

Objetivos da avaliação

Em uma abordagem construtiva, com níveis de complexidades dos desafios crescendo passo a passo, esta lista de tarefas foi desenvolvida para entender o conhecimento existente em cima dos problemas comuns que enfrentamos em clientes, nos projetos de Inteligência Artificial e Machine Learning.

Cada desafio proposto tem uma necessidade que nos ajuda a entender o resultado.

Para cada desafio é importante que a **matriz de talento**, que é obrigatória, seja preenchida. É a partir desta matriz que cruzamos o conhecimento com a entrega.

Não é obrigatório que todos desafios sejam respondidos, faça os que acredita que tem autonomia para trabalhar profissionalmente. Não buscamos um unicórnio, buscamos uma equipe plural. Caso decida não fazer algum desafio, siga para o próximo sem problemas.

Caso algum desafio que escolher resolver, dependa de outro, informe que parte do desafio já foi feita anteriormente (e avise qual é o número do desafio). Tome cuidado para assegurar que, ao resolver o problema que se referencia ao problema anterior, o problema anterior esteja funcional.

Pré-requisitos para os desafios de ferramenta

1. Instalação do **Visual Studio Community 2017**:
 - C# | R ou Python
 - .Net Core
2. Windows 10 SDK Build 17110 ou mais recente, que pode ser baixado em:
<https://developer.microsoft.com/windows/downloads/windows-10-sdk>
3. Ambiente do R ou Python

Matriz de talentos

Marque com um X cada elemento dos blocos abaixo (complexidade, desenvolvimento e tempo), informando o nível de complexidade que acredita que o desafio cobrou, se precisou de ajuda para resolver e a estimativa de tempo que levou para entregar sua solução.

Importante: Não existe resposta correta nesta matriz. Esta informação será usada apenas para entender seu nível de habilidade para resolver o problema. Sabemos que muitas soluções são feitas em conjunto, e isso é normal. Sabemos que pessoas tem experiências distintas, e isso é normal. Queremos um time multidisciplinar, e estas respostas nos ajudam a construir isso.

[illegible]

Desafio #1

Objetivo:

Este exercício é para cobrir o processo de consumo de API do Azure Machine Learning, onde será passado um elemento como entrada e terá outro como saída.

Necessidade:

Entregue um projeto Console em C# com o nome DESAFIO1.ZIP, que atenda os pontos:

- Acessar a URL da API:
<https://ussouthcentral.services.azureml.net/subscriptions/6e273f811bbd48fe870d6fe9187155e1/services/8116ce2309a64e2e95287513ca45dcb4/execute?api-version=2.0&format=swagger>
- É necessário passar uma chave de autenticação, use esta:
`v8tlTcwmjFs2G6jAGlvbx4JbfeVKw50bLZPmbQ1u7hzfb/SdmNj8G77+fE/E1180cY3CIAF6umuzaA6YvYm3zA==`
- Garanta a impressão do retorno, com três casas decimais de precisão, com a entrada de 40.

O que será avaliado:

- Sua técnica para criar projetos que consomem APIs;
- Pode ser que este link ajude a construir seu desafio:
<https://docs.microsoft.com/pt-br/azure/machine-learning/studio/consume-web-services>

Desafio #2

Objetivo:

A partir do resultado do Desafio #1, aumentar a capacidade do projeto. Atualmente o Desafio #1 recebe um valor numérico de entrada e retorna outro valor numérico. Espera-se que o valor numérico de entrada seja uma temperatura em Graus Célsius e sua saída seja um valor decimal que representa o % de chance de quebra da máquina, dada aquela temperatura.

Necessidade:

Entregue um projeto Console em C# com o nome DESAFIO2.ZIP, que atenda os pontos:

- Estender o projeto do #Desafio 1;
- Solicitar ao usuário qual é o tipo de temperatura que ele pretende informar (Célsius ou Fahrenheit);
- Receber a temperatura de acordo com o tipo informado. Caso seja Fahrenheit, deve fazer a conversão antes de enviar para a API.

O que será avaliado:

- Sua técnica para resolução de desafios no C#;

Desafio #3

Objetivo:

A partir do resultado do Desafio #2, aumentar a capacidade do projeto. Independente do tipo de temperatura e sua entrada, o modelo sempre espera que seja informada a temperatura em Célsius. A saída é um valor decimal. Trabalharemos com intervalos da saída para definir a atuação humana no modelo, usando a tabela abaixo:

- Menos de 40% de chance de quebra, retornar: **Tranquilo**;
- Entre 41 e 75% de chance de quebra, retornar: **Atenção**;
- Mais de 75% de chance de quebra, retornar: **Requer atuação**.

Necessidade:

Entregue um projeto Console em C# com o nome DESAFIO3.ZIP, que atenda os pontos:

- Estender o projeto do #Desafio 2;
- A partir da saída e os intervalos (tabela acima) como se fossem KPIs de um dashboard;
- Cada um dos intervalos deve possuir um tipo de retorno (mensagem impressa no console) diferente para o usuário final.

O que será avaliado:

- Sua técnica para resolução de desafios no C#;

Desafio #4

Objetivo:

Utilizar o ONNX chamado ReconhecimentoDeImagem.onnx (fornecido junto ao desafio) para criar um aplicativo UWP que vai rodar localmente e reconhecer um Sticker Figure (boneco de palito). Você pode desenhar o seu boneco em qualquer ferramenta e submeter para o seu projeto da forma que desejar.

Para ajudar na integração, leia esta documentação:

<https://docs.microsoft.com/en-us/windows/uwp/machine-learning/overview>

Necessidade:

Entregue um projeto Universal Windows em C# com o nome DESAFIO4.ZIP, que atenda os pontos:

- Possuir uma forma de submeter a sua imagem;
- Integrar o arquivo ONNX no projeto;
- Receber a categoria da imagem submetida;
- Apresentar a saída da classificação e o valor probabilístico do retorno.

O que será avaliado:

- Sua técnica para integrar arquivos ONNX com C#;

Desafio #5

Objetivo:

Criar um modelo de classificação a partir da base de dados pública Iris que pode ser encontrada nesta URL: <https://archive.ics.uci.edu/ml/datasets/Iris>

Necessidade:

Criar um modelo de classificação em R ou Python, salvar todos elementos como DESAFIO5.ZIP, que tenha:

- O dataset deve ser o Iris;
- Explicar o porque escolheu o algoritmo de classificação usado (e porque não usou outros);
- Receber as entradas específicas (escolher as variáveis pertinentes);
- Justificar a escolha das variáveis;
- Apresentar a saída classificada;
- Justificar a performance do modelo de classificação, com base nos métodos de avaliação pertinentes.

O que será avaliado:

- Sua técnica para escrever modelo de classificação;

Desafio #6

Objetivo:

Criar um modelo de Regressão a partir da base de dados pública Air Quality, que pode ser encontrada nesta URL: <https://archive.ics.uci.edu/ml/datasets/Air+Quality>

Necessidade:

Criar um modelo de regressão em R ou Python, salvar todos elementos como DESAFIO6.ZIP, que tenha:

- O dataset deve ser o Air Quality;
- A variável alvo é a **PT08.S4** (óxido de tungstênio);
- Predizer o valor da variável alvo, dada uma combinação de valores das variáveis explicativas;
- Explicar o porque escolheu o algoritmo de regressão usado (e porque não usou outros);
- Receber as entradas específicas (escolher as variáveis pertinentes);
- Justificar a escolha das variáveis;
- Apresentar a saída estimada;
- Justificar a performance do modelo de regressão, com base nos métodos de avaliação pertinentes.

O que será avaliado:

- Sua técnica para escrever modelo de Regressão;

Desafio #7

Objetivo:

Criar um modelo de agrupamento a partir da base de dados pública Daily and Sports Activities, que pode ser encontrada nesta URL: <https://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities>

Necessidade:

Criar um modelo de cluster em R ou Python, salvar todos elementos como DESAFIO7.ZIP, que tenha:

- O dataset deve ser o Daily and Sport Activities;
- Escolher e informar dois esportes (um mais calmo e outro mais agitado)
- Analisar os eixos X, Y e Z do sensor Acelerômetro apenas do Dorso.
- Explicar o porque escolheu o algoritmo de cluster usado (e porque não usou outros);
- Receber as entradas específicas (escolher as variáveis pertinentes);
- Justificar a escolha das variáveis;
- Apresentar os clusters de forma gráfica;
- Justificar a performance do modelo de cluster, com base nos métodos de avaliação pertinentes.

O que será avaliado:

- Sua técnica para escrever modelo de agrupamento;

Desafio #8

Objetivo:

Criar um modelo de Recomendação de Produtos a partir da base de dados pública Online Retail que pode ser encontrada nesta URL: <http://archive.ics.uci.edu/ml/datasets/Online+Retail>

Necessidade:

Criar um modelo de recomendação em R ou Python, salvar todos elementos como DESAFIO8.ZIP, que tenha:

- O dataset deve ser o Online Retail;
- Explicar o porque escolheu o algoritmo de recomendação usado (e porque não usou outros);
- Receber as entradas específicas (escolher as variáveis pertinentes);
- Justificar a escolha das variáveis;
- Justificar a performance do modelo de recomendação, com base nos métodos de avaliação pertinentes.

O que será avaliado:

- Sua técnica para escrever modelo de Sistema de Recomendação;

Desafio #9

Objetivo:

Criar um modelo de Evasão a partir da base de dados pública Teaching Assistant Evaluation que pode ser encontrada nesta URL: <http://archive.ics.uci.edu/ml/datasets/Teaching+Assistant+Evaluation>

Necessidade:

Criar um modelo de evasão em R ou Python, salvar todos elementos como DESAFIO9.ZIP, que tenha:

- O dataset deve ser o Teaching Assistant Evaluation;
- Explicar o porque escolheu o algoritmo de churn usado (e porque não usou outros);
- Receber as entradas específicas (escolher as variáveis pertinentes);
- Justificar a escolha das variáveis;
- Justificar a performance do modelo de evasão, com base nos métodos de avaliação pertinentes;
- **Importante: Considere a possibilidade a variável Class Attribute [6ª coluna] como sendo a variável alvo para o experimento (qual a chance de evolução ser da classe 1 para 2, ou da classe 2 evoluir para a 3). Escolha apenas uma das “evoluções”, e informe qual foi sua escolha.**

O que será avaliado:

- Sua técnica para escrever modelo de Cálculo de Evasão (Churn);

Desafio #10

Objetivo:

Criar um modelo de Séries Temporais Históricas a partir da base de dados pública Diabetes, que pode ser encontrada nesta URL: <http://archive.ics.uci.edu/ml/datasets/Diabetes>

Necessidade:

Criar um modelo de Séries Temporais em R ou Python, salvar todos elementos como DESAFIO10.ZIP, que tenha:

- O dataset deve ser o Diabetes;
- Explicar o porque escolheu o algoritmo de Séries Temporais usado (e porque não usou outros);
- Receber as entradas específicas (escolher as variáveis pertinentes);
- Justificar a escolha das variáveis;
- Justificar a performance do modelo de séries temporais, com base nos métodos de avaliação pertinentes;
- **Importante: Remover um conjunto de dados do final do dataset para usar como validação (sendo que estes dados não devem ser usados para treino, mas depois deve ser usado para comparar o desempenho do que foi predito com estes realizados)**

O que será avaliado:

- Sua técnica para escrever modelo de Análise de Séries Temporais Históricas;

Desafio #11

Objetivo:

Escolher uma das bases de dados dos desafios acima e trabalhar com Análise Exploratória;

Necessidade:

Criar um modelo de Exploração em R ou Python, salvar todos elementos como DESAFIO11.ZIP, que tenha:

- Informar qual dos dataset acima foi escolhido;
- Apresentar as explorações de forma gráfica;
- Explorar o máximo que puder.

O que será avaliado:

- Sua técnica para Explorar Dados sem um contexto prévio;