Web Mining Final Report

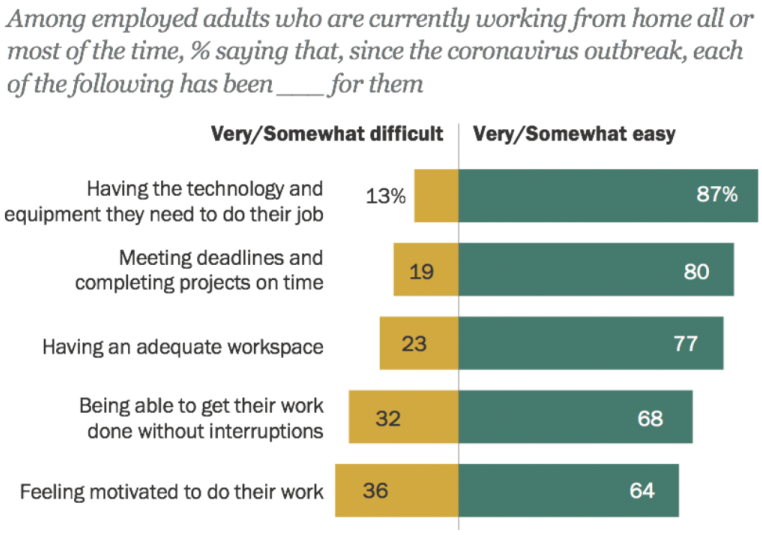
Veronica Abramson and Pranshu Savani

BIA 660-WS
Professor Sun

1. Introduction

The COVID-19 pandemic has undoubtedly changed the lives of many people drastically over just two years. In the matter of just a couple weeks, working people from all around the world left their offices due to upcoming concern about the novel coronavirus. Offices were left empty and within weeks millions of employees were working from their homes. Weeks turned into months, months turned into a year and today, over one and half years after the initial outbreak of the coronavirus, many workers across the globe are still working from home. Is working from home the new normal?

A survey conducted at Pew Research Center shows that many workers would like to telework even after the pandemic is over. Though not exactly a seamless transition, many working adults have mentioned that the transition to telework has been relatively easy. 77% of the responses stated that it has been easy to implement an adequate workspace and 87% stated that they had no difficulties obtaining the necessary equipment and technology needed for their jobs. A whopping 80% also expressed no issues with productivity and stated that they have been able to keep motivated, meet deadlines and complete tasks at hand without major interruptions to their workflow. Furthermore, for those who are working from home by choice, most say they prefer working from home as a major reason behind their decisions with other major reasons being coronavirus exposure concerns and child care responsibilities as well.



With such a positive survey outcome regarding work-from-home situations, many have been left to wonder whether work-from-home is here to stay. Driven by our own curiosity of this new future of the workplace, our group has decided to perform our own research on the subject. We have scraped a dataset of thousands of recent job listings flagged as “remote” in order to analyze what industries and skills are most likely to provide remote working options for their employees. We hope to conduct and report our findings to help those in situations who are desperate for work and whose circumstances allow them to only work remotely, whether that is due to caretaking issues, health concerns or any other issues they may have.

2. Literature Review

Our primary inspiration for this research topic is driven by a data-based article written by Mckinsey, a world-renowned consulting company. Their article “What’s next for remote work: An analysis of 2,000 tasks, 800 jobs, and nine countries” digs into the depth of a remote-work world. Though written in November 2020, McKinsey brings up the idea that “the potential for remote work is determined by tasks and activities, not occupations.” The article defines activities and occupations that can be done from home to better understand the need for remote work. Furthermore, they have analyzed these tasks and activities in a range of countries to understand this remote-work possibility on a global scale. The McKinsey analysis finds that remote work potential is concentrated in several sectors with finance and insurance coming up on top as the highest potential due to the fact that 75% of time spent on tasks can be done remotely without any major disruptions in workflow. The next highest potential is in management, business services and IT with more than 50% of time being spent on tasks that can be done remotely without productivity loss. McKinsey also points out that remote work potential is higher in

advanced economies and those industries with high remote potential are more likely to first adapt a hybrid remote work situation before becoming fully remote.

Our research, however, is not only driven by the data provided by McKinsey. A research paper written by the National Bureau of Economic Research on "What Jobs are Being Done at Home During the Covid-19 Crisis?" provided us with very useful information regarding the remote work subject as well. The paper describes that industries with more educated workers have a higher chance of smaller productivity loss resulting from remote work. The research also concludes that work in capital intensive factories or hospitality and leisure services tends to perform much worse when conducted over Zoom. They also estimate that 40% of large and small firms expect over 40% of their employees to continue working remotely even after the pandemic and roughly 16% of all American workers switching to some sort of hybrid remote work situation with employees going into the office at least two days per week. This analysis shows us that there is a very apparent expected shift to occur in remote work post-pandemic and this will most likely be very concentrated in industries that do not involve hospitality and other strong client-facing jobs.

To form our research, however, the data presented to us in the mentioned studies are only the initial steps. We were driven to dig deeper in the current trends surrounding company decisions to provide remote or hybrid work. We were curious how the data would line up with actual decisions- especially for those roles where it was proven that a remote work environment does not negatively impact the efficiency of the employees. Before the pandemic remote jobs were extremely uncommon. Has the pandemic changed how the future of work life will look like in many industries? The mentioned articles generalize the industries which are most likely to switch to a remote work situation. This is where our research project hopes to dig deeper

between the gaps and analyze the details behind what kind of skills are being considered directly as an option for remote work. We are curious to see if these skills are applicable in multiple industries allowing for a larger opportunity for work-from-home jobs.

Other references listed at the conclusion of this report are scholarly articles written on the complexities of data sets, machine learning algorithms and resources on performing clustering on large scale data sets.

3. Research Question

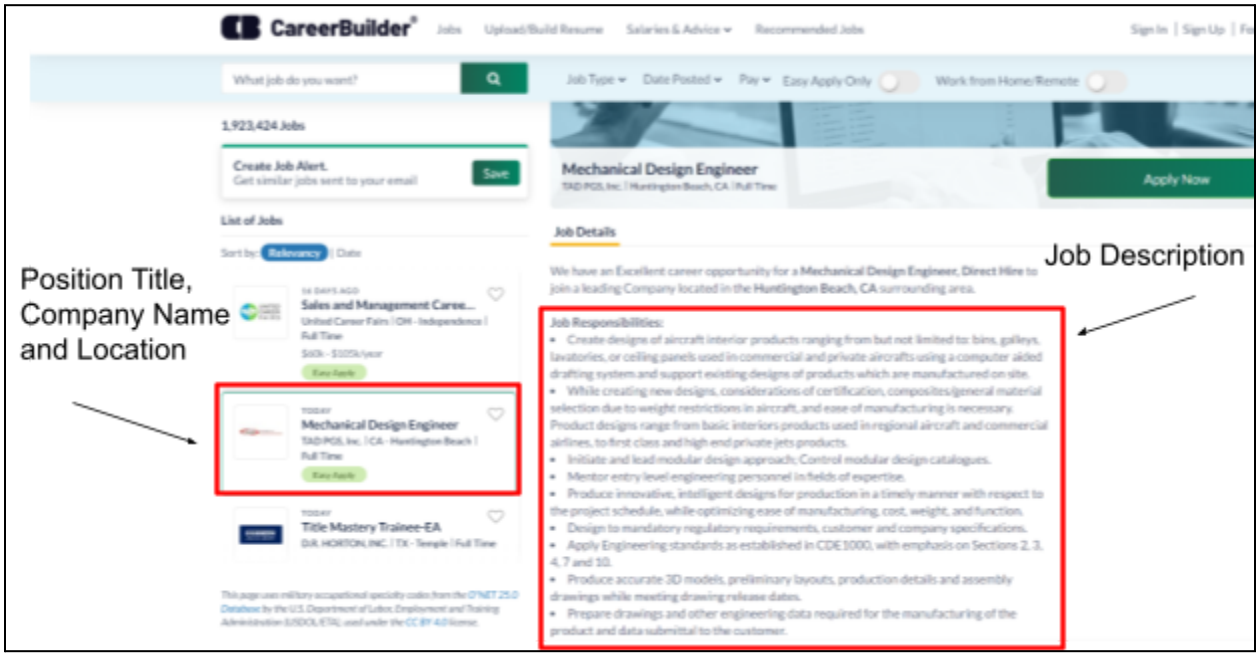
Between the data presented by our references, we have decided to research the true implications of remote work using our own data set scraped from scratch. We will then perform analysis to answer the question “What are the most common skills necessary for remote jobs currently available?” We will then present our findings and make recommendations to those who value remote work options. We believe that our contribution will yield very specific details and skills regarding these types of roles, something the research articles tend to generalize into things like “management” or “hospitality” sectors without detail-oriented information.

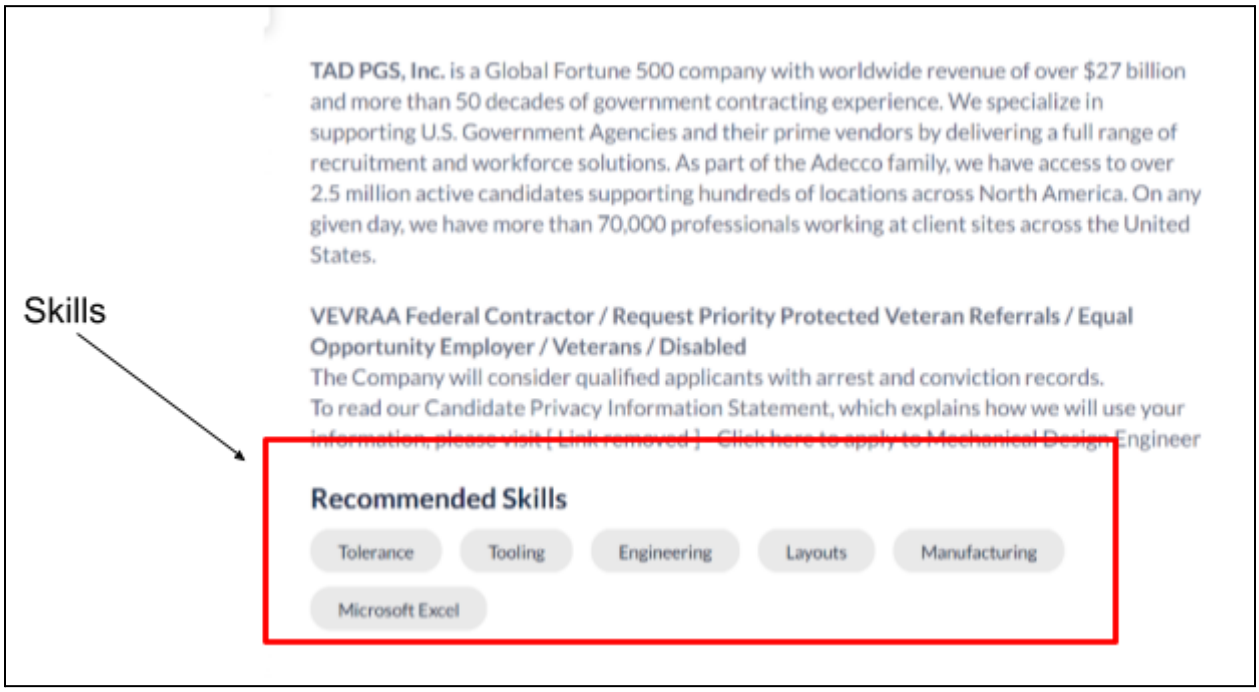
4. Methodology

Data Crawling

Our data is collected from the CareerBuilder website. This is a job search platform that is built around providing tools and opportunities that are specifically catered to a user's interests, skill, background and starting point. This platform is one of the leading job search platforms in North America and Europe. The platform uses machine learning and data science to cater a

user's experience on a personal level, in an effort to assist employers in finding and employing perfect candidates around the world.





We scraped the first 100 pages of the “remote” filtered listings of the CareerBuilder website. Each page has 25 listings, giving us 2500 job listings in our dataset Using a python script, we scraped the following attributes from each job listing:

- Position Company
- Position Title
- Location/Remote/WFH
- Position Description
- Date of listing
- Skills
- Pay

Script code below:

```
BASE_URL_careerbuilder = 'https://www.careerbuilder.com/jobs?cb_apply=false&cb_workhome=true&emp=all&keywords=&location=&pay=&posted=7&radius='

driver = webdriver.Chrome()

num_pages_careerbuilder = 100
job_cleaned_data_careerbuilder = pd.DataFrame()
try:
    for i in range(1, num_pages_careerbuilder+1):

        page_url = ''.join([BASE_URL_careerbuilder,'&page_number=',str(i+100)])
        soup = bs4.BeautifulSoup(requests.get(page_url).content, "html.parser")
        jobs_on_current_page = soup.find_all(class_="data-results-content-parent")
        JOB_BASE_URL_careerbuilder = 'https://www.careerbuilder.com'
        for job in jobs_on_current_page:
            posted_date = job.find_all(class_="data-results-publish-time")[0].get_text()
            job_title = job.find_all(class_="data-results-title")[0].get_text()
            job_details_class = job.find(class_="data-details")
            job_details_span = job_details_class.find_all('span')
            if(len(job_details_span)==3):
                job_location = job_details_span[1].get_text()
            else:
                job_location = "N/A"
            job_company = job_details_span[0].get_text()
            j_link = job.find_all('a',href=True)[0]
            job_url=''.join([JOB_BASE_URL_careerbuilder,j_link['href']])
            driver.get(job_url)
            time.sleep(1)
            source = driver.page_source
            soup_sel = bs4.BeautifulSoup(source, "html.parser")
            job_description_div = soup_sel.select("div.col-2 div.col.big.col-mobile-full")[1]
            if(job_description_div!=None and len(job_description_div)>0):
                job_description = job_description_div.get_text()
                if(job_description==""):
                    job_description="N/A"
            else:
                job_description="N/A"
            salary_div = soup_sel.select("div.data-snapshot div.block")
            if(salary_div!= None and len(salary_div)>0):
                salary = salary_div[0].get_text()
                if(salary==""):
                    salary="N/A"
            else:
                salary="N/A"
            skills_div = job_description_div.find_all(class_="check-bubble")
            list_of_skills=[]
```

```
            for skill in skills_div:
                list_of_skills.append(skill.get_text())
            remote_work = contains_key_word(job_description,['remote','work from home','working from home','work at home'])
            job_desc = re.sub(r'Recommended Skills.*','',job_description.replace('\n',' '))
            job_cleaned_data_careerbuilder = job_cleaned_data_careerbuilder.append({'job_title':job_title,
                                                    'job_company':job_company,
                                                    'date':posted_date,
                                                    'job_location':job_location,
                                                    'skills':list_of_skills,
                                                    'salary':salary,
                                                    'job_desc':job_desc,
                                                'remote':remote_work},ignore_index=True)
except IndexError:
    print('index error')
except Exception as e:
    print(e)
cols = ['date','job_title','job_company','job_location','skills','salary','job_desc','remote']
job_cleaned_data_careerbuilder = job_cleaned_data_careerbuilder[cols]
path="/Users/pranshusavani/Desktop/jobscraped2.csv"
job_cleaned_data_careerbuilder.to_csv(path)
```

Preliminary Data Description and EDA:

Preprocessing and cleaning:

Scraped data:

	Unnamed: 0	date	job_title	job_company	job_location	skills	salary	job_desc
0	0	6 days ago	Program Manager Non Tech 2 - REMOTE	eXcell	WA - Redmond	['Scheduling', 'Interpersonal Skills', 'Multit...	NaN	At [Link removed] - Click here to apply to...
1	1	6 days ago	Software Quality Assurance Specialist	DEVNET Inc.	Work From Home	['Coordinating', 'Attention To Detail', 'Troub...	NaN	DEVNET is a privately held company that was f...
2	2	6 days ago	Part Time Typing Data Entry - Work From Home C...	Red Panda Remote Jobs	CT - Madison	['Advertising', 'Customer Service', 'Data Entr...	$17.00 - $32.00/Hour	Basic Data Entry Clerk Wanted - Work From Hom...
3	3	6 days ago	Customer Service Representative	UnitedHealth Group	NaN	['Client Rapport', 'Finance', 'Health Care', '...	NaN	UnitedHealth Group is hiring entry-level Cust...
4	4	6 days ago	Full & Part Time Work From Home Typing / Data ...	Lotus Dreams Remote Jobs	UT - Salt Lake City	['Advertising', 'Customer Service', 'Data Entr...	$16.90 - $32.10/Hour	Part Time Work From Home Typing & Data Entry...

- Remove irrelevant columns. Eg. unnamed : 0, this is quite common for scraped data converted into csv format.
- Standardize the date format, so that it can be used for visualization
- Treating null values/ missing data if any
    - If there is null value for the job_desc column we drop the entire row.
- Separate list of skills into individual rows for each job, for easier processing and visualization.
- Getting salary values to Float type if in case any operation is required.

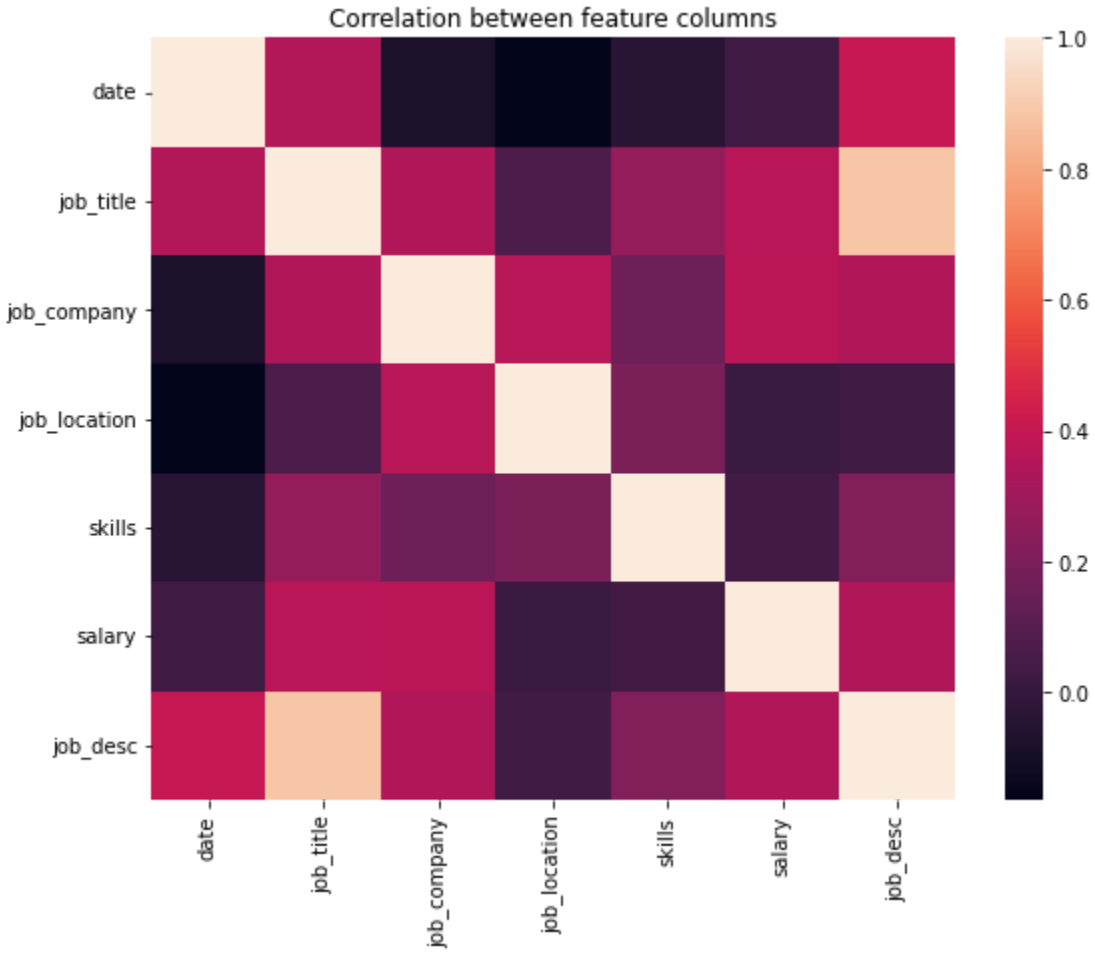
```
dataset = pd.read_csv("/Users/pranshusavani/Desktop/jobscraped.csv")
# Removing Unnamed Column
dataset = dataset.drop(columns=['Unnamed: 0'])
# Replacing 'day' with 'days' in date for better processing
tempData = dataset["date"].replace(to_replace ="1 day ago", value = "1 days ago")
dataset["date"] = tempData
# Replacing date with numerical date values
for i in range(len(dataset)):
    if(dataset.date[i] == "Today"):
        tempData = dataset["date"].replace(to_replace ="Today",
        value = datetime.datetime.now().strftime("%x"))
        dataset["date"] = tempData

    elif((dataset.date[i][-1]) == 'o' and dataset.date[i] != "30+ days ago"):
        today = datetime.datetime.now()
        daysAgo = int((dataset.date[i]).split(' days ago')[0])
        DD = datetime.timedelta(days= daysAgo)
        earlier = today - DD
        earlier_str = earlier.strftime("%x")
        tempData = dataset["date"].replace(to_replace = dataset.date[i],
        value = earlier_str)
        dataset["date"] = tempData
```

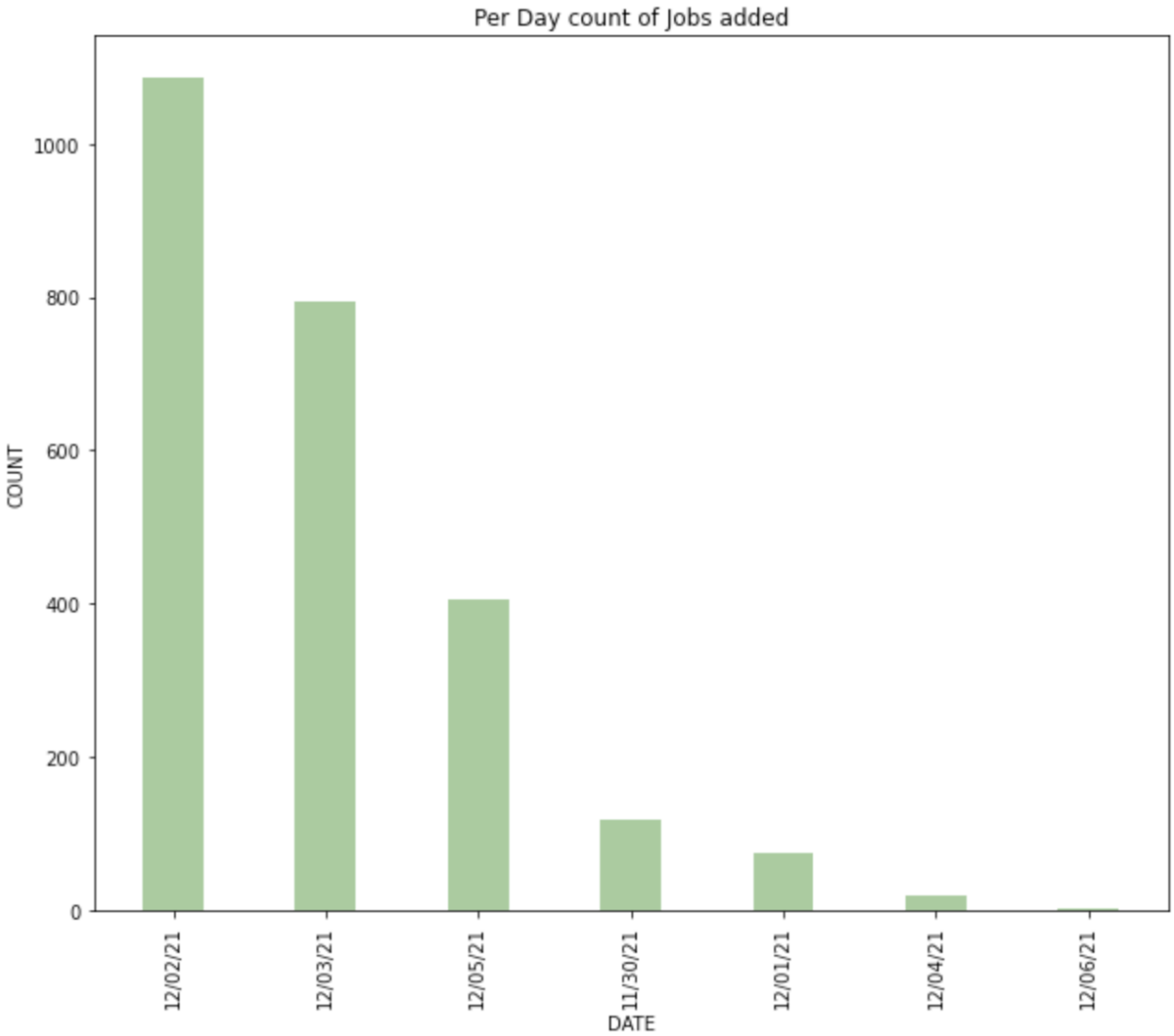
```
# Seperating list of skills into seperate rows for simplicity
tempData = dataset.copy()
x = tempData["skills"].apply(lambda s: list(ast.literal_eval(s)))
tempData["skills"] = x
lst_col = 'skills'
tempData.skills.apply(pd.Series)
tempData = pd.DataFrame({
        col:np.repeat(tempData[col].values, tempData[lst_col].str.len())
       for col in tempData.columns.difference([lst_col])
   }).assign(**{lst_col:np.concatenate(tempData[lst_col].values)})[tempData.columns.tolist()]

cleaned_data = tempData.copy()
# Saving dataframe to csv after complete pre-processing
cleaned_data.to_csv(r'/Users/pranshusavani/Downloads/preprocessed_data.csv')
cleaned_data.head()
```

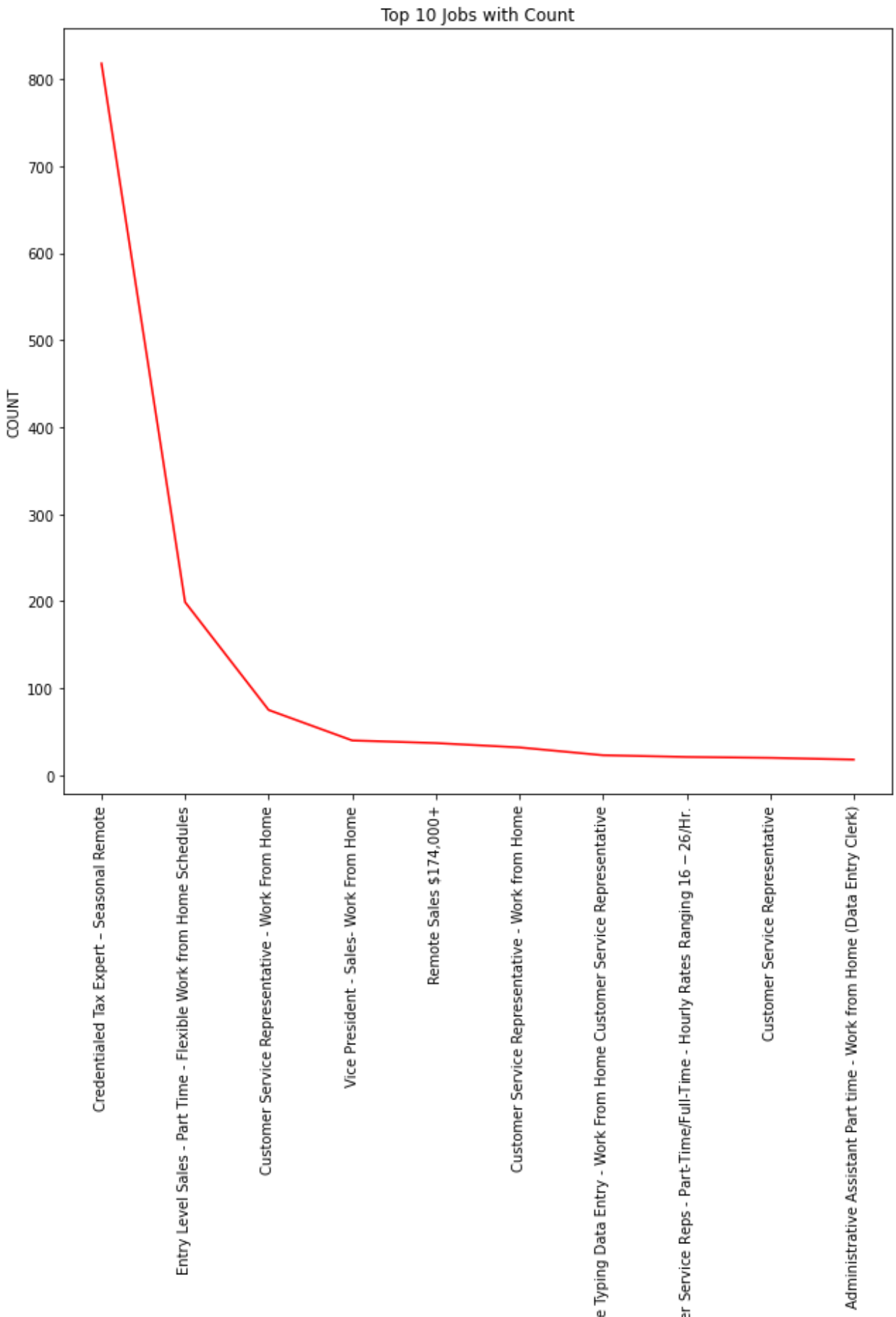
EDA:



The primary idea is to cluster the data based on skills and phrases extracted from the job description, we can see that the correlation between other columns and these two are relatively high when compared to other data which is a good indicator of performance of the clustering.

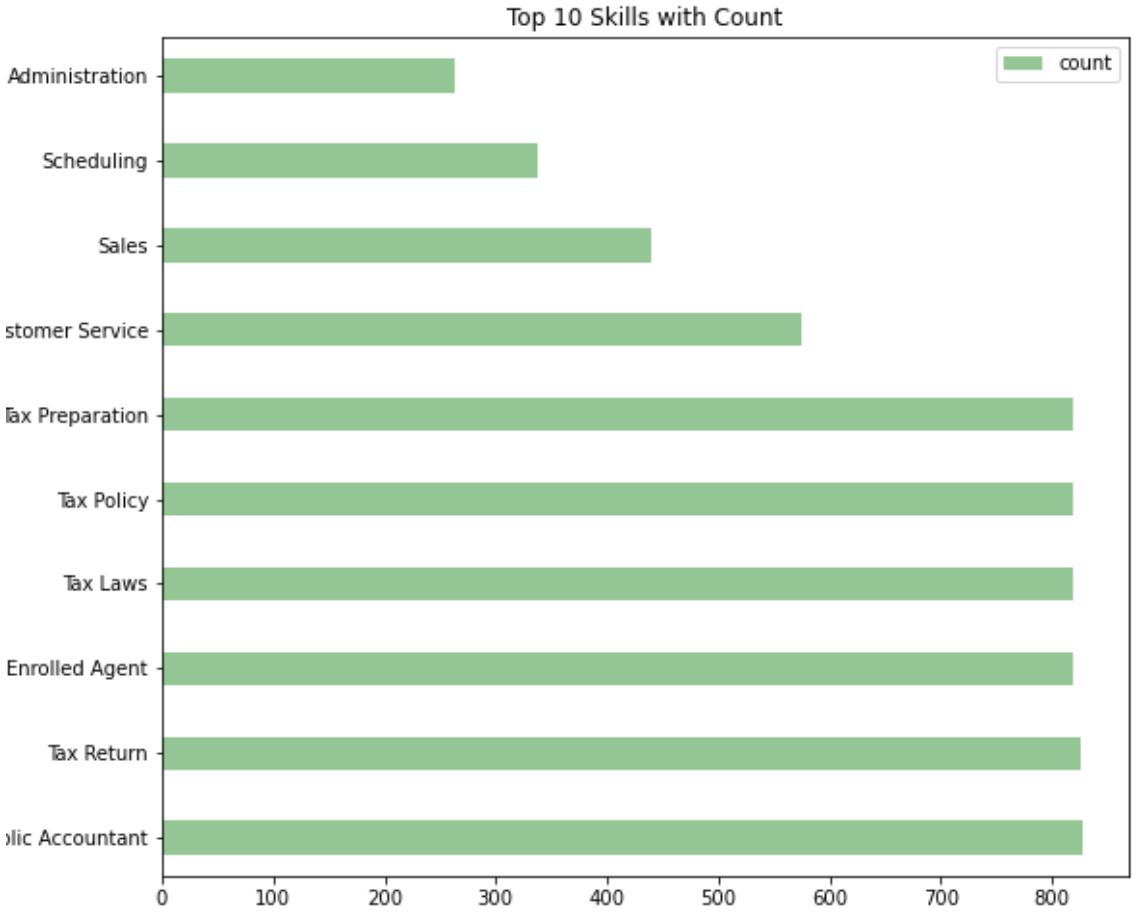


This plot gives us an insight that there's no pattern in the jobs being added on the website, we can see that the highest frequency of the jobs added are on 2nd december and 3rd december(could be the start of the month?), the later jobs being added towards the end of the month show a low frequency of addition in those dates.

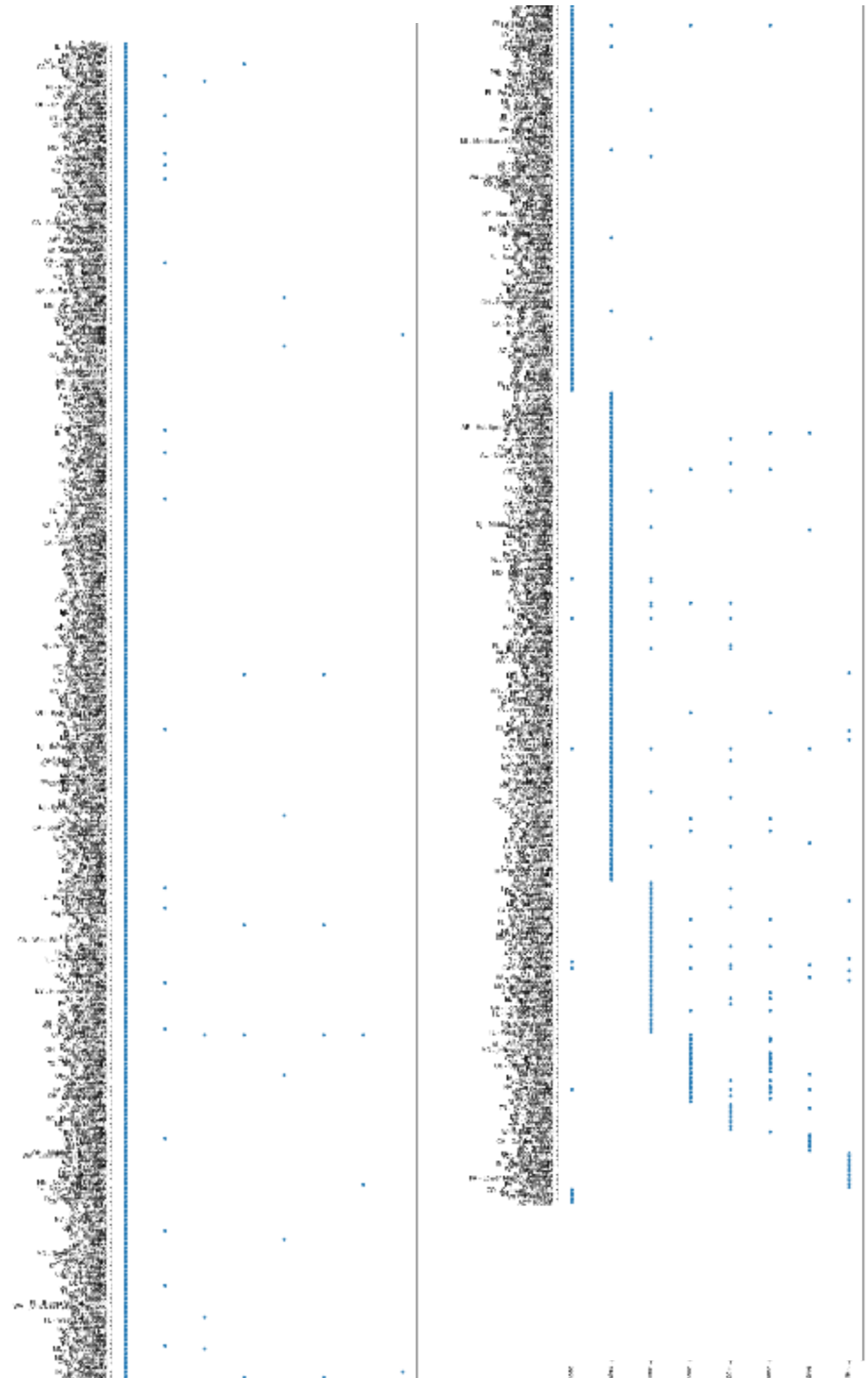

Top 10 Jobs with Count
COUNT
0
100
200
300
400
500
600
700
800
Credentialed Tax Expert – Seasonal Remote
Entry Level Sales - Part Time - Flexible Work from Home Schedules
Customer Service Representative - Work From Home
Vice President - Sales- Work From Home
Remote Sales $174,000+
Customer Service Representative - Work from Home
Part Time Typing Data Entry - Work From Home Customer Service Representative
Customer Service Reps - Part-Time/Full-Time - Hourly Rates Ranging 16 – 26/Hr.
Customer Service Representative
Administrative Assistant Part time - Work from Home (Data Entry Clerk)

The above plot helps us see how seasonal jobs may carry significant weight in the data depending on the time of scraping, for our data, it seems the time the data is scraped overlaps with the seasonal hiring of Credentiated Tax expert Jobs for many companies.

A way to mitigate this would be scraping data multiple times throughout the year to balance out this kind of high frequency presence of just one job.



Again we see the same behaviour for Business services related to Tax/Accounting. But the jobs cannot be reduced in size or eliminated at the risk of information loss affecting the clustering, we keep them and we would scrape more data at different times in the future.



The graph above shows the relationship between top 10 jobs and the locations they are offered at. We can clearly see that in the work from home setting, the same job titles are offered in multiple locations, indicating a growth of opportunities which are not capped by physical infrastructure.

Extracting Activity Phrases:

After Understanding and going through the data we can see that the recommended skills listed for each job may sometimes be quite abstract or vague, it may not be vague by itself ,meaning it can still be considered a relevant skill but it is not context.

Mitigating this can be quite difficult and isn't a well thought out domain yet, but we approached it with the basic idea of Bag of words approach, but it doesn't capture the context which leaves us with the same problem as before. But using POS tagging we can assign parts of speech tags to each word in the job description, following that we can create a chunk tree based on a very small grammar logic to extract what could be relevant prefixes for the abstract skills in the SKILLS column based on similarity check. The steps can be listed as follows:

- Tokenize the text in job description, we used nltk for this
- Remove stop words
- Assign POS tags to each token
- Create chunk tree of words based on the grammar, adjective+nouns+nouns for example 'basic knowledge C#' or 'advanced english writing'
- The above examples are phrases which are the leaves of the chunk tree.

For similarity check we used the Word2Vec model found in the gensim package. Which gives a word vector for each word as the output. Once we get the word vector for the words in each phrase, we calculate the mean vector by averaging the individual vectors and use cosine similarity(using scipy.spatial) to compare them with the corresponding skills in the same row of the job description.

The phrases with a similarity > 0.5 are deemed acceptable and appended to the dataset.

```
def avg_feature_vector(sentence, model, num_features, index2word_set):
    words = sentence.split()
    feature_vec = np.zeros((num_features, ), dtype='float32')
    n_words = 0
    for word in words:
        if word in index2word_set:
            n_words += 1
            feature_vec = np.add(feature_vec, model[word])
    if (n_words > 0):
        feature_vec = np.divide(feature_vec, n_words)
    return feature_vec
```

```
chunkGram = r"""SKILLS: {<JJ.?>+<NN|NNS|NNP.?>+<NN|NNS|NNP.?>*}"""
chunkParser = nltk.RegexpParser(chunkGram)
data_final = cleaned_data.copy()
for idx,row in cleaned_data.iterrows():
    text = row['job_desc']
    skill = row['skills']
    words = nltk.word_tokenize(text.strip())
    filt_words = [word for word in words if word not in nltk.corpus.stopwords.words('english')]
    pos_tagged = nltk.pos_tag(filt_words)
    chunks = chunkParser.parse(pos_tagged)
    chunk_list = []
    for subtree in chunks.subtrees(filter=lambda t: t.label() == 'SKILLS'):
        chunk_list.append(" ".join([a for (a,b) in subtree.leaves()]))
    model = Word2Vec()
    model.build_vocab(chunk_list)
    model.train(chunk_list, total_examples=model.corpus_count,epochs=model.epochs)
    index2word_set = set(model.wv.index_to_key)
    for phrase in chunk_list:
        s1_afv = avg_feature_vector(phrase, model=model,num_features=300, index2word_set=index2word_set)
        s2_afv = avg_feature_vector(skill, model=model,num_features=300, index2word_set=index2word_set)
        sim = 1 - spatial.distance.cosine(s1_afv, s2_afv)
        if(sim>=0.5):
            tempcleaned_data = pd.DataFrame({'date':[row['date']],'job_title':[row['job_title']]\
                ,'job_company':[row['job_company']],'job_location':[row['job_location']]\
                ,'skills':[phrase],'salary':[row['salary']],'job_desc':[row['job_desc']]})
            data_final.append(tempcleaned_data)

data_final.to_csv(r'/Users/pranshusavani/Downloads/preprocessed_data.csv')
```

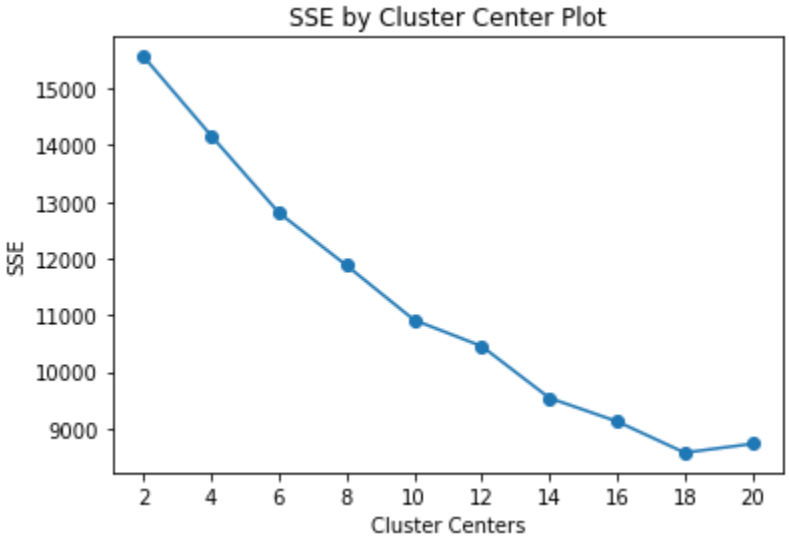
Analytical Strategy

We will be performing two types of clustering for our analysis and comparing their performance.

K-Means:

***k*-means clustering** is a method of clustering sparse data into multiple clusters/classes, that aims to partition all observations into *k* clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into regions close to each of a given set of objects. We'll be using nltk's implementation of the algorithm for clustering.

We find the optimal K-value by the elbow-method, where we run the k-means algorithm multiple times through n clusters for each run, and plot the errors to find an elbow in the plot.



We run the k means algorithm for clusters 1 through 20 and find that we get a clear elbow at 10 clusters.

Using this k value we run the K-means algorithm to predict clusters for our dataset.

For feeding our data into the algorithm, we use TfidfVectorizer from sklearn.feature_extraction.text to transform our text data into a meaningful representation of numbers for machine learning algorithms. TfidfVectorizer weights the word counts by a measure

of how often they appear in the documents and returns a sparse matrix of the weight representations.

```
# generate tfidf matrix
cleaned_data = pd.read_csv(r'/Users/pranshusavani/Downloads/preprocessed_data.csv')
tfidf_vect = TfidfVectorizer(stop_words="english",\
                             min_df=1)


# Train the model
dtm= tfidf_vect.fit_transform(cleaned_data["skills"])

num_clusters=10

# Run Clustering
clusterer = KMeansClusterer(num_clusters, \
                            cosine_distance, \
                            repeats=30,avoid_empty_clusters = True)

# samples are assigned to cluster labels
clusters = clusterer.cluster(dtm.toarray(), \
                             assign_clusters=True )
print(clusters[0:25])
cleaned_data['clusters'] = clusters
cleaned_data.to_csv(r'/Users/pranshusavani/Downloads/data_with_clusters.csv')
```

```
silhouette_score(dtm,clusters)
```

✓ 2.1s

```
0.40441381203137644
```

We also compare this algorithm with **Hierarchical Agglomerative clustering** to see if we can achieve better results with a different implementation for our dataset. In this algorithm, unlike K-means, each data point is considered as an individual cluster and at each iteration, similar clusters merge with each other until k-clusters are formed.

We use ward linkage and k=10 for our clustering to put the two algorithms into the same plane of comparison.

```
# generate tfidf matrix
tfidf_vect = TfidfVectorizer(stop_words="english",\
                             min_df=1)


# Train the model
dtm= tfidf_vect.fit_transform(cleaned_data["skills"])

num_clusters=10

# Run Clustering
clusterer = AgglomerativeClustering(num_clusters)

# samples are assigned to cluster labels
clusters1 = clusterer.fit_predict(dtm.toarray() )
print(clusters1[0:25])
cleaned_data1['clusters'] = clusters1
```

```
silhouette_score(dtm,clusters1)
```

✓ 2.2s

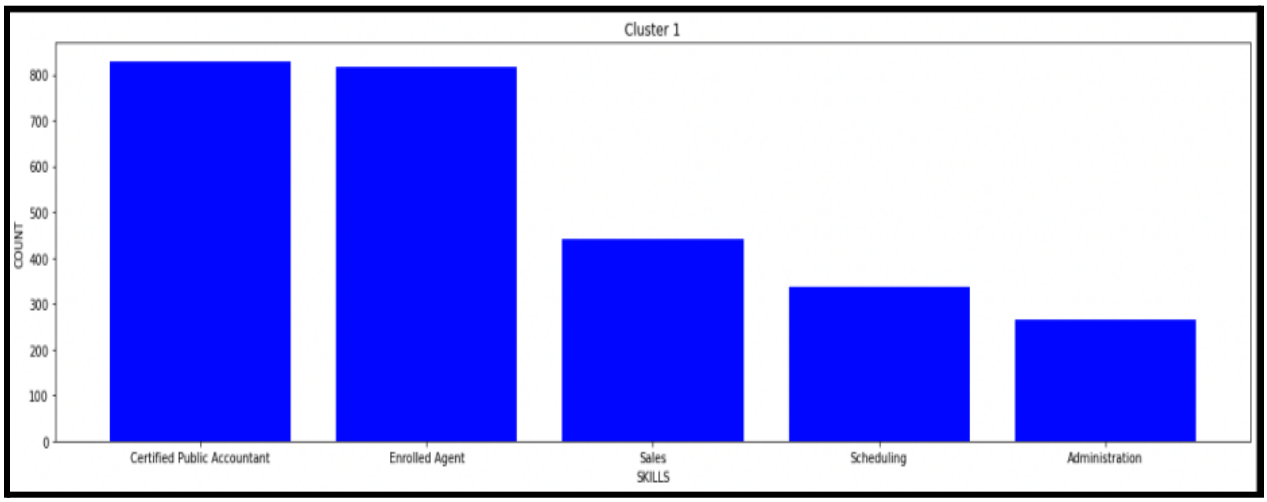
```
0.2198907153765274
```

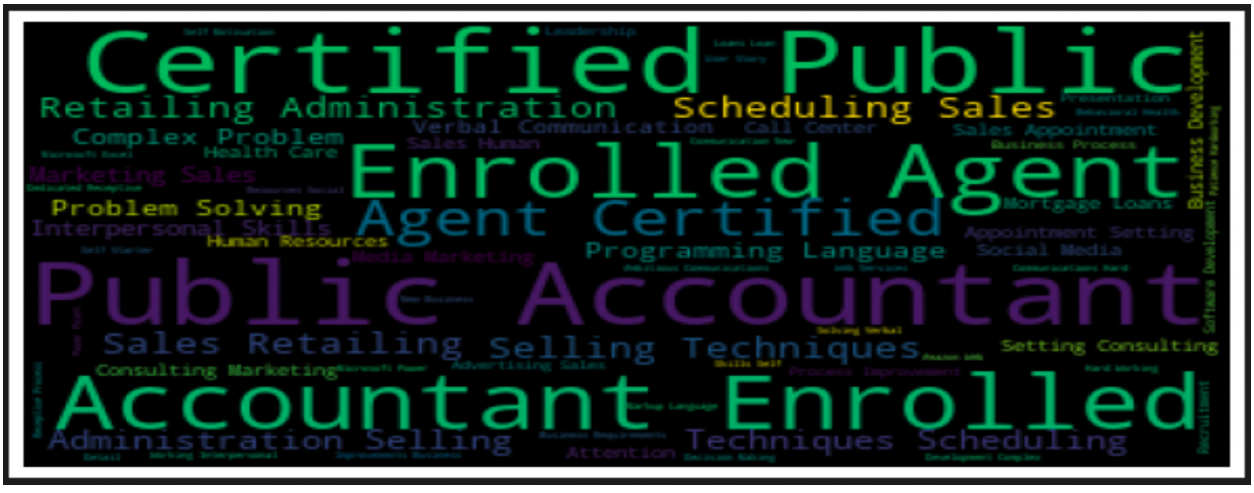
We can see that **K-means clustering performs better on our dataset**.

Results

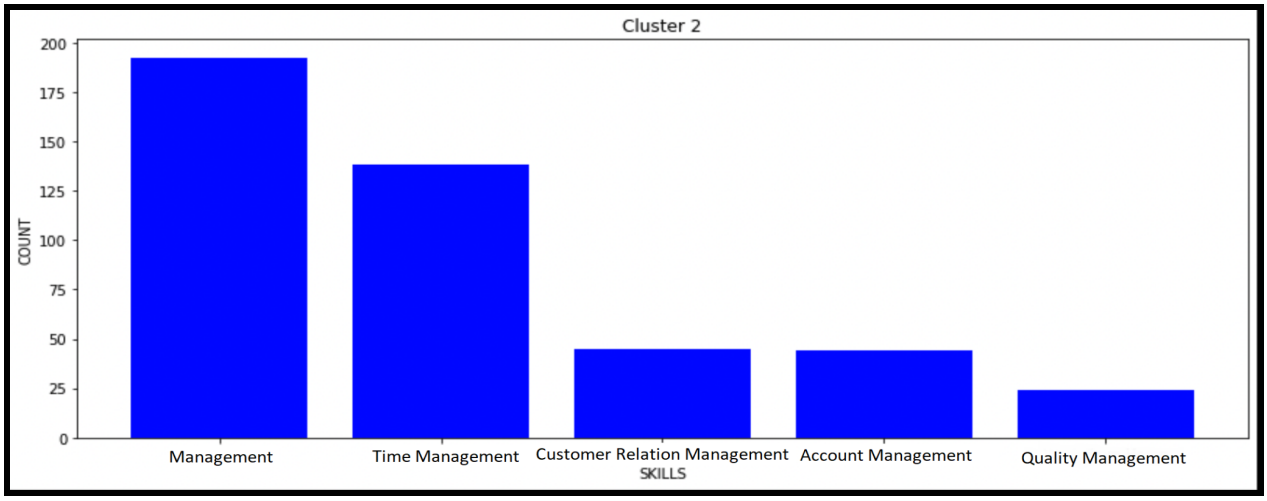
Although 10 clusters were created, we have decided to show the top 5 clusters for our results. Below is a bar chart representing the top 5 skills of each cluster and a Word Cloud showing the most common skills in each respective cluster.

Cluster 1- Functional Skills Cluster

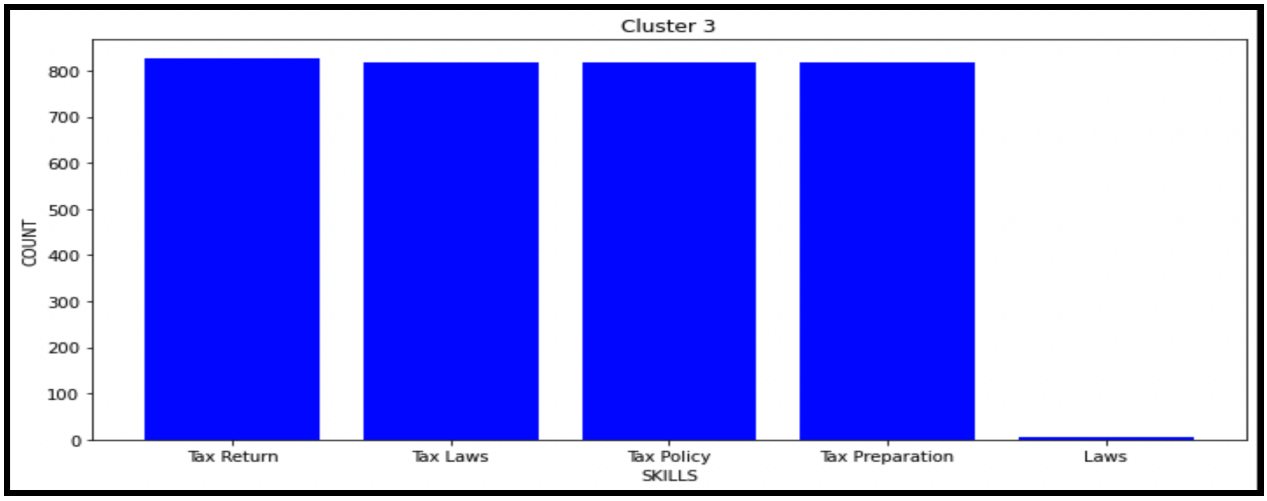




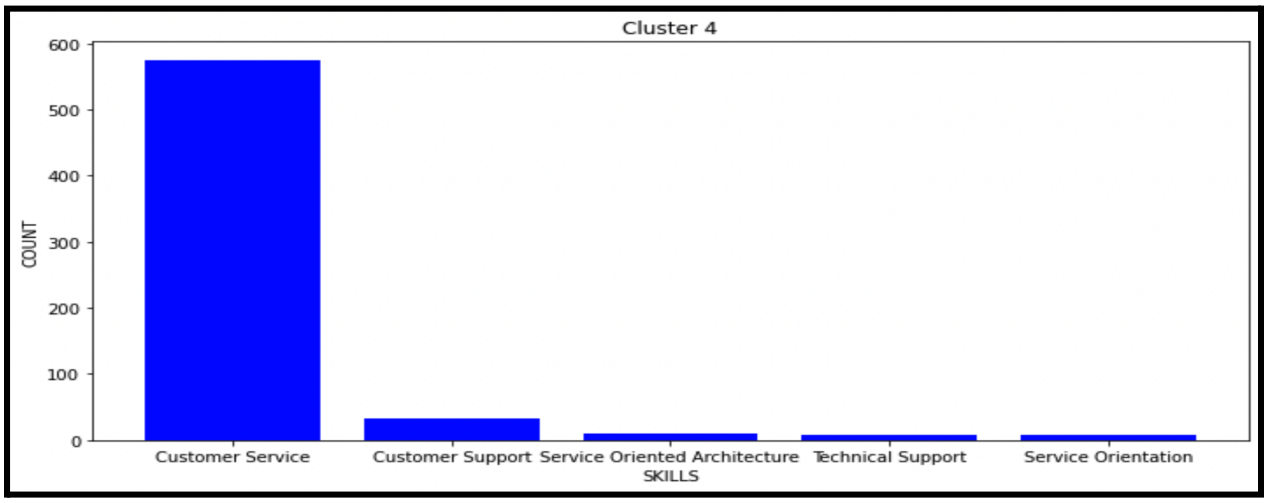
Cluster 2- Management Skills Cluster



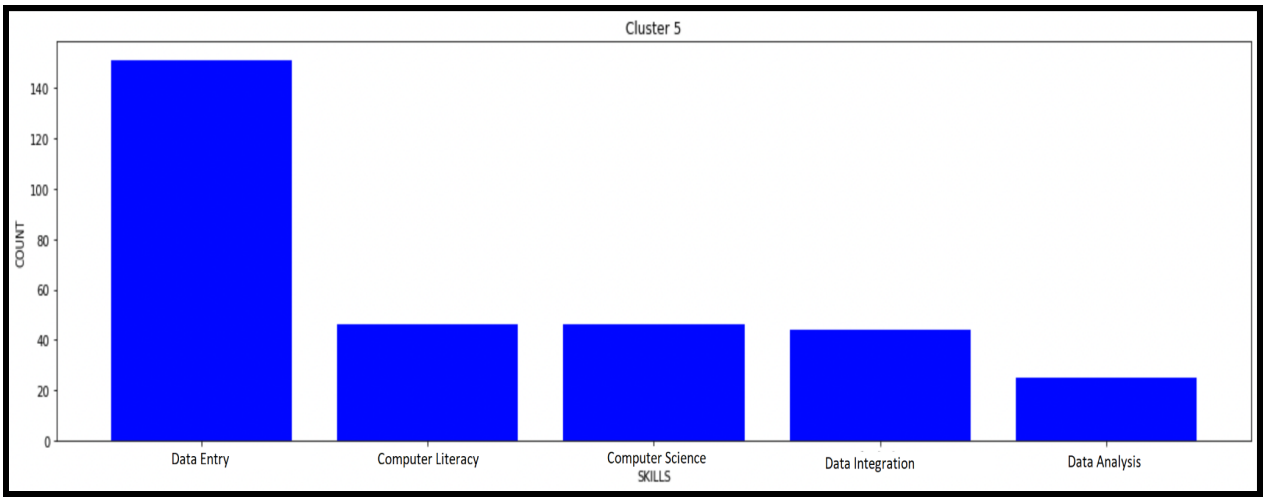
Cluster 3- Business Services Cluster

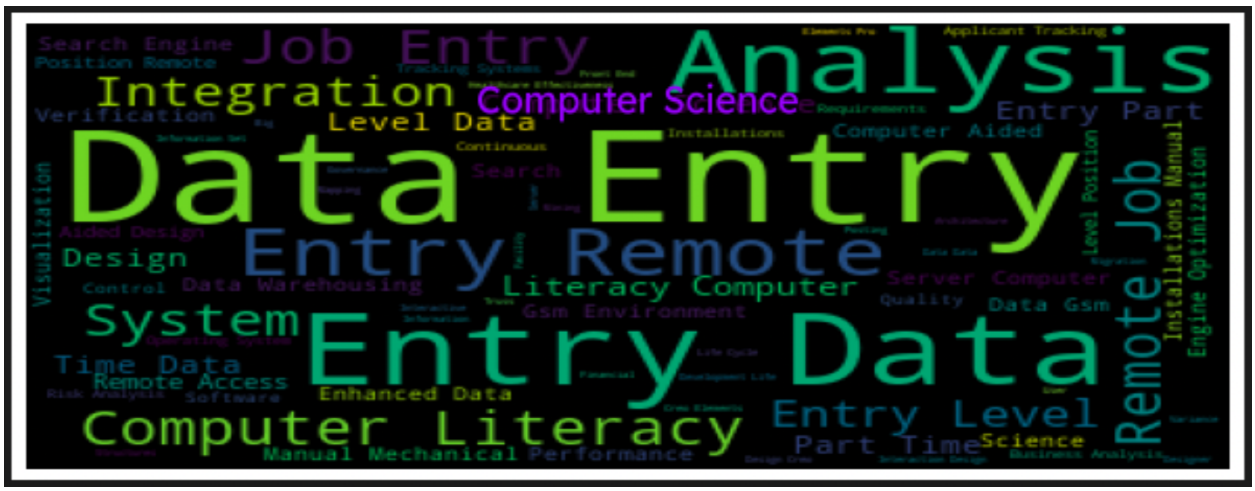


Cluster 4- Customer Services Cluster

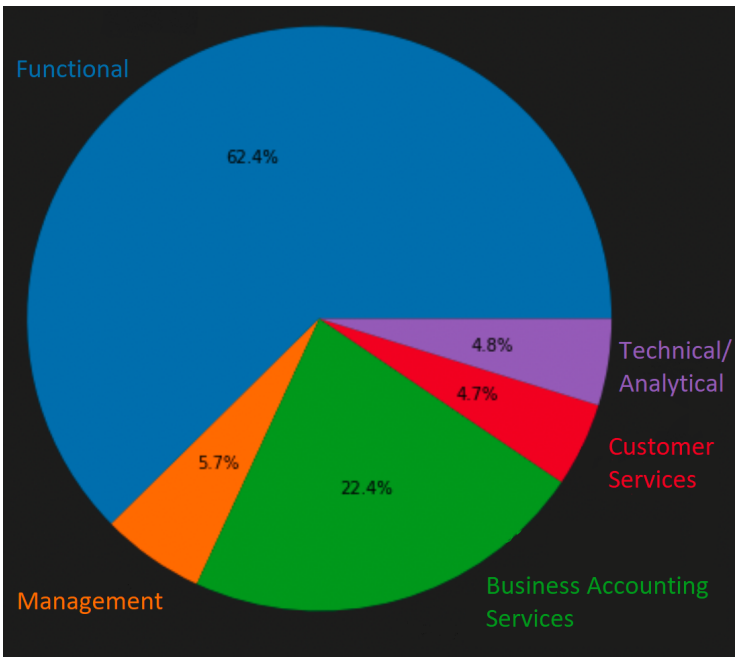


Cluster 5- Technical/Analytical Cluster





We have also outputted a pie graph to represent the size of each cluster relative to each other.



As we can see, the functional cluster is the biggest cluster. The next biggest are Business Accounting Services, Management, Technical/Analytical Skills and Customer Services being the smallest. Based on our data, it can be concluded that business related functional skills are the most commonly seen in current remote work listings. Our findings are in line with McKinsey's conclusion about remote work potential being concentrated in several sectors such as finance, insurance, management, business services and IT. Our findings are also in line with the NBER conclusion that industries with more educated workers have a higher chance of smaller productivity loss resulting from remote work and therefore are more likely to work remote jobs. This can be seen from our results as Certified Public Accounting, which is a highly educated title, is amongst the most common skills mentioned in our analysis.

5. Discussion and Conclusion

Based on our research and results, we have concluded that work-from-home jobs are best interpreted in terms of job skills rather than the entire sector. This is an interesting discovery because it means that people with job roles who use primarily skills that are able to be done remotely, will be able to perform this job remotely without losing productivity. Commonly mentioned skills such as data entry, sales and administration are skills that are used by workers in jobs of all types of sectors from construction to engineering or even retail. These are sectors that were not mentioned by any of our resources as "likely to offer remote work" but still have employees with specific jobs that **can** be done remotely. This opens the opportunity for even more industries to look into the possibility of offering permanently remote job opportunities especially for those people in situations where remote work is their only option to work.

Our methodology does work for the most part. We have successfully been able to scrape a large dataset and make some inferences regarding the common trends in skills we found. We do believe that, however, to get truly meaningful results in order to make accurate recommendations on skills, we would have to scrape a dataset much larger than this one over the course of a much longer period of time (such as a year) in order to truly understand the trends. Our data was only scraped over a period of a month and may have resulted in unfair judgement regarding how important each skill from our analysis truly is. If we conducted the same methodology on a huge data set scraped over an entire year, Certified Public Accounting may very well no longer be the most common skill in the dataset. We do believe, however, that our conclusion regarding the importance of interpreting WFH jobs as a set of skills rather than concentrated industries would still hold true.

We would also like to mention that if we were to perform an analysis such as this one again, we would work to perfect our chunking algorithm in order to produce more accurate results. We made an attempt at implementing such an algorithm, however, it still yielded very general skills such as “work-from-home”, “scheduling” and other skills which are very general and not very specific. Yielding more specific skills will help with more oriented clusters rather than very general ones such as our “Functional Skills” cluster which has a very wide range of skills.

6. References

Bartik, Alexander W., et al. “What Jobs Are Being Done at Home during the COVID-19 Crisis? Evidence from Firm-Level Surveys.” *NBER*, 29 June 2020, https://www.nber.org/papers/w27422.

García, Salvador. “Salvador García Julián Luengo ... - Users.ece.utexas.edu.” *Data Preprocessing in Data Mining*, 2015, https://users.ece.utexas.edu/~ethomaz/courses/dm/papers/data-preprocessing-book.pdf.

Jianliang, Meng. “The Application on Intrusion Detection Based on K-Means Cluster Algorithm.” *IEEE Xplore*, IEEE, 2009, https://ieeexplore.ieee.org/abstract/document/5231545?casa_token=FlALjy7M2f4AAAAA%3ABB146i8QA-JJxFaByd9pgiHGO2p6fO0zOYOdvJgR66Tw7wCjKIYId4otRKcrkpyySd6Q2yYG.

Kanungo, T. “An Efficient K-Means Clustering Algorithm: Analysis and Implementation.” *IEEE Xplore*, https://ieeexplore.ieee.org/abstract/document/1017616?casa_token=6UWv-PHCRLoAAAAA%3AQFrCJKFDvPR7WIEl14GtE1dcqsHmXKQKQ-X535osgY1W7wtCJn3_yQ2CkDencqLP9KTl3V4a.

Lund, Susan, et al. “What's next for Remote Work: An Analysis of 2,000 Tasks, 800 Jobs, and Nine Countries.” *McKinsey & Company*, McKinsey & Company, 3 Mar. 2021, https://www.mckinsey.com/featured-insights/future-of-work/whats-next-for-remote-work-an-analysis-of-2000-tasks-800-jobs-and-nine-countries.

Parker, Kim, et al. “How Coronavirus Has Changed the Way Americans Work.” *Pew Research Center's Social & Demographic Trends Project*, Pew Research Center, 25 May 2021, https://www.pewresearch.org/social-trends/2020/12/09/how-the-coronavirus-outbreak-has-and-hasnt-changed-the-way-americans-work/.

“Research on K-Means Clustering Algorithm: An Improved K-Means Clustering Algorithm.” *IEEE Xplore*,

https://ieeexplore.ieee.org/abstract/document/5453745?casa_token=yyUIn9xZ9vUAAAAA%3A9vaNjOzKBbcDrN4Y02E0XJ9iMukPub0cH5DLW_uLYZo3_PvKAB8oEeVdA_KsiUO0rJX557Lq.

Zhu, Qing, et al. "Analyzing Commercial Aircraft Fuel Consumption during Descent: A Case Study Using an Improved K-Means Clustering Algorithm." *Journal of Cleaner Production*, Elsevier, 8 Mar. 2019, https://www.sciencedirect.com/science/article/pii/S0959652619306407?casa_token=Vtg8wKEKz0YAAAAA%3AFglCjn4tUqwSO522LKltw9A1-MXkWw6PBhREDCY_PGnsJsM4s_nZDH-HfH_mYgx5rhznUTs2jQ.

Project Progress Logs

Progress Logs				
Task	**Person**	**Action**	**Date**	**Comments**
Team created	Pranshu and Veronica	Complete	20-Sep	
Send member names to TA	Veronica	Complete	20-Sep	
Brainstorm project ideas	Pranshu and Veronica	Complete	22-Sep	Considering scraping a job listing website and anlyzing patterns in remote jobs
Meet with professor to discuss idea	Pranshu and Veronica	Complete	30-Sep	Some confusion regarding how to compose clustering. Possibly make remote jobs be a cluster?
Create/submit proposal	Veronica	Complete	9-Oct	Attempt at clearing up confusion. Awaiting professor feedback.
Decide which career website to scrape	Pranshu and Veronica	Complete	10-Oct	
Scrape small sample size of remote job listings	Pranshu	Complete	16-Oct	
Preprocess data and EDA analysis on subset	Pranshu and Veronica	Complete	20-Oct	
Compose and submit mid-term report	Pranshu and Veronica	Complete	28-Oct	Attempt at revision of research question. Awaiting professor feedback
Scrape entire dataset (first 100 pages)	Pranshu	Complete	5-Nov	Finalized research question and finish entire datatset scrape
Preprocess data and EDA	Pranshu	Complete	12-Nov	
Write and run both clustering scripts	Pranshu	Complete	12-Nov	
Perform analysis and report accuracies	Pranshu and Veronica	Complete	18-Nov	
Produce charts for summaries	Pranshu	Complete	19-Nov	
Compile data into written report	Veronica	Tentative	2-Dec	
Create presentation slides	Veronica	Tentative	3-Dec	
In-class presentation	Pranshu and Veronica	Tentative	9-Dec	

Signatures

We pledge our honor that we have abided by the Stevens Honor System.

Veronica Abramson — 12/09/2021

Veronica Abramson — **Date**

Pranshu Savani — 12/09/2021

Pranshu Savani — **Date**