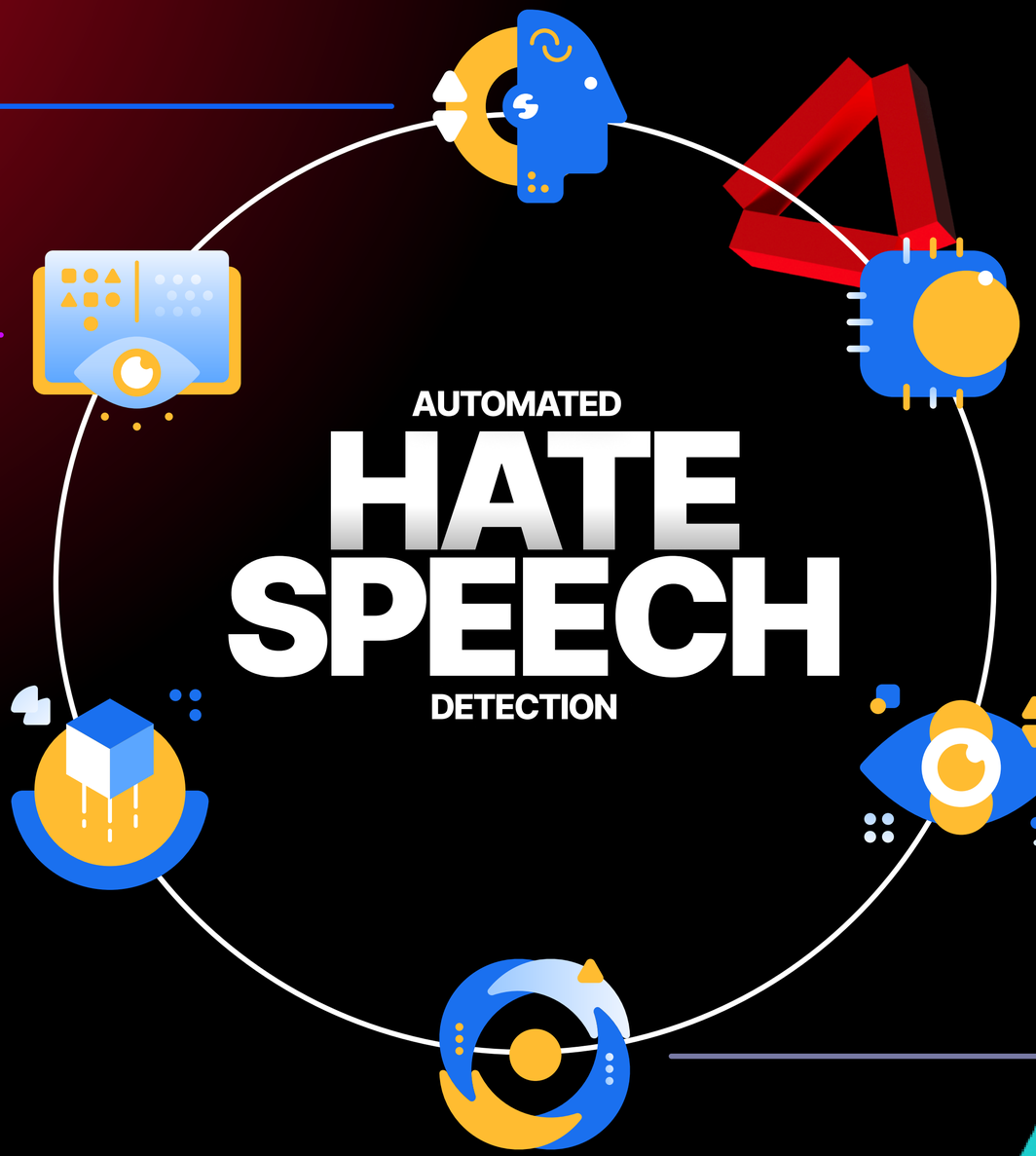
Agenda

BBS NATURAL LANGUAGE PROCESSING

1 Business Understanding

2 Data Understanding

3 Data Preparation



4 Modelling

5 Features

6 Conclusions

Business Understanding

BBS NATURAL LANGUAGE PROCESSING

A biggest challenge for automated hate speech detection on social media is the separation of hate speech from other instances of offensive language.

There are important qualitative differences between different types of potentially abusive language that need to be considered.

For example, a tweet quoting rap lyrics that contain potentially racist or sexist terms should not be regarded in the same way as a tweet that directs racist slurs at another user.

Black people like @ChiefKeef disgust me laughing at someone who died dumb monkey ass n*ggas like you need to be locked up just for stupidity "We hate n*ggers, we hate f*ggots and we hate sp*cs"-kkk rally	@bonoxx haha b*tch ima draw a webb in bullets round yo fuggin forehead #BeamThat b*tch i know where you reside You a h*e if you crying because yo *ss pregnant. It was all smiles while you was riding that d*ck b*tch.	The rich kids of beverly hills is trash and not even entertaining trash. And why do they all look mid thirties? Already more yellows than the last 2 years combined #IndyCar #Honda200
Got ya crackas, sp*cs, ch*nks, and carpet riders but no n*ggas MLK ain't fight for nothing #GimmeABlackEmoji @rad_daddy14 @lilJohn_Flw6 and dis wus after i offered to help f*ggot *ss n*gga get custody of his kids.... Im lowkey heated..	Big b**bs and big *ss = ugly face Big b**bs , big *ss , pretty face = h*e. RT @Hebesarcastic: Math city b*tch, math math city b*tch, math city b*tch, math math city b*tch, 10, 10, 10 and 20 equals 50 b*tch	@LifeAsBros: Charlie Sheen is too real http://t.co/gGGdK3kOV7" major f*cking respect for Charlie Sheen. RT @MacNarse: Yes, the bird tweets in the morning are lovely but remember, what you're hearing is: BIRD C*CK HERE! GET SOME ROCK HARD BIRD
Viola, Auntie...you can't call White trash women White trash like this during primetime! #HTGAWM It's stuff like this that makes black people look r*tarded. It's not expressing yourself its stupidity http://t.co/MxbGfPvICF	RT @DntWearCondom: If u acted like a h*e after we broke up I'm not wrong for thinking u were a h*e all along @EdwardPxr: Overdosing on heavy drugs doesn't sound bad tonight." I do that p*ssy shit every day.	cleaned all the bird nests out of t he downspouts and ready for mo re rain, the baby birds are out no w. respect the birds.....and bees (: This recipe calls for a leftover brownie. What the hell is a LEFTOVER brownie? That's just a brownie you haven't eaten yet."- @EricNa29

Data Understanding

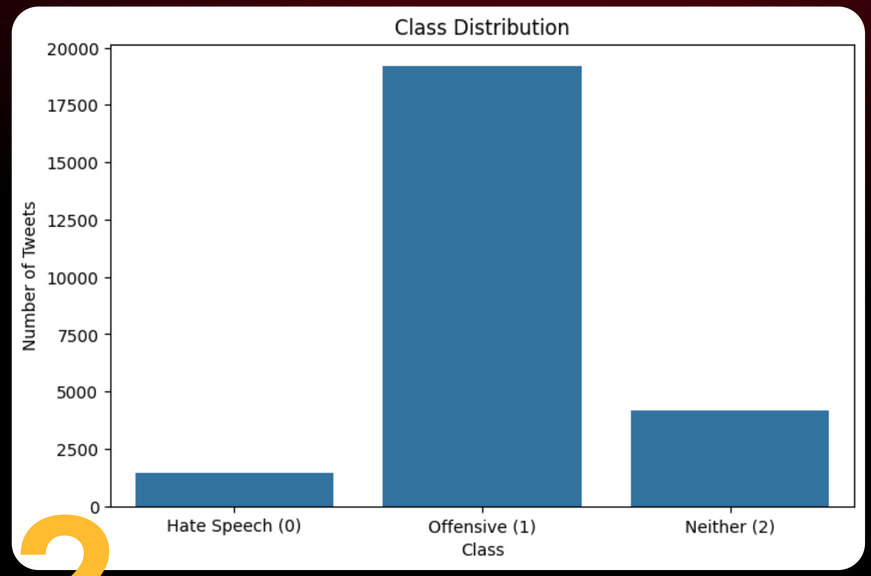
BBS NATURAL LANGUAGE PROCESSING

Exploratory Data Analysis

1

Understand dataset structure, types and check for missing values

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24783 entries, 0 to 24782
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Unnamed: 0          24783 non-null  int64
 1   count               24783 non-null  int64
 2   hate_speech         24783 non-null  int64
 3   offensive_language  24783 non-null  int64
 4   neither             24783 non-null  int64
 5   class               24783 non-null  int64
 6   tweet               24783 non-null  object
dtypes: int64(6), object(1)
```



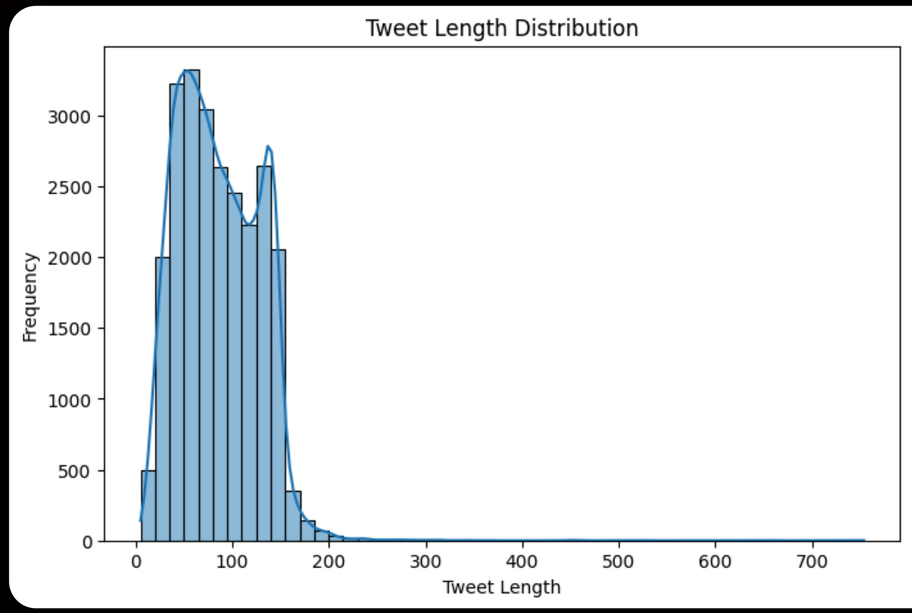
2

Visualize class distribution to understand dataset

- The dataset is **highly imbalanced**: Class 1 (~19000 samples)has >10× more samples than class 0 (~1500 samples).
- Imbalance can bias the model toward the majority class, hurting minority recall.

3

Analyze tweet lengths to check variability



Data Preparation

BBS NATURAL LANGUAGE PROCESSING

Data Cleaning 1

- Converts text to lowercase
- Removes mentions, URLs, hashtags, special characters
- Escapes HTML entities
- Strips extra whitespace

```
Before and after cleaning (first 5 tweets):

Original: !!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. &amp;amp; as a man you should always take the trash out...
Cleaned : rt as a woman you shouldnt complain about cleaning up your house as a man you should always take the trash out

Original: !!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
Cleaned : rt boy dats coldtyga dwn bad for cuffin dat hoe in the 1st place

Original: !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit
Cleaned : rt dawg rt you ever fuck a bitch and she start to cry you be confused as shit

Original: !!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
Cleaned : rt she look like a tranny

Original: !!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya &#57361;
```

Data Preparation

BBS NATURAL LANGUAGE PROCESSING

Text Preprocessing 2

- Tokenize each tweet
- Remove stopwords
- Lemmatize remaining words

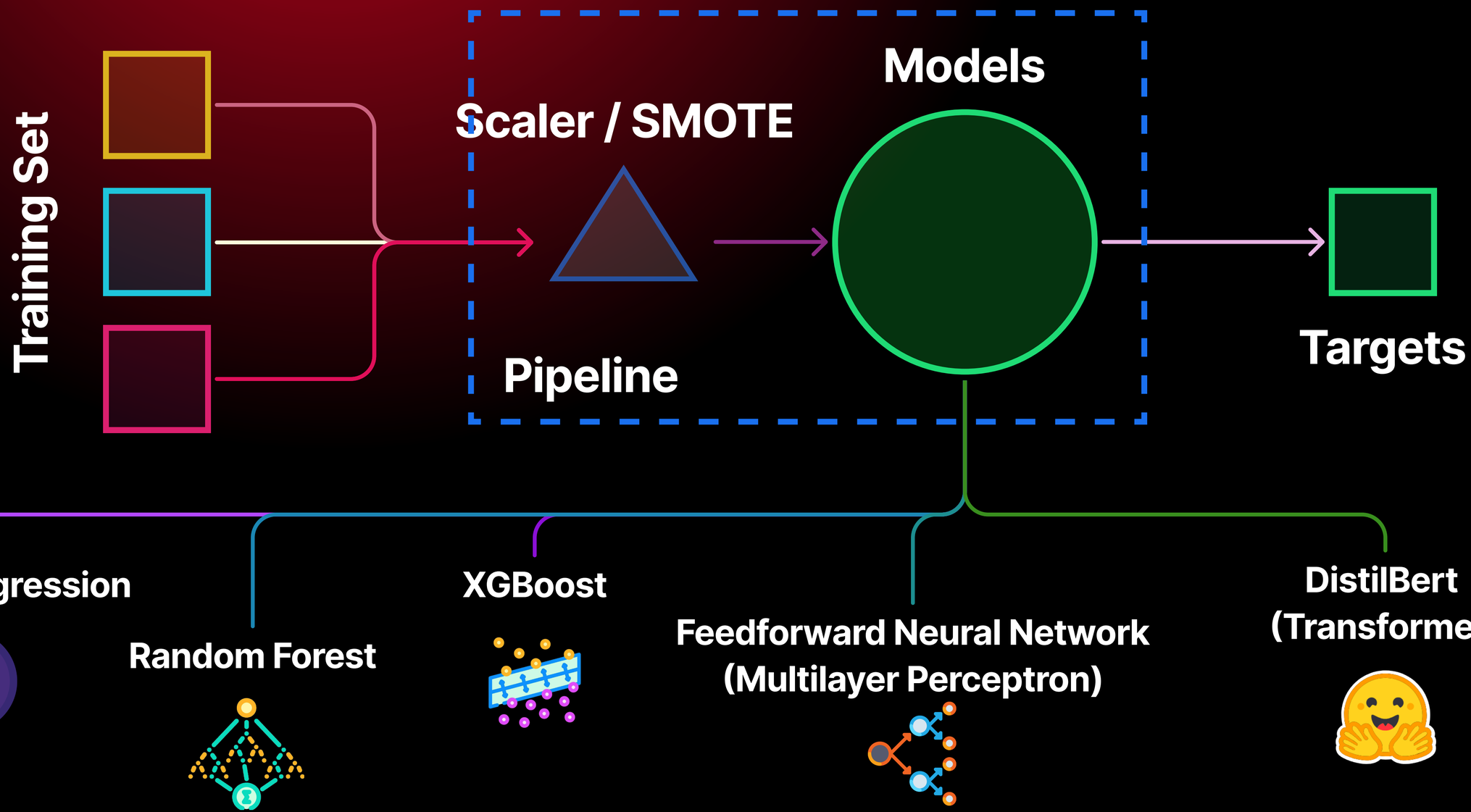
Feature Engineering 3

- Text-based features: Word count, Average word length
- Style-based features: Number of uppercase words, Exclamations, Hashtags, Mentions, Links
- Sentiment-based features: Polarity and subjectivity scores from TextBlob

Handle Class Imbalance with SMOTE 4

Modelling

BBS NATURAL LANGUAGE PROCESSING



Modelling / Machine Learning

BBS BOLOGNA BUSINESS SCHOOL NATURAL LANGUAGE PROCESSING

Classification Report

Random Forest

Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.539877	0.307692	0.391982	286.00000
1	0.918016	0.945284	0.931451	3838.00000
2	0.802850	0.811525	0.807164	833.00000
accuracy	0.886020	0.886020	0.886020	0.88602
macro avg	0.753581	0.688167	0.710199	4957.00000
weighted avg	0.876846	0.886020	0.879440	4957.00000

Logistic Regression

Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.314000	0.548951	0.399491	286.00000
1	0.958575	0.862168	0.907819	3838.00000
2	0.772139	0.931573	0.844396	833.00000
accuracy	0.855760	0.855760	0.855760	0.85576
macro avg	0.681571	0.780897	0.717235	4957.00000
weighted avg	0.890056	0.855760	0.867832	4957.00000

XGBoost

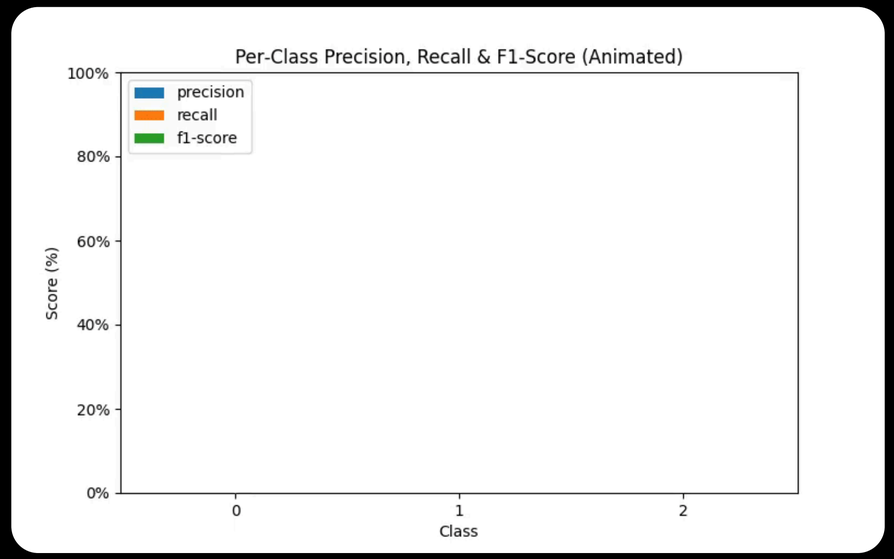
XGBoost Classification Report:

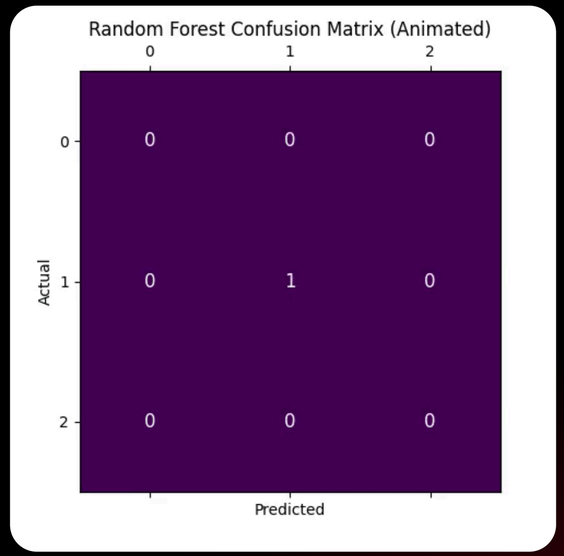
	precision	recall	f1-score	support
0	0.606061	0.209790	0.311688	286.000000
1	0.938565	0.947368	0.942946	3838.000000
2	0.796748	0.941176	0.862961	833.000000
accuracy	0.903772	0.903772	0.903772	0.903772
macro avg	0.780458	0.699445	0.705865	4957.000000
weighted avg	0.895549	0.903772	0.893084	4957.000000

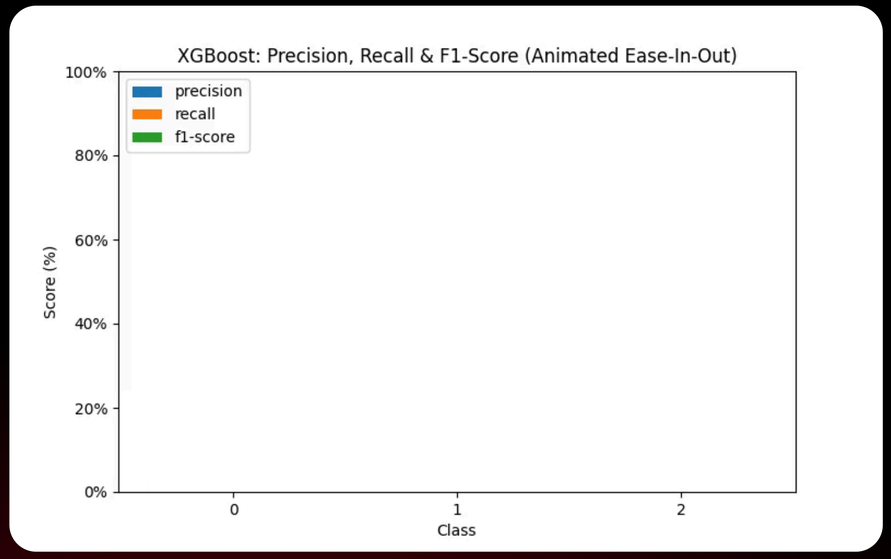
Modelling / Machine Learning

BBS NATURAL LANGUAGE PROCESSING

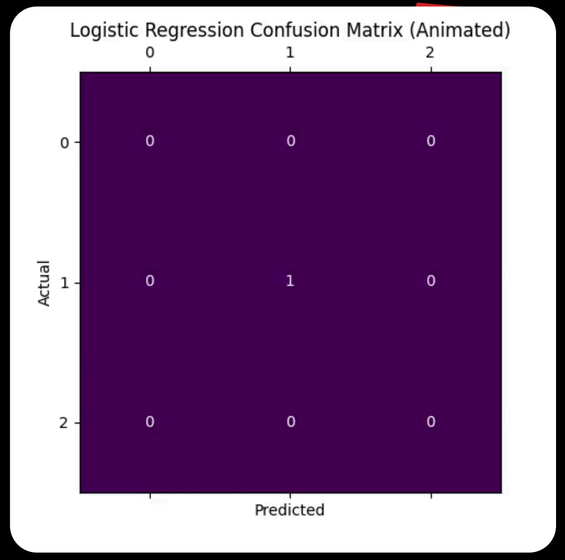
Random Forest

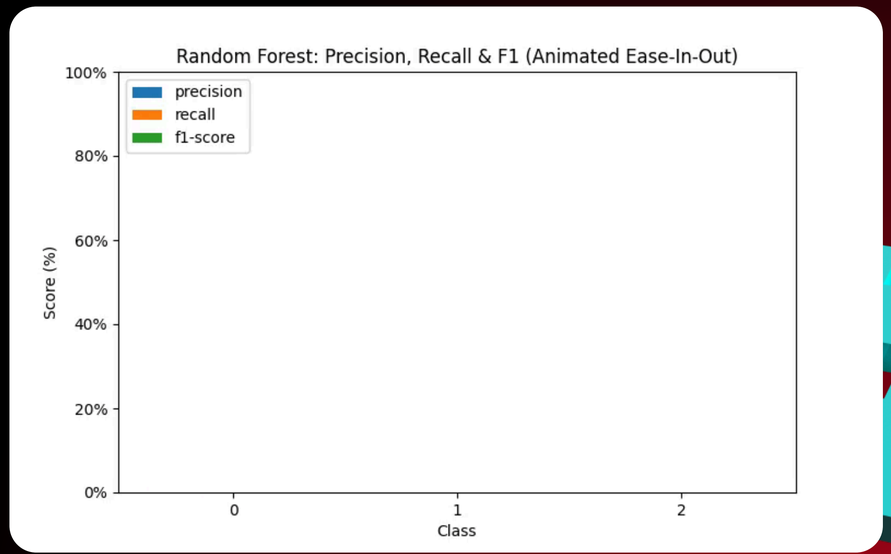




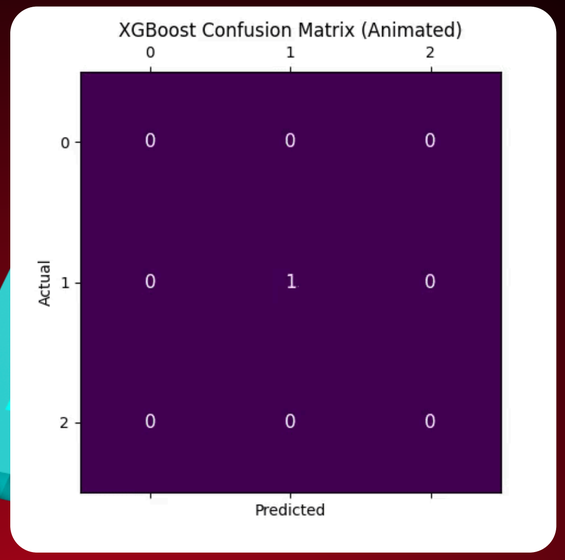


Logistic Regression





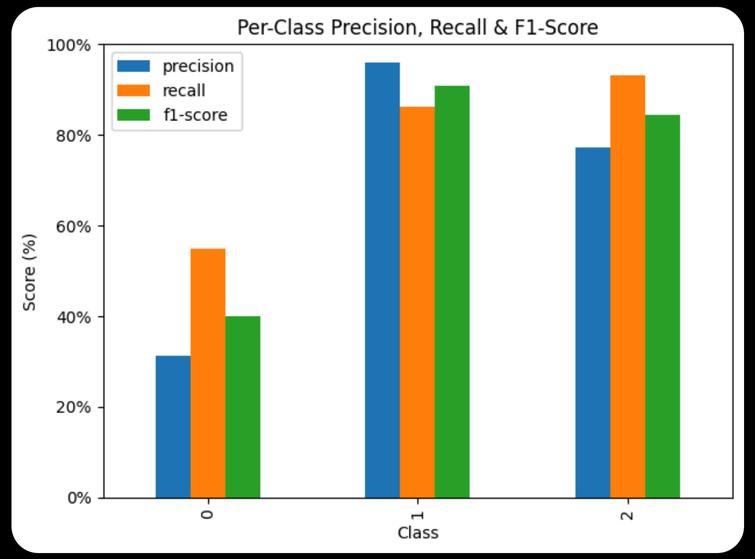
XGBoost

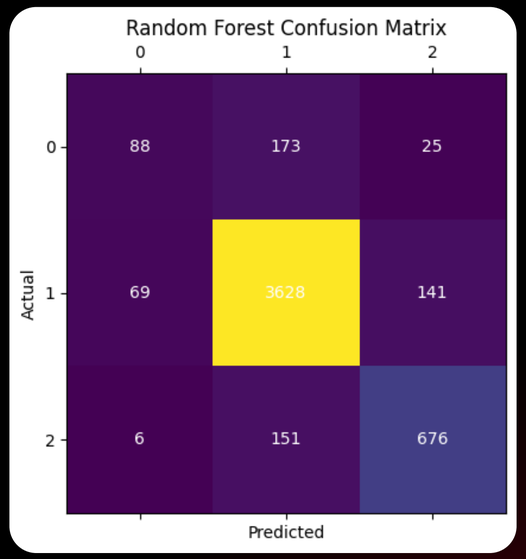


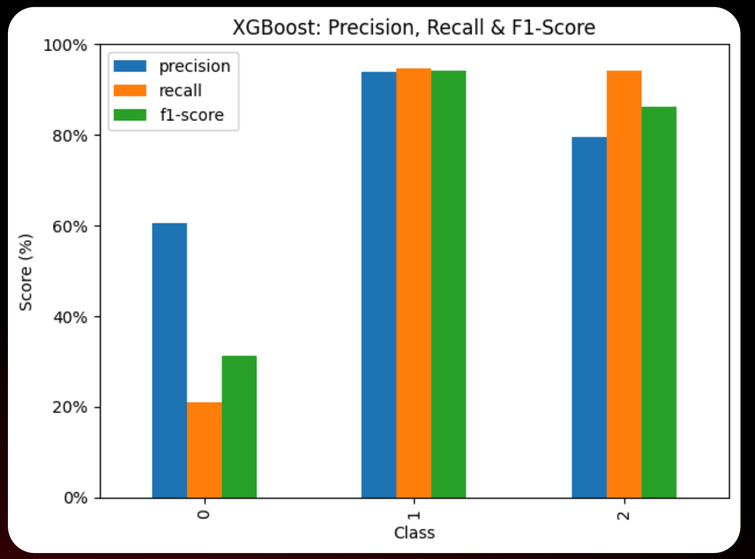
Modelling / Machine Learning

BBS NATURAL LANGUAGE PROCESSING

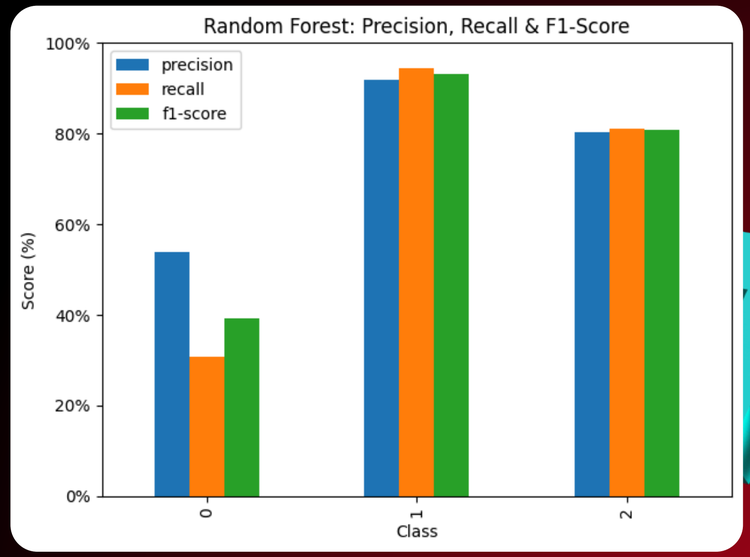
Random Forest



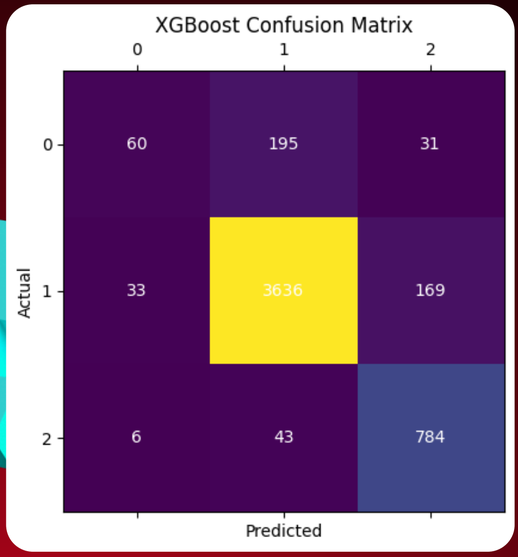




Logistic Regression



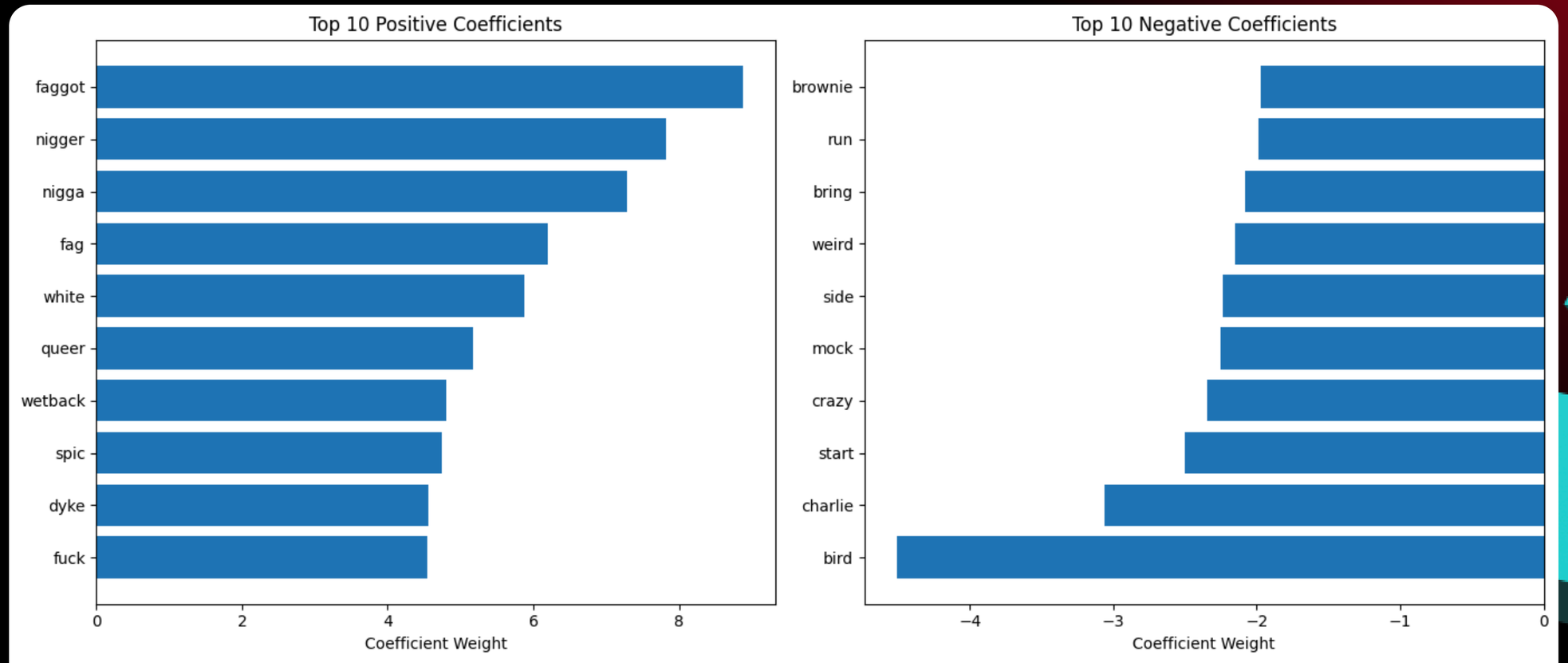
XGBoost



Modelling / Machine Learning

BBS NATURAL LANGUAGE PROCESSING

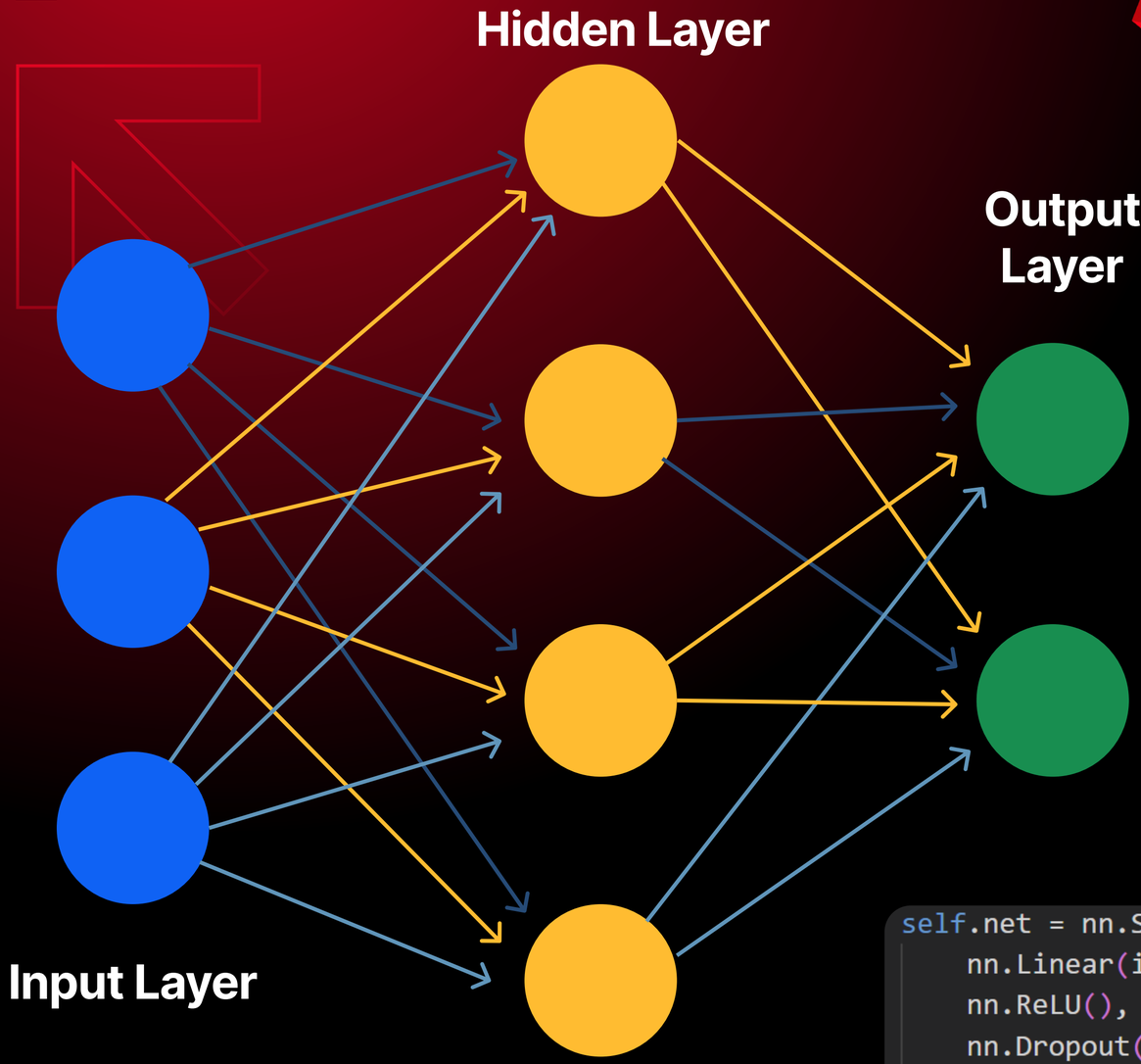
Feature Importance



	feature	weight
0	faggot	8.878060
1	nigger	7.823929
2	nigga	7.286853
3	fag	6.195476
4	white	5.880571
5	queer	5.179053
6	wetback	4.805923
7	spic	4.750245
8	dyke	4.565576
9	fuck	4.538732
10	bird	-4.507051
11	bitch	4.407417
12	whitey	4.223597
13	coon	4.134531
14	kill	4.109638
15	chink	3.960911
16	beaner	3.904320
17	as	3.859999
18	racist	3.734082
19	fucking	3.669097

Modelling / Multilayer Perceptron

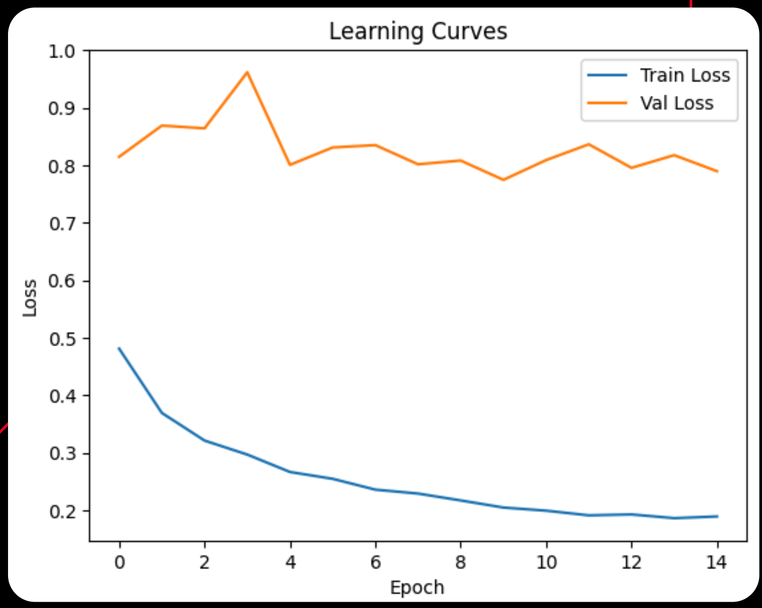
BBS NATURAL LANGUAGE PROCESSING



Feedforward Neural Network

```
self.net = nn.Sequential(
    nn.Linear(input_dim, hidden_dim),
    nn.ReLU(),
    nn.Dropout(dropout),
    nn.Linear(hidden_dim, num_classes)
)
```

```
Epoch 1/20 — train_loss: 0.4811 — val_loss: 0.8149
Epoch 2/20 — train_loss: 0.3693 — val_loss: 0.8691
Epoch 3/20 — train_loss: 0.3212 — val_loss: 0.8644
Epoch 4/20 — train_loss: 0.2969 — val_loss: 0.9618
Epoch 5/20 — train_loss: 0.2666 — val_loss: 0.8008
Epoch 6/20 — train_loss: 0.2547 — val_loss: 0.8311
Epoch 7/20 — train_loss: 0.2358 — val_loss: 0.8352
Epoch 8/20 — train_loss: 0.2290 — val_loss: 0.8019
Epoch 9/20 — train_loss: 0.2171 — val_loss: 0.8083
Epoch 10/20 — train_loss: 0.2047 — val_loss: 0.7748
Epoch 11/20 — train_loss: 0.1993 — val_loss: 0.8091
Epoch 12/20 — train_loss: 0.1911 — val_loss: 0.8366
Epoch 13/20 — train_loss: 0.1927 — val_loss: 0.7956
Epoch 14/20 — train_loss: 0.1862 — val_loss: 0.8177
Early stopping at epoch 15
```



Modelling / Multilayer Perceptron

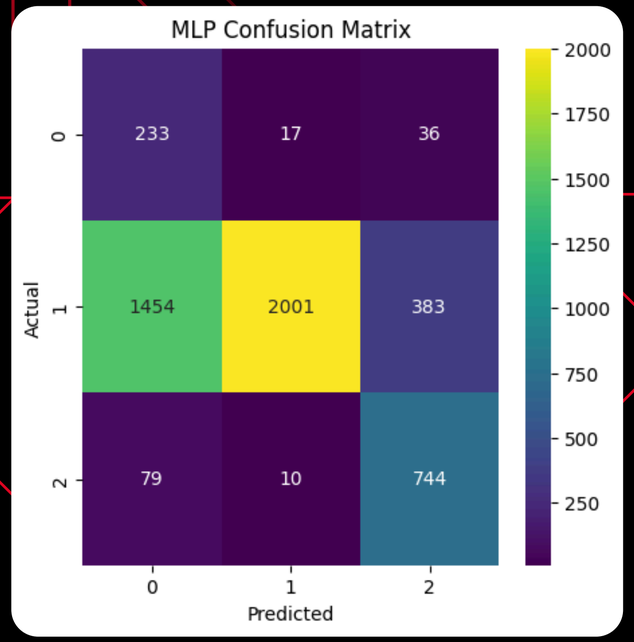
BBS NATURAL LANGUAGE PROCESSING

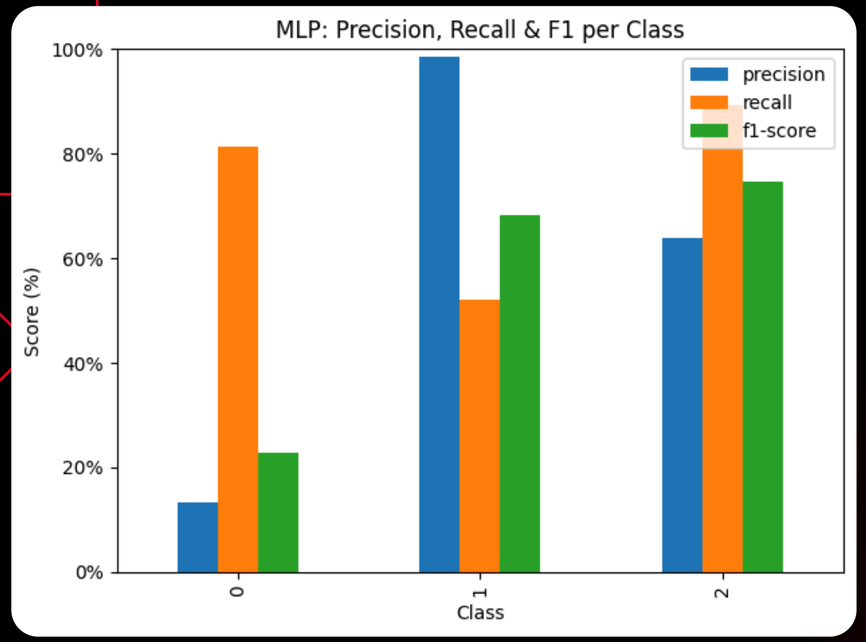
MLP Classification Report:

	precision	recall	f1-score	support
0	0.131937	0.814685	0.227096	286.000000
1	0.986686	0.521365	0.682237	3838.000000
2	0.639725	0.893157	0.745491	833.000000
accuracy	0.600767	0.600767	0.600767	0.600767
macro avg	0.586116	0.743069	0.551608	4957.000000
weighted avg	0.879065	0.600767	0.666606	4957.000000

- **Accuracy**: Low
- **F1-score for Class 0**: Low
- **Class 1**: Suffers most, many samples misclassified as class 0 (1454 samples)

⇒ Feedforward Neural Network Model underperforms, especially on Class 1!





Modelling / Transformers

BBS NATURAL LANGUAGE PROCESSING

```python
# --- Model & Trainer Setup ---
model = DistilBertForSequenceClassification.from_pretrained('distilbert-base-uncased', num_labels=3)
training_args = TrainingArguments(
    output_dir=train_output_dir,
    overwrite_output_dir=True,    # ← this wipes ./results before each run
    per_device_train_batch_size=8,
    num_train_epochs=5,
    learning_rate=1e-5,
    weight_decay=0.01,
    warmup_steps=100,
    logging_steps=50,
    fp16=False,
    no_cuda=False, # Use GPU
)
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset['train'],
    eval_dataset=tokenized_dataset['test'],
)
```

```python
# --- Train or Load  ---
if not os.path.isdir(fine_tuned_dir):
    print(f"Checkpoint not found in {fine_tuned_dir}. Starting training on GPU...")
    trainer.train()
    trainer.evaluate()
    trainer.save_model(fine_tuned_dir)
    tokenizer.save_pretrained(fine_tuned_dir)
else:
    print(f"Loading fine-tuned model from {fine_tuned_dir} (GPU)...")
    model = DistilBertForSequenceClassification.from_pretrained(fine_tuned_dir)
    tokenizer = DistilBertTokenizerFast.from_pretrained(fine_tuned_dir)
```

DistilBERT

```python
import torch.nn as nn

# Match these counts to your training split distribution
counts = torch.tensor([260, 3839, 858], dtype=torch.float)
weights = counts.sum() / (3 * counts)
# Move weights to GPU if available:
weights = weights.to(device)

loss_fn = nn.CrossEntropyLoss(weight=weights)
```

[12395/12395 12:54, Epoch 5/5]

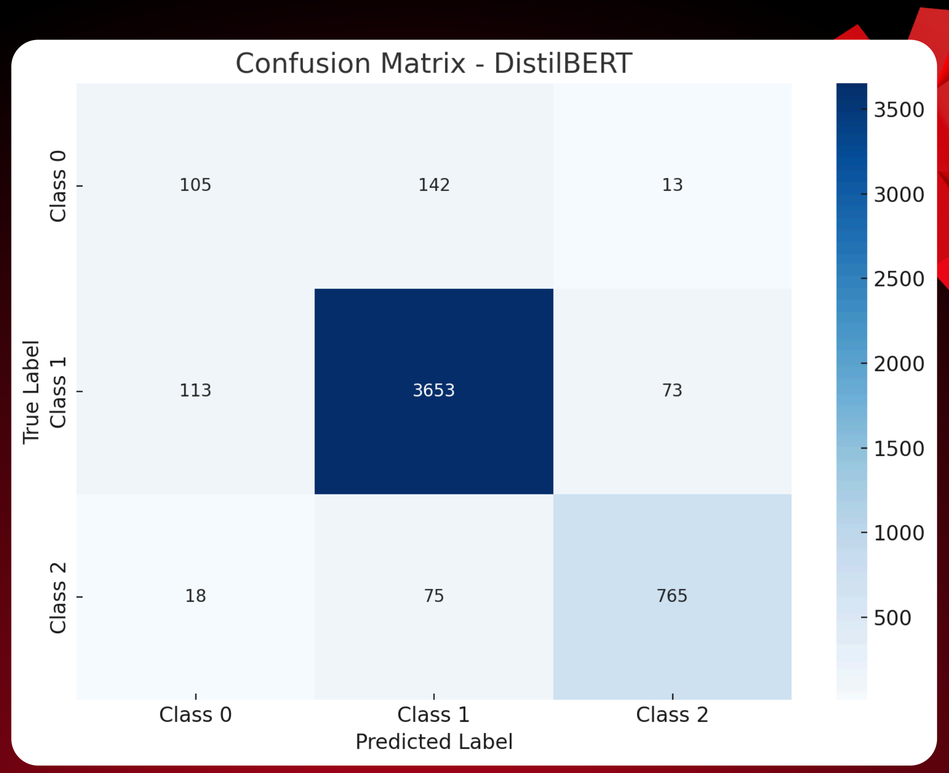
Step	Training Loss
50	1.105800
100	1.081500
150	0.993200
200	0.965800
250	0.763900
300	0.742600
350	0.585000
400	0.765500
450	0.755500
500	0.843400
550	0.672100
600	0.686000
650	0.744200
700	0.612100

Modelling / Transformers

BBS BOLOGNA BUSINESS SCHOOL NATURAL LANGUAGE PROCESSING

- **Accuracy**: 91.2%, is very high, excellent overall performance
- **Macro F1-score**: 0.755, is good balance across all classes (useful for imbalanced data)
- **Weighted F1-score**: 0.911 is strong average performance, weighted by class size

	precision	recall	f1-score	support
0	0.444915	0.403846	0.423387	260.000000
1	0.943928	0.951550	0.947723	3839.000000
2	0.898942	0.891608	0.895260	858.000000
accuracy	0.912447	0.912447	0.912447	0.912447
macro avg	0.762595	0.749001	0.755457	4957.000000
weighted avg	0.909967	0.912447	0.911141	4957.000000

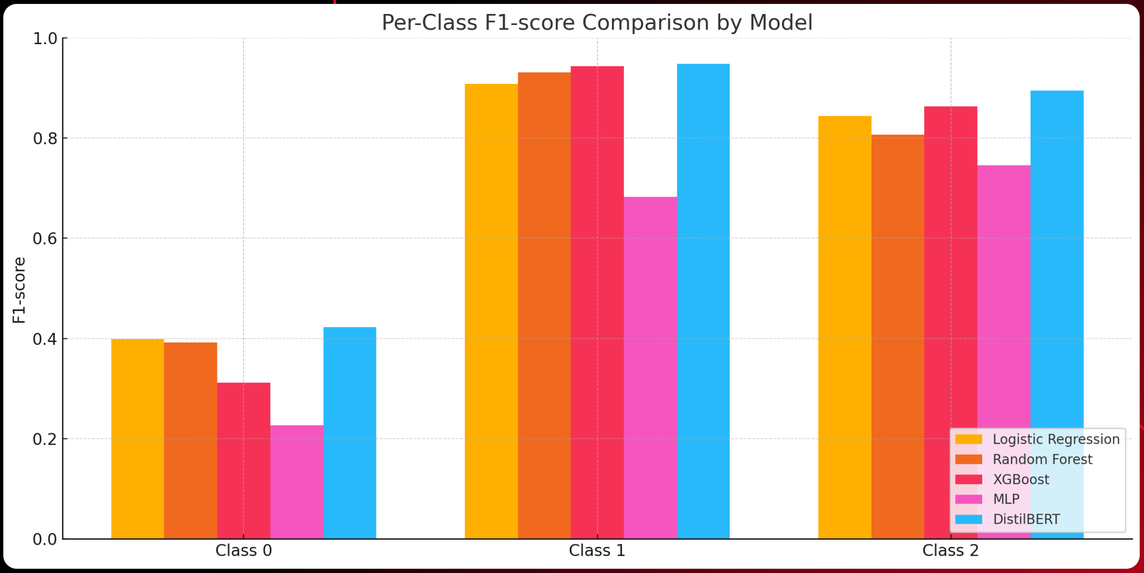


- **Class 0**: Mostly confused with Class 1
- **Class 1**: Almost perfect prediction
- **Class 2**: Some confusion with class 1, but still strong (F1 = 0.895)

Summary

BBS NATURAL LANGUAGE PROCESSING

No.	Model	Accuracy	Macro AVG F1	Weighted AVG F1
1	DistilBERT	0.912447	0.755457	0.911141
2	XGBoost	0.903772	0.705865	0.893084
3	Random Forest	0.88602	0.710199	0.87944
4	Logistic Regression	0.85576	0.717235	0.867832
5	Feedforward Neural Network	0.600767	0.551608	0.666606



Best Model: DistilBERT

- Highest Accuracy (91.2%)
- Highest Macro F1 score, important for imbalanced datasets.
- Highest Weighted F1 score, indicating overall balanced performance across all classes.

F1-Score Analysis

- **Class 0** (Hate Speech): All models struggle. DistilBERT performs best (F1 = 0.42). MLP performs worst.
- **Class 1** (Offensive Speech): All models perform well. DistilBERT and XGBoost are top performers (F1 = 0.94–0.95).
- **Class 2** (Neither): DistilBERT again leads. MLP is weakest here too.