

JobGo

12.07.2018

Andong Ma, Hao Zheng, Yifan Li, Changsheng Yan

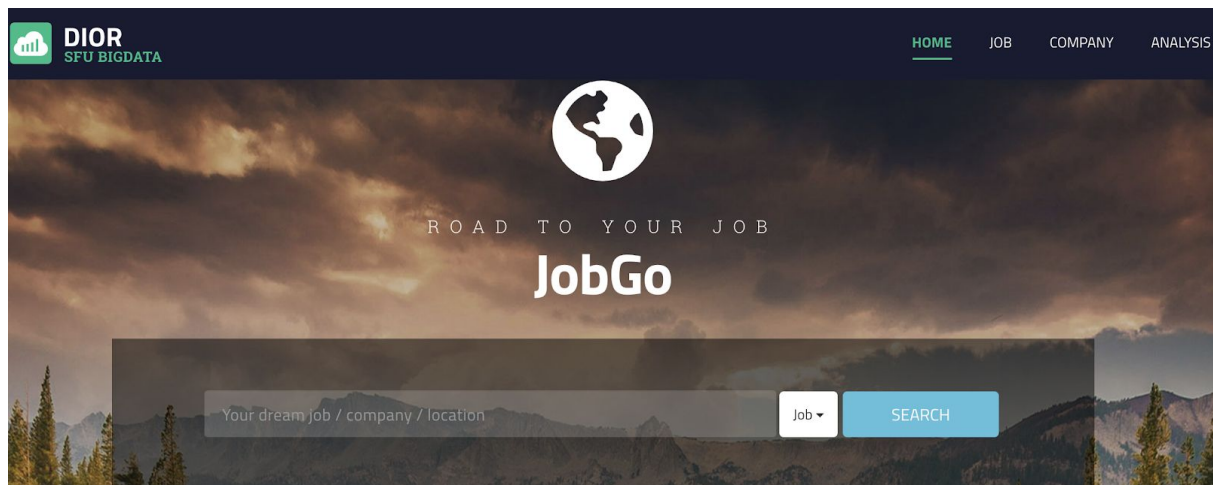
DIOR

Get Your Dream Job

Report

- Group Name: DIOR
- Project Name: JobGo (<http://nml-cloud-231.cs.sfu.ca:8080/>)
- Group Member: Andong Ma, Hao Zheng, Yifan Li, Changsheng Yan

Main Page:



Problem definition:

- The existing employment-related search websites (e.g. Glassdoor, Indeed) have enabled us for looking up information regarding job postings, company reviews, job salaries, etc. But if we want to gain a social connection insight of the employers and their employees, we need to search in another social networking website(e.g. LinkedIn).
- What we are trying to build is a job search platform that incorporates both sides of the information, bridging the needs of employment-related searching and understanding social networking. In this system, one can search for jobs in terms of company, location, title, interview and understand statistical insights(e.g. salary by job types, alumni by companies, job posting by locations, etc.) based on our data analysis.
- Another thing is that Glassdoor and Indeed have different parts of the job posting information. We want to provide our users with the most complete job hiring information combining all job positions distributed on each job searching website. So we integrate these information to help our users better make their decisions.

Methodology:

- Scrapy is used for web scraping as it is now the most successful and open source spider framework. It enables developers to easily send and receive requests and responses. With its distributed engine and customized item pipelines, we can easily scrape the information we want.
- BeautifulSoup is used during ETL because we mainly dealt with HTML files. It can parse HTML string into BeautifulSoup object and extract data from HTML tags.
- RabbitMQ is used to decouple the web server and data analysis tools, so as to increase the system flexibility. Also, it enables concurrency in our system and enhances the scalability and system robustness.
- Spark is used to ensure distributed computing power for future data expansion.
- Spring Boot is used for building our web server infrastructure and automatically taking care of various dependencies.
- MySQL is designed with a focus on the Web, Cloud and Big Data, which is able to store data in a more reliable way and doesn't tend to hog resources. Currently, our data size perfectly fits in MySQL. Plus, it can be scaled to MySQL cluster in the future.
- Bootstrap is a great standardized framework with all the basic styles and components that we needed, which is lightweight, customizable, support for all major browsers and designed with responsive structures and styles for different sized devices.

Problems:

Some of the challenges that we have encountered:

1. Acquisition of data by web scraping from three mainstream websites(Glassdoor, Indeed and LinkedIn).

Some websites do not have consistent HTML formatting style and some websites like LinkedIn is very hard to scrape. The response acquired by the web spider from LinkedIn was not HTML-formatted data but only some basic raw data so we cannot use the BeautifulSoup to parse the data directly. We wrote different regular expressions to parse it. And since there are many different subpages and subtags under same companies, we have to write different logics to jump from page to page. Furthermore, the Scrapy uses async request, which might lead to earlier requests to be handled later due to network traffic or server-side performance. Thus, in order to put each company's HTML pages in respective folders, we had to scrape companies one by one, not just simply put them in a loop.

Unfortunately, our LinkedIn account was blocked by the detection mechanism during web scraping so we have to manually save the HTML page by page for more data. But still, data obtained from LinkedIn is limited without the automatic web scraping approach. Wish that someday we can acquire data more easily by partnering with LinkedIn.

2. Data ETL: from raw data to structured, clean data.

Though the size of our raw data was not too large (around 14GB), it took us a long time to extract and format data from HTML files to clean data. Because with different layouts of HTML, different parsing files are needed (we wrote more than 10 python files to parse different pages). Some of the HTML files are formatted in multiple layers, which make it really time-consuming to figure out the structure of the whole page and proper extracting formula.

Also, after extract data from HTML files, a lot of work were done to transform the data into a proper format before loading into our database. For instance, when we tried to match salary data for specific job title and its description (data scraped from different sources), we cannot join by their job title because job titles in these two tables are not exactly the same. Thus we add another field in these two tables called 'type' by manually selecting particular fields of job type that might be interesting for our big data students. Then we can easily connect the job table and the salary table.

3. Spring Boot Framework:

- a. Configuration and dependency setup: We searched for multiple sources to figure out what dependencies we need for the projects and how to make different components work well with each other.
- b. ORM(Object-Relational Mapping): We revised our database schema several times to better fit our data and system user cases, and changed the POJOs accordingly to successfully map the objects to the data in our database.
- c. Connect to the SFU virtual machine remotely: It took us a while to setup the configuration. For example, we had to forward our local port to enable our local development environment to connect with the RabbitMQ and the VM MySQL on the gateway.

4. Spark:

We tried many ways to set up a long-running spark context. Our first try was to directly create a spark context instance in our web server. By using `org.apache.spark.SparkConf` and `org.apache.spark.api.java.JavaSparkContext`, we managed to initialize a context based on our local stand-alone Spark, which ran smoothly in our

development environment. However, when we changed the spark master to "Yarn" which runs on SFU gateway, we faced a problem that beyond our ability to resolve and we couldn't find any useful information online. Therefore we decided to use RabbitMQ as a middleware, and finally got the web server interacting with Spark via the Message Queue.

5. RabbitMQ:

- a. How to build the connection between the producer(Spring Boot server) and the consumer(python server on the gateway): We created two exchanges on both sides with the same name in order to make them connect to the same queue on the RabbitMQ deployed on the gateway.
- b. How to handle the messages sending and results receiving synchronously and asynchronously: Since this is a RPC(Remote Procedure Call) problem, it really took us a lot of time to nail it. We can just use the function `convertSendAndReceive()` to send our messages to the queue, and our python server would handle the requests and send back the results to the reply queue synchronously. However, since each request would take Spark to compute for a while, we need an asynchronous approach to deal with this situation. We built a listener wrapped in a container to listen to the messages in the reply queue and receive the results from our python server.

6. Load data to MySQL database: Our data were JSON formatted parsed by Beautiful Soup. However, since the data sources are different, we cannot assure the correctness and distinctiveness of our data. Therefore, when loading data to the database, we need to apply the fault-tolerant mechanism to ensure that the data format is consistent with our table schema.

Results:

- a. Outcomes: A web-based job search platform(<http://nml-cloud-231.cs.sfu.ca:8080/index>) which allows users to search for jobs in terms of company, location, title, interview and understand statistical insights(e.g. salary by job types, alumni by companies, job posting by locations, etc.). Please see the Appendix for some screenshots of our web.
- b. From the data analysis, we learned various facts from the job salary, position numbers, and alumni distribution. For instance, we learned that while the average salaries of Data Science jobs are similar to Software Development, the number of Software Development Jobs is about 2 times of Data Science in Vancouver. We also learned that Toronto has much more Data Science positions than Vancouver, and RBC has the most SFU alumni among some famous Canadian companies.

- c. From the implementation. We learned lots of knowledge and new technologies from data collection to data visualization, from frontend UI to backend server, and from database design to distributed computing. We also learned that a rich and clean dataset is extremely hard to obtain.

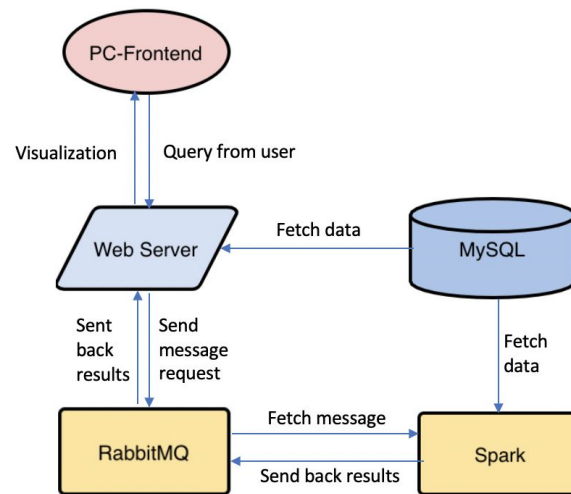


Figure1. Data Flow

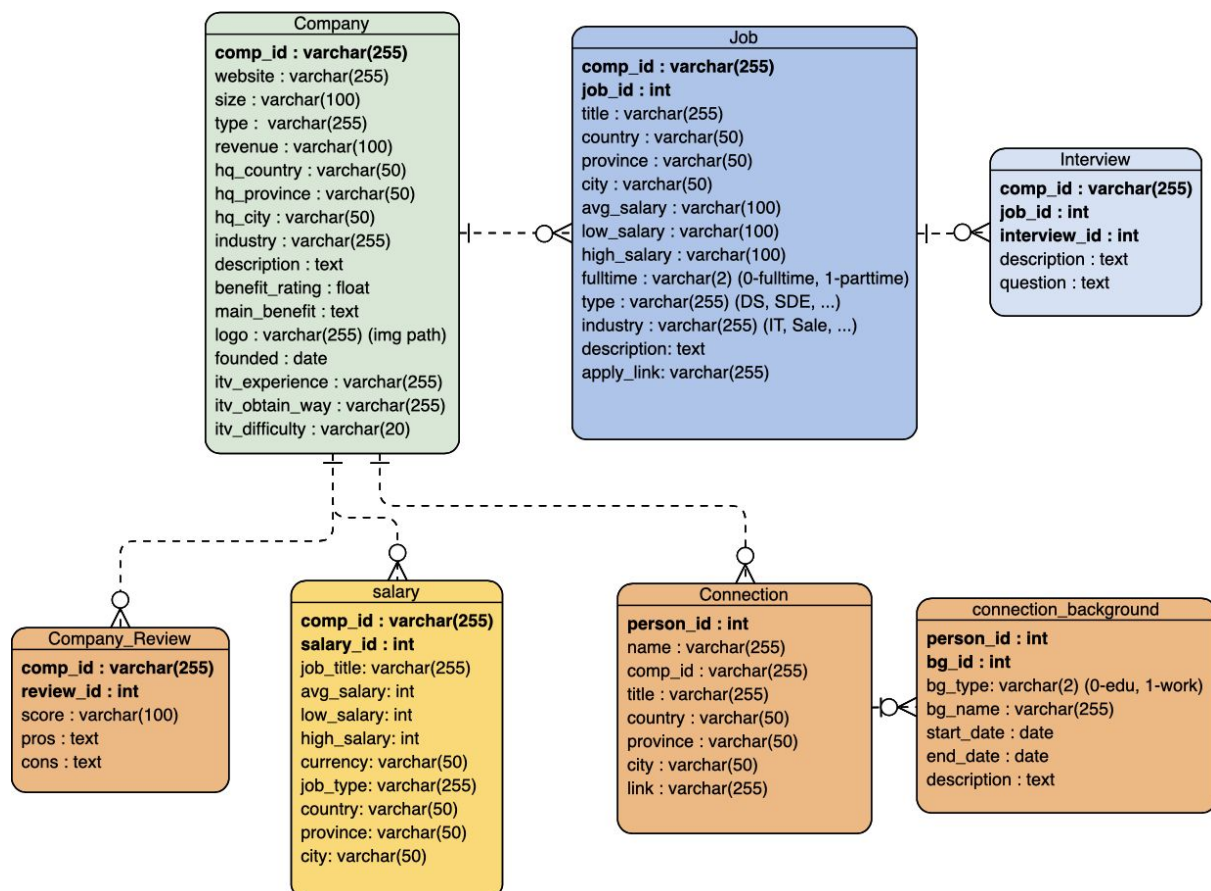


Figure2. Entity Relationship Diagram


Project Summary:

1. Getting the data(3.5): Data was acquired by web scraping from Indeed, Glassdoor, and LinkedIn.
 2. ETL(2.5): HTML to JSON to MySQL.
 3. Bigness/parallelization(3): Through RabbitMQ, our web server is able to communicate with Spark to utilize its computing power for parallelly handling data analysis on big data sets.
 4. UI(1.5): Unified, animated and responsive frontend UI to improve user experience and meet industrial standards.
 5. Visualization(3.5): Interactive web frontend enabling users to explore job, salary, company and alumni information as well as data analysis based on their needs.
 6. Technologies(5):
 - Frontend: Thymeleaf, BootStrap, JQuery, AJAX, ChartJs, DataTables, FontAwesome, P-Loading, BootStrap-Select
 - Backend: Spring Boot Framework, JPA, RPC, Tomcat, Log4J, Maven, JSch
 - MiddleWare: RabbitMQ
 - Database: MySQL, PhpMyAdmin
 - Data ETL: Beautiful Soup, Regular Expression
 - Web Scraping: Scrapy
 7. Problem(0.5): Our platform has been able to provide basic information for job seekers and get connected with people in particular backgrounds, with proper visualization approaches. However, due to the project schedule and data source restriction, our dataset currently are limited. This platform would be much more helpful after getting more data.
 8. Algorithmic work(0.5): Wrote smart search algorithms to retrieve desired data from MySQL database following JPA criteria.
- Future work: With more data, we could apply data mining and machine learning techniques to build a job recommendation system based on users' education background, work experience, and their preferences.

References:

- <https://spring.io/guides/gs/spring-boot/>
- <https://spring.io/guides/gs/messaging-rabbitmq/>
- <https://www.rabbitmq.com/tutorials/tutorial-six-java.html>
- <https://ca.indeed.com/>
- <https://www.glassdoor.ca/index.htm>
- <https://www.linkedin.com/>
- <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- <https://scrapy.org>
- <https://getbootstrap.com>

APPENDIX (Some of the Web Pages)


DIOR
DATA SCIENTIST

[HOME](#)
[JOB](#)
[COMPANY](#)
[ANALYSIS](#)








Search Result For :

Find the job interests you most, then get it — See what help you can get from us.

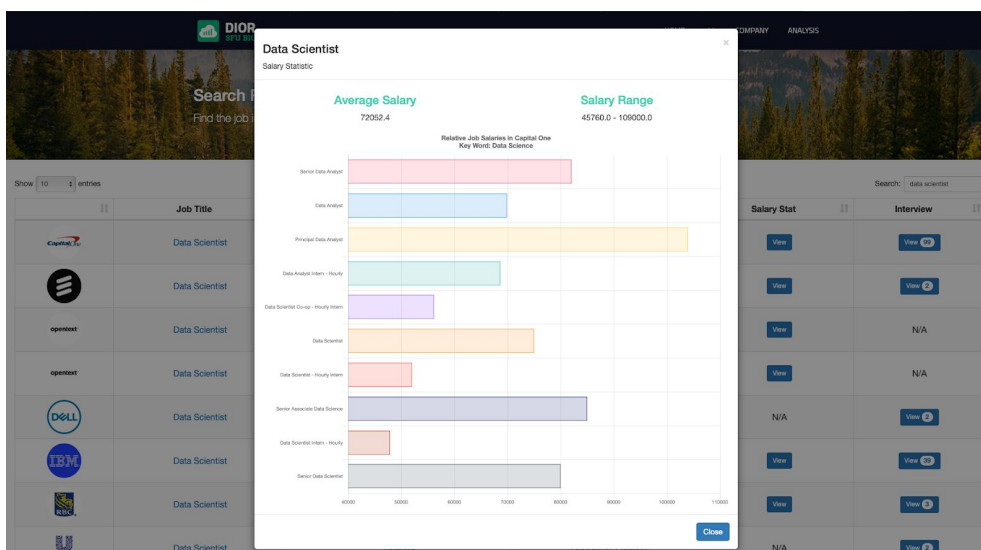
Show
10
1
entries

Search:

data scientist

	Job Title	Company	Location	Salary Stat	Interview
	Data Scientist	Capital One	Toronto	View	View
	Data Scientist	Ericsson-Worldwide	Noida	View	View
	Data Scientist	OpenText	Tinton Falls, NJ	View	N/A
	Data Scientist	OpenText	Sydney	View	N/A
	Data Scientist	Dell	Round Rock, TX	N/A	View
	Data Scientist	IBM	Armonk, NY	View	View
	Data Scientist	RBC	Toronto	View	View

Job List



Job Salary Statistics

DIOR

SFU 000

Search F

Find the job

Show 10 entries

Job Title
Data Scientist
Data Scientist
Data Scientist
Data Scientist
Data Scientist
Data Scientist
Data Scientist
Data Scientist

COMPANY

ANALYSIS

Data Scientist

Interview Questions

2016-10-23

Name a time you went above and beyond to help someone.

2016-09-18

How often are you late for meetings? Do some calculations based on GDP

2016-09-13

Business cases were nothing unexpected. Google a few practice cases and you should be good to go. I got stuck a few times and needed help, and the interviewer was more than willing to provide it.

2016-08-24

(onsite) How would you explain the multinomial distribution and write python code on a whiteboard to represent this distribution

2016-07-24

Salary Stat

Interview

View

View

View

View

View

N/A

View

N/A

N/A

View

View

View

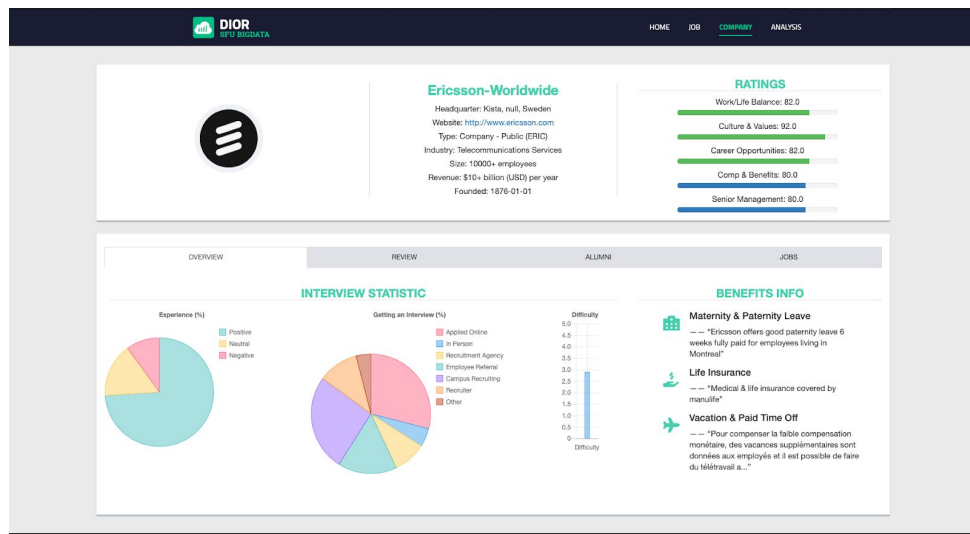
View

View

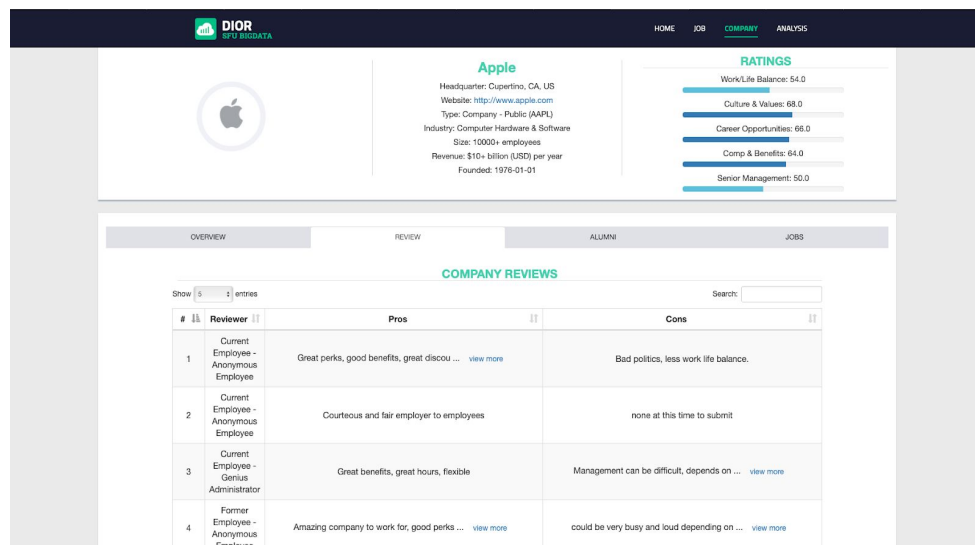
N/A

View

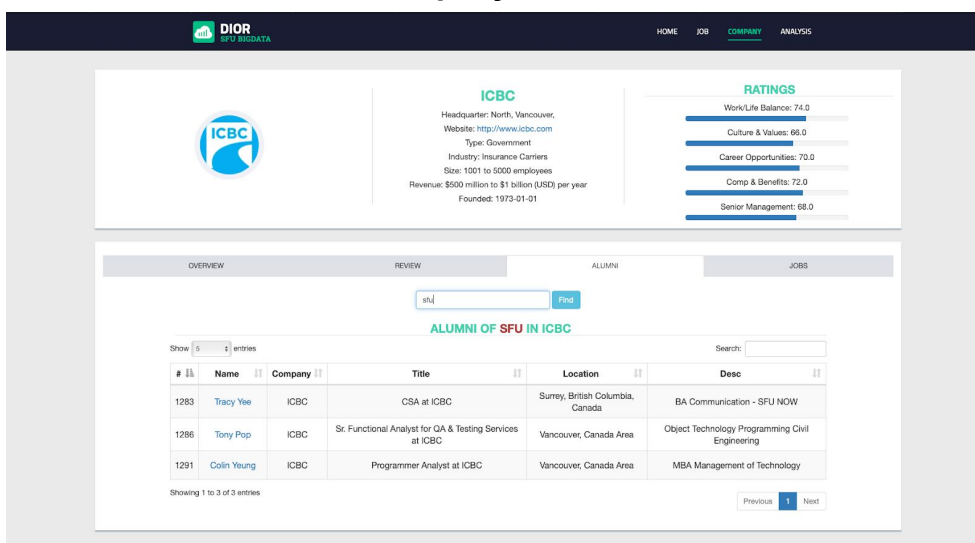
Job Interview Questions



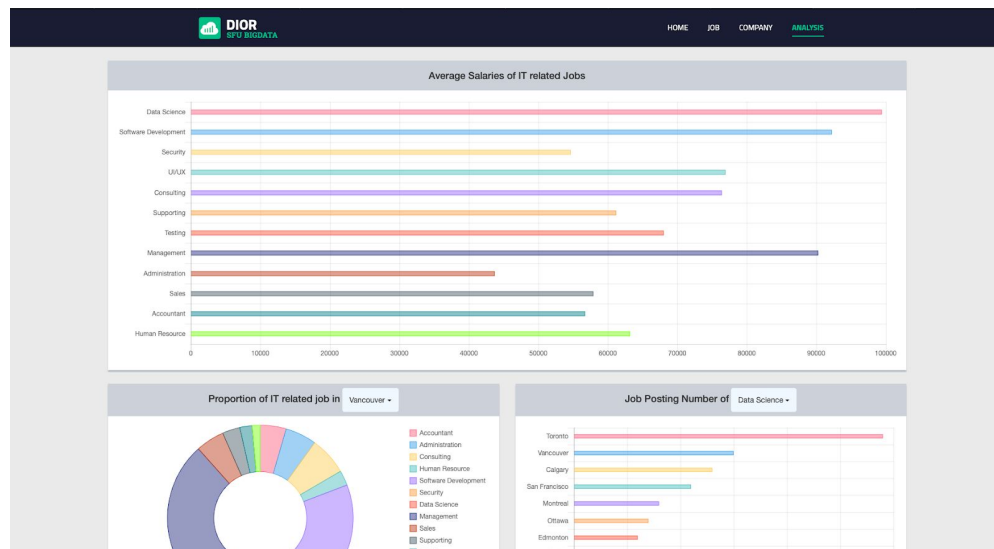
Company Overview



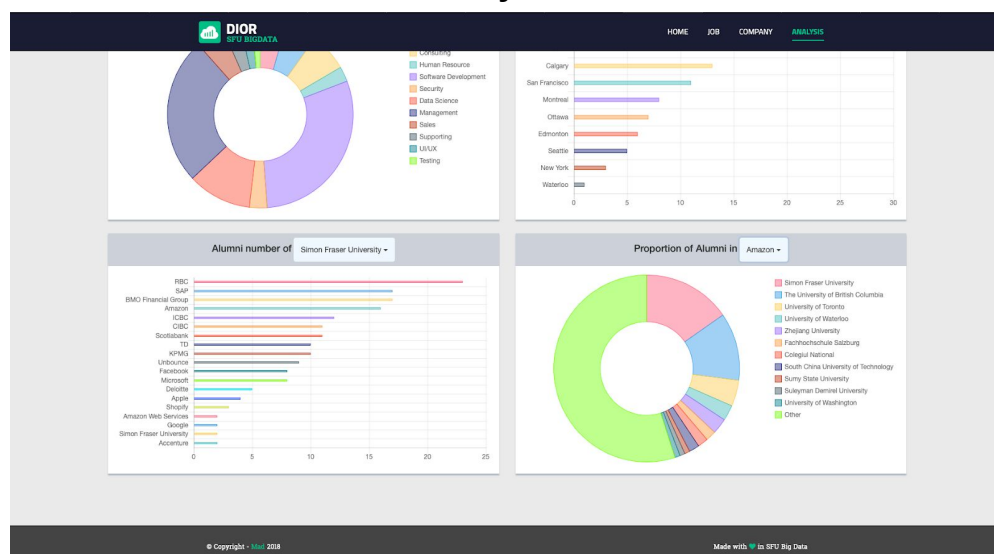
Company Review



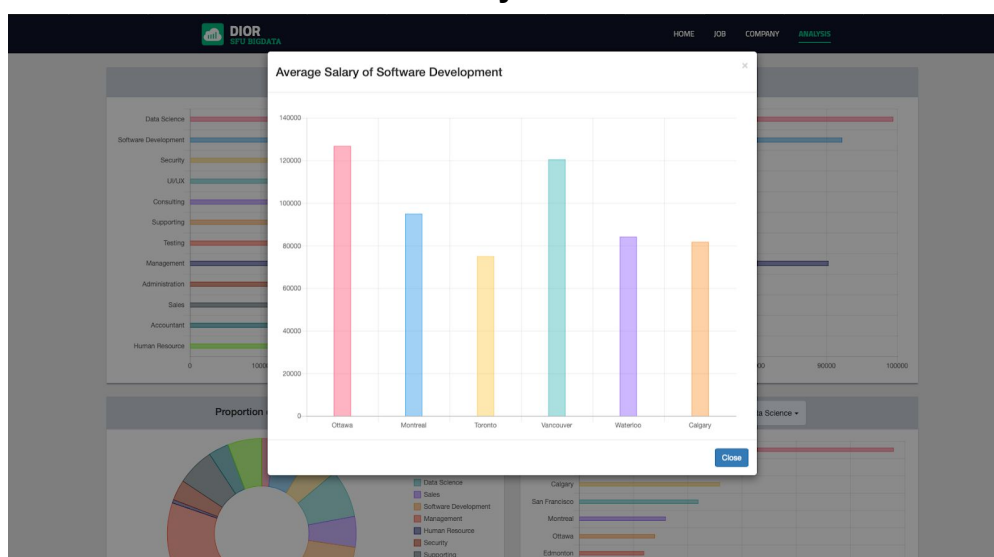
Company Alumni



Analysis 1



Analysis 2



Analysis 3 (By clicking a certain type of job type in the graph on top)