Designing a path for career conversion

Name – Prathamesh Karve CMPE-256 Individual Project Report Student ID – 013850928

Introduction

Nowadays, many people want to switch from a software developer role to a data scientist or data engineer role. Change of career requires learning new skills. Online courses and degrees are a convenient and efficient way of earning these skills.

There are many profiles of people on LinkedIn who have made such career moves. Finding out such profiles and analyzing their skills, online courses taken, and degrees earned can help in drafting a path for similar person in making their own career transition.

Aim of this project is to develop a system which provides personalized recommendations for –

1. Skills to learn
2. Online courses to take
3. Degrees to acquire

The systems contains data from 200 LinkedIn profiles of people mostly from the Tech industry.
Some of the job titles of the profiles in dataset are Student, Backend Engineer, Engineering Manager and Data Scientist.
To tailor these recommendations for the user, the system asks the user to create a profile with current skills, courses taken till now and degrees earned.
Based on this data, the system rates the user's profile from "Beginner" to "All-Star" status and recommends next steps for career conversion.

Data collection

LinkedIn does not provide APIs for collection of data from multiple profiles.
Custom data crawler is required to collect data from profiles.

LinkedIn crawler NodeJS script – https://github.com/linkedtales/scrapedin-linkedin-crawler
Edited the code from this repo to collect data from LinkedIn profiles similar to me (target user).
For first prototype – Data from **660 LinkedIn profiles** collected.
For the final version, the same script was used to collect data for 200 users with varied profiles and experiences.

Designing a path for career conversion

Name – Prathamesh Karve CMPE-256 Individual Project Report Student ID – 013850928

Data Preparation

For code, please refer to the **Jupyter Notebook** attached to my submission.
For this first prototype, the project scope is to **recommend skills to learn for one individual profil**e.
The recommendations are done using collaborative filtering.

Data cleaning
Goal is to recommend skills in order to get 'Software Engineer Intern' job.
'headline' field of each profile is scanned and only those that have 'Intern' word are retained.
'skills' field is a JSON object consisting skill name and number of recommendations.

New dataframe is created that has skills as columns and users as rows.
Values of each cell are the number of recommendations that the user has received for that skill.

Feature extraction
The matrix generated is sparse, making it difficult to find correlation.
Only top 10 skills for each user profile are retained and other values are set to 0.
Now, remove all skills that have only 1 user. Such skills are anyways not useful for the RS.

Hybrid Recommendation System

Knowledge-based go no-go
For the mid-term review, the dataset was smaller and targeted only for a conversion from Student Assistant to Software Engineer Intern.
For the final project, the dataset contains 660 profiles from varied fields.
This required a filtering of profiles in the first stage before the collaborative filtering approach is used.
For this, I used the knowledge-based go no-go approach that filters profiles based on

Collaborative filtering
Similar profiles are found using correlation.
Skills of similar profiles that the target user doesn't already have are added to a new dataframe.
This dataframe is used to make final recommendation to target user.

Designing a path for career conversion

Name – Prathamesh Karve CMPE-256 Individual Project Report Student ID – 013850928

Evaluation

Top-10 skills were found for 1 user who is currently a Student in the mid-term review.
Metrics were – RMSE of 0.94 and MAE of 0.74 was calculated using a k-fold method, with k = 3.

For the final project, the recommendations for 100 users were considered and 3 metrics were used.
RMSE = 0.987
MAE = 0.7656 was calculated using a k-fold method, with k = 3.
Leave-one-out cross validation Hit rate – 0.75721

References

1 – https://ieeexplore.ieee.org/document/7916597
2 – https://www.sciencedirect.com/science/article/pii/S1110866515000341
3 – https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/
4 – https://www.npmjs.com/package/linkedin-scraper