Manobras no Rocket League

Classificador de Bayes para classificar manobras



Colaboradores: Henrique César Higino Holanda Cordeiro João Pedro Barbosa Marins

Marcelo Arcoverde Neves Britto de Rezende

Vinicius Lima Sá de Melo





Banco de dados



This dataset contains data of players of the game Rocket League, performing different skillshots.

Dataset Characteristics

Subject Area

Associated Tasks

Tabular, Time-Series

Games

Classification

Feature Type

Instances

Features

Real

297

929



https://github.com/Romathonat/RocketLeagueSkillsDetection





Compreendendo os dados

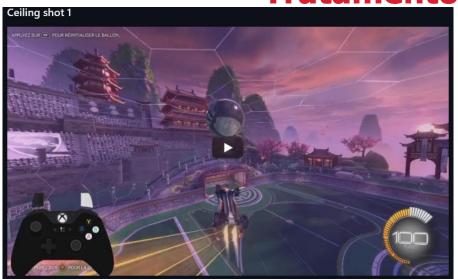
- 1. BallAcceleration;
- 2. Time;
- 3. DistanceWall;
- 4. DistanceCeil;
- 5. DistanceBall;
- 6. PlayerSpeed;
- 7. BallSpeed;

- 8. up;
- 9. accelerate;
- 10. slow;
- 11. goal;
- 12. left;
- 13. boost;
- 14. camera;
- 15. down;
- 16. right;
- 17. slide;
- 18. jump;





Tratamento dos dados











Tratamento dos dados

1	BallAcceleration Time DistanceGall DistanceGall DistanceGall PlayerSpeed BallSpeed up accelerate slow goal left boost camera down right slide jump
2	·
3	1616.7987723122642 8.8 3498.81 2012.98 299.6582700923136 184267.42623178152 99035.84933750001 0 0 0 0 0 0 0 0 1 0 1
4	3108.020106508704 0.13880319999990827 3494.08 2012.98 229.89677966426592 124248.03198843835 102233.87873400871 0 0 0 0 0 1 0 0 1 0 0 1
5	8.8 8.171G16588888881 3494.88 2812.98 237.35859911447462 124248.83198841835 182948.35898954592 0 0 0 0 1 0 0 0 0 1
G	9914.765241818943 0.312509599999985 3500.08 2012.08 151.8809207899399 115248.0160005549 112883.12523135486 0 0 0 0 0 0 0 0 0 0 0
7	5907.747165307177 @.0250190999999994 1521.65 2013 154.55610437637202 125001.25417324318 118790.87230767704 @ @ @ @ @ @ @ @ @ @ 1 @ 1
	0.0 0.83335440000001 3533.4 2013.1 149.40061400847845 134705.20067586182 128733.31106082373 0 0 0 0 0 1 0 0 0 0
9	-581.005135022022 0.0722643090909057 3548.04 2013 148.71516230700000 125937.15823774067 128152.21593480153 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11	647.344733631633 1.64179439999995 3548.84 361 392.9467317865557 12597.1561774657 12879.512746650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12	8.0 1.13501399999998 3170.77 281.82 27.1135139413621 32688.4187138621 11288.7852746512 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13	137.7055214751093 1.493144408N98080 2735.4.2 2011.85 143.67157441280146 15003.693080158 14447.5540502775 8 8 8 8 8 1 8 8 1 8 8 1 8 8
14	740.4545964539262 1.5973344999999915 3781.4 2013.05 139.33989156437713 146000.56629595185 145218.08866678077 0 0 0 0 0 0 0 0 0 0 0 0
15	B60.1809881718011 1.880477400000000 3860.50 2011.14 102.75102423203M 141352.01552631478 140078.11455445466 0 0 0 0 0 0 0 0 1 0 1
16	4154.8881744841675 1.944579698080923 3068.51 7813.60 175.59018109877 158777.18521827856 150233.002828293903 0 0 0 0 0 0 0 0 0 0 0 0
17	2009.1541988880002285 2.0140244800000003 4010.02 2013 158.64181344071373 157403.8236410746 153142.19780074806 0 0 0 0 0 0 0 1 1 0
18	-648.5783137179969 2.152914480000003 3967.36 2013.01 145.82514083655127 148187.79233121735 152493.61949683006 0 0 0 0 0 0 0 0 1 0 0
19	-1554.7753860008739 2.2571087 3025.57 2013.11 149.18333480542188 146921.80253794874 150918.8441691292 0 0 0 0 0 0 1 1 0
20	-875.1533875232292 2.326553499999997 3889.69 2813.16 151.38884508183917 151468.4185953887 150063.69882162596 0 0 0 0 0 0 0 0 0 1 1 1
21	8347.386885538548 2.4654599989999926 3864.64 2813.16 189.92224777524681 158823.87978197736 158411.8776871365 0 0 0 0 0 0 0 0 0 0 1 0 1
22	8.8 2.67379959999999 3838.95 2813.08 167.63235188948445 177824.81111414862 162625.25464392855 6 8 6 8 6 8 6 8 8 1 1 6
23	8.8 2.741349899999966 3829.96 2813.15 153.46173848859412 172578.00118960417 162273.94787487593 8 8 8 1 8 8 8 8 8 8 1 8
24	-483.7625688955689 2.8127882599999995 3848.91 2013.1 144.31392205882284 167369.98468572655 161870.18458597997 0 0 0 1 0 0 0 0 0 0
25	8.0 3.264127800000007 3025.61 2010.07 215.10507508715305 150052.87658673203 171985.85951971526 0 1 0 0 0 0 1 0 0 0 0
26 27	-1125.6730544665004 3.333578200000016 3045.12 1986.9 286.44352000263747 148011.0124503802 170850.MF746524876 0 1 8 8 8 8 8 8 8 8 8
28	8.8 1.4377575 3078.36 1040.75 238.540115875012 107271.870734541 160158.0158471860 0 1 0 0 0 0 1 0 0 0 0 1
20	33485.16487851564 3.7156343999999994 4815.56 1937.96 238.2335688425657 192221.3536284176 282563.28871778193 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0
20	-115.1745737551 4.36466400000000 3847.44 197 198.54 19871004745 193543, 8187404100 197615, 2780171051 0 0 0 0 0 0 1 0 0
21	-195880.9238225193 4.618612299999999 1892.64 2005.89 5586.826337731952 188825.315446058 1762.0 0 0 1 0 0 0 0 1 1 0
12	2163.8 4.65333758989893 3894.8 1997.84 5743.826949619865 191541.33898625281 3925.8 8 8 8 8 8 8 8 8 8 8 8 1 1
33	-3648.0 4.7922093 3784.07 1854.1100000000001 5078.528854085082 159439.70030825408 277.0 0 0 0 0 0 0 0 1 0 1
34	8.8 5.870184400000002 1758.84 1507.11 1868.428914811868 140058.6889151451 620.0 0 0 0 0 0 1 0 0 1 0 0
35	4
36	8.8 6.8 39.7E0000000002 1784.5 193.42114749943976 141312.10493727177 13685.E39512361366 0 0 0 0 0 1 0 0 1 0 1
37	127859.248789516E 0.13895888988888124 36.848888888888124 1592.63 110.3152989542165 138682.70127757695 168745.08024197817 0 1 0 0 1 0 1 0 1
36	-7285.559298530626 0.27795000000000014 36.0300000000002 1379.34 390.3187359605085 135231.76387594742 153539.52095144754 0 1 0 0 1 0 0 1 0 0
19	-3283.64585498847 0.34740000000000 35 1258.04 448.38587101655675 130547.57683647799 150255.87508546787 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
40	-8258.080123629454 8.5211131808080809 36 1838.09 654.2488014778534 124143.01988681332 141997.86596283762 0 8 0 0 0 0 0 0 1 1 0
61	-1507.0016986954608 8.5558459000000013 35 1818.69 693.443773423628 124143.01980681132 146488.56426414216 0 0 0 0 0 0 0 0 0 1 1 1
62	-6167.428085113162 8.6947409990909091 36 825.7 855.702302464434 120275.37804480833 134326.135279829 8 8 8 8 8 8 8 8 8 8 8 8 1 8 1
43 44	-0131.607011508541 0.0378457900000001 31.46000000000000 467.73 1105.0321865831016 143714.46008575618 125192.41736743846 0 0 0 0 0 0 0 0 1 1 1 -5652.7678735156460 1.11151790000000 28.60000000000018 200.8100000000000 1226.7765873631815 137862.5180000054 110519.66070208100 0 0 0 0 0 0 1 1 0
45	-2012. EMPRICASION 1.1111/16/16/000000018 156.179999999991 185.1800000000001 1279.EMPRICASION 12890.111111111111111111111111111111111111
46	-M12.20192485507 1.09157100000000 201.650099999999 11.1000000000005 1166.700747772886 11810.0.6260111044 10923.58574095093 0 0 0 0 0 1 0 1 0 0
67	-1387.515594138248 1.632439999999994 449.349999999999 57.7099999999999 51.87.7111975981368 19014.8499397290 187441.86814462869 0 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
68	8.8 1.771334299999995 528.5 74.45800090000005 1186.1517511508542 141109.3214299704 100188.97233705579 8 8 8 1 8 1 8 8 1 8 1
49	-768.7348040413137 1.8751700000000016 594.77 133.150000000001 1875.228718413971 157880.02736571842 185428.21753056147 0 0 0 0 0 0 0 0 1 0 1
50	8.8 2.153370000000024 923.400000000000 364.4500000000000 942.988220623393 185546.6400719776 106137.23123190773 0 8 0 0 0 0 0 0 1 0
51	995.7552161812897 2.2576000000000036 1886.4899999999999 319.3199999999994 897.9048651165673 187079.79582861472 187132.986484888992 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
52	1618 .8805058287777 2.106529999999985 1293.56 643.8499999999995 721.7616621849625 193245.83884496587 188761.87617918975 8 8 8 8 1 8 8 1 8 8 1
53	555.6245515600005 2.4312540000000027 1391.56 441.04090909090905 756.4126074027460 193245.0300400507 109317.50073056004 0 0 0 1 0 1 0 0 1 0 0
54	3914.513681679179 2.6M99670000001 1512.69 593.06 647.7473796164674 199293.84458353702 113232.01381234902 0 0 0 0 0 0 0 0 0 0 0 0
55	8.0 2.636c156090909095 1615.8400000000001 676.23 517.7746206603794 208140.3824009171 113232.01381214002 0 1 0 0 0 1 0 0 1 0 0
56	1638.2252068716432 2.674364699999998 1615.E400000000000 676.23 577.6872650166371 208140.3824000171 114862.23918842066 0 1 0 0 0 1 1 0 1 0 0
57	861.0400030541E73 2.7800E51000000004 1712.12 7E1.05 500.037E23007E5615 211B5E.7501E511765 115725.2E50123E4E5 0 1 0 0 0 0 1 0 1 0 0
58 59	2259.4865814411466 2.811290888888802 1888.81 852.350909090909 581.4285441615784 215619.05126766262 118984.60582681631 0 1 0 0 0 1 0 1 0
60	2819.2130825895135 2.882710000000022 1985.98999999999 1087.35 482.04145817553933 219391.4822321040 121803.08825942584 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0
61	10174. 1021M1407880 1.57773100909900007 2048.00 1003.00 Gar 1807086704485 107487.77841200231 107145.8801238857 0 0 0 0 0 0 0 1 1 0
62	4168.7905404E21577 3.681562000000002 2715.8 1979.61 578.512859440007 185483.75166206202 151514.59827316826 8 8 8 1 8 8 8 8 1 1 8
63	-57002.1755303237 4.48022000000000 2818.04 2012.00 626.000708181775 52427.181608000400 91822.42343186056 1 0 0 1 0 0 0 0 1 1 0
64	-3877.6756885318006 4.540685890000000 3788.880000000007 2012.90 788.85010377115013 50274.8007000718015 91744.74783313285 0 0 0 1 0 0 0 0 1 0
65	-34344.262241580314 5.85928000000003 3234.08 1921.83 882.5080153175262 187877.68040702395 67408.48559172255 0 0 0 1 0 0 0 0 1 0 0
66	1973E.54411942353 6.216555300000003 3569.17 1998.5 653.6697557688611 97612.26000971279 87139.02971114608 0 0 0 0 0 0 0 1 1 0
67	-31820.322441363565 6.56384536988884 3913.26 2812.67 418.58621751578393 186127.59448489455 55318.78726978352 0 0 0 1 0 0 0 0 1 1 0
68	-12858.669283495893 6.841654400000003 1848.51 2013 134.21257570216415 114130.79290883771 43268.038007286625 0 0 0 1 1 0 0 0 1 1 0
69	-3436.4560801413518 6.04582239999997 3765.68 2013 370.20247581560017 121417.29103385564 39831.58732734527 0 0 0 1 1 0 0 0 1 1 0
70	-448.18196884517674 6.98854359999998 1765.68 2013 375.7814684688426 121417.20103185564 30301.285427186896 0 0 0 1 1 0 0 0 1 1 1
71	-18170-205427108005 7.258145100800002 1520.539099090907 2013 4743.758018168163 145808.6186375825 1221.0 0 0 1 0 0 0 0 0 1 1 1
72	1623.0 7,293664689808993 3623.85 2013 4619.643434695823 151845.19755585483 2804.0 0 0 0 0 0 0 0 0 0 0 1 0 1
73	-218.8 7.4110021099009000 3464.86 2013 4705.634637301285 162663.03700176922 2634.8 0 0 0 0 0 0 0 0 0 0 0 0 0 0
74	-2587.0 8.161284100000003 3228.0299999999997 2013 3308.56142797721353 162648.98600657616 47.0 0 0 0 0 0 0 0 1 1 0

- 1. BallAcceleration Útil;
- 2. Time Útil;
- 3. DistanceWall Útil;
- 4. DistanceCeil Útil;
- 5. DistanceBall Útil;
- 6. PlayerSpeed Útil;
- 7. BallSpeed Útil;
- 8. up Útil;
- 9. accelerate Útil;

- 10. slow Inútil;
- 11. goal Inútil;
- 12. left Inútil;
- 13. boost Útil;
- 14. camera Inútil;
- 15. down Útil;
- 16. right Inútil;
- 17. slide Inútil;
- 18. jump Útil;





Análise exploratória dos dados

```
for i in range(len(dataset)):
    if pd.isna(dataset.iloc[i,1]) or i == len(dataset) - 1:
        if qtdLinhas > 0:
            matrizFinal.append([dataset.iloc[i-qtdLinhas:i,0].mean(), timeFinal, dataset.iloc[i-qtdLinhas:i,2].var(),
                                dataset.iloc[i-qtdLinhas:i,3].var(),
                                dataset.iloc[i-qtdLinhas:i,4].var(), dataset.iloc[i-qtdLinhas:i,5].mean(),
                                dataset.iloc[i-qtdLinhas:i,6].mean(), dataset.iloc[i-qtdLinhas:i,7].sum(),
                                dataset.iloc[i-qtdLinhas:i,8].sum(),
                                dataset.iloc[i-qtdLinhas:i,10].sum(), dataset.iloc[i-qtdLinhas:i,10].sum(),
                                classel)
        classe = dataset.iloc[i,0]
        qtdLinhas = 0
        timeFinal = 0
        qtdLinhas += 1
        timeFinal = dataset.iloc[i,1]
finalFeatures = features + ["classe"]
processedData = pd.DataFrame(matrizFinal,columns=finalFeatures)
processedData.head()
```





Análise exploratória dos dados

	BallAcceleration	Time	DistanceWall	DistanceCeil	DistanceBall	PlayerSpeed	BallSpeed	up	boost	down	jump	classe
0	-3853.867885	5.070104	3.059738e+04	6917.679671	3.546207e+06	152961.454638	129791.809510	0.0	7.0	0.0	9.0	6.0
1	-848.813985	8.161284	1.981624e+06	561237.455429	1.326933e+06	147893.764224	96963.150194	1.0	8.0	0.0	10.0	-1.0
2	-1264.766385	8.130110	2.308898e+06	564614.846333	1.084948e+06	147510.308898	96556.211527	0.0	10.0	0.0	7.0	-1.0
3	-2133.341430	3.610510	9.436052e+05	176821.531636	1.358482e+05	171583.561188	121592.704957	0.0	7.0	0.0	11.0	-1.0
4	-7148.505767	3.681763	1.537356e+04	857.226355	1.308056e+06	163413.423195	149543.364532	1.0	8.0	0.0	5.0	-1.0

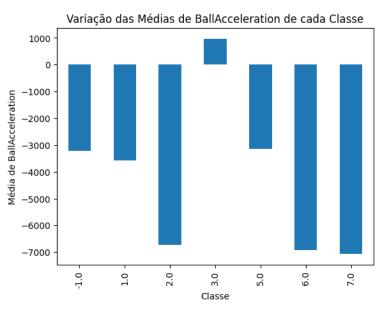
	BallAcceleration	Time	DistanceWall	DistanceCeil	DistanceBall	PlayerSpeed	BallSpeed	ир	boost	down	jump	classe
count	298.000000	298.000000	2.980000e+02	2.980000e+02	2.980000e+02	298.000000	298.000000	298.000000	298.000000	298.000000	298.000000	298.000000
mean	-4452.819314	4.074706	2.543351e+07	1.044325e+05	2.009341e+06	154533.729820	123599.619123	0.929530	6.077181	0.238255	9.436242	3.268456
std	4767.413370	1.704921	3.068464e+08	2.082541e+05	1.705919e+06	30163.620006	42925.508545	1.722812	4.194443	1.038135	6.683133	2.605694
min	-26128.652035	1.009000	0.000000e+00	5.686867e-26	3.321846e-02	400.374959	0.000000	0.000000	0.000000	0.000000	0.000000	-1.000000
25%	-6807.479333	2.785029	3.238837e+04	1.177797e+03	7.241118e+05	140020.088387	108000.574347	0.000000	3.000000	0.000000	4.000000	2.000000
50%	-4064.452173	3.940157	1.060573e+05	3.328337e+03	1.542971e+06	155827.752649	129459.608099	0.000000	6.000000	0.000000	9.000000	3.000000
75%	-1295.464847	5.015213	1.340235e+06	1.520008e+05	3.026675e+06	172907.893940	147342.590456	1.000000	9.000000	0.000000	13.000000	6.000000
max	11599.404268	13.470363	4.805408e+09	2.203612e+06	7.306665e+06	228114.832665	210877.285380	8.000000	20.000000	9.000000	37.000000	7.000000

cin.ufpe.br

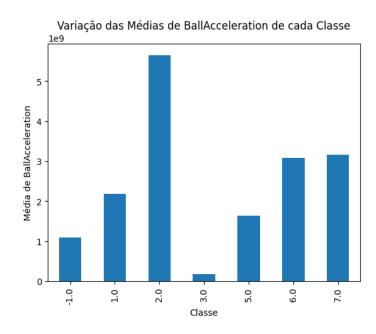




Análise explanatória dos dados



Média das médias



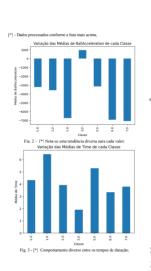
Média das Variâncias

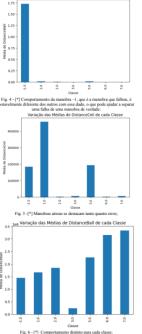


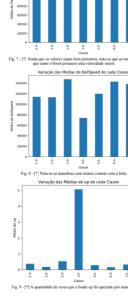


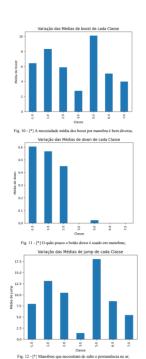
Análise exploratória dos dados

- 1. BallAcceleration Média;
- 2. Time Valor Final;
- 3. DistanceWall Variância;
- 4. DistanceCeil Variância;
- 5. DistanceBall Variância;
- 6. PlayerSpeed Média;
- 7. BallSpeed Média;
- 8. up Somatório;
- 9. boost Somatório;
- 10. down Somatório;
- 11. jump Somatório;







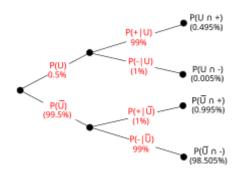


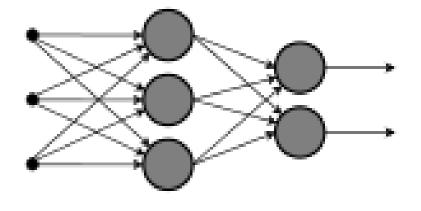




Classificador ingênuo de Bayes

$$P(A/B) = \frac{P(B/A)xP(A)}{P(B)}$$









Resultados

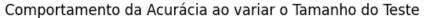
```
Y = processedData.classe
X = processedData.drop(['classe'], axis=1)

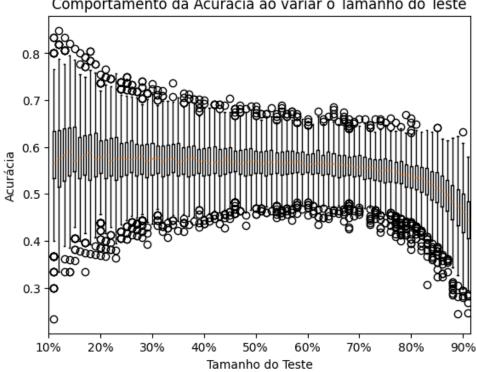
for test_size in test_sizes:
    acc_list = []
    for i in range(1000):
        X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=test_size)
        model = GaussianNB()
        model.fit(X_train, Y_train)
        Y_pred = model.predict(X_test)
        acc_list.append(accuracy_score(Y_test,Y_pred))
    results.append(acc_list)
```





Resultados









Conclusão

- Eficiência do Bayes Ingênuo
- Relevância
- Comparação

