Chronic Kidney Disease (CKD) Prediction using Support Vector Machine (SVM)

1. Introduction

This project aims to predict Chronic Kidney Disease (CKD) using machine learning techniques, specifically the Support Vector Machine (SVM) algorithm. The dataset undergoes preprocessing, feature scaling, and model training with hyperparameter tuning to achieve optimal performance.

---

2. Data Preprocessing

2.1 Loading the Dataset

- The dataset is loaded using Pandas.
- Columns **"Albumin_to_Creatinine_Ratio"** and **"Glomerular_Filtration_Rate"** are dropped.

2.2 Exploratory Data Analysis (EDA)

- The dataset's structure and summary statistics are examined.
- The distribution of the target variable is visualized using bar plots.
- A correlation heatmap is plotted to understand feature relationships.

2.3 Splitting Data

- Features (**X**) and target variable (**y**) are separated.
- Data is split into **80% training** and **20% testing** using `train_test_split`.

2.4 Feature Scaling

- Standardization is applied using `StandardScaler`.

---

3. Model Training - Support Vector Machine (SVM)

3.1 Model Selection and Hyperparameter Tuning

- An SVM classifier is used with different kernels (`poly`, `rbf`, `sigmoid`).
- A grid search is performed with **Repeated Stratified K-Fold cross-validation** to find the best parameters.

3.2 Model Evaluation

- **Training Accuracy** and **Testing Accuracy** are calculated.
- If the training accuracy is significantly higher than testing accuracy, overfitting is suspected.
- **Classification Report** provides performance metrics (F1-score, Precision, Recall).
- **Confusion Matrix** is plotted to analyze predictions.

3.3 ROC-AUC Curve

- The **ROC-AUC score** is calculated to evaluate model performance.
- The **ROC curve** is plotted to visualize the trade-off between True Positive Rate and False Positive Rate.

---

4. Model Saving and Deployment

- The trained model is saved using **Pickle (`svm_model.pkl`)** for future use.

---

5. Summary

- The dataset was preprocessed, including feature selection and scaling.
- The **SVM model** was trained with hyperparameter tuning using Grid Search.
- Model evaluation metrics indicated good performance.
- The trained model was saved for deployment.

---

6. Future Improvements

- Testing with additional feature selection techniques.
- Experimenting with different machine learning models (e.g., Random Forest, XGBoost).
- Implementing a user-friendly web interface for predictions.
