HACKATHON

National Institute of Technology Puducherry

IntelliInspect: Real-Time Predictive Quality Control with Kaggle Instrumentation Data

TEAM 2 - MEMBERS

ASHWIN SANTHOSH

SANDEEP U

T VAISHNAVI SINGH

NIRMAL SASEENDRAN

MENTOR

SAI SUDHANE G

Table of Contents

Project Deliverables

☑ Must-Have Deliverables

📄 Design Documents
- System Architecture Diagram – architecture_diagram.md (In Github as markdown file)
- Data Flow – dataflow_architecture.md (In Github as markdown file)
- API Contracts – Documentation.pdf

📁 GitHub Repository
- Full-stack implementation with:
  - Angular Frontend
  - .NET Backend
  - Python ML Engine

📘 README
- Comprehensive setup and usage guide with:
  - Docker instructions
  - Developer onboarding

🐳 Dockerized Setup
- docker-compose.yaml for complete environment boot-up
- Runs all services (Frontend, Backend, ML API) in isolated containers

🎥 Demo Video (3 mins)
- Covers the end-to-end pipeline:
  - ✅ Dataset Upload
  - ✅ Training, Testing, Simulation Window Setup
  - ✅ Model Training Results
  - ✅ Real-Time Prediction Streaming

🌟 Good to Have – Implemented

🔁 Retry Simulation Logic
- Supports graceful recovery and task resumption in case of failure

---

✨ Nice to Have – Implemented

⚙ Large CSV Optimization
- Chunked CSV processing for high-performance handling of million-row files

🌓 Dark/Light Mode Toggle
- User-friendly UI theme toggle for accessibility and aesthetics

API contract and payload structure

Dotnet API features and payload structure

5.1.1 Dataset APIs

POST /api/Dataset/upload

Description: Upload a CSV dataset file.

- **Request**: multipart/form-data
  - file: (binary) The CSV file.
- **Response**: 200 OK

json

```
{
  "numberOfRecords": 0,
  "numberOfColumns": 0,
  "passRatePercentage": 0,
  "firstTimestamp": "2023-01-01T00:00:00",
  "lastTimestamp": "2023-12-31T23:59:59",
  "fileName": "example.csv",
  "fileSize": "1.2 MB"
}
```

- 400 Bad Request: Invalid file
- 500 Internal Server Error: Server issue

POST /api/Dataset/validate-ranges

Description: Validate date ranges for training, testing, and simulation.

- **Request**: application/json

json

```
{
  "trainStart": "2023-01-01T00:00:00",
  "trainEnd": "2023-06-30T23:59:59",
  "testStart": "2023-07-01T00:00:00",
  "testEnd": "2023-09-30T23:59:59",
  "simulationStart": "2023-10-01T00:00:00",
  "simulationEnd": "2023-12-31T23:59:59"
}
```

- **Response**: 200 OK

POST /api/Dataset/split-data

Description: Split uploaded dataset into training, testing, and simulation sets.

- **Request**: application/json (same schema as /validate-ranges)
- **Response**: 200 OK

5.1.2 Training APIs

POST /api/Training/start

Description: Start model training using defined date ranges.

- **Request**: No parameters
- **Response**: 200 OK

GET /api/Training/status/{task_id}

Description: Get the training status and metrics.

- **Path Parameter**:
  - task_id: (string) Task ID from training start.
- **Response**: 200 OK

5.1.3 Simulation APIs

POST /api/Simulation/start

Description: Start real-time inference simulation.

- **Request**: No parameters
- **Response**:

json

```
{
```

"task_id": "string"
}

POST /api/Simulation/stop/{task_id}

Description: Stop the running simulation task.

- **Path Parameter**:
  - task_id: (string)
- **Response**: 200 OK

GET /api/Simulation/status/{task_id}

Description: Fetch current simulation progress for a task.

- **Path Parameter**:
  - task_id: (string)
- **Response**: 200 OK

FastAPI features and payload structure

5.2.1. Dataset Ingestion & Feature Selection

POST /dataset/store

Purpose: Store Dataset and Initiate Feature Selection

Description

This endpoint accepts a CSV dataset file from the client and starts a **background feature selection process** asynchronously.
The server **immediately acknowledges** the request with a 202 Accepted response while continuing the heavy processing in the background.

Request Format

- **Method**: POST
- **Endpoint**: /dataset/store
- **Content Type**: multipart/form-data
- **URL Parameters**: *None*

- **Request Body**:

Field	Type	Required	Description
file	Binary	Yes	**The raw CSV file to be uploaded and processed**

✅ Successful Response (202 Accepted)

- **Content-Type**: application/json
- **Description**: Confirms that the dataset has been accepted and the background process has started.

Example Payload:

json

```
{
  "message": "Task accepted. Feature selection is running in the background.",
  "dataset_path": "/path/to/storage/data/full_dataset_with_ts.csv"
}
```

Field	Type	Description
message	string	Confirmation message for background processing
dataset_path	string	Internal path to where the dataset was stored

---

❌ Validation Error Response (422 Unprocessable Entity)

- **Occurs When**: The file is missing, not a valid CSV, or malformed
- **Content-Type**: application/json

Example Payload:

json

```
{
  "detail": [
    {
      "loc": ["body", "file"],
      "msg": "Invalid file type. Please upload a CSV file.",
      "type": "value_error.filetype"
    }
  ]
}
```

Error Object Fields:

Field	Type	Description
loc	array	Location of the issue (e.g., ["body", "file"])
msg	string	Human-readable error message
type	string	Machine-readable error code (e.g., value_error.filetype)

5.2.2. Data Splitting Endpoint

🔹 Endpoint

- **Method:** POST
- **URL:** /process/split-data

🔹 Purpose

Splits the master dataset into three subsets — **training**, **testing**, and **simulation** — based on the provided ISO 8601 date-time ranges.

- Filters an already uploaded dataset (no file or ID is passed in the payload).
- Saves each subset separately.
- Executes **synchronously** — client waits for completion.

🔹 Request

- **Content-Type:** application/json
- **Body Fields (all required):**

Field	Type	Description
train_start_date	string	Start of training period (ISO 8601)
train_end_date	string	End of training period (ISO 8601)
test_start_date	string	Start of testing period (ISO 8601)
test_end_date	string	End of testing period (ISO 8601)
simulation_start_date	string	Start of simulation period (ISO 8601)
simulation_end_date	string	End of simulation period (ISO 8601)

Example Payload

json
```
{
  "train_start_date": "2025-01-01T00:00:00Z",
  "train_end_date": "2025-05-31T23:59:59Z",
  "test_start_date": "2025-06-01T00:00:00Z",
  "test_end_date": "2025-06-30T23:59:59Z",
  "simulation_start_date": "2025-07-01T00:00:00Z",
  "simulation_end_date": "2025-07-31T23:59:59Z"
}
```

🔹 Successful Response

- **Status Code:** 200 OK
- **Content-Type:** application/json
- **Body:**

Field	Type	Description
message	string	Confirmation message
train_set_path	string	Path to saved training dataset
train_set_rows	integer	Row count in training dataset
test_set_path	string	Path to saved testing dataset
test_set_rows	integer	Row count in testing dataset
simulation_set_path	string	Path to saved simulation dataset
simulation_set_rows	integer	Row count in simulation dataset

Example Response

json


```
{
 "message": "Dataset successfully split and saved.",
 "train_set_path": "/path/to/storage/data/train_set_20250806.csv",
 "train_set_rows": 15000,
 "test_set_path": "/path/to/storage/data/test_set_20250806.csv",
 "test_set_rows": 3000,
 "simulation_set_path": "/path/to/storage/data/simulation_set_20250806.csv",
 "simulation_set_rows": 3100
}
```

🔹 Validation Error Response

- **Status Code:** 422 Unprocessable Entity
- **Content-Type:** application/json
- **Possible Validation Errors:**
  - Date format invalid
  - Start date after end date
  - Date ranges overlap
  - Date ranges fall outside dataset bounds

❌ Example Error

json


```
{
 "detail": [
   {
    "loc": ["body", "train_end_date"],
```

```
    "msg": "Train end date cannot be before train start date.",
    "type": "value_error.date.ordering"
   }
  ]
}
```

5.2.3.Start Model Training

🔹 Endpoint

- **Method:** POST
- **URL:** /process/train/start

🔹 Purpose

Initiates **asynchronous** model training using Celery.

- Responds immediately with a task ID.
- No input payload or parameters required.
- Client can use the returned task_id to track progress or results later.

🔹 Request

- **Content-Type:** application/json
- **Request Body:** *None*

🔹 Successful Response

- **Status Code:** 200 OK
- **Content-Type:** application/json
- **Body:**

Field	Type	Description
task_id	string	Unique identifier for the background task

✅ Example Response

json

```
{
```

"task_id": "ab1234cd-5678-efgh-9101-ijklmnopqrst"
}

5.2.4. Get Training Status

🔹 Endpoint

- **Method:** GET
- **URL:** /process/train/status/{task_id}

🔹 Purpose

Polls the status of a background training task using the task_id returned from /process/train/start.

- Returns:
  - Current task status (e.g., PENDING, IN_PROGRESS, COMPLETED, FAILED)
  - Progress updates (if any)
  - Final training result upon completion

🔹 Request

- **Path Parameter:**

Name	Type	Required	Description
task_id	string	✅	Unique ID of the training task

- **Request Body:** *None*
- **Content-Type:** application/json

🔹 Successful Response

- **Status Code:** 200 OK
- **Content-Type:** application/json
- **Body Fields:**

Field	Type	Description
task_id	string	ID of the task queried
status	string	Current task status (PENDING, IN_PROGRESS, COMPLETED, FAILED, etc.)
progress	object	Optional progress information (custom content)

Field	Type	Description
result	object	Final training result (available only when status === 'COMPLETED')

result.metrics Object

Metric	Type	Description
accuracy	number	Accuracy of the model
precision	number	Precision score
recall	number	Recall score
f1_score	number	F1 score

result.training_chart Object

Field	Type	Description
train_loss	array	List of loss values over epochs { x: string, y: number }
train_accuracy	array	List of accuracy values over epochs { x: string, y: number }

result.confusion_matrix Object

Field	Type	Description
true_positives	number	Count
false_positives	number	Count
true_negatives	number	Count
false_negatives	number	Count

result (additional fields)

Field	Type	Description
model_path	string	Path to saved model file
curves_path	string	Path to saved training curve image/chart

✅ Example Success Response

json


```
{
 "task_id": "ab1234cd-5678-efgh-9101-ijklmnopqrst",
 "status": "COMPLETED",
 "progress": {},
 "result": {
  "metrics": {
   "accuracy": 0.91,
   "precision": 0.89,
   "recall": 0.92,
```

```
    "f1_score": 0.905
   },
   "training_chart": {
    "train_loss": [
     { "x": "Epoch 1", "y": 0.8 },
     { "x": "Epoch 2", "y": 0.5 }
    ],
    "train_accuracy": [
     { "x": "Epoch 1", "y": 0.65 },
     { "x": "Epoch 2", "y": 0.91 }
    ]
   },
   "confusion_matrix": {
    "true_positives": 150,
    "false_positives": 10,
    "true_negatives": 130,
    "false_negatives": 15
   },
   "model_path": "/models/model_20250806.pkl",
   "curves_path": "/charts/curve_20250806.png"
  }
}
```

🔹 Validation Error Response

- **Status Code:** 422 Unprocessable Entity
- **Content-Type:** application/json
- **Reason:** Invalid task_id format or missing task

❌ Example Error Response

json


```
{
 "detail": [
  {
   "loc": ["path", "task_id"],
   "msg": "Task ID not found",
   "type": "value_error.task_id"
  }
 ]
}
```


5.2.5. Start Real-Time Simulation

🔹 Endpoint

- **Method:** POST
- **URL:** /simulation/start

🔹 Purpose

Triggers real-time inference simulation in the background.

- Runs asynchronously via a Celery task.
- Responds immediately with a task_id that can be used to monitor the simulation progress or fetch results.

🔹 Request

- **Parameters:** *None*
- **Request Body:** *None*
- **Content-Type:** application/json

🔹 Successful Response

- **Status Code:** 200 OK
- **Content-Type:** application/json

Response Fields

Field	Type	Description
task_id	string	ID of the background simulation task

✅ Example Success Response

json

```
{
  "task_id": "sim-task-4572cba1-e9e2-4df3-a123-21d56789abcd"
}
```


5.2.6. Stop Real-Time Simulation

🔹 Endpoint

- **Method:** POST
- **URL:** /simulation/stop/{task_id}

🔹 Purpose

Stops a running real-time simulation task identified by its task_id.

🔹 Request

- **Path Parameter:**

Name	Type	Required	Description
task_id	string	✅ Yes	The ID of the simulation task to stop

- **Request Body:** *None*
- **Content-Type:** application/json

🔹 Successful Response

- **Status Code:** 200 OK
- **Content-Type:** application/json

Response Fields

Field	Type	Description
task_id	string	The ID of the stopped task
message	string	Human-readable confirmation message

✅ Example Success Response

json

```
{
  "task_id": "sim-task-4572cba1-e9e2-4df3-a123-21d56789abcd",
  "message": "Simulation task stopped successfully."
}
```

🔹 Validation Error

- **Status Code:** 422 Unprocessable Entity
- **Content-Type:** application/json

❌ Example Error Response

json

```
{
 "detail": [
   {
    "loc": ["path", "task_id"],
    "msg": "Invalid task ID format.",
    "type": "value_error"
   }
 ]
}
```


5.2.7.Get Simulation Status

🔹 Endpoint

- **Method:** GET
- **URL:** /simulation/status/{task_id}

🔹 Purpose

Polls for the **status** and **live data** of an ongoing real-time simulation task, including current prediction, quality score, and statistics.

🔹 Request

- **Path Parameter:**

Name	Type	Required	Description
task_id	string	✅ Yes	The ID of the simulation task to check

- **Request Body:** *None*
- **Content-Type:** application/json

🔹 Successful Response

- **Status Code:** 200 OK
- **Content-Type:** application/json

Response Fields

Field	Type	Description
task_id	string	The ID of the simulation task

Field	Type	Description
status	string	Current status of the simulation task
progress	object	Contains live progress and predictions
result	object	Final result (empty or populated when completed)

- **progress object includes:**

Field	Type	Description
current_row_index	int	Index of the current row being simulated
total_rows	int	Total number of rows in the dataset
quality_score	float	Current quality score based on predictions
live_prediction	object	Latest prediction with confidence & features
live_stats	object	Summary stats of predictions

- **live_prediction object includes:**

Field	Type	Description
timestamp	string (ISO 8601)	Time of prediction
sample_id	string	Unique ID of the sample
prediction	string	Predicted class or output
confidence	float	Confidence of prediction (0 to 1)
top_features	object	Key-value map of feature importances

- **live_stats object includes:**

Field	Type	Description
total_predictions	int	Number of predictions made so far
pass_count	int	Count of "pass" predictions
fail_count	int	Count of "fail" predictions
average_confidence	float	Mean confidence over all predictions

✅ Example Success Response

json


```
{
 "task_id": "sim-task-912be4a1-48e7-4821-bff7-14f43de7abcd",
 "status": "running",
 "progress": {
  "current_row_index": 1280,
  "total_rows": 10000,
  "quality_score": 0.842,
  "live_prediction": {
   "timestamp": "2025-08-06T20:29:50.249Z",
   "sample_id": "sample_1280",
```

```
    "prediction": "pass",
    "confidence": 0.93,
    "top_features": {
      "feature_1": 0.25,
      "feature_3": 0.21,
      "feature_5": 0.19
    }
   },
   "live_stats": {
    "total_predictions": 1280,
    "pass_count": 950,
    "fail_count": 330,
    "average_confidence": 0.88
   }
  },
  "result": {}
}
```

🔹 Validation Error

- **Status Code:** 422 Unprocessable Entity
- **Content-Type:** application/json

❌ Example Error Response


```
json
{
  "detail": [
    {
      "loc": ["path", "task_id"],
      "msg": "Invalid task ID format.",
      "type": "value_error"
    }
  ]
}
```


Dotent Schemas

6.1.1.Shared Schemas

Dataset Metadata


```
json
{
  "numberOfRecords": 0,
  "numberOfColumns": 0,
  "passRatePercentage": 0,
  "firstTimestamp": "string",
  "lastTimestamp": "string",
  "fileName": "string",
  "fileSize": "string"
}
```

6.1.2DateRanges

```
json
{
  "trainStart": "string",
  "trainEnd": "string",
  "testStart": "string",
  "testEnd": "string",
  "simulationStart": "string",
  "simulationEnd": "string"
}
```

FastApi Schemas

6.2.1. Common & Core Schemas

These are foundational schemas used in multiple places across the API, primarily for handling errors and basic data structures.

ChartDataPoint

A single data point for use in plotting charts, consisting of an X and a Y coordinate.

Field Name	Data Type	Description
x	any	The value for the X-axis (e.g., an epoch number or timestamp).
y	number	The value for the Y-axis (e.g., a loss or accuracy metric).

HTTPValidationError

The standard response schema when a validation error occurs.

Field Name	Data Type	Description
detail	array of ValidationError	A list containing one or more specific validation error objects.

ValidationError

Provides specific details about a single validation error.

Field Name	Data Type	Description
loc	array of (string | integer)	The location of the error in the request (e.g., ["body", "field_name"]).
msg	string	A human-readable message explaining the error.
type	string	A machine-readable code for the type of error (e.g., value_error).

6.2.2. Dataset Ingestion Schemas

Schemas used for uploading a dataset and starting the feature selection process. *(Corresponds to POST /dataset/store)*

DatasetStoreRequest

The request body for uploading the dataset. *(This corresponds to the Body_store_dataset_and_select_features_dataset_store_post schema)*.

This is a multipart/form-data request, not a JSON object. It contains a single part:

- **file**: The CSV file being uploaded.

TaskAcceptedResponse

The successful response after a long-running task has been accepted.

Field Name	Data Type	Description
message	string	A confirmation message that the task is running in the background.
dataset_path	string	The internal server path where the dataset was stored.

6.2.3. Data Splitting Schemas

Schemas for defining how the master dataset is split into train, test, and simulation sets. *(Corresponds to POST /process/split-data)*

DateSplitRequest

The request payload defining the date ranges for splitting the dataset.

Field Name	Data Type	Description
train_start_date	string (date-time)	The inclusive start date for the training set (ISO 8601 format).
train_end_date	string (date-time)	The inclusive end date for the training set (ISO 8601 format).
test_start_date	string (date-time)	The inclusive start date for the test set (ISO 8601 format).
test_end_date	string (date-time)	The inclusive end date for the test set (ISO 8601 format).
simulation_start_date	string (date-time)	The inclusive start date for the simulation set (ISO 8601 format).
simulation_end_date	string (date-time)	The inclusive end date for the simulation set (ISO 8601 format).

DataSplitResponse

The successful response after the data has been split and saved.

Field Name	Data Type	Description
message	string	A confirmation message of the successful split.
train_set_path	string	The server path to the saved training set file.
train_set_rows	integer	The number of rows in the training set.
test_set_path	string	The server path to the saved test set file.
test_set_rows	integer	The number of rows in the test set.
simulation_set_path	string	The server path to the saved simulation set file.
simulation_set_rows	integer	The number of rows in the simulation set.

6.2.4. Model Training Schemas

Schemas for starting, monitoring, and retrieving results from the model training process.
(Corresponds to POST /process/train/start and GET /process/train/status/{id})

TrainingStartResponse

The response received immediately after starting a training job.

Field Name	Data Type	Description
task_id	string	The unique ID for the background training task.

TrainingStatusResponse

The comprehensive response from polling the training status endpoint.

Field Name	Data Type	Description
task_id	string	The ID of the training task being queried.
status	string	The current state of the task (e.g., IN_PROGRESS, SUCCESS).
progress	object | null	An object containing real-time progress (structure may vary).
result	TrainingResult | null	The final results, populated only when the task is complete.

TrainingResult

The detailed results of a completed training job.

Field Name	Data Type	Description
metrics	Metrics	An object containing key performance metrics of the model.
training_chart	TrainingChart	Data points for plotting training loss and accuracy charts.
confusion_matrix	ConfusionMatrix	The confusion matrix values from the model's test run.
model_path	string	The server path to the saved, serialized model file.
curves_path	string	The server path to saved chart data like ROC/PR curves.

Metrics

A collection of standard classification model performance metrics.

Field Name	Data Type	Description
accuracy	number	The accuracy score of the model.
precision	number	The precision score of the model.
recall	number	The recall score of the model.
f1_score	number	The F1-score of the model.

TrainingChart

Contains arrays of data points for plotting the training process.

Field Name	**Data Type**	**Description**
train_loss	array of ChartDataPoint	A list of loss values for each epoch/step.
train_accuracy	array of ChartDataPoint	A list of accuracy values for each epoch/step.

ConfusionMatrix

The four components of a confusion matrix for a binary classification task.

Field Name	**Data Type**	**Description**
true_positives	integer	The count of true positives.
false_positives	integer	The count of false positives.
true_negatives	integer	The count of true negatives.
false_negatives	integer	The count of false negatives.

6.2.5. Real-Time Simulation Schemas

Schemas for starting, stopping, and monitoring the real-time simulation. *(Corresponds to /simulation/start, /simulation/stop/{id}, and /simulation/status/{id})*

SimulationStartResponse

The response received immediately after starting a simulation.

Field Name	Data Type	Description
task_id	string	The unique ID for the background simulation task.

SimulationStopResponse

The response received after successfully sending a stop signal to a simulation.

Field Name	Data Type	Description
task_id	string	The ID of the task that was signaled to stop.
message	string	A confirmation that the stop signal was sent.

SimulationStatusResponse

The comprehensive response from polling the simulation status endpoint.

Field Name	Data Type	Description
task_id	string	The ID of the simulation task being queried.
status	string	The current state of the task (e.g., IN_PROGRESS, SUCCESS).
progress	SimulationProgress | null	A rich object with live data, populated only when the task is running.
result	object | null	The final summary results, populated only when the task is complete.

SimulationProgress

Contains live data and statistics about a running simulation.

Field Name	Data Type	Description
current_row_index	integer	The index of the data sample currently being processed.

Field Name	Data Type	Description
total_rows	integer	The total number of samples in the simulation dataset.
quality_score	number	A running score indicating the performance of the simulation.
live_prediction	LivePredictionData	An object containing details of the most recent prediction.
live_stats	LiveStatistics	An object containing cumulative statistics for the simulation.

LivePredictionData

Detailed information about a single, real-time prediction.

Field Name	Data Type	Description
timestamp	string (date-time)	The timestamp of when the prediction was made.
sample_id	string	The unique ID of the data sample being predicted.
prediction	string	The prediction output from the model (e.g., "PASS").
confidence	number	The model's confidence score for the prediction.
top_features	object	A dictionary of key-value pairs showing the most influential features.

LiveStatistics

A collection of cumulative statistics for the running simulation.

Field Name	Data Type	Description
total_predictions	integer	The total number of predictions made so far.
pass_count	integer	The cumulative count of "PASS" predictions.
fail_count	integer	The cumulative count of "FAIL" predictions.
average_confidence	number	The running average of the model's confidence scores.