**Navegando o cenário da Saúde Mental na Indústria Tech**

Uma análise baseada em aprendizado de máquina

Augusto Scardua Oliveira, Bruno Santiago de Oliveira, Carolina Morais Nigri, Fábio Freire Kochem, Pedro Heinrich Sales Pena, Pedro Miranda Rodrigues

Ciência da Computação, PUC Minas - Belo Horizonte, MG, Brasil

[augusto.oliveira.1388188@sga.pucminas.br](mailto:augusto.oliveira.1388188@sga.pucminas.br), [bruno.santi.oli@email.com](mailto:bruno.santi.oli@email.com), c[arolina.nigri@sga.pucminas.br](mailto:carolina.nigri@sga.pucminas.br), [fabio.freire@sga.pucminas.br](mailto:fabio.freire@sga.pucminas.br), [pedroheinri@gmail.com](mailto:pedroheinri@gmail.com), [pedro.rodrigues.1373336@sga.pucminas.br](mailto:pedro.rodrigues.1373336@sga.pucminas.br)

**RESUMO**

Este artigo traz uma análise dos resultados de uma pesquisa realizada pela *Open Sourcing Mental Health* (OSMI), em 2014, sobre a saúde mental dos trabalhadores da indústria de tecnologia. Utilizando técnicas de aprendizado de máquina, este estudo busca identificar padrões e tendências nos dados coletados pela pesquisa descrita. Após realizar um pré-processamento da base de dados, removendo e alterando atributos conforme necessário, foram aplicados os algoritmos de *Random Forest*, uma variação do *XGBoost* e a Árvore de Decisão para prever se os funcionários procurariam ou não tratamento. Os resultados dos modelos criados e suas respectivas análises são apresentados neste documento. Com isso, busca-se, como objetivo principal, contribuir para a promoção de um ambiente de trabalho mais saudável na indústria tech, visando auxiliar na formulação de estratégias que tragam melhorias para a saúde mental desses trabalhadores.

***KEYWORDS***

Saúde Mental no Trabalho, Indústria da Tecnologia, Aprendizado de Máquina, Classificador Random Forest, Modelagem Preditiva, Cultura no Ambiente de Trabalho Tech, Análise de Dados, Pesquisa com empregados, Suporte de Decisão, Algoritmo de Classificação, Iniciativas de Apoio no Local de Trabalho.

**ACM Reference format:**

Augusto S. Oliveira, Bruno S. de Oliveira, Carolina M. Nigri, Fábio F. Kochem, Pedro H. Sales Pena, Pedro M. Rodrigues. 2023. Navegando o cenário da Saúde Mental na Indústria Tech: Uma análise baseada em aprendizado de máquina. PUC Minas, Belo Horizonte, Brasil*.*

**1. Introdução**

A *Open Sourcing Mental Health* (OSMI) é uma organização sem fins lucrativos dedicada a aumentar a conscientização, educar e fornecer recursos para apoiar o bem-estar mental nas comunidades de tecnologia e código aberto.[[1]](https://osmihelp.org/about/about-osmi.html) Em busca dessa missão, a OSMI conduziu uma pesquisa abrangente em 2014, com o objetivo de avaliar as atitudes em relação à saúde mental e a prevalência de transtornos mentais no ambiente de trabalho de tecnologia.[[2]](https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey)

De maneira notável, em toda a Europa, tem havido um preocupante aumento nas faltas relacionadas à doença e aposentadorias precoces atribuídas a problemas de saúde mental.[[3]](https://policycommons.net/artifacts/1943653/eu-compass-for-action-on-mental-health-and-well-being/2695422/) Essa tendência preocupante não afeta apenas o bem-estar das pessoas, mas também representa um crescente fardo para a economia devido aos custos crescentes associados à saúde e à produtividade perdida.

Neste artigo, aprofundaremos as informações obtidas na pesquisa da OSMI em 2014 e aproveitaremos o poder da aprendizagem de máquina para analisar esses dados. Usando modelos de aprendizado de máquina, nosso objetivo é descobrir padrões e tendências valiosas na interseção entre saúde mental e a indústria de tecnologia. Em última análise, nosso objetivo é contribuir para uma abordagem mais informada, baseada em dados e voltada para o futuro para promover uma força de trabalho de tecnologia mais saudável e resiliente.

**2. Descrição da base de dados**

A saúde mental na indústria de tecnologia é um tema de crescente importância e atenção. À medida que a demanda por inovação e produtividade nesse setor aumenta, também aumenta a pressão sobre os profissionais de tecnologia. Longas horas de trabalho, prazos apertados e a natureza competitiva do campo podem levar a desafios significativos para o bem-estar mental. Nesse contexto, se torna fundamental promover pesquisas e estudos que possam contribuir para o crescimento da área, sendo essa a abordagem da base de dados utilizada neste artigo.

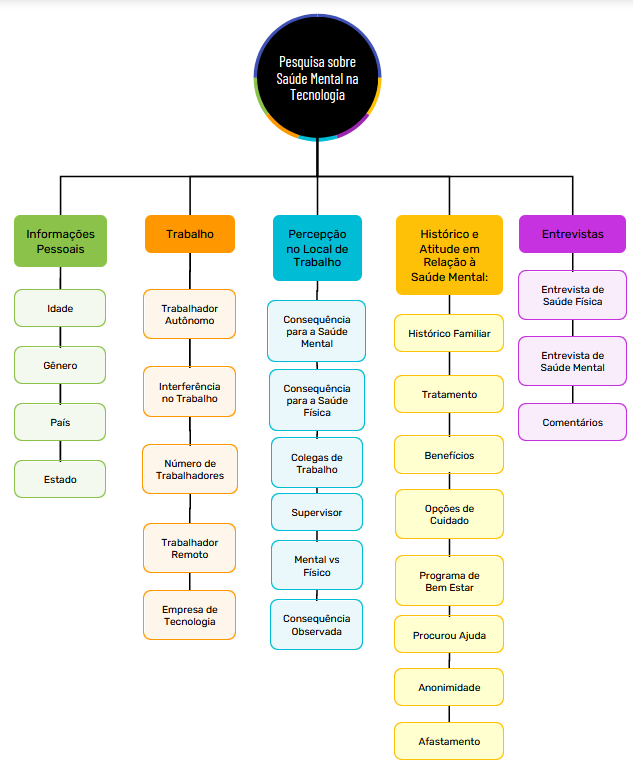
O conjunto de dados usado neste estudo apresenta 1260 instâncias e abrange uma ampla gama de informações. O mesmo inclui detalhes demográficos, como idade, gênero, país e estado (para residentes dos EUA). Também se aprofunda em aspectos relacionados ao emprego, com perguntas relacionadas ao status de autoemprego, histórico familiar de doenças mentais e se os indivíduos buscaram tratamento para condições de saúde mental. Contudo, é importante ressaltar que a base possui ausências nos atributos *work\_infere* e *self\_employed,* onde o primeiro representa 20% dos dados, e os nulos foram substituídos por “Unknown”, enquanto o segundo representa 1.4% dos dados, e os valores nulos foram substituídos por “No”.

Além disso, o conjunto de dados explora o impacto da saúde mental no trabalho, observando se os indivíduos sentem que sua condição interfere em seu emprego. Fatores relacionados ao empregador, como a oferta de benefícios de saúde mental, opções de cuidados, programas de bem-estar e recursos para buscar ajuda são examinados. São feitas perguntas sobre anonimato, licença médica e percepções sobre discutir questões de saúde mental com empregadores e colegas lançam luz sobre a atitude do ambiente de trabalho em relação à saúde mental. [[2]](https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey)

Por fim, o conjunto de dados inclui um campo aberto para comentários adicionais, dando aos participantes a oportunidade de compartilhar insights e experiências pessoais relacionadas à saúde mental.[[2]](https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey) Totalizando 26 atributos, tendo como maior parte atributos categóricos com respostas limitadas, a exceção de alguns com alta variedade de respostas, visto que as perguntas eram abertas. Idade sendo o único atributo numérico presente na base.

A seguir, é apresentado na Figura 1 todos os atributos citados acima de forma visual, organizando-os por tipo de informação fornecida de forma a facilitar o entendimento dos dados.

**Figura 1 – Descrição visual dos atributos.**



**3. Etapas de pré-processamento**

O pré-processamento da base iniciou-se pela remoção dos campos de *timestamp*, que no contexto de aprendizado de máquina era irrelevante, *comments*,que são campos em sua maioria vazios e sem padrão, e *state*,que era válido somente para os Estados Unidos. Em seguida, foi tratado o atributo idade, que possuía alguns dados incorretos, como idades negativas, e outros que foram considerados incoerentes e podem ser *outliers* que prejudicam a análise, como idades menores que 15, por se tratar de uma pesquisa em um local de trabalho, ou maiores que 100. Assim, filtrou-se as idades entre 15 e 100 anos.

O próximo atributo tratado foi o de gênero, que possuía respostas muito variadas e que, em alguns casos, indicavam o mesmo dado, como respostas “M” equivalentes a *“Male”*. Esses dados foram tratados de forma que houvesse apenas três tipos de resposta: *“Male”*, *“Female”* e *“Other”*. A base escolhida já estava balanceada, logo essa etapa não foi necessária.

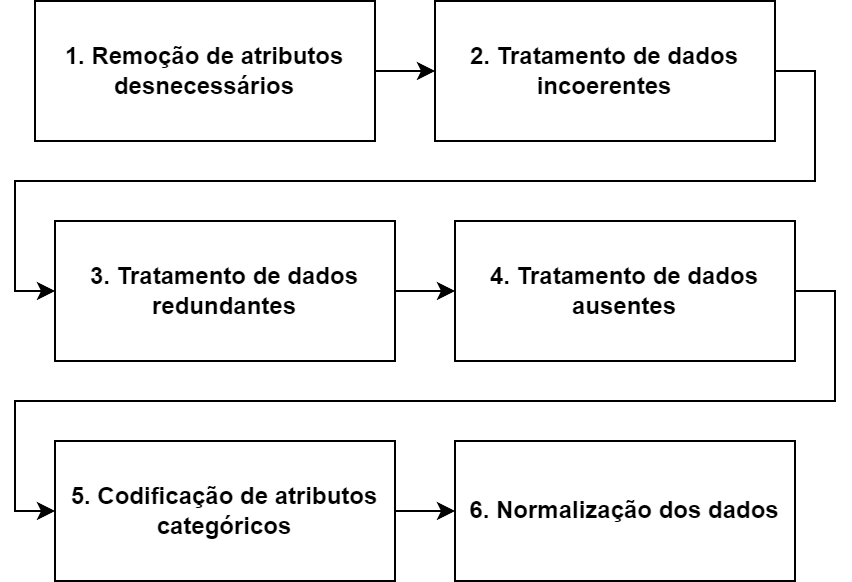
Por fim, foram identificados valores ausentes apenas nos atributos *work\_interfere* e *self\_employed*. Para o primeiro atributo, optou-se por substituir os dados ausentes por “*Unknown*”, indicando que não se sabe se a condição mental daquela pessoa interfere no seu trabalho. Já os dados que faltavam no segundo atributo foram preenchidos com “*No*”, preferindo-se considerar que aqueles que deixaram o campo em branco não eram trabalhadores autônomos.

A codificação dos atributos categóricos foi feita, em sua maioria, utilizando-se da técnica do *OneHotEncoder*, tendo em vista que não havia uma ordenação entre as respostas e os atributos em sua maioria não eram binários. Apenas os atributos *country* e *treatment* foram codificados pelo *LabelEncoder*, o primeiro por possuir muitos tipos de resposta (logo o *OneHot* geraria muitas colunas, dificultando o processo) e o segundo por ser binário, indicando as classes “Sim” e “Não”.

A última etapa de pré-processamento realizada foi a normalização dos dados. Para isso, foi utilizado o *MinMaxScaler*, que dimensiona os valores do conjunto de dados em um intervalo de 0 a 1, de forma que os valores mínimos se tornam 0 e os valores máximos se tornam 1, mantendo a proporção relativa dos valores originais. Essa técnica foi aplicada em todos os atributos remanescentes na base, exceto o atributo *target*.

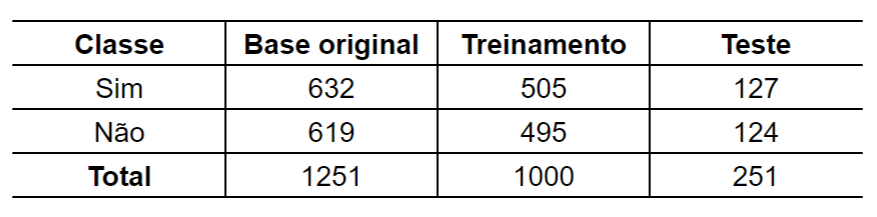
A Figura 2 mostra as etapas descritas, na ordem em que foram realizadas.

**Figura 2 – Ordem das etapas de pré-processamento.**



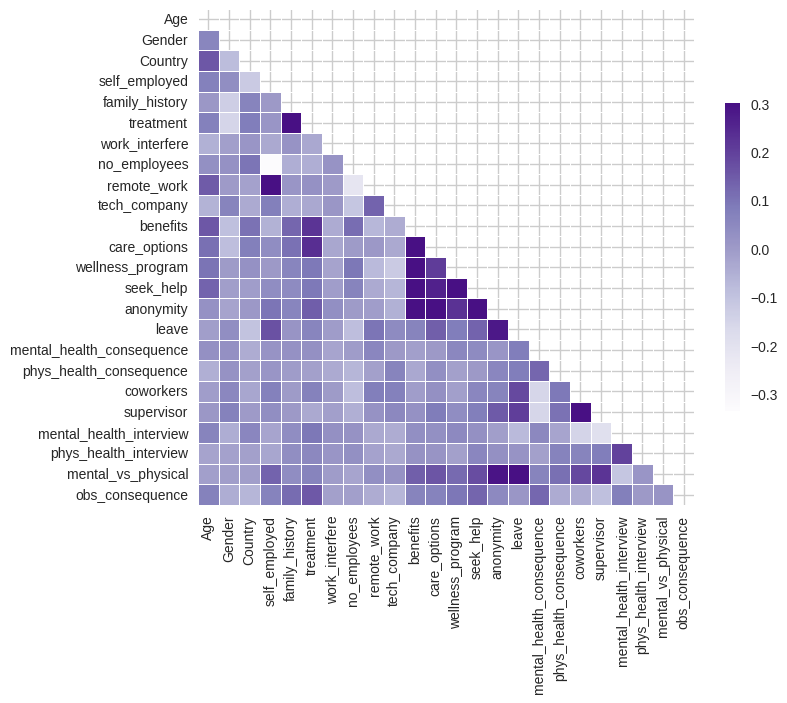
Após o processamento da base, separou-se a coluna referente ao *target* da classificação, o atributo *treatment*, do restante das colunas para realizar o treinamento do modelo. Para o conjunto de teste, foram separados 20% da amostragem da pesquisa. A Tabela 1 traz a quantidade de instâncias, de cada classe, divididas para teste e treinamento além do seu total.

**Tabela 1 – Divisão dos dados em teste e treinamento.**

****

Vale ressaltar que foi feita uma análise de correlação entre os atributos, sendo o resultado mostrado a seguir, na Figura 3.

**Figura 3 – Correlação entre os atributos.**



**4. Descrição dos métodos utilizados**

No contexto desta análise, utilizamos, primeiramente, um método de aprendizado de máquina: o *Random Forest*. Optamos por este como a principal abordagem devido à sua capacidade de aprimorar a precisão e a robustez do modelo. Este algoritmo é uma técnica de *ensemble* que combina múltiplas Árvores de Decisão para fornecer previsões mais confiáveis e reduzir o risco de *overfitting*. Desta forma, o uso do *Random Forest* nos permite explorar relações complexas nos dados, ajudando-nos a identificar qual método se adapta melhor ao nosso conjunto de dados e aos objetivos da análise.[[4]](https://link.springer.com/chapter/10.1007/978-3-642-16239-8_8)

Para selecionar os hiperparâmetros que gerassem modelos de árvores melhores no *Random Forest*, utilizou-se o *GridSearch*, que testa diferentes combinações de hiperparâmetros em busca da melhor. Utilizou-se, para isso, o seguinte espaço de busca: critérios (gini ou entropia), profundidade máxima (4, 5, 6, 7 ou 8) e cálculo da quantidade de atributos (raiz quadrada ou logaritmo na base 2). A combinação de hiperparâmetros retornada pelo *GridSearch* e utilizada no modelo treinado foi a seguinte: critério de entropia, profundidade máxima da árvore de 6 níveis e escolha da raiz quadrada para calcular a quantidade de atributos usados. Além disso, a quantidade de árvores geradas pelos *Random Forest* foi de 100.

Em seguida, foi implementado outro algoritmo para classificação, o XGBRFClassifier, uma variação do Extreme Gradient Boosting que utiliza Random Forest. Assim como o Random Forest, os algoritmos de Boosting utilizam da técnica ensemble, mas se diferenciam do primeiro por treinarem os modelos sequencialmente, corrigindo os erros dos modelos anteriores por meio de ponderações. Essa abordagem aumenta o peso dos registros que foram classificados de forma errada, permitindo uma melhoria no desempenho do modelo.

O *Random Forest*, por outro lado, é um algoritmo de *Bagging* e treina as Árvores de Decisão de forma paralela e independente. Tendo em vista a diferença de abordagem entre essas duas técnicas, o *XGBRFClassifier* foi implementado para permitir sua comparação com o *Random Forest* padrão.

Por fim, optou-se por implementar a Árvore de Decisão, escolhendo-a devido à sua capacidade de interpretabilidade, uma diferença importante em relação aos métodos anteriores. A Árvore de Decisão é uma estrutura hierárquica que representa um conjunto de regras de decisão, construídas a partir da análise dos atributos que melhor classificam as instâncias. Essa característica possibilita uma interpretação mais clara dos resultados do algoritmo, pois permite compreender o raciocínio e a lógica por trás das classificações efetuadas pelo modelo.

Apesar de ser possível utilizar ferramentas de interpretabilidade nos outros algoritmos implementados, a Árvore de Decisão é naturalmente interpretável, o que facilitaria esse processo. Assim, ao analisar os resultados da classificação utilizando esse método em comparação aos outros, busca-se escolher aquele com o melhor resultado na classificação. Contudo, não foi possível apresentar a árvore plotada devido ao fato de que sua altura ter sido relativamente grande, dificultando a visualização em imagem, e percebeu-se que a redução desse hiperparâmetro afetava significativamente o resultado da classificação.

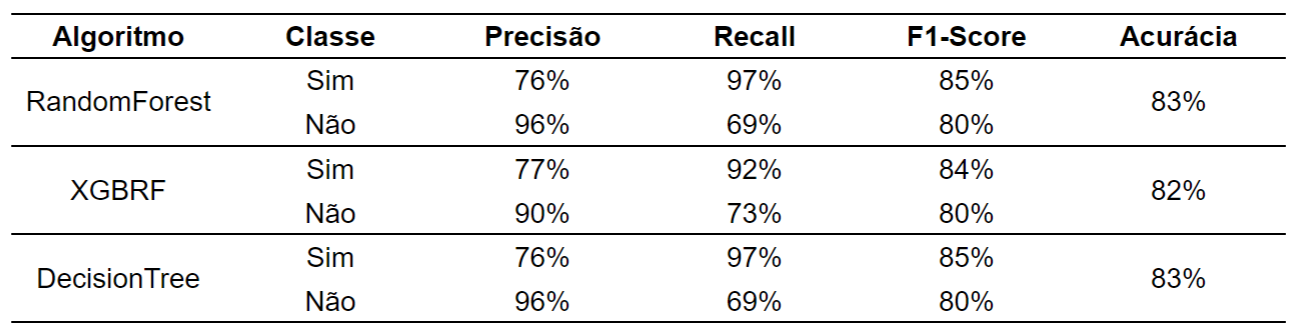
Assim como no *Random Forest*, foi utilizado o *GridSearch* para definir os hiperparâmetros da árvore. O *RandomizedSearch* também foi testado, mas o primeiro método demonstrou melhores resultados. O espaço de busca foi o seguinte: critérios (gini ou entropia), profundidade máxima (Nenhuma, 2, 4, 6, 8 ou 10) e cálculo da quantidade de atributos (Nenhum, raiz quadrada, logaritmo na base 2, 0.2, 0.4, 0.6, 0.8).

O código foi feito em um *notebook* do *Google Colab*, que utiliza o Python 3.10.12. Para execução do código, é necessário fazer o upload da [base](https://drive.google.com/file/d/140nin5nLechqeLB4Vj1ElQ7-yPbMxzp0/view?usp=drive_link) de dados em formato csvna pasta ‘*/content/sample\_data/’*.

**5. Resultados e discussões**

Após o treinamento dos modelos, as instâncias de teste foram classificadas e obtiveram-se os resultados apresentados a seguir. Cabe ressaltar que todos atributos foram utilizados, ou seja, todos foram relevantes para o resultado obtido pelo algoritmo. A Tabela 2 mostra as métricas de avaliação de cada um dos modelos criados pelos algoritmos implementados, para a classe “Sim” e para a classe “Não”, além da acurácia dos mesmos.

**Tabela 2 – Métricas de avaliação dos modelos.**



Como observado na Tabela 2, a acurácia dos modelos foi de 82% (*XGBRFClassifier*) e 83% (*Random Forest* e Árvore de Decisão), o que, no caso da base de dados estudada, é um indicativo bom sobre o modelo como um todo, visto que a base está balanceada. Esse fato diminui o risco da acurácia indicar um acerto alto do modelo não pelo fato dele estar acertando bem ambas as classes, mas pelo fato dele acertar mais a previsão de uma classe majoritária do que da minoritária. Portanto, percebe-se que os modelos treinados conseguiram classificar corretamente boa parte das instâncias, tanto da classe “Sim” quanto da classe “Não”.

Ao realizar uma análise mais particular de cada classe, observa-se que, para a classe “Sim”, o *recall* é mais alto que a precisão, em todos os algoritmos implementados, o que indica que há uma taxa mais baixa de falsos negativos. Já para a classe “Não”, ocorreu o contrário, a precisão foi maior que o *recall*, mostrando que há uma taxa menor de falsos positivos. Por fim, os valores do *F1-Score* de ambas as classes, que foram calculados pela média harmônica padrão, sem colocar mais peso na precisão ou no *recall*, foram similares entre os modelos, de exatamente 80% para a classe “Não” e entre 84% e 85% para a classe “Sim”.

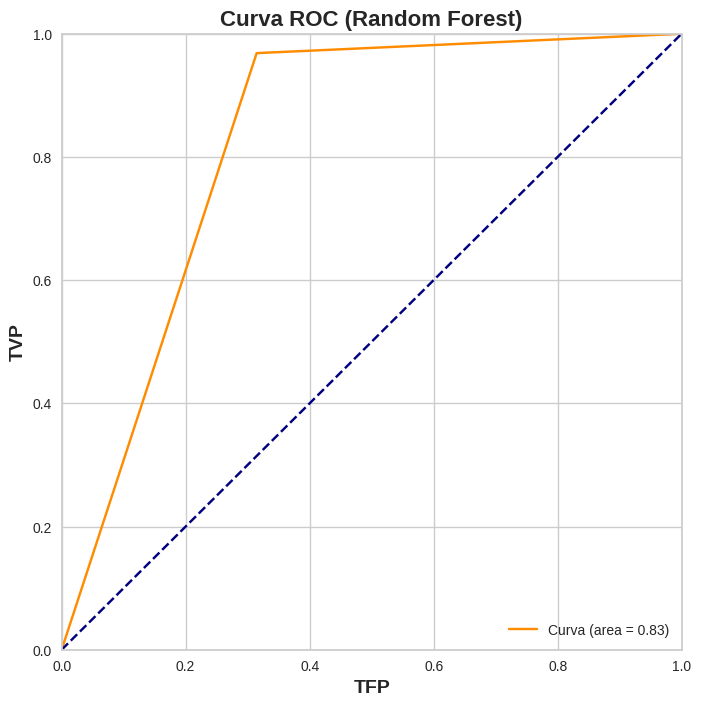
Além disso, percebe-se pela Tabela 2 que a acurácia do *XGBRFClassifier* foi de apenas um ponto percentual a menos em relação aos modelos de *Random Forest* e Árvore de Decisão. Essa pequena diferença indica que os resultados de ambos foram significativamente similares. Vale ressaltar, porém, que o *Random Forest* com *Grid Search* leva cerca de 2 minutos para concluir a execução, enquanto o *XGBRFClassifier* demanda menos de 10 segundos para alcançar uma acurácia superior. Isso demonstra não apenas a similaridade nos resultados, mas também a notável eficiência do modelo *XGBRFClassifier* em termos de tempo de processamento.

Outro ponto relevante é o fato de que a Árvore de Decisão, que também utiliza *Grid Search*, teve os mesmo resultados que o *Random Forest*. Além disso, da mesma forma que o *XGBRFClassifier*, este algoritmo gastou muito menos tempo que o *Random Forest* para ser executado, finalizando em menos de 30 segundos. Portanto, percebe-se que este algoritmo, apesar de ter apresentado bons resultados, se mostrou menos eficiente que os outros dois.

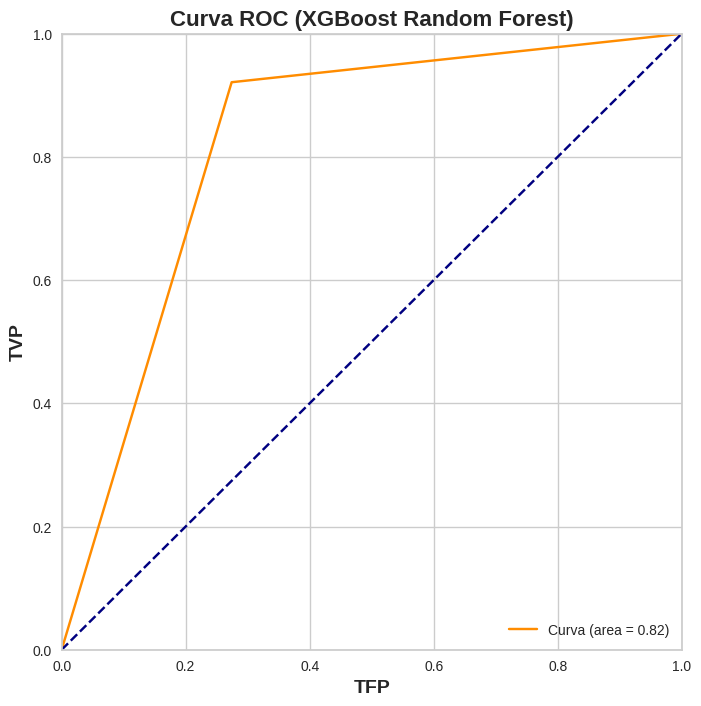
Outra métrica analisada após a execução dos algoritmos foi a Curva ROC (*Receiver Operating Characteristic*), uma ferramenta gráfica para aprendizado de máquina e estatística para avaliar o desempenho de classificadores binários, como algoritmos de classificação. Ela representa a relação entre a taxa de verdadeiros positivos (sensibilidade) e a taxa de falsos positivos (1 - especificidade).

Como os resultados da Árvore de Decisão e do *Random Forest* foram iguais, o gráfico plotado pelo cálculo da Curva ROC também foi o mesmo, sendo este apresentado na Figura 6. O mesmo foi feito para o *XGBRFClassifier* e seu gráfico é mostrado na Figura 7.

**Figura 4 – Curva ROC (*Random Forest* e Árvore de Decisão).**

****

**Figura 5 – Curva ROC (XGBRFClassifier).**

****

**6. Código desenvolvido**

Todo o código desenvolvido está disponível no [*COLAB*](https://drive.google.com/file/d/1rT4bym2l5PXnz221O1U4VAJrdtScTxOg/view?usp=sharing).

**7. Referências Bibliográficas**

1. Open Sourcing Mental Health (OSMI). ([https://osmihelp.org](about:blank))
2. Kaggle Dataset: Mental Health in Tech Survey. (<https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey>)
3. European Union - Compass for Action on Mental Health and Well-being. ([https://policycommons.net/artifacts/1943653/eu-compass-for-action-on-mental-health-and-well-being/2695422/](https://policycommons.net/artifacts/1943653/eu-compass-for-action-on-mental-health-and-well-being/2695422/)))
4. Devetyarov, D., Nouretdinov, I. (2010). Prediction with Confidence Based on a Random Forest Classifier. In: Papadopoulos, H., Andreou, A.S., Bramer, M. (eds) Artificial Intelligence Applications and Innovations. AIAI 2010. IFIP Advances in Information and Communication Technology, vol 339. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-16239-8_8>