Avaliação de dados para identificação de crianças atípicas na pré-escola

Júlia de Araújo Pires[1], Gustavo Henrique Silveira[1], Ayslan Trevizan Possebom[1], Lynnier Beatrys Ruiz Aylon[2]

[1]Instituto Federal do Paraná (IFPR) - Campus Paranavaí
Rua José F. Tequinha, 1400, Jardim das Nações. CEP 87703-536 – Paranavaí, PR - Brasil.

[2]Universidade Estadual de Maringá. Departamento de Ciência da Computação. Av. Colombo, 5790, Jd. Universitário. CEP 87020-900 - Maringá - PR

julia03.pires08@gmail.com, gustavosiveiraghs@gmail.com, ayslan.possebom@ifpr.edu.br, lbruiz@uem.br

Abstract. *The diagnosis of neurodivergent people involves a multidisciplinary assessment and time-consuming and expensive family interactions. This study seeks to optimize the diagnosis process for Autism Spectrum Disorder (ASD) through the use of data processing frameworks and machine learning algorithms. The existing database was refined based on DSM-5 and Q-CHAT 10 criteria. The implementation was carried out in Python, based on a Dataset available online. The results were analyzed to understand which algorithm is the most efficient in this case.*

Resumo. *O diagnóstico de neurodivergentes envolve uma avaliação multidisciplinar e interações familiares demoradas e dispendiosas. Este estudo busca otimizar o processo de diagnóstico do Transtorno do Espectro Autista (TEA) por meio do uso de frameworks de processamento de dados e algoritmos de machine learning. A base de dados existente foi refinada com base nos critérios do DSM-5 e Q-CHAT 10. A implementação foi realizada em Python, com base em um Base de Dados disponível online. Os resultados foram analisados para entender qual algoritmo é o mais eficiente nesse caso.*

1. Introdução

Segundo o DSM-5 Manual Diagnóstico e Estatístico de Transtornos Mentais 5.ª [APA, 2015], o Transtorno do Espectro Autista (TEA) é um transtorno de neurodesenvolvimento que afeta a interação e comunicação social, padrões de comportamentos, resistência a mudanças e déficit socioemocional do indivíduo. Além disso, as pessoas com TEA podem apresentar interesses restritos em coisas ou atividades como o hiperfoco em matemática, cubo mágico, letras, musculação ou natação, por exemplo.

Sendo assim, o diagnóstico deve levar em consideração diversos critérios e também uma análise do histórico familiar, escolar e social do paciente para uma investigação mais precisa [Rojas, Rivera e Nilo 2019]. Como consequência, para um

laudo ser conclusivo, é necessária a avaliação de uma equipe multiprofissional geralmente composta por neuropsiquiatras, pedagogos, pediatras e psicólogos (podendo envolver mais ou menos especialistas). Portanto, o diagnóstico costuma demorar um período considerável em decorrência da falta de profissionais qualificados, da frequência que a família alterna o médico que acompanha a criança e também das condições socioeconômicas e educacionais da população [Basso, Backes e Alves 2017].

Em outro aspecto, vê-se um avanço exponencial de inúmeras tecnologias e sistemas de informação que nos auxiliam em diferentes tarefas e são eficientes na otimização de tempo, sendo um forte exemplo as Inteligências Artificiais (IA). Elas são uma área interdisciplinar que combina conhecimentos de ciência da computação, matemática, biologia, neurociência e engenharia para construir máquinas capazes de simular a cognição humana.

Em sua essência, a IA busca replicar as funções cognitivas humanas em máquinas, permitindo que elas processem grandes quantidades de dados, aprendam com esses dados e tomem decisões com base em padrões identificados [Menezes 2023]. Esse campo oferece uma série de aplicações e oportunidades que estão moldando nosso mundo de maneiras surpreendentes.

Nesse sentido, pode-se afirmar que os algoritmos de aprendizado de máquina são a espinha dorsal da inteligência artificial, capacitando sistemas de computador a aprender e melhorar com a experiência, de maneira autônoma e sem a necessidade de muita codificação de programas de forma explícita. Eles permitem que as máquinas identifiquem padrões em dados, façam previsões, tomem decisões informadas e até mesmo compreendam e respondam à linguagem humana.

Portanto, o presente trabalho trata do desenvolvimento de uma aplicação utilizando técnicas da ciência de dados e aprendizagem de máquinas na análise dos dados de pessoas e na identificação de crianças atípicas em idade pré-escolar. Nesse sentido, a aplicação deverá fornecer ao profissional apontamentos sobre a presença, ou não, do transtorno do espectro autista, retornando certa probabilidade de acerto. Com isso, esta aplicação poderá ser utilizada por profissionais da educação como um fator de diagnóstico preliminar e triagem para encaminhamento a profissionais especialistas.

Para isso, foram realizadas buscas na internet sobre *datasets* prontos para estudo e implementação em linguagem de programação, em especial em sites como Kaggle[1], NCBI[2], repositório de ML da UCI[3], entre outros. Em seguida, após a preparação e normalização da base de dados, foram aplicados diferentes algoritmos de aprendizagem de máquinas para avaliação dos resultados obtidos sobre a classificação ou não da criança dentro de sua atipicidade.

Os algoritmos de classificação fornecem estimativas de avaliação, entre elas, a probabilidade de acerto conforme padrões encontrados no *dataset*. Como resultados, pretende-se: disponibilizar o *dataset* criado e tratado para pesquisas futuras na área, desenvolver o algoritmo para a predição e classificação dos dados de crianças e

[1] Disponível em: https://www.kaggle.com/
[2] Disponível em: https://www.ncbi.nlm.nih.gov/datasets/
[3] Disponível em: https://archive.ics.uci.edu/datasets

fornecer uma metodologia para a análise dos dados. Desta forma, os profissionais da área da educação poderão utilizá-lo como uma ferramenta adicional durante suas observações ao identificar comportamentos estereotipados e sem laudos de seus alunos.

Este artigo está organizado da seguinte forma. A seção 2 traz a fundamentação teórica sobre os algoritmos supervisionados. A seção 3 apresenta a metodologia para o desenvolvimento do trabalho. Os testes e os resultados obtidos são discutidos na seção 4. Por fim, as conclusões são apresentadas na seção 5.

2. Fundamentação teórica

Essa seção descreve o arcabouço teórico utilizado para elaboração deste artigo. O experimento utilizou os algoritmos supervisionados de machine learning com algoritmos de K-Nearest Neighbors (KNN), Regressão Linear, Naive Bayes, Árvore de Decisão e Support Vector Machines (SVM), que serão explicados no decorrer desta seção para analisar qual deles apresenta a maior precisão de acertos.

2.1. Algoritmos supervisionados

Os algoritmos não-supervisionados são aqueles que utilizam tabelas de dados (cada atributo é chamado de característica ou feature), sem qualquer tipo de informação (chamada de rótulo, resposta, classe ou atributo target) sobre o que este conjunto de dados representa. A ideia é que o algoritmo faça a análise dos valores de cada atributo e identifique possíveis rótulos. Diferentemente dos algoritmos não-supervisionados, os algoritmos supervisionados são uma classe de algoritmos de aprendizado de máquina que envolvem a aprendizagem a partir de um conjunto de dados rotulados. Em outras palavras, esses algoritmos são treinados usando uma tabela de dados que contém atributos de entradas (features) e também, em especial, uma coluna adicional com a saída (resposta, rótulo) correspondente aos dados de entrada, permitindo que o algoritmo faça previsões ou classificações com base nos padrões identificados entre os atributos de entrada e na saída obtida. A Figura 1 apresenta um exemplo de tabela de dados para uso em algoritmos de aprendizagem de máquina supervisionado e não-supervisionado.

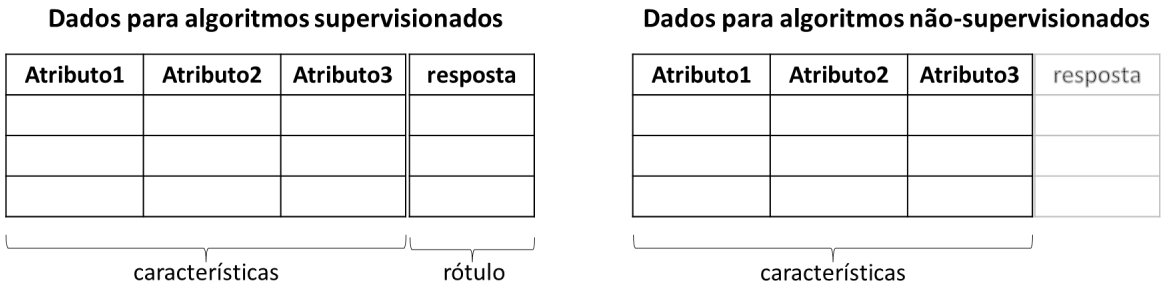

Figura 1. Datasets para algoritmos supervisionados (possui a coluna de rótulo) e não-supervisionado (sem a presença da coluna de rótulo)

Os algoritmos supervisionados são aplicados quando se tem um conjunto de dados de treinamento que é composto por pares de entrada e saída. As entradas representam características ou atributos dos dados, enquanto as saídas são os rótulos ou as respostas corretas que espera-se que o modelo preveja. O processo de treinamento envolve a exposição do modelo a esses exemplos rotulados, permitindo que ele

aprenda a relacionar as entradas às saídas correspondentes.

O objetivo central dos algoritmos supervisionados é criar um modelo capaz de fazer previsões precisas sobre novos dados que não fazem parte do conjunto de treinamento. Isso significa que o modelo deve aprender padrões e relações nos dados que podem ser generalizados para situações não vistas anteriormente. Por exemplo, em um problema de classificação, o modelo pode aprender a distinguir entre categorias de dados com base em características específicas. Em um problema de regressão, o modelo pode aprender a prever valores numéricos com base nas características de entrada.

Esse tipo de algoritmo tem aplicação nas mais diversas áreas: medicina e diagnósticos, classificação de documentos, setores financeiros, personalização de conteúdos, entre outros. Nas seções abaixo, explicam-se alguns desses algoritmos e serão apresentados os resultados obtidos com os mesmos.

2.2. K-Nearest Neighbors (KNN)

O algoritmo K-Nearest Neighbors (KNN), ou algoritmo dos vizinhos mais próximos, é uma técnica de aprendizado de máquina que se baseia no princípio de aprendizado por similaridade.

O KNN opera com base na ideia fundamental de que objetos semelhantes tendem a estar próximos uns dos outros no espaço de características. Portanto, para fazer previsões ou classificações, o algoritmo procura pelos K pontos de dados mais próximos (vizinhos) ao ponto de entrada que está sendo avaliado. A proximidade é geralmente medida por meio de métricas de distância, como a distância Euclidiana.

Uma vez que os K vizinhos mais próximos são identificados, o KNN atribui uma classe (no caso de classificação) ou um valor (no caso de regressão) com base na maioria dos rótulos dos vizinhos (no caso da classificação) ou na média de valores (no caso da regressão). Isso significa que o KNN toma uma decisão com base na votação da maioria ou na média dos K vizinhos mais próximos.

Um aspecto importante do algoritmo KNN é o parâmetro "K", que determina o número de vizinhos a serem considerados ao fazer uma previsão. A escolha apropriada de "K" é crucial, pois um valor muito pequeno pode tornar o modelo sensível a ruídos nos dados, enquanto um valor muito grande pode resultar em uma simplificação excessiva do modelo.

Em suma, a saída é determinada com base no parâmetro K e em situações de empate, várias técnicas podem ser aplicadas, sendo as mais comuns a escolha aleatória do vencedor ou a redução gradual do valor de K até que um vencedor absoluto seja encontrado (Frota et. al. 2020).

2.3. Regressão Linear

A Regressão Linear é uma técnica estatística essencial para modelar a relação entre uma variável dependente (rótulo) e uma ou mais variáveis independentes (características). O processo envolve a coleta de dados, escolha do modelo (simples ou múltiplo), treinamento do modelo para encontrar a melhor reta/plano que se ajusta aos dados, avaliação do ajuste e uso do modelo para fazer previsões. É uma ferramenta fundamental em análise estatística e aprendizado de máquina, aplicada em diversas

áreas para entender e verificar a correlação entre duas ou mais variáveis e testar o quanto se pode confiar nas estimativas encontradas [Chein 2019].

A Regressão Linear envolve uma equação fundamental $Y = \beta_0 + \beta_1 X + \epsilon$, em que Y é a variável dependente, X é a variável independente, β_0 e β_1 são os coeficientes, e ϵ é o erro residual. O treinamento do modelo busca ajustar β_0 e β_1 para encontrar a reta que melhor se ajusta aos dados, minimizando os erros residuais. Após o treinamento, o modelo pode ser usado para fazer previsões com base em novos valores de X. É uma técnica importante para a estatística aplicada em diversas áreas para compreender e prever relações entre variáveis de forma simples e interpretável.

2.4. Naive Bayes

O algoritmo de Naive Bayes é uma técnica de aprendizado de máquina, frequentemente utilizada em tarefas de classificação e categorização de dados. Sua base é o teorema de Bayes, que descreve como calcular probabilidades condicionais. No entanto, o termo "naive" (ingênuo) no nome do algoritmo refere-se à suposição simplificadora de independência condicional entre as características dos dados, mesmo que essa suposição nem sempre seja realista na prática.

Nesse sentido, o ponto central do algoritmo de Naive Bayes reside na utilização do teorema de Bayes para determinar a probabilidade condicional de uma classe (ou categoria) considerando um conjunto de características. A fórmula engloba a probabilidade da classe em questão, a probabilidade das características ocorrerem dado que a classe é a correta e a probabilidade das características ocorrerem de forma independente da classe.

Para treinar o algoritmo Naive Bayes, é necessário um conjunto de dados rotulados, onde as classes são conhecidas. Durante o treinamento, o modelo aprende as probabilidades das características para cada classe. Na fase de classificação, o modelo calcula a probabilidade de pertencimento a cada classe e atribui a classe com a maior probabilidade como a previsão.

2.5. Árvore de Decisão

As árvores de decisão são amplamente utilizadas para tarefas de classificação e regressão. Elas oferecem uma abordagem interpretativa e eficaz para tomar decisões com base em regras lógicas derivadas de dados. De acordo com o trabalho de Quinlan [1993], "a árvore de decisão é uma representação gráfica de uma estratégia de decisão e, por isso, é uma técnica de aprendizado muito natural e intuitiva."

A estrutura de uma árvore de decisão consiste em nós e arestas, em que cada nó representa um teste ou decisão em relação a uma característica dos dados, e cada aresta representa um resultado possível desse teste [Frota et. al. 2020]. No topo da árvore, encontra-se o nó raiz, que reflete a característica mais crucial para a classificação ou previsão. Os nós internos representam testes, enquanto os nós folha representam as classes ou valores previstos.

O processo de treinamento de uma árvore de decisão envolve a seleção das características mais relevantes para tomar decisões em cada nó. O algoritmo procura dividir o conjunto de dados de treinamento de forma a maximizar a homogeneidade das classes dentro de cada subdivisão.

2.6. Support Vector Machines (SVM)

O SVM visa encontrar um limite de decisão (hiperplano) que melhor separe os dados em diferentes classes. Este hiperplano maximiza a margem, que é a distância entre o hiperplano e os pontos de dados mais próximos de cada classe, conhecidos como vetores de suporte. A ideia principal é encontrar o hiperplano que maximize essa margem, tornando o modelo robusto e melhor generalizado para novos dados.

Esse tipo de algoritmo pode realizar classificação linear, onde os dados são separados por um hiperplano. Se os dados não puderem ser separados linearmente, uma técnica chamada "truque do kernel" é aplicada, mapeando os dados em um espaço de dimensão superior onde a separação linear se torna possível.

No SVM, as entradas são representadas como pontos em um hiperplano em R^1, onde 'l' denota a quantidade de atributos. O algoritmo constrói um plano separador entre esses pontos, dividindo-os em duas classes. A separação é possível quando o problema é linearmente separável, ou seja, quando é viável separar as classes usando um hiperplano de dimensão 'l - 1'. Essa abordagem pode ser estendida para classificação de múltiplas classes, criando-se classificadores binários para cada par de classes. A interpretação geométrica do SVM envolve a busca pelo hiperplano equidistante das duas classes, visando encontrar a separação ideal em uma superfície [Frota et. al. 2020].

3. Metodologia

Esta seção trata da metodologia utilizada para desenvolvimento da estrutura para identificação de crianças autistas.

Inicialmente, foi realizado um levantamento bibliográfico a respeito do Transtorno do Espectro Autista para identificar suas características, critérios de avaliação e suas variáveis, os resultados destas avaliações e principais desafios enfrentados no processo de diagnóstico. Os critérios de inclusão aplicados foram: artigos publicados nos últimos 7 anos, entre 2017 e 2023, disponíveis no Google Academy e publicados na língua portuguesa, inglesa ou espanhola. Os critérios de exclusão são: artigos fora do período ou da língua requisitada, capítulos de livros que não atendessem ao tema proposto.

Posteriormente, foi realizada uma busca de bases de dados de domínio público sobre o diagnóstico do TEA em crianças. Foram considerados Banco de Dados dos últimos 7 anos, em língua portuguesa e inglesa e que apresentam dados a respeito da diagnose em pessoas menores de 10 anos. O critério de definição foi a base que apresentava uma maior quantidade de dados e que apresentasse embasamento em algum protocolo profissional relevante.

3.1. Banco de dados

O dataset escolhido, denominado Autism screening data for toddlers e disponível no Kaggle[4], contém dados de 1024 pessoas e é 100% baseado no Q-CHAT 10, desenvolvido pela Universidade de Cambridge com o propósito de ser um teste rápido e eficiente para indicar sintomas alertas de TEA. A Figura 2 apresenta a tabela com as perguntas do Q-CHAT 10.

[4] Disponível em https://www.kaggle.com/datasets/fabdelja/autism-screening-for-toddlers

Para cada item, escolha a resposta que melhor se aplica ao seu filho:

	A	B	C	D	E
1- A sua criança olha para si quando chama pelo nome dela?	Sempre	Habitualmente	Às vezes	Raramente	Nunca
2- Quão fácil é para si conseguir contacto ocular com a sua criança?	Muito fácil	Bastante fácil	Bastante difícil	Muito difícil	Impossível
3- A sua criança aponta para indicar que quer alguma coisa? (ex: um brinquedo que está fora do alcance)	Muitas vezes por dia	Algumas vezes por dia	Algumas vezes por semana	Menos de uma vez por semana	Nunca
4- A sua criança aponta para partilhar um interesse consigo? (ex: apontar para um cenário interessante)	Muitas vezes por dia	Algumas vezes por dia	Algumas vezes por semana	Menos de uma vez por semana	Nunca
5- A sua criança "faz de conta"? (ex: ao cuidar de bonecas, falar num telefone de brincar)	Muitas vezes por dia	Algumas vezes por dia	Algumas vezes por semana	Menos de uma vez por semana	Nunca
6- A sua criança segue o seu olhar?	Muitas vezes por dia	Algumas vezes por dia	Algumas vezes por semana	Menos de uma vez por semana	Nunca
7- Se você ou alguém da sua família estiver visivelmente aborrecido, a sua criança mostra sinais de querer confortá-lo? (ex: acariciando o cabelo, abraçando)	Sempre	Habitualmente	Às vezes	Raramente	Nunca
8- Descreveria as primeiras palavras da sua criança como:	Muito comuns	Bastante comuns	Ligeiramente incomuns	Muito incomuns	A minha criança não fala
9- A sua criança usa gestos simples? (ex: acenar adeus)	Muitas vezes por dia	Algumas vezes por dia	Algumas vezes por semana	Menos de uma vez por semana	Nunca
10-A sua criança olha fixamente para nada sem razão aparente?	Muitas vezes por dia	Algumas vezes por dia	Algumas vezes por semana	Menos de uma vez por semana	Nunca

Cotação: Para as perguntas 1-9: se escolher uma resposta nas colunas C, D ou E, cote 1 ponto por pergunta. Para a pergunta 10: se escolher uma resposta nas colunas A, B ou C, cote 1 ponto. Some os pontos para a totalidade das 10 perguntas. Se o seu filho obtiver mais do que 3 em 10, o profissional de saúde pode considerar o encaminhamento da sua criança para uma avaliação multidisciplinar.

Referência chave: Allison C, Auyeung B, and Baron-Cohen S, (2012) Journal of the American Academy of Child and Adolescent Psychiatry 51(2):202-12.

Figura 2. Tabela com as perguntas do Q-CHAT 10

A Figura 3 apresenta um extrato da base de dados utilizada, contendo os atributos do dataset e algumas instâncias (linhas da tabela representando cada caso de estudo). Algumas adaptações nestes dados devem ser realizadas, tais como exclusão de colunas (atributos) que não sejam de interesse do estudo sendo realizado (apresentado na Figura 4) e normalização de dados (apresentado na Figura 5). Para otimização do tempo e garantir facilidade na comunicação dos envolvidos no trabalho, todos os códigos foram realizados em um notebook virtual através do software Google Colab.

```
import pandas as pd

df = df = pd.read_csv('Toddler Autism dataset July 2018.csv')

df
```

	Case_No	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age_Mons	Qchat-10-Score	Sex	Ethnicity	Jaundice	Family_mem_with_ASD	Who completed the test	Class/ASD Traits
0	1	0	0	0	0	0	0	1	1	0	1	28	3	f	middle eastern	yes	no	family member	No
1	2	1	1	0	0	0	1	1	0	0	0	36	4	m	White European	yes	no	family member	Yes
2	3	1	0	0	0	0	0	1	1	0	1	36	4	m	middle eastern	yes	no	family member	Yes
3	4	1	1	1	1	1	1	1	1	1	1	24	10	m	Hispanic	no	no	family member	Yes
4	5	1	1	0	1	1	1	1	1	1	1	20	9	f	White European	no	yes	family member	Yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1049	1050	0	0	0	0	0	0	0	0	0	1	24	1	f	White European	no	yes	family member	No
1050	1051	0	0	1	1	1	0	1	0	1	0	12	5	m	black	yes	no	family member	Yes
1051	1052	1	0	1	1	1	1	1	1	1	1	18	9	m	middle eastern	yes	no	family member	Yes
1052	1053	1	0	0	0	0	0	0	1	0	1	19	3	m	White European	no	yes	family member	No
1053	1054	1	1	0	0	1	1	0	1	1	0	24	6	m	asian	yes	yes	family member	Yes

1054 rows × 19 columns

Figura 3 - Dataset aberto para visualização (detalhamento no item 6)

```
#Colunas "Who complete the test", "Ethnicity", "Case_No" não são interessantes
df2 = df.drop(["Who completed the test", "Ethnicity", "Case_No"], axis = 1)

#Coluna Qchat-10-Score é relevante?
df2 = df2.drop(["Qchat-10-Score"], axis=1)
```

Figura 4. Removendo colunas que não eram interessantes

```
#Bibliolteca utilizada para normalização
from sklearn.preprocessing import MinMaxScaler
from pickle import dump

normalizador = MinMaxScaler()
normalizador.fit(df2[["Age_Mons"]])
dump(normalizador, open("normalizador_idade.pkl", "wb"))
df2["Age_Mons"] = normalizador.transform(df[["Age_Mons"]])
```

Figura 5. Normalizando dados entre [0,1] baseando-se nos valores da coluna “Age_Mons”

Adicionalmente, alguns tratamentos de dados devem ser realizados. Por exemplo, alguns algoritmos podem não aceitar valores categóricos (Strings) na sua execução. Para contornar um possível problema com a base de dados, os valores do atributo Sex (“f”, “m”), Jaudice (“yes”, “no”) e Family_mem_with_ASD (“yes”, “no”) podem ser convertidos para valores binários 0 e 1. A Figura 6 apresenta esta conversão.

```
df2['Sex'] = df2['Sex'].apply({'m':1, 'f':0}.get)
df2['Jaundice'] = df2['Jaundice'].apply({'yes':1, 'no':0}.get)
df2['Family_mem_with_ASD'] = df2['Family_mem_with_ASD'].apply({'yes':1, 'no':0}.get)
```

Figura 6. Conversão de valores categóricos para valores numéricos

Para a realização de testes, pode-se dividir os dados da base de dados em dois conjuntos: treinamento e testes. A base de treinamento será utilizada pelos algoritmos de aprendizado de máquinas para identificar padrões e realizar classificações. A base de teste pode ser utilizada para a realização de testes e avaliação dos resultados obtidos. A Figura 6 demonstra como dividir a base de dados em treinamento (80% dos dados) e teste (20% dos dados).

```
#Características
X = df2.drop("Class/ASD Traits ", axis=1)
y = df2["Class/ASD Traits "]

#Divisão de trienamento e teste
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42, shuffle=True)
```

Figura 6. Divisão dos dados para treinamento do algoritmo e testes/avaliação

Em um segundo momento, pode ser utilizado framework de processamento e análise de dados, juntamente a algoritmos de machine learning, para ensinar a máquina a realizar uma avaliação dos dados recebidos através dos usuários (do dataset) acerca de informações da criança. Todo o trabalho foi desenvolvido em Python, utilizando a biblioteca Pandas, Scikit-Learn e, para apresentação dos dados, a biblioteca MatplotLib, todas muito conhecidas e utilizadas para limpeza e análise de dados.

4. Resultados e testes

Nesta seção, os resultados de testes feitos em cada um dos algoritmos serão apresentados. Será apresentado também a importância das variáveis em cada modelo e como isso interfere ou não na precisão dos modelos. O trabalho avalia ao final qual a melhor classificação e análise dos dados contidos na base mencionada.

4.1. Matriz de confusão e acurácia

Em nossa estrutura, foi escolhido avaliar o desempenho dos modelos usando apenas a acurácia e a matriz de confusão. Essas métricas são de suma importância para entender a precisão e eficácia dos algoritmos desenvolvidos. Essas métricas combinadas nos ajudam a ter uma compreensão completa do quão bem nossos modelos estão realizando suas tarefas e qual apresenta a maior eficácia.

A matriz de confusão é uma ferramenta essencial na avaliação de desempenho de modelos de classificação em aprendizado de máquina e análise estatística. Ela permite analisar de forma detalhada como um classificador se comporta ao fazer previsões sobre um conjunto de dados.

Essa matriz divide os resultados da classificação em quatro categorias distintas, conforme apresentado na Figura 7:

1. **Verdadeiro Positivo (VP):** Representa os casos em que o modelo classificou corretamente uma instância como positiva.

2. **Falso Positivo (FP):** Esses são os casos em que o modelo erroneamente previu uma instância como positiva quando, na realidade, ela pertence à classe negativa.

3. **Verdadeiro Negativo (VN):** Casos em que o modelo classificou corretamente uma instância como negativa.

4. **Falso Negativo (FN):** Representa os casos em que o modelo erroneamente previu uma instância como negativa quando, na verdade, ela pertence à classe positiva.

		Valor Previsto	
		Positivo	Negativo
Valor Real	Positivo	Verdadeiros Positivos	Falsos Negativos
	Negativo	Falsos Positivos	Verdadeiros Negativos

Figura 7 - Tabela exemplificando os resultados da matriz de confusão

A acurácia é uma métrica de avaliação de desempenho comumente utilizada em problemas de classificação em aprendizado de máquina e análise estatística. Ela mede a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões realizadas. Em termos simples, a acurácia responde à pergunta: "Quantas das previsões feitas pelo modelo estão corretas?". Quanto maior a acurácia, melhor o desempenho do modelo em fazer previsões corretamente. Para realizar o cálculo, é utilizada uma fórmula onde: $Acurácia = \frac{Previsões\ Corretas}{Total\ de\ Previsões}$.

4.2. Avaliação de desempenho

Os gráficos desta seção apresentam a matriz de confusão de cada um dos modelos em questão, além de mostrarem a acurácia e precisão de cada um. Para a realização das classificações, os seguintes classificadores foram utilizados, conforme apresentado na Tabela 1:

Tabela 1. Bibliotecas utilizadas para execução da aprendizagem de máquina

kNN	Biblioteca	`from sklearn.neighbors import KNeighborsClassifier`
	Objeto classificador	`classificador= KNeighborsClassifier(n_neighbors=5)`
Árvore de Decisão	Biblioteca	`from sklearn.tree import DecisionTreeClassifier`
	Objeto classificador	`classificador=DecisionTreeClassifier(random_state=0)`
Naive Bayes	Biblioteca	`from sklearn.naive_bayes import GaussianNB`
	Objeto classificador	`classificador= GaussianNB()`
Regressão Linear (Logística para classificação)	Biblioteca	`from sklearn.linear_model import LogisticRegression`
	Objeto classificador	`classificador= LogisticRegression(random_state=0, solver='liblinear')`
Support Vector Machine	Biblioteca	`from sklearn.svm import SVC`
	Objeto classificador	`classificador= SVC(kernel='linear')`

Para as execuções de treinamento e predição, assim como para calcular a acurácia e a matriz confusão, as seguintes instruções apresentadas na Figura 8 foram executadas:

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

classificador.fit(X_train, y_train)      #treinar o algoritmo
y_pred = classificador.predict(X_test)   #executar as predições

acuracia = accuracy_score(y_test, y_pred)
print("Acurácia:", acuracia)

cm = confusion_matrix(y_test, y_pred, labels=classificador.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
                              display_labels=classificador.classes_)
disp.plot()
plt.show()
```

Figura 8 - Execução do algoritmo de aprendizagem de máquinas

As acurácias obtidas foram:

- kNN: 0.9383886255924171
- Árvore de decisão: 0.9052132701421801
- Naive Bayes: 0.95260663507109
- Regressão Logística: 1.0
- SVM: 1.0

As matrizes de confusão obtidas para cada algoritmo são apresentadas na Figura 9.

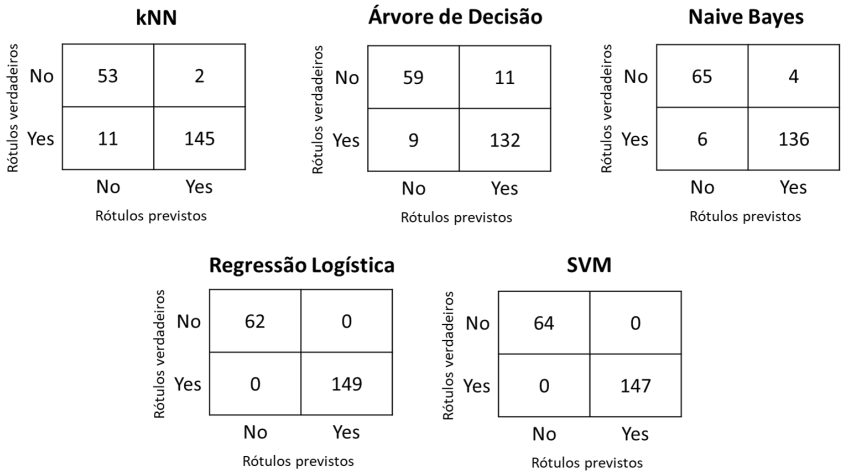

Figura 9 - Matrizes confusão para os algoritmos de classificação

5. Conclusão

Este trabalho trata do desenvolvimento de uma aplicação que visa auxiliar profissionais da educação a realizar uma pré-análise de crianças potencialmente atípicas, mediante percepção de comportamentos estereotipados durante as atividades do dia-a-dia na escola. Por meio da aplicação, os profissionais poderão sugerir às famílias destas crianças pela procura de profissionais especializados para que novos diagnósticos, mais precisos, possam ser elaborados. Desta forma, a criança pode receber uma atenção diferenciada e melhor acompanhamento durante as atividades escolares.

Conforme experimento realizado utilizando a linguagem Python e os algoritmos de classificação disponíveis por meio da biblioteca sklearn, observou-se uma certa probabilidade de acerto considerável, todas acima de 90% de precisão. Vale observar que o algoritmo foi treinado com dados obtidos em um *dataset* disponível na Internet e que estes dados poderão ser validados com o decorrer do uso da aplicação pelos profissionais da educação.

Além disso, foi implementado uma aplicação web com um formulário para que os profissionais possam informar os dados individuais de uma criança e obterem a probabilidade desta criança pertencer ou não ao espectro autista. Todas as perguntas foram retiradas do dataset normalizado, que possui como base o Q-CHAT 10.

Como trabalhos futuros, sugere-se a implementação de uma aplicação mobile e a adaptação da web já existente, de forma que os profissionais possam adicionar novas instâncias para serem usadas com o treinamento do algoritmo.

6. Legenda

A Figura 03 apresenta as seguintes colunas (features):

- **A1**: A criança olha para você quando chama o nome dela?
- **A2**: O quão fácil é para você ter contato visual com a criança?
- **A3**: A criança aponta para indicar que ela deseja algo?
- **A4**: A criança aponta para compartilhar algum interesse dela com você?
- **A5**: A criança finge? ex: cuidar de bonecas, fala em um telefone de brinquedo?
- **A6**: A criança segue para onde você está olhando?

- **A7**: Se você ou alguém da família está visivelmente chateado, a criança mostra sinais de desânimo para consolá-lo? por exemplo. acariciando os cabelos, abraçando-os
- **A8**: Você descreveria a primeira palavra da criança como:
- **A9**: Ela usa gestos simples (por exemplo, acenar adeus)?
- **A10**: Ela fica olhando para o nada sem propósito aparente?
- **Age_Mons:** é a idade em meses da criança
- **Q-chat-10-Score:** é a soma de valores 1 das questões A1 a A10
- **Ethnicity:** é o local que a criança mora
- **Jaundice:** diz se a criança tem icterícia ou não
- **Family_mem_with_ASD:** se algum membro da família tem ou não autismo
- **Who completed the test:** quem preencheu o formulário com a criança
- **Class/ASD Traits:** diz se a criança é ou não autista

Agradecimentos

Este trabalho tem o apoio da Fundação Araucária, grupo de pesquisa Manna Team e demais instituições envolvidas.

Referências

APA. American Psychiatric Association et. al. (2014). *DSM-5: Manual diagnóstico e estatístico de transtornos mentais*. Artmed Editora. https://www.institutopebioetica.com.br/documentos/manual-diagnostico-e-estatistico-de-transtornos-mentais-dsm-5.pdf

Basso Z., R., Backes, B., Alves B., C. (2017). *Diagnóstico do autismo: relação entre fatores contextuais, familiares e da criança. Psicologia: Teoria e Prática [em linha]. 2017*. https://www.redalyc.org/articulo.oa?id=193851916009

Chein, F (2019). I*ntrodução aos modelos de regressão linear.* Metodologias Coleção, https://repositorio.enap.gov.br/bitstream/1/4788/1/Livro_Regress%C3%A3o%20Linear.pdf

Frota, M. et. al. (2020). *Análise de Características a partir de Algoritmos de Aprendizagem de Máquina para Auxílio ao Diagnóstico do Transtorno do Espectro Autista*. Revista de Sistemas e Computação-RSC, Vol. 10, p. 94-103. jhttps://revistas.unifacs.br/index.php/rsc/article/view/6476

Quinlan, J. R. (1993). *C4.5 Programas para aprendizado de máquina*. Morgan Kaufmann Publishers, Inc., 1993. *Mach Learn* 16 , 235–240 (1994). https://doi.org/10.1007/BF00993309

Rojas, V.; Rivera, A.; Nilo, N. (2019). *Actualización en diagnóstico e intervención temprana del Trastorno del Espectro Autista*. Rev. chil. pediatr., Santiago, v. 90, n. 5, p. 478-484

Menezes, Marcos A. (2023). *A Inteligência Artificial versus a Inteligência Humana*. Saberes Humanos, Vol. 13, n. 22, p. 220-239.