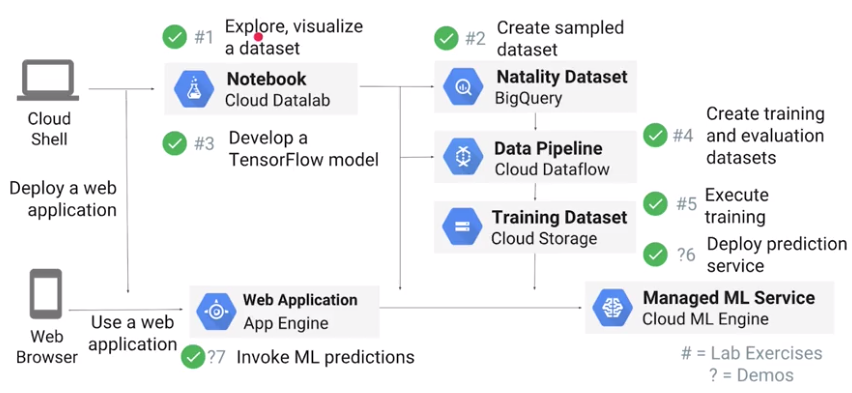
# End to End Machine Learning

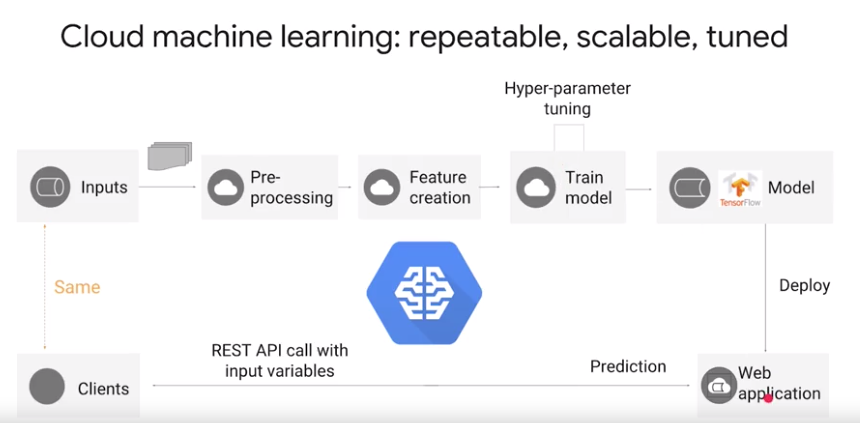


Cloud ML Engine is orthogonal to this hierarchy. Regardless of which abstraction level you're writing your code at, CMLE gives you a managed service for training and deploying TensorFlow models.

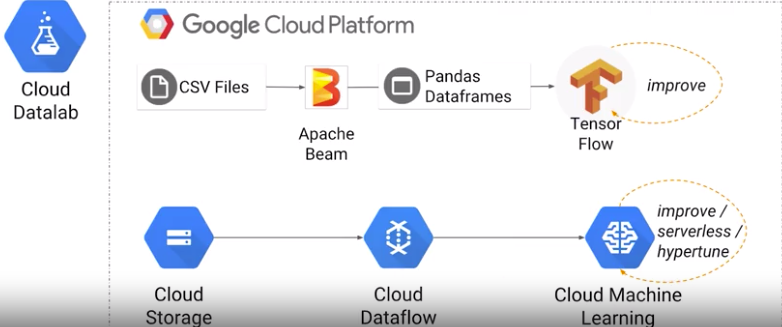
Once your data sets get larger, your data will not fit into memory and you need a more sophisticated performant ML framework. This is where TensorFlow comes in. You can use TensorFlow estimators not just for deep learning but also for things like Boosted Regression Trees.

Train the model on as much data as possible. As your data size increases, batching and distribution become extremely important. Things like gradient descent optimizations are not embarrassingly parallel. You will need parameter servers that form a shared memory that's updated during each epoch. Scaling out is the answer not scaling up.

So, why use TensorFlow? Because it can work with big data, it can capture many types of feature transformations, and it has implementations of many kinds of model architectures.

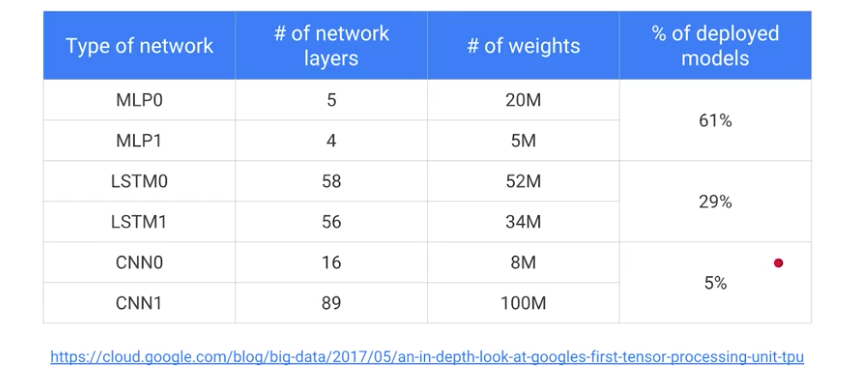


You can't pass in the raw input variables to the trained model because a trained model expects scaled transformed inputs. (How the feature Scaling is missed out in prediction, which causes the training serving queue.)

we will start in Cloud Datalab, a Jupiter notebook environment. We'll run sequel statements to aggregate the data in BigQuery and pull the data into a Pandas DataFrame. We can then do exploration and feature selection and pre-processing using Pandas, and we will write this Pandas DataFrame out to CSV. Then, we'll start experimenting with TensorFlow. Once we have a working TensorFlow model on the smaller dataset, we can then scale it out to Google Cloud Platform. We'll do this using serverless technologies.  
  

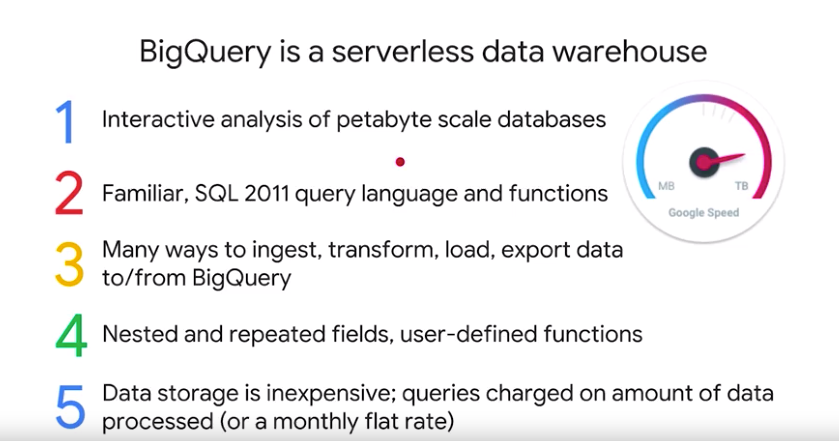
We'll do a pre-processing in Cloud Dataflow so that the pre-processing can be scaled out to many machines. We'll create sharded CSV files on cloud storage. The TensorFlow model will remain the same, but we'll train the model on Cloud ML Engine so that we get distributed training. We'll do hyper-parameter tuning and deployment also on ML Engine. So Dataflow and ML Engine, these are both serverless technologies.

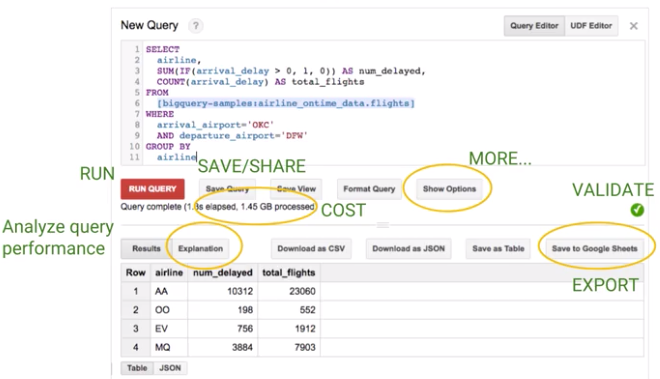
How google works on structured data?



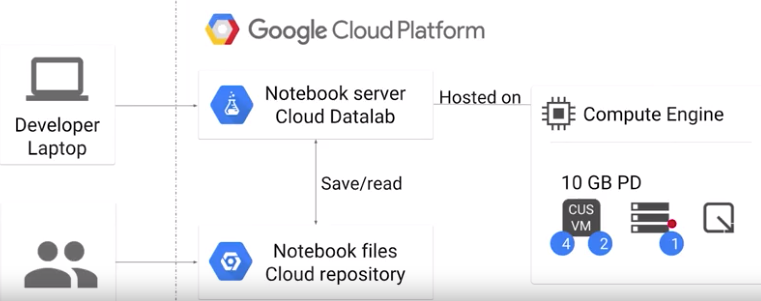
MLP is multilayer perceptron, your traditional feedforward fully connected neural network with four or five layers, and that's what you tend to use for structured data. Nearly two thirds of our models are MLPs. LSTM, long short-term memory models, are what you tend to use on text and time series models. That's 29% of all of our models. CNNs, convolutional neural networks, these are the models you tend to use primarily for image models. Although you can also successfully use them for tasks like text classification. CNNs are just five percent of models. This explains why we have focused so much on structured data models.

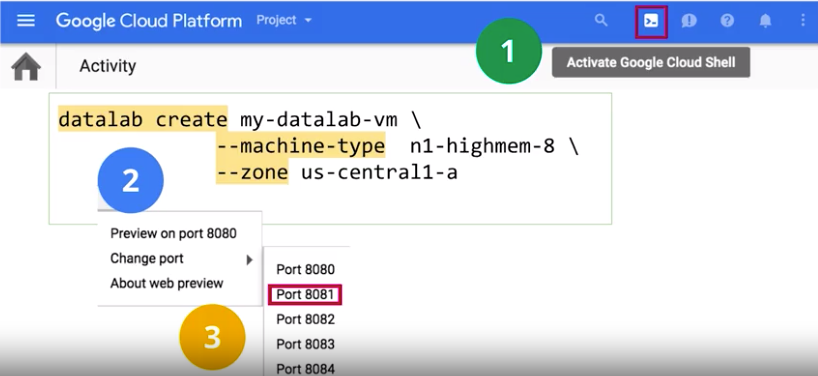
BigQuery:

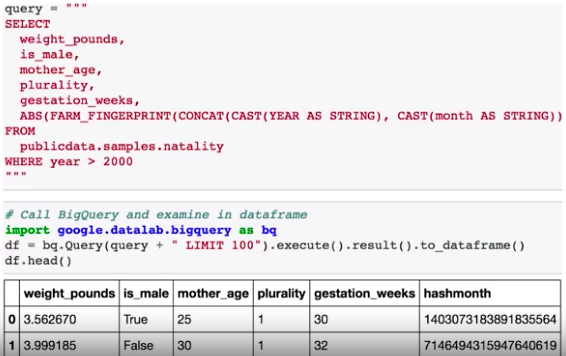


****

**Datalab:**

****

****

****