Session 3.1: Foundations of Artificial Intelligence: Enterprise Value Creation and Decision Automation

PEDAGOGICAL AGENDA

1:2 Delivery Rhythm

Session Learning Architecture

Structured 3-hour applied computing session designed for business students with zero prior coding experience.

- **Phase 1 (60 Min Lecture):** Computational foundations, business vocabulary, and syntactic mechanics.
- **Phase 2 (120 Min Lab):** Guided hands-on implementation, error diagnosis, and AI co-pilot resolution.
- **Zero Fluff Mandate:** Focus directly on executable Python code, tabular data wrangling, and commercial intelligence.

STRATEGIC OUTCOME

Operational Autonomy

Business Analytical Competency

Transforming passive spreadsheet users into autonomous programmatic decision makers.

- **Speed & Scale:** Processing datasets exceeding spreadsheet limits with instantaneous vectorized execution.
- **Repeatable Pipelines:** Replacing manual copy-paste workflows with fully automated Python scripts.
- **AI Debugging Literacy:** Leveraging modern IDEs to diagnose and resolve errors independently.

Core Computational Keywords & Business Meaning

CONCEPT 01 — Core Keyword

Artificial Intelligence (AI)

Broad computational systems capable of performing tasks that typically require human cognition, including perception, language understanding, and automated decision-making.

CONCEPT 02 — Core Keyword

Machine Learning (ML)

A subfield of artificial intelligence where statistical models learn patterns and relationships directly from data rather than following rigid, hand-coded rules.

CONCEPT 03 — Core Keyword

Deep Learning (DL)

A subset of machine learning based on multi-layered artificial neural networks, specialized in unstructured data processing (images, audio, natural language).

CONCEPT 04 — Core Keyword

Rule-Based vs. Statistical Systems

Transitioning from deterministic `if-then` heuristics to probabilistic algorithms that generalize across noisy, unseen real-world data.

Computational Foundations & Business Operations

PILLAR 01 Architecture

**Core Programming & Computational Foundations

**

PILLAR 02 Architecture

The Artificial Intelligence Hierarchy

Deconstructing industry hype. Machine Learning is not magic; it is applied computational statistics. AI encompasses the broader vision, ML provides data-driven statistical learning, and Deep Learning handles high-dimensional unstructured tasks.

PILLAR 03 Architecture

The Economics of Machine Learning (Prediction as a Commodity)

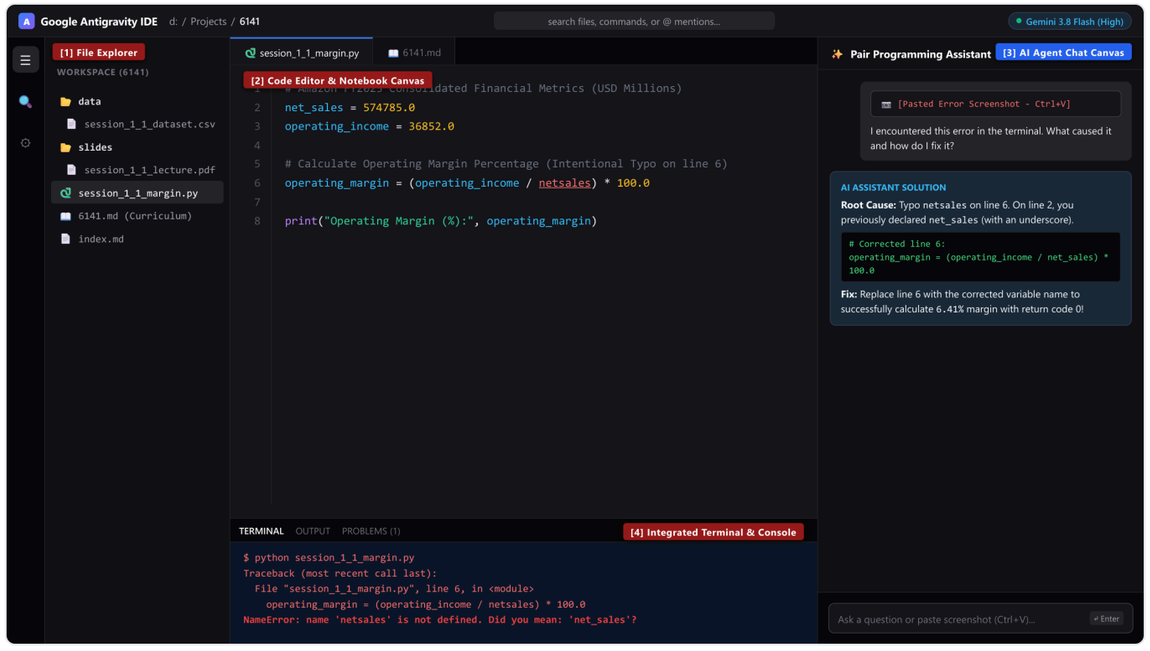
When the cost of quantitative prediction drops toward zero, prediction is incorporated into every operational decision: fraud screening, demand forecasting, inventory routing, and pricing.

PILLAR 04 Architecture

Enterprise ROI Evaluation Matrix

Evaluating AI projects requires rigorous financial modeling. Students quantify Implementation Cost (compute, labeling, salaries), Annual Labor Savings, and Projected Payback Periods.

Google Antigravity IDE Architecture & AI Debugging Protocol



[1] File Explorer

Project Hierarchy

Manage datasets in `data/`, scripts in root, and exported artifacts.

[2] Code Editor

Syntax Workspace

Modern syntax highlighting, lint diagnostics, and line-by-line script authoring.

[3] AI Agent Chat

AI Pair Programmer

Paste error screenshots directly to inspect root causes and get verified fixes.

[4] Terminal

Execution Engine

Run Python scripts (`python script.py`) and view console logs and tracebacks.

Zero-Friction AI Error Resolution Protocol

1. **Capture:** Press `Win + Shift + S` to clip terminal error traceback.
2. **Paste:** Press `Ctrl + V` into AI Chat [3] with "Explain root cause & fix".
3. **Apply:** Review the explanation and apply verified fix into Editor [2].

Script Demonstration 1: Script Demonstration 1

session_3_1_demo_1.py — Python 3.10

```
"kw">class="kw">import pandas "kw">class="kw">as pd

    df = pd."fn">read_csv("data/session_3_1_dataset.csv")

    df["Net_Annual_Savings_USD"] = df["Projected_Cost_Savings_USD"] - (df["Impl
    df["Calculated_Payback_Months"] = (df["Implementation_Cost_USD"] / df["Net_
    df["3Year_Net_ROI_Pct"] = (((df["Net_Annual_Savings_USD"] * 3.0) - df["Impl

    "fn">print(df[["Project_ID", "Department_Name", "Calculated_Payback_Months"
```

```
$ python session_3_1_demo_1.py [Exit 0]
```

1 Syntactic Structure

Importing production libraries and defining clean analytical data structures.

2 Vectorized Transformation

Applying deterministic operations across the full dataset without fragile cell references.

3 Console Verification

Emitting formatted business KPIs to the terminal for auditability and compliance.

Script Demonstration 2: Script Demonstration 2

session_3_1_demo_2.py — Python 3.10

```python
# Rule-Based Heuristic
    "kw">class="kw">def rule_based_flag(spend):
        "kw">class="kw">return "Flagged" "kw">class="kw">if spend > 250000.0 "
    # Probabilistic Scoring Equivalence
    "kw">class="kw">def statistical_score(spend, risk_score):
        prob = (spend / 500000.0) * 0.5 + (risk_score / 100.0) * 0.5
        "kw">class="kw">return "High_Risk" "kw">class="kw">if prob >= 0.65 "kw
    "fn">print("Rule-Based Evaluation:", rule_based_flag(260000.0))
    "fn">print("Probabilistic Evaluation:", statistical_score(260000.0, 75))
```

```
$ python session_3_1_demo_2.py [Exit 0]
```

1 Syntactic Structure

Importing production libraries and defining clean analytical data structures.

2 Vectorized Transformation

Applying deterministic operations across the full dataset without fragile cell references.

3 Console Verification

Emitting formatted business KPIs to the terminal for auditability and compliance.

Script Demonstration 3: Script Demonstration 3

session_3_1_demo_3.py — Python 3.10

```
high_priority = df[(df["Calculated_Payback_Months"] <= 6.0) & (df["Annual_Labor_
    "fn">print(f"High Priority AI Projects Count: {">len(high_priority)}")
    "fn">print(high_priority[["Project_ID", "Department_Name", "Calculated_Payb
```

```
$ python session_3_1_demo_3.py [Exit 0]
```

1 Syntactic Structure

Importing production libraries and defining clean analytical data structures.

2 Vectorized Transformation

Applying deterministic operations across the full dataset without fragile cell references.

3 Console Verification

Emitting formatted business KPIs to the terminal for auditability and compliance.

Interactive Two-Tier Lab Protocol & AI Diagnosis

EXERCISE 1

Guided Lab

Foundational Implementation

Step-by-step guided practice establishing baseline syntax mastery in Antigravity IDE.

- **Step 1:** Ingest dataset from `data/session_3_1_dataset.csv`.
- **Step 2:** Implement core analytical functions and compute primary business metrics.
- **Step 3:** Format console outputs and verify calculations against baseline ledger totals.
- **Step 4:** Run in Terminal [4] and confirm clean execution with exit code 0.

EXERCISE 2+

Coding Challenge

Advanced Scripting & AI Debugging

Applied programming challenge expanding pipeline logic and resolving intentional exceptions.

- **Task 1 (Pipeline Expansion):** Add secondary business unit logic and multi-tier calculations.
- **Task 2 (Deliberate Error & AI Capture):** Introduce intentional syntax or type error, capture traceback with `Win + Shift + S`, and diagnose via AI Chat [3].
- **Task 3 (Verified Production Run):** Apply verified fix, export audit output, and confirm zero errors with return code 0.

Interactive Lab Protocol & Delivery Rhythm

Hands-On Code Execution

The Pedagogical Contract

Computational mastery develops through active problem-solving. By creating intentional errors and using AI co-pilots to resolve them, students build lifelong debugging autonomy.