HOPE AI ASSIGNMENT - 5
Classification

GitHub Link for dataset:
https://github.com/JayachandraPrabha/Assignment-5-Classification/blob/main/CKD.csv

Problem Statement / Requirement:

A requirement from the Hospital Management asked us to create a predictive model which will predict Chronic Kidney Disease (CKD) based on several parameters. The Client has provided the dataset of the same.

1.) Identify your problem statement

2.) Tell basic info about the dataset (Total number of rows, columns)

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

4.) Develop a good model with a good evaluation metric. You can use any machine learning algorithm; you can create many models. Finally, you have to come up with a final model.

5.) All the research values of each algorithm should be documented. (You can make a tabulation or screenshot of the results.)

6.) Mention your final model, justify why you have chosen the same. .

---

1. Research values (Classification_report & roc_auc_score values of the models):

The finalized models are **Gaussian Naive Bayes(NB), Bernoulli NB and Random Forest.** After analyzing with various algorithms and tuning its hyper/tuning parameters whose roc_auc_score values were as follows:

S.No	Name of the Algorithm	roc_auc_score value	Model output
1	**Gaussian Naive Bayes (NB)**	**1.0**	**Good**
2	Multinomial NB	0.8776	Poor
3	Complement NB	0.8776	Poor
4	**Bernoulli NB**	**1.0**	**Good**
5	Support Vector Machine (SVC)	0.8631863171770662	Moderate
6	Decision Tree Classifier (DTC)	0.9733333333333334	Moderate
7	**Random Forest (RF)**	**1.0**	**Good**
8	KNN Classifier	0.8542222222222222	Moderate
9	Logistic Regression	0.9986979166666666	Moderate

Screen snips:

The best models obtained are,

1. **Gaussian Naive Bayes(NB)**

```
# from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report
clf_report=classification_report(y_test, grid_pred)
print(clf_report)
```

```
              precision    recall  f1-score   support

           0       0.97      1.00      0.98        32
           1       1.00      0.98      0.99        48

    accuracy                           0.99        80
   macro avg       0.98      0.99      0.99        80
weighted avg       0.99      0.99      0.99        80
```

```
# from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import roc_auc_score
roc_score=roc_auc_score(y_test, classifier.predict_proba(x_test)[:,1])
roc_score
```

```
1.0
```

2. **Bernoulli NB**

```
from sklearn.metrics import classification_report
clf_report=classification_report(y_test, y_pred)
print(clf_report)
```

```
              precision    recall  f1-score   support

           0       0.86      1.00      0.93        32
           1       1.00      0.90      0.95        48

    accuracy                           0.94        80
   macro avg       0.93      0.95      0.94        80
weighted avg       0.95      0.94      0.94        80
```

```
# from sklearn.naive_bayes import BernoulliNB
from sklearn.metrics import roc_auc_score
roc_score=roc_auc_score(y_test, classifier.predict_proba(x_test)[:,1])
roc_score
```

```
1.0
```

3. Random Forest

```
# from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
clf_report=classification_report(y_test, grid_pred)
print(clf_report)
```

```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        45
           1       1.00      0.99      0.99        75

    accuracy                           0.99       120
   macro avg       0.99      0.99      0.99       120
weighted avg       0.99      0.99      0.99       120
```

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

```
1.0
```

Conclusion:

The **Gaussian Naive Bayes(NB), Bernoulli NB and Random Forest** algorithms provided the roc_auc_score values are 1.0 (nearly 100% of the accuracy).
Hence **Gaussian Naive Bayes(NB), Bernoulli NB and Random Forest** machine learning classification algorithms were finalized as the best models.
