Natural Language Processing

- from Academic Theory to Business Application

PyData Bristol - 26th Meetup

Jerry Mundondo

PyData Bristol

WHO AM I?

- Data Scientist with a fairly recent Academic Background
- MSc Data Science And Artificial Intelligence
- Working within the video game marking industry

UNIVERSITY OF LIVERPOOL

FOURTH FLOOR

WHAT IS THIS TALK ABOUT?

- The difference between learning Data Science and implementing it
- How a Natural Language Processing project changed my mindset around data science

THE BIG PICTURE

Think of a game → Develop the game → ??? →

???

↓

Digital Marketing

FOURTH FLOOR

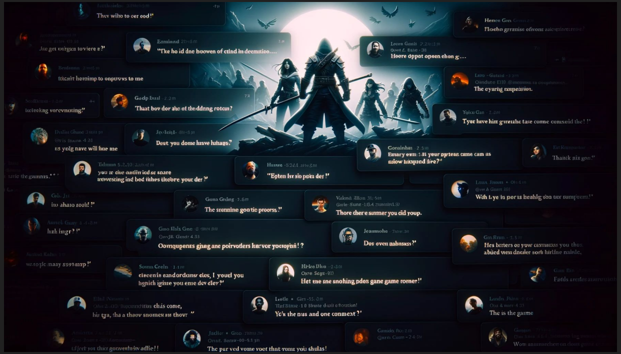
→

THE PROBLEM AT HAND

Players leave comments on the marketing ads

Code a tool that does two things :

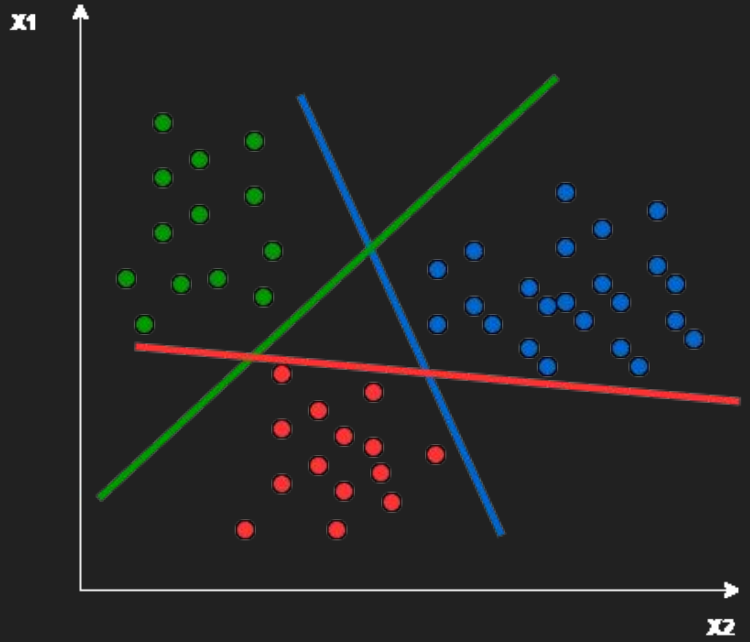
1) Assesses the sentiment of comments
2) Creates a thematic summary of the comments



“the gameplay sucks because …”

ACADEMIC PROBLEM SOLVING

- Sentiment - Positive, Negative, Neutral
- Themes - Price, Gameplay, Competitors, Release Information, Game Information
- Academic approach :
  - Build a classification algorithm
    - Use data to train the algorithm
    - Test the algorithm
    - Keep iterating on it until accuracy improves
    - Present my findings



ROAD BLOCK #1

- Data availability
- Data quality
- Unlabelled Data
- Limited time

PyData Bristol

APPROACH #1 - OFF THE SHELF

- Sentiment Analysis
- Experimented with various iterations like:
  - TextBlob
  - Vader
  - BERT (and variations)
- The accuracy within all these models was poor

REASON

- The language used by gamers is domain specific
- Could not find a model trained specifically for gamers
- Example :

"The game's shooting mechanics are sharp."

Results:

sentence = "The game's shooting mechanics are sharp."

blob = TextBlob(sentence)

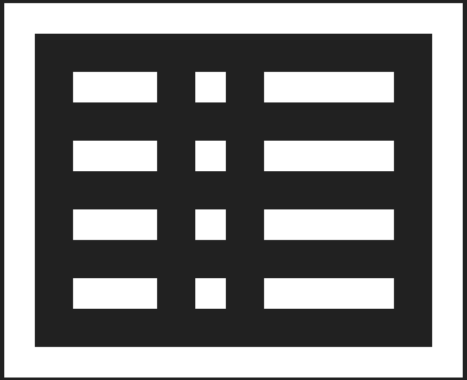
Polarity = -0.2625, ***Subjectivity*** =0.575)

APPROACH #2 - FINE TUNING

- Fine-tune the existing sentiment models
- Led to the similar problems:
    - Would still need structured data to train on
    - The Data would need to be labelled
    - Would not address the problem with thematic classification
    - Would be time consuming

PROBLEM SOLVING - IN THE REAL WORLD

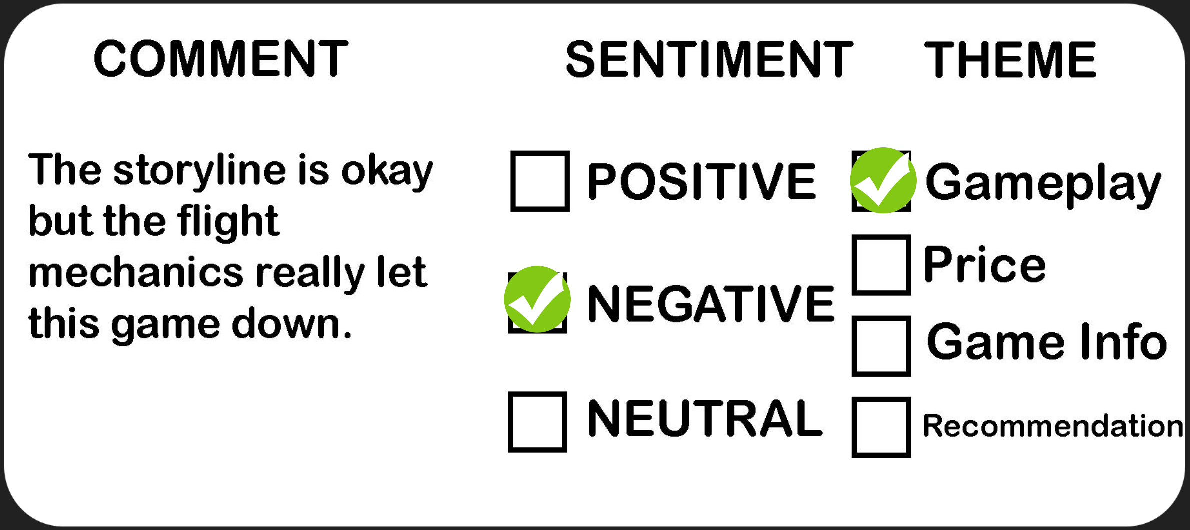
- Unstructured and unlabelled data
- Needed thousands of comments correctly labelled
- Time consuming
- Boring



- Leverage the knowledge of 100 employees

DATA GATHERING APPROACH

- Developed a website with a simple interface :



- Use that data to finetune the off-the-shelf model

FAILURE #3 - DATA COLLECTION

- My co-workers did not have the full context of what we were trying to achieve
- Resulted in mass misclassification of comments
- Less comments labelled than expected
- Result : sparse, inaccurate data

A GLIMMER OF HOPE

```python
from transformers import pipeline
classifier = pipeline("zero-shot-classification")
sequence = "The fight mechanics of this game are great."
candidate_labels_Sentiment = ["Positive", "Negative", "Neutral"]
classifier(sequence, candidate_labels_Sentiment)
```

```
{'sequence': 'The fight mechanics of this game are great.',
 'labels': ['Positive', 'Negative', 'Neutral'],
 'scores': [0.880562007427216, 0.0643775537610054, 0.05506044998764992]}
```

```python
candidate_labels_Theme = ["Gameplay", "Price", "Release Date"]
classifier(sequence, candidate_labels_Theme)
```

```
{'sequence': 'The fight mechanics of this game are great.',
 'labels': ['Gameplay', 'Price', 'Release Date'],
 'scores': [0.9307592511177063, 0.049659911543130875, 0.019580841064453125]}
```

```python
candidate_labels_Theme = ["Gameplay", "Price", "Release Date"]
hypothesis_template = "The comment focuses on the  {} of the game."
classifier(sequence, candidate_labels_Theme, hypothesis_template=hypothesis_template)
```

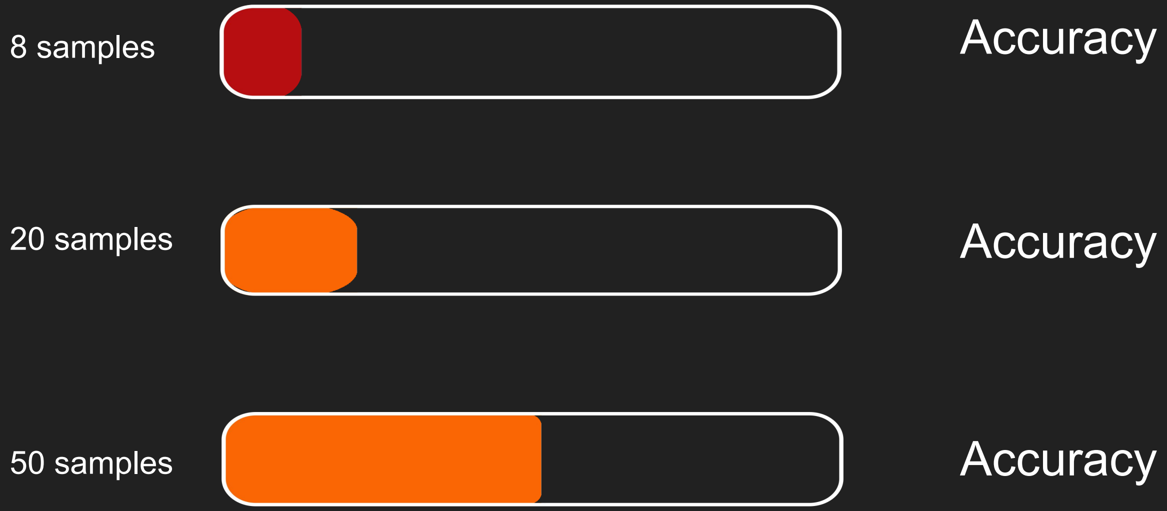
```
{'sequence': 'The fight mechanics of this game are great.',
 'labels': ['Gameplay', 'Release Date', 'Price'],
 'scores': [0.9934735298156738, 0.0038008813280612230, 0.0027255986351519823]}
```

FEW SHOT CLASSIFICATION

- Provide a few labelled pairs for training
- Use HuggingFace's 'SetFit' trainer to fine tune the classifier
- Increased the accuracy

FURTHER PROGRESS

- The more labelled data that I provided it, the more it improved

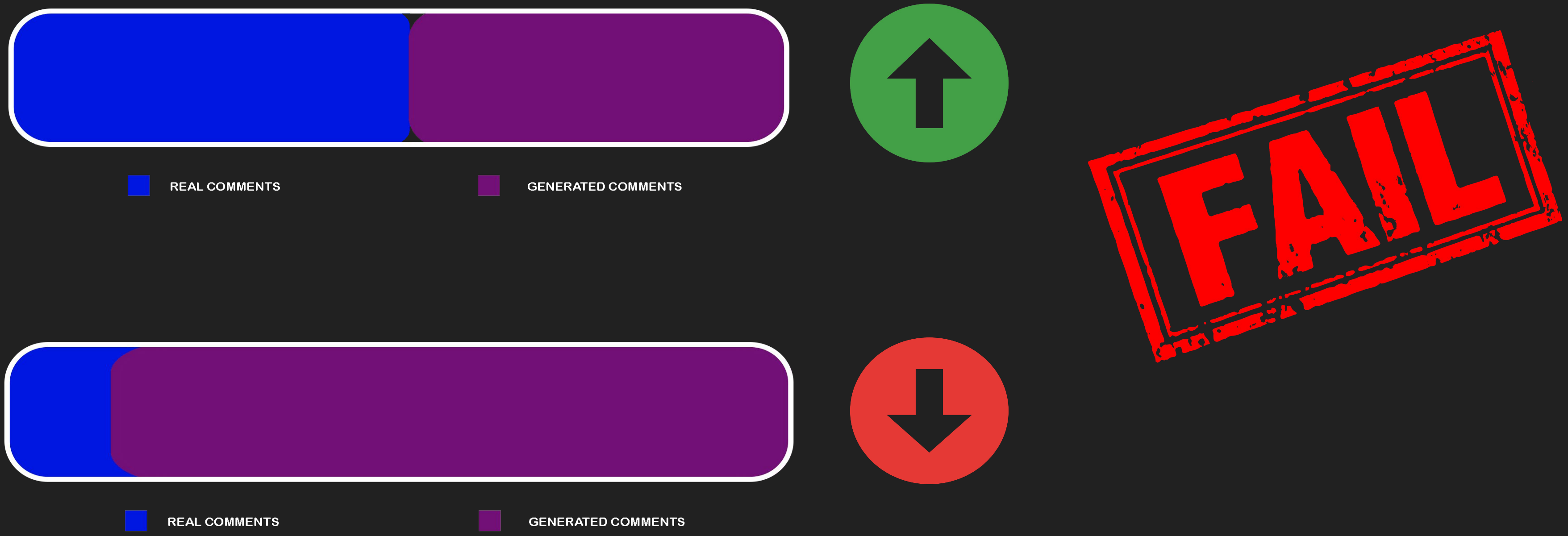


* representative

REALISATION

- I could manually labelling thousands of comments for each category
- Or use Generative AI to augment the data
- Feed in a few example comments and labels
- Ask it to generate similar comments and label them
- Theoretically increased comments and labels

FAIL #? - SYNTHETIC DATA



NEW APPROACH

- I decided that for each category I would manually label e.g. 500 comments
- Ask gpt for 200 more comments
- Use the augmented data set to train a model using few shot classification
- Monitor accuracy

SUCCESS

- Finally achieved good accuracy
- No need to label excessive number of comments
- And fulfilled the objectives laid out for the project

KEY TAKEAWAYS

- Universities do a great job of teaching technical skills
- A larger emphasis on dealing with real world data is needed
- A greater focus on problem solving
- Rather than having the most complex models
- Bonus : caution needs to be exercised when augmenting data using LLMs

QUESTIONS?

RESOURCES

- HuggingFace zero-shot-classification : https://huggingface.co/tasks/zero-shot-classification
- HuggingFace few-shot-classification/SetFit : https://huggingface.co/blog/setfit
- Me : https://www.linkedin.com/in/jerrypmundondo/