Escuela de Negocios. Master in Management + Analytics

Job profile demand understanding in international financial organizations: a natural language processing approach

Autoría: Rivas, Richard
Año: 2024

UNIVERSIDAD TORCUATO DI TELLA

MASTER IN MANAGEMENT + ANALYTICS

JOB PROFILE DEMAND UNDERSTANDING IN INTERNATIONAL FINANCIAL ORGANIZATIONS: A NATURAL LANGUAGE PROCESSING APPROACH

THESIS

Richard Rivas

May, 2024

Tutor: Luciano del Corro, PhD

Abstract

This study presents an innovative approach in understanding the job profiles demanded in international financial organizations by using Natural Language Processing (NLP) techniques. Multiple job descriptions published on the employment portals of two international financial institutions were analyzed using a language model to generate text embeddings. Subsequently, various supervised and unsupervised machine learning methods were applied to these embeddings. The K-Means algorithm was used to segment job profiles, and an XGBoost model was employed to predict the recruitment process duration for each position. This integrated NLP and machine learning approach yielded valuable information to identify the most challenging positions and skills to find in the International Financial Institutions space, which allow to enhance and optimize the recruitment and selection processes in these organizations.

Resumen

Este estudio presenta un enfoque innovador para comprender los perfiles laborales demandados en organizaciones financieras internacionales utilizando técnicas de Procesamiento del Lenguaje Natural (NLP). Se analizaron múltiples descripciones de empleo publicadas en los portales de dos instituciones financieras internacionales utilizando un modelo de lenguaje para generar *embeddings* de texto. Posteriormente, se aplicaron diversos métodos de aprendizaje automático supervisado y no supervisado a estos datos. Se utilizó el algoritmo K-Means para segmentar los perfiles laborales, y se empleó un modelo XGBoost para predecir la duración del proceso de contratación para cada posición. Este enfoque integrado de NLP y aprendizaje automático proporcionó información valiosa para identificar las posiciones y habilidades más desafiantes de encontrar en el espacio de las Instituciones Financieras Internacionales, lo que permite mejorar y optimizar los procesos de reclutamiento y selección en dichas organizaciones.

En todo amar y servir

Table of Contents

1. Introduction

1.1 Context

In recent years, a new paradigm has surfaced in organizational development: "Skill-Based Organizations." This approach advocates for the segmentation of positions within an organization based on the required skills, regardless of job titles.

In today's labor market, numerous organizations are reshaping their organizational architectures and redefining roles and positions. However, the challenge lies in the diverse nomenclature used by different companies and international organizations for their positions. This diversity might make it difficult to comprehend the current demand and supply of skills, impacting talent acquisition and internal mobility processes. Positions with the same title across various departments may require different skills, knowledge, or experience, leading to delays in candidate searches and increased uncertainty regarding resource allocation for these roles.

To grasp the existing skill landscape within an institution, it is crucial to classify skill profiles and their corresponding demand within the organization, as well as understanding the ability to fill positions with these profiles and assessing the supply of job seekers willing to apply for roles requiring specific skill sets, either internally or externally.

Some international financial institutions (IFIs) are presently undergoing a transition toward this skill-based infrastructure, recognizing that it entails a highly intricate process of organizational change. A key step in establishing these skill groups involves surveying the skills in demand within the company. This enables the association of different skill collections with the current position architecture in the organization (Goldberg, 2020).

This work aims to establish a foundation for creating a job profile architecture and skill classification for the most in-demand professionals within these institutions through an NLP-based approach. To the best of our knowledge, this is the first study to propose this integrated methodology in the field of development finance, job market understanding, and time-to-fill prediction.

1.2 Problem

The focus of this work consists of solving a dual problem, one of them being providing recommendations for the reduction of one of the most important Key Performance Indicators for Human Resources departments, which is the ***Average Time to Fill*** an open position.

The ***Time to Fill*** metric is calculated by measuring the number of days from the day of approval of a job requisition to the day the candidate accepts the job offer. The ***Average Time to Fill*** consists of calculating the average of the ***Time to Fill*** of all positions in a given period of time, by following the formula:

$$1)\quad TTF_{avg} = \frac{TTF_1 + TTF_2 + TTF_3 + \dots}{Total\, number\, of\, positions\, filled}$$

Reducing the overall average of this metric could have an important impact on the business. It allows for understanding the average time for either replacing a departed employee or hiring new ones, also, having vacant positions for a long time decreases productivity within the organization, with business units having to redistribute tasks within a limited number of employees.

The models currently being used to predict Time to Fill perform quite poorly, these tend to show a high variance and are being trained by using conventional variables such as the position level, and number of applicants to predict if a position might take longer to fill than average. Which could suggest that using non-conventional variables such as the text data on the job description could yield more significant results in terms of predictive performance.

However, the mere prediction of the Time to Fill a position doesn’t necessarily help in improving the current recruitment processes, this implies that an analysis to understand the factors that contribute to positions filling being delayed is required in order to source candidates and search professional profiles under a more targeted approach.

This leads to the second problem that this work aims to solve, which is that in the organizations where the present work was carried out, there is no classification of positions based on collections of skills that can be used as a common language. This implies that the two positions may have the same name, but the required skills for each of them might be completely different.

A clear example of this can be two positions of "Financial Analyst," where one may only require skills in Excel, financial modeling, and investment project evaluation, while the other position may require programming skills in VBA, Python, and automation, without necessarily requiring domain knowledge in finance. Both positions belong to different profiles despite having the same name. Therefore, the resources allocated to the recruitment of both positions will be different, as well as the number of applicants that both positions will receive.

Understanding these differences will allow for a better understanding of the current demand and supply of skills in the job market in which some International Financial Organizations operate.

1.3 Objective

The present work aims to create a Machine Learning model using unstructured text data in order to predict whether a position is prone to taking a longer Time to Fill than the overall average. As well as building an initial categorization of profiles within the organizations in which it was carried out, this will help to provide insights that allow understanding both the demand and supply of different job profiles using different Natural Language Processing and Unsupervised Machine Learning techniques.

The processing of the text data will be done by using an open source Language Model (LLM) in order to generate their corresponding document embeddings.

1.4 Summary of Results

The results in this work reveal that certain positions encounter sourcing challenges due to low application volumes, highlighting the need for innovative strategies such as candidate pools or referral programs to expedite the recruitment process. This could also help with managing high application loads, as this emerges as a possible explanation for a longer Time to Fill for other positions.

This study recommends implementing the developed Time to Fill prediction model in order to identify positions prone to delays and allocating resources effectively. Furthermore, it presents an initial job profile map to enhance skill identification and a more granular analysis of the recruitment, sourcing and candidate selection process to better support business decisions. It provides a framework for interpreting model outputs that could allow for better use of the existing domain knowledge of the organization’s operations in terms of sourcing candidates and assigning resources to filling positions.

Overall, this thesis provides valuable insights into enhancing Time to Fill metrics in organizational recruitment processes, offering practical recommendations to streamline operations and improve efficiency.

2. Description of the Data

The data used for this study consists of two sets, one unstructured and one structured.

The first one being a corpus of 7,853 documents containing the selection criteria for the positions recruited during the last 3 years among both organizations.

The second one being a dataset containing information related to these positions, such as position level and applications received.

2.1 Unstructured Data

Unstructured data refers to data that lacks a predetermined form or structure, making it difficult to fit into a conventional table. This type of data is usually found in multiple forms, such as emails, reports, web pages (Ingle, 2012).

Due to the unavailability of a reporting tool within the organization´s position management system that could allow for mass downloading the advertised job descriptions, a Web Scraper was developed using the Selenium framework and the *BeautifulSoup* library in Python. This scraper navigates the links within the institution’s internal career and jobs platforms, where job descriptions were published and saves each of them in HTML format.

Job descriptions of filled positions from December 2021 to December 2023 were successfully retrieved, extracting 7,853 job descriptions from two International Financial Institutions (IFIs).

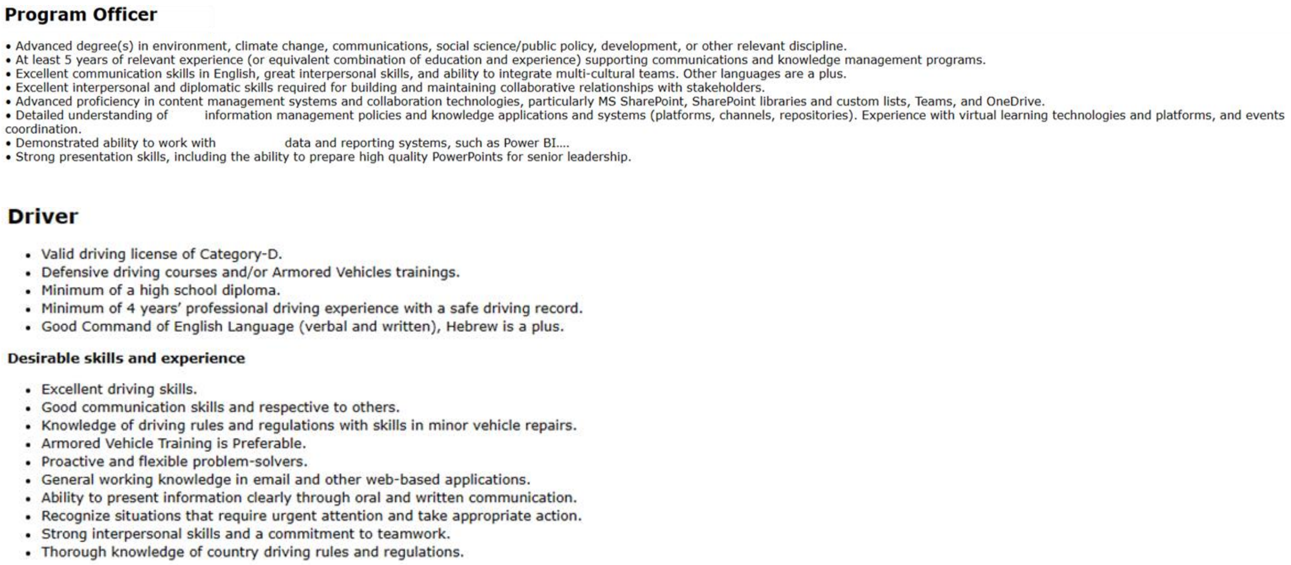
The general structure of the documents includes the following elements:

- A brief description of the institution's operations.
- General description of the position and its corresponding department.
- Tasks and responsibilities.
- Required Competencies.

Given the focus of this work on understanding the skills in demand and creating skill-based profiles, the HTML documents were processed by removing tags and corresponding sections to extract only the Required Competencies section. This section contains various educational, experiential, and skill requirements necessary for performing the role.

For the present work, most of the data used is unstructured, as it consists of *.txt* documents processed from these job descriptions using multiple Natural Language Processing techniques.

The following are examples of the documents used for this research. These contain the required skills for the ideal candidate under the *Selection Criteria* section.



Program Officer

- Advanced degree(s) in environment, climate change, communications, social science/public policy, development, or other relevant discipline.
- At least 5 years of relevant experience (or equivalent combination of education and experience) supporting communications and knowledge management programs.
- Excellent communication skills in English, great interpersonal skills, and ability to integrate multi-cultural teams. Other languages are a plus.
- Excellent interpersonal and diplomatic skills required for building and maintaining collaborative relationships with stakeholders.
- Advanced proficiency in content management systems and collaboration technologies, particularly MS SharePoint, SharePoint libraries and custom lists, Teams, and OneDrive.
- Detailed understanding of information management policies and knowledge applications and systems (platforms, channels, repositories). Experience with virtual learning technologies and platforms, and events coordination.
- Demonstrated ability to work with data and reporting systems, such as Power BI....
- Strong presentation skills, including the ability to prepare high quality PowerPoints for senior leadership.

Driver

- Valid driving license of Category-D.
- Defensive driving courses and/or Armored Vehicles trainings.
- Minimum of a high school diploma.
- Minimum of 4 years' professional driving experience with a safe driving record.
- Good Command of English Language (verbal and written), Hebrew is a plus.

Desirable skills and experience

- Excellent driving skills.
- Good communication skills and respective to others.
- Knowledge of driving rules and regulations with skills in minor vehicle repairs.
- Armored Vehicle Training is Preferable.
- Proactive and flexible problem-solvers.
- General working knowledge in email and other web-based applications.
- Ability to present information clearly through oral and written communication.
- Recognize situations that require urgent attention and take appropriate action.
- Strong interpersonal skills and a commitment to teamwork.
- Thorough knowledge of country driving rules and regulations.

Figure 1. *Examples of Job Descriptions used in this work.*

2.2 Structured Data

The following table describes each of the columns in the dataset used in this work. The dataset contains 7,853 records of positions closed since year 2021 for 4 important International Financial Institutions (IFIs). However, for the supervised training algorithms implemented in this work, only 5,476 rows contained information in the predicted variable (y: discretized "*time_to_fill_scaled*", see section 4.3).

The following table (Table 1) contains a description of every variable contained in the structured dataset.

Column Name	Variable Type	Description
position_code	String	Id of the position
position_desc	String	Job description text

preprocessed_text	List	List containing the preprocessed job description tokens
tokens_len	Integer	Number of tokens in the description
position_level	Integer	Level of the Position
internal_external	Categorical (String)	Flags if the position is internal or external
job_title	String	Title of the position
organization	Categorical (String)	Organization
internal_applications	Integer	Number of applications the position received from internal candidates
external_applications	Integer	Number of applications the position received from external candidates
time_to_fill	Integer	Time to Fill the position (in days)
total_applications	Integer	Number of applications the position received from both internal and external candidates
total_applications_scaled	Float	Z-Score normalized number of total applications
time_to_fill_scaled	Float	Z-Score normalized Time to Fill the position
position_level_group	Categorical (String)	Describes the type of work being performed by the position (Operational, Non-Operational, Executive)
bigrams	List	List of preprocessed bigrams in the job description

Table 1. Description of Structured Data

3. Background

The following chapter provides a theoretical framework for every instance of data processing performed in the current work. It can be used as a reference when reviewing the Methodology chapter.

3.1 Data Collection

3.1.1 Web Scraping

Web Scraping, also known as *Web Crawling,* is a series of techniques and methods used to extract data from a website using an automated software (Khder, 2021)

In this work, the BeautifulSoup and Selenium python libraries were used to scrape the documents from the institution's internal career platforms, in which jobs were posted.

3.2 Data Preparation: Unstructured Data Processing

The following techniques were used for text processing to perform descriptive statistical analysis on the corpus and enhance interpretability of the model results when providing recommendations.

3.2.1 Text Preprocessing

Text preprocessing is one of the most important steps when it comes to performing Natural Language Processing tasks and data preparation.

The treatment of this text data was performed to structure the information and build a consolidated dataset for the subsequent tasks.

The methods to process the text data were the following:

Lowercasing:

Lowercasing involves converting all text data to lowercase, ensuring uniformity. This helps with addressing variations in words due to capitalization (e.g., “Financial Analysis” is transformed into “financial analysis”). Facilitating a consistent representation of the term across the dataset (Camacho-Collados, 2017).

Tokenization:

Tokenization breaks down text into discrete units (tokens), such as words or *sub-words*". Working with individual tokens enables subsequent analyses, such as feature extraction, descriptive analysis, and visualization, for example: the term "Natural Language Processing" is tokenized into ["Natural", "Language", "Processing"] (Bird, Klein, & Loper, 2009).

Stop-Word Removal:

Stop words (e.g., "the," "and," "in") lack substantial meaning and can introduce noise into the model or analysis. Eliminating stop words streamlines the focus on terms that retain content, for example: the sentence "Demonstrated ability to work with databases" simplifies to "Demonstrated ability work databases (Bird, Klein, & Loper, 2009).

The Stop-Word list used in this work was the default list from the NLTK library in python.

Lemmatization:

Lemmatization reduces inflected words to their base or dictionary form (lemmas). Normalizing word variations enhances consistency (e.g., "running" → "run"). The transformation of "better" to "good" is an example of lemmatization (Bird, Klein, & Loper, 2009).

The lemmatization performed in the documents of this work was done using the default lemmatizer contained in the NLTK library in python.

3.2.2 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF stands for Term Frequency-Inverse Document Frequency. It quantifies the importance or relevance of terms within a document relative to a collection of documents (corpus). This technique plays a crucial role in information retrieval, natural language processing, and machine learning. It can be used to extract keywords and relevant information within a corpus, as well as clustering documents.

The model consists of two parts:

Term Frequency (TF)

The TF score measures how frequently a term occurs in a document. It is calculated as the number of occurrences of a term divided by the total number of terms in the document. Higher TF indicates greater relevance of the term within the document.

$$2)\ \mathrm{TF} = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

Inverse Document Frequency (IDF)

The IDF score of a term represents the importance of a term across the entire document corpus. IDF penalizes common terms and emphasizes rare ones.

It is calculated as the logarithm of the total number of documents divided by the number of documents containing the term.

$$3)\ \mathrm{IDF} = \log\left(\frac{\text{number of documents in the corpus}}{\text{number of documents in the corpus containing the term}}\right)$$

The TF-IDF of a term is calculated by multiplying the TF and IDF scores:

$$4)\ \text{TF-IDF} = \mathrm{TF} \times \mathrm{IDF}$$

The higher the score, the more relevance the term contains within the analyzed corpus.

The TF-IDF implementation used in this work was the one contained in the *scikit-learn* framework in python.

3.2.3 Word and Document Embeddings

The foundational concept of embeddings refers to a representation of textual or images information that can be used by machine learning models. It is a mathematical representation of an object.

The following definitions of embeddings are relevant to the current work:

Word Embeddings:

Word embeddings are a type of vector representation for words in a high-dimensional space. These vectors capture semantic relationships between words based on their context in a given corpus (Poetsch, Correa, Freitas, 2019).

Each word is mapped to a dense vector, where similar words have vectors that are close together. This representation allows machine learning models to understand and work with words more effectively.

Popular word embedding techniques include Word2Vec, GloVe, and FastText.

For instance, Word2Vec learns word embeddings by predicting the context of a word based on its neighboring words in a large text corpus (Meijer, H.J., Truong, Karimi, 2021)

Document Embeddings:

Document embeddings extend the concept of word embeddings. These are numerical representations of textual content, which can include words, sentences, or entire documents. They are designed to capture the semantic meaning of the text in a way that is interpretable by computers.

The primary goal of document embeddings is to convert text into a machine-readable format. This is achieved by representing the text as vectors in a high-dimensional space, typically consisting of hundreds of dimensions (Meijer, H.J., Truong, Karimi, 2021).

To create document embeddings, algorithms analyze the text and map it to vectors based on the context and relationships between words. This process often involves techniques like word embedding, where individual words are first converted into vectors, and then combined to form the document embedding.

Document embeddings are crucial for various natural language processing (NLP) tasks, such as semantic search, document classification, and information retrieval.

Using document embeddings can help overcome the limitations of traditional text representation methods like one-hot encoding or TF-IDF, which can be sparse and inefficient. Embeddings provide a dense and meaningful representation, allowing for the capture of semantic relationships between different pieces of text (Meijer, H.J., Truong, Karimi, 2021)

To better illustrate the usage of document embeddings and how these were generated for the purpose of this work, it is important to introduce the concept of Large Language Models.

3.2.4 Large Language Models (LLMs)

Language Models were utilized as the main tool for generating the document embeddings used for Unsupervised and Supervised Machine Learning tasks.

Large Language Models (LLMs) are advanced artificial intelligence systems capable of processing and generating text with coherent communication. They are designed to approximate human-level performance on various tasks such as translation, summarization, information retrieval, and conversational interactions. LLMs are characterized by their large scale, often consisting of billions of parameters, and are trained on extensive datasets and to perform a wide range of natural language processing tasks with high accuracy (Naveed, H. et al, 2024).

These models are trained with multiple text corpora using self-supervised learning techniques. During this phase, these models learn to predict missing words in sentences in order to understand the context of the word, which makes these models capable of retaining the semantic meaning of the terms (Devlin, Chang, Lee & Toutanova, 2018)

There are several LLM architectures: BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) the most popular ones. The latter was developed by *OpenAI* and released in June 2018. Introduced by Google in 2018, BERT is a milestone in Natural Language Processing. It leverages the transformer architecture and bidirectional context to achieve state-of-the-art performance on various tasks.

A reduced version of a LLM was used in this work due to computational resource constraints, which is technically called as a Small Language Model (SLM) and could be used for specific use cases where efficient processing and embedding generation is required, more details of the model used are provided in section 4.2.3, figure 3.

3.3 Data Preparation: Structured Data Processing

3.3.1 Data Cleaning

To prepare the features for the Supervised and Unsupervised models used in this work, the structured data was also treated to ensure its quality and completeness.

According to Dasu and Johnson (2003), data cleaning is defined as the process of identifying and rectifying errors, inconsistencies and inaccuracies in raw data. It is a crucial step for the subsequent analysis as it helps in ensuring their accuracy and reliability.

In this work, the structured data was organized and clean to some extent, however, one of the key steps to ensure the confidentiality of the data was to standardize the Time to Fill and number of applicants variables, taking its average as the center of the distribution.

The Standardization or Z-Score Normalization formula is:

$$5)\, Z = \frac{x - \mu}{S}$$

Where:

- Z represents the Z-Score
- x is the raw data value
- μ is the mean of the data
- S represents the standard deviation of the data

3.3.2 Exploratory Data Analysis (EDA)

EDA is the initial phase of data analysis. It involves visually and statistically exploring data to understand its underlying patterns, relationships, and distributions (Anscombe, 1973).

Some fundamental techniques in this stage are:

Summary and Descriptive Statistics

Consists of calculating mean, median, variance, etc., to summarize data and identify distributions.

Data Visualization

Consists of creating histograms, scatter plots, box plots, etc., to visualize distributions and relationships between variables.

Correlation Analysis

Consists of viewing correlations between variables.

3.4 Models Used

3.4.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a standard tool in modern data analysis, used across diverse fields. It is a simple, non-parametric method for extracting relevant information from complex datasets. The goal of PCA is to identify the most meaningful basis to re-express a given dataset into a fewer number of features, while retaining the highest amount of information possible (Kurita, 2020).

In multiple research fields, datasets with several variables might show a relationship between them, which is statistically reflected by their covariance (Hilbert & Bühner, 2020). To find the common components in the covariance structure of a set of items, principal components can be mathematically identified through analysis of the covariance matrix of these items.

PCA has applications in exploratory data analysis, visualization, and data preprocessing. The data is linearly transformed onto a new coordinate system such that the directions (principal components) capturing the largest variation in the data can be easily identified (Hilbert & Bühner, 2020).

In this work, the PCA model was used to reduce the dimensions of the embedding vectors generated by the Language Model used, each of them consisted of 768 elements. The implementation of PCA used for this work was the one included in the *scikit-learn* framework in Python.

The following are the steps for executing the Principal Component Analysis algorithm.

Standardization of Variables:

First, continuous initial variables are standardized to have zero mean and unit variance. This ensures that all variables contribute equally during the analysis.

Let X be the matrix of standardized data, where each row corresponds to an observation and each column represents a feature.

Covariance Matrix:

Next, we compute the covariance matrix C based on the standardized data X. The covariance between features $i.$ and $j.$ is given by:

$$C_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(X_{ki} - \bar{X}i)\big(Xkj - \bar{X}_j\big)$$

6)

where n is the number of observations, $(\bar{X}i)$ is the mean of feature (i), and (Xki) represents the value of feature (i) for the kth observation.

Eigenvalues and Eigenvectors:

The eigenvalues are computed $(\lambda_1, \lambda_2, \ldots, \lambda_p)$ and corresponding eigenvectors $(v_1, v_2, \ldots, v_p)$ of the covariance matrix (C).

The eigenvalues indicate the amount of variance explained by each principal component, and the eigenvectors represent the directions (principal components) along which the data varies the most.

Selection of Principal Components:

The eigenvalues are sorted in descending order. The eigenvector corresponding to the largest eigenvalue is the first principal component, the second largest eigenvalue corresponds to the second principal component, and so on.

To reduce dimensionality, the top (k) eigenvectors (principal components) that explain most of the variance are chosen. Typically, components that account for a significant portion of the total variance are selected.

Feature Transformation:

The feature vector (V) is created, containing the selected eigenvectors (columns of (V)).

The transformed data (Y) is obtained by projecting the standardized data (X) onto the principal components:

$$7)\ Y = X \cdot V$$

The transformed data (Y) will now contain fewer dimensions ((k) components) while retaining essential information from the original data.

3.4.2 K-Means

K-Means clustering is an unsupervised machine learning method for finding cluster centers within a given set of unlabeled data (Hastie, Tibshirani & Friedman 2017).

This method has multiple applications in several domains, such as Document Classification, Customer Segmentation and Search Engines.

The following is a step-by-step explanation of how the K-Means algorithm works:

Initialization:

The number of clusters is initially defined by the user, denoted as K. After, each data point is randomly assigned to one of the K initial clusters and the centroid (mean) for each cluster is calculated.

Assignment and Update:

For each data point, the nearest centroid (cluster center) is detected based on the Euclidean distance. Then the data point is assigned to the cluster associated with the closest centroid.

This process iterates multiple times by recalculating the centroids for each cluster, which is set as the average of all data points within that cluster. Once the assignment of data points to clusters no longer changes significantly, the algorithm stops executing, this phenomenon is called *Convergence* (Hastie, Tibshirani & Friedman 2017)

As objective of K-Means is to minimize the within-cluster variance (also known as the sum of squared distances from data points to their cluster centroids. However, there are multiple ways of determining an optimal number of clusters and assessing their definitions.

The following are the evaluation methods used in this work to determine the optimal number of clusters:

Elbow Method:

The Elbow Method helps determine the optimal number of clusters K by analyzing the within-cluster variance (also known as the sum of squared distances from data points to their cluster centroids) by providing a visual representation.

This is achieved by fitting the clustering algorithm (in this case, K-Means) to the data for different values of K (usually ranging from 1 to a maximum value).

The within-cluster sum of squares (WCSS) for each K is calculated, then every WCSS value is plotted against the number of clusters. The point in which the curve of the plot forms an “elbow” is supposed to be the point in which the rate of decrease of the WCSS starts to slow down, which indicates a good trade-off between model complexity and variance reduction.

Davies–Bouldin Index (DBI):

The DBI is a metric for evaluating clustering algorithms based on the separation between clusters and the tightness within clusters.

It considers both inter-cluster distance (separation) and intra-cluster distance (compactness), a lower DBI values indicate better separation and tighter clusters.

The formula for DBI involves comparing the average distance between clusters with the average distance within clusters. It is an internal evaluation metric, meaning it uses features inherent to the dataset. However, a good DBI value doesn’t necessarily guarantee the best information retrieval.

Silhouette Score:

The Silhouette Score assesses the quality of clusters by measuring how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

For each data point, the average distance is computed against other points in the same cluster (a), then the average distance to points in the nearest neighbor cluster is calculated (b) (Rousseeuw, 1987)

Then the silhouette score for the data point is given by:

$$S = \frac{b - a}{\max(a, b)}$$

8)

The overall silhouette score is the average of individual scores across all data points. A silhouette score close to 1 indicates well-separated clusters, a score close to 0 suggests overlapping clusters and a negative score means the data point is likely misclassified.

Silhouette scores help choose the optimal K by maximizing the separation while minimizing overlap amongst clusters.

3.4.2 Extreme Gradient Boosting (XGBoost)

After the K-Means model was trained in order to identify the job profiles, multiple supervised learning algorithms were trained for predicting if the Time to Fill for each position would be higher than the average, some of the used models were Decision Trees and Random Forests, which yielded better results than other models like Logistic Regression or Gradient Boosting. However, the XGBoost model yielded the highest predictive power.

XGBoost is a supervised machine learning algorithm that has gained extensive popularity due to its efficiency, scalability, and remarkable performance across multiple domains.

Boosting:

This model leverages the concept of Boosting, which combines multiple weak learners such as decision trees into a strong ensemble model and iteratively corrects the errors by adding new trees (Meir & Rätsch, 2003). This allows for optimizing a loss function by adjusting the sample weights.

Regularization:

To prevent overfitting, XGBoost incorporates L1 (Lasso) and L2 (Ridge) regularization, which helps on limiting the complexity of the model and penalizing the exclusive usage of certain variables or make predictions based on noise.

Efficiency and Scalability:

One of the advantages of XGBoost is its speed and scalability, as it allows for handling large datasets efficiently (Bentéjac et al, 2019) at a lower computational cost, which will be an important feature when it comes to deploying the model into production.

Model Hyperparameters:

The following table (Table 3) shows the most commonly used hyperparameters for XGBoost in binary classification according to its documentation.

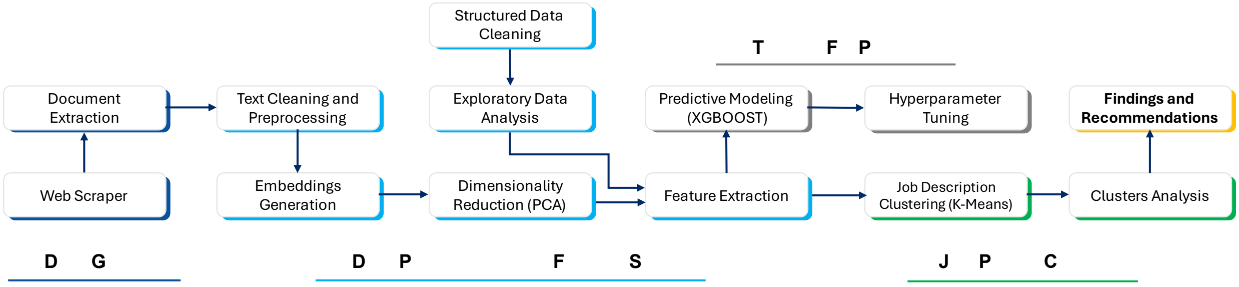
Hyperparameter	Description	Range of Possible Values
learning_rate (eta)	Step size shrinkage to prevent overfitting.	(0, 1]
n_estimators	Number of boosting rounds or trees to build.	[1, ∞)
max_depth	Maximum depth of a tree, controls model complexity.	[1, ∞)
min_child_weight	Minimum sum of instance weight (hessian) needed in a child.	[0, ∞)
subsample	Proportion of training data to randomly sample for growing trees.	(0, 1]
colsample_bytree	Proportion of features to randomly sample for building each tree.	(0, 1]
gamma	Minimum loss reduction required to make a further partition on a leaf node.	[0, ∞)
scale_pos_weight	Controls the balance of positive and negative weights, useful for imbalanced classes.	[0, ∞)

max_delta_step	Maximum delta step allowed for each tree's weight estimation to be.	[0, ∞)
colsample_bylevel	Proportion of features to randomly sample at each level.	(0, 1]
eval_metric	Evaluation metric used during training (e.g., logloss, auc).	Varies based on the task, e.g., 'logloss', 'auc'

Table 3. XGBoost Hyperparameters Description. XGBoost Documentation

4. Methodology

The following chapter describes the methods used to carry out the present work and solve the problems outlined in the *Introduction* chapter. This research consisted of four main stages; these are presented in the following figure (figure 2).



Structured Data Cleaning
T F P
Document Extraction
Text Cleaning and Preprocessing
Exploratory Data Analysis
Predictive Modeling (XGBOOST)
Hyperparameter Tuning
Findings and Recommendations
Web Scraper
Embeddings Generation
Dimensionality Reduction (PCA)
Feature Extraction
Job Description Clustering (K-Means)
Clusters Analysis
D G
D P F S
J P C

Figure 2. *Methodology Scheme*

The first stage consisted of outlining the process of gathering the documents to construct the corpus, which mainly entailed developing an application to automatically save job descriptions into a local folder.

The second stage describes the steps to prepare both structured and unstructured data to be consumed by the algorithms used. It explains how the document embeddings were generated and then consolidated into their corresponding aggregated features using the PCA algorithm. It also describes how the text data in each document was processed for its qualitative analysis and labeling of the clusters.

The third stage comprises the training and fine-tuning of the model for predicting if an open position will take a longer Time to Fill than the business-defined metric, which is the average Time to Fill for all positions.

The fourth and last stage consisted of the steps taken to analyze the clusters and assign their corresponding labels in order to better interpret and understand the demand of job profiles for the organizations taking part in this research.

4.1 Data Gathering

4.1.1 Web Scraping

Initially, the main challenge for this work lay in the extraction of the unstructured data. Given the multiple platforms in which the Job Descriptions were stored and the inability to massively download these documents, a web scraper was developed using the Selenium framework in Python.

4.1.2 Document Extraction

The developed application simulated a human typing a position code from a list in a .csv file containing the codes of the closed positions for the past three years, since the platform only retains the information for posted positions within this timeframe.

The program would input this position code into a search text within the jobs platform of the organization, which consisted of a website where the job descriptions were posted. After inputting the position code, the website would perform a search and show the corresponding job description in a floating window within an HTML "*<iframe>*" tag. The scraper would identify the HTML document containing the job description within this tag and save it locally as a ".html" file. The document was saved as a ".html" file in order to preserve the tags and better identify the sections when performing the corresponding text processing.

This sequence of instructions was done for every position code present in the institution's jobs platform, achieving a corpus of 7,837 documents.

4.2 Data Preparation and Feature Selection

4.2.1 Data Preparation: Structured Data Processing

Data Anonymization

The *Position Level* was mapped to a scale of positions from 1 to 17. Position levels from 1 to 9 are permanent staff positions, out of which, levels from 1 to 7 reflect Operational and Non-Executive positions, levels 8 and 9 correspond to Executive positions. Position levels above 10 are temporary positions, matching the same level of seniority than staff positions.

The *Organization* column was also anonymized under the names *Org 1* and *Org 2*.

Missing Value Handling

Several rows in the dataset didn’t contain any values for the *Time to Fill* column, given that this column was used in order to predict if a position would take a longer time than the average to be filled, values could not be imputed. This reduced the dataset for model training to 5,476 rows.

Standardization

Another step to ensure the quality of the data and its confidentiality was to perform a Z-Score Normalization. This step was also crucial for visualizing multiple features in the Exploratory Data Analysis and Cluster Analysis steps.

Outlier Detection and Treatment

The *Time to Fill* and *Total Applications* columns in the dataset showed abnormal values. These values were capped to the 95th percentile.

4.2.2 Text Cleaning and Preprocessing

There were several terms that had to be removed to ensure the confidentiality of the data in terms of the organizations in which the work was performed and their location.

The relevant section for each document was extracted, since these contained the main qualifications required to perform at the position by the candidate (Figure 1). This text data was saved into "*.txt*" files.

These files were consolidated into a single data frame with all of their corresponding information from the structured data table, such as the position level, organization, Time to Fill and applications received. The text data was stored in a column of this data frame and multiple methods were applied in order to extract the relevant terms.

The text data in this column was preprocessed by lowercasing, tokenizing and lemmatizing the terms and an additional column was created to store the list of tokens for each document. The column containing the text data was used to generate the embeddings using a Large Language Model with the *SentenceTransformers* framework.

4.2.3 Embeddings Generation

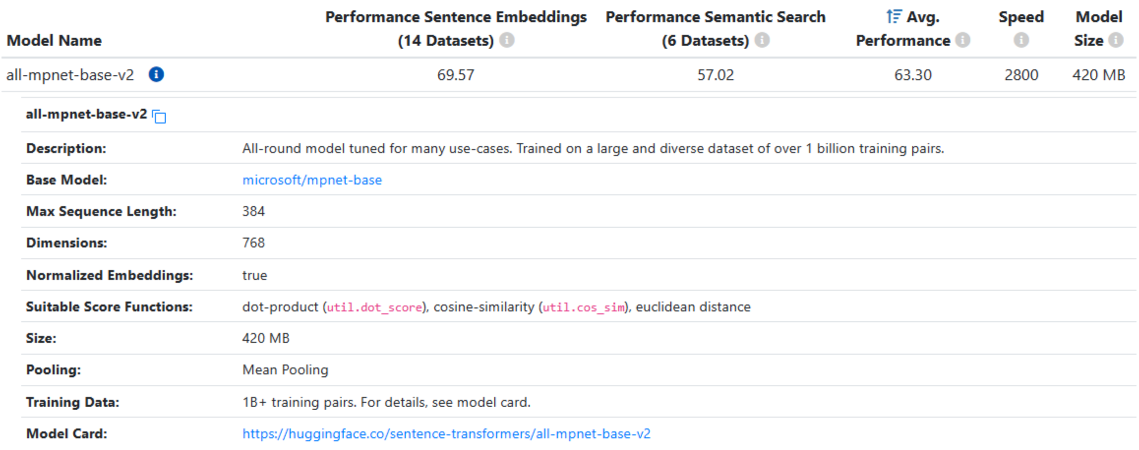
The cornerstone of the variables used for the modeling tasks of this work were the document embeddings.

A Python framework for generating embeddings using Large Language Models (for the scope of this work, a Small Language Model was used, see section 3.2.4 for reference) was used. After testing multiple pre-trained Language Models to assess the quality of the embeddings, a model based on the MPNet model architecture was used.

The MPNet model was introduced by Microsoft to overcome some of the BERT model limitations when it comes to training and is able to provide higher quality embeddings.

To generate the document embeddings used for this work, the *SentenceTransformers* framework was used. This framework acts as a wrapper to easily handle Open Source Language Models and generate sentence and document embeddings.

The Open Source pre-trained Language Model used in this work (Figure 3) contained the following description:



Model Name	Performance Sentence Embeddings (14 Datasets)	Performance Semantic Search (6 Datasets)	Avg. Performance	Speed	Model Size
all-mpnet-base-v2	69.57	57.02	63.30	2800	420 MB

all-mpnet-base-v2	
Description:	All-round model tuned for many use-cases. Trained on a large and diverse dataset of over 1 billion training pairs.
Base Model:	microsoft/mpnet-base
Max Sequence Length:	384
Dimensions:	768
Normalized Embeddings:	true
Suitable Score Functions:	dot-product (`util.dot_score`), cosine-similarity (`util.cos_sim`), euclidean distance
Size:	420 MB
Pooling:	Mean Pooling
Training Data:	1B+ training pairs. For details, see model card.
Model Card:	https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Figure 3. *Technical Specifications of the used Language Model.*

The main rationale for selecting this model is due to its high performance and quality of embeddings. Several other models were used to perform the encoding of the text data, however, after evaluating the performance and key metrics on the prediction and job profile clustering stages, the results yielded by this model were superior, also, according to the documentation for the *Sentence Transformers* Framework, this model yields the best quality embeddings. See Appendix 2 for more details in terms of the evaluation of the quality of embeddings.

The output of this model consists of a vector of 768 elements, which required additional processing by using a dimensionality reduction model. The model training process is explained in the ***Time to Fill Prediction*** chapter.

4.2.4 Dimensionality Reduction (PCA)

In order to utilize the embeddings generated by the LLM, a dimensionality reduction model was used to decrease the number of features of its output from

768 elements to a more manageable number of features. The selected model for this task was the Principal Component Analysis (PCA).

The following figure (Figure 4) shows the accumulated percentage of information retained by the variables condensed in the model. A total of 95 features were used to train the K-Means model used in this work, since these 95 features contained 90% of the information in the matrix of embeddings generated by the Large Language Model. This allowed for both reducing the computation time of the K-Means model training, along with reducing the complexity of the XGBoost model trained for predicting delays in Time to Fill.

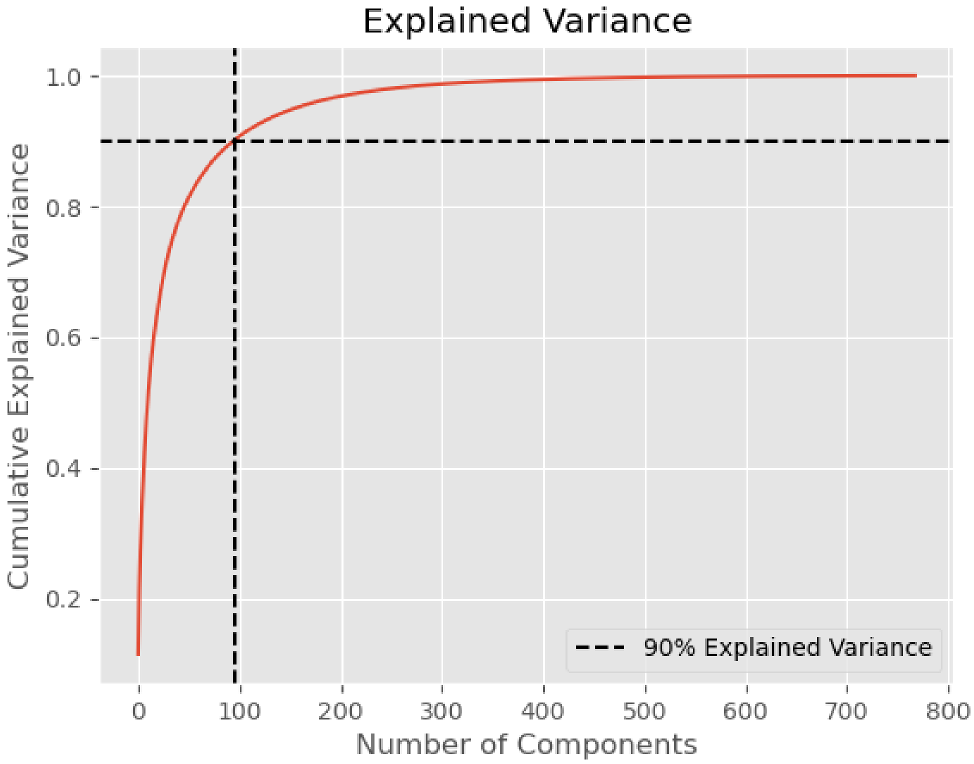


Figure 4. *Explained Variance*

4.2.5 Feature Extraction

After reducing the dimensionality of the text data, several tests were performed in order to determine the number of variables to use to train the predictive model.

Feature Selection:

The following features were used in order to train the model, they contain information related to the position.

- cluster_number
- tokens_len
- internal_applications
- external_applications
- position_level

The “*cluster_number”* feature refers to the cluster prediction from the K-Means model, this process is explained in detail in the section *4.4 Job Profile Clustering*.

In terms of the job description data extracted from the documents, the first 29 components of the PCA model were used as well. Adding up to a total of 34 features.

4.3 Time to Fill Prediction

The decision of lowering the Time to Fill below the current average was made by these organizations to address one of their key business priorities, which consists of reducing the variance across the Time to Fill of their positions, along with identifying positions that might have delays given the required skills for the job.

Initially, several models were used to accomplish this task, such as Decision Trees, Random Forest algorithms, and Logistic Regression under both an embedding based approach and a keyword-based approach using a TF-IDF model for the latter, for details in the results of these tasks, see appendix 3.

Although models had a similar performance in terms of prediction, one of the key factors to consider in the development and implementation of the final model within the organization's business processes is the ability to interpret its outputs.

When testing the TF-IDF based approach, the keyword features were not as intuitive as expected when interpreting the most important features of the model, since the n-grams itself would lack the context in which the other requirements of the job description were defined. For example, some of the top features of the Logistic Regression model were terms like "sector" or "instrument" which doesn't necessarily provide enough context by themselves.

Given the difficulties of interpreting the model's output, valuing the predictive capability of the model seemed like a more practical approach, therefore, using the Language Model generated embeddings for retaining the semantic meaning and context of the terms in the job description would enable for more accurate predictions, as well as a better handling of out-of-vocabulary terms. However, in order to add interpretability to the model's output, an unsupervised learning model was used to cluster the embeddings and identify some common keywords and terms within each cluster, these clusters will allow for slicing the data by different variables, such as the country, region and level of the position. This process is described in section 4.4.

The predictive model that showed the highest performance was XGBoost, however, the other evaluated models showed similar scores, the process of training and fine-tuning the model is described in the following section.

4.3.1 Model Training - XGBoost

During the training process of this model, a matrix of 5,476 rows and 34 columns (X) was used. The training set contained 80% of the rows.

The predicted variable (y) consisted of a vector of 5,476 elements, with the class labels $\{0,1\}$. The class "*0*" was assigned to values with a Time to Fill below the average Time to Fill for all positions, which is the metric defined by these organizations as a target. The class "*1*" was assigned to days with a higher Time to Fill than the one set by the organization. For example, if the position is prone to delaying by more days than the average of all positions, the model would yield a probability higher or equal to 0.5, which translates into class "1", if the position would take less than the average Time to Fill, the model will predict a chance lower than 0.5, assigning the class "0".

The train and test set split also considered the proportion of class labels, since approximately 44% of the labels pertained to the class 1, for which the *stratify* parameter was set for the predicted variable (y) and a class weight hyperparameter to address the class imbalance.

The split for both train and test datasets were 80 to 20 percent respectively.

The features used to train the model were:

- Internal Applications
- External Applications
- Token length of the requirements in the job description
- Level of the position
- PCA components from 0 to 30

The following were the initial hyperparameters set for the XGBoost model in the initial training iteration:

Hyperparameter	Values
learning_rate	1
n_estimators	100
max_depth	10

min_child_weight	1
scale_pos_weight	{0: 0.8859, 1: 1.1478}
eval_metric	'auc'

Table 4. Initial Hyperparameters

The remaining hyperparameters were left under their default setting. The main reason for setting the evaluation metric to "*auc*" is to ensure that there will be enough flexibility when implementing the model, so the business strategy can be set to prioritize either precision or recall as required and tune the prediction threshold accordingly.

The initial results with these parameters yielded the following results (Table 5):

Metric	Score
Accuracy	0.662
F1	0.659
Precision	0.659
Recall	0.662
AUC	0.696

Table 5. Initial Model Results for XGBoost

These initial results were higher than the ones yielded by other models used to solve this prediction problem (Decision Trees, Random Forest and Logistic Regression) on every metric by at least 0.03 points.

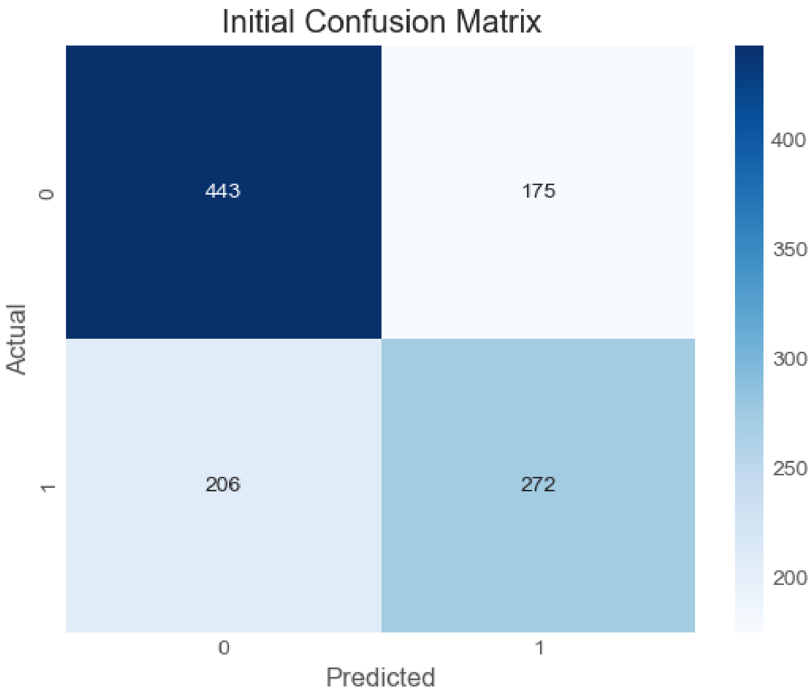


Figure 5. Initial Confusion Matrix

4.3.2 Hyperparameter Optimization

To fine-tune the model hyperparameters, a parameter grid was built with the following values:

Hyperparameter	Values
learning_rate	[0.01, 0.05, 0.1, 0.2]
n_estimators	[50, 100, 200, 300]
max_depth	[5, 10, 15, 20]
subsample	[0.1 ,0.3, 0.5, 0.7, 0.9, 1]
gamma	[0, 1, 5]
colsample_bytree	[0.2, 0.4, 0.6, 0.8, 1]
colsample_bylevel	[None, 0.2, 0.4, 0.6, 0.8, 1]
min_child_weight	[1, 5, 10]

Table 6. Parameter Optimization Grid

Subsequently, the Randomized Search Cross Validation (*RandomizedSearchCV*) implemented in the *sklearn* framework was used to find the best hyperparameters for the model. The search consisted of 50 iterations under 5 folds, which resulted in 250 fits.

The following are the model results for the optimized hyperparameters:

Metric	Score
Accuracy	0.673
F1	0.673
Precision	0.674
Recall	0.673
AUC	0.720

Table 7. Optimized Model Results

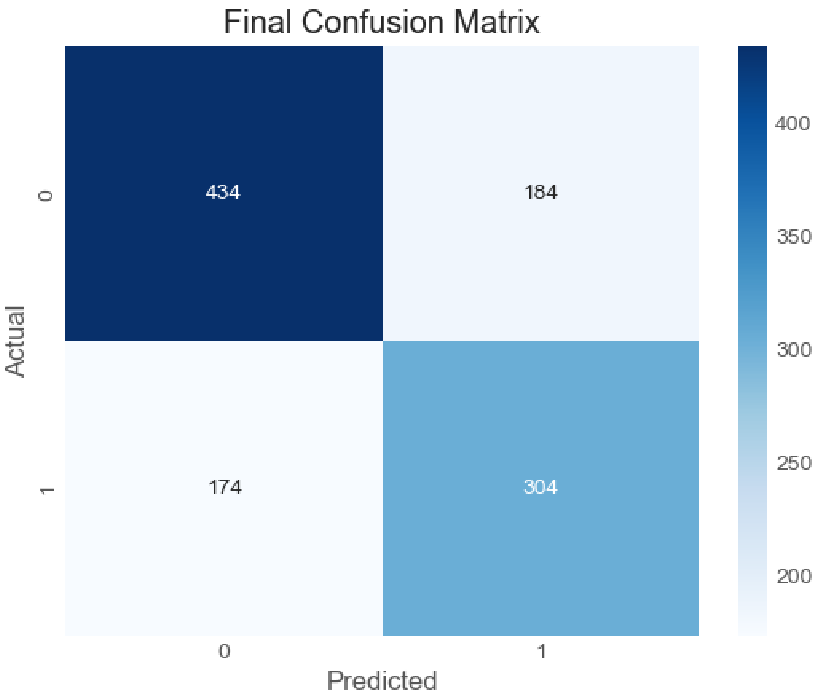


Figure 6. Optimized Model Confusion Matrix

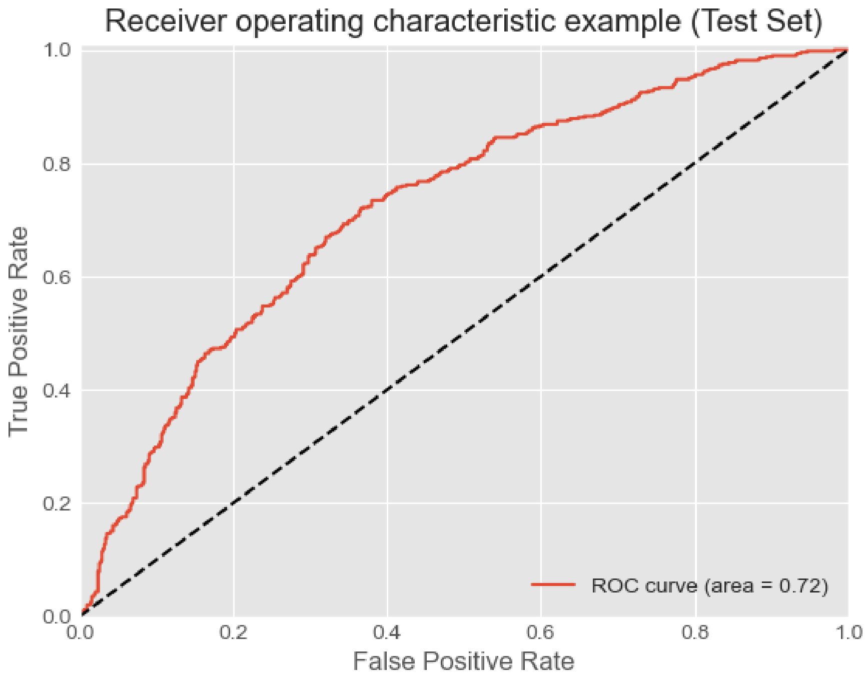


Figure 7. Optimized Model ROC Curve

The optimized hyperparameters led to an increase of approximately 3% in every metric. It is important to consider two factors into the development of this model given the small increase in performance after the hyperparameter optimization:

- Additional variables should be added in order to continue to improve the performance of the model that are not necessarily related to textual data. Additional components from the PCA model were added in order evaluate if the performance of the model would improve with more features, however, it didn´t yield any increase in the evaluation metrics. Given the confidential nature of the data and the limited amount of information that can be used for this work, these were the best obtained results. Part of the additional valuable information for the model is related to the region and country for which the position is being recruited or information related to the salary and compensation conditions for the position.
- Another challenge in terms of predicting the delays in Time to Fill are related to information that is not necessarily recorded in the applicant tracking systems of these organizations, such as changes in the business needs in the middle of the recruitment process, which might require adding

more skill skills or changing compensation/location conditions, which could translate into sourcing new candidates, or leaving the position on hold.

A threshold optimization was also performed in order to find the best possible balance between Precision and Recall. However, this value turned out to be approximately at 0.5 as shown in the following figure:

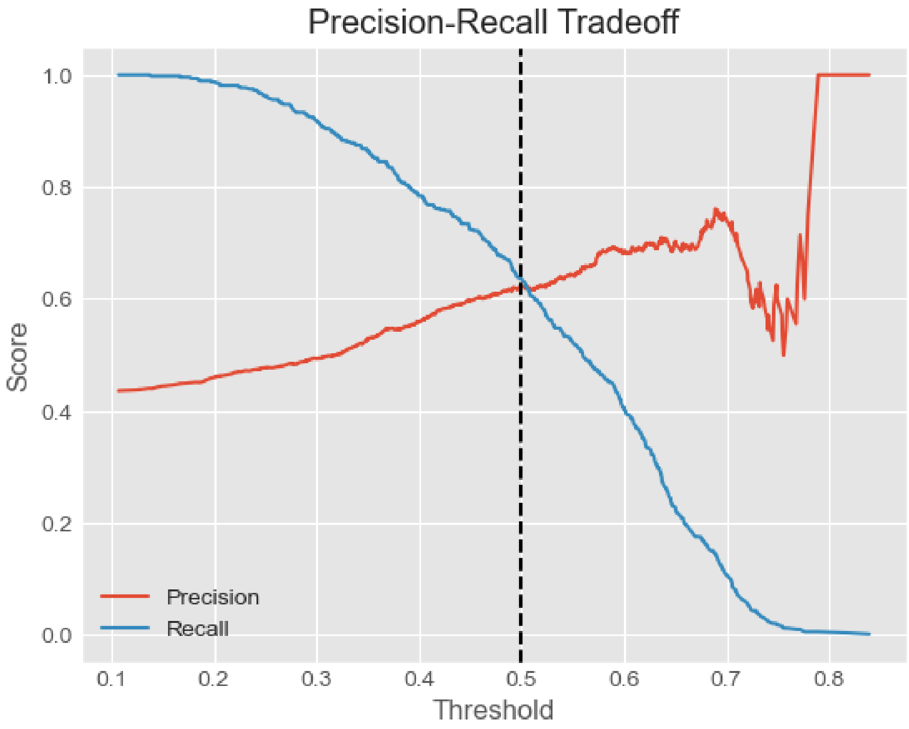


Figure 9. Precision-Recall Tradeoff chart

In terms of the feature importance, the following figure shows the ranking of features that add the most information to the model. The features named from 0 to 29 refer to the features taken from the PCA model.

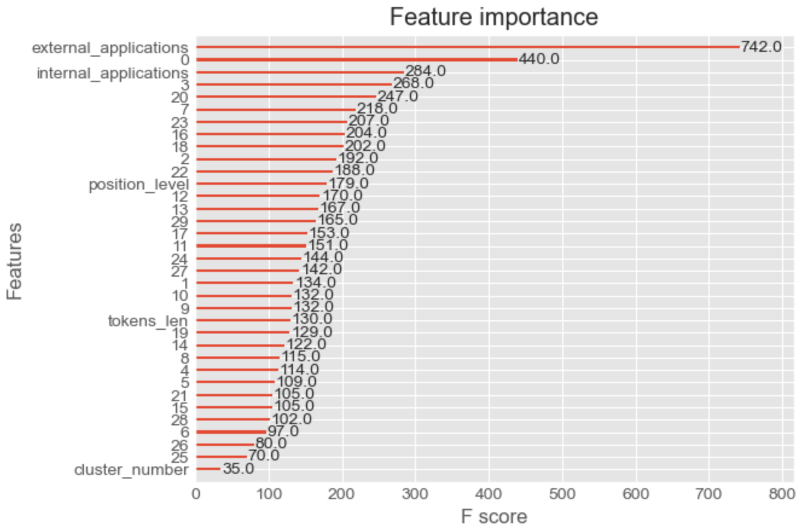


Figure 10. Feature Importance for the XGBoost model

Although the number of external applications add the most information to the model, it is worth noticing that the job description text data (from the components of the resulting PCA matrix) make important contributions to the model in terms relevant information and enhance its ability to predict Time to Fill delays.

In order to illustrate this point, another model was trained without the usage of the text data and the results were quite inferior, with an average of 10 percentual points less in every metric, including a 0.62 AUC score.

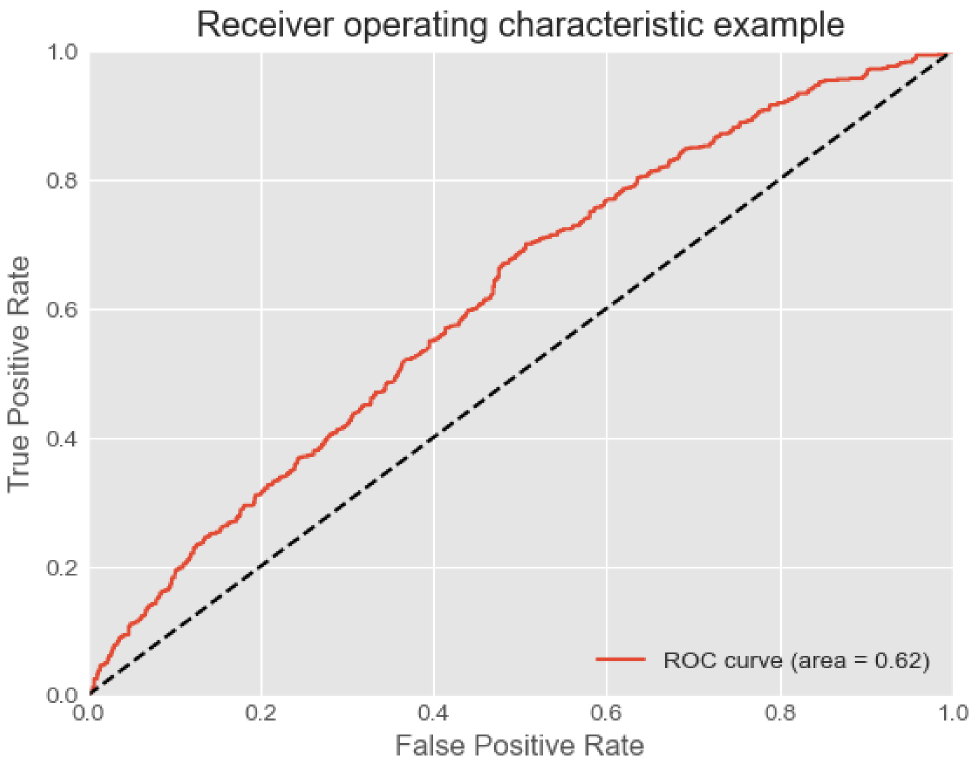


Figure 11. ROC curve for model without text data

Even though the model results were satisfactory, it still lacks interpretability in terms of the behavior of the variables that impact the Time to Fill a position, therefore, an analysis was performed for every cluster to better identify potential delays in different positions and provide recommendations.

4.4 Job Profile Clustering

4.4.1 Job Description Clustering

Given the objective of this work in terms of providing interpretability to the supervised machine learning model prediction of Time to Fill, as well as recommendations in terms of optimizing recruitment and candidate sourcing strategies, a job profile clustering was performed that would allow for understanding similar terms within the studied job descriptions.

After processing the LLM-Generated embeddings and reducing their dimensions, these variables were ready to be used in the two main models used in this work to solve the problems mentioned in the first chapter. Multiple unsupervised learning algorithms for clustering such as HDBSCAN and K-Means were used, however, the K-Means model yielded higher quality results with greater interpretability.

The K-Means model was used to group the job requirements data into clusters, this allowed for clustering job profiles based on similar collections of skills and educational requirements to perform the role.

K-Means is a powerful tool for exploratory data analysis and clustering tasks. However, one of the main challenges when it comes to using this algorithm is that the number of clusters K is defined by the user, which means that multiple values should be used to obtain an optimal number of clusters.

Given that this work required not only a robust predictive modeling of the Time to Fill delays, but also the ability to interpret the job profiles, there were three key metrics used in order to define the number of clusters.

These three metrics (Figure 12) were used to assess an optimal, yet interpretable and manageable number of clusters. The results of the number of clusters according to each method are shown in the following figure:

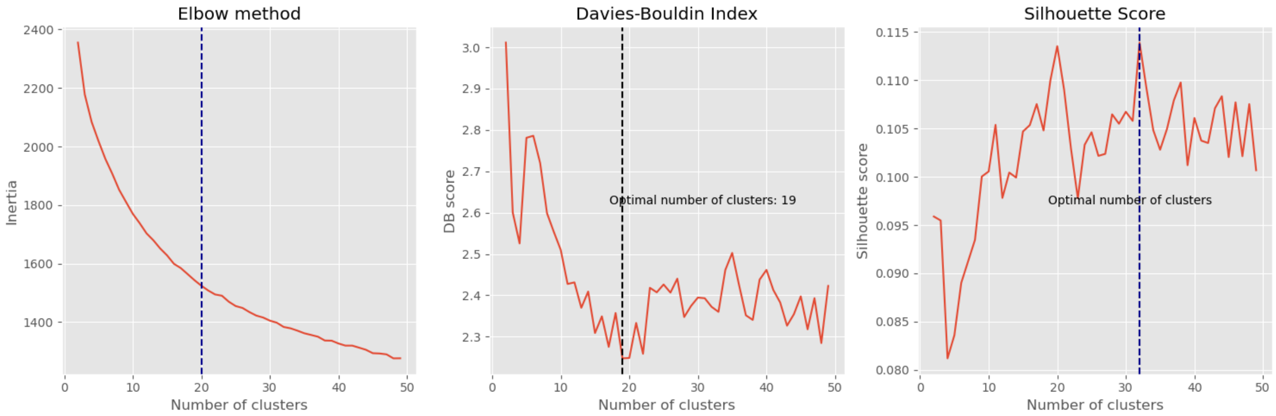


Figure 12. *K-Means Evaluation Metrics*

The Elbow method chart suggested between 15 to 20 clusters, the DBI chart suggested either 19 or 20 clusters and the Silhouette Score chart suggested an optimal number of 32 clusters. However, the second-best number of clusters for the latter was set as 20 clusters. Given these results, 20 clusters were selected as the optimal number of clusters.

A key step to investigating why the silhouette score was low was to analyze each cluster's score, this showed that there were several clusters with higher scores and showed a more specific domain knowledge, while there were other profiles that tended to require a more generic set of knowledge and skills. This indicated why they were overlapping with other clusters.

4.4.2 Clusters Analysis

The TF-IDF model was used in multiple instances of this work, the first one being on a general level to initially explore the most relevant terms of the corpus, and the second one being a more qualitative analysis of the profiles extracted from the K-Means model to interpret the keywords present in every cluster.

A qualitative analysis was performed using a TF-IDF method to interpret the most common and relevant terms within the clusters along with some frequent job titles.

This was achieved by selecting the documents corresponding to each cluster and applying the TF-IDF vectorizer to extract the most relevant monograms and bi-grams within that collection of documents. This allowed for assigning a name for the cluster based on the skills required for the role. The job titles for the positions within that cluster were also consulted, along with a manual review of ten random documents.

For example, one of the clusters showed several relevant terms such as: “quantitative”, “CFA”, “SQL”, “data”, it had position titles with terms such as: “Data Analyst”, “Quantitative Analyst”, “Econometrics”. After reviewing ten random samples of the documents within this cluster and leveraging on the existing knowledge of the organizations in which the work was carried out, the “Quantitative Analysts” label for the cluster seemed appropriate. This process was repeated for all of the clusters.

There were three cases in which the cluster´s keywords included collections of generic skills, such as working proficiency of the MS Office package and written communications (clusters 8, 2 and 10). These skills tend to be related to administrative and program management positions. After reviewing the document samples for each of these three clusters, these were grouped into the “Program Management and Administrative Support” label (see appendix 1).

The following (Table 2) is the list of clusters after performing the described analysis.

Job Profile	Cluster Number
Energy and Sustainability	9
HR Professionals	13
Procurement and Acquisitions	18
Communications and External Affairs Specialists	19
Legal Professionals	1
Corporate Finance and Resource Management	14
Program Management and Administrative Support	8, 2 and 10
Healthcare and Public Health	11
Transport and Urban Development Specialists	4
Quantitative Analysts	6

Drivers and Security Professionals	0
Econometrics and Statistics Professionals	3
Environment, Climate Change and Sustainability Professionals	16
IT Professionals	15
Core Operations and Program Management	7
Investment and Advisory	5
Social Development	17
Senior Investment and Advisory	12

Table 2. Job Profile Labels and Clusters

5. Results

The following section consists of the results of the corresponding Exploratory Data Analysis and analysis of clusters, job profile demand understanding and recommendations.

5.1 Key Variables Influencing Time to Fill

The two key variables to consider when analyzing the results are the behavior of Time to Fill and Applications Received. The following figure shows the Cumulative Distribution Function for both variables. For the Time to Fill, approximately 58% of the positions sit at an average number of days or less, as shown in Figure 13, which means that an important part of positions tends to get delayed. For applications received, this number goes up to approximately 67%.

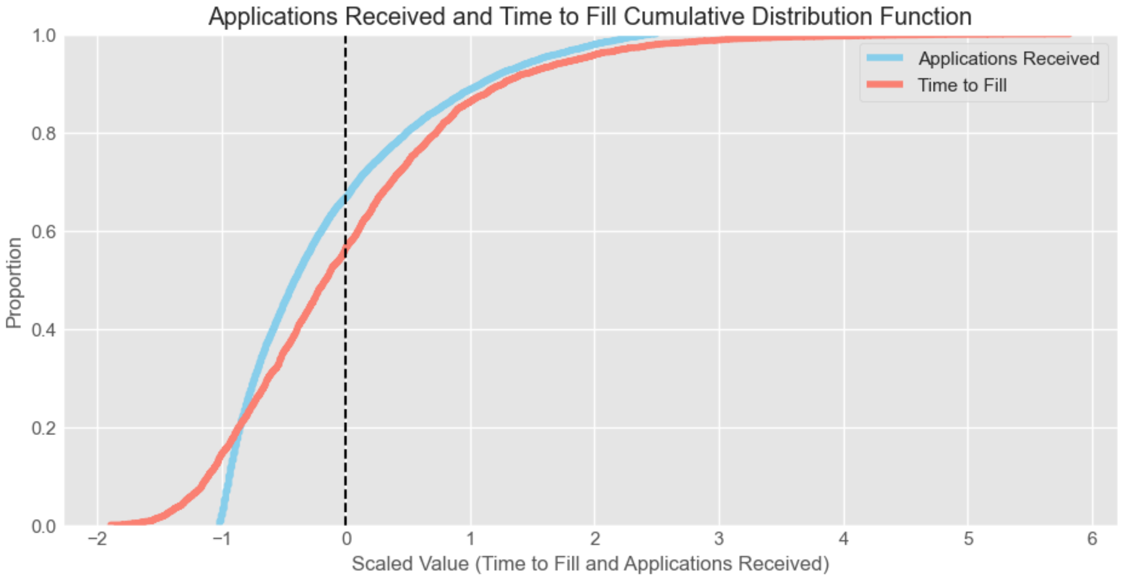


Figure 13. Distribution of Positions by Time to Fill

Another key variable related to the Time to Fill lies in the type of position. The following figure shows the distribution of Time to Fill for internal and external positions. It is noticeable that internal positions tend to be filled in a quicker manner.

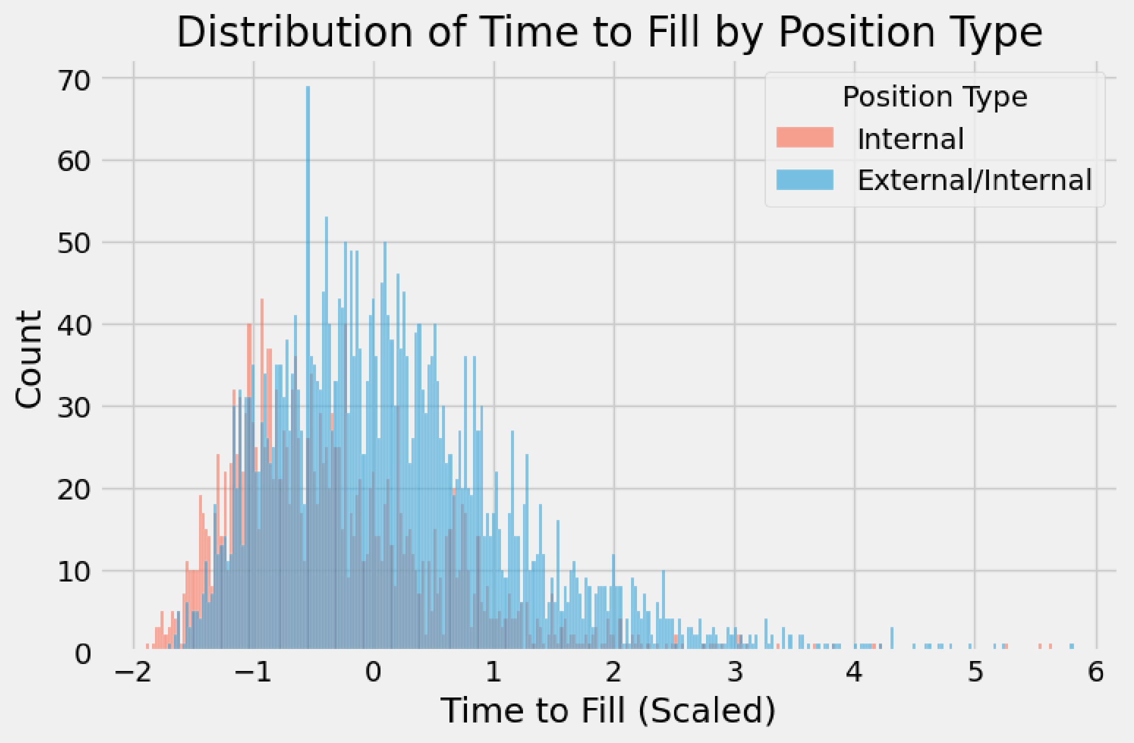


Figure 14. Distribution of Positions by Time to Fill

When viewing the same chart for the Number of Applications, it is noticeable that the applications for Internal positions also tend to be much lower. However, it is important to note that the distribution of applications for External positions tends to be skewed to a lower number of applications and have a high variance, which could help explain the Time to Fill variance as well.

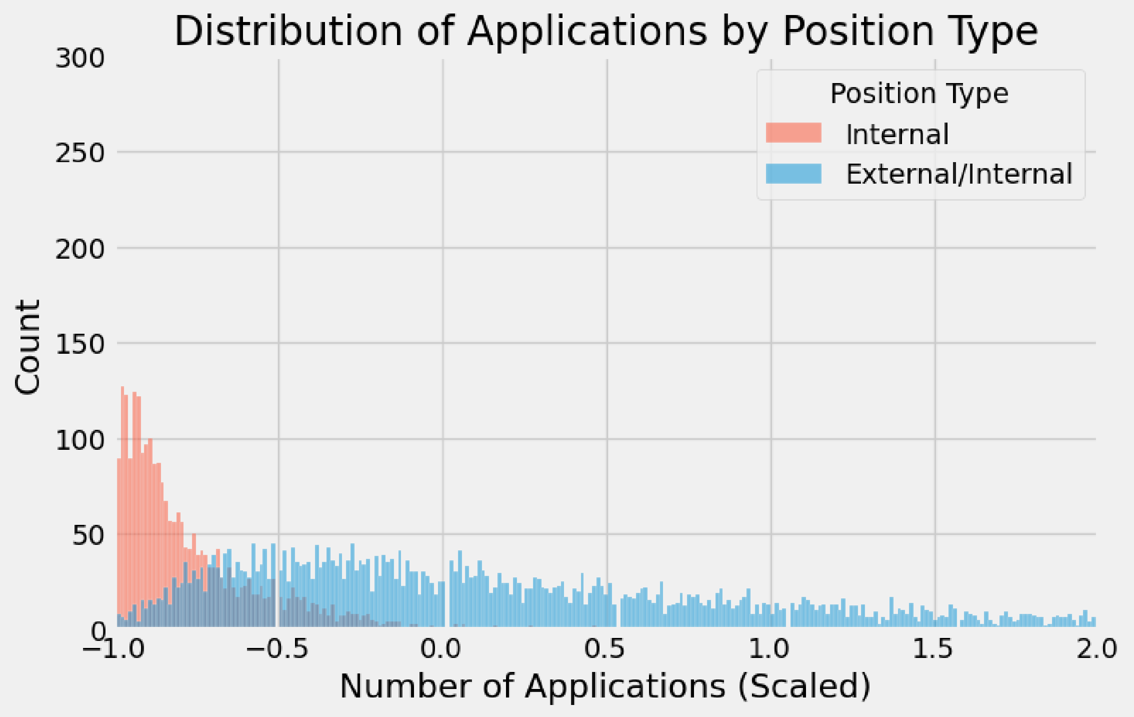


Figure 15. Distribution of Positions by Applications Submitted

When analyzing the relationship between the Time to Fill and Number of Applications, these variables show a correlation of 0.29, after removing outliers, a linear regression was performed between the two variables and it was found that the relationship is significant, with a p-value of 2e-92.

These two variables have a heteroskedastic relationship, and although it is important to notice that the number of applications received could represent a high operational load, given the effort that takes to screen and review applications, the existing business knowledge in terms of recruitment processes suggests that there could be cases in which the position is reposted, since in the first posting of the position no suitable candidates were found, which leads to a second instance of receiving and reviewing application and repeating process instances such as scheduling the interviews for the new batch of candidates.

Further analysis should be undertaken with additional data to better understand this relationship and identify potential confounders.

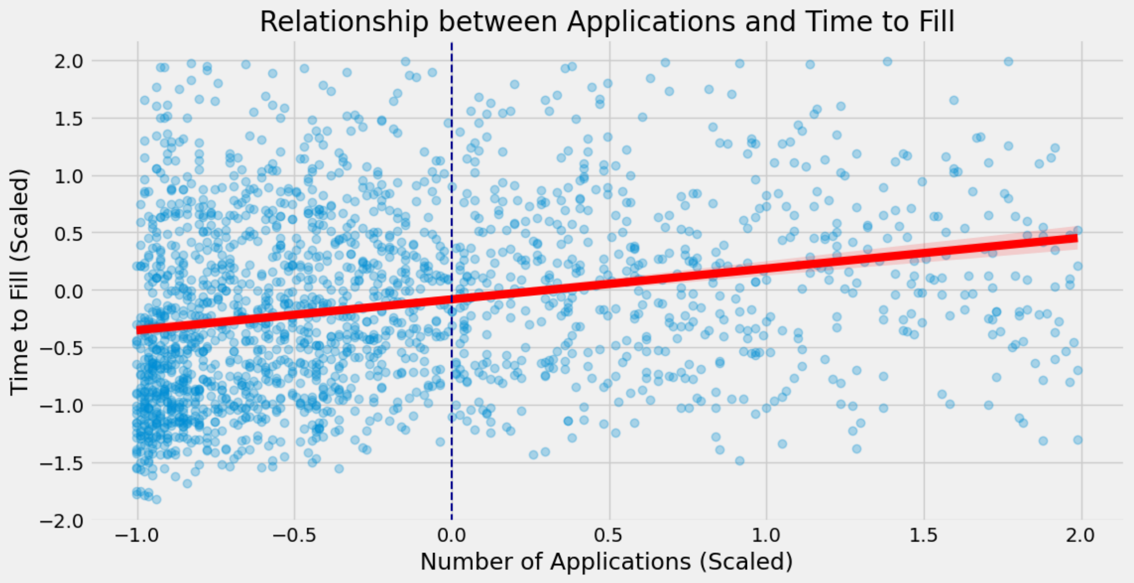


Figure 16. Time to Fill vs Applications

Another Variable that should be looked at is the distribution of position levels (Figure 17). Levels above 10 are temporary positions, while positions from level 1 to level 9 are permanent positions.

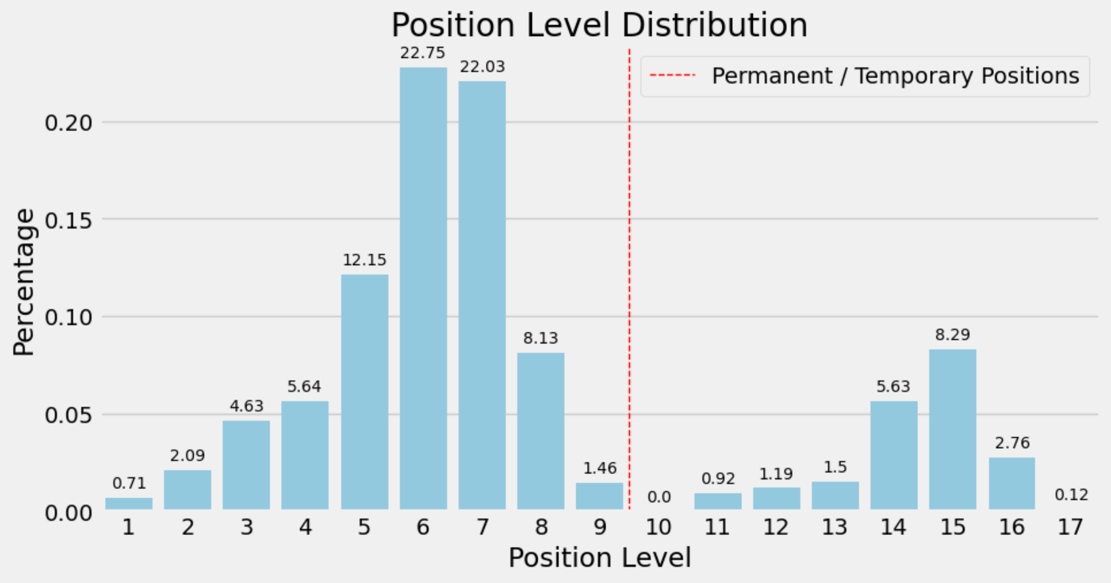


Figure 17. Position Level Distribution

It is noticeable that Temporary positions constitute approximately 20% of all the posted positions. However, the difference between these types of positions in terms of our variable of interest is significant.

In the following chart (Figure 18), the average number of applicants vs the Time to Fill for every position level is shown.

Position levels from 1 to 4 are equivalent to operational, administrative and supporting positions within the organizations studied in this work. Levels 5 to 7 are mid-level positions such as analysts and specialists, with 8 and 9 positions being equivalent to managers or high seniority positions.

Positions levels from 10 and above mirror the same structure: 10 to 13 for administrative positions, 14 to 16 for mid-level positions and 17 for higher seniority positions, however, these are set as temporary positions.

Positions corresponding to levels 3 and 4 tend to have a shorter Time to Fill and receive a number of applications close to the average. However, level 1 and 2 positions, as well as level 5+ positions tend to have a lengthier timespan, with level 8 and 9 positions being the hardest ones to find and a have low number of applicants. These positions demand more senior professionals, as they are managerial positions.

For Temporary positions, the Time to Fill tends to sit below the average in most of the cases, they also receive the least number of applications.

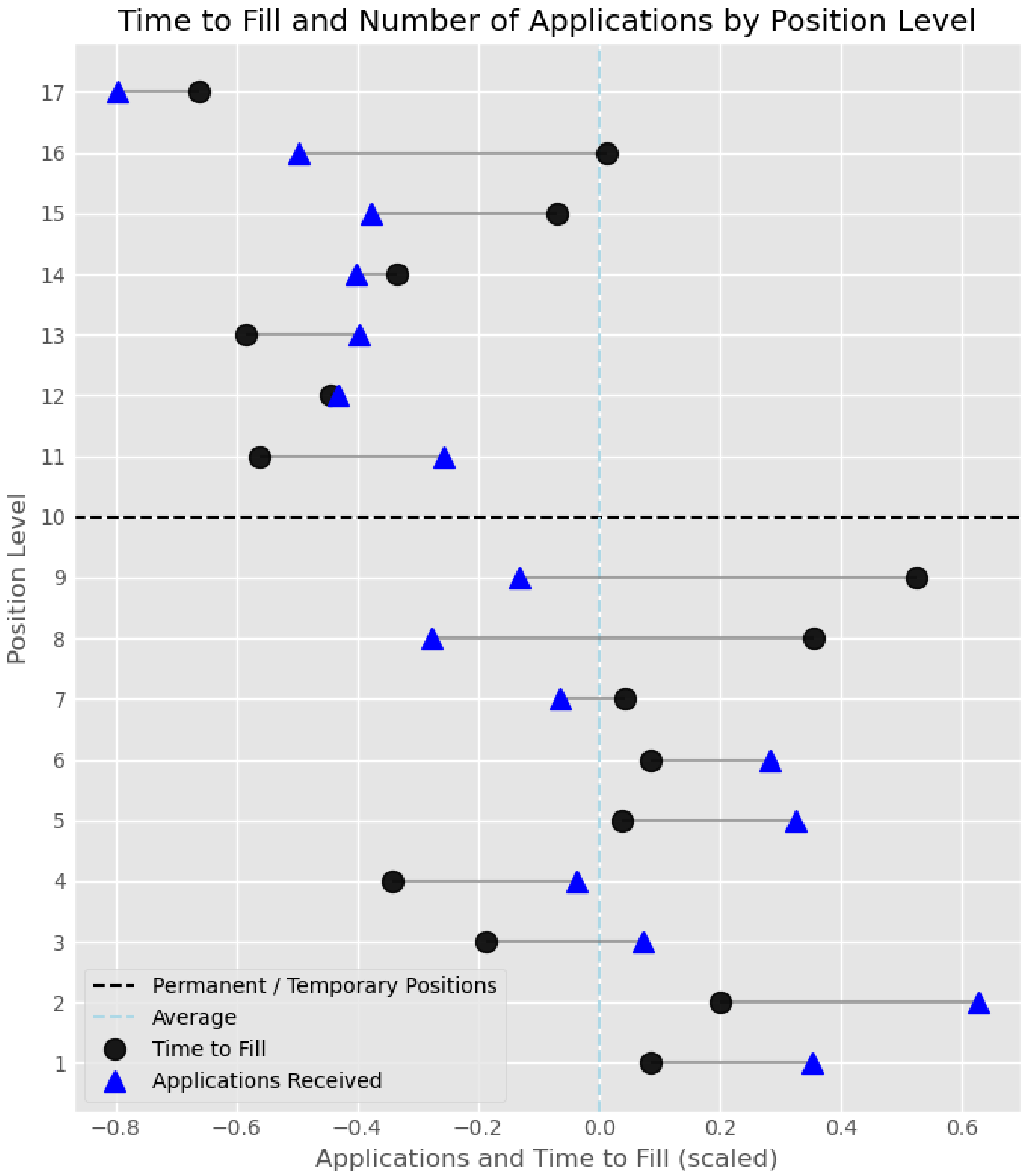


Figure 18. Time to Fill and Applications by Position Level

5.2 Job Profile Analysis

The following chart (Figure 19) shows the clusters with the highest demand withing the organizations. These are Program Management and Administrative Support positions, Investment and Advisory and Core Business Operations.

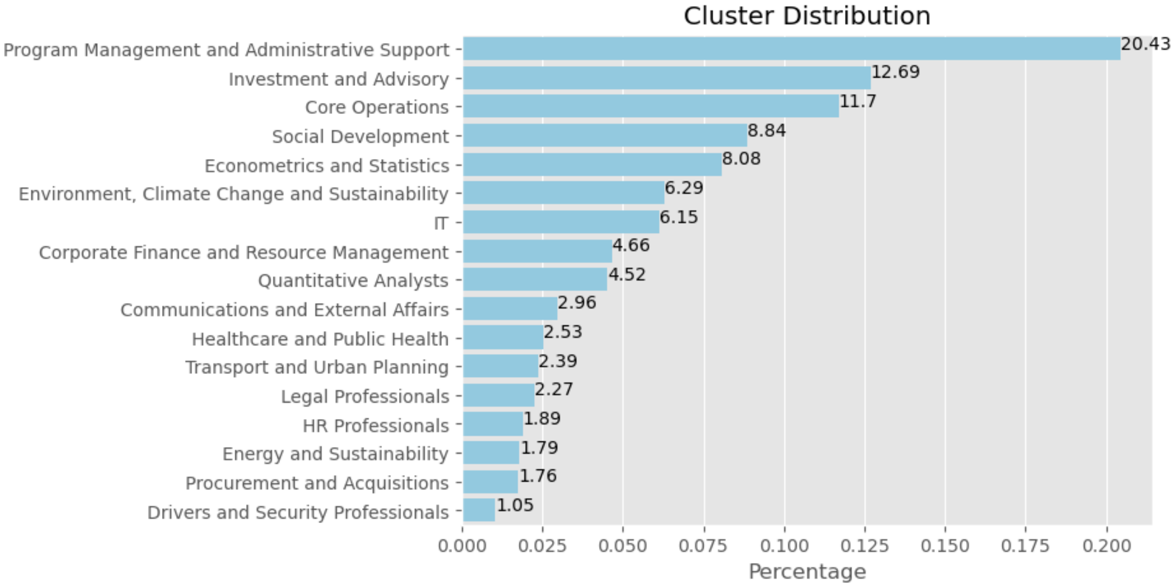


Figure 19. Distribution of Job Profiles

It is important to view the demand proportion of these clusters by grade, since this could indicate that a more specialized skillset or experience could be required for the position.

As shown in the next figure (Figure 20), the levels for the most in-demand clusters tend to be evenly distributed, as is shown in the figure below. The highest concentration of positions by level are amongst levels 5, 6 and 7. For temporary positions, most of the demand sits at the levels 14 and 15. It is also important to notice that there are some outstanding cells in the figure that show a demand of more than 40%. Which are level 1 Drivers and Security Professionals, level 7 Energy and Sustainability professionals, level 5 Quantitative Analysts and level 7 Transport and Urban Planning Professionals.

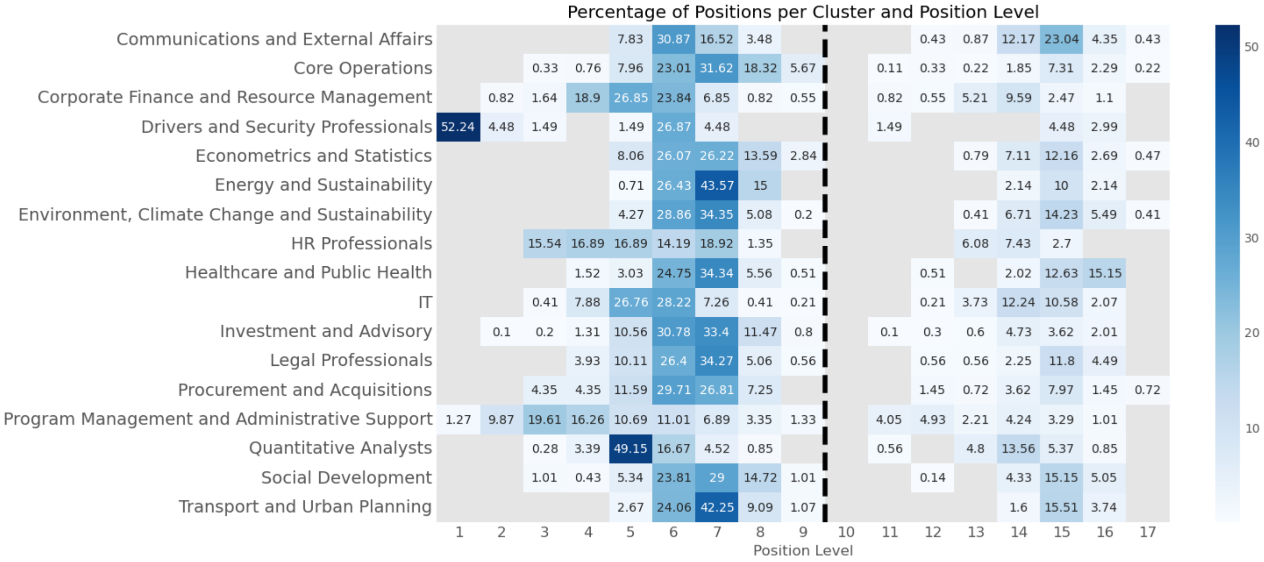


Figure 20. Distribution of Clusters by Grade

The same chart was analyzed but using the scaled Time to Fill (Figure 21). However, in order to account for the most relevant positions, only the clusters and grade levels that hired at least 30 positions are shown. The following figure shows that Core Operations professionals at level 9 tend to have a mean Time to Fill of 0.71 standard deviations. Overall, positions above level 6 tend to have a higher than average Time to Fill for staff positions. Time to Fill tends to be much lower for consultant and temporary positions.

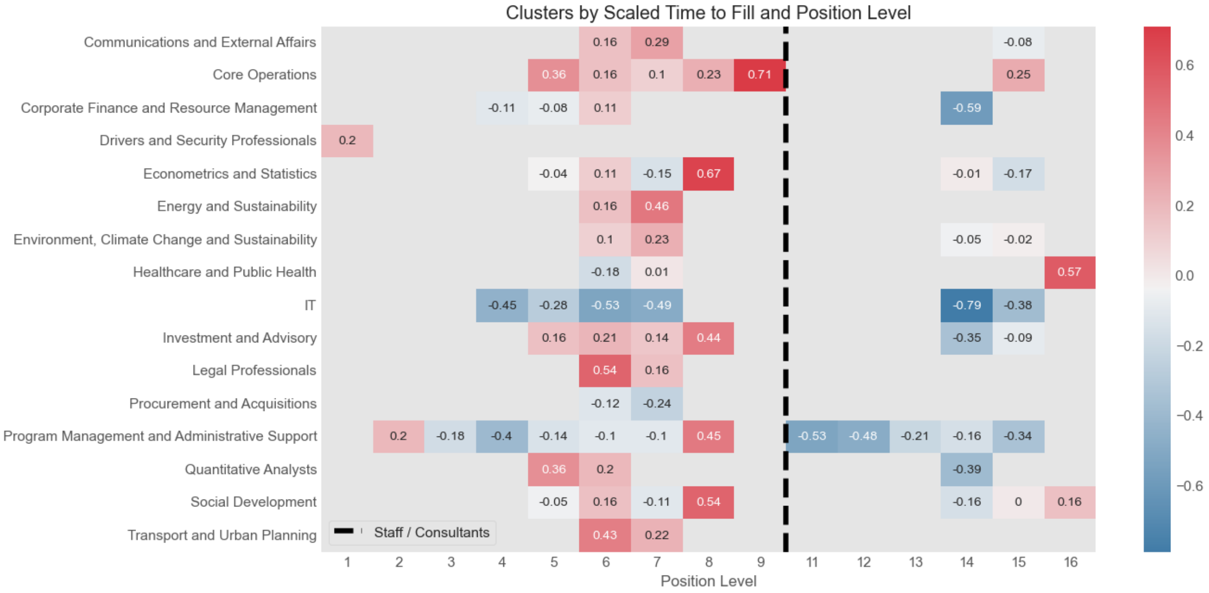


Figure 21. Distribution of Clusters by Grade (Time to Fill)

To have a visual representation of the highest Time to Fill by cluster and the applications received, a ridgeline plot was constructed. The following visualization shows the distribution for every cluster along with the Overall and Cluster Average Time to Fill. Only clusters that account for at least 2% of all positions were taken into consideration, as well as permanent (levels 1 to 9) positions, which tend to represent the highest proportion of positions.

It is noticeable that the Transport and Urban Planning cluster has the highest average Time to Fill and tends to receive the least number of applications. It is also worth noting that for the Quantitative Analysts, Legal Professionals, Environment, Climate Change and Sustainability and Investment and Advisory clusters the distribution of applications received tends to be flatter and showing a higher variance.

The first five clusters in the following figure account for 28.92 percent of all positions within this segment.

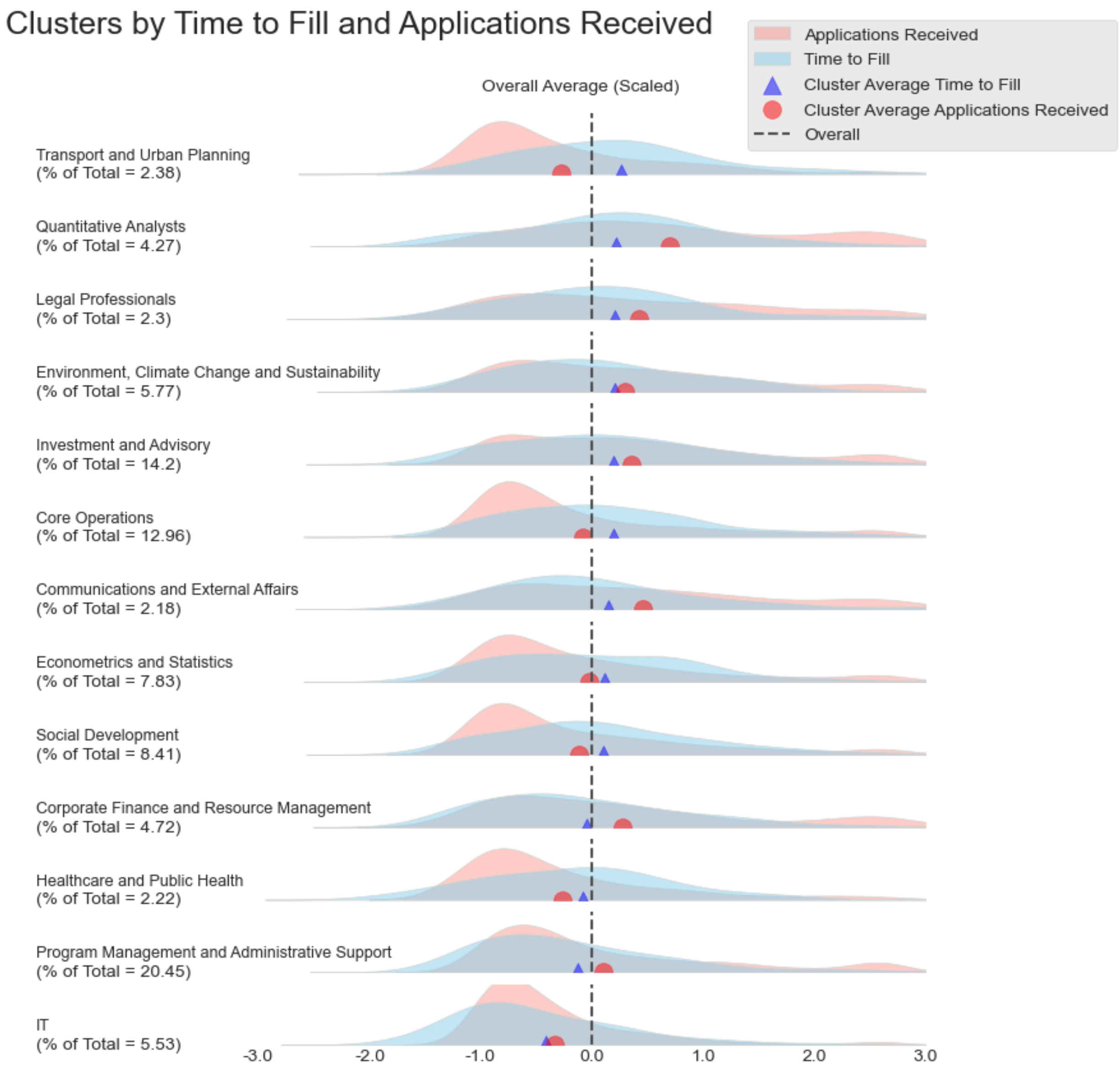


Figure 22. Distribution of Clusters by Time to Fill and Applications Received

6. Conclusions and Recommendations

An important part of this work was to understand the current demand of job profiles within the studied organizations to provide recommendations for enhancing the Time to Fill.

Given the confidential nature of the data used to carry out the current work, some additional variables for every position should be added into the analysis to help explain the behavior of the Time to Fill variable, these variables could include the time in which the position was posted, the number of concurrently open positions by the time the position was posted, the tenure of the employee involved in filling the position, the department, etc. However, the creation of a job profile map to better understand positions as collections of skills was a key step to achieve the first objective of this work. This map, when used as a variable within the analyzed dataset, provided key insights in terms of the Time to Fill, along with the number of applications received.

One of the results shown in the comparison of Time to Fill vs Applications Received by cluster indicated how certain types of positions might have a challenging time being sourced due to the high variance of applications received. Which suggests two critical issues to be addressed. One being the difficulty of finding candidates due to the small number of applicants, and the other one being the high operational load it requires to review the CVs and perform the corresponding longlisting.

There are multiple strategies that can be used to enhance the Time to Fill by addressing the mentioned situations, one being to have a pool of candidates that were not hired for previously closed positions. This could help on shortening the time it takes to source additional candidates, as well as relying on referral programs.

In terms of reducing the operational load when a high number of applications are received, a feasible solution would be to either develop a system that allows for matching CVs to job descriptions, initially using Cosine Similarity to establish baseline results and then move towards an embeddings-based approach using LLMs. It could also be worth exploring solutions from external vendors.

There could be multiple other factors that might interfere with the recruitment and selection process that are not captured by the data used in this study, for example: internal policies or shifts in business needs. However, in terms of detecting positions that can be prone to delays, it is recommended to implement the model trained for this study and set its threshold to enhance precision, given that almost 30 percent of the positions with the lengthiest Time to Fill are in the top 5 clusters.

It is worth noting that allocating additional human resources for the handling of high-risk positions could also yield better results, either for assisting in the review of applications or administrative support in the scheduling of interviews.

As a recommended subsequent step, it is recommended to further explore methods to produce a more granular job profile map along with key skills associated with them, along with adding more features that can bring more contextual information to the model, such as the compensation conditions for the position, country/region and the tenure of the recruiter in charge of filling the position. This will help in better understanding key skills that are difficult to find in the market, as well as providing more useful information for the model. It would also be useful in terms of defining strategies for upskilling the current workforce and specialize in specific domains.

References

- Anscombe, F. (1973). Graphs in Statistical Analysis. The American Statistician, 195-199 .
- Anscombe, F., & Tukey, J. W. (1963). The Examination and Analysis of Residuals. Technometrics, 141-160 .
- Bentéjac, C., Csörgo, A., & Martínez-Muñoz, G. (2019). A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54, 1937 - 1967.
- Berman, J. J. (2013). Providing structure to unstructured data. In *Elsevier eBooks* (pp. 1–14).
- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.
- Camacho-Collados, J., & Pilehvar, M.T. (2017). On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis. ArXiv, abs/1707.01780.
- Chen T, Guestrin C. 2016. Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery.
- Cunningham, P., Cord, M., & Delany, S. (2018). Chapter 2 Supervised Learning.
- Dasu, T., & Johnson, T. (2003). Exploratory Data Mining and Data Cleaning .
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- Emami, A., Papineni, K., & Sorensen, J.S. (2007). Large-Scale Distributed Language Modeling. 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, 4, IV-37-IV-40.
- Fan, L., Li, L., Ma, Z., Lee, S., Yu, H., & Hemphill, L. (2023). A Bibliometric Review of Large Language Models Research from 2017 to 2023. ArXiv, abs/2304.02020.

- Hastie, T., Tibshirani, R., & Friedman, J. (2017). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.
- Hilbert, S., & Bühner, M. (2020). Principal Components Analysis. Springer.
- V. A. Ingle, "Processing of unstructured data for information extraction," 2012 Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, India, 2012, pp. 1-4)
- Kannadasan, K., Pandiyan, P., Vinoth, R., & Saminathan, R. (2012). Time To Recruitment in an Organization through Three Parameter Generalized Exponential Model.
- Khder, M.A. (2021). Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. International Journal of Advances in Soft Computing and its Applications.
- Kim, S. W., & Gil, J. M. (2019). Research paper classification systems based on TF-IDF and LDA schemes. Human-centric Computing and Information Sciences, 9(1), 30.
- Kurita, T. (2020). Principal component analysis (PCA). In Computer vision (pp. 1-7). Springer, Cham.
- Le, Q. V., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents.
  arXiv preprint arXiv:1405.4053
- Lukauskas, M., Šarkauskaitė, V., Pilinkienė, V., Stundžienė, A., Grybauskas, A., & Bruneckienė, J. (2023). Enhancing Skills Demand Understanding through Job Ad Segmentation Using NLP and Clustering Techniques. Applied Sciences, 13(10), 6119.
- Meijer, H. J., Truong, J., & Karimi, R. (2021). Document embedding for scientific articles: Efficacy of word embeddings vs TFIDF.
  arXiv preprint arXiv:2107.05151.
- Meir, R., & Rätsch, G. (2003). An introduction to boosting and leveraging. In S. Mendelson & A. J. Smola (Eds.), Advanced lectures on machine learning (pp. 118-183). Springer.
- Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2024). A comprehensive overview of large

language models. arXiv preprint arXiv:2307.06435v8. Retrieved from https://arxiv.org/abs/2307.06435

- Poetsch, M., Corrêa, U.B., & Freitas, L.A. (2019). A Word Embedding Analysis towards Ontology Enrichment. Res. Comput. Sci., 148, 153-164.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.
  arXiv preprint arXiv:1908.10084
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis1. Journal of Computational and Applied Mathematics, 20, 53-65.
- Sefton, P., Barnes, I., Ward, R., & Downing, J. (2009). Embedding Metadata and Other Semantics in Word Processing Documents. Int. J. Digit. Curation, 4, 93-106.
- Sammut, C., & Webb, G. I. (eds). (2011). TF-IDF. In Encyclopedia of Machine Learning.
- Sentence Transformers. (2023). all-mpnet-base-v2. Hugging Face. Retrieved from https://huggingface.co/sentence-transformers/all-mpnet-base-v2
- Shlens, J. (2014). A Tutorial on Principal Component Analysis. arXiv.org. Retrieved from https://doi.org/10.48550/arXiv.1404.1100
- Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. Advances in Neural Information Processing Systems, 33, 16857-16867.
- Starbuck, C. (2023). The Fundamentals of People Analytics: With Applications in R. Springer Cham.
  https://doi.org/10.1007/978-3-031-28674-2
- Tennenholtz, G., Chow, Y., Hsu, C., Jeong, J., Shani, L., Tulepbergenov, A., Ramachandran, D., Mladenov, M., & Boutilier, C. (2023). Demystifying Embedding Spaces using Large Language Models. *ArXiv, abs/2310.04475*.
- Wu, B. (2021). K-means clustering algorithm and Python implementation. 2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE), 55-59.

- Wüthrich, Mario V, and Michael Merz. “Natural Language Processing.” Springer Actuarial. Switzerland: Springer International Publishing AG, 2022. Web.

- XGBoost Developers. (n.d.). XGBoost Parameters1. Retrieved March 11, 2024, from https://xgboost.readthedocs.io/en/latest/parameter.html

- Xu, S., Lu, B., Baldea, M., Edgar, T.F., Wojsznis, W.K., Blevins, T.L., & Nixon, M. (2015). Data cleaning in the process industries. Reviews in Chemical Engineering, 31, 453 - 490.

Appendix 1: Job Profiles and Keywords

Note: *Frequent job titles were removed due to confidentiality reasons.*

Cluster	Silhouette Score	TD-IDF Keywords	Job Profile
9	0.35641903	supporting, approach, poverty, supervision proven, africa experience, strategies utilities, excel, multidisciplinary, proficient, comprehensive, advantage excellent, investment projects, access hydropower, proven record, cross, application active, delivering, ipf, desirable including, desirable experience, project demonstrates, experience fiduciary, added advantage, status, fluency, delivery, demonstrates solid, knowledge power, sector experience, quantitative	Energy and Sustainability
13	0.30661812	duration years, grasp, skills attention, cases, skill, good narrative, situations continuous, situation, skills flexibility, duration, domain knowledge, domain, knowledgeable, relationships necessary, hr analytics, disciplines, disciplines years, hours defined, hours, developing, leverages, proficiency word, competencies professional, proficiency microsoft, skills responsiveness, distressed situations, communication including, professional manner, teamwork collaborative, systems software excellent	HR Professionals
18	0.27356663	category strategy, certificates, certificates procurement, teams strong, minimum 12, systems process, systems knowledge, power point data, managing relationships, practices processes, prepare, competencies demonstrates, manage complex, pricing, self initiated, deliver results specific, degree business, process planning, cycle proven, skills demonstrated, desirable knowledge, limited tactical, coupled courses, consultant, construction, privacy, conflict ability, states, courses certificates, oral communication	Procurement and Acquisitions
19	0.2104928	prioritize work, operations key, operations understanding, print, feedback, field minimum, prepare edit, internal external partners, constituencies foster, assess, experience excellent, confidential, skills demonstrated, evaluation, audience outreach, available, products develop, large, benefits, social political, skills multitask, existing, creative suite, skills spoken, campaign management, tools awareness, tools ability, nurture maintain, tact, multitask initiative	Communications and External Affairs Specialists

1	0.18735065	french arabic, tackling highly, tact, impact identifies, occasionally, component parts articulates, occasionally conflicting, boundaries internally, skill, concepts, doctor, tasks demanding, tasks effectively, intelligence, boost morale, impact behaviors, boost, bilateral development, team spirit, bilateral, teams brings, behaviors context, teams work, reporting experience, intended, doctor llb, techniques, officials civil, intranet internet, management government	Legal Professionals
14	0.17891029	cnt solutions, useful, useful explanation, conducts analysis, unit outside, management andor, framework ensure, significant concern, effects establishes, required relevant, serves best, language skill, effectively evaluate, leads example, capital, flexible work, concern opportunity, adopts inclusive, capacity deliver, logical accounting, longterm vpuwide, operations objectives, implications decision, implementation project, pieces, look data, looking information, trends missing, commerce related, processes including	Corporate Finance and Resource Management
8	0.16858521	inquiries, actions, functional responsibility, staff managers, boundaries initiates, clients colleagues, clients manage, skills attention, sensitive matters, seek, retrieve, continued, continued learning, responsibilities demonstrates, respond requests, deadline, relevant internal, related timely, timely processing, adaptability able, promote, pressure excellent, positive professional, absence, focus, follow team, organize data, organize coordinate, hold, related field	Program Management and Administrative Support
11	0.15766272	mutuality, mutuality respect, conduct research, positive, adapt, guide, technology systems, deliver high, evidenced, problemsolving capabilities, healthcare, public financial, electronic, electronic medical, emergency context, make smart, hsd, agencies organizations, collaborating, management work, africa experience, highquality analysis, managers interact, strong written, structure, highly desirable, professionals, advise, commitment, organizational boundaries	Healthcare and Public Health

4	0.14447464	assessing realistic, assess, arena translates, individual able, transport including, approach, transport systems, trends, good oral, goals, bringing, calls, platforms, skills knowledge, planning engineering, planning disaster, client countriesexcellent, social inclusion, socioeconomic, evaluation climate, clearly, experience providing, experienced, strategy conducting, formulating, formulating assessing, french required, gender impact, experience dealing, 10	Transport and Urban Development Specialists
6	0.13249621	experience years, pressure selfmotivated, financial statements, preferably specialization, packages, good team, rdbms, rdbms sql, oracle rdbms, distill, techniques, big, build, maintain improve, strong quantitative, cfa program, strong data, client stakeholder, think, statements, consulting ideally, latin, ms access, degree equivalent, demonstrated quantitative, institution rating, requests, income markets, stakeholder needs, reporting applications	Quantitative Analysts
0	0.1308647	safetygood, training certificate, minibuses, mechanical, dressed presentably, driver preferably, timings, driving knowledge, effectively french, english added, english knowledge, environment safe, equivalent diploma, examinations, experience multicultural, medical clearance, tasks, flexible dressed, french spoken, french verbal, functions, include, initiative help, knowledge working, languages, specialized skillsin, license candidate, licensed, related, 05	Drivers and Security Professionals
2	0.11519858	supervision demonstrate, areas, demeanor, demonstrate good, degree equivalent, strong communication, degree adaptability, stay, demonstrates ability, emerging, english spanish, routine nonroutine, exhibit good, confidential nature, arabic, experience ms, reference, undertake diverse, good understanding, quality control, proven track, identify prevent, proficiency prior, know, knowledge ability, behavior, bachelor, attention details, resolve routine, environment function	Program Assistants and Administrative Professionals

3	0.10485982	field economics, strategically, mission, staff levels, spoken, collaborate teams, simultaneously, securing new, communication demonstrates, computer, consistently, contributing, quality technically, country economics, teamwork collaboration, teamwork skills, data analytical, produce userfriendly, presentations, development policies, policy making, analyses produce, banks, plus strong, plays, perspective, multitask need, efficiency, mentoring, data collection	Econometrics and Statistics Professionals
16	0.094340734	environmentally related, minimal supervision, possesses, supports, longterm, policies practices, environmental issues, law, knowledge understanding, strong plus, knowledge sustainability, management climate, accountability meet, phd environmental, client needs, decisions interprets, personal ownership, project preparation, climate risk, extractive, manufacturing, africa, projects programs, proven relevant, field minimum, orientation ability, minimal, contributing, banks safeguards, field relevant	Environment, Climate Change and Sustainability Professionals
15	0.09237449	cloud platforms, working experience, sdlc, sector, extensive experience, positive, computing, direct, devices, tasks tight, education relevant, department, technical nontechnical, deal, persistent proactive, ly, current business, exchange, agile persistent, advising looks, transformation, small, review, skills proven, build effective, creates, integrated, experience strong, programming languages, client needs	IT Professionals
7	0.083528765	necessary, motivation, senior staff, short deadlines, people skills, knowledge analytical, project cycle, economics management, interpersonal diplomatic, policy related, selfstarter, courage convictions, anchor integrate, power, skills required, administration social, adapt, integrate overall, stakeholder engagement, stakeholders coherent, respond quickly, messages, strong integrative, record delivery, overall economic, race, capacity work, bank experience, anticipates, political judgment	Core Operations

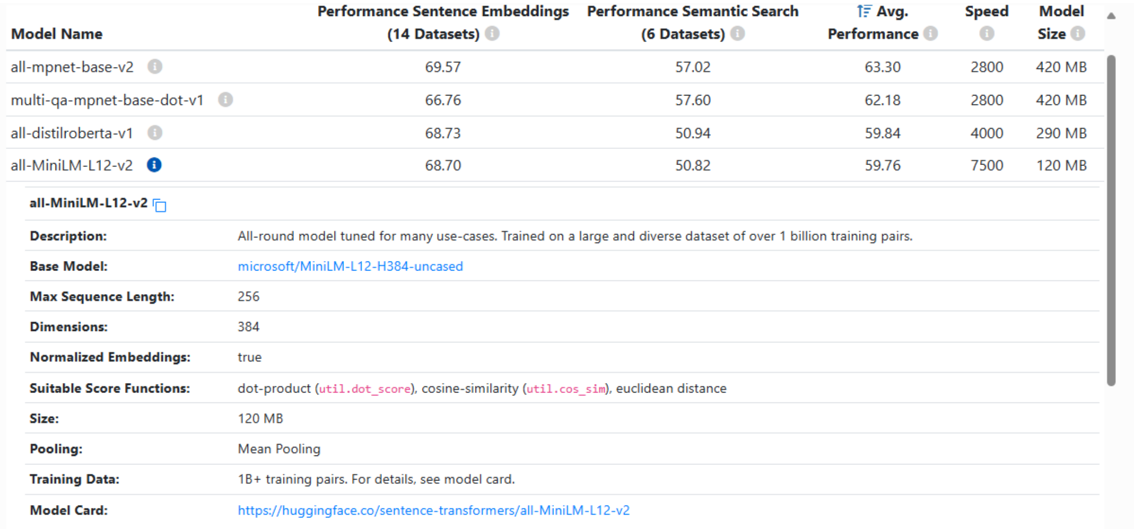
5	0.06503343	coordination, mobilization, articulate, knowledge countries, enable participate, innovation ability, institutional, knowledge sector, practice, transaction advisory, demonstrating longterm, advanced degree, andor advisory, longterm perspective, time management, combination education, commitment development, understand enable, record portfolio, loan, bank private, backed references, degree economics, using, learning, required fluency, standards willingness, problem solving, understanding local, skills backed	Investment and Advisory
17	0.039745808	arena understands, achieves, contributions country, development challenges, think strategically, coach, good understanding, sensitivity, strong organizational, ability support, approach, highly desirable, planning instruments, varied, applications adapt, sound logical, resilience, research translate, applications contributions, queries, achieves results, better, ability influence, development programs, lending operations, academic, facts data, recommend, instruments resettlement, support sound	Social Development
10	0.03713427	execute tasks, carry work, information pushes, skilled communicating, information complex, logical business, displaying, displaying sense, skills proficiency, using charts, techniques appropriately, make informed, urges focus, urges, pertinent systems, solutions seeks, unstated needs, solve problems, making analyzing, changing business, engages active, policy regulation, understanding functions, stay abreast, certified public, matters, plus provides, player strong, uptodate, degree preferably	Program Assistants and Administrative Professionals
12	0.015792748	leading bank, leadership skills, lead innovate, internalexternal, markets including, self, financial market, matrix, institutions related, command, tact, minimum 12, input, education experience, role, demands, clients business, multitask deliver, circumstances, certifications, notice, initiative drive, independently senior, capacity multitask, business risk, diversity, successful, teamwork skills, clients demonstrates, queries	Senior Investment and Advisory

Appendix 2: Other Small Language Models used for embedding generation

To assess the embeddings quality of the tested models, the PCA, K-Means and HDBSCAN algorithms were used. This helped on evaluating the retention of information when reducing the dimensionality of the models output, as well as assessing how well defined the clusters would be after applying the corresponding clustering methods.

Model: all-MiniLM-L6-v2

Model Card

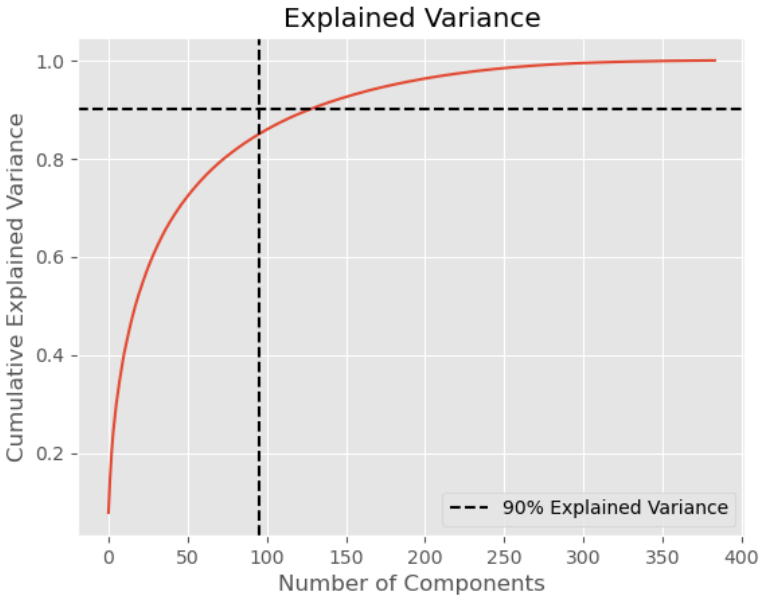


Model Name	Performance Sentence Embeddings (14 Datasets)	Performance Semantic Search (6 Datasets)	Avg. Performance	Speed	Model Size
all-mpnet-base-v2	69.57	57.02	63.30	2800	420 MB
multi-qa-mpnet-base-dot-v1	66.76	57.60	62.18	2800	420 MB
all-distilroberta-v1	68.73	50.94	59.84	4000	290 MB
all-MiniLM-L12-v2	68.70	50.82	59.76	7500	120 MB

all-MiniLM-L12-v2	
Description:	All-round model tuned for many use-cases. Trained on a large and diverse dataset of over 1 billion training pairs.
Base Model:	microsoft/MiniLM-L12-H384-uncased
Max Sequence Length:	256
Dimensions:	384
Normalized Embeddings:	true
Suitable Score Functions:	dot-product (`util.dot_score`), cosine-similarity (`util.cos_sim`), euclidean distance
Size:	120 MB
Pooling:	Mean Pooling
Training Data:	1B+ training pairs. For details, see model card.
Model Card:	https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

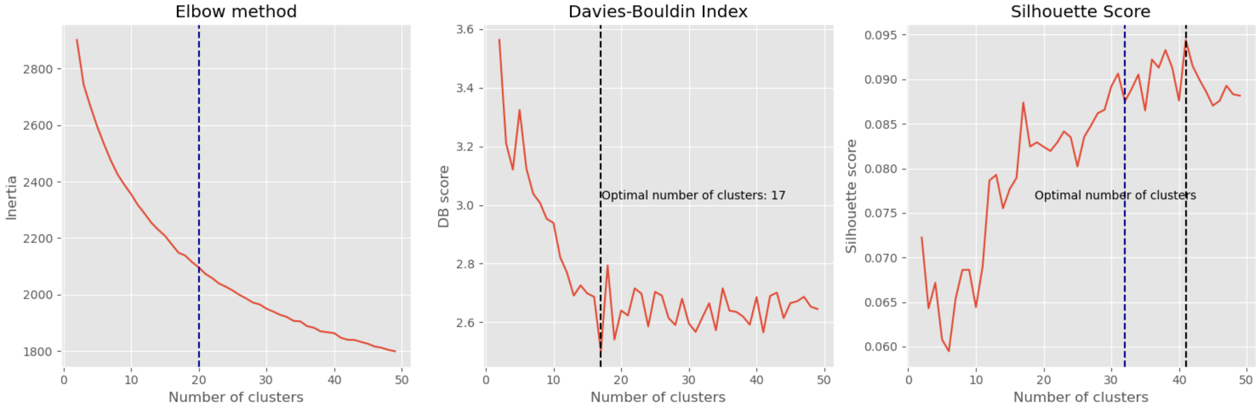
PCA Results

In comparison to the selected model for this work, the explained variance curve below shows less information when selecting the same number of components (95).



K-Means Results

The average silhouette score for 40 clusters with this model was around 0.087, which was lower than the selected model for this work, also, the Silhouette Score suggest a high number of clusters, which could represent a challenge in terms of implementation of recruitment strategies.



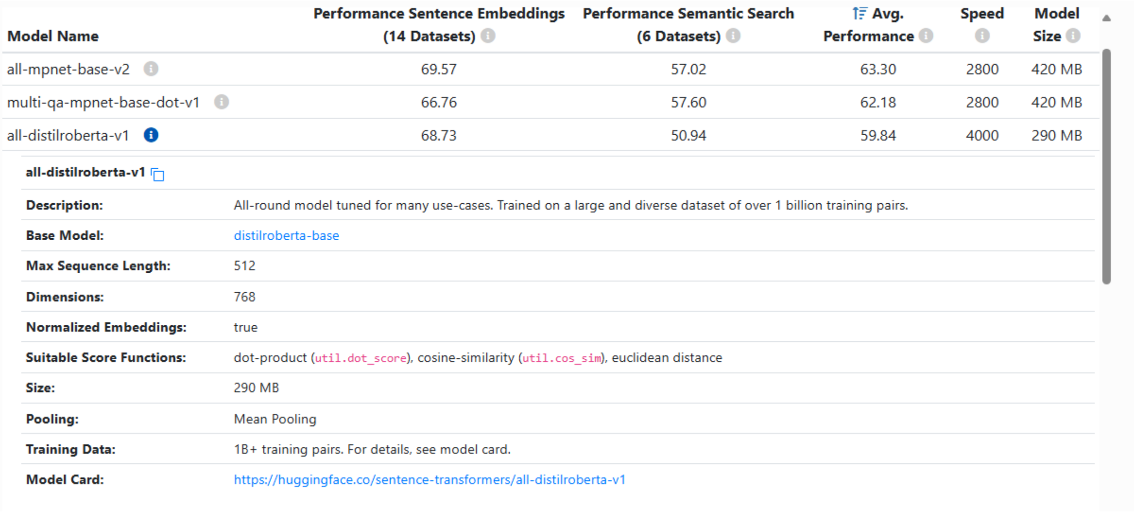
HDBSCAN Results

Silhouette scores for this model was quite low, presenting even a negative number for the iterations with a higher number of clusters. Also, there was a leap from 12 to 34 clusters during the iteration, which could make the analysis and interpretation of clusters more challenging, given than 12 clusters might underrepresent the actual job profiles and 34 might be too many to handle and design simpler and executable recruitment strategies.

Cluster Size	Minimum Samples	Davies Boulding Score	Silhouette Score	Number of Clusters
10	10	6.698	0.024	3
30	30	5.834	0.015	3
30	30	5.712	0.006	3
50	50	5.148	0.006	3
20	20	5.992	0.005	3
30	30	4.581	0.002	4
30	30	4.581	0.002	4
50	50	4.274	0.001	4
50	50	4.274	0.001	4
40	40	4.401	0.001	4
40	40	4.401	0.001	4
40	40	4.401	0.001	4
40	40	4.401	0.001	4
40	40	4.401	0.001	4
50	50	3.530	-0.009	5
50	50	3.530	-0.009	5
10	10	2.784	-0.032	10
10	10	2.544	-0.047	13
30	30	2.168	-0.080	12
20	20	2.086	-0.094	12
10	10	1.866	-0.097	34
20	20	1.878	-0.114	17
10	10	1.623	-0.120	67
20	20	1.816	-0.124	23
20	20	1.731	-0.131	29

Model: all-distilroberta-v1

Model Card

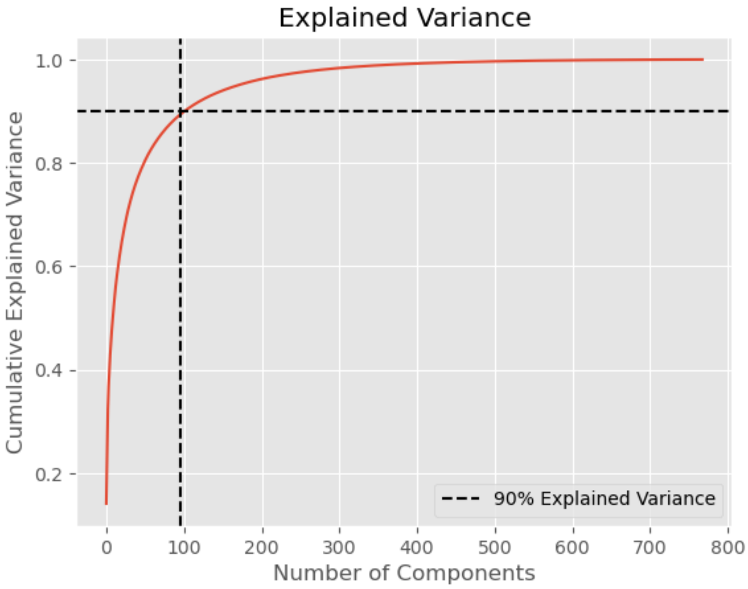


Model Name	Performance Sentence Embeddings (14 Datasets)	Performance Semantic Search (6 Datasets)	Avg. Performance	Speed	Model Size
all-mpnet-base-v2	69.57	57.02	63.30	2800	420 MB
multi-qa-mpnet-base-dot-v1	66.76	57.60	62.18	2800	420 MB
all-distilroberta-v1	68.73	50.94	59.84	4000	290 MB

all-distilroberta-v1	
Description:	All-round model tuned for many use-cases. Trained on a large and diverse dataset of over 1 billion training pairs.
Base Model:	distilroberta-base
Max Sequence Length:	512
Dimensions:	768
Normalized Embeddings:	true
Suitable Score Functions:	dot-product (`util.dot_score`), cosine-similarity (`util.cos_sim`), euclidean distance
Size:	290 MB
Pooling:	Mean Pooling
Training Data:	1B+ training pairs. For details, see model card.
Model Card:	https://huggingface.co/sentence-transformers/all-distilroberta-v1

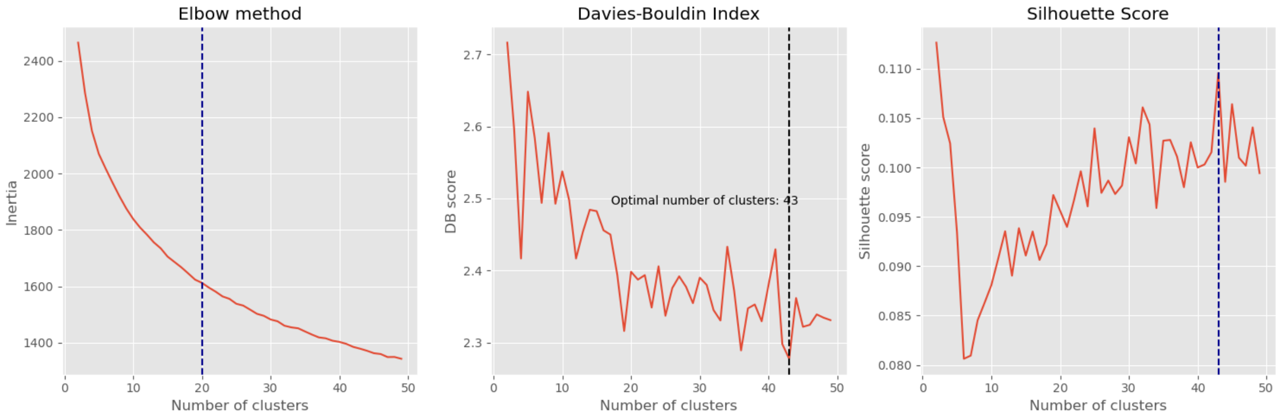
PCA Results

With the embeddings generated by this model, the PCA algorithm was able to retain about the same amount of information with the same number of selected components (95).



K-Means Results

With this model, similar results were obtained when comparing the three metrics to the selected model for this work, however, a challenge was encountered when selecting the optimal number of clusters, given that the Davies-Bouldin Index and Silhouette Score suggested a high number of clusters (43)



HDBSCAN Results

A similar challenge is shown for this model compared to the previous one, in which low silhouette scores are found, which tend to be negative as the number of clusters increases, indicating poorly defined clusters.

Cluster Size	Minimum Samples	Davies Boulding Score	Silhouette Score	Number of Clusters
20	20	5.086	0.063	3
10	10	4.519	0.036	4
30	30	5.307	0.009	3
30	30	5.307	0.009	3
30	30	5.307	0.009	3
30	30	5.307	0.009	3
50	50	4.806	0.009	3
50	50	4.806	0.009	3
50	50	4.806	0.009	3
50	50	3.982	-0.005	4
20	20	4.461	-0.009	4
20	20	4.461	-0.009	4
40	40	4.001	-0.020	4
40	40	4.001	-0.020	4
40	40	4.001	-0.020	4
40	40	4.001	-0.020	4
50	50	3.665	-0.022	3
40	40	3.097	-0.042	6
20	20	2.106	-0.100	13
20	20	2.106	-0.100	13
30	30	2.001	-0.119	13

10	10	2.066	-0.120	17
10	10	2.020	-0.120	19
10	10	1.946	-0.128	23
10	10	1.521	-0.168	71

Appendix 3: Supervised Learning Models Initial Results

To solve the prediction problem presented in this work, two different approaches were tested by using a bag-of-words/TF-IDF model for the feature extraction of the data, and an embeddings based approach using the embeddings generated by the selected Small Language Model.

Classification Using TF-IDF

For this approach, the number of features set for the TF-IDF Vectorizer was set to 1000 after several tests, with an n-gram range of (1,2). Internal and External applications from candidates were also used as features.

Logistic Regression

Model evaluation metrics:

Metric	Score
Accuracy	0.64
F1	0.63
Precision	0.63
Recall	0.64
ROC/AUC	0.62

Top 10 features:

Feature (Terms)	Coefficient
Instrument	1.559
Community	1.505
Planning	1.478
Sector	1.208
French	1.185
Infrastructure	1.186
Plus	1.171
Advanced	1.153
Global	1.141
Economy	1.131

Classification Using Embeddings

The features used in this approach were:

- Internal Applications
- External Applications
- Token length of the requirements in the job description
- Level of the position
- PCA components from 0 to 30

These were the same variables used for the final version of the selected model (XGBOOST).

Decision Tree

Metric	Score
Accuracy	0.626
F1	0.621
Precision	0.622
Recall	0.626
ROC/AUC	0.616

Random Forest

Metric	Score
Accuracy	0.657
F1	0.598
Precision	0.664
Recall	0.543
ROC/AUC	0.665

Logistic Regression

Metric	Score
Accuracy	0.656
F1	0.657
Precision	0.660
Recall	0.656
ROC/AUC	0.695

XGBOOST

Metric	Score
Accuracy	0.662
F1	0.659
Precision	0.659
Recall	0.662
ROC/AUC	0.696