Preliminary Information

First of all, we are so grateful for your interest in being our teammate. At *Teknasyon*, we have broad-based experience in building data-intensive applications, overcoming complex architectural, and scalability issues in diverse domains. We are at the forefront of leveraging machine learning and artificial intelligence to drive innovation and solve complex challenges. We are a team of dedicated professionals committed to harnessing the power of data to unlock new possibilities. As we continue to grow, we are seeking a talented MLOps/Data Engineer to join our team and play a key role in shaping the future of our data-driven initiatives and substantial deliverables.

The data denominated 'the new oil' in the past is spreading into our universe incredibly. We know that there are lots of services, systems, databases, and so on. We avoided preparing a strict challenge and created some sections for you. The result of this case study does not directly affect our decision. Motivation and communication are more important for us.

There are three different sections. You can choose one or do all of them. You don't have to do other sections technically, you can propose high-level design (sketch). However, we expect you to do practically **at least one** of all sections.

Section I - *Designing MLOps Pipeline*

An End-to-End Anomaly Detection

- Background

Getcontact having been launched in 2016 has consolidated itself as one of the leading mobile identity and fraud prevention services globally, with more than 60 million active users per month and a growing presence across four continents. Such a rapid growth of the Getcontact application means keeping many basic KPIs, such as registered users and verified users, under control on the technology side. Daily review of KPIs and supporting them with machine learning algorithms help to plan growth strategies more accurately. Monitoring the changes in our users' behaviors closely helps us to prevent and balance system overload in some critical cases. Then, the loads can be distributed wisely if we detect the correct anomaly points in KPIs counts.

- Dataset Description

The dataset contains daily API log data between Feb 1, 2024, and April 14, 2024. All data masked. Each record in the dataset includes the following information:

➔ date: The date of the log entry.
➔ country: The geographic location of the API request (country code; ISO 3166-1 alpha-2).
➔ os: The operating system used by the device making the API request.

➔ count: The number of API requests made on that date from that county and OS combination.

❖ Task I

➢ Explore the provided dataset to understand its structure and characteristics.
➢ Identify any data quality issues or anomalies that need to be addressed before proceeding with the analysis.
➢ Develop a machine learning model or algorithm to detect anomalies in the API log data. We expect you to predict the last two weeks. Which day, which country, and which OS is an anomaly?
➢ Consider different approaches for anomaly detection, such as statistical methods, machine learning algorithms, or time series analysis (**Hint**: IQR calculation).
➢ Evaluate the performance of your anomaly detection solution using appropriate metrics.

❖ Task II

Your task II is to develop an end-to-end solution that encompasses the design and management of an MLOps pipeline to automate the ML lifecycle.

➢ Design an MLOps pipeline to automate the ML lifecycle, including data ingestion, preprocessing, model training, evaluation, deployment, and monitoring.
➢ Implement the MLOps pipeline using tools and platforms of your choice (e.g., Apache Airflow, MLflow, Kubernetes).
➢ Ensure the pipeline is scalable, robust, and reproducible, with proper error handling and logging mechanisms.

❗ *We don't set any conditions for every two tasks. Section I was prepared to solve without any given tool, service, or language dependency. You have full autonomy to choose the tools, libraries, and methodologies for this challenge. You can use Python in modeling or even write SQL to detect outliers. You are also free to serve the results. You can choose a serving method whatever you want. An API is the first thing that comes to mind.*

Deliverables

- Any code/scripts for data exploration, anomaly detection modeling, and MLOps pipeline implementation.
- Documentation detailing the MLOps pipeline design, architecture, and workflow.
- A summary report documenting your findings, methodology, MLOps pipeline design, and recommendations.
- Visualizations and plots to support your analysis and findings.

Section II - *Managing the ML Lifecycle*

Adding New Feature to Spam Algorithm

- Background

Getcontact is a great “Spam Blocking” and “Caller Identification” application with more than 60 million active users per month and a growing presence across four continents. Getcontact filters disturbing calls and allows only the people you prefer to communicate with you. You can identify the calls you receive from numbers not registered in your contacts. Getcontact alerts instantly if you get an unwanted call so that you get real-time protection from robocalls, telemarketers, and scam calls.

- Dataset Description

The dataset contains call detail records and its time grain is a only week (April 8-15, 2024). All data masked. Each record in the dataset includes the following information:

- ➔ id: User number id.
- ➔ country: User country code (ISO 3166-1 alpha-2).
- ➔ call_count: Shows the total number of calls made by the relevant number.
- ➔ unique_call_count: Shows how many different people the number has called.
- ➔ total_duration: Shows the total number of seconds the number has spoken.
- ➔ answered_call_count: It shows how many of the calls made by the number are answered.
- ➔ rejected_call_count: It shows how many of the calls made by the number are rejected.
- ➔ missed_call_count: It shows how many of the calls made by the number are missed.
- ➔ weekdays_call_count: Shows how many of the outgoing calls were made during the week.
- ➔ weekend_call_count: Shows how many outgoing calls were made over the weekend.
- ➔ overtime_call_count: It shows how many of the outgoing calls are made during working hours.
- ➔ out_of_work_call_count: Shows how many outgoing calls are made outside of working hours.

New Feature:

- ➔ manual_spam_count: Shows the number of times it was marked as spam manually.

❖ Task I

- ➢ The aim of the project is to analyze whether the relevant user is making spam calls based on users' call histories.
- ➢ Explore the provided dataset to understand its structure and characteristics.
- ➢ Identify any data quality issues or anomalies that need to be addressed before proceeding with the analysis.
- ➢ Develop a machine learning model to predict and identify spam calls based on the provided features.
- ➢ Evaluate the performance of your updated spam detection model using appropriate metrics

❖ Task II

- ➢ Your task II is to automate the addition of the new feature (manual_spam_count) to the existing spam detection algorithm as part of the MLOps lifecycle.
- ➢ Design an MLOps pipeline to automate the process of feature engineering, model retraining, evaluation, and deployment whenever a new feature like manual_spam_count is added.
- ➢ Implement the MLOps pipeline using tools and platforms of your choice (e.g., Apache Airflow, MLflow, Kubernetes).
- ➢ Ensure the pipeline is scalable, robust, and reproducible, with proper error handling, logging mechanisms, and version control for models and features.
- ➢ Develop a mechanism to monitor the performance of the updated spam detection model in production and trigger retraining if necessary.

Deliverables

- Code/scripts for data exploration, model development, and MLOps pipeline implementation.
- Documentation detailing the updated spam detection model, feature engineering process, MLOps pipeline design, architecture, and workflow.
- A summary report documenting your findings, methodology, MLOps pipeline design, and recommendations.
- Visualizations and plots to support your analysis, model performance evaluation, and MLOps pipeline components.

Section III - *Project Management Experience*

Now, It's Your Turn!

This section is a little bit different than others. We want you to solve this task by using your experience and one of the past projects.

- ➢ What project are you most proud of?
- ➢ If you have a chance to refactor it, which part do you develop?
- ➢ Could you give a high-level design of that project? Plus, we are waiting for you to specify alternative methods/solutions/structures. For example, if you use the GCP Bigquery as a cloud DWH or source, you should indicate another cloud DWH product such as Redshift, Snowflake, etc.

Deliverables

- A summary of the project. *Ex: What was the purpose of the project? How much data were processed? Where were the outputs used?*
- A report/diagram for the flow of your project. ***(draw.io, lucid.co)***
- An explanation of the used tech stack in detail. *Ex: Which database or cloud provider was used? Why did you choose it?*