Big Data Project

This documentation should be supported by a presentation of several slides to summarise the main points. Please refer to the provided presentation example here:

https://pitch.com/public/ff26389f-278f-4226-bac7-4d323eb1ad31

"If you don't reveal some insights soon, I'm going to be forced to slice, dice, and drill!"

Introduction

We have 3 datasets:

- **facebook_dataset.csv**
- **google_dataset.csv**
- **website_dataset.csv**

Those datasets contain basic information about companies (eg. Domain, company name, phone number, address, category, etc) from different sources. We want to create a 4th dataset that joins the information in all of those 3 datasets with a high accuracy.

Terminology & details

- language of choice - python
- facebook_df - referring to the facebook dataset (**facebook_dataset.csv**)
- google_df - referring to the google dataset (**google_dataset.csv**)
- website_df - referring to the sites dataset (**website_dataset.csv**)
- merged_df - facebook and google and website

Data Cleaning & Preprocessing

- convert categories, address, domain,city, country & regions to lowercase

  eg. "England" → "england'

EDA

Reading the data

The first thing that we have to do is read the CSV files. Unfortunately, pandas doesn't seem to be able to read the facebook and google datasets as there are commas (",") inside columns that contain text data and those commas trigger the creation of a new column, which breaks the CSV format. We can easily fix this by concatenating all columns that are generated by those commas in one single column, and then read the CSV data in pandas.

Initial Plots

Firstly, in order to read the csv file I have used the csv.reader function to parse a single line of a csv file.
* Where the a string represents a line in a csv file.
* The character separates values in the csv file. It defaults to a comma (,)
* The output would be a list of values obtained by splitting the input line using the specified delimiter.
*The function is named as split_csv_line.

In the second stage for read the csv files,
* A function is used to read a csv file and returns its contents in pandas data frames and lets discuss in detail:

For the function read_csv the input is the filepath and the delimiter.
* In this, it read the lines of the give path file.
* I have initialized an empty list called data to store the data rows in it.

* It used the split_csv_line function to parse the first line to get the headers and it itereates over the rest of the line excluding the header split each line into fields using the split_csv_line.
* If a row has more fields than headers which could happen if a fields value contains the delimiter more times, then it concatenates to the adjacent fields until the number of fields matches the number of headers or drop the row if it can't be corrected.
* Then it adds the list to the fields to data.
* Then it creates a pandas dataframe from data list using the headers as columns names.
* Returns a Dataframe.

We can discover whether or not the data is organized on rows by company name, or by other columns.

It does look a lot better, now it seems like all the fields are organised and read by pandas dataframes which can be used for more analysis.

We can also find a small problem with the Facebook and google datasets. There are some duplicate company names in those datasets. But the address belonging to the domain name or company name are different which is to be considered. That, where it has a possibility the a single company has a different branches or located in different palaces.

Perfect! Now all datasets are organized by company name, and the other columns seem to show a friendly pattern. It seems like some features such as region_name are categorical features as they do repeat a lot. This is great because we can now get more insight into how the data is distributed in those datasets.

Merging Strategy

I will merge the datasets based on different approaches. In the end, there will be a data frame full of rows that represent companies. A row from this data frame will be a combination of information from the other datasets.

I tried joining all the information about a company together in one single row. This has been done in a combination of fuzzy name linking, and domain, category, country & region matching. But I have selected mainly the fields named domain to merge the three data frames called the facebook_df, google_df and website_df to create a new merged_df based the join condition.

In this process, if the same column is present in the different data frames, I have provide the suffixes to be the organised accordingly and Infer to it. So, it can be identified easily from which data frames it is exactly belongs to.

Resolving Columns Conflicts

The main purpose of this to solve the conflicts and discrepancies in column values between different sources or data frames.

I. t is defined as function as resolve column conflict data.It mainly indicates to take a data frame, names of two conflicting columns and an optional third column for comparison and the name of the new column to store the resolved data in the new column.

After that, at the end the columns are arranged in a specific order to that merged dataframe is organised accordingly as per the requirement.

Also, writing the dataframe to a csv file. By this we can save the merged and processed dataframe to a csv file.

Results

The decisions I have made in order to join the 3 datasets are:

1. What column will you use to join?

Answer: I selected the **'domain'** column as the primary joining key. This choice is grounded in the notion that a domain is typically unique to a company.

Assuming all three datasets from Facebook, Google, and the Company Website consistently use the same domains for corresponding companies, the 'domain' column seems to be an apt primary key for merging.

2. If you have data conflicts once you join, which one do you believe?

Answer: To address data conflicts, it's essential to rank the data sources based on their perceived reliability. In this situation:

Company Website: Given that companies directly oversee their websites, the data from this source is presumed to be the most accurate and current, making it the top priority.

Google: Ranking second in reliability, Google's vast data scraping and validation systems offer a trustworthy data set.

Facebook: This platform is last in the trust hierarchy. The rationale is that its data might derive from user-generated content or business profiles that aren't updated as frequently.

I would use the following priority order to pick the first available non-null value:

Company Website > Google > Facebook.

This sequence is premised on the belief that data from a company's website is more dependable than data from third-party platforms such as Google or Facebook.

3. If you have very similar data, what information will you keep?

Answer: When confronting similar data points, the objective is to conserve the most precise and detailed information:

If the same data is presented in two datasets, but one offers a more comprehensive version, the richer dataset should be favored.

For minor discrepancies in similar data (like phone numbers or addresses with different formatting), it's essential to standardize this information to a unified format, retaining the more detailed variant.

If two datasets present entirely disparate data for a specific column (like two distinct phone numbers attributed to one company), both entries should be kept for subsequent manual review unless a clear priority hierarchy exists.

Data Cleaning: The data was further refined using the read_csv and lowercase_and_strip functions for better consistency and to handle potential discrepancies.

Performance: Although I've employed Pandas for this solution, using Apache Spark could offer swifter processing for larger datasets. Performance enhancements can further be explored for improved efficiency.

Thank you !!