Group Project: Credit Application Governance Analysis

Data Ecosystems and Governance in Organizations (2606)

MSc Business Analytics | Nova SBE

Contents

1 Overview

Your team will act as a **Data Governance Task Force** at a fintech company called "NovaCred." You have been given a raw dataset of credit applications and must analyze it for data quality issues, detect algorithmic bias, and propose governance interventions.

1.1 Deliverables

1. **6-minute video presentation**
2. **GitHub repository** with all code, analysis notebooks, and documentation

1.2 Deadline

All deliverables must be submitted by **23:59 on the day before Session 6**. Late submissions will not be accepted.

2 The Scenario

NovaCred is a fintech startup that uses machine learning to make credit decisions. Recently, they received a regulatory inquiry about potential discrimination in their lending practices. Your team has been hired to:

1. **Audit the data** for quality issues
2. **Detect bias patterns** in historical decisions
3. **Propose governance controls** to prevent future issues
4. **Demonstrate compliance** with GDPR and AI Act requirements

3 The Dataset

File: `raw_credit_applications.json`

The dataset contains 500+ credit applications in nested JSON format. Each record has the following structure:

Warning

Intentional Issues: The dataset contains **intentional data quality issues and bias patterns** for you to discover. These include problems related to:

- Data completeness and consistency
- Data type mismatches
- Sensitive personal data handling
- Potential discrimination in lending decisions

Your job is to find them all!

Field	Type	Description
`_id`	String	Application ID (e.g., `app_001`)
applicant_info *(nested object)*		
`.full_name`	String	Applicant full name
`.email`	String	Email address
`.ssn`	String	Social Security Number
`.ip_address`	String	IP address at time of application
`.gender`	String	Gender
`.date_of_birth`	String	Date of birth
`.zip_code`	String	ZIP/postal code
financials *(nested object)*		
`.annual_income`	Number	Annual income
`.credit_history_months`	Integer	Months of credit history
`.debt_to_income`	Number	Debt-to-income ratio
`.savings_balance`	Number	Current savings balance
spending_behavior *(array of objects)*		
`[].category`	String	Spending category
`[].amount`	Number	Monthly spending amount
decision *(nested object)*		
`.loan_approved`	Boolean	`true` or `false`
`.interest_rate`	Number	Assigned rate (if approved)
`.approved_amount`	Number	Loan amount (if approved)
`.rejection_reason`	String	Reason (if denied)

Table 1: Dataset schema — note that this is a nested JSON document, not a flat table

Tip

Pay close attention to the *structure* of the data, not just the values. Some fields may behave unexpectedly, some records may be incomplete, and relationships between fields may reveal hidden patterns.

4 Team Roles

Each team has 4 members. Assign these roles:

Role	Responsibilities
Data Engineer	Data loading, cleaning, pipeline code, repository structure
Data Scientist	Bias analysis, fairness metrics, statistical testing
Governance Officer	GDPR mapping, policy recommendations, compliance analysis
Product Lead	Presentation, coordination, README documentation

Table 2: Team roles and responsibilities

Important

Important: All members must contribute to Git. We will verify commit history. Every team member must have meaningful commits — not just a single commit at the end.

5 Deliverables in Detail

5.1 Video Presentation (6 minutes)

Format: Pre-recorded video, MP4 format, max 100MB

Structure:

- Introduction and team (30 sec)
- Data quality findings (90 sec)
- Bias analysis results (90 sec)
- Governance recommendations (90 sec)
- Conclusion (30 sec)

Requirements:

- All team members must appear and speak
- Show key visualizations from your analysis
- Cite specific numbers from your analysis (e.g., "We found X duplicates," "DI ratio of Y")

5.2 Live Q&A (Session 6)

After each video is shown in class, a **randomly assigned group** will ask the presenting team a question about their work. This means every group participates twice in each Q&A round: once as the team answering and once (for a different presentation) as the team questioning.

How it works:

- The video is played in class
- The assigned questioning group has **1 minute** to formulate and ask a question
- The presenting team has **2 minutes** to answer
- The instructor may ask follow-up questions

What makes a good question:

- Targets a specific technical choice (e.g., "Why did you use mean imputation instead of dropping rows?")
- Challenges an assumption or finding with reasoning
- Connects to governance concepts from the course (e.g., "How would your pseudonymization approach handle GDPR Article 17 requests?")

What makes a good answer:

- Demonstrates genuine understanding of the code and methodology (not just reading from slides)
- Provides specific evidence or references to the analysis
- Acknowledges limitations honestly when applicable

Warning

The Q&A follows a structured format. The instructor directs questions to the **Product Lead**, who is responsible for answering on behalf of the team. The Product Lead may, at their discretion, invite a specific team member to elaborate on a technical detail or provide additional depth. It is also the **Product Lead** who poses questions to the instructor — after consulting with the rest of the team if needed. This structure mirrors real-world product ownership, where a single point of accountability coordinates both communication and clarification. The Q&A is designed to verify that **all team members** understand the work: if the Product Lead cannot explain code or analysis that the team supposedly contributed to, this will affect the team's Q&A score and may trigger a peer evaluation review.

5.3 GitHub Repository

Required structure:

```
project-teamX/
|-- README.md                    # Project overview & findings summary
|-- data/                        # Data files (or links)
|-- notebooks/                   # Analysis notebooks
|   |-- 01-data-quality.ipynb
|   |-- 02-bias-analysis.ipynb
|   +-- 03-privacy-demo.ipynb
|-- src/                         # Reusable code (optional)
|   +-- fairness_utils.py
+-- presentation/                # Video file or link
```

Requirements:

- Repository must have a **description** (the short text that appears under the repo name on GitHub — e.g., "DEGO 2606 Group Project – Credit Application Governance Analysis")
- Minimum 10 meaningful commits across the project
- All team members must have commits under their own GitHub accounts
- Clear, descriptive commit messages
- Working code that runs without errors
- A comprehensive `README.md` summarizing your findings and governance recommendations

Important

The `README.md` is your primary written deliverable. It should include an executive summary of findings, key metrics, and actionable governance recommendations. Think of it as the document your CTO would read.

6 Technical Requirements

6.1 Required Analyses

1. Data Quality Assessment:

- Identify and document all data quality issues (completeness, consistency, validity, accuracy)

- Quantify the extent of each issue (e.g., number of affected records, percentage)
- Propose and demonstrate remediation steps

2. Bias Detection — Disparate Impact Ratio for gender:

$$DI = \frac{\text{Approval rate of unprivileged group}}{\text{Approval rate of privileged group}}$$

A value below 0.8 indicates potential disparate impact (the "four-fifths rule").

3. Proxy Discrimination Analysis:

- Investigate whether any non-protected attributes serve as proxies for protected characteristics
- Analyze correlations between fields and outcomes

4. Privacy Demonstration:

- Identify all personally identifiable information (PII) in the dataset
- Show pseudonymization or anonymization of at least one PII column
- Map your findings to GDPR requirements (lawful basis, data minimization, storage limitation)

6.2 Recommended Libraries

```
# Data manipulation
import pandas as pd
import numpy as np

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Fairness (optional but recommended)
# pip install fairlearn
from fairlearn.metrics import demographic_parity_difference

# MongoDB (if using)
from pymongo import MongoClient
```

7 Timeline

Session	Milestone
Session 3	Teams announced, project introduced, dataset distributed
Session 4	GitHub repo created, all members onboarded, initial exploration
Session 5	Data analysis complete, start preparing presentation
Day before Session 6 (23:59)	**Deadline:** Video uploaded, repository finalized
Session 6	Presentations in class

Table 3: Project timeline

8 Submission Instructions

1. **GitHub Repository**
   - Repository must be **public** and include a **repository description**
   - Submit the repository URL via Moodle before the deadline
   - Final commit must be before **23:59 on the day before Session 6**
2. **Video**
   - Upload to YouTube (unlisted) OR include in repo (if under 100MB) OR provide a Google Drive link in the README

9 Frequently Asked Questions

Q: Can we use external data?
A: No. Use only the provided dataset.

Q: What if we cannot find all the bias patterns?
A: Document what you found and your methodology. Partial credit is given for a rigorous process, even if not all issues are identified.

Q: Can we use ChatGPT/Claude for code?
A: Yes, but you must understand and be able to explain every line of your code. The Q&A session will test this.

Q: What if one team member does not contribute?
A: Peer evaluation will be considered. Document individual contributions clearly in the README.

Q: What format should the README use?
A: Markdown. Include sections for executive summary, data quality findings, bias analysis, privacy assessment, and governance recommendations.

10 What to Look For

Use this checklist to guide your analysis. Not all items carry equal weight — focus on the issues with the greatest governance impact.

10.1 Data Quality Issues

☐ Duplicate records

☐ Inconsistent data types across records

☐ Missing or incomplete records

☐ Inconsistent coding/formatting of categorical fields

☐ Invalid or impossible values

☐ Inconsistent date formats

10.2 Bias Patterns

☐ Gender disparate impact (DI < 0.8)

☐ Age-based discrimination patterns

☐ Proxy variables for protected attributes

☐ Interaction effects between attributes

10.3 Privacy and Governance Gaps

☐ Sensitive PII stored without protection

☐ No consent tracking mechanism

☐ Missing data retention policy

☐ No audit trail for decisions

☐ Lack of human oversight documentation

☐ Sensitive behavioral data collection

11 Grading Rubric

The group project is worth **40%** of the final course grade. The project is evaluated across eight criteria. The total is 100 points, which is then scaled to the 40% course weight.

Criterion	**Points**	**Weight**
Data Quality Analysis	20	20%
Bias Detection & Fairness	20	20%
Privacy & Governance	15	15%
Code Quality & Repository	15	15%
Video Presentation	10	10%
Q&A — Answering	5	5%
Q&A — Questioning	5	5%
Git Collaboration	10	10%
Total	**100**	**100%**

Table 4: Grade distribution summary

Each criterion is evaluated on a four-level scale: **Excellent** (90–100%), **Good** (75–89%), **Satisfactory** (60–74%), and **Needs Work** (<60%).

11.1 Data Quality Analysis (20 points)

Level	Descriptor
Excellent (18–20 pts)	Identifies all or nearly all data quality issues: duplicate records, missing/incomplete records, inconsistent data types (e.g., income stored as string), inconsistent gender coding, inconsistent date formats, and invalid values (e.g., negative credit history months). Quantifies each issue with specific counts and percentages. Proposes and demonstrates remediation steps in code. Clearly maps issues to data quality dimensions (completeness, consistency, validity, accuracy).
Good (15–17 pts)	Identifies most data quality issues (at least 4 of 6 categories). Provides reasonable quantification for most issues found. Proposes remediation strategies but may not implement all of them in code. Shows understanding of data quality dimensions.
Satisfactory (12–14 pts)	Identifies some data quality issues (at least 2–3 categories) but misses significant ones. Limited quantification. Remediation steps are vague or incomplete. Basic understanding of data quality concepts demonstrated.
Needs Work (<12 pts)	Misses most data quality issues or incorrectly identifies non-issues. Little to no quantification. No meaningful remediation proposed. Demonstrates weak understanding of data quality concepts.

11.2 Bias Detection & Fairness (20 points)

Level	Descriptor
Excellent (18–20 pts)	Correctly calculates the disparate impact ratio for gender and interprets it against the four-fifths rule. Identifies proxy discrimination (e.g., geographic attributes correlating with protected characteristics). Investigates age-based bias patterns. Explores interaction effects (e.g., age combined with gender). Uses appropriate statistical methods or fairness libraries. Provides clear visualizations of bias patterns.
Good (15–17 pts)	Correctly calculates DI ratio and identifies at least one proxy variable. Shows gender-based and either age-based or geographic bias. Visualizations are present and generally clear. Minor errors in interpretation or methodology.
Satisfactory (12–14 pts)	Calculates DI ratio but with some errors or incomplete interpretation. Identifies gender bias but misses proxy discrimination or age patterns. Limited use of visualizations. Methodology is present but not rigorous.
Needs Work (<12 pts)	DI ratio is incorrect or missing. Fails to identify major bias patterns. No proxy analysis. Little or no statistical support for claims. Visualizations absent or misleading.

11.3 Privacy & Governance (15 points)

Level	Descriptor
Excellent (14–15 pts)	Identifies all PII fields in the dataset (names, emails, SSNs, IP addresses, dates of birth). Successfully demonstrates pseudonymization or anonymization of at least one field. Maps findings to specific GDPR articles (lawful basis, data minimization, storage limitation, right to erasure). References the EU AI Act classification for credit scoring systems. Proposes concrete, actionable governance controls (audit trails, human oversight, consent mechanisms, data retention policies).
Good (11–13 pts)	Identifies most PII fields. Demonstrates pseudonymization. References GDPR principles but may lack specificity on articles. Governance recommendations are reasonable but could be more detailed. Some mention of AI Act.
Satisfactory (9–10 pts)	Identifies some PII but misses critical fields (e.g., SSNs or IP addresses). Basic GDPR discussion without mapping to specific articles. Governance recommendations are generic. No pseudonymization demonstration or it is incomplete.
Needs Work (<9 pts)	Fails to identify most PII. No privacy demonstration. GDPR discussion is absent or incorrect. No meaningful governance recommendations.

11.4 Code Quality & Repository (15 points)

Level	Descriptor
Excellent (14–15 pts)	All notebooks run without errors. Code is clean, well-commented, and logically structured. Repository follows the required folder structure. README is comprehensive and well-written, serving as an effective executive summary of findings and recommendations. Reusable utility functions are appropriately separated.
Good (11–13 pts)	Code runs with minor issues or requires small fixes. Generally clean and commented. Repository structure is mostly followed. README covers key findings but may lack depth in some areas.
Satisfactory (9–10 pts)	Code runs but requires notable fixes or has unclear logic. Limited comments. Repository structure is partially followed. README is present but incomplete or lacks substantive analysis summary.
Needs Work (<9 pts)	Code does not run or contains critical errors. Poorly organized or uncommented. Repository structure not followed. README is missing or trivial.

11.5 Video Presentation (10 points)

Level	Descriptor
Excellent (9–10 pts)	Within the 6-minute limit. All team members speak. Presentation is well-structured and engaging. Key visualizations are shown and clearly explained. Specific numbers and metrics are cited. Demonstrates deep understanding of findings.

Level	Descriptor
Good (7–8 pts)	Close to the time limit (within 30 seconds). All members speak. Good structure and content. Most key findings are covered with supporting evidence. Minor issues with clarity or flow.
Satisfactory (5–6 pts)	Exceeds the time limit or significantly under. Most members speak. Content covers the main topics but lacks depth or specific evidence. Visualizations are present but not well-explained.
Needs Work (<5 pts)	Significantly over or under time. Not all members participate. Content is superficial, disorganized, or missing key sections. No visualizations or supporting data cited.

11.6 Q&A — Answering (5 points)

Level	Descriptor
Excellent (5 pts)	Answer demonstrates genuine understanding of the methodology and code. Responds with specific references to the analysis (e.g., exact metrics, design choices). Acknowledges limitations honestly. Any team member can explain any part of the project.
Good (4 pts)	Answer is correct and shows understanding. Provides some specific evidence but may be less precise. One or two team members dominate the response while others could have contributed.
Satisfactory (3 pts)	Answer is partially correct but vague or overly general. Shows surface-level understanding. Team struggles to provide supporting evidence for their claims.
Needs Work (<3 pts)	Cannot answer the question or provides an incorrect response. Team members cannot explain their own code or analysis. Suggests the work may not be genuinely understood.

11.7 Q&A — Questioning (5 points)

Level	Descriptor
Excellent (5 pts)	Question is specific, technical, and insightful. Targets a concrete methodological choice, assumption, or finding in the presented work. Demonstrates that the questioning group watched carefully and engaged critically with the content.
Good (4 pts)	Question is relevant and shows engagement with the presentation. Addresses a specific aspect of the work but may lack depth or could have been more targeted.
Satisfactory (3 pts)	Question is generic and could apply to any project (e.g., "What was the hardest part?"). Shows limited engagement with the specific content presented.
Needs Work (<3 pts)	No question asked, or question is off-topic, trivial, or clearly not based on having watched the presentation.

11.8 Git Collaboration (10 points)

Level	Descriptor
Excellent (9–10 pts)	10 or more meaningful commits. All team members have substantive commits under their own accounts. Commit messages are clear and descriptive. Repository has a description set. Commit history shows steady progress over time (not a single bulk upload).
Good (7–8 pts)	At least 8 commits. All members contributed, though distribution may be uneven. Most commit messages are descriptive. Work spread across multiple sessions.
Satisfactory (5–6 pts)	At least 5 commits. Most members contributed. Some commit messages are vague (e.g., "update," "fix"). Work may be concentrated in a short period.
Needs Work (<5 pts)	Fewer than 5 commits, or only 1–2 team members contributed. Commit messages are absent or meaningless. All work committed at the last minute in a single session.

12 Penalties

Item	Points
Repository not public at deadline	−5
Video exceeds 7 minutes	−3
Missing team member from video without justification	−5
Evidence of fabricated Git commits (e.g., empty commits to inflate count)	−10
Missing peer evaluation submission	−5

Table 13: Penalties

13 Peer Evaluation

Peer evaluation is **mandatory for all students**. Every team member must submit an individual, confidential evaluation of their teammates. This is not optional — failing to submit a peer evaluation will result in a −5 **point penalty** on the student's individual project grade.

13.1 Why Mandatory?

Making peer evaluation obligatory ensures the instructor receives complete and unbiased data. When only students with complaints submit, the process becomes punitive rather than informative. When everyone participates, it normalizes the process and provides a fair, complete picture of team dynamics.

13.2 What You Will Evaluate

For **each** of your teammates, you will rate the following on a scale of 1 (strongly disagree) to 5 (strongly agree):

1. **Contribution:** "This team member contributed a fair share of the work"

2. **Quality:** "This team member's contributions were of good quality"
3. **Reliability:** "This team member met internal deadlines and was dependable"
4. **Collaboration:** "This team member communicated well and was responsive"

You will also have one open-text field:

- *"Is there anything else the instructor should know about team dynamics or individual contributions?"* (optional)

13.3 How It Affects Grades

Peer evaluations may be used to adjust individual grades by up to ±20% of the group grade. In practice:

- If all teammates rate each other similarly (the normal case), no adjustment is made.
- If one member consistently receives low scores *and* this is corroborated by Git commit history, the instructor may reduce that student's grade.
- If one member is consistently rated as having contributed above and beyond, a small positive adjustment may be applied.

13.4 Moodle Submission

The peer evaluation will be set up as a **Moodle Feedback activity** in the course page. Details:

- The activity will appear under the Session 6 section on Moodle
- **Deadline:** 48 hours after Session 6
- Responses are **confidential but not anonymous** — the instructor can see who submitted each evaluation (to track completion and verify consistency), but your teammates will **never** see your individual ratings or comments
- The activity takes approximately 2–3 minutes to complete
- You must evaluate **all** teammates (not yourself)

Important

If there is a serious contribution imbalance, the instructor reserves the right to assign different grades to team members based on Git commit history and peer evaluations. Document individual contributions clearly in the README to support this process.

Good luck!