1. Why would you want to use the Data API?

APIs are built as an attempt to free the data from the clutches of these web interfaces, but come with their own set of challenges. CRM companies these days all seem to promise that they have an “easy to use API” and that their customers can use this API to fetch and process the data collected by the CRM

1. What are the benefits of splitting a large dataset into multiple files?

Splitting a data set creates more files, which occupy more disk space and forces more I/O operations. I/O and disk access is often the most expensive part of your SAS processing, performance-wise. But if the boss asks for broken-up data sets, you might as well spend the least possible effort on the task.

3. During training, how can you tell that your input pipeline is the bottleneck? What can you do

to fix it?

A CPU bottleneck occurs when the GPU resource is under utilized as a result of one, or more of the CPUs, having reached maximum utilization. In this situation, the GPU will be partially idle while it waits for the CPU to pass in training data. This is an undesired state. Being that the GPU is, typically, the most expensive resource in the system, your goal should always be to maximize its utilization

1. Can you save any binary data to a TF Record file, or only serialized protocol buffers?

The TFRecord format is a simple format for storing a sequence of binary records. Protocol buffers are a cross-platform, cross-language library for efficient serialization of structured data. Protocol messages are defined by .proto files, these are often the easiest way to understand a message type.

5. Why would you go through the hassle of converting all your data to the Example protobuf

format? Why not use your own protobuf definition?

Google also provides a protocol buffer compiler to generate source code from .proto file in Java programming language and an API to read and write messages on protobuf object. You don't need to bother about any encoding, decoding detail, all you have to specify is your data-structure, in a Java, like format.

6. When using TF Records, when would you want to activate compression? Why not do it

systematically?

A TFRecord is when a sequence of such records serializes to binary. The binary format takes less memory for storage in comparison to any other data formats. That’s what I’m going to do now. I will convert all the records of a dataset to TFRecords which can be serialized into binary and can be written in a file