

```
1 # Required libraries
2 import numpy as np
3 import pandas as pd
4 from scipy import stats
5 from scipy.stats import mode
6 import matplotlib.pyplot as plt
7 import seaborn as sb
8 from sklearn.preprocessing import LabelEncoder
9 from sklearn.model_selection import train_test_split
   , cross_val_score
10 from sklearn.svm import SVC
11 from sklearn.naive_bayes import GaussianNB
12 from sklearn.ensemble import RandomForestClassifier
13 from sklearn.metrics import accuracy_score,
   confusion_matrix
14
15 #Reading Training Data
16 data_train = pd.read_csv("../Dataset/Training.csv").
   dropna(axis=1)
17
18 #Converting Prognosis Object part into numerical form
   using sklearn LabelEncoder()
19 le = LabelEncoder()
20 detected = le.fit_transform(data_train["prognosis"])
21
22 #Splitting Data For Training And Testing
23 X = data_train.iloc[:, :-1]
24 y = data_train.iloc[:, -1]
25 X_train, X_test, y_train, y_test = train_test_split(X
   , y, test_size=0.2, random_state=42)
26
27
28 #Implementing K-Fold Cross Validation, K=12
29 def cv_scoring(estimator, X, y):
30     return accuracy_score(y, estimator.predict(X))
31
32
33 model_set = {
34     "SVC": SVC(),
35     "Gaussian NB": GaussianNB(),
36     "Random Forest": RandomForestClassifier(
```

```
36 random_state=16)
37 }
38 for i in model_set:
39     current_model = model_set[i]
40     scores = cross_val_score(current_model, X, y, cv=
41     12, n_jobs=-1, scoring=cv_scoring)
42
43 #Reading Test Data
44 data_test = pd.read_csv("../Dataset/Testing.csv").
45     dropna(axis=1)
46
47 #Selct Test Data
48 test_X = data_test.iloc[:, :-1]
49 test_Y = data_test.iloc[:, -1]
50
51 #Training Using SVM Algorithm
52 main_model_SVC = SVC()
53 main_model_SVC.fit(X, y)
54
55 #Training The Model Using Naive Bayes Algorithm
56 main_gnb = GaussianNB()
57 main_model_NB = main_gnb.fit(X, y)
58
59 #Training The Model Using RandomForestClassifier -
60 Decision Tree Algorithm
61 main_RFC = RandomForestClassifier(n_estimators=100,
62     random_state=16)
63 main_model_RFC = main_RFC.fit(X, y)
64
65 #Testing Model
66 main_result_svc = main_model_SVC.predict(test_X)
67 main_result_NB = main_model_NB.predict(test_X)
68 main_result_RFC = main_RFC.predict(test_X)
69 print(f"Accuracy % Of SVC Model: {accuracy_score(
70     test_Y, main_result_svc)*100}%")
71 print(f"Accuracy % Of Naive Bayes Model Model: {
72     accuracy_score(test_Y, main_result_NB)*100}%")
73 print(f"Accuracy % Of RandomForestClassifier Model: {
74     accuracy_score(test_Y, main_result_RFC)*100}%")
75
```

```

70
71
72 #####
#####
#####
73
74
75 ##GUI For App
76
77 import tkinter as tk
78 from tkinter import ttk
79
80 # create a tkinter window
81 window = tk.Tk()
82 window.geometry("720x250")
83 window.title("DigiMed-Python by Md Mahir Alam")
84
85 # define the values for the dropdown list
86 values = ["itching","skin_rash","
nodal_skin_eruptions","continuous_sneezing","
shivering,chills","joint_pain","stomach_pain","
acidity","ulcers_on_tongue","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","diarrhoea","
mild_fever","yellow_urine","yellowing_of_eyes","
acute_liver_failure","fluid_overload","
swelling_of_stomach","swelled_lymph_nodes","malaise"
,"blurred_and_distorted_vision","phlegm","
throat_irritation","redness_of_eyes","sinus_pressure"
","runny_nose","congestion","chest_pain","
weakness_in_limbs","fast_heart_rate","
pain_during_bowel_movements","pain_in_anal_region","
bloody_stool","irritation_in_anus","neck_pain","

```

```

86 dizziness", "cramps", "bruising", "obesity", "
   swollen_legs", "swollen_blood_vessels", "
   puffy_face_and_eyes", "enlarged_thyroid", "
   brittle_nails", "swollen_extremeties", "
   excessive_hunger", "extra_marital_contacts", "
   drying_and_tingling_lips", "slurred_speech", "
   knee_pain", "hip_joint_pain", "muscle_weakness", "
   stiff_neck", "swelling_joints", "movement_stiffness", "
   spinning_movements", "loss_of_balance", "unsteadiness"
   , "weakness_of_one_body_side", "loss_of_smell", "
   bladder_discomfort", "foul_smell_of", "urine", "
   continuous_feel_of_urine", "passage_of_gases", "
   internal_itching", "toxic_look_(typhos)", "depression"
   , "irritability", "muscle_pain", "altered_sensorium", "
   red_spots_over_body", "belly_pain", "
   abnormal_menstruation", "dischromic_patches", "
   watering_from_eyes", "increased_appetite", "polyuria",
   "family_history", "mucoid_sputum", "rusty_sputum", "
   lack_of_concentration", "visual_disturbances", "
   receiving_blood_transfusion", "
   receiving_unsterile_injections", "coma", "
   stomach_bleeding", "distention_of_abdomen", "
   history_of_alcohol_consumption", "fluid_overload", "
   blood_in_sputum", "prominent_veins_on_calf", "
   palpitations", "painful_walking", "pus_filled_pimples"
   , "blackheads", "scurring", "skin_peeling", "
   silver_like_dusting", "small_dents_in_nails", "
   inflammatory_nails", "blister", "red_sore_around_nose"
   , "yellow_crust_ooze"]

```

```

87
88 # create a label for the combobox
89 label = ttk.Label(window, text="Select values:")
90
91 # create a combobox
92 combobox = ttk.Combobox(window, state="readonly")
93
94 # create a Listbox widget for the dropdown list
95 listbox = tk.Listbox(window, selectmode="multiple",
   exportselection=0)
96 for value in values:
97     listbox.insert(tk.END, value)

```

```
98
99
100 # define a function to update the combobox when the
    user selects or deselects a value
101 def update_combobox():
102     # Get selected values from the Listbox widget
103     selected_values = [listbox.get(idx) for idx in
        listbox.curselection()]
104
105     # Update the combobox with the selected values
106     combobox.configure(width=40, height=7)
107     combobox.set(", ".join(selected_values))
108
109
110 # bind the update_combobox function to the Listbox
    widget
111 listbox.bind("<<ListboxSelect>>", lambda _:
    update_combobox())
112
113 # pack the label, combobox, and Listbox widget
114 label.pack(side="top", anchor="w", pady=30)
115 combobox.pack(side="top", pady=30)
116 listbox.pack(side="top")
117
118 # start the main loop
119 window.mainloop()
120
121
```