



# Detection of Offensive Language in Football Comments

Paul Roth  
Akshun Dogra  
Jan Baumgart  
Ali Kirac  
Burak Cinar

Final Presentation in Master Course NLP for Business at FAU

06 March 2024

# Agenda



01 Introduction

---

02 Research

---

03 Data Understanding

---

04 Data Preparation

---

05 Machine Learning Models

---

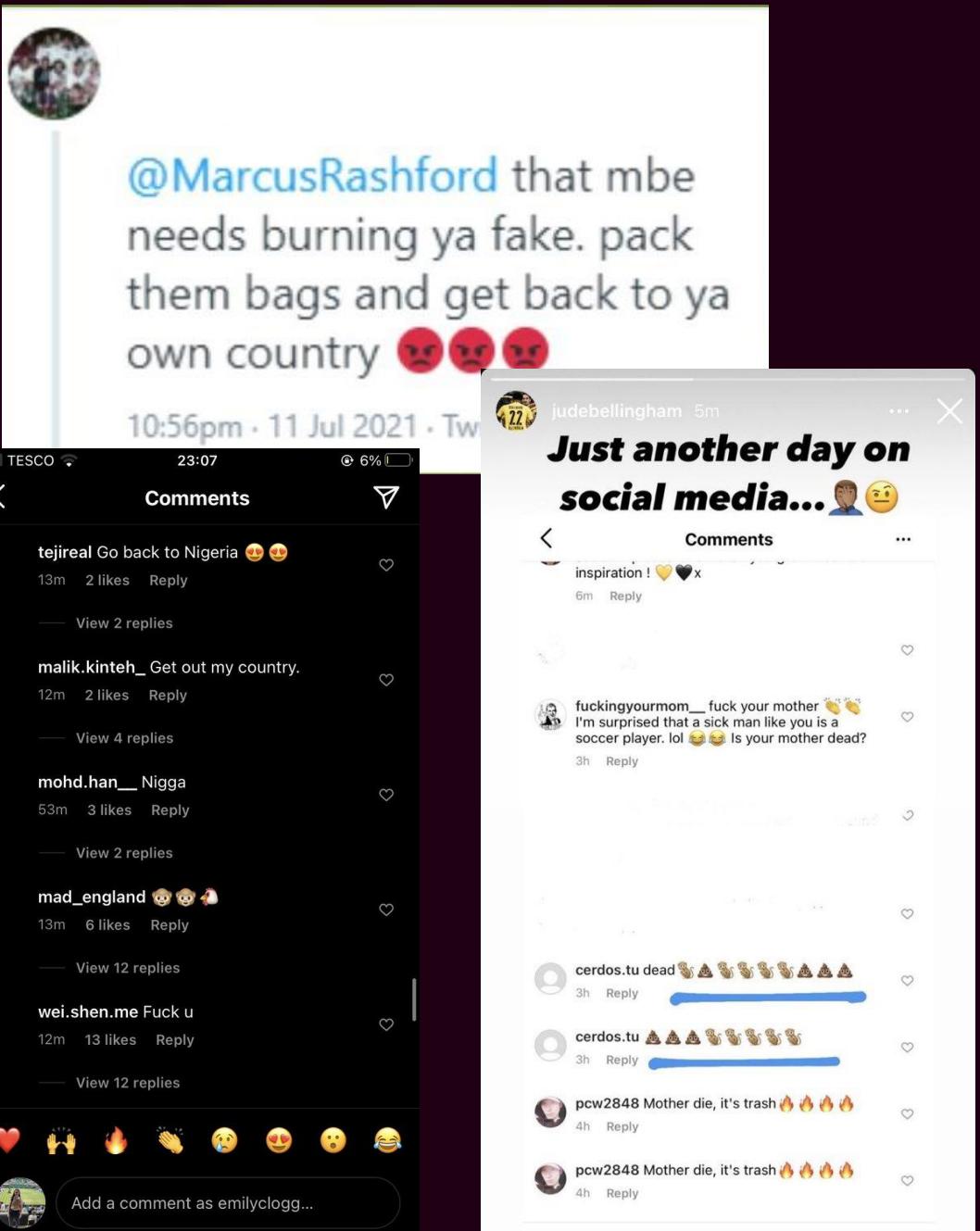
06 Application

---

07 Summary

# Introduction

- Nowadays, when Two teams play, and One team loses. The losing team players not only have to deal with the loss of the game but also the backlash they will experience on social media.
- Mainly it includes Offensive Language and racist remarks against the player, club, or country.



# Key Objectives & Goals

1

Make an Offensive Language detection model with the help of NLP

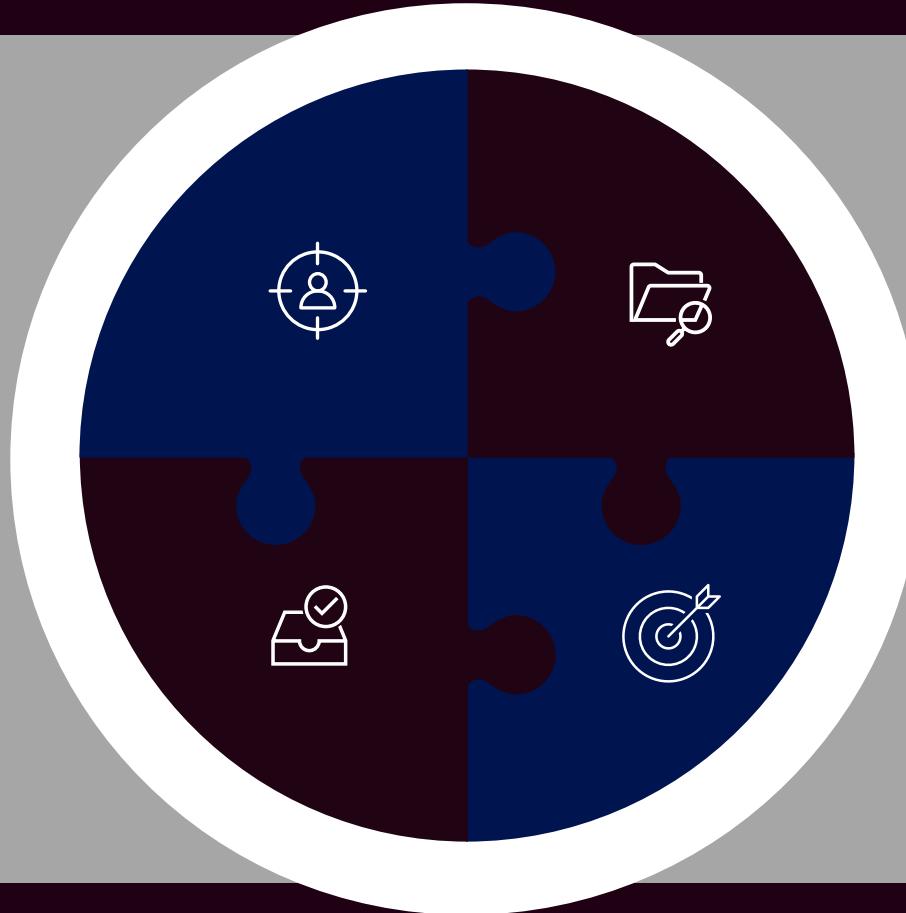
Comparing different Models

2

Clear Guideline

Compare the Levels of Offensive Language between League and International Games.

4



3

# Project Management



# Research

- Business Understanding
- Data collection: Scrape-> Clean -> Analyze
- Which NLP and Methodologies to choose
- Challenges and Limitations
- Result Of the Project

The image shows a social media interface with a dark background. A user has posted a comment about racism, which has received 271 likes and 21 replies. Another user responds with a message about Indian principles and unity. A third user expresses support for NFL players and the same respect for NBA players. Two more users, one from Liverpool and another from Serbia, express their views on racism. A final user from Zinck82 expresses sadness at seeing racism still exist in 2019.

**@fredotachi8634 2 years ago (edited)**  
I didn't finish even watching this clip. I cried on behalf of those great players. RIP to racism  
 271 Reply  
21 replies

**@shilajitc.9609 2 years ago (edited)**  
I'm an Indian and I've learnt from my childhood the very main Indian principle, "Unity in Diversity"  
I hope that the world also accepts this slogan  
Say no to Racism   
 211 Reply  
29 replies

**@darelsis1307 1 year ago (edited)**  
In USA you could never see this against NBA players  Football players deserve the same RESPECT AND LOVE  
 24 Reply  
3 replies

**@AlexAppleby 4 years ago (edited)**  
Being a Liverpool fan I am not a massive fan of Raheem Sterling but what he's doing to help prevent racism is inspiring and no-one can ever take that away from him. #saynotoracism  
 7 Reply

**@serbia4823 4 years ago (edited)**  
I really don't understand the meaning of racism. It's just a skin color, it doesn't change anything. We are all the same. Respect is what people should be doing. #NoToRacism  
 195 Reply  
9 replies

**@Zinck82 4 years ago**  
So sad to see this kind of behavior still takes place in 2019  kick racism out!  
 22 Reply

# Data Scraping

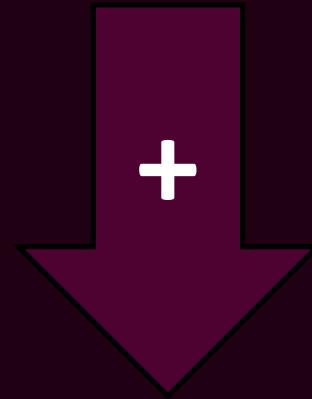
- Platforms with interaction ability (Social-Media and Blog Discussions)
- **Reddit and Instagram**
  - ~~Reddit~~ Reddit API
  - Comments from Reddit Game Threads
  - 200 - 300 Comments per Thread
- **Youtube**
  - Scraping via *GoogleAPIClient v3* Package for Python
  - Different English Soccer Game summaries
  - 100 - 200 Comments per Video
- **Kaggle**
  - Around 500 Tweets from X (Twitter)
  - Hate-Speech Content (different kinds)
  - Diversify the vocabulary of the Models



# Data Scraping



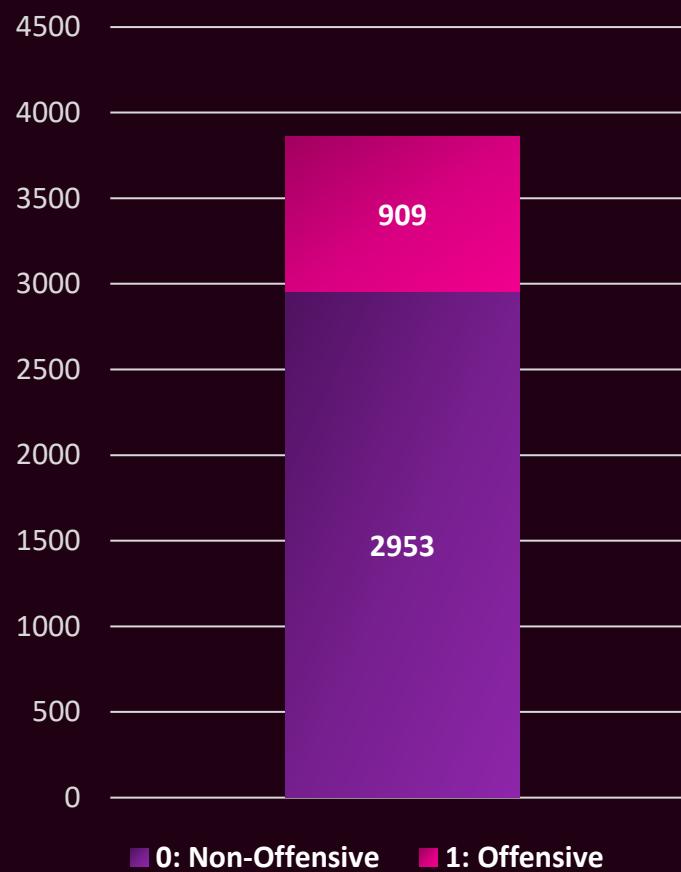
→ Over 2700 Comments  
from 17 YT-Videos, 3 Reddit Threads and 1 Kaggle Dataset



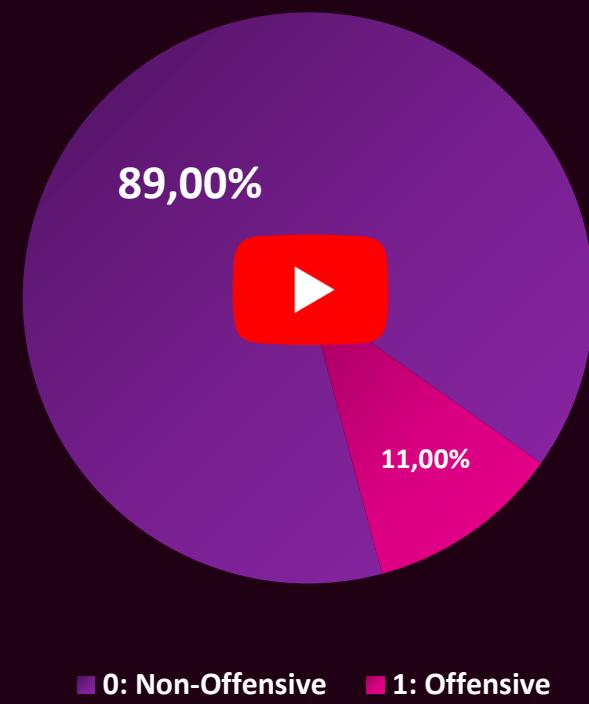
→ Nearly 4000 Comments  
from 18 YT-Videos, 14 Reddit Threads and 1 Kaggle Dataset

# Data Analysis

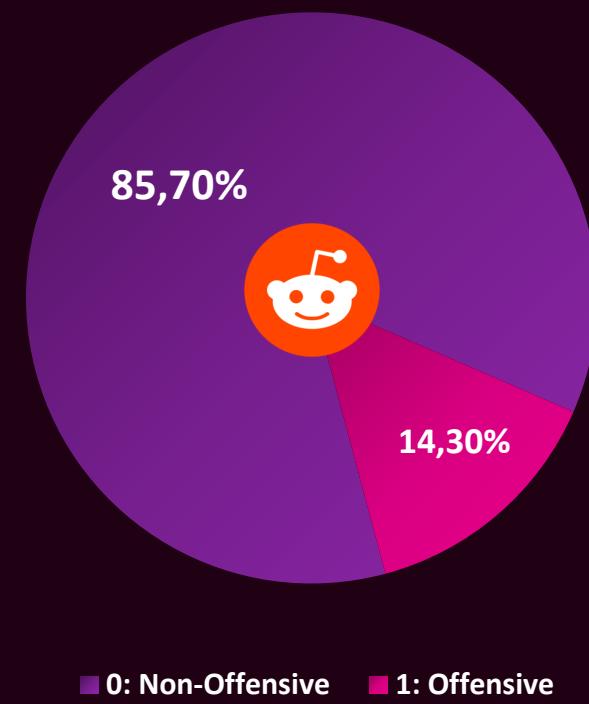
Amount of Data by Label



Label Distribution of YouTube

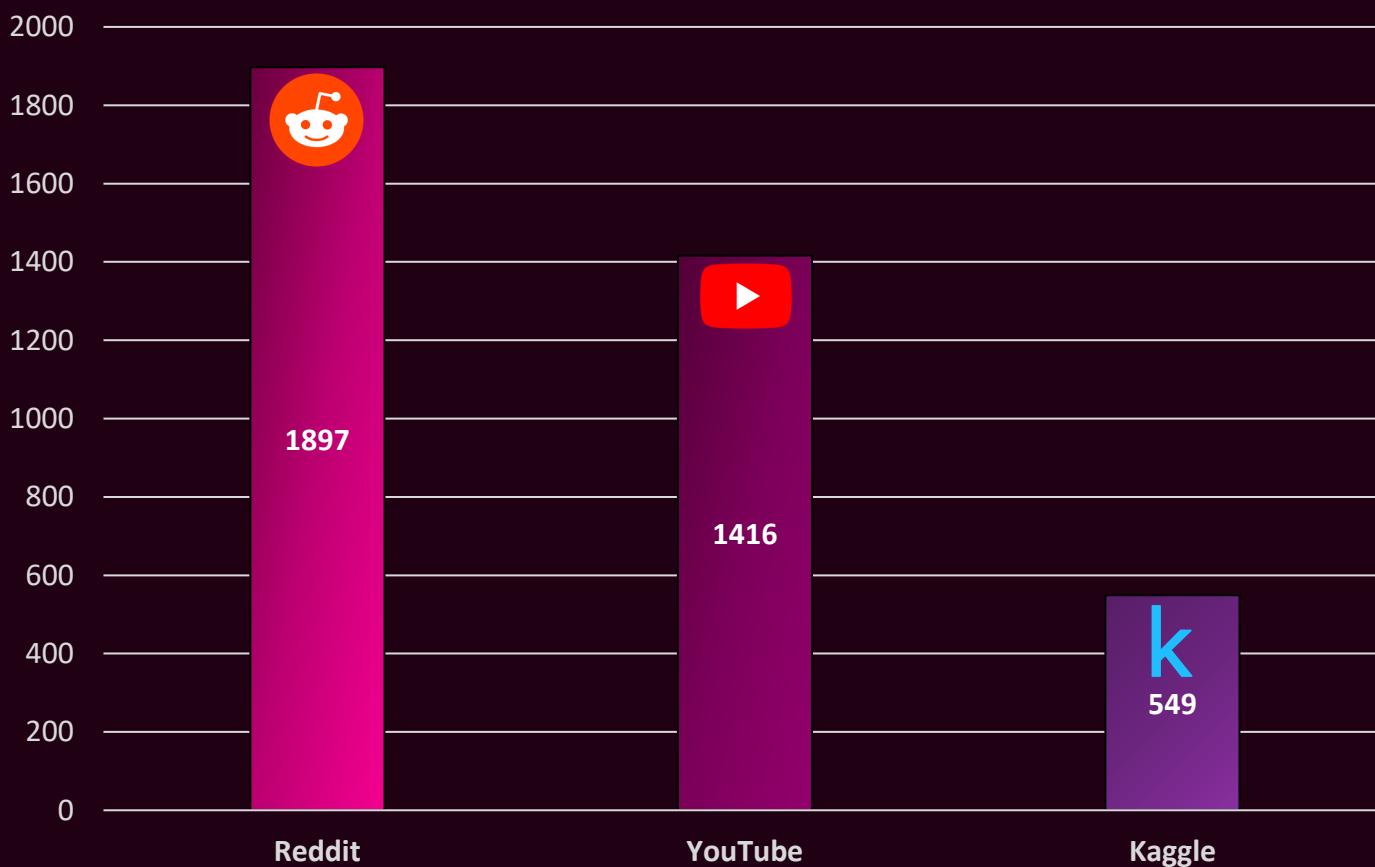


Label Distribution of Reddit

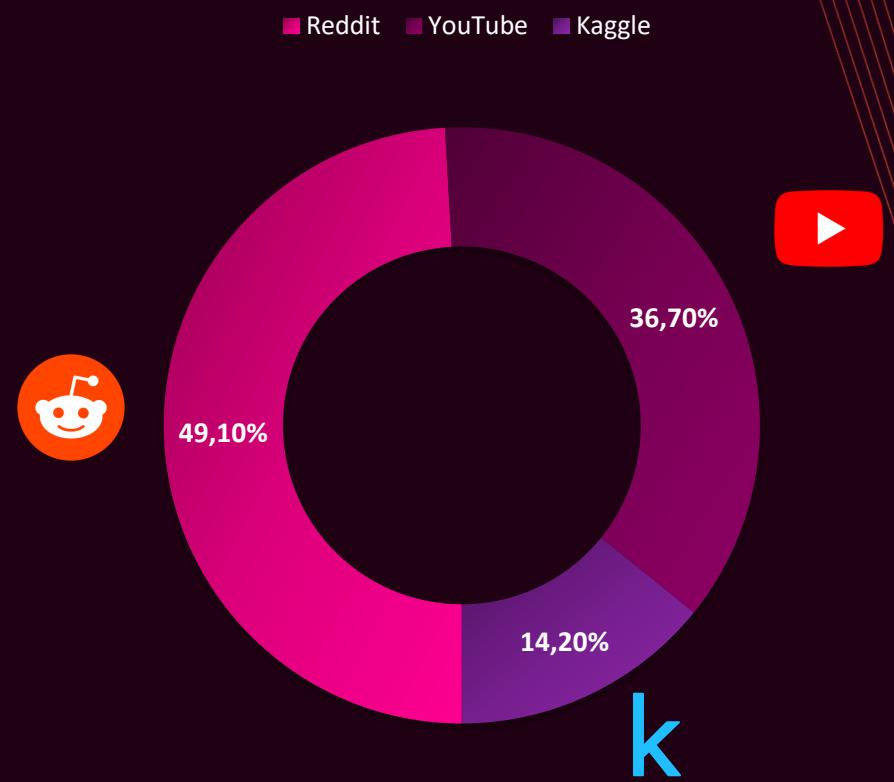


# Data Analysis

Number of Comments per Source



Distribution of the Sources



# How-to Labelling?

**Challenge: Subjectivity of offensive language**

Team Bias

1

*"Manchester United played terrible today"*

Context

2

*"This is the African national team"*

**Our Answer: 14 Clear labelling guidelines**



Racial Slurs	"You [racial slur] [nigga] are ruining the game for everyone else. Get out of here!"
Discriminatory Language	"Football isn't for [derogatory term] [monkey] like you. Go back to where you came from!"
Personal Threats	"I hope someone injures you in the next match. You deserve it."
Ethnic or National Origin Insults	"You [derogatory term] [ape] from [specific country] [niggaland] are a disgrace to the sport."

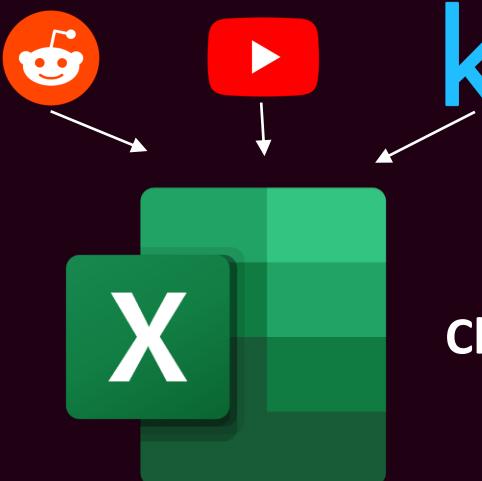
...

...

# Labelling Process

- Easy readability for comments by line
- Easy task delegation among team members
- Real-time updates and seamless collaboration by simultaneous editing

Labelling



Cleaning

- Removal of non-English comments
- Removal of irrelevant comments
- Pandas module used for easy import/export of CSV and Excel files

2657	Whatever happens, big Gareth has earned all the love in the world.2018 and 2021 weren't even tournaments we were targeting winning originally but he got the whole country hoping and believing.	reddit	0
2658	<a href="https://youtu.be/ox1rhIQ2VoU">https://youtu.be/ox1rhIQ2VoU</a>	reddit	
2659	Hoping for a great game :)Think Italy is the stronger team, but you never know.	reddit	0
2660	Did I just hear right?! Where they got fucked by Spain !	reddit	1
2661	Time for Tinnies	reddit	0
2662	Euro finals and NBA finals all in the same day Life is good	reddit	0
2663	Alexa, play Zitti e buoni	reddit	

# Data Cleaning for BERT

- Importance of emojis in social media comments for sentiment analysis

Usage of emoji library → pip install emoji



```
>>> import emoji  
>>> print(emoji.emojize('Python is :thumbs_up:'))  
Python is 👍  
>>> print(emoji.emojize('Python is :thumbsup:', language='alias'))  
Python is 👍  
>>> print(emoji.demojize('Python is 👍'))  
Python is :thumbs_up:  
>>> print(emoji.emojize("Python is fun :red_heart:"))  
Python is fun ❤
```

- 
- Check **comment length** to comply with BERT's 512 (word) token input limitation
- 

< 512

- Tokenization** via BertTokenizer.from\_pretrained function, ensuring compatibility with BERT models

- + large vocabulary
- + context-sensitive representations
- + subword tokenization

Bert Tokenizer

```
model_name = "bert-base-uncased"  
tokenizer = BertTokenizer.from_pretrained(model_name)  
model = BertForSequenceClassification.from_pretrained(model_name, num_labels=2)
```

# Data Cleaning for OLR

- Rigorous text cleaning procedures employed for dataset **reliability and precision**
- **Extraction and systematic removal of**
  - User handles
  - Hashtags
  - URLs
  - Punctuations
  - and specific numbers to prevent interference
- **Standard text preprocessing techniques** applied
  - Decapitalization
  - Tokenization
  - Removal of stopwords
  - Elimination of special characters to streamline textual data



```
Remove user handles
df.replace(r'@[^\w]*', '', regex=True, inplace=True)

# Remove hashtags
df.replace(r'#[^\w]*', '', regex=True, inplace=True)

# Remove URLs
df.replace(r'https?://[A-Za-z0-9./]+', '', regex=True, inplace=True)

# Remove punctuations
df.replace(r'[^w\s]', '', regex=True, inplace=True)

# Remove digits
df.replace(r'\d+', '', regex=True, inplace=True)

# Replace underscores with spaces
df.replace(r'_', ' ', regex=True, inplace=True)

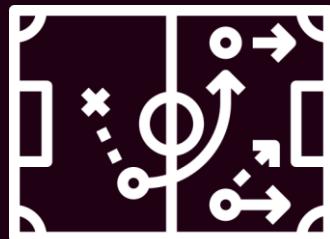
# show random 5 rows
df.sample(5, random_state=1)

# print row 663 to see the result
print(df['comment'].iloc[663])
```

# Ordinary Logistic Regression (OLR)

## Benefits of Logistic Regression:

- Ideal for binary classification tasks like offensive language detection
- Proficient in handling high-dimensional data, making it suitable for analyzing textual data
- Fast model training, compared to more complex machine learning algorithms



## Model Building:

- Train-test split (80/20) for evaluation and validation
- Feature Engineering through TF-IDF vectorization

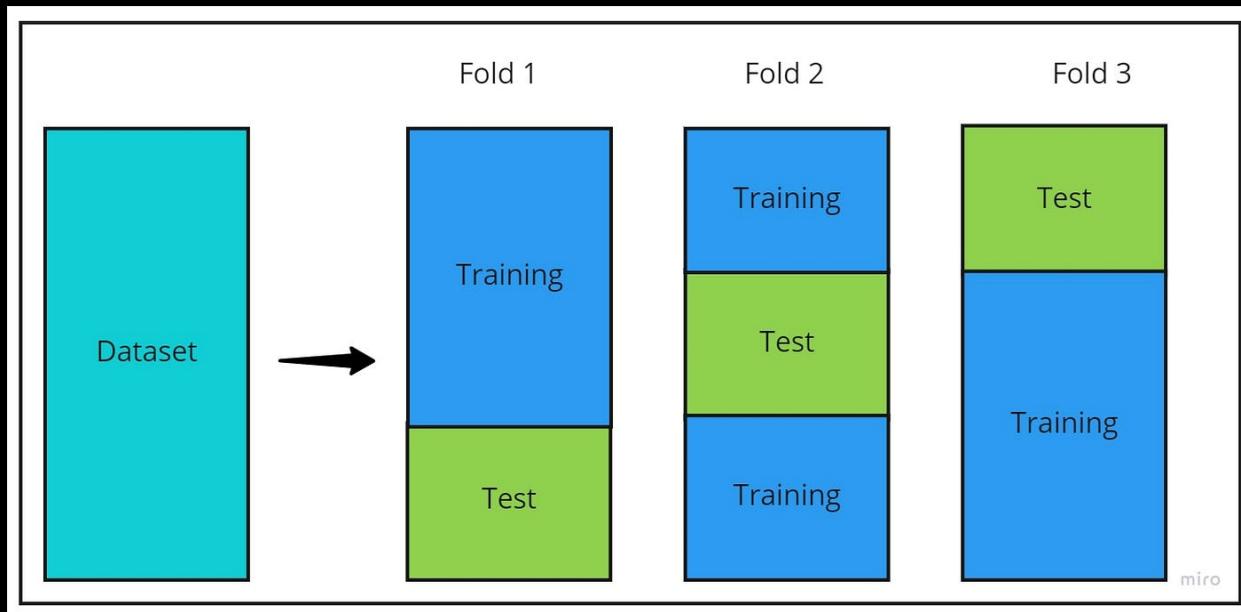
## Model Training:

- Learns to predict likelihood of comments belonging to offensive or non-offensive class
- **Regularization:** L1 (Lasso) promotes a simpler model by emphasizing only the most relevant features
- **Hyperparameter tuning:**
  - Grid search employed, optimizing model performance with parameter C for regularization strength
  - Using `.best_params_` most suitable C extracted and applied

```
#instantiate the best model
logreg = LogisticRegression(C=10, class_weight={0: 1.0, 1: 13}, penalty='l1', solver='liblinear')
```



# BERT



## K-Fold Cross Check Validation

- Better use of data
- More available metrics
- Detection of overfitting

# First Run

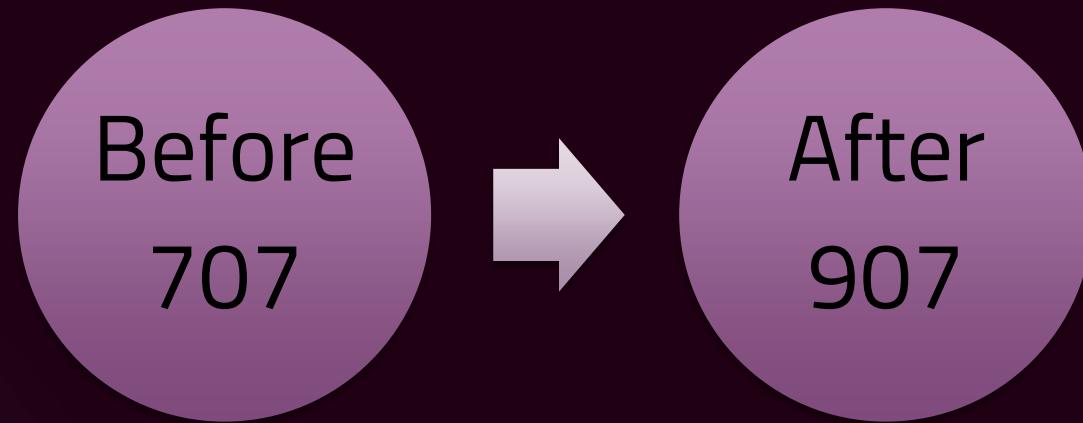
	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
<b>Training 1</b>	0,910	0,861	0,788	0,823
<b>Training 2</b>	0,958	0,922	0,934	0,928
<b>Training 3</b>	0,904	0,857	0,822	0,839
<b>Average</b>	<b>0,924</b>	<b>0,880</b>	<b>0,848</b>	<b>0,863</b>

# Problem: False Negatives

- ✖ *why dont you just go back to where you came from*
- ✖ *the more i watch the more i want to kill*
- ✖ *man this was a shitty team*
- ✖ *did you see those untalented bitches*
- ✖ *i gotta say that I hope your career crashes and burns. You don't deserve success*

		Predicted Value	
		Positive	Negative
Actual Value	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

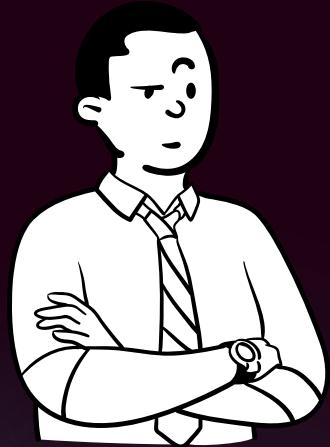
# Solution: More Data



# Second Run

	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
<b>Training 1</b>	0,898	0,877	0,671	0,760
<b>Training 2</b>	0,929	0,813	0,912	0,860
<b>Training 3</b>	0,987	0,966	0,980	0,973
<b>Average</b>	<b>0,938</b>	<b>0,885</b>	<b>0,854</b>	<b>0,864</b>
<b>Average (First Run)</b>	<b>0,924</b>	<b>0,880</b>	<b>0,848</b>	<b>0,863</b>

# Problem: False Negatives



- ✖ *why dont you just go back to where you came from*
- ✖ *the more i watch the more i want to kill*
- ✖ *man this was a shitty team*
- ✖ *did you see those untalented bitches*
- ✖ *i gotta say that I hope your career crashes and burns. You don't deserve success*

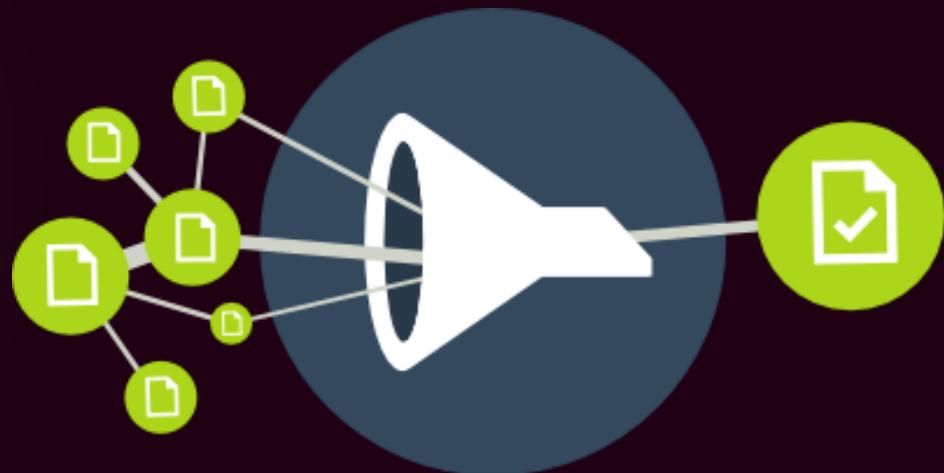
	Accuracy	Precision	Recall	F1
Average	0,849	0,667	0,500	0,571

# Takeaway: Data understanding is important!



**Model was trained with filtered data.**

**But was tested with unfiltered data.**



# Final Results

	Accuracy	Precision	Recall	F1
Average (Unfiltered Data)	0,849	0,667	0,500	0,571
Average (Filtered Data)	0,982	0,875	0,875	0,874

# Good-Better-Best

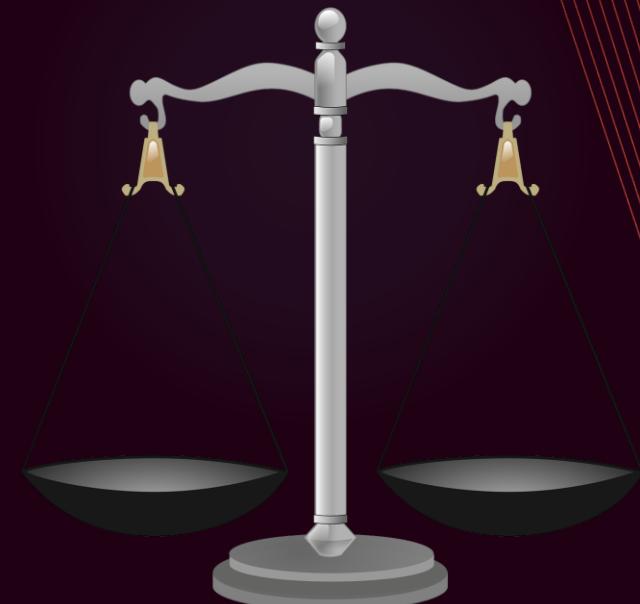
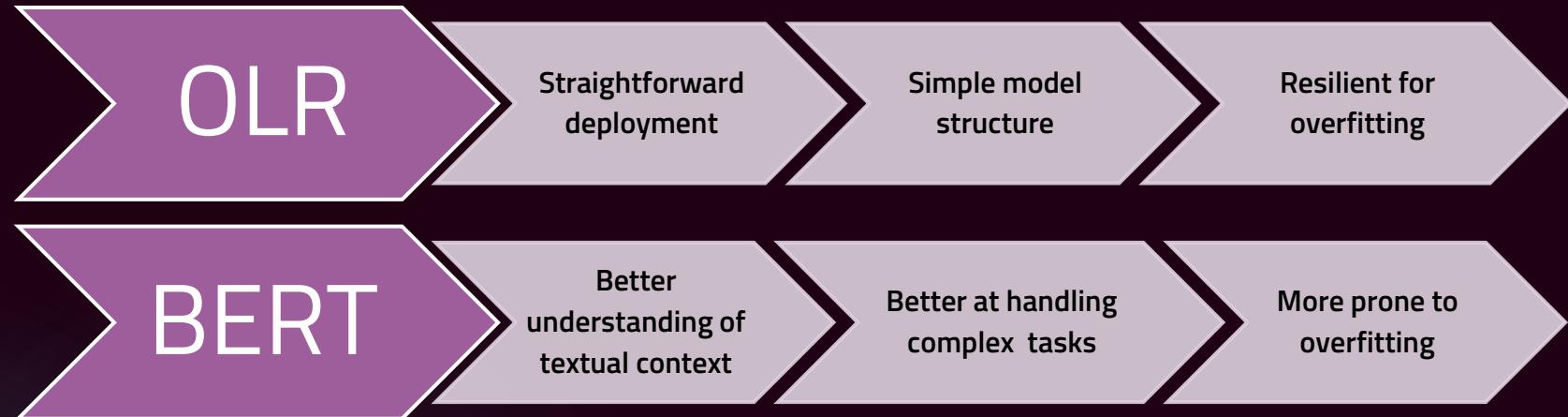
Content based improvement

- "Africa vs africa"

Detecting nuances

- "*Jude fucking Bellingham.... What a player!!! ❤️*"

# Comparison



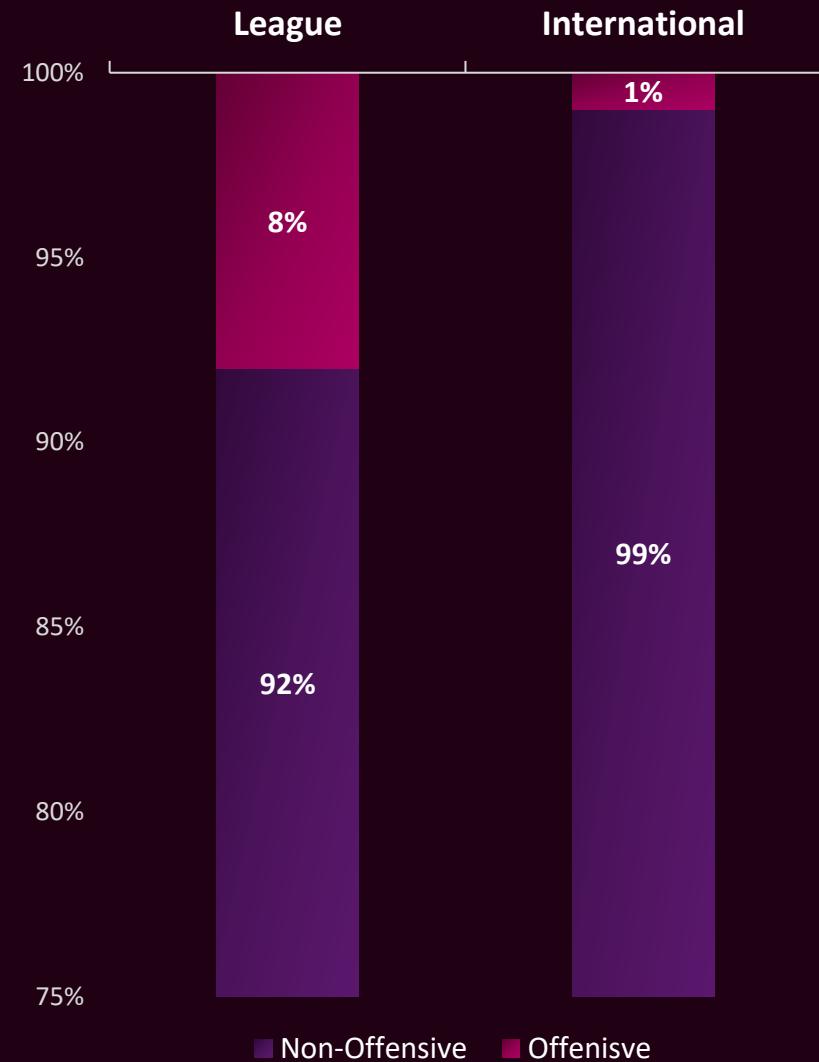
# Modelapplication and Evaluation

- Extensive preprocessing and clear labeling guidelines were applied to the collected data to ensure compatibility with machine learning algorithms.
- Obtaining results from the BERT and OLR models that we have applied in accordance with the purpose of our project.
- 1700 YouTube comments to Label from League and International Matches

# Results of the BERT Model on Data

Our analysis provided valuable insights into the distribution of offensive language.

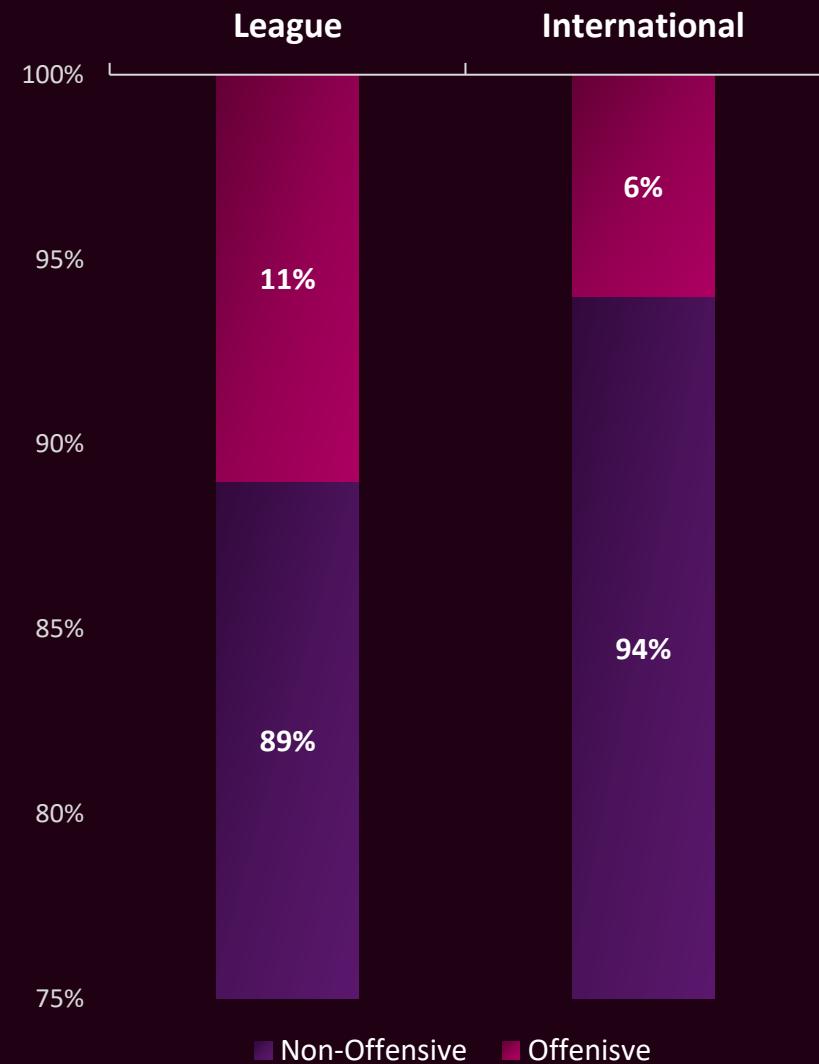
- 1% International Match
- 8% League Matches



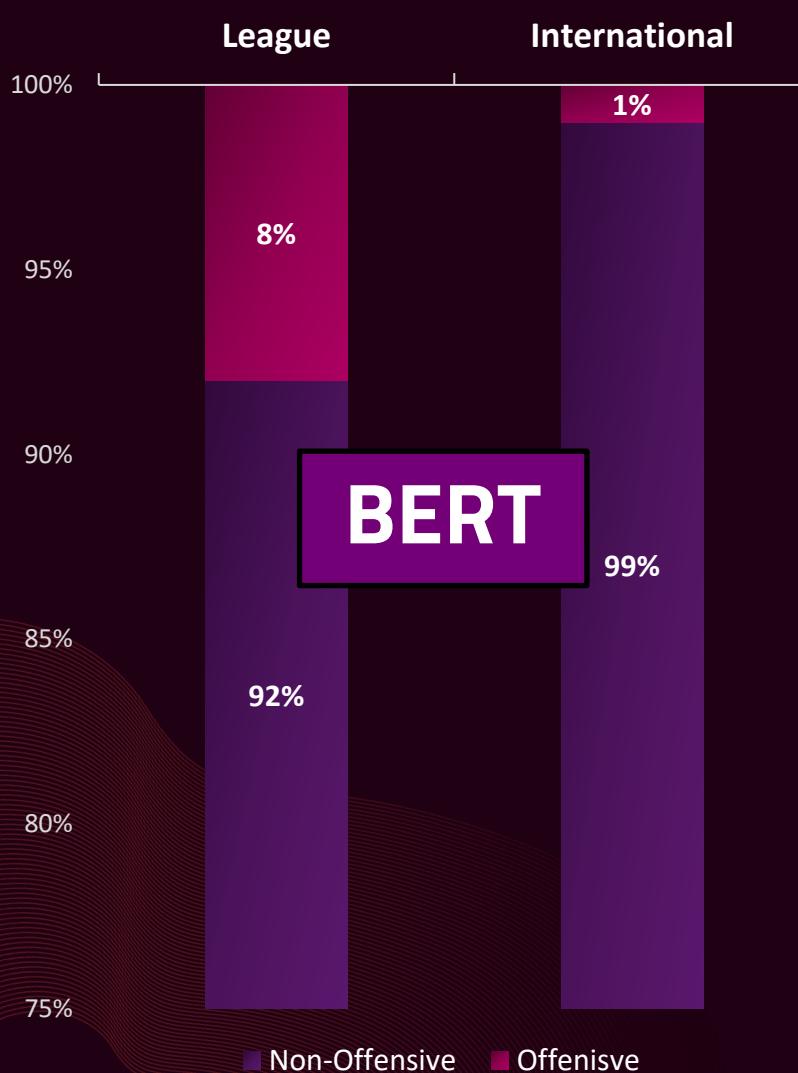
# Results of the OLR Model on Data

League-related comments have a higher count of offensive language, with 81 comments labeled as offensive

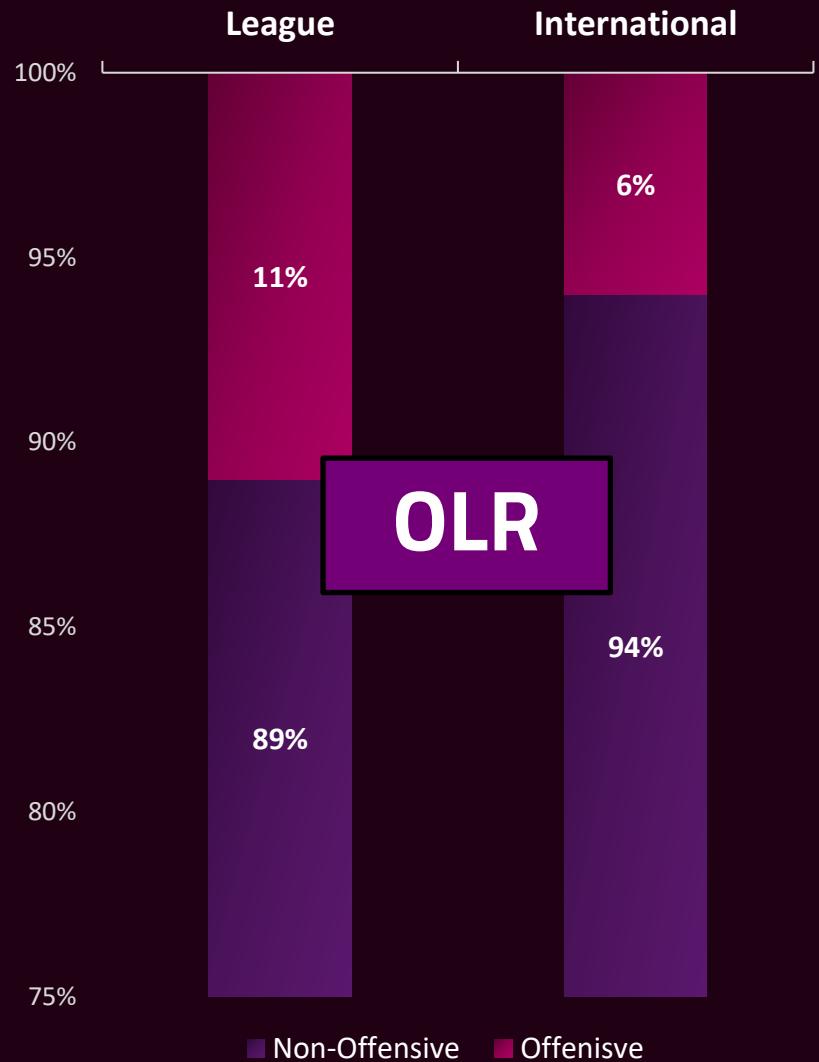
- 5,7% International Match
- 11,1% League Matches



# Comparison of the Results



- OLR predicts Offensive Comments better
- League Comments are more often Offensive



# Summary



# Limitations

## 1. Internal Validity Limitation:

- The dataset size (4000 comments)
- Data quantity (Vocabulary / Number of Sources)

## 2. External Validity Limitation:

- The model only identifies hate speech written in English
- Ignoring linguistic variations

## Overall Project Goal Alignment:

- Despite the limitations, the assessment underscores the importance of fostering a welcoming and constructive online community among football fans.
- The models play a crucial role in promoting civil discourse and enhancing football fans' online experience by effectively detecting offensive language.
- Continuous improvement and optimization are essential to ensure the model's sustained efficacy and alignment with the project's main goals.



GITHUB REPOSITORY:



[https://github.com/prizyou/NLP4B\\_Football](https://github.com/prizyou/NLP4B_Football)



GitHub

# Thank you!

Feel free to ask questions!