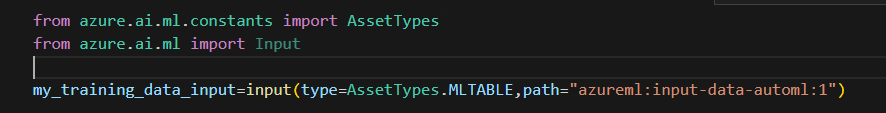
Monitoramento do Treinamento de Modelos em Notebook Jupyter com MLFLOW

Antes de executar um experimento de AutoML, Você precisa preparar os dados.

* A classificação requer dados tabulares.
* Criar um ativo de dados no Azure Machine Learning.
* Criar um ativo de dados MLTable: Armazene seus dados em uma pasta junto com um arquivo MLTable.



Entender o dimensionamento e a normalização

* O AutoML aplica o dimensionamento e a normalização a dados numéricos automaticamente, ajudando a impedir que qualquer recurso de grande escala domine o treinamento
* Durante um experimento de AutoML, várias técnicas de dimensionamento ou normalização serão aplicadas.
* Configurar a definição de recursos opcional:
  + Imputação de valor ausente para eliminar os nulos do conjunto de dados de treinamento
  + Codificação categórica para converter recursos categóricos em indicadores numéricos
  + Remoção de recursos de alta cardinalidade, como Ids de registros
  + Engenharia de recursos (por exemplo, derivação de partes de data individuais dos recursos de DateTime).

Executar um experimento de MLAutomatizado

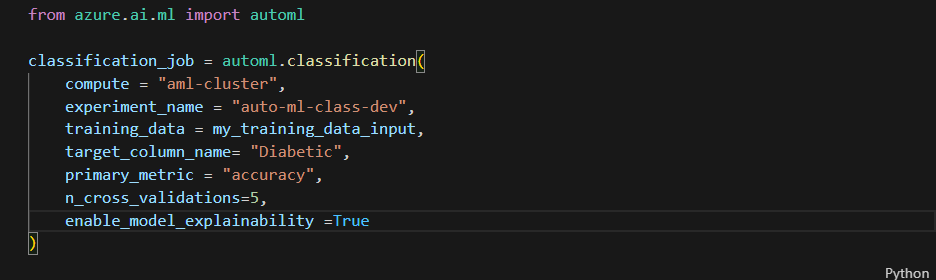
* Quando você quiser treinar um modelo de classificação, o AutoML escolherá entre uma lista de algoritmos de classificação:
  + Regressão logística
  + GBM (Computador de gradiente boosting) leve
  + Árvore de decisão
  + Floresta aleatória
  + Naïve Bayes
  + SVM (Computador de vetor de suporte) Linear
  + XGBoost
  + E outros
* Por padrão, o AutoML fará uma seleção aleatória entre o intervalo completo de algoritmos para a tarefa especificada.
* Você pode optar por bloquear a seleção de algoritmos individuais, o que poderá ser útil se souber que os dados não são adequados a um tipo específico de algoritmo.
* Talvez você também queira bloquear determinados algoritmos se precisar cumprir uma política que restrinja o tipo de algoritmos de aprendizado de máquinas que pode usar em sua organização.

Configurar um experimento de AutoML

Ao usar o SDK do Python (v2) para configurar um experimento ou trabalho de AutoML, configure o experimento usando o AutoML class.

1. Especifique a métrica primária: O “melhor” modelo é baseado no primary\_metric
2. Defina os limites: Para minimizar os custos e o tempo gasto com treinamento, você pode definir limites para um experimento ou trabalho de AutoML usando set\_limits().
3. Defina as propriedades do treinamento: O autoML experimentará várias combinações de definição de recursos e algoritmos para treinar um modelo de machine Learning.

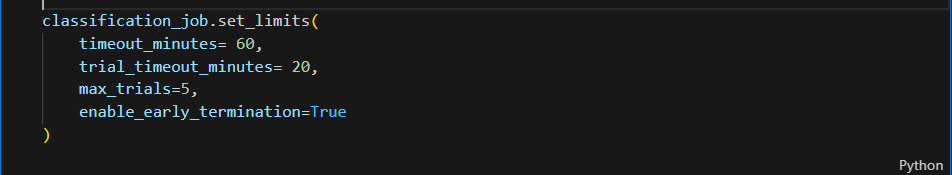
Para configurar um experimento de AutoML para classificação, use a função automl.classification:



Definir os limites

Há varias opções para definir limites para um experimento de AutoML:

1. Timeout\_minutes: Número de minutos após o qual o experimento de AutoML completo é encerrado.
2. Trial\_timeout\_minutes: Número máximo de avaliações ou de modelos que serão treinados.
3. Enable\_early\_termination: Se deseja encerrar o experimento se a pontuação não estiver melhorando no curso prazo.

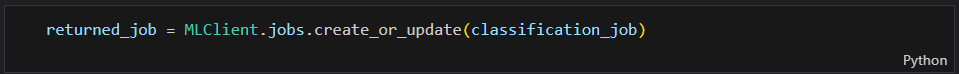


Enviar um experimento de AutoML

Você pode enviar um experimento de AutoML com o seguinte código:

O experimento consistirá em trabalhos filhos:

1. A definição de recursos é realizada em um trabalho filho.
2. Cada modelo é treinado em um trabalho filho separado.



Recuperar a melhor execução e o modelo

* Ao examinar os modelos no AutoML, você pode identificar facilmente a melhor execução com base na métrica primária especificada.
* Na guia Modelos do experimento de AutoML, você poderá editar as colunas se quiser mostrar outras métricas na mesma visão geral.
* Para explorar ainda mais um modelo, você pode gerar explicações para cada modelo que foi treinado.

**Hands-on: Encontrar o melhor modelo de classificação com Machine Learning Automatizado.**

**Materiais de Apoio**

**Os materiais complementares e de apoio que oferecemos têm como objetivo fornecer informações para facilitar e enriquecer a sua jornada de aprendizado no curso "Monitoramento do Treinamento de Modelos em Notebooks Jupyter com MLflow". Aqui você encontrará links úteis, como slides, repositórios e páginas oficiais, além de dicas sobre como se destacar na DIO e no mercado de trabalho 😉**

**Recursos Adicionais**

**Durante este conteúdo, compreendemos os fundamentos da engenharia de prompts. Para ajudá-lo a aprofundar o conhecimento, disponibilizamos a seguir o material complementar contendo os conteúdos e links apresentados no curso:**

**Slide: Monitoramento do Treinamento de Modelos em Notebooks Jupyter com MLflow.pptx**

**Dicas e Links Úteis**

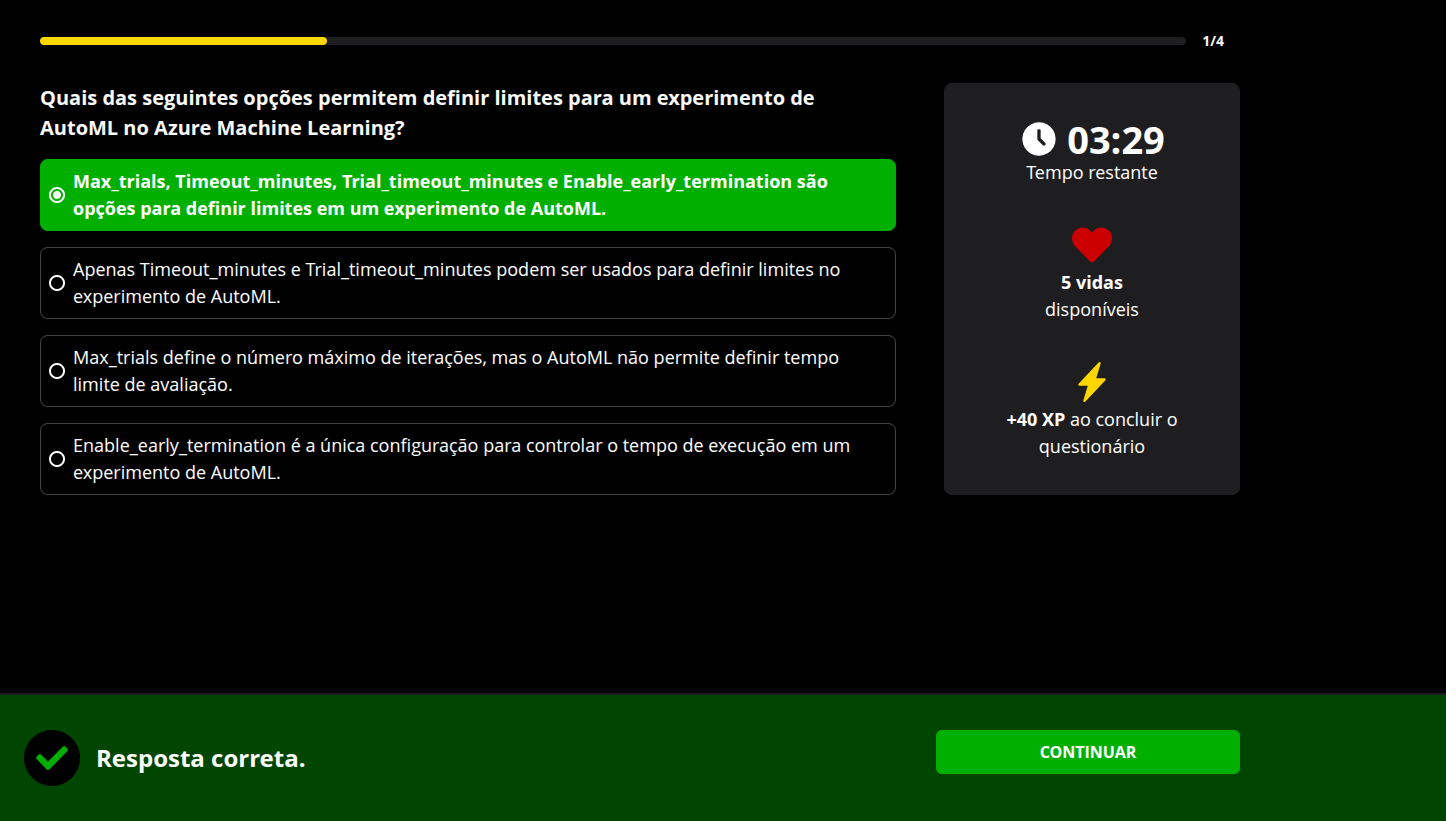
**Para se desenvolver ainda mais e se destacar na DIO e no mercado de trabalho, sugerimos os seguintes recursos:**

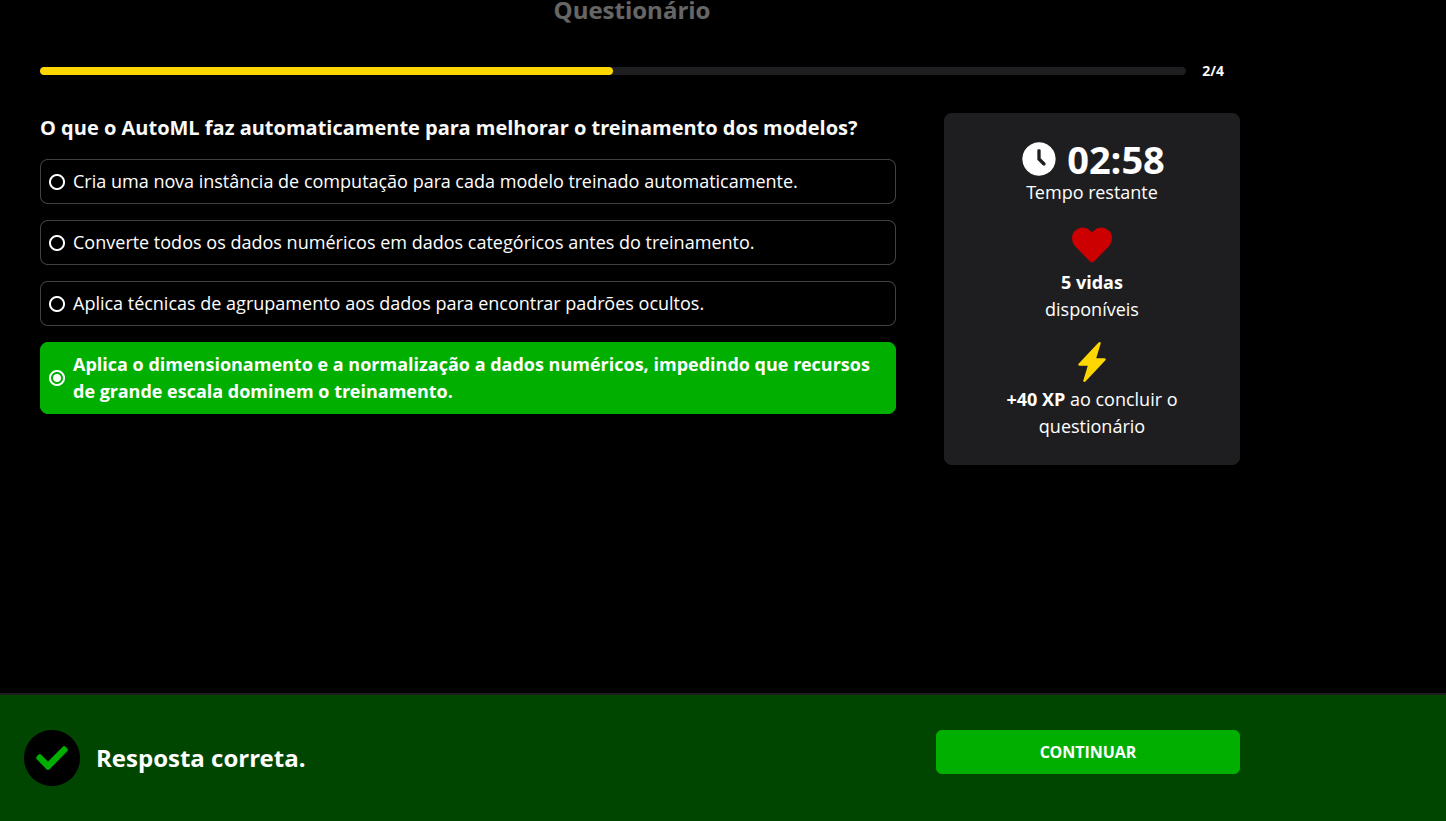
**Artigos e Fórum da DIO: Compartilhe seus conhecimentos e dúvidas através dos artigos (visíveis globalmente na plataforma da DIO) e nos fóruns específicos para cada experiência educacional, como nossos Bootcamps.**

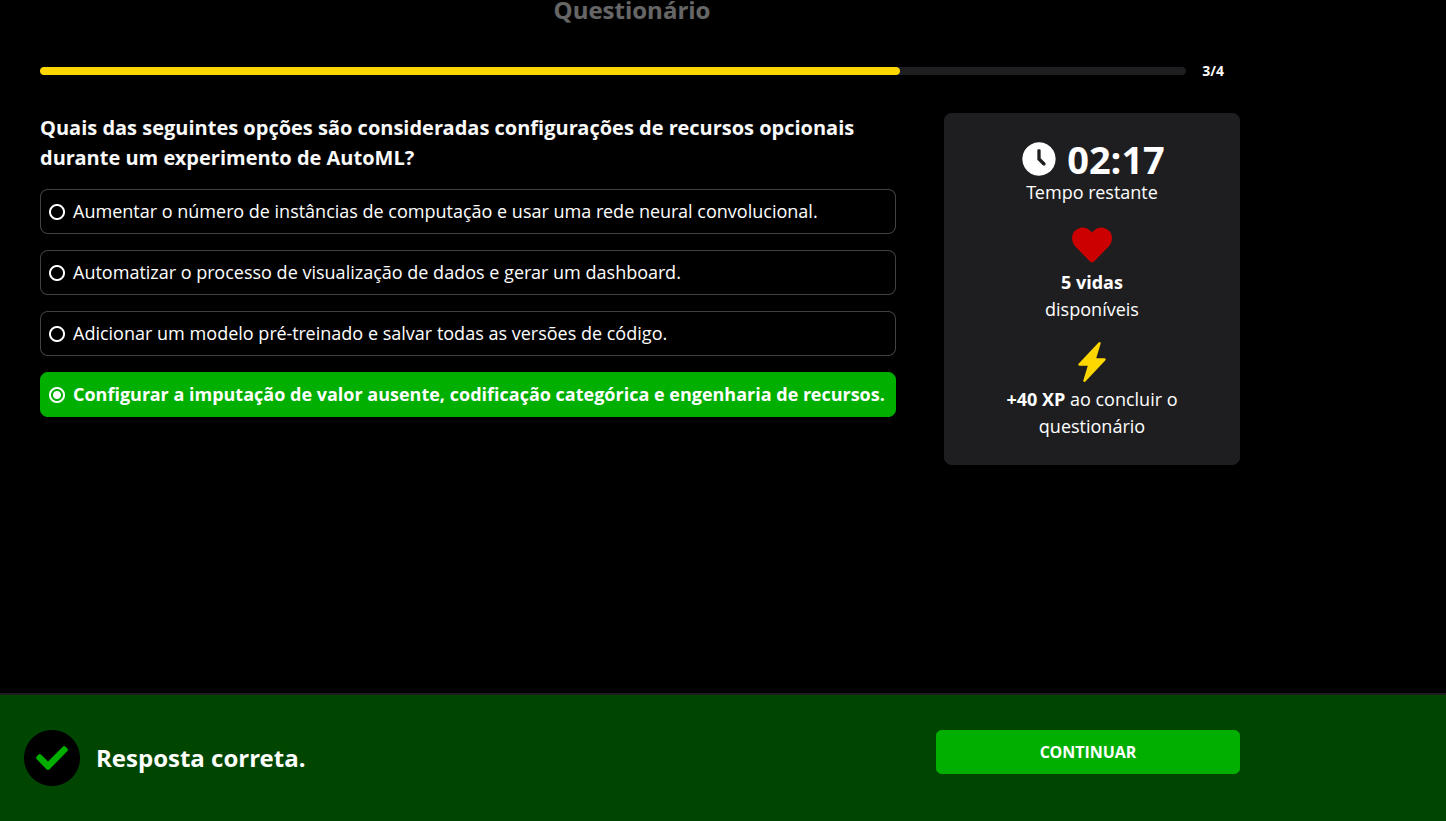
**Rooms: Participe do Rooms, uma ferramenta de bate-papo em tempo real onde você pode interagir com outros participantes dos nossos Bootcamps, compartilhando dúvidas, dicas e snippets de código.**

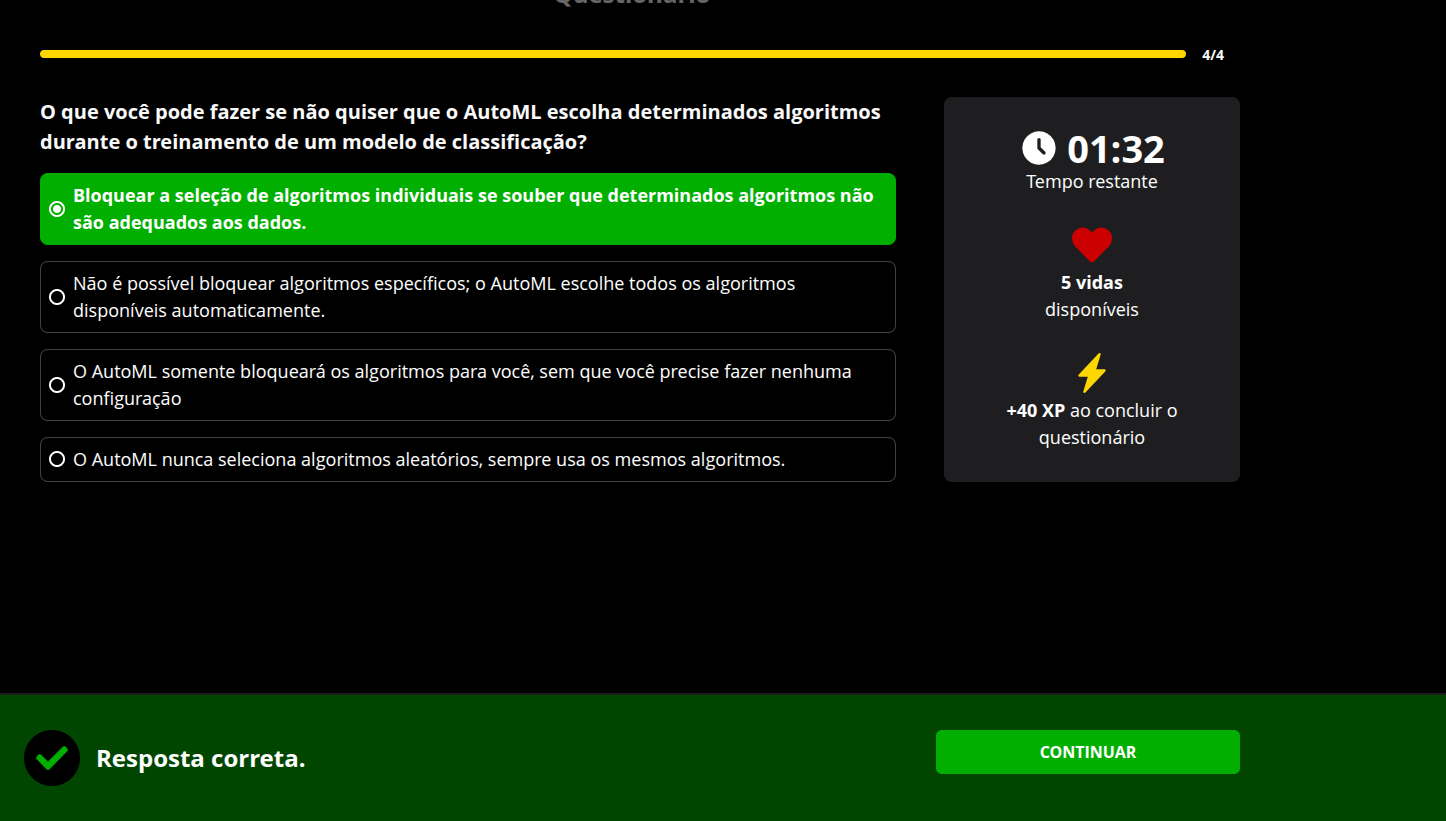
**Exploração na Web: Utilize motores de busca para aprofundar seu conhecimento sobre temas específicos. Páginas como o StackOverflow são recursos valiosos para encontrar soluções e expandir seu entendimento.**

**Com esses materiais complementares, você estará bem equipado para explorar todo o potencial e se destacar em suas iniciativas. Continue aproveitando as oportunidades de aprendizado, e não hesite em buscar mais conhecimento e compartilhar suas descobertas com a comunidade!**

****

****

****

****