F1 project report

Project: LISTINGS

Names: Kristjan Karl Pevgonen, Kauri Remm, Magnus Vaikla

Project repository: https://github.com/m2ger2020/listings/

Business understanding

Background

The Tallinn tech job market is dynamic and evolving. Understanding its patterns can benefit job seekers, HR departments, educators, and students.

Goals

The goal of this project is to identify current regular patterns and tendencies of the job market in the Tallinn tech job scene, with the motivation of using this information to increase the employability of someone looking for a job in the tech scene.

Business goals

- Identify hiring trends and popular fields in the tech sector.
- Determine optimal job-search periods for different roles (internship vs. senior positions).
- Highlight companies with repeated or long-term job postings, which would imply unsuccessful hiring.

Business Success criteria

- Clear understanding of recent tech scene hiring patterns.
- Ability to filter out and report information beneficial to the job seeker.
- Ability to report changes in hiring frequency for specific companies or fields
- Ability to reproduce the steps of the analysis for different time periods

Data-Mining goals

- Detect job market patterns (posting frequency, field popularity, company hiring trends).
- Identify short-term and long-term patterns in the timing of job listings, for example, if there are weekdays when more listings get posted, or if it is true that summer is a bad period for hiring.
- Identify top job boards.
- Correlate job listing timelines with hiring success.

Data-Mining Success criteria

- Data cleaned and processed
- Data categorized for analysis
- Data able to be visualized
- Accurate identification of trends and anomalies.

The potential benefits of this project are not limited to just someone looking for a job, as the knowledge could also be useful for improving the hiring process (i.e useful to the human resources departments of the companies discussed in this project), teachers of IT departments in universities (i.e people responsible for educating those who will be looking for jobs), and last but not least students who would like to know what to expect from the job market.

Current situation

The original data is scraped from https://techscene.ee/, which is an Estonian-made website that aggregates job listings in the tech job scene. The original data contains information about job listings and their details. The scraped data covers a larger part of the year and contains a fairly large number of listings.
The constraints of making assumptions based on the gathered data mainly comes from not knowing the definitive success of any given listing and that we have no details on the people seeking the jobs, meaning that we will likely need to presume or predict some information about the seeker and the successfulness of the listing.
The rather small amount of features of our data may give rise to contingencies, where we might not be able to always accurately predict valuable data about some given metric that is needed.
In terms of time the data gathering was done over the period of the gathered data and regarding resources it was done using a lightweight Python script.

Data understanding

Gathering data

The data has been gathered by means of a Python script that scrapes the listings from the website using Google Chrome and then inserts them to a local database using PostgreSQL.
The data was converted into a .csv file for analysis.
The python script was run from 2024-03-26 to 2024-11-10 every day, except from 2024-03-27 to 2024-04-01 (included) and 2024-09-06.

Data

Each row in the data is a series of fields scraped from a single job listing. The fields are:

- Company name
- Job name
- Job listing website URL

- Date of scraping
- Job domain

Since the original data doesn't have a date when the listing was posted or taken down, just dates when the listing was spotted, it is necessary to process the data. One way of compacting the data is to look at the first time the listing was scraped and the last time it was scraped, then assume it has been up between these dates. We can also check if the listing was not up on a date in between first sighting and last sighting, so we can confidently say that it has been reposted. Listings that might have been posted multiple times should be normalized within the dataset using the aforementioned information and a logical set of rules, to classify whether or not the listing is a repost.

How do we know that different rows in the dataset are for the same job position? Well, usually the job id is in the web URL, so if the URLs are the same, it is safe to say that they are for the same position. Some companies use the same page for different jobs, there we can look at the job name and domain.

We also don't have data on when more people are looking for a job and if someone was hired for a particular job, so we have to assume that when the job listing was taken down quickly the person was found. Some companies want to hire multiple people and keep their listings up even if they hired someone, so we don't know anything in these cases.

More manual cleaning is needed also, as there are some weird job listings, for example a job named "facebook" and some lowercase and uppercase differences for the same job id.

Verifying data quality

Selection criteria for choosing data for this project are lax, since we simply accept everything scraped by the script; the website doesn't contain much more information than exactly what we need. I couldn't find any information about the website where the data was scraped from. Gathering from information found in articles and business information websites about Tech Scene, we could find that it is a website that serves to be a competitor to other job listing sites in Estonia, with the specific focus on providing listings for jobs in technology. The site doesn't make any profit, judging by publicly provided company information, meaning that there should not be any profit-related bias present.

Exploring the data

Most of the fields in the data are strings, and since the companies that post listings have no exact agreed-upon terminology regarding fields and job titles, there will have to be some processing to combine job titles and fields into one categorical value. For instance, "Tarkvaraarendaja (Java)" and "Senior Java Software Developer" have the same meaning. This processing can be done by grouping all unique values and defining categorical values to combine them, e.g by combining the two aforementioned job titles under "Java developer". Another way to process this data is to define a new field, "tools", and include these two listings with the value "Java" and the job title "developer".

Some job titles are very company-specific, for example "Senior Scala Engineer, Courier Group" from Bolt in March 2024. In the general analysis of the whole data, company-specific job titles will have to be removed during processing and combined into one value.

Since we are interested chiefly in the *times* at which listings are posted and removed, an insightful way to visualize the data would be to plot out listings as overlapping bars on a timeline. The bars on this timeline could also be color-coded according to fields and companies. One way of achieving this using matplotlib is outlined in this blog post. This necessitates processing the data so that consecutive instances of the same job listing are combined into a single entry in the data, containing both the beginning and end times. As for comparing the popularity of different companies, fields or websites, it would be useful to plot out histograms of their respective frequencies over a certain time period.
The data contains a few entries that have missing fields, and there are also some brief time periods from which we have no data. These problems will have to be accounted for in processing the data.

Planning your project

	Kauri	Kristjan	Magnus
Planning/reporting	4 h	3 h	5 h
Formatting data	7 h	7 h	9 h
Cleaning data	4 h	7 h	10 h
Making conclusions	10 h	10 h	4 h
Postering (in Canva)	4 h	3 h	2 h

Tools

Python

- Pandas for processing data
- Matplotlib for plotting
- ChatGPT for helping with Python
- Selenium, webdriver_manager for data scraping

Canva for making the poster