COMP30027 Machine Learning

Sophia Xiao (Student ID: 1072038)

Fri 15:15 Practical 1

April 6th, 2021

Project 1 Report

2. The Gaussian na¨ıve Bayes classifier assumes that numeric attributes come from a Gaussian distribution. Is this assumption always true for the numeric attributes in this dataset? Identify some cases where the Gaussian assumption is violated and describe any evidence (or lack thereof) that this has some effect on the NB classifier's predictions.

The assumption is not always true. There are some cases that the assumption is violated:

- The calculation of likelihood is attribute-wised which I did in the project. However, if the prediction needs to be based on instances, we should not assume the points over an instance are normally distributed because they are positions of body parts, which does not have specific meaning in the math sense.
- We use the coordinate system to support the calculation. However, if the unit or quadrant is not consistent between instances, we cannot assume the normality.
- For Normal distribution, minor missing values will not affect the overall pattern as long as the sample size is huge enough. However, for our dataset, because the sample size is relatively small, and the missing value is not always meaningless (might be body part overlapped) which can affect the model learning. Therefore, when there are too many missing values and with small samples, we cannot assume normality.

5. Na¨ıve Bayes ignores missing values, but in pose recognition tasks the missing values can be informative. Missing values indicate that some part of the body was obscured and sometimes this is relevant to the pose (e.g., holding one hand behind the back). Are missing values useful for this task? Implement a method that incorporates information about missing values and demonstrate whether it changes the classification results.

In some cases, missing values are useful. For example, if an instance is missing two values, it might represent the hands and legs are overlapped. However, if there are too many of the missing values such as instance 38 in Figure 1 (all values are missing, infact), then the instance is useless because we cannot get any information out of it.

A1 bridge

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
13	bridge	9999	116.6055	9999	9999	9999	9999	16.7877	-3.4468	-30.059	-41.0149	-58.8725	9999	59.3746	9999	9999	9999	9999	53.34	10.894	-1.2233	-23.0051	-99.3801
14	bridge	154.2075	111.48	9999	9999	29.1509	-34.3625	29.1841	-90.696	-198.964	9999	9999	-2.0463	-8.9787	9999	9999	-28.6573	-34.9102	47.8313	39.3304	-12.5692	9999	9999
15	childs	1.6267	-41.4882	9999	9999	56.2959	120.8356	-137.27	9999	9999	9999	9999	26.1303	21.5752	9999	9999	-61.5984	-36.4199	50.3127	9999	9999	9999	9999
16	childs	9999	-51.0835	4.7563	46.3272	9999	9999	9999	9999	9999	9999	9999	9999	24.4596	-20.9928	-3.4668	9999	9999	9999	9999	9999	9999	9999
17	childs	82.07	56.8582	-14.052	-56.1548	9999	9999	-42.8275	39.9534	-65.8472	9999	9999	-5.3628	19.8305	-2.9058	-19.002	9999	9999	27.7523	-8.2588	-12.0533	9999	9999
18	childs	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
19	childs	-20.0349	-53.971	8.5043	65.5016	9999	9999	9999	9999	9999	9999	9999	7.5846	22.7416	-13.047	-17.2791	9999	9999	9999	9999	9999	9999	9999
20	childs	9999	60.2982	9999	9999	9999	9999	-39.0956	20.8136	-42.0142	9999	9999	9999	8.5797	9999	9999	9999	9999	19.5746	-15.6468	-12.5075	9999	9999
21	childs	80.3719	38.6375	-57.8227	9999	9999	9999	-61.1867	9999	9999	9999	9999	-22.2001	26.2811	-36.2892	9999	9999	9999	32.2083	9999	9999	9999	9999
22	childs	-70.7986	-52.9316	9999	9999	3.3939	43.3408	47.0198	-0.82988	65.6342	-34.8287	9999	-5.7012	22.0608	9999	9999	-18.7131	-15.2903	25.1801	9.3291	-8.319	-8.5465	9999
23	childs	75.9388	57.2092	1.057	-52.1312	0.28464	9999	-41.2729	23.5188	-64.6044	9999	9999	-6.743	12.9734	2.2275	-17.2486	-12.0278	9999	30.3309	8.2343	-17.7466	9999	9999
24	childs	9999	63.25	9999	9999	-5.3236	-57.9264	9999	9999	9999	9999	9999	9999	16.8622	9999	9999	-5.0846	-11.7776	9999	9999	9999	9999	9999
25	childs	-71.6412	-45.2287	9999	9999	9999	9999	54.7353	-33.8112	71.0063	-33.1505	58.09	-7.3048	23.7925	9999	9999	9999	9999	21.1087	-3.7531	-14.2922	-5.0802	-14.4709
26	childs	-39.4848	-24.0406	9999	9999	-71.4003	-117.243	75.8844	-2.9478	89.1794	1.3343	88.7188	-11.7924	24.8718	9999	9999	-3.2145	-12.4071	28.7432	-2.6124	-7.0538	-7.4978	-9.037
27	childs	9999	67.6307	7.7051	-75.3358	9999	9999	9999	9999	9999	9999	9999	9999	31.8827	-8.0288	-23.8539	9999	9999	9999	9999	9999	9999	9999
28	downwardd	9999	52.4085	9999	9999	0.37786	-52.7863	9999	9999	9999	9999	9999	9999	41.8511	9999	9999	-7.9166	-33.9345	9999	9999	9999	9999	9999
29	downwardd	9999	-16.5962	84.2596	184.0895	9999	9999	-91.8857	9999	9999	-159.867	9999	9999	0.65462	-93.8952	-146.913	9999	9999	66.4687	9999	9999	173.6846	9999
30	downwardd	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
31	downwardd	9999	9999	9999	9999	9999	9999	-5.674	46.4702	63.4406	-51.1864	-53.0505	9999	9999	9999	9999	9999	9999	71.3317	1.004	-58.8032	14.2402	-27.7727
32	downwardd	9999	60.0125	9.5174	-38.3682	14.1496	-45.3113	9999	9999	9999	9999	9999	9999	61.3413	10.0164	-21.4628	-3.9547	-45.9401	9999	9999	9999	9999	9999
33	downwardd	9999	9999	9999	9999	9999	9999	-6.6175	56.0203	77.7427	-54.4203	-72.7251	9999	9999	9999	9999	9999	9999	70.3845	9.2167	-24.7186	4.8531	-59.7356
34	downwardd	9999	9999	9999	9999	9999	9999	62.3007	10.6179	-56.9383	18.8163	-34.7966	9999	9999	9999	9999	9999	9999	61.2053	-9.4622	-54.9369	21.9552	-18.7613
35	downwardd	44.4239	44.5022	-62.2554	-146.357	-53.2183	-155.482	102.0445	9999	9999	226.3421	9999	-28.4721	45.7171	-64.8186	-121.677	-64.8564	-157.924	127.5027	9999	9999	264.5276	9999
36	downwardd	46.3884	43.7455	83.0487	128.5048	55.8631	9999	-19.8224	-65.6259	-109.915	-64.5048	-97.6822	-9.9479	18.7926	-16.4055	-43.6283	-34.5773	9999	95.9879	34.6885	-39.6752	31.3241	-36.5589
37	downwardd	9999	9999	9999	9999	9999	9999	-2.5996	50.2557	84.0979	-36.2044	-95.5497	9999	9999	9999	9999	9999	9999	75.5532	5.7584	-61.6718	19.0119	-38.6517
38	downwardd	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
39	downwardd	9999	9999	9999	9999	9999	9999	-15.3805	-68.1374	-85.636	62.8614	106.2925	9999	9999	9999	9999	9999	9999	81.7279	11.8585	-27.4602	21.6754	-87.8017
40	downwardd	52.8279	36.8724	59.4621	95.3937	64.8569	95.5083	-40.7949	-77.6613	-118.887	-76.3092	-91.2687	13.867	28.5465	-20.7407	-59.3036	-0.77684	-28.7319	91.5369	23.1772	-48.0262	24.5109	-24.0593
41	downwardd	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
42	downwardd	9999	9999	9999	9999	9999	9999	44.4419	-0.43153	-44.0104	9999	9999	9999	9999	9999	9999	9999	9999	79.6935	4.5178	-84.2113	9999	9999
43	downwardd	9999	9999	9999	9999	9999	9999	-0.29153	-59.6754	-81.0675	60.6134	80.4211	9999	9999	9999	9999	9999	9999	79.7157	15.3841	-8.8729	-2.3188	-83.9082
44	downwardd	9999	48.6276	9999	9999	10.1272	-58.7548	9999	9999	9999	9999	9999	9999	63.0106	9999	9999	2.1689	-65.1795	9999	9999	9999	9999	9999
45	downwardd	9999	-58.6116	9999	9999	9999	9999	-13.3296	-61.5512	9999	45.4288	88.0636	9999	-4.6085	9999	9999	9999	9999	84.5516	6.9877	9999	-1.751	-85.1799
46	mountain	0.78433	-1.1958	-36.3275	-46.9166	37.0375	43.6996	-1.2087	-13.6323	-7.986	15.7286	10.0168	170.7359	125.0568	62.4581	7.7529	62.4444	7.7542	25.0568	-68.2799	-163.344	-66.9721	-162.664
47	mountain	-1.4201	0.51683	-38.608	-46.9283	40.3096	48.6499	0.81333	-17.7961	-9.2875	14.3344	9.4159	166.6545	120.5194	63.4213	7.3744	61.3324	7.3444	20.5198	-63.2805	-157.974	-68.4438	-157.469
48	mountain	1.4154	-0.3254	-40.6665	-50.1107	40.3611	48.7715	-1.0042	-14.3397	-9.8798	17.2366	8.5417	168.5641	121.3532	64.1533	6.2822	64.1622	7.337	21.3555	-67.4018	-159.74	-65.2757	-160.79
49	mountain	-1.9764	-0.82728	-44.4066	-64.4084	41.0083	62.2855	1.8489	-13.6832	-9.038	18.2367	10.9604	166.5054	118.5077	61.1221	15.8253	59.8753	11.827	18.5435	-65.4561	-162.474	-65.4584	-158.818
50	mountain	1.4246	0.51492	-43.1046	-63.4257	44.724	64.9684	-1.7381	-16.1325	-10.0474	12.2866	10.5295	169.2809	119.6926	60.714	12.0586	61.9892	16.0722	19.718	-64.9626	-161.964	-67.622	-164.977
51	mountain	-1.6718	0.41788	-48.8466	-67.3111	45.6679	63.1613	3.3315	-15.1212	-8.8709	17.4763	11.7667	164.3521	117.4221	60.885	15.2813	58.7529	10.9157	17.4646	-62.9882	-159.958	-62.9965	-159.131

test

Figure 1

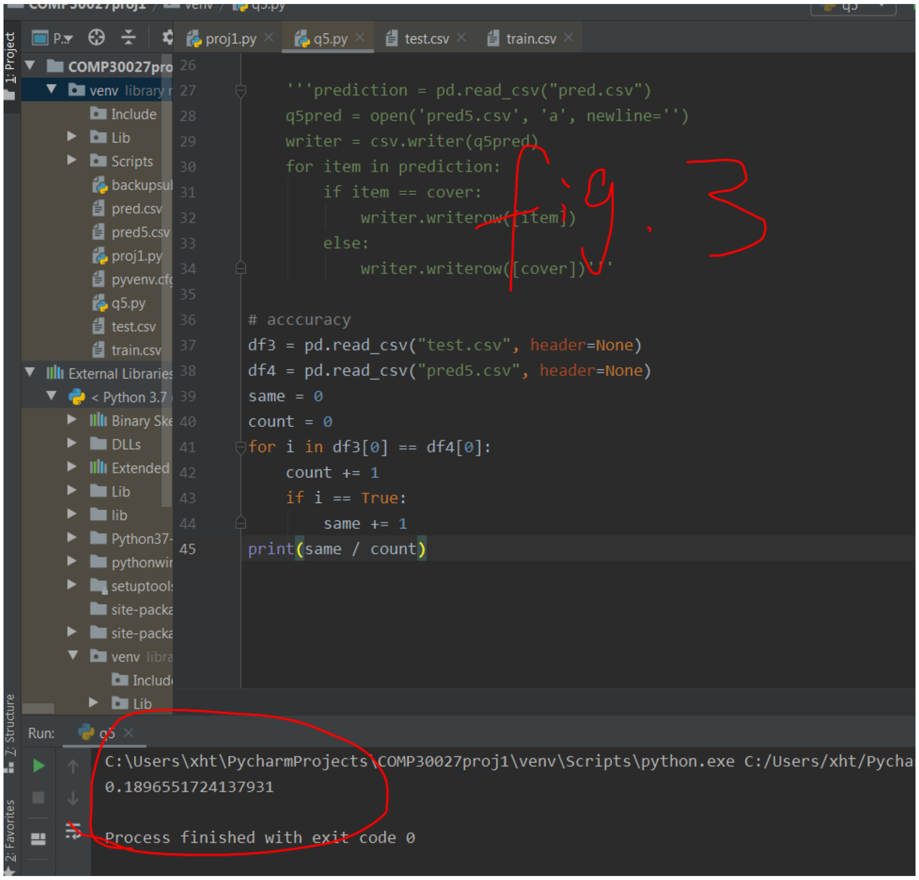
Similarly, if half of the body points are missing, the instance will not be as informative as well because of its unusuality.

Therefore, by looking at the picture of Yoga pose, I write a function (see in submitted template Q5) trying to create a dictionary that allows certain missing values for each pose and use this as support evidence to make the prediction.

However, the accuracy(True Positive) is very low (19%, Figure 3) compared to using NB only (74%, Figure 4 -- submitted code). The classification results are hugely affected, and I think it might because of the following reasons:

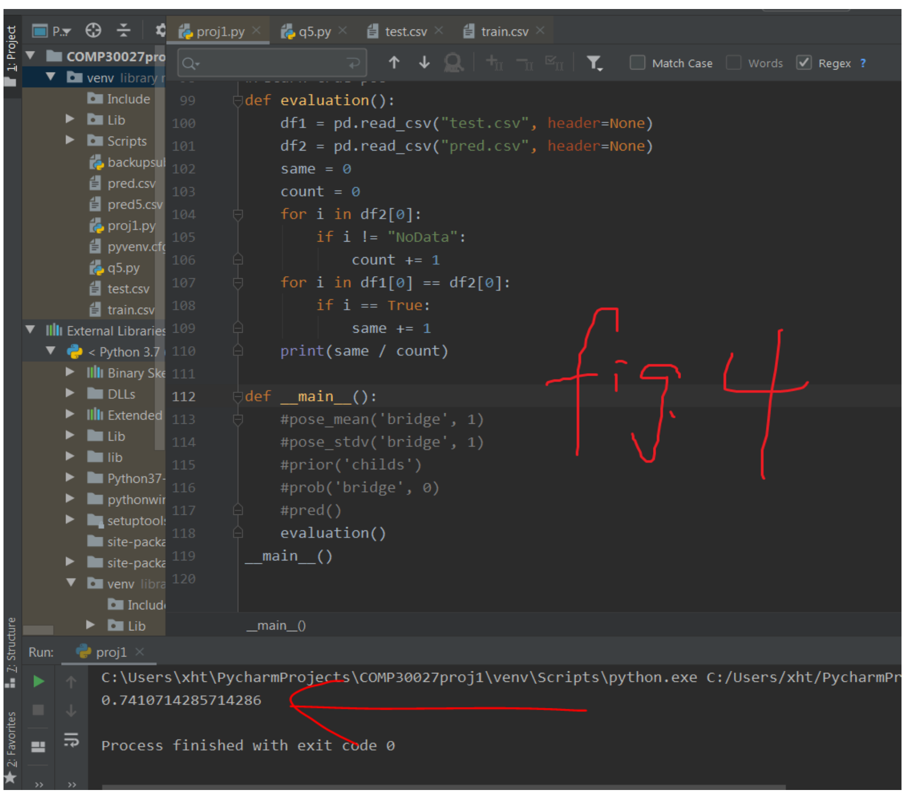
- I do not have Yoga experience, I do not know the “right number” of missing values corresponding to each pose, so the dictionary I built is very biased.
- When there are exactly the same number of missing values, or missing values are in the same position, it is hard to differentiate between poses.
- It is hard to decide the weighted importance between normal distribution results and missing value information.

Moreover, the result shows that there are actually still some right predictions made by this model, which shows there are some useful informative missing value.



COMP30027proj1
proj1.py
q5.py
test.csv
train.csv
'''prediction = pd.read_csv("pred.csv")
q5pred = open('pred5.csv', 'a', newline='')
writer = csv.writer(q5pred)
for item in prediction:
if item == cover:
writer.writerow([item])
else:
writer.writerow([cover])'''
# acccuracy
df3 = pd.read_csv("test.csv", header=None)
df4 = pd.read_csv("pred5.csv", header=None)
same = 0
count = 0
for i in df3[0] == df4[0]:
count += 1
if i == True:
same += 1
print(same / count)
Run: q5
C:\Users\xht\PycharmProjects\COMP30027proj1\venv\Scripts\python.exe C:/Users/xht/Pycha
0.1896551724137931
Process finished with exit code 0

fig. 3



COMP30027proj1
proj1.py
q5.py
test.csv
train.csv
Match Case
Words
Regex
def evaluation():
df1 = pd.read_csv("test.csv", header=None)
df2 = pd.read_csv("pred.csv", header=None)
same = 0
count = 0
for i in df2[0]:
if i != "NoData":
count += 1
for i in df1[0] == df2[0]:
if i == True:
same += 1
print(same / count)
def __main__():
#pose_mean('bridge', 1)
#pose_stdv('bridge', 1)
#prior('childs')
#prob('bridge', 0)
#pred()
evaluation()
__main__()
Run: proj1
C:\Users\xht\PycharmProjects\COMP30027proj1\venv\Scripts\python.exe C:/Users/xht/PycharmPr
0.7410714285714286
Process finished with exit code 0

fig. 4