LCCS 2026 Coursework – Modelling Wildfire Risk

1. What Are We Modelling?

For this project, we are focusing on **forests** and the **dangers they may face.** We refer to these dangers as **environmental risks**. Forests face environmental risks including:

- wildfire
- Flooding
- Landslides
- Pests
- Disease (Diseases kill trees).

Key Point: We model the **risk** of an event occurring, **not** the event itself

Risk = the chance or likelihood that an event could occur under certain conditions

Basic Requirements: Microbit {Collecting Data}

Advanced Requirements: Model

- Create a computer model to estimate risk level for a chosen threat (e.g., wildfire)
- Support decision-making by identifying when action might be needed
- The model does **NOT** predict **definite** outcomes
- Instead, it explores what **could** happen under different conditions using '**what-if**' scenarios

Example: What if temperature increased by 3°C and humidity decreased by 5%?

A model estimates how risk level might change and therefore **supports** mitigation planning

Basic Requirements

2.a Choosing Input Data (Variables) BR1

Task: To model anything, e.g., wildfire risk, we **collect** environmental **data that influences how fires can start or spread.** These measurements are inputs to the model.

- Use **at least one analogue input** e.g., temperature, humidity, soil moisture
- Use **at least one digital input** e.g., a button to start or calibrate the system)
- Produce **at least one output** e.g., LEDs, display, sound, or message
- Operate **automatically** once started

Marker Check {Replace with Tick ONLY if Completed} ☑

☐ Use at least one **analogue input** (e.g., temperature, humidity, soil moisture)

☐ Use at least one **digital input** (e.g., a button to start or calibrate the system)

☐ Produce at least one **output** (e.g., LEDs, display, sound, or message)

☐ Operate **automatically** once started

Note: Segment 1 of Video {Approx. 45 Seconds}: Video Check List

Purpose of the System

☐ **Show on video:** The complete embedded system powered on, with the micro:bit, **sensors**, and output visible.

☐ **Say:** Briefly state (approx. 2 sentences) what **data** your embedded system is **designed to gather {name sensors}**

Digital Input

☐ **Show on video:** The **button** or switch visible, and **activate** it to start or calibrate the system.

☐ **Say:** Identify the digital input used (e.g. button or switch) and explain that it is used to start or calibrate the system.

Calibrate: Using a button or switch to set the system's starting conditions or baseline before it runs automatically e.g., maybe a **reset** button of some kind.

Automatic Operation

☐ **Show on video:** Show the system continuing to run with no user interaction after button is pressed.

☐ **Say:** Explain that after starting, the system runs automatically without further user input.

Analogue Input

☐ **Show on video:** The analogue sensor e.g., **temp** and a visible change to its input e.g., **warming the sensor**

☐ **Say:** Identify the analogue sensor used (e.g. temperature, light, soil moisture) and state what **range of values** it produces. {an examiner might not know the range of values of a soil moisture sensor}.

Primary Output

☐ **Show on video:** The primary output clearly active (LED, sounder, or display).

☐ **Say:** Identify the main output used (digital <u>or</u> analogue) and explain **what the output is showing** when the system is running. This will primarily be the values from the sensor. You can add an alert if it goes 'above' a certain level that you define as 'dangerous'

MakeCode Program {or any other code used here}

☐ **Show on video:** The MakeCode editor with the program visible, focusing on the main blocks (start, input reading, rules, output).

☐ **Say:** Explain {your code} briefly - that the program uses simple rules to read inputs and control the output.

2.b Raw Data Collection BR2

Your Task: Your embedded system is used to **collect and store environmental data**. This data is **raw numerical data** e.g., daily environmental readings. You need to:

- **collect at least one** of these but you may collect more.

  e.g., Day 1: Temp 18°C, Humidity 65%, Wind 10 km/h, Soil Moisture 42%

- **validate** this data e.g., checking data is in correct **format** <u>**or**</u> within **expected range**
- **clean** this data {e.g., replacing missing values etc.}
- **store** this data e.g., CSV
- **data types** e.g. int, float, string

Marker Check {Replace with Tick ONLY if Completed} ☑

☐ **Collect** raw environmental data (at least one type, may collect more)

e.g., Day 1: Temp 18°C, Humidity 65%, Wind 10 km/h, Soil Moisture 42%

☐ **Validate** e.g., checking data is in correct **format** <u>**or**</u> within **expected range**

☐ **Clean** the data {e.g., replacing missing values etc.}

☐ **Store** the data e.g., CSV

☐ **Data Types** e.g. int, float, string

Note: Segment 2 of Video {Approx. 45 Seconds}

Purpose of Data Collection

☐ **Show on video:** The embedded system powered on with the micro:bit and the data-collection sensor visible.

☐ **Say:** Briefly state **what data your sensor is collecting**

Collect Raw Data

☐ **Show on video:** Live sensor readings being taken (LED display, serial monitor, or data output).

☐ **Say:** Explain that the system collects raw environmental data directly from the sensor.

Validate Data

☐ **Show on video:** The data validation that you are performing.

☐ **Say:** Explain what type of data validation you are performing e.g., - is data in **int** format.

- range of acceptable data

Clean Data

☐ **Show on video:** How you cleaned the data

☐ **Say:** Explain how you cleaned the data.

Store Data

☐ **Show on video:** Evidence of data being saved (CSV file} or any other database.

☐ **Say:** Briefly describe the data being **stored** and the data 'type', e.g., int, float, string

**** Identify which input data you will be using in your model**

2.c Process Simulation Using Rules BR3

Task: You must use your embedded system to:

- **simulate** a **real-world process** related to wildfire risk by showing how the system changes behaviour when environmental conditions change.
- demonstrate THREE different test conditions (safe/normal, moderate, extreme)

Example: Simulating Temperature Change

- The embedded system measures **temperature** using a sensor.
- A **hair dryer** (set to cool or low heat, held at a distance) is used to gently increase the air temperature near the sensor.
- As the temperature rises, **the system automatically changes its output**:

Marker Check {Replace with Tick ONLY if Completed} ☑

☐ System simulates a **real-world process** related to the theme (e.g., temperature changes, soil moisture levels, drought conditions)

☐ **THREE different test conditions** demonstrated (safe/normal, moderate, extreme)

☐ System **output changes** appropriately as environmental conditions change

☐ Tests show how system **responds** to different inputs or changing environmental data

Difference between Digital and Analog Output

Digital Output: Cooler air → 1 LED on
Warmer air → 2 LEDs on
Hotter air → 3 LEDs on

or

Analog Output: Gradually increasing sound volume to show temperature **changing smoothly over time**:

Note: This simulation in BR3 is **not** a rules-based risk model. {We address this in Advanced Requirements}

Note: Segment 3 of Video {Approx. 45 Seconds}

Simulation

☐ **Show on video:** What real-world process you are simulating e.g., soil moisture, temp, etc

☐ **Say:** Explain the real-world process you're simulating and its relevance to your solution {project}.

Test 1

☐ **Show on video:** The system running while environmental input values are set to safe conditions, and the output displaying a low or safe status.

Note: **You will need to artificially simulate conditions e.g., a hair dryer….**

Display: **One** LED is turned on to indicate a safe display. **{Digital}**

Display: The needle/servo pointer moves to the low end of a scale (e.g. 0–2 on a 0–10 dial) to show a safe/low status. **{Analog}**

☐ **Say:**
- What conditions they are using (safe)
- What the output shows
- Explain how the output makes sense for those conditions {name} being simulated

Test 2

☐ **Show on video:** The system running while environmental input values are changed to **moderate** conditions, and the output displaying a medium or caution status.

Note: **You will need to artificially simulate conditions e.g., a hair dryer….**

Display: **Two** LEDs are turned on to indicate a moderate or caution display. **{Digital}**

Display: The needle/servo pointer moves to the middle of the scale to indicate a medium or caution status. **{Analogue}**

☐ **Say:**
- What conditions they are using (moderate)
- What the output shows
- Explain how the output makes sense for those conditions {name} being simulated

Test 3

☐ **Show on video:** The system running while environmental input values are changed to more **extreme** conditions, and the output displaying a high or warning status.

Note: **You will need to artificially simulate conditions e.g., a hair dryer….**

Display: **Three** LEDs are turned on to indicate a high or warning display. **{Digital}**

Display: The needle/servo pointer moves to the **high end** of the scale to indicate a high or dangerous status. **{Analogue}**

☐ **Say:**

- What conditions they are using (extreme)
- What the output shows
- Explain how the output makes sense for those conditions {name} being simulated

Advanced requirement 1

4. Build Model

Your Task: Using Python, develop a computer model of a chosen disaster risk scenario related to forests {wildfire, drought, pest outbreak, air quality, landslides, flooding}.

The brief is **intentionally flexible**. You could use other **model types** such as:

- Rules-based scoring system (with justified thresholds) [My Approach]
- Weighted scoring systems
- Statistical regression models
- Machine learning approaches
- Probability-based models

EXAMPLE APPROACH 1:

Rules- Based Scoring Method: This example shows **ONE** way to build a risk model.

This involves making **justified assumptions BASED** on **real-world research**.

Note: Reference this research also.

Requirements:

- Model must be written in **Python** addresses a **disaster risk scenario** related to forests
- Model must use SOME data collected from your embedded system
- Model can be combined with open-source data OR simulated data
- Model must calculate or estimate risk levels for the chosen scenario
- **Research justification** provided for modeling approach

Marker Check {Replace with Tick ONLY if Completed} ☑

☐ Computer model in **Python** addresses a **disaster risk scenario** related to forests

☐ Model uses some data from the embedded system

☐ Model combined with **open-source** OR **simulated data**

☐ Model **calculates/estimates risk levels**

☐ **Research justification** provided for modeling approach

How Risk Scores Are Decided

Risk scores are calculated using simple **rules based on research**.
For each set of input values, the model applies these rules to produce a **single risk level**.

Data Types: - Numerical Data e.g., Temperature 18(°C)
- Ordinal Data e.g., Wet, Dry etc.

Step 1: **Data is ready {includes Microbit Data AND data downloaded/generated}**
**** Reminder:** **What data will you collect using the Microbit?**

Step 2: **Set a 'Risk Scale'. Here is an example:**

Key
0–2 points → Risk Level 1 (Lo)
3–5 points → Risk Level 2 (Med)
6–8 points → Risk Level 3 (Hi)

Step 3: **Our data does NOT come with a risk level**

Reading	Temperature (°C)	Humidity (%)	Soil Moisture	Wind Speed	Risk Level
A	18	65	Wet	Low	
B	22	55	Damp	Low	
C	25	45	Dry	Med	
D	28	35	Dry	Med	
E	32	25	Very Dry	High	

Risk Levels: **How DO we arrive at these actual Risk Levels?**

Risk levels are **NOT** given in the data. They are **calculated** by **applying scoring rules** {**rule or assumptions based on real world-research**} to each set of readings and converting the total score into a risk level.

Risk Scoring Rules and Risk Level Scale

Scoring Rules: **Based on real-world research and justified assumptions**
The Higher the Score = The Greater the Perceived Risk

Risk Level: **The final value calculated by applying the scoring rules to the data**

Step 3a: Scoring Rules {Justified assumptions BASED on real-world research}			
Temperature: 18°C ≤ 25°C Score: +0 26–30°C Score: +1 >30°C Score: +2	**Humidity:** 65% ≥ 50% Score: +0 30–49% Score: +1 < 30% Score: +2	**Soil Moisture:** Wet Wet Score: +0 Damp Score: +1 Dry/Vr. Dry Score: +2	**Wind Speed:** Low Low: Score +0 Med: Score +1 High: Score +2

Step 3b: Add up Total Score:	**Step 3c: Convert Total Points to Risk Level**
Row A: Temp 18, Humidity 65, Soil Wet, Wind Low Temperature contributes +0 Humidity contributes +0 Soil contributes +0 Wind contributes +0 **Total Score = 0 points**	0-2 points → Risk Level 1 (Lo) 3-5 points → Risk Level 2 (Med) 6-8 points → Risk Level 3 (Hi) **Final Risk Level: 1**

4: Research Justification [WHY we chose these specific {scoring} rules] For Video

Here is an example ONLY for Temperature

Temperature rule
According to the Irish Forest Service (2023), wildfire risk increases significantly above 25°C. Based on this research, the model assigns higher risk scores as temperature rises beyond this value.
[Source: https://www.coillte.ie/our-forests/forest-protection/]

Video Evidence: ALL of this is PRIORITY EVIDENCE FOR YOU VIDEO

Approach 2: Data-Driven Method of deriving the RULES

Students with access to historical data that already includes risk classifications may use statistical analysis to discover patterns and derive scoring rules.

This approach requires:
- Historical data with risk levels already assigned
- Statistical analysis skills (correlation, regression)
- Python data analysis libraries (pandas, matplotlib, scipy)

Note: This is an advanced alternative. Most students will use the Rules-Based Method (Approach 1) shown above.

AR1: **Note:** Segment 4 of Video {Approx. 50 Seconds}

Disaster Risk Model Overview

☐ **Show on video:** The Python code editor with your model visible.

☐ **Say:** State **what disaster risk** you are modelling (e.g., wildfire, drought, flooding) and briefly explain the purpose of your model.

Data Sources

☐ **Show on video:** The data file(s) being used - show both the **microbit data** AND any **open-source** or simulated data.

☐ **Say:** Explain **what data your model** uses. Identify which data came from your **embedded** system and which came from **other sources** (open-source or simulated).

Model Calculation Process

☐ **Show on video:** The section of code that performs the risk calculation.

☐ **Say:** Explain step-by-step how your model calculates risk:

- What inputs go in

- How the calculation works (e.g., scoring rules, formulas)

- What output comes out

Risk Output

☐ **Show on video:** Run the model and show the risk level output being displayed (e.g., 'Risk Level 2' or 'Medium Risk').

☐ **Say:** Explain what the risk level output means and what values it can produce.

Research Justification

☐ **Show on video:** Display or reference the research source that justifies your modeling approach.

☐ **Say:** Briefly explain why you chose this modeling approach and cite at least one research source that **supports your scoring rules** or calculation method.

Example: 'According to the Irish Forest Service, wildfire risk increases significantly above 25°C, which is why my model assigns higher scores when temperature exceeds this threshold.'

Advanced requirement 2: **'what-if' scenario Testing**

Requirements

- **TWO** 'what-if' scenario simulations created and tested
- Each scenario involves **changing one or more key variables**
- **Observe** the resulting **changes** in system behaviour or predicted risk

Marker Check {Replace with Tick ONLY if Completed}☑

☐ TWO **'what-if'** scenario simulations created and tested

☐ Each scenario involves **changing one** or **more** key variables

☐ Results show changes in system behaviour or predicted risk

Recommended Approach: Use as baseline for comparison

1. Establish a **baseline** representing **normal** or **low-risk** conditions {This could be hardcoded in your model <u>or</u> set as initial input values}

Analogy: Imagine you want to lose weight, say 2 kg in a month. But, in a month's time when you weigh yourself, **you can't tell if you gained or lost weight unless you know your starting weight (baseline).**

```
BASELINE: Low Risk Conditions
Example Baseline:  Temperature: 22°C, Humidity: 55%, Soil: damp, Wind: low
Total Points: 1 | Risk Level: 1
```

2. **YOU choose test scenarios** (e.g., What if temp rises to 35°C?)

```
WHAT-IF TEST #1
Temperature (°C): 34
Humidity (%): 12
Soil (wet/damp/dry/very dry): dry
Wind (low/moderate/high): high
```

3. **Compare to baseline** (Did risk increase, decrease, or stay the same?

```
RESULT
Total Points: 8 | Risk Level: 3
Compared to baseline: Risk INCREASED.
```

Hint! Take Screenshots AND Video of Model Working as you go. {Splice Together and Voiceover Later}

AR 2: What-If Scenario Testing

Note: Segment 5 of Video {Approx. 50Seconds}

Baseline Conditions

☐ **Show on video:** Your model running with baseline/normal conditions displayed on screen.

☐ **Say:** State what your baseline conditions are (e.g., "Temp 22°C, Humidity 55%, Soil damp, Wind low") and what risk level they produce.

Example: 'My baseline represents normal forest conditions with a risk level of 1.'

What-If Test 1

☐ **Show on video:** The model running with changed variables for your first what-if scenario.

☐ **Say:** - State which variables you changed (e.g., 'What if temperature rises to 34°C and humidity drops to 12%?')

- Show the new risk level output
- Compare to baseline (e.g., "Risk increased from level 1 to level 3")
- Briefly explain why this change makes sense based on your model

What-If Test 2

☐ **Show on video:** The model running with changed variables for your second what-if scenario.

☐ **Say:** - State which variables you changed (e.g., "What if soil moisture becomes very dry but temperature stays moderate?")

- Show the new risk level output
- Compare to baseline (e.g., "Risk increased from level 1 to level 2")
- Briefly explain why this change makes sense based on your model

Comparison Summary

☐ **Show on video:** A summary view showing baseline and both what-if scenarios side by side (table, **Bar** chart etc., or text output).

☐ **Say:** Briefly summarize what both scenarios demonstrated about how environmental conditions affect risk.

Advanced requirement 3 Adaptive System 'feedback' mechanism.

Your Task: Extend your system to include a feedback mechanism that enables it to adapt automatically to changing conditions. The adaptive response can be either physical {extending the embedded system} or virtual {changing the model so that it adjusts automatically}.

Requirements:

- Include a feedback mechanism
- System adapts automatically to changing conditions
- Can be EITHER physical (embedded system) OR virtual (model)

Marker Check {Replace with Tick ONLY if Completed} ☑

☐ Feedback mechanism included in the system
☐ System adapts automatically to changing conditions
☐ Adaptive response is clearly demonstrated (physical OR virtual)

Recommended Approach: Virtual Adaptation (Python Model)

What Makes AR3 'Adaptive'?

1. **Reads new data automatically** (from CSV file)
2. **Remembers the last risk level** (so it can compare)
3. **Tells you what changed** (Risk went UP or Risk went DOWN)

```
NEW ROW OF DATA READ -> MODEL UPDATED
Time: Day 1 09:00
Temp: 22 Humidity: 55 Soil: damp Wind: low
Total Points: 1 | Risk Level: 1
Risk Comparison with Previous: Risk stayed the same ( 1 )

NEW ROW OF DATA READ -> MODEL UPDATED
Time: Day 1 12:00
Temp: 25 Humidity: 48 Soil: damp Wind: moderate
Total Points: 3 | Risk Level: 2
Risk Comparison with Previous: Risk increased from 1 to 2
```

Note: In AR3 of the brief, I **am 'adjusting predictions based on the most recent data'**.

Your CSV Should Have Changes

Make sure your risk goes up and down so the model can demonstrate adaptation.

e.g. Example Progression: `Risk increased from 1 to 2`

Hint! Take Screenshots AND Video of Model Working as you go. {Splice Together and Voiceover Later}

AR 3: Adaptive System Feedback Mechanism

Note: Segment 6 of Video {Approx. 50 Seconds}

Feedback Mechanism Overview

☐ **Show on video:** The system component that provides feedback - either physical (embedded system with adaptive output) OR virtual (Python model that adapts).

☐ **Say:** **Explain** whether your adaptive system is **physical** {microbit} or **virtual** {model} and describe what feedback mechanism you implemented.

Physical Adaptation Example: 'My embedded system triggers a red LED alert when risk exceeds level 2'

Virtual Adaptation Example: 'My Python model automatically reads new data and compares risk levels to previous readings'

Demonstrating Initial State

☐ **Show on video:** The system in **its initial state** with **first set of conditions**.

☐ **Say:** State the initial conditions and risk level.

Example: 'Starting with temperature 22°C, humidity 55%, the risk level is 1.'

Demonstrating Automatic Adaptation

☐ **Show on video:** The system automatically detecting changed conditions and adapting its response.

☐ **Say:** Explain **what** changed and **how** the system adapted automatically.

Physical Example **'**As temperature increased, the system automatically changed from 1 LED to 3 LEDs without manual intervention.'

Virtual Example: **'**The model automatically read the next data row, recalculated risk, and reported 'Risk increased from 1 to 2.'

Showing Comparison/Feedback

☐ **Show on video:** Clear evidence of the system comparing current state to previous state.

☐ **Say:** Explain how the feedback mechanism enables the system to respond to changes.

Example: 'The model remembers the previous risk level and automatically tells us whether risk has increased, decreased, or stayed the same - this is the feedback mechanism that makes the system adaptive.'