

SMART RESUME
BI TOOL TO SHORTLIST CVs FOR A JOB VACANCY

Senarath S.P - IT15099778
Madhushika K.A - IT15097798
Yureshani H.B.D - IT15040404
Y.I Kodithuwakku - IT14115776

Bachelor of Science Special (Honors) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute Of Information Technology
Sri Lanka

September 2018

SMART RESUME
BI TOOL TO SHORTLIST CVs FOR A JOB VACANCY

Senarath S.P - IT15099778
Madhushika K.A - IT15097798
Yureshani H.B.D - IT15040404
Y.I Kodithuwakku - IT14115776

Final Report

The dissertation was submitted in partial fulfillment of the requirements for the B.Sc.
Special (Honors) Degree in Information Technology.

Department of Software Engineering

Sri Lanka Institute Of Information Technology
Sri Lanka

September 2018

DECLARATION

“I declare that this is my own work and this dissertation1 does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:

The supervisor/s should certify the dissertation with the following declaration.

The above candidate has carried out research for the B.Sc Dissertation under my supervision.

Signature of the supervisor:

Date

ABSTRACT

Today in Sri Lanka, most industries follow up one traditional process in hiring new employees. The normal process includes, advertising the vacancy, calling Curriculum Vitaes (CV), short listing them by referring the CVs and interviewing the short listed candidates. Having the right set of CVs is vital since a CV is the representation of the qualifications of an applicant. Also, when it comes to an emergency project, the employer should be able to hire the best employee set within a minimum time period. In this case help of a third party CV storage which already have a collection of related CVs, and have the ability of generating the list of most qualified applicants among them, would be helpful. The submitted CVs should be read properly and check several attributes such as skills, experiences and some personal information in order to select the best. It is much time consuming for a human to read and draw a mind image about the applicant. There is a shortage of tools that support in selecting the best qualified set of employees to an employer. Smart Resume is a business intelligence tool for IT sector, which analyse and classify operational data with classification algorithms to present complex and competitive information to decision makers, in order to dynamically fulfil the business needs. It is built to satisfy the task of generating the list of most suitable candidates. In this paper, we present a combination of desktop and web application that facilitates the task of automating the selection of the most suitable and qualified candidates depending on the attributes given by the user like Age, Gender, Work Experience, Soft skills and Education Qualifications. Depending on the relationship of the attributes (Internal and External) **Smart Resume** will dynamically visualize the most optimal or feasible candidate list.

ACKNOWLEDGEMENT

The work described in this research paper was carried out as our 4th year research project for the subject Comprehensive Design Analysis Project. The completed final project is the result of combining all the hard work of the group members and the encouragement, support and guidance given by many others. Therefore, it is our duty to express our gratitude to all who gave us the support to complete this major task.

We are deeply indebted to our supervisor Mr. Lakmal Rupasinghe, Lecturer of Sri Lanka Institute of Information Technology whose suggestions, constant encouragement and support in the development of this research, particularly for the many stimulating and instructive discussions. We are also extremely grateful to Mr. Jayantha Amararachchi, Senior Lecturer/ Head-SLIIT Centre for Research who gave and confirmed the permission to carry out this research and for all the encouragement and guidance given.

We also wish to thank all our colleagues and friends for all their help, support, interest and valuable advice. Finally, we would like to thank all others whose names are not listed particularly but have given their support in many ways and encouraged us to make this a success.

Table of Contents

DECLARATION	i
ABSTRACT	ii
LIST OF TABLES.....	vi
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS.....	viii
1. INTRODUCTION	1
1.1. Problem to be addressed	1
1.2. Background Context	2
1.3. Research Gap	3
1.4. Research Objective	4
1.4.1 Specific Objectives	4
1.4.2 General Objective	4
1.5. Research Questions.....	4
2. METHODOLOGY	5
2.1. Addressing the Literature.....	5
2.1.1 Web Page Downloading and Classification.....	5
2.1.2 Text recognition from image using ANN and Genetic Algorithm	7
2.1.3 Data migration from a product to a data warehouse using ETL	8
2.1.4 A New Tool for ETL Process	9
2.1.5 The research and application of Business Intelligence System in Retail Industry.....	9
2.1.6 Text recognition from using Artificial Neural Network and Generic Algorithm	9
2.1.7 A Review On Evaluation Metrics For Data Classification Evaluations	10
2.1.8 Performance Evaluation of Predictive Classifiers For Knowledge Discovery From Engineering Materials Data Sets.....	10
2.2. Methodology	11
2.2.1 Download the CVs in automatically	11
2.2.2 read the downloaded CV's.....	12
2.2.3 Classify the data in to relevant columns.	13
2.2.4 Save the classified data in CSV format.....	15
2.2.5 ETL Methodology.....	16
2.2.5.1 ETL Requirement Gathering.....	17

2.2.5.2 Implementation	18
2.2.6 Predictive Model Building Methodology	23
2.2.6.1 Support Vector Machine (SVM).....	25
2.2.7 Model Evaluation & Dashboard Simulation.....	26
2.2.7.1 Model Evaluation.....	26
2.2.7.2 Dashboard Simulation.....	28
2.3. Tools and Techniques	29
2.4. Research Findings.....	30
2.5. Testing	30
3. Results & Discussion	42
3.1. Evidence.....	42
3.2. Results.....	42
3.3. Discussion.....	44
4. Conclusion	45
5. References.....	46
6. Glossary	47
7. Appendices.....	47

LIST OF TABLES

Table 1 List of Abbreviations	viii
Table 2 Comparison with existing systems	4
Table 3 Candidate Attribute List	20
Table 4 Test Case 1	32
Table 5 Test case 2	33
Table 6 Test Case 3	33
Table 7 Test Case 4	34
Table 8 Test Case 5	34
Table 9 Test Case 6	35
Table 10 Test Case 7	35
Table 11 Test Case 8	36
Table 12 Test Case 9	37
Table 13 Test Case 10	37
Table 14 Test Case 11	38
Table 15 Test case 12	39
Table 16 Test case 13	39
Table 17 Test case 14	40
Table 18 Test case 15	41
Table 19 Test case 16	41

LIST OF FIGURES

<i>Figure 1- Neural network proposed for digits</i>	8
<i>Figure 2- Download CVs in automatically</i>	12
<i>Figure 3- Node modules</i>	12
<i>Figure 4- Import pdfminer</i>	13
<i>Figure 5- convert pdf</i>	13
<i>Figure 6- Identify Names</i>	14
<i>Figure 7- Pattern identify</i>	15
<i>Figure 8- Save data in csv file</i>	16
<i>Figure 9 Extraction Interface</i>	20
<i>Figure 10 Extraction Interface</i>	20
<i>Figure 11 Attribute Mapper Interface</i>	21
<i>Figure 12 Transformation Interface</i>	23
<i>Figure 13 SVM Model</i>	26
<i>Figure 14 Confusion Matrix</i>	27
<i>Figure 15 Confusion Matrix for each Algorithm</i>	28
<i>Figure 16 Listing all selected candidates</i>	43
<i>Figure 17 ROC Curve for the predictive Model</i>	44

LIST OF ABBREVIATIONS

Acronyms	Definition
BOT	A bot (short for "robot") is an automated program that runs over the Internet
HTTPS	Hypertext Transfer Protocol Secure
SIFT	Scale-invariant feature transform
CV	Curriculum vitae
CSV	CSV file format
JS	JavaScript
NLP	Natural language processing
AI	Artificial Intelligence
HD	High Definition
GUI	Graphical User Interface
SVM	Support Vector Machine
PDF	Portable Document Format

Table 1 List of Abbreviations

1. INTRODUCTION

1.1. Problem to be addressed

The world is a chain of businesses. As the businesses get bigger day by day, the complexity and the competition are highly increasing. New trends are being incorporated with business ecosystems. So day by day small to large all companies have to update themselves in terms of resources, manpower and infrastructures in order to maintain a competent and business system.

When company hiring new employees, company will have to spend much time, effort and cost on finding suitable candidates among thousands of CV's in educated and qualified ones. Nowadays, company recruiting process have to spend a huge cost and time on selecting the perfect ones for the vacant position.

1. Advertise the vacancy
2. Call Curriculum Vitaes of the interested candidates.
3. Short listing the applied candidate list by referring their Curriculum Vitaes.
4. Interview the shortlisted candidates and recruit the most suitable ones for the position.

But practically, it takes a lot of time and effort for a human to judge an employee's skill and talent just by reading their Curriculum Vitaes. Normally, a CV should contain 2 to 3 pages and all the relevant qualifications should be listed there. Because, according to normal policy, the time dedicated to reading one CV is 6 to 7 seconds. The reader should be able to grab the relevant information within that time period.

But, practically, there may be well qualified, talented candidates, who have a large skill set and a CV extended from 7 to 8 pages since it has to hold each and every qualification they achieved. Sometimes, the required qualifications for the specific position they applied, would be included in the last pages of the CV. In this kind of scenario, the reader would miss the important skills or points because they cannot waste much time on one CV. It is much time consuming for a human reader to read one CV end to end. And also, the most qualified candidates would not be called to the

interview just because their CV is too long or not well formatted. It is a huge disadvantage not only to the candidate but also to the company, since the company may lose the best employee to their vacancy.

On the other hand, there may be hundreds of applicants for a vacancy of a large IT industry. In such scenario, it is very hard and time consuming to download each and every CV and read them one by one in order to shortlist in human hands.

1.2. Background Context

One of the best way to download the CV's to use automated web site. There are technology cold Noad.js. Node.js is a platform built on Chrome's JS. It runtime building easily and fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. So in that case can create web-site to download the candidates CV in automatically.

There is a python library call pdfminer. Pdfminer convert the pdf file in to text. Natural language processing (NLP) is a branch of Artificial Intelligence (AI). It ability of a computer program to understand, interpret and manipulate human language. NLP has an open-source library SpaCy.

Pdfminer use to convert pdf file to text and split the word. Then use the pattern and NLP to identify the words. SpaCy library support for 31+ languages. So SpaCy library use to identify the candidates' name and regular expression use to identify the pattern of age, address, e-mail etc...

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. This function creates a Pandas DF with one row for every input resume, and columns including the resumes file path and raw text

1.3. Research Gap

Even though there are existing proposed products in the market area, they do not address most of the problems that the proposed system is going to address.

Most of the available tools hasn't a way to download CV's in automatically and read the downloaded CV's. There are no way to identify the details and write into a CSV format in a proper way.

The following table shows a comparison of features between the existing products or applications and the proposed solution “**Smart Resume**”.

Features	Oracle BI	Birst	Jobscan	Smart Recruit	Smart Resume
BOT(Automated)					✓
ETL Tool					✓
Optimal Solution			✓		✓
Feasible Solution					✓
All the user does not have to interact with the system	✓	✓	✓	✓	✓
Visualization of data in an abstract way		✓	✓	✓	✓

Predictive	✓	✓	✓		✓
Analytics					

Table 2 Comparison with existing systems

1.4. Research Objective

1.4.1 Specific Objectives

- Download the CVs in the automatically
- To read the downloaded CVs
- Classify the data in to relevant columns
- Save the classified data in CSV format

1.4.2 General Objective

The outcome of this research project will produce a hybrid application to download the candidate's CV's, read the CV's and convert it in to csv file with in proper format.

1.5. Research Questions

Below research problems are being addressed by this system.

- Increasing the efficiency and fast in the web-site
- Increasing the correctness of identify the candidates name.
- Correctness of the identifying the pattern.
- Increasing the efficiency of reading and writing the CV's

2. METHODOLOGY

2.1. Addressing the Literature

2.1.1 Web Page Downloading and Classification

According to the research paper, the system downloads the Web pages using basically using Microsoft's Windows Internet API Tool (Winlnet). Also to go through the links, PDFs, to identify texts, generate the successors of the downloading Web pages, Breadth-First search algorithm, and the Constraint Satisfaction method is used. There are two processes happening in the system; Downloading and Classification. Winlnet is there to connect to web servers when downloading data in different formats; HTML, images, and PDF. Several tasks are done by Winlnet such as requests to the web server for downloading the pages, determining a transfer mode (ASCII or binary) based on the relevant Web page's header. For controlling data flow and to track the downloading progress Breadth-First search algorithm is used. Furthermore to make sure that the downloading moves smoothly without any error, the same Web page is not downloaded twice, and to revisit the unsuccessfully downloaded Web pages again. Breadth-First search algorithm is used because the links among Web pages are similar to a tree structure.

The Breadth-First search is implemented by using two lists. There are open list and closed list. It's to keep track of the progress through the state space. Open list is maintained as a queue. Queue means first-in-first-out (FIFO). It contains all states that have been generated but whose children haven't been examined. The order in which states are removed from open list and recorded in closed list determines the order of the search. When the search is finished, the closed list contains the path of states that have been examined through the search process.

Each list consists of a series of nodes, which contain the uniform resource locator (UIU) addresses of the Web pages. The lists are defined as follows:

- Open list: stores the addresses of pages waiting to be downloaded.
- Closed list: stores the addresses of pages that are successfully downloaded.
- Revisited list: stores the addresses of pages that failed during the downloading process, and are to be revisited later.

```
//Start algorithm
begin
Open := [So 3; // set flag(1) to So
Closed := [];
Revisited := [];
while Open != []
begin
pick the head node in Open list, call it X;
if download X succeeds
generate children of X with flag( I);
add those children to the tail of Open;
remove X from Open;
put X on Closed;
else //download X fails
remove X from Open;
if node has flag( 1)
```



```

        add X to the head of Revisited;
    else //node has flag(0)
        add X to the tail of Revisited;
    end if
end if
if Open = [] &&Revisited != [] .
    if the head node of Revisited has flag( I )
        remove the node from Revisited;
        set flag(0) to the node;
        put it on Open;
    end if
end if
end while
end
//End of algorithm

```

Under the classification, contents of the hyperlinks will be categorized into texts, abstract PDF files and etc. For this purpose Constraint Satisfaction method is used. As the Classification is happening when downloading, only the necessary pages will be downloaded and placed in the relevant directories according to their formats. [1]

2.1.2 Text recognition from image using ANN and Genetic Algorithm

According the paper an artificial neural network and genetic algorithm is used to solve effective text recognition problem. Artificial neural networks (ANNs) are a family of statistical learning models inspired by biological neural networks (the central nervous systems of animals, in particular the brain). ANNs are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which send messages to each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning.

In order to do that a hetero-associative neural network is used to train the system for deciphering digits from pdf or jpeg images which are not readable.

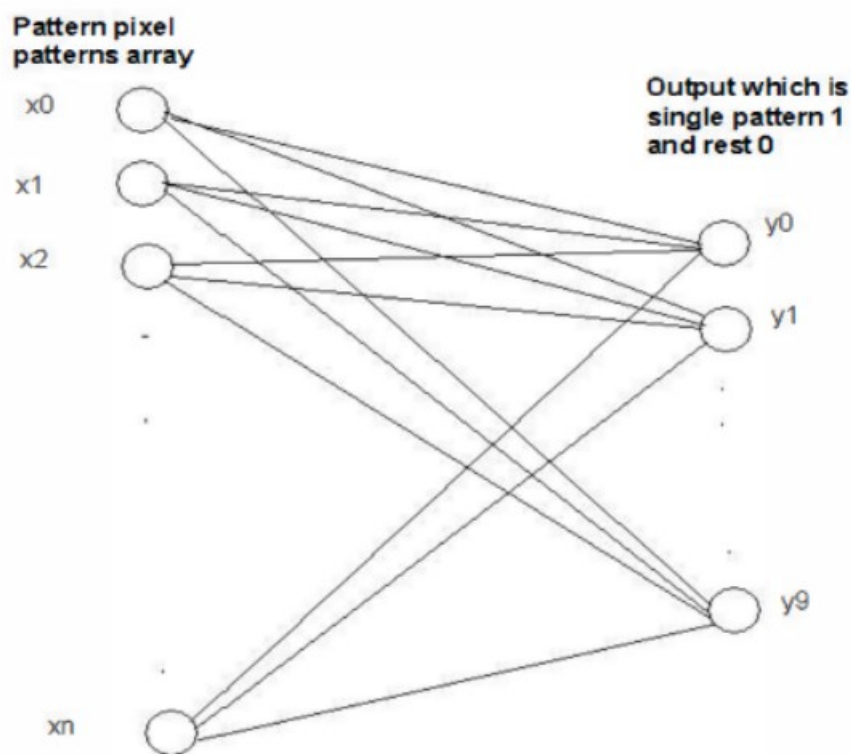


Figure 1- Neural network proposed for digits

For the purpose of analyzing texts from handwritten or text file a crossover based genetic algorithm used. The algorithm solves the problem of deciphering digits and characters from image. It's done by parsing image and converting it to a pixel array. The algorithm selects digits and characters and performs crossover with trained patterns with variable heights. [2]

2.1.3 Data migration from a product to a data warehouse using ETL

The Objective of this research is to migrate Historical Data of Risk Management Product to a Data Warehouse. [3] And Get Previous Day Reports for Further Analysis and Creating Reports. And They Use Large No of Data Such as Risk Management data, Ledger Data and Financial Application Data.

In Methodology, they considered about migration Data from ETL tool instead of PL/SQL based transformation, Manually Transformation when Mapping Data. In Mapping Data- First, they identify the Source Tables then what are the Entities That They want to Map into the Oracle Database. They Use Agile Methodology For their Approach in Research.

2.1.4 A New Tool for ETL Process

The Objective of this research is to create a Model to active ETL tool. Use SQL Queries to Approach the Process for Mapping Data. [4]

SQL Queries use for all construction Process like data extraction, cleaning Procedures, direct Storage and for front-end information delivery. Try to reduce the complexity of the process by applying SQL queries.

2.1.5 The research and application of Business Intelligence System in Retail Industry

The objective of this research paper addressing the important of a BI tool in retail industry [5] including ETL tool. Because of business decision makings processes complicated traditional database system is not support for analytical and data intelligence processing. so it come up with a solution which includes ETL, data warehouse and data mining in order to implement BI tool which support in retail industry[5].

2.1.6 Text recognition from using Artificial Neural Network and Generic Algorithm

According the paper an artificial neural network and genetic algorithm is used to solve effective text recognition problem. In order to do that a hetero-associative neural network is used to train the system for deciphering digits from pdf or jpeg images which are not readable. For the purpose of analyzing texts from handwritten or text file a crossover based genetic algorithm used. The algorithm solves the problem of deciphering digits and characters from image. It's done by parsing image and converting it to a pixel array. The algorithm selects digits and characters and performs crossover with trained patterns with variable heights. [10]

2.1.7 A Review On Evaluation Metrics For Data Classification Evaluations

This Research paper presents that selecting the appropriate metric to determine the optimal solution for obtaining an optimized classifier is a decisive step. The correct choice of the metric ensures that generative type classification training classifier is optimal. In this article, it is expected that the reviews of some metrics to recognize the optimum solution will sensitize data mining Researchers on this topic and encourage them to think carefully before choosing and applying metrically suitable to optimize the classification of the training.[11] In addition, this article also suggests several important aspects in constructing a better metric for recognizing the optimal solution for the generative type of classification algorithms. [11]

2.1.8 Performance Evaluation of Predictive Classifiers For Knowledge Discovery From Engineering Materials Data Sets

In this paper, to classify the engineering materials into different classes for the selection of materials that suit the input design specifications, naïve Bayesian and C4.5 decision tree classifiers(DTC) are applied. Here, classification are analyzed individually and their performance evaluation is analyzed with the predictive parameters of confusion matrix and standard measurements, and the result of the classification is analyzed on different class categories. [12]

2.2. Methodology

This section indicates detailed descriptions about the techniques and mechanisms used to make this project a success. And also how our project is carried out, what are the materials and data needed, and how they will be collected. Apart from that, the research areas that we have identified are explained logically.

Main goal of this research part is to develop an intelligent boot. A bot (short for "robot") is an automated program that runs over the Internet will be implemented to do the following tasks.

- Download the CVs automatically
- To read the downloaded CV's.
- Classify the data in to relevant columns.
- Save the classified data in CSV format.

2.2.1 Download the CVs in automatically

The candidates will upload the CV in company website, it will automatically download and save the company local space.

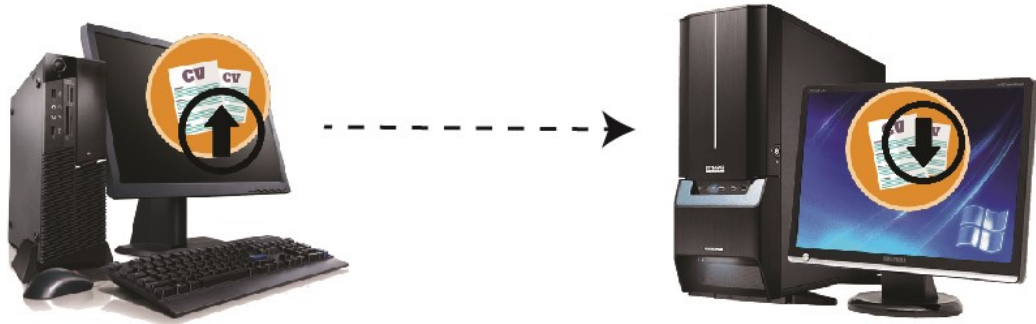


Figure 2- *Download CVs in automatically*

Node.js is used to implement the web-site. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

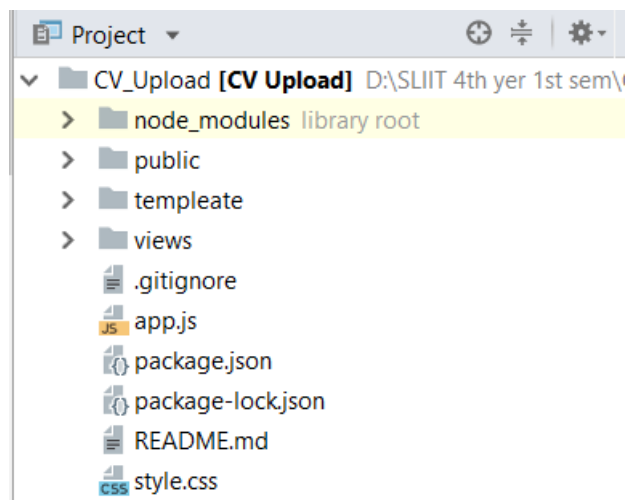


Figure 3- Node modules

2.2.2 read the downloaded CV's

Using python library pdfminer convert the pdf file in to text.

```
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter
from pdfminer.pdfpage import PDFPage
from cStringIO import StringIO
```

Figure 4- Import pdfminer

```
def convert_pdf_to_txt(input_pdf_path):
    try:
        logging.debug('Converting pdf to txt: ' + str(input_pdf_path))
        # Setup pdf reader
        rsrcmgr = PDFResourceManager()
        retstr = StringIO()
        codec = 'utf-8'
        laparams = LAParams()
        device = TextConverter(rsrcmgr, retstr, codec=codec, laparams=laparams)
        interpreter = PDFPageInterpreter(rsrcmgr, device)
        password = ""
        maxpages = 0
        caching = True
        pagenos = set()

        # Iterate through pages
        path_open = file(input_pdf_path, 'rb')
        for page in PDFPage.get_pages(path_open, pagenos, maxpages=maxpages, password=password,
                                      caching=caching, check_extractable=True):
            interpreter.process_page(page)
        path_open.close()
        device.close()

        # Get full string from PDF
        full_string = retstr.getvalue()
        retstr.close()

        # Normalize a bit, removing line breaks
        full_string = full_string.replace("\r", "\n")
        full_string = full_string.replace("\n", " ")

        # Remove awkward LaTeX bullet characters
        full_string = re.sub(r"\(cid:\d{0,2}\)", " ", full_string)

        return full_string.encode('ascii', errors='ignore')
```

Figure 5- convert pdf

2.2.3 Classify the data in to relevant columns.

Identify the candidates name using SpaCy NLP library. SpaCy is the best way to prepare text for deep learning. SpaCy can

- Non-destructive **tokenization**
- **Named entity** recognition
- Support for **31+ languages**
- **13 statistical models** for 8 languages
- Pre-trained **word vectors**
- Easy **deep learning** integration
- Part-of-speech tagging
- Labelled dependency parsing
- Syntax-driven sentence segmentation

```
d@: check_name(string_to_name):
|
| try:
|     string_to_name = unicode(string_to_name)
|
|     nlp = spacy.load('en_core_web_sm')
|
|     doc = nlp(string_to_name)
|
|     doc_entities = doc.ents
|
|     # doc_2 = nlp(my_text)
|     for ent in doc.ents:
|         if ent.label_ == "PERSON":
|             print('{}'.format(ent))
|             a = ent
|             break
|     result = a
|
|     return result
| except Exception, exception_instance:
|     logging.error('Issue parsing name: ' + string_to_name + str(exception_instance))
|     return None
```

Figure 6- Identify Names

Regular expression use to identify the pattern of age, address, e-mail etc...


```

def check_email(string_to_search):
    try:
        regular_expression = re.compile(r"[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}", re.IGNORECASE)
        result = re.search(regular_expression, string_to_search)
        if result:
            result = result.group()
            return result
    except Exception, exception_instance:
        logging.error('Issue parsing email number: ' + string_to_search + str(exception_instance))
        return None

def check_address(string_to_search):
    try:
        regular_expression = re.compile(r"[0-9]+ [a-z0-9\.\# ]+\bst\b", re.IGNORECASE)
        result = re.search(regular_expression, string_to_search)
        if result:
            result = result.group()
            return result
    except Exception, exception_instance:
        logging.error('Issue parsing email number: ' + string_to_search + str(exception_instance))
        return None

```

Figure 7- Pattern identify

2.2.4 Save the classified data in CSV format.

After the identify the data, it's write a XL sheet in a proper format and save the xl file in local space in csv format.

```

def create_resume_df(data_path):
    # Create a list of documents to scan
    logging.info('Searching path: ' + str(data_path))

    # Find all files in the data_path which end in '.pdf'. These will all be treated as resumes
    path_glob = os.path.join(data_path, '*.pdf')

    # Create list of files
    file_list = glob.glob(path_glob)

    logging.info('Iterating through file_list: ' + str(file_list))
    resume_summary_df = pd.DataFrame()

    # Store metadata, raw text, and word count
    resume_summary_df["file_path"] = file_list
    resume_summary_df["raw_text"] = resume_summary_df["file_path"].apply(convert_pdf_to_txt)
    resume_summary_df["num_words"] = resume_summary_df["raw_text"].apply(lambda x: len(x.split()))

    # Scrape contact information
    resume_summary_df["name"] = resume_summary_df["raw_text"].apply(check_name)
    resume_summary_df["gender"] = resume_summary_df["raw_text"].apply(check_gender)
    resume_summary_df["age"] = resume_summary_df["raw_text"].apply(check_age)
    resume_summary_df["languages"] = resume_summary_df["raw_text"].apply(check_languages)
    resume_summary_df["phone_number"] = resume_summary_df["raw_text"].apply(check_phone_number)
    resume_summary_df["area_code"] = resume_summary_df["phone_number"].apply(functiontools.partial(term_match, term=r"\d(3)"))
    resume_summary_df["email"] = resume_summary_df["raw_text"].apply(check_email)
    resume_summary_df["email_domain"] = resume_summary_df["email"].apply(functiontools.partial(term_match, term=r"@[.a-zA-Z]+"))
    resume_summary_df["address"] = resume_summary_df["raw_text"].apply(check_address)
    resume_summary_df["linkedin"] = resume_summary_df["raw_text"].apply(functiontools.partial(term_count, term=r"linkedin"))
    resume_summary_df["github"] = resume_summary_df["raw_text"].apply(functiontools.partial(term_count, term=r"github"))

    # Scrape education information
    resume_summary_df["phd"] = resume_summary_df["raw_text"].apply(functiontools.partial(term_count, term=r"ph.d.?"))

```

Figure 8- Save data in csv file

2.2.5 ETL Methodology

This section contains detailed information about the tools and techniques used to make an ETL (Extract Transform Load). Including the way the software is implemented, how much data and functions it needs and how they have been extracted. And also includes time frames and schedules that are necessary for some purposes. In addition to this, the research areas identified to carry out this component have been explicitly explained in this section.

To carry out this research project, follow the Agile Methodology. Prior to the start of the process, a research was made of the identity of the major problems faced by IT industries that when selecting suitable candidates. After reviewing the field problems, we have made a literature review for BI (Business Intelligence) tools available and what are the drawbacks of them. Then we come up with a most suitable solution for

creating Business Intelligence tool by addressing major business problems in order to select most suitable candidates for a job vacancy.

The feasibility study has been proved that, the system is technically, operatively and financially feasible, because it is built on open source technology and there is no limitations and dependencies. Then we have a focus on the demands of gathering functional and non-functional requirements

During design phase, we developed the high-level architecture design in order to incorporate the gathered functional and non-functional requirements of

- Desktop Application

This includes core components of Smart Resume which are automated ETL tool. These components will be displayed using controls in the main user interface.

- Web Application

Use for Visualization of Data.

2.2.5.1 ETL Requirement Gathering

Requirement gathering was done as a team as it was a common information for the entire system. There are so many software designed to be necessary for the business analysis. Unfortunately, these tools are very expensive because they are mainly focused on high level organization. It's true that medium and small-sized company cannot benefit greatly from these BI tools that exist in the market. Our primary objective is to develop business intelligence tool that can select most suitable candidates for IT industry vacancies even for a small company afford these business intelligence tools.

Smart Resume BI Tool mainly focused on IT industry as its initial step.

The information gathering was difficult, since we had to gather secure data (Candidates related data) from IT Industry giants who supply candidates for companies which need candidates for their vacancies on time. The accuracy of the system depends on the amount of data gathered.

We gathered data and find existing business tools, what are the drawbacks, what are the technologies used, and finally come up with selected attributes list by brainstorming with team members and Supervisor.

When it comes to Predictive models and the results of each vacancy, manipulation of data is more important. The data may contain different formats with different impurity levels. That's why we have to use most efficient methods for extraction and data transformation. We cannot miss any data as it's affected to the prediction model and analyze. So we come up with highly reliable ETL tool for Data warehousing.

2.2.5.2 Implementation

ETL (Extract, Transform and Load) Tool

Data in a business can be very useful if and only if analyzed properly [6] which leads to take strategic business decisions. It is acceptable that the data can be in various formats and locations. Thus **Smart Resume** should be able to extract data automatically in whatever form it is available.

The user can select only required attributes to analyses the data hence **Smart Resume** will only consume the selected attributes and precede for predictive analysis. Furthermore **Smart Resume** will automatically detects odd data and remove it from processing. So that it will result in high- accuracy and best quality output. We only focused on Flat files as Data Source.

Basically ETL consists of 3 parts – Extract, Transform, and Load.

Extract includes 2 sub-parts, “identification of the data sources” and “extraction” of selected data automatically. In most of the organizations, there can be different data sources related to the system, stored in different locations in different formats such as CSV, XML, SQL, and JSON etc. So the system should be able to identify all these data sources and process them. After identification, necessary attributes has to be extracted and stored in a suitable way for local processing.

After that, data needs to be transformed to a more organized format. This is done in two steps, “data cleansing” and “data transformation”. Data cleansing is performed by

detecting and removing and/or correcting a database's dirty data (i.e., data that is incorrect, out-of-date, redundant, incomplete, or formatted incorrectly). Data imputation techniques will be applied in order to cope up with missing data values and remove duplicate data so as to fit it to the models [7]. Then in the transformation step, data may need to be merged, aggregated, enriched, summarized, or filtered depending on the nature of the integration scenario.

Finally the cleansed and transformed data are loaded and stored in data warehouse for further processing.

Implementing ETL Tool

In the implementation stage, as first step, Focused on what are the technologies use in order to make ETL Tool

NetBeans 8.2 IDE was mainly used to develop the GUI of the ETL tool. Python together with JAVA was used to implement the ETL process. The main reason for choosing python is, it is comparatively fast in data mining. Database management of Smart Resume ETL was done using MySQL.

Following are the programming languages and technologies used in implementation stage of ETL tool.

- Java with JDK 1.8 and Swing for interface development of ETL Tool.
- Python 3.5

During the first phase, the attributes which are related to IT industry was collected through existing literature surveys and questionnaires from IT companies. This will give a clear idea about the suitable attributes that system want to focus in order to make decisions. These attributes were very helpful for next phases.

The Data Warehouse generation occurs via 3 major steps in the ETL tool as follows

Extraction

As in the figure 9, the Extract step includes the data extraction from flat files like CSV (Comma Separated Value). The core target of the extraction is to retrieve all the relevant data from the data source with the minimal resources as possible and display the extracted data in table as figure 10.

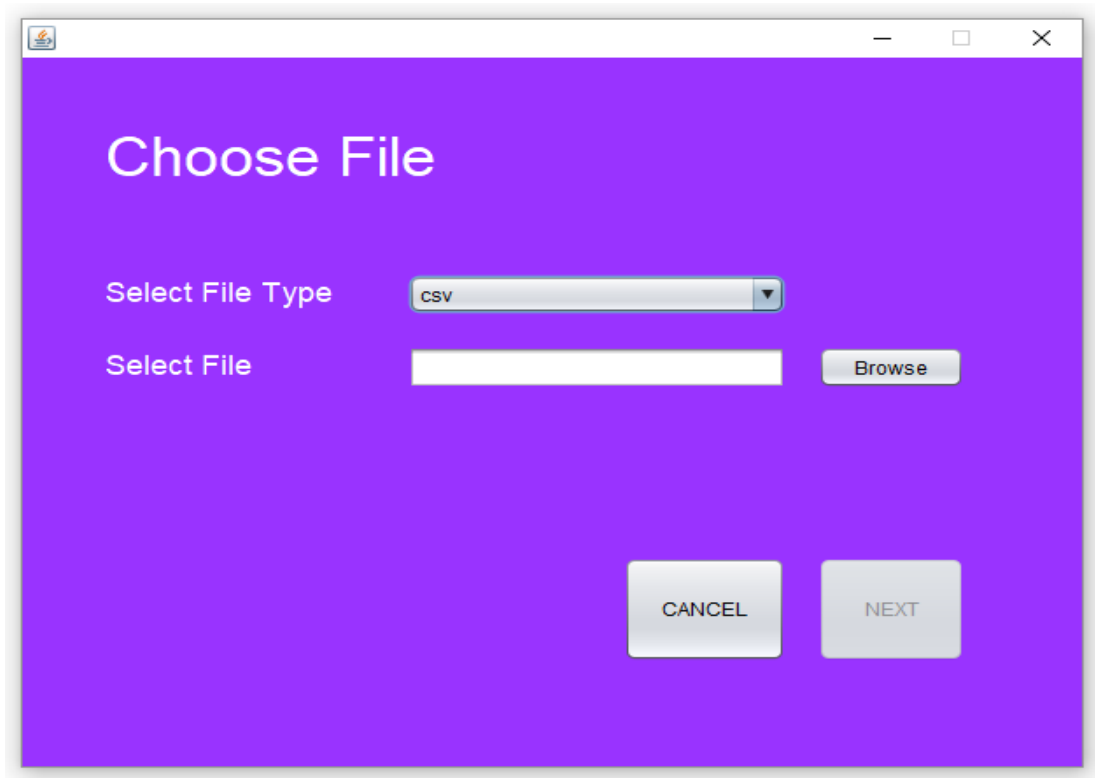


Figure 10 Extraction Interface

Table 3 Candidate Attribute List

Attributes of Train dataset	Attributes of Prediction dataset
<ol style="list-style-type: none"> 1. Age 2. Address 3. Gender 4. Civil Status 5. Email 6. Educational Level 7. Previous Experience 8. Communication Skill 	<ol style="list-style-type: none"> 1. Age 2. Address 3. Gender 4. Civil Status 5. Email 6. Educational Level 7. Previous Experience 8. Communication Skill

9. Languages	9. Languages
--------------	--------------

The extraction has been done in separate thread so it doesn't affect the performance, response time of the source system. The size of the extracted data varies from hundreds



The screenshot shows a window titled "Attribute Mapper" with a purple background. It contains a list of attributes on the left and corresponding dropdown menus on the right. The attributes and their values are as follows:

Attribute	Value
ID	
Fname	fName
Lname	lName
Gender	
Email	
Age	
Phone	
Address	
Experience	
Communication Skills	
Language1	
Language2	
Language3	
Language4	
Ed Qulification1	

Figure 11 Attribute Mapper Interface

of kilobytes up to gigabytes. This depends on the company size.

Next map the attributes which are mentioned above with the column names as in figure 12. Attribute mapping part is 90% automated and attributes which are not mapped correctly needs to be mapped accordingly. Library JAVACSV was used to accomplish the mapping process.

Transformation

Transformation is one of the most vital as it certifies the quality of the data in the data warehouse. Transformation is performed according to the following base rules, such as:

- Remove rows which contain null values for given field.
- Validate numerical fields (remove rows which contain characters other than numbers for given field).
- Remove duplicate values.
- Fill null value columns with average value

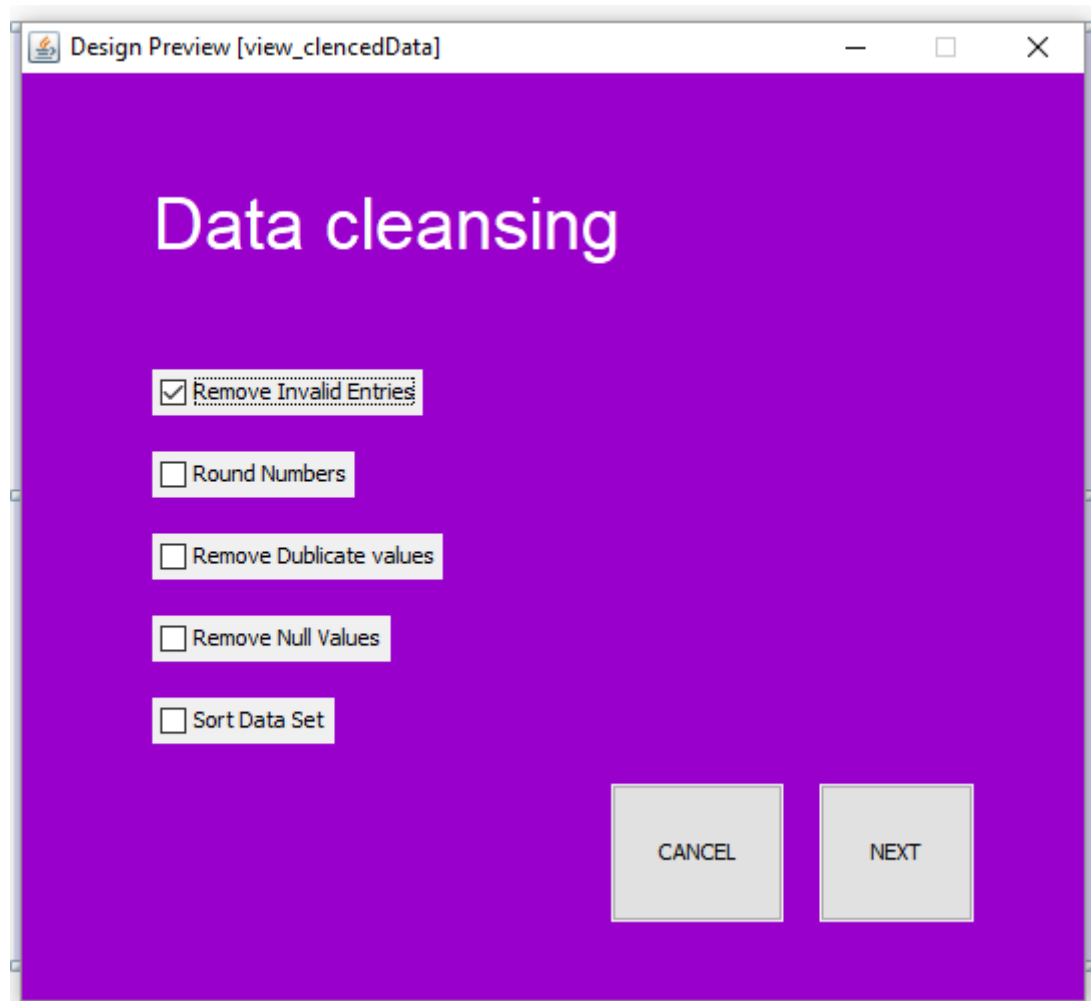


Figure 12 Transformation Interface

Load

Loading phase focuses on creating a data warehouse. Data warehouse has created using MySQL.

2.2.6 Predictive Model Building Methodology

This section includes detailed descriptions of the techniques and mechanism employed to make the predictive model of Smart Resume reality. The descriptions include how

software implementation of the project is carried out, what are the materials and data needed, and how they will be collected. It also includes time frames and schedules that are required in achieving its objectives. In addition to them, the research areas that we have identified in order to carry out this project are explained rationally.

When it comes to devising the best model for predicting the best team for IT industry, there is a need for identifying the most influencing attributes that drive an employee is suitable or not. For this, Smart Resume's prediction model has undergone lots of inspection and interviews with industry experts in the IT sectors which gave the clear and vivid idea on how the model should approach and devise. There can be hundreds of attributes of an employee and a job role that are required by an IT company. Not each and every attribute are responsible for identifying the suitability of an employee.

It has been found that only a few attributes are responsible for identifying the employee and his or her skill set for a job role. The identification of these has been done based on research, expert analysis and few statistical modeling techniques. The predictive model has followed the best, optimal and feasible approach in analyzing the most influencing attributes for statistical modeling.

When it comes to predicting the outcomes of any classification problem, either it can be specified prediction or the default prediction, correction manipulation of data is most important. Data can be in various formats with different impurity level. Hence, we focused our approach on developing best possible and efficient data extraction and transformation techniques. The other key findings we got is when we are cleansing the data, we shouldn't lose any valuable data. It is because data is most important for any company. So BI engineers in Sri Lanka based companies were consulted on how we can come up accurate and highly reliable ETL tool.

After data is loaded to ETL, it cleanses, transforms and process the telco data that are now ready to feed into the models. Models are trained by the imputed data. Once the models are trained, then real data set are fed to the prediction model using ETL tool again and then the result are predicted. Generally, training happens only one time if

we are dealing with the same set of data. If the data is different, the model can be trained anytime with a diversely new set of data.

Predictive model is made with some predictors, and those predictors are variable factors which can make a change in behavior or result. As the first step the data is sampled by using proper sampling mechanism. As examples we can use randomized sampling or probability sampling. Here cluster- probability sampling is used. By using them system will produce Test Set, Training set and Holdout set.

Next suitable classification algorithm is being used to generate the most qualified candidate list. Support vector Machine has been chosen as the main classification algorithm to our approach.

2.2.6.1 Support Vector Machine (SVM)

In SVM algorithm, the data will be mapped into higher dimensional input space and build a hyper plane in this space. The built hyper plane will divide the input data into classes according to their behavior, by provide an optimal separation. In this algorithm, we can decide the correct class of an input data by evaluating the sign of,

$$y(x) = w^T \phi(x) + b \quad (1)$$

If $y(x) > 0$ we assign to class +1 and if $y(x) < 0$, we assign it to class -1. Here $\phi(x)$ is a feature-space transformation, which can map to a space of higher, possibly infinite, dimensions. This is more effective in higher dimensional spaces. Below figure, illustrates the behavior of SVM algorithm.

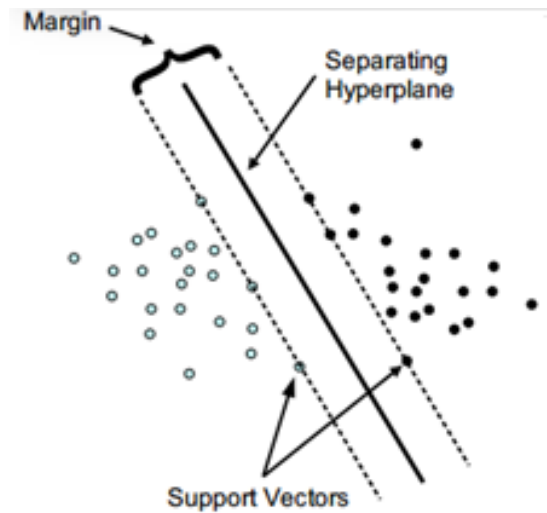


Figure 13 SVM Model

So in building the predictive model smart Resume will follow the following steps;

- Hypothesis Testing
- Data Sampling
- Algorithm building; Classification, Association

This will generate a team for a given job role of a given company depending on whether it is optimal or feasible. This optimality or feasibility will be selected during the prediction of the predictive model. With the help of this predictive model, the business performance can be shown as how it has been in the past, present and what will happen in the future, thus business predictions can be made wisely and easily.

2.2.7 Model Evaluation & Dashboard Simulation

2.2.7.1 Model Evaluation

The evaluation of the predictive model is carried out using three methods.

- ✓ Classification Table (Confusion Metrix)
- ✓ ROC (Receiver Operating Characteristics)
- ✓ Model Accuracy

The classification table shows the number of correct and incorrect forecasts made in comparison with the actual results (objective value) to the data. The ROC curve is a graphical mark that illustrates the operation of a binary classifier system, such as it's the threshold of discrimination is different. A curve is created by tracking the true positive rate against false positive speed in various threshold parameters.

The precision of the model is measured by the proportion of correct predictions to the total number of cases evaluated.

After the Model creation it is very important to test the accuracy of that model. I have used two main model evaluation techniques; Confusion Matrix and Cross Validation to test the overall accuracy.

- **Confusion Matrix**

A classification is generally evaluated by a confusion matrix. Figure.1, illustrates that TN denotes the successfully classified negative examples count. FN denotes the incorrectly classified positive examples count. FP denotes the incorrectly classified negative examples count and TP denotes the successfully classified positive examples count.

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Figure 14 Confusion Matrix

$$\text{Overall Accuracy} = \frac{TP+FN}{(TN+FP+FN+TP)}$$

When is each candidate eligible for a job vacancy, does the classifier accurately predict it? This measurement is called “Recall”. When a classifier predicts a candidate will

suitable, how often does that candidate actually suitable? This measurement is called "precision".

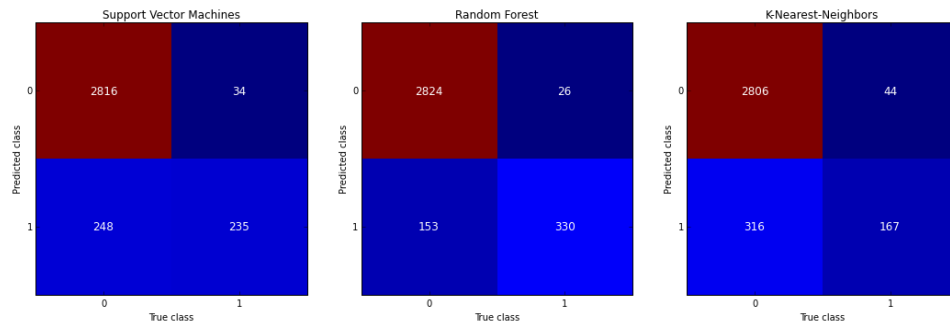


Figure 15 Confusion Matrix for each Algorithm

As mentioned above Support Vector Machine has the highest accuracy, precision, and recall from the algorithms I trained. But this will not always be the same. This will heavily depend on the dataset. Since different job vacancies have different datasets, different patterns, and attributes; we used all three algorithms and will show the accuracies of each in the interface. So that the company can decide on which algorithm to select according to their need or they can allow the system to select the best model automatically by changing the feature given in system settings of the dashboard.

- **Cross Validation**

Cross validation attempts to avoid overfitting (training on and predicting the same data point) while still producing a prediction for each observation dataset. This is trained in a variety of datasets while training a systematic set of models. After training, according to the hypothesis that has hide from each model, many train tests are successively subdivided. When done correctly, each observation has a 'fair' correspondence. Here, I used the Python Scikit Learn for cross validation, it calculates the Model Accuracy is measured by the correct predictions ratio to the total number of cases evaluated.

2.2.7.2 Dashboard Simulation

After analyzing the filtered result data will present it in a format that will make user perfectly understands the difference between raw data and predictable data that would be able to represent the number of candidates got selected for the interview.

These data representations will help the Evaluators/Interviewers take the decisions on selecting most suitable professional candidates according to the company requirement. The dashboard will represent the data in an interactive way using graphs, charts, hierarchies, and tables. The dashboard will provide you Summary of results and key points of analyzed data.

This approach enables the user to identify optimal or feasible CVs according to the specified attributes (requirement) given by the company. So that, the applicants who haven't submitted a good quality CV, also got the opportunity. It is not an only a static panel, where the user presents a set of predefined data each time they are loaded. This system is expected that it will provide a better user interaction through the interactive dashboard.

2.3. Tools and Techniques

Tools

- Notepad++
- MySQL Workbench
- Netbeans 8.2
- PyCharm
- Microsoft Excel
- Inscape
- Photoshop
- Spyder

Technologies:

- Python 3.6
- MySQL
- Java
- Jython Library

2.4. Research Findings

Final Smart Resume system is consists with ETL Component, predictive model and Dashboard features. The results are shown in this section.

For ETL component, Candidate selection attributes were needed. Following are the attributes which were identified mainly from research papers.

1. Age
2. Address
3. Gender
4. Civil Status
5. Email
6. Educational Level
7. Previous Experience
8. Communication Skill
9. Languages

To implement ETL, I have found PETL which is a python library that is helpful in building ETL tool.

Final Smart Resume system has provided a complete tool to classify and short list CV for IT industry. There in Predictive model building I have found a method to select the most suitable candidate list to a specific job role in a company.

In Predictive Model Builder and Visualizer Component for IT industry, in order to build the models, we have used scikit-learn a machine learning library in Python.

2.5. Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can

also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs (errors or other defects).

It involves the execution of a software component or system to evaluate one or more properties of interest.

Unit Testing

Unit testing, also known as a component or module testing, refers to tests that verify the functionality of a specific section of code, usually at the function level. We are planning to carry out testing individual modules at the latter stages of the research project.

Integration Testing

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

System Testing

System testing, or end-to-end testing, tests a completely integrated system to verify that it meets its requirements. We intend to perform a system testing to ensure that we have achieved all the objectives of our research up to the level of performance expected.

In the testing phase of the ETL tool I have followed below steps in order to maintain the product quality.

- Unit Testing – Each interface of the ETL tool was tested by me produced a defects free unit of coding. Following are the test cases I have followed in my Unit Testing phase.

Test case ID	TC1
Test case Description	Validate empty fields of Login Interface
Pre-Condition	Login Interface of the Smart Resume is loaded
Test Procedure	Press Login button without typing anything in Username and Password fields.
Test Input	Username : <Blank> Password : <Blank>
Expected Output	Error should be displayed saying empty fields are detected
Actual Output	Display an error message saying “Empty fields detected. Please fill up all the fields”

Table 4 Test Case 1

Test case ID	TC2
Test case Description	Validate invalid credentials
Pre-Condition	Login Interface of the People Clues is loaded
Test Procedure	Type username and password Press Login button

Test Input	Username : abc Password : 123
Expected Output	Error should be displayed saying invalid credentials are detected.
Actual Output	Display an error message saying “Incorrect Login Credentials”

Table 5 Test case 2

Test case ID	TC3
Test case Description	Check for valid login credentials
Pre-Condition	Login Interface of the Smart Resume is loaded
Test Procedure	Type username and password Press Login button
Test Input	Username : ashi Password : 1234
Expected Output	Data Extraction Interface should be displayed.
Actual Output	People Clues home page is displayed.

Table 6 Test Case 3

Test case ID	TC4
Test case Description	Validate empty field of Extraction Interface

Pre-Condition	Have valid login credentials and Extraction Interface of the Smart Resume is loaded
Test Procedure	Press Extract button without selecting any source files
Test Input	File Type : CSV File Source : <Blank>
Expected Output	Error should be displayed saying empty source file is detected
Actual Output	Display an error message saying “Please select a source file.”

Table 7 Test Case 4

Test case ID	TC5
Test case Description	Validate empty field of Extraction Interface
Pre-Condition	Have valid login credentials and Extraction Interface of the Smart Resume is loaded
Test Procedure	Press Extract button without selecting any source files
Test Input	File Type : CSV File Source : <Blank>
Expected Output	Error should be displayed saying empty source file is detected
Actual Output	Display an error message saying “Please select a source file.”

Table 8 Test Case 5

Test case ID	TC6
Test case Description	Change of source file type in Extraction Interface
Pre-Condition	Have valid login credentials and Extraction Interface of the Smart Resume is loaded
Test Procedure	Select another source file type from the dropdown
Test Input	File Type : CSV
Expected Output	Warning should be displayed saying change of source file
Actual Output	Display a warning message saying “Would you like to remove the selected file?”

Table 9 Test Case 6

Test case ID	TC7
Test case Description	Validate valid file in Extraction Interface
Pre-Condition	Have valid login credentials and Extraction Interface of the Smart Resume is loaded
Test Procedure	Select a valid source file and click on Extract Button
Test Input	File Type : CSV Source File : \etl\datasets\candidate.csv
Expected Output	People Clues Data View page should be displayed.
Actual Output	People Clues Data View page is displayed.

Table 10 Test Case 7

Test case ID	TC8
Test case Description	Validate valid file in Extraction Interface
Pre-Condition	Have valid login credentials and Extraction Interface of the Smart Resume is loaded
Test Procedure	Select a valid source file and click on Extract Button
Test Input	File Type : CSV Source File : \etl\datasets\candidate.csv
Expected Output	Smart Resume Data View page should be displayed.
Actual Output	Smart Resume Data View page is displayed.

Table 11 Test Case 8

Test case ID	TC9
Test case Description	Check Map Attribute button of Data View Interface is working
Pre-Condition	Have valid login credentials and Data View Interface of the Smart Resume is loaded
Test Procedure	Select all the columns needed and click on Map Attributes button Click on Done button
Test Input	Dataset type : Train Dataset
Expected Output	Smart Resume Attribute Mapper page should be displayed.

Actual Output	Smart Resume Attribute Mapper page is displayed.
----------------------	--

Table 12 Test Case 9

Test case ID	TC10
Test case Description	Check Next button of Attribute Mapper Interface is working
Pre-Condition	Have valid login credentials and Attribute Mapper Interface of the Smart Resume is loaded
Test Procedure	Check all the attributes are mapped correctly Click on Next button
Test Input	-
Expected Output	Smart Resume Data Cleansing page should be displayed.
Actual Output	Smart Resume Data Cleansing page is displayed.

Table 13 Test Case 10

Test case ID	TC11
Test case Description	Check Finish button of Data Cleansing Interface is working
Pre-Condition	Have valid login credentials and Add Rule for Attributes Interface of the Smart Resume is loaded
Test Procedure	After selecting data cleansing type click on Finish button

Test Input	Check Cleansing type
Expected Output	Data Insertion Successfully message should be displayed
Actual Output	Data Insertion Successfully message should be displayed

Table 14 Test Case 11

- Integration Testing – In this testing level I integrated each interface of the ETL tool and tested.
- System Testing – In this testing level we have integrated the ETL tool, prediction tool and dashboard and tested as a whole system.

Test Cases for Predictive model building

Test Case ID	TC12
Use case description	Validate valid User Credentials
Pre –Condition	Login interface is loaded
Test Procedure	Type username and password Click login button
Test Input	User name = admin Password = admin123

Expected Output	User should be able to successfully log in to the system
Actual Output	User logs into the system

Table 15 Test case 12

Test Case ID	TC13
Use case description	Validate invalid User Credentials
Pre –Condition	Login interface is loaded
Test Procedure	Type username and password Click login button
Test Input	User name = abc Password = abc1
Expected Output	User should not be able to log in to the system and error message should be displayed
Actual Output	User cannot log in to the system and error message is displaying

Table 16 Test case 13

Test Case ID	TC14
Use case description	Insert job role requirements to the system
Pre –Condition	Requirement gathering page should be displayed
Test Procedure	Fill the form Click Submit button
Test Input	Values for all the fields in the form
Expected Output	The data should be submitted
Actual Output	Data is submitting

Table 17 Test case 14

Test cases for Model Evaluation and Dashboard

Test Case ID	TC15
Use case description	Validate valid User Credentials
Pre –Condition	Login interface is loaded
Test Procedure	Type username and password

	Click login button
Test Input	User name = admin Password = admin123
Expected Output	User should be able to successfully log in to the system
Actual Output	User logs into the system

Table 18 Test case 15

Test Case ID	TC16
Use case description	View Predictive model evaluation results
Pre –Condition	Should be logged into the system
Test Procedure	Type username and password Click login button Click model evaluation results button
Test Input	User name = admin Password = admin123
Expected Output	User should be able to successfully log in to the system
Actual Output	User logs into the system

Table 19 Test case 16

3. Results & Discussion

3.1. Evidence

3.2. Results

Gathering data was the hardest and the most important part of this research because in ETL process I have used attributes which use to predict suitable candidates. These candidates' data were secured in those company and it is not accessible by outsiders. Secure data of candidates in the company which is not given to the outsiders of the company. After spent lot of times and gathering information by reading research papers also finally find the set of most suitable attributes use to implement ETL Tool. Smart Resume has the following ETL tool interfaces.

For the extraction part I have implemented Data extraction interface as Figure 10, in order to select source files such as CSV (comma separated values). Extraction interface is very user friendly and simple. This extraction process is implemented mostly for CSV files to extract Data.

After the source file selection, user can select the necessary columns as needed for data warehouse creation. Then the user navigate to the data mapping interface where the raw data column names are mapped according to column names defined in the data warehouse.

In order to map these columns with the data source which is user has selected interface in Figure 10 can be used.

After data mapping is done data needs to be transformed. In order to have a better cleansing for data I have provided a way to check the rules that need to be cleansed.

After all these steps done extracted, transformed and cleansed data loaded into Data warehouse as the result.

Smart Resume takes data from different sources in different formats like from PDF format and CSV format. Those data can be unorganized and dirty. Hence Smart Resume has to take care of the Data Cleansing and Transformation. After that Smart Resume will make the predictions using the predictive models and the cleansed data saved in the db. Then the results will be saved again to the db. The visualizer will

present those data using dashboard so that the non-technical users can easily understand the results.

The final output of the system is an attractive web-based dashboard which includes detailed graphs, charts, tables and hierarchies. According to the selection made by the Predictive Model, selected candidates will be display in a table with their required personal details. This would be a great help for the company to easily and efficiently select some candidates. That table can sort, search the results. Every candidate is shown a summary of his/her chosen attributes.

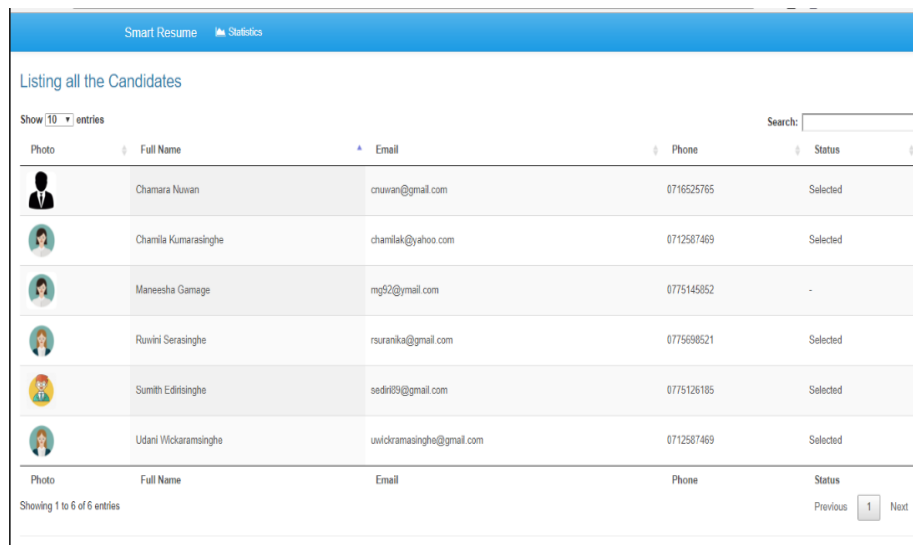








Photo	Full Name	Email	Phone	Status
	Chamara Nuwan	cnuwan@gmail.com	0716525765	Selected
	Chamila Kumarasinghe	chamilak@yahoo.com	0712587469	Selected
	Maneesha Gamage	mg52@gmail.com	0775145852	-
	Ruvini Serasinghe	rsuranika@gmail.com	0775598521	Selected
	Sumith Edirisinghe	sediri89@gmail.com	0775126185	Selected
	Udani Wickramasinghe	uwickramasinghe@gmail.com	0712587469	Selected

Figure 16 Listing all selected candidates

ROC Curve

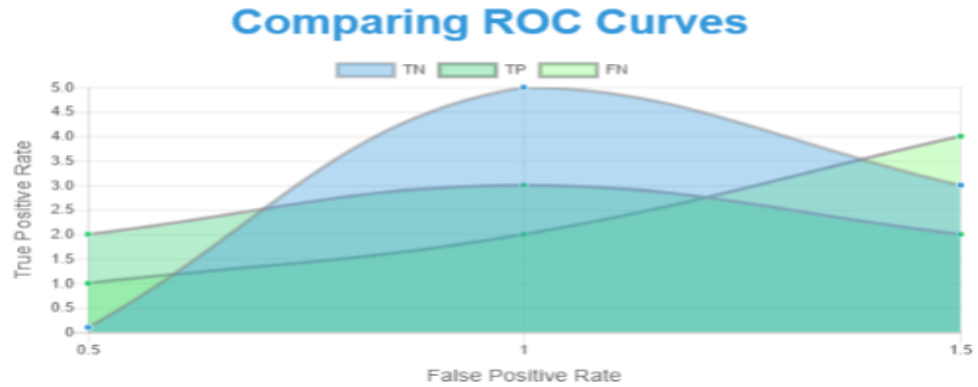


Figure 17 ROC Curve for the predictive Model

3.3. Discussion

With hard working through the year, we have completed the Smart Resume BI tool for selecting CVs for IT Industry. The research was successful because of the help of our supervisor Mr. Lakmal Rupasinghe and my team members who supported to achieve this goal.

Candidates CVs data is the most important data set in this system. To gather these data it was very hardest part of the project as these Candidates data are not given to outsiders as they are very important to the company. Spending lots of time we gathered Data sets and select the attributes which are needed to store in data warehouse for further analysis. These attributes were gathered by reading many research papers, brainstorming and studying related BI tools.

As ETL tool is a core of our system. We need to identify which technologies used for better performance and libraries to create most powerful and useful ETL tool for the system.

Also the predictive model is another key component of the system. Here we have tried our best to give an accurate and understandable result to the non-technical users.

Due to the time constraint, we wouldn't be able to finish some feature which is useful for future researchers and it will be discussed in the next chapter.

4. Conclusion

Today for most companies like IT, receive huge number of CVs for a vacancy as there are a lot of graduates coming out from a university within a year. The quality of the company depends with the capabilities of the recruiters. Therefore getting the best qualified people from incoming applications is very difficult for them. Currently the selection of CVs is done manually. As the huge number of CVs are summarized by manually, qualified CVs can be skipped by CV errors and human errors. That is where Smart Resume comes in to play by reducing all the difficult tasks of selecting optimal set of candidates in place of referring CVs.

Smart Resume contains a Desktop application with ETL (Extract, Transform, Load) Tool to transform data into a meaningful order and Smart Resume has a Dashboard as a Web application in order to predict best candidates for a given job opportunity by analyzing the CV data.

Smart Resume has been developed to choose the suitable Candidate for IT Industry.

As the initial step Smart Resume analyzes and extracts the professional skills, personal skills, personal details and etc. from the CV according to the job vacancy. Finally, Smart Resume displays the results in an attractive proper manner in a dashboard with simple graphs and charts which shows the user friendliness of the system.

In next step our team will be focusing on creating ETL tool which extracts data from any file types like word file, json files, access files and sql files.

5. References

- [1] L. Q. Tran, C. W. Moon, D. X. Le, and G. R. Thoma, "Web Page Downloading and Classification," *Proc. IEEE Symp. Comput. Med. Syst.*, pp. 321–326, 2001.
- [2] M. Agarwal, "Text recognition from image using Artificial Neural Network and Genetic Algorithm," pp. 1610–1617, 2015.
- [3] C. Shrinivasan, "Data migration from a product to a data warehouse using ETL tool," *Proc. Eur. Conf. Softw. Maint. Reengineering, CSMR*, pp. 63–65, 2011.
- [4] Z. Chen and T. Zhao, "A new tool for ETL process," *Proc. 2012 Int. Conf. Image Anal. Signal Process. IASP 2012*, pp. 269–273, 2012.
- [5] T. Gang, C. Kai, and S. Bei, "The research & application of Business Intelligence system in retail industry," *Autom. Logist. 2008. ICAL 2008. IEEE Int. Conf.*, no. September, pp. 87–91, 2008.
- [6] R. Samarasinghe, G. Perera, N. Perera, P. Senaratna, and L. Samarasingha, "People Clues : Business Intelligence Tool for Team Dynamics," pp. 179–184, 2017
- [7] T. Gang, C. Kai, and S. Bei, "The research & application of Business Intelligence system in retail industry," *Autom. Logist. 2008. ICAL 2008. IEEE Int. Conf.*, no. September, pp. 87–91, 2008.
- [8] Agustín-Blas, Luis E., Salcedo-Sanz, Sancho, Ortiz-García, Emilio G., Portilla-Figueras, Antonio, Pérez-Bellido, Ángel M., Jiménez-Fernández, Silvia, "Team formation based on group technology: A hybrid grouping genetic algorithm approach *Computers & Operations Research*, vol. 38, issue. 2, p. 484-495, 2011. [Online serial]. Available: sciencedirect, <http://www.sciencedirect.com/science/article/pii/S0305054810001371> [Accessed March 06, 2016].
- [9] Yang, J., Dai, C. and Ding, Z. (2017). A Scheme of Terminal Mobility Prediction of Ultra Dense Network Based on SVM. 1st ed. [ebook] Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8078755> [Accessed 3 Aug. 2018].
- [10] M. Agarwal, "Text recognition from image using Artificial Neural Network and Genetic Algorithm," pp. 1610–1617, 2015.
- [11] H. M. and S. M.N., "A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION VALUATIONS", *Pdfs.semanticscholar.org*, 2015. [Online].

Available:<https://pdfs.semanticscholar.org/6174/3124c2a4b4e550731ac39508c7d18e520979.pdf>. [Accessed: 01- Apr- 2018].

[12] H. K.S., "Performance Evaluation of Predictive Classifiers For Knowledge Discovery From Engineering Materials Data Sets", Arxiv.org, 2008. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1209/1209.2501.pdf>. [Accessed: 03-Apr- 2018].

6. Glossary

7. Appendices