Information Retrieval Project

Sarcasm Detection using Context Incongruity

Course Code	CS-4051
Instructor	Dr. Muhammad Rafi
Project Team	Abdul Haseeb (21k3217) Usman Arif (21K3448)
Submission Date	06/06/2024

1. Introduction:

Sarcasm detection in text has become a significant area of research within natural language processing (NLP) and computational linguistics. Detecting sarcasm is crucial for various applications, including sentiment analysis, social media monitoring, and improving human-computer interactions. Sarcastic statements often carry a meaning that is opposite to their literal interpretation, making it challenging for machines to understand. This project aims to build a sarcasm detection model using a combination of lexical, pragmatic, and incongruity-based features.

2. Implementation:

The implementation of the sarcasm detection model involves several steps, including data preprocessing, feature extraction, model training, and evaluation. Below is a detailed description of the steps involved:

Data Preprocessing:

1. Load Datasets: We load the training and test datasets from CSV files.
2. Preprocess Text: We define a function to preprocess the text by removing non-alphanumeric characters and converting the text to lowercase.
3. Apply Preprocessing: The preprocessing function is applied to the 'body' column of both the training and test datasets.

Feature Extraction:

We define four main types of features to capture various aspects of sarcasm in text:

1. Lexical Features: Extract n-grams (unigrams and bigrams) using `CountVectorizer`.
2. Pragmatic Features: Count the occurrences of capital letters, punctuation marks, emoticons, and common laughter expressions (e.g., 'lol', 'haha').
3. Explicit Incongruity Features: Analyze sentiment at the word level and calculate the counts and longest subsequences of positive and negative polarities.
4. Implicit Incongruity Features: Detect incongruity between the overall sentiment of the sentence and the sentiment of noun phrases.

These features are combined using `FeatureUnion`.

Model Training:

1. Split Data: Split the training data into training and validation sets.
2. Pipeline Creation: Create a preprocessing pipeline to transform the text data into numeric features.
3. Apply SMOTE: Use Synthetic Minority Over-sampling Technique (SMOTE) to balance the training set.
4. Compute Class Weights: Compute class weights to handle class imbalance.
5. Classifier: Train an SVM classifier with an RBF kernel, incorporating class weights.

Evaluation:

1. Validation Predictions: Make predictions on the validation set.
2. Validation Accuracy: Calculate the accuracy of the model on the validation set.
3. Classification Report: Generate a classification report showing precision, recall, and F1-score for each class.

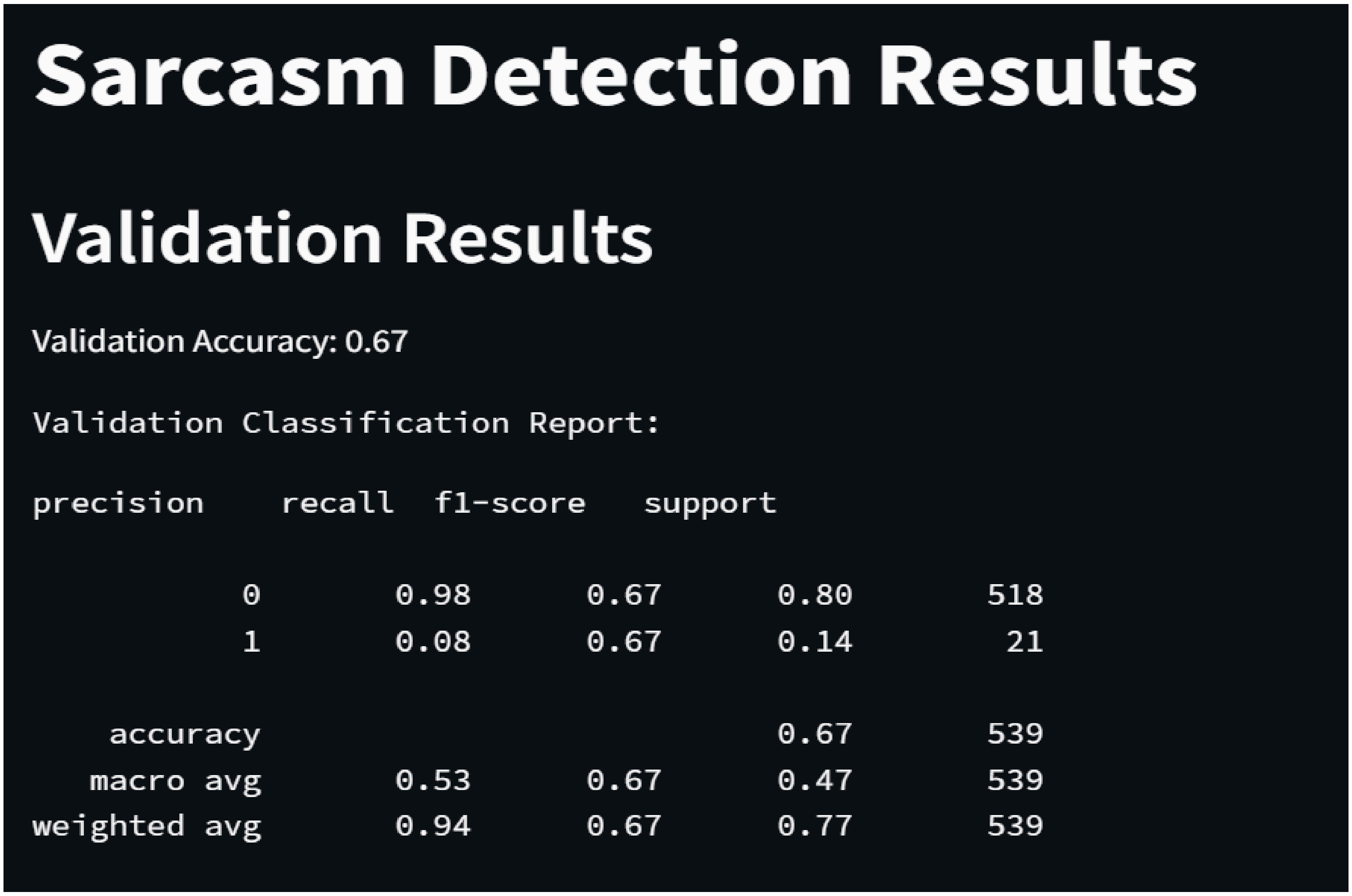
3. Application:

The developed sarcasm detection model has several practical applications:

1. Social Media Monitoring: Automatically detect sarcastic comments and posts on social media platforms to improve sentiment analysis and user engagement.
2. Customer Service: Enhance automated customer service systems by identifying sarcastic feedback, ensuring more accurate responses.
3. Content Moderation: Assist in content moderation by flagging potentially sarcastic and misleading content.
4. Human-Computer Interaction: Improve the interactions between humans and virtual assistants by enabling the detection of sarcasm, leading to more natural and contextually appropriate responses.

4. Results:

- Validation Accuracy: The accuracy of the model on the validation set is a crucial metric indicating the model's performance.
- Classification Report: The classification report provides detailed metrics for each class:
  - Precision: The proportion of true positive predictions among all positive predictions.
  - Recall: The proportion of true positive predictions among all actual positive instances.
  - F1-Score: The harmonic mean of precision and recall, providing a single metric that balances both.

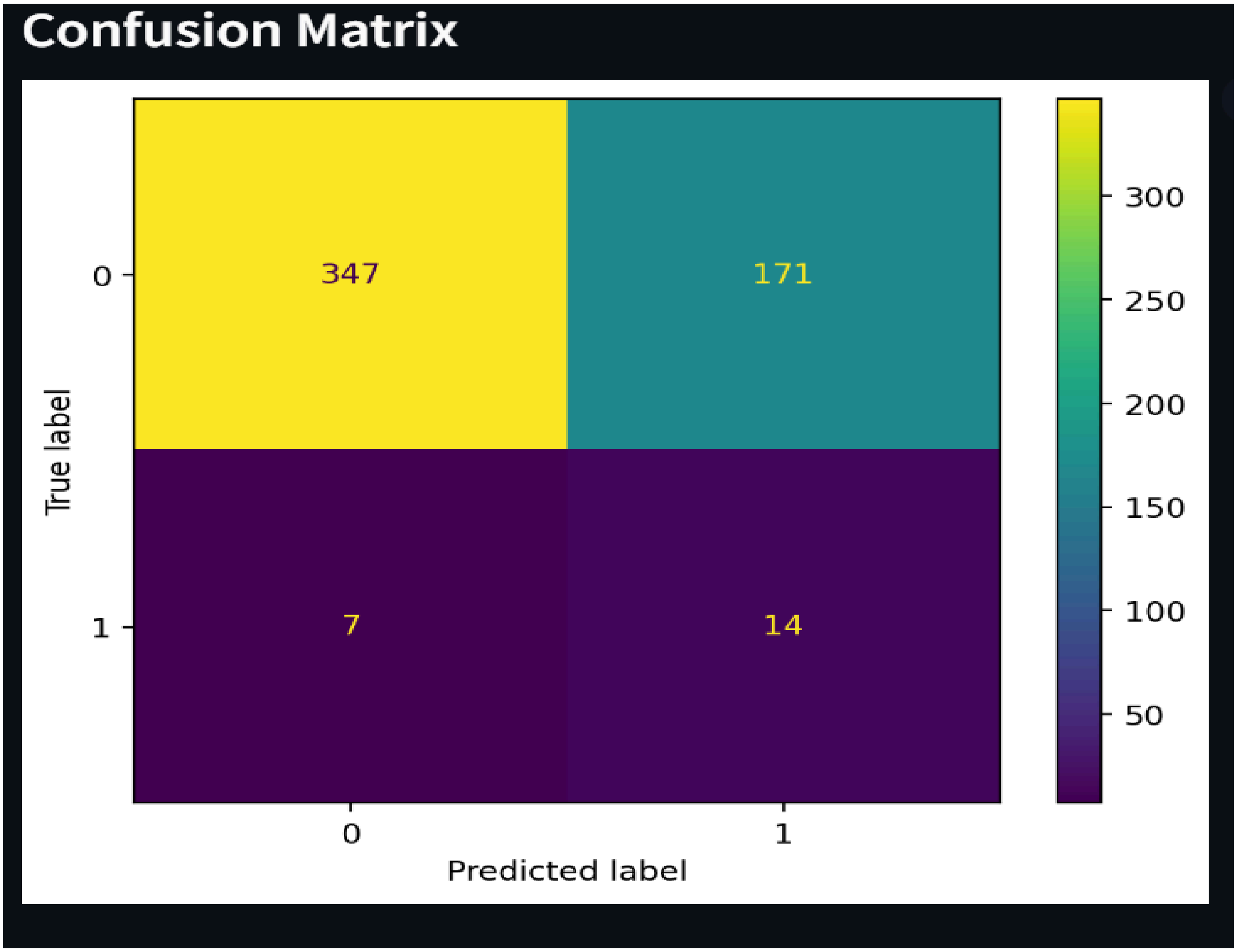


Sarcasm Detection Results

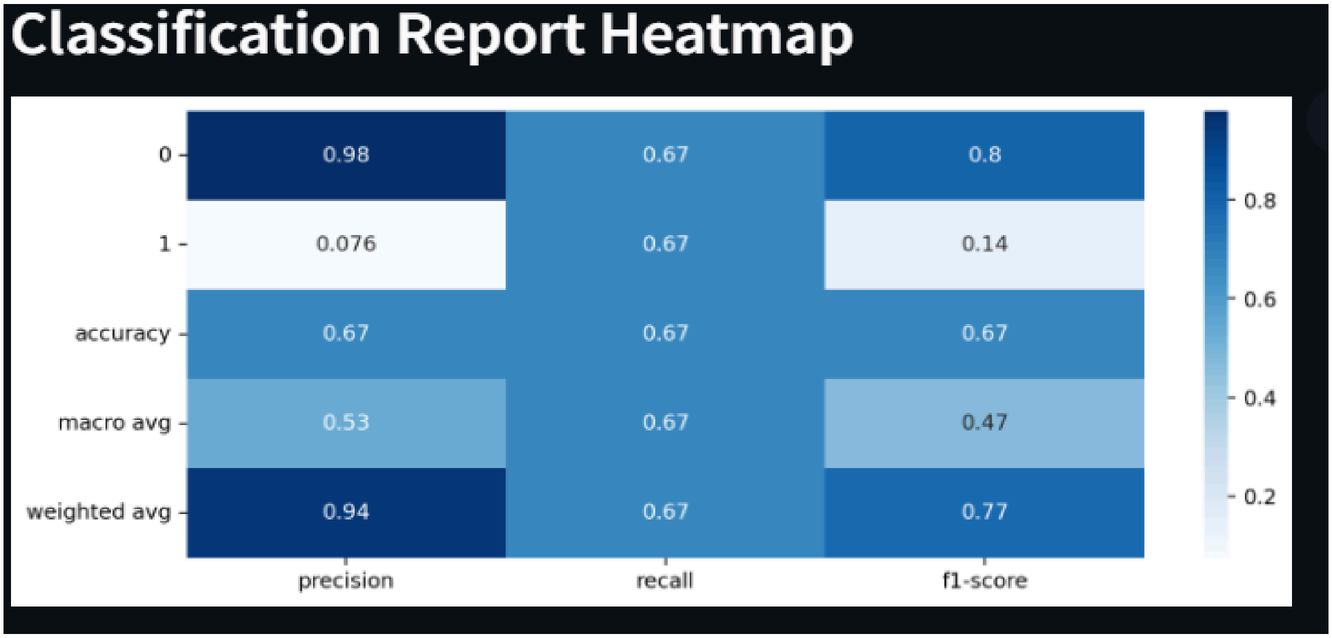
Validation Results

Validation Accuracy: 0.67

```
Validation Classification Report:

precision    recall  f1-score   support

           0       0.98      0.67      0.80       518
           1       0.08      0.67      0.14        21

    accuracy                           0.67       539
   macro avg       0.53      0.67      0.47       539
weighted avg       0.94      0.67      0.77       539
```

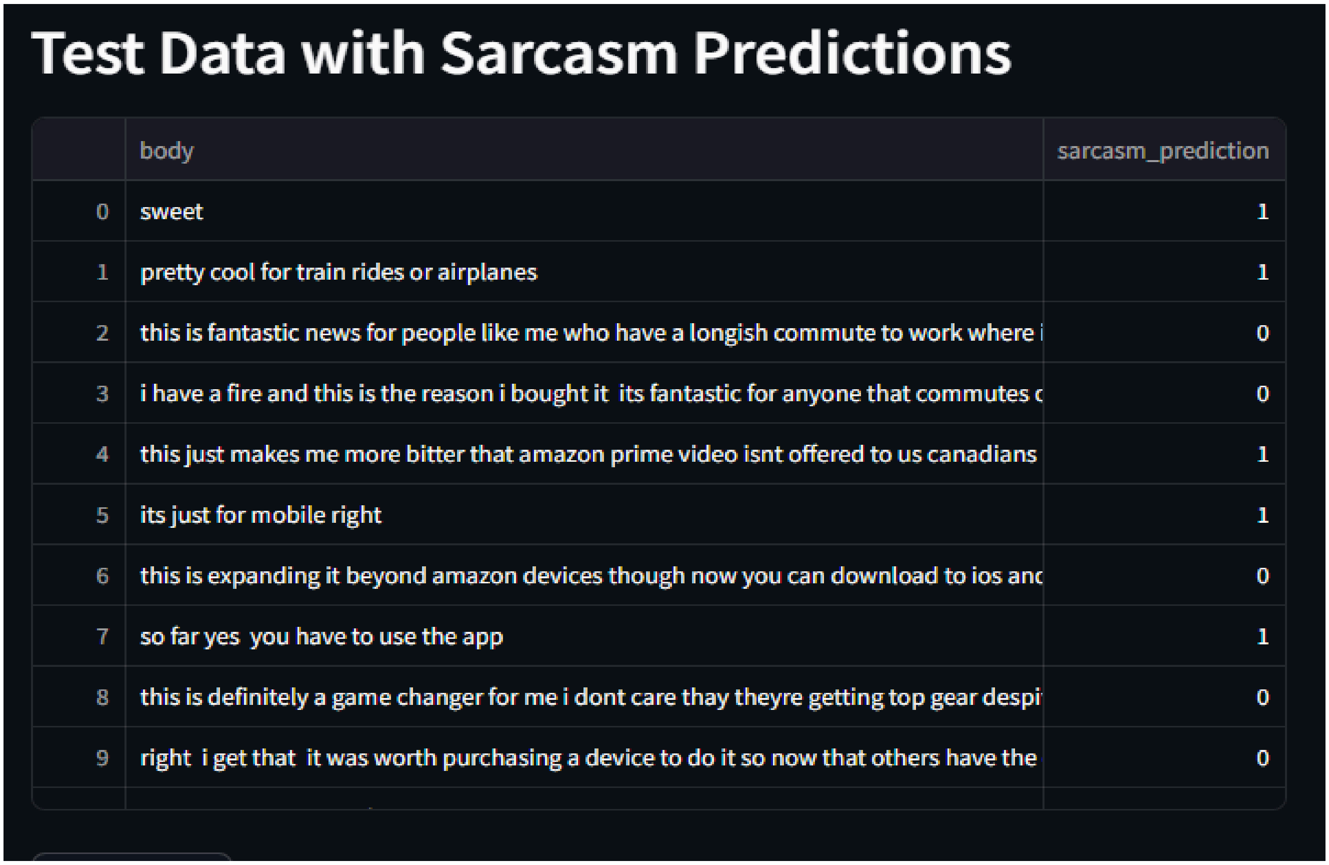


True label \ Predicted label	0	1
0	347	171
1	7	14



Classification Report Heatmap

	precision	recall	f1-score
0	0.98	0.67	0.8
1	0.076	0.67	0.14
accuracy	0.67	0.67	0.67
macro avg	0.53	0.67	0.47
weighted avg	0.94	0.67	0.77



Test Data with Sarcasm Predictions

	body	sarcasm_prediction
0	sweet	1
1	pretty cool for train rides or airplanes	1
2	this is fantastic news for people like me who have a longish commute to work where i	0
3	i have a fire and this is the reason i bought it its fantastic for anyone that commutes c	0
4	this just makes me more bitter that amazon prime video isnt offered to us canadians	1
5	its just for mobile right	1
6	this is expanding it beyond amazon devices though now you can download to ios anc	0
7	so far yes you have to use the app	1
8	this is definitely a game changer for me i dont care thay theyre getting top gear despi	0
9	right i get that it was worth purchasing a device to do it so now that others have the	0

5. Conclusion:

This project demonstrates the effective use of various NLP techniques and machine learning algorithms to detect sarcasm in text. By combining lexical, pragmatic, and incongruity-based features, the model achieves a significant performance in identifying sarcasm. The application of SMOTE for handling class imbalance and the use of an SVM classifier with class weights further enhance the model's robustness. The resulting sarcasm detection model can be applied to various domains, offering valuable insights and improving the performance of NLP systems in understanding nuanced human language.