

Business Problem Solving with Intelligent Analytics: A Python-Based Approach

Research Assignment for Computer Science Students CS31A

Due Date: 1 week from assignment date (e.g., if assigned July 17, 2025, due July 24, 2025)

I. Assignment Objective

This assignment challenges you to act as a data analyst/intelligent systems developer for a hypothetical company facing a common business challenge. You will:

- * Understand and define a given business problem from an analytical perspective.
- * Apply business analytics principles to propose a data-driven solution.
- * Implement a prototype solution using Python, leveraging intelligent system concepts (e.g., machine learning, rule-based systems, optimization) to demonstrate how data can drive actionable insights or automated decisions.
- * Communicate your findings and solution in a clear, concise report.

II. The Business Problem Scenario

You are working for a growing E-commerce Retailer that sells a wide variety of products online. They are struggling with customer churn, which is the rate at which customers stop doing business with them. High churn rates directly impact revenue and growth.

The current approach to retaining customers is reactive (e.g., offering discounts after a customer stops buying). The company wants to move to a proactive, data-driven approach to identify customers at risk of churning before they leave, and then implement targeted retention strategies.

Your Task: Develop an intelligent system prototype that identifies customers at risk of churn and suggests potential proactive interventions.

III. Research Questions & Development Tasks

Your research report and Python solution should address the following:

Part 1: Business Analytics Understanding & Problem Definition (Report)

* Business Problem Analysis:

* Explain why customer churn is a critical business problem for an e-commerce retailer. Quantify its potential impact (e.g., loss of revenue, increased marketing costs for new customer acquisition).

* What are the typical characteristics of a "churned" customer in an e-commerce context?

* How can business analytics transform the company's approach to customer retention from reactive to proactive?

* Data Requirements & Metrics:

* Identify the types of data an e-commerce retailer would likely collect that could be relevant for churn prediction (e.g., purchase history, website activity, customer demographics, customer service interactions).

* What key performance indicators (KPIs) or metrics would you use to define and measure churn? (e.g., last purchase date, frequency of purchases, average order value, Browse behavior).

* Intelligent System Approach:

* Which type of business analytics (Descriptive, Predictive, Prescriptive) is most relevant for solving this churn problem, and why?

- * Briefly explain how an intelligent system (e.g., using machine learning, rule-based logic, or a combination) can be used to address this problem. What advantages does it offer over manual methods?

Part 2: Python Solution & Implementation (Code + Report)

- * Proposed Solution Design:

- * Outline the steps you would take to build your intelligent system to predict customer churn. (e.g., data collection, data cleaning/preparation, feature engineering, model selection, training, evaluation, deployment).

- * Describe the intelligent system concept you plan to implement (e.g., a classification model like Logistic Regression, Decision Tree, or a simple rule-based system if data is not available). Justify your choice given the 1-week timeframe and typical data availability.

- * Python Prototype Implementation:

- * Simulated Data: Since real-world e-commerce data is sensitive and complex, you will create a small, synthetic dataset in Python that mimics realistic customer behavior, including features that could indicate churn (e.g., `customer_id`, `last_purchase_days_ago`, `total_orders`, `avg_order_value`, `website_visits_last_month`, `customer_service_tickets`, `is_churned` (target variable)). Ensure there's a reasonable balance of churned/non-churned customers.

- * Data Preprocessing: Demonstrate basic data cleaning and preparation steps on your simulated data (e.g., handling missing values if you introduced them, creating new features if relevant).

- * Intelligent System Logic: Implement your chosen intelligent system (e.g., a simple machine learning classification model using scikit-learn, or a clear set of if-else rules to categorize customers as 'at-risk').

- * If using ML: Train your model on a portion of the data and evaluate its performance (even with synthetic data, this is good practice).

- * If using rules: Clearly define the rules for identifying "at-risk" customers.

- * Prediction/Recommendation: Show how your system identifies "at-risk" customers and, for those identified, propose a generic proactive intervention (e.g., "Offer a personalized discount," "Send a re-engagement email," "Initiate a customer service call").

- * Code Documentation: Your Python code must be well-commented and organized.

- * Evaluation & Limitations (Report):

- * Discuss how you would ideally evaluate the effectiveness of your intelligent system in a real business scenario (e.g., accuracy, precision, recall, F1-score for classification; or business impact of interventions).

- * Identify the limitations of your prototype given the simulated data and the 1-week timeframe. What would be the next steps in a full-scale project?

IV. Research Work Format & Timeline (1 Week)

Deliverables:

- * Research Report (PDF): Approximately 1000-1500 words, addressing all research questions in Part 1 and Part 2 (Proposed Solution Design, Evaluation & Limitations).

- * Python Code (.py file): A single Python script containing your simulated data generation, data preprocessing, intelligent system implementation, and prediction/recommendation logic. Ensure it runs without errors.

* Simulated Dataset (CSV - optional but recommended): If you generate a complex synthetic dataset, save it as a CSV and include it.

Recommended Timeline:

- * Day 1: Understand Problem & Research (3-4 hours)
 - * Thoroughly read the assignment.
 - * Research customer churn in e-commerce, common data points, and general business analytics approaches.
 - * Start brainstorming how you'll simulate data.
- * Day 2: Data Simulation & Initial Setup (4-6 hours)
 - * Plan the structure of your synthetic dataset (columns, value ranges).
 - * Start writing Python code to generate the synthetic data.
 - * Set up your project environment (e.g., virtual environment, necessary libraries like pandas, numpy, scikit-learn).
- * Day 3: Intelligent System Design & Basic Implementation (6-8 hours)
 - * Refine your chosen intelligent system approach.
 - * Implement data preprocessing steps on your simulated data.
 - * Write the core logic for your intelligent system (ML model training/prediction or rule-based logic).
- * Day 4: Refinement & Prototyping (6-8 hours)
 - * Test your Python code thoroughly.
 - * Ensure your system identifies "at-risk" customers and suggests interventions.
 - * Add comments and make your code clean and readable.
 - * Begin drafting sections of your report related to the Python solution.
- * Day 5: Report Writing - Part 1 Focus (6-8 hours)
 - * Focus on writing the "Business Problem Analysis," "Data Requirements & Metrics," and "Intelligent System Approach" sections of your report.
 - * Ensure a clear introduction and logical flow.
- * Day 6: Report Writing - Part 2 Focus & Integration (6-8 hours)
 - * Complete the "Proposed Solution Design," "Evaluation & Limitations" sections.
 - * Review the entire report for consistency, clarity, and accuracy.
 - * Ensure all research questions are addressed.
- * Day 7: Final Review & Submission (2-3 hours)
 - * Proofread the report for grammar, spelling, and formatting.
 - * Verify all code runs correctly and is well-commented.
 - * Ensure all deliverables are included.
 - * Submit your work.

V. Report Structure

Your research report (PDF) should follow this structure:

- * Title Page:
 - * Assignment Title: Business Problem Solving with Intelligent Analytics: A Python-Based Approach
 - * Your Name, Student ID, Course, Instructor, Date
- * Table of Contents
- * Abstract (approx. 150-200 words):

- * Briefly state the business problem, your chosen analytical approach, the intelligent system implemented, and key findings/recommendations.

- * 1. Introduction:

- * Overview of the importance of business analytics and intelligent systems.

- * Introduce the e-commerce churn problem and the objective of this report/project.

- * 2. Business Problem Analysis: Customer Churn in E-commerce

- * Detailed discussion of Research Question 1 (why churn is critical, characteristics of churned customers, analytics transformation).

- * 3. Data for Churn Prediction

- * Detailed discussion of Research Question 2 (data types, KPIs/metrics).

- * 4. Intelligent System Approach for Churn Prediction

- * Detailed discussion of Research Question 3 (type of analytics, how intelligent systems help).

- * 5. Proposed Python-Based Solution

- * Detailed discussion of Research Question 4 (solution design, intelligent system concept chosen and justification).

- * 6. Python Prototype: Implementation Details

- * Explain your synthetic data generation process.

- * Describe your data preprocessing steps.

- * Explain the logic of your intelligent system (ML model or rule-based system).

- * Show examples of output (e.g., a few identified "at-risk" customers with suggested interventions). Do not paste large chunks of code here; refer to your attached .py file.

- * 7. Evaluation and Limitations

- * Detailed discussion of Research Question 5 (how to evaluate, limitations of prototype, next steps).

- * 8. Conclusion:

- * Summarize your findings and the impact of an intelligent analytics approach on the business problem.

- * Reiterate the value of computer science skills in this domain.

- * References/Bibliography:

- * Cite all sources used (academic papers, articles, documentation).

VI. Evaluation Criteria

Your submission will be evaluated based on:

- * Understanding of the Business Problem: How well you define and articulate the problem and its business impact.

- * Analytical Depth: Your ability to identify relevant data, metrics, and appropriate analytical approaches.

- * Intelligent System Design & Implementation: The clarity of your proposed solution, the correctness and effectiveness of your Python code prototype, and appropriate use of intelligent system concepts.

- * Code Quality: Readability, comments, organization, and adherence to Python best practices.

- * Report Quality: Clarity, logical flow, accuracy, completeness, and adherence to formatting guidelines.

- * Critical Thinking: Your insights into the limitations of your prototype and next steps for a real-world solution.

* Adherence to Timeline: Timely submission of all deliverables.

This assignment provides a hands-on experience, bridging theoretical knowledge of business analytics with practical application using Python, a crucial skill for computer science students entering the data-driven world.