

C964: Computer Science Capstone

Task 2 parts A, B, and D

Part A: Letter of Transmittal	2
Part B: Project Proposal Plan	3
Project Summary.....	Error! Bookmark not defined.
Data Summary.....	3
Implementation	4
Timeline.....	4
Evaluation Plan.....	5
Resources and Costs	5
Part D: Post-implementation Report	6
Solution Summary.....	6
Data Summary.....	6
Machine Learning.....	6
Validation	8
Visualizations	8
User Guide	10

Part A: Letter of Transmittal

10/31/2023

Management@gamingnews.com

Gaming News

Dear Management,

I am writing to present an opportunity that promises to increase user engagement and satisfaction on our Gaming News platform. After running many market analysis, our team has identified a gap that can be strategically filled to benefit our readers and our brand: the need for personalized gaming content recommendations.

To address this, we propose the development of an intelligent recommendation system designed to present users with game news and articles aligned with their interests. This application will use state of the art machine learning techniques to analyze user preferences and reading habits, thereby predicting and suggesting highly relevant content. The integration of this technology will streamline user experience, increase website traffic, and enhance our market positioning by offering a personal browsing experience.

The benefits will not only encourage longer browsing sessions as users discover content that resonates with their preferences but also position us as a leader in digital content curation. This, in turn, can lead to increased advertising revenue and partnership opportunities with game developers.

Our proposed plan is for development to take about three months and a budget that includes the costs of development, testing, and deployment, ensuring almost no disruptions to our current operations.

As an experienced lead engineer, I am confident in our team's ability to deliver this project efficiently, using modern machine learning algorithms as those used by other companies.

We look forward to discussing this exciting venture further and securing your endorsement to proceed.

Warm regards,

Kenna Heng

Senior Data Analyst

Part B: Project Proposal Plan

Using our expertise in data analysis we would like to tackle enhancement of user engagement on the Gaming News platform. The current system lacks a dynamic method to present readers with related gaming content, which results in a lackluster user experience and potential missed opportunities for longer engagement.

Our client, Gaming News, serves a diverse readership eager for tailored content. Their need is for a sophisticated yet user-friendly recommendation system that can adapt to individual user preferences, suggesting similar or related game news articles seamlessly.

The deliverables for this project will include:

1. A fully functional machine learning-based recommendation system integrated into the Gaming News website. This system will analyze the users current article and history to suggest relevant content.
2. A user guide detailing the operation and maintenance of the recommendation system.
3. A manual for administration that provides instructions for system updates, data management, and customization features.

This application aims to significantly benefit Gaming News by facilitating user engagement through personalized content recommendations, and increasing time spent on site. This enhancement will also amplify advertising revenue through heightened user activity. These changes will modernize our content offerings for customized gaming news, putting us ahead of our competitors.

Data Summary

For sourcing the data, we will scour the current Gaming News database for titles and corresponding genres. Any future new games will be added after going through a verification process for legitimacy.

We will use an unsupervised learning algorithm to generate recommendations based on similar genres. We will implement a cosine similarity matrix to compare a list of games and their genres to match the current article being read. Maintenance will be regularly scheduled removing any duplicate values or inaccurate genre classifications. Further fine tuning will be performed by either staff or user contributions via voting for updating a games genre labels or reporting inaccuracies.

The data is pivotal to the project's objective to create a personalized user experience, and it has been selected for its direct correlation with user preferences and behavior on the Gaming News platform.

We should not have any worries about any ethical concerns regarding the data. Video game titles and articles are protected under fair use, and are sourced legally.

Implementation

We will implement an Agile development methodology, which is the industry standard for dynamic project management. This approach uses flexible and incremental improvements, allowing our team to continuously integrate user feedback and make adjustments in real-time. It encourages collaboration, daily stand-ups, and sprints, Agile will enable us to break down the project into manageable phases, with each creating a potentially shippable product.

The project's implementation plan is structured in phases:

- Initial setup, where we define requirements and prepare data;
- Development, involving cycles where the machine learning model is trained and refined;
- Deployment, where the model is integrated into Gaming News' existing platform.

Timeline

Milestone or deliverable	Duration (days)	Projected start date	Anticipated end date
Initial Setup & Requirement Gathering	5	November 1st, 2023	November 5th, 2023
Data Preparation & Processing	10	November 6th, 2023	November 15th, 2023
Development: Model Training	15	November 16th, 2023	December 1st, 2023
Development: Model Refinement	10	December 2nd, 2023	December 11th, 2023
Deployment & Integration	5	December 12th, 2023	December 16th, 2023
Testing & Feedback Gathering	7	December 17th, 2023	December 23rd, 2023
Final Adjustments & Post-Deployment Support	7	December 24th, 2023	December 30th, 2023

Evaluation Plan

During each stage of development, our primary verification method will be unit and integration testing. We will use continuous integration tools that automatically run test suites whenever code updates are made, ensuring that every component adheres to its specifications and requirements. Regular manual inspections and peer code reviews will be done to maintain coding standards and prevent any potential risks.

Validation of our machine learning model will revolve around its real-world performance and user feedback. Metrics like accuracy and precision will be used to evaluate the model's performance. Every recommendation for a game should be of a similar genre. Feedback from end-users will be crucial, providing insights into the model's utility and areas for refinement. Incorrect genre labels will be removed and updated to fit the correct criteria.

Resources and Costs

- **Hardware and Software Costs:**
 - Server Infrastructure: \$10,000
 - Workstations for Development: \$5,000
 - Machine Learning Software Licenses: \$2,500
- **Labor Time and Costs:**
 - Development Team (500 hours at \$50/hour): \$25,000
 - QA Team (150 hours at \$40/hour): \$6,000
 - Project Management (100 hours at \$60/hour): \$6,000
- **Application Costs:**
 - Deployment and Integration: \$3,500
 - Cloud Hosting (annual): \$4,500
 - Maintenance and Support (annual): \$5,500

Part D: Post-implementation Report

Solution Summary

The core issue was that there was no personalized content recommendations leading to decreased user time-on-site and missed ad revenue potential. Our application addresses this problem by using advanced machine learning algorithms to curate tailored gaming news and information, ensuring a personalized user experience. By utilizing an unsupervised learning algorithm with an expansive and accurate dataset, we made improvements in user loyalty, increased ad revenues through higher engagement rates, and positioning us as a market leader of gaming news providers.

Data Summary

The data was sourced from here:

<https://www.kaggle.com/datasets/arnabchaki/popular-video-games-1980-2023>

Unnecessary columns were deleted, keeping only the title and genres. The genres were split up into separate columns using a process called “One Hot Encoding.” Duplicate titles and non-descript genres were removed.

Data is processed by running through a cosine similarity matrix to find similar games to whatever title is searched. New games added to the list are first vetted for legitimacy. Fine tuning of a games genre is further tuned by user voting and reporting to help tag appropriate labels.

Machine Learning

What?

The product is a recommendation engine that integrates with the Gaming News website to suggest similar or related video game articles to users. The goal is to enhance user engagement by seamlessly guiding them to content that resonates with their interests.

The heart of this product is powered by Machine Learning, which uses a genre-based filtering method to discern and anticipate user preferences. The machine learning aspect leverages cosine similarity—a tool that gauges the likeness between documents regardless of their volume—across various features to categorize video games of differing genres.

Algorithms and Python Libraries used:

Cosine Similarity: Algorithm to compute the degree of similarity between a video games genres.

Scikit-learn: Library for Python, used for it's efficient tools for data mining and data analysis, including the implementation of the cosine similarity metric.

Pandas: A library providing easy-to-use data structures and data analysis tools for Python.

How?

Data was collected of released video games from targeted years. Most importantly, each game had labeled genres, ensuring that it would be easier to classify and compare games. Following this, a cosine similarity matrix was constructed to quantify the likeness between game features.

Main Functions used in the program:

Data Loading: games_edit.csv is imported using Pandas, forming the initial dataset that includes various features related to video games.

Cosine Similarity Matrix: With Scikit-learn, we computed a similarity matrix that encapsulates the cosine similarity scores between all pairs of games in the dataset.

Recommendation Function: This function ranks games based on their similarity score to the user's currently viewed game and suggests the top matches.

Model Improvement:

Since it is an unsupervised learning application, future improvements can be made by adding more games with labeled genres into the list. Genre types can be more specific and diverse to get more fine-tuned recommendations. User voting and feedback can help to ascertain areas that need to be worked on.

Why?

Algorithm Choice:

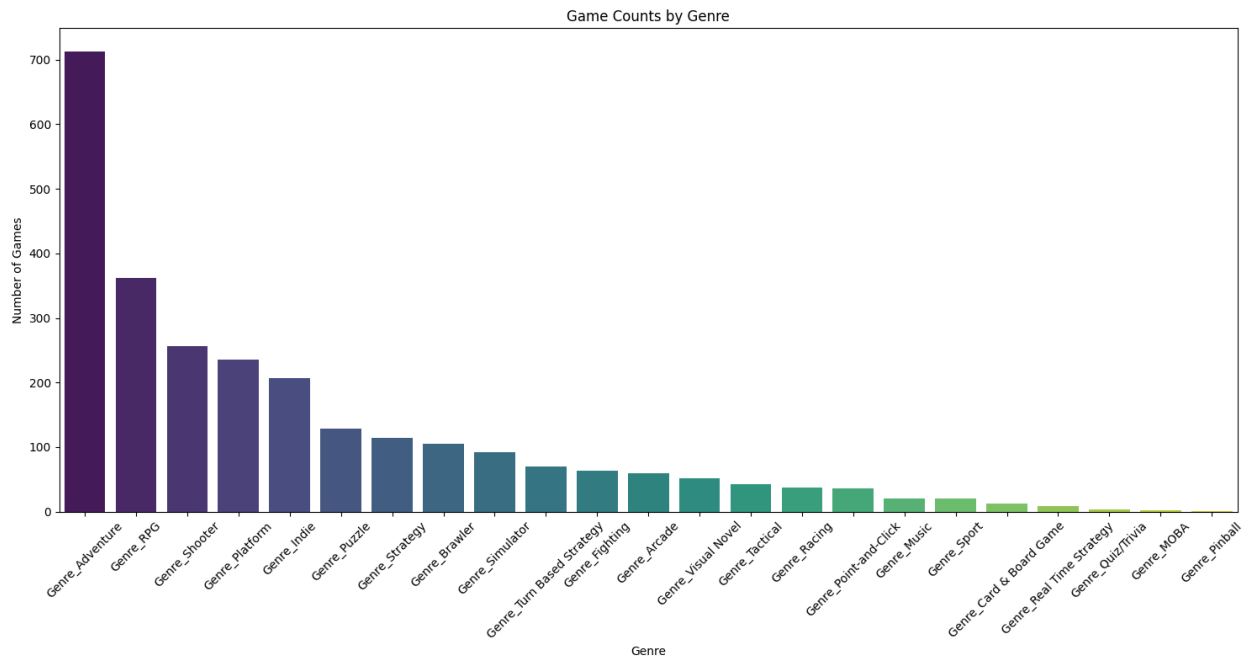
Cosine similarity is well-suited for this application because it handles large datasets efficiently, crucial for a website with an expanding library of game articles. It measures similarity based on content features, which aligns with the goal of recommending games with similar attributes.

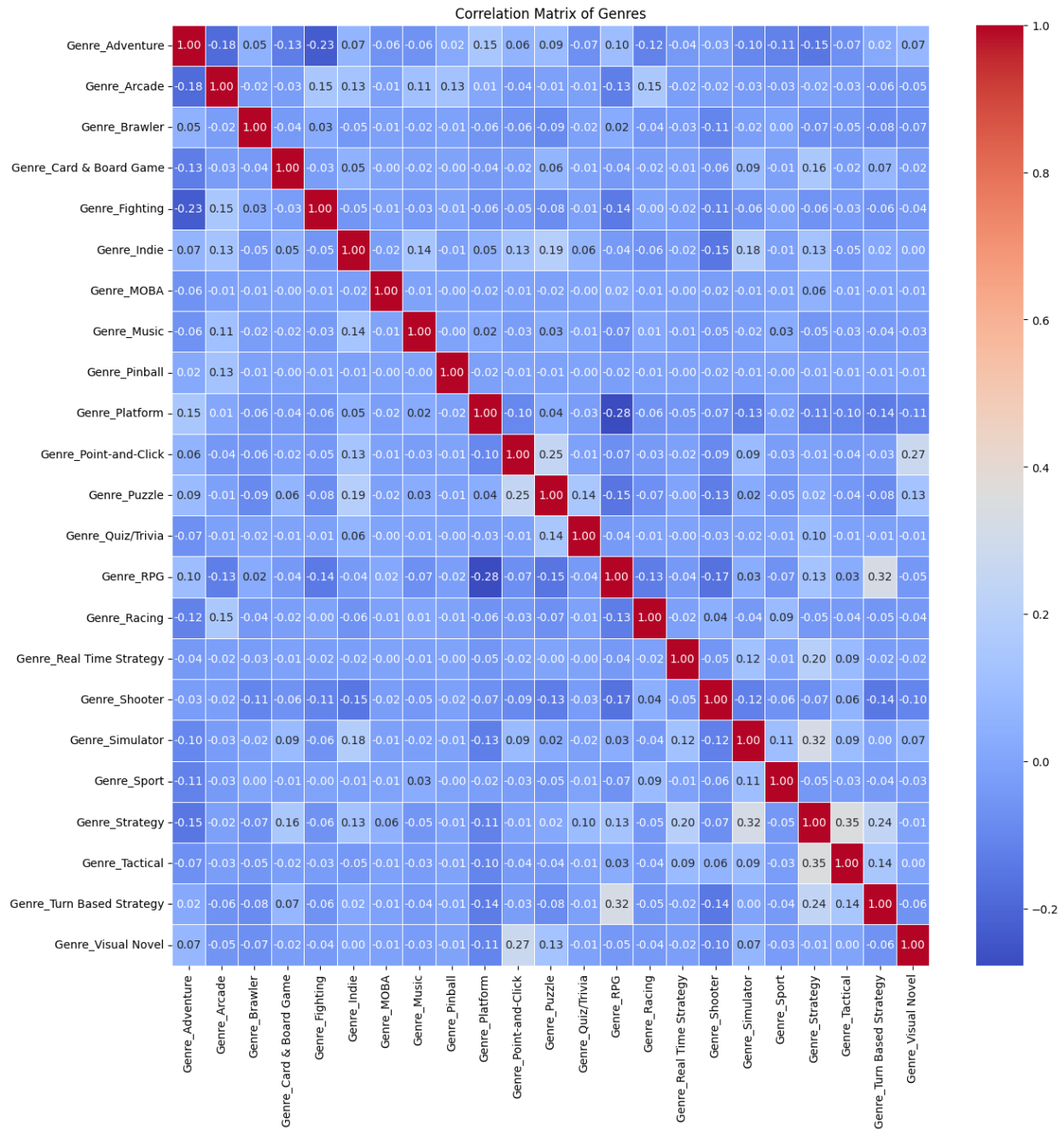
Validation

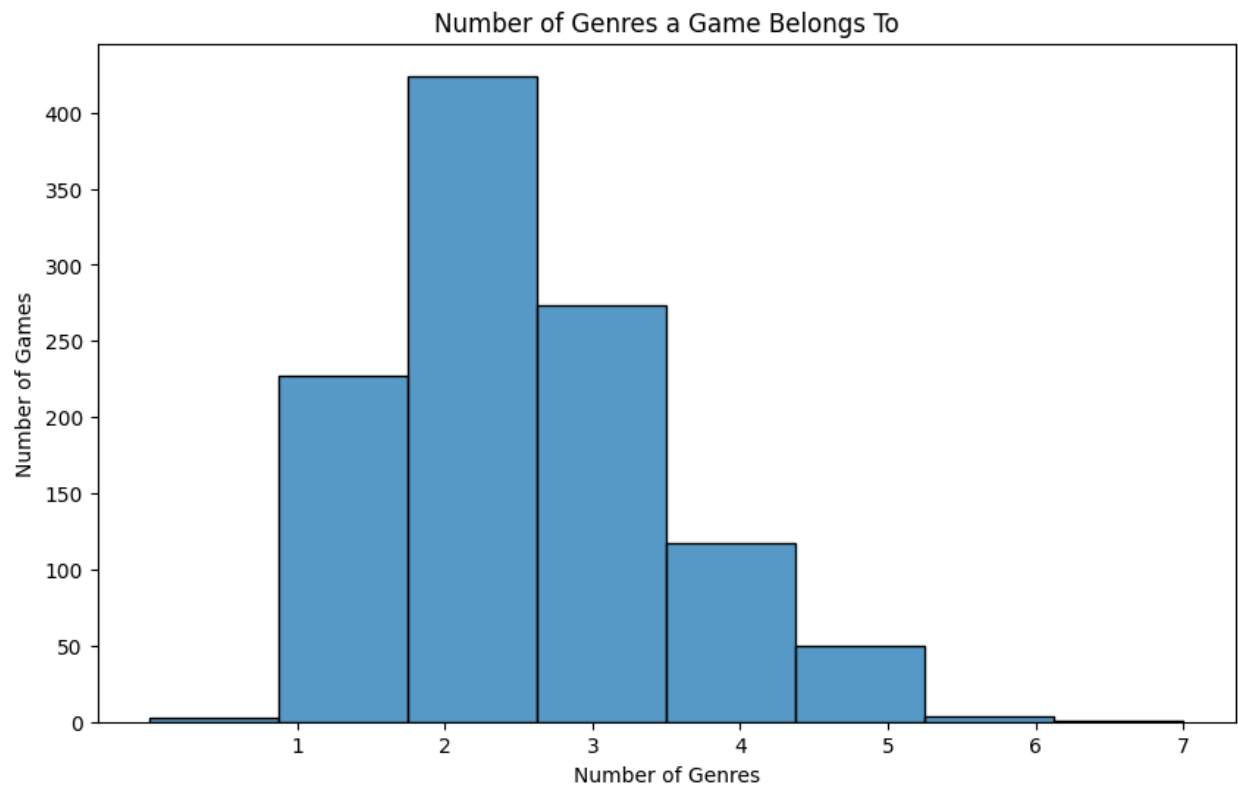
Traditional supervised learning metrics do not apply directly because our task is to recommend items rather than predict a numeric value or class. Instead, we look at user engagement metrics to assess performance. Our approach is to measure the Click-Through Rate (CTR) for the recommended games, which indicates the relevance of our recommendations—the higher the CTR, the more users find the suggestions useful. Additionally, implementing a feedback mechanism allows users to rate the recommendations.

Visualizations

You can also find the visualizers in the application:







User Guide

An interactive Jupyter Notebook hosted on Binder by following these instructions:

- Go to this website: <https://notebooks.gesis.org/binder/>
- Insert this link into the GitHub repository link box:
<https://github.com/rickyheng19/C964-Capstone>
- In the Path to notebook file box, change it to URL and paste in the following:
voila/render/app.ipynb
- Press launch to launch the app. It will take a couple minutes to load if it's the first time building the image.



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

New to Binder? Get started with a [Zero-to-Binder tutorial](#) in Julia, Python, or R.

Build and launch a repository

GitHub repository name or URL

GitHub

Git ref (branch, tag, or commit)

URL to open (optional)

URL

Copy the URL below and share your Binder with others:



Expand to see the text below, paste it into your README to show a binder badge:



- Once on the app, simply follow the on screen instructions to receive video game recommendations. Visualizers for the data is also included.

C964 Capstone Project

Video Game Recommender

Enter a video game title to get recommendations.

Press reset to search again.

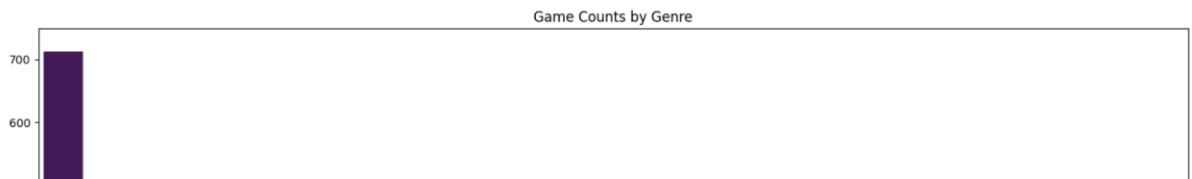
Confidence score ranges from -1.0 (inaccurate) to 1.0 (accurate).

Game Title:

Reset

Visualizers

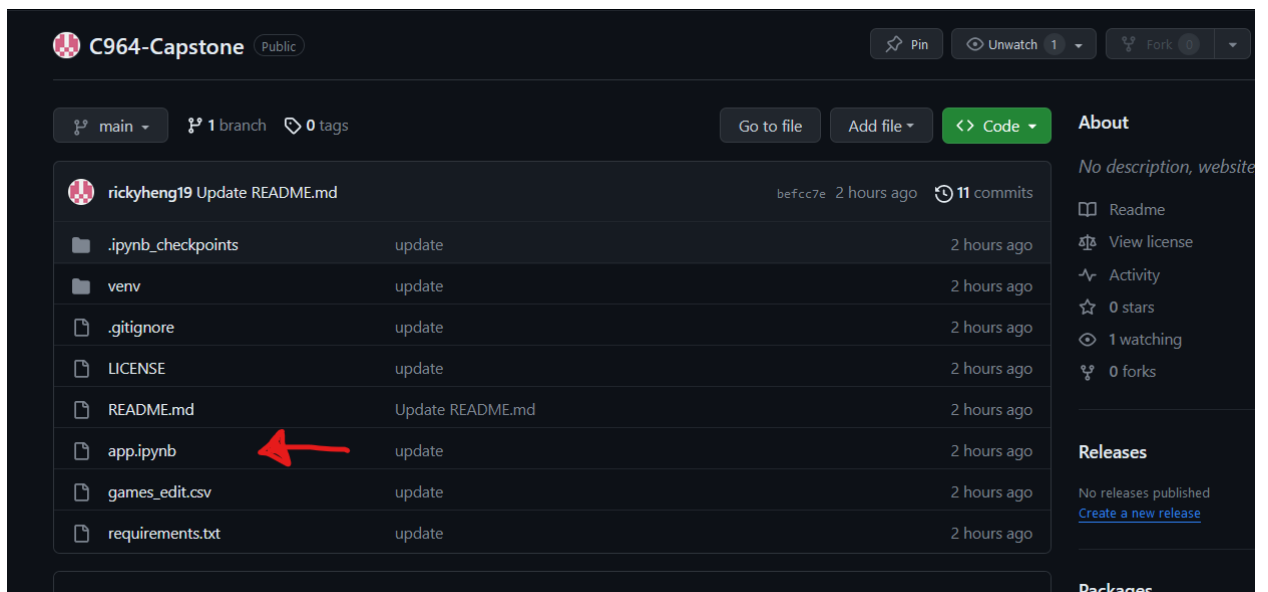
Barplot (Genre Distribution):



- Files for the app can also be viewed in this repository:

<https://github.com/rickyheng19/C964-Capstone/tree/main>

- The program code can be viewed in the app.ipynb file



If app is unresponsive, you can run it on your local machines by following these instructions (for windows 10).

□ **Download and Install Python:**

- Visit the official Python website at python.org, download the latest version for Windows 10, and install it. Ensure you check the box to 'Add Python to PATH' during installation.

□ **Install Required Libraries:**

- Open a command prompt and install the following libraries using pip:
- `pip install voila ipywidgets scikit-learn IPython pandas matplotlib seaborn wordcloud`

□ **Launch Voila:**

- Make sure you are in the directory containing the files found in the repository in the command prompt.
- Run the following command: `voila app.ipynb`

You should be able to run the notebook and view it as a web application via Voilà on your local machine.