# *Detecting Indirect Racism and Sexism through Speech and Text Analysis*

## Machine Learning Project

Agori Fotini - Tolia Anna

12 February, 2024

# *Introduction*

**Problem**: Identifying implicit racism/sexism through linguistic and vocal features.

**Data Sources**: Greek TV programs, news, debates with multiple speakers.

**Analysis:** Combination of transcription, NLP, and speech analysis.

# *Steps*

## Gathered Data

Youtube videos from TV Shows, Livestreams, news broadcasts. Multiple speakers , diverse topics.

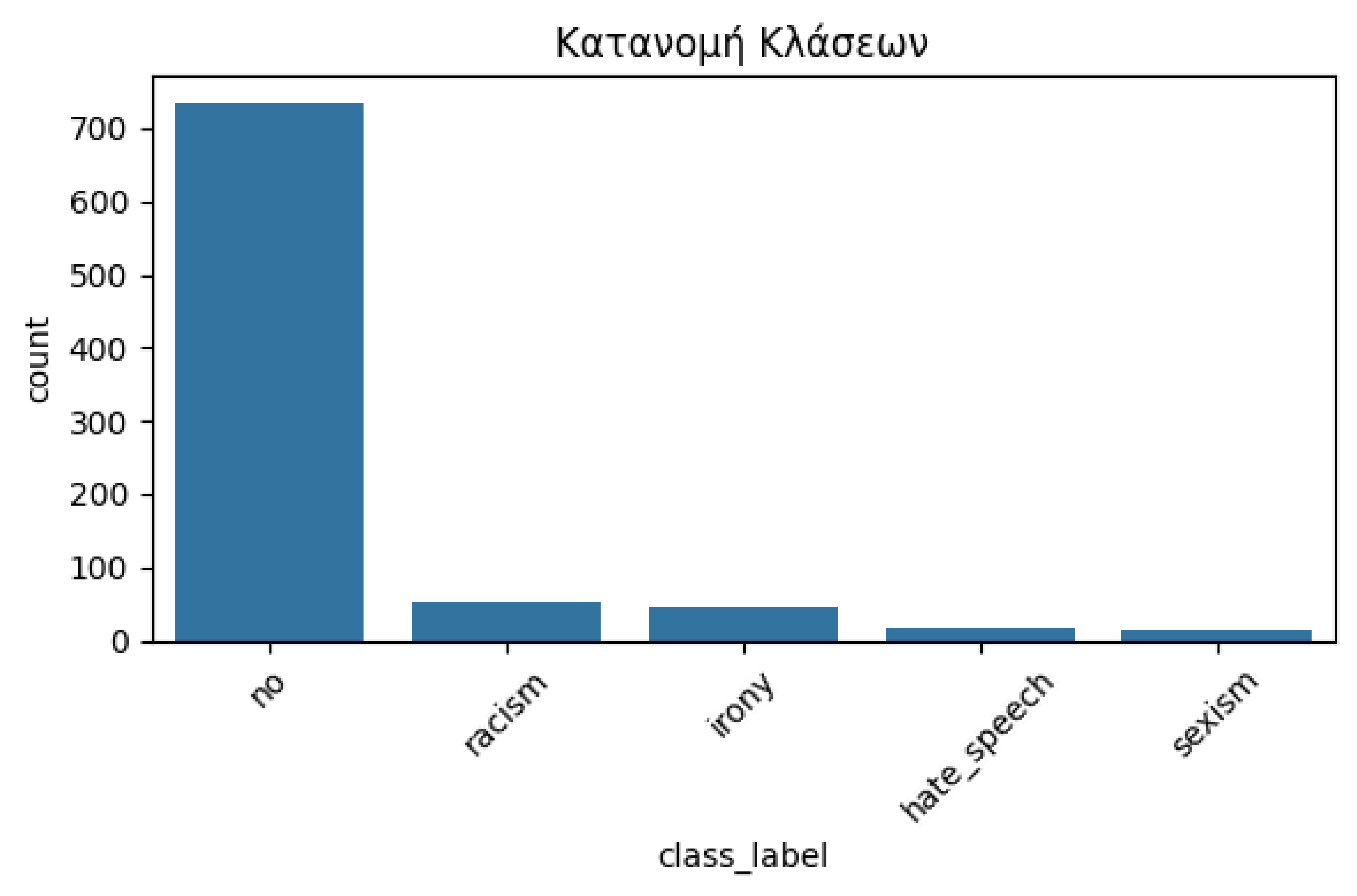
## Transcribed videos

**WhisperX** for ASR (Automatic Speech Recognition).
**Manual Corrections** due to errors in Greek transcription.

## Data annotation

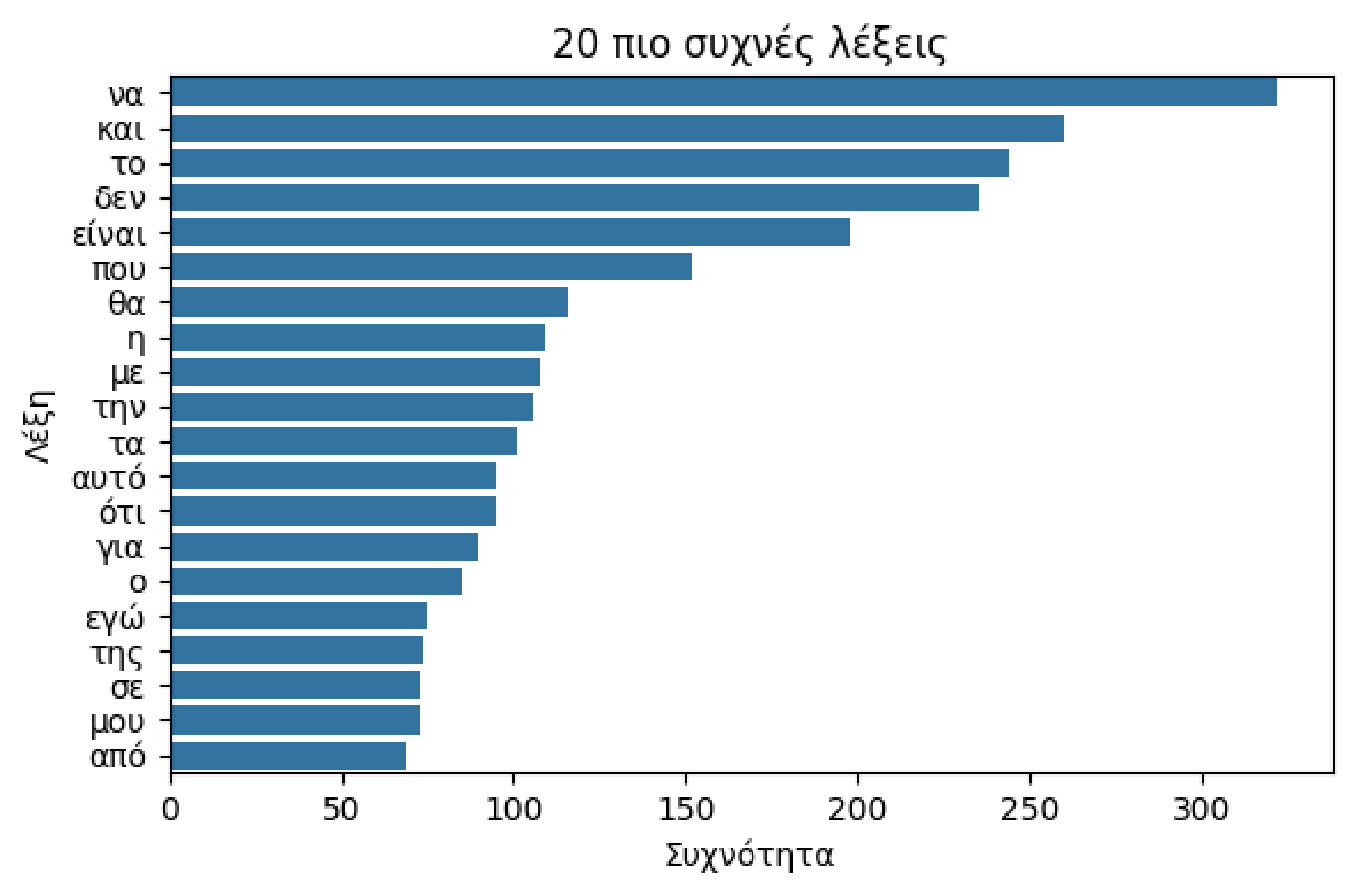
**Utterance** level
**Single-label** classification (each utterance tagged as hatespeech, racism, sexism, irony, or no).
**Manual annotation**

# *Steps*




## Data Load & Exploratory Analysis

We did exploratory data analysis and founf the class distribution, number of words per text and frequent words




## Undersampling

After we started training without undersampling of the data we decided to undersample major class

## Data Cleaning & Stopwords

```python
# Ορισμός σημαντικών λέξεων που θέλουμε να κρατήσουμε
important_words = {"ρατσισμός", "σεξισμός", "ξενοφοβία", "κατά", "δικαιώματα", "προσβολή", "διάκριση",
                   "γυναίκα", "άντρας", "μαύρος", "λευκός", "ξένος", "αλλοδαπός"}
custom_stopwords = {word for word in custom_stopwords if word not in important_words}
```

```
Παραδείγματα tokens μετά την εφαρμογή stopwords:
                                         cleaned_text
0  μιας πρότασης  σελίδων η οποία δεν έγινε χωρίς...
1                                         στα τέσσερα
2                                   στα τέσσερα εσείς
3                                         στα τέσσερα
4                                              λοιπόν

                                               tokens
0  [μιας, πρότασης, σελίδων, οποία, έγινε, χωρίς,...
1                                      [στα, τέσσερα]
2                               [στα, τέσσερα, εσείς]
3                                      [στα, τέσσερα]
4                                            [λοιπόν]
```

## Tokenization and Feature extraction

TF-IDF Feature Matrix: (190, 920)
iltering out common words and giving more weight to important terms

# Training Method

## Leave-One-Out

- **Why?**

  Small dataset required maximizing evaluation efficiency.

- **Mechanism**

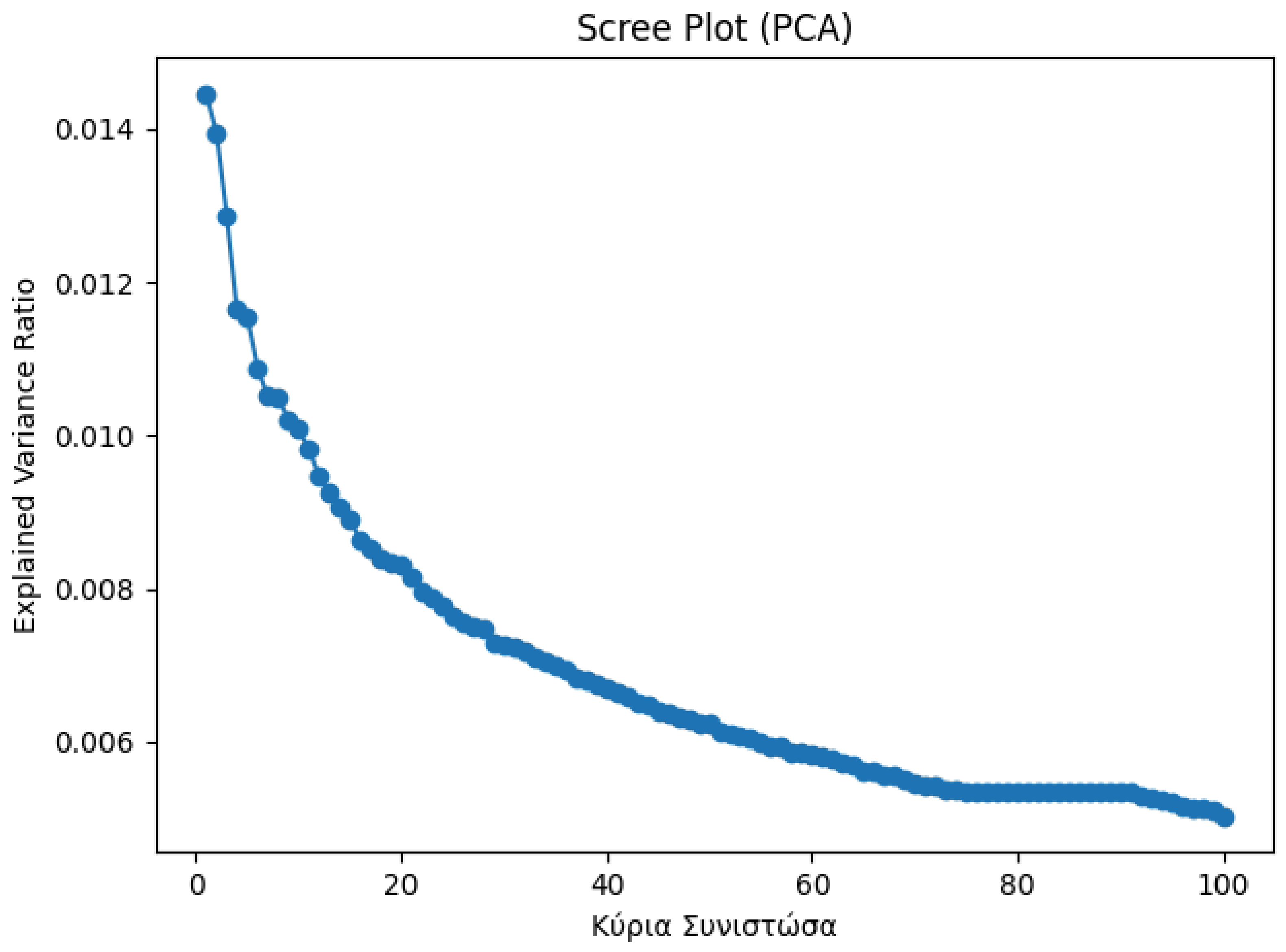
  One sample is left out each time while training on the rest.

- **Benefits**

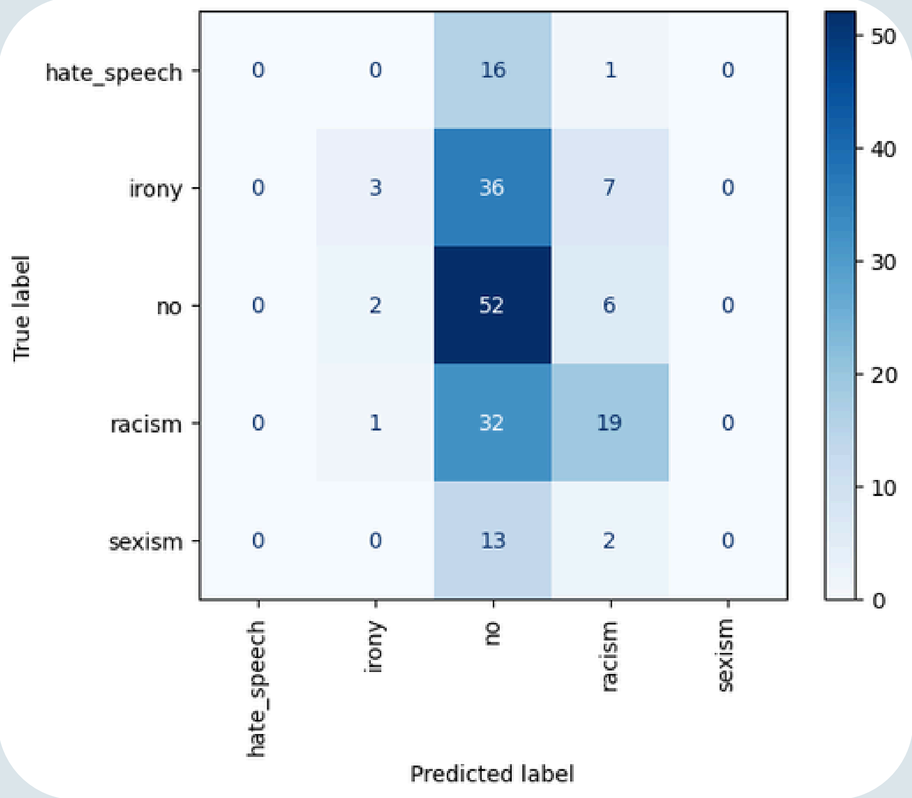
  Improved generalization and performance assessment.

# *Principal Component Analysis*

- There was no significant elbow
- We did manually runs with 50 - 200 components

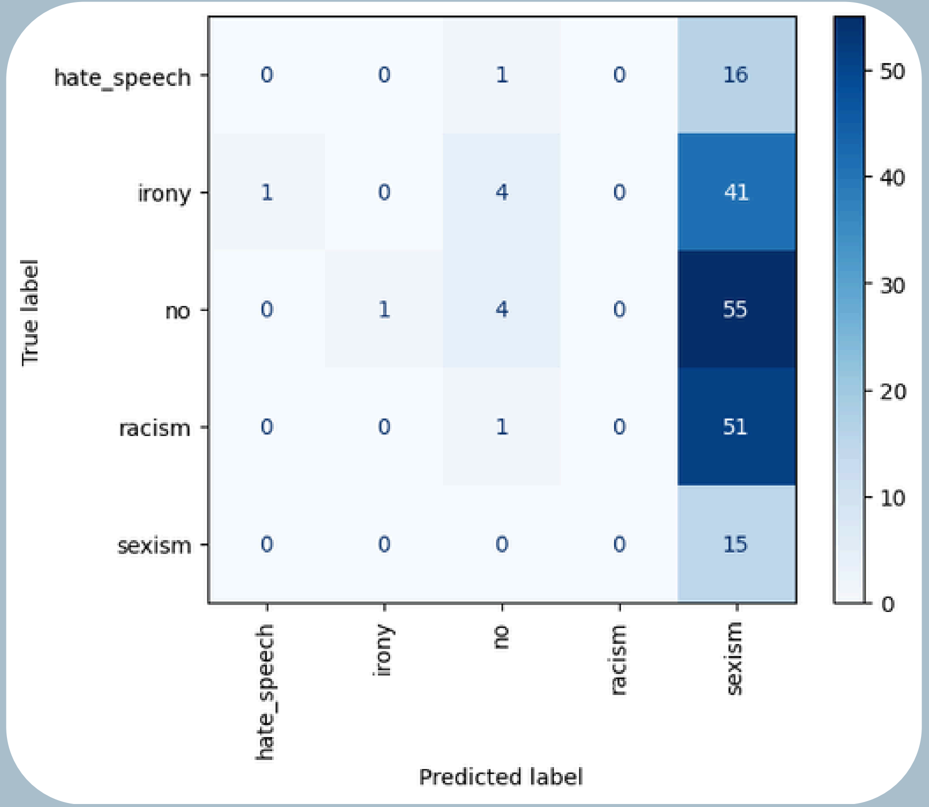


# Model Training with PCA before TFIDF



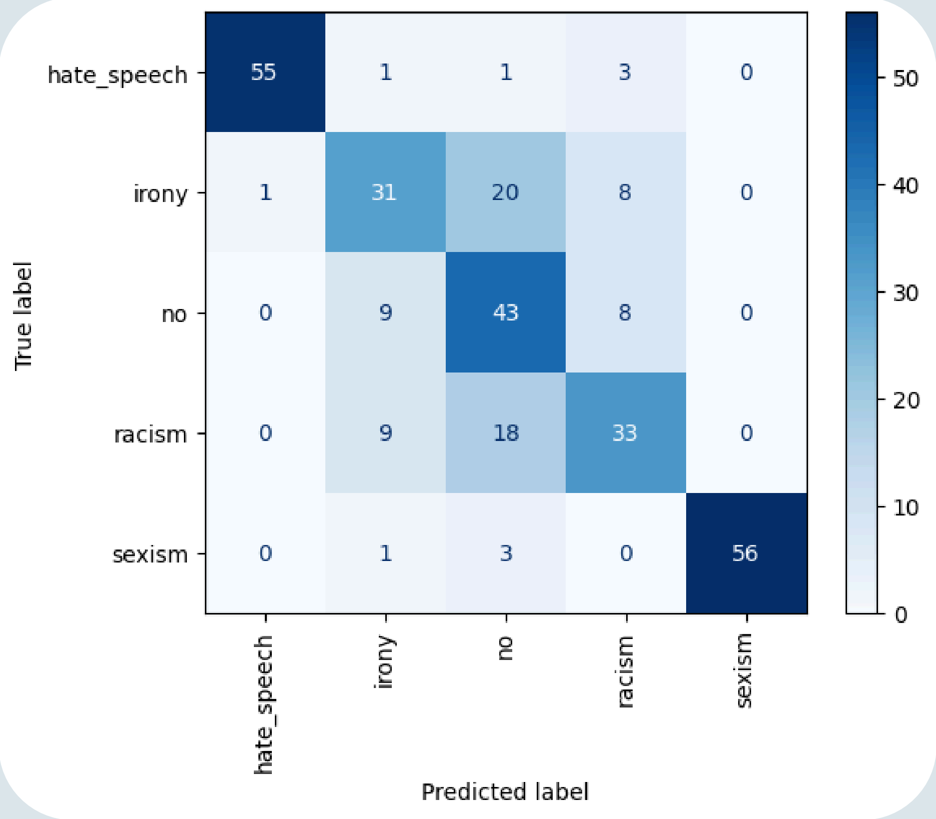

## Logistic Regression

- 60 Components
- Did not predict 2 of the classes
- **Macro f1: 0.21**




## Naive Bayes

- Did not work well with PCA and we used Gausian NAive Bayes
- **Macro-f1: 0,05**

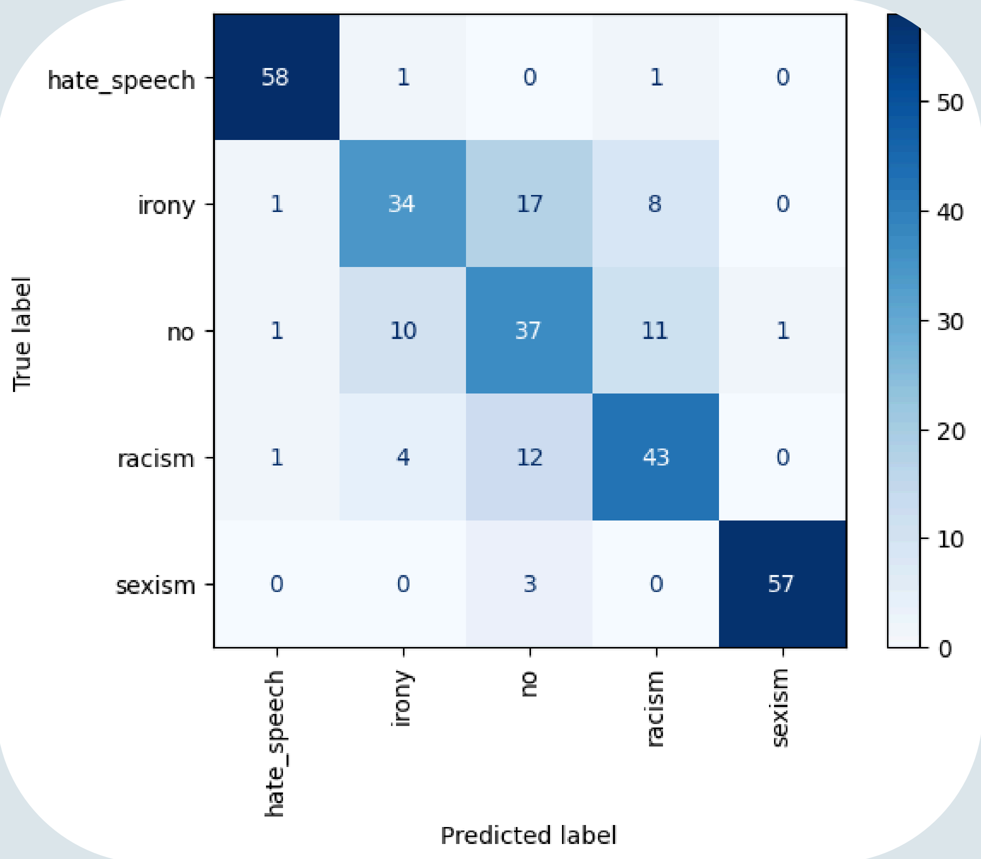



## SVM

- 150 Components
- **Macro-f1: 0.28**
- **Smote macro-f1: 0,73**

# Model Training without PCA

## Logistic Regression




- without smote:

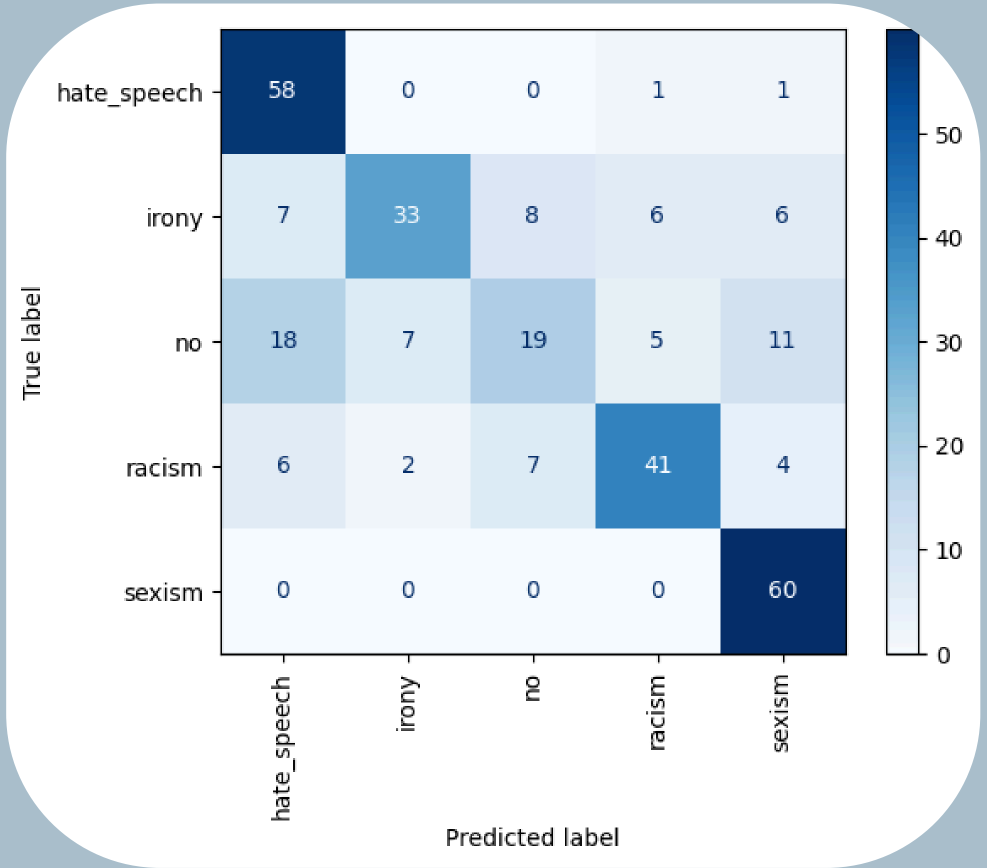
  Macro f1 -> 21%

- smote :

  Macro f1 -> **76%**

**261.90% increase**

## Naive Base




- without smote:

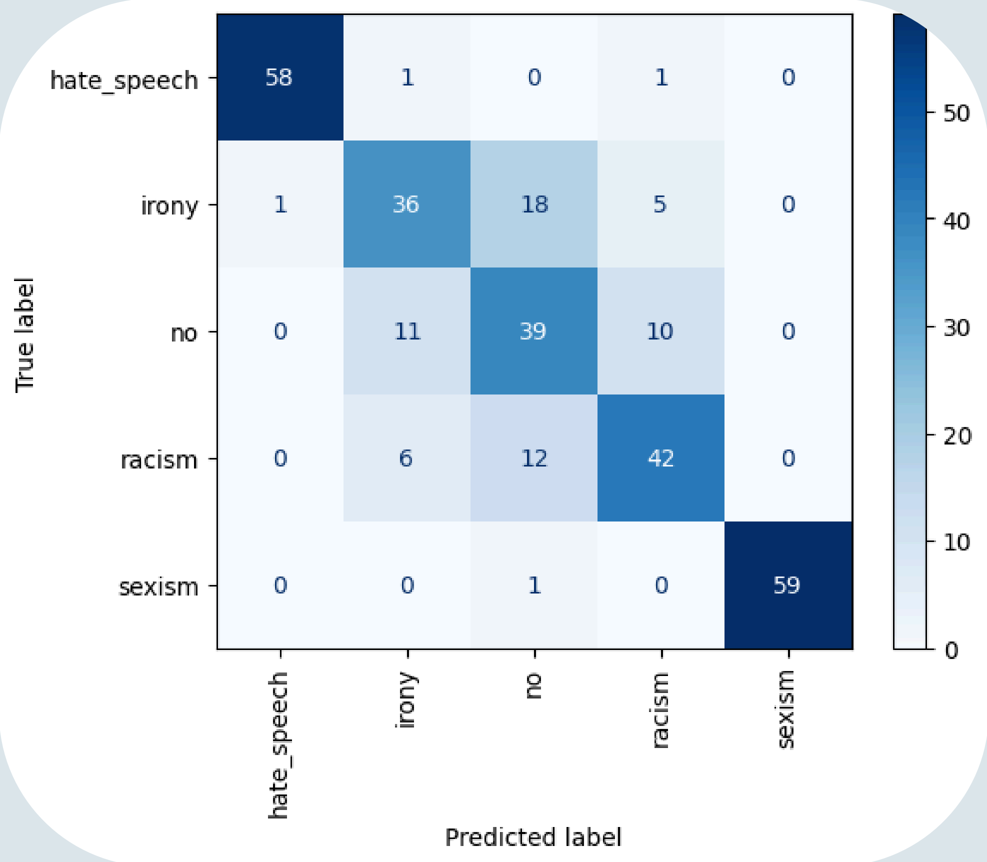
  Macro f1 -> 22%

- smote :

  Macro f1 -> **68%**

**209.09% increase**

## SVM




- without smote:

  Macro f1 -> 32%

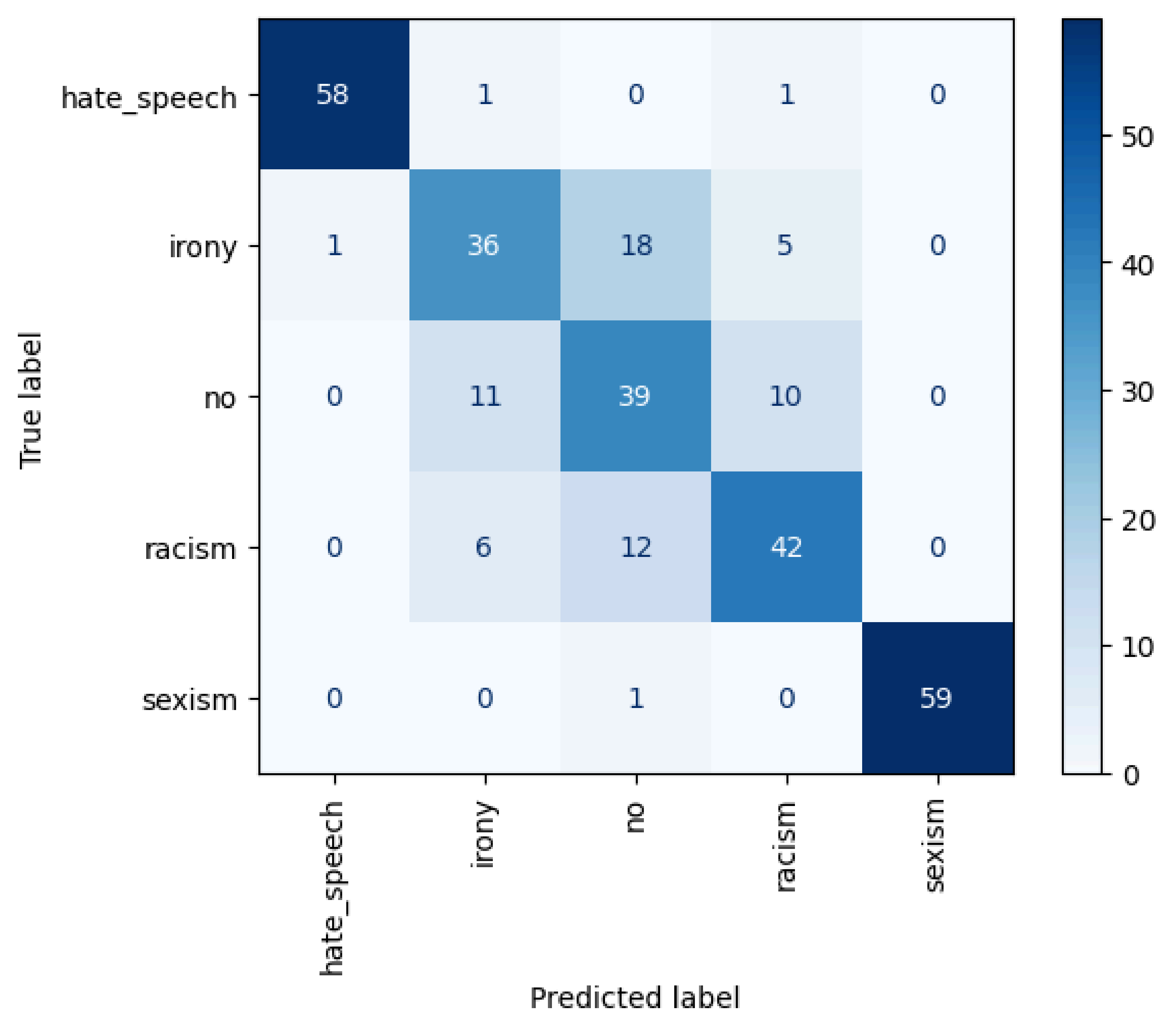
- smote :

  Macro f1 -> **78%**

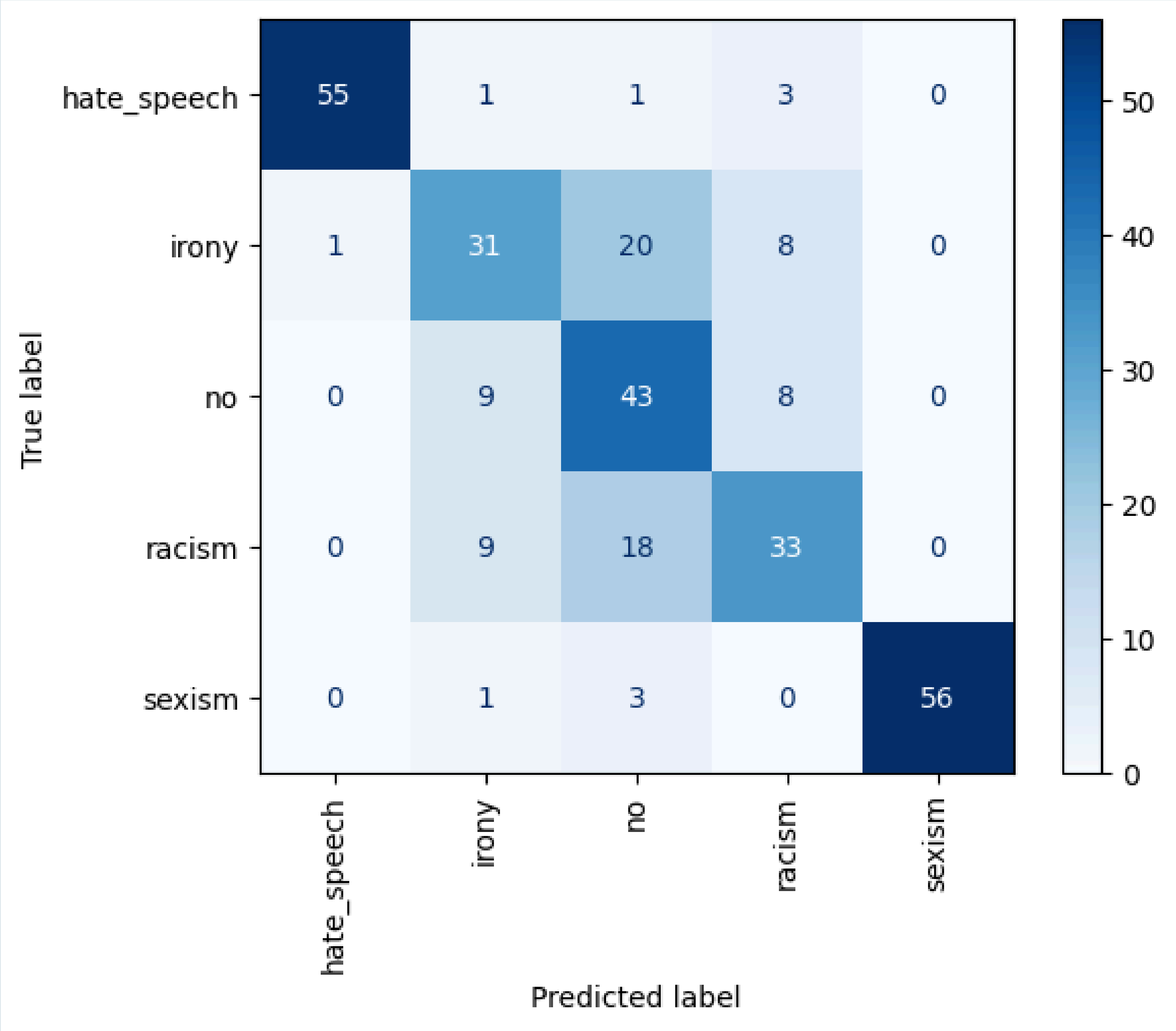
**143.75% increase**

# SVM BEST

## Without PCA




## With PCA



# SVM BEST

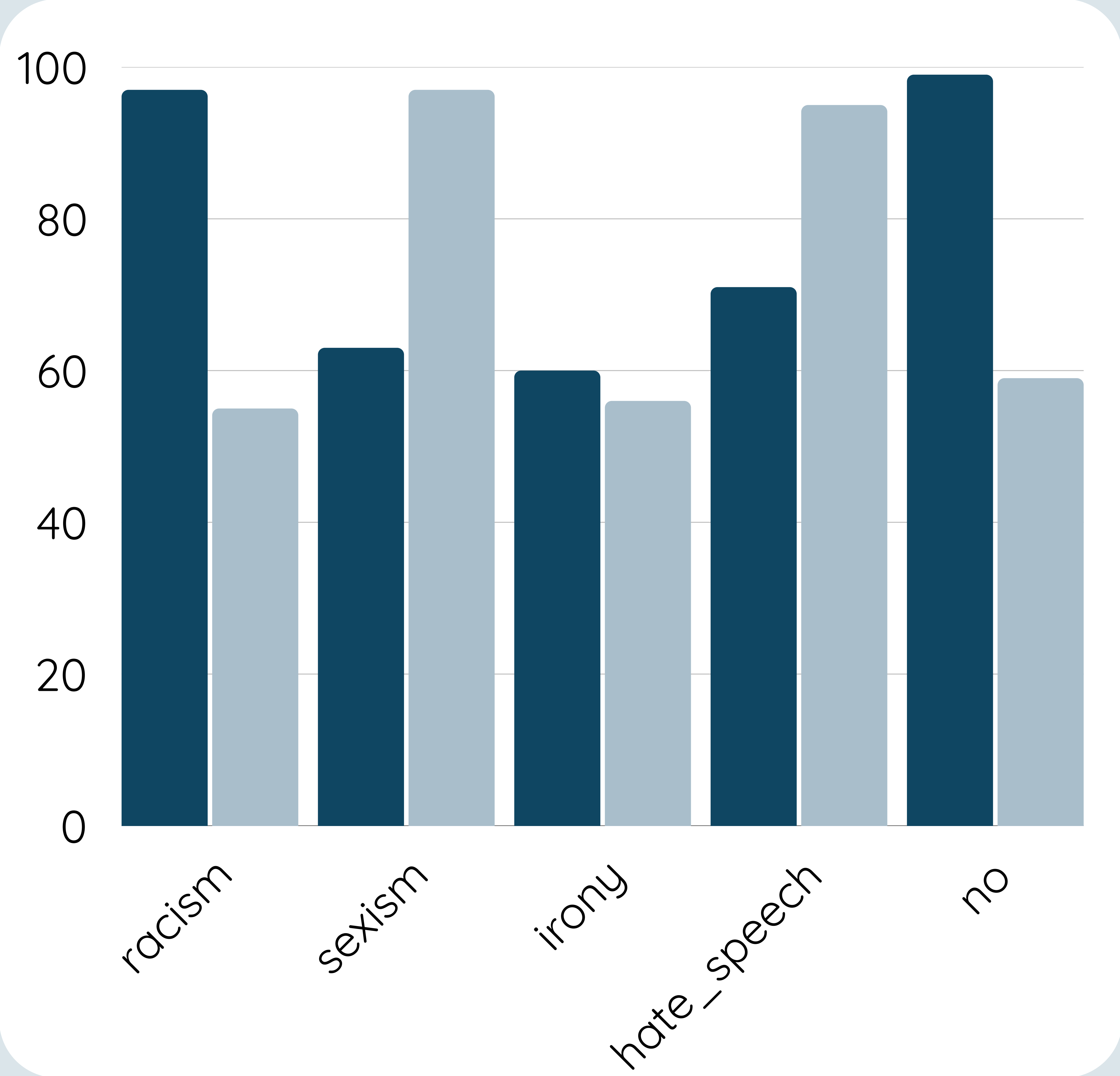
## Without PCA

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| racism | 0.98 | 0.97 | 0.97 | 60 |
| sexism | 0.67 | 0.60 | 0.63 | 60 |
| irony | 0.56 | 0.65 | 0.60 | 60 |
| hate_speech | 0.72 | 0.70 | 0.71 | 60 |
| no | 1.00 | 0.98 | 0.99 | 60 |
| accuracy | | | 0.78 | 300 |
| macro avg | 0.79 | 0.78 | 0.78 | 300 |
| weighted avg | 0.79 | 0.78 | 0.78 | 300 |

## With PCA

Classification report:

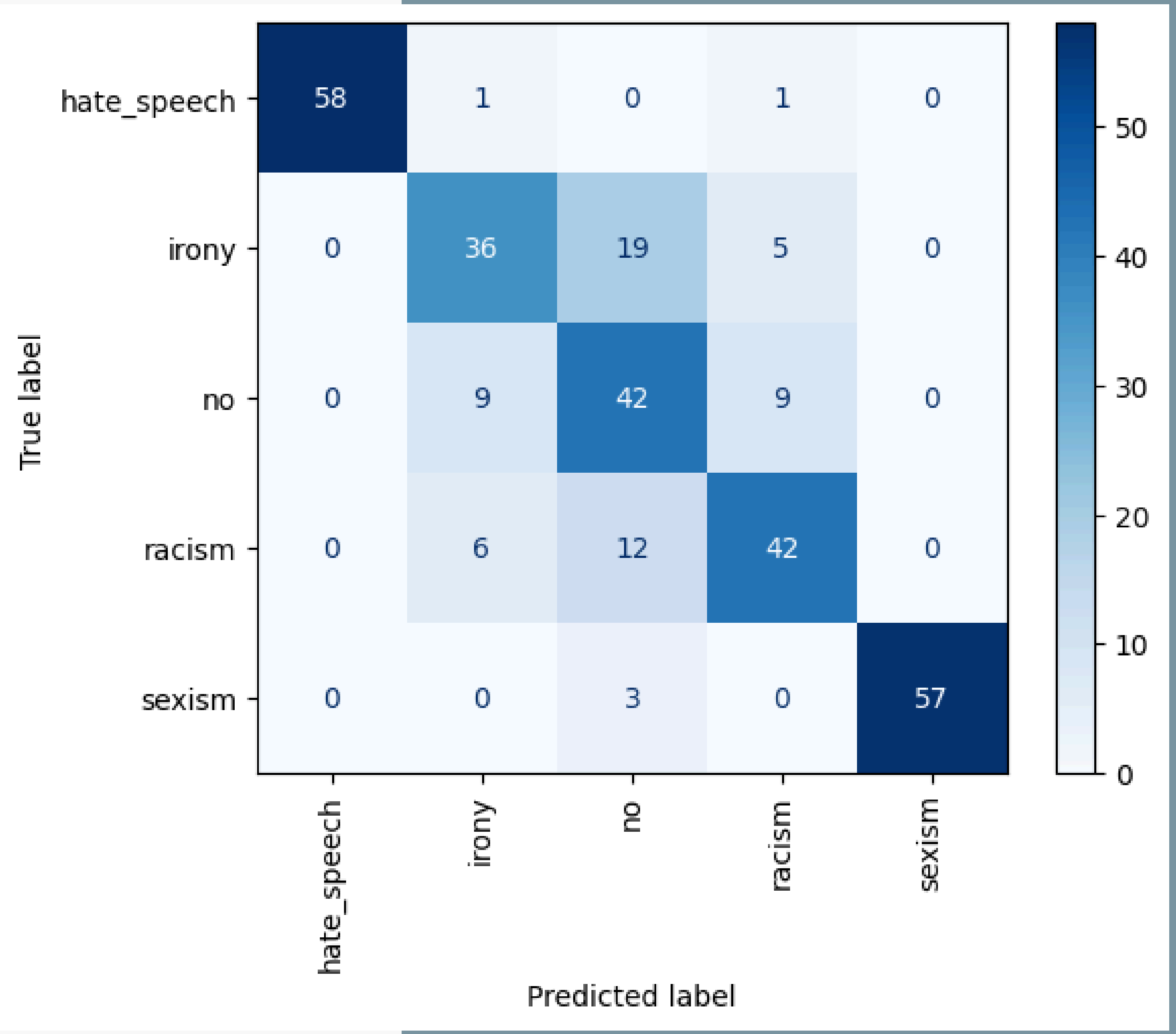
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| hate_speech | 0.98 | 0.92 | 0.95 | 60 |
| irony | 0.61 | 0.52 | 0.56 | 60 |
| no | 0.51 | 0.72 | 0.59 | 60 |
| racism | 0.63 | 0.55 | 0.59 | 60 |
| sexism | 1.00 | 0.93 | 0.97 | 60 |
| accuracy | | | 0.73 | 300 |
| macro avg | 0.75 | 0.73 | 0.73 | 300 |
| weighted avg | 0.75 | 0.73 | 0.73 | 300 |



# *Fusion Between the Best Classifiers (SVM & LR)*

**Experiment:** Combining SVM + LR for fusion voting.

- Average accuracy : 77.00%, identical to SVM.
- Fusion Voting did not improve performance, confirming that SVM dominates predictions.
- "Racism" & "No" classes had extremely high precision & recall (~98-100%).
- "Irony" remained the most challenging class (~62% recall), with no significant improvement.
- Slight improvement in overall prediction stability, but not enough to justify using Fusion Voting over SVM.

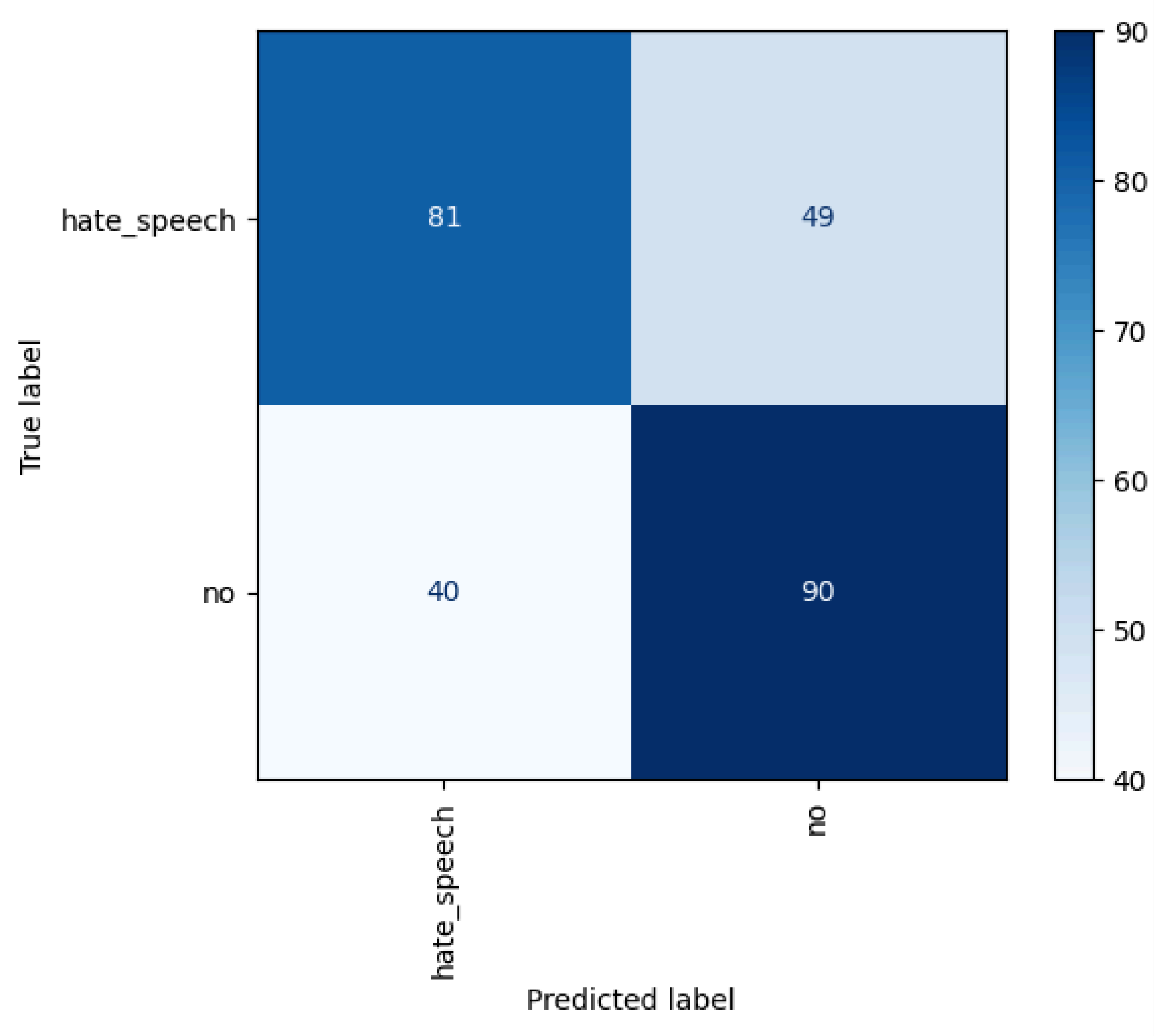


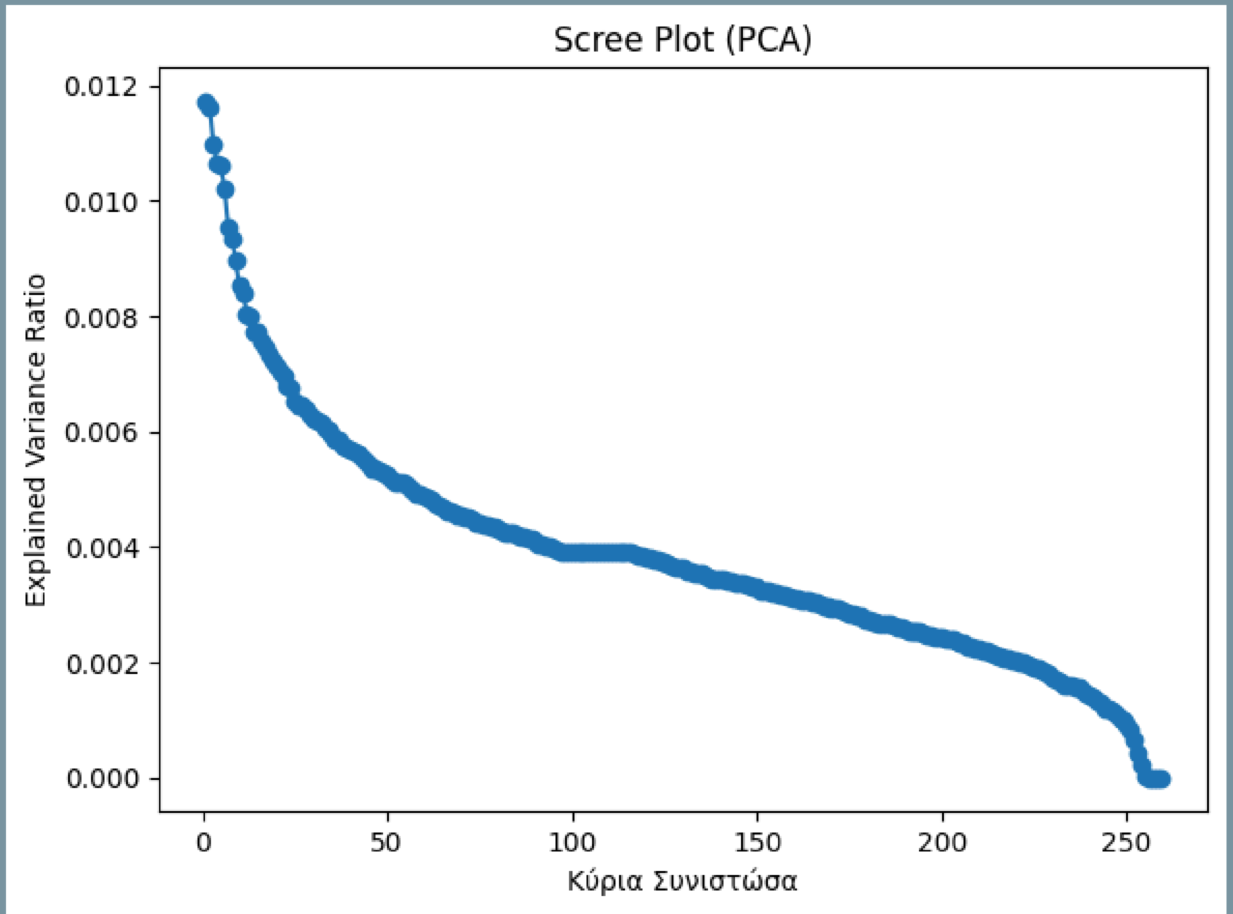

**Conclusion: SVM is the best-performing model.**

# Binary

## Without PCA

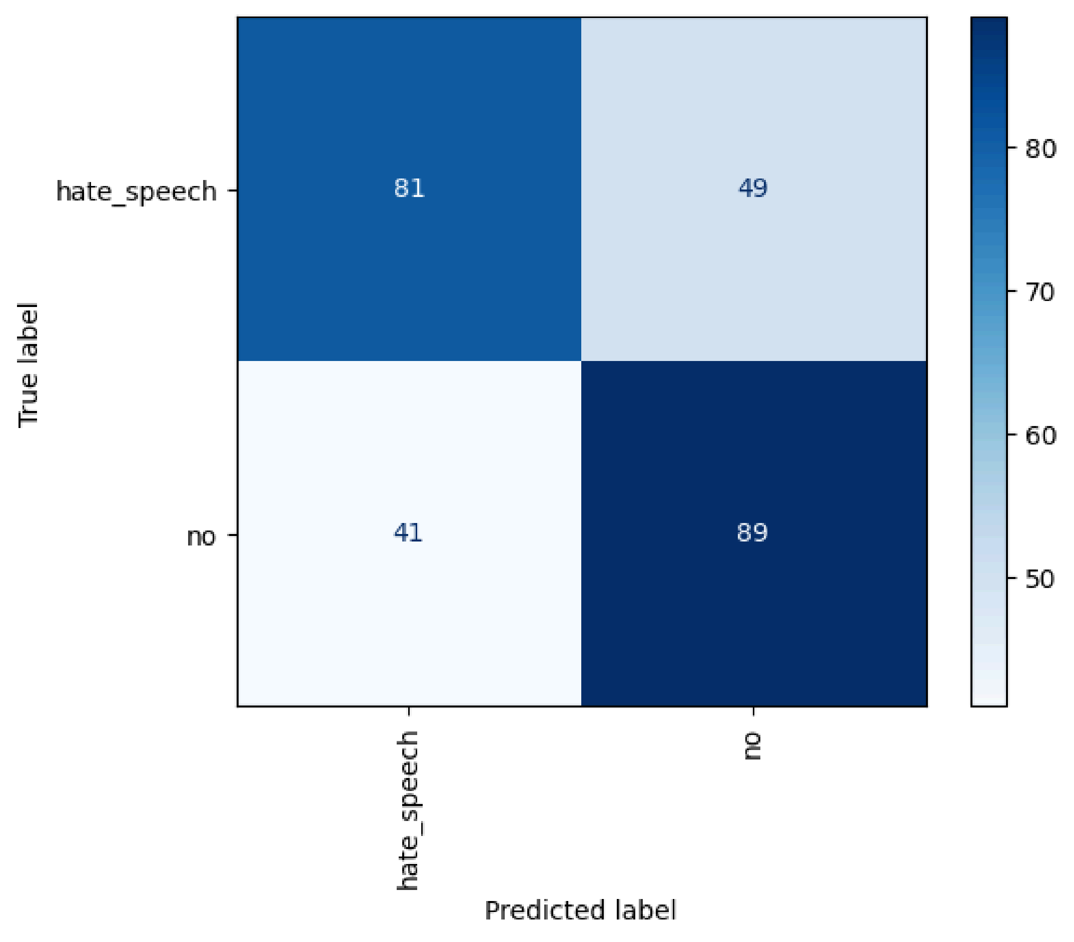
- macro - f1: 0.66




## With PCA

- Scree Plot was not show helpfull
- We chose 250 Components
- macro - f1: 0.65



# *Conclusion*

**SVM** -> Best model for text classification.

**Fusion** -> No significant improvement over SVM alone.

**Speech Analysis** -> Challenging due to technical limitations.

*Thank you*