User Feedback Classification Model

By Sheila Torrico

DISCORD

- Understands the value that an engaged user community brings to its products. It created public Feedback Forums
- Users really embraced them and are constantly providing feedback/suggestions/requests around the clock
- Product management is looking to take advantage of all these ideas that covers from text chat to server management
- Customer Success is the conduit to deliver the users ideas to Product management

What is the problem?

While looking to comply with the request of Product Management, Customer Success team noticed that not all the submitted feedback are labeled correctly. Leading to ignore relevant insights that might contain the next big feature idea

What is the solution?

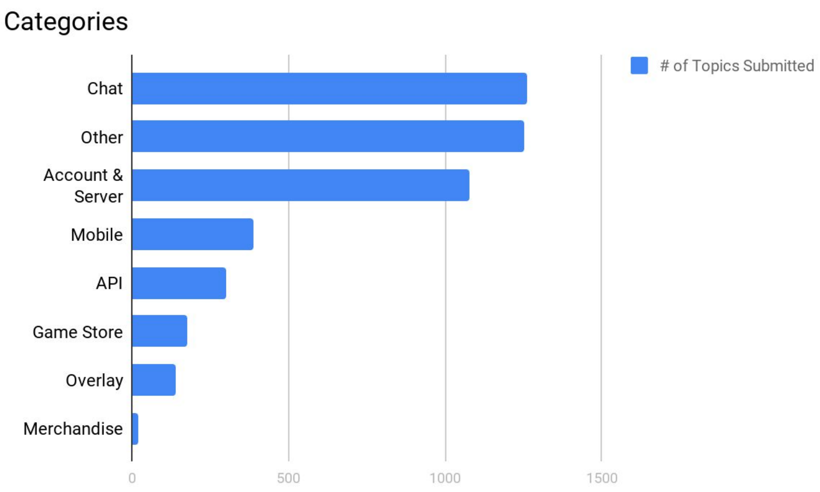
Build a Classification model that will classify any submitted topics to their proper category

Solution Approach

- Phase 1: Perform Exploratory Data Analysis
- Phase 2: Evaluate Results
- Phase 3: Call to action

Phase 1: Exploratory Data Analysis

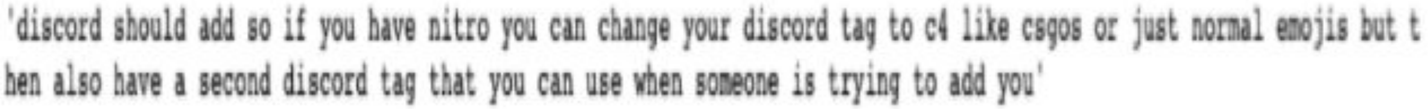
- Dataset was scrapped from Discord's Feedback site. It contained over 4.5K observations
- The label to predict is Categories and has 8 options
    - Chat, Other and Account Server Management were categories with most submitted topics
- The variables used to predict Categories are Votes and Submission Title



Phase 1: Exploratory Data Analysis

- Results from statistical analysis to numerical variable, Votes showed that:
  - Observations are truly disperse due to the high standard deviation.
  - Min and Max values range from a negative value to a very large positive numbers

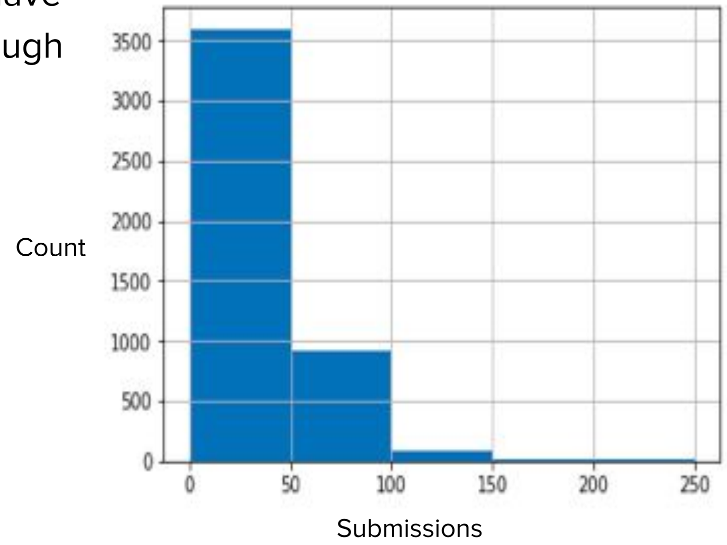
- Key takeaway: Voting indicates that Discord users are showing their agreement or disagreement on certain submissions, which is good way to gauge engagement. I.e., below suggestion obtained the lowest number of votes with -53.



```
'discord should add so if you have nitro you can change your discord tag to c4 like csgos or just normal emojis but t
hen also have a second discord tag that you can use when someone is trying to add you'
```

Phase 1: Exploratory Data Analysis

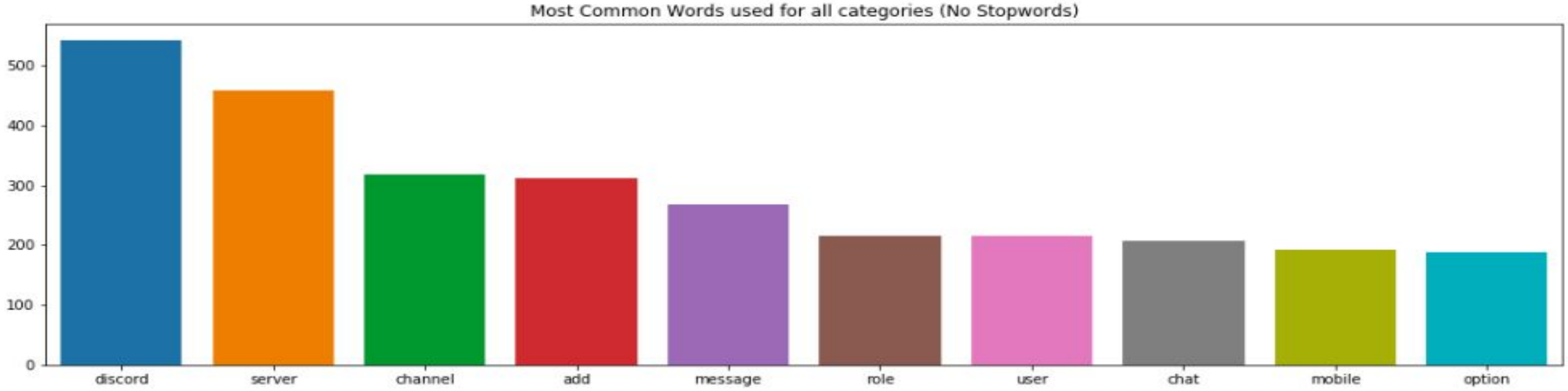
- Histogram shows the majority of topic submissions have a length of 50 characters. Is the average length enough for the model to learn?
- The max length from existing submission is 250.



Phase 1: Exploratory Data Analysis

Further analysis on the text from the submitted topics showed:

- “Discord” claimed the top spot as the most used word within a submitted topic.
- Other words to pay attention to were: “add”, “option”, “chat”. Are users suggesting to add an option that could be included in the next release of Discord’s chat features?



Phase 2: Model Building/ Results Evaluation

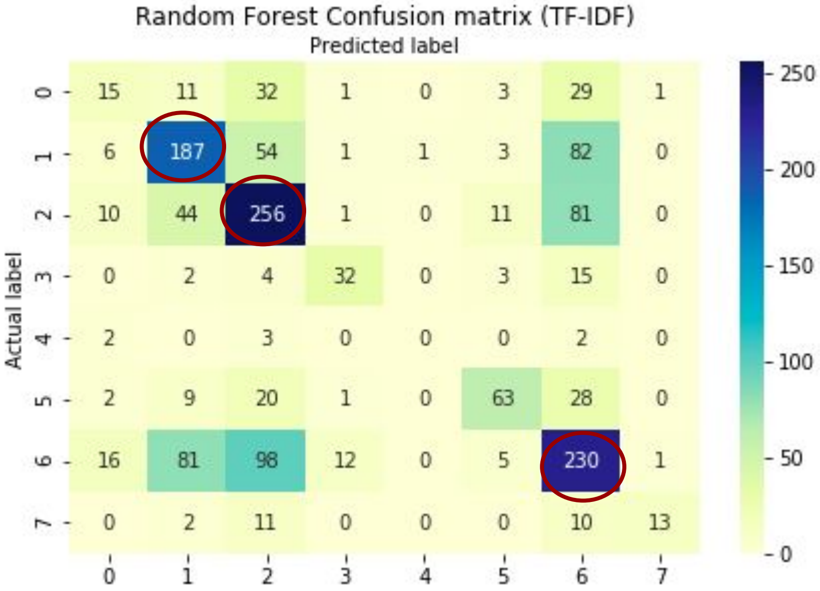
53%

- F1- Score of the best performing Classification model: Random Forest

Phase 2: Model Building/ Results Evaluation

Random Forest Model did a great job:

1. Classifying Topics with low number of observations such as API, Game Store, Mobile and Overlay. Except for Merchandise
2. Returning most of the relevant results for Chat and Account and Server Management
3. Classifying Topics that are truly “Other” and suggesting potential categories for the non-Others
4. Identifying the important words that helped predict the category a topic should be classified under



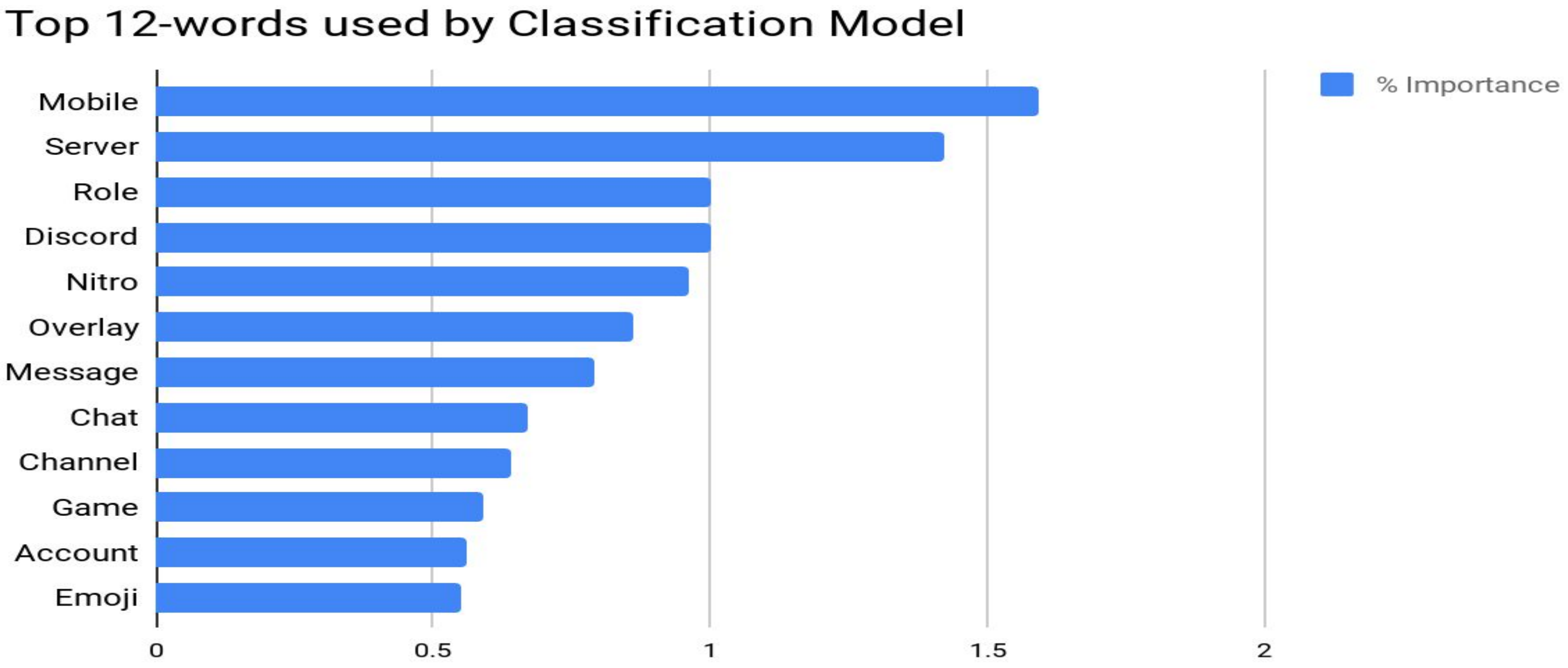
Random Forest Confusion matrix (TF-IDF)

Actual label \ Predicted label	0	1	2	3	4	5	6	7
0	15	11	32	1	0	3	29	1
1	6	187	54	1	1	3	82	0
2	10	44	256	1	0	11	81	0
3	0	2	4	32	0	3	15	0
4	2	0	3	0	0	0	2	0
5	2	9	20	1	0	63	28	0
6	16	81	98	12	0	5	230	1
7	0	2	11	0	0	0	10	13

Category 0: API | 1: Account & Server Management | 2: Chat |

3: Game Store | 4: Merchandise | 5: Mobile |6: Other | 7: Overlay

Phase 2: Results Evaluation



Phase 3: Results/Call to Action

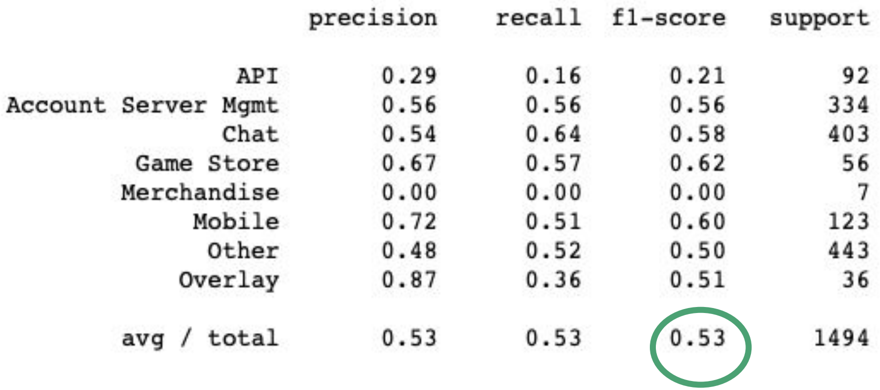
- **Customer Success now has a model that is able to automate the Classification process and filter records to their corresponding categories 50% of the time!**

Call to actions:

  - Evaluate performance of the model, particularly keeping an eye on the Recall score.
  - Compare the words identified as top contributors by the model against existing tag list used to filter/route submissions and decide if any of them should be included.

Appendix: Model Building/ Results Evaluation

- Built and tested 3 different models over multiple scenarios: Logistic Regression, Gradient Boosting and Random Forest
- Random Forest (TF-iDF with stopwords removed)) was the best performing model with a F1 score of .53 by looking at its Classification report



```
                     precision    recall  f1-score   support

                API       0.29      0.16      0.21        92
Account Server Mgmt       0.56      0.56      0.56       334
               Chat       0.54      0.64      0.58       403
         Game Store       0.67      0.57      0.62        56
        Merchandise       0.00      0.00      0.00         7
             Mobile       0.72      0.51      0.60       123
              Other       0.48      0.52      0.50       443
            Overlay       0.87      0.36      0.51        36

        avg / total       0.53      0.53      0.53      1494
```