Computer Science Capstone

Task 2: Parts A, B, C, and D

[Part A: Letter of Transmittal 2](#_Toc203402596)

[Part B: Project Proposal Plan 4](#_Toc203402597)

[Project Summary 4](#_Toc203402598)

[Data Summary 4](#_Toc203402599)

[Implementation 5](#_Toc203402600)

[Timeline 5](#_Toc203402601)

[Evaluation Plan 5](#_Toc203402602)

[Resources and Costs 6](#_Toc203402603)

[Part C: Application 7](#_Toc203402604)

[Part D: Post-implementation Report 8](#_Toc203402605)

[Solution Summary 8](#_Toc203402606)

[Data Summary 8](#_Toc203402607)

[Machine Learning 8](#_Toc203402608)

[Validation 9](#_Toc203402609)

[Visualizations 9](#_Toc203402610)

[User Guide 12](#_Toc203402611)

[References 15](#_Toc203402612)

# Part A: Letter of Transmittal

May 2, 2025

Jacob James Nubert   
Nubert Organization  
1650 Broadway

New York, NY 10019

Dear Mr. Nubert,

Broadway productions often involve high-stakes financial investments with unpredictable outcomes. Most Broadway shows close without recouping their production costs, causing significant losses for producers and investors. Despite the wealth of available historical data, decisions are still frequently made based on intuition. My project proposes a solution to this problem: a machine learning-powered web application that predicts the commercial success of upcoming Broadway shows.

The proposed data product uses historical Broadway performance data—such as weekly grosses, attendance figures, seasonality, and show metadata—to train predictive models capable of estimating the revenue potential of new shows. These models, built using supervised machine learning algorithms, will be integrated into an interactive web application for practical use by producers, investors, and marketing strategists.

This tool benefits the organization by reducing the guesswork and subjectivity in greenlighting Broadway projects. The application will provide actionable insights into the most promising types of shows, optimal opening times, and expected weekly grosses. The application will include both descriptive visualizations of Broadway show patterns and a predictive model that estimates gross revenue. This tool will empower stakeholders to make informed, data-driven investment decisions.

The application will be developed using Python (with Flask’s web framework), HTML, CSS, and JavaScript, as well as various libraries and tools. The dataset, sourced from CORGIS (The Collection of Really Great, Interesting, Situated Datasets), includes Broadway weekly performance records from 2000 to 2016 and is aggregated from publicly available sources (i.e.,The Broadway League).

The project will follow the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology to ensure a structured and iterative development process. It begins with business understanding to define the financial risks Broadway producers face, followed by data understanding to explore key variables from the CORGIS Broadway dataset. The next steps include data preparation—such as cleaning, transforming, and engineering features—followed by model training using Linear Regression. Finally, the best-performing model will be evaluated using metrics like R² and MSE and deployed in a web-based application with interactive visualization capabilities.

The primary objective is to develop an accurate predictive model capable of forecasting a show's success, measured by metrics such as revenue and attendance. I hypothesize that with proper feature engineering and training, the Linear Regression model will achieve at least 70% predictive accuracy, enabling a 10% improvement in profitability through optimized launch conditions.

Total development is projected to require 140 hours. We will start on June 2, 2025 and complete the project by July 9, 2025 for final stakeholder review with the final version completed by July 11, 2025. While the basic implementation will utilize open-source tools and local development, a professional deployment—including stakeholder reviews, secure hosting, and risk audits—would cost approximately $120,000. This includes resources such as data scientists, infrastructure, third-party audits, and visualization platforms.

Security and privacy remain a top priority. The dataset includes no personally identifiable information, and all data will be stored securely, with high-end encryption. Model outputs and reports will be shared only with authorized personnel and access logs will be maintained to ensure compliance.

I bring to this project my hands-on experience in Python, data science, web development, and project management. I have previously completed similar projects involving predictive modeling, data cleaning, visualization, and stakeholder reporting. This expertise positions me to deliver a solution that is both technically sound and practically valuable.  
  
Thank you for your consideration.  
  
Sincerely,

# Part B: Project Proposal Plan

## Project Summary

Broadway producers and investors face immense financial risk when backing new productions. Many shows close early without recouping their investments due to uncertain audience reception, timing, or marketing effectiveness. Despite decades of publicly available Broadway performance data, decision-makers lack predictive tools to guide investments. This project proposes a machine learning-powered web application that predicts the likelihood of a Broadway show's commercial success. The application will serve as a decision-support tool by modeling key success indicators using historical data, and presenting actionable insights through an interactive web interface.

The clients include Broadway investors, producers, and marketing teams who will benefit from reduced financial uncertainty and more data-driven show selection. Deliverables will include: (1) a web-based application with an embedded predictive model, (2) stakeholder-friendly data visualizations and dashboards, and (3) a user guide for interpreting model outputs. These assets will collectively support investment forecasting and marketing strategy development.

This solution will allow stakeholders to forecast show performance and fine-tune their decisions with greater accuracy. With a projected predictive accuracy of at least 70%, stakeholders can expect up to a 10% improvement in profitability when leveraging model insights.

## Data Summary

The application will use the "Broadway" CSV file publicly available on CORGIS, which includes weekly grosses, attendance, and metadata for Broadway shows from 1990 to 2016. The dataset is compiled from trusted industry sources like Playbill and The Broadway League. During the design phase, raw data will be cleaned, formatted, and analyzed for consistency. Although the original CORGIS dataset includes Broadway data from 1990 to 2016, we will exclude data prior to 2000 to reduce dataset size and improve model relevance. This decision was made to avoid potential inaccuracies caused by outdated economic trends and structural changes in the Broadway industry over the decades.

The dataset will be processed using Python libraries like pandas and numpy. Missing values will be addressed through imputation or removal depending on severity. Since the data contains no PII, it poses no legal or ethical concerns, but confidentiality around model outputs and insights will be maintained through encryption and limited-access permissions.

## Implementation

The development process will follow the CRISP-DM methodology. In the Business Understanding phase, key challenges facing show investment will be defined. Data Understanding will involve profiling and identifying trends and inconsistencies in the dataset. Data Preparation includes feature selection, encoding, and handling of missing values. In the Modeling phase, the supervised algorithm—Linear Regression—will be trained and evaluated. The best-performing model will be integrated into the web application. The final Evaluation and Deployment phases will validate the solution using stakeholder testing and secure deployment strategies.

## Timeline

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Start** | **End** | **Tasks** |
| Planning & Designing (20 hours) | | | |
| 0 | June 02 | June 02 | Project is accepted by sponsor team |
| 1 | June 03 | June 08 | Data acquisition, cleaning, and EDA (exploratory data analysis), missing value review |
| Development (95 hours) | | | |
| 2 | June 09 | June 12 | **REPLACE:** Create season, show type, duration, and encode categorical fields |
| 3 | June 13 | June 20 | Train Linear Regression model |
| 4 | June 21 | June 27 | Cross-validation, residual analysis, hyperparameter tuning, and additional feauture engineering |
| Documentation (25 hours) | | | |
| 5 | June 28 | July 08 | Final visualizations, business write-up, and Start Up Guide documentation |
| 6 | July 09 | July 11 | Final stakeholder review and formal submission of the project |

## Evaluation Plan

During development, dataset integrity will be validated through consistency checks, missing value analysis, and feature correlation analysis. During modeling, five-fold cross-validation will be used to assess model generalization. Final evaluation metrics will include R² and Mean Squared Error (MSE) on a holdout test set. Upon integration, model predictions will be validated for business relevance through stakeholder feedback sessions. At least 85% of users surveyed will be expected to rate the tool as “useful” and easy to use.

## Resources and Costs

Hardware: Existing MacBook Pro (no new cost) $0  
Software: Python (with Flask), HTML/CSS/JS (all open-source) $0

Labor for Data Science Personnel: 140 hours: $90,000  
  
Infrastructure (AWS, GitHub, Tableau): $5,000  
Legal & Compliance: $4,000  
Risk Review & QA: $8,000  
Stakeholder Review: $3,000

Contingency: $1,000

**TOTAL Estimated Professional Deployment Cost:** **$111,000**

# Part C: Application

Attached as a ZIP file named “broadway-insights.zip” and also downloadable/clonable via GitHub at: https://github.com/gregbarbs/broadway-insights.

# Part D: Post-implementation Report

## Solution Summary

## Broadway productions represent significant financial investments, yet most shows fail to recoup their production costs. This project addressed the lack of data-driven forecasting tools available to Broadway stakeholders by developing a machine learning-powered web application that predicts weekly gross revenue for Broadway shows. The application helped producers, investors, and marketers make informed, evidence-based decisions.

## The solution integrated a trained linear regression model within a Flask-based web app that allows users to input data (such as attendance, capacity filled, gross potential, and number of performances) and receive a revenue prediction with a confidence interval and model accuracy score. Additionally, a set of static visualizations supported interpretability and data understanding.

## Data Summary

**Data Source**:

The dataset was obtained from the CORGIS Project (The Collection of Really Great, Interesting, Situated Datasets), which contains Broadway weekly gross performance data from 1990 to 2016. The raw CSV was pre-cleaned and limited to data from 2000 to 2016 to better reflect modern economic conditions and reduce noise from outdated historical trends.

**Data Lifecycle Management**:

* **Design Phase**: Identified relevant features such as attendance, show capacity, potential gross, and number of performances.
* **Development Phase**: Implemented preprocessing in prepare\_data.py, which parsed and cleaned the dataset, handled nulls, converted date formats, and engineered features.
* **Maintenance & Testing**: Logged data access and prediction events using logs.txt for audit purposes. This log also supports post-deployment monitoring.

## Machine Learning

Method Used: Linear Regression

What: Linear regression was used to model the relationship between independent features (e.g., weekly attendance, capacity, gross potential, and number of performances) and the dependent variable (weekly gross revenue).

How:

* Used scikit-learn’s LinearRegression model.
* Split the dataset into training and testing sets (70/30 split).
* Trained the model using train\_linear\_model.py.
* Stored the trained model, R² score, and residual standard deviation in a serialized .pkl file for use by the Flask app.

Why: Linear regression is a transparent, interpretable baseline model suitable for continuous outcome prediction. Given the linear nature of Broadway financial metrics and the small-to-medium dataset size, it provided strong performance without overfitting.

## Validation

Validation Metric: R² (Coefficient of Determination) and Mean Squared Error (MSE)

Results:

* **R² Score**: 76.59% (indicating the model explains approximately 77% of the variance in gross revenue).
* **MSE**: Approximately $37 billion (raw dollar squared error; expected given the revenue scale).

These results validated the model's ability to generalize on unseen test data. Users are shown the R² score and a 95% confidence interval around predictions to understand the model's reliability.

Future improvements may include testing ensemble regressors, tuning regularization parameters, or incorporating temporal/contextual features like seasonality and genre tags.

## Visualizations

The location of the three visualizations are located in the following path relative to the root:

“broadway-insights/backend/static”

There are three files in this folder:

1. attendance\_vs\_gross.png
2. gross\_by\_showtype.png
3. predictions\_vs\_actuals.png

The visualizations can also be located in the webpage “visuals.html” in the web application (link at the bottom of the main “index.html” page.)

Here are the three visualizations below:

**attendance\_vs\_gross.png**

A diagram of a graph

AI-generated content may be incorrect.

**gross\_by\_showtype.png**

A diagram of a diagram

AI-generated content may be incorrect.

**predictions\_vs\_actuals.png**

A graph with blue dots and a red line

AI-generated content may be incorrect.

## Broadway Insights User Guide

This document walks users through downloading, setting up, and using the Broadway Insights web application. Broadway Insights uses historical performance data to predict weekly gross revenue for Broadway shows.

The following User Guide is for Windows 10 machines.

1. Software Requirements

- Operating System: Windows 10

- Python 3.10 or higher (https://www.python.org/downloads/)

- (Optional) Git for Windows (https://git-scm.com/)

- A modern browser (e.g., Chrome, Firefox, Edge)

- Internet access to install dependencies

2. Download the Project Files

Option A - Using Git:

1. Open Command Prompt or PowerShell.

2. Run: ‘git clone <https://github.com/gregbarbs/broadway-insights.git>’

3. Navigate to the folder: ‘cd broadway-insights’

Option B – Download ZIP:

1. Download attached ZIP file or via GitHub by going to: <https://github.com/gregbarbs/broadway-insights>
2. Click the green “Code” button 🡪 “Download ZIP”
3. Extract the ZIP to an easily accessible location (i.e., “Documents”)

3. Install Python & Set Up Environment

1. Download and install Python from <https://www.python.org/downloads/>
2. Open Command Prompt and navigate to the extracted folder: cd (Documents)\broadway-insights
3. Create a virtual environment: “python -m venv .venv”
4. Activate the environment: “.ven\Scripts\activate”
5. Install required libraries: “pip install -r requirements.txt”

4. Launch the Application

Option A (Easy method):

1. Double-click the ‘run\_app.bat’ file in the root project directory.

Option B (Advanced method):

1. Run the backend manually: “python backend\app.py”
2. Open a browser and go to: <http://127.0.0.1:5000>

5. Use the Web App to Predict Gross (with an example)

1. On the homepage, enter values into the fields:
   1. Weekly Attendance (i.e., 10,000)
   2. Capacity Filled (percentage): (i.e., 85)
   3. Gross Potential (percentage): (i.e., 95)
   4. Number of Performances: (i.e., 8)
2. Click the “Predict Gross” button.
3. The app will return:
   1. Estimated Gross: (i.e., $978,265.31)
   2. Confidence Interval (+/- range): ($602,197.29 - $1,354,333.34)
   3. Model Accuracy: 76.95%

6. View Visualizations

1. At the bottom of the homepage, click the link: “View Project Visualizations”
2. This will open the ‘visuals’ page that shows:
   1. Attendance vs Gross Revenue
   2. Gross by Show Type
   3. Predicted vs Actual Gross

7. View Log Output

All prediction requrests (with timestamps and input values) are logged in the file named “logs.txt”

8. Security & Data

* Dataset includes ***no*** personal data
* App runs locally *only* – no internet server needed
* Logs are saved locally
* Source code and data are open-source

You now have a working Broadway revenue prediction system, built with Python, Flask, and machine learning. Explore different inputs and visualize trends using the tools provided!

For questions or support, contact: gbarb14@wgu.edu

# References

Project CORGIS. (n.d.). *Broadway* *[CSV file]*. The CORGIS Project. https://corgis-edu.github.io/corgis/csv/broadway/