Assignment Classification

1.Problem Identification
Machine Learning ->Supervised->Classification

2.Dataset Name :CKD (Chronic kidney disease) dataset , numerical and categorical dataset.
age-numerical
bp - blood pressure
sg - specific gravity
al - albumin
su - sugar
rbc - red blood cells
pc - pus cell
pcc - pus cell clumps
ba - bacteria
bgr - blood glucose random
bu - blood urea
sc - serum creatinine
sod - sodium
pot - potassium
hemo - hemoglobin
pcv - packed cell volume
wc - white blood cell count
rc - red blood cell count
htn - hypertension
dm - diabetes mellitus
cad - coronary artery disease
appet - appetite
pe - pedal edema
ane - anemia
class - class

As a preprocessing step, I transformed categorical variables into numerical features. I utilized the **get_dummies** function, which created binary columns for each category, effectively one-hot encoding the categorical data.

Logistic Grid Classification is a simpler model compared to SVM and Random Forest Grid which may be advantageous in terms of interpretability and computational efficiency. But all three models perform similarly, choosing the simpler model might be a better choice .

jupyter 2.Logistic Grid Classification-CKD Last Checkpoint: Last Thursday

File Edit View Insert Cell Kernel Widgets Help

Run Code

In [26]:
```
print(cm)
print(clf_report)
```

```
[[51  0]
 [ 1 81]]
              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
```

jupyter CKD-DecisionTree Grid Clasification Last Checkpoint: Last Thursday a

Edit View Insert Cell Kernel Widgets Help

Run Code

In [18]:
```
print(cm)
print(clf_report)
```

```
[[50  1]
 [ 8 74]]
              precision    recall  f1-score   support

           0       0.86      0.98      0.92        51
           1       0.99      0.90      0.94        82

    accuracy                           0.93       133
   macro avg       0.92      0.94      0.93       133
weighted avg       0.94      0.93      0.93       133
```

jupyter CKD- SVM Grid Classfication Last Checkpoint: Last Thursday

Edit View Insert Cell Kernel Widgets Help

Run Code

```
print("The f1_macro value for best parameter {}:".format(g
```

```
The f1_macro value for best parameter {'C': 10, 'gamma': '
```

In [19]:

```
print(cm)
print(clf_report)
```

```
[[51  0]
 [ 1 81]]
              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
```

jupyter CKD -Naive Bayes Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

In [11]:

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

```
              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
```

```
[[51  0]
 [ 3 79]]
```

```
from sklearn.naive_bayes import BernoulliNB
classifier = BernoulliNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

```
              precision    recall  f1-score   support

           0       0.86      1.00      0.93        51
           1       1.00      0.90      0.95        82

    accuracy                           0.94       133
   macro avg       0.93      0.95      0.94       133
weighted avg       0.95      0.94      0.94       133

[[51  0]
 [ 8 74]]
```

CKD-Random Forest Grid Classification Last Checkpoint: Last Thur

View Insert Cell Kernel Widgets Help

Run Code

```
print(clf_report)
```

```
[[51  0]
 [ 1 81]]
              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
```