```
from google.colab import files
upload = files.upload()
Choose Files mushroom.csv
       mushroom.csv(text/csv) - 3171031 bytes, last modified: 4/18/2024 - 100% done
     Saving mushroom csv to mushroom csv
from google.colab import files
upload = files.upload()
₹
    Choose Files dataPrep.py
       dataPrep.py(text/x-python) - 468 bytes, last modified: 2/6/2025 - 100% done
     Saving dataPren nv to dataPren nv
import pandas as pd
df = pd.read_csv('mushroom.csv')
df
₹
              cap-diameter cap-shape gill-attachment gill-color stem-height stem-width stem-color
                                                                                                                 season class
                                                                                                                                   畾
        0
                                     2
                                                        2
                      1372
                                                                   10
                                                                                                           11 1.804273
                                                                           3.807467
                                                                                            1545
                                                                                                                                   ılı.
                                     2
                                                        2
        1
                      1461
                                                                   10
                                                                           3.807467
                                                                                            1557
                                                                                                            11
                                                                                                               1.804273
                                                                                                                              1
        2
                      1371
                                     2
                                                        2
                                                                   10
                                                                           3.612496
                                                                                            1566
                                                                                                            11
                                                                                                               1.804273
                                                                                                                              1
        3
                      1261
                                     6
                                                        2
                                                                    10
                                                                           3.787572
                                                                                            1566
                                                                                                               1.804273
                                                                                                                              1
        4
                      1305
                                     6
                                                        2
                                                                    10
                                                                            3.711971
                                                                                            1464
                                                                                                               0.943195
                                                                                                               0.943195
      54030
                        73
                                     5
                                                        3
                                                                     2
                                                                           0.887740
                                                                                             569
                                                                                                           12
      54031
                        82
                                     2
                                                        3
                                                                     2
                                                                           1.186164
                                                                                             490
                                                                                                           12 0.943195
                                                                                                                              1
      54032
                        82
                                     5
                                                        3
                                                                     2
                                                                           0.915593
                                                                                             584
                                                                                                           12
                                                                                                               0.888450
      54033
                        79
                                     2
                                                        3
                                                                     2
                                                                            1.034963
                                                                                             491
                                                                                                           12 0.888450
                                                                                                                              1
                                     5
                                                                     2
      54034
                                                        3
                                                                                                           12 0.888450
                        72
                                                                            1 158311
                                                                                             492
                                                                                                                              1
     54035 rows × 9 columns
 Next steps: ( Generate code with df
                                      View recommended plots
                                                                     New interactive sheet
from dataPrep import dataPreperation
df2 = dataPreperation(df)
df2
<del>_</del>
                                                                                                                                   \blacksquare
              cap-diameter cap-shape gill-attachment gill-color stem-height stem-width stem-color
                                                                                                                 season class
        11
                       642
                                     6
                                                        2
                                                                   10
                                                                           0.286062
                                                                                                           11 0.943195
                                                                                            1311
                                                                                                                                   16
        12
                       814
                                      4
                                                        2
                                                                    10
                                                                            1.189292
                                                                                            1681
                                                                                                               0.943195
                                                        2
                                                                   10
                                                                           0.548675
                                                                                                               0.888450
        13
                       550
                                     4
                                                                                            1220
                                                                                                           11
                                                                                                                              1
        14
                       606
                                     6
                                                        2
                                                                    10
                                                                           0.254230
                                                                                            1239
                                                                                                               0.943195
                                                        2
        15
                       721
                                     6
                                                                   10
                                                                           0.950553
                                                                                            1445
                                                                                                            11
                                                                                                               0.943195
                                                                                                                              1
        ...
      54030
                        73
                                     5
                                                        3
                                                                     2
                                                                           0.887740
                                                                                             569
                                                                                                           12 0.943195
      54031
                                     2
                                                                     2
                                                                                                           12 0.943195
                        82
                                                        3
                                                                           1.186164
                                                                                             490
                                                                                                                              1
      54032
                         82
                                     5
                                                        3
                                                                     2
                                                                           0.915593
                                                                                             584
                                                                                                               0.888450
                                     2
                                                                     2
      54033
                        79
                                                        3
                                                                            1.034963
                                                                                             491
                                                                                                           12
                                                                                                               0.888450
                                                                                                                              1
      54034
                                     5
                                                        3
                                                                     2
                                                                                                           12 0.888450
                        72
                                                                            1.158311
                                                                                             492
                                                                                                                              1
     44504 rows × 9 columns
 Next steps: Generate code with df2
                                       View recommended plots
                                                                      New interactive sheet
df3 = dataPreperation(df2)
df3
```

cap-diameter	cap-shape	gill-attachment	gill-color	stem-height	stem-width	stem-color	season	class	\blacksquare
642	6	2	10	0.286062	1311	11	0.943195	1	11.
814	4	2	10	1.189292	1681	11	0.943195	1	+/
550	4	2	10	0.548675	1220	11	0.888450	1	
606	6	2	10	0.254230	1239	11	0.943195	1	
721	6	2	10	0.950553	1445	11	0.943195	1	
	•••			•••		•••			
73	5	3	2	0.887740	569	12	0.943195	1	
82	2	3	2	1.186164	490	12	0.943195	1	
82	5	3	2	0.915593	584	12	0.888450	1	
79	2	3	2	1.034963	491	12	0.888450	1	
72	5	3	2	1.158311	492	12	0.888450	1	
ws × 9 columns									
Generate code	with df3	View recommen	ded plots (New interactive	sheet				
manation(dfa)									
eperacton(at3)									
can-diameter	can-shane	gill-attachment	gill-color	stem-height	stem-width	stem-color	Sazenn	class	
-									
									11
									1
	5	3	2	1.158311	492	12	0.888450	1	
ws × 9 columns									
Generate code	with df4	View recommend	ded plots (New interactive	sheet				
eperation(df4)									
cap-diameter	cap-shape	gill-attachment	gill-color	stem-height	stem-width	stem-color	season	class	
		0	40	0.000000	1311	11	0.943195	1	ıl.
642	6	2	10	0.286062					
642 814	6	2	10	1.189292	1681	11	0.943195	1	+/
							0.943195 0.888450	1	7
814	4	2	10	1.189292	1681	11			*/
814 550	4	2	10 10	1.189292 0.548675	1681 1220	11 11	0.888450	1	7
814 550 606	4 4 6	2 2 2	10 10 10	1.189292 0.548675 0.254230	1681 1220 1239	11 11	0.888450 0.943195	1	7
814 550 606 721	4 4 6 6	2 2 2 2	10 10 10 10	1.189292 0.548675 0.254230 0.950553	1681 1220 1239 1445	11 11 11 	0.888450 0.943195 0.943195	1 1 1	7
814 550 606 721	4 4 6 6	2 2 2 2 	10 10 10 10 	1.189292 0.548675 0.254230 0.950553	1681 1220 1239 1445	11 11 11 12	0.888450 0.943195 0.943195 	1 1 1 	*/
814 550 606 721 73	4 4 6 6 5	2 2 2 2 3	10 10 10 10 2	1.189292 0.548675 0.254230 0.950553 0.887740	1681 1220 1239 1445 569	11 11 11 12	0.888450 0.943195 0.943195 0.943195	1 1 1 	*/
814 550 606 721 73 82	4 4 6 6 5 2	2 2 2 2 3 3	10 10 10 10 2	1.189292 0.548675 0.254230 0.950553 0.887740 1.186164	1681 1220 1239 1445 569 490	11 11 11 12 12	0.888450 0.943195 0.943195 0.943195 0.943195	1 1 1 1	*/
814 550 606 721 73 82	4 4 6 6 5 2 5	2 2 2 3 3 3	10 10 10 10 2 2	1.189292 0.548675 0.254230 0.950553 0.887740 1.186164 0.915593	1681 1220 1239 1445 569 490 584	11 11 11 12 12 12	0.888450 0.943195 0.943195 0.943195 0.943195 0.888450	1 1 1 1 1	*/
	642 814 550 606 721 73 82 82 79 72 ws × 9 columns Generate code eperation(df3) cap-diameter 642 814 550 606 721 73 82 82 79 72 ws × 9 columns	642 6 814 4 550 4 606 6 721 6 73 5 82 2 82 5 79 2 72 5 ws × 9 columns Generate code with df3 eperation(df3) cap-diameter cap-shape 642 6 814 4 550 4 606 6 721 6 73 5 82 2 82 5 79 2 5 ws × 9 columns	642 6 2 814 4 2 550 4 2 606 6 2 721 6 2 73 5 3 82 2 3 82 5 3 79 2 3 72 5 3 ws × 9 columns Generate code with df3	642 6 2 10 814 4 2 10 550 4 2 10 606 6 2 10 721 6 2 10 73 5 3 2 82 2 3 2 82 5 3 2 79 2 3 2 72 5 3 2 ws × 9 columns Generate code with df3 © View recommended plots experation(df3) cap-diameter cap-shape gill-attachment gill-color 642 6 2 10 814 4 2 10 550 4 2 10 606 6 2 10 721 6 2 10	642 6 2 10 0.286062 814 4 2 10 1.189292 550 4 2 10 0.548675 606 6 2 10 0.254230 721 6 2 10 0.950553 73 5 3 2 0.887740 82 2 3 2 1.186164 82 5 3 2 0.915593 79 2 3 2 1.034963 72 5 3 2 1.158311 ws × 9 columns Cenerate code with df3 © View recommended plots New interactive states and the second of the se	642 6	642 6 2 10 0.286062 1311 11 814 4 2 10 1.189292 1681 11 550 4 2 10 0.548675 1220 11 606 6 6 2 10 0.254230 1239 11 721 6 2 10 0.950553 1445 11	642 6 2 10 0.286062 1311 11 0.943195 814 4 2 10 1.189292 1681 11 0.943195 550 4 2 10 0.548675 1220 111 0.888450 606 6 2 10 0.950553 1445 11 0.943195 721 6 2 10 0.950553 1445 11 0.943195 82 2 3 2 1.186164 490 12 0.943195 82 2 3 2 1.186164 490 12 0.943195 82 5 3 2 0.915593 584 12 0.888450 79 2 3 2 1.034963 491 12 0.888450 89 cap-diameter cap-shape gill-attachment gill-color stem-height stem-width stem-color season 642 6 2 10 0.286062 1311 11 0.943195 85 9 0 4 2 10 0.286062 1311 11 0.943195 864 1 0.943195 87 9 0 0.888450 89 cap-diameter cap-shape gill-attachment gill-color stem-height stem-width stem-color season 864 0 0 0 0.286062 1311 11 0.943195 89 0 0 0.888450 89 0 0 0 0.950553 1445 11 0.943195 89 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0.950553 1445 11 0.943195 80 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	642 6

df6 = dataPreperation(df5) **₹** \blacksquare cap-diameter cap-shape gill-attachment gill-color stem-height stem-width stem-color season class 11 0.943195 0.286062 th 1.189292 0.943195 0.548675 0.888450 0.254230 0.943195 0.950553 0.943195 ... 0.943195 0.887740 1.186164 0.943195 0.915593 0.888450 1.034963 0.888450 1.158311 12 0.888450 43338 rows × 9 columns New interactive sheet Next steps: (Generate code with df6) View recommended plots df7 = dataPreperation(df6) df7 ₹ cap-diameter cap-shape gill-attachment gill-color stem-height stem-width stem-color 丽 season class 0.286062 0.943195 th 1.189292 0.943195 +/ 0.548675 0.888450 0.943195 0.254230 0.943195 0.950553 ... 0.943195 0.887740 1.186164 0.943195 0.888450 0.915593 0.888450 1.034963 1.158311 12 0.888450 42787 rows × 9 columns Next steps: (Generate code with df7 View recommended plots New interactive sheet df8 = dataPreperation(df7)

	•		gill-attachment							
11	642	6	2	10	0.286062	1311	11	0.943195	1	11.
12	814	4	2	10	1.189292	1681	11	0.943195	1	1
13	550	4	2	10	0.548675	1220	11	0.888450	1	
14	606	6	2	10	0.254230	1239	11	0.943195	1	
15	721	6	2	10	0.950553	1445	11	0.943195	1	
54030	73	5	3	2	0.887740	569		0.943195	1	
54031	82	2	3	2	1.186164	490		0.943195	1	
54032	82	5	3	2	0.915593	584		0.888450	1	
54033	79	2	3	2	1.034963	491		0.888450	1	
54034 2633 ro	72 ows × 9 columns	5	3	2	1.158311	492	12	0.888450	1	
203310	W3 × 3 COIGITITIS									
steps: (Generate code	with df8	 View recommend 	ded plots	New interactive	sheet				
dataPr	eperation(df8)									
	cap-diameter	cap-shape	gill-attachment	gill-color	stem-height	stem-width	stem-color	season	class	
11	642	6	2	10	0.286062	1311	11	0.943195	1	11.
12	814	4	2	10	1.189292	1681	11	0.943195	1	+//
13	550	4	2	10	0.548675	1220	11	0.888450	1	0
14	606	6	2	10	0.254230	1239	11	0.943195	1	
15	721	6	2	10	0.950553	1445	11	0.943195	1	
54030	73	5	3	2	0.887740	569	12	0.943195	1	
54031	82	2	3	2	1.186164	490	12	0.943195	1	
54032	82	5	3	2	0.915593	584	12	0.888450	1	
54033	79	2	3	2	1.034963	491	12	0.888450	1	
54034	72	5	3	2	1.158311	492	12	0.888450	1	
	72 ows × 9 columns	5	3	2	1.158311	492	12	0.888450	1	
2598 ro	ows × 9 columns						12	0.888450	1	
	ows × 9 columns		View recommendation		1.158311 New interactive		12	0.888450	1	
2598 ro 	ows × 9 columns	with df9					12	0.888450	1	
2598 ro 	ows × 9 columns Generate code	with df9					12	0.888450	1	
2598 ro 	Generate code reperation(df9	with df9		ded plots	New interactive	sheet		0.888450 season		⊞
2598 ro 	Generate code reperation(df9	with df9	View recomment	ded plots	New interactive	sheet	stem-color			Ⅲ
2598 ro	Generate code reperation(df9	with df9	© View recommen	ded plots	New interactive	sheet stem-width	stem-color	season	class	_
2598 rosteps: (dataP	Generate code reperation(df9 cap-diameter 642	with df9 (© View recommend	ded plots gill-color 10	New interactive stem-height 0.286062	sheet stem-width 1311	stem-color 11 11	season 0.943195	class	11.
2598 ro steps: (dataP	Generate code of reperation (df9) cap-diameter 642	with df9 (2) (2) (2) (2) (2) (3) (4) (4) (4) (4) (4) (4) (4) (4) (4) (4	gill-attachment 2 2	gill-color 10 10	Stem-height 0.286062 1.189292	stem-width 1311 1681	stem-color 11 11 11	season 0.943195 0.943195	class	11.
2598 ro	Generate code reperation(df9 cap-diameter 642 814 550	with df9 (2) cap-shape 6 4 4	gill-attachment 2 2 2	gill-color 10 10	stem-height	stem-width 1311 1681 1220	stem-color 11 11 11 11	season 0.943195 0.943195 0.888450	class 1 1	11.
2598 ro	Generate code reperation(df9 cap-diameter 642 814 550 606	cap-shape 6 4 6	gill-attachment 2 2 2 2	gill-color 10 10 10	stem-height	stem-width	stem-color 11 11 11 11	season 0.943195 0.943195 0.888450 0.943195	class 1 1 1	11.
2598 ro steps: (dataP	Generate code reperation(df9 cap-diameter 642 814 550 606 721	cap-shape 6 4 6 6	gill-attachment 2 2 2 2	gill-color 10 10 10 10	stem-height	stem-width 1311 1681 1220 1239 1445	stem-color 11 11 11 11 11	season 0.943195 0.943195 0.888450 0.943195 0.943195	class 1 1 1 1	11.
2598 ro steps: (dataP 11 12 13 14 15 54030	Generate code reperation(df9 cap-diameter 642 814 550 606 721	cap-shape 6 4 6	gill-attachment 2 2 2 2 2	gill-color 10 10 10 10	stem-height	stem-width 1311 1681 1220 1239 1445	stem-color 11 11 11 11 11 11 11	season 0.943195 0.943195 0.888450 0.943195 0.943195	class 1 1 1 1	11.
2598 ro dataP 11 12 13 14 15	Generate code reperation(df9 cap-diameter 642 814 550 606 721 73	cap-shape 6 4 6 6 5	gill-attachment 2 2 2 2 2 3	gill-color 10 10 10 10 2	stem-height	stem-width 1311 1681 1220 1239 1445 569	stem-color 11 11 11 11 11 11 11 12	season 0.943195 0.943195 0.943195 0.943195 0.943195	class 1 1 1 1 1 1 1	11.
2598 ro steps: (dataPi 11 12 13 14 15 54030 54031	Generate code reperation(df9 cap-diameter 642 814 550 606 721 73	cap-shape 6 4 6 5	gill-attachment 2 2 2 2 3 3	gill-color 10 10 10 10 2 2	stem-height	stem-width 1311 1681 1220 1239 1445 569 490	stem-color 11 11 11 11 11 12 12 12	season 0.943195 0.943195 0.888450 0.943195 0.943195 0.943195	class 1 1 1 1 1 1 1 1 1 1	11.
2598 ro steps: (dataP 11 12 13 14 15 54030	Generate code reperation(df9 cap-diameter 642 814 550 606 721 73 82 82	cap-shape 6 4 6 6 5 2 5	gill-attachment 2 2 2 2 2 3 3 3 3	gill-color 10 10 10 10 2 2 2	stem-height	stem-width 1311 1681 1220 1239 1445 569 490 584	stem-color 11 11 11 11 11 11 12 12 12 12	season 0.943195 0.943195 0.943195 0.943195 0.943195 0.943195 0.943195	class 1 1 1 1 1 1 1 1 1 1 1 1	11.

```
df11 = dataPreperation(df10)
₹
             cap-diameter cap-shape gill-attachment gill-color stem-height stem-width stem-color
                                                                                                            season class
                                                                                                                             扁
       11
                      642
                                    6
                                                     2
                                                                 10
                                                                        0.286062
                                                                                         1311
                                                                                                       11 0.943195
                                                                                                                              ılı.
                                                     2
       12
                      814
                                    4
                                                                 10
                                                                        1.189292
                                                                                        1681
                                                                                                       11 0.943195
                                                                                                                         1
                                                                                                                              10
       13
                      550
                                    4
                                                     2
                                                                 10
                                                                        0.548675
                                                                                        1220
                                                                                                       11
                                                                                                           0.888450
       14
                      606
                                    6
                                                     2
                                                                 10
                                                                        0.254230
                                                                                        1239
                                                                                                       11 0.943195
                                                                                                                         1
                      721
                                    6
                                                     2
                                                                 10
                                                                        0.950553
                                                                                        1445
                                                                                                       11 0.943195
       15
      54030
                       73
                                    5
                                                     3
                                                                  2
                                                                        0.887740
                                                                                         569
                                                                                                       12 0.943195
                                                                                                                         1
      54031
                       82
                                    2
                                                     3
                                                                  2
                                                                        1.186164
                                                                                         490
                                                                                                       12 0.943195
      54032
                       82
                                    5
                                                     3
                                                                  2
                                                                        0.915593
                                                                                          584
                                                                                                       12 0.888450
      54033
                       79
                                    2
                                                     3
                                                                  2
                                                                        1.034963
                                                                                          491
                                                                                                       12 0.888450
                                    5
                                                                  2
                                                                                                       12 0.888450
      54034
                       72
                                                     3
                                                                         1.158311
                                                                                         492
                                                                                                                         1
     42595 rows × 9 columns
 Next steps: ( Generate code with df11 ) ( View recommended plots )
                                                                    New interactive sheet
X = df11.drop('class', axis=1)
y = df11['class']
y.value_counts()
→▼
             count
      class
             23870
        1
        0
             18725
     dtyne int64
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=30)
def runModel(model, X_train, X_test, y_train, y_test):
  model.fit(X_train, y_train)
  y_pred = model.predict(X_test)
  y_train_pred = model.predict(X_train)
  acc_test = accuracy_score(y_test, y_pred)
  acc_train = accuracy_score(y_train, y_train_pred)
  y_pred_proba = model.predict_proba(X_test)
  cm = confusion_matrix(y_test, y_pred)
  # cr = classification_report(y_test, y_pred)
  print(f'Train Accuracy: {acc_train}')
  print(f'Test Accuracy: {acc_test}')
  return acc_test, acc_train, cm, y_pred_proba, y_pred
from sklearn.linear_model import LogisticRegression as LR
from sklearn.tree import DecisionTreeClassifier as DT
from xgboost import XGBClassifier as \mathsf{XGB}
from \ sklearn.neighbors \ import \ KNeighbors Classifier \ as \ KNN
from sklearn.ensemble import RandomForestClassifier as RF
from \ sklearn.metrics \ import \ accuracy\_score, \ confusion\_matrix, \ classification\_report
# model = LogisticRegression(max_iter= 10000)
models\_arr = [LR(max\_iter=10000), DT(), XGB(), KNN(), RF()]
for model in models_arr:
  print(model.__class__.__name__)
```

```
runModel(model, X_train, X_test, y_train, y_test)
acc_test, acc_train, cm, y_pred_proba, y_pred = runModel(model, X_train, X_test, y_train, y_test)
```

```
→ LogisticRegression
    Train Accuracy: 0.6397236383149987
    Test Accuracy: 0.6373738164175601
    [[2921 2740]
     [1894 5224]]
    Train Accuracy: 0.6397236383149987
    Test Accuracy: 0.6373738164175601
    [[2921 2740]
     [1894 5224]]
    DecisionTreeClassifier
    Train Accuracy: 1.0
    Test Accuracy: 0.9701854605211675
    [[5451 210]
     [ 171 6947]]
    Train Accuracy: 1.0
Test Accuracy: 0.9689334063698255
    [[5446 215]
     [ 182 6936]]
    XGBClassifier
    Train Accuracy: 0.9946337536892943
    Test Accuracy: 0.9854448704906488
    [[5567 94]
     [ 92 7026]]
    Train Accuracy: 0.9946337536892943
    Test Accuracy: 0.9854448704906488
    [[5567
            941
     [ 92 7026]]
    KNeighborsClassifier
    Train Accuracy: 0.8135564797424202
    Test Accuracy: 0.7082713827373034
    [[3769 1892]
     [1836 5282]]
    Train Accuracy: 0.8135564797424202
    Test Accuracy: 0.7082713827373034
    [[3769 1892]
     [1836 5282]]
    RandomForestClassifier
    Train Accuracy: 1.0
    Test Accuracy: 0.986540417873073
    [[5570 91]
     [ 81 7037]]
    Train Accuracy: 1.0
    Test Accuracy: 0.9873229517176618
    [[5582 79]
     [ 83 7035]]
```

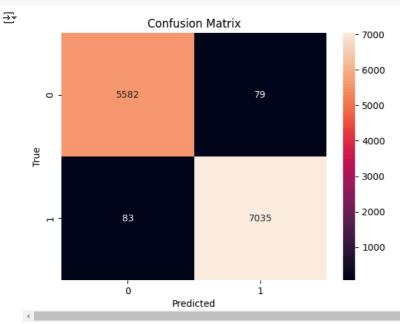
df11

_ _ *		cap-diameter	cap-shape	gill-attachment	gill-color	stem-height	stem-width	stem-color	season	class	
	11	642	6	2	10	0.286062	1311	11	0.943195	1	ılı
	12	814	4	2	10	1.189292	1681	11	0.943195	1	*/
	13	550	4	2	10	0.548675	1220	11	0.888450	1	
	14	606	6	2	10	0.254230	1239	11	0.943195	1	
	15	721	6	2	10	0.950553	1445	11	0.943195	1	
54	4030	73	5	3	2	0.887740	569	12	0.943195	1	
54	4031	82	2	3	2	1.186164	490	12	0.943195	1	
54	4032	82	5	3	2	0.915593	584	12	0.888450	1	
54	4033	79	2	3	2	1.034963	491	12	0.888450	1	
54	4034	72	5	3	2	1.158311	492	12	0.888450	1	
42	595 ro	ws × 9 columns									

```
# prompt: visualize cm
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'cm' is your confusion matrix from the previous code
sns.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Predicted')
```

```
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```



```
from sklearn.preprocessing import MinMaxScaler
minmaxscaller = MinMaxScaler()
df11_scaled = minmaxscaller.fit_transform(df11)
df11_scaled = pd.DataFrame(df11_scaled, columns=df11.columns)
df11_scaled
```

→	cap-diameter	cap-shape	gill-attachment	gill-color	stem-height	stem-width	stem-color	season	class	
0	0.448324	1.000000	0.333333	0.909091	0.145379	0.454545	0.916667	1.0	1.0	
1	0.568436	0.666667	0.333333	0.909091	0.605092	0.582929	0.916667	1.0	1.0	4
2	0.384078	0.666667	0.333333	0.909091	0.279040	0.422970	0.916667	0.0	1.0	
3	0.423184	1.000000	0.333333	0.909091	0.129177	0.429563	0.916667	1.0	1.0	
4	0.503492	1.000000	0.333333	0.909091	0.483582	0.501041	0.916667	1.0	1.0	
42590	0.050978	0.833333	0.500000	0.181818	0.451613	0.197085	1.000000	1.0	1.0	
42591	0.057263	0.333333	0.500000	0.181818	0.603500	0.169674	1.000000	1.0	1.0	
42592	0.057263	0.833333	0.500000	0.181818	0.465789	0.202290	1.000000	0.0	1.0	
42593	0.055168	0.333333	0.500000	0.181818	0.526544	0.170021	1.000000	0.0	1.0	
42594	0.050279	0.833333	0.500000	0.181818	0.589324	0.170368	1.000000	0.0	1.0	
42595	rows × 9 columns									

Next steps: Generate code with df11_scaled

© View recommended plots

(New interactive sheet

```
X_scaled = df11_scaled.drop('class', axis=1)
y_scaled = df11_scaled['class']
```

X_train_scaled, X_test_scaled, y_train_scaled, y_test_scaled = train_test_split(X_scaled, y_scaled, train_size=0.7, random_state=30)

 ${\tt X_train_scaled.shape,\ X_test_scaled.shape,\ y_train_scaled.shape,\ y_test_scaled.shape}$

→ ((29816, 8), (12779, 8), (29816,), (12779,))

```
for model in models_arr:

print(model.__class__.__name__)

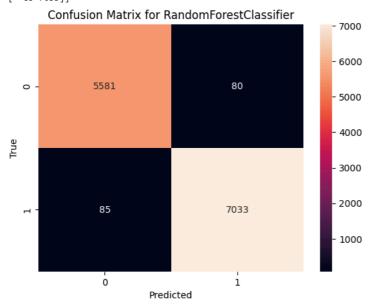
runModel(model, X_train_scaled, X_test_scaled, y_train_scaled, y_test_scaled)
```

LogisticRegression
Train Accuracy: 0.6393882479205796
Test Accuracy: 0.637452069802019
[[2925 2736]
[1897 5221]]

```
DecisionTreeClassifier
     Train Accuracy: 1.0
     Test Accuracy: 0.9698724469833321
     [[5450 211]
      [ 174 6944]]
     XGBClassifier
     Train Accuracy: 0.9946337536892943
     Test Accuracy: 0.9854448704906488
     [[5567 94]
      [ 92 7026]]
     KNeighborsClassifier
     Train Accuracy: 0.9894687416152401
     Test Accuracy: 0.9827842554190469
     [[5538 123]
[ 97 7021]]
     RandomForestClassifier
     Train Accuracy: 1.0
     Test Accuracy: 0.9871664449487441
     [[5569 92]
      [ 72 7046]]
y_train_scaled.value_counts()
₹
             count
      class
       1.0
             16752
       0.0
            13064
     dtyne int64
from imblearn.over sampling import SMOTE
smote = SMOTE()
X_train_smote, y_train_smote = smote.fit_resample(X_train_scaled, y_train_scaled)
y_train_smote.value_counts()
Đ
             count
      class
            16752
       0.0
       1.0
             16752
     dtyne: int64
for model in models_arr:
  print(model.__class__.__name__)
  runModel(model, X_train_smote, X_test_scaled, y_train_smote, y_test_scaled)
\longrightarrow LogisticRegression
     Train Accuracy: 0.6423412129894938
     Test Accuracy: 0.6384693637999843
     [[3672 1989]
      [2631 4487]]
     DecisionTreeClassifier
     Train Accuracy: 1.0
     Test Accuracy: 0.9744111432819469
     [[5479 182]
      [ 145 6973]]
     XGBClassifier
     Train Accuracy: 0.9951946036294174
     Test Accuracy: 0.9860708975663197
     [[5582 79]
[ 99 7019]]
     {\it KNeighborsClassifier}
     Train Accuracy: 0.9905981375358166
     Test Accuracy: 0.9828625088035058
     [[5555 106]
[ 113 7005]]
     RandomForestClassifier
     Train Accuracy: 1.0
     Test Accuracy: 0.9870881915642852
     [[5585 76]
      [ 89 7029]]
 visualise the randomforestclassifier cm and write the train and test accuracy
                                                                                                                             Q
                                                                                                                                     Close
               Use code with caution
 < 1 of 1 >
# nnownt: visualise the nandomfonestelassifier cm and white the thair and test assurance
```

```
π prompt. Visuaitist the randomiorestetassifier om and write the crain and test accuracy
\mbox{\tt\#} Assuming 'cm' is your confusion matrix from the previous code
# You need to call runModel for RandomForestClassifier and capture its output
# Example:
model = RF()
acc_test, acc_train, cm, y_pred_proba, y_pred = runModel(model, X_train, X_test, y_train, y_test)
sns.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix for RandomForestClassifier')
plt.show()
print(f'Train Accuracy for RandomForestClassifier: {acc_train}')
print(f'Test Accuracy for RandomForestClassifier: {acc_test}')
```

→ Train Accuracy: 1.0
Test Accuracy: 0.9870881915642852 [[5581 80] [85 7033]]



Train Accuracy for RandomForestClassifier: 1.0

Tact Accuracy for RandomEnractClassifian @ 0870881915642852

df_smote = pd.concat([X_train_smote, y_train_smote], axis=1) df_smote

_		cap-diameter	cap-shape	gill-attachment	gill-color	stem-height	stem-width	stem-color	season	class	
	0	0.729749	1.000000	0.000000	0.909091	0.082599	0.860861	0.500000	1.0	0.0	th
	1	0.209497	0.333333	0.166667	0.454545	0.552871	0.192575	0.500000	0.0	1.0	+/
	2	0.646648	1.000000	0.333333	0.636364	0.566614	0.588480	0.916667	1.0	0.0	
	3	0.151536	0.166667	0.000000	0.454545	0.337770	0.095767	1.000000	0.0	1.0	
	4	0.837291	0.333333	1.000000	0.454545	0.161580	0.737335	0.500000	0.0	0.0	
	33499	0.502108	0.833333	0.833333	0.636364	0.191351	0.589266	0.833333	1.0	0.0	
	33500	0.699389	1.000000	0.333333	0.636364	0.588786	0.630264	0.916667	1.0	0.0	
	33501	0.517636	0.666667	0.000000	0.909091	0.353267	0.453976	0.250000	1.0	0.0	
	33502	0.407532	1.000000	0.166667	0.000000	0.032293	0.429930	0.916667	1.0	0.0	
	33503	0.249683	1.000000	0.166667	0.545455	0.013697	0.247544	0.583333	1.0	0.0	
(33504 rd	ws × 9 columns									

Next steps: (Generate code with df_smote)

View recommended plots

New interactive sheet

df12 = dataPreperation(df_smote) df12

	cap-diameter	cap-shape	gill-attachment	gill-color	stem-height	stem-width	stem-color	season	class	=
0	0.729749	1.000000	0.000000	0.909091	0.082599	0.860861	0.500000	1.0	0.0	il.
1	0.209497	0.333333	0.166667	0.454545	0.552871	0.192575	0.500000	0.0	1.0	+//
2	0.646648	1.000000	0.333333	0.636364	0.566614	0.588480	0.916667	1.0	0.0	-
3	0.151536	0.166667	0.000000	0.454545	0.337770	0.095767	1.000000	0.0	1.0	
4	0.837291	0.333333	1.000000	0.454545	0.161580	0.737335	0.500000	0.0	0.0	
33499	0.502108	0.833333	0.833333	0.636364	0.191351	0.589266	0.833333	1.0	0.0	
33500		1.000000	0.333333	0.636364	0.588786	0.630264	0.916667	1.0	0.0	
33501		0.666667	0.000000	0.909091	0.353267	0.453976	0.250000	1.0	0.0	
33502		1.000000	0.166667	0.000000	0.032293	0.429930	0.916667	1.0	0.0	
33503		1.000000	0.166667	0.545455	0.013697	0.247544	0.583333	1.0	0.0	
	rows × 9 columns		0.100001	0.010100	0.0.000.	0.2	0.000000		0.0	
t steps:	Generate code	with df12	View recomment	nded plots (New interactive	sheet				
= dataF	Preperation(df1	12)								
	cap-diameter	cap-shape	gill-attachment	gill-color	stem-height	stem-width	stem-color	season	class	
0	0.729749	1.000000	0.000000	0.909091	0.082599	0.860861	0.500000	1.0	0.0	ıl.
1	0.209497	0.333333	0.166667	0.454545	0.552871	0.192575	0.500000	0.0	1.0	1
2	0.646648	1.000000	0.333333	0.636364	0.566614	0.588480	0.916667	1.0	0.0	
3	0.151536	0.166667	0.000000	0.454545	0.337770	0.095767	1.000000	0.0	1.0	
4	0.837291	0.333333	1.000000	0.454545	0.161580	0.737335	0.500000	0.0	0.0	
33499	0.502108	0.833333	0.833333	0.636364	0.191351	0.589266	0.833333	1.0	0.0	
33500	0.699389	1.000000	0.333333	0.636364	0.588786	0.630264	0.916667	1.0	0.0	
33501	0.517636	0.666667	0.000000	0.909091	0.353267	0.453976	0.250000	1.0	0.0	
33502	0.407532	1.000000	0.166667	0.000000	0.032293	0.429930	0.916667	1.0	0.0	
33503	0.249683	1.000000	0.166667	0.545455	0.013697	0.247544	0.583333	1.0	0.0	
33287 rd	rows × 9 columns									
t steps:	Generate code	with df13	View recomment	nded plots (New interactive	sheet				
= dataF	Preperation(df1	13)								
uu cu	r eper dezon (dr.	,								
	•		gill-attachment							
0	0.729749	1.000000	0.000000	0.909091	0.082599	0.860861	0.500000	1.0	0.0	11.
1	0.209497	0.333333	0.166667	0.454545	0.552871	0.192575	0.500000	0.0	1.0	7
2	0.646648	1.000000	0.333333	0.636364	0.566614	0.588480	0.916667	1.0	0.0	
3	0.151536	0.166667	0.000000	0.454545	0.337770	0.095767	1.000000	0.0	1.0	
	0.837291	0.333333	1.000000	0.454545	0.161580	0.737335	0.500000	0.0	0.0	
4				0.636364	0.191351	0.589266	0.833333	1.0	0.0	
	0.502108	0.833333	0.833333		3.101001	5.000200				
 33499	0.502108	0.833333	0.833333		0.588786	0 630264	() 9 Innn /	1.0	()()	
 33499 33500	0.502108 0.699389	1.000000	0.333333	0.636364	0.588786	0.630264	0.916667	1.0	0.0	
 33499 33500 33501	0.502108 0.699389 0.517636	1.000000 0.666667	0.333333 0.000000	0.636364 0.909091	0.353267	0.453976	0.250000	1.0	0.0	
 33499 33500 33501 33502	0.502108 0.699389 0.517636 0.407532	1.000000 0.666667 1.000000	0.333333 0.000000 0.166667	0.636364 0.909091 0.000000	0.353267 0.032293	0.453976 0.429930	0.250000 0.916667	1.0 1.0	0.0	
 33499 33500 33501 33502 33503	0.502108 0.699389 0.517636 0.407532	1.000000 0.666667	0.333333 0.000000	0.636364 0.909091	0.353267	0.453976	0.250000	1.0	0.0	

```
df15 = dataPreperation(df14)
\overline{\Rightarrow}
              cap-diameter cap-shape gill-attachment gill-color stem-height stem-width stem-color season
                                                                                                                        class
                                                                                                                                  \blacksquare
        0
                  0.729749
                              1.000000
                                                 0.000000
                                                              0.909091
                                                                            0.082599
                                                                                         0.860861
                                                                                                      0.500000
                                                                                                                    1.0
                                                                                                                           0.0
                                                                                                                                  th
        1
                  0.209497
                              0.333333
                                                 0.166667
                                                              0.454545
                                                                            0.552871
                                                                                         0.192575
                                                                                                      0.500000
                                                                                                                    0.0
                                                                                                                           1.0
                                                                                                                                  +1
        2
                  0.646648
                              1.000000
                                                 0.333333
                                                              0.636364
                                                                            0.566614
                                                                                         0.588480
                                                                                                      0.916667
                                                                                                                    1.0
                                                                                                                           0.0
        3
                  0.151536
                              0.166667
                                                 0.000000
                                                              0.454545
                                                                            0.337770
                                                                                         0.095767
                                                                                                      1.000000
                                                                                                                    0.0
                                                                                                                           1.0
        4
                  0.837291
                              0.333333
                                                 1.000000
                                                              0.454545
                                                                            0.161580
                                                                                         0.737335
                                                                                                      0.500000
                                                                                                                    0.0
                                                                                                                           0.0
      33499
                  0.502108
                              0.833333
                                                 0.833333
                                                              0.636364
                                                                            0.191351
                                                                                         0.589266
                                                                                                      0.833333
                                                                                                                           0.0
                                                                                                                    1.0
      33500
                  0.699389
                              1.000000
                                                 0.333333
                                                              0.636364
                                                                            0.588786
                                                                                         0.630264
                                                                                                      0.916667
                                                                                                                    1.0
                                                                                                                           0.0
      33501
                  0.517636
                              0.666667
                                                 0.000000
                                                              0.909091
                                                                            0.353267
                                                                                         0.453976
                                                                                                      0.250000
                                                                                                                    1.0
                                                                                                                           0.0
      33502
                  0.407532
                              1.000000
                                                 0.166667
                                                              0.000000
                                                                                         0.429930
                                                                                                      0.916667
                                                                            0.032293
                                                                                                                    1.0
                                                                                                                           0.0
                  0.249683
      33503
                              1.000000
                                                 0.166667
                                                              0.545455
                                                                            0.013697
                                                                                         0.247544
                                                                                                      0.583333
                                                                                                                           0.0
                                                                                                                    1.0
     33250 rows × 9 columns
 Next steps: ( Generate code with df15 )
                                        View recommended plots
                                                                       New interactive sheet
df16 = dataPreperation(df15)
df16
<del>____</del>
              cap-diameter cap-shape gill-attachment gill-color stem-height stem-width stem-color season class
                                                                                                                                  \blacksquare
        0
                  0.729749
                              1.000000
                                                 0.000000
                                                              0.909091
                                                                            0.082599
                                                                                         0.860861
                                                                                                      0.500000
                                                                                                                    1.0
                                                                                                                           0.0
                                                                                                                                  d.
        1
                  0.209497
                              0.333333
                                                 0.166667
                                                              0.454545
                                                                            0.552871
                                                                                         0.192575
                                                                                                      0.500000
                                                                                                                    0.0
                                                                                                                           1.0
                                                                                                                                  +/
        2
                  0.646648
                              1.000000
                                                 0.333333
                                                              0.636364
                                                                            0.566614
                                                                                         0.588480
                                                                                                      0.916667
                                                                                                                    1.0
                                                                                                                           0.0
        3
                  0.151536
                              0.166667
                                                 0.000000
                                                              0.454545
                                                                            0.337770
                                                                                         0.095767
                                                                                                      1.000000
                                                                                                                    0.0
                                                                                                                           1.0
        4
                  0.837291
                              0.333333
                                                 1.000000
                                                              0.454545
                                                                            0.161580
                                                                                         0.737335
                                                                                                      0.500000
                                                                                                                    0.0
                                                                                                                           0.0
      33499
                  0.502108
                              0.833333
                                                 0.833333
                                                              0.636364
                                                                            0.191351
                                                                                         0.589266
                                                                                                      0.833333
                                                                                                                    1.0
                                                                                                                           0.0
      33500
                  0.699389
                              1.000000
                                                 0.333333
                                                              0.636364
                                                                            0.588786
                                                                                         0.630264
                                                                                                      0.916667
                                                                                                                    1.0
                                                                                                                           0.0
      33501
                  0.517636
                              0.666667
                                                 0.000000
                                                              0.909091
                                                                            0.353267
                                                                                         0.453976
                                                                                                      0.250000
                                                                                                                    1.0
                                                                                                                           0.0
      33502
                  0.407532
                              1.000000
                                                 0.166667
                                                              0.000000
                                                                            0.032293
                                                                                         0.429930
                                                                                                      0.916667
                                                                                                                    1.0
                                                                                                                           0.0
                              1.000000
      33503
                  0.249683
                                                 0.166667
                                                              0.545455
                                                                            0.013697
                                                                                         0.247544
                                                                                                      0.583333
                                                                                                                    1.0
                                                                                                                           0.0
     33250 rows × 9 columns
 Next steps: ( Generate code with df16 )

    View recommended plots

                                                                       New interactive sheet
X = df16.drop('class', axis=1)
y = df16['class']
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=30)
models output = []
for model in models arr:
  print(model.__class__.__name__)
  acc_test, acc_train, cm, y_pred_proba, y_pred= runModel(model, X_train, X_test, y_train, y_test)
  models_output.append([model.__class__.__name__, acc_test, acc_train, cm, y_pred_proba,y_pred])
→ LogisticRegression
     Train Accuracy: 0.6435660580021483
     Test Accuracy: 0.6368922305764411
     [[3348 1627]
      [1995 3005]]
     DecisionTreeClassifier
     Train Accuracy: 1.0
     Test Accuracy: 0.9711278195488722
     [[4825 150]
      [ 138 4862]]
     XGBClassifier
     Train Accuracy: 0.9957894736842106
     Test Accuracy: 0.9884711779448622
```

```
[[4927 48]
  [67 4933]]

KNeighborsClassifier

Train Accuracy: 0.989301825993553

Test Accuracy: 0.983358395989975
[[4891 84]
  [82 4918]]

RandomForestClassifier

Train Accuracy: 1.0

Test Accuracy: 0.9895739348370928
[[4929 46]
  [58 4942]]
```

```
# prompt: visualise the randomforestclassifier cm and write the train and test accuracy

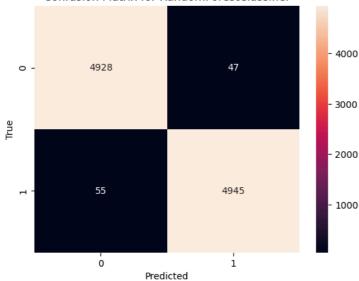
# Assuming 'cm' is your confusion matrix from the previous code
# You need to call runModel for RandomForestClassifier and capture its output
# Example:
model = RF()
acc_test, acc_train, cm, y_pred_proba, y_pred = runModel(model, X_train, X_test, y_train, y_test)

sns.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix for RandomForestClassifier')
plt.show()

print(f'Train Accuracy for RandomForestClassifier: {acc_train}')
print(f'Test Accuracy for RandomForestClassifier: {acc_test}')
```

```
Train Accuracy: 1.0
Test Accuracy: 0.9897744360902255
[[4928 47]
[ 55 4945]]
```

Confusion Matrix for RandomForestClassifier



Train Accuracy for RandomForestClassifier: 1.0

Tast Accuracy for RandomForestClassifier: 0 98977///360902255

```
models_output[0][4]
```

models_output[0][5]

```
\rightarrow array([1., 1., 1., ..., 0., 0., 1.])
```

```
df_y_pred = pd.DataFrame(models_output[2][5], columns=['Prediction'])
df_y_pred
```

```
→
            Prediction
                           0
                     0
                           ılı.
       1
                           +1
       2
       3
                     1
      9970
                     0
      9971
      9972
      9973
                     0
 Next 9974 Generate code with df_y_pred View recommended plots
                                                                       New interactive sheet
     9975 rows × 1 columns
df_pred_proba = pd.DataFrame(models_output[2][4], columns=['0', '1'])
df_pred_proba
₹
                            1
                                 \blacksquare
                  0
       0
           0.945217 0.054783
       1
            0.076579 0.923421
            0.000010 0.999990
       2
       3
            0.008000 0.992000
            0.000188 0.999812
       4
      9970 0.999559 0.000441
      9971 0.997815 0.002185
      9972 0.271758 0.728242
      9973 0.999607 0.000393
      9974 0.000339 0.999661
     9975 rows × 2 columns

    View recommended plots

                                                                            New interactive sheet
 Next steps: (Generate code with df_pred_proba)
df_both = pd.concat([df_y_pred, df_pred_proba], axis=1)
df_both
₹
                                             \blacksquare
            Prediction
                               0
                                        1
       0
                     0 0.945217 0.054783
                                             th
       1
                     1 0.076579 0.923421
       2
                        0.000010 0.999990
       3
                        0.008000 0.992000
                        0.000188 0.999812
      9970
                     0 0.999559 0.000441
      9971
                     0 0.997815 0.002185
      9972
                        0.271758 0.728242
      9973
                     0 0.999607 0.000393
                     1 0.000339 0.999661
      9974
     9975 rows × 3 columns
```