YouTube Video Classification on Twitter

Giuliano Di Giuseppe, Raffaele Russo, Alessandro Vanacore

Progetto BDE Vincenzo Moscato

2022/2023

Outline

- Introduction
- Dataset
- Analysis
- Data Management
- Models
- Conclusion

Introduction - Context

- Social media platforms shape the digital information landscape and foster discussions.
- Unrestricted expression on these platforms poses risks.
- Facebook and Twitter have implemented moderation interventions.
- Moderation interventions address inappropriate content and user accountability.
- They include content filtering, reporting systems, and community guidelines.
- AI and human moderation teams play a role.
- Balancing freedom of expression and content control is challenging.
- Moderation interventions are crucial for a healthier online discourse.

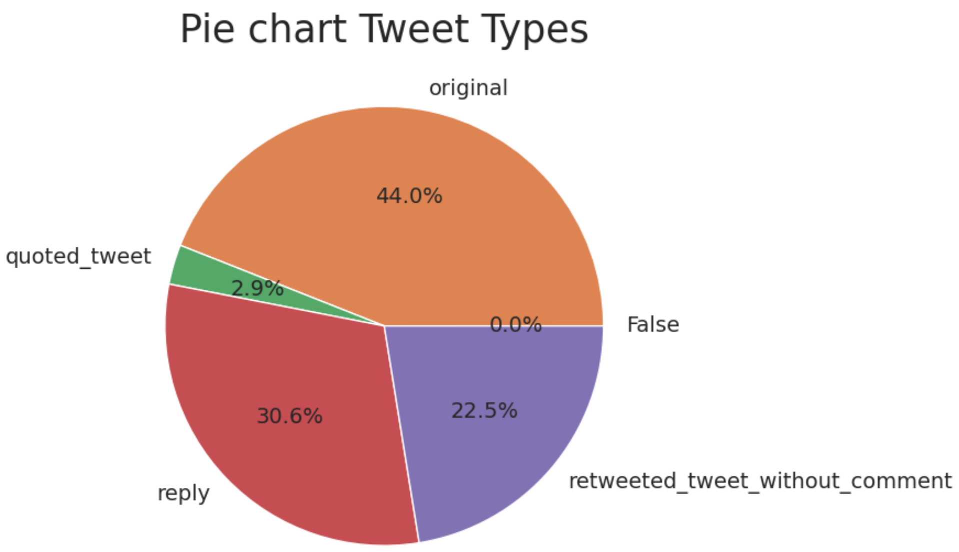
Dataset

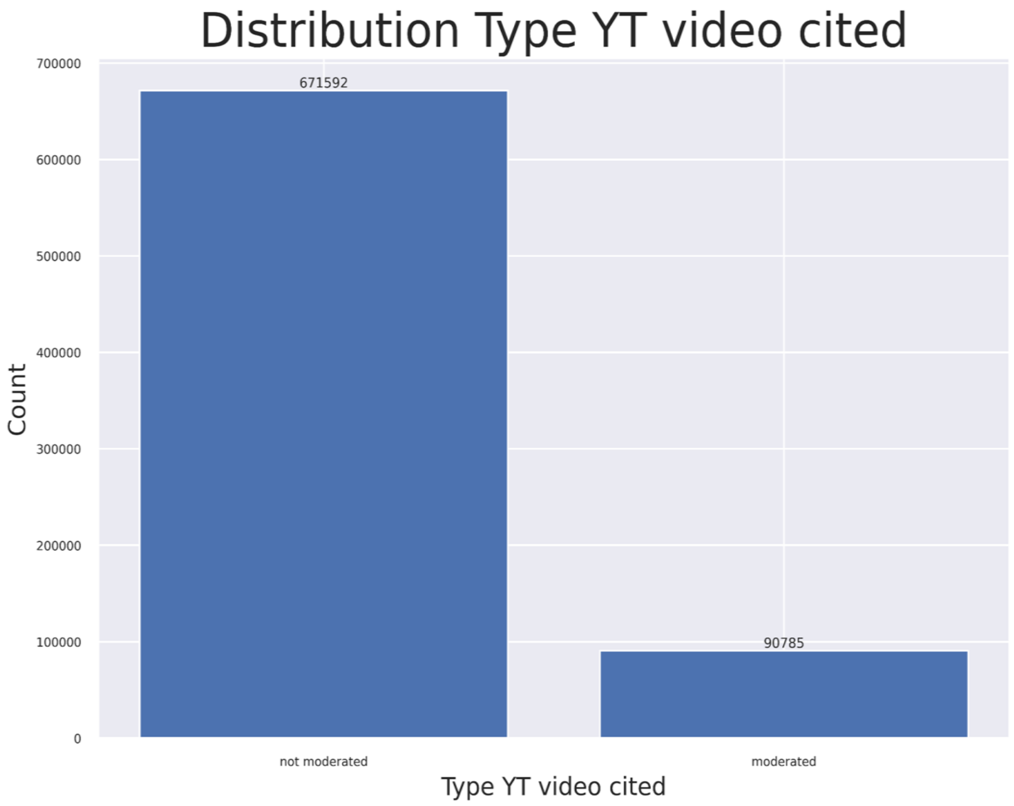
- train.csv: a table with three columns, i.e., the video identifier, the video URL(s) (the same video can be shared through different URLs), the video status (the video could either moderated or not moderated).
- test.csv: it shares the same structure of train.csv without the last column.
- df youtube.csv.zip: the archive with all tweets sharing the videos listed in the above-mentioned files.

Dataset

- train_csv has 78 column, example of rappresentative column
  - _id: Unique identifier of the tweet in the dataset.
  - tweetid: Unique identifier of the tweet on Twitter.
  - userid: Unique identifier of the user who posted the tweet.
  - description: User's description.
  - place_id: Identifier of the place associated with the tweet.
  - tweet_type: Type of tweet.
  - friends_count: Number of user's friends.
  - listed_count: Number of users who have added the user to their lists.
  - followers_count: Number of user's followers.
  - statuses_count: Number of tweets published by the user.
  - verified: Account authentication information.
  - hashtag: Hashtags used in the tweet.
  - date_first_tweet: Date of the user's first tweet.
  - account_creation_date: Account creation date.
  - rt_hashtag: Hashtags present in the retweet.
  - qtd_rt_count: Count of retweets of the mentioned tweet.
  - qtd_reply_count: Count of replies to the mentioned tweet.

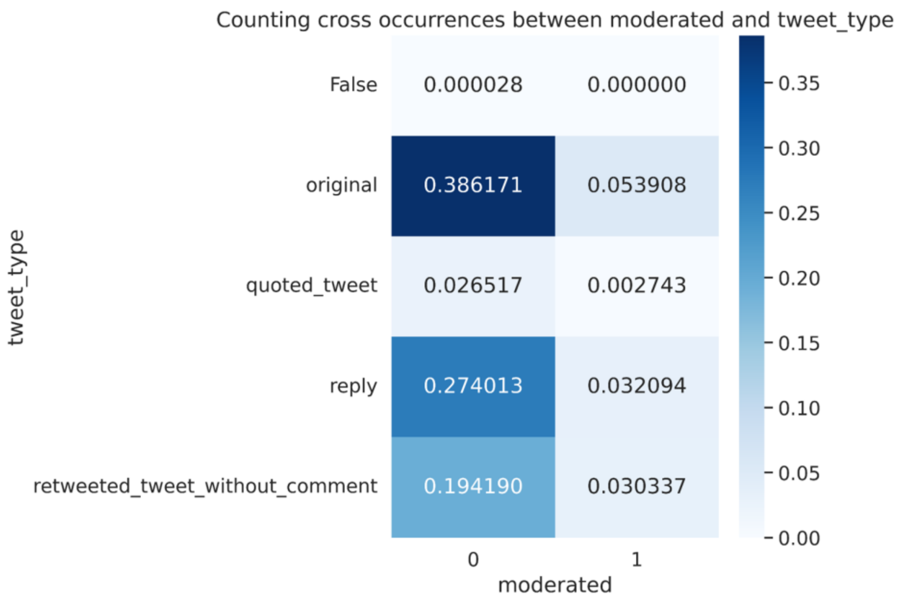
Data Analysis





Data Analysis

Type YT video cited	Count	avg follower	avg favourites	avg acc_age	avg listed	number of user
not moderated	671592	3678,798218	21279,42023	1884,012024	25,63432709	671572
moderated	90785	2567,31177	19247,90457	1742,868723	12,39711406	90785



MongoDB

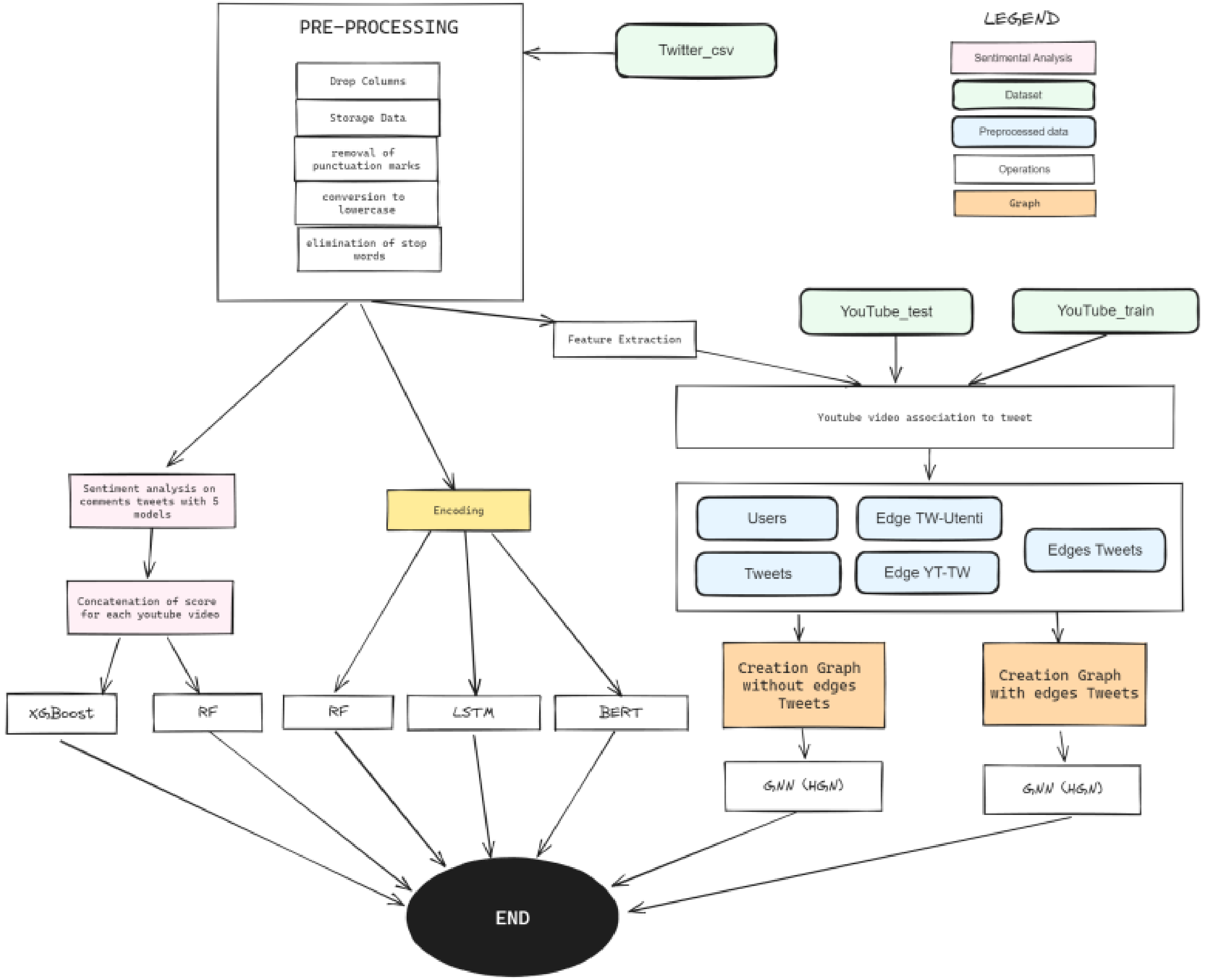
- To handle and preprocess our large dataset
- To execute complex querying operations efficiently.
  - Mapped each YouTube video to its corresponding list of tweets
- After mapping the YouTube videos to their respective tweets, this information led to a natural creation of a graph in Python.
- This approach was preferred over Neo4j

Pre-Processing

- Storage data in MongoDB
- Encoding ‘moderationStatus’
- Filtering tweets based on tweet types
- Dropping insignificant columns
- Create collection of User, tweet , videoYT
- Association videoYT to tweets
- Pre-processing test (drop stopwords, punctuation removal , tokenization)
- Generate sentimental analysis by 5 model with *TweetNLP* for all text
- Embedding text

Models

- XGBoost
- SVM
- Random Forest
- Improved Random Forest
- LSTM
- BERT
- GNN without tweet edges
- GNN with tweet edges



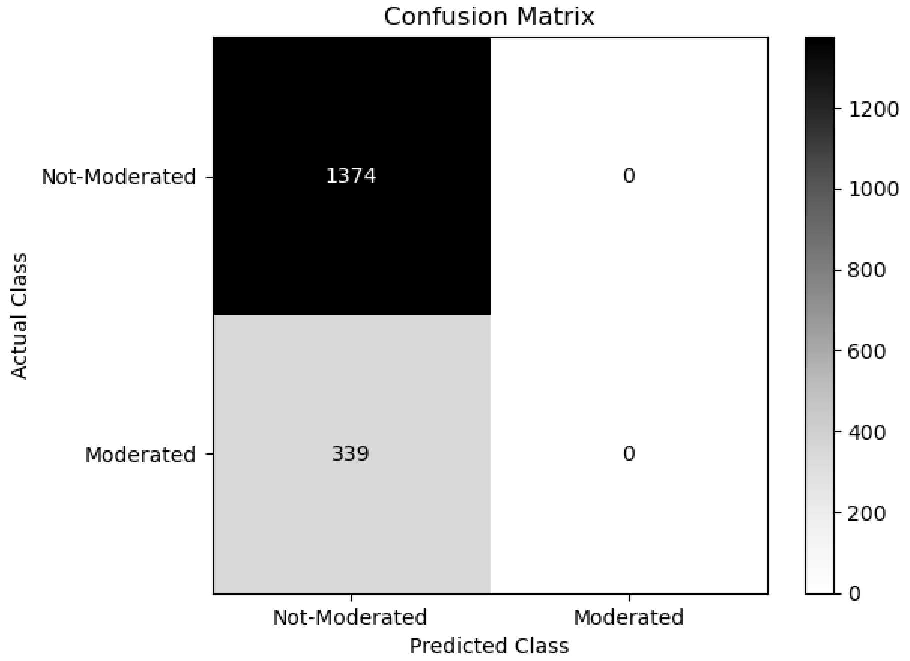
Models - XGBoost

Table 1: Numbers and Meanings

Class	F1-Score	Precision	Recall
Not-moderated	1.0	1.0	1.0
Moderated	0.0	0.0	0.0

Table 2: Numbers and Meanings

Accuracy	Macro F1-Score	Precision	Recall
80.21%	0.4451	0.0	1.0



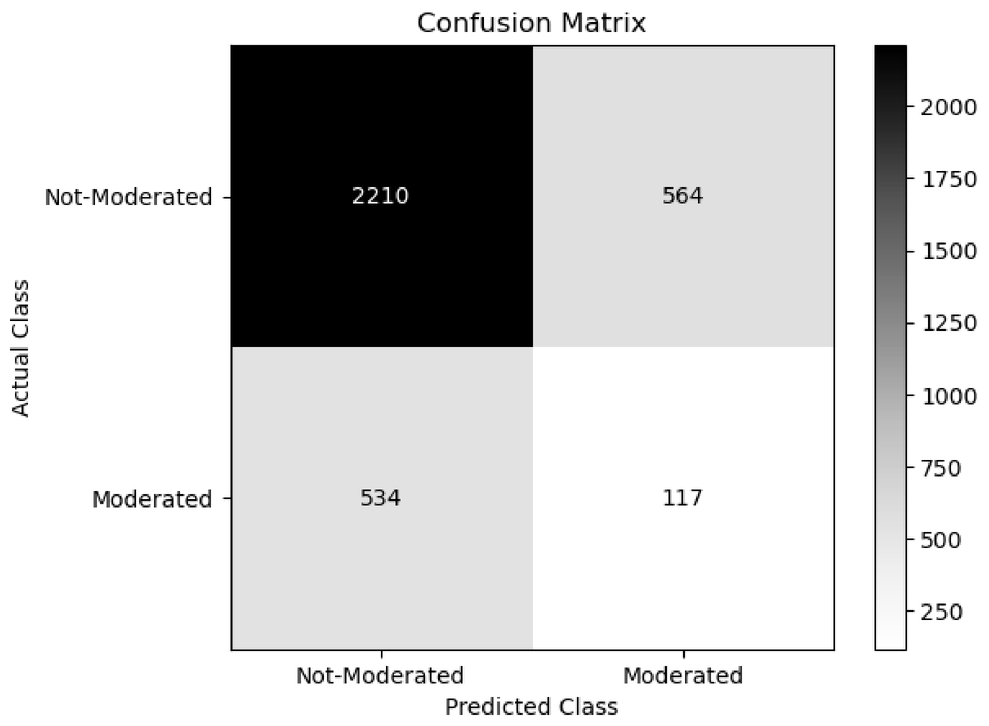
Models - SVM

Table 3: Numbers and Meanings

Class	F1-Score	Precision	Recall
Not-moderated	0.8011	0.8056	0.7966
Moderated	0.1752	0.1712	0.1792

Table 4: Numbers and Meanings

Accuracy	Macro F1-Score	Precision	Recall
67.94%	0.4883	0.1718	0.1797



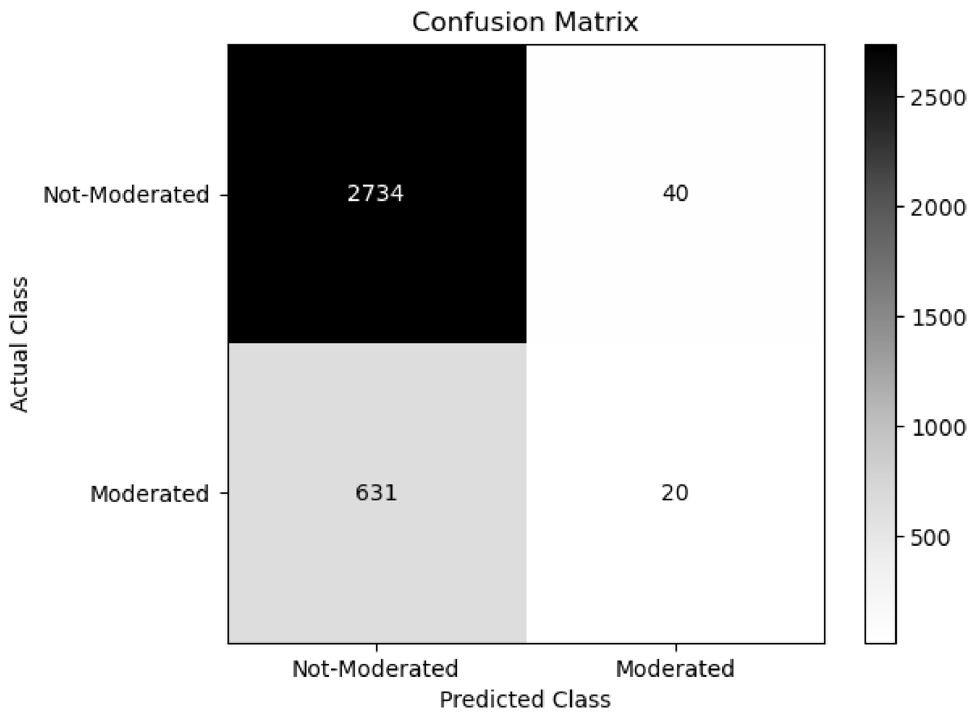
Models – Random Forest

Table 5: Numbers and Meanings

Class	F1-Score	Precision	Recall
Not-moderated	0.8912	0.8125	0.9855
Moderated	0.0562	0.3333	0.0309

Table 6: Numbers and Meanings

Accuracy	Macro F1-Score	Precision	Recall
80.41%	0.47347	0.3333	0.03072



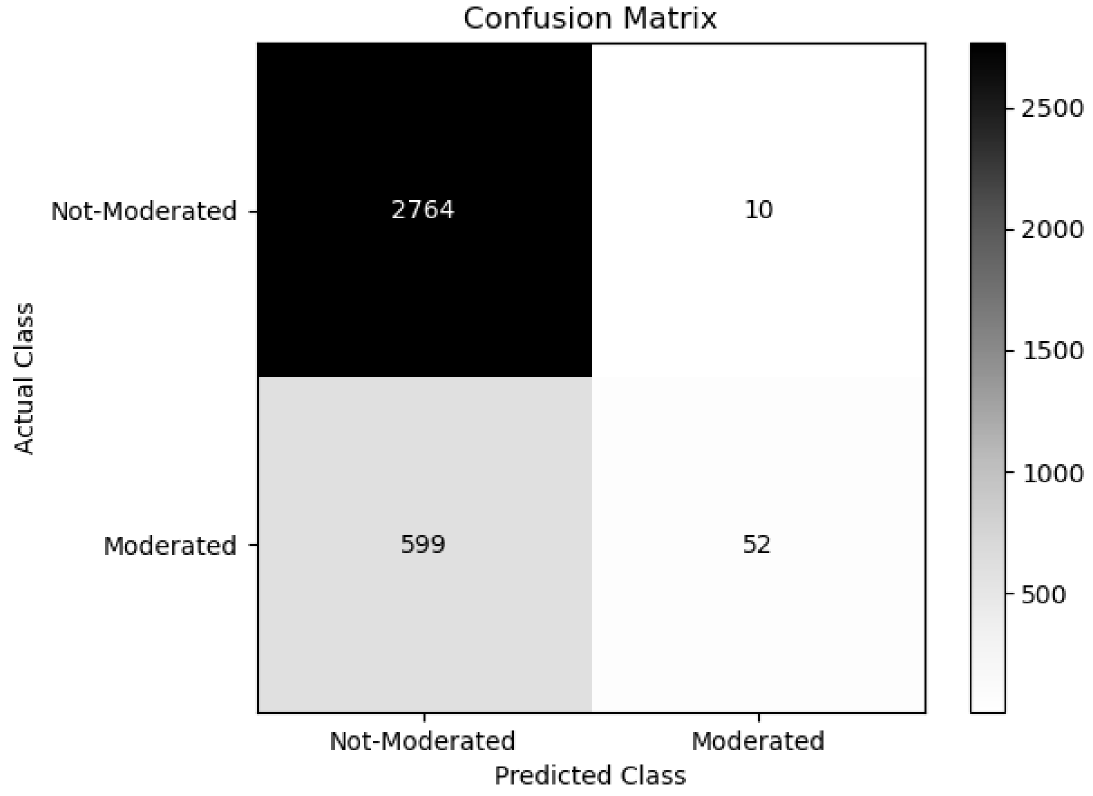
Models – Improved Random Forest

Table 7: Numbers and Meanings

Class	F1-Score	Precision	Recall
Not-moderated	0.9001	0.9964	0.8210
Moderated	0.1442	0.8387	0.0795

Table 8: Numbers and Meanings

Accuracy	Macro F1-Score	Precision	Recall
82.22%	0.5233	0.8387	0.07987



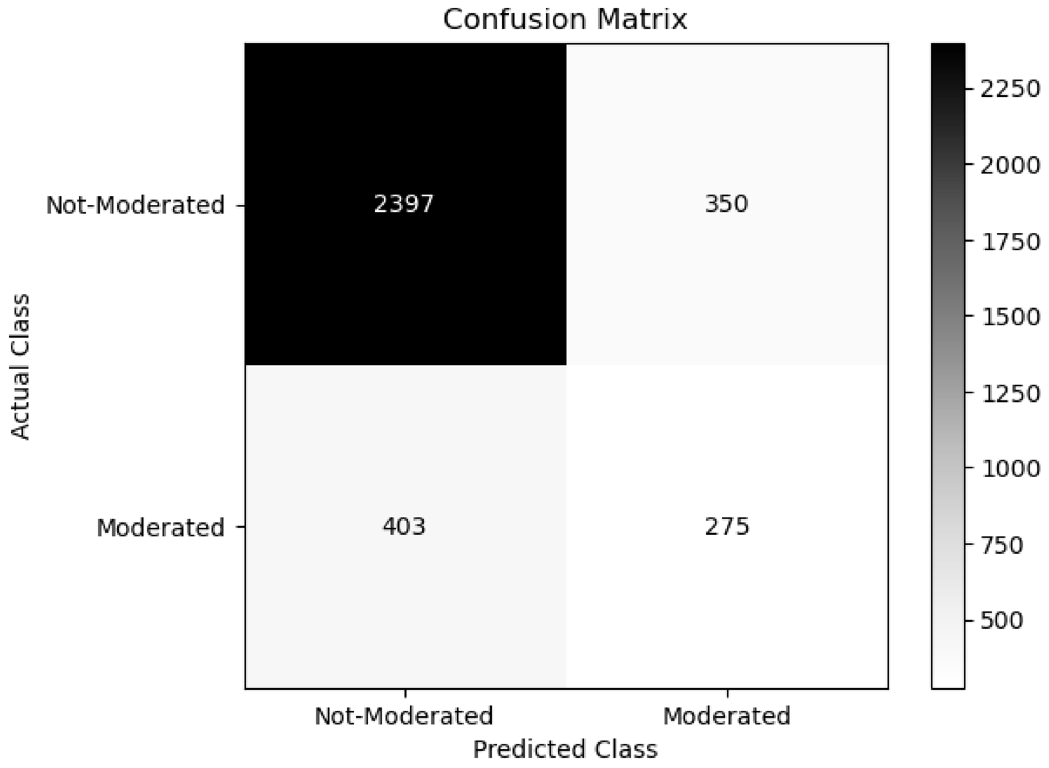
Models - LSTM

Table 9: Numbers and Meanings

Class	F1-Score	Precision	Recall
Not-moderated	0.8639	0.8557	0.8723
Moderated	0.4219	0.4390	0.4058

Table 10: Numbers and Meanings

Accuracy	Macro F1-Score	Precision	Recall
78.01%	0.64317	0.4056	0.44



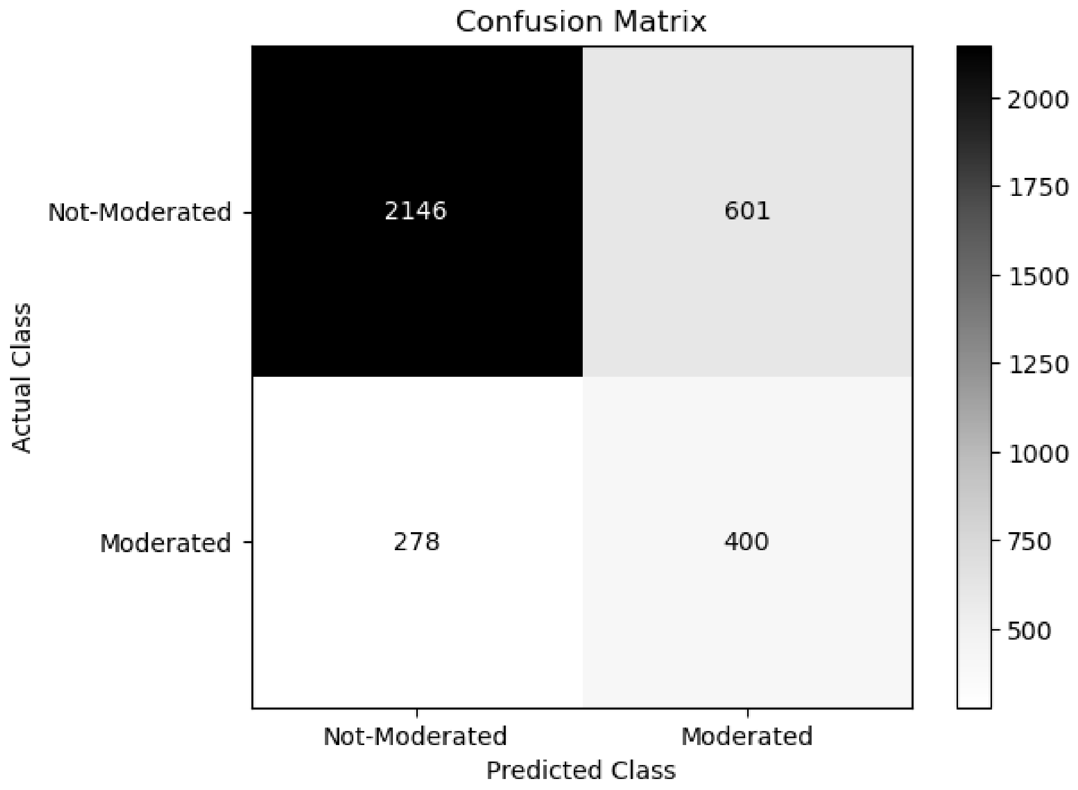
Models - BERT

Table 11: Numbers and Meanings

Class	F1-Score	Precision	Recall
Not-moderated	0.8301	0.8857	0.7811
Moderated	0.4791	0.3996	0.5893

Table 12: Numbers and Meanings

Accuracy	Macro F1-Score	Precision	Recall
74.34%	0.65324	0.39960	0.58997

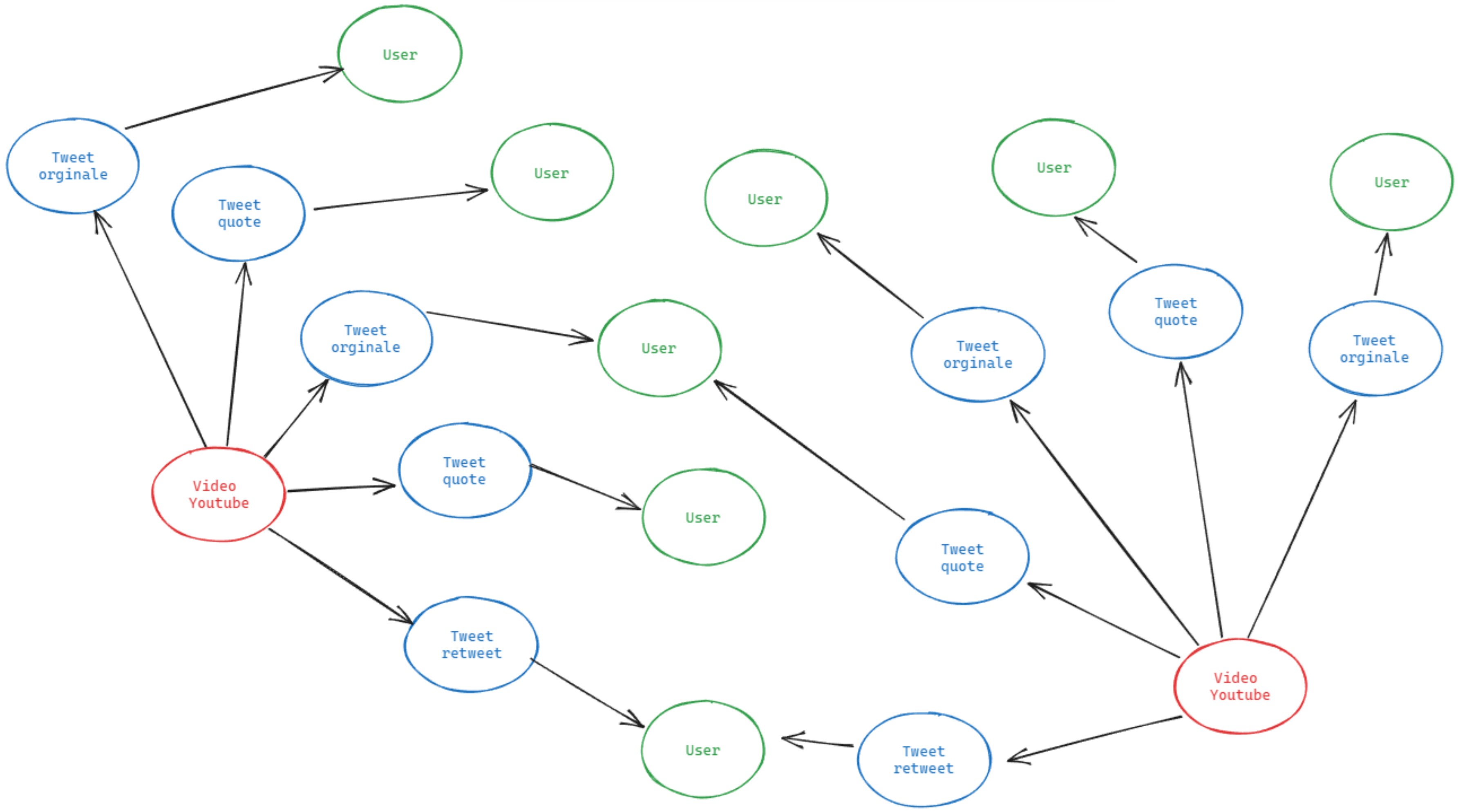


Models – Graph Neural Network (Code)

```
class HGT(torch.nn.Module):
    def __init__(self, data,hidden_channels, num_heads, num_layers):
    super().__init__()
    self.lin_dict = torch.nn.ModuleDict()
    for node_type in data.node_types:
        self.lin_dict[node_type] = Linear(-1, hidden_channels)
    self.convs = torch.nn.ModuleList()
    for _ in range(num_layers):
        conv = HGTConv(hidden_channels, hidden_channels,data.metadata(),num_heads, group='sum')
        self.convs.append(conv)
    self.lin_label_prediction = nn.Linear(hidden_channels, 1)
    self.sigmoid = nn.Sigmoid()
    self.drop_out = nn.Dropout(p=0.5)

def forward(self, x_dict, edge_index_dict):
    for node_type, x in x_dict.items():
        x_dict[node_type] = self.lin_dict[node_type](x).relu_()
    for conv in self.convs:
        x_dict = conv(x_dict, edge_index_dict)
    yt_x = x_dict['yt']
    yt_x = self.drop_out(yt_x)
    yt_logits = self.lin_label_prediction(yt_x)
    y_pred = self.sigmoid(yt_logits).squeeze()
    return y_pred
```

Models – GNN without tweet edges (Structure)



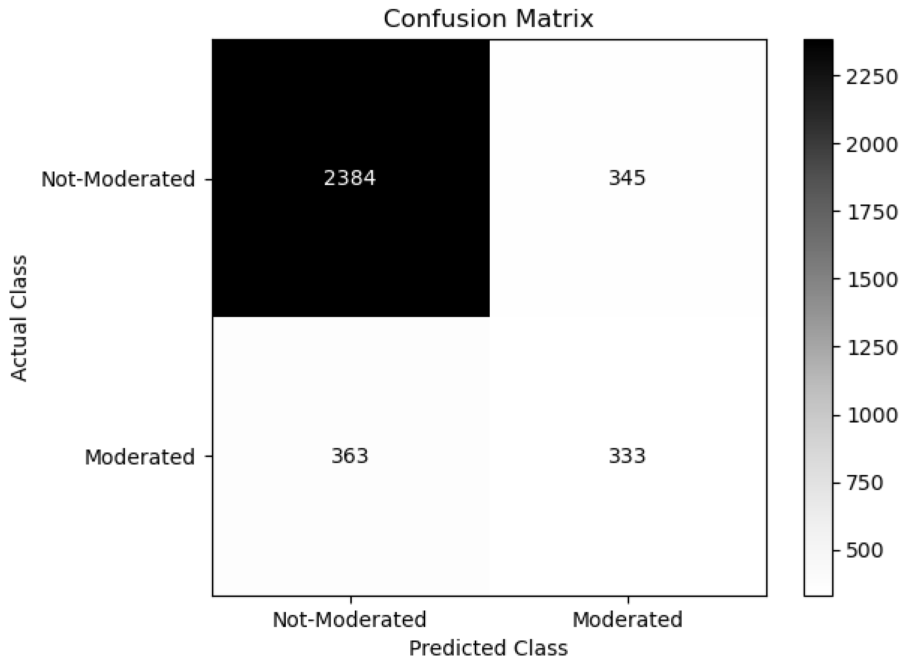
Models – GNN without tweet edges (Results)

Table 13: Numbers and Meanings

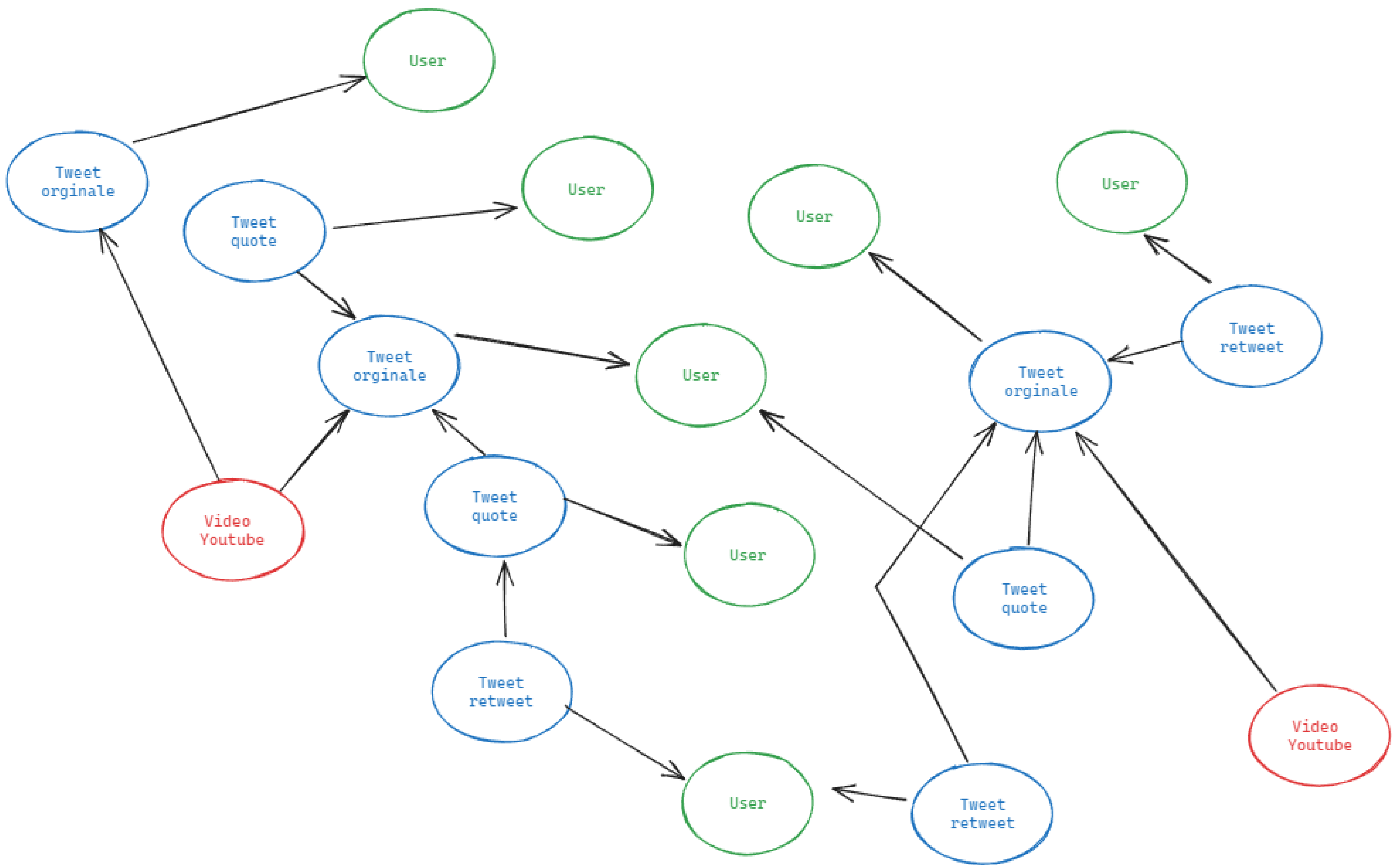
Class	F1-Score	Precision	Recall
Not-moderated	0.8685	0.8847	0.8529
Moderated	0.4115	0.3808	0.4486

Table 14: Numbers and Meanings

Accuracy	Macro F1-Score	Precision	Recall
78.48%	0.64	0.3805	0.4486



Models – GNN with tweet edges (Structure)



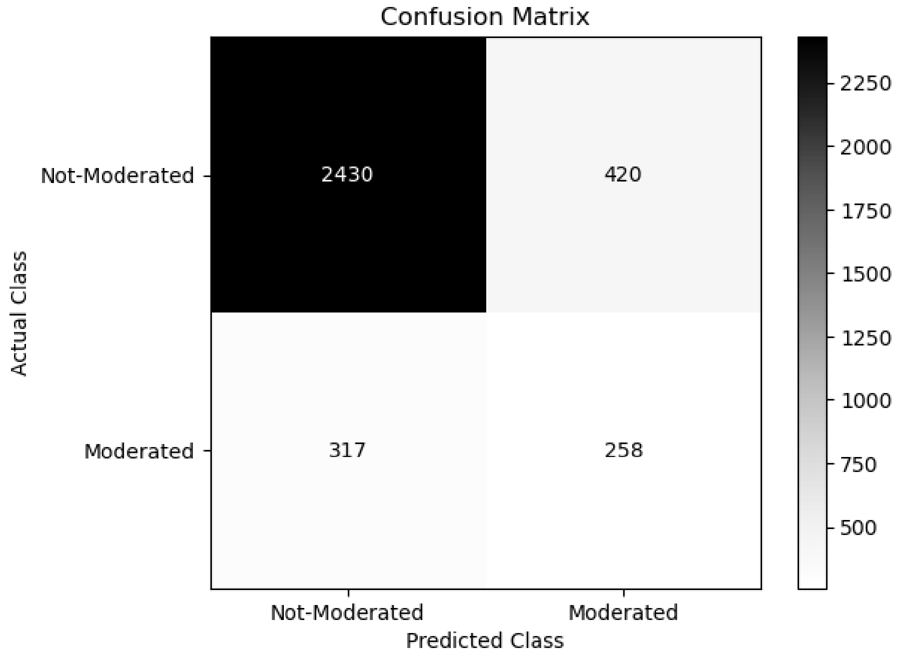
Models – GNN with tweet edges (Results)

Table 15: Numbers and Meanings

Class	F1-Score	Precision	Recall
Not-moderated	0.8706	0.8673	0.8739
Moderated	0.4851	0.4786	0.4919

Table 16: Numbers and Meanings

Accuracy	Macro F1-Score	Precision	Recall
79.33%	0.67771	0.49115	0.47844



Conclusions

- While dealing with skewness in the dataset presents a significant challenge, it is not insurmountable.
- However, despite this challenge, graph modeling has consistently demonstrated its effectiveness in producing the best results.
- Moreover, there is exciting potential for further improvement by exploring new relationships and architectures.
- By continuously refining and expanding our understanding, we can push the boundaries of performance even further and unlock new possibilities in data analysis and modeling.

Thank YOU