

# Predição de Desempenho Acadêmico no Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Bahia (ADS-IFBA)

Pedro M. Mendes<sup>1</sup>, Flávia M. S. Nascimento<sup>1</sup>

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA)

mmendes.pedro@outlook.com, flaviamsn@ifba.edu.br

**Abstract.** *This work proposes the use of Machine Learning algorithms to anticipate the performance of students in the Technology in Analysis and Development of Systems course at the Federal Institute of Education, Science and Technology of Bahia (Salvador Campus). More specifically, the proposal consists of adopting students' performance in the Introduction to Logic subject as a predictive attribute for inferring their approvals in the Object Oriented Programming subject, thus evaluating how strongly their performances in these subjects are associated. The results demonstrated some appropriateness of the employed techniques, achieving an average recall of up to 88%.*

**Resumo.** *Este trabalho propõe o uso de algoritmos de Aprendizado de Máquina a fim de antecipar o desempenho de alunos do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Bahia (Campus Salvador). Mais especificamente, a proposta consiste em adotar o desempenho dos alunos na disciplina de Introdução à Lógica como atributo preditor das inferências de suas aprovações em Programação Orientada a Objetos e, assim, avaliar o quão fortemente estão associados os seus desempenhos nessas disciplinas. Os resultados demonstraram alguma apropriabilidade do uso das técnicas empreendidas, alcançando uma revocação média de até 88%.*

## 1. Introdução

O baixo desempenho acadêmico de estudantes em cursos superiores é um assunto complexo que depende de várias condições internas e externas ao processo educacional [Czibula et al. 2019]. Ainda assim é nobre estabelecer esforços na busca de meios que representem uma melhora real no itinerário estudantil. Neste contexto, o uso de recursos de tecnologia da informação (TI) tem ganhado destaque não só como infraestrutura, mas também no processo de apoio à tomada de decisão [Garcia et al. 2023]. Tecnologias de análise de dados gerados por sistemas de TI têm sido propostas a fim de estabelecer novas visões sobre o comportamento de estudantes, professores e gestores [Santos et al. 2021].

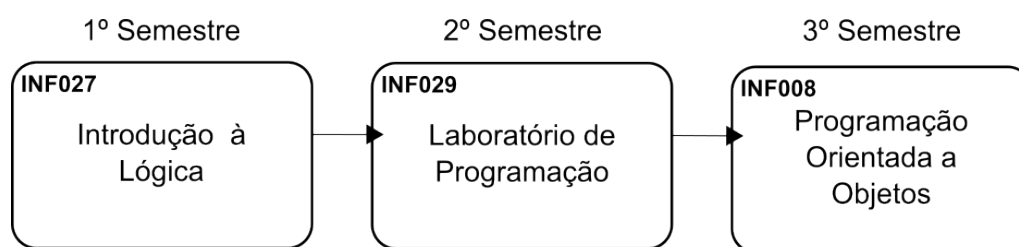
Dentre as estratégias empreendidas, destaca-se a predição de desempenho acadêmico, que pode fornecer subsídios na descoberta de padrões de comportamento. Em cursos de TI, as estratégias de predição de desempenho comumente encontradas na literatura se baseiam no estabelecimento de perfis de programação de estudantes e usam métodos de inferência estatística para deduzir o desempenho futuro [Pereira et al. 2020].

Uma outra abordagem a ser considerada é a aplicação de técnicas de Aprendizado de Máquina (AM) a partir de dados históricos estudantis armazenados em sistemas de gestão. Neste sentido, o alto desempenho preditivo visto em trabalhos anteriores é animador e atesta a viabilidade do uso dos percursos curriculares como atributos previsores.

Ademais, a ampla difusão de sistemas de TI responsáveis pela gestão acadêmica em instituições de ensino proporciona um terreno fértil para análises exploratórias de dados que podem ser facilmente replicadas em ambientes semelhantes. Essa vantagem de aplicabilidade promove um espaço oportuno e colaborativo para essas investigações, com as quais o presente trabalho pretende contribuir no contexto do curso de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS) do Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA) — Campus Salvador.

Técnicas preditivas de classificação estipulam classificadores que inferem valores de atributos categóricos relevantes. Por exemplo: considerando que cada aluno de um curso pode ser descrito por um conjunto de atributos correspondentes à sua vida acadêmica e que um deles representa a sua aprovação em uma determinada disciplina, é possível construir modelos que, levando em conta a totalidade dos registros, sugiram a valoração desse atributo para novos alunos. No caso particular do curso de ADS, observa-se a necessidade de estudar possíveis conexões de desempenho entre disciplinas, especialmente aquelas da área de programação, de modo a antecipar o desempenho acadêmico dos estudantes.

Assim, com o intuito de colaborar com esses estudos, o objetivo deste trabalho é desenvolver uma solução de processamento e análise de dados capaz de investigar a relação entre os desempenhos dos alunos nas disciplinas de Introdução à Lógica (IL) e de Programação Orientada a Objetos (POO). A disciplina de IL compõe o currículo do primeiro semestre do curso de ADS, e POO o do terceiro. A aprovação dos alunos em POO foi escolhida como rótulo, que é o atributo categórico a ser inferido; e as características selecionadas foram a nota de aprovação do aluno em IL e o número de repetições nessa disciplina. Ou seja, pretende-se avaliar quão bem o desempenho de um aluno em IL serve como atributo previsor para a inferência de sua aprovação em POO. A Figura 1 mostra de que modo as disciplinas estão encadeadas no percurso curricular.



**Figura 1. Excerto do Fluxograma do Curso de ADS do IFBA (Campus Salvador) [ADS-IFBA, Fluxograma 2025]**

Para isso, estabelecem-se os seguintes objetivos específicos: (1) promover uma melhor compreensão sobre o itinerário dos estudantes nas disciplinas de IL e POO através de gráficos gerados a partir dos registros no Sistema Unificado de Administração Pública (SUAP)<sup>1</sup>; (2) identificar os algoritmos de classificação mais apropriados para a inferência

<sup>1</sup>O SUAP foi desenvolvido pela Diretoria de Gestão de Tecnologia da Informação (DGTI) do Insti-

da aprovação dos alunos na disciplina de POO; (3) aplicar os algoritmos de classificação e (4) discutir os resultados observados.

O restante do trabalho está organizado do seguinte modo. A Seção 2 introduz conceitos de AM, aponta os algoritmos utilizados e explica resumidamente a lógica interna de cada um deles. Os trabalhos relacionados são apresentados na Seção 3. A Seção 4 apresenta as métricas de avaliação do desempenho preditor dos algoritmos, que foram escolhidas segundo as particularidades do problema. A Seção 5 descreve a proposta empreendida neste trabalho. Os resultados obtidos são avaliados na Seção 6. E na Seção 7 encontram-se as considerações finais.

## 2. Algoritmos de Predição

Homens e computadores são inerentemente aptos a realização de tarefas distintas. Por exemplo, calcular a raiz cúbica de um número imenso é fácil para computadores mas difícil para seres humanos. Por outro lado, reconhecer objetos em uma imagem é tarefa simples para um ser humano, mas tem sido tradicionalmente difícil para sistemas automatizados [Aggarwal 2018]. O Aprendizado de Máquina (AM) é o campo da Inteligência Artificial (IA) que pretende aproximar as capacidades de um computador ao potencial humano por meio de exemplos, com os quais um computador estabelece parâmetros e aprende [Kubat 2017a].

Neste campo, diversas estratégias de predição — classificação, regressão, clusterização etc. — são utilizadas segundo a natureza de cada problema. No presente trabalho, pretende-se inferir o valor do atributo categórico referente à aprovação dos alunos na disciplina de POO a partir de seus desempenhos na disciplina de IL. Configura-se, portanto, uma tarefa de classificação.

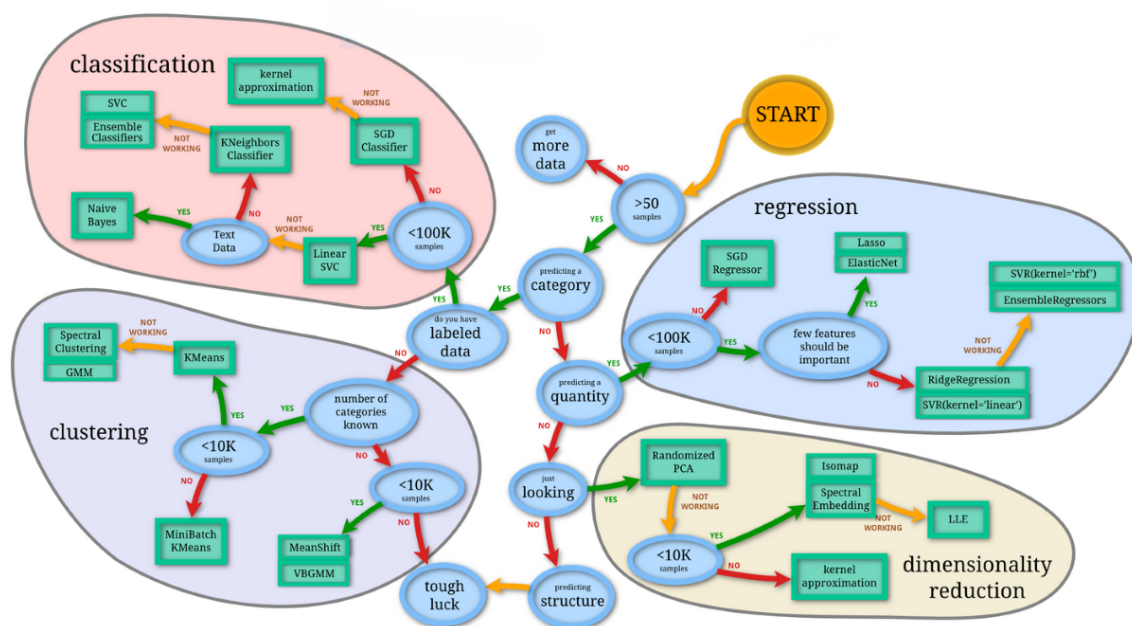


Figura 2. Guia de algoritmos [scikit-learn Docs, Algorithm Sheet 2025]

tuto Federal do Rio Grande do Norte (IFRN) para auxiliar na gestão de seus processos administrativos e acadêmicos.

Algoritmos de classificação atribuem rótulos a dados de entrada de categoria desconhecida. Considerando cada característica dos dados de entrada como um eixo ou dimensão em um espaço, o objetivo de um algoritmo de classificação é dividir esse espaço de características de modo que rótulos sejam atribuídos corretamente para cada exemplo.

A escolha dos algoritmos foi realizada segundo as recomendações encontradas na documentação da biblioteca *scikit-learn*, que forneceu os algoritmos utilizados neste trabalho [scikit-learn Docs 2025]. Os critérios seguidos podem ser consultados na Figura 2. Cada aluno foi representado na base de dados por um número de identificação, a sua nota de aprovação em IL, a quantidade de vezes que cursou essa disciplina e a sua aprovação em POO, que recebe os rótulos "Aprovado" ou "Reprovado". Não houve uso de informações de identificação pessoal e, portanto, a identidade dos alunos foi preservada.

## 2.1. Classificador K-Nearest Neighbors

É um algoritmo de classificação que aplica o cálculo da distância entre dois pontos a fim de inferir a classe de um registro. Isso é possível porque cada exemplo pode ser representado por um ponto em um espaço  $n$ -dimensional, em que a proximidade entre os pares denota similaridade mútua [Kubat 2017b].

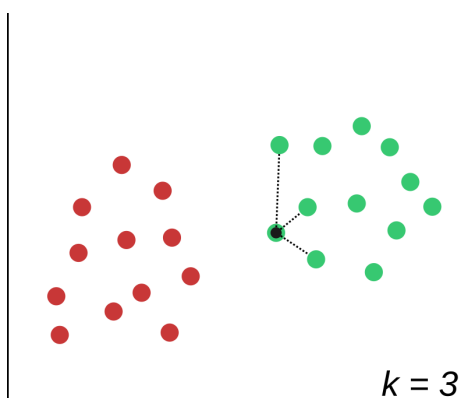


Figura 3. Fonte: Elaborado pelo autor

Assim, a partir de uma base de dados com categorias conhecidas, é possível inferir a classe de um novo registro. A nova entrada é classificada com base na categoria predominante do conjunto dos  $k$  dados mais próximos (Figura 3). Em outras palavras, se os vizinhos mais próximos forem majoritariamente de uma classe, a amostra em questão será classificada nessa categoria. Se  $k = 1$ , o ponto mais próximo definirá a classe do novo registro. Se  $k > 1$ , a categoria mais frequente entre os  $k$  registros mais próximos será atribuída.

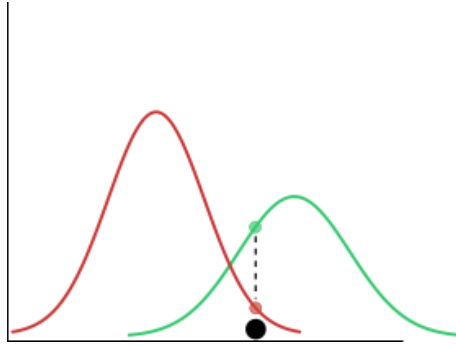
## 2.2. Naive Bayes Gaussiano

Um problema de classificação pode ser compreendido como o estabelecimento de uma hipótese  $h$  para cada dado novo  $d$ . O Teorema de Bayes [Aggarwal 2015] fornece uma maneira de calcular a probabilidade de uma hipótese considerando conhecimentos prévios. O teorema é apresentado na Equação 1.

$$P(h|d) = \frac{(P(d|h) * P(h))}{P(d)}, \quad (1)$$

onde  $P(h|d)$  é a probabilidade da hipótese  $h$  tendo o dado  $d$ .

Isso recebe o nome de probabilidade posterior;  $P(d|h)$  é a probabilidade do dado  $d$  tendo a hipótese  $h$  como verdadeira;  $P(h)$  é a probabilidade da hipótese  $h$  ser verdadeira (independentemente do dado). Isso recebe o nome de probabilidade anterior;  $P(d)$  é a probabilidade do dado  $d$  (independentemente da hipótese). Após calcular a probabilidade posterior para um conjunto de hipóteses, é possível seleccionar aquela de maior probabilidade.

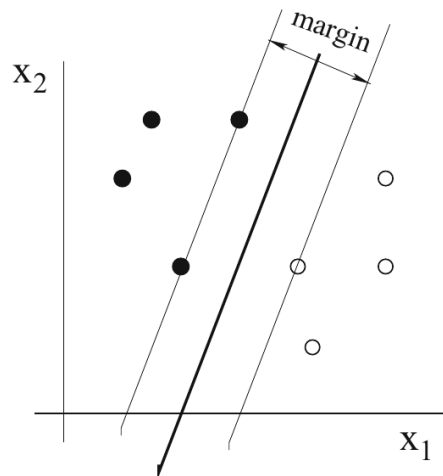


**Figura 4. Fonte: Elaborado pelo autor**

O método Naive Bayes Gaussiano apoia-se nas distribuições Gaussianas que representam os atributos dos registros. Tem-se, por exemplo, uma base de dados composta por alunos aprovados ou reprovados em uma disciplina. É possível traçar curvas de distribuição Gaussiana que representam as notas dos alunos para cada classe. Um novo registro sem classificação prévia é introduzido ao conjunto. Aplica-se o Teorema de Bayes considerando as probabilidades correspondentes para cada classe e a categoria de máxima probabilidade é atribuída ao novo aluno (Figura 4). A técnica é conhecida por ter alto viés, o que pode significar melhor desempenho em bases de dados que contenham um número pequeno de exemplos.

### 2.3. Máquina de Vetor de Suporte (MVS)

O objetivo de um algoritmo de MVS é identificar o hiperplano ideal que separa um conjunto de objetos em diferentes classes (Figura 5).



**Figura 5. Fonte: An Introduction to Machine Learning, Página 86**

A tarefa do classificador é selecionar o vetor de suporte que maximiza a distância entre a fronteira de decisão e os registros mais próximos (margem) [Kubat 2017c]. O classificador de melhor desempenho é selecionado através de um processo de validação cruzada.

### 3. Trabalhos Relacionados

Com a expansão da área de Inteligência Artificial, algoritmos de predição tem sido amplamente utilizados nos mais diversos campos de conhecimento. Neste trabalho, buscamos investigar o uso destes algoritmos quando aplicados para antever o comportamento de estudantes dentro de seus respectivos itinerários acadêmicos.

Em [Garcia et al. 2023] foi desenvolvida uma ferramenta de predição de desempenho para estudantes de graduação, especificamente para as disciplinas a serem cursadas. Os modelos são concebidos considerando as aprovações obtidas pelos alunos em todas as disciplinas já cursadas. Esse vetor de características é submetido à 4 modelos de predição, que produzem inferências das aprovações em disciplinas ainda não cursadas. Foram computados dados de 15 disciplinas de um curso de Ciência da Computação no período letivo de 2022.1. Os resultados demonstraram uma alta taxa de acerto, com uma média de 86% de predições corretas para o desempenho em geral, e 90% para predição específica de casos de reprovação.

[Trajano 2023] fez uso dos algoritmos MVS, *Decision Tree*, *K-Nearest Neighbors* e *Logistic Regression* a fim de selecionar o melhor modelo de predição para a classificação de alunos como evadidos ou concluintes em cursos superiores do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB). Utilizou-se um conjunto de 16 atributos previsores extraídos da plataforma de gestão Nilo Peçanha. Não foram incluídos atributos de desempenho, mas sim características individuais como sexo, renda, cor, idade, tipo de curso etc. Uma limitação do trabalho foi a impossibilidade de acesso a dados escolares como as notas dos alunos que contribuiriam para resultados mais rigorosos.

Uma tese realizada na Universidade Federal do Rio de Janeiro (UFRJ) apresentou uma arquitetura de predição de desempenho baseada em três camadas (de dados, de aplicação e de apresentação), utilizando exclusivamente dados acadêmicos históricos [Manhães 2015]. O projeto proposto trabalha acoplado ao sistema de gestão acadêmica e automatiza a predição do desempenho dos estudantes (progresso ou não progresso) a cada semestre letivo. Os resultados obtidos nos estudos de caso avaliados mostraram que a arquitetura possibilita o monitoramento do progresso dos graduandos a cada semestre letivo e ajuda a identificar aqueles que apresentam maiores riscos de abandonar o sistema educacional.

A fim de colaborar com as investigações sobre o tema da evasão estudantil, [Santos et al. 2021] estabeleceu, a partir de dados referentes ao desempenho de alunos de uma universidade brasileira, um classificador binário (Evasão ou Formatura) que resultou em um modelo com taxa de acerto entre 79% e 98%. O trabalho tinha como objetivo verificar se é possível prever a evasão somente com dados de desempenho dos alunos durante curso. As árvores de decisão promoveram modelos eficientes e de fácil compreensão, o que deve facilitar a assimilação dos resultados obtidos pelos atores educacionais.

Por fim, nota-se que os resultados observados nos trabalhos elencados são encorajadores quanto ao uso de dados históricos estudantis em tarefas de predição de desempe-

no, o que se pretende realizar no presente trabalho para o caso já descrito. Contudo, aqui temos um escopo reduzido que está apoiado na intuição básica de que os desempenhos dos alunos nas disciplinas de IL e POO estão associados.

Propostas				
	Escopo	Dados	Número de algoritmos	Interface de usuário
Garcia et al. 2023	Aprovação	Desempenho	3	Sim
Trajano 2023	Evasão	Outros	4	Sim
Manhães 2015	Aprovação	Desempenho	11	Não
Santos et al. 2021	Evasão	Desempenho	2	Não
ADS.predict	Aprovação	Desempenho	3	Sim

#### 4. Métricas de Avaliação

A avaliação correta de modelos de predição é uma das mais importantes questões em reconhecimento de padrões [Ferri et al. 2008]. Toda métrica utilizada para avaliar modelos de classificação estabelece suposições sobre o problema ou sobre o que é importante no problema. Assim, as métricas escolhidas devem ser compatíveis àquilo que é considerado importante compreender sobre o modelo e suas inferências. No caso de classes desbalanceadas, a dificuldade é enfatizada porque os procedimentos relativamente robustos usados em dados balanceados falham drasticamente [Haibo and Yunqian 2013].

Para escolher métricas que privilegiem os aspectos de maior interesse, faz-se necessário compreender bem a natureza de cada problema de classificação. O foco deste trabalho está na classe minoritária, ou classe positiva, que é a do grupo de alunos reprovados na disciplina de POO. O maior interesse é, através dos algoritmos empreendidos, apontar com boa taxa de acerto aqueles alunos que serão reprovados, de modo a facilitar a ação dos atores educacionais na proposição de mecanismos de assistência. Assim, o critério de priorização das métricas deve ser a capacidade de expressão da ocorrência dos falsos negativos (alunos reprovados mas classificados como aprovados). Nesta seção apresentaremos as métricas selecionadas para a avaliação dos modelos de predição aplicados neste trabalho.

##### 4.1. Acurácia

É uma métrica de avaliação de modelos de classificação que indica a proporção das previsões corretas [Kubat 2017d]. É formalmente definida como visto na Equação 2.

$$\text{Acurácia} = \frac{\text{Número total de previsões corretas}}{\text{Número total de previsões}} \quad (2)$$

Para bases de dados desbalanceadas a acurácia pode causar uma impressão incorreta sobre o desempenho dos modelos. Por exemplo: se 80% dos dados de um conjunto pertencem à classe A e 20% à classe B, pode-se obter uma Acurácia de 0,8 sem que se tenha inferido corretamente nenhum registro de classe B.

## 4.2. Precisão

A precisão aponta a proporção de identificações positivas corretas. Ou seja, a razão dos registros positivos classificados corretamente sobre todos os registros classificados como positivos. Um modelo que não produz falsos positivos tem precisão de 1,0. É formalmente definida de acordo com a Equação 3 [Kubat 2017e].

$$\text{Precisão} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos positivos}}, \quad (3)$$

## 4.3. Revocação

A revocação é a proporção dos positivos verdadeiros identificados corretamente. Ou seja, é a razão dos registros positivos classificados corretamente sobre todos os registros realmente positivos. Um modelo que não produz falsos negativos tem revocação de 1,0. É formalmente definida segundo a Equação 4 [Kubat 2017e].

$$\text{Revocação} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos negativos}} \quad (4)$$

No contexto deste trabalho, os falsos negativos — alunos reprovados classificados como aprovados — são mais importantes que os falsos positivos — alunos aprovados classificados como reprovados. Assim, estabelecemos que a revocação tem precedência em relação à precisão.

## 4.4. Pontuação $F\beta$

É uma métrica de avaliação que combina as pontuações de precisão e revocação de um modelo.  $F\beta$  usa um fator positivo real  $\beta$ , onde  $\beta$  determina a importância de cada critério. Se  $\beta > 1$ , então o peso maior é conferido à revocação. Se  $\beta < 1$ , então o peso é dado à precisão. O valor atribuído a  $\beta$  nos experimentos do presente trabalho será 2,0; o que é consistente com o maior interesse nos falsos negativos. A pontuação  $F\beta$  é definida segundo a Equação 5 [Kubat 2017f].

$$F\beta = (1 + \beta^2) \frac{\text{Precisão} \times \text{Revocação}}{(\beta^2 \times \text{Precisão}) + \text{Revocação}} \quad (5)$$

## 4.5. Precisão na classe negativa

Aponta a proporção de identificações negativas corretas. Ou seja, a razão dos registros negativos classificados corretamente sobre todos os registros classificados como negativos. É formalmente definida conforme a Equação 6.

$$\text{Precisão (-)} = \frac{\text{Verdadeiros negativos}}{\text{Verdadeiros negativos} + \text{Falsos negativos}} \quad (6)$$

#### 4.6. Revocação na classe negativa

A revocação na classe negativa sumariza o quão bem a classe negativa foi categorizada. É a razão dos registros negativos classificados corretamente sobre todos os registros realmente negativos. É formalmente definida segundo a Equação 7 [Kubat 2017g].

$$\text{Revocação (-)} = \frac{\text{Verdadeiros negativos}}{\text{Verdadeiros negativos} + \text{Falsos positivos}} \quad (7)$$

### 5. Proposta

A análise preditiva realizada neste trabalho busca identificar os alunos com maior probabilidade de reprovação na disciplina de POO. Essa iniciativa pretende potencializar a ação dos atores do processo educacional, que podem estabelecer meios de assistência direcionados às necessidades dos discentes. Para isso, propõe-se a implementação da plataforma ADS.predict que utiliza estratégias de visualização de dados e técnicas de AM para estabelecer novas visões sobre o itinerário dos estudantes. A investigação consiste, mais especificamente, em adotar o desempenho dos alunos na disciplina de IL como atributo previsor das inferências de suas aprovações em POO e, assim, avaliar o quão fortemente estão associados os seus desempenhos nessas disciplinas.

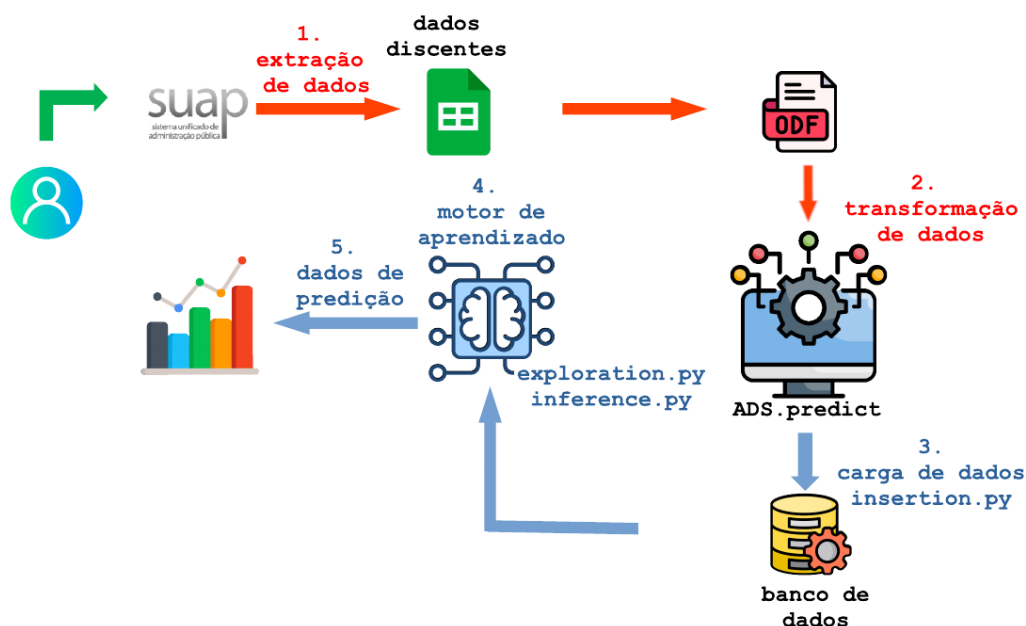


Figura 6. Fluxo de dados, ADS.predict

O fluxo de dados que resume a estratégia empreendida na construção do sistema pode ser observado na Figura 6. A primeira etapa descrita é a **extração de dados**. Essa etapa está fortemente acoplada ao SUAP, que é responsável pela gestão acadêmica no IFBA e foi utilizado para a obtenção das características de interesse [IFRN 2023]. O sistema é utilizado por vários Institutos Federais para os quais o IFRN cede, de forma colaborativa, o seu uso.

As informações extraídas do SUAP foram: (a) as notas finais dos estudantes na disciplina de IL, (b) o número de vezes que cada discente cursou essa disciplina e (c)

suas aprovações em POO. Embora essas informações possam ser coletadas em qualquer formato, parametrizamos o `ADS.predict` para receber arquivos `.odf`, `.ods` e `.odt`.

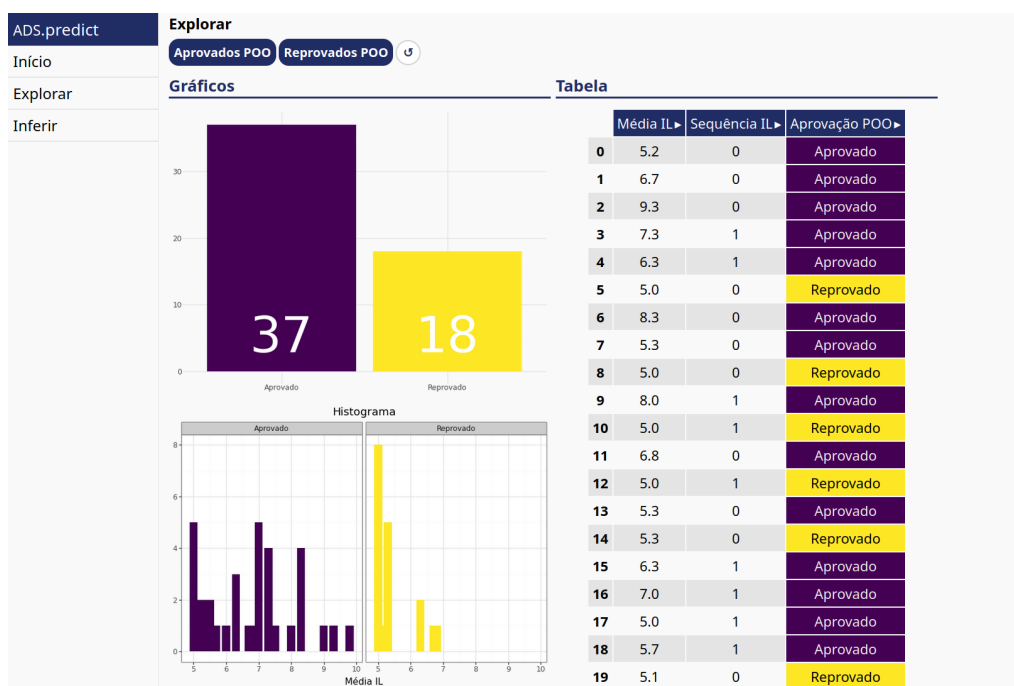


Figura 7. Visualização de dados, interface de usuário do `ADS.predict`

A próxima etapa é a **transformação**, que converte os dados importados em uma estrutura manipulável. Depois, segue-se para a terceira etapa, que é a **carga de dados**, em que a estrutura é armazenada na base de dados. Por fim, o **motor de aprendizado**, que implementa os algoritmos de AM, gera os **dados de predição** que são apresentados ao usuário através de relatórios gráficos. A figura 7 apresenta uma das páginas da interface de usuário da aplicação, que inclui as funcionalidades de visualização de dados.

### 5.1. Arquitetura da Aplicação

A arquitetura do `ADS.predict` implementa o modelo de três camadas (*model-view-controller*, ou MVC), assim desenvolvido:

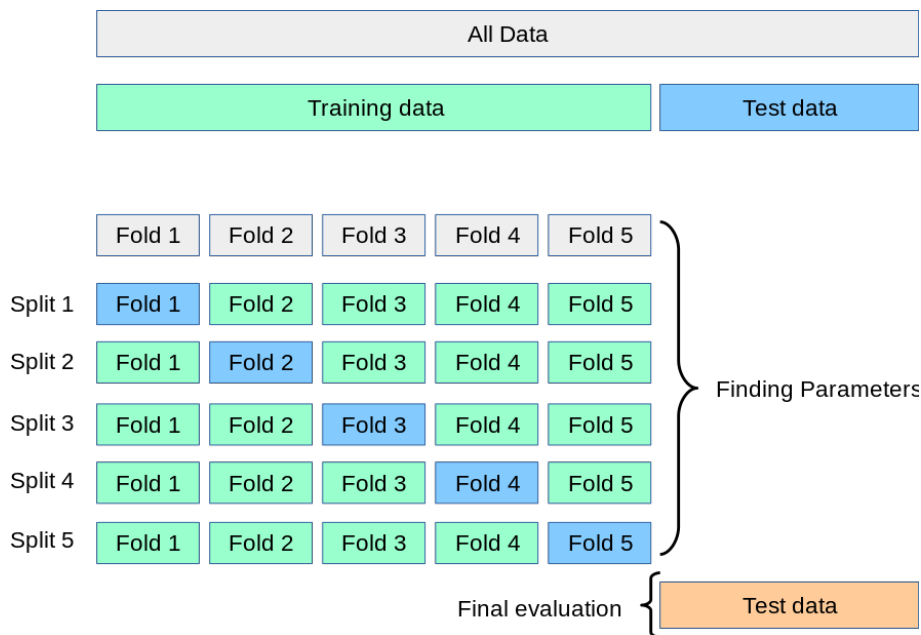
1. **camada de apresentação:** essa interface foi construída a partir das linguagens HTML e CSS. Possibilita a exibição dos componentes gráficos, inclusive as métricas retornadas pelos algoritmos de AM;
2. **camada de negócio:** foi desenvolvida na linguagem *Python* com o *microframework* de servidores *Flask* e faz uso das bibliotecas *pandas* (manipulação de registros), *matplotlib* (traçado de gráficos), *plotnine* (gramática de gráficos), *numpy* (operações sobre matrizes) e *scikit-learn* (algoritmos de AM);
3. **camada de dados:** foi implementada através de um banco de dados SQLite, que consiste em uma tabela de registros correspondentes a cada aluno.

## 6. Resultados

### 6.1. Validação

Em tarefas de classificação são estabelecidos dois conjuntos: um conjunto de treinamento — dados de entrada utilizados pelos algoritmos de modo a definir parâmetros de classifi-

cação — e um conjunto de testes, o qual recebe os rótulos inferidos a partir dos critérios determinados na etapa anterior. O desempenho dos algoritmos pode variar dramaticamente de acordo com a porção dos dados utilizada em cada execução. Se, por exemplo, o primeiro quarto de registros contiver a maior parte dos casos atípicos, e for selecionado em um experimento como o conjunto de testes, o modelo certamente não responderá bem. Assim, em vez de exaustivamente buscar as combinações de melhor desempenho, aplica-se o método da validação cruzada, que automatiza o processo e sumariza os resultados.



**Figura 8. Validação cruzada [scikit-learn Docs, Cross-validation 2025]**

Neste trabalho, utilizou-se o módulo de validação cruzada da biblioteca *scikit-learn* [scikit-learn Docs, Cross-validation 2025] com a estratégia de divisão *5-fold*, em que o conjunto de treinamento é dividido em cinco grupos. Como ilustrado na Figura 8, os algoritmos são treinados com um dos subconjuntos como o conjunto de testes e os demais como o conjunto de treinamento, o que se repete até que todos os subconjuntos tenham sido avaliados.

Assim, criamos uma base de dados composta por 55 alunos (18 de categoria "Reprovado" e 37 de categoria "Aprovado") que considerou estudantes que ingressaram no período de 2016 à 2018. O corte no ano de 2018, se deu a fim de eliminar possíveis dados enviesados. Por exemplo, alunos que ingressaram no primeiro semestre letivo de 2019 (2019.1), se aprovados em todas as disciplinas que são pré-requisitos para POO, deveriam se matricular em POO em 2020.1, ano em que aconteceu a pandemia de COVID-19. Durante este período, tanto as aulas quanto as avaliações seguiram critérios específicos, o que poderia distorcer o resultado final. Da mesma forma, antes do ano de 2016, os dados acadêmicos dos discentes do IFBA foram armazenados em uma aplicação disponível apenas para gestores.

Cada aluno foi representado por duas características (Nota de aprovação em IL e Repetições em IL) e um rótulo (Aprovação em POO) que é a categoria inferida para o conjunto de testes. O desempenho reportado é a média dos valores computados a cada

ciclo. A etapa de avaliação final foi desprezada já que a totalidade dos registros foi utilizada durante o processo. Cada algoritmo classificador foi configurado através de uma instância de uma classe correspondente que pode ser ajustada por meio de parâmetros de construtor.

No caso do classificador K-Nearest Neighbors, os parâmetros com impacto significativo nas pontuações obtidas foram *n-neighbors* (número de vizinhos votantes) e *weights*. Observou-se que maiores pontuações foram obtidas com *n-neighbors* = 5; mesmo que resultando em valores inferiores de precisão, a configuração *weights* = 'distance' foi escolhida, por obter uma maior revocação, em vez de *weights* = 'uniform'. Isso quer dizer que a distância dos vizinhos foi levada em consideração nos votos que determinam a classe do novo registro. A configuração final pode ser observada no excerto abaixo.

---

```
sklearn.neighbors.KNeighborsClassifier(n_neighbors=5,
    weights='distance', algorithm='auto', leaf_size=30, p=2,
    metric='minkowski', metric_params=None, n_jobs=None)
```

---

A implementação do classificador Naive Bayes Gaussiano permite estabelecer, através do parâmetro *priors*, as probabilidades anteriores para cada classe. Se não especificado, os valores são ajustados segundo a distribuição encontrada na base de dados. Configurar as probabilidades anteriores para 40% na classe "Aprovado" e 60% na classe "Reprovado" alcançou melhores resultados de revocação, mantendo a acurácia em níveis aceitáveis.

---

```
sklearn.naive_bayes.GaussianNB(priors=[0.4 , 0.6],
    var_smoothing=1e-09)
```

---

O modelo MVS tem diversos parâmetros importantes para o processo de treinamento. A lista a seguir descreve alguns deles:

- *C* (float): parâmetro de regularização; a força da regularização é inversamente proporcional à *C*.
- *kernel* (linear, poly, rbf, sigmoid, precomputed ou callable): especifica a função utilizada para calcular o hiperplano que define a fronteira de decisão.
- *gamma* (scale, auto ou float): determina o coeficiente da função do *kernel*.
- *class-weight* (dict ou balanced, default=None): determina um peso anterior para cada classe.

O grande número de combinações possíveis impôs o uso da técnica de busca de matrizes, que aplica o método de validação cruzada variando os parâmetros especificados pelo usuário. O resultado é uma matriz que exhibe as combinações e suas pontuações calculadas para cada métrica especificada. O módulo *GridSearchCV* da biblioteca *scikit-learn* foi utilizado para esse fim. A técnica foi aplicada com as configurações de *kernel* rbf e poly, já que apresentaram resultados superiores em testes iniciais.

A matriz resultante para o *kernel* rbf apresentou 50 combinações, as quais algumas podem ser observadas na figura 9. Para a escolha dos parâmetros apropriados, priorizou-se alcançar ótimos resultados de revocação mantendo níveis aceitáveis de precisão. A combinação de parâmetros de número 15 foi arbitrariamente selecionada entre outras que obtiveram os mesmos resultados de revocação e precisão. A configuração final pode ser

	Parâmetros	Revocação	±	Ranking	Precisão	±	Ranking
0	{'C': 0.25, 'class_weight': None, 'gamma': 'scale'}	0.10	0.20	48	0.07	0.13	49
1	{'C': 0.25, 'class_weight': None, 'gamma': 'auto'}	0.10	0.20	48	0.07	0.13	49
2	{'C': 0.25, 'class_weight': None, 'gamma': 1}	0.30	0.27	46	0.37	0.37	43
3	{'C': 0.25, 'class_weight': None, 'gamma': 10}	0.23	0.29	47	0.23	0.29	47
4	{'C': 0.25, 'class_weight': None, 'gamma': 100}	0.10	0.12	48	0.20	0.24	48
5	{'C': 0.25, 'class_weight': 'balanced', 'gamma': 'scale'}	0.72	0.16	11	0.58	0.17	18
6	{'C': 0.25, 'class_weight': 'balanced', 'gamma': 'auto'}	0.72	0.16	11	0.58	0.17	18
7	{'C': 0.25, 'class_weight': 'balanced', 'gamma': 1}	0.72	0.16	11	0.58	0.17	18
8	{'C': 0.25, 'class_weight': 'balanced', 'gamma': 10}	0.67	0.09	23	0.65	0.23	1
9	{'C': 0.25, 'class_weight': 'balanced', 'gamma': 100}	0.57	0.18	32	0.58	0.11	15
10	{'C': 0.5, 'class_weight': None, 'gamma': 'scale'}	0.37	0.31	44	0.27	0.23	45
11	{'C': 0.5, 'class_weight': None, 'gamma': 'auto'}	0.37	0.31	44	0.27	0.23	45
12	{'C': 0.5, 'class_weight': None, 'gamma': 1}	0.42	0.26	42	0.37	0.19	43
13	{'C': 0.5, 'class_weight': None, 'gamma': 10}	0.52	0.15	36	0.60	0.23	8
14	{'C': 0.5, 'class_weight': None, 'gamma': 100}	0.40	0.16	43	0.60	0.20	8
15	{'C': 0.5, 'class_weight': 'balanced', 'gamma': 'scale'}	0.77	0.12	3	0.60	0.15	11
16	{'C': 0.5, 'class_weight': 'balanced', 'gamma': 'auto'}	0.77	0.12	3	0.60	0.15	11

Figura 9. Saída do GridSearchCV com kernel rbf

observada no excerto abaixo.

---

```
sklearn.svm.SVC(C=0.5, kernel='rbf', gamma='scale', coef0=0.0,
shrinking=True, probability=False, tol=0.001,
cache_size=200, class_weight='balanced', verbose=False,
max_iter=-1, decision_function_shape='ovr',
break_ties=False, random_state=None)
```

---

22	{'C': 0.75, 'class_weight': 'balanced', 'gamma': 1}	0.72	0.16	16	0.43	0.09	25
23	{'C': 0.75, 'class_weight': 'balanced', 'gamma': 10}	0.82	0.15	9	0.51	0.13	10
24	{'C': 1, 'class_weight': None, 'gamma': 'scale'}	0.23	0.29	33	0.30	0.40	33
25	{'C': 1, 'class_weight': None, 'gamma': 'auto'}	0.18	0.26	34	0.27	0.39	34
26	{'C': 1, 'class_weight': None, 'gamma': 1}	0.67	0.09	21	0.68	0.19	1
27	{'C': 1, 'class_weight': None, 'gamma': 10}	0.57	0.18	27	0.48	0.15	15
28	{'C': 1, 'class_weight': 'balanced', 'gamma': 'scale'}	0.82	0.15	9	0.43	0.04	28
29	{'C': 1, 'class_weight': 'balanced', 'gamma': 'auto'}	0.87	0.16	6	0.45	0.07	19
30	{'C': 1, 'class_weight': 'balanced', 'gamma': 1}	0.72	0.16	16	0.43	0.09	25
31	{'C': 1, 'class_weight': 'balanced', 'gamma': 10}	0.82	0.15	9	0.51	0.13	10
32	{'C': 10, 'class_weight': None, 'gamma': 'scale'}	0.67	0.09	21	0.68	0.19	1
33	{'C': 10, 'class_weight': None, 'gamma': 'auto'}	0.67	0.09	21	0.68	0.19	1
34	{'C': 10, 'class_weight': None, 'gamma': 1}	0.52	0.15	31	0.60	0.23	7
35	{'C': 10, 'class_weight': None, 'gamma': 10}	0.57	0.18	27	0.48	0.15	15
36	{'C': 10, 'class_weight': 'balanced', 'gamma': 'scale'}	0.72	0.16	16	0.45	0.09	20
37	{'C': 10, 'class_weight': 'balanced', 'gamma': 'auto'}	0.72	0.16	16	0.45	0.09	20
38	{'C': 10, 'class_weight': 'balanced', 'gamma': 1}	0.88	0.15	5	0.52	0.12	9
39	{'C': 10, 'class_weight': 'balanced', 'gamma': 10}	0.82	0.15	9	0.51	0.13	10

Figura 10. Saída do GridSearchCV com kernel poly

A matriz resultante para o kernel poly apresentou 40 combinações, as quais algumas podem ser observadas na figura 10. A combinação de parâmetros de número 38 foi arbitrariamente selecionada entre outras que obtiveram os mesmos resultados de revocação e precisão. O excerto abaixo mostra a configuração final.

---

```
sklearn.svm.SVC(C=10, kernel='poly', gamma=1, coef0=0.0,
shrinking=True, probability=False, tol=0.001,
```

```
cache_size=200, class_weight='balanced', verbose=False,
max_iter=-1, decision_function_shape='ovr',
break_ties=False, random_state=None)
```

---

Tendo selecionado a melhor combinação de parâmetros para cada algoritmo — aquelas que priorizaram a revocação, o que é consistente com o interesse em minimizar a quantidade de falsos negativos — é possível explorar os resultados obtidos pelos algoritmos para cada métrica relevante.

### 6.1.1. Taxa de acerto geral

Algoritmos de Predição		
	Acurácia	Desvio-padrão
KNNNeighbors	0.75	0.12
NB Gaussiano	0.71	0.13
MVS (rbf)	0.75	0.12
MVS (poly)	0.69	0.11

A acurácia indica a proporção das previsões corretas e aponta para a habilidade geral de inferência de cada algoritmo. Os resultados obtidos revelam alguma apropriabilidade do uso das técnicas empreendidas. Porém, os resultados de taxa de acerto geral para base de dados desbalanceadas nada dizem sobre o sucesso das técnicas de predição para cada classe. Ademais, considerando que a classe de maior interesse para este trabalho (a positiva) constitui apenas 1/3 dos registros, os resultados de acurácia alcançados são especialmente ineficazes em sua expressão. Assim, é importante expandir o conjunto de métricas de modo a incluir aquelas sensíveis à cada categoria.

### 6.1.2. Classe negativa

Algoritmos de Predição		
	Precisão (-)	Revocação (-)
KNNNeighbors	0.85 ± 0.11	0.76 ± 0.1
NB Gaussiano	0.85 ± 0.08	0.68 ± 0.16
MVS (rbf)	0.86 ± 0.08	0.73 ± 0.16
MVS (poly)	0.92 ± 0.11	0.6 ± 0.11

A precisão na classe negativa aponta a proporção de identificações negativas corretas e a revocação na classe negativa sumariza o quão bem essa categoria foi inferida. Os resultados obtidos demonstram uma boa capacidade dos algoritmos em classificar como aprovados alunos realmente aprovados. Contudo, os valores inferiores de revocação revelam menor habilidade em capturar todo o conjunto de alunos aprovados. Ou seja, uma quantidade não desprezível de alunos realmente aprovados foi classificada erroneamente. Em vista do interesse deste trabalho em apontar os alunos com maior probabilidade de reprovação para que sejam assistidos, errar por excesso de classificações positivas é mais tolerável que errar por excesso de classificações negativas.

### 6.1.3. Classe positiva

Algoritmos de Predição			
	Precisão	Revocação	F2
KNNNeighbors	0.59 ± 0.16	0.72 ± 0.16	0.69 ± 0.16
NB Gaussiano	0.56 ± 0.18	0.77 ± 0.12	0.71 ± 0.14
MVS (rbf)	0.6 ± 0.15	0.77 ± 0.12	0.72 ± 0.12
MVS (poly)	0.52 ± 0.12	0.88 ± 0.15	0.77 ± 0.13

A precisão aponta a proporção de identificações positivas corretas, a revocação é a proporção dos positivos verdadeiros identificados corretamente e a pontuação F2 combina as métricas anteriores atribuindo maior importância à revocação. As taxas inferiores de precisão e os bons resultados de revocação revelam uma dinâmica de oposição entre as duas métricas. Quanto mais se pretenda capturar a ocorrência dos alunos reprovados, maior será o número de alunos aprovados erroneamente classificados. Se todos os alunos fossem classificados como reprovados, revocação e precisão seriam respectivamente 1,0 e 0,33.

$$\text{Revocação} = \frac{\text{Verdadeiros positivos (18)}}{\text{Verdadeiros positivos (18)} + \text{Falsos negativos (0)}} = 1,0.$$

$$\text{Precisão} = \frac{\text{Verdadeiros positivos (18)}}{\text{Verdadeiros positivos (18)} + \text{Falsos positivos (37)}} = 0,3272\dots$$

Portanto, os modelos devem ser considerados menos aptos na avaliação da classe positiva quanto mais próximos os resultados de precisão estejam desse limiar.

## 6.2. Inferências

Tendo selecionado os melhores parâmetros de configuração para cada algoritmo, é possível realizar inferências de modo a obter as classificações para cada aluno do conjunto de testes. Devido ao número reduzido de registros disponíveis, a divisão da base de dados em conjuntos de treinamento e testes (respectivamente 80% e 20%) ocasiona resultados bastante diferentes para cada divisão distinta. Espera-se que com a expansão da base de dados os resultados apresentados sejam cada vez mais consistentes.

A execução de cada algoritmo na página de inferências do `ADS.predict` retorna uma tabela que exibe os rótulos reais e os inferidos para cada aluno; também são computadas as métricas relevantes (Figura 11). Além disso, as inferências são exibidas através de uma matriz de confusão e de uma fronteira de decisão. Uma matriz de confusão permite a visualização do desempenho de um algoritmo de maneira resumida e uma fronteira de decisão é uma hipersuperfície que separa diferentes classes em um espaço de características; representa a área em que a predição do algoritmo muda de uma classe para a outra.



Figura 11. Inferências, interface de usuário do ADS.predict

## 7. Conclusão

Este trabalho propôs uma ferramenta capaz de investigar a relação entre os desempenhos dos alunos do referido curso nas disciplinas de Introdução à Lógica e Programação Orientada a Objetos. Isso foi realizado por meio de estratégias de visualização de dados e técnicas de predição de desempenho.

O ADS.predict foi desenvolvido com as linguagens *Python*, *HTML* e *CSS*. Fez uso do *microframework Flask*, de uma base de dados *SQLite* e das bibliotecas *pandas*, *matplotlib*, *plotnine*, *numpy* e *scikit-learn*. Os dados estudantis empregados foram extraídos do SUAP, sistema de gestão utilizado pelo curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFBA (Campus Salvador).

Os resultados de acurácia obtidos durante o processo de validação mostraram-se ineficazes em expressar o desempenho dos algoritmos, especialmente na classe positiva; o que é evidente por causa da distribuição das classes na base de dados. Os resultados da validação na classe negativa demonstraram uma boa capacidade dos algoritmos em classificar corretamente alunos aprovados.

Os resultados da validação na classe positiva revelaram uma dinâmica de oposição entre precisão e revocação. Quanto mais se pretenda capturar a ocorrência dos alunos reprovados, maior será o número de alunos aprovados erroneamente classificados. No caso do algoritmo MVS com *kernel rbf*, a precisão média, mesmo que modesta, foi 83% superior ao limiar de inaptidão sugerido pela distribuição das classes na base de dados; o que aponta para a validade da intuição inicial de que os desempenhos dos alunos nas disciplinas estão relacionados. Os números de revocação obtidos foram satisfatórios, considerando o interesse em identificar os discentes com maior probabilidade de reprovação.

A quantidade reduzida de registros disponíveis limitou a qualidade das inferências. Contudo, os processos estabelecidos poderão ser retomados quando houver maior

disponibilidade de dados. Por fim, uma oportunidade a ser explorada é a expansão do escopo das investigações, incluindo outras características dos alunos como atributos previsores (desempenho em diversas disciplinas, frequência etc.).

## Referências

- ADS-IFBA, Fluxograma (2025). Endereço do fluxograma do curso de ADS. <https://portal.ifba.edu.br/salvador/ensino/cursos/superior/graduacao/ads/documentos/fluxograma-ads-rev03-vf.pdf>. Acesso: 24-02-2025.
- Aggarwal, C. (2015). *Data Mining The Textbook*. Springer. Página 306.
- Aggarwal, C. (2018). *Neural Networks and Deep Learning*. Springer. Página 3.
- Czibula, G., Mihai, A., and Crivei, L. (2019). Sprar: A novel relational association rule mining classification model applied for academic performance prediction. *Procedia Computer Science, Volume 159*.
- Ferri, C., Hernández-Orallo, J., and Modroi, R. (2008). An experimental comparison of performance measures for classification. *Elsevier Pattern Recognition Letters, Volume 30*.
- Garcia, L. M. L. S., Lara, D. F., and Salcedo, R. (2023). Ferramenta para predição do desempenho acadêmico no ensino superior. *Congresso Brasileiro de Informática na Educação*.
- Haibo, H. and Yunqian, M. (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley-IEEE Press. Página 189.
- IFRN (13-02-2023). IFRN - Serviços SUAP. <https://portal.ifrn.edu.br/institucional/tecnologia-da-informacao/servicos/suap/>. Acesso: 20-08-2024.
- Kubat, M. (2017a). *An Introduction to Machine Learning*. Springer. Página xi.
- Kubat, M. (2017b). *An Introduction to Machine Learning*. Springer. Página 44.
- Kubat, M. (2017c). *An Introduction to Machine Learning*. Springer. Página 85.
- Kubat, M. (2017d). *An Introduction to Machine Learning*. Springer. Página 212.
- Kubat, M. (2017e). *An Introduction to Machine Learning*. Springer. Página 215.
- Kubat, M. (2017f). *An Introduction to Machine Learning*. Springer. Página 219.
- Kubat, M. (2017g). *An Introduction to Machine Learning*. Springer. Página 220.
- Manhães, L. M. B. (2015). *PREDIÇÃO DO DESEMPENHO ACADÊMICO DE GRADUANDOS UTILIZANDO MINERAÇÃO DE DADOS EDUCACIONAIS*. Pós-graduação em Engenharia de Sistemas e Computação. PhD thesis, Universidade Federal do Rio de Janeiro.
- Pereira, F. D., Souza, L. M., Oliveira, E. H. T., Oliveira, D. B. F., and Carvalho, L. S. G. (2020). Predição de desempenho em ambientes computacionais para turmas de programação: um mapeamento sistemático da literatura. *Anais do XXXI Simpósio Brasileiro de Informática na Educação*.

Santos, C. H. D. C., Martins, S. L., and Plastino, A. (2021). É possível prever evasão com base apenas no desempenho acadêmico? *Simpósio Brasileiro de Informática na Educação*, 32.

scikit-learn Docs (2025). Endereço da documentação da biblioteca scikit-learn. <https://scikit-learn.org/stable/index.html>. Acesso: 15-01-2025.

scikit-learn Docs, Algorithm Sheet (2025). Endereço da documentação da biblioteca scikit-learn. [https://scikit-learn.org/1.3/tutorial/machine\\_learning\\_map/index.html](https://scikit-learn.org/1.3/tutorial/machine_learning_map/index.html). Acesso: 15-01-2025.

scikit-learn Docs, Cross-validation (2025). Endereço da documentação da biblioteca scikit-learn. [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html). Acesso: 17-01-2025.

Trajano, F. M. V. (2023). Análise preditiva e exploratória de dados para auxiliar no combate à evasão estudantil nos cursos superiores do IFPB, Instituto Federal de Educação, Ciência e Tecnologia da Paraíba.