PURDUE
KRANNERT
SCHOOL OF MANAGEMENT

MGMT 590- Analysing Unstructured Data

Project Report

Job Classifier & Recommender For Craigslist Website

Team Name: Vector Machinists
Team Members:

Sai Anudeep Bodaballa PUID - 0032675246
Vishnuvardhan Reddy Alla PUID - 0032852945
Anupama Sunil, PUID - 0032862145
Hasit Yarlagadda PUID - 0033009190
Mayank Jha PUID – 0032876223

PURDUE
KRANNERT
SCHOOL OF MANAGEMENT

Project Background

About Our Client:

Craigslist is an American classified advertisement website for a plethora of domains. It is a well-known global online platform connecting "providers" of jobs, housing, products, and services, among others, with "seekers" of those same offerings. Unlike other advertisement platforms, Craigslist is free for all users -both providers and seekers- with few exceptions.

It was founded by Craig Newmark, who started it as a mailing list to friends about happenings in the San Francisco Bay Area, in 1995. The website's been around since 1999, and now serves 700 cities in 70 countries, supporting 13 different languages. The platform has consistently ranked in the top 20 most visited websites in the U.S. and among the top 118 websites globally with an estimated 50bn views per month, all essentially without advertising or marketing since its inception. It has crossed the $1 Billion mark in annual revenue for the year 2018 and is currently valued at ~ $3 Billion.

Job Postings and Resume Sub-Sections:

Studies estimate more than 2 million Americans voluntarily leave their jobs every month, looking for new opportunities. Coupling these with the involuntary unemployment numbers means that there are a lot of people looking for jobs at any given point of time. The top channels people use to look for new jobs are online job portals (60%), professional social networks (56%), and word of mouth (50%). Global job portal service is a ~ $36 Billion dollar market with a projected growth to $52 Billion by the end of 2027. LinkedIn and Monster are the largest players in the US while other job portals in the country include indeed.com, simplyhired.com.

Craigslist also offers the facility of posting advertisements related to jobs on their portal, but this category is not nearly as popular as compared to websites like monster.com or indeed.com. There might be several reasons for this, but we would like to address a specific challenge that many white-collar jobs providers face using Craigslist as their go-to platform. Throughout the project we will try to resolve the disconnect between seekers of such offerings and their suited recruiters.

PURDUE
KRANNERT
SCHOOL OF MANAGEMENT

Problem Statement:

Craigslist has a lot of job listings that are classified into very broad categories. When a user navigates through the job postings, they can only see the date on which a job was posted and the title. The title is a user's free input text, thus, there is neither a format nor structure for the jobs listed. Overall, the postings are unstructured, vague, and strenuous to consume from an end-user's standpoint. Recruiters use fancy words to attract traffic, but they lack the basic information about the position for which the job is posted.

Additionally, users cannot find most of the relevant information such as employment type, compensation, the name of the organization unless he clicks on a particular link. This lack of information on the search page forces the user to click on the links only to find that the job is not suitable for him. Given many job postings, it is humanely impossible to navigate through 100s of pages. Thus, even when there are 1000s of jobs available, because of the lack of text architecture, the purpose of matching a recruiter's need to user's remains difficult.

Similarly, the resume sub-section ranks low in terms of usability and accuracy w.r.t relevant search results. There are no job categories attached to the resumes making this section painful to navigate and filter appropriate resumes. This fails the entire purpose of hiring suitable fits through craigslist website and one of the crucial factors affecting Craigslist's performance in the white-collar job market.

Our project tries to accomplish the task of bridging the gap between job seekers and recruiters on the platform. For this we will utilizing multiple NLP techniques. We aim to build a classification model that provides recommendations for job postings and uploaded resumes using topic modelling.

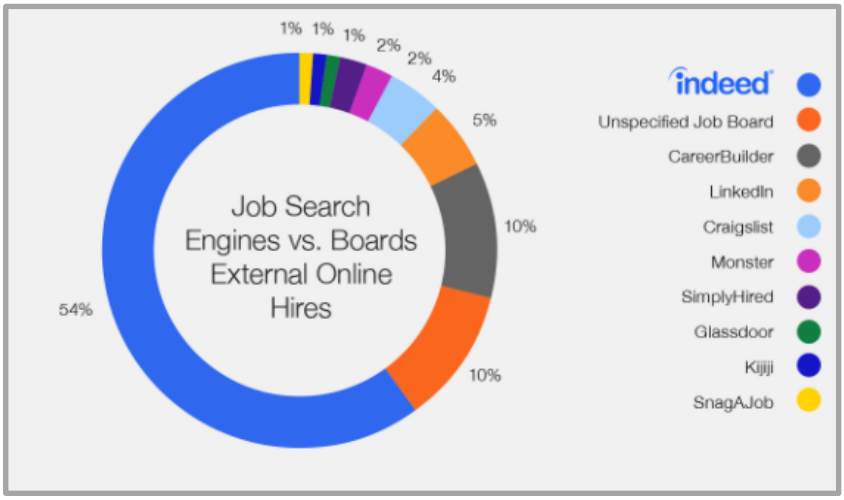

Fig. 1 : Percentage of External Online Hires

PURDUE
KRANNERT
SCHOOL OF MANAGEMENT

Business Analysis

Our project goal is to build a job classifier to provide job recommendations based on job postings and uploaded resumes using Topic modelling. This would improve the listings of the job search results by providing a well-defined structure consisting of relevant information to the job searchers and furthermore enhance the experience by providing them pertinent recommendations aiding their job search. The data driven goal is to establish Craigslist as one of the top job portals in all the job categories including white collar jobs.

Currently, the categorisation given under the job section is very vague and the postings under each category does not follow an organised structure and varies according to user. In comparison, the job postings in competitor sites like Indeed.com are more organised and provides a higher ease of use to users.

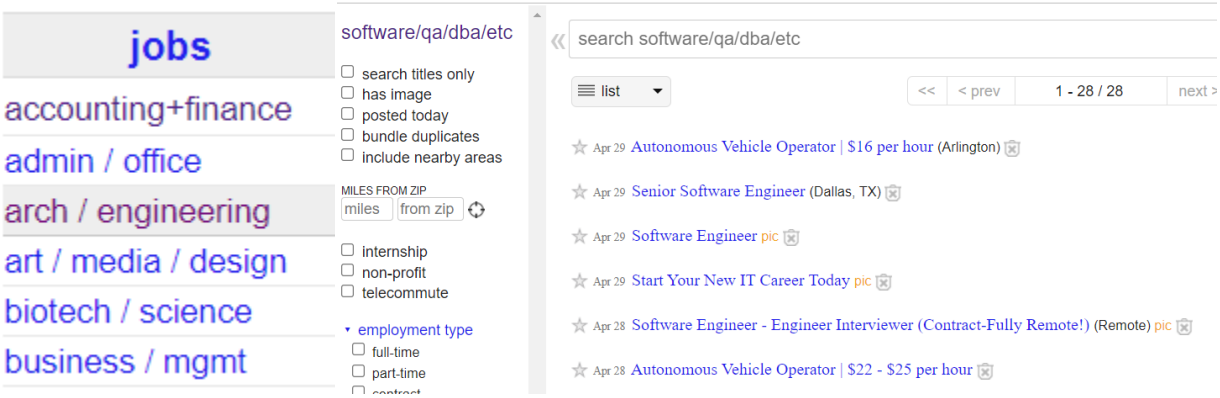

Fig. 2 : Screenshot Job Postings– Craiglist.org

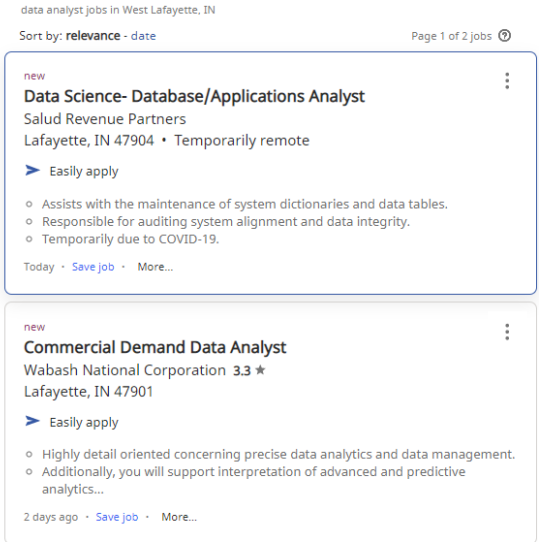

Fig.3: Screenshot Job Postings– Indeed.com

Due to this lack of proper job categorization, the usability of the job section is less especially for white collar job searchers as proper filtering methods and job recommendations are required to find matching opportunities for their skills. In Craigslist, as the job headings does not give relevant information related to the posted job, the users need to click on multiple links to identify what each job posting is about, and this leads to wastage of user time which can discourage traffic in the job section. Coming to the Resume section, there is no well-defined structure or field wise division of data. Similar to the job section, this leads to wastage of time for the job providers using the site as they need to click on multiple links to find even basic details of the Resume. These technical difficulties can drive away users from Craigslist resulting in significant loss of revenue.

We propose to implement an algorithm wherein the machine learning model takes in the job posting details and extract keywords to create a job title and this job title can be further utilised to identify top 3 related jobs which can be provided as recommendation to the user. Furthermore, from the resume section, the relevant details from each resume can be extracted to match with the job titles to recommend top 3 matching jobs for the resume.

Most of the revenue for Craigslist comes in from monetising some of the premiere sections like 'Jobs'. An increase in user traffic and job postings would directly increase the revenue for the company.

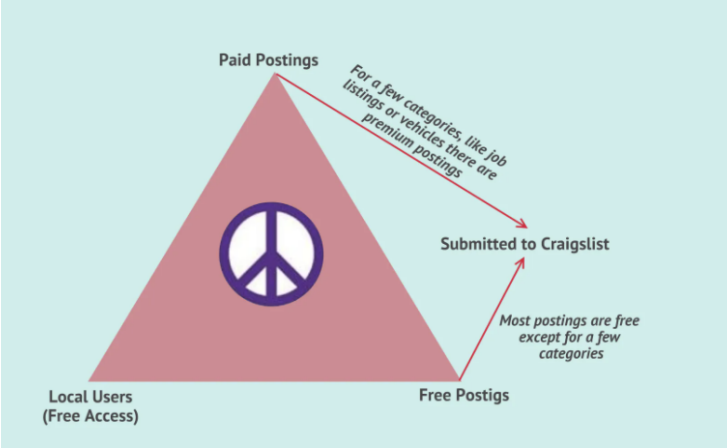

Fig. 4 : Business Model of Craigslist

PURDUE
KRANNERT
SCHOOL OF MANAGEMENT

Data Analysis

Below are the different stages of our data analysis:

1. Data Scraping
2. Data Pre-Processing and Text Processing
3. Model Training
4. Topic Modelling
5. Job Recommendations for a selected Job Posting
6. Job recommendations based on Resume

Data Scraping

Craigslist website was more popular for blue collar jobs and the categories in those were decent. But it was very bad for white collar jobs like the IT or software sector. Hence we extracted data IT/software jobs from multiple cities. We extracted the following fields for around 600 job posting records.

Columns in Craigslist data - Id, Category, Location, Title, Compensation, Type, URL, Description

We found Indeed website has a very good job classification compared to Craigslist. Hence we scraped data from Indeed to train our model. We extracted Job card data for around 2000 records with the following columns.

Columns in Indeed data – Job title, Company, Location, Job summary, Salary, Post Date, URL

All the above scraping was done using the BeautifulSoup library.

Data Pre-Processing and Text Processing

The scraped Indeed data was mostly clean and did not have many issues. We created a dataframe and changed a few column names as required. Next we split the data into Train and Test sets in the ratio 70:30, so that we would have 30% of data to check our model performance.
We are considering the Job description as the predictor variable and the Job Title as our response variable.

Using nltk library we performed the below text processing steps:

1. Converted the job description (predictor variable) into lower case and tokenized it using the word_tokenize function from the 'nltk' library.
2. Lemmatized the above generated tokens to get the root form using the function nltk.stem.WordNetLemmatizer().
3. Removed stop words and punctuation marks from the output of step 2.
4. Combined the output in Step 3, to get a string of tokens.
5. Next, we created a TF-TDF matrix using the 'TfidfVectorizer' with n-gram range from 1 to 3.
6. Repeated Steps 1-5 for response variable 'Job Title'.

7. Repeated Steps 1-6 for test dataset (30%)

Model Training

Next, we ran multiple models on the above processed data to predict the Job title based on the Job description.

1. **Logistic regression** – We started with this model since this is one of the most basic classification models and can act as a baseline model for comparison. We performed logistic regression using the 'sklearn.linear_model. LogisticRegression' function.
2. **Random Forest** – Next we tried the random forest model in the tree family since decision trees generally tend to overfit. We chose this model expecting that overfitting would be minimal and performance would be high since this is an ensemble model. For this we used the 'sklearn.ensemble.RandomForestClassifier' function.
3. **Support Vector Machine** – Next we tried the SVM model since this works well on classification involving higher dimensions. We used 'sklearn.linear_model.SGDClassifier' function.
4. **XGBoost** – XGBboosting is fastest of all the boosting algorithms. Generally boosting algorithms tend to give better accuracies since it is an ensemble of weak learners and specifically XGBoost has proved to give the best results in different types of problems. We used xgboost.XGBClassifier for implementing this.

Validation of the Models

We have used the following Machine Learning models to classify the job postings. Below are the prediction accuracies for different models employed on the test data. We infer that the results not that great due the fact that our test job postings lack proper structure and contain way too much noise. The below figure shows the accuracy level when we used 600 records from indeed but when we increased the number of records the accuracy fell down because of more unique job titles. This could have been avoided with further pre-processing.

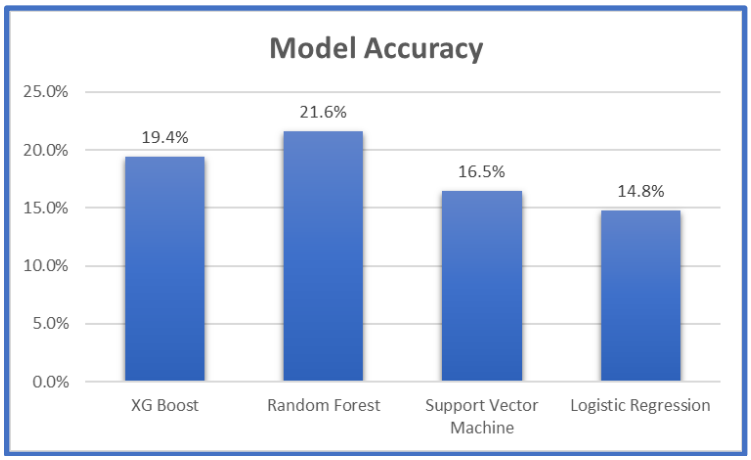

Fig. 5 : Validation Accuracy of models for 600 Indeed records

Eventually, we have decided to go with the Random forest model to classify the job postings as it has a comparatively better accuracy.

Topic Modelling

We used a TFIDF Vectorizer to count word occurrences and produce a bag-of-words matrix. We chose TFIDF Vectorizer over the traditional Count Vectorizer because it gave us more meaningful topics. Next we fed the transformed "document word matrix" into a Non-Negative Matrix Factorization Model (NMF) to produce our top 6 topics. Using the keywords we decided the below topics:

- Topic 0: Experienced Software Developer
- Topic 1: Intern Software Developer
- Topic 2: Software Testing Engineer
- Topic 3: Application/Software Developer
- Topic 4: Data Engineer
- Topic 5: Entry Level Software Engineer

Job Recommendations for a selected Job Posting

After we obtain the topics, we created recommendations using the original TFIDF document word matrix and cosine distance. We have verified this recommendation system by using a particular job posting as input and number of recommendations =3.

Below is the output of the job recommendations provided to the candidate:

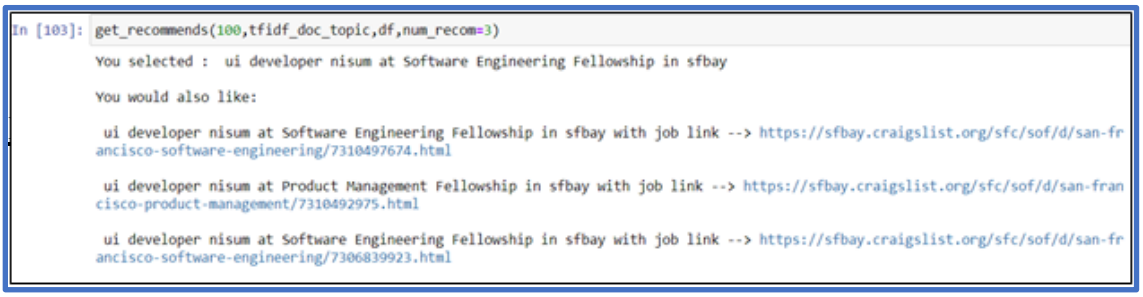

Fig. 6 : Result of Job Recommendation based on Job posting

Job recommendations based on Resume

In this part we scan candidate resume and find similarity match with job requisitions by. We Extract keywords from resume and matches it with the job description and skills in the csv file.

We are comparing the job requisition dictionary and the resume dictionary using the 'gensim.similarities.Similarity()' function which counts the frequency of the word in all the

requisitions and the resume. The integer then becomes the key part of the dictionary and the frequency becomes the value part. We then recommend the top job postings related to the same.

Below is the output of the recommendation based on resume:

```
sorted_d = dict( sorted(jobsugdict.items(), key=operator.itemgetter(1),reverse=True))
print(" Top 5 Jobs suggested for your resume are:")
c=0
for i in sorted_d:
    c+=1
    #print("\n",c,".",i,", Similarity :",jobsugdict[i])
    print("\n {} . {} Similarity : {}".format(c,i,jobsugdict[i]))
    if c==5:
        break
```

```
Top 5 Jobs suggested for your resume are:

1 . Business Intelligence (BI) Analyst | HS Finance Similarity : 0.07929764688014984

2 . Business Intelligence (BI) Analyst | Enterprise Data & Analytics | Multi-City Similarity : 0.07929764688014984

3 . Business Intelligence Analyst | EDA Financial/HR | Multi-City Similarity : 0.07929764688014984

4 . HR Technology Administrator (HRIS) | HR Operations Similarity : 0.0535132959485054

5 . Manager, Accounting | Corporate Accounting Similarity : 0.0
```

Fig. 7 : Job Recommendation based on Resume

Conclusion

Currently, the website is disorganized. Employers have a hard time finding the suitable fits for the company. It is time consuming for the Job seekers to find suitable jobs as each job posting provides very little information (below snippet of the job posting) and they must go through tons of job postings to find a suitable job.

Implementing our model into the website will prove to resolve the problem of irrelevancy and it connects the seekers to providers in an efficient manner. It helps users to easily navigate through the sub-sections and find right posts. What might seem a basic need from a business' standpoint is missing in the current website. Our project takes the first step in improving that part of usability and efficiency of the Craigslist search engine.

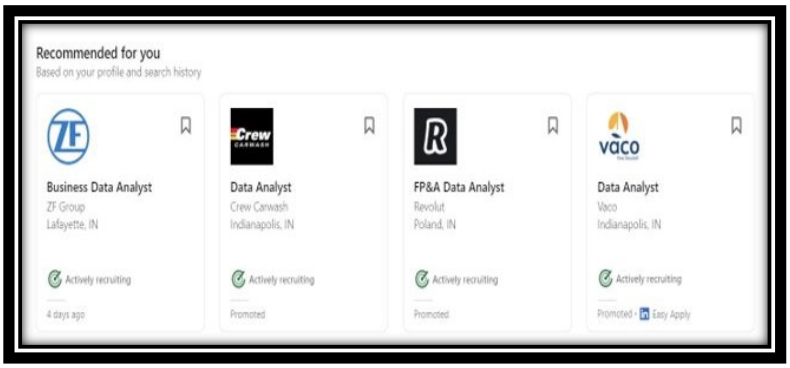

Fig. 8 : Expected Job Recommendation on Craigslist

Further Improvements:

The accuracy scores of our models are low and was expected as we trained the models using limited data points to predict too many target classes. The model was unable to extract the right key words and predict the categories accurately.

These results can be significantly improved if the data from Craigslist is structured. Also, Craigslist needs to implement a template for the Job providers while posting their ads. Craigslist needs to mandate few fields like Employment Type, Salary, Designation, Experience etc for the Job providers and Job seekers so that our model will be improved significantly. Eventually, we believe our model can help Craigslist to be one of the top sites for Job Postings and Job hunting.

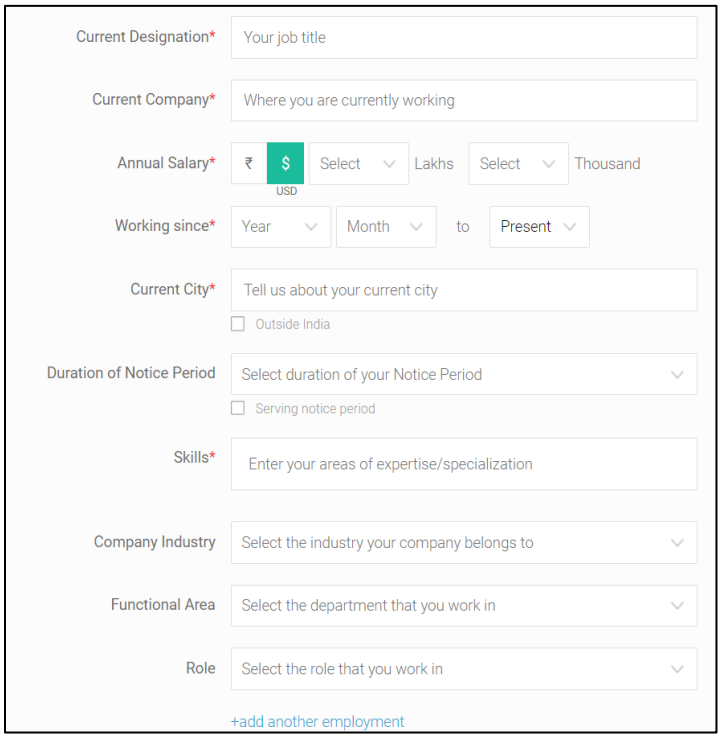

Fig. 9 : Expected template of Resume in Craigslist