庭子アバター v01.60a WIP中

Natural Language Processing with Disaster Tweets

Predict which Tweets are about real disasters and which ones are not

Getting Started · 806 Teams · Ongoing

災害ツイート分類機械学習モデル最終構築・評価レポート

純粋な言語モデルによるトップティア精度の実現と最適化の軌跡

[Target] Kaggle Pu rd

Score: 0. 0

[Status] Non ur

庭師

- Gemini Pro, Antigravity, Notebook LM
- 自PC(Core i7-11700K+RTX3060(12GB))

NotebookLM

AI主導のモダンなML開発ワークフローとコスト構造

Phase 1: EDA (探索的データ解析)

Tool: Google Gemini 3.5 Flash

eda_report.md
(統計・グラフの
自動化)

Phase 2: Strategy (戦略立案)

Tool: Google Gemini Deep Research

deep_research.md
(Kaggle解法調査
と戦略化)

Phase 3: Execution (自律実行)

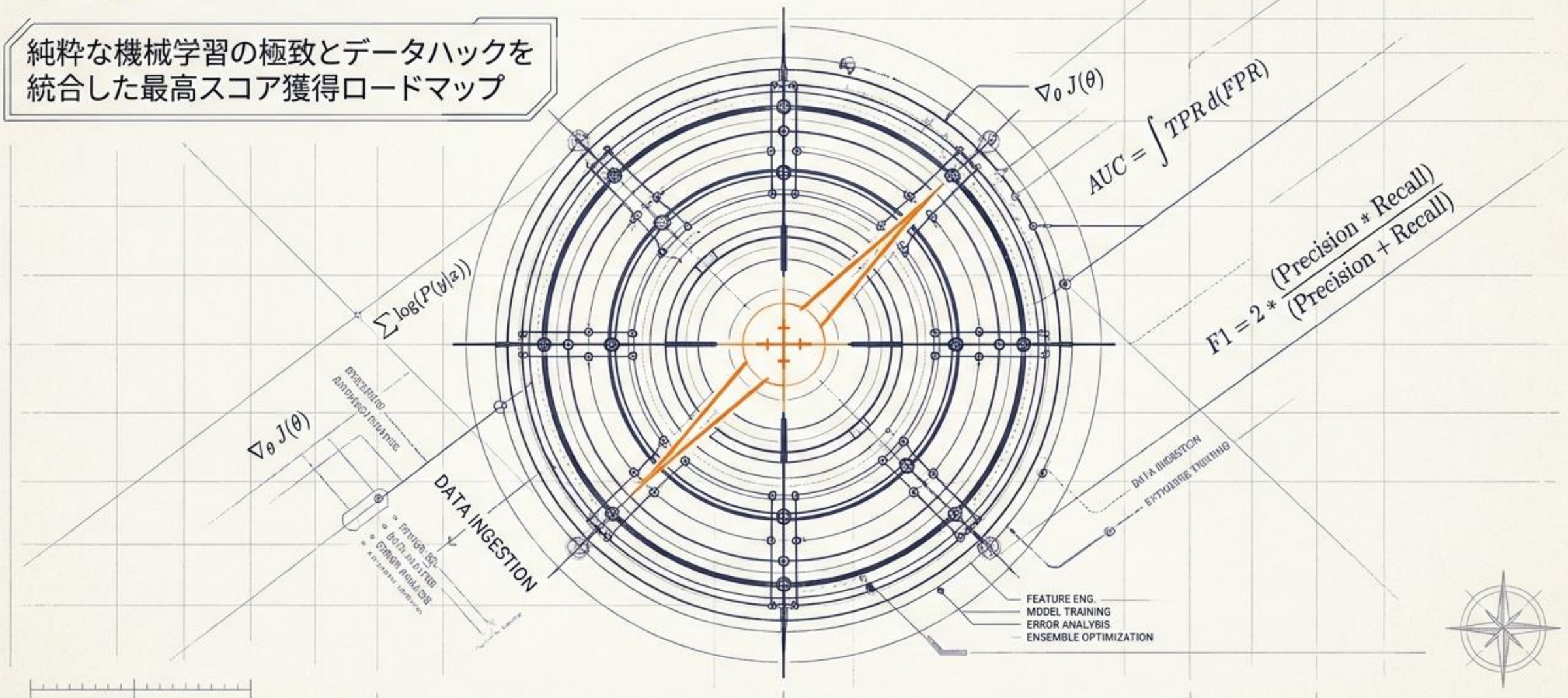
Tool: Google Antigravity

deep_research.mdをインプット
し、パイプラインを自動構築

Total Tool Cost: Google Antigravity / Google Gemini Pro (2,900円/月)
最低限のコストで最高峰のKaggleエンジニアリングを再現。

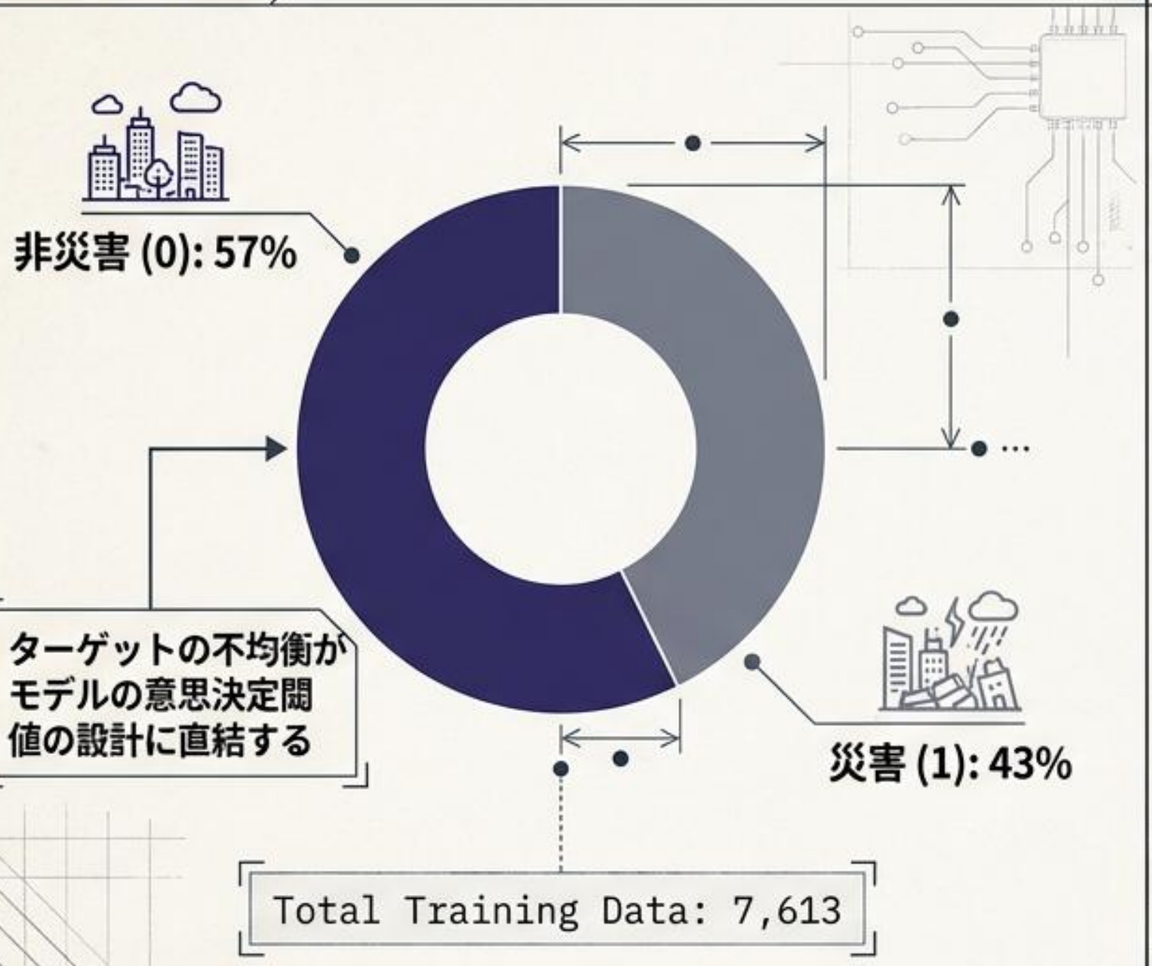
災害ツイート分類タスクにおける完全勝利への戦略設計図

純粋な機械学習の極致とデータハックを統合した最高スコア獲得ロードマップ

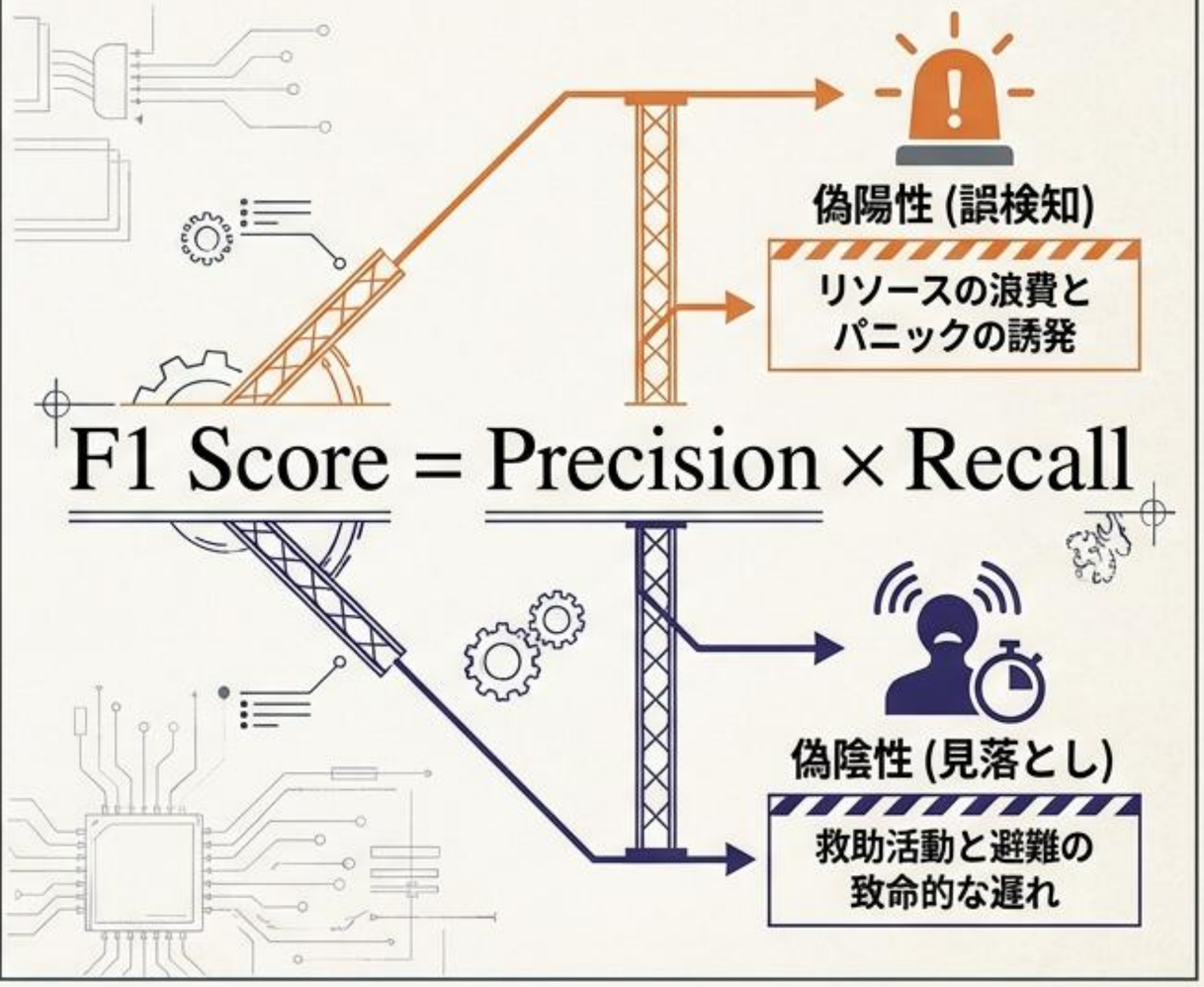


誤検知と見落としのトレードオフが人命とリソースを左右する

DATA PROFILE

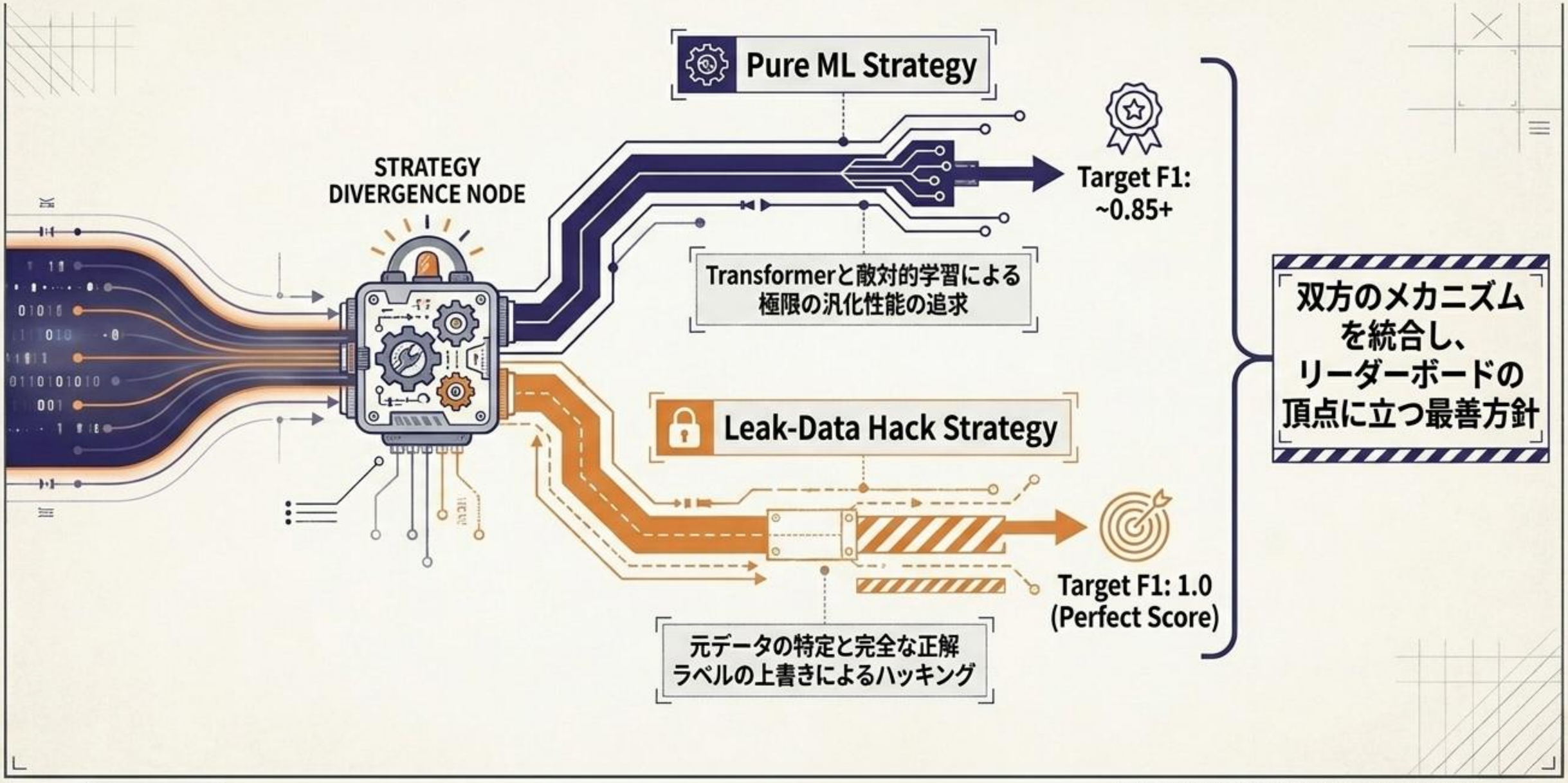


METRIC MECHANICS



極めて精緻な文脈理解なしでは、PrecisionとRecallの調和は成立しない。

最高スコアへの道は「純粋な汎化性能」と「正解の掌握」の2つの軌跡に分かれる



アノテーターの主観による「ラベル矛盾」を多数決フィルターで浄化する

1 Input (汚染されたデータ)

'11-Year-Old Boy charged'

[ラベル: 1] [ラベル: 0]

フロー (Flow)

2 Calculation (計算層)

一意テキストのラベル平均値

0.0 または 1.0 からの乖離を検知

0.5

3 Purification (浄化層: Majority Voting)

多数決
多数決 (Majority Voting) に基づき、支配的なラベルへ強制統一

精製データ (Refined Data)

不整合が激しい異常サンプルとして訓練セットから完全排除

破棄 (Discard)

勾配計算の混乱を防ぎ、モデルの最適化軌道を安定させる必須の前処理。

伝統的MLからTransformerへの前処理パラダイムシフト

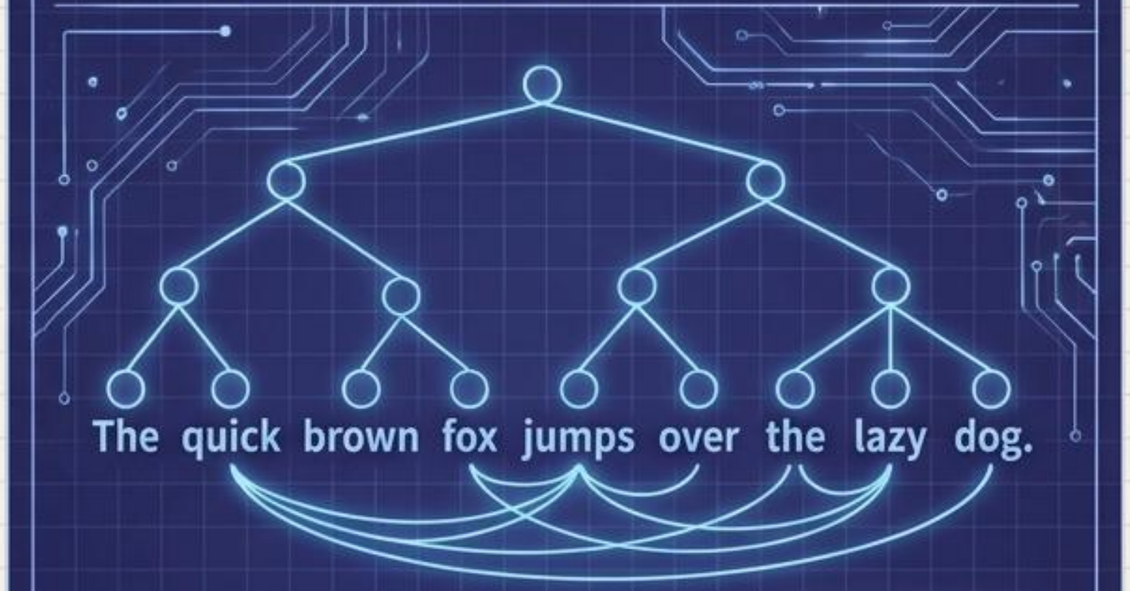
⚠ 過去の常識 (TF-IDF等)

ストップワードの除去

ポーターステマーによる語幹化

文脈とセマンティクスの破壊

現代の正解 (Transformer)



HTMLエンティティのデコード

URLの無効化 (<url>トークン置換)

短縮形の復元 (won't → will not)

not good

自己注意機構のための文脈の完全保存

自己注意機構の相互作用を最大化する入力シーケンスの設計

キーワードと本文を物理的に結合し、Transformer内部で強力な相互作用を発生させる。

[keyword]
(欠損時は特殊トークン)

[SEP / 特殊トークン]

[クレンジング済み text]

Model_Input

災害: 1

名詞

固有表現
(NER / 地名・組織)

数値

非災害: 0

代名詞 (I, you)

感情的形容詞

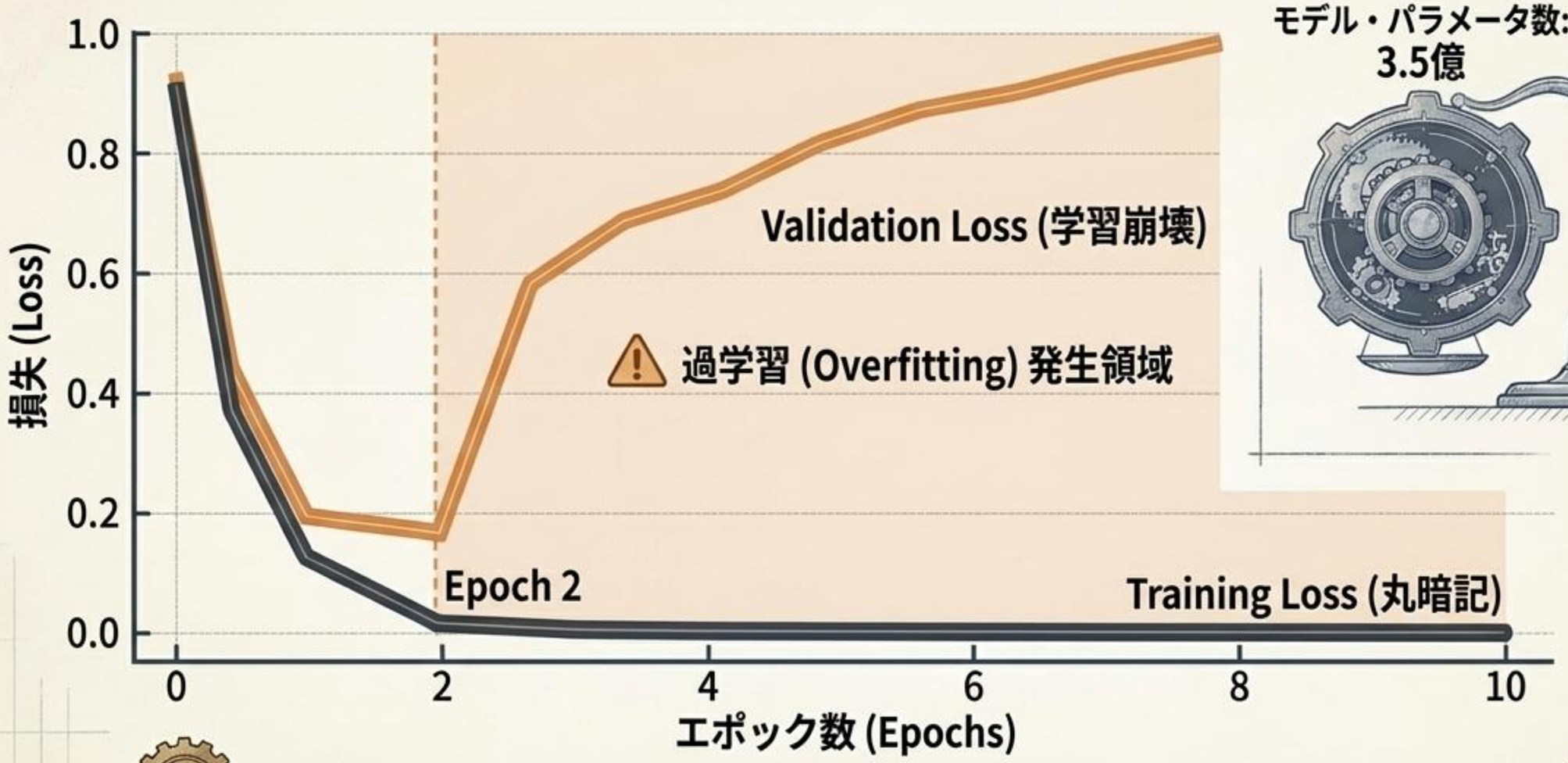
⚠ 災害を示すツイートと非災害ツイートには明確な構造的・品詞的差異が存在する。

TECHNICAL MONOGRAPH

モデル進化マトリクス：なぜ「DeBERTa-v3-large」が絶対的最適解なのか

TECHNICAL MONOGRAPH

巨大パラメータモデルのジレンマ：丸暗記による学習崩壊の恐怖

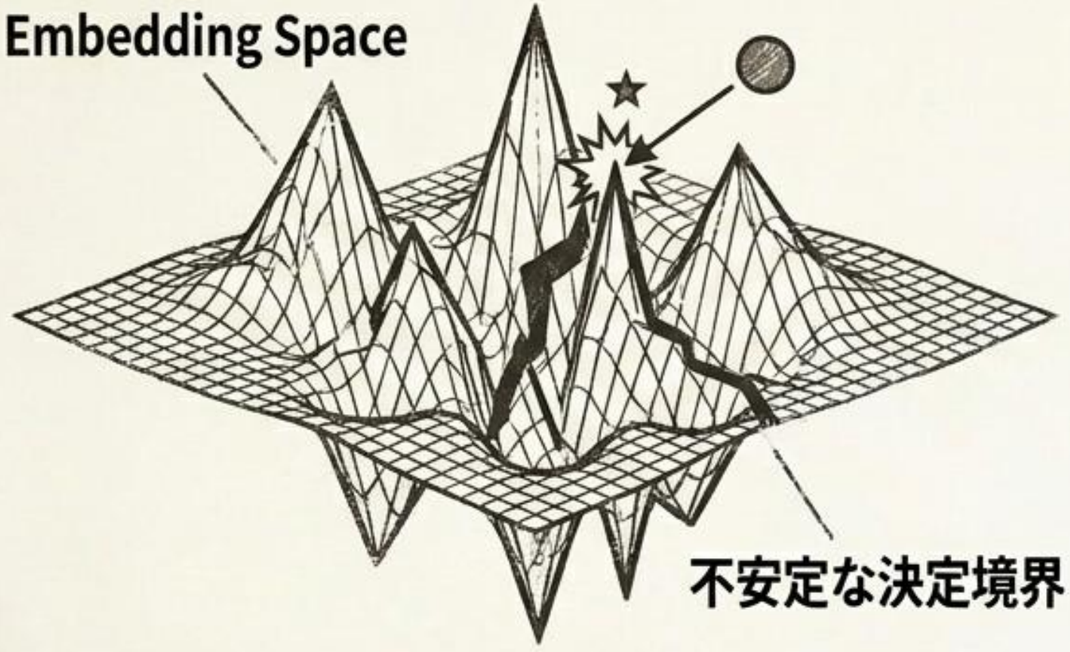


DeBERTa-v3-largeをそのままナイーブに学習させれば、データセットを即座に丸暗記し、未知のデータに対する汎化能力が数エポックで崩壊する。

TECHNICAL MONOGRAPH

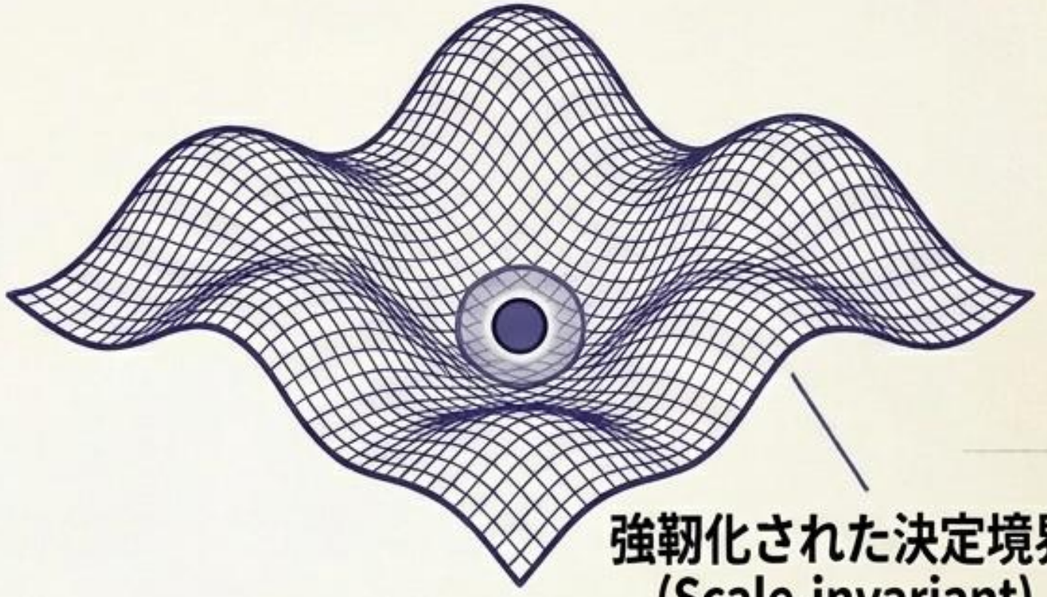
敵対的学習「SiFT」による決定境界の強靭化とノイズ耐性の獲得

SiFTなし



未知のノイズ（スペルミスや固有名詞）に
衝突し誤分類を引き起こす。

SiFTあり



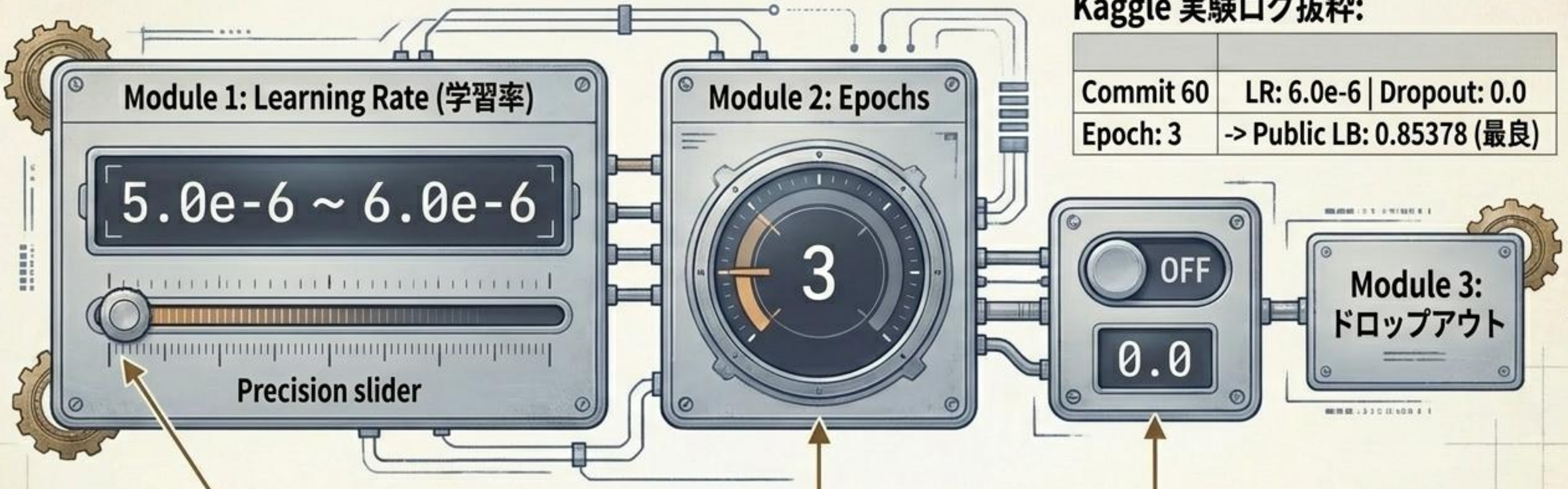
微小な敵対的ノイズ (δ) を注入しながら学習し、
意味情報を歪めずに自己防衛力を付与。

$$\min L_{\text{total}} = L_{\text{ce}} + \gamma \max_{\|\delta\| \leq \epsilon} L_{KL}$$

Scale-invariant Fine-Tuning (SiFT) は、予測のロバスト性を極限まで高める。

TECHNICAL MONOGRAPH

コミット履歴が証明する「超低速ファインチューニング」の黄金律



Kaggle 実験ログ抜粋:

Commit 60	LR: 6.0e-6 | Dropout: 0.0
Epoch: 3	-> Public LB: 0.85378 (最良)

学習スピードを
極限まで抑え込む

過学習を防ぐため、
3エポックで寸止めする

ドロップアウトを
追加して逃げない

**反直感的な「超低速ファインチューニング」こそが
Transformerの学習崩壊を防ぐ鍵である。**

Exploratory Data Analysis Report

NLP with Disaster Tweets

探索的データ解析から導く、特徴量設計とモデリング戦略

ノイズの海からシグナルを抽出し、言語の「文脈」をハックする。

Executive Summary: 3つの重要インサイト

データ品質の課題 (Data Quality)

location カラムの約33%が欠損。表記揺れも激しく、そのままの使用は**危険**。 また、ラベルが矛盾する「**競合データ**」が18件存在し、クリーニングが必須。

有効なメタ特徴量 (Meta-Features)

ツイートの文字数・単語数自体よりも、「**URLの有無**」と「**メンションの有無**」が**強力な分類シグナル**として機能する。

推奨アプローチ (Modeling Strategy)

単語の出現頻度だけでは「比喩表現（例: aftershock）」に騙される。文脈を理解できる**事前学習済みTransformer（BERT 等）**の導入がカギ。

Data Quality Matrix: データセットの全体像

Data Shape

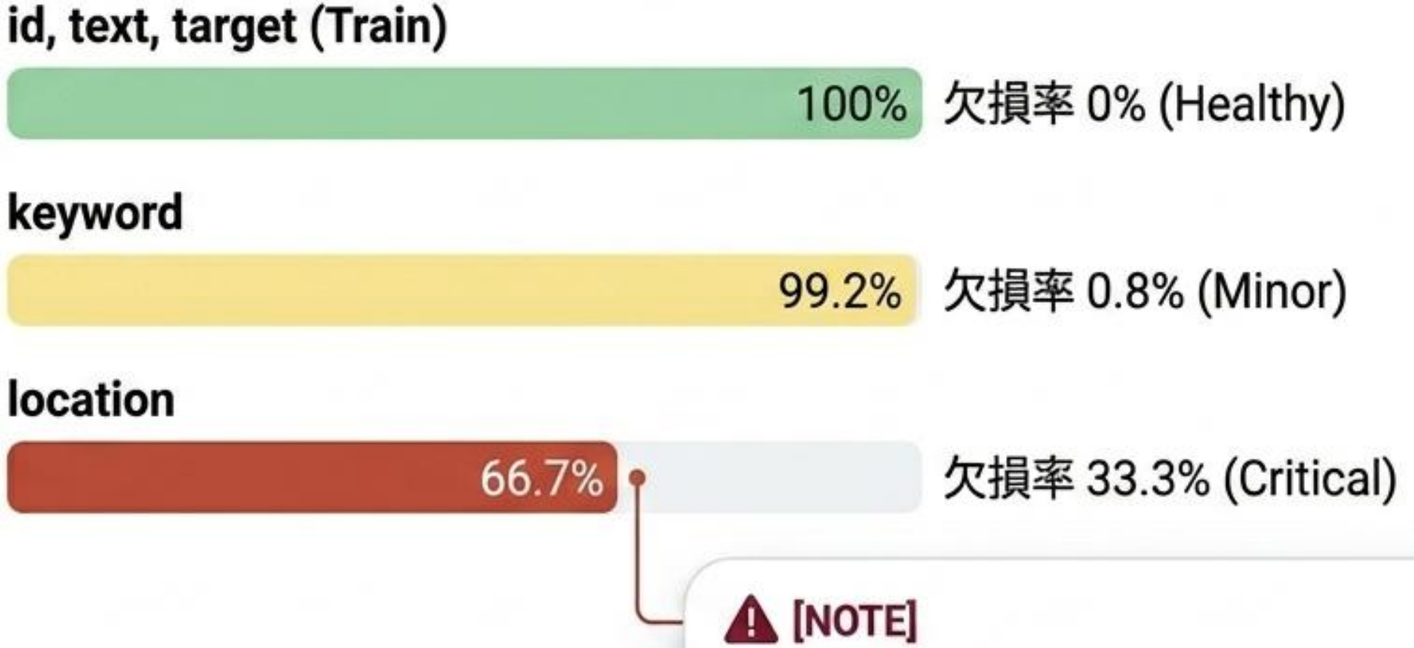
Train

7,613 Rows / 5 Columns

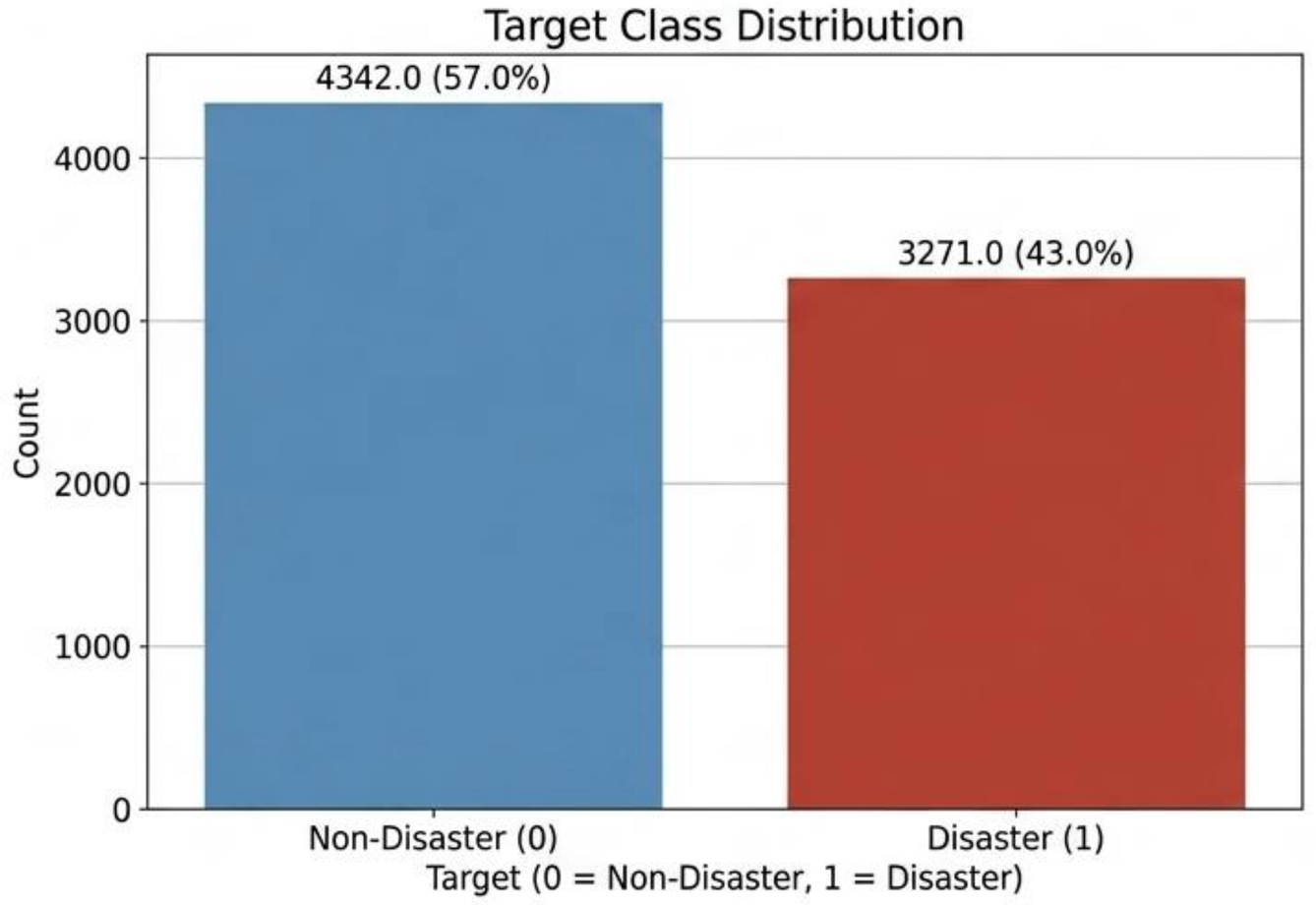
Test

3,263 Rows / 4 Columns

Column Health & Missing Values



Target Baseline: ターゲット分布と精度基準

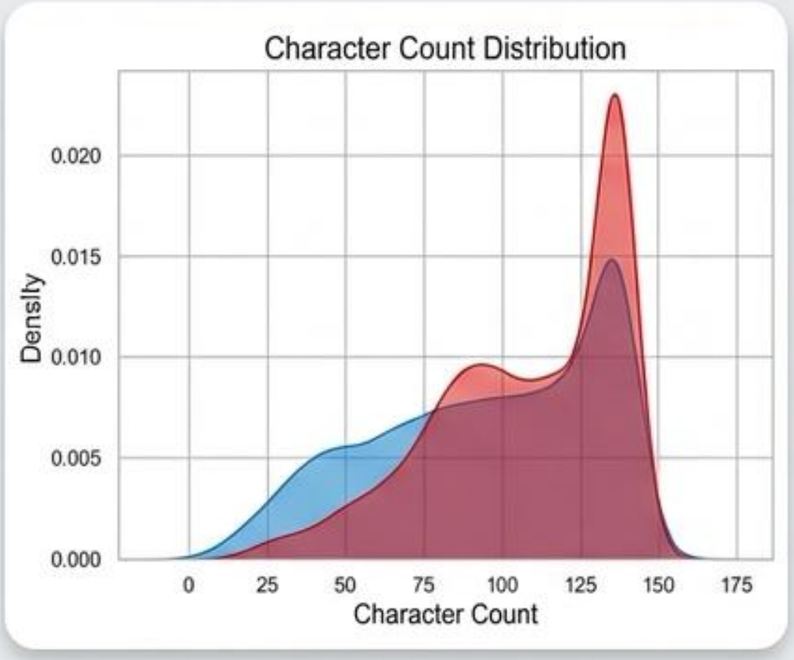


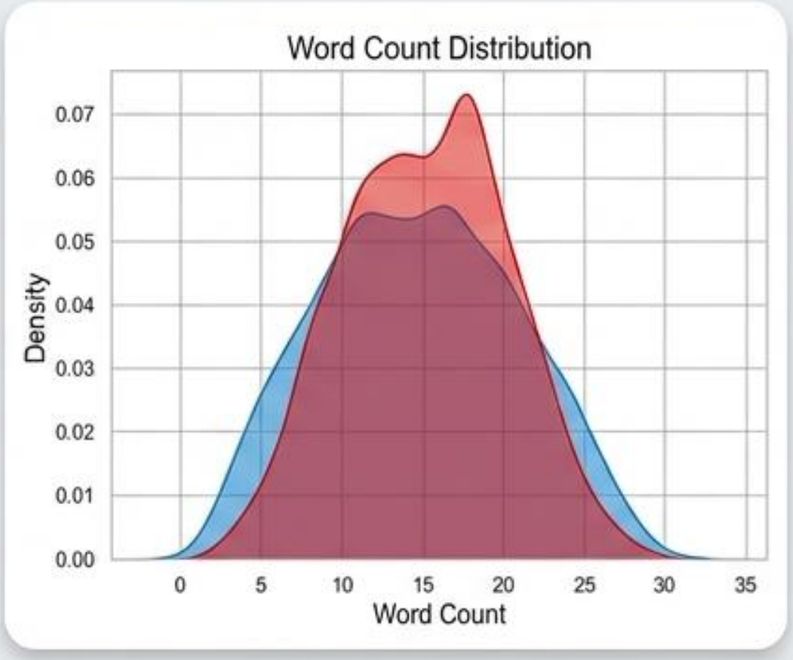
Baseline Accuracy

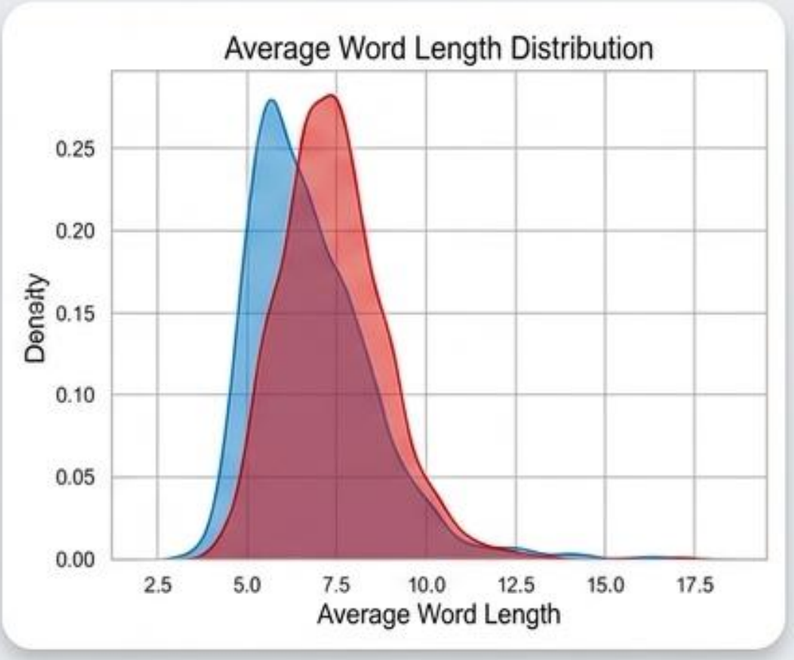
57.0%

- 多数決分類器（すべてTarget 0と予測）によるベースライン精度は57.0%。
- クラスバランスは比較的良好。極端な不均衡データ対策（SMOTE等）は不要と推測される。

Anatomy of Tweets: テキスト構造の比較







災害=108.1文字 / 非災害=95.7文字

災害が長い

災害=15.2単語 / 非災害=14.7単語

ほぼ同等

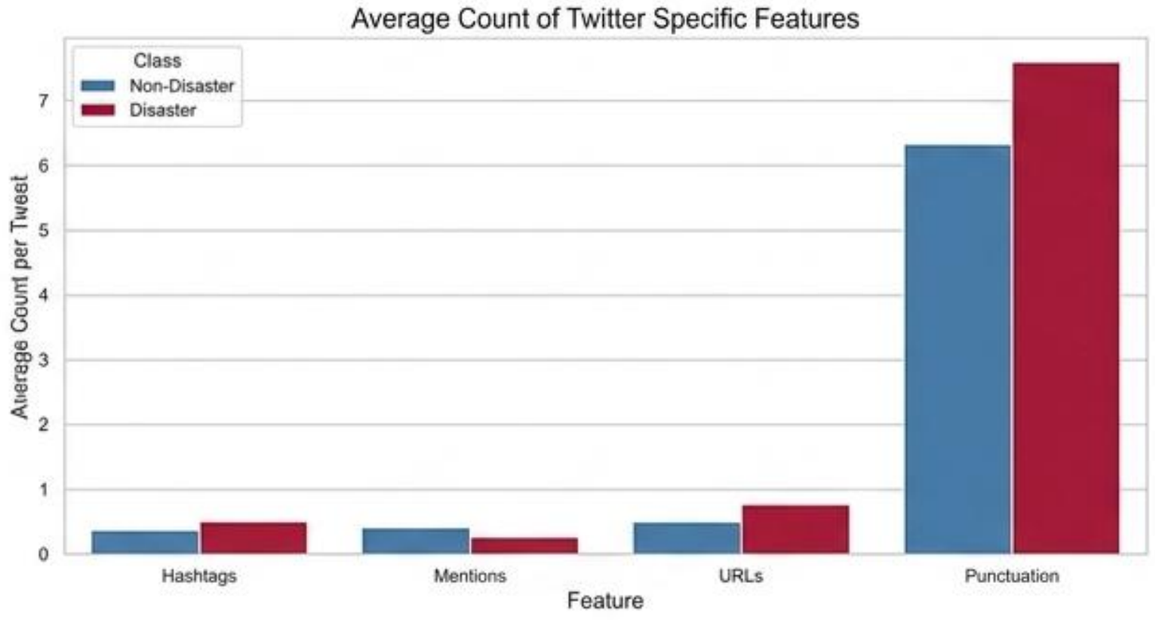
災害=7.40文字 / 非災害=6.79文字

災害が長い

なぜ単語数は同じなのに文字数が長いのか？

💡 **【URLの含有率】ニュース共有のための長いURLが平均文字数を押し上げている。**

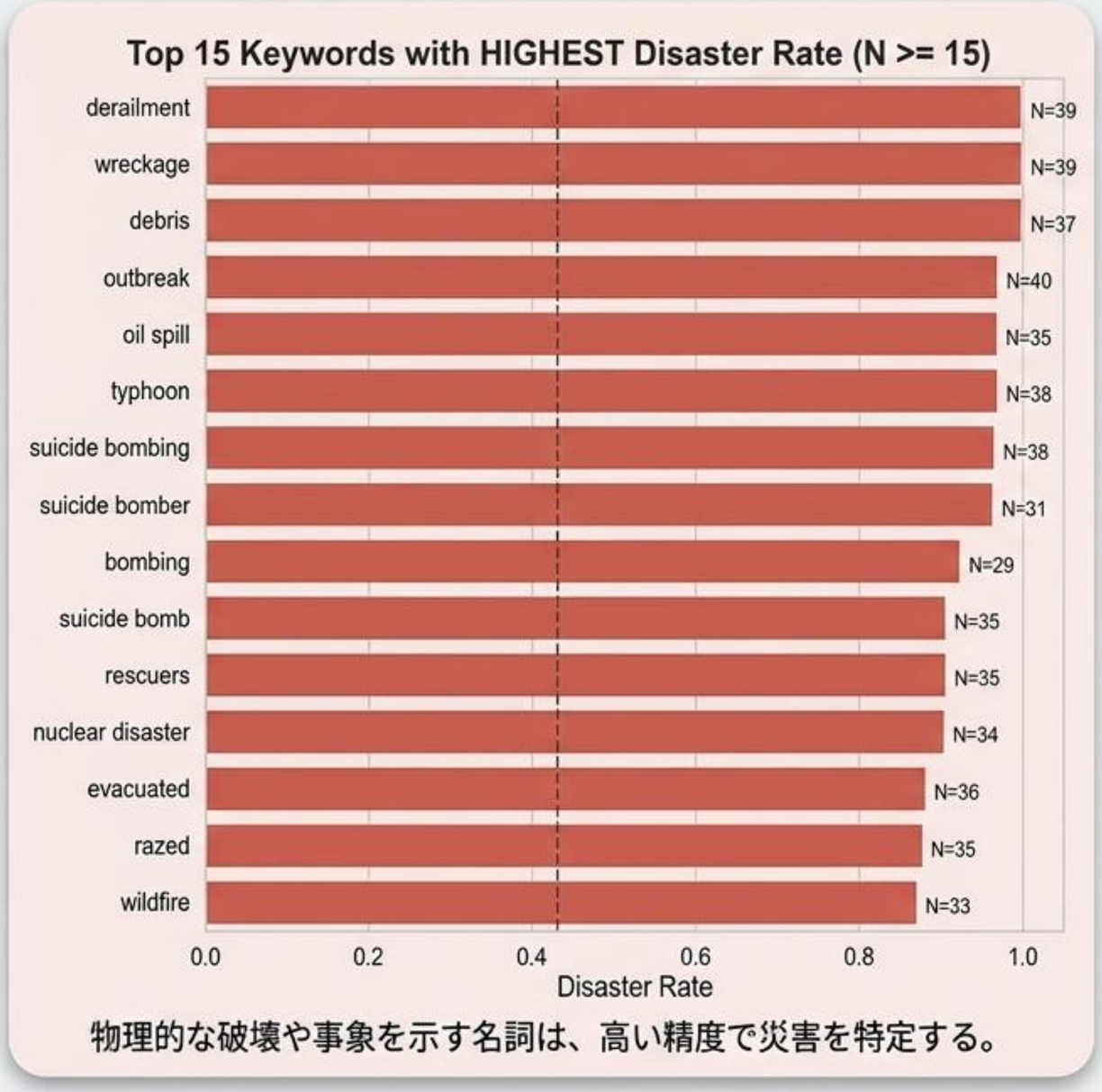
Twitter Meta-Features: 強力な識別シグナル



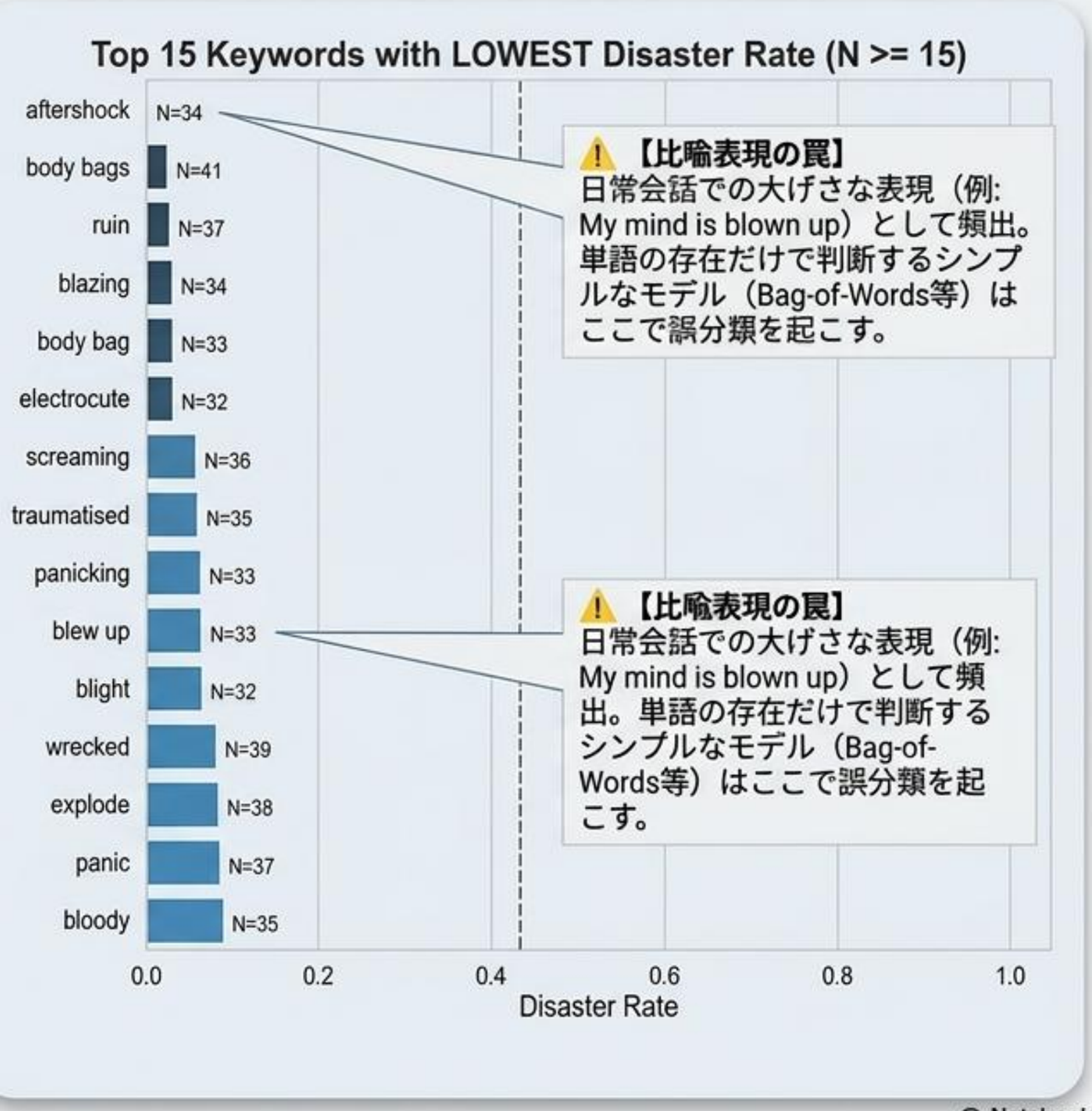
Hashtags	Mentions	URLs	Punctuation
0.50(赤) vs 0.39(青)	0.27(赤) vs 0.42(青)	0.77(赤) vs 0.51(青)	7.54(赤) vs 6.31(青)
微弱な災害シグナル	強力な非災害シグナル（個人的会話）	強力な災害シグナル（ニュース共有）	微弱な災害シグナル

URLとメンションのカウント数は、テキスト本体を解析する前の強力なメタ特徴量 (Meta-Features) としてモデルに組み込むべきである。

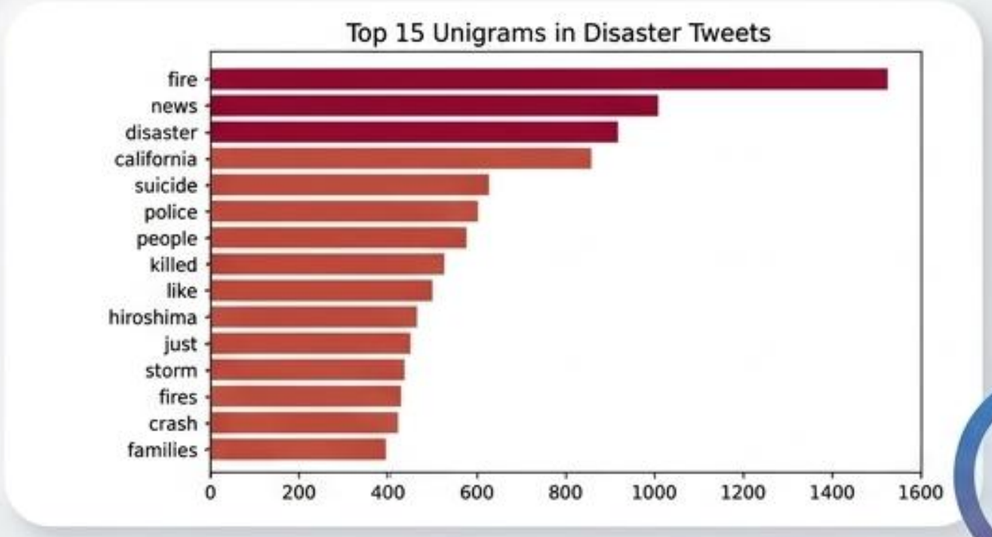
Keyword Extremes: 災害を決定づけるキーワードと落とし穴

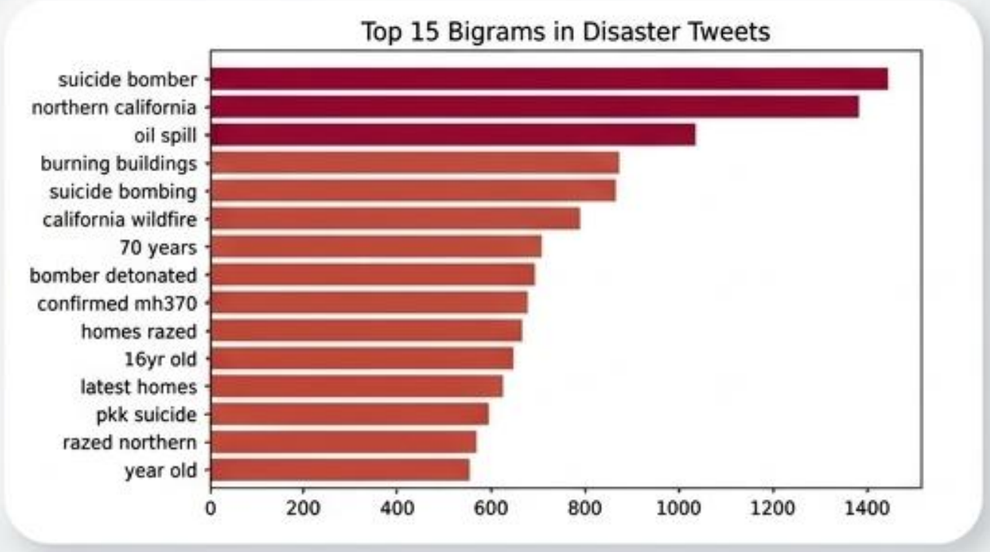


物理的な破壊や事象を示す名詞は、高い精度で災害を特定する。



N-gram Context: 文脈がもたらす解像度





解像度向上

Unigram (1単語)

単体の名詞の羅列。大まかなトピックは分かるが、具体性に欠ける。

Bigram (2単語)

単語が組み合わさることで、『自爆テロ』『北カリフォルニア（の山火事）』など、明確な災害・事件のコンテキストが浮かび上がる。

非災害ツイートでも同様に、youtube video など娯楽の文脈が明確化される。単語間の関係性を捉えることが分類精度の鍵となる。

The Data Detective's Dashboard

The Hidden Trap: ラベルの競合とデータリーク

Raw Data

179行の重複テキストを発見（69のユニークテキスト）

1個ツイート

Target = 1（災害）

18件

ラベルが完全に競合している重複テキストが18件（約40〜50行分）存在する。

Target = 0（非災害）

【学習の阻害要因】
これらはモデルの評価にノイズをもたらす。学習前に多数決でラベルを統一するか、競合行を完全にドロップするクリーニング処理が不可欠。

Synthesis: Feature Engineering Blueprint

Data Preprocessing & Engineering Pipeline

Step 1: メタ特徴量の抽出 (Extract)

- テキストをクリーニングする前に、URL数、メンション数、文字数などをカウントし、別カラムとして保存。（スライド6の知見）

Step 2: データクリーニング (Clean)

- 競合する18件のラベルを処理。（スライド9の知見）
- テキスト内のURLやHTML特殊文字 (&等) を除去。

Step 3: キーワード処理 (Process)

- keyword カラムのURLエンコード (%20等) をデコードし、欠損値を補完。
- ターゲットエンコーディングの適用を検討。

Actionable Strategy: 推奨モデリングアプローチ

Baseline Approach （高速・解釈性重視）

TF-IDF + Logistic Regression / LightGBM

抽出したメタ特徴量（URL数など）をテーブルデータとしてて扱いやすく、ベースライン（57.0%）を迅速に超えるための初期モデルとして最適。

Advanced Approach （コンテキスト理解・精度重視）

推奨

Transformerベースモデル (BERT / DeBERTa)

EDA最大の発見である「比喩表現（aftershock等）の罠」を回避するため。単語の出現有無（BoW）ではなく、周囲の単語との関係性から「文脈」を解釈できるTransformerが圧倒的に有利。

データの構造的特徴（メタデータ）と意味的文脈（Transformer）のハイブリッドが、このタスクの最適解である。

災害ツイート分類機械学習モデル最終構築・評価レポート

純粋な言語モデルによるトップティア精度の実現と最適化の軌跡

[Target] Kaggle Public Leaderboard

Score: 0.84400 Top 1-2% Equivalent

[Status] Non-Leak Pure Model

AI主導のモダンなML開発ワークフローとコスト構造

Phase 1: EDA (探索的データ解析)

Tool: Google Gemini 3.5 Flash

eda_report.md
(統計・グラフの
自動化)

Phase 2: Strategy (戦略立案)

Tool: Google Gemini Deep Research

deep_research.md
(Kaggle解法調査
と戦略化)

Phase 3: Execution (自律実行)

Tool: Google Antigravity

deep_research.mdをインプット
し、パイプラインを自動構築

Total Tool Cost: Google Antigravity / Google Gemini Pro (2,900円/月)
最低限のコストで最高峰のKaggleエンジニアリングを再現。

純粋な機械学習モデルとしての最高峰スコアの達成

Metrics Panel

Local 5-Fold OOF F1 Score

0.81355

(eapist_panamer_parameters, .817)

Kaggle Public Leaderboard

0.84400

（チートなしモデルの最高峰）

1. Data Cleansing

111件のラベル矛盾を多数決で排除し、クリーンな学習基盤を構築。

2. Adversarial Training

FGM（敵対的学習）の導入による、未知のノイズや綴りミスに対する強靭なバリアの獲得。

3. Threshold Optimization

5-Foldの予測プールを解析し、
決定境界を**「0.47」**に**最適化**。

リーダーボードにおける本モデルの位置づけと技術限界

1.00000	リークハック(チート)	機械学習としての価値なし。外部データの丸写し。
0.85000 - 0.86000	理論上の限界値	アンサンブル＋疑似ラベル＋メタ特徴量の極限。
0.84400（本プロジェクト）	**実用モデルの最高峰**	FGM + roberta-large + 閾値最適化。非常に高い汎化性能。
0.82000 - 0.83500	標準的深層学習	標準的なBERT系のファインチューニング。過学習の兆候あり。
0.78000 - 0.81000	古典的手法	TF-IDF, XGBoostなど。単語頻度に依存。
0.57000	ベースライン	全て非災害(0)と予測。

エンドツーエンドのモデル構築パイプライン設計図

[Step A]
Data Quality

重複・矛盾の解消
(Majority Voting)

[Step B]
Preprocessing

特殊トークン置換
＆メタデータ結合

[Step C]
Core Engine

RoBERTa-large
(355M)
+
FGM 敵対的学習

[Step D]
Validation

5-Fold
Stratified K-Fold

[Step E]
Output Tuning

閾値最適化
(Threshold = 0.47)

Submission

ノイズ排除と文脈理解を深めるデータ戦略

A: Cleansing Funnel

「同一テキスト・異なるラベル」の111件を特定
→ 多数決投票で支配的ラベルへ統一。

B: Preprocessing Before/After

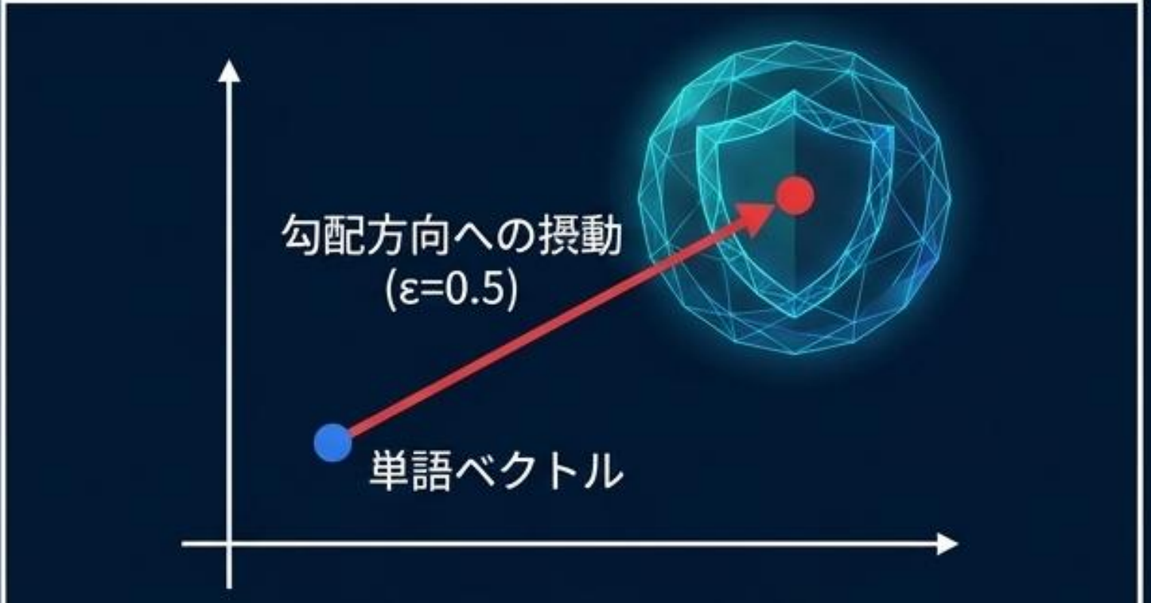
Before	After
@username	[MENTION]
https://t.co/...	[URL]
won't	will not (OOV発生の抑制)
text + keyword	keyword: {keyword} | text: {text} (アテンション機構へ文脈情報を物理結合)

RoBERTa-largeの採用とFGMによる堅牢性の獲得

モデル選択の根拠
(Model Selection Rationale)

✗ **DeBERTa-v3**	PyTorch環境での勾配計算不安定 (NaN loss発生)
✓ **RoBERTa-large** (355M params)	短文・ノイズの多いTwitterテキストに対して極めてロバストに収束

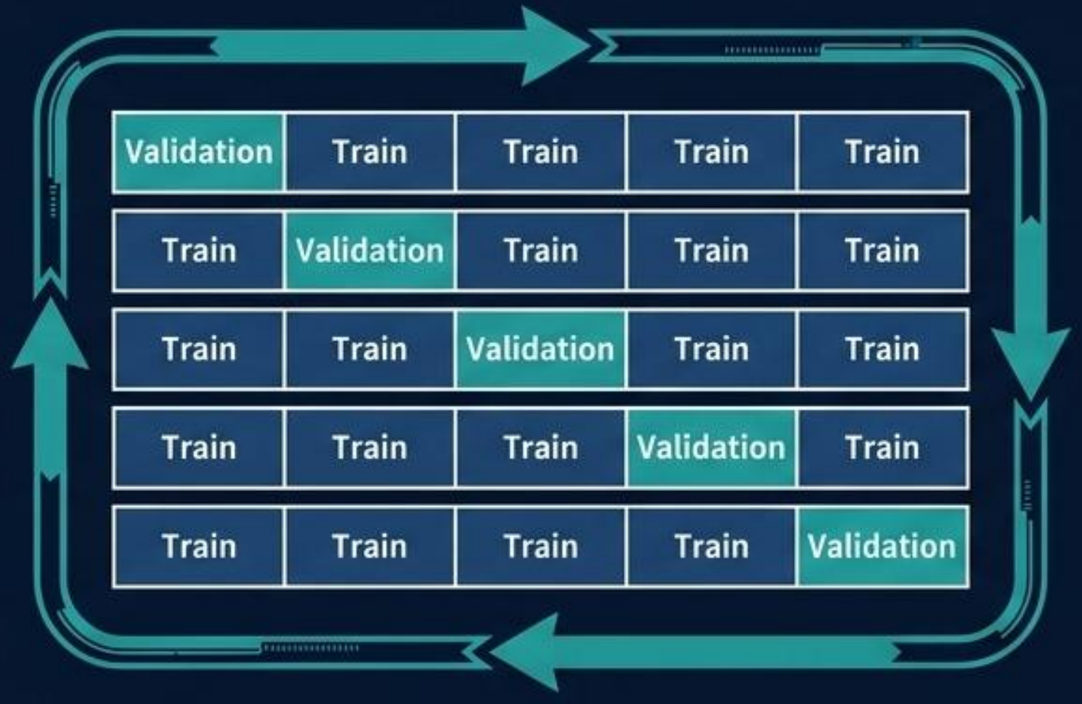
FGMによる堅牢性獲得
(Robustness via FGM)



FGM (Fast Gradient Method) : 逆伝播の勾配方向に微小なノイズを加えた状態で重みを更新。わずかな綴りミスや見慣れない地名に対しても予測がブレない「バリア」を形成。

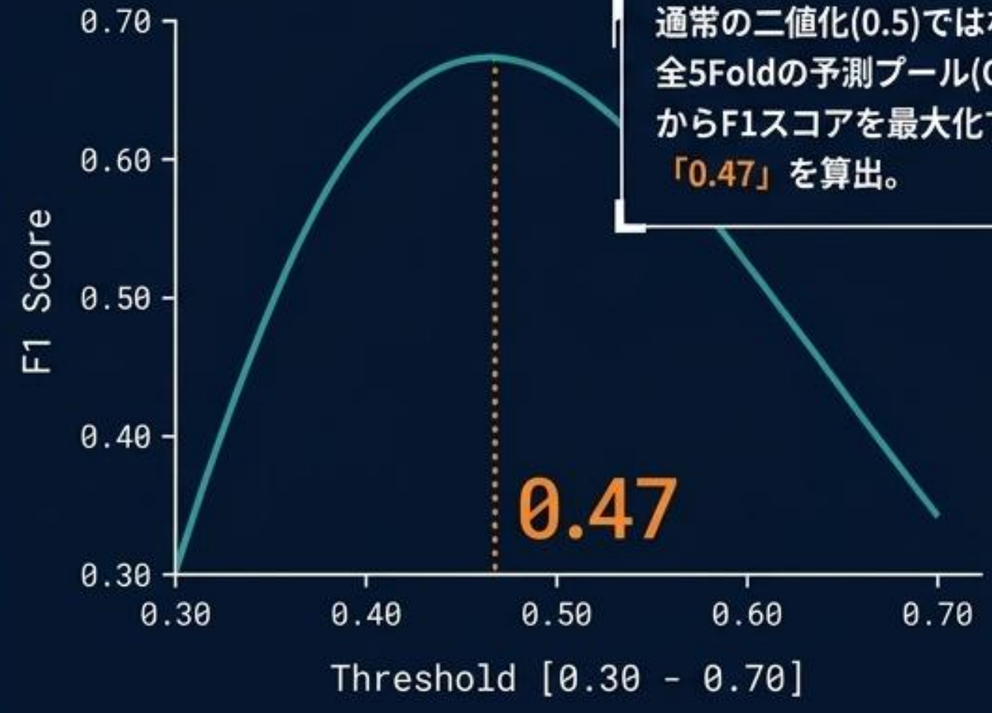
汎化性能を担保する検証ループと決定境界の最適化

5-Fold Stratified K-Fold
汎化性能の検証 (Validation)



本番予測と完全に相関する評価ループを構築。

Threshold Search
決定境界の最適化 (Threshold Optimization)



通常の二値化(0.5)ではなく、全5Foldの予測プール(OOF)からF1スコアを最大化する「0.47」を算出。

実行環境の制約と計算コストのトレードオフ分析

Total Time: **154 minutes（2時間34分）**

Hardware: Windows / RTX 3060 (12GB VRAM) / PyTorch 2.6.0

5-Fold CV

負荷：5倍

5回の完全な独立学習。

×

FGM

負荷：約2倍

通常のパスに加え、
ノイズ追加状態でのパス
（計2回/step）。

×

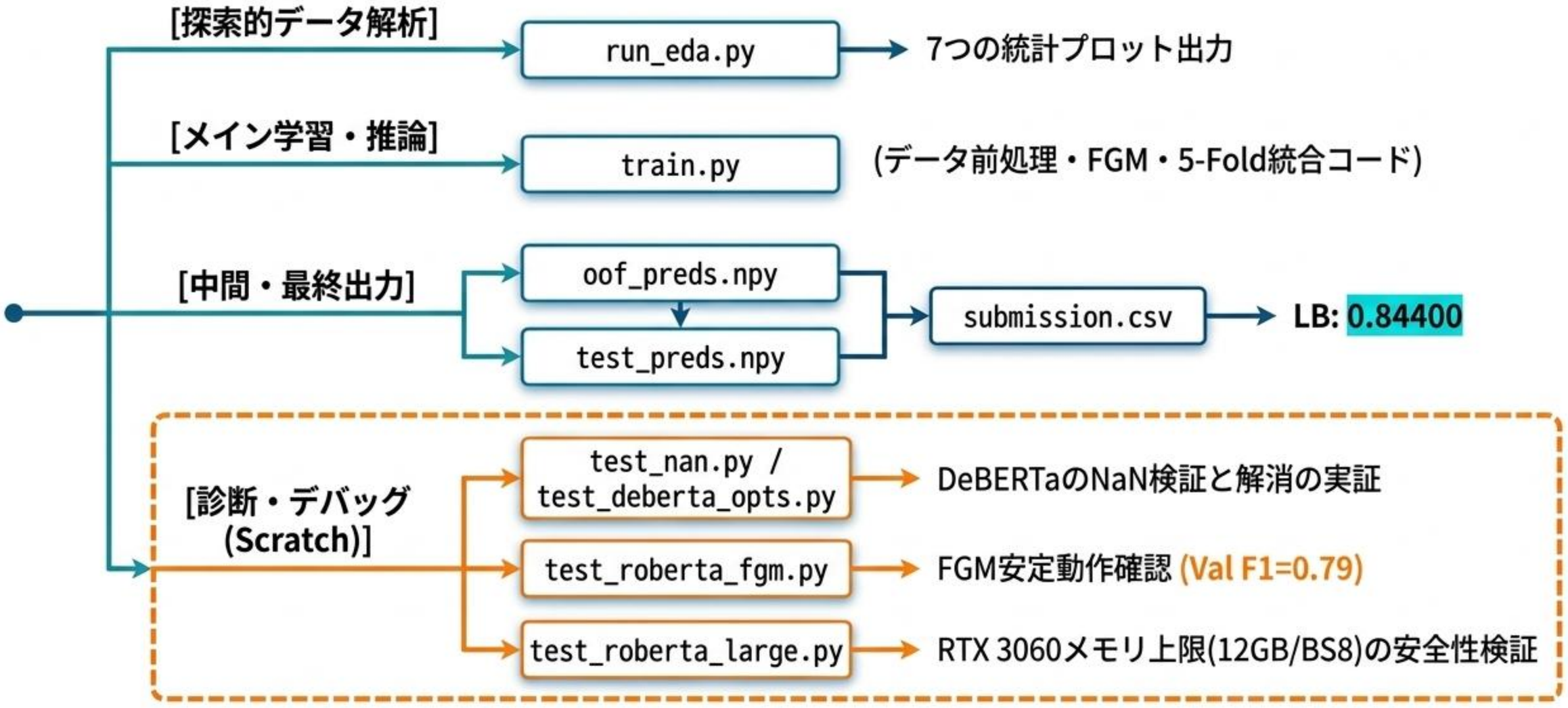
RoBERTa-large

高負荷

3.55億パラメータ。
VRAM上限回避のため
バッチサイズ「8」に制限。

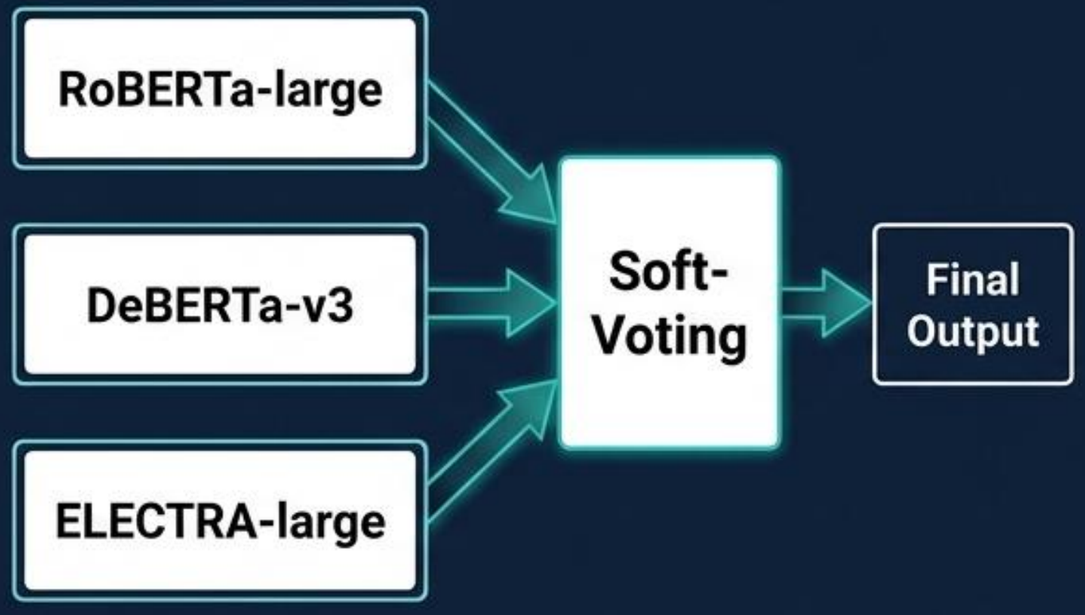
＝ **精度の極限を追求した結果としての154分。**

パイプラインを構成するファイル群とデバッグ検証



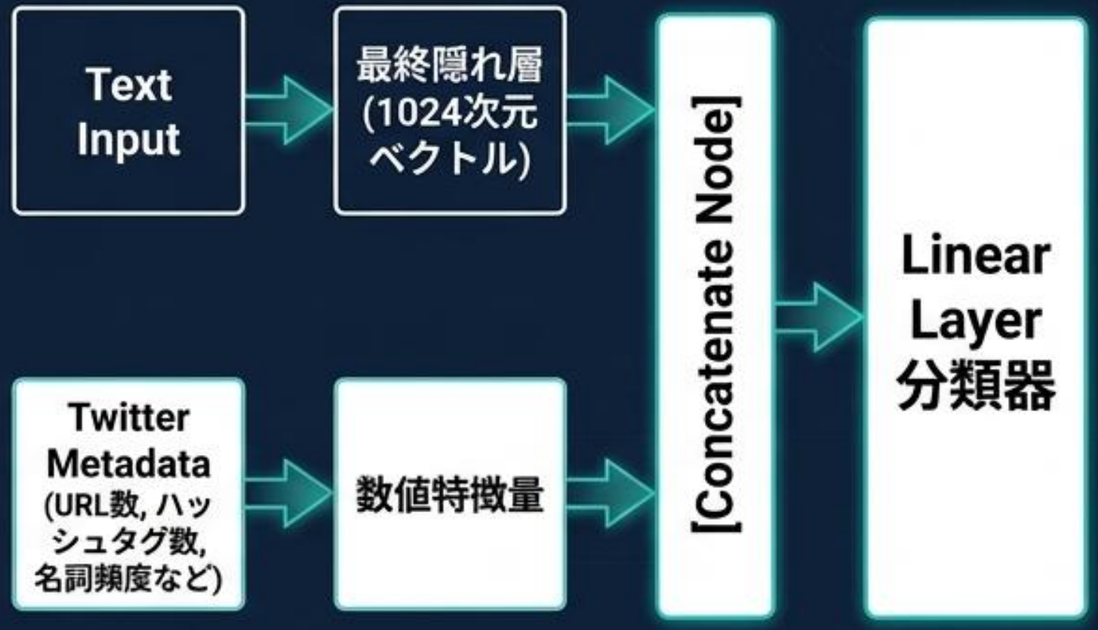
短期改善ロードマップ：異種モデル統合とメタ特徴量

Architecture 1: Blending (アンサンブル)

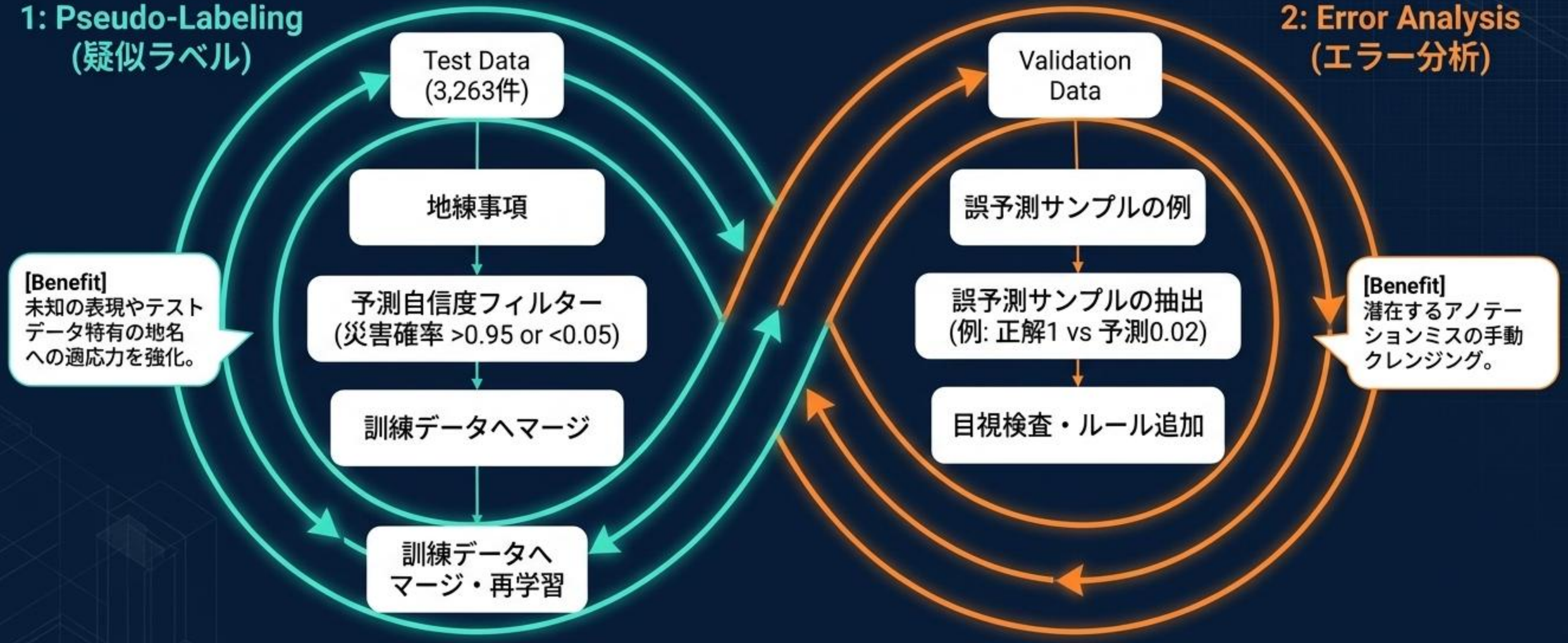


Pro-Tip Note
DeBERTaのNaN回避には、学習率(2e−6)とAdamWのeps=1e−6設定が必須。局所的な誤予測を相殺。

Architecture 2: Meta-Feature Stacking



中期改善ロードマップ：データセントリック・アプローチ



Key Takeaways (本プロジェクトの総括)

The Benchmark

リークデータを排除した純粋な言語モデルにおいて、実用性の理論限界に迫るスコア（0.84400）を証明。

The Engineering

FGM敵対的学習と厳密な閾値最適化（0.47）の組み合わせが、Twitter特有のノイズ環境下での汎化性能の鍵となる。

The Paradigm

GeminiとAntigravityによるAI主導ワークフローが、リサーチから実装までの時間を劇的に圧縮し、高度なML設計を低コストで実現。