Movie Project

BZAN 544

Spring 2020

# Objective

Implement “server-side” analytics in an app to visualize Cinema Showtime schedules.

# Flow

Theatres booking and theatre detail csv files will be stored in subfolders of DataIn.

The files will be read in. If there has not been a schedule created previously, then a schedule will be created (by creating and solving a model) and stored in the subfolder Schedules.

Once all the schedules have been created, the app will present the user with the option to choose a schedule to visualize. The app will display the schedule as a Gantt chart and a Table with movie start times etc. When the user changes the file selection, the presented schedule will change.

# File Structure

We will organize our app using folders. You will zip the folder containing your app before submitting it.

## Project Folder

Start with a new folder. Name it whatever you like. I named mine Movie\_Project.

Your project folder will have two subfolders named DataIn and Schedules. It will also contain at least 3 python files and a readme file. I also suggest that you store any notebooks or documentation at this level.

### Subfolder: DataIn

This folder will contain subfolders with 2+ files each. We will develop a schedule for each of DataIn’s subfolders. In fact, we will use the subfolder name as the name for the schedule we store.

Each subfolder of DataIn MUST contain files named Theatre\_Bookings.csv and Theatre\_Details.csv.

### Subfolder: Schedules

This folder will hold the schedules that have been created. Each schedule file MUST end in “\_Schedule.csv”

### app.py

You will have a file named app that will run your app. It MUST be named app.py.

### Model python file

This file will define a function that takes in 2 dataframes and other arguments to create and solve a model. It returns a dataframe.

### Schedule Creating python file

This file will define a function that compares the contents of the DataIn and Schedules folders. Then it will create schedule csv files as necessary. This file will call the function in your model python file. The function in this file will be called by app.py

# App requirements

## Words

There should be text to orient the user. Make sure that one can tell what your app does just by looking at it. Make sure that your app is easy to use.

## User selects schedule

Create a component that allows the user to select a schedule to view. What the user sees should be user friendly. (Don’t show the user a lot of options that all end in the same string.)

## Gantt chart

Create a Gantt chart. Make sure that it has all the features that are present when we run movie\_LP.py.

Additionally, add blocks for the trailers, adds, and clean time. Make sure that these groups are colored consistantly. (This feature will require modifying the file that runs your model.)

Use annotations to make the schedule as easy to interpret as possible.

## Start Times Table

Create a table to display the Movie, theatre, start time in an easy-to-read format.

## Optional

Create a button to allow the user to download the schedule in a csv format.

# Expectations

I will be able to unzip the folder you created and place it anywhere on my computer (and it should run).

I will be able to add a subfolder to your DataIn folder containing two appropriately named files. Then, I will be able to run your app.py with the command python app.py from your project folder. Your app file will create a new schedule for my files and place the schedule in your Schedules subfolder. When I go to the appropriate url, I will see your dashboard and my schedule will be in the list to pick from. After picking it, I will see the schedule your code created in the form of a Gantt chart and a table.

**Make sure that you set the time limit to one minute!** If you hang my computer, you will lose points.

# Grading Breakdown

|  |  |
| --- | --- |
| “Expectations” works without issues or errors | 55% |
| Gantt chart | 25% (annotation, 5%) |
| Table | 10% |
| Layout/Words | 5% |
| Over-Achieving | 5% |