

Session 4.2: AI Applications in Business Decision-Making Part 1: Predictive Customer Analytics and Churn

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

Customer Churn Probability

The continuous probability output (`predict_proba`) generated by a classification model estimating the likelihood of a subscriber canceling services.

CONCEPT 02

Core Keyword

Customer Lifetime Value (CLV)

The present value of future net profit margins generated by a customer relationship across their projected relationship tenure.

CONCEPT 03

Core Keyword

Revenue At Risk

Quantifying the total annual revenue at risk by multiplying individual churn probabilities by annualized subscriber contract values.

CONCEPT 04

Core Keyword

Targeted Retention Incentives

Programmatically matching customer churn tiers to specific cost-effective intervention strategies (e.g., proactive customer support outreach vs. discount incentives).

Computational Foundations & Business Operations

PILLAR 01

Architecture

****Core Programming & Computational Foundations**

**

PILLAR 02

Architecture

The Economics of Retention vs. Acquisition

Acquiring a new customer costs 5 to 7 times more than retaining an existing one. Predictive churn analytics allows subscription businesses to intervene before high-value customers cancel.

PILLAR 03

Architecture

Translating Probability Scores into Financial Exposure

A raw churn probability of 0.85 means nothing to a CEO. Translating that score into ` \$1,800 annual revenue at risk ` allows executive leadership to evaluate retention budgets quantitatively.

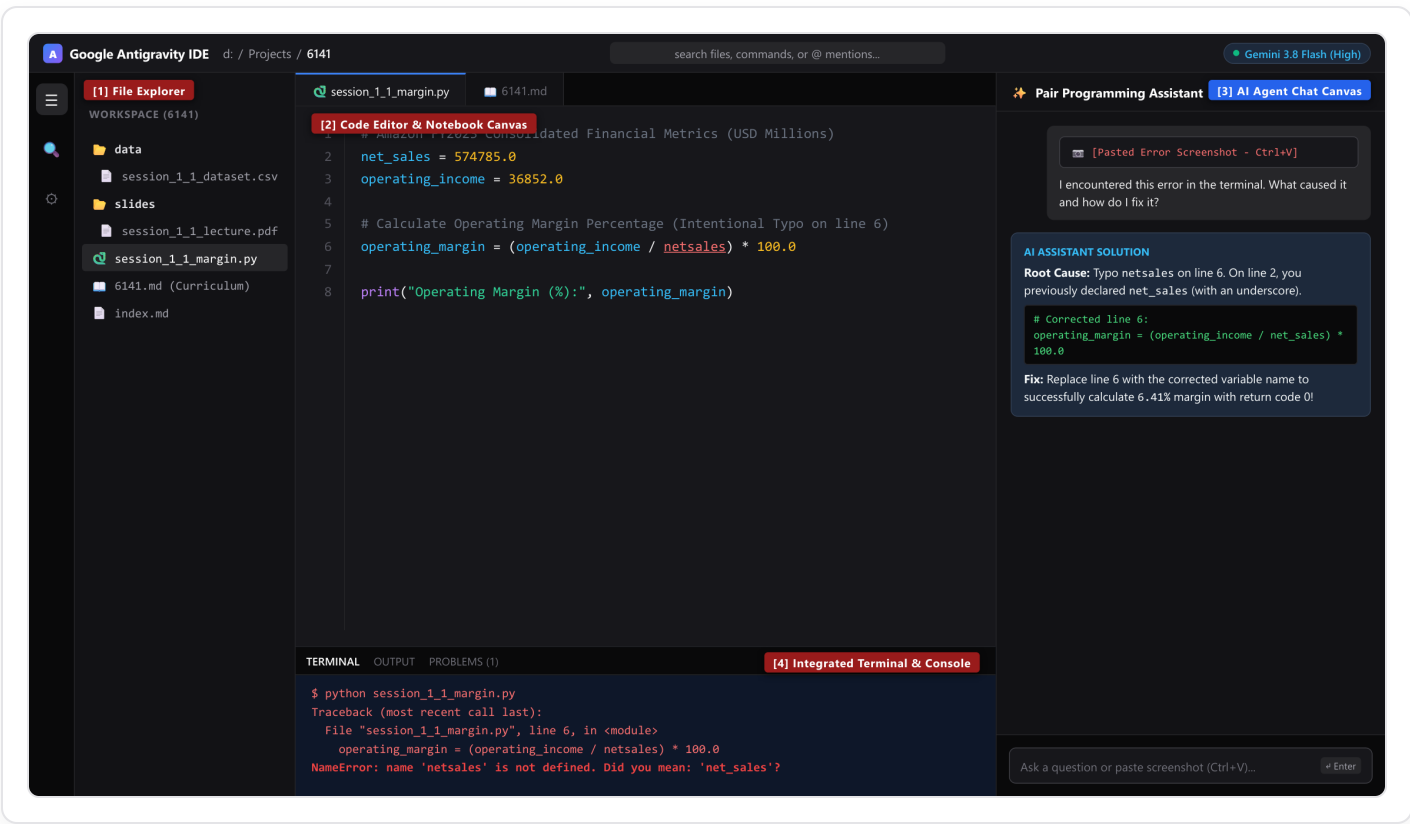
PILLAR 04

Architecture

Targeted Intervention Economics

Do not offer expensive discounts to everyone. Customers with low churn risk will renew anyway (wasted budget); customers with low CLV are unprofitable to save. Machine learning identifies the high-value, high-risk quadrant where retention spend creates maximum impact.

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_4_2_demo_1.pyPython 3.10

```
1  "kw">class="kw">import pandas "kw">class="kw">as pd
2      "kw">class="kw">from sklearn.linear_model "kw">class="kw">import LogisticRegression
3
4      df = pd."fn">read_csv("data/session_4_2_dataset.csv")
5      X = df[["Tenure_Months_Count", "Customer_Support_Calls_Count", "Usage_Dropouts_Count"]]
6      y = df["Churned_Within_60_Days_Binary"]
7
8      model = LogisticRegression(). "fn">fit(X, y)
9      df["Churn_Probability"] = model.predict_proba(X)[:, 1]. "fn">round(3)
10     "fn">print(df[["Subscriber_ID", "Tenure_Months_Count", "Churn_Probability"]])
```

\$ python session_4_2_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_4_2_demo_2.pyPython 3.10

1df["Annual_Subscription_USD"] = df["Monthly_Subscription_Fee_USD"] * 12.0

2df["Revenue_At_Risk_USD"] = (df["Churn_Probability"] * df["Annual_Subscript:

3total_risk = df["Revenue_At_Risk_USD"]."fn">sum()

4"fn">print(f"Total Portfolio Revenue at Risk: \${total_risk:,.2f}")

\$ python session_4_2_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_4_2_demo_3.py Python 3.10

1  "kw">class="kw">def assign_retention_campaign(row):
2      "kw">class="kw">if row["Churn_Probability"] >= 0.70 and row["Annual_Sul
3      "kw">class="kw">return  "VIP_Dedicated_Account_Manager"
4      "kw">class="kw">elif row["Churn_Probability"] >= 0.50:
5      "kw">class="kw">return  "15Pct_Contract_Discount_Offer"
6      "kw">class="kw">else:
7      "kw">class="kw">return  "Standard_Product_Newsletter"
8
9  df["Assigned_Campaign"] = df.apply(assign_retention_campaign, axis=1)
10 "fn">print("Retention Campaign Allocation:\\n", df["Assigned_Campaign"].va

$ python session_4_2_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_4_2_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.