Kamran Rasim Asgarov

Email: askerovk@gmail.com | Mobile: +49 171 2787742 LinkedIn: linkedin.com/in/k-r-asgarov
Location: Wuppertal, Germany

Yara International

Data Quality Control with Great Expectations. *AWS Redshift, AWS S3, AWS ECR, AWS ECS, Docker* 2024

- Implemented an ECS Task based Great Expectations microservice to check quality of production data.
- Designed and implemented a data model to store validation result in Redshift, used to power an anlytics dashboard.
- Wrote a python framework to extend GX functionality, handle batched validation requests and automatically populate fact/dimension Redshift tables with validation results.

PII (Personally Identifiable Information) security. *Lucid Charts, Confluence, Docker* 2024

- Developed an administrative procedure for other internal teams to requests long term access to PII data, utilising RBAC procedure in Redshift.
- Showcased the extensive impact of Tokenization and Just-in-time Decoding practices on downstream PII data security.

DBT as a Containerized Microservice *AWS MWAA, AWS ECS, AWS ECR, DBT, Docker* 2023

- Solved a dependency hell issue, by migrating DBT from MWAA server-side installation to a containerized microservice running on AWS ECS.

Airflow migration to MWAA *Airflow, AWS MWAA, AWS Secret Manager, Behave Framework* 2023

- Orchestrated the migration of our Airflow instance to AWS MWAA, from POC to production.
- Setup the AWS Secret Manager as a backend for Airflow connections and variables. Wrote a smoke test DAG in Behave framework to monitor MWAA status.

Active Campaign ETL *Python (pandas, requests), Airflow, Zuora, Active Campaign, Redshift* 2023

- Wrote python code to automate setup of a staging Redshift database, with identical schema, populated by sample data.
- Extended, documented and diagrammed legacy Airflow DAGS.

User Data Anonymization Pipelines *Python (pandas, boto3, requests), Airflow, S3, Redshift, Amplitude* 2022

- Wrote an Airflow DAG to download RudderStack logs in json/parquet formats from S3, parse and anonymize data for a given user and re-upload the file to S3.
- Wrote an Airflow DAG to anonymize Active Campaign data in Redshift.
- Wrote an Airflow DAG to send and track progress of anonymization requests to Amplitude API

Amplitude Historical Data *Amplitude, RudderStack, AWS S3, Python (boto3, pandas)* 2022

- Built a RudderStack ETL to parse RudderStack logs in an S3 bucket and push archived application data to an Amplitude project.

Metrics Layer POC *Transform/Metrics Flow, PostgreSQL* 2022

- Conducted a Proof of Concept project for implementing a Metrics Layer in the company, using the Transform platform and a sample SQL database.
- Showcased Transform's metric visualization capabilities to the Analyst Team.

Data Collectors Analytics *Python (pandas, boto3), MongoDB, Power Bi, SharePoint* 2021

- Built a Dockerized ETL to collect agent data from MongoDB, calculate performance metrics and write it as a .csv to SharePoint.
- Built a Power BI dashboard to highlight high/low performing data collector agents, as well as overall progress towards team goal.

5 Amigos Analytics *Python (pandas, sklearn, requests, boto3, GPSPhoto) AWS S3, Zendesk* 2021

- Scraped 25k comment threads from Zendesk and experimented with Latent Dirichlet Allocation to develop an automatic topic tagging system.
- Coded an ETL pipeline to parse metadata from images stored in S3 bucket, utilizing multiprocessing to cut down code execution time.
- Performed translation comparison study between Microsoft Azure, IBM Watson, Google and AWS translation APIs.

Agronomic Content Delivery Framework *Data modelling, knowledge graphs,* 2021

- Developed a solution-agnostic framework which governs the ingestion, customization, delivery and localization of agronomic content.
- Dynamic, farm specific roster of advisories/alerts based on crop type, region and farmer knowledge/available tech.
- Shared microservices for location mapping, crop calendar, fertilizer recommendation, etc...
- Agile content localization of agronomic advisories, by reducing them to fact bundles.
- Advice triggered by farm specific calendar, regional alerts or farmer activity.

Crop Calendar *Python (pandas, matplotlib, seaborn)* 2020

- Developed and implemented a data framework for unifying regional crop calendars, based on BBCH crop stages.
- Created a series of visualizations to critique legacy crop calendar data, obtained from external sources.

Soil Modelling *Python (pandas, sklearn, folium, geopandas, shapely, plotly, matplotlib), PostgreSQL* 2020

- Used a folium, a python based API for the Leaflet JavaScript library to generate interactive soil maps of several Indian States.
- Trained kriging and inverse-distance mean interpolation methods for generating soil composition models.
- Built a algorithm for visualizing and evaluating the performance of fertilizer recommendation models, given real world soil data.
- Used findings to critique the use of district level soil compositions in India.

Soil Health Cards *Python (boto3, requests, pandas), PostgreSQL, AWS RDS, S3, Docker, Proxy server* 2019

- Wrote scraping code to acquire 20 million soil composition samples from an Indian government website as excel tables, stored in S3 bucket.
- Scraping code was containerized with Docker. Each container was self-sufficient and obtained tasks from a centralized SQL table, allowing to run any number of parallel scraping jobs.
- Wrote parsing script, which read excel files in S3, parsed tables within, de-normalized data and uploaded to a PostgreSQL database on RDS.
- Spearheaded a landmark agreement with Indian government for direct access to the data.

Intern Projects *Python (boto3, voronoi), AWS Sagemaker* 2019

- Supervised creation of a Voronoi Diagram, in order to find the nearest major city for any location in India. Our approach was 3 times faster than previously used algorithm.
- Supervised creation of an S3 to S3 file migration code, optimised to run on several processes, deployed on AWS Sagemaker.

Crop Disease Models *Python (unit testing, object oriented programming), API* 2018

- Wrote python implementation of 5 crop diseases forecasting models for a farmer app.
- Used regional call center data to estimate crop disease outbreaks, along with IBM weather API, in order to validate the predictive power of crop disease models.

FarmPulse *Python (requests, beautifulsoup), PostgreSQL, cronjob, AWS (Lambda, RDS, S3)* 2018

- Wrote a python script to scrape 200k call centre records from a web widget.
- Used serverless framework to deploy scraping/parsing code to AWS Lambda, triggered by a cronjob. Raw files were stored in S3 bucket, while parsed data was added to a PostgreSQL database on RDS.