May 5, 2025

Jaxon Adams

C964: Computer Science Capstone Template

**Warning:** Though it is not stated in the official resources, evaluators do not like outlines.  Write narratively using paragraphs with complete sentences. Use these [C964 examples](https://ashejim.github.io/C964/resources.html#examples) to see what evaluators typically expect.

Task 2 parts A, B, C, and D

[Part A: Letter of Transmittal 1](#_Toc98598250)

[Letter of Transmittal Requirements 2](#_Toc1738085866)

[Letter Template 2](#_Toc1133266111)

[Part B: Project Proposal Plan 3](#_Toc1370766476)

[Project Summary 4](#_Toc904507251)

[Data Summary 4](#_Toc1736393957)

[Implementation 4](#_Toc1241988654)

[Timeline 4](#_Toc1357178365)

[Evaluation Plan 5](#_Toc623361460)

[Resources and Costs 5](#_Toc1538507987)

[Part C: Application 5](#_Toc1471073175)

[Part D: Post-implementation Report 6](#_Toc651895932)

[Solution Summary 7](#_Toc1134136520)

[Data Summary 7](#_Toc182221765)

[Machine Learning 7](#_Toc1505466430)

[Validation 7](#_Toc391434166)

[Visualizations 7](#_Toc201059345)

[User Guide 7](#_Toc1365484010)

[Reference Page 8](#_Toc1702353417)

# Part A: Letter of Transmittal

## Letter of Transmittal Requirements

The *Letter of Transmittal* should convince senior leadership to approve your project. Write a brief cover letter (suggested length 1-2 pages) describing the problem, how the application (part C) applies to the problem, the practical benefits to the organization, and a brief implementation plan. Include all artifacts typical of a professional (business) letter, e.g., subject line, date, greeting, signature, etc.

The letter should be concise and target a non-technical audience. Include the following:

* A summary of the problem.
* A proposed solution centering around your application.
* How the proposed solution benefits the organization.
* A summary of the costs, timeline, data, and any ethical concerns (if relevant).
* Your relevant expertise.

## Letter Template

[Today’s date]

[Recipient’s name]

[Company name]

[Address]

Dear [Recipient’s name],

(Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer at sodales leo, nec fermentum tellus. Vivamus leo lorem, semper eget erat nec, ultricies vulputate sapien. Integer eget nisi at erat condimentum dictum. Nam sed ornare leo, non pretium nulla. Nam dolor arcu, condimentum et maximus quis, bibendum ut risus. Nam hendrerit ac erat sit amet luctus. Quisque lacinia sapien sed nisl porta, id rutrum odio tristique. Sed sodales nisi a condimentum fermentum. Suspendisse lobortis diam in orci consectetur congue. Sed ligula felis, accumsan eu venenatis ac, hendrerit ac nulla. In ultrices, sem a semper ultrices, orci ex dictum sapien, ac dapibus ligula lectus at est. Quisque at posuere purus, vitae ultricies risus. Donec tincidunt, ipsum eget euismod lobortis, tortor nisl luctus sem, quis blandit tortor dui eu dui.

Praesent sagittis, leo vitae sodales cursus, purus orci rhoncus ligula, quis facilisis ante est nec lectus. Curabitur augue quam, ultricies at arcu eget, molestie eleifend neque. Sed interdum tempor purus, et luctus tellus feugiat imperdiet. Maecenas scelerisque viverra orci tincidunt luctus. Vestibulum sodales eros ut ex luctus tempor. Curabitur eget leo vehicula, malesuada urna ut, eleifend nisi. Quisque sapien tellus, ornare ac magna quis, ultricies consectetur mauris. Nunc erat ligula, mattis id tempor ut, venenatis ac mi. Sed sit amet odio ac ligula tincidunt iaculis sit amet vel ipsum.

Etiam lobortis aliquam metus, eu aliquam ante aliquam ut. Nam tristique sagittis mauris vel tempor. Quisque rhoncus, justo sed lobortis porta, nulla libero pulvinar tortor, ut ullamcorper justo enim ut erat. Praesent lobortis ut leo in aliquet. Suspendisse aliquet velit nulla, a rhoncus nibh vestibulum iaculis. Praesent mollis nibh nec ultrices blandit. Pellentesque felis elit, pretium at risus in, commodo consectetur tortor. Etiam fringilla mi quis erat mollis ultricies. Phasellus vestibulum elementum commodo. Sed congue vulputate orci in porta. Pellentesque scelerisque facilisis justo, a bibendum ligula tempus quis. Aenean efficitur eleifend lorem, et tempus risus consequat quis. Cras varius metus sapien, ac malesuada sem volutpat sed.

Sincerely,

[Sign here: e.g., Jane Smith]

[Your name, title]

# Part B: Project Proposal Plan

The project proposal should target your client’s middle management. This audience may be IT professionals but have limited computer science expertise. Use appropriate industry jargon and sufficient technical details to describe the proposed project and its application. Remember, you’re establishing the technical context for your project and how it will be implemented for the client. **Write everything in the future tense.**

## Project Summary

* Describe the problem.
* Summarize the client and their needs as related to the problem.
* Provide descriptions of all deliverables. For example, the finished application and a user guide.
* Provide a summary justifying how the application will benefit the client.

## Data Summary

* Provide the source of the raw data, how the data will be collected, or how it will be simulated.
* Describe how data will be processed and managed throughout the application development life cycle: design, development, maintenance, etc.
* Justify why the data meets the needs of the project. If relevant, describe how data anomalies, e.g., outliers, incomplete data, etc., will be handled.
* Address any ethical or legal concerns regarding the data. If there are no concerns, explain why.

## Implementation

* Describe an industry-standard methodology to be used.
* An outline of the project’s implementation plan. The focus can be the project’s development or the implementation of the machine learning solution.

## Timeline

* Provide a projected timeline. Include each milestone and deliverable, its dependencies, resources, start and end dates, and duration. (a table is not required but encouraged).
* Dates should be in the future. Write ‘NA’ where an item is not applicable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone or deliverable** | **Project Dependencies** | **Resources** | **Start and End Date** | **Duration** |
|  |  |  |  |  |
|  |  |  |  |  |

## Evaluation Plan

* Describe the verification method(s) to be used at each stage of development.
* Describe the validation method to be used upon completion of the project.

## Costs

Include the itemized costs of the project. Include specific item names where applicable, e.g., ‘PyCharm Professional Ed. 2024.3.5.’

* Itemize hardware and software costs.
* Itemize estimated labor time and costs.
* Itemize estimated environment costs of the application, e.g., deployment, hosting, maintenance, etc.

# Part C: Application

Part C is your submitted application. This part of the document can be left blank or used to include a list of any submitted files or links.

The minimal requirements of the submitted *application* are as follows:

1. **The application functions as described.** Following the ‘User Guide’ in part D, the evaluator must be able to review your application on a Windows 10 machine successfully.
2. **A mathematical algorithm applied to data,** e.g., supervised, unsupervised, or reinforced machine learning method.
3. **A “user interface.”** Following the ‘User Guide’ in part D, the client must be able to use the application to solve the proposed problem (as described in parts A, B, and D). For example, the client can input variables, and the application outputs a prediction.
4. **Three visualizations.** The visualizations can be included separately when including them in the application is not ideal or possible; e.g., the visualizations describe proprietary data, but the application is customer-facing.
5. **Submitted files and links are static and accessible.** All data, source code, and links must be accessible to evaluators on a Windows 10 machine. If parts of the project can be modified after submission, matching source files must be submitted. For example, if the application is a website or hosted notebook, the `.html` or `.ipynb` files must be submitted directly to assessments.

Ideally, submitted applications should be reviewable using either Windows or Mac OS, e.g., Jupyter notebooks, webpages, Python projects, etc. If the source files exceed the 200 MB limit, consider providing screenshots or a Panopto video of the functioning application and contact your course instructor.

# Part D: Post-Implementation Report

Create a post-implementation as outlined below. Provide sufficient detail so that a reader knowledgeable in computer science but unfamiliar with your project can understand what you have accomplished. Using examples and visualizations (including screenshots) beyond the three required is recommended (but not required). **Write everything in the past tense.**

## Solution Summary

* Summarize the problem and solution.
* Describe how the application solves the problem from parts A and B.

## Data Summary

* Provide the source of the raw data, how the data was collected, or how it was simulated.
* Describe how data was processed and managed throughout the application development life cycle: design, development, maintenance, etc.

## Machine Learning

The machine learning model’s predictive functionality is accessed through the method “predict()” in the file “src/server/app.py”:

A computer screen shot of code

AI-generated content may be incorrect.

This method exposes an endpoint in the Flask API for model prediction. This allows for loan application data to be sent in a JSON payload to the server, which queries the model and returns a prediction to the user. An example of the model’s output is shown in the Postman request below:

A screenshot of a computer

AI-generated content may be incorrect.

This method achieves this functionality by using the list of features cached on the server to format the request data into a pandas data frame. The data frame is submitted to the cached model through the model’s built-in “predict\_proba()” method, where the resulting probability is converted to a Boolean true/false prediction, where “true” means the borrower will likely default on their loan.

This method was selected and developed to make model predictions easily accessible to a frontend web application. Running computationally expensive methods, such as my machine learning model, on the server and making such functionality available to the frontend application through an API is a common software engineering practice, so a prediction method that creates an API endpoint was a desirable way to expose machine learning functionality in my project.

## Validation

For each machine learning algorithm described in the section above, do the following:

* Identify the model’s machine learning category, e.g., supervised, unsupervised, or reinforced. For blended approaches, identify the category most relevant to the model’s application.
* An appropriate validation method for the model’s performance.

For supervised learning and reinforcement learning

* + Describe an appropriate metric(s) for testing the model’s performance.
  + Provide results of testing using the described metric.

For unsupervised learning

* + Describe an appropriate method(s) for testing the model’s performance.
  + Provide the results of testing using the above method. The method should provide an example of the model’s output and how the output is relevant or a metric measuring the model’s performance, e.g., the Rand index or Silhouette Coefficient.

## Visualizations

Identify the location of at least three unique visualizations. They can additionally be included here.

## User Guide

The application can be used either in a live environment or hosted locally. To use the live application, navigate to the following page in your web browser of choice:

<https://wgu-capstone-client.onrender.com/>

If you wish to run the application locally, first ensure you have a recent version of Python (3.11+) and Node.js (19+) installed on your computer. Python can be downloaded from <https://www.python.org/downloads/>, and Node.js can be downloaded from <https://nodejs.org/en/download/>. Next, perform the following steps:

1. Open a command prompt (PowerShell, Git Bash, etc.) and navigate to the project directory. For example, run the command “cd path/to/project”.
2. Initialize a Python virtual environment with the following command: “python -m venv .venv”.
3. Activate your new virtual environment; for example, run “./.venv/Scripts/activate”. The actual command may vary depending on your environment. For more details, visit <https://docs.python.org/3/library/venv.html>.
4. Run the following command to install dependencies, download a sample of the full LendingClub dataset, train a Random Forest classifier, and run the full-stack application locally: “npm run for:evaluator”. This process may take several minutes. When the development environment is ready, it should automatically open the web page in your preferred browser.

Once you have the web application (hosted locally or live) open in your web browser, perform the following steps to simulate a client using the application:

1. At the top right of the web page, click the link “Loan Default Prediction” to navigate to the next page.
2. Review the possible inputs for submitting a loan application for prediction. For convenience, defaults are populated for most fields.
3. Click the “Submit Application” button at the top of the form to submit the provided data to the ML model for prediction. After a moment, you should see the model’s prediction listed above the application form.
4. Change a few input fields to simulate a risky borrower; for example, set “Grade” to “D”, “Loan Amount” to 50000, “Interest Rate” to “19.99”, and “Employment Length” to “< 1 year”. Submit the application and note the model’s prediction.

# Reference Page

Include references for cited works, e.g., (Author, year), following an accepted writing style. References are not required; this page can be removed if no references are used. To cite sources used for code, you should include the references as code comments within the source code.