Technion – Israel Institute of Technology **The Faculty of Data and Decision Sciences**

096224: Distributed Database Management

Spring 2025
Project Assignment 1

Background

As a talented group of students who are learning how to process big data, you have been called upon to assist the "Mayor of Townsville". While the Powerpuff Girls have been doing an amazing job of protecting the city from villains and chaos, they are faced with new challenges only YOU know how to solve with what you have learned in this class!

In this assignment, you will protect the city of Townsville from getting brainwashed by Mojo Jojo (oh no!), who has messed with the TV programs offered to the people of Townsville.

Dataset

The data in this project is divided into 4 files.

Reference Data

Contains information that maps devices to a household.

- **Device ID** - A unique identifier of the device.
- **DMA** - The Nielsen defined designated market area.
- **DMA Code** - The DMA code as defined by Nielsen for each DMA.
- **Household ID** - A unique identifier for each household.
- **Household Type** - A descriptor mapping each household to its generating company.
- **System Type** - The system type for the data collection application.

Daily Program Data

Contains information on programs.

- **prog_code** - Unique program identifier.
- **prog_title** - Program title of a movie, show, episode, or sports event.
- **prog_genres** - Comma separated words or groups of words that classify a show, episode, movie, or sports event.
- **air_date** - Air date of the program (YYYYMMDD).
- **air_time** - Air time of the program (HHMMSS).
- **duration** - Duration of the program in minutes.

Program Viewing Data

Contains information to map an airing event to a program.

- **mso_code** - Unique 5-digit code assigned by TMS to MSO.
- **device_id** - Globally unique device identifier.
- **event_date** - Date viewing event occurred (YYYYMMDD).
- **event_time** - Time viewing event occurred (HHMMSS).
- **station_num** - Station ID number.
- **prog_code** - Unique program identifier.

Demographic Data

Contains various household characteristics information.

- **household_id** - The unique and persistent identifier for the household.
- **household_size** - Number of people residing in the household.
- **num_adults** - Number of adults residing in the household.
- **num_generations** - Number of Generations in Household.
- **adult_range** - A set of flags to indicate the age/genders residing in the household.
- **marital_status** - Code indicating marital status of the head of household.

- **race_code** - Code to indicate the race of the head of household.
- **presence_children** - Presence of Children in the household flag. ('Y' represents the presence of children, 'N' represents that there are no children present and 'Blank' if it is unknown)
- **num_children** - Number of children residing in the household.
- **age_children** - Age of each child residing in the household with one year granularity.
- **age_range_children** - A set of flags to indicate the age/genders of the children residing in the household.
- **dwelling_type** - Dwelling type indicator code.
- **home_owner_status** - Home owner/renter code.
- **length_residence** - Length of time in years that the head of household has lived in the household.
- **home_market_value** - Value of the household.
- **num_vehicles** - Number of vehicles owned by an adult residing in the household.
- **vehicle_make** - Make code of the first (newest) vehicle in the household.
- **vehicle_model** - Model code of the first (newest) vehicle in the household.
- **vehicle_year** - Year of the first (newest) vehicle in the household.
- **net_worth** - Net worth of the household.
- **income** - Household income. 0-9 and then A-D for higher values.
- **gender_individual** - Gender of the individual used to identify the household.
- **age_individual** - Age household member used to identify the household in 2 year increments.
- **education_highest** - Code that indicates the highest level of education.
- **occupation_highest** - Occupation that indicates the highest level in the household.
- **education_1** - Code that indicates the highest level of education of the head of household (1st individual).
- **occupation_1** - Occupation of the head of household.

- **age_2** - Age in two year increments of the second adult residing in the household.

- **education_2** - Code that indicates the highest level of education of the second adult in the household.

- **occupation_2** - Occupation of the second adult in the household.

- **age_3** - Age in two year increments of the third person residing in the household.

- **education_3** - Code that indicates the highest level of education of the 3rd individual in the household.

- **occupation_3** - Occupation of the 3rd Individual residing in the household.

- **age_4** - Age in two year increments of the 4th person in the household.

- **education_4** - Code that indicates the highest level of education of the 4th individual in the household.

- **occupation_4** - Occupation of the 4th Individual residing in the household.

- **age_5** - Age in two year increments of the 5th person in the household.

- **education_5** - Code that indicates the highest level of education of the 5th individual in the household.

- **occupation_5** - Occupation of the 5th Individual residing in the household.

- **polit_party_regist** - Political party indicator for the head of household from registration information.

- **polit_party_input** - Political party indicator for the head of household from individual input.

- **household_clusters** - Personicx cluster codes.

- **insurance_groups** - Insurance classification via Personicx.

- **financial_groups** - Financial classification via Personicx.

- **green_living** - Green Living is an indicator of environmentally friendly households.

Assignment

In this assignment, you will use PySpark to deal with large-scale data processing. The assignment is divided into 2 parts:

Part 1 - Brainwash Detection

After countless failed attempts to defeat the Powerpuff Girls, Mojo Jojo is making one last desperate move—tampering with the TV data to brainwash the people of Townsville!

The Powerpuff Girls need your help to identify and remove malicious data. Luckily, they managed to recover the algorithm Mojo Jojo used to select corrupted records, but their data skills don’t match yours.

Your mission: Detect the **malicious titles** - titles of series or films, in the ”Daily Program Data” table, that Mojo Jojo has tampered with. Since a series can have multiple episodes, each title can be associated with multiple ’prog_code’ values.

To identify them:

- First, flag malicious records—program airings that satisfy **at least 4** of the 7 conditions mentioned below.
- Then, if over 40% of a title’s records are malicious, label the title as malicious.

The conditions for determining malicious records are:

1. The ’duration’ of the program is more (>) than the average duration of all programs in the dataset.
2. The ’prog_code’ of the record was viewed (at least once) by a device associated with a household with a vehicle made by ’Toyota’ (make code ’91’).
3. The ’prog_code’ was watched by a device from a family with exactly 2 adults and the difference between their ages is less than or equal to 6.
4. The ’prog_code’ was aired (at least once) on Friday the 13th in any month.
5. The ’prog_code’ was watched (at least once) by a device from a household with more than 3 devices (exclusive) and the income of that household is lower than the average household income in the data.
6. The program contains at least one of the genres [’Collectibles’, ’Art’, ‘Snowmobile’, ’Public affairs’, ’Animated’, ’Music’]. (case sensitive)
7. The title of the program contains at least two of the words [’better’, ’girls’, ’the’, ’call’] (case insensitive)

1.1 Extract and Transform (15 points)

a) Load data, preprocess it, and apply transformations that will help you in solving the rest of this part of the assignment.

For simplicity, you may filter out unnecessary columns and may also filter records by value, keeping just the relevant data for this part.
Additionally, add **at least** 3 new helpful columns to any dataframe you'd like in order to optimize calculations.

In the Dry-PDF of this part, briefly describe the columns you added and their purpose. Feel free to include examples from your code and outputs.

1.2 Computation and Detection (35 points)

a) Notice that the data is divided among many schemata. One approach is to join all four schemata into a single schema and work with it exclusively. However, other methods exist.

Suggest another solution and write at least 2 pros and 2 cons for the solution we've suggested (joining all 4 schemata into one).

Your written answer should be in the Dry-PDF file of this part.

b) Find the **malicious titles** in the data. You may use your results from 1.1.

First, find all the records (program airings) where at least 4 out of the 7 conditions presented above are met. Then, determine the titles that over 40% of their records are malicious records.

Finally, display the top-20 titles ordered by percentage of malicious records.

Tips and Notes

- a 'prog_code' can be found malicious at instances and not malicious at other instances, that is because Mojo Jojo was smart enough to tamper with some airings of the same episode or program but not with others.
- The data might contain missing values and duplicates that might impact your performance and results. If a given record lacks the required information for a condition, this record is automatically labeled as a negative for that condition.
- In cases of "clashing" records (e.g. same prog_code with different genre) you can rely on the tie-breaking made by spark.
- Where applicable convert letters to appropriate numbers - A-D = 10-13. Empty/Null values can be dismissed.
- Use `display(df.limit(n))` to display your DataFrames - where n is the number of rows you want to show.

What to submit?

1. **Jupyter notebook (.ipynb)** containing your entire code for parts 1.1 and 1.2. The code must be clear and documented, with markdown cells as clear titles.
   File name: project1_part1_[ID1]_[ID2].ipynb

2. **PDF version of your code** - INCLUDING OUTPUTS.
   File name: project1_part1_[ID1]_[ID2].pdf.

3. **HTML version of your code** - INCLUDING OUTPUTS.
   File name: project1_part1_[ID1]_[ID2].html.

4. **PDF file (.pdf) containing your answers to the dry questions** - Please describe your considerations and explain your decisions (Why you chose to filter the columns and why you chose to add certain new columns and how they can help save time, etc.).
   File name: project1_part1_dry_[ID1]_[ID2].pdf

Part 2 - Un-brainwashing

Congratulations! You have helped the Powerpuff Girls identify the malicious data!

Unfortunately, many are still under the effects of Mojo Jojo's brainwashing!

The good news is that the Powerpuff Girls have discovered that certain TV programs have the power to reverse the effects of the brainwashing! However, the Mayor only has 5 broadcast slots available — so we must choose wisely.

2.1 Identifying Slots (30 points)

In this subpart, your task is to help the Powerpuff Girls identify the top 5 slots (by genre, DMA, and program) that will help "un-brainwash" the largest number of citizens using the viewership data.

You will do so by running the following three queries:

- **Display** the top 5 most popular genres, by the amount of people who viewed it ('household_size') and **print** the total number of people who viewed those genres.
- **Display** the top 5 most popular DMAs (by amount of devices) and **print** the total number of people who reside in households in those DMAs.
- **Display** the top 5 most popular programs ('prog_title'), by the amount of people who viewed it that live households with children present ('presence_children') and **print** the **total** number of people who viewed them.

In the Dry-PDF file of this part, discuss based on which criteria (by genre, DMA, and program) the mayor should choose the 5 slots to display the "un-brainwashing" programs.

2.2 Money and Corruption (20 points)

Unfortunately, the mayor of "Townsville" was corrupted by money. Instead of treating every DMA equally, he wants to consider some DMAs more than others according to their wealth score, determined by their net worth and income.

For each DMA, calculate their wealth score, which is the normalized average of the net worth and the income:

$$\frac{\#avg\_net\_worth\_in\_dma}{\#max\_net\_worth\_all\_data} + \frac{\#avg\_income\_in\_dma}{\#max\_income\_all\_data}$$

For each DMA in the top 10 wealthiest DMAs return its 11 most popular genres (for some reason they are needed for some mysterious calculation) - with a limitation - a genre can't be duplicated, meaning:

The wealthiest DMA chooses the top 11 genres according to its popularity rating. After that, the 2nd wealthiest DMA chooses its top 11 genres, from a pool that does not contain the previously taken genres.

We continue like this until we finish our DMA list or there are no genres left - and place empty lists at the remaining DMAs.

Your output columns should include each relevant DMA's wealth score and ordered list of genres in this order: [”DMA NAME”, ”WEALTH SCORE”, ”ORDERED LIST OF GENRES”].

Tips and Notes

- You may ignore ”Unknown” DMA in both parts 2.1 & 2.2.
- The data might contain missing values and duplicates that might impact your performance and results. If a given record lacks the required information for a condition, this record is automatically labeled as a negative for that condition.
- Where applicable convert letters to appropriate numbers - A-D = 10-13. Empty/Null values can be dismissed.
- Use `display(df.limit(n))` to display your DataFrames - where n is the number of rows you want to show.

What to submit?

1. **Jupyter notebook (.ipynb)** containing your entire code for parts 2.1 and 2.2. The code must be clear and documented, with markdown cells as clear titles.
   File name: project1_part2_[ID1]_[ID2].ipynb
2. **PDF version of your code** - INCLUDING OUTPUTS.
   File name: project1_part2_[ID1]_[ID2].pdf.
3. **HTML version of your code** - INCLUDING OUTPUTS.
   File name: project1_part2_[ID1]_[ID2].html.
4. **PDF file (.pdf) containing your answers to the dry questions** - Please describe your considerations and explain your decisions. Make sure to clearly state which criteria is best in your opinion and back up your answer accordingly.
   File name: project1_part2_dry_[ID1]_[ID2].pdf

Bonus (2 points)

Congratulations! The day is saved, thanks to [Insert your names].

Attach the following image, or another variation of it, with your **names** inside it, at the end of the Dry-PDF file for 2 bonus points on the assignment grade:)

Good Luck!
Course Staff and the mayor of Townsville

Tips, Guidelines, and Submission Checklist

Tips and Guidelines

- Work on Databricks where all of the data is also provided for you. Use *Spark 3* and higher.
- Write clean code and document functions when necessary.
- Questions related to the project will be answered in the forum, via Moodle, **exclusively**.
- Only one of the team members needs to submit.
- There are 48 hours of automatic extension to the due date on Moodle. Rules are detailed in Moodle forum.
- This assignment is not straightforward and requires creativity:) Enjoy it!
- Use of generative AI tools (e.g., ChatGPT) is allowed, but we recommend using them sparingly and only after trying to solve the problem yourself. If used, briefly state where and how it helped. This won't affect your grade.
- Pretty and visually appealing assignments will be graded more favorably:)

Submission Checklist

- **Part 1 – Code:**
  - `project1_part1_[ID1]_[ID2].ipynb`
  - `project1_part1_[ID1]_[ID2].pdf`
  - `project1_part1_[ID1]_[ID2].html`
- **Part 1 – Dry Questions:**
  - `project1_part1_dry_[ID1]_[ID2].pdf`
- **Part 2 – Code:**
  - `project1_part2_[ID1]_[ID2].ipynb`
  - `project1_part2_[ID1]_[ID2].pdf`
  - `project1_part2_[ID1]_[ID2].html`
- **Part 2 – Dry Questions:**
  - `project1_part2_dry_[ID1]_[ID2].pdf`
- **Bonus (optional):**
  - The Powerpuff Girls image with team members names embedded inside.

Make sure your IDs are included in the filenames exactly as specified, and all files are properly exported with visible outputs.