

In-Depth Analysis of Relevant Factors for University Selection

Sharanya Ravichandran, Disha Talreja, Aravind Ramalingam, Akshay Rajvanshi
sharavi@iu.edu, dtalreja@iu.edu, aramali@iu.edu, aksrajva@iu.edu

Abstract—In a society where higher education is a near necessity, examining the true cost of university is an important factor to consider when determining the benefits of higher education. Selecting a university with an appropriate major is usually an overwhelming experience for a majority of incoming high school students. This paper examines factors such as program cost, public vs private universities, their admit ratio, SAT score and their ROI based on the major preference which a student doesn't know about just after coming out of the high school. Using in-depth analysis, we strive to help the student determine the best university for them.

I. INTRODUCTION

The economics of higher education is a popular area of study in recent years, as advanced degrees have become a near necessary component of long run career success. This increase in value has been met with an exponential rise in cost, causing the rise in university tuition to outpace inflation over the past decade.

While earning an undergraduate degree is a sizeable investment, the costs are analyzed from an economic standpoint. Usually, students focus on factors such as cost, diversity, major, and their salary after graduation in the long run. This project aims to provide a list of universities to the student based on their preferences and their eligibility. The analysis in the project focuses on taking certain search parameters into consideration, and exploring the data set to give a granularized suggestion of universities to the user.

The main concern here is that there are many web applications that provide a similar kind of service, but they are quite one dimensional. We have decided to focus on multiple factors that need to be considered and gives a consolidated result that incorporates all the relevant factors and a multi-dimensional view towards selecting a

university.

The following key areas formed the main text of this paper.

- Data consolidation
- Data processing and cleaning
- Setting up a cloud environment
- Setting up the visualization framework
- Data loading and creating visualization
- Developed a minimal web application for the user

A. Data consolidation

We searched across the web for appropriate data sets and in that process, we came across multiple data sets for university admissions in Kaggle including different domains such as tuition fee, majors, diversity, salaries and SAT scores needed to get into the university. Eventually, we combined 4 data sets which included all the relevant information for us to make a good prediction system for the user.

B. Data processing and cleaning

This is an important step to pre-process any data set. We found certain incomplete fields in the data set, which had to be processed to enable successful upload of data into elastic search. We created Python scripts which helped us to clean the data.

We replaced empty fields with either 0's or empty string depending on the type of the field. We also found several fields having the value NAN which made it difficult to perform calculations to perform further analysis. Most of the fields corresponding to the tuition fee or salaries were strings with the dollar sign in them. We removed the dollar sign from all of the currency fields and converted them to integers to enable various calculations on them.

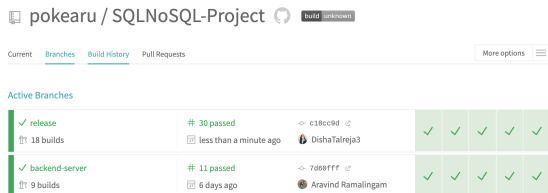
C. Setting up a cloud environment

In order to create a Web application to that is provides high availability and is fault tolerant, we looked into setting a cloud environment that would support these requirements. We decided on using Jetstream VMs as our cloud provider to configure our environment. For orchestration we set up a kubernetes cluster, using Ansible scripts and Kubeadm. Kubernetes abstracts the underlying infrastructure layer, this enabled us to focus on building the application, then deploy them to any environment.

We decided on going with Docker as our container bed and set up our cluster to run Dockerized pods. Docker enabled us to easily pack, deploy, and run our application as a lightweight, portable, self-sufficient container, which can run virtually anywhere. In addition, Docker containers are easy to deploy in a cloud.

Furthermore, we set up a CI pipeline using Travis CI. Travis CI supported our development process by automatically building and testing code changes, providing immediate feedback on the success of the change. Travis CI also helped automate other parts of your development process by deploying changes to Dockerhub and giving us build notifications.

Finally, we configured the Master VM to expose a public DNS that can be used accessed by the users.



The screenshot shows the Travis CI interface for the 'SQLNoSQL-Project'. It displays a table of active branches with their build status and history.

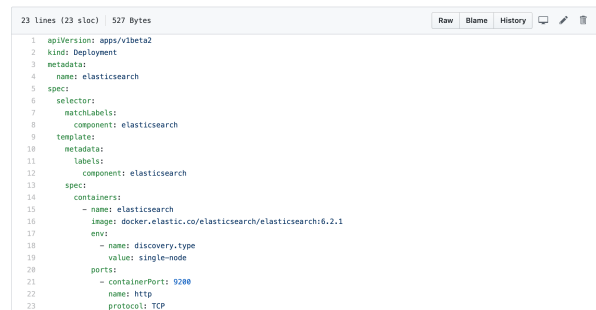
Branch	Status	Build History	Build Time	Build Date
release	30 passed	18 builds	less than a minute ago	30 builds
backend-server	11 passed	9 builds	6 days ago	11 builds

D. Setting up the visualization framework

As mentioned above, we setup Docker and Kubernetes as our environment to enable containerization and have an orchestration tool to maintain robustness and fault-tolerance. We decided to make use of the ELK stack which consists of the ElasticSearch NoSQL database and the Visualization software Kibana.

The ELK Stack helps by providing users with a powerful platform that collects and processes data from multiple data sources, stores that data in one centralized data store that can scale as data grows, and that provides a set of tools to analyze the data. Furthermore, to make use of the given cloud environment, we wrote deployment YAML scripts to add ElasticSearch and then Kibana images to our Kubernetes environment.

The *kubectyl apply* commands enable us to create deployments and services. After creating these essential images, we turned them into a load balancer service which enabled our team to open and collaborate together on these platforms when creating visualizations, and loading the data set into these services.



```
1 apiVersion: apps/v1beta2
2 kind: Deployment
3 metadata:
4   name: elasticsearch
5 spec:
6   selector:
7     matchLabels:
8       component: elasticsearch
9   template:
10     metadata:
11       labels:
12         component: elasticsearch
13     spec:
14       containers:
15         - name: elasticsearch
16           image: docker.elastic.co/elasticsearch/elasticsearch:6.2.1
17           env:
18             - name: discovery.type
19               value: single-node
20           ports:
21             - containerPort: 9200
22             name: http
23             protocol: TCP
```

E. Data loading and creating visualization

In order to create visualizations using kibana's dashboard we loaded the data into elastic search. Elastic search stores data in the form of indices. Four different indexes were created. The indexes were populated by using the *File Data Visualizer* feature of kibana which enables the import of csv files. While importing ,advanced options were set to modify the date type of various fields to be set to the type 'keyword'. They are typically used for filtering , for sorting, and for aggregations. Keyword fields are only searchable by their exact value.

The index corresponding to the college tuition data set required an additional column to calculate the admit ratio. We used the script fields feature of kibana to calculate admit ratio. Script fields compute values on the fly and can work on fields that are not stored and allow to return custom values to be returned (the evaluated value of the

script). After setting all the indices we created visualizations using the ELK stack's visualize feature.

Visualize enables you to create visualizations of the data from your Elasticsearch indices, which you can then add to dashboards for analysis. Kibana visualizations are based on Elasticsearch queries. By using a series of Elasticsearch aggregations to extract and process your data, you can create charts that show you the trends, spikes, and dips you need to know about.



F. Developed a minimal web application for the user

For our web application, we decided to implement it with HTML, Bootstrap and JavaScript as our development stack. We created 3 main landing pages, such as Graphs, Prediction and Analysis. The *Graphs* page, focuses on the general visualizations that would help a user to become familiar with the factors relevant for selecting a university. The visualizations perform a comparison on multiple metrics such as SAT score, admit ratio, tuition fee etc.

The *Prediction* page provides the user the ability to run our analysis framework based on search parameters defined above. These parameters are obtained as input based on the user selection. We dynamically form a request object based on the user selected search parameters, and request the back end for the response from running the query on the data set. After the response has been fetched, we run a recursive data-processing algorithm to coalesce the data from multiple data retrieving requests in order to provide the user with the top 10 university suggestions that matches the criteria based on the user input. The *Analysis* page, shares our findings from analyzing our data sets. We have provided the user with extensive information about how college decisions can affect ones

career. The study shows how employment salary can vary based on multiple factors, depending on the college decisions that one might make for their Undergraduate studies. We try to focus on Major selection, i.e. how the undergraduate major can affect your salary on different stages in a students career.

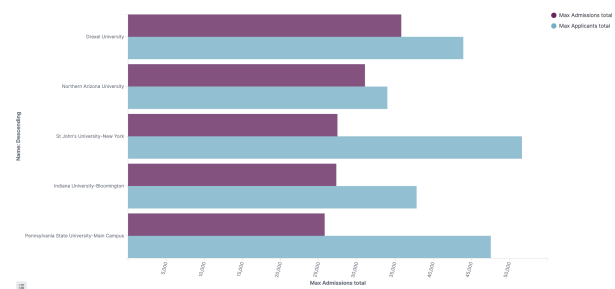
For our backend server, we decided to go with Go. Golang is compiled to machine code, and it naturally outperforms languages that are interpreted or have virtual runtimes. Go servers also compile extremely fast, and the resulting binary is very small and light weight to deploy. Our API server compiled in seconds and produced an executable that we then containerized using Docker. The builds were carried out with our Travis pipeline and deployed to Dockerhub.

Finally, our static UI files were packaged along with our Go server and they were deployed in our kubernetes cloud infrastructure.

II. ANALYSIS OF DATA SET

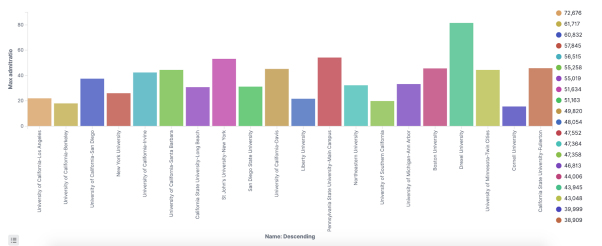
A. Calculating the success rate of getting into a university

The college tuition data set had two columns corresponding to the total number of applicants that applied to a university and the total number of admissions that were given out. We sorted the total number of applicants and admissions in the descending order to create visualizations which would help us given an idea about the admit rate. The results of the analysis are shown in the below figure.



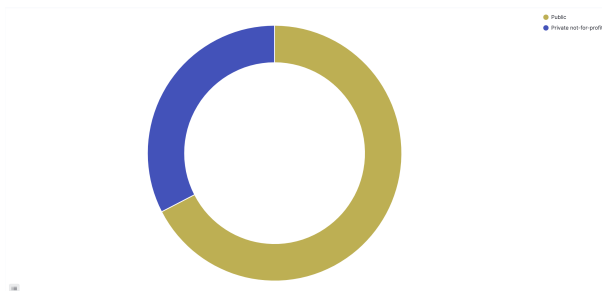
Based on the visualization created on the maximum number of admissions vs applicants we pre-computed the admit ratio and created

a graph which shows the admit ratio for every university ordered by the number of applicants and it was observed that college's like UCLA had the highest number of applicants but with the lowest number of admits. Also universities like Drexel and Pennsylvania have the highest admit ratios. Thus helping a student decide his odds of success against each university based on the admit ratio. The following graph was generated based on the admit ratio of each university.



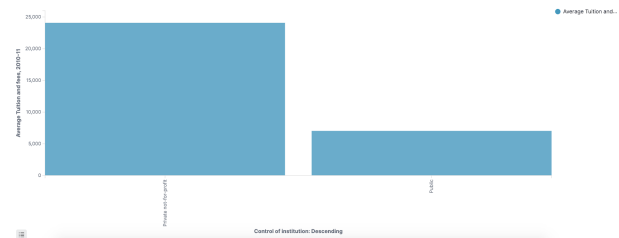
B. Choosing a university based on the control of institution

We wanted to deep dive into the number of students that enrolled in a university based on the control of institution. i.e we wanted to know the number students who enrolled in a public or a private college and we observed that a large number of students preferred joining public universities over private universities. The results of the above observation can be found below. The blue part of the pie chart corresponds to the private institutions.

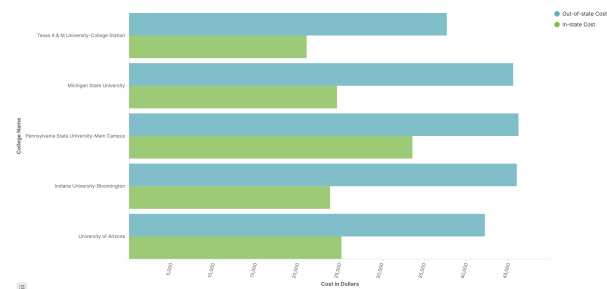


We wanted to know the reason behind why a student preferred a public university over a private university and hence we computed the average

tuition cost of joining a university based on the control of institution namely public or private. We then create a pie chart depicting the same. The visualization depicted that the private universities had higher tuition rates compared to the public universities. Thus being the main factor behind the consideration of the control of institution. The figure below is the visualization of the average tuition cost by control of institution.

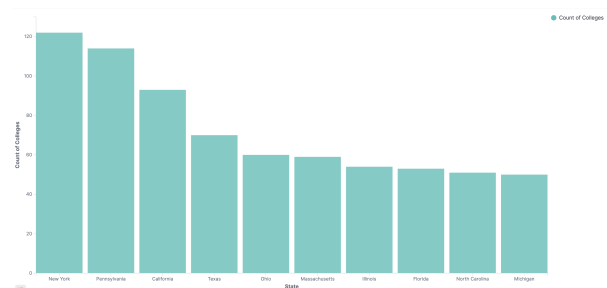


C. Choosing a university based on the instate vs out of state tuition cost



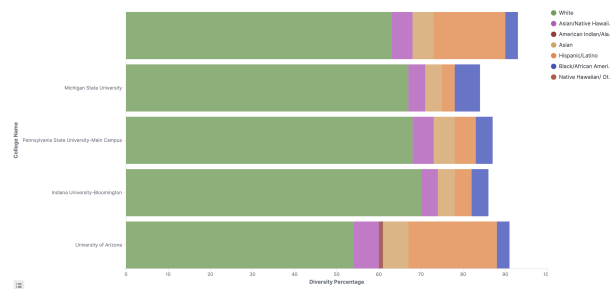
Based on the comparative graph between Out of State and In State College Cost, we realised that moving out of state escalates the expenses by at least USD 10,000. Analyzing the graph further, Indiana University Bloomington showed the maximum difference(USD 20,000) in the cost for In State Students and Out of State students.

D. Choosing a university based on colleges in a region



The above bar graph concerning various states against the number of universities depicts that there are over 120 universities in New York which keeps it at the top in the race. Followed by NY are Pennsylvania, California and Texas with approximately 110,90 and 70 universities respectively.

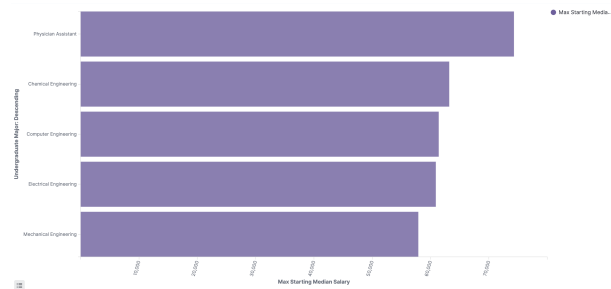
E. Choosing a university based on the Diversity



We wanted to find out the top universities with the maximum diversity. Based on the visualization generated for Diversity against University, we found out that Texas AM promotes diversity the highest with students from different racial backgrounds. Michigan State University and Pennsylvania State University were the two top contenders in this competition.

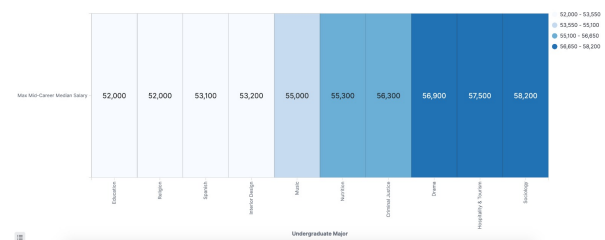
F. Impact of major on Starting Salary

We created a graph for calculating the maximum starting salary based on the major. And we observed that Physician's Assistants have the highest starting salaries, followed by engineering majors. Nursing major have salaries higher than one might expect. Next in the line are majors such as Business Management, Political Science, Marketing. At the bottom, we have the art majors: Music, Drama, Art, followed by Education, Religion, and Spanish. Graph of the above observation can be found below.



G. Impact of major on Mid Career Salary

We created a graph for calculating the maximum mid career salary based on the major. And we observed that Engineering majors have the highest starting salaries, followed by physics and economics majors. Graph of the above observation can be found below.



III. INDIVIDUAL CONTRIBUTIONS

We divided our project into four main parts - Data Consolidation and Processing, Setting up of the cloud and visualization infrastructure, Data analysis and at the end, Website creation. For this project, all the team members worked on developing and designing the approach from selecting the data set until deciding the content of the deliverable.

- 1) **Disha Talreja:** Helped with creating visualizations in Kibana and the designing of the UI design including putting together the UI components, visualizations, and interactive content for the the users.
- 2) **Sharanya Ravichandran:** Helped with the creating the front end logic of the website along with the sending of the data object to the back end and processing the data

from the ElasticSearch server with the help of data collating algorithm. She also played an important role in the helping with creating the visualizations and performing data analysis on the dataset.

- [2] <https://www.kaggle.com/wsj/college-salaries>
- [3] <https://www.kaggle.com/jessemostipak/college-tuition-diversity-and-pay>
- [4] <https://www.elastic.co/guide/en/kibana/7.6/docker.html>

- 3) **Aravind Ramalingam:** Helped with the creation of the cloud infrastructure and the CI pipeline in the system. Also, he helped with the frontend JavaScript logics. Along with that, he played a role in creating the Go backend for fetching the data from Elasticsearch with the help of the Elasticsearch API.
- 4) **Akshay Rajvanshi:** Helped with the initial data preprocessing with the Python scripts and data analysis. After that, he worked on creating the ElasticSearch and Kibana deployments on the Kubernetes cluster and worked in the Go backend of the prediction system with getting the data from the respective indexes and formulating the functions to make them dynamic.

IV. CONCLUSIONS

Having a college degree and the skill sets that come along with it often times leads to the ability to pick and choose better career paths. College graduates typically enjoy a lower unemployment rate, also they can transfer skill sets and knowledge across a broad range of industries and organizations. Hence the simple decision of what college to attend can be a life changing one.

Furthermore, college and university majors are designed to teach you what's been done before. One must know what exists now if you wish to build on top of it. For most people, majors hold great value in regards to the skills they need for a fruitful future career.

Based on our studies we can conclude that public colleges are more suited for a budget friendly perspective. An Engineering degree would prove to pay back well, thus having good return on investment. Finally, SAT do play a crucial role on the whether you get into the college of your choice.

REFERENCES

- [1] <https://www.tuitiontracker.org/>