# Chronic Kidney Disease Predictor

**1.) Problem statement**

Stage 1.Domain--Machine Learning

Stage 2.Learning method--Supervised

Stage 3.Classification/Regression--  Classification

**2.) Basic info about the dataset**

Total number of rows -399
columns-28

**3.) Mention the pre-processing method if you're doing any (like converting string to number – nominal data)**

We are going to use Machine learning domain. In ML algorithm we have to pass all input as number. In this problem statement we are going to convert **rbc,pc,pcc,ba,htn,dm,cad,appet,pe,ane and classification** these column string values as number-nominal data using one hot encoding.

**4.)To find following the machine learning classification  method using in confusion matrix value**

# 1.Logistic Regression:

Out[50]:

| | mean_fit_time | std_fit_time | mean_score_time | std_score_time | param_penalty | param_solver | params | split0_test_score | spli |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.012001 | 0.001633 | 0.001000 | 1.123916e-07 | l2 | newton-cg | {'penalty': 'l2', 'solver': 'newton-cg'} | 0.977654 | |
| 1 | 0.006334 | 0.001247 | 0.001333 | 4.713704e-04 | l2 | lbfgs | {'penalty': 'l2', 'solver': 'lbfgs'} | 0.977654 | |
| 2 | 0.002334 | 0.000472 | 0.001000 | 1.123916e-07 | l2 | liblinear | {'penalty': 'l2', 'solver': 'liblinear'} | 0.966561 | |
| 3 | 0.011334 | 0.000471 | 0.001667 | 4.714266e-04 | l2 | saga | {'penalty': 'l2', 'solver': 'saga'} | 0.977654 | |

Out[50]:

| param_solver | params | split0_test_score | split1_test_score | split2_test_score | mean_test_score | std_test_score | rank_test_score |
|---|---|---|---|---|---|---|---|
| newton-cg | {'penalty': 'l2', 'solver': 'newton-cg'} | 0.977654 | 0.988797 | 1.000000 | 0.988775 | 0.009114 | 1 |
| lbfgs | {'penalty': 'l2', 'solver': 'lbfgs'} | 0.977654 | 0.988797 | 1.000000 | 0.988775 | 0.009114 | 1 |
| liblinear | {'penalty': 'l2', 'solver': 'liblinear'} | 0.966561 | 0.966561 | 0.977397 | 0.970146 | 0.005098 | 4 |
| saga | {'penalty': 'l2', 'solver': 'saga'} | 0.977654 | 0.966561 | 0.988669 | 0.977587 | 0.009017 | 3 |

The report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 1.00 | 0.99 | 51 |
| 1 | 1.00 | 0.99 | 0.99 | 82 |
| accuracy | | | 0.99 | 133 |
| macro avg | 0.99 | 0.99 | 0.99 | 133 |
| weighted avg | 0.99 | 0.99 | 0.99 | 133 |

**Logistic Regression** use confusion matrix value **(Accuracy): 0.99**

# 2.Decision Tree:

Out[21]:

| | mean_fit_time | std_fit_time | mean_score_time | std_score_time | param_criterion | param_max_features | param_splitter | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.020000 | 2.121301e-02 | 0.005000 | 2.247832e-07 | gini | auto | best | {' 'max_t |
| 1 | 0.005000 | 2.247832e-07 | 0.005000 | 0.000000e+00 | gini | auto | random | {' 'max_t |
| 2 | 0.005000 | 1.946680e-07 | 0.005000 | 1.123916e-07 | gini | sqrt | best | {' 'max_t |
| 3 | 0.005000 | 1.946680e-07 | 0.001667 | 2.357077e-03 | gini | sqrt | random | {' 'max_t |
| 4 | 0.010000 | 4.082576e-03 | 0.003333 | 2.357077e-03 | gini | log2 | best | {' 'max_t |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| best | {'criterion': 'gini', 'max_features': 'auto', ... | 0.943637 | 0.921581 | 0.920683 | 0.928664 | 0.010624 | 11 |
| random | {'criterion': 'gini', 'max_features': 'auto', ... | 0.933485 | 0.922498 | 0.966172 | 0.940623 | 0.018518 | 8 |
| best | {'criterion': 'gini', 'max_features': 'sqrt', ... | 0.943986 | 0.944486 | 0.931818 | 0.940128 | 0.005846 | 9 |
| random | {'criterion': 'gini', 'max_features': 'sqrt', ... | 0.988797 | 0.922280 | 0.988669 | 0.966499 | 0.031356 | 2 |
| best | {'criterion': 'gini', 'max_features': 'log2', ... | 0.932584 | 0.933262 | 0.943001 | 0.936257 | 0.004750 | 10 |
| random | {'criterion': 'gini', 'max_features': | 0.977654 | 0.966182 | 0.977121 | 0.973639 | 0.005292 | 1 |

The report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 1.00 | 0.94 | 51 |
| 1 | 1.00 | 0.91 | 0.96 | 82 |
| accuracy | | | 0.95 | 133 |
| macro avg | 0.94 | 0.96 | 0.95 | 133 |
| weighted avg | 0.95 | 0.95 | 0.95 | 133 |

**Decision Tree Classification** use confusion matrix value **(Accuracy): 0.95**

# 3. Random Forest

Out[22]:

| | mean_fit_time | std_fit_time | mean_score_time | std_score_time | param_criterion | param_max_features | param_n_estimators |
|---|---|---|---|---|---|---|---|
| 0 | 0.024335 | 0.016781 | 0.004667 | 1.247362e-03 | gini | auto | 10 |
| 1 | 0.090339 | 0.002357 | 0.012667 | 2.054862e-03 | gini | auto | 100 |
| 2 | 0.014001 | 0.004243 | 0.002667 | 4.714266e-04 | gini | sqrt | 10 |
| 3 | 0.125007 | 0.026282 | 0.012001 | 2.160348e-03 | gini | sqrt | 100 |

| res | param_n_estimators | params | split0_test_score | split1_test_score | split2_test_score | mean_test_score | std_test_score | rank_test_score |
|---|---|---|---|---|---|---|---|---|
| uto | 10 | {'criterion': 'gini', 'max_features': 'auto', ... | 0.982143 | 0.943396 | 0.982143 | 0.969179 | 0.018282 | 11 |
| uto | 100 | {'criterion': 'gini', 'max_features': 'auto', ... | 0.982456 | 0.953271 | 0.982143 | 0.972588 | 0.013698 | 8 |
| qrt | 10 | {'criterion': 'gini', 'max_features': 'sqrt', ... | 0.982143 | 0.953271 | 0.982143 | 0.972483 | 0.013623 | 9 |
| qrt | 100 | {'criterion': 'gini', 'max_features': 'sqrt', ... | 0.982143 | 0.962963 | 0.982143 | 0.975726 | 0.009050 | 6 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| qrt | 100 | {'criterion': 'gini', 'max_features': 'sqrt', ... | 0.982143 | 0.962963 | 0.982143 | 0.975726 | 0.009050 | 6 |
| g2 | 10 | {'criterion': 'gini', 'max_features': 'log2', ... | 0.981818 | 0.933333 | 0.982143 | 0.965703 | 0.022954 | 12 |
| g2 | 100 | {'criterion': 'gini', 'max_features': 'log2', ... | 0.991150 | 0.962963 | 0.982143 | 0.978739 | 0.011775 | 1 |
| uto | 10 | {'criterion': 'entropy', 'max_features': 'auto... | 0.982143 | 0.953271 | 0.982143 | 0.972483 | 0.013623 | 9 |
| uto | 100 | {'criterion': 'entropy', 'max_features': | 0.982456 | 0.962963 | 0.982143 | 0.975830 | 0.009125 | 3 |

The report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 0.98 | 0.98 | 51 |
| 1 | 0.99 | 0.99 | 0.99 | 82 |
| accuracy | | | 0.98 | 133 |
| macro avg | 0.98 | 0.98 | 0.98 | 133 |
| weighted avg | 0.98 | 0.98 | 0.98 | 133 |

The **Random Forest** Regression use confusion matrix value **(Accuracy): 0.98**

5.) In the screenshot format, all the research values (confusion matrix value of the models) documented.

6.)Developed a good model with confusion matrix . I have used "Logistic Regression" machine learning algorithm to create final model. We have used many machine learning algorithm to test this dataset. Finally for this dataset " Logistic Regression " algorithm provided almost **0.99(99%)** accuracy.