Glassdoor Company Reviews

Will Holt and Jonathan Murphy

DATA 440: Machine Learning

Dr. Eitel Lauria

December 07, 2022

Abstract

This paper examines the relationship between Glassdoor reviews and whether or not the reviewer recommends the job/company. We used a dataset from Kaggle containing Glassdoor company reviews from the UK. We then cleaned the data and began modeling. We created models from 4 subsets of the data: categorical and numerical, text, sentiment analysis scores of the text, and categorical and numerical plus sentiment analysis of the text. We used decision tree models for all of the data subsets except the text data where we used a naive bayes bag of words model. We concluded that a random forest model trained on the categorical and numerical data plus the sentiment analysis of the text data was the best and most accurate model to predict whether a reviewer will recommend a company/job on Glassdoor.

Introduction

What are the characteristics of a good company? To answer this question we decided to take a deeper dive into company reviews by analyzing a dataset from the popular job review website Glassdoor. This dataset, covering a period between 2008 and 2021 contains various numerical, categorical, and textual features. Our main goal in this project was to use these features to accurately predict whether or not a participant would recommend a certain company. While people surely have different preferences as to what's important to them when working at a company, we were curious what those general preferences might be.

Literature Review

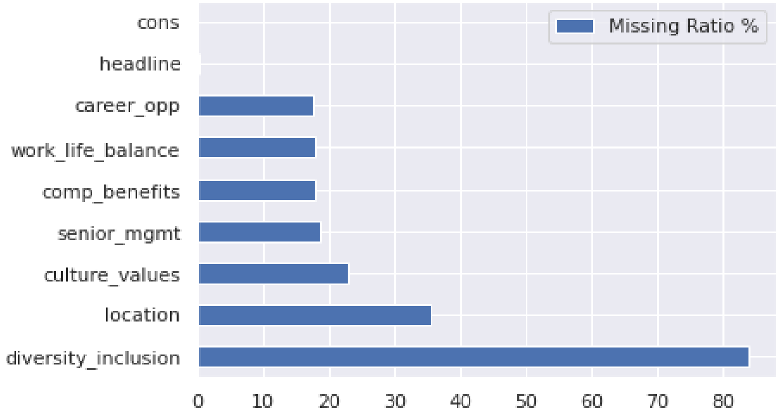
While the naive bayes classifier has certainly earned its spot as one of the top classifiers in the field, by its very nature it sacrifices accuracy for speed, based on the assumption that all predictor variables are independent. In addition, the influence of each predictor is equal within the naive bayes classifier. Because of the independence of each predictor, it's generally recommended to remove highly correlated variables from within the dataset (Gandhi). Meanwhile, models such as Random Forest, XGBoost, and VADER sentiment analysis are at the forefront of data science. " XGBoost is able to solve realworld scale problems using a minimal amount of resources" (Chen). "Random forest (RF) algorithm has been successfully applied to high-dimensional neuroimaging data for feature reduction and also has been applied to classify the clinical label of a subject using single or multi-modal neuroimaging datasets" (Dimitriadis). VADER "works best when applied to social media text, but it has also proven itself to be a great tool when analyzing the sentiment of movie reviews and opinion articles" (Calderon).

These tools do have some limitations though. For example, XGBoost algorithms “are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent” (Mavuduru). These algorithms also require large amounts of processing power to train. This makes it difficult to work with very large datasets.

Exploratory Data Analysis

A large motivation behind the dataset that we chose was the sheer size of it. In total, there are 838,566 records in this dataset with 18 features on each row. This volume of data allows us to take full advantage of our methods while maintaining the integrity of the data itself. One thing that was apparent upon initial inspection was the number of null values included in the dataset, with some features like *diversity_inclusion* having up to 85% null values (see figure 1). Because the ratio of nulls was so high on this particular feature, we decided to drop *diversity_inclusion* from the dataset entirely. Afterwards we dropped the rest of the rows with null values from our dataset. This was duly motivated by a desire for clean data, but also to shrink the dataset to make it more manageable from a training time perspective.

Figure 1



Apart from dropping null values, it also became apparent through our exploratory data analysis that our target variable was imperfect, containing a “no opinion” field that a significant proportion of the respondents filled in. There was also an imbalanced representation of our target variable overall. Since our overall goal was to predict if a participant would have a positive or negative recommendation for a particular company, we decided to drop all entries with “no opinion” checked off in order to simplify our analysis. Furthermore, in order to balance the dataset, we implemented the use of the *make_imbalance* library from *imblearn*. This allowed us to randomly choose records from the “positive” and “negative” pools in an even ratio in order to limit the false positives and negatives in our results. In order to deal with our categorical variables we decided to one hot encode them as there were only a few and it would not explode the dimensionality.

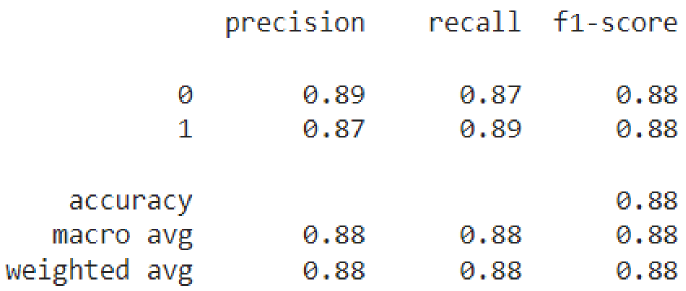
Categorical and Numerical Analysis

To begin our analysis we only considered the numerical variables of our dataset. This included not only the numerically scaled features from 0 - 5 (with 0 meaning very displeased and

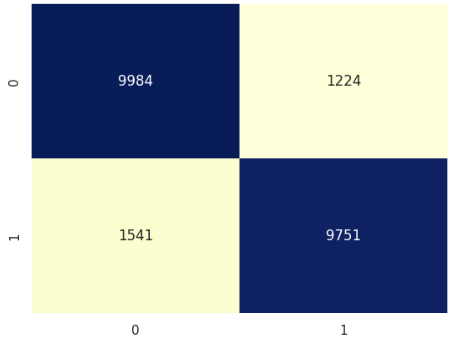
5 meaning very pleased) but also the one hot encoded values for the *ceo_approv* and *outlook* variables. It should also be noted that after simplifying the target variable, *recommend* down into the two values of would recommend and would not recommend, we binary encoded these values into a new variable *bin_recommend* with 1 taking the place of a positive recommendation and 0 taking the place of a negative recommendation. We then took our numerical features *overall_rating, work_life_balance, culture_values, career_opp, comp_benefits, senior_mgmt, ceo_approv,* and *outlook* with our target variable *bin_recommend* and partitioned them into a training and testing set, keeping the default ratio of 75% training and 25% testing.

For models, we chose to keep things relatively simple at first, using the random forest classifier to create a decision tree model. We tried three different variations of the classifier, using a random walk, a grid search, and a bayes search. These all provided virtually identical results with variations easily explained by the randomized partitioning for each method. We also used XGBoost as a part of this analysis with a random walk, however the results of this were also virtually identical to the previous three models. With this in mind, the results of our best performing model the Grid Search were as follows with a strong 88% accuracy. As mentioned though, this best performer would often change depending on the randomization.

Figure 2



```
              precision    recall  f1-score

           0       0.89      0.87      0.88
           1       0.87      0.89      0.88

    accuracy                           0.88
   macro avg       0.88      0.88      0.88
weighted avg       0.88      0.88      0.88
```

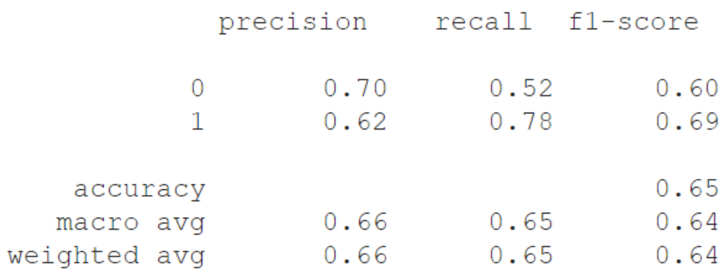


Apart from the relatively good performance, the only other thing to note is the even distribution across the confusion matrix, with false positives and false negatives nearly matching each other, and the same for the true positives and true negatives. We believe this is in large part due to the balancing of the dataset before the analysis.

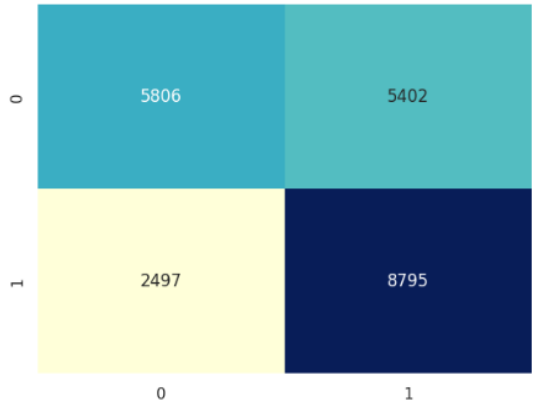
Bag of Words Analysis

To begin our textual analysis we started off with a straightforward Bag of Words approach, using the *headline, pros,* and *cons,* features from the original dataset. Luckily there were no null values in these fields so we were able to skip further data manipulation and move right into the model. For this approach, we opted for the multinomial naive bayes model, in large part due to its popularity and simplicity. While we could have branched out here and tried a few other models for Bag of Words, after looking at the initial results from this model, it was clear that we would have to try a different approach. Below are the performance metrics for this model.

Figure 3



```
              precision    recall  f1-score

           0       0.70      0.52      0.60
           1       0.62      0.78      0.69

    accuracy                           0.65
   macro avg       0.66      0.65      0.64
weighted avg       0.66      0.65      0.64
```

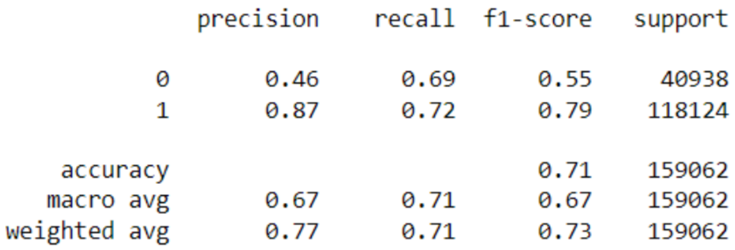


As we can see there was a much lower accuracy of only 65% on this model and a large number of false positives compared to the previous numerical analysis. This was a bit unexpected, since this dataset was also balanced like the previous one.

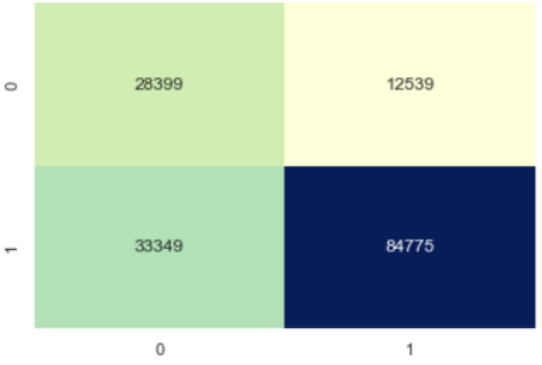
Sentiment Analysis

For sentiment analysis, we used the VADER sentiment analyzer from NLTK. VADER is a lexicon and rule-based sentiment analysis tool that was trained on social media data. This is especially helpful because glassdoor reviews often use slang and imperfect grammar similar to that of social media. From VADER the sentiment intensity analyzer was used to create a compound polarity score from -1 to 1 for every text value for the 3 text variables: headline, pros, and cons. -1 means that the sentiment was extremely negative and 1 means that the sentiment was very positive. A random forest decision tree model tuned with random grid search was then trained on the 3 sentiment variables. Below are the performance metrics from this model.

Figure 4



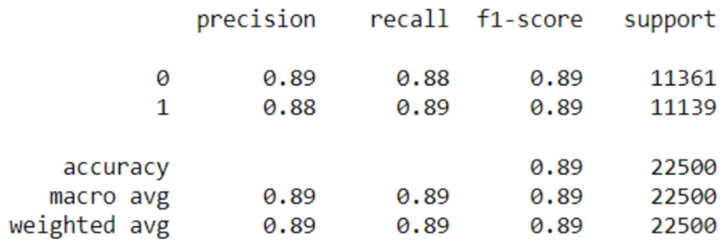
```
              precision    recall  f1-score   support

           0       0.46      0.69      0.55     40938
           1       0.87      0.72      0.79    118124

    accuracy                           0.71    159062
   macro avg       0.67      0.71      0.67    159062
weighted avg       0.77      0.71      0.73    159062
```



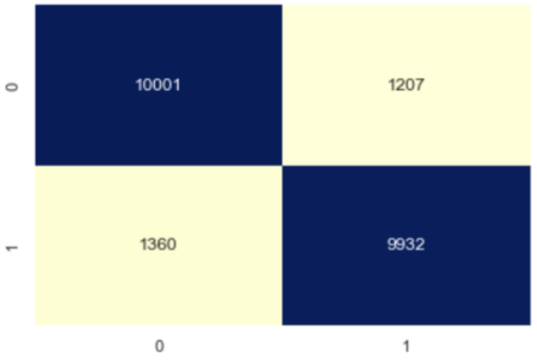
Numerical and Categorical Plus Sentiment Analysis

For this analysis we combined the original numerical and categorical variables with the calculated sentiment analysis variables. This model used by far the most amount of data than all of the other models, so we expected this model to perform the best out of all of them. A random forest tuned with a random grid search was used to create the analysis model. The naive bayes and grid search tuning methods weren’t used as they provided nearly identical results to the random search with much slower training times. XGBoost wasn’t used for the same reasons. Below are the performance metrics from this model.

Figure 5



```
              precision    recall  f1-score   support

           0       0.89      0.88      0.89     11361
           1       0.88      0.89      0.89     11139

    accuracy                           0.89     22500
   macro avg       0.89      0.89      0.89     22500
weighted avg       0.89      0.89      0.89     22500
```



Conclusion

In conclusion, The best performing model was the model trained on the categorical and numerical data plus the calculated sentiment analysis data. This model had an accuracy of 89%. This model was closely followed by the one trained on just the categorical and numerical variables which had an accuracy of 88%. The models trained on just the text data were significantly worse with the model trained on the sentiment analysis polarity scores having an accuracy of 71% and the naive bayes bag of words model having an accuracy of 65%. The numerical and categorical variables proved to be very strong predictive variables with the sentiment analysis data adding a little bit extra predictive potential.

Next Steps

Some next steps would include performing regressions to determine which variables are the strongest predictors in determining if someone is going to recommend a company or not. More data could also be gathered from outside the UK. This data could then be added into the existing dataset or compared against it to find work culture differences between countries. Different modeling techniques could also be used such as neural networks and BERT to possibly lead to different conclusions. Another next step could include training a sentiment analyzer model specifically on Glassdoor review data to create a model more tuned to that specific data. This could provide more accurate polarity scores on the text data and also show which words/phrases are most commonly associated with positive or negative company reviews.

Works Cited

Dg. “Glassdoor Job Reviews.” *Kaggle*, 5 Nov. 2022, https://www.kaggle.com/datasets/davidgauthier/glassdoor-job-reviews.

Calderon, Pio. “Vader Sentiment Analysis Explained.” Medium, Medium, 31 Mar. 2018, https://medium.com/@piocalderon/vader-sentiment-analysis-explained-f1c4f9101cd9.

Chen, Tianqi, and Carlos Guestrin. “XGBoost.” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, https://doi.org/10.1145/2939672.2939785.

Dimitriadis, StavrosI, et al. “How Random Is the Random Forest? Random Forest Algorithm on the Service of Structural Imaging Biomarkers for Alzheimer's Disease: From Alzheimer's Disease Neuroimaging Initiative (ADNI) Database.” Neural Regeneration Research, vol. 13, no. 6, 2018, p. 962., https://doi.org/10.4103/1673-5374.233433.

Gandhi, Rohith. “Naive Bayes Classifier.” *Medium*, Towards Data Science, 17 May 2018, https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c.

Mavuduru, Amol. “Why XGBoost Can't Solve All Your Problems.” Medium, Towards Data Science, 10 Nov. 2020, https://towardsdatascience.com/why-xgboost-cant-solve-all-your-problems-b5003a62d12a.