

Detecção de Habilidades em Rocket League

Identificação de manobras realizadas no jogo Rocket League

Henrique César Higino Holanda
Cordeiro

João Pedro Barbosa Marins

Marcelo Arcoverde Neves Britto
de Rezende

Vinicius Lima Sa de Melo

Centro de Informática (CIn) –
Universidade Federal de
Pernambuco (UFPE)

Centro de Informática (CIn) –
Universidade Federal de
Pernambuco (UFPE)

Centro de Informática (CIn) –
Universidade Federal de
Pernambuco (UFPE)

Centro de Informática (CIn) –
Universidade Federal de
Pernambuco (UFPE)

Recife, Brasil
hchhc@cin.ufpe.br

Recife, Brasil
jpbm@cin.ufpe.br

Recife, Brasil
manbr@cin.ufpe.br

Recife, Brasil
vlsm@cin.ufpe.br

Abstract—Esse trabalho apresenta o resultado de projeto para a classificação de dados relacionados a movimentos em um jogo chamado Rocket League. Para analisar e categorizar os dados coletados de um banco de dados específico, fizemos um modelo preditivo em Python usando o classificador ingênuo de Bayes. O projeto visou identificar padrões e tendências nas movimentações de jogadores ao realizar uma manobra. Os resultados obtidos mostram a limitação do classificador ingênuo de Bayes quanto a classificação de dados mais complexos.

Keywords—Rocket League, Classificador ingênuo de Bayes, Futebol, Video Game, Previsão.

I. INTRODUÇÃO

Jogos, sejam eles virtuais ou físicos, são criados sempre com o intuito de entreter, mas podem conter um caráter mais competitivo ou mais casual. Quando se trata de um jogo mais competitivo, a questão do desempenho sempre é levantada tanto para aqueles que seguem a carreira profissional quanto para os jogadores casuais que querem mostrar seu nível para os amigos ou rivais.

O jogo Rocket League não é diferente. Estratégias, leitura de situação, pensamento rápido e assertivo são algumas das habilidades que um bom jogador deseja sempre aperfeiçoar. Possuir uma base de referência das manobras realizadas e como melhorar sua execução é um bônus desejado.

Esse jogo consiste em uma partida de futebol com carros que podem se movimentar em três dimensões, usando propulsores, saltos e giros. As manobras são ações específicas que os jogadores realizam para atingir um objetivo, como driblar, chutar, defender ou se posicionar. Cada manobra requer um conjunto de movimentos e uma situação adequada para ser executada com sucesso.

II. OBJETIVO

Por meio da análise de um banco de dados coletado e disponibilizado na *IEEE Conference on Games 2020*, que registrou a movimentação de jogadores em intervalos de tempo realizando 7 manobras, nosso objetivo é classificar cada

manobra que o jogador realizou. Utilizando informações como a posição relativa da bola com o jogador, a parede e o teto, além de várias outras disponíveis no banco de dados, com o classificador ingênuo de Bayes, classificaremos uma série de manobras, ajustando os atributos conforme o necessário, e nos certificaremos dos resultados através de uma validação cruzada.

III. JUSTIFICATIVA

A análise e classificação dos estilos de movimentação e manobras dos jogadores são uma peça fundamental quando se trata de melhorar o desempenho de um jogador e criar boas estratégias. Ao utilizar o classificador ingênuo de Bayes, podemos identificar alguns padrões de comportamento e aprimorar as táticas utilizadas durante o jogo.

Portanto, a contribuição que traz para uma experiência mais competitiva e desafiadora é enorme. Além de auxiliar na criação de novas estratégias e diversificar mais as já existentes, melhorará também a experiência de quem assiste aos jogos.

IV. METODOLOGIA

Almejamos, nesse projeto, classificar uma manobra do banco de dados através do classificador ingênuo de Bayes. Portanto devemos fazer uma análise explanatória dos dados para separar quais informações são úteis e quais não são, além de ajustá-las para que o classificador seja capaz de funcionar, uma vez que a forma como os dados estão dispostos não possibilita um uso imediato dele. Esperamos um desempenho baixo, devido ao nível de complexidade dos dados, e faremos uma comparação de resultados de desempenho da nossa implementação com outros modelos de aprendizado de máquina para classificar. Faremos nossa implementação completa na plataforma *Google Colab* completamente em *Python* com o uso das bibliotecas *Numpy*, *Pandas*, *Scikit Learn* e *Matplotlib*.

A. Banco de Dados

O banco de dados que utilizaremos é o *Rocket League Skills Detection*, disponibilizado por Romain Mathonat em seu Github [1], que fornece 298 manobras, algumas observações pessoais do trabalho dele e um coletor de dados para aumentar o banco, mas que não foi utilizado.

B. Pré-processamento dos Dados

De acordo com o repositório original, não há qualquer dado faltando, o que verificamos através de um simples programa em Python que leu cada linha e verificou se havia espaços vazios. Após uma breve análise, fomos capazes de concluir a utilidade dos 18 atributos da seguinte maneira:

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;

C. Teorema de Bayes

O teorema de Bayes é um princípio matemático que permite calcular a probabilidade de um evento a partir do conhecimento prévio de certas condições relacionadas. O teorema se baseia na ideia de que a probabilidade de um evento depende não apenas da frequência com que ele ocorre, mas também de quão provável é que ele ocorra dada a ocorrência de outro evento, expresso da seguinte forma:

$$P(A | B) = P(B | A) \times \frac{P(A)}{P(B)}$$

Com $P(A|B)$ é a probabilidade posterior de A dado B, $P(B|A)$ é a probabilidade condicional de B dado A, $P(A)$ é a probabilidade priori de A, e $P(B)$ é a probabilidade marginal de B.

O teorema de Bayes permite atualizar as probabilidades à medida que novas informações são obtidas, e assim refinar as estimativas sobre a incerteza de um fenômeno. Devido sua simplicidade e praticidade, possui uma fácil implementação e boa precisão quando se trata de atributos discretos.

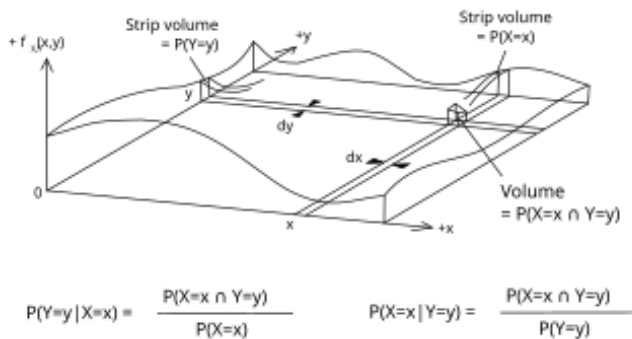


Fig. 1 – Teorema de Bayes representado tridimensionalmente.

D. Classificador Ingênuo de Bayes

O classificador ingênuo de Bayes é um método de aprendizagem supervisionada baseado na aplicação do teorema de Bayes, que calcula a probabilidade de uma classe dado um conjunto de atributos. A palavra "ingênuo" se refere à suposição de que os atributos são independentes entre si, ou seja, não há relação de dependência ou correlação entre eles.

Essa simplificação permite que o modelo seja de fácil computação e bastante preciso em certos casos, além de lidar bem com grandes quantidades de dados e recursos. O classificador ingênuo de Bayes também é robusto a ruídos e outliers, e pode lidar com dados incompletos ou ausentes.

No entanto, a suposição de independência é um problema nativo desse modelo, e pode levar a resultados imprecisos ou enviesados em alguns casos. Por isso, casos com dependência temporal, por exemplo, costumam ter uma baixa eficiência de previsão ao usá-lo. Dependendo do tipo de distribuição dos dados, existem diferentes variações do modelo, como gaussiano, polinomial, bernoulli e categórico, que adaptam a fórmula do teorema de Bayes aos diferentes cenários.

A eficácia do classificador ingênuo de Bayes em modelos preditivos depende de vários fatores, como a qualidade e a quantidade dos dados, a relevância e a independência dos atributos, a escolha do modelo adequado, e a validação dos resultados. Quando esses aspectos são considerados, o classificador ingênuo de Bayes pode fornecer resultados rápidos, simples e confiáveis, superando até mesmo modelos mais complexos e sofisticados.

V. ANÁLISE EXPLORATÓRIA DOS DADOS

Antes de realizar o treinamento, é importante realizar uma análise exploratória dos dados, que consiste em examinar os dados disponíveis, verificar suas características, distribuições, relações, inconsistências, e identificar possíveis padrões ou tendências. Essa etapa é fundamental para compreender melhor o problema, selecionar os atributos relevantes, e escolher a técnica mais adequada para o aprendizado de máquina.

No nosso caso, realizamos uma análise exploratória dos dados do banco de dados que contém informações sobre as posições, velocidades, acelerações e alguns comandos especiais dos jogadores e da bola durante diversas jogadas. Nessa análise, observamos os tipos de manobras realizadas pelos jogadores e como elas se relacionam com os atributos dos dados. Alguns dados foram previamente processados nesse processo, pois, ao analisarmos os vídeos que mostravam cada uma das manobras, alguns dados foram descartados por serem considerados inúteis ou redundantes para a classificação das manobras, como o a ocorrência de gol ou se o carro deu um "slide" (refere-se à quando o jogador não é capaz de controlar o movimento do carro sem usar o "boost").

Os dados são coletados enquanto a manobra é executada, logo eles possuem uma certa dependência temporal, criando a necessidade de ajustarmos a leitura de cada atributo. Para isso utilizamos atributos estatísticos que julgamos serem os que mais se encaixava:

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;

[*] - Dados processados conforme a lista mais acima;

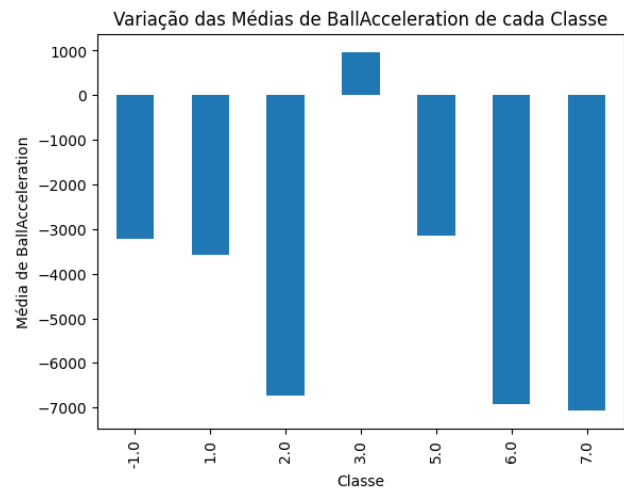


Fig. 2 – [*] Nota-se uma tendência diversa para cada valor;

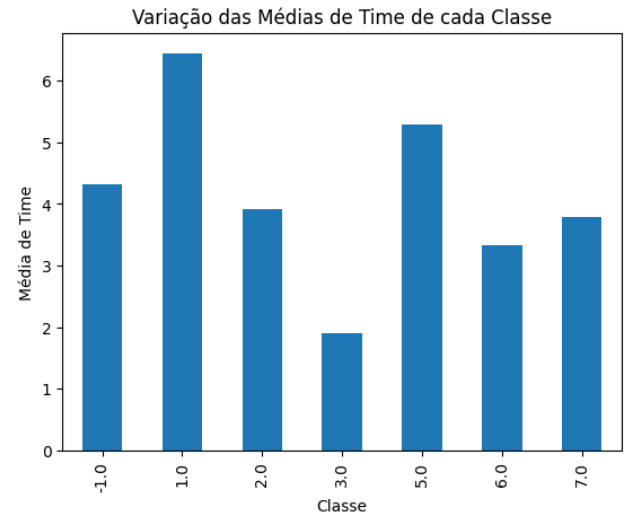


Fig. 3 - [*] Comportamento diverso entre os tempos de duração;

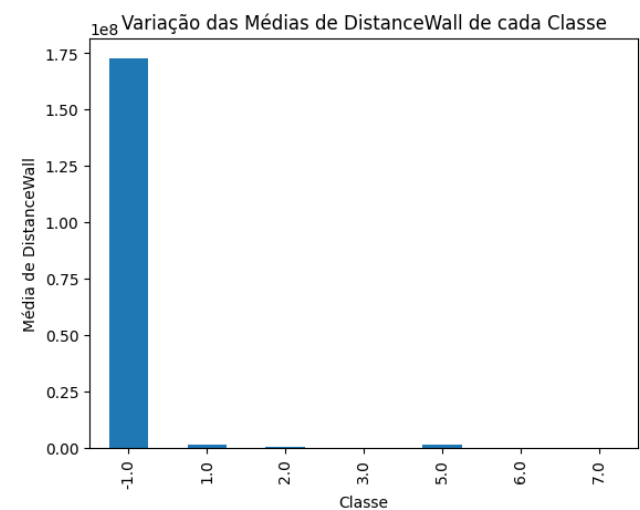


Fig. 4 - [*] Comportamento da manobra -1, que é a manobra que falhou, é notavelmente diferente dos outros com esse dado, o que pode ajudar a separar uma falha de uma manobra de verdade;

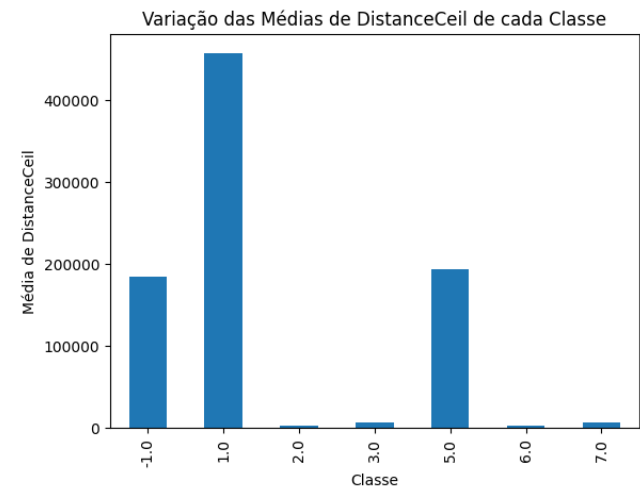


Fig. 5 -[*] Manobras aéreas se destacam tanto quanto erros;

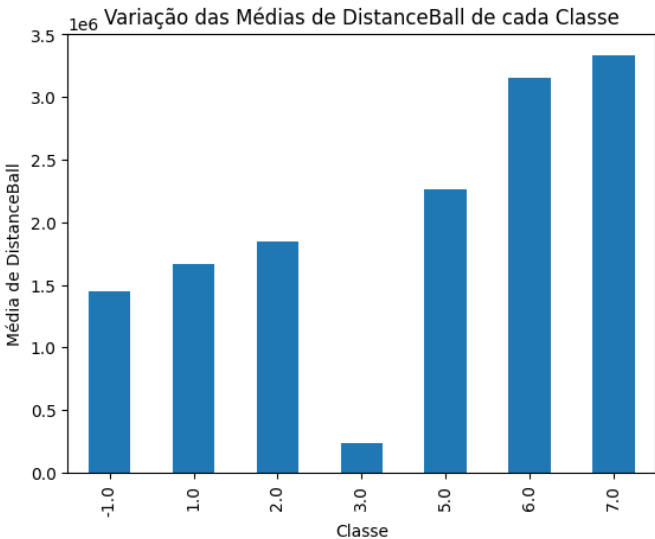


Fig. 6 - [*] Comportamento distinto para cada classe;

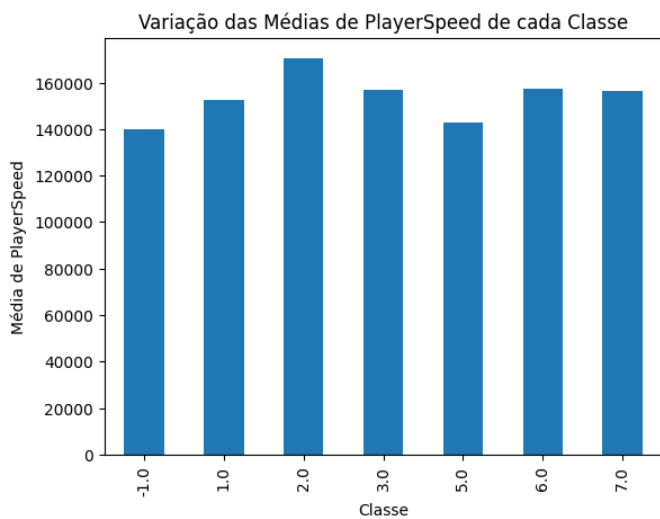


Fig. 7 - [*] Ainda que os valores sejam bem próximos, nota-se que as manobras que usam o boost possuem uma velocidade maior;

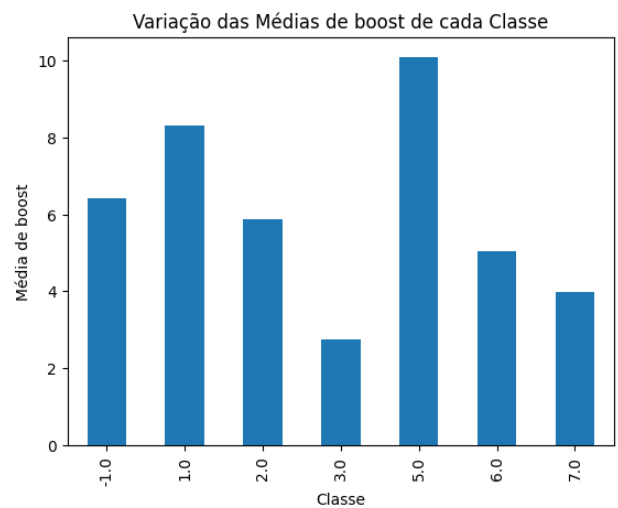


Fig. 10 - [*] A necessidade média dos boost por manobra é bem diversa;

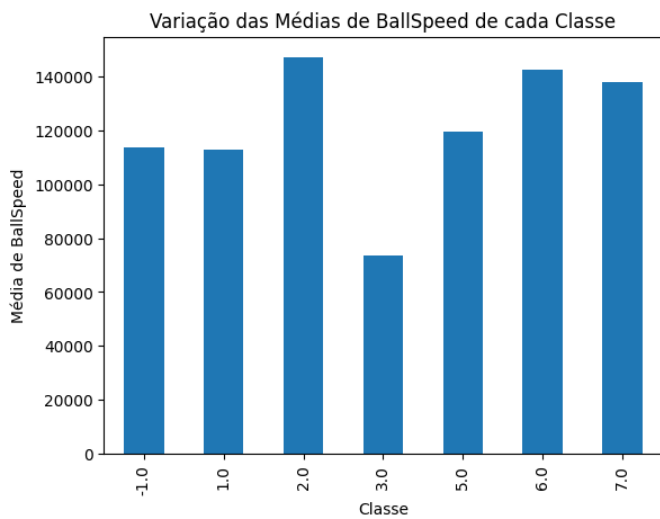


Fig. 8 - [*] Nota-se as manobras com menos contato com a bola;

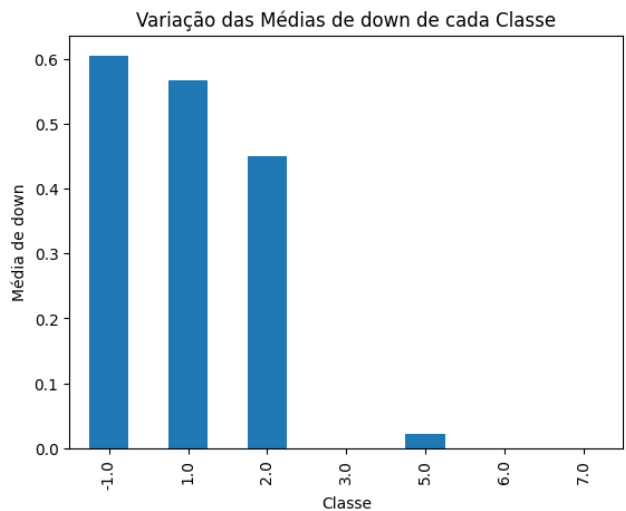


Fig. 11 - [*] O quão pouco o botão down é usado em manobras;

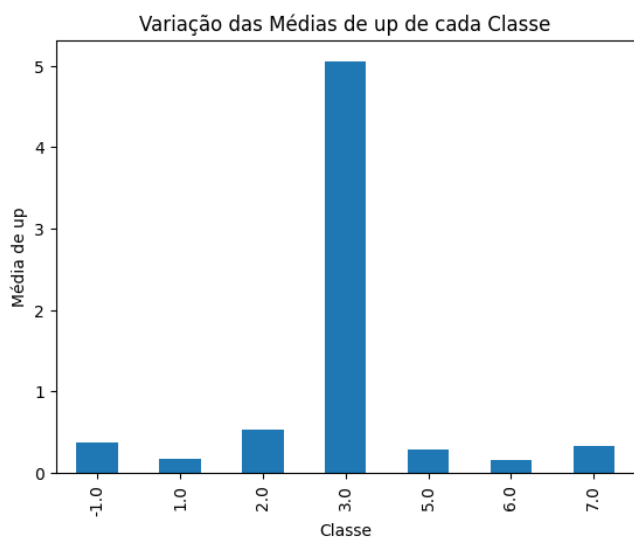


Fig. 9 - [*] A quantidade de vezes que o botão up foi apertado por manobra;

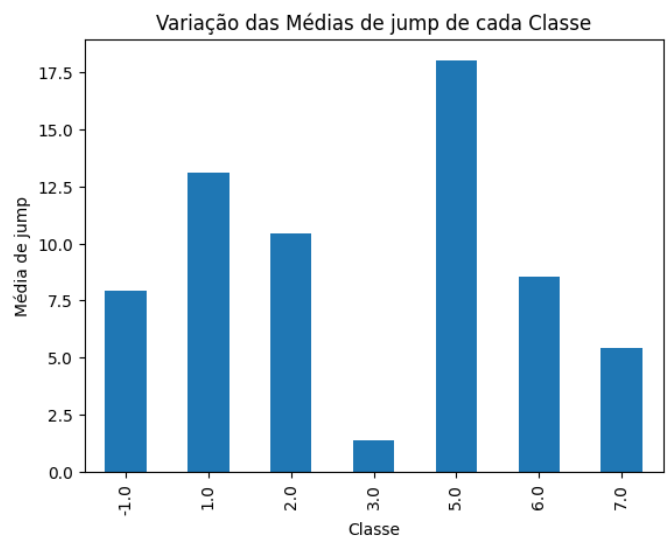


Fig. 12 - [*] Manobras que necessitam de salto e permanência no ar;

Diante da análise dos atributos acima, podemos concluir que realizamos uma escolha bastante assertiva para cada valor e maneira de representá-lo, pois todos apresentam uma boa variação média entre as classes possíveis. Então, com todos os dados devidamente separados e ajustados para valores discretos, podemos, enfim, usar o Classificador Ingênuo de Bayes, optando pelo gaussiano devido a característica geral dos valores.

VI. ANÁLISE DE RESULTADOS

Após a implementação do código, utilizamos o método de validação cruzada para ter uma melhor amostragem dos resultados e os colocamos em uma série diferente de valores de treino para entendermos o comportamento.

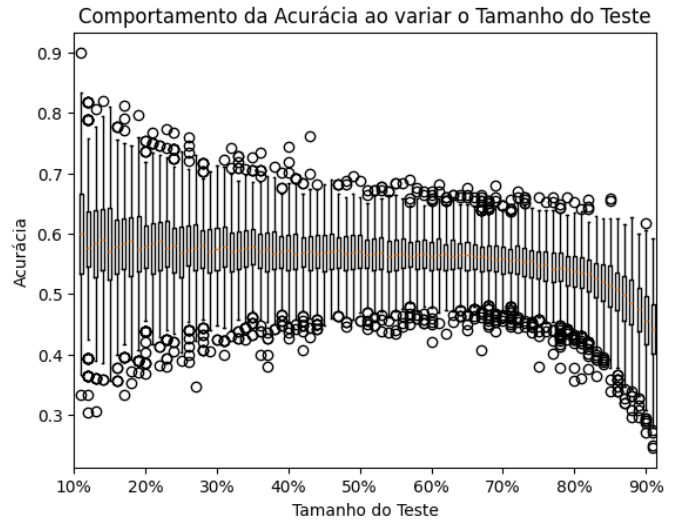


Fig. 13 - Distribuição cariando de 1% de 10% até 90%, operados 1000 vezes em cada caso;

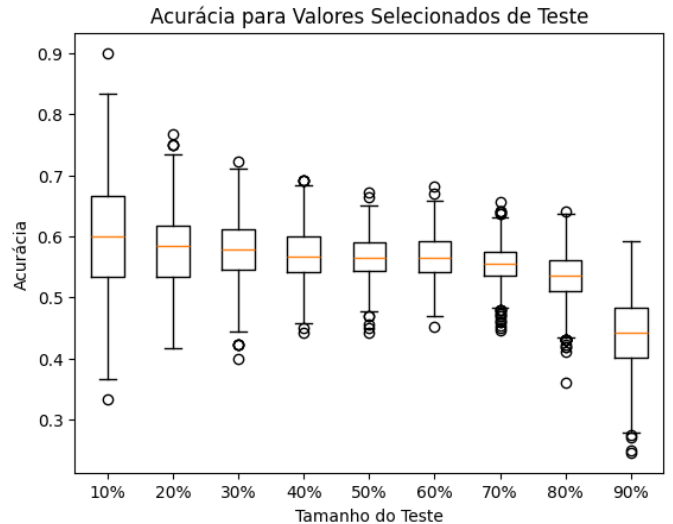


Fig. 14 – Valores mais filtrados de 10% em 10%;

Para o nosso banco de dados, a precisão fica máxima com 70% da base para treino, mas com uma precisão inferior a 60% de acerto médio, um desempenho fraco, como previsto.

Analisando o banco de dados do próprio Romain Mathonat, podemos encontrar também outros modelos de treinamento:

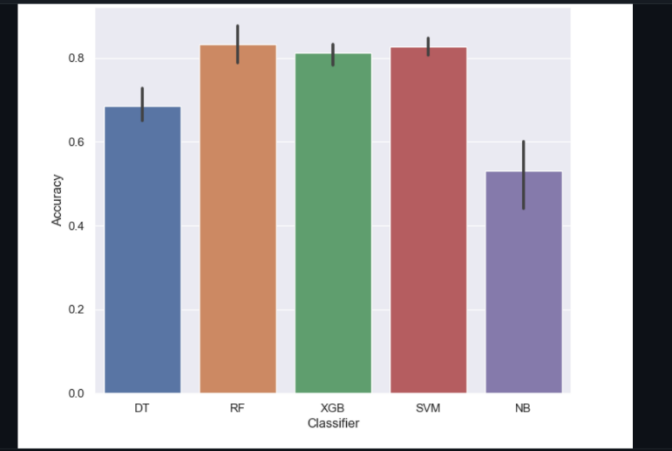


Fig. 15 – Treinamento de Romain Mathonat usando Árvore de decisão (DT), Floresta Aleatória (RF), Reforço de Gradiente (XGB), Máquina de Vetores de Suporte (SVM) e Ingênuo de Bayes (NB);

Portanto, podemos comprovar que o desempenho esperado está de acordo com nosso resultado, além de vermos o quão ineficiente é o classificador ingênuo de Bayes para esse tipo de dado.

VII. CONCLUSÃO

Diante da análise comportamental da classificação que implementamos, torna-se evidente o baixo desempenho desse modelo de treinamento para a classificação de dados com alta complexidade e dependência, mesmo que devidamente tratado para que essa dependência não afete tanto. Podemos perceber também que o tamanho do banco dados que apresenta uma certa limitação também, pois com 298 casos os resultados com menores erros foram os com 70% dos testes para o treinamento.

Porém, mesmo com um baixo desempenho, o classificador de Bayes se mostrou mais eficiente do que o esperado, deixando claro o quão poderoso é essa ferramenta. Classificar com praticidade é um grande bônus para ele, mas devemos sempre lembrar nas limitações inerentes ao modelo.

VIII. REFERÊNCIAS

[1] Repositório GitHub *RocketLeagueSkillsDetection*. Disponível em: <https://github.com/Romathonat/RocketLeagueSkillsDetection>. Acesso em: 01 junho 2024;

[2] *Naive Bayes*. Disponível em: https://scikit-learn.org/stable/modules/naive_bayes.html?authuser=0. Acesso em: 02 julho 2024;

[3] Validação Cruzada. Disponível em: <https://analisemacro.com.br/econometria-e-machine-learning/validacao-cruzada-em-modelos-preditivos-tecnicas-para-dados-ordenados-e-nao-ordenados/>. Acesso em: 15 julho 2024;

[4] Rennie, J. D. M., Shih, L., Teevan, J. e karger, D. R. (2003). *Tackling the Poor Assumptions of Naive Bayes Text Classifies*. In: *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 3, pp. 616-623;