A Chance to Work

Understanding the composition of foreign workers pursuing specialty occupations on the United States H1-B visa

Alexander Buddenbaum
Georgia Institute of Technology
Shenzhen, China
alex.budd@gatech.edu

Qinrui Li
Georgia Institute of Technology
Shenzhen, China
qli449@gatech.edu

Tianyu Li
Georgia Institute of Technology
Shenzhen, China
tli303@gatech.edu

Chuanqi Liu
Georgia Institute of Technology
Shenzhen, China
cliu732@gatech.edu

Tianshu Tao
Georgia Institute of Technology
Shenzhen, China
ttao35@gatech.edu

1 INTRODUCTION

The H1-B visa program allows employers in the United States to hire temporary foreign workers specializing in one of many key occupations. To date, numerous studies have been conducted on recent years' certification data in an attempt to analyze hiring trends. However, the usefulness of these studies remains to be questioned as they generally lack interactive visualizations for the user to gain insight from this information. In this study, we seek to understand recent hiring trends among US employers as well as the makeup of applications for these roles to project economic sectors with growth potential, creating meaningful visualizations for potential work visa applicants to interpret current data, and predict the outcome of future applications.

2 RELATED WORK

As annual H1-B application records are publicly available through the US Department of Labor[cite], this data has been a popular topic for machine learning researchers as well as online enthusiasts. The topic has also inspired several versions of truncated datasets that have been aggregated by several websites as a resource for international applicants, and basic analyses appear frequently in academic journals. Interestingly, none of these studies or websites have successfully integrated the database, visualizations, and prediction algorithms under one application. We have considered several methods for data cleaning and integration. H1-B application data from the US DOL total over 3 million records from 2017 to 2021. Chatterjee et al.[Chatterjee], use Python packages Pandas and Numpy for data cleansing, including the completion of incomplete data in H1-B application data, column name renaming and subset selection. Specific operations can be done by calling Python packages. Looking closely at their data, however, we found that not only is it from a non-official and unverifiable source; their records are also severely truncated, and did not effectively consolidate job titles with negligible spelling variations. Perhaps this truncation and lack of cleansing was out of convenience, but unfortunately this was the case for the majority of studies we read during preliminary research. As the datasets among the different years generally only differ by a few column names while retaining similar organization, we find it more convenient to use open-sourced applications such as OpenRefine to clean the data. Dombe et al.[Dombe] use SQLite for its lightweight data storage and access. Several online databases exist providing similar information. The most accurate and comprehensive websites include H1-B Grader and One Point Three Acres, which have aggregated US Department of Labor statistics into textual tables which allow for filtering based on attributes such as country of origin, job title, sponsoring employer, and salary band[cite][cite]. These are relatively complete relational database systems with interactive visualizations. The data is also collected directly from the US DOL and up to date. As the data provided by US DOL is in XLS format and consists of over 26 attributes in its initial form, the SQL-style database used by these websites appears most appropriate. However, these websites are unable to predict application outcomes for users, thus the users must perform their own extrapolations based on their subjective interpretations of recent trends. The majority of the research to date compares the performance of several machine learning algorithms in predicting application outcomes. Swain et al.[Swain] compared the accuracy of random forest, k-means clustering, and logistic regression algorithms for predicting H1-B application acceptance based on roughly 3 million data points from 2011 to 2016, and Chatterjee et al. propose an artificial neural network. They also show a variety of static charts, but the charts only compare across one dimension at a time. The data is also nearly ten years old. While these studies set us in a good direction for doing some basic analysis, we seek to augment them through an interactive visualization allowing the user to view results based on multiple attributes such as salary and location. The method espoused by Roy Raunak [Raunak] uses the collected feedback in applying the analytic hierarchy process and entropy weight method to evaluate the data analysis and prediction model. This method decomposes the decision-making problem into different hierarchical structures according to the order of general objectives, sub objectives at all levels, evaluation criteria and specific standby choice. Then, the problem is reduced to the determination of the relatively important weight of the lowest level, such as schemes and measures for decision-making, relative to the highest level - the overall goal or the arrangement of the relative advantages and disadvantages, so as to judge whether the prediction scheme we provide can meet the needs of users on whether to apply for H1-B visa.

3 PROPOSED METHOD

In order to show comprehensive aspects of H1-B statistics to potential users, we propose a set of interactive visualizations, employing multiple widely used D3.js charts implemented in Vue and Flask. The data is publicly available, and all the tools are open source and can be deployed from a consumer-grade computer. As such, this project requires no special funding. We retrieve H1-B application data from years 2017 to 2021 from the US Department of Labor[DOL]. All records from these five years in their original XLS format totaled over 1.5Gb. By accessing data directly from its original source, we can guarantee our data will have the highest possible integrity for more accurate evaluation and predictions. There are some inconsistencies in the naming and ordering of columns from year to year, as well as the addition of columns that are not pertinent to our analysis. We use OpenRefine to clean and standardize the data, dropping columns that are not pertinent to our analysis, thus making the file sizes manageable before consolidating into CSV files aggregated by year. Users will select the attributes they are interested in on our portal page. Their request will be passed to our backend REST API, implemented by Python Flask. The API will then send the corresponding statistics back to the front-end interface. The user interface will then process the data and render the chart on the newly-directed page, where the user can view and interact with the chart using their cursor. The front-end application, based on Vue and D3.js, will have an interactive interface to provide the user rich choices of viewing, comparing, selecting and searching by certain fields. As mentioned in Related Work, other implementations exist on the Web. Our implementation is distinct from these earlier implementations by including several innovations or improvements. Our system provides users with more selections and chart types that are more relevant to the user, such as a US choropleth with state-by-state comparison. We provide more relevant details with the ability to combine search parameters such as state and salary. We also introduce H1-B application outcome prediction through machine learning methods, a feature we have yet to find on any frequently accessed website. A list of innovations and improvements is listed in Appendix B. We also construct a model that can predict the probability that a given application will be certified by the DOL. As mentioned above, the data requires minimal cleansing and feature engineering, dropping some unimportant features by the results of correlation and manual selection. We then split the data 70%-30% for training and testing. We define our task as a binary classification, setting the "Certified" status as 1 and all other non-certified statuses as 0 to train our model. For the model, we choose a gradient boosting decision tree model named LightGBM because the data is nonlinear as we believe that this tree model shows improved performance over some of the commonly applied linear models.

4 EXPERIMENTS AND EVALUATION

Our application is currently at the alpha state and therefore difficult to evaluate beyond limited use. As of April 1, we have successfully launched the web application in a local testing environment using Flask for the backend and Vue for the frontend. We have preprocessed all data, reading it and producing a choropleth chart of applications by state. The choropleth (Figure 1) shows the greatest concentrations of applications in California, Texas, and New York respectively, and are unchanged for each year. As mentioned in our proposal, moving forward, we will implement the multiple criteria filters for the choropleth as well as implement an interactive bar chart for single criteria comparisons. We also will implement a SQLite database for fast querying and include an interface for the user to run predictions through the machine learning algorithm. We have followed the original work distribution, which is distributed evenly among all 5 team members, with each member responsible for one component as well as making limited contributions to other components (see Appendix A).

5 CONCLUSION

The progress report serves as a checkpoint halfway through the project duration. During the first phase of this project, we have successfully applied the skills and leveraged the tools touched upon in class to devise a novel solution for a real need among international students. In an effort to quickly become proficient in this domain, we have also studied the intricacies of the US work visa application process, familiarizing ourselves with the geography and industry makeup of the United States. We are on schedule to complete our experiment and have a working application by our mid April deadline, at which time we look forward to testing its efficacy with a student survey.

A WORK DISTRIBUTION

- Project coordinator: Tianshu
- Data collection, cleansing, standardization, feature selection: Alexander
- ML algorithm design: Chuanqi
- UI design, frontend implementation: Tianshu
- Database design, backend implementation: Tianyu
- Documentation: Alexander, Qunrui, Chuanqi, Tianshu

All team members have contributed similar amount of effort.

B INNOVATIONS

- More selections and chart types that are more relevant to the user, such as a US choropleth with state-by-state comparison
- Ability to combine search parameters such as state and salary
- Interactive H1-B application outcome prediction through machine learning

REFERENCES

[1] H-1B Program. *United States Department of Labor.* (2022). Retrieved 1 April 2022. https://www.dol.gov/agencies/whd/immigration/h1b.

[2] Chatterjee, P., Velpuru, M. S., & Jagadeeswari, T. (2021). Success of H1-B VISA Using ANN. *In Machine Learning and Information Processing (pp. 491-499).* Springer, Singapore.

[3] H1B Database 2022 - Sponsors, Salaries, Approvals, Grades!. (2022). Retrieved 1 April 2022.
https://h1bgrader.com/

[4] Visa Tracker - 1Point3Acres. (2022). Retrieved 1 April 2022.
https://visa.1point3acres.com/.

[5] Dombe, A., Rewale, R., & Swain, D. (2020). A deep learning-based approach for predicting the outcome of H-1B VISA application. *In Machine Learning and Information Processing (pp. 193-202).* Springer, Singapore.

[6] Swain, D., Chakraborty, K., Dombe, A., Ashture, A., & Valakunde, N. (2018, December). Prediction of H1B Visa Using Machine Learning Algorithms. *In 2018 International Conference on Advanced Computation and Telecommunication (ICACAT) (pp. 1-7).* IEEE.

[7] Roy, R. (2021). Data Analysis of H1B Visa Applications.

[8] LightGBM. *Microsoft*. Retrieved 1 April 2022. *https://github.com/microsoft/LightGBM.*

[9] Flask. Retrieved 1 April 2022. *https://flask.palletsprojects.com/en/2.1.x/.*

[10] Vue. Retrieved 1 April 2022. *https://vuejs.org/.*

[11] OpenRefine. Retrieved 1 April 2022. textithttps://openrefine.org/.

[12] D3: Data Driven Documents. Retrieved 1 April 2022. *https://d3js.org/.*