ETL Mini Project

By: Marta Baker, Paris Lee, and Earl Lewis

Introduction

This project involved designing and implementing a database for managing crowdfunding data. The dataset included information on 1000 crowdfunding projects and their outcomes, as well as contact information for each project, which was stored in a separate table. The primary goals are to create a robust database using appropriate keys and constraints, develop an ETL pipeline to process and load the data, and perform data analysis to generate insights.

Database Design Considerations

When creating the database, we had a few goals that included ensuring there were no null values, and removing extraneous columns. Additionally, we needed to create look-up tables for the categories and subcategories and we had to extract information for the contacts table that had been stored in 1 column as a dictionary.

The entity-relationship diagram (ERD) was created with the use of Quick Database Diagrams. It was created after the data transformation was conducted, which helped to ensure that the correct tables were being used in the schema with the correct columns. This helped prevent unnecessary errors and backtracking. Special consideration was taken when designing the database regarding the order of table creation, so as to avoid foreign-key errors. The final ERD included 4 tables, 2 of which were look-up tables. These look-up tables were key additions that helped to increase efficiency, simplify the storage of the data, and add to the overall robust nature of the database design.

Extract/Transform/Load (ETL) Code Overview

Extract

Data was extracted from Excel files. The data included 2 tables including one table with information on crowdfunding campaigns, the relevant categories, subcategories, campaign outcome, campaign launch date, and when the campaign ended among other information. The other table had data showing the relevant contact person for each campaign.

Transform

The data from the tables was transformed in a number of ways. For the 2 look-up tables, we took the “category & subcategory” column provided in the tables with the campaign details, split the column into 2 separate columns and then found the unique values for each. Using numpy to generate arrays matching the number of unique categories and subcategories, 2 separate look-up tables were created and saved off as CSVs. The campaign CSV was created by renaming some columns, adjusting data types, merging the 2 look-up tables so only the category_ids and subcategory_ids were included, and then deleting extraneous columns. Finally, the second Excel file that included contact information was transformed by reading each line in the file as json and then extracting the contact_id, name, and email. From there, the names were split into 2 columns housing the first and last names.

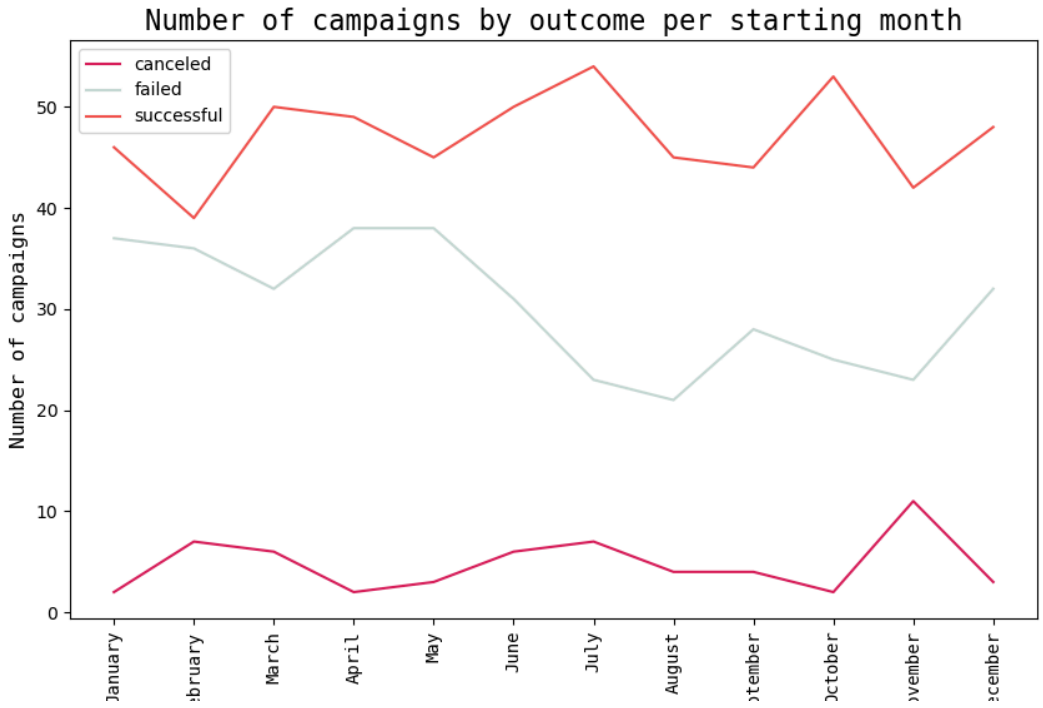
Load

After the schema file was run in PostgreSQL, Psycopg2 was used to load the CSVs into the database with the to_sql function in Pandas.

Analysis

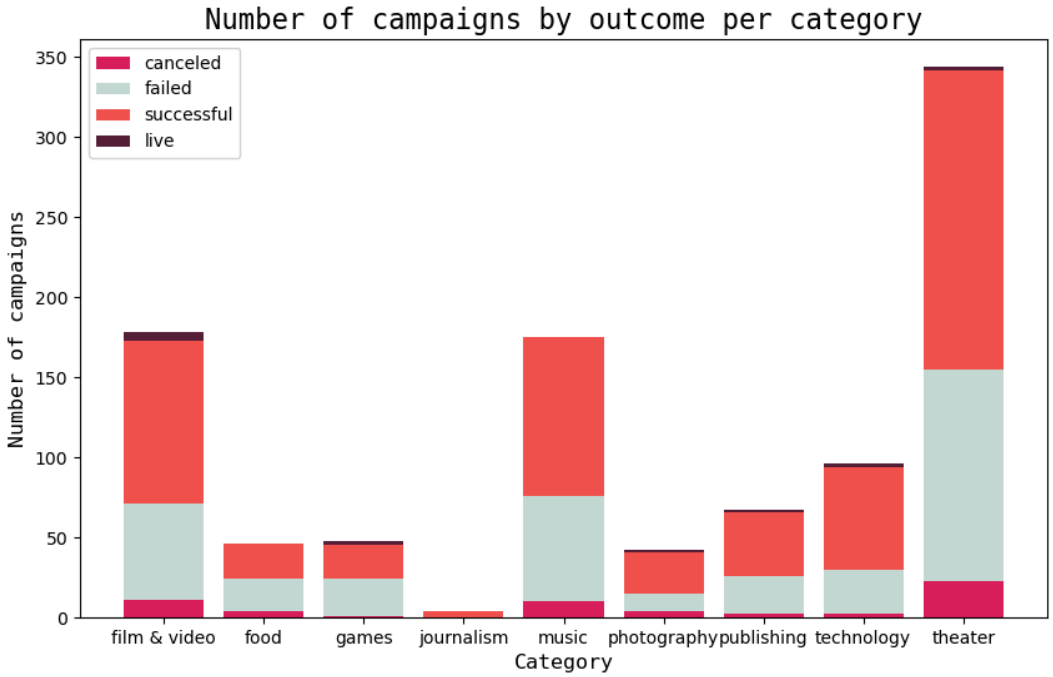
In the analysis, we wanted to recreate some visualizations from the Week 1 homework (where the dataset mimicked the dataset provided here) using the SQLAlchemy ORM.

The first visualization that we sought to recreate was the outcomes by month that projects were started. While there were 4 outcomes, we did not include campaigns that were still live. We ran 2 separate queries against the database to create the necessary DataFrame for the analysis. The first query only extracted the launch date from the Campaign table. Once we had the launch date, we used the month property in the Pandas to_datetime function to extract the month and added it to a list named “months”. Next, we queried for the outcome and country from the campaign table. While the country was irrelevant, we could have used that to add an additional filter on the visual we ended up with. After we had the information we needed, we added the months list to the DataFrame with the outcome and country. With the information we had, we were able to create the following multi-trace line chart.



The second visualization looked to get the outcomes per each category. In order to do this, we queried against the database to get the outcome from the campaign table and the category from the category table joined on the category_id from both the campaign and category tables. With this information, we were almost able to create the desired stacked bar chart. However, because, with the exception of successful campaigns, not all categories had

specific outcomes. For example, all ‘journalism’ campaigns were successful (although there weren’t many of them). Additionally, the categories were read as the index, which added an extra element to solve the problem. In order to remedy this, we had to create new DataFrames for each outcome type that included a value of 0 in the outcome column and had to set the index for the category. Using the new DataFrames, we concatenated them with the pre-existing DataFrame for the specific outcome and then reindexed the tables so the indices were in the same order for all of the categories. Following this work, we were able to create the following stacked bar chart.



Bias/Limitations

There were a number of limitations on the dataset that could have introduced bias. For example, the dataset was unlikely to include all crowdfunding projects. As a result, it’s unknown if the outcomes that were found are representative. Another limitation included that the dataset is static. Because the data isn’t being updated, it might not reflect current trends, outcomes, or category popularities.

Conclusions/Reflection

This project showed how to read Excel files into Jupyter Notebook, how to transform them into CSVs that can be loaded into PostgreSQL, and how to write queries against a PostgreSQL database using both SQLAlchemy and Psycopg2. The project provided a good example of how an ETL pipeline works and how various coding languages and layers can be used to provide security to the database.

The visualizations provided insights into how campaign start dates might influence the outcome of the campaign and into various categories' performance. However, the analysis is

limited by potential biases in the dataset and the static nature of the data. Future work could involve integrating real-time data updates and expanding the dataset to include more diverse projects.