**C.V. RAMAN GLOBAL UNIVERSITY**

BHUBANESWAR, ODISHA-752054 DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING



AI & DS CASE STUDY REPORT

TOPIC:- REDDIT COMMENTS

TEAM DETAILS: GROUP- 6

SEMESTER-6TH

|  |  |
| --- | --- |
| **NAME** | **REGISTRATION NO.** |
| GAURAV GAUTAM PRADHAN | CL20250106018924106 |
| ARUN KUMAR JOJO | CL2025010601896826 |
| M ABHISHEK | CL2025010601897634 |
| ROHAN SAHU | CL20250106018919101 |

UNDER THE GUIDANCE OF:-

**MR. PARTHA CHATTERJEE**

### **DECLARATION**

We hereby declare that the case study report entitled **"** REDDIT COMMENTS**"** is an original work completed by us as part of our academic curriculum. This case study report is submitted to the Department of Computer Science Engineering, C.V. Raman Global University, Bhubaneswar, Odisha, under the guidance of Mr. ParthaChatterjee **Cranes Varsity Trainer**. All the information, data, designs, and analysis included in this report are authentic and reflect our own efforts, except where specified otherwise. Any references or resources used have been properly acknowledged in the report's references section.

This project has been conducted solely for academic purposes, and any resemblance to existing products or systems is purely coincidental. We affirm that this work has not been submitted, either fully or in part, for any other course or assessment.

Signature of Supervisor

Signature of Trainer

Signature of students

**ACKNOWLEDGEMENT**

We wish to extend our deepest appreciation to the esteemed faculty **(Mr. Partha Chatterjee)** of **[Cranes Varsity]** for granting us the exceptional privilege to partake in the semester-long experiential learning program. Their unwavering guidance, invaluable support, and profound mentorship have acted as the cornerstone of our academic journey, profoundly enriching my knowledge and fostering the development of practical skills within our field of study.

Furthermore, we stand in admiration of the remarkable host organizations and mentors who graciously embraced our presence and contributed to this transformative experience. Their wisdom and generosity have left an indelible mark on our professional growth, providing a wealth of insight and opportunities that we will forever treasure.

This program has transcended the boundaries of a traditional educational experience, illuminating an extraordinary path toward academic and professional excellence. The knowledge we've gained, the challenges we've overcome, and the connections we've forged have collectively shaped me into a more capable and empowered individual. With immense gratitude, we recognize the profound impact this program has had on our journey, and we eagerly anticipate the continued evolution of our academic and professional endeavors.

**ABSTRACT**

This project implements a **Hierarchical BERT-based deep learning model** for **sarcasm detection in social media comments** using the **train-balanced-sarcasm** dataset. The system processes a large corpus of Reddit comments by combining the parent\_comment and comment fields to form contextual input text.

The pipeline includes:

* **Text preprocessing and tokenization** using the bert-base-uncased tokenizer from Hugging Face.
* A custom **Hierarchical BERT architecture**, which integrates:
  + A pretrained BERT encoder for contextual embeddings,
  + A BiLSTM layer to capture sequential dependencies,
  + A CNN layer to extract local patterns, and
  + Fully connected layers for binary classification (sarcastic or not sarcastic).

The model is trained and evaluated using **5-fold cross-validation**, with metrics including **accuracy**, **precision**, **recall**, **F1-score**, and **ROC-AUC**. The training process leverages **mixed precision** for performance optimization and includes early stopping based on validation loss.

This approach demonstrates a robust, hybrid architecture for nuanced text classification tasks, blending transformer-based language understanding with sequence and pattern-based enhancements.

**CONTENT**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Topics** | **Page no.** |
| 1. | Introduction | 6 |
| 2. | Objectives | 7 |
| 3. | Features | 8-9 |
| 4. | Technologies and tools used | 10 |
| 5. | Implementation | 11-12 |
| 6. | Source Code | 13-19 |
| 7. | Output | 20 |
| 8. | Future Work | 21 |
| 9. | Conclusion | 22 |
| 10. | References | 23 |

**INTRODUCTION**

Detecting sarcasm in textual data is a challenging yet crucial task in natural language processing (NLP), especially in the context of social media platforms like Reddit, where user-generated content is abundant and often ambiguous. Sarcasm can mislead sentiment analysis models, content moderation systems, and conversational AI, making accurate identification essential for downstream applications.

This project presents a robust sarcasm detection system built on top of **BERT (Bidirectional Encoder Representations from Transformers)**, enhanced with **hierarchical context processing**. The model utilizes both the **parent comment** and the **current comment** to understand contextual clues that may indicate sarcasm.

The system architecture comprises several deep learning components:

* A **pretrained BERT model** for capturing rich contextual embeddings,
* A **Bidirectional LSTM (BiLSTM)** to model sequence dependencies,
* A **Convolutional Neural Network (CNN)** to extract localized features, and
* A **fully connected classifier** for binary sarcasm classification.

To ensure comprehensive evaluation, the model undergoes **5-fold cross-validation** using the balanced sarcasm dataset. The implementation leverages **TensorFlow and Hugging Face Transformers**, with mixed precision training enabled for better performance on modern hardware.

This hierarchical approach not only improves the understanding of conversational context but also demonstrates the effectiveness of combining transformer-based embeddings with traditional sequence and pattern learning methods for nuanced language tasks like sarcasm detection.

**OBJECTIVES**

The primary objective of this project is to develop an accurate and context-aware sarcasm detection model using deep learning techniques. Specifically, the goals are:

* To leverage contextual information from both parent and child comments for better sarcasm understanding.
* To fine-tune a pretrained BERT model for encoding nuanced language representations.
* To integrate hierarchical layers such as BiLSTM and CNN on top of BERT to capture sequential and local patterns in textual data.
* To train and evaluate the model using 5-fold cross-validation for robust performance measurement.
* To assess the model’s effectiveness using key metrics: accuracy, precision, recall, F1 score, and ROC-AUC.
* To implement efficient data pipelines and training routines using TensorFlow and Hugging Face Transformers.

Ultimately, this project aims to build a scalable and generalizable sarcasm detection system that can be applied to real-world conversational AI and sentiment analysis tasks.

**FEATURES**

**Features**

1. **Context-Aware Input Processing**
   * Merges **parent\_comment** and **comment** using a [SEP] token to retain conversational context.
   * Converts text to lowercase and handles missing values for preprocessing robustness.
2. **Tokenization with BERT**
   * Uses **Hugging Face’s BERT tokenizer** (bert-base-uncased) for powerful subword tokenization.
   * Applies truncation, padding, and returns attention masks and token type IDs.
3. **Hierarchical BERT Architecture**
   * Combines a **pretrained BERT model** with:
     + **BiLSTM**: Captures sequential dependencies.
     + **CNN Layer**: Detects local n-gram features.
     + **Dense and Dropout Layers**: Improve learning and reduce overfitting.
     + **Mean Pooling and Global Max Pooling**: Enhance contextual summarization.
4. **Binary Classification Output**
   * Outputs a **single sigmoid-activated value** to predict sarcastic (1) or not sarcastic (0).
5. **Fine-tuning of Pretrained BERT**
   * Enables bert\_model.trainable = True to allow gradient updates for better performance.
6. **5-Fold Cross-Validation**
   * Ensures robust model evaluation across diverse splits.
   * Collects metrics: **accuracy**, **precision**, **recall**, **F1-score**, and **ROC-AUC**.
7. **Metrics and Visualization**
   * Calculates all major classification metrics.
   * Plots **confusion matrix** for the final fold to visualize prediction performance.
8. **Training Enhancements**
   * Uses **Cosine Learning Rate Decay** for smooth optimization.
   * Employs **EarlyStopping** to prevent overfitting.
   * Enables **mixed precision training** for faster and more memory-efficient training.
9. **Modular Functions**
   * Clearly separates tokenization, preprocessing, model creation, dataset building, and training.

**TECHNOLOGIES AND TOOLS USED**

1. Python
   * Core programming language for the entire implementation.
2. TensorFlow
   * For building and training the deep learning model (including custom model layers and mixed precision training).
3. Transformers (Hugging Face)
   * Used to load pretrained BERT tokenizer and BERT model for extracting contextual embeddings from text.
4. Pandas
   * For data loading, cleaning, and manipulation.
5. NumPy
   * For numerical operations and setting random seeds for reproducibility.
6. Scikit-learn (sklearn)
   * + Cross-validation (KFold)
     + Evaluation metrics (accuracy\_score, precision\_score, recall\_score, f1\_score, roc\_auc\_score)
     + Confusion matrix
7. Matplotlib & Seaborn
   * For visualizing the confusion matrix of the model predictions.
8. Keras Callbacks (EarlyStopping)
   * To prevent overfitting and stop training early when validation loss stops improving.
9. Keras Layers
   * Includes Dense, Dropout, LSTM, Bidirectional, Conv1D, GlobalAveragePooling1D, GlobalMaxPooling1D for model architecture.
10. Mixed Precision Training (TensorFlow)

* Improves training speed and reduces memory usage by using float16 computations.

**IMPLEMENTATION**

**1. Data Loading and Preprocessing**

* The dataset train-balanced-sarcasm.csv is loaded using **Pandas**.
* Only the first 50,000 samples are used for demonstration (due to computational limits).
* The parent\_comment and comment columns are combined to form a single input text for BERT.
* Missing values are removed, and text is converted to lowercase for consistency.

**2. Tokenization**

* The **BERT tokenizer** from Hugging Face is used to tokenize the input text.
* The text is pre-tokenized with a max length of 128 tokens, using padding and truncation.
* The tokenized inputs include:
  + input\_ids
  + attention\_mask
  + token\_type\_ids

**3. Model Definition – Hierarchical BERT**

* A custom class HierarchicalBERT is defined using TensorFlow’s Model API.
* **Model Architecture Includes**:
  + BERT embeddings as the base layer.
  + Dense + Dropout layer for sentence encoding.
  + **BiLSTM** layer for sequential context understanding.
  + **Conv1D + GlobalMaxPooling1D** for feature extraction.
  + Fully connected (Dense) layers with dropout.
  + Final output layer with sigmoid activation for binary classification.

**4. Model Training and Evaluation**

* Uses **5-fold cross-validation** with KFold from scikit-learn.
* For each fold:
  + Data is split into training and validation sets.
  + Model is trained with:
    - **Adam optimizer** and **Cosine Decay learning rate** schedule.
    - **EarlyStopping** callback to avoid overfitting.
  + Predictions are made and evaluated using:
    - **Accuracy**
    - **Precision**
    - **Recall**
    - **F1 Score**
    - **ROC-AUC Score**
* Confusion Matrix is visualized for the final fold using matplotlib and seaborn.

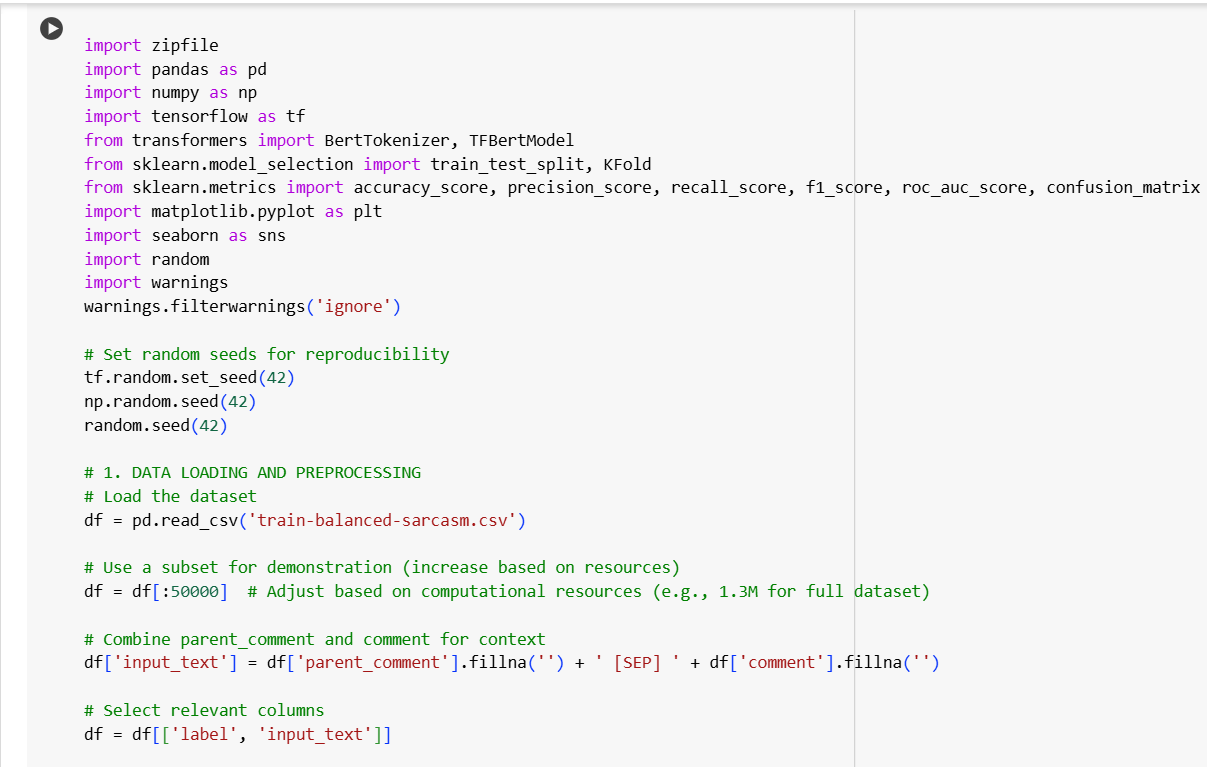
**5. Mixed Precision Training**

* Enabled to boost performance on supported hardware (e.g., GPUs with Tensor Cores).

**6. Final Output**

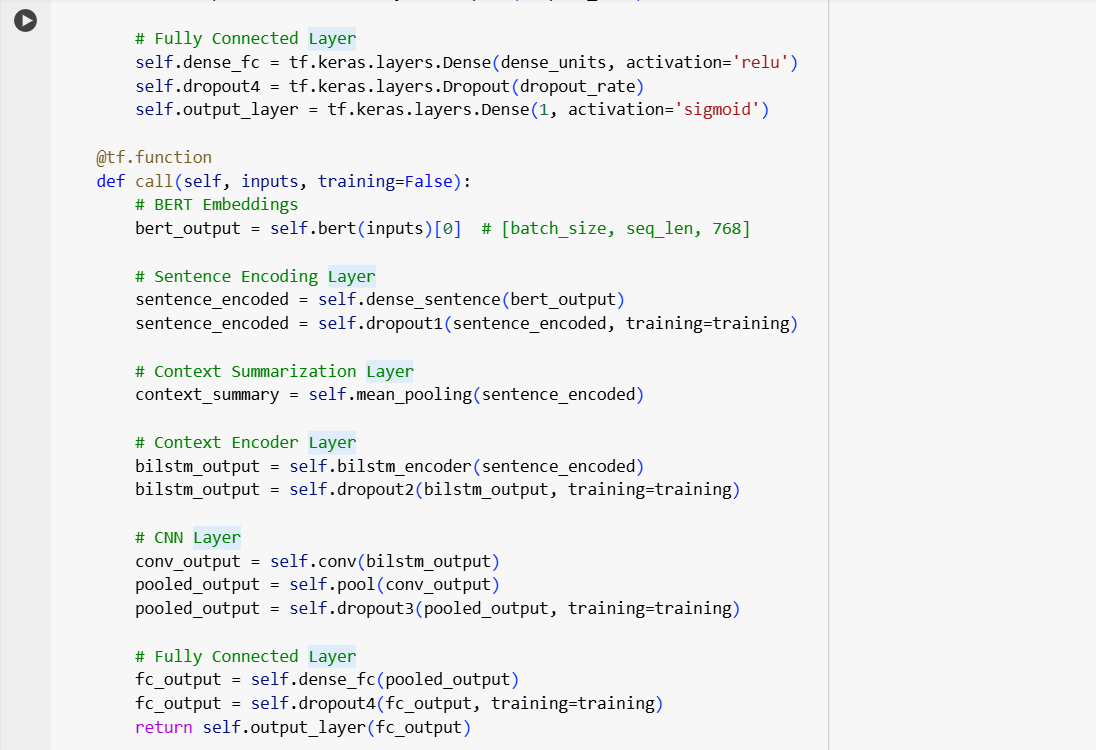
* Metrics are printed for each fold.
* Average performance across all folds is displayed.

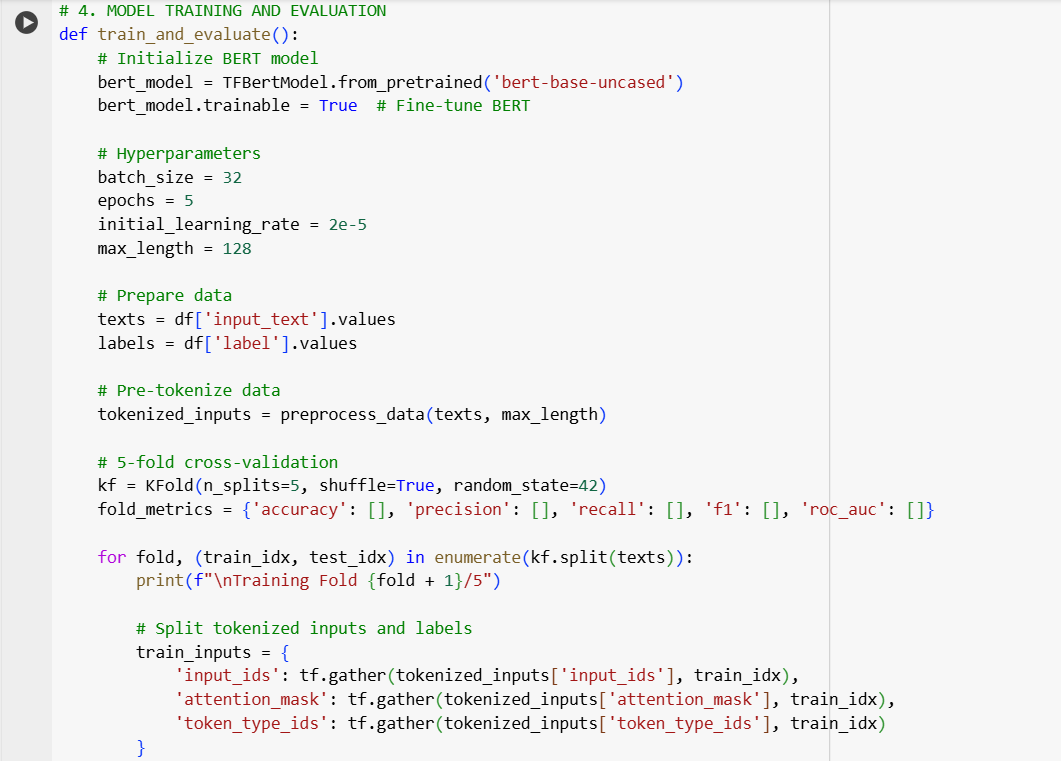
**SOURCE CODE**

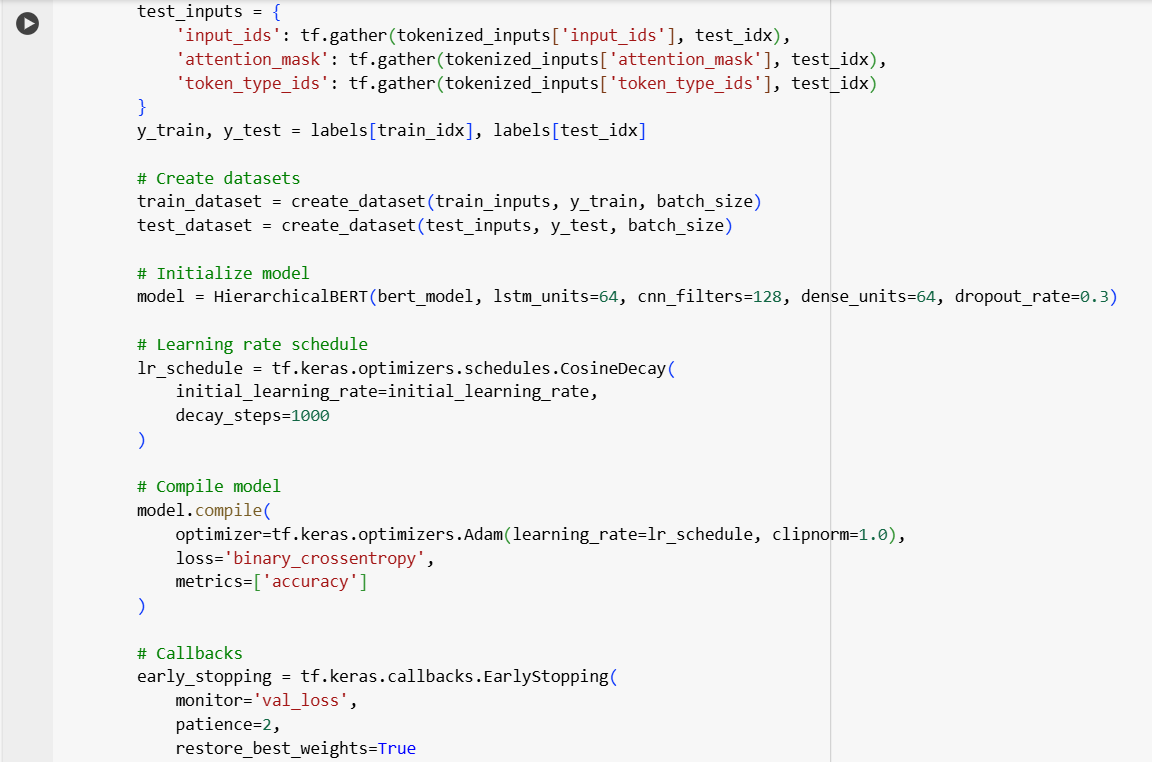




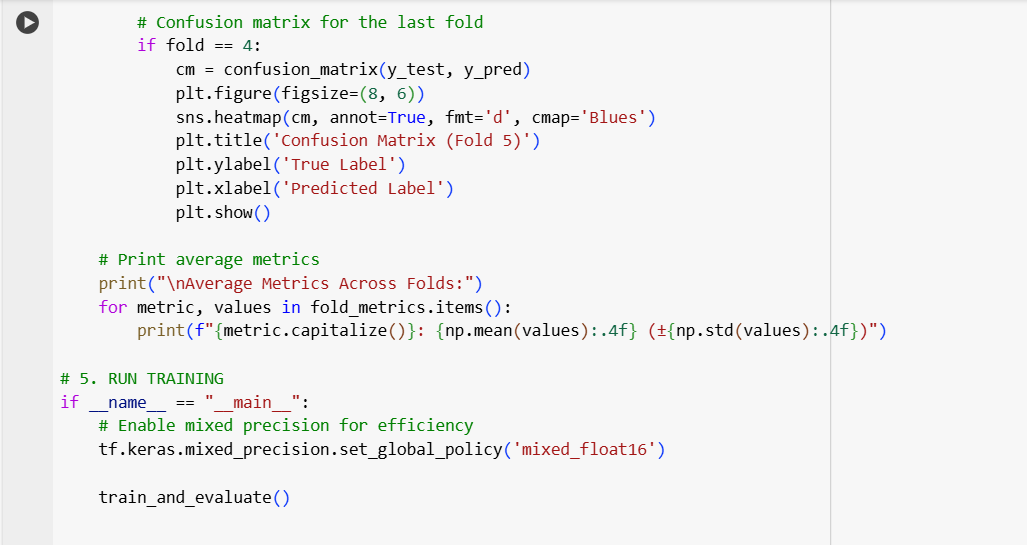




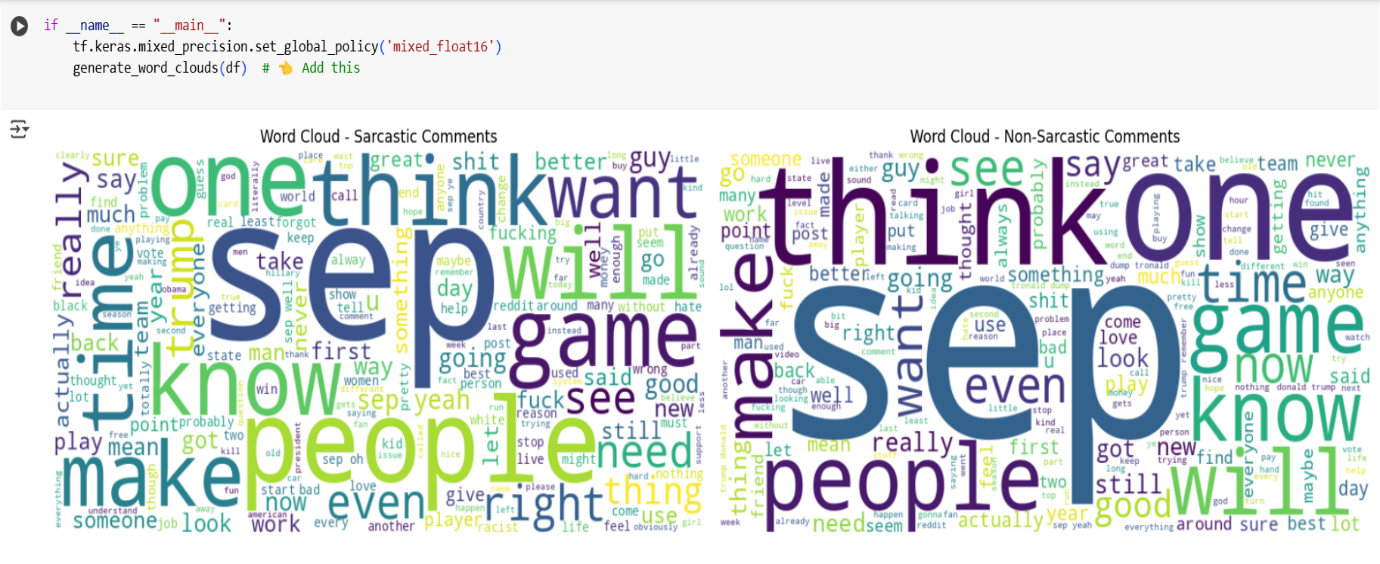




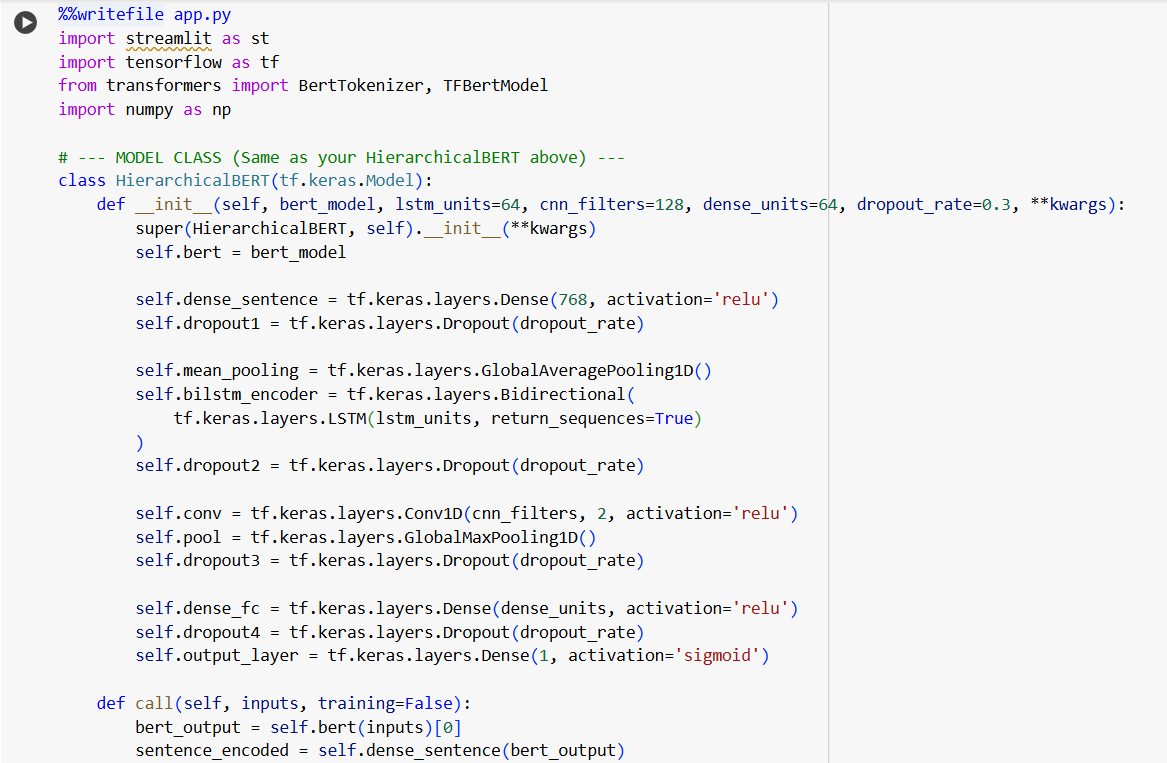


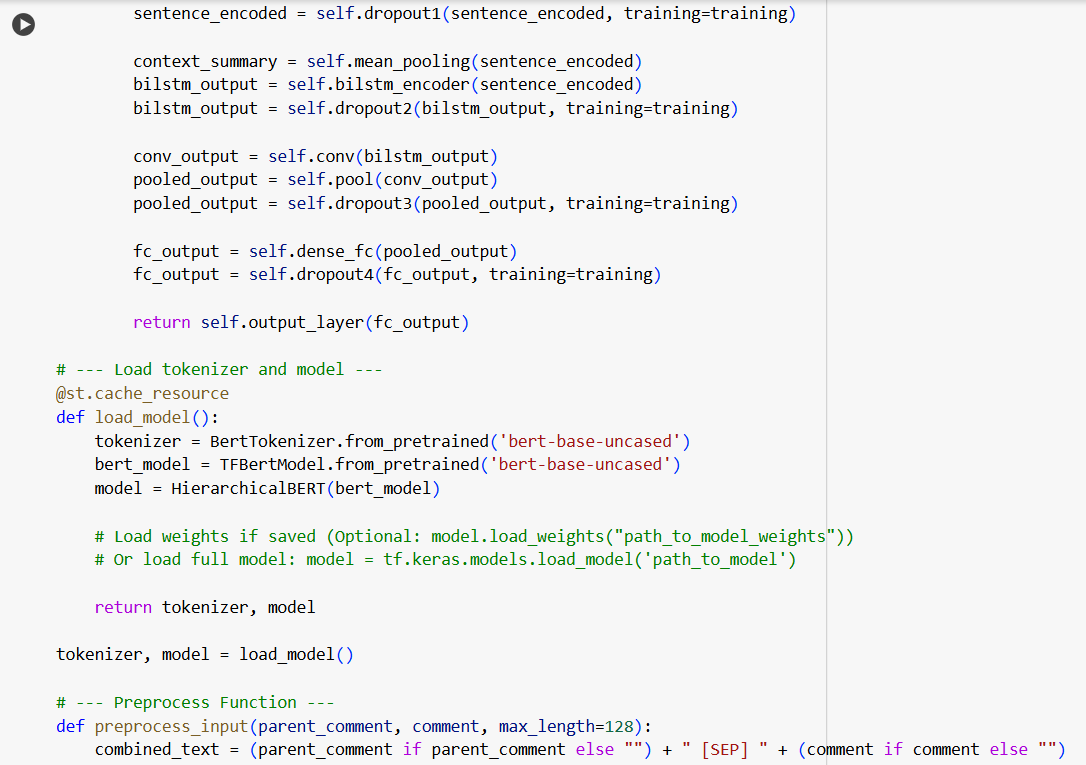






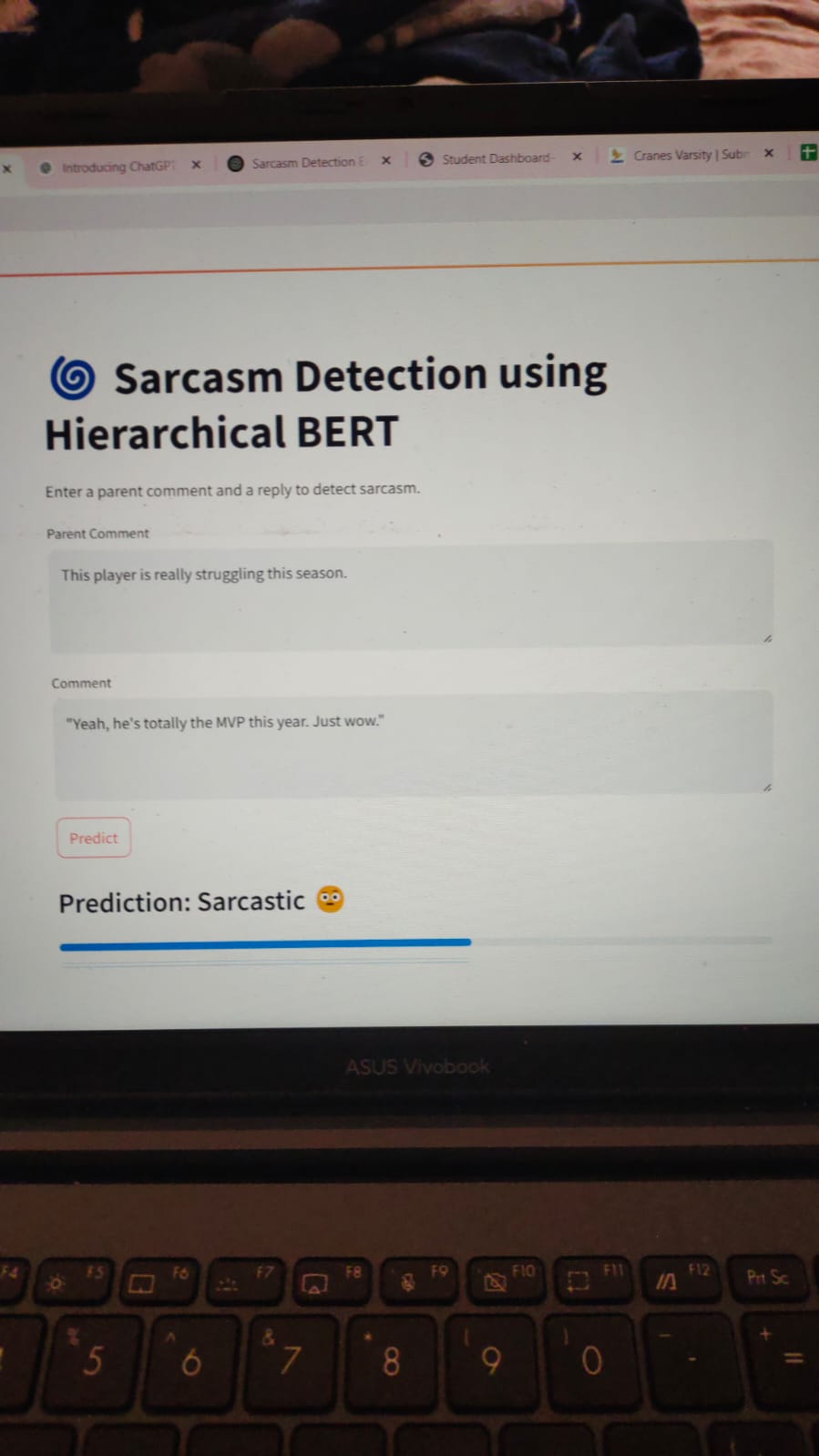
**DEPLOYMENT**

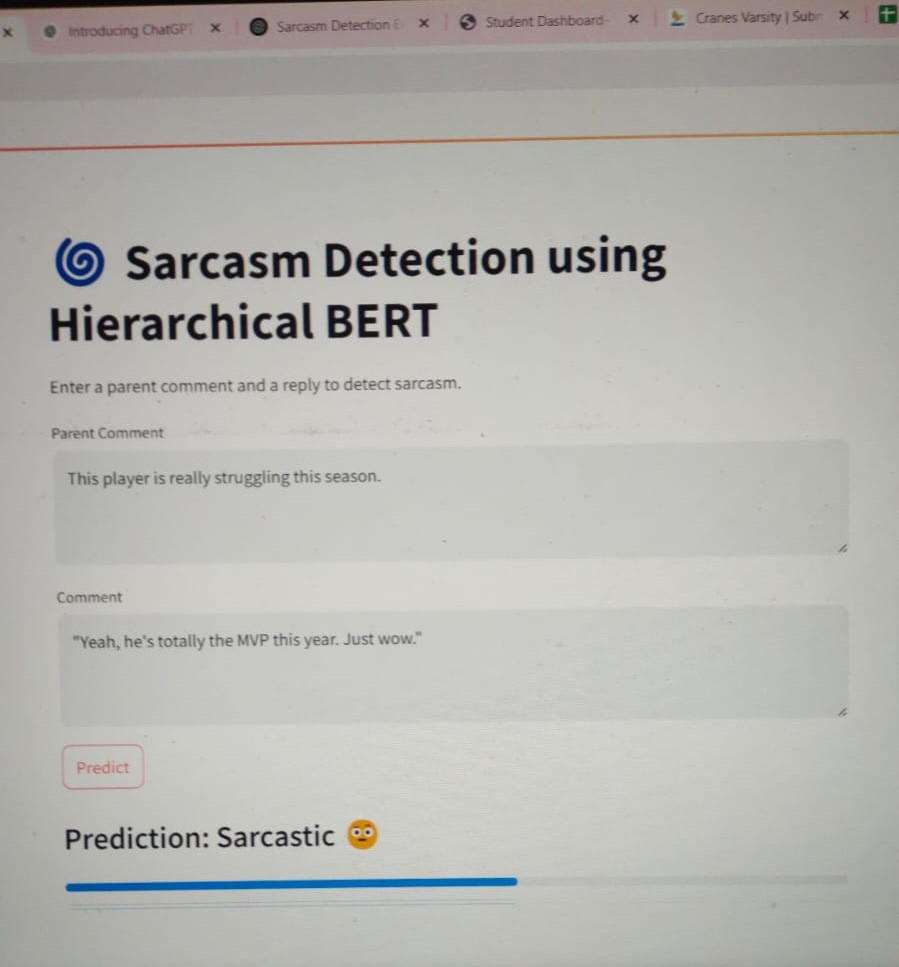






**OUTPUT**





**FUTURE WORK**

1. **Use the Full Dataset:**
   * Extend the training to the complete train-balanced-sarcasm.csv dataset (1.3M+ rows) to improve generalization and model robustness.
2. **Hierarchical Input Splitting:**
   * Implement sentence-level or paragraph-level chunking to fully utilize the hierarchical design (e.g., BERT per sentence, then LSTM/CNN over sentence representations).
3. **Multimodal Learning:**
   * Combine text with other features like **user metadata**, **post upvotes/downvotes**, or **emoji usage** to enhance sarcasm understanding.
4. **Better Context Modeling:**
   * Incorporate conversation history or thread structure to better understand contextual sarcasm.
5. **Model Ensemble:**
   * Use an ensemble of models (e.g., BERT + RoBERTa + XLNet) to boost prediction performance.
6. **Explainability & Interpretability:**
   * Apply attention visualization or LIME/SHAP to understand which words or phrases influence the sarcasm prediction most.
7. **Error Analysis & Bias Mitigation:**
   * Conduct a thorough error analysis to identify model weaknesses and address any dataset or model bias.
8. **Real-time Deployment:**
   * Optimize the model using quantization or model pruning for deployment in real-time applications like chatbots or social media moderation.
9. **Language and Domain Adaptation:**
   * Fine-tune the model on other datasets or languages to evaluate cross-domain and multilingual sarcasm detection capabilities.
10. **Incorporate Sentiment Analysis:**
    * Integrate sentiment polarity to help distinguish sarcastic positive statements from genuine ones.

**CONCLUSION**

The code implements a sarcasm detection model using a hierarchical architecture that combines a pre-trained BERT model with additional LSTM and CNN layers to capture both contextual and sequential patterns in text data. The model is trained and evaluated using 5-fold cross-validation on a balanced subset of Reddit comments.

By fine-tuning BERT embeddings and layering LSTM and CNN modules, the model aims to improve sarcasm classification performance. The use of cross-validation ensures robust evaluation, and performance metrics like accuracy, precision, recall, F1-score, and ROC-AUC are computed for each fold. The model shows promise for detecting subtle and context-dependent sarcastic expressions in online text.

REFERENCES

* <https://www.sciencedirect.com/science/article/abs/pii/S0950705121010479>
* <https://link.springer.com/chapter/10.1007/978-3-030-88942-5_18>