10 Academy Batch 4 - Weekly Challenge: Week 9

Data Engineering: Speech-to-text data collection with Kafka, Airflow, and Spark

Overview

Business Need

In weeks 4 & 5 you built an AI model for either Swihali or Amharic language. For both of the cases, you have experienced the lack and quality of data. Had you had a diverse and large training set, your model could have improved and your model could have transformed the lives of many in East Africa.

This week, 10 Academy is your client. Recognizing the value of large data sets for speech-to-text data sets, and seeing the opportunity that there are many text corpuses for both languages, and understanding that complex data engineering skills is valuable to your profile for employers, this week's task is simple: design and build a robust, large scale, fault tolerant, highly available Kafka cluster that can be used to post a sentence and receive an audio file.

By the end of this project you should produce a tool that can be deployed to process posting and receiving text and audio files from and into a data lake, apply transformation in a distributed manner, and load it into a warehouse in a suitable format to train a speech-to-text model.

Data

The purpose of this week's challenge is to build a data engineering pipeline that allows recording millions of Amharic and Swahili speakers reading digital texts in app and web platforms. There are a number of large text corpuses we will use, but for the purpose of testing the backend development, you can use the recently released Amharic news text classification dataset with baseline performance dataset: IsraelAbebe/An-Amharic-News-Text-classification-Dataset: An Amharic News Text classification Dataset (github.com)

You can read a brief description of the data here.

Alternative data

Ready made Amharic data collected from different sources here

Expected Outcomes

Skills:

- Create and maintain an Apache Kafka cluster
- Work with Apache Airflow and Apache Spark
- Apply Structured Streaming to process streaming data.
- Building data pipelines and orchestration workflows

Knowledge:

- Enterprise-grade data engineering - using Apache and Databricks tools

Competency Mapping

The tasks you will carry out in this week's challenge will contribute differently to the 17 competencies 10 Academy identified as essential for job preparedness in the field of data science, and Machine Learning engineering. The mapping below shows the change (lift) one can obtain through delivering the highest performance in these tasks.

MC0: Marginal contribution - causes no significant change

MC1: Minor contribution - recognized for routine performance gain

MC2: Measurable contribution - will contribute a value towards portfolio and job readiness metric

MC3: Major contribution - the best performance of these types of tasks at least three times within our training leads one to attain a job-ready level along that competency dimension.

Competency	**Value**	**Potential contributions from this week**
Business Understanding	MC3	Understanding and reasoning the business context. Thinking about suitable analysis that matches the business need. Thinking about clients and their interests.
Data Engineering	MC3	Thinking about how to store data for easy analysis, and what format to use to build responsive dashboards.

Data Understanding	MC3	Understanding the data provided and extract insight. Exploring different techniques, algorithms, statistical distributions, sampling, and visualization techniques to gain insight.
Dashboard & Visualization	MC1	Building a dashboard to explore data as well as to communicate insight. Advanced use of modules such as plotly, seaborn, matplotlib etc. to build descriptive visualizations. Reading through the modules documentation to expand your skillset.
Mathematics and Statistics	MC0	Thinking about statistical distributions, sampling, bias, overfitting, correlations.
MLOps & Continuous Delivery	MC2	Using Github for code development, thinking about feature store, planning analysis pipeline, using MLOps tools for code, data, model, artifact versioning, setting up docker containers for automated microservice deployment.
Modeling and evaluation	MC0	Comparing multiple Deep learning techniques; training and validating DL models; choosing appropriate architecture, loss function, and

		regularisers; hyperparameter tuning; choosing suitable evaluation metrics.
Python programming	MC3	Advanced object-oriented python programming. Python package building.
SQL programming	MC3	Building feature stores using SQL or NoSQL databases.
Fluency in the Scientific Method	MC1	Thinking about evidence. Generating hypothesis, testing hypothesis. Thinking about different types of errors.
Ethics	MC1	data privacy, data security, ethical use of data. The 8 principles of responsible machine learning
Statistical & Critical Thinking	MC1	Thinking about the difference between causal vs chance correlation. Giving reasonable recommendations. Thinking about uncertainties.
Software Engineering & Dev Environment	MC3	Reading articles on software project planning. Unit testing.
Impact & Lifelong learning	MC3	Learning new concepts, ideas, and skills fast, and applying them to the problem at hand.

Professional Culture & Communication	MC2	Writing a well-formatted presentation with no mistakes, formatted nicely.
Social Intelligence & Mentorship	MC2	Asking for help early, providing help for those who need it, avoiding being stuck.
Career Thinking	MC1	Working within groups in a successful way

Team

Instructors: Yabebal, Abubakar, Mahlet, Kevin

Key Dates

- **Discussion on the case** - 11:30 UTC time on Monday 06 September 2021. Use #all-weeks9-10 to ask questions.
- **Interim Submission** - 8:00 PM UTC time on Wednesday 08 September 2021.
- **Final Submission** - 8:00 PM UTC time on Saturday 11 September 2021

Leaderboard for the week

There are 100 points available for the week.

Badges

Each week, one user will be awarded one of the badges below for the best performance in the category below.

In addition to being the badge holder for that badge, each badge winner will get +20 points to the overall score.

Visualization - the quality of visualizations, understandability, skimmability, choice of visualization

Quality of code - reliability, maintainability, efficiency, commenting - in the future this will be CICD

An innovative approach to analysis -using latest algorithms, adding in research paper content and other innovative approaches

Writing and presentation - clarity of written outputs, clarity of slides, overall production value

Most supportive in the community - helping others, adding links, tutoring those struggling

The goal of this approach is to support and reward expertise in different parts of the Machine learning engineering toolbox.

Group Work Policy

This week, you are expected to complete the project with your assigned group. In the table below, your name is assigned to one of the groups we formed.

The deliverables are outlined below - reporting is to be done individually - this means that every person must submit their own work. Discussion is allowed, copying is not allowed.

Each person's code contributions must be demonstrated as branches or commits.

Roles:
The Team Lead is responsible for having team members deliver on time, on schedule, and on quality. This implies that the other team members must coordinate their tasks with her.

Group Name	Group Members
Chang	Stella K (team lead) Smegnsh (deputy team lead) daniel zelalem Yosef Alemneh Ethani Caphace

	Binyam Sisay Germain Rukundo Boris Papineau Hirwa
Benkart	Dibora (team lead) Toyin (deputy team lead) Elias Andualem Abreham Gessesse Euel Fantaye Yosef Engdawork Michael Darko Ahwireng Mubarak Sani
Reiten	Bethlehem (team lead) Harriet S (deputy lead lead) Milky Bekele Natnael Sisay Michael Tekle mizan abaynew Luel Hagos Chimdessa Tesfaye Hordofa
Choquet-Bruhat	Khairat A (team lead) Eyerusalem (deputy team lead) Zelalem Getahun Bereket Kibru Natnael Teshome Amon Kimutai Desmond Onam christian ZANOU
Hu	Kate N (team lead) Saba (deputy team lead) Rachel (deputy team lead) Same Michael Nebiyu Samuel blaise papa Haftom Tekleweyni jakinda oluoch
Morawetz	Bezawitalem (team lead) Dorothy (deputy team lead) Azaria Tamrat Fumbani Banda Behigu Gizachew Maelaf Estiphanos D mukuzi

Late Submission Policy

Our goal is to prepare successful learners for the work and submitting late when given enough notice, shouldn't be necessary.

For interim submissions, those submitted 1-6 hours late will receive a maximum of 50% of the total possible grade. Those submitted >6 hours late may receive feedback, but will not receive a grade.

For final submissions, those submitted 1-24 hours late, will receive a maximum of 50% of the total possible grade. Those submitted >24 hours late may receive feedback, but will not receive a grade.

Instructions

The fundamental tasks in this week's challenge are the following

1. Work in a group to design a data capture pipeline.
2. Following Installing a Kafka Cluster and Creating a Topic - Hands-on Labs | A Cloud Guru, create a Kafka cluster and set it up.
3. Starting from the text corpus provided, create a javascript tag that can be used to track when a user is presented with a sentence and sends an Audio transcription of the sentence.
4. Create a data lake - S3 bucket
5. Write a DAG script to orchestrate the storage of the events collected to a database.
6. Using Spark, apply a transformation to load data to an S3 bucket.
7. Test pipeline.

The workflow for this week's challenge is as follows

- Read instructions and understand the business needs, the type of data available, the data engineering process(es) that needs to be carried out, the Workflow requirements, and the resources required/available to complete the project
- Plan your work and set up a development environment to assist in completing the project
- Explore a sample of the dataset, understand its structure, the information stored within, and develop intuition on how it can be used
- Set up a GitHub repo, integrate unit testing and CICD for proper code package test and deployment

Task 1: Plan your work

- In your assigned group, plan the flow of your work. Prepare backlogs and assign people to tasks. Use Github issues and project capabilities.
- Build or simulate a Kafka event source for the text corpus - you **should** read Breaking News: Everything Is An Event! (Streams, Kafka And You) (florimond.dev)

- Develop an overview of your approach and document it. Explain why this approach and why these tools. Explain how this approach will provide a good data source for the clients' speech-to-text ML engine. Explain the purpose of each of these tools - should defend it if one asks them why, not simple python code.

Task 2: Create a Kafka cluster

- Based on Installing a Kafka Cluster and Creating a Topic - Hands-on Labs | A Cloud Guru, set up a cluster in your assigned AWS machine.
- Your cluster will be responsible for creating a Delta Lake - a bucket in S3 where Spark transformed streaming data from users reading the texts you showed them are stored. (hint You will write a code that can generate an ID for a randomly selected text and its audio equivalent, receives an ID from an API, sends back as json the ID + audio to Kafka like URL

Task 3: Create a javascript tag

- The tag shall be used in front-end applications to communicate with your Kafka cluster - present a sentence to be read by a user and send back audio and other necessary metadata to your Kafka cluster.
- You should look at the following to understand how an app or a browser captures and sends audio and text events to your kafka cluster
    - Using the MediaStream Recording API - Web APIs | MDN (mozilla.org)
    - Handling Large Messages with Apache Kafka (CSV, XML, Image, Video, Audio, Files) - Kai Waehner (kai-waehner.de)

Task 4: Use Spark to transform and load from your Kafka cluster

- Using PySpark, write code that will transform and load the data from the data lake
- By using Kafka as an input source for Spark Structured Streaming and Delta Lake as a storage layer, build a complete streaming data pipeline to consolidate our data - you should read From Kafka to Delta Lake using Apache Spark Structured Streaming (michelin.io)

Submission

Interim 1: Due Wednesday 8 Sep 20:00 UTC

1. Task 1 (done in groups, the whole group can have the same submission). Your work plan submission should clearly show the client the work plan, the division of tasks, and how they are synchronized.
2. Task 2 (done in groups, the whole group can have the same submission) your task2 submission successfully created kafka topic in kafka cluster.
3. A one page brief (could be an email update to your client) including a detailed diagram explaining to the client (10 Academy) how Kafka, Airflow, and Spark will be used to provide them with data to improve their speech to text model

Interim 2: Due Saturday 11 Sep 2000UTC

1. To be added

Final Due Tuesday 14 Sep 2000UTC

1. To be added

Feedback

You will receive comments/feedback in addition to a grade.

References

Examples

- Keeping your ML model in shape with Kafka, Airflow and MLFlow | by Mike Kraus | VantageAI | Medium
- Real time Analytics Dashboard Using Apache Spark | CloudxLab Blog
- NavyaSreeKanakala/kafka-spark-nodejs: Building an analytics dashboard using Spark, Kafka and node.js (github.com)
- Traffic Data Monitoring Using IoT, Kafka and Spark Streaming (infoq.com)
- Using Kafka for Collecting Web Application Metrics in Your Cloud Data Lake | by Lucio Daza | Towards Data Science
- Create your own data stream for Kafka with Python and Faker (aiven.io)
- Using the MediaStream Recording API - Web APIs | MDN (mozilla.org)

Key references

- Breaking News: Everything Is An Event! (Streams, Kafka And You) (florimond.dev)
- How to process streams of data with Apache Kafka and Spark (microsoft.com)
- Structured Streaming + Kafka Integration Guide (Kafka broker version 0.10.0 or higher) - Spark 3.1.2 Documentation (apache.org)
- Installing a Kafka Cluster and Creating a Topic - Hands-on Labs | A Cloud Guru
- Why Apache Airflow Is a Great Choice for Managing Data Pipelines | by Kartik Khare | Towards Data Science
- From Kafka to Delta Lake using Apache Spark Structured Streaming (michelin.io)
- Handling Large Messages with Apache Kafka (CSV, XML, Image, Video, Audio, Files) - Kai Waehner (kai-waehner.de)
- Building-ML-driven-streaming-applications-Apache-Kafka-Final.pdf (awscloud.com)
- https://www.goavega.com/install-apache-kafka-on-windows/

Kafka event source

- confluentinc/kafka-rest: Confluent REST Proxy for Kafka (github.com)
- Apache Kafka Streams
- Event-driven microservice - ksqlDB Documentation
- Event Sourcing Using Apache Kafka | Confluent

Spark & Databricks

- https://docs.databricks.com/getting-started/try-databricks.html
- https://academy.databricks.com/elearning/INT-FDDBML-v1-SP
- https://spark.apache.org/sql/
- https://www.tutorialspoint.com/spark_sql/index.htm
- https://data-flair.training/blogs/spark-sql-tutorial/
- https://www.w3schools.com/js/js_whereto.asp
- https://sparkbyexamples.com/spark/show-top-n-rows-in-spark-pyspark/

Airflow

- https://airflow.apache.org/docs/apache-airflow/stable/start/docker.html
- https://pypi.org/project/apache-airflow/

General

- What is Kafka? A super-simple explanation of this important data analytics tool | Bernard Marr

Existing Similar Products

- Training data for Machine Learning — Toloka