Project Description

Dvora Goncharok
&
Arbel Shifman

Project: "MediGuard"

Risk classification: General, Critical

Task:

- **Input:** Free-text medication-related question (e.g., dosage, interactions, side effects)
- **Output:** Risk Level classification – General (safe) or Critical (dangerous)
- **Task Type:** Binary class text classification
  (Goal: identify level of potential clinical risk posed by the question)

Data and Evaluation:

- **Dataset:** MedInfo2019-QA-Medications (publicly available on GitHub) [Link to the Data Set]
- **Labels:** Manual annotation of ~700 examples with new Risk_Level (General / Critical)
- **Evaluation Metrics:** Accuracy, Precision, Recall, F1-score (per class), Confusion Matrix
- **Evaluation Method:** 80/20 Train-Test split, with k-fold cross-validation for robustness

Prior Art

Source / Title	Approach / Model	Data	Metrics	Results
Risk of mortality and cardiopulmonary arrest in critical patients presenting to the emergency department using machine learning and natural language processing	XGBoost	40,218 emergency department (ED) patient questions	AUROC = 0.96	High accuracy in predicting mortality and cardiac arrest within 24 hours
Identifying the Perceived Severity of Patient-Generated Telemedical Queries Regarding COVID: Developing and Evaluating a Transfer Learning–Based Solution	SBERT contextual embeddings	11,746 telemedicine queries from eConsult platform	F1 score = 0.917	Effective at classifying severe vs. non-severe queries
COMPARISON OF PERFORMANCES OF OPEN ACCESS NATURAL LANGUAGE PROCESSING BASED CHATBOT APPLICATIONS IN TRIAGE DECISIONS	GPT-4	130,974 high-acuity patient queries categorized as ESI-1 or ESI-2 (Emergency Severity Index)	F1 score = 0.899	High agreement with emergency medicine experts
Comparison of Diagnostic and Triage Accuracy of Ada Health and WebMD Symptom Checkers, ChatGPT, and Physicians for Patients in an Emergency Department: Clinical Data Analysis Study	ChatGPT 3.5 & 4.0, Ada, WebMD	40 real patient cases from an emergency department	Top-1 Match: ChatGPT 4.0: 33% Physicians: 47%	ChatGPT models showed lower diagnostic accuracy than physicians
Performance of emergency triage prediction of an open access natural language processing based chatbot application (ChatGPT): A preliminary, scenario-based cross-sectional study	ChatGPT	50 simulated ED patient scenarios	Overall F1 score = 0.461; For high-acuity cases (ESI-1/2): F1 score = 0.821, AUC = 0.846	Moderate agreement with emergency physicians; ChatGPT showed good performance in identifying high-acuity cases (ESI-1/2), but tended to under-triage and misclassify non-critical cases.
Human intelligence versus Chat-GPT: who performs better in correctly classifying patients in triage?	ChatGPT	30 simulated triage case vignettes	Sensitivity = 0.93	Triage nurses outperformed ChatGPT in accuracy across all ESI levels, but ChatGPT showed high sensitivity in detecting critical cases.

Steps and Pipeline

Pipeline Overview:

- **Input:** Free-text medication-related question (e.g., about dosage, side effects, interactions)
- **Output**: Risk level classification – General (safe) or Critical (dangerous)
- **Task Type:** Binary class text classification problem using an NLP pipeline

Preprocessing

- **Text cleaning**: lowercasing, punctuation removal
- **Manual annotation** of ~700 questions with new Risk_Level labels (General / Critical)
- **Label balancing techniques** to address class imbalance (SMOTE).

Feature Representation

- **TF-IDF** vectorization for feature extraction
- **Critical Similarity** feature generation based on TF-IDF cosine similarity
- **Dimensionality reduction (SVD)** to reduce feature space and address high feature-to-sample ratio
- **Data includes**: Question text, existing question type, drug focus, and URL source

Steps and Pipeline

Models to Compare

- **SMOTE** (Synthetic Minority Over-sampling Technique) applied to balance class distribution in training data
- Models used:
  - Logistic Regression (and SGD with regularization)
  - Random Forest
  - SVM
  - Gradient Boosting
  - KNN

Evaluation Strategy

Metrics: Accuracy, Precision, Recall, F1-Score (per class)
Method: 80/20 Train-Test split, with k-fold cross-validation for robustness

Exploration & Baseline

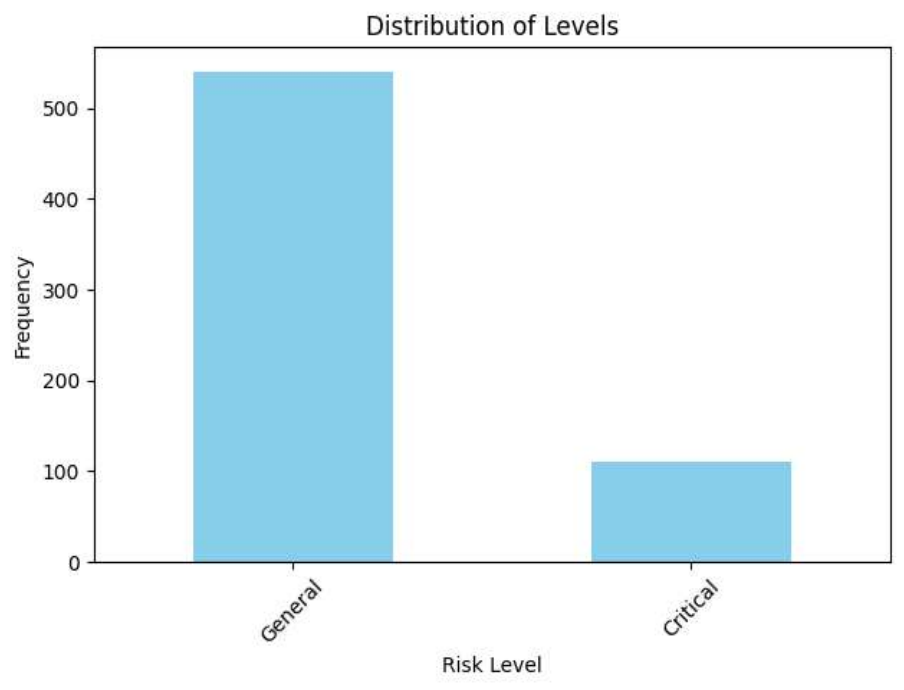
Dataset:

- Question Text + Risk Level Column:

  General: Questions associated with lower risk.

  Critical: Questions associated with higher risk that may require immediate attention.

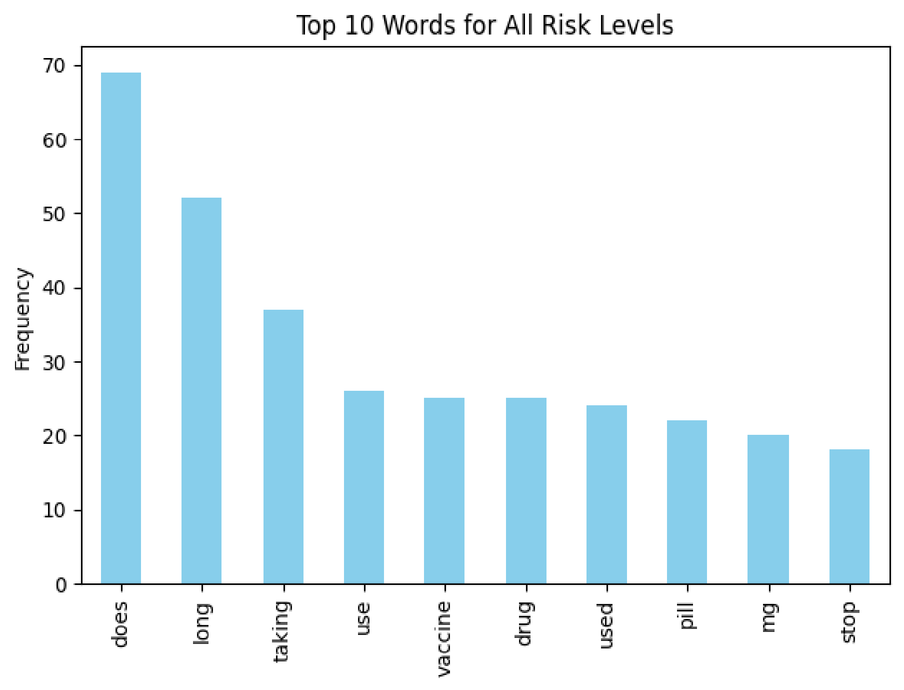
- 652 questions after cleaning and preprocessing.
- Mean question length: 50 +/- 35 words.
- **Data Imbalance:** The dataset is unbalanced, with more questions in the General category than in the Critical category.
  - **SMOTE** was used to balance the dataset by generating synthetic examples for the Critical category.



EDA Process:

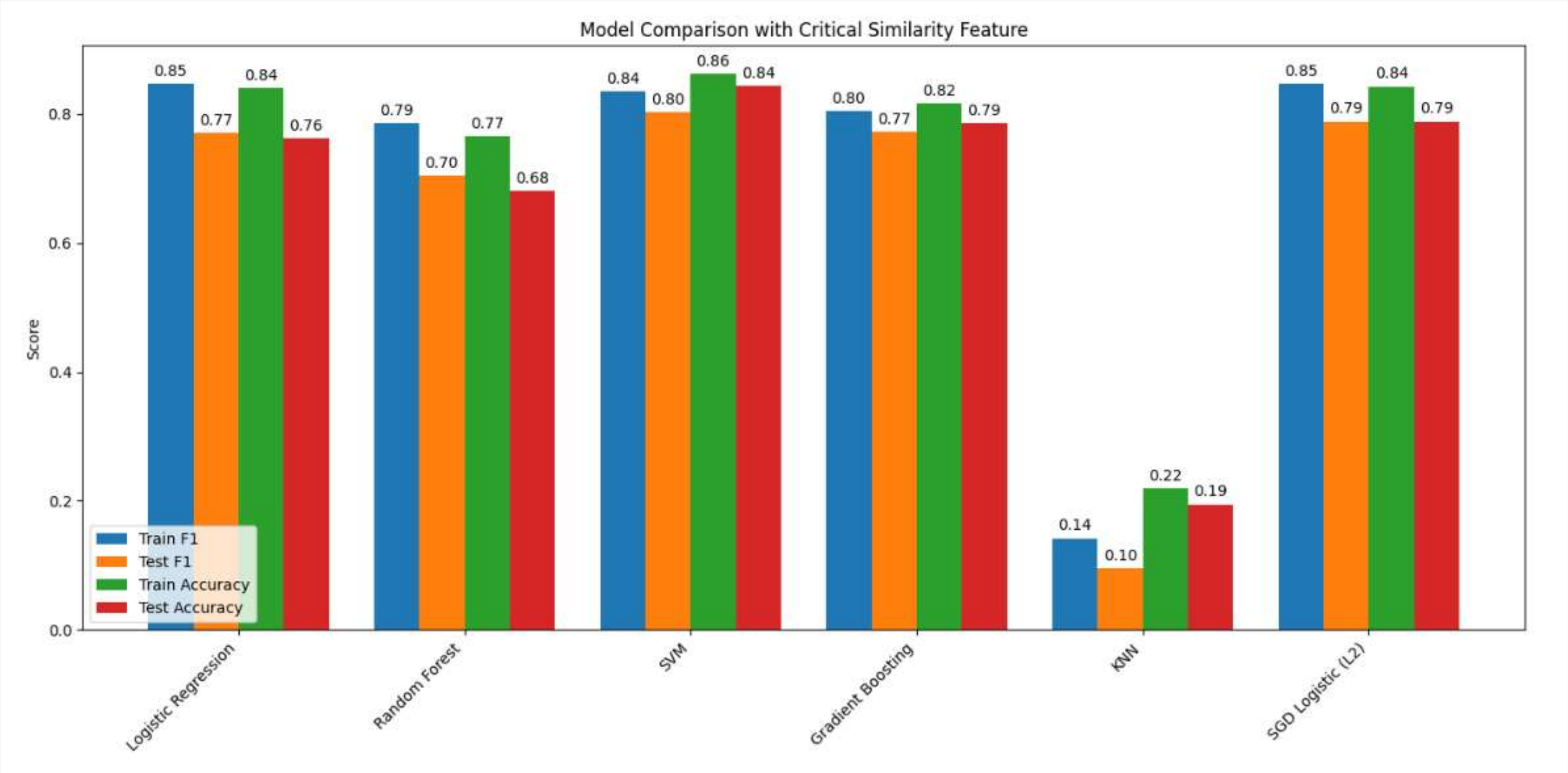
Text Preprocessing, tokenization and Vectorization:

1. **Text Preprocessing**: This step involves cleaning the text by removing irrelevant characters (Stop Words) and formatting issues.
2. **Tokenization**: The cleaned text is then split into tokens (words), which are the fundamental units of analysis.
3. **TF-IDF Vectorization**: After tokenization, the words are converted into numerical vectors using the TF-IDF technique, which helps capture the importance of each word relative to the entire dataset.
4. **Additional feature engineering** was later performed to enrich the data, including the creation of a **Critical Similarity feature.**



Baseline Model Evaluation Results:

- The key metrics used were **Accuracy, F1 Score, Precision, and Recall**.
- **SVM** showed the **best** overall performance with high accuracy and a high F1 Score, no overfitting shown.
- **SGD and Gradient Boosting** performed well with slight drop in test performance.
- **Logistic Regression** demonstrated stable performance.
- **Random Forest** showed weaker results.
- **KNN** showed poor results.
- Train vs Test Accuracy: Indicates how well the models generalize.
- F1 Score evaluates model performance by factoring both Precision and Recall.



Insights from Data Exploration (EDA)

Data Quality:

- Text is of appropriate length, but the dataset is **imbalanced** (more General questions than Critical).

Challenge:

- Difficulty **distinguishing** between categories.
    - Solution: **SMOTE** to **balance** the dataset.

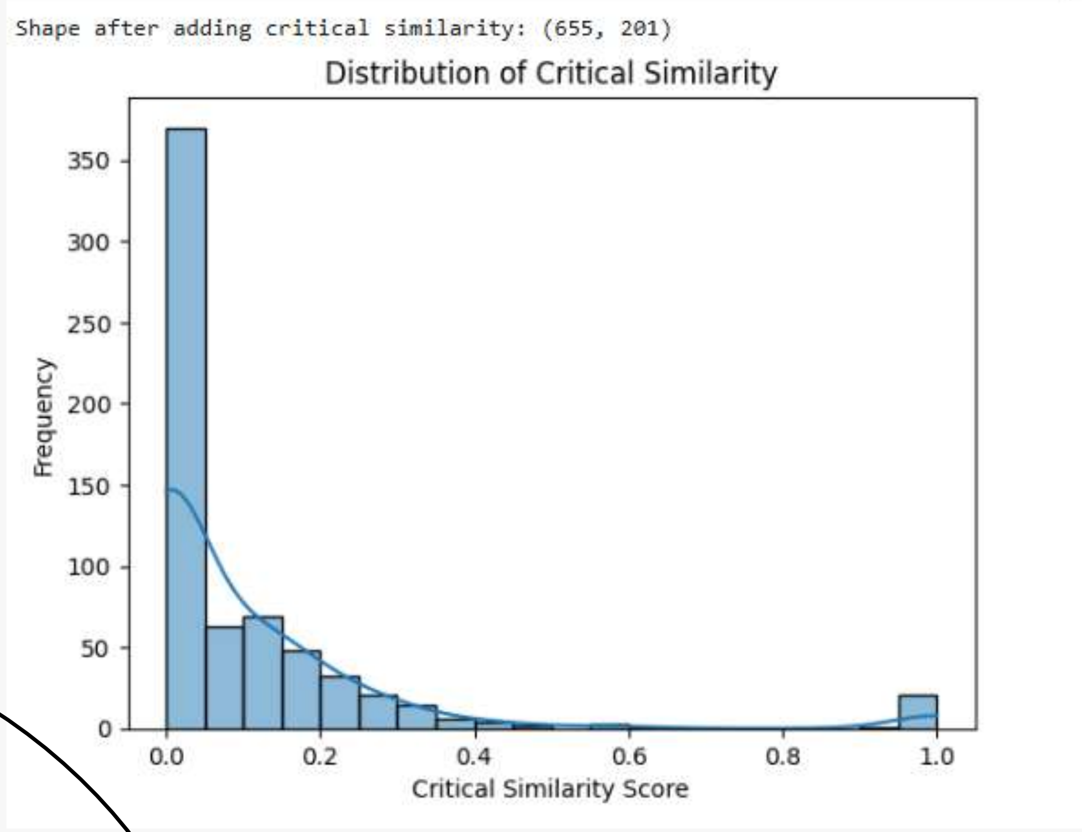
Text Preprocessing:

- **TF-IDF** was used to vectorize the questions, but the feature-to-sample ratio was high, risking overfitting.
    - Solution: Dimensionality reduction (**SVD**) to reduce features.

Feature Engineering:

- To enrich the available information, we created a new feature: **Critical Similarity**.
    - Steps:

1. Selected truly critical questions from the original dataset.
2. Represented questions and critical examples using TF-IDF.
3. Calculated **cosine similarity** between each question and the set of critical questions.
4. Assigned each question a similarity score.
5. Added the Critical Similarity feature after SVD.



Observations from Critical Similarity:
- Most questions had **low similarity** scores (0–0.4).
- A small subset showed higher similarity (≥0.4), indicating strong relation to critical questions.

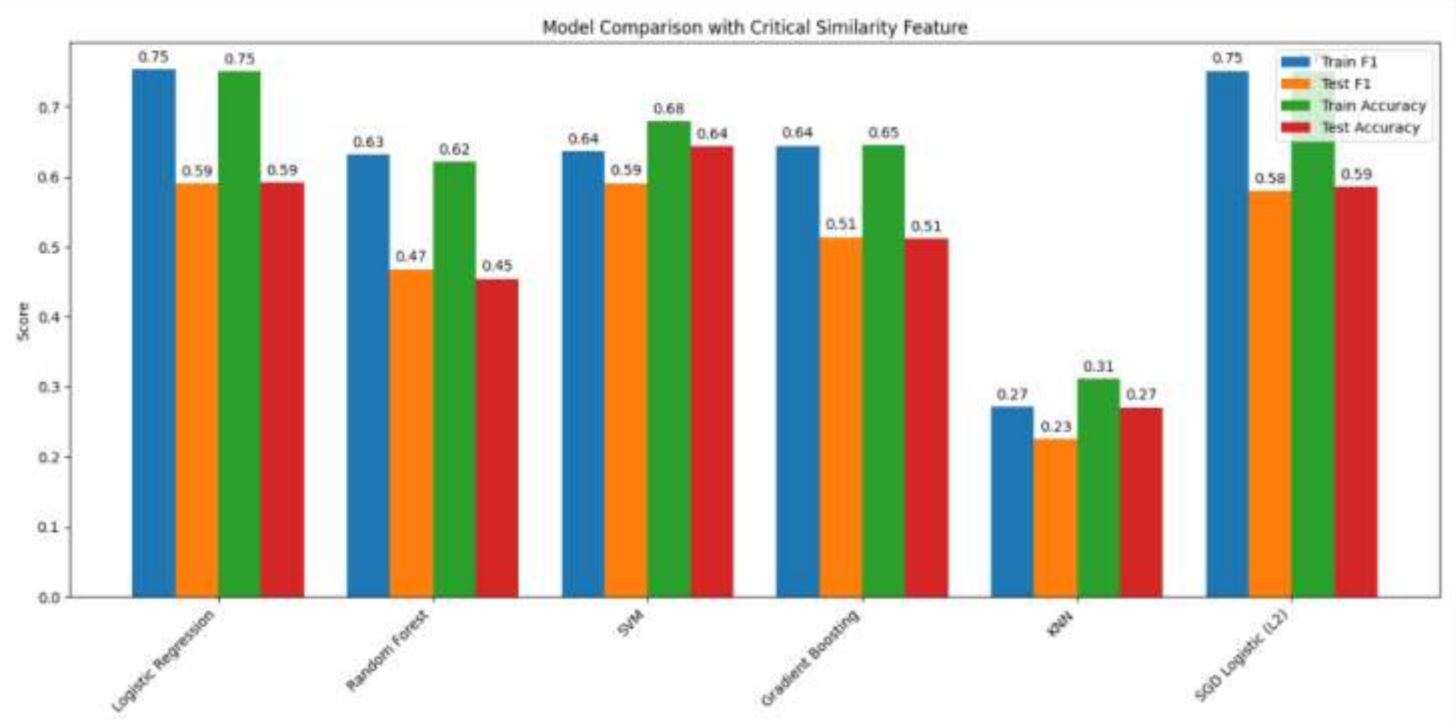
Baseline Performance:
- Most performances of F1 were with ~77% except for KNN (10%).
- Challenge:
- **Dataset imbalance**, high feature-to-sample ratio, and lack of semantic signals.
- Solution:
  - Applied **SMOTE** for balancing.
  - Used **SVD** for dimensionality reduction.
  - Enriched vectors with the **Critical Similarity** feature.
  - Experimented with ensemble learning techniques.

Conclusion:
- Adding the Critical Similarity feature provided the model with valuable semantic hints.
- Further improvement could be achieved by expanding the pool of critical examples or incorporating external medical knowledge.

Binary vs. Multi-Class Performance:

- When switching to binary classification (Critical vs. General), all models (except KNN) achieved more stable results around 77% **without signs of overfitting**.
- In contrast, multi-class classification (Critical, Personal, General) **showed significant overfitting** and lower performance (~53% accuracy).

Triangular classification:



Binary classification: