Studying the Best Methods of Classifying the Status of Missing Persons Using Machine Learning

CSC 400 Fall 2024

Independent Study Final Report (Completed 12/11/2024)

Author: Jerry Huo

Advisor: Franz J. Kurfess

Cal Poly San Luis Obispo, Computer Science and Software Engineering Department

Abstract

Missing persons searches are time-sensitive operations where the first 51 hours are critical for successful outcomes. This project explores the application of machine learning techniques to classify the status of missing persons, aiming to optimize resource allocation and improve search and rescue (SAR) success rates. The study implements and compares multiple classification approaches, including logistic regression, support vector machines, and a stacking model, built using Python with pandas and sklearn libraries. The models incorporate various features such as age, fitness level, experience, incident environment, and temporal factors, trained and validated on the ISRID 3 dataset containing over 1,000 entries. The challenges of imbalanced datasets and model interoperability are also addressed in this research. Performance evaluation utilizes metrics including accuracy, precision, recall, and F1 score, with cross-validation ensuring robust results.

Introduction

In search and rescue (SAR) operations, time is the most critical factor in determining survival rates for missing persons. Studies have consistently shown that the first hours after a person goes missing are crucial, with survival probabilities decreasing significantly beyond this window up until a cutoff of 51 hours (Adams et al). Despite technological advancements in search capabilities, SAR teams still face substantial challenges in resource allocation and decision-making during these critical hours. These challenges are compounded by the diverse nature of missing person cases, varying environmental conditions, and the limited resources available to search teams.

The efficiency of SAR operations heavily depends on the ability to quickly and accurately assess the likely status and location of missing persons based on available information. Traditional approaches rely heavily on human expertise and historical precedents, which, while valuable, may be subject to cognitive biases and limitations in processing multiple variables simultaneously. This is particularly challenging when SAR teams must make rapid decisions with incomplete information under high-pressure situations.

Machine learning (ML) offers a promising solution to enhance the decision-making process in SAR operations. By analyzing patterns in historical missing persons data, ML models can provide data-driven insights to support and augment human expertise. Recent advancements in ML techniques, particularly in classification algorithms and model interpretability, have created new opportunities for developing robust prediction systems that can assist SAR teams in prioritizing search areas and allocating resources more effectively. This is particularly crucial given the challenges facing SAR teams globally. Gary Bloom, a seasoned search and rescue volunteer with decades of field experience, notes that “Given the volunteer nature of Search and Rescue teams around the globe, limited resources and at times, limited experience can slow down search operations. AI bridges this gap by using technology to predict where the missing subject will be located, allowing the limited resources to be properly assigned to the most likely locations and filling the experience gap to improve the odds of a positive outcome for the missing subject.” (Bloom 2024). ML models could help capture and formalize this expertise for future generations of SAR professionals.

Research

Machine learning is a branch of artificial intelligence that utilizes algorithms and data to reach a conclusion (IBM “What is machine learning?”). Essentially, it identifies and takes patterns within data to make predictions. In the context of Search and Rescue operations, machine learning models can analyze historical incident data to identify complex relationships between various factors that human operators might not readily discern in a timely manner.

So what factors within the data would be important and relevant to these predictions? In the dataset used, the important factors include age, level of fitness, level of experience, type of activity, environment, status, and total hours. Age is a critical factor that affects survival capability and behavior patterns. Level of fitness indicates physical capability to handle environmental challenges. Level of experience reflects their ability to make critical survival-oriented decisions. Type of activity explains the intended purpose and potential risks of the subject's presence in the area. Incident environment relates to the terrain type. Finally, the subject’s status is what the model will predict, and indicates the state of the subject upon discovery.

This research explores multiple machine learning approaches to address the classification challenge: logistic regression, support vector machine, and the stacking model. Logistic regression, despite its name, is a classification algorithm that estimates the probability of a missing person belonging to each status category. It applies a logistic function to a linear combination of input features (like age, fitness level, and environment), and transforms them into probabilities between 0 and 1. For example, given a 45-year-old hiker with moderate fitness in mountainous terrain, the model might calculate a 75% probability of being found 'Alive_Well’ (alive and well). (IBM “What is logistic regression?”)

Support Vector Machine (SVM) takes a geometric approach to classification by finding optimal boundaries, called hyperplanes, that maximize the separation between different status categories in a multidimensional feature space. SVM excels at handling both linear relationships (where the relationship between variables can be represented by a straight line, such as how survival probability might decrease proportionally with age) and non-linear relationships (where the relationship follows a curved or more complex pattern, such as how survival probability may vary with temperature in a way that peaks at moderate temperatures and decreases at extremes). The algorithm achieves this through kernel functions that transform the data into higher dimensions where linear separation becomes possible. For example, while it might be impossible to draw a straight line to separate 'Alive_Well' from 'DOA' (dead on arrival) cases when looking at age and fitness level alone, SVM can transform these features where such separation becomes feasible. This capability is particularly valuable in SAR applications where the relationship between features like environmental conditions and survival outcomes may be highly complex and non-linear. (GeeksforGeeks)

The stacking model is an ensemble approach that combines the strengths of multiple base models, which, in this case, are logistic regression and SVM. It works by training these base models independently and then using their predictions as inputs for a meta-model that learns how to optimally combine their individual predictions. This hierarchical structure allows the stacking model to capture different aspects of the SAR data - for instance, logistic regression might better handle the direct relationship between age and survival probability, while SVM might better capture complex interactions between environment and experience level. The stacking model then learns which base model's predictions are most reliable under different circumstances, potentially improving overall prediction accuracy compared to any single model approach. (Brijesh)

SAR datasets face a critical issue of having inconsistent factors across datasets, which poses significant challenges for machine learning applications. This inconsistency can appear in several ways: varying terminology for similar status conditions, different methods of recording environmental conditions, non-standardized approaches to documenting subject characteristics across different SAR organizations and regions, etc. For example, environmental conditions might be recorded with precise meteorological data in some cases but only basic descriptors in others. These inconsistencies make it difficult to merge datasets from different sources, limiting the amount of training data available for machine learning models. Furthermore, this variation in data collection and recording practices can impact model performance when deployed across different regions or organizations, as the model may need to account for these discrepancies in data representation.

Related Works

The application of machine learning in emergency response and search operations has gained significant traction across various fields. Three key areas demonstrate particularly relevant applications: medical diagnosis systems, disaster response operations, and wildlife conservation technologies.

In medical diagnosis, machine learning models have proven effective in time-critical decision making. A notable example is the Breast Cancer Wisconsin Dataset, used in one of Kaggle's machine learning competitions, which demonstrates how ML models can accurately classify tumors as benign or malignant based on features extracted from fine needle aspirate images of breast masses. These models achieve high accuracy rates by analyzing cellular features such as radius, texture, perimeter, and area to predict tumor status, showing how machine learning can support critical medical decisions. The success of these diagnostic models, particularly in their ability to process multiple features simultaneously for binary classification (benign/malignant), parallels the needs of SAR status prediction where multiple variables must be considered to classify subject conditions. These diagnostic applications share similarities with SAR operations in their need to make accurate predictions under time pressure using complex, multivariate data. (Matsui)

Disaster response operations have also benefited from machine learning applications. As demonstrated by Viswambharan et al., advanced AI models are deployed to "detect or classify damaged infrastructure, as well as identify threats such as wildfires and floods" in the wake of disasters. These models process images from satellites, aerial drones, and/or ground-based cameras to assess damaged areas and calculate adequate resource allocation. For instance, the Prithvi model, developed by NASA and IBM, utilizes machine learning to identify burn scars in the "aftermath of wildfires, enabling emergency responders to assess damage and plane recovery efforts more efficiently". These applications mirror those of SAR operations, particularly their need for rapid assessment and efficient resource allocation.

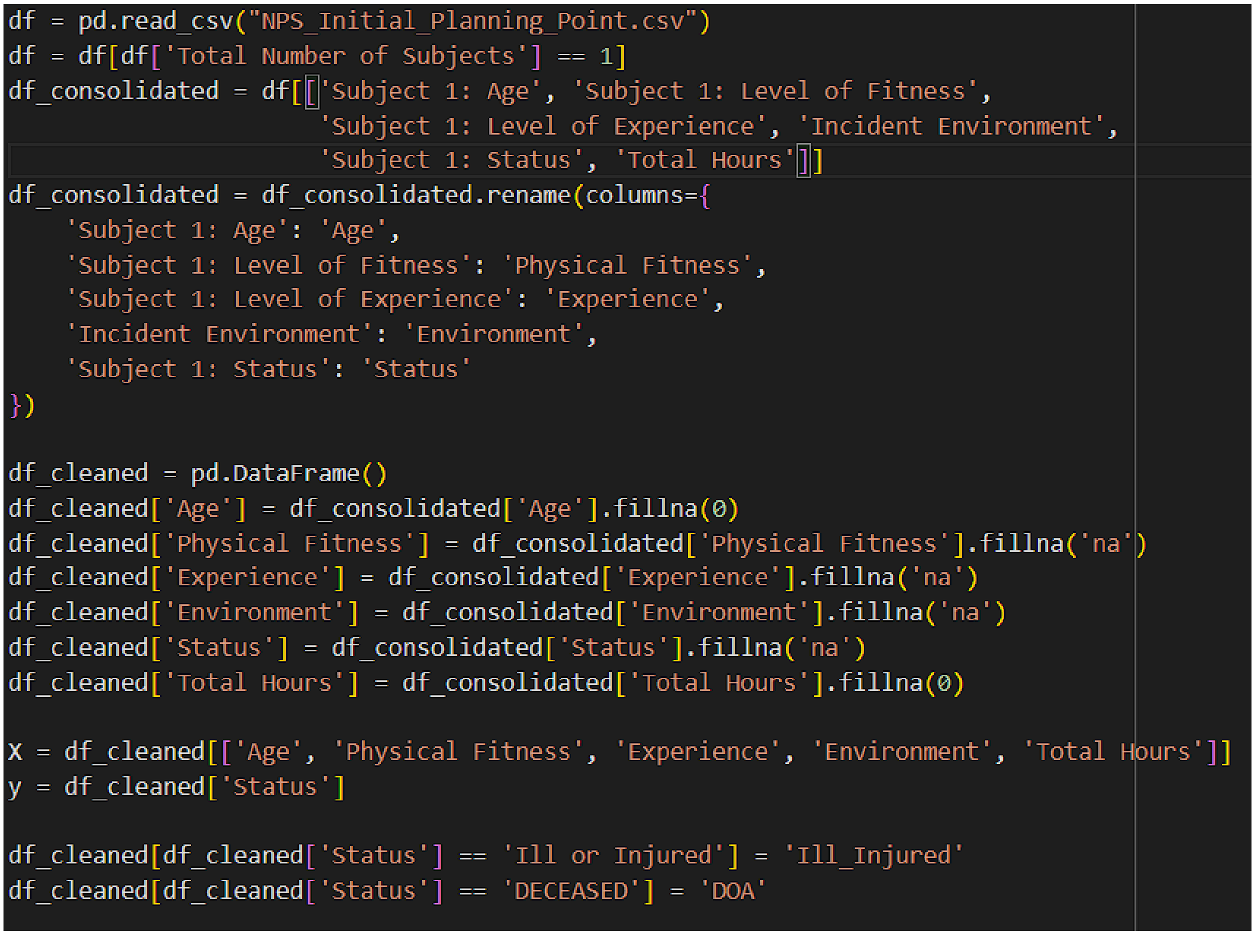
Wildlife conservation presents another relevant application area, specifically in how machine learning can assist in training and knowledge transfer. As Bloom noted earlier regarding the volunteer nature of SAR teams and their varying degrees of experience, this creates a significant training challenge. Researchers argue "animal ecologists can capitalize on large datasets generated by modern sensors by combining machine learning approaches with domain knowledge … This approach will require close interdisciplinary collaboration to ensure the quality of novel approaches and train a new generation of data scientists in ecology and conservation." (Tuia et al.). In this context, the ML models can bridge the experience gap by helping newcomers quickly identify patterns in the search data, providing an output that would have otherwise taken precious minutes to come to.

While some current SAR operations already utilize various technological tools, they are not common in the field and machine learning integration remains limited. As Bloom observes, "Most Search and Rescue operations are manually run processes on paper. Automating the Search and Rescue process not only improves efficiency of the search operation, but also opens up the opportunity to apply AI technologies to an ongoing search operation. Net result - - find the missing faster." This reliance on manual processes highlights a significant opportunity for technological advancement in the field. The transition from paper-based systems to automated solutions would not only streamline operations but also create a foundation for implementing more sophisticated machine learning approaches.

Furthermore, although there exists commercial SAR software such as SARTopo and CalTopo, they provide sophisticated mapping and planning tools but lack predictive capabilities for missing person status. These systems excel at data organization and visualization but rely entirely on human expertise for decision-making. This gap presents an opportunity for machine learning integration to enhance, rather than replace, existing tools.

System Design and Implementation

The raw data often requires transformation to be most useful for machine learning models. The feature engineering process begins with the ISRID 3 dataset, focusing specifically on single-subject incidents to ensure consistency in the analysis. This filtering strategy helps eliminate the additional complexity and potential confusing factors present in multi-subject search and rescue operations. The initial dataset is consolidated to focus on six key features that are the most relevant in predicting subject status: age, level of fitness, level of experience, type of activity, environment, status, and total hours. Data cleaning and standardization procedures are implemented to handle missing values. Figure 1 shows this process. Numerical features (age, total hours) are filled with 0 when missing. Categorical features (physical fitness, experience, environment) are assigned ‘na’ (not available) when missing. Status classifications are standardized, with ‘Alive_Well’ and ‘Not_Found’ staying the same, and 'Ill or Injured' converted to 'Ill_Injured' and 'DECEASED' to 'DOA' for consistency.



```python
df = pd.read_csv("NPS_Initial_Planning_Point.csv")
df = df[df['Total Number of Subjects'] == 1]
df_consolidated = df[['Subject 1: Age', 'Subject 1: Level of Fitness',
                      'Subject 1: Level of Experience', 'Incident Environment',
                      'Subject 1: Status', 'Total Hours']]
df_consolidated = df_consolidated.rename(columns={
    'Subject 1: Age': 'Age',
    'Subject 1: Level of Fitness': 'Physical Fitness',
    'Subject 1: Level of Experience': 'Experience',
    'Incident Environment': 'Environment',
    'Subject 1: Status': 'Status'
})

df_cleaned = pd.DataFrame()
df_cleaned['Age'] = df_consolidated['Age'].fillna(0)
df_cleaned['Physical Fitness'] = df_consolidated['Physical Fitness'].fillna('na')
df_cleaned['Experience'] = df_consolidated['Experience'].fillna('na')
df_cleaned['Environment'] = df_consolidated['Environment'].fillna('na')
df_cleaned['Status'] = df_consolidated['Status'].fillna('na')
df_cleaned['Total Hours'] = df_consolidated['Total Hours'].fillna(0)

X = df_cleaned[['Age', 'Physical Fitness', 'Experience', 'Environment', 'Total Hours']]
y = df_cleaned['Status']

df_cleaned[df_cleaned['Status'] == 'Ill or Injured'] = 'Ill_Injured'
df_cleaned[df_cleaned['Status'] == 'DECEASED'] = 'DOA'
```

Figure 1

The feature set is then split into predictor variables (X) and the target variable (y). The predictor matrix X contains five features: age, physical fitness, experience, environment, and total hours, while y contains the status outcome. This separation prepares the data for subsequent model training and evaluation phases.

Preprocessing is done to balance the need to maintain data integrity while ensuring the dataset is complete and consistent for machine learning applications. The choice to retain rather than remove records with missing values helps preserve potentially valuable information, while the standardization of status categories simplifies the classification task without losing critical outcome distinctions. As seen in Figure 1, the dataset's missing categorical features are filled with the value 'na' which represents 'not available', and the missing quantitative features with the value '0'.

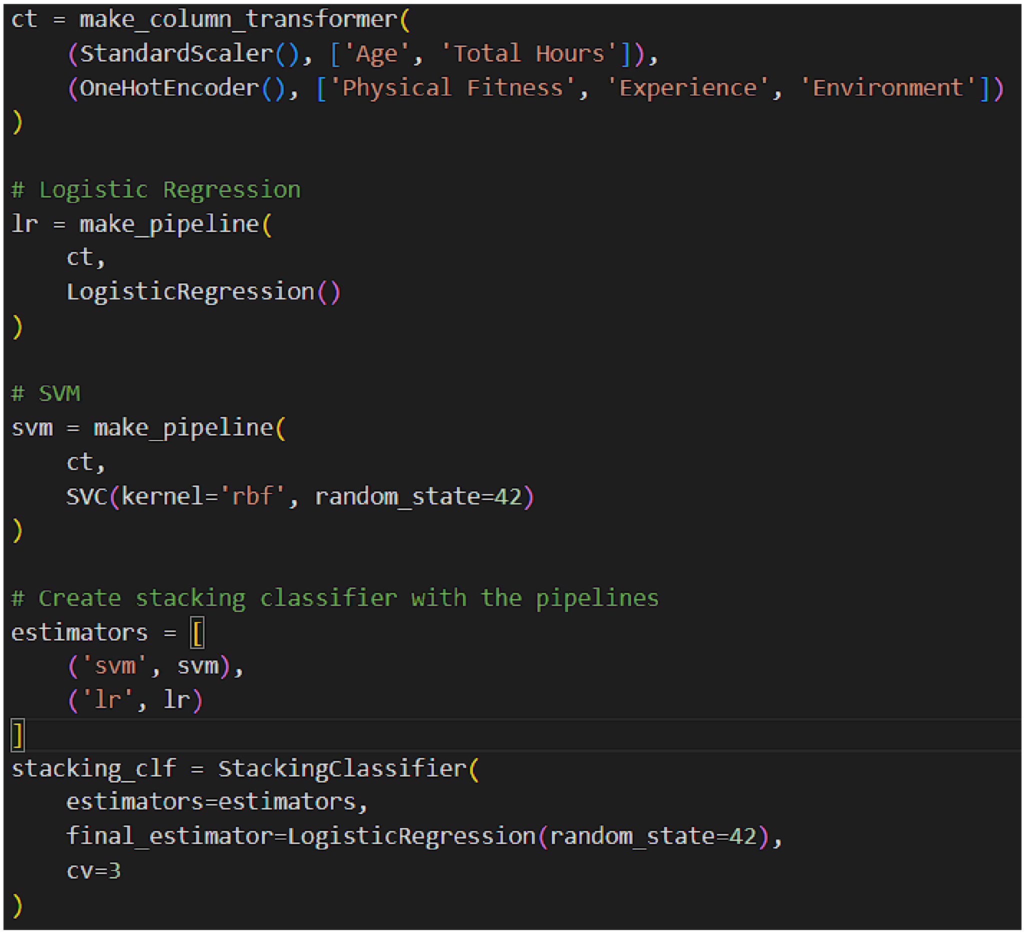
Following data preprocessing, the three previously mentioned ML models are implemented. Logistic regression serves as the baseline model, using scikit-learn's LogisticRegression class with default parameters to establish initial performance metrics. The Support Vector Machine model is implemented using scikit-learn's SVC class with a radial basis function (RBF) kernel, chosen for its ability to handle non-linear relationships between features. The RBF kernel transforms the feature space to find optimal separation boundaries between status categories. Finally, a stacking model is constructed to leverage the strengths of both base models. The stacking implementation follows scikit-learn's StackingClassifier framework, with cross-validation used during training to prevent leakage between the base models and meta-classifier.

Figure 2 illustrates the model development process, beginning with the creation of a column transformer—a preprocessing tool essential for handling our dataset's mixed data types. This transformer applies two distinct types of preprocessing: scaling for numerical features and encoding for categorical features.

Scaling is applied to quantitative variables (age and total hours) to normalize them to comparable ranges. This transformation is crucial because while age typically ranges from 0 to 100, total hours can exceed 100, and having features on vastly different scales can adversely affect model performance. The scaling process transforms these features to a standardized range between 0 and 1.

For categorical features (physical fitness, experience, environment), one-hot encoding is implemented to convert text categories into a machine-readable format. For instance, the physical fitness feature, which contains categories like 'excellent', 'good', 'fair', 'poor', and 'unknown', is transformed into separate binary columns. Each new column represents one category (e.g., Physical_Fitness_Excellent), with a value of 1 indicating the presence of that category and 0 indicating its absence.

Finally, these preprocessed features are integrated into model-specific pipelines. Each pipeline combines the column transformer with its respective classification model (logistic regression, SVM, or stacking classifier), creating a streamlined workflow that ensures consistent data preprocessing and model application while preventing data leakage between training and testing phases.



```
ct = make_column_transformer(
    (StandardScaler(), ['Age', 'Total Hours']),
    (OneHotEncoder(), ['Physical Fitness', 'Experience', 'Environment'])
)

# Logistic Regression
lr = make_pipeline(
    ct,
    LogisticRegression()
)

# SVM
svm = make_pipeline(
    ct,
    SVC(kernel='rbf', random_state=42)
)

# Create stacking classifier with the pipelines
estimators = [
    ('svm', svm),
    ('lr', lr)
]
stacking_clf = StackingClassifier(
    estimators=estimators,
    final_estimator=LogisticRegression(random_state=42),
    cv=3
)
```

Figure 2

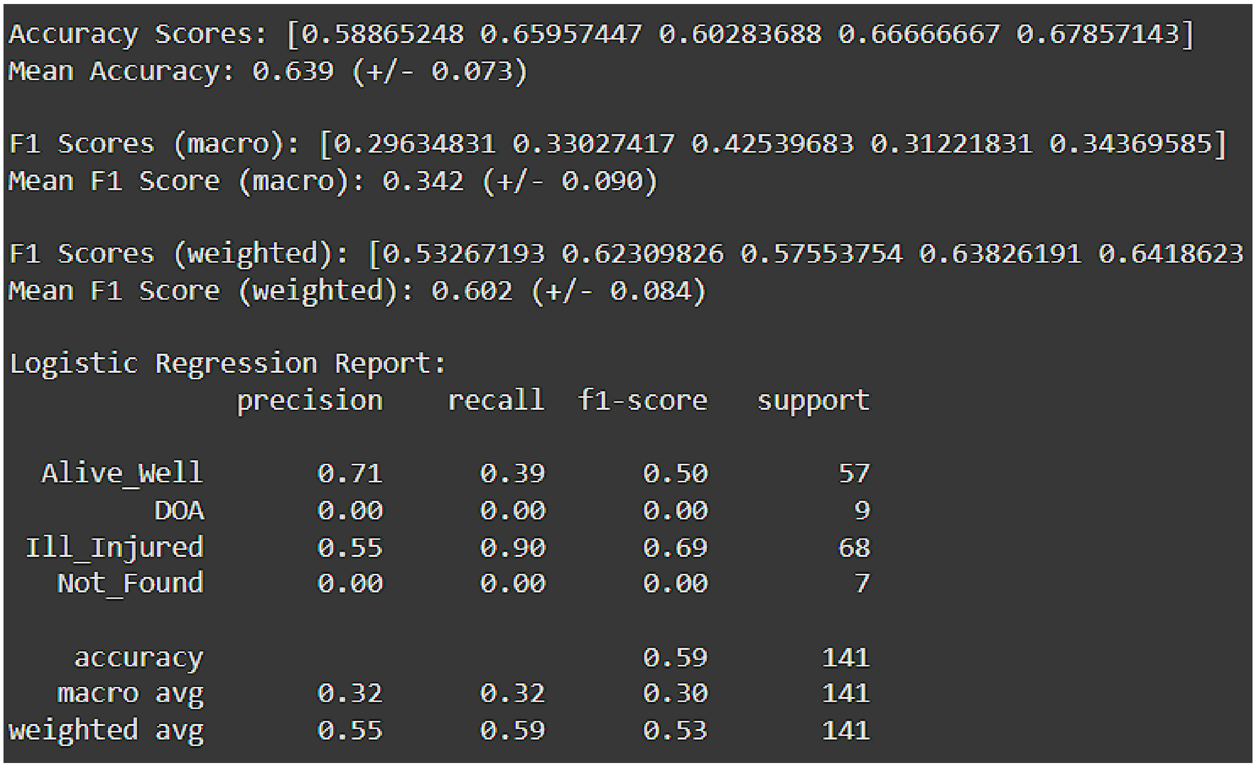
Testing

The model is trained and tested on Dr. Robert J. Koester's ISRID-3 NPS Initial Planning Point dataset, which contains over 1000 entries. The dataset is divided in a systematic approach: 80% of the data is used for model training, and 20% is used for final model evaluation. A K-fold cross-validation is a statistical technique used to evaluate the performance of a model. Cross-validation works by splitting the dataset into k groups, and for each group, it is taken as the testing data set while the other groups are used as the training dataset. The evaluation score of each group is saved, and later summarized. For this implementation, training is divided into 5 folds.

There are multiple evaluation metrics used to provide a comprehensive analysis of model performance. Accuracy measures overall correct predictions, and is calculated as

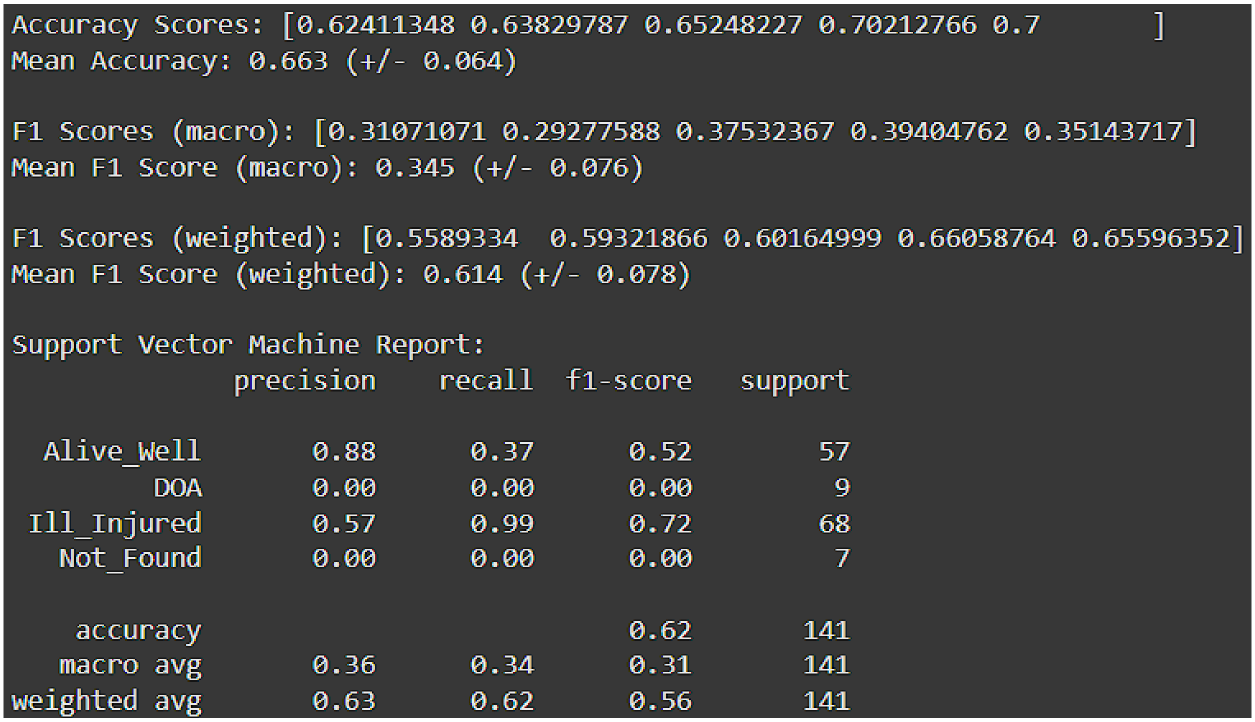
$$(Num\ True\ Positives\ +\ Num\ True\ Negatives)\ /\ Num\ Total\ Predictions.$$

True positives are correct predictions of a positive outcome (a yes is predicted as a yes) and true negatives are correct predictions of a negative outcome (a no is predicted as a no). While useful, potential imbalances cause this metric to be insufficient alone. Precision indicates the proportion of correct positive predictions, and is critical to minimizing false positives. Recall measures the proportion of actual positive cases correctly identified. The F1-score is the harmonic mean of precision and recall and provides a balanced measure of model performance. Each metric is given a score between 0.00 and 1.00, where higher a score indicates better performance.



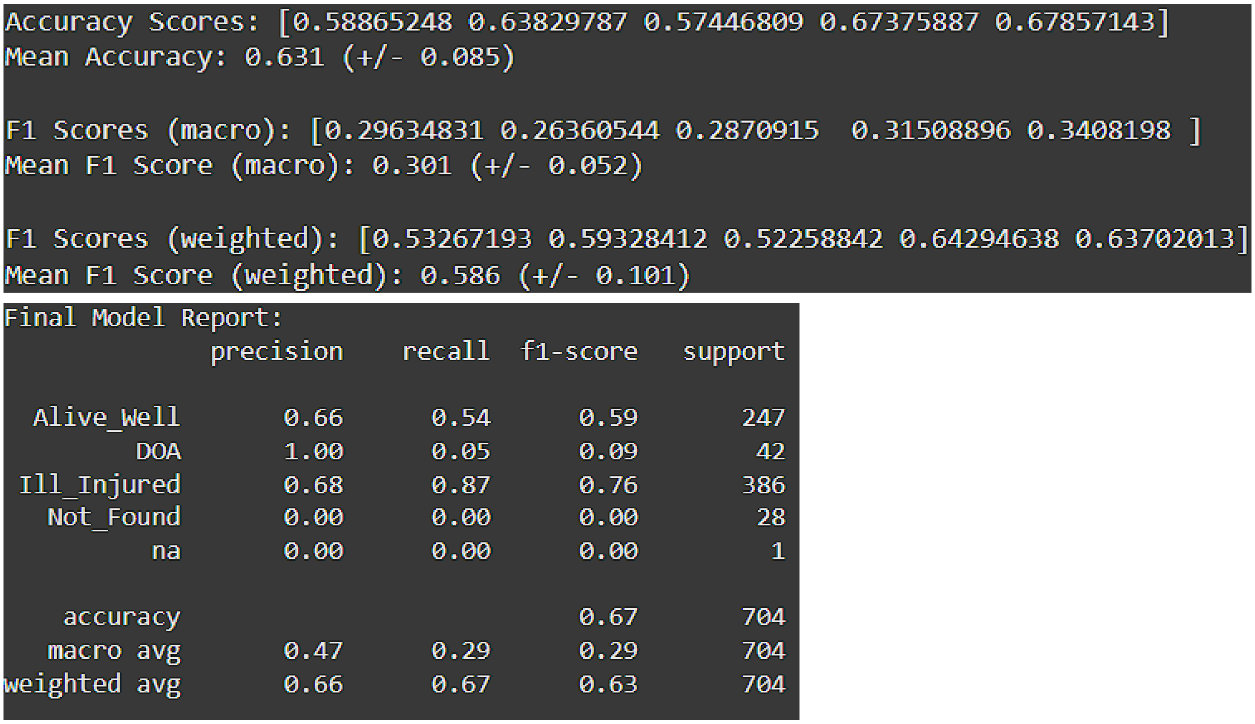
```
Accuracy Scores: [0.58865248 0.65957447 0.60283688 0.66666667 0.67857143]
Mean Accuracy: 0.639 (+/- 0.073)

F1 Scores (macro): [0.29634831 0.33027417 0.42539683 0.31221831 0.34369585]
Mean F1 Score (macro): 0.342 (+/- 0.090)

F1 Scores (weighted): [0.53267193 0.62309826 0.57553754 0.63826191 0.6418623
Mean F1 Score (weighted): 0.602 (+/- 0.084)

Logistic Regression Report:
              precision    recall  f1-score   support

  Alive_Well       0.71      0.39      0.50        57
         DOA       0.00      0.00      0.00         9
 Ill_Injured       0.55      0.90      0.69        68
   Not_Found       0.00      0.00      0.00         7

    accuracy                           0.59       141
   macro avg       0.32      0.32      0.30       141
weighted avg       0.55      0.59      0.53       141
```

Figure 3



```
Accuracy Scores: [0.62411348 0.63829787 0.65248227 0.70212766 0.7       ]
Mean Accuracy: 0.663 (+/- 0.064)

F1 Scores (macro): [0.31071071 0.29277588 0.37532367 0.39404762 0.35143717]
Mean F1 Score (macro): 0.345 (+/- 0.076)

F1 Scores (weighted): [0.5589334  0.59321866 0.60164999 0.66058764 0.65596352]
Mean F1 Score (weighted): 0.614 (+/- 0.078)

Support Vector Machine Report:
              precision    recall  f1-score   support

  Alive_Well       0.88      0.37      0.52        57
         DOA       0.00      0.00      0.00         9
 Ill_Injured       0.57      0.99      0.72        68
   Not_Found       0.00      0.00      0.00         7

    accuracy                           0.62       141
   macro avg       0.36      0.34      0.31       141
weighted avg       0.63      0.62      0.56       141
```

Figure 4



```
Accuracy Scores: [0.58865248 0.63829787 0.57446809 0.67375887 0.67857143]
Mean Accuracy: 0.631 (+/- 0.085)

F1 Scores (macro): [0.29634831 0.26360544 0.2870915  0.31508896 0.3408198 ]
Mean F1 Score (macro): 0.301 (+/- 0.052)

F1 Scores (weighted): [0.53267193 0.59328412 0.52258842 0.64294638 0.63702013]
Mean F1 Score (weighted): 0.586 (+/- 0.101)
```

```
Final Model Report:
              precision    recall  f1-score   support

  Alive_Well       0.66      0.54      0.59       247
         DOA       1.00      0.05      0.09        42
 Ill_Injured       0.68      0.87      0.76       386
   Not_Found       0.00      0.00      0.00        28
          na       0.00      0.00      0.00         1

    accuracy                           0.67       704
   macro avg       0.47      0.29      0.29       704
weighted avg       0.66      0.67      0.63       704
```

Figure 5

Logistic regression (Figure 3), serving as the baseline model, achieved a mean accuracy of 63.9% (±7.3%) across cross-validation folds. It demonstrated a notable bias toward predicting the Ill_Injured class, achieving 90% recall but only 55% precision for this category. While it showed moderate success with Alive_Well predictions (71% precision, 39% recall), it completely failed to identify DOA and Not_Found cases.

The Support Vector Machine model (Figure 4) showed modest improvements over logistic regression, achieving a higher mean accuracy of 66.3% (±6.4%). Most notably, it achieved significantly better precision for Alive_Well predictions (88% compared to logistic regression's 71%), though at a similar recall rate. The model showed particularly strong performance in identifying Ill_Injured cases, with an impressive 99% recall rate and 57% precision. However, like the logistic regression model, it failed to identify any cases in the DOA and Not_Found categories.

The stacking model (Figure 5), achieved a mean accuracy of 63.1% (±8.5%). While this overall accuracy was slightly lower than the other models, it showed more balanced performance across classes. It achieved the best performance for Ill_Injured cases with an F1-score of 0.76, and demonstrated improved capability in handling DOA cases with perfect precision but very low recall (5%). The model also showed more balanced performance for Alive_Well predictions (66% precision, 54% recall) compared to the other models. However, it still struggled with Not_Found cases, highlighting persistent challenges with minority classes despite the ensemble approach.

Future Work

The current implementation of the machine learning model for SAR status classification presents several opportunities for enhancement and expansion. There are four main areas for future development: model refinement, result interpretability, API development, and user interface design.

Model refinement presents two challenges. First, the current dataset of 1000 entries, while valuable, greatly limits the models' ability to learn complex relationships and generalize effectively. Particularly, for classes like 'DOA' and 'Not_Found', the linear regression model and SVM model both reported to have only used 9 and 7 entries respectively. Expanding the dataset through partnerships with additional SAR organizations, like was done with Dr. Koester, would likely improve the models' performance and reliability. Second, SAR datasets face inconsistencies with how organizations record and categorize data. Future work should focus on developing robust feature engineering techniques to harmonize different data sources.

Result interpretability could be significantly improved through SHAP (SHapley Additive exPlanations) values. SHAP explains "the predictions of Machine Learning models in a way humans can understand" (mikesuperman). This way, transparency could be provided through explanations of individual predictions. Search and rescue teams would then easily understand why the model makes certain predictions, building trust in the system and providing valuable insights for operational decision-making.

API development would focus on creating a scalable interface that allows seamless integration to existing SAR devices. The API would need to handle real-time data input, process predictions efficiently, and provide standardized output formats that can be easily interpreted by various client applications. The API should also include proper documentation and example implementations to promote adoption by different SAR organizations.

User interface design would be following a similar path to API development. It would prioritize creating an intuitive and accessible platform under conditions typical of SAR missions.

The interface should direct users to where they can input data, and display model predictions along with confidence levels and key factors. Given that “Most Search and Rescue operations are manually run processes on paper” (Bloom), the interface should be designed to facilitate a smooth transition from paper-based systems to digital tools.

Conclusion

Although the predictive capability of the models suggest room for improvement, this research project demonstrates the potential for machine learning applications to enhance Search and Rescue operations through improved status classification of missing persons. The development and implementation of this system has yielded several insights and contributions to both the technical and operational aspects of SAR activities. The analysis of logistic regression, support vector machine, and stacking models reveals both the promise and challenges of applying machine learning to SAR operations. As Bloom emphasizes, AI technology offers the opportunity to help find missing persons faster, and this research provides a foundation for that goal. These findings, combined with the proposed future developments in model refinement, interpretability, API development, and user interface design, establish a roadmap for continuing advancement in this critical field where technological innovation could directly impact survival outcomes.

References

Adams, Annette Lynn, et al. "Search Is a Time-Critical Event: When Search and Rescue Missions May Become Futile." Wilderness and Environmental Medicine, vol. 18, no. 2, Feb. 2007, pp. 95-101, DOI: 10.1580/06-WEME-OR-035R1.1.

Bloom, Gary. Personal Interview. 12 Dec. 2024.

Bryant, Charles W. "How Search and Rescue Works." MapQuest Travel, www.mapquest.com/travel/search-and-rescue.htm.

Koester, Robert J. "ISRID-3 NPS Initial Planning Point Dataset."

"What is logistic regression?" IBM, www.ibm.com/topics/logistic-regression.

"What is machine learning?" IBM, www.ibm.com/topics/machine-learning.

Matsui, Tohgoroh. "Breast Cancer Diagnosis." Kaggle, www.kaggle.com/c/1056lab-breast-cancer-diagnosis/overview.

mikesuperman. "SHapley Additive exPlanations or SHAP: What is it?" Datascientest, www.datascientest.com/en/shap-what-is-it.

Soni, Brijesh. "Stacking to Improve Model Performance: A Comprehensive Guide on Ensemble Learning in Python." Medium, 1 May 2023, www.medium.com/@brijesh_soni/stacking-to-improve-model-performance-a-comprehensive-guide-on-ensemble-learning-in-python-9ed53c93ce28.

"Support Vector Machine (SVM) Algorithm" GeeksForGeeks, www.geeksforgeeks.org/support-vector-machine-algorithm/.

Tuia, D., Kellenberger, B., Beery, S. et al. "Perspectives in machine learning for wildlife conservation." Nature Communications, vol. 13, no. 792, 2022, doi.org/10.1038/s41467-022-27980-y.

Viswambharan, Vinay, et al. "New Pretrained Geospatial AI Models for Disaster Response." ArcGIS Blog, ESRI, www.esri.com/arcgis-blog/products/arcgis-pro/public-safety/new-pretrained-geospatial-ai-models-for-disaster-response.