**The Project Outline**

1. **The Problem Statement**
2. **Description of Data**
3. **Application Architecture and module division**
4. **Code Explanation**
5. **Data Ingestion**
6. **Data Validation**
7. **Data Preprocessing**
8. **Model Selection**
9. **Model Tuning**
10. **Prediction**
11. **Logging Framework**
12. **Deployment**

**The Problem Statement:**

1. Do a complete EDA in the python notebook file
2. Build a solution design architecture for end to end solution starting from data ingestion to deployment with a detail documentation.
3. Deploythe end to end automated solution to GCP.
4. Create a user interface for bulk testing uploaded through excel sheet and for a single record entry both.
5. Maintain log for each and every prediction request into any database
6. Create a low level documentation for end to end solution and deployment
7. Define a retraining approach in your documentation
8. Create an end to end video of the working of the project

**Data Description**

A backorder is the order which could not be fulfilled by the company. Due to high demand of a product, the company was not able to keep up with the delivery of the order. The backordering can lead to upsetting customer as they couldn't get what they ordered and the loyalty will decrease.

Also, company cannot overstock every product in their inventory to avoid such situation.

There has to be a way for the company to know for which products they can face this problem.

So, the company has shared a data file with different input features for each product and it hopes to find a pattern inside this data which can give them some insight.

The data file contains the historical data for some weeks prior to the week we are trying to predict.

The data has 23 columns including 22 features and one target column.

To model and predict the target, we’ll use the features columns, which are:

**sku** – Random ID for the product

**national\_inv** – Current inventory level for the part

**lead\_time** – Transit time for product (if available)

**in\_transit\_qty** – Amount of product in transit from source

**forecast\_3\_month** – Forecast sales for the next 3 months

**forecast\_6\_month** – Forecast sales for the next 6 months

**forecast\_9\_month** – Forecast sales for the next 9 months

**sales\_1\_month** – Sales quantity for the prior 1 month time period

**sales\_3\_month** – Sales quantity for the prior 3 month time period

**sales\_6\_month** – Sales quantity for the prior 6 month time period

**sales\_9\_month** – Sales quantity for the prior 9 month time period

**min\_bank** – Minimum recommend amount to stock

**potential\_issue** – Source issue for part identified

**pieces\_past\_due** – Parts overdue from source

**perf\_6\_month\_avg** – Source performance for prior 6 month period

**perf\_12\_month\_avg** – Source performance for prior 12 month period

**local\_bo\_qty** – Amount of stock orders overdue

**deck\_risk** – Part risk flag

**oe\_constraint** – Part risk flag

**ppap\_risk** – Part risk flag

**stop\_auto\_buy** – Part risk flag

**rev\_stop** – Part risk flag

**went\_on\_backorder** – Product actually went on backorder. This is the target value.

**Architecture Application flow and module division**

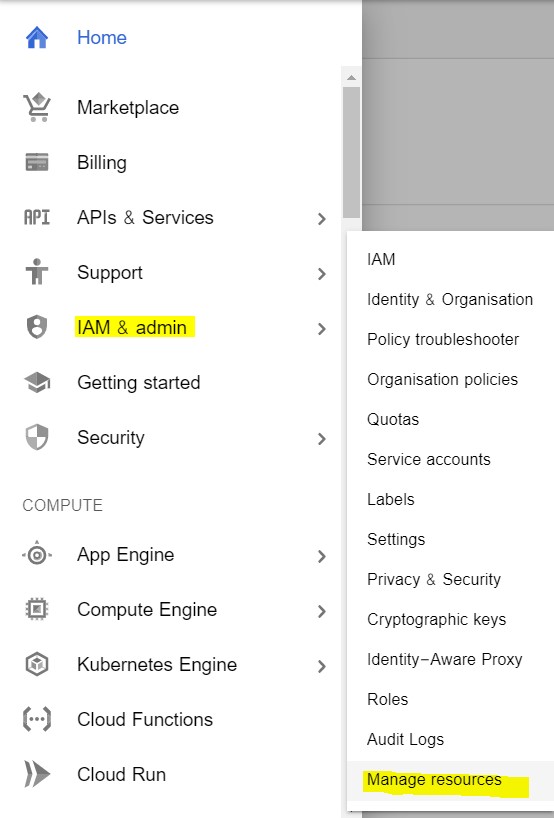


**Deployment to G-cloud**

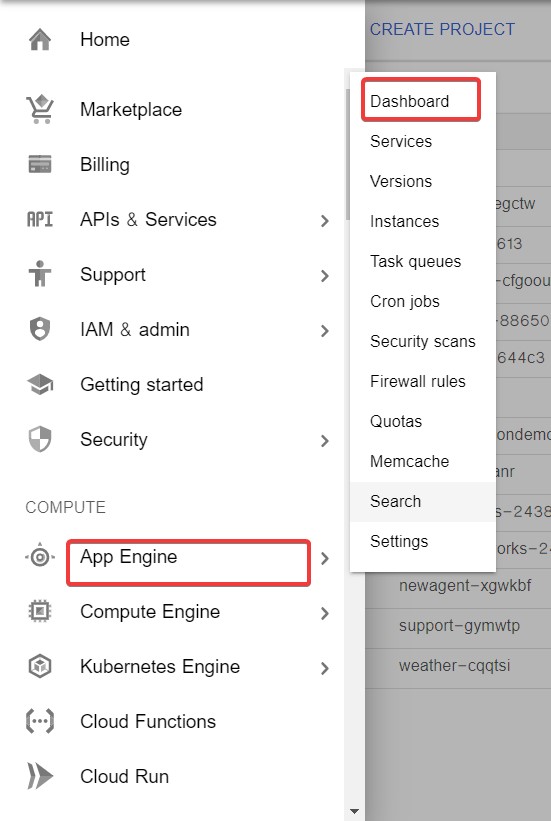
* Go to <https://cloud.google.com/>and create an account if already haven’t created one.

Then go to the console of your account.

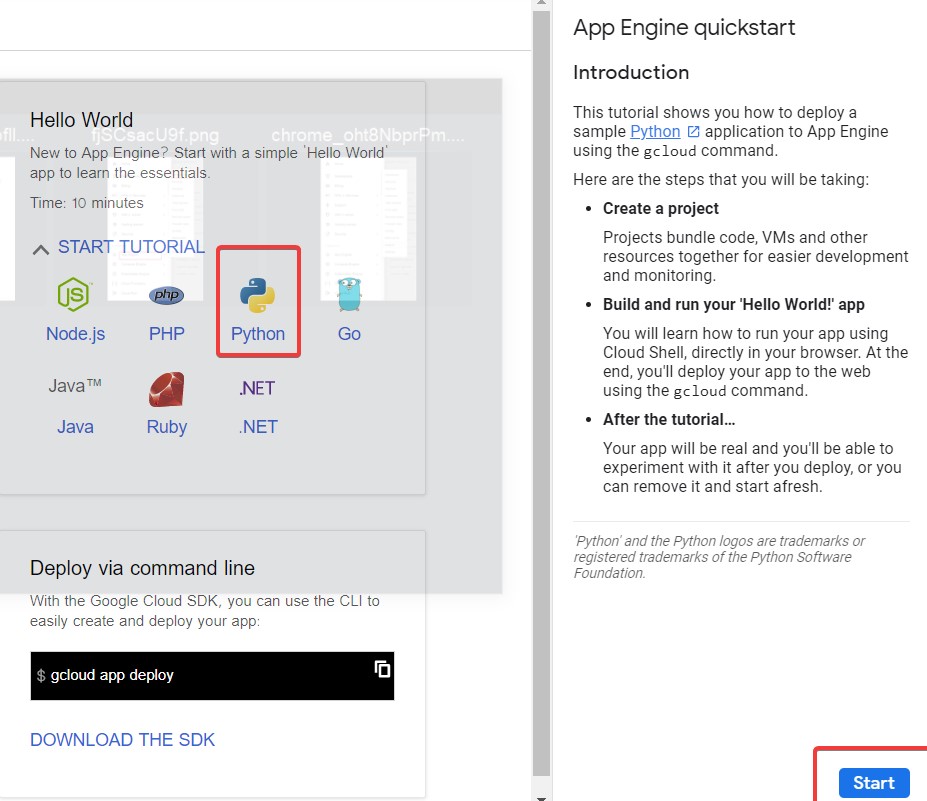
* Go to *IAM and admin*(highlighted) and click *manage resources*.



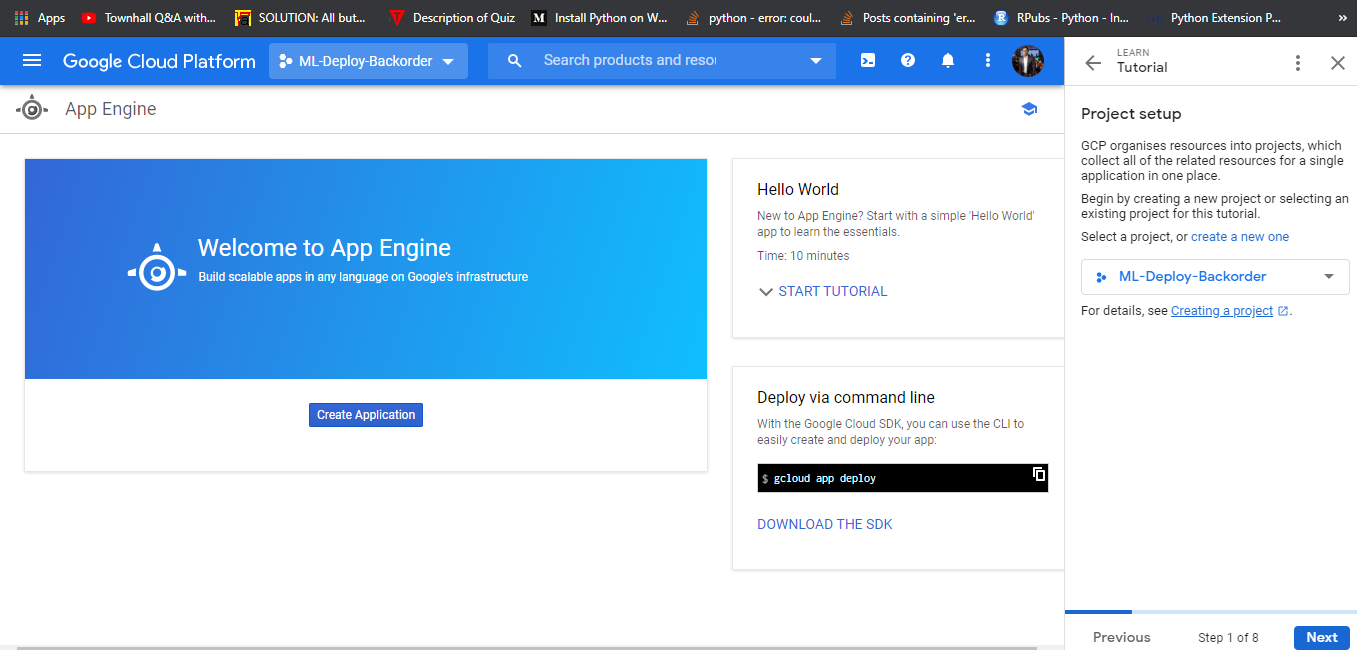
* Click *CREATE PROJECT* to create a new project for deployment.
* Once the project gets created, select *App Engine* and select *Dashboard.*



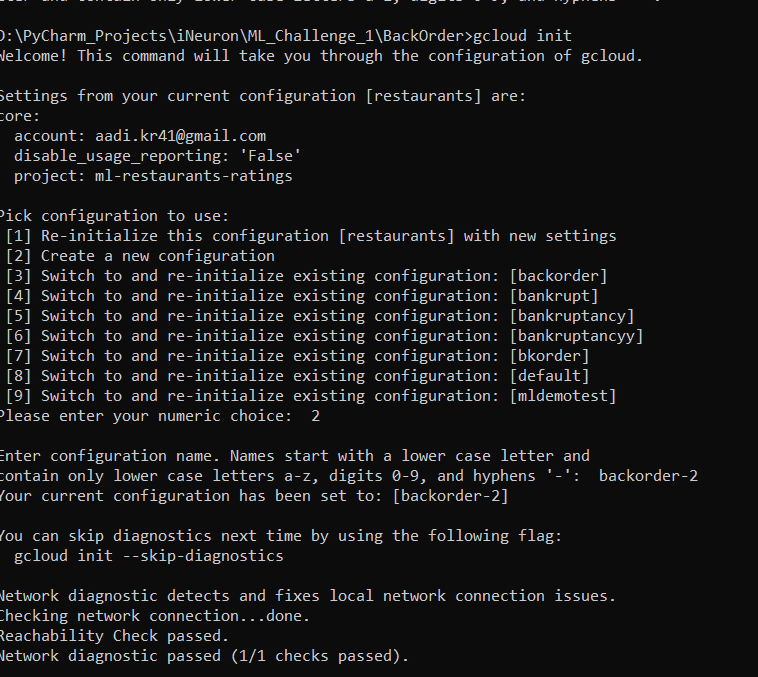
* Go to <https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe>to download the google cloud SDK to your machine.
* Click *Start Tutorial* on the screen and select Python app and click start.



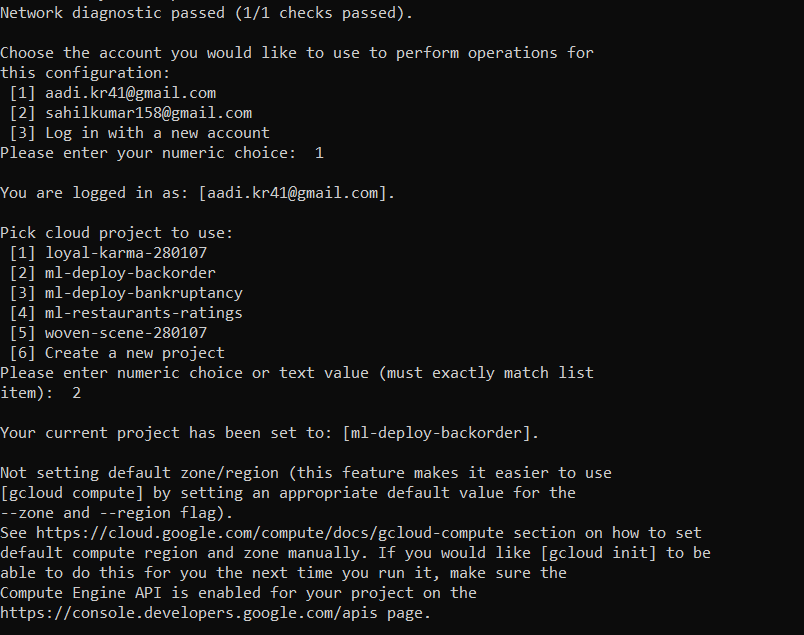
* Check whether the correct project name i.e ML-Deploy-Backorder is displayed and then click next.



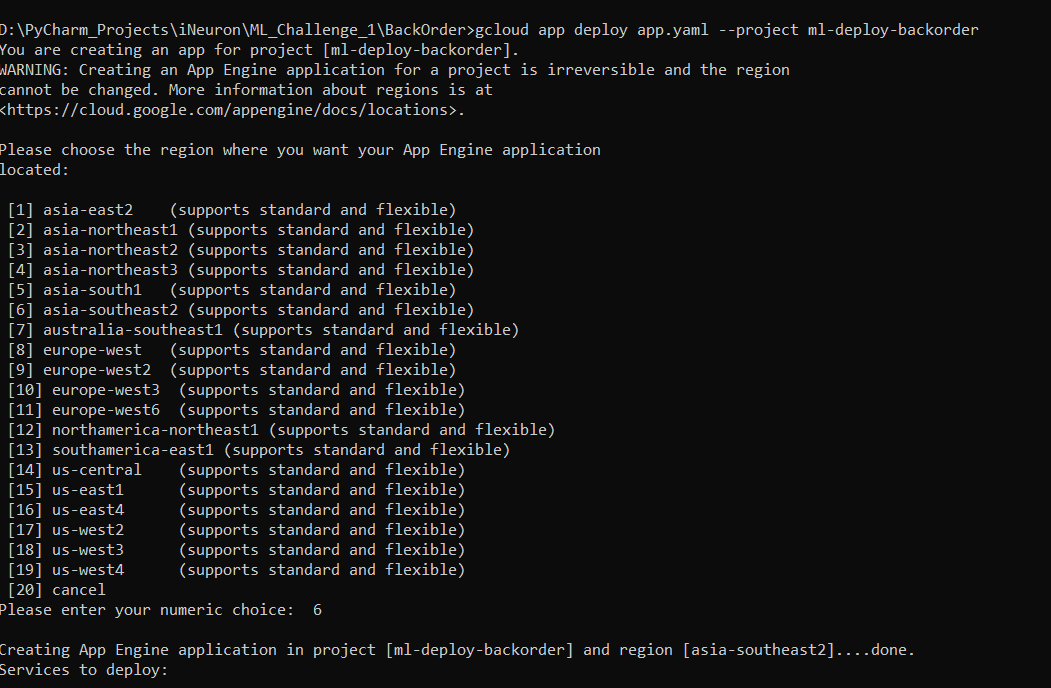
* Create a file ‘app.yaml’ and put ‘runtime: python37’ in that file.
* Create a ‘requirements.txt’ file by opening the command prompt/anaconda prompt, navigate to the project folder and enter the command ‘pip freeze > requirements.txt’. It is recommended to use separate environments for different projects.
* Your python application file should be called ‘main.py’. It is a GCP specific requirement.
* Open command prompt window, navigate to the project folder and enter the command *gcloud init* to initialise the gcloud context.
* It asks you to select from the list of available projects.

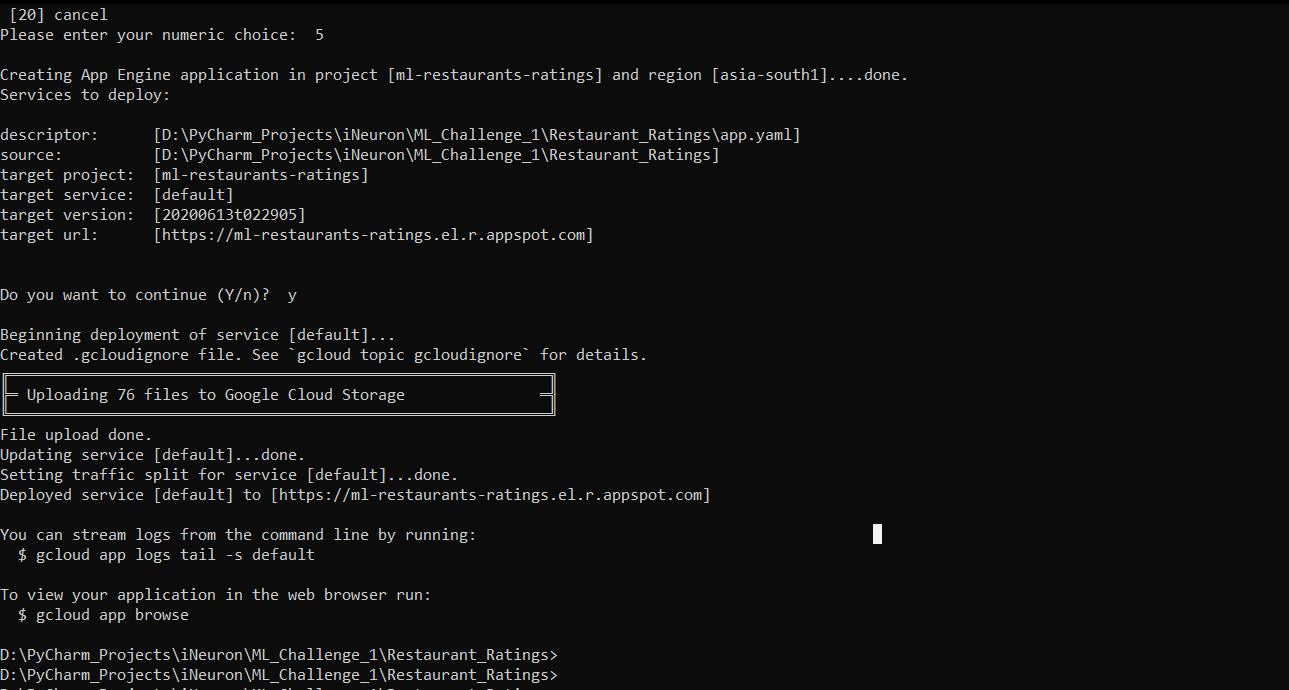


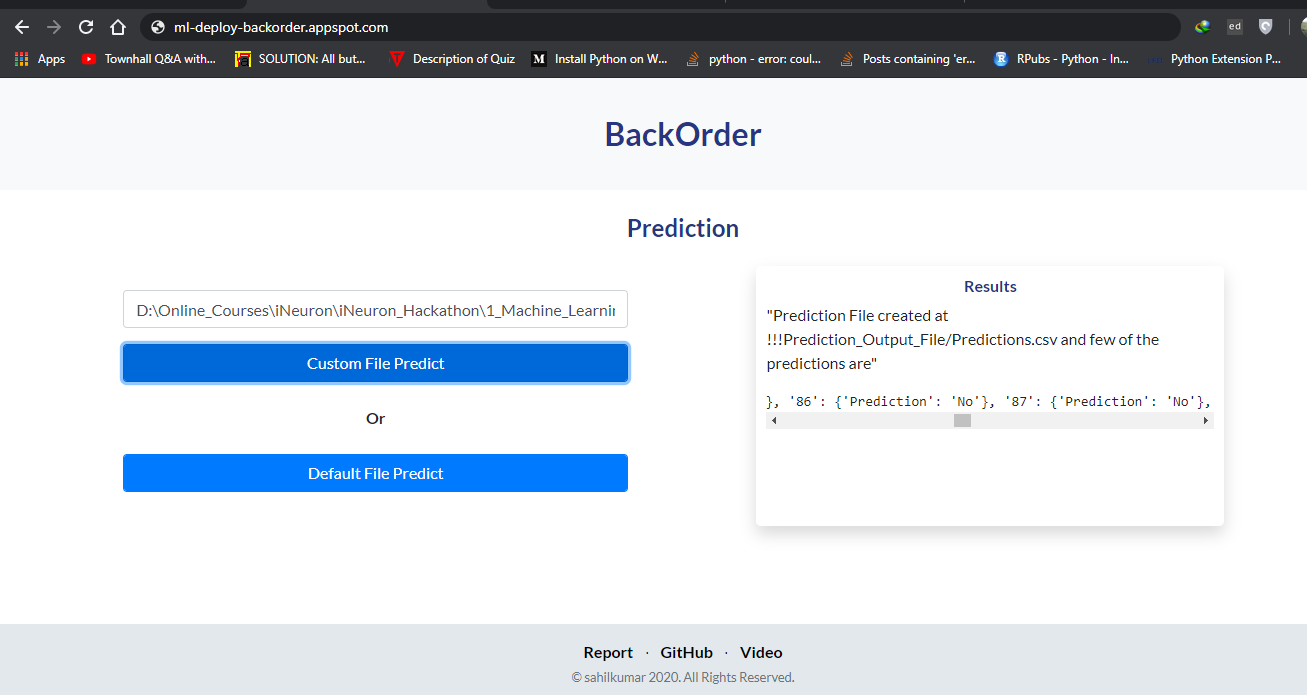
* I selected 2 option to create new configuration as backorder\_2.
* Select email id in which want to deploy the project.



* Once the project name is selected i.e ml-deploy-backorder , enter the command gcloud app deploy app.yaml --project ml-deploy-backorder.
* After executing the above command, GCP will ask you to enter the region for your application. Choose the appropriate one.



* I selected 6 option as asia-southeast2.
* GCP will ask for the services to be deployed. Enter ‘y’ to deploy the services.
* And then it will give you the link for your app, and the deployed app looks like:



* To save money, go to App Engine > settings and disable your app.

