Sentiment Analysis of Steam Games Reviews

Optimization using Bayesian search

By Muhammad Rafif Rabbani

Project Overview

Goal
To classify whether a reviewer recommended the game or not and analyze what words are associated with each sentiment.

Dataset
Reviews of the top 25 top-selling games with mixed reviews (<80%) since the beginning of 2022.

Focus
- End-to-end project on NLP: from scraping the data to model deployment.
- Text processing and visualization.
- Hyperparameter tuning using Bayesian search.

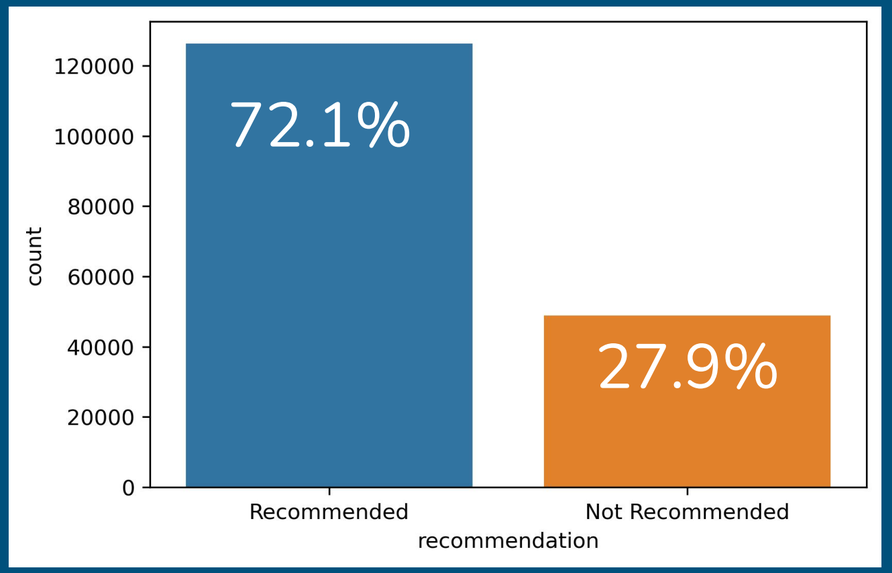
Dataset

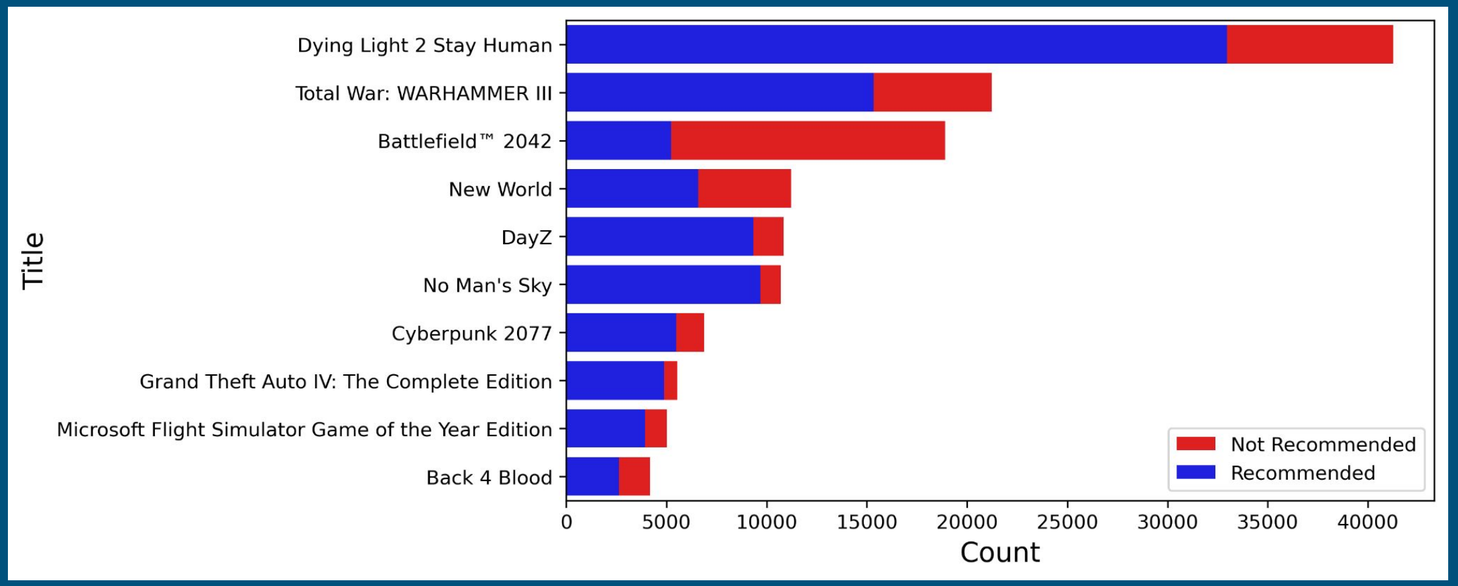
- Top 25 best-selling games with mixed reviews → scraped with BeautifulSoup4 and Selenium.
- The reviews since 1 January 2022 → from the Steam API. Obtained ~170k reviews.

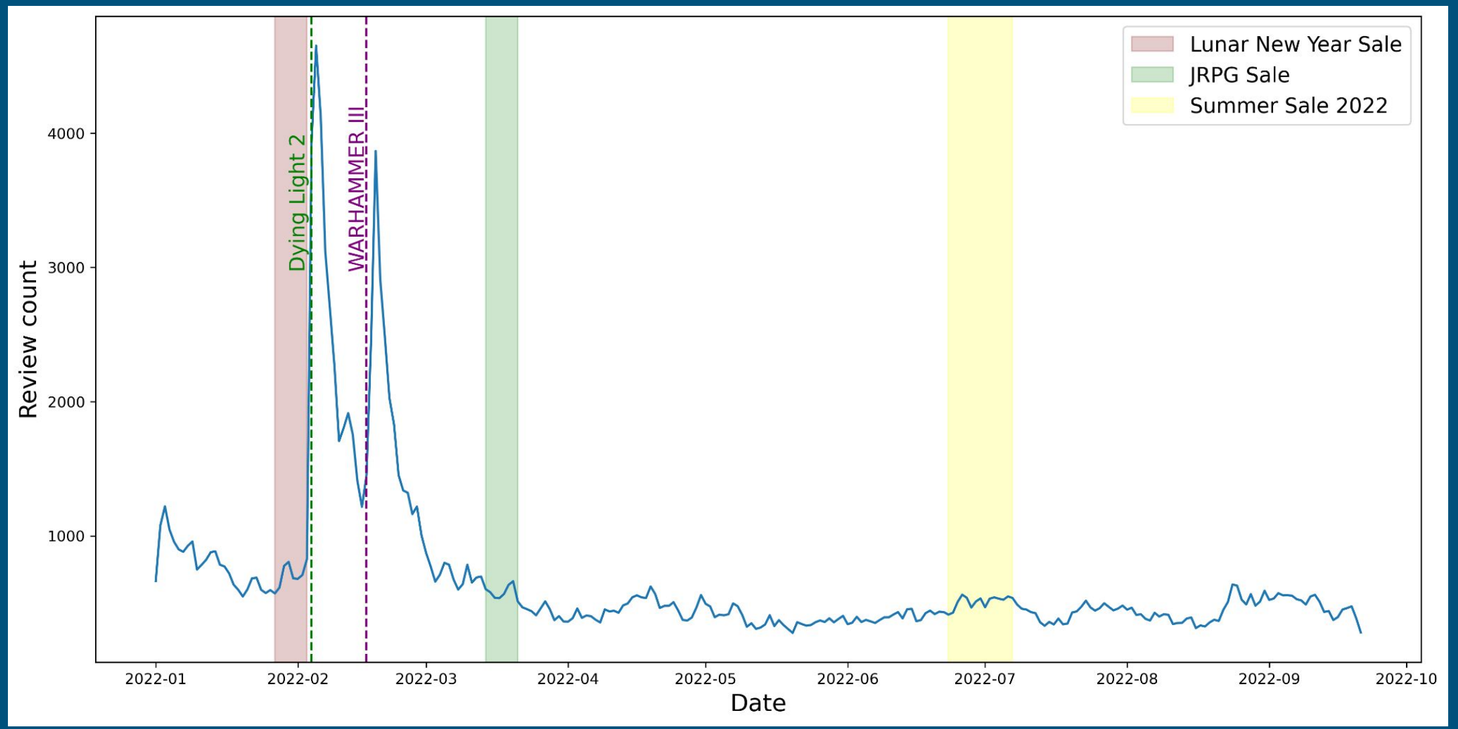
Columns	Type	Description
author	obj	Information about the author
review	obj	review text
timestamp_created	int	Unix timestamp when the review was created.
voted_up	bool	recommended or not
votes_up	int	Total users who gave ‘helpful’ vote.
votes_funny	int	Total users who gave ‘funny’ vote.
name	obj	Game title.

EDA: Visualization

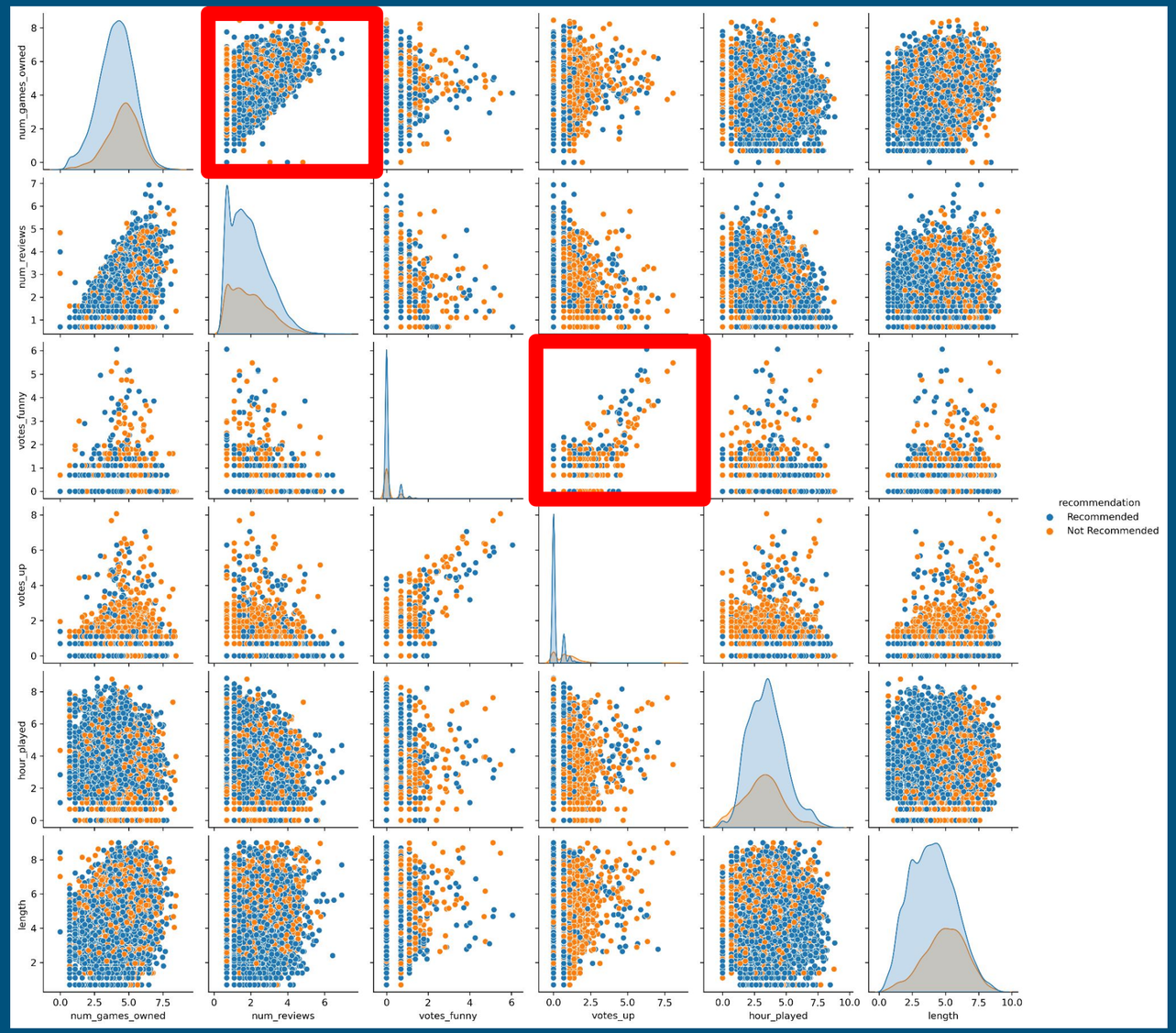
- Class imbalance (positive > negative)
- Two new games: Dying Light 2 and WARHAMMER III.

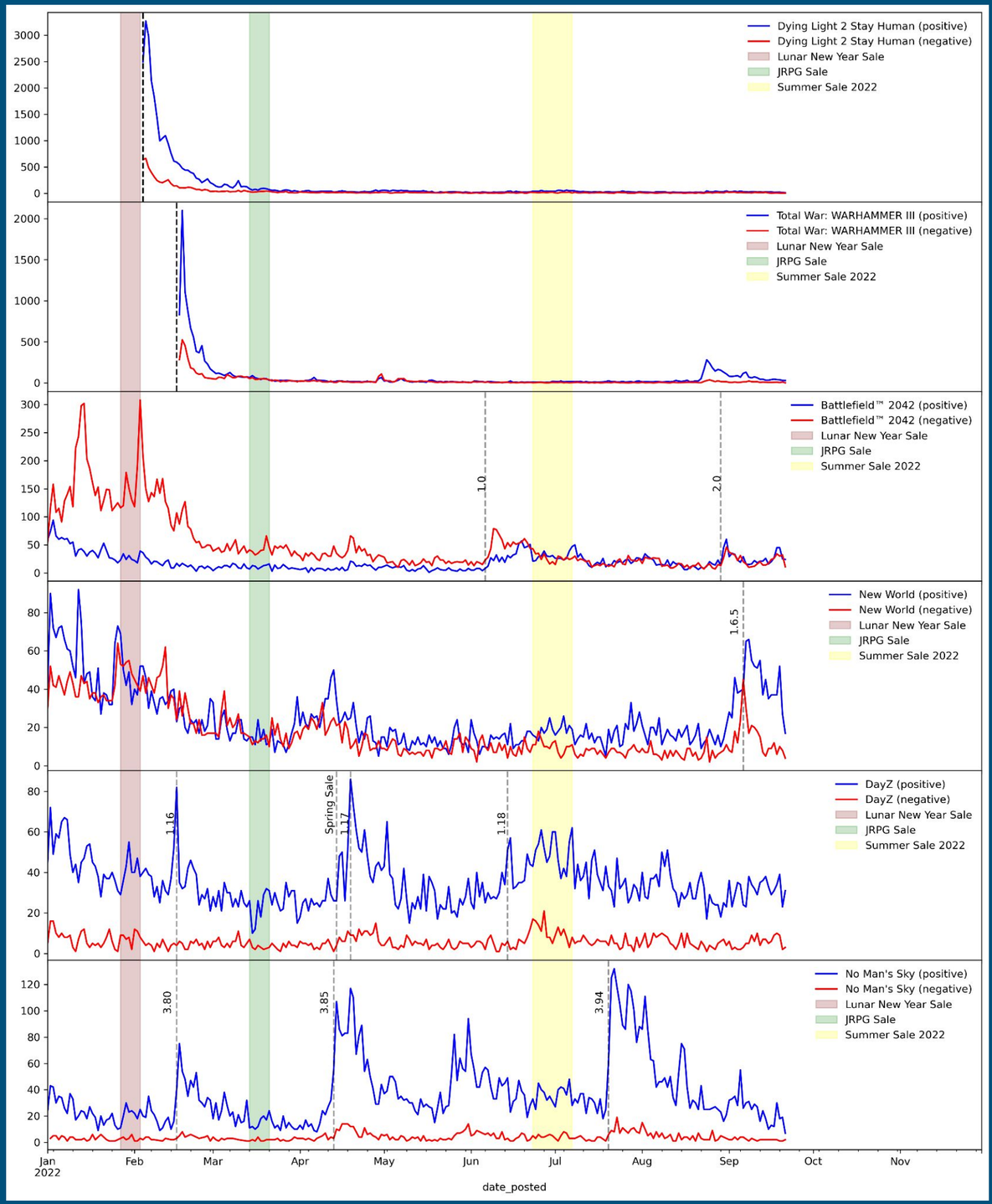






EDA: Visualization





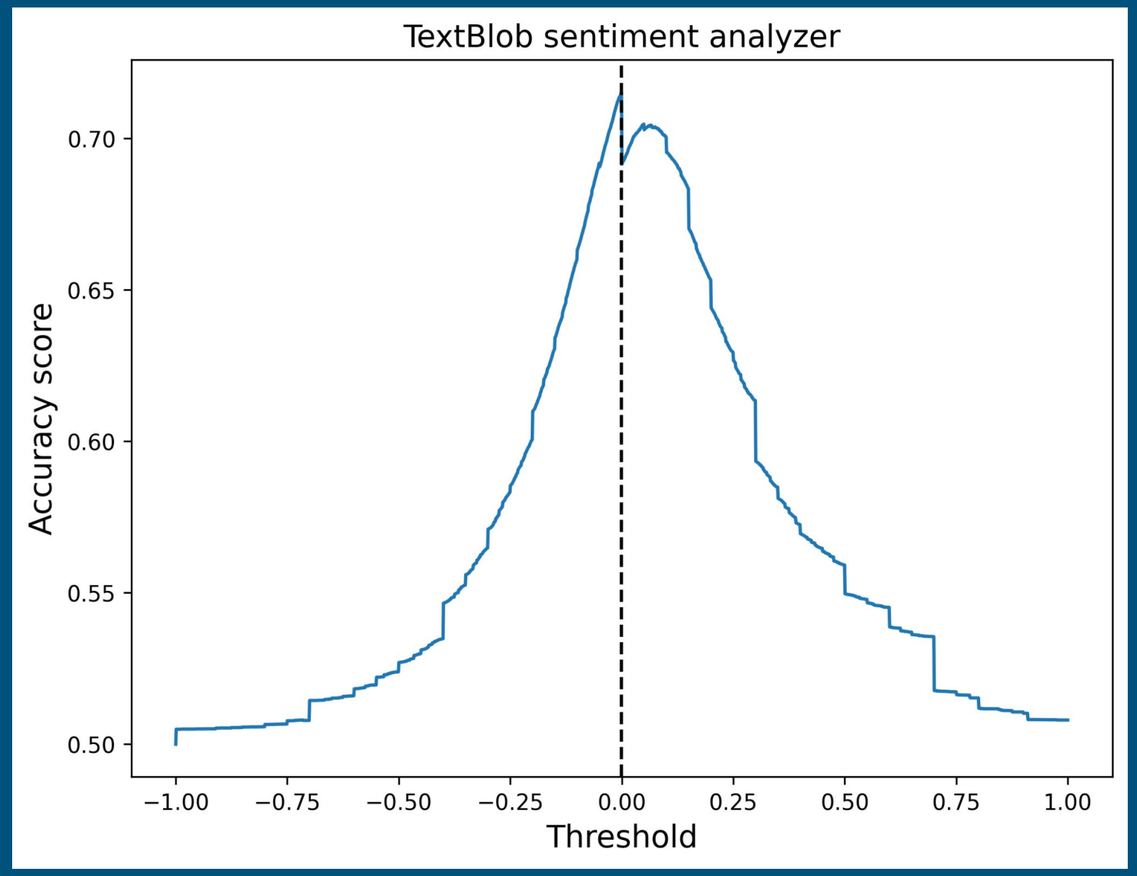
EDA: Visualization

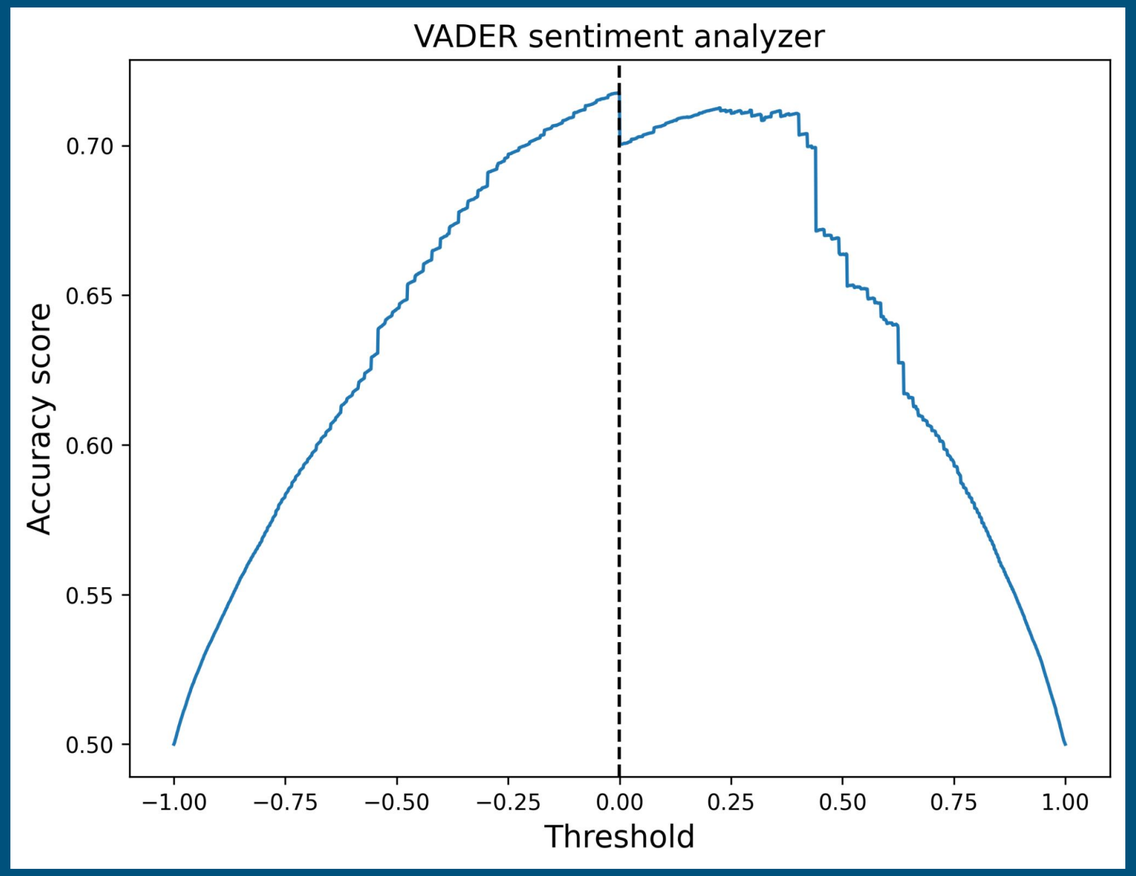
Positive reviews: ‘fun’, ‘worth’, ‘great game’, ‘amazing’.

Negative reviews: ‘bug’, ‘issue’, ‘boring’.

Rule-based: TextBlob and VADER

- Unsupervised method, no need for text cleaning.





- Low performance (typical).

Text Processing

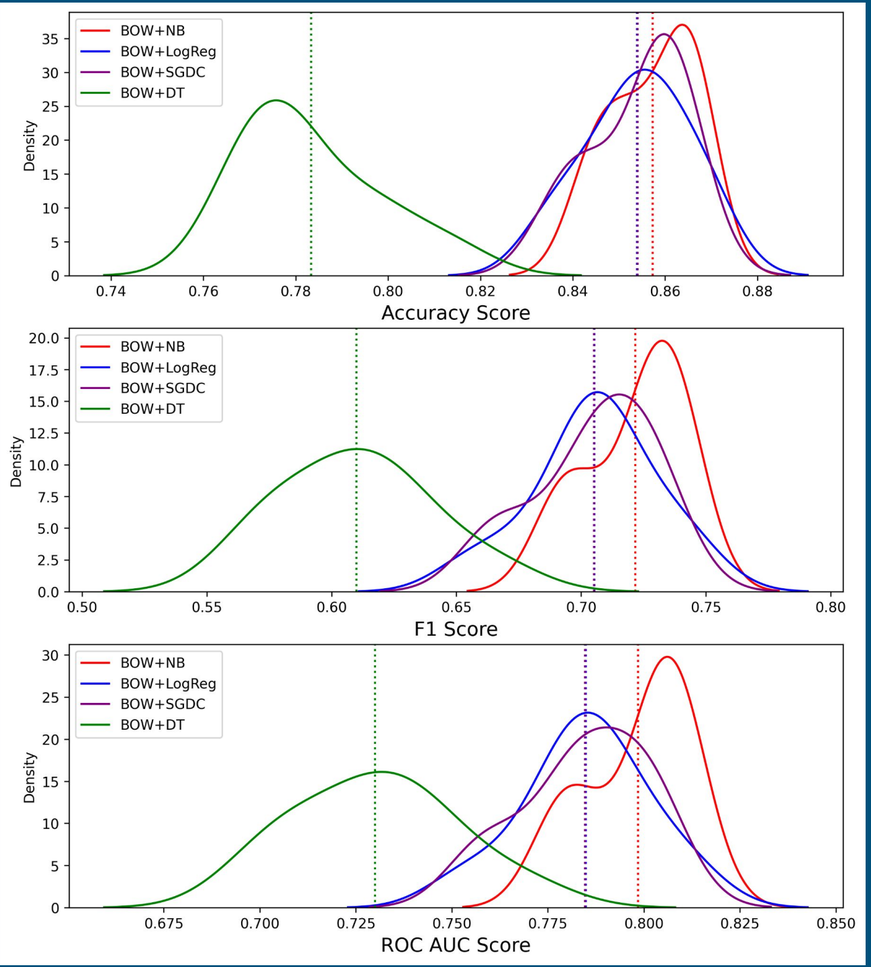
1. Remove links with regex.
2. Remove hyperlinks and markups.
3. Remove non-alphabetical characters (punctuations, digits, emojis, special characters)
4. Remove stop words with the nltk library e.g. I, the, you, and, what.
5. Stemming with the nltk library.
6. Remove troll messages (more than 5 consecutive consonants e.g. wwwwwwww).

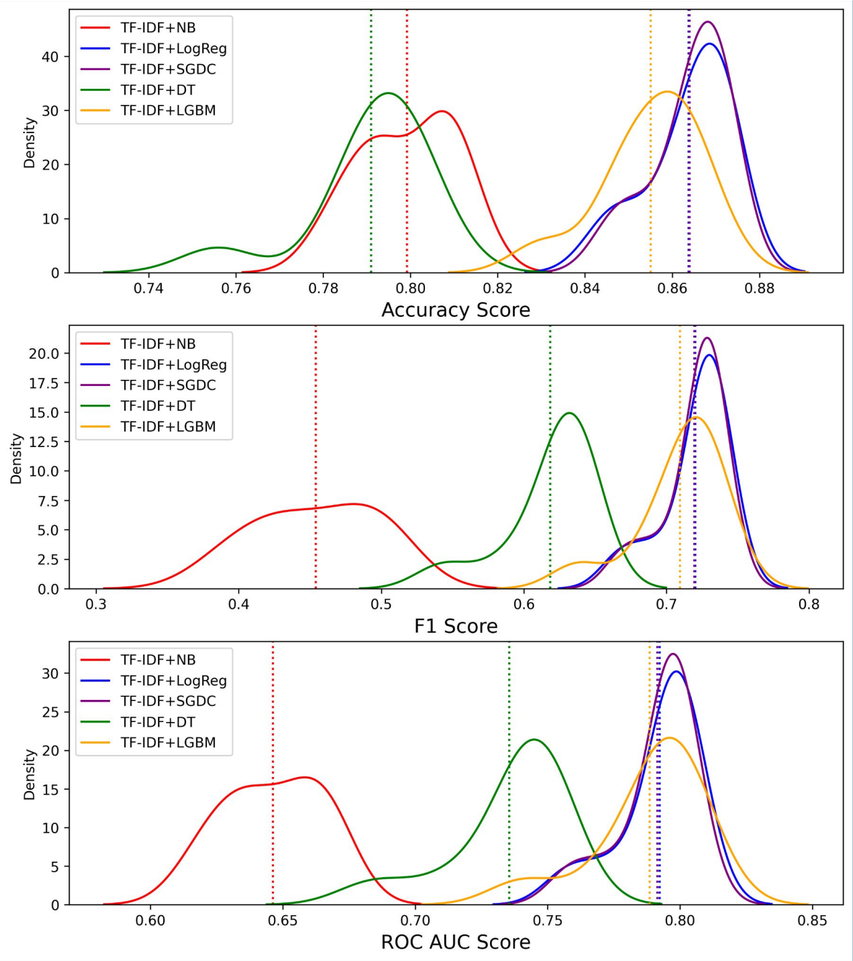
- Example result:

```
Original review:

Brand New video showing Resident Evil 3 in true virtual reality https://youtu.be/WvTyGZbLVh4

==========================================================================================
Cleaned review:

brand new video show resid evil true virtual realiti
```

Model Building: BOW and TF-IDF

- 10-folds cross-validation to get the distributions.



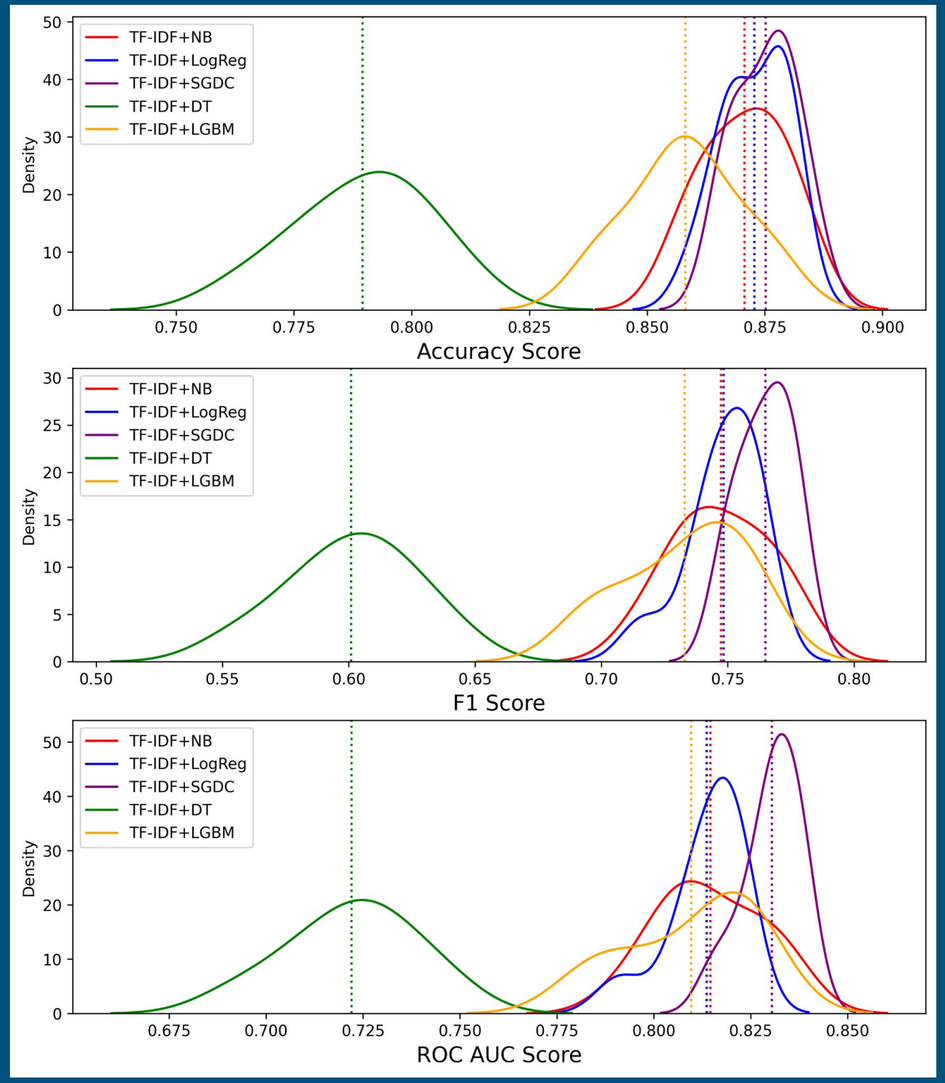


- Baseline: BOW+NB.
- Overall better performance with TF-IDF.
- Best model: TF-IDF+SGDC.

Model Building: Bayesian Search

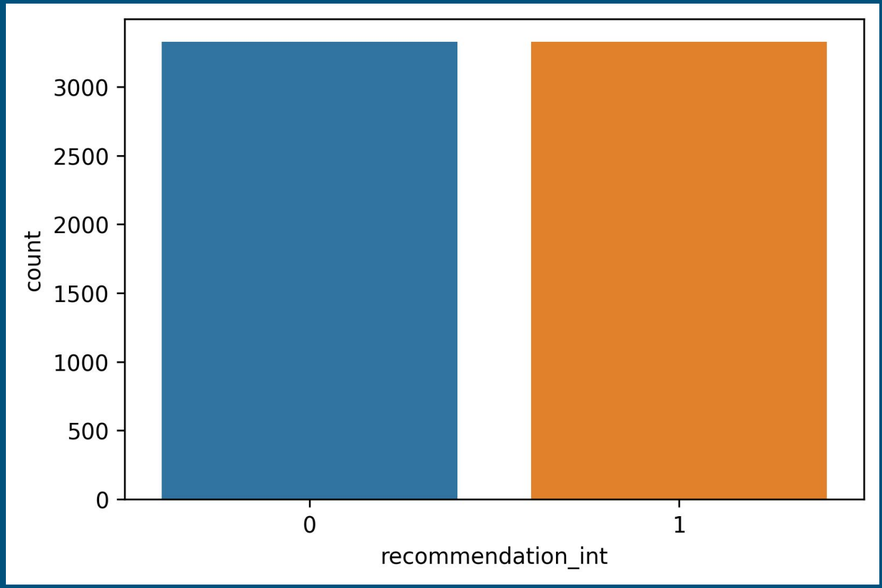
	Parameter name	Range
TF-IDF Vectorizer		
Maximum document frequency	`max_df`	0.5, 0.75, 1.0
Minimum document frequency	`min_df`	1, 5, 10
Maximum features	`max_features`	None, 500, 1000
Naive Bayes		
Smoothing parameter	`alpha`	logU[-3,1]
Logistic regression		
Regularization strength	`C`	logU[-3,1]
Norm of the penalty	`penalty`	l2, None
Optimization algorithm	`solver`	newton-cg, lbfgs
SGDC		
Smoothing parameter	`alpha`	logU[-5,0]
Decision Tree		
Splitting criterion at each node	`criterion`	gini, entropy
Maximum tree depth	`max_depth`	10, 20, 30, 40, 50
Minimum samples for a node	`min_samples_leaf`	5, 10, 20, 50, 100
LightGBM		
Maximum tree depth	`max_depth`	3, 4, 5, 6, 7, 8, 9, 10
Number of leaves	`num_leaves`	U[20,3000]
Learning rate	`learning_rate`	U[0.05,0.2]

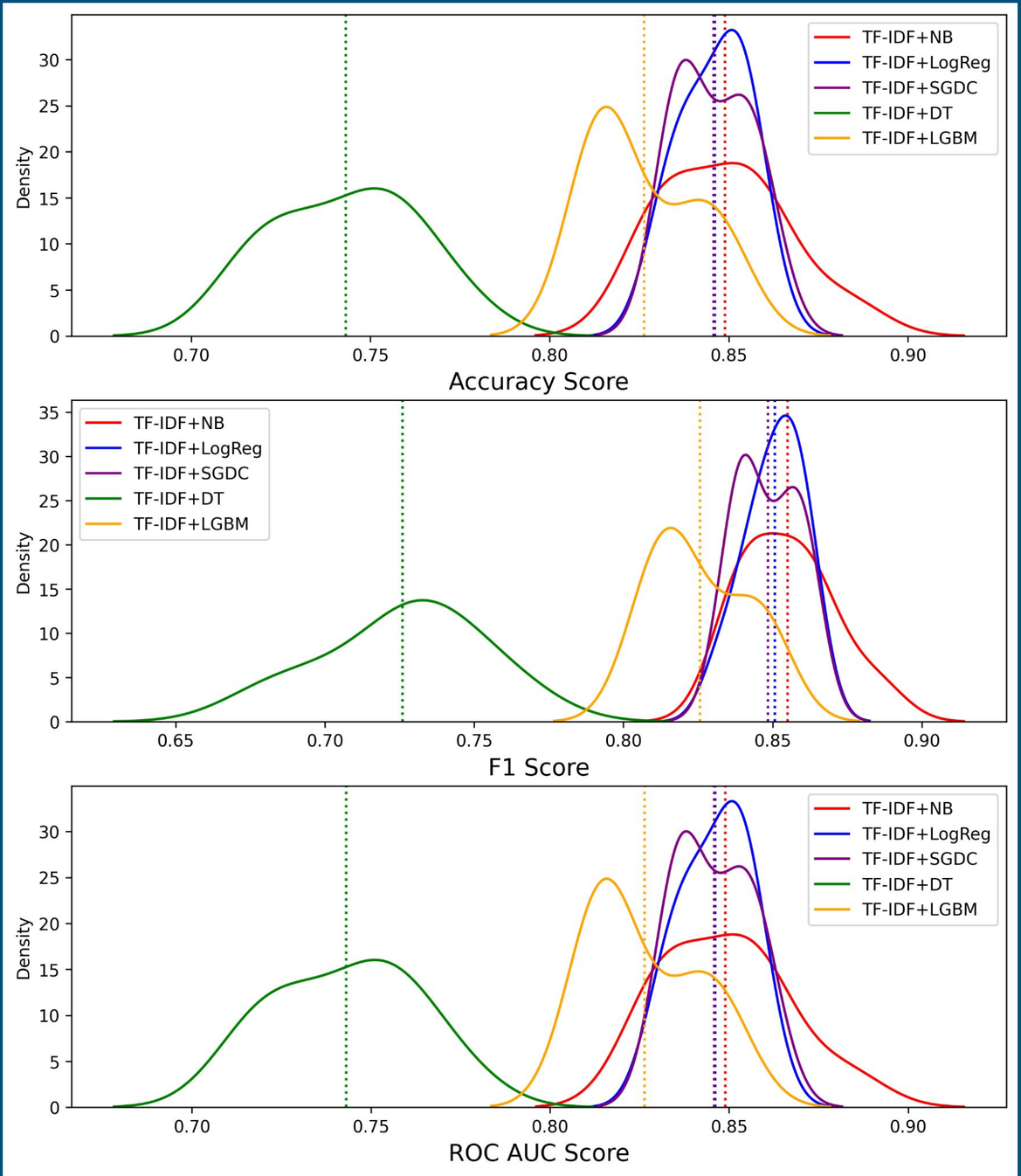
- Using hyperopt, maximizing F1-score.



Model Building: Class Balancing

- Undersampling the majority class.
- No significant improvements → use original (imbalanced) training data.

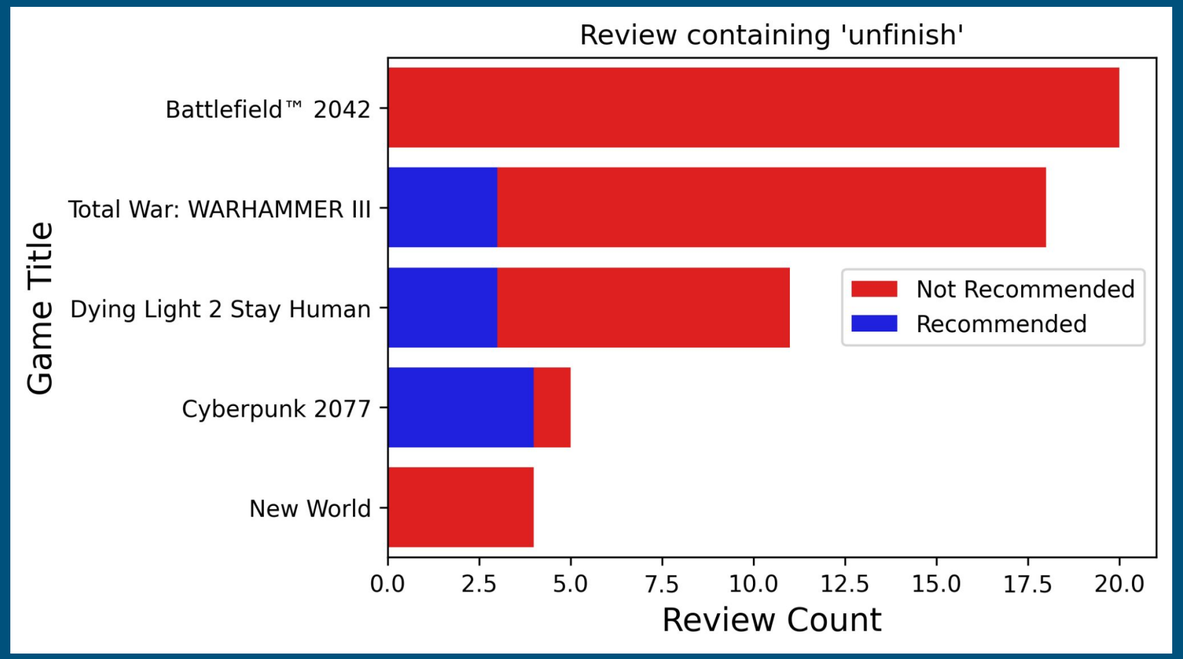




Final Model: relevant feedback

- Best model: TF-IDF + SGDC

Positive		Negative	
Word	Weight	Word	Weight
great	+8.084	crash	-6.711
best	+7.196	ubisoft	-5.904
amazing	+6.801	bug	-4.701
love	+6.348	unfinish	-5.037
awesome	+6.069	ban	-4.966
good	+4.981	bore	-4.417
nice	+4.025	downgrad	-4.359
addicting	+3.755	item	-4.272
fun	+3.692	bug	-3.789
friend	+3.624	nerf	-3.787

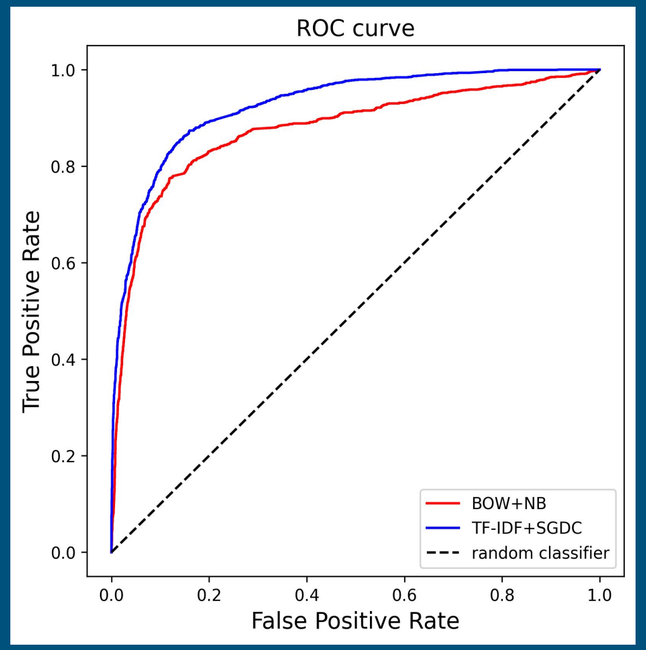


- Crashing problem → WARHAMMER III, DL2, BF.
- Game felt like unfinished → Battlefield 2042.
- The first game was better → DL2.
- Needs more items, don’t nerf them → DL2

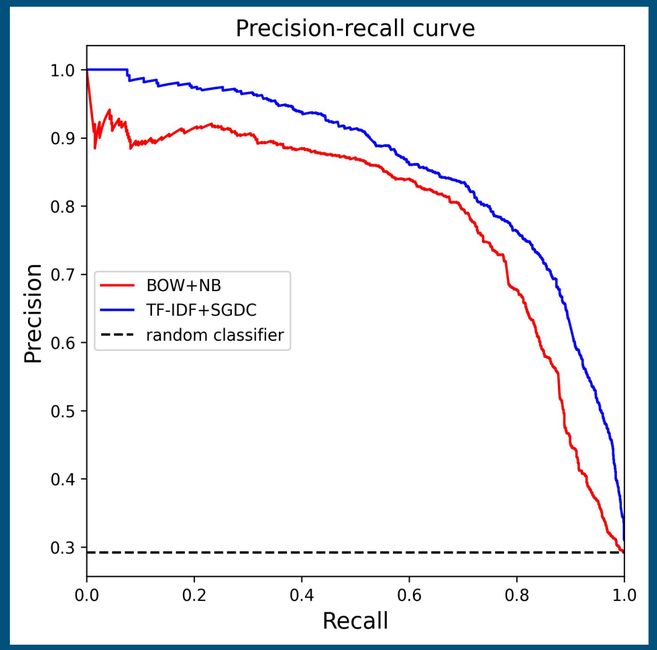
Model Evaluation: test data

		Predicted	
		Not Recommended	Recommended
Actual	Not Recommended	1087	256
	Recommended	410	3376

ROC curve



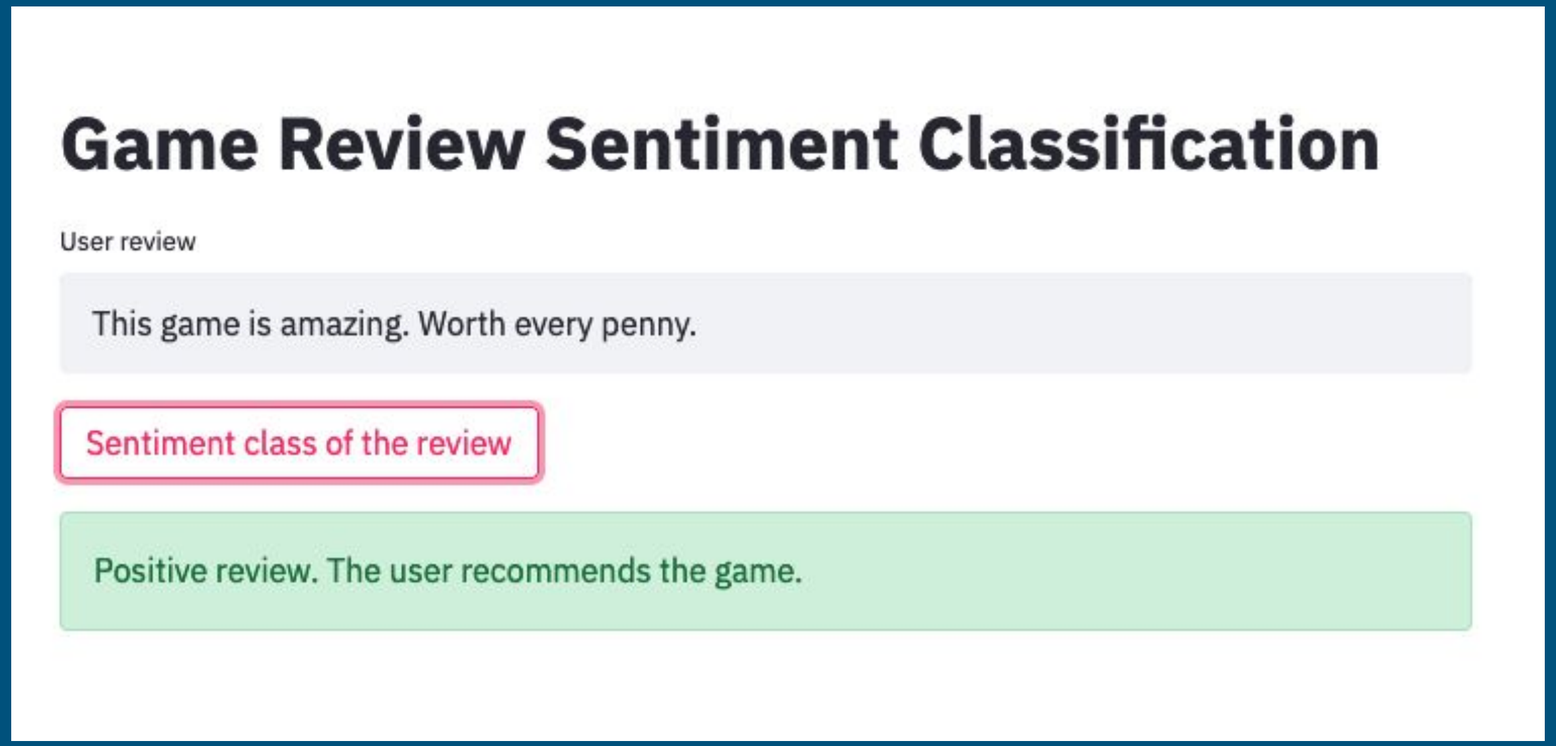
Precision-recall curve



- Evaluation summary:
  - F1-score = 76.5%
  - ROC AUC = 0.828
- Baseline model: F1-score is 72.1%, ROC AUC score is 0.793 → improved.
- Good chance this model can distinguish the two sentiments on larger, unseen data.

Model Deployment: Streamlit

- Still very simple. Access here.



- More features coming soon!

Conclusions

- I have scraped reviews of the top 25 top-selling games with mixed (<80%) reviews on Steam since 1 January 2022, obtaining ~170k reviews with labels.
- Class imbalance: more positive reviews than negatives.
- Best model: TF-IDF+SGDC with hyperparameters from Bayesian search.
- F1-score increases from 72.1% to 76.5%, and ROC AUC score increases from 0.793 to 0.828 → may perform well on larger, unseen data.
- Model deployed via Streamlit, but still needs polishing and adding more features.