Big Data and Intelligent Analytics
Spring 2021

INSTRUCTOR: Sri Krishnamurthy
analyticsneu@gmail.com

Assignment 4

The goal of the first part of this assignment is to create API that

1. Anonmyizes the data through:
    - Masking
    - Anonymization

Then, building upon the Infrastructure for login and server less functions using Cognito in Assignment 1, integrate the APIs so that

1. Only authenticated users can call these APIs

2. Use Amazon Step functions and Lamda functions to make it server less where feasible (This is a design decision; You may host servers and then call those APIs or call readily available APIs like Amazon Comprehend through lambda functions)

Refer:

1. Complete and submit the following tutorials: https://aws.amazon.com/blogs/machine-learning/detecting-and-redacting-pii-using-amazon-comprehend/

(Links to an external site.)

2. Presidio: https://github.com/microsoft/presidio

Implementation:

Create three APIs:

API 1: Access

This API should retrieve the EDGAR filings data from the S3 bucket

API 2: Named entity recognition

This API should take a link to a file on S3 and:

- Call Amazon Comprehend OR Google OR Presidio or any tool of your choice to find entities. (You can define the list of entities or use the default ones like Name, SSN, Date etc.)
- Store these on S3

API 3: Implement masking, and anonymization functions.

Note: You have to define the API so as to indicate which entities need to be masked, which needs to be anonymized. You also need to get the location of the file/files as input and output the files back to S3. You can choose a method of your choice!

Part 2:

In this part of the assignment, you will build upon the pre-processed (anomymized/masked data) and build a sentiment analysis model that could take the location of the anonymized file as an input and generate sentiments for each sentence.

To build this service, you need a Sentiment analysis model that has been trained on “labeled”, “Edgar” datasets. Note that you need to have labeled training data which means someone has to label the statements. We will use the IMDB dataset as a proxy and build a sentiment analyzer that can be tested on the anonymized datasets you prepared in the prior assignment

Preparation:

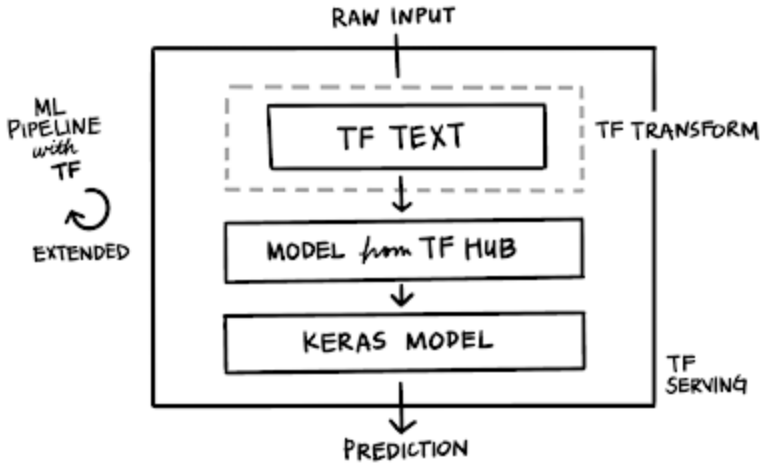
WHy TFX?

Read https://blog.tensorflow.org/2020/09/brief-history-of-tensorflow-extended-tfx.html#TFX for the history of TFX and MLOps
Watch for an overview:
https://www.youtube.com/watch?v=YeuvR6m6ACQ&list=PLQY2H8rRoyvxR15n04JiW0ezF5HQRs_8F&index=1

Goal: To deploy a sentiment analysis model to create a Model-as-a-service for anonymized data

Step 1:Train TensorFlow models using TensorFlow Extended (TFX)

Replicate the architecture to train the model for the anonymized data using BERT and this architecture that leverages **TensorFlow Hub**, **Tensorflow Transform, TensorFlow Data Validation and Tensorflow Text and Tensorflow Serving**



The pipeline takes advantage of the broad TensorFlow Ecosystem, including:

- **Loading the IMDB dataset via TensorFlow Datasets**
- **Loading a pre-trained model via tf.hub**
- **Manipulating the raw input data with tf.text**
- **Building a simple model architecture with Keras**
- **Composing the model pipeline with TensorFlow Extended, e.g. TensorFlow Transform, TensorFlow Data Validation and then consuming the tf.Keras model with the latest Trainer component from TFX**

Ref:
https://blog.tensorflow.org/2020/03/part-1-fast-scalable-and-accurate-nlp-tensorflow-deploying-bert.html
https://blog.tensorflow.org/2020/06/part-2-fast-scalable-and-accurate-nlp.html
Sample Code:
https://colab.research.google.com/github/tensorflow/workshops/blob/master/blog/TFX_Pipeline_for_Bert_Preprocessing.ipynb#scrollTo=WWni3fVVafDa
Note: Use AlBert instead of BERT (https://tfhub.dev/google/albert_base/3)
Also, Note, you will be implementing sentiment analysis.. So you will have to change the dataset. Use the IMDB dataset as a a proxy. See (https://www.tensorflow.org/tutorials/keras/text_classification_with_hub) for details.

Step 2: Serve the model as a REST API

Use the TENSORFLOW TFX RESTFUL API to serve the model
(https://www.tensorflow.org/tfx/serving/api_rest)
See for sample code:
https://www.tensorflow.org/tfx/tutorials/serving/rest_simple
OR
https://towardsdatascience.com/serving-image-based-deep-learning-models-with-tensorflow-servings-restful-api-d365c16a7dc4

OR

USE FAST API to serve the model
See for sample code:
https://medium.com/python-data/how-to-deploy-tensorflow-2-0-models-as-an-api-service-with-fastapi-docker-128b177e81f3
OR
https://testdriven.io/blog/fastapi-streamlit/

Step 3:Dockerize the API service.

For sample code on how to Dockerize API:

See https://www.tensorflow.org/tfx/serving/docker

OR

2

https://medium.com/python-data/how-to-deploy-tensorflow-2-0-models-as-an-api-service-with-fastapi-docker-128b177e81f3

Step 4: Build a Reference App in Streamlit to test the API

Input: Link to an anonymized/deanonymized file in Amazon S3

Output: Sentiment scores.

Sample Code:

See https://testdriven.io/blog/fastapi-streamlit/ for samples

Step 5: Write unit tests & Load tests to test your api

- You will have to show test cases you have tested (using pytest)
- Load test the API with Locust

Deliverables (Due April 16th)

1. A 2-5 page report in https://github.com/googlecodelabs/tools format to illustrate your understanding of how various steps are integrated in the pipeline.
2. A fully functional pipeline that when run generates the output in a streamlit app.
3. You will be given 10 minutes to present your company analysis in class.