I have decided to examine different cloud service providers.

{% include figure.liquid loading="eager" path="assets/img/cloud/happy-cloud.jpg" title="example image" class="img-fluid rounded z-depth-1" %}
I will first present my thoughts about using the [Azure Texting Analytics API] (https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/), in particular, the Azure cloud API for sentiment analysis. Then we will dive into the [AutoML Tables] (https://cloud.google.com/automl-tables/) product on GCP.

Please keep in mind that cloud services are likely to improve over time. My comments would only be relevant at the time of writing (May 2019).

Sentiment Analysis on Azure

I would like to start by saying I understand the basic concepts in natural language processing such as n-gram, a bag of words, one-hot encoding, etc, but I am by no means an expert nor experienced in NLP. I was very excited that Cloud services start to have text analytics APIs so that even novices like me can perform sentiment analysis without having to find my own training data and train my own models.

Feel free to check out my notebook. I have called the sentiment analysis REST API to perform sentiment analysis.

Findings

- At the current release of the sentiment analysis API, each document analyzed would be truncated to 100 tokens. This means that if one wanted to analyze a long document, one would have to break the text up into different chunks of less than 100 tokens. This is not ideal but this is the current limitation.
- It is great that the cheapest subscription can analyze 5,000 documents for free each month. This is enough for anyone interested in testing this out to have a good feel of how the API works.

Results

I have only made 1 API call to analyze the sentiment of my introduction, namely the text "Charles is a Data Scientist working in Toronto. He is also a big fan of Pokemon". I was hoping this introduction statement gives a very exciting feeling for the reader, but, according to the sentiment analysis API, the introduction has a score of 0.5, with 0 being negative sentiment and 1 being a positive sentiment. I should rewrite this introduction sometime in the future.

AutoML Tables on GCP

AutoML Tables is a product offered on GCP that can automate your model training process. It supports relational data and can automate the process of the model selection process for supervised learning. This is often the most tedious part of a data scientist's job so I was very excited to test it out.

Set up

I went on Kaggle and found this compeition. I downloaded the data and ensured the dataset can be used in AutoML Tables. Currently, AutoML tables requires the training data to have at least 1,000 records.

I uploaded the datasets (both training and test) onto Google Cloud Storage within the same bucket and have AutoML Tables read the datasets from there. Alternatively, AutoML Tables can also use data from BigQuery.

Once a project is created in AutoML Tables, I selected the training data from Storage. In the Schema tab, it automatically detects which fields are nullable and asks you for the target for supervised learning.

{% include figure.liquid loading="eager" path="assets/img/cloud/dataselection.png" title="example image" class="img-fluid rounded z-depth-1" %}

Note that I did not perform any data processing and feature engineering as I simply wanted to test the AutoML Tables functionalities.

Immediately in the Analyse tab, it creates quick summary statistics about your training set. This is very helpful in helping us determine data quality and summary statistics rather quickly.

{% include figure.liquid loading="eager" path="assets/img/cloud/stats.png" title="example image" class="img-fluid rounded z-depth-1" %}

After setting training parameters (such as training budget so the service would not become too expensive) the training began. It is to my surprise that training was quite slow (roughly 1 hour). This can be because of my resource settings. Nonetheless, I received an email informing me the training was complete and we were able to see this screen.

{% include figure.liquid loading="eager" path="assets/img/cloud/training.png" title="example image" class="img-fluid rounded z-depth-1" %}

It is great that AutoML Tables provides different fitting metrics and variables importance for the training. However, this is still a long way to model transparency and interpretability. I was unable to retrieve the model name and parameters of the trained model, nor can I do anything with the model asides from performing predictions.

After training, I have generated results using the provided test set from Kaggle. I have noticed there are few records that the model was unable to predict the results for. They returned parsing errors for some input fields. This seems like a data quality issue but I simply replaced the error predicted values with the mean of all other predicted values. Feel free to check out my notebook for how I processed the data.

I then uploaded the results on Kaggle to see how accurate the predictions are. I am placed in 2908 out of 4542 participants. This is at the 36th percentile which is decent considering I did not do much work besides uploading data onto GCP.

{% include figure.liquid loading="eager" path="assets/img/cloud/results.png" title="example image" class="img-fluid rounded z-depth-1" %}

Potential Future Work

Performing feature engineering before model fitting with AutoML Tables would enhance model performance. Also, it would be an awesome idea to test out other cloud providers for their AutoML capabilities to test their performance.