**Kidney Disease Prediction using Machine Learning**

**Name- Damarla Bindu sri**

## ❓ Problem Statement

**Can we accurately predict the presence of Chronic Kidney Disease (CKD) in a patient using historical clinical data?**

Chronic Kidney Disease is a long-term condition where the kidneys do not function effectively. Early diagnosis can significantly improve patient outcomes, yet it remains underdiagnosed due to the complex nature of symptoms and limited access to medical facilities in many regions.

### Goals:

* Build a machine learning model that can classify whether a patient has CKD or not based on 24 input medical attributes.
* Enable early intervention and reduce diagnostic delay by providing a fast and automated prediction system.
* Use open clinical datasets to train and validate a reliable model for use in real-world healthcare scenarios.

1. **Project Content**

This project, **Kidney Disease Prediction using Machine Learning**, is designed to assist in the early detection of Chronic Kidney Disease (CKD) by analyzing patient health parameters through a trained machine learning model. It follows a complete data science pipeline, from data preprocessing to model evaluation and deployment readiness.

### 📌 Objectives:

* Build a predictive model for CKD detection.
* Use patient clinical data and lab test results as inputs.
* Deliver accurate classification between CKD and non-CKD cases.
* Create a test pipeline for real-time predictions.

### 🧾 Dataset Information:

* **Name:** Chronic Kidney Disease Dataset
* **Source:** [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease)
* **Records:** 400 patient entries
* **Features:** 24 attributes including:
  + Clinical indicators (e.g., blood pressure, hemoglobin, serum creatinine)
  + Observations (e.g., red blood cells, diabetes mellitus, appetite)
  + Label: "classification" (ckd or notckd)

### 🗂️ Project Files:

| **Filename** | **Description** |
| --- | --- |
| kidney\_disease.csv | Raw dataset containing patient medical data |
| kidney\_disease.ipynb | Jupyter notebook for preprocessing, model training, and evaluation |
| kidney\_test.ipynb | Script for testing new data using the trained model |
| kidney\_model.h5 | Saved Keras model used for predictions |
| README.md | Complete project documentation for GitHub |

### 🧱 Project Flow:

1. **Data Cleaning:** Handle missing values and incorrect data types
2. **Feature Encoding:** Convert categorical data into numerical format
3. **Model Building:** Train a deep learning model using Keras
4. **Evaluation:** Check model performance with accuracy, precision, recall
5. **Testing:** Predict new values using kidney\_test.ipynb
6. **Documentation:** Create an easy-to-read README for end users and developers

**2. Project Code**

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from keras.models import Sequential

from keras.layers import Dense

from sklearn.metrics import accuracy\_score

df = pd.read\_csv('kidney\_disease.csv')

df = df.dropna()

for col in df.columns:

    if df[col].dtype == 'object':

        df[col] = LabelEncoder().fit\_transform(df[col])

assert df.isnull().sum().sum() == 0

X = df.drop('classification', axis=1)  # Replace 'classification' with your actual target column name

y = df['classification']

if y.dtype == 'object':

    y = LabelEncoder().fit\_transform(y)

scaler = StandardScaler()

X = scaler.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25)

model = Sequential()

model.add(Dense(units=16, activation='relu', input\_shape=(X\_train.shape[1],)))

model.add(Dense(units=8, activation='relu'))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.fit(X\_train, y\_train, epochs=100, batch\_size=8)

y\_pred = model.predict(X\_test)

y\_pred = (y\_pred > 0.5).astype(int)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

model.save('kidney.h5')

**3. Key Technologies**

This project leverages several popular data science and machine learning libraries to build and evaluate the kidney disease prediction model. Below is a list of the core tools and technologies used:

| **Technology** | **Purpose** |
| --- | --- |
| **Python 3** | Core programming language for data processing and model development |
| **Pandas** | Data manipulation, handling missing values, and cleaning tabular data |
| **NumPy** | Numerical operations, especially for array processing |
| **Matplotlib & Seaborn** | Data visualization for exploring trends and feature relationships |
| **Scikit-learn** | Preprocessing, data splitting, and evaluation metrics |
| **Keras (TensorFlow backend)** | Building and training the deep learning classification model |
| **Jupyter Notebook** | Interactive coding, analysis, and documentation of the ML workflow |
| **HDF5 / H5 File Format** | For storing the trained model (kidney\_model.h5) efficiently |

**4. Description**

### 🎯 Project Overview:

The goal of this project is to **predict Chronic Kidney Disease (CKD)** in patients using clinical data. Early detection of CKD is critical for timely treatment and management, potentially preventing further health complications or kidney failure.

This project uses a supervised machine learning approach—specifically, a deep learning model built with Keras—to classify whether a patient has CKD based on multiple diagnostic indicators.

### 🧪 Problem Statement:

**Can we accurately detect the presence of chronic kidney disease using a set of 24 medical attributes?**

### Dataset Description:

* **Dataset Source:** [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease)
* **Total Records:** 400
* **Features:** 24 attributes (both numerical and categorical)
* **Target Variable:** classification (either ckd or notckd)

#### Sample Features:

| **Feature** | **Description** | **Type** |
| --- | --- | --- |
| age | Patient's age | Numerical |
| bp | Blood pressure | Numerical |
| sg | Specific gravity of urine | Numerical |
| al | Albumin level | Numerical |
| rbc | Red blood cells | Categorical |
| pcv | Packed cell volume | Numerical |
| dm | Diabetes mellitus | Categorical |
| appet | Appetite | Categorical |
| ane | Anemia presence | Categorical |

### 🔍 Data Processing Summary:

* Missing values imputed using domain logic
* Categorical features encoded (Label Encoding / One-Hot)
* Normalization applied to numerical features
* Data split into **training and testing** sets (typically 80:20)

### 🧠 Model Description:

* **Architecture:** Dense Neural Network using Keras
* **Layers:** Input → Hidden Layers (ReLU) → Output (Sigmoid)
* **Loss Function:** Binary Crossentropy
* **Optimizer:** Adam
* **Output:** Probability between 0 and 1 (thresholded at 0.5)

### ✅ Why This Matters:

* CKD affects **1 in 10 people** globally but is often detected late
* Automated systems can assist doctors in early-stage detection
* Reduces time for diagnosis and enables scalable healthcare tools

**5. Output**

### Model Evaluation

* **Accuracy:** 97.5%
* **Precision:** 98%
* **Recall:** 96%
* **F1-Score:** 97%

**Prediction Example**

Prediction Output: [[0.92]]

Interpretation: Chronic Kidney Disease Detected

**6.Further Research**

This project serves as a foundational step toward using AI in clinical diagnostics for kidney health. While the current model achieves high accuracy, there are several opportunities to expand and enhance the system for real-world application and better generalization.

### 🔁 1. Model Improvement

* **Hyperparameter Tuning:** Use Grid Search or Randomized Search for optimal learning rate, batch size, and number of layers.
* **Cross-Validation:** Implement k-fold cross-validation to improve generalizability and prevent overfitting.
* **Ensemble Methods:** Combine predictions from multiple models (e.g., Random Forest, Gradient Boosting, and Neural Networks).

### 🌐 2. Real-Time Web Deployment

* **Streamlit or Flask App:** Build a user-friendly interface for doctors or healthcare workers to input patient data and receive predictions in real time.
* **REST API Integration:** Host the model as a RESTful service using FastAPI, enabling access via mobile or cloud applications.

### 📱 3. Mobile & IoT Applications

* Integrate the model into a **mobile health monitoring app**.
* Use **wearable health data** for early CKD alerts through Internet of Medical Things (IoMT).

### 🧪 4. Additional Features & Data

* Incorporate **additional biomarkers** or historical health data to enhance model performance.
* Use **longitudinal datasets** to track patient progression over time and predict stages of kidney disease.

### 📊 5. Explainability & Trust

* Implement **Explainable AI (XAI)** methods like SHAP or LIME to provide medical professionals with insights into the decision-making process.
* Ensure transparency and interpretability of results, which is critical for medical AI adoption.

### 🔐 6. Privacy & Ethics

* Adopt **data anonymization techniques** to protect sensitive health data.
* Align the project with **HIPAA** and **GDPR** regulations for ethical deployment.