**Burmese Money Scam Classification:**

**Project Report**

**Abstract**

Financial fraud and scams have surged in Myanmar since the February 2021 coup, with social media platforms becoming hotspots for deceptive activities. This project aims to develop a machine learning-based classification model to detect fraudulent messages in Burmese, a low-resource language. We collected and processed a dataset of approximately 19,000 messages from various sources, implemented several feature extraction techniques, and trained multiple machine learning models. Our study highlights the best-performing model and outlines strategies to mitigate bias and improve generalization. Finally, we discuss deployment and future improvements.

**Introduction**

**Background and Problem Statement**

The rise of digital fraud in Myanmar has been exacerbated by political instability, economic hardship, and widespread internet access. Various scams, including gambling schemes, phishing, fake advertisements, and financial fraud, have victimized many, particularly the youth and uneducated population. Fraudulent activities, especially on Facebook, have become increasingly sophisticated, making detection difficult. Given that Burmese is a low-resource language with limited NLP tools, detecting scams using machine learning presents unique challenges. Our project aims to address these challenges by developing a reliable scam classification model that can be deployed for public use.

**Data Collection**

**Data Sources**

Data was collected from:

* Public social media posts (Facebook, Telegram)
* Victim-shared experiences
* Common scam groups
* Personal messages sent to mobile phones

**Data Characteristics**

Due to the scarcity of scam-related Burmese text data, we ensured:

* Collection of legitimate advertisements from banks and official sources
* Inclusion of diverse scam categories (e.g., gambling, investment fraud, fake job postings)
* Generation of 400 synthetic scam messages to augment data

**Dataset Composition**

Total dataset size: **19,347 messages**

* Non-scam: **10,905**
* Potential scam: **2,481**
* Scam: **5,961**

**Strategies to Improve Data Quality**

To avoid model bias and ensure robust performance, we:

* Balanced dataset length distribution
* Augmented scam messages with variations
* Included edge cases (e.g., misleading but legitimate content)
* Collected data from diverse sources

**Data Preprocessing**

**Standardization and Cleaning**

* Placeholder replacement (e.g., URLs, phone numbers, money amounts)
* Emoji removal and counting
* Stopword removal (English and Burmese)
* Standalone number removal
* Hashtag processing
* Punctuation counting
* Text normalization (lowercasing, removing extra spaces)

**Tokenization**

* Burmese and English text were tokenized separately using:
  + Custom myTokenize for Burmese (syllable and word tokenization)
  + nltk for English

**Model Training**

**Feature Extraction**

We experimented with the following feature extraction techniques:

* **TF-IDF** (syllable and word tokenization)
* **Word2Vec** (trained on dataset)
* **FastText** (trained on dataset)

**Model Selection**

We trained and evaluated six models:

* Logistic Regression
* Decision Tree
* Random Forest
* XGBoost
* SVM
* Naive Bayes

**Training and Testing Strategy**

* **Stratified K-Fold Cross-Validation** to handle class imbalance
* **Train-Validation-Test Split** (80%-20%, with validation split from training)
* **Hyperparameter Tuning** for the best model

**Results**

**Training Performance (F1-Score)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Tfidf(syllable tokenization)** | **Tfidf(word tokenization)** | **word2vec** | **Fasttext** |
| Logistic regression | 0.8673 | 0.8765 | 0.8586 | 0.8388 |
| Decision Tree | 0.8211 | 0.8232 | 0.7882 | 0.771 |
| Random Forest | 0.877 | 0.8775 | 0.8518 | 0.8389 |
| XGBoost | 0.8831 | 0.8902 |  |  |
| **SVM** | **0.8878** | **0.893** | **0.8738** | **0.8644** |
| Naive Bayes | 0.8476 | 0.8594 |  |  |

**Testing Performance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Extraction** | **Best model** | **F1-Score** | **Precision** | **Recall** | **ROC-AUC** |
| Tfidf(syllable tokenization) | SVM | 0.8852 | 0.8932 | 0.8814 | 0.9673 |
| **Tfidf (word tokenization)** | **SVM** | **0.8935** | **0.9019** | **0.8897** | **0.9713** |
| word2Vec | SVM | 0.8812 | 0.8963 | 0.8742 | 0.9657 |
| Fasttext | SVM | 0.8759 | 0.8906 | 0.869 | 0.9629 |

# ****Final Evaluation on Test Set****

**Overall Performance Metrics:**

* **Accuracy**: 0.8897
* **F1-Score**: 0.8935
* **Precision**: 0.9019
* **Recall**: 0.8897
* **ROC-AUC**: 0.9713
* **Log Loss**: 0.2826

**Confusion Matrix:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual \ Predicted** | **0 (Potential Scam)** | **1 (Non-Scam)** | **2 (Scam)** |
| **0 (Potential Scam)** | **416** (TP) | 42 (FN) | 38 (FN) |
| **1 (Non-Scam)** | 147 (FP) | **1991** (TP) | 43 (FN) |
| **2 (Scam)** | 90 (FP) | 67 (FP) | **1036** (TP) |

### ****Classification Report:****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| **0 (Potential Scam)** | 0.64 | 0.84 | 0.72 | 496 |
| **1 (Non-Scam)** | 0.95 | 0.91 | 0.93 | 2181 |
| **2 (Scam)** | 0.93 | 0.87 | 0.9 | 1193 |
| **Overall** | - | - | - | **3870** |

### ****Findings & Observations:****

* **Potential scam (Label 0) has the lowest performance** among the three categories, with lower precision (0.64) compared to recall (0.84). This suggests that the model is good at identifying most potential scam messages but struggles with false positives.
* **Scam (Label 2) and Non-Scam (Label 1) classifications perform better**, with high precision and recall, leading to strong F1-scores (0.90+).
* **Class imbalance may have contributed to the performance gap**, as potential scam messages (496 samples) are much fewer than non-scam messages (2181 samples), leading to less representation in test data set.
* **False positives in class 0 (Potential Scam) are high**, possibly due to overlapping linguistic features with other classes, making differentiation challenging.
* **ROC-AUC (0.9713) suggests strong overall discrimination ability** across all classes, but fine-tuning is needed for underperforming categories.

### ****Failure Analysis:****

1. **Misclassification of Potential Scam (Label 0) as Non-Scam (Label 1):**
   * Observed from 147 false positives in the confusion matrix.
   * This could be due to subtle linguistic similarities between promotional messages and actual scam messages.
   * Possible Solution: Augment training data for class 0 and introduce more context-specific linguistic markers.
2. **Misclassification Between Scam (Label 2) and Non-Scam (Label 1):**
   * 67 non-scam messages were misclassified as scam.
   * 43 scam messages were misclassified as non-scam.
   * Possible Solution: Further refining feature engineering to detect nuanced fraudulent patterns.
3. **Overlapping Features Across Classes:**
   * Some legitimate financial or promotional messages may resemble scam patterns.
   * Possible Solution: Use more advanced NLP techniques like transformers to capture better contextual differences.

### ****Next Steps for Improvement:****

* **Data Augmentation**: Increase representation of potential scam messages.
* **Feature Engineering**: Introduce topic modelling or syntactic features to better differentiate scams from non-scams.
* **Model Optimization**: Experiment with ensemble models or transformer-based architectures.
* **Active Learning**: Incorporate human feedback to improve classification in ambiguous cases.

While the model performs well overall, further refinement is needed for potential scam detection. Addressing data imbalance, enhancing linguistic features, and leveraging contextual learning can improve future iterations of the classifier.

**Deployment**

Using **Streamlit**, we developed a simple user interface where:

* Users input a message
* The message undergoes preprocessing and tokenization
* The **SVM model** predicts the classification
* The output includes:
  + **Danger Level:** 🟥 High Risk, 🟧 Potential Scam, 🟩 Safe
  + **Prediction Confidence**
  + **Suggestion for the user**

**Findings**

* **TF-IDF with word tokenization** performed the best
* **SVM** consistently outperformed other models
* **Word2Vec and FastText** were less effective, likely due to limited training data

**Limitations**

* **Scammers frequently change tactics**, requiring regular data updates
* **Low-resource NLP challenges** make feature extraction difficult
* **Limited labeled data**, impacting generalization

**Future Improvements**

1. **Regular Dataset Updates**: Continuously collect new scam data
2. **Semi-Supervised Learning**: Leverage unlabeled data to improve performance
3. **Multilingual Transfer Learning**: Use multilingual models to enhance Burmese text processing
4. **Anomaly Detection Techniques**: Identify new scam trends using anomaly detection
5. **Ethical Considerations**: Ensure privacy and compliance with data protection laws

**Conclusion**

This project successfully developed a **Burmese money scam detection model** with **SVM and TF-IDF** achieving the best performance. The model has been deployed for public use through **Streamlit**, providing an accessible tool to help people in Myanmar identify fraudulent messages. Future enhancements will focus on improving dataset quality, integrating multilingual models, and ensuring long-term usability.