INFO 6210

Data Management and Database Design

[Section 3]

Gathering, Scraping, Munging and Cleaning Data

Assignment 1

Authors

Purvang Jayesh Thakkar 001387983

Megha Ponneti Nanda 001388342

Contents

Abstract

Purpose

Data munging/wrangling process is performed on real world data where the database tables are populated with it.

Pokemon’s statistical data is chosen for data munging and analysis purposes. Pokémon are fictitious animal-like monsters that live in the (of course, also invented) Pokémon world. The most attractive way of describing the Pokémon is that of the Role Playing Games (RPG). Thus, we can statically analyze the wide variety of variables used to describe the Pokémon, to find relationships between them and to cluster the Pokémon according to some criteria. In the rest of the report we will explore the Pokémon and their corresponding variables that appear in the RPGs.

Data Sources

Data having thematic relationship is gathered from 3 different sources.

1. A web scraper
2. A web API
3. CSV format Dataset

A web scraper

Web Scraping, also termed as Screen Scraping, Web Data Extraction, Web Harvesting etc. is a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format, for later retrieval or analysis. Web scraping is used for contact scraping, and as a component of applications used for web indexing, web mining and data mining.

Pokemon's statistical data that includes Name, Type 1, Type 2, Total, HP, Attack, Defence, Sp.Attack, Sp.Defense, Speed are scraped from below website. The data received in the form of JSON is converted into DataFrame, ie. poke_df

Website: https://pokemondb.net/pokedex/all

DataFrame is stored in the variable poke_df.

A web API

A Web API is an application programming interface for either a web server or a web browser.

Advantages of using Web API are:

- It retrieves data in bulk or with great specificity which would be time consuming otherwise.
- It provides a way to access information that doesn't exist on the web and are only stored in a database attic which is hidden from users.
- It automates a news app that needs live data from other sources.
- It provides more direct interface for reading and writing data to a service.

PokeAPI is a full RESTful API linked to an extensive database detailing everything about the Pokémon main game series.

Pokemon’s statistical data that includes Name, Base Attack, base Defense, Base Stamina, Type 1, Type 2, Max_CP, Attack_Probabiility, Base_Capture_Rate, Base_Flee_Rate, Max_Pokemon_Action_Frequency, Min_Pokemon_Action_Frequency are fetched from below API. The data received in the form of JSON is converted into DataFrame, ie. poke_api_df

API: https://pogoapi.net/api/v1/

DataFrame is stored in the variable poke_api_df.

CSV format Dataset

A common text-based data interchange format is the comma-separated value (CSV) file. This is often used when transferring spreadsheets or other tabular data.

Kaggle Dataset is used as the third resource where data is gathered in the form of CSV file. Kaggle is a platform for predictive modelling and analytics competitions in which companies and researchers post data and statisticians and data miners compete to produce the best models for predicting and describing the data.

Pokemon’s statistical data that includes Name,Type 1, Type 2, Total, HP, Attack, Defense, Sp.Attack, Sp.Defense, Color, hasGender, Pr_Male, Egg Group 1, Egg Group 2, Mega Evaluation, Height_m, Weight_kg, Catch_rate, Body_Style are present in the CSV file extracted from below website. The data is converted into DataFrame, ie. kaggle_df

Kaggle Dataset: https://www.kaggle.com/alopez247/pokemon/home

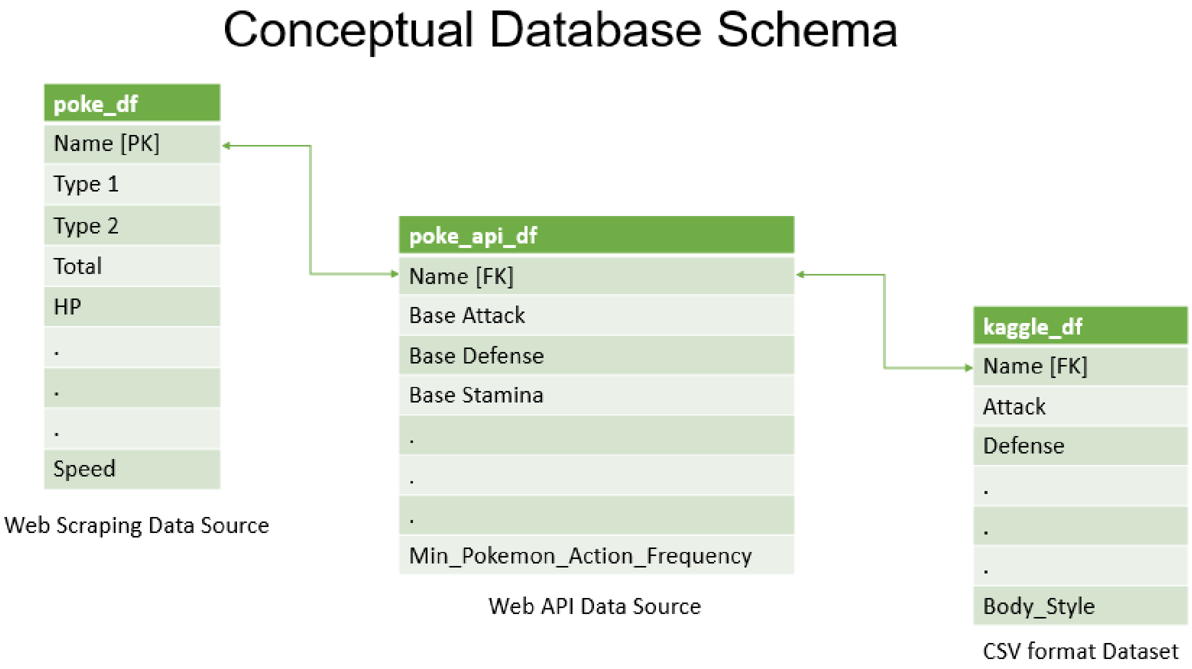
DataFrame is stored in the variable kaggle_df.

Data Fields/Variables

- **Name:** Name of the Pokemon
- **Type1.** Primary type of the Pokémon. This value can take 18 different values: *Bug, Dark, Dragon, Electric, Fairy, Fighting, Fire, Flying, Ghost, Grass, Ground, Ice, Normal, Poison and so on.*
- **Type_2.** Pokémon can have two types, but not all of them do.
- **Total.** The sum of all the *base battle stats* of a Pokémon. It should be a good indicator of the overall strength
of a Pokémon.
- **HP.** Base health points of the Pokémon.
- **Attack.** Base attack of the Pokémon.
- **Defense.** Base defense of the Pokémon.
- **Sp_Atk.** Base special attack of the Pokémon.
- **Sp_Def.** Base special defense of the Pokémon.
- **Speed.** Base speed of the Pokémon.
- **Generation.** The generation where the Pokémon was released.
- **isLegendary.** Boolean indicating whether the Pokémon is legendary or not
- **Color.** Color of the Pokémon according to the Pokédex.
- **hasGender.** Boolean indicating the Pokémon can be classified as male or female.
- **Pr_Male.** In case the Pokémon has Gender, the probability of its being male. The probability of being
female is, of course, 1 minus this value.
- **Egg_Group_1.** Categorical value indicating the egg group of the Pokémon. It's 15 possible values are:
Amorphous, Bug, Ditto, Dragon, Fairy, Field, Flying, Grass, Human-Like and so on.
- **Egg_Group_2.** Similarly to the case of the Pokémon types, Pokémon can belong to two egg groups.
- **hasMegaEvolution.** Boolean indicating whether a Pokémon can mega-evolve or not.
- **Height_m.** Height of the Pokémon according to the Pokédex, measured in meters.
- **Weight_kg.** Weight of the Pokémon according to the Pokédex, measured kilograms.
- **Catch_Rate.** Numerical variable indicating how easy is to catch a Pokémon when trying to capture it to
make it part of your team.
- **Body_Style.** Body style of the Pokémon according to the Pokédex.

Conceptual Database Model

A conceptual schema or conceptual data model is a map of concepts and their relationships used for databases. This describes the semantics of an organization and represents a series of assertions about its nature. Specifically, it describes the things of significance to an organization (entity classes), about which it is inclined to collect information, and characteristics of (attributes) and associations between pairs of those things of significance (relationships).



Conceptual Database Schema

poke_df
Name [PK]
Type 1
Type 2
Total
HP
.
.
.
Speed

Web Scraping Data Source

poke_api_df
Name [FK]
Base Attack
Base Defense
Base Stamina
.
.
.
Min_Pokemon_Action_Frequency

Web API Data Source

kaggle_df
Name [FK]
Attack
Defense
.
.
.
Body_Style

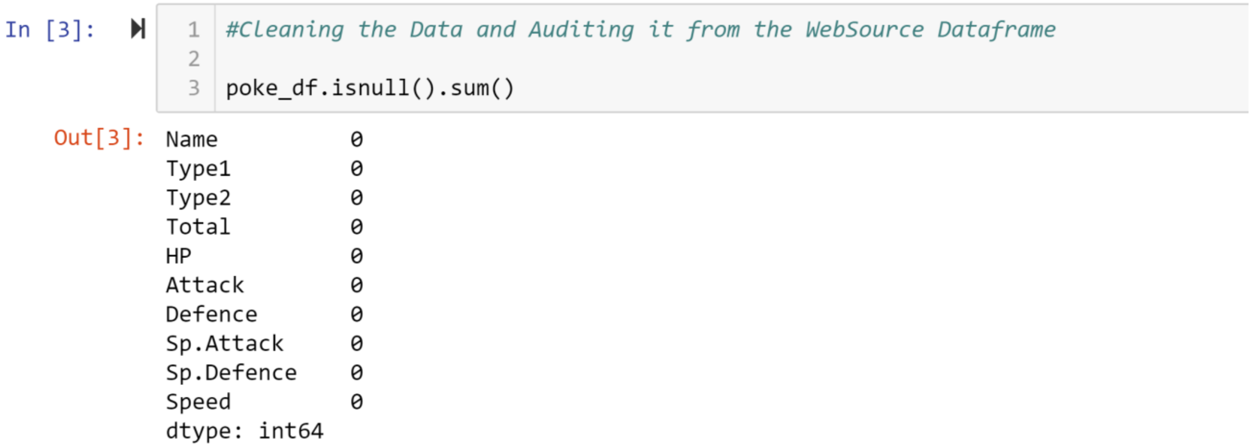
CSV format Dataset

As shown in the above image, ‘Name’ field from poke_df(Web source) table is the primary key which relates to the second table poke_api_df (Web API) on the Foreign Key ‘Name’ and the last table kaggle_df(Kaggle Source CSV) connects the second table again on’ Name’ of the Pokemon. All the three tables are related to each other via ‘Name’ of the Pokemon as the key.

Data Auditing

Data gathered from all three sources are validated by detecting and correcting inaccurate records. The modified data is then checked for consistency and displayed in the form of dataframe.

Data cleaning process is performed on web scraped data as following:



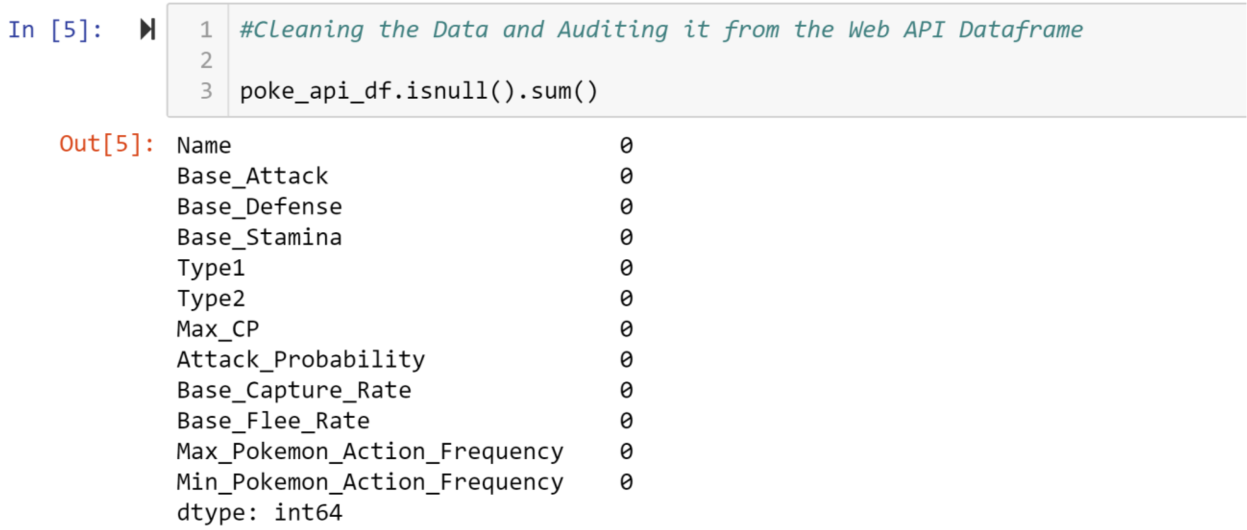
```
In [3]:

#Cleaning the Data and Auditing it from the WebSource Dataframe

poke_df.isnull().sum()
```

```
Out[3]: Name          0
        Type1         0
        Type2         0
        Total         0
        HP            0
        Attack        0
        Defence       0
        Sp.Attack     0
        Sp.Defence    0
        Speed         0
        dtype: int64
```

Data cleaning process is performed on data obtained through web API as following:



```
In [5]:

#Cleaning the Data and Auditing it from the Web API Dataframe

poke_api_df.isnull().sum()
```

```
Out[5]: Name                          0
        Base_Attack                   0
        Base_Defense                  0
        Base_Stamina                  0
        Type1                         0
        Type2                         0
        Max_CP                        0
        Attack_Probability            0
        Base_Capture_Rate             0
        Base_Flee_Rate                0
        Max_Pokemon_Action_Frequency  0
        Min_Pokemon_Action_Frequency  0
        dtype: int64
```

Data cleaning process is performed on data obtained through Kaggle is as following:

Inaccurate data detected were corrected and validated again for data consistency and accuracy purpose.

In [9]:

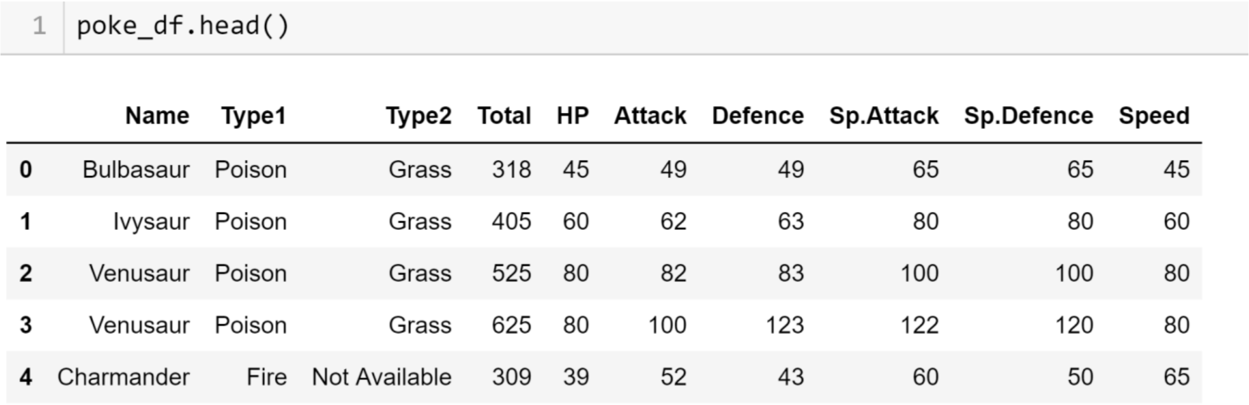
```
kaggle_clean_df.isnull().sum()
```

Out[9]:

```
Number              0
Name                0
Type1               0
Type2               0
Total               0
HP                  0
Attack              0
Defense             0
Sp_Atk              0
Sp_Def              0
Speed               0
Generation          0
isLegendary         0
Color               0
hasGender           0
Pr_Male             0
Egg_Group_1         0
Egg_Group_2         0
hasMegaEvolution    0
Height_m            0
Weight_kg           0
Catch_Rate          0
Body_Style          0
dtype: int64
```

Conclusions

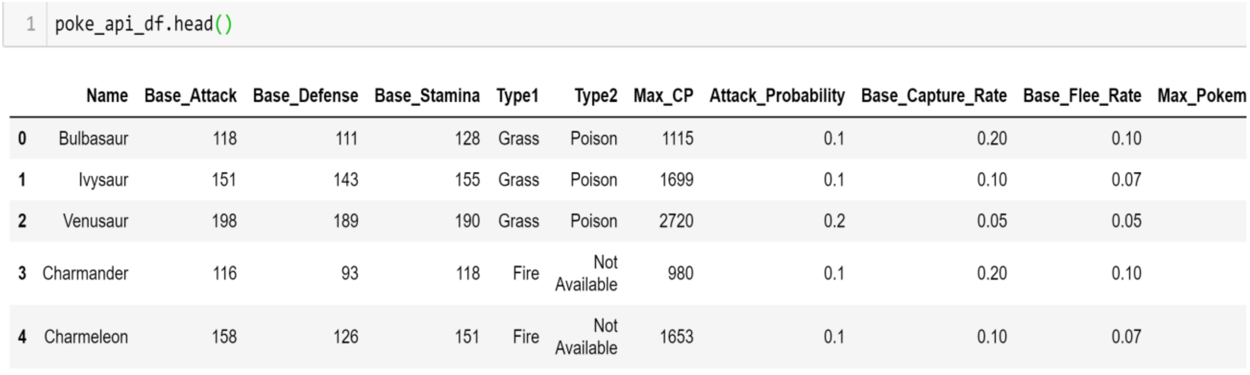
Data gathered from web scraping process.



```
poke_df.head()
```

	Name	Type1	Type2	Total	HP	Attack	Defence	Sp.Attack	Sp.Defence	Speed
0	Bulbasaur	Poison	Grass	318	45	49	49	65	65	45
1	Ivysaur	Poison	Grass	405	60	62	63	80	80	60
2	Venusaur	Poison	Grass	525	80	82	83	100	100	80
3	Venusaur	Poison	Grass	625	80	100	123	122	120	80
4	Charmander	Fire	Not Available	309	39	52	43	60	50	65

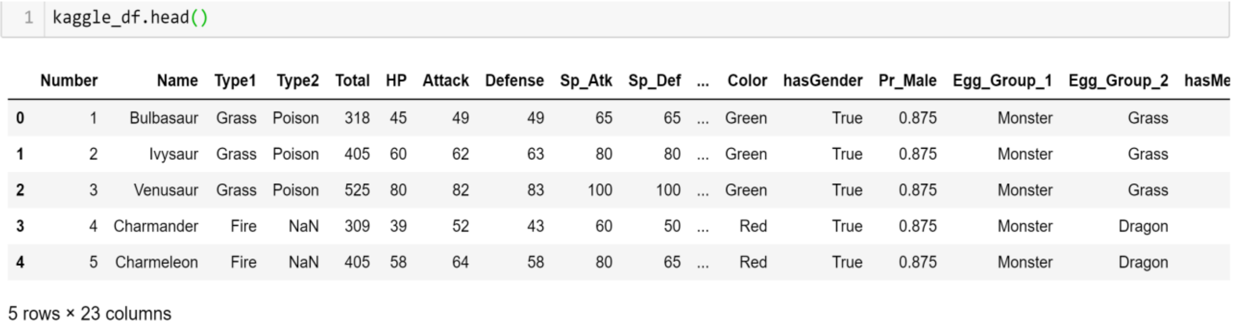
Data gathered from Web API.



```
poke_api_df.head()
```

	Name	Base_Attack	Base_Defense	Base_Stamina	Type1	Type2	Max_CP	Attack_Probability	Base_Capture_Rate	Base_Flee_Rate	Max_Pokem
0	Bulbasaur	118	111	128	Grass	Poison	1115	0.1	0.20	0.10	
1	Ivysaur	151	143	155	Grass	Poison	1699	0.1	0.10	0.07	
2	Venusaur	198	189	190	Grass	Poison	2720	0.2	0.05	0.05	
3	Charmander	116	93	118	Fire	Not Available	980	0.1	0.20	0.10	
4	Charmeleon	158	126	151	Fire	Not Available	1653	0.1	0.10	0.07	

Data gathered from CSV format dataset.



```
kaggle_df.head()
```

	Number	Name	Type1	Type2	Total	HP	Attack	Defense	Sp_Atk	Sp_Def	...	Color	hasGender	Pr_Male	Egg_Group_1	Egg_Group_2	hasMe
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	...	Green	True	0.875	Monster	Grass	
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	...	Green	True	0.875	Monster	Grass	
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	...	Green	True	0.875	Monster	Grass	
3	4	Charmander	Fire	NaN	309	39	52	43	60	50	...	Red	True	0.875	Monster	Dragon	
4	5	Charmeleon	Fire	NaN	405	58	64	58	80	65	...	Red	True	0.875	Monster	Dragon	

5 rows × 23 columns

Contributions

Purvang worked on fetching data through Web API and from the Kaggle Dataset.

Megha worked on fetching data from the Web Source.

Purvang carried out the Data Auditing.

Megha designed and developed the Conceptual Database Model.

Both the authors discussed the results and contributed to the final report.

References

[1] Prof.Nik Bear Brown https://github.com/nikbearbrown/INFO_6210

[2] Web Scraping: https://pokemondb.net/pokedex/all

[3] Web API: https://pogoapi.net/documentation/

[4] Raw CSV file taken from Kaggle: https://www.kaggle.com/alopez247/pokemon/home

[5] Python Tutorials: https://www.tutorialspoint.com/python/

[7] https://www.webharvy.com/articles/what-is-web-scraping.html

[8] https://en.wikipedia.org/wiki/Web_scraping

[9] https://github.com/shadforth/pokemon-web-scraper/blob/master/src/scraper.py

[10] https://towardsdatascience.com/web-scraping-html-tables-with-python-c9baba21059

[11] https://en.wikipedia.org/wiki/Web_API

[12] https://schoolofdata.org/2013/11/18/web-apis-for-non-programmers/

[13] https://www.e-education.psu.edu/geog485/node/141

[14] https://en.wikipedia.org/wiki/Conceptual_schema

[15] https://en.wikipedia.org/wiki/Conceptual_schema

[16] https://www.quora.com/What-is-Kaggle-and-how-exactly-should-I-use-it

License