AIM: Implementation of Bagging Algorithm: Decision Tree, Random Forest

THEORY:

1) Decision Tree:

A decision tree is a flowchart-like structure in which each internal node represents a test on a feature (e.g. whether a coin flip comes up heads or tails) , each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of features that lead to those class labels. The paths from root to leaf represent classification rules. Below diagram illustrate the basic flow of decision tree for decision making with labels (Rain(Yes), No Rain(No)).

Decision tree is one of the predictive modelling approaches used in statistics, data mining and machine learning.

Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a nonparametric supervised learning method used for both classification and regression tasks.

Tree models where the target variable can take a discrete set of values are called classification trees. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

2) Random Forest:

Random forest is an ensemble machine learning algorithm.

It is perhaps the most popular and widely used machine learning algorithm given its good or excellent performance across a wide range of classification and regression predictive modeling problems.

It is also easy to use given that it has few key hyperparameters and sensible heuristics for configuring these hyperparameters.

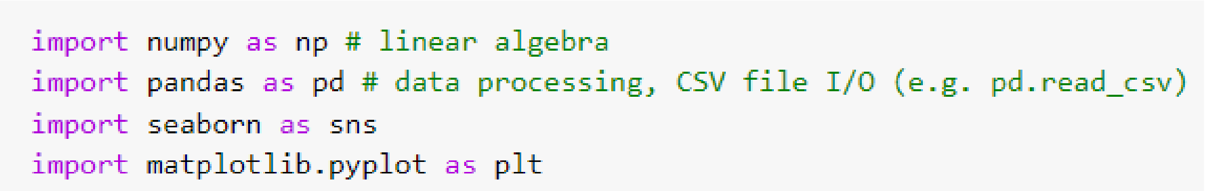
It is an extension of bootstrap aggregation (bagging) of decision trees and can be used for classification and regression problems.

In bagging, a number of decision trees are created where each tree is created from a different bootstrap sample of the training dataset. A bootstrap sample is a sample of the training dataset where a sample may appear more than once in the sample, referred to as sampling with replacement.

Bagging is an effective ensemble algorithm as each decision tree is fit on a slightly different training dataset, and in turn, has a slightly different performance. Unlike normal decision tree models, such as classification and regression trees (CART), trees used in the ensemble are unpruned, making them slightly overfit to the training dataset. This is desirable as it helps to make each tree more different and have less correlated predictions or prediction errors.

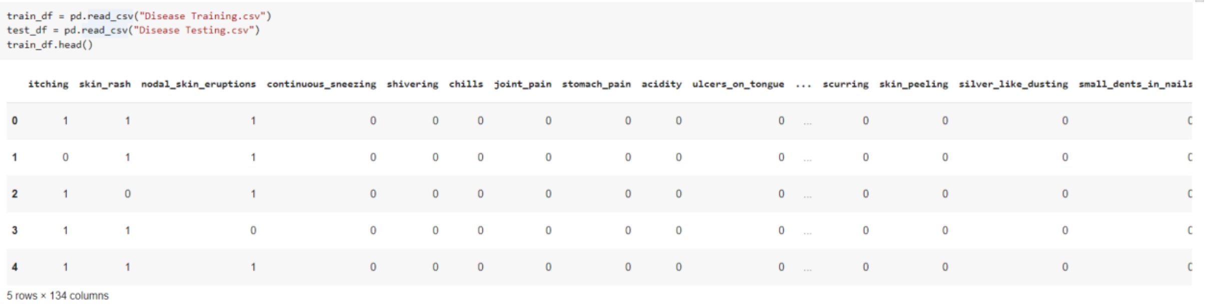
Predictions from the trees are averaged across all decision trees resulting in better performance than any single tree in the model.

1) IMPORTING LIBRARIES:



```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
```

2) READING DATASET [TRAINING & TESTING]:

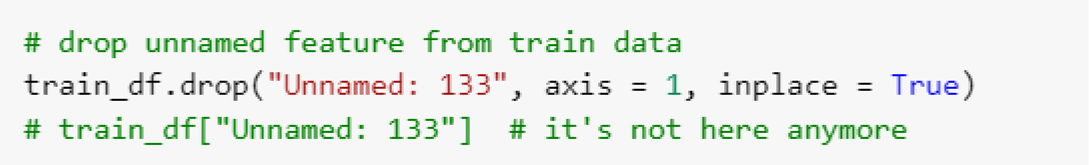


```
train_df = pd.read_csv("Disease Training.csv")
test_df = pd.read_csv("Disease Testing.csv")
train_df.head()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	scurring	skin_peeling	silver_like_dusting	small_dents_in_nails
0	1	1	1	0	0	0	0	0	0	0	...	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	...	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	...	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	...	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0	...	0	0	0	0

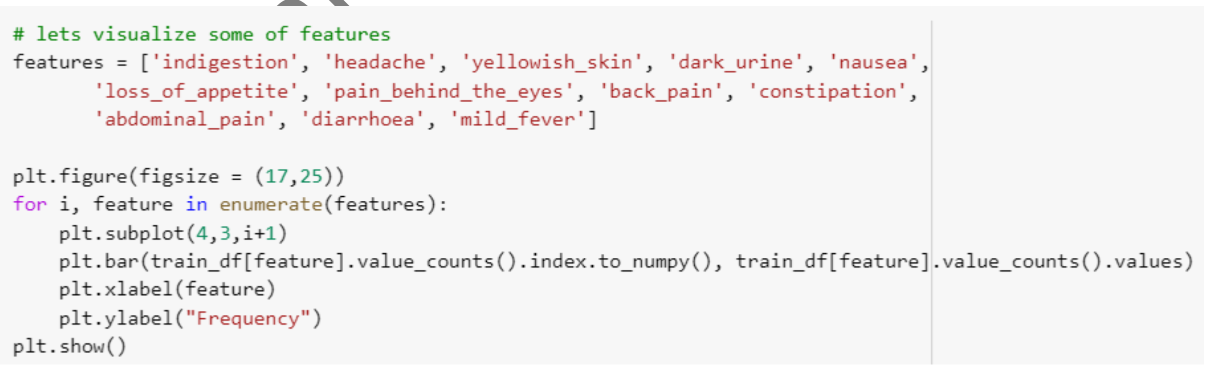
5 rows × 134 columns

3) DATA CLEANING:

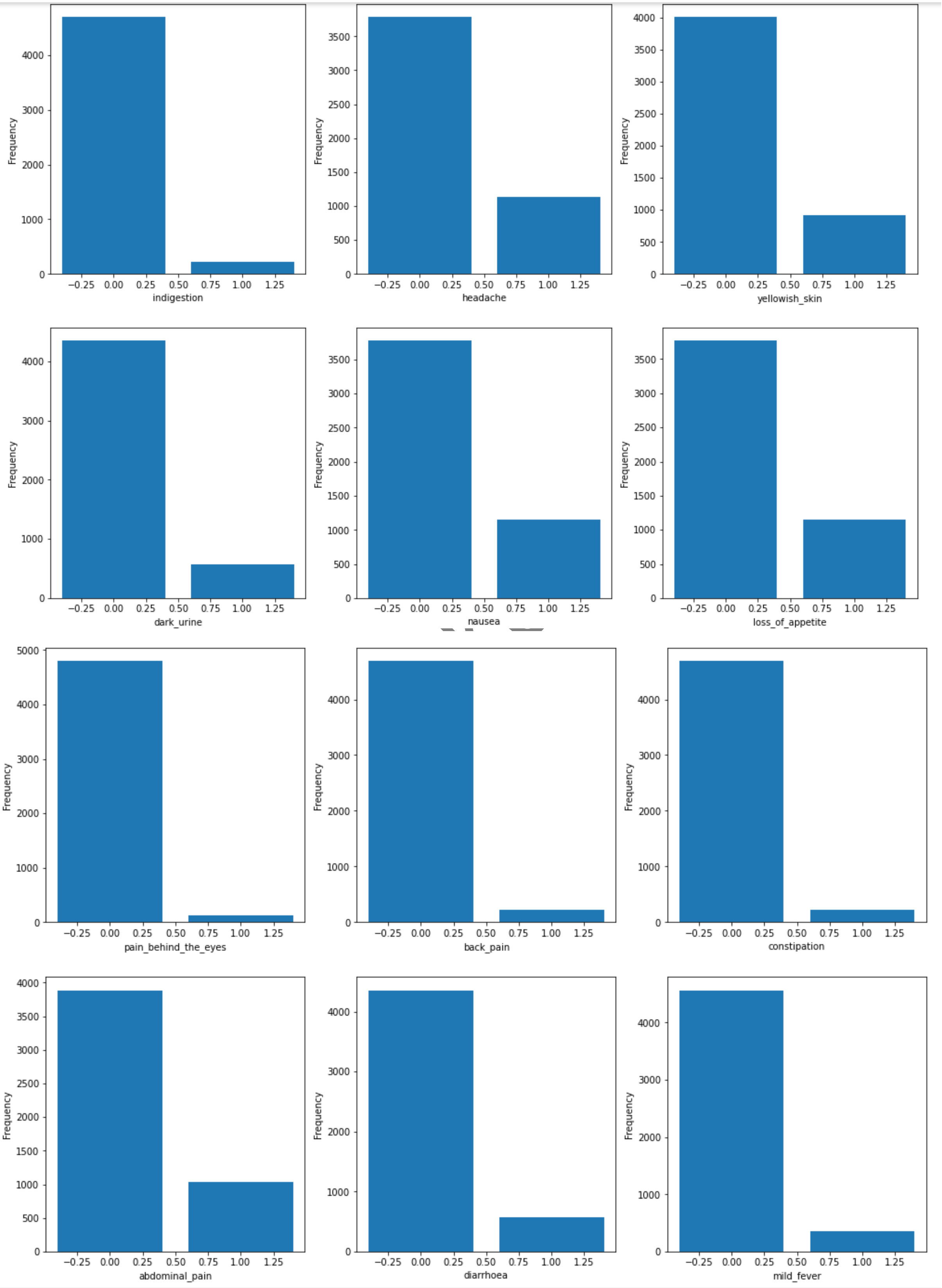


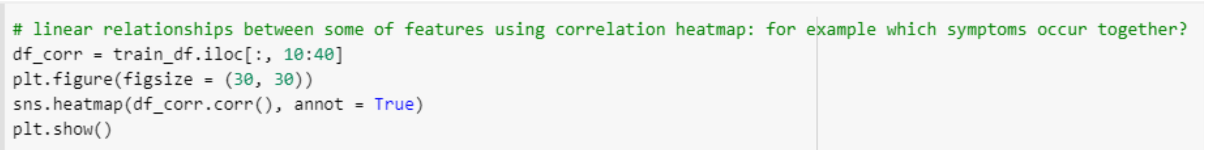
```
# drop unnamed feature from train data
train_df.drop("Unnamed: 133", axis = 1, inplace = True)
# train_df["Unnamed: 133"]  # it's not here anymore
```

4) FEATURES VISULALIZATION:

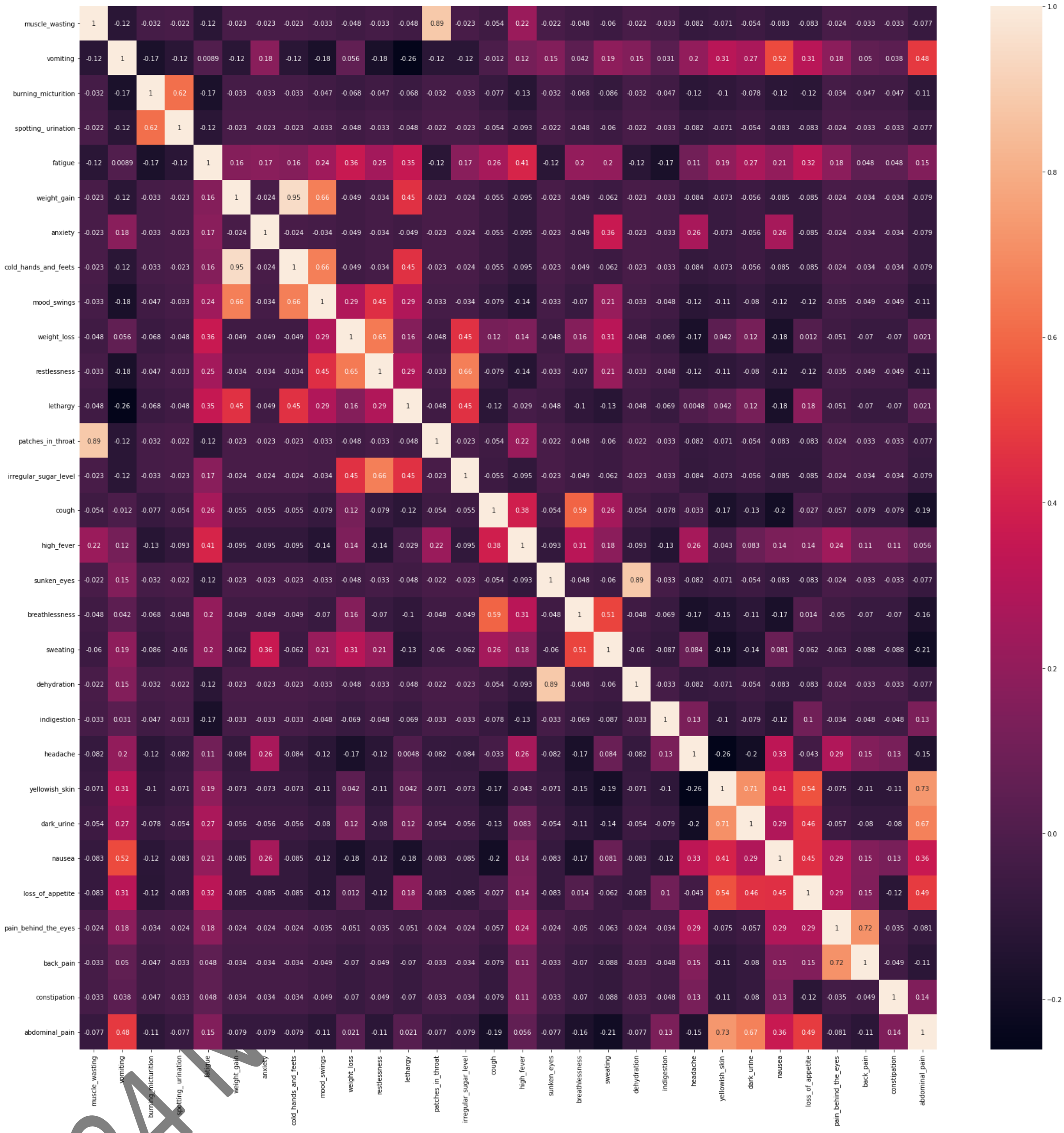


```
# lets visualize some of features
features = ['indigestion', 'headache', 'yellowish_skin', 'dark_urine', 'nausea',
       'loss_of_appetite', 'pain_behind_the_eyes', 'back_pain', 'constipation',
       'abdominal_pain', 'diarrhoea', 'mild_fever']

plt.figure(figsize = (17,25))
for i, feature in enumerate(features):
    plt.subplot(4,3,i+1)
    plt.bar(train_df[feature].value_counts().index.to_numpy(), train_df[feature].value_counts().values)
    plt.xlabel(feature)
    plt.ylabel("Frequency")
plt.show()
```

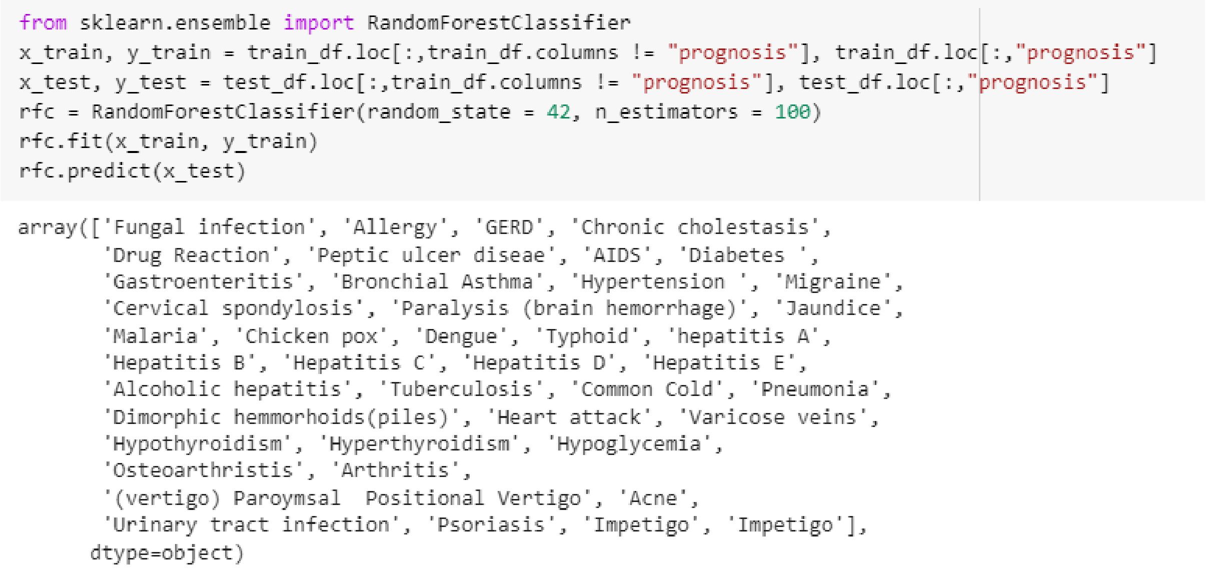




```
# linear relationships between some of features using correlation heatmap: for example which symptoms occur together?
df_corr = train_df.iloc[:, 10:40]
plt.figure(figsize = (30, 30))
sns.heatmap(df_corr.corr(), annot = True)
plt.show()
```


muscle_wasting 1 -0.12 -0.032 -0.022 -0.12 -0.023 -0.023 -0.023 -0.033 -0.048 -0.033 -0.048 0.89 -0.023 -0.054 0.22 -0.022 -0.048 -0.06 -0.022 -0.033 -0.082 -0.071 -0.054 -0.083 -0.083 -0.024 -0.033 -0.033 -0.077
vomiting -0.12 1 -0.17 -0.12 0.0089 -0.12 0.18 -0.12 -0.18 0.056 -0.18 -0.26 -0.12 -0.12 -0.012 0.12 0.15 0.042 0.19 0.15 0.031 0.2 0.31 0.27 0.52 0.31 0.18 0.05 0.038 0.48
burning_micturition -0.032 -0.17 1 0.62 -0.17 -0.033 -0.033 -0.033 -0.047 -0.068 -0.047 -0.068 -0.032 -0.033 -0.077 -0.13 -0.032 -0.068 -0.086 -0.032 -0.047 -0.12 -0.1 -0.078 -0.12 -0.12 -0.034 -0.047 -0.047 -0.11
spotting_ urination -0.022 -0.12 0.62 1 -0.12 -0.023 -0.023 -0.023 -0.033 -0.048 -0.033 -0.048 -0.022 -0.023 -0.054 -0.093 -0.022 -0.048 -0.06 -0.022 -0.033 -0.082 -0.071 -0.054 -0.083 -0.083 -0.024 -0.033 -0.033 -0.077
fatigue -0.12 0.0089 -0.17 -0.12 1 0.16 0.17 0.16 0.24 0.36 0.25 0.35 -0.12 0.17 0.26 0.41 -0.12 0.2 0.2 -0.12 -0.17 0.11 0.19 0.27 0.21 0.32 0.18 0.048 0.048 0.15
weight_gain -0.023 -0.12 -0.033 -0.023 0.16 1 -0.024 0.95 0.66 -0.049 -0.034 0.45 -0.023 -0.024 -0.055 -0.095 -0.023 -0.049 -0.062 -0.023 -0.033 -0.084 -0.073 -0.056 -0.085 -0.085 -0.024 -0.034 -0.034 -0.079
anxiety -0.023 0.18 -0.033 -0.023 0.17 -0.024 1 -0.024 -0.034 -0.049 -0.034 -0.049 -0.023 -0.024 -0.055 -0.095 -0.023 -0.049 0.36 -0.023 -0.033 0.26 -0.073 -0.056 0.26 -0.085 -0.024 -0.034 -0.034 -0.079
cold_hands_and_feets -0.023 -0.12 -0.033 -0.023 0.16 0.95 -0.024 1 0.66 -0.049 -0.034 0.45 -0.023 -0.024 -0.055 -0.095 -0.023 -0.049 -0.062 -0.023 -0.033 -0.084 -0.073 -0.056 -0.085 -0.085 -0.024 -0.034 -0.034 -0.079
mood_swings -0.033 -0.18 -0.047 -0.033 0.24 0.66 -0.034 0.66 1 0.29 0.45 0.29 -0.033 -0.034 -0.079 -0.14 -0.033 -0.07 0.21 -0.033 -0.048 -0.12 -0.11 -0.08 -0.12 -0.12 -0.035 -0.049 -0.049 -0.11
weight_loss -0.048 0.056 -0.068 -0.048 0.36 -0.049 -0.049 -0.049 0.29 1 0.65 0.16 -0.048 0.45 0.12 0.14 -0.048 0.16 0.31 -0.048 -0.069 -0.17 0.042 0.12 -0.18 0.012 -0.051 -0.07 -0.07 0.021
restlessness -0.033 -0.18 -0.047 -0.033 0.25 -0.034 -0.034 -0.034 0.45 0.65 1 0.29 -0.033 0.66 -0.079 -0.14 -0.033 -0.07 0.21 -0.033 -0.048 -0.12 -0.11 -0.08 -0.12 -0.12 -0.035 -0.049 -0.049 -0.11
lethargy -0.048 -0.26 -0.068 -0.048 0.35 0.45 -0.049 0.45 0.29 0.16 0.29 1 -0.048 0.45 -0.12 -0.029 -0.048 -0.1 -0.13 -0.048 -0.069 0.0048 0.042 0.12 -0.18 0.18 -0.051 -0.07 -0.07 0.021
patches_in_throat 0.89 -0.12 -0.032 -0.022 -0.12 -0.023 -0.023 -0.023 -0.033 -0.048 -0.033 -0.048 1 -0.023 -0.054 0.22 -0.022 -0.048 -0.06 -0.022 -0.033 -0.082 -0.071 -0.054 -0.083 -0.083 -0.024 -0.033 -0.033 -0.077
irregular_sugar_level -0.023 -0.12 -0.033 -0.023 0.17 -0.024 -0.024 -0.024 -0.034 0.45 0.66 0.45 -0.023 1 -0.055 -0.095 -0.023 -0.049 -0.062 -0.023 -0.033 -0.084 -0.073 -0.056 -0.085 -0.085 -0.024 -0.034 -0.034 -0.079
cough -0.054 -0.012 -0.077 -0.054 0.26 -0.055 -0.055 -0.055 -0.079 0.12 -0.079 -0.12 -0.054 -0.055 1 0.38 -0.054 0.59 0.26 -0.054 -0.078 -0.033 -0.17 -0.13 -0.2 -0.027 -0.057 -0.079 -0.079 -0.19
high_fever 0.22 0.12 -0.13 -0.093 0.41 -0.095 -0.095 -0.095 -0.14 0.14 -0.14 -0.029 0.22 -0.095 0.38 1 -0.093 0.31 0.18 -0.093 -0.13 0.26 -0.043 0.083 0.14 0.14 0.24 0.11 0.11 0.056
sunken_eyes -0.022 0.15 -0.032 -0.022 -0.12 -0.023 -0.023 -0.023 -0.033 -0.048 -0.033 -0.048 -0.022 -0.023 -0.054 -0.093 1 -0.048 -0.06 0.89 -0.033 -0.082 -0.071 -0.054 -0.083 -0.083 -0.024 -0.033 -0.033 -0.077
breathlessness -0.048 0.042 -0.068 -0.048 0.2 -0.049 -0.049 -0.049 -0.07 0.16 -0.07 -0.1 -0.048 -0.049 0.59 0.31 -0.048 1 0.51 -0.048 -0.069 -0.17 -0.15 -0.11 -0.17 0.014 -0.05 -0.07 -0.07 -0.16
sweating -0.06 0.19 -0.086 -0.06 0.2 -0.062 0.36 -0.062 0.21 0.31 0.21 -0.13 -0.06 -0.062 0.26 0.18 -0.06 0.51 1 -0.06 -0.087 0.084 -0.19 -0.14 0.081 -0.062 -0.063 -0.088 -0.088 -0.21
dehydration -0.022 0.15 -0.032 -0.022 -0.12 -0.023 -0.023 -0.023 -0.033 -0.048 -0.033 -0.048 -0.022 -0.023 -0.054 -0.093 0.89 -0.048 -0.06 1 -0.033 -0.082 -0.071 -0.054 -0.083 -0.083 -0.024 -0.033 -0.033 -0.077
indigestion -0.033 0.031 -0.047 -0.033 -0.17 -0.033 -0.033 -0.033 -0.048 -0.069 -0.048 -0.069 -0.033 -0.033 -0.078 -0.13 -0.033 -0.069 -0.087 -0.033 1 0.13 -0.1 -0.079 -0.12 0.1 -0.034 -0.048 -0.048 0.13
headache -0.082 0.2 -0.12 -0.082 0.11 -0.084 0.26 -0.084 -0.12 -0.17 -0.12 0.0048 -0.082 -0.084 -0.033 0.26 -0.082 -0.17 0.084 -0.082 0.13 1 -0.26 -0.2 0.33 -0.043 0.29 0.15 0.13 -0.15
yellowish_skin -0.071 0.31 -0.1 -0.071 0.19 -0.073 -0.073 -0.073 -0.11 0.042 -0.11 0.042 -0.071 -0.073 -0.17 -0.043 -0.071 -0.15 -0.19 -0.071 -0.1 -0.26 1 0.71 0.41 0.54 -0.075 -0.11 -0.11 0.73
dark_urine -0.054 0.27 -0.078 -0.054 0.27 -0.056 -0.056 -0.056 -0.08 0.12 -0.08 0.12 -0.054 -0.056 -0.13 0.083 -0.054 -0.11 -0.14 -0.054 -0.079 -0.2 0.71 1 0.29 0.46 -0.057 -0.08 -0.08 0.67
nausea -0.083 0.52 -0.12 -0.083 0.21 -0.085 0.26 -0.085 -0.12 -0.18 -0.12 -0.18 -0.083 -0.085 -0.2 0.14 -0.083 -0.17 0.081 -0.083 -0.12 0.33 0.41 0.29 1 0.45 0.29 0.15 0.13 0.36
loss_of_appetite -0.083 0.31 -0.12 -0.083 0.32 -0.085 -0.085 -0.085 -0.12 0.012 -0.12 0.18 -0.083 -0.085 -0.027 0.14 -0.083 0.014 -0.062 -0.083 0.1 -0.043 0.54 0.46 0.45 1 0.29 0.15 -0.12 0.49
pain_behind_the_eyes -0.024 0.18 -0.034 -0.024 0.18 -0.024 -0.024 -0.024 -0.035 -0.051 -0.035 -0.051 -0.024 -0.024 -0.057 0.24 -0.024 -0.05 -0.063 -0.024 -0.034 0.29 -0.075 -0.057 0.29 0.29 1 0.72 -0.035 -0.081
back_pain -0.033 0.05 -0.047 -0.033 0.048 -0.034 -0.034 -0.034 -0.049 -0.07 -0.049 -0.07 -0.033 -0.034 -0.079 0.11 -0.033 -0.07 -0.088 -0.033 -0.048 0.15 -0.11 -0.08 0.15 0.15 0.72 1 -0.049 -0.11
constipation -0.033 0.038 -0.047 -0.033 0.048 -0.034 -0.034 -0.034 -0.049 -0.07 -0.049 -0.07 -0.033 -0.034 -0.079 0.11 -0.033 -0.07 -0.088 -0.033 -0.048 0.13 -0.11 -0.08 0.13 -0.12 -0.035 -0.049 1 0.14
abdominal_pain -0.077 0.48 -0.11 -0.077 0.15 -0.079 -0.079 -0.079 -0.11 0.021 -0.11 0.021 -0.077 -0.079 -0.19 0.056 -0.077 -0.16 -0.21 -0.077 0.13 -0.15 0.73 0.67 0.36 0.49 -0.081 -0.11 0.14 1
muscle_wasting vomiting burning_micturition spotting_ urination fatigue weight_gain anxiety cold_hands_and_feets mood_swings weight_loss restlessness lethargy patches_in_throat irregular_sugar_level cough high_fever sunken_eyes breathlessness sweating dehydration indigestion headache yellowish_skin dark_urine nausea loss_of_appetite pain_behind_the_eyes back_pain constipation abdominal_pain
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

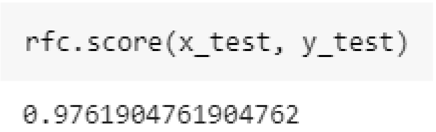
5) BUILDING RANDOM FOREST MODEL:



```
from sklearn.ensemble import RandomForestClassifier
x_train, y_train = train_df.loc[:,train_df.columns != "prognosis"], train_df.loc[:,"prognosis"]
x_test, y_test = test_df.loc[:,train_df.columns != "prognosis"], test_df.loc[:,"prognosis"]
rfc = RandomForestClassifier(random_state = 42, n_estimators = 100)
rfc.fit(x_train, y_train)
rfc.predict(x_test)
```

```
array(['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
       'Drug Reaction', 'Peptic ulcer diseae', 'AIDS', 'Diabetes ',
       'Gastroenteritis', 'Bronchial Asthma', 'Hypertension ', 'Migraine',
       'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundice',
       'Malaria', 'Chicken pox', 'Dengue', 'Typhoid', 'hepatitis A',
       'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E',
       'Alcoholic hepatitis', 'Tuberculosis', 'Common Cold', 'Pneumonia',
       'Dimorphic hemmorhoids(piles)', 'Heart attack', 'Varicose veins',
       'Hypothyroidism', 'Hyperthyroidism', 'Hypoglycemia',
       'Osteoarthristis', 'Arthritis',
       '(vertigo) Paroymsal  Positional Vertigo', 'Acne',
       'Urinary tract infection', 'Psoriasis', 'Impetigo', 'Impetigo'],
      dtype=object)
```

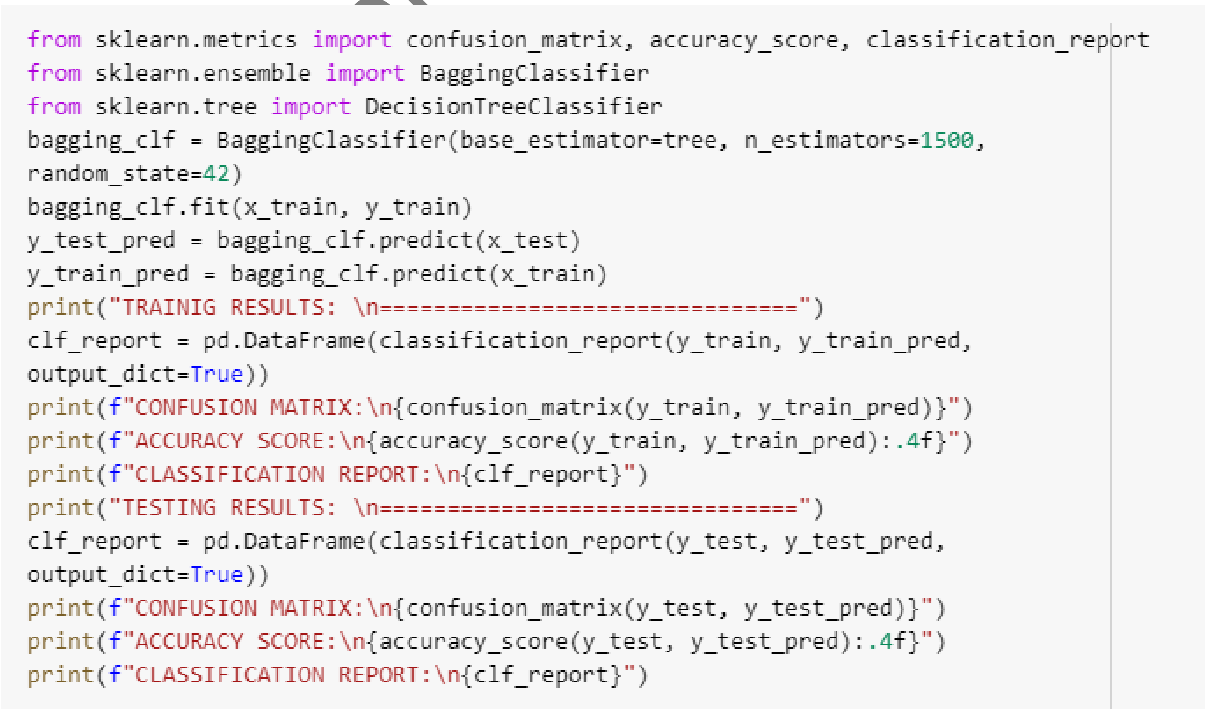
6) CHECKING RANDOM FOREST SCORE:



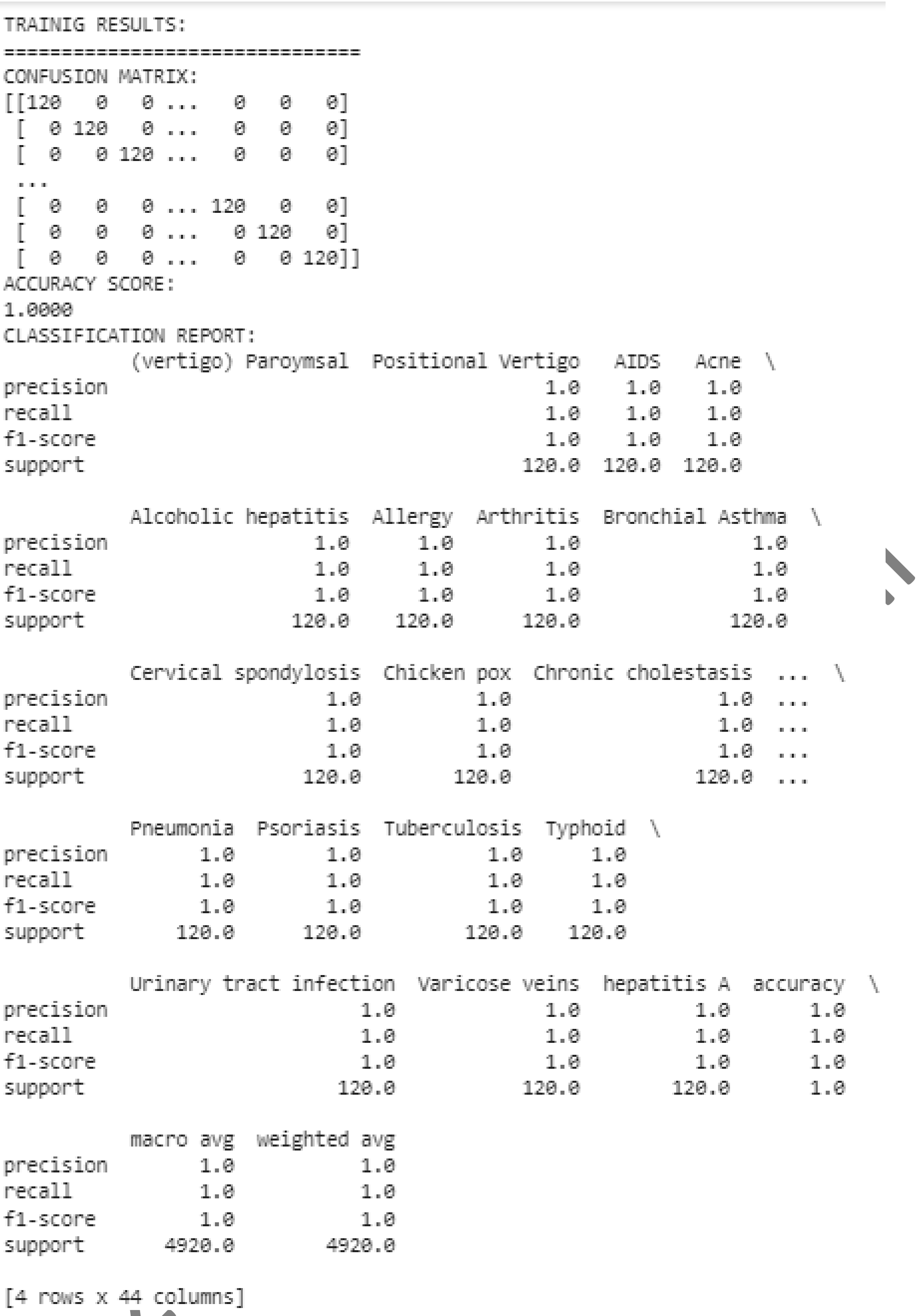
```
rfc.score(x_test, y_test)
```

```
0.9761904761904762
```

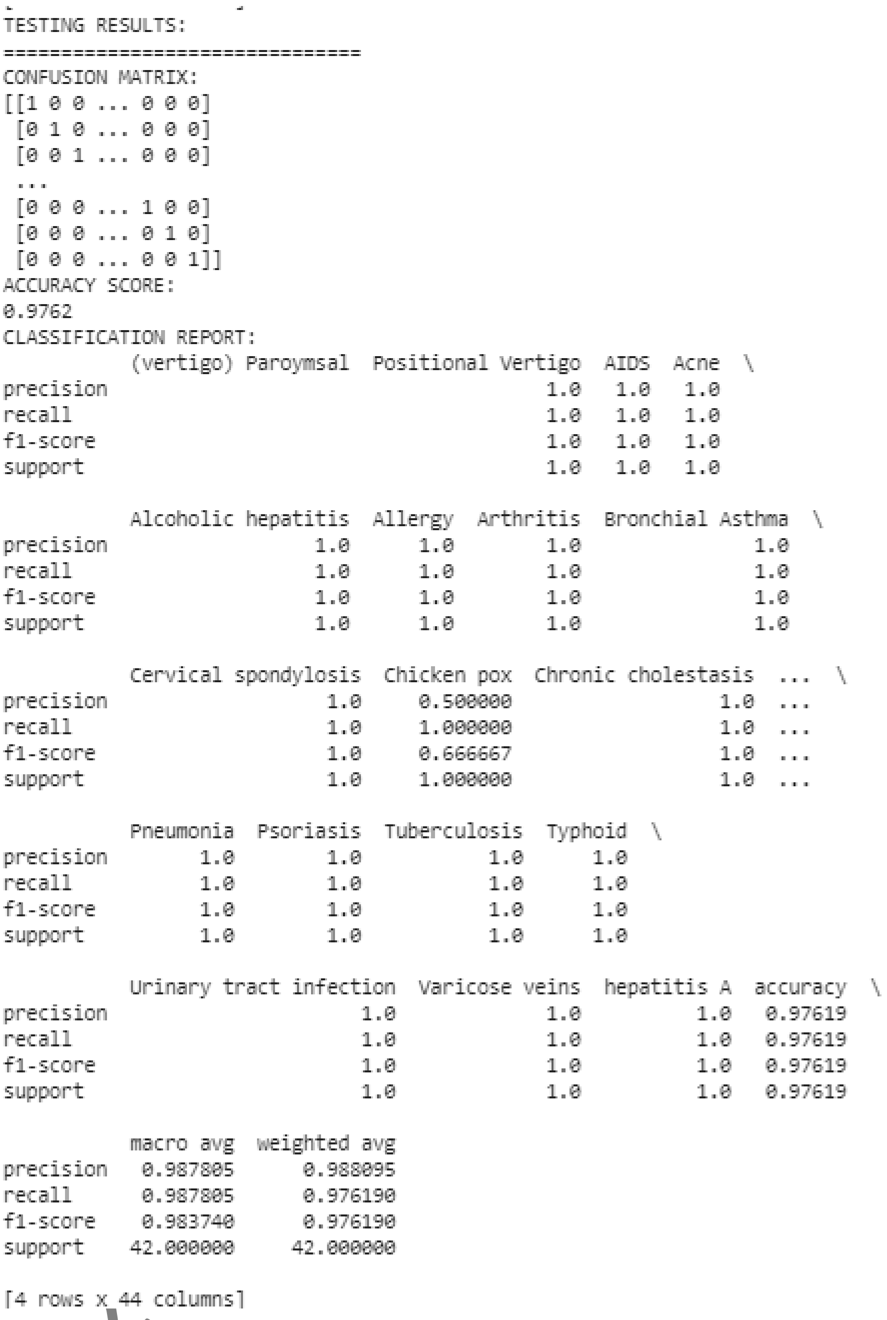
7) BUILDING BAGGING CLASSIFIER:



```
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
bagging_clf = BaggingClassifier(base_estimator=tree, n_estimators=1500,
random_state=42)
bagging_clf.fit(x_train, y_train)
y_test_pred = bagging_clf.predict(x_test)
y_train_pred = bagging_clf.predict(x_train)
print("TRAINIG RESULTS: \n===============================")
clf_report = pd.DataFrame(classification_report(y_train, y_train_pred,
output_dict=True))
print(f"CONFUSION MATRIX:\n{confusion_matrix(y_train, y_train_pred)}")
print(f"ACCURACY SCORE:\n{accuracy_score(y_train, y_train_pred):.4f}")
print(f"CLASSIFICATION REPORT:\n{clf_report}")
print("TESTING RESULTS: \n===============================")
clf_report = pd.DataFrame(classification_report(y_test, y_test_pred,
output_dict=True))
print(f"CONFUSION MATRIX:\n{confusion_matrix(y_test, y_test_pred)}")
print(f"ACCURACY SCORE:\n{accuracy_score(y_test, y_test_pred):.4f}")
print(f"CLASSIFICATION REPORT:\n{clf_report}")
```

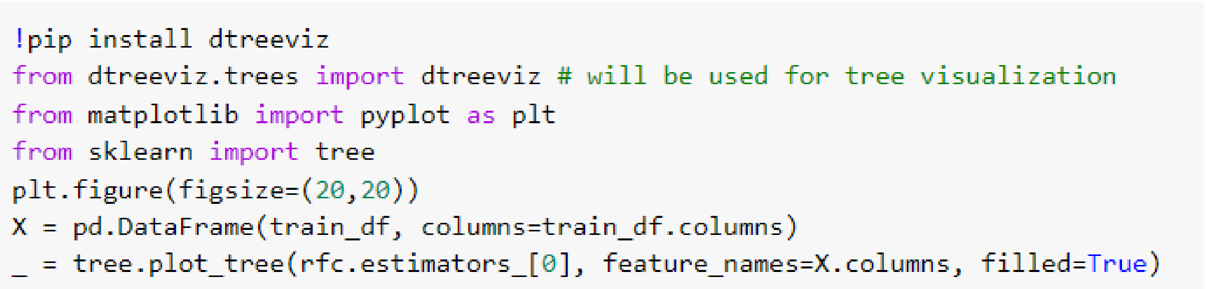


```
TRAINIG RESULTS:
===============================
CONFUSION MATRIX:
[[120   0   0 ...   0   0   0]
 [  0 120   0 ...   0   0   0]
 [  0   0 120 ...   0   0   0]
 ...
 [  0   0   0 ... 120   0   0]
 [  0   0   0 ...   0 120   0]
 [  0   0   0 ...   0   0 120]]
ACCURACY SCORE:
1.0000
CLASSIFICATION REPORT:
           (vertigo) Paroymsal  Positional Vertigo   AIDS   Acne  \
precision                                     1.0    1.0    1.0
recall                                        1.0    1.0    1.0
f1-score                                      1.0    1.0    1.0
support                                     120.0  120.0  120.0

           Alcoholic hepatitis  Allergy  Arthritis  Bronchial Asthma  \
precision                  1.0      1.0        1.0               1.0
recall                     1.0      1.0        1.0               1.0
f1-score                   1.0      1.0        1.0               1.0
support                  120.0    120.0      120.0             120.0

           Cervical spondylosis  Chicken pox  Chronic cholestasis  ...  \
precision                   1.0          1.0                  1.0  ...
recall                      1.0          1.0                  1.0  ...
f1-score                    1.0          1.0                  1.0  ...
support                   120.0        120.0                120.0  ...

           Pneumonia  Psoriasis  Tuberculosis  Typhoid  \
precision        1.0        1.0           1.0      1.0
recall           1.0        1.0           1.0      1.0
f1-score         1.0        1.0           1.0      1.0
support        120.0      120.0         120.0    120.0

           Urinary tract infection  Varicose veins  hepatitis A  accuracy  \
precision                      1.0             1.0          1.0       1.0
recall                         1.0             1.0          1.0       1.0
f1-score                       1.0             1.0          1.0       1.0
support                      120.0           120.0        120.0       1.0

           macro avg  weighted avg
precision        1.0           1.0
recall           1.0           1.0
f1-score         1.0           1.0
support       4920.0        4920.0

[4 rows x 44 columns]
```

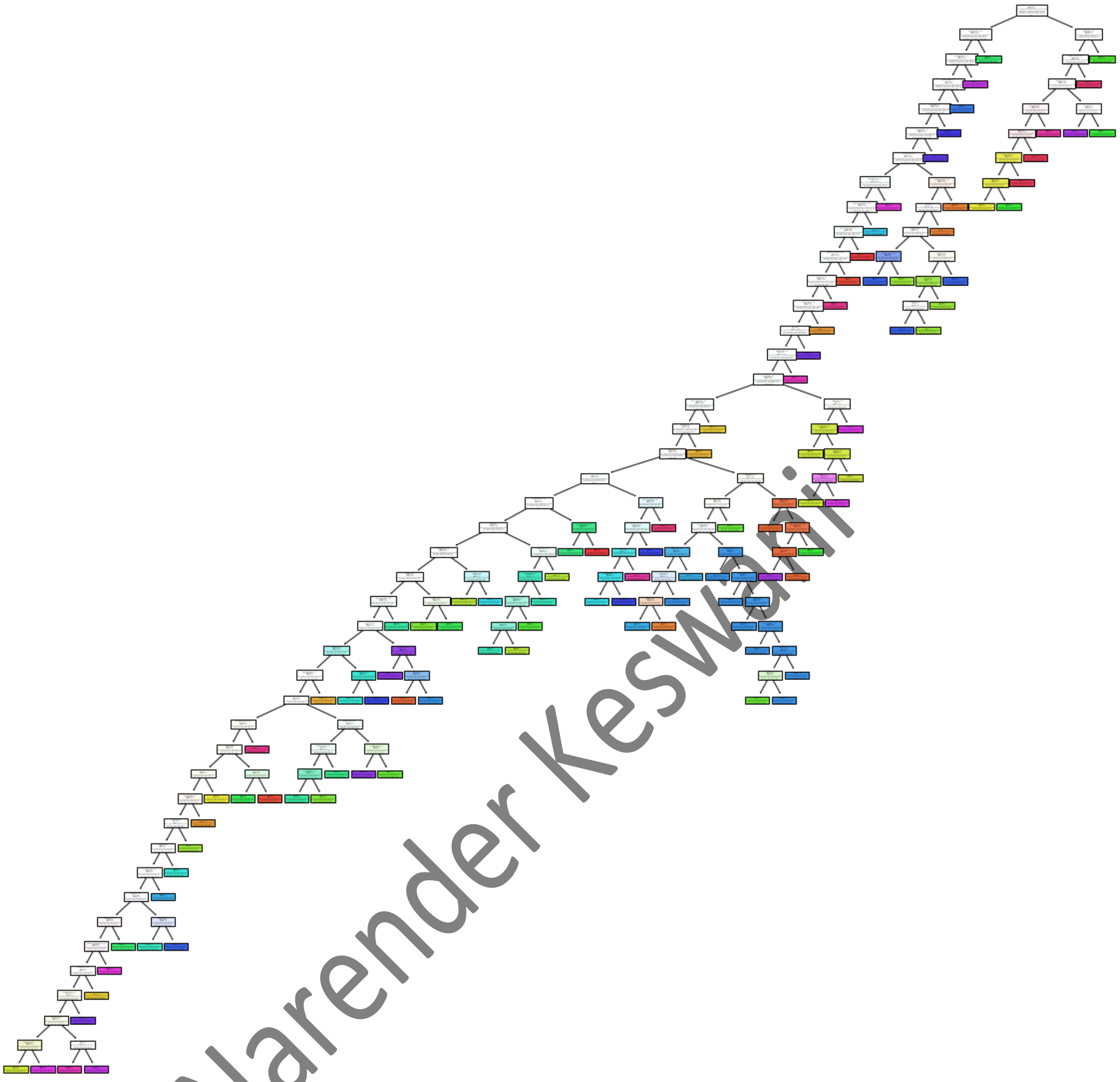


```
TESTING RESULTS:
===============================
CONFUSION MATRIX:
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
ACCURACY SCORE:
0.9762
CLASSIFICATION REPORT:
           (vertigo) Paroymsal  Positional Vertigo  AIDS  Acne  \
precision                                      1.0   1.0   1.0
recall                                         1.0   1.0   1.0
f1-score                                       1.0   1.0   1.0
support                                        1.0   1.0   1.0

           Alcoholic hepatitis  Allergy  Arthritis  Bronchial Asthma  \
precision                  1.0      1.0        1.0               1.0
recall                     1.0      1.0        1.0               1.0
f1-score                   1.0      1.0        1.0               1.0
support                    1.0      1.0        1.0               1.0

           Cervical spondylosis  Chicken pox  Chronic cholestasis  ...  \
precision                   1.0     0.500000                  1.0  ...
recall                      1.0     1.000000                  1.0  ...
f1-score                    1.0     0.666667                  1.0  ...
support                     1.0     1.000000                  1.0  ...

           Pneumonia  Psoriasis  Tuberculosis  Typhoid  \
precision        1.0        1.0           1.0      1.0
recall           1.0        1.0           1.0      1.0
f1-score         1.0        1.0           1.0      1.0
support          1.0        1.0           1.0      1.0

           Urinary tract infection  Varicose veins  hepatitis A  accuracy  \
precision                      1.0             1.0          1.0   0.97619
recall                         1.0             1.0          1.0   0.97619
f1-score                       1.0             1.0          1.0   0.97619
support                        1.0             1.0          1.0   0.97619

           macro avg  weighted avg
precision   0.987805      0.988095
recall      0.987805      0.976190
f1-score    0.983740      0.976190
support    42.000000     42.000000

[4 rows x 44 columns]
```

8) **PLOTTING TREE:**



```
!pip install dtreeviz
from dtreeviz.trees import dtreeviz # will be used for tree visualization
from matplotlib import pyplot as plt
from sklearn import tree
plt.figure(figsize=(20,20))
X = pd.DataFrame(train_df, columns=train_df.columns)
_ = tree.plot_tree(rfc.estimators_[0], feature_names=X.columns, filled=True)
```



CONCLUSION:

From this practical, I have learned and implemented the random forest algorithm in python.