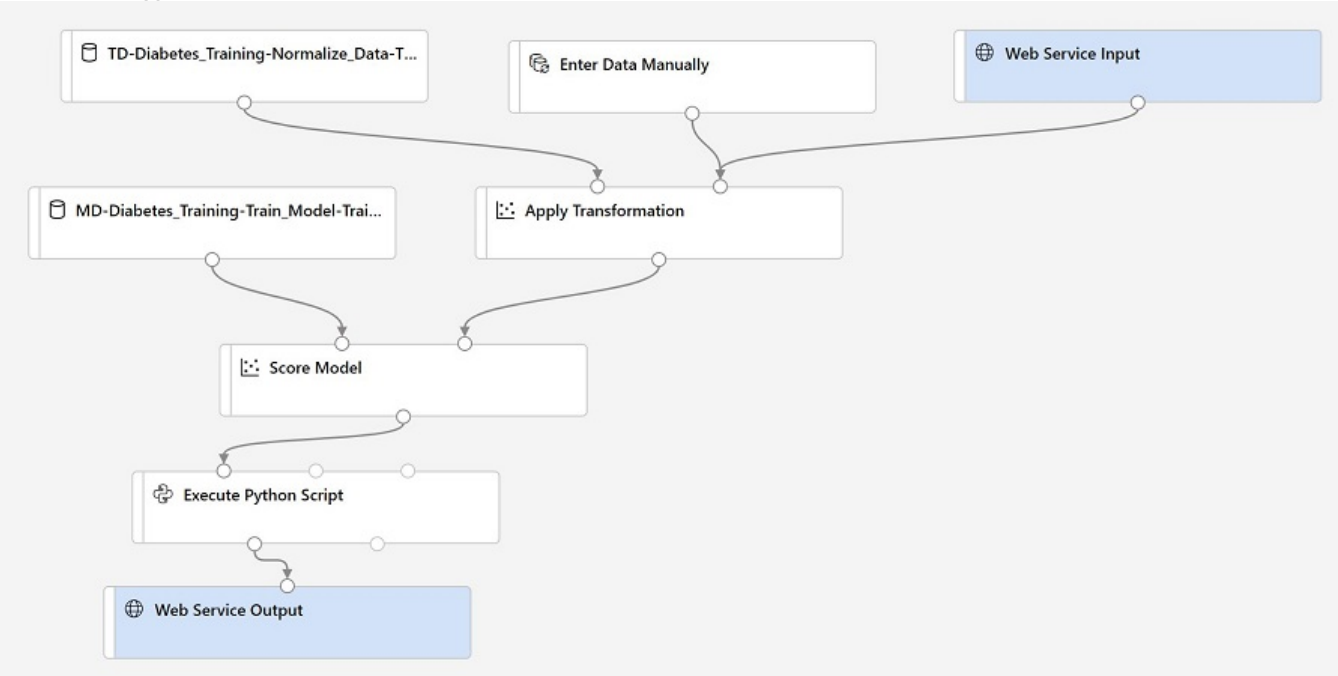
Highlight Note

Inference Pipelines

Having trained a model using a *training pipeline*, you can use it to create an *inference pipeline* for either real-time or batch prediction.



An inference pipeline encapsulates the steps required to use the trained model in a web service that predicts labels for new data. It differs from the training pipeline in the following respects:

- A web service input defining an interface for new data to be scored is added to the beginning of real-time inference pipelines. By default, this is based on the schema of the training dataset.
- Steps that rely on statistics from the training data (such as feature normalization or categorical encoding) are encapsulated in transformation datasets that are applied to new data.
- The trained model is encapsulated in a dataset, removing the algorithm and model training modules.
- A Web service output containing the scored results is added at the end of real-time inferencing pipelines to define the output returned to applications consuming the service.

Modifying the Inference Pipeline

Before deploying an inference pipeline as a web service, you may want to make some changes to it. For example:

- For supervised learning models, consider replacing the training dataset at the beginning of the pipeline with an alternative data definition that does not include the label column. This has the effect of removing the label column from the web service input schema, which is more intuitive for client application developers (who would otherwise need to submit a value for the label that they want the model to predict).
- If you choose to remove the label column from the input schema, ensure it is not explicitly referenced in any other modules in the pipeline, as this will cause a runtime exception.

- Remove any modules that are not required - for example, if the training pipeline includes an **Evaluate Model** module, it will be included by default in the inference pipeline, even though it is not used.
- Consider filtering the output columns. The **Score Model** module returns its input data as well as the scored label and probability columns, so by default the web service will return all of these to the client application. The web service may be more efficient if you filter these to include only the required output, such as a row identifier and the scored label and probability.
- Consider adding *parameters*, which can be passed by calling applications to add flexibility to the pipeline. Typically, parameters are used to enable a choice of data sources to be used in the pipeline.