Chronic Kidney Disease Detection

Aeris Li

Cheryl Jiao

Zhe Sun

Richard Yang

Jean Chao

Group 6 03/08/2023

Agenda

CONTENTS

Problem Statement 01

- Project Background
- Problem Statement
- Project Purpose

Definition of variables & Exploratory Data Analysis 02

- Definition of Variables
- Data Preprocessing
- Exploratory Data Analysis

Methodology 03

- Feature Selection
- Model Construction
- Model Result & Comparison
- Model Validation

Conclusion & Future Improvement 04

- Final Model & Feature Importance
- Business Value
- Future Improvements

/01 Problem Statement

Problem Statement

Project Background

According to CDC, an estimated **15%** of US adults have Chronic Kidney Disease (CKD), and **9 in 10 adults** with CKD do not know they have CKD

Problem Statement

The high prevalence of CKD in the U.S. motivates us to develop a **predictive model** that uses a patient's health-related information, such as medical history, lab results, and demographic factors, to **identify patients** who are at high risk of developing CKD and thus **empower healthcare providers**

Purpose

1. **Early Detection: I**dentify individuals at risk of developing CKD at an early stage, allowing for early intervention and treatment
2. **Improved Diagnosis:** A predictive model can provide a more accurate diagnosis, reducing the rate of misdiagnosis
3. **Cost Savings:** Early detection and effective treatment can reduce the cost of managing CKD

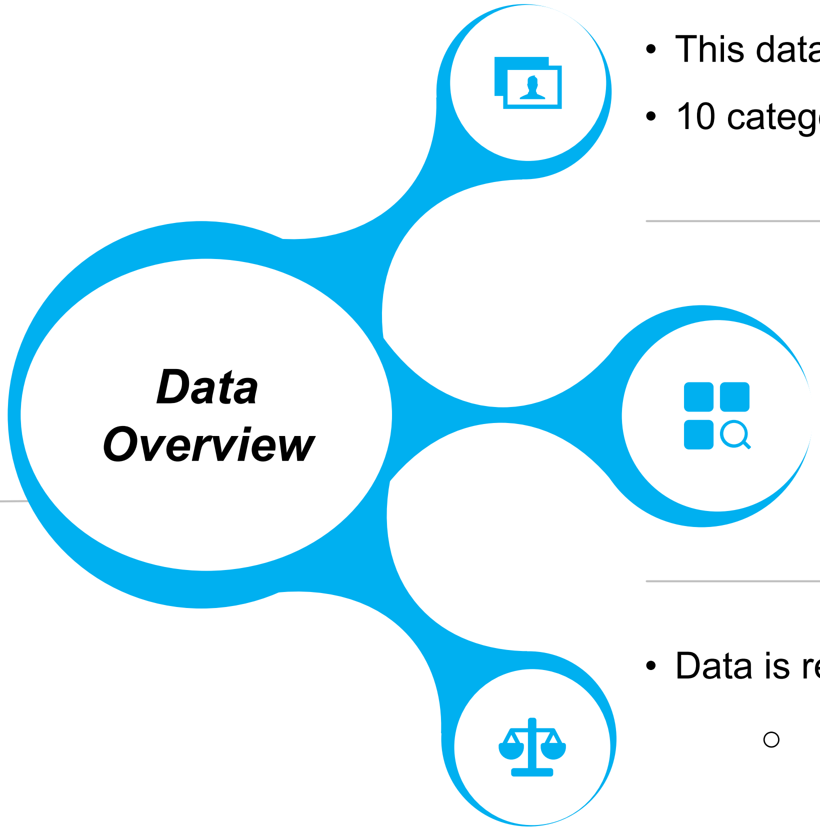
/02

Definition of variables & Exploratory Data Analysis

Definition of variables

Data Source:

Early stage of Indians Chronic Kidney Disease (CKD) dataset was originally posted on UC Irvine Machine Learning Repository in 2015. The data was collected over a 2-month period from hospitals in India.

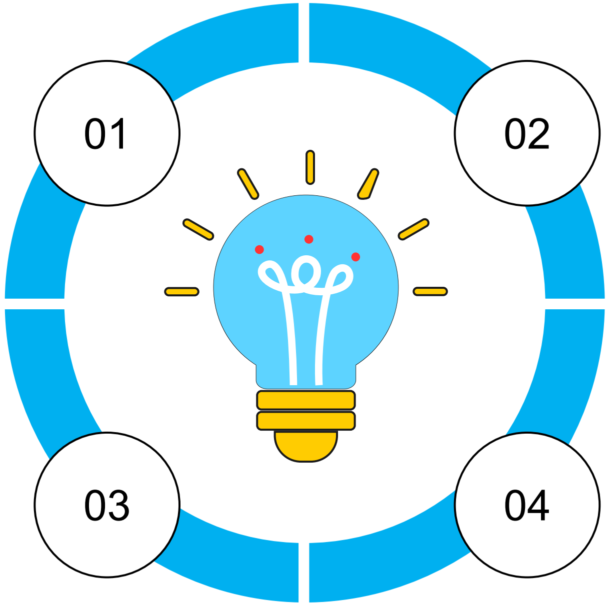


- This dataset contains 24 features
- 10 categorical and 14 numerical features

- This is a classification problem, with target variable being binary outcomes (CKD/non-CKD)

- Data is relatively balanced
  - CKD: Non-CKD = 6 : 4

Data Preprocessing



Data type conversion

Converting necessary columns to numeric type:

- packed_cell_volume
- white_blood_cell_count
- red_blood_cell_count

Data Wrangling

- '\tno' -> 'no’
- '\yes' -> 'yes’
- 'ckd\t' -> 'ckd’
- 'notckd' -> 'not ckd'

Maps the target values

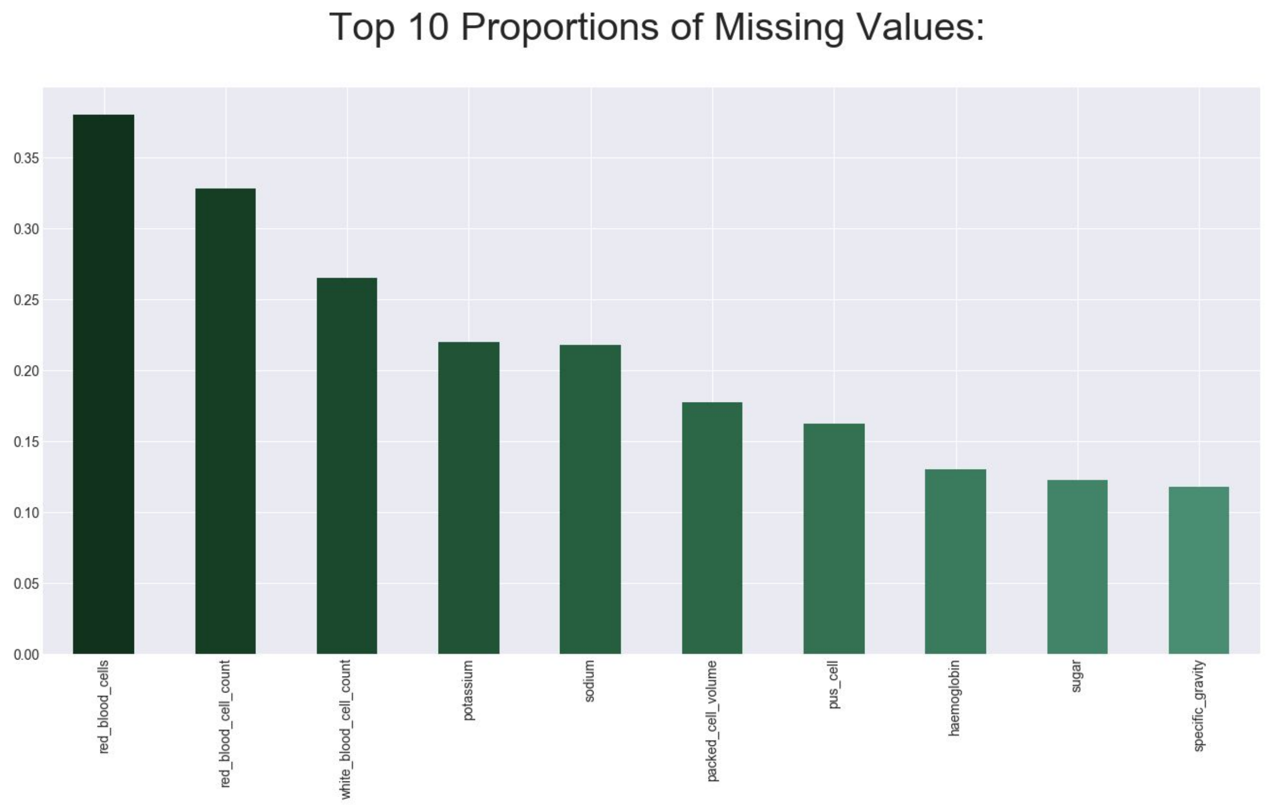
Maps the target values to 0 or 1:

Encoding for categorical variables

Since all categorical variables have only 2 categories, use label encoder to convert categorical variables into binary variables(0s and 1s).

Data Preprocessing Continued

Handling missing values: Random imputation & Mode imputation

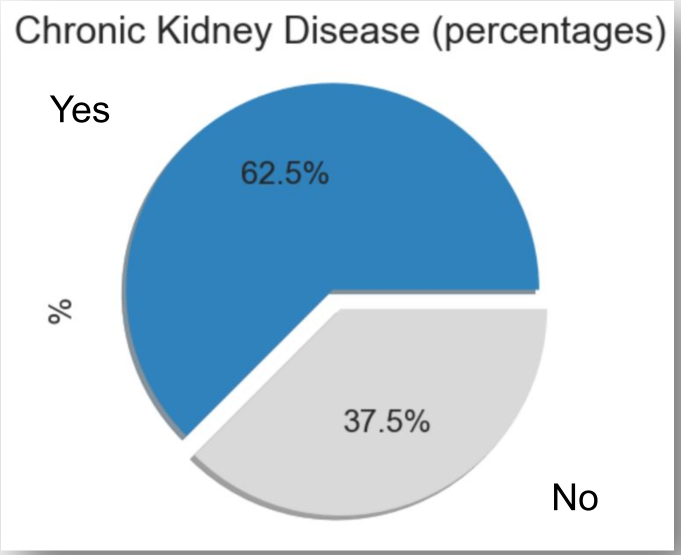


- **Random imputation:** for all numerical variables and categorical variables with a large portion of missing values ("red_blood_cells" and "pus_cell")
- This method preserves the statistical properties of the original data and avoids bias.

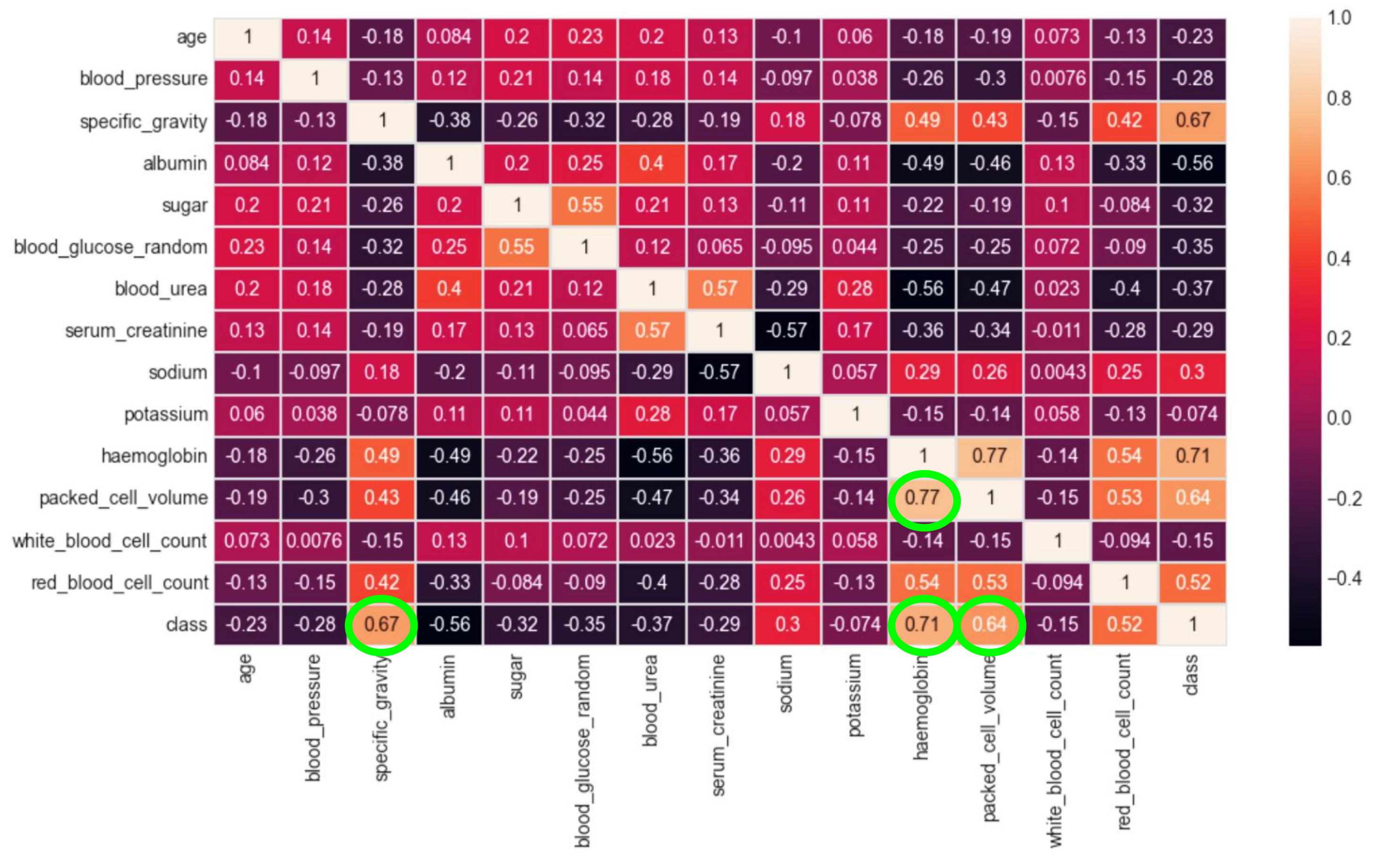
- **Mode imputation:** for categorical variables with a small portion of missing value

Exploratory Data Analysis

Our response variable is relatively balanced
(CKD: 62.5% vs Non-CKD: 37.5%)



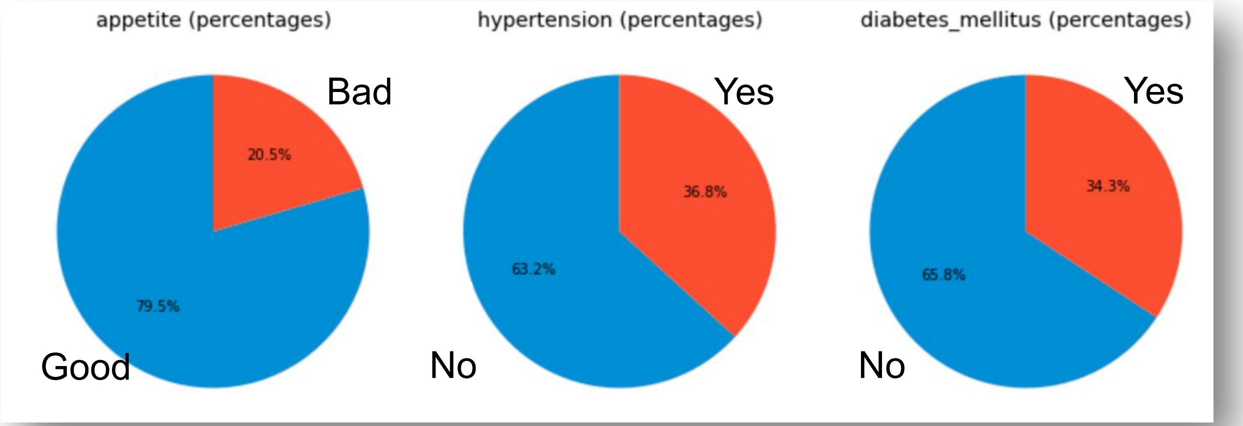
Low correlation and linearly dependency between features

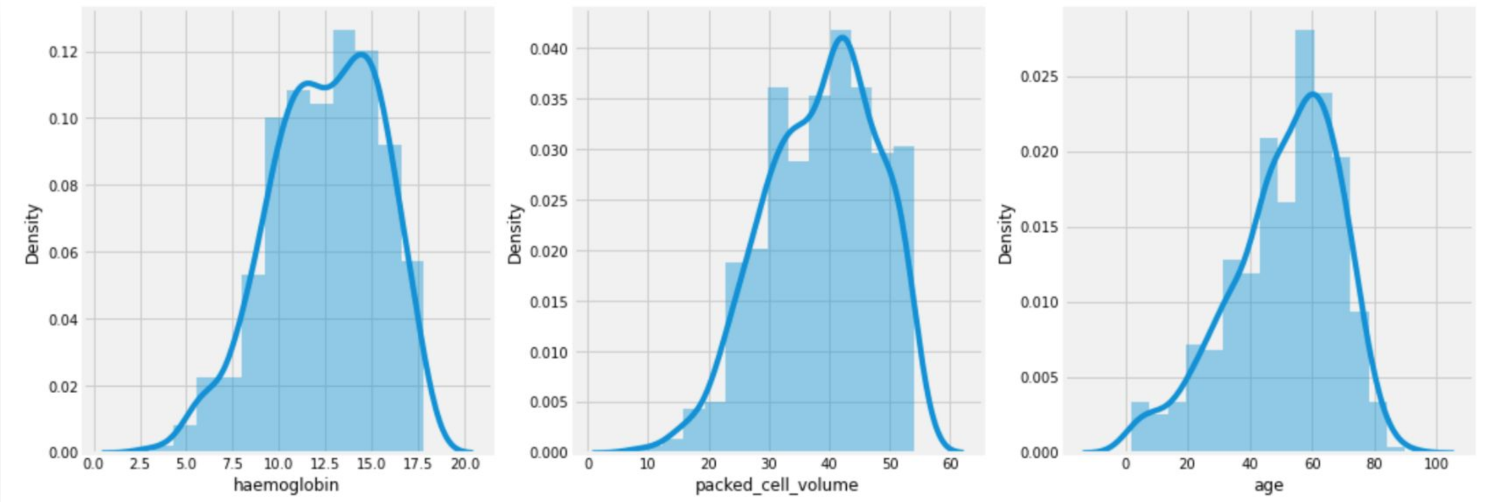


Feature Visualization

Categorical variables part

- Most of the categorical features are relatively balanced





Numerical variables part

- Most of our numerical features have a normal distribution

/03

Methodology

Feature Selection

We used **Recursive Feature Elimination (RFE)** method to fit a **Random Forest Regressor**, which generates the optimal number of features to select, as well as the features to select given that optimal number.

01

Methodology

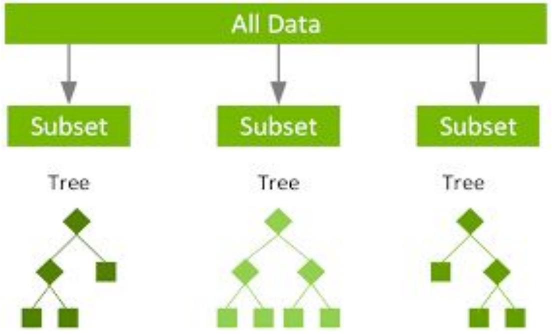
02

Result

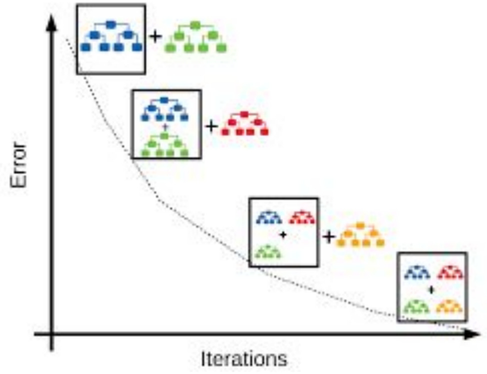
12 Features Selected:
Age, Specific_Gravity, Albumin, Blood_Glucose_Random, Blood_Urea, Serum_Creatinine, Haemoglobin, Packed_Cell_Volume, Red_Blood_Cell_Count, Hypertension, Diabetes_Mellitus, Appetite

Model Overview

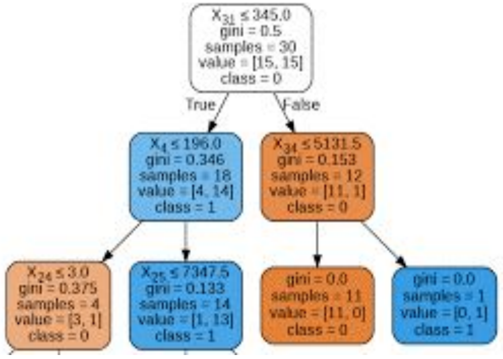
1 XgBoost



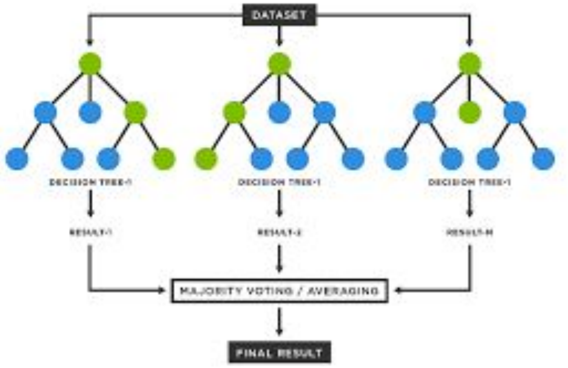
2 Gradient Boosting Classifier



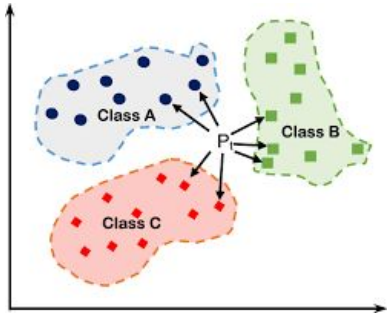
3 Decision Tree Classifier



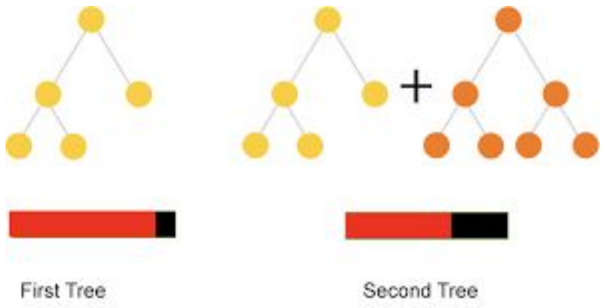
4 Random Forest Classifier



5 KNN



6 Cat Boost



Model Methodology

01

Preparation

- **Set a seed** to ensure reproducibility of the results
- Use the **70-30 split** (70% of the data is used for training and 30% is used for testing) for our dataset.

02

Construction

- Consider factors such as the size and complexity of our dataset, the interpretability of the model, and the computational resources available to **choose models** for our **binary classification** problem

03

Optimization

- Use **grid search hyperparameter tuning technique** to improve the performance of models and ensure that the hyperparameters are well-defined and reproducible

04

Validation

- Compare their performance using metrics such as Precision, Recall, F-1 Score and Adjusted R Squared.
- Use **K-fold Cross-Validation** to better evaluate the performance of models and **detect overfitting**.

Model Result & Comparison

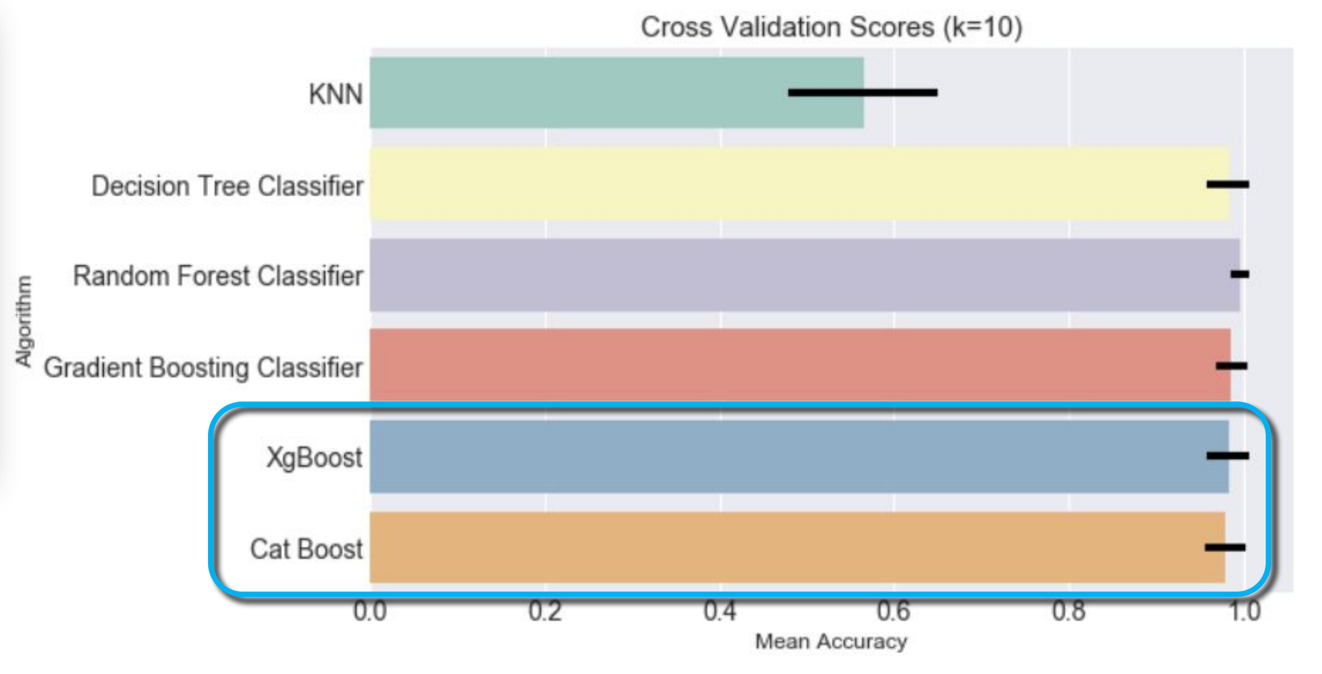
	Gradient Boosting Classifier	XgBoost	Decision Tree Classifier	Random Forest Classifier	KNN	Cat Boost
F1	0.97	0.98	0.90	0.97	0.64	0.98
Precision	1.00	1.00	0.88	1.00	0.67	1.00
Recall	0.94	**0.96**	0.92	0.94	0.60	**0.96**
Adj R^2	0.89	0.85	0.89	0.93	-0.22	0.89

Based on the result, XGboost and Cat Boost has the best recall performance

Model Validation

K-fold cross-validation:
- Helps to get a more accurate estimate of the model's true performance on unseen data
- Helps to identify the model with the best generalization performance, which is less likely to overfit to the training data

	CrossValMeans	CrossValerrors	Algorithm
0	0.564286	0.085714	KNN
1	0.982143	0.023958	Decision Tree Classifier
2	0.996429	0.010714	Random Forest Classifier
3	0.985714	0.017496	Gradient Boosting Classifier
4	0.982143	0.023958	XgBoost
5	0.978571	0.023690	Cat Boost



- Higher means and lower errors are preferred

/04

Conclusion & Future Improvement

Why XgBoost?

Recall:

Cat Boost	XgBoost
0.96	0.96

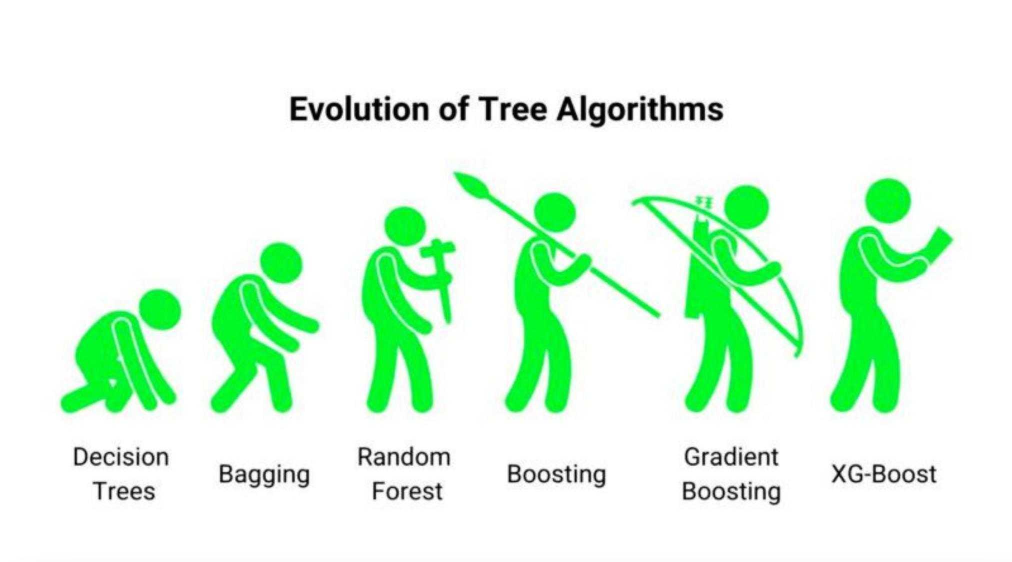
Compared to Cat Boost, XgBoost has:

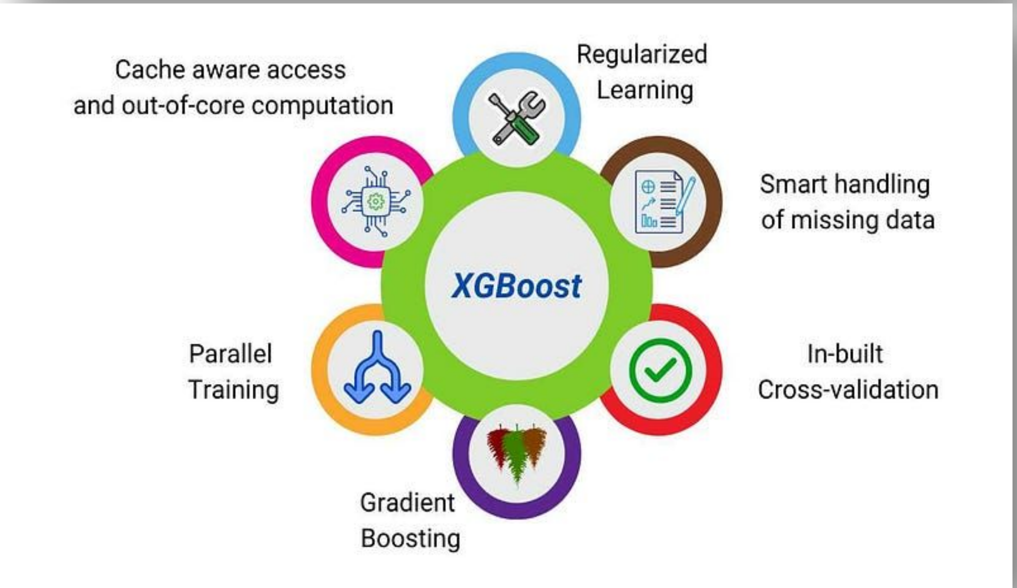
1. **Faster training times**
2. **More flexible hyperparameter tuning**
3. **Better handling of high-dimensional data**

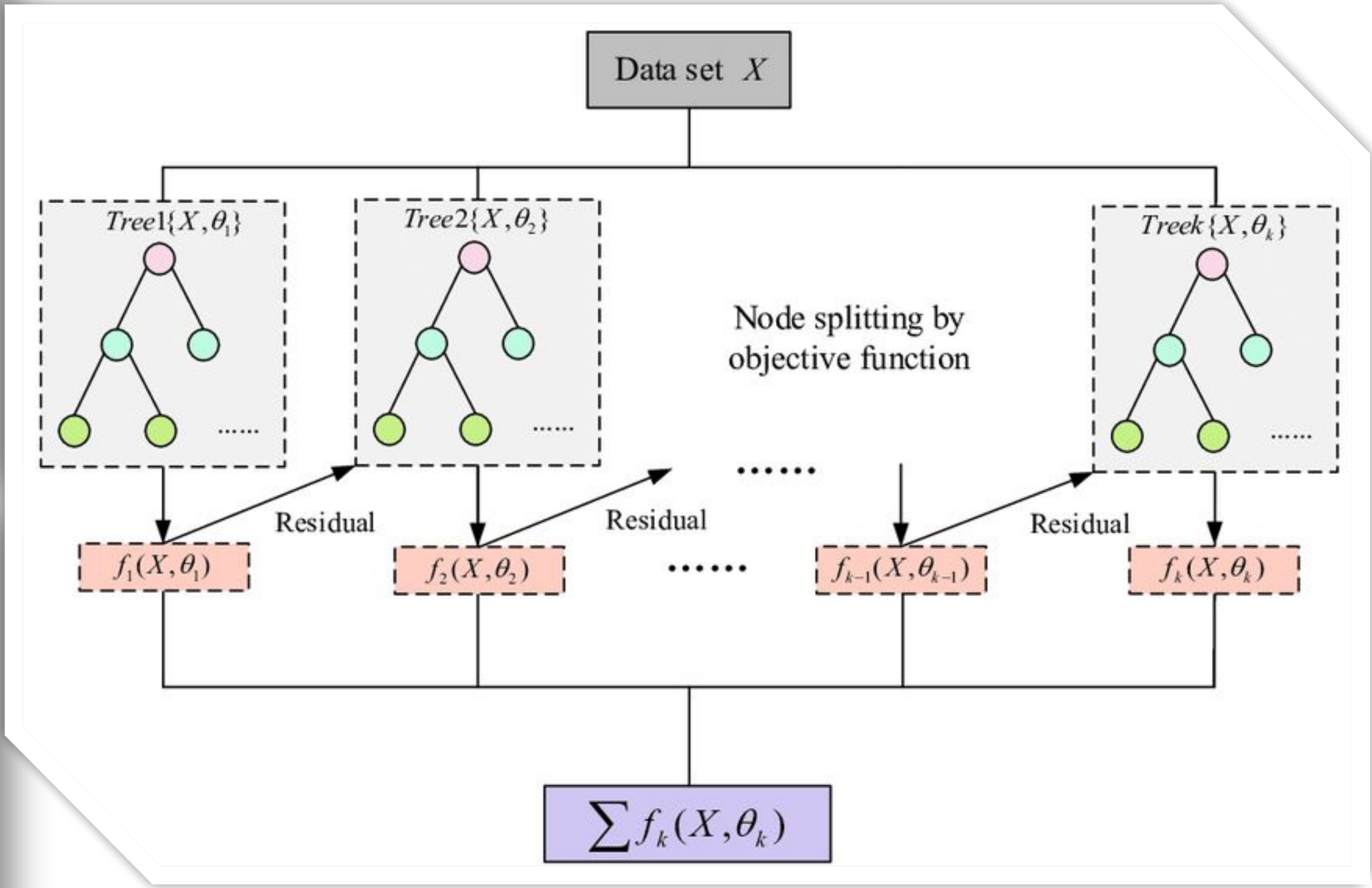
As XGboost is so efficient and accurate,

We choose XGboost as our final Model!

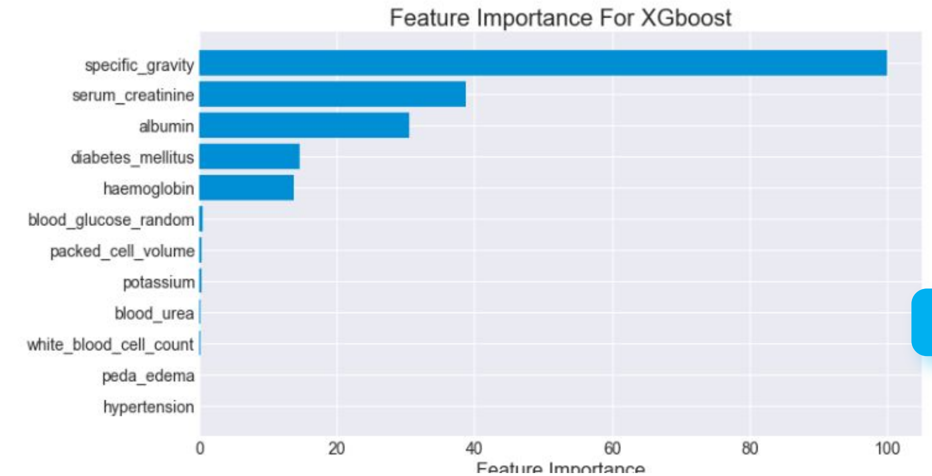
Final Model

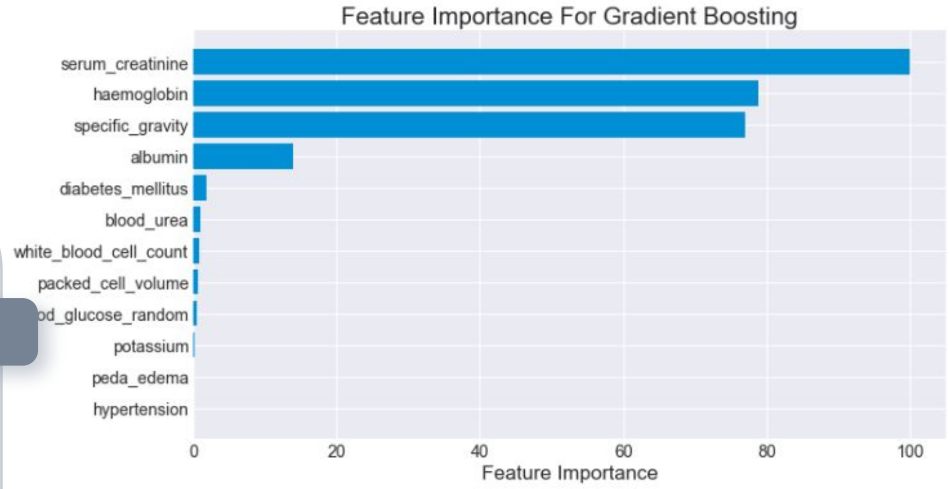

Evolution of Tree Algorithms
Decision Trees
Bagging
Random Forest
Boosting
Gradient Boosting
XG-Boost


Cache aware access and out-of-core computation
Regularized Learning
Smart handling of missing data
XGBoost
Parallel Training
In-built Cross-validation
Gradient Boosting


Data set X
$Tree1\{X,\theta_1\}$
$Tree2\{X,\theta_2\}$
$Treek\{X,\theta_k\}$
Node splitting by objective function
Residual
Residual
Residual
$f_1(X,\theta_1)$
$f_2(X,\theta_2)$
$f_{k-1}(X,\theta_{k-1})$
$f_k(X,\theta_k)$
$\sum f_k(X,\theta_k)$

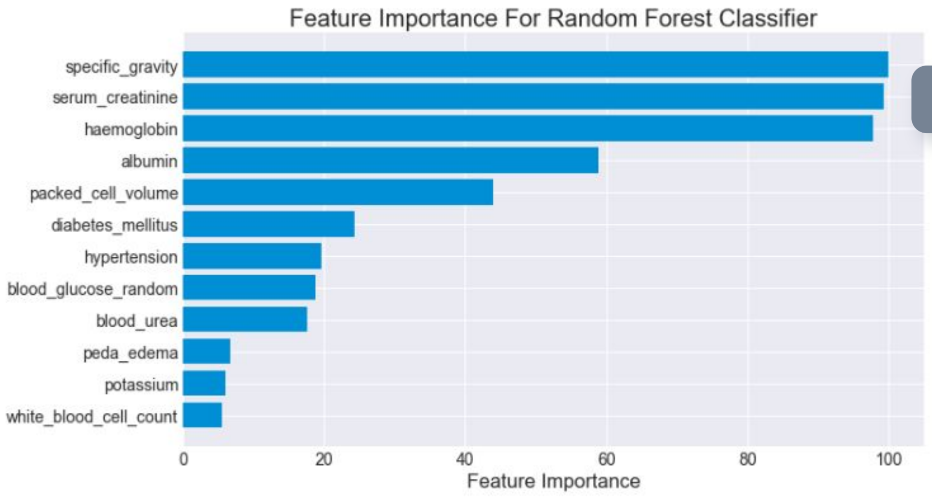
Feature Importance

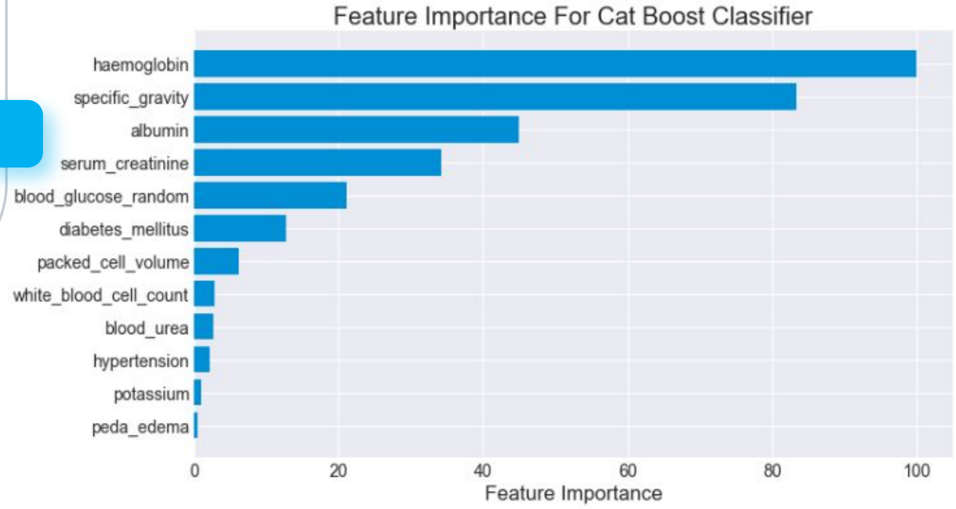




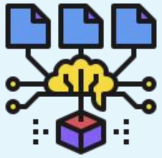
We Generated a Feature Importance Score on a 0-100 scale.
Based on Visualization results, it is clear that some features are commonly important among different models:

- **Specific Gravity**
- **Haemoglobin**
- **Hypertension**
- **Albumin**





Business Value

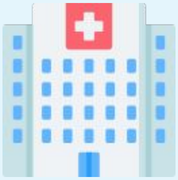


Our predictive model

Health management App

Self-test CKD

- ❑ Upload their medical report
- ❑ The model can identify risk of CKD



Partner with healthcare providers

- ❑ If result is positive, advise the patient to schedule an appointment with doctors for further evaluation

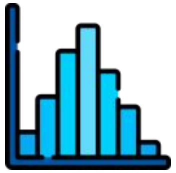
Partner with insurance providers

- ❑ Patient can upload their medical report on insurance app
- ❑ Insurance fee can be determined on CKD risk

Future Improvements

More data observation

- Help reduce the sampling error, capture more diverse patterns, and allow model to generalize well in unseen dataset



Transform the skewed variables

- A few of our numerical variables have skewness, we may use Preprocessing / MinMaxScaler / StandardScaler to transform them and to reduce the potential bias.

Model efficiency

- Consider running time, memory usage, and energy consumption when selecting models

Case by case Analysis

- Allows healthcare provider to examine each patient’s medical history to make informed decision on predict outcomes.

Thanks

Q&A

Reference:

L. Jerlin Rubini. (2015, July 3). *Chronic_Kidney_Disease Data Set*. UCI Machine Learning Repository: Chronic_kidney_disease Data set. Retrieved March 8, 2023, from https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease