
Seleção de atributos para modelagem preditiva de estratégias em jogos de Pôquer

Otávio Soares Ribeiro



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Monte Carmelo - MG
2021

Otávio Soares Ribeiro

**Seleção de atributos para modelagem preditiva
de estratégias em jogos de Pôquer**

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Prof. Dr. Murillo Guimarães Carneiro

Monte Carmelo - MG

2021

Este trabalho é dedicado aos meus pais, que sempre desejaram para mim o que não tiveram a oportunidade de conquistar. Essa conquista é nossa.

Agradecimentos

Primeiramente a Deus, pela saúde e força concedida para enfrentar uma graduação longe de casa.

Aos meus pais, Orlei de Paula e Abadia Evangelista, as pessoas mais importantes da minha vida. Obrigado por sempre me apoiarem todas minhas decisões e me encorajar a buscar meus sonhos.

Aos meus amigos de infância Hiago Henrique e Guilherme Colmanetti, que mesmo de longe nunca deixaram de mandar mensagens positivas e de apoio. Obrigado pelas conversas que me mantiveram firme.

Aos amigos que se tornaram uma segunda família pelo laço comum com a Universidade. Emmanuel Reis, Gabriel Lisboa, Guilherme Tanaka, Guilherme Suzuki obrigado pelos anos compartilhados nas salas de aula, na república, nos estudos e nas festas.

A minha namorada Maria Gabriela, a pessoa que mais me incentivou a finalizar a graduação. Obrigado por sempre estar comigo em todos os momentos e por dividir sua vida comigo.

A todos os professores que tive a honra de conhecer e ser seu aluno. Obrigado por todo o conhecimento ao longo desse tempo. Em especial ao orientador Murillo Guimarães Carneiro, por acreditar no meu potencial.

Por fim, agradeço a mim, por todo o trabalho duro, por sempre buscar o melhor e por nunca desistir de mim mesmo.

“Pra quem tem pensamento forte, o impossível é só questão de opinião.”
Alexandre Magno Abrão (Chorão)

Resumo

O pôquer é um jogo de cartas com o objetivo bem simples, basicamente os jogadores apostam que sua combinação de cartas seja melhor que a do seus oponentes. Dado suas características, um ambiente multi-agente competitivo, parcialmente observável e estocástico, o jogo é utilizado como objeto de pesquisa nas áreas de Inteligência Artificial e Aprendizado de Máquina. O problema de aprender as estratégias de um jogador através de seus registros anteriores foi recentemente abordado na literatura através de um conjunto de dados modelado para representar quatro aspectos do jogo: posição, agressividade, qualidade da mão e situação atual. Várias técnicas de classificação tiveram resultado satisfatório, porém a importância dos atributos não foi analisada. Uma vez que atributos redundantes ou irrelevantes podem prejudicar o desempenho preditivo dos classificadores, este trabalho visa analisar como o processo de seleção de atributos se comporta para o problema. A abordagem ideal para selecionar o melhor subconjuntos de atributos seria experimentar todos os possíveis subconjunto, porém é um método inviável pela grande quantidade de subconjuntos possíveis. Dessa forma, foi investigado na literatura métodos de seleção de atributos como filtros e *wrappers* para serem utilizados em um ambiente computacional reproduzido da literatura e adaptado para a seleção de atributos. Foi identificado que certos atributos são importantes em determinadas etapas, porém podem se tornar irrelevantes no decorrer do jogo. Houve filtros que se destacaram tanto no desempenho preditivo quanto na redução de atributos, porém nem sempre a quantidade de atributos removidos está relacionada com melhoria de desempenho. A abordagem *wrapper* se mostrou mais efetiva, em ambas questões (redução de atributos e desempenho preditivo), porém com um custo computacional maior. De forma geral, a seleção de atributos se comprovou como uma importante etapa no pré-processamento de dados, uma vez que houve melhoria em vários classificadores em todas as etapas do jogo.

Palavras-chave: Seleção de atributos, Inteligência Artificial, Mineração de dados, Pôquer, Aprendizado de Máquina.

Abstract

Poker is a card game with a very simple objective: players bet that their card combination is better than the ones from their opponents. Given its characteristics, like competitive multi-agent environment, partially observable and stochastic, the game is used as a research problem in the areas of Artificial Intelligence and Machine Learning. The problem of learning a player's strategies through of his previous gamelogs was recently addressed in the literature. In that work, a dataset was modeled to represent four aspects of the game: position, aggression, hand quality and current situation. Some classification techniques had a satisfactory result, but the importance of the attributes was not analyzed. Since redundant or irrelevant attributes may affect the predictive performance of the classifiers, this work aims to analyze how does the attribute selection process may help to deal with the problem. The ideal approach to select the best subset of attributes would be to try out all possible subsets, but it is an unfeasible method due to the large amount of possible subsets. Thus, we investigated in the literature methods for selecting attributes such as filters and wrappers to be used in a computational environment reproduced from the literature and adapted to perform attribute selection. It was identified that certain attributes are important in certain stages, but can become irrelevant during the game. There were filters that stood out both predictive performance and attribute reduction, but the amount of attributes removed is not always related to the improvement of performance. The wrapper approach proved to be more effective, in both points (attribute reduction and predictive performance), although requiring a higher computational cost. In general, the selection of attributes demonstrated to be an important step to data pre-processing, as there was improvement in several classifier in all stages of the game.

Keywords: Attribute Selection, Artificial Intelligence, Data mining, Poker, Machine Learning.

Lista de ilustrações

Figura 1 – Etapas de um jogo de pôquer.	12
Figura 2 – Fluxograma do processo de descoberta de conhecimento em dados (KDD).	16
Figura 3 – Estados possíveis de um conjunto de quatro atributos.	17
Figura 4 – Tipos de abordagens para seleção e avaliação de subconjunto de atributos.	19
Figura 5 – Fluxograma do funcionamento de um algoritmo genéticos.	21
Figura 6 – Funcionamento de um algoritmo de aprendizado com o objetivo de classificar dados.	22
Figura 7 – Ranking das mãos de pôquer.	24

Lista de siglas

IA Inteligência Artificial

ANOVA Análise de Variância

AGs algoritmos genéticos

CBF *Consistency-Based Filter*

CSE *Classifier Subset Evaluator*

DT *Decision Tree*

EHS *Effective Hand Strength*

KNN *K-Nearest Neighbors*

LR *Logistic Regression*

NB *Naive Bayes*

PO *Pot Odds*

RF *Random Florest*

SA seleção de atributos

SelectFDR *Select False Positive Rate*

SelectFDR *Select False Discovery Rate*

SelectFWE *Select Family Wise Error*

Sumário

1	INTRODUÇÃO	11
1.1	Motivação	11
1.2	Objetivos e Desafios da Pesquisa	13
1.3	Hipótese	13
1.4	Organização da Monografia	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Seleção de Atributos	16
2.1.1	Filtros	18
2.1.2	Wrappers	19
2.2	Classificação de dados	21
2.3	Pôquer - Texas Hold'em No Limit	22
2.4	Trabalhos relacionados	24
3	MATERIAIS E MÉTODOS	27
3.1	Formulação do problema	27
3.2	Base de dados	27
3.3	Atributos no pré-flop	28
3.4	Atributos no pós-flop	30
3.5	Algoritmos de Seleção de Atributos	32
3.6	Análise de Variância (ANOVA)	33
3.7	Algoritmos de Classificação de Dados	34
3.8	Método para a Avaliação	35
4	EXPERIMENTOS E ANÁLISE DOS RESULTADOS	37
4.1	Resultados originais (sem seleção de atributos)	37
4.2	Resultados usando filtros	37
4.2.1	Pré-Flop	38

4.2.2	Flop	39
4.2.3	Turn	40
4.2.4	River	41
4.2.5	Discussão dos resultados	42
4.3	Resultados usando <i>wrapper</i>	43
5	CONCLUSÃO	46
	REFERÊNCIAS	49

Introdução

1.1 Motivação

O pôquer é um jogo de cartas no qual os jogadores apostam (ou blefam) que sua combinação de cartas é melhor do que a de seus oponentes. Dentre todas as vertentes do pôquer, a Texas Hold'em é a mais jogada e mais estudada na literatura.

O pôquer já foi enquadrado como um jogo de azar, quando acreditava-se que o fator de sorte era decisivo entre ganhar ou perder. Porém, avaliando um breve histórico do pôquer é possível identificar jogadores que ficaram milionários ganhando vários campeonatos contra diversas pessoas. Seria injusto dizer que eles só ganharam pois tem mais sorte que outros jogadores (PRATICA DO PÔQUER, 2018).

Seguindo essa afirmação a Associação Internacional dos Esportes da Mente, reconheceu o pôquer como um esporte da mente colocando-o ao mesmo nível de xadrez, dama e gamão (PRATICA DO PÔQUER, 2018). No Brasil, obteve destaque quando o Ministério do Esporte reconheceu o pôquer como um esporte da mente e registrou a Confederação Brasileira de Texas Hold'em (CBTH) em seus quadros, onde a mesma descreve o pôquer como: “uma prática de competição em que exige-se do praticante inteligência, capacidade, habilidades intelectuais e comportamentais para se obter sucesso” (MINISTERIO DO ESPORTE, 2012).

O Texas Hold'em é uma vertente muito complexa do pôquer, composta por quatro etapas com características diferentes: Pré-Flop, Flop, Turn, River, conforme apresentado na Figura 1. Em cada uma das etapas é possível executar várias ações diferentes (Fold, Bet, Call, Raise ou Check), sendo que a configuração do jogo possibilita uma variedade enorme de estratégias, onde não existe uma correta, mas sobre certas condições de jogo há algumas mais lucrativas (SKLANSKY; MALMUTH, 1999).

Desse modo, por se tratar de um jogo de habilidades humanas onde o jogador deve observar os comportamentos dos adversários (e do jogo) e escolher sua estratégia, os dados podem ser abstraídos e técnicas de mineração de dados e Inteligência Artificial (IA) podem ser aplicadas para reproduzir ações de um jogador, como pode ser observado na literatura.



Figura 1 – Etapas de um jogo de pôquer. Disponível em (CARNEIRO; LISBOA, 2018).

A dificuldade do pôquer se torna um desafio interessante aos olhos das pesquisas na área de IA devido a suas características como ambiente multi-agente competitivo, pois o jogador disputa com diversos adversários, com intenção de maximizar o seu lucro e simultaneamente maximizar o prejuízo dos outros adversários (RUSSEL; NORVIG, 2013).

Por existirem informações que não são possíveis de adquirir, como por exemplo as cartas dos oponentes, o pôquer também tem a característica de um ambiente parcialmente observável. Também é estocástico, pois apesar da próxima ação de um jogador ser determinada por estados anteriores, como ação de um adversário por exemplo, sempre há ao menos uma carta aleatória em qualquer etapa do jogo (RUSSEL; NORVIG, 2013).

Na literatura existem trabalhos que mostram como essas dificuldades foram superadas. Em um deles, um agente jogador de pôquer chamado Poki foi desenvolvido onde ele joga em um ambiente multi-agente contra jogadores humanos. O Poki guarda informações como cartas da mesa, informações disponíveis de outros jogadores e suas informações privadas, como suas cartas e modelos dos oponentes, que são as probabilidades de todas as cartas possíveis. Com essas informações, Poki consegue decidir que ação tomar (DAVIDSON et al., 2002). Outro trabalho apresentou um agente que joga a versão de dois jogadores contra outros agentes (BROWN; GANZFRIED; T., 2015) o qual foi capaz de ganhar a Competição Anual de Poker de Computador em 2014.

Além de trabalhos relacionados com IA, a literatura contém trabalhos que lidam com a modelagem de jogadores. Por exemplo, o trabalho de (CARNEIRO; LISBOA, 2018) abordou o problema de modelar as ações de um jogador humano a partir de seus registros anteriores. O trabalho utilizou um modo de jogo chamado Zoom Texas Hold'em. Nesse modo, os conjuntos de jogadores da mesa mudam a cada mão jogada. Isso traz benefícios como mais tempo de jogo, porém as informações sobre oponentes se tornam escassas uma

vez que não são sempre os mesmos (CARNEIRO; LISBOA, 2018).

Para recolher os dados necessários, os autores utilizaram logs de um cassino online de um jogador humano e formularam o problema de aprendizado sob duas formas distintas: 4 fases do jogo, uma forma comum utilizada na literatura, e em 2 fases (Pré-Flop e Post-Flop), uma nova proposta dado que as diferenças entre pré-flop e pós-flop são grandes, porém bem menores entre as etapas do pós-flop (flop, turn e river). O conjunto de atributos foi projetado em quatro categorias visando atingir as propriedades das estratégias de um jogador porém de forma genérica e simplificada. As categorias foram a qualidade da mão, posição, agressividade e situação atual (CARNEIRO; LISBOA, 2018).

Os conjuntos de dados foram avaliados por técnicas de aprendizado de máquina que mostraram bons desempenhos preditivos, evidenciando que as técnicas conseguiram de fato aprender boa parte das estratégias do jogador. Porém, um dos tópicos que os autores deste trabalho não chegaram a avaliar foi a relevância dos atributos criados e a possibilidade de incluir outros relevantes. Nessa direção, este trabalho visa investigar este primeiro ponto.

1.2 Objetivos e Desafios da Pesquisa

A abordagem ideal para selecionar o melhor subconjunto de atributos seria experimentar todos os possíveis subconjuntos com o algoritmo de mineração de dados desejado e escolher o que reproduziu o melhor resultado. Porém, esse método não é prático pois o número de subconjuntos de n atributos é 2^n , logo outros métodos devem ser investigados.

Para conseguir um subconjunto de atributos mais apropriado, este trabalho tem o objetivo de investigar a literatura na busca de métodos de seleção de atributos relevantes, como técnicas de filtros e wrappers. Dessa forma, pretende-se reproduzir o ambiente computacional com vários dos algoritmos de classificação considerados em (CARNEIRO; LISBOA, 2018). Especificamente, deve-se projetar um ambiente computacional que visa a seleção dos atributos, levando em consideração tanto os parâmetros das técnicas de filtro e wrappers quanto dos classificadores escolhidos, e analisar os subconjuntos de atributos encontrados em termos de simplicidade e desempenho preditivo.

1.3 Hipótese

Quando se utiliza técnicas de mineração de dados em um conjunto de dados com muitos atributos, pode ocorrer um fenômeno chamado maldição da dimensionalidade, onde os objetos ficam dispersos no espaço em que estão. Isso significa que quanto mais atributos que descrevem um conjunto de dados (alta dimensionalidade) os algoritmos de classificação, por exemplo, não terão dados suficientes para criação de um modelo confiável, e

assim o desempenho preditivo tende a ser menos preciso (PANG-NING; MICHEL; VIPIN, 2009).

Desse modo, a hipótese investigada aqui é que a identificação de um subconjunto de atributos através de técnicas de seleção de atributos pode melhorar o desempenho preditivo dos algoritmos no problema do pôquer em questão.

Pode parecer que ao selecionar um subconjunto de atributos acontecerá uma perda de informações, entretanto tais abordagens visam remover atributos menos relevantes de acordo com algum critério, de modo a simplificar o problema. Assim espera-se também identificar quais os atributos que mais contribuem e quais não estão contribuindo tanto, e dessa forma, novos atributos podem ser modelados visando cobrir novas informações relevantes.

1.4 Organização da Monografia

No Capítulo 2 está a fundamentação teórica deste trabalho. Apresentando conceitos sobre a seleção de atributos, uma parte importante do pré-processamento de dados que serão utilizados na mineração de dados. A seleção dos atributos visa buscar o melhor subconjunto possível com a premissa de que o desempenho da técnica de mineração de dados será melhorada. Dessa forma, toda a teoria de como as abordagens de seleção de atributos (filtros e *wrappers*) se comportam. Após a seleção de atributos, a tarefa de mineração de dados estudada nesse trabalho é a classificação, logo, temos uma sessão descrevendo como ocorre a classificação dos dados em determinadas classes. Por fim, neste capítulo temos conceitos básicos sobre o jogo de pôquer, onde é descrito de forma simples, como acontece uma rodada do jogo de pôquer passando por todas as etapas. Também quais e o que é cada ação do jogador e conceitos presentes na literatura do pôquer, como se dá força da mão do jogador e o *Pot Odds* (valor esperado da mão em relação a aposta).

No Capítulo 3 são apresentados os materiais e métodos deste trabalho. São descritos a formulação do problema, mais comum na literatura, o problema é dividido nas etapas do jogo. Também são apresentados a base de dados e atributos modelados, ambas disponíveis em (CARNEIRO; LISBOA, 2018); por fim, os algoritmos de seleção de atributos utilizados, tanto os filtros como os *wrappers* e como foi criado o método para a seleção de atributos utilizando ambas abordagens.

Os resultados são apresentados no Capítulo 4. Os resultados originais, sem utilizar a seleção de atributos, são recriando o ambiente disponível em (CARNEIRO; LISBOA, 2018). Os resultados dos filtros são divididos em sub capítulos para cada etapa do jogo, onde foram obtidos através de um ambiente criado para todos os filtros serem utilizados com todos os classificadores. Por fim, temos os resultados dos *wrappers*, para utilizar essa abordagem foram utilizados os melhores classificadores após passarem pelos filtros afim de comparar a melhor abordagem de filtro com wrapper.

Por fim, no Capítulo 5 é apresentado a conclusão deste trabalho após a análise dos resultados.

Fundamentação Teórica

Nesse capítulo serão apresentados os conceitos utilizados no trabalho de seleção de atributos, algoritmos genéticos, classificação de dados e conceitos necessários para entender o problema de aprendizado das estratégias de jogadores de pôquer abordado neste trabalho.

2.1 Seleção de Atributos

A seleção de atributos (SA) faz parte do pré-processamento de dados, que por sua vez é uma parte importante do processo de descoberta de conhecimento em dados (KDD - Knowledge Discovery in Data Bases) um processo que converte dados brutos em informações úteis (PANG-NING; MICHEL; VIPIN, 2009).



Figura 2 – Fluxograma do processo de descoberta de conhecimento em dados (KDD). Adaptado de (PANG-NING; MICHEL; VIPIN, 2009).

Os dados podem ser armazenados de diversas formas e formatos, assim o pré processamento costuma ser a etapa mais trabalhosa e demorada no processo do KDD. Essa etapa abrange diversos processos como agregação de dados, remoção de ruídos, discretização ou binarização, redução de dimensionalidade com seleção de atributos, etc (PANG-NING; MICHEL; VIPIN, 2009).

A SA tem o objetivo de selecionar um subconjunto de dados utilizando critérios ou medidas que atribuem importância aos atributos. Embora pareça que informações serão perdidas nesse processo, a intenção dessas técnicas é remover os recursos que são irrelevantes, informações que não são úteis, e redundantes.

Logo, é possível observar a importância do processo de selecionar os atributos visando o subconjunto ótimo. Entretanto, os números possíveis de subconjuntos para n atributos são de 2^n , logo se o n for grande o método de buscar todas as possíveis soluções é inviável. Para escapar dessa busca exaustiva, um algoritmo de SA é caracterizado em quatro tópicos que tornam a busca do subconjunto ótimo possível, os tópicos são: (1) o(s) ponto(s) de partida, (2) estratégia de pesquisa, (3) critério de parada e a (4) avaliação dos subconjuntos gerados (LEE, 2005).

A questão (1) influencia a direção em que a busca será realizada e os operadores que serão utilizados na geração dos próximos subconjuntos. Na Figura 3 é possível ver todos os estados possíveis de um conjunto de 4 atributos, então deve-se escolher um ponto de partida a partir desses estados para buscar o melhor subconjunto de atributos.

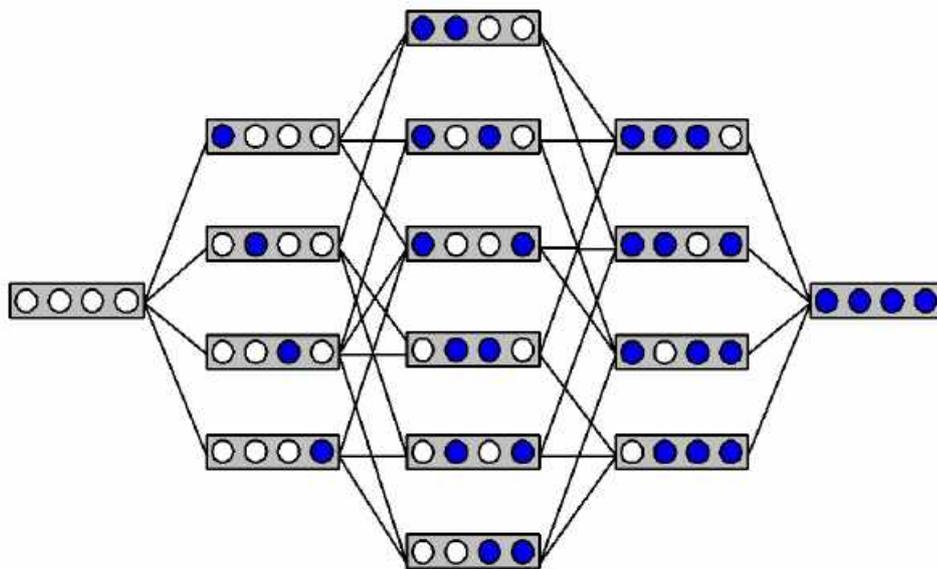


Figura 3 – Estados possíveis de um conjunto de quatro atributos. Disponível em (LEE, 2005).

Forward selection é uma abordagem onde o estado inicial é um subconjunto vazio, o mais à esquerda da figura, no próximo estado um atributo é adicionado no subconjunto e assim sucessivamente até algum critério de parada ser alcançado.

A abordagem de eliminação *backward*, funciona de forma análoga ao *forward selection*, sendo a diferença o ponto de partida, que é o subconjunto com todos os atributos, o mais à direita da figura, e a cada passo um atributo é retirado do subconjunto (LEE, 2005).

Existem outros tipos de abordagem, como a busca bidirecional, onde duas buscas começam simultaneamente, uma partindo do subconjunto inicial e outra partindo do

subconjunto final. O ponto de encontro dessa abordagem é o centro ou até uma das buscas encontrar o melhor subconjunto. Outro exemplo de abordagem escolhe o ponto de partida e direção aleatoriamente (PARMEZAN et al., 2012).

A questão (2) se trata da estratégia de busca onde a cada ponto desse processo as modificações nos subconjuntos são consideradas, uma dessas é selecionada e outra iteração acontece (LEE, 2005). Devido a uma busca exaustiva em todos os subconjuntos ser inviável (PANG-NING; MICHEL; VIPIN, 2009), existem três estratégias para lidar com o problema: busca completa, heurística e não-determinística.

Em alguns casos, uma busca ser completa, não significa que a busca seja exaustiva. Na busca completa nenhum subconjunto de atributos ótimo é perdido, mesmo sem percorrer todos os 2^n subconjuntos (LEE, 2005).

A segunda opção utiliza de algum tipo de heurística, ou seja, algum conhecimento específico para alcançar a solução de forma mais eficiente. Dessa forma, evita-se a busca exaustiva e a complexidade da busca pode ser reduzida. Em contra partida essa abordagem corre o risco de não encontrar o subconjunto ótimo. A busca não-determinística busca pelos subconjuntos de atributos de maneira estocástica.

Devido a possibilidade de haver um número enorme de subconjuntos e ser normalmente impossível passar por todos, a questão (3) é importante. O critério de parada é baseado em uma ou mais condições seguintes:

O número de iterações, se o valor da medida da avaliação de subconjunto é ótima ou excede um determinado limite, se um subconjunto de um determinado tamanho foi obtido, se critérios simultâneos de avaliação e de tamanho foram obtidos e se alguma melhoria pode ser alcançada pelas opções disponíveis a estratégia de pesquisa (PANG-NING; MICHEL; VIPIN, 2009).

Com um subconjunto selecionado, o mesmo deve seguir algum critério de avaliação para compreender o quão bom ele é, ou seja, como ele interage com os algoritmos de aprendizagem (LEE, 2005). Esses critérios podem ser divididos em duas abordagens maiores: os filtros e os wrappers.

- **a) Filtro:** O processo de SA acontece antes da aprendizagem, ou seja, é independente do algoritmo de aprendizagem alvo. Como o nome sugere, essa abordagem filtra atributos seguindo algum critério.
- **b) Wrapper:** Nessa abordagem os algoritmos de aprendizagem são utilizados como uma “caixa preta”. O SA busca um subconjunto de atributos e de acordo com o melhor desempenho do algoritmo.

2.1.1 Filtros

O processo de filtragem dos atributos acontece como uma etapa de pré-processamento, antes da aplicação do algoritmo de mineração desejado. O objetivo da filtragem é retirar

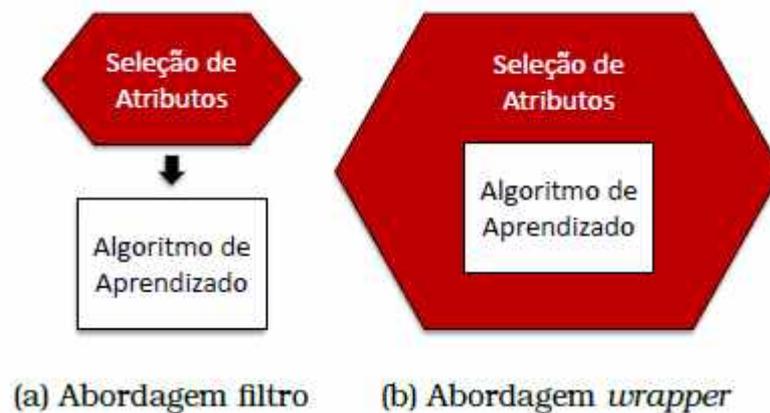


Figura 4 – Tipos de abordagens para seleção e avaliação de subconjunto de atributos. Adaptado de (PARMEZAN et al., 2012).

os atributos irrelevantes e/ou redundantes antes do processo de aprendizagem acontecer (LEE, 2005).

Essa abordagem leva em consideração as características dos subconjuntos independente do algoritmo que será utilizado, normalmente utilizando medidas estatísticas, como por exemplo, o algoritmo SelectKBest. Esse algoritmo tem como parâmetro uma função de teste estatístico e K atributos que deseja selecionar. Desse modo, esse algoritmo selecionará os K atributos que tiverem a pontuação mais alta de acordo com o teste escolhido (PEDREGOSA et al., 2011).

Um algoritmo que também segue essa abordagem é o *Consistency-Based Filter* (CBF). Ele utiliza a medida de consistência para determinar um subconjunto mínimo que satisfaz uma proporção de inconsistência definida pelo usuário. A inconsistência corresponde a ocorrência de dados que possuem valores de atributos iguais, porém com classes diferentes (PARMEZAN et al., 2012).

O CBF tem como parâmetros o conjunto original de dados, o número de atributos do conjunto original, um limiar aceitável de inconsistência e o número máximo de tentativas de gerar um subconjunto de atributo ótimo. A ideia do algoritmo é gerar o número máximo de subconjuntos avaliando de acordo com o tamanho do subconjunto e a medida de inconsistência de acordo com a classe. O subconjunto selecionado será aquele de menor tamanho e menor inconsistência dentre todos os subconjuntos gerados (PARMEZAN et al., 2012).

2.1.2 Wrappers

O processo de SA na abordagem wrapper também é separada do algoritmo de aprendizagem; porém, para validar o subconjunto selecionado, o próprio algoritmo de aprendizagem é utilizado. Ou seja, quando um subconjunto de atributos é gerado, essa abordagem

executa o algoritmo de classificação, o qual irá funcionar como uma caixa preta. O subconjunto que conseguir melhor desempenho preditivo será o subconjunto selecionado (PANG-NING; MICHEL; VIPIN, 2009).

Normalmente os subconjuntos selecionados por essa abordagem têm melhor desempenho em comparação ao filtro, devido ao fato que o algoritmo alvo é o mesmo utilizado no processo de selecionar os atributos. Entretanto, o custo computacional é maior pois é necessário criar um novo classificador a cada subconjunto gerado (PARMEZAN et al., 2012).

Um exemplo de algoritmo utilizando esta abordagem é o *Classifier Subset Evaluator* (CSE), que avalia os subconjuntos a partir do original. Ele tem como parâmetro um classificador para identificar a importância de determinado subconjunto, o conjunto original de dados e o número máximo de iterações (PARMEZAN et al., 2012).

Ele funciona da seguinte forma, para cada iteração o algoritmo seleciona um subconjunto de atributos e estima o desempenho preditivo desse subconjunto. Desse modo, se o desempenho for o melhor que o último escolhido então o subconjunto é o novo escolhido.

Outro exemplo de algoritmo que utiliza esse tipo de abordagem são os algoritmos genéticos os quais constituem de uma técnica de otimização e busca, inspirada no princípio Darwiniano de seleção natural e produção genética onde os indivíduos mais aptos ao ambiente sobrevivem (KALFELS, 2017). Segundo a teoria de Charles Darwin, os indivíduos mais aptos tem uma maior longevidade e, dessa forma, maior probabilidade de reprodução. Indivíduos com mais descendentes têm maior probabilidade de repassarem seus códigos genéticos para próximas gerações, tornando a população cada vez mais apta a sobrevivência (KALFELS, 2017).

Utilizando esses princípios os algoritmos genéticos buscam a melhor solução para um determinado problema, por meio da evolução de populações que representam o problema em questão. Uma população consiste em um conjunto de indivíduos que por sua vez, consiste em um conjunto de cromossomos. Os indivíduos são submetidos a avaliações de aptidão, seleção e operadores genéticos, ou seja, são mutados e combinados. O processo é executado diversas vezes (várias gerações) tornando a população cada vez melhor, ou seja, até alcançar a melhor solução possível para o problema (RODRIGUES et al., 2018). Na Figura 5 é apresentando um fluxograma do funcionamento básico de um algoritmo genético.



Figura 5 – Fluxograma do funcionamento de um algoritmo genéticos. Disponível em (RODRIGUES et al., 2018).

2.2 Classificação de dados

A classificação de dados é uma das tarefas mais conhecidas da mineração de dados. Essas tarefas de mineração de dados geralmente são divididas em dois grandes grupos, as tarefas descritivas e preditivas. Tarefas descritivas tem como objetivo as relações dos dados entre si como correlações, tendências, anomalias, etc. Por outro lado, as tarefas preditivas têm o objetivo de pré-dizer o valor de um determinado parâmetro baseado nos valores de outros (PANG-NING; MICHEL; VIPIN, 2009).

Desse modo, a classificação de dados é uma tarefa preditiva, na qual utilizando algoritmos busca-se prever o valor de um objeto (classe) dadas suas características (atributos). Abrange diversas aplicações, como detecção de spam em e-mails, categorizar diagnóstico de câncer de mama (RODRIGUES et al., 2018), entre tantas outras.

Para tarefas de classificação utiliza-se um conjunto de dados. Cada registro é caracterizado por uma tupla descrita como (X, y) , onde X é o conjunto de atributos que descrevem as características do objeto e y um valor que define o objeto, a classe do objeto. Com isso, a classificação tem o objetivo de aprender como classificar cada objeto x e uma determinada classe y , ou seja, criar um modelo de classificação utilizando uma base de treino. Uma parte do conjunto de dados é utilizada para o algoritmo aprender a como classificar tais objetos.

Cada técnica utiliza um algoritmo de aprendizado diferente para identificar o melhor modelo que se encaixa na relação entre os atributos e classe (PANG-NING; MICHEL; VIPIN, 2009). O modelo gerado deve ser capaz de receber o objeto x sem uma classe

definida e prever corretamente a classe y do mesmo. Para isso é utilizada outra parte do conjunto, a base de testes. Na Figura 6 indica o fluxograma do funcionamento da classificação de dados.

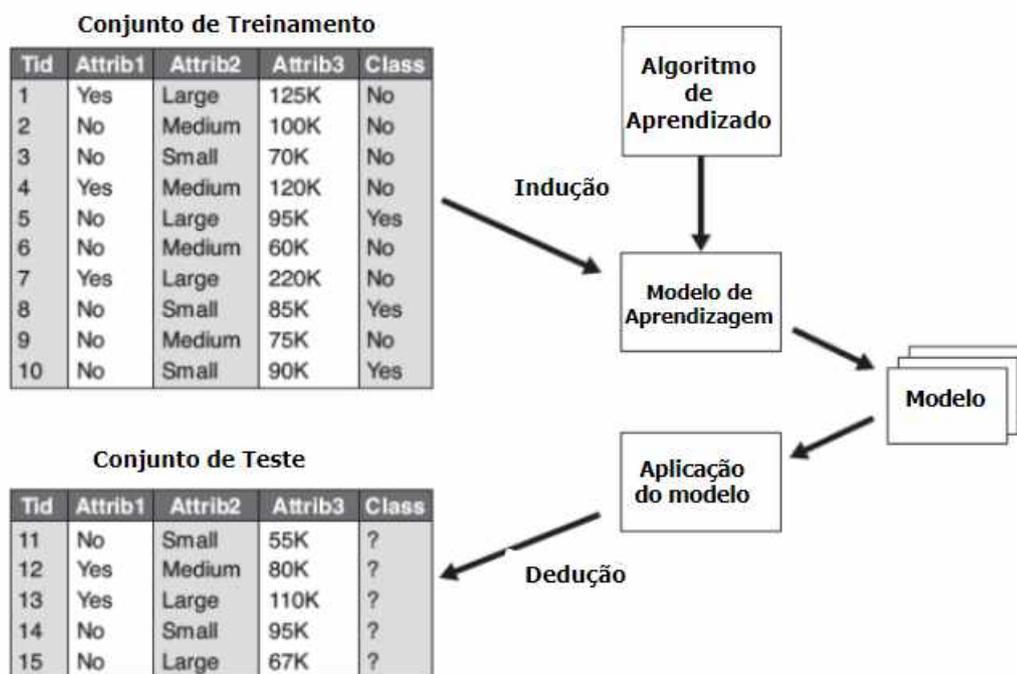


Figura 6 – Funcionamento de um algoritmo de aprendizado com o objetivo de classificar dados. Adaptado de (PANG-NING; MICHEL; VIPIN, 2009).

2.3 Pôquer - Texas Hold'em No Limit

O Texas Hold'em No Limit é uma modalidade do Pôquer dividido em quatro etapas que são: Pré-Flop, Flop, Turn e River. As apostas não têm quantidade máxima, assim um jogador pode apostar tudo que portar. As posições dos jogadores na mesa são importantes, pois terá vantagem aquele que for o último a fazer sua ação. O último jogador na mesa é marcado com o botão de dealer, ao terminar todas as etapas do jogo, esse botão é passado para o próximo jogador a esquerda.

O jogo se inicia no Pré-flop, onde os jogadores recebem duas cartas, denominadas de cartas privadas. Existem duas apostas obrigatórias que se inicia à esquerda do dealer, onde uma é a metade da aposta mínima e a outra a aposta mínima, o small blind e big blind respectivamente, desse modo haverá um pote (dinheiro em disputa) mínimo.

Os jogadores seguintes decidem se continuam no jogo (apostando) ou se desistem. No Pré-flop, todos os jogadores estão na disputa e as informações disponíveis para o jogador decidir que ação deve tomar são bem escassas pelo fato do jogo estar iniciando.

Na próxima etapa, no Flop, um volume maior de informações que contribuem para a decisão do jogador começam aparecer. Nessa etapa três cartas são colocadas na mesa para todos os jogadores utilizarem, cartas comunitárias. Além das cartas comunitárias, o jogador começa analisar seus adversários através de suas ações, contribuindo para sua decisão de continuar ou não no jogo.

As etapas de Turn e River são parecidas, somente uma carta comunitária é revelada. O River é a última etapa e se após as apostas finais houver em mais de dois jogadores disputando o pote, o ganhador é definido pela mão mais forte entre eles. O ganhador também pode aparecer em qualquer outra etapa, sendo aquele que fizer todos os outros adversários desistirem do pote.

Uma mão de pôquer é uma combinação de cinco cartas, sendo que elas podem ser as privadas e as comunitárias. O ranking de combinação das cartas é apresentado na Figura 7. A cada rodada de apostas, os jogadores têm cinco opções de ação para escolher:

1. **Check:** O jogador não aposta, continua com suas cartas privadas e passa a decisão para o próximo, entretanto não deve existir nenhuma aposta anterior.
2. **Bet:** O jogador aposta a quantia deseja, embora tivesse a opção de Check.
3. **Call:** O jogador aposta a mesma quantia do adversário.
4. **Raise:** O jogador aumenta a aposta do jogador adversário.
5. **Fold:** O jogador desiste da disputa, suas cartas não têm mais valor.

Na literatura do pôquer existem diversas ferramentas usadas frequentemente para avaliar situações de jogo e as mãos dos jogadores. Nesse trabalho, são empregados algumas dessas ferramentas que são apresentadas a seguir. O *Effective Hand Strength* (EHS) quantifica a força de uma mão comparando com todas as outras possíveis mãos, levando em consideração o potencial dela melhorar ou piorar. Ela utiliza da seguinte equação:

$$EHS = HS^n(1 - NPot) + (1 - HS^n)PPot, \quad (1)$$

onde HS é força da mão, sendo obtida por métodos que indicam a probabilidade da mão ser a melhor a qualquer momento do jogo; PPot é o potencial da mão melhorar de acordo com as próximas cartas comunitárias; NPot é o potencial negativo, ou seja, a probabilidade da mão piorar no futuro e n é o número de oponentes.

O *Pot Odds* (PO) determina o valor esperado da mão de um jogador em relação a aposta, onde c é a quantidade necessária para pagar a aposta e p o montante do pote na seguinte equação:

$$PO = \frac{c}{c + p}. \quad (2)$$

<p>Royal Flush Uma sequência do 10 ao Ás com todas as cartas do mesmo naipe.</p>	
<p>Straight Flush Cinco cartas consecutivas do mesmo naipe.</p>	
<p>Quadra Consiste em quatro cartas do mesmo valor e uma carta diferente.</p>	
<p>Full House Consiste em uma trinca e um par</p>	
<p>Flush Consiste em 5 cartas não consecutivas, mas do mesmo naipe.</p>	
<p>Sequência Consiste em 5 cartas consecutivas de naipes diferentes</p>	
<p>Trinca Três cartas do mesmo valor</p>	
<p>Dois Pares Quando você tem dois pares na sua mão</p>	
<p>Par Quando você consegue duas cartas de mesmo valor em sua mão.</p>	
<p>Carta Alta Quando você não tem nenhum par, sequência, etc.então é carta mais alta da sua mão.</p>	

Figura 7 – Ranking das mãos de pôquer. Adaptado de (CARNEIRO; LISBOA, 2018).

2.4 Trabalhos relacionados

Existem diferentes trabalhos que são relacionados com o tema do pôquer utilizando técnicas de aprendizagem de máquina, como por exemplo, a criação de modelos que visam explicar o que faz um jogador de pôquer vitorioso ou não (JOHANSSON; SONSTROD; NIKLASSON, 2006). Para isso, alguns atributos foram levantados e duas técnicas de árvore de decisão foram aplicadas para extração de regras. Ambas as técnicas tiveram boa acurácia e conseguiram identificar quais estratégias que fazem um jogador ganhar ou perder mais (JOHANSSON; SONSTROD; NIKLASSON, 2006).

Outros trabalhos visam criar modelos de jogadores profissionais de pôquer Texas Hol'dem, a partir do seu conjunto de dados aplicando diversas técnicas de classificação, dado que o objetivo é prever a ação efetuada pelo jogador. Técnicas com Naives Bayes,

tabelas de decisão, árvores de decisão, entre outras foram aplicadas (SILVA; REIS, 2016).

Em relação a técnicas de seleção de atributos, um trabalho relacionado buscou avaliar algumas técnicas de SA importantes para o aprendizado de máquina. Os autores escolheram sete conjuntos de dados e neles aplicaram cinco técnicas de seleção de atributos, entre filtros e *wrappers* (PARMEZAN et al., 2012). Após selecionarem os subconjuntos, foram criados modelos com o subconjunto selecionado e outros modelos com o conjunto de dados originais utilizando duas técnicas de classificação. Desse modo, foram comparados os atributos selecionados por cada técnica de SA e o desempenho preditivo do conjunto de dados original e o selecionado. Assim, foi possível observar que após reduzir o número de atributos, algumas técnicas tiveram o desempenho preditivo maior e os conjuntos de dados que tinham muitos atributos, foram bem reduzidos (PARMEZAN et al., 2012).

Os algoritmos genéticos (AGs) são utilizados como uma seleção de atributos baseados em *wrappers*, uma vez que podem ser utilizados algoritmos de classificação para serem a função de aptidão para a escolha dos indivíduos, sendo um indivíduo um subconjunto de atributo. Na literatura, temos trabalhos como o apresentado em (RODRIGUES et al., 2018) que utilizam seleção de atributos e algoritmo genético para o apoio ao diagnóstico do câncer de mama utilizando imagens. Neste trabalho utilizaram imagens termográficas com diversos diagnósticos (sem lesão, cistos, lesões benignas e lesões malignas). Após técnicas de pré-processamento de imagens e extração de atributos as imagens foram classificadas utilizando todos seus atributos, 169 no total. Após aplicado o algoritmo genético, os atributos foram reduzidos em 70%, porém com uma acurácia reduzida nos classificadores utilizados (RODRIGUES et al., 2018). Entretanto, o resultado obtido nesse trabalho condiz com a literatura uma vez que taxas de mutação muito alta e um valor de gerações muito baixa podem corroborar em uma queda no desempenho preditivo (RODRIGUES et al., 2018). De modo geral, o trabalho mostrou que a abordagem ao utilizar algoritmos genéticos para seleção de atributos foi positiva, a qual foi evidenciada por uma queda significativa de atributos sem uma grande perda de desempenho preditivo (RODRIGUES et al., 2018).

Por fim, o trabalho de (CARNEIRO; LISBOA, 2018), apresentou uma abordagem para modelar as ações de jogador humano de pôquer a partir dos seus registros anteriores. O trabalho utilizou um modo de jogo diferente, denominado Zoom. Nesse modo, o conjunto dos jogadores muda a cada mão jogada. A vantagem desse modo é o maior tempo de jogo, porém as informações obtidas dos oponentes são escassas uma vez que eles sempre estão mudando. Os autores recolheram os dados a partir de logs de um cassino online e formularam o problema de duas formas. Nas 4 fases do jogo e em 2 fases, somente pré-flop e pós-flop. Dessa forma, projetaram um conjunto de atributos genéricos e simples, retratando as propriedades de estratégias de um jogador de pôquer. Aplicando as técnicas de aprendizagem de máquina, o trabalho apresentou um modelo que consegue aprender boa parte das estratégias do jogador com bons desempenhos preditivos. Entretanto, o

trabalho deixou pontos que podem ser explorados, como por exemplo, analisar a relevância dos atributos propostos pelos autores. Este trabalho busca explorar esse ponto em aberto. Para isso, neste trabalho, serão utilizadas técnicas de seleção de atributos, como filtros e wrappers, para encontrar subconjunto de atributos relevantes para analisar quais atributos foram selecionados, comparar o desempenho preditivo desses subconjuntos e se possível criar novos atributos.

Materiais e Métodos

3.1 Formulação do problema

Neste trabalho estaremos lidando com um formato do Texas Hold'em em que os oponentes do herói (jogador a ser estudado) mudam ao fim de cada mão. Nesse formato chamado Zoom, o número de oponentes que o herói enfrenta é grande devido ao fato do conjunto da mesa sempre estar mudando, e isso se torna um desafio para a área de IA uma vez que as informações disponíveis de cada oponente são escassas (CARNEIRO; LISBOA, 2018).

Nesse trabalho será usada a formulação do problema mais comum na literatura, que problema maior ser dividido nas quatro fases do jogo. As estratégias de um jogador tendem a mudar ao decorrer do jogo, desse modo, dividir o problema em pedaços menores tende a diminuir a complexidade para lidar com o problema.

3.2 Base de dados

Serão utilizadas neste trabalho quatro bases de dados geradas em (CARNEIRO; LISBOA, 2018), são elas Pré-Flop, Flop, Turn, River. Essas quatro bases representam o problema, sendo a divisão em cada etapa do jogo (Pre-Flop, Flop, Turn e River). As bases foram obtidas através de um cassino online, Poker Stars, em jogos com dinheiro real em mesas de 2 a 5 dólares contra adversários humanos (CARNEIRO; LISBOA, 2018).

Inicialmente, foi realizado o pré-processamento dos dados, com o objetivo de limpar as bases retirando objetos ambíguos e duplicados. Objetos ambíguos podem ocorrer por diversas razões, como por exemplo, um movimento errado, uma mudança de estratégia do jogador, etc. Esses são os objetos que possuem os mesmos atributos porém com classes diferentes, e para retirar esses objetos seguem os critérios: manter o objeto mais frequente e, em caso de empate, aquele em que a classe tiver menos objetos (CARNEIRO; LISBOA, 2018).

Na Tabela 1 são apresentados a quantidade de dados e distribuição das classes após as etapas de pré-processamento.

Base de dado	Qtd.Objetos	Qtd.Attr	Classes [Distribuição das classes]
ZP-PreFlop	37685	11	4 [75.7%, 12.7%, 9.1%, 2.5%]
ZP-Flop	25983	14	5 [9.2%, 48.4%, 8.2%, 32.6%, 1.6%]
ZP-Turn	15649	14	5 [8.4%, 49.6%, 11.0%, 29.5%, 1.5%]
ZP-River	10342	14	5 [11.7%, 52.9%, 8.9%, 24.8%, 1.7%]

Tabela 1 – Distribuição da base de dados disponível em (CARNEIRO; LISBOA, 2018)

Os atributos que descrevem o problema foram divididos em quatro categorias pelos autores, conforme segue abaixo:

1. **Qualidade da mão:** Um conjunto de atributos que visam descrever a força da mão.
2. **Posição:** Atributos que apresentam uma vantagem ou desvantagem da posição do jogador na mesa.
3. **Agressividade:** Atributos que caracterizam jogadores passivos-agressivos e mostram cenários que possibilitam um blefe.
4. **Situação atual:** Atributos que descrevem eventos mais recentes do jogo, como ações anteriores, dinheiro em disputa, etc.

3.3 Atributos no pré-flop

A fase do pré-flop tem a característica de conter poucas informações, especialmente no formato Zoom onde as informações sobre os oponentes é quase nula devido ao fato dos oponentes sempre estarem mudando. Os atributos modelados em (CARNEIRO; LISBOA, 2018) nessa etapa são apresentados na Tabela 2.

ID-Nome atributo	Descrição	Valores possíveis
FC-FirstCard	Valor da primeira carta	{A,K,...,2}
SC-Sec.Card	Valor da segunda carta	{A,K,...,2}
SU-Suited	Se as duas cartas são do mesmo naipe	{1,0}
BR-Broadway	Se uma das cartas são de valor alto (A,K,Q,J ou 10)	{1,0}
CON-Connect	Indica a relação das cartas	
	Se as cartas estão em sequência	10000
	Se as cartas tem um pulo entre elas, por exemplo, 7 e 8	01000
	Se as cartas tem um pulo entre elas, por exemplo, 7 e 9	00100
	Se as cartas tem o mesmo valor, por exemplo A e A	00010
	Se nenhuma das anteriores	00001
PO-Position	Descreve a posição do jogador na mesa. Small- Blind(1), BigBlind(2), ..., Dealer(6)	{1, 2, ..., 6}
PAPF-Prev.Act.PF	India as acções dos oponentes antes da ação do herói	
	Se todos desistiram antes do herói	1
	Se um jogador pagou o valor do big-blind e os restantes desistiram	2
	Se dois ou mais jogadores pagaram o valor do big-blind e os restantes desistiram	3
	Se qualquer oponente aumentou a aposta antes do herói	4
	Se qualquer oponente aumentou a aposta e um ou mais jogadores pagaram a aposta antes do herói	5
	Se qualquer oponente aumentou a aposta depois do herói	6
	Se qualquer oponente aumentou a aposta depois do herói e um ou mais jogadores pagaram a aposta	7
	Se dois ou mais jogadores aumentaram a aposta e nenhum jogador pagou antes do herói	8
	Se dois ou mais jogadores aumentaram a aposta e pelo menos um jogador pagou	9
TP-TotalPot	Indica a quantidade total em disputa	$R > 0$
BV-BetVillain	O valor da última aposta do oponente	$R > 0$
PO-PotOdds	Determina o valor esperado da mão de um jogador em relação a aposta	$R > 0$
ACT-Act.Hero	A ação decidida pelo herói: Fold(1), Check(2), Call(3), Bet(4) or Raise(5)	{1, 2, ..., 5}

Tabela 2 – Descrição dos atributos do Pré-Flop

3.4 Atributos no pós-flop

Nas etapas de pós-flop é possível adquirir mais informações do que no pré-flop. Entretanto, as etapas de pós-flop têm características similares. Desse modo foi levantado um conjunto de atributos simples capaz de capturar as características de cada etapa e ao mesmo tempo diferenciar as estratégias que podem ser utilizadas nelas (CARNEIRO; LISBOA, 2018). Seguindo as categorias de atributos, qualidade da mão, posição, agressividade e situação atual são apresentadas na Tabela 3.

ID-Nome atributo	Descrição	Valores possíveis
EHS-Eff.Hand.Str.	Quantifica a força de uma mão comparando com todas as outras possíveis mãos, levando em consideração o potencial dela melhorar ou piorar, dado pela equação 1	[0,1]
PO-Position	Descreve a posição do jogador na mesa. Small- Blind(1), BigBlind(2), ..., Dealer(6)	{1, 2, ..., 6}
AGG-AgressorPos	Indica o jogador que fez o último aumento no pré flop Se o herói decide antes do agressor Se o herói é o agressor Se o herói decide após o agressor	1 2 3
IP-InPosition_VS	Números de jogadores antes do herói	{1, 2, ..., 5}
OP-OutPosition_VS	Números de jogadores após o herói	{1, 2, ..., 5}
PRA-Prev.RoundAct.	Descreve a ação do herói na rodada anterior: Check(1), Call(2), Bet(3) or Raise(4)	{1, 2, 3, 4}
BSU-BoardSuit	indica se há cartas comunitárias do mesmo naipe Se todas as cartas são de naipes diferentes Se há duas cartas do mesmo naipe Se há pelo menos três cartas do mesmo naipe	1 2 3
BCA-BoardCards	indica se há cartas comunitárias do mesmo valor Se não há cartas do mesmo valor Se há duas cartas do mesmo valor Se há três cartas do mesmo valor	1 2 3
BCON-BoardConnect	indica se há cartas comunitárias em sequência Se há três cartas em sequência, por exemplo 5,6,7 Se há três cartas em sequência com pelo menos dois "pulos", por exemplo, 753 Se não acontecer nenhum dos itens acima	3 2 3
PA-Prev.Act	Indica as ações anteriores ao herói Se o herói é o primeiro a jogar Se todos os jogadores anteriores ao herói escolheram Check Se o qualquer jogador fez uma aposta antes do herói Se o herói fez uma aposta e pelo menos um oponente pagou a aposta Se o qualquer jogador fez uma aposta e pelo menos um oponente aumentou a aposta	0 1 2 3 4
RO-Round	Indica a etapa do pós-flop: flop(1), turn(2) ou river(3)	{1, 2, 3}
TP-TotalPot	Indica a quantidade total em disputa	$R > 0$
BV-BetVillain	O valor da última aposta do oponente	$R > 0$
PO-PotOdds	Determina o valor esperado da mão de um jogador em relação a aposta	$R > 0$
ACT-Act.Hero	A ação decidida pelo herói: Fold(1), Check(2), Call(3), Bet(4) or Raise(5)	{1, 2, ..., 5}

Tabela 3 – Descrição dos atributos do Pós-Flop

3.5 Algoritmos de Seleção de Atributos

Os algoritmos de seleção baseados em filtros investigados neste estudo são divididos em dois grupos: algoritmos de remoção de atributo com baixa variação e algoritmos de seleção univariada, no qual a seleção se dá pelos melhores atributos pontuados com base em testes estatísticos. Para a pontuação dos atributos é utilizada a função de análise de variância (ANOVA) (PEDREGOSA et al., 2011). Também temos a abordagem *wrapper* que usa o próprio algoritmo de classificação para selecionar o melhor subconjunto de atributos, ou seja, a cada subconjunto gerado, o mesmo é testado no modelo classificador, sendo o subconjunto com melhor desempenho selecionado. Dessa forma, espera-se que no mínimo o desempenho do subconjunto escolhido seja igual do filtro. Para o algoritmo baseado nessa abordagem foi utilizado um algoritmo genético para seleção de atributos.

As escolhas dos parâmetros para os algoritmos foram feitas de forma empírica.

- **Variance Threshold:** Uma abordagem mais simples de seleção de atributo. Basicamente, o *variance threshold* remove os atributos cujo a variação do seus valores não atinge um determinado limite. A ideia por trás dessa abordagem nos diz que um atributo que não varia muito, tem um baixo poder de predição, logo ele pode ser descartado. O algoritmo recebe somente um parâmetro, o limiar (*threshold*) máximo permitido de variância de um atributo (PEDREGOSA et al., 2011) $\in \{0.0, 0.01, 0.02, 0.03\}$.
- **Select K-Best:** Seleciona os K atributos com a maior pontuação. Para isso, recebe dois parâmetros a função de pontuação (Análise de Variância (ANOVA)) e um número $K \in \{3, 4, 5, 6, 7, 8, 9\}$ de atributos que se deseja selecionar (PEDREGOSA et al., 2011).
- **Select Percentile:** Recebe dois parâmetros, a função de pontuação (ANOVA) e uma porcentagem desejada de atributos para serem selecionados. Dessa forma, a porcentagem dos atributos melhores classificados são os atributos selecionados por esse algoritmo (PEDREGOSA et al., 2011). Sendo *porcentagem* $\in \{30, 40, 50, 60, 70\}$.
- **Select False Discovery Rate (SelectFDR):** Considera um valor *alpha* como a taxa de falsos positivos para cada atributo calculado a partir do valor da função de pontuação de ANOVA e selecionará os atributos com a taxa *alpha* abaixo do limite estabelecido. Recebe dois parâmetros, a função de pontuação (ANOVA) e *alpha* $\in \{0.02, 0.03, 0.04, 0.05, 0.06, 0.07\}$, que o limite máximo da taxa de falsos positivos (PEDREGOSA et al., 2011).
- **SelectFDR:** Esse algoritmo utiliza a taxa de descoberta falsa (FDR) para selecionar os atributos. Recebe dois parâmetros, a função de pontuação (ANOVA) e com os valores-p da função é calculado o FDR de cada atributo. O segundo parâmetro

é um $alpha \in \{0.02, 0.03, 0.04, 0.05, 0.06, 0.07\}$, como um limite máximo permitido, onde os valores abaixo desse são os atributos selecionados (PEDREGOSA et al., 2011).

- **Select Family Wise Error (SelectFWE):** Semelhantes ao anteriores esse algoritmo também recebe dois parâmetros, a função de pontuação (ANOVA) e um $alpha$ como limite máximo. O selectFwe calcula a taxa de erro familiar, que é a probabilidade de conter ao menos um falso positivo, e se essa taxa for abaixo do limite permitido o atributo é selecionado, sendo $alpha \in \{0.02, 0.03, 0.04, 0.05, 0.06, 0.07\}$ (PEDREGOSA et al., 2011).
- **Algoritmo Genético:** Algoritmo de busca de conjunto de atributos baseado na evolução genética. Recebe como parâmetro o tamanho da população (100), o número de gerações (20), a probabilidade de mutação dos indivíduos (20%), probabilidade de combinações de indivíduos (50%), o modelo de classificação utilizado e a *fitness function*, a função que verificará o quão apta está a população, sendo nesse caso, a função de treinamento do modelo de classificação. Com a população inicial são selecionados os melhores indivíduos (pais) para criar novos indivíduos com a mistura dos cromossomos (atributos) dos pais ou por mutação de cromossomos do indivíduo (alteração de atributos), e esse processo é repetido até uma nova população ser criada, e assim temos uma geração (RODRIGUES et al., 2018).

3.6 Análise de Variância (ANOVA)

A análise de variância (ANOVA) é uma função estatística que faz uma análise de uma variância total de um grupo. Para isso, a variação é analisada de duas formas: a validação dentro dos grupos, ou seja, avalia a diferença dos valores (atributos) em relação a média do grupo (classe) em que pertence; e a variação entre os grupos, que avalia a diferença de valores entre os diversas classes.

Para realizar os cálculos de tais variações, a ANOVA utiliza duas hipótese:

- Hipótese nula: Não existe diferenças estatisticamente significantes entre as médias dos grupos.
- Hipótese não nula: Há diferenças entre as médias dos grupos

Com essas hipóteses, é possível calcular uma estatística F, que se dá pela equação seguinte.

$$F = \frac{MQ_{entre}}{MQ_{dentro}}, \quad (3)$$

sendo o numerador a média quadrática da variação entre os grupos e o denominador a média quadrática da variação dentro do grupo. Assim teremos um f-valor que indica o tamanho da diferença entre os grupos com relação a variação dentro de cada grupo.

3.7 Algoritmos de Classificação de Dados

A tarefa de classificação de dados é a etapa em que associamos os objetos em classes, de acordo com o aprendizado de cada algoritmo. Para isso, utilizamos cinco dos classificadores adotados em (CARNEIRO; LISBOA, 2018) possíveis de reproduzir no ambiente disponível, uma máquina pessoal com poder de processamento limitado.

- **Decision Tree (DT):** Como o nome sugere, uma estrutura de árvore onde os nós decidem para que caminho a decisão irá seguir (PANG-NING; MICHEL; VIPIN, 2009). Espera-se dois parâmetros, o número mínimo de amostras para dividir um nó interno $min_samples_split \in \{0.10, 0.20, 0.30\}$ e o número mínimo de amostrar para ser um nó folha $min_samples_leaf \in \{0.10, 0.20, 0.30\}$
- **Random Florest (RF):** Cria diversas árvores de decisão aleatórias e as combinadas com a intenção de aumentar a acurácia do algoritmo (PEDREGOSA et al., 2011). Ajusta-se um parâmetro, o número de árvores na floresta $t \in \{2^1, 2^2, \dots, 2^{10}\}$.
- **K-Nearest Neighbors (KNN):** Classifica o objeto baseado na classe majoritária dentre os k vizinhos mais próximos do objeto a ser classificado (PANG-NING; MICHEL; VIPIN, 2009). Sendo k o único parâmetro do classificador, $k \in \{5, 10, 15, \dots, 30\}$.
- **Naive Bayes (NB):** É um classificador probabilístico que utiliza o Teorema de Bayes para classificar os objetos. Calcula a probabilidade de um evento A dado que ocorreu um evento B (PANG-NING; MICHEL; VIPIN, 2009), ou seja, calcula a probabilidade de um objeto ter a classe y (evento A) sendo que ele tem determinadas características (evento B). O teorema de Bayes se dá pela seguinte equação.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad (4)$$

Não há parâmetros esperados.

- **Logistic Regression (LR):** A predição é feita com base no efeito em que cada atributo exerce em relação à classe do objeto (PEDREGOSA et al., 2011). Tem dois parâmetros, a norma de penalidade $p = l1$ e a força de regularização $C \in \{2^2, 2^4, \dots, 2^{14}\}$

3.8 Método para a Avaliação

Cada simulação foi conduzida executando uma validação cruzada aninhada tal como em (CARNEIRO; LISBOA, 2018). Ela é basicamente uma repetição da validação cruzada k -fold em n vezes. Ou seja, o conjunto de dados é dividido em k partes e uma dessas partes é utilizada como um conjunto de teste. Para as $k-1$ partes é feita outra validação cruzada, onde é dividida em n partes e realizada a seleção do melhor modelo (hiper-parâmetros). Por fim, tal modelo é avaliado com o conjunto de teste separado anteriormente, assim temos k avaliações do classificador e obtemos uma média para o valor de acurácia final.

Nesse trabalho, utilizamos para a validação cruzada externa o valor de k como 10 e a validação interna, o valor de n igual a 3. Para cada base, foi combinado um classificador com todos os seletores utilizando um pipeline, responsável pela transformações nos dados, a seleção dos atributos, e em seguida o processo de classificação. Segue abaixo um pseudo-código do método de avaliação desenvolvido utilizando uma validação cruzada.

Algoritmo 1 Algoritmo de base de seleção de atributo e classificador utilizando filtros

```

0: for pipe of pipelines do
0:   filter = filtro do pipe
0:   classifier = classificador do pipe
0:   Divide a base de dados em k partes
0:   for  $i = 1, 2, \dots, k - 1$  do
0:     Divide a base i em n partes
0:     for  $j = 1, 2, \dots, n - 1$  do
0:       Aplica filtro em base j
0:       Treina modelo classificador com filtro aplicado em base j
0:       Testa modelo do classificador com filtro aplicado em base j
0:       if acurácia do modelo atual > acurácia modelo anterior then
0:         Seleciona modelo atual como melhor modelo
0:       end if
0:     end for
0:     Testa o melhor modelo de classificador com filtro aplicado em base k - 1
0:     Salva acurácia e atributos selecionados
0:   end for
0: end for=0

```

Por fim, temos k resultados de valor da acurácia bem como atributos selecionados, dessa forma é calculada a média da acurácia, uma média de quantidade de atributos selecionados e os atributos mais selecionados de acordo com a média.

Após a análise dos resultados dos seletores baseados em filtros, o próximo passo é utilizar a abordagem *wrapper* utilizando o algoritmo genético. Para cada etapa do jogo, utilizaremos o algoritmo de classificação com melhor desempenho com o filtro aplicado.

Para testarmos a aptidão do subconjunto gerado (indivíduo) utilizamos o parâmetro *fitness_function*, uma função de treinamento de dados utilizando um modelo de classificação. Nessa função, utilizamos uma validação cruzada de 5 partes para calcular a média

de acurácia do classificador em questão, e internamente ocorre outra validação cruzada de 5 partes para a escolha dos melhores hiper-parâmetros do classificador.

No próximo capítulo são apresentados os resultados das combinações dos filtros e com todos os classificadores em todas as bases de dados, e os resultados do algoritmo genético aplicado nos melhores classificadores em cada etapa anterior.

Experimentos e Análise dos Resultados

4.1 Resultados originais (sem seleção de atributos)

Os resultados sem a seleção de atributos apresentado na Tabela 4, mostra que os classificadores têm dificuldades nas etapas de pós flop (flop, turn e river), sendo etapas que o desempenho fica bem abaixo ao comparar com a etapa de pré-flop.

Dessa forma, ao aplicarmos os algoritmos de seleção espera-se que na etapa de pré-flop não ocorra uma queda de performance, uma vez que o desempenho dessa etapa é alta, e por outro lado que ocorra um aumento de acurácia no pós-flop.

Algoritmos	PréFlop	Flop	Turn	River
DT	80.2±0.3	73.94±0.4	74.0±0.6	77.40±0.8
RF	92.9±0.3	76.5±0.5	74.0±1.0	81.4±0.9
kNN	90.6±0.4	77.5±0.8	72.0±1.3	75.4±1.1
NB	80.6±0.6	70.8±1.2	62.4±1.2	72.1±1.3
LR	86.6±0.5	73.4±0.7	71.1±1.0	76.8±0.7

Tabela 4 – Resultados dos classificadores.

4.2 Resultados usando filtros

Os resultados usando filtros são apresentados na ordem das etapas do jogo (pré-flop, flop, turn e river). Para cada etapa há uma discussão sobre os resultados baseado em duas tabelas: em uma contém os valores originais de acurácia dos classificadores e total de atributos. Contém também, a acurácia que provém da combinação dos filtros e classificadores, a quantidade de atributos selecionados e o número de classificadores melhorados por determinado filtro; na outra contém todos os atributos indicando quais foram selecionados e quais não foram, para todos os filtros.

4.2.1 Pré-Flop

A Tabela 5 apresenta a acurácia média obtida por diferentes estratégias de filtros e algoritmos de classificação na etapa do pré-flop. O valor em negrito indica o melhor resultado obtido, e os valores sublinhados representam um aumento de acurácia em relação aos resultados originais.

Algh	Original	Variance T.	K-best	Percentil	FPR	FDR	FWE
DT	80.25±0.30	<u>80.26±0.4</u>	<u>80.27±0.3</u>	<u>80.29±0.2</u>	<u>80.27±0.4</u>	<u>80.27±0.4</u>	80.24±0.55
RF	92.97±0.32	92.98±0.3	84.65±0.5	86.71±1.3	92.95±0.3	92.87±0.2	92.97±0.4
kNN	90.65±0.4	<u>90.70±0.4</u>	85.67±0.3	85.94±0.3	90.60±0.4	<u>90.72±0.4</u>	90.57±0.4
NB	80.65±0.6	<u>81.18±0.4</u>	<u>80.88±0.6</u>	<u>80.84±0.3</u>	<u>80.69±0.5</u>	<u>80.67±0.6</u>	<u>80.68±0.5</u>
LR	86.65±0.5	83.65±0.6	83.19±0.4	83.87±0.4	83.73±0.3	83.56±0.4	83.71±0.6
#Atr	14	13	7	8	14	14	13
#Melh	-	4/5	2/5	2/5	0/5	0/5	1/5

Tabela 5 – Resultados da seleção de atributos por Filtros na base Pré Flop.

No Pré-Flop a maioria dos algoritmos seletores conseguiu melhorar a acurácia em pelo menos um classificador, sendo o destaque o seletor Variance Threshold que melhorou em 4 dos 5 classificadores com a redução de somente um atributo. O algoritmo Select K-best foi o destaque no requisito de redução de dimensionalidade, diminuindo 50% dos atributos, com 7 atributos retirados, porém em questão de desempenho, em apenas 2 classificadores (Arvore de Decisão e Naive Bayes) houve uma melhora, com uma perda de até 8% na acurácia nos 3 classificadores restantes.

A combinação de classificador e seletor que obteve o melhor resultado foi o Random Forest com Variance Threshold, apresentando 92.98% de acurácia. Foi necessário somente um atributo a menos para o desempenho do classificador apresentar uma melhora. Isso demonstra que filtrar muitos atributos não é sinônimo de melhoria de desempenho, uma vez que o método de filtro leva em condição somente os atributos e não o classificador utilizado.

Os filtros FPR e FDR foram considerado sem melhora nos seus desempenhos pois não houve uma redução na quantidade de atributos. Alguns resultados desses filtros mostraram uma melhoria apesar de não reduzir seus atributos, porém pode ocorrer que no momento da validação cruzada alguns modelos foram filtrados alguns atributos e obtiveram um resultado melhor afetando a acurácia média final, porém na média de atributos selecionados se manteve igual ao original.

Por fim, na etapa de Pré-Flop, tivemos um total de 9 combinações que resultaram em um desempenho melhor em um total de 30, que representa uma melhoria de 30%.

O algoritmo Variance Treshold melhorou a acurácia em 4 dos 5 classificadores retirando o atributo TOTAL_POT, que representa a quantidade de dinheiro em disputa, como podemos observar na Tabela 6, a qual mostra os atributos escolhidos por cada algoritmo

de seleção. Na tabela, o nome em negrito indica o algoritmo que obteve o melhor resultado na melhoria de desempenho dos classificadores e o nome sublinhado indica o algoritmo que reduziu mais atributos. Nesse sentido, o Variance Treshold elimina atributos de variância baixa, com a hipótese de que atributos que variam pouco seu valor não tem peso grande no momento de decisão. No começo do jogo o dinheiro em disputa, geralmente, é um valor semelhante devido as valores de apostas iniciais obrigatórias e um padrão de aposta inicial que praticamente todo jogador de pôquer executa, logo o valor em disputa tem uma variação baixa no início do jogo.

Features/Filtros	Variance T.	<u>K-best</u>	Percentil	FPR	FDR	FWE
FIRST_CARD	X	X	X	X	X	X
SECOND_CARD	X	X	X	X	X	X
POSITION	X	-	-	X	X	X
SUITED	X	-	X	X	X	X
BROADWAY	X	X	X	X	X	X
CONNECTOR	X	-	-	X	X	X
ONEGAPPER	X	-	-	X	X	X
TWOGAPPER	X	-	-	X	X	-
PAIR	X	X	X	X	X	X
DISCONNECT	X	X	X	X	X	X
PREVIOUS_ACTION	X	X	X	X	X	X
TOTAL_POT	-	-	-	X	X	X
POT_ODDS	X	-	-	X	X	X
BET_VILLAIN	X	-	-	X	X	X

Tabela 6 – Atributos selecionados por Filtros na base Pré Flop.

4.2.2 Flop

A Tabela 7 é referente a segunda etapa do jogo, e com o total de 13 atributos, o desempenho dos classificadores se mantiveram acima de 70%. Ao aplicar os seletores de filtragem, a maioria dos classificadores se mantiveram com o mesmo limite de acurácia do original, entretanto foi a etapa na qual menos houve melhora, sendo em somente 7 combinações resultando em 26.6% de melhora. Os seletores que filtraram mais atributos foram o *Select K-best* e *Select Percentil*, com 9 atributos no total em ambos filtros, porém em ambos somente um classificador apresentou aumento no desempenho.

Na Tabela 8 vemos que novamente o seletor *Variance Treshold* obteve uma melhora em 4 classificadores e também, novamente, retirando somente um atributo, o TOTAL_POT, indicando que esse atributo é pouco variável nessa etapa do jogo no modo analisado. O melhor resultado nessa etapa, foi com o algoritmo de seleção univariada, o FWE, retirando somente um atributo que representa a ação do jogador na rodada anterior (PREV_ROUND_ACTION), indicando ser o atributo menos pontuado pelo algoritmo. Entretanto, somente dois classificadores mostraram melhora, logo o atributo pode ser importante nessa etapa para alguns classificadores uma vez que na teoria do pôquer o

jogador deve contar com sua ação anterior para decidir a atual, caso sua jogada anterior seja um blefe, é um atributo importante para a decisão de manter o blefe (ou não) na jogada atual, por exemplo.

Algh	Original	Variance T.	K-best	Percentil	FPR	FDR	FWE
DT	73.94±0.6	<u>74.10±0.7</u>	73.93±0.6	73.90±1.0	73.90±0.7	73.90±0.7	73.85±0.4
RF	76.58±0.5	<u>76.66±0.51</u>	<u>76.94±0.6</u>	74.39±0.7	76.49±0.8	76.54±0.6	76.42±0.4
kNN	77.58±0.8	<u>77.63±0.6</u>	<u>74.47±0.6</u>	76.96±0.9	77.53±0.5	77.55±0.7	77.78±0.6
NB	70.85±1.2	<u>71.54±0.7</u>	70.17±0.8	70.13±0.8	70.69±0.8	70.79±1.2	70.70±0.9
LR	73.49±0.7	<u>73.48±0.5</u>	72.61±1.1	<u>72.59±1.1</u>	<u>73.51±0.9</u>	<u>73.55±0.8</u>	<u>73.55±0.8</u>
#Atr	13	12	9	9	13	13	12
#Melh	-	4/5	1/5	1/5	0/5	0/5	2/5

Tabela 7 – Resultados da seleção de atributos por Filtros na base Flop.

Features/Filtros	Variance T.	K-best	Percentil	FPR	FDR	FWE
POSITION	X	X	X	X	X	X
EHS	X	X	X	X	X	X
TOTAL_POT	-	X	X	X	X	X
POT_ODDS	X	X	X	X	X	X
BOARD_SUIT	X	X	X	X	X	X
BOARD_CARDS	X	-	-	X	X	X
BOARD_CONNECT	X	-	-	X	X	X
PREV_ROUND_ACTION	X	-	-	X	X	-
PREVIOUS_ACTION	X	X	X	X	X	X
BET_VILLAN	X	X	X	X	X	X
AGG	X	X	X	X	X	X
IP_VS	X	X	X	X	X	X
OP_VS	X	-	-	X	X	X

Tabela 8 – Atributos selecionados por Filtros na base Flop

4.2.3 Turn

A etapa do Turn foi a mais equilibrada entre as demais, todos os filtros melhoraram em pelo menos 1 classificador, com um total de 16 melhorias sendo equivalente a uma melhoria de 56,6% e a etapa com mais melhorias nos classificadores até o momento. Na Tabela 9 vemos que o classificador Naive Bayes destacou-se com todas as combinações de filtros mostrando uma melhora significativa, indicando que a seleção de atributos é importante em determinados classificadores, influenciando para o alcance de um desempenho maior que o esperado.

Em todos os algoritmos de seleção univariada o atributo PREV_ROUND_ACTION foi eliminado, vide Tabela 10, o que indica que é um atributo com a pontuação abaixo da esperada por seus respectivos filtros. Entretanto, houve uma melhora em diversos classificadores onde esse atributo foi retirado, isso pode indicar que se encaminhar para

as etapas finais o jogador pode ter a tendência de não utilizar as informações de suas jogadas anteriores para decidir sua atual, ou seja, uma mudança de estratégia que não depende da sua ação anterior.

Alg	Original	Variance T.	K-best	Percentil	FPR	FDR	FWE
DT	74.09±0.6	73.99±0.6	70.43±1.0	70.28±0.8	73.74±0.8	74.01±1.0	74.67±1.0
RF	74.04±1.0	<u>74.12±0.9</u>	69.23±0.8	69.16±0.8	74.03±1.4	73.89±0.9	73.37±0.8
kNN	72.00±1.0	<u>72.4±1.2</u>	<u>72.10±1.1</u>	71.96±0.9	<u>73.75±1.1</u>	<u>73.55±1.1</u>	74.24±1.2
NB	62.47±1.2	<u>69.48±0.8</u>	<u>69.19±1.3</u>	<u>69.34±0.8</u>	<u>67.28±0.6</u>	<u>67.30±0.9</u>	<u>67.39±1.4</u>
LR	71.15±1.0	<u>70.93±1.1</u>	<u>71.35±1.2</u>	<u>71.29±0.7</u>	<u>71.16±0.7</u>	<u>71.17±0.7</u>	<u>71.26±0.7</u>
Atr	13	11	8	8	12	12	11
#Melh	-	3/5	3/5	2/5	3/5	3/5	3/5

Tabela 9 – Resultados da seleção de atributos por Filtros na base Turn.

Features/Filtros	Variance T.	K-best	Percentil	FPR	FDR	FWE
POSITION	X	X	X	X	X	X
EHS	X	X	X	X	X	X
TOTAL_POT	-	X	X	X	X	X
POT_ODDS	X	X	X	X	X	X
BOARD_SUIT	-	X	X	X	X	X
BOARD_CARDS	X	-	-	X	X	-
BOARD_CONNECT	X	-	-	X	X	X
PREV_ROUND_ACTION	X	-	-	-	-	-
PREVIOUS_ACTION	X	-	-	X	X	X
BET_VILLAN	X	X	X	X	X	X
AGG	X	X	X	X	X	X
IP_VS	X	X	X	X	X	X
OP_VS	X	-	-	X	X	X

Tabela 10 – Atributos selecionados por Filtros na base Turn

4.2.4 River

A última etapa do jogo, o River, apresentou a melhor performance de melhoria de acurácia dos classificadores com 66,67% de melhoria, com 20 das combinações melhoradas. Na Tabela 11 vemos como destaque três filtros que igualaram a marca de 4 classificadores melhorados. Nessa etapa, vemos um comportamento não comum ao restante das etapas. Enquanto, nas etapas anteriores, principalmente nas primeiras etapas (Pre-Flop e Flop) filtros que retiram muitos atributos não apresentavam uma melhoria de desempenho no classificador, na maioria da vezes, nas etapas finais, grande parte dos classificadores apresentaram um desempenho maior mesmo com poucos atributos. Isso pode indicar que a medida que o jogo se encaminha pro final, menos atributos são necessários para o jogador decidir sua ação.

Ao chegar na etapa final, é possível identificar atributos não selecionados em comum em todo o pós flop como vemos na Tabela 12, por exemplo atributos que caracterizam as cartas comunitárias, ou seja, cartas que todos os jogadores compartilham para realizar a combinação. São esses atributos o BOARD_CARDS e BOARD_CONNECT, representando se há cartas comunitárias do mesmo valor e sequência, respectivamente. No pós flop, pelo menos 2 filtros retiraram ambos atributos, e no river 5 dos 6 filtros retiraram esses atributos. Indicando que ambos atributos são pouco pontuados pelas funções estatísticas, dessa forma, podem ser pouco relevantes no momento de decisão. Em cada etapa do flop, o jogador estudado aparentemente não se utiliza das informações provenientes desses atributos, uma vez que houve melhora significativa sem esses atributos. Isso pode indicar que o jogador analisado poucas vezes decide sua ação utilizando informações provenientes de cartas comunitárias.

Algoritmos	Original	Variance T.	K-best	Percentil	FPR	FDR	FWE
DT	77.40±0.8	77.26±0.7	77.15±0.7	<u>77.42±1.1</u>	77.37±1.2	<u>77.46±1.2</u>	<u>76.99±1.1</u>
RF	81.49±0.9	81.53±0.8	78.44±1.1	<u>78.43±1.0</u>	80.69±1.2	81.1±1.2	80.6±0.7
kNN	75.43±1.1	<u>75.69±0.6</u>	<u>79.70±0.5</u>	<u>79.84±1.0</u>	<u>79.68±1.0</u>	<u>79.81±0.7</u>	<u>80.15±1.2</u>
NB	72.10±1.3	<u>74.60±1.3</u>	<u>76.52±1.2</u>	<u>76.69±0.9</u>	72.00±1.2	<u>72.21±1.3</u>	<u>72.21±1.0</u>
LR	76.81±1.0	<u>77.20±0.9</u>	<u>77.86±1.0</u>	<u>78.06±1.1</u>	<u>77.27±1.2</u>	<u>77.38±1.2</u>	<u>77.27±0.8</u>
Atr	13	12	7	7	10	10	10
#Melh	-	4/5	3/5	4/5	2/5	4/5	3/5

Tabela 11 – Resultados da seleção de atributos por Filtros na base River.

Features/Filtros	Variance T.	K-best	Percentil	FPR	FDR	FWE
POSITION	X	X	X	X	X	X
EHS	X	X	X	X	X	X
TOTAL_POT	-	-	-	X	X	X
POT_ODDS	X	X	X	X	X	X
BOARD_SUIT	X	X	X	X	X	-
BOARD_CARDS	X	-	-	-	-	-
BOARD_CONNECT	X	-	-	-	-	-
PREV_ROUND_ACTION	X	-	-	-	-	-
PREVIOUS_ACTION	X	-	-	X	X	X
BET_VILLAN	X	X	X	X	X	X
AGG	X	X	X	X	X	X
IP_VS	X	X	X	X	X	X
OP_VS	X	-	-	X	X	X

Tabela 12 – Atributos selecionados por Filtros na base River.

4.2.5 Discussão dos resultados

Na Tabela 13, vemos o quanto cada algoritmo de seleção foi eficiente com a quantidade total de classificadores melhorados e quantos atributos foram retirados somado em todas

as etapas. Vemos que o algoritmo Variance Threshold se destaca dos demais, com poucos atributos retirados durante as etapas houve melhora na grande parte dos classificadores. Logo, o Variance Threshold é um filtro que se comporta de forma positiva com diversos algoritmos de classificação, sendo a melhor opção dentre os filtros estudados. Esse filtro eliminou o atributo TOTAL_POT em todas as etapas do jogo o que indica que é um atributo pouco variável durante a etapa desse modo de jogo. Isso pode ocorrer pelo padrão de apostas de jogadores regulares, que entendem o mínimo da teoria do jogo e assim a forma de aposta quase sempre semelhante juntamente com o fato de que as apostas são sempre um múltiplo do valor do big blind, logo em vários dados da mesma etapa podem coincidir em valores semelhantes, tornando essa variância baixa.

Filtros	Variance T.	K-best	Percentil	FPR	FDR	FWE
#classificadores melhorados	15/20	9/20	8/20	5/20	7/20	9/20
#atributos retirado	5/53	21/53	21/53	4/53	4/53	9/53

Tabela 13 – Quantidade de melhorias das combinações sobre total

Por outro lado, os seletores K-best e Percentil foram os que mais retiraram atributos porém com pouca efetividade no desempenho dos algoritmos. Pela abordagem do filtro ocorrer antes da classificação de dados pode ocorrer a remoção de atributos importantes para uma classificação efetiva e desse modo é necessário uma remodelação dos hiper-parâmetros dos filtros para uma possível melhora de desempenho.

Por fim, a abordagem filtro demonstra que a mesma pode ser efetiva porém para isso deve-se escolher o conjunto de hiper-parâmetros com cautela sendo um fator decisivo para a filtragem de atributos irrelevantes a melhoria de desempenho da mineração de dados. Por outro lado, com os filtros é possível identificar atributos que funcionam ou não com determinados algoritmos de classificação. Porém, se torna difícil afirmar que o atributo realmente é desnecessário para o classificador, uma vez que o filtro não utiliza do mesmo para validar sua seleção.

4.3 Resultados usando *wrapper*

Para o algoritmo genético, que utiliza a abordagem wrapper, é apresentado o resultados do mesmo aplicado nas etapas do pós flop (flop, turn e river) com o classificador que obteve maior desempenho na abordagem de filtro. Na Tabela 14 vemos as informação de valor da acurácia original, do filtro, wrapper e as quantidades de atributos selecionados pelos respectivos algoritmos de seleção. Na Tabela 15 vemos quais os atributos selecionados e não selecionados pelo algoritmo genético em cada etapa.

Já a abordagem wrapper se utiliza do próprio classificador para testar o subconjunto selecionado para verificar se o mesmo será o melhor conjunto. Dessa forma, espera-se

que seu desempenho seja no mínimo aproximadamente da acurácia utilizando filtro e o desempenho original. Dessa forma, nas etapas do pós flop, utilizamos o classificador que teve o melhor desempenho respectivamente de cada etapa. Dessa forma, o KNN para etapas de flop e turn e o random forest na etapa de turn. Aplicamos um algoritmo genético, com uma população de 100 indivíduos com 20 gerações. Para a função de aptidão, a fitness function (função de treinamento), utilizando uma validação cruzada 5-folds para o treinamento do classificador com outra validação cruzada 5-folds para escolha dos melhores hiper parâmetros para o mesmo.

Etapa	Classificador	Acc. Original	Filtro	atrib. filtros	Algh Genético	atrib wrappers
Flop	KNN	77.58±0.8	77.78±0.6	12	77.42±0.7	9
Turn	KNN	72.00±1.0	74.24±1.2	11	75.33±0.9	7
River	RF	81.49±0.9	81.53±0.8	12	81.52±0.2	10

Tabela 14 – Resultados da seleção de atributos na abordagem wrapper utilizando algoritmo genético.

A abordagem wrapper demonstrou resultados satisfatórios em relação a diminuição de dimensionalidade, o algoritmo genético reduziu a quantidade de atributos em todas as etapas do pós flop. Na etapa do turn, a redução dimensionalidade apresentou uma melhora de acurácia tanto em comparação da acurácia original e do melhor filtro aplicado. O algoritmo genético selecionou menos atributos que a etapa de filtro, o que pode indicar que na etapa do turn o jogador estudado precisa de menos atributos para escolher sua ação. Ou seja, ao chegar no fim do ciclo do jogo o jogador utiliza menos informações para decidir sua jogada.

No flop, houve redução de dimensionalidade em comparação ao filtro, mas não melhora na acurácia, entretanto a redução foi baixa. No river houve uma melhora em relação ao original e em relação ao filtro foi uma acurácia muito próxima. Isso pode indicar que menos atributos nessas etapas não afeta drasticamente a acurácia do classificador, mostrando que algumas características levantadas nessas etapas podem não ser decisivas para o jogador.

Vemos que a abordagem wrapper precisou de menos atributos para alcançar (ou melhorar) a acurácia em relação aos filtros. Na etapa do flop, por exemplo, houve filtros que também selecionaram 9 atributos, porém não obtiveram a mesma acurácia dos 9 atributos do algoritmo genético. Dessa forma, a abordagem wrapper se mostra mais eficiente devido ao fato de que os atributos mantidos pelo mesmo são realmente os mais relevantes, sem prejudicar o desempenho do classificador. Em contrapartida, a abordagem pode não selecionar atributos importantes e relevantes, afetando negativamente o desempenho do algoritmo.

Na Tabela 15 são apresentados os atributos selecionados pelo algoritmo genético nas etapas do pós flop. Nessa tabela podemos ver que atributos são realmente relevantes em todas as etapas e outros que não são utilizados em todas as etapas. podendo indicar que

o jogador utiliza de características diferentes em cada etapa para decidir a estratégia que irá seguir.

Features	Flop	Turn	River
POSITION	X	-	-
EHS	X	X	X
TOTAL_POT	-	-	X
POT_ODDS	X	X	X
BOARD_SUIT	-	X	-
BOARD_CARDS	-	-	X
BOARD_CONNECT	X	-	X
PREV_ROUND_ACTION	X	-	X
PREVIOUS_ACTION	X	X	X
BET_VILLAN	X	X	-
AGG	-	-	X
IP_VS	X	X	X
OP_VS	X	X	X

Tabela 15 – Atributos selecionados pelo algoritmo genético em cada etapa.

Entretanto temos atributos que foram mantidos em todas as etapas, como o EHS e POT_ODDS, sendo respectivamente a força da combinação de cartas do jogador e o valor esperado da mão em relação a aposta. São atributos importantes, pois é necessário ter uma combinação forte o suficiente para ganhar a aposta e deve valer a pena arriscar a disputar o pote (valor total em disputa) com tal combinação.

A teoria de pôquer diz que a posição em que se está na mesa é importante, ao fato de que o último a realizar uma ação tem vantagem aos demais. Somente no flop, o algoritmo genético selecionou o atributo POSITION, que indica a posição do jogador na mesma, entretanto em todas as etapas foram selecionados os atributos IP_VS, adversários após o jogador estudado, e OP_VS, adversários antes do jogador estudado respectivamente. É natural que ao passar as etapas menos jogadores se mantêm na disputa, dessa forma a posição real do jogador se torna menos relevante mas a informação de jogadores antes e após pode suprir o atributo não selecionado.

De forma geral, a abordagem wrapper se mostrou efetiva, selecionando atributos que aumentaram ou mantiveram o desempenho semelhante a acurácia do filtro aplicado. Vamos que os atributos realmente são relevantes ao comparar com a abordagem de filtros, uma vez que em algumas etapas a quantidade de atributos selecionados foram iguais (mas não os mesmos atributos) porém a acurácia da abordagem wrapper foi maior. Em contrapartida, o tempo de processo computacional da abordagem wrapper foi muito maior em comparação aos filtros, uma vez que o processo do algoritmo de classificação é utilizado para escolher o subconjunto.

Conclusão

O pôquer é um esporte da mente, como o xadrez, dama e gamão, exigindo do praticamente uma habilidade intelectual para obter um sucesso, onde a sorte não é o fator decisivo de sucesso. Existem diversos fatores que tornam o jogo difícil e interessante para a área de inteligência artificial, como por exemplo, o seu ambiente multi-agente competitivo, informações ocultas para o jogador e situações aleatórias durante o jogo, como as cartas comunitárias e as cartas de cada jogador.

Com um trabalho que estudou as ações do jogador de pôquer online, modelou um conjunto de atributos e aplicou algoritmos de classificação em todas as etapas do jogo (CARNEIRO; LISBOA, 2018), iniciamos o estudo com um tópico não abordado, a relevância dos atributos modelados e as possibilidades de novos atributos.

Desse modo quase todos os objetivos traçados inicialmente foram alcançados, como a investigação da literatura com a escolha dos algoritmos de seleção atributos para serem utilizados no trabalho. Reproduzimos o ambiente proposto, também criamos um ambiente visando a seleção de atributos e analisamos os resultados dados o desempenho preditivo e quais atributos foram selecionados ou não.

Uma abordagem ideal seria testar todos os subconjuntos possíveis e escolher o que reproduziu o melhor desempenho, porém é uma abordagem inatingível dado que o número de subconjuntos de n atributos é 2^n , logo buscamos métodos de seleção de atributos que tornam essa busca viável, utilizando métodos de seleção de atributos baseados em filtros e wrappers, como Select K-Best, Select Percentil, Variance Threshold e algoritmo genético.

Reproduzimos ambiente computacional apresentado em (CARNEIRO; LISBOA, 2018), utilizando as mesmas bases de dados e os classificadores possíveis de reprodução para que os resultados originais não sejam afetados por algum processamento divergente entre a máquina do trabalho citado e deste trabalho. Também foi projetado um ambiente utilizando a abordagem de filtro, o qual visa selecionar os atributos para que em seguida seja feita a classificação dos dados com os atributos selecionados. Em seguida, com os melhores resultados foi utilizado o algoritmo genético, para fins de comparação da base original em ambas abordagens.

Com a análise dos resultados obtidos com a seleção dos filtros, vemos que o desempenho dos classificadores se manteve o mesmo ou houve um aumento, mostrando que uma redução de atributo pode contribuir para um aumento de desempenho e consequentemente uma diminuição de processamento, uma vez que, com menos atributos para serem analisados o processo de mineração de dados demanda menos processamento. Houve casos em que uma diminuição muito grande dos atributos teve uma queda de desempenho em determinados classificadores, uma vez que os filtros não consideram o classificador que será utilizado e somente visam filtrar os atributos de acordo com seus próprios métodos. Em contrapartida, houve classificadores que aumentaram o desempenho preditivo somente ao retirar poucos atributos, ou seja, os atributos descartados não contribuiriam positivamente para determinados classificadores.

Com a seleção do wrapper, vemos que essa abordagem aumentou o desempenho do classificador com menos atributos necessários em comparação ao filtro demonstrando que essa abordagem seleciona melhor seus atributos, ou seja, os atributos selecionados realmente tem um valor preditivo alto uma vez que a acurácia é superior com menos atributos. Em contrapartida, essa abordagem tem um gasto computacional maior, uma vez que utiliza do próprio classificador para selecionar os atributos.

O conjunto de atributos estudado contém um número de 13 a 14 atributos, dependendo da etapa do jogo, sendo considerado de uma dimensionalidade pequena comparado a grandes bases de dados. Entretanto, a seleção de atributos se mostrou eficiente em todas as etapas com pelo menos um classificador em ambas as abordagens, mesmo em uma base de dados com uma dimensionalidade relativamente pequena.

Ao utilizar os filtros, foi identificado que a quantidade de atributos retirados não está diretamente ligado a melhora de desempenho, uma vez que os filtros que retiraram mais atributos não melhoraram muitos classificadores. Porém, de forma geral, vemos que a abordagem de filtro pode ser efetiva como foi visto ao utilizar o algoritmo Variance Threshold, que melhorou um total de 15 classificadores em todas as etapas.

Com a abordagem wrapper utilizando um algoritmo genético houve um grande gasto computacional, entretanto isso foi compensado pelos resultados obtidos. Essa abordagem mostrou-se mais eficiente ao reduzir mais atributos em comparação ao filtro e aumentando o desempenho. Com isso, vemos quais atributos realmente são mais relevantes para o jogador nas etapas de pós flop como os atributos que a força da mão e o quanto vale a pena pagar a aposta para ganhar o pote do momento, respectivamente EHS e POT_ODDS.

Houve etapas em que os algoritmos de seleção retiraram diversos atributos, porém em alguns casos o tivemos um baixo aumento na acurácia, isso pode indicar que nem toda a informação retirada é de fato irrelevante. Dessa forma, como trabalhos futuros, uma reformulação dessas informações retiradas afim de criar novos atributos pode contribuir para um aumento maior do desempenho. Um novo ou maior conjunto de hiperparâmetros para os seletores pode contribuir uma seleção de atributos mais refinada aumentando o

desempenho dos classificadores. Utilizar outros algoritmos de seleção de atributos e de classificação de dados, podendo surgir de novas opções de combinações com desempenho melhor.

Por fim, é possível afirmar a hipótese traçada no início do estudo, ao aplicar técnicas de seleção de atributos (filtros e wrappers) foi possível identificar o melhor subconjunto em cada etapa do jogo aumentando o desempenho preditivo, juntamente com quais atributos foram mais relevantes. A seleção de atributos se mostrou uma etapa importante para o processo de descoberta de conhecimento, pois ao reduzirmos a dimensionalidade dos dados, podemos obter melhores resultados escolhendo corretamente o tipo de abordagem para o tipo de dados sob análise.

Referências

- A prática do pôker no Brasil: questões legais e a visão do poder judiciário. [S.l.], 2018. Disponível em: <<http://zmb.adv.br/a-pratica-do-poker-no-brasil-questoes-legais-e-a-visao-do-poder-judiciario/>>. Citado na página 11.
- BROWN, N.; GANZFRIED, S.; T., S. Hierarchical abstraction, distributed equilibrium computation, and post-processing, with application to a champion no-limit texas hold'em agent. **International Conference on Autonomous Agents and Multiagent Systems**, p. 1–10, 2015. Citado na página 12.
- CARNEIRO, M. G.; LISBOA, G. A. What's the next move? learning player strategies in zoom poker games. **IEEE International Congress on Evolutionary Computation**, p. 1–8, 2018. Citado 11 vezes nas páginas 12, 13, 14, 24, 25, 27, 28, 30, 34, 35 e 46.
- DAVIDSON, A. et al. The challenge of poker. **Artificial Intelligence**, p. 1–7, 2002. Citado na página 12.
- JOHANSSON, U.; SONSTROD, C.; NIKLASSON, L. Explaining winning poker – a data mining approach. **International Conference on Machine Learning and Applications**, 2006. Citado na página 24.
- KALFELS, E. H. H. Algoritmos genéticos como uma abordagem para a alocação de grade de horários da ufsc. p. 24–50, 2017. Citado na página 20.
- LEE, H. D. Seleção de atributos para a extração de conhecimento de base de dados. **Tese Doutorado ICMC-USP**, 2005. Citado 3 vezes nas páginas 17, 18 e 19.
- MINISTERIO DO ESPORTE. **Confederação Brasileira de Texas Hold'em**: Informação de entidades. [S.l.], 2012. Disponível em: <<http://portal.esporte.gov.br/cen/detalhesEntidades.do?idEntidade=74>>. Citado na página 11.
- PANG-NING, T.; MICHEL, S.; VIPIN, K. **Introdução ao data mining: Mineração de dados**. [S.l.]: Ciência Moderna, 2009. Citado 7 vezes nas páginas 14, 16, 18, 20, 21, 22 e 34.
- PARMEZAN, A. . R. et al. Avaliação de métodos para seleção de atributos importantes para aprendizado de máquina supervisionado no processo de mineração de dados. p. 48–50, 2012. Citado 4 vezes nas páginas 18, 19, 20 e 25.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado 4 vezes nas páginas 19, 32, 33 e 34.

RODRIGUES, A. L. et al. Seleção de atributos para apoio ao diagnóstico do câncer de mama usando imagens termográficas, algoritmos genéticos e otimização por enxame de partículas. **II SIMPÓSIO DE INOVAÇÃO EM ENGENHARIA BIOMÉDICA ©2018 SABIO**, p. 1–8, 2018. Citado 4 vezes nas páginas 20, 21, 25 e 33.

RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. [S.l.]: Elsevier, 2013. Citado na página 12.

SILVA, N.; REIS, L. P. Modelação de jogadores profissionais de poker utilizando data-mining. **2016 11th Iberian Conference on Information Systems and Technologies (CISTI)**, 2016. Citado na página 25.

SKLANSKY, D.; MALMUTH, M. **Texas Hold'em for Advance Players LLC**. [S.l.]: Two Plus Two Publishing, 1999. Citado na página 11.