CS F407

Artificial Intelligence

Hate Speech Detection

TEAM MEMBERS

- ISHIKA BHOLA
- YASHI KHANDELWAL
- RIYA SINGH
- JALI VIGNESHWARA REDDY

WHAT IS HATE SPEECH?

Hate speech is an attack towards a specific person or group based on characteristics such as race, ethnicity, religion, gender, age, disability. In recent years, the number of people using social media platforms has increased dramatically. These platforms have become a suitable place for people to express their feelings and anger toward each other.

NEED FOR HATE SPEECH DETECTION

Developing an AI model that can detect hate speech in online content has become essential to protect individuals and communities from harm. Hate speech detection models can help social media platforms and law enforcement agencies identify and remove hateful content, preventing it from spreading and causing harm. Identifying patterns and trends in hate speech, these models can help policymakers and organizations develop targeted interventions and strategies to address the root causes of hate speech and promote more inclusive and tolerant societies.

OUR IMPLEMENTATION

The idea of our project is based on genetic algorithm optimization used to obtain the optimal subset of features. We demonstrate how the genetic algorithm approach for every choice of text representation affects the classification performances. To get an idea of the accuracy of GA, we've also implemented Particle Swarm Optimisation and compared F1 scores of the same.

DATA PREPROCESSING

In the case of a hate speech detection model, data preprocessing plays a critical role in ensuring the accuracy and reliability of the model. By cleaning and transforming the raw data, the model can better identify patterns and trends in hate speech and differentiate it from non-hateful speech.

Overall, data preprocessing helps to improve the quality and relevance of the data used to train and evaluate the hate speech detection model, leading to a more effective and reliable AI model.

DATA PREPROCESSING

TOKENIZATION-

The first step is to transform the sample text into tokens. We transform everything to lowercase

STEMMING-

Stemming is also a very important step because for every word, they may have many variations like future or past tense of the words.

STOP WORDS-

Stop words are those words that need to be filtered out beforehand. Generally, they will be those words that are too common in a language.

DATA PREPROCESSING

MINIMUM DOCUMENT FREQUENCY-

There may be some words so rare that they appear in just one or two documents. Those words might be useless so we will remove them.

TF(TERM FREQUENCY)-

We will count the frequency of each term(token), which is called term frequency. The term with higher frequency is more related to that document.

TF-IDF(TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY)-

The inverse document frequency, denoted as idf(t, D), is a measure of whether the term is common or rare across all documents.

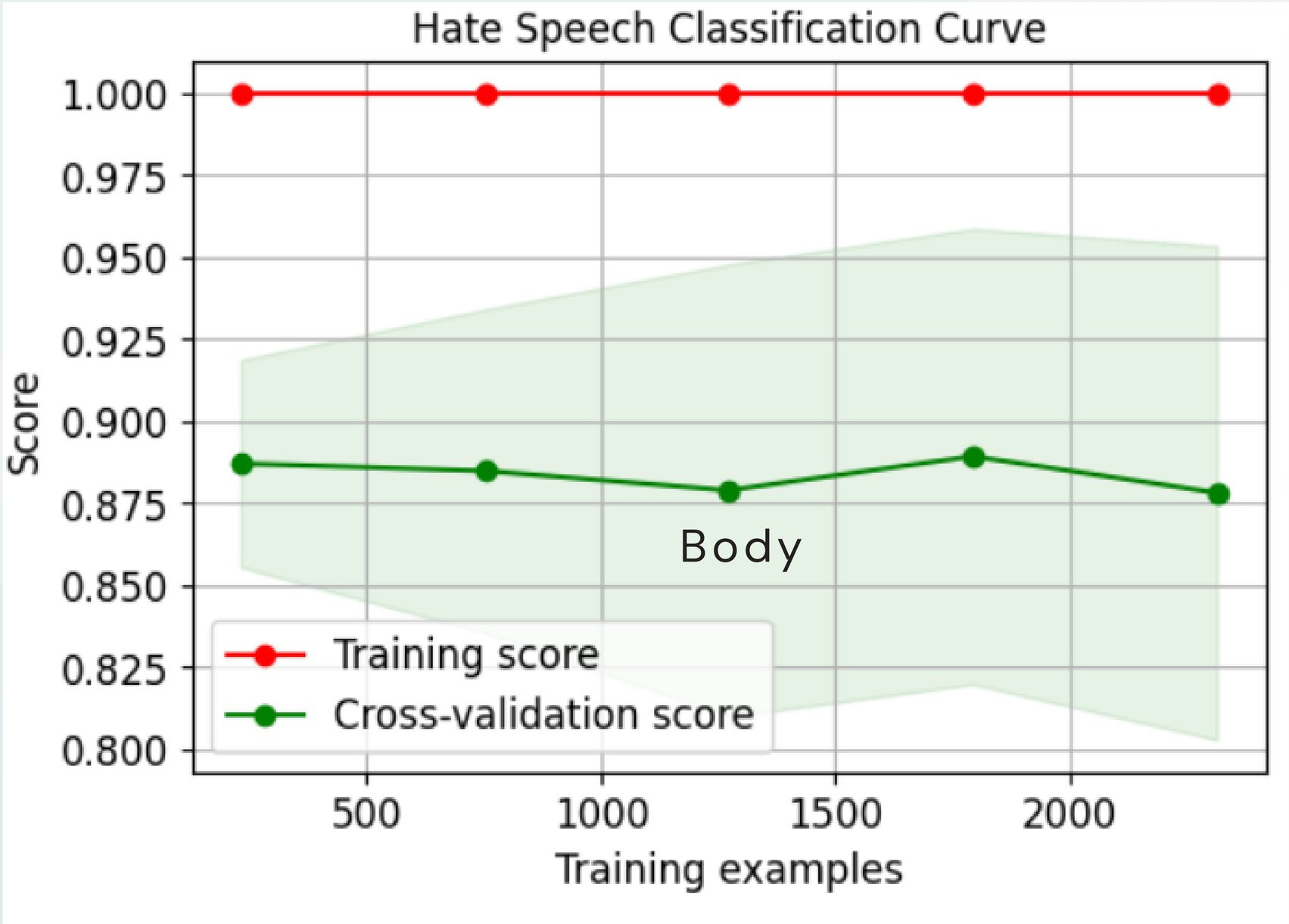
DECISION TREE

A decision tree is a type of supervised learning algorithm that is mostly used for classification tasks. It creates a model in the form of a tree structure that breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.
In the context of hate speech detection, decision trees can be trained on preprocessed text data to classify text as either hate speech or non-hate speech. The decision tree can be built by recursively partitioning the dataset into subsets based on the values of the features.

GENETIC ALGORITHM

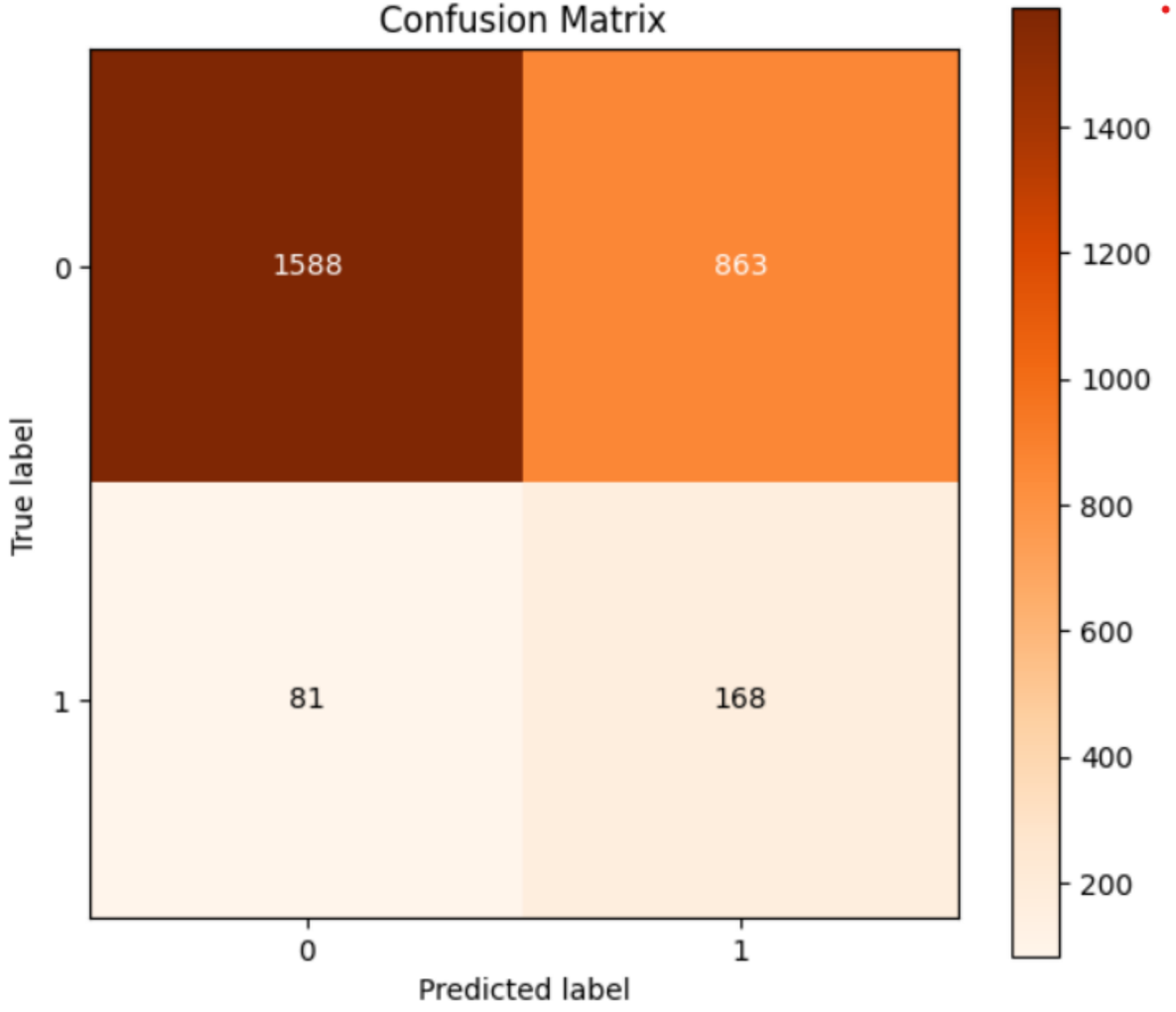
Genetic Algorithm is a major Heuristic Algorithm which mimics Darwin's theory of evolution. The main issues to be addressed in applying GAs to any problem are selecting an appropriate representation and an adequate evaluation function. The process of genetic algorithms is as follows. Genetic Algorithms randomly generate initial individuals to form an initial population encoding with different methods. Each individual consists of a variable gene that represents a solution to the given problem and is encoded by a chromosome. The genetic Algorithm design has to include the following three important operators: Selection, Crossover and Mutation. The best individuals with high Fitness Values are selected to breed a new generation.

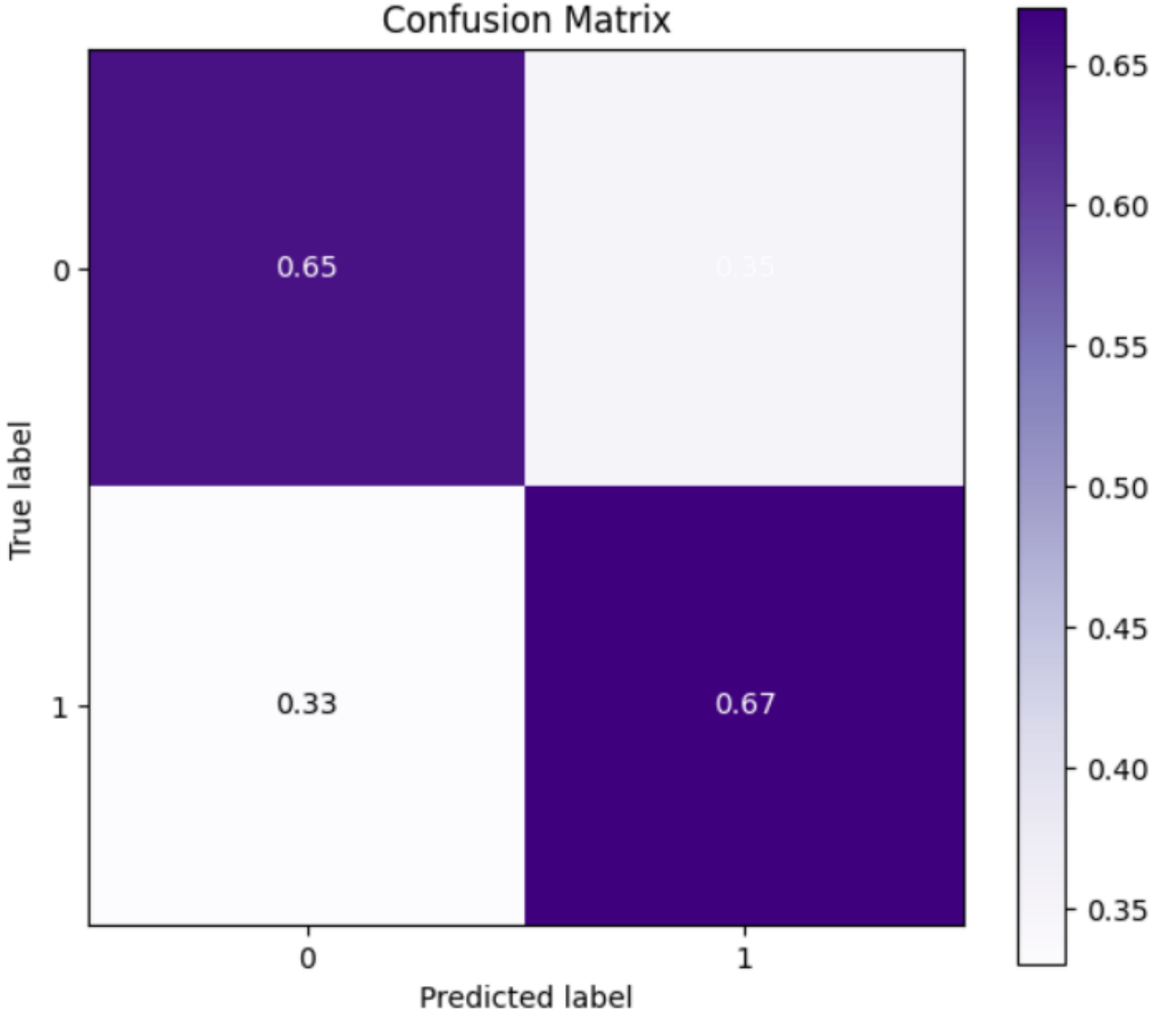
Population Size- 50



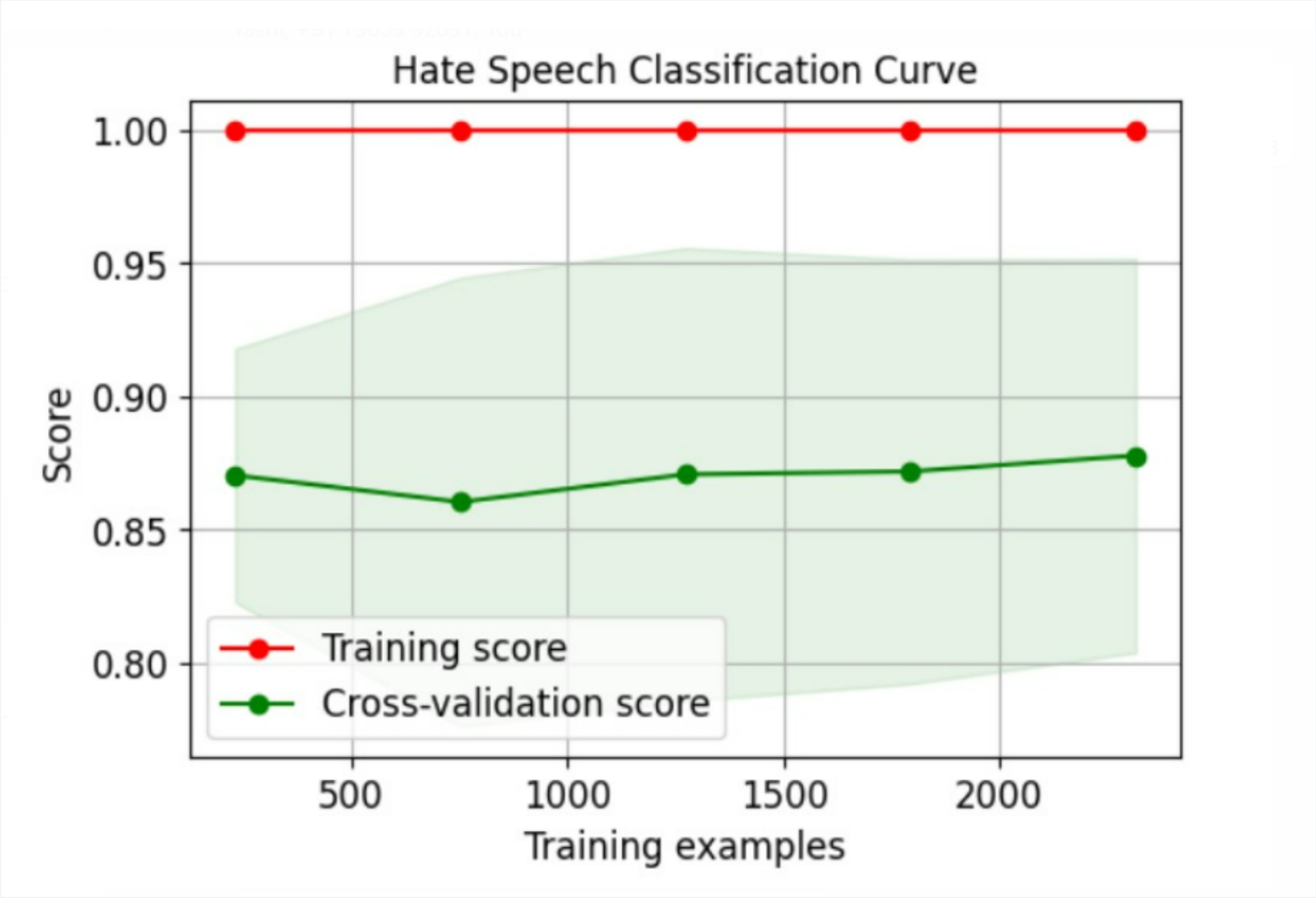
RESULTS-

```
Accuracy: 0.6137037037037038
Confusion Matrix:
 [[1472  979]
 [  64  185]]
```



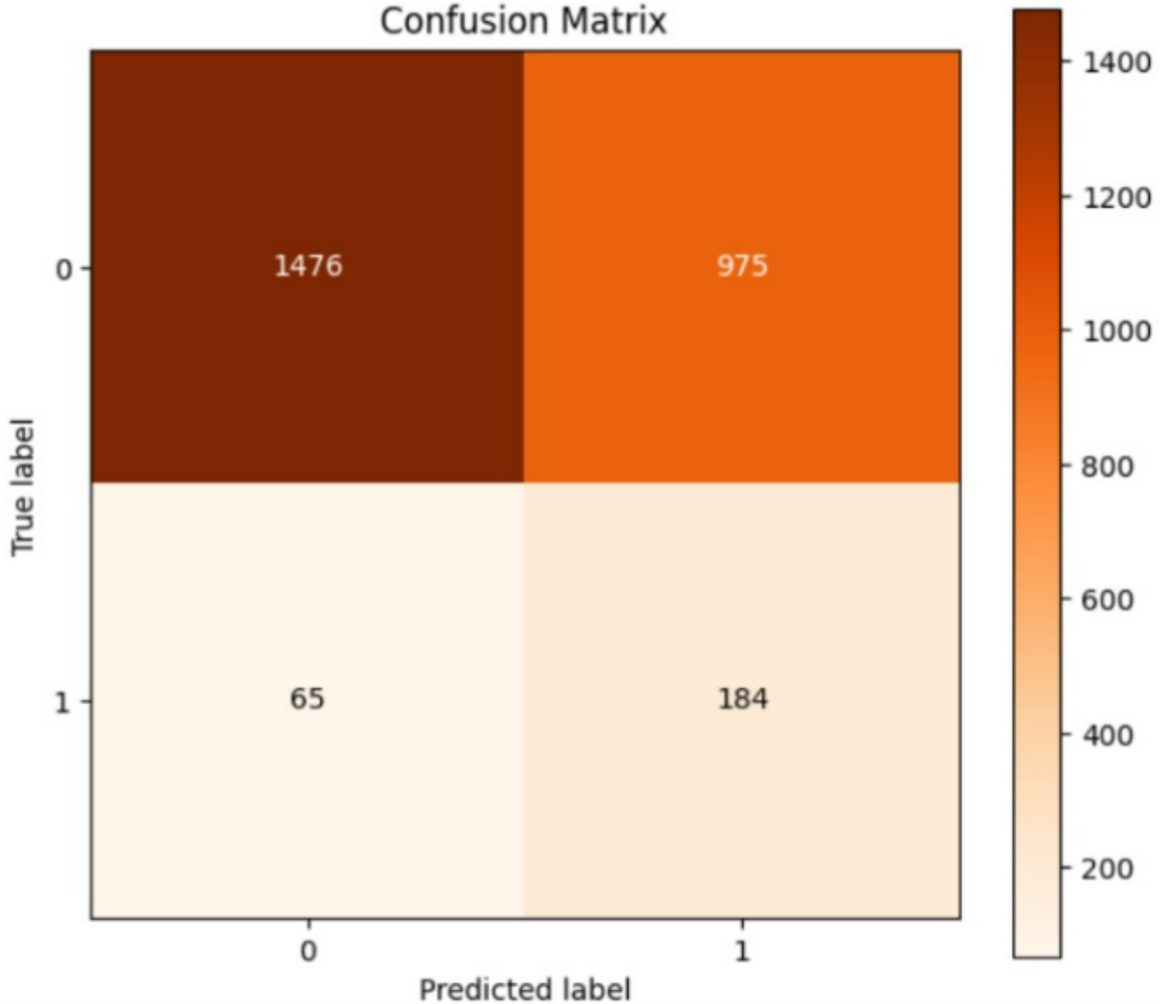


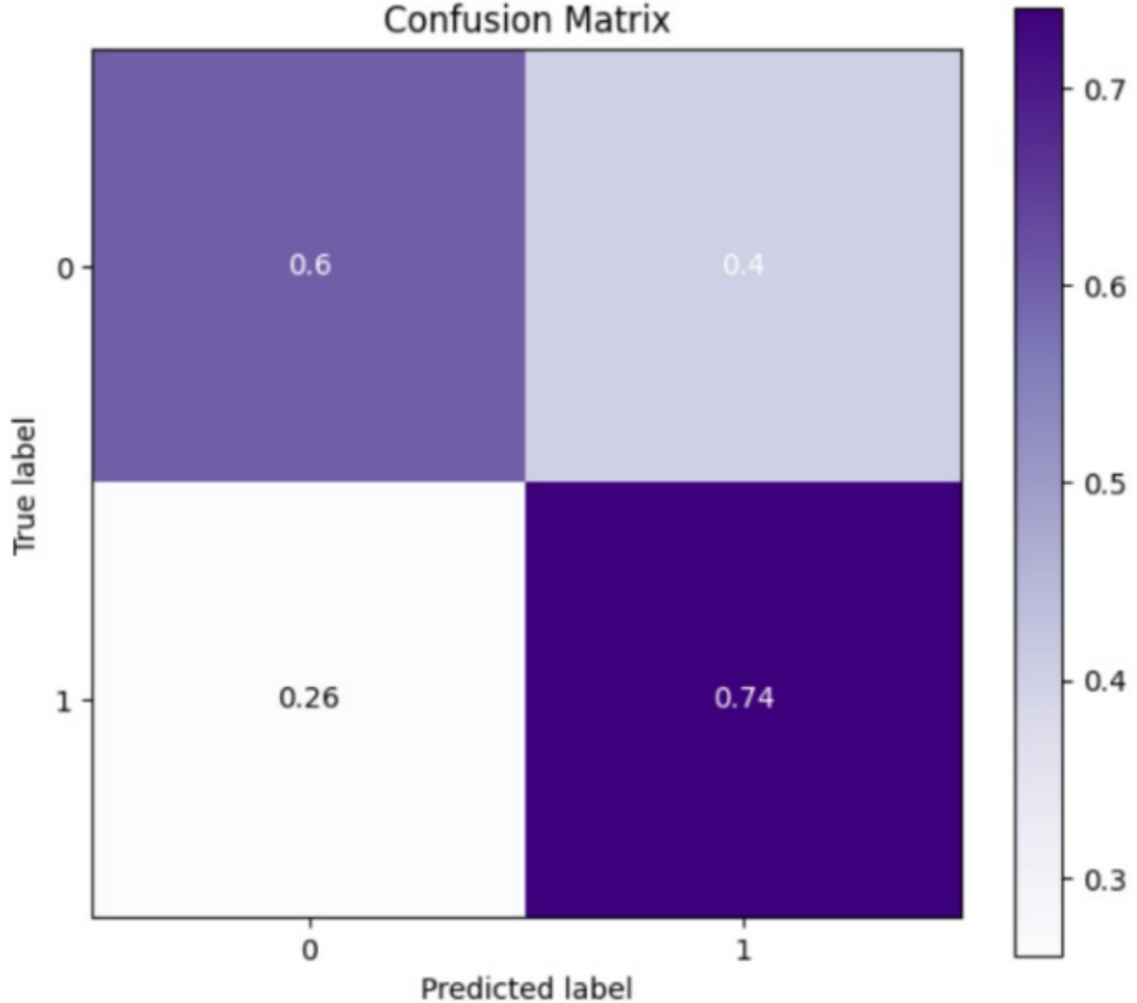
Population Size- 100



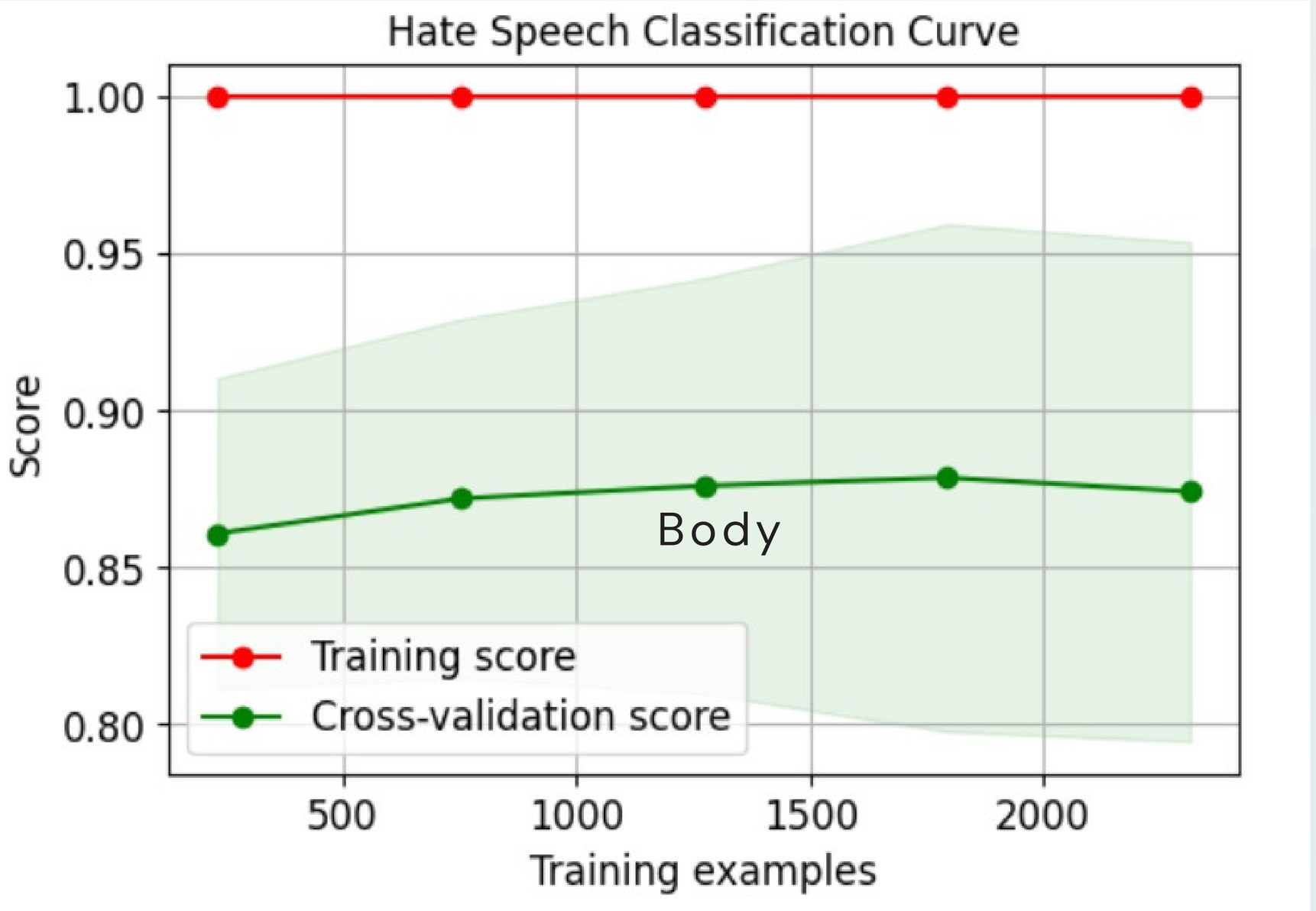
RESULTS-

```
Accuracy: 0.6148148148148148
Confusion Matrix:
 [[1476  975]
 [  65  184]]
```



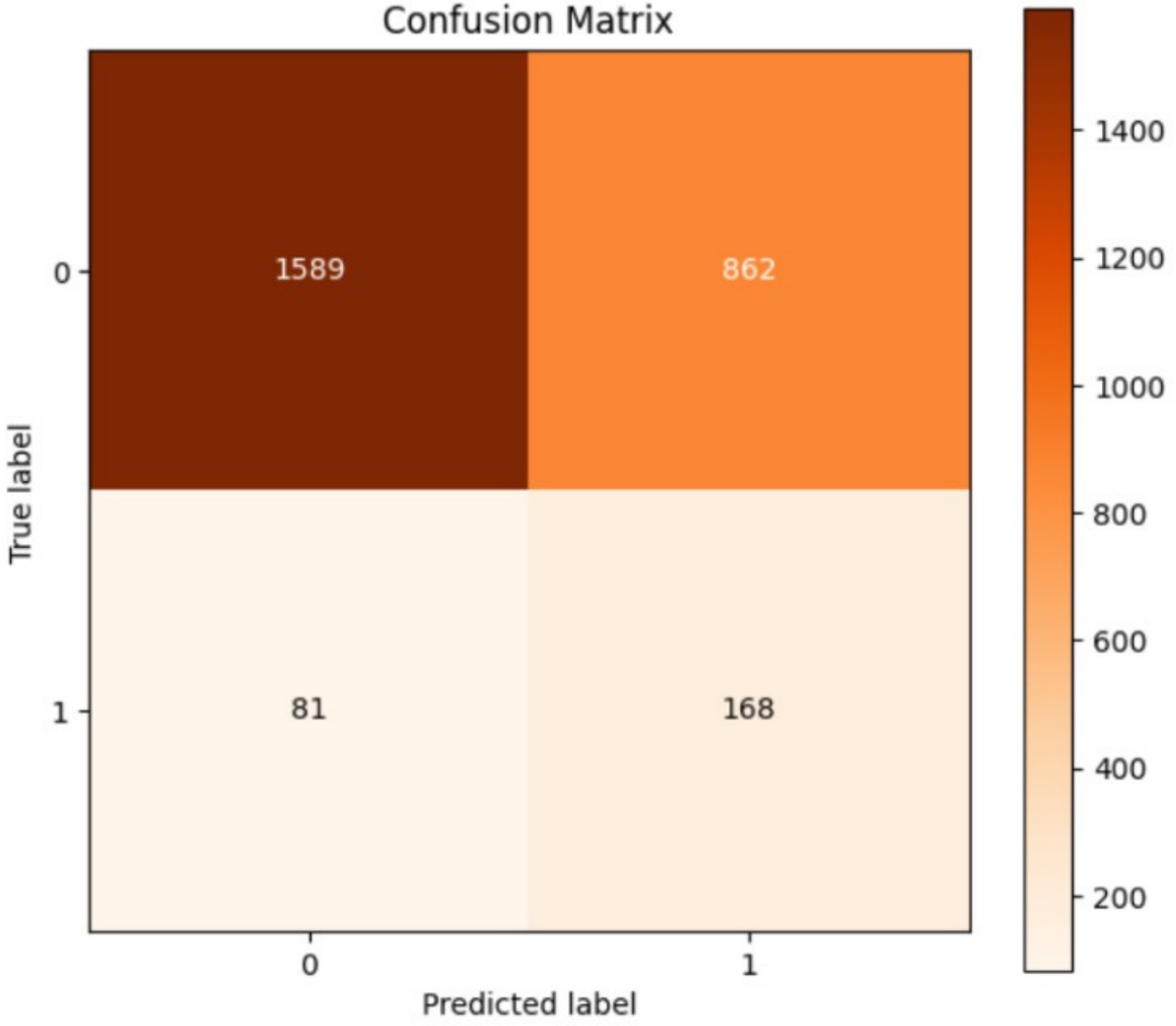


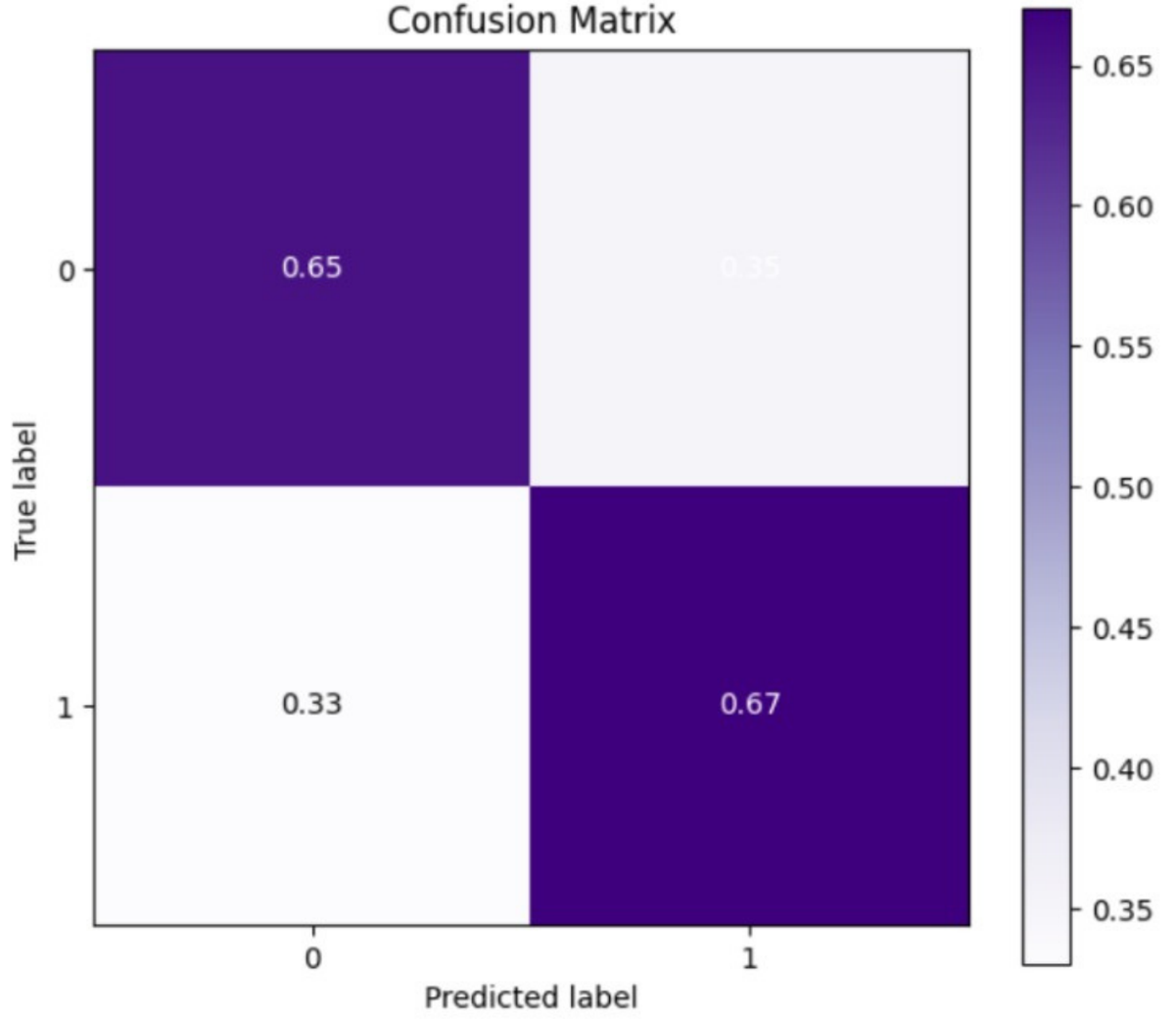
Population Size- 150



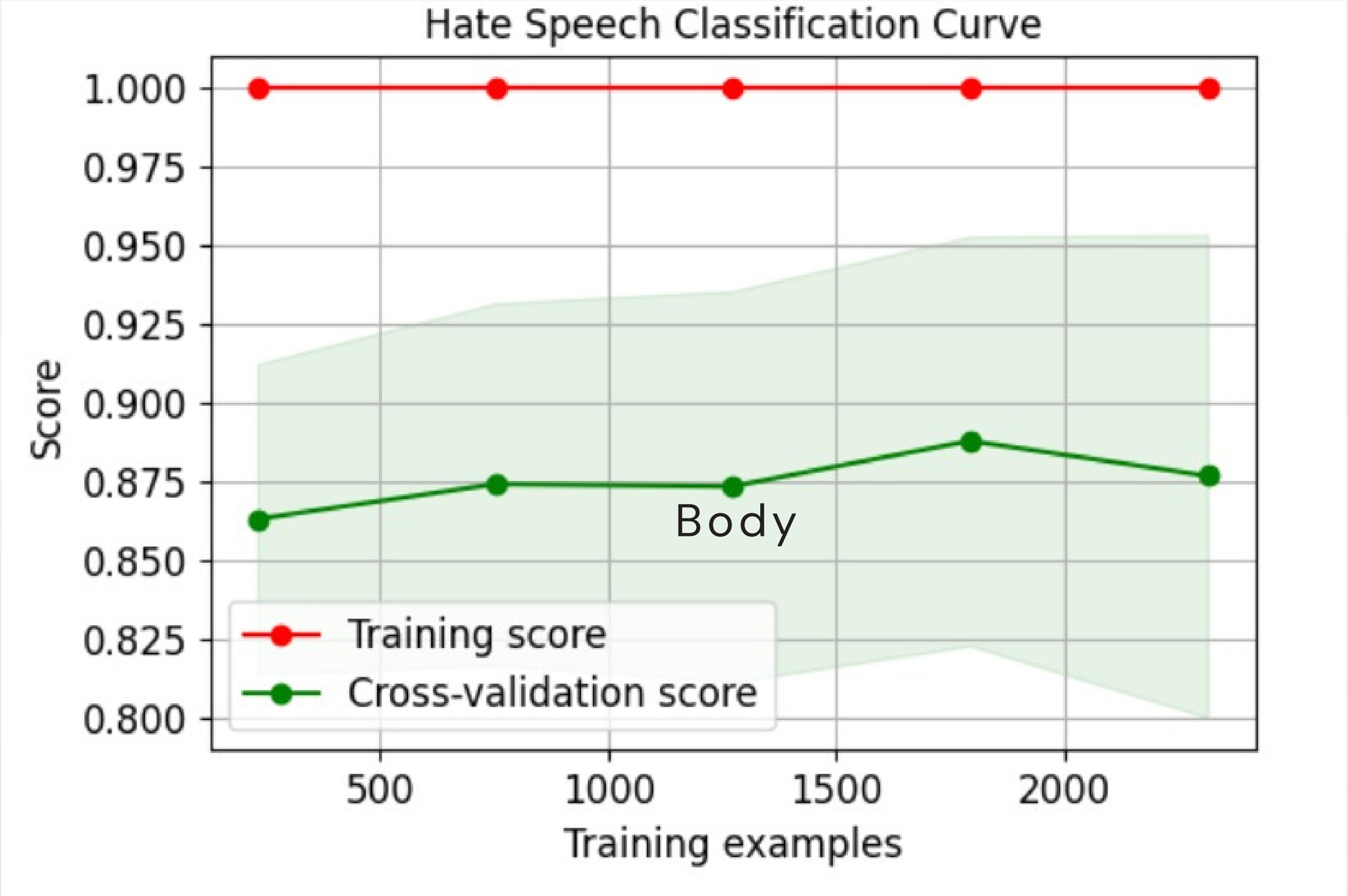
RESULTS-

```
Accuracy: 0.6507407407407407
Confusion Matrix:
 [[1589  862]
 [  81  168]]
```



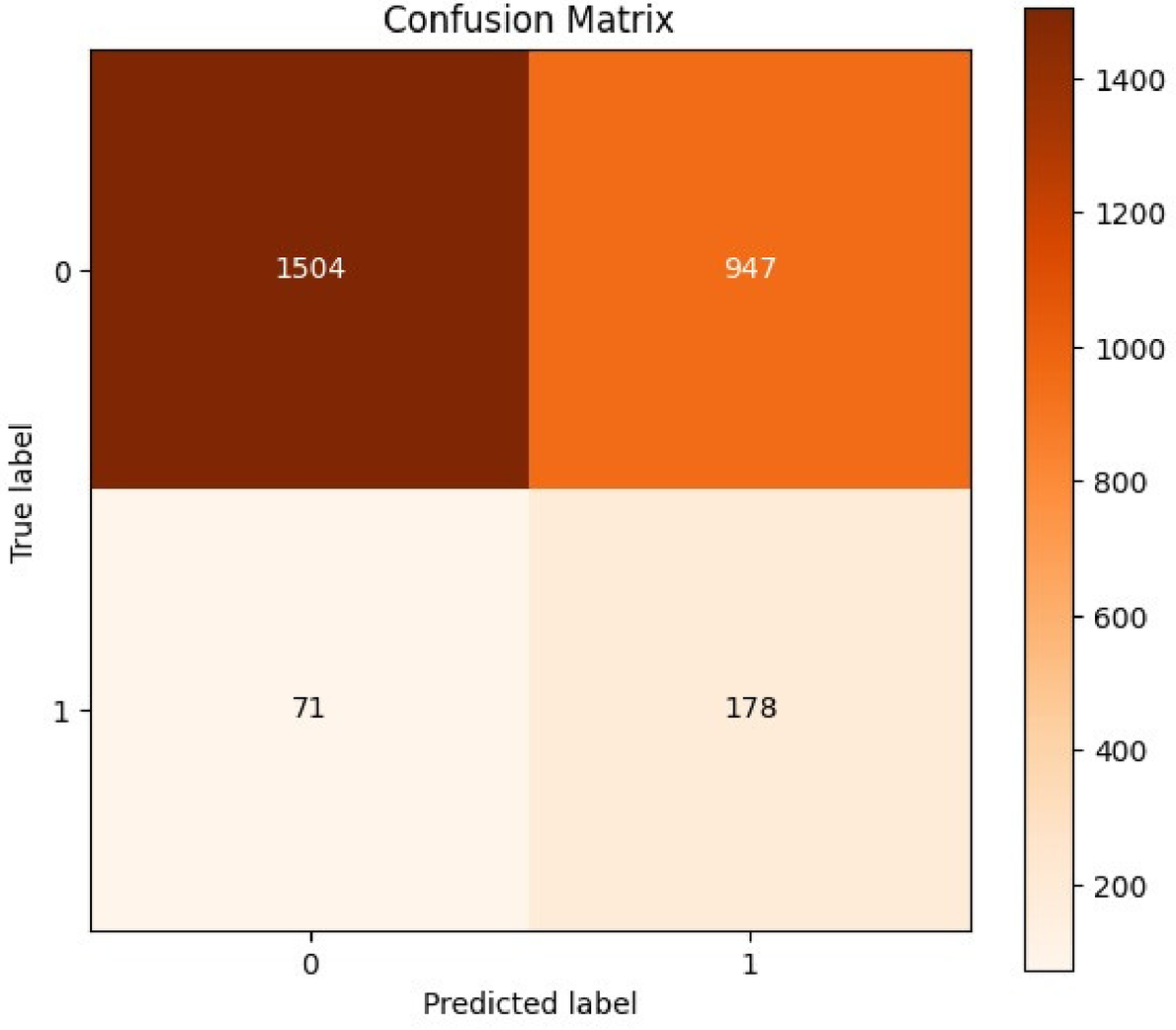


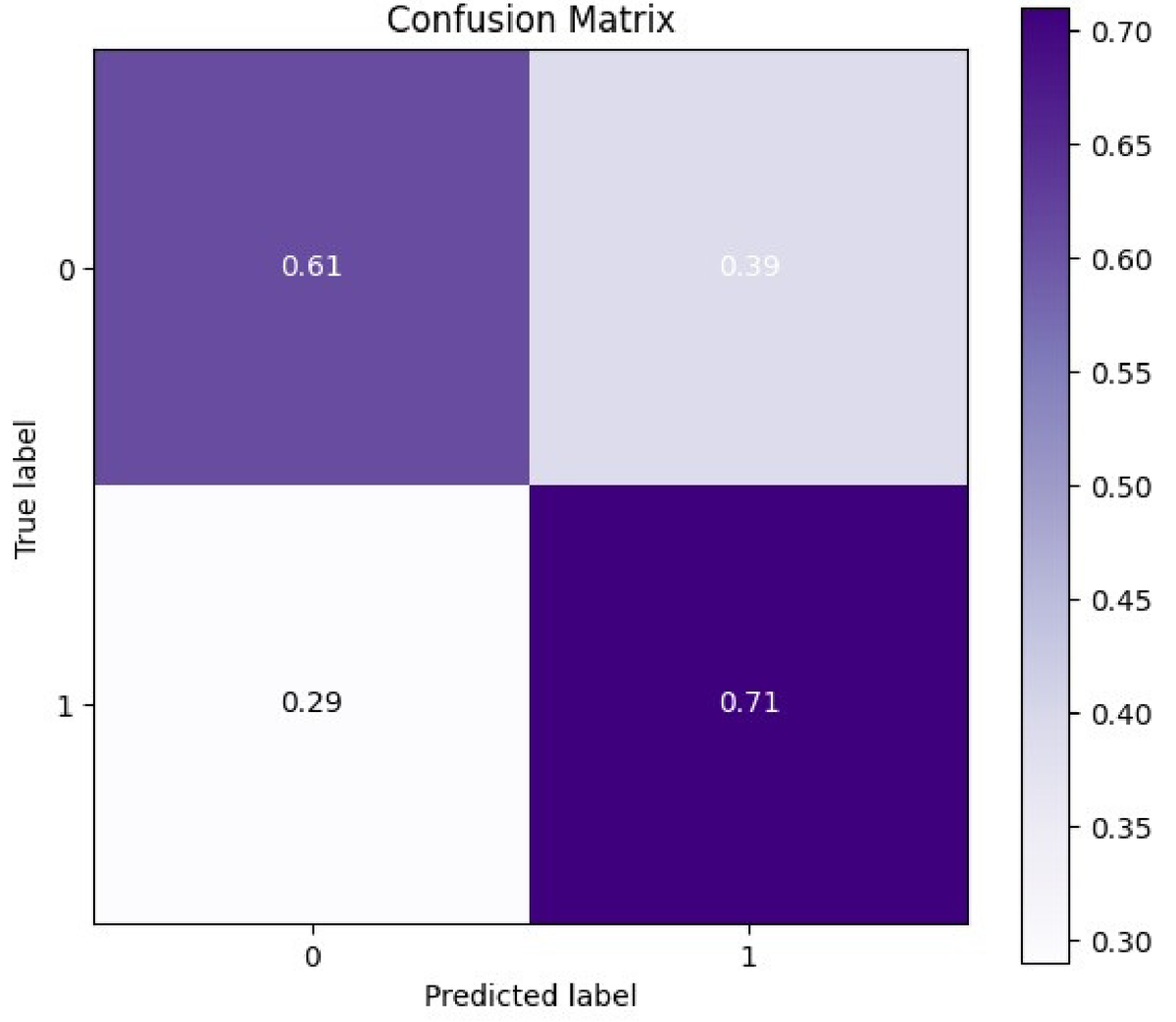
Population Size- 200



RESULTS-

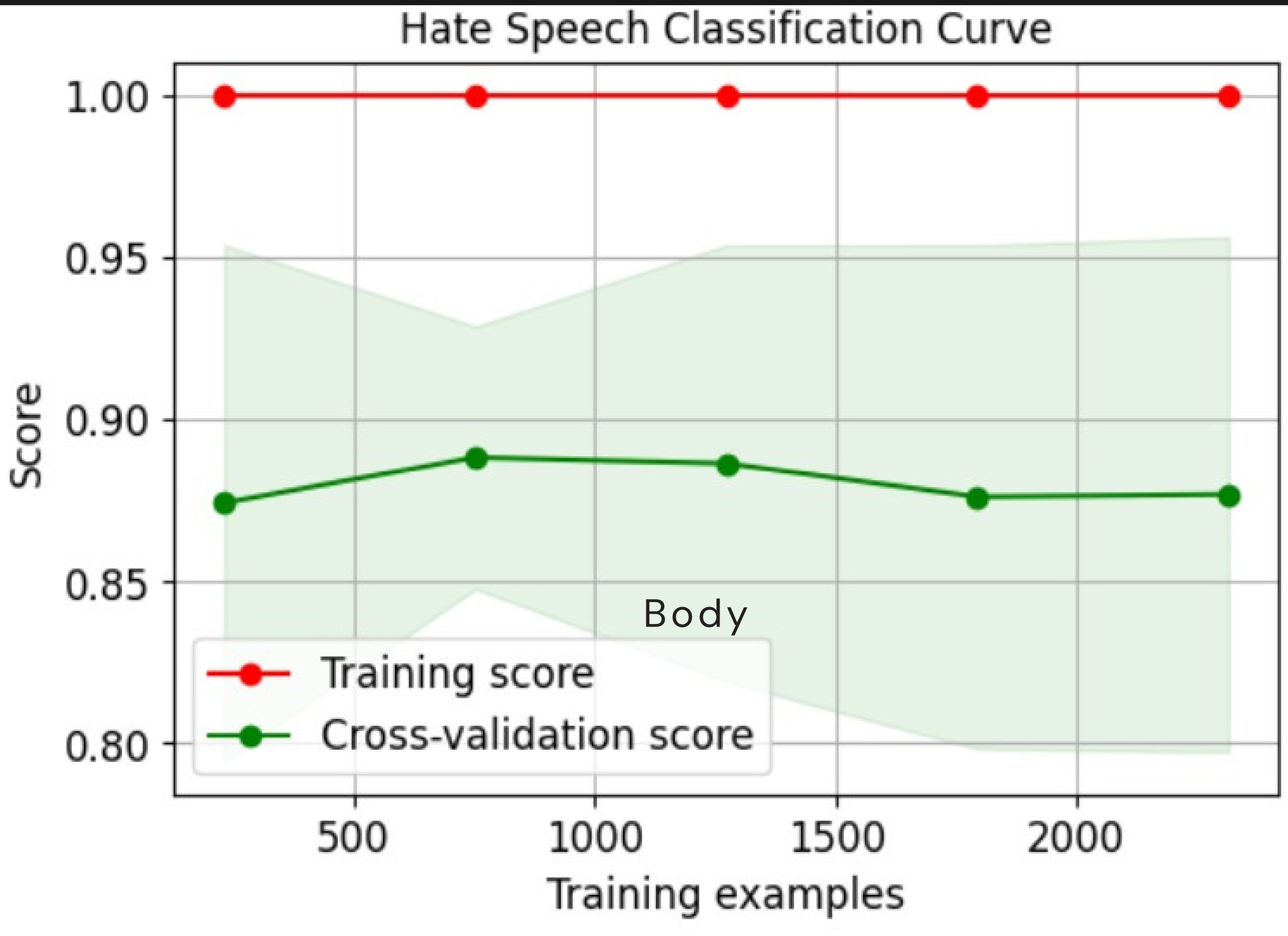
```
Accuracy: 0.622962962962963
Confusion Matrix:
 [[1504  947]
 [  71  178]]
```





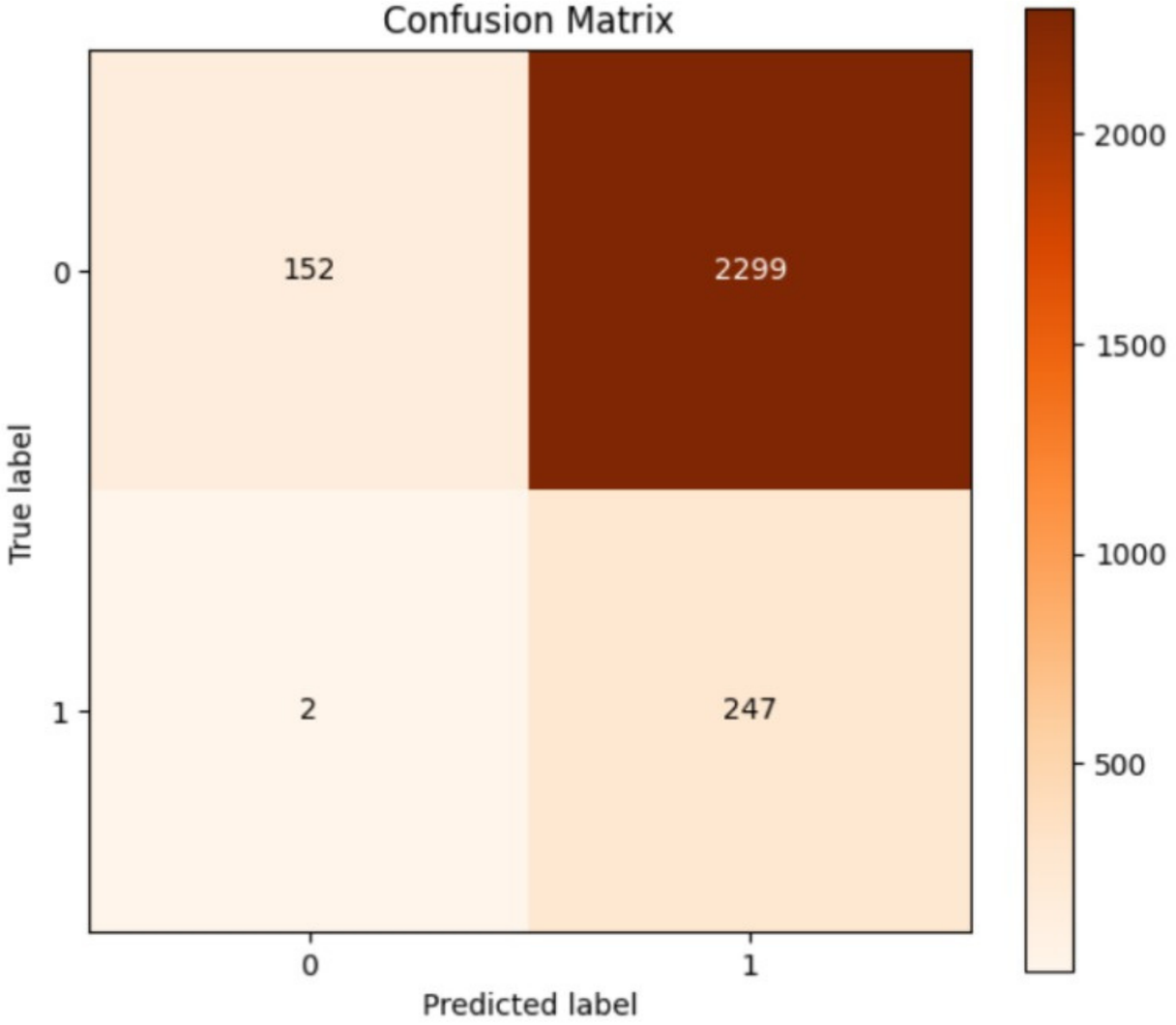
PARTICLE SWARM OPTIMISATION

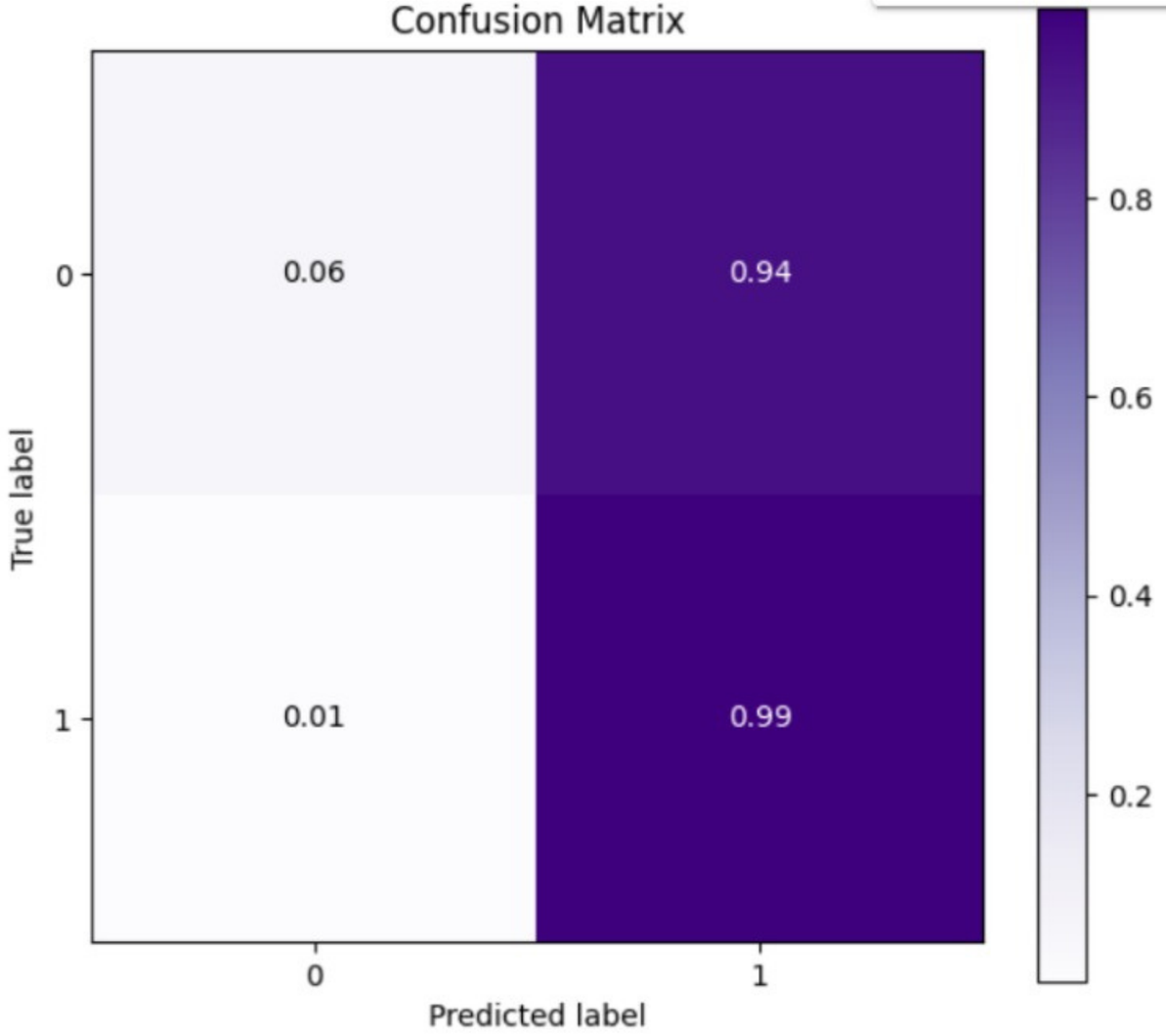
Particle Swarm Optimization (PSO) is a metaheuristic algorithm that can be used to optimize the parameters of a machine learning model for hate speech detection. PSO works by iteratively adjusting the position and velocity of particles in the swarm, aiming to find the optimal set of parameters that results in the highest accuracy. PSO's ability to escape local optima makes it a powerful tool for optimizing machine learning models. To apply PSO to the problem of hate speech detection, we can use it to optimize the parameters of a machine learning model, such as a support vector machine (SVM) or a neural network.


Hate Speech Classification Curve
1.00
0.95
0.90
0.85
0.80
Score
Body
Training score
Cross-validation score
500
1000
1500
2000
Training examples

RESULTS-

```
Accuracy: 0.14777777777777779
Confusion Matrix:
 [[ 152 2299]
 [   2  247]]
```





CONCLUSION

The goal was to prove the effectiveness of genetic algorithms attribute selection in text classification using machine algorithms for every text representation mode possible and a smaller size of feature subsets. The results are good enough to allow us to say: It is a good perspective. As for future work we will be interested in using other evolutionary and meta-heuristic algorithms or hybrid solution to improve textual document classification.

THANK YOU