
Efficient Job Search Using Clustering

Aman Shrestha, Nishith Atreya
Project Advisor - Jiebo Luo

Introduction

- An overwhelming number of candidates are applying for a limited number of jobs.
- For most applicants, it's very difficult to access what kind of jobs openings are appropriate for them based on their background and skills.
- Our project aims to solve these problems by applying clustering techniques to perform efficient job search system that recommends jobs that are best fit for applicants.

The competition is very fierce specifically for people due to lack of industry experience and professional network.

Methodology

- Dataset obtained from Kaggle where a user, Elroy compiled data scrapped from Indeed.com
- Data preparation to fit to clustering algorithms
- Clustering Algorithms used:
 - K-Means Clustering
 - Mini Batch KMeans Clustering
 - Birch Clustering
 - Agglomerative Clustering
 - Affinity Propagation

We obtained the dataset from kaggle where a user Elroy scrapped indeed.com
On this dataset, we performed some data preparation which will be explained in the coming slides

We used the given clustering algorithms on the dataset and tried to find an intersection of all these clustering algorithms so that we can recommend jobs better

Data Preprocessing

- Queried_Salary: Categorized to bins and encoded as columns
- Job_Type: Encoded as columns
- Skill: Scraped specific words from the skills provided and encoded as columns
- Number of Reviews: Replaced 'na' with median of column and used MinMaxScaler to scale
- Number of Stars: Replaced 'na' with mean and used MinMaxScaler to scale
- Description: Scraped common words from description and encoded as columns
- Location: kept common location encoded as columns
- Company_Revenue: Categorized and encoded as columns
- Company_Employees: Categorized and encoded as columns
- Company_Industry: Used common values as columns

K-Means Clustering

- Clusters the data based on centroids whose centers are defined as the mean value of the points
- Main challenge finding right number of clusters
- Used elbow method to decide on a cluster number of 35
- Resulted in clusters having a maximum number of 252 in a cluster and minimum of 48 in a cluster.

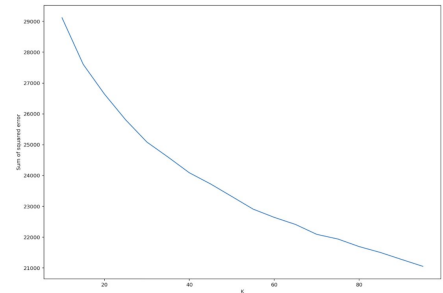


Fig : Result of using elbow method with number of clusters 10 to 100.

This is a popular Clustering algorithm where the data is clustered based on centroids whose centers are defined as the mean value of the points. Since we have multiple features, this algorithm will try and find the mean based on those features when clustering datasets together. The main challenge in this algorithm is to find the right number of clusters. For this, we used the elbow method where we run K-Means with different numbers of clusters and see which one reduces the sum of squared errors the most. Although a significant value wasn't found, we decided to take 35. With this, we have a result of 35 clusters with a maximum of 252 and a minimum of 48 in one cluster.

Mini Batch K-Means Clustering

- A modification of the K-Means algorithm
- Uses mini-batch to reduce time in very complex and large-scale calculations
- The same number of clusters, 35 used
- Resulted in a maximum number of 282 in a cluster and minimum of 45 in a cluster

This is a modification of the K-Means algorithm where mini-batches are used to reduce time in a very complex and large-scale calculations. For consistency, we used 35 clusters again. This algorithm resulted in a maximum number of 282 in a cluster and minimum of 45 in a cluster

Birch Clustering

- Uses a hierarchical data structure that calls CF-tree for increment and dynamically clusters data points
- The same number of clusters, 35 used
- Resulted in a maximum number of 378 in a cluster and a minimum of 78 in a cluster

Balanced iterative reducing and clustering using hierarchies (birch clustering) uses a hierarchical data structure that calls CF-tree for increment and dynamically clusters data points. For consistency in the number of clusters, the same number, 35 was used to fit this model. This algorithm resulted in a maximum number of 378 in a cluster and a minimum of 78 in a cluster.

Agglomerative Clustering

- Organizes objects into a hierarchy using a bottom-up or top-down strategy, respectively
- Start with individual objects as clusters, which are iteratively merged to form larger clusters
- The same number of clusters, 35 used
- Resulted in a maximum number of 354 in a cluster and a minimum of 37 in a cluster

This algorithm organizes objects into a hierarchy using a bottom-up or top-down strategy, respectively. Agglomerative methods start with individual objects as clusters, which are iteratively merged to form larger clusters

For consistency in the number of clusters, the same number, 35 was used to fit this model. This algorithm resulted in a maximum number of 354 in a cluster and a minimum of 37 in a cluster.

Affinity Propagation Clustering

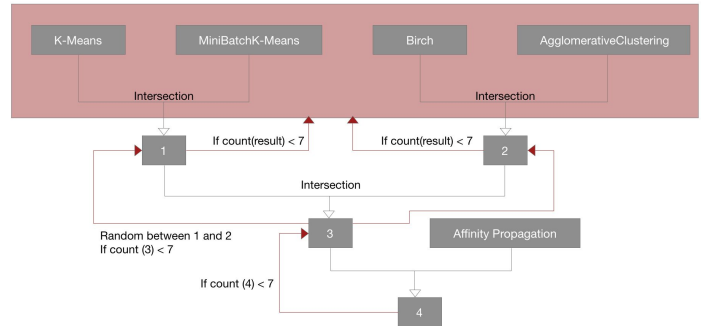
- Based on message passing between data points
- Unlike the algorithms previously used in this project, this model doesn't require a pre-defined number of clusters
- A "preference" parameter changed to -35
- Resulted in 173 clusters with a maximum of 76 in a cluster and a minimum of 4 in a cluster

Affinity Propagation is an algorithm based on message passing between data points. In affinity Propagation, as per Thavikulwat's description "Each item being clustered sends messages to all other items informing its targets of each target's relative attractiveness to the sender. Each target then responds to all senders with a reply informing each sender of its availability to associate with the sender, given the attractiveness messages that it has received from all other senders. Senders absorb the information, and reply to the targets with messages informing each target of the target's revised relative attractiveness to the sender, given the availability messages it has received from all targets. The message-passing procedure proceeds until a consensus is reached on the best associate for each item, considering relative attractiveness and availability"

Unlike the algorithms previously used in this project, this model doesn't require a pre-defined number of clusters. A parameter "preference" has been changed to -35. "Preferences for each point - points with larger values of preferences are more likely to be chosen as examples. The number of examples, i.e. clusters, is influenced by the input preferences value." With this, the algorithm resulted in 173 clusters with a maximum of 76 in a cluster and a minimum of 4 in a cluster.

Combining Clustering Algorithms

1. Intersection was taken of the results of clusters given by K-Means and Mini Batch K-Means (1)
2. Intersection of the values in the list was taken from results from Birch and Agglomerative clustering (2)
3. Using the result from (1) and (2), an intersection list was formed (3)
4. Intersection between (3) and result from affinity propagation was found (4)
5. (4) would be the final output
6. If size of (4) is less than 7, result from (3) was selected
7. If resulting size still less than 7, a random list between (1) and (2) was selected
8. If resulting size still less than 7, a random list between the results of the individual clustering algorithms was selected



Note: The red arrows are followed only after 4 is reached and the count is less than 7.

The information was changed to a format that can be used in the models. Each model put the given row of information in a particular cluster. In order to find the best result, an intersection was taken of the results of clusters given by K-Means and Mini Batch K-Means (1). Similarly, an intersection of the values in the list was taken from results from Birch and Agglomerative clustering (2). Using the result from (1) and (2), an intersection list was formed (3). Then, the intersection between (3) and result from affinity propagation was found (4). This final intersection, (4) would be the final output. However, there were cases where no intersection was shown. In such cases, a minimum value of 7 was sought for. Therefore, result from (3) was selected in that instance. If the list was still smaller than 7, a random list between (1) and (2) was selected. If the list was still smaller than 7, then a random list between the results of

the individual clustering algorithms was selected.

Experiment

- Every row in the previous dataset was put through the algorithm
- The information was changed to a format that can be used in the models
- Each model put the given row of information in a particular cluster
- Applied the algorithm of combining clustering algorithms
- Found a minimum cluster of 7 recommendations and maximum of 378 recommendations
- A subjective view of these lists supports our project
- A better metric will be necessary in the future

Aspects like salary, company reputation, industry, location etc, were primarily focused to find information about the kind of job/company they were looking for.

Subjects were also asked to include any keywords that they thought might be significant for their job search (for e.g -- education or specific field that they are interested in -- cloud, finance, research etc.)

Conclusion

- Searching for job is an intimidating task and many features need to be considered
- This project applies different clustering algorithms to dataset obtained from indeed.com on kaggle
- An intersection of the resulting lists is used as final recommended list
- This list is intended to consider the various features that need to be noted while searching for jobs

The key difference between our approach and the existing approaches is that we try to optimize the result by finding the maximum intersection between results i.e the resulting output will consist of the list of same decisions made from different clustering techniques.

Future Work

- More time and resources should be spent in finding quality data and data preparation
- Use clustering techniques to generate clusters based on applicant's data like age group, experience in the industry etc and it can be integrated with our current approach for better results.
- Extend this project by including other jobs in addition to data science jobs

Future Work -- 1) Spending more time and resource on data collection and preparation

2) Use applicant's data to generate better clusters that can lead to better results

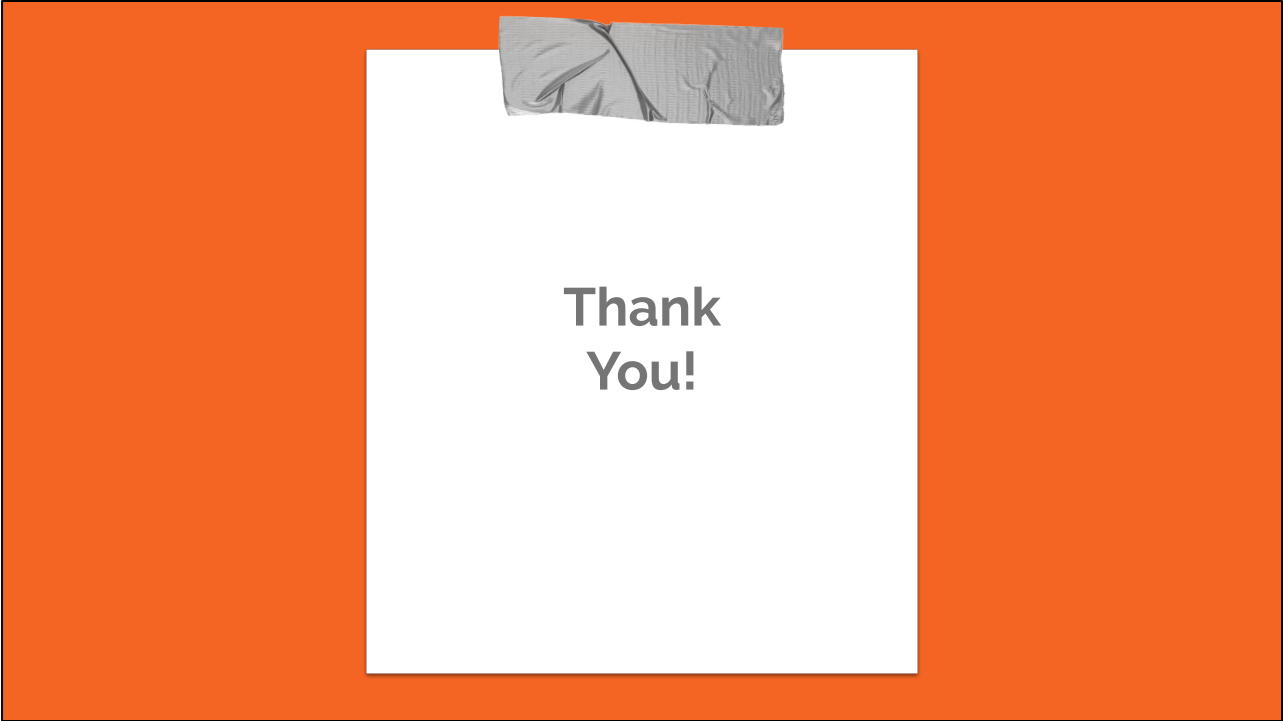
3) We focused this project on jobs in data science, but future works can be done in other industries and fields.

References

- Elroy. 2018. Indeed Dataset - Data Scientist/Analyst/Engineer). <https://www.kaggle.com/elroyggj/indeed-dataset-data-scientistanalystengineer>.
- Bo Li, Yibin Liao, and Zheng Qin. 2014. Precomputed Clustering for Movie Recommendation System in Real Time. *Journal of Applied Mathematics* 2014 (2014), 1–9. DOI:<http://dx.doi.org/10.1155/2014/742341>
- Jain, H. and Kakkar, M. 2019. Job Recommendation System based on Machine Learning and Data Mining Techniques using RESTful API and Android IDE. 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (Jan. 2019). DOI:<https://doi.org/10.1109/CONFLUENCE.2019.8776964>
- Zhou, Q. et al. 2019. Job recommendation algorithm for graduates based on personalized preference. CCF Transactions on Pervasive Computing and Interaction. 1, 4 (Nov. 2019), 260–274. DOI:<https://doi.org/10.1007/s42486-019-00022-1>.

References

- Cintia Ganesha Putri, D. et al. 2020. Design of an Unsupervised Machine Learning-Based Movie Recommender System. Symmetry. 12, 2 (Jan. 2020), 185. DOI:<https://doi.org/10.3390/sym12020185>.
- Nilashi M. et al. 2016. A New Method for Collaborative Filtering Recommender Systems: The Case of Yahoo! Movies and TripAdvisor Datasets (2016).
- Han, J. et al. 2012. Data mining. Elsevier/Morgan Kaufmann.
- Thavikulwat 2008. Affinity propagation: a clustering algorithm for computer-assisted business simulations and experimental exercises. In Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference (Vol. 35).
- Pedregosa et al. 2011. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830. Retrieved May 7, 2020 from <http://jmlr.org/papers/v12/pedregosa11a.html>



**Thank
You!**