

## 1) What are the assumptions you have made for this service and why?

I assumed:

- Exploring the right configurations to train the models for predictions. Because there are several copies of the model with different configurations.
- The code will be messy as it is just exploring patterns. Because Data Scientists are focused on finding outcomes, not developing code.
- This is the pre-stages of an evaluated business goal and use case. Because it just says to use some “magic” in the task section. I had the impression there’s no business goal and/or use case.
- There will be more data exploration than predicting. Because most of the time, data cleaning is required.
- If it’s data exploration, then they’ll use Jupyter Notebooks because it’s easier to visualise data (plug and play).
- I assumed the data was already checked before processing the statistics.
- There wasn’t much commenting and too much unused code. At a research level, it doesn’t need to express a hypothesis if it’s unknown.

Some comments are added in the codebase as well.

## 2) How would you do to transition this solution into a cloud-based solution?

You can run it on Lambda and save the statistics and models in an S3 bucket (models will be pickle/joblib).

Another way is running a Cronjob in an EC2, using Python CDK Sagemaker to deploy models to Docker (that’s assuming you know what hyper-tuning is).

If you prefer a notebook (Data Scientists love it), you can use AWS Sagemaker notebook connected to a database (or similar) to run. If the analysis is satisfied, you can then deploy the model in the ECS (Docker) URL. Then link that REST API service to produce predictions. You can also save the analysis in an S3 bucket.

### 2a) What changes (if any) to your code would need to be made?

You need to add Python CDK to access the cloud for S3, ECS and Sagemaker tools. You may need to create a REST API app (Flask) to interact with the models and return it back via JSON. The code needs decoupling to narrow down errors (if any).

## **2b) What services would be required to implement this and which teams would you need to support you? Feel free to use any cloud provider you are familiar with or a diagram if helpful.**

For services, it needs a cloud service (AWS: Sagemaker, ECS, S3, Lambda, API Gateway). I will need a DevOps/MLOps to deploy the work, and a backend developer(s) for the REST API (if going down that route). Data Engineer to provide data (assuming it's stored in one).

S3: Storing the statistic files

Sagemaker: the model to build and deploy ML models (URL).

ECS: the storage of the model's object.

API Gateway & Lambda: To request predictions through the API Gateway service, the Lambda calls the model from ECS to produce predictions.

## **2c) How would you ensure reliability of newly developed code before releasing to a production environment?**

Everyone should be following a version control service like GitHub, which should include stages to avoid production errors. There should be at least a development and production stage (UAT as an option). Each stage should have a separate cloud service (develop cloud account, production cloud account) and should test the project in development and then in production. The code must be reviewed and approved, and the CI/CD pipeline should provide auto-deployment per stage.

## **3) What considerations are there to ensure the business can leverage this service? Is there anything you would need to improve or confirm with stakeholders that would increase the efficacy of this service?**

A few:

- We should make sure the resources can maintain any new service.
- We should check if our current service is sufficient to maintain the new service. If not, then we need to scale up or upgrade.
- Several stakeholders like the CTO, Head of Data, Lead Engineer or equivalent who is managing the entire operation need to know. Potentially a meeting to explain the dynamics. If budget changes are involved, then financial stakeholders need to know.

## **4) What is in and out of scope for your responsibility?**

**The in of scope:**

To understand the task and plan out the operation (if a new project, then taking ownership). To make sure the project is developed, robust to production level and ensure strong communication within the code (documentation, testing). To communicate with team members occasionally.

**The out of scope:**

To make sure the project aligns with a business goal (approval if necessary). To make sure there is validation done before coding. Understanding the use case and why it's here, what problem it solves and why. Is there an alternative solution that doesn't require resources?