Titre: Title:	Local VS. Global Models for Job-Candidate Matching
Auteur: Author:	Javier Abraham Rosales Tovar
Date:	2020
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Rosales Tovar, J. A. (2020). Local VS. Global Models for Job-Candidate Matching [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/5422/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/5422/
Directeurs de recherche: Advisors:	Bram Adams
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Local VS. Global Models for Job-Candidate Matching

JAVIER ABRAHAM ROSALES TOVAR
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Août 2020

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Local VS. Global Models for Job-Candidate Matching

présenté par **Javier Abraham ROSALES TOVAR**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Amal ZOUAQ, présidente
Bram ADAMS, membre et directeur de recherche
Jinghui CHENG, membre

DEDICATION

To my lovely mother Leticia Tovar, who has been supporting me in all possible ways, even sometimes doing the impossible, possible. I wont have enough life to thank you for all you have done for me.

Thanks to everyone that contributes to the free software community and everyone who contributes to an inclusive, free and open society, you are my inspiration.

ACKNOWLEDGEMENTS

I am really thankful to had Dr. Bram Adams as my advisor, who bring to me a life changing opportunity to continue my development and from who i learned a lot and who was always supportive. I would like to extend this acknowledgment to the jury members who review my thesis.

Thanks to my mom Leticia Tovar who has been supporting me in all possible ways so i can be here, and who has always work hard and has teach me to overcome any problem that can present in life.

Thanks to the Airudi company and specially to Amanda Arciero for providing us with the necessary data to perform our study and for being always pendant on the necessities for the approach development.

RÉSUMÉ

Avec le développement des technologies de l'information et la croissance continue du marché du recrutement électronique, l'automatisation du processus de sélection pour trouver le meilleur candidat pour un poste a suscité l'intérêt des chercheurs et des ingénieurs en logiciels ce qui a conduit au développement de modèles complexes, d'algorithmes et de techniques qui exploitent le traitement du langage naturel, la similitude sémantique et l'apprentissage automatique. Cette thèse vise à compléter ce travail, en se concentrant sur la façon d'exploiter les données existantes pour améliorer les performances.

Nous évaluons la notion de modèles locaux qui sont des modèles personnalisés construits dans des sous-ensembles de données connexes ayant des caractéristiques similaires. Pour l'évaluation, nous la comparons avec les modèles globaux qui sont un modèle complexe unique sans classification préalable. Pour ce faire, nous avons travaillé avec Airudi, une société de ressources humaines Française Canadienne qui nous a fourni des données réelles que nous utilisons pour construire notre cas d'étude où nous répondons aux questions de recherche suivantes : RQ1. Comment les modèles globaux se comparent-ils en performance aux modèles locaux? RQ2. Comment la précision et le rappel fonctionnent-ils sur différents seuils?

Pour notre cas d'étude, nous utilisons l'algorithme k-means pour faire le clustering. Pour trouver le nombre idéal k de clusters, nous avons utilisé la "elbow et silhouette methodology", pour notre distribution de données. Pour traiter nos données à l'algorithme k-means, nous transformons nos descriptions de poste en une matrice tf-idf. Pour chaque cluster nous avons généré un modèle en utilisant les caractéristiques du candidat et le score de similitude entre le poste et la description du candidat.

Les résultats suggèrent que les modèles locaux fonctionnent nettement mieux dans l'intersection précision-rappel lorsque nous remplaçons les clusters peu performants par le modèle global. Nous avons également constaté que la majorité des modèles fonctionnent mieux que le modèle global dans l'intersection précision-rappel.

Nos résultats indiquent également que 5 des 9 modèles locaux fonctionnent nettement mieux que le modèle global en termes de précision et de rappel à travers les différents seuils. Un cluster fonctionne sans aucune différence et trois clusters sous-performent par rapport au modèle global.

En tant que facteurs influençant la performance des modèles locaux, nous avons constaté que les descriptions de poste répétées jouent un rôle dans la performance d'un modèle local.

L'approche des modèles locaux pour fonctionner correctement dépend de la distribution correcte des données.

Nos résultats sont un bon point de départ qui montre les avantages des modèles locaux par rapport au modèle global. Dans le cadre de travaux futurs, nous visons à découvrir les meilleures techniques de clustering en fonction de la distribution des données, et quel algorithme de formation est plus approprié pour chaque cluster.

ABSTRACT

Selecting the best candidate for a job position is a challenging topic that has been gaining interest in research and practice. This has led to increasingly more complex models, algorithms and techniques exploiting natural language processing, semantic similarity, and machine learning. This thesis complements this work by taking a step back and focusing on how to better exploit available data in order to further improve model performance. In particular, we empirically evaluate the notion of using "local" models for subsets of the data having similar characteristics (job descriptions) as opposed to using a single, complex "Global Model." Using job candidate and description data, we found that local models perform better than the global models in terms of precision and recall, with median improvements up to 11.64%. If we substitute the under-performing models with the global model, thus creating a hybrid local model, the difference becomes significant.

Our results suggest that local models for job recommendation brings performance advantages in terms of precision and recall over a global model, motivating further research in local models for job recommendation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

TF-IDF	term frequency–inverse document frequency
e-recruitment	Electronic recruitment
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

CHAPTER 1 INTRODUCTION

Hiring skilled workers for specific positions in growth markets has become a challenge, better screening techniques to automatically rank or even select the best candidate for a given position [1]. The size of the e-recruiting market in 2020 in terms of revenue is $10.3 billion dollars, and is expected to grow by 8.6% in 2020. In the US the e-recruiting market has increased faster than the economy overall [2].

These trends have led to an increasing number of techniques that have been proposed to improve candidate and job matching. Candidate matching is the process of finding suitable candidates for a given job position based on the candidate's CVs. Models for candidate matching typically leverage information retrieval and information filtering or knowledge-based techniques making use of advanced AI models [1]. These techniques, especially the semantic ones, exploit the language context of the CVs and job descriptions, as well as any other available data that could be used for training AI Models [3].

Complementary to the existing work that explores advanced AI algorithms (deep learning) [4] for matching candidates to jobs, we instead go back to the data dimension, where we conjecture that a lot more can be done using the available job and resume data to improve the model performance in this domain. A given model might be too complicated for the data quantity we have, this is called high variance causing over-fitting in the model. Such high variance can be reduced by decreasing the number of features used or by adding more data.When the model we have is too simple for the data we have is called high bias [5].

In particular, instead of building advanced complex artificial intelligence models across all the data, we go back to the data dimension where we conjecture that building even simpler local models on subsets of related data to form individual models can improve the performance in the e-recruiting context. For example, in the software engineering domain in the context of defect prediction, the idea of local vs. global model by Bettenburg et al. [6] produces significantly better fits of statistical prediction models with a difference of 88.6957% in goodness of fit for the lucene data set which, is a defect log that contains the report of defects and errors of 340OO Java classes. Bettenburg et al. also discussed the advantage of using local models over global models with respect to prediction performance, where they found that the local approach significantly increases the predictive power of statistic models, with up to three times lower prediction error.

The idea also has been proven to work in domains with high-dimensional data, such as medicine, where a clustering-based local model approach had been proposed for detecting

cancer [7] and for lung nodule classification [8].

1.1 Research Objectives

In this thesis, we propose a local model approach that works by forming models on clusters of related job data, where we hypothesize that local models have performance improvements over a single global model. To do this, we use a clustering algorithm to form clusters based on related job descriptions, with the aim of training individual models, one per cluster, that are called "local models,' instead of a single "global model" for the entire dataset.

We then empirically evaluate local models in terms of precision and recall performance on a large data set from our industrial partner Airudi, a French Canadian Company that specializes in AI powered human resources. The data set is composed by anonymized French job descriptions and the selected candidates for each one.

We find that in terms of precision 2/3 of the clusters perform significantly better in terms of precision and recall in comparison to the global model, while 1/3 of the clusters perform significantly worse in terms of recall, only one cluster perform significantly worse. In terms of recall, only one cluster performed significantly worse. When we realize that the local models with worse performance were causing high variance, we substitute the local models of the clusters with worst performance by the global model, effectively outperforming the global models with a median percentage of 11.6% in the precision-recall intersection. These findings show that even relatively simple clustering techniques are already able to show significant improvements in candidate matching, hence more advanced clustering algorithms can only further improve the performance. Consequently, combining both more advanced algorithms and local models can improve the results of e-recruitment models.

1.2 Concepts and Definitions

This section provides the necessary concepts and definitions that relate to our work:

Candidate Matching

In this research, we will refer to candidate matching as the process of finding the most suitable candidates for a certain job position. This process typically starts by a company publishing a textual job description for interested candidates. To express their interest, prospective candidates have to send or (in e-recruitment) upload their resume to the company directly, or to a 3rd party HR company in charge of finding job candidates.

The Human resources department of the company, or the 3rd party HR company, will review the features of every candidate. The process has multiple phases: first they do a legal/practical based on factors like work permit, living close, background check to build a a candidate shortlist. To then with detailed interviews arrive at a final ranking. Some of the features that are reviewed by the human resources department when evaluating a candidate profile are: abilities, experience, qualifications, unemployed time, to mention some.

Doing this process manually has advantages and disadvantages. The advantages are that humans can easily parse and understand CVs in different formats, use prior experience in a domain to help their judgment, and can interpret body language during interviews. The disadvantage is that the manual recruitment process is time-intensive and could be biased by the physical/mental state of the interviewers (tired, bias towards gender/nationality of candidates, etc.). The premise of the domain of e-recruitment is that the core activities of the process (i.e., mostly the early phases) should be automated as much as possible, yielding a shortlist of candidates that humans could then interview/assess manually, basically leveraging the strong points of both humans and automation [9].

Global Model

A Global Model for is a model trained on the whole dataset regardless of any classification.

Local Models

Local models are individual models trained on subsets of the data.

Clustering

Clustering is when we group a set of objects where each set of grouped objects is called a cluster.

Semantic Similarity

A big challenge for the e-recruitment market is to make semantic interpretation/matching from textual data sources (job description and resumes). The metric used to analyze the similarities and relationships between two texts is called semantic similarity. It measures in a set of texts, documents and/or terminology where we consider the difference between each document and/or terminology as a distance based on the similarity in meaning/semantic content, of the documents, i.e, not just lexicographical similarity.

Mathematical techniques, like word2vec and term-based similarity, exist to measure in a numeric way the relationship between each term in a language, the relation of concepts, words, and instances [10].

Model - A model is a mathematical algorithm that replicates a decision process to automate

it and understand it. A model is trained on data based on a human expert's input to replicate the decisions an expert will do when provided with the same data. Ideally a model, should reveal the rationality behind it to help interpret the decision process.

Artificial Intelligence - Refers to a interdisciplinary wide-ranging of computer science techniques and algorithms with the objective of building smart machines capable of performing tasks that normally will require human intelligence.

Human Resources - is the conjunction of the human workforce that makes an organization, sector, economy or industry, sometimes also called human capital

Screening - in the concept of human resources is the process of filtering the candidates that are not suitable for the job and selecting the possible candidates.

Parsing - Process of analysis the syntax's of the resume data

Ontology - An ontology is the naming, definition of the properties, types and relationships of entities that exist on a determine domain. One of the objectives is to represent knowledge, sharing knowledge and reusing the knowledge.

e-recruiting - is a series of techniques and tools that in an automatic form accomplish the different tasks in the selection process of a candidate.

1.3 Thesis Plan

This thesis is organized in the following way:

- In Chapter 2, we review the literature and previous work that has been done in the e-recruitment domain.
- In Chapter 3, we explain the current usage of the models in the context of e-recruitment and give a conceptual explanation and comparison of the local models approach.
- In Chapter 4 we present the research questions for our case study and their motivations, we also present the description of the prototype we built for the case study, as well as the design of the study.
- In Chapter 5 we present the evaluation of the results, its discussion, and the threats to validity.
- In Chapter 6 we present the thesis conclusions and the future direction of our work.

CHAPTER 2 LITERATURE REVIEW

People analytics is "the application of math, statistics and modeling to worker-related data to see and predict patterns" [11] for the management of a company's hiring process [12]. At a minimum, modern Applicant Tracking Systems (ATS) incorporate resume parsers that use AI techniques to parse all kinds of CV formats into a structured database, reportedly achieving accuracy's close to 95% [13]. Parsing CVs is only the tip of the people analytics iceberg, since automatically matching or recommending the right candidate for a given job, or vice versa, is the real goal. This is because, in today's professional climate, from multinationals to SMEs, there is a high demand for technically specialized jobs, for which most of the candidates live abroad.

The works done on "e-recruiting" around in 2012 were mainly focused on simple boolean search methods. Al-Otaibi et al. [1] evaluated the state-of-the-art in job recommendation systems up until 2012, dividing existing techniques into: (1) hybrid job recommendation systems combining several techniques in order to overcome the problems of the individual ones, and (2) content-based job recommendation systems using content filtering techniques. The authors found a number of disadvantages with the state-of-the-art systems: obtaining good performance is not straightforward, normally there is a lack of historical data about previous job recommendations, making it hard to find similar candidates and job descriptions in the past.

Furthermore, the surveyed systems [1] only provide a one-way recommendation (either candidate recommendation or job recommendation, not both ways), and often consider only some of the fields of CVs and job descriptions, in order to simplify the job recommendation models. Important fields like candidate preferences and choices, as well as the tools and skills that the candidate knows, are excluded, and links between different skills and job transitions are rarely considered, since they are difficult to extract. Finally, techniques based on collaborative filtering and AI-based models have been found to have scalability problems in real-life scenarios, since many of them focus on more complex models or require complex historical data for building more complex models.

We classify the previous approaches that improved the screening process into two categories that are discussed next:

2.1 Ontology-based matching approaches

Kumaran et al. [14] used ontology mapping technologies to recommend the most likely candidates for a given job description. The authors automatically learned an ontology extracting the qualification, CGPA, Total Experience and Skills sections of CVs in the IT subdomain, identifying co-occurring or closely related concepts across the CVs, using first a concept extractor and then a concept linking approach, they collect all the relevant information. They then construct a job description ontology containing the requirements of a job position extracting the qualifications, giving a weight for each requirement depending on importance. Finally they map the job description and candidate ontology using the instant mapping approach which calculate similarity between the properties, using both ontology's, they calculate matching scores for each pair of candidates and job descriptions, returning all pairs where the matching score is higher than a given threshold. The authors achieved up to 90% accuracy using this methodology. The main issue with this approach is that it does not consider the specific job descriptions a candidate is interested in and that building an ontology for each area and subarea does not scale.

Guo et al. [15] address the e-recruiting problem using more detailed ontologies. They take into account subdomain particularities like the required degree, programming languages and skills for a certain position, using semantic labeling to prioritize criteria. Semantic Labeling works by trying to unify related tokens to the same concepts (labels) like "candidate's degree" or "candidate's experience", taking care of both hyponyms (a word of more specific meaning than a general or superordinate term applicable to it) and hypernyms (a word whose meaning includes a group of other words). Finally, they used a similarity index that computes the weighted sum of the computed features to give a score to the matches between a CV and a job description. However, scaling up this method will require a lot of specialized people of human resources for each domain and subdomain in order to build the ontology relationship for all domains.

2.2 AI and Machine learning matching approaches

Lee et al. [16] proposed a matching method based on the skills the student candidates have and the skills the company is looking for matching graduated students to companies. Lee et al. developed the method they named Artificial Intelligence-based design platform (AID) that is based on finding patterns between the company required skills and the skill the candidate has. The approach uses MILP which is a linear program modification where some variables are constrained to take only integer values where constraints on the variables enable the

inclusion of discrete decisions in the optimization, MILP Was used to assign the right student to the right company. The student skills were represented on a numeric way depending the area, if a student has a skill for a certain area was marked with a boolean number. In the case of the companies, the preference on a skill on certain area was marked on the same way. With this approach, Lee et al. accomplish zero miss-matching for student skills and company needs, compared to other statistical methods that produce 30% (perfect matching) of miss-matching. This approach was developed only with a limited number of companies and areas, which makes it difficult to scale, since a correct extraction and interpretation of the skills is necessary in each area and subarea. The approach of Lee et al. only recommends jobs to students or students to companies based on the preferences of both.

Xu and Barbosa [17] did a stacked model approach where they stack the prediction of three different models (gradient trees, random forest, and neural networks) to do the prediction. The authors compared their approach with three different models: gradient trees, random forest and neural networks. Xu et al. found that their stacked model has a higher accuracy than the other methods outperforming them in all the models. This approach depends on using multiple AI models and uses all the results of all the models, which requires the use of multiple models to have a result.

Purohit et al. [18] support recruiters by generating interview questions for them. They do this by analyzing the candidates' CV and classifying the answers of candidates using keyword matching. Then, the accuracy of an interview is improved based on the previous responses of the candidates using a Natural Language Processing (NLP) model. However, this work is still a high-level proposal without details about how NLP technologies, prototype or experiments were done.

Recently, AI has been adopted to reduce the degree of human intervention during the actual interview, since substantial time is wasted there (e.g., out of politeness even very bad interviews are seldom cut short). The most spectacular application of this is Stafory's Robot Vera (https://ai.robotvera.com), used by, amongst others, SAP (Rohr et al., 2019) [19] , Ikea, PepsiCo and Loréal. Robot Vera is an AI with a customizable graphical avatar that also performs the interviews herself using 8-minute video calls, up to 1 hour after discovering a candidate's CV online. Vera summarizes her findings for human recruiters, who are still in charge of ranking the candidates and making a final decision, giving a shortlist of the recommended candidates.

The Vera avatar supports around 70 languages and has been trained on 100,000 job descriptions and candidate questions about the job descriptions, the authors used Wikipedia articles, TV series, even subtitles from Game of Thrones to teach Vera how to speak naturally, using

NLP techniques matches questions to answers appropriately [19]. Early reports [20] mention a 33% reduction in the time taken by the recruitment process using Robot Vera, performing fifty thousand interviews per day (across all customers of Stafory). The avatar is most successful for jobs for which (technical) skills can easily be checked, but is less appropriate for more executive positions, where human judgment and soft skills matter more. This work is a private development that cannot be compared or evaluated with other approaches, but shows how AI is applied to go through the different stages of the recruiting process.

The works done in candidate matching have been focused mainly on information filtering, complex ontologies or more complex AI models, which make them difficult to scale. Given the impact of job recommendation platforms on the lives of humans, it seems essential that they should be evaluated on real data. In the following section proposed our own approach, by focusing on the data by working on subsets of related data called local models for job recommendation.

CHAPTER 3 BACKGROUND

This thesis does not explore better classification algorithms for job recommendations, nor better ways to represent the data of a given model. Instead, we try to find better ways to exploit the existing data such that even using more basic learning algorithms, one can achieve better candidate matches. We first discuss the current usage of models before explaining the concept of "local" models at the heart of this thesis approach.

3.1 Global Models

A Global Model for job recommendation is a trained model based on the entirety of prior job data regardless of the kind or domain of job. For example, a single (global) model is trained on all job descriptions of the year 2012 with its matching candidates regardless of the domain. The following steps are used to build a global model:

- 1) Given a set of prior job descriptions with the selected candidates, we perform the extraction of the candidate and job description features.
- 2) We use the extracted features to train a single model able to classify suitable candidates for a given job. Typically, most classification algorithms like random forest provide a classification probability as output, which is compared to a threshold to filter out unsuitable candidates, then used to rank the surviving (suitable) candidates from high to low probability.
- 3) Given a Job Description with potential candidates, we will use the newly built Global Model to rank the candidates from the best to the least suitable candidate.

In terms of advantages, the global models are conceptually easy to build (see Figure Figure 3.1). Furthermore, its typical conceptual design is easier to understand and fit into the data engineering pipeline because it is easier to build, test and deploy a single model architecture, than a multi-model architecture. [21].

However, the global models have some disadvantages: First of all, Global Models are biased against most common types of jobs. Domains with more jobs dominate the data, therefore are more likely to have a better performance than the ones with fewer jobs or jobs with deviating descriptions.) [6]; Second they are susceptible to high variance, since a model may be to complex for the data it has. This can lead to model overfitting. [5]

3.2 Local Models - Naive Version

The Naive Local Models Approach uses a pre-existing or manual classification of job descriptions. For example “retail” and “software engineering”, or even finer-grained sub-categories like “developer” and “tester”. Instead of just having one single model across all the job descriptions, a model is generated for each category in the data.

The process for building the naive version of local models is shown in the Figure Figure 3.2.

- 1) Given a set of prior job descriptions with their selected candidates, and with the job descriptions pre-classified in existing categories, we extract the features of the candidates and job descriptions.
- 2) We use the extracted features to train a model for each existing category in the data, generating k models where k is the number of categories that exist.
- 3) Given a job description with potential candidates, we will determine the most appropriate category the job description belongs to, then use the corresponding model to get the ranking of the candidates from the best to the least suitable.

Using a model trained only on one category may improve the performance in terms of precision and recall when choosing a job of the same category, since the model can be more specialized and differ along with each category. The training time may be faster since models are trained on only one category.

As disadvantages of the Naive local models, the naive local models depend on a stable data distribution to fit each category evenly, since some models may not have enough data for a good performance. This unbalance could produce high data variance causing over-fitting in a model and a bad performance in that model.

New job descriptions of currently unknown categories may not map to any existing category and might require new models to be built (whereas a global model could be applied immediately to such job descriptions).

The data we are working with may not be categorized, so if we do not have an existing category the naive approach is not viable. From our experience interacting with our industry partner Airudi, this is a major problem in actual job description data.

Depends on a good data distribution to fit each category evenly, since some models may have not enough items for a good performance. This unbalance could produce a high variance

causing over-fitting in the model and a bad performance in that cluster. New job descriptions of currently unknown categories may not be correctly represented on an existing category, and might require new models to be built (whereas a global model could be applied immediately to such descriptions). The data we are working with may not be categorized, so if we don't have an existing category the naive approach won't be viable. From our experience interacting with our industry partner Airudi, this is a major problem in actual job description data.

3.3 Local Models - Smart Version

The smart version of the local models overcomes the naive model disadvantages, which in practice makes the approach more pragmatic and usable in practice. First, the data doesn't require any prior classification. Second, the number of categories is determined in accordance to our data distribution. The smart version works by discovering the underlying classification existing on our data, creating a model for each discovered category.

Such local models have been in use with great effect in other domains. In the Biology domain Aydadenta and al. used the k-means algorithm to categorize data, then the relief algorithm is used to select the best scoring element for each cluster. Aydenta et al. found that the accuracy was higher for Colon, Lung and Prostate Cancer with the clustered approach compared to the approach without clustering. [7].

Lee et al. [8] made an automated lung nodule detection system that can help to detect abnormalities in lung images, Lee et al. used the CAC method to build local models, achieving the best sensitivity of 98.33% (ROC A(z) of 0.9786). For training their models the authors used random forest.

Since the "smart" local model approach does not assume jobs to be categorized according to a domain, a data clustering algorithm is used to discover the natural grouping of the data based on a set of patterns, points or objects. There are two types of clustering algorithms.

- Hierarchical - Hierarchical algorithms find recursively nested clusters, they start with each one of the data points in its own cluster and then, they try to merge the similar pairs, forming a hierarchy.
- Partitional - Partitional clusters find all clusters at the same time as a part of the data without imposing a hierarchy or structure.

The necessary input for a partitional clustering algorithm is a (nxn) similarity matrix, where n is the number of data points to cluster. The matrix elements have to contain the pairwise

similarity between all points. The similarity matrix can be derived from a pattern matrix that represents the relation between the words of the job descriptions [22]. Often multidimensional reduction is necessary, which works "by projecting the data to a lower dimensional subspace which captures the “essence” of the data" [23].

Local Models reduce the bias against a specific category dominating in the data and they do not require previous job descriptions classification. The categories are built depending on the data distribution, generating clusters of job descriptions which are trained to generate models with better reliability and accuracy.

Even though local models can bring some improvements, it still relies on the data distribution for a good performance. It is not practical to built local models in small sets of data since they would not have enough elements for a good train. If all data is on the same category or the data is to similar, the results will be biased to that category.

3.3.1 Local Models Building Process

As Figure Figure 3.3 shows, the process for building the Local models begins similar to the process of building a global model, with a dataset with the same requirements as for the Global Model. As step 1, we have to determine the ideal number of clusters to be built. A cluster will be a subset of the original dataset of job descriptions which contains related job descriptions. Charrad et al. [24] evaluated different methodologies for determining the ideal number of clusters. In step 2) based on the results of the ideal k number of clusters, we generate k clusters using a clustering algorithm. In step 3) we extract the candidate and job description features from each cluster. As step 4) using the features to train a model for each cluster, k models are generated. In step 5) given a Job Description with potential candidates, first we have to determine the cluster where the Job Description belongs to and use that model to get the ranking of the candidates in descending order from the model that was determined.

The smart version of the local models is more suitable in most cases than the naive version because it doesn’t require any prior job classification, and the classification is done in a way where it produces models with better reliability and accuracy.

In this chapter we saw the theoretical advantages that local models can bring over global models and we propose a smart version that does not depends on having previously classified data.

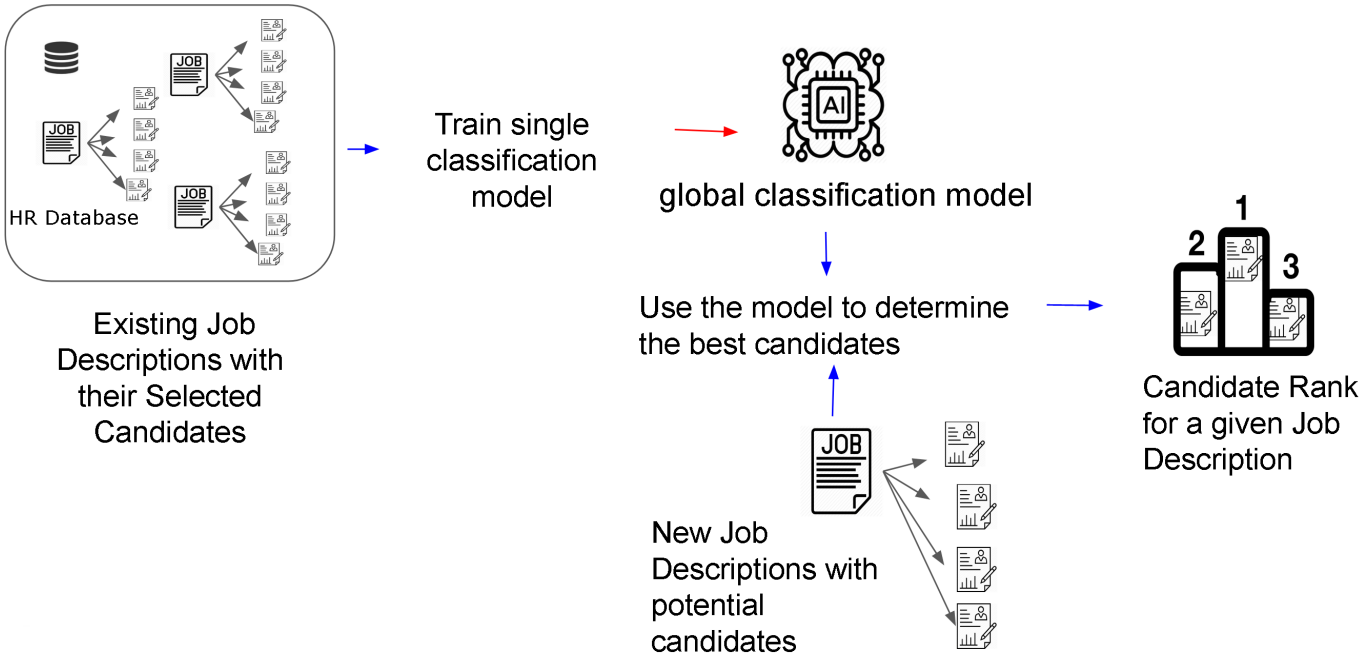

Figure Figure 3.1 Global Model process

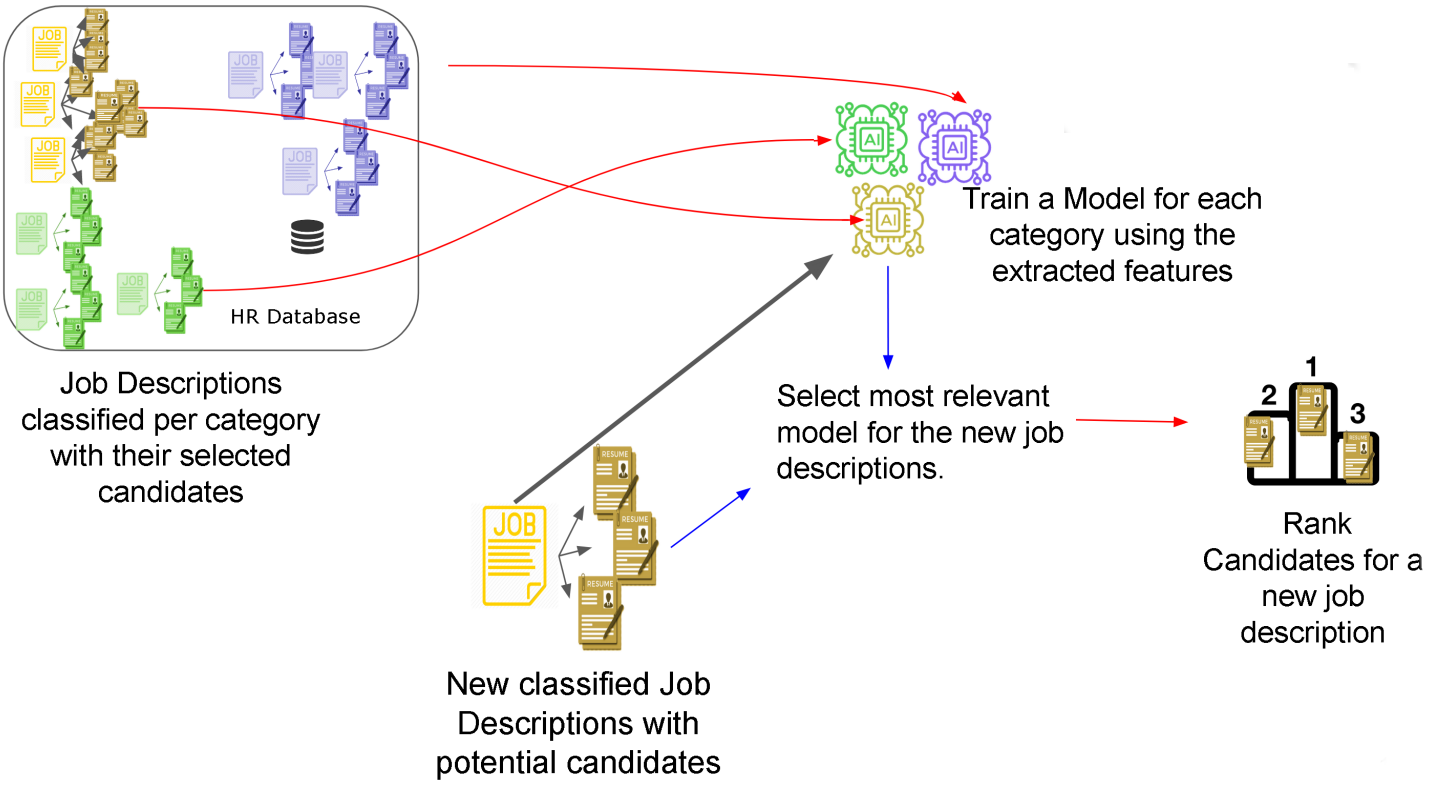

Figure Figure 3.2 Local Models process - Naive Version

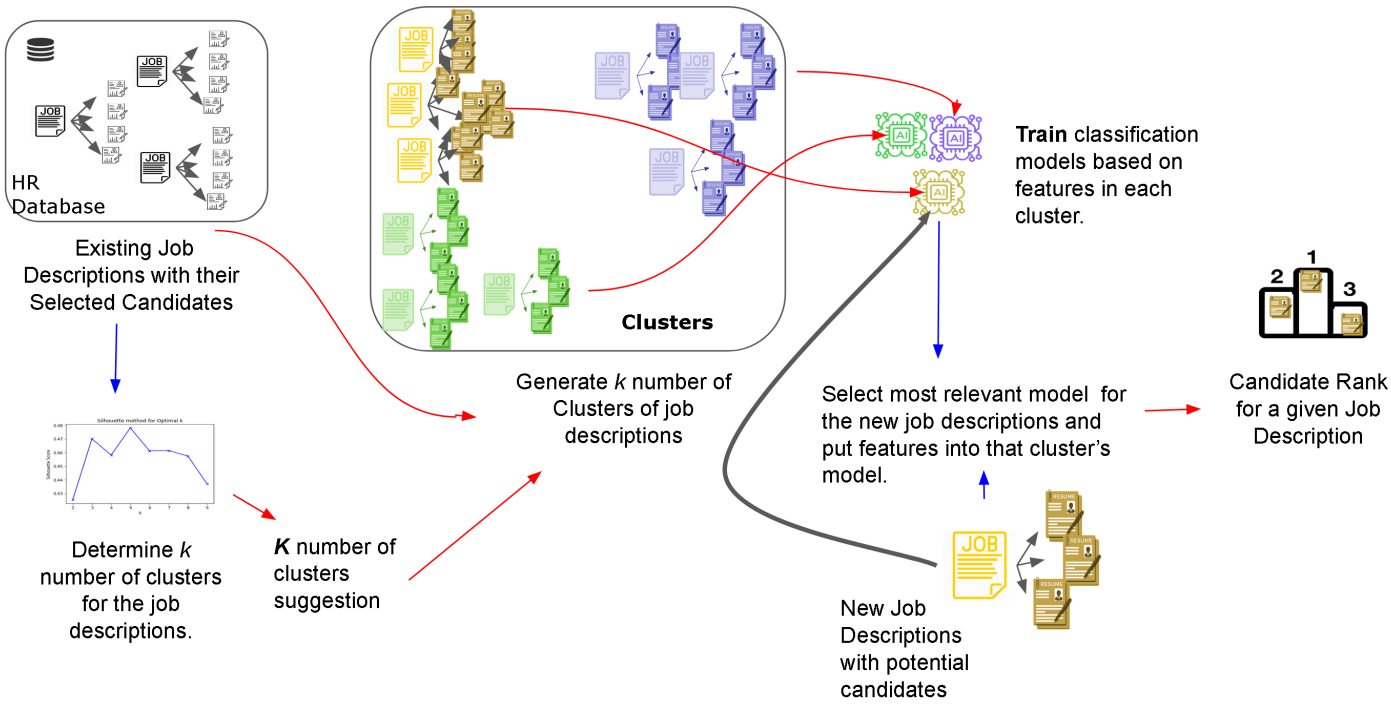

Figure Figure 3.3 Local Models process

CHAPTER 4 CASE STUDY

This section presents the design of our empirical validation of local models for job recommendation that was done with data from our industrial partner Airudi. We will begin by presenting the research questions for our study, the motivations behind the research questions and the approach used to address them. Then, we present the dataset with its particularities and at the end of this section we present the performance metrics that are used to evaluate our research questions.

4.1 Prototype Approach

In this section, we present the prototype implementation of the global and local model approaches discussed in the previous chapter.

4.1.1 Research Questions

We now present the research questions for our studies and their motivations:

RQ1. How do global models compare in performance to local models?

Motivation.

Since Local Models have had promising results in areas like cancer prediction [7] where we have sets of noisy data, and many times the data comes without any previous classification, we want to evaluate it in the "e-recruiting" context. In particular, using real data from our industrial partner Airudi, we intend to evaluate the concept of global models compared to the local models concept in terms of classification performance. For a given job description with a shortlist of candidates, that meet the required and desired criteria, we will rank the candidates in descending order from the most to the least suitable. This is the input for the human interviews which is the last phase of recruitment process explained in Chapter I.

Approach.

To evaluate the concept of local model against the global model in the context of e-recruiting, we compared it in terms of precision, that is how many selected items are relevant and recall, that is the representation of how many relevant items are selected. Since the interpretation of a classification model that returns a probability depends on a threshold to decide which

candidates should be put on the shortlist for a given job, and which ones not, RQ1 evaluates the approach for the threshold value when $precision = recall$. This is because both the absence of false positives (precision) and the ability to find all relevant candidates (recall) matter equally for job recommendations. For determining if the precision-recall distribution is normal we used the Shapiro test, in case is normal, we determine its significance with the T-test, if data distribution its not normal we used the Mann-Whitney test with Bonferroni correlation in order to determine the significance of our results.

RQ2. How do the precision and recall perform across different thresholds?

Motivation.

Since precision and recall values change across the threshold used to interpret the classification model's probability output, we compare the precision and recall distribution values across a threshold from 0 to 1 to evaluate the models indepently of the threshold used.

The second aim of this question is to see the impact of the unbalanced nature of the composition of each cluster and the data that is composing it and why we may have performance differences not only between "local models" and "global models", but also across the clusters.

Approach.

To visualize the performance differences in terms of precision and recall across the thresholds, we are going to plot the performance distribution across the different thresholds in every individual cluster and in the global model. After this, we are going to compare every individual model against the Global Model to see which clusters perform better, similar or worse.

4.1.2 Dataset and Feature Extraction

Our industrial partner Airudi Inc. Provide us classified and anonymized data in the French language, mainly for positions in Canada and France. A json file with job descriptions was supplied by the same enterprise labeled with the candidates shortlist.

In particular, the candidates could have a different status in relation to the job they are matched to: we categorized the candidates state into "accepted", which represents the candidates that got that job, and into "not accepted", which are the candidates that were not in the shortlist for that job. The status were represented into these categories because we are looking to rank the shortlist candidates. This status was used as labels (what we want to predict) in each one of the models we generated.

After dropping data with null values, we were left with 3023 unique job descriptions. The job descriptions consist mainly of two features: Job Title and Job Description. Each job description is matched with one or more selected or considered candidates for a job. In total, all job descriptions are matched with 7430 “accepted” candidates and an additional candidate pool of 45296 unsuitable candidates provided to us by Airudi.

In Table Table 4.1 we can see the features we extracted of the candidates and job descriptions. We used these features to train our global and local models. The features we are using are based on Kumaran et al [14]. The dates were parsed from the data while the years of experience and years without working were calculated from the dates of debut and years without job.

The dataset of 3023 unique job descriptions and 52726 unique candidates was used to build a global model and to build local models based on clusters. For each job description cluster a model was built that is trained only on that cluster job description features and the features of the candidates matched to that job description.

The candidate and job descriptions were used to calculate a semantic similarity score, using the python library Spacy [25] which works by comparing word vectors or “word embedding”. Word embeddings uses a neural network model to learn word associations, which are represented in a multi-dimensional space. When comparing the job and candidate description a score is given where the higher it is, the more similar a job description to a candidate description is. Job and candidate descriptions were tokenized, and the stopwords were removed. As word embedding model we used the "default" french dictionary that the Spacy package provides which is trained on Wikipedia [26].

The candidate ranking in descending order from the best to the worst candidate for a given job description is the output of the models (global and local) trained using the described features.

Table Table 4.1 Extracted Features

Candidate	**Job Description**	**Common Features**
Year of Debut - Integer	Job Title - String	Similarity score - Integer
Month of Debut - Integer	Job Description - String	
Last Year of Work - Integer		
Years of Experience - Integer		
Years without working - Integer		
Candidate description - String		

4.1.3 Determining K number of clusters with K-means

As clustering algorithm, we chose k-means since it has been effective already in previous works with different contexts [7].

K-means is a partitional clustering algorithm that finds partitions such that the squared error between the empirical means of the cluster is minimized and the most common similarity metric used for k-means is the Euclidean metric, where k-means finds a spherical cluster shaped data. [22]

(Jain and Dubes, 1988) defined the steps to use the k-means algorithm:

- Define a k number of partitions
- Generate a partition by assigning a pattern to the closest distance center
- compute the centers of the new clusters
- repeat the steps until each cluster membership stabilize

The parameters necessary for using the k-means algorithm are:

- k number, that is the number of clusters
- cluster initialization
- distance metric

The algorithm for calculate k-means can be defined in the following way:

Algorithm 1 K-means algorithm

Require: k and dataset
  $centroids = getRandomCentroids(numberofFeatures, k)$
  while NOT stop(count, centroid, pastcentroid) **do**
    $pastcentroid = centroid$
    $count++$
    $labels = getlabels(dataset, centroid_list)$
    $centroids = calculateCentroids(labels, dataset, k)$
  end while
  return centroids

Where the function *stop* will return TRUE or FALSE if the k-means algorithm finishes, the function *getlabels* will return a label that represents a cluster for each element on the dataset.

The function *calcuclateCentroids* will return k random centroids each one of the dimensions of the label where each centroid is the geometric mean of the points that have the centroid label [27].

The K-means algorithm was used to build clusters of job descriptions to generate local models. These clusters were trained on the candidate and job descriptions of each generated cluster of job descriptions. In order to calculate centroids, k-means requires a similarity measure between each data element. For this, we created a tf-idf matrix between all job descriptions. Term Frequency Inverse Document Frequency (TF-IDF) is a measure that determines the relative frequency of the words in a document or text compared proportionally inverse of the word on the entire corpus of documents.

What TF-IDF determines is, how relevant a word is in a particular document. The tf-idf approach works given a document collection D, taking $d \epsilon D$ as a single document and word as w, this can be calculated like:

$$wd = f_{w,d} * log(|D|/(fw, d)) \tag{4.1}$$

$f_{w,d}$ equals the number of times w appears in $d, |D|$ is the size of the corpus and $f_{w,D}$ will equal the number of documents in which w appears in D. This is done running a sum of the frequency of each word in the collection of documents and per document.

A process commonly associated when using tf-idf algorithm is the removal of stopwords, that are words that don't provide any meaning to the text like prepositions, pronouns and articles. Tokenization is the task of chopping into simpler forms called tokens [28]. As part of the tf-idf implementation, the job descriptions were tokenized and stemmed using the nltk snowballstemmer [29] using the French dictionary, to have simpler versions of the words.

Charrad et al. [24] evaluated methodologies for determining the ideal number of clusters, using the silhouette and Elbow [30] methodologies. The Figure Figure 4.1 shows the visual implementation of the elbow method. The elbow score is the sum of the squared errors on all points and can be used with Euclidean and Manhattan distance. The elbow can be visually seen when we plot the sum of square error across the number of clusters, and can be computed using the Equation:

$$W_k = \sum_{r=1}^{k} (1/n_r)(D_r) \tag{4.2}$$

In the formula, k represents the number of clusters, n_r is the number of points is in the cluster, and r is the sum of the distances between all points which is represented by Dr:

$$D_r = \sum_{i=1}^{(n_r)-1} \sum_{j=1}^{n_r} ||d_i - d_j||_2 \tag{4.3}$$

Where d_i and d_j are the intra-cluster distances between points in a cluster C_k which contains n_r points. [31]

Sometimes the result can be ambiguous when plotting since more than one elbow might be observed. In that case, the silhouette method can be used for determining the ideal number of clusters. The silhouette as shown in the graphical representation in Figure Figure 4.2, measures how similar a point is to its own cluster in comparison to the other clusters, the value is between +1 and -1. for each k comparison a score is given, where the score will reach its maximum when k is optimal. The value for the Silhouette for a certain data point is defined in the following way:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1 \tag{4.4}$$

Here $s(i)$ will be zero if it is the only point in the cluster. $a(i)$ is the similarity of a point i to its own cluster, and $b(i)$ is the dissimilarity of a i point to other clusters.

Based on Charrad et al. [24] work, to know the ideal number of clusters to do the k-means clustering, we used the elbow method methodology where we had a coincidence with both methodologies for nine clusters in accordance to the silhouette and elbow methodologies as we can see in the Figure Figure 4.1 and in Figure Figure 4.2.

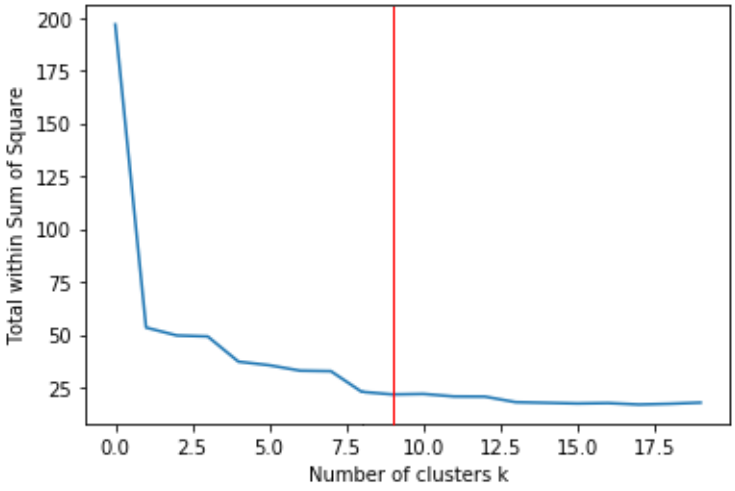

Figure Figure 4.1 Elbow Method graph

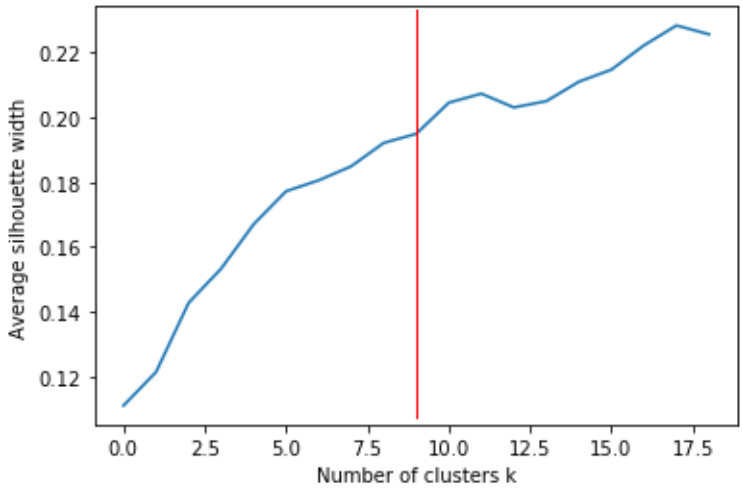

Figure Figure 4.2 Silhouette Method graph

Hence, we fit the tf-idf matrix of job descriptions as input into the k-means algorithm to build nine clusters

4.1.4 Train Classification Models with Random Forest

In this section we present the algorithm we use to train our model using the features and labels described in chapter 4.1.2 and the cross-validation technique we used.

4.1.5 Random Forest

We worked with an ensemble learning method [32] to not just relying on one Decision Tree hoping we made the right decision, which uses multiple learning algorithms for getting better predictive performance in the predictions than the one that can be obtained by each individual learning algorithm [7].

As ensemble algorithm we used random forest, which works by using several decision trees. Each tree votes for the most popular class in the classification trees, given a certain input.

For each model we trained a random forest model on the extracted features of the candidates and job descriptions of a given job description cluster, using the features explained in chapter 4 . using as labels if the candidates were selected or not for the job position. Once built, the output of a random forest model for the feature values of a new pair of job recommendation and resume is the probability that a candidate will be chosen for that job position.

4.1.6 Cross-Validation

In order to validate that the results we got for our classification models are independent of the specific data set used for our evaluation, we used k-fold validation using 10 folds. K-fold divides a dataset in k folds where the original data is partitioned into k sub-samples of the original data [33]. We used k-fold with 10 folds to check stability of performance across subsets of the data, no bias towards one specific data set. When the difference gets smaller [34].

We used k-fold cross-validation dividing the samples in k group of samples, called folds of equal sizes when possible (stratified). Each fold is then used once as a validation set using $k - 1$ folds as training set. Table Table 4.2 has the size of each of our generated clusters where we use 10-fold cross-validation.

This type of k-fold validation is done when a single fold is used with the validation and the test set which is split in k sets. For each set one is selected as a test set and the other part as validation set and the rest as training sets where all possible combinations are evaluated [35].

With 10-fold validation we end up with 9 times 10 local models (10 per cluster), and ten times a global model which we will use to compare the performance to.

4.2 Performance Metrics

To compare the performance of the local models classification against a global model classification we evaluate them in terms of precision and recall. Precision is how many selected items were relevant, defined as: Precision $= \frac{TP}{TP+FP}$ where TP = True positive, TN = True negative, $FP+$ False positive and FN = False Negative. Recall is how many relevant items

Cluster	Cluster Size
cluster 0	110464
cluster 1	17408
cluster 2	59160
cluster 3	42094
cluster 4	28908
cluster 5	18798
cluster 6	103850
cluster 7	8742
cluster 8	123872
Global	513296

Table Table 4.2 Size of each job description cluster in terms of job descriptions

are selected and is defined as Recall = $\frac{TP}{TP+FN}$.

To evaluate the overall performance of the models classification, we calculate the precision-recall intersection, i.e., when *precision* = *recall* across the threshold from 0 to 1, since it provides a single metric that assumes that both precision and recall are equally important.

Individual precision and recall is evaluated across a threshold that goes from 0 to 1 to be able to see the impact on each model's performance. This threshold is due to the probability that random forest returns: to turn that probability in a classification, a threshold is needed such that probabilities higher than the threshold correspond to a suitable candidate, while lower probabilities do not.

The higher the threshold, the lower the number of candidates deemed suitable for a job, hence the lower the recall of the corresponding model, while precision is expected to be higher (since only the candidates with the highest likelihood of being suitable, remain).

4.3 Null Hypothesis

To answer our research questions, we formulate the following null hypotheses:

Hypothesis $\mathbf{H_0}$ (Test hypothesis)**:** There are no performance difference between Local Models and Global Models in the precision-recall intersection

Hypothesis $\mathbf{H_1}$: There are no precision differences between the local models and global models across the different thresholds

Hypothesis $\mathbf{H_2}$: There are no recall differences between the local models and global models across the different thresholds

4.4 Statistical Analysis

In order to determine the statistical significance of our classification results, first we have to determine if our data has a normal distribution or not. For this we used the Shapiro test, which is defined as:

$$W = \frac{\left(\sum_{i=1}^{n} a_i y_i\right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2} \tag{4.5}$$

Where y_i is the ith order statistic and $\bar{y}$ is the mean of the sample. In case $p - value < 0.05$ the hypothesis that the data is normally distributed will be rejected. [36]

In the case of the comparison of the global and local models the precision and recall values

were normally distributed where we used the T-test which has a null hypothesis that the differences between the true mean of the global model and the global model is not significant. To measure the effect size we used Cohen's d, which works by calculating the standardized mean difference between two groups, all of these divided by the standard deviation [37]

For the comparison of each local model against the global model since the distribution is not normal so we used the Mann-Whitney test which is a non-parametric methodology with as null hypothesis that the mean ranks of the local and global model are the same [38]. To compare the effect size we used Cliff's Delta statistic which is a non-parametric effect size to measure the difference between two groups, afar p-value interpretation [39].

In order to correct the p-values because multiple comparisons are being done, we used the Bonferroni correction statistical test to avoid spurious positives since when we conduct a multiple analyses are made on the same dependent variable the change of having a false positive, rejecting the null hypothesis incorrectly. It works by lowering the alpha-value depending on the number of comparisons being performed, dividing the original α-value by the number of analyses on the dependent variable [40].

CHAPTER 5 EVALUATION

In this chapter, we present the results of the experiments we performed to answer the RQs presented in section 4.

5.1 Case Study Experiment Results

In this section we will give a response to the first research question:

5.1.1 RQ1. How do global models compare in performance to local models?

In Figure Figure 5.1 we can observe that the local Models have a significantly better classification performance than the global model. Precision-recall intersection is 2.179% higher than the lowest precision-recall intersection.

The median of the precision-recall intersection distribution (see Figure Figure 5.1) for the global model classification, calculated the median across the 10 folds, is 0.173, with a q1 (first quartile) of 0.044, q3 (third quartile) of 0.180 and a F1 score of 0.0685. For the global model, in contrast, for the local models, the median across 9 clusters of the precision-recall intersection distribution is 0.200 between a q1 of 0.149 and a q3 of 0.223, having a F1 score of 0.0941.

To determine if the difference between the local and global models is significant, we use a t-test, (since the data is normally distributed according to the Shapiro test) with a p-value of 0.89 at a 0.05 significance level, not being able to reject the null hypothesis that the difference is significant. To measure the effect size we used Cohen's d test, where we got a effect size of 0.6, being a medium relative size, representing a 69% of the global model below the mean of the local models approach.

As Table Table 5.1 shows, we have six clusters performing better than the Global Model and only three performing worse in the precision-recall intersection (red color). The measure represents when precision=recall across the threshold from 0 to 1.

After closer analysis, we observed that the few clusters performing worse are responsible for the lack of statistical improvement. A possible way to counter this would be to create a hybrid local model approach that replaces the worst local models by the global model and keeps the local models for the better performing clusters.

Figure Figure 5.1 shows that for these hybrid local models, the median of the precision-recall

distribution is at 0.200 with q1 at 0.17 and the q3 at 0.22, having a F1 score of 0.0941. This represents non-hybrid difference in the median of 11.6% In contrast, for the local models, the same median is obtained, but with much smaller variance: q3 at 0.30 and q1 at 0.22. Median precision is 2.179% than the lowest precision. The t-test now is able to reject the null hypothesis. The hybrid local models had an effect size of 1.325 representing a large relative size where 92% of the global model are below the mean of the local models approach.

In the models where we substitute the under-performing clusters (cluster2, cluster 6 and cluster 8) with the global model, using the t-test we accept the alternative hypothesis that the difference between the global and hybrid local models approach in the precision-recall intersection has a significant difference.

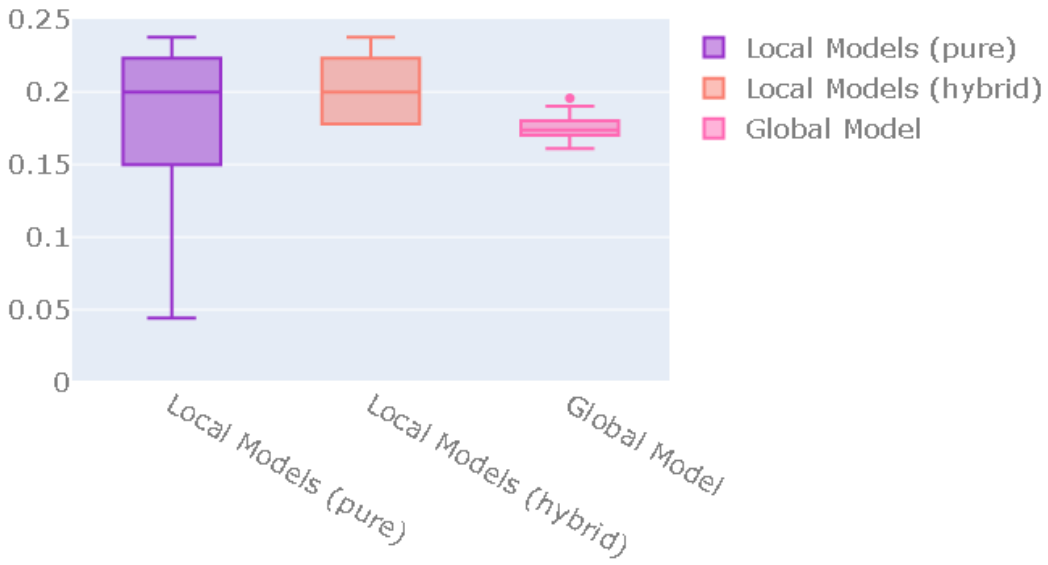

Figure Figure 5.1 Precision-Recall Intersection distribution (RQ1)

5.1.2 RQ2. How Does the precision and recall perform across different thresholds?

Across the studied threshold values, in terms of Precision and recall 5 clusters perform significantly better, while only 3 perform significantly worse

Figure Figure 5.4 shows the performance metrics in terms of precision of every individual cluster being compared to the global model, evaluated across the threshold from 0 to 1. The median precision of the Global model across the thresholds is at 0.040. To determine if the differences between each cluster and the global model are significant, we used the Shapiro test. This test rejected the hypothesis that the distribution is normal. Mann-Whitney with

Table Table 5.1 Clusters (Pure) precision-recall intersection values (high to low). In green the clusters performing better than the global model, in red the ones performing worst (RQ1)

Cluster	**Precision-Recall intersection**
Cluster 7	0.237
Cluster 4	0.230
Cluster 1	0.220
Cluster 3	0.200
Cluster 5	0.200
Cluster 0	0.180
Global Model	0.173
cluster 2	0.150
cluster 6	0.147
cluster 8	0.044

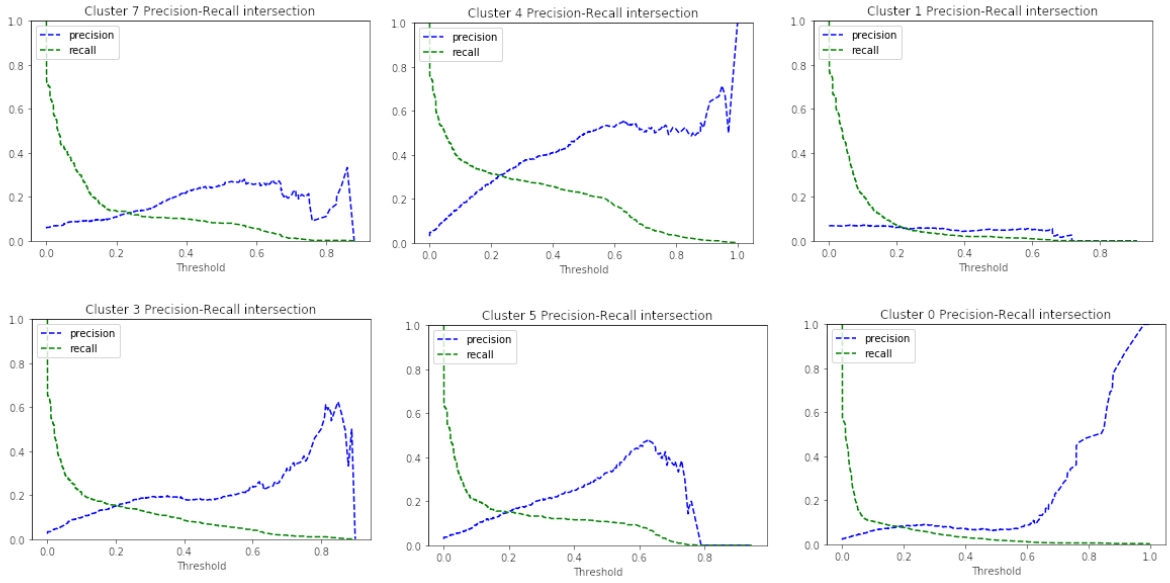

Figure Figure 5.2 Distribution of precision and recall of clusters that perform better than the global model (RQ2)

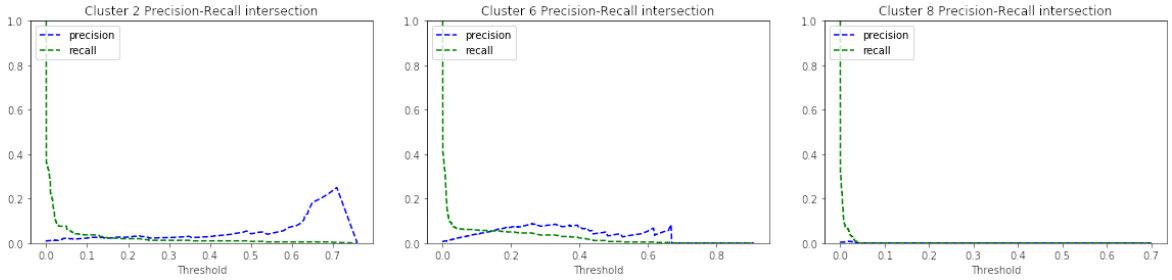

Figure Figure 5.3 Distribution of precision and recall of clusters that perform worse than the global Model (RQ2)

Bonferroni correction test was used to determine if the difference between the global model and each cluster are significant.

To counteract the problem of multiple comparisons and to avoid spurious positives, we corrected the p-values of each cluster using the Bonferroni Correction. The Bonferroni Correction works by dividing the alpha of 0.05 by 9, since we are working with 9 clusters.

In the Table Table 5.2 we can see the results of the median of each cluster compared with the global model. The values that are in green have a statistically significant improvement, while the values that are in white did not have any statistical difference in comparison to the global model. The values that are in red represent the clusters that perform significantly worse in comparison to the global model.

In terms of precision, the highest-performing cluster we have the cluster 4, followed by the cluster 7, 1, 3 and the lowest better-performing cluster is the cluster 5. In Figure Figure 5.2 we can see the precision-recall curve across the threshold. For cluster 4 and 7, when the threshold is in one, the precision reach the highest level, while for the other clusters decreases.

The only cluster that did not have a significant difference in comparison to the global model in terms of precision is the cluster 0, rejecting the hypothesis that the difference between the global model and the cluster 0 is significant since the p-value in the Mann Whitney test was $> 0.05/9$

The clusters with a significantly lower performance in comparison to the global model are the cluster 2, followed by the cluster 6 and as the worst-performing cluster in terms of precision we have the cluster 8, since the median was below the global model median and the p-values of the Mann Whitney test were $> 0.05/9$ .In figure Figure 5.3 we can see that its precision-recall curve is the first to drop in comparison to all the other clusters precision-recall curve.

In terms of recall as Figure Figure 5.5 shows, 5 clusters performed better than the global model, while only 3 performed significantly worse.

The results are mostly the same as for precision, except for the clusters 0 and 1, where cluster 1 did not have a significant difference in respect to the global model in terms of recall.

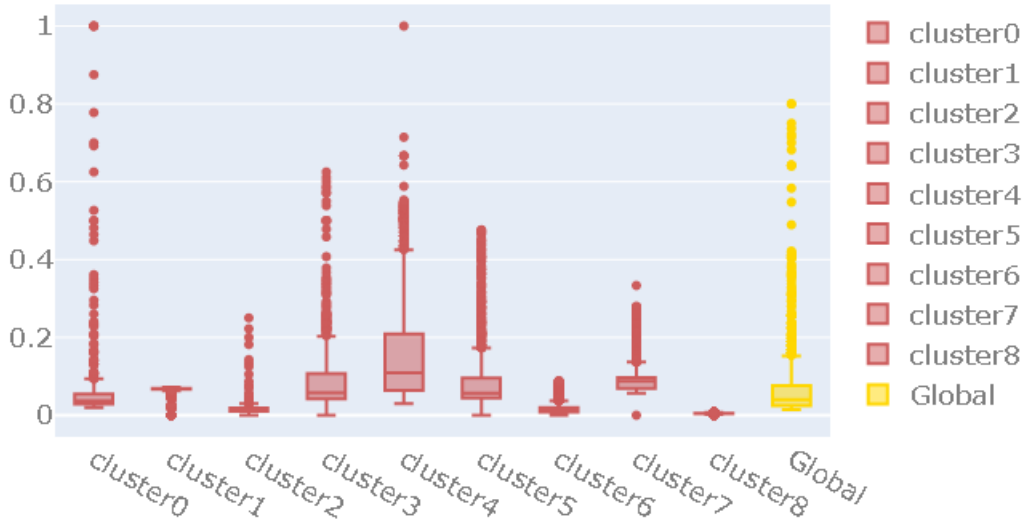

Figure Figure 5.4 Precision distribution across the threshold comparing each local model with the global model

The current results in terms of precision and recall mean that the majority of clusters retrieve more relevant items than the global model and in a more precise way. Local models (pure) bring performance advantages over the Global model approach and the majority of models perform different across the threshold values. The results reject the null hypothesis that

Model	Precision Median	Recall Median	F1 score
Global Model	0.040	0.24	0.0685
cluster 0	0.036	0.27	0.0635
cluster 1	0.058	0.25	0.0941
cluster 2	0.012	0.17	0.0224
cluster 3	0.058	0.33	0.0986
cluster 4	0.109	0.48	0.1776
cluster 5	0.056	0.32	0.095
cluster 6	0.009	0.146	0.0169
cluster 7	0.087	0.322	0.136
cluster 8	0.04	0.97	0.07

Table Table 5.2 Median precision, recall and F1 score across all thresholds

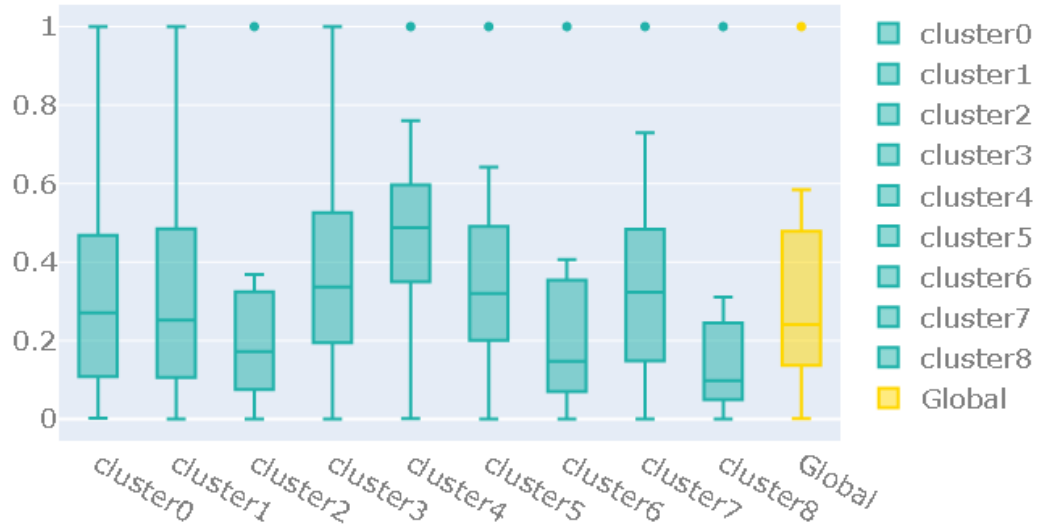

Figure Figure 5.5 Recall distribution across the threshold comparing each local model with the global model

there is no precision and recall differences between the local models and global models across the different thresholds.

5.1.3 Results Discussion

After answering RQ1 and RQ2 we have a better understanding of the performance differences between local models and global models. We saw that local models generally perform as good (local models) or even better (hybrid local models) as the global model, however, why? Here we explore two hypotheses.

A first hypothesis is that repeated job descriptions in the clusters might influence the performance of the clusters. By exploring the job description data set of our study, and how they are distributed across the 9 clusters, we observed several cases of repeated descriptions, i.e., the same description posted multiple times, for example because the job is recurrent or because a generic description is used for those specific jobs. The more repetition within a cluster, the less variance and hence the more specific that cluster's model could be, leading to better performance (especially when a new job has identical description as the rest of the cluster). As such, here we want to analyze whether job repetition could explain the success of clusters 3, 4, 5 and 7, and potentially 0/1, in RQ2.

To see the impact of repeated items across each individual cluster, we plotted the percentage of repeated job descriptions across the clusters. Figure Figure 5.6 shows that cluster 5,

4, 7 and cluster 1, have the biggest number of repeated items, suggesting that the highest performing clusters have a considerable amount of repeated job descriptions. Cluster 8 and 6, the lowest performing clusters are of the ones that have less repeated job descriptions. These results show that repeated items influence in the performance of a model. More research has to be done in order to determine the other factors that can make a model perform better or worse.

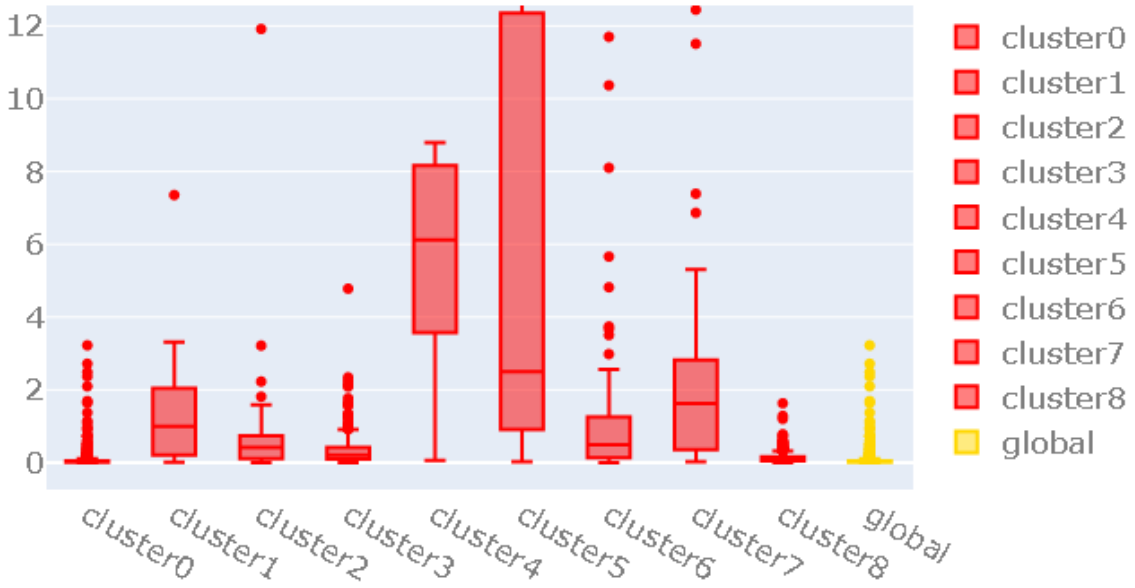

Figure Figure 5.6 Percentage of repeated job descriptions across the clusters.

A second hypothesis that might affect how a cluster model will perform is the data size and data distribution. At the moment of building the clusters, the k-means classification works by receiving a similarity matrix as input. Even if we determine correctly the number of clusters, some of the models may underperform because they may not have enough elements to train that particular cluster. More research has to determine which other factors influence the performance of each cluster and how can we improve these flaws.

5.1.4 Cross-validation importance

Without cross-validation, the results we see can be a reflection of the way the data was sorted or a coincidence on the way it was split for training and testing.

When we evaluate the clusters without cross-validation in terms of precision and recall, the results differ in comparison when the cross-validation is done. In Figure Figure 5.7 and

Figure Figure 5.8 we can see the precision and recall distribution values across the threshold being evaluated without cross-validation. If we compare it with the precision and recall values with cross-validation in Figure Figure 5.4 and Figure Figure 5.5, we can see that the cluster 8 won't be the lowest-performing cluster, in the evaluation without cross-validation the lowest-performing cluster is cluster 0. It's noticeable that the biggest impact of the cross-validation is in the precision values. Figure Figure 5.9 shows the precision-recall intersection comparison without cross-validation. It overestimates the performance of the local models if we compare it with the Figure Figure 5.1 where it was evaluated with cross-validation.

The evaluation of the local models is relevant to identify the clusters that are underperforming so they can be substituted by the global model. iIf the evaluation of these models is done without cross-validation there is a big risk of replacing the incorrect clusters and misinterpreting the model's performance.

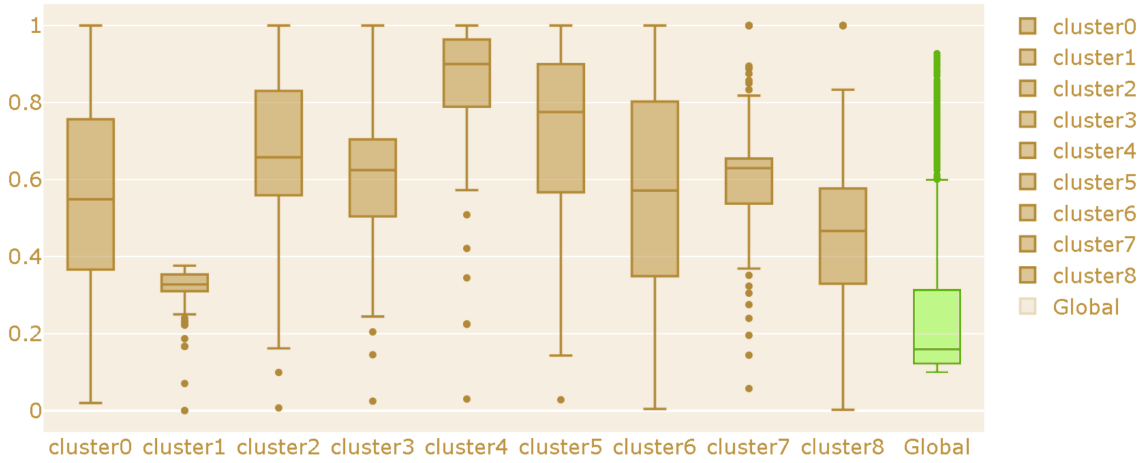

Figure Figure 5.7 Precision distribution across the threshold without cross-validation.

5.2 Threats to Validity

In terms of external validity, this case study was performed on anonymized data provided by the human resources company Airudi, in French. Although we cannot generalize the results for all types of data distribution, this case study provides an interesting point of departure for future studies with different classification techniques and data distributions, and seems to confirm that the concept of local models for candidate matching provides better performance than the use of global models.

Our method for determining the ideal number of clusters is based on earlier work by Charrad et al. [24]. Using the elbow and silhouette method for our case study, we had a suggestion for

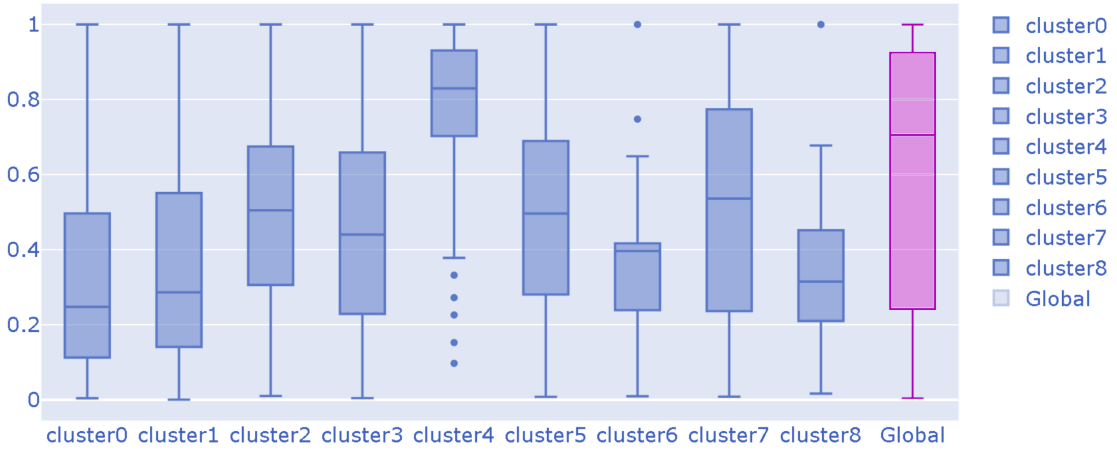

Figure Figure 5.8 Recall distribution across the threshold without cross-validation.

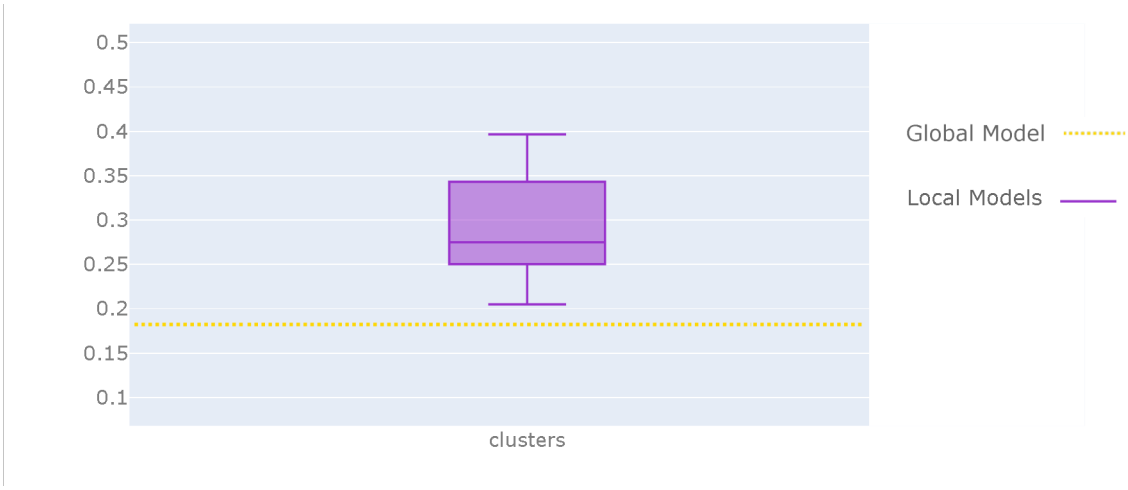

Figure Figure 5.9 Precision-recall intersection comparison without cross-validation.

$k = 9$. Nonetheless, with the two methodologies, we had different ideal k suggestions, where the nearest ideal k we had for both cases was nine. We do not know if this methodology can be generalized for the entire job-candidate matching context. Despite this, we were able to see the performance improvements the local models over the global models. More research has to be done to determine which is the best way to determine the clusters depending on the data particularities.

For a correct performance, local models depend on a proper data distribution. Therefore we cannot generalize our results for all kinds of data distributions and sizes.

In terms of construct validity, we were working with noisy data, i.e the candidate and job descriptions in some cases were inconsistent, incomplete, illogical, noisy or null. After identifying these cases, we discarded 8% of the job descriptions that didn't have enough features to be trained. Since it is difficult to try to predict each particular case, it is possible some of this noisy data still went through. The job or candidate descriptions may be incomplete, with orthographic errors or just noise. This data will impact differently depending on its distribution across the clusters but their impact is negligible.

Our approach used Random Forest for the model training based on previous studies [8] where they have a good result using Random Forest on a clustered approach. More research has to be done in order to know which is the best algorithm to train each cluster and which is the necessary tuning to perform, depending on the data distribution on each cluster.

Regarding internal validity, we worked with data that was in Canadian French for the largest majority. The parts of the approach that are language specific is: the tf-idf matrix creation since its based on semantic similarity, the tokenization of the words, stop-words removal and the semantic similarity between the job and candidate description. In order to do the tf-idf matrix and calculate the semantic similarity, the "md" French dictionary from the Spacy python package was used. The dictionary has multitask capabilities and is trained on UD French Sequoia dictionary and Wikiner, that is trained on Wikipedia pages [26]. This can make it inconsistent in the associations it makes to the words in certain contexts. This has an impact when calculating the semantic similarity. For the stop-words removal and tokenization we used the nltk [41] french dictionary that includes the french corpora and lexical resources. The results can be generalized to all languages if the dictionaries used have the same quality to do the correct associations.

CHAPTER 6 CONCLUSION

In the context of human resources, candidate matching is a basic process to select the most suitable candidates given a certain job description. To do the matching, researchers have used different techniques such as semantic similarity and Artificial Intelligence Models for classifying the best candidates, or hybrid techniques. However, the use of local models in other areas has shown to bring better performance when models are built for clusters of related data instead of on the entire data. Therefore, the individual local models have the potential to improve the performance in comparison to a single global model.

In the context of our study, we investigated the performance differences between local and global models using real data (3023 job descriptions and 103237 candidates) provided by the Airudi Enterprise. We extracted the job descriptions and candidate features from resumes and then clustered the job descriptions using the k-means algorithm. Then, We trained each cluster individually using the random forest algorithm.

We observed that local models perform significantly better than a global model when precision=recall across the threshold. In the local models, the median of the precision-recall intersection outperform by 11.6% the global model median. The difference is especially significant when we substitute the under-performing local models with the global model, obtaining hybrid local models. In addition, the performance of each cluster changes across the threshold values and the majority of clusters outperform in terms of precision and recall.

We also observed an influence in performance improvement when there were more repeated job description. Such clusters had a better performance than the ones with fewer repeated job descriptions. As such, this factor seems to have an influence on the performance of each cluster.

In terms of precision, five clusters had a significant improvement in relation to the global model. However, in the case of one cluster, there is no significant difference compared to the global model while only three models perform significantly worst.

In terms of recall, five clusters perform significantly better than the global model. There is no significant difference with only one cluster in respect to the global model. Three clusters perform significantly worst in comparison to the global model.

According to our case study, the concept of local models, either in its pure sense or when combined with the global model (hybrid), built on a hybrid way can bring performance advantages over the use of a single global model in the context of candidate matching.

The case study results show the advantages that local models can bring in the e-recruiting domain, establishing a point of departure to continue our work. The following research topics are worth investigating in future work:

- Explore and compare different clustering algorithms and determine the most suitable for the e-recruiting domain. Explore if an algorithm will be more suitable or not depending on the data distribution.
- Compare the different learning algorithms for training local models, determine which one is more suitable in the e-recruiting context and the correct model tuning for each cluster on the learning algorithm.
- Explore the use of word embeddings in the e-recruiting context and compare it with the tf-idf approach. Determine which is the most appropriate way to represent the semantic similarity.

REFERENCES

[1] S. T. Al-Otaibi and M. Ykhlef, "A survey of job recommender systems," *International Journal of Physical Sciences*, vol. 7, no. 29, pp. 5127–5142, 2012.

[2] e. a. Ibis, world, "Online recruitment sites in the us market size 2005–2025," https://www.ibisworld.com/industry-statistics/market-size/online-recruitment-sites-united-states/, accessed: 2020-07-25.

[3] P. Brézillon, "Context in artificial intelligence: Ii. key elements of contexts," *Computers and artificial intelligence*, vol. 18, pp. 425–446, 1999.

[4] S. Maheshwary and H. Misra, "Matching resumes to jobs via deep siamese network," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 87–88.

[5] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, 2009.

[6] N. Bettenburg, M. Nagappan, and A. E. Hassan, "Think locally, act globally: Improving defect and effort prediction models," in *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)*. IEEE, 2012, pp. 60–69.

[7] H. Aydadenta *et al.*, "A clustering approach for feature selection in microarray data classification using random forest." *Journal of Information Processing Systems*, vol. 14, no. 5, 2018.

[8] S. L. A. Lee, A. Z. Kouzani, and E. J. Hu, "Random forest based lung nodule classification aided by clustering," *Computerized medical imaging and graphics*, vol. 34, no. 7, pp. 535–542, 2010.

[9] "Screening process," https://www.hrtechnologist.com/articles/recruitment-onboarding/what-is-candidate-screening-and-selection/, accessed: 2020-10-05.

[10] S. Harispe, S. Ranwez, and S. Janaqi, *Semantic similarity from natural language and ontology analysis*. Morgan & Claypool Publishers, 2015.

[11] "Definition: people analytics (hr analytics)," http://searchhrsoftware.techtarget.com/definition/human-resources-analytics-talent-analytics, accessed: 2019-12-09.

[12] "The 8 hr analytics every manager should know about". in forbes tech," https://www.forbes.com/sites/bernardmarr/2016/03/01/the-8-hr-analytics-every-manager-should-know-about/#5e3ba218788f, accessed: 2020-07-25.

[13] "Baby steps in hr technology: What is resume parsing?" https://recruiterbox.com/blog/baby-steps-in-hr-technology-what-is-resume-parsing-2/, accessed: 2019-12-09.

[14] V. S. Kumaran and A. Sankar, "Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (expert)." *International Journal of Metadata, Semantics and Ontologies*, vol. 8, no. 1, pp. 56–64, 2013.

[15] S. Guo, F. Alamudun, and T. Hammond, "Résumatcher: A personalized résumé-job matching system," *Expert Systems with Applications*, vol. 60, pp. 169–182, 2016.

[16] D. Lee, M. Kim, and I. Na, "Artificial intelligence based career matching," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 6, pp. 6061–6070, 2018.

[17] P. Xu and D. Barbosa, "Matching résumés to job descriptions with stacked models," in *Canadian Conference on Artificial Intelligence*. Springer, 2018, pp. 304–309.

[18] J. Purohit, A. Bagwe, R. Mehta, O. Mangaonkar, and E. George, "Natural language processing based jaro-the interviewing chatbot," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, pp. 134–136.

[19] "Sap and russian startup robot vera boost hr recruiting efficiency," https://news.sap.com/2018/10/sap-joins-forces-robot-vera/, accessed: 2020-28-05.

[20] "Meet the robot that's hiring humans for some of the world's biggest corporations," https://news.sap.com/2018/10/sap-joins-forces-robot-vera, accessed: 2020-28-05.

[21] M. de Prado, J. Su, R. Dahyot, R. Saeed, L. Keller, and N. Vallez, "Ai pipeline-bringing ai to you. end-to-end integration of data, algorithms and deployment tools," *arXiv preprint arXiv:1901.05049*, 2019.

[22] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[23] C. Robert, "Machine learning, a probabilistic perspective," 2014.

[24] M. Charrad, N. Ghazzali, V. Boiteau, and A. Niknafs, "Nbclust package: finding the relevant number of clusters in a dataset," *J. Stat. Softw*, 2012.

[25] "Spacy semnatic similarity," https://spacy.io/usage/vectors-similarity, accessed: 2020-28-05.

[26] "Spacy dictionary," https://spacy.io/models/fr#fr_core_news_sm, accessed: 2020-28-05.

[27] A. N. Chris Piech, "Kmeans," https://stanford.edu/~cpiech/cs221/handouts/kmeans.html, accessed: 2020-07-25.

[28] "Tokenization," https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html, accessed: 2020-10-05.

[29] "Nltk stemer," https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html, accessed: 2020-28-05.

[30] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[31] "Determining the optimal number of clusters: 3 must know methods," , accessed: 2020-03-09.

[32] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999.

[33] G. J. McLachlan, K.-A. Do, and C. Ambroise, *Analyzing microarray gene expression data*. John Wiley & Sons, 2005, vol. 422.

[34] M. Kuhn, K. Johnson *et al.*, *Applied predictive modeling*. Springer, 2013, vol. 26.

[35] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *The Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.

[36] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[37] S. McLeod, "What does effect size tell you? simply psychology," 2019.

[38] J. Pearce and B. Derrick, "Preliminary testing: The devil of statistics?" *Reinvention: An International Journal of Undergraduate Research*, vol. 12, no. 2, 2019.

[39] G. MACBETH, E. RAZUMIEJCZYK, and R. A. D. LEDESMA, "Cliff's Delta Calculator: A non-parametric effect size program for two groups of observations," *Universitas Psychologica*, vol. 10, pp. 545 – 555, 05 2011. [Online]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1657-92672011000200018&nrm=iso

[40] "Bonferrorni correction," https://mathworld.wolfram.com/BonferroniCorrection.html, accessed: 2020-07-25.

[41] "Nltk dictionary," http://www.nltk.org/book/ch02.html, accessed: 2020-28-05.