Polymerize Hiring Challenge

ML Engineer

We'd suggest that you keep all ipynb, py, csv files you create at a single location and share it with us at the end. Also, write a small essay in a readme to help us better understand some of the decisions you made.
If you decide to upload your workings on GitHub, don't forget to make the repo private as public repos will hinder the process and as a result would mostly be ignored.
Your implementation will be evaluated on **readability, performance,** and **research reproducibility**
Submit the finished files in a zip/link to kunal@polymerize.io

Step 1:- EDA

Exploratory Data Analysis would be one of your primary focuses as an ML Engineer.

Since the Polymerize AI Team usually deals with structured data, the dataset is fairly simple. Go here to download the csv and get started with the EDA.

CSV Link :- https://polymerize-misc.s3.ap-southeast-1.amazonaws.com/3dprinterdata.csv

Don't forget to add the ipynb file for your Analysis in the submission.

Step 2:- Predictive Analysis

Develop models/algorithms to predict the **Roughness, Tensile Strength & Elongation** variables using the information from the analysis you performed earlier. Choose the appropriate metrics to explain how good your models are and why you chose the model, its hyper-parameters, and the metrics. Don't forget to serialize your model/s for use in the next challenge.

Step 3:- Exploration & Optimization

For the main part of the project, we need to explore the search space a bit. Write a function to generate input points for which the output will be in a given range, this range, along with the number of points should

be parameters for the function. Decide the distribution for your points and explain why you made each decision. Set appropriate input constraints to filter out incorrect values

For the Multi-Objective Optimization part, you need to optimize the variables **[Roughness, Tensile Strength, Elongation]** in the directions **[Minimize, Maximize, Maximize]** respectively

Write a function to optimize the outputs in the given directions. Use or implement whatever algorithms you think would work, there are no library constraints. Set appropriate input constraints to filter out incorrect values

Step 4:- Deployment / Operations

Finally, you'll need to deploy the developed algorithms and write Handlers and API endpoints so that we may test the model outputs for Forward as well as Inverse Predictions, along with functionalities to monitor and retrain the models. Ideally, deploy the APIs on a hosting site like Heroku/AWS, however, you could also deploy on your localhost and share the json docs generated. Do remember that you'd need to demonstrate your process and workings in the call so make sure everything is working beforehand.

BONUS

- Parallelise

  Try to parallelize your algorithms to speed them up.

- Meta-Models

  You might try to combine multiple models into one using a multi-tiered Meta-Model architecture.

- Deep Learning

  You could add an element of NAS to the research process

- Handling Data on NoSQL databases.

  Imagine if the data used for training could be kept in a database instead of puny csv files. Wouldn't it be great for adding new functionalities in the future?

- Using celery to track processes in parallel

  Training Models could be a time-intensive process, as such the API might not need to be open for a response waiting for the model to get trained. Celery might come to your rescue here.

- Dockerize

  Dockerizing/Containerizing the app would be your ideal scenario.