```python
# Required libraries
import numpy as np
import pandas as pd
from scipy.stats import mode
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
, cross_val_score
from sklearn.svm import SVC
from sklearn.model_selection import KFold
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score,
confusion_matrix

#Reading Training Data
data_train = pd.read_csv("../Dataset/Training.csv")

#Converting Prognosis Object part into numerical form
 using sklearn LabelEncoder()
le = LabelEncoder()
detected = le.fit_transform(data_train["prognosis"])

#Splitting Data For Training And Testing
X = data_train.iloc[:, :132]
y = data_train.iloc[:, 132]
X_train, X_test, y_train, y_test = train_test_split(X
, y, test_size=0.2, random_state=42)

#Implementing K-Fold Cross Validation, K=12
model_set = {
    "SVC": SVC(),
    "Gaussian NB": GaussianNB(),
    "Random Forest": RandomForestClassifier(
random_state=16)
}
for i in model_set:
    current_model = model_set[i]
    scores = cross_val_score(current_model, X, y, cv=
12, n_jobs=-1)
```

```
36      print(f"Model In Use: {current_model}")
37      print(f"Model Score: {scores}")
38      print(f"Mean Of Scores: {np.mean(scores)}")
39
40  #Training Using SVM Algorithm
41  model_SVC = SVC()
42  model_SVC.fit(X_train, y_train)
43  result_svc = model_SVC.predict(X_test)
44  print(f"Accuracy % Of SVC Model: {accuracy_score(
    y_test, result_svc)*100}")
45  matrix = confusion_matrix(y_test, result_svc)
46  plt.figure(figsize=(10,8))
47  sb.heatmap(matrix, annot=True)
48  plt.title("Confusion Matrix for SVM Model - Test Data
     Part")
49  plt.show()
50
51  #Training The Model Using Naive Bayes Algorithm
52  gnb = GaussianNB()
53  result_NB = gnb.fit(X_train, y_train).predict(X_test)
54  print(f"Accuracy % Of Naive Bayes Model: {
    accuracy_score(y_test, result_NB)*100}")
55  matrix = confusion_matrix(y_test, result_NB)
56  plt.figure(figsize=(10,8))
57  sb.heatmap(matrix, annot=True)
58  plt.title("Confusion Matrix for Naive Bayes Model -
    Test Data Part")
59  plt.show()
60
61  #Training The Model Using RandomForestClassifier -
    Decision Tree Algorithm
62  clf = RandomForestClassifier(n_estimators=100,
    random_state=16)
63  clf.fit(X_train, y_train)
64  result_RFC = clf.predict(X_test)
65  print(f"Accuracy % Of RandomForestClassifier Model: {
    accuracy_score(y_test, result_RFC)*100}")
66  matrix = confusion_matrix(y_test, result_RFC)
67  plt.figure(figsize=(10,8))
68  sb.heatmap(matrix, annot=True)
69  plt.title("Confusion Matrix for
```

```
RandomForestClassifier Model - Test Data Part")
plt.show()

```