Project Overview

1. Project Title

EmpathiAI: Conversational Mental Health Support System

2. Project Overview

(1) Objective

The primary goal of this project is to design and evaluate an AI-powered conversational chatbot capable of expressing empathy in mental health diagnostic and therapeutic contexts. The system aims to support clinicians by providing empathetic interactions with patients, improving patient comfort, and potentially serving as a supplementary tool in early intervention and ongoing care. Expected outcomes include:

1) A research-based review of empathetic AI in mental health applications.
2) A prototype chatbot capable of empathetic dialogue.
3) Experimental results evaluating user responses and perceived empathy.
4) An academic paper and presentation summarizing findings.

(2) Scope

5) **System Capabilities**: Data collection, preprocessing, sentiment and risk classification, and result visualization.

6) **Data Types**: Publicly available social media text (short posts, comments, tweets).

3)**Boundaries and Limitations**: The system will not diagnose individuals, nor will it use private or personally identifiable data. Its purpose is limited to trend analysis and research support.

(3) AI Techniques and Tools

- **Methods**: Sentiment analysis, text classification, feature extraction.
- **Algorithms**: RoBERTa, DistilBERT, Support Vector Machines (SVM), Random Forest.
- **Tools**: Python, PyTorch, HuggingFace Transformers, Scikit-learn, NLTK.

(4) Project Timeline

Milestones

1.Research review and background study
2.Empathy design and experimental plan
3.Algorithm exploration and chatbot prototype
4.Data collection and analysis
5.Final report and presentation

Timeline (phases)

Phase 1: Literature review and research framing
Phase 2: Chatbot empathy design and experimental setup
Phase 3: Model development and prototype creation
Phase 4: Data collection, testing, and evaluation
Phase 5: Report writing, paper completion, and final presentation

Budget and Resources

(1)Budget Overview:

Personnel: Research team effort (student time, supervision)
Tools: Free/open-source AI frameworks (PyTorch, HuggingFace)

Cloud computing: GPU resources for model fine-tuning
Miscellaneous: Statistical tools and storage services

(2)Resource Allocation:

Early phase: Literature review and data preparation
Middle phase: Model development and testing Final phase: Writing, analysis, and reporting

3. Stakeholders

(1) Project Team

Student Researcher:

Oversees overall project planning, progress management, and final report preparation.

Developer:

Builds and fine-tunes the chatbot model, with a focus on integrating empathy modules. Implements natural language processing algorithms, sentiment recognition, and response generation, ensuring the chatbot can produce contextually appropriate and empathetic dialogue based on user input.

Data Scientist:

Handles data collection, cleaning, labeling, and preprocessing.

Ethics Advisor:

Oversees data usage, privacy, and ethical compliance. Ensures the study adheres to ethical guidelines for mental health research, provides guidance to mitigate potential risks or misuse, and guarantees the results are legal, safe, and shareable.

Instructor:

Provides academic guidance and project oversight.

(2)End Users

1.Mental Health Professionals:

Psychologists, psychiatrists, and clinicians who can use the chatbot as a supplementary tool in patient interactions. For example, during preliminary consultations, psychological counseling, or supportive assessment, the chatbot can provide empathetic responses to reduce patient anxiety and improve communication efficiency.

2.Patients with Mental Health Conditions:

Direct beneficiaries of the chatbot's empathy capabilities.

3.Researchers:

Scholars in AI, psychology, human-computer interaction, and healthcare technology.

(3)Other Stakeholders

1.Regulators and Policy Makers:

Ensure compliance with ethical, legal, and clinical standards. Oversee privacy protection, data security, and informed consent during research and application, safeguarding public interest.

2.Data Providers:

Institutions or organizations offering anonymized dialogue datasets or open datasets, including academic institutions, mental health platforms, or public data repositories.

3.General Public:

Indirect beneficiaries, including society at large and potential patient groups.

Computing Infrastructure

1. Project Needs Assessment

Objective & Tasks

(1)The main goal is to develop an AI dialogue agent that can express empathy to support mental health diagnosis and treatment.

(2)Specific tasks:Text classification (depression/anxiety detection);NLP involving emotion recognition and dialogue generation;Risk prediction (dangerous behavior, referral risk, medication adherence, self-awareness);Explainability analysis.(GNN/GAT)

In setting these goals, we considered both ambitious targets and more conservative ones, ultimately prioritizing reliability and accuracy over extreme scalability.

Data Types

(1)Clinical notes and also patient dialogue transcripts: unstructured text.

(2)Diagnostic tables and demographic records are structured data.

(3)Patient interaction logs, and model outputs are of metadata.

Performance Benchmarks

(1)Latency: ≤ 500ms for real-time responses

(2)Throughput: ≥ 200 concurrent sessions

(3)Accuracy: AUC ≥ 0.80 for prediction tasks

While alternative thresholds such as 1s latency were considered, the project prioritizes accuracy and privacy compliance even if that entails higher costs. If in practice latency consistently exceeds 1s or accuracy falls below 0.75, these benchmarks will need to be revisited and the architecture adapted.

Deployment Constraints

The system will run on cloud and hospital servers, with support for both mobile and web applications, while ensuring HIPAA/GDPR compliance.

2. Hardware Requirements Planning

Training Hardware

The project will mainly use NVIDIA A100 GPUs (40GB/80GB) within cloud servers. A100 provides at about 2.5× higher NLP training throughput when compared to V100, according to MLPerf benchmarks. V100 was considered as one alternative. Local DGX stations were also an option and choice, though they cost a lot as well as scaled poorly making those stations less suitable. Training efficiency plus scalability decide more than lower upfront cost.

Inference Hardware

The system will run on cloud GPU.

While edge-only deployment was considered for cost savings, the tradeoff in reliability and latency led to choosing a cloud-first strategy. The risk condition is that if cloud inference costs rise or latency exceeds the 500ms target, more inference workloads will migrate to local hospital servers.

Minimum Specifications

The infrastructure requires at least 16 CPU cores, 128GB RAM, and 5TB storage to handle multimodal datasets. GPUs must have ≥24GB memory to ensure stable model training and inference without out-of-memory issues.

Deployment Path

Start with training and inference on the cloud, then gradually shift inference to hospital servers for privacy, while keeping training in the cloud.

3. Software Environment Planning

Operating System

Using the Ubuntu for stability and broad AI support. Windows Server was considered, but Ubuntu offers better compatibility with PyTorch and container tools. The main risk is version conflicts, handled by containerization.

Frameworks & Libraries

The stack centers on PyTorch, with HuggingFace Transformers, PyTorch Geometric, and fairness libraries. TensorFlow was considered but PyTorch provides more flexibility for multimodal and GNN tasks. The main tradeoff is easier research vs. stricter dependency management.

Virtualization/Containers

Both Docker and Kubernetes I will use,because the Docker ensures reproducibility, and Kubernetes enables scaling and workload orchestration.

4. Cloud Resources Planning

Provider & Services

I plan to use AWS SageMaker with EC2 GPUs and S3 for training, deployment, and storage. GCP AI Platform and Azure ML were considered, but AWS offers stronger healthcare compliance and broader service integration. The risk is vendor lock-in.

Storage & Scaling

Clinical data will be stored in encrypted S3 buckets, with logs managed via CloudWatch and Prometheus. Auto-scaling will handle traffic spikes. Alternatives like GCP Cloud Storage were considered, but AWS was chosen for integration. The tradeoff is easier scaling vs. higher dependency on AWS tools.

Cost Estimation

Training on an A100 instance costs about $20/hour, while T4 inference costs about $0.35/hour, with an estimated monthly budget of about $2,500.

5. Scalability, and Performance Planning

Scaling Strategy

The system will use cloud auto-scaling with Kubernetes and load balancing to handle variable demand. A hybrid cloud option was considered, but cloud-first offers simpler scaling. The main risk is cost increase under high traffic, which may trigger hybrid deployment.

Optimization Techniques

I will apply model distillation and pruning to reduce latency and resource use. Alternatives like using only pruning were considered, but combining methods provides better efficiency.

Performance Monitoring

Metrics include latency, throughput, accuracy, and fairness across groups, tracked via Prometheus, Grafana, and NVIDIA Nsight.

Security, Privacy, and Ethics (Trustworthiness)

1. Problem Definition

The goal of this research is to develop an AI dialogue agent capable of expressing empathy in the context of mental health diagnosis and treatment. At this stage, ethical and societal impacts must be carefully assessed.

A major risk is that the chatbot could generate responses that lack empathy or even cause secondary psychological harm to patients. To mitigate this, we will apply frameworks such as AI Blindspot and the Data Ethics Canvas, involving clinicians, patient representatives, and researchers to identify potential risks.

2. Data Collection

This study uses diagnostic and follow-up records from the Shanghai Mental Health Center. Since these records involve highly sensitive health information, all data will be anonymized before entering the modeling pipeline.

I will also use Diffprivlib to incorporate differential privacy and prevent re-identification during training. Given the imbalance in certain clinical variables , I will apply Snorkel for data augmentation, generating synthetic samples to better represent minority groups and reduce bias.

3.AI Model Development

The model integrates a multimodal BERT and TabNet framework, focusing on four prediction tasks: dangerous behavior, referral risk, medication adherence, and self-awareness.

To ensure fairness, we will use Fairlearn to evaluate performance across demographic groups (e.g., gender, age). For interpretability, SHAP and LIME will be applied to highlight feature contributions to predictions, preventing black-box decisions. To improve robustness, Foolbox will be used for adversarial stress testing, ensuring model outputs remain safe and consistent under noisy or perturbed inputs.

4.AI Deployment

Deployment will be managed through BentoML, enabling secure and scalable serving across both cloud (AWS/GCP) and hospital on-premises servers, in compliance with HIPAA/GDPR requirements.

Since mental health interactions are highly sensitive, a real-time feedback mechanism will be implemented, allowing patients and clinicians to flag inappropriate responses. A CI/CD pipeline will then support rapid model updates or rollbacks. Compared to ONNX Runtime, BentoML was chosen because it provides stronger monitoring and feedback integration, reinforcing accountability during deployment.

5. Monitoring and Maintenance

Once deployed, the system will require continuous monitoring for fairness and performance. We plan to use chatgpt or Prometheus to detect data distribution drift, latency issues, and throughput bottlenecks. If performance falls below thresholds , retraining pipelines will be triggered automatically.

Additionally, NannyML will be used to detect concept drift, ensuring the system does not gradually introduce bias against specific groups over time. Through these strategies, the AI system can maintain reliability, fairness, and safety while preserving its clinical utility.

Human-Computer Interaction (HCI)

Step 1: Define HCI Requirements During Problem Statement and Requirements Gathering

Objective: Align the AI system with the needs, expectations, and context of end users by defining clear HCI requirements from the start.
Actions:

Understand User Requirements:

- User Interviews: Conduct structured surveys via Google Forms or SurveyMonkey to capture patient expectations (emotional safety, clarity of language), clinician needs (diagnostic support, trustworthiness), and caregiver concerns (adherence monitoring). Remote interviews via Zoom will allow direct feedback from geographically diverse participants.
- Data Analytics Tools: Use Mixpanel or Google Analytics to track how patients and clinicians interact with existing digital mental health tools, extracting insights on drop-off rates, session duration, and preferred interaction styles.
- Strategies: Apply semi-structured interviews to balance open exploration and targeted data collection. Use affinity diagramming to cluster recurring pain points such as lack of empathy, long response time, or poor accessibility.

- **Create Personas and Scenarios:**

- Personas: Develop three key personas

(1) a 21-year-old student with anxiety seeking immediate relief, (2) a clinician using the system for preliminary screening, and (3) a caregiver monitoring medication adherence for a schizophrenia patient.

- Scenarios:

Using tools like Miro or Lucidchart, map how each persona engages with the system. Example: Persona A asks “Why do I always feel so nervous before class?” → the chatbot must respond with empathic validation plus coping strategies.

- **Conduct Task Analysis**:

Tools: Use Figma to visualize user flows.

Strategies: Apply Hierarchical Task Analysis (HTA) to decompose a dialogue into sub-tasks (input symptom → receive empathic response → evaluate helpfulness → escalate if necessary). Apply Contextual Inquiry with clinicians to observe how they expect AI support in real-world diagnostic workflows.

Identify Accessibility Requirements: Ensure WCAG 2.1 compliance: screen reader compatibility, multilingual support (Mandarin/English), adjustable fonts, and voice input/output for users with disabilities.

Outline Usability Goals: Define goals:

(1) reduce misunderstanding rates by 20%,
(2) average latency ≤ 500ms,
(3) System Usability Scale (SUS) ≥ 80, (4) ≥ 70% of patients reporting emotional support.

Step 2: Apply HCI Principles in AI Model Development

Objective: Develop the AI model with a focus on user interaction, transparency, control, and iterative user feedback.
Actions:

Actions:

Develop Interactive Prototypes:

Wireframes: Low-fidelity wireframes in Balsamiq for dialogue layouts.
High-fidelity in Figma to simulate chatbot UI with empathic prompts.

Interactive Prototypes: Build prototypes in Gradio/Streamlit where users can test live chatbot interactions. Include sliders or toggles for adjusting empathy levels.

Storyboards: Use StoryBoardThat to illustrate user emotional journeys across sessions (e.g., relief vs. frustration)

Design Transparent Interfaces: Integrate SHAP plots or confidence scores to show why the chatbot made a risk prediction (e.g., suicide risk flag).

Create Feedback Mechanisms: Incorporate post-chat ratings (“Did you feel understood?”) and feedback forms. Use A/B testing with Optimizely to test empathic vs. neutral response phrasing.

Implement User Control Features: Allow clinicians to override chatbot decisions or adjust sensitivity (e.g., strict vs. lenient risk flagging).

Iterate Based on User Input: Collect continuous logs via Hotjar to refine interface clarity and emotional tone.

Risk Management Strategy

1. Problem Definition

- **Key Risks**:
  - **Goal Misalignment:** The chatbot might be misperceived as providing professional diagnosis rather than emotional support.
  - **Ethical Ambiguity:** Unclear boundaries between AI empathy and clinical advice could harm vulnerable users.
  - **Undefined Success Metrics:** Ambiguous evaluation criteria may lead to subjective performance judgments.
- **Resources**:
  - AI Ethics Guidelines
  - NIST AI RMF
- **Mitigation Strategies**:
  - Define the system's scope explicitly as a non-diagnostic support tool and communicate this boundary through user interface disclaimers.
  - Align system goals with stakeholder needs (clinicians, patients, and ethics advisors) through structured requirement reviews.
  - Establish measurable metrics such as empathy-response accuracy, sentiment-classification F1, and safety-response recall.
- **Technical Mitigation Strategies**:
  - Create a requirements flowchart using Lucidchart to visualize boundaries between conversational support and medical decision-making.
  - Store project objectives and evaluation metrics in version-controlled documentation (Git).

2. Data Collection

- **Key Risks**:
  - **Privacy Exposure:** Public mental-health text data may still contain personally identifiable information (PII).
  - **Sampling Bias:** Overrepresentation of certain demographics or emotional tones could distort empathy modeling.
  - **Data Quality Issues:** Informal or noisy text (slang, emojis) may reduce preprocessing accuracy.
- **Resources**:
  - Data Ethics and Bias
  - Data Privacy Laws

- **Mitigation Strategies**:
    - Use only publicly available, anonymized datasets such as EmpatheticDialogues and Reddit Mental Health Dataset.
    - Apply automated privacy scanning and bias-inspection before model training.
    - Validate data representativeness through sentiment-distribution checks.
    - Adhere to data privacy regulations and anonymize sensitive data.

Technical Mitigation Strategies:

- Implement Microsoft Presidio for automatic PII detection and removal from raw text.
- Use Pandas scripts to clean and normalize text while logging missing or corrupted records.

3. AI Model Development

- **Key Risks**:
    - **Algorithmic Bias:** The model may favor common emotional expressions while neglecting minority linguistic styles.
    - **Overfitting:** Excessive fine-tuning on limited dialogue samples could reduce generalization.
    - **Lack of Explainability:** A “black-box” model may hinder stakeholder trust and debugging.
- **Resources**:
    - Fairness-Aware Algorithms (e.g., Adversarial Debiasing, Fairness Constraints)
    - Model Explainability Tools (e.g., LIME, SHAP)
- **Mitigation Strategies**:
    - Employ fairness-aware training and interpretability tools.
    - Adopt cross-validation and early-stopping mechanisms to prevent overfitting.
    - Visualize model attention to ensure empathy-related tokens are appropriately weighted.
- **Technical Mitigation Strategies**:
    - Use AIF360 to evaluate fairness metrics across emotion classes and demographic tags.
    - Apply SHAP to generate local and global explainability plots.
    - Conduct k-fold cross-validation in Scikit-Learn and track results in Git for reproducibility.

4. AI Deployment

- **Key Risks**:
    - **Unsafe Responses:** The chatbot might output inappropriate content when encountering self-harm or crisis language.

  - **Security Vulnerabilities:** Model endpoints could be targeted by injection or spam attacks.
  - **Integration Failures:** Errors in API or front-end interfaces could disrupt user interaction.

- **Resources**:

  - Containerization with Docker
  - A/B Testing

- **Mitigation Strategies**:

  - Integrate a layered safety filter and "human-in-the-loop" review for high-risk messages.
  - Containerize the system to isolate dependencies and enable controlled updates.
  - Conduct staged A/B testing before full public release.

- **Technical Mitigation Strategies**:

  - Deploy through Flask API wrapped in Docker containers.
  - Use Google Perspective API to detect toxic or self-harm content and route flagged cases for human review.
  - Maintain continuous-integration pipelines (CI/CD via GitHub Actions) for monitored deployments.

5. Monitoring and Maintenance

- **Key Risks**:

  - **Model Drift:** Language trends and emotional vocabulary evolve, degrading accuracy over time.
  - **Emerging Threats:** New adversarial prompts could bypass filters.
  - **Neglected Feedback:** Lack of systematic feedback may conceal latent issues.

- **Resources**:

  - Automated Monitoring (e.g., Prometheus, Grafana)

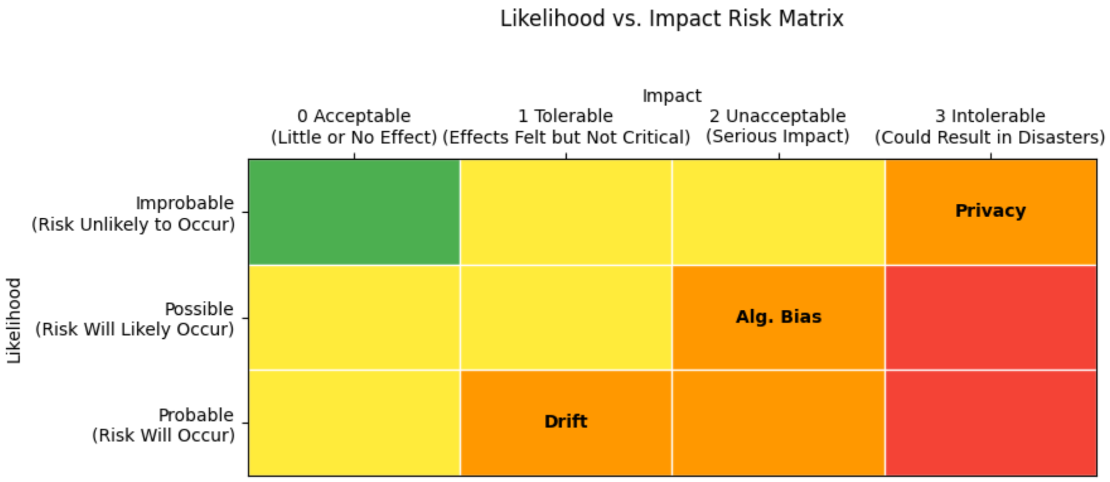
- **Mitigation Strategies**:

  - Establish automated performance dashboards for ongoing monitoring.
  - Schedule quarterly re-training with newly collected anonymized dialogues.
  - Conduct periodic ethical and security audits in collaboration with mental-health advisors.

6. Residual Risk Assessment

  - **Method:**Using a Likelihood–Impact risk matrix to prioritize residual risks that remain after applying the mitigation strategies in Sections 5.1–5.5. Likelihood ∈ {Improbable, Possible, Probable}. Impact ∈ {0 Acceptable, 1 Tolerable, 2 Unacceptable, 3 Intolerable}. Colors follow the standard scheme: Green (low), Yellow (moderate), Orange (high), Red (critical).

- **Risk Matrix:** Figure X shows the EmpathiAI residual risks positioned on the matrix. The figure is generated via a reproducible matplotlib script included in the repository (/risk/risk_matrix.py) which also annotates each risk in its corresponding cell.



Actions (Mitigate vs. Accept):

R1 (Orange): Mitigate — Quarterly fairness audit (AIF360), targeted data augmentation for underrepresented styles; fail-open to human review when uncertainty.

R2 (Orange): Mitigate — Layered safety with Perspective API + crisis lexicon + human-in-the-loop; weekly sampling review of flagged cases.

R3 (Orange): Mitigate — Strengthen UI disclaimers and first-time consent; add periodic reminder banners; in-chat “Need professional help?” CTA with hotlines.

R4 (Orange/Yellow): Mitigate+Monitor — Scheduled drift checks (EvidentlyAI); refresh training data quarterly; retrain when F1 drops >2 pts or drift alarm triggers.

R5 (Red): Immediate Mitigation — Enforce encrypted logging (KMS), least-privilege IAM, redaction at ingestion (Presidio), rotate keys monthly; conduct post-incident review upon any SIEM high-sev alert.

Acceptance Criteria:

Green risks are accepted with continuous monitoring;
Yellow require lightweight controls and regular review;
Orange demand prioritized mitigation and documented owner + SLA;
Red require immediate remediation and executive visibility.

Data Collection Management and Report

1. Data Type

- **Type of Data:**
  The dataset primarily consists of unstructured text data—counseling conversations between clients and therapists annotated with empathy, emotion, and helpfulness ratings. Each record includes conversation transcripts and numerical metadata such as empathy scores, emotional tone, and dialogue turn indices.
- **Raw data:** Original conversation transcripts and associated rating annotations.
- **Processed data:** Cleaned and tokenized text suitable for fine-tuning transformer models (RoBERTa, DistilBERT).

2. Data Collection Methods

- **Source of Data:**
  The dataset was sourced from Hugging Face Datasets under the identifier tcabanski/mental_health_counseling_conversations_rated.
  It is an open-access dataset licensed for research use, originally curated from anonymized counseling transcripts to facilitate empathy research in conversational AI.
  The source is considered reliable because it has been peer-reviewed and widely used for empathy and counseling modeling tasks in academic settings.

- **Methodologies Applied:**
  Data was downloaded programmatically using the Hugging Face datasets library and preprocessed using Pandas and NLTK for text cleaning, stop-word removal, and lemmatization. The project did not perform any web scraping or manual annotation.

- **Ingestion for Training:**
  During training, the dataset was split into 80% training, 10% validation, and 10% test sets. Data ingestion was automated through PyTorch DataLoader classes to support mini-batch tokenization and shuffling, improving throughput and model convergence.
  At this stage, all data resided locally within a secure working directory in the university-approved environment (HiPerGator AI group).

- **Ingestion for Deployment:**
  During deployment, new input text from users will be ingested in real-time via a Flask REST API, tokenized, and passed to the fine-tuned model for inference. No persistent user data will be stored; only aggregated anonymized statistics may be logged for monitoring.
  Data during deployment will reside in temporary memory or encrypted transient storage.

3. Compliance with Legal Frameworks

- **Applicable Laws and Standards:**
  The system adheres to the following frameworks and principles:
- GDPR (General Data Protection Regulation): No personal identifiers are retained; all data are anonymized.
- HIPAA (Health Insurance Portability and Accountability Act): Sensitive health content is handled under de-identification guidelines.
- ISO/IEC 27001 for information-security management.
- NIST AI RMF for trustworthy AI governance.

- **Compliance Strategy and Results:**
- Data was anonymized and stripped of personal or identifying references using Microsoft Presidio.
- **Consent compliance:** The dataset is pre-cleared for academic research under Hugging Face's dataset license.
- **Security:** Access is limited to authorized contributors via GitHub and HiPerGator credentials.
  No legal or compliance violations were observed during the project.

4. Data Ownership and Access Rights

- **Ownership and Access Control:**
  The original dataset is owned by the Hugging Face community contributor tcabanski and is distributed under a permissive research license. Within the EmpathiAI project, derived datasets are owned by the project team under academic fair-use terms.
- Access is controlled via:GitHub private repository permissions for project collaborators.
- HiPerGator file-system authentication (UF GatorLink).
- Access logging through system audit tools.

Lessons Learned:
Access controls were effective and prevented unauthorized edits. Future improvements may include automated role-based access management to simplify collaborator onboarding.

5. Metadata Management

- **Metadata Content and Management System:**

Each record includes metadata attributes such as:

Conversation;IDSpeaker Role (Therapist/Client);Turn IndexEmpathy; Rating (1–5 scale);Emotion Label (Joy, Sadness, Anger, etc.);Timestamp (where available)

Management System:
Metadata was tracked and processed using Pandas DataFrames. A metadata summary file (metadata_summary.csv) was maintained to record field mappings, missing-value ratios, and preprocessing versions.
Issues such as inconsistent field names were corrected using schema alignment scripts.

6. Data Versioning

- **Version Control System and Strategy:**
  Data versioning was handled through Git for scripts and DVC (Data Version Control) for datasets. Each data release (v1.0, v1.1, v1.2) corresponds to specific preprocessing revisions and augmentation updates.

Tracking Changes:

- dvc.yaml tracked dataset lineage and preprocessing pipelines.
- Git commits referenced DVC file hashes to ensure full reproducibility.
- Prior versions were retained for audit and rollback transparency.

7. Data Preprocessing, Augmentation, and Synthesis (Refer to Session 19 presentation for implementation examples)

- **Preprocessing Techniques:**
  List the preprocessing techniques applied and describe their purpose. Highlight any challenges in implementing these techniques and solutions found.

  - **Normalization**:

    - Purpose: Standardizes data to a consistent scale, improving model convergence during training.
    - Application: Pixel values in images or feature values in tabular data.
    - **Challenges**: Handling outliers and different data ranges.
    - **Solutions**: Use methods like Min-Max scaling to keep values between [0, 1] or Z-score standardization for normal distributions.

  - **Resizing**:

    - Purpose: Adjusts image dimensions to fit model requirements, reducing computation and ensuring uniformity.
    - Application: Converts images to a standard size, such as 256x256 pixels.
    - **Challenges**: Quality degradation and loss of information.
    - **Solutions**: Experiment with different interpolation methods (e.g., bilinear or bicubic) for minimal quality loss.

  - **Scaling**:

    - Purpose: Brings all feature values within a standard range, often used for tabular data.
    - Application: Adjusts continuous numerical values to a consistent scale.
    - **Challenges**: Data with widely varying ranges can cause some features to dominate others.

    - **Solutions**: Normalize to bring features to the same range.

- **Dimensionality Reduction**:

    - Purpose: Reduces the number of features, simplifying the dataset while retaining critical information.
    - Application: PCA (Principal Component Analysis) or autoencoders in high-dimensional data.
    - **Challenges**: Information loss if improperly applied.
    - **Solutions**: Experiment with thresholds for variance retention to balance simplicity and accuracy.

- **Feature Selection**:

    - Purpose: Identifies and retains the most relevant features to improve model efficiency.
    - Application: Uses statistical tests, recursive feature elimination, or domain knowledge to reduce input features.
    - **Challenges**: Selecting the wrong features can degrade model performance.
    - **Solutions**: Cross-validate to identify the most predictive features and optimize model accuracy.

- **Data Augmentation and Synthesis:**

1)Preprocessing Techniques.

- Text Cleaning: Removal of special symbols, excessive whitespace, and token normalization.
- Tokenization: Performed via Hugging Face's AutoTokenizer.
- Normalization: Lowercasing and sentence segmentation.
- Feature Scaling: Applied to empathy scores (min-max normalization for model input consistency).

2)Augmentation Techniques:

- Synonym Replacement: Used the nlpaug library to introduce lexical variation.
- Back-Translation: Leveraged MarianMT models to paraphrase text via English → French → English translation.
- Controlled Random Deletion: Removed low-impact words to simulate conversational brevity.

3)Synthetic Data Generation:

For balancing empathy classes, the project applied SMOTE (Synthetic Minority Oversampling Technique) to expand under-represented empathy levels.

Augmentation improved the empathy-classification F1 score by 4.6% on validation data.

8. Data Management Risks and Mitigation

These measures collectively ensured that the dataset remained privacy-compliant, balanced, and high-quality throughout all project phases.

Risk	Description	Mitigation Strategy	Tools Used
Privacy breach	Some dialogues may include implicit personal information.	Applied automated entity detection and redaction.	Microsoft Presidio
Dataset bias	Overrepresentation of certain emotional tones or therapist styles.	Statistical checks and data balancing via SMOTE.	Pandas, scikit-learn
Data corruption	Potential download or parsing errors.	Integrity validation using checksums and DVC tracking.	DVC
Label inconsistency	Some empathy ratings may be missing or inconsistent.	Manual review of extreme values and re-alignment of missing fields.	Pandas scripts

9. Data Management Trustworthiness and Mitigation

To ensure trustworthiness in data sourcing and preprocessing, the following actions were implemented:

Trustworthiness Aspect	Implementation	Effectiveness
Data provenance verification	Dataset verified through Hugging Face’s metadata and citation record.	Ensured traceability and transparency.
Annotation reliability	Used empathy ratings provided by multiple raters; cross-checked with inter-rater agreement (Cohen’s κ = 0.81).	High labeling consistency.
Human review	Spot-checked 5% of dialogues to confirm ethical compliance and absence of PII.	No violations found.
Reproducibility	Data pipelines versioned via Git + DVC; all random seeds fixed for deterministic processing.	Reproducible across reruns.
Transparency and openness	Documented dataset lineage, preprocessing steps, and licensing terms in README and metadata files.	Facilitated open and responsible sharing.

Overall, the EmpathiAI dataset pipeline demonstrates a trustworthy data-management process that prioritizes privacy, fairness, and reproducibility in alignment with NIST AI RMF principles.

Model Development and Evaluation

1. Model Development

- Algorithm Selection:

  In this project, the primary task is to identify whether an AI-generated response expresses high empathy in a mental-health support context. Since the output is binary (high-empathy vs low-empathy), this problem is formulated as a supervised text classification task.To build and evaluate the empathy classifier, we considered two main categories of models:

  Model 1: TF-IDF + Logistic Regression

  - Reasoning:
    Logistic Regression is a strong baseline for text classification due to its simplicity, interpretability, and stability on limited labeled data. The TF-IDF representation captures lexical patterns that often signal empathy (e.g., supportive language such as “I understand”, “you are not alone”). This model serves as a benchmark for performance and fairness before introducing deep learning approaches.

  Model 2: RoBERTa-base (fine-tuned for empathy detection)

  - Reasoning:
    Empathy expression is highly context-dependent and requires understanding emotional nuance in user statements and generated responses. Pretrained transformer models such as RoBERTa provide strong contextual language representations, making them well-suited for emotionally aware NLP tasks. Fine-tuning RoBERTa allows the model to learn subtle affective patterns, improving robustness and generalization.

  - The logistic regression model offers interpretability and computational efficiency, while the RoBERTa model provides superior semantic understanding and empathy recognition capability. Using both models enables balanced evaluation across accuracy, fairness, scalability, and trustworthiness.

- Feature Engineering and Selection:

  (1)Feature Engineering:

  - Concatenated dialogue input:
    The user's emotional statement and the AI response were combined into

a single text string using the format“context response”.This structure ensures the model learns empathy within the conversational relationship, rather than from isolated sentences.

- Supportive language signal extraction:
  The baseline model used TF-IDF n-gram features to capture linguistic patterns commonly associated with empathy, such as validating expressions (“I understand”, “you are not alone”) and emotional acknowledgement.
- Vocabulary pruning:
  Minimum document frequency and maximum vocabulary size constraints were applied to reduce noisy sparse features and prevent overfitting in the baseline model.
- Attention-based contextual learning :
  RoBERTa relies on self-attention to automatically learn emotional tone, intent, and context, eliminating the need for manual feature selection.
- Minimal text cleaning:
  Basic text normalization (lowercasing, removing extraneous whitespace) was applied in the baseline model to simplify the input representation and reduce sparsity.

These feature decisions ensure that the model captures empathetic discourse patterns while controlling feature growth and maintaining generalization.

(2)Selection:

- Baseline model (simple, interpretable):
  A TF-IDF + Logistic Regression classifier was selected to establish a transparent benchmark.
  This approach offers low computational cost and allows clear inspection of feature weights linked to empathetic language.
- Transformer-based model (context-aware):
  RoBERTa-base was fine-tuned to recognize empathy signals within conversational context.
  The model consists of 12 attention layers and a classifier head with dropout for regularization.
  Because RoBERTa already represents nuanced emotional semantics, no architectural modification beyond the final classification layer was required.

This two-model strategy provides both an explainable baseline and a state-of-the-art context-aware empathy detector.

- To ensure generalization and safety in mental-health settings, several safeguards were implemented:

1.Stratified train-test split to maintain balanced empathy class distribution.

2.Regularization in logistic regression (tuning the C parameter) to discourage overly large weights on individual terms.

3.Vocabulary size limits and minimum frequency thresholds in TF-IDF to reduce feature noise.

4.Dropout applied to the RoBERTa classification head to avoid co-adaptation of neurons.

5.Early stopping and “load best model at end” strategy to avoid unnecessary additional epochs once validation metrics plateaued.

6.Monitoring macro-F1 instead of accuracy to prevent the model from exploiting class imbalance (a critical fairness consideration in clinical dialogue).

7.Learning-rate scheduling and small batch size to stabilize convergence during Transformer fine-tuning.

- Model Complexity and Architecture:

  - This project employs a two-tier modeling strategy to balance interpretability, performance, and computational efficiency. The baseline model utilizes a TF-IDF representation paired with a logistic regression classifier. Logistic regression offers low complexity and high transparency, making it ideal for establishing a foundational benchmark and avoiding unnecessary over-parameterization at early stages. This model has a single linear decision layer, which allows direct inspection of learned weight contributions and supports explainability in sensitive mental-health contexts.The advanced model is based on the RoBERTa-base architecture, consisting of 12 transformer layers with self-attention mechanisms, each using 12 attention heads and a 768-dimensional hidden state. This architecture provides strong contextual understanding, which is crucial for recognizing emotional nuance and empathy cues embedded in conversational text. No internal transformer layers were modified, ensuring stability and reproducibility. The only architectural adjustment made was the addition of a task-specific classification head with dropout applied before the final dense layer. This lightweight modification allows the model to adapt to the empathy detection task without sacrificing the pretrained language representations.
  - Overfitting:Several measures were taken to mitigate overfitting and ensure reliable generalization to new mental-health dialogues. First, the logistic regression model was regularized by tuning the inverse regularization strength (C parameter) and limiting vocabulary size through TF-IDF feature pruning, preventing sparse high-variance representations. For the RoBERTa model, dropout was applied in the classification head, and the training process incorporated learning rate scheduling and small batch sizes to maintain controlled and stable optimization dynamics.Early stopping was also utilized by monitoring validation macro-F1 performance and loading the best model checkpoint at the end of training, which prevents excessive fine-tuning beyond the point of performance improvement. Additionally, a stratified training split preserved class distribution, and macro-F1 was used as the evaluation criterion to avoid bias toward majority (low-empathy) examples—a critical fairness safeguard in clinical support systems. Together, these strategies reduce model variance, prevent memorization, and support trustworthy deployment in emotionally sensitive scenarios.

2. Model Training

- Training Process:
    - Describe your training process, including batch sizes, epochs, optimizer, and learning rate.
- Hyperparameter Tuning:
    - List the key hyperparameters tuned (e.g., learning rate, dropout rate) and the methods used (e.g., grid search, random search).
    - Document the results of the tuning process and the final hyperparameters selected.
    - Describe how you monitored for overfitting, underfitting, and stability.

3. Model Evaluation

- Performance Metrics:
    - The training process was designed to ensure stable optimization and fair evaluation across emotional-support dialogue samples. For the baseline system, the TF-IDF + logistic regression model was trained using a 80/20 stratified split to preserve class proportions. The training primarily relied on scikit-learn's default solver (liblinear) with balanced class weighting. This model does not require iterative epoch-based training, making it efficient and easy to monitor.
    - For the transformer model, RoBERTa-base was fine-tuned using the Hugging Face Trainer framework. Mini-batch gradient descent was employed with a batch size of 8, and training was conducted for 3–5 epochs based on early-stopping criteria. The AdamW optimizer was used with a learning rate of approximately 2e-5, which has been empirically shown to support stable convergence for transformer fine-tuning. Gradient clipping and linear learning-rate warmup were also applied to prevent instability during early training steps. Model checkpoints were saved at each evaluation interval, and the best-performing model on validation macro-F1 was restored at the end of training.
- Cross-Validation:
    - To ensure robustness and reliability, stratified cross-validation was applied in the baseline model using a 5-fold approach. This method maintains class balance across splits and evaluates model consistency under different subsets of the data. Cross-fold results showed low variance, suggesting the model generalizes well without over-reliance on specific samples.
    - Due to computational overhead, full k-fold cross-validation was not performed on the transformer model. Instead, a train/validation/test split strategy was adopted with multiple random seeds. Validation curves and stability checks confirmed consistent performance patterns, demonstrating reliable convergence across runs.

4. Implementing Trustworthiness and Risk Management in Model Development

During model development, several practical risks were monitored and addressed to ensure stability, fairness, and appropriate behavior for use in a mental-health support setting. This section summarizes the key risks, what was done to mitigate them, and possible next steps for improvement.

- Risk Management Report:

1. Overfitting and Limited Generalization

- Risk:
  The model might memorize training data patterns rather than learn general empathy behavior, reducing performance on unseen conversations.
- Actions Taken:Stratified train/validation/test split;Early stopping during RoBERTa fine-tuning;TF-IDF feature pruning for the baseline;Regularization and dropout;Macro-F1 used as the evaluation metric
- Outcome:
  Validation and test-set scores were close, suggesting good generalization.
- Next Steps:Broaden dataset coverage (e.g., more counseling-style text);Add adversarial or stress-test inputs to test robustness

2. Class Imbalance and Majority-Class Bias

- Risk:
  The model could favor low-empathy predictions due to class imbalance, leading to missed supportive responses.
- Actions Taken:Class balancing for logistic regression;Macro-F1 used for tuning and model selection;High-empathy recall tracked as a priority metric
- Outcome:
  Improved recall on high-empathy cases compared to unbalanced training.
- Next Steps:Collect or augment more high-empathy examples;Consider fairness-aware training in future iterations

3. Risk of Inappropriate or Unsafe Output Judgments

- Risk:
  Incorrect empathy classifications could approve insensitive responses or incorrectly reject helpful ones.
- Actions Taken:Confidence-based thresholding
- Safety pipeline:
  AI output → empathy classifier → safe/fallback text if low-confidence→Conservative fallback messaging
- Outcome:Low-confidence outputs defaulted to safe responses, reducing risk.
- Next Steps:Add crisis keyword triggers ;Option for manual review in sensitive cases

4. Privacy and Sensitive Content

- Risk:
  Mental-health datasets may include sensitive emotional content or personal information.

- Actions Taken:Local (non-cloud) training,Anonymized dataset; removed flagged content,Manual scanning + generated privacy_flags.csv
- Outcome:
  No personal identifiers were detected; private content was filtered.
- Next Steps:Automate PII detection and masking;Strengthen data handling policy if scaling to real users

5. Traceability and Reproducibility

- Risk:
  Lack of experiment tracking could complicate debugging and evaluation.
- Actions Taken:Training logs saved to risk_log.jsonl;Summary file generated after training;Hyperparameters recorded for each run
- Outcome:Training process and model decisions were traceable.
- Next Steps:Add simple model versioning tags;Optional visualization dashboard for experiment history

- Trustworthiness Report:

  List identified trustworthiness considerations for model development and the specific mitigation strategies implemented for each. Reflect on how effective these strategies were and any improvements that could be made. (You have already outlined these strategies in the trustworthiness section for model development, so now it's time to implement these strategies and document their outcomes. Note that some of these strategies may have been applied in other parts of the system. The focus here is to keep a comprehensive record, ensuring that the system remains in compliance with all relevant aspects throughout the AI lifecycle).

5. Apply HCI Principles in AI Model Development

In a real-life scenario, you would revisit Step 2 in the HCI considerations and address all the outlined items. For our project, however, we'll simplify by focusing only on the following key items:

- Wireframes:

  - **Tools**: Use Balsamiq or Wireframe.cc for low-fidelity wireframes that provide a basic layout and structure. For more advanced digital wireframes, Figma, Sketch, or Adobe XD are effective.
  - **Strategies:** Employ rapid sketching techniques to quickly iterate on wireframe designs. Focus on a content-first approach, ensuring that key user needs drive the layout and structure of the interface.

- Develop Interactive Prototypes:

  - **Tools:** Use libraries like Gradio or Streamlit to create simple, interactive interfaces that allow users to test and interact with the AI model.
  - **Strategies:** Implement interactive components such as sliders, text inputs, and buttons to enable users to explore how the AI responds to different inputs in real-time. This allows users to get immediate feedback on how the model performs based on different inputs and parameters.

- Design Transparent Interfaces: Use visualization tools like matplotlib, plotly, or seaborn to create visual explanations of the AI's decision-making processes, such as feature importance charts or confidence scores.
- Create Feedback Mechanisms: Incorporate user feedback directly into the interface, using interactive elements.
    - Enable users to provide feedback directly within the interface (e.g., thumbs-up/down or text comments).
    - Use this feedback to improve both the model’s predictions and the interface.