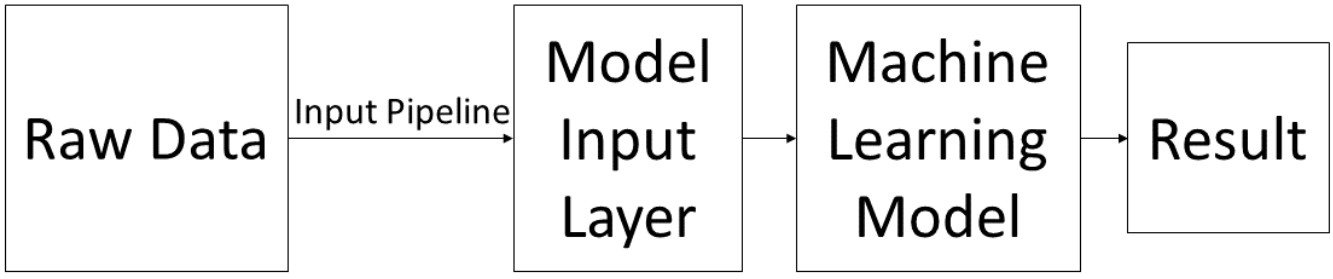
Introduction

In this section of the course you will be using TensorFlow to load and store data. Specifically, you will use protocol buffers to efficiently store large amounts of data and then use TensorFlow's `tf.data` API to load the data. We will be using a dataset comprised of college student data to test your code.

A. Input pipeline

Machine learning models are almost always trained on extremely large datasets. Depending on the application, a training dataset can have hundreds of thousands, or even millions, of training examples. The sheer size of these datasets makes it incredibly important to store the data in an efficient manner.

The process in which data is loaded from files and fed into a machine learning model is known as the *input pipeline*. Since the input pipeline handles a large amount of data for machine learning projects, we need it to be as efficient as possible.



Input pipeline for a machine learning model.

A flexible and efficient format for storing large amounts of data is Google's protocol buffer. The protocol buffer is similar to JSON and XML (another feature-based data format), but uses less space and is faster to process. When used with TensorFlow, protocol buffers make the input pipeline for large datasets much more streamlined.

B. Loading data

One of the main reasons why protocol buffers work so well with TensorFlow is the ease with which they can be loaded as input into a machine learning model. Since TensorFlow is also a Google-developed product, the TensorFlow API contains functions that make it simple to quickly load data from a protocol buffer. This is why most of the official TensorFlow open-source models use protocol buffers for storing and loading data.

In particular, the `tf.data` API provides us with all the tools necessary to create an efficient input pipeline. While it works very well with protocol buffers (which are used to store feature-based data), it can also be used effectively with NumPy data. In the following chapters, you'll create input pipelines from both protocol buffers and NumPy data.