

CKD Disease Features/Independent values

1. age
2. bp – blood pressure - 0-180
3. sg – urine specific gravity
4. al – albumin – 0 - 5
5. su – sulfonylureas(high sugar in blood)
6. rbc – red blood cells
7. pc – polycystic(puss cell) – NORMAL, ABNORMAL
8. pcc – puss cell clumps – PRESENT, NOTPRESENT
9. ba – bicarbonate(acid-base balance in blood)
- 10.bgr – blood glucose random - 0 - 490
- 11.bu – blood urea nitrogen 0 - 391
- 12.sc – serum creatine
- 13.sod – sueperoxide dismutase
- 14.pot – potassium binders
- 15.hrmo – highly resistent micro organism
- 16.pcv – low packed cell volume
- 17.wc – waist circumference a measure of abdominal obesity associated with CDK risk
- 18.rc – Renal Cell carcinoma a type of kidney cancer
- 19.htn – hypertension – NO ,YES
- 20.dm – Diabetes Mellitus - NO, YES
- 21.cad – coronary artery disease – NO , YES
- 22.appet- appetite(hunger lack) – POOR, GOOD
- 23.pe – pulmonary embolism
- 24.ane – anemia(blood lack) – NO, YES

CKD Disease Target/Dependent values

- 1) classification – NOTCKD, CKD

1. Problem Identification:


- a. Machine Learning Algorithm
- b. Supervised learning – 1. Requirement present 2. Input and output(dataset) is present
- c. Classification - Converting the categorical values to Numerical value(Regression).

2. No of rows and columns in dataset : 25 cols * 400 columns, after preprocessing 40 * 400

3. Preprocessing methods : Categorical value conversion , Standardization of values.

4. Results : Algorithm's & its Score as follows with Screenshot.

a. Random Forest Classification – Accuracy .98%.

jupyter Grid_RF_classification-assignment Last Checkpoint: 03/15/2025 (autosaved)  Log

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)


The confusion matrix:
[[52 1]
[2 77]]

```
In [20]: print("The report:\n",clf_report)
```

The report:

	precision	recall	f1-score	support
0	0.96	0.98	0.97	53
1	0.99	0.97	0.98	79
accuracy			0.98	132
macro avg	0.98	0.98	0.98	132
weighted avg	0.98	0.98	0.98	132

b. KNN-Classification: Accuracy score .74%

jupyter Grid_KNN-classification-assignment Last Checkpoint: 3 hours ago (autosaved)  Log

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
clf_report = classification_report(Y_test, Y_prediction)
```

```
In [22]: from sklearn.metrics import f1_score  
macro_value = f1_score(Y_test,Y_prediction,average='weighted')  
print("The macro_value for best parameter {}".format(grid_KNN.best_params_),macro_value)
```

The macro_value for best parameter {'algorithm': 'auto', 'weights': 'distance'}: 0.7437622983077529

```
In [23]: print(clf_report)
```

	precision	recall	f1-score	support
0	0.67	0.72	0.69	53
1	0.80	0.76	0.78	79
accuracy			0.74	132
macro avg	0.73	0.74	0.74	132
weighted avg	0.75	0.74	0.74	132

C. LogisticRegression(Classification) – roc_auc(Receiver Operating Characteristic _ Area Under the Curve) – Metrics .99% which is best model among all.

jupyter Grid_Logistic_classification-assignment Last Checkpoint: 03/15/2025 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipyk

```
In [21]: print("The confusion matrix:\n",cm)
The confusion matrix:
[[52  1]
 [ 3 76]]

In [22]: print("The report:\n",clf_report)
The report:
              precision    recall  f1-score   support

      0       0.95      0.98      0.96         53
      1       0.99      0.96      0.97         79

 accuracy      0.97
 macro avg      0.97
 weighted avg      0.97

In [23]: from sklearn.metrics import roc_auc_score
roc_auc_score(Y_test,grid_lr.predict_proba(X_test)[:,:1])

Out[23]: 0.9990446620492
```

D. SVC-Classification : Accuracy .96%

jupyter Grid_SVC_classification-assignment Last Checkpoint: 03/15/2025 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipyk

```
The confusion matrix:
[[52  1]
 [ 4 75]]

In [23]: print("The report:\n",clf_report)
The report:
              precision    recall  f1-score   support

      0       0.93      0.98      0.95         53
      1       0.99      0.95      0.97         79

 accuracy      0.96
 macro avg      0.96
 weighted avg      0.96

In [24]: from sklearn.metrics import roc_auc_score
roc_auc_score(Y_test,grid_svc.predict_proba(X_test)[:,:1])
```

E: Decision Tree : Accuracy .97%

jupyter Grid_DT_classification-assignment Last Checkpoint: a few seconds ago (autosaved)



File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3 (ip

Run

```
In [21]: print("The confusion matrix:\n",cm)
```

The confusion matrix:

```
[[51  2]
 [ 2 77]]
```

```
In [22]: print("The report:\n",clf_report)
```

The report:

	precision	recall	f1-score	support
0	0.96	0.96	0.96	53
1	0.97	0.97	0.97	79
accuracy			0.97	132
macro avg	0.97	0.97	0.97	132
weighted avg	0.97	0.97	0.97	132

F: NaiveBaye's -

BernoulliNB – Accuracy - .91%

MultiNomialNB – Accuracy - .83%

ComplementNB – Accuracy - .83%

CategoricalNB – Accuracy - .93%