Report #3

Classifier for Humanitarian Response

Final Report: One- and Multi-Label Classification for Humanitarian Assistance Analysis

Authors: Ghiath Al Jebawi, Bercin Ersoz, Allaeldene Ilou, Caspar Stordeur

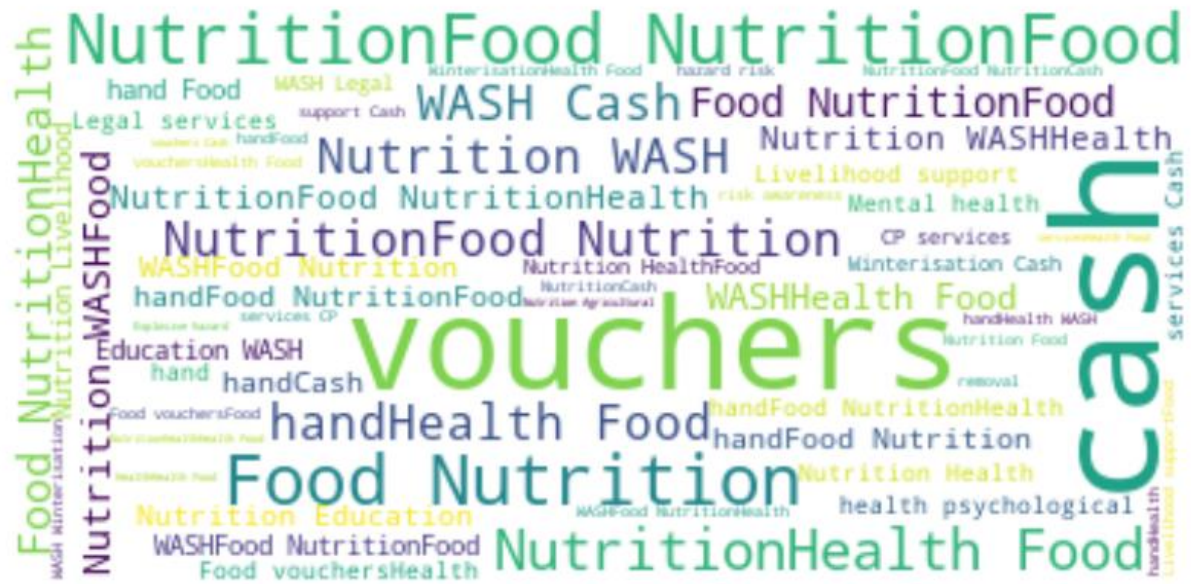

Table of Contents

List of Figures

List of Tables

1. Context

1.1 Project Context

In humanitarian crises, international non-governmental organizations (NGOs) face significant challenges in identifying needs and targeting communities due to the complexity and volume of data. Despite the availability of large datasets, data often remains underutilized, not transformed into actionable insights, or easily accessible for NGOs.

Syria, experiencing one of the largest humanitarian crises in recent decades, has a vast ecosystem of NGOs working to support communities. The availability of humanitarian data in Syria is substantial compared to other conflict zones, making it a suitable case for this project.

1.2 Technical Context

NGOs such as UNHCR and IMPACT REACH Initiatives have developed a consistent database over the past six to seven years, updated monthly, describing thousands of communities with internally displaced people (IDPs). The data includes eleven key indicators (e.g., Health, Shelter, Education), each divided into numerous sub-indicators. Data is not easily summarized or utilized by local NGOs. The proposed model aims to base NGO interventions on three key aspects:

- **Aid** provided to communities by NGOs.
- **Needs** defined by the communities themselves.
- **Vulnerability** criteria, such as the ratio of income to survival costs.

1.4 Feasibility Context

The goal is to make the available data easily usable by creating a system that describes each community's needs and priorities for intervention. The model will predict future vulnerability trends and recommend interventions, optimizing the costly process of resource allocation. If successful, this model could significantly improve the efficiency of aid distribution, ensuring that resources reach the most vulnerable communities in the shortest possible time.

2. Objectives

2.1 Main Objectives

The primary objective is to transform available data (monthly time series over five years) into a predictive and classifier model. The model will learn from the data to predict future vulnerability trends and assist decision-makers in prioritizing aid and planning interventions.

2.2 Team Expertise

The team consists of four members with backgrounds in data analytics, data science, and statistics:

- **Ghiath Al Jebawi**: Data scientist with five years of experience in humanitarian work in Syria, particularly in IDP camps.
- **Bercin Ersoz**: Statistician with experience in data management, previously working at the Turkish Central Bank.
- **Allaeldene Ilou**: Freelancer with a background in Computer Science and Data Science, focusing on humanitarian and environmental projects.
- **Caspar Stordeur**: Social scientist with expertise in data transformation and analysis, with a focus on demographics and community dynamics.

3. Framework

The primary dataset is the **Humanitarian Situation Overview in Syria (HSOS)**, which provides monthly data from 2019 to 2024 for **Northeast Syria (NES)** and **Northwest Syria (NWS)**. We are interested in all variables that provide information about the 11 indicator groups that the HSOS surveyed. The groups are:

1. Demographics
2. Shelter
3. Electricity & Non-food Items (NFIs)
4. Food Security
5. Livelihoods
6. Water, Sanitation, and Hygiene (WASH)
7. Health
8. Education
9. Protection
10. Accountability and Humanitarian Assistance
11. Priority Needs

In total, we had 110 HSOS files. To ensure data consistency, we limited the dataset to the years 2021 to 2023, resulting in 61 files. Notably, there is no consistent catalog of questions across the files. These 61 files alone contained over 5,000 indicators. To maintain consistency, we retained only those indicators that were present in 57 of the 61 files. Unfortunately, this process excluded indicators related to education, which we manually added back into our sample. The final analysis sample consists of 1,668 indicators and 52,095 observations.

An additional aspect is derived from the Joint Market Monitoring Initiative (JMMI) data, which focuses on the Survival Minimum Expenditure Basket (SMEB) across different regions of Syria. The analysis explores price trends, regional variations, and the relationships between various components of the SMEB. The JMMI was initiated by the Cash Working Group to better understand market dynamics and challenges in Syria. All time points for the regional subset are consolidated into a single file Similarly, to ensure consistency, we restricted the data to the years 2021 to 2023. For this analysis, we utilized the January 2024 files for Northeast Syria (NES) and Northwest Syria (NWS). The original dataset included 123 columns, featuring detailed prices for individual items. However, since the dataset also contained subtotals for these items, we decided to exclude the individual item prices and focus on the subtotals, totals, and categorical data. The final dataset contains 21 indicators and over 4,700 observations.

4. Pre-processing and Feature Engineering

4.1 Final Dataset Structure and Sampling

The key variables under investigation include:

Target Variables: Humanitarian assistance provided to IDP households in the community over the last 30 days. This group consists of 16 binary columns representing different types of assistance (e.g., Shelter, Food, Health). We approach in this report our target variables in 2 ways:

> Approach 1: A single numeric target aggregating assistance metrics. This simplifies modeling but may obscure granularity.
>
> Approach 2: A matrix of 16 binary targets, where each represents a specific type of aid. This allows for multi-output classification modeling.

Features: Challenges and barriers to accessing aid, community demographics, coping strategies, market prices and so on.

4.2 Data Cleaning and Treatment

For further analysis, we made an important decision regarding the HSOS data: to focus on the IDP (Internally Displaced Persons) data rather than the resident population data. This decision was made primarily to reduce data processing and preparation time. However, a fully functional classifier model should ideally consider both datasets. In communities where NGO aid is provided only to IDPs, tensions can arise with the resident population. As a result, NGOs have increasingly begun to treat IDPs and residents as a single community, expanding their needs assessments and aid delivery to include both groups proportionately. The dataset focuses on humanitarian aid provided to IDP households. For this analysis **32.912 rows** were available.

To streamline the dataset and focus on actionable insights, redundant or irrelevant columns, such as those ending with "- Other", were removed.

The **target matrix** contained **53 missing values** across each category. Given the small number of missing entries, these rows were **excluded** from the dataset rather than being imputed, to avoid introducing bias.

Other pre-processing steps are summarized as follows.

4.2.1 Mapping and Encoding

- **Binary responses** (e.g., Yes/No) were mapped to **1/0**.
- Unnecessary categories such as **"No Data"** or **"NA"** were replaced with **np.nan** to ensure usability.
- **Proportion and percentage ranges** were mapped using thresholds (e.g., 10%, 20%, 25%).

4.2.2 Categorical Variables

- **One-Hot Encoding** was applied to variables related to **Governorate** (6 unique values).
- **One-Hot Encoding** was also applied to the **Month** variable to capture potential seasonal fluctuations in the data. This approach ensures the model can account for temporal variations without assuming a linear relationship between months.

- **One-Hot Encoding** was implemented for the **Top Priority (Need)** variables to preserve their categorical nature. This allows the model to consider the unique importance of each need without imposing ordinal or linear relationships between them.
- **Frequency Encoding** was used for variables related to **District** (23 unique values). This approach was chosen to avoid increasing the number of columns, which were already numerous.

4.2.3 Transformation, Normalization, and Standardization

Both the HSOS and Market datasets contain price-related columns. These columns are planned to be normalized using StandardScaler prior to modeling. Numerical features with different scales, such as expenditures and **daily wages**, are normalized to ensure a fair comparison and to prevent differences in scale from disproportionately affecting the analysis. These transformations are critical to ensure that all features contribute appropriately to the analysis and that no single feature dominates due to its scale or coding. The second dataset includes **Survival Minimum Expenditure Basket (SMEB)** components across different regions of Syria.

Data were collected from the Northwest Syria (NWS) and Northeast Syria (NES) datasets, including median prices at the community level. Since each region uses different currencies, exchange rates (TRY/USD and SYP/USD) were included to standardize costs into a single currency (USD). HSOS dataset were also collected in SYP, which we also converted to USD to deal with a single currency.

In the modeling part, after train test split, missing values were imputed using KNN Imputation (k=5) to preserve the underlying data distribution. Min-Max normalization was applied to the remaining numerical columns with values exceeding 1 to ensure consistency across features.

4.3 Dimension Reduction Techniques

Since working with too many columns reduces both the speed and efficiency of the model, we aim to reduce the number of columns in the next step. Although we have already removed unnecessary or redundant categorical columns, initially, the dataset contained 975 features. First visualizations were made with this dataset.

In the modeling part, after train test split, we applied a 20% missing value threshold to the features, the count was reduced to 789**.** At the modeling part further reduction to 497 features was achieved by removing low-variance features (threshold = 0.01).

5. Visualizations and Statistics

5.1 HSOS Dataset

5.1.1 Target Variable

The pie chart (Figure 1) visualizes the percentage distribution of general responses (as opposed to binary responses, where responses are text separated by commas) related to "Humanitarian assistance provided to IDP households in the community over the last 30 days."

The largest category, Food and Nutrition (28.4%), highlights its prominence in humanitarian assistance. The category NA - No humanitarian assistance reported (26.9%) represents cases where no aid was reported, underscoring gaps in support. Other categories, such as Health (15.8%), WASH (10%), and Cash assistance vouchers or cash in hand (8.4%), reflect varying priorities in assistance types. The Other (10.5%) category represents the sum of all categories below 5%.

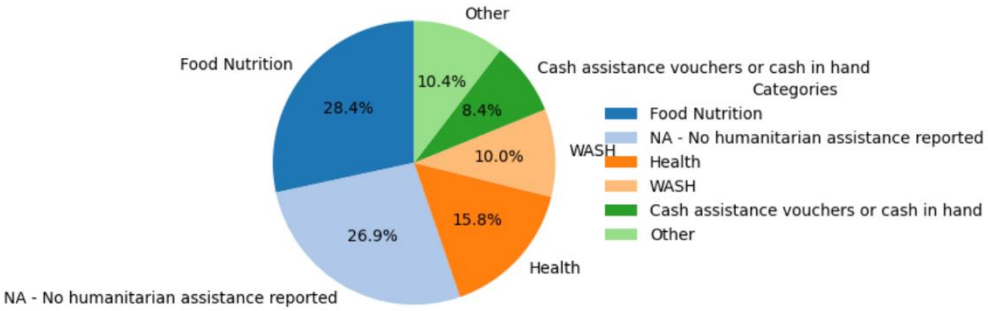

Figure 1: Percentage distribution for "Humanitarian assistance provided to IDP households in the community over the last 30 days"

Binary targets (Humanitarian assistance categories) showed no multicollinearity, confirming their independence (Figure 2).

Figure 2: Cramérs V Correlation Matrix - Target Variable

The stacked bar chart (Figure 3) illustrates the monthly percentage distribution of the types of humanitarian assistance provided to IDP households over the past 30 days. Binary target variables were grouped by month, and percentages were calculated to normalize the distribution for each month. The graph shows a consistent dominance of food and nutrition assistance across most months, while other categories, such as health and WASH, exhibit notable fluctuations. For example, WASH assistance spikes in certain months, likely due to seasonal needs or external factors

such as infrastructure challenges. These trends highlight the dynamic allocation of resources and the potential impact of contextual factors on aid prioritization.

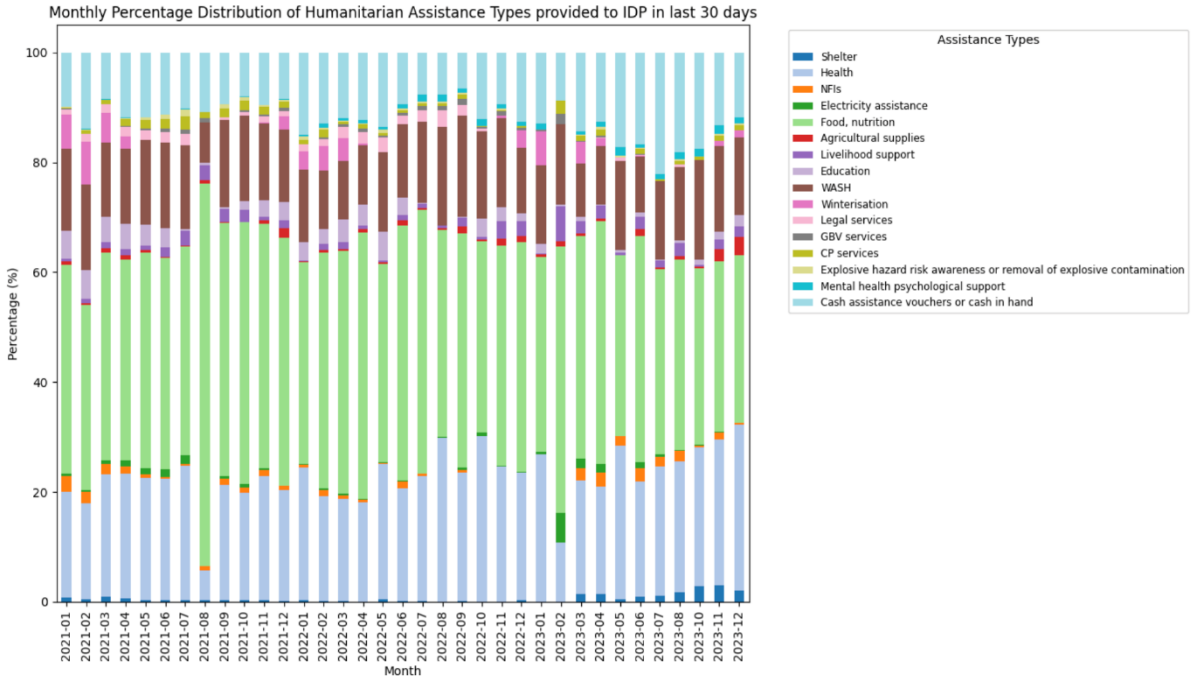

Figure 3: Monthly percentage distribution of types of humanitarian assistance provided to IDP households over the past 30 days

Time-series visualizations revealed trends in support across months, providing insights into seasonality and variation. As a result, it was decided to encode the months using One-Hot Encoding and include them as features in the model. The Augmented Dickey-Fuller test results indicated mixed stationarity across variables. While some variables, such as health and cash assistance vouchers or cash on hand, show stationarity with p-values well below the significance level, others, such as shelter and awareness of explosive hazards, fail to reject the null hypothesis, suggesting non-stationarity. Given the nature of the HSOS data, conducting time-series analyses over this short period is complex and was not pursued. Since our focus is on classification rather than prediction, the inclusion of month as a feature is considered sufficient to capture potential temporal patterns.

The value counts table clearly demonstrates that the dataset is unbalanced, with most assistance categories dominated by "0s" (indicating no assistance provided). This imbalance highlights the potential challenge of biased learning in machine learning models. Models may become overly sensitive to the majority class (0s) and fail to accurately predict the minority class (1s), which is often critical in humanitarian analysis. Algorithms that handle class imbalance well, such as gradient boosting or random forests, can be tuned to account for class weights. Additionally, applying oversampling techniques (e.g., SMOTE) to increase the representation of the minority class may be a viable solution. Finally, adjusting the classification threshold to optimize recall for the minority class can ensure that the model prioritizes the detection of critical instances where assistance is provided.

Category	Count of 0s	Count of 1s
Shelter	32683	229
Health	24959	7953
NFIs	32521	391
Electricity assistance	32672	240

Food, nutrition	18678	14234
Agricultural supplies	32665	247
Livelihood support	32396	516
Education	32014	898
WASH	27897	5015
Winterisation	32257	655
Legal services	32513	399
GBV services	32782	130
CP services	32567	345
Explosive hazard risk awareness or removal of explosives	32793	119
Mental health psychological support	32717	195
Cash assistance vouchers or cash in hand	28675	4237

Table 1: Distribution of Target Variables

5.1.2 Feature Variables

Missing value analysis reveals a significant proportion of missing data in the dataset, necessitating careful consideration of feature inclusion during modeling. For feature selection, an appropriate threshold for missing values should be determined to strike a balance between retaining valuable information and avoiding the noise introduced by sparse data. A preliminary threshold of 20% missing values appears reasonable, as it retains most of the dataset while excluding the most incomplete columns. However, this threshold can be revisited based on the specific requirements of the model and its sensitivity to missing data. It is recommended to apply the feature selection process after train-test splitting to avoid data loss. This approach ensures that the missing value threshold is applied independently to both the training and test sets, enhancing the model's robustness to unseen data.

Number Columns	Overall Columns	Percentage Columns	Missing Threshold
59	914	6.46 %	50 %
89	914	9.74 %	40 %
151	914	16.52 %	30 %
169	914	18.49 %	20 %
170	914	18.60 %	10 %

Table 2: Missing Value Analysis

5.1.2.1 Demographic Features

The top pie chart (Figure 4) illustrates the percentage distribution of the assessed population across governorates. Idleb and Aleppo dominate, with similar proportions (31.3% and 30.8%, respectively), while Deir-ez-Zor and other smaller regions contribute minimally to the dataset.

The second pie chart (Figure 5) shows the distribution across districts, with a significant proportion (36.1%) categorized as "Other" (the sum of categories below 5%), followed by Ar-Raqqa (10.9%)and Harim (9.7%). This highlights the geographic spread and variability in representation at the district level, which includes 23 unique values.

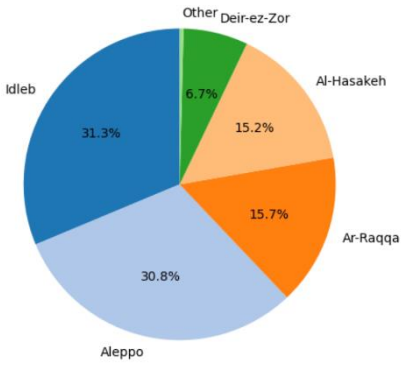

Figure 4: Percentage distribution across governorates

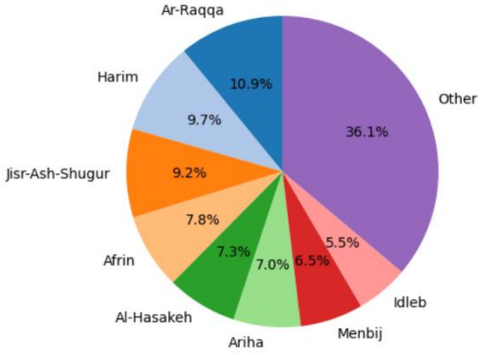

Figure 5: Percentage distribution across districts

5.1.2.2 Important Features

To analyze priority needs, three questions regarding the most critical needs (Top 1, Top 2, Top 3) were aggregated with weighted values of 3, 2, and 1, respectively. The data was then reshaped using a melt operation and aggregated on a monthly basis to observe trends over time. From the resulting dataset, the top five priority needs, Food, Health, Livelihoods, Shelter, and WASH were identified and plotted as a time series.

The graph (Figure 6) illustrates the weighted monthly percentage distribution of the top five needs over time. Food consistently appears as the dominant priority, followed by Livelihoods and Shelter, indicating that these are the most critical areas of need. Health and WASH display significant but smaller shares, with varying levels of importance depending on the month. The identified priority needs are expected to have a strong relationship with the target variable, as addressing these needs directly correlates with improving humanitarian outcomes for the IDP population.

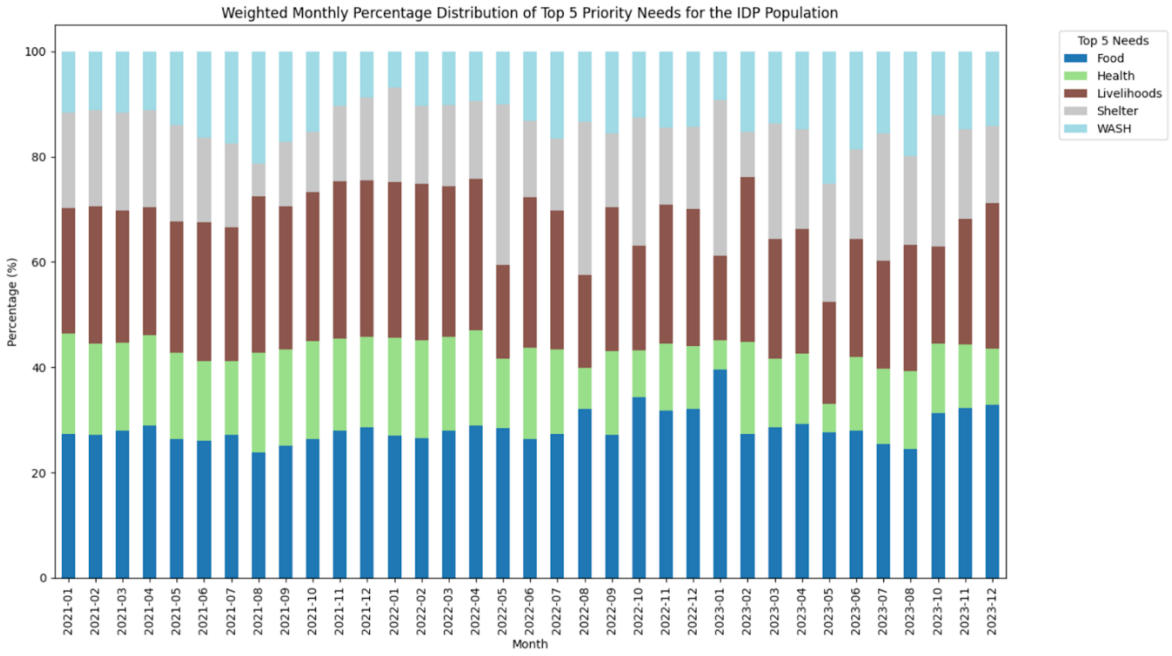

Figure 6: Weighted monthly percentage distribution of the top 5 Priority Needs for the IDP Population?

The following graph (Figure 7) highlights the main challenges faced by IDPs in accessing humanitarian assistance. The "NA - No humanitarian assistance reported" category (22.6%) indicates cases where no assistance was provided, and thus no challenges were reported.

Other challenges include insufficient quantity (20.6%), types of assistance not relevant (15.8%), and not meeting eligibility criteria (8.1%). These factors could significantly impact the distribution and accessibility of aid, and may be related to how well aid is tailored to the specific needs of the community.

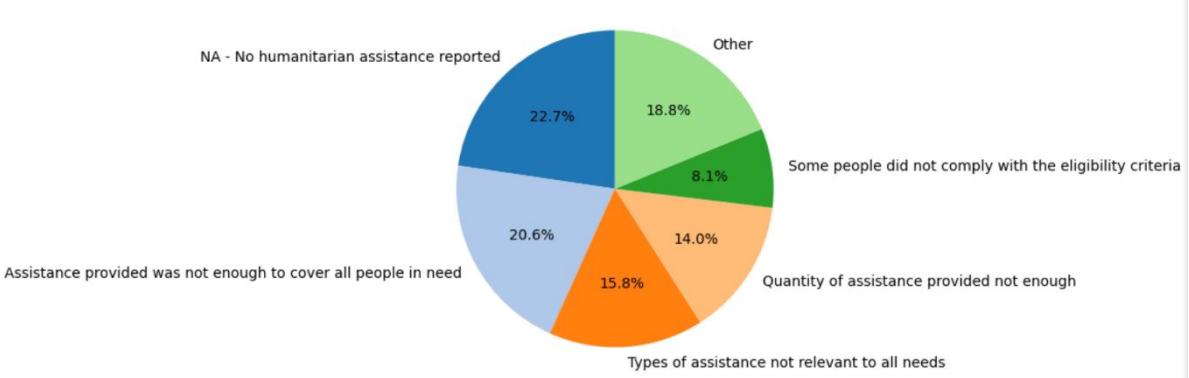

Figure 7: Percentage Distribution - Most important missing information identified by households?

The pie chart (Figure 8) illustrates the preferred methods of providing information to households in the location. "Personal face-to-face" is the preferred method at 32.4%, followed by "WhatsApp or other mobile phone-based platforms" at 28.5%.

This distribution highlights the importance of direct and personal communication channels in humanitarian efforts. The target variable may be influenced by the effectiveness of these communication methods in disseminating critical information, potentially impacting access to aid.

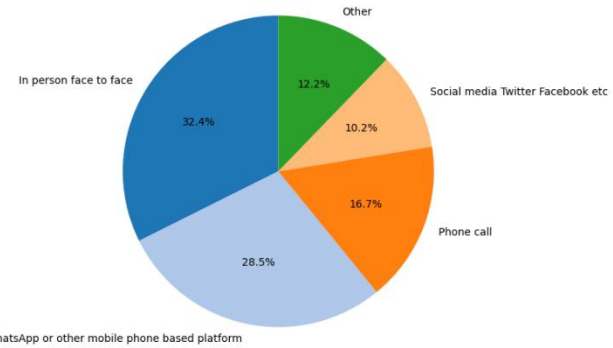

Figure 8: Percentage Distribution - Are households in the location aware of a humanitarian assistance feedback of complaints mechanism?

5.1.3 Relationship between target variable and features

Cramer's chi-square test was performed to account for non-linear relationships. The number of significant features (p-value < 0.05) and insignificant features, based on the test statistics, are shown in the following table:

Humanitarian assistance Categories to IDPs	Important Feature Relations	Unimportant Feature Relations
Shelter	374	540
Health	687	227
NFIs	394	520
Electricity assistance	455	459
Food, nutrition	674	240
Agricultural supplies	286	628
Livelihood support	448	466
Education	548	366
WASH	664	250
Winterisation	490	424
Legal services	516	398
GBV services	353	561
CP services	479	435
Explosive hazard risk	309	605
Mental health psychol	410	504
Cash assistance vouch	605	309

Table 3: Relationship between Target and Features

Analysis of the characteristics of the humanitarian assistance categories provided to IDPs reveals varying degrees of importance and irrelevance in terms of their predictive power.

For example, categories such as Shelter and Food/Nutrition show a higher number of important feature relationships, suggesting that these areas have strong associations with other variables in the dataset. In contrast, categories such as Agricultural Supplies and Explosive Hazard Risk Awareness have a relatively higher proportion of unimportant feature relationships, indicating limited or indirect connections with other explanatory variables.

The pattern underscores the complexity of humanitarian data, where some categories are more robustly related to contextual variables (e.g., economic conditions or shelter dynamics), while others are likely more situational or region-specific.

Chi-square analysis further supports these findings, as high chi-square values in some categories indicate statistically significant relationships between variables, while others show negligible influence. This duality highlights the importance of targeted modeling approaches. For instance, reducing noise from irrelevant features for less

connected categories and emphasizing key variables for categories with stronger associations is likely to improve model performance.

5.2 Market Dataset

The Joint Market Monitoring Initiative (JMMI) data focuses on the Survival Minimum Expenditure Basket (SMEB) across different regions in Syria. The analysis examines price trends, regional variations, and relationships between SMEB components.

The JMMI was developed by the Cash Working Group (CWG) to understand market functionality and challenges in Syria. It monitors prices and stock levels of basic commodities monthly to inform cash-based responses and food assistance programs.

The analysis focuses on SMEB components selected by the CWG, which represent the minimum culturally-adjusted items needed by an average six-person household in Syria per month. Data collection spans multiple governorates, with regular monitoring of selected retailers in major markets.

In this section, the SMEB cost (total) and its components—food, non-food items, cooking fuels, water, and mobile data—are analyzed. To avoid multicollinearity, only the subtotals of the components are planned to be added to the model.

Figure 9 already shows that food dominates the SMEB calculations. Over 80% of the component analysis is food, followed by non-food items, cooking fuels, and water.

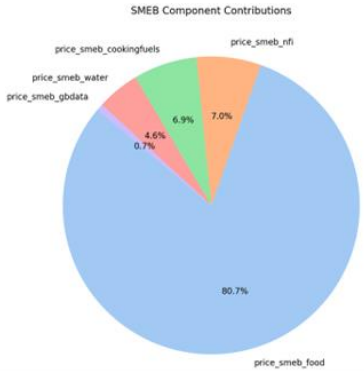

Figure 9: Overall Distribution of SMEB Components

The graph (Figure 10) illustrates SMEB cost and component trends over time for the Northeast and Northwest regions.

The time series analysis shows a similar pattern. Food has the largest share and a rising trend until mid-2023, when food prices drop abruptly. The other categories show a smoother trend with less fluctuation. The exchange rate used here for Syrian Pound/USD is not a floating rate but a fixed one.

The sharp drop in July 2023, with an 80% devaluation, is the primary reason for this significant change. While the Northeast shows significant volatility with a sharp decline in mid-2023 due to currency devaluation, the Northwest exhibits more stability, with SMEB costs remaining relatively consistent over the observed period.

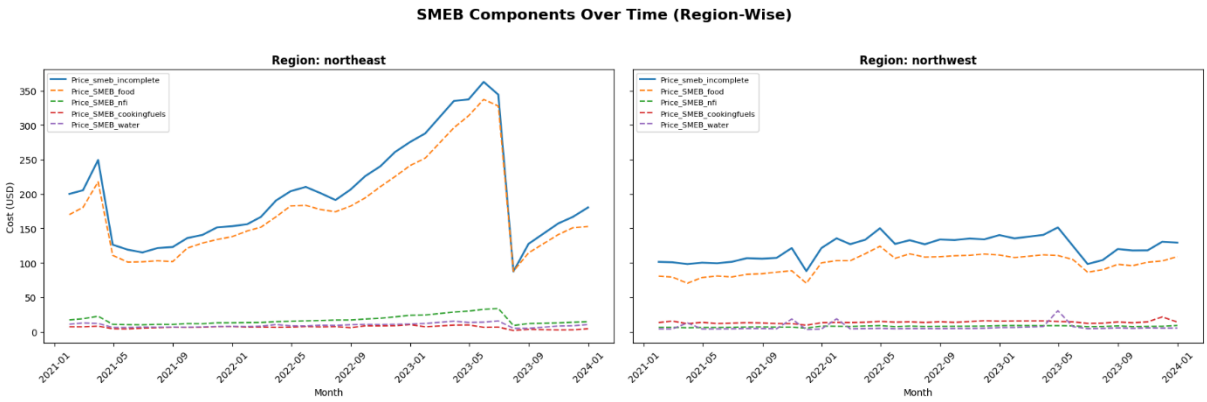

Figure 10: Overall Distribution of SMEB Components

The correlation matrix shows that only food and non-food items have a high positive correlation. This could indicate multicollinearity, which is a problem for our future model estimation. VIF results also indicated multicollinearity.

A possible solution during the modeling step could be to exclude the NFI values. In high-dimensional datasets, multicollinearity can lead to overfitting, where the model captures redundant information from correlated features.

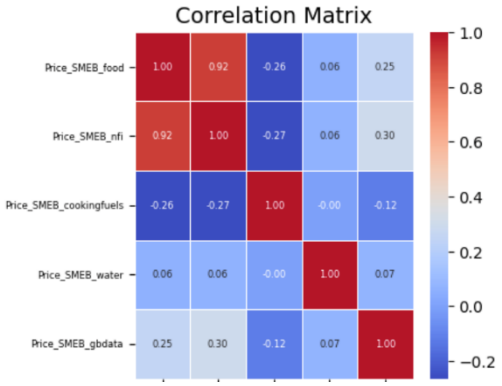

Figure 11: Correlation Matrix of SMEB Components

Variables such as price_smeb_food, price_smeb_nfi exhibit high VIF values, indicating significant multicollinearity.This suggests potential redundancy in these variables for regression models. price_smeb_food and price_smeb_nfi have a strong correlation correlation of 0.92. For this reason, nfi prices are excluded from dataset to avoid overfitting.

6. Final Models

6.1. One Target Approach

The primary concept that emerged was the formulation of a single action-no-action target variable, derived from the 16 target categories. To accomplish this objective, two methodologies were considered. The initial approach entailed the creation of a variable that would assume a value of 1 if any of the 16 targets equaled 1, and 0 otherwise. In the created target variable, 41.7% (13,720 obs.) of the 32,912 observations are ones and 58.3% (19,192 obs.) are zeroes.

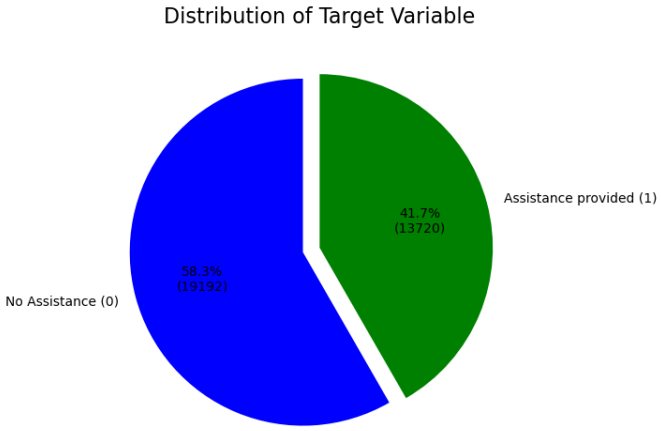

Figure 12: OneTarget - Distribution

6.1.1 Defining Baseline Models

Following the completion of the train test (with a test size of 0.2 and a random state of 1234), the decision was taken to proceed with the evaluation of four baseline models. Thereafter, the most promising results will be refined to identify the optimal model.

6.1.1.1 Logistic Classifier

A logistic classifier was selected as the initial model due to its rapid and straightforward interpretation. These classifiers are most effective when the data can be divided linearly. The model was trained initially with 500 iterations, a maximum solver setting, in order to ensure convergence and avoid converging to an undesired solution.

6.1.1.2 Random Forest

Our data is tabular and in binary coding. To catch non-linear relationships, we deployed a Random Forest (RF) model with 100 estimators and a random state of 42.

6.1.1.3 XGBoost

The XGBoost classifier is a powerful and fast model. It also captures non-linear relationships and returns the importance of features, which can be used to eliminate unimportant features. This is a great asset to reduce overfitting and improve model performance by eliminating unnecessary features.

6.1.1.4 Support Vector Machine

The Support Vector Machine (SVM) demonstrates a high degree of proficiency in identifying complex classifications between classes. It provides stable differentiation in small to medium-sized datasets. The radial basis function (RBF) kernel enables the application of SVM to non-linear problems. Despite its effectiveness as a model, SVM exhibits suboptimal performance on larger datasets. The calculations required for the model took approximately half an hour to complete on our dataset.

6.1.1.5 Stochastic Gradient Descent Classifier

The present classifier implements a regularized linear model with stochastic gradient descent (SGD) learning. The SGD Classifier is, by default, a linear support vector machine (SVM) optimized for large datasets (scikit-learn.org, 2025b). The model's chosen loss function was a logistic loss function, with a random state of 42.

Logistic Classifier				
	precision	recall	f1-score	support
0	0.88	0.95	0.91	2,739
1	0.96	0.91	0.93	3,844
accuracy			0.92	6,583
Random Forest				
0	0.92	0.95	0.94	2,739
1	0.96	0.94	0.95	3,844
accuracy			0.95	6,583
XGBoost				
0	0.92	0.97	0.94	2,739
1	0.98	0.94	0.96	3,844
accuracy			0.95	6,583
SVM				
0	0.90	0.96	0.93	2,739
1	0.97	0.93	0.95	3,844
accuracy			0.95	6,583
SGD				
0	0.91	0.89	0.90	2,739
1	0.93	0.94	0.93	3,844
accuracy			0.92	6,583

Table 4: OneTarget - All Results

Intuitively, precision is defined as the ability of the classifier to avoid labeling a negative sample as positive, while recall is defined as the classifier's ability to identify all positive samples. The F1-measure can be interpreted as a weighted harmonic mean of the precision and recall. Accuracy computes the count (or fraction) of correct predictions out of all predictions (scikit-learn.org, 2025a). As illustrated in Table 4, logistic regression accuracy is 0.9246, indicating that a mere 8% of the predictions fall outside the categories of true positives or true negatives. To assess the model's overfitting, the cross-validation function was employed with five iterations. The resulting cross-validation accuracy of 0.9200 closely mirrors the initial value, indicating that the model is not overfitting. The random forest performance overall is better than the logistic classification. The precision, recall, f1-score and accuracy improved. The XGBoost results show great performance. There is a slight improvement in Precision for 1 and Recall for 0.

The SVM demonstrates a performance that is not as proficient as that of the XGBoost model, yet it exhibits a level of proficiency that is analogous to that of the Random Forest model. Among these three models, the SVM exhibits the highest number of False Negatives. However, it demonstrates a superior performance in terms of False Positives when compared with the Random Forest model. The XGBoost and RF models both exhibit higher F1 rates, yet their levels of accuracy are approximately equivalent.

The SGD demonstrates commendable performance; however, it does not reach the level of proficiency exhibited by RF, XGBoost, and SVM. The recall value for 0 is less than 0.90, and the f1 score is also inferior to that of the previous models.

6.1.2 Model Optimization

6.1.2.1 GridSearch & BayesSearch XGBoost

In the context of the baseline test, XGBoost was identified as the most effective model. It is acknowledged that XGBoost is dependent on hyperparameter tuning for its optimization. To enhance the model's performance, the tuning process was executed using GridSearch and the intelligent optimizer, BayesSearch. The GridSearch approach yielded the optimal results, as outlined below: The optimal configuration parameters were determined to be 'colsample_bytree': 1, 'learning_rate': 0.2, 'max_depth': 9, 'n_estimators': 300, and 'subsample': 1. This configuration yielded an accuracy of 0.9530, representing a marginal enhancement over the baseline. The F1-Score reached 0.9603.

BayesSearch identified the feature sample per tree as 0.6, the learning rate as 0.1056, the maximum depth as 8, the number of estimators as 500, and a subsample of 1.0. The accuracy was 0.9526, indicating a negligible enhancement.

6.1.2.2 XGBoost Feature Importance & Feature Removal

XGBoost shows the most influential features on its classification. Following the execution of all potential thresholds, it was determined that 0.000877577 emerged as the optimal threshold. This resulted in the retention of 267 features within the dataset, yielding an F1-score of 0.9624.

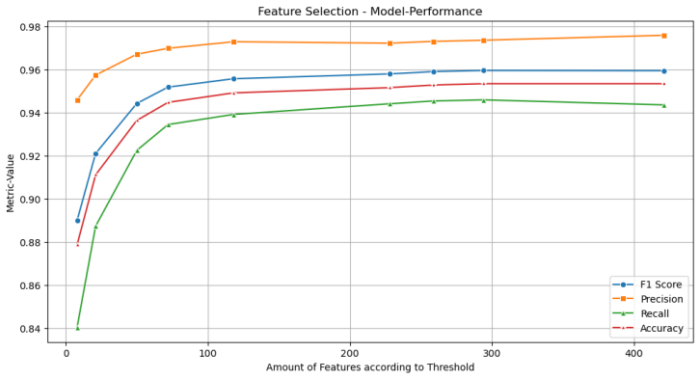

Figure 13: OneTarget - Feature Removal Threshold

6.1.2.3 Voting Classifier XGBoost + Random Forest + Logistic Classifier

In the preliminary evaluation, XGBoost and Random Forest exhibited notable outcomes, prompting the exploration of a combined approach through a Voting Classifier to further enhance performance. This investigation involved the implementation of a Logistic Classifier as a baseline model. The initial trial was executed using a soft voting method, while the subsequent trial employed a hard voting method. The first run yielded an accuracy of 0.9535 and an F1-Score of 0.9594, while the second run attained an accuracy of 0.9504 and an F1-Score of 0.9568. The results obtained from the combination of RF and XGBoost through a Voting Classifier were not significantly different from those of XGBoost alone.

6.1.3 Interpretability

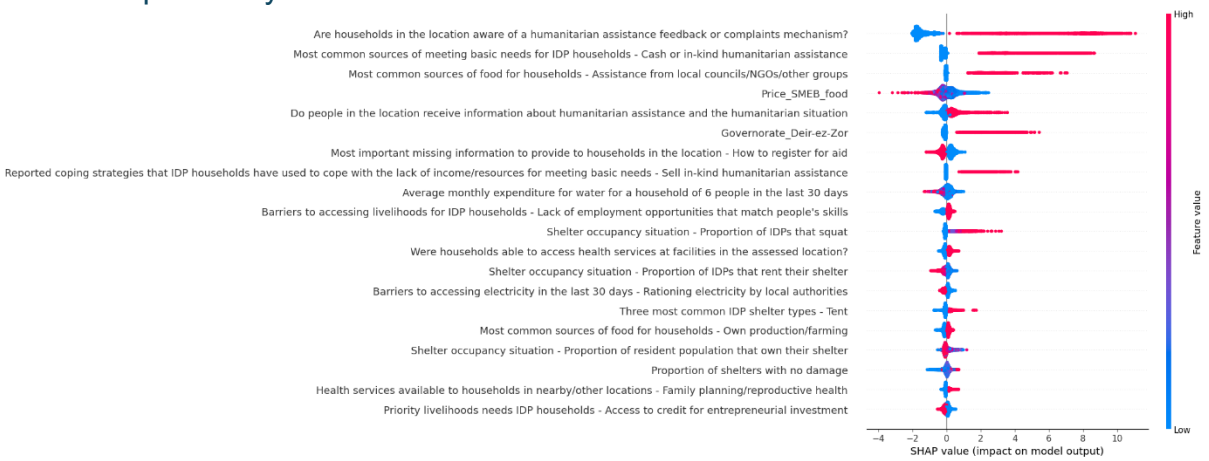

Figure 14: OneTarget - XGBoost SHAP

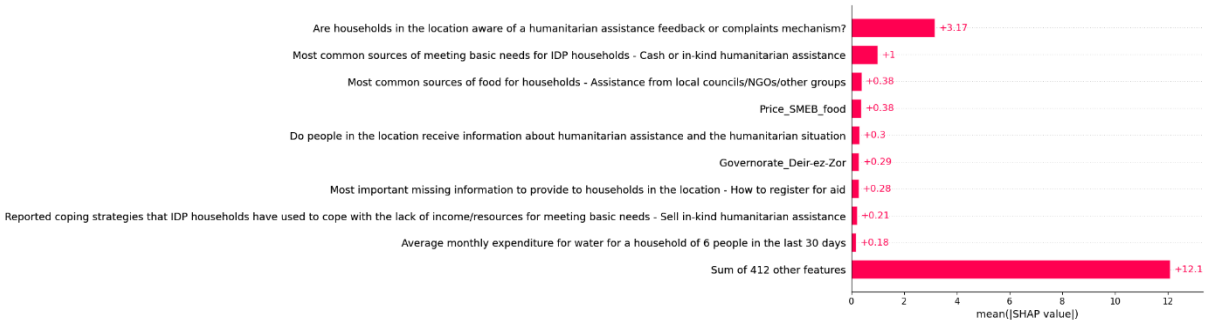

Figure 15: OneTarget - XGBoost Shap Important Features

The SHAP summary plot is a graphical representation of the impact of key features on the XGBoost model's predictions. Features are ranked by importance, with the most influential features at the top. Each data point represents a SHAP value, indicating the extent to which a feature contributes to a higher or lower prediction outcome. Red points indicate high feature values, while blue points represent low values. If high values (red) are predominantly located to the right, the feature enhances the prediction probability. Conversely, features positioned to the left of the plot contribute to a reduction in prediction probability. The analysis identifies awareness of humanitarian assistance, food sources, and economic conditions as the most critical factors. While certain features exhibit a unidirectional impact, others demonstrate a more multifaceted effect.

6.1.4 Deep Learning

The initial model evaluated was a sequential dense neural network (DNN) with a binary crossentropy loss function and a sigmoid activation function. After 30 epochs, the model demonstrated an accuracy of 0.9966, with a validation accuracy of 0.9368.

These results suggest that the model is overfitting, indicating that it has overtrained on the training set.

In order to address the issue of overfitting, a more complex model was created, incorporating tuning dropout layers and batch normalization layers. Additionally, the Adam optimizer was enabled to select the learning rate from a range of 0.001, 0.0005, and 0.0001. Along with the more complex model, the hyperparameters were tuned using a 10-fold random search from the KerasTuner package. The optimal parameters were then employed in a model comprising 30 epochs with the lr_scheduler callback (monitor='val_loss', factor=0.5, patience=3). The callback was activated eight times in epochs 9, 12, 15, 18, 21, 24, 27, and 30. The final accuracy achieved was 0.9947 and 0.9373 on the validation set.

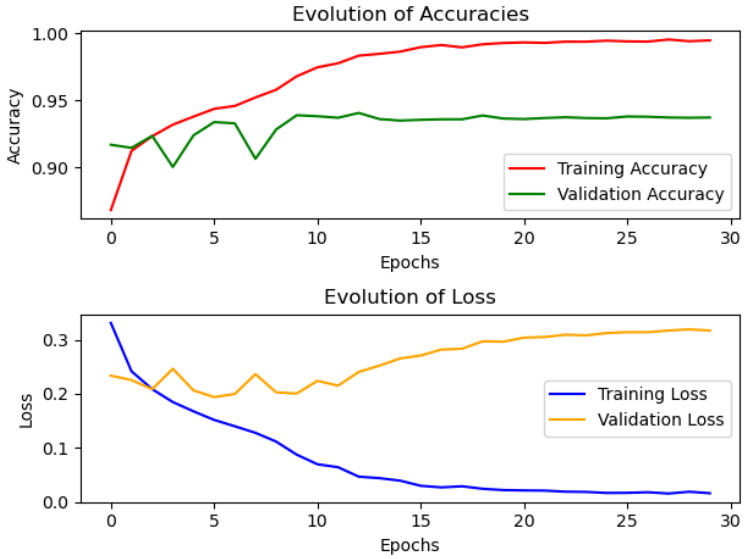

Figure 16: OneTarget - DL Improved Model

Notwithstanding the remarkably high accuracy observed in the training set, the validation accuracy remains distinctly lower, thereby suggesting the occurrence of overfitting. In contrast, machine learning approaches, particularly XGBoost, exhibited notably robust performance without encountering this issue.

6.2. Multi Label Approach

6.2.1 Classification of the Problem

Machine Learning Type: This part takes into consideration models into MultiOutputClassifier to handle a multi-label classification problem, where each instance may belong to multiple categories simultaneously. The model is designed to predict different forms of assistance provided in a humanitarian setting, categorizing each instance into one or more relevant aid types.

6.2.1.1 Performance Metrics:

The Hamming Loss directly measures the proportion of misclassified labels in multi-label classification. F1 Score (Macro & Micro) evaluates the balance between precision and recall across all labels. The Macro F1-Score calculates the F1-Score for each individual label and then takes the unweighted average across all labels. The Micro F1-Score first calculates the sum of True Positives (TP), False Positives (FP) and False Negatives (FN) across all labels and then calculates the F1-Score from these totals.

$$F1_{micro} = 2 * \frac{Total\ Precision * Total\ Recall}{Total\ Precision + Total\ Recall}$$

Accuracy Provides an overall measure of the correctness of the model but is not the primary focus due to the multi-label nature of the model.

6.2.2 Defining Baseline Models

We evaluated multiple models to establish baseline performance and identify the most effective approach:

- Logistic Regression (MultiOutputClassifier): Baseline linear classifier, interpretable but not optimal for complex interactions.
- Random Forest (MultiOutputClassifier): Robust to non-linearity and missing data, performed moderately well.
- SVM (MultiOutputClassifier): Strong performance but computationally expensive for large datasets.
- Naive Bayes: Discarded due to its assumption of feature independence, which does not hold in this dataset.
- Decision Tree (MultiOutputClassifier): Performed poorly according to Hamming Loss due to high variance. As a single tree model, it lacks the generalization power of ensemble methods like Random Forest and XGBoost, which combine multiple trees to improve robustness and reduce overfitting.
- KNN (MultiOutputClassifier): KNN was excluded due to its sensitivity to irrelevant features and computational inefficiency with high-dimensional data. Despite reasonable performance, it struggles with scalability, making it less suitable for our dataset with high number of features. Additionally, KNN tends to perform poorly when features are highly imbalanced.

Model	Hamming Loss	F1-Score (Macro)	F1-Score (Micro)	Accuracy
Logistic Regression	0.0293	0.50	0.77	0.6922
Random Forest	0.0216	0.44	0.81	0.7597
Naives Bayes	0.1007	0.31	0.50	0.3840
SVM	0.0304	0.49	0.77	0.6859
KNN	0.0261	0.54	0.80	0.6913
Decision Tree	0.0310	0.49	0.77	0.6868

Table 5: MultiTarget - Base Models

Based on the evaluation metrics, SVM and Random Forest show solid performance, especially in terms of accuracy and micro-F1 score. However, XGBoost consistently outperforms them in most metrics, making it a strong candidate for inclusion in the final model selection. Moving forward, we will continue with SVM, Random Forest, and XGBoost, leveraging the robustness of XGBoost to further improve predictive performance in the next phase.

6.2.3 Model Optimization

6.2.3.1 Hyperparameter Tuning with GridSearchCV

To optimize model performance, GridSearchCV (3-fold cross-validation) was applied to Random Forest, SVM, and XGBoost. The best hyperparameters obtained were:

- Random Forest: max_depth=20, min_samples_split=2, n_estimators=200
- SVM: C=10, kernel='rbf'
- XGBoost: learning_rate=0.1, max_depth=10, n_estimators=200

Model	Hamming Loss	F1-Score (Macro)	F1-Score (Micro)	Accuracy
Random Forest	0.0218	0.456789	0.825133	0.758013
SVM	0.0202	0.592332	0.846237	0.771229
XGBoost	0.0195	0.599793	0.850648	0.783381

Table 6: MultiTarget - Hyperparameter Tuning Models

Based on the results, XGBoost and SVM demonstrated superior performance, leading to their selection for ensemble methods.

6.2.3.2 Ensemble Models (Stacking & Voting)

6.2.3.2.1 Stacking Classifier

We chose the base models that performed best: XGBoost and SVM. We trained these models separately for each label. We selected the meta model Logistic Regression because it can generalize well.

We implemented the separate stacking classifiers for each label using StackingClassifier. This ensured that each label had its own specialized ensemble.

6.2.3.2.2 Voting Classifier (Hard & Soft Voting)

Hard voting uses the most common label among models. Soft voting uses averaged probabilities for the final decision.

Model	Hamming Loss	F1-Score (Macro)	F1-Score (Micro)	Accuracy
Stacking	0.019311	0.587003	0.852073	0.781710
Hard Voting	0.020052	0.562929	0.841465	0.773204
Soft Voting	0.01914	0.596515	0.852890	0.783837

Table 7: MultiTarget - Stacking & Voting Classifier

Soft Voting performed better than both individual models and Hard Voting. Stacking was slightly more accurate than Soft Voting but had a lower F1-score. However, Soft Voting did not perform significantly better than the other models, meaning it did not contribute much to the overall predictive performance. Therefore, instead of focusing on ensemble voting, we should analyze feature importance and model interpretability, particularly within XGBoost, which provides built-in feature importance metrics.

6.2.3.3 Bagging and Boosting Multi-Label RF and SVM

This section focuses on Random Forest (RF) and Support Vector Machine (SVM) models, which performed best when used with the MultiOutputClassifier. The goal was to apply Bagging and Boosting techniques in various ways: separately, combined (bagged then boosted), and using Bayes search for hyperparameter optimization. Bayesian Optimization was applied to fine-tune hyperparameters. Search spaces included:

- **AdaBoost:** n_estimators (10–100), learning_rate (0.01–2.0)
- **Bagging**: n_estimators (10–100), max_samples (0.5–1.0)
- **RF**: n_estimators (50–200), max_depth (5–20)
- **SVM**: C (0.1–10.0), max_iter ([1000, 3000, 5000, 7000]

Model	Hamming Loss	F1-Score (Macro)	F1-Score (Micro)	Accuracy
Bagging RF	0.02224	0.43	0.82	0.75375
Boosting RF	0.02259	0.48	0.82	0.74130
Bagging SVM	0.04062	0.50	0.75	0.61582
Boosting SVM	0.07123	0.39	0.62	0.49840
Boosting of bagging RF	0.02225	0.44	0.82	0.75148
Boosting of bagging SVM	0.07052	0.39	0.61	0.50539
RF with Bayes Search	0.06362	0.44	0.65	0.53729
SVM with Bayes Search	0.07423	0.37	0.69	0.49017

Table 8: MultiTarget – Bagging Boosting Results

In all the tests, the SVM always did better than RF, especially in terms of Hamming loss and recall. Ensemble techniques like bagging and boosting often make models better, but in this specific multi-label classification situation, they didn't help much. Using boosting with bagged models for both RF and SVM didn't improve things much. Trying to find the best hyperparameters for both models by using a Bayesian search and reducing the combination set to minimize computation also didn't improve things much.

The SVM model was the most effective approach because of its inherent strength, and it did not require additional ensemble complexity.

6.2.3.4 SMOTE-Enhanced Models (XGBoost, SVM, Random Forest)

To address the issue of class imbalance in multi-label classification, MLSMOTE was applied, following the methodology from the thesis "Handling Data Imbalance in Multi-label Classification" by Sukhwani (2020). Traditional SMOTE is not directly compatible with multi-label problems, as it assumes binary or multi-class settings. Instead, MLSMOTE was used to oversample minority labels, ensuring that each underrepresented label reached the median frequency of 744.

This approach was necessary to avoid overfitting to dominant labels while improving the recall of minority classes. The SMOTE-enhanced models were evaluated, and the results are compared below.

Model	Hamming Loss	F1-Score (Macro)	F1-Score (Micro)	Accuracy
Random Forest SMOTE	0.02293	0.44	0.81	0.74479
SVM SMOTE	0.02047	0.59	0.84	0.76910
XGBoost SMOTE	0.02106	0.59	0.84	0.76530

Table 9: MultiTarget - MLSMOTE Results

XGBoost and SVM consistently outperform Random Forest across all major evaluation metrics. Among them, SVM achieves the highest Exact Match Accuracy (76.91%), narrowly surpassing XGBoost. However, XGBoost demonstrates a better balance between precision and recall, effectively mitigating overfitting while maintaining strong predictive performance. In contrast, Random Forest struggles significantly with minority classes, particularly in GBV Services, Explosive Hazard Risk Awareness, and Agricultural Supplies, where both precision and recall remain notably low. Additionally, the macro F1-score is substantially lower than the micro F1-score across all models, highlighting the ongoing challenge of accurately predicting underrepresented labels in this multi-label classification setting.

SMOTE slightly improved the micro F1-score, indicating better performance across all labels, especially for minority classes. However, the drop in macro F1-score suggests that predicting minority classes remains a challenge. Additionally, the slight increase in Hamming loss indicates more label-level misclassifications. Despite applying SMOTE, XGBoost and SVM continue to be the best-performing models. Given that overall accuracy decreased slightly and the improvements in minority classes were minimal, it seems more practical to proceed with the original models combined with feature selection rather than relying on SMOTE.

6.2.4 Feature importance analysis using XGBoost

Several criteria can be considered to determine a threshold based on feature importance scores: Natural Breakpoint (Elbow Method) analysis showed us a decrease of around 0.01 in the graph. The few most important features (e.g. top 10-20) have significantly higher importance scores. Below 0.005, the decline becomes less pronounced, so it may be less effective from here on.

Then we looked a narrow selection, that take features at ‘0.01 and above’. Then, for a wider but balanced selection, it makes sense to take ‘0.005 and above’ or ‘0.001 and above’.

XGBoost Threshold	Selected Features	Hamming Loss	F1-Score (Macro)	F1-Score (Micro)	Accuracy
0.001	340	0.019615	0.58	0.84	0.780799
0.005	33	0.035024	0.41	0.72	0.674009
0.01	6	0.049484	0.08	0.54	0.615069

Table 10: MultiTarget - Feature Selection & Performance

The table presents the impact of different feature importance thresholds on model performance. As the threshold increases, the number of selected features decreases significantly—from 340 at 0.001 to just 6 at 0.01.

From the results, we observe that selecting only the most important features (higher thresholds) leads to a sharp decline in performance across all key metrics (Hamming Loss, F1-Score, and Accuracy). The model with 340 selected features (Threshold: 0.001) achieves the best balance, maintaining high accuracy (78.08%) and strong F1 scores (Macro: 0.58, Micro: 0.85) while keeping Hamming Loss low.

Given this analysis, continuing with the 340 selected features is the optimal approach, as it retains strong predictive power while reducing dimensionality compared to using all features.

6.2.5 Interpretability

6.2.5.1 Error Analysis

Since this is multi-label classification, we can generate label-wise confusion matrices rather than a standard one. For each label, we generated a confusion matrix to compare true vs. predicted values. Then, we visualized it using a heatmap to easily spot misclassifications.

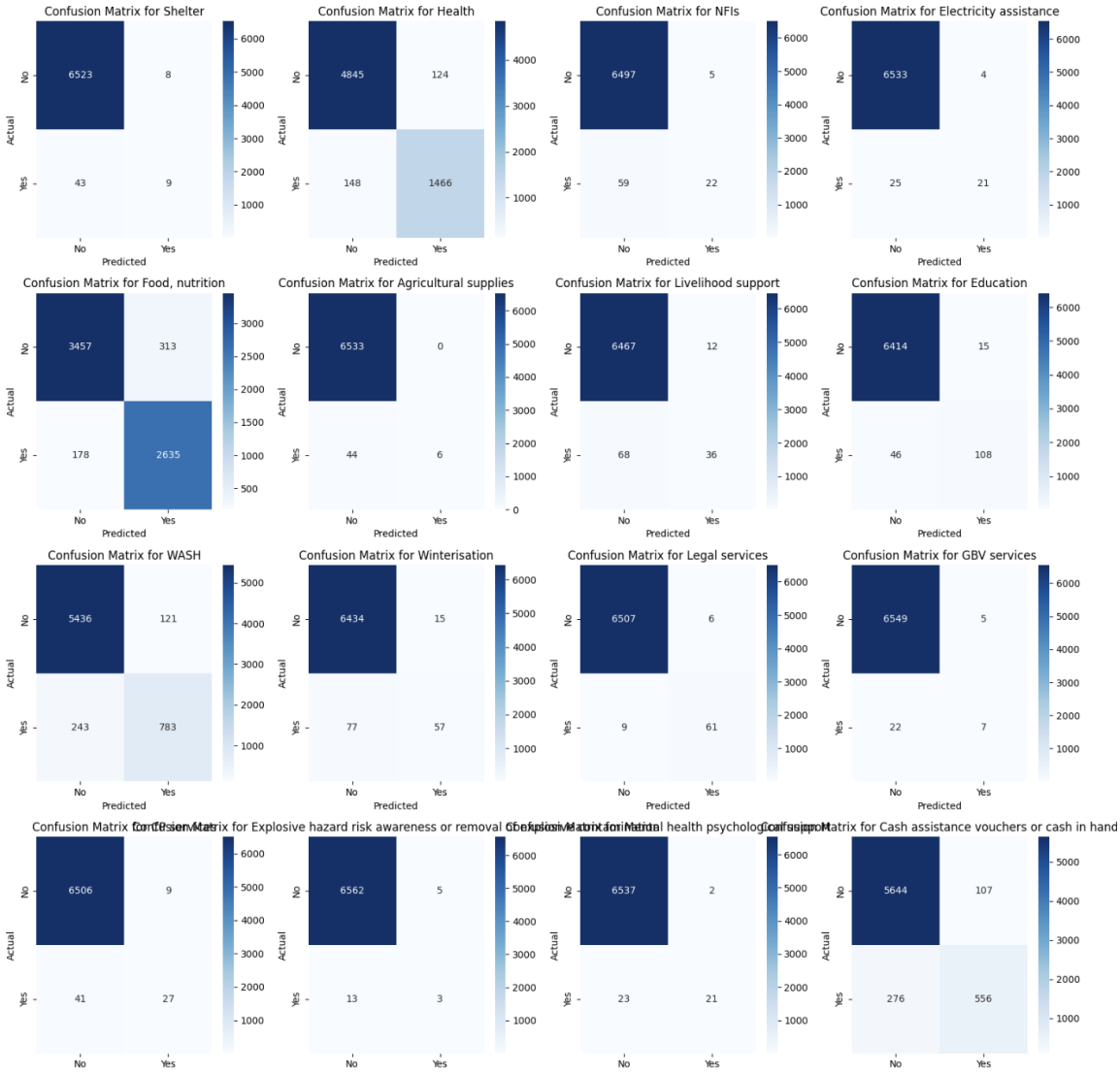

Figure 17: MultiTarget - Confusion Matrix Heatmap

The confusion matrices reveal key insights into the model's performance across different labels. High-imbalance issues are evident in categories like Agricultural Supplies, Explosive Hazard Risk Awareness, and Mental Health Psychological Support, where very few positive cases were correctly predicted, indicating poor recall. In contrast, labels like Food, Nutrition, WASH, and Cash Assistance Vouchers show better performance, with a higher number of true positives and relatively fewer false negatives. However, some labels, such as Education and Livelihood Support, exhibit moderate false negatives, suggesting that the model struggles to correctly classify minority classes in certain categories.

Overall, the model performs well on frequently occurring labels but still struggles with rare ones, reinforcing the need for further optimization, such as threshold tuning, or alternative loss functions tailored for imbalanced multi-label classification.

6.2.5.2 Per-Class Threshold Optimization for XGBoost (Threshold: 0.001)

```
XGBoost (Per-Class Thresholds) Results:
Hamming Loss: 0.019538963998177124
F1-Score (Macro): 0.635697740089314
F1-Score (Micro): 0.8540425531914894
Accuracy: 0.7792799635424579

Optimized Per-Class Thresholds:
{'Shelter': 0.39999999999999997, 'Health': 0.39999999999999997, 'NFIs': 0.3, 'Electricity assistance': 0.3, 'Food, nutrition': 0.4999999999
9999994, 'Agricultural supplies': 0.3, 'Livelihood support': 0.3, 'Education': 0.39999999999999997, 'WASH': 0.3, 'Winterisation': 0.3, 'Leg
al services': 0.3, 'GBV services': 0.3, 'CP services': 0.35, 'Explosive hazard risk awareness or removal of explosive contamination': 0.3,
'Mental health psychological support': 0.3, 'Cash assistance vouchers or cash in hand': 0.35}
```

Figure 18: MultiTarget - Per-Class Threshold Optimization for XGBoost

The per-class threshold tuning has notably improved performance, especially for macro F1-score and micro F1-score, while maintaining accuracy at a similar level.Some common labels (e.g., Food, nutrition, Cash assistance, CP services) have higher thresholds (0.35 - 0.5). This prevents over-prediction and reduces false positives.

Some minority labels (e.g., GBV services, Electricity assistance, NFIs, WASH) have lower thresholds (0.3 - 0.35). This improves recall, helping to detect rare events better. In conclusion, the macro F1-score improved significantly (from 0.5809 to 0.6357), suggesting better performance across all labels, including minority classes.Adjusting thresholds helped balance precision and recall for underrepresented labels.

The micro F1-score increased slightly (from 0.8492 to 0.8540), showing that the overall model got better without losing general performance.Accuracy stayed about the same (from 0.7808 to 0.7793), which indicates that the per-class thresholding did not lead to more incorrect predictions at the sample level.

Hamming Loss, which measures the number of incorrect predictions, slightly improved from 0.0196 to 0.0195, suggesting fewer individual label misclassifications.

6.2.5.3 SHAP analysis using XGBoost

Chen (2021) demonstrates that SHAP (SHapley Additive exPlanations) offers intuitive and comprehensive model interpretability through three key benefits. It provides global interpretability by quantifying each feature's overall impact on predictions using Shapley values, accounting for feature interactions and correlations. SHAP also enables local interpretability, assigning specific contribution values to individual predictions, enhancing transparency beyond traditional global metrics. Additionally, its model-agnostic framework allows SHAP values to be applied across various machine learning models, unlike other methods limited to simpler surrogate models.

The SHAP summary plot visualizes the impact of features on the model’s predictions. Even if this is a multi-label classification problem with 16 targets, the SHAP values shown in the plot belong to a selected target namely Shelter as an example.

Features with a wide spread of SHAP values (large variance) have a stronger effect on predictions. If a feature has a mostly positive impact, it increases the likelihood of predicting a target label as "Yes" (1). If a feature’s SHAP values are centered around 0, it has little influence (Dong et al., 2021).

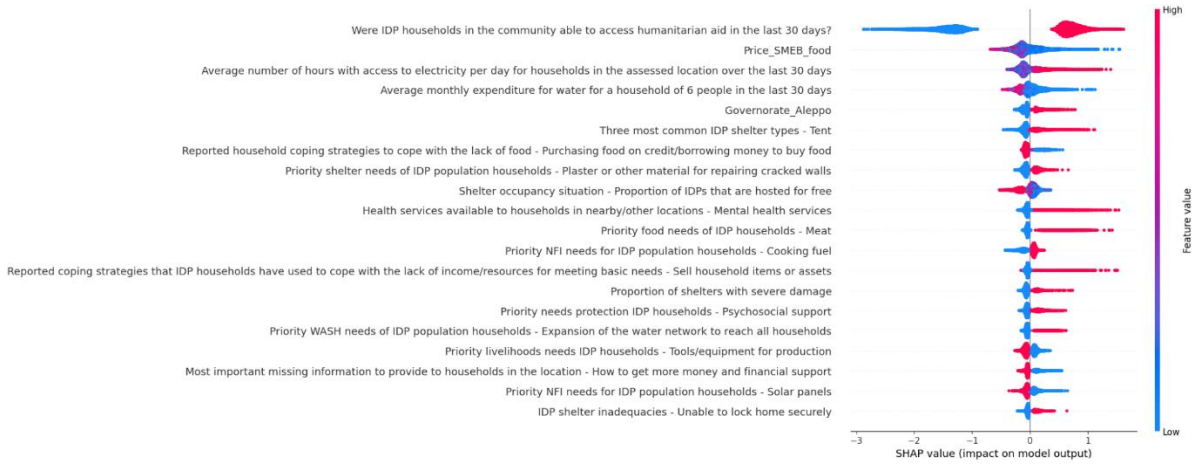

Figure 19: MultiTarget - SHAP Values XGBoost

This SHAP summary plot illustrates the impact of the 340 most important features in the XGBoost model for the first target variable Shelter. The left axis lists the top contributing features, where those at the top have the highest influence on the model's predictions. The SHAP values on the x-axis indicate how much each feature pushes the prediction positively (right) or negatively (left). Red points represent high feature values, while blue points indicate low values.

For instance, "Were IDP households in the community able to access humanitarian aid in the last 30 days?" has the most significant effect on predictions. Other key features, such as food prices, access to electricity, and shelter inadequacies, play a crucial role in determining outcomes. The distribution patterns suggest that some features consistently drive predictions in a certain direction, reinforcing their importance. This analysis helps validate feature selection and guides further model refinement by focusing on the most impactful variables.

6.2.5.4 SHAP Interpretation Across Labels

The SHAP analysis has provided critical insights into the drivers of humanitarian assistance predictions across 16 labels. These insights can help prioritize resources, target interventions more effectively, and build trust in the model's predictive capabilities through transparent, interpretable outputs.

Assistance Type	Top Contributing Features	SHAP Insights
Shelter Assistance	Access to humanitarian aid, shelter damage, material shortages	Locations with severe shelter damage and resource limitations show increased need for assistance.
Health Assistance	Barriers to healthcare, malnutrition treatment gaps	Lack of healthcare access strongly correlates with higher health assistance needs.
NFIs (Non-Food Items)	Cooking fuel, shelter repair materials, coping strategies	Communities selling household items indicate a higher demand for NFIs.
Electricity Assistance	Fuel costs, generator usage, power reliability	Areas with frequent power outages and high fuel expenses see increased predictions.
Food Assistance	Food price inflation, meal skipping, reliance on credit	Food insecurity and high dependency on credit purchases predict higher food assistance needs.
Agricultural Supplies	Food crop production levels, financial limitations	Low agricultural output correlates with increased need for agricultural assistance.
Livelihood Support	Employment barriers, financial insecurity	Unemployment rates strongly influence livelihood support predictions.
Education Assistance	School availability, child protection risks	Limited educational facilities and high early marriage risks drive education support needs.
WASH Assistance	Water infrastructure issues, drinking water concerns	Poor water network conditions increase WASH assistance demands.

Winterization Assistance	Heating deficiencies, seasonal conditions	Cold seasons and heating shortages predict higher winterization needs.
Legal Services Assistance	Documentation challenges, protection risks	Households without legal documents show increased legal assistance requirements.
GBV Services Assistance	Early marriage risks, protection concerns	Higher GBV risks in a region predict increased allocation of services.
CP Services Assistance	Child labor prevalence, school dropout rates	Protection issues like child labor and dropout rates influence CP assistance needs.
Explosive Hazard Awareness Assistance	Reported contamination risks, historical conflict data	Regions with known contamination show heightened awareness predictions.
Mental Health Assistance	Psychological distress reports, lack of mental health services	Communities experiencing mental health crises predict increased support demand.
Cash Assistance	Employment opportunities, financial barriers	Areas with severe financial insecurity predict higher cash assistance needs.

Table 11: MultiTarget - SHAP Interpretation Table

6.2.3 Multi Label Deep Learning Models

This section presents the deep learning models applied to the multi-label classification problem. The models were selected based on their ability to handle tabular data, capture complex feature interactions, and generalize well.

6.2.3.1 Multi-Layer Perceptron (MLP)

MLP is a universal approximator and can model complex, non-linear relationships. Works well with structured/tabular data with sufficient hidden layers and regularization. Baseline deep learning model for comparison with other architectures.

6.2.3.2 Optimized MLP (Hyperparameter Tuning)

The initial MLP model was improved through hyperparameter tuning using Keras Tuner. The best architecture was found with 384 and 128 neurons, L2 regularization, and a learning rate of 0.0005.

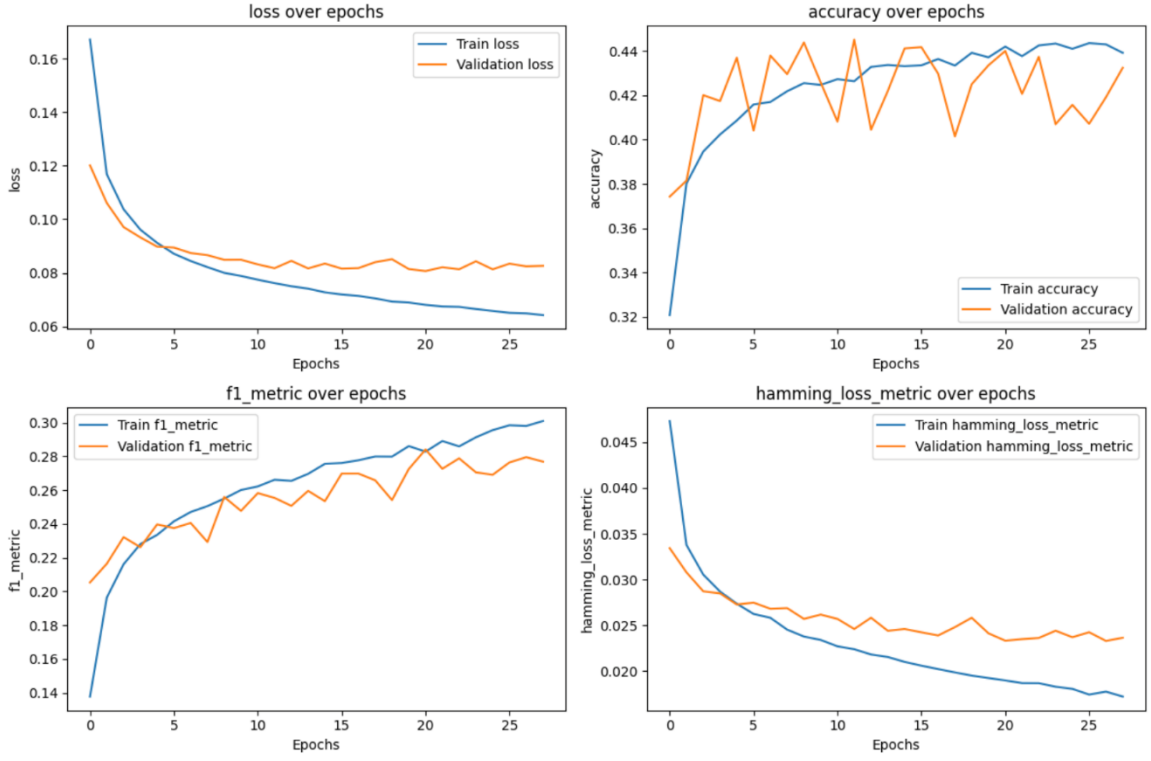

Table 12: MultiTarget - Optimized MLP Metrics

The Best MLP model demonstrates a well-optimized training process, with steady improvements across loss, accuracy, F1-score, and Hamming loss. The loss curves indicate a smooth decline in both training and validation loss, suggesting that the model is learning effectively without signs of overfitting. The accuracy graph shows rapid improvements in the early epochs, stabilizing around 0.44, with validation accuracy exhibiting some fluctuations, which may indicate sensitivity to batch variations. The F1-score trends suggest consistent performance gains, with validation F1-score closely tracking training performance, further confirming generalization. Finally, the Hamming loss reduction over epochs shows improved multi-label classification performance. Overall, the model effectively balances learning and regularization, benefiting from the tuned hyperparameters.

6.2.3.3 TabNet Classifier

The attention-based feature selection technique has been demonstrated to enhance interpretability. This approach has proven to be effective for tabular data by emulating decision tree-based models through the utilization of deep learning methodologies. Furthermore, the incorporation of sparse representation has been shown to mitigate the occurrence of overfitting. The TabNetMultiTaskClassifier was used.

6.2.3.4 Autoencoder + MLP

Reduces the high-dimensional input space to a latent representation of 100 features. Helps denoise and extract hidden patterns before classification.

6.2.3.5 Convolutional Neural Network (CNN)

CNNs, typically used in image processing, have been adapted for tabular data by treating features as a 1D spatial sequence. This allows the network to capture local feature interactions, potentially improving multi-label classification performance. However, NNs can work for multi-label classification, but they are typically better suited for spatial or sequential data rather than purely tabular data. Since our features are mostly categorical and binary, a fully connected MLP or TabNet is more appropriate.

Model	Hamming Loss	F1-Score (Macro)	F1-Score (Micro)	Accuracy
MLP	0.02508	0.47954	0.80345	0.73583
MLP Tuned	0.02397	0.48027	0.81587	0.73963
TabNET	0.02856	0.32815	0.77748	0.71669
Autoencoder + MLP	0.02759	0.43861	0.77921	0.71669
CNN Initial	0.09867	0.14681	0.67523	0.64481
CNN Optimized	0.07865	0.19521	0.74145	0.64443

Table 13: MultiTarget – Deep Learning Model Results

The tuned MLP model shows a well-optimized training process, with steady improvements across loss, accuracy, F1-score, and Hamming loss. Overall, the model effectively balances learning and regularization, benefiting from the tuned hyperparameters. TabNet performs worse than the tuned MLP model, particularly with regard to the F1-Score (Macro). This suggests that TabNet may not fully capture multi-label dependencies. However, TabNet is just as accurate as the Autoencoder + MLP model.The Autoencoder + MLP model is slightly worse than the tuned MLP, but it performs similarly to TabNet. Reducing the number of dimensions improved generalization, but it also led to some loss of information. The CNN performed competitively with other deep learning models, but it did not outperform the optimized MLP. The validation loss fluctuated slightly, indicating some sensitivity to batch variations.

7. Model Evolution

The modeling pipeline followed a structured, iterative approach, beginning with baseline algorithms to establish foundational benchmarks and gradually advancing towards more sophisticated techniques. The process commenced with the implementation of traditional machine learning models, including Random Forest, Support Vector Machine (SVM), and Logistic Regression, which served as initial performance references. This baseline evaluation highlighted the need for models capable of better handling complex, multi-label data structures.

Building on these insights, the focus shifted to XGBoost, a gradient boosting framework renowned for its scalability and superior performance on structured data. XGBoost quickly emerged as a strong contender due to its ability to manage high-dimensional data and its robustness against overfitting. Recognizing its potential, further refinements were applied through per-class threshold optimization, which tailored decision thresholds for each label, significantly improving the balance between recall and precision, particularly for minority classes.

To explore ensemble strategies, Voting and Stacking Classifiers were integrated, combining the strengths of multiple algorithms. Additionally, bagging and boosting methods were tested to enhance model stability and accuracy. However, these ensemble methods did not yield significant improvements over the standalone XGBoost model, which maintained superior efficiency and predictive performance. XGBoost consistently delivers superior results due to its ability to efficiently handle large feature spaces while maintaining computational efficiency. Its robust performance on imbalanced datasets makes it particularly well-suited for complex classification tasks. Additionally, XGBoost leverages parallelization, significantly reducing training time compared to traditional boosting methods. Another key advantage is its built-in regularization mechanisms, which help prevent overfitting, ensuring better generalization on unseen data. These factors collectively contribute to its strong predictive capabilities and make it a powerful choice for high-dimensional and imbalanced classification problems.

For comprehensive benchmarking, a Deep Learning approach using a Multi-Layer Perceptron (MLP) was also implemented. While deep learning demonstrated potential in capturing complex data relationships, it was ultimately outperformed by traditional methods like XGBoost in this specific multi-label context, especially when considering interpretability and computational efficiency.

The pipeline's iterative refinement, from baseline models to advanced boosting techniques and deep learning benchmarks, ensured a robust exploration of modeling strategies, ultimately validating XGBoost with per-class thresholding as the most effective solution for this multi-label classification problem.

Final Comparison Table

Model	Hamming Loss	F1-Score (Macro)	F1-Score (Micro)	Accuracy
Random Forest Tuned	0.0218	0.45679	0.82513	0.75801
SVM Tuned	0.02019	0.59233	0.84624	0.77122
XGBoost Tuned	0.01946	0.59979	0.85065	0.78338
Random Forest SMOTE	0.02293	0.44	0.81	0.74479
SVM SMOTE	0.02047	0.59	0.84	0.76910
XGBoost SMOTE	0.02106	0.59	0.84	0.76530
XGBoost (0.001 feature selec.)	0.01961	0.58088	0.84921	0.78079
XGBoost (0.001+Per Class Threshold)	0.019538	0.63569	0.85404	0.779279
Bagging SVM	0.04062	0.50	0.75	0.61582
Boosting SVM	0.07123	0.39	0.62	0.49840
MLP	0.02508	0.47954	0.80345	0.73583
MLP Tuned	0.02397	0.48027	0.81587	0.73963
TabNET	0.02856	0.32815	0.77748	0.71669
Autoencoder + MLP	0.02759	0.43861	0.77921	0.71669
CNN Tuned	0.07865	0.19521	0.74145	0.64443

Table 14: MultiTarget – All Model Results

8. Conclusion

This project successfully tackled the intricate challenge of both one target and multi-label classification within the humanitarian aid sector, focusing on predicting diverse assistance needs for Internally Displaced Persons (IDPs). The classification problem was complex, not only due to the multi-label nature but also because of significant class imbalances and the high dimensionality of the dataset. Through a methodical and iterative approach that encompassed traditional machine learning models, ensemble methods (including bagging and boosting), and deep learning architectures, the XGBoost model with optimized per-class thresholds ultimately emerged as the most effective solution both for one and multi label problem.

A comprehensive evaluation framework was employed to assess model performance, utilizing metrics such as Hamming Loss, F1-Score (Macro and Micro), and Exact Match Accuracy. While global thresholding initially provided strong results, further refinement through per-class threshold optimization significantly improved the model's sensitivity to minority classes. This nuanced approach enabled the final model to strike an optimal balance between precision and recall, a critical factor in humanitarian data modeling where misclassification can have real-world consequences.

One of the pivotal conclusions of this project is the superiority of the XGBoost model when enhanced with threshold tuning, which allowed for tailored decision boundaries across the 16 target labels. This flexibility resulted in a substantial increase in the Macro F1-Score, reflecting better predictive performance on underrepresented classes. In addition, SHAP (SHapley Additive exPlanations) analysis was leveraged to interpret model predictions, providing transparency into feature importance. Key features such as “Access to humanitarian aid in the last 30 days” were consistently highlighted, aligning with domain knowledge and reinforcing the model's credibility.

While Deep Learning (MLP) models were explored and demonstrated potential, they did not surpass the performance of the optimized XGBoost model for both one label and multi label. Despite promising F1-Scores and the capacity to model complex nonlinear relationships, the MLP required further hyperparameter tuning and increased computational resources, which limited its immediate applicability.

The final XGBoost model is not just a theoretical exercise but is fully primed for practical deployment within humanitarian aid ecosystems. Its integration into existing platforms could significantly enhance operational efficiency and impact in the following ways:

Predicting Assistance Needs: By identifying specific needs across different communities, the model enables targeted intervention strategies.

Optimizing Resource Allocation: Aid organizations can use model outputs to allocate resources more efficiently, prioritizing areas with the most urgent needs.

Supporting Proactive Planning: NGOs and local governments can leverage predictive insights to prepare for future crises, focusing on at-risk communities before issues escalate.

In conclusion, this project not only achieved its core objectives but also laid the groundwork for data-driven humanitarian assistance. By blending machine learning with domain expertise, it offers a scalable solution capable of adapting to evolving humanitarian challenges while maintaining a focus on ethical and impactful outcomes.

9. Challenges and Limitations

9.1 Data Complexity and Limitations

The dataset presented several inherent complexities that posed challenges throughout the project:

High Dimensionality:
With 497 features after preprocessing, the dataset's sheer volume increased computational time and model training complexity. While dimensionality reduction techniques and feature importance analyses helped streamline the data, the high feature count still demanded robust computing power.

Prevalence of Binary and Categorical Variables:
The dataset was dominated by binary and low-cardinality categorical variables, which, when encoded, further expanded the feature space. This not only complicated data interpretation but also increased the risk of model overfitting.

9.2 Managing Multi-Label Classification in Imbalanced Data

Traditional classification algorithms often underperform in such contexts, as they are not inherently designed to handle the correlations between labels or the varying distributions across classes. In this project, this imbalance led to biased predictions skewed towards more common assistance types, such as Food, Nutrition and Health, while underrepresenting crucial but less frequent labels like GBV Services and Explosive Hazard Risk Awareness.

To mitigate this, strategies such as Synthetic Minority Over-sampling Technique (SMOTE) and per-class threshold tuning were applied. While these methods improved recall for minority classes, they also introduced challenges, such as slight overfitting and increased computational demands.

The other significant scientific challenge encountered during the project was managing the complexity of multi-label classification in an imbalanced dataset. Unlike single-label classification, where each sample belongs to only one class, multi-label classification requires predicting multiple classes for each instance. This adds layers of complexity, especially when labels are imbalanced resulting in models that tend to favor majority classes while neglecting minority ones.

9.3 Time-Intensive Processes

Several project phases demanded more time and resources than initially anticipated:

Feature Engineering:
The dataset contained a large proportion of binary and categorical variables, leading to high dimensionality. Iterative feature engineering was required to reduce noise while preserving meaningful relationships, particularly in handling low-cardinality categorical variables that risked inflating the feature space when encoded.

Handling Missing Values:
A considerable portion of the dataset contained missing values, necessitating careful imputation or exclusion strategies. While imputation helped retain data volume, it also introduced the risk of bias, whereas exclusion strategies reduced the overall dataset size, impacting model generalizability.

Hyperparameter Tuning:
Conducting Grid Search Cross-Validation for complex models like XGBoost and Random Forest demanded substantial computational resources. The multi-output nature of the problem further extended the search space, leading to longer runtimes.

Threshold Tuning:
The process of per-class threshold optimization significantly improved minority label performance but was highly time-consuming. Unlike a global threshold approach, this method required extensive experimentation to identify optimal decision boundaries for each label.

9.4 Infrastructure and Computational Constraints

Ensemble models with SVM, XGBoost and Random Forest, especially when subjected to Grid Search CV, demanded extensive computational time. Training multi-label models further compounded the issue due to the necessity of training a separate estimator for each label.

While the initial focus was on traditional machine learning models, the introduction of Deep Learning (MLP) architectures revealed substantial limitations in computational power. Training deep networks efficiently required GPU acceleration, which was not consistently available, limiting the scope and depth of hyperparameter tuning.

Managing large datasets with high dimensionality, particularly during SHAP analysis and oversampling processes, pushed memory limits. Running interpretability analyses on the entire feature set proved resource-intensive, necessitating downsampling strategies that could compromise the depth of insights.

10. Future Directions

10.1 Avenues for Improvement

While the project successfully developed a high-performing and interpretable multi-label classification model, several avenues remain open for future enhancements to further optimize predictive performance and generalizability.

10.1.1 Deep Learning Architectures

Future iterations of this project could benefit from exploring advanced deep learning techniques tailored for sequential and structured data. Architectures such as Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), or Transformer-based models could be highly effective, especially if data is analyzed as temporal or time-series data. These models can capture complex dependencies over time, providing richer insights into evolving humanitarian needs and enabling dynamic resource allocation.

10.1.2 Advanced Oversampling Methods

Although SMOTE was initially applied to address class imbalance, it introduced overfitting in certain instances. As an alternative, the implementation of Adaptive Synthetic Sampling (ADASYN)—a more flexible oversampling technique that focuses on harder-to-learn samples—could offer better handling of class imbalances in multi-label datasets. Additionally, exploring multi-label-specific oversampling methods which is a developing area now,could further improve recall for minority classes without compromising overall model performance.

10.1.3 Threshold Optimization Techniques

While per-class threshold tuning significantly improved macro F1-scores, more sophisticated optimization strategies, such as Bayesian Optimization or Genetic Algorithms, could refine these thresholds further. This could help in minimizing bias towards dominant classes while maximizing performance for underrepresented labels.

10.1.4 Feature Engineering and Dimensionality Reduction

Although feature importance was addressed using SHAP and Xgboost best parameters, additional/manual dimensionality reduction might streamline the feature space and improve computational efficiency without sacrificing model accuracy.

10.2 Contribution to Scientific Knowledge

This project has made meaningful contributions to the field of machine learning, particularly in the context of multi-label classification within humanitarian data analysis:

10.2.1 Per-Class Threshold Tuning in Multi-Label Classification

The project demonstrated the effectiveness of per-class threshold tuning as a strategy to address class imbalance, significantly improving the Macro F1-Score while maintaining a low Hamming Loss. This nuanced approach highlights the importance of tailored decision boundaries in multi-label tasks, especially in datasets where certain classes are underrepresented.

10.2.2 Model Interpretability in Complex Multi-Label Settings

By incorporating SHAP (SHapley Additive exPlanations) into the evaluation process, the project underscored the critical role of interpretability in AI models, particularly in high-stakes domains like humanitarian aid. SHAP provided a transparent view of feature contributions, enabling stakeholders to trust and understand the rationale behind model predictions—an essential factor in ethically grounded decision-making.

10.2.3 Development of a Replicable Pipeline for Multi-Label Tasks

The project established a comprehensive and adaptable pipeline for multi-label classification that can be replicated across various domains, such as medical diagnostics, fraud detection, and environmental monitoring. This end-to-end framework—from data preprocessing and feature selection to model training, hyperparameter tuning, and interpretability—serves as a blueprint for similar data-intensive applications.

10.3. Final Remarks

In conclusion, this project has successfully delivered a high-performing, interpretable one label and multi-label classification model that directly supports humanitarian aid decision-making. Through the application of advanced machine learning techniques, including XGBoost with per-class threshold tuning and SHAP-based interpretability, the project provides a robust tool for predicting diverse humanitarian needs with both accuracy and transparency.

The methodological advancements made, particularly in handling imbalanced multi-label data, contribute valuable insights to the broader scientific community. While the current model achieves strong predictive performance, the outlined avenues for improvement, especially through deep learning and advanced oversampling pave the way for future research and enhanced applications.

Ultimately, this project not only fulfills its immediate goal of supporting more efficient and data-driven humanitarian interventions but also offers a scalable framework applicable to various sectors facing complex, multi-label classification challenges.

11. References

Chen, S. (2021). *Interpretation of multi-label classification models using shapley values* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2104.10505

Dong, H., Sun, J., & Sun, X. (2021). A Multi-Objective Multi-Label Feature Selection Algorithm Based on Shapley Value. *Entropy*, *23*(8), Article 8. https://doi.org/10.3390/e23081094

scikit-learn.org. (2025a). *3.4.4.9 Precision, recall and F-measures*. Scikit-Learn. https://scikit-learn/stable/modules/model_evaluation.html

scikit-learn.org. (2025b). *SGDClassifier*. Scikit-Learn. https://scikit-learn/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

Sukhwani, N. (2020, Juni 16). Handling Data Imbalance in Multi-label Classification (MLSMOTE). *TheCyPhy*. https://medium.com/thecyphy/handling-data-imbalance-in-multi-label-classification-mlsmote-531155416b87