Design Doc

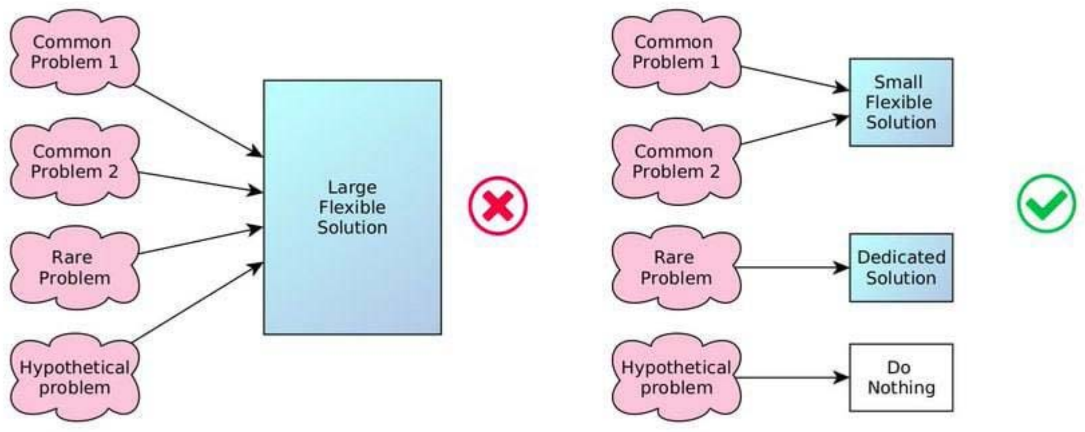
Mark of changed but not reviewed text

Mark for text to be changed

Introduction

Project focuses on the area of technical **skills** in Data Science, actual **vacancies** and **their relationships**.

Despite the similar needs for different areas we'll focus just on Data Science. Each area has it's own specifics and audience. It's better to make something small and specialized than trying to cover everything at once.



Needs and problems

Let's consider a **student** who wants to try himself in a new area. Or an **applicant** who is looking for a new job. They both need to navigate what skills will be required, how much they are in demand and how to get them.

The employer is interested in improving skills of its employees and encourages the completion of advanced training courses. **The HR manager** does not necessarily know the technical skills, but he must know which training course will bring more benefit to the organization. What skills are needed, what skills you already have, and what needs to be improved.

For a **specialist** in the modern world, education does not end at the university or after receiving a doctor degree, the IT world is constantly changing and our future is in constant learning, all our lives.

We try not only to deepen in our narrow specialty, but also to move in breadth, mastering new areas and new specialties. It is important to move in the right direction so that the acquired skills form a puzzle into a full-fledged specialty, and this is especially important for beginners in the profession. For example, a person wants to do machine learning, learn math, algorithms, and apply new skills in C#. In this case, a lot of time can be lost until it becomes obvious that there is a lack of competence in the Python language. Another example, a student has completed a Data Science course and is faced with a choice of a narrower specialization. If he chooses a training course, then you need to correctly assess how the course will bring you closer to the goal, will affect the movement in the specialties or career. Conversely, in order to create a useful and cohesive course, authors should have a good understanding of what skills should be covered in it.

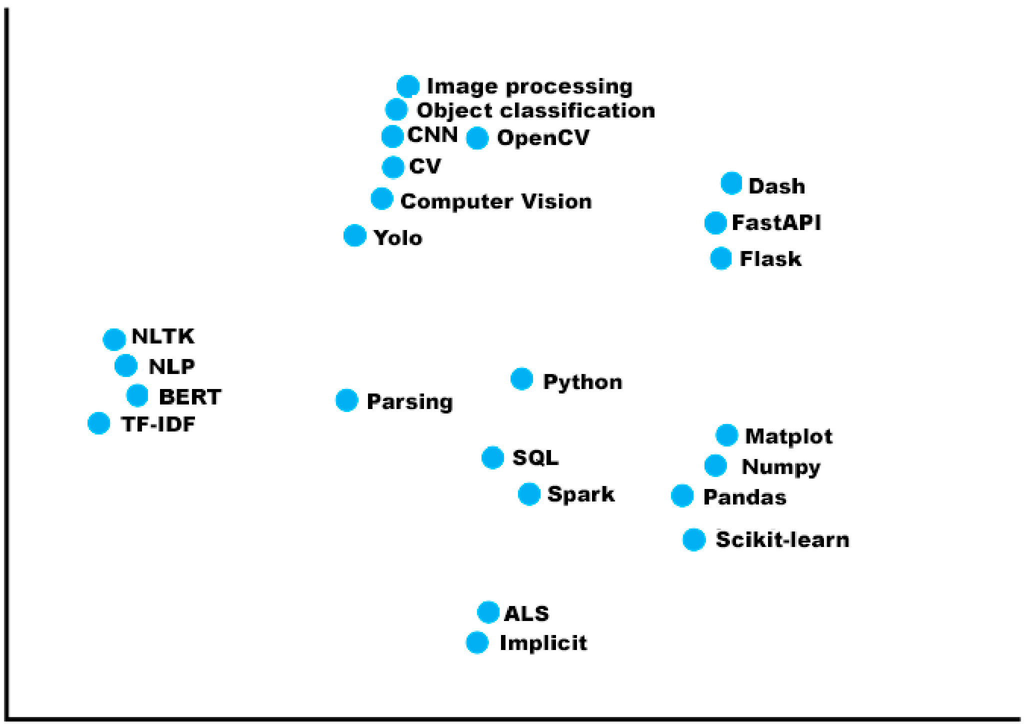
Product Manager and **Project Manager** may not understand DS, but they must lead their teams. It will require an understanding of the applicability of skills without firsthand knowledge of them.

Authors of training courses should make a research concerning what skills potential students need. If the course contains a lot of information, but the student will not be able to implement them at work, then the value of the course will be small.

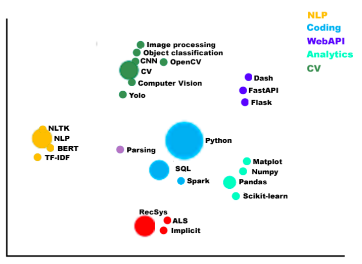
Solution

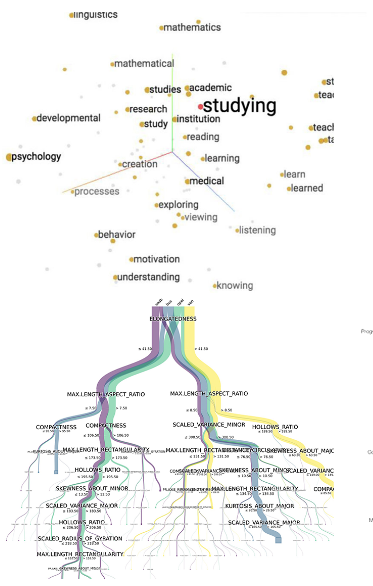
In all of the above cases, a tool or service is required that provides up-to-date information about skills and professions.

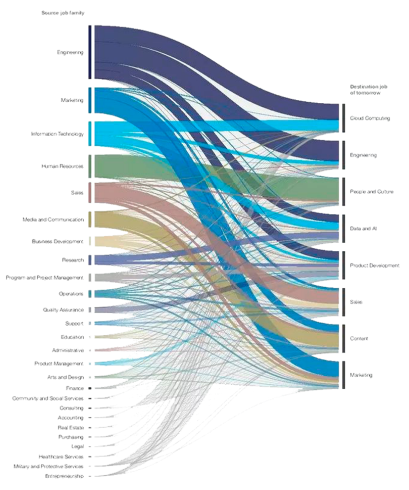
The user of the service, by going to a web page, gets the relationship, which skills are most often in demand along with others. Similarly with specialties and connections skill-specialty. Visualization and interactivity will be improved as we work on the service. We will call such a representation “a map”. An example of the first baseline version:

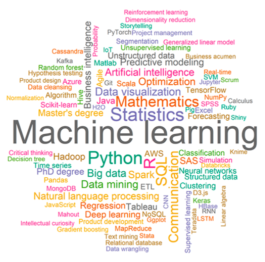


Visualization will be improved in the next versions, examples are below:

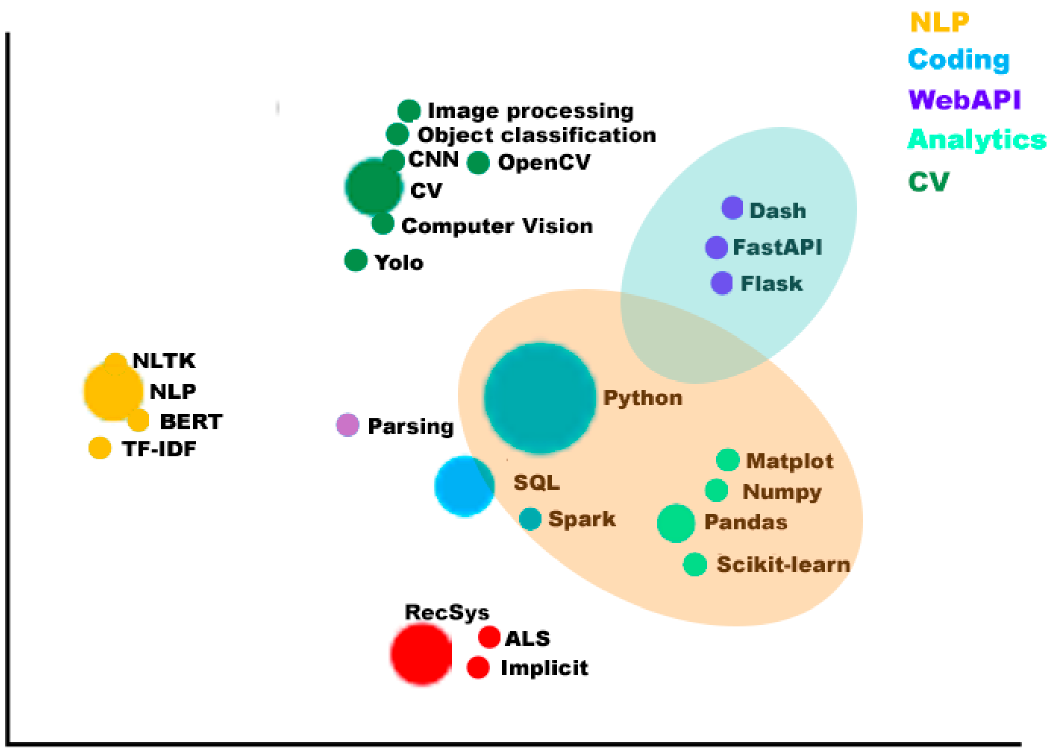




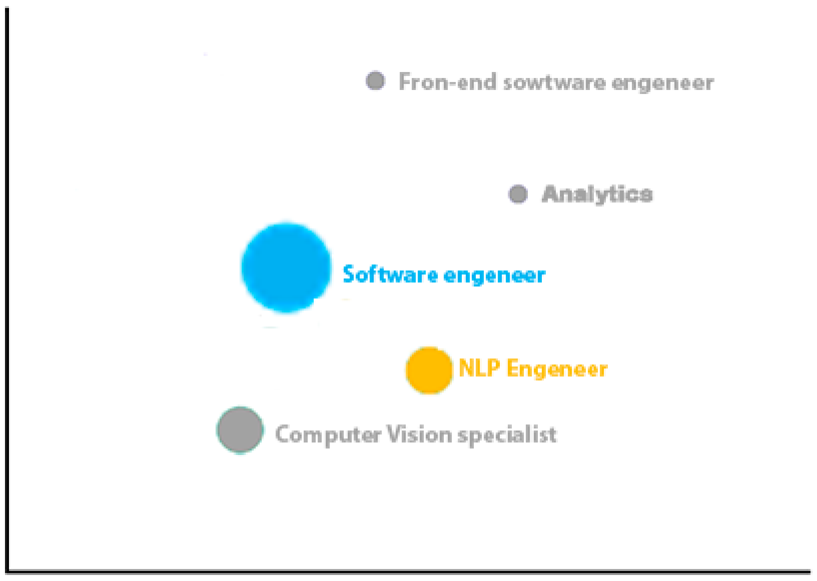




Information about training courses is added. On example it is represented as clouds.

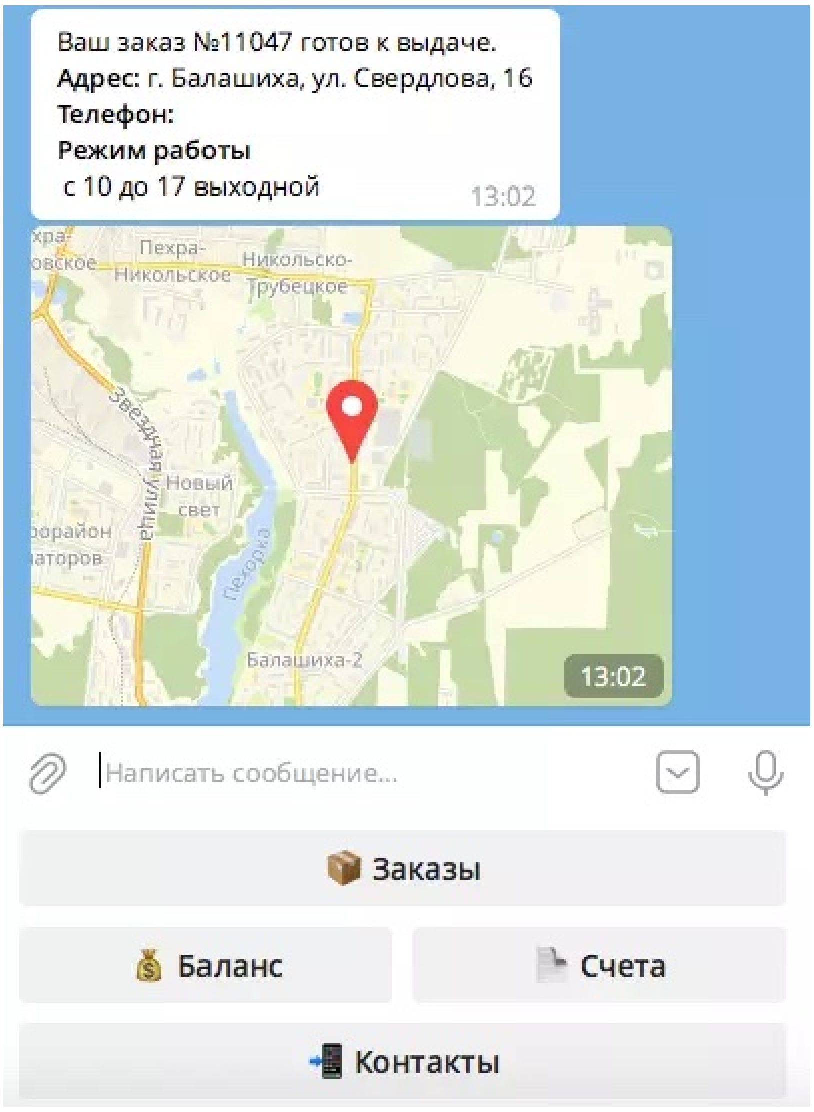


User can provide some information about his skills or choose from the offered and get recomendations on specialties and related skills. To make it clearer, look how the recommendation looks like in the diagram (recommended is in color, non-recommended is highlighted in gray):



As a development, interactivity will be added to the visual map, such as displaying skills when highlighting a profession and filtering.

User can use the service via Telegram. Bot for Telegram has the capabilities of messaging, charting, displaying the keyboard and embedding a special version of the web page. An example of a bot for placing orders with a similar interface:



Ваш заказ №11047 готов к выдаче.
Адрес: г. Балашиха, ул. Свердлова, 16
Телефон:
Режим работы
с 10 до 17 выходной 13:02

Thus, the functionality of visualizations and recommendations will be available in a user-friendly way, through a web page or Telegram. The service will help in training and career guidance, in the preparation of training courses.

Success Criteria

- the service should be used by: the number of requests / users of the bot; number of hits per page
- the bot/service is involved in the course selection process
- service contributes to the success of employment

Implementation

We offer a solution that provides users with up-to-date information about skills and specialties. To do this, vacancies are collected from popular sites and processed, which skills in which professions are most in demand. The end result is available to the user through an interactive visualization on a web page or via telegram chat. For initial career guidance, it is possible to indicate your skills and get recommendations for suitable specialties.

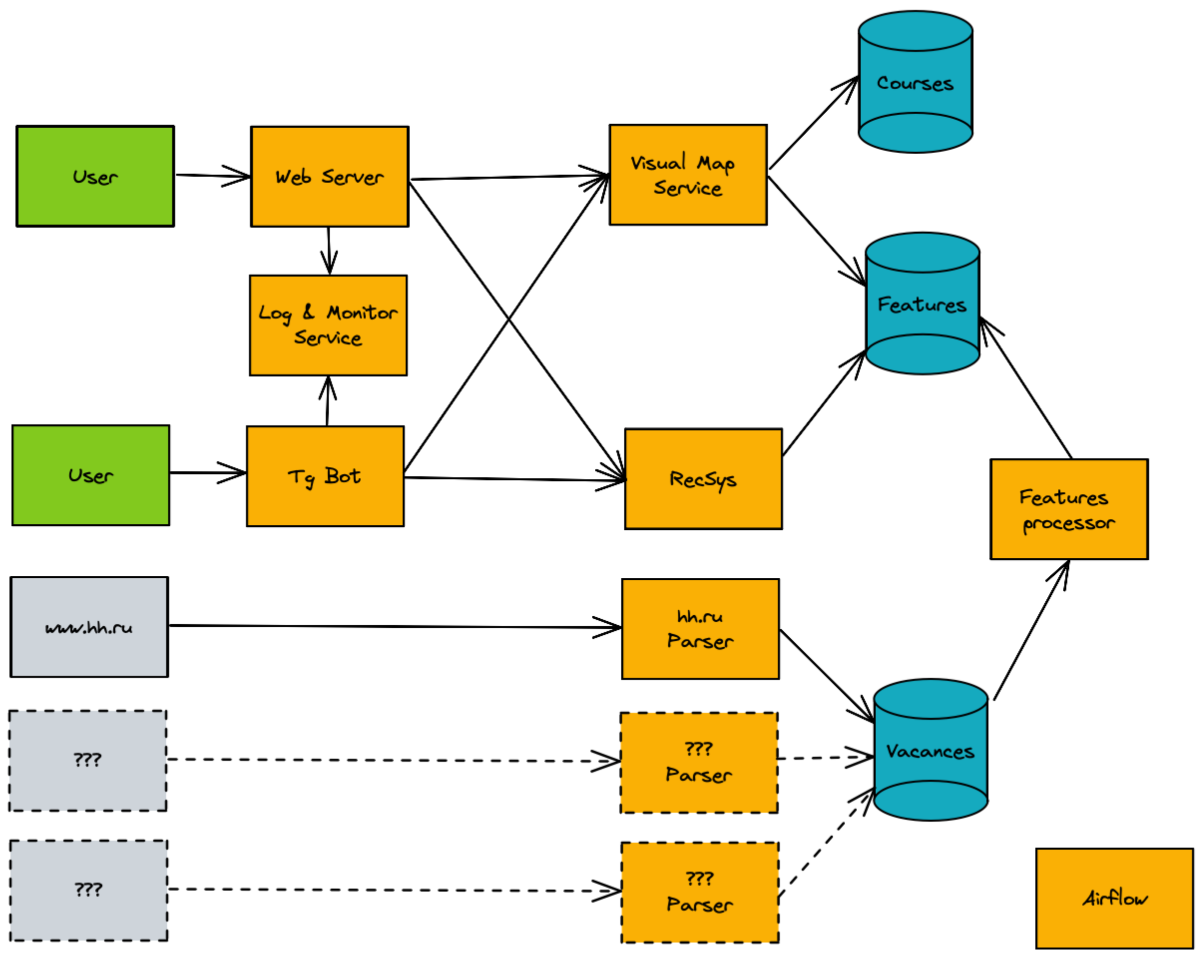
Functional and non-functional requirements

- **the relevance** of information, its constant updating, is permissible on a schedule or in the background
- **visualization** of the proximity of skills, specialties and their relationships in the form of an interactive object, which we will further call "a skills map",
- visualization on the map - what **training course** should bring
- the ability to send information about skills and get specialty **recommendations**
- the possibility of additional **analytics** can be considered in the future, such as highlighting groups of related skills, visualization of demand or the impact on average wages

---

High level design

The solution implementation consists of several components:



We use hh.ru and other sites as a source of vacancies. We collect new vacancies with some frequency, by default every day. After the stage of collecting new vacancies, we process current vacancies and form signs of skills, specialties and skills-specialties.

Back-end services `Visual Map Service` and `RecSys Service` are responsible for the formation of the map and recommendations. User can access the system both through a web application and a Telegram chat.

Visualization baselines

There are several figure types:

- 2D scatter plot
    - Add diversity from most simple to hard and complex
    - T-SNE / ALS / PCA / SVD

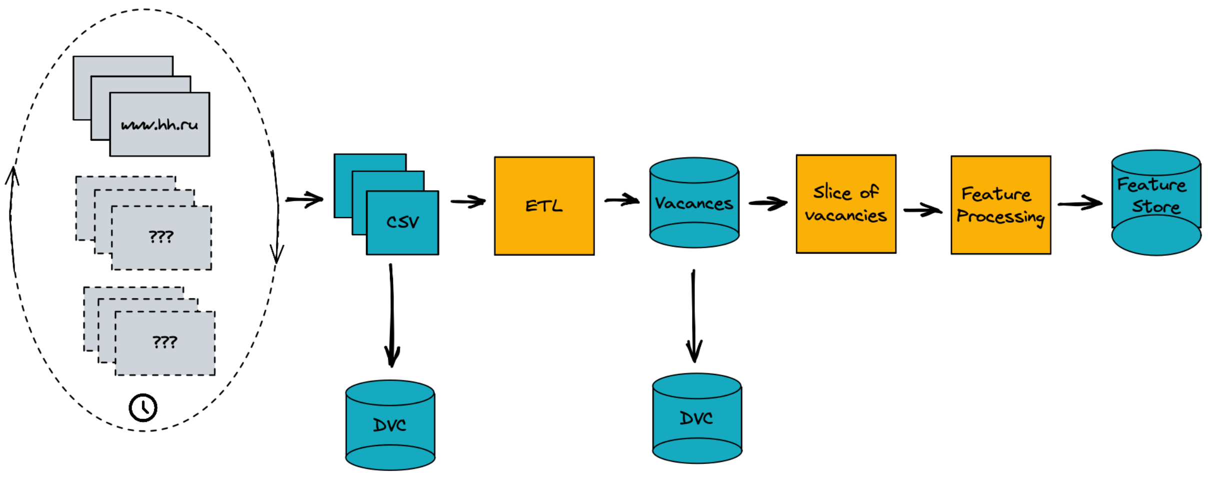
- Tree-like structure
    - WIP
- Flow-like structure
    - WIP

ML Design

Data collection

The sources of data are the websites where vacancies are published. At first it will be hh.ru, after that other sites may be parsed.

Entire data collection process is in the diagram:



Website **parsers** will need to bypass blocking, including making requests from different proxy servers. Thus, we have many processes that will be launched sequentially in order to save resources.

As a result of the work, the parsers create **CSV** files. Files may have duplicate entries due to making requests to the same site from different proxies. Therefore, they cannot be immediately stored in the database and require an additional preprocessing step.

We'll save intermediate CSV files in **DVC (Data Version Control)**. This can be useful for reproducibility of the next processing step and reprocessing if it changes.

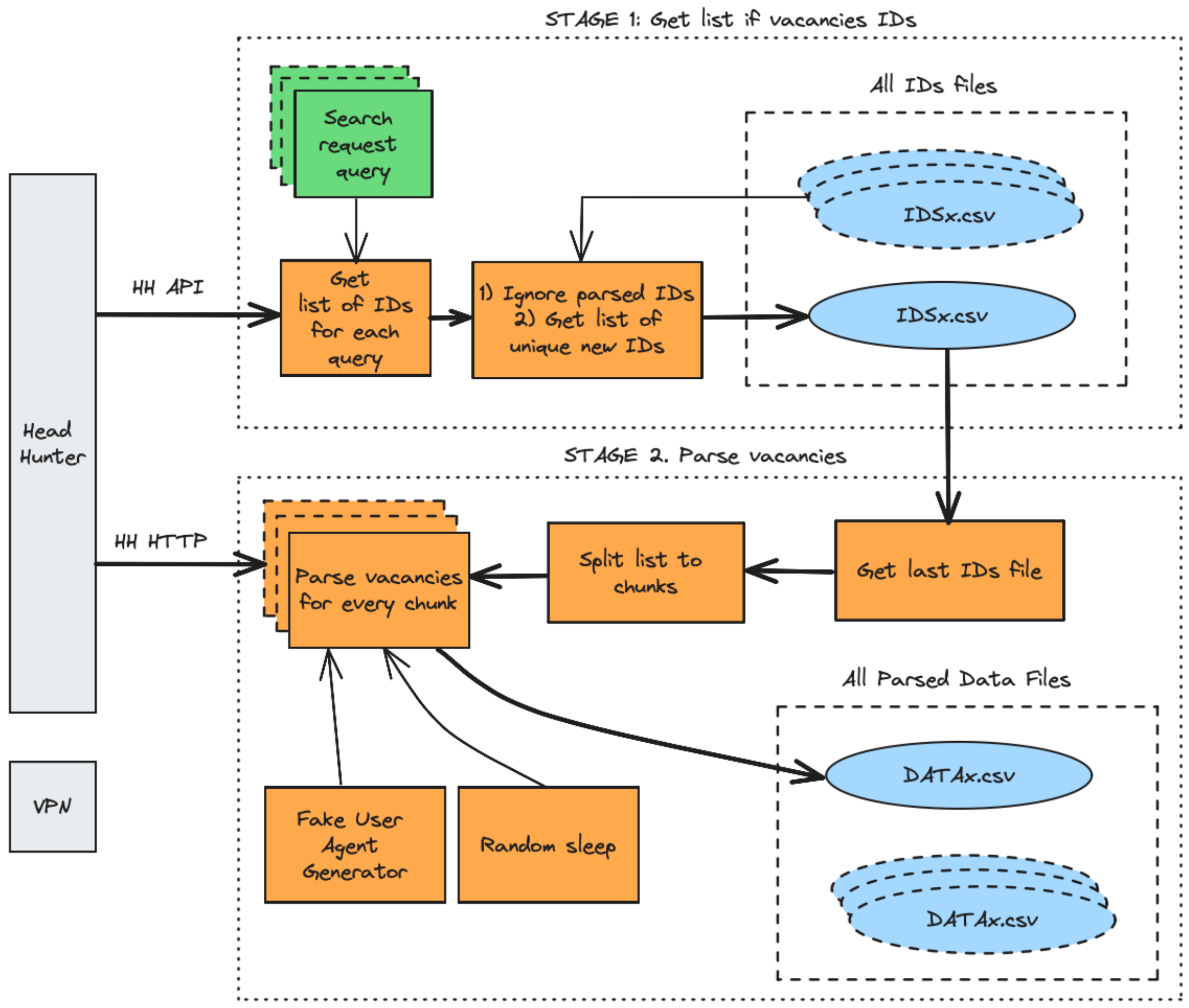
ETL (Extract, Transform, Load) gets files, cleans, removes duplicates, and loads into the **Vacances** database of the PostgreSQL database management system

For further processing, we will use the **Slice of vacancies** for the last month (the interval can be changed) and only for the DataScience area. This data slice is also saved in **DVC** for reproducibility

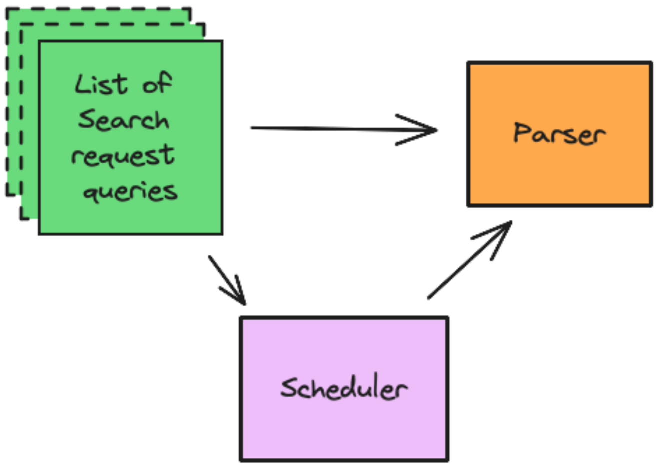
Feature Processing creates and processes features. This module also generates vector representations of skills and vacancies. Finished features are stored in the database tables **Feature Store**.

Parsing vacancies from hh.ru

See structure of parsing vacancies from hh.ru. Parsing is divided into two stages. Vacancies are parsed and stored in chunks. Already downloaded vacancies are not downloaded again. This approach allows to reduce the number of calls to the server and minimize reoperations in case of failure or blocking by the site.



Different search request queries allow to schedule it with individual periods. It reduce number of requests (and minimize time of processing) and decrease professions imbalance.



Preprocessing

Preparing vector representations

To visualize the map, we need to reduce the dimensionality of the data - create vector representations for skills and specialties, called **embeddings**.

Metrics

Как перейти от метрик из критериев успеха к “нашим” меткам?

Правильно ли, что в нашем случае оффлайн метрики построены на исторических данных, а онлайн будут те же самые метрики, только уже на новых данных?

The goal of building embeddings is to create such vectors that will have the smallest possible dimension, but at the same time have the following properties:

- similar objects are mapped to close embedding vectors
- the relationship between skill and specialty can be restored without significant loss of quality

In other words, initially we have a complete sparse matrix

`A` = `Skill` x `Speciality` of dimension `Ns` x `Nf`,

where `Ns` is the number of skills, `Nf` is the number of vacancies, and the values of the matrix elements are the frequency of inclusion of the i-th skill in the j-th specialty for all vacancies, optionally weighted by the number of the skill in the vacancy.

The task is to build such representations `Es` and `Ef`, which can be used to restore **relationships**, that is, the matrix `A`. Statement of the problem in this form also solves the problem of **proximity** of embeddings of similar objects. As a **metric** we will take a value showing how much it is possible to restore the original matrix A from the generated embeddings. The metric value can be from 0 to 1.

How will we calculate the **value of the metric**. We are not interested in all the values of matrix A, but only in the TOP-K skills for each specialty. Then our metric is:

$$metric = \sum_{i=1}^{N_f} NDCG@K(E_s \cdot E_f[i])$$

Here we have used the following property:

$$A[i, j] \approx E_s[i] \cdot E_f[j]$$

Target

In our scheme, there is no explicit target, the target is the matrix A, which needs to be best approximated. This approach is justified from the point of view of the visualization task: at this step, you need to make a representation that best approximates A

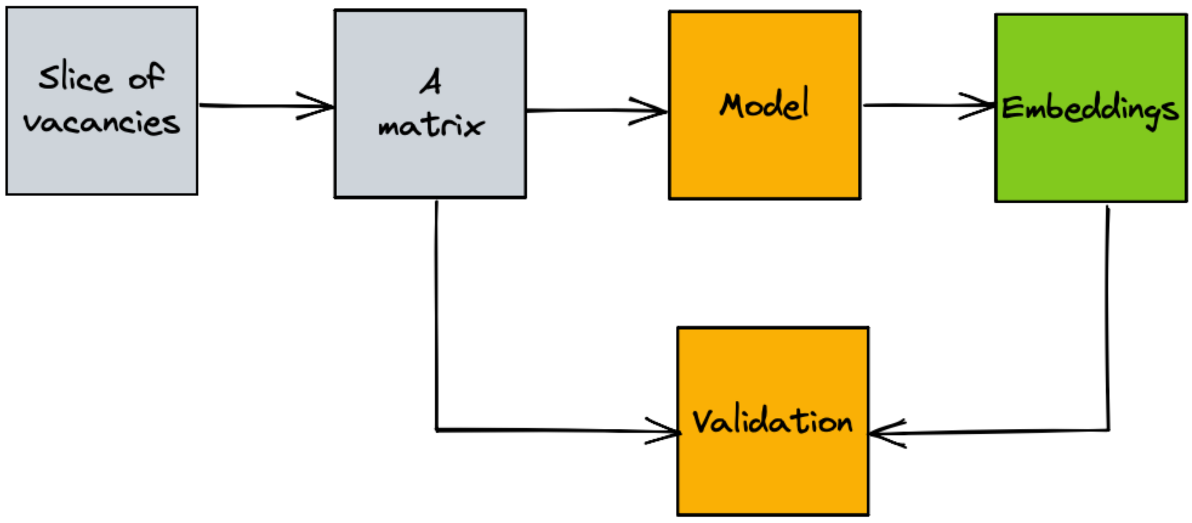
Validation

As part of the vector representations creation, we will validate on the initial matrix of connections between skills and specialties `A`, which corresponds to the task, goal and target

We will consider the recommender system separately below. It needs to be divided into training and test sets.

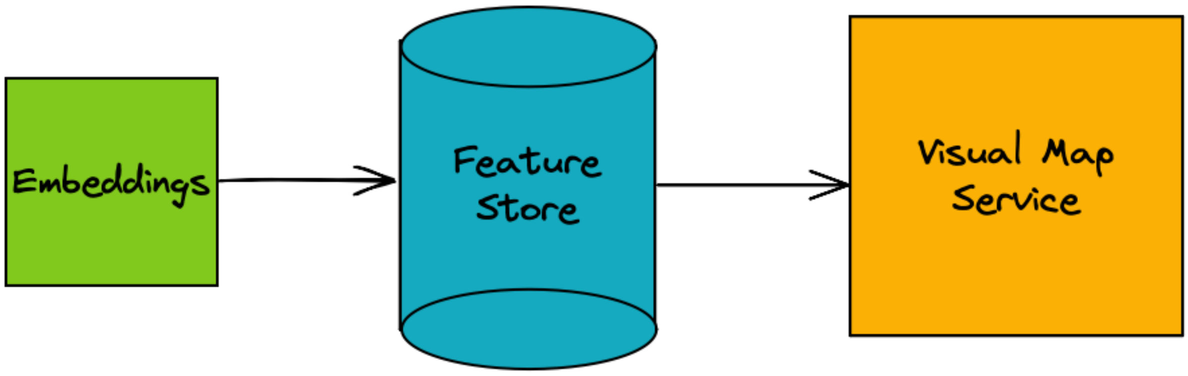
Learning pipeline

Based on the **slice of data** about vacancies, we build a sparse **relationship matrix A**, on which the **model** is trained, which forms **embeddings**. Relationship matrix and embeddings are used in **validation**



Пайплан инференса Pipeline inference

Already prepared embeddings are used for visualization:



Baselines models

- Alternating Least Square (ALS) Matrix Factorization

- ALS + normalization of the original matrix, multiplication by alpha, optimization of the dimension of vectors

---

Monitoring

WIP

- model metrics
- technical metrics (response time, RAM consumptions..)
- users metrics (traffic, dau, wau…)

Error Analysis

WIP

Deploy

- Implementation in docker containers
- Integration testing and configuration will be conducted, and scaling methods will be identified to ensure stable operation.
- The solution is deployed as a separate site on the KK servers. Requirements will be determined tater.

Future Steps

Recommendation system

WIP

The recommendation model triggered on request from the user. Receives information about skills as input and recommends specialties.

Target - a list of K recommendations of specialties, ranked by decreasing relationship with skills. How exactly the correlation coefficient will be calculated will be decided later.

Metric - NDCG@K for the recommendation list

Validation scheme - we split vacancies dataset into training and test subsamples with cross-validation and mixing.

Learning pipeline - training with validation, retraining on a full sample. The admissibility of using already prepared embeddings will be discussed later.

Inference pipeline - a list of skills with their level of proficiency as input, a list of recommendations as output

Baselines models

- handcrafted tree
- ALS
- other RecSys.

Visualization on the skill map - displaying the area that is covered by the user's skills

Interaction via **Telegramm Bot -** the user can enter his skills with their level of proficiency or answer a few questions.

Bot

WIP

Growth zones

- displaying area of users skills, providing recommendations for skill gaps
- adding education courses
- searching among applicants for potential course clients
- recommendation system for courses
- monetization, subscription system
- other professional areas