Chronic Kidney Disease Prediction

Problem Statement:

To predict the CKD based on the various parameters as per the given data.

About DataSet:

Dataset consists 25 Columns and 399 Rows

24 columns are the Input parameters before the Preprocessing

1 column is the Output and the Column name is “Classification”

If it is Yes then there is a possibility of having CKD and if it is No then there is no possibility of having CKD.

Domain :

Machine Learning (Since datasets are numbers)

Learning :

Supervised Learning

Requirements are clear

Both I/P's and O/P's are present in the dataset

It is Regression since the O/P values are continuous values.

PreProcessing

Nominal Values are converted to numerical values using One Hot Encoding and it is achieved by using Pandas get_dummies function.

The Columns are expanding from 25 columns to 28 columns

Various Algorithm Outputs

Using GridSearch method to get the best parameters for Support Vector Machin, Decision Tree and Random Forest, KNN.

Trying the Gaussian & Multinomial NaiveBayes methods

- Support Vector Machine

```
Best Parameters: {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
Test Accuracy: 0.9849624060150376
```

	precision	recall	f1-score	support
0	0.96	1.00	0.98	51
1	1.00	0.98	0.99	82
accuracy			0.98	133
macro avg	0.98	0.99	0.98	133
weighted avg	0.99	0.98	0.99	133

- Decision Tree

```
Best Parameters: {'criterion': 'entropy', 'max_depth': 20,
'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split':
2}
Test Accuracy: 0.9774436090225563
```

	precision	recall	f1-score	support
0	0.94	1.00	0.97	51
1	1.00	0.96	0.98	82
accuracy			0.98	133
macro avg	0.97	0.98	0.98	133
weighted avg	0.98	0.98	0.98	133

- Random Forest

```
Best Parameters: {'max_depth': 10, 'max_features': 'sqrt',
'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
Test Accuracy: 0.9849624060150376
```

	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

- KNN

```
Best Parameters: {'metric': 'manhattan', 'n_neighbors': 3, 'p': 1,
'weights': 'uniform'}
Best CV Accuracy: 0.9737246680642906
Test Accuracy: 0.9699248120300752
```

	precision	recall	f1-score	support
0	0.93	1.00	0.96	51
1	1.00	0.95	0.97	82
accuracy			0.97	133
macro avg	0.96	0.98	0.97	133
weighted avg	0.97	0.97	0.97	133

- NaiveBayes
  - Gaussian NB

```
Test Accuracy: 0.9774436090225563
```

	precision	recall	f1-score	support
0	0.94	1.00	0.97	51
1	1.00	0.96	0.98	82
accuracy			0.98	133
macro avg	0.97	0.98	0.98	133
weighted avg	0.98	0.98	0.98	133

  - Multinomial NB

```
Test Accuracy: 0.9849624060150376
```

	precision	recall	f1-score	support
0	0.96	1.00	0.98	51
1	1.00	0.98	0.99	82
accuracy			0.98	133
macro avg	0.98	0.99	0.98	133
weighted avg	0.99	0.98	0.99	133

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.985	0.98	0.98	0.98
Decision Tree	0.977	0.97	0.98	0.97
Random Forest	0.985	0.98	0.98	0.98
KNN	0.97	0.96	0.98	0.97
Gaussian NB	0.977	0.97	0.98	0.98
Multinomial NB	0.985	0.98	0.99	0.98

Result:

SVM, Random Forest and Multinomial Naïve Bayes giving score of 98.5%. Over all Random Forest will be best for the large datasets.