1) Problem Statement Identification

To predict the Chronic Kidney Disease (CKD)

DOMAIN SELECTION:

1) Machine Learning
2) Supervised Learning
3) Classification

2) Data Set

No. of Rows: 399

No. of Columns: 28

3) Pre Processing Method

Preprocessing **Nominal Data** using **One Hot Encoding**

Standard Scaler

4) Machine Learning Algorithms

1. Support Vector Machine with Grid SearchCV
2. Decision Tree with Grid SearchCV
3. Random forest with Grid SearchCV
4. Logistic Regression with Grid SearchCV
5. K-Nearest-Neighbour with Grid SearchCV
6. Navie Bayes with Grid SearchCV

5) Comparison of Classification reports of all algorithms

1. Support vector machine Classification Report:

```
The classification 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
```

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(x_test)[:,1])
```

```
1.0
```

2. Decision Tree Classification Report:

```
              precision    recall  f1-score   support

           0       0.94      0.98      0.96        51
           1       0.99      0.96      0.98        82

    accuracy                           0.97       133
   macro avg       0.97      0.97      0.97       133
weighted avg       0.97      0.97      0.97       133
```

```
# Get probability estimates for the test set
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(x_test)[:,1])
```

```
0.9719033955045432
```

3. Random Forest Classification 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

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(x_test)[:,1])
```

```
0.9997608799617408
```

4. Logistic Regression Classification 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

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(x_test)[:,1])
```

```
1.0
```

5. **K-Nearest-Neighbour Classification Report**

	precision	recall	f1-score	support
0	0.91	1.00	0.95	51
1	1.00	0.94	0.97	82
accuracy			0.96	133
macro avg	0.96	0.97	0.96	133
weighted avg	0.97	0.96	0.96	133

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(x_test)[:,1])
```

```
0.9695121951219512
```

6. **Naive Bayes Classification Report**

	precision	recall	f1-score	support
0	0.72	0.98	0.83	51
1	0.98	0.77	0.86	82
accuracy			0.85	133
macro avg	0.85	0.87	0.85	133
weighted avg	0.88	0.85	0.85	133

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(x_test)[:,1])
```

```
0.9356767097082734
```

6) Best Model

Support Vector Machine & Logistic Regression both the algorithms gives a best model based on the classification reports (accuracy=0.99) and the roc_score(1.0) of all algorithms.