Classifying Big Tech Employee Reviews

Grace Miles
Princeton Sociology 2019
gmiles@princeton.edu

Madeleine Cheyette
Princeton Computer Science 2019
msc3@princeton.edu

Amy Liu
Princeton Computer Science 2019
al13@princeton.edu

Abstract

The big technology companies – Amazon, Apple, Facebook, Google, Microsoft and Netflix – are worth trillions of US dollars and present great employment opportunities for new college graduates. Many recent grads are concerned with is what is it like to work at these organizations and how the cultures compare between them. In this report we examine a set of 67,529 reviews from the anonymous employee review site, Glassdoor, to build a model that classifies the reviews to each company. We tried a variety of dimension reduction methods, ultimately representing the text as a bag of words with separate vocabularies for the three text based columns. We concatenated the rating data in the reviews to the bag of words representation to feed into four separate models: Multinomial Naive Bayes, K Nearest Neighbors, Logistic Regression, and Linear Support Vector Classification with a focus on Linear Support Vector Classification. Ultimately, this work presents a potential insight into the employee perspective at these high powered institutions.

1 Introduction

As graduating seniors interested in computer science, the reality of working in the tech industry is of obvious interest. Big companies like Google, Amazon, Facebook, Apple, Microsoft, and Netflix are not only highly influential in our daily lives, but are also often prized as ideal environments for ambitious new grads to work. Employee reviews and ratings, though, allow a peek behind these reputations, potentially exposing idiosyncrasies that distinguish what working at these companies is really like.

By analyzing this data set of current and former employee reviews taken from Glassdoor, we hope to gain insight into how the experiences working at these tech giants compare. What might all of them have in common, and where might they differ? Do textual reviews and ratings alone contain sufficiently distinctive information about a company such that the data when deidentified will be enough to identify the company in question? And finally, are the rumors about the reputation about companies the truth, a part of it, or downright misleading? The answers to these questions are of interest not only in the context of big tech companies, but reflect on the helpfulness and comprehensiveness of online reviews, and potentially how both the companies themselves and the existing feedback mechanisms can be improved. In this report, we will first describe our data and approach, then look at multiple data representations and compare the performance of several classifiers in the multiclass classification problem of labeling reviews by their respective tech company. Based on these results, we describe the challenges of and insights drawn from this particular analysis, before suggesting pathways for future progress in this direction.

1.1 Related Work

Analysis of Glassdoor reviews has been explored previously, but no one has attempted to predict a company's name based on the employee's review. Instead, most researchers have used employee reviews to find correlations with companys' economic success: for instance, in 2016, Luo et al. mined Glassdoor reviews to find which aspects of employee satisfaction (communication, safety, etc.) are most correlated with performance of a company [8]. In 2017, Ji et al. similarly examined how different aspects of employee satisfaction correlated with a company's likelihood of financial risk [15].

Other Kaggle users have previously explored the same dataset we use in this paper by plotting numeric rating values for each company. For instance, one user attempted to examine which corporation has better culture by creating plots of employee opinions regarding work balance, culture, and other features by company, and computing the way this correlates with a companys overall rating [2]. Another user examined similar questions by looking at the way overall numeric ratings differ based on whether or not a user is a former or current employee [1]. Others attempted to visualize how numeric ratings differed by job title [9].

Kaggle users have also attempted to uncover more information about the text data in the reviews. One user used LDA topic analysis to find the most common topics within the reviews (some topics that were found, for instance, seemed to be centered around company perks, corporate structure, or pace of work) [14]. Some users also tried to use word clouds to visualize what the reviews commonly talked about [6]. Similarly, others tried to uncover more information about the text portions of the reviews, but via sentiment analysis. For instance, one user used the NLP python package TextBlob to analyze the polarity and subjectivity of text reviews based on each company [6].

However, none of the Kaggle kernels from this dataset tried to predict a company name based on the employee review. Our report thus takes existing analysis a step further: we attempt to find underlying patterns in the data beyond what can be seen through numeric plots, word clouds, or sentiment analysis. In doing so, we use advanced machine learning techniques to examine whether or not company culture can be more precisely measured than what is first seen through the numeric ratings and text portions in the reviews. We extensively explore the dataset, evaluate the oversights in our model, and suggest areas of improvement for future classification tasks as well.

2 Data

The data for this project is from the website Kaggle, which is an online community and repository of datasets for data scientists and machine learners [13]. This data was scraped from Glassdoor and contains detailed employee reviews from Google, Amazon, Facebook, Apple, Microsoft, and Netflix. There are 67,529 total reviews and 17 features for each review, including company, location, employment status of the reviewer (current or former), pros of working at the company, cons of working at the company, and a number of 1-5 star ratings on dimensions such as culture or work life balance.

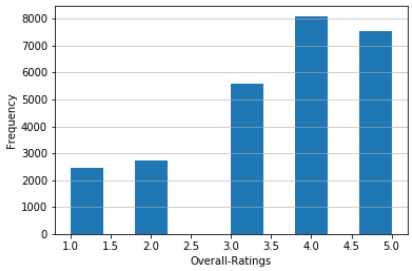
Table 1: Summary Review Information by Company

Company	Total Reviews	Overall-Rating	Percent Current Employee
Amazon	26,430	3.58	66.62%
Apple	12,950	3.96	56.21%
Facebook	1,590	4.51	80.44%
Google	7,819	4.34	59.84%
Microsoft	17,930	3.82	62.96%
Netflix	810	3.41	50.00%

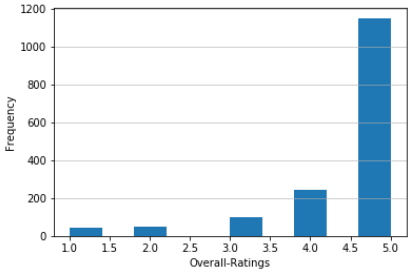
Given our goal of classifying the company reviews we first broke down the feature set into those reviews we believed would be most influential in classification. This meant dropping the column designating the location of the employee, which we felt might give away too much discriminatory information, the link to the review, the date and the job title. We converted the job title column into a binary representation that contained a 1 if the reviewer was a current employee or a 0 if they were

Figure 1: Sample Overall-Ratings Distributions

(a) Amazon employee ratings (n = 26,430)



(b) Facebook employee ratings (n = 1,590)



a former employee. From there we looked at how much missing data was present. The company name, summary, and overall-rating columns were completely filled out. The other ratings columns had a range of 10-20% null entries which we filled in using simple imputation.

In Table 1, one can see the distribution of review counts and the overall rating of what percentage of reviews completed by current employees. The number of reviews per company is not evenly distributed across the dataset, with Amazon having 26,430 reviews and Netflix only having 810. Additionally, the two companies with the highest overall rating were Facebook with a 4.51 and Google at 4.32. Furthermore, in Figure 1 one can see how the number of reviews seems to be correlated with the number of reviews that a company has. Facebook, with the highest rating has a very different distribution of reviews than Amazon.

2.1 Word Frequencies

When we started looking at the text data we quickly realized that there was a high degree of similarity among the words used in each company's reviews. Unsurprisingly "work" was unanimously the most common word used for each company (Appendix: Table 1). Before altering the text data, we first went through each of the columns to remove any mention of the company names in order to make the classification more tractable.

3 Methods

3.1 Subsetting into different datasets and BOW representations

In dealing with the text data we developed two theories about the nature of employee reviews from which we separated the data. First we thought that current employees may have a different tone in their reviews than former employees, so we split the core data into current and former and then from there split these sets into 80/20 train/test sets using `sklearn` method `train_test_split` [10]. In the current/former split we also had concatenated all three text columns (summary, pro, con) into one review column. Given the similarity of words in the overall dataset, though, we wanted to try to keep as much review structure of the data as possible. We were also concerned that a word might have a different meaning in the "pro" column than in the "con" column (e.g. "compensation" should be interpreted differently given one label or the other). Therefore, we made sure to create another dataset that maintained separate "summary," "pro," and "con" columns.

To prepare the text columns in these datasets for our models, we used two methods: bag of words and TFIDF. However, we found that bag of words consistently preformed better with models (Appendix: Table 2).

First, we created bag of words representations for each dataset which specified employee status (datasets which included only current employees, former employees, and the dataset which included both). To prepare the dataset with separated "summary, "pros," and "cons," columns, we made separate bag of words representations for the each of these three columns, then made a dataset

with the concatenated bag of words representations of all three together. The threshold for word frequency used to construct the BoW was one.

In the end, we have four representations to train our model: current employee reviews (vocab length: 11869), former employee reviews (vocab length: 11877), current and former reviews together (vocab length: 11808), and reviews with separated pros, cons, and summary BOW representations (vocab length: 35445). We expected that the datasets containing only former and current employee reviews would be more differentiating to models than the dataset containing both. This is because these sets would have distinctive tones (current employees might leave more positive reviews, and former employees might have left the company with unfavorable conditions) that would lend insight to the model. We also expected our vocabs dataset, which represents the split vocabularies, to perform the best because it would maintain more of the underlying structure of the data.

3.2 Attempted Dimension Reduction

The downside of creating bag of words representations of the data is that it creates high dimensional data with many features, most of which are 0, since it generates word counts for each word in the reviews. We tried to reduce the dimensionality of our bag of words representations with SelectKBest feature selection using the chi2 score function (Appendix: Table 3). We also tried to use PCA to reduce our feature set to 200 and 400 dimensions, but this also did not improve accuracy (Appendix: Table 4). Reasons feature selection on bag of words, PCA, and TFIDF representations may not have helped the accuracy of models are further elaborated in Section 5.

3.3 Models

3.3.1 Baseline Model: MNB

We use the Multinomial Naive Bayes model as a baseline. Multinomial Naive Bayes serves as a good baseline since it is a simple probabilistic classifier: it calculates the probability of each company label using Bayes theorem. Due to its simplicity, Multinomial Naive Bayes often has fast performance. However, MNB makes a strong assumption that every feature in the data is independent of other features. Since our data likely does not have independent features (for instance, the same user who leaves a strongly negative summary will likely leave a very low company rating) we hypothesize that the Multinomial Naive Bayes model will have poor accuracy performance compared to other models which do not make the same naive assumption. Thus, Multinomial Naive Bayes provides a simple, fast, and naive baseline which we aim to improve [5].

3.3.2 K Nearest Neightbors

If there is a lot of overlap in types of reviews for each company, K Nearest Neightbors will likely preform poorly since it classifies samples based on the closeness of other samples. We hypothesize that this model might have better accuracy on PCA data if the PCA data is able to successfully separate the data [12]. Since our data is very high dimensional after bag of words transformations, we only analyze K Nearest Neighbors with the reduced dimension data from PCA.

3.3.3 Logistic Regression and LinearSVC

We also train Logistic Regression and LinearSVC models. LinearSVC is similar to Logistic Regression in that they both try to fit a line to best separate data. However, LinearSVC is often advantageous in that it minimizes a hinge loss while logistic regression minimizes logistic loss. Logistic loss diverges faster than hinge loss, which means it is more sensitive to outliers. It is likely that our data contains a fair amount of outliers, since some employees at a company may have had very different experiences from the general consensus at that company. As such, LinearSVC should perform similarly to logistic regression, but will likely perform better [4]

We use the L1 parameter because it helps implement feature selection among our high dimensional data, as it encourages a sparse solution in the optimization function [3].

4 Results

As detailed above, we trained four models – Multinomial Naive Bayes, K Nearest Neighbor, Logistic Regression and Linear Support Vector Classification – in order to try to classify which company a reviewer was writing about based off of their Glassdoor submission. We expected dimension reduction to be very helpful for K Nearest Neighbors and the data containing all three text columns and the ratings columns separately to be the most informative. Out of our datasets, we expected that datasets containing only current or only former employee reviews, and the dataset with separate BOW representations for "summary," "pro," and "cons" to preform the best.

4.1 The Data Representation

We initially started by making four BOW datasets on our datasets. We refer to them as 'full,' 'current,' 'former' and 'vocabs.' Full is the baseline with all reviews concatenated, regardless of employment status. Current contains only current employees and former only contains former employees. With full as the baseline representation, we tested current versus former because we expected that the words used and tone might be different among people who were still employed versus those who had left or been terminated. Across all the models, we found that the current and former groups had similar accuracy, but consistently performed better than the full baseline. As expected, the vocabs dataset out-performed the baseline as well as the current/former groups (Table 2). For the rest of this section we will discuss the performance on the split vocabulary dataset.

Table 2: Accuracy by model and dataset

	Full	Current	Former	Vocabs
MNB	0.68	0.68	0.68	0.70
Logistic Regression	0.71	0.72	0.71	0.72
LinearSVC	0.69	0.70	0.69	0.71

4.2 Basic Model Performance

We chose MNB to be our baseline model and hypothesized it to perform the worst among the four. The accuracy for MNB on the vocabs dataset was 70%. In classification problems, accuracy is the number of correct predictions made divided by the total number of predictions made. So in this case for MNB, it correctly predicted the true company for approximately 70% of the reviewers in the test set. We hypothesized that Logistic Regression would be an improvement upon MNB, which it is: the accuracy score increased to 72% with this model. We then tried K Nearest Neighbor classification, expecting it to perform better than MNB, but found that with the large dataset combined with our limited computation power made it incomputable. When we did PCA and then fed the reduced dataset into KNN we got an accuracy score of 48% which was much lower than expected, possibly because of the similiarities between the words in each company's reviews made clusters in data indistinguishable. From there we also hypothesized that LinearSVC would perform the best since Logistic Regression can be sensitive to outliers and is less suited to large datasets [4]. Additionally, since the reviews have so much similarity, the dataset is less linearly separable, an aspect which LinearSVC handles better than Logistic Regression [11]. We found that LinearSVC performed slightly worse with a 71% accuracy. We decided to investigate what was causing LinearSVC to perform slightly worse than Logistic Regression, and see if we could improve its performance.

4.3 Adding the Ratings Data

Since we also wanted to analyze how numeric ratings in the company reviews impact accuracy, we first prepared them to be a binary representation via one-hot-encoding, then concatenated this column of the data to the bag of words representations of the pros, cons, and summary data.

We tested how adding the ratings data affected the prediction accuracy with LinearSVC. Given the low variance in ratings found in the data exploration stage we expected that the rating on their own would not be good predictors, but that adding the ratings to the text data would improve the accuracy. As we expected, using the ratings alone with LinearSVC produced only a 45% accuracy, but adding the ratings to the LinearSVC vocab representation made the accuracy 72%.

4.4 LinearSVC Deep Dive

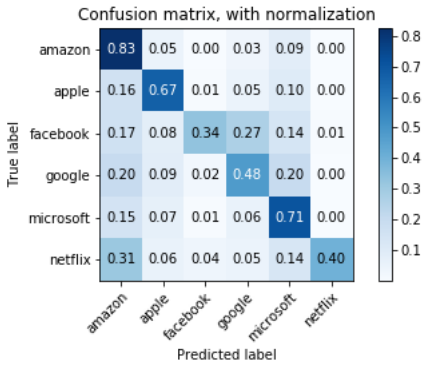
In analyzing the performance of LinearSVC, we noticed the model was better to distinguish some companies rather than others. In Table 3 one can see how much the recall, or the percentage of total relevant results correctly classified, varies widely by company. Amazon has a high recall score of 0.83 but Facebook is at a mere 0.34. One of the main differences between these two companies is the support (or number of reviews) the model had in the training process. Recall is seems more sensitive to the size of the data, but precision, a measure of result relevancy, appeared to be a more stable: for instance, some companies with little support such as Netflix can still have high precision.

Table 3: Classification Report of LinearSVC

	Precision	Recall	F1-Score	Support
Amazon	0.75	0.83	0.79	6522
Apple	0.72	0.67	0.69	3237
Facebook	0.49	0.34	0.40	390
Google	0.56	0.48	0.52	2003
Microsoft	0.70	0.71	0.71	4502
Netflix	0.73	0.40	0.52	229

Figure 2 presents a confusion matrix which in which row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). In this one can see where our algorithm accidentally predicted the wrong class. Interestingly enough one can see that our model struggled distinguishing between Amazon and Netflix employees. 31% of the time LinearSVC predicted Amazon when it should have been Netflix. A company such as Facebook was hardly ever predicted when the true label was not Facebook, but it was often confused with Google. LinearSVC predicted Google 27% of the time when the true review was Facebook.

Figure 2: Confusion Matrix of LinearSVC



5 Discussion

5.1 Data Representation

The best accuracy was obtained using the bag of words representation of review summary, pros, and cons in addition to the numerical ratings. We suspect that this was the most effective feature set because the keywords mentioned in the reviews are much more discriminative than the ratings alone, which are relatively similar on average across the companies but can add information about how positive or negative the review was in reference to those keywords and overall.

The high degree of overlap among the reviews is also consistent with the improvement in accuracy when we use a low word count threshold, and refrain from using dimensionality reduction methods like PCA. Although we hypothesized that adding the ratings data would result in greater improvement if not overwhelmed by our 35,000+ BoW features, transforming this feature set into

400 PCA components only captured 62% of the total variance, was very computationally expensive, and showed inferior performance.

Similarly, the improvement observed in BoW and lack of improvement using TF-IDF is likely due to the reduction in features necessitated by computational limitations. For computational feasibility given our large data set, we adjusted the minimum and maximum frequency parameters and only included unigrams to limit the number of resultant features, limiting to some extent the potential advantages of the model. Additionally, the TF-IDF model may not be optimal for our problem given that we are interested in multiclass classification involving a small number of classes each with a very large number of reviews. In this context, the counts used in the BoW model may be a more desirable representation of the reviews than the TF-IDF values, which are adjusted on the basis of inverse document frequency.

5.2 Most Important Features

Given that the best performing model was LinearSVC on the combined BOW and ratings data, we then inspected the weights assigned to each feature, which are simply the coefficients in the primal problem [4]. The top 15 features corresponding to each class label is given in Table 4. As expected, these words are relatively discriminative, so no ratings-based features appear. A notable share of the words fall into the category of direct company-identifying information, such as the location of the companies (”Cupertino,” ”Redmond,” ”MTV” for Mountain View, and ”MPK” for Menlo Park), which happen to all differ, the names of the CEOs, or even abbreviations of the company name (”FB”). The remaining words fall more in line with our initial objective of harnessing the language and topics which arise for each company. Indeed, the table does reflect some commonly known aspects of each company’s reputation: ”frugal” and ”24x7x365” as cons for Amazon, ”philanthropi,” ”mastermind,” and ”quirki” for Google, and ”nazi” and ”overdr” for Netflix. In this sense, it does seem that these rumored traits do, for better or for worse, have some basis in true employee experience. Of 128 reviews in the test set containing the word ”frugal,” 100% of them were correctly classified as being Amazon.

Still, it is important to note that each of these words represents a small share of actual reviews, appearing on the order of tens to several hundreds of reviews in the training set and many fewer in the test set. The vast majority of reviews, as seen in Table 1 (Appendix) was composed of the same generic words, making separation difficult. This is exacerbated by the relatively bland and limited vocabulary that employees have to describe various elements of the company which only indicate magnitude (”good,” ”poor”) but lack distinctive texture. Often, words used frequently enough to be seen as providing signal are not particularly unique. The example of the word ”frugal” is deceptively ideal because ”frugality” is one of Amazon’s famous 14 leadership principles, which are touted extensively throughout the company. If this were not the case, reviews would likely be a more diffuse mix of language around stinginess, cost cutting, thrift, etc. In this regard, the results suggest that the classification of these six companies, particularly from entirely de-identified reviews, is likely still rather clumsy, impaired by weak and overlapping signals. One example of a review archetype is the summary ”fulfilling work,” pros ”best company to work for!! focuses on individual contributions,” and cons ”sometimes there is high work pressure.” This review was left by a Google employee, but was misclassified as Amazon, perhaps because of the reference to work pressure. While this may have some founding, it also suggests that for many similarly general reviews, the classification is operating in overly broad strokes.

5.3 Future Work

Based on this analysis, we hypothesize that more expansive deidentification of the reviews, including the selection and removal of company products and acronyms as well as potentially industry keywords, might remove a significant amount of the signal fueling the classifier performance and help in either providing more insight into company culture or in showing that the current database of reviews does not provide sufficiently discriminative information. Another latent variable that complicates classification is the existence of diverse positions within some companies. Amazon, for example, employs many people in fulfillment centers and warehouses as well as in engineering, management, recruitment, and creative roles. Currently, the precise effect of the agglomeration of

Table 4: Most Important Features by Company
(prefixes indicate words found in review subcategories: summary, pros, and cons)

Amazon	Apple	Facebook	Google	Microsoft	Netflix
cons_shred	sum_detect	sum_concentrix	cons_philanthropi	cons_redmond	cons_overdr
cons_truck	sum_mould	cons_vamp	pros_mtv	cons_nokia	cons_dvd
sum_epitom	sum_deed	cons_detect	cons_correspond	cons_intrigu	pros_reed
pros_commerc	sum_marcom	cons_idiosyncrasi	cons_mastermind	pros_strip	cons_gato
sum_wick	cons_cupertino	cons_php	sum_adword	cons_fanat	sum_nazi
pros_wiki	pros_sap	pros_fb	sum_gsx	cons_blatantli	pros_utah
cons_frugal	cons_represent	cons_symptomat	sum_quirki	cons_plethora	cons_reed
cons_pager	pros_laser	pros_coo	pros_mk	sum_bing	sum_tsr
cons_bezo	sum_nearbi	cons_meme	pros_larri	pros_inquisit	sum_vacat
pros_dfw	pros_irrat	sum_fb	cons_65k	pros_premera	cons_readili
cons_24x7x365	pros_cupertino	cons_murder	sum_pod	sum_365	pros_agent
pros_department	cons_tim	sum_obviou	sum_evalu	cons_lion	pros_deck
cons_fabric	cons_cio	pros_intersect	pros_spare	sum_suprem	sum_websit
cons_nanni	sum_patent	pros_hustl	cons_unsupport	cons_scorecard	sum_protect
pros_downturn	pros_propag	cons_mpk	cons_funni	sum_gtsc	sum_amsterdam

all reviews regardless of role is unclear, but might be another factor contributing to the disparity in classifier performance across the companies.

Another direction of interest for future work might be variants in the combination of the numerical rating and textual data. While PCA was not found to be an effective method over concatenation of these feature sets, the slight improvement that was observed when the ratings were added raise the possibility of another proposed method in which we could train separate models for these data and then synthesize the resultant probabilities using an empirically tuned function for the final classification.

6 Conclusion

From our dive into this dataset, we conclude that Glassdoor reviews do contain valuable insights into various facets of big tech company culture and that these reviews can be evaluated with machine learning to distinguish between the companies and the experiences of their employees to some extent. In implementing classification, we found that a BOW representation of the data best captured this information, with feature reduction through PCA and SelectKBest proving ineffective. LinearSVC and Logistic Regression both performed well, though we may question whether these results could be reproduced with fully de-identified reviews given the high degree of similarity the reviews viewed as a whole. If we value a discriminative dimension in reviews, then we may need to look to revisions in our language for the best improvement in classification performance.

Another point of interest is what kind of people leave online reviews. While it is a fairly ubiquitous experience to read online reviews, according to Pew Research Center, less than 1 in 10 American always leave online reviews [7]. The low amount of online review writers suggests that those who do are probably employees who have strong sentiment about the company. The lack of representative sampling from employees represents another point data loss in the funnel from the realized company culture to what is represented through an online review system.

Zooming out, not only do our findings corroborate well-known company reputations, but they also suggest that in the context of our objective of predicting company identity from reviews, the major source of information loss may actually be in the step between the reality of employee experience and its representation through reviews rather than in the step between the reviews and machine learning analysis.

References

[1] Aayushi Agrawal. kernel2239509538. *Kaggle*, 2019. https://www.kaggle.com/aayushi2307/review-analysis/.

[2] Michal Bogacz. Which corporation is worth working for? *Kaggle*, 2019. https://www.kaggle.com/michau96/which-corporation-is-worth-working-for/.

[3] Cameron Davidson-Pilon. Least squares regression with l1 penalty. *Data Oragami*, 2012. https://dataorigami.net/blogs/napkin-folding/79033923-least-squares-regression-with-l1-penalty.

[4] Georgios Drakos. Support vector machine vs logistic regression. *Towards Data Science*, 2018. https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f.

[5] Olli Huang. Applying multinomial naive bayes to nlp problems: A practical explanation. *Medium*, 2017. https://medium.com/syncedreview/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation-4f5271768ebf.

[6] Kushal Mahindrakar. Glassdoor reviews—eda—analysis—sentiment analysis. *Kaggle*, 2019. https://www.kaggle.com/kushal1996/glassdoor-reviews-eda-analysis-sentiment-analysis.

[7] Aaron Smith Monica Anderson. Online reviews. *Pew Research Center*, 2016. https://www.pewinternet.org/2016/12/19/online-reviews/.

[8] Jon Shon Ning Luo, Yilu Zhou. Employee satisfaction and corporate performance: Mining employee reviews on glassdoor.com. *Fordham University Data Science and Business Analytics*, 2016.

[9] Yoav O. Big tech analysis. *Kaggle*, 2019. https://www.kaggle.com/yoavo1984/big-tech-analysis.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] Lalit Sachan. Logistic regression vs decision trees vs svm: Part i. *Edvancer Eduventures*, 2015. https://www.edvancer.in/logistic-regression-vs-decision-trees-vs-svm-part1/.

[12] Tavish Srivastava. Introduction to k-nearest neighbors: Simplified (with implementation in python). *AnalyticsVidhya*, 2018. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/.

[13] Peter Sunga. Google, amazon and more employee reviews, version 1, 2019. Retrieved May 8, 2019 from https://www.kaggle.com/petersunga/google-amazon-facebook-employee-reviews/kernels.

[14] vtonmail. kernel3de22037de. *Kaggle*, 2019. https://www.kaggle.com/vtonmail/pyldavis-on-employee-reviews.

[15] Kyle T. Welch Yuan Ji, Oded Rozenbaum. Corporate culture and financial reporting risk: Looking through the glassdoor. *SSRN*, 2017.

7 Appendix

Table 1: Top 10 most common words across the text review columns by company

Amazon	Apple	Facebook	Google	Microsoft	Netflix
work	work	work	work	work	work
manag	great	peopl	great	compani	peopl
good	compani	compani	compani	good	manag
compani	peopl	great	peopl	great	compani
peopl	manag	cultur	good	peopl	great
great	benefit	manag	manag	manag	get
get	good	place	place	benefit	cultur
time	get	lot	get	lot	good
lot	retail	get	lot	get	time
hour	time	good	benefit	team	job

Table 2: TFIDF accuracy scores using the LinearSVC model. TFIDF representations of text data preformed poorly compared to using a BOW representations

Dataset	LinearSVC accuracy with TFIDF	LinearSVC accuracy with BOW
Combined Current and Former Employees	0.433	0.69
Current Employees	0.436	0.7
Former Employees	0.435	0.69
Combined TFIDF representations of Pros, Cons, Summary Columns	0.576	0.708

Table 3: Accuracy scores using SelectKBest and the chi2 score function on the BOW representations of the text and the LinearSVC model. This negatively impacted the accuracy score of the models (without feature selection, we achieved accuracy of 0.71. For more detail, see Results section)

K number of features	LinearSVC accuracy with feature selection
100	0.611
200	0.639
300	0.659
400	0.673

Table 4: Accuracy scores using PCA dimension reduction on the BOW representations of the text and the LinearSVC model. This negatively impacted the accuracy score of the models (without feature selection, we achieved accuracy of 0.71. For more detail, see Results section)

Number of Dimensions	LinearSVC accuracy with PCA
100	0.615
200	0.634
400	0.657