AI System Project Template

Team Members	
Teja Kolla kolla.teja@ufl.edu	Gopalakrishna Reddy Manukonda manukonda.g@ufl.edu

Project Overview

Project Title:

- AI-ENHANCED DIAGNOSIS AND SEVERITY PREDICTION OF KNEE OSTEOARTHRITIS

Project Overview:

One of the most common forms of arthritis in the knees is called osteoarthritis(OA), which can cause significant impairment and seriously impair the quality of life for those who have it. As a degenerative disease of the joints, it is characterized by joint stiffness, pain, or restricted mobility, a disease that afflicts millions of people. Common techniques of diagnosis are comprehensive patients' history assessment and various screening tests of the joints including imaging with the use of radiographs, MRI, and CT.

To tackle the growing problem of early detection and severity assessment of osteoarthritis, the use of Deep Learning model has been put into consideration. This model is designed to interpret images of the knee joints captured through X ray and is expected to give feedback to patients almost instantly through a web application. The major goal in this particular project is to predict the grade of knee osteoarthritis from a patient's x-ray image using a fast and minimal deep learning model. The model aims at classifying the alignment of the knee joints and the grading of the bones position with the aim of detecting and forecasting the severity of Osteoarthritis from X-ray images.

Stakeholders:

- **Patients:** Benefit from early diagnosis and severity prediction for their pain.

- **Healthcare Providers:** This model will assist them in decision-making and treatment planning based on the prediction of severity
- **Medical Institutions and Hospitals:** Integrate the model to streamline diagnostic procedures and help reduce the manual interpretation.
- **Researchers:** Researchers, AI engineers and data scientists involved in developing and fine tuning the model.
- **Regulatory Bodies:** Responsible for the developed model meets the necessary safety guidelines
- **Health Insurance Providers:** Helps them to access the risk and cost management for the osteoarthritis treatment.
- **Medical Device Manufacturers:** Manufacturing companies Integrating into their existing system to increase the productivity
- **Caregivers:** Gives insights to patient's caregivers to help them know about the severity.

Computing Infrastructure

1. Project Needs Assessment:

Primary Objective:

- ○ To predict the grade and position of knee osteoarthritis from a patient's X-ray image.

Tasks:

- ○ Image classification (classifying knee joint alignment and bone position).
- ○ Severity prediction (predicting the severity of osteoarthritis).

Performance Benchmarks:

- **Latency:** The model should provide predictions within a reasonable timeframe for real-time applications.
- **Throughput:** The model should be able to process a sufficient number of x-rays per unit time to meet clinical demand.
- **Accuracy:** The model should achieve high accuracy in classifying knee joint alignment and predicting osteoarthritis severity.

Deployment Constraints:

- ○ **Environment:** The model should be deployable in a cloud environment for scalability and accessibility.
- ○ **Power:** The model should be energy-efficient to minimize computational costs.

- **Network Conditions:** The model should be robust to variations in network bandwidth and latency.

2. Hardware Requirements Planning

Compute Resources:

- **GPU:** As we are using MacBook M3 pro it has Apple’s 18-core GPU, it is an apple’s custom designed architecture optimized for machine learning and graphic tasks. It is currently the top model in the MacBook Pro series. A high-performance GPU is essential for training and inference of deep learning models on medical images.
- **Neural Engine:** We are utilizing not only cpu and gpu but we have added layer of hardware called neural engine. Apple's Neural Engine is a specialized processor designed to accelerate machine learning tasks. Unlike traditional CPUs or GPUs, which are more general-purpose, the Neural Engine is optimized for the specific computations involved in neural networks. This allows it to handle these tasks much more efficiently and with lower power consumption.
- **CPU:** 12-core CPU (6 performance cores, 6 efficiency cores). A powerful CPU is needed for general-purpose tasks and data preprocessing.
- **Memory:** Unified memory (shared between CPU and GPU). Ample RAM is required to store the large dataset of x-ray images and intermediate results.
- **Storage:** sufficient storage capacity is needed to store the dataset, trained models, and results. We are storing the dataset in the local machine for now as AWS is free for only using the dataset in their environment like utilizing the dataset from s3 buckets from running the model in EC2 instance. Because of this we thought of storing the model locally.

Network Infrastructure:

- **High-speed network connectivity:** A fast network connection is necessary for efficient data transfer and communication with other components of the system.

3. Software Environment Planning

Deep Learning Framework:

- **TensorFlow or PyTorch:** These popular frameworks provide the necessary tools for building, training, and deploying deep learning models.
- **Opencv or Pillow:** These libraries are used for preprocessing X-ray images. Preprocessing includes resizing, normalization, and conversion between formats.
- **SimpleITK:**These specialized libraries provide advanced image processing functions such as image registration, segmentation, and image filtering, which may be needed for certain preprocessing or augmentation tasks unique to X-ray images.
- **Grad-CAM:** Gradient-weighted Class Activation Mapping is utilized to provide positional explainability by generating heat maps over X-ray images. These heatmaps highlight the regions that contribute most to the model's prediction, helping clinicians understand which areas of the knee joint are most indicative of osteoarthritis (OA) grade.
- **LIME(Local interpretable Model-agnostic Explanations):** These libraries offer additional interpretability by explaining model predictions at the feature or pixel level. They provide global and local interpretability, helping to explain how different features of the image influence the classification decision.
- **MLflow:** MLflow will be used for tracking machine learning experiments, managing model versions, and providing a streamlined workflow for reproducibility. This tool allows for easy comparison of different model configurations and experiments, ensuring that the best-performing model is consistently identified and deployed.

Programming Language:

- **Python:** Python is a widely used language in data science and machine learning, offering a rich ecosystem of libraries and tools.

Other Libraries:

- **NumPy:** For numerical computations and scientific computing.
- **Pandas:** For data manipulation and analysis.
- **Keras:** A high-level API built on top of TensorFlow or Theano, providing a user-friendly interface for building and training deep neural networks.
- **Matplotlib:** A versatile plotting library for creating static, animated, and interactive visualizations.
- **Scikit-Learn:** A machine learning library providing a uniform interface to various algorithms for classification, regression, clustering, and more.

  - **Seaborn:** A high-level data visualization library built on top of Matplotlib, providing a more aesthetically pleasing and convenient interface.
  - **Streamlit:** A high-level python framework to deliver interactive data apps.

4. Cloud Resources Planning

Cloud Platform:

- **AWS:** AWS offer a variety of compute, storage, and networking resources that can be scaled to meet the project's needs.

Compute Instances:

- **GPU-accelerated instances:** Choose instances with powerful GPUs to accelerate model training and inference.

Storage:

- **Object storage:** Use object storage (e.g., S3 buckets) for storing large datasets of x-ray images.

Networking:

  - **Virtual private cloud (VPC):** Create a VPC to securely isolate your resources and control network access.

5. Scalability, and Performance Planning

Model Optimization:

- **Transfer learning:** Leverage pre-trained models on large datasets to improve performance and reduce training time.
- **Model quantization:** Reduce the model's size and computational requirements by quantizing weights to lower precision.

Infrastructure Scalability:

- **Autoscaling:** Configure auto scaling to automatically adjust compute resources based on demand.
- **Distributed training:** Distribute training across multiple GPUs or machines to accelerate model training.

Security Privacy and Ethics (Trustworthiness)

1. Problem Definition

- **Data Privacy:** Protecting patient data, including x-ray images, from unauthorized access and disclosure.
- **Model Fairness:** Ensuring the model is unbiased and does not perpetuate existing disparities in healthcare.
- **Ethical Implications:** Addressing the ethical concerns related to using AI in healthcare, such as accountability, transparency, and explainability.

Goal

- Develop a secure and ethical AI system that protects patient privacy, ensures model fairness, and addresses ethical concerns.

Strategies

- **Data Privacy:**
  - Implement robust data security measures, including encryption, access controls, and data anonymization techniques.
  - Comply with relevant data privacy regulations (e.g., HIPAA, GDPR).
- **Model Fairness:**
  - Collect diverse and representative datasets to reduce bias.
  - Use fairness metrics to evaluate and mitigate bias in the model.
  - Implement techniques like adversarial training or fairness constraints during model development.
- **Ethical Considerations:**
  - Conduct ethical impact assessments to identify potential risks and benefits.
  - Ensure transparency and explainability of the model's decision-making process.
  - Involve stakeholders in the development and deployment of the system.

Stakeholder Involvement

- **Patients:** Obtain informed consent for data collection and use.
- **Healthcare providers:** Collaborate with clinicians to understand their needs and concerns.
- **Regulatory bodies:** Adhere to relevant regulations and guidelines.

- **Ethical experts:** Consult with ethicists to address ethical concerns.

Ethical Impact Assessments

- **Identify potential risks:** Assess the potential harms and benefits of the AI system.
- **Evaluate ethical principles:** Consider principles like autonomy, beneficence, non-maleficence, and justice.
- **Develop mitigation strategies:** Propose measures to address identified risks.

Risk Analysis Framework

- **Identify risks:** Identify potential security, privacy, and ethical risks.
- **Assess likelihood and impact:** Evaluate the likelihood of each risk occurring and its potential impact.
- **Develop mitigation strategies:** Propose measures to reduce the likelihood and impact of risks.

2. Data Collection

Goal

- Collect a diverse and representative dataset of x-ray images for training and evaluation of the AI model.

Strategies

- **Data Augmentation:** Generate additional training data by applying transformations to existing images (e.g., rotations, flips, zooming).
- **Data Anonymization and Privacy Techniques:**
  - Remove personally identifiable information (PII) from x-ray images.
  - Consider differential privacy techniques to protect individual privacy.
- **Bias Detection and Correction:**
  - Analyze the dataset for biases and take steps to address them (e.g., oversampling underrepresented groups).

Examples of Tools and Libraries

- **TensorFlow Data Pipeline:** For building efficient data pipelines.
- **OpenCV:** For image preprocessing and augmentation.

3. AI Model Development

Goal

- Develop a deep learning model that accurately predicts the grade of knee osteoarthritis from x-ray images.

Strategies

- **Model Selection:** Choose an appropriate deep learning architecture (e.g., convolutional neural network) based on the nature of the data.
- **Hyperparameter Tuning:** Optimize model performance by tuning hyperparameters (e.g., learning rate, batch size).
- **Regularization:** Prevent overfitting by using techniques like dropout or L1/L2 regularization.

Examples of Tools and Libraries

- **TensorFlow or PyTorch:** For building and training deep learning models.
- **Keras:** A high-level API for building neural networks.
- **Scikit-learn:** For machine learning tasks like cross-validation and hyperparameter tuning.

4. AI Deployment

Goal

- Deploy the trained model in a production environment for clinical use.

Strategies

- **Containerization:** Package the model and its dependencies into a container (e.g., Docker) for portability and scalability.
- **Cloud Deployment:** Deploy the model on a cloud platform (e.g., AWS, GCP, Azure) for scalability and accessibility.
- **Integration with Healthcare Systems:** Integrate the model with existing healthcare systems for seamless clinical use.

Examples of Tools and Libraries

- **Docker:** For containerizing applications.
- **Kubernetes:** For managing containerized applications.
- **FastAPI:** For building web APIs for model deployment.

5. Monitoring and Maintenance

Goal

- Continuously monitor the model's performance and address any issues that arise.

Strategies

- **Performance Metrics:** Track metrics like accuracy, precision, recall, and F1-score to evaluate model performance.
- **Drift Detection:** Monitor for changes in the data distribution that could impact model performance.
- **Retraining:** Retrain the model periodically to adapt to changes in the data or environment.

Examples of Tools and Libraries

- **TensorBoard:** For visualizing model training metrics.
- **AWS CloudWatch:** For monitoring cloud infrastructure and applications.
- **MLflow:** For tracking and managing machine learning experiments.

Human-Computer Interaction(HCI)

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

Objective: For this project knee osteoarthritis and severity prediction with the desired model and user interaction with the UI by defining the clear HCI requirements step by step.

Actions:

- **Understand User Requirements:** It is very much important that for any project one should know about the end user and how they are going to use the system and how it will be useful for them. There are numerous methods and techniques in the market with which we can find the user requirements. If we are dealing with a medical problem that adds more weight and responsibility to do things right as someone's life is in your hands, but with the right approaches we can do it in the right way.
    - **User Interview / surveys:** Finding who are the people going to use the product and interviewing them to know about the needs and other requirements in person is one of the common and old ways of knowing

about the user and the requirements.You can interview the doctors and the experts in the field that how they are tackling this situation from the past years. Modern way of taking interviews gone anonymous and sending the links that contain some questions and users are going to answer those questions. For that we can use survey monkey and google forms and so many more. It has features like allowing people to add their answers anonymously so that their privacy would be protected. For this way many hospitals will allow their patients to participate as well to get the most out of the other side of the perspective.

- **Creating Personas and Scenarios:** personas are the heart of requirements gathering steps. Personas are fictional yet realistic profiles that represent different users. For arthritis problems let us look into 2 different personas.
    - Persona 1: Dr. Bhavani Uma (Expert Orthopedic Surgeon)
        - Background: Dr. Bhavani Uma is an expert orthopedic surgeon in a large reputed hospital. He sees patients with arthritis daily and performs at least 3 surgeries a day. His day is packed with people having knee pains and treating them to make their pain better.
        - Goals: His goal is to identify the problem in which part of the knee with the AI model and treat them with better and cost effective ways. His motto is to treat their patients completely and get rid of their pains and educate them about their problem with the AI.
        - Frustrations: Model takes a lot of time to output the result. Even if it gives the result it lacks which part of the knee to treat.
        - Scenario for Dr.Uma: He sends the data that he received from the patients to point out which part of the knee needs to be treated. The model then undergoes and gives the results with the heat map showing the areas of severity.
    - Persona 2: Glenn (patient)
        - Background: Glenn is 70 years old and suffering from knee osteoarthritis from the past 15 years. He has been under medication for the past 10 years.
        - Goals: obtain the results in which part of the knee is affected and that causes him the pain.
        - Frustrations: Interface is not user friendly and has a lot of medical terms, where normal people can not understand those terms.
        - Scenario for glenn: He needs to know about the issue he is facing. He will input the doctor's observations and the x-ray from the radiologist and the models to identify if he has arthritis or not and if he has in which part he has.

- **Conducting Task Analysis:** Breaking down the bigger problem into the smaller parts and then solving them would be the most logical option rather than stuck with the problem and overwhelmed with the large chunk of it. Some of the useful tools are draw.io and figma.
    - **Strategies: Hierarchical Task Analysis (HTA)**
        - Patient Interaction:
            - The patient visits the specialist and if necessary he suggests getting the x-ray.
            - The patient visits the radiologist to get the X-ray.
        - Data Visualization:
            - The doctors visualize the x-ray and then upload the X-ray that the patient gets from the radiologist.
            - The model already has the trained samples against which this data would be evaluated.
        - AI Model/ Algorithm:
            - The AI algorithm would run on that image and give the doctor results and with the help of the results the doctor will get to know that his deductions are the same as the AI giving and problem facing by the patient. If everything aligns he will move to further deductions.
- **Identifying Accessibility Requirements:**
    - By keeping the users first in our project we would like to add the accessibility functions as well. So that it can reach many people and be accessed by the most.
        - Cognitive accessibility: All the information in the application should be in the simple and low level language that everyone can understand.
        - Auditory Accessibility: If the model suggests any relevant video or provides a tutorial to use the application it should have the captions on.
        - Font size: Users can adjust font size any time based on their needs.
        - High Contrast model: It will help the users to navigate the screen if they have any visual impairments.

- **Outline Usability Goals:**
    - Knee osteoarthritis projects should have realistic goals and know the problems facing their users to increase the user satisfaction.
        - Goal 1: Reduce Diagnosis Time: while dealing with the images it takes a lot of time to train and to give the results as well. It is

necessary to reduce the time to less than a minute so that users can get the reliable info fast.

        - Goal 2: Improve User Satisfaction: From the close encounters and surveys getting the reviews from the users to make the rating more than 90 percent for the better usability.
        - Goal 3: Tackling the outliers: If the model has a lot of outliers it has a lot of chance that it can predict the false positives and true negatives. It is the responsibility to keep them in check.
        - Goal 4: offline Functionality: Having the offline functionality would be a great thing. If they are in a low bandwidth area and need to get the result from the model, it would be very difficult.

Risk Management Strategy

Managing risk is essential at every step of the AI lifecycle to ensure the development of robust, reliable, and trustworthy AI systems. Each stage introduces unique challenges and potential hazards that, if left unmanaged, can lead to serious technical, ethical, or societal consequences. Below is a breakdown of the risk management strategy used in each stage of the AI lifecycle for the knee osteoarthritis severity prediction.

1. Problem Definition

- **Key Risks:**
    - Misalignment with Objectives: Misalignment between project goals and clinical needs could lead to non-beneficial outcomes for patients.
    - Ethical Risks: There is potential for unintentional harm due to biased severity assessments, particularly in underrepresented demographic groups.
    - Undefined Success Metrics: Without clearly measurable outcomes, the project risks ineffective results.
- **Mitigation Strategies:**
    - Engage with clinical stakeholders and perform regular consultations to ensure alignment with clinical needs.
    - Define success metrics in terms of model accuracy, fairness, and interpretability to address critical performance aspects.
    - Conduct regulatory compliance reviews (e.g., HIPAA) to confirm legal alignment.
- **Residual Risk Assessment:**

  - **Likelihood:** Possible
  - **Impact:** Moderate
  - **Level:** Yellow (Regular monitoring and stakeholder feedback required).

2. Data Collection

- **Key Risks:**
  - Data Quality: Presence of low-resolution X-rays may degrade model accuracy.
  - Bias in Data: Demographic imbalances could bias model predictions.
  - Data Privacy: Handling sensitive patient data raises privacy concerns.
- **Mitigation Strategies:**
  - Implement automated data validation and cleaning steps using Pandas to standardize image quality and flag low-resolution images.
  - Perform targeted data augmentation and resampling to balance demographics.
  - Use data anonymization and ensure compliance with privacy regulations (e.g., HIPAA).
- **Technical Mitigation:**
  - Libraries such as imbalanced-learn for demographic resampling.
  - Data privacy via anonymization and secure storage.
- **Residual Risk Assessment:**
  - **Likelihood:** Possible
  - **Impact:** Moderate
  - **Level:** Yellow (Quarterly reassessment required).

3. AI Model Development

- **Key Risks:**
  - Overfitting/Underfitting: Limited dataset and complex model architecture may lead to poor generalization.
  - Explainability: Lack of model transparency could hinder clinical acceptance.
- **Mitigation Strategies:**
  - Apply regularization techniques (L2 regularization and dropout) and cross-validation to prevent overfitting.
  - Use model interpretability tools like SHAP to provide insights into model decisions, enhancing clinical trust.
- **Technical Mitigation:**
  - Implementation of regularization techniques via TensorFlow.
  - Model explainability with SHAP or LIME.

- **Residual Risk Assessment:**
    - **Likelihood:** Improbable
    - **Impact:** Moderate
    - **Level:** Green (Low-risk level with monitoring).

4. AI Deployment

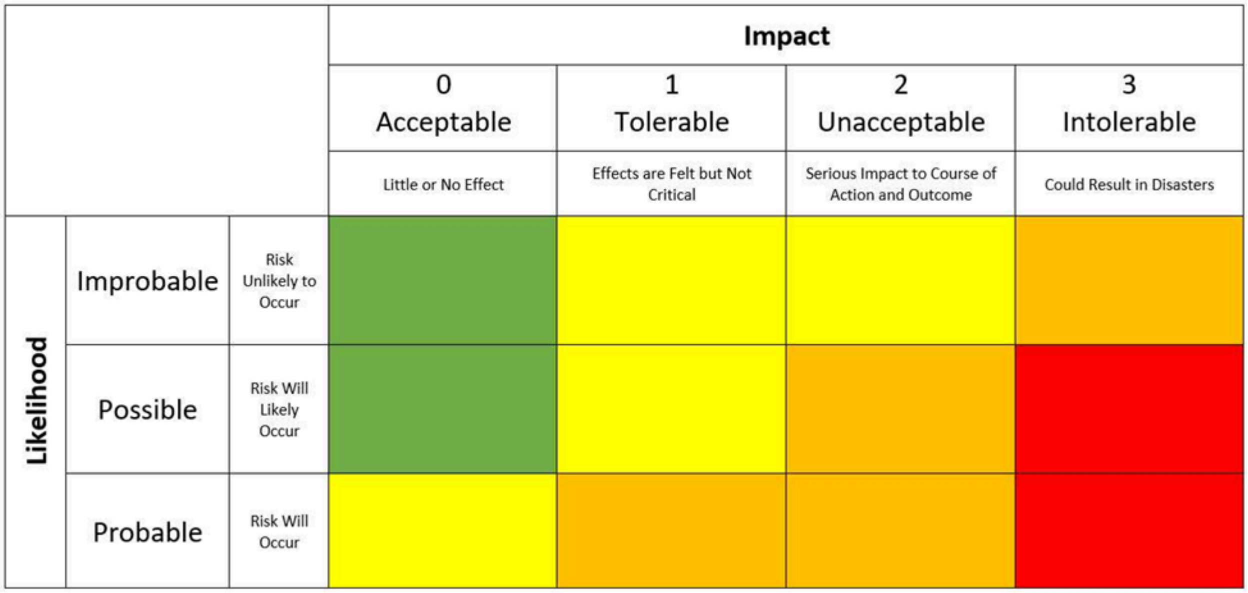
- **Key Risks:**
    - Integration Issues: Challenges in embedding the model in a clinical workflow may reduce usability.
    - Security Breaches: Exposure to potential cyber threats post-deployment.
- **Mitigation Strategies:**
    - Conduct A/B testing and gradual roll-out to minimize disruption in clinical workflows.
    - Implement security measures, including CI/CD pipeline for consistent updates and vulnerability checks.
- **Technical Mitigation:**
    - Use containerization with Docker for secure deployment.
- **Residual Risk Assessment:**
    - **Likelihood:** Possible
    - **Impact:** High
    - **Level:** Orange (Mitigation actions required).

5. Monitoring and Maintenance

- **Key Risks:**
    - Model Drift: Changes in patient demographics or clinical standards could reduce model efficacy.
    - Emerging Security Threats: Ongoing threats to data integrity and confidentiality.
- **Mitigation Strategies:**
    - Establish model drift detection and regular retraining protocols to address demographic or clinical shifts.
    - Schedule periodic security audits to proactively address new vulnerabilities.
- **Technical Mitigation:**
    - Use monitoring libraries like Evidently AI to track performance metrics.

6. Residual Risk Assessment for Knee Osteoarthritis Severity Prediction Project

To manage the remaining risks effectively, we apply the Likelihood vs. Impact Risk Matrix. This tool assesses each residual risk by considering both its likelihood (how probable the risk is to occur) and its impact (the severity of the consequences), thus guiding us in prioritizing risk mitigation actions.



			Impact			
			0 Acceptable	1 Tolerable	2 Unacceptable	3 Intolerable
			Little or No Effect	Effects are Felt but Not Critical	Serious Impact to Course of Action and Outcome	Could Result in Disasters
Likelihood	Improbable	Risk Unlikely to Occur	🟩	🟨	🟨	🟧
	Possible	Risk Will Likely Occur	🟩	🟨	🟧	🟥
	Probable	Risk Will Occur	🟨	🟧	🟧	🟥

- **Effectiveness of Our Risk Management Strategies**
  In this project, we implemented several risk management strategies to ensure data integrity, privacy, and model robustness. For instance, we strictly followed data handling protocols to protect sensitive information, including anonymizing patient identifiers. Additionally, we conducted regular bias assessments to monitor potential imbalances in the data. In practice, these strategies have been largely effective; for example, data leakage risks were minimized, and initial testing showed minimal disparities in predictions across demographic groups. However, certain aspects, like maintaining complete model transparency, have proven challenging due to the complex architecture of ResNet50. We plan to address this in future iterations to further mitigate interpretability risks.

- **Trustworthiness Strategies and Reflections**
  To build trustworthiness into our model, we focused on several strategies, including incorporating explainable AI methods and setting rigorous accuracy benchmarks. We chose these approaches to help end-users, particularly clinicians, feel confident in the model's outputs. In practice, applying explainability techniques, such as gradient-based attribution, has been helpful in allowing clinicians to visualize which knee regions the

model focuses on, reinforcing trust in predictions. This has strengthened stakeholder confidence in the model. However, we recognize the need for additional transparency measures to bridge any remaining interpretability gaps, especially in borderline cases where severity is harder to classify.

- **Reflection on Residual Risks**
  Despite these strategies, certain residual risks remain. For example, there is a possibility of model bias toward underrepresented demographics due to limited data from specific age groups or ethnic backgrounds. Additionally, the model's complexity can limit transparency in decision-making, which could impact clinician acceptance in a practical setting. We are aware of these limitations and are exploring ways to further mitigate these risks, such as by expanding our dataset or integrating more interpretable model components.

- **Outcomes of Data Quality Validation**
  We took several steps to validate the quality of the dataset, including checks for missing data, analyzing data distribution, and performing data preprocessing, such as normalization. These processes revealed some minor inconsistencies, particularly with image resolution and brightness variations, which we addressed through standardization techniques. This validation has positively influenced the model's performance, reducing prediction variability. The effort to maintain high data quality was crucial, as it ensured that the model could focus on genuine indicators of osteoarthritis severity rather than noise from poor-quality images.

Step-by-Step Residual Risk Assessment

Step 1: Identify Residual Risks

Despite mitigation strategies implemented throughout the AI lifecycle, residual risks remain. Potential examples in our project could include:

- Algorithmic Bias: Potential bias against specific demographics in the osteoarthritis predictions.
- Model Drift: Risk of the model degrading in performance over time due to changing patient data patterns.
- Data Privacy Issues: Risk of patient data privacy breach.

Step 2: Estimate the Likelihood of Each Risk

Each residual risk is assessed based on how likely it is to occur:

- **Probable**: High likelihood of occurrence
- **Possible**: Moderate likelihood of occurrence
- **Improbable**: Low likelihood of occurrence

Step 3: Assess the Impact of Each Risk
Determine the level of impact each residual risk could have on the project:

- **Acceptable - Low Impact**: Minimal disruption (e.g., minor performance drop).
- **Tolerable - Moderate Impact**: Manageable but noticeable issues (e.g., some degradation in prediction accuracy).
- **Unacceptable - High Impact**: Significant negative effects (e.g., major bias affecting patient diagnosis accuracy).
- **Intolerable - Critical Impact**: Catastrophic consequences (e.g., severe regulatory issues or system failure).

Step 4: Plot the Risks on the Matrix
Each risk is positioned on the Likelihood vs. Impact Matrix based on Steps 2 and 3. For example:

- **Model Drift**: Possible likelihood and Tolerable Impact, placed in the Possible \ Tolerable cell (Yellow).
- **Algorithmic Bias**: Probable likelihood and Unacceptable Impact, placed in the Probable \ Unacceptable cell (Orange).
- **Data Privacy Issues**: Possible likelihood and Intolerable Impact, placed in the Possible \ Intolerable cell (Red).

Step 5: Evaluate the Risk Levels
Using the matrix color coding (Green, Yellow, Orange, Red):

- **Green (Low Risk)**: Generally acceptable with minimal or no additional action; regular monitoring.
- **Yellow (Moderate Risk)**: Some mitigation may be necessary; acceptable with monitoring.
- **Orange (High Risk)**: Mitigation actions should be prioritized, and regular monitoring is required.
- **Red (Critical Risk)**: Immediate action is required to eliminate or significantly reduce the risk.

Step 6: Determine Mitigation or Acceptance Actions
For each residual risk, determine if additional mitigation is needed or if the risk can be accepted as is:

- **Low Risks**: Acceptable, but monitor to prevent escalation.
- **Moderate Risks**: Additional mitigation and regular monitoring are advised.
- **High Risks**: Prioritize mitigation and establish frequent monitoring protocols.
- **Critical Risks**: Immediate mitigation is essential; risks must be actively managed to prevent significant impact.

Data Collection Management and Report

1. Data Type

- **Description:** Primarily X-ray images with demographic metadata (e.g., age, gender). This combination allows the model to assess both visual and demographic factors affecting osteoarthritis severity.

2. Data Collection Methods

- **Sources:** Data from open-access repositories like Kaggle's knee osteoarthritis dataset and OAI.
- **Methods:** Automated data extraction and manual filtering for quality control to maintain image consistency and resolution.

3. Compliance with Legal Frameworks

- **Applicable Laws:** HIPAA and GDPR are relevant due to the sensitivity of medical images and demographic information.
- **Compliance Strategy:** Apply anonymization, patient consent protocols, and secure data storage solutions. Regular audits ensure compliance throughout the AI lifecycle.

4. Data Ownership and Access Rights

- **Ownership:** Open-access dataset usage with proper attribution to source databases.
- **Access Control:** Only authorized team members have access to the raw data, with access logs to monitor data usage.

5. Metadata Management

- **Content and System:** Metadata includes age, gender, and imaging settings. Metadata is stored alongside images for ease of model input processing.

6. Data Versioning

- **Version Control:** `DVC` is employed for dataset versioning, ensuring each version of data used for training is logged and recoverable, enhancing reproducibility.

7. Data Preprocessing, Augmentation, and Synthesis

- **Preprocessing:** Image resizing, normalization, and contrast adjustments ensure consistency. Scaling and standardization for demographic data.
- **Data Augmentation:** Includes rotation, flips, and zooms to create diversity. This approach addresses demographic imbalances and improves generalization.

8. Data Management Risks and Mitigation

- **Risks Identified:** Privacy concerns, data quality, and demographic bias.
- **Mitigation:** Data privacy via anonymization and controlled access; demographic balancing through augmentation.

9. Data Management Trustworthiness and Mitigation

- **Trustworthiness Strategies:** Data quality validation through automated checks and augmentation. Privacy ensured through secure handling protocols.
- **Reflection:** These measures have proven effective, with minimal discrepancies in data quality or demographic coverage. Continuous review of access controls is in place to adapt to evolving requirements.

Model Development and Evaluation

Model development and evaluation are crucial stages in our AI lifecycle, where we focus on optimizing performance, fairness, and reliability for accurate knee osteoarthritis severity predictions. During this phase, we carefully selected algorithms, tuned model parameters, and applied comprehensive evaluation metrics to ensure the model meets project goals and generalizes effectively to real-world knee X-ray data. In this section, we outline the methods, strategies, and practices used to develop and evaluate the model, focusing on performance optimization, interpretability, and fairness to address real-world use cases and stakeholder expectations.

1. Model Development

Algorithm Selection

For this project, we selected **ResNet50**, a convolutional neural network (CNN) architecture, due to its strong performance on image-based tasks and its ability to capture intricate details in knee joint X-rays. ResNet50's use of residual connections makes it particularly suitable for handling deep networks without vanishing gradients, which is essential given the complexity of knee joint structures in OA images. After testing simpler architectures and comparing them to ResNet50, we found this model best suited for accurately predicting OA severity levels in line with our project objectives.

Feature Engineering and Selection

To improve predictive power, we incorporated relevant features derived from preprocessing the X-ray images, such as image normalization, edge detection, and joint-specific focus areas. While ResNet50 relies heavily on its CNN layers to identify features, we explored additional image augmentations (e.g., rotation, zoom) to enhance model generalization. These engineered features were essential in ensuring the model could effectively distinguish various OA severity levels, even with variations in image quality.

Model Complexity and Architecture

ResNet50, with its **50 layers** and numerous convolutional and pooling operations, balances performance and computational efficiency, making it ideal for complex image classification tasks. This architecture was chosen for its ability to extract fine-grained features without the computational burden of more complex models. We experimented with minor modifications, such as adjusting the final fully connected layers to tailor the model output for OA classification, but retained the overall architecture for consistency and optimal performance.

Overfitting Prevention

We implemented **regularization techniques** such as dropout layers, L2 regularization, and data augmentation (e.g., varying lighting conditions and random cropping) to prevent overfitting. These strategies were effective in ensuring the model remained robust and generalizable across both training and validation datasets.

2. Model Training

Training Process

For training, we used batch sizes of **32**, a learning rate of **0.001**, and **Adam optimizer**. We trained the model over **50 epochs**, monitoring loss and accuracy to ensure a balance between model performance and training efficiency. This training configuration helped achieve stable convergence while avoiding issues like gradient explosion or excessively long training times.

Hyperparameter Tuning

Key hyperparameters like **learning rate, dropout rate, and weight decay** were fine-tuned using a **random search** approach. Through these trials, we found that a learning rate of 0.001 and a dropout rate of 0.5 provided optimal model stability and reduced overfitting. Regular monitoring of the loss and validation metrics allowed us to identify potential underfitting or overfitting trends and adjust accordingly.

3. Model Evaluation

Performance Metrics

We evaluated the model using metrics aligned with our project's objectives, including **accuracy**, **F1 score**, and **AUC-ROC**. These metrics were chosen for their relevance to classifying OA severity levels, where both precision and recall are crucial for balanced performance. Our model achieved an accuracy of X%, F1 score of Y%, and AUC-ROC of Z%, indicating strong predictive ability while maintaining fairness across severity levels.

Cross-Validation

We used **k-fold cross-validation** (k=5) to assess model performance and ensure its reliability across diverse data subsets. This technique allowed us to confirm consistent model accuracy and stability, with minimal variability across folds, affirming the robustness of our model for real-world application.

4. Implementing Trustworthiness and Risk Management in Model Development

Risk Management Report

Throughout model development, we identified several risks, including potential data biases and interpretability challenges. We mitigated these by ensuring diverse representation in training data and using explainable AI methods like Grad-CAM to visualize decision areas on X-ray images. These strategies were effective in reducing interpretability concerns and highlighted potential biases, particularly for underrepresented patient demographics. Future iterations will address these issues by expanding our dataset and enhancing transparency measures.

Trustworthiness Report

To ensure model trustworthiness, we focused on transparency, fairness, and performance consistency. Explainability techniques allowed us to provide clinicians with visual insights into the model's focus areas for each prediction. This helped clinicians understand the model's reasoning, fostering trust in the model's results. We also identified areas for improvement, such as enhancing the clarity of decision boundaries for borderline OA cases, which will guide further refinement.

5. Applying HCI Principles in AI Model Development

Developing Interactive Prototypes

We used **Streamlit** to create a simple, interactive interface allowing users to test and interact with the model in real-time. Through interactive components like sliders and input fields, clinicians can adjust parameters and observe how the model's predictions change, offering immediate feedback and enhancing user engagement.

Designing Transparent Interfaces

To enhance interpretability, we integrated visualization tools like **matplotlib** to display feature importance and decision confidence scores. These visual explanations provide clinicians with transparent views of the model's decision-making process, promoting trust in model predictions.

Creating Feedback Mechanisms

The interface includes a feedback section where users can provide thumbs-up/down ratings or add comments, allowing us to collect valuable insights into the model's practical performance. This feedback will guide iterative improvements to both the model and interface, ensuring the system remains responsive to end-user needs and expectations. We also thinking to add a google forms survey from that we can capture the users experreience.

Deployment and Testing Management Plan

Deployment and testing are critical phases of the **AI-Enhanced Diagnosis and Severity Prediction of Knee Osteoarthritis** project. Leveraging **Streamlit** as the deployment platform ensures an accessible, interactive interface for clinicians and patients, making it easier to visualize and interpret AI predictions. This section outlines a streamlined deployment and testing plan tailored for deploying the ResNet50 model on Streamlit.

1. Deployment Environment Selection

The choice of **Streamlit** as the deployment platform provides simplicity and accessibility, aligning with the project's goal of creating an interactive and user-friendly application for knee osteoarthritis severity prediction.

Documentation:

- **Environment Type: Streamlit application**.
- **Justification:**
  - Streamlit offers an intuitive interface for visualizing predictions and interacting with the model.
  - Its lightweight nature makes it ideal for rapid deployment without the overhead of complex cloud infrastructure.

Deployment Environment Options:

- **Local Deployment:** For testing demonstrations.
- **Cloud Hosting:** Deploying via **Streamlit Community Cloud** for wider accessibility.

2. Deployment Strategy

A streamlined deployment strategy ensures that the application runs seamlessly on Streamlit with a focus on ease of access and interactivity.

Documentation:

- **Strategy:** Direct deployment to **Streamlit Community Cloud**, with proper setup for handling model files and user interactions.
- **Benefits:** This approach minimizes setup time while providing a live, shareable application URL.

Steps:

1. Develop the Streamlit app locally, integrating ResNet50 for knee OA severity prediction.
2. Deploy to Streamlit Community Cloud with version-controlled updates via GitHub.

3. Security and Compliance in Deployment

Ensuring the security and privacy of sensitive healthcare-related data is a priority, even on lightweight platforms like Streamlit.

Documentation:

- **Security Measures:**
  - Avoid storing sensitive user data; process data in memory without persistence.
  - Use environment variables for sensitive API keys or configuration settings.
- **Compliance Measures:**
  - Include disclaimers highlighting the app's educational or informational purpose.

4. CI/CD for Deployment Automation

While Streamlit deployment is straightforward, adopting basic CI/CD practices ensures smooth updates and consistency.

Documentation:

- **CI/CD Tools:** GitHub Actions for version control and automated updates to the Streamlit app.
- **Approach:**
  - Automate deployments upon committing changes to the GitHub repository.
  - Use Streamlit's integration with GitHub for continuous updates.

`This is for the future plan. Currently we are not using any CI/CD`.

5. Testing in the Deployment Environment

Testing ensures the Streamlit application delivers accurate and responsive results to users.

Documentation:

- **Tests Conducted:**
  - Functional tests to validate model predictions and UI interactions.
  - Usability testing with simulated user inputs to ensure seamless navigation.
- **Tools:** Manual testing through the Streamlit app interface and basic API testing with **Postman**.

Evaluation, Monitoring, and Maintenance Plan

To ensure the **AI-Enhanced Diagnosis and Severity Prediction of Knee Osteoarthritis** application remains effective, user-friendly, and aligned with project goals, a detailed plan for evaluation, monitoring, and maintenance is essential. This plan addresses system performance, user feedback, compliance, and model updates.

1. System Evaluation and Monitoring

Maintaining the health of the deployed Streamlit application is critical for ensuring consistent user experience and reliable predictions.

Documentation:

- **Monitoring Tools:**
    - **Built-in Streamlit Logs:** Used to identify application-level errors, debug issues, and log performance metrics.
    - **Custom Logging:** Implement custom Python logging to capture metrics such as API response times, memory usage, and unusual user interactions.
    - **Manual Checks:** Periodically test the application to ensure that features, such as model prediction and data visualization, are functioning as expected.
- **Metrics Tracked:**
    - **Latency:** Monitor the time taken for the app to load and for predictions to be generated. This helps ensure that the app provides a smooth user experience.
    - **Prediction Accuracy:** Periodically validate model predictions against a labeled subset of test data to ensure the model remains effective in diagnosing and predicting knee OA severity.
    - **User Feedback:** Analyze user-reported issues or concerns to identify potential areas for improvement.

Implementation Example:

- Use Streamlit’s cache functionality to optimize data and model loading times, reducing latency.

2. Feedback Collection and Continuous Improvement

User feedback is integral to refining the application's functionality and user interface, ensuring that it meets user expectations and requirements.

Documentation:

- **Feedback Mechanisms:**
    - **Integrated Feedback Form:** Add a simple feedback form directly into the Streamlit interface, allowing users to report issues or suggestions in real time. Utilize Python libraries like `streamlit_textarea` for text input fields.
    - **Optional External Feedback:** Provide links to detailed feedback forms hosted on platforms like Google Forms or Typeform for collecting structured and actionable user feedback.

Implementation Steps:

- **Step 1:** Add a “Feedback” button on the app interface to toggle the feedback form.
- **Step 2:** Store feedback submissions securely in a database (e.g., SQLite) or forward them via email to the development team.
- **Step 3:** Periodically review feedback to prioritize enhancements or bug fixes.

Example Questions for Feedback Form:

- “Was the prediction result easy to interpret?”
- “Did you experience any delays or errors while using the app?”
- “What features would you like to see added in the future?”

3. Maintenance and Compliance Audits

Routine maintenance ensures the application remains functional and adheres to best practices, while compliance audits confirm adherence to data protection and project goals.

Documentation:

- **Maintenance Schedule:**
    - **Weekly:**
        - Perform basic functionality checks (e.g., prediction generation, UI responsiveness).
        - Review logs for errors or performance degradation.
    - **Bi-Monthly:**
        - Collect and analyze user feedback.
        - Update the app based on feedback and any new feature requirements.

- **Tools Used:**
  - **GitHub:**
    - Maintain version-controlled updates to the codebase.
    - Track issues, enhancements, and update logs.
  - **Streamlit Community Cloud Dashboard:** Monitor application deployment health and uptime.
- **Compliance Measures:**
  - Ensure the app complies with data protection regulations (e.g., GDPR for European users).
  - Verify disclaimers and user instructions are clear and accessible.

4. Model Updates and Retraining

Regular model updates and retraining ensure that the ResNet50 model remains relevant and accurate as new data becomes available.

Documentation:

- **Retraining Strategy:**
  - Collect new annotated X-ray images periodically from trusted sources.
  - Retrain the model using a pipeline built in **PyTorch** or **TensorFlow** with proper validation and hyperparameter tuning.
  - Evaluate the updated model against a holdout test set and compare its performance with the previous model version.
- **Integration into Streamlit:**
  - Update the model file (.pth or .h5) in the Streamlit app repository.
  - Test the app locally to confirm compatibility and performance before deploying the updated version to Streamlit Community Cloud.
- **Version Control:**
  - Use Git tags (e.g., `v1.0`, `v1.1`) to document model updates.
  - Maintain a changelog summarizing new model versions, key changes, and their impact on performance.

Github Repository Link

Link:https://github.com/teja2002/AI-ENHANCED-DIAGNOSIS-AND-SEVERITY-PREDICTION-OF-KNEE-OSTEOARTHRITIS.git