

# Introdução à Inteligência Artificial

Lab 1 – Introdução ao Azure Machine Learning

## Visão Geral

Neste laboratório, você aprenderá a treinar e avaliar modelos de Machine Learning usando o Machine Learning do Azure.

## O que você precisa

Para completar este laboratório, você vai precisar:

- Uma conta Microsoft (como por exemplo um endereço *outlook.com*, *live.com* ou *hotmail.com*);
- Uma assinatura Microsoft Azure;
- Um computador Windows, Linux ou Mac OS X;
- Os arquivos do laboratório para este curso.

**Nota:** Para configurar o ambiente necessário para o laboratório, siga as instruções no [guia de configuração](#) deste curso.

## Exercício 1: Provisionando um Azure Machine Learning Workspace

Neste exercício, você criará um Azure Machine Learning Workspace para que você possa experimentar dados.

**Nota:** O portal do Microsoft Azure é aprimorado continuamente em resposta ao feedback dos clientes. As etapas deste exercício refletem a interface do usuário do portal do Microsoft Azure no momento da gravação, mas podem não corresponder exatamente ao design mais recente do portal.

### Criar e planejar uma Workspace do Azure Machine Learning

Antes de poder usar o Azure Machine Learning, você deve provisionar um espaço de trabalho. Nesse caso, você provisionará o Workspace em sua assinatura do Azure (observe que você também pode provisionar um Azure Machine Learning Workspace gratuito que não exija uma assinatura do Azure – Workspaces gratuitos estão sujeitos a algumas restrições).

1. No navegador web, navegue para <http://portal.azure.com>, e, se solicitado, faça login usando a conta da Microsoft associada à sua assinatura do Azure.
2. Dentro do portal Microsoft Azure, clique em **+ Create a resource**. Em seguida procure por **Machine Learning Studio Workspace** e crie uma nova Workspace Machine Learning Studio com as seguintes configurações:
  - **Workspace name:** Digite um nome exclusivo para a Workspace.
  - **Subscription:** Selecione sua Assinatura Azure

- **Resource Group:** Crie um novo grupo de recursos com um nome exclusivo.
  - **Location:** Selecione qualquer região disponível.
  - **Storage account:** Crie uma nova conta de armazenamento com um nome exclusivo.
  - **Workspace pricing tier:** Standard
  - **Web service plan:** Crie um novo plano de serviço da web com um nome exclusivo.
  - **Web service plan pricing tier:** DEVTEST Standard
  - **Pin to dashboard:** Not selected
3. Dentro do Portal Azure, visualize as **Notificações** para verificar se a implantação foi iniciada. Aguarde até que o espaço de trabalho seja implantado (isso pode levar alguns minutos).
  4. Clique em **All resources** e verifique se sua assinatura agora inclui os novos recursos a seguir:
    - Um Machine Learning Workspace.
    - Um Machine Learning Plan.
    - Uma Storage Account.

## Exercício 2: Explorando Dados com o Azure Machine Learning

Neste exercício, você criará um experimento no Azure Machine Learning para explorar um conjunto de dados que consiste em observações clínicas para pacientes em um estudo sobre indicadores que podem ser usados para detecção precoce de diabetes..

**Nota:** Os dados utilizados no exercício foram gerados por uma simulação baseada nos dados do conjunto de dados do Diabetes dos Índios Pima publicados pela Universidade da Califórnia, Escola de Informação e Ciência da Computação.

### Abra o Azure Machine Learning Studio

Agora que você tem um espaço de trabalho, pode usar o Azure Machine Learning Studio para trabalhar com dados.

1. No portal do Azure, navegue até o workspace criado no procedimento anterior.
2. Na aba dentro de sua Workspace, clique em **Launch Machine Learning Studio**. Isso abre uma nova página do navegador.
3. Na nova página do navegador, entre no Azure Machine Learning Studio usando a conta da Microsoft associada à sua assinatura Azure.
4. No Azure Machine Learning Studio, no canto superior direito, verifique se o nome da área de trabalho que você criou no procedimento anterior é exibido.

**Nota:** Se um nome de espaço de trabalho diferente for exibido, talvez você já tenha algumas workspaces associadas à sua conta - nesse caso, selecione seu novo workspace na lista suspensa.

### Visualizar conjuntos de DataSets

Os dados do exercício são fornecidos como arquivo de valores separados por vírgula (CSV). Além das amostras fornecidas, você pode carregar seus próprios arquivos de conjunto de dados e pode inserir dados de várias fontes, como o Armazenamento do Azure, o banco de dados SQL do Azure e outros.

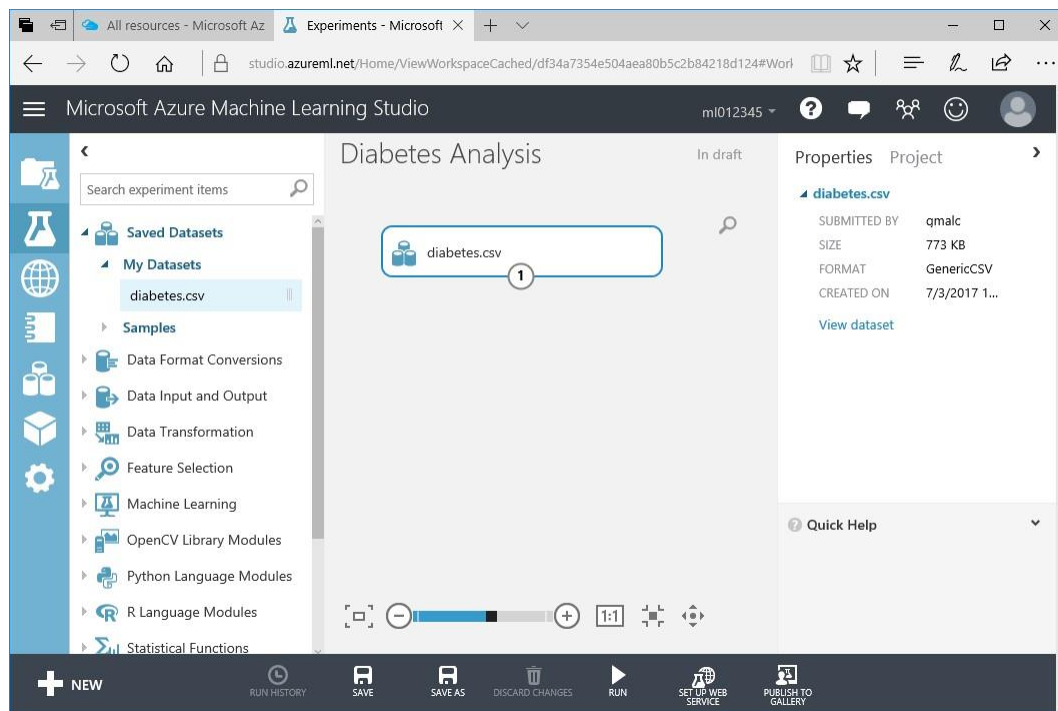
1. No Azure Machine Learning Studio, clique em **DATASETS**. Você não deve ter datasets próprios (clcando em **Samples** exibirá alguns conjuntos próprios de DataSets).
2. No canto inferior esquerdo, clique em **+ NEW**, e certifique-se de que a guia **DATASET** esteja selecionada.
3. Clique em **FROM LOCAL FILE**. Em seguida, na caixa de diálogo **Upload a new dataset**, navegue para selecionar o arquivo **diabetes.csv** na pasta **Lab01** onde você extraiu os arquivos do laboratório em seu computador local e insira os seguintes detalhes, conforme mostrado na imagem abaixo, e clique no botão (✓).

- **This is a new version of an existing dataset:** Não selecionado
  - **Enter a name for the new dataset:** diabetes.csv
  - **Select a type for the new dataset:** Generic CSV file with a header (.csv)
  - **Provide an optional description:** Dados do paciente.
4. Aguarde até que o upload do conjunto de dados seja concluído e verifique se ele está listado.

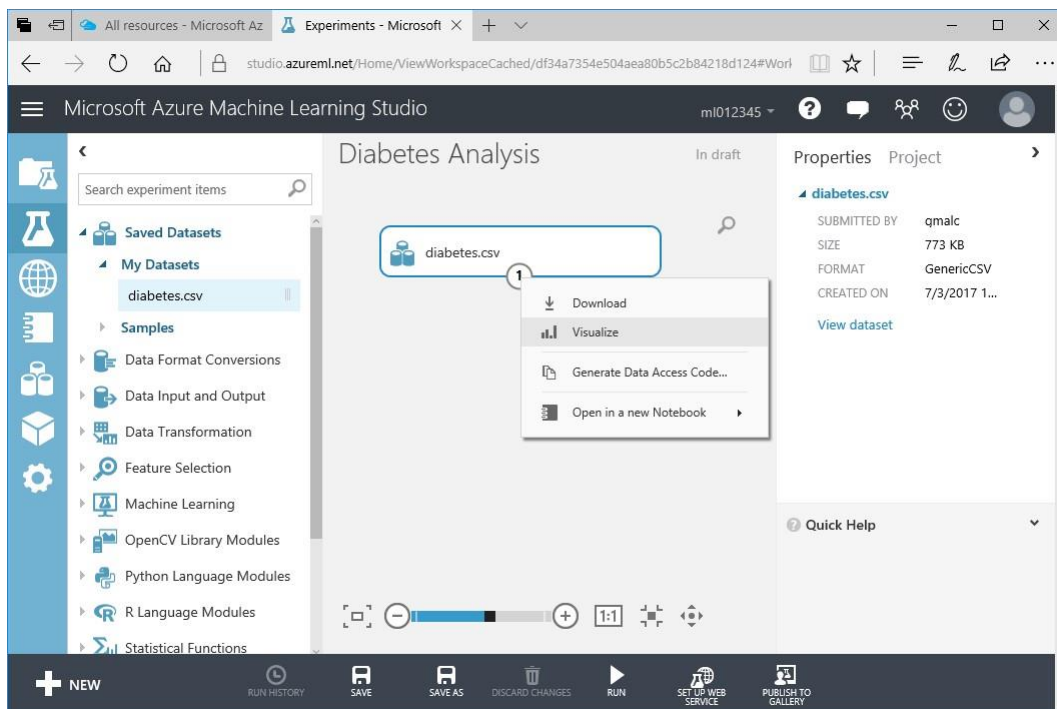
### Criando um experimento

Agora que você enviou seus dados, crie um experimento para explorá-lo.

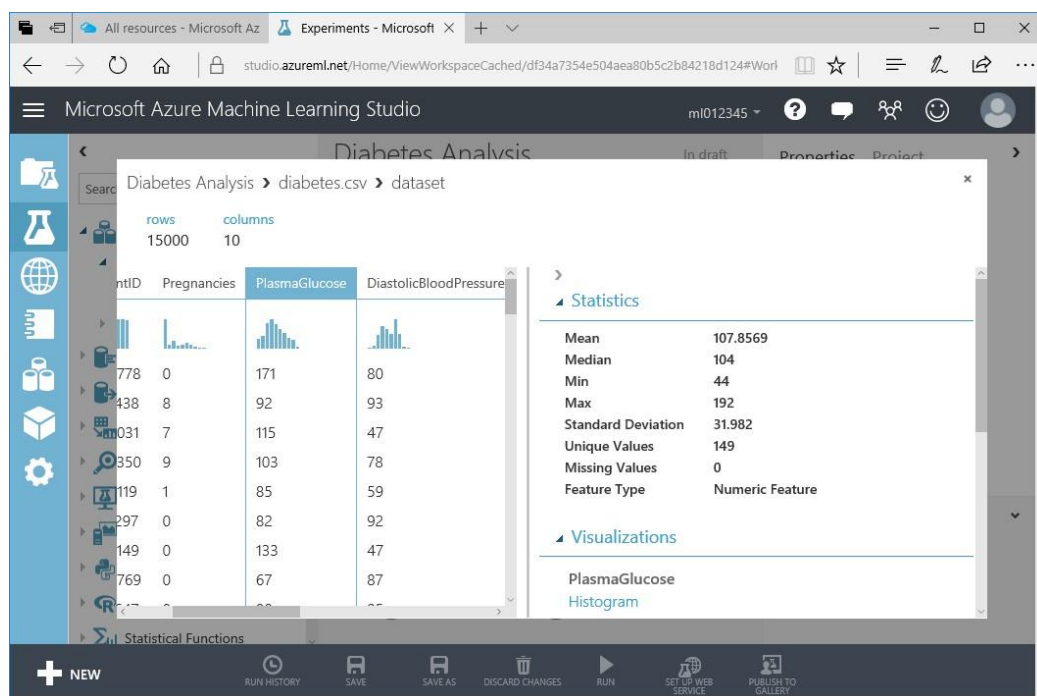
1. Dentro do Azure Machine Learning Studio, clique em **EXPERIMENTS**. Você ainda não deve ter experiências em seu Workspace.
2. No canto inferior esquerdo, clique em **+ NEW**, e verifique se a guia **EXPERIMENT** está selecionada. Em seguida, clique no bloco **Blank Experiment** para criar um novo experimento em branco.
3. Na parte superior da tela da experiência, altere o nome da experiência para **Diabetes Analysis**.
4. No painel de Itens de Experiência, expanda **Saved Datasets** e **My Datasets**, e arraste o Dataset **diabetes.csv** para a tela do experimento, como mostrado aqui:



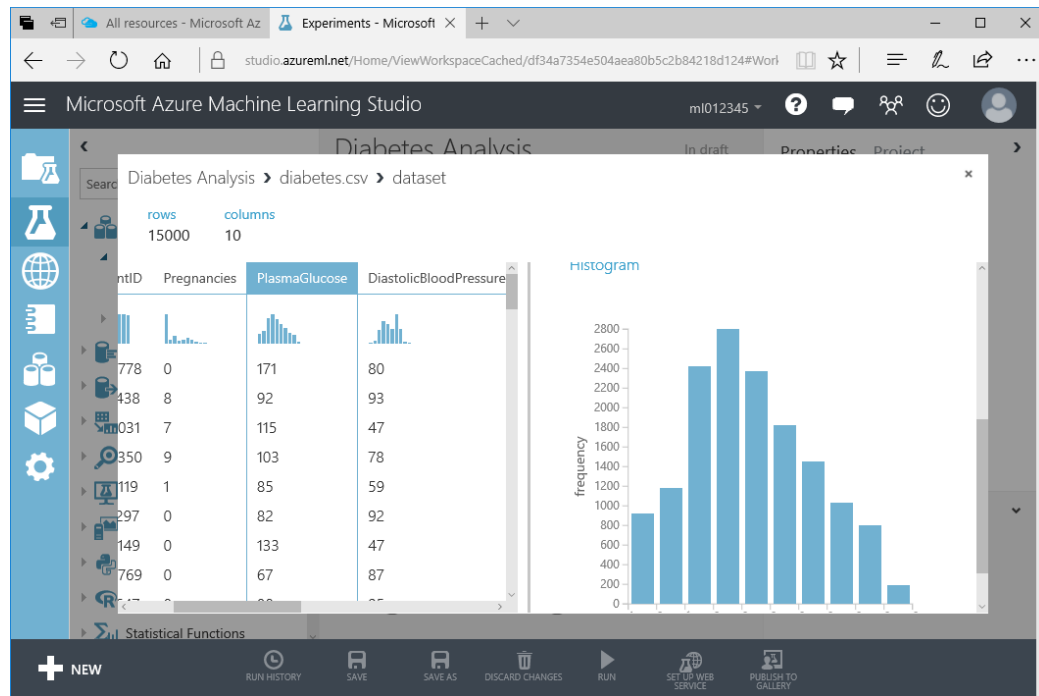
5. Clique com o botão direito do mouse na saída do dataset **diabetes.csv**, e clique em View Dataset como mostrado aqui:



6. Na visualização de dados, observe que o DataSet inclui um registro, geralmente chamado de Observação ou Caso, para cada paciente (identificado por um **PatientID**) e cada caso inclui características do paciente - neste exemplo, medições clínicas e detalhes médicos. O conjunto de dados também inclui uma coluna chamada **Diabetic**, que indica se o paciente é diabético - este é o rótulo que, em última análise, você deve treinar um modelo de Machine Learning para prever com base nas características de novos pacientes para os quais o diagnóstico de diabetes é desconhecido.
7. Observe o número de linhas e colunas no dataset e, em seguida, selecione o cabeçalho da coluna **PlasmaGlucose** e observe as estatísticas sobre essa coluna que são exibidas, conforme mostrado aqui.



8. Na visualização de dados, role para baixo, se necessário, para ver o histograma da **PlasmaGlucose**. Isto mostra a distribuição de diferentes pesos dentro dos registros do paciente no Dataset.



**Nota:** Esta distribuição está próxima do que os cientistas chamam de distribuição normal, na qual os valores mais frequentes para uma variável (neste caso, o nível de glicose no sangue do paciente) tendem a estar no meio do intervalo, com taxas aproximadamente semelhantes de declínio à medida que os valores se movem em direção às extremidades alta e baixa extremas. Em um histograma, isso cria uma visualização com uma forma de "curva em forma de sino".

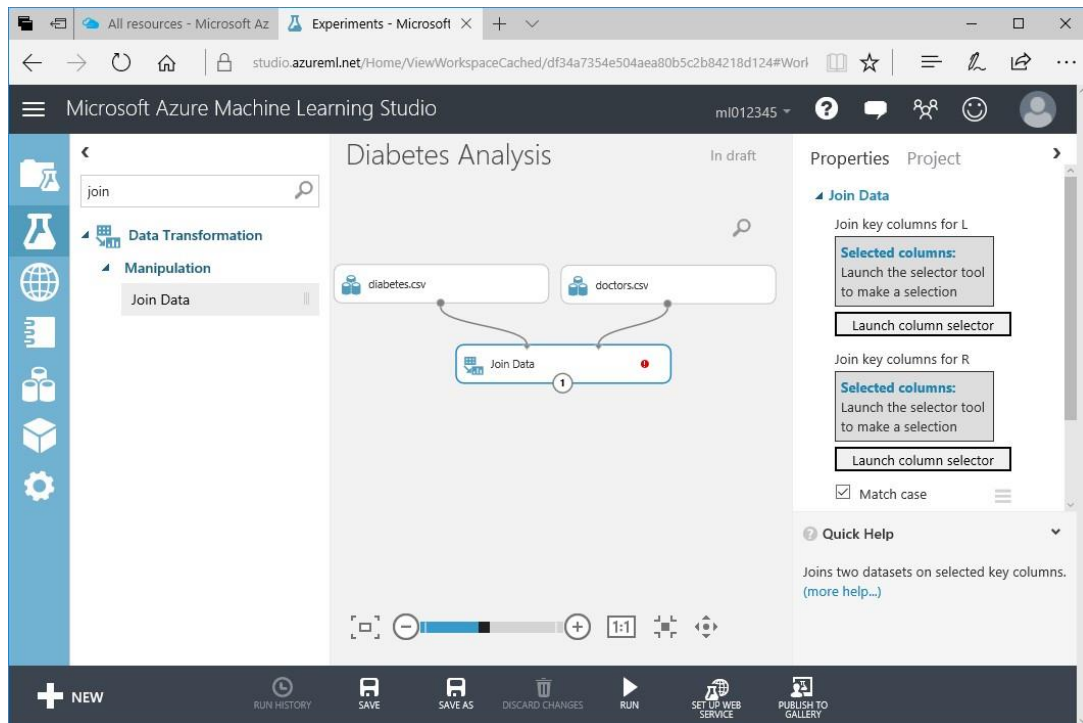
9. Feche a visualização e retorne à tela da experiência.

Os médicos de cuidados primários dos pacientes são armazenados em um arquivo separado, que você deve fazer upload como um dataset e adicioná-lo ao experimento

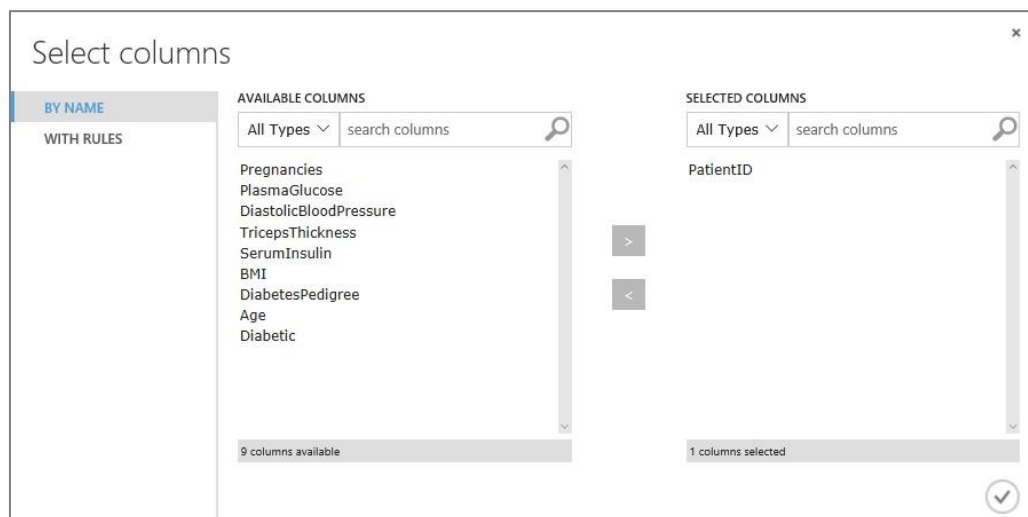
10. Com o botão esquerdo do mouse, clique em **+ NEW**, e selecione a aba **DATASET**.
11. Clique em **FROM LOCAL FILE**. Em seguida clique em **Upload a new dataset**, navegue até a pasta onde você extraiu a pasta **Lab01** e selecione **doctor.csv** e entre com os seguintes detalhes, depois clique no botão(✓):
- **This is a new version of an existing dataset:** Não selecionado
  - **Enter a name for the new dataset:** doctors.csv
  - **Select a type for the new dataset:** Generic CSV file with a header (.csv)
  - **Provide an optional description:** Dados dos médicos.
12. Aguarde até o upload do **doctors.csv** estiver completo, em seguida verifique se ele está listado no **My Datasets**.
13. Arraste o dataset **doctors.csv** pra a área de experimento, à direita do dataset **diabetes.csv**.
14. Visualize a saída dos **doctors.csv** e observe que ela contém uma linha para cada paciente, junto com o nome do médico do paciente.
15. Feche a visualização e retorne para a tela de experimento.

Agora, sua experiência contém dois datasets com um campo comum, **PatientID**. Você pode usar esse campo para combinar os dois datasets.

16. Dentro da caixa **Search experiment items**, digite *Join*, em seguida, na lista de itens filtrados, arraste o módulo **Join Data** para dentro da área de experimento abaixo das duas datasets.
17. Conecte a saída do conjunto de dados **diabetes.csv** à entrada **Dataset1** (esquerda) do módulo Join Data e conecte a saída do dataset **doctors.csv** à sua entrada **Dataset2** (direita), conforme mostrado aqui:

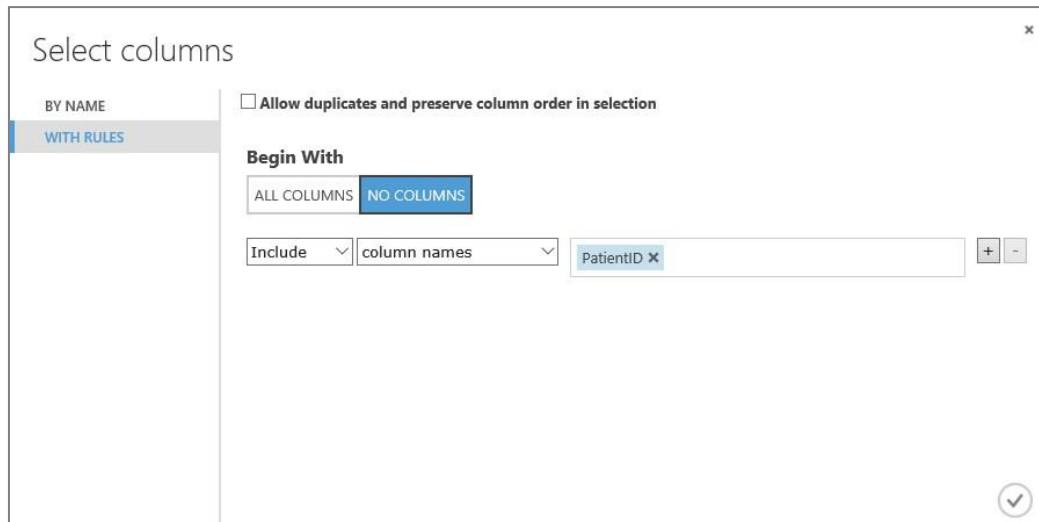


18. Com o módulo **Join Data** selecionado, dentro do painel **Properties**, em **Join key columns for L**, clique em **Launch column selector**.
19. Com a aba **BY NAME** selecionada, selecione a coluna **PatientID** e clique em **[>]** para adicioná-lo a lista **Selected Columns** como mostrado aqui:

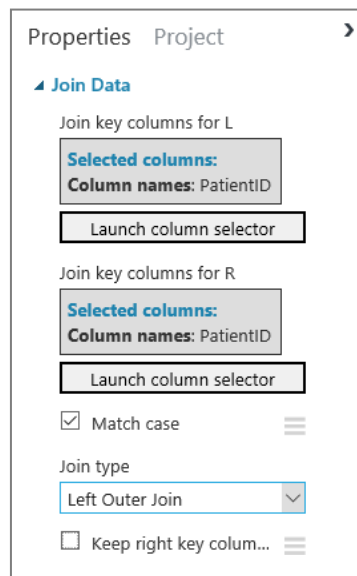


20. Clique no botão (✓) para confirmar a seleção da coluna.
21. Com o módulo **Join Data** selecionado, dentro do painel **Properties**, em **Join key columns for R**, clique em **Launch column selector**.

22. Clique na aba **WITH RULES**, e observe que você pode definir as regras para selecionar as colunas com base em seu nome, index e tipo de dado. Em **Begin With**, selecione **No Columns**. Em seguida selecione **Include**, **column names**, e **PatientID** como mostrado aqui (os nomes das colunas devem aparecer quando você começar a digitar). Em seguida clique no botão (✓).



23. Com o módulo **Join Data** selecionado, no painel **Properties**, desmarque a opção **Keep right key column in joined table** como mostrado aqui:



**Nota:** O uso de um Left Outer Join garante que a tabela unida inclua todos os usuários no DataSet “diabetes.csv” e seu médico correspondente do dataset “doctors.csv”. Se houver observações no dataset “diabetes.csv” sem registro de doctors.csv, os dados de diabetes serão retidos e o valor médico correspondente será nulo.

24. Na barra de ferramentas na parte inferior da tela de experimento, clique em **Salvar**. Em seguida, clique em **Run** para iniciar o experimento. Aguarde até que o experimento termine a execução (um sinal verde ✓ será exibido no módulo **Join Data**).
25. Visualize o **Result Dataset** no módulo **Join Data**, e observe que ela contém todas as colunas diabetes.csv e a coluna **Physician** correspondente ao DataSet “doctor.csv”.



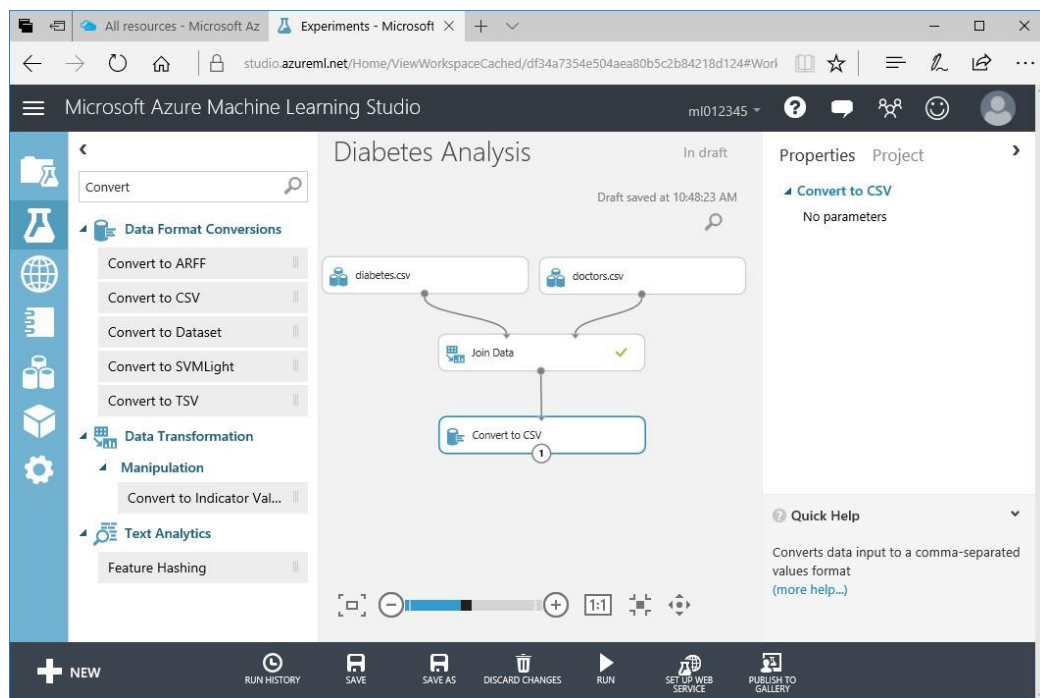
26. Feche a visualização.

**Nota:** Os experimentos do Azure Machine Learning normalmente incluem vários módulos que formam um fluxo de dados no qual os dados de um conjunto são limpos, filtrados e preparados para análise ou modelagem. O Azure Machine Learning inclui uma ampla variedade de módulos para operações de dados comuns, bem como módulos que permitem implementar lógica personalizada em Python, R ou SQL.

## Criar um Notebook

Os notebooks fornecem uma maneira conveniente para os cientistas de dados explorarem usando o código R ou Python.

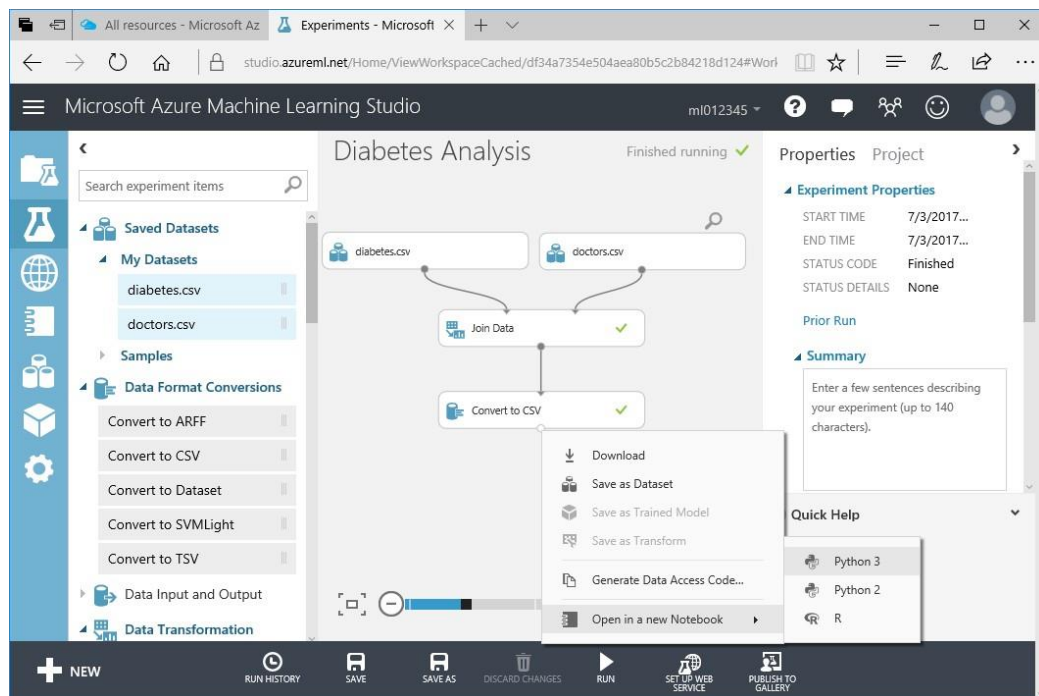
1. Na caixa **Search experiment items**, digite *Convert*, e em seguida na lista de itens filtrados, arraste o módulo **Convert to CSV** para a tela colocando-o abaixo do módulo **Join Data**.
2. Conecte o **Results Dataset** do módulo **Join Data** à entrada dataset “**Convert to CSV**” como mostrado aqui:



**Nota:** Você pode abrir um dataset CSV diretamente em um bloco de anotações. Se você tiver aplicado quaisquer transformações nos dados, você deverá convertê-los novamente em CSV antes de abri-los em um bloco de anotações.

3. Selecione o módulo **Convert to CSV** e, em seguida, no botão **Run**, clique em **Run Selected**. Isso executa o módulo selecionado usando a saída **UPSTREAM** da execução anterior. Aguarde até que ela tenha terminado a execução.
4. Clique com o botão direito do mouse no **Result Dataset** do módulo **Convert to CSV** e, em seguida, no **Open in a new Notebook** menu, clique em **Python 3**, como mostrado aqui:





**Nota:** Atualmente os Notebooks Jupyter do Azure Machine Learning suportam código personalizado escrito em R, Python 2 e Python 3. Os cientistas de dados podem usar qualquer a linguagem que preferir.

- Na nova guia aberta no navegador, visualize o Jupyter Notebook que foi criado. Observe que o notebook contém duas células. A primeira célula contém código que carrega o dataset CSV em um quadro de dados denominado **Frame**, similar a este:

```
from azureml import Workspace
ws = Workspace()
experiment = ws.experiments['335fdf9...f-id.0d127ca2e02245...']
ds = experiment.get_intermediate_dataset(
    node_id='ee772e8e-1600-49a5-a615-a6cfd9661208-21',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()
```

A segunda célula contém o código a seguir, que exibe um resumo do quadro de dados:

```
frame
```

- No menu **Cell**, clique em **Run All** para executar todas as células na pasta de trabalho. À medida que o código é executado, o símbolo **O** ao lado de **Python 3** no canto superior direito da página muda para um **●** e, em seguida, retorna para **O** quando o código tiver terminado de executar.
- Observe a saída da segunda célula, que mostra as primeiras linhas de dados do dataset, conforme mostrado aqui:

The screenshot shows a Jupyter Notebook interface. The first code cell (In [1]) contains the following Python code:

```
from azureml import Workspace
ws = Workspace()
experiment = ws.experiments['df34a7354e504aea80b5c2b84218d124.f-id.6fc2e3ad2de547c2adc4a441845d8f26']
ds = experiment.get_intermediate_dataset(
    node_id='93159db1-d48f-411e-a052-0c2b5b1bda6f-116',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()
```

The second code cell (In [2]) contains the variable 'frame', which displays a table of patient data:

	PatientID	Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	Seruminsulin	BMI
0	1354778	0	171	80	34	23	43.509726
1	1147438	8	92	93	47	36	21.240576
2	1640031	7	115	47	52	35	41.511523
3	1883350	9	103	78	25	304	29.582192
4	1424119	1	85	59	27	35	42.604536

8. Clique na célula 2 (que contém o código Frame) e, em seguida, na **Insert** menu, clique em **Insert Cell Below**. Isso adiciona uma nova célula ao notebook, sob a saída gerada pela célula 2.
9. Adicione o seguinte código à nova célula vazia (você pode copiar e colar esse código de **NotebookCode.txt** na pasta **Lab01**):

```
# Create Faceted Histograms
%matplotlib inline

import seaborn as sns
sns.set_style("whitegrid")

def create_facethists(df):
    import numpy as np
    cols = df.columns.tolist()[:-1]
    for col in cols:
        if(df[col].dtype in [np.int64, np.int32, np.float64]
            and df[col].name != "Diabetic"):
            g = sns.FacetGrid(frame, col="Diabetic")
            g.map(sns.distplot, col)
    return('Done')

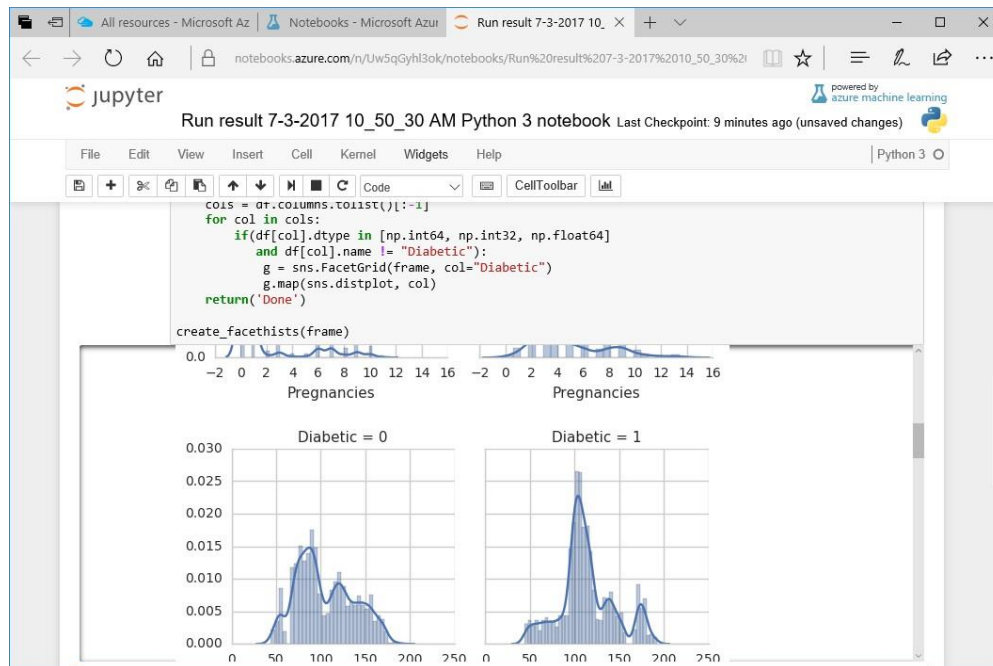
create_facethists(frame)
```

**Nota:** Este código cria histogramas para cada variável numérica, facetada pelo rótulo **Diabético**— em outras palavras, permite comparar a distribuição dos valores numéricos para pacientes diabéticos e não diabéticos.

10. Com a célula que contém o novo código selecionado, no menu **Cell**, clique em **Run Cells and Select Below** (ou clique no botão ►| na barra de ferramentas) para executar a célula, criando uma nova célula abaixo.

**Nota:** Pode ignorar os avisos que são gerados.

11. Visualize a saída do código, que consiste em uma série de pares de histogramas em um painel rolável, como mostrado aqui (você pode clicar na margem esquerda da saída da plotagem para mostrar os gráficos na janela principal do Notebook):



Observe que, a partir desse diagrama, você pode ver diferenças na distribuição de alguns recursos numéricos, dependendo se o paciente é ou não diabético.

12. Na célula vazia no final do bloco de notas, adicione o seguinte código (você pode copiar e colar esse código do **NotebookCode.txt** na pasta **Lab01**):

```
# Create Boxplots

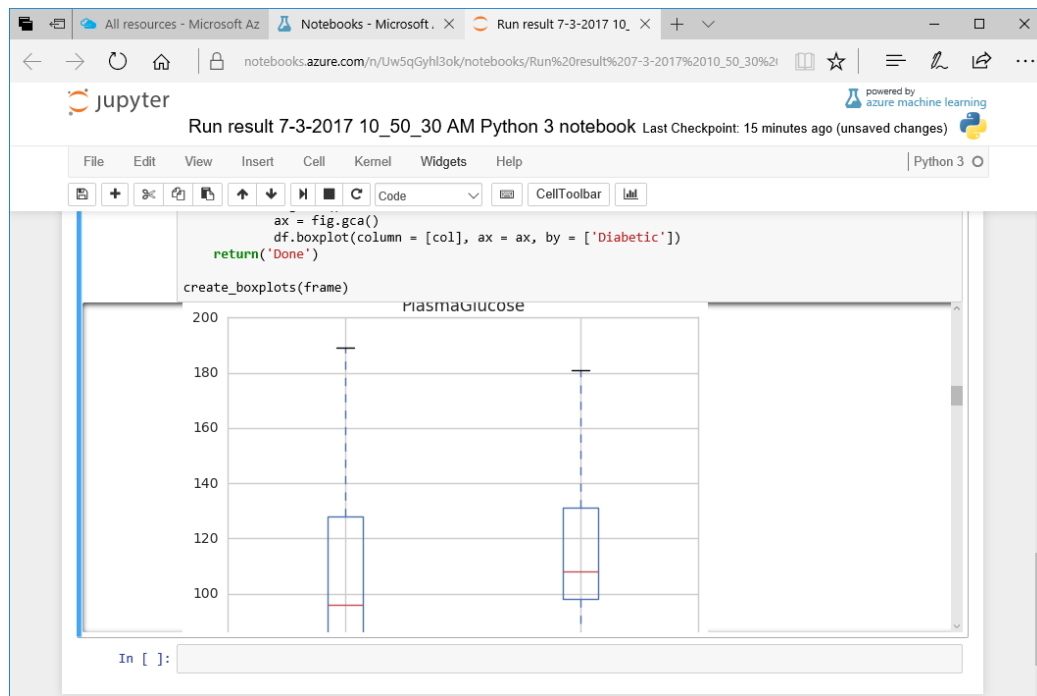
def create_boxplots(df):
    import numpy as np
    import matplotlib.pyplot as plt

    cols = df.columns.tolist()[::-1]
    for col in cols:
        if(df[col].dtype in [np.int64, np.int32, np.float64]
           and df[col].name != "Diabetic"):
            fig = plt.figure(figsize = (6,6))
            fig.clf()
            ax = fig.gca()
            df.boxplot(column = [col], ax = ax, by = ['Diabetic'])
    return('Done')

create_boxplots(frame)
```

**Nota:** Este código mostra a mesma comparação de distribuições de valores de recursos que os gráficos de caixa, o que facilita a visualização dos valores mediano e quartil nas distribuições.

13. Execute a célula e visualize a saída do código, que consiste em uma série de boxplots, conforme mostrado aqui:



14. No menu **File**, clique em **Save and Checkpoint** para salvar o notebook. Em seguida, no menu **File**, clique em **Close and Halt** para fechar o notebook.
15. No Azure Machine Learning Studio, salve o experimento, e em seguida clique na página **Notebooks** à esquerda e verifique se o notebook está listado lá.
16. Selecione o notebook e, na parte inferior da tela, clique em **Rename**. Em seguida, renomeie o notebook para **Diabetes Analysis**.

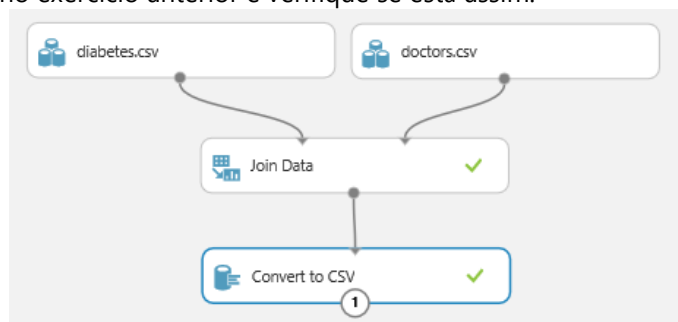
### Exercício 3: Treinando um Modelo de Classificação

Neste exercício, você usará o Azure Machine Learning para criar um modelo de classificação. O objetivo deste modelo é prever se um paciente é diabético com base nos recursos diagnósticos.

#### Prepare os dados

Como é frequentemente o caso com o aprendizado de máquina de qualquer tipo, alguma preparação de dados é necessária antes que você possa usar os dados para treinar um modelo.

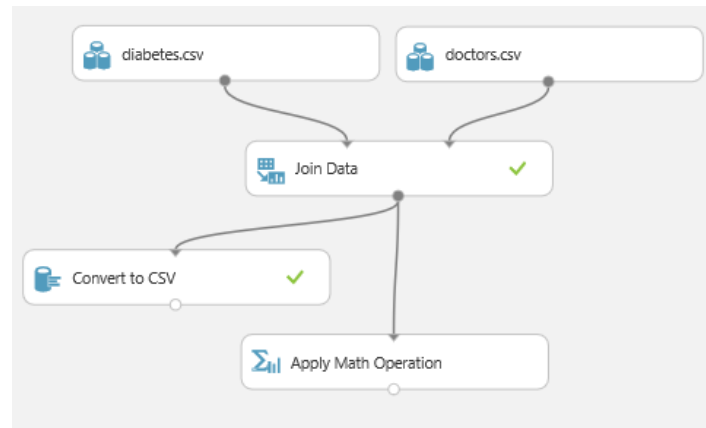
1. No Azure Machine Learning Studio, na página **Experiments**, abra a **Diabetes Analysis** experiência que você criou no exercício anterior e verifique se está assim:



A distribuição da coluna **Age** no dataset **diabetes.csv** é distorcida porque a maioria dos pacientes está na faixa etária mais jovem. Criar uma versão desse recurso que usa uma transformação de log natural pode ajudar a criar um relacionamento mais linear entre o **Age** e outros recursos e melhorar a capacidade de prever o rótulo **Diabetic**. Esse tipo de engenharia de recursos, como é chamado, é comum na preparação de dados de Machine Learning.

2. Adicione o módulo **Apply Math Operation** ao experimento e conecte a saída do módulo **Join Data** à sua entrada (ignorando o módulo **Convert to CSV**, que era necessário apenas para abrir os dados em um notebook).
3. Configure o módulo **Apply Math Operation** com as seguintes propriedades:
  - **Category:** Basic
  - **Basic math function:** Ln
  - **Column set:** Selecione apenas a coluna **age**
  - **Output mode:** Append

Sua experiência agora deve ficar assim:



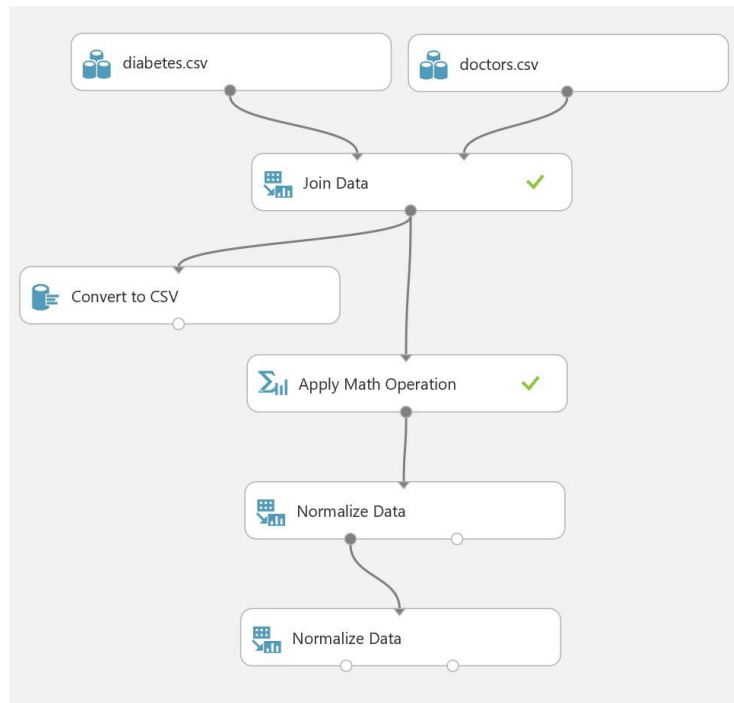
4. Salve sua experiência e selecione **Run** para atualizar o esquema de saída do módulo **Apply Math Operation** para verificar se a coluna Ln (Age) foi adicionada ao dataset.

É comum, ao preparar dados para o treinamento de modelo, dimensionar ou normalizar recursos numéricos, para que aqueles com grandes valores numéricos, não dominem o treinamento do modelo de Machine Learning, simplesmente porque os valores são grandes. A técnica de escala específica usada depende da distribuição estatística dos valores de dados – O módulo **Normalize Data** suporta cinco métodos de normalização. Nesse caso, as distribuições das colunas numéricas são bem diferentes. Algumas das características têm uma distribuição aproximadamente Normal (visualizada como histograma com uma curva em forme de sino). A normalização do **ZScore** (média-variância) é apropriada aqui. Outros recursos possuem distribuições distantes do Normal, portanto, a normalização do **MinMax** (forçando todos os valores em um intervalo limitado {0,1}) é mais apropriada para essas colunas. Portanto, você usará dois módulos **Normalize Data** para executar a normalização de todos os recursos numéricos.

5. Adicione um módulo **Normalize Data** ao experimento e conecte a saída do módulo **Apply Math Operation** à sua entrada.
6. Configure as propriedades do módulo **Normalize Data** da seguinte maneira:
  - **Transformation method:** ZScore
  - **Use 0 for constant columns when checked:** Checked
  - **Select columns** (Talvez será necessário digitá-los se você não tiver executado o experimento):
    - PlasmaGlucose
    - DiastolicBloodPressure
    - TricepsThickness
    - SerumInsulin
    - BMI
7. Adicione um segundo módulo **Normalize Data** ao experimento e conecte a saída do dataset **Transformed** (esquerda) do primeiro módulo **Normalize Data** para sua entrada.
8. Configure o novo módulo **Normalize Data** da seguinte forma:

- **Transformation method:** MinMax
- **Use 0 for constant columns when checked:** Checked
- **Select columns:**
  - Pregnancies
  - DiabetesPedigree
  - Age
  - Ln(Age)

Seu experimento agora deverá ficar assim:

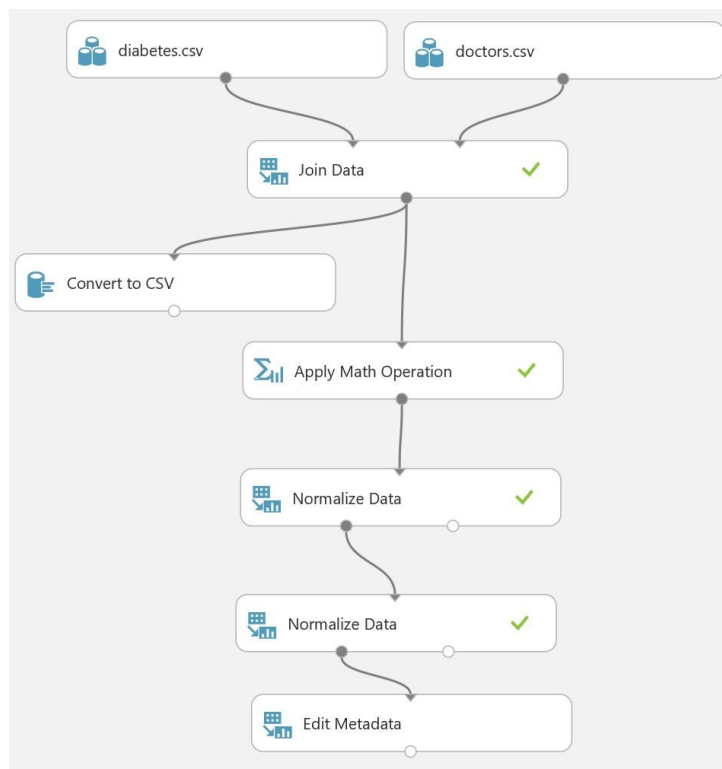


9. Salve sua experiência, selecione e execute o último módulo **Normalize Data**. Em seguida, visualize a saída esquerda do dataset **Transformed** do último módulo **Normalize Data** e observe que os valores dos recursos numéricos foram dimensionados.

É improvável que as colunas **PatientID** e **Physician** contribuam com qualquer informação preditiva para o diagnóstico de diabetes. Você irá, portanto, limpar esses recursos para que eles não sejam usados para treinar o modelo.

10. Adicione um módulo **Edit Metadata** ao experimento e conecte o dataset **Transformed** à saída esquerda do segundo módulo **Normalize Data** à sua entrada.
11. Configure o módulo **Edit Metadata** da seguinte forma:
  - **Selected columns:** *PatientID* and *Physician*
  - **Data type:** Unchanged
  - **Categorical:** Unchanged
  - **Fields:** Clear feature
  - **New column names:** *Leave blank*

Seu experimento agora deverá ficar assim:



12. Salve e execute a experiência e visualize a saída do módulo **Edit Metadata** e verifique se, quando você seleciona o cabeçalho da coluna **PatientID**, sua propriedade **Feature Type** é listada como numérica, mas quando você seleciona o cabeçalho da coluna **Pregnancies**, seu **Feature Type** é um recurso numérico.

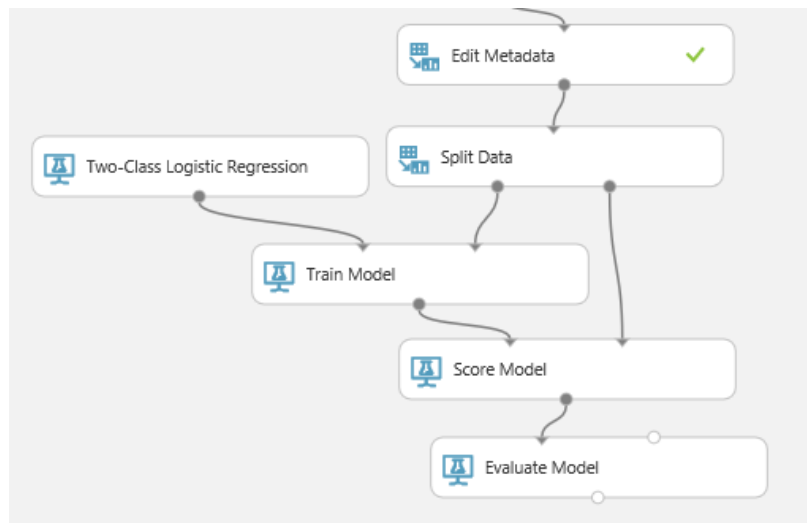
### Treine e Avalie o Modelo de Classificação

Agora que você preparou o dataset, você o usará para treinar e avaliar um modelo classificador de Machine Learning. Normalmente, ao treinar um modelo de aprendizado supervisionado, no qual os dados de treinamento incluem valores de rótulos conhecidos, você divide os dados em um conjunto de treinamento com o qual o modelo e o conjunto de teste validará previsões geradas pelo modelo treinado.

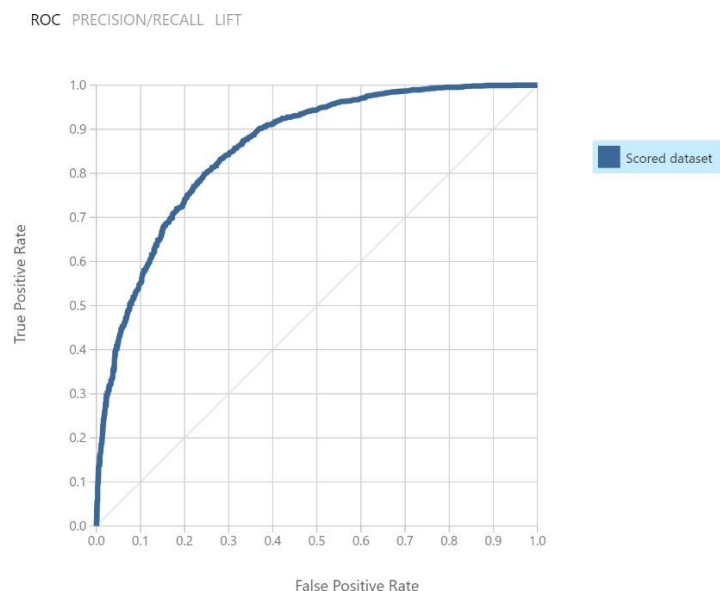
1. Adicione um módulo **Slit Data** ao experimento e conecte a saída do módulo **Edit Metadata** à sua entrada.
2. Configure o módulo **Slit Data** da seguinte maneira:
  - **Splitting mode:** Split Rows
  - **Fraction of rows in the first output dataset:** 0.7
  - **Random seed:** 1234
  - **Stratified split:** False
3. Adicione um módulo **Train Model** ao experimento e conecte a saída esquerda do módulo **Split Data** à entrada direita.
4. Configure as propriedades do módulo **Train Model** para definir a coluna **Label** como **Diabetic**.
5. Adicione um módulo **Two Class Logistic Regression** ao experimento e conecte sua saída à entrada esquerda do **Untrained Model** do módulo **Train Model**.
6. Configure o módulo **Two Class Logistic Regression** da seguinte maneira:
  - **Create trainer mode:** Single Parameter
  - **Optimization tolerance:** 1E-07
  - **L1 regularization weight:** 1
  - **L2 regularization weight:** 1



- **Memory size for L-BFGS:** 20
  - **Random number seed:** 1234
  - **Allow unknown categories:** Checked
- Adicione um **Score Model** ao experimento e conecte a saída do módulo **Train Model** à sua entrada esquerda. Em seguida, conecte a saída direita **Dataset2** do **Split Data** à entrada direita do dataset.
  - Adicione um **Evaluate Model** ao experimento e conecte a saída do módulo **Score Model** à entrada esquerda do dataset **Evaluate Model**.
  - Verifique se a parte inferior da sua experiência se parece com a figura abaixo.



- Salve e execute este experimento.
- Visualize a saída do módulo **Score Model** e examine os resultados. Compare a coluna de rótulos (Diabetic), que indica se o paciente é realmente diabético ou não, e a coluna **Scored Labels** que contém as previsões do modelo para pacientes diabéticos. Observe que a maioria dos casos é classificada corretamente, mas há alguns erros.
- Para examinar estatísticas de resumo, visualize a saída do módulo **Evaluate Model** e examine os resultados. Veja a curva ROC, que deve se parecer com a figura abaixo.



A curva ROC mostra o trade-off entre o **True Positive Rate** (Casos positivos corretamente classificados) e o **False Positive Rate** (Casos positivos incorretamente classificados). Esta curva está acima e à esquerda da linha de luz de 45 graus. Isso indica que o classificador é mais eficaz do que simplesmente adivinhar.

13. Role para baixo e examine as métricas de desempenho, que devem aparecer como mostrado abaixo.

True Positive	False Negative	Accuracy	Precision	Threshold		AUC
893	614	0.786	0.719	0.5		0.861
False Positive	True Negative	Recall	F1 Score			
349	2644	0.593	0.650			
Positive Label	Negative Label					
1	0					

Observe o seguinte sobre essas métricas:

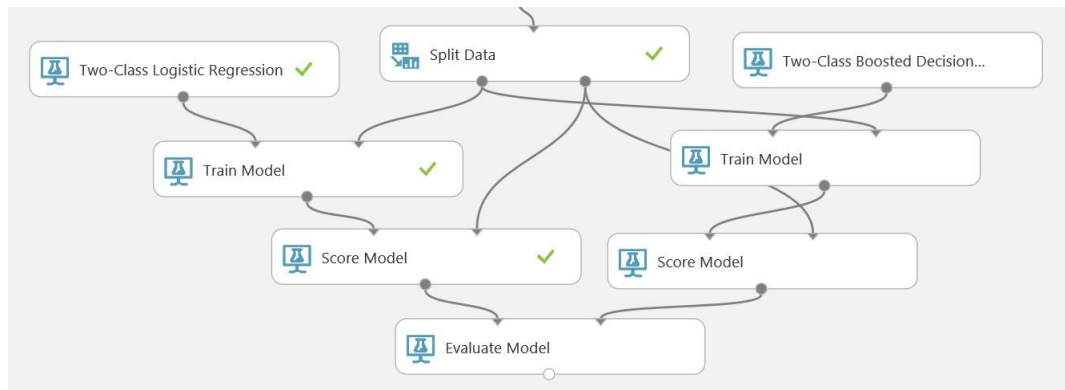
- **Confusion Matrix** mostra o número de Verdadeiros-Positivos e Verdadeiros-Negativos (casos classificados corretamente) e Falso-Negativos e Falso-Positivos (casos classificados incorretamente).
- **AUC** (*Área sob a curva*) é a área sob a curva ROC. Um classificador perfeito teria uma AUC de 1.0, indicando nenhum trade-off entre **True** e **False Positive Rates**.
- **Accuracy** é a fração de casos corretamente classificados.
- **Recall**, é a fração de casos positivos classificados corretamente. Observe que esse valor é de apenas 0,593, o que significa que o classificador classifica mais de 4 dos 10 pacientes diabéticos como não diabéticos.
- **Precision** é a fração de casos negativos corretamente classificados.

No geral, esse classificador é significativamente melhor do que adivinhação aleatória, mas possui precisão limitada.

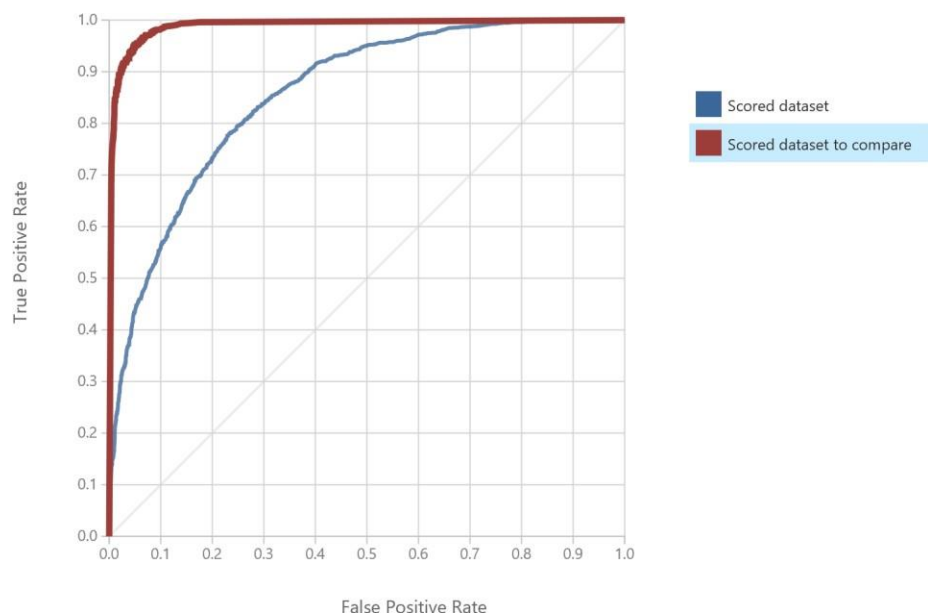
## Tente um Algoritmo Diferente

Dado desempenho questionável do modelo de classificação, um cientista de dados normalmente trabalharia em maneiras de melhorar o resultado. Uma abordagem é encontrar um algoritmo de modelo alternativo que melhor se adapte ao problema específico.

1. Adicione um módulo **Two Class Boosted Decision Tree** ao experimento.
2. Configure o módulo **Two Class Boosted Decision Tree** da seguinte maneira:
  - **Create trainer mode:** Single Parameter
  - **Maximum number of leaves per tree:** 20
  - **Minimum number of training instances required to form a leaf:** 10
  - **Learning rate:** 0.2
  - **Number of trees constructed:** 100
  - **Random number seed:** 1234
  - **Allow unknown categories:** Checked
3. Selecione os módulos **Train Model** e **Score Model**, copie e cole-os para adicionar novas cópias desses módulos ao experimento.
4. Conecte a saída do módulo **Two Class Boosted Decision Tree** à saída esquerda do módulo **Train Model** que você acabou de copiar.
5. Conecte a saída esquerda do módulo **Split Data** à entrada direita do módulo **Train Model** copiado (além da conexão existente com o módulo **Train Model** original).
6. Conecte a saída direita do módulo **Split Data** à entrada do novo módulo **Score Model** (além da conexão existente ao módulo **Score Model** original).
7. Conecte a saída do novo módulo **Score Model** à entrada do módulo **Evaluate Model**, conforme mostrado aqui:



8. Salve e execute este experimento.
9. Visualize a saída do módulo **Evaluate Model** e, na legenda do gráfico ROC, selecione **Scored dataset to compare** (que reflete a saída do módulo **Two-Class Boosted Tree**). Note que a curva para este modelo é significativamente maior e mais para a esquerda do que a curva do módulo **Two-Class Logistic Regression**.



10. Com **Scored dataset to compare** selecionado, role para baixo até as métricas de desempenho que devem se assemelhar à figura abaixo.

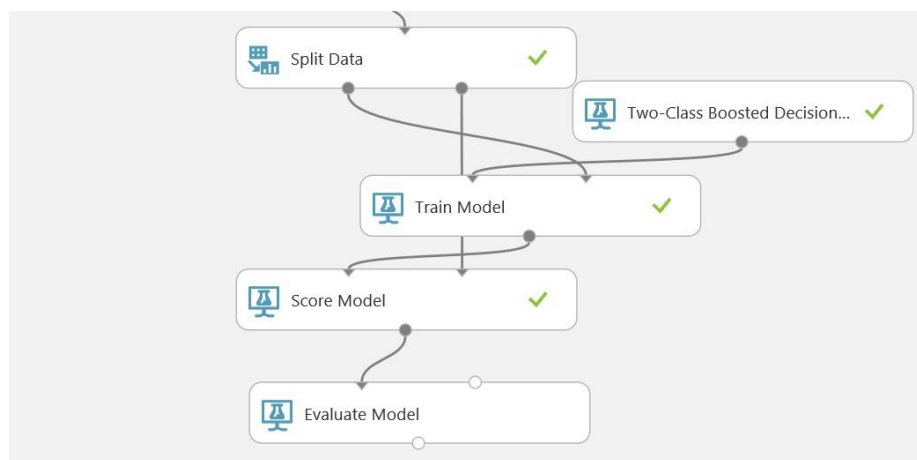
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
1399	108	0.954	0.934	0.5	0.991
False Positive	True Negative	Recall	F1 Score		
99	2894	0.928	0.931		
Positive Label	Negative Label				
1	0				

Compare esses resultados com os obtidos com o módulo **Two Class Logistic Regression**, observando o seguinte:

- **AUC** está acima de 0.99, indicando que o classificador é quase ideal.
- **ccuracy** está agora, acima de 95%
- **Recall** está quase 0.93.
- **False Positives** foram bastantes reduzidos, refletidos pela **Accuracy** de 0.934.

Esse classificador é muito mais satisfatório quando comparado ao anterior.

- Exclua o módulo **Two-Class Logistic Regression** e seus módulos **Train Model** e **Score Model** e, em seguida, altere a conexão de saída do módulo **Scored Model** para o modelo **Two-Class Boosted Tree** para a entrada esquerda do módulo **Evaluate Model** como mostrado abaixo:



- Salve e execute o experimento e visualize a saída do módulo **Evaluate Model** para verificar se apenas o modelo **Two-Class Boosted Tree** é refletido nos resultados.

## Exercício 4: Publicando e Consumindo um Serviço Web

Neste exercício, você publicará e consumirá o modelo de classificação de diabetes como um serviço Web.

### Criando uma experiência preditiva

Agora que você criou e avaliou o Machine Learning, você publicará um modelo como Web Service. A primeira etapa desse processo é criar um experimento preditivo a partir do seu experimento de treinamento e fazer as modificações necessárias antes de operá-lo como um serviço web.

- No Azure Machine Learning Studio, abra o experimento **Diabetes Analysis** que você criou nos exercícios anteriores e, se ele tiver sido modificado desde a última execução, execute-o. Observe que o ícone **Set up Web Service** está desativada até que a experiência tenha sido executada.
- No ícone **Set Up Web Service** na parte inferior da tela do Studio, clique em **Predictive Web Service [Recommended]**. Uma nova guia contendo **Predictive Experiment** será exibida.
- Mude o nome da experiência preditiva para **Diabetes Predictor**.  
A entrada de serviço web no **Predictive Experiment** é baseada no dataset de treinamento de seu experimento (**diabetes.csv**). Esses dados incluem rótulos (Diabetes), que é o que o serviço web irá prever – portanto, não parece como um parâmetro de entrada. Você resolverá esses problemas definindo uma entrada personalizada para o **Predictive Experiment**.
- Exclua o dataset **diabetes.csv** da tela de experiência.
- Adicione um módulo **Enter Data Manually** ao experimento e conecte sua saída à entrada esquerda do módulo **Join Data** (ao qual a entrada de Web Service também está conectada).
- Configure o módulo **Enter Data Manually** da seguinte forma para fornecer um cabeçalho e alguns dados de testes para definir o esquema de entrada do Web Service (Você pode copiar e colar os dados do manual **Input.txt** na pasta **Lab01**).

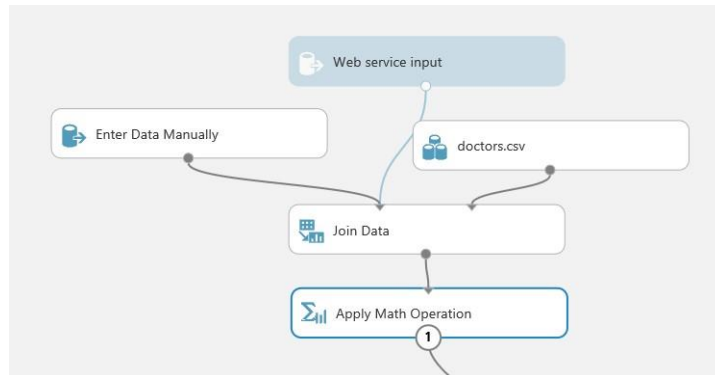
- DataFormat:** CSV

- HasHeader:** Checked

- Data:**

```
PatientID,Pregnancies,PlasmaGlucose,DiastolicBloodPressure,TricepsThickness,SerumInsulin,BMI,DiabetesPedigree,Age
1882185,9,104,51,7,24,27.36983156,1.350472047,43
1662484,6,73,61,35,24,18.74367404,1.074147566,75
1228510,4,115,50,29,243,34.69215364,0.741159926,59
```

7. Verifique se a parte superior de sua experiência está assim agora:



8. Salve e execute a experiência para verificar se não há erros como resultado da alteração de entrada.
9. Visualize a saída do módulo **Score Model**, que determina a saída do serviço Web, e verifique se ela contém apenas três linhas – uma para cada linha de teste no módulo **Enter Data Manually**.

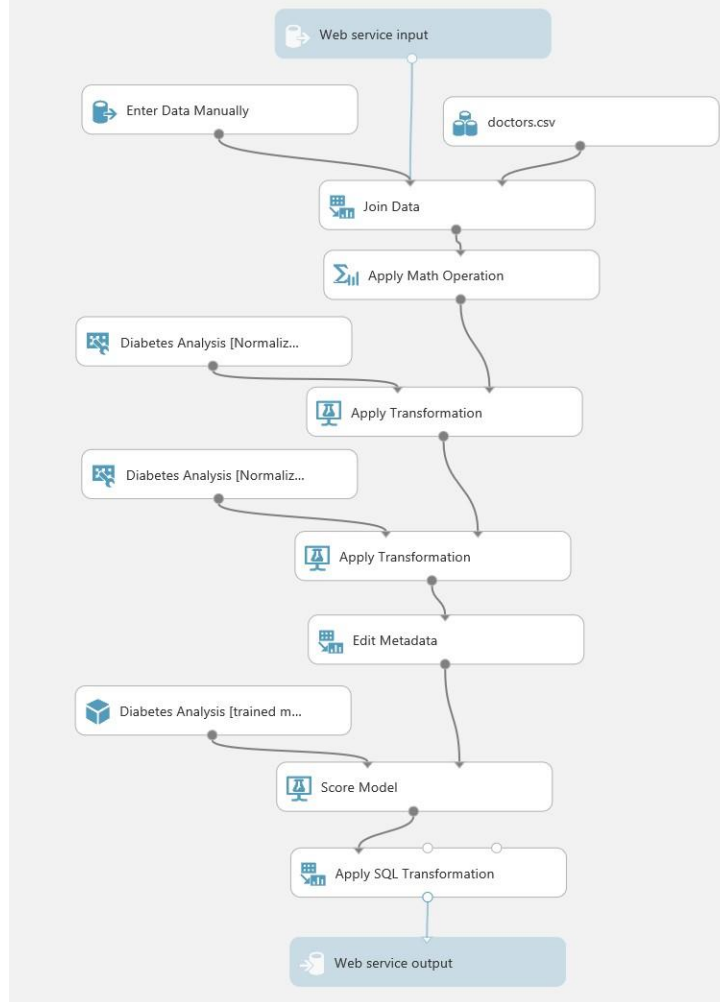
**Note:** A saída inclui todas as colunas no fluxo de dados. Agora você modificará o experiment para que apenas o ID do paciente, o medico, a previsão do diabético e a probabilidade sejam retornados.

10. Exclua a conexão entre o módulo **Score Model** e **Web Services Output**.
11. Inclua um módulo **Apply SQL Transformation** no experimento e conecte a saída do módulo **Score Model** na entrada **Table1**.
12. Configure as propriedades do módulo **ApplySQL Transformation** para atualizar o script de consulta SQL para o código a seguir (que você pode copiar e colar a partir do **DiabetesSQL.txt** na pasta **Lab01**).

```
SELECT PatientID,  
       Physician,  
       [Scored Labels] AS DiabetesPrediction,  
       [Scored Probabilities] AS Probability  
FROM t1;
```

13. Conecte a saída do módulo **Apply SQL Transformation** à entrada da saída do **Web Service**.
14. Verifique se, com um pouco de reorganização, sua experiência se parece com a figura abaixo:

# Diabetes Predictor



15. Salve e execute o experimento.

16. Visualize a saída do módulo **Apply SQL Transformation** e verifique se apenas 4 colunas são retornadas.

## Implantando Web Service

Agora você está pronto para publicar e testar o Web Service.

1. Com a experiência preditiva do **Diabetes Predictor** aberta, no menu **Deploy Web Services**, clique em **Deploy Web Service [New] Preview**. Uma página da web com um cabeçalho **Deploy Diabetes Predictor Experiment** como um Web Service.
2. Defina a configuração do Web Service da seguinte maneira:
  - **Web Service Name:** DiabetesPredictor
  - **Storage Account:** The Azure storage account associated with your workspace
  - **Pricing Plan:** The Azure Machine Learning pricing plan you created with your workspace.

3. Clique em **Deploy Box** no canto inferior esquerdo da página. Depois de algum tempo, a página de **QuickStart** do seu **Diabetes Predictor** Web Service deve aparecer.

## Consumindo o Web Service

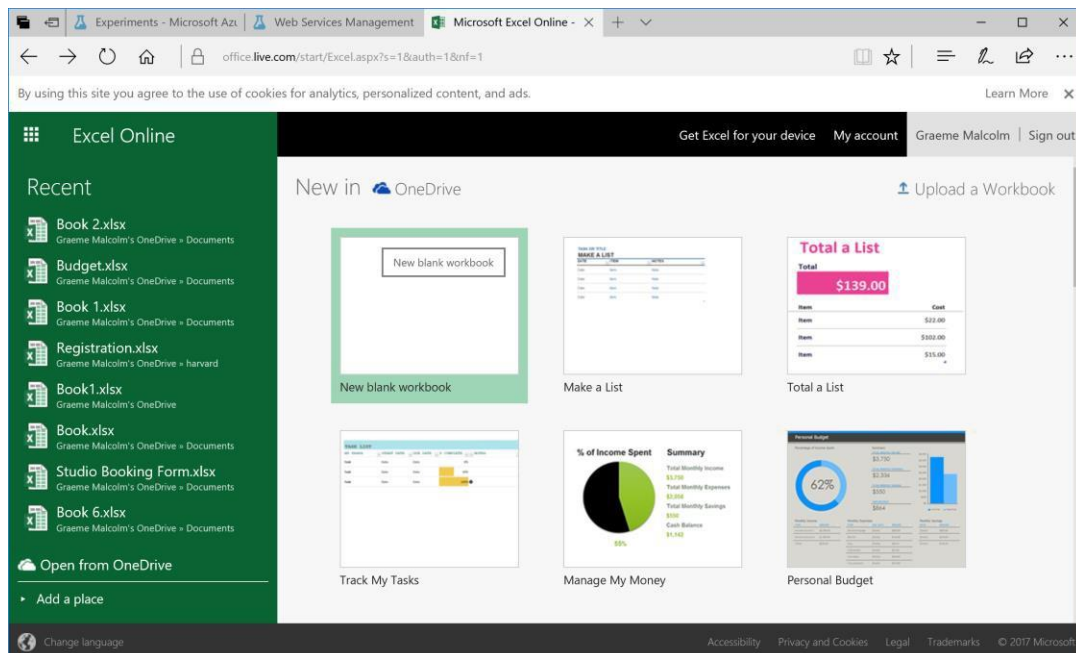
Agora que você publicou o Web Service preditivo, pode consumi-lo através dos aplicativos clientes.

**Note:** Neste exercício, você consumirá o Web Service do Excel Online, que é um serviço incluído em sua conta da Microsoft (por exemplo, uma conta do outlook.com). Se você tiver o Excel 2010 ou posterior instalado em seu computador local, poderá usá-lo em vez do Excel Online, se preferir.

1. Clique na página **Configure** para o Web Service e, ao lado de **Sample Data Enable?**, clique **Yes**. Em seguida, clique em **Save** e aguarde até que as alterações sejam salvas.

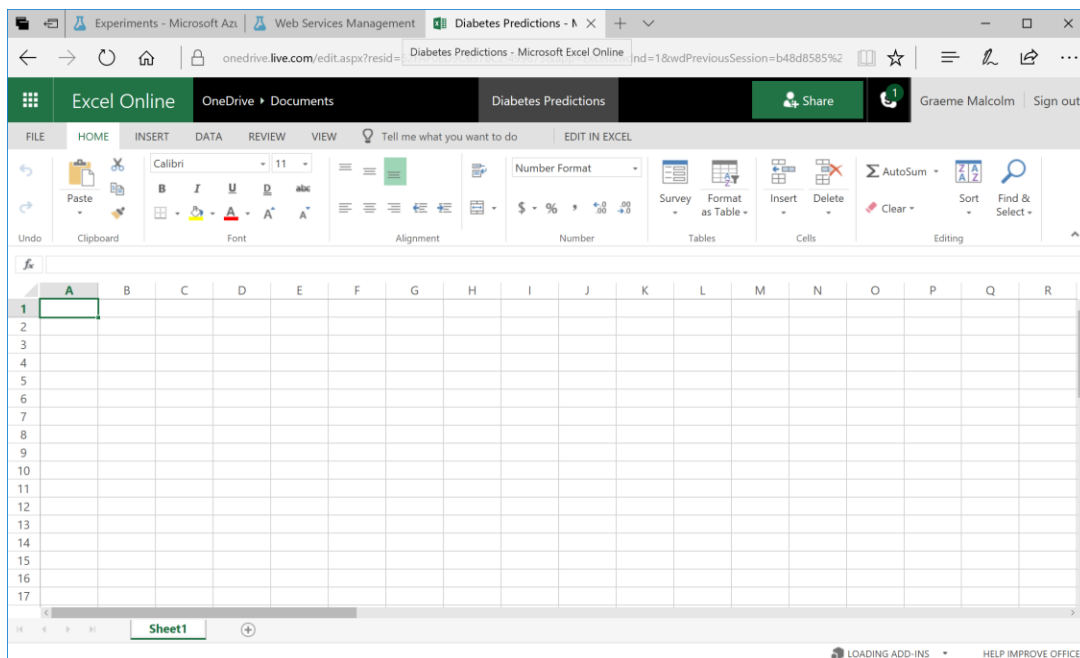
**Note:** Ao ativar os dados de amostra, os aplicativos clientes podem fazer o download de amostras definidas no módulo **Enter Data Manually** na experiência preditiva para testar o Web Service.

2. Na página **Configure** do seu Web Service, clique em **Consume** e observe que esta página lista as chaves primárias e secundárias do seu Web Service e suas URLs **Request-Response** e **Bath Requests** endpoint.
3. Abra uma nova guia no seu navegador, navegue até <https://office.live.com/start/Excel.aspx> e entre usando sua conta da Microsoft.
4. Crie uma nova pasta de trabalho em branco, como mostrado aqui:

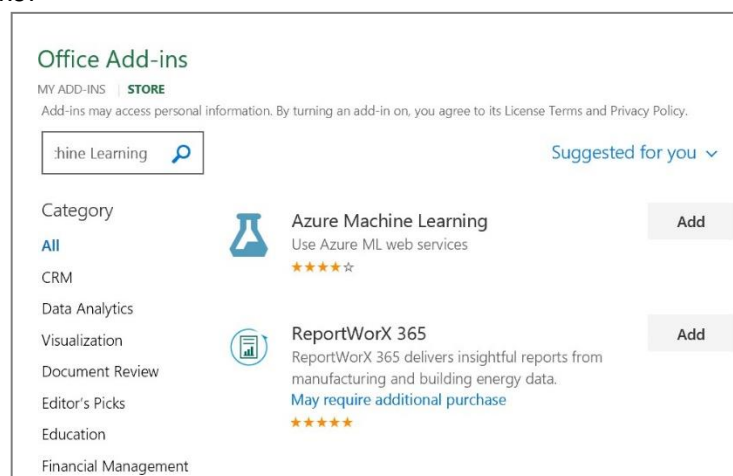


5. Na parte superior da pasta de trabalho em branco, altere o nome para **Diabetes Predictions** como mostrado aqui:





6. Na aba **Insert**, clique em **Office Add-ins**. Em seguida, em **Office Add-ins**, selecione **Store**, pesquise por **Azure Machine Learning** e adicione o **Azure Machine Learning add-in** como mostrado abaixo:



Depois que o add-in for instalado, ele aparecerá como um painel chamado **Azure Machine Learning** no lado direito da sua pasta de trabalho do Excel.

7. No painel **Azure Machine Learning**, clique em **Add Web Service**. Caixas para o URL e a chave de API do Web Service serão exibidas.
8. Na guia do navegador que contém a página **Consume** do seu **Web Service**, clique no botão **Copy** para o URL de **Request-Response**. Em seguida, retorne à guia do navegador que contém a pasta de trabalho do Excel Online e cole a URL na caixa URL.
9. Na guia do navegador que contém a página **Consume** do seu **Web Service**, clique no botão **Copy** na chave primária. Em seguida, retorne à guia do navegador que contém a pasta de trabalho do Excel Online e cole-a na caixa da chave da API.
10. Verifique se o painel do **Azure Machine Learning** na sua pasta de trabalho agora se parece com isso e clique em **Add**.

**Azure Machine Learning**

Web Services

- Titanic Survivor Predictor (Excel Add-in Sa...
- Text Sentiment Analysis (Excel Add-in Sam...

URL

<https://ussouthcentral.services.azureml.net/subscriptions/7a81077e659242f58dbc631b59a08f0c/services/67215c8153034eeb8d2b17f16a613c58/execute?api-version=2.0&format=swagger>

API key

LQYNXzMBAxR+QyJvZhPI+7/1LyWbh+B/FTDwW30BNiwaWVy61h2JLs4rXAnY4rUF7AJIP/0REpFfCof19YmdQ==

Cancel Add

☐ Auto-predict Predict All

[Help](#) [Privacy Statement](#)

- Depois que o Web Service tiver sido adicionado, na página do **Azure Machine Learning**, clique em **1.View Schema** e observe as entradas esperadas pelo Web Service, as saídas retomadas pelo Web Service e os parâmetros globais do Web Service.
- Na planilha do Excel, selecione a coluna A1. Em seguida, no painel Azure Machine Learning, contraia a seção **1.View Schema** e, na seção **2.Predict**, clique em **Use Sample Data**. Isto insere valores de entrada de amostra na planilha.
- Com as células que contêm os dados de entrada de amostra (células A1 e I4) selecionadas, no painel Azure Machine Learning, clique no botão pra selecionar o intervalo de entrada e confirme se ele é **'Sheet1'!A1:I4**.
- Verifique se a Caixa **My Data has headers** está marcada.
- Na Caixa **Output**, digite J1 e verifique se a caixa **Include headers** está marcada.
- Clique no botão **Predict** e, após alguns segundos, visualize as previsões nas células J1:M4, conforme mostrado abaixo:

The screenshot shows the Excel Online interface with a data table and the Azure Machine Learning Diabetes Predictor sidebar.

	D	E	F	G	H	I	J	K	L	M	N
1	Diastoli	Triceps	Serum	BMI	Diabetes	Age	Patient	Physician	Diabetes	Probab	
2	04	51	7	24	27.36983	1.350472	43	1882185	Carolos Lam	1	0.999691
3	73	61	35	24	18.74367	1.074148	75	1662484	Rajab Sham	0	2.78E-05
4	15	50	29	243	34.69215	0.74116	59	1228510	Jilbran Qurat	1	0.999995

The sidebar on the right is titled "Azure Machine Learning" and "Diabetes Predictor". It has two sections: "1. VIEW SCHEMA" and "2. PREDICT". Under "2. PREDICT", there are input and output fields. The input field is labeled "Input: input1" and contains "Sheet1!A1:I4". There is a checkbox "My data has headers" which is checked. Below it is a button "Use sample data". The output field is labeled "Output: output1" and contains "Sheet1!J1". There is a checkbox "Include headers" which is checked. At the bottom, there is a red warning message: "Predicting will override existing values. This can't be undone." and a "Got it!" button.

17. Tente alterar algumas das variáveis de entrada e prever as classificações de diabetes. Você pode adicionar várias linhas ao intervalo de entrada e tentar várias combinações.

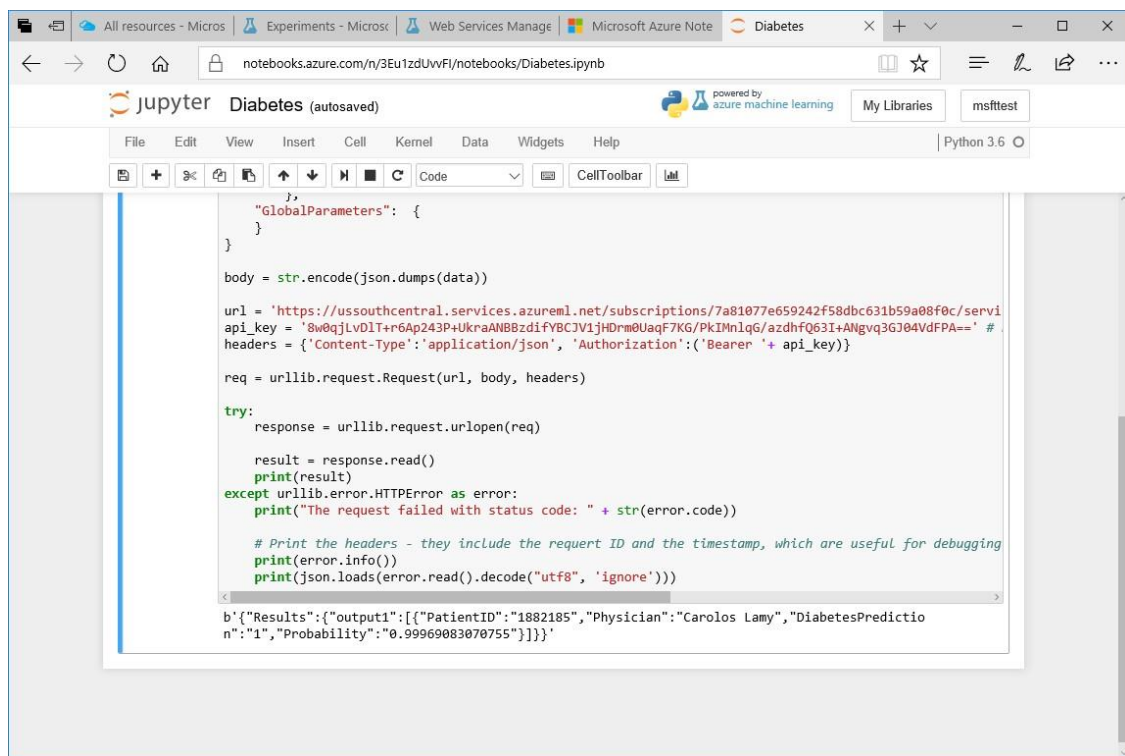
## Crie um Custom Client

Seu Azure Machine Learning Web Service está disponível para qualquer aplicativo cliente compatível com REST via HTTP. Assim, além de usar o suplemento do Excel, você pode implementar facilmente um aplicativo cliente personalizado para consumir seu Web Service.

1. Volte para o guia do navegador que contém o site de gerenciamento do Azure Machine Learning Web Services.
2. Na página **Consume**, nas chaves primárias e secundárias do seu Web Service e suas URLs **Request-Response** e **Bath-Requests** endpoints. Observe que o Azure Machine Learning gerou o código C#, Python e R que você pode usar para chamar seu Web Service.
3. Abra uma nova guia no seu navegador e navegue até <https://notebooks.azure.com>. Este é um serviço gratuito do Azure que fornece Jupyter Notebooks independentemente do seu Azure Machine Learning Workspace.
4. Entre usando sua conta da Microsoft. Se esta for a primeira vez que se inscreve no Azure Notebooks, talvez seja necessário conceder algumas permissões ao aplicativo.
5. Depois de iniciar sessão, clique em **Libraries**. Em seguida, clique em **New Library**.
6. Crie uma nova biblioteca com sua escolha de nome, ID exclusiva e descrição. Se desejar, você pode optar por tornar a biblioteca pública (para permitir que outros usuários visualizem seus blocos de anotações).
7. Na sua nova biblioteca, clique em **New** para criar um novo notebook. Nomeie o novo notebook como **Diabetes.ipynb** e escolha **Python 3.6** como a linguagem.
8. Abra o notebook **Diabetes.ipynb** e observe que a primeira célula está vazia, aguardando seu código.

9. Volte para a guia **Web Services Management**, na qual a página **Consume** para o seu Web Service, deve estar visível e visualize o código Python 3.x para o **Request-Response** endpoint de seu Web Service.
10. Clique no botão **Copy** para copiar o código para a área de transferência e, em seguida, volte para o notebook **Diabetes.ipynb** e cole o código na primeira célula.
11. No código que você acabou de colar, na definição JSON do request body, localize a seguinte linha de código:  

```
api_key = 'abc123' # Replace this with the API key for the web service
```
12. Retorne o notebook **Diabetes.ipynb** e, na linha de código acima, substitua abc123 com a chave primária que você copiou.
13. No menu **Cell**, clique em **Run All**, e exiba a saída JSON retornada pelo Web Service, conforme mostrado aqui:



```
},
  "GlobalParameters": {
  }
}

body = str.encode(json.dumps(data))

url = 'https://ussouthcentral.services.azureml.net/subscriptions/7a81077e659242f58dbc631b59a08f0c/servi
api_key = '8w0qjLvD1t+r6Ap243P+UknaANBBzdiFYBC3V1jHDrmm0UaqF7KG/PKIMnlqG/azdhfQ63I+ANGvq3GJ04VdFPA==' #
headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer ' + api_key)}

req = urllib.request.Request(url, body, headers)

try:
    response = urllib.request.urlopen(req)
    result = response.read()
    print(result)
except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

# Print the headers - they include the request ID and the timestamp, which are useful for debugging
print(error.info())
print(json.loads(error.read().decode("utf8", 'ignore'))))

b'{"Results":{"output1":[{"PatientID":"1882185","Physician":"Carolos Lamy","DiabetesPredictio
n":"1","Probability":"0.99969083070755"}]}}'
```

## Summary

Neste laboratório, você usou o Azure Machine Learning para treinar e avaliar um modelo de classificação. Isso exigia que você explorasse e preparasse os dados para modelagem por meio da Engenharia e normalização de recursos, dividindo os dados em datasets de treinamento e teste e avaliando métricas de desempenho para classificação. Em seguida, você publicou seu modelo preditivo como um Web Service e o consumiu no Excel e de um aplicativo cliente personalizado.