Naive Bayes Homework

Eren ATAS - 1334129

Instructions

I have used Python 3 for this assignment. The required libraries are:
pandas
numpy
sklearn
pandas_ml

You can install them with “pip install <package_name>”.
Or, I have created a Dockerfile as you can see in the folder. If you have Docker Community Edition installed, change directory (cd) to the root folder of the homework, and simply type:

```
docker build -t naive_bayes .

This will create a docker image for my assignment, if you run this image with the command:

docker run -it --rm --name naive-bayes naive-bayes

It will start and give the output of the python code so that you can avoid environmental issues.

I have also put a proof video so that you can see it, if any case that the code would not work.

Note: Rarely, the code gives ZeroDivisionError while trying to find probability of X given Y. I couldn't think of a solution for it. If you encounter with this problem. Simply run the code once again. Or if you are trying the Docker image, run the image once again.
```

Example Output of the Code

```
/Users/erenatas/.conda/envs/CannyEdgeDetection/bin/python /Users/erenatas/Documents/Homeworks/NaiveBayes/NaiveBayes.py
/Users/erenatas/.conda/envs/CannyEdgeDetection/lib/python3.6/site-packages/sklearn/externals/joblib/externals/cloudpickle/cloudpickle.py:47: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  import imp
Test 1:
/Users/erenatas/.conda/envs/CannyEdgeDetection/lib/python3.6/site-packages/pandas_ml/confusion_matrix/stats.py:60: FutureWarning: supplying multiple axes to axis is deprecated and will be removed in a future version.
  num = df[df > 1].dropna(axis=[0, 1], thresh=1).applymap(lambda n: choose(n, 2)).sum().sum() - np.float64(nis2 * njs2) / n2
Confusion Matrix:

Predicted    1    2    3  __all__
Actual
1          701   91   21     813
2          418  712  1195    2325
3           44  287  607     938
__all__    1163 1090 1823    4076
```

Overall Statistics:

Accuracy: 0.4955839057899902
95% CI: (0.4801199075056834, 0.511054239675202)
No Information Rate: ToDo
P-Value [Acc > NIR]: 3.3215839462979336e-10
Kappa: 0.2664362926822778
Mcnemar's Test P-Value: ToDo

Class Statistics:

Classes	1	2	3
Population	4076	4076	4076
P: Condition positive	813	2325	938
N: Condition negative	3263	1751	3138
Test outcome positive	1163	1090	1823
Test outcome negative	2913	2986	2253
TP: True Positive	701	712	607
TN: True Negative	2801	1373	1922
FP: False Positive	462	378	1216
FN: False Negative	112	1613	331
TPR: (Sensitivity, hit rate, recall)	0.862239	0.306237	0.647122
TNR=SPC: (Specificity)	0.858413	0.784123	0.612492
PPV: Pos Pred Value (Precision)	0.602752	0.653211	0.332968
NPV: Neg Pred Value	0.961552	0.459812	0.853085
FPR: False-out	0.141587	0.215877	0.387508
FDR: False Discovery Rate	0.397248	0.346789	0.667032
FNR: Miss Rate	0.137761	0.693763	0.352878
ACC: Accuracy	0.859176	0.511531	0.620461
F1 score	0.709514	0.416984	0.439696
MCC: Matthews correlation coefficient	0.637703	0.101058	0.219777
Informedness	0.720651	0.0903599	0.259614
Markedness	0.564303	0.113023	0.186052
Prevalence	0.19946	0.570412	0.230128
LR+: Positive likelihood ratio	6.08979	1.41857	1.66996
LR-: Negative likelihood ratio	0.160484	0.884763	0.576136
DOR: Diagnostic odds ratio	37.9464	1.60334	2.89855
FOR: False omission rate	0.0384483	0.540188	0.146915

Test 2:
/Users/erenatas/.conda/envs/CannyEdgeDetection/lib/python3.6/site-packages/pandas_ml/confusion_matrix/stats.py:60: FutureWarning: supplying multiple axes to axis is deprecated and will be removed in a future version.
  num = df[df > 1].dropna(axis=[0, 1], thresh=1).applymap(lambda n: choose(n, 2)).sum().sum() - np.float64(nis2 * njs2) / n2
Confusion Matrix:

Predicted	1	2	3	__all__
Actual				
1	533	67	33	633
2	353	374	1087	1814
3	46	131	552	729
__all__	932	572	1672	3176

Overall Statistics:

Accuracy: 0.4593828715365239
95% CI: (0.4419379464525925, 0.4769026799359114)
No Information Rate: ToDo
P-Value [Acc > NIR]: 0.9999999999999829
Kappa: 0.24685119474778341
Mcnemar's Test P-Value: ToDo

Class Statistics:

Classes	1	2	3
Population	3176	3176	3176
P: Condition positive	633	1814	729
N: Condition negative	2543	1362	2447
Test outcome positive	932	572	1672
Test outcome negative	2244	2604	1504
TP: True Positive	533	374	552
TN: True Negative	2144	1164	1327
FP: False Positive	399	198	1120
FN: False Negative	100	1440	177
TPR: (Sensitivity, hit rate, recall)	0.842022	0.206174	0.757202
TNR=SPC: (Specificity)	0.843099	0.854626	0.542297
PPV: Pos Pred Value (Precision)	0.571888	0.653846	0.330144
NPV: Neg Pred Value	0.955437	0.447005	0.882314
FPR: False-out	0.156901	0.145374	0.457703
FDR: False Discovery Rate	0.428112	0.346154	0.669856
FNR: Miss Rate	0.157978	0.793826	0.242798
ACC: Accuracy	0.842884	0.484257	0.591625
F1 score	0.68115	0.313495	0.459808
MCC: Matthews correlation coefficient	0.601067	0.0783052	0.252251
Informedness	0.685121	0.0607998	0.299498
Markedness	0.527325	0.100851	0.212457
Prevalence	0.199307	0.571159	0.229534
LR+: Positive likelihood ratio	5.36657	1.41823	1.65435
LR-: Negative likelihood ratio	0.187378	0.928858	0.447722
DOR: Diagnostic odds ratio	28.6404	1.52685	3.69504
FOR: False omission rate	0.0445633	0.552995	0.117686

Test 3:
/Users/erenatas/.conda/envs/CannyEdgeDetection/lib/python3.6/site-packages/pandas_ml/confusion_matrix/stats.py:60: FutureWarning: supplying multiple axes to axis is deprecated and will be removed in a future version.
  num = df[df > 1].dropna(axis=[0, 1], thresh=1).applymap(lambda n: choose(n, 2)).sum().sum() - np.float64(nis2 * njs2) / n2
Confusion Matrix:

Predicted	1	2	3	__all__
Actual				
1	363	44	25	432
2	223	271	762	1256
3	25	85	378	488
__all__	611	400	1165	2176

Overall Statistics:

Accuracy: 0.4650735294117647
95% CI: (0.44394590020807906, 0.4862953518013109)
No Information Rate: ToDo
P-Value [Acc > NIR]: 0.999999999977563

Kappa: 0.2550627863147716
Mcnemar's Test P-Value: ToDo

Class Statistics:

Classes	1	2	3
Population	2176	2176	2176
P: Condition positive	432	1256	488
N: Condition negative	1744	920	1688
Test outcome positive	611	400	1165
Test outcome negative	1565	1776	1011
TP: True Positive	363	271	378
TN: True Negative	1496	791	901
FP: False Positive	248	129	787
FN: False Negative	69	985	110
TPR: (Sensitivity, hit rate, recall)	0.840278	0.215764	0.77459
TNR=SPC: (Specificity)	0.857798	0.859783	0.533768
PPV: Pos Pred Value (Precision)	0.594108	0.6775	0.324464
NPV: Neg Pred Value	0.955911	0.445383	0.891197
FPR: False-out	0.142202	0.140217	0.466232
FDR: False Discovery Rate	0.405892	0.3225	0.675536
FNR: Miss Rate	0.159722	0.784236	0.22541
ACC: Accuracy	0.85432	0.488051	0.587776
F1 score	0.696069	0.327295	0.45735
MCC: Matthews correlation coefficient	0.619641	0.0963505	0.257877
Informedness	0.698076	0.0755469	0.308358
Markedness	0.550019	0.122883	0.21566
Prevalence	0.198529	0.577206	0.224265
LR+: Positive likelihood ratio	5.90905	1.53878	1.66138
LR-: Negative likelihood ratio	0.1862	0.912133	0.422299
DOR: Diagnostic odds ratio	31.7349	1.68702	3.93413
FOR: False omission rate	0.0440895	0.554617	0.108803

Test 4:
/Users/erenatas/.conda/envs/CannyEdgeDetection/lib/python3.6/site-packages/pandas_ml/confusion_matrix/stats.py:60: FutureWarning: supplying multiple axes to axis is deprecated and will be removed in a future version.
  num = df[df > 1].dropna(axis=[0, 1], thresh=1).applymap(lambda n: choose(n, 2)).sum().sum() - np.float64(nis2 * njs2) / n2
Confusion Matrix:

Predicted	1	2	3	__all__
Actual				
1	700	103	10	813
2	419	964	942	2325
3	53	348	537	938
__all__	1172	1415	1489	4076

Overall Statistics:

Accuracy: 0.5399901864573111
95% CI: (0.5245428784489874, 0.5553801201854697)
No Information Rate: ToDo
P-Value [Acc > NIR]: 8.724549929059474e-114
Kappa: 0.3036054965459106
Mcnemar's Test P-Value: ToDo

Class Statistics:

Classes	1	2	3
Population	4076	4076	4076
P: Condition positive	813	2325	938
N: Condition negative	3263	1751	3138
Test outcome positive	1172	1415	1489
Test outcome negative	2904	2661	2587
TP: True Positive	700	964	537
TN: True Negative	2791	1300	2186
FP: False Positive	472	451	952
FN: False Negative	113	1361	401
TPR: (Sensitivity, hit rate, recall)	0.861009	0.414624	0.572495
TNR=SPC: (Specificity)	0.855348	0.742433	0.696622
PPV: Pos Pred Value (Precision)	0.59727	0.681272	0.360645
NPV: Neg Pred Value	0.961088	0.488538	0.844994
FPR: False-out	0.144652	0.257567	0.303378
FDR: False Discovery Rate	0.40273	0.318728	0.639355
FNR: Miss Rate	0.138991	0.585376	0.427505
ACC: Accuracy	0.856477	0.555447	0.668057
F1 score	0.70529	0.515508	0.442522
MCC: Matthews correlation coefficient	0.632442	0.163309	0.235246
Informedness	0.716356	0.157057	0.269117
Markedness	0.558358	0.16981	0.205639
Prevalence	0.19946	0.570412	0.230128
LR+: Positive likelihood ratio	5.95227	1.60977	1.88707
LR-: Negative likelihood ratio	0.162497	0.788457	0.613683
DOR: Diagnostic odds ratio	36.63	2.04167	3.07499
FOR: False omission rate	0.0389118	0.511462	0.155006

Test 5:

/Users/erenatas/.conda/envs/CannyEdgeDetection/lib/python3.6/site-packages/pandas_ml/confusion_matrix/stats.py:60: FutureWarning: supplying multiple axes to axis is deprecated and will be removed in a future version.

num = df[df > 1].dropna(axis=[0, 1], thresh=1).applymap(lambda n: choose(n, 2)).sum().sum() - np.float64(nis2 * njs2) / n2

Confusion Matrix:

Predicted	1	2	3	__all__
Actual				
1	547	73	13	633
2	352	571	891	1814
3	45	210	474	729
__all__	944	854	1378	3176

Overall Statistics:

Accuracy: 0.5012594458438288
95% CI: (0.4837185482524268, 0.5187980215623292)
No Information Rate: ToDo
P-Value [Acc > NIR]: 1.365261581928805e-14
Kappa: 0.2746543754882344
Mcnemar's Test P-Value: ToDo

Class Statistics:

Classes	1	2	3
Population	3176	3176	3176

P: Condition positive	633	1814	729
N: Condition negative	2543	1362	2447
Test outcome positive	944	854	1378
Test outcome negative	2232	2322	1798
TP: True Positive	547	571	474
TN: True Negative	2146	1079	1543
FP: False Positive	397	283	904
FN: False Negative	86	1243	255
TPR: (Sensitivity, hit rate, recall)	0.864139	0.314774	0.650206
TNR=SPC: (Specificity)	0.843885	0.792217	0.630568
PPV: Pos Pred Value (Precision)	0.579449	0.668618	0.343977
NPV: Neg Pred Value	0.96147	0.464686	0.858176
FPR: False-out	0.156115	0.207783	0.369432
FDR: False Discovery Rate	0.420551	0.331382	0.656023
FNR: Miss Rate	0.135861	0.685226	0.349794
ACC: Accuracy	0.847922	0.519521	0.635076
F1 score	0.693722	0.428036	0.449929
MCC: Matthews correlation coefficient	0.618857	0.119425	0.238242
Informedness	0.708024	0.106991	0.280774
Markedness	0.540919	0.133304	0.202153
Prevalence	0.199307	0.571159	0.229534
LR+: Positive likelihood ratio	5.53528	1.51492	1.76001
LR-: Negative likelihood ratio	0.160995	0.864947	0.554729
DOR: Diagnostic odds ratio	34.3818	1.75146	3.17275
FOR: False omission rate	0.0385305	0.535314	0.141824

Test 6:
/Users/erenatas/.conda/envs/CannyEdgeDetection/lib/python3.6/site-packages/pandas_ml/confusion_matrix/stats.py:60: FutureWarning: supplying multiple axes to axis is deprecated and will be removed in a future version.
  num = df[df > 1].dropna(axis=[0, 1], thresh=1).applymap(lambda n: choose(n, 2)).sum().sum() - np.float64(nis2 * njs2) / n2
Confusion Matrix:

Predicted	1	2	3	__all__
Actual				
1	368	56	8	432
2	232	461	563	1256
3	29	158	301	488
__all__	629	675	872	2176

Overall Statistics:

Accuracy: 0.5193014705882353
95% CI: (0.4980651016552343, 0.5404857866440511)
No Information Rate: ToDo
P-Value [Acc > NIR]: 4.027060798040976e-29
Kappa: 0.28647059856196655
Mcnemar's Test P-Value: ToDo

Class Statistics:

Classes	1	2	3
Population	2176	2176	2176
P: Condition positive	432	1256	488
N: Condition negative	1744	920	1688
Test outcome positive	629	675	872
Test outcome negative	1547	1501	1304

TP: True Positive	368	461	301
TN: True Negative	1483	706	1117
FP: False Positive	261	214	571
FN: False Negative	64	795	187
TPR: (Sensitivity, hit rate, recall)	0.851852	0.367038	0.616803
TNR=SPC: (Specificity)	0.850344	0.767391	0.66173
PPV: Pos Pred Value (Precision)	0.585056	0.682963	0.345183
NPV: Neg Pred Value	0.95863	0.470353	0.856595
FPR: False-out	0.149656	0.232609	0.33827
FDR: False Discovery Rate	0.414944	0.317037	0.654817
FNR: Miss Rate	0.148148	0.632962	0.383197
ACC: Accuracy	0.850643	0.536305	0.651654
F1 score	0.693685	0.477473	0.442647
MCC: Matthews correlation coefficient	0.617878	0.143563	0.23707
Informedness	0.702196	0.13443	0.278533
Markedness	0.543685	0.153316	0.201779
Prevalence	0.198529	0.577206	0.224265
LR+: Positive likelihood ratio	5.69207	1.57792	1.8234
LR-: Negative likelihood ratio	0.174221	0.824823	0.579083
DOR: Diagnostic odds ratio	32.6715	1.91304	3.14878
FOR: False omission rate	0.0413704	0.529647	0.143405

Process finished with exit code 0