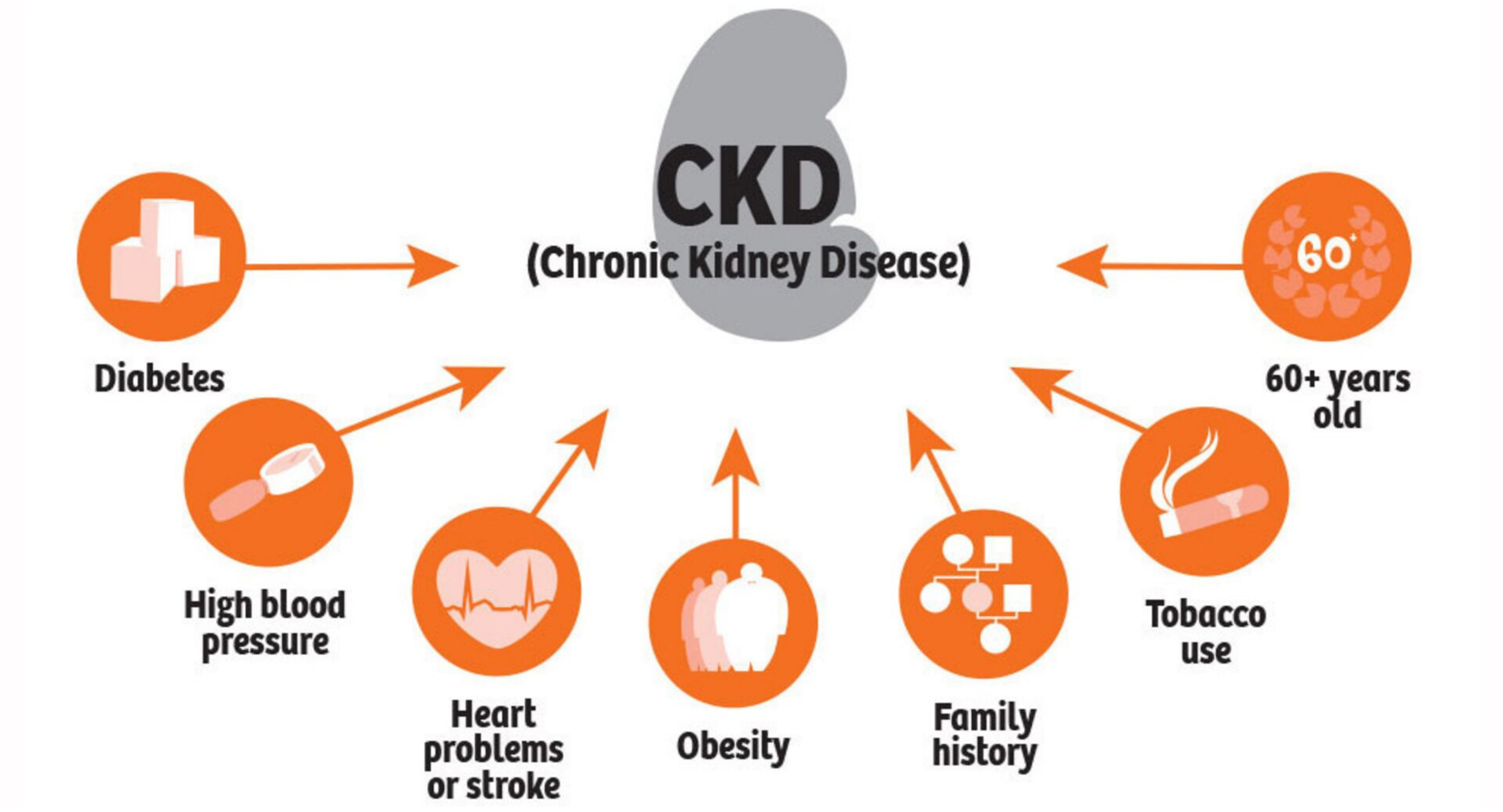
COMPARATIVE ANALYSIS OF SUPERVISED MACHINE LEARNING ALGORITHMS FOR CHRONIC KIDNEY DISEASE DETECTION

LAB 5-6 REPORT



OVERVIEW

CHRONIC KIDNEY DISEASE

Kidney diseases are disorders that affect the functions of the kidney. During the late stages, kidney diseases can cause kidney failure to prevent chronic kidney disease-CKD by utilising machine learning techniques to diagnose kidney disease at an early stage. We describe the most prominent supervised machine learning algorithms (SML), their characteristics, Generalisation capacity of each method, Time complexity, Hyper-parameter tuning, and Advantages and disadvantages of each technique comparatively.

The Kidney Disease dataset obtained from Kaggle was used to determine and test its highest percentage of accuracy and benchmark.

DATASET DESCRIPTIONS

The Chronic Kidney Disease Dataset consists of 24 features and one target variable.

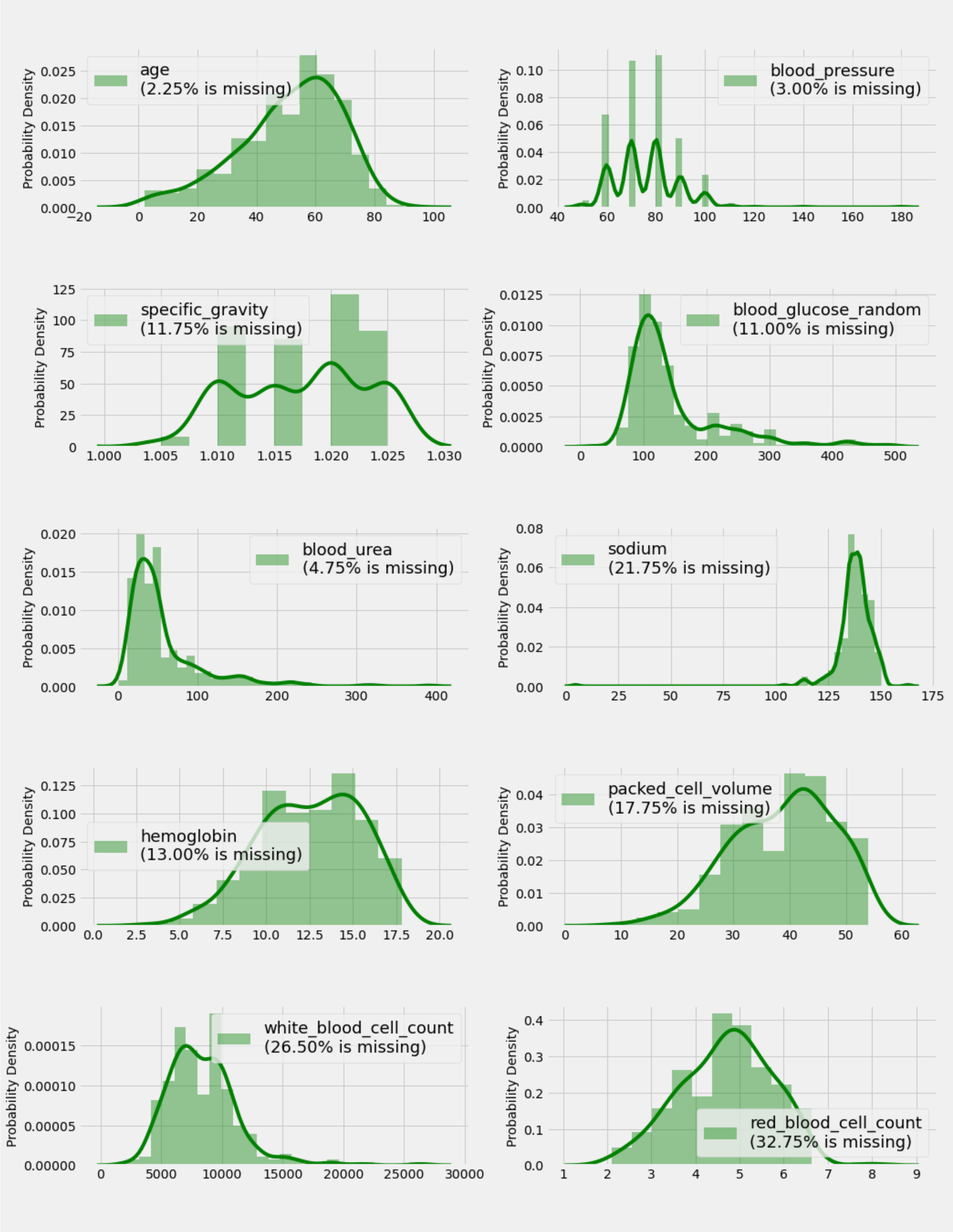
- It is a binary classification problem.
- The numerical features include:

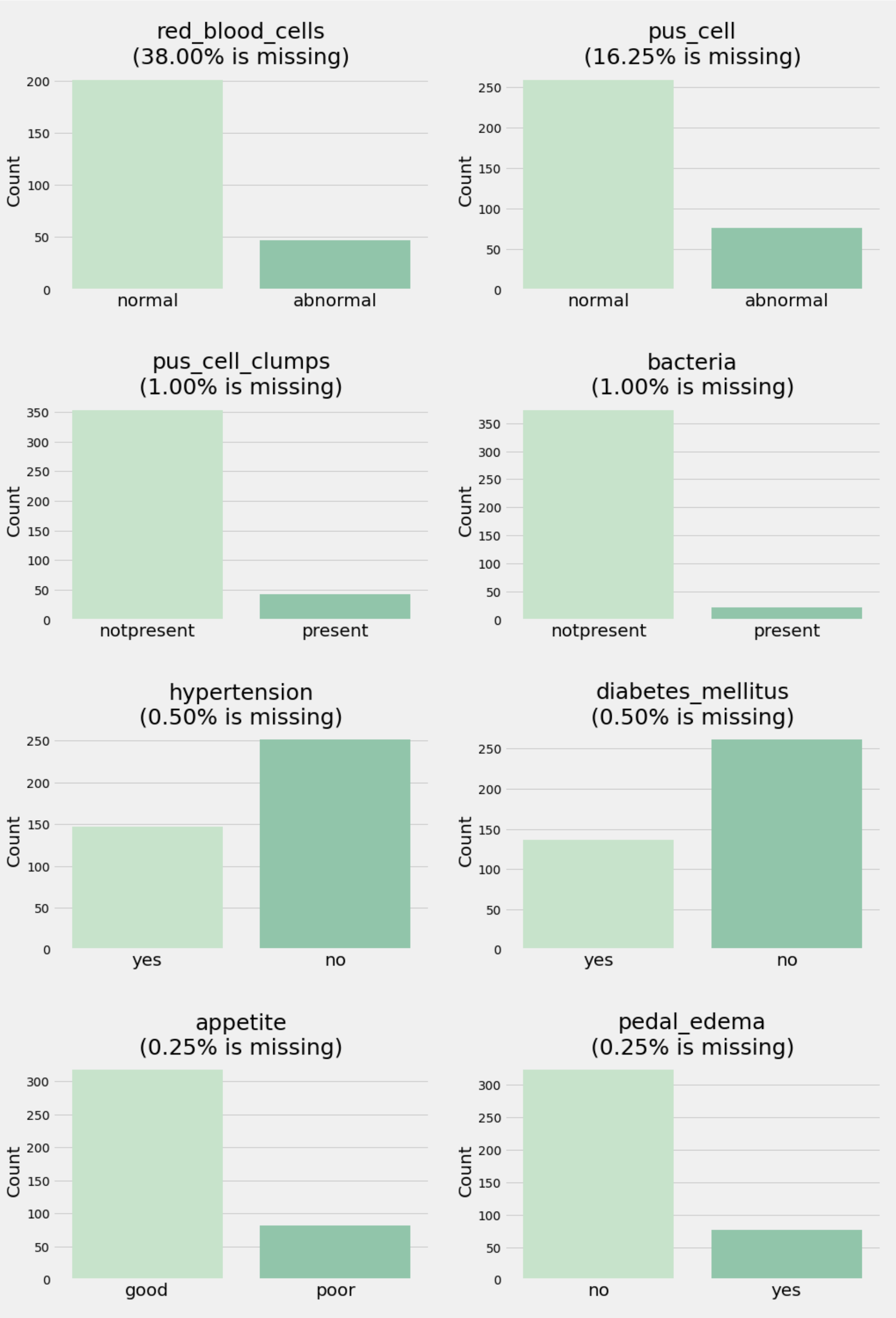
Blood Glucose Random(numerical) bgr in mgs/dl
Blood Urea(numerical) bu in mgs/dl
Serum Creatinine(numerical) sc in mgs/dl
Sodium(numerical) sod in mEq/L
Potassium(numerical) pot in mEq/L
Hemoglobin(numerical) hemo in gms
Packed Cell Volume(numerical)
White Blood Cell Count(numerical) wc in cells/cumm
Red Blood Cell Count(numerical) rc in millions/cmm
Blood Pressure(numerical) bp in mm/Hg
Specific Gravity(nominal) sg - (1.015,1.020,1.025)
Albumin(nominal)al - (0,1,2,3,4,5)
Sugar(nominal) su - (0,1,2,3,4,5)

- The categorical features include:

Red Blood Cells(nominal) rbc - (normal,abnormal)
Pus Cell (nominal)pc - (normal,abnormal)
Pus Cell clumps(nominal)pcc - (present,notpresent)
Bacteria(nominal) ba - (present,notpresent)
Hypertension(nominal) htn - (yes,no)
Diabetes Mellitus(nominal) dm - (yes,no)
Coronary Artery Disease(nominal) cad - (yes,no)
Appetite(nominal) ppet - (good,poor)
Pedal Edema(nominal) pe - (yes,no)
Anemia(nominal)ane - (yes,no)
Classification (nominal) class - (ckd,notckd)

OBSERVING THE SUMMARISED INFORMATION OF DATA





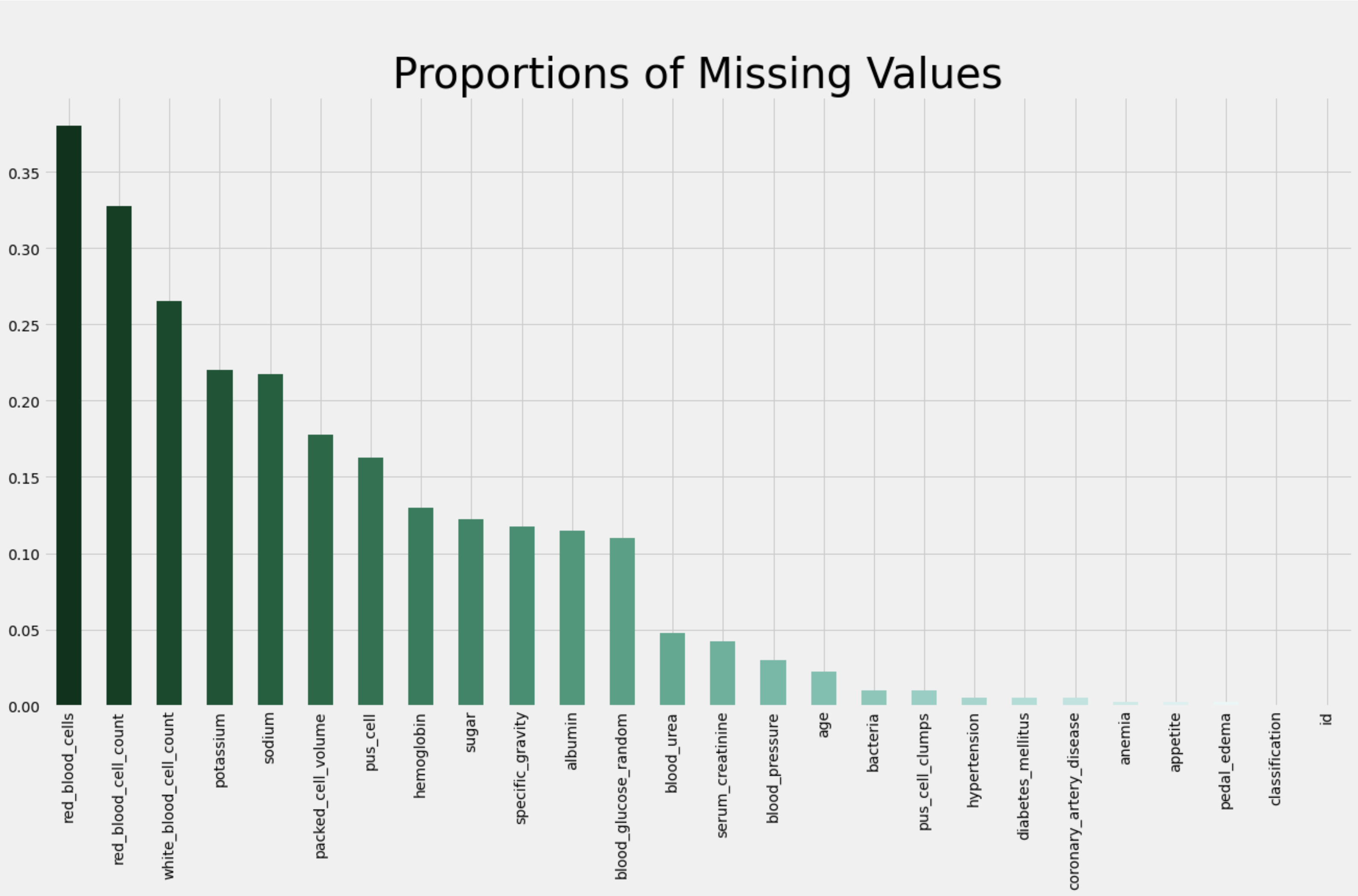
DATASET INTERPRETATION

MISSING VALUE AND IMPUTATION

Here, Chronic Kidney Disease data set contains both categorical and numerical features equally. Also, when we check for distribution using respective python graph namely Histogram and Bar plot.

Numeric features:- While diving into detailed EDA we could see that some of the features represent good distributions, some are skewed by right & left and shows us positive and negative insights.

Categorical features:- Bar graph is used to analysis categorical values. Since it's an binary classification dataset, we could see that Most of the uniques features in categorical data we're 'good', 'poor', 'yes', 'no', 'normal', 'abnormal' and finally our classification label 'ckd' and 'notckd'.



Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor fermentum.

Data cleaning process were the key to projecting model quality and good accuracy score. In CKD dataset we can clearly view the percentage of Missing values. Which is later handled using Sklearn missing value imputer. Sklearn is one of the best library for imputation.

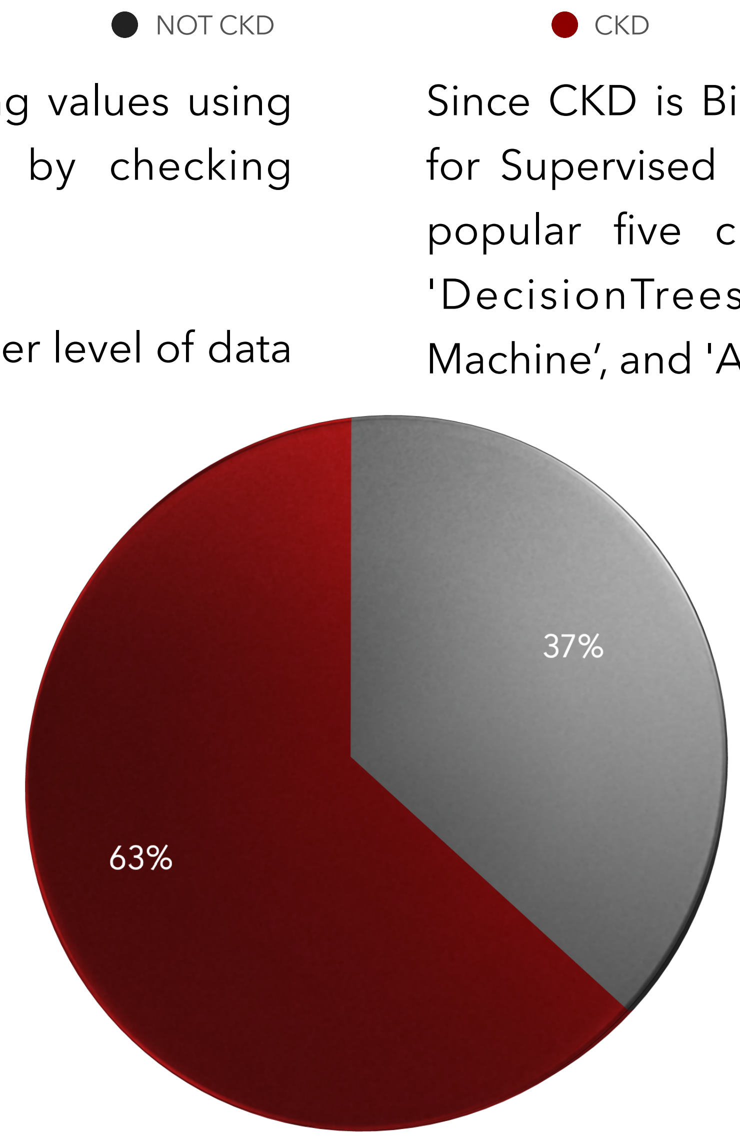
REFINED DATASET

DATA SPLITTING AND MODELLING

After required taking acting on Missing values using Sklearn package, Outliers were found by checking uniqueness values of each features.

Outliers removal:- Since we don't have higher level of data outliers. We took necessary outliers fix based on the domain knowledge by updating invalid data entry. Removing outliers will help to improve our data quality, which will be resulting in the model performance.

After taking required data cleaning and data preprocessing works our dataset were liking to be segmented by 63 percentage if CKD sample record and remaining 37 percentage of Non-CKD data sample. Followed by data cleaning and preprocessing work, Data splitting of 80:20 ratio was randomly taken for Modelling and classification process.



Since CKD is Binary classification dataset, we need to go for Supervised Machine algorithm. Here I've select most popular five classification namely 'LogisticRegression', 'DecisionTrees', 'RandomForests', 'Support Vector Machine', and 'Artificial neural network'.

Addition to Refined dataset, I gonna Implement and compare Standard Scalar transformation for finding their efficiency. Hyper-parameter tuning was applied for all the model to find the Best parameters, Best score, and Best Estimator. Also we're statically Inferring AUC and ROC curve.

Comparative details were listed in below Table-1, which contains Accuracy percentage before and after implementing PCA, AUC percentage Before and after normalising our dataset. Confusion matrix also validated for all the selected classification models.

COMPARATIVE TABLE

HYPER-PARAMETER TUNING & AUC-ROC CURVE

TABLE 1: COMPARISON OF BEST PARAMETERS, ESTIMATOR, ACCURACY, AUC VALUES

	ACCURACY BEFORE PCA	ACCURACY AFTER PCA	BEST ACCURACY	AUC BEFORE PCA	AUC AFTER PCA	BEST PARAMETERS	BEST ESTIMATOR

Logistic Regression	0.9125	1.0	0.9781	0.90	1.0	{
						'C': 1
						}
							(
							C=2275.845926074791
							)

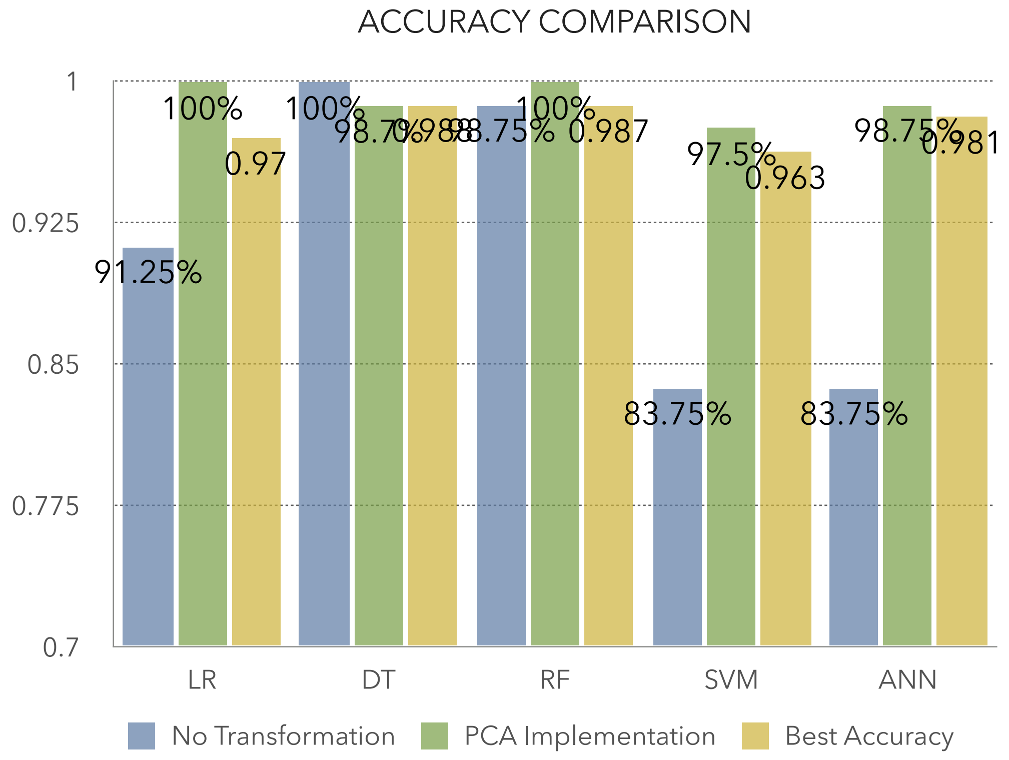
Decision Tree	1.0	0.985	0.987	1.0	0.99	{
						'min_samples_split': 20,
						'max_leaf_nodes': 128,
						'max_features': 0.4,
						'max_depth': 8,
						'criterion': 'gini',
						‘class_weight': {0: 1, 1: 3}
						}
							(
							class_weight={0: 1, 1: 3},
							max_depth=8,
							max_features=0.4,
							max_leaf_nodes=128,
							min_samples_split=20
							)

Random Forest	0.9875	1.0	0.9872	0.99	1.0	{
						’n_estimators': 90,
						'min_weight_fraction_leaf': 0.2,
						'min_samples_split': 119,
						'min_samples_leaf': 46,
						'max_leaf_nodes': 46,
						'max_depth': 3
						}
							(
							max_depth=3,
							max_leaf_nodes=46,
							min_samples_leaf=46,
							min_samples_split=119,
							min_weight_fraction_leaf=0.2,
							n_estimators=90
							)

Support Vector Machine	0.8375	0.975	0.9875	0.77	0.98	{
						'C': 10,
						'gamma': 0.001,
						'kernel': ‘rbf'
						}
							(
							C=10,
							gamma=0.001
							)

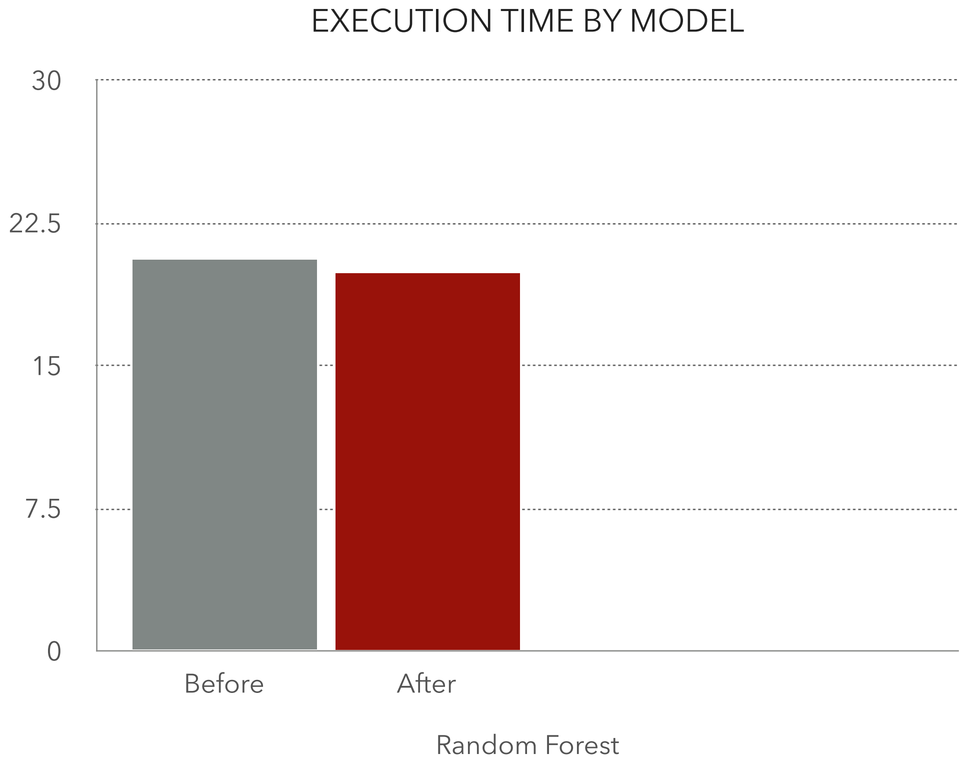
Artificial Neural Network	0.8375	0.9875	0.98125	0.88	0.99	{
						'max_iter': 1000
						}
							(
							max_iter=1000
							)

DATA SPLITTING AND MODELLING



Logistic Regression shows some improvement in accuracy score after implementing. In other hand Decision Tree showing some decrement state in accuracy score. Random Forest leads to overfitting since it already has best accuracy score. Support Vector Machine also lead to leadership board with boost up score after implementing PCA. Artificial Neural Network showing the tremendous boost in their Accuracy alike Support vector Machine. Here I'm concluding that Tree based algorithm doesn't need any Principal component Analysis for reduce the dimensions.

Excluding Artificial Neural Network (ANN) all other selected model Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM) taken comparatively less time for execution after implementing dimensionality reduction method PCA. Notably Artificial Neural Network (ANN) showing the tremendous boost in execution time alike Accuracy. Here we can assume that dimensionality reduction might affect the execution (performance) of the model which is based on Neural Network, which also boosting accuracy score.



SUPPORT VECTOR MACHINE

Advantages of SVM

- It works really well with a clear margin of separation.
- It is effective in high dimensional spaces.
- It is effective in cases where the number of dimensions is greater than the number of samples.
- It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Disadvantages of SVM

- It doesn’t perform well when we have large data set because the required training time is higher.
- It also doesn’t perform very well, when the data set has more noise i.e. target classes are overlapping.
- SVM doesn’t directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.
- It is included in the related SVC method of Python scikit-learn library.

LOGISTIC REGRESSION

Advantages

- Logistic Regression performs well when the dataset is linearly separable.
- Not only gives a measure of how relevant a predictor is, but also its direction of association (positive or negative).
- Logistic regression is easier to implement, interpret and very efficient to train.
- Less prone to over-fitting but it can overfit in high dimensional datasets. Consider Regularisation (L1 and L2) techniques to avoid over-fitting in these scenarios.

Disadvantages

- Main limitation is the assumption of linearity between the dependent variable and the independent variables. In the real world, the data is rarely linearly separable. Most of the time data would be a jumbled mess.

- If the number of observations are lesser than the number of features, It should not be used, otherwise it may lead to overfit.

- It can only be used to predict discrete functions. Therefore, the dependent variable is restricted to the discrete number set. This restriction itself is problematic, as it is prohibitive to the prediction of continuous data.

ARTIFICIAL NEURAL NETWORK

Advantages of ANN

- Neural networks are flexible and can be used for both regression and classification problems. Any data which can be made numeric can be used in the model, as neural network is a mathematical model with approximation functions.

- Neural networks are good to model with nonlinear data with large number of inputs; for example, images. It is reliable in an approach of tasks involving many features. It works by splitting the problem of classification into a layered network of simpler elements.

- Neural networks can be trained with any number of inputs and layers and work best with more data points.

Disadvantages of ANN

- Neural networks are black boxes, meaning we cannot know how much each independent variable is influencing the dependent variables.

- It is computationally very expensive and time consuming to train with traditional CPU's.

- It depend a lot on training data. This leads to the problem of over-fitting and generalisation. The mode relies more on the training data and may be tuned to the data.