UNIVERSITÀ DI PISA

Data Mining & Machine Learning

Resume Classification
Project Report

Developed by
Daniele Laporta

Summary

Introduction **3**
Design **4**
Dataset Description & EDA 4
Feature Transformation 8
Model Implementation **11**
Classification Results **13**
Conclusions **14**
Possible Improvements **15**
Additional Study **15**

Introduction

The aim of the project is to build a resume classification model using a Linkedin dataset obtained from Kaggle.

This report is divided into 6 main sections: *Design*, *Model Implementation*, Classification *Results*, *Conclusion, Possible Improvements* and *Additional Study.*

The project originates from the need to gain a deeper understanding of the job market. In fact this task will evolve into an *Additional Study*, which will be presented in the corresponding section.

The *Design* phase begins with the exploration of the dataset, including necessary data preprocessing steps and exploratory data analysis. This phase encompasses data understanding, brainstorming solutions, feature transformation, and knowledge extraction from the data.

Following the *Design* phase, the *Model Implementation* phase regards the model pipelines that include the evaluation with stratified k-fold cross-validation.

Lastly, the *Classification Results* section compares the performance of different classifiers. A statistical significance test is also performed.

Additionally, some *Possible Improvements* are presented and then the analysis delves into a more vertical job market of particular interest by developing a recommender system.

The project is available at this Drive folder.

Design

Dataset Description & EDA

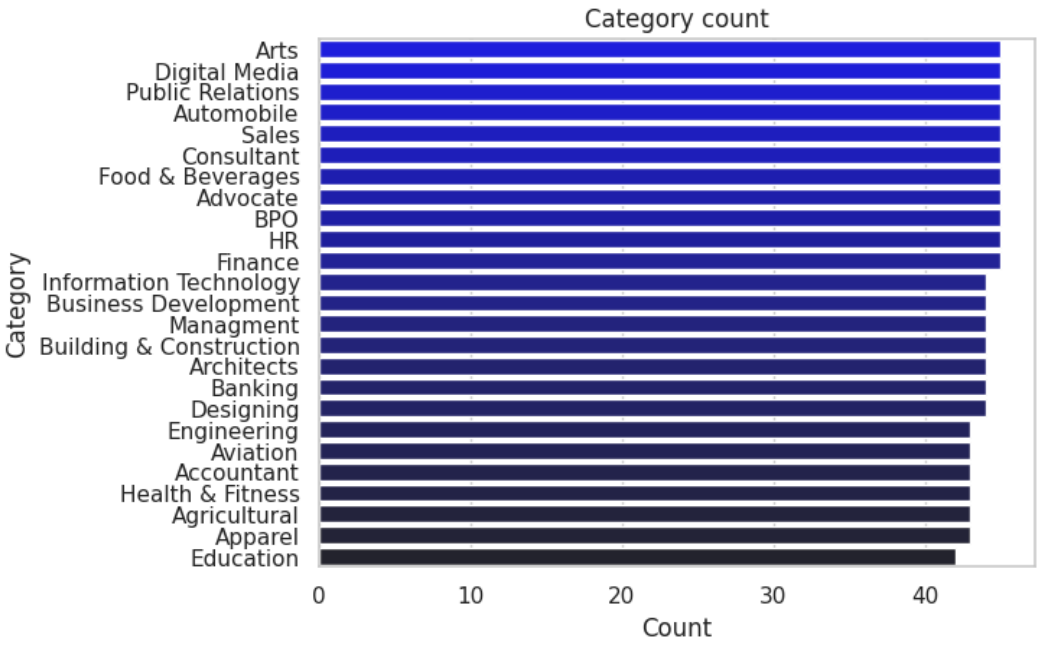
The dataset is about real people's resumes scraped from LinkedIn and publicly available on Kaggle.

The initial dataset columns are: 'category', 'linkedin', 'profile_picture', 'description', 'Experience', 'Name', 'position', 'location', 'skills', 'clean_skills'.

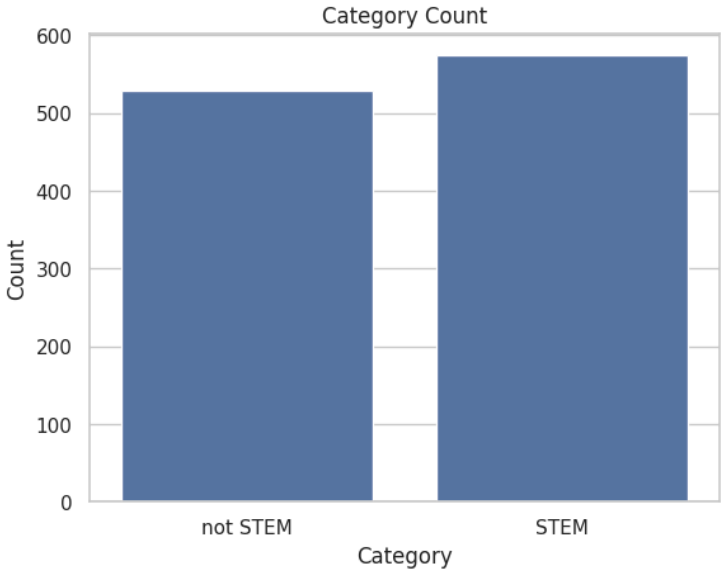
category	linkedin	profile_picture	description	Experience	Name	position	location	skills	clean_skills
HR	https://in.linke...	https://media-ex...	An experienced H...	Senior Vice Pres...	Sameer Wadhawan	Senior Vice Pres...	Gurgaon, Haryana...	['\nPerformance ...	['Performance Ma...
HR	https://in.linke...	https://media-ex...	Head Talent Acqu...	Head of Talent A...	Adarsh Krishna	Head Talent Acqu...	Pune, Maharashtr...	['\nTalent Acqui...	['Talent Acquisi...
HR	https://in.linke...	data:image/gif;b...	A Talent Acquisi...	Company NameIBM ...	Shrivas Mohit	HR@IBM	Bengaluru, Karna...	['\nHuman Resour...	['Human Resource...
HR	https://in.linke...	data:image/gif;b...	NaN	HR/Admin/Personn...	HR Hopes	HR	Pune Area, India	[]	[']
HR	https://in.linke...	https://media-ex...	Over 18 Years of...	Company NameEXLT...	Rakesh Kumar	Vice President -...	Central Delhi, D...	['\nTeam Managem...	['Team Managemen...
...	...	...	...	...	...	...	...	...	...
Aviation	https://in.linke...	data:image/gif;b...	NaN	airline and avia...	britishairhostes...	airline and avia...	Chandigarh, Chan...	[]	[']
Aviation	https://in.linke...	data:image/gif;b...	NaN	Production Manag...	Ramaiah Manjunath	Production Manag...	Krishnagiri, Tam...	[]	[']
Aviation	https://in.linke...	https://media-ex...	An MBA Graduate ...	Human Resources ...	Shubham Pradhan	HR Executive at ...	Mumbai, Maharash...	['\nManagement\n...	['Management', '...
Aviation	https://in.linke...	data:image/gif;b...	NaN	Institute of Log...	Alfiya Shaikh	Student at Insti...	Mumbai, Maharash...	['\nTally ERP\n'...	['Tally ERP', 'W...
Aviation	https://in.linke...	https://media-ex...	Currently I'm pu...	Executive Revenu...	ABHISHEK TIWARI	Pursuing MBA in ...	Thane, Maharasht...	['\nSix Sigma\n'...	['Six Sigma', 'M...

Upon initial inspection of the data, it becomes evident that there are certain columns that serve no purpose for the intended analysis. Additionally there are some null values that require attention, as well as a few recurrent mistaken words. These issues can be addressed through various data cleaning techniques.

The label identifying the ground truth is ‘category’ and, as shown below, there are 25 distinct classes, each containing approximately 40 samples.



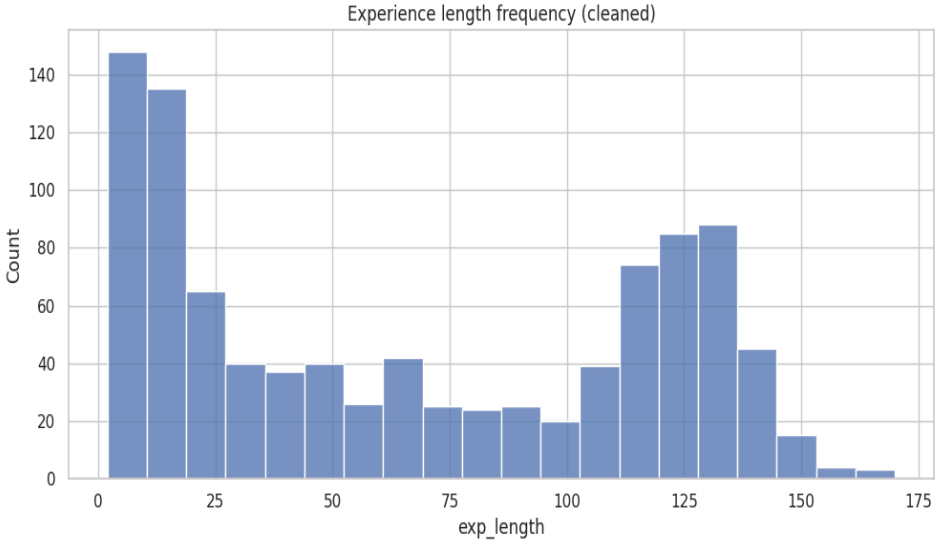
There is not enough data to perform such a varied classification task. As a result, a decision is made to simplify the task to a binary classification problem, classifying resumes as ‘STEM’ or ‘not STEM’.

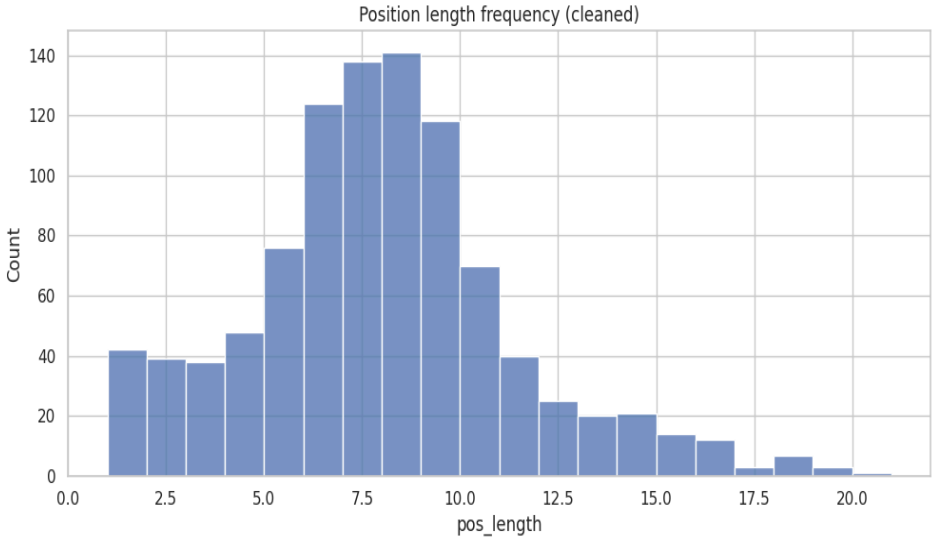


The division of classes is based on the knowledge extracted from the data. It's worth noting that different divisions can yield varied results. Several combinations were experimented with, and the one currently outlined in the notebook seems to perform better than others. Essentially classes were grouped into the STEM label if there was an affinity with this broad sector.

Some exploratory data analysis (EDA) is conducted to glean insights from the data and below are presented a few graphs that illustrate these findings.

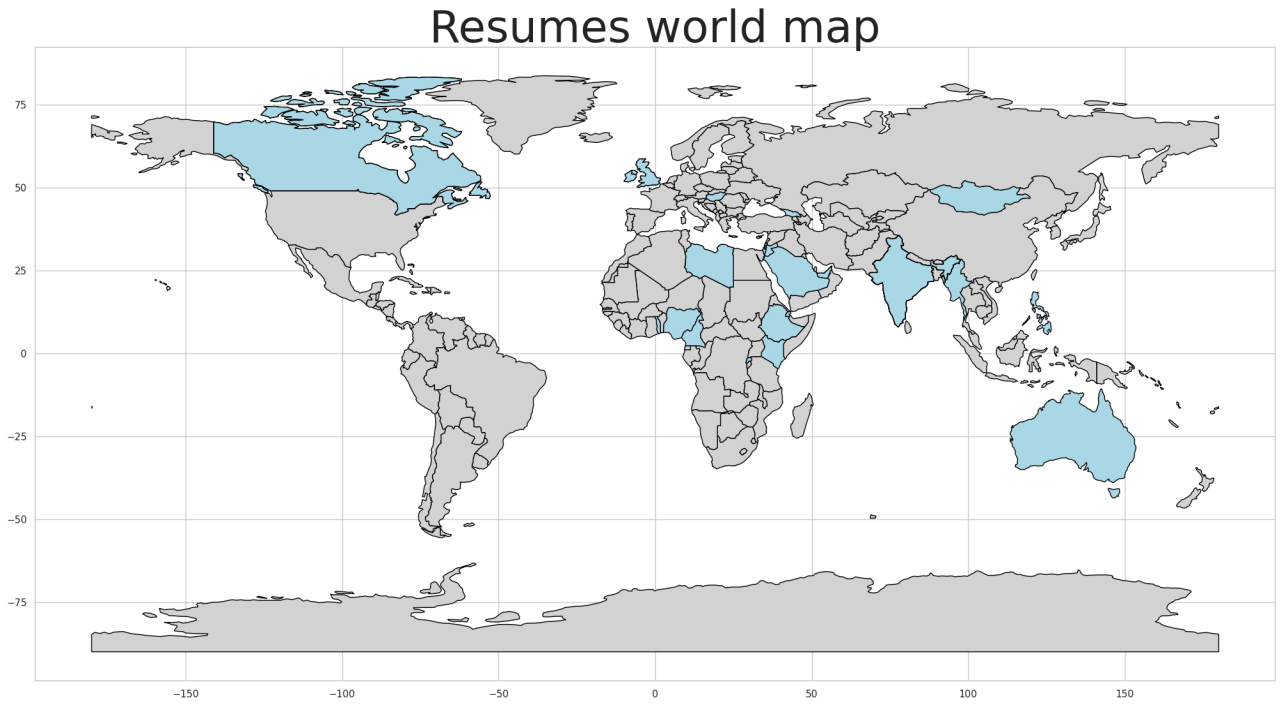
'skills' feature word cloud





The ‘position’ feature explains the job role, such as “Vice President” or “Production Manager” and as expected the length (in characters) assumes a normal distribution and is on average shorter than the ‘experience’ length which is similar, but can present more than one role.

For the ‘location’ column is made an attempt to represent countries on a word map using a dataset from GeoPandas (‘naturalearth_lowres’) and here is shown the result.

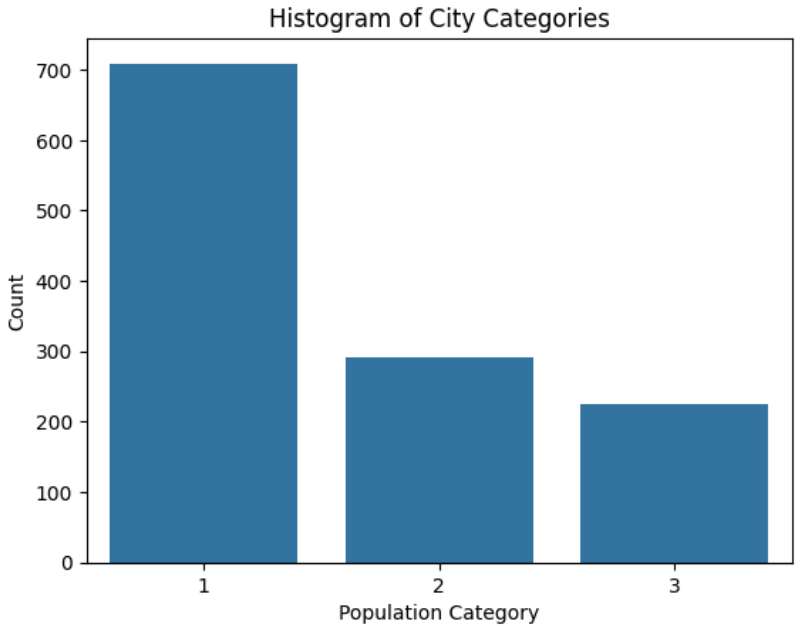


Unluckily certain countries are not highlighted in the graph, despite their presence, such as the USA. This discrepancy arises because the matching between the two datasets cannot occur for these countries. The geo dataset being used lacks specific country names like ‘California’, ‘Missouri’ or similar, but has just ‘USA’.

Feature Transformation

Continuing analysing the dataset, it is time for creating new features and extracting knowledge from data.

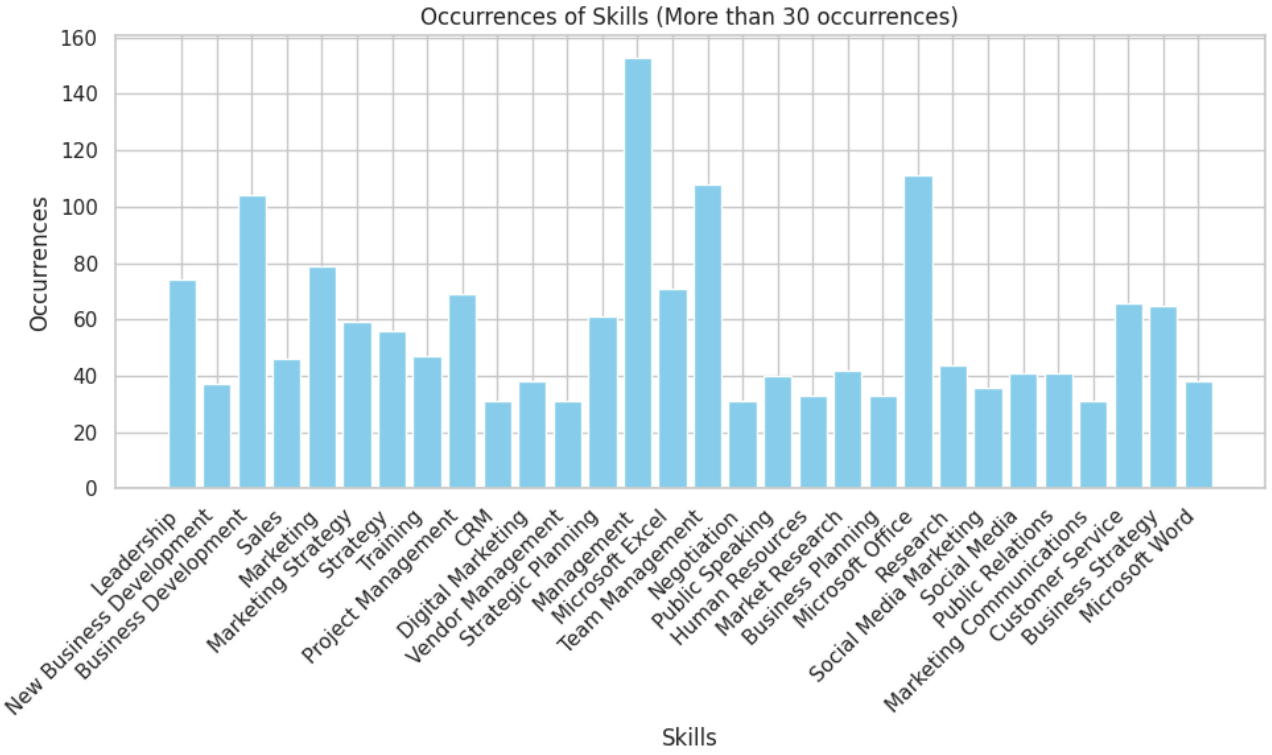
In doing so, for the ‘location’ column there are values like “Pune, Maharashtra, India”, “Worcester, United Kingdom”, “Winter Park, Florida” that present more than one word, highlighting specific information in a recurrent layout, that is the use of a comma as separator. To extract more granular information, it was decided to split these values to extract the city, the country and the region (if present). Subsequently, a dataset of around 147 thousand cities with their respective population sizes is loaded from public.opendatasoft.org. The objective is to substitute the cities in the dataset with a discretization of the population, categorising them as 'small', 'medium' or 'big', based on the following bin edges [0, 100000, 5000000] and corresponding to ‘1’, ‘2’, and ‘3’.



For the ‘experience’, ‘position’ and ‘skills’ column, which are the most informative features, a sentiment score is also computed using 2 pre-trained models specific for identifying job skills and job descriptions loaded from Hugging Face. It is, understandably, the only section that takes a little while for the running.

In addition, it is notable that some skills are more common than others, as shown in the previous word cloud. So the idea is to discretize them considering their popularity, that is, in other words, the occurrence of a skill in the entire dataset.

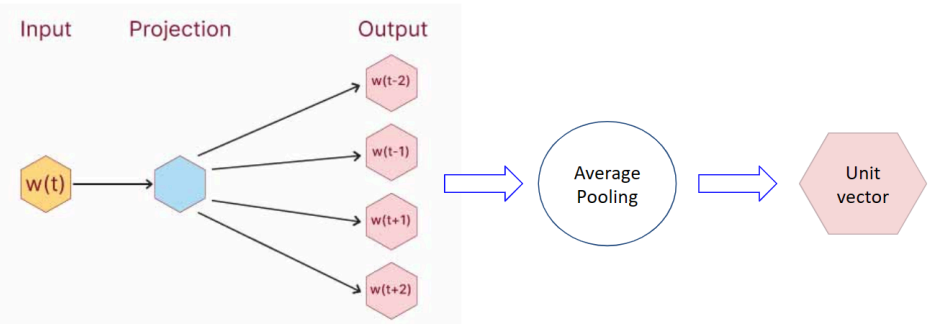
If a skill appears more frequently, it becomes less important for the model, while a skill that appears very few times can be considered rare. The training set is used to build up the skill list, which is then applied to both the training and the test set.



Out of around 6 thousand skills, most of them appear really few times, meaning that they can be considered rare. The discretization used is the following:

- x = 0 occurrences --> unknown : 0
- x <=10 occurrences --> rare: 3
- 10 < x <= 30 --> common: 2
- x > 30 --> popular: 1

To get another point of view and create new features for the model, word embeddings are created for the ‘skills’ and ‘experience’ features (Word2Vec, Doc2Vec and BERT). The average pooling is performed to obtain a unit vector suitable for the classification task.



A trial is also made by performing 1-hot encoding and label encoding on categorical columns like 'skills' but it led to the curse of dimensionality problem. Having a couple of thousands features on just 1 thousand samples is not optimal so this solution has not been followed.

Model Implementation

In this section is presented the implementation of the classification models.

To better organize the work, a pipeline is built and it comprehends the following functions:

- Transform ‘skills’ column
- Word cloud on 'skills'
- Word embedding on 'skills'
- Word embedding on 'experience'
- Normalize data and drop some columns.

Then, the classification algorithm is added to the pipeline separately using a custom function.

For the reuse of the pipeline, just pop out the old classifier and add the new one.

To get the averaged metrics at the end of the cross validation is used a custom function.

The totality of the pipelines use the stratified k-fold cross validation (with k=5) and accuracy, precision, recall and f1 score as evaluation metrics.

Every pipeline uses the same data folds, in fact a ‘skf’ object is created, saved and loaded to be reused.

The models used and compared are:

- Logistic regression
- Random Forest
- Support Vector Classifier (SVC)
- Multinomial Naive Bayes
- XGBoost

The choice of these models aims at exploring different types of algorithms, ranging from a simple logistic regression to a more robust solution such as the gradient boosting.

Before building and using the pipelines, a min-max normalization is applied to the data.

For binary classification, it is necessary to reduce the categories to two classes, ‘STEM’ and ‘not STEM’, as already explained in the dataset description section.

To better view the effects of the use of features created with word embeddings, 3 dataset configurations have been used. The distinctive configuration are the following:

- Basic configuration: 9 Features: exp_length', 'pos_length', 'population_category', 'position_scores_1', 'position_scores_2', 'experience_scores_1', 'experience_scores_2', 'skill_popularity'
- +3we configuration: same 9 features + 3 word embeddings on ‘skills’ feature.
- +5we configuration: same 9 features + 3 word embeddings on ‘skills’ feature + 2 word embeddings on ‘experience’ feature.

Classification Results

Dataset Configuration	Averaged metrics	LOG REG	RF	SVC	MULTIN. NB	XGBOOST
basic	Accuracy Precision Recall F1 Score	0.60 0.62 0.63 0.62	0.60 0.62 0.59 0.60	0.61 0.62 0.68 0.64	0.56 0.55 0.85 0.67	0.59 0.61 0.57 0.59
+3we		0.59 0.60 0.68 0.63	0.59 0.61 0.59 0.60	0.59 0.59 0.71 0.64	0.57 0.56 0.82 0.66	0.58 0.59 0.65 0.61
+5we		0.66 0.68 0.68 0.67	0.67 0.72 0.61 0.63	0.66 0.68 0.71 0.68	0.58 0.57 0.80 0.66	0.67 0.70 0.66 0.67

To assess which classifier is better, a statistical significance t-test is performed with the accuracy distributions.

No significant difference is found between any pair of two accuracy distributions of different classifiers at a 95% confidence level (alpha = 0.05).

For the accuracy distributions of a classifier are used the 5 accuracies of the 5 folds. It is not a good amount of values for a distribution but for the sake of learning the test is performed and the normality assumption to hold the paired t-tests described is checked with the Shapiro-Wilk test.

Conclusions

Is stated that there is no statistical evidence that one classifier is better than another, but looking at the classification averaged metrics it can be concluded that the best performances are obtained with the “+5we” configuration, as imagined.

Therefore, focusing on the mentioned configuration, the Logistic Regression is the simplest model and metrics are stable resulting in a solid solution.

The worst accuracy value is given by the MultinomialNB, which outperforms positively in terms of recall.

This application doesn’t require to get as many candidates as possible, but should be addressed to discard profiles surely not adapt for a STEM role and missing few suitable candidates is not impacting as can be a false negative in an health environment application.

On the contrary, every candidate marked as STEM goes through the selection and becomes a cost interviewing and pursuing it, so is better to predict it correctly to not lose time and money.

These reasonings leads to the Random Forest as the preferable classifier because it reveals to be the best in terms of precision, as for 72% of times it predicts a candidate as STEM, it is actually correct.

Possible Improvements

The results obtained are not really satisfying but with the data at disposal it is difficult to get better performances. Few possible improvements can be:

- Data augmentation or generally more data at disposal (E.g. salary, cost and quality of living).
- Focus on a niche job market (additional study).
- Study historical and evolving trends of required skills per job.
- Develop an application for resume-job matching.

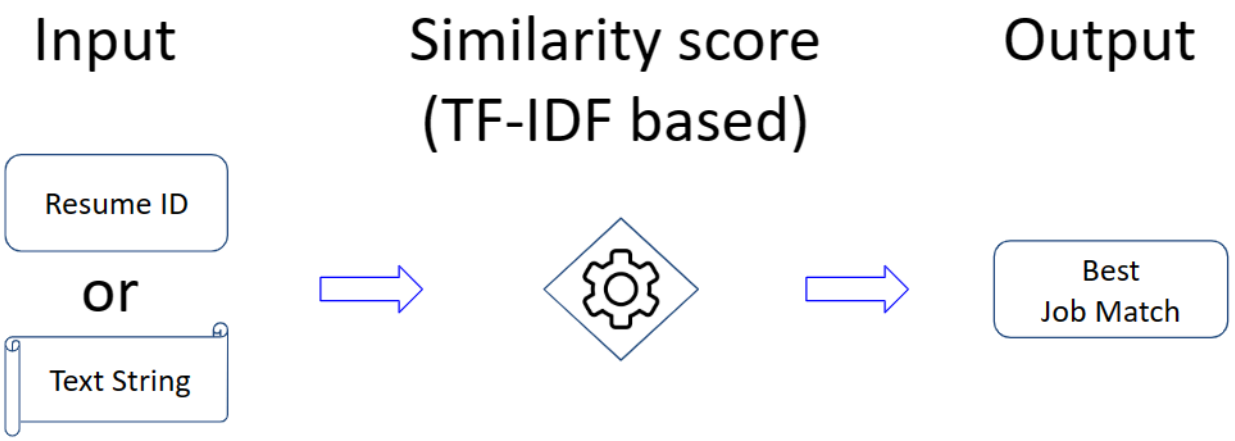
Additional Study

Python notebook reference: *Job recommender.ipynb* and *EDA_jobs.ipynb*.

As anticipated in the *Introduction* section, this task evolves into a more vertical job market analysis, focusing primarily on the "Data Science" field, which is of particular interest.

For this reason, a specific job posting dataset provided by Kaggle is used.

After some data preprocessing and EDA, a job recommender is built, based just on the cosine similarity between TF-IDF vectors previously calculated from resumes and job offers.



Given a resume ID or a text string as input, it is possible to find the best job match.

With more valuable data and technique improvements, this could evolve into a valuable tool for the recruitment process, benefiting both recruiters and job seekers.