Model Scheduling with ADF - Implementation Details

remaining things need to be covered by the pipelines

- databricks cluster for scoring notebook
  - new job cluster, needs cluster settings (libs, etc.) either in the ADF repo, or manually input by the ADF model scheduling developer. Cannot control the size of job cluster to be created
    - option1 - use notebook activity libs: manually input the append libraries
    - option2 - use notebook activity libs: store a config/env file per model in the adf repo (or reference to the model repo), and then trigger a devops pipeline to update the adf/pipeline/xxxx.json's libraries object and commit itself to the repo branch, so that the adf pipeline's notebook activity will be updated with the libs. This approach requires the adf pipeline or notebook activity uses specific names, so that the devops pipeline knows which model to be updated by what libs.
    - ~~option2 - use databricks linked service advanced section: it does not seem to supporting specify libs here~~
    - option3 - auto-generate the adf pipeline notebook activities based on the model repos, triggered when/after mlops QA release.
  - existing cluster, needs cluster ID of both qa and prd ops databricks per model in the ADF repo
- common notebook params to add to every scoring notebook activity