UNIVERSITÀ DEGLI STUDI
DI SALERNO

Corso di Laurea Triennale in Informatica

TC Reccomender AI

Repository: antoninoLorenzo/RecommenderSystem_TC

Autori:
Claudio Gaudino
Antonino Lorenzo
Jacopo Passariello

Docente:
Prof. Palomba Fabio

Contents

1 Introduzione: Sistema attuale e Sistema proposto

Attualmente il mercato dello sviluppo software è tra quelli più in crescita e di conseguenza lo è anche la richiesta di professionisti. Nonostante ciò, il processo di assunzione sia per le aziende che per gli sviluppatori presenta molteplici sfide. In particolare, per gli sviluppatori, sia freelance che non, trovare un lavoro adatto alle proprie necessità e competenze è sempre più difficile in quanto richiede una lunga ricerca che può protrarsi per mesi, richiedendo il destreggiarsi tra piattaforme aziendali eterogenee. D'altro canto, anche per le aziende il processo di recruitment può essere dispendioso, sia in termini di tempo che di manodopera: richiede infatti molte interviste a molti individui. A tal proposito il sistema software proposto, Turing Careers, ha lo scopo di fornire una piattaforma digitale che provveda a semplificare il processo di ricerca di lavoro per gli sviluppatori, assieme al processo di ricerca di nuovi candidati per i datori di lavoro.

2 Definizione del problema

2.1 Obiettivi

Lo scopo del progetto *TC Reccomender AI* consiste nella realizzazione di **un modulo di intelligenza artificiale**, il quale dovrà essere integrato ad sito web, *Turing Careers* (sviluppato congiuntamente come progetto di Ingegneria del Software), che permetta agli utenti di ottenere raccomandazioni adatte al tipo di utente, in particolare:

- Raccomandazione di Offerte di Lavoro a **Sviluppatori**;
- Raccomandazione di Profili di Sviluppatori a **Datori di Lavoro**;

2.2 Specifica PEAS

Di seguito viene riportata la **specifica PEAS**:

- **Performance**: La misura di performance dell'agente è la sua capacità di restituire a datori di lavoro i migliori sviluppatori disponibili per una data posizione e per sviluppatori la migliore posizione di lavoro data la loro conoscenza;
- **Environment**: L'ambiente in cui opera l'Agente è lo spazio degli sviluppatori iscritti e lo spazio delle offerte di lavoro presenti;
- **Actuators**: L'agente agisce riportando le offerte di lavoro o i profili di sviluppatori all'utente;
- **Sensors**: L'agente percepisce input utilizzando la barra di ricerca e i relativi input e informazioni dell'utente o offerta di lavoro.

2.2.1 Caratteristiche dell'ambiente

L'ambiente su cui il agente opera risulta:

- **Completamente Osservabile**: L'agente ha una vista degli sviluppatori e delle offerte di lavoro completa;
- **Deterministico**: Le raccomandazioni che il agente fornisce in merito a dipendono dall'input inserito, assumendo che l'ambiente non sia variato;
- **Episodico:** L'agente intelligente non influenza decisioni future prendendo decisioni (una ricerca o una raccomandazione non influenza quelle successive);
- **Dinamico:** L'ambiente è soggetto a cambiamento mentre l'agente sta processando un input (nuovi sviluppatori si registrano o nuove offerte di lavoro vengono pubblicate);
- **Discreto**: L'ambiente fornisce un numero limitato di profili e offerte di lavoro sulle quali basarsi per effettuare raccomandazioni;
- **Agente Singolo**: Nell'ambiente è presente un'unico agente che effettua la raccomandazione.

2.3 Analisi del problema

L'agente, in base alle competenze fornite in input dall'utente va a restituire un insieme di Item (può trattarsi di Offerte di Lavoro oppure Profili di Sviluppatori in base al tipo di utente). Si è deciso di approcciare il problema tramite l'utilizzo di **Machine Learning**, in particolare risulta che questo problema sia un'istanza di **apprendimento non supervisionato** da risolvere attraverso **clustering**: nel caso di un Datore di Lavoro verrà effettuato clustering su Profili di Sviluppatori e verrà restituito il cluster che si avvicina di più ai parametri dati in input, analogamente per gli Sviluppatori ma effettuando clustering sulle Offerte di Lavoro. L'utilizzo di apprendimento non supervisionato permette uno sviluppo senza una conoscenza approfondita del Dominio del Problema (HR, Job Hunting e Recruiting) oltre a risolvere il problema del trovare dataset che descrivano e categorizzino le interazioni precedentemente citate. La strategia che si è scelta per questa istanza di problema è quella del clustering gerarchico, questo verrà discusso meglio nella sezione 5.2.

3 Data Understanding

3.1 Data Acquisition

A fini di training e validazione del modello, vi è la necessità di collezionare dati riguardanti **Offerte di Lavoro** nel settore IT, **Profili di Sviluppatori** e delle **Competenze** sia richieste nelle Offerte che presenti nei Profili. Possibili approcci all'acquisizione dei dati sono:

- **Dataset Pubblici (es. Kaggle):**
  - Pro: possono essere reperiti semplicemente;
  - Contro: non sempre sono disponibili fonti di dati che soddisfano i requisiti;
- **Generazione:**
  - Pro: è possibile ottenere dei dati che aderiscono meglio al problema;
  - Contro: i dati sintetici possono mancare della variabilità dei dati reali, inoltre il processo di generazione non garantisce sempre il rispetto delle relazioni nascoste tra gli attributi dei dati;
- **Web Scraping:**
  - Pro: permette di accedere a fonti di dati pubbliche, potenzialmente si hanno a disposizione grandissime quantità di dati;
  - Contro: il web scraping è una pratica spesso definita "grigia": non vi è un esplicito divieto della pratica, **ma deve rispettare delle linee guida**, che verranno discusse nella sezione 7.1.

3.1.1 Acquisizione delle Competenze

I dataset relativi alle Competenze si sono rivelati spesso incompleti o difficili da pre-processare, si è optato anche in questo caso per il **Web Scraping**, utilizzando come sito target **StackOverflow**, che pubblica ogni anno statistiche sulle tecnologie più utilizzate. E' stato realizzato quindi uno script Python analogo a quello usato per le offerte.

3.1.2 Acquisizione delle Offerte

Non essendo stati trovati dataset pubblici che soddisfino i requisiti specifici dell'agente, è stato identificato **Indeed** come un sito donatore dal quale ricavare una dataset tramite Web Scraping. Per realizzare lo scraping è stato utilizzato uno script esterno realizzato da un membro del team come progetto personale, in particolare sono stati usati Python e Selenium, facendo particolare attenzione alla frequenza di richieste inviate ai server di Indeed; la procedura utilizzata per fare scraping è stata la seguente:

- Sono stati raccolti i link di offerte in Italia, Spagna, Inghilterra e Francia.
- Sono stati raccolti nome, descrizione e location usando i link salvati.
- I dati sono stati infine esportati in un database SQLite.

3.1.3 Acquisizione degli Sviluppatori

Per motivi di etica e di privacy, si è deciso di generare un **dataset sintetico** per i **Profili di Sviluppatori**, la scelta è giustificata per seguenti motivi:

- Non sono stati trovati dataset utilizzabili nella realizzazione del modello.
- Effettuare scraping di **informazioni personali** non è compatibile con le linee guida autoimposte sull'uso etico del Web Scraping.

La generazione del dataset è stata realizzata usando il modello GPT 3.5 di Open AI tramite **ChatGPT**. Sono state generate tuple del tipo (ID,Linguaggi di Programmazione,Framework, Database, Tools,Framework Cloud) con ognuno degli attributi rappresentanti una categoria di competenze estratte con il processo di Acquisizione delle Competenze. Le tuple sono state generate mediante i seguenti prompt in ordinati:

1. **Prompt per la generazione di Linguaggi di Programmazione**:Genera un dataset di 70 tuple in formato csv di profili di sviluppatori che includano i seguenti attributi: ID, Linguaggi di Programmazione. I linguaggi di programmazione sono in formato di una lista, il cui contenuto varia da 2 a 5 linguaggi di programmazione, correlati tra di loro. I linguaggi di programmazione da utilizzare sono contenuti nella seguente lista: **Lista di Linguaggi di Programmazione**
2. **Prompt per la generazione dei framework**:Adesso, partendo dal testo in formato che hai generato, crea una nuova categoria e associa ad essa per ogni tupla da 0 a 3 framework. I framework devono corrispondere alle skill generate. Esempio: uno sviluppatore non può conoscere NumPy se non conosce Python. Esempio: se uno sviluppatore conosce Java, allora potrebbe conoscere Spring o JakartaEE. Ecco la lista: **Lista di Framework**
3. **Prompt per la generazione dei database**:desso, partendo dal testo in formato che hai generato, crea una nuova categoria "Database" e inserisci da 1 a 2 degli elementi forniti; MySQL e Oracle devono essere i più comuni. Includi gli ultimi tre in almeno 10 tuple. **Lista di Database**
4. **Prompt per la generazione dei tools**:Adesso, partendo dal testo in formato CSV che hai generato crea una nuova categoria "Tools" e inserisci da 0 a 3 degli elementi forniti: **Lista di Tool**
5. **Prompt per la generazione delle Piattaforme Cloud**:Adesso, partendo dal testo in formato CSV che hai generato crea una nuova categoria "Cloud" e inserisci da 0 a 1 degli elementi forniti: **Lista di Piattaforme Cloud**
6. **Prompt per l'estensione del Dataset**: Genera altre 50 tuple ma differenti dalle precedenti, numera gli ID a partire da **ultimo id**. Il risultato è un dataset in formato csv di 508 tuple, sono tutte incomplete in quanto mancano di Identità Geografica, Lingue Parlate e informazioni personali, che verranno generate al pre-processing dei dati.

   Il risultati sono quindi stati copiati dalla chat col LLM e inseriti all'interno di un generatore di documenti in formato CSV. Infine, sono state **generate informazioni personali** per ogni profilo di sviluppatore al fine di **far aderire i dataset al database** del Sistema in cui il modello andrà dispiegato. Sono stati generati per ogni profilo nome, cognome e location tramite l'utilizzo della libreria Faker di Python. Da questi poi si sono generati indirizzo email, password (*che verrà criptata all'inserimento nel database del sistema*) e insieme di lingue parlate (la prima assegnata per **principio di località**, le altre tramite inferenza statistica). Queste informazioni sono contenute in un dataset separato che verrà unito al dataset principale al termine del processo di Feature Engineering.

3.2 Data examination

I dataset utilizzati sono 3:

- "developers_dataset.csv": dataset esportato in CSV realizzato tramite GPT 3.5 riportante profili di sviluppatori mancanti nome, cognome e descrizione, contiene **508 istanze**.
- "offers_dataset.db": dataset esportato in formato .db contente le offerte ottenute tramite Web Scraping da Indeed, contiene **592 istanze**.
- "skills_dataset.db": dataset esportato in formato .db contente le competenze ottenute tramite Web Scraping da StackOverflow, contiene **98 istanze**.

3.3 Data exploration

Di seguito sono riportate le anteprime dei dataset utilizzati:

Dataset Competenze:

ID integer	SKILL text	TYPE text
0	JavaScript	Programming Language
1	HTML	Programming Language
2	Python	Programming Language
4	TypeScript	Programming Language
5	Bash	Programming Language
6	Java	Programming Language
7	C#	Programming Language
8	C++	Programming Language
9	C	Programming Language
10	PHP	Programming Language
11	PowerShell	Programming Language
13	Rust	Programming Language
14	Kotlin	Programming Language
15	Ruby	Programming Language
16	Lua	Programming Language
17	Dart	Programming Language
18	PostgreSQL	Database
19	MySQL	Database
20	SQLite	Database
21	MongoDB	Database
23	Redis	Database
24	MariaDB	Database
25	Elasticsearch	Database
26	Oracle	Database
27	Dynamodb	Database
30	AWS	Cloud
31	Azure	Cloud
32	Google Cloud	Cloud
33	Firebase	Cloud

Per via di come è stato ottenuto, questo dataset risulta già immediatamente utilizzabile ed è infatti la base del processo di generazione di dati sintetici per i profili di sviluppatore.

Dataset Offerte:

Name text	Description text	Location text
(Azure) Cloud Solutions Engineer	If you like this offer, please send your CV mentioning the job title to: recruitm...	Valencia, Valencia provincia
(Senior) Fullstack Developer Marketplace (m...	Ready to digitalise retail?Let's Go!(Senior) Fullstack Developer Marketplace (...	Barcelona, Barcelona provincia
18 stagiaires Business Developer APEROL S...	Date: Jan 11, 2024Location: Paris, FRAdditional Location: Moitié Sud de la Fr...	Paris (75)
2024 Intern - Software Development Enginee...	Our CompanyChanging the world through digital experiences is what Adobe's...	Edinburgh EH11
3D Artist	DETAILSDate01/10/2024Contact addressjobs@blackmouthgames.comLoca...	Madrid, Madrid provincia
AI / GenAI Manager (F/H)	L'équipe Accenture Data & AI , est le point focal de l'ensemble des services d...	Paris (75)
AI Developer - MILANO [DIG]	Trasforma le tue aspirazioni professionali in una storia di successo, entra in ...	Italia
AI Engineer	- Paris, Île-de-France, France ↩ - Technology and Research & Development ...	Paris (75)
ARTIFICIAL INTELLIGENCE ENGINEER	Artificial Intelligence Engineer ↩ Location: Rome, Italy ↩ Department: ...	Roma, Lazio
AWS Security Engineer	DescriptionAWS Security EngineerProgramme Name: LCST ↩ Location: B...	(NULL)
AZURE Cloud Engineer (H/F)	ContratCDILocalisationParis La VilletteRéférenceR-35589CatégorieBureaux	La Villette (75)
Advanced Support Engineer- Database	Advanced Support Engineer- Database-2300044XApplicants are required to r...	Madrid, Madrid provincia
Airbus UpNext - Machine Learning Engineer (...	Job Description:The new Airbus UpNext demonstrator will pave the way to s...	Toulouse (31)
Airbus UpNext - Machine Learning Engineer (...	Job Description:The new Airbus UpNext demonstrator will pave the way to s...	Toulouse (31)
AI Engineers-Machine Learning & Deep Lear...	ProgrammazioneScadenza candidatura 29 settembre, 2024Sede lavorativa P...	Padova, Veneto
AI Engineers-Machine Learning & Deep Lear...	ProgrammazioneScadenza candidatura 29 settembre, 2024Sede lavorativa P...	Padova, Veneto
Angular Front-End Developer (Málaga)	What does our company do?:GoldenRace is a global market leader for virtual ...	Málaga, Málaga provincia
Animateur d'atelier - développeur de jeux vid...	Cette offre est en partenariat avec TUMO ↩ Le centre TUMO pour les techn...	13002 Marseille 2e
Appartenente alle categorie protette Softwar...	Annuncio dedicato alle persone appartenenti alle categorie protette - legge 6...	20864 Agrate Brianza
Application Support Technician – Database ...	If you like this offer, please send your CV mentioning the job title to: recruitm...	Brindisi, Puglia
Art Design Project Manager	Department: Art DesignLocation: Barcelona, SpainDublin, IrelandSao Paolo, B...	Barcelona, Barcelona provincia
Assistant Producer	DETAILSDate12/12/2023Contact addressjobs_hr@mercurysteam.comLocati...	Madrid, Madrid provincia
Associate Software Developer	Overview:SITA FOR AIRCRAFT – Associate Software DeveloperLocation: Bar...	08005 Barcelona, Barcelona provincia
Associate Solution Engineer	Here at Appian, our core values of Respect, Work to Impact, Ambition, and Co...	Sevilla, Sevilla provincia
Associate Translation Specialist (German or ...	We are looking for an Associate Translation Specialist to join our talented tra...	Barcelona, Barcelona provincia
Athonet - Elixir Software Engineer for Mobile ...	Athonet - Elixir Software Engineer for Mobile CoreThis role has been designe...	Cernusco sul Naviglio, Lombardia
Azure Cloud Engineer	DP331-2024DPWAY S.r.l. società con esperienza decennale in soluzioni e ser...	Roma, Lazio
Azure Cloud Engineer (Attività di Operation)	by Dolmen Group ↩ ↩ Data: 2024-01-26Luogo: RemoteIn collaborazione ...	(NULL)
BUSINESS DEVELOPER H/F BTOC - BTOB	Wall Street English est un groupe international. Notre réseau de franchises c...	42000 Saint-Étienne

Data la struttura dei dati estratti è necessario un processo di feature extraction al fine di isolare le competenze richieste da ciascuna delle offerte e le lingue richieste. Inoltre, per via dei metodi utilizzati durante lo scraping, nel dataset non sono presenti duplicati.

Dataset Profili:

A	B	C	D	E	F
1	HTML,CSS,JavaScript	React,Angular,Vue.js	MySQL,Oracle	npm,Webpack,Visual Studio	Heroku
2	Python,JavaScript	Django,Flask,NumPy	MongoDB,MySQL	Pip,Homebrew,Composer	AWS
3	TypeScript,JavaScript,CSS	React,Angular,Vue.js	PostgreSQL,MySQL	Yarn,Webpack,CMake	Google Cloud
4	Bash,Python	NumPy,PyTorch,Scikit-Learn	MongoDB,Redis	Docker,Pacman,Homebrew	Docker
5	Java,C++,HTML,CSS,JavaScript	Spring,JavaEE,React,Angular,Vue.js	MySQL,Oracle,Microsoft SQL Server	Maven,Gradle,Kubernetes	Azure
6	C#,JavaScript	.NET,React,Angular,Vue.js	Microsoft SQL Server,MySQL	Visual Studio,MSBuild,Chocolatey	Azure
7	C++,C,Java	Spring,JavaEE	Oracle,MySQL	GCC,CMake,Cargo	Heroku
8	C,C++,JavaScript	React,Angular,Vue.js	MySQL,Oracle	GCC,CMake,Ansible	Cloudflare
9	Go,JavaScript	React,Vue.js	Redis,MariaDB	Go Modules,Docker,Chocolatey	AWS
10	PowerShell,Python	NumPy,PyTorch,Scikit-Learn	MongoDB,MySQL	Powershell,Pacman,Chocolatey	AWS
11	Rust,JavaScript	React,Vue.js	Elasticsearch,Redis	Cargo,LLVM's Clang,CMake	Google Cloud
12	Kotlin,Java,JavaScript	Spring,JavaEE,React,Angular,Vue.js	PostgreSQL,MySQL	Gradle,Maven,Kotlin	Heroku
13	Ruby,JavaScript	Ruby on Rails,React,Angular,Vue.js	MongoDB,MySQL	RubyGems,Bundler,Homebrew	Heroku
14	Lua,JavaScript	React,Angular,Vue.js	MySQL,Oracle	LuaRocks,CMake,Pacman	Google Cloud
15	Dart,JavaScript	Flutter,React,Angular,Vue.js	MySQL,PostgreSQL	Flutter,Visual Studio,Chocolatey	Firebase
16	Swift,JavaScript	React Native,Flutter,React,Angular,Vue.js	Oracle,MySQL	Swift Package Manager,Xcode,CocoaPods	AWS
17	CSS,JavaScript	React,Angular,Vue.js	MySQL,PostgreSQL	npm,Webpack,Visual Studio	Netlify
18	HTML,JavaScript	React,Angular,Vue.js	MySQL,Oracle	npm,Webpack,Visual Studio	Vercel
19	PHP,JavaScript	Laravel,WordPress,React,Angular,Vue.js	MySQL,PostgreSQL	Composer,PHP,Homebrew	Heroku
20	JavaScript,CSS	React,Angular,Vue.js	MySQL,Redis	npm,Webpack,Visual Studio	Vercel
21	Python,HTML,CSS	Django,Flask,NumPy	MongoDB,MySQL	Pip,Homebrew,Composer	AWS
22	TypeScript,JavaScript,CSS	React,Angular,Vue.js	PostgreSQL,MySQL	Yarn,Webpack,CMake	Netlify
23	Bash,Python,JavaScript	NumPy,PyTorch,Scikit-Learn	MongoDB,Redis	Docker,Pacman,Homebrew	AWS
24	Java,C++,HTML,CSS,JavaScript	Spring,JavaEE,React,Angular,Vue.js	MySQL,Oracle,Microsoft SQL Server	Maven,Gradle,Kubernetes	Azure
25	C#,JavaScript,HTML,CSS	.NET,React,Angular,Vue.js	Microsoft SQL Server,MySQL	Visual Studio,MSBuild,Chocolatey	Azure
26	C++,C,Java,JavaScript	Spring,JavaEE,React,Angular,Vue.js	Oracle,MySQL	GCC,CMake,Cargo	AWS
27	C,C++,JavaScript,HTML,CSS	React,Angular,Vue.js	MySQL,Oracle	GCC,CMake,Ansible	Azure
28	Go,JavaScript,HTML,CSS	React,Vue.js	Redis,MariaDB	Go Modules,Docker,Chocolatey	Google Cloud
29	PowerShell,Python,JavaScript	NumPy,PyTorch,Scikit-Learn	MongoDB,MySQL	Powershell,Pacman,Chocolatey	AWS
30	Rust,JavaScript,HTML,CSS	React,Vue.js	Elasticsearch,Redis	Cargo,LLVM's Clang,CMake	Heroku
31	Kotlin,Java,JavaScript,CSS	Spring,JavaEE,React,Angular,Vue.js	PostgreSQL,MySQL	Gradle,Maven,Kotlin	AWS
32	Ruby,JavaScript,HTML,CSS	Ruby on Rails,React,Angular,Vue.js	MongoDB,MySQL	RubyGems,Bundler,Homebrew	Heroku
33	Lua,JavaScript,HTML,CSS	React,Angular,Vue.js	MySQL,Oracle	LuaRocks,CMake,Pacman	AWS
34	Dart,JavaScript,HTML,CSS	Flutter,React,Angular,Vue.js	MySQL,PostgreSQL	Flutter,Visual Studio,Chocolatey	Firebase
35	Swift,JavaScript,HTML,CSS	React Native,Flutter,React,Angular,Vue.js	MySQL,Oracle	Swift Package Manager,Xcode,CocoaPods	AWS
36	CSS,JavaScript,HTML	React,Angular,Vue.js	MySQL,PostgreSQL	npm,Webpack,Visual Studio	Vercel
37	HTML,CSS,JavaScript	React,Angular,Vue.js	MySQL,Redis	npm,Webpack,Visual Studio	Netlify
38	Python,JavaScript,CSS	Django,Flask,NumPy	MongoDB,MySQL	Pip,Homebrew,Composer	AWS
39	TypeScript,JavaScript	React,Angular,Vue.js	PostgreSQL,MySQL	Yarn,Webpack,CMake	AWS
40	Bash,Python,JavaScript,CSS	NumPy,PyTorch,Scikit-Learn	MongoDB,Redis	Docker,Pacman,Homebrew	AWS
41	Java,C++,HTML,CSS,JavaScript	Spring,JavaEE,React,Angular,Vue.js	MySQL,Oracle,Microsoft SQL Server	Maven,Gradle,Kubernetes	Azure
42	C#,JavaScript,CSS	.NET,React,Angular,Vue.js	MySQL,Oracle	Visual Studio,MSBuild,Chocolatey	AWS

Il dataset degli sviluppatori presenta dati ben formati e privi di elementi mancanti (che può essere visto anche come un difetto della generazione dei dati sintetici), ma con le seguenti criticità:

- Il dataset presenta istanze molto simili tra di loro e una mancanza di rappresentazione di molte competenze.
- Alcune competenze sono state sovra-rappresentante, mentre altre ignorate.
- I profili risultano incompleti: occorre generare informazioni personali sintetiche come nome, cognome, email, biografia, location e lingue parlate.

4 Data Preparation

4.1 Data Cleaning

Offerte: Al fine di effettuare data cleaning sulle offerte, è necessario prima uno passo fondamentale di Feature Extraction sul dataset: estrarre dalla descrizione le Competenze richieste dall'offerta di lavoro. Di seguito è riportato il codice utilizzato:

```
from commons import (
    remove_symbols,
    extract_symbols,
    translate_skills,
    REMOVAL
)

with sqlite3.connect('../datasets/skills_dataset.db') as skills_connection:
    skills_frame = pd.read_sql_query('SELECT * FROM skills', skills_connection)
    skills_frame.set_index('ID', inplace=True)
```

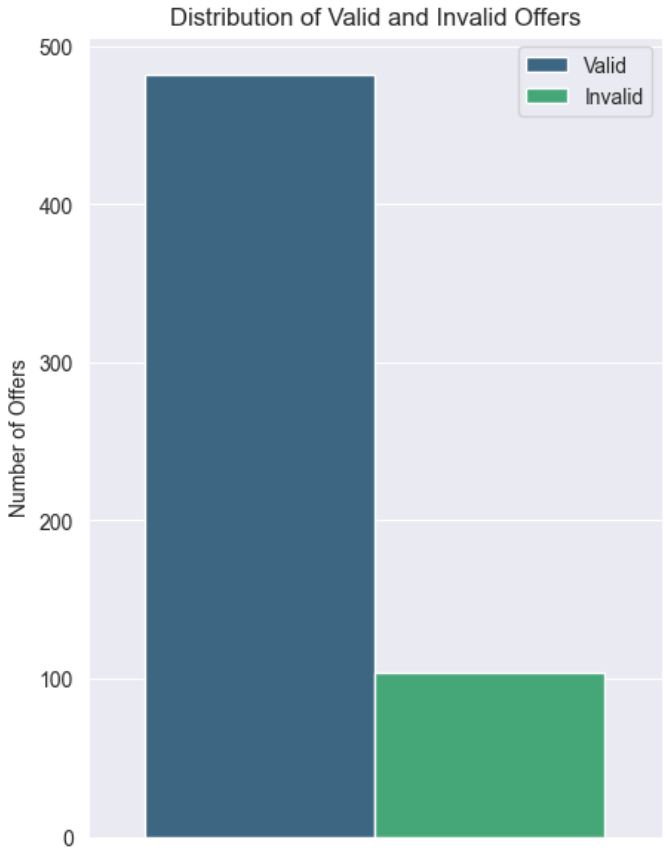
```
# Get Skill list
skills_list = [skill.lower() for skill in skills_frame['SKILL'].tolist()]

required_skills = []
for i, offer_description in enumerate(offers_frame.loc[:, 'Description']):
    desc = remove_symbols(offer_description, REMOVAL)
    offer_skills = extract_symbols(desc, skills_list)
    required_skills.append(translate_skills(offer_skills, skills_frame, to_id=True))
offers_frame.insert(len(offers_frame.columns), "RequiredSkills", required_skills)
```

Il risultato è un dataset con un nuovo attributo chiamato "RequiredSkills" che contiene una lista di skill estratte dalla descrizione.

Alcune delle offerte sottoposte a questo processo però non hanno prodotto alcuna skill. Si è deciso quindi di effettuare un **taglio orizzontale** delle offerte senza skill, questo per due motivi:

- Le offerte senza skill risultano inutili al modello.
- Il sistema in cui verrà utilizzato il modello non permette la creazione di offerte senza competenze richieste.

In seguito è riportato un grafico a barre che mostra il numero di offerte valide e non valide. E' inoltre deducibile dal grafico un rapporto offerte non valide / offerte valide di 1 a 5.



Dopo questo processo, **il dataset presenta ancora dei campi vuoti** nell'attributo "**Location**". Questo è previsto in quanto lo script di scraping non ha generato Location nel processare offerte di tipo "Remote". Si è scelto di non effettuare cleaning, in quanto le Location vuote serviranno durante il Feature Engineering per identificare le **Offerte in Remoto**.

4.2 Feature Engineering

4.2.1 Feature Extraction

Il passo successivo nel processing dei dati per il modello è il **Feature Extraction**: dalle Offerte viene estratto il tipo di posizione di lavoro (Remote oppure On Site), mentre nei Profili di Sviluppatore viene realizzata un'unica lista di Competenze.

Offerte: Viene aggiunta una nuova categoria chiamata "Location Type", questa serve a marcare la differenza tra offerte in remoto e offerte "on site". Di seguito è mostrato il codice utilizzato:

```
# Le Location En-Remoto devono essere trasformate in None
offers_frame["Location"] = offers_frame['Location'].apply(lambda location: np.NaN if location == 'En remoto' else location)
```

```
LOCATION_TYPES = {"Remote" : "Remote", "On Site" : "OnSite"}
locations_present = offers_frame["Location"].notna()
types = []

for is_present in locations_present:
    if is_present:
        types.append(LOCATION_TYPES["On Site"])
    else:
        types.append(LOCATION_TYPES["Remote"])

offers_frame.loc[:, "LocationType"] = types

offers_frame
```

Vengono identificate le lingue richieste dall'offerta di lavoro utilizzando NLP: viene identificata prima la lingua in cui l'offerta è scritta, e poi viene effettuato parsing della descrizione per identificare menzioni di lingue. Viene quindi creata una nuova categoria chiamata "Languages" che contiene la lista delle lingue estratte. Di seguito è mostrato il codice utilizzato:

```
from langdetect import detect
from string import punctuation
offers_langs = []
remove_map = {p:'' for p in punctuation}

eqs = {'it' : ('italiano', 'italian', 'italien', 'italiano'),
       'en' : ('inglese', 'english', 'anglais', 'inglés'),
       'fr' : ('francese', 'french', 'français', 'francés'),
       'es' : ('spagnolo', 'spanish', 'espagnol', 'español')}

supported_langs = [x for v in eqs.values() for x in v]


for i, desc in enumerate(offers_frame.Description):
    lang = detect(desc)

    try:
        lang = eqs[lang][0]
    except KeyError:
        print(f'offer {i} unsupported "{lang}" detected')
        lang = None

    desc = cm.remove_symbols(desc, remove_map)
    found = cm.extract_symbols(desc, supported_langs)

    found = list(found)
    for i, l in enumerate(found):
        for e in eqs.values():
            if l in e:
                found[i] = e[0]
                break

    found = set(found)

    if lang is not None:
        found.add(lang)
    offers_langs.append(found)

offers_frame.loc[:, "Languages"] = offers_langs

offers_frame
```

Profili di Sviluppatore: Al fine di operare più facilmente sulle competenze durante applicazione dell'algoritmo, vengono collassate in un'unica categoria chiamata "skills". Di seguito è mostrato il codice utilizzato:

```
skills_list = [skill.lower() for skill in skills_frame['SKILL'].tolist()]
for row in gpt_generated_frame.itertuples():
    raw_skill_set = f'{row.progLang} {row.framework} {row.database} {row.tools} {row.cloud}'

    sset = extract_symbols(
        remove_symbols(raw_skill_set),
        list(skills_list)
    )
    skills.append(sset)
```

Per entrambi i dataset infine, è stato applicato un procedimento di **Label Encoding** con la sostituzione delle stringhe contenute nella categoria "skills" e "RequiredSkills" dei rispettivi dataset con l'id della competenza stessa al fine di **alleggerire il processo di confronto tra le competenze** (in vista della creazione di una matrice delle distanze nella sezione 4.2.3) e ridurre la memoria utilizzata dal dataset.

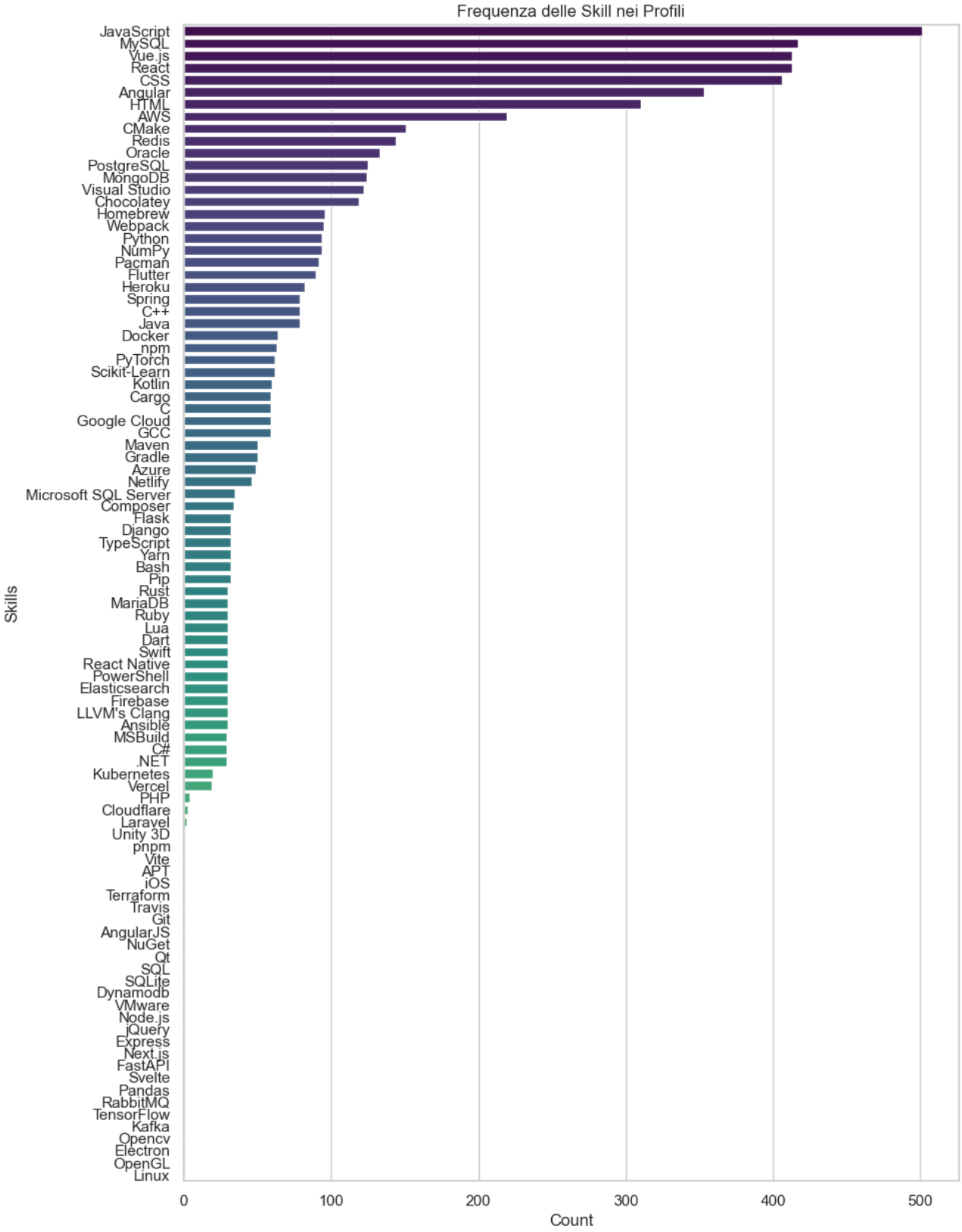
4.2.2 Data Balancing

In questo passo verranno risolte le problematiche derivate dall'aver utilizzato dataset sintetici oppure ottenuti tramite Web Scraping.
Profili di Sviluppatore: al fine di rendere più **realistico** il dataset degli sviluppatori,viene effettuata un'operazione di **bilanciamento** delle "Competenze" basata su inferenza logica e statistica: vengono create delle mappe tramite le quali effettuare **tre operazioni** nel seguente ordine:

1. **Aggiunta**: una competenza viene aggiunta ad ogni profilo con una determinata probabilità.
2. **Sostituzione**: alla presenza di una competenza questa viene rimossa e sostituita da un'altra con una particolare probabilità.
3. **Implicazione**: alla presenza di una particolare skill viene aggiunta con una determinata probabilità un'altra skill.

Queste operazioni, se applicate sulle competenze errate, possono causare una perdita di relazione logica tra le skill presenti in un profilo, cosa particolarmente curata durante la generazione del dataset tramite GPT 3.5. I dizionari sono stati quindi realizzati rispettando il più possibile le **relazioni logiche tra le competenze**.

Segue un grafico a barre della frequenza delle apparizione delle competenze **prima del bilanciamento**:

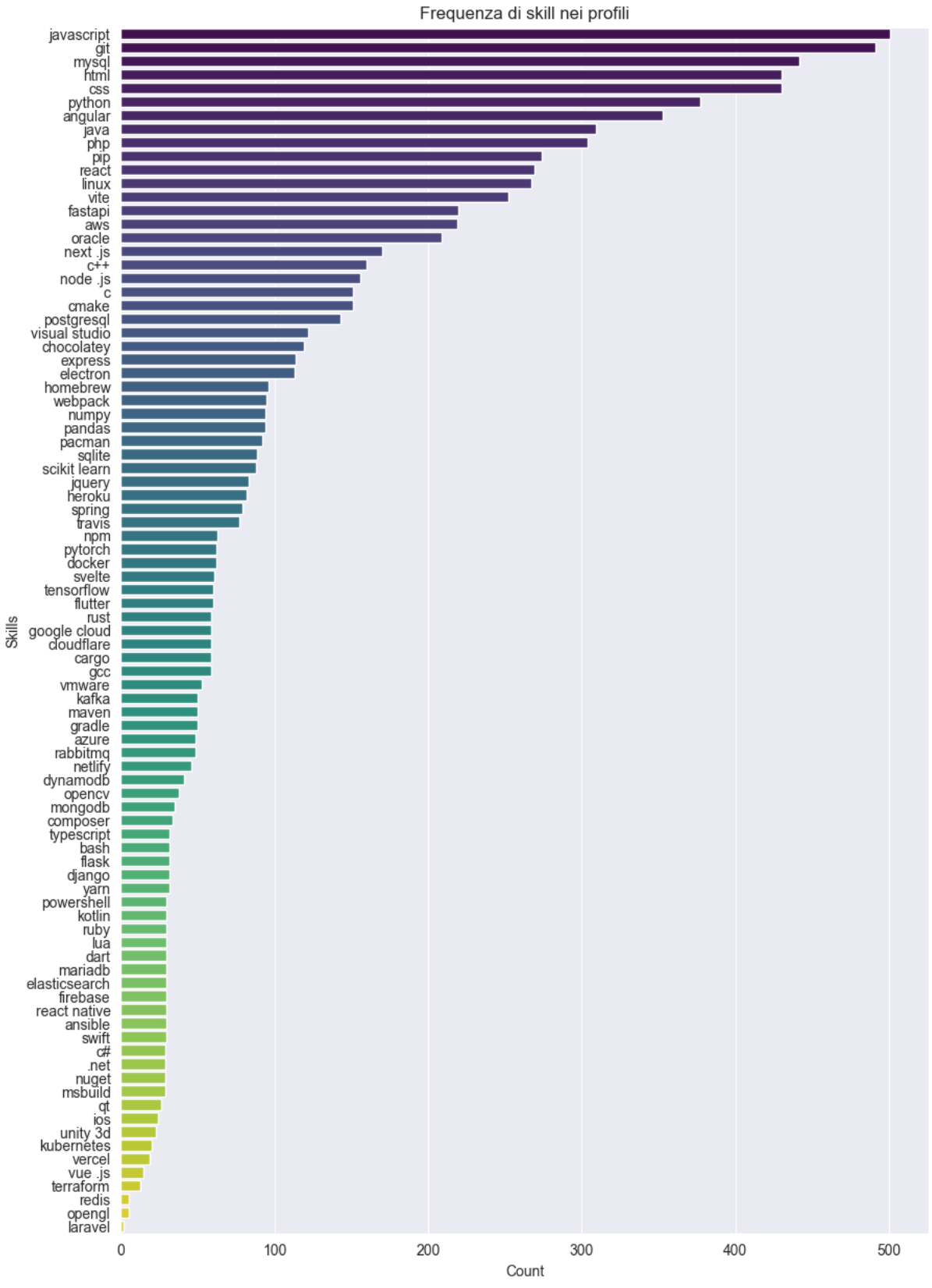


Di seguito vi è invece le varie mappe applicate per il bilanciamento:

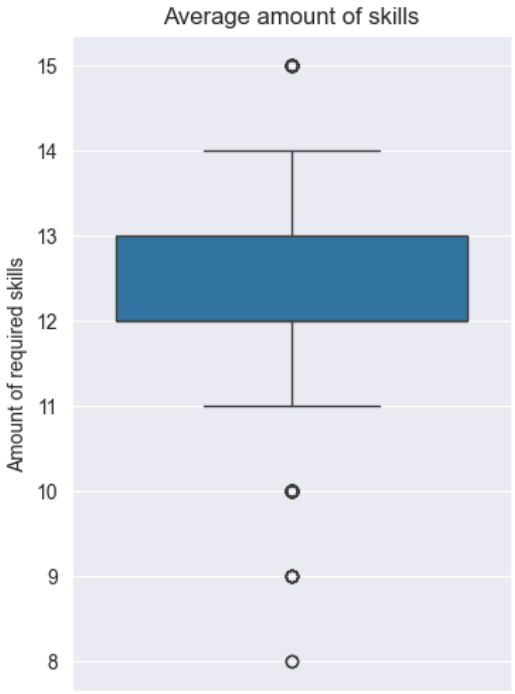
- balance_map

```
balance_map = {
    'vue .js' : [(20, 'vue .js'), (30, 'node .js'), (20, 'jquery'), (20, 'express'), (10, 'next .js')],
    'redis' : [(40, 'oracle'), (30, 'mysql'), (20, 'redis'), (10, 'postgresql')],
    'react' : [(80, 'react'), (20, 'next .js')],
    'mongodb':[(50, 'mongodb'), (50, 'sqlite')]
}

imply_map = {
    'javascript': [(5,'svelte'),(10,'electron')],
    'css' : [(100, 'html')],
    'html' : [(100, 'css'),(50,'php')],
    'cmake' : [(100, 'c'), (70, 'c++')],
    'numpy' : [(100, 'pandas'), (50, 'scikit learn')],
    'rust' : [(100, 'cargo')],
    'cargo' : [(100, 'rust')],
    'scikit learn' : [(50, 'tensorflow')],
    'git' : [(10, 'travis')],
    'python' : [(50, 'pip'),(40,'fastapi'),(10,'rabbitmq'),(5,'opencv'),(5,'qt')],
    'pip' : [(100, 'python')],
    'gcc' : [(70, 'c'), (70, 'c++')],
    'java' : [(10,'kafka')],
    'c':[(5,'terraform'), (15,'ios')],
    '.net':[(100,'nuget')],
    'vue .js':[(50,'vite')],
    'react':[(50,'vite')],
    'c++':[(5,'opengl')],
    'c#':[(50,'unity 3d')]
}

add_map = [(80, 'git'), (45, 'python'), (5,'dynamodb'), (5,'cloudflare'),(5,'vmware'),(30,'java'),(30,'linux'),]
```

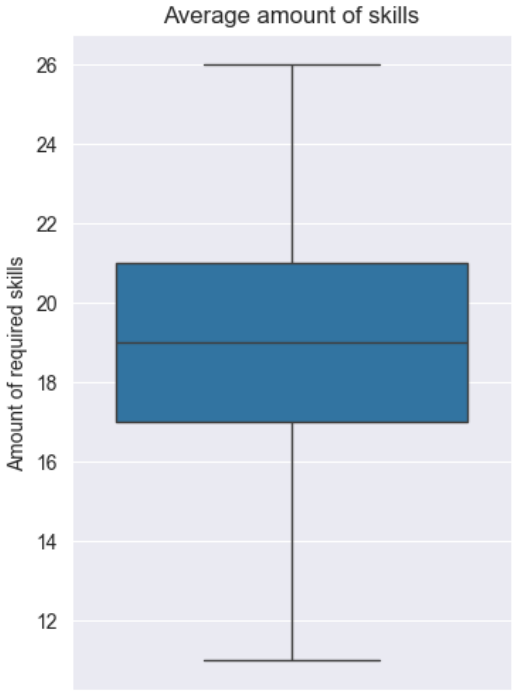
Dopo l'applicazione del bilanciamento, il risultato è una frequenza di apparizioni delle skill più bilanciata, come mostra il seguente grafico:



Come conseguenza di queste operazioni, anche il numero di competenze presenti per profilo è cambiato. Segue boxplot che misura il numero di competenze prima del bilanciamento:



E lo stesso boxplot dopo il bilanciamento:



Come si evince dai grafici, si è passati da una media di circa 12,5 ad una media di 19, oltre ad aver alzato il valore minimo da 8 a 11 ed il massimo da 15 a 26.

Anteprima dei profili degli sviluppatori dopo data balancing e unione con dataset contenente informazioni personali sintetiche.

	name	surname	email	password	bio	location	languages	skills
0	Mariana	Puglisi	mariana.puglisi@gmail.com	MarianaPuglisi123_	This is an automatically generated profile for...	Peio, Italy	{spagnolo, inglese, italiano}	{0, 1, 2, 6, 10, 19, 26, 36, 44, 45, 47, 54, 7...
1	Edelmiro	Carrera	edelmiro.carrera@yahoo.com	EdelmiroCarrera123_	This is an automatically generated profile for...	Murcia, Spain	{spagnolo, inglese}	{0, 64, 2, 100, 75, 76, 50, 19, 20, 52, 58, 59...
2	Rosario	Aloisio	rosario.aloisio@gmail.com	RosarioAloisio123_	This is an automatically generated profile for...	Arcugnano, Italy	{francese, italiano}	{0, 1, 2, 4, 6, 8, 9, 18, 19, 32, 41, 42, 44, ...
3	Roger	White	roger.white@gmail.com	RogerWhite123_	This is an automatically generated profile for...	Westminster, United Kingdom	{inglese}	{64, 97, 34, 2, 100, 67, 6, 58, 104, 73, 5, 75...
4	Eugenio	Estévez	eugenio.estévez@outlook.com	EugenioEstévez123_	This is an automatically generated profile for...	Córdoba, Spain	{spagnolo}	{0, 1, 99, 100, 6, 8, 42, 44, 45, 80, 82, 19, ...
...	...	...	...	...	...	...	...	...
504	María	Gracia	maría.gracia@gmail.com	MaríaGracia123_	This is an automatically generated profile for...	Zamora, Spain	{spagnolo, francese}	{0, 1, 99, 100, 37, 6, 104, 41, 74, 42, 44, 10...
505	Graziella	Capuana	graziella.capuana@gmail.com	GraziellaCapuana123_	This is an automatically generated profile for...	Zuppino, Italy	{inglese, italiano}	{0, 1, 2, 99, 100, 38, 6, 104, 72, 74, 75, 44,...
506	Noa	Riba	noa.riba@outlook.com	NoaRiba123_	This is an automatically generated profile for...	Zamora, Spain	{spagnolo}	{0, 1, 2, 10, 19, 21, 30, 50, 52, 54, 56, 58, ...
507	Sancho	Águila	sancho.águila@gmail.com	SanchoÁguila123_	This is an automatically generated profile for...	Tarragona, Spain	{spagnolo, inglese}	{0, 1, 2, 4, 6, 8, 9, 18, 19, 30, 39, 41, 42, ...
508	Estrella	Lillo	estrella.lillo@yahoo.com	EstrellaLillo123_	This is an automatically generated profile for...	Valencia, Spain	{spagnolo, inglese}	{0, 97, 2, 64, 100, 67, 32, 5, 104, 73, 58, 75...

4.2.3 Feature Construction

Al fine di poter utilizzare particolari implementazioni di algoritmi (approfondimento nella sezione 5.2) è stato deciso di costruire a partire dalle liste di competenze (sia in Offerte che in Profili di Sviluppatore) una matrice delle distanze. La matrice della distanze viene ricavata calcolando la **Distanza di Jaccard**, in particolare è stata scelta per i seguenti motivi:

- E' particolarmente utile per i dati categorici, come nel caso delle Competenze.
- In quanto metrica intrinseca può essere utilizzata senza *Ground Truth* (conoscenza empirica della soluzione migliore) e quindi adatta al problema.
- Questa Metrica è *scale invariant*, cioè non dipende dalla grandezza assoluta degli insiemi che confronta ma solo dalla intersezione tra i due, rendendola un'ottima metrica per il problema in esame visto che gli insiemi di competenze possono variare molto in dimensione.

Segue un esempio della matrice delle distanze calcolata sul dataset delle Offerte:

	0	1	2	3	4	5	6	7	8	9	...	472	473	474	475	476
0	0.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.666667	1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000
1	1.0	0.000000	0.888889	0.400000	0.846154	0.833333	0.600000	1.000000	1.000000	0.833333	...	1.000000	0.888889	0.777778	0.933333	0.833333
2	1.0	0.888889	0.000000	1.000000	0.636364	0.833333	0.833333	1.000000	0.857143	0.833333	...	0.833333	0.888889	1.000000	0.857143	1.000000
3	1.0	0.400000	1.000000	0.000000	1.000000	1.000000	0.750000	1.000000	1.000000	1.000000	...	1.000000	0.857143	0.714286	0.923077	0.750000
4	1.0	0.846154	0.636364	1.000000	0.000000	0.800000	0.909091	1.000000	0.700000	0.909091	...	0.800000	0.846154	1.000000	0.687500	1.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
477	1.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000
478	1.0	0.875000	1.000000	0.833333	1.000000	1.000000	0.800000	1.000000	1.000000	1.000000	...	1.000000	0.875000	0.750000	0.846154	0.500000
479	1.0	0.857143	0.857143	1.000000	0.916667	0.750000	0.750000	1.000000	1.000000	0.750000	...	1.000000	1.000000	0.714286	0.923077	0.750000
480	1.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000
481	1.0	1.000000	1.000000	1.000000	0.900000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	0.800000	1.000000	0.909091	1.000000

482 rows × 482 columns

Viene infine applicata **Riduzione della Dimensionalità** tramite l'utilizzo dell'algoritmo **Principal Component Analysis**. L'algoritmo **PCA** si applica su dati ad elevata dimensionalità e ricava un nuovo spazio (con dimensionalità inferiore) in cui le variabili (chiamate componenti principali) sono frutto di una combinazione lineare. L'algoritmo **PCA** restituisce un insieme dei dati facilmente utilizzabile per via della ridotta dimensionalità, garantendo una **perdita di informazione minima**. Le conseguenze negative dell'applicazione di PCA sono però la **riduzione di Explainability del Modello**. La scelta di utilizzare PCA nonostante questo problema è argomentata nella sezione 5.2.3)

5 Data Modeling

5.1 Scelta della Metrica

E' stato scelto di utilizzare come metrica la **Distanza Euclidea** per via della sua facilità applicativa, specialmente sulla Matrice delle Distanze ricavata durante la Feature Construction.

5.2 Scelta dell'algoritmo

La scelta dell'algoritmo di clustering ricade in due macro categorie:

- Clustering partizionale: un tipo di algoritmi di clustering che partiziona gli elementi dell'insieme di partenza al fine di produrre cluster. I pro di questa strategia sono la semplicità, l'efficienza e la scalabilità degli algoritmi, i punti a sfavore sono la necessità di specificare il numero di cluster da generare e non riuscire a gestire forme di cluster irregolari, preferendo dati di partenza ben separati e con forme semplici.
- Clustering gerarchico: questo tipo di clustering mira a costruire strutture ad albero unendo o dividendo i dati di partenza in maniera gerarchica. Il maggior punto di forza è quello di riuscire a partizionare con successo forme complesse o cluster che si sovrappongono, inoltre non è necessario specificare un numero di cluster.

Al fine di valutare la migliore soluzione, si è deciso di comparare diversi algoritmi di clustering. Sono stati selezionati l'algoritmo **K-Means** per rappresentare la strategia di clustering partizionale e l'algoritmo **BIRCH** per rappresentare la strategia di clustering gerarchico. La metrica usata per misurare l'ottimalità dell'algoritmo è il **Silhouette Score**, essendo più informativa come metrica rispetto ad altre come l'**Elbow Point**.
Per interpretare i risultati del Silhouette Score è importante notare che i valori variano tra -1 e +1 e hanno i seguenti significati:

- 1 = gli Item sono assegnati a un cluster specifico e i cluster sono facilmente distinguibili.
- 0 = i cluster si sovrappongono.
- -1 = gli Item vengono assegnati a cluster errati.

Gli algoritmi saranno testati sul dataset delle Offerte.

5.2.1 Algoritmo K-Means

L'algoritmo K-Means è stato scelto per la sua semplicità applicativa. E' stato applicato sulla matrice delle distanze sia con che senza riduzione della dimensionalità tramite PCA, è stato provato per diversi valori di K per cui è stato misurato il Silhouette Score.(Tabella1).

5.2.2 Algoritmo BIRCH

L'algoritmo BIRCH (*Balanced Iterative Reducing and Clustering using Hierarchies*) è un'algoritmo di tipo gerarchico scelto per la sua efficace gestione della risorse della macchina e la capacità di gestire flussi di dati incrementali. È stato valutato tramite tecniche di **Hyperparameter tuning**, in particolare è utilizzando il **GridSearch** di *Sklearn.*(Tabella 2).

K Value	Silhouette Score (no PCA)	Silhouette Score (PCA)
3	0.012	0.016
4	0.016	0.005
5	0.020	0.002
6	0.008	0.010

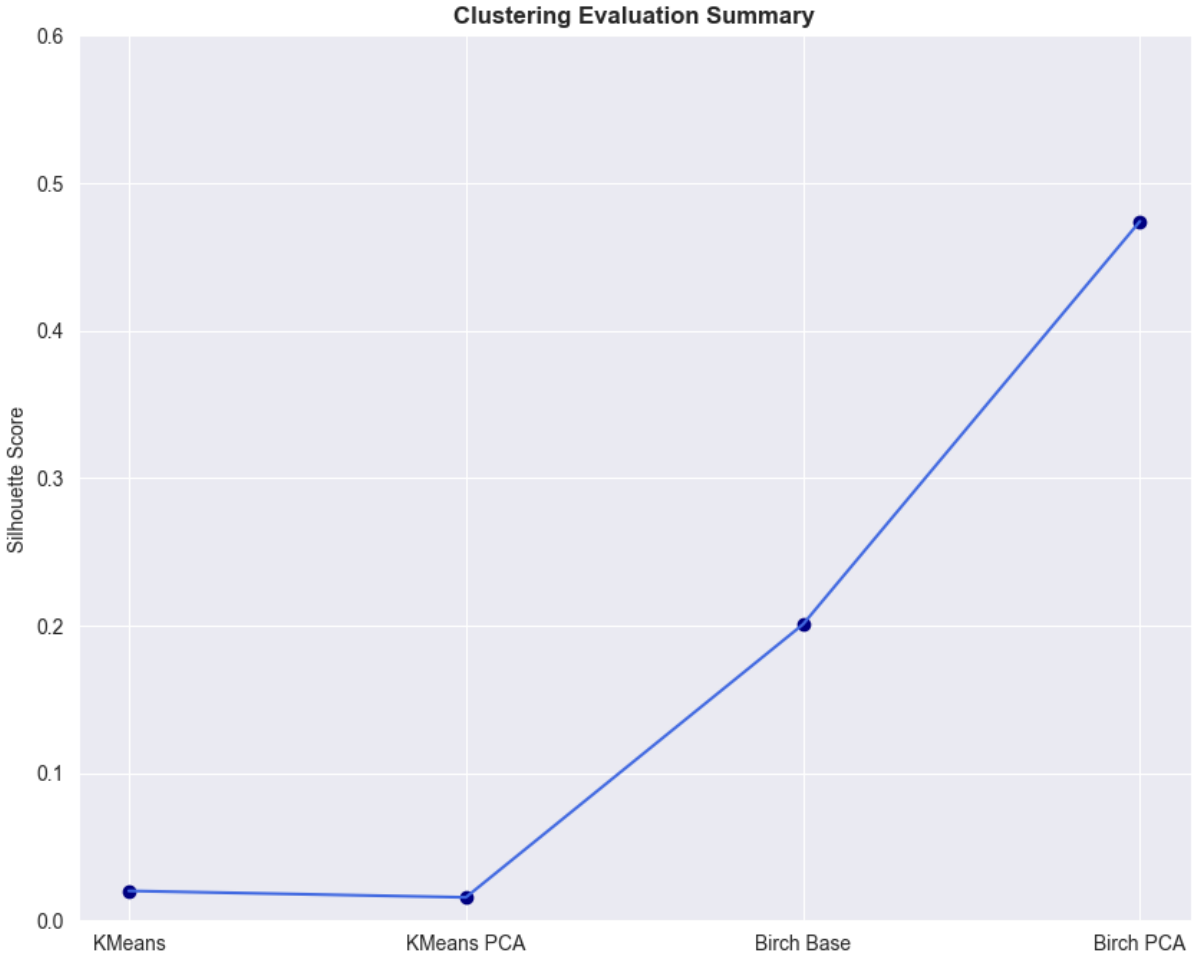
Table 1: Silhouette Score for K-Means Clustering Algorithm

Silhouette Score (no PCA)	Silhouette Score (PCA)
0.20	0.47

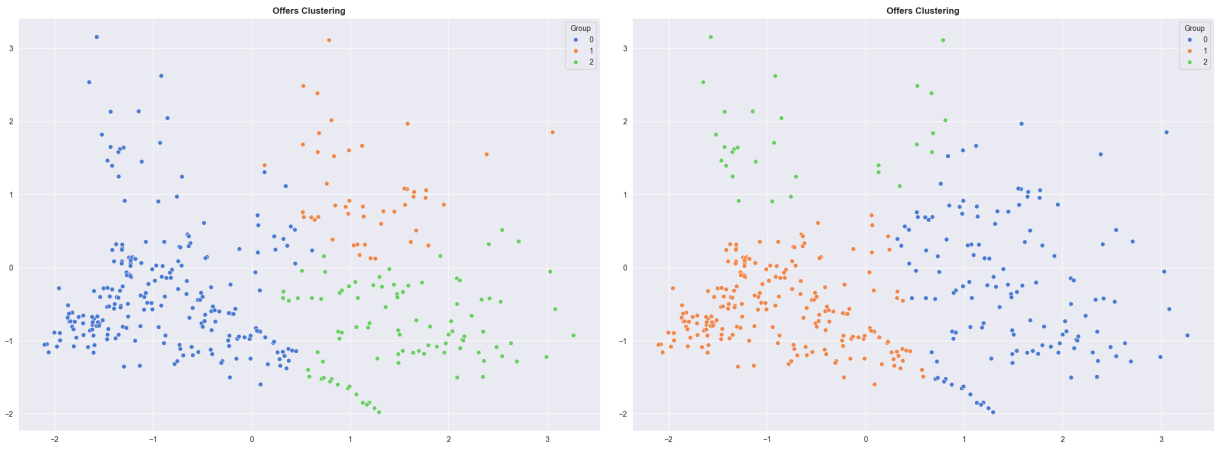
Table 2: Silhouette Score for BIRCH Algorithm

5.2.3 Confronto tra modelli

Il confronto tra modelli rivela la netta superiorità dell'**algoritmo BIRCH** rispetto a quello K-Means, portandoci a **scegliere questo come algoritmo di clustering per il modello**. Inoltre viene dimostrata l'efficacia dell'uso di Dimensionality Reduction nella Matrice delle Distanze, con il K-Means PCA che restituisce risultati di poco peggiori, mentre il BIRCH PCA mostra un miglioramento di ben 0.27 punti, cioè un **incremento del 135%**. Visto il notevole aumento di performance derivate dall'uso di PCA, **si è scelto di utilizzarlo nonostante questo penalizzi l'explainability del modello**. Di seguito sono riportati grafici che riassumono la differenza tra gli algoritmi usando come metrica il **Silhouette Score**:



Infine, si mostrano le differenze di risultati tra i due algoritmi (entrambi utilizzanti PCA) sotto forma grafica:



A sinistra il clustering realizzato utilizzando l'algoritmo **BIRCH**, a destra il clustering realizzato con un algoritmo **3-Means**.

6 Deployment

Il modello è parte del sistema "Turing Careers", sviluppato per il corso di Ingegneria del Software (maggiori dettagli presenti nella sezione 2.1). TC Reccomender AI rappresenta quindi il servizio di Ricerca e Raccomandazione del sistema e verrà deployato utilizzando Uvicorn Web Server per la comunicazione con il sistema "Core". In particolare, tramite l'utilizzo di Fast API, viene realizzata una API che astrae l'intero sottosistema e permette al "Core System" di comunicare tramite un endpoint HTTP, scambiandosi informazioni in formato JSON.

7 Ulteriori Informazioni

7.1 Linee Guida per Web Scraping

Di seguito è riportata una serie di linee guida che si è scelto di seguire durante lo scraping del dataset delle offerte di lavoro:

1. Agire da buon cittadino digitale: non sovraccaricare il sito dal quale si fa scraping.
2. Disponibilità pubblica dei dati: i dati raccolti devono essere accessibili pubblicamente senza registrazione.
3. Informazioni che non violano copyright: la raccolta dei dati non deve violare il diritto d'autore dell'entità giuridica che possiede il sito.
4. I dati non devono essere di natura personale: è illegale secondo il General Data Protection Regulation (**GDPR**) effettuare scraping di dati sensibili come:
    - Nome e Cognome.
    - Indirizzi di Posta Elettronica di privati individui.
    - Codice Fiscale.
    - Indirizzo di Residenza.
    - Informazioni di Impiego.
    - Numero di Telefono.
    - etc.
5. I dati raccolti non devono essere usati per danneggiare il sito dal quale si raccolgono: i dati raccolti vanno usati in modo *trasformativo*, cioè per creare nuovi prodotti e non cannibalizzare quote di mercato dal sito da cui si fa scraping.