ETL Project - Week 12

An ETL Demonstration using Pokémon

Paul Bernert and Vincent Heningburg

May 16, 2020

ABSTRACT

This report aims to demonstrate the benefits of following the Extract, Transform and Load (ETL) processes of data warehousing through a basic example using information related to Pokémon and their moves. Using an external data-source from Kaggle, we explain each of the three steps to ETL and the logic/justifications for actions within each of those steps.

I. INTRODUCTION

ETL is the process of copying data from one or more sources into a destination system which represents the data differently from the original source source or in a different context than the original source. It does this through Extraction, Transformation, and Loading.

The first part of an ETL process involves extracting the data from the source system. In the data transformation stage, a series of rules or functions are applied to the extracted data in order to prepare it for loading into the end target. The load phase loads the data into the end target, which can be any data store including a simple delimited flat file or a data warehouse.

ETL is beneficial for several reasons. First and foremost, ETL is designed to provide an easy-to-understand flow of the analytical logic, with three easy to understand steps. Data warehousing is also designed to increase speed of data retrieval and analysis. This project aims to give a brief demonstration of the ETL process and demonstrate through example why it is a valuable exercise for aspiring data analysts.

II. EXTRACT

The data-sets used in this ETL Project come from Kaggle–an website designed for data scientists to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists, and more. A link to the Pokémon data-sets can be found here.

The two data-sets provided were titled *pokemon-data.csv* (hereon referenced as *Pokémon Dataset*) and *move-data.csv* (hereon referenced as *Moves Dataset*).

The Pokémon Dataset includes information pertaining to each Pokémon, such as: (i) their combat stats; (ii) what type of Pokémon they are; and (iii) their rank on the Pokémon ”tier-list”[1]. The data-set also includes a column containing a comprehensive list of every move that Pokémon is capable of learning.

The Moves Dataset provides data related to every move Pokémon can do, including information on: (i) What type of Pokémon are able to complete the move; (ii) the much the move ”costs” to do; and (iii) a complete breakdown of all stats related to that move.

[1] A Pokémon's rank on the tier list is used to determine the general consensus on a Pokémon's performance in competitive matches

The Pokémon Dataset and the Moves Dataset were provided in the *Comma-Separated Value* (or CSV) format. While the Moves Dataset was a conventional CSV file, the Pokémon Dataset stored their columns as **semi-colon separated values**. This was because some of the columns have their contents stored as lists (such as the list of moves), and the information in the lists were separated by commas. As a result, the author of the data-set decided to separate columns with semi-colons.

III. TRANSFORM

Using the *Pandas* library, we were able to import the two data-sets as DataFrames. The list entries in these DataFrames were stored as strings. The first necessary transformation was to parse these strings and turn them back into lists.

The next step was to pull out the columns that weren't relevant for this particular project, such as the next generation of the Pokémon and their Abilities. Extracting these irrelevant columns brought the total number of columns from twelve to eight. However, some of the columns still hadn't been separated out properly, such as a Pokémon's 'Type' column sometimes only having a Type 1 and sometimes having both a Type 1 and Type 2. Splitting up the columns that were still containing lists expanded our number of columns from eight back up to eleven.

There was one more column containing way too much information: the 'Move' column. Because some Pokémon are able to learn well over 100 different moves, we decided the best transformation to do for this category is to randomly select one move from the list as the representative move for that Pokémon.

The original column names were difficult to work with, often containing spaces and other difficult characters. As a result, a small transformation that needed to be done was to re-name several of these difficult-to-operate column names into something simpler (such as adding an under-score between 'Special' and 'Attack' in the column 'Special Attack').

The Moves Dataset required several transformations that were very similar to those in the Pokémon Dataset. The first step was to remove the columns that weren't relevant (bringing the column count from nine down to six). The second step was to do some column-renaming here as well, such as changing 'Name' to 'Move', or 'Type' to 'Move_Type'.

With all transformations complete on their respective DataFrames, it was time for the final transformation–merging the two DataFrames on the 'Move' column. After completing these transformations, there is

IV. LOAD

The final combined DataFrame was loaded into a Postgres database with a table consisting of the following information: 'Name', 'HP', 'Attack', 'Defense', 'Special_Attack', 'Special_Defense', 'Speed', 'Type_1', 'Type_2', 'Move', 'Move_Type', 'Category', 'PP', 'Power', 'Accuracy'.