MLOPS MINI PROJECTREPORT

KIDNEY DISEASE DETECTION USING MULTI-LAYER PERCEPTRON AND STREAMLIT APPLICATION

INTRODUCTION

Chronic Kidney Disease (CKD) is a major health concern worldwide, often leading to significant morbidity if not diagnosed early. Machine Learning offers powerful tools for early detection by analyzing clinical parameters. In this project, we developed a machine learning model based on a Multi-Layer Perceptron (MLP) classifier to predict the presence of CKD. Furthermore, we built a Streamlit-based web application to allow users to input patient data and receive real-time predictions.

MODEL TRAINING AND DEVELOPMENT

The dataset used for training was provided in CSV format named "new_model.csv". It included both numerical and categorical features relevant to kidney health such as Blood Pressure (Bp), Specific Gravity (Sg), Albumin (Al), Sugar (Su), Blood Urea (Bu), and others.

The initial step involved installing necessary libraries like mlflow, numpy, pandas, matplotlib, and scikit-learn. After loading the dataset, a data cleaning process was conducted. The script checked for and reported duplicate entries and missing values, ensuring a clean and reliable dataset.The dataset was then divided into features (X) and labels (y). A 70-30 train-test split was performed, ensuring that the model could be evaluated on unseen data. Standardization was applied to the features using StandardScaler, improving the model's ability to converge during training.The classification model employed was an MLPClassifier from scikit-learn with the following setup:

- Hidden Layers: One hidden layer with 100 neurons.
- Max Iterations: 1000 for sufficient convergence.
- Random State: 42 for reproducibility.

The model was trained on the scaled training data. After training, predictions were made on the test set and evaluated using Accuracy, Precision, Recall, and F1 Score metrics:

- Accuracy: Approximately 96%,
- Precision: High weighted precision,

- Recall: Excellent sensitivity to CKD instances,
- F1 Score: Balanced measure indicating robust performance.

Additionally, a confusion matrix was plotted for better visualization of true vs. predicted classifications.Finally, the model along with the scaler and label encoders was saved into a single Pickle file (kidney_disease_model.pkl) for portability and deployment. The scaler was also saved separately as scaler.pkl for modular reuse.

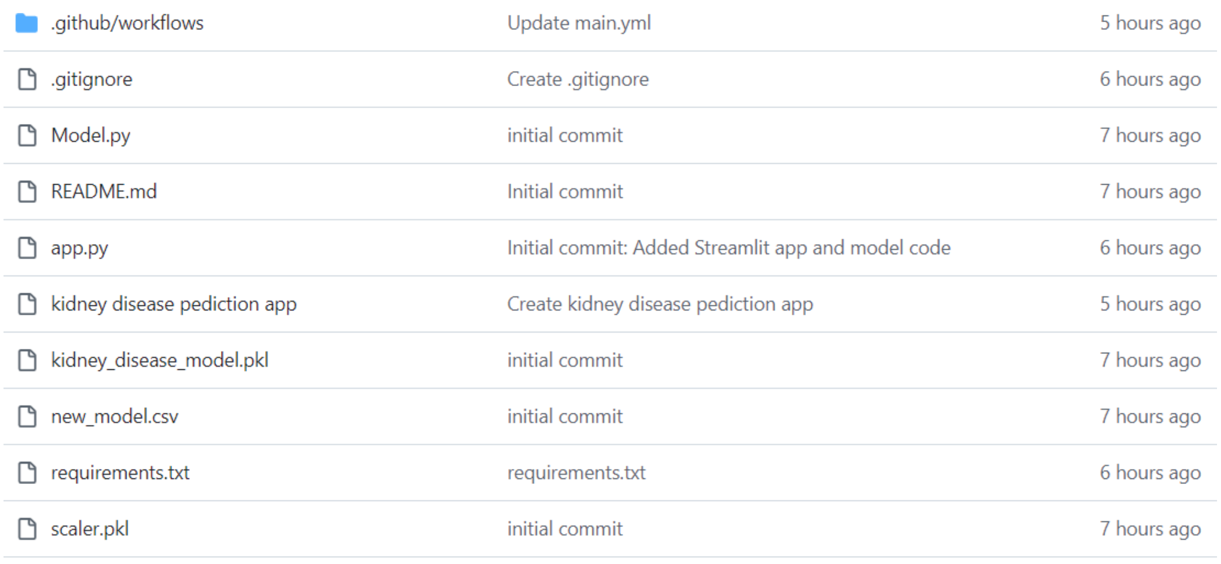
APPLICATION DEVELOPMENT AND DEPLOYMENT WITH STREAMLIT CLOUD

After successfully training and saving the model, we moved toward deploying it for public use by creating an interactive web application using Streamlit. The deployment process was systematic, involving several important steps.The first step was to create a new repository on GitHub. All necessary files, including the trained model (kidney_disease_model.pkl), the scaler (scaler.pkl), the Streamlit app script (app.py), the training script (Model.py), the dataset (new_model.csv), the requirements.txt, and a .gitignore file, were committed and pushed to the repository. The repository was organized with clear commit messages to track the development process efficiently. We also included a workflow file inside the .github/workflows directory for continuous integration support.

Following the GitHub setup, we implemented Continuous Integration (CI). A GitHub Actions workflow (main.yml) was configured to automate the installation of dependencies and verify the code whenever changes were pushed. This ensured that the application remained stable, and any future changes could be smoothly integrated without manual intervention. The CI pipeline automatically set up a Python environment, installed the necessary dependencies from requirements.txt, and prepared the Streamlit app for deployment.Once the CI/CD pipeline was successfully tested and validated, we moved to deploy the application using Streamlit Cloud. In Streamlit Cloud:

- We connected our GitHub repository directly to Streamlit Cloud.
- We selected the branch and app.py as the main entry point.
- The platform automatically installed the required packages listed in requirements.txt.
- After successful deployment, Streamlit Cloud continuously monitored the GitHub repository. Whenever a new commit was pushed, the app was automatically rebuilt and updated.

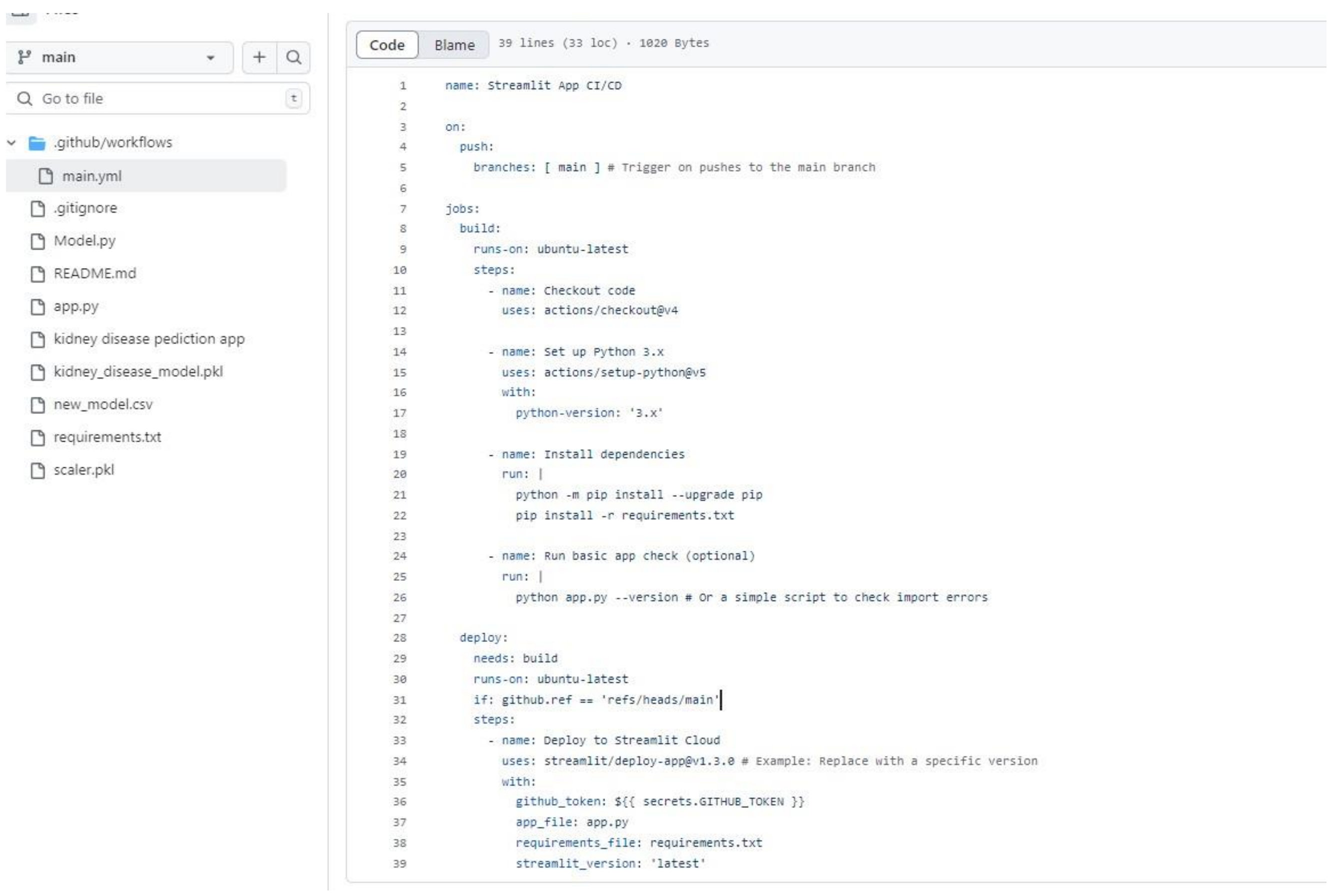
The deployed Streamlit app offered a simple yet functional user interface. Users could manually input various health parameters related to kidney function such as blood pressure, specific gravity, albumin level, blood urea, and more. The app then processed the inputs using the pre-loaded scaler and model to predict the likelihood of Chronic Kidney Disease. The real-time prediction and instant feedback system made the app highly interactive and easy to use.



.github/workflows	Update main.yml	5 hours ago
.gitignore	Create .gitignore	6 hours ago
Model.py	initial commit	7 hours ago
README.md	Initial commit	7 hours ago
app.py	Initial commit: Added Streamlit app and model code	6 hours ago
kidney disease pediction app	Create kidney disease pediction app	5 hours ago
kidney_disease_model.pkl	initial commit	7 hours ago
new_model.csv	initial commit	7 hours ago
requirements.txt	requirements.txt	6 hours ago
scaler.pkl	initial commit	7 hours ago

Figure 1 showing the GitHub repository

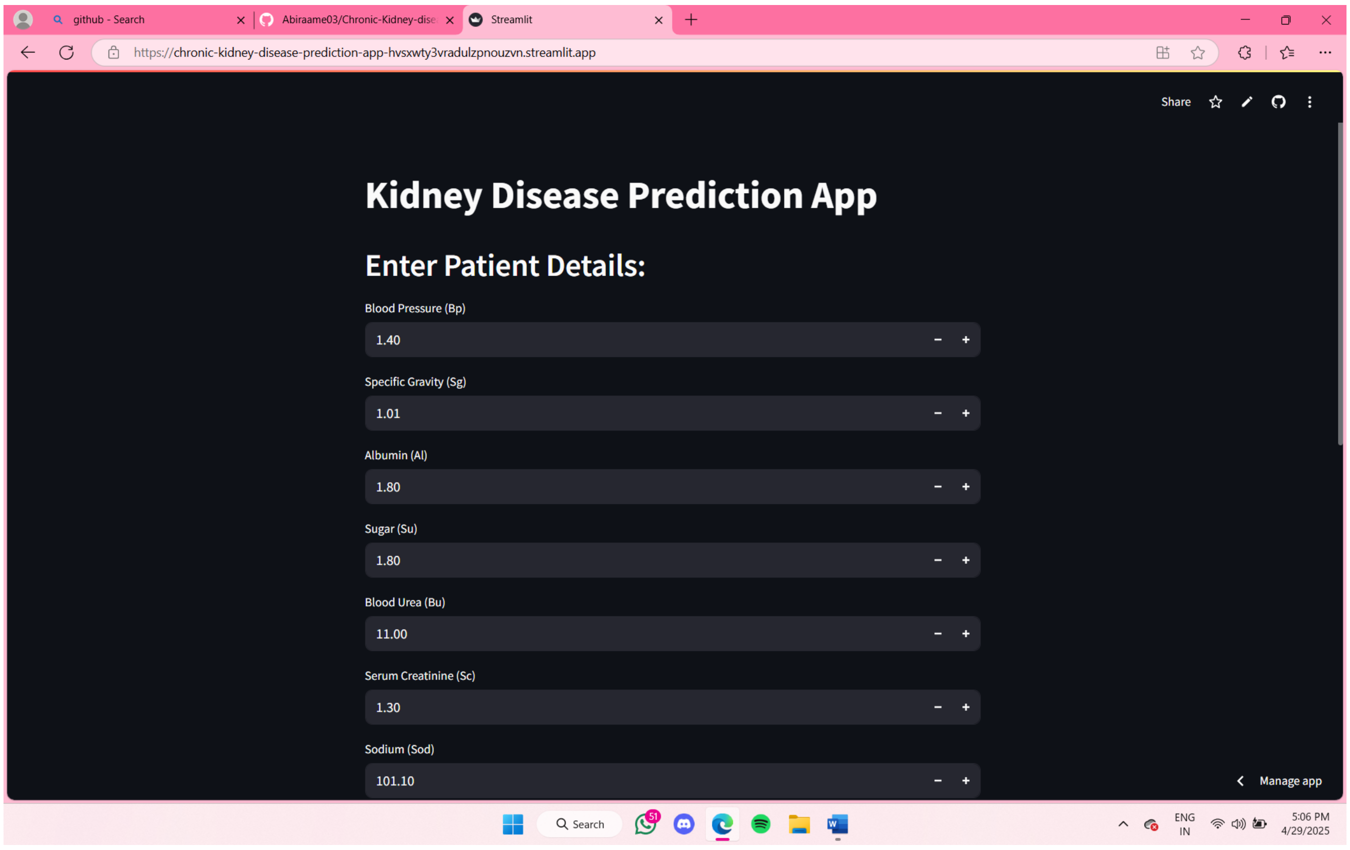
CI INTEGRATION CODE:



```yaml
name: Streamlit App CI/CD

on:
  push:
    branches: [ main ] # Trigger on pushes to the main branch

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Set up Python 3.x
        uses: actions/setup-python@v5
        with:
          python-version: '3.x'

      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt

      - name: Run basic app check (optional)
        run: |
          python app.py --version # Or a simple script to check import errors

  deploy:
    needs: build
    runs-on: ubuntu-latest
    if: github.ref == 'refs/heads/main'
    steps:
      - name: Deploy to Streamlit Cloud
        uses: streamlit/deploy-app@v1.3.0 # Example: Replace with a specific version
        with:
          github_token: ${{ secrets.GITHUB_TOKEN }}
          app_file: app.py
          requirements_file: requirements.txt
          streamlit_version: 'latest'
```

CONCLUSION

This project successfully demonstrated the development of an end-to-end machine learning system for Chronic Kidney Disease prediction. Starting from thorough data preprocessing and model training using a Multi-Layer Perceptron classifier, we achieved high evaluation metrics, ensuring reliability for real-world use. The deployment pipeline was carefully designed by integrating GitHub version control, setting up continuous integration workflows, and finally hosting the application using Streamlit Cloud. The Streamlit application enables users to easily interact with the predictive model through a user-friendly interface, promoting accessibility beyond traditional clinical settings. The adoption of a CI/CD approach ensures that the system remains scalable, maintainable, and ready for continuous improvement.


github - Search
Abiraame03/Chronic-Kidney-dise
Streamlit
https://chronic-kidney-disease-prediction-app-hvsxwty3vradulzpnouzvn.streamlit.app
Share
Manage app
Search
ENG IN
5:06 PM 4/29/2025

Kidney Disease Prediction App

Enter Patient Details:

Blood Pressure (Bp): 1.40

Specific Gravity (Sg): 1.01

Albumin (Al): 1.80

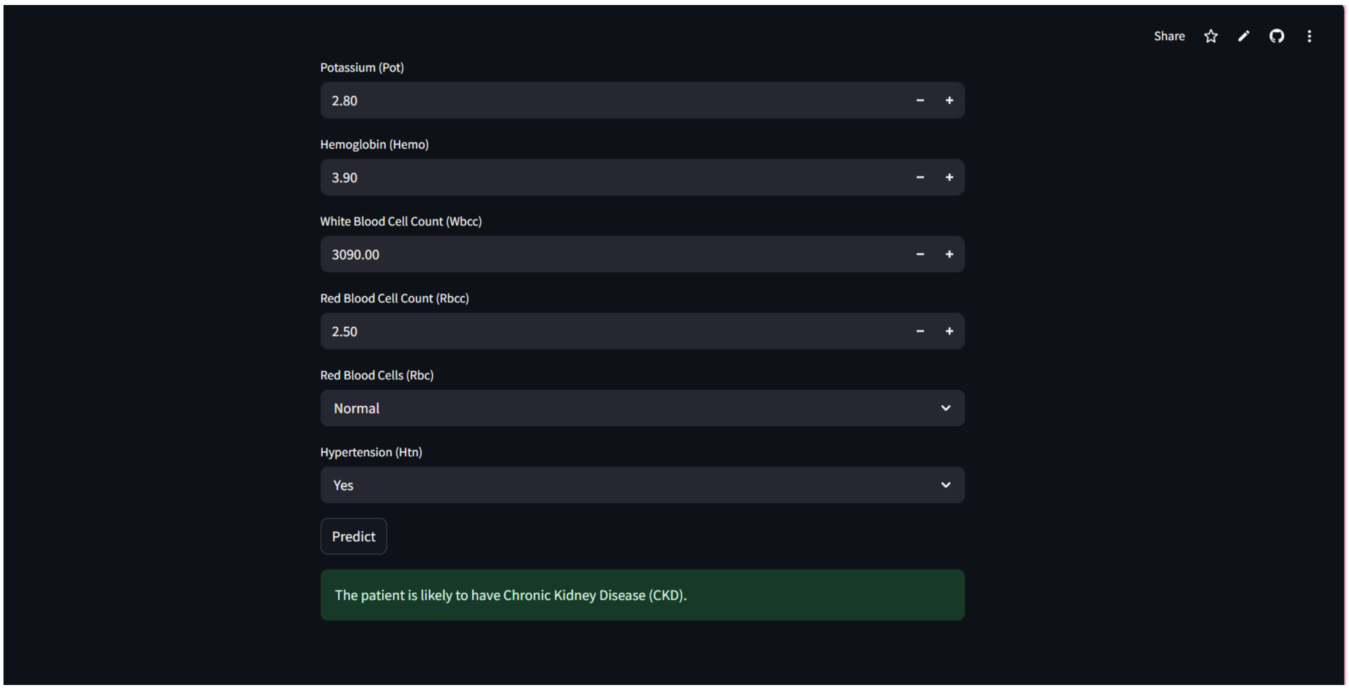
Sugar (Su): 1.80

Blood Urea (Bu): 11.00

Serum Creatinine (Sc): 1.30

Sodium (Sod): 101.10

By entering the health parameters it has predicted :



Potassium (Pot)
2.80

Hemoglobin (Hemo)
3.90

White Blood Cell Count (Wbcc)
3090.00

Red Blood Cell Count (Rbcc)
2.50

Red Blood Cells (Rbc)
Normal

Hypertension (Htn)
Yes

Predict

The patient is likely to have Chronic Kidney Disease (CKD).

Github Repository link: Abiraame03/Chronic-Kidney-disease-prediction-app: Predicts the kidney disease with the parameters

Streamlit app link:

https://chronic-kidney-disease-prediction-app-hvsxwty3vradulzpnouzvn.streamlit.app/

By: Abiraame A.J – 2022116043
Afraah .I -2022116028
B.E Biomedical department
3rd year- 6th sem