Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# A NEW OPTIMIZED MACHINE LEARNING APPROACH FOR EXTRACTION AND CLASSIFICATION OF INFORMATIONS IN CURRICULUM VITAE

**SAFIA BAALI[1], HICHAM MOUTACHAOUEK[2] ABDELAZIZ MARZAK[1] , IBRAHIM HAMZANE[1]**

[1] LTIM Laboratory, Ben M'Sik Faculy of science, Hassan II University, Morocco

[2] Strectural Engineering, Inteligent Systems, and Electrical Energy, ENSAM, Morocco

E-mail: [1]safia.baali@gmail.com

**ABSTRACT**

The hiring process has evolved It refers to the process of finding, selecting and recruiting new employees to a company/organisation through CVs. We have presented in this paper a solution in the form of an automated and flexible system to extract vital information from unstructured CVs and transform it into a structured format and then use it in a recommendation system according to the available positions.

**Keywords:** *CV; Machine Learning; Information extraction; NLP; Recommendation system.*

## 1. INTRODUCTION

Online recruiting platforms play an important role in recruiting channels with the rapid growth of the internet. Today, almost all companies and services publish their job vacancies on various online recruitment platforms. There are several thousand job applications uploaded per minute. Online recruiting is extremely helpful in saving time for employers and employees. It allows job seekers to submit their resumes to many employees at the same time, without having to go to the office, and it also saves employees time in organizing a job fair. At the same time, many portals act as a third-party service between job seekers and the human resources of companies, which makes it possible to collect many CVs. For example, LinkedIn.com has collected over 300 million personal resumes uploaded by users. Due to the increasing amount of data, efficient analysis of each CV is a serious problem that has caught the attention of researchers.

In the real world, job seekers typically use a variety of resume text formats and layouts to grab attention. Many CVs are not written according to a standard format or a specific template file. This phenomenon means that the data structure of the CV has great uncertainty. It decreases the success rate of recommending recruits who meet most of the employer's requirements and takes too much time for human resources to perform job matching. In order to improve the efficiency of the job search, it is important and necessary to explore an effective method of matching jobs and candidates. Besides, CV mining is also useful for modeling recruiting platform users.

Depending on their use cases, personal CV data has the following properties. First of all, job seekers write their CVs using different types of composition, but most CVs have general blocks of text, such as personal information, contacts, education, and work experience. Second, personal CVs share the hierarchical contextual structure at the document level, which is shared between different elements in the corresponding text block of each CV. The main reason for this phenomenon is that elements of a block of text sharing the same hierarchical information can make the entire CV more comfortable for readers. First and foremost, a CV can be segmented into multiple blocks of text, and then the facts can be identified based on specific hierarchical background information.

## 2. RELATED WORK

In the literature, many works have tried to propose new techniques for extracting information and classifying CVs. Das et al [1] used the hierarchical approach of extracting information from the CV to extract skills. The idea of this proposed skills extraction work is carried out in two main phases: first, the segmentation phase of the CV into sections classified according to their content and from which the terms representing the skills (basic

skills) are extracted; and the second phase of prediction which consists on the basis of the characteristics extracted previously, in predicting a set of skills that an expert would have deduced, on the other hand in the work [2] the authors proposed a CV analyzer model using an entity extraction process and Big Data tools [3,4]. The work [5] of et al. discusses resume analysis, the recent key part of the text analysis process. It depends on the extraction of the entity. The authors described parsing and extracting features from plain text. This can be a research process for any integrated software system. The candidate can use it. CV parsing is typically the conversion of a free-form CV document into a form of structured information in XML format. On our part we have proposed in [6] a comparative and evaluative study, this work presents a comparative study between the different approaches used for job matching and candidate CV. We have described a new approach to recommend a potential candidate for a specific area of work, our work will be supported by a Morocco-based computer engineering company whose goal is to automate the recruitment process to ensure the candidate's assignment. to the right task and ensure the success of the business, then customer satisfaction. And in our second article [7], we developed a human resources data recommendation system with content-based and collaborative filtering, this recommendation system makes it possible to recommend potential collaborators for a new job offer, using the multi-criteria analysis (AHP) [8] and the matching between the job offer of the new project and employee profiles. We first propose a model of criteria for measuring the skills of the IT team, we validate it by a survey carried out in the IT services company based in Morocco. The data collected is analyzed using the dimensionality reduction (PCA) method [9]. The results indicate that six factors can measure the competence of the collaborator in the team. The document is organized as follows. Sections 3 and 4 describe the CVs, and the entity extraction process, and the problem, sections 5 and 6 concern the proposed model, and the description of our solution, then we conclude in section 7 of this article.

## 3. BACKGROUND

### 3.1 Curriculum vitæ

The curriculum vitae (abbreviated CV) is a document detailing the background and skills acquired by an individual. This is generally the educational and/or professional background that demonstrates the competence of a candidate in a position to be filled. This document constitutes the junction point between the job offer and the demand. The CV can also lend itself to other uses such as introducing yourself to a group, but its role is more in the search for a job.

### 3.2 Hiring history

The hiring process has evolved over time. In the first generation recruiting model, companies advertised their vacancies in newspapers and on television. Candidates sent their CVs by mail and it was sorted manually. Once shortlisted, the recruiting team called the candidates for further rounds of interviews. Needless to say, this procedure took a long time. But industries began to develop and so did the recruitment needs. Companies have therefore started to outsource their recruitment process. Recruitment consulting agencies have sprung up. These agencies required applicants to upload their resumes to their websites in specific formats. The agencies then reviewed the structured data and shortlisted candidates for the company. This process has a major drawback.

### 3.3 Steps of the hiring process

**a) Identify the need for recruitment**

**b) Planning**

**c) Creation of the job description**

When creating a job description, it should include the following: company description, job requirements, duties and responsibilities, and finally, cover letter. In addition to mentioning all the important details in the job description.

**d) Admission meeting between the recruiter and the hiring manager**

The objective of a welcome meeting between the hiring manager and the recruiter is to fully understand all the needs and expectations of the new employee, including the technical aspects of the position. This meeting takes place before the publication of a job posting online and its main purpose is to define all the important details regarding the future position.

**e) Publish and promote job offers**

Posting job offers online has become one of the essential ways to make sure people see and apply for the job. This is why companies often try to spread the word as much as possible by posting their job postings on various job boards, different social media, or certain online portals.

**f) Selection of candidates**

Once applicants have completed their application, it is time to go through their application forms and assess their CVs, cover letters, or any other type of documents that they may have attached

to their application. The pre-selection process allows you to eliminate candidates who clearly do not match the position to be filled and to proceed to the next steps with those who are the most qualified.

**g) Interviews**

When an interview is scheduled, there are several things to consider before conducting it. Interviews require planning and preparation. First, you need to create a list of interview questions. Next, you need to determine what the maintenance process will look like. It is about whether the interview will be conducted live or over the phone, how long it will last and what kind of knowledge will be examined during the interview.

**h) Assessment of candidates' talents**

Candidate skills assessment tests are tests designed to help employers assess the skills of their job applicants and employees.

**i) Checking background and references**

Reference checks allow recruiters not only to verify the validity of information but also to make decisions based on a conversation with someone who has worked with your potential future colleague.

**j) Pre-employment test**

Pre-hire assessments are simple, fast, and fun for recruiters, interviewers, hiring managers, and applicants alike.

**k) Hiring**

It is only when the candidate decides to accept the offer and signs the contract that we can talk about hiring. However, the work is not yet done.

**m) Integration**

More and more companies are starting to understand the importance of good onboarding. Especially since once a company has hired new talent, some people think the job is just getting started. Onboarding candidates involves introducing them to the culture and people of the company, as well as providing them with all the information and training they need to be able to excel in their jobs as quickly as possible.

### 3.4 Benefits of parsing CVs

Save time: By identifying and organizing applications that contain relevant skills and information, and eliminating those that do not, CV parsing helps to hire managers to save hours spent manually reading every CV and cover letter that arrives on their desk.

## 4. PROBLEM & OBJECTIVE OF THE PROJECT

### 4.1 Problem

When a company advertises a vacancy, it's always a pleasure to see tons of potential candidates taking an interest in it and applying. But whether it's a small business or an international company, recruiting and onboarding new talent can be difficult. HR managers and investigators are often misunderstood in what they do to process and select viable resumes for available positions.

Recruiters have been sorting CVs manually for a long time. They read all candidates' CVs and assess them on the basis of skills, knowledge, abilities, and other desired factors. However, it would take a long time for the recruiter to review each CV in detail. In practice, recruiters are therefore forced to do one of the following two things:

- They go through a limited number of CVs, scan them thoroughly and make a selection.
- They go through a limited number of CVs, scan them thoroughly and make a selection.

Either way, organizations lose quality candidates and recruiters waste their time and effort.

According to Small Biz Genius, corporate job postings receive an average of 250 CVs, while 45% of employers say they can't find candidates with the right skills. Facilitating the selection process today is imperative, so this has motivated us to build a more flexible and automated solution.

### 4.2 Objective of the project

Even though CVs are not fully structured, the information is stored in separate sets. Each set contains data: personal information, professional experience, training ... Despite this, CVs are difficult to analyze. Indeed, they vary according to the types of information, their order, the style of writing, etc. In addition, they can be written in different formats. Some of the most common are PDF, DOC, DOCX, TXT, etc. To analyze data from different types of CVs effectively and efficiently, the model should not rely on order or type of data.

The main goal is to build an automated intelligent system capable of extracting all vital information from unstructured CVs and transforming them into a common structured format which can then be classified for a specific position via a recommendation system.

The information analyzed includes:

- The name
- Email-address
- Telephone number

- Social profiles
- Training/diplomas
- Skills

## 5. MACHINE LEARNING

### 5.1 Definition:

Machine Learning was mentioned in 1948 by Alan Turing; his idea was to offer "learning machines capable of building their own codes" [10]. In 1959, Arthur Samuel formulated a first definition of learning approaches as "a field of study in which the computer is given the possibility of learning without having been explicitly programmed". Machine learning approaches can be summarized as techniques that allow a computer system to adapt its analysis and behavior based on its input data. A more formal and recent definition of this field proposed by Tom Michael. Mitchell is that a "computer program learns from experience E, matched with tasks T and a performance measure P, if its performance on tasks T, measured by P, improves with experience E". Thus within the learning currents, two main synthesized subcategories can be distinguished, including:

• supervised learning corresponds to techniques aimed at matching input data with outputs of discrete (we also speak of classification) or continuous (we then speak of regression) values,

• unsupervised learning corresponds to exploratory approaches, in which the computer tries to identify correlations within the data (referred to as clustering),

• semi-supervised learning [11] uses the previous theories mentioned. This domain attempts to extrapolate knowledge of annotated data to knowledge of unannotated data in order to exploit all of the available information.

### 5.2 Supervised learning

Supervised learning consists in determining the relation making it possible to match input data x and output data y and this, in order to produce on new input data x 0, new outputs y 0. approach involves the use of a learning base, defined as a set of input-output pairs denoted {(x1, y1),. . . , (xn, yn)} with n ∈ N. Thus, humans try to give meaning to this data in the form of annotations (or labels) for which the machine must be able to determine the existing relationship. In order to complete this task, the variables specific to each observation are assumed to be sufficiently discriminating to allow the identification of the annotations. This problem is defined by the relation y = f (x), where f is an unknown function corresponding to the observed phenomenon, by an approach function g called the prediction function such that y = g (x), with x = {x1, x2,. . . , xn} a vector of features with sufficient information [12]. This learning is divided into two distinct processes:

• learning or training, which consists of approximating the function f by a function g taking into account labeled data,

• prediction or inference, which consists of predicting on new data from g such that g (x0) = y0 The training phase makes it possible to determine the relationships between data and expectations, on known so-called learning data. The prediction phase reuses the relationships determined during the training phase.

These so-called supervised approaches are grouped together through two major categories:

• classification, that is to say the prediction of discrete values, that is to say the set of relative integers noted Z. The terms of binary classification are also used when the problem formulated has only two classes, and of multi- classification. classes when the situation requires predicting N classes with N> 2.

• regression, that is to say the prediction of continuous values, or the set of real numbers noted R.

The work of this manuscript focuses on classification methods, which result in the categorization of clinical pathological images or lesions according to various levels of dangerousness.

Based on one of the studies [14], a summary of these methods and their respective principle is provided, including:

• logical approaches which correspond to approaches by succession of decisions [14]. In this category, the main representative is that of decision trees [18], also known under the term Classification And Regression Trees, CART (alt .: Tree of Classification and Regression, ACR), the principle of which can be summarized as the construction of a graph, in which each node is associated with a decision defined by the choice of a characteristic and a threshold [20]. The whole complexity of these approaches' rests on the choice of these characteristics, the priority of which depends on the order of importance in the separation of the problem.

• statistical approaches which correspond to probabilistic approaches, which determine the probability of belonging of an element to a class, which can be extended in the form of a network [21]. The Naive Bayes Model, NBM ( Naive Bayesian Model, MBN) [22] is a typical example of a probabilistic approach, in which each characteristic is considered independent. The relationship between

characteristic and annotation is thus defined as the product of the probabilities of each of them belonging to the assumed class, governed according to:

$$\hat{y} = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}}\ p(C_k)\prod_{i=1}^{n} p(x_i \mid C_k)$$

• instance-based approaches which qualify methods consisting of an accumulation of samples in order to enrich a space and not of their interpretation leading to a form of knowledge. The best-known method based on this principle is that of the kNearest Neighbors algorithm, KNN ( Method of k nearest neighbors, KNN) [23], whose major theory is based on the fact that a data belongs to a class. if it is sufficiently close in terms of its characteristics, pre-existing samples of the latter.

It is then necessary to define a distance criterion, the main one used being that of the Euclidean distance.

• Support Vector Machine approaches, SVM ( Support Vector Machine, MVS) correspond to fairly recent approaches whose principle boils down to determining a separation boundary between the various classes, which maximizes a margin sufficient allowing, among other things, better noise tolerance. In order to best determine this boundary, various kernels have been established, the simplest of which is the linear kernel. The presence of a radial basis function kernel or Radial Basis Function is to be underlined, considered as a universal approximator if the data available are sufficient [13]. set approaches represent a category that qualifies methods that employ a set of predictive models in order to build a single, more robust model.

One of these methods, Random Forest, RF ( Forêts Aléatoires, RF) [19] is an extension of the previously mentioned decision tree principle, intended to reduce the risk of over-learning the initial model. This method performs N random selections of observations, for which N decision trees are generated. In order to diversify these trees, a mechanism of random selection of characteristics is also implemented at the level of each decision node. Another of these methods, Gradient Boosting, GB (Increasing the Gradient, AG) is an operating mode generally made up of decision trees. In contrast to RF where each decision tree corresponds to a weak prediction model, GB consists of sequentially decreasing a cost function with each new tree generated.

Faced with these numerous methods, it is difficult to determine an optimal model for a particular situation, especially in the context of highly complex data. Indeed, an empirical evaluation of these various methods is still the best solution in order to determine this optimal model, in return for a significant calculation time. Nonetheless, one of the studies carried out on the performance of supervised models made it possible to determine a certain number of criteria for comparison between these various techniques, generally highlighting the advantages and disadvantages of each of them.

For that. The classification performance which seems undeniable in this field of work and which represents one of the main strengths of SVM models. The learning speed has also been specified for information. Indeed, although restrictive during the experiments of this manuscript, it does not constitute a brake and remains the main drawback of SVMs. Finally, the latter are recognized as robust in the face of situations that may include irrelevant or redundant characteristics. In view of these elements, it seems more than necessary to consider the use of SVMs within this work. Likewise, it is necessary to consider the use of the two mentioned set approach methods, RF and GB, for which no generalist comparative work has been found. Nevertheless, a genomic study seems to argue in favor of SVM and GB. Finally, it is also necessary to consider these classification models within the processes proposed by works close to this theme. These elements are discussed in more detail in their respective part.

### 5.3 Unsupervised learning

Unsupervised or descriptive learning is a second approach to learning in which the computer attempts to independently discover correlations within datasets. These approaches emerge from various issues, such as:

• the reduction of the costs, which are usually human, necessary to obtain annotated data, that is to say data for which the pairs of inputs and outputs are known.

• the discovery of the various relationships that may exist within a cluster of data. Indeed, an annotation corresponds only to a sample of information and does not allow us to obtain the relationships that can govern models of complex interactions.

• the exploration of new cause-effect relationships, in reaction to the mass of data produced by connected objects.

This principle is made concrete through methods, such as k-means which attempt to determine correspondences by minimizing the energy difference between points of the same group, a diagram of which is visible in Figure 3.4. Various applications can thus result, such as:

• grouping by data classes, in order to automatically define new annotations.

• downsizing, in order to keep only essential information. The bag-of-words method is an example originally intended for text analysis [24] and used in fields such as skin image analysis[25].

• the discovery of relationships within information, and the most robust relationships between variables and dependencies

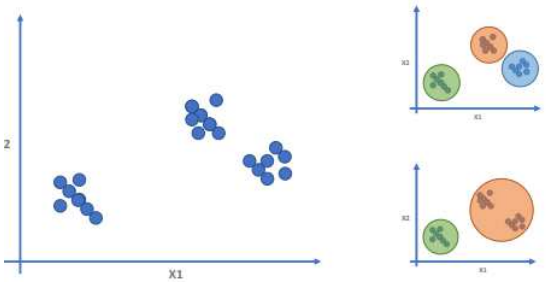


*Figure 1: Example of data not associated with prior annotations, and for which unsupervised approaches can be used in order to discover groups of data or clusters. The number of groups most often defined by the user before treatment, strongly influences the vision of the problem*

### 5.4 Semi-supervised learning

Semi-supervised learning techniques attempt to combine the previous two principles and are the consequence of the cost of annotating data which often requires expert work. Indeed, it can be difficult to have access to a ground truth that covers an entire dataset without losing the contribution of this unannotated information. This last point is particularly interesting in the context of large-dimensional data where the space of dimensions grows exponentially and dilutes the samples (known as the dimension bane).

Thus, semi-supervised learning is based on several assumptions with which the data must confront [27], including:

• a criterion of homogeneity, that is to say that data from a high density area share the same annotations,

• a low density separation criterion, that is, if there is a separation between several types of annotations, it is located in a low density area.

In practice, it is difficult to ensure that the data made available respects these commitments. In order to better visualize this concept, the example in Figure 3.5 demonstrates the interest of such approaches. This principle makes it possible to avoid extrapolating borders that are difficult to define in a situation of low density of labeled information.

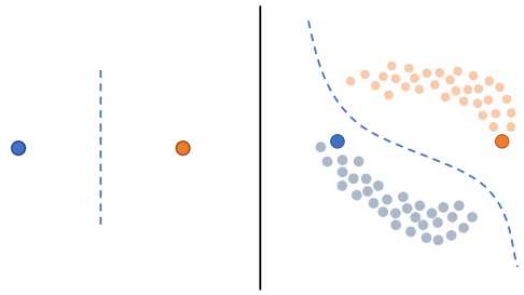

*Figure. 2.Example frequently used to demonstrate the value of semi-supervised learning from the point of view of the density principle. On the left, a classification obtained from two labeled data; On the right, the same situation with the addition of unlabeled data.*

### 5.5 Deep learning

Deep learning is a subset of machine learning, encouraged by recent contributions from neuroscience to the functioning and role of neurons in complex decision making [26]. This is because the brain has varying levels of information processing, which deep learning attempts to mimic by providing multiple layers of processing.

In the context of the classic learning methods mentioned in the previous part, the dimension of processing layers is not integrated. Simple learning approaches provide a three-layer architecture, one of the layers of which is allocated to the correlation between input and output data. In contrast, the deep learning approaches propose structures in n layers, with n ∈ N and n> 1. These intermediate layers are also qualified as hidden layers.

In a first subsection, a brief presentation of the principle of Artificial Neural Network, ANN (alt .: Artificial Neural Network, RNA) is made, then in a second subsection is detailed their extension to the Convolutional Neural Network, CNN ( alt .: Convolutional Neural Network, RNC). Finally, aspects related to the principle of knowledge transfer are proposed in a final subsection.

**Clustering**

The centers of clusters are represented by triangles, while the data points are represented by circles. The colors indicate membership in a cluster. We specified that we are looking for three clusters, so the algorithm was initialized by declaring three random data points as cluster centers.

In the presence of new data points, the k-means will affect each of them at the nearest cluster center. The following example shows the boundaries of the cluster centers that were learned previously.

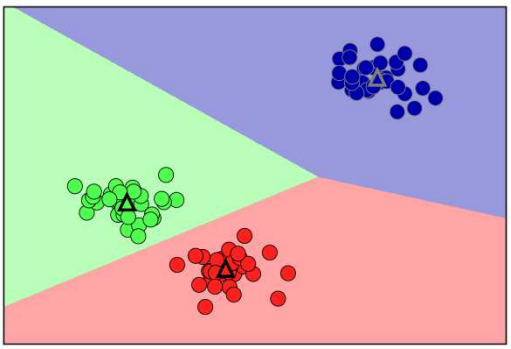

*Figure. 3.Cluster centers and cluster perimeters are found by the k-means algorithm.*

You can see that clustering is somewhat similar to classification, in that each item is labeled. However, there is no basic truth and therefore the labels themselves have no a priori meaning.

### 5.6 NLP - Natural language processing

1) Definition of natural language processing

Natural language processing (NLP) [15] is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language. NLP draws on many disciplines, including computer science and computational linguistics, to bridge the gap between human communication and computational understanding.

2) NLP in real world

NLP is an important component of a wide variety of software applications that we use in our daily lives.

3) Tasks of NLP

There is a collection of foundational tasks that frequently appear in various NLP projects. Due to their repetitive and fundamental nature, these tasks have been studied extensively.

a) Language modeling

This is the task of predicting what will be the next word in a sentence based on the history of previous words. The purpose of this task is to learn the probability of occurrence of a sequence of words in a given language. Language modeling is useful for developing solutions to a wide variety of problems, such as speech recognition, optical character recognition, handwriting recognition, machine translation, and spell checking.

b) Text classification

This involves classifying the text into a known set of categories based on its content. Text classification [16] is by far the most popular task in NLP and is used in a variety of tools, from spam identification to sentiment analysis.

c) Information extraction

As the name suggests, it's about extracting relevant information from a text, like calendar events in emails or the names of people mentioned in a social media post.

d) Information retrieval

This is the task of finding documents relevant to a user query from a large collection. Apps like Google Search are well-known use cases for information search.

e) Conversational agent

This is the task of building dialogue systems capable of conversing in human languages. Common applications of this task are Alexa, Siri, etc.

f) Text summarization

This task aims to create short summaries of longer documents while retaining the main content and preserving the general meaning of the text.

g) Question answering

It is about building a system capable of automatically answering questions asked in natural language.

h) Machine translation

It is about converting a text from one language to another. Tools like Google Translate are common applications for this task.

i) Topic modeling

It is about discovering the thematic structure of a large collection of documents. Thematic modeling is a common text exploration tool and is used in a wide range of fields, from literature to bioinformatics.

## 6. IMPLEMENTATION

### 6.1 Part of extracting information from Cvs

Resumes are a great example of unstructured data. Each CV has its own style of data formatting and many forms of data formatting. This makes reading difficult. Recruiters spend a lot of time going through resumes and selecting the best ones for their jobs. The giants of the big companies receive dozens of resumes for different positions every day, and recruiters cannot view every resume. That's why resume analyzers are a great deal for people like them. Resume the analyzers, it is easy to select the perfect CV from the many CVs received.

**1) Step 1: CV reading**

CVs do not have a fixed file format and can therefore be in any file format such as .PDF, .doc or .docx. So our main challenge is to read the resume

and convert it to plain text. For this, we can use two Python modules: pdfminer and doc2text.

These modules help extract text from .pdf and .doc, .docx file formats.

```
from pdfminer.converter import TextConverter
from pdfminer.pdfinterp import PDFPageInterpreter
from pdfminer.pdfinterp import PDFResourceManager
from pdfminer.layout import LAParams
from pdfminer.pdfpage import PDFPage
from pdfminer.pdfparser import PDFSyntaxError
import docx2txt
```

*Figure. 4.Import libraries that extract text from a PDF or DOC file.*

For this step, we create 3 functions each extracting text of a specified format (pdf, doc and docx)

```
def extract_text_from_docx(doc_path):
    try:
        temp = docx2txt.process(doc_path)
        text = [line.replace('\t', ' ') for line in temp.split('\n') if line]
        return ' '.join(text)
    except KeyError:
        return ' '
```

*Figure. 5.Function which extracts the text from the Docx file*

```
def extract_text_from_doc(doc_path):
    try:
        try:
            import textract
        except ImportError:
            return ' '
        text = textract.process(doc_path).decode('utf-8')
        return text
    except KeyError:
        return ' '
```

*Figure. 6.Function which extracts the text from the Doc file.*

```
def extract_text_from_pdf(pdf_path):
    if not isinstance(pdf_path, io.BytesIO):
        # extract text from local pdf file
        with open(pdf_path, 'rb') as fh:
            try:
                for page in PDFPage.get_pages(
                        fh,
                        caching=True,
                        check_extractable=True
                ):
                    resource_manager = PDFResourceManager()
                    fake_file_handle = io.StringIO()
                    converter = TextConverter(
                        resource_manager,
                        fake_file_handle,
                        laparams=LAParams()
                    )
                    page_interpreter = PDFPageInterpreter(
                        resource_manager,
                        converter
                    )
                    page_interpreter.process_page(page)

                    text = fake_file_handle.getvalue()
                    yield text

                    # close open handles
                    converter.close()
                    fake_file_handle.close()
```

*Figure. 7.Function that extracts text from PDF file.*

### 2) Step 2: Name extraction

To extract names from a CV, we could use regular expressions. But instead, we'll be using a more sophisticated tool which is SpaCy. Spacy is an industrial natural language processing module used for word and language processing. It comes with preformed templates for tagging, analyzing and recognizing features.

Our main motto is to use entity recognition (NER) to extract names (after all, name is an entity).

```
pip install -U spacy
```

*Figure. 8.Installation of SpaCy.*

**SpaCy Models**

SpaCy [17] is primarily pipeline driven, which is a very powerful design and provides access to various properties and operations of SpaCy. A pipeline is created by loading a SpaCy model. Without the installation of a template, SpaCy is virtually useless. The coolest thing about SpaCy's models, unlike any other natural language processing framework and library, is that they are also treated like Python packages and can be installed just like any other Python module.

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
$ python -m spacy download fr_core_news_sm


>>> import spacy
>>> nlp = spacy.load("fr_core_news_sm")
```

*Figure. 9.Installation and use of a spaCy model.*

Rules Based Match: Spacy gives us the ability to process text or language based on criteria based match.

We will use this feature of spaCy to extract the first and last name from the CV.

```
def extract_name(nlp_text, matcher):

    pattern = [{'POS': 'PROPN'}, {'POS': 'PROPN'}]
    matcher.add('NAME', [pattern])

    matches = matcher(nlp_text)

    for _, start, end in matches:
        span = nlp_text[start:end]
        if 'name' not in span.text.lower():
            return span.text
```

*Figure. 10.Function for extracting name from a CV.*

As we can see above, we first defined a template that we want to look for in the CV. We first created a simple model based on the fact that a person's first and last name are always proper names. Using SPACY, we researched this template in the CV taking advantage of the NLP process of part-of-speech tagging that spaCy offers.

### 3) Step3: Phone number extraction

To extract the phone numbers, we used regular expressions. The telephone numbers can have several forms, with or without code, separated by a hyphen or by a space etc…, It is therefore necessary to define a generic regular expression which can correspond to all the similar combinations of telephone numbers.

Regular Expressions (RE)

RE are a very powerful and very fast system for searching strings. It's kind of a very extensive Find-Replace feature.

*Fig. 11. Function for extracting the phone number from a CV.*

### 4) Step 4: Extraction of the e-mail address

To extract the email from the CV, we can use a similar approach as used to extract the phone numbers.

```
import re #importer la bibliothèque d'expressions régulières
def extract_email(text):
    email = re.findall(r"([^@|\s]+@[^@]+\.[^@|\s]+)", text)
    if email:
        try:
            return email[0].split()[0].strip(';')
        except IndexError:
            return None
```

*Figure. 12.function for extracting the e-mail address from a CV.*

### 5) Step 5: Age extraction

To extract the age of the CV we will use regular expressions

But this time there is more than one possibility to extract this value, it can be written in a simple way, for example: ”23 years old”, or we can find the date of birth and calculate the age at from this one.

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```python
def extract_age(text):
    age = re.findall(r"\d\d ans", text)
    if age:
        try:
            return age[0].split()[0]
        except Exception:
            pass

    naissance = re.findall(r"Né[e]? le .*\d\d\d\d", text)
    if naissance:
        try:
            text_year_of_birth = naissance[0]
            year_of_birth = text_year_of_birth[len(text_year_of_birth)-4:
            year = datetime.datetime.now().year
            result= year - int(year_of_birth)
            if (result > 18 and result < 80):
                return result
        except Exception:
            return None
```

*Figure. 13.Function for extracting age from a CV.*

### 6) Step 6: Skills extraction

After extracting the basic information from the CV, we need to extract what makes the difference between a candidate and another "Skills".

We extracted the skills using the tokenization technique of turning a text into a series of individual tokens.

Before implementing tokenization, we created a dataset that contains all the possible skills that could exist in a CV and against which we can compare the skills of a given CV. This file can always be updated to add or remove skills according to business needs.

```
java
HTML
CSS
react
angular
jasper
datalake
datawarehouse
datastage
ssis
power
talend
drupal
elastic
hook
solr
vue
Spring
JAVASCRIPT
C#
MVC
ASP
```

*Figure. 14.Competency data file.*

```python
import pandas as pd
from collections import defaultdict


def extract_skills(nlp_text, noun_chunks):
    tokens = [token.text for token in nlp_text if not token.is_stop]
    data = pd.read_csv('data/skills_all.txt', header=None,usecols=[0], encoding='utf-8')
    skills = list(data[0].values.tolist())

    skillset = defaultdict(int)
    # vérifier par one-grams
    for token in tokens:
        if token.lower() in map(str.lower, skills):
            skillset[token.capitalize()] += 1

    # vérifier par bi-grams et tri-grams

    for token in noun_chunks:
        token = token.text.lower().strip()
        if token in map(str.lower, skills):
            skillset[token.capitalize()] += 1

    return sorted(skillset.items(), key=lambda w:w[1], reverse=True)
```

*Figure. 15. Function for extracting the list of skills exists in a CV.*

### 7) Step 7: Extraction of studies

After extracting the skills from the CV, it is now necessary to extract the details of the studies.

For each CV training, we will try to obtain the start date, the end date, the school and the diploma obtained by this training (BTS, license, master, etc…).

The output of each training would be like this:

```
studies :
  [
      {
          'school': 'Université Internationale de Casablanca',
          'end_year': '2017',
          'start_year': '2017',
          'diplome': 'Diplôme d'ingénieur en informatique ( option SI )',
          'bac+': 5
      },
      {   'school': 'Université Cadi Ayyad',
          'end_year': '2021',
          'start_year': '2021',
          'diplome': 'Licence en sciences mathématiques et informatiques',
          'bac+': 3
      }
  ]
```

*Figure. 16. Expected result of the extraction of studies.*

a) Procedures for obtaining the studies

First of all we define two constants, the first contains all the possible values for the title of the training part in the CV, and the 2nd constant represents all the possible values of the title of all the other parts except the formations.

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
FORMATIONS_KEYWORDS = ['education','éducation', 'éducations','educations','diplômes','formation','formations',
                       'parcours academique','parcours académique','etudes','études','parcours universitaire']
NON_FORMATIONS_KEYWORDS = ['expérience','expériences','expérience','expériences',
                   'intérêts','interets','publications','projets','compétences','competences','certifications',
                   'objectif', 'langues']
```

*Figure. 17.Constants possible titles of a study in a CV.*

After that, we use the constants defined to know the perimeter of the training section in the CV.

We have also prepared a file which contains a list of all Moroccan schools and universities separated by a new line, each line contains the name of the school plus all of its abbreviations.

```
INPT,L'institut National des Postes et Télécommunications,Institut National des Postes et Télécommunications
EMI,EMI Rabat,Ecole Mohammadia d'Ingénieurs
ENSIAS,Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes Rabat
EHTP,Centre D'Enseignement Des Sciences Appliquées
ESITH,ÉCOLE SUPÉRIEURE DES INDUSTRIES DU TEXTILE ET DE L'HABILLEMENT
AIAC,Académie internationale Mohammed VI de l'aviation civile
ENSA,École nationale des sciences appliquées,ENSAT,ENSAA,ENSAMA,ENSAO,ENSAB,ENSAS,ENSAF,ENSAKH,ENSAH,ENSAJ,ENSAK,ENSATE
EST,Ecole supérieure de technologie ,ESTC,ESTB,ESTBM,ESTG,ESTK,ESTL,ESTS,ESTE,ESTO,ESTF,ESTM,ESTS,ESTA
FP,Faculté Polydisciplinaire,FPK,FPS
FS,Faculté de sciences,FSAC,FSJ,FSF,FSK,FSM,FSO,FSR,FSM
FST,Faculté des Sciences et Techniques,Faculté Sciences et Techniques,FSTB,FSTE,FSTF,FSTM,FSTS,FSTT
HESTIM,Hautes études des sciences et techniques de l'ingénierie et du managament
UIR,Université Internationale de Rabat
UPM,Université Privée de Marrakech
```

*Figure. 18.List of universities and schools.*

Using the latter and using SpaCy NLP matcher, which is PhraseMatcher which matches sequences of words, based on documents we could identify the name of the school or university from the predefined part of the CV that we defined at the beginning.

And after marking all the schools and universities from the CV, the rest is to try to get other information related to this study like the start date, the end date and the degree obtained, and the number of years of 'studies after the baccalaureate.

To extract all this information, we used many techniques ranging from NLP to regular expressions.

```
import spacy
from spacy.matcher import PhraseMatcher
from spacy.tokens import Span
import constants
import datasource
import helpers

def extract_studies(text):
    nlp = spacy.load("fr_core_news_sm")
    doc = nlp(text)

    #Matcher to get studies part
    matcher_part = PhraseMatcher(nlp.vocab, attr='LOWER')
    patterns_part = list(nlp.pipe(constants.FORMATIONS_KEYWORDS))

    matcher_part.add("Schools", patterns_part)
    matched_part = matcher_part(doc)
    start_index = 0
    for match_id, start, end in matched_part:...
    #Matcher to know when studies part ends
    matcher_part = PhraseMatcher(nlp.vocab, attr='LOWER')
    patterns_part = list(nlp.pipe(constants.NON_FORMATIONS_KEYWORDS))
    matcher_part.add("OtherParts", patterns_part)
    matched_part = matcher_part(doc[start_index:])
    end_index = len(doc[start_index:])
    for match_id, start, end in matched_part:
        end_index = start
        break


    #Matcher to match schools
    matcher = PhraseMatcher(nlp.vocab, attr='LOWER')
    patterns = list(nlp.pipe(datasource.get_schools()))
    #print(doc[start_index end_index])
    matcher.add("Schools", patterns)
    matches = matcher(doc[start_index:end_index])
    if len(matches) == 0:
        matches = matcher(doc[start_index:])

    studies = []
    from_index = start_index
    date_first = True

    for match_id, start, end in matches:
        #print(start)
        from_index = start
        # print(from_index)
        study = {}
        # Create a Span with the label for "SKILL"
        span = Span(doc, start, end, label="SCHOOL")
        # doc.ents += (span,)
        study['school'] = span.text
        # print(from_index,start)
        year = helpers.getYear(nlp.vocab, doc[start_index:start])

        if (len(year) > 0 and date_first):
            _id, _start, _end = year[-1]
            year_value = doc[start_index + _start: start_index + _end]
            study['end_year'] = year_value.text
            for i in range(1, 5):
```

*Figure. 19.Part of the function that extracts the studies.*

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

### 6.2 Classification part of CVs

The information extracted from the CV contains the following columns:

- age: the candidate's age
- year_experience: the total number of years of experience
- bac_plus: number of years of study after the baccalaureate
- skills: a list of skills that the candidate has
- Interview: whether the candidate qualified for the interview or not

| age | annee_experience | bac_plus | skills | entreti |
|---|---|---|---|---|
| 28 | 9 | 3 | ['Radius', 'Spring', 'Support', 'Java', 'Maintenance', 'Javascript', 'C#', 'Cisco', 'Active directory', 'Objective'] | |
| 27 | 4 | 2 | ['Vlan', 'Sql', 'Firewall', 'Radius', 'VMware', 'Ip', 'Lan', 'Switch', 'Francais', 'Vue', 'C#', 'AWS', 'Android', 'Rï¿½eau'] | |
| 35 | 8 | 3 | ['AWS', 'Ionic', 'Maintenance', 'Jquery', 'C#', 'San', 'Switch', 'Php'] | |
| 32 | 9 | 1 | ['VMware', 'Javascript', 'Html', 'Ionic', 'Dhcp', 'Vlan', 'Hardware'] | |
| 27 | 7 | 2 | ['Power', 'Switch', 'Css', 'C#', 'Sql', 'Php', 'Linux'] | |
| 22 | 4 | 5 | ['TCP IP', 'Linux', 'Rï¿½eau', 'Radius', 'Ip', 'Gpo'] | |
| 28 | 4 | 4 | ['Firewalls', 'Kubernetes', 'Asp', 'Android', 'VMware'] | |
| 30 | 7 | 2 | ['Kubernetes', 'Support', 'Azure', 'Dhcp'] | |
| 24 | 9 | 5 | ['Hardware', 'San', 'Rï¿½eau', 'Fortigate', 'Css', 'Php', 'Java'] | |
| 31 | 9 | 0 | ['Wsus', 'Vpn', 'Linux', 'Node', 'Javascript', 'Vlan', 'Android', 'Ios', 'Azure', 'Kubernetes', 'Datawarehouse'] | |

*Figure. 20.Extract from the list of data.*

#### 6.2.1. Data analysis

Here is the crucial step in which we will have to clean the data. This is necessary to refine the variables so that they better adapt to machine learning algorithms.

a) Importing bookstores

The first step is to import the libraries that we will use to import, cleanse, visualize, and prepare our data for machine learning. In our case, we are going to work with three libraries which are absolutely essential for the development of our project, they are:

- Pandas: is a fast, powerful, flexible and easy-to-use open source data analysis and manipulation tool built from the Python programming language.
- Matplotlib: is a comprehensive library for creating static, animated and interactive visualizations in Python.
- Seaborn: is a Python data visualization library based on matplotlib. It provides a high level interface for drawing attractive and informative statistical graphs.

```
# Importation les libraries
import pandas as pd
import ast
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

*Figure.21. Import libraries for data preparation and visualization.*

b) Datasets importation

To import the dataset, we'll use the pandas library to read the dataset and assign the result to a variable that will represent it throughout the project.

```
df = pd.read_csv('elo.csv', encoding='cp1252')
```

```
df['skills']=df['skills'].apply(ast.literal_eval)
```

```
df.head()
```

| | age | annee_experience | bac_plus | skills | entretien |
|---|---|---|---|---|---|
| 0 | 28 | 9 | 3 | [Radius, Spring, Support, Java, Maintenance, J... | 1 |
| 1 | 27 | 4 | 2 | [Vlan, Sql, Firewall, Radius, VMware, Ip, Lan,... | 1 |
| 2 | 35 | 8 | 3 | [AWS, Ionic, Maintenance, Jquery, C#, San, Swi... | 0 |
| 3 | 32 | 9 | 1 | [VMware, Javascript, Html, Ionic, Dhcp, Vlan, ... | 0 |
| 4 | 27 | 7 | 2 | [Power, Switch, Css, C#, Sql, Php, Linux] | 0 |

*Figure. 22. Importing data by Pandas.*

c) First glimpse of the data

We will get some basic information about the data using Pandas info and describe methods.

```
df.describe()
```

| | age | annee_experience | bac_plus | entretien |
|---|---|---|---|---|
| count | 76.000000 | 76.000000 | 76.000000 | 76.000000 |
| mean | 29.276316 | 4.263158 | 2.710526 | 0.434211 |
| std | 3.733091 | 3.180643 | 1.663857 | 0.498946 |
| min | 22.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 27.000000 | 1.000000 | 1.000000 | 0.000000 |
| 50% | 29.000000 | 4.000000 | 3.000000 | 0.000000 |
| 75% | 32.000000 | 7.000000 | 4.000000 | 1.000000 |
| max | 35.000000 | 9.000000 | 5.000000 | 1.000000 |

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 76 entries, 0 to 75
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   age               76 non-null     int64
 1   annee_experience  76 non-null     int64
 2   bac_plus          76 non-null     int64
 3   skills            76 non-null     object
 4   entretien         76 non-null     int64
dtypes: int64(4), object(1)
memory usage: 3.1+ KB
```

*Figure. 23. Result of the info and describe method on our dataset.*

#### 6.2.2. Data preparation

Machine learning algorithms are almost always optimized for raw, detailed source data. The data environment must therefore provide large amounts of raw data for discovery-oriented analysis

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

practices, such as data mining, data mining, statistics, and machine learning.

a) Missing DATA

Many real-world datasets can contain missing values for a variety of reasons. These are often encoded as NaN, blanks, or other substitutes. Training a model with a dataset that contains many missing values can have a significant impact on the quality of the machine learning model.

For our dataset, a single missing value can have a huge impact on our results. To solve this problem, we are going to treat each column separately that has a missing value; for example, in the "age" column, we are going to fill the missing values with the average of all the values. On the other hand, to process the columns “annee_experience” and “bac_plus”, we will simply fill the missing values with 0. And finally for the target variable “X” we will delete the whole row which has an undefined target value :

```
#remplir les âges nuls avec la moyenne de la colonne
df['age'].fillna((df['age'].mean()), inplace=True)
#remplir le NA de la colonne annee_experience avec des 0
df['annee_experience'].fillna(0, inplace=True)
#remplir le NA de la colonne bac_plus avec des 0
df['bac_plus'].fillna(0, inplace=True)
#supprimer toutes les lignes où la variable cible est nulle
df = df[df['entretien'].notna()]
```

*Figure. 24. Treatment of missing data.*

b) Processing of the skills

From the data obtained, we see that the "skills" column contains a much more complex type than the other columns, for each record the value "skills" represents a list containing all the skills mentioned in the candidate's CV. And unfortunately, machine learning algorithms cannot process these types of data. So we need to find a way to turn this data into an acceptable form for machine learning algorithms.

The approach used to solve this problem:

The first step is to prepare a dataset containing many job titles with all the skills to be taken into account for the specified job, in order to optimize our machine learning model so that it is based only on important skills and not ignore irrelevant skills when training the model.

```
skills_par_poste = pd.read_csv('skills_per_poste.csv')
skills_par_poste['skills']=skills_par_poste['skills'].apply(ast.literal_eval)
```

```
skills_par_poste.head()
```

| | poste | skills |
|---|---|---|
| 0 | administrateur réseau | [Radius, Wsus, Active directory, Dhcp, Azure, ... |
| 1 | développeur .net | [C#, ASP.NET, MVC, Javascript, Jquery, Html, C... |
| 2 | développeur java | [java, jee, spring, Javascript, Jquery, Html, ... |

*Figure. 25. Skills dataset by workstation.*

After having prepared the list of skills by job position, to prepare our model, we need to base ourselves on the skills of a specific job offer. We take the network administrator in this example.

```
#acquérir les compétences d'un poste de travail spécifique
#nous prenons l'administrateur réseau par exemple
offre_skills=skills_par_poste[skills_par_poste['poste']=='administrateur réseau']['skills'][0]
```

```
offre_skills
```

```
['Radius',
 'Wsus',
 'Active directory',
 'Dhcp',
 'Azure',
 'Switch',
 'San',
 'Gpo',
 'Kubernetes',
 'Cisco',
 'Fortigate',
 'Firewall',
 'Maintenance',
 'Réseau',
 'Anglais',
 'Français',
 'AWS',
 'VMware',
 'TCP IP',
 'Hardware',
 'Lan',
 'Linux',
 'Support',
 'Vpn',
 'Vlan']
```

*Figure. 26. List of skills for a specific position.*

After preparing the list of important skills for a job, we update the data frame by creating a new column for each skill in the list and fill it with 1 if the candidate has the specified skill and 0 if does not have it. By doing this, we can remove the “skills” column since we have its data (the skills) distributed in several columns with a format that could be processed by machine learning algorithms.

```
for item in offre_skills:
    df[item] = df['skills'].apply(lambda x: int(item in x))
```

```
del df['skills']
```

*Fig. 27. List of skills processing.*

```
df.head()
```

| | age | annee_experience | bac_plus | skills | entretien |
|---|---|---|---|---|---|
| 0 | 28 | 9 | 3 | [Radius, Spring, Support, Java, Maintenance, J... | 1 |
| 1 | 27 | 4 | 2 | [Vlan, Sql, Firewall, Radius, VMware, Ip, Lan, ... | 1 |
| 2 | 35 | 8 | 3 | [AWS, Ionic, Maintenance, Jquery, C#, San, Swi... | 0 |
| 3 | 32 | 9 | 1 | [VMware, Javascript, Html, Ionic, Dhcp, Vlan, ... | 0 |
| 4 | 27 | 7 | 2 | [Power, Switch, Css, C#, Sql, Php, Linux] | 0 |

```
for item in offre_skills:
    df[item] = df['skills'].apply(lambda x: int(item in x))
del df['skills']
```

```
df.head()
```

| | age | annee_experience | bac_plus | entretien | Radius | Wsus | Active directory | Dhcp | Azure | Switch | ... | Français | AWS | VMware | TCP IP | Hardware | Lan | Linux | Supp |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 28 | 9 | 3 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 27 | 4 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | ... | 0 | 1 | 1 | 0 | 0 | 1 | 0 | |
| 2 | 35 | 8 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | |
| 3 | 32 | 9 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 1 | 0 | 0 | |
| 4 | 27 | 7 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |

*Figure. 28. Data before and after the dissemination of skills.*

### 6.2.3. Component analysis

The main idea of principal component analysis is to reduce the dimensionality of a dataset. An easy way to draw a heat map in Python is to import and implement the Seaborn library.

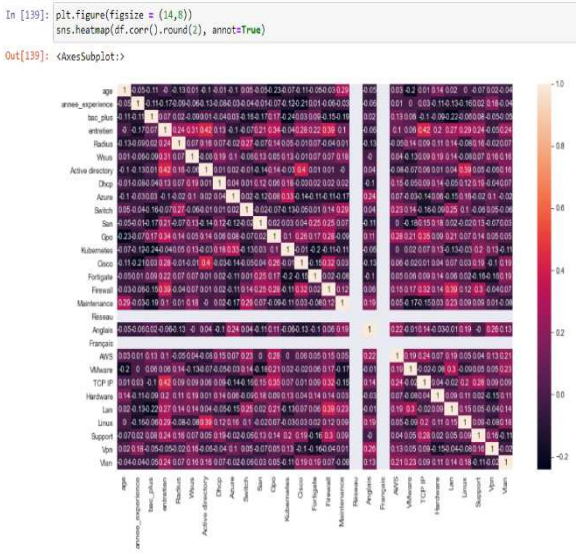


*Figure. 29. Heatmap for checking the correlation between variables.*

b) Data Relationship

Visualization of the output distribution

Interview = 0: the candidate is not qualified to take the interview

Interview = 1: the candidate is qualified to take the interview

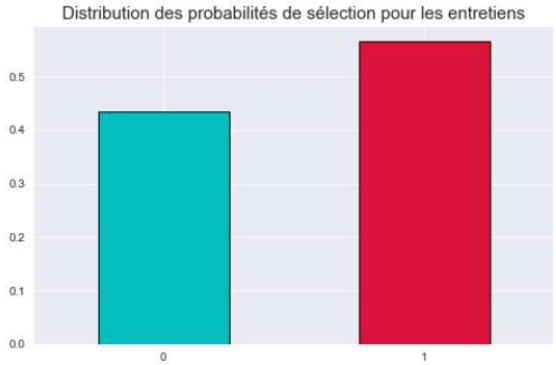


*Figure. 30. Distribution of selection probabilities for interviews.*

The graph shows us the probability of having selected for an interview.

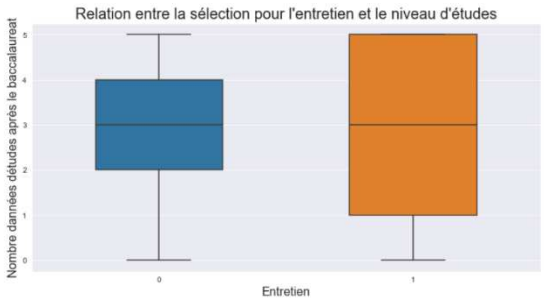


*Figure. 31. Relationship between interview selection and educational level.*

The graph shows us the relationship between the level of education and the probability of having selected for an interview.

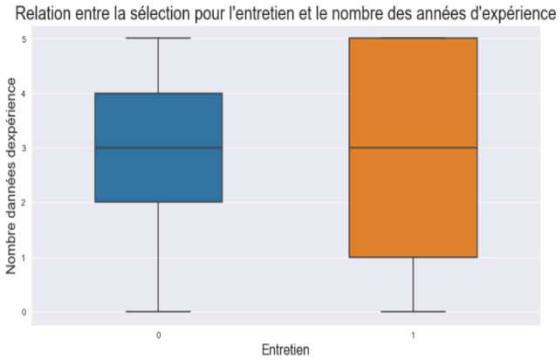


*Figure. 32. Relationship between selection for interview and number of years of experience.*

The graph shows us the relationship between the number of years of experience and the probability of having selected for an interview.

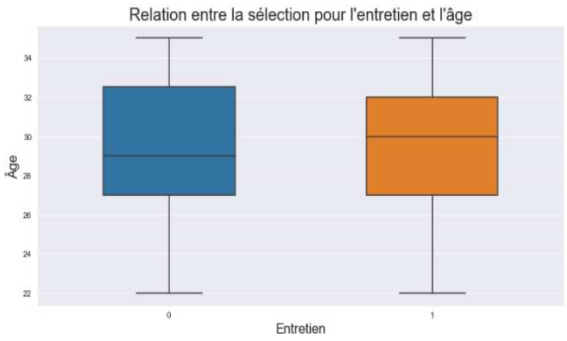


*Figure. 33. Relationship between interview selection and age.*

The graph shows us the relationship between age and the probability of having selected for an interview.

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

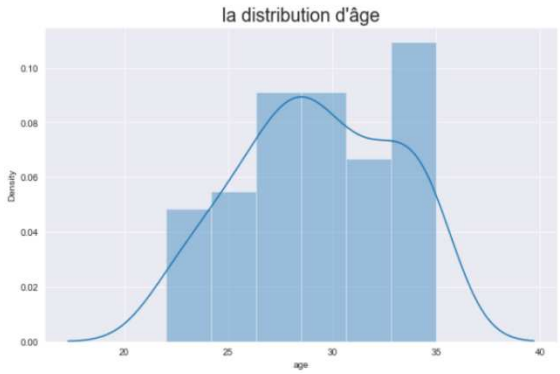


*Figure. 34. Age distribution.*

We have found relationships between the different attributes. This will allow us to have a better prediction, and since we now have a better view of our data and understand their relationship, we can take the next step to perform the scaling and splitting. (splitting).

### 6.2.4. Scaling et Fitting

Scaling and Standardization: This is a data preprocessing step that is applied to independent variables or data characteristics. Essentially, it allows data to be normalized to a particular interval.

In our case we imported MinMaxScalar from the sklearn library to standardize the columns "Age", "annee_experience" and "bac_plus".

```
In [202]: from sklearn.preprocessing import MinMaxScaler
          # Scalling Age & annee_experience &
          scaler = MinMaxScaler()
          df['age'] = scaler.fit_transform(df[['age']])
          df['annee_experience'] = scaler.fit_transform(df[['annee_experience']])
          df['bac_plus'] = scaler.fit_transform(df[['bac_plus']])

          df.head()
```

Out[202]:

| | age | annee_experience | bac_plus | entretien | Radius | Wsus | Active directory | Dhcp | Azure | Switch | ... | Français | AWS | VMware | TCP IP | Hardware | Lan | Linux |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.461038 | 1.000000 | 0.6 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0.384615 | 0.444444 | 0.4 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | ... | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 2 | 1.000000 | 0.888889 | 0.6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0.769231 | 1.000000 | 0.2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 4 | 0.384615 | 0.777778 | 0.4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

*Figure. 35. Scalling some columns of the dataframe.*

### 6.2.5. Splitting of the dataset

After checking the relationships and correlation in the previous steps, we can now divide the data to train the model.

Python Scikit-Learn [19,32] provides a very handy function for splitting datasets, called "train_test_split".

Here we are asking for a split of the dataset of 30% for test data and the rest for training.

```
from sklearn.model_selection import train_test_split
y = df['entretien']
columns = ['age','annee_experience','bac_plus'] + offre_skills
X = df[columns]
# setting X and y for train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 0)
```

*Figure. 36. Divide data into test and trainning data*

### 6.2.6. Modeling

a) Logistic regression

Instead of predicting exactly 0 or 1, logistic regression generates a probability: a value between 0 and 1, exclusive. Take for example our logistic regression model for classifying candidates for a job posting. If the model infers a value of 0.893 on a specific candidate, the probability that the candidate will be suitable for the job is 89.3%. Specifically, this means that within the limit of infinite learning examples, the set of examples for which the model predicts 0.893 will indeed be a suitable candidate for the job 89.3% of the time, and will not be suitable for the job. post the remaining 10.7% of the time.

```
from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression().fit(X_train, y_train)
print(lr_model.score(X_test, y_test))
```

```
0.782608695652174
```

```
scores = []
for i in [0.0001, 0.001, 0.01, 0.1, 1, 2, 5, 10]:
    lr_model = LogisticRegression(C = i).fit(X_train, y_train)
    scores.append(lr_model.score(X_test, y_test))

plt.figure(figsize = (12,5))
plt.plot(scores)
plt.title('Scores de la classification de logistic regression - C = 0.0001 à 10', fontsize = '20')
plt.xlabel('C', fontsize = 15)
plt.ylabel('Score', fontsize = 15)
plt.xticks([0.0001, 0.001, 0.01, 0.1, 1, 2, 5, 10])

print('Max score: {:.4f}'.format(max(scores)))
```

```
Max score: 0.7826
```

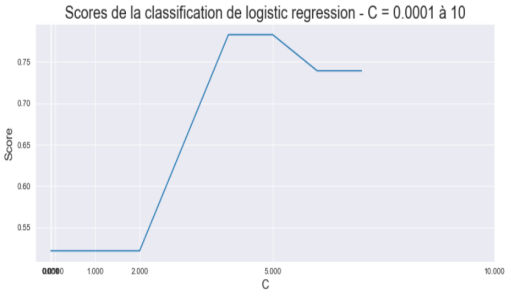


*Figure.37.Logistic regression classification scores*

In the model tuning part, we tried several values for the hyperparameter C in order to find the best one for a better model.

### 6.2.7. Scenario of a real case

In a real case, the model will take a list of candidates for a job offer with all their information and skills. Then, for each candidate, the model predicts the probability of a match with the workstation.

| | nom | age | annee_experience | bac_plus | skills |
|---|---|---|---|---|---|
| 0 | Safia Baali | 27 | 4 | 5 | [c#, java, asp, mvc] |
| 1 | Amine Tace | 26 | 2 | 2 | [Radius, Wsus, Active directory, Dhcp, Azure, ... |
| 2 | Ibrahim Hamzane | 30 | 3 | 5 | [Vpn, Vlan, Active directory, Cisco, linux, An... |
| 3 | Aymane Alerlaly | 25 | 0 | 3 | [Cisco, Maintenance, Radius, linux, AWS, Activ... |
| 4 | Ahmed Talee | 20 | 0 | 0 | [] |

*Figure. 38. The list provided to the model*

*Using our model.*

we predict the probability of matching a candidate with a job and we add its value in a new column called match which is a percentage by which we sort the values from the candidate most matching the job to the candidate the least. corresponding.

| | nom | age | annee_experience | bac_plus | skills | matching |
|---|---|---|---|---|---|---|
| 1 | Amine Tace | 26 | 2 | 2 | [Radius, Wsus, Active directory, Dhcp, Azure, ... | 72.573580 |
| 3 | Aymane Afertaly | 25 | 0 | 3 | [Cisco, Maintenance, Radius, linux, AWS, Activ... | 72.493539 |
| 2 | Ibrahim Hamzane | 30 | 3 | 5 | [Vpn, Vlan, Active directory, Cisco, linux, An... | 39.206366 |
| 4 | Ahmed Talee | 20 | 0 | 0 | [] | 14.370613 |
| 0 | Safia Baali | 27 | 4 | 5 | [c#, java, asp, mvc] | 12.050031 |

*Figure. 39. The list of results using the model.*

### 6.3 Evaluation & comparison with other existing systems

In this section, we present a comparison between our system and existing resume recommender systems. According to this experiment, the results obtained show that our system has a score of 92%, compared to another CV recommender system based on Big Data technologies such as Spark, Apache Pig, and Apache Hive[28,29,30] which achieved a score of 87% despite the fact that the latter is very powerful in terms of processing massive data. If recommendation tools have begun to be used at the human resources level, it is Big Data technologies and the power of statistical learning algorithms that have considerably improved their performance. Although the two approaches described above have benefited from developments in these two fields, Today, modern techniques of exploration (data mining) and prediction (machine learning), combined with a Big Data infrastructure allowing the exploitation massive data [31], have given rise to hybrid strategies to circumvent this problem: the recommendation is developed in the air of a tailor-made model, borrowing from the two types of approaches.

## 7. CONCLUSIONS

The hiring process has evolved over time. It refers to the process of finding, selecting and recruiting new employees to a company through CVs. We have presented in this paper a solution in the form of an automated and flexible system to extract vital information from unstructured CVs and transform it into a structured format and then use it in a recommendation system according to the available positions.

## REFERENCES:

[1] Das, Papiya, Manjusha Pandey, and Siddharth Swarup Rautaray. "A CV parser model using entity extraction process and big data tools." IJ Information Technology and Computer Science 9 (2018): 21-31.

[2] Das, Papiya, Manjusha Pandey, and Siddharth Swarup Rautaray. "A CV parser model using entity extraction process and big data tools." IJ Information Technology and Computer Science 9 (2018): 21-31.

[3] Banane, Mouad. "Real-Time Semantic Web Data Stream Processing Using Storm." 2020 International Conference on Computing and Information Technology (ICCIT-1441). IEEE, 2020.

[4] Banane, Mouad, and Abdessamad Belangour. "A Big Data Solution To Process Semantic Web Data Using The Model Driven Engineering Approach." International Journal of Scientific & Technology Research 9.02 (2020).).

[5] Ferguson, Mike. "Architecting a big data platform for analytics." A Whitepaper prepared for IBM 30 (2012).

[6] Baali, Safia, et al. "A Multi-Criteria Analysis and Advanced Comparative Study of Recommendation Systems."International Journal of Engineering Trends and Technology 69.3(2021):69-75.

[7] Baali, Safia, Hicham Moutachaouik, and Abdelaziz Marzak. "Toward a Recommendation System: Proposition of a New Model to Measure Competences Using Dimensionality Reduction." International Conference on Smart Applications and Data Analysis. Springer, Cham, 2020.

[8] Ossadnik, Wolfgang, and Oliver Lange. "AHP-based evaluation of AHP-Software." European journal of operational research 118.3 (1999): 578-588.

[9] Keijzer, Niels. "Feigned ambition. Analysing the emergence, evolution and performance of the ACP Group of States." Third World Thematics: A TWQ Journal 1.4 (2016): 508-525.

[10] Turing, Alan M. "Computing machinery and intelligence." Parsing the turing test. Springer, Dordrecht, 2009. 23-65.

[11] Robert, Christian. "Machine learning, a probabilistic perspective." (2014): 62-63.

[12] Foulds, James, and Eibe Frank. "A review of multi-instance learning assumptions." The knowledge engineering review 25.1 (2010): 1-25.

[13] Wang, Junping, Quanshi Chen, and Yong Chen. "RBF kernel based support vector machine with universal approximation and its application." International symposium on neural networks. Springer, Berlin, Heidelberg, 2004.

[14] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." Emerging

artificial intelligence applications in computer engineering 160.1 (2007): 3-24.

[15] Kang, Yue, et al. "Natural language processing (NLP) in management research: A literature review." Journal of Management Analytics 7.2 (2020): 139-172.

[16] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. and Brown, D., 2019. Text classification algorithms: A survey. Information, 10(4), p.150.

[17] Schmitt, Xavier, et al. "A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate." 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS). IEEE, 2019.

[18] Breiman, Leo, et al. Classification and regression trees. Routledge, 2017.

[19] Hao, Jiangang, and Tin Kam Ho. "Machine learning made easy: a review of scikit-learn package in python programming language." Journal of Educational and Behavioral Statistics 44.3 (2019): 348-361.

[20] Quinlan, J. Ross. "Induction of decision trees." Machine learning 1.1 (1986): 81-106.

[21] Kononenko, Igor. "Bayesian neural networks." Biological Cybernetics 61.5 (1989): 361-370.

[22] Zhang, Harry. "The optimality of naive Bayes." AA 1.2 (2004): 3

[23] Cover, Thomas, and Peter Hart. "Nearest neighbor pattern classification." IEEE transactions on information theory 13.1 (1967): 21-27

[24] Zhang, Yin, Rong Jin, and Zhi-Hua Zhou. "Understanding bag-of-words model: a statistical framework." International Journal of Machine Learning and Cybernetics 1.1-4 (2010): 43-52.

[25] Situ, Ning, et al. "Malignant melanoma detection by bag-of-features classification." 2008 30th annual international conference of the IEEE engineering in medicine and biology society. IEEE, 2008.

[26] Quartz, Steven R., and Terrence J. Sejnowski. "The neural basis of cognitive development: A constructivist manifesto." Behavioral and brain sciences 20.4 (1997): 537-556.

[27] Zhu, Xiaojin, and Andrew B. Goldberg. "Introduction to semi-supervised learning." Synthesis lectures on artificial intelligence and machine learning 3.1 (2009): 1-130

[28] Banane, M., & Belangour, A. (2020). A new system for massive RDF data management using Big Data query languages Pig, Hive, and Spark. International Journal of Computing and Digital Systems, 9(2), 259-270.

[29]. Erraissi, A. (2021). Using model Driven Engineering to transform Big Data query languages to MapReduce jobs. International Journal of Computing and Digital Systems, 10, 619-628.

[30]. A. Erraissi and M. Banane, "Managing Big Data using Model Driven Engineering: From Big Data Meta-model to Cloudera PSM meta-model," 2020 International Conference on Decision Aid Sciences and Application (DASA), 2020, pp. 1235-1239, doi: 10.1109/DASA51403.2020.9317292.

[31]. A. Erraissi, M. Banane, A. Belangour and M. Azzouazi, "Big Data Storage using Model Driven Engineering: From Big Data Meta-model to Cloudera PSM meta-model," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-5, doi: 10.1109/ICDABI51230.2020.9325674.

[32]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.