CAPSTONE PROJECT

SARCASM DETECTION

PRESENTED BY

STUDENT NAME: CHIRUNOMULA VAMSHI KRISHNA BABU

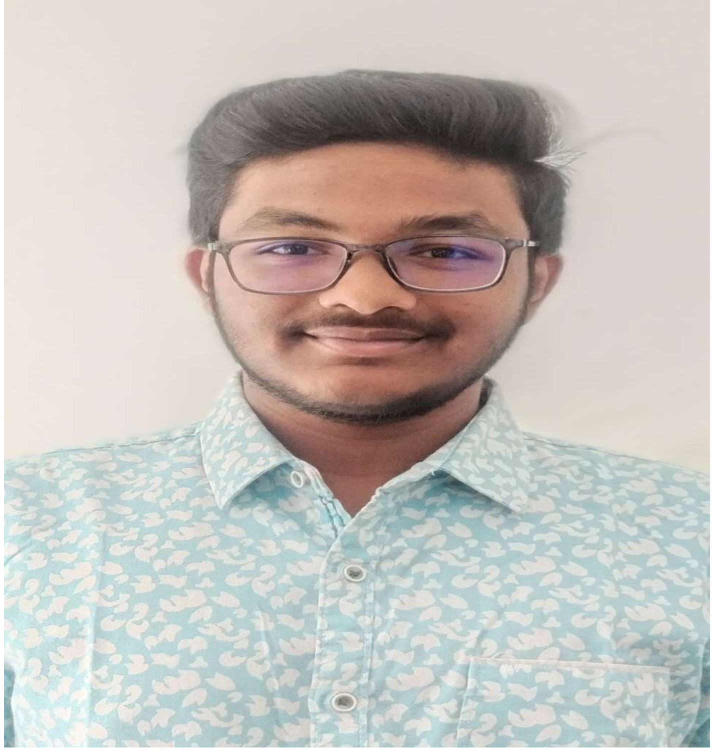
COLLEGE NAME: MAHINDRA UNIVERSITY

DEPARTMENT: ARTIFICIAL INTELLIGENCE

EMAIL ID:

CHIRUNOMULAVAMSHIKRISHNABABU@GMAIL.COM

AICTE STUDENT ID: STU67c7e49f922511741153439



OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

Sarcasm detection in text is challenging due to its nuanced and context-dependent nature. Traditional NLP methods often fail to capture sarcastic intent, leading to misinterpretations in sentiment analysis, chatbots, and social media monitoring. This project aims to build a robust machine learning model to classify text as sarcastic or non-sarcastic accurately.

NOTE:

Firstly, I used the Bernoulli naïve bayes algorithm later to increase the accuracy
of model the used

1. Logistic Regression
2. Random Forest
3. SVM with Linear Kernel
4. Ensemble Learing(Logistic, Bernoulli, SVC)

PROPOSED SOLUTION

- **Proposed Solution**

Data Collection:

Dataset: 26,709 headlines labeled as "Sarcasm" or "Not Sarcasm" (from Kaggle).

Data Preprocessing:

Transformed binary labels (0/1) to readable classes.

Used CountVectorizer and TF-IDF for text vectorization.

Model Selection:

Tested Bernoulli Naive Bayes, Logistic Regression, Random Forest, and SVM.

Implemented Ensemble Learning (Voting Classifier) for improved accuracy.

Deployment:

Interactive input to test custom sentences (e.g., "Cows lose their jobs as milk prices drop" → Predicted as "Sarcasm").

SYSTEM APPROACH

Libraries Used:

```
import pandas as pd, numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import VotingClassifier
from sklearn.metrics import classification_report
```

WorkFlow:

Text Vectorization → Train-Test Split → Model Training → Evaluation.

ALGORITHM & DEPLOYMENT

Algorithms:

- **Bernoulli Naive Bayes**: Baseline (84.48% accuracy).
- **Logistic Regression**: 84.54% accuracy.
- **Ensemble Model (Voting Classifier)**: **85.38% accuracy** (Best).

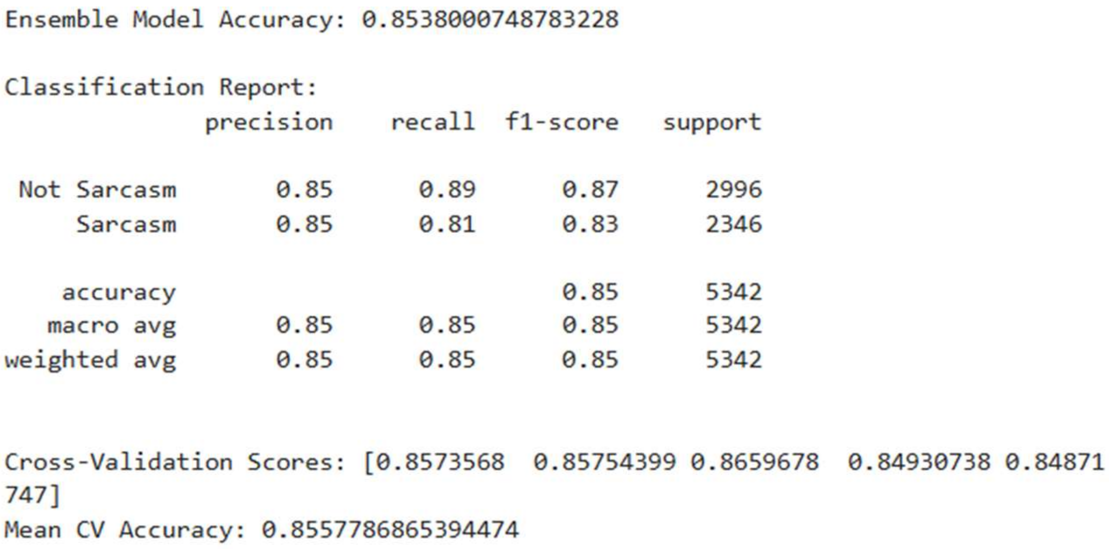
Key Metrics:

Classification Report:

	Precision	Recall	F1-Score
Not Sarcasm	0.85	0.89	0.87
Sarcasm	0.85	0.81	0.83

RESULT

Visualization:



```
Ensemble Model Accuracy: 0.8538000748783228

Classification Report:
              precision    recall  f1-score   support

 Not Sarcasm       0.85      0.89      0.87      2996
     Sarcasm       0.85      0.81      0.83      2346

    accuracy                           0.85      5342
   macro avg       0.85      0.85      0.85      5342
weighted avg       0.85      0.85      0.85      5342


Cross-Validation Scores: [0.8573568  0.85754399 0.8659678  0.84930738 0.84871
747]
Mean CV Accuracy: 0.8557786865394474
```

Sample Output:

Input: "Cows lose their jobs as milk prices drop"
Output: ['Sarcasm']

CONCLUSION

- The ensemble model achieved **85.38% accuracy**, outperforming individual classifiers.
- Challenges: Sarcasm heavily relies on context; neutral phrases may be misclassified.

FUTURE SCOPE

1. Incorporate contextual embeddings (BERT, GPT).
2. Expand dataset to include multilingual sarcasm.
3. Deploy as a web app/API for real-time use.

REFERENCES

1.Kaggle Sarcasm Dataset.

2.Scikit-learn Documentation.

3.Research papers on NLP for sarcasm detection.

GitHub Link: VKB2005/Sarcasm-Detection

Thank you