WEB SCRAPING FOR RECRUITMENT ANALYTICS

With Continental Tire LLC, North America

Team:

Ketki S.
Megha K.
Rohan D.
Rohit D.
Smruti R.

WEB SCRAPING FOR RECRUITMENT ANALYTICS

INTRODUCTION

1.1. Project Overview

Web Scraping for Recruitment Analytics is a project in which we have built a fully automated tool that provides our client Continental, with immediate market intelligence on employment conditions in targeted geographies.

The primary objective of this project was to build an automated tool for client to dynamically scrape web sources, store the data in a structured form and then visualise & analyze it. The analysis would be done on the data about the different jobs available in the market and details like salary, job title, location, etc. The HR executives should be able to analyze that data and make the necessary business decisions. The data would be scrapped from various online job portals and according to the requirements, the data will be used for various analysis. Our Team worked with Dr. Mohsen Dorodchi, Professor, College of computing and informatics UNCC & Mr. Wayne Pennell from Continental, for support, guidance and feedback on the alignment of the project to their requirements.

1.2. Theoretical background

The primary requirements of Continental were, the User should be able to track what kind compensation are Continental's chief competitors offering for a specific role in a particular location. This information could possibly reduce a considerate amount of money in Staffing Operations and aid in planning expansion of the organization as well.

1.3. Learning Experience

While working on this project, we experienced what it is like to be working in a professional environment, with a liberty to make mistakes and learn from those mistakes. It provided us with insights on the various software development practices followed by the Industry.

Below mentioned are a few key lessons:

- Working individually is completely different than working in a team. Each individual has diverse set of skills and working manner. This project helped us understand how to work as a team and develop a complete product.
- We learnt how to work in an Agile Environment.
- We gained experience on delivering a software product, end-to-end.
- We also honed our Presentation and Public Speaking capabilities.

PROJECT DESCRIPTION:

The objective of this project was to create a data model for the job data of Continental and its competitors. Data needs to be extracted from various data sources available on web and mined in an efficient manner to be able to derive insights from the data. Our client, Continental could use this information to decide upon the factors while they expand in the states of Michigan, Illinois and Texas. Looking at their competitor's hiring trends in these three states, Continental shall come up with strategic techniques in terms of hiring their future employees.

Scope:

- To develop a dashboard that will load the data dynamically which is extracted from various data sources and create visual representation of the data.
- Develop a fully automated backend engine that can scrape, clean and combine data into a SQL database.
- Data must include job title, Location, Salary, Data source URL, Timeline of job posted along with employment incentive details
- CReate a filterable interactive dynamic dashboard to visualize this data.

Assumption:

- End user of the Dashboard will be the hiring team that includes HR, the Strategy team and the Finance team.
- Personnels are familiar with Tableau functionalities.
- Data extracted from more than one source
- Data collected from Monster.com will not have Salary details and we will have to look up for similar positions for obtaining missing information for those job postings.
- Data collected from Glassdoor will have Job title, Location, Salary, Timeline, Skills information and can be utilized for consistent mining.
- Data collected from Indeed.com would not have salary for all postings and similar to Monster data, these records shall be compared with Glassdoor to be able to maintain the consistency in data.
- Unique data values are maintained while storing the cleaned data in the Database.
- Data is stored in the third normalized form.
- Analysis is performed on the cleaned data after being scraped from three different online sources.
- Dataset is restricted to data for the states of Michigan, Texas and Illinois.

User Scenarios/ Business Scenario:

Following are the critical user scenarios for the complete system:

- As a data analyst, I would be able to run the GUI and get the scraped data as cleaned CSV files to perform further mining in the model.
- As a data analyst, I would be able to view the dashboard to view different aspects and insights of the scraped data.
- As a database manager, I can view the structured database tables and schema in SQL to run queries and better understand the data.
- As a database manager, I can apply filters in database to view the data that I want.
- As a business analyst, I can view the stored and visualized data to analyze critical issues and make business decisions.
- As an admin, I can modify the scrapers from the backend of the GUI to scrape different parameters and sources.
- As an admin, I can modify the database view to better understand the data.
- As an admin, I can change the layout for GUI to make it more appealing and smooth.

Business scenario:

- Describes a significant business need: A significant business need is that as a business analyst, I should be able to view the stored and visualized data to analyze critical issues and make business decisions.

- Identifies, documents, and ranks the problem that is driving the scenario: One of the major problems that is driving the scenario is realizing that the hiring process is an integral part of running any business and it can be tough on its own. According to CareerBuilder, 40 percent of employees plan to change jobs in 2018. That means this is the year to step up the recruitment process to stand out among competition. With the huge amount of data flowing through the process, every hour, it's important to drive the data through one single channel to analyze it and narrow down the process to optimize it. This is done not just for Continental but also its competitors which would help track the market well.

- Describes the business and technical environment that will resolve the problem: Technically, there is scraper that's running on the backend which is constructed using Python. It scrapes the data from 3 job sources for Continental and its 3 companies. It merges them and clean the data by removing missing values, keeping unique ones, and standardizing the parameters throughout. The cleaned and merged data would be passed to a GUI and a database to store and analyze. The database would be connected to the dashboard to visualize all the data at one single place.

- States the desired objectives: The objective is to make an efficient model to drive the companies job data so as the business team can draw insights from them and eventually contribute it to decision making in terms of hiring.

- List the “Actors” and where they fit in the business model: The actors are data analyst, business analyst, database manager, admin. Their roles in the business model can be seen from user scenarios.

- Metrics for success: Few insights that we came up with in respect to this use case were- hiring trend, cost per posting, Comparative incentives insights, hiring with time, female hiring percentage with time. For example: Hiring trend is one metric that matters tremendously to business stakeholders. If recruiters are submitting low-quality talent, hiring managers are wasting valuable time and resources filtering through them.

Tools used:

Python 3:

- BeautifulSoup and Request Libraries for scraping.
- Node JS with Python 3 Qt GUI for designing front-end desktop application.

Tableau Public:

- For Data Visualization & Analytics.

MySQL

- For storing and retrieving data.

AGILE DEVELOPMENT

The Web Scraping for Recruitment Analytics team chose the Agile methodology for the development of the project. As part of the agile development life cycle, we came up with a set of tasks and divided them into 5 sprints.

We held Scrum meetings regularly to discuss individual progress. At the end of each sprint, we used to integrate our tasks to come up with a prototype of the product. Following the agile manifesto helped us work efficiently and is highly recommended for the successful delivery of the project.

Task Identifier	**Priority**	**Estimation (hours)**
Preliminary Research (Sprint 0)		60
Searching for websites to scrape data	high	8
Brainstorming on technologies	medium	16
Defining user stories	high	6
Preparation of Sprint 0 presentation	medium	5
Collection of data about job postings of Continental in TX,MI & IL	high	4
Collection of data about Job postings of Bosch in TX, MI & IL	high	4
Collection of data about Job postings of Delphi Technologies in TX, MI & IL	high	4

Collection of data about Job postings of Siemens in TX, MI & IL	high	4
Collection of Incentive information	high	5
Preliminary visualization of data	medium	4

Task Identifier	**Priority**	**Estimation (hours)**
Data Scraping-Building custom crawlers (Sprint 1)		130
Write code to scrap jobs from indeed.com	high	35
Write code to scrap jobs from glassdoor.com	high	35
Write code to scrap jobs from monster.com	high	35
Testing end-to-end services of the task.	medium	25

Task Identifier	**Priority**	**Estimation (hours)**
Data Processing (Sprint 2)		165

Write code to parse HTML Data	high	35
Write code to parse XML Data	high	35
Write code to parse JSON Data	high	35
Write script to integrate the custom crawlers and generate merged CSV	high	35
Testing end-to-end services of the task.	medium	25

Task Identifier	**Priority**	**Estimation (hours)**
Data Storage (Sprint 3)		190
Brainstorm to make the Data base scalable	medium	35
Designing the Data Base Schema	high	35
Write code to import the data in to the database	high	35

Write custom classes for Error Handling	high	35
Writing code to fetch data from the Data Base	high	25
Testing end-to-end services of the task.	medium	25

IMPLEMENTATION COMPLETED

We developed an end-to-end solution for web scraping and dashboard using desktop GUI and Tableau.

1. **Desktop Application:**

    - This application is developed for execution of web scraping. Our application can scrap the data independently from each data source that we have chosen as well as for all three data sources at once.
    - Backend functionality of this app is an object oriented Python module which consists of scraping scripts for indeed, monster and glassdoor.
    - The second part of the module deals with cleaning the data and storing it into csv files and the last part deals with connecting to the MySQL server and inserting these columns into SQL database in 3NF format.

2. **Tableau Dashboard:**

    - By connecting Tableau with MySQL database, we can load the data seamlessly.
    - Our dashboard is available online on the Tableau server and can be accessed by an end user easily by accessing the URL.
    - The dashboard is segregated into four tabs. First part shows, distribution of jobs in three states wrt their data sources. Here we can see how many jobs of Bosch are posted on Glassdoor, Monster and Indeed separately. This way we can check it for Continental, Delphi Technologies and Siemens as well.
    - The second part deals with the Location component as we want to know jobs per three state and this is a map based representation of jobs in the USA. This is an interactive element as one can choose the filter as per company name or state and we distinct information.
    - We identified Key Metrics like the Count of Job Ads in specific locations and organizations etc.This part mainly deals with statistical distribution of data. In this we could examine total job ads posted per company per state, maximum salaries offered by each firm. This information is stored for each quarter of the year 2017 to present and can be viewed in this fashion by selecting appropriate filters.
    - The last part consists of forecast for job advertisement trends, salaries and total jobs created by competitors.

PROJECT PIPELINE

Data source selection:

For implementing this project, we needed as much data as possible. Thus, we decided to explore as many data sources as possible. To begin with, we scrapped multiple job portals however, our biggest challenge was consistent data and less redundancy in extracted details. We defined certain attributes for this business problem such as job title, description, salary, skills, location, timeline information of job posting. While finalizing the data sources, we went with the one that has most suitable data with respect to our defined variables and information which is unique for that data source. Sources like LinkedIn which are most popular on board these days needed paid API. When we tried our first run of scraping, we were blocked by the site and thus there was no scope to pick up any information from this resource.

We finalized three data sources and those are monster.com, glassdoor.com and indeed.com

Process of scraping

We used BeautifulSoup library of NLTK for scraping the data. Our choice was unanimous for the simple nature and ease in implementation when it comes to choosing BS4. We used Python 3 over 2 for the familiarity of this language within team members.

Data preprocessing:

In the extracted data from different resources, we observed that the fields have different headers. One source had “Title” and another had no header for “Job_title”. We took care of this inconsistency first followed by maintaining missing values.

For salaries, we primarily used glassdoor jobs for salary insights. For missing values, we handled it using an average salary per state. Also, for values, we used pandas for maintaining numerical values by removing special characters used for currency, etc.

For location, we maintained consistency by maintaining separate columns for “City” and “State”. This helped us drawing some insights in visual implementation. We could find distribution of details per city per state.

For organization, we experienced that a few records were of another organizations apart from chosen 4 companies. We had to remove such records to ensure that there are no noisy entries in data.

After cleaning our extracted data, we maintained a clean, merged csv file with consistent column headers. We stored data extracted for all 4 companies and from each source into this file.

Desktop application

We have developed a desktop based application for web scraping for the three data sources and for chosen companies..The front end is build to assure that any employee i.e. HR analyst, Data Analyst or recruitment analyst can run or use the web scraper .
We built an interactive GUI(Graphical User Interface) which was supported by python 3 in the backend. Database connectivity with python was utilized to feed data from GUI to SQL smoothly. This application is built with using an element of Node.js that is Electron and NPM which helps us in connecting GUI with python scraping program.

Main Screens:

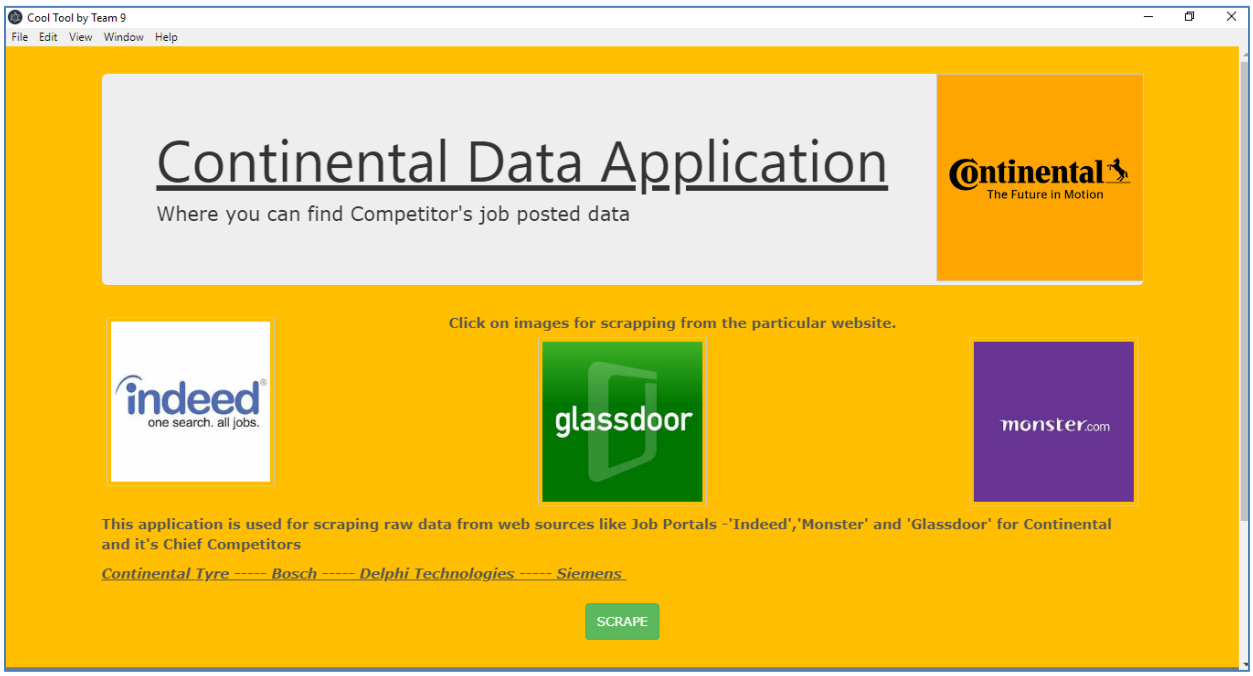

Fig: GUI for scraping

This screen is developed by using programming languages like HTML/CSS, some javascript and we also used Node.js as it helps in connecting HTML and python in both passing of input and output. Here we can see that we have one button for scrapping which calls the python web scraper program and scraps all the web elements.
This program scraps all the data source.

Second Screen (Data Source):

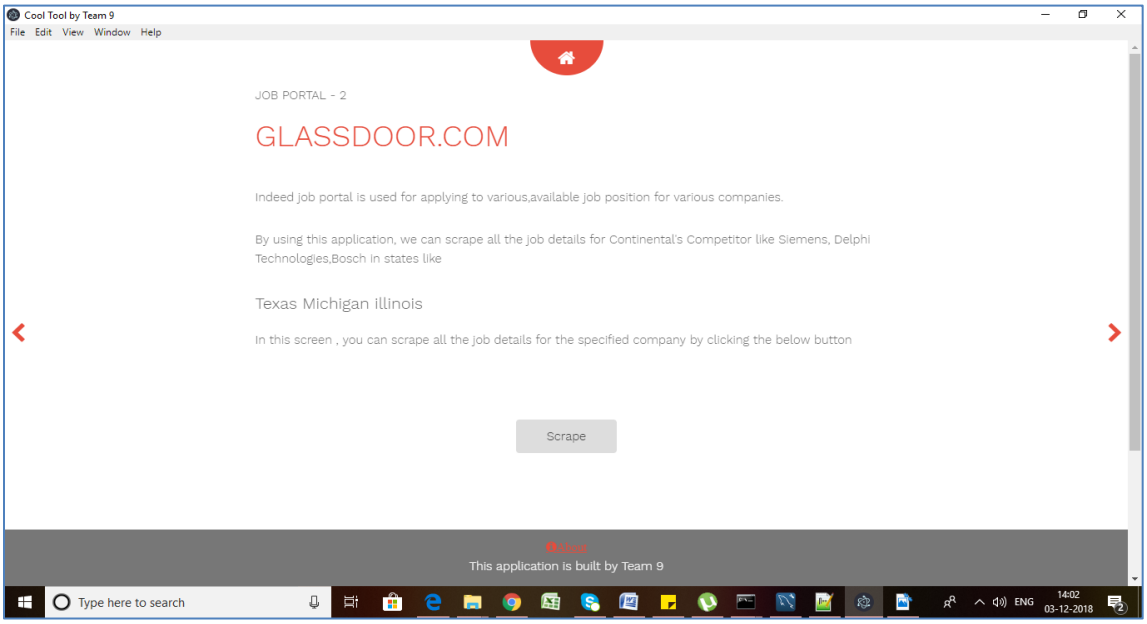

Fig: Independent data source scraping page

This screen is for all the individual data sources and we can scrape them individually with all the job details. We can also see the details of continental if we click on the continental image if the user is new.

Database design

Database is designed to be in 3N form. The database is stored in MySQL. The merged and cleaned data file that is created from Python Scraper is directly connected to this database. Hence, everytime the program scrapes the data, it is updated in the database in MySQL. The database has 3 tables named: jobmining, salary, location. Jobmining stores all the data scraped for all the companies from 3 sources in its clean formed and consisting of all parameters. Since, it could be a huge file to draw every insight from, the database is made in 3N form and a salary table and a location table is created to exclusively look for only salaries and locations respectively with respect to the company and position.

Python supports relational database systems and SQL cursors. Hence, database connectivity through the scraper can be smooth. We have used mysql.connector module, created the connection using connect() method and passed the hostname, user and password for the database that I wanted to access. A cursor object is called to be capable to send commands to the SQL. Cursor is a control structure used to traverse and fetch the records of the database. Cursor has a major role in working with Python. All the commands will be executed using cursor object only. The python merged file is read and all the sql queries are passed in the file and the cursor method is called and executed to establish connectivity to the database.

TABLEAU RESULTS

The following dashboard was designed using the data that we scrapped from indeed, monster and glassdoor.com. This data was cleaned and stored in database to which we connect Tableau to extract the records for visualization.

Below are the preliminary dashboard insights.

1. **Key Metrics:**

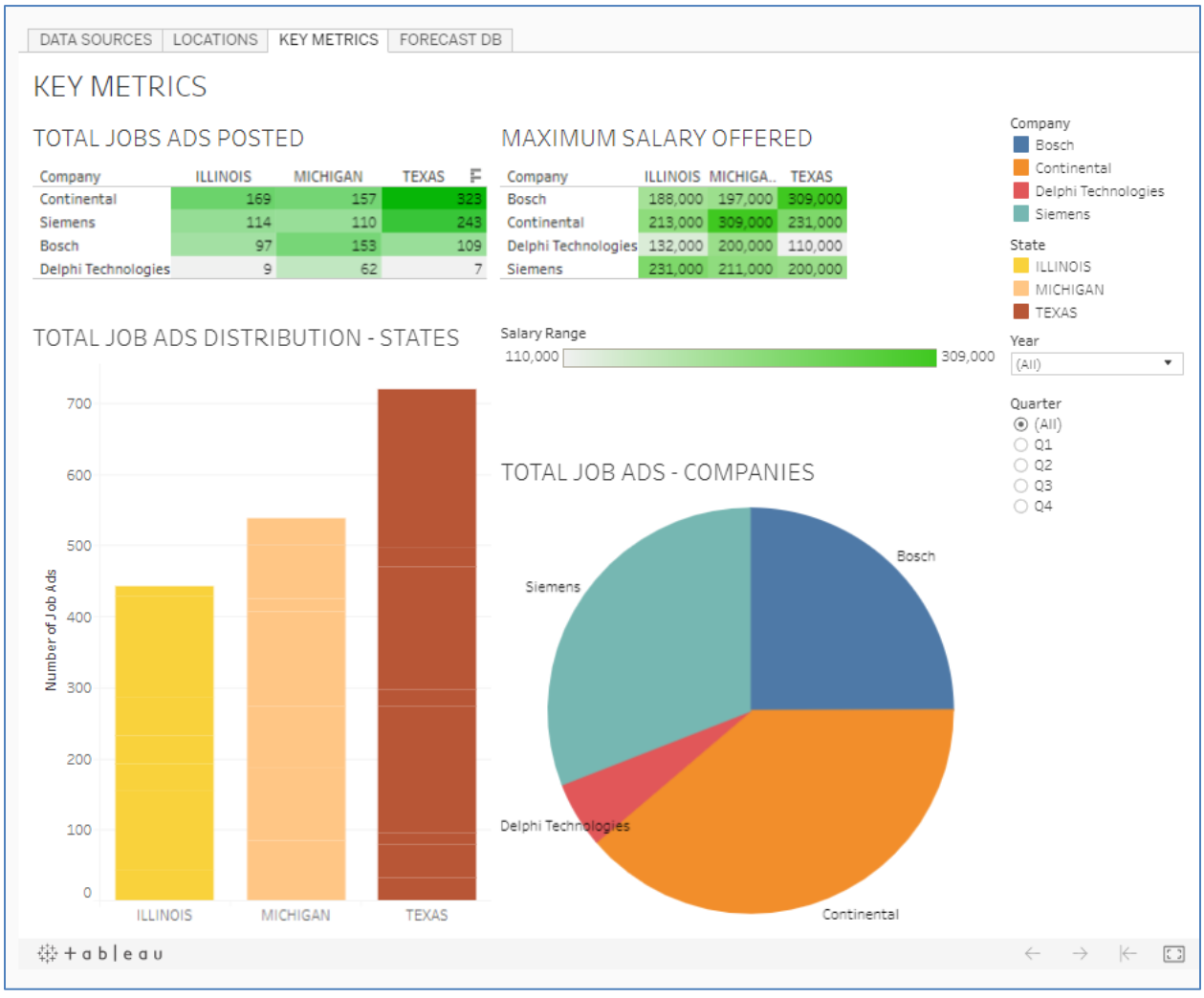

Fig: Key metrics of data to demonstrate distribution of job ads and salaries per company per state

Inference:

Using this dashboard we can examine total number of job ads posted per company in the state of TX, MI, IL. This chart also shows the salary distribution for these 3 states. From this metric we can see that Bosch offers maximum salary of 188k for the state of IL where as for TX their highest salary is 303k USD. This shows how different salary structures are wrt the location. Similarly, we can examine those aspects for other companies as well.
The bar and pie chart is a pictorial representation of the same details for quick grasp.

2. Forecast DB:

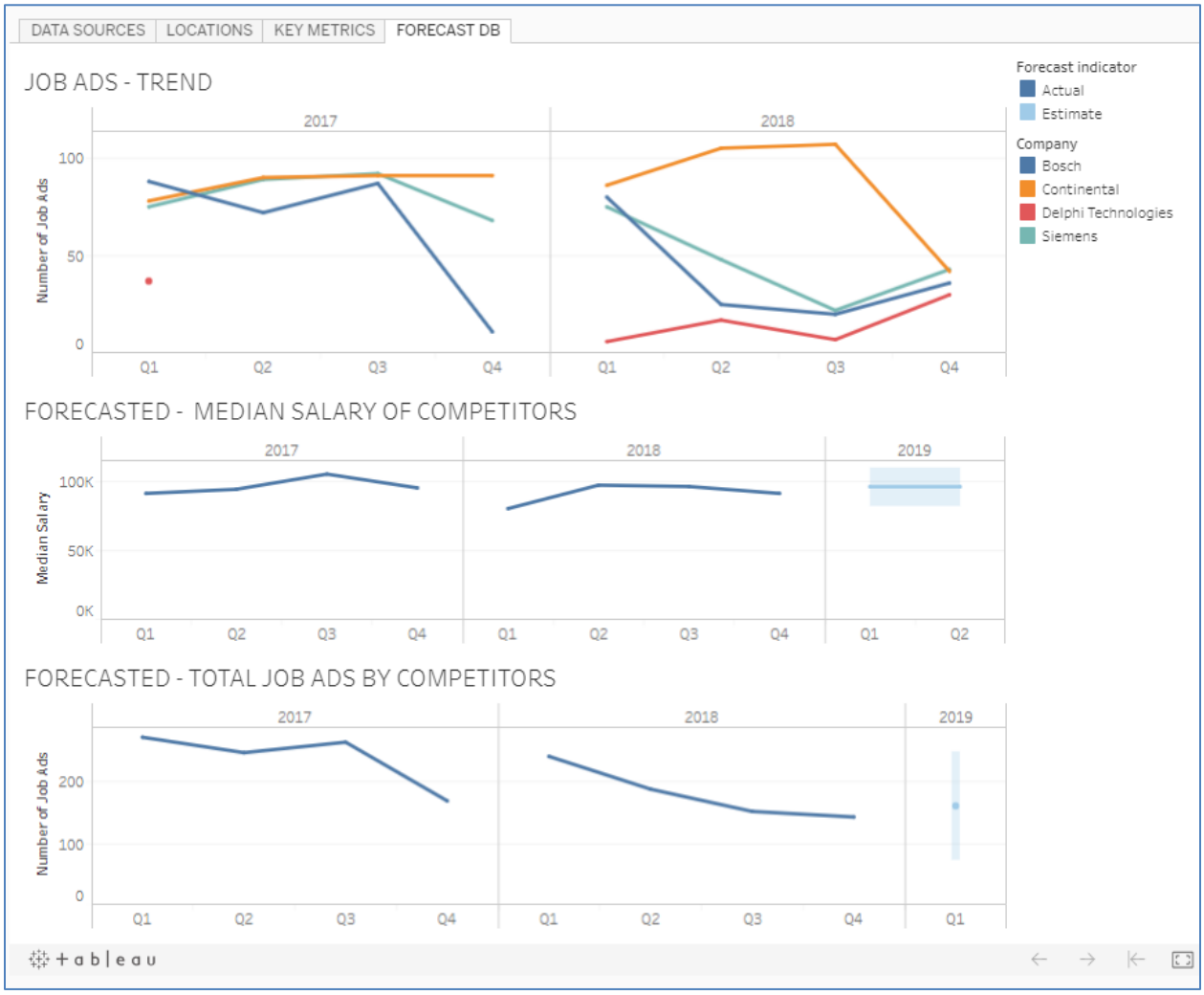

Fig: Job ads and salary forecast based on data

Inference:

In this task we have derived certain key element forecasts based on the information we have extracted. We have number of day data when the job was posted. Taking that into consideration, we calculated date and time for each record. Using “Time” details as primary, we could draw forecast for job trends, salary and number of jobs that will be created by competitors.

3. Data Sources:

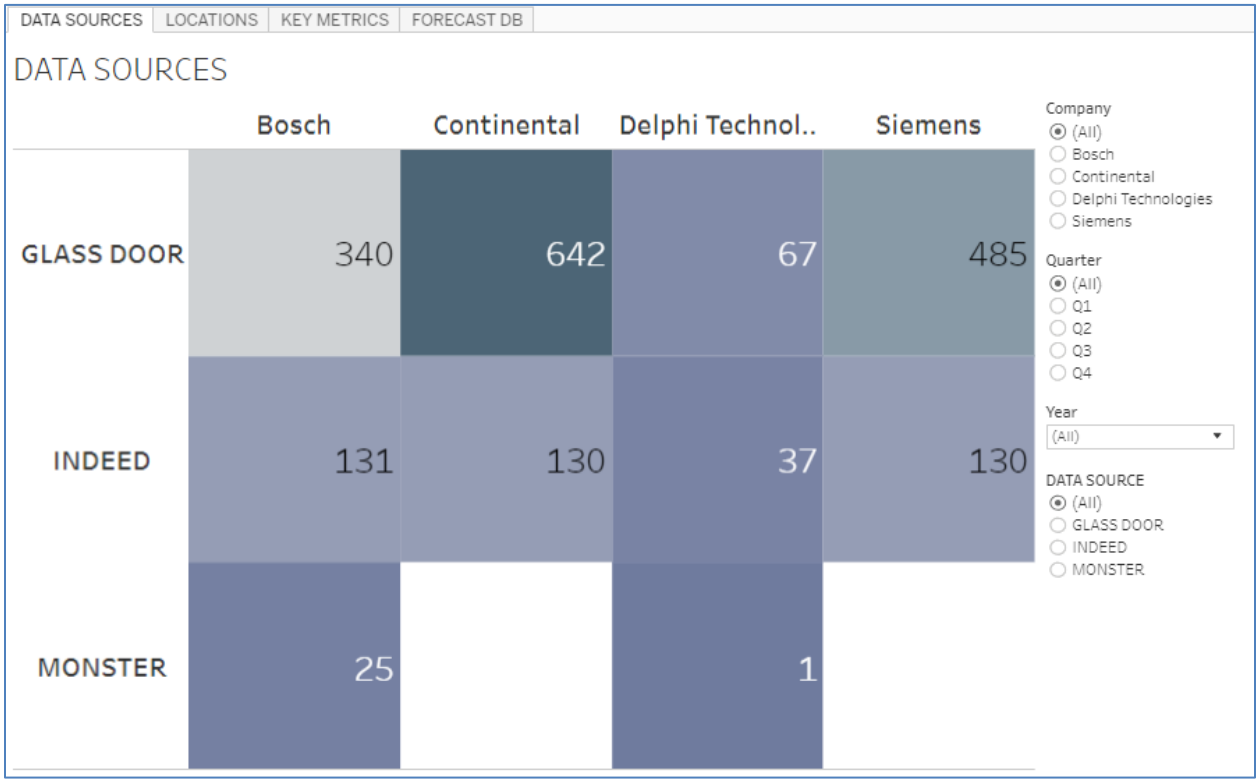

Fig: Number of jobs per data source per company for each quarter.

Inference:

This chart is useful to derive insights such as extracting total number of jobs per company on every data source. For example, here we can see that for Continental, we have 642 jobs on Glassdoor and 130 jobs on Indeed.com. Similarly, we have added more filters to select the data as per company or data source in each quarter.

4. Location:

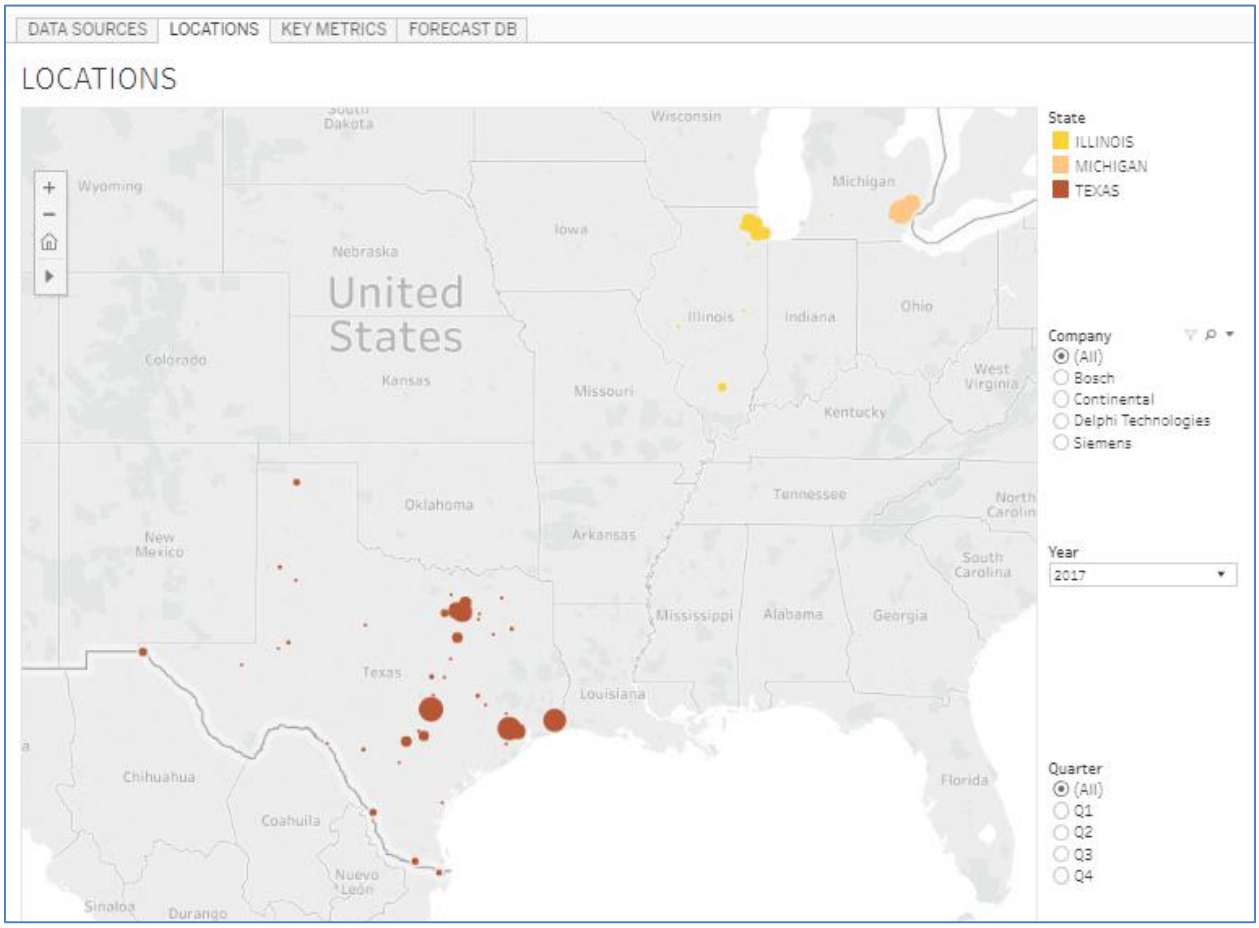

Fig: Map representation of number of jobs scattered across TX, MI, IL for each company

Inference:

This map is simply utilized to show density of jobs per state for each company. Here a user can select a year to examine the trend.

STATISTICS

Based on our data and some additional scraping, we could draw some statistical inference.

Below are the details:

1. **Employment Growth:**

Fig: Hiring trend from 2016 to 2017 for Continental

We extracted the data for total number of employees in the year of 2016 and 2017. Here we can infer that the company has grown by almost 1000 employee in a year.

2. Female Employment:

Fig: Percentage of female employment to total number of employees

As the company has grown by 1000 employees, number of female employees also have increased from 2016-17 and there is one percent increase in number of female employees in top management.

3. Hiring trend

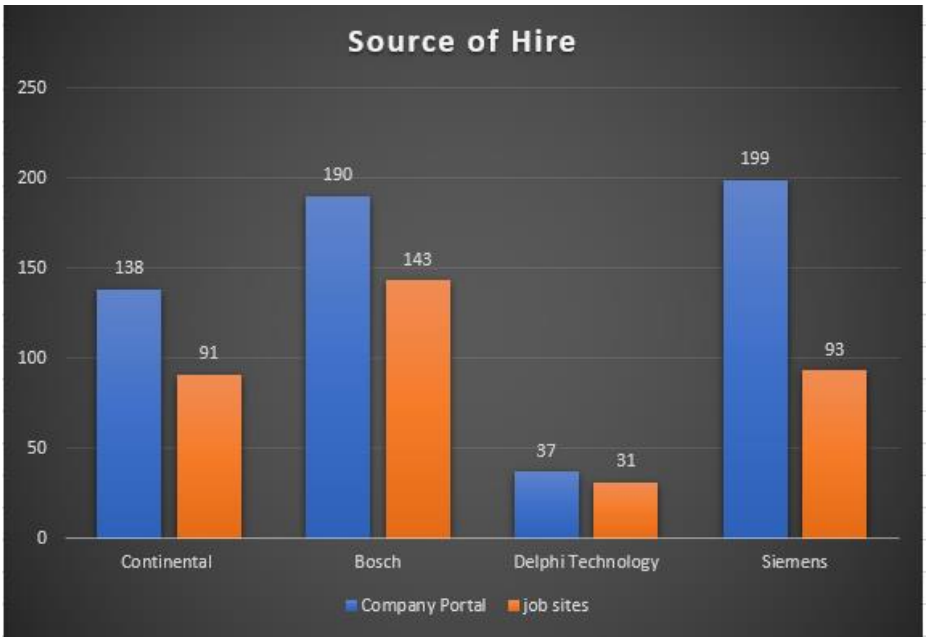

Fig: Number of jobs posted on portal vs number of jobs available on company's own website.

This chart shows the hiring trend of the company. We can examine that out of 138 jobs, 91 jobs are posted on job portals. This means that for the rest of the job, company may prefer internal vacancy, referral, etc. Similarly, for Bosch, 143 recruits would be from portal and the rest 47 would be from another resources.

4. Expenditure on staff

Company	Parameters	October	November
Continental	Total number of posting	91	5
	Avg Salary per posting	89000	
	Total spent	8.09M	445k
Bosch	Total number of posting	87	55
	Avg Salary per posting	102000	
	Total spent	8.87M	5.61M
Delphi Technologies	Total number of posting	25	7
	Avg Salary per posting	105000	
	Total spent	2.62M	735k
Siemens	Total number of posting	58	35
	Avg Salary per posting	108000	
	Total spent	6.26M	3.78M

Table: Expenditure through salary for each organization

This table represents how the company is spending. For example, Continental has 91 posting with average salary being 89k. This way we see that Continental has spent 8.09 million USD on recruitment in the month of October and 445k in November.
Similarly, we can examine the budget of another companies that they are using for recruitment.

5. Incentives

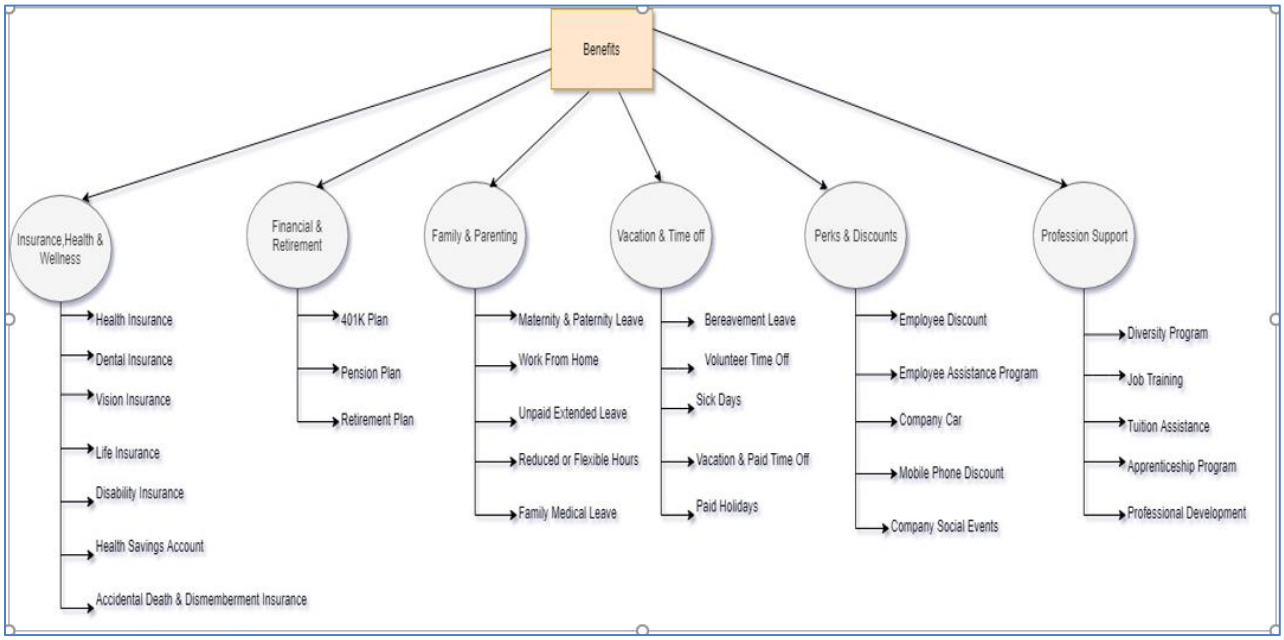

Fig: Employment incentives offered in market

There are multiple incentives that all organizations give. We have filtered the incentives provided by each company. There were total 55 incentives and below is the percentage distribution of these incentives as per company.

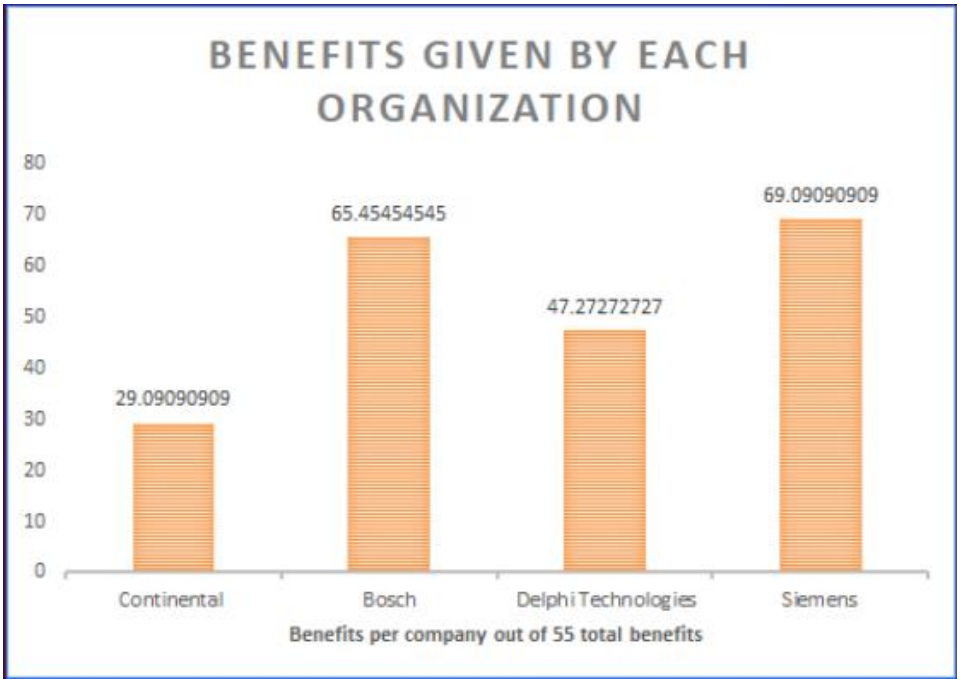

Fig: Percentage distribution of total benefits per organization.

FUTURE ENHANCEMENTS

Web Scraping:

In our project, we have used indeed, monster and glassdoor website. Here we do not have any Ajax or JavaScript pages to scrape from. If we want to fetch the data from such sites where JavaScript pages are to be used for data extraction, then libraries like Scrappy and Selenium could be used.

Also, for location, we have used search criteria being a specific state and that can be generalized to USA as country. This type of scrapping should fetch these four company's data for all states.

SQL:

Instead of using local MySQL database, we can opt for AWS platform and maintain the database on cloud. The advantage of this approach is storage! In real time scenario, we can store as much information as we want over a period of time using cloud.

Tableau:

We can add more visuals for the information extracted other than already implemented ones. Here since the primary agenda was to build a backend, we have not implemented all the possible visuals in Tableau.

Machine Learning Predictive Analytics:

Our data is also stored into a csv format which makes it ready to use for training various machine learning models. These models are handy for predictive analytics and time series analysis.