**Final Report\_IS-733**

**Title:** Modeling Quality of Suggestions

**Abstract:** The project focuses on optimizing the review process of employee suggestions in a large human resource company's online forum. The primary challenge was to create a sustainable and scalable method to filter out less impactful suggestions, a task previously performed manually. The major findings that informed our approach include anomalies in the data set, such as instances where votes (both up and down) exceeded the number of views, and cases where the number of responses surpassed the views. Additionally, rows with zero views were also identified. These discrepancies highlighted the need for a more robust and automated analysis system. Leveraging these insights, we developed a machine learning algorithm capable of identifying and prioritizing suggestions based on engagement metrics and content relevance. This method significantly reduces manual sorting and ensures a more accurate and efficient evaluation of employee feedback. The results demonstrate the project's success in enhancing the company's ability to identify and act on valuable suggestions, thereby fostering a more responsive and employee-centric corporate culture.

**Background/motivation:**

EDA (Exploratory Data Analysis):

**About The Dataset:** The dataset is obtained from an online forum of a human resources company.

This forum facilitated the submission of suggestions by employees to upper management, with each unique suggestion corresponding to an individual thread identified by a distinct attribute known as the suggestion ID. At its inception, the dataset comprised 16,429 instances, featuring 10 attributes, each with its own descriptive information.

* **Recommended:** Indicates whether the suggestion was recommended (0 for not recommended, 1 for recommended).
* **Suggestion\_Id:** The unique identifier for each suggestion.
* **Responses:** The number of responses to the suggestion.
* **Views:** The number of views the suggestion received.
* **Votes\_Up:** The number of upvotes for the suggestion.
* **Votes\_Down:** The number of downvotes for the suggestion.
* **Author\_Id:** The unique identifier for the author of the suggestion.
* **Author\_Join:** The number of days since the author joined the platform.
* **Author\_TotalPosts:** The total number of posts made by the author.
* **Author\_PostsPerDay:** The average number of posts made per day by the author.

**Data cleaning:**

The dataset contains no null values as well as missing values

But we have observed that some attributes like Suggestion\_ID and Author \_ID are identified as integer type, so we applied the type conversion and changed them to object as they need to be constant as well as to restrict them from any future mathematical operations.

**Duplicate Values:**

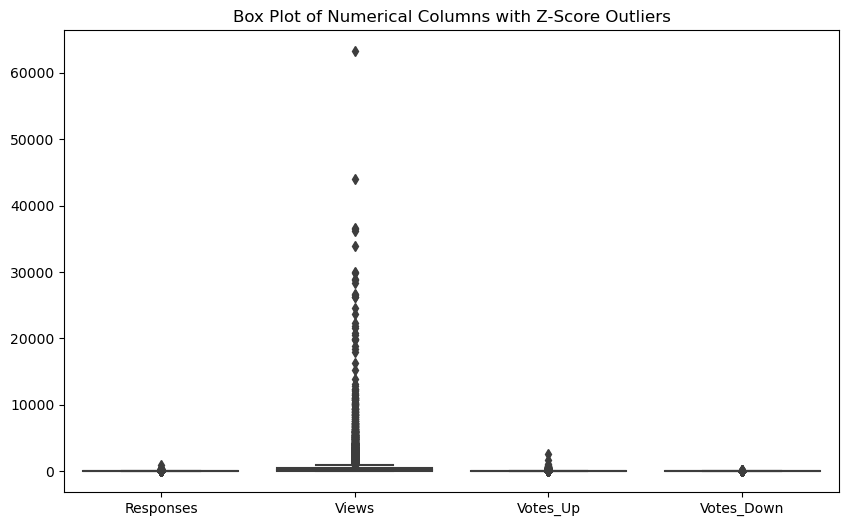
Furtherly, the duplicates from the Suggestion\_ID were eliminated and this particular feature made it an unique attribute. A few duplicate values were identified and deleted while retaining the first occurrence of those instances.

**Outliers:**

Initially for the cleaned dataset, we have applied the following two methods to identify outliers. The methods are as follows:

1. Z-Score Method
2. InterQuartile Range (IQR)

We have used the Z-Score Method in order to identify the outliers and removed it because the IQR method yielded us with a higher number of outliers when compared to that of the Z-Score Method.



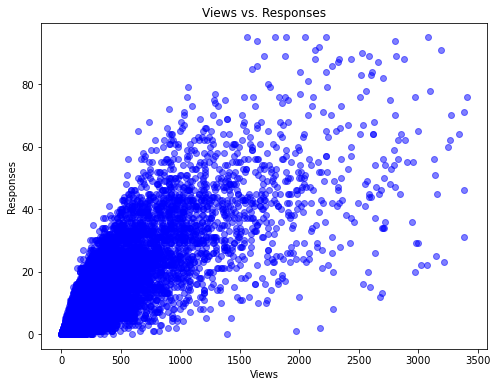
These are the counts of the outliers in the following numerical columns after applying the Z-Score Method.   
**{'Responses': 262, 'Views': 164, 'Votes\_Up': 197, 'Votes\_Down': 202}**

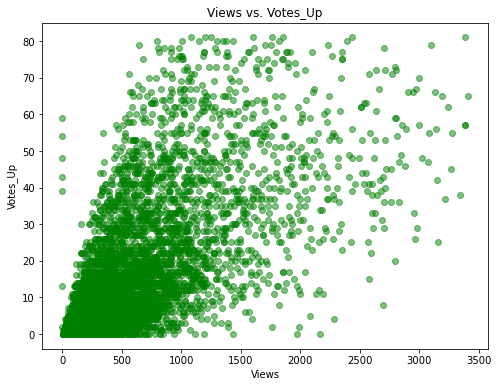
Original row count: 14563

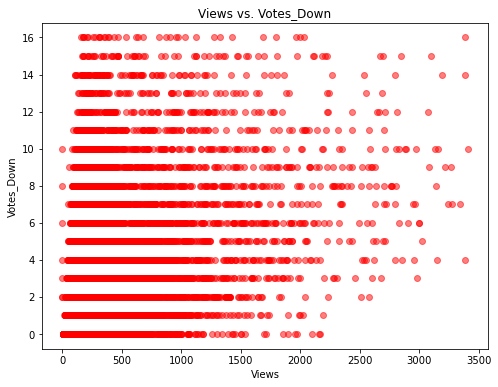
New row count after removing outliers: 13545

Percentage decrease in data: 6.990317928998147

The following scatter plots represent the Views Vs Responses, Views Vs Votes\_up and Views Vs Votes\_Down after removing the outliers from the dataset.







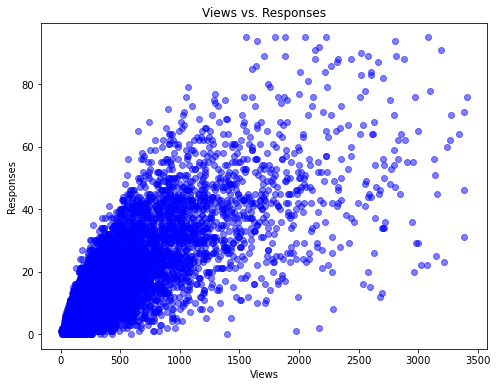
We can see from the above visualizations that when the Views are zero, Votes\_up and Votes\_down are recorded, and ultimately, these results provide us with inaccurate suggestions.

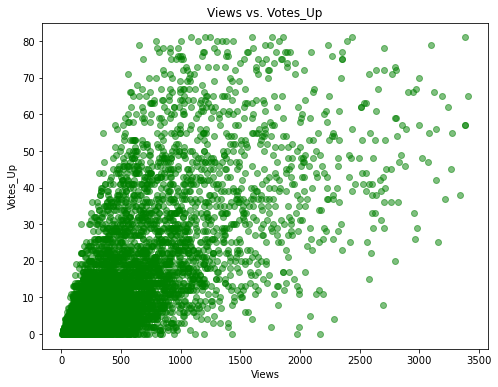
Later on, as the final step of the data cleaning, the following conditions were applied on the dataset in order to eliminate the rows which yielded the incorrect recommendation.

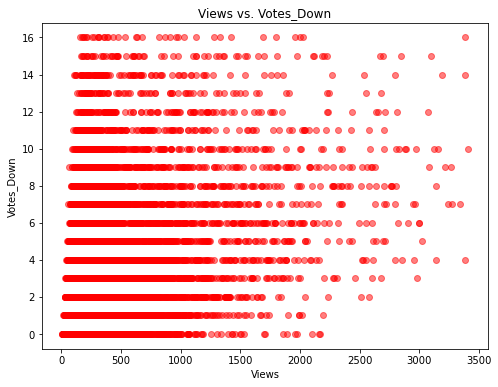
The conditions of it are as follows:

* The rows with **votes\_up > Views.**
* The rows with **votes\_down > Views.**
* The rows where the **sum of votes\_up and votes\_down are greater than the number of views**
* The rows where the **responses are more than views.**
* The rows where the **views are equal to 0.**

The graphs have been plotted after applying the conditions, which are as follows:

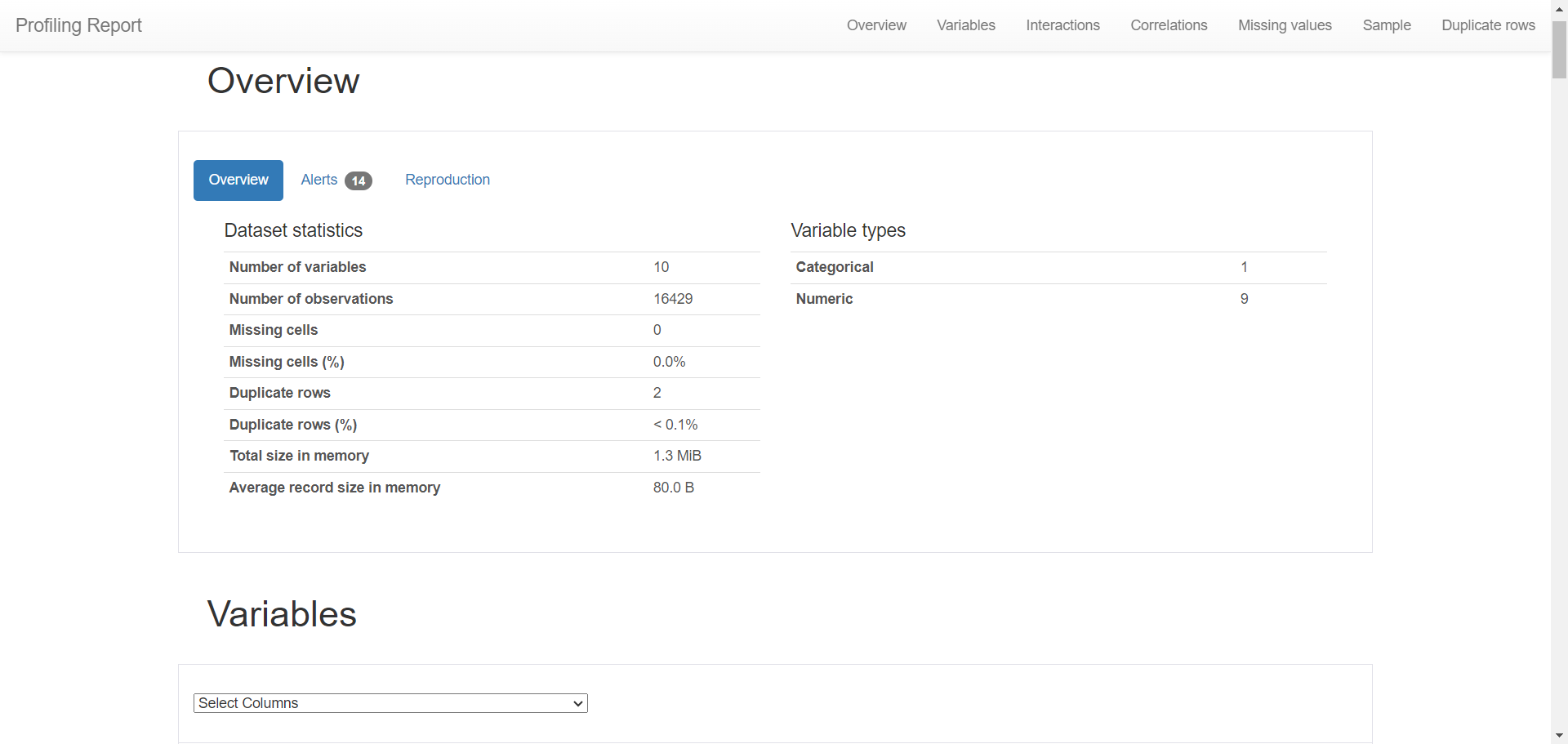


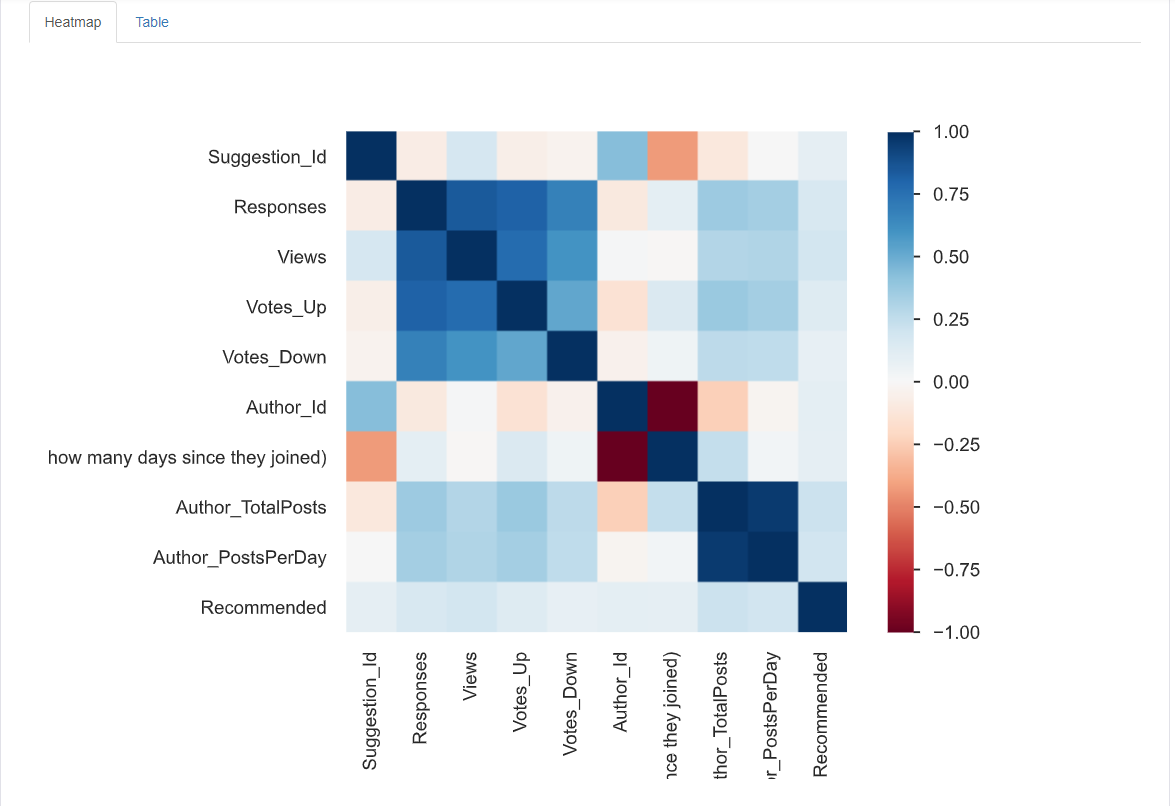


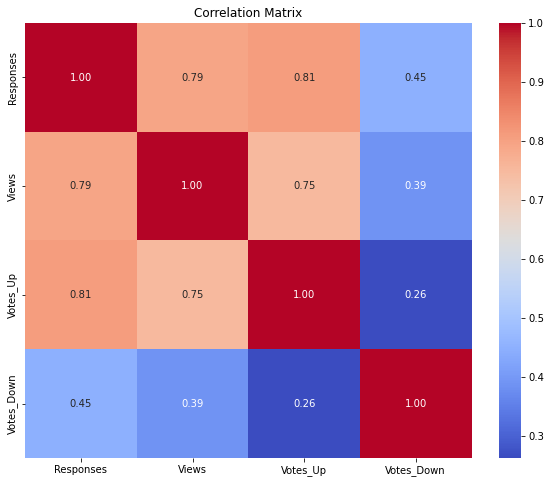


We can observe that when the views are zero, the other observations have been eliminated.

**Data Profiling :**





**Analyzing relationships :** 

Responses have a strong positive correlation with Views (0.79) and Votes\_Up (0.81), indicating that as the number of responses increases, the views and upvotes tend to increase as well. There is a moderate positive correlation with Votes\_Down (0.45), suggesting that suggestions with more responses may also receive more downvotes, albeit to a lesser extent.

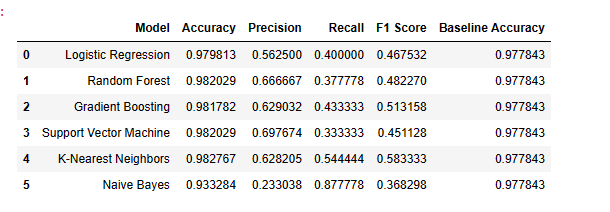
Views are strongly positively correlated with Votes\_Up (0.75), meaning suggestions with more views tend to have more upvotes. There is a moderate positive correlation with Votes\_Down (0.39), which suggests that more viewed suggestions may also get more downvotes, though this relationship is less strong.

Votes\_Up has a weak positive correlation with Votes\_Down (0.26), indicating that suggestions with more upvotes might get slightly more downvotes, but the relationship is not very strong.

The values on the diagonal are all 1, as they represent the correlation of each variable with itself. Overall, the strongest correlations are between Responses and Votes\_Up, and between Views and Votes\_Up, indicating that engagement in terms of responses and views is a good predictor of upvotes.

**Model Development:**

The machine learning methods have been applied on the latest dataset, and the performance metric has been obtained.



Logistic Regression performed with high accuracy (approximately 97.98%) but had relatively low precision and recall, leading to a modest F1 score of about 0.467.

Random Forest and Support Vector Machine (SVM) both had identical accuracies (around 98.20%) with Random Forest having slightly better precision and SVM having better recall.

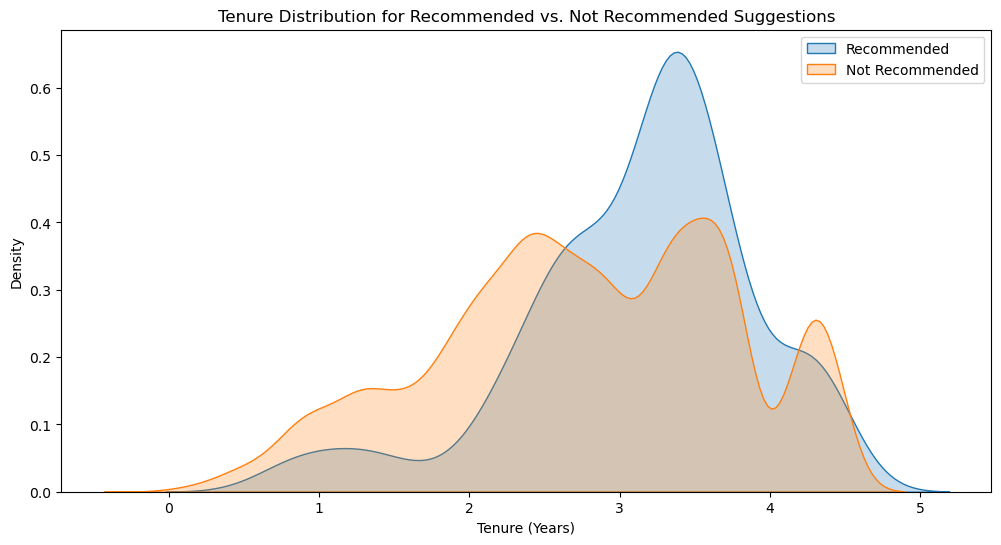
Gradient Boosting showed similar accuracy to Random Forest and SVM, with a slightly better precision and F1 score (approximately 0.629 and 0.513 respectively).

K-Nearest Neighbors had comparable accuracy (approximately 98.27%) and the highest F1 score among all models (about 0.583), suggesting a more balanced performance in terms of precision and recall.

Naive Bayes had the lowest precision (approximately 0.233), but the highest recall (about 0.877), leading to the lowest F1 score (approximately 0.369).

All models had a high baseline accuracy of approximately 97.78%, indicating a potentially skewed dataset or a high prevalence of the majority class. In such cases, accuracy might not be the best performance metric, and one may need to consider the F1 score, precision, and recall more closely, especially in imbalanced datasets.

**Results and Flow charts:**



* Conclusion
  + It should flow naturally from the result

**Future work:**

* Can the same data be used to rank employees based on their demonstrated ability to make predominantly good suggestions? Can it be used to identify groups of employees whose suggestions could be aggregated to provide more reliable suggestions than made by the best individuals?
* Making recommendations to the IT department about better ways they could collect this data in the future. What other attributes would prove useful and why? Would it be possible to build a completely automated suggestion ranking system?

**Team reflection (if applicable):**

* + What you learned from this project, what lesson learned, what could be improved etc.

**Reference (if any):**

Chat GPT(Code and other purpose)

Quill Bot(Paraphrasing)