Artificial Intelligence (AI) and Machine Learning (ML) Bootcamp

Level 5 Final Project – EDA Plan

Project Overview

I have chosen to develop a machine learning system for robot execution failure prediction and classification. This project addresses a real-world problem in robotics where early failure detection can prevent costly downtime and equipment damage. The project combines data science with practical hardware implementation, creating a comprehensive learning resource for developers new to AI and ML.

Business Question: Can machine learning models accurately classify different types of robot execution failures from force and torque sensor data, enabling proactive maintenance and safety improvements?

Section 1: Project Selection and Business Question Definition

What is the task?

I need to clearly define the business problem I'm addressing and establish the research questions that will guide my project. This involves researching the robotics industry, understanding common failure modes, and identifying how ML can provide value.

How long will it take?

Week 1 (2 days)

I'll spend the first two days researching robotics failure detection, reviewing existing literature, and refining my business question. I'll document the problem statement, research questions, and expected outcomes.

Why am I doing it?

A well-defined business question is essential because it guides every decision I make throughout the project. Without a clear objective, I could end up building

models that don't address real needs. This foundation ensures my work has practical value and demonstrates understanding of the business context.

What's my alternative if it doesn't work out?

If I struggle to define a clear business question initially, I'll start with a broader problem statement and refine it as I explore the data. I can also consult with my assessor or review similar projects to understand what makes a good research question. The key is to have something defined by the end of Week 1 so I can proceed with data selection.

Section 2: Dataset Selection and Ethical Considerations

What is the task?

I need to identify and obtain a suitable dataset for robot failure prediction. The dataset must contain sensor data from robot operations with failure labels. I'll also need to document the dataset's source, structure, and any ethical considerations around its use.

How long will it take?

Week 1 (1 day)

I'll research available datasets, focusing on the UCI Machine Learning Repository which has a Robot Execution Failures dataset. I'll download the data, examine its structure, and document any ethical considerations such as data privacy, consent, and appropriate use.

Why am I doing it?

The dataset is the foundation of my entire project. I need data that's:

- Publicly available and shareable (for the learning resource aspect)
- Contains sufficient examples for meaningful analysis
- Has clear failure labels for supervised learning
- Represents real-world scenarios

The UCI Robot Execution Failures dataset is perfect because it's well-documented, publicly available, and contains 463 samples with 16 unique failure types.

Although small, it provides enough variety for meaningful supervised learning while remaining manageable for a Level 5 project.

What's my alternative if it doesn't work out?

If the UCI dataset proves insufficient or unavailable, I have backup options:

- Kaggle has several robotics datasets I could explore
- I could use synthetic data generation combined with real patterns
- I could collect my own data from the robot arm I'm building (though this would be limited)

The key is ensuring I have data by the end of Week 1 so I can begin EDA in Week 2.

Section 3: Project Methodology and Tools Selection

What is the task?

I need to plan my technical approach, selecting the appropriate tools, libraries, and methodologies for the project. This includes choosing Python as my primary language, selecting ML libraries (scikit-learn, XGBoost), visualisation tools (matplotlib, seaborn), and planning my development environment.

How long will it take?

Week 1 (ongoing alongside other tasks)

I'll research and document my methodology choices, set up my development environment (Jupyter Notebooks), and create a project structure. I'll also plan which models to explore and justify my selections.

Why am I doing it?

Having a clear methodology ensures I work efficiently and make informed decisions. For this project, I'm choosing:

- **Python** - Industry standard, extensive ML libraries
- **Jupyter Notebooks** - Perfect for exploratory work and creating a learning resource
- **scikit-learn** - Comprehensive ML toolkit with consistent API
- **XGBoost** - Advanced ensemble method for comparison

- **pandas/numpy** - Essential for data manipulation
- **matplotlib/seaborn** - Professional visualisations

These tools let me cover both the basics and some more advanced techniques, which suits the learning-resource requirement.

What's my alternative if it doesn't work out?

If certain libraries prove problematic, I have fallbacks:

- If XGBoost installation fails, I'll focus on scikit-learn's GradientBoostingClassifier
- If visualisation libraries have issues, I can use simpler matplotlib-only approaches
- If Jupyter has problems, I can use Python scripts with detailed comments

The core scikit-learn library is stable and sufficient for the project requirements.

Section 4: Exploratory Data Analysis (EDA) Planning and Execution

What is the task?

I need to conduct comprehensive EDA to understand my dataset's characteristics. This includes examining data distributions, identifying patterns, checking for missing values, analysing correlations, and visualising relationships between features and failure types.

How long will it take?

Week 2 (5 days)

I'll spend most of Week 2 on EDA:

- Data loading, initial inspection, basic statistics
- Univariate analysis - distributions of individual features
- Bivariate analysis - correlations, relationships between features
- Class distribution analysis, identifying class imbalance
- Feature importance exploration, documenting key insights

Why am I doing it?

Thorough EDA is crucial because:

- It reveals data quality issues early
- It informs my preprocessing decisions (e.g., do I need scaling?)
- It helps me understand class imbalance (affects model selection)
- It identifies which features are most important (guides feature engineering)
- It provides insights I'll reference when justifying model choices

The EDA section in my notebook will be particularly valuable for developers learning ML, as it shows the thought process behind data science decisions.

What's my alternative if it doesn't work out?

If EDA reveals major issues:

- **Severe class imbalance:** I'll document it and use techniques like class weighting or focus on metrics that handle imbalance well
- **Missing data:** I'll implement appropriate imputation strategies
- **Data quality issues:** I'll clean the data and document the process
- **Insufficient data:** I'll acknowledge limitations and focus on what can be learned

I'll make sure to explain each step clearly so another learner could follow the process.

Section 5: Data Preprocessing and Feature Engineering

What is the task?

I need to prepare my data for machine learning by applying appropriate preprocessing techniques. This includes label encoding for categorical failure types, feature scaling for algorithms that require it, train-test splitting, and potentially feature engineering based on EDA insights.

How long will it take?

Week 2 (2 days)

I'll implement preprocessing pipelines:

- Label encoding, train-test split with stratification
- Feature scaling (StandardScaler), creating separate scaled datasets for models that need them

Why am I doing it?

Proper preprocessing is essential for model performance:

- **Label encoding** converts text labels to numbers (required by ML algorithms)
- **Stratified splitting** maintains class distribution in train/test sets (important given class imbalance)
- **Feature scaling** is required for algorithms like SVM and neural networks
- **Separate scaled/unscaled datasets** allow me to use the right data for each model type

This demonstrates understanding of different model requirements, which is valuable for the learning resource.

What's my alternative if it doesn't work out?

If preprocessing reveals issues:

- **Encoding problems:** I'll use one-hot encoding instead if needed
- **Scaling issues:** I'll try different scalers (MinMaxScaler, RobustScaler)
- **Split problems:** I'll adjust the split ratio or use cross-validation

The preprocessing code will be well-documented so developers can understand each step.

Section 6: Model Selection and Building

What is the task?

I need to select, implement, and train multiple machine learning models to compare their performance. Based on my EDA findings (class imbalance, non-linear relationships), I'll choose a diverse set of algorithms including tree-based models, ensemble methods, and linear models.

How long will it take?

Week 3 (4 days)

I'll implement and train models:

- Simple baseline models (Logistic Regression, Decision Tree, Naive Bayes)
- Tree-based ensemble models (Random Forest, Gradient Boosting, XGBoost)

- Advanced models (SVM variants, k-NN, and optionally a simple MLP for comparison)
- Stacking ensemble (combining best models)

Why am I doing it?

Training multiple models allows me to:

- Compare different approaches and their strengths/weaknesses
- Demonstrate understanding of when to use different algorithms
- Find the best-performing model for my specific problem
- Show the iterative nature of ML development

I'm particularly interested in ensemble methods because my EDA showed class imbalance and complex relationships - ensembles often handle these well. The Stacking Classifier combines multiple models, which should give good performance.

What's my alternative if it doesn't work out?

If certain models fail or perform poorly:

- **XGBoost issues:** I'll use scikit-learn's GradientBoostingClassifier instead
- **Neural network problems:** I'll simplify the architecture or skip it entirely
- **SVM too slow:** I'll use linear SVM or reduce the dataset size for testing
- **All models poor:** I'll revisit preprocessing, try feature engineering, or acknowledge dataset limitations

The goal is to have at least 3-4 working models for comparison, which demonstrates understanding of different approaches.

Section 7: Model Evaluation and Refinement

What is the task?

I need to evaluate all my models using appropriate metrics, compare their performance, identify the best model, and potentially refine it through hyperparameter tuning. I'll use metrics like accuracy, precision, recall, F1-score, and confusion matrices.

How long will it take?

Week 3 (3 days)

I'll conduct comprehensive evaluation:

- Calculate metrics for all models, create comparison visualisations
- Detailed analysis of best model (confusion matrix, per-class performance, error analysis)
- Hyperparameter tuning for the best model, final refinement

Why am I doing it?

Thorough evaluation demonstrates:

- Understanding of different evaluation metrics and when to use them
- Ability to interpret results and identify model weaknesses
- Skills in model refinement and optimisation
- Critical thinking about model performance

Given the class imbalance I identified in EDA, I'll pay particular attention to per-class metrics and use weighted averages. The confusion matrix will show which failure types are most confused, which is valuable insight.

What's my alternative if it doesn't work out?

If evaluation reveals poor performance:

- **Low accuracy:** I'll analyse why (class imbalance? insufficient data? wrong model choice?) and document limitations
- **Specific classes failing:** I'll investigate those classes, potentially using class weighting or focusing on important classes
- **Overfitting:** I'll use cross-validation, reduce model complexity, or increase regularization

Even if performance isn't perfect, documenting the analysis and limitations is valuable for a learning resource.

Section 8: Robot Arm Hardware Development and Integration

What is the task?

I need to research, plan, acquire, and assemble a physical robot arm to demonstrate the practical application of my ML findings. This includes 3D printing components, ordering electronics, assembling the arm, and programming it to work with my ML-identified patterns.

How long will it take?

Weeks 1-3 (ongoing, approximately 8-10 days total, spread across weeks)

Hardware development spans Weeks 1-3, with specific tasks allocated based on when materials arrive and when printing completes:

Week 1:

- Research robot arm designs, select EezyBotArm Mk2 design, download STL files

Week 2:

- Order 3D printing materials and electronics components (servos, Arduino, PCA9685, LCD) - account for 2-3 day shipping
- Begin 3D printing components (base, main arm, horizontal arm, gripper) - approximately 2-3 days printing time

Week 3:

- Assemble robot arm mechanical components, test fit and movement
- Wire electronics (Arduino, PCA9685, servos, LCD), test basic servo control
- Program Arduino with basic servo control, test each servo individually
- Integrate LCD display, implement demonstration system based on ML-identified patterns using rule-based logic inspired by my model's insights

Why am I doing it?

The physical robot arm demonstration:

- Shows practical application of ML findings in a tangible way
- Makes the project more engaging and memorable for developers learning ML
- Demonstrates the integration architecture from my notebook (sensor data → ML model → robot response)
- Provides a unique differentiator for my project

The hardware component also shows I can bridge the gap between data science and practical implementation, which is valuable in industry.

What's my alternative if it doesn't work out?

If hardware development faces issues:

- **3D printing fails:** I'll use a pre-made robot arm kit or simplify to fewer components
- **Electronics don't arrive in time:** I'll focus on software simulation and document the planned hardware integration
- **Assembly problems:** I'll simplify the design, use fewer servos, or focus on a working subset
- **Programming difficulties:** I'll use simpler Arduino code, focus on LCD display of patterns rather than full movement control

The key is having a working demonstration, even if simplified. The ML analysis is the core deliverable; the hardware is enhancement.

Section 9: ML Model Integration and Demonstration Development

What is the task?

I need to create a demonstration system that shows my ML findings in action. This involves adapting my ML model's insights into a rule-based system for the robot arm, displaying failure types on the LCD, and creating a workflow that demonstrates the integration architecture from my notebook.

How long will it take?

Week 4 (3 days)

I'll develop the integration:

- Adapt ML findings (key features, failure types) into Arduino code structure
- Implement demonstration system that shows ML-identified failure patterns using rule-based logic
- Test and refine the demonstration, ensure LCD displays work correctly

Why am I doing it?

This integration:

- Demonstrates practical application of ML findings
- Shows how theoretical analysis translates to real-world systems
- Creates an engaging demonstration for the learning resource
- Validates the "Robotics Application" section from my notebook

The demonstration will show movements based on patterns identified by ML analysis (like high-risk angles or problematic servo combinations) and display relevant failure categories, making the ML findings tangible.

What's my alternative if it doesn't work out?

If integration proves challenging:

- **Full ML model too complex for Arduino:** I'll use simplified rule-based detection based on ML-identified patterns (key features, thresholds) - which is what I'm actually doing anyway
- **Real-time prediction issues:** I'll use pre-computed patterns or simulated demonstrations
- **Hardware limitations:** I'll focus on LCD display of failure categories rather than full robot control

The demonstration doesn't need to run the full ML model on Arduino - showing ML-informed behaviour through rule-based logic is sufficient and more realistic.

Section 10: Final Report Writing and Documentation

What is the task?

I need to write a comprehensive final report that documents my entire project, from initial planning through to results and conclusions. The report should be tailored for developers new to ML, explaining concepts clearly while demonstrating technical competence.

How long will it take?

Week 4 (4 days)

I'll structure the report writing:

- Write introduction, methodology, and data description sections

- Document EDA findings, model building process, and evaluation results
- Write discussion, conclusions, and robotics application sections
- Final editing, formatting, references, and submission preparation

Why am I doing it?

The final report:

- Demonstrates my ability to communicate technical work clearly
- Shows understanding of the entire ML workflow
- Serves as the primary deliverable for assessment
- Creates a learning resource for developers (vocational scenario)

I'll structure it to match the assignment requirements: introduction, data description, EDA findings, methodology, results, discussion, and conclusions. I'll also include the robotics application section showing practical implementation.

What's my alternative if it doesn't work out?

If report writing faces challenges:

- **Too much content:** I'll focus on key findings and most important sections, use appendices for detailed code
- **Time constraints:** I'll prioritise results and discussion sections, ensure all required elements are covered
- **Formatting issues:** I'll use a simple, clear structure - the content matters more than fancy formatting

The Jupyter notebook itself serves as detailed documentation, so the report can be more focused and narrative.

Timeline Summary

Week 1 (13th - 19th November) - 3 days available

- Project selection, business question definition
- Dataset selection, ethical considerations
- Methodology and tools selection
- Robot arm research, design selection, STL file preparation

Week 2 (20th - 26th November) - 7 days

- Comprehensive EDA (data loading, univariate/bivariate analysis, class distribution, feature importance) - 5 days
- Data preprocessing (encoding, scaling, train-test split) - 2 days
- Order electronics and materials (accounting for 2-3 day shipping)
- Begin 3D printing robot arm components (2-3 days printing time)

Week 3 (27th November - 3rd December) - 7 days

- Model building (baseline models, ensemble methods, advanced models, stacking) - 4 days
- Model evaluation and refinement (metrics, analysis, hyperparameter tuning) - 3 days
- Robot arm mechanical assembly
- Electronics wiring and basic testing
- Arduino programming and servo control
- LCD integration and demonstration system implementation

Week 4 (4th December - Submission Deadline) - 4 days

- ML integration and demonstration development - 3 days
- Final report writing, editing, and submission - 4 days (overlapping with integration work)

Risk Assessment and Contingency Planning

Technical Risks

Risk 1: Dataset proves insufficient

- **Mitigation:** I've verified the UCI dataset is available and suitable before starting
- **Contingency:** Use synthetic data generation or focus on available classes

Risk 2: Model performance is poor

- **Mitigation:** I'm using multiple models and ensemble methods known to work well

- **Contingency:** Document limitations, focus on analysis quality over perfect accuracy

Risk 3: Hardware delivery delays

- **Mitigation:** Ordering early in Week 2, allowing 2-3 day shipping buffer
- **Contingency:** Focus on software demonstration, document planned hardware integration

Risk 4: 3D printing failures

- **Mitigation:** Start printing early and prepare alternative components if needed
- **Contingency:** Simplify design, use fewer components, or use pre-made kit

Time Management Risks

Risk 5: EDA takes longer than planned

- **Mitigation:** Focused 5-day EDA block, well-defined tasks
- **Contingency:** Prioritise most important analyses, document others briefly

Risk 6: Report writing rushed

- **Mitigation:** Starting report in Week 4 with 4 full days allocated
- **Contingency:** Use notebook as detailed documentation, report as executive summary

---

Expected Outcomes and Success Criteria

Primary Outcomes

I. A comprehensive Jupyter notebook demonstrating end-to-end ML workflow
II. A trained model achieving reasonable accuracy (target: >60% given class imbalance)
III. A working robot arm demonstration showing ML findings in practice
IV. A well-structured final report suitable for developers learning ML

Success Criteria

- **Pass:** All deliverables completed, basic ML workflow demonstrated, report covers all required sections

- **Merit:** Thorough EDA, multiple models compared, good evaluation, well-organised report
- **Distinction:** Comprehensive analysis, advanced models, excellent evaluation, professional presentation, innovative hardware integration

---

Ethical Considerations

Data Privacy

- The UCI Robot Execution Failures dataset is publicly available and contains no personal information
- All data consists of anonymised sensor readings from robot operations
- No ethical approval required for use of this dataset

Academic Integrity

- All code will be my own work, with libraries and methodologies properly cited
- Dataset source will be clearly documented
- Any inspiration from other projects will be acknowledged

Professional Standards

- The project will be suitable for educational use
- Safety considerations for robot arm operation will be documented
- Limitations and assumptions will be clearly stated

Conclusion

This project plan provides a clear roadmap for completing a comprehensive AI/ML project that combines theoretical machine learning analysis with practical hardware implementation. The timeline is realistic, accounts for potential delays, and ensures all deliverables can be completed by the 4th December deadline. The project will serve as both an assessment submission and a valuable learning resource for developers new to AI and ML.