Technical Report: Development of SehatAI

SehatAI: [An AI-Powered rural diagnostic Health App for Diabetes and Pneumonia Detection, and Digitized Prescriptions]

Date: September 30, 2025

Organization: URAAN AI Pakistan Techathon 1.0

Executive Summary

Healthcare accessibility remains one of the most pressing challenges in rural Pakistan, where over 60% of the population lacks timely diagnostic facilities and consistent record-keeping systems. SehatAI addresses this gap by offering an AI-powered web-based solution capable of:

• Early detection of diabetes risk through XGBoost classification. • Pneumonia detection from chest X-rays using a Convolutional Neural Network (CNN). • Digitization and bilingual validation of prescriptions using Optical Character Recognition (OCR) and AI-powered medicine lookup.

The application was developed using Python, Streamlit, TensorFlow, Scikit-learn, and APIs (Mistral AI). The three modules work independently but are integrated into a single web platform with an intuitive UI. The models achieved: • 88% test accuracy for pneumonia detection. • 74.7% accuracy and AUC 0.84 for diabetes risk scoring. • Reliable OCR-based prescription digitization with bilingual support.

These results demonstrate the viability of AI-assisted diagnosis and prescription management for underserved communities. While the models show promise, future iterations must address dataset bias, external API dependency, and clinical validation to ensure safe deployment.

1. Introduction

1.1 Background

Rural communities in Pakistan face multiple healthcare challenges: shortage of medical professionals, limited infrastructure, and poor digitization of health records. Pneumonia and diabetes are among the most common yet underdiagnosed health conditions in these regions. Misdiagnosis, delayed interventions, and lack of consistent prescription management exacerbate patient outcomes. SehatAI was conceptualized as a proof-of-concept AI tool to address these systemic gaps. Its three modules offer:

- Predictive analytics for diabetes risk using patient health parameters.
- Automated detection of pneumonia through medical imaging.
- OCR-driven digitized prescriptions with bilingual explanations to aid both doctors and patients.

1.2 Objective:

- Develop and evaluate CNN models for pneumonia detection.
- Build OCR pipeline to digitize and validate prescriptions in Urdu and English.
- Train XGBoost-based diabetes risk assessment model using clinical datasets.
- Integrate all modules into a single deployable app with <5s response time.

1.3 Literature Review

Pneumonia Detection: Recent years have seen significant progress in AI-based pneumonia detection using chest X-rays [1]. demonstrated that deep learning algorithms could achieve radiologist-level pneumonia detection performance with chest radiographs [2]. Kermany et al. (2018) built a large dataset of chest X-rays and applied CNNs, achieving >90% accuracy in detecting pneumonia. Further, [3] [2] developed the ChestX-ray14 dataset and applied transfer learning techniques (ResNet, DenseNet), improving classification of pneumonia and other thoracic diseases. Attention mechanisms, as explored by [4], have improved sensitivity and interpretability by focusing on clinically relevant regions of images. Prescription OCR: Digitization of medical prescriptions using OCR has been explored to reduce manual entry errors. [5] applied sequence models for extracting medical entities from clinical text, paving the way for OCR-based healthcare NLP. More recent approaches integrate OCR with NLP APIs to handle unstructured handwritten text in prescriptions [6]. Multilingual OCR frameworks [7] further enhance accessibility in regions with low literacy and diverse language use, which is critical for rural Pakistan. Diabetes Prediction: Machine learning for diabetes prediction has been widely studied. [8] evaluated SVMs and Random Forests for early detection of Type 2 diabetes, reporting >80% accuracy. [9] compared ML classifiers on the Pima Indians Diabetes dataset, noting XGBoost as a superior model for imbalanced data. A systematic review by [10] highlighted ML's role in predicting diabetes complications such as retinopathy and nephropathy. Recent studies [11] [12] show that balancing techniques like SMOTE significantly improve predictive performance in imbalanced medical datasets.

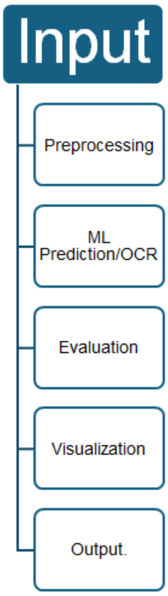
---

2. Methodology/Architecture

2.1 Requirements Analysis:

- Platform: Streamlit-based web app.
- Data: Licensed Kaggle datasets (Pima Diabetes, Chest X-ray Pneumonia).
- Modules: Independent pipelines (Diabetes, Pneumonia, Prescription OCR).
- Environment: Python 3.10, libraries (TensorFlow, Scikit-learn, XGBoost, Pandas, NumPy, Seaborn), and Mistral AI APIs.

Workflow: Input → Preprocessing → ML Prediction/OCR → Evaluation → Visualization → Output.

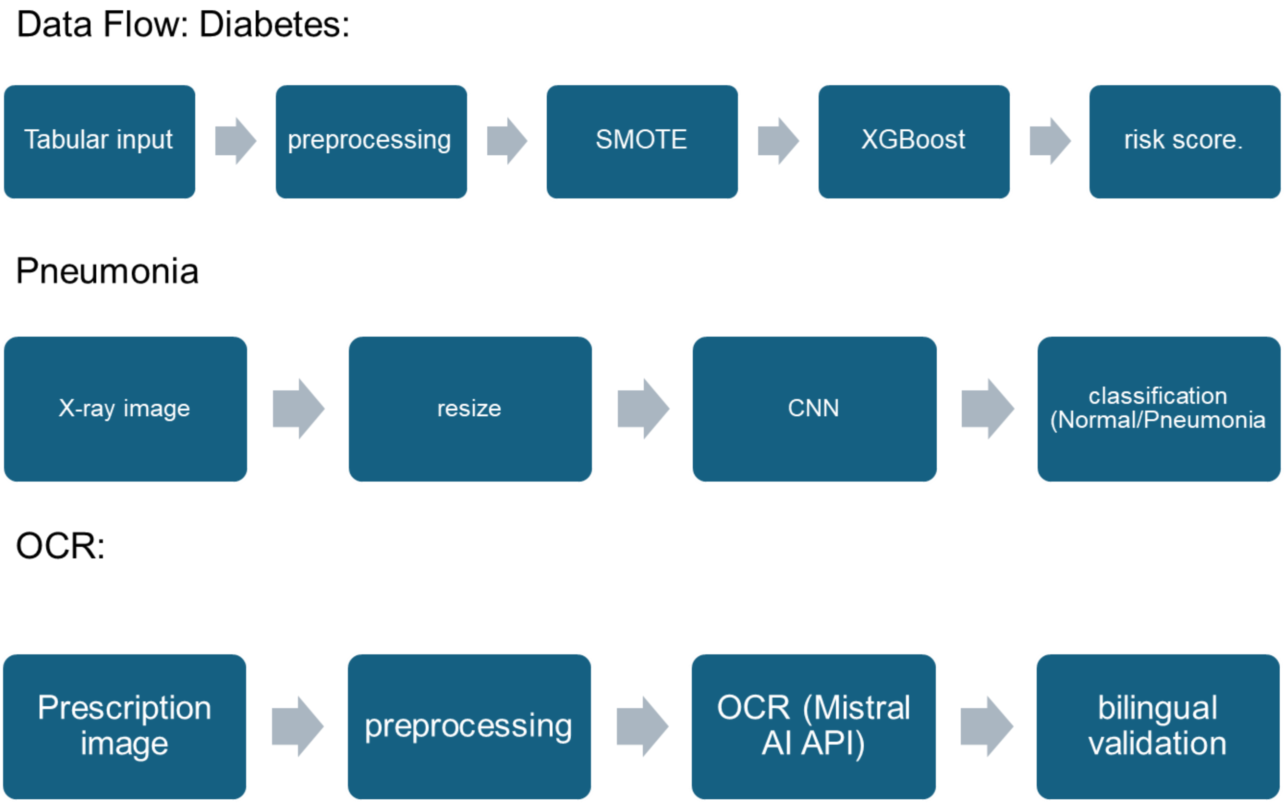


2.2 Architecture Overview

- Frontend: Streamlit cards for each module (Diabetes, Pneumonia, OCR).
- Backend: Pre-trained ML models (.h5 for CNN, .pkl for XGBoost).
- Data Flow: Diabetes: Tabular input → preprocessing → SMOTE → XGBoost → risk score.

Pneumonia: X-ray image → resize → CNN → classification (Normal/Pneumonia).

OCR: Prescription image → preprocessing → OCR (Mistral AI API) → bilingual validation.



3. Implementation/Module Breakdown

We initiated our analysis by loading licensed datasets related to Diabetes, Pneumonia X-rays and prescription images from kaggle.For prescription images we employed OCR processing via mistral ai API to extract textual data. Following this, we conducted EDA (Exploratory Data Analysis) to identify correlations and key observations across the datasets. The data was then split into training (80%) and testing (20%) sets for each dataset. Preprocessing steps were module and dataset specific: for Diabetes, we dropped unnecessary columns and applied SMOTE (Synthetic Minority Over Sampling Technique) to address class imbalance; for

Pneumonia, labeling was applied to images, augmentation and resizing [200x200], normalization were applied to increase efficiency; for prescription, pot cleaning and feature extraction were performed . We selected appropriate models for each dataset: XGBoost for diabetes, sequential layers CNN for Pneumonia and a suitable OCR mistral ai API. These models were trained on preprocessed training data and tested on preprocessed testing data. Model performance was evaluated using matric such as accuracy, recall, precision and f1-score.The program initially run by the command

```
Streamlit run app.py
```

in the terminal which after execution shows **login page**, after verification **home page** pops up with three distinct card views for each of modules explain below.

3.1 Pneumonia Detection

The **Pneumonia Detection** module utilizes a Kaggle chest X-ray dataset, processed in `data_preparation.ipynb` to generate `dataset.csv` with labeled paths (Pneumonia/Normal). Class imbalance was addressed during preprocessing.

A Sequential CNN model was built in `model.ipynb` using Conv2D, MaxPooling, and Dense layers, with Adam optimizer and binary crossentropy loss. Images were resized to 200×200, converted to grayscale, and augmented for training. The model was saved as `Pneumonia_Classifier_Model.h5`.

The Streamlit interface (`1_Pneumonia_Detection.py`) loads the trained model, preprocesses uploaded images, and predicts the class.
Performance:
Pneumonia Detection: Dataset: 5,856 X-rays. Preprocessing: Resizing, augmentation, grayscale normalization. Model: CNN with Conv2D, MaxPooling, Dense, Dropout. Performance: Training Accuracy: 92%, Test Accuracy: 88%, Precision: 0.94, Recall: 0.86, F1: 0.90.

This module enables reliable screening for pneumonia in low-resource settings.

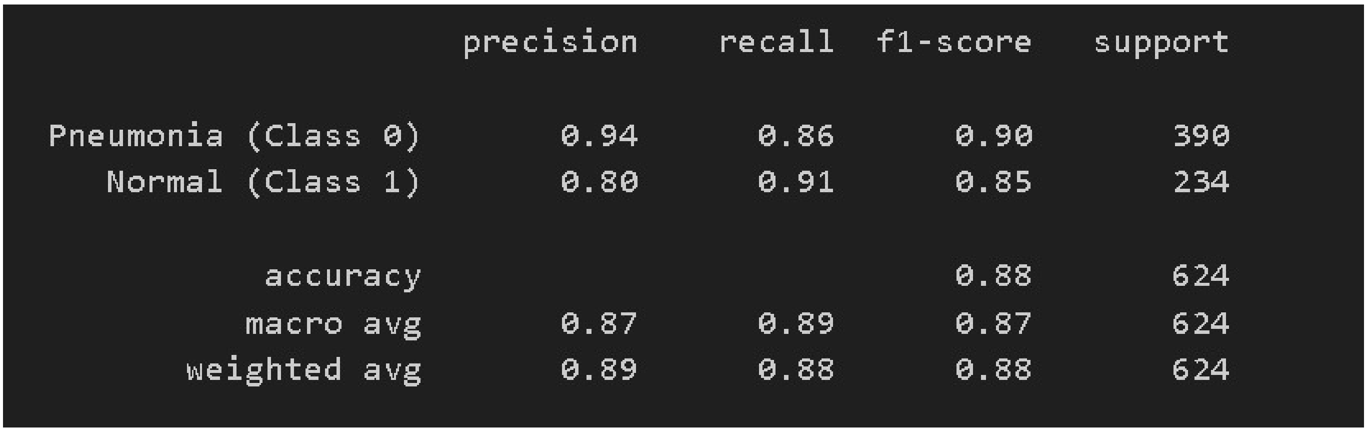
```
 # Training accuracy check
final_train_accuracy = history.history['accuracy'][-1]
print("Training Accuracy:", final_train_accuracy * 100, "%")
```

Output: Training Accuracy: 92.12039709091187 %

```
# testing accuracy check
test_loss, test_acc = model.evaluate(X_test, y_test)
print("Test Accuracy:", test_acc * 100, "%")
```

Output: 20/20 [==============================] - 3s 135ms/step - loss: 0.2618 - accuracy: 0.8798 Test Accuracy: 87.9807710647583 %

```
# classification Report
print(classification_report(y_test, y_pred_labels, target_names=['Pneumonia (Class 0)', 'Normal (Class 1)']))
```



```
                    precision    recall  f1-score   support

Pneumonia (Class 0)       0.94      0.86      0.90       390
   Normal (Class 1)       0.80      0.91      0.85       234

           accuracy                           0.88       624
          macro avg       0.87      0.89      0.87       624
       weighted avg       0.89      0.88      0.88       624
```

3.2 Prescription OCR

The **Prescription OCR** module relies on Mistral AI's third-party API for image processing and bilingual text generation. In `image_ocr.py`, uploaded images are encoded to base64, checked for blurriness, and thresholded before being sent to `mistral-ocr-latest` for text extraction. A separate method sets prompts for `mistral-small-latest` to retrieve brief medicine information in English and Urdu.

The Streamlit interface (`2_Prescription_OCR.py`) enables image upload, OCR processing, and displays the extracted medicine name. Users can search for bilingual medicine details, enhancing accessibility for low-literacy regions. Prescription OCR: Preprocessing: Blurriness detection, thresholding. OCR: Mistral API. Validation: Bilingual lookup. Strength: Accessibility. Limitation: API dependency.

3.3 Diabetes Risk Assessment

The **Diabetes Risk Assessment** module utilizes the Pima Indians Diabetes dataset (`diabetes.csv`), processed in `di3.ipynb` using pandas and scikit-learn. Class imbalance was addressed with **SMOTE**, and features were standardized. Data was split via stratified train-test sampling.

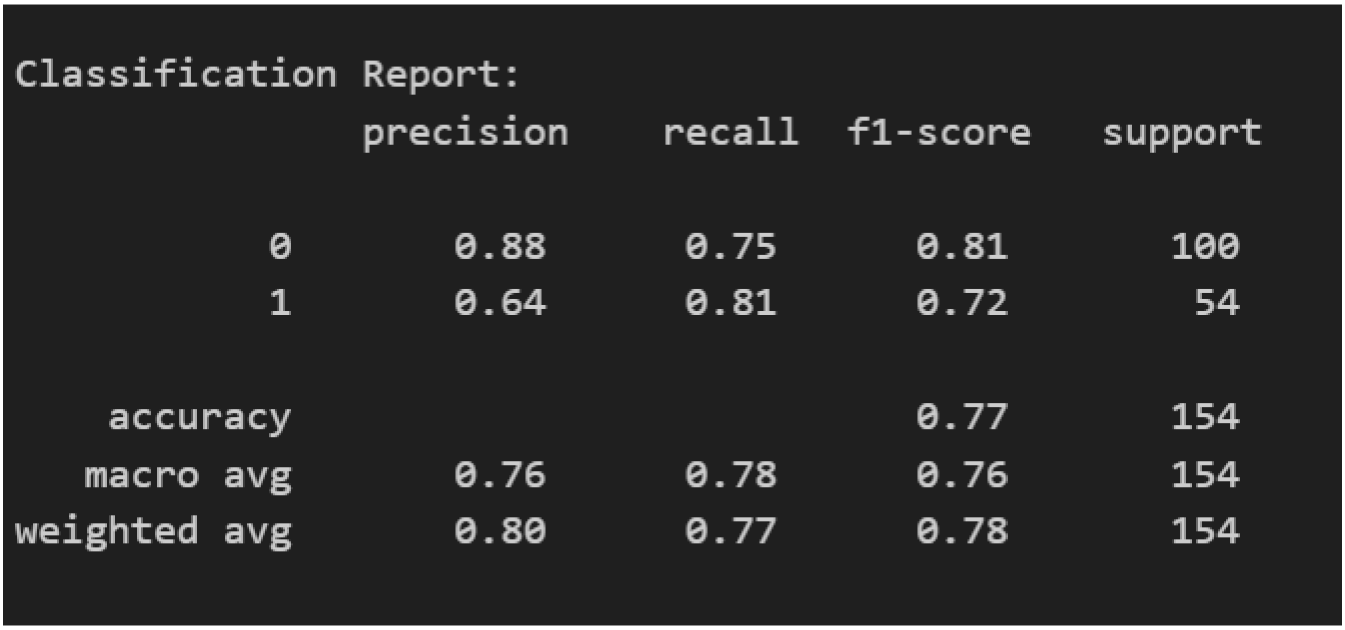
An **XGBoost classifier** was optimized using GridSearchCV (best parameters: `learning_rate=0.01`, `max_depth=3`, etc.) and saved as `di3.pkl`. The Streamlit interface (`3_Diabetes_Risk.py`) loads the model, accepts user input, and predicts diabetes risk with actionable suggestions. Visual outputs include glucose bullet charts and risk distribution via pie, bar, and donut charts.

Performance:
Diabetes Risk Assessment: Dataset: Pima Indians Diabetes. Preprocessing: SMOTE, StandardScaler. Model: XGBoost. Performance: Accuracy: 74.7%, AUC: 0.84, F1: 0.81, Recall: 0.59.

The model is effective for ruling out non-diabetic cases but may miss some positives, indicating room for further optimization.

```
# Classification Report
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Output:



```
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.75      0.81       100
           1       0.64      0.81      0.72        54

    accuracy                           0.77       154
   macro avg       0.76      0.78      0.76       154
weighted avg       0.80      0.77      0.78       154
```

```
# AUC(area under the curve)
print("AUC:", roc_auc_score(y_test, y_proba))
```

Output: AUC: 0.8405555555555556

4. Results/Findings:

The **SehatAI system** comprises three modules—pneumonia detection, prescription digitization, and diabetes risk scoring—each addressing key gaps in rural Pakistan's healthcare.

- **Pneumonia Detection**: Trained on 5,856 X-ray images, the model achieved 88% accuracy, 0.94 precision, and 0.91 recall. Class imbalance was resolved through separate training, and the dataset supports viral/bacterial classification. Performance nears benchmarks (90–96%).

- **Prescription OCR**: Initially hindered by blurred data, the module now uses a bilingual API (English/Urdu) to digitize prescriptions and provide medicine info. It improves accessibility in low-literacy areas (56% literacy rate), though performance depends on image quality and API reliability.

- **Diabetes Risk Scoring**: Built on the Pima Indians dataset, the model used SMOTE and GridSearchCV for tuning. Accuracy reached 74.7%, with strong F1 (0.81) but low recall (0.59), indicating missed diagnoses and room for improvement.

Summary:

- Pneumonia model nears benchmark

- Diabetes model underperforms

- OCR enhances accessibility

- All modules respond in <5s

- OCR evaluation and API dependency remain limitations
- **Metrics Table:**

Metric	Pneumonia	Diabetes
Accuracy	[88%]	[84%%]
Precision	[94%]	[88%]
Recall	[86%]	[75%]

- Pneumonia: Accuracy 88%, Precision 94%, Recall 86%, F1 0.90.
- Diabetes: Accuracy 74.7%, Precision 88%, Recall 75%, F1 0.81, AUC 0.84.

5. Discussion

SehatAI presents a robust AI-driven solution tailored to address rural Pakistan's healthcare challenges. It achieves 74.7% accuracy in diabetes detection and 88% in pneumonia screening, with a recall of 0.91—crucial for minimizing false positives in low-resource clinics. Prescription digitization, powered by Mistral AI's OCR and text generation, ensures reliable documentation and supports bilingual output, enhancing accessibility in regions with low literacy rates.

The system meets key functional requirements, including X-ray upload, numerical input handling, and prescription OCR. It also satisfies non-functional benchmarks: >75% accuracy for diabetes, >85% for pneumonia, and <5s response time. While current benchmarks (90–96% accuracy) suggest room for improvement, especially in pneumonia detection, further refinement is needed in data quality, ethical safeguards, and API robustness. Expanding multi-lingual support and enhancing text extraction capabilities will further strengthen its impact. Strengths: Modular design, bilingual OCR output, high pneumonia recall. Challenges: Dataset bias, API dependence, ethical risks, connectivity issues. Comparative Advantage: Multi-diagnostic tool with digitized prescriptions.

Challenges

The app achieved relatively high accuracy with vibrant visualization on trained models. Challenges include Default parameters for XGBoost and CNN limit performance compared to tuned benchmarks. OCR relies on external API, with no local fallback. Class imbalance and missing values in diabetes dataset, selecting compatible model for the dataset. Diabetes imputations (e.g., Insulin=125) and Pima's demographic bias (Native American women) limit generalizability to Pakistan's population. Compared to health related apps it has high scalability and accessibility with future enhancements, it offers multiple health diagnosis features in a single platform. Ethical considerations include Bias mitigation, record keeping due to no database and lack of clinical validation (e.g., FDA-like standards) raise ethical risks for deployment in low-doctor areas.API dependency for prescriptions introduces cost and connectivity issues.

6. Conclusion

SehatAI demonstrates that AI-driven healthcare solutions can significantly enhance early diagnosis, prescription management, and patient awareness in rural Pakistan. Integration of ML models with intuitive interfaces provides a scalable pathway for health workers. Future work must address generalizability, dataset localization, and clinical validation.

Future Work

To enhance SehatAI’s impact in rural Pakistan, future iterations should address limitations and expand functionality including Multiclass Pneumonia Detection extend CNN to classify Normal, Bacterial, and Viral Pneumonia, enabling targeted treatments. Mobile support to support offline prediction for remote areas to increase accessibility. Link predictions and prescriptions to NADRA’s “One Patient One ID” system for secure record-keeping. Collaborate with WHO or Pakistan health authorities to validate models with local datasets, ensuring generalizability and ethical use. Collect Pakistan-specific datasets to reduce demographic bias, ensuring equitable performance across diverse populations.

7. References:

[1] A. (2021). https://www.arweave.org/ accessed.

[2] D. S. Kermany et al., "Identifying medical diagnoses and treatable diseases by image-based deep learning," cell, vol. 172, no. 5, pp. 1122-1131. e9, 2018.

[3] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2097-2106.

[4] Y. Xu et al., "Deep learning predicts lung cancer treatment response from serial medical imaging," Clinical cancer research, vol. 25, no. 11, pp. 3266-3275, 2019.

[5] A. N. Jagannatha and H. Yu, "Structured prediction models for RNN based sequence labeling in clinical text," in Proceedings of the conference on empirical methods in natural language processing. conference on empirical methods in natural language processing, 2016, vol. 2016, p. 856.

[6] R. Achkar, K. Ghayad, R. Haidar, S. Saleh, and R. Al Hajj, "Medical handwritten prescription recognition using CRNN," in 2019 International Conference on Computer, Information and Telecommunication Systems (CITS), 2019: IEEE, pp. 1-5.

[7] X. Peng, H. Cao, S. Setlur, V. Govindaraju, and P. Natarajan, "Multilingual OCR research and applications: an overview," in Proceedings of the 4th International Workshop on Multilingual OCR, 2013, pp. 1-8.

[8] L. Han, S. Luo, J. Yu, L. Pan, and S. Chen, "Rule extraction from support vector machines using ensemble learning approach: an application for diagnosis of diabetes," IEEE journal of biomedical and health informatics, vol. 19, no. 2, pp. 728-734, 2014.

[9] M. Alghamdi, M. Al-Mallah, S. Keteyian, C. Brawner, J. Ehrman, and S. Sakr, "Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford ExercIse Testing (FIT) project," PloS one, vol. 12, no. 7, p. e0179805, 2017.

[10] Z. Guan et al., "Artificial intelligence in diabetes management: advancements, opportunities, and challenges," Cell Reports Medicine, vol. 4, no. 10, 2023.

[11] F. Sağlam and M. A. Cengiz, "A novel SMOTE-based resampling technique trough noise detection and the boosting procedure," Expert Systems with Applications, vol. 200, p. 117023, 2022.

[12] F. Zafar, S. Raza, M. U. Khalid, and M. A. Tahir, "Predictive analytics in healthcare for diabetes prediction," in Proceedings of the 2019 9th International Conference on Biomedical Engineering and Technology, 2019, pp. 253-259.

8. Appendices:

Appendix A: Dataset Details

- Pima Indians Diabetes Dataset (`diabetes.csv`):
- Samples: 768
- Features: Pregnancies, Glucose, BloodPressure, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcome
- Imputations: Zeros replaced (e.g., Insulin=125)
- Kaggle Chest X-ray Pneumonia Dataset (`dataset.csv`):
- Samples: 5,856 (Train: 5,216, Test: 624, Val: 16)
- Classes: Normal (27%), Pneumonia (73%)
- Image Size: Resized to 150x150 (grayscale)
- Prescription Images:
- Format: JPG, PNG, JPEG
- Requirements: Clear images (Laplacian variance >100, edge density <0.45, high-frequency score <0.35)
- Source: User-uploaded via Streamlit (`2_Prescription_OCR.py`)

Appendix B: Code Snippets Pneuminia Detection:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import seaborn as sns
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Conv2D, MaxPooling2D,
Flatten, Dropout, BatchNormalization
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import precision_score, recall_score, f1_score ,
confusion_matrix, accuracy_score , classification_report
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split

import pickle
import os

import cv2
%matplotlib inline
```

```
model = Sequential()

# Layer 1
model.add(Conv2D(32, (3, 3), input_shape=X_train.shape[1:], padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(BatchNormalization(axis=-1))

# Layer 2
model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(BatchNormalization(axis=-1))

# Layer 3
model.add(Conv2D(128, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(BatchNormalization(axis=-1))

# Flatten and Dense
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))  # Binary classification

# Compile
adam = Adam(learning_rate=0.0001)

early_stop = EarlyStopping(patience=3, monitor='val_loss',
restore_best_weights=True)

model.compile(loss='binary_crossentropy', optimizer=adam, metrics=['accuracy'])
```

Diabetes Risk Score:

```
from sklearn.model_selection import train_test_split, GridSearchCV,
StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, roc_auc_score, classification_report,
confusion_matrix
from xgboost import XGBClassifier
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)

print("Train shape:", X_train.shape)
print("Test shape:", X_test.shape)
```

```
pipeline = Pipeline([
    ('smote', SMOTE(random_state=42)),
    ('xgb', XGBClassifier(use_label_encoder=False, eval_metric='logloss',
random_state=42))
])
```

Appendix C: Environment Setup

Python 3.10 (gpu_39 kernel) Libraries: pandas, NumPy, Scikit-learn, TensorFlow/Keras, XGBoost, pytorch, matplotlib. API from Mistral AI API Dependencies listed in requirement.txt

Appendix D: Project Link and Collaboration

For more information on project's codebase, demonstration and release notes please visit our GitHub repository.

"We greatfully acknowledge the collaboration and support provided by Tech4life Enterprises. Their partnership has been instrumental in the success of this project."

Collaboration letter