Report Phase 1

Team Name:

Ciphernauts

Team Members :

Unnati Kotian

Jalp Kumar

Simran Malakar

Arastu Thakur

Problem Statement

Now-a-days, government portals get a lot of complaints about a variety of problems, such as fraud and other concerns. It takes a lot of effort and due to human mistake it is difficult to manually analyze and classify these complaints, particularly when dealing with a variety of complaint categories, victims, and distinct fraud tendencies. Both operational efficiency and customer happiness are impacted by this manual procedure, which slows down response times, makes it more difficult to spot trends, and delays the essential steps to combat fraud. So, to tackle this challenge, we aim to develop an NLP (Natural Language Processing) model capable of automatically classifying complaints based on several key parameters. This model will not only streamline complaint categorization but also enable quick identification of recurring issues, detect emerging fraud patterns, and provide actionable insights. By automating this process, firms/organizations can respond faster to complaints, improve fraud detection accuracy, and ultimately enhance the customer experience by resolving concerns proactively.

Project Description

1. Initial Dataset Challenges and Preprocessing

Upon examining the dataset, we observed that certain columns, particularly the sub-category and crimeadditionalinfo columns, had numerous empty or sparse values. These missing values presented a challenge for effective categorization and analysis. The lack of detailed information within these columns could potentially hinder the model's ability to correctly classify complaints and identify patterns within different types of fraud or crime. To address these gaps, we opted for filling the values with firstly filling rows with dummy values and later on removing some null values while performing oversampling.

2. Exploratory Data Analysis (EDA) on the Raw Dataset

The purpose of EDA is to gain a deeper understanding of the data’s structure, distributions, and relationships between variables, as well as enable us to identify specific characteristics or trends within complaint categories, such as which sub-categories of complaints occur most frequently and any common themes within the crimeadditionalinfo column. The training and testing dataset was first merged, and then EDA followed by other processing tasks were performed.

3. Addressing Class Imbalance with SMOTE

Many complaint datasets tend to have an uneven distribution of classes, where certain types of complaints or fraud categories are overrepresented while others are significantly underrepresented. This imbalance can lead to biased models that perform well on the majority classes but poorly on minority classes, which are often the most critical to detect accurately. To address this, we plan to apply **SMOTE (Synthetic Minority Over-sampling Technique)**. SMOTE is a powerful technique for generating synthetic samples in the minority class to create a more balanced dataset.

4. Model Implementation and Performance Evaluation

With the dataset preprocessed, balanced, and analyzed, the next phase is **model training and evaluation**. This stage involves selecting a suitable algorithm for text classification and exploring potential models that best fit the nature of the data and classification objectives. Model selection will be driven by cross-validation, hyperparameter tuning, and careful analysis of each algorithm’s strengths and limitations for this classification problem.

Once the model is trained, we will assess its performance using standard metrics, including:

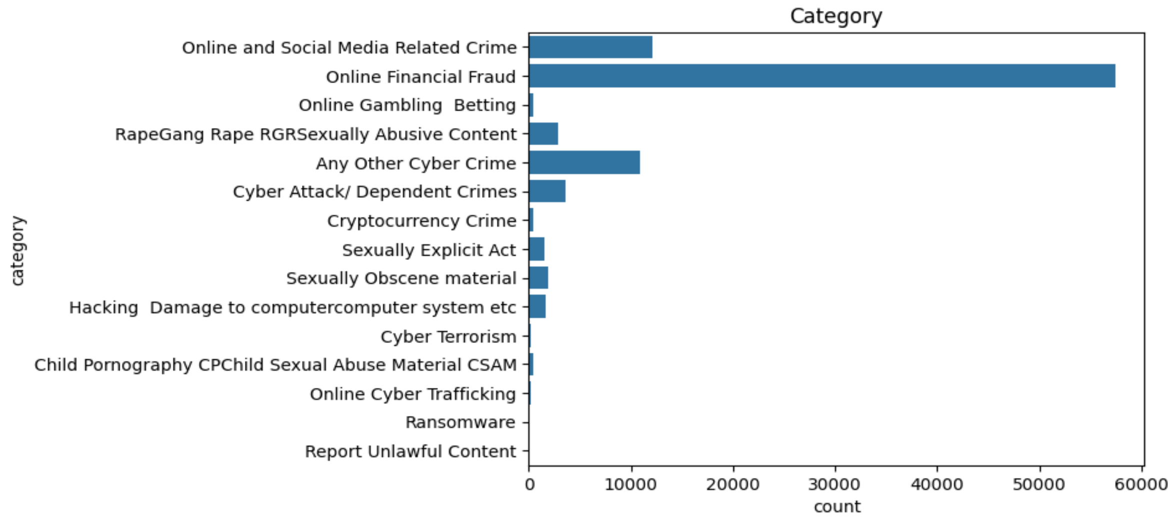
- **Accuracy**: To understand the overall correctness of the model.
- **Precision and Recall**: Precision is critical to reduce false positives in fraud detection, while recall ensures that we capture as many relevant cases as possible, minimizing false negatives.
- **F1-Score**: This metric provides a balance between precision and recall, offering a holistic measure of the model’s reliability, particularly for datasets with class imbalance.
- **And various other parameters**

Implementation:

Exploratory Data Analysis (EDA)

Using summary statistics and graphical representations, analysts analyze a dataset to find patterns, identify anomalies, test hypotheses, and verify assumptions.

Bar Chart



The provided bar chart displays the counts of different types of cybercrimes reported. Here's a breakdown of what it represents:

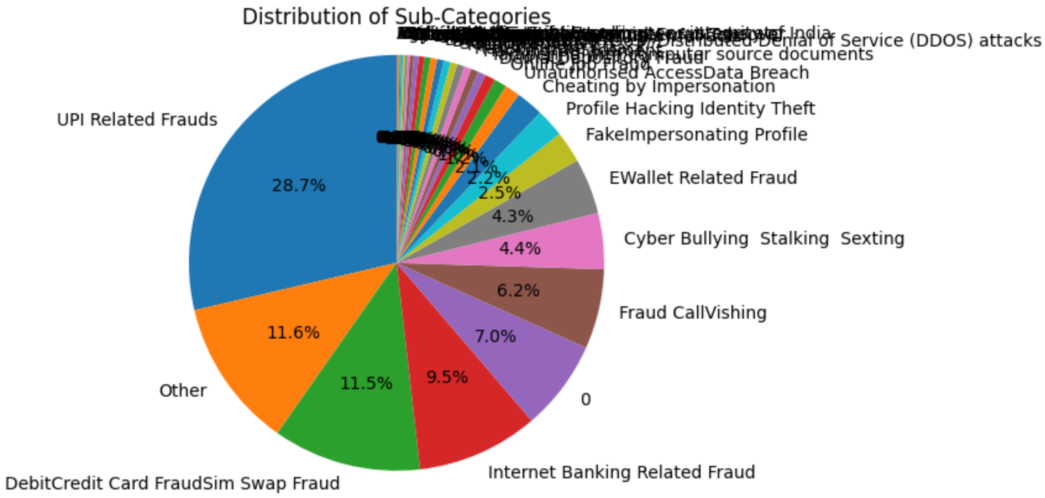
- **X-Axis (Count)**: This axis shows the number of reports or occurrences for each cybercrime category.
- **Y-Axis (Category)**: The various types of cybercrimes are listed along this axis. Some of these categories include:
  - "Online Financial Fraud"
  - "Online and Social Media Related Crime"
  - "Rape Gang Rape RGR Sexually Abusive Content"
  - "Cyber Attack/Dependent Crimes," and so on.

From this chart, we can observe the following key insights:

1. **Most Common Crime Category**:
   - The bar for "Online Financial Fraud" is the longest, indicating that this is the most frequently reported category, with almost 60,000 incidents.
2. **Other Notable Categories**:
   - "Online and Social Media Related Crime" and "Any Other Cyber Crime" are also common, though significantly less frequent than financial fraud.
3. **Less Frequent Categories**:

- Categories such as "Ransomware," "Child Pornography/Child Sexual Abuse Material (CSAM)," and "Cyber Terrorism" have very low counts, suggesting they are less frequently reported or less common in this dataset.

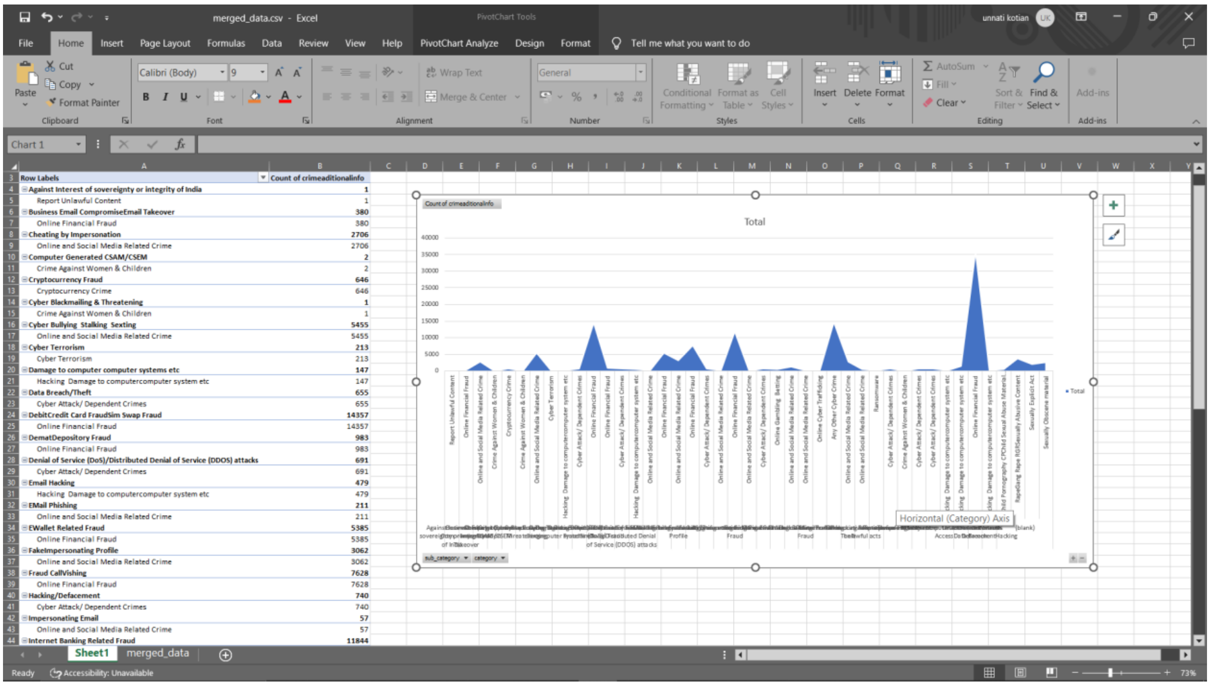
Pie Chart

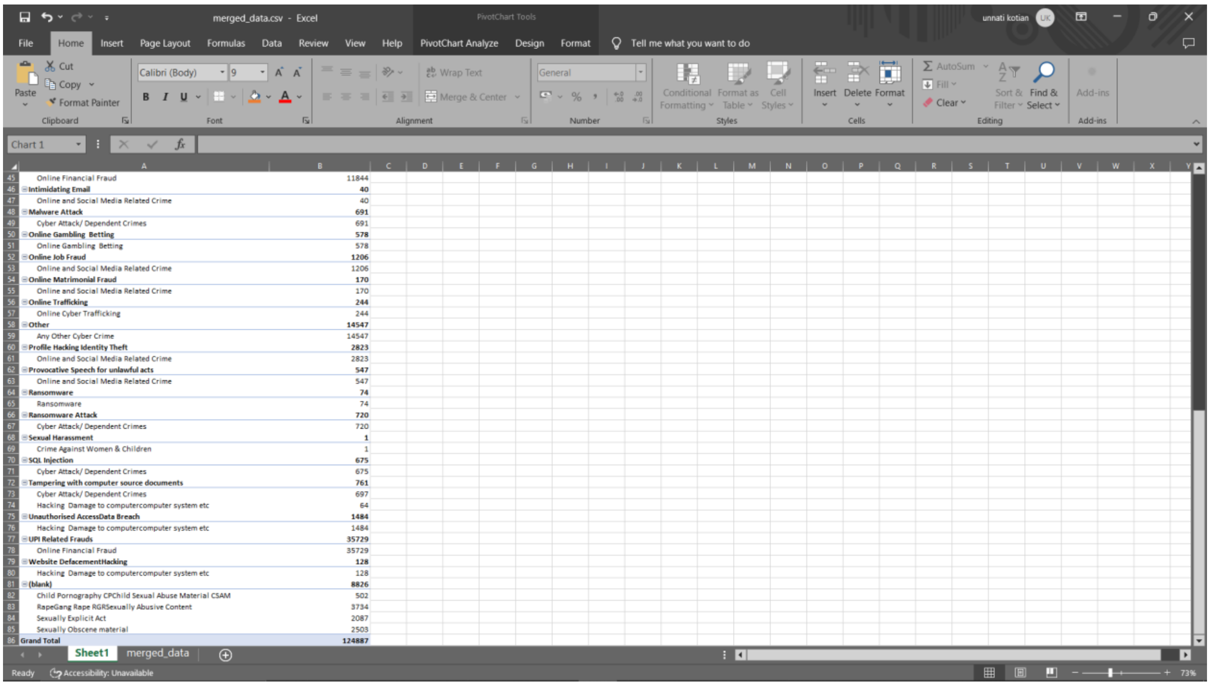


This pie chart shows the **distribution of sub-categories of cybercrimes** reported in the dataset. Each slice of the pie represents a specific sub-category, and the size of each slice corresponds to the proportion of that sub-category within the entire dataset. Here's a breakdown of the significant findings:

Key Observations

1. **Most Common Sub-Category - "UPI Related Frauds"**:
   - The largest portion of the pie (28.7%) is taken up by **UPI (Unified Payments Interface) Related Frauds**.
2. **Other Notable Sub-Categories**:
   - **"Other" (11.6%)**: This category is broad and might include various types of cybercrimes that do not fall under specific named categories.
   - **"Debit/Credit Card Fraud/Sim Swap Fraud" (11.5%)**: This type of fraud is also quite common, likely due to increased online transactions and associated risks.
   - **"Internet Banking Related Fraud" (9.5%)**: Similarly, frauds related to internet banking represent a significant portion, pointing to vulnerabilities in online banking security.
3. **Smaller Sub-Categories**:
   - Several other sub-categories are present with smaller proportions, such as:
     - **"Cyber Bullying, Stalking, Sexting" (4.3%)**
     - **"EWallet Related Fraud" (2.5%)**
     - **"Fraud Call/Vishing" (7.0%)**
     - **"Fake/Impersonating Profile"** and **"Profile Hacking/Identity Theft"**





From the above two charts, first one represents all subcategories along with their values and a bar chart, the second chart is continuous of first chart of subcategories separated in groups individually, wherein at the end total amount of rows is written together.

Model Evaluation Report for Cybercrime Classification using various models

The Random Forest model achieved an overall accuracy of **73%**. This means that approximately 73 out of every 100 cases were correctly classified according to the model's predictions.

```
[ ] import pandas as pd
    from sklearn.preprocessing import LabelEncoder
    from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score, classification_report

    # Load the train and test data
    train_df = pd.read_csv('train_filled.csv')
    test_df = pd.read_csv('test_filled.csv')

    # Fit LabelEncoder on the union of 'sub_category' from both train and test
    label_encoder = LabelEncoder()
    all_labels = pd.concat([train_df['sub_category'], test_df['sub_category']]).unique()
    label_encoder.fit(all_labels)
```

```
[ ] train_df['sub_category_encoded'] = label_encoder.transform(train_df['sub_category'])
    test_df['sub_category_encoded'] = label_encoder.transform(test_df['sub_category'])

    # Separate features and target for train data
    X_train = train_df.drop(columns=['sub_category', 'sub_category_encoded', 'crimeaditionalinfo'])
    y_train = train_df['sub_category_encoded']

    # One-Hot Encode 'category' column in train and test
    X_train = pd.get_dummies(X_train, columns=['category'], drop_first=True)
    X_test = pd.get_dummies(test_df.drop(columns=['sub_category', 'sub_category_encoded', 'crimeaditionalinfo']),
                            columns=['category'], drop_first=True)

    # Align columns of X_test with X_train to ensure they have the same features
    X_test = X_test.reindex(columns=X_train.columns, fill_value=0)

    # TF-IDF Transformation on 'crimeaditionalinfo' for train and test
    tfidf = TfidfVectorizer(max_features=500)  # Adjust max_features if necessary
    tfidf_train = tfidf.fit_transform(train_df['crimeaditionalinfo']).toarray()
    tfidf_test = tfidf.transform(test_df['crimeaditionalinfo']).toarray()

    # Convert TF-IDF result to DataFrames and concatenate with other features
    tfidf_train_df = pd.DataFrame(tfidf_train, columns=[f"tfidf_{i}" for i in range(tfidf_train.shape[1])])
    tfidf_test_df = pd.DataFrame(tfidf_test, columns=[f"tfidf_{i}" for i in range(tfidf_test.shape[1])])

    X_train = pd.concat([X_train.reset_index(drop=True), tfidf_train_df], axis=1)
    X_test = pd.concat([X_test.reset_index(drop=True), tfidf_test_df], axis=1)

    # Train a RandomForest Classifier without oversampling
    clf = RandomForestClassifier(random_state=42)
    clf.fit(X_train, y_train)

    # Make predictions on the test set
    y_pred = clf.predict(X_test)

    # Calculate accuracy
    accuracy = accuracy_score(test_df['sub_category_encoded'], y_pred)

    # Generate classification report with only the classes in the test set
    report = classification_report(
        test_df['sub_category_encoded'],
        y_pred,
        target_names=label_encoder.classes_[test_df['sub_category_encoded'].unique()]
    )

    print(f"Accuracy: {accuracy}")
    print("Classification Report:")
    print(report)
```

```
Accuracy: 0.7256076083127861
Classification Report:
                                                           precision    recall  f1-score   support

                                                   Unknown       0.33      0.01      0.02        90
                            DebitCredit Card FraudSim Swap Fraud       0.48      0.45      0.46       719
                                             SQL Injection       0.00      0.00      0.00         2
                                         Fraud CallVishing       1.00      1.00      1.00       166
                                                     Other       0.00      0.00      0.00         1
                             Internet Banking Related Fraud       0.61      0.83      0.70      1366
                             Unauthorised AccessData Breach       1.00      0.52      0.68        52
                                       UPI Related Frauds       1.00      0.03      0.05        39
                       Damage to computer computer systems etc       0.14      0.13      0.14       171
                                 Cheating by Impersonation       0.77      0.69      0.73      3556
                                            Malware Attack       0.40      0.01      0.02       222
                                      EWallet Related Fraud       0.13      0.14      0.14       187
                                             EMail Phishing       0.50      0.02      0.04        54
                             Profile Hacking Identity Theft       0.74      0.32      0.45      1338
                                         Data Breach/Theft       0.75      0.52      0.61       130
                                 FakeImpersonating Profile       0.55      0.47      0.51       763
                                             Email Hacking       0.61      0.31      0.41      1827
                                          Online Job Fraud       0.17      0.16      0.16       200
                             Cyber Bullying  Stalking  Sexting       0.00      0.00      0.00        13
                                       Hacking/Defacement       0.77      0.54      0.63      2973
                                     Cryptocurrency Fraud       0.00      0.00      0.00        11
                                   Online Matrimonial Fraud       0.10      0.10      0.10       170
                     Tampering with computer source documents       1.00      0.99      1.00       134
Denial of Service (DoS)/Distributed Denial of Service (DDOS) attacks       0.84      0.60      0.70       294
                                     DematDepository Fraud       0.00      0.00      0.00        38
                          Provocative Speech for unlawful acts       1.00      0.57      0.73        61
                                   Online Gambling  Betting       0.98      1.00      0.99      3670
                                          Ransomware Attack       0.63      0.58      0.60       751
                   Business Email CompromiseEmail Takeover       0.67      0.06      0.11       130
                                         Online Trafficking       0.00      0.00      0.00        18
                                            Cyber Terrorism       0.12      0.12      0.12       186
                                        Impersonating Email       0.11      0.12      0.12       167
                                 Website DefacementHacking       0.00      0.00      0.00         1
                                                 Ransomware       0.17      0.15      0.16       194
                                 Computer Generated CSAM/CSEM       0.69      0.93      0.79      8890
                                         Intimidating Email       0.70      0.95      0.81       370
                              Cyber Blackmailing & Threatening       1.00      0.99      1.00      2236
                                         Sexual Harassment       0.00      0.00      0.00        39

                                                  accuracy                           0.73     31229
                                                 macro avg       0.47      0.35      0.37     31229
                                              weighted avg       0.72      0.73      0.70     31229
```

Results:

Algorithms	Accuracy
Tf-Idf	56%
SVM	56%
Decision Tree	60%
Random Forest	73%
XGBoost	89%
LightBGM	90%

Key Findings

1. **High-Performing Categories**:
    - **Online Gambling/Betting**: Achieved high precision (0.98), recall (1.00), and F1-score (0.99), indicating the model's strong ability to classify this category accurately.
    - **Cyber Blackmailing & Threatening**: With both precision and recall at 1.00, this category saw perfect classification performance, likely due to well-defined patterns in the data.
    - **Computer Generated CSAM/CSEM**: This category also performed well with an F1-score of 0.79, showing the model's effectiveness in identifying this sub-category.
2. **Moderate-Performing Categories**:
    - **Internet Banking Related Fraud**: Achieved a decent F1-score of 0.70, with high recall (0.83), indicating it can capture a significant portion of this fraud type.
    - **Ransomware Attack**: F1-score of 0.60 suggests moderate success in classifying ransomware-related incidents.
3. **Low-Performing Categories**:
    - **Cyber Bullying, Stalking, Sexting** and **Cryptocurrency Fraud**: Both categories had very low recall and F1-scores, reflecting poor classification accuracy. This could be due to limited training data or overlapping features with other categories.
    - **Email Phishing, Impersonating Email, and Online Trafficking**: Also had low performance metrics, indicating a need for more data or improved feature engineering for these sub-categories.
4. **Challenging Categories**:
    - **Other**: The "Other" category, with zero precision and recall, may have too broad a definition, leading to poor model performance. This category could benefit from further refinement or re-labeling.

Evaluation Metrics

The **macro average** precision, recall, and F1-score are relatively low (precision: 0.47, recall: 0.35, F1-score: 0.37), highlighting that the model performs unevenly across different categories. The **weighted average** F1-score of 0.70 indicates that the model generally performs well but struggles with less common classes.

Other models suggest that models like catboost and various deep learning models will make the project work better.

https://github.com/unnatikotian/cyberguardAIhackathon

Github link contains:

1. EDA colab file as pdf
2. EDA colab file ipynb
3. Phase1(basic models) pdf
4. Phase1(basic models) ipynb
5. Updated file (xlc.. - main) – pdf

6. Updated file (xlc.. - main) ipynb