Exploring Data-Related Jobs Around the World

Team 14 Final Report

Benoit Bailly, Marc-Henri Bleu-Laine, Alexander Gurung, Hymee Huang, Haley Xue

Introduction:

Our objective with this project was to improve the job search experience within the US for roles in the data industry by providing detailed drilldowns into desired skills and qualifications obtained from popular job sites (ex. Indeed). Additionally, we created visualizations of geographic trends and segmented job types into a series of sub-classifications based on their descriptions. We chose to focus on technology-oriented roles such as Data Scientist as there is prior research that demonstrates a need for more specific job categorization (Saltz et al. [9] and Davenport et al. [15]). We believe this project to be useful to both job seekers and employers. Job seekers want to optimize their searches for jobs they're qualified for and interested in. Similarly, employers want to reduce the time they spend interviewing applicants who do not match the position's requirements.

Problem Definition:

The data industry is particularly affected by the lack of clear boundaries between data scientist and other data roles [9]. Our solution alleviates this problem in two ways: by giving job seekers better ways to search for jobs that best align with their skillset, and by helping employers target candidates best suited to their business needs by better tailoring their job descriptions. Moreover, few job postings include salary data which is one of the most important factors candidates consider when searching for a job. Our analysis aims to predict missing salary data in order to provide job seekers with a more complete picture of a job.

Literature Survey:

In [2] and [15], the authors attempted to reduce the ambiguity in the definition of data science by comparing big data job postings using Latent Semantic Analysis (LSA) and by defining key topics in data-related jobs. They downloaded 1,200 jobs posts for various roles within the data industry across multiple years from Monster.com. The downloaded jobs were analyzed to discern skills and qualifications required to fulfill each of the roles and to identify how roles changed over time. The main limitation of this approach is that Data Scientist roles can be ambiguous and desired attributes vary significantly based on company and industry.

Researchers in [3] analyzed job posts related to big data using the Latent Dirichlet Allocation (LDA) algorithm which enabled them to create a competency map that includes the required knowledge, skills, and tools to work in big data software engineering. Similar to [2], the paper also focuses on multiple jobs in the big data industry and does not overcome the ambiguous nature and high variance in Data Science postings.

Random Forests are combinations of multiple decision trees [8] that can used for regression and classification tasks. These methods were used with relative success in predicting salaries in a variety of use cases [1], [4]. We employed NLP techniques to glean meaningful data/features from the job descriptions. Common preprocessing includes removing stop words, stemming, etc. as detailed in [4]. Similarly, embedding techniques can be used to convert sentences into a numerical vector

representation. TI-IDF vectorization reflects the importance of a word in relation to the corpus based on the frequency of each word [18]. More modern techniques utilize deep learning and advanced transformer-based architectures, like those used in BERT [16], which has shown great success in accounting for the context of a sentence.

Methods:

Data Aggregation:

The primary risk of this project was data quality issues resulting from web-scraping. Specifically, the format of Indeed's web data is not structured and changes often. However, due to the limitations of job site APIs web scraping was ultimately chosen. The quality of the data is especially important for the topics clustering and predictive salary models, and context sensitivity (Schierle et al. [11]) must be considered to maximize the labels' relevance.

We leveraged methods highlighted in [12] to source data from Indeed in Python using BeautifulSoup, Selenium, and the standard Requests API with relative success. However, we ran into significant problems with rate limiting even when distributing the data scraping process. The data we were able to pull includes the following fields extracted from web pages (and formatted using Regex):

Job Title	Company	Company Rating	Location	Salary	Job Description

As we were only able to pull a few thousand rows of data from Indeed using this method, we chose to combine our data with Indeed and Glassdoor datasets available online (such as that scraped by Andrew Sionek available on Kaggle). Note that although we sourced a dataset of >11K rows of Indeed job postings, our final dataset is much smaller since we filtered for postings with salary information (further illustrating the need for a model to predict missing salary information). After aligning the fields, this combined data was used for our Topic Clustering, ML models, and data visualization.

Dataset	**Size**	**Rows**	**Application**
Kaggle Glassdoor Jobs Dataset	870MB	~ 165K	Tableau Visualizations, Topics Model
Indeed Dataset (scraped postings + data from various sources)	4.4MB	~ 1K	Salary Model

Data Cleaning:

Due to the lack of standardization in job descriptions, extra work was needed to extract meaningful information. We focused on cleaning up the salary, location, and descriptions using a combination of Regex rules and standard string cleanup (e.g. removing special tokens). We chose this subset of the data as it was the most pertinent to our ML models and visualizations, and thus was necessary to ensure good results down the line.

Topics Clustering:

As part of our analysis, we leveraged the unsupervised algorithm LDA [7] on job descriptions to determine "hidden" topics (or sub-categories of jobs) within common job titles. The authors describe LDA in [7] as a generative probabilistic model, as it leverages a three-level hierarchical Bayesian model. The algorithm models documents as a finite mixture of underlying hidden topics while modeling the topics by a distribution over words. This approach is helpful as it enables us to get "topics" that will

characterize data science jobs. After hyper-parameter tuning the topics and applying LDA to a US-centric subset of our data, human interpretation was used to interpret and label each topic. We then assigned each data point to a label to display to end-users, who would then be able to see generated topics for each job and decide which ones relate best to their personal preferences and skillsets.

Our topic clustering method includes a series of pre-processing steps to get the job description data into a proper format:

- **Split job description into passages**: Similar to [2], multiple passages were created for a given job description. We created the passages by splitting the description text based on carriage return. This enabled having shorter sentences that corresponded to approximately one to three topics, depending on the carriage position.
- **Extract relevant passages:** Passages that included key skills previously found to be relevant to data science were kept, and the ones that did not were removed. The list of skills was retrieved from [20] and is presented in figure 7.
- **Word Tokenization:** Split each job description into a list of words. Common bigrams were also tokenized.
- **Stop words Removal:** English stop words such as "I, me, my, etc..." are removed from the tokenized words.
- **Lemmatization:** Usage of vocabulary and morphological analysis of words to remove inflectional endings of a given word
- **Miscellaneous processing steps:** Filtered out words that were contained in less than 5 job descriptions, filter out words contained in more than 65% of the job descriptions.
- **Term Frequency-Inverse Document Frequency (TF-IDF):** Compute the TF-IDF for each word. TF-IDF puts an emphasis on rare words, allowing job descriptions to be more similar when they share less common words.

Finally, we developed the pipeline to perform a grid-search to find the LDA model's optimal hyperparameters that yields the best topic coherence score. We used the C_V coherence score [21] to access the ideal number of topics.

Predictive Salary Model

Using a subset of data points that included salary data, we parsed and normalized the salary data by converting the scraped data into an annual salary rate. For postings in which the salary was given as a range, the midpoint of the range was used to calculate the yearly salary.

Intuitively, a key predictive feature of salaries would be the job location. To better reflect location as a feature, we pulled the cost-of-living indices for each city. For cities that did not have available data, the average cost of living index across all cities in the same state of the job location was used. This allows our end-users to better compare positions between areas, as a better paying job in a more expensive area may make less economic sense. We also parsed the data to create features for job level and category of job. The normalized data was then mapped into one of five salary buckets.

We implemented and compared the model approaches detailed below:

- **TF-IDF Vectorization + Random Forest Classifier**: Job descriptions are fed through a TF-IDF vectorizer and combined with categorical (company, title, level) and numerical features (cost_of_living_index). The input data is then classified into salary buckets using random forest. Logistic regression and SGD were also tested.

- **DistilBERT + Random Forest Classifier**: Job descriptions were first cleaned to remove stop words and special characters. The cleaned data was then fed through a pretrained DistilBERT model [19]. The CLS tokens of the last hidden states were then fed into a random forest model. Since the model can only accommodate a maximum sequence length of 512 and is memory and computationally intensive, we took the middle portion of each description (as this is where the job responsibilities and qualifications are typically found) and truncated the embeddings to 100 tokens before feeding it into the model.
- **BERT**: To test an end-to-end text-only predictive system, we also fed the first 300 tokens into a pre-trained BERT model and appended a linear classification layer with cross entropy loss to predict salary bucket [16]. We hypothesized that this model would perform the worst out of the approaches as it does not include other feature data and relies entirely on what we found to be very noisy inputs.

Interactive Interface:

To visualize our results, we created a Tableau dashboard with several views; more details can be found in the Tableau Dashboards section. Data sources were stored locally as .csv/.xlsx files and pulled into Tableau. The salary model was pickled and then deployed as a widget that is embedded into Tableau. The app was built using Flask and is hosted on Heroku, a cloud platform, which allows the app to be publicly accessible to anyone with the link.

Innovations:

1. Utilizing BERT to generate features for a classification model.
2. Enhancing job search by capturing different skills and tasks for the same job title and allowing job seekers to use this information to apply to jobs that fit their profile.
3. Using LDA to perform topics modeling on job descriptions.

Experiments & Evaluation:

Topic Modeling

For the LDA model, we performed a grid-search on the subset of U.S. job postings from our overall dataset to test a multiple number of topics. We use the C_V coherence score as a metric to estimate the created topic quality based on its top words. Research showed high correlation between this score and human topic ranking [21]. Figure 1 below in the appendix shows the values obtained for each number of topics.

Using the metric, we found that the number of topics which yielded the highest score was 4. A qualitative analysis of the results for this number of topics showed that the algorithm seemed to retrieve four types of skills (as seen in table 1): communication, technical skills, educational requirements, and project and product management. These topic groups make intuitive sense. However, we targeted more granular skills for this project and increased the number of topics to 9. The top 20 words for each topic are presented in table 2.

Adding more topics provided better results as technical skills were then divided into software engineering skills (topic 6), machine learning and algorithms (topic 3), and knowledge of MS Office and cloud technology (topic 5). Since the algorithm was applied to chunks of job descriptions, each job description had multiple topics assigned to it. In order to provide only one topic to a particular job, we selected the mode as the most representative. The use of chunks of the job description significantly improved the interpretability of our results because it allowed us to get rid of unwanted passages about

company information or benefits. Ultimately, we were able to label clear subsets of data scientist-esque roles. The table below shows the name for each of our nine topics, which are displayed in our Tableau dashboard.

Topic 1: Communication and Teamwork	
Topic 2: Project Management/Product Development and Testing	Topic 6: Software Engineering w/ Statistics and Cloud Knowledge
Topic 3: Machine Learning, Statistical Analysis and Algorithm	Topic 7: Research Focus
Topic 4: Educational Credidentials	Topic 8: Project Management (emphasis on control/support)
Topic 5: MS Office, Cloud Tech, Deployment	Topic 9: Software Engineering Paradigm

Predictive Salary Model

Precision, recall, and F1 scores are used to compare our salary models. The TF-IDF + Random Forest model performed the best (App. Fig. 2) with an accuracy of ~0.60, outperforming our DistilBERT + Random Forest classifier (App. Fig. 3) and significantly outperforming our BERT classifier (App. Fig. 4). The F1 score is fairly equal across all salary buckets for the TF-IDF model, but better feature engineering (separately parsing skills as an input, for example) could potential further improve our results. We believe these results can be explained primarily by the noisiness of our data and the importance of key terms and engineered features in salary prediction. Future work could also explore class imbalance, as in particular the high precision on higher salary classes by our first two models may imply a lower relative count.

Tableau Dashboards:

We chose to implement our visualizations in Tableau as it allows the integration of complex graphics, permits embedding of external websites (while keeping their interactive properties), and includes various customizable, interactive features. The visualization has been divided in 4 different sections:

1. World View – describes job postings around the world with an emphasis on the required skills, the most popular job titles per country, and the salary distribution of common job titles.
2. USA View – describes job postings in the United States with a breakdown per state and per industry and highlights the average overall salary for all job postings in our dataset. The salary app is embedded in the view and the user can adjust the inputs, copy in a job description, and obtain a salary prediction.
3. Topics Model – shows a full listing of jobs in the United States and the identified topics per posting. Users can filter by various columns (title, state, etc.) and can filter by a specific topic. Users can also view the full job description for a job ID by adjusting the dropdown for the table on the right.
4. LDA Visualization – provides an in-depth visualization of the LDA analysis. This view is an embedded Jupyter Notebook; users can view the intertopic distance and view most salient terms by topic.

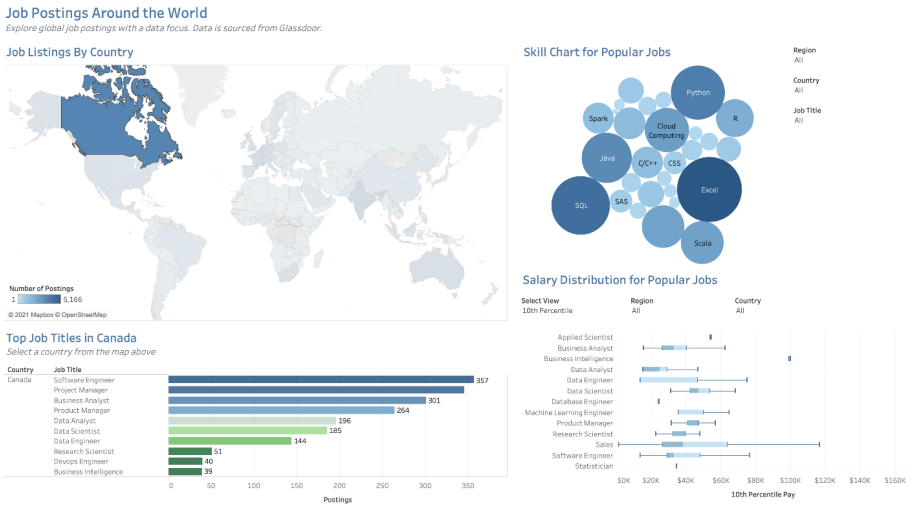

Figure 1: World View

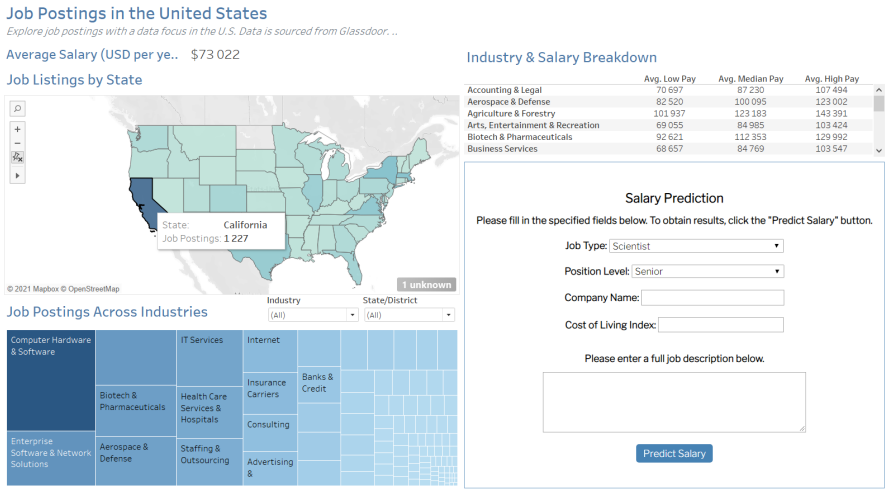

Figure 2: USA View

Figure 3: Topics Model

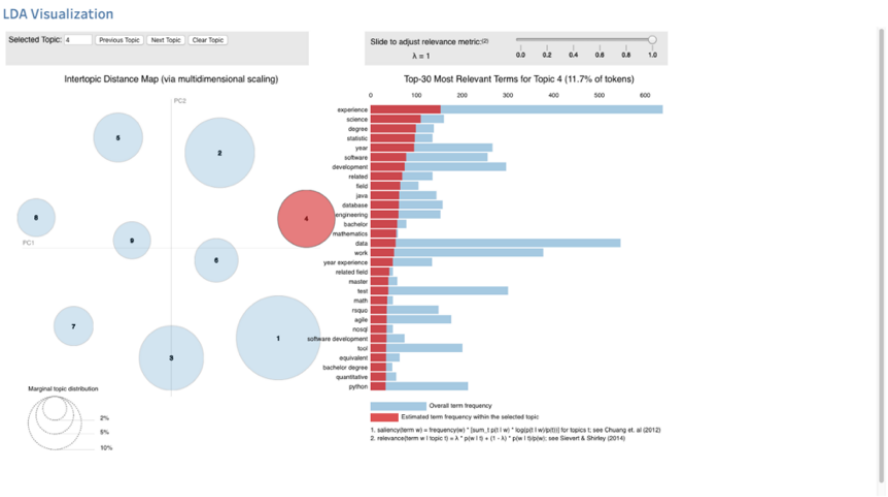

Figure 4: LDA Visualization

Conclusion:

Although we encountered significant challenges in the beginning while gathering and cleaning data, we were able to complete the majority of our goals for this project. Our improved topic modeling produced clear and interpretable sub-sections of the broad “Data Scientist” job title, which we believe is extremely helpful for job seekers hoping to align their background with roles suited for them. Our salary prediction allowed us to, with reasonable confidence, predict a normalized salary given a job description, giving job seekers an insight into what a company’s compensation may look like even when it is not explicitly stated in the posting. Putting the two together with our map visualizations, we believe our Tableau dashboards provide an intuitive and useful product that streamlines the job seeking process and adds value beyond traditional job boards. All group members have contributed equally to this work.

References:

[1] P. Khongchai and P. Songmuang, "Random Forest for Salary Prediction System to Improve Students' Motivation," *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Naples, Italy, 2016, pp. 637-642, doi: 10.1109/SITIS.2016.106.

[2] Halwani. M.A., Amirkiaee, S.Y., Evangelopoulos, N. and Prybutok, V. (2021), "Job Qualification Study for Data Science and Big Data Professions", *Information Technology & People*, Vol. Ahead-of-print, No. ahead-of-print. https://doi.org/10.1108/ITP-04-2020-2021

[3] F. Gurcan and N. E. Cagiltay, "Big Data Software Engineering: Analysis of Knowledge Domains and Skill Sets Using LDA-Based Topic Modeling," in *IEEE Access*, vol. 7, pp. 82541-82552, 2019, doi: 10.1109/ACCESS.2019.2924075.

[4] S. Jackman and G. Reid (2013) "Predicting Job Salaries from Text Descriptions" doi:http://dx.doi.org/10.14288/1.0075767

[5] R. B. Mbah, M. Rege and B. Misra, "Discovering Job Market Trends with Text Analytics," *2017 International Conference on Information Technology (ICIT)*, Bhubaneswar, India, 2017, pp. 137-142, doi: 10.1109/ICIT.2017.29.

[6] H. Wallach, M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009, June). "Evaluation methods for topic models," in *Proceedings of the 26th annual international conference on machine learning* (pp. 1105-1112).

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3, null (3/1/2003), 993–1022.

[8] Ali, Jehad, et al. "Random forests and decision trees." International Journal of Computer Science Issues (IJCSI) 9.5 (2012): 272.

[9] J. S. Saltz and N. W. Grady, "The ambiguity of data science team roles and the need for a data science workforce framework," 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 2017, pp. 2355-2361, doi: 10.1109/BigData.2017.8258190.

[10] D. Düdder et al. , "BlockNet Report: Exploring the Blockchain Skills Concept and Best Practice Use Cases"(2021), arXiv preprint arXiv:2102.04333

[11] Schierle, M., Schulz, S., & Ackermann, M. (2008). From Spelling Correction to Text Cleaning – Using Context Information. In Data Analysis, Machine Learning and Applications (pp. 397–404). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-78246-9_47

[12] Karthikeyan T., Sekaran, K., Ranjith D., Vinoth Kumar V., & Balajee J M. (2019). Personalized Content Extraction and Text Classification Using Effective Web Scraping Techniques. International Journal of Web Portals, 11(2), 41–52. https://doi.org/10.4018/ijwp.2019070103

[13] Keim, Daniel, et al. "Big-Data Visualization." IEEE Journals & Magazine | IEEE Xplore, IEEE, 17 July 2013, ieeexplore.ieee.org/abstract/document/6562707.

[14] Pandey, Anshul Vikram, et al. “The Persuasive Power of Data Visualization.” IEEE Journals & Magazine | IEEE Xplore, IEEE, 31 Dec. 2014, ieeexplore.ieee.org/abstract/document/6876023/authors#authors.

[15] Davenport, T. H., & Patil, D. J. (2012). Data scientist. Harvard business review, 90(5), 70-76.

[16] Devlin, J. et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *NAACL-HLT* (2019).

[17] Teh, Yee Whye, et al. "Hierarchical dirichlet processes." Journal of the american statistical association 101.476 (2006): 1566-1581.

[18] Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (Vol. 242, No. 1, pp. 29-48).

[19] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[20] Song Y. et al. “What Skills Do Data Scientists Need? A Text Analysis of Job Postings. Retrieved April 20, 2021 from https://sites.northwestern.edu/msia/2020/11/30/what-skills-do-data-scientists-need-a-text-analysis-of-job-posting/

[21] S. Syed and M. Spruit, "Full-Text or Abstract? Examining Topic Coherence Scores Using Latent Dirichlet Allocation," *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2017, pp. 165-174, doi: 10.1109/DSAA.2017.61.

Appendix

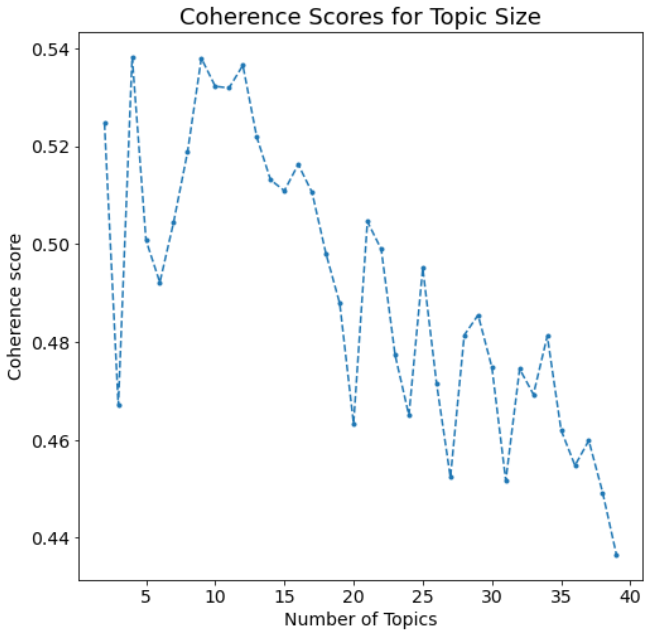

Figure 5-Topic Modeling Hyperparameter Search

Table 1- Top Words for Clusters (4 Topics)

	Top 20 words
Topic 1	project, product, data, test, support, design, research, development, process, requirement, team, management, ensure, work, business, develop, analysis, system, include, provide
Topic 2	experience, science, year, statistic, data, agile, learn , software, development, degree, database, machin e, machine learn, work, test, year experience, java, algorithm, related, field
Topic 3	python, experience, excel, data, knowledge, cloud, microsoft, research, language, proficiency, tool, azu re, statistical, program, bull, model, plus, visualizati on, java, proficient
Topic 4	project, skill, communication, communication skill, write, management, team, excellent, strong, work, experience, ability, project management, verbal, bu siness, manager, project manager, data, write com munication, year

Table 2- Top Words for Clusters (9 Topics)

	Top 20 words
Topic 1	kill, communication, project, communication skill, write, excellent, management, strong, verbal, team, ability, work, project management, manager, experience, business, project manager, write communication, write verbal, technical
Topic 2	project, product, test, development, design, management, requirement, support, ensure, customer, research, work, process, include, team, provide, develop, software, plan, system
Topic 3	data, learn, model, machine, analysis, machine learn, algorithm, research, team, analytics, statistical, experience, develop, project, work, visualization, business, technique, scientist, insight
Topic 4	experience, science, degree, statistic, year, software , development, related, field, java, database, engine ering, bachelor, mathematics, data, work, year expe rience, related field, master, test
Topic 5	excel, experience, microsoft, data, knowledge, profi ciency, tool, spark, tableau, word, powerpoint, offic e, python, hadoop, year, word excel, advanced, dat abase, microsoft office, power
Topic 6	python, language, experience, java, program, azure, script, cloud, program language, linux, language pyt hon, knowledge, research, proficient, bull, matlab, s tatistical, regression, scala, script language
Topic 7	research, project, management, team, data, experie nce, software, test, work, research scientist, clinical, design, project management, plan, rsquo, scientist, support, development, analysis, opportunity
Topic 8	project, data, support, version control, test, researc h, schedule, version, experience, control, maintain, team, application, work, architecture, technology, d jango, understand, project schedule, title
Topic 9	experience, agile, scrum, experience agile, methodo logy, server, plus, database, agile scrum, oracle, pyt hon, work agile, bull, work, year, experience work, a gile methodology, relational, kubernetes, experienc e python

	precision	recall	f1-score	support
120K – 150K	1.00	0.36	0.53	25
60K – 90K	0.49	0.77	0.60	53
90K – 120K	0.56	0.51	0.54	43
<60K	0.72	0.70	0.71	44
>150K	0.89	0.61	0.72	28
accuracy			0.62	193
macro avg	0.73	0.59	0.62	193
weighted avg	0.69	0.62	0.62	193

Figure 2: TF-IDF + Random Forest

	precision	recall	f1-score	support
120K – 150K	1.00	0.24	0.39	25
60K – 90K	0.45	0.64	0.53	53
90K – 120K	0.48	0.51	0.49	43
<60K	0.58	0.59	0.58	44
>150K	0.70	0.50	0.58	28
accuracy			0.53	193
macro avg	0.64	0.50	0.52	193
weighted avg	0.59	0.53	0.52	193

Figure 3: DistilBERT + Random Forest

	precision	recall	f1-score	support
120K - 150K	0.55	0.49	0.52	45
60K - 90K	0.25	0.20	0.22	41
90K - 120K	0.00	0.00	0.00	31
<60K	0.33	0.57	0.42	47
>150K	0.44	0.55	0.49	38
accuracy			0.39	202
macro avg	0.31	0.36	0.33	202
weighted avg	0.33	0.39	0.35	202

Figure 4: BERT

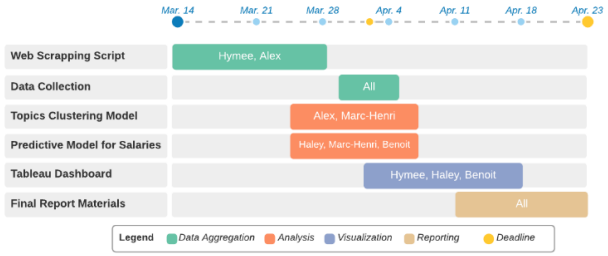

Figure 5: New Timeline

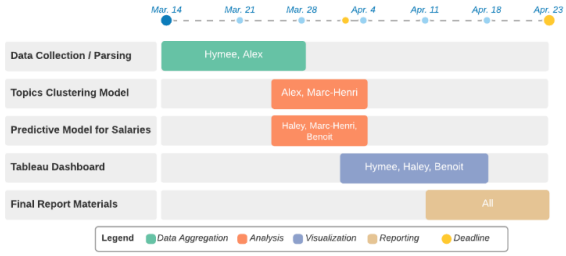

Figure 6: Old Timeline

Statistics	Machine Learning	Deep Learning	R	Python	NLP	Data Engineering	Business	Software	Other
statistical modeling	regression	neural network	r	python	nlp	aws	product design	java	project
probability	boltzman machine	keras	ggplot	flask	natural language processing	ec2	project management	c++	research
normal distribution	k-means	theano	shiny	django	topic modeling	redshift	business development	tableau	ml
poisson distribution	random forest	face detection	cran	pandas	lda	s3	budgeting	d3	ai
survival analysis	xgboost	neural network	dplyr	numpy	named entity recognition	docker	goverance and compliance	html	bi
hypothesis testing	svm	convolutional neural network	tidyr	scikitlearn	pos tagging	kubernetes	systems administration	css	agile
bayesian inference	naive bayes	cnn	lubridate	sklearn	word2vec	scala	communication	git	
factor analysis	pca	recurrent neural network	knitr	matplotlib	word embedding	teradata	a, b test	bash	
forecasting	decision trees	rnn		scipy	lsi	google big query	written communication	latex	
markov chain	svd	yolo		bokeh	spacy	aws lambda	verbal communication	excel	
monte carlo	ensemble models	gpu		statsmodel	gensim	aws emr	problem solving	matlab	
unstructured data	machine learning	cuda			nltk	hive	data analysis	sas	
structured data	optimization	tensorflow			nmf	hadoop		visualization	
math	data mining	lstm			doc2vec	sql		software development	
bayesian statistics	clustering	gan			cbow	big and distributed data		javascript	
scientific method	predictive modeling	opencv			skip gram	database administration		bash	
statistics	time series	object detection			bert	cloud management		linux	
	support vector machines	deep learning			chatbot	backend		jupyter	
	knn	generative adversarial network			bag of words	frontend		shell	
	logistic regression	computer vision			sentiment analysis	graphical models			
	gbm					algorithms			
	feature engineering					data management			
						database management			
						spark			
						mapreduce			
						hbase			
						nosql			
						cloud computing			
						big data			
						distributed computing			
						apache			
						architecture			
						testing			
						azure			
						data warehouse			
						data cleaning			
						data collection			
						data ingestion			
						data quality			

Figure 7: Data Science Common Skills [20]