**Lab for Chapter 4:**

**Decision Trees**

In this lab you will explore the use of tree-based machine learning models. Tree-based models are powerful and widely used in machine learning. Specifically, in this lab you will investigate using ADA-boosted tree models for classification.

**What You’ll Need**

To complete this lab, you will need the following:

An Azure ML account

A web browser and Internet connection

The lab files for this lab

**Note**: To set up the required environment for the lab, follow the instructions in the [Setup Guide f](https://aka.ms/edx-dat203.2x-setup)or this course.

**Classification with Ada-Boosted Tree Models**

Ada-boosted tree models are a powerful ensemble machine learning model. Ada-boosted tree models can be used for classification or regression. In this lab, you will perform classification of the diabetes patients using a two-class boosted tree model.

For this lab, you can copy clean starting experiments to your Azure ML workspace from the Cortana Intelligence Gallery using the links for your preferred programming language below:

**R**: <https://aka.ms/edx-dat203.2x-lab4-class-r>

**Python**: <https://aka.ms/edx-dat203.2x-lab4-class-py>

**Create a Boosted Decision Tree Model**

1. In Azure ML Studio, open your **Optimized Diabetes Classification** experiment (or the corresponding starting experiment in the Cortana Intelligence Gallery as listed above), and save it as **Boosted Classification**.

2. Add a **Two Class Boosted Decision Tree** module to the experiment.

3. Copy the **Train Model**, **Score Model**, and **Evaluate Model** modules, and paste the copies into the experiment under the **Two Class Boosted Decision Tree** module:

4. Edit the comment of the new **Train Model** module, and change it to *Boosted Tree*.

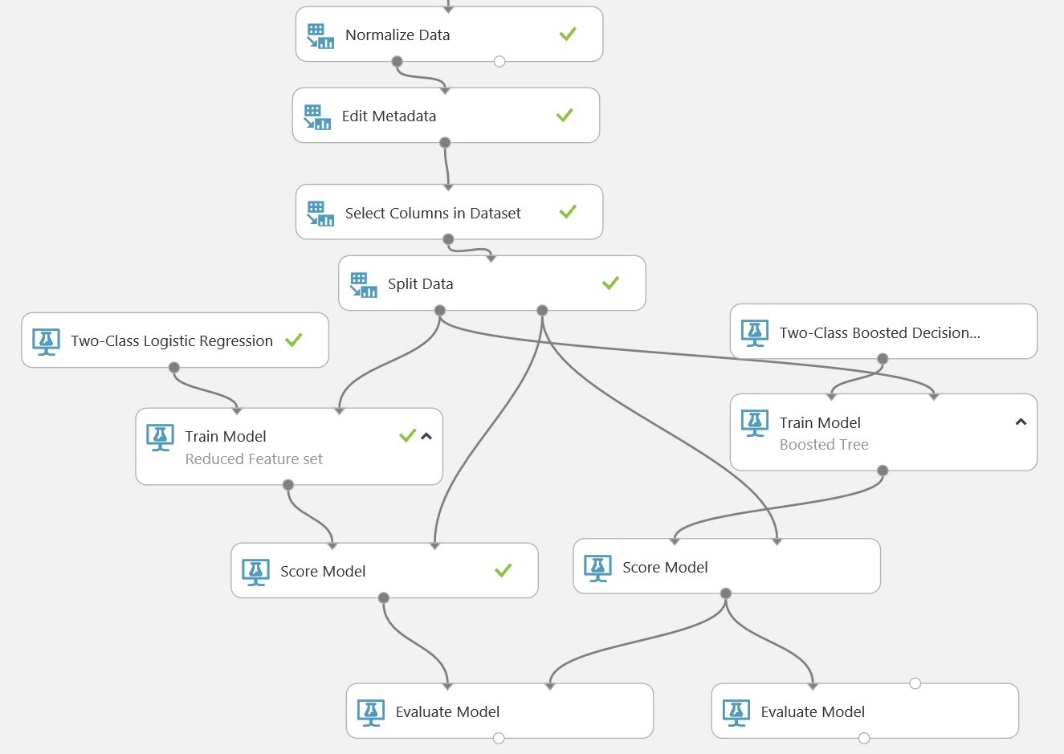
5. Connect the output of the **Two Class Boosted Decision Tree** module to the **Untrained Model** (left) input of the new *Boosted Tree* **Train Model** module. Then connect the left output of the **Split Data** module to the **Dataset** (right) input of the new *Boosted Tree* **Train Model** module.

6. Connect the output of the new *Boosted Tree* **Train Model** module to the **Trained Model** (left) input of the new **Score Model** module. Then connect the right output of the **Split Data** module to the **Dataset** (right) input of the new **Score Model** module.

7. Connect the output of the new **Score Model** module to the **Scored dataset to compare** (right) input of the original **Evaluate Model** module (the left input of which the **Scored Model** module for the original linear regression model is already connected).

8. Connect the output of the new **Score Model** module to the **Scored dataset** (left) input of the

new **Evaluate Model** module. Then ensure that the bottom portion of your experiment looks like this:



**Compare Model Performance**

1. Save and run the experiment.

2. When your experiment has finished running, visualize the output of the original **Evaluate Model** module and examine the ROC curve. The **Scored dataset** (Blue) curve represents the original Linear Regression model, and the **Scored dataset to compare** (Red) curve represents the two- class boosted decision tree model. The higher and further to the left the curve, the better the performance of the model.

3. Scroll down further in the visualization of the down and examine the **Accuracy** and **AUC** model performance metrics, which indicate the accuracy and area under the curve of the original linear regression model.

4. Visualize the output of the new **Evaluate Model** module and examine the **Accuracy** and **AUC** model performance metrics, which indicate the accuracy and area under the curve of the new two-class boosted decision tree model. Compare this with the same metrics for the linear regression model – the model with the higher metrics is performing more accurately.