Hazard Analysis
SFWRENG 4G06

Team 28, Cowvolution Minds
Aryan Patel
Harshpreet Chinjer
Krish Patel
Martin Ivanov
Shazim Rahman

Table 1: Revision History

Date	**Developer(s)**	**Change**
October 20th, 2024	Harsh Chinjer	Introduction, Scope and Purpose
Date2	Name(s)	Description of changes
...	...	...

Contents

[You are free to modify this template. —SS]

1 Introduction

This project involves developing a machine learning tool in collaboration with Cattleytics Inc., aimed at predicting key outcomes for dairy farming. The tool will help dairy farmers make informed decisions regarding breeding, herd management, and productivity by analyzing historical and genetic data. By predicting traits such as milk output, health conditions, and breeding success, the project strives to optimize herd performance and improve overall farm management.

A hazard in this context refers to any event, condition, or system failure that could result in negative consequences such as financial losses, breaches of ethical standards, or harm to livestock. Hazards can arise from various factors, including data inaccuracies, mispredictions from the model, technical malfunctions, or security breaches, all of which may impair decision-making. Since the system is designed to influence critical choices—like selecting sires or predicting future traits—it is vital to identify and address potential hazards to ensure the tool's reliability, ethical operation, and overall benefit to dairy farming.

2 Scope and Purpose of Hazard Analysis

The purpose of this hazard analysis is to systematically identify, assess, and mitigate potential risks associated with the machine learning model and its integration into the broader farm management system (Cattleytics). The primary scope includes the hazards that could affect the accuracy and reliability of the predictions, the security of sensitive farmer and animal data, and the ethical handling of dairy cattle based on the recommendations generated by the tool.

The potential losses that could be incurred due to these hazards include:

- Financial Loss: If the model makes incorrect predictions regarding breeding success rates, milk production, or herd turnover, farmers could experience financial setbacks, such as poor breeding decisions leading to reduced milk yield or ineffective herd management.

- Reputation Damage: Incorrect or biased predictions could lead to poor decision-making, negatively affecting the credibility of both the tool and the farmers who rely on it. Additionally, improper handling of personal information could damage trust with stakeholders.

- Legal Repercussions: Non-compliance with the Code of Practice for the Care and Handling of Dairy Cattle or PIPEDA could result in legal action or fines for mishandling sensitive data or violating animal welfare standards.

- Ethical Concerns: A failure to comply with the ethical treatment standards of livestock could lead to the mistreatment of cattle, which would not only result in legal issues but also ethical concerns around animal welfare.

This hazard analysis aims to identify such risks early in the project, allowing for proactive strategies to minimize harm and ensure the safe, secure, and effective operation of the tool in dairy farming.

3 System Boundaries and Components

[Dividing the system into components will help you brainstorm the hazards. You shouldn't do a full design of the components, just get a feel for the major ones. For projects that involve hardware, the components will typically include each individual piece of hardware. If your software will have a database, or an important library, these are also potential components. —SS]

4 Critical Assumptions

[These assumptions that are made about the software or system. You should minimize the number of assumptions that remove potential hazards. For instance, you could assume a part will never fail, but it is generally better to include this potential failure mode. —SS]

5 Failure Mode and Effect Analysis

[Include your FMEA table here. This is the most important part of this document. —SS] [The safety requirements in the table do not have to have the prefix SR. The most important thing is to show traceability to your SRS. You might trace to requirements you have already written, or you might need to add new requirements. —SS] [If no safety requirement can be devised, other mitigation strategies can be entered in the table, including strategies involving providing additional documentation, and/or test cases. —SS]

6 Safety and Security Requirements

[Newly discovered requirements. These should also be added to the SRS. (A rationale design process how and why to fake it.) —SS]

7 Roadmap

[Which safety requirements will be implemented as part of the capstone timeline? Which requirements will be implemented in the future? —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?
4. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?