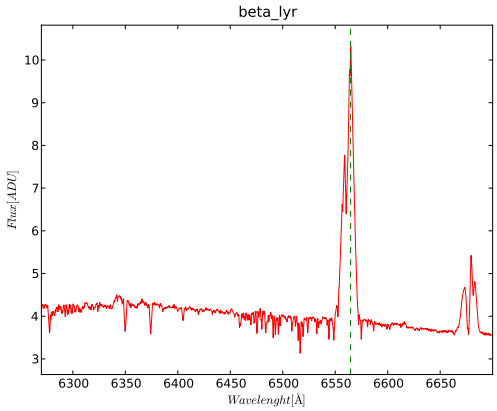


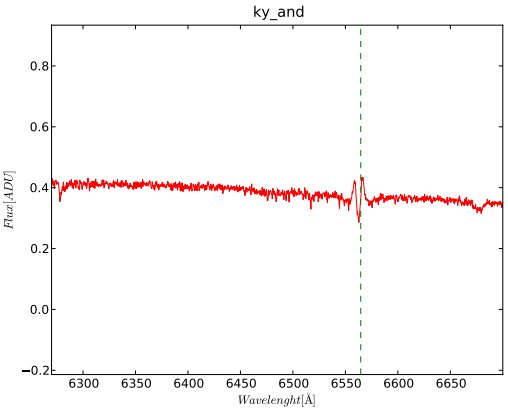

60_cyg
Flux[ADU]
Wavelenght[Å]

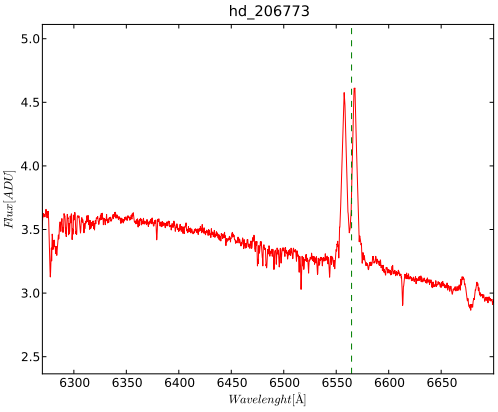

kap_dra
Flux[ADU]
Wavelenght[Å]

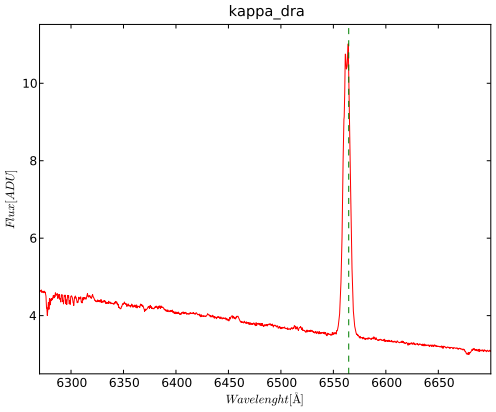

6_cep
Flux[ADU]
Wavelenght[Å]


88_her
Flux[ADU]
Wavelenght[Å]

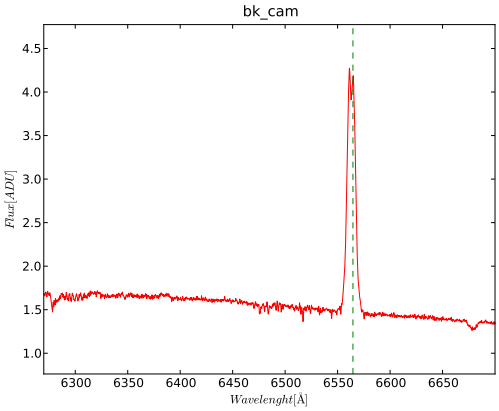

beta_lyr
Flux[ADU]
Wavelenght[Å]

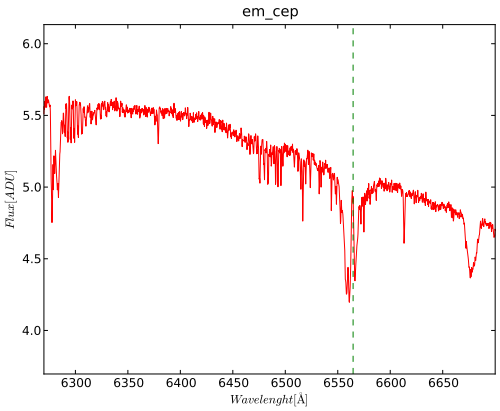

ky_and
Flux[ADU]
Wavelenght[Å]

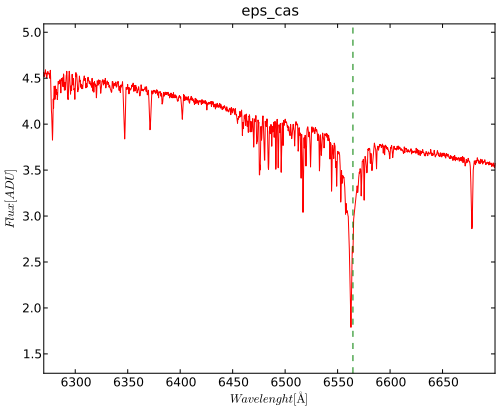

hd_206773
Flux[ADU]
Wavelenght[Å]

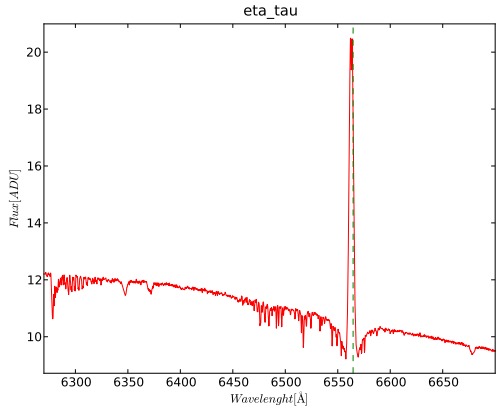

kappa_dra
Flux[ADU]
Wavelenght[Å]

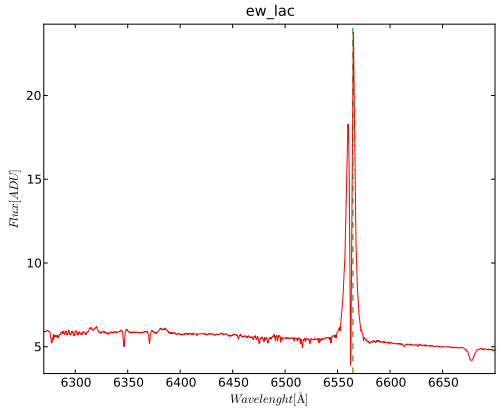
APPENDIX2: BE STARS FROM ONDŘEJOV

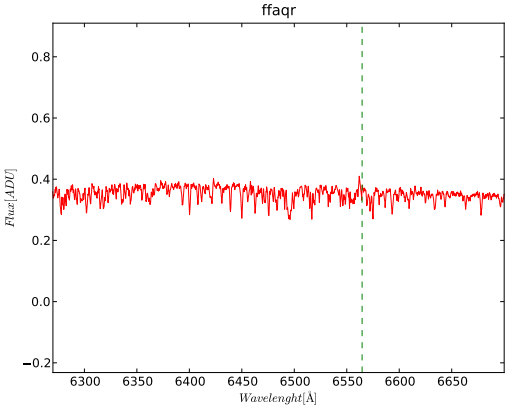


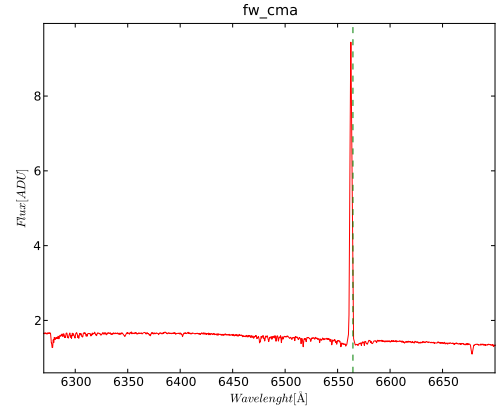


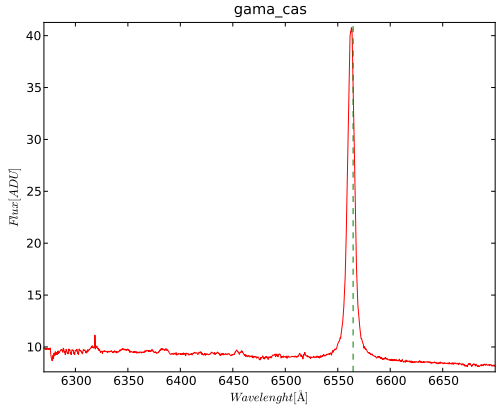


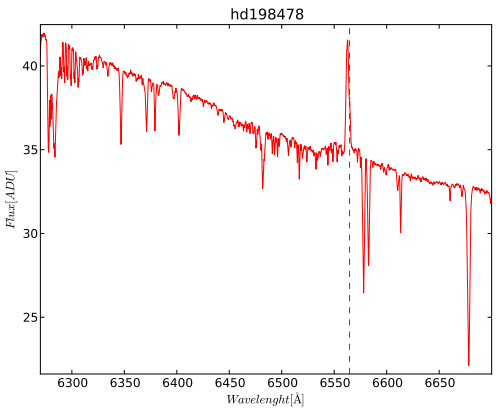


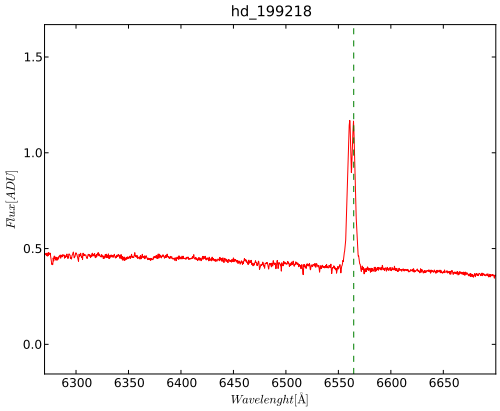


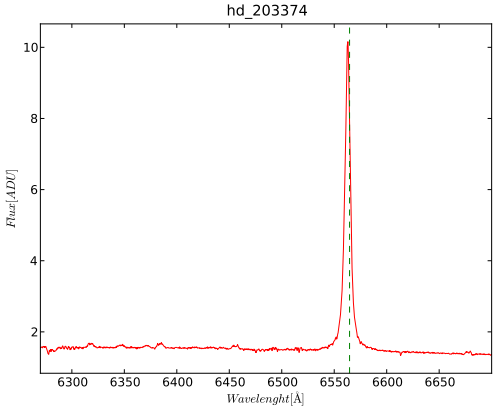


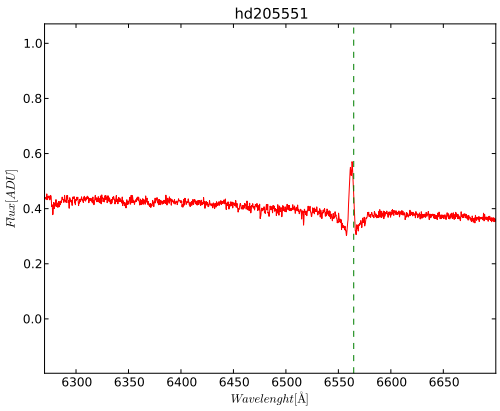


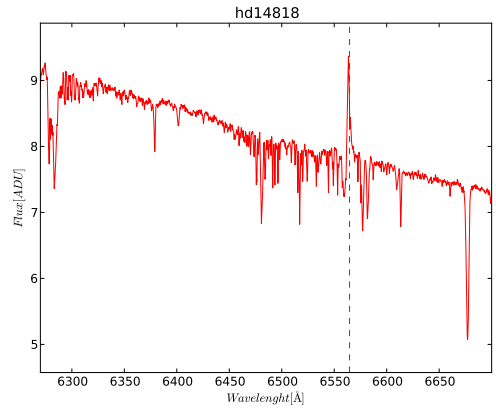


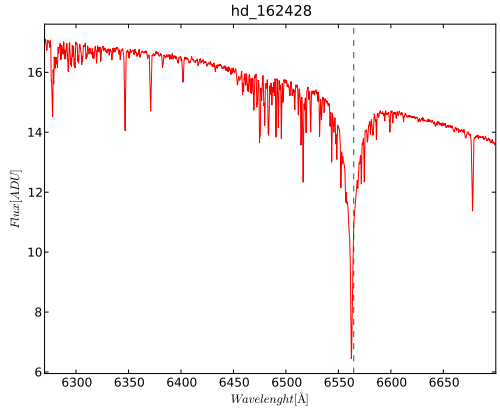

hd198478
Flux[ADU]
Wavelenght[Å]

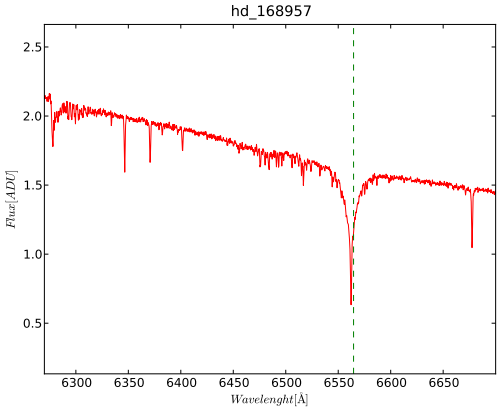

hd_199218
Flux[ADU]
Wavelenght[Å]

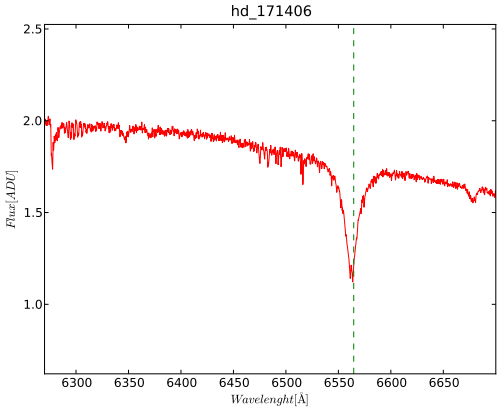

hd_203374
Flux[ADU]
Wavelenght[Å]


hd205551
Flux[ADU]
Wavelenght[Å]

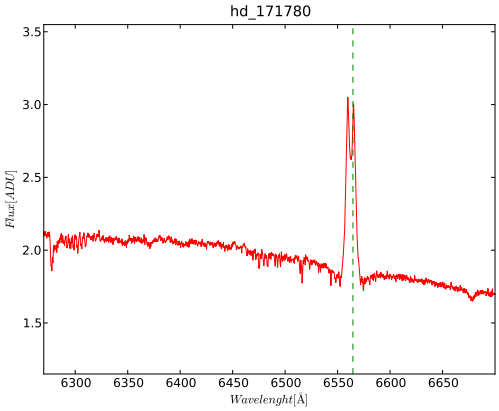

hd14818
Flux[ADU]
Wavelenght[Å]

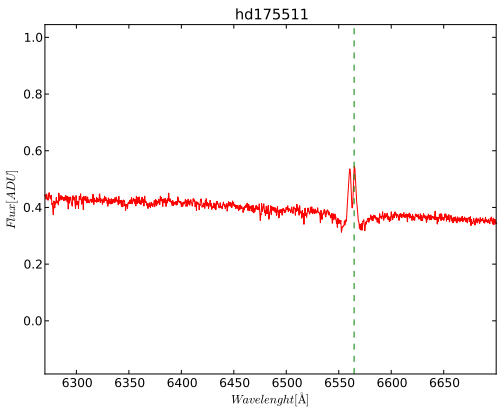

hd_162428
Flux[ADU]
Wavelenght[Å]

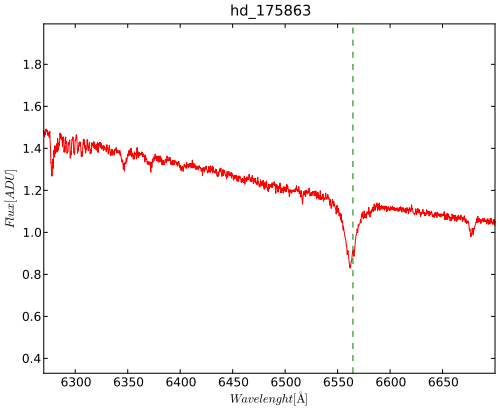

hd_168957
Flux[ADU]
Wavelenght[Å]

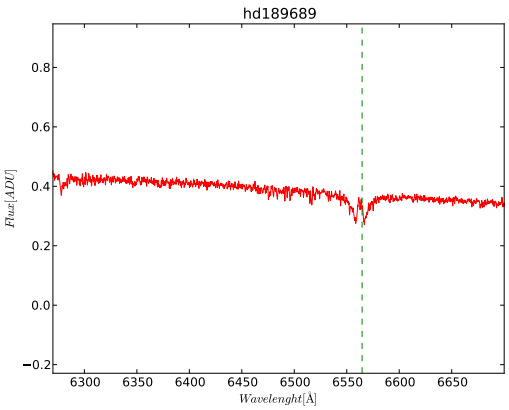

hd_171406
Flux[ADU]
Wavelenght[Å]

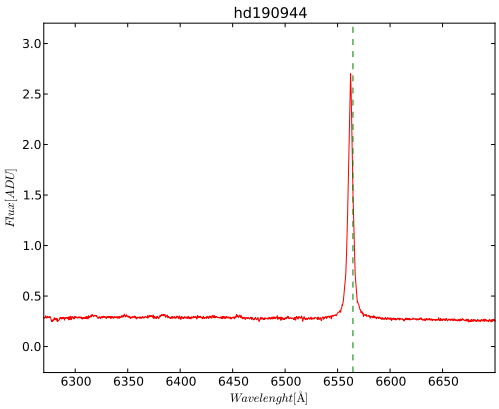
APPENDIX2: BE STARS FROM ONDŘEJOV

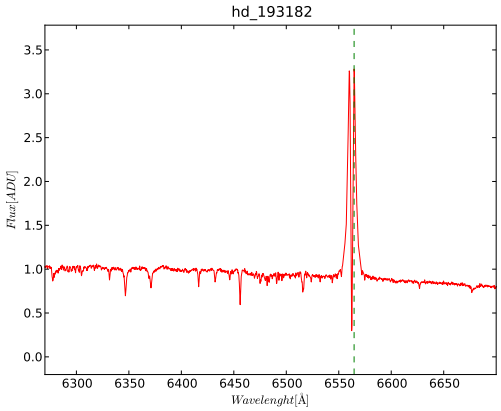


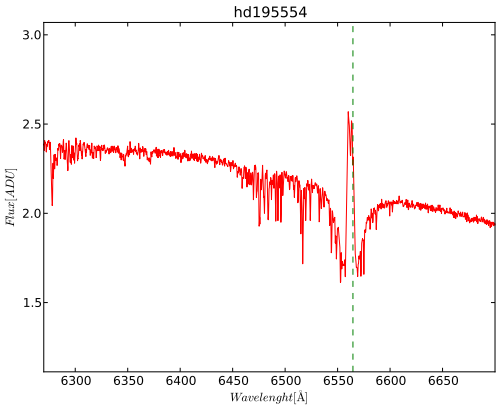


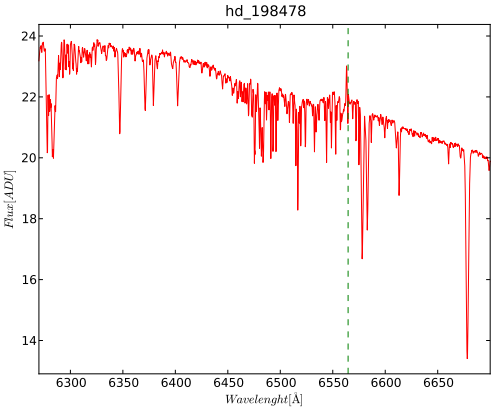












References

Ball, N., Brunner, R., & Gregory, R. 2010, International Journal of Modern Physics D, 19, 1049 15

Ball, N. & Schade, D. 2010, The Long Range Plan for Canadian Astronomy: 2010-2020 vii, 1

Becla, J., Hanushevsky, A., Nikolaev, S., et al. 2006, Arxiv preprint cs/0604112 3

Benson, K., Plante, R., Auden, E., et al. 2009, IVOA Working Draft 5

Berka, P. 2003, Dobývání znalostí z databází (Academia) 16

Berners-Lee, T. & Cailliau, R. 1990, European Particle Physics Laboratory (CERN) 4

Bukvić, S., Spasojević, D., & Žigman, V. 2008, Astronomy and Astrophysics, 477, 967 43

Cohen, M., Wheaton, W., & Megeath, S. 2003, The Astronomical Journal, 126, 1090 24

Free Software Foundation. 2007, GNU General Public License iii

Fukugita, M., Ichikawa, T., Gunn, J., et al. 1996, The Astronomical Journal, 111, 1748 17

Hall, M., Frank, E., Holmes, G., et al. 2009, ACM SIGKDD Explorations Newsletter, 11, 10 iii

Hanisch, R. & Quinn, P. 2010, http://www.ivoa.net/pub/info/TheIVOA.pdf, 24 3, 4

Hirata, R. & Kogure, T. 1984, Bulletin of the Astronomical Society of India, 12, 109 vii, 22

Kotsiantis, S., Zaharakis, I., & Pintelas, P. 2007, Emerging artificial intelligence applications in computer engineering, 160, 3 16

Launer, R. 1979, Robustness in statistics: proceedings of a workshop (Academic Pr) 32

Perryman, M., Lindegren, L., Kovalevsky, J., et al. 1997, Astronomy and Astrophysics, 323, L49 24

Porter, J. & Rivinius, T. 2003, Publications of the Astronomical Society of the Pacific, 115, 1153 1, 22

Schlesinger, B. 1997, A Users Guide for the Flexible Image Transport System 11

Skrutskie, M., Cutri, R., Stiening, R., et al. 2006, The Astronomical Journal, 131, 1163 24

Slettebak, A. 1988, Publications of the Astronomical Society of the Pacific, 100, 770 vii, 22

Thibodeau, K. 1995, Preserving Scientic Data on our Physical Universe 11

Van Kerkwijk, M., Waters, L., & Marlborough, J. 1995, Astronomy and Astrophysics, 300, 259 24

Wenger, M., Ochsenbein, F., Egret, D., et al. 2000, Arxiv preprint astro-ph/0002110 8

Witten, I. & Frank, E. 2005, Data Mining: Practical machine learning tools and techniques (Morgan Kaufmann Pub) 16