In [22]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, Con
import matplotlib.pyplot as plt
```

In [23]:
```
data = pd.read_csv('/content/fetal_health.csv')
```

In [24]:
```
print("Shape of the dataset:", data.shape)
```

```
Shape of the dataset: (2126, 22)
```

In [25]:
```
# check number of missing values
print(data.isnull().sum())
```

```
baseline value                                            0
accelerations                                             0
fetal_movement                                            0
uterine_contractions                                      0
light_decelerations                                       0
severe_decelerations                                      0
prolongued_decelerations                                  0
abnormal_short_term_variability                           0
mean_value_of_short_term_variability                      0
percentage_of_time_with_abnormal_long_term_variability    0
mean_value_of_long_term_variability                       0
histogram_width                                           0
histogram_min                                             0
histogram_max                                             0
histogram_number_of_peaks                                 0
histogram_number_of_zeroes                                0
histogram_mode                                            0
histogram_mean                                            0
histogram_median                                          0
histogram_variance                                        0
histogram_tendency                                        0
fetal_health                                              0
dtype: int64
```

In [26]:
```
# Split the dataset into features and target variable
x = data.drop(columns=['fetal_health'])
y = data['fetal_health']
```

In [27]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42
```

In [28]:
```
# standardization the features
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

In [29]:
```
# Training of the KNN model
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train, y_train)
```

Out[29]:
```
KNeighborsClassifier(n_neighbors=3)
```

In [30]:
```
# Model predictions with the trained model
```

```
y_pred = knn.predict(x_test)
```

In [31]:
```
# Calculate the accuracy of the KNN classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Calculate precision, recall, and F1-score for KNN model classifier
print("\nKNN Classification Report:")
print(classification_report(y_test, y_pred, target_names=['Normal', 'Suspect', 'Patholog
```

```
Accuracy: 0.92018779342723

KNN Classification Report:
              precision    recall  f1-score   support

      Normal       0.94      0.97      0.96       333
     Suspect       0.81      0.69      0.75        64
Pathological       0.84      0.90      0.87        29

    accuracy                           0.92       426
   macro avg       0.87      0.85      0.86       426
weighted avg       0.92      0.92      0.92       426
```

In [32]:
```
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Normal', 'Suspect',
disp.plot(cmap=plt.cm.Greens)
plt.show()
```