CHRONIC KIDNEY DISEASE

An AI-powered predictive system that analyzes routine clinical and laboratory data to identify patients at risk of chronic kidney disease for early intervention.

Presented to

Hope AI

Presented by

Madhu

TABLE OF CONTENTS

PROBLEM STATMENT AND OBJECTIVE

Problem Statement

A hospital management needs an automated tool to flag patients at risk of Chronic Kidney Disease (CKD) early, based on routine lab and clinical parameters.

Objective

Build a binary classifier that predicts CKD (1) vs. Non-CKD (0).

DATASET OVERVIEW

Size

PATIENT RECORDS
399

FEATURES
25

Class balance / Target

CKD - YES
250 records (62.7%)

CKD - NO
149 records (37.3%)

Feature types

Continuous numeric (11):
age, bp, bgr, bu, sc, sod, pot, hrmo, pcv, wc, rc

Ordinal categorical (3):
sg , al, su

Nominal categorical (11):
rbc, pc, pcc, ba, htn, dm, cad, appet, pe, ane, classification

DATASET OVERVIEW

Data-Type & Cardinality Summary

Numerical Data

FEATURES	TYPE	UNIQUE LEVELS
Age, Bp, Bgr, Bu, Sc Sod, Pot, Hrmo, Pcv, Wc, Rc	Numeric	_____

Ordinal Data

FEATURES	TYPE	UNIQUE LEVELS
Sg (1-5)Sod, Pot	Mapped	5 levels (a→1...e→5)
Al (0-5)	Numeric	6 levels
SU (0-5)	Numeric	6 levels

DATASET OVERVIEW

Nominal Data

FEATURES	TYPE	UNIQUE LEVELS
Rbc	Binary	Normal, Abnormal
Pc	Binary	Normal, Abnormal
pcc	Binary	Present, Not Present
Ba	Binary	Present, Not Present
Htn, Dm, Cad, pe, Ane, Classification	Binary	Yes, No
Appet	Binary	Good, Poor

DATA PRE-PROCESSING

Loaded raw CSV
Read in the CKD.csv file and inspected each column for type and missing values.

Mapped ordinal variables to integers
sg: a–e → 1–5
al, su: “0”–“5” → 0–5 integers

Cast continuous features to numeric
age, bp, bgr, bu, sc, sod, pot, hrmo, pcv, wc, rc → floats

One-hot encoded nominal categoricals
one-hot encoded, retaining all levels (no dummy drop)

MODEL DEVELOPMENT AND EVALUATION

Maximize F1-score

Cross Validation	Pipeline	Classifer
5-fold StratifiedKFold (shuffle=True, random_state=42	StandardScaler	scoring="f1_weighted" refit=True n_jobs=-1, verbose=3,

Used GridSearchCV for hyperparameter tuning

MODEL DEVELOPMENT AND EVALUATION

Evaluation Metrics

- ACCURACY
- PRECISION, RECALL, F1-SCORE
- AUC-ROC
- CONFUSION MATRIX

Algorithms Tried

- LOGISTIC REGRESSION
- DECISION TREE
- SUPPORT VECTOR MACHINE
- RANDOM FOREST

MODEL DEVELOPMENT AND EVALUATION

Parameter Grids

Logistic Regression

- penalty: ["l1", "l2"]
- solver: ["liblinear"]
- C: [0.01, 0.1, 1, 10]

Decision Tree

- criterion : ["gini", "entropy"]
- max_features: ["auto", "sqrt"]
- splitter : ["best", "random"]

Random Forest

- n_estimators : ["10", "100","200"]
- max_depth : ["5", "l0"]
- min_samples_split : ["2", "5"]

SVM

- criterion : ["gini", "entropy"]
- max_features: ["auto", "sqrt"]
- splitter : ["best", "random"]

Minimal Code Snippet

```
param_grid = {
    'solver':       ['newton-cg', 'lbfgs', 'liblinear', 'saga'],
    'penalty':      ['l1', 'l2', 'elasticnet', 'none'],   # Not all solvers support all penalt
    'C':            [0.01, 0.1, 1, 10, 100],              # Inverse of regularization strength
    'max_iter':     [100, 200, 500],                      # Maximum iterations for the solver
    'fit_intercept': [True, False],                       # Whether to fit the intercept (bias
    'l1_ratio':     [0, 0.5, 1]                           # Used only if penalty='elasticnet'
}
```

FINAL RESULTS

After preprocessing, feature engineering, and hyperparameter tuning via 5-fold GridSearchCV, we evaluated four candidate models on the 20% hold-out test set. The key metrics are:

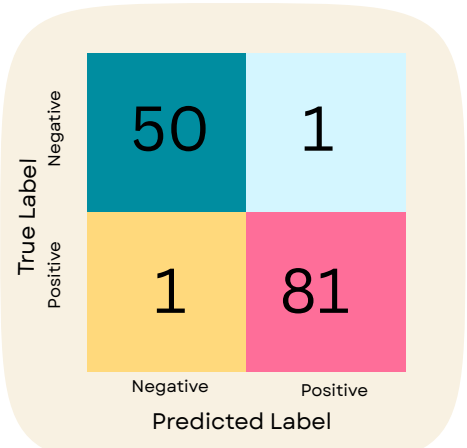
Performance Comparsion

Algorithm	Accuracy	ROC AUC	F1 Score
Logistic Regression	0.97	0.97	0.97
Random Forest	0.98	0.99	0.98
Decision Tree	0.97	0.97	0.97
SVM	0.96	0.99	0.96

Key takeaway: *Random Forest achieved the best balance of sensitivity and specificity, with the highest F1-Score (0.98) and ROC-AUC (0.99) on unseen data.*

Confusion Matrix(Random Forest)

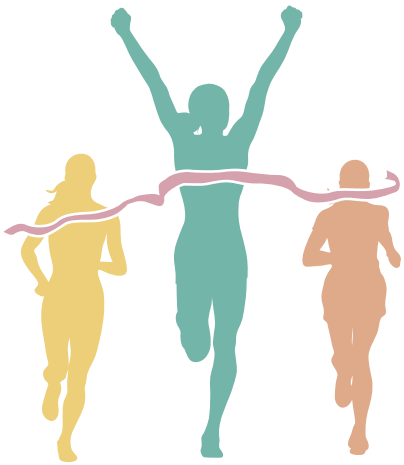
```
==================== CKD Prediction Results====================
 Test Case  1:  CKD Detected
 Test Case  2:  No CKD
```



FINAL RESULTS

In a powerful study of Chronic Kidney Disease prediction, four models—Logistic Regression, Decision Tree, SVM, and Random Forest—were rigorously tuned via 5-fold GridSearchCV. Random Forest dominated with 98 % accuracy, 0.99 ROC AUC, and 0.98 F1-score, delivering exceptional sensitivity and specificity. Future work: external validation and ensemble stacking for impactful clinical insights.

Random Forest



PROJECT STRUCTURE AND GITHUB REPO

```
CKD-Prediction/
├── 01_data/
│   ├── raw/                  # Original CKD.csv
│   └── processed/            # CKD_cleaned.csv
├── 02_notebooks/             # Jupyter analyses (EDA, FE, modeling, evaluation)
├── 03_src/                   # Python scripts (preprocessing.py, modeling.py, etc.)
├── 04_reports/               # Final report PDF
├── 05_models/                # Serialized model (best_model.pkl)
├── requirements.txt          # Dependencies
├── README.md                 # Project overview & instructions
└── .gitignore
```

Access the full code, notebooks, and report at:
chronic-kidney-disease-classifier

FOR INQUIRIES, CONTACT ME

www.madhusudhanan.online

+1-972-946-6351

madhusudhanan.jayaram@gmail.com