

Crypto Casino Analytics Pipeline

Purpose

This script is a data pipeline for processing betting/game data: it extracts, merges, flattens, enriches with historical cryptocurrency USD prices via an API, computes USD amounts, classifies by game type, and generates both CSV and Excel analytics reports.

Setup & Paths

- The process determines the base directory of the script.
 - Sets paths for:
 - Input `.rar` file (`complete_data.rar`)
 - Extraction folder for the `.rar` contents
 - Output CSV for cleaned data
 - Cache JSON for storing crypto prices
 - Ensures the extraction folder exists.
 - Checks for the presence of `unar` command-line tool (required to extract `.rar` files).
-

Data Extraction

- Extracts the `.rar` archive to the extraction folder.
 - Recursively searches for all JSON files in the extraction folder.
 - Merges all JSON files into a single data structure.
 - Flattens nested records (especially the `statePlinko` sub-structure) for easier tabular analysis.
-

Data Processing

- Converts timestamps from milliseconds to Python `datetime` objects and extracts dates.
- Loads cached crypto prices if available.
- For each unique cryptocurrency in the data:
 - Fetches historical USD prices from the CryptoCompare API for missing dates.

- Updates the cache with fetched prices.
 - Maps USD prices back to the main dataset.
 - Computes amounts in USD:
 - `amount_usd = amount * price_usd`
 - `payout_usd = payout * price_usd` if `payoutMultiplier > 0`, otherwise uses `amount_usd`.
-

Classification

- Normalizes `game`, `type`, and `gameName` fields for classification.
 - Classifies each record into verticals such as:
 - `sports`
 - `crash`
 - `instant`
 - `live`
 - `slot`
 - `casino_other`
 - Adjusts classification for certain types like `softswiss` and `thirdparty`.
-

Final Data Preparation

- Selects and orders relevant columns for final output.
 - Ensures numeric types are consistent (`expectedAmount`).
 - Saves cleaned, processed data as CSV.
-

Analytics & Summary

- Computes overall financial summary (Total Bet, Total Payout, Net Profit in USD).
 - Generates slot vs sports summary and breakdown by game type.
 - Prepares a sheet of historical coin prices from the cache.
 - Saves all results into an Excel file with multiple sheets:
 - `summary`
 - `main_data`
 - `coin_prices_by_date`
-

Cache Management

- Saves updated crypto price cache for future runs to avoid redundant **API** calls.
-

Output

- Outputs:
 - Cleaned CSV (`cleaned_data_with_crypto.csv`)
 - Analytics Excel file (`analytics.xlsx`)
 - Prints status messages at key stages.
-

Summary:

Script processes raw betting/game data by extracting, merging, and flattening JSON records, **integrating historical crypto-to-USD prices using a dedicated API**, computing USD amounts and payouts, classifying each game by type, and generating both a cleaned CSV and an Excel report with detailed analytics and summaries.

Guide: Using unar on Windows to Extract .rar Files and Run the Script

Purpose

This guide explains how to extract `.rar` files and run the data processing script, which generates CSV and Excel analytics reports.

Step 1: Download unar (RAR extraction tool)

- Go to: <https://cdn.theunarchiver.com/downloads/unarWindows.zip>
 - Download the ZIP file to your computer.
-

Step 2: Extract unar

- Right-click the downloaded `unarWindows.zip` -> **Extract All...**
- Choose a folder, e.g., `C:\unar`
- Inside that folder, you should see `unar.exe`.

Note: You can also leave `unar.exe` in the same folder as the Python script; it will work automatically.

Step 3: Add unar to System PATH (Optional but recommended)

1. Open Start Menu -> type **Environment Variables** -> **Edit the system environment variables** -> **Environment Variables**
 2. Under **System variables**, select **Path** -> **Edit** -> **New**, then add the folder where `unar.exe` is located (e.g., `C:\unar`)
 3. Click **OK** to save.
-

Step 4: Verify Installation

1. Open Command Prompt
2. Type:

`unar`

If installed correctly, you should see the `unar` help text.

Step 5: Prepare the Script

Place the following files in the **same folder**:

- Your `.rar` file (`complete_data.rar`) or the extracted folder `complete_data_extracted` with JSON files
- The Python script (`main.py`) or the executable (`main.exe`)

Do not move these files after this step.

Step 6: Run the Script

Option A: Using Python

1. Install Python from [python.org](https://www.python.org) if it's not already installed.

Make sure to check **Add Python to PATH during installation.**

2. Open Command Prompt in the folder with your script and `.rar` file.

- Hold **Shift**, right-click the folder -> **Open PowerShell/Command Prompt here**

3. Run the following commands:

```
python -m pip install --upgrade pip
python -m pip install -r requirements.txt
python main.py
```

The script will extract the `.rar` file, process the data, and generate:

- `cleaned_data_with_crypto.csv` -> cleaned dataset
- `analytics.xlsx` -> detailed analytics report with summaries and crypto prices

Option B: Using the Executable

- Double-click `main.exe`
 - The program will automatically extract the `.rar` file, process the data, and create the outputs in the same folder.
-

Step 7: Check Your Outputs

Look in the folder where your script and `.rar` file are located. You should see:

- `cleaned_data_with_crypto.csv` -> cleaned dataset
 - `analytics.xlsx` -> detailed analytics report
-

Guide: Cloning the crypto-casino-analytics-pipeline Repo

1. Install Git

- Go to <https://git-scm.com/download/win>
- Download and install Git using default options.
- Ensure “**Git from the command line**” option is selected.

2. Open Command Prompt

- Press `Win + R`, type `cmd`, hit Enter.

3. Navigate to Your Desired Folder Example (Desktop):

```
cd %USERPROFILE%\Desktop
```

4. Clone the GitHub Repository

```
git clone https://github.com/FiverrDevProjects/crypto-casino-analytics-pipeline.git
```

This creates a folder `crypto-casino-analytics-pipeline` with all files and Git history.

5. Go into the Cloned Repository

```
cd %USERPROFILE%\Desktop\crypto-casino-analytics-pipeline
```

6. Install Dependencies

```
python -m pip install --upgrade pip
python -m pip install -r requirements.txt
```

7. Run the Script

```
python main.py
```

Notes

- Always keep the `.rar` file in the same folder while running the script or EXE.
 - No internet connection is required for extraction; only the crypto price fetch requires internet.
 - If the script complains about `unar` missing, make sure `unar.exe` is either in the same folder or installed in your PATH.
-