Challenge:

Dataset Overview

The dataset is having incidents raised by customers. Which contains an event log of an incident management process extracted from a service desk platform of an IT company.
Once incident is raised it needs to be assigned to support group and support engineer.

This challenge is divided into several parts:

Part 1

IT team has limited capacity and therefore, they are interested in solution that could help them prioritize important incidents.
Every incident is manually evaluated and labeled with ***Impact*** value which provide us information about incident importance. IT team wants to solve incidents with high impact as soon as possible. Impact value may be updated during troubleshooting process.

Try to create solution for this problem and create quick PoC where you will evaluate your idea.

Expected deliverables:

Code:

Preferably commented, self-explaining ipython notebook
(Note: try to avoid sharing the notebook as attached ipynb file, use repository or any cloud notebook hosting service)

Presentation:

How would you present the idea to business users from IT team?

- Max. 3 slides
- List findings, explain approach, and explain benefits and why and how will they benefit from selected approach.

Part 2

a) What would be your next steps if IT team likes your PoC from part 1?

b) IT team would like to understand if they can utilize ML/AI for other processes. Think about use-cases and problems that could be solved by ML capabilities, list your ideas. (Base your ideas on provided dataset).

c) How would you productionize suggested solution?

Dataset details:

Values provided at Incident creation / re-opening:

- **ID**
  Incident identifier (24,918 different values)
- **ID_status**
  Eight levels controlling the incident management process transitions from opening until closing the case
- **active**
  Boolean attribute that shows whether the record is active or closed/canceled
- **count_reassign**
  Number of times the incident has the group or the support analysts changed
- **count_opening**
  Number of times the incident resolution was rejected by the caller
- **count_updated**
  Number of incident updates until that moment
- **ID_caller**
  Identifier of the user affected
- **opened_by**
  Identifier of the user who reported the incident
- **opened_time**
  Incident user opening date and time
- **Created_by**
  Identifier of the user who registered the incident
- **created_at**
  Incident system creation date and time
- **updated_by**
  Identifier of the user who updated the incident and generated the current log record
- **updated_at**
  Incident system update date and time
- **type_contact**
  Categorical attribute that shows by what means the incident was reported
- **location**
  Identifier of the location of the place affected
- **Category Id**
  First-level description of the affected service
- **user_symptom**
  Description of the user perception about service availability

Values selected [manually] after Incident is raised and:

- **Impact**
  Description of the impact caused by the incident (values: 1-High; 2-Medium; 3-Low)
- **Support_group**
  Identifier of the support group in charge of the incident
- **support_incharge**
  Identifier of the user in charge of the incident
- **Doc_knowledge**
  Boolean attribute that shows whether a knowledge base document was used to resolve the incident
- **confirmation_check**
  Boolean attribute that shows whether the priority/Impact field has been double-checked
- **Notify**
  Categorical attribute that shows whether notifications were generated for the incident
- **Problem_id**
  identifier of the problem associated with the incident
- **change_request**
  identifier of the change request associated with the incident