### ****Report on the Analysis of Provided Web Resources****

#### Resource Analysis

1. **VGInsights**  
   VGInsights provides the most detailed information about video games and was used as the primary data source. Key features:
   * Data on sales, developers, publishers, genres, and other game parameters.
   * The paid version offers advanced metrics such as DAU (daily active users) and MAU (monthly active users).
   * **Limitations:**
     + Analytics are available exclusively for the Steam platform.
     + No information on games in development.
2. **GamePressure (AJAX-based site)**  
   This site is targeted at the general audience and provides basic game information. Main features:
   * Includes data on game titles, ratings, and gameplay features.
   * Was used to supplement the dataset with platforms where VGInsights games are available.
   * **Limitation:** Lacks detailed analytical data.
3. **SensorTower**  
   SensorTower provides detailed analytics for mobile games. Key features:
   * Data on sales, downloads, popular countries, and rankings in app stores.
   * **Limitations:**
     + Analysis is available only for the last week.
     + Most features are locked in the free version.

### Conclusions

Each of the reviewed resources has its strengths and weaknesses. VGInsights is the most detailed source but is limited to the Steam platform. GamePressure provides basic information suitable for a general audience but lacks deep analytics. SensorTower is useful for analyzing mobile games but offers limited data in the free version. Moving forward, it is advisable to combine data from multiple sources to obtain a more comprehensive view of the gaming market.

### Key Attributes Collected and Their Importance for Analysis

#### ****VGInsights****

1. **vgi\_id** – Unique game identifier in VGInsights.
2. **name** – Game title, a key attribute for identification.
3. **released** – Release date, important for trend analysis.
4. **steam\_id** – Steam game ID, used for data matching.
5. **price** – Game price, crucial for assessing affordability and monetization.
6. **units\_sold** – Number of copies sold, helps evaluate popularity.
7. **reviews** – Number of reviews, reflects audience engagement.
8. **followers** – Number of followers, an indicator of audience expectations.
9. **rating** – Game rating, shows quality based on user feedback.
10. **genres** – List of genres, useful for market segmentation.
11. **developers** – Game developers, important for industry analysis.
12. **publishers** – Publishers, influencing marketing strategies.
13. **publisher\_type** – Type of publisher (indie, AAA, etc.), helps understand project scale.
14. **revenue\_vgi** – Revenue generated by the game, a key success metric.
15. **tags** – Tags used to describe the game, useful for categorization.

#### ****GamePressure****

1. **name** – Game title.
2. **platforms** – Platforms where the game is available (PC, PS, Xbox, etc.).
3. **genre** – Game genre, important for analyzing user preferences.
4. **release\_date\_raw** – Release date in raw format, may require processing.

### ****Application of Data Cleaning and Preparation Methods****

In my project, I used several methods to clean raw data, saving time and simplifying further analysis.

#### ****Date Processing****

In the gamepressure\_scrap.py file, I convert dates from formats like "December 13, 2023" to the standard YYYY-MM-DD format. This helps unify the data and simplify processing.

#### ****Handling Missing Data****

If the date format does not match the expected pattern, I assign the value "Unknown", preventing errors during processing.

#### ****Using Jupyter Notebook for Analysis****

In the **Notebooks** folder, I have files like merge\_df.ipynb, which I use for:

* Removing duplicates,
* Filling in missing values,
* Splitting or merging columns,
* Filtering data.

#### ****Data Filtering****

In gamepressure\_scrap.py, I filter data by extracting only the necessary elements from HTML (.select, .find\_previous, .select\_one).

#### ****Using CSV****

When writing data to a CSV file, I check if the file is empty (if file.tell() == 0). This prevents duplicate headers.

### ****Limitations of the Approach****

1. **Bypassing IP Blocking**  
   I did not account for potential website IP blocking, as the sites were responsive during testing. If issues arise, adding multiple proxy servers could ensure stable operation.
2. **Using Multiple Sources**  
   Parsing two websites was necessary because the first one lacked platform information and details on games in development. For more stability (in case of blocking), adding additional sources would be beneficial.
3. **Limited Information on Mobile Games**  
   SensorTower's free version provides very little data, making it difficult to gather sufficient information on mobile games. Finding alternative resources is necessary.