AN ADVANTAGEOUS AND USER-FRIENDLY MOBILE PROGRAM TO BENEFIT STUDENTS IN SEEKING THEIR SUITABLE COLLEGES THROUGH THE USE OF WEB SCRAPING, MACHINE LEARNING, AND FRONTEND DESIGN

Sibo Tao[1], Yu Sun[2]

[1] Yorba Linda High School, 19900 Bastanchury Rd,
Yorba Linda, CA 92886
[2] Computer Science Department, California State Polytechnic University,
Pomona, CA 91768

ABSTRACT

The abstract describes a research project on the importance of college decision-making for high school students, and how college reviews and reflections can help students make informed decisions. The project focuses on developing an online platform that allows students to view reviews and reflections of former and current college students, covering not only academics, but also other aspects of college life such as culture, geographical location, campus, and tuition. The project includes two experiments, one on the accuracy of a sentiment analysis model and one on the user experience of the software frontend design. The results show that the sentiment analysis model is able to accurately predict positive and negative sentiment comments, but struggles with neutral comments due to data imbalance. The user experience experiment identifies several areas for improvement in the app's UI design. Overall, the research project seeks to provide high school students with a valuable resource for making informed college decisions.

KEYWORDS

Data, Analysis, Convenience, Reflect

1. INTRODUCTION

The growing pursuit of education among high school students has steadily grown in the past centuries [1]. Educational opportunities offered by a university have increased in value as they transfer to essential skills and productive work in society. Therefore, college decisions that students face during their senior year become increasingly important; students not only have to spend at least two years of their lives there but the style and quality of the university also directly influence students' future success. Choosing a college that suits the individual is crucial; individuals who attend a college that doesn't match their work style, culture, or expectations may find it difficult to endure their time there, leading to inefficient learning. Thoroughly understanding all aspects of a college before making a decision allows one to avoid obstacles and challenges that one will meet. Consequently, my research and project topic — College Reviews

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 137-147, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130512

and Reflections — is vital, as it will lead the student population down the correct route. Upcoming college students can view former and current students' opinions that describe various aspects of the colleges before making final decisions. In this way, students can fully evaluate a specific college and determine if it is the most favorable path based for them.

College websites and other services provide information to prospective students, particularly regarding academic abilities and credibility, allowing users to consider their college options based on academic qualities [2]. However, academics are not the only important characteristic, as other components of college make up a student's years of college life: culture, geographical position, campus, and tuition. It is necessary to look at student reviews to create a reputable and accurate reflection of a college, as they provide insight into its extracurricular qualities. Although the internet offers some resources for students' reviews and reflections, most resources lack sufficient data, which is the key for students to grasp what a college is like accurately. The opinions of a few students cannot accurately represent the entire population, creating a flawed representation of a college. A sufficiently-sized sample would better show where a college excels and fails in certain aspects. However, most resources online collect their data by interviewing students, requiring them to post a comment, and giving the rating of their college, which would not be efficient due to the volume of data and the time and resource costs.

The first experiment was designed to test the accuracy of the sentiment analysis model [3]. The experiment was set up so the model would predict fifteen comments of each type of sentiment, and its accuracies would be analyzed to verify whether it can precisely analyze each type of sentiment. We found out from the experiment results that the model perfectly predicted positive and negative comments but failed to predict neutral comments accurately. We analyzed that this stems from the imbalance of data for each type of sentiment – there are significantly more positive and negative comments than neutral ones. The second experiment was designed to test the user experience of the software frontend design: whether the app is user-friendly enough and whether experiment participants can provide beneficial feedback for the program. For this experimental setup, we asked ten participants to try the app and provide feedback regarding their experience. As a result, the users consistently highlighted three possible improvements: more information for each college, a search feature for the reviews, and more filters for the recommendation system on the main page. These comments signal that the UI design was not convenient enough for the users and later work is needed to remedy this problem.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. The Difficulty of Data Collection

The difficulty of data collection was the first challenge I confronted as I was initially developing my app [4]. Finding useful data sources that satisfy my project's goal took work [5]. For my app, the comments I seek to gather are students' reflections and evaluations of the college they have gone to. Many of my early attempts to gather data failed due to irrelevant information. For instance, a flaw occurred as I attempted to gather data from social media, including Twitter and Reddit [6]. At first, I saw social media as an extensive database of valuable student opinions because millions of conversations are going on every hour on social media platforms. However, most comments there could have been more beneficial for my project. Students' comments are mostly unrelated to their college's positive and negative sides. Instead, these comments are filled with memes, gossip, and complaints, which are liked and commented on the most. Consequently, another route for my data collection – web scraping – met difficulties. The websites protect some

online data resources: their HTML code would be intentionally altered by the website owners or is set to be uninterpretable by web scraping methods [7][8]. Specific lines of web scraping code could circumvent this issue occasionally; however, an optimal solution for me is to look for alternative sites with available information.

2.2. My App's Operation on A Large Quantity of Data

Moreover, another potential obstacle I need to address is my app's operation on extensive data. There are thousands of colleges and universities in the United States and potentially millions of gatherable reviews on the internet. Storing a majority of them would undoubtedly meet issues: a sizable amount of data would slow the running speed of my app and would not be supported by much app-building software. Upon resolving this problem, the app developer must first support this quantity of data; others, such as Thunkable, would not function for my project [10]. An app developer well suited for storing extensive data is FlutterFlow, which also has optimal functions for filtering comments and colleges. In addition, I could implement ways to minimize the quantity of the data, including splitting the colleges based on the first letters of their names or their states.

2.3. The Artificial Intelligence Model

Consequently, the app's functionality is strongly impacted by the artificial intelligence model I am using. The accuracy of the model is crucial to the effectiveness of the app. As the model needs to analyze the sentiments of the comments – whether they are positive, negative, or neutral – it undoubtedly needs training on various data to ensure its ability of accurate predictions. The sentiment analysis model relies on the words – usually related to particular sentiments – that they were previously trained on to predict the sentimentality of the whole comment. Therefore, an adequate amount of manually-labeled data is required for training the model. After sufficient data has been collected, I will use a minimum of one thousand manually labeled comments to train the data. I would use the model to predict another one-thousand comments and verify its accuracy, securing its precision on further comments my app used and presented.

3. SOLUTION

The main structure of my program could be evaluated in three components: a web scraping backend, analytics system, and software developing frontend. All the data the app uses is stored within a database that would be processed by a sentiment analysis AI model [9]. For this program, I used Jupyter Notebook and multiple Python libraries: BeautifulSoup, Pandas, and Json for web scraping of data; Google Colab and other analysis-related Python libraries: sklearn and nltk to operate my analytic model; and Flutterflow for frontend user interactions with the program. With the failed attempts to develop my app using Thunkable, I eventually settled on FlutterFlow due to its convenience for designing components and extensive capacity to accumulate data. In addition, I decided to use Python to web scrape and program my sentiment analysis AI because numerous existing Python libraries provide a more straightforward and more advantageous way to accomplish these goals. My program design is based on convenient access to college students' reflections on their college. The program holds an immense amount of data, collected through web scraping and APIs, which is stored in a database that would eventually contain all the reviews, college names, tuition numbers, standardized test score ranges, and other characteristics of a college. All the reviews in the database would be labeled with their sentimentality utilizing a trained artificial intelligence analytic model, a key app feature. This feature would demonstrate the distribution of sentiments for all the comments to a college, allowing users to understand the ratio of students' sentimentality toward their college.

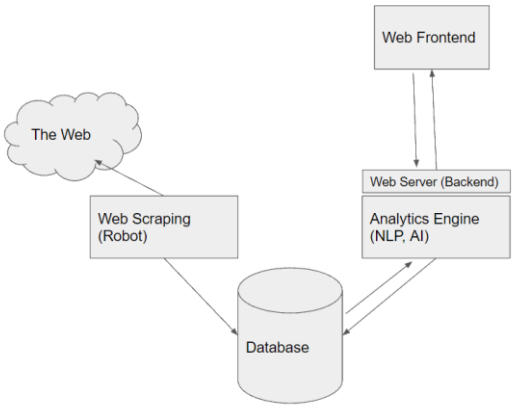


Figure 1. Overview of the solution

An essential part of my program is the web scraping backend. It relies on programming and APIs to gather an immense collection of students' reviews, their colleges, and other useful characteristics of colleges, including tuition and standardized testing score ranges. My program relies on this data to demonstrate the users' aspects of colleges through direct opinions of previous students, which is my primary purpose for designing this app.

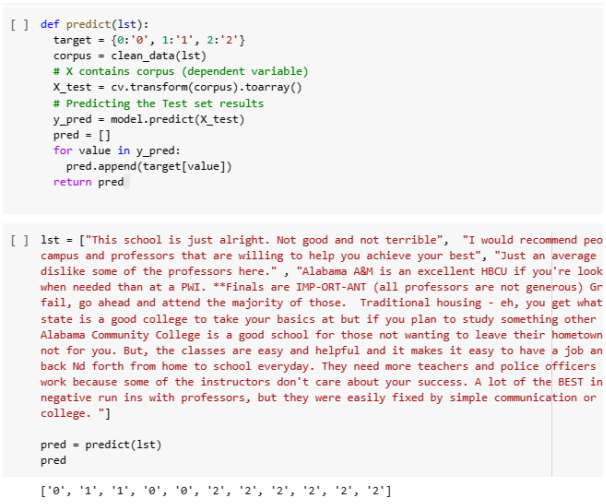
```
In [6]: print(len(specific_college_urls))
reviews = []
stars = []
college_names = []
sat_lower_end = []
sat_higher_end = []
act_lower_end = []
act_higher_end = []
acceptance_rates = []
tuitions = []
sat_ranges = []
act_ranges = []
for i in specific_college_urls:
    counter = 0
    page = requests.get(i)
    soup = BeautifulSoup(page.text, "html.parser")

    res = soup.findAll("div", class_= "overall-college-user-review-container review_div")

    for i in res:
        if counter < 30:
            var = i.findAll("i", class_= "fa fa-star")
            stars.append(len(var))

            if(i.find('p').text):
                reviews.append(i.find('p').text)
                #print(counter)
                counter += 1

            college_names_res = soup.find("div", class_= "college-header-banner-content")
            college_names.append(college_names_res.find("h1").text.strip())
            table = soup.find("table", class_= "table table-bordered")
            acceptance_rates.append(table.findAll("tr")[0].text)
            #print(acceptance_rates)
            sat_ranges.append(table.findAll("tr")[3].text)
            act_ranges.append(table.findAll("tr")[4].text)
            tuition_res = soup.find("div", class_ ="costs-container")
            print(college_names_res.find("h1").text.strip())
            tuitions.append(tuition_res.findAll('strong')[0].text)


print(college_names)
```

Figure 2. Screenshot of code 1

This screenshot illustrates the process of web scraping data from different colleges at one site, and this code runs in my program after storing each college's URL into an array. It collects all the necessary data from a specific college's page and stores them in different arrays, which would later be organized in separate columns in a CSV file. Multiple variables represent the fundamentality of my program: reviews array and college_name array — the two fundamental variables — and others like tuitions and acceptance_rates would be helpful for the recommendation system — which, based on the users' input of their characteristics, would provide them with recommended colleges in the main page — that would be implemented later in my program. Overviewing the process, this code first stores the URLs of all colleges and then webscrapes these pages by emphasizing the section that holds the vital components, including student reviews, SAT/ACT ranges, acceptance rate, and tuition [11]. Consequently, the array variables that were initialized would be filled with information from the different colleges, and later these arrays would be organized in a CSV file that would be uploaded to firebase.

Moreover, another fundamental part of my app is the analytics system. This system is requisite for the program's performance because it uses machine learning to perform the analysis action to

predict the sentiment of each comment. This analytic model is based upon Python-imported libraries like Natural Language Toolkit and Scikit Learn, which compute the human language into data and recognize human sentiments from the provided data [15].



```
[ ] def predict(lst):
      target = {0:'0', 1:'1', 2:'2'}
      corpus = clean_data(lst)
      # X contains corpus (dependent variable)
      X_test = cv.transform(corpus).toarray()
      # Predicting the Test set results
      y_pred = model.predict(X_test)
      pred = []
      for value in y_pred:
        pred.append(target[value])
      return pred
```

```
[ ] lst = ["This school is just alright. Not good and not terrible",  "I would recommend peo
    campus and professors that are willing to help you achieve your best", "Just an average
    dislike some of the professors here." , "Alabama A&M is an excellent HBCU if you're look
    when needed than at a PWI. **Finals are IMP-ORT-ANT (all professors are not generous) Gr
    fail, go ahead and attend the majority of those.  Traditional housing - eh, you get what
    state is a good college to take your basics at but if you plan to study something other
    Alabama Community College is a good school for those not wanting to leave their hometown
    not for you. But, the classes are easy and helpful and it makes it easy to have a job an
    back Nd forth from home to school everyday. They need more teachers and police officers
    work because some of the instructors don't care about your success. A lot of the BEST in
    negative run ins with professors, but they were easily fixed by simple communication or
    college. "]

    pred = predict(lst)
    pred

    ['0', '1', '1', '0', '0', '2', '2', '2', '2', '2', '2']
```

Figure 3. Screenshot of code 2

The image above contains the sentiment analysis model's prediction method and demonstrates its use. This code runs at the end of the analytics model portion – after the data has been resampled due to the imbalance between each type of comment and after the data has been split into training data and testing data for the model to be trained. The "predict" function initially cleans the data of the parameter "lst" by removing punctuations and lowering cases, and it sets the cleaned data to the corpus variable. Afterward, the model is applied to the data and predicts the sentiment of the data based on the words contained in each phrase. The sentiment of each data is then added into the "pred" array in the order of the input "lst," and the function returns the predicted-sentiment array. The cell below the predict demonstrates how the sentiment analysis process is performed through code: an array of sentences is provided and inputted as a parameter of the "predict" function. The predict function then would return an array of the predicted sentiments of each comment.

Consequently, the front-end would serve as the program's last major component. This component consists of designing a user interface that directly influences users' experience, and different features are implemented for users' convenience. The frontend component was initially implemented using Thunkable — which was not optimal for my program — before it was switched to FlutterFlow. As the program is designed for student usage, the software aspect of the project would be crucial since we want to offer a user-friendly and attractive application for them to use.

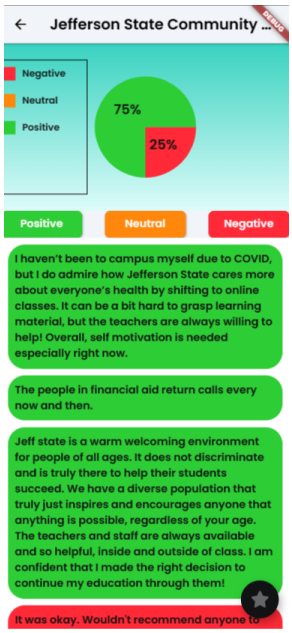


Figure 4. The reviews page of a specific college

The screenshot shows a specific college's reviews page, demonstrating various components designed for users' convenience. The pie chart at the top informs the reader of the proportion of positive, negative, and neutral reviews for the college. The positive, negative, and neutral buttons – denoted by a specific color – filter the comments by sentiment. Specifically, highlighting a type of sentiment means that only comments of that sentiment, whether positive, negative, or neutral, can be viewed. For instance, if only the green "Positive" button is highlighted, then only positive comments are filtered for users to perceive. Each comment has a background color based on its sentiments, which helps users quickly identify it. Furthermore, the star button on the bottom right corner stores a college to the "favorites page," where students can go through a shorter list of colleges more reflective of their interests. Consequently, it illustrates how the program is developed for users' convenience.

4. EXPERIMENT

4.1. Model Accuracy Experiment

An essential experiment I would conduct is testing the accuracy of the sentiment analysis model, which is vital to my program because the model's accuracy determines whether my app achieves its goal of illustrating a correct representation of the student population's sentiment toward a specific college.

We must test comments with diverse diction for the experiment to reflect whether the model is accurate. With this in mind, we came up with fifteen positive, negative, and neutral comments that would be used to test the model's accuracy. This ensures that all three possible outcomes – positive, negative, and neutral – are tested to see if the model can precisely predict them. After computing the comments into the model, we would compare the predicted sentiment with manually-labeled sentiment to examine the reliability of the analysis through its accuracy in predicting the correct sentiment.

Comment	Actual Sentiment	Predicted Sentiment
1	Neutral	Positive
2	Neutral	Neutral
3	Neutral	Negative
4	Neutral	Neutral
5	Neutral	Positive
6	Neutral	Positive
7	Neutral	Neutral
8	Neutral	Neutral
9	Neutral	Neutral
10	Neutral	Negative
11	Neutral	Neutral
12	Neutral	Positive
13	Neutral	Positive
14	Neutral	Neutral
15	Neutral	Neutral

Comment	Actual Sentiment	Predicted Sentiment
1	Positive	Positive
2	Positive	Positive
3	Positive	Positive
4	Positive	Positive
5	Positive	Positive
6	Positive	Positive
7	Positive	Positive
8	Positive	Positive
9	Positive	Positive
10	Positive	Positive
11	Positive	Positive
12	Positive	Positive
13	Positive	Positive
14	Positive	Positive
15	Positive	Positive

Comment	Actual Sentiment	Predicted Sentiment
1	Negative	Negative
2	Negative	Negative
3	Negative	Negative
4	Negative	Negative
5	Negative	Negative
6	Negative	Negative
7	Negative	Negative
8	Negative	Negative
9	Negative	Negative
10	Negative	Negative
11	Negative	Negative
12	Negative	Negative
13	Negative	Negative
14	Negative	Negative
15	Negative	Negative

Figure 5. Table of experiment 1

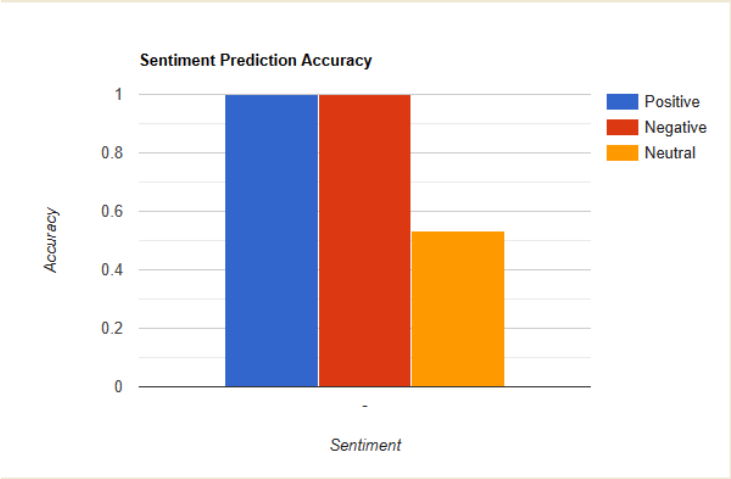


Figure 6. Sentiment Prediction Accuracy

From this experiment, the overall prediction accuracy is 84%, with a positive prediction accuracy of 100%, negative prediction accuracy of 100%, and neutral prediction accuracy of 53%. This statistic demonstrates that my model can accurately predict positive and negative comments but is ineffective for neutral ones. The model's accuracy in predicting neutral comments did not meet my expectations, as its accuracy drastically differs between positive and negative comments. Reflecting on this data, I believe it turned out this way because there was a lack of neutral sentiment comments from the training data and the internet compared to the positive and negative ones. This issue leads to the model's incapability to accurately predict neutral comments; the model is more likely to predict a comment as positive or negative than neutral. Therefore, this explains how the model can accurately predict positive and negative sentiment comments while predicting neutral comments inaccurately.

4.2. User Experience Experiment

Another potential blind spot that needs to be tested and fixed to improve the program is user experience. Although several components are integrated for the app to be more convenient to use — for example, recently searched colleges, favorite college page, and college recommendation system on the main page — different users could have different suggestions on which the app could be improved for convenience. This aspect is crucial because it shows the professionality of the program and attracts the users.

The experiment would allow users to test the app and offer feedback on potential improvements. Ten users would try out the app and navigate through all its features, and they would reflect and fill out a feedback form on ideas for new components that could make the program more convenient to use. Consequently, the feedback would be analyzed, and the program would be adjusted accordingly based on this feedback. In this way, we would acquire the indicative reflection from those who had the user experience on the program; therefore, this potential blind spot could be detected and addressed optimally.

User	Feedback (paraphrase)
User 1	For users' convenience, the program could implement a searching feature that could search and highlight words from specific comments, and the users could skip through comments and go to ones with these specific words.
User 2	For the filtering feature on the main page, the app should have more components that could be filtered; besides SAT score and tuition, other ones could be GPA, acceptance rate, state of the college/university, private schools only, etc.
User 3	On the college page, it could show more components of a specific college, such as the ranking of the college, its exceeding majors, and students enrolled each year.
User 4	The number of reviews for each college could be more for users to reflect on. Besides that, a searching tool for keywords in the comments would be beneficial.
User 5	I noticed that the filtering feature for the recommendation system on the main page only has "SAT score and tuition" on there. For the app to be more user-friendly, I believe adding more components to the filtering aspect would be great.
User 6	My suggestion for this program is that more features could be added; right now the program is limited to reviews of colleges, and I believe it would be better for the users to see more information about each college.
User 7	Going through the app, a feature that I thought would be helpful is another filtering feature in the reviews section for each college. I noticed that the app has a filtering feature based on sentiment, which is very helpful, but it could also filter something else such as food-related comments, campus-related comments, or housing-related comments.
User 8	Something that could enhance my experience of using this app is more reviews for each college. In this way users can understand more of each aspect of a college.
User 9	Everything seems fine with this app; I believe it could be especially useful for high school students who are about to make the college-decision. One thing that could improve the app is more filters for the recommending colleges component on the main page.
User 10	In order for the app to be more user-friendly, more useful descriptions of each college should be provided, such as its 75 percentile SAT score, its average GPA of accepted students, and its best majors.

Figure 10. Table of experiment 2

After analyzing users' feedback, three significant improvements became clear: including more reviews and information for each college, adding a searching or filtering feature on the college reviews, and making more available filters for the recommendation system. These reflections are beneficial as they demonstrate areas of the project that require enhancement. In the first version of this app, we expected there to be insufficient aspects. Indeed, the next step of this project is to resolve the issues above and refine the system to be additionally advantageous for users. More information would be gathered from the internet – including reviews and other vital details of a college. A search feature would be implemented to assist users in finding what meets their intentions. Lastly, the recommendation system would include more practical filters such as GPA, acceptance rate, and location.

5. RELATED WORK

The College Finder & Recommender Web Application System research paper provided a solution involving a web-based application that allows users to efficiently find universities that they can apply to and match users' demands [14]. Improving upon the existing solution, which is searching each independent college manually, this paper provides an effective solution of utilizing a filtering system, recommendation system, and a wish list. The filtering system helps users find colleges that match the required properties. The recommendation system helps users by recommending colleges that match the input of users' profiles. The wish list provided easier and quicker access to a shorter list of desired colleges. Overall, the system is effective and influential as it provides increased efficiency in searching for suitable colleges; however, there are crucial aspects that the system ignores. For instance, the app focuses on academic suitability for a student and ignores most of the other aspects of a college. My app improves on what they tried as it provides students' reviews on colleges that reflect other perspectives of colleges. With this improvement, the app's users could better visualize their lives in the college they are looking for and would further assist high school students in making the college decision.

6. CONCLUSIONS

The College Finder & Recommender Web Application System research paper provided a solution involving a web-based application that allows users to efficiently find universities to apply to and match users' demands [14]. Improving upon the existing solution, which is searching each independent college manually, this paper provides an effective solution of utilizing a filtering system, recommendation system, and a wish list. The filtering system helps users find colleges that match the required properties. The recommendation system helps users by recommending colleges that match the input of users' profiles. The wish list provided easier and quicker access to a shorter list of desired colleges. Overall, the system is effective and influential as it provides increased efficiency in searching for suitable colleges; however, there are crucial aspects that the system ignores. For instance, the app focuses on academic suitability for a student and ignores most of the other aspects of a college. My app improves on what they tried as it provides students' reviews on colleges that reflect other perspectives of colleges. With this improvement, the app's users could better visualize the college life they are looking for and would further assist high school students in making college decisions.

Building on the intention of informing students about colleges alongside their academic capabilities, various features, and ideas are implemented to help students understand the different perspectives of colleges. As technology advances and researchers make innovations, students later on will more easily find the colleges that suit them and will not regret the years of life they spent there.

REFERENCES

[1] Brint, Steven, and Charles T. Clotfelter. "US higher education effectiveness." RSF: The Russell Sage Foundation Journal of the Social Sciences 2.1 (2016): 2-37.

[2] Samuelowicz, Katherine, and John D. Bain. "Revisiting academics' beliefs about teaching and learning." Higher education 41 (2001): 299-325.

[3] Ullah, Mohammad Aman, et al. "An algorithm and method for sentiment analysis using the text and emoticon." ICT Express 6.4 (2020): 357-360.

[4] Joorabchi, Mona Erfani, Ali Mesbah, and Philippe Kruchten. "Real challenges in mobile app development." 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE, 2013.

[5] Ohno-Machado, Lucila, et al. "Finding useful data across multiple biomedical data repositories using DataMed." Nature genetics 49.6 (2017): 816-819.
[6] Anderson, Katie Elson. "Ask me anything: what is Reddit?." Library Hi Tech News 32.5 (2015): 8-11.
[7] Samant, Nishank, et al. "College Finder & Recommender Web Application System." (2017).techniques." 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT). IEEE, 2019.
[8] Khder, Moaiad Ahmad. "Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application." International Journal of Advances in Soft Computing & Its Applications 13.3 (2021).
[9] Birnbaum, Lawrence, Margot Flowers, and Rod McGuire. "Towards an AI model of argumentation." Proceedings of the First AAAI Conference on Artificial Intelligence. 1980.
[10] Lim, Soo Ling, and Peter J. Bentley. "How to be a successful app developer: Lessons from the simulation of an app ecosystem." Acm Sigevolution 6.1 (2012): 2-15.
[11] Coyle, Thomas R., and David R. Pillow. "SAT and ACT predict college GPA after removing g." Intelligence 36.6 (2008): 719-729.
[12] Emeakaroha, Vincent C., et al. "A cloud-based iot data gathering and processing platform." 2015 3rd International Conference on Future Internet of Things and Cloud. IEEE, 2015.
[13] Kaplan, Andreas M., and Michael Haenlein. "Users of the world, unite! The challenges and opportunities of Social Media." Business horizons 53.1 (2010): 59-68.
[14] Samant, Nishank, et al. "College Finder & Recommender Web Application System." (2017).
[15] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." The Journal of machine Learning research 12 (2011): 2825-2830.