Indeed Resume Analysis

BIA-660-B Team 16

Manank Valand
Mehul Mistry
Ayushi Chaturvedi
Shefali Chhabria

Acknowledgement: We would like to thank the our Professor R Liu and TA for their participation and continuous help in this project who supported our work in this way and helped us get the results of better quality. We are also grateful to all our fellow students of class BIA-660.

1. Preface

As we know, getting jobs and finding right candidates are getting more and more complicated now a days for both candidates and companies. In order to increase the chances of getting more number of interviews, a student has to apply in a significant amount of job openings. Which includes both relevant and irrelevant job openings. According to the Linkedin Survey - 2017, chances of getting an interview call is on an average 1 out 200 applications in USA. This consumes a lot of time of students and affects their education as well. In order to solve this problem, companies came up with the platform/portal where one can apply to multiple openings at once. But that still could not help much. Our project focuses on finding an accurate and effective method for both applicants and companies which can analyze the resumes across the humongous dataset scraped from indeed.com and predict the chances/probability of being a data scientist, software developer or vice president. This will help saving a significant amount of time for both students as well as companies.

Keywords:
Text mining, Resume Analysis, Jobs, CNN, Naive Bayes, Classification.

2. Introduction

In the world of text mining and web-analytics, and that too if specifically in the field such as resume analytics, we barely get any structured data by scraping. One of the main challenges here is to labelize the data in order to apply the effective methods such as Classification and Convolutional Neural Networks. Nevertheless, due to the decent variety of normal dialect, it's hard to analyze the quality specifically by the survey substance. Rather, the basic practice in web based resume/text analysis enterprises is to rank by the survey the audit accommodation.

3. Objective

Nowadays due to increased amount of jobs in the field of Data science, Software Developer and computer related fields, we need some tool to analyze our resume which correlates our profile with the profile of people who are actually in that field. The output of the analysis will tell us that the field which we want to be in required what kind of skills and technologies, and what our profile is up to.

Many companies nowadays use ATL tool (Application Tracking Tool). That tool fetches the data from the resume we upload during our application form and matches the skills and experiences that company required. If it completely matches with the applicants resume or nearly that, then they will call them for further interview or application process.

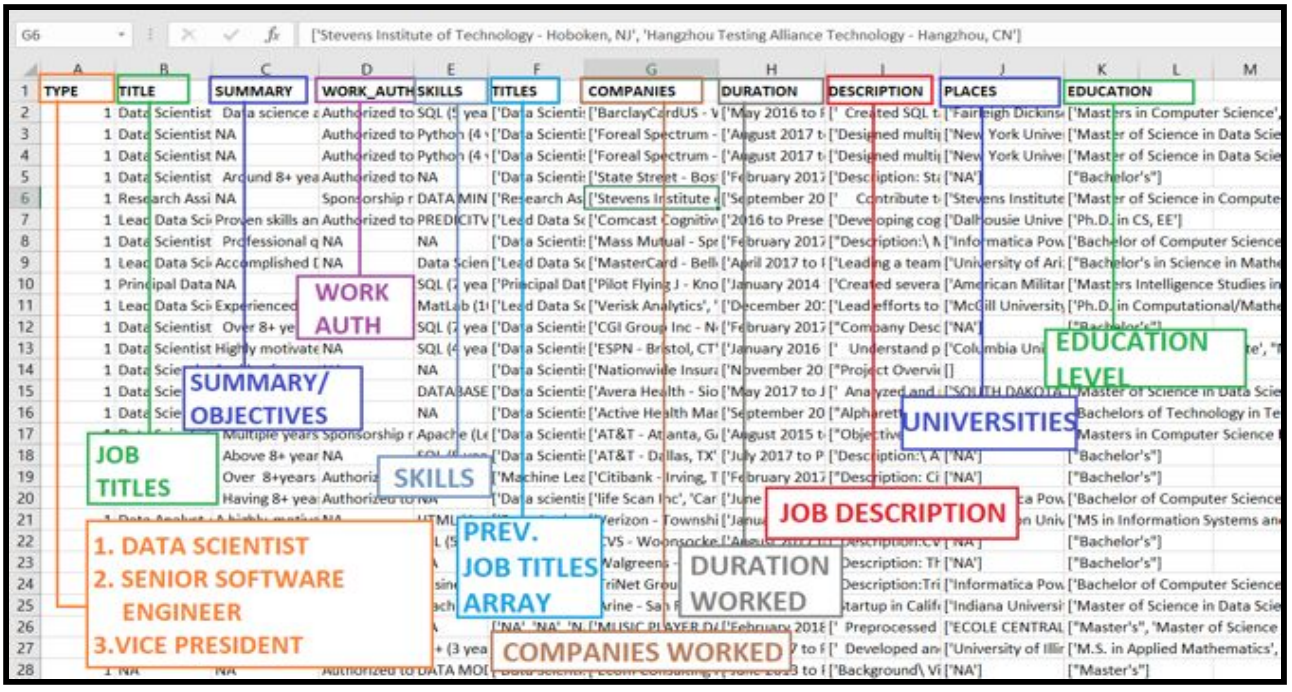
The purpose of Resume Analysis tool is to make aware the applicants about the latest skills in the market the company is currently looking for, what type of qualities ideal candidates should possess? average work experiences, type of degree and many more in the particular field which he/she is looking. With this information it helps individuals to stand apart from the other candidates.

4. Dataset

We have scraped almost 3000 resumes from indeed.com for 3 major job titles.

- Data Analyst
- Senior Software Engineer
- Vice President

A snapshot from the whole dataset. Where each column indicates the list of all similar type of information in the resume. First column is the manually given label for each different type of resume.



5. Scraping

```
res_page = requests.get("https://www.indeed.com" + rows[j])
if res_page.status_code == 200:
res_soup = BeautifulSoup(res_page.content, 'html.parser')
resume = []
headline = None
summary = None
eligibility = None
skills = None

# get headine of resume
res_headline = res_soup.select("h2#headline")
if res_headline != []:
    headline = res_headline[0].get_text()
    headline_list.append(headline)
else:
    headline_list.append("NA")
```

We have used the leading tool and HTML Parser in web scraping named Beautifulsoup 4.0 to scrape resumes. Although we had to manipulate the input data by manually giving the labels to each input row/ resume, It came very useful while collecting the data. Unlike Selenium, this tool automatically did the crawling in the backend which saved a lot of time and processing energy.

6. Dataset Analysis

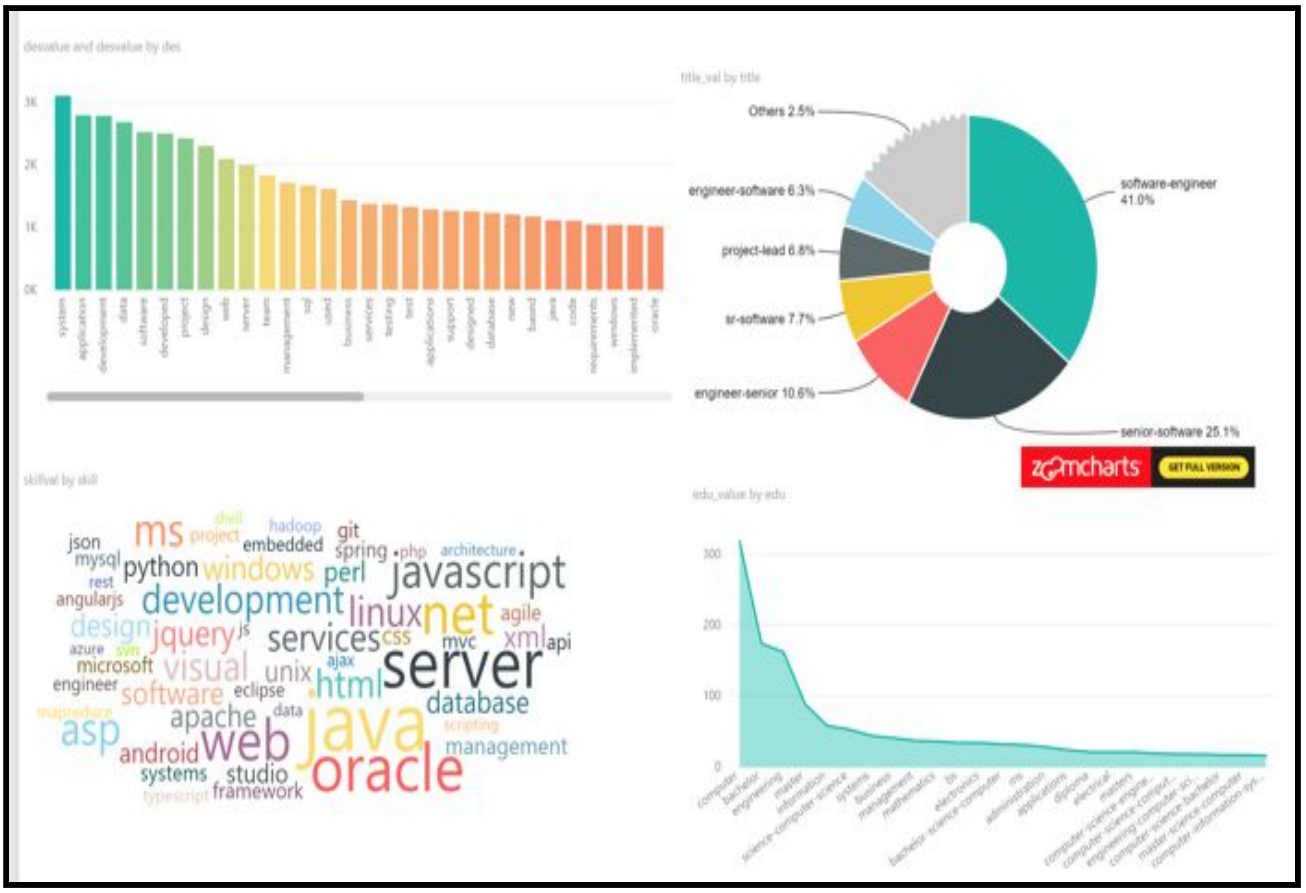
We applied the preprocessing on all of the fields in order to clean the data. Which includes

- dates to numbers in i.e.(January 2017 to January 2018 will be converted to 1).
- Removal of Non ASCII characters and stop words.
- Tokenization using NLTK corpus, collocations and KERAS Preprocessing.

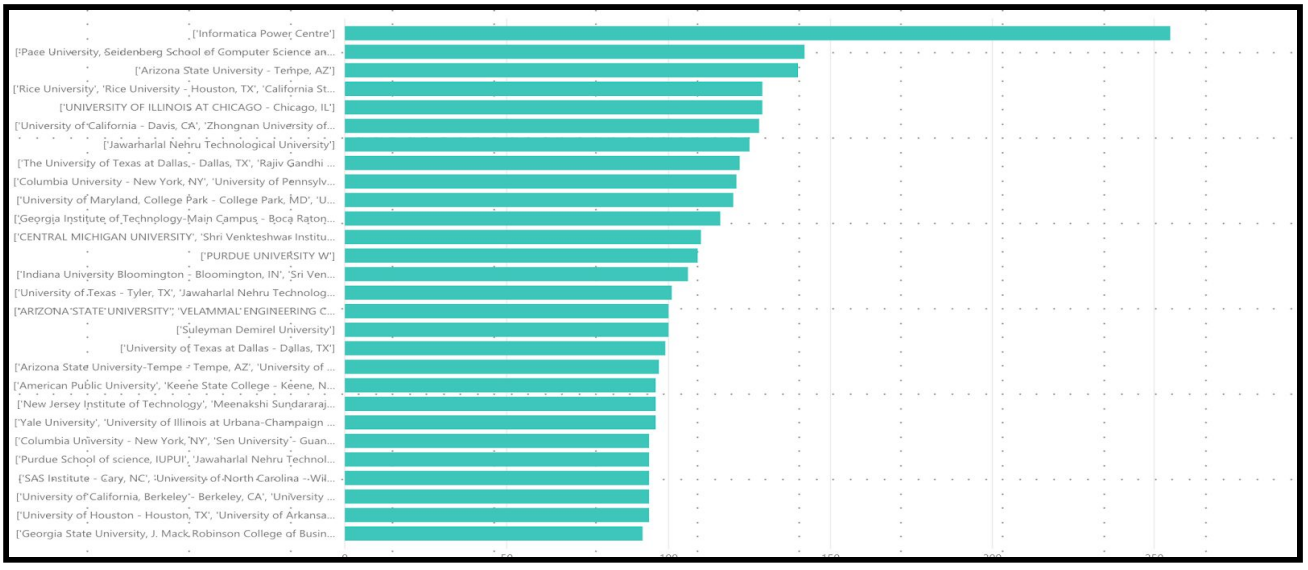
We processed these data in order to get the clean format on which analysis can be done.

These graphs are the analysis results from the relevant and meaningful tokens from the fields like 1.Job Description, 2. Job Titles, 3. skills and 4. Education Level columns of the dataset of Senior Software Engineer.

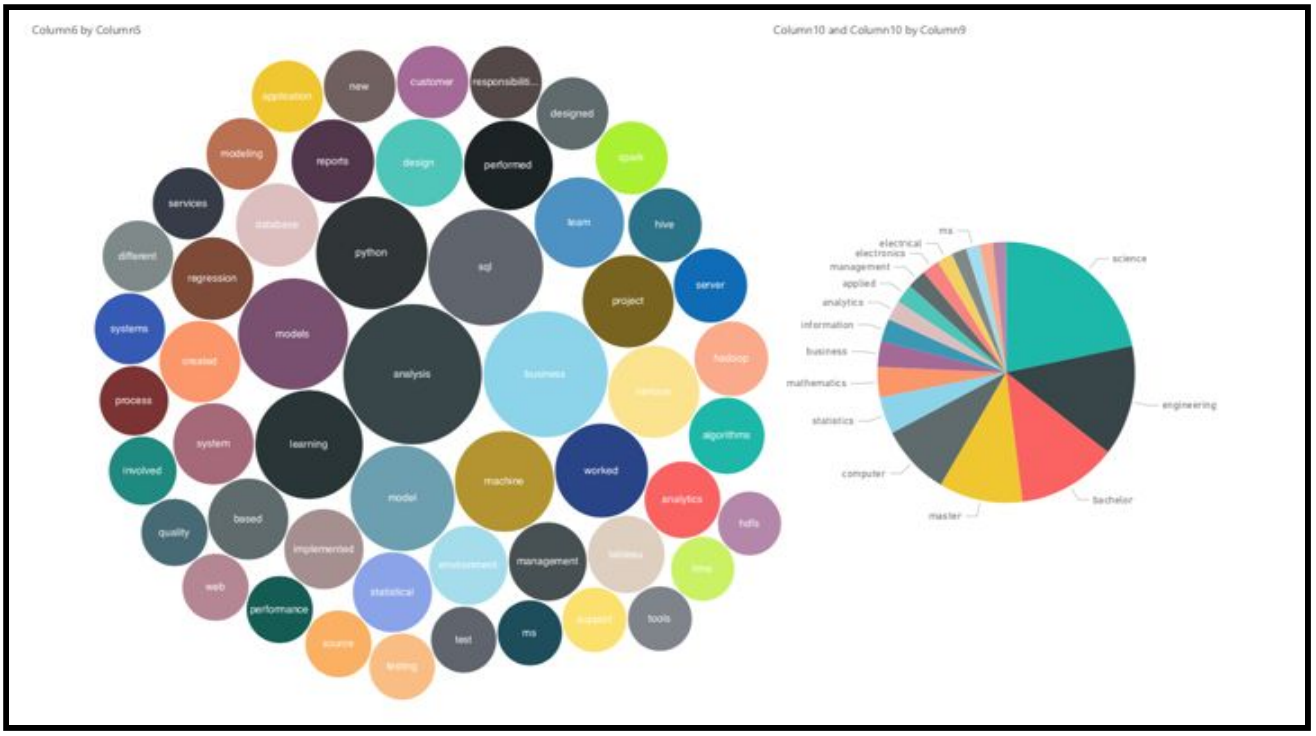
- As we can say the most frequent words from 1. Job Description graph are “system”, “application”, “software”, “Project”, “Developed” etc.
- Top frequent words from the analysis on 2. Job Titles column are “software engineer” and “Senior Software Engineer”. Top skills have the words such as “Java”, “Oracle”, “Server” etc.
- And Education details have words such as Computer Engineering, Bachelor’s, System engineering and so on.



Likewise the result below is the analysis on meaningful tokens from the education and previous universities. By looking at the same we can say that, if you want to be a data scientist, these are a few decent universities to start with - which includes, Pace university, Arizona State University, UC Davis etc.



These 2 graphs represent the Skills and Education level columns' meaningful tokens of the dataset of Data Scientist.



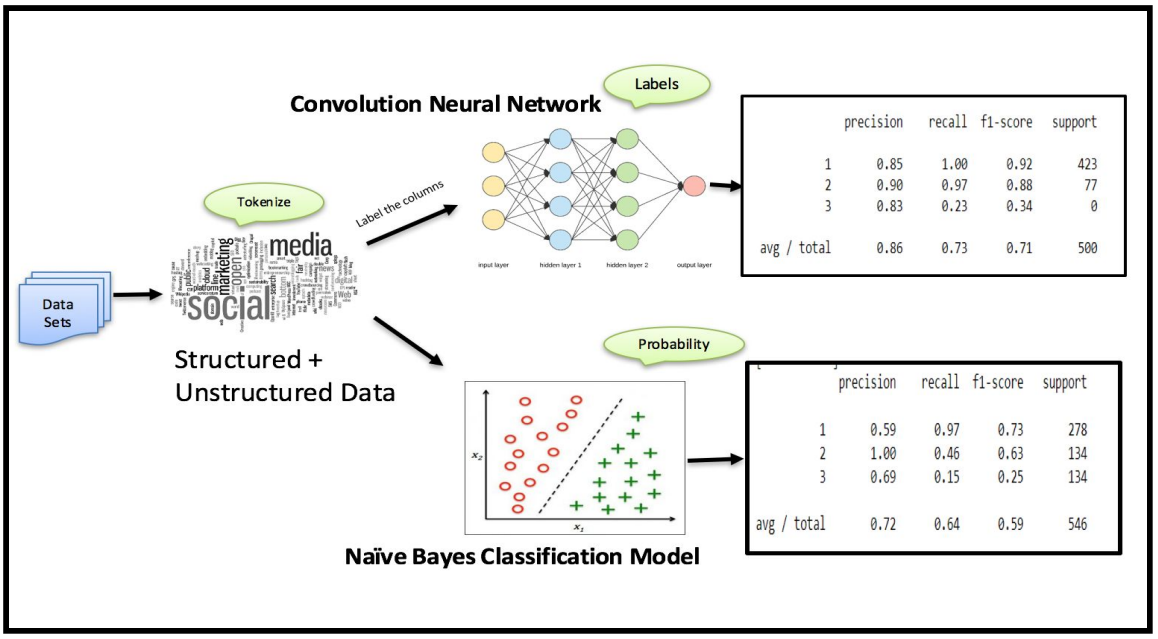
7. Methodology

We have raw data which is combination of structured and unstructured columns. It's challenging task to convert the unstructured data to structured data. Before applying any model, we manually did the labelling in the unstructured data.

Now as we have perfect labels for 3 different job titles, we can now apply Classification or CNN model on the dataset.

We have used two models:

1) Convolution Neural Network
2) Naive Bayes Classification Model



Flow of the project:

1. Input Data sets
2. Tokenize all the columns -> output in the form of frequency and word in ascending order
3. Labelise the description column(unstructured) for input to models
4. We tried different models for better accuracy.
5. First started with Naive Bayes classification with output as probability of three labels with 72% accuracy.
6. Second model tried was CNN with output as single label with accuracy of 86%.

Classification Model:

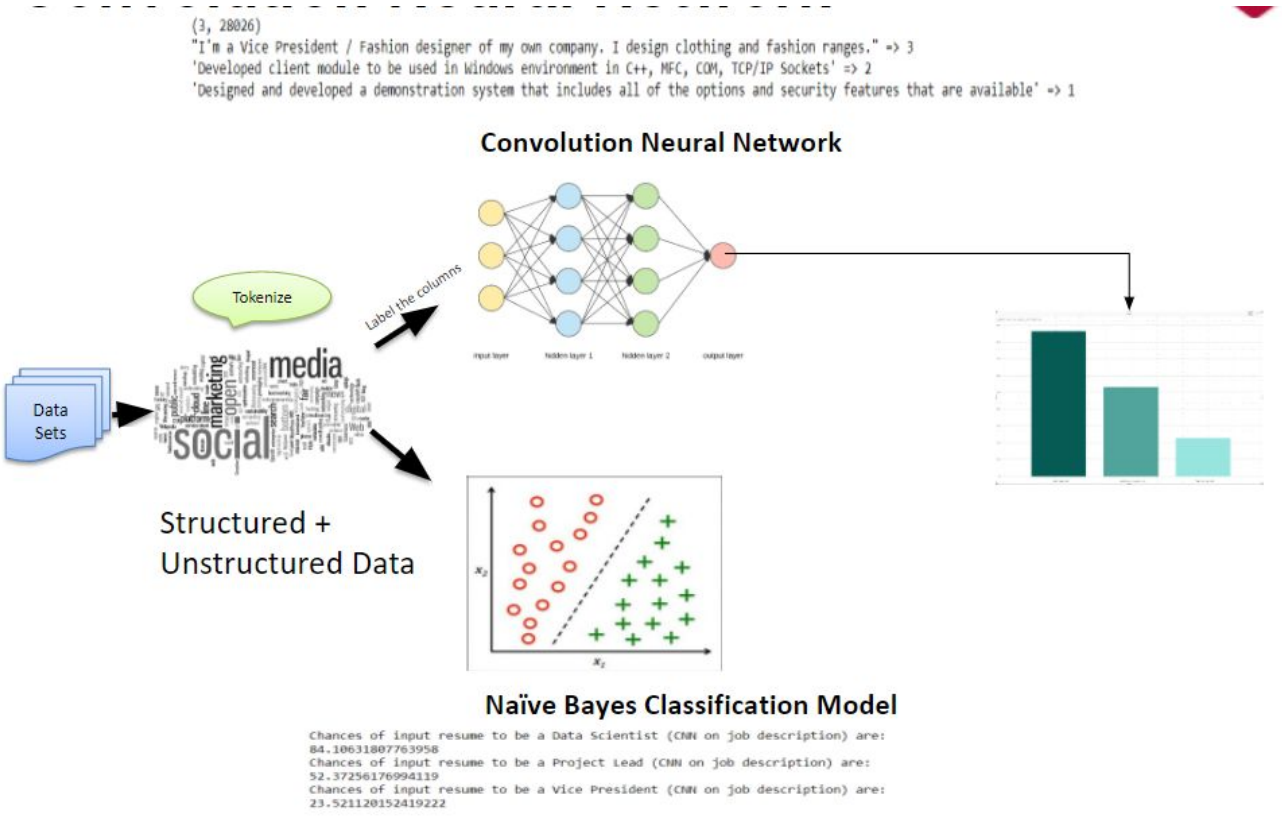
	precision	recall	f1-score	support
1	0.59	0.97	0.73	278
2	1.00	0.46	0.63	134
3	0.69	0.15	0.25	134
avg / total	0.72	0.64	0.59	546

Convolution Neural Network:

	precision	recall	f1-score	support
1	0.85	1.00	0.92	423
2	0.90	0.97	0.88	77
3	0.83	0.23	0.34	0
avg / total	0.86	0.73	0.71	500

Sometimes, ambiguous output will result into false prediction. Combination of both the models' output will help to improve the prediction results.

So after comparing and trying both with different inputs, we finalized the CNN model for the better accuracy.



8. Steps to run the project

To run the project in smooth manner, follow the below steps:

The project is tested on jupyter 5.0.0 and Pycharm and runs on Python 3. Total time for executing the project is around 10 seconds.

i. For data gathering of different profiles, just run scraper.py file in python shell. Make sure to change the link variable with desired profile. For e.g. For Vice President profile data scraping change the link variable to "?q=vice+president&co=US&cb=jt&start=50".

ii. Now, we need to clean the gathered data, so run preprocessing.py file which will clean the data which is not in 'utf-8' format.

iii. After cleaning the data, run analysis.py file for the output.

iv. Lastly, you can also run classification.py to see the classified output which uses sklearn library. Change the column number in row variable for different classification of columns.

9. Experiment Results

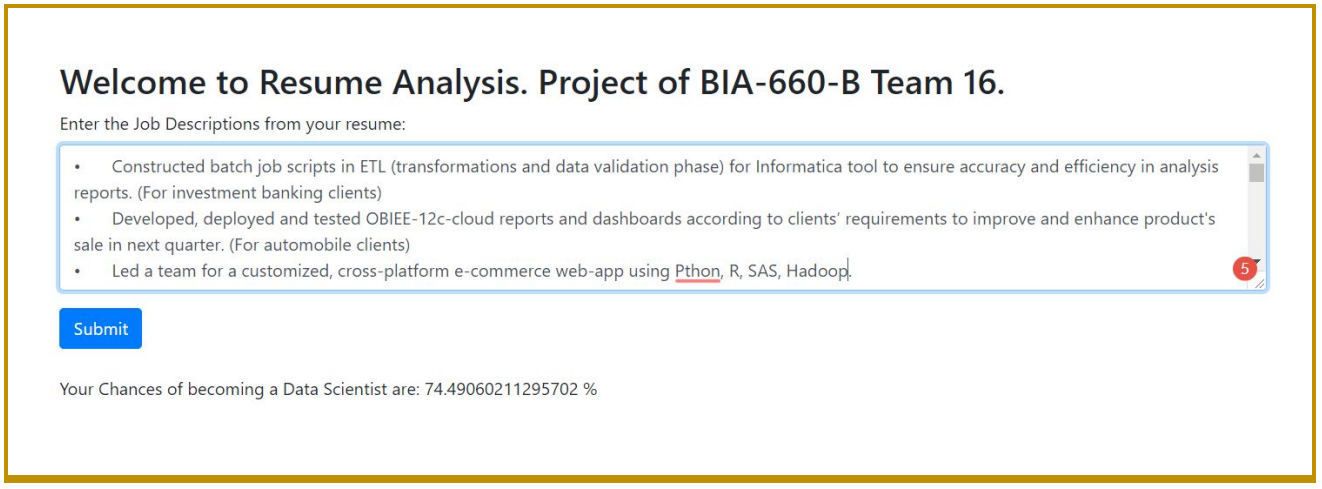
Below is the chances or probability results from summing function:

```
Chances of input resume to be a Data Scientist (CNN on job description) are:
84.10631807763958
Chances of input resume to be a Project Lead (CNN on job description) are:
52.37256176994119
Chances of input resume to be a Vice President (CNN on job description) are:
23.521120152419222
```

```
|:
```

The input job description in this result is from the resume of data scientist.

Below is the implementation and integration of our whole project into an interactive web based user friendly website. In which user needs to enter the details from his resume and our project will come up with the probability of being the data scientist.



10. Analysis of Methodologies

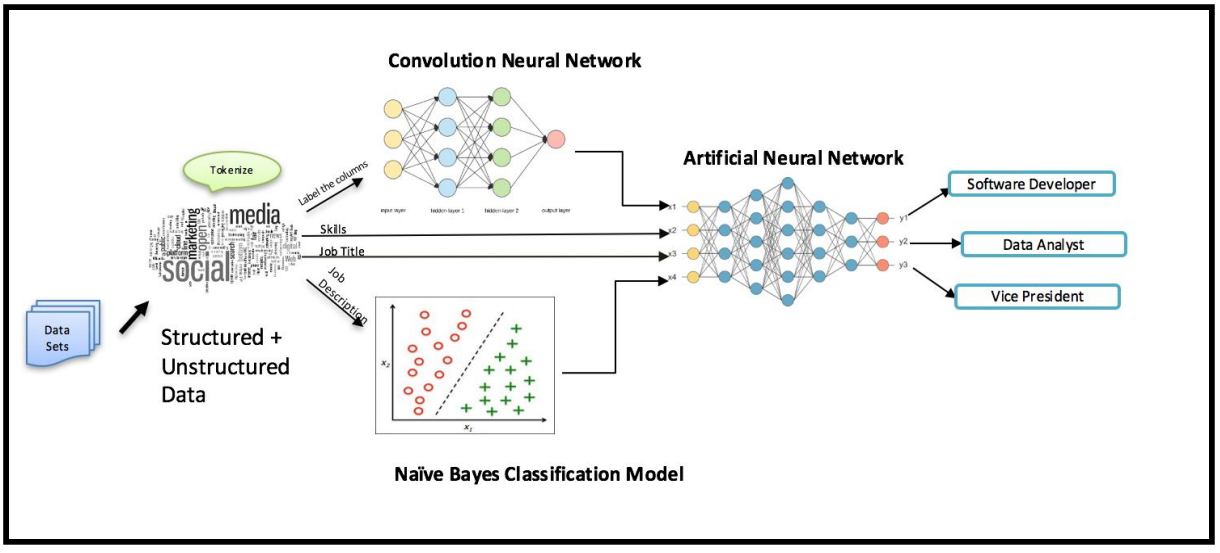
- As described earlier, after implementing and trying different methods such as svm, logistic regression, CNN and Naive Bayes Classification methods, the output and accuracy of CNN was very high as it trains itself by sending the errors and results to hidden layers.
- Before that, Classification with Naive Bayes method gave almost accurate results in percentages.
- But sometimes, ambiguous output had given a result into false prediction.So we came across the Combination of both the models' output which helped to improve the prediction results.
- So after comparing and trying both with different inputs, we finalized the CNN model for the better accuracy.
- We can still **improve** the accuracy after sending these results into the ANN model as suggested by professor Liu. Also it gives better results with unstructured data. So we are planning the combine the same with our existing model's output our future work. The flow diagram of the same is described in the very next following section.

Business Insights:

- It will help the student to analyze their resume and further help them to predict their chances of getting the desired job. It's one step further of Application Tracker tool which gives you visual representation of all the applications.
- Simply uploading of resume will help the companies to sort and analyze them according to the requirements.

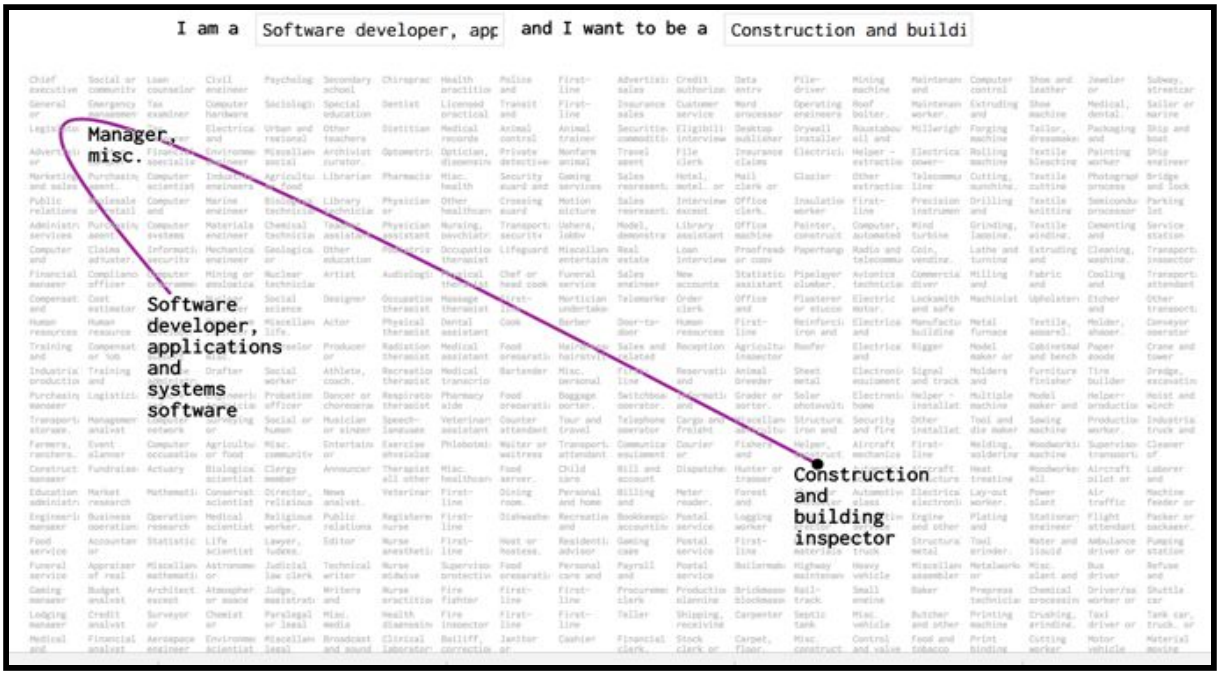
11. Conclusion and Future Work

1. Collecting more data from different profiles
2. Adding other attributes like education, previous titles and many other in prediction.



3. When the data samples get increased, the CNN training time increases as well. So, we will put it into some kind of distributed system.

4. Building career path from current and desired position.



5. Publish a python package (or a API) for third party use.

12. References

[1] Lecture Notes and Jupyter notebooks
[2] www.flowingdata.com
[3] www.indeed.com
[4] www.scikit-learn.org