**Data Science Job Postings using MongoDB**

Antonio Garza, Shravan Kuchkula, Jack Nelson

Master of Data Science, Southern Methodist University

Dallas, TX

Tel.(A.G.) 832-452-8696, Tel.(S.K.) 404-402-1993, Tel.(J.N.) 248-231-4260

e-mail : antoniog@mail.smu.edu, skuchkula@mail.smu.edu, nelsonjohn@mail.smu.edu

**Abstract:** This is a study of data science jobs across the United States and it uses data gathered from the Dice.com Application Programming Interface (API). Our group used the NoSQL database, MongoDB, to analyze our JSON-formatted data. Additionally, we packaged the queries using Python. Specifically, we used PyMongo which contains tools for using MongoDB from Python. To analyze our results we used pandas, a Python package that provides fast, flexible, and expressive data structures that can be passed to visualization packages. Matplotlib, Seaborn (a python visualization library based on matplotlib) and plotly were then all utilized to create bar charts and heatmaps.

We ranked the data science demand by state, and also found the top company in each state with the most open data science positions. To assist our fellow classmates and others interested in finding available data science positions in their area, we created a script that can be run to query these job openings by zipcode. Lastly, we looked at how many unique companies are posting for data science jobs and which software skills are the most in demand across all job openings.

**1. Related Work**

As you can imagine, there are articles that have studied data science trends. Many however, were over a year old. We used these sources to both ensure that Dice.com data was on par with other job sites, as well as to find out what existed already and how we could enhance the analysis landscape. An article from gigaom.com from 2014 shows data science jobs by state.[1] We found that our analysis of job openings was similar in that California, Texas, and New York were part of the top four states for job openings. However, the article does not delve into the top companies hiring. In 2016, Forbes magazine published an article on data science jobs that pay $100k or more using the career site paysa.com.[2] Salary is an aspect we did not include. Our data gathering did not uncover any existing analysis that merged data about location and company together. Additionally, we could not find any analysis that was reproducible or flexible enough to search on the user’s specific location or to search for the most job openings across the United States.

**2. Introduction**

Data Science is an emerging field that has quickly grown in popularity. Numerous companies have begun to realize the value that comes with hiring a Data Scientist. With an increase in demand, it is important that those with plans to align their career path with a data science field understand job demand demographics. Unveiling the location density of data science related jobs, the companies embracing the data science community, and the more concentrated data science regions all are key pieces of information that individuals should know prior to pursuing their careers. This study will provide a github link[3] that will contain all the necessary data to reproduce our research with open source tools. Key requirements and specifications are as follows:

**Data Source:** The study uses the Dice.com API. Dice is a technology job search site that caters to Information Technology and engineering positions.

**Data Size:** Total size on disk: 4GB, each pull from Dice API is 0.004GB, each collection has about ~50,000 jobs.

**Hardware/Experimental Specifications:**

No high performance computing is required. This analysis was run on three average laptops:

1. MacBookPro, 2.3 GHz Intel Core i7, 4GB 1333 MHz DDR3 RAM
2. MacBookPro, 3.1 GHz Intel Core i7, 16GB 1867 MHz DDR3 RAM
3. MacBookPro, 2.5 GHz Intel Core i5, 4GB 1600 MHz DDR3 RAM

**3. Background & Existing Work**

While there are many job search sites, our focus was on sites which targeted technology jobs. One of the key selection criteria for choosing an API was that it returned responses in a JSON format. The reason for choosing JSON over XML as the desired response format is that development is simplified, as documents map naturally to modern object-oriented programming languages like python. Further, using JSON and a document-based database like MongoDB removes the complex object-relational mapping (ORM) layer that translates objects in code to relational tables.

Our decision to choose the Dice API over other APIs was its ease of use, availability to the public, and features to search for jobs based on a text string and location.

Table 1: Feature comparison of job search website APIs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Json Response** | **Need API Key** | **Free to public** | **Job search feature** |
| **LinkedIn API** | Yes | Yes | Yes – only basic | Available via 3rd party |
| **Glassdoor API** | Yes | Yes | Yes – only basic | Available via 3rd party |
| **Careerbuilder API** | Yes | Yes | Yes - only XML | Available via 3rd party |
| **Indeed API** | Yes | Yes | No | Available via 3rd party |
| **Dice API** | Yes | No | Yes | Available to public |

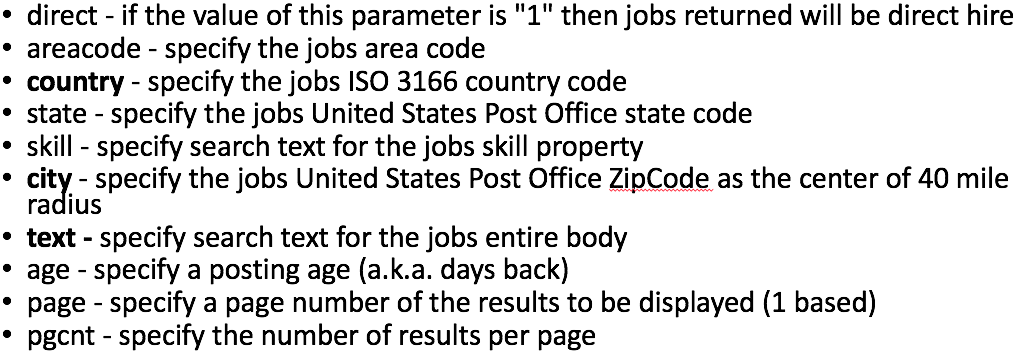
MongoDB was our choice of database as it stores data in JSON-like documents that can vary in structure. Related information is stored together for fast query access through the MongoDB query language and aggregation framework. MongoDB also supports a wide range of drivers for different programming languages. Our choice was to use the PyMongo driver since we use python to retrieve and visualize the data.

**4. Analysis Methods & Results**

The following is an overview of the questions the analysis answers and the technologies used.

**4.1 Dice API**

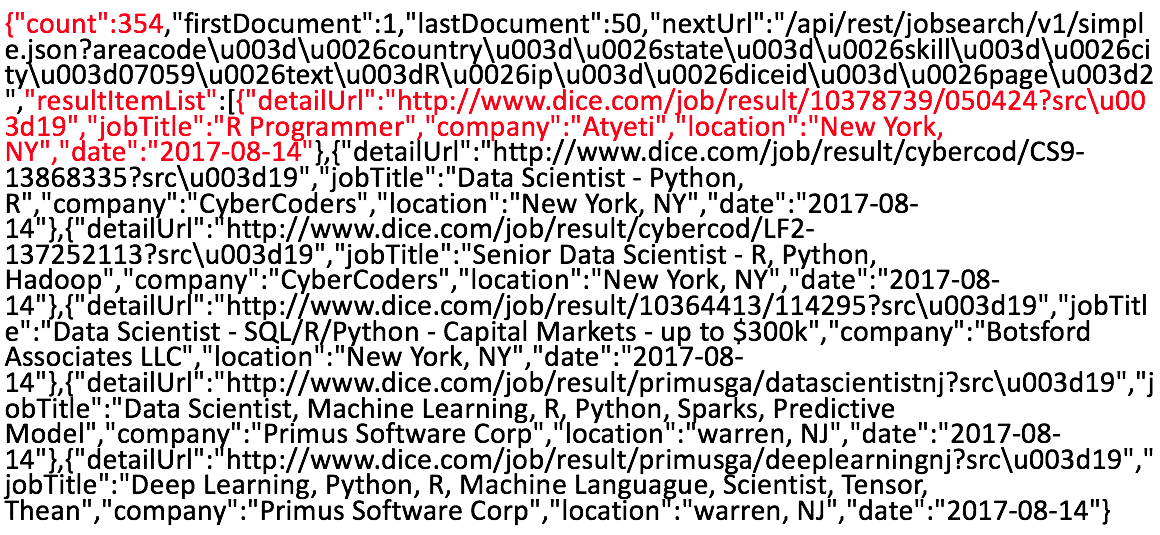
**URL Query:** The Dice API is quite simple and elegant. The query is constructed through a url address that specifies certain search criteria based on optional parameters. The optional parameters are listed below.



Construction of the url address requires the following string: “<http://service.dice.com/api/rest/jobsearch/v1/simple.json>?” concatenated with the filter parameter set to its criteria (i.e. “text=<criteria>”). The use of multiple parameters are separated by an “&” in the url address. To illustrate, using the API to query job postings that contain the word ‘data’ that are in the zip code 90210 would result in the following address:

“<http://service.dice.com/api/rest/jobsearch/v1/simple.json?text=data&city=90210>”

**API Results:** Query results from the API are given in a JSON format. The results consist of search metadata along with nested documents each being a job posting containing particular parameters. A example of the out is shown below:



The first set of red text shows some search result metadata, a count of job postings, while the second red text highlights the parameters returned for one specific job posting document.

**4.2 MongoDB**

**PyMongo Information Input:** The PyMongo package provides methods for connecting to a running instance of mongod process, to perform CRUD operations and to transform data through aggregation method. Data is retrieved from the Dice API using python’s request package, cleaned to retrieve the desired information and stored into MongoDB using the PyMongo driver.

**Aggregation Framework:** MongoDB’s aggregation framework is modeled on the concept of data processing pipelines. Documents enter a multi-stage pipeline that transforms the documents into an aggregated result.

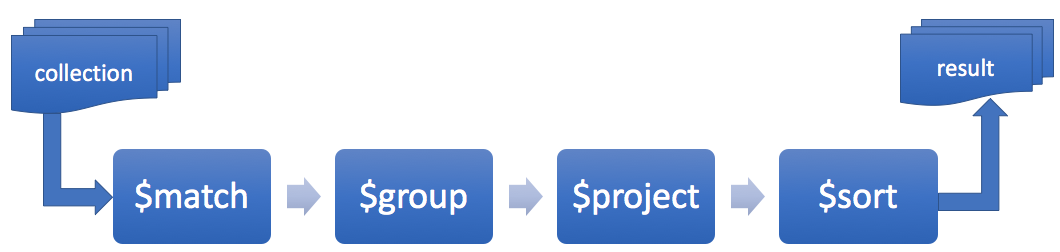


Figure 1: MongoDB aggregation pipeline

**4.3 Visualization Python Packages**

Both Seaborn and Plotly python packages were used to visualize the aggregated query results. Seaborn was utilized to generate simple bar plots that convey the highest number of job postings attributed to a certain parameter while Plotly was leveraged to generate a US map showing job posting density dependent on location. In order for aggregated query results be in an acceptable data type, the Pandas package was used to convert data into data frames.

**4.4 Reproducibility & Repeatability**

Our group created an open github site [3] that can be used to reproduce and repeat this study. The site is organized as follows: a folder for all analysis scripts, a reports folder with detailed information on the study and a readme file. Any user can fork the github site files, and use the readme file to understand how it works. Specific steps for downloading and recreating the analysis are given in a jupyter notebook on the github link [3].

More detailed explanation on how we utilized MongoDB are located in the report folder where we describe the MongoDB queries and overall approach both in this paper and an accompanying MS PowerPoint presentation. Additionally, all python scripts are adequately commented to understand what is being accomplished with each section of code.

**4.5 Results**

Five key questions were answered using the methods listed above:

1. **Number of data scientist jobs per state:** We pass the collection of data scientist jobs retrieved for the entire country through the following stages of the aggregation pipeline:

a. *$match* – filter documents without location

b. *$project* – derive state from location

c. *$match* – filter documents with valid state code

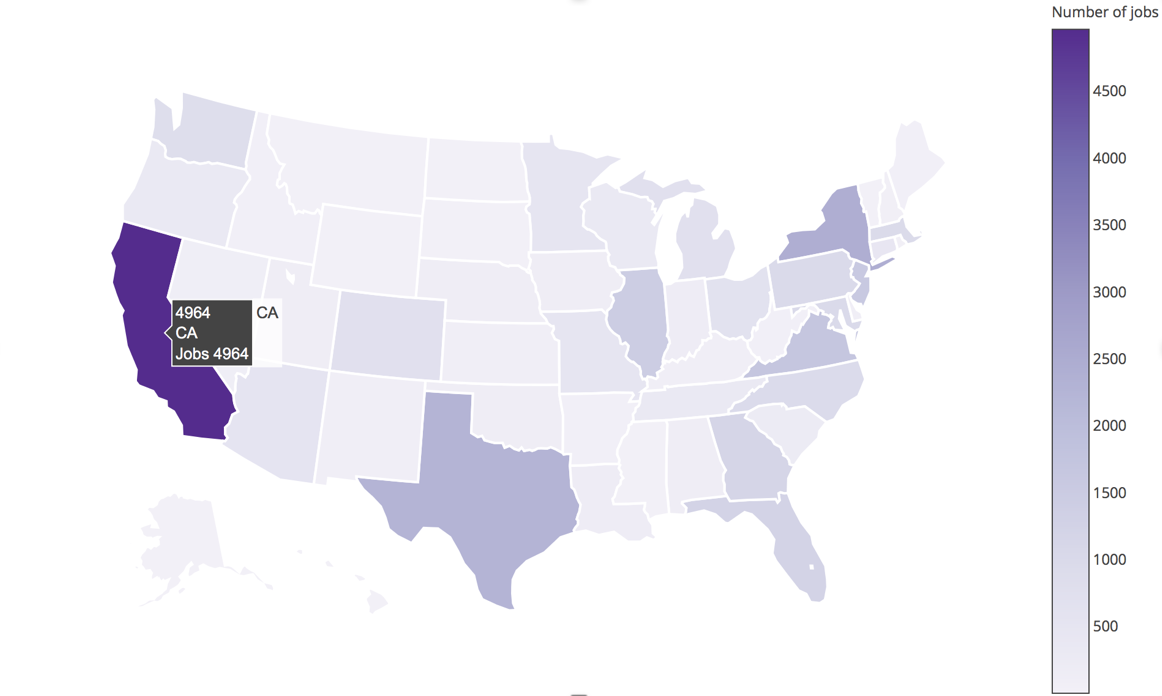
d. *$group* – aggregate documents by state

e. *$project* – rename the column names

Output from the aggregation pipeline is passed to the plotly visualization package. Plotly expects a pandas dataframe, which is easily constructed from the result obtained through the aggregation pipeline.

**Result:** A color heat map of US showing the distribution of data scientist jobs.

Figure 2: Distribution of data scientist jobs



2. **Top company within each state:**  To get the top company hiring within a state, we pass the collection through the following stages:

a. *$match* – filter documents without location

b. *$project* – derive state from location

c. *$match* – filter documents with valid state code

d. *$group* – aggregate by state and company

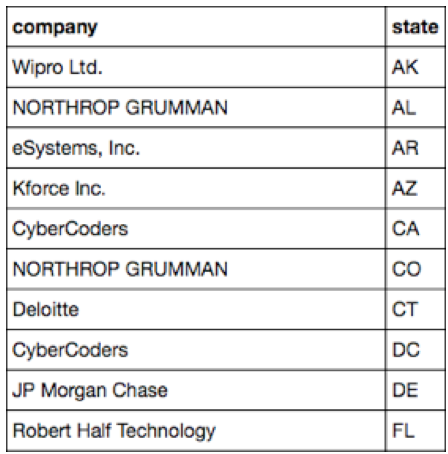
e. *$sort* – arrange by job count descending

*f.* *$group* – aggregate by state; get first element

In the first group stage, documents are grouped by state and company, the groups are sorted based on job count, they are then grouped by state and first element of the group is retrieved. A feature of aggregation pipeline is that multiple grouping stages can be applied to the documents.

**Result:**

Table 2: Displaying ten states with the top company



3. **Who is hiring more data scientists in your vicinity:** We queried the Dice API to get data scientist jobs in a 40-mile radius of a US zipcode. We pass this collection through the following stages:

a. *$group* – aggregate by company and count

b. *$sort* – arrange by job count descending

c. *$project* – rename the columns

d. *$limit* – limit to top 20 companies

Output from the aggregation pipeline is passed to python’s data visualization package for statistical plots – seaborn.

**Result:**

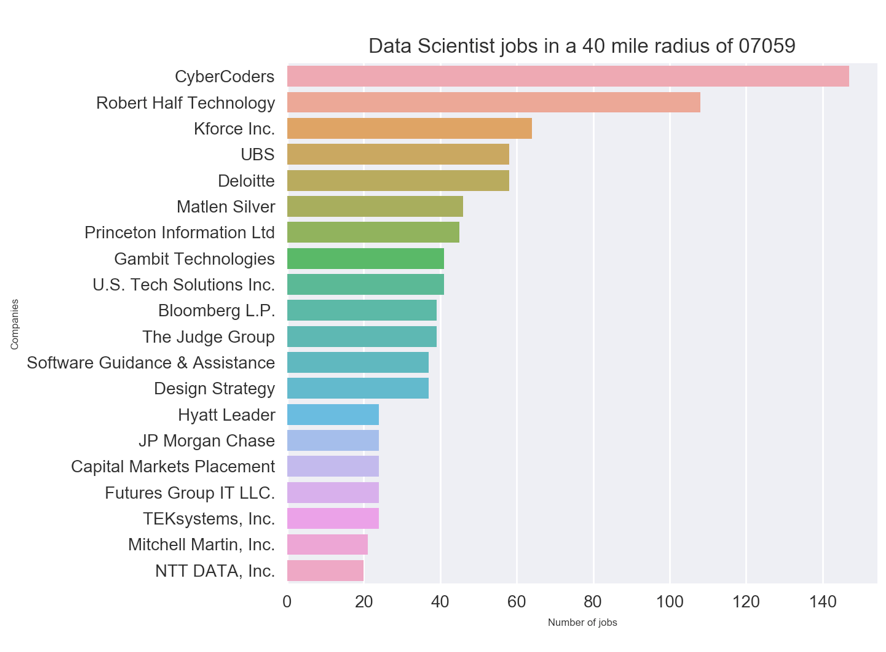
****

Figure 3: Companies hiring data scientists for zip code 07059

4. **Total number of employers hiring data scientists in US:** To get the total number of distinct employers hiring data scientists, we pass the collection of documents through two grouping stages:

a. *$group* – group documents by company

b. *$group* – group again by \_id and count

**Result:** At the time when we conducted this analysis, there were 4428 unique employers hiring data scientists in US.

5. **Comparison of demand for programming skills in US job market:** A comparison of the demand for top programming language skills can help potential data scientists to understand which languages organizations are using to solve their business problems and invest their time in learning those languages.

**Result: java, javascript, python, r, scala, sas vs job count**

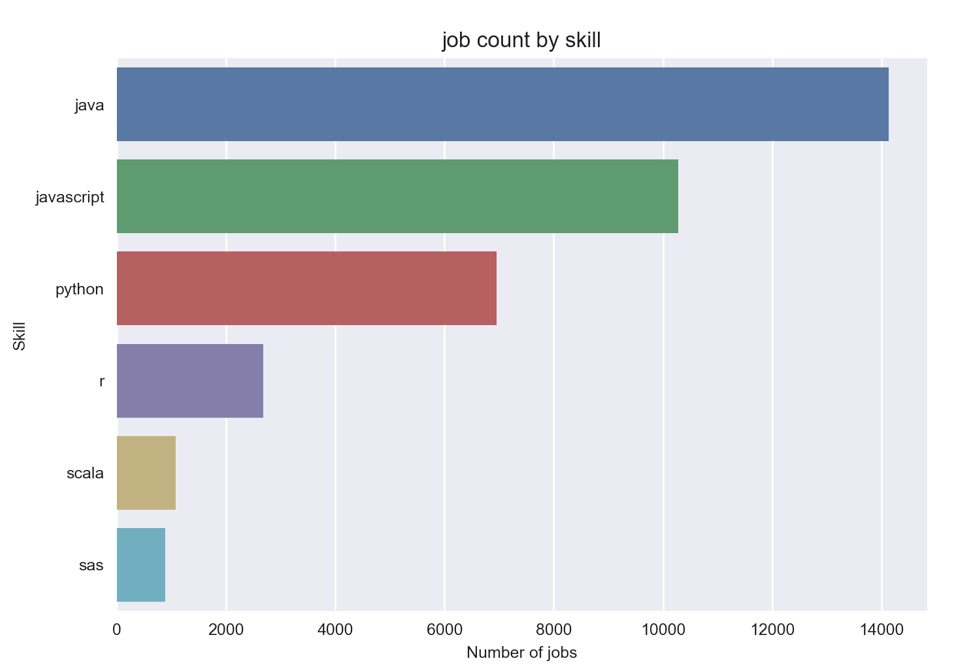
****

Figure 4: Comparison of programming skills in US

**5. Conclusion**

With the use of the Dice API, MongoDB, and the appropriate packages installed into python, a quick yet comprehensive analysis can be constructed to understand certain data science job demand demographics. Insights include job demand per state, top hiring companies per state, top hiring companies within a particular region, unique employers seeking data scientists, and job demand for particular skill sets.

The analysis code also has characteristics that allow for an individual to mold certain parameters to their liking. These features can broaden the a scope of the analysis beyond data science job demand and/or the parameters specified. We enjoyed learning about manipulating MongoDB via Python as well as learning about multiple visualization techniques that will assist us in our future endeavors. Additionally, we are pleased that we have created a useful tool that will help others with their job searches in the future.

### **References**

[1] Alex Salkever, Silk, “*Where the data science jobs are by sector and state*”, [https:///gigaom.com/2014/11/02/where-the-data-science-jobs-are-by-sector-and-by-state](about:blank), Nov 2014

[2] L. Columbus, “*15 Data Scientist Jobs That Pay $100K or More*”, <https://www.forbes.com/sites/louiscolumbus/2016/10/22/15-data-scientist-jobs-that-pay-100k-or-more/#337b76b7626c>, Oct 2016

[3] https://github.com/shravan-kuchkula/TermProject