LAPORAN

PENERAPAN NAIVE BAYES TERHADAP DATASET PARKINSONS

Mata Kuliah Kecerdasan Buatan

Nama : Iyan Zuli Armanda

NIM : 23051204165

Kelas : TI 23 E

S1 - Teknik Informatika

Teknik

Universitas Negeri Surabaya

2024

PENDAHULUAN

A. Tugas

- Penerapan Naive Bayes terhadap dataset Parkinson yang ada di "parkinsons.xlsx". Ketentuan sesuai PPT, 100 sampel pertama sebagai training set dan sisanya sebagai data uji/test.
- Manualisasi di excel seperti contoh di ppt yg 7 langkah itu. Terus buat model machine learningnya.
- Dibuat laporan (Penjelasan DataSet, Kode, Hasil)

B. Pengertian

Naive Bayes adalah algoritma klasifikasi yang didasarkan pada teorema Bayes dengan asumsi bahwa setiap fitur dalam data bersifat independen satu sama lain (naive). Algoritma ini menghitung probabilitas terjadinya suatu kelas berdasarkan nilai fitur-fitur input, kemudian memilih kelas dengan probabilitas tertinggi sebagai hasil prediksi. Naive Bayes sering digunakan dalam aplikasi seperti klasifikasi teks (misalnya, spam email) dan analisis sentimen karena kesederhanaannya dan kemampuannya untuk bekerja dengan data besar secara efisien.

PEMBAHASAN

A. Dataset

Dataset terdiri dari 195 data orang yand diklasifikasikan terkena penyakit Parkinson (PD) dan tidak terkena penyakit Parkinson (Healthy). Terdapat 23 kolom dengan kolom ke 23 berupa class identifikasi sehat atau tidak (PD atau Healthy). Jumlah data ialah 195, namun 100 pertama digunakan sebagai training dan 95 sisanya digunakan sebagai testing. Dataset disimpan dalam file excel dengan nama Parkinson.xlsx.

B. Manualisasi di Excel

1. Menghitung Prior Probability

class	jumlah	prior
Healthy	51	0,51
PD	49	0,49

2. Menghitung Mean dan Variance tiap Class

Semua fitur, a - v

	Mean/Variance	a	b	c	d	e	f	g	h	i	j
Healthy	Mean	145,48494	175,4205	118,3972	0,006093	4,62E-05	0,003282	0,003385	0,009845	0,033771	0,315863
	Variance	1139,4156	4913,53	1666,692	1,13E-05	8,97E-10	3,96E-06	2,86E-06	3,56E-05	0,000284	0,02711
PD	Mean	159,54049	187,6765	120,9274	0,004946	3,66E-05	0,002356	0,002889	0,007068	0,026271	0,242612
	Variance	2326,1086	2753,418	2768,235	6,47E-06	6,21E-10	1,69E-06	3,65E-06	1,52E-05	0,000201	0,018876

3. Menghitung Likelihood

Semua fitur, a - v dengan dataset 101-195 (testing data)

Healthy:

Healthy										
	a	b	c	d	e	f	g	h	i	j
Data-101	0,009943125	0,0050473	0,00976	9,08E-13	1,15E-07	1,68E-16	5,26E-09	5,31E-17	0,063394	0,00543
Data-102	0,010405714	0,0053414	0,005646	2,323353	641,3621	3,005202	0,79064	0,993374	5,997191	0,643357
Data-103	0,011617119	1,926E-10	0,004309	9,12E-10	0,000651	3,45E-11	5,33E-11	1,11E-11	0,038361	0,002308
Data-104	0,011701114	0,0054459	0,005605	66,59863	9076,114	106,8937	115,1152	35,63114	9,740282	1,083739
Data-105	0,011448304	0,0055599	0,00947	53,06732	6406,845	89,82046	87,17085	29,99957	8,976603	0,977403
Data-106	0,011727372	0,0055616	0,009117	68,52351	9076,114	113,5551	113,4857	37,92107	11,45148	1,246253
Data-107	0,011350911	0,0056128	0,008009	50,09014	6406,845	85,2911	81,44384	28,42522	9,230453	1,001531
Data-108	0,011608209	0,0055136	0,009105	68,73774	9076,114	112,947	103,8753	37,71816	12,31194	1,343057
Data-109	0,011601309	0,0055051	0,009178	51,27271	6406,845	89,24927	84,28231	29,74545	8,932159	0,969419
Data-110	0,004382903	0,0050778	0,006322	106,5688	13035,79	166,2092	225,8002	55,35793	23,21149	2,349486
Data-111	0,00309927	0,0044771	0,007631	118,7668	11500,88	195,869	236,0912	65,33898	19,19504	1,947247
Data-112	0,002067035	0,004636	0,00139	118,8392	11500,88	196,483	236,1	65,49845	14,94698	1,503404
Data-113	0,002541763	0,004594	0,002133	93,6468	13035,79	136,8895	162,1554	45,63061	19,8558	1,97052
Data-114	0,001887455	0,0040755	0,002556	115,8889	11500,88	200,326	222,3532	66,80865	14,2574	1,449858
Data-115	0,00232861	0,0043708	0,007931	112,1495	9076,114	198,7299	203,227	66,2579	17,80307	1,962804
Data-116	0,01160899	2E-07	0,004712	90,71189	11986,06	186,4233	227,3387	62,20159	15,6422	2,147693
Data-117	0,011006915	3,995E-06	0,005115	109,8334	11500,88	170,0393	174,3902	56,69957	12,37879	1,440935
Data-118	0,008930196	2,614E-06	0,006139	117,2962	11500,88	183,3044	212,5114	61,08254	14,64065	1,680397
Data-119	0,007371181	3,937E-06	0,006576	107,936	11500,88	138,4767	163,7675	46,2346	10,90725	1,272503
Data-120	0,001243693	0,0040386	0,008171	98,57479	9076,114	120,2637	108,6427	40,09152	11,06877	1,228828
Data-121	0,010480963	4,609E-07	0,00744	118,4185	13216,59	182,1188	233,6262	60,7303	17,01815	2,052421
Data-122	0,007680608	0,0048412	0,006847	107,6524	11500,88	167,5531	185,3708	55,92601	10,03481	1,168378
Data-123	0,011545883	0,0052519	0,006759	114,2011	13216,59	185,7015	221,9212	61,92101	19,11488	1,81517
Data-124	0,006579785	0,0054238	0,006161	93,45403	13216,59	134,4802	190,6697	44,89387	20,69115	2,121595

PD:

PD										
	a	b	c	d	e	f	g	h	i	j
Data-101	0,006461	0,005125	0,007554	2,95E-25	5,55E-14	4,12E-45	1,87E-07	1,36E-45	0,000666	4,23E-05
Data-102	0,00672	0,005911	0,005234	0,028112	59,08716	0,000525	1,075588	0,000173	1,247386	0,132495
Data-103	0,007569	2,15E-15	0,004411	8,63E-20	2,74E-08	5,99E-32	4,47E-09	1,97E-32	0,0003	1,06E-05
Data-104	0,00812	0,006234	0,00521	98,02688	12821,62	185,3117	151,6228	61,76415	16,12688	1,783984
Data-105	0,008217	0,006633	0,007502	74,0501	9051,113	144,4162	128,1069	48,28727	14,9238	1,617648
Data-106	0,008101	0,00664	0,007365	101,3857	12821,62	201,0549	150,3498	67,16886	18,70267	2,025453
Data-107	0,008236	0,006851	0,006879	68,76772	9051,113	133,6393	122,8042	44,54395	15,32709	1,655917
Data-108	0,008168	0,006463	0,007361	101,758	12821,62	199,6292	142,6046	66,69394	19,9277	2,160945
Data-109	0,008171	0,006433	0,00739	70,86346	9051,113	143,0526	125,455	47,68104	14,85286	1,604919
Data-110	0,0065	0,007006	0,005627	88,76834	15866,71	78,62755	182,0181	26,10825	26,91921	2,824186
Data-111	0,005746	0,005984	0,006357	138,6439	15459,82	178,3509	202,2983	59,61645	27,30555	2,80783
Data-112	0,004939	0,006266	0,00252	141,7755	15459,82	182,6489	201,7542	60,73051	23,3315	2,369495
Data-113	0,00534	0,006192	0,003233	62,04433	15866,71	37,47785	123,2774	12,5001	27,67601	2,823095
Data-114	0,004771	0,005261	0,003591	154,9964	15459,82	247,3962	208,6005	82,3262	22,49642	2,30222
Data-115	0,005167	0,005794	0,00652	156,8663	12821,62	269,4795	203,6357	89,94096	26,26705	2,818147
Data-116	0,008168	3,42E-10	0,004666	57,15557	10306,56	132,5036	183,9865	44,32452	24,12831	2,899307
Data-117	0,008269	5,72E-08	0,004916	156,447	15459,82	302,9198	190,5089	100,9598	20,02074	2,290769
Data-118	0,008051	2,78E-08	0,005522	152,5061	15459,82	306,4799	206,6165	102,1651	22,96583	2,572589
Data-119	0,00767	5,58E-08	0,005772	155,5821	15459,82	255,7796	184,6065	85,38012	17,90268	2,062851
Data-120	0,004056	0,005194	0,00665	147,2193	12821,62	216,5768	146,4982	72,18335	18,14206	2,000368
Data-121	0,006764	1,43E-09	0,006252	148,0727	13860,85	306,8577	207,0241	102,2741	25,55528	2,868054
Data-122	0,007757	0,00662	0,005924	155,4205	15459,82	300,7667	196,0586	100,2942	16,58247	1,911801
Data-123	0,0075	0,007264	0,005875	111,7722	13860,85	129,8691	177,3813	43,29938	12,00004	1,023723
Data-124	0,00742	0,007479	0,005535	61,71246	13860,85	35,16722	146,9591	11,79242	27,9983	2,893407
Data-125	0,008252	0,007527	0,005588	115,3607	15866,71	133,8289	178,9577	44,6192	27,54812	2,877237

4. Menghitung perkalian likelihood

	Healthy	PD
Data-101	1,52031E-95	0
Data-102	0,002406252	3,321E-47
Data-103	3,25785E-83	0
Data-104	6725808963	1,893E+13
Data-105	301281767,2	1,378E+12
Data-106	5,50499E+11	3,695E+14
Data-107	1582934740	3,028E+12
Data-108	9,77431E+11	1,666E+15
Data-109	1080281884	2,558E+12
Data-110	3,2146E+15	4,482E+15
Data-111	8,51888E+14	4,828E+16
Data-112	2,07954E+12	3,055E+15
Data-113	7,23545E+13	2,502E+14
Data-114	2,27125E+13	1,811E+16
Data-115	5,70921E+12	3,206E+15
Data-116	120500129,6	4,4869889
Data-117	1199818272	6,309E+10
Data-118	626485635,3	3,37E+09
Data-119	29868073,2	4,118E+09
Data-120	3554318,755	1,226E+13
Data-121	15682717340	5,157E+09
Data-122	2,07655E+11	1,31E+15
Data-123	3,09204E+15	9,204E+12

5. Menghitung Evidence

Evidence = (Prior Healthy * Likelihood Healthy) + (Prior PD * Likelihood PD)

	Evidence
Data-101	7,75357E-96
Data-102	0,001227189
Data-103	1,6615E-83
Data-104	9,28133E+12
Data-105	6,75237E+11
Data-106	1,81314E+14
Data-107	1,48438E+12
Data-108	8,17027E+14
Data-109	1,2539E+12
Data-110	3,83562E+15
Data-111	2,40907E+16
Data-112	1,498E+15
Data-113	1,59498E+14
Data-114	8,88744E+15

6. Menghitung Posterior Probability

Posterior = (Prior * Likelihood) / Evidence

	Healthy	PD
Data-101	1	0
Data-102	1	1,326E-44
Data-103	1	0
Data-104	0,000369577	0,9996304
Data-105	0,000227555	0,9997724
Data-106	0,001548448	0,9984516
Data-107	0,00054386	0,9994561
Data-108	0,000610127	0,9993899
Data-109	0,000439383	0,9995606
Data-110	0,427426719	0,5725733
Data-111	0,018034485	0,9819655

7. Class Decision

pred Class = - jika posterior Healthy > posterior PD, maka classnya ialah Healthy,

- jika posterior PD > posterior Healthy, maka classnya ialah PD

Akurasi = perbandingan pred Class dengan class pada dataset

	Pred Class	True Class	Accuracy
Data-101	Healthy	Healthy	1
Data-102	Healthy	Healthy	1
Data-103	Healthy	Healthy	1
Data-104	PD	Healthy	0
Data-105	PD	Healthy	0
Data-106	PD	Healthy	0
Data-107	PD	Healthy	0
Data-108	PD	Healthy	0
Data-109	PD	Healthy	0
Data-110	PD	Healthy	0

Hasil akhir:

Data-191	PD	PD	1
Data-192	PD	PD	1
Data-193	Healthy	PD	0
Data-194	Healthy	PD	0
Data-195	PD	PD	1
		Accuracy	0,494737

Akurasi = 49,47%

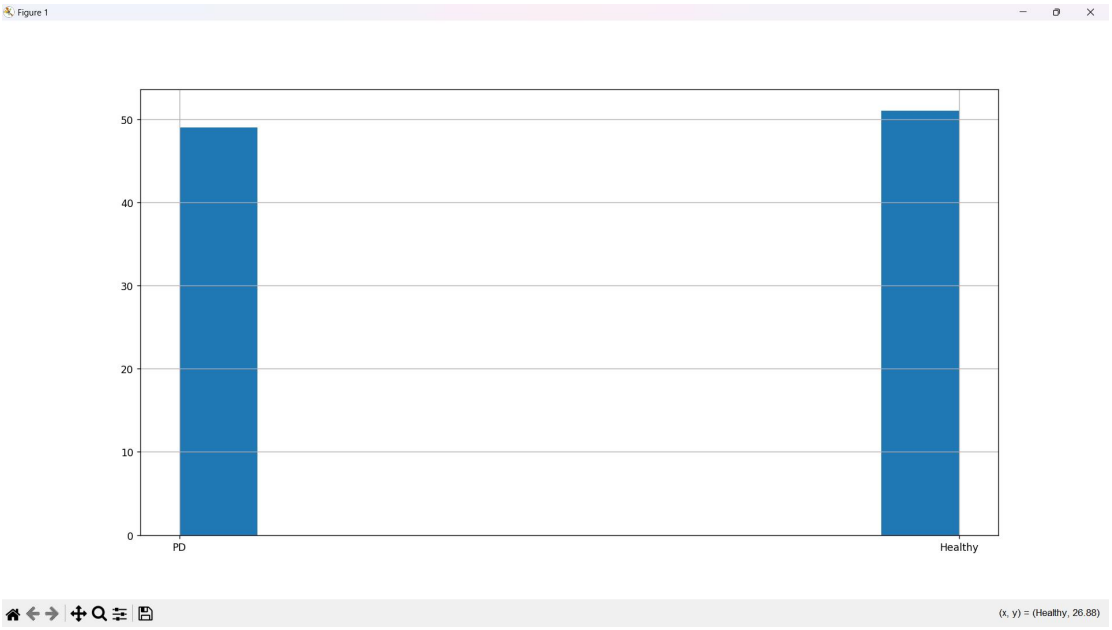
C. Model Machine Learning di Python

Code :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

data = pd.read_excel('parkinsons.xlsx')

data['Class'] = data['Class'].fillna('Healthy')
data = data.replace({',': '.'}, regex=True)

data_train = data.iloc[:100]
data_test = data.iloc[100:]

x_train = data_train.drop(columns=['Class'])
y_train = data_train['Class']
x_test = data_test.drop(columns=['Class'])
y_test = data_test['Class']

y_train.hist()
plt.show()

model = GaussianNB()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy} || Accuracy %: {accuracy * 100}%')
print(f'Error Rate: {1 - accuracy} || Error Rate %: {(1 - accuracy) * 100}%')
```

Hasil training :

plt.show()



Hasil Terminal :

Accuracy: 0.5368421052631579 || Accuracy %: 53.68421052631579%

Error Rate: 0.4631578947368421 || Error Rate %: 46.31578947368421%

```
PS D:\kuliah\sem3\AI\tugas1> python -u "d:\kuliah\sem3\AI\tugas1\tugas1.py"
Accuracy: 0.5368421052631579 || Accuracy %: 53.68421052631579%
Error Rate: 0.4631578947368421 || Error Rate %: 46.31578947368421%
PS D:\kuliah\sem3\AI\tugas1>
```