
Natural Language Processing

SUICIDAL THREAD DETECTION FOR REDDIT

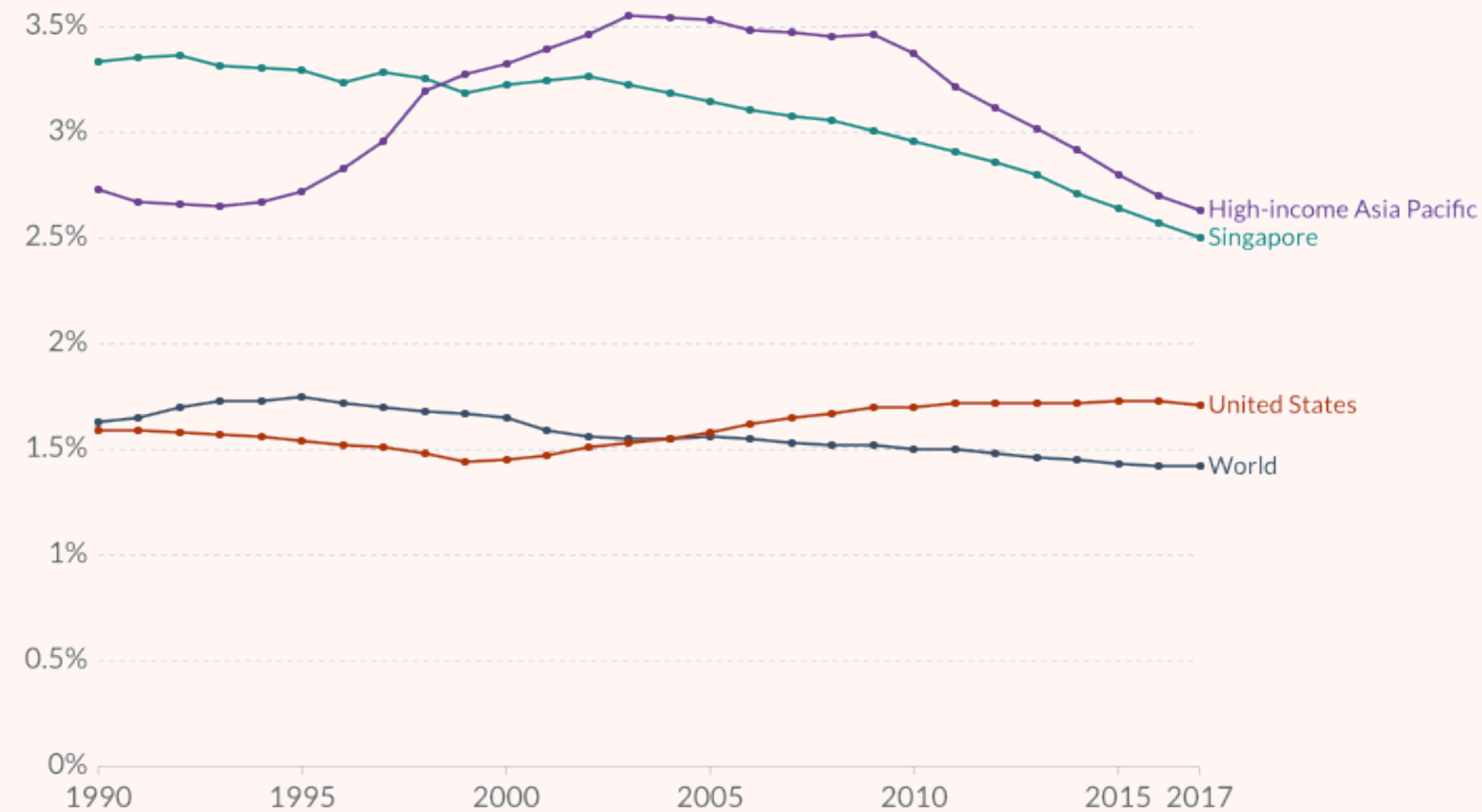
PROJECT BACKGROUND

“Despite progress in the international effort to curb suicidal activity, one person still dies every 40 seconds from suicide.”

**- Dr. Tedros Adhanom Ghebreyesus
Secretary General of WHO**



Share of deaths from suicide, 1990 to 2017



Source: IHME, Global Burden of Disease

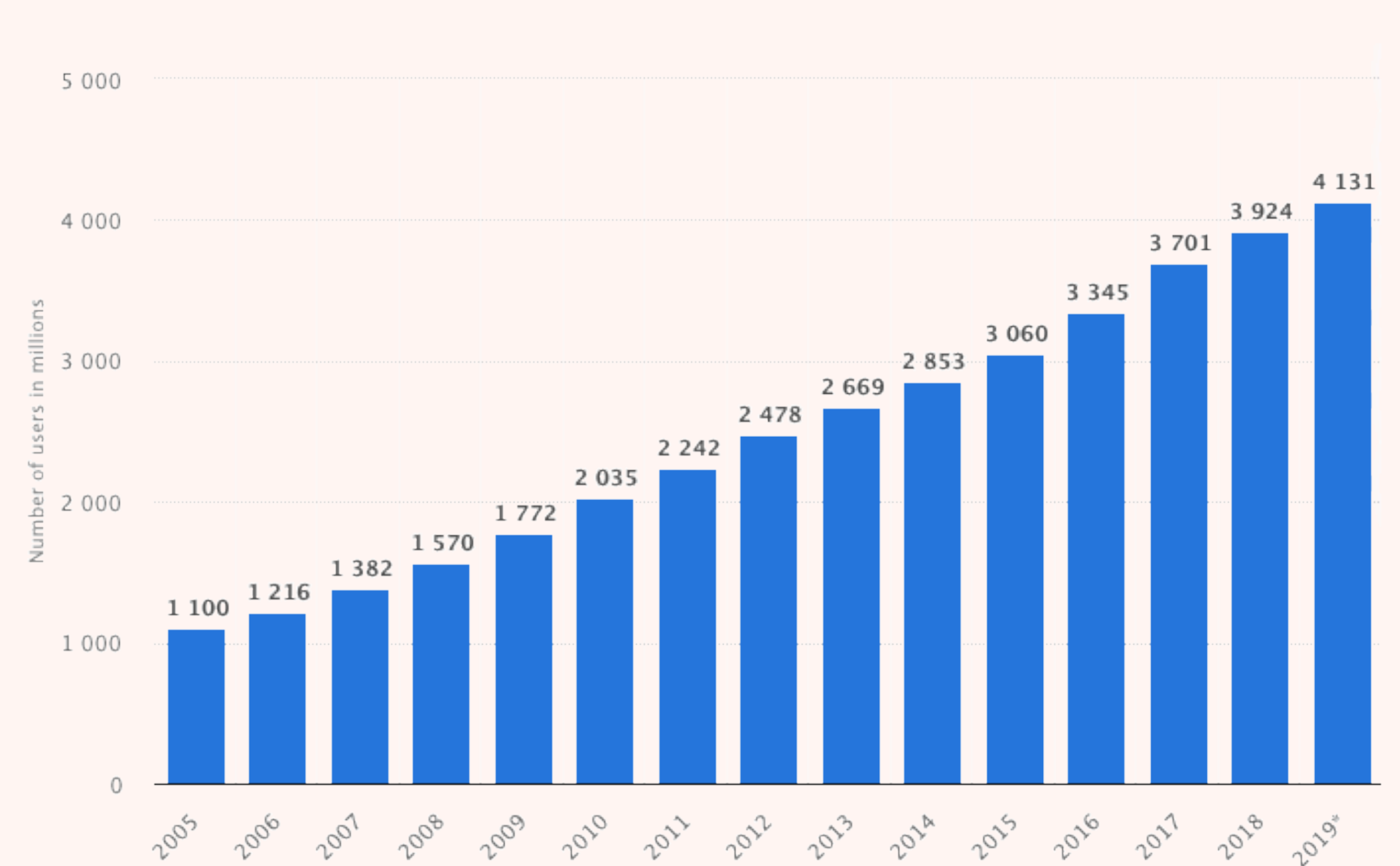
SOCIO-ECONOMIC BURDEN

Suicide rate is often highest in high-income countries, and it is the second leading cause of death among young people.

RIDE ON THE TRAIN OF INTERNET

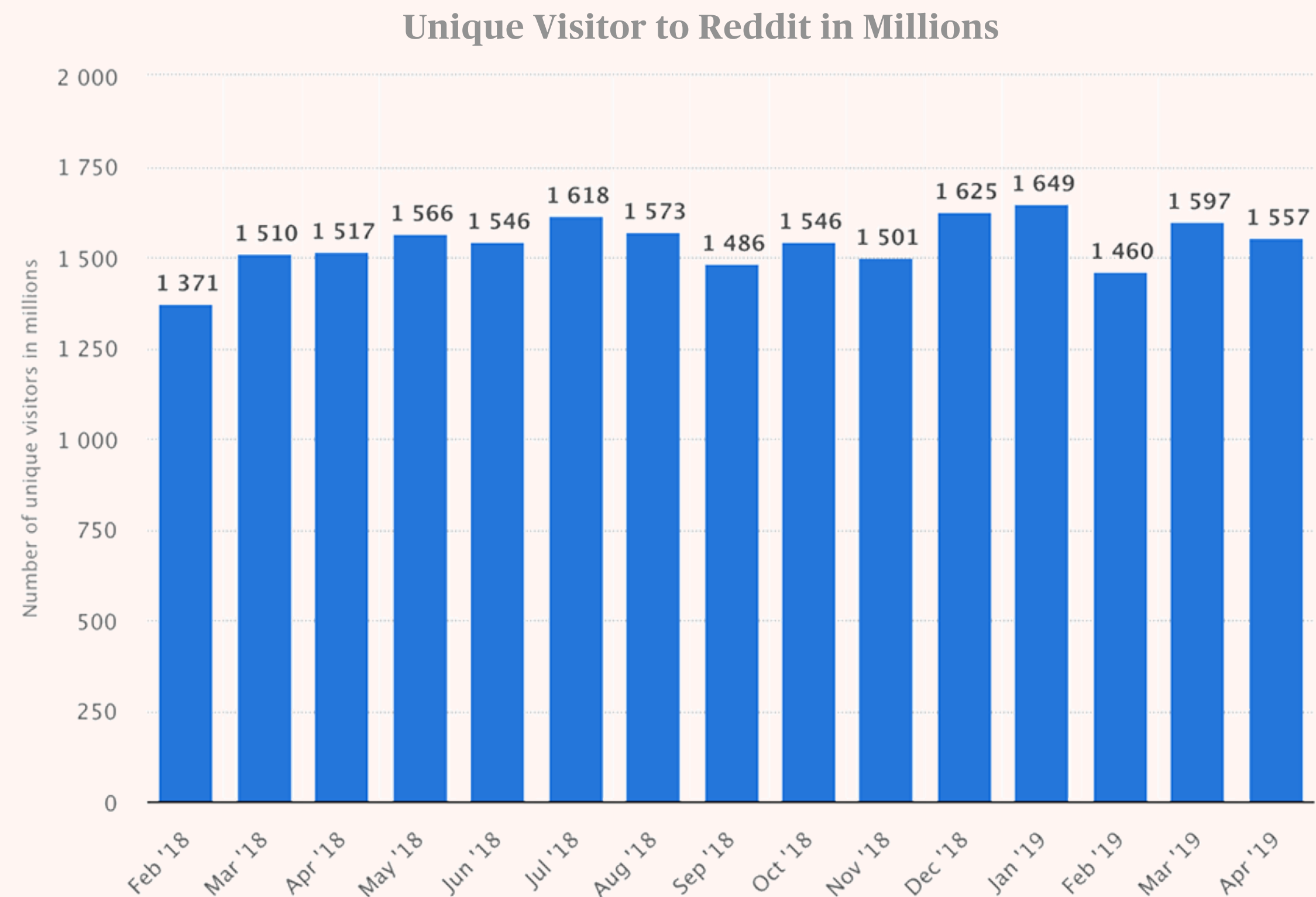
**More people are getting online every year.
More people are sharing their stories online.
It would be imperative to use internet as
vehicle to level up the effort to battle
suicide cases.**

Number of Internet User Worldwide, 2005 - 2019



Source: Statista, 2019

REDDIT



Source: Statista, 2019

Reddit has a substantial amount of subscribers that used it as a platform to ask for opinion or share an opinion.

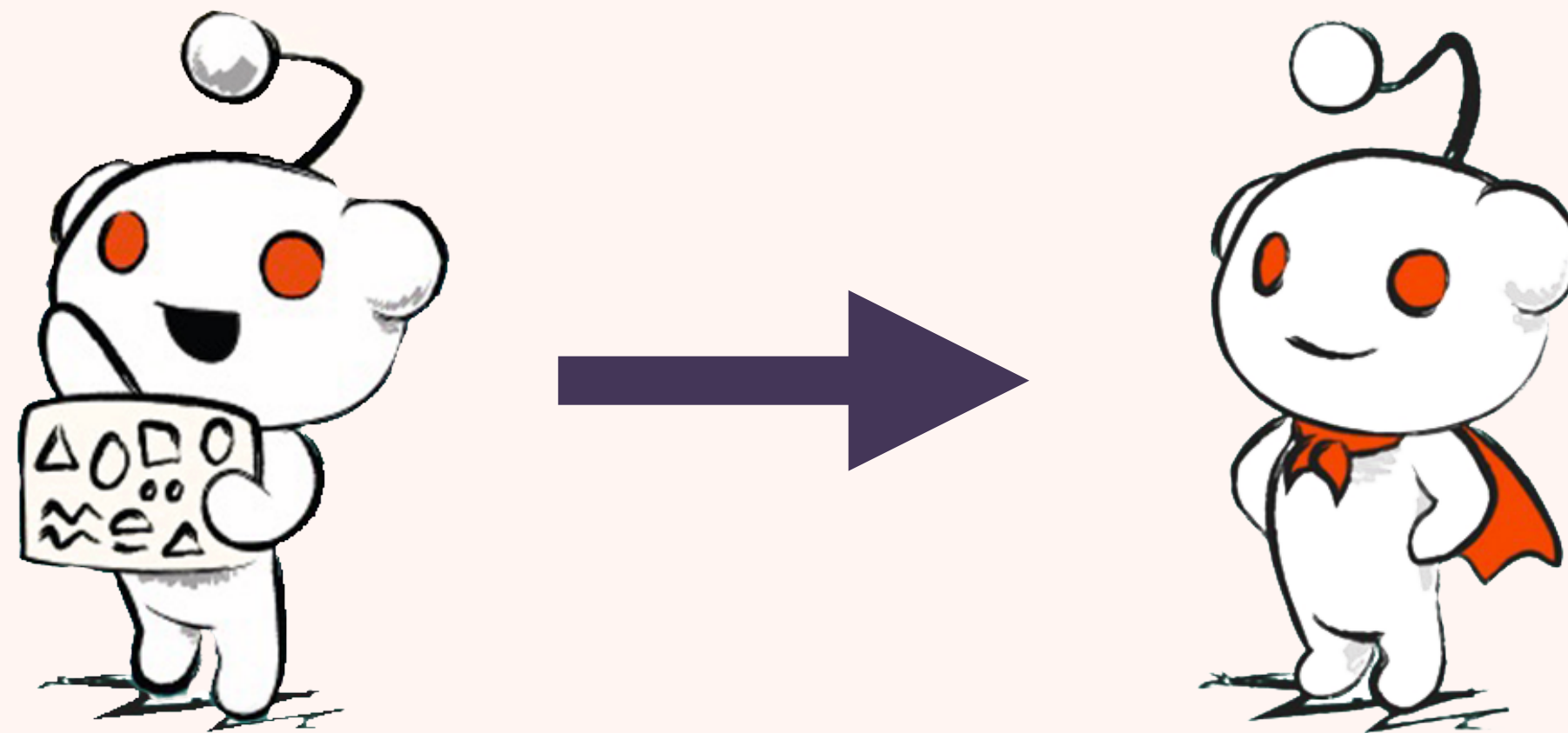
It is made up subcategories called subreddit. Each has its own community for a specific purpose or interest.

And there is a suicide watch subreddit to provide support for user with suicidal thought.

PROBLEM STATEMENT

Sometimes thread from potential suicidal victim might be tagged wrongly and fall into the other subreddit.

A machine learning model might be able to capture any thread with suicidal signs that ended up in the wrong place and transferring them under the radar and care of suicide watcher.



CLASS 0 - GENERAL ADVICE

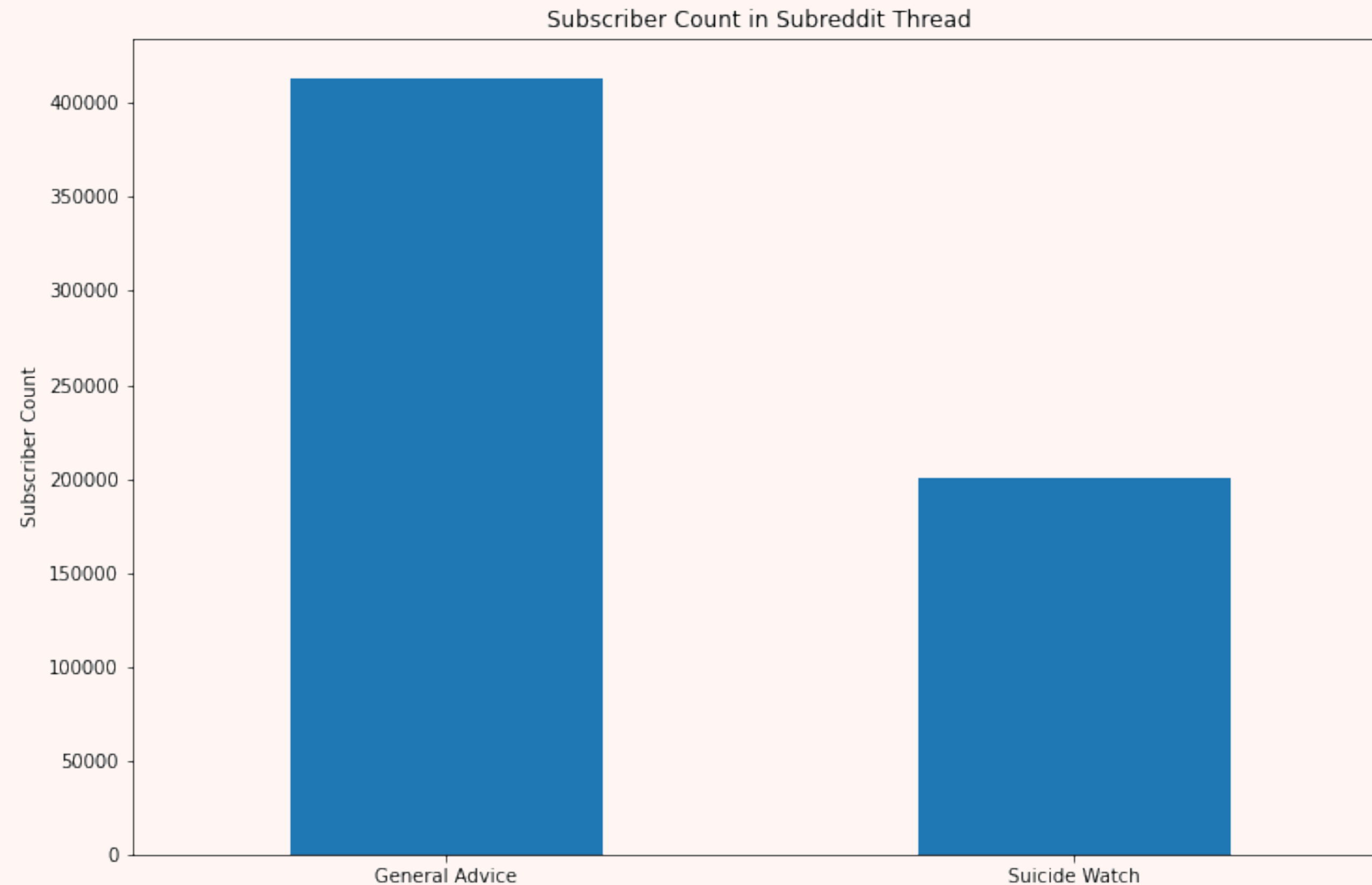
[r/Advice](#)

CLASS 1 - SUICIDE WATCH

[r/SuicideWatch](#)

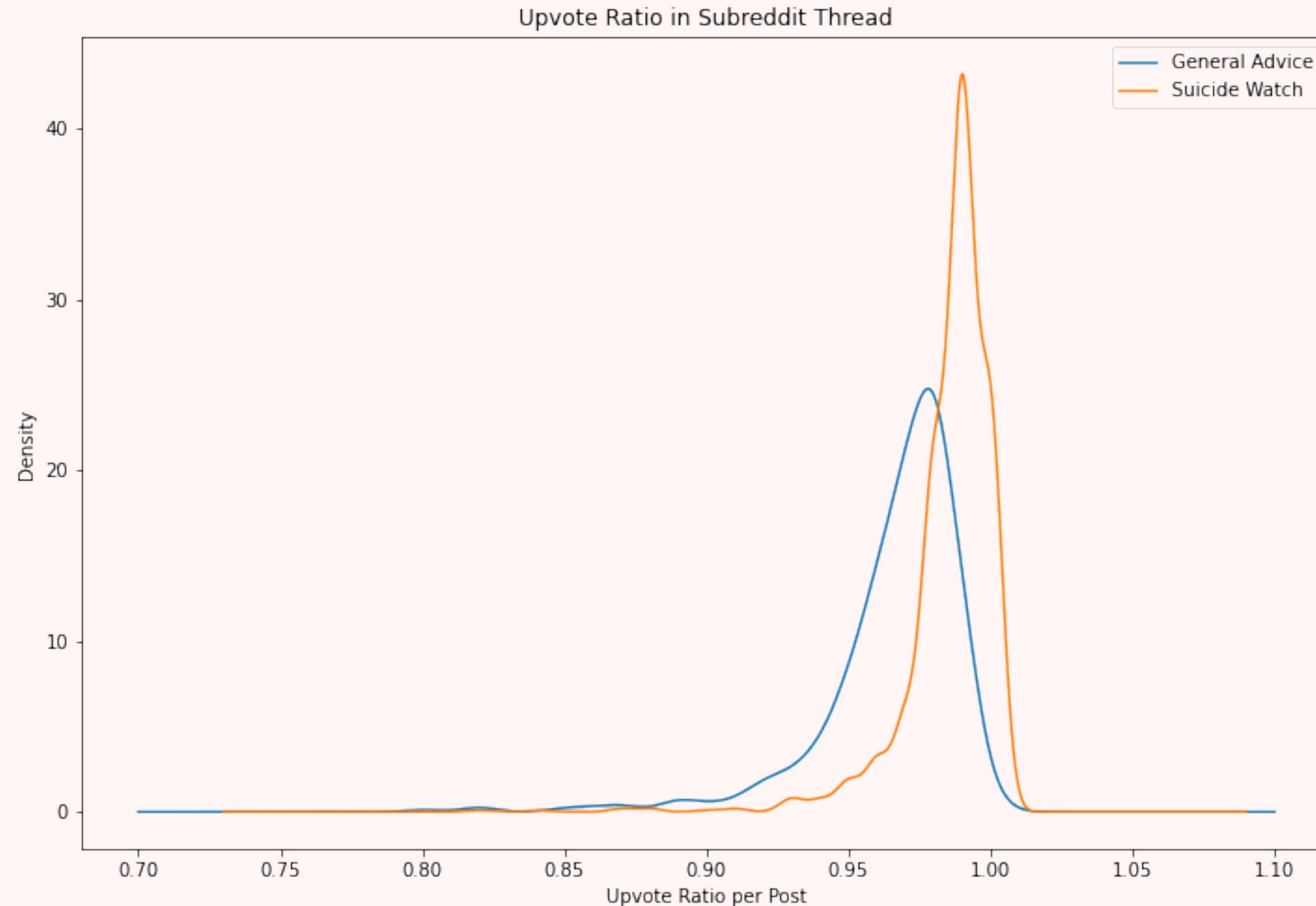


HEAD-TO-HEAD



SUBSCRIBER COUNT

Suicide Watch subreddit has only about half of General Advice. However, a community made up of 200,000 users is still considered as a formidable force.



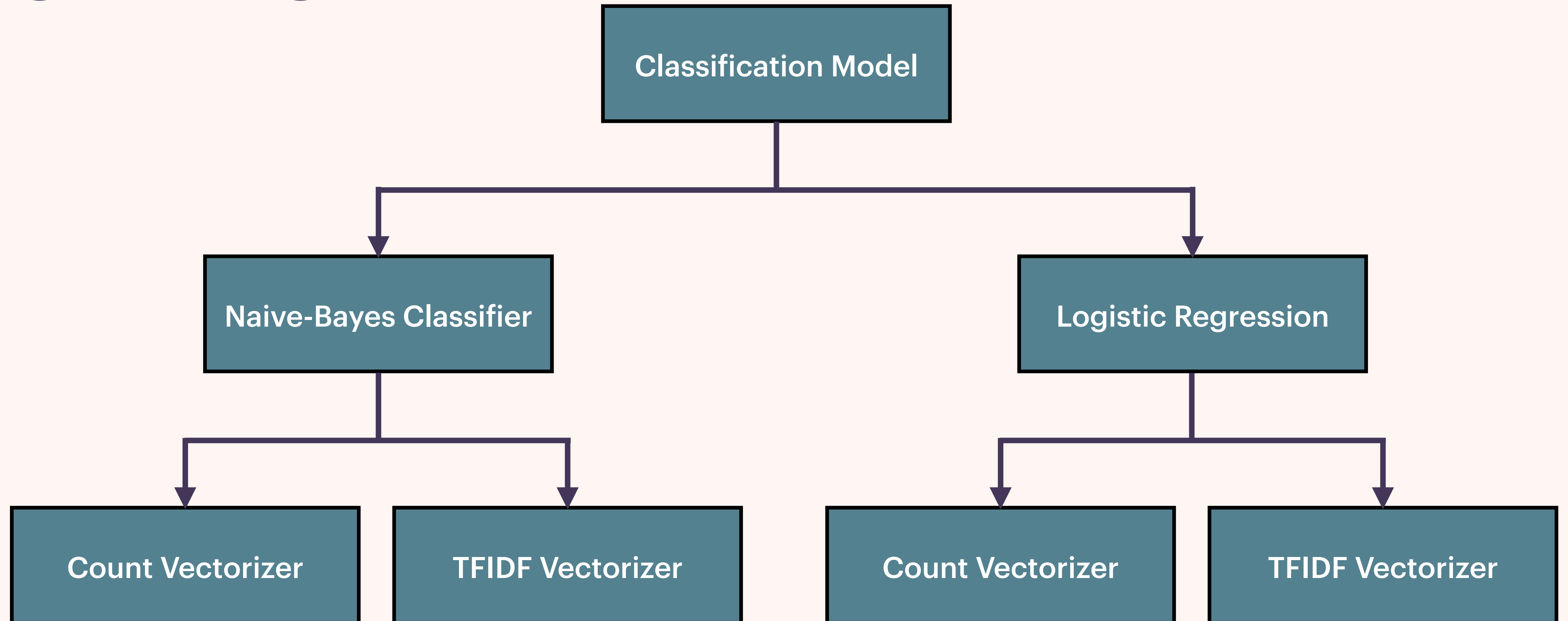
**SIZE IS NOT AN
ISSUE, IF IT'S THE
RIGHT COMMUNITY**

**Albeit the lower subscriber and
activity in Suicide Watch, the average
upvote ratio is very close to 1.**

=

More Positive Community

MODELING



RESULTS

MODEL	VECTORIZER	CROSS VALIDATION ACCURACY	TEST SET ACCURACY
Baseline	-	0.501	0.501
Logistic Regression	Count Vectorizer	0.986	0.863
Logistic Regression	TFIDF Vectorizer	0.942	0.888
Naive-Bayes Classifier	Count Vectorizer	0.936	0.868
Naive-Bayes Classifier	TFIDF Vectorizer	0.918	0.870

** All models tested are slightly overfitted*

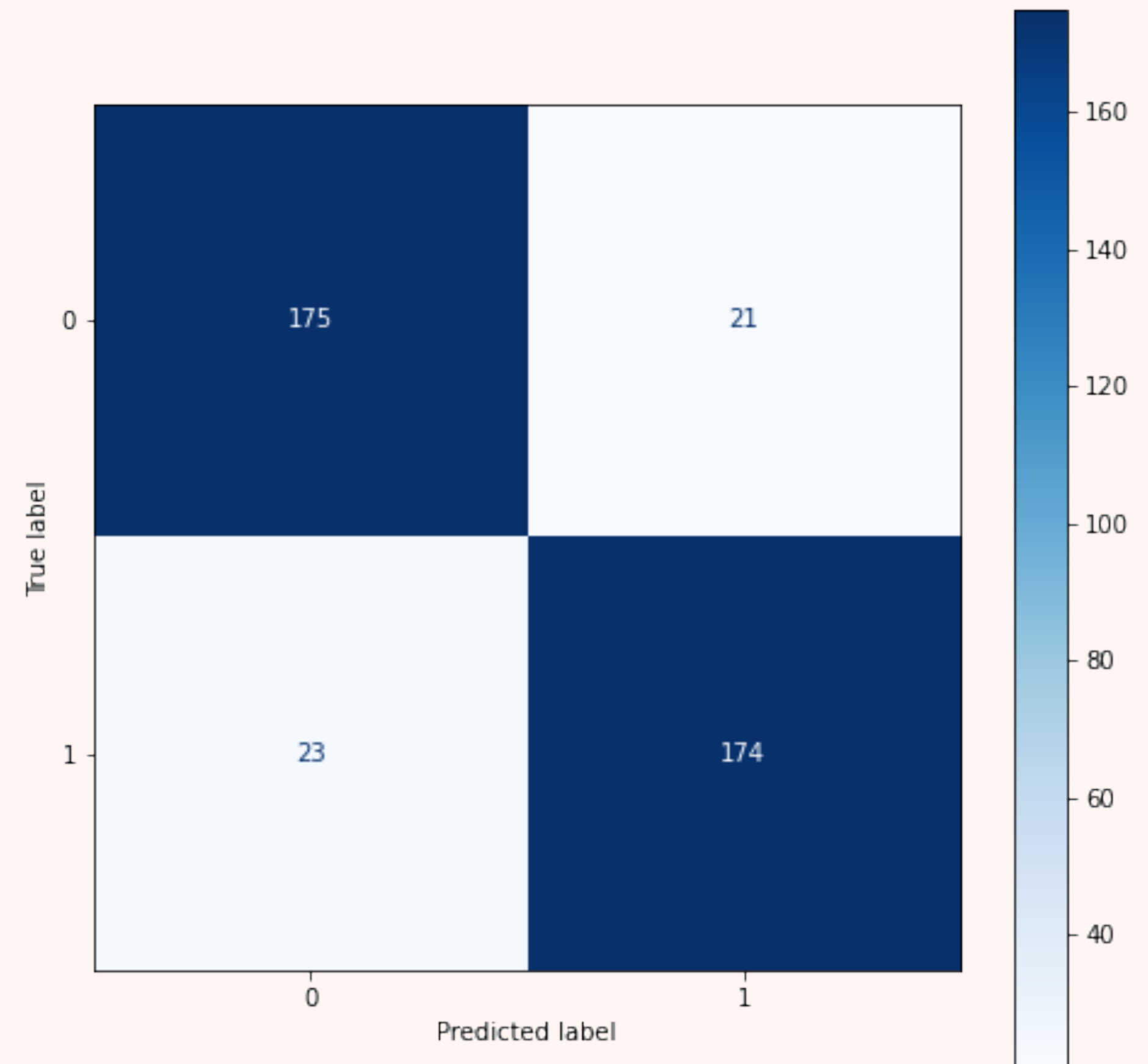
THE WINNER IS...

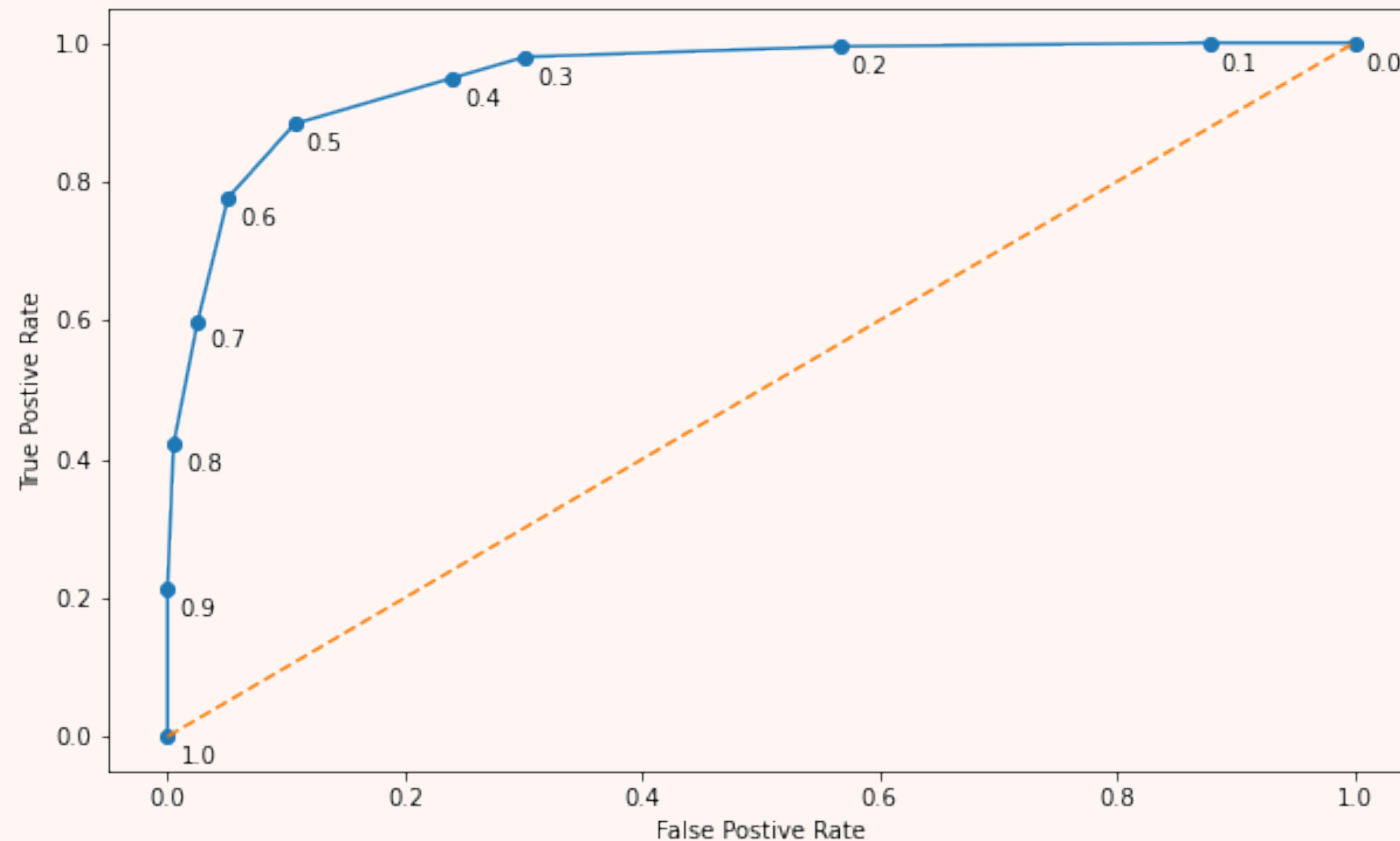
Logistic Regression with TFIDF Vectorizer

Accuracy : **$(174+175) / 393 = 0.888$**

Sensitivity : **$174 / (174 + 23) = 0.883$**

Specificity : **$175 / (175 + 21) = 0.893$**

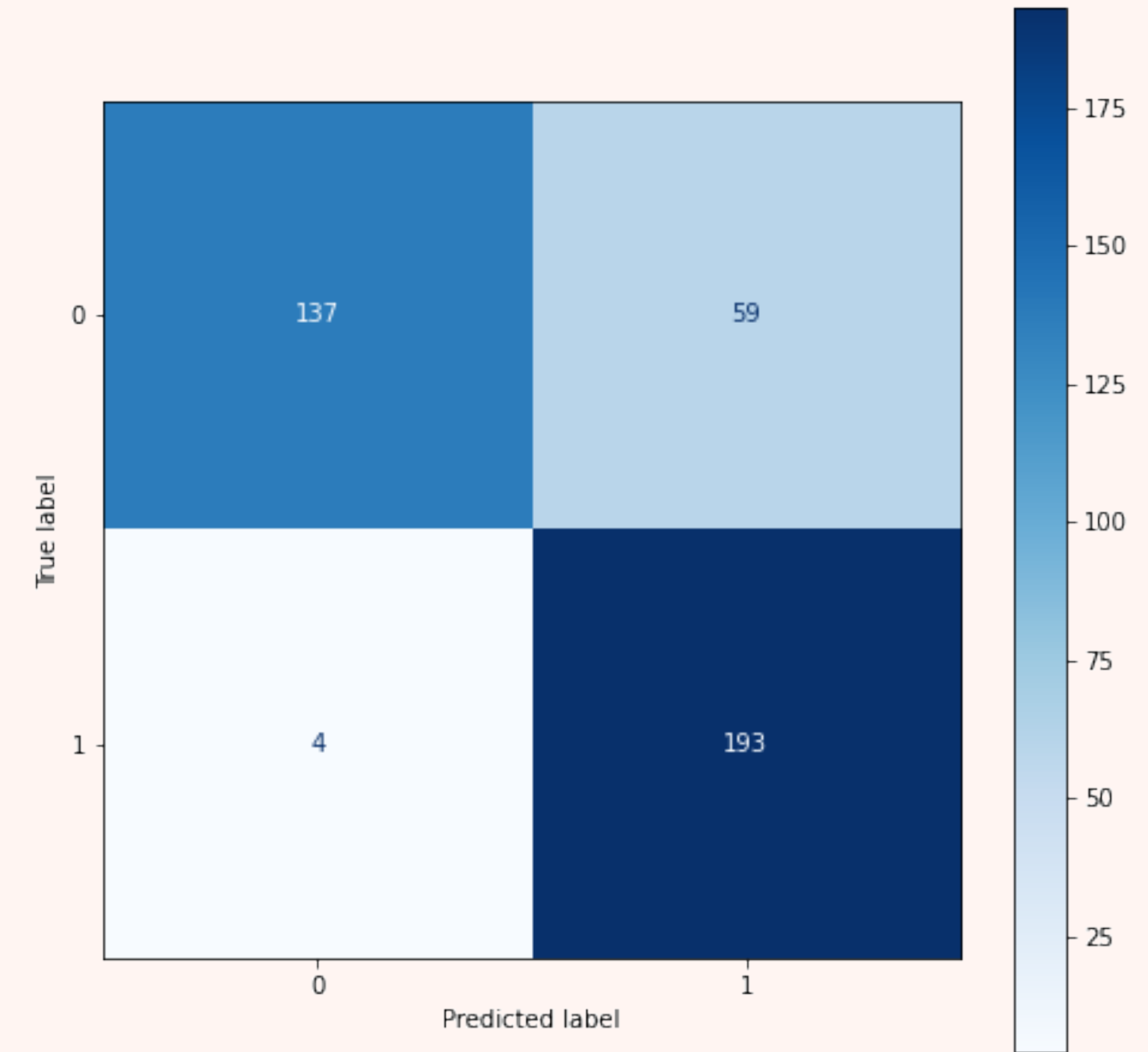
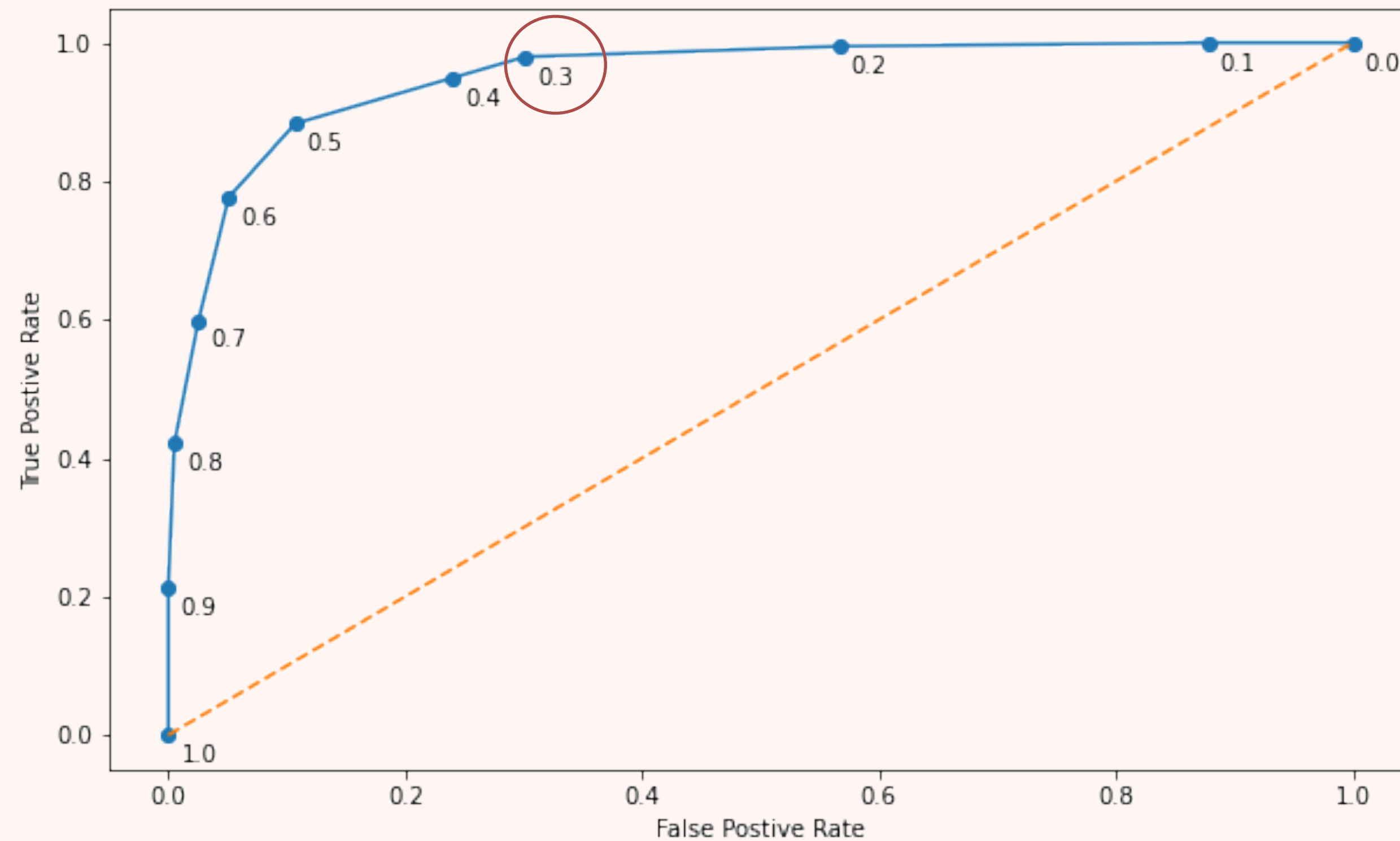




SEARCH OF THE OPTIMUM THRESHOLD

We can adjust the threshold of our logistic regression to optimise between sensitivity and specificity.

POST-ADJUSTMENT



Sensitivity : $193 / (193 + 4) = 0.980$ ↑

Specificity : $59 / (137 + 59) = 0.699$ ↓

TOP 10 WORDS COMPARISON

Pos	Words	Probability
1	Kill	61.5
2	F*ck	14.3
3	Wish	10.2
4	Life	7.7
5	People	7.3
6	Die	6.8
7	End	6.4
8	Hotline	6.3
9	Want Die	5.7
10	Depress	5.3

CLASS 1

Pos	Words	Probability
1	Edit	0.1
2	Know	0.1
3	Update	0.2
4	Reality	0.2
5	Friend	0.2
6	Said	0.3
7	Work	0.3
8	Guy	0.3
9	Ask	0.3
10	Thank	0.3

CLASS 0

CONCLUSION

- **Our model is able to achieve 90% of accuracy**
 - **Managed to minimise Type-II error with tuning threshold to 0.3**
 - **However, it comes at a cost of specificity**
 - **Model can be improved by using advanced technique, eg. Voting Classifier**
 - **Combine current models with CART technique**
 - **Complement weakness of each other**
-

THANK YOU !
