

ALCATEA – APLICATIVO DE AUTOMATIZAÇÃO DE APLICAÇÃO DE TESTES PSICOLÓGICOS DO TRANSTORNO DO ESPECTRO AUTISTA

Gustavo da Gama Souza¹ (IC), Alan Bendasoli Pavan (PQ)¹

¹Universidade Federal de Itajubá

²Grupo de Estudos e Pesquisa em Machine Learning - GREMLING

Palavras-chave: Autismo. TEA. Testes Psicológicos.

Introdução

Neste trabalho, foi desenvolvida uma aplicação para dispositivos móveis, a qual o principal enfoque é automatizar testes psicológicos do transtorno do espectro autista. Testes não automatizados são realizados com anotações em fichários ou em modelos impressos predefinidos. Este método de avaliação, consiste em uma soma de pontos conforme o valor de cada resposta obtida. Tal mecanismo, além de exigir uma excessiva quantidade de materiais, pode gerar imprecisões e falhas cometidas na soma e manipulação dos dados coletados, sem o auxílio de tecnologias. Ao longo do trabalho, várias ferramentas computacionais são aplicadas no intuito de promover eficiência aos aplicadores e facilitar o manuseio de dados e armazenamento. Posteriormente, com os dados coletados, é aplicado camadas de *Machine Learning* para impulsionar o estudo dos diagnósticos.

A importância na aplicação de Data Science no contexto de tratamento do TEA é de extrema relevância. De acordo com estudos realizados por profissionais do Jornal de Autismo e Transtornos do Desenvolvimento, em New Haven, foi verificado que o diagnóstico do TEA pode corresponder a conflitos gerados por outros transtornos mentais. Uma amostra composta por 65 crianças, com idades entre 18 meses e 11 anos, foi realizada e encontrada concordâncias entre outros transtornos, com a aplicação de testes próprios para o TEA. Foi mostrado que o teste ABC (Autism Behavior Checklist) não distingue indivíduos com transtornos autistas de outros casos de transtornos do desenvolvimento, e que o número de falsos negativos é alto (46%). Inúmeras clínicas e setores de tratamento de transtornos psicológicos utilizam testes traduzidos, de modo com que a aplicação do Aprendizado de Máquina também pode ser útil no enriquecimento de dados locais e aumento da precisão referente aos diagnósticos coletados.

Um protótipo do aplicativo desenvolvido foi disponibilizado para as Associações de Pais e Amigos dos Excepcionais (APAIE) de Itajubá, sendo atualmente utilizado para a assistência dos aplicadores e também

para a coleta de dados.

Metodologia

O desenvolvimento do projeto foi dividido em etapas, sendo elas:

1. Coleta de referências das escalas de testes utilizadas na aplicação;
2. Montagem de códigos e compilação direta entre o *backend* e *frontend*;
3. Criação e configuração do banco de dados para armazenamento dos resultados obtidos;
4. Testes de precisão com aplicações manuais;
5. Tratamento dos dados em *Machine Learning*;
6. Distribuição do protótipo;

As referências utilizadas foram baseadas em artigos gratuitamente disponíveis referente aos três testes utilizados: M-CHAT (Modified Checklist for Autism in Toddlers), ABC (Autism Behavior Checklist) e a Escala ATA (Avaliação de Traços Autísticos). Para montagem dos códigos, foi utilizado o software Android Studio inteiramente na linguagem de programação Java. Outras linguagens de programação como: Kotlin e HTML também foram utilizadas, porém, com menor frequência, sendo principalmente manipuladas no ambiente de *frontend*. O software VSCode foi utilizado nativamente com a linguagem de programação Python para o tratamento dos dados coletados pelo aplicativo. Para armazenamento dos resultados, foi criado gratuitamente um banco de dados pela plataforma Firebase, disponibilizada gratuitamente para *Mobile Developers*. Para fins de organização, foram semanalmente feitas reuniões com o professor orientador de forma híbrida (presencial e virtualmente no ambiente Google Meet).

Resultados e discussão

Neste trabalho comparamos os resultados finais gerados pelo aplicativo como com os resultados já existentes, diferenciando a sua exatidão. Para a obtenção dos diagnósticos, foi utilizado a função Canvas, cujo objetivo

é gerar arquivos contendo o resultado dos algoritmos:

```
>>> PdfDocument.Page newpage =  
documentPDF.startPage(pagetails);  
>>> Canvas canvas = newpage.getCanvas();
```

Algumas informações não são exibidas no arquivo gerado ao fim da aplicação. Com o intuito de indicar específicos alvos de desenvolvimento para cada indivíduo testado, gráficos são mostrados ao fim de cada teste na sua última *activity*. Cada teste possui determinadas características e categorias de divisão e, dependendo da quantidade de categorias, gráficos são criados mostrando percentagens e dando informações visuais. Para o *plot* de gráficos, é utilizado um *implement* feito pelo desenvolvedor Phil Jay:

```
>>> implementation  
'com.github.PhilJay:MPAndroidChart:v3.0.3'
```

O método adicionado permite com que sejam utilizados gráficos sem limites de componentes, de fácil manuseio nos seus eixos e com animações visuais.

Para fazer o armazenamento das informações é necessário que seja possível criar um banco de dados seguro e facilmente acessado pelos desenvolvedores. Uma plataforma desenvolvida pelo Google e de acesso gratuito, Firebase, voltada para criação de aplicativos móveis e da web, foi utilizada para realizar o processo de armazenamento de dados e autenticação de usuários. É necessário primeiramente integrar ao Android Studio a ferramenta, utilizando o passo a passo disponibilizado no próprio site da plataforma. Foi possível trabalhar com duas ferramentas da plataforma: autenticação e base de dados. A função *firebase-auth* foi utilizada com um processo trivial de cadastro, recebimento de confirmação por e-mail e autenticação direta.

Para o tratamento dos dados aplicados ao aprendizado de máquina, uma base de dados foi feita aleatoriamente para ser utilizada. Primeiramente foi importado a biblioteca *pandas*, para compilar a tabela de dados e tratar o algoritmo:

```
<<< import pandas as pd  
<<< arquivo =  
pd.read_csv('C:/ICDados/testes_dataset.csv')  
<<< arquivo.head();
```

Foi utilizado no tratamento de dados o sistema binário para classificar o indivíduo diagnosticado com possível presença de autismo (1) ou baixa possibilidade de presença de autismo (0):

```
<<< arquivo['diagnostic'] =  
arquivo['diagnostic'].replace('autism', 1)  
<<< arquivo['diagnostic'] =  
arquivo['diagnostic'].replace('noautism', 0)
```

A base de dados utilizada para a aplicação do aprendizado de máquina foi obtida a partir da escala Modified Checklist for Autism in Toddlers (M-CHAT):

```
<<< 0-"Diagnóstico: Positivo para TEA" ← 1  
<<< 1-"Nome: Paciente 23"  
<<< 2-"Data de nascimento: 29/10/2019"  
<<< 3-"Cidade: Itajubá"  
<<< 4-"(2, 2, 2, 4, 4, 4, 1, 4, 4, 1, 4, 2, 2, 4, 1,  
4, 3, 4, 4, 2, 4, 1, 1, 1, 4, 3, 2, 4, 1, 1, 3, 2, 1,  
4, 1, 1, 4, 4, 4, 1, 4, 3, 1, 1, 1, 3, 3, 4, 1, 4, 3, 4,  
4, 4, 2, 4, 1, 2)
```

Os algoritmos presentes na linha 5 denotam a intensidade de resposta para cada questão. É possível notar que o Paciente 23 foi diagnosticado como positivo para TEA, então para a base de dados ele contará como 1 no *.head*. Muitos outros dados gerados aleatoriamente, foram adicionados na tabela. Assim, é necessário criar uma tabela para leitura, onde na coluna *y* estará o diagnóstico de cada teste e nas colunas adicionais a pontuação em cada questão.

Foi utilizado 40% dos dados para teste e 60% para treino da máquina. Para classificar então os testes, foi utilizado a função *ExtraTreesClassifier* capaz de criar árvores de decisão:

```
<<< from sklearn.ensemble import  
ExtraTreesClassifier  
<<< model = ExtraTreesClassifier()  
<<< model.fit(x_treinarmaquina,  
y_treinarmaquina)  
<<< result = modelo.score(x_testarmaquina,  
y_testarmaquina)
```

Para uma maior precisão nas classificações, é necessária uma grande quantidade de dados de treino. A precisão dos resultados obtidos pela máquina, não fazem uma referência concreta aos dados obtidos por uma aplicação manual.

Conclusões

O desenvolvimento e a aplicação de ferramentas computacionais na área de saúde e tratamento psicológico se mostram cada vez mais necessários. O trabalho feito assume que é fundamental concentrar-se em facilitar a aplicação de testes e utilizar computadores para prever

diagnósticos. A utilização de dispositivos móveis mostra ser uma poderosa ferramenta e indica uma forte possibilidade de aplicação na área de Data Science. As ferramentas tiveram uma funcionalidade completa e o uso para aplicação dos testes se torna seguro ao fim do desenvolvimento. A aplicabilidade de *Machine Learning* conceitualmente mostra uma forte eficiência e que também é necessário a aquisição de uma quantidade abundante de dados para uma boa precisão.

Agradecimento

À minha família e amigos, agradeço imensamente o incentivo e companheirismo durante todo período de pesquisa. Ao meu orientador Alan Bendasoli Pavan, pela paciência e apoio constante, sendo muito importante para meu desenvolvimento como aluno e pesquisador. Ao NIT-UNIFEI pelo apoio e assistência técnica no registro do aplicativo no Instituto Nacional da Propriedade Industrial (INPI). Ao CNPQ, órgão responsável pelo financiamento do projeto e auxílio frente à pesquisa.

Referências

- [1] Rellini, E., Tortolani, D., Trillo, S. et al. Childhood Autism Rating Scale (CARS) and Autism Behavior Checklist (ABC) Correspondence and Conflicts with DSM-IV Criteria in Diagnosis of Autism. *J Autism Dev Disord* 34, 703–708 (2004). <https://doi.org/10.1007/s10803-004-5290-2>.
- [2] Naoko I., Tomonori K., Eiko I., Reliability and validity of the Japanese version of the Modified Checklist for autism in toddlers (M-CHAT), *Research in Autism Spectrum Disorders*, Pages 330-336, ISSN 1750-9467 (2011). <https://doi.org/10.1016/j.rasd.2010.04.016>.
- [3] Júnior A., Baptista F., Matteo J., Scale for the assessment of Autistic Behaviour: (ATA): *Med. rehabil*, 41-44 (2008).
- [4] Baştanlar, Y., Özuysal, M. (2014). Introduction to Machine Learning. In: Yousef, M., Allmer, J. (eds) *miRNomics: MicroRNA Biology and Computational Analysis. Methods in Molecular Biology*, vol 1107. Humana Press, Totowa, NJ. https://doi.org/10.1007/978-1-62703-748-8_7