ML Module Workflow

Dataset

* Supported Formats - .csv, .xlsx
* Validate File Type and Size before Upload.
* Actions:
  + Upload a Dataset
  + Create a Dataset
  + Preview Dataset
* Backend Tasks
  + Validate uploaded dataset (check format consistency, missing data)
  + Store datasets securely with version control.

Dataset Analysis and Pre-Processing (Pre-Processing Team)

* Features Expected:
  + Basic EDA
    - Display Summary Statistics
    - Correlation matrix for numerical features
    - Visualisation
  + Pre-Processing Options
    - Data Cleaning
    - Feature Engineering
    - Splitting Dataset into Training, Validation and Testing.

Model Type Selection

* Supervised Learning
  + Classification – select Target Column (must be Categorical)
  + Regression – select Target Column (must be numerical)
* Unsupervised Learning
  + Clustering – Specify the number of Clusters
  + Dimensionality Reduction
* Backend Tasks:
  + Dynamically Update UI components based on the selected type (of model)
  + Validate target column compatibility (data type checks)

Model Selection

* Actions:
  + Display a catalog (or a dropdown menu – refer the design) of available models. (categorised by the user’s choice between supervised/ unsupervised learning)
  + Provide a description of the model (Optional)
  + Model recommendations based on the dataset characteristics (Optional)
* Backend Tasks:
  + Dynamically update the UI to load the list of models (based on the selected option)

Parameter Configuration

* Actions:
  + General Parameters – For All the models
    - Training Epochs, batch size, learning rate
    - Cross validation splits
    - Random state
  + Model-Specific Parameters – depends on the respective model
    - Display a form/ dropdown (refer the design again) that displays the options for the selected model
    - Provide tooltips to explain each parameter (Optional)
  + Parameter Search Options
    - Grid search/ Random search for Hyper Parameter tuning.
    - Default vs user-defined ranges for parameter tuning. (Every parameter must have a pre-defined value as default, and the user will be able to change it per their requirement)
* Backend Tasks:
  + Validate input ranges for parameters (if there’s an option for the user to input – not for selecting options)
  + Implement a mechanism to save and reuse parameter configurations.

Training and Evaluation

* Actions:
  + Training:
    - Display a Progress bar (or something similar) for real-time feedback.
    - Log metrics (accuracy, loss etc) for each epoch.
  + Evaluation:
    - Generate Evaluation metrics based on model type:
      * Classification - Accuracy, Precision, Recall, F1-Score, ROC-AUC.
      * Regression: Mean Absolute Error, Mean Squared Error, R².
    - Model Performance Comparison:
      * Users should be able to compare multiple models on the same dataset.
  + Backend Tasks:
    - Log training and evaluation history for audit purposes/ future reference.

Visualization

* Actions:
  + Pre-Model Visualisation (contact Visualisation Team)
    - Feature Importance Graphs
    - Correlation heat maps
    - Custom visualisations
  + Model Specific Visualisations (Post-Model Visualisation)
    - Classification: Confusion matrix, ROC curve, Precision-Recall curve.
    - Regression: Residual plots, Predicted vs. Actual plots.
    - Clustering: Scatter plots (if 2D/3D), Silhouette score.
  + Provide options to filter, zoom, and download visualizations (user should be able to download the visualizations separately as pdfs/ images as well as be able to download them at the final stage along with the other metrics)
* Backend Tasks:
  + Ensure Plots are consistent across devices (Responsiveness) and all the options are available across devices.

Model Export

* Actions:
  + Export Trained models in the form of .pkl or .h5 for reuse
  + Provide code snippets in python, JS
* Backend Tasks:
  + Ensure model security during deployment



