Data Challenge ‘22

The submission guidelines for competition entries as well as the process for evaluating models are outlined in the following slides

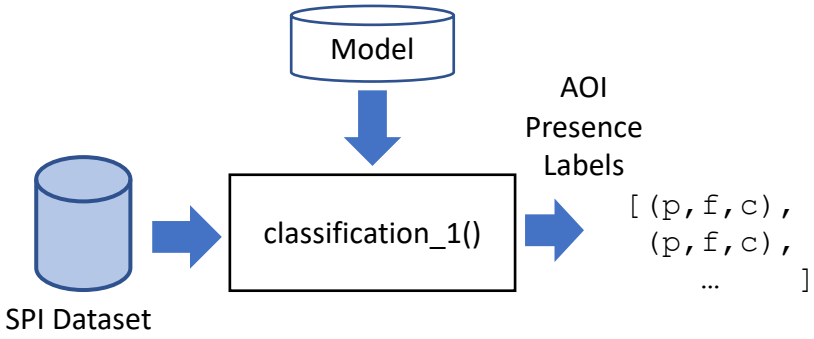
Submission Guidelines

Submission Format

- Each submission entry must include:
- **Test Classification:** the jupyter notebook that runs the classification tasks on new data.
  - Use the provided `solution.ipynb` as base notebook.
- **Models:** the trained models that will be used to classify new data
  - Your notebook must read and use them to provide classification
- **Short Paper**: a short paper (Max 4 pages) describing the team solution

Task 1: AOILabel

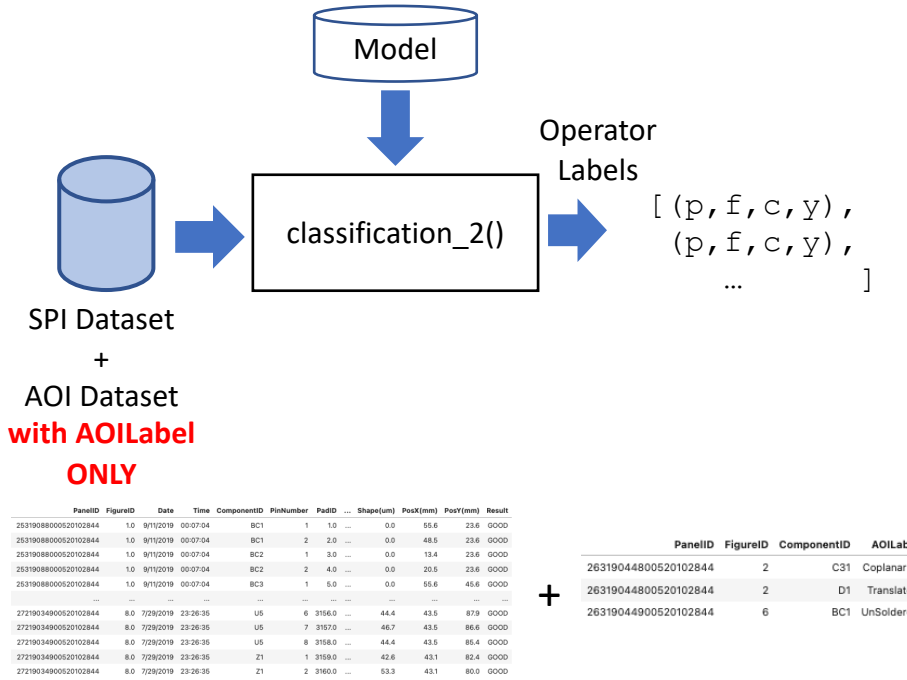
- **Input**:
  - **SPI DataFrame**: a Pandas DataFrame as read from the CSV file using pd.read_csv()
- ***classification_1()*** reads a pre-trained model that you should store in the JupyterHub directory (or you can attach to your submission)
  - Predicts whether the AOI will report a defect in a component.
- **Output:**
  - **AOILabels: the predicted defects in the form** a list of tuples in the form **(Panel (p), Figure (f), Component (c))**
  - ***Note*: the list must include the *only the defects components***. If for a component your classifier **does not predict a defect, it must not** be included in the list.



PanelID	FigureID	Date	Time	ComponentID	PinNumber	PadID	...	Shape(um)	PosX(mm)	PosY(mm)	Result
253190880005201028 44	1.0	9/11/2019	00:07:04	BC1	1	1.0	...	0.0	55.6	23.6	GOOD
253190880005201028 44	1.0	9/11/2019	00:07:04	BC1	2	2.0	...	0.0	48.5	23.6	GOOD
253190880005201028 44	1.0	9/11/2019	00:07:04	BC2	1	3.0	...	0.0	13.4	23.6	GOOD
253190880005201028 44	1.0	9/11/2019	00:07:04	BC2	2	4.0	...	0.0	20.5	23.6	GOOD
253190880005201028 44	1.0	9/11/2019	00:07:04	BC3	1	5.0	...	0.0	55.6	45.6	GOOD
...	...	...	...	...	...	...	...	...	...	...	...
272190349005201028 44	8.0	7/29/2019	23:26:35	U5	6	3156.0	...	44.4	43.5	87.9	GOOD
272190349005201028 44	8.0	7/29/2019	23:26:35	U5	7	3157.0	...	46.7	43.5	86.6	GOOD
272190349005201028 44	8.0	7/29/2019	23:26:35	U5	8	3158.0	...	44.4	43.5	85.4	GOOD
272190349005201028 44	8.0	7/29/2019	23:26:35	Z1	1	3159.0	...	42.6	43.1	82.4	GOOD
272190349005201028 44	8.0	7/29/2019	23:26:35	Z1	2	3160.0	...	53.3	43.1	80.0	GOOD

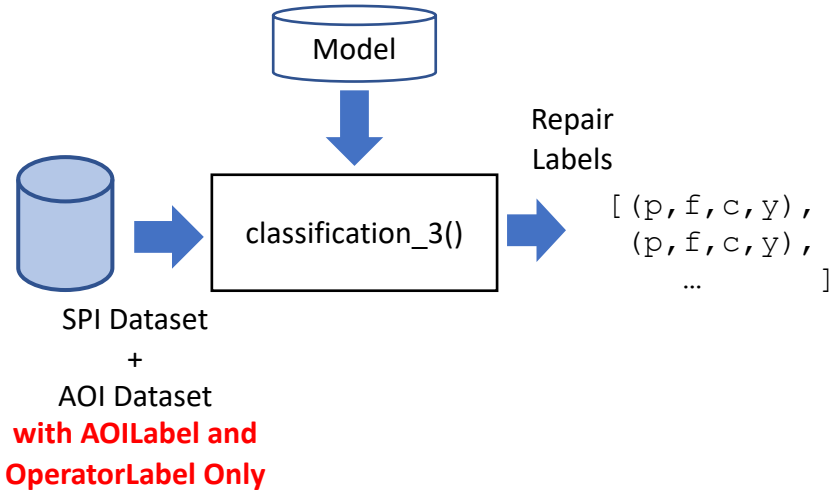
Task 2: OperatorLabel

- **Input**:
  - **SPI DataFrame**: a Pandas DataFrame as read from the CSV file using pd.read_csv()
  - **AOI Label:** a Pandas DataFrame that we read from the AOI CSV file using pd.read_csv(). We include only a subset of the columns. They are:
    - `"PanelID","FigureID","MachineID","ComponentID","PinNumber",` **`"AOILabel"`**
    - **Note:** The only useful column in the AOI dataset is the **AOILabel**. This is the only additional feature available. The other columns must be used to join the **SPI** and the **AOI** dataframes.
- ***classification_2()*** reads a pre-trained model that you should store in the JupyterHub directory (or you can attach to your submission).
  - Predicts the operator label for the components present in **both** the SPI and the AOI datasets.
- **Output:**
  - The list of predicted **operator labels (Good or Bad).**
  - Each entry is a **tuple** in the form (Panel (p), Figure (f), Component (c), PredictedOperatorLabel (y))



Task 3: RepairLabel

- **Input**:
  - **SPI DataFrame**: a Pandas DataFrame as read from the CSV file using pd.read_csv()
  - **AOI and Operator Label:** a Pandas DataFrame that we read from the AOI CSV file using pd.read_csv(). We include only a subset of the columns. They are:
    - `"PanelID","FigureID","MachineID","ComponentID","PinNumber", "AOILabel", "OperatorLabel"`
    - **Note:** The only useful columns in the AOI dataset are **AOILabel** and **OperatorLabel**. These are the only additional feature available. The other columns must be used to join the **SPI** and the **AOI** dataframes.
- ***classification_3()*** reads a pre-trained model that you should store in the JupyterHub directory (or you can attach to your submission)
  - Predicts the repair label for the components present in **both** the SPI and the AOI datasets and for which the OperatorLabel is **bad**.
- **Output:**
  - The list of predicted **repair labels (NotPossibleToRepair** or **FalseScrap).**
  - Each entry is a **tuple** in the form (Panel (p), Figure (f), Component (c), PredictedRepairLabel (y))



PanelID	FigureID	Date	Time	ComponentID	PinNumber	PadID	…	Shape(um)	PosX(mm)	PosY(mm)	Result
263190880005201028 44	1.0	9/11/2019	00:07:04	BC1	1	1.0	…	0.0	55.6	23.6	GOOD
263190880005201028 44	1.0	9/11/2019	00:07:04	BC1	2	2.0	…	0.0	48.5	23.6	GOOD
263190880005201028 44	1.0	9/11/2019	00:07:04	BC2	1	3.0	…	0.0	13.4	23.6	GOOD
263190880005201028 44	1.0	9/11/2019	00:07:04	BC2	2	4.0	…	0.0	20.5	23.6	GOOD
263190880005201028 44	1.0	9/11/2019	00:07:04	BC3	1	5.0	…	0.0	55.6	45.6	GOOD
…	…	…	…	…	…	…	…	…	…	…	…
272190349005201028 44	8.0	7/29/2019	23:26:35	U5	6	3156.0	…	44.4	43.5	87.9	GOOD
272190349005201028 44	8.0	7/29/2019	23:26:35	U5	7	3157.0	…	46.7	43.5	86.6	GOOD
272190349005201028 44	8.0	7/29/2019	23:26:35	U5	8	3158.0	…	44.4	43.5	85.4	GOOD
272190349005201028 44	8.0	7/29/2019	23:26:35	Z1	1	3159.0	…	42.6	43.1	82.4	GOOD
272190349005201028 44	8.0	7/29/2019	23:26:35	Z1	2	3160.0	…	53.3	43.1	80.0	GOOD

+

PanelID	FigureID	ComponentID	AOILabel	OperatorLabel
263190822005201028 44	7	C22	Size	Bad
263190634005201028 44	6	L2	Translated	Bad

Model Performance Evaluation

The Notebook TestPerformance is provided to illustrate the performance evaluation process

RepairLabel: Test Classification

1. Task 1 runs to predict the AOI presence labels
   - Outputs the list of defects
2. Task 2 runs to predict the Operator labels
   - Outputs a list of tuples indicating the operator label of a component
3. Task 3 runs to predict the Repair labels
   - Outputs a list of tuples indicating the repair label of a component