IndiaAI CyberGuard AI Hackathon - Project Report

- Harsh Pare

Declaration of Originality

I hereby declare that this project work is my original work and no part of it has been submitted elsewhere for any other purpose. All external sources have been properly cited and acknowledged.

- Harsh Pare

Introduction

The IndiaAI CyberGuard AI Hackathon presents a challenging text classification task involving unstructured data related to cyber crime reporting. The dataset consists of 93686 rows of text messages submitted by people across India, describing various types of cyber crimes they have experienced or witnessed.

The primary objective of this project is to develop a robust natural language processing (NLP) model that can accurately categorize these text-based cyber crime reports into a predefined set of 15 high-level categories and 36 subcategories. Accurately classifying these complaints is crucial for the organisation to better understand the nature and trends of cyber crimes.

This report contains details about the dataset provided for the hackathon, the challenges in the dataset, the models I used to perform the classification task, the approach I used finally and the results I obtained. All the code that I wrote for this classification task is available in the Github repository. There are 4 Jupyter notebooks that I used for this project.

Problem Statement

The objective of this task is to develop Natural Language Processing (NLP) models for a text classification task. The training dataset provided for this task contains 93686 entries. However, there are several repeated entries in the training dataset. After removing the duplicate entries, the training dataset has 85883 data points. For example, there is a specific message in the category "RapeGang Rape RGRSexually Abusive Content" which is repeated around 2000 times.

Dataset Overview

The dataset contains three columns: category, sub_category, and crimeaditionalinfo. The column crimeaditionalinfo contains messages sent by people reporting instances of cyber crimes that they have witnessed. These messages have been classified into categories and subcategories.

Category and Subcategory Overview:

The category column captures broad categories of cybercrimes. Each category may have either 0, 1 or multiple associated sub_category to specify the type of offense further. This subcategorization provides a finer-grained classification as compared to the category. However, there are some subcategories which seem to have overlapping meanings, and to a human, it might feel that some crimes can belong to multiple subcategories.

Messages Overview (crimeaditionalinfo):

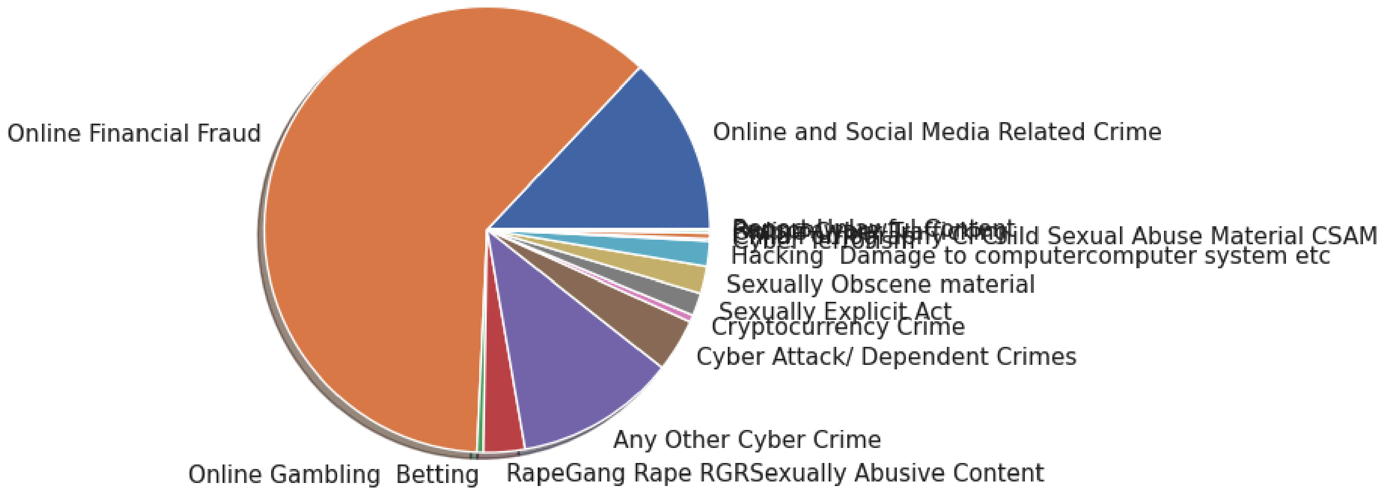
The crimeaditionalinfo column consists of complaint messages submitted by people of India, and therefore it contains varied text quality. Some messages are written in proper English whereas some messages have some irregularities like spelling mistakes, grammatical issues, mixed languages like Hindi, Bengali etc, informal formatting, punctuation errors, etc.

The dataset contains some rows which have NaN values, and there are some duplicate rows as well:

Number of data points with missing values = 6612

Number of duplicate data points = 7803

The following flow chart shows the distribution among categories in the dataset:



Category and Subcategory Distribution Summary

Categories:

As can be seen in the above flowchart, the dataset is dominated by the category Online Financial Fraud, with 57,434 messages. This is followed by Online and Social Media Related Crime (12,140) and Any Other Cyber Crime (10,878).

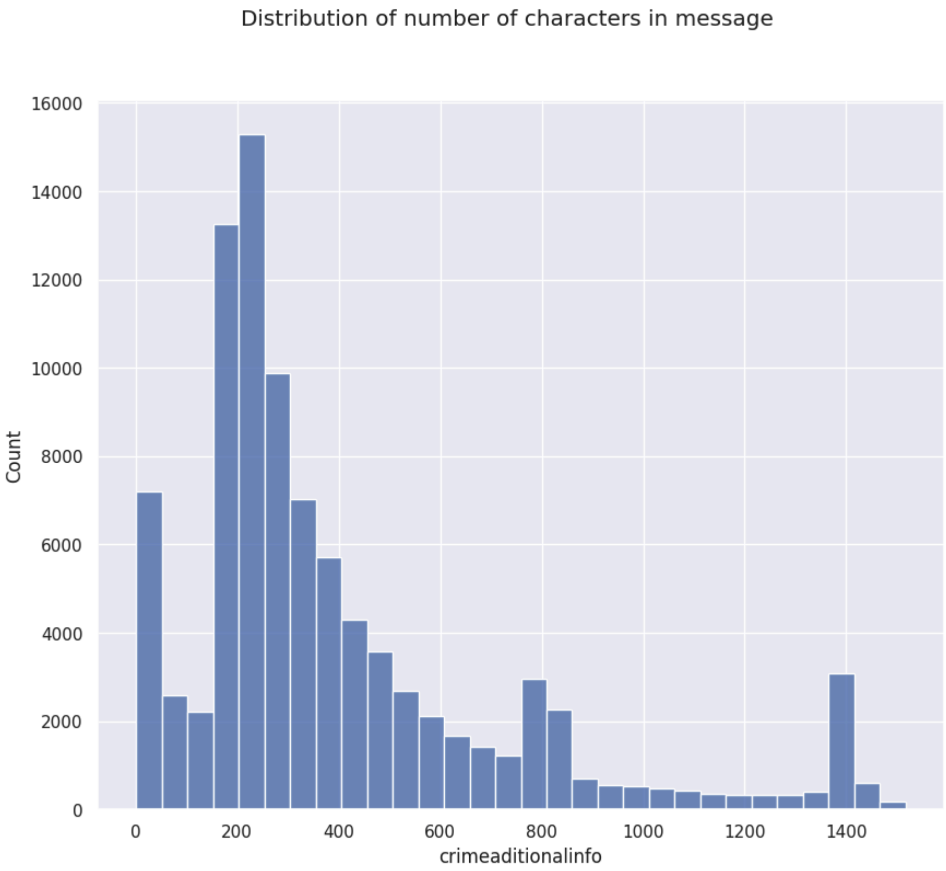
Less common categories include Cryptocurrency Crime (480), Online Gambling/Betting (444), and the rarely reported categories like Report Unlawful Content (1) and Ransomware (56).

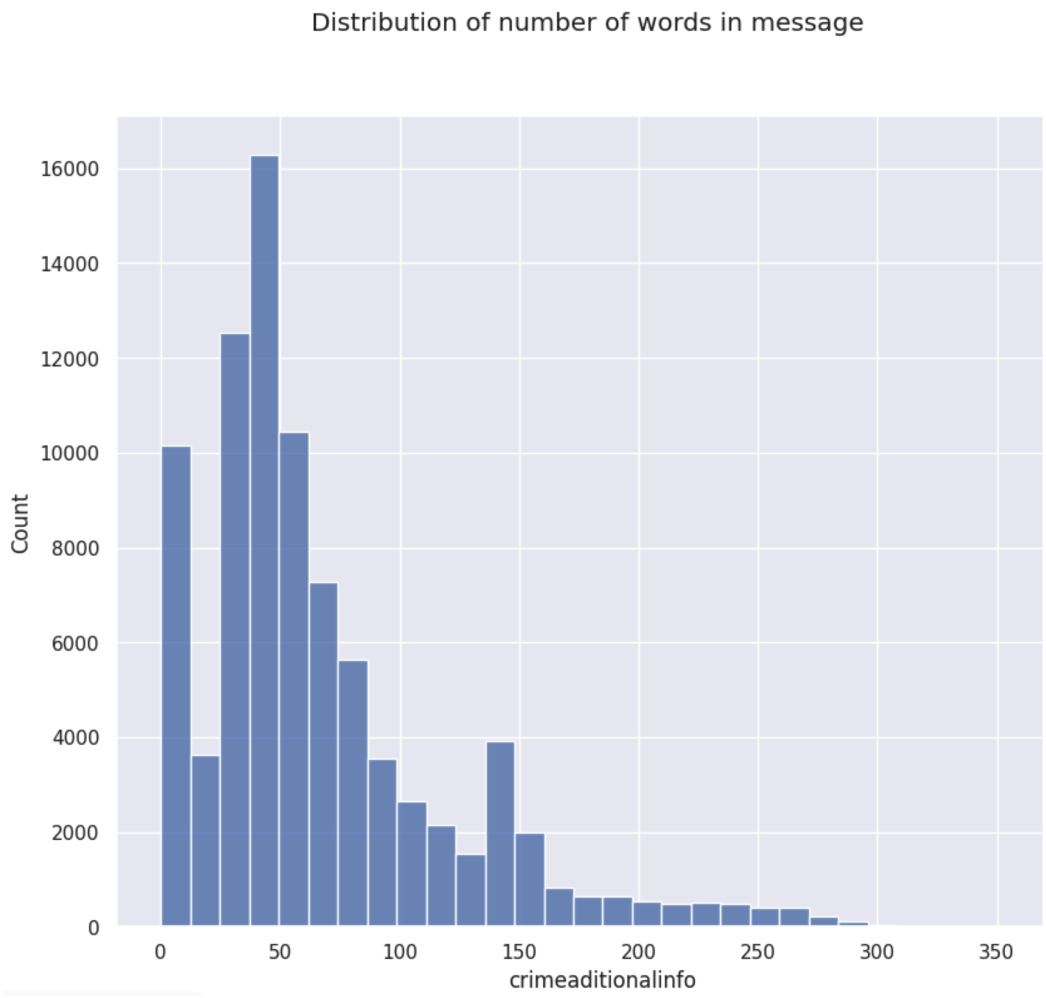
Subcategories:

The most frequent subcategory is UPI Related Frauds with 26,856 messages, which indicates a high number of financial fraud reports related to UPI.

Subcategories like Other, Debit/Credit Card Fraud/Sim Swap Fraud, and Internet Banking Related Fraud are also highly reported, which highlights the fact that financial scams are a significant concern.

There is a mix of prevalent and rare subcategories. Some subcatefories have been rarely reported, like Intimidating Email (29) and Against Interest of sovereignty or integrity of India (1).





The above two histograms show the distribution of the number of characters and number of words in the messages in the dataset. We can see that most of the messages lie in the range of around 50 words.

Message Quality Observations

Empty and Duplicate Messages:

There are 21 empty messages, and for the classification, I filtered these out.
8,672 messages are duplicates, which indicates redundancy in the message entries. The duplicate messages were removed to avoid skewness in the classification task.

Quality Issues in Messages:

Sampled messages reveal typical issues: informal language, spelling errors, mixed language usage (Hindi, English, etc), and inconsistent formatting. Examples of quality issues include incomplete words, mixed punctuation, and irregular line breaks, which shows that the dataset has been generated manually by crowdsourcing.

Categories and Subcategories

There are total 15 categories in the dataset. These categories are further divided into 36 subcategories. There are some categories which do not have further divisions into subcategories. Following is the list of categories and the corresponding subcategories inside them.

1. Sexually Obscene material
2. RapeGang Rape RGRSexually Abusive Content
3. Any Other Cyber Crime
   - 3.1. Other
4. Sexually Explicit Act
5. Child Pornography CPChild Sexual Abuse Material CSAM
6. Online Cyber Trafficking:
   - 6.1. Online Trafficking
7. Report Unlawful Content
   - 7.1. Against Interest of sovereignty or integrity of India
8. Ransomware
   - 8.1. Ransomware
9. Cryptocurrency Crime
   - 9.1. Cryptocurrency Fraud
10. Hacking Damage to computercomputer system etc
    - 10.1. Damage to computer computer systems etc
    - 10.2. Email Hacking
    - 10.3. Tampering with computer source documents
    - 10.4. Unauthorised AccessData Breach
    - 10.5. Website DefacementHacking
11. Cyber Attack/ Dependent Crimes

11.1. Data Breach/Theft

11.2. Denial of Service (DoS)/Distributed Denial of Service (DDOS) attacks

11.3. Hacking/Defacement

11.4. Malware Attack

11.5. Ransomware Attack

11.6. SQL Injection

11.7. Tampering with computer source documents

12. Cyber Terrorism

12.1. Cyber Terrorism

13. Online and Social Media Related Crime

13.1. Cheating by Impersonation

13.2. Cyber Bullying Stalking Sexting

13.3. EMail Phishing

13.4. FakeImpersonating Profile

13.5. Impersonating Email

13.6. Intimidating Email

13.7. Online Job Fraud

13.8. Online Matrimonial Fraud

13.9. Profile Hacking Identity Theft

13.10. Provocative Speech for unlawful acts

14. Online Financial Fraud

14.1. Business Email CompromiseEmail Takeover

14.2. DebitCredit Card FraudSim Swap Fraud

14.3. DematDepository Fraud

14.4. EWallet Related Fraud

14.5. Fraud CallVishing

14.6. Internet Banking Related Fraud

14.7. UPI Related Frauds

15. Online Gambling Betting

15.1. Online Gambling Betting

Following is an analysis of all the 15 categories and how they are different from each other.

Sexually Obscene Material

Meaning: Complaints involving obscene material shared online which is sexually explicity content.
Common Messages: Reports of inappropriate images, messages, or media shared online.

Rape Gang Rape RGR Sexually Abusive Content

Meaning: Involves content that references or depicts rape or other sexually abusive behavior.
Common Messages: Messages containing references to rape or abusive content that is sexual in nature.

Differentiation: This category focuses specifically on content linked to sexual violence, while "Sexually Obscene Material" covers a broader range of inappropriate sexual content.

Any Other Cyber Crime

Meaning: A broad category for any cybercrime that doesn't fit into other predefined types.
Common Messages: Miscellaneous complaints about unusual or less common cyber offenses.

Subcategories:
Other: A broad subcategory that contains messages that do not belong to any other specific subcategory.

Sexually Explicit Act

Meaning: Explicit sexual activities reported in online media or messages.
Common Messages: Complaints about sexually explicit videos, images, or text content.
Differentiation: This category sounds very similar to the category "Sexually Obscene Material", but there might be some nuances. The messages in both these categories look very similar to

each other. It looks like the messages belonging to the category “Sexually Explicit Act” are more severe in comparison to “Sexually Obscene Material”.

Child Pornography CP Child Sexual Abuse Material CSAM

Meaning: Involves illegal content related to minors.
Common Messages: Reports of underage content, explicit images or videos involving minors.

Differentiation: This category strictly involves minors, and is therefore different from general sexually explicit material or obscene content.

Online Cyber Trafficking

Meaning: Cyber-enabled trafficking activities, often involving human trafficking or illegal trade.
Common Messages: Reports of exploitation or illegal trading of individuals online.

Subcategories:
Online Trafficking: Complaints where individuals are trafficked or exploited online.

Report Unlawful Content

Meaning: Content that poses a threat to national security or public order.

Common Messages: Messages targeting the integrity or security of India, such as calls for violence or hate speech against an individual or community.

Subcategories:
Against Interest of Sovereignty or Integrity of India: Specific to threats against national security.

Ransomware

Meaning: Ransomware incidents where users' files or systems are locked until a ransom is paid.
Common Messages: Reports of data or system lockouts with ransom demands.

Subcategories:
Ransomware: Specific to incidents involving ransomware or system control.

Cryptocurrency Crime

Meaning: Crimes related to cryptocurrency, like scams or unauthorized transactions.
Common Messages: Complaints about fraudulent crypto transactions or unauthorized access to crypto wallets.

Subcategories:
Cryptocurrency Fraud: Scams involving cryptocurrencies.

Hacking / Damage to Computer Systems

Meaning: Complaints about unauthorized access or damage to computing systems.
Common Messages: Reports of system breaches or data theft.

Subcategories:
Damage to Computer Systems: Incidents causing system or data loss.
Email Hacking: Unauthorized email account access.
Tampering with Source Documents: Altering source codes or software.
Unauthorized Access/Data Breach: Cases of data theft or exposure.
Website Defacement/Hacking: Defacement or hacking of websites.

Differentiation: Subcategories differ by the nature of the breach - damage to computer system, email-specific, data exposure, or web defacement.

Cyber Attack / Dependent Crimes

Meaning: A broad category for cyber attacks affecting infrastructure or services.

Common Messages: Reports on DDoS, SQL injections, malware, and other attacks.

Subcategories:

Data Breach/Theft: Theft or exposure of private data.

DoS/DDoS Attacks: Attacks making services unavailable.

Malware Attack: Reports of harmful software.

Ransomware Attack: Ransom-related attacks within larger attack patterns.

SQL Injection: Attacks targeting databases.

Differentiation: Covers a variety of attacks; ransomware here might be part of a larger attack strategy.

Cyber Terrorism

Meaning: Cyber activities intending to cause fear or instability, often linked to extremist threats.

Common Messages: Threats that could cause harm or induce fear at a large scale.

Online and Social Media Related Crime

Meaning: Crimes conducted via social media or online interactions.

Common Messages: Includes impersonation, bullying, and scams on social media.

Subcategories:

Cyber Bullying / Stalking / Sexting: Harassment online.

Fake/Impersonating Profile: Fake accounts.

Online Job Fraud: Job scams on social media.

Online Matrimonial Fraud: Scams on matrimonial sites.

Differentiation: Distinguishes between social media impersonation, direct threats (cyberbullying), and fraud targeting personal profiles or relationships.

Online Financial Fraud

Meaning: Encompasses financial fraud via online systems, especially banking.

Common Messages: Complaints about unauthorized transactions and fake calls.

Subcategories:

UPI Related Frauds: Scams through UPI payment systems.

Debit/Credit Card Fraud: Issues with unauthorized card use.

E-Wallet Related Fraud: Scams involving digital wallets.

Fraud Call/Vishing: Phishing scams via calls.

Differentiation: Differentiated by the type of financial transaction platform.

Online Gambling / Betting

Meaning: Issues related to online gambling or betting.

Common Messages: Complaints of scams or losses due to online betting.

Subcategories:

Online Gambling/Betting: Gambling-related activities on digital platforms.

Data Preprocessing

The data provided for the classification task contains messages sent by various people, therefore there are a lot of problems with the data quality. For this reason, the data needs to be preprocessed so that it can be used by NLP models. I created a data processing pipeline to preprocess the data and remove all the irregularities from the text data. The following section describes the data processing pipeline.

Data Preprocessing Pipeline

The data preprocessing pipeline that I created made use of the following tools and methods to clear the data.

1. Convert to Lowercase: Since there is no significance of case-sensitivity, therefore we convert the text to lowercase, so that the models do not differentiate between upper case and lower case letters.
2. Remove Whitespaces: We remove the unnecessary empty spaces from the text messages.
3. Remove the punctuations: Punctuations do not play any role in classifying text. Therefore we remove punctuations so that they do not affect the NLP models.
4. Remove Unicode characters: The messages might contain emojis, URLs, punctuation and other symbols that do not contribute semantically to the text messages. So we remove these irrelevant characters from the messages using regular expressios.
5. Substitute acronyms: Acronyms are shortened forms of phrases, generally found in informal messages. E.g fyi, btw. We replace these acronyms by the full text.
6. Substitute contractions: Contractions are a shortened form of a word or a phrase, obtained by dropping one or more letters. We replace these contractions with the full word or phrase. E.g. - won’t -> would not, I’m -> I am.
7. Remove Stopwords: There are a lot of words, for example pronouns, prepositions etc, which do not have much effect on the classification process. These are called stop words. We get rid of these unwanted stop words.
8. Stemming and Lemmatization: Stemming is the process of reducing the words to their root form or stem. It reduces related words to the same stem even if the stem is not a

dictionary word. For example, the words introducing, introduced, introduction reduce to a common word introduce. Lemmatization offers a more sophisticated approach by utilizing a corpus to match root forms of the words. Unlike stemming, it uses the context in which a word is being used.

9. Remove non-alphabetic words: Non-alphabetic words do not add anything semantically, therefore we remove them to remove noise.

Methodology and Implementation:

This classification task was approached in two phases: first, by establishing a baseline with traditional machine learning models, and then by leveraging more advanced techniques with BERT to improve performance and using some other methods to address class imbalance.

Data Preprocessing and Cleaning

Text Cleaning: The crimeaditionalinfo column, containing user-generated complaint messages, required extensive cleaning to handle spelling mistakes, grammar errors, mixed languages (Hindi and English), and inconsistent formatting. I used the data processing pipeline explained in the previous section to take care of the preprocessing steps.

Tokenization: For the tokenization, I used the Regexp tokenizer from NLTK for the initial baseline models, whereas for the BERT models I used the BERT tokenizer.

TFIDF Vectorization: Cleaned text data was then transformed using Term Frequency-Inverse Document Frequency (TFIDF) vectorization. This step converted the text into numeric features, capturing the relevance of words within each category, essential for baseline model performance.

BERT Embeddings: For the second phase of the project, I used BERT. BERT, which stands for Bidirectional Encoder Representations from Transformers, is a groundbreaking model in the field of natural language processing (NLP) and deep learning. It was introduced by researchers at

Google in 2018 and has since become one of the most influential and widely used models in NLP.

BERT is a type of transformer-based neural network architecture that learns contextualized word representations by leveraging the bidirectional nature of language. Unlike previous models that only consider the surrounding words in a unidirectional manner, BERT can capture the context from both the left and right sides of a given word. This bidirectional approach allows BERT to better understand the nuances and dependencies within a sentence or a paragraph.

The core idea behind BERT is pre-training and fine-tuning. In the pre-training phase, BERT is trained on a massive amount of unlabeled text data, such as books, articles, and web pages. During this phase, the model learns to predict missing words in a sentence by considering the surrounding words. This process enables BERT to acquire a deep understanding of the language's syntactic and semantic structures. For this project, I used the pre-trained BERT representations and fine tuned it on the classification task.

Baseline Model Selection

Model Selection: A diverse set of traditional machine learning models was used to establish a baseline. These models included:

- Logistic Regression
- K-Nearest Neighbors (KNN) Classifier
- Decision Tree
- Linear Support Vector Machine (SVM)
- Random Forest
- Stochastic Gradient Descent (SGD) Classifier
- Ridge Classifier
- XGBoost
- AdaBoost

Performance Evaluation: Each model was evaluated on the TFIDF-transformed data. Linear SVM emerged as the best-performing model, establishing a solid initial benchmark for classification performance.

Hyperparameter Tuning

Given that Linear SVM achieved the best results among the traditional models, hyperparameter tuning was performed to refine its performance further. Following hyperparameters were adjusted:
C (Regularization Parameter): Balancing training error minimization with overfitting prevention.
Kernel Selection: Optimizing the SVM's decision boundaries by exploring different kernel functions. I used the linear and rbf kernels for this task.
These adjustments improved the model's predictive power, particularly on the more nuanced categories.

Class Imbalance Handling

Class Imbalance Problem: The dataset had significant class imbalance (for both categories and subcategories), with some categories being highly underrepresented. Initially, accuracy was used as the evaluation metric, but it wasn't the best evaluation metric due to this imbalance. Therefore, alternative metrics like F1 score and ROC-AUC were introduced for a more comprehensive assessment.

Strategies to Address Imbalance:

- **SMOTE and Undersampling Pipeline:** A pipeline combining SMOTE (Synthetic Minority Oversampling Technique) and undersampling was implemented. SMOTE oversampled the minority classes, while undersampling balanced the overrepresented classes, creating a more balanced distribution for training. This is a popular technique which is used for tackling the class imbalance problem.

- **Synthetically Generated Messages:** Leveraging the OpenAI LLM API, synthetic messages were generated for the underrepresented classes. Sample messages were provided to the LLM to produce similar messages that used mixed Hindi and English words, ensuring alignment with the original dataset. The code for generating these synthetic messages is in the notebook. Included synthetically generated messages helped enrich the dataset by adding more examples for the minority classes.
- **Alternative Loss Function (Focal Loss):** Focal loss was experimented with as a substitute for the traditional loss function to prioritize learning on harder-to-classify, underrepresented classes. However, results did not meet expectations, and this approach was not used in the final model. However, I have kept the code for focal loss in the notebook for demonstration purpose.

Hierarchical Classification Approach

This problem was unique because of the inclusion of subcategories in the text classification task. We had to first classify the messages into categories and then further into subcategories. However, in this dataset, the subcategories were not equally distributed among the categories. For example, some categories did not have any subcategories, some categories had only one subcategory within them, therefore these type of categories did not require further classification into subcategories. However, the messages belonging to the 4 categories required further classification into subcategories. For this, I used a hierarchical classification approach, as there were total 36 subcategories. Using text classification directly for 36 classes would have given poor results, especially because of the class imbalance issue. Using hierarchical classification reduces the chances of errors, since at a time, we are focusing on only 1 category. Following were the categories for which I used hierarchical clustering for classifying the subcategories:

- Online and Social Media Related Crime
- Online Financial Fraud
- Cyber Attack/ Dependent Crimes
- Hacking Damage to computercomputer system etc

Advanced Model Selection: BERT for Text Classification

I chose BERT as the advanced model for text classification. The reason for choosing BERT as the model was that it is an advanced NLP model which is pre-trained on a vast corpus, making it highly effective for tasks which involve understanding the nuances language, which was required in this classification task. Unlike traditional models, BERT captures contextual dependencies within the text, enabling it to handle mixed-language inputs and minor variations more effectively. This deep, contextualized understanding significantly enhances performance in complex text classification problems, especially those with informal or user-generated data.
Implementation and Fine-Tuning: The BERT model was fine-tuned on the dataset, allowing it to learn category-specific language patterns.

Evaluation Metrics

Evaluation Adjustments: Due to the initial class imbalance, accuracy was not a reliable metric. Therefore, the final evaluation relied on:
F1 Score: Balancing precision and recall, making it well-suited to measure model performance across all classes, including underrepresented ones.
ROC-AUC: Assessed the model's capability to distinguish between classes, especially in scenarios with severe class imbalance.

Results and Analysis

Model Performance

Baseline Models:

The following table summarises the performance of the baseline models:

Classifier	**Training F1 Score**	**Validation F1 score**
Linear SVM	0.8299691123875582	0.780175566934894
XGBoost	0.8352118781835917	0.7784686661789807
Logistic Regression	0.8193887504064159	0.7782248232138502
Ridge Classifier	0.8622114446732416	0.7706656912948061
SGD Classifier	0.7743714099924136	0.7655449890270666
Random Forest	0.993578627939742	0.7639600097537186
Decision Tree	0.993578627939742	0.6992197025115825
KNN Classifier	0.6923702178389509	0.6872713972201902
AdaBoost	0.6466890647014197	0.6458180931480126

Among all the baseline models that were tested, SVM performed the best on the validation set. Therefore I explored SVM model and performed hyperparamter tuning for SVM. I tried 2 kernels for this task: linear and rbf.

SVM Hyperparameter Tuning:

I tried various values of the hyperparameter C and tried two kernels, namely linear and Rbf. The best validation F1 score was obtained on using the parameters: C=1, kernel=”rbf”. Following were the results obtained on using the best model on the test set:

F-1 Score = 0.7429724691016926

BERT Model:

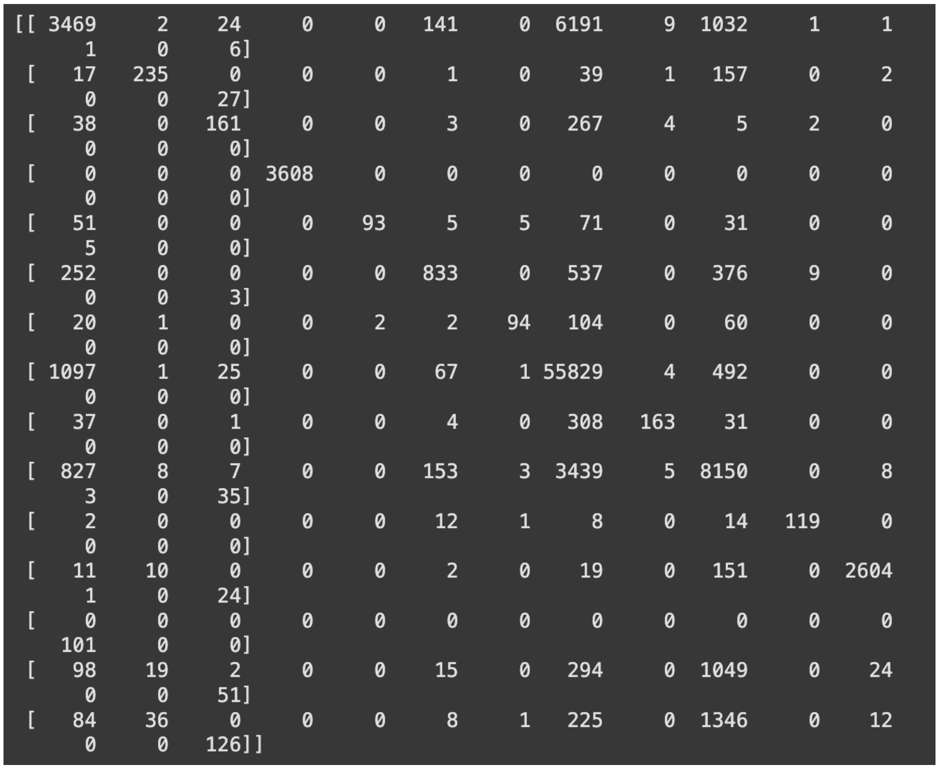
I used the BERT model for classifying both the category as well as the subcategory. This was done in a hierarchical manner, wherein first I classified the messages into categories and then for the 4 categories which were further divided into subcategories, I finetuned 4 separate models. The models were too big, so I wasn't able to upload them on the Github repo.

Following results were obtained using the BERT model on teh test set:

Classification into Category:

F-1 Score = 0.7942520884779068

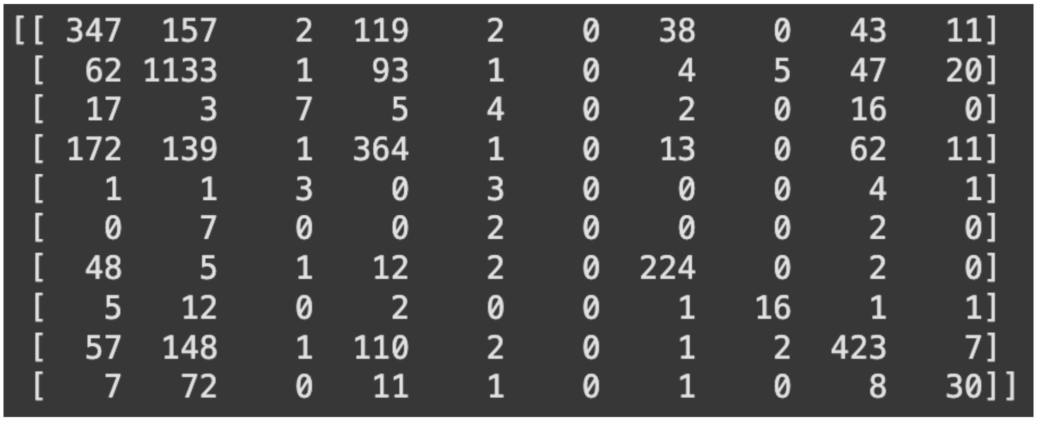
Confusion Matrix:



```
[[ 3469     2    24     0     0   141     0  6191     9  1032     1     1
      1     0     6]
 [   17   235     0     0     0     1     0    39     1   157     0     2
      0     0    27]
 [   38     0   161     0     0     3     0   267     4     5     2     0
      0     0     0]
 [    0     0     0  3608     0     0     0     0     0     0     0     0
      0     0     0]
 [   51     0     0     0    93     5     5    71     0    31     0     0
      5     0     0]
 [  252     0     0     0     0   833     0   537     0   376     9     0
      0     0     3]
 [   20     1     0     0     2     2    94   104     0    60     0     0
      0     0     0]
 [ 1097     1    25     0     0    67     1 55829     4   492     0     0
      0     0     0]
 [   37     0     1     0     0     4     0   308   163    31     0     0
      0     0     0]
 [  827     8     7     0     0   153     3  3439     5  8150     0     8
      3     0    35]
 [    2     0     0     0     0    12     1     8     0    14   119     0
      0     0     0]
 [   11    10     0     0     0     2     0    19     0   151     0  2604
      1     0    24]
 [    0     0     0     0     0     0     0     0     0     0     0     0
    101     0     0]
 [   98    19     2     0     0    15     0   294     0  1049     0    24
      0     0    51]
 [   84    36     0     0     0     8     1   225     0  1346     0    12
      0     0   126]]
```

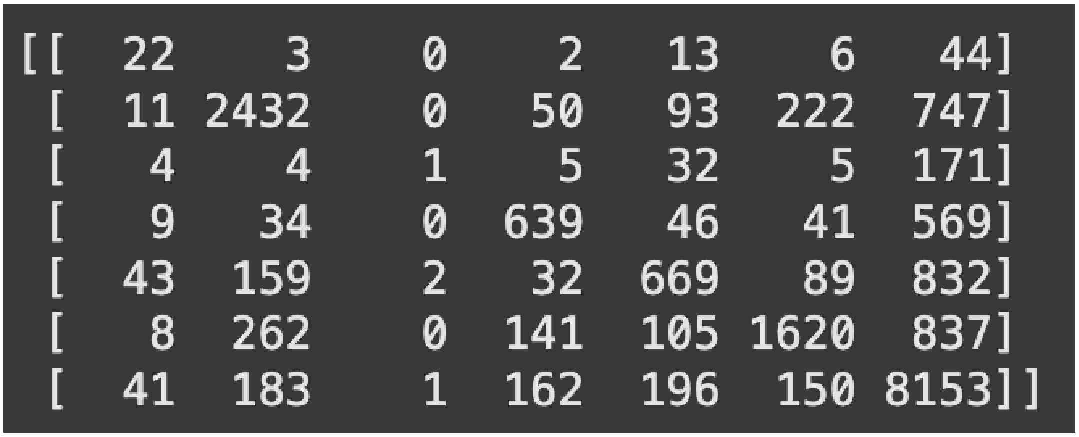
Classification into Subcategories:

- Online and Social Media Related Crime
  - F-1 Score = 0.6153660304421358
  - Confusion Matrix



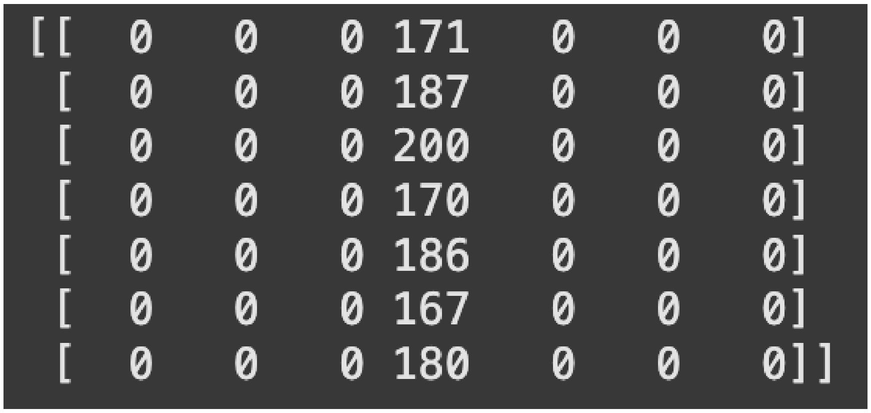
```
[[ 347  157    2  119    2    0   38    0   43   11]
 [  62 1133    1   93    1    0    4    5   47   20]
 [  17    3    7    5    4    0    2    0   16    0]
 [ 172  139    1  364    1    0   13    0   62   11]
 [   1    1    3    0    3    0    0    0    4    1]
 [   0    7    0    0    2    0    0    0    2    0]
 [  48    5    1   12    2    0  224    0    2    0]
 [   5   12    0    2    0    0    1   16    1    1]
 [  57  148    1  110    2    0    1    2  423    7]
 [   7   72    0   11    1    0    1    0    8   30]]
```

- Online Financial Fraud
  - F-1 Score = 0.716569613552144
  - Confusion Matrix



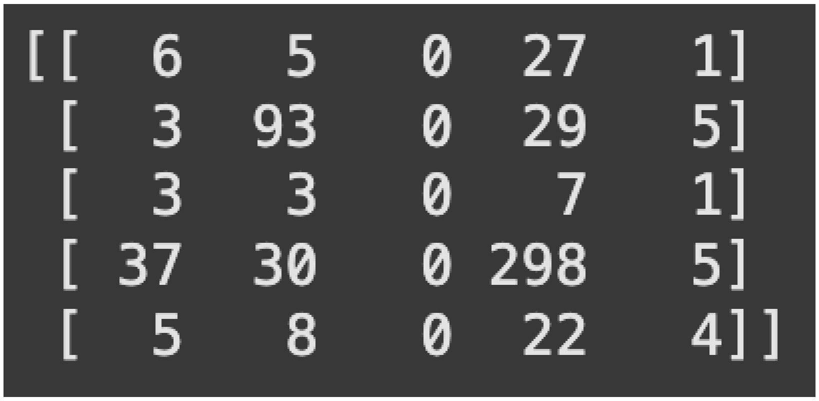
```
[[  22    3    0    2   13    6   44]
 [  11 2432    0   50   93  222  747]
 [   4    4    1    5   32    5  171]
 [   9   34    0  639   46   41  569]
 [  43  159    2   32  669   89  832]
 [   8  262    0  141  105 1620  837]
 [  41  183    1  162  196  150 8153]]
```

- Cyber Attack/ Dependent Crimes
  - F-1 Score = 0.13481363996827914
  - Confusion Matrix



```
[[  0   0   0 171   0   0   0]
 [  0   0   0 187   0   0   0]
 [  0   0   0 200   0   0   0]
 [  0   0   0 170   0   0   0]
 [  0   0   0 186   0   0   0]
 [  0   0   0 167   0   0   0]
 [  0   0   0 180   0   0   0]]
```

- Hacking Damage to computercomputer system etc
    - F-1 Score = 0.6773648648648649
    - Confusion Matrix



```
[[  6   5   0  27   1]
 [  3  93   0  29   5]
 [  3   3   0   7   1]
 [ 37  30   0 298   5]
 [  5   8   0  22   4]]
```

Conclusion

The BERT model outperforms the SVM model, which was the best performing model amongst all the traditional ML models used in this task.

The BERT model gives good results while predicting the categories, but does not perform very well while predicting the subcategories of the category “Cyber Attack/ Dependent Crimes”, mainly because of the faulty data. In this category, the same message has been categorised into various subcategories, thus confusing the model. The model predicts the same subcategory for all the messages. Though it might have decent accuracy, it does not have a good f1 score.

References

The following libraries and frameworks were used for this project:

- Numpy
- Pandas
- Matplotlib
- Seaborn
- NLTK

- Spacy
- Scikit Learn
- Transformers - BERT
- Langchain
- OpenAI

Appendices

Appendix A: Training Logs and Parameters

For this project, I funetuned BERT models for text classification. There are 5 models in total. Following are the training logs obtained while finetuning the model:

```
======== Epoch 1 / 4 ========
Training...

  Average training loss: 1.16
  Training epoch took: 0:38:20

Running Validation...
  Accuracy: 0.64

======== Epoch 2 / 4 ========
Training...

  Average training loss: 1.01
  Training epoch took: 0:38:18

Running Validation...
  Accuracy: 0.65
```

```
======== Epoch 3 / 4 ========
Training...

  Average training loss: 0.93
  Training epoch took: 0:38:42

Running Validation...
  Accuracy: 0.66

======== Epoch 4 / 4 ========
Training...

  Average training loss: 0.89
  Training epoch took: 0:38:20

Running Validation...
  Accuracy: 0.65

Training complete!
Total training took 2:33:40 (h:mm:ss)
```

Appendix B: Code Repository Structure

The Github repository has the following structure:

- indiaai_hackathon
    - data/
        - synthetic_train.csv
        - test.csv
    - models

- notebooks
    -
- readme.md