```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"])

print(detected)

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)


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=
```

```
36 12, n_jobs = -1)
37     print(f"Model In Use: {current_model}")
38     print(f"Model Score: {scores}")
39     print(f"Mean Of Scores: {np.mean(scores)}")
40
41
42
43 #Training Using SVM Algorithm
44 model_SVC = SVC()
45 model_SVC.fit(X_train, y_train)
46 result_svc = model_SVC.predict(X_test)
47
48 #Training The Model Using Naive Bayes Algorithm
49 gnb = GaussianNB()
50 result_NB = gnb.fit(X_train, y_train).predict(X_test)
51
52 #Training The Model Using RandomForestClassifier -
   Decision Tree Algorithm
53 clf=RandomForestClassifier(n_estimators=100,
   random_state=16)
54 clf.fit(X_train,y_train)
55 result_RFC = clf.predict(X_test)
```