**AI Systems Project Proposal**

**Project Title:**

DermAI – Innovative AI model for early Skin Cancer detection.

**Project Overview:**

* **Project Objective:** Skin cancer if not detected at an early stage, it may prove fatal, particularly when a significant portion of the population does not have access to dermatological treatment because of insufficient resources, insurance problems, or poor infrastructure. Uninsured patients have difficulty in accessing specialty care, including dermatology. Over 40% of US citizens live in areas underserved by dermatologists. Teledermatology allows providers to diagnose and recommend treatment and address the limited dermatology specialty care access in underserved population. One of the main teledermatology applications is to triage dermatology patients with higher morbidity and mortality risks to facilitate earlier in-person visits. By identifying benign or malignant lesions from low-quality photographs taken with a smartphone camera the project aims to supplement the dermatologists and people.

The procedure currently in-practice involve patient to visit dermatology clinic where the affected area is examined by an expert using a dermatoscope. Dermatoscope is a medical instrument often found only in dermatology clinics that combines light and magnification to improve vision and highlight morphological details typically invisible to the unaided eye. The expert uses their experience and reliance on visual signals to draw conclusions. The project proposes the use of AI model to process low-quality images and a detailed information describing the lesion to compliment the current practice, triage the patients and aid in early detection of skin cancer (size, color, position, etc.) and if the results are either inconclusive or affirmative, the algorithm suggests seeing a dermatologist. The metric (reasons why algorithm believes it's positive / inconclusive) might also be used by the doctor as a heads-up.

* **Project Scope:** Identification of skin cancer using a low-quality image of the lesion and the description of the lesion provided by the user (optional but recommended). The model emits a list of reasons why it thinks the lesion is either benign or malignant. This process doesn’t replace dermatologists. The system would accept an image and a description form would be provided to the user to describe the affected area (optional but recommended). The system analyzes the image and description and predicts whether the lesion is benign or malignant. Additionally, it emits a list, trying to reason its prediction (reasons based on model’s learning only).

Triaging application: Which individuals need to see a clinician.

Beneficiaries : Under-served population

Benefits : Combat the lack of infrastructure and finances.

Early detection : Algorithms should be adept to evaluating lower quality images

Beneficiaries : General population / people in primary care / clinicians

Benefits : Process optimization, Early detection

* **AI Techniques and Tools:** CNN for Image classification, NLP techniques for Examining descriptions and concluding topics / specific mentions and Classification Models such as Logistic Reg / Random Forest also Data Synthesizing to counter class imbalance.

**Stakeholders:**

* **Project Team:**

Vivek Hiremath, Nikhil Reddy

* **End Users:**

General public, Clinicians

* **Other Stakeholders:**

Healthcare Providers: Doctors and Tele dermatology Experts.

Investors and Financial Stakeholders: People supporting the project.

Technology Partners: People helping with core terminologies related to the medical machines and its software.

**Computer Infrastructure Considerations:**

1. Project Needs Assessment
   * Objective: The project aims to detect skin cancer through low-quality images and textual descriptions, addressing a significant healthcare challenge.
   * Performance Benchmarks: Sensitivity: Targeting a minimum of 75%, recognizing the need for high sensitivity in medical applications.

Specificity and Precision: Including precision and F1-score as additional metrics to comprehensively assess model performance.

Confidence Interval: Strive for a shorter confidence interval to enhance robustness.

* + Challenges: Model Deployment: Ensuring compliance with HIPAA and GDPR for European users. Conduct an ethical review to guide handling sensitive data.

Power Consumption: Assess strategies to minimize the carbon footprint associated with high resource demand.

Network Stability: Planing for reliable network connections to support cloud-based operations.

* + Interpretability: Specifying methods for enhancing model interpretability to foster trust among medical professionals.

1. Hardware Requirements Planning
   * Infrastructure: Choosing high-performance GPUs like A100 and V100 for training and inference.
   * RAM Requirements: Specifying necessary RAM for each server or cloud instance to handle memory-intensive medical image processing.
   * Backup Solutions: Outlining backup power solutions and an emergency plan for data redundancy, recognizing the critical nature of healthcare services.
2. Software Environment Planning
   * Compatibility: Ensuring the model is compatible with the cloud service OS and specify a front-end framework (e.g., React or Vue.js) for web access.
   * Software Stack: In addition to core libraries (Pandas, NumPy, Scikit-Learn), explore specialized medical imaging libraries (e.g., SimpleITK or ITK) to enhance model capabilities.
   * Docker: Utilizing Docker for consistency across development and production environments, ensuring portability.
3. Cloud Resources Planning
   * Cloud Services: AWS SageMaker is suitable for model training and deployment, with Azure Cloud as a flexible alternative.
   * Disaster Recovery: Including a disaster recovery plan with multi-region storage and automated backups to ensure data availability during outages.
   * Data Security: Specifying encryption methods and access control policies to ensure compliance with HIPAA for sensitive medical data.

1. Scalability and Performance Planning
   * Model Testing: Implementing strategies for experimenting with various models and parallelizing training across multiple GPUs.
   * Load Testing: Using tools like Locust to simulate real-world traffic scenarios, ensuring the system can handle increased demand effectively.
   * Automated Alerts: Setting up alerts for performance monitoring to quickly address

latency or error rate spikes.

**Security, Privacy, and Ethics (Trustworthiness) Considerations:**

1. Problem Definition:
   * Stakeholder Engagement: Engaging patients, dermatologists, and community organizations through surveys to understand concerns about skin cancer detection.
   * Ethical Guidelines Review: Collaborate with ethicists and legal experts to establish guidelines for data handling, privacy, and fairness.
2. Data Collection:
   * Informed Consent: Implementing a robust consent process through secure web interfaces that clearly outline data collection and usage.
   * Data Encryption: Employ encryption techniques to protect sensitive data, ensuring regulatory compliance.
3. AI Model Development:
   * Fairness and Bias Mitigation: Using Fairlearn and AIF360 to analyze and mitigate biases in training datasets, ensuring equitable model performance.
   * Transparency: Implement explainable AI techniques (e.g., SHAP) to clarify model predictions for users.
4. AI Deployment:
   * Secure Practices: Implementing HTTPS for data transmissions to protect sensitive information.
   * Regular Security Audits: Conducting routine assessments to identify vulnerabilities and enhance user confidence in the system.
5. Monitoring and Maintenance:
   * Continuous Ethical Review: Establish a framework for ongoing evaluations of AI performance and user experiences, gathering feedback to inform improvements.
   * Anomaly Detection: Utilize libraries like Scikit-learn for monitoring unexpected model behavior, allowing for timely corrections.

**Human-Computer Interaction (HCI) Considerations:**

1. Understanding User Needs:
   * User Research: Conducting specific surveys and interviews to uncover challenges patients face in early skin cancer detection and the tools healthcare professionals currently use.
   * Key Aspects: Exploring users’ technological literacy, ease of access to services, and emotional stress associated with waiting for results.
2. Creating User Personas and Scenarios:
   * User Personas: Developing detailed personas like “The Anxious Patient” and “The General Practitioner,” outlining their specific struggles and needs.
   * Scenarios: Illustrate situations where these personas encounter roadblocks, such as issues with image quality or understanding diagnostic results, and how DermAI can address them.
3. Conducting Task Analysis:
   * Task Breakdown: Detailing steps users will take to achieve objectives, such as capturing and uploading images, and the obstacles they may face at each step.
   * Common Mistakes: Identifying frequent errors users make (e.g., unclear image capture) and outline system guidance to mitigate these issues.
4. Addressing Accessibility Requirements:
   * Specific Needs: Addressing how DermAI will support users with low vision and integrate assistive technologies for cognitive disabilities.
   * Performance Metrics: Setting clear accessibility performance metrics, such as ensuring 95% of users can complete tasks without assistance.
5. Defining Usability Goals
   * Specific Metrics: Establishing concrete performance metrics, such as target time for image uploads and accuracy rates for user interpretation of results.
   * User Satisfaction: Aiming for an 85% satisfaction score in post-use surveys to ensure the system effectively meets user needs.

**Risk Management Strategy:**

**Data Collection and Preprocessing:** Data quality and security is of utmost importance in an Artificial intelligence System. If any of these are compromised, it could deeply impact how a system performs and its trustworthiness. Below are a few risks related to data collection and

management with respect to our project that we have identified **Risks:**

1. Data Bias: The data we collect might be skewed towards a certain demographic or skin tones, leading to a biased prediction.
2. Data Privacy: Since we are dealing with medical data, user privacy is a huge aspect. Additionally, we need to comply with the regulations of government agencies controlling / validating medical data privacy (like HIPPA for eg.).
3. Class Imbalance: An imbalance in target class could cause problems with the training phase and heavily impact the outcome of the system. Since we are already dealing with lower quality images, having a heavy class imbalance just steepens the learning curve.

**Management Strategies:**

1. Bias Solving: We will have to manually assess the data for underrepresentation of skin tone (Better if we could automate this process because there is around 300,000 + samples and no background is available). Evaluating the sensitivity / specificity of the results might also give us some insight into this. Additionally, synthesizing data for the underrepresented skin tone would help us. Same applies to the text data.
2. Privacy Assurance: Randomizing the data with the help of an algorithm would help us ensure that a breach won’t give up all the available information, however, this is a reactive approach. A proactive approach would be to work with a secure cloud platform and ensure that there are no breaches in the first place. We are utilizing Google’s data storage service Google Drive for this purpose. And finally, we found that HIPPA is one of the regulating authorities whose compliances we would follow.
3. Data Quality: The data quality is low (however, this is the project scope so its fine). The data has a significant class imbalance and is presenting problems with model training. To solve this problem, we have tried out different classification models with a variety of hyperparameters and performed hyperparameter tunning to obtain the best results we could. Additionally, we plan to augment data for the underrepresented class to help the model effectively learn the difference.

**Model Development:**

**Risks:**

1. Model Bias: Since we plan to utilize the power of transfer learning (use ResNet model) the model might have an inherent bias that might misclassify / might favor a certain skin tone. This also includes the formatting and other things that a pretrained model might require (like in case of ResNet the size must be (224,224), this reduces the resolution of the already low-quality images resulting in information loss).
2. Irregular Learning: The data available for this project is significantly imbalanced and this might result in the model learning only one class. This means if we use accuracy as the evaluation matrix, the model could just classify every sample as the dominant class and get away with above average accuracy.
3. Difficult to Explain: A CNN model (pretrained / new) is difficult to explain. This might cause distress within the users / experts and raise doubts about the process.

**Management Strategies:**

1. Eliminating Model Bias: To reduce model bias, we avoid relying on the metrices like accuracy that could be misleading. Instead, we interpret the effectiveness of the model from the sensitivity, specificity, recall, etc that give us specific information about the model performance. Then we combine this with domain expertise to select the best suited one. Additionally, we have trained multiple models and performed hyperparameter tuning to obtain the best results we could.
2. Avoiding Irregular Learning: To avoid the model learning only one class we could synthesize data for the underrepresented class. In addition to data synthesis, we would employ methods like cross-validation and regularization to counter overfitting.
3. Model Explainability: To enhance explainability of the CNN & the classification model based on textual data, we would rely on libraries like LIME / SHAP. This would avoid raising suspicion from experts. This might also help us during the testing phase where experts review the system.

**Model Deployment:**

**Risks:**

1. Data Privacy & Security: Dealing with medical data puts data security at the top of the risk management pyramid. Any sort of data breaches could be catastrophic for the system and the organization handling the system.
2. Performance & Scalability: Since this is not an edge system, it might struggle with scalability issues like sudden increase in the incoming traffic, thus, slowing the real-time application.
3. Latency Issues: The system being a cloud-based system might suffer from delays due to network instability, thus leading to poor user experience.

**Management Strategies:**

1. Addressing Privacy Issues: Privacy issues would be addressed by using a safe cloud-based server with strict access controls, good encryption and regular audits to ensure HIPPA standards are met.
2. Performance Monitoring & Scaling: Performance monitoring could be automated by employing a threshold for evaluation matrix. Additionally, to ensure the data drift has not skewed the model, we could conduct regular audits. Since this is a cloud based systems, the problem for scalability (on-premises, updating the hardware, etc) is avoided (assuming the cloud service provider also provides on-demand scalability).
3. Avoiding Latency Related Issues: Using servers across different regions would help reduce this issue.

**Monitoring & Maintenance:**

**Risks:**

1. Data Drift: The data is known to change over time (this might be due to the changes in requirement or effect of other factors). This throws the model off its trajectory, reducing its effectiveness.
2. Bias: The data drift could introduce bias again into the model.

**Management Strategies:**

1. Data Drift Monitoring: Monitoring data drift could be automated by setting up an algorithm that tracks the model’s primary evaluation matrix (could be multiple), setting up a desired / acceptable threshold below which the model would render useless (in our case not reliable), and finally add a trigger that alerts the team about the drift.
2. Bias Monitoring: This is a slightly different process as it is not easy to track without EDA at times. Sometimes the bias affects the accuracy, and this would trigger the alert, however, at most times the bias has an exponential effect which could have a sudden effect on the performance of the model. This also affects the trustworthiness of the model on a large scale.

**Human-Computer Interaction (HCI):**

**Risks:**

1. User Trust: Users might have suspicions about relying on an AI system for health-related decisions.
2. Usability Issues: If the UI is not explanatory / user friendly this could lead to a lot of incorrect user engagement and misinterpretation of information (incomplete tasks or completion of tasks in a wrong way, etc. Also misinterpreting the information could lead to panic / dissatisfaction of users).
3. Accessibility Barriers: People with disabilities might have problems accessing / understanding the UI.

**Management Strategies:**

1. Addressing Trust Issues: This issue could be reduced to some extent by explaining the similarities in the processes followed by the dermatologists and AI system. Additionally, using a threshold that plays it safe (prioritizes on classifying the benign lesions, intending to avoid misclassifying a malignant lesion as benign at all costs) would reduce the risk the system would carry. Since we are relying on 2 models image based and text based, this should act as a 2nd opinion (expect it to, cannot validate this claim at this point of time). Clearly communicating the limitations of the model would again help us reduce the ethical risk the system carries.
2. Usability Testing: Receiving reviews from real users to improve the UI. This would also help ensure user friendliness of the application. Reduce interactions, this would reduce the possibility of a task going wrong to some extent. Utilize the power of visual cues to convey tasks / messages.
3. Accessibility: Following the WCAG guidelines would help us ensure that the platform is accessible to users with disabilities. We would also have to incorporate screen readers / language options and customizable UI options like color contrasts, etc.

**Ethical Considerations:**

**Risks:**

1. Biased Output: The model might be misdiagnosed based on skin-tone, demographics, gender, age, etc.
2. Ethical Dilemmas: An incorrect classification might result in the users either panicking / avoiding a much-needed visit to the clinician.
3. Privacy Issues: Misuse of confidential information about an individual would lead to significant consequences.

**Management Strategies:**

1. Auditing: Biases could be countered with regular auditing to ensure fairness of the system. 2. Ethical Frameworks: We could implement ethical frameworks like AI360 toolkit to continually assess potential ethical risks related to fairness / transparency & privacy.

3. Privacy: Utilizing anonymization techniques such as randomizing / encrypting data would prevent loss of confidential information.

**Data Collection Management and Report:**

1. **Data Type:** The project will primarily collect image data (low-quality photographs of skin lesions) and textual data (descriptions of the lesions provided by users). Additionally, demographic information (age, gender) may be collected to ensure diverse representation in the dataset.

1. **Data Collection Methods:** Data will be collected through a web-based interface where users can upload images and fill out descriptive forms. We may also gather data from existing medical records (with consent) and collaborate with healthcare providers for anonymized datasets to improve the model’s training and also public datasets (e.g., Kaggle, UCI Machine Learning Repository) for backup if we couldn’t achieve data collection from above method.

**Ingestion for Training:**

* + **Data Ingestion Process**: The training data is ingested using custom packages to handle batch processing efficiently. The data is preprocessed to normalize images and extract features.
  + **ETL Tools**: Extract, Transform, Load (ETL) tools facilitate the integration of diverse data sources into a unified dataset, improving the quality and efficiency of training.
  + **Optimizations**: Techniques like data augmentation are applied during training to enhance model robustness and address class imbalances, resulting in improved model performance.

**Ingestion for Deployment:**

* + **Deployment Ingestion Plan**: During deployment, APIs (e.g., REST) will be utilized for real-time image submissions from users, allowing for immediate analysis.
  + **Message Queues**: Message queues (e.g., RabbitMQ or Kafka) will manage incoming data streams effectively, ensuring smooth processing of user requests and maintaining system responsiveness.
  + **Batch Processing**: Scheduled batch processing may also be implemented for periodic updates and analysis of aggregated user data, enhancing system insights and model refinement.

1. **Compliance with Legal Frameworks:** The project will comply with **HIPAA (Health Insurance Portability and Accountability Act)** regulations to ensure the protection of patient health information. We will implement secure data handling practices and obtain informed consent from users prior to data collection. Legal counsel will be consulted to ensure full compliance with local and national laws regarding health data.

**CCPA (California Consumer Privacy Act)**: For users in California, CCPA compliance is necessary, which includes providing users with clear information about data collection practices and their rights regarding personal data.

**ISO/IEC 27001**: This standard provides a framework for establishing, implementing, maintaining, and continuously improving an information security management system (ISMS). Compliance ensures that DermAI follows best practices for managing sensitive information. **NIST Standards**: Following NIST guidelines for cybersecurity will help DermAI establish strong security measures to protect data against breaches and other cyber threats.

**Compliance Strategy and Results:**

* 1. **Anonymization and Data Security**:
     + DermAI employs data anonymization techniques to protect user identities. All patient images and descriptions are stripped of personally identifiable information before storage and processing. o Robust encryption protocols are implemented to secure data in transit and at rest, ensuring that sensitive information remains protected.
  2. **Informed Consent Protocols**:
     + An informed consent process is established, requiring users to explicitly agree to data collection practices. This is facilitated through clear consent forms that outline the purposes of data usage and the rights users hold over their data.
  3. **Regular Audits and Monitoring**:
     + Regular security audits are conducted to ensure compliance with HIPAA and other relevant laws. This includes assessing data handling practices and reviewing access logs to verify that only authorized personnel can access sensitive information. o An internal review framework is set up to continuously monitor adherence to compliance standards and address any discrepancies promptly.

1. **Challenges and Resolutions**:

o **Challenge**: Ensuring full compliance with diverse regulations across different jurisdictions, especially concerning data collection and user consent. o **Resolution**: The team consulted legal experts specializing in data privacy to develop a comprehensive compliance framework that addresses the requirements of HIPAA and CCPA. Additionally, user feedback mechanisms were implemented to adapt the consent process to meet users' needs effectively.

1. **Data Ownership:**

**Ownership and Access Control:**

* 1. **Data Ownership**: Users retain ownership of their personal data, including images and descriptions of skin lesions submitted to DermAI. The platform operates under the principle that users have full rights to their data and can request its deletion or modification at any time.
  2. **Access Rights**: Access rights are clearly defined based on user roles within the system. Authorized personnel including healthcare providers and developers are granted access based on necessity, ensuring that only those who require access to perform their duties can view sensitive information.
  3. **Permissions Process**: A formal permissions process will be established, requiring users and staff to submit requests for data access. Each request is reviewed and approved based on the role and purpose of the data access, maintaining a strict oversight mechanism.
  4. **Access Logging**: All access to user data is logged in a secure database, capturing details such as the user ID, timestamp, and nature of the access (e.g., view, edit). This logging ensures accountability and allows for tracking any unauthorized access attempts.
  5. **Secure Authentication Mechanisms**: Two-factor authentication (2FA) is being implemented for all users accessing the system, adding an extra layer of security. This ensures that even if credentials are compromised, unauthorized users cannot gain access without the second factor of authentication.
  6. **Periodic Access Audits**: Regular audits will be conducted to review access logs, ensuring compliance with access control policies and identifying any irregularities. This helps maintain the integrity of data access and usage.
  7. **Data Usage Agreements**: Data usage agreements will be established with all stakeholders who may access user data, clearly outlining the permitted uses, restrictions, and responsibilities regarding data privacy and security.

**Lessons Learned:**

* 1. **Effectiveness of Access Controls**: Initial implementation highlights the importance of stringent access controls, as early testing revealed potential vulnerabilities in data access management. By refining the permissions process and integrating 2FA, the system significantly improved its security posture.
  2. **Improvements Made**: Continuous feedback from stakeholders will leed to enhancements in the permissions process, streamlining requests for access while maintaining robust security measures. The introduction of automated alerts for suspicious access patterns further bolstered security, allowing for timely responses to potential breaches.
  3. **User Awareness**: Educating users about their rights and the importance of data privacy was crucial. This achieves greater user engagement and trust in the system.

1. **Metadata:** Metadata will include information about the images (e.g., resolution, file format) and descriptions. This will aid in organizing the dataset, improving searchability, and facilitating future analyses.

**Type of Metadata Managed:**

DermAI manages several types of metadata, including:

* + **Data Source:** Information about where the data originated, such as patient submissions or publicly available datasets.
  + **Timestamp:** Dates and times when the data was collected or modified, providing context for each dataset version.
  + **Format:** Details on the file formats used (e.g., JPEG for images, CSV for tabular data) to ensure compatibility during processing.
  + **Version:** Versioning information linked to the datasets and models, tracking changes over time.
  + **Quality Indicators:** Metrics assessing data quality, such as completeness, accuracy, and consistency.
  + **Field Mappings:** Descriptions of how different data fields correspond to model inputs, facilitating easier integration and understanding.

**System or Method Used for Metadata Management:**

* + Metadata is managed using a combination of structured documentation within the Git repository and a dedicated metadata management tool integrated with the project workflow. This approach allows for systematic tracking of all metadata attributes associated with datasets and model artifacts.

**Issues Encountered:**

* + One issue encountered was inconsistent documentation practices, leading to gaps in metadata records that made it difficult to track data lineage and assess quality over time. **Solutions Implemented:**
  + To address this, a standardized metadata template was developed, outlining required fields and descriptions for each dataset. Regular audits of metadata completeness and accuracy were instituted to ensure adherence to documentation standards. Additionally, training sessions were held for team members to emphasize the importance of consistent metadata entry.

1. **Versioning:** DermAI employs **Git** as the primary version control system for managing code, configuration files, and related documentation. The versioning strategy involves creating branches for major features and experiments, allowing parallel development without affecting the main codebase. Each significant update, including model training and data processing scripts, is committed with detailed messages that describe the changes made. Git tracks changes through a system of commits, where each commit represents a snapshot of the project at a specific point in time. Commit messages include descriptions of the updates, ensuring clarity on what has changed. Tags are used to mark specific releases or milestones, enabling easy navigation through the project’s history. Different versions of scripts and configurations are maintained to ensure transparency and reproducibility. The version history allows team members and stakeholders to review past changes, making it easier to trace the evolution of the code and understand the rationale behind specific updates. This approach is essential for validating model performance and supporting collaboration among team members.

1. **Data Preprocessing, Augmentation, and Synthesis:**

**Preprocessing Techniques:**

* + **Normalization:**
    - * **Purpose:** Standardizes pixel values to a consistent scale, improving model convergence during training.
      * **Application:** Applied to pixel values in skin lesion images to facilitate uniform input to the model.
      * **Challenges:** Handling outliers in the dataset and varying ranges of pixel values.
      * **Solutions:** Implementing Min-Max scaling to constrain values between [0, 1] or Z-score normalization to standardize around a mean of zero with a standard deviation of one.
  + **Resizing:**

o **Purpose:** Adjusts image dimensions to meet model input requirements, ensuring computational efficiency and uniformity. o **Application:** All images are resized to a standard dimension, such as 224x224 pixels. o **Challenges:** Potential quality degradation and loss of critical information during resizing. o **Solutions:** Utilizing high-quality interpolation methods (e.g., bicubic interpolation) to minimize quality loss during the resizing process.

* + **Scaling:**
    - * **Purpose:** Brings continuous feature values within a consistent range to improve model performance.
      * **Application:** Adjusts any additional numerical features associated with lesions, like size measurements.
      * **Challenges:** Features with widely varying ranges may dominate model training. o **Solutions:** Normalizing feature values using standard scaling techniques to ensure equitable treatment across features.
  + **Feature Selection:**
    - * **Purpose:** Identifies and retains the most relevant features to enhance model efficiency and accuracy.
      * **Application:** Uses statistical tests and domain knowledge to filter out irrelevant features from the dataset. o **Challenges:** Incorrect feature selection can negatively impact model performance. o **Solutions:** Implementing cross-validation techniques to verify the predictive power of selected features and refine model input.

**Data Augmentation and Synthesis:**

* + **Image Transformations:**
    - * **Techniques:** Includes rotations, flips, brightness adjustments, and zooms.
      * **Purpose:** Enhances image diversity to improve model robustness against variations in skin lesion presentations.
      * **Challenges:** Over-augmentation can create distorted or unrealistic images.
      * **Solutions:** Setting reasonable limits on transformation parameters to ensure that augmented images retain the characteristics of real lesions.
  + **Text Data Augmentation:**
    - * **Techniques:** Utilizes synonym replacement, random deletion, and backtranslation on lesion descriptions. o **Purpose:** Enriches textual data and enhances generalization by varying the phrasing of descriptions.
      * **Challenges:** Careful application is necessary to avoid altering the intended meaning of descriptions. o **Solutions:** Using controlled vocabularies and perform coherence checks on augmented text to maintain clarity.
  + **Synthetic Data Generation:**
    - * **Methods:** Utilizes techniques like GANs (Generative Adversarial Networks) for generating synthetic images of skin lesions. o **Purpose:** Addresses class imbalance by creating additional samples for underrepresented lesion types.
      * **Challenges:** Training GANs can be resource-intensive and may introduce biases if not managed correctly. o **Solutions:** Regularly evaluating synthetic images for quality and diversity, adjusting training processes to prevent overfitting or mode collapse.

1. **Report on Risk Management in Data Collection:**

**Identified Risks and Mitigation Strategies:**

**1. Privacy Breaches:**

* + - **Risk:** Handling sensitive medical data increases the likelihood of privacy breaches, which could expose user information.
    - **Mitigation Strategy:** Data Encryption: All sensitive data, including images and personal information, is encrypted both in transit and at rest using robust encryption standards.

Access Controls: Implemented strict access controls with secure authentication mechanisms (e.g., two-factor authentication) to limit data access to authorized personnel only.

* + - **Effectiveness:** These measures significantly reduce the risk of unauthorized access. Regular security audits will be conducted to verify compliance with HIPAA standards, ensuring ongoing protection of sensitive data.

**2. Data Corruption:**

* + - * **Risk:** Corruption of data during collection, processing, or storage could compromise the quality of the training datasets.
      * **Mitigation Strategy:** Regular Backups: Data is regularly backed up to prevent loss and facilitate recovery in case of corruption.

Data Validation: Implementation of automated data validation checks to identify and rectify inconsistencies or corrupt entries in datasets.

* + - * **Effectiveness:** The combination of backups and validation checks will effectively minimize instances of data corruption, with regular reviews ensuring data integrity.

**3. Class Imbalance:**

* + - * **Risk:** An imbalance in the dataset could lead to biased model predictions, affecting overall system accuracy.
      * **Mitigation Strategy:** Data Augmentation: Utilizing techniques such as image augmentation to synthesize additional data for underrepresented classes, balancing the dataset.

Resampling Techniques: Employing resampling strategies during training to address class imbalances dynamically.

* + - * **Effectiveness:** These strategies have positive outcomes in model training, improving sensitivity and specificity across different skin lesion classes.

**4. Data Quality Issues:**

* + - * **Risk:** The presence of low-quality or mislabeled data can lead to inaccurate model predictions.
      * **Mitigation Strategy**: Quality Assurance Protocols: Establishing rigorous quality assurance protocols during data collection, including expert review and annotation of training images.

User Feedback Loop: Implementation of a user feedback mechanism to identify and rectify potential inaccuracies in the dataset.

* + - * **Effectiveness:** Continuous monitoring and feedback will improve data quality and adjustments made based on user inputs and experts.

**5. Compliance Failures:**

* + - **Risk:** Non-compliance with relevant data privacy laws (e.g., GDPR, CCPA) could lead to legal penalties and damage to reputation.
    - **Mitigation Strategy:** Regular Compliance Audits: Conducting periodic audits to ensure adherence to GDPR and HIPAA requirements, focusing on data handling practices and user consent protocols.

Training Programs: Providing training for all team members on data privacy regulations and ethical considerations in AI.

* + - **Effectiveness:** The compliance strategy will be effective, with no reported incidents of non-compliance. Regular training and audits ensure that the team remains informed and proactive.

**Reflection on Effectiveness and Improvements:**

Overall, the implementation of these strategies will prove effective in managing data-related risks throughout the AI lifecycle of DermAI. Continuous monitoring and adjustments based on user feedback have enhanced the system's resilience to potential data management issues.

Improvements:

* + - **Enhanced Anomaly Detection:** Implementing more sophisticated anomaly detection techniques could further strengthen the system's ability to identify and mitigate risks related to data corruption and quality.
    - **Stronger User Engagement**: Increasing user engagement through regular feedback sessions and educational resources could help in early identification of data quality issues, promoting a collaborative approach to data integrity.
    - **Scalability of Compliance Audits:** As the user base grows, developing automated compliance monitoring tools could streamline audits and ensure that the system adapts to evolving regulatory requirements efficiently.

**Residual Risk Assessment**

Despite implementing initial mitigation strategies across various stages of the model development lifecycle, certain residual risks remain, requiring further evaluation and management. We used a **Likelihood vs. Impact Matrix** to systematically assess these remaining risks, prioritizing those with **unacceptable** and **intolerable** levels that demand additional mitigation efforts.

**Likelihood vs. Impact Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Residual Risk** | **Likelihood** | **Impact** | **Risk Level** | **Additional Mitigation Strategies** |
| **Model Bias** | High | High | Critical | Regular fairness audits, use of Fairlearn/AIF360; ongoing evaluation of demographic diversity in data. |
| **Data Privacy & Security** | Moderate | High | High | Strengthened encryption protocols, enhanced access control policies, and biannual security audits to align with HIPAA/CCPA standards. |
| **Class**  **Imbalance** | Moderate | High | High | Continued application of synthetic data techniques (e.g., GANs); adjust class weights dynamically during training. |
| **Residual Risk** | **Likelihood** | **Impact** | **Risk Level** | **Additional Mitigation Strategies** |
| **Irregular Learning** | Moderate | Moderate | Moderate | Cross-validation with emphasis on minority classes, early stopping to prevent overfitting. |
| **Explainability Challenges** | Moderate | Moderate | Moderate | Expand use of  SHAP/LIME for  interpretability;  communicate model limitations clearly to users. |
| **Data Drift** | Moderate | Moderate | Moderate | Implement automated drift detection with alerts; periodic model retraining based on user feedback and new data patterns. |
| **Scalability & Latency** | Low | High | Moderate | Cloud-based autoscaling and geographic distribution of servers to ensure consistent response times. |

**Additional Mitigation Strategies for High and Critical Residual Risks**

* + - **Model Bias**: Given the criticality of this risk, periodic fairness audits will be conducted using tools like Fairlearn and AIF360 to actively identify and mitigate any demographic biases in the model. Data diversity will be continually assessed, with augmented sampling strategies used to balance underrepresented groups.
    - **Data Privacy & Security**: Data privacy remains a high-priority risk, and additional layers of security will be implemented to safeguard sensitive medical information. This includes advanced encryption techniques, biannual compliance audits, and strict access controls aligned with regulatory requirements (HIPAA, CCPA).
    - **Class Imbalance**: With class imbalance posing a high residual risk, further data augmentation techniques, such as GANs for synthetic image generation, will be utilized. Additionally, dynamic class weighting will be applied during training to reduce dependency on the dominant class.

1. Report on Trustworthiness in Data Collection:

- To enhance trustworthiness, we will engage stakeholders, including users and dermatology experts, in the design and implementation process. Regular feedback mechanisms (surveys, focus groups) will be established to gather user insights and concerns. Transparency about data usage, security measures, and AI model decision-making will be prioritized to build user confidence in the DermAI platform. The ethical guidelines established will be revisited regularly to ensure alignment with user needs and ethical standards.

**Identified Trustworthiness Strategies:**

1. **Transparency in Data Usage**:
   * **Strategy**: Maintained clear documentation on how data is collected, processed, and utilized, providing stakeholders with insights into the data lifecycle. o **Effectiveness**: This transparency foster trust among users and stakeholders, as they were able to understand data handling practices. Feedback indicated a high level of user confidence due to clear communication about data usage and privacy policies.
2. **User Consent Mechanisms**:
   * **Strategy**: Implemented robust consent protocols, ensuring that users were informed and gave explicit permission before their data was collected or used.
   * **Effectiveness**: The consent process is user-friendly, and periodic reviews might confirm that users appreciate having control over their data. This proactive approach to consent contributed to a trusting environment, reducing the risk of non-compliance.
3. **Data Anonymization**:
   * **Strategy**: Employed data anonymization techniques to protect personal identifiers in datasets, ensuring that data could not be traced back to individual users.
   * **Effectiveness**: Anonymization enhanced user trust, as participants feel more secure sharing sensitive information. Regular audits showed that the anonymization methods effectively protected user identities while still allowing for meaningful data analysis.
4. **Regular Stakeholder Communication**:
   * **Strategy**: Established a communication framework for regular updates to stakeholders about data management practices, compliance status, and system improvements. o **Effectiveness**: Consistent communication lead to increase stakeholder engagement and trust. Feedback indicated that stakeholders value being kept informed and involved in the data management process.
5. **Continuous Training and Education**:
   * **Strategy**: Provided ongoing training for team members on ethical data practices, compliance regulations, and the importance of maintaining trustworthiness in data management. o **Effectiveness**: Training sessions improve team awareness and responsiveness to data integrity issues. Evaluations indicate a higher level of confidence among team members in handling data responsibly.
6. **Third-Party Audits**:
   * **Strategy**: Engaged independent auditors to assess data management practices and compliance with relevant legal frameworks.
   * **Effectiveness**: External audits validate the effectiveness of internal practices and provide an objective view of the system's trustworthiness. Recommendations from auditors will be implemented promptly, enhancing overall data management.

**Reflection on Effectiveness and Improvements:**

The trustworthiness strategies employed in the DermAI project have been largely effective in building confidence among users and stakeholders. The combination of transparency, user control, and regular communication has established a solid foundation for trust in data management practices.

**Improvements**:

* **Enhanced Feedback Mechanisms**: Implementing more robust channels for user feedback could further refine trustworthiness strategies. Encouraging users to share their experiences and concerns would promote a culture of open communication.
* **Proactive Risk Assessment**: Developing a proactive risk assessment framework for trust-related issues would enable the team to identify potential vulnerabilities before they impact trust levels.
* **Expanded User Education**: Offering comprehensive educational resources on data privacy and management practices could empower users, making them more informed participants in the data lifecycle.

**Model Evaluation and Development**

1. **Model Development**

**Algorithm Selection**

We used three algorithms: Random Forest, Logistic Regression, and Decision Tree, testing each extensively with varied configurations to determine the most effective model.

* + **Linear Regression**: This algorithm was evaluated with a detailed grid of parameters using GridSearchCV. The grid included variations in: o **Penalty Types:** Both L1 and L2 penalties were tested to examine how regularization affected model complexity and performance.
    - **Solvers:** Four solvers (liblinear, saga, newton-cg, lbfgs) were considered, as they provide different optimization methods that may impact convergence speed and solution accuracy, especially with regularized terms.
    - **Max Iterations (max\_iter):** The maximum number of iterations for solver convergence was adjusted between 100, 200, and 300 to ensure that the optimization processes for L1 and L2 penalties were stable and that the model didn’t prematurely converge.
    - **Inverse of Regularization Strength (C):** We tested selective & discrete values (0.001 - 100). This parameter helps control the strength of regularization thus, avoiding overfitting.
  + **Decision Tree:** To explore tree-based models, a comprehensive parameter grid was employed for Decision Trees:
    - **Criterion:** We compared both gini and entropy criteria, each of which affects the tree-splitting strategy.
    - **Max Depth:** Multiple depths (e.g., None, 10, 20, 30, 40) were tested to control for complexity. By limiting depth, we reduce the risk of overfitting while exploring the model’s capacity to capture data patterns.
    - **Min Samples Split (min\_samples\_split):** The grid tested values from 2 to 20 to see the effect of requiring more samples for a split. Higher values can lead to simpler trees that are less likely to overfit.
    - **Min Samples Leaf (min\_samples\_leaf):** Values between 1 and 8 were tested to find an optimal leaf size, further controlling complexity by reducing the likelihood of creating small, unreliable nodes.
  + **Random Forest:** Following tuning and comparison, Random Forest emerged as the preferred model due to its performance improvements in F1-score and cross-validation. Key configurations included:
    - **Number of Estimators (n\_estimators):** The forest was constructed using 200 trees, which allows the model to average predictions over many trees, increasing robustness.
    - **Max Features (max\_features):** The sqrt option was chosen, meaning that each tree considers a subset of features, helping to increase variance reduction across trees.
    - **Min Samples Split and Leaf:** These were set at 10 and 4 respectively, based on the grid search findings, achieving a balanced model that avoided overfitting while capturing patterns effectively.

This rigorous tuning process validated the Random Forest model’s robustness and reliability, solidifying it as the optimal choice for this project.

**Feature Engineering and Selection:**

The feature engineering and selection process involved several code-driven steps focused on improving data quality and ensuring model readiness.

* + **Data Cleaning:** Columns with over 50% missing values were removed to maintain a strong signal-to-noise ratio, preventing the model from being influenced by incomplete data. After this step, rows with any remaining missing values were also discarded, ensuring a complete dataset.
  + **Dimensionality Reduction with Variance Inflation Factor (VIF):** To address multicollinearity, VIF analysis was conducted on all numerical features. Features with high VIF scores, such as tbp\_lv\_perimeterMM and tbp\_lv\_minorAxisMM, were removed, as multicollinear features can lead to model instability and inflated variances in estimates. Removing these features improves interpretability and stability.
  + **Feature Encoding and Scaling:** After VIF-driven selection, categorical features were one-hot encoded to create binary indicators for each category. All numerical features were then standardized using StandardScaler, a critical step that ensures each feature contributes proportionally to the model’s performance and avoids issues arising from different scales.

**Model Complexity & Architecture:** Given the medical application of the system, model complexity was not our top priority. Instead, we focused on extensive feature selection, which effectively reduced model complexity while retaining interpretability. Additionally, we opted not to apply PCA or other dimensionality reduction techniques, as it is crucial for the features to remain meaningful in a medical context. PCA, for instance, would introduce linear combinations of features that replace the original variables—an approach that would compromise the interpretability and clinical relevance we require for this application.

However, we did incorporate principles such as Occam’s Razor throughout the system to favor simplicity where possible. For example, in our final model, which takes the probability outputs as input and provides the overall prediction, we selected logistic regression over a decision tree. Although the decision tree offered minimal improvement in predictive accuracy, the logistic regression model is simpler and thus preferable in this context, aligning with our priority for a more interpretable and streamlined solution

1. **Model Training**  **Training Process**

The training process is designed to address data imbalance and optimize model performance through targeted data augmentation and tuning.

* + **Data Augmentation via ADASYN:** Class imbalance was a notable issue, with a significantly smaller representation of the minority class (target=1). We applied ADASYN, an advanced oversampling technique that generates synthetic samples for the minority class, focusing on regions where class boundaries are less defined. This helps the model better recognize patterns in both classes and enhances sensitivity.

* + **Detailed Tuning Strategy:** Each of these parameters has its own strengths and weaknesses, and our goal was to identify the optimal combination to achieve the best overall performance. To achieve this, we implemented Grid Search Cross-Validation with 10 folds to systematically evaluate and select the best combination of parameters. This approach allowed us to explore a comprehensive range of parameter values while ensuring robust performance across different subsets of the data.
    - **Logistic Regression:** Using GridSearchCV, we tuned parameters including different solvers (liblinear, saga, newton-cg, lbfgs) and penalties (l1, l2). The parameter C was also optimized, representing inverse regularization strength, which helps to prevent overfitting by penalizing complex models.
    - **Random Forest:** We fine-tuned parameters such as max\_depth, min\_samples\_split, and min\_samples\_leaf to carefully balance complexity and generalization (again using Grid Search). By adjusting these values, the Random Forest Classifier could capture complex patterns without becoming overly sensitive to noise, ensuring consistent performance on both training and validation data.

1. **Model Evaluation**

**Performance Metrics**

* + **Detailed Metrics Calculation:** In addition to standard F1 score, the code calculates sensitivity (recall for the positive class) and specificity (true negative rate). These metrics are particularly useful for understanding the model’s effectiveness in accurately classifying each class, especially when data is imbalanced.
  + **Threshold Optimization:** By calculating the precision, recall and F1-score across various thresholds, the code identifies an optimal threshold for classifying probabilities.

This approach maximizes predictive performance by selecting a threshold that balances precision and recall, aligning the model’s output with the desired classification outcomes.

* + **ROC and AUC Analysis:** ROC curves are plotted to visualize the model’s performance across different probability thresholds, and AUC (area under the curve) is calculated as a summary metric. A higher AUC indicates that the model performs well across thresholds, providing an aggregate measure of model robustness and accuracy over various cutoff points.

**Cross-Validation**

* + **Cross-Validation Strategy:** A stratified K-fold cross-validation approach is implemented to validate the model’s stability. By using stratification, each fold preserves the proportion of classes, which is crucial for imbalanced datasets. This method allows consistent performance evaluation and helps ensure that the model generalizes well across diverse subsets of data.

1. **Implementing Trustworthiness and Risk Management in Model Development**
   * **Risk Management Report:** In our model development, we identified and addressed key risks such as privacy breaches, data corruption, class imbalance, data quality issues, and compliance failures. For privacy risks, we implemented data encryption and access controls during model training and storage, which have effectively minimized unauthorized access risks. Regular security audits are part of our approach, ensuring compliance with standards like HIPAA. Data corruption was mitigated through regular data backups and automated validation checks, both of which have helped maintain data integrity across model development stages.
   * To handle class imbalance, we applied ADASYN, generating synthetic samples for the minority class to enhance training quality. This strategy improved the model's sensitivity and specificity, especially in underrepresented classes. Data quality risks were managed with expert-reviewed data annotations and strict quality assurance protocols, which have significantly reduced errors due to low-quality or mislabeled data. User feedback mechanisms were also integrated, allowing continuous quality improvements based on real user input. For compliance risks, regular audits and team training on GDPR and HIPAA guidelines ensure that data privacy regulations are followed, with consent protocols enhancing user data protection. As the project scales, we may implement automated compliance monitoring tools to maintain this standard.
   * **Trustworthiness Report:**
   * We prioritized several trustworthiness considerations in our model development, including transparency, user consent, data anonymization, stakeholder communication, continuous team training, and third-party audits. Transparency in data usage was achieved through clear documentation on how data is processed and used in the model, with additional insights provided through feature importance and confidence scores.

Feedback has shown that this transparency strengthens user trust, as stakeholders understand and support our data handling practices.

* + User consent was maintained with robust protocols, ensuring participants were informed and gave explicit permission before data was used. This approach has been effective in fostering a trusting environment and enhancing user control over their data. Data anonymization techniques were employed to protect individual identities while still allowing meaningful model analysis, with audits verifying the effectiveness of these measures. Regular updates to stakeholders on data management, compliance, and system improvements have helped maintain their trust and engagement. Continuous training for team members on ethical data practices has strengthened our internal capacity to handle data responsibly, and third-party audits have validated our practices, providing an objective view of our trustworthiness.
  + To further enhance these efforts, we are considering improvements such as more robust feedback channels to capture user insights, proactive risk assessment frameworks for trust-related issues, and expanded educational resources to empower users with a better understanding of our data management practices. These steps will support a collaborative and transparent model development environment as we continue to prioritize trustworthiness and compliance throughout the AI lifecycle.

1. **Application of HCI Principles in AI Model Development**

**Developing Interactive Prototypes**

To facilitate user interaction with the AI model, we will implement interactive prototypes using **Gradio** or **Streamlit**. These libraries enable the creation of user-friendly web applications, allowing end-users to interact directly with the model. Specifically, we will incorporate:

* + **Sliders** for adjusting input values of numerical features, enabling users to experiment with the model’s response to varying inputs.
  + **Text Inputs** and **Dropdowns** for categorical features to provide flexibility and realism in testing different scenarios.
  + **Real-Time Feedback** mechanisms will be established, allowing users to see predictions update immediately as inputs change. This real-time interaction will help users understand the model’s behavior intuitively.

**Designing Transparent Interfaces**

For transparency in the AI model’s decision-making, we will use **matplotlib**, **plotly**, or **seaborn** to create visual explanations. Key visual components will include:

* + **Feature Importance Charts** to display which features most significantly influence predictions, helping users to see the relative impact of different inputs.
  + **Confidence Scores** to convey the model’s certainty for each prediction, displayed as bar charts or gauges for clarity.
  + **Threshold Visualization** tools, such as precision-recall and ROC curves, to illustrate the model’s sensitivity-specificity trade-offs, helping users to understand how the optimal threshold affects prediction performance.

**Creating Feedback Mechanisms:** To capture user insights and improve model interaction, we will integrate feedback mechanisms within the interface. These will include:

* + **Thumbs-Up/Down Options** to allow users to quickly indicate if predictions meet their expectations.
  + **Text Comment Fields** where users can provide specific feedback on model outputs, adding qualitative data to enhance model understanding.

**Feedback Utilization** will be structured through a logging system that tracks user interactions. This data will inform both interface adjustments and model refinements. User feedback will be used to retrain the model on flagged instances, establishing a continuous improvement loop.

**Deployment and Testing Management Plan:**

**1. Deployment Environment Selection**

**Environment Chosen: Local Deployment Using Docker Containers**

**Justification:**

* **Project Scale and Requirements**: This project requires running a machine learning-based Gradio application locally. Using Docker ensures portability across various local machines, which is crucial during development and testing phases.
* **Resource Optimization**: Running the application locally allows for resource monitoring and optimization without relying on costly cloud services during the testing phase.
* **Operational Goals**: The primary goal of this deployment is to test the application end-to-end while integrating tools like Prometheus and Grafana for monitoring and feedback collection. A local environment simplifies debugging and iteration cycles.
* **Future Scalability**: Docker's containerization provides the flexibility to scale to a cloud-based Kubernetes cluster or edge devices if required in the future.

**2. Deployment Strategy**

**Strategy Used: Containerization with Docker and Monitoring Setup**

**Key Tools and Frameworks**:

* **Docker**: Used for containerization of the Gradio application, ensuring that dependencies and configurations are encapsulated in a single portable environment.
* **Prometheus**: Instrumentation and monitoring tool to collect metrics, such as latency and success rates, from the Gradio application.
* **Grafana**: Visualization and alerting platform to track and analyze real-time metrics.
* **Docker-Compose**: Orchestrates multiple services like the Gradio app, Prometheus, and Grafana.

**Steps**:

1. **Containerization**:
   * Created a Dockerfile for the Gradio application that includes Python dependencies (pandas, numpy, torch, etc.) and Prometheus metrics setup.
   * Added feedback functionality and latency monitoring using Prometheus in the application code (ui\_notebook.py).
   * Metrics are exposed on port 8000 using the prometheus\_client library.
2. **Service Orchestration**:
   * A docker-compose.yml file was configured to manage:
     + The Gradio application.
     + Prometheus, which scrapes metrics at a regular interval (configured in prometheus.yml).
     + Grafana, which visualizes these metrics.

**Benefits**:

* **Scalability**: The containerized application can easily be scaled across multiple machines or migrated to a cloud-based environment like AWS or GCP.
* **Reliability**: Docker ensures consistency across environments, reducing issues related to dependency mismatches.
* **Observability**: Integration with Prometheus and Grafana provides real-time insights into application performance and potential issues.

**3. Security and Compliance in Deployment (Trustworthiness and Risk Management)**

**Data Protection and Secure Access**:

* **Data Privacy**: Sensitive data such as user feedback is stored locally in the /app/feedback directory. Access is restricted to authorized personnel by limiting container volumes.
* **Metrics Protection**: Prometheus metrics are exposed only on port 8000, and access is limited to internal services defined in docker-compose.yml.

**Compliance Frameworks**:

* **General Data Protection Compliance (GDPR)**: While this is a test environment, the feedback mechanism ensures user consent is obtained for any data collection (through explicit form submission).
* **Network Isolation**: Docker’s network configuration ensures that internal communication between services (Gradio app, Prometheus, Grafana) is secure and isolated from external access.

**Monitoring and Alerting**:

* Alerts are configured in Grafana based on Prometheus metrics:
  + High latency alerts: Triggered when http\_request\_latency\_seconds exceeds a threshold for consecutive requests.
  + Service downtime: Alerts when the up metric (availability) for the Gradio app is 0.

**Risk Management**:

* **Container Isolation**: Each service (Gradio app, Prometheus, Grafana) runs in its container, preventing cross-service interference.
* **Dependency Pinning**: All dependencies, such as Python packages and Prometheus versions, are explicitly defined to avoid compatibility issues during deployment.

**Evaluation, Monitoring, and Maintenance Plan**:

**1. System Evaluation and Monitoring**

**Monitoring Tools and Metrics Tracked**

* **Tools Used**:
  + **Prometheus**: For collecting and storing real-time metrics from the deployed system.
  + **Grafana**: For visualizing the metrics and setting up alerts.
* **Metrics Tracked**:
  + **Latency**: Measures the response time of the system, helping to ensure the AI model processes requests efficiently.
    - Tracked using request\_processing\_time\_seconds histogram metric added in the ui\_notebook.py code.
  + **Error Rate**: Tracks the number of failed requests, providing insights into system reliability.
    - Monitored using the error\_count counter metric.
  + **System Uptime**: Using the up metric from Prometheus, indicating whether the system is running.

**Drift Detection Methods:**

* **Methodology**:
  + Model performance metrics, such as prediction accuracy, are periodically compared against a validation dataset to detect shifts in data patterns.
  + User feedback is analyzed for recurring errors or reduced accuracy in predictions.
  + Continuous monitoring of input feature distributions via Prometheus/Grafana to detect deviations from training data.
* **Outcomes**:
  + No significant drift detected in initial testing phase. Alerts for latency and error rate have been configured to ensure early detection of potential issues.

**2. Feedback Collection and Continuous Improvement**

**Feedback Mechanisms Implemented**

* **Feedback Integration**:
  + A dedicated feedback form was added to the ui\_notebook.py using Gradio.
  + Feedback form questions:
    - "Was the system user-friendly?" (Yes/No)
    - "How would you rate the system?" (Good/Better/Best)
    - Additional comments section for open-ended user feedback.
  + Feedback data is stored in a CSV file (feedback.csv) inside the /app/feedback directory for further analysis.

**Continuous Improvement:**

* Feedback collected is periodically reviewed to identify areas for improvement in user experience, model accuracy, and overall system reliability.
* Insights gained are used to retrain the model or enhance the system’s features.
* Example:
  + If multiple users report dissatisfaction with response time, the system latency is optimized by improving model inference pipelines or upgrading hardware.

**3. Maintenance and Compliance Audits**

**Trustworthiness and Risk Management Strategies**

* **Trustworthiness**:
  + Regular evaluation of model predictions for bias or incorrect classifications using test datasets.
  + Logs of predictions and user interactions are reviewed to ensure transparency and accountability.
* **Risk Management**:
  + Regular updates to system dependencies and libraries to address security vulnerabilities.
  + Periodic reviews of access controls to ensure only authorized users can modify or interact with the system.
* **Compliance**:
  + Ensured compliance with data privacy regulations (e.g., GDPR):
    - User feedback data stored securely in the /app/feedback directory.
    - Model inputs (e.g., images, feature data) are not stored persistently unless anonymized for monitoring purposes.
  + Audit logs are maintained to track system changes and access history.

**References:**

1. (National Library of Medicine - Increasing access to specialized dermatology care)

[https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7858729/]

1. (National Library of Medicine - Limited Access to Dermatology & Teledermatology)

[https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9946088/]