Advanced Data Science Capstone Course

Disaster Detection System Architecture Decision Document (ADD)

Instructor

Romeo Kienzler

Alireza Heidari

Fall 2023

1. Introduction

1.1. Purpose

The primary aim of this document is to detail the decisions, methodologies, and thought processes behind the design and development of the Disaster Detection System. Ensuring clarity on these choices is imperative for all stakeholders, from developers to end-users, so that there's a unified understanding of the system's foundation.

1.2. Scope

This ADD focuses on delineating the system's blueprint, specifically its architectural components, data flow, interactions, and the reasoning behind certain design choices, taking into account both traditional machine learning and modern deep learning models.

1.3. Definitions, Acronyms, and Abbreviations

- **BERT (Bidirectional Encoder Representations from Transformers):** A state-of-the-art deep learning model specifically designed for understanding context within text, by examining text from both left to right and right to left.
- **NB (Naive Bayes)**: A probabilistic algorithm that's celebrated for its simplicity and efficiency, especially in text classification tasks. It calculates the probability of a particular class given a set of features using Bayes' theorem.
- **LLM (Language Learning Models):** Advanced models trained on vast textual datasets, enabling them to capture intricate linguistic patterns. They provide a strong foundation for many NLP tasks, given their expansive training on varied textual data.
- **SGD (Stochastic Gradient Descent):** A popular optimization algorithm frequently employed in neural networks and deep learning models. It aims to find the values of parameters that minimize a loss function, updating parameters iteratively based on a subset or a single training example.
- **TF (TensorFlow):** An open-source platform developed by Google, tailored for building and deploying machine learning models.
- **F1-Score:** A metric used to evaluate the performance of binary classification systems, it considers both precision and recall to compute the score. The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and the worst at 0.

2. System Overview

2.1. Objective

The Disaster Detection System aspires to predict and identify real-time disasters by diligently analyzing Twitter data. Beyond prediction, the system serves as a pivotal tool in timely disaster response and management, making it a linchpin in proactive disaster management strategies.

2.2. Dataset Information

2.2.1. Context

The file contains over 10,000 tweets associated with disaster keywords like “crash”, “quarantine”, and “bush fires” as well as the location and keyword itself. The data structure was inherited from Disasters on social media. These tweets, with their distinct labels indicating relevance to a real disaster or otherwise, become the backbone of our natural language processing and analysis endeavors.

The tweets were collected on Jan 14th, 2020.

Some of the topics people were tweeting:

- The eruption of Taal Volcano in Batangas, Philippines
- Coronavirus
- Bushfires in Australia
- Iran downing of the airplane flight PS752

2.2.2. Schema

- **id**: A unique identifier for each tweet
- **keyword**: A particular keyword from the tweet
- **location**: The location the tweet was sent from (may be blank)
- **text**: The text of the tweet
- **target**: Denotes whether a tweet is about a real disaster (1) or not (0)

2.2.3. Inspiration

The intention was to enrich the already available data for this topic with newly collected and manually classified tweets.

Note: Dataset can be freely downloaded from here.

3. Data Quality Assessment

3.1. Rationale for Data Quality Approach

In the vast realm of data-driven systems, data quality stands paramount. The choices made in assessing this quality stem directly from the challenges and unique characteristics intrinsic to the dataset, ensuring a tailored and optimal approach.

3.2. User Input and Noise Management

The unpredictability of user inputs, particularly in the location feature, introduces significant noise to our dataset. This noise, if unchecked, could skew our analysis. However, by deploying advanced filtering algorithms and normalization techniques, this system transforms potential noise into structured, valuable insights. This is the reason we **omitted the location feature from the dataset.**

3.3. Missing Value Imputation Strategy

Data isn't just about what's present, but also what's missing. Identical missing value ratios between the datasets hint at shared sampling methodologies. To handle these absences and ensure a consistent dataset, placeholders **no_keyword** and **no_location** effectively fill the gaps without introducing bias. As far as the keyword feature is highly correlated with the target label, it is both dangerous and challenging to utilize interpolating methods.

3.4. Potential Enhancements to Data Quality Methodology

Even the most refined strategies can benefit from continuous improvement. Enhanced outlier detection or more nuanced imputation strategies could be integrated in future iterations. By continuously exploring these and other advancements, the system ensures it remains at the cutting edge of data quality.

4. Feature Engineering

4.1. Stratification Based on Keywords

Keywords serve as a lighthouse, guiding the model towards patterns. Stratifying based on this relation not only enhances training but ensures a more holistic understanding of the relationship between data features and disaster relevance.

4.2. Linguistic Features

The tweets' linguistic features provide essential insights. Features such as word_count, unique_word_count, stop_word_count, url_count, mean_word_length, char_count, punctuation_count, hashtag_count, and mention_count are crucial to understand the relevance of our textual features to the target label. They can potentially assist the model in understanding the structure and semantics of the tweets, enhancing prediction accuracy.

4.3. Data Cleaning

Tweets, by nature, are informal and often cluttered. Cleaning operations are performed to counteract this:

- special characters are removed
- contractions expanded
- URLs deleted
- character entity references substituted with actual symbols
- slang and typos rectified
- informal abbreviations expanded
- and certain words grouped or abbreviated for consistency

Over 600 regex strings were used to achieve the above cleaning.

5. Algorithm Choice

5.1. BERT (Bidirectional Encoder Representations from Transformers)

BERT (Bidirectional Encoder Representations from Transformers) is a cutting-edge deep learning model designed for natural language processing tasks. One of BERT's standout features is its training objective, known as the "**masked language model**" (MLM) approach. In MLM, random words from a sentence are masked out, and the model is tasked with predicting these masked words based on their surrounding context. This bidirectional understanding, examining text from both left to right and right to left, allows BERT to capture nuanced linguistic patterns from vast amounts of text. Through this training approach, BERT develops a profound grasp of context within language.

5.2. Rationale for BERT Algorithm Selection

Training deep classifiers like BERT requires vast data, making it infeasible given our dataset's size. Therefore, we chose to fine-tune BERT, a pre-trained Language Model (LLM). This decision, driven by practical and performance considerations, allowed us to harness BERT's extensive linguistic knowledge. Empirical studies showed BERT vastly outperforms traditional algorithms like Naive Bayes, validating our choice. **Fine-tuning BERT is thus a superior strategy**, especially when faced with limited data.

5.3. Advantages Over Traditional ML Algorithms

BERT's pre-training on massive text datasets enables it to achieve higher accuracy, making it more effective than algorithms like Naive Bayes, which assumes features are independent.

6. Framework Selection

6.1. Advantages of Tensorflow 2.0

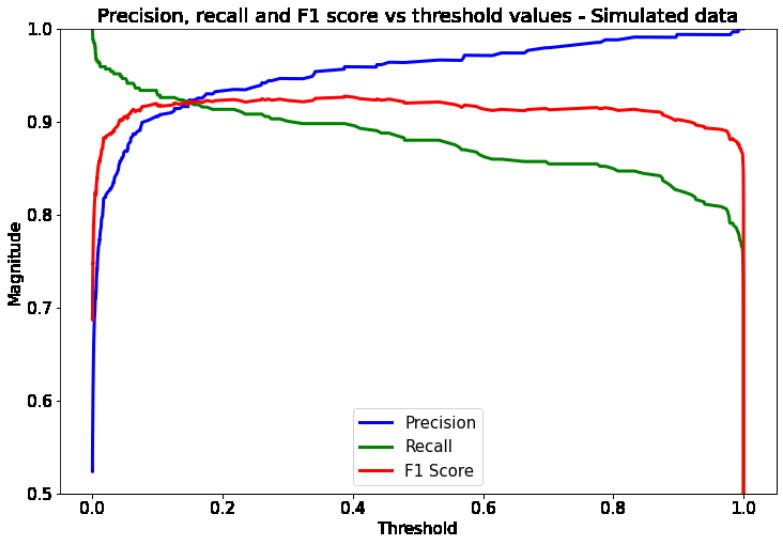
Tensorflow 2.0, compared to alternatives like PyTorch, offers various benefits:

- **Easier Learning Curve:** TensorFlow 2.0 integrates Keras as its high-level API, simplifying the modeling process.
- **Optimized Performance:** It's tailored for both beginners and experts, supporting advanced operations while remaining user-friendly.
- **Distributed Training:** Tensorflow 2.0 supports distributed training, allowing models to scale across multiple GPUs and TPUs seamlessly.
- **Robust Ecosystem:** With a vast community and support from Google, TensorFlow provides a plethora of resources, including TensorBoard for visualization.
- **Open-source Nature:** TensorFlow 2.0 is an open-source framework, granting developers access to a plethora of tools and ensuring transparent, community-driven development.

7. Model Performance Indicators

7.1. Choice of Metrics

Given the nature of this classification task, coupled with a slight class imbalance, relying solely on accuracy would be misleading. As a result, the **Mean harmonic F1-score is selected as the primary evaluation metric.** The accompanying metrics - Accuracy, Precision, Recall, and the F1-Score - provide a holistic view of model performance. Given the minor imbalance, the F1-score, the harmonic mean of Precision and Recall, becomes indispensable for an unbiased evaluation.



7.2. Loss Function Selection

For this classification problem, Cross Entropy Loss is the chosen loss function. This loss function, when paired with the aforementioned metrics, enables optimal model structure selection and hyperparameter tuning.

8. Future Directions and Recommendations

In the dynamic world of data analytics and disaster prediction, continuous evolution is the key to success. To ensure our system remains at the forefront of technological advancements, several enhancements are proposed:

- **Exploration of Advanced Feature Engineering Methods:** While our current feature set provides meaningful insights, there's always room for enhancement. Diving deeper into alternative feature extraction and transformation techniques can unveil hidden patterns and correlations in the data, potentially improving the predictive capability of the system.
- **User Tweet History Analysis:** A user's historical tweets can provide context and aid in understanding the veracity and urgency of their current tweets. Incorporating an analysis of user tweet patterns and history can add another layer of precision in disaster detection.
- **Location Data Utilization:** The present system doesn't fully leverage the location data due to its noisy nature. Future iterations should investigate methods to make effective use of this data, as geographic context can be crucial in disaster prediction.
- **Improved Imputation Techniques:** While the current method for imputing missing keywords and locations is effective, researching advanced imputation strategies could yield even more accurate and insightful results.
- **Ensemble Learning:** Combining predictions from different models can often lead to better accuracy and robustness. Exploring ensemble methods might boost the system's performance further.
- **Metaphor Analysis:** Disasters and emergencies often use metaphorical language. Establishing a corpus of metaphors and integrating it into the system can refine the detection mechanisms, especially for tweets that don't directly reference disasters.
- **Integration with Diverse Data Sources:** To achieve a holistic analysis, the system can be integrated with other relevant data sources, potentially offering a richer contextual understanding of emerging threats.
- **Real-time Adaptive Learning:** As the data landscape changes, a system that learns and adapts in real-time will always have the edge. Investigating mechanisms to continuously update the model can keep it attuned to the latest trends and patterns.

9. Conclusion

The Disaster Detection System's architecture and decisions, as laid out in this document, are not just a reflection of rigorous technical choices but also mirror the tangible real-world implications they can address. By rooting our decisions in robust data quality, meticulous feature engineering, and the discerning selection of algorithms and frameworks, we've positioned the system for peak performance. But beyond its technical prowess, the system's real merit lies in its vast **applications**:

- **Emergency Response Coordination:** In the wake of natural calamities or other crises, our system can provide real-time insights to rescue and humanitarian agencies, enabling them to mobilize resources more efficiently.
- **Public Awareness and Safety:** By identifying potential disasters early, local governments and communities can receive timely alerts, aiding in evacuation or other necessary precautions.
- **Infrastructure Preparedness:** For urban planning and infrastructure development authorities, understanding potential disaster patterns can guide the development of more resilient structures and cities.
- **Insurance and Risk Assessment**: Businesses, especially insurance companies, can leverage the system's predictions to assess risks better and draft policies more in line with potential threats.
- **Research and Academic Studies:** Scholars studying disaster patterns, urban development, or social responses can utilize the system's predictions as a data source for deeper analysis.

As the digital and physical landscapes evolve, so too will our system, always pushing the boundaries for heightened accuracy in disaster prediction and broader application in safeguarding and improving human lives.