BYTE &
BEYOND

25 APRIL 2025

INTRODUCTION TO CLOUD DATA ENGINEERING

Google Colab + Cloud Storage Basics

PRESENTED TO

Francis Makombe

PRESENTED BY

Byte and Beyond

TABLE OF CONTENTS

PROJECT SUMMARY

Introduction

This project focuses on building a simple ETL (Extract, Transform, Load) data pipeline using Python in Google Colab. The aim is to demonstrate fundamental data engineering concepts, including data ingestion, transformation, and storage. As part of a group assignment, we implemented a pipeline that extracts data from a CSV file stored in Google Drive, performs necessary data cleaning and transformation operations, and then saves the cleaned data to a new CSV file.

We selected a dataset titled "Food_Drink.csv", which contains entries such as the number of downloads and reviews with various formats and abbreviations (e.g., "1K", "1.2M", "10L", "2Cr", etc.). Our transformation logic focuses on standardizing these values into a consistent and human-readable numerical format. The entire process is logged step-by-step using a custom logging function, which records progress to a separate log file for traceability.

The tools used include Google Colab, Python, and Pandas, all of which allowed for effective data handling, processing, and documentation. This project serves as an introductory hands-on exercise into how real-world data pipelines operate in a simplified environment.

IMPLEMENTATION OF CODE

The ETL pipeline is implemented in Python using Google Colab, with Google Drive integration for data access and storage. The implementation consists of three main phases—Extract, Transform, and Load—each structured into reusable functions for modularity and clarity. Here's a breakdown of the implementation:

Extract phase

```
DEF EXTRACT():
                    DF          =
PD.READ_CSV('/CONTENT/DRIVE/MYDRIVE
/COLAB NOTEBOOKS/FOOD_DRINK.CSV')
  RETURN DF
```

The extract() function reads a CSV file named "Food_Drink.csv" from Google Drive using the pandas.read_csv() method. This function loads the dataset into a pandas DataFrame for further processing. The file path is hardcoded and assumes prior mounting of Google Drive.

Load phase

```
DEF LOAD_DATA(TARGET_FILE, DF):
    DF.TO_CSV(TARGET_FILE,
INDEX=FALSE)
```

The load_data(target_file, transformed_data) function saves the transformed DataFrame to a new CSV file in Google Drive. This serves as the final output of the ETL process.

Load phase

```
DF['REVIEWS'] =
DF['REVIEWS'].APPLY(CONVERT_AND_FOR
MAT)
DF['DOWNLOADS'] =
DF['DOWNLOADS'].APPLY(CONVERT_AND_F
ORMAT)
```

The transformation(df) function applies cleaning and formatting operations on two columns: Reviews and Downloads. These fields contain numerical values with abbreviations like 'K', 'M', 'Cr', etc. A helper function convert_and_format(val) is used to:

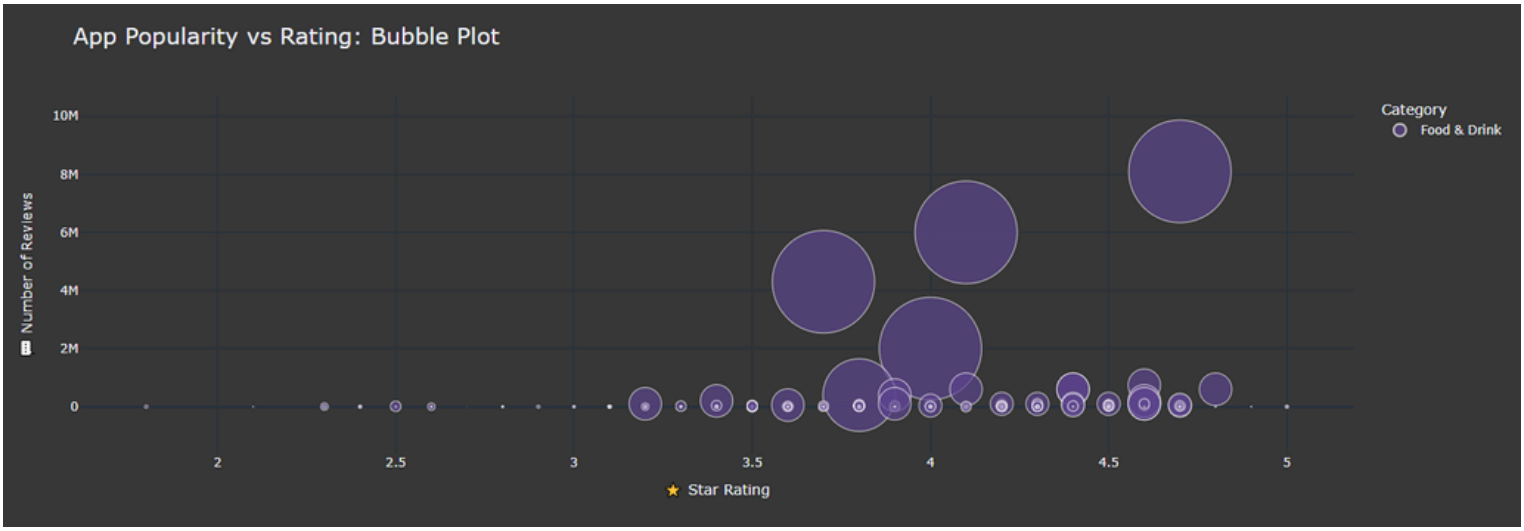
- Convert these abbreviations to full numeric values.
- Format the output with space-separated thousands for better readability.
- Handle missing or malformed entries gracefully.

DATA VISUALIZATION

To enhance the interpretability and creativity of our data insights, we developed an interactive Bubble plot. This visualization represents total number of reviews across different star ratings for each category

We customized the chart with a vibrant color scale and dynamic hover text. Marker sizes are also proportional to sales volume. These enhancements not only elevate the visual appeal but also make the data exploration more engaging and insightful for end users.

The code for this scatter plot will be included in our google colab notebook.



CHALLENGES FACES AND SOLUTIONS

01

Handling Abbreviated Numerical Values (e.g., 'K', 'M', 'Cr')

A custom transformation function was developed using string manipulation and conditional logic to convert these abbreviated values into standard numerical formats. The function also included error handling for malformed or missing entries.

02

Logging Progress for Debugging and Monitoring

A logging function (log_progress) was implemented to write timestamped messages to a log file. This made it easy to monitor execution flow and detect any failures during different ETL phases.

03

File Access and Path Management in Google Colab

Google Drive was mounted within the Colab notebook using drive.mount(), and all files were accessed using absolute paths within the mounted directory. This ensured persistent access to the input dataset and allowed saving output files to Drive as well.

04

Missing or Inconsistent Data

The transformation function was designed to check for NaN values and handle unexpected types safely using pd.isna() and try-except blocks. This prevented the pipeline from crashing and ensured data integrity.

05

Balancing Simplicity and Creativity

The dataset sourced from Kaggle contained missing values in key columns like Description and CustomerID, as well as formatting inconsistencies in InvoiceDate.

APPENDIX

Tools & Technologies Used

- Google Colab – Cloud-based Python notebook environment used for running code and visualizations.
- Google Drive – Used for storing the original dataset and saving the cleaned dataset.
- Pandas – Python library for data loading, cleaning, transformation, and manipulation.
- Plotly – For creating interactive and visually appealing 3D scatter plots.
- Kaggle – Source of the open dataset used in this project.

Dataset Source

- Title: Food And Drink Data
- Source: Kaggle
- URL: https://www.kaggle.com/datasets/shakthidhar/google-play-store-category-wise-top-500-apps?select=Food++Drink.csv

File Paths Used in Google Colab

- Input File:
/content/drive/MyDrive/Colab Notebooks/Food_Drink.csv

- Output File (Cleaned Dataset):
/content/drive/MyDrive/Colab Notebooks/transformed_data.csv

QUESTIONS?
CONTACT US.

BYTE &
BEYOND

www.byteandbeyond.com
hello@byteandbeyond.com
123-456-7890