Lab Report

186.868 Visual Data Science 2024W

11912007 - Yahya Jabary

Code: `github.com/sueszli/uwaterloos-sunshines`

Contents

This project adheres to the traditional stages of a data science pipeline: Discover, Wrangle, Profile, Model, and Report. Each stage involves specific tasks and requirements, which are detailed in the subsequent chapters. The primary objective is to achieve a comprehensive understanding of the data and effectively communicate the insights through an interactive dashboard to a wider audience.

Significant emphasis is placed on the quality of the code and the reproducibility of the results. The `makefile` located in the root directory of the project repository includes commands for generating a `pip-compile`'d requirements file, ensuring an isolated virtual environment. Additionally, short scripts are available for execution within Docker or Conda environments.

Discover Stage

The tasks of this section are to:

- Discover your topic
  - Either select one of the suggested topics, or come up with your own idea. → Our custom topic was confirmed by the lecturer via email.
  - The topic should be possible to easily understand the data context, even if you are not an expert.
- Describe the two datasets you selected
  - Use at least two independent datasets.
  - The selected datasets must be multidimensional: they have to contain more than 5 variables.
  - The selected datasets must contain a sufficient amount of data rows.
- Keep the length to 1-2 A4 pages

This stage is dedicated to defining the research question, motivation, and context of the study. It also includes the selection of datasets and the methodology for combining them to address the research question. The primary goal is to provide a clear and concise overview of the project's scope and objectives.

Motivation and Context Salary transparency is a complex and often contentious issue in the workplace. While it can promote equity and reduce discrimination, it may also foster jealousy and resentment among employees. The approach to salary disclosure varies significantly between European countries and North America, particularly in the context of public institutions and academia.

In European countries, it is common practice to disclose compensation brackets for public officials, including employees at public universities. However, these disclosures typically do not reveal exact salaries or identifiable information, limiting the ability to study personal income inequality in detail.

North America takes a different approach, with many states and provinces enacting laws that mandate public institutions to disclose the salaries of top-earning employees. These individuals, often referred to as "Sunshines," are those earning more than $100,000 annually in salary, excluding additional benefits. The published lists, known as "Sunshine Lists," include the names of these employees, enabling more in-depth studies of personal income inequality.

The academic setting provides a particularly interesting context for studying income correlation. Academics typically publish their work openly and have established performance metrics, such as the total number of publications, citations, and h-index. This transparency in academic output creates a unique opportunity to examine the relationship between income and academic performance.

Given our personal connection to the University of Waterloo, we have chosen to focus their study on this institution's sunshine list. This decision provides a familiar and accessible dataset for analysis.

Datasets and Methodology The primary dataset for this study consists of the publicly available sunshine lists from the University of Waterloo. This dataset provides both tabular and time-series data, as it includes annual checkpoints. To enrich this information, we plan to join it with data from `csrankings.org`, which contains scholar IDs. These IDs can then be used to query additional resources such as Google Scholar and the Semantic Scholar API, providing more detailed information about the research output of individual employees.

By combining these datasets and API resources, the study aims to investigate the correlation between compensation and academic performance at the University of Waterloo. While specific research questions are yet to be formulated due to the exploratory nature of the approach, potential areas of inquiry include: (1) The relationship between changes in compensation and role over time, (2) Correlation between academic performance metrics and salary and (3) Potential gender-based salary disparities, inferred through natural language processing techniques applied to employee names.

We have selected the following data sources to support our study:

- University of Waterloo Salary disclosure for 2020-2023
- CS Rankings CSV based on the `csrankings.org` Github repository
- Google Scholar API
- Semantic Scholar API

Ethical Considerations and Data Handling While web scraping is legal in the European Union and the data used in this study is publicly available, we have committed ourselves to handling the information ethically. To avoid potential conflicts of interest and protect individual privacy, we will not visualize any identifiable information about specific employees or derive any conclusions that could harm individuals.

The focus of the study will be on detecting general trends and patterns rather than singling out individuals. All published data and insights will be aggregated to maintain anonymity. This approach ensures that the research can proceed without compromising ethical standards or potentially harming individuals, even in cases where significant discrepancies between pay and performance might be discovered. By adhering to these ethical guidelines, we aim to contribute valuable insights into the relationship between academic performance and compensation while respecting the privacy and dignity of the individuals whose data forms the basis of the study.

Wrangle Stage

The tasks of this section are to:

- Join the two or more datasets you selected into one big data table
  - How did you join the datasets? Which keys did you use to join the data? Did all keys match? Did you have to introduce new keys?
- Solve issues like formatting issues, missing data, faulty values, and non-matching keys.
  - Which data cleaning steps have been necessary? Did you experience data issues, and if so, which ones? How did you solve them? Did you use automated methods? Did you use visualization to inspect data issues?
- Visually show and explain the data quality of your dataset (for example, before and after cleaning steps). Come up with your own, creative, solution here.
- Use a charting library, not fully-featured applications.
- Keep the length to $\frac{3}{4}$ to 1 A4 page.

This stage is dedicated to data wrangling and information retreival which involves combining multiple datasets, cleaning the data, and ensuring its quality. The primary goal is to prepare the data for further analysis and visualization, addressing any issues that may arise during the process. It is the foundation for the subsequent stages of the data science pipeline in which insights are derived and models are built.

Data Retrieval and Integration Initially, we scraped the University of Waterloo's sunshine list for the years 2020 to 2023. Using BeautifulSoup, we converted HTML tables into CSV format, ensuring that unnecessary nested span tags were removed for cleaner data. We validated the CSV schema to maintain consistency across files and merged these into a single dataset, partially formatted as timeseries data in JSONL format. String cleaning and validation using CSVLint were crucial steps in preparing this dataset for further integration.

Our next step involved downloading the CSRankings data, which we aimed to merge with the sunshine list. The CSRankings dataset, sourced from a GitHub repository, included fields such as name, affiliation, homepage, and scholar ID. Given that this dataset is updated quarterly and allows scholars to update their affiliations via pull requests, it was vital to ensure accuracy in matching. We employed fuzzy matching techniques to align names between datasets, setting a threshold of 0.8 to avoid duplicates. Although this method resulted in relatively few matches – 107 out of 149 University of Waterloo entries in CSRankings matched with the sunshine list – it was an effective approach given the complexity of name variations.

The fuzzy matching process was particularly interesting as it required balancing precision and recall. By using a threshold of 0.8, we minimized false positives while still capturing relevant matches. This technique proved essential in dealing with variations in name spellings and formats across datasets.

Our attempts to integrate Google Scholar data were unsuccessful due to IP blocking issues and the lack of a dedicated API for information retrieval. This setback highlighted the challenges associated with accessing certain online resources without incurring significant costs through proxy services.

We then turned to Semantic Scholar for additional data enrichment. By leveraging its API and employing VPNs to manage IP switching, we were able to search for researchers using fuzzy name matching with a similar threshold of 0.8. Although some fields like affiliations and homepage were consistently empty – limiting their utility – we focused on extracting valuable metrics such as paper count, citation count, and h-index. These metrics provided insights into the academic impact of researchers who could be matched with the sunshine list.

Despite these efforts, there was notable data loss; only 68% of employees from the sunshine list could be joined with Semantic Scholar data. This was partly due to some employees not being researchers or not having sufficient presence in academic databases.

It's also worth mentioning that we have no certainty in whether the retrieved performance metrics from the API just based on the name matching are correct as these are not unique identifiers. This could lead to potential errors in the analysis, which we will need to consider in the subsequent stages and interpretations – but we can with certainty say that this is the best heuristic we could come up with given the publicly available data.

Data Preprocessing and Quality Assurance In our preprocessing phase, we combined data from all sources into a unified dataset joining: Sunshines List × CSRankings × Semantic Scholar API. To enhance query performance, we converted JSONL files into a CSV format by using a self-implemented version of R's `pivot_wider` function. We then dropped unnecessary fields to reduce data redundancy, inferred the employee's sex based on their name by using a DistilBERT model for text classification with a test set accuracy of 1 and clustered the 500+ roles into 25 clusters using sentence embeddings from the HuggingFace library and k-means clustering.The details on the machine learning algorithms used in the preprocessing stage are described in the relevant subsequent sections.

To ensure data quality and consistency, we maintained a detailed log of all data cleaning and integration steps, enabling reproducibility and transparency in our approach and validated the final (`v4`) dataset using CSVLint in every step. Additionally we encoded all substrings in UTF-8 to ensure compatibility with downstream tools and libraries and dropped all empty rows and columns. In the case of missing data, we opted to retain all records and adding `null` values rather than dropping them, as they could still provide valuable insights into the dataset's structure and potential biases. We didn't drop or impute any data other than rows with missing matches in the inner joins.

The following code snippet and chart illustrate the distribution of the dataset before and after the cleaning and integration steps.

```
$ find ./* -type f -exec wc -l {} +
   29361 ./data/csrankings.csv
    2514 ./data/sunshines-v1.jsonl
    2514 ./data/sunshines-v2.jsonl
    1709 ./data/sunshines-v3.jsonl
    1709 ./data/sunshines-v4.csv
    1762 ./data/sunshines2020.csv
    1857 ./data/sunshines2021.csv
    1904 ./data/sunshines2022.csv
    2140 ./data/sunshines2023.csv
    ...
```

As shown in Figure 1, the initial merge of the 4 sunshine lists (each 2140, 1904, 1762, 1857 rows) resulted in a dataset with 2514 rows. By joining with CSRankings, we simply extended the dataset with features but didn't drop any rows. After fuzzy joining the dataset with the Semantic Scholar API for the final dataset however we lost 806 rows (= 2514 - 1708) or 32% of the data due to query misses.

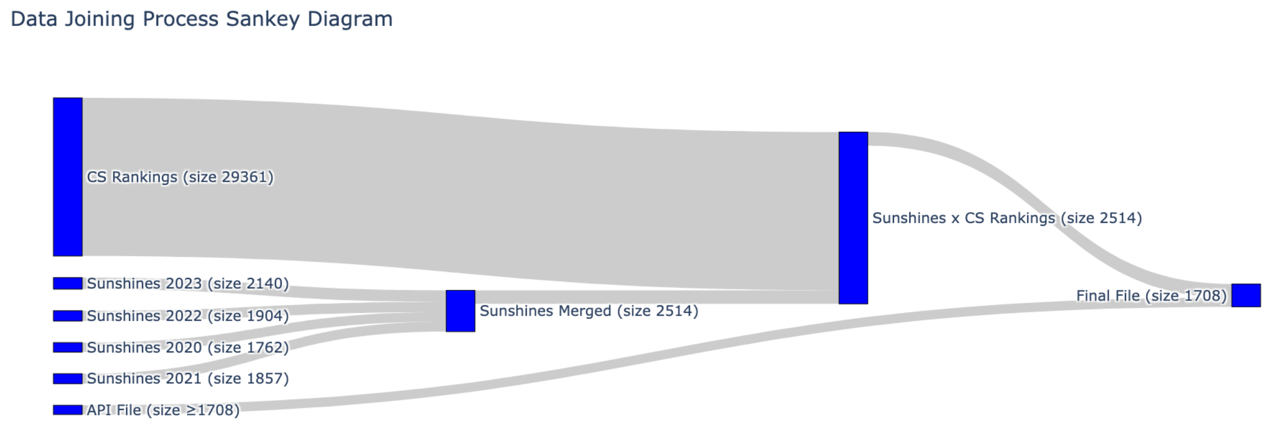
Data Joining Process Sankey Diagram
CS Rankings (size 29361)
Sunshines x CS Rankings (size 2514)
Sunshines 2023 (size 2140)
Sunshines 2022 (size 1904)
Sunshines 2020 (size 1762)
Sunshines 2021 (size 1857)
API File (size ≥1708)
Sunshines Merged (size 2514)
Final File (size 1708)

Figure 1: Sankey Diagram of file sizes after multiple merge and inner join operations.

These efforts in addition to some in-memory preprocessing before the profiling stage resulted in the following dataset schema:

```
name                    object
sex                     object
paper_count              int64
citation_count           int64
h_index                  int64
role_2020               object
role_cluster_2020      float64
salary_2020            float64
benefits_2020          float64
```

```
role_2021               object
role_cluster_2021      float64
salary_2021            float64
benefits_2021          float64
role_2022               object
role_cluster_2022      float64
salary_2022            float64
benefits_2022          float64
role_2023               object
role_cluster_2023      float64
salary_2023            float64
benefits_2023          float64
latest_totalcomp       float64
latest_role             object
latest_role_cluster    float64
perf_combined            int64
totalcomp_2020         float64
totalcomp_2021         float64
totalcomp_2022         float64
totalcomp_2023         float64
```

Where:

- `name`: the name of the employee
- `sex`: inferred based on the name using a text-classifier
- `paper_count`, `citation_count`, `h_index`: metrics retrieved from the Semantic Scholar API as of October 2024
- `role_{YYYY}`, `role_cluster_{YYYY}`: role and role cluster of the employee in the respective year
- `salary_{YYYY}`, `benefits_{YYYY}`, `totalcomp_{YYYY}`: salary, benefits, and total compensation (consisting of salary and benefits) of the employee in the respective year
- `latest_totalcomp`, `latest_role`, `latest_role_cluster`: total compensation, role, and role cluster of the employee in the latest year available

Additionally, we computed Δ values per year for all the numerical attributes to facilitate time series analysis in the subsequent stages, starting from 2021 as the base year given that we had no predecessor data for 2020 to compute the changes with.

Profile Stage

In this section we explore the data in detail, to completely understand its structure, and to discover any interesting patterns that can be found in there.

The tasks of this section are to:

- Find at least 3 informative insights in your dataset. For each one add a short text describing the insights plus one visualization.
- Keep the length to $\frac{3}{4}$ to 1 A4 page per insight.

2

3

4

5

6

Model Stage

The tasks of this section are to:

- Build a model of the data you studied, to find answers to the questions you selected (possible models: linear regression, clustering, pca, anomaly detection, etc.)
- describe the modeling process
- Create one or more visualization(s) that describe the results of your model
  - How would you increase trust of your customers/colleagues in your modeling approach by using data visualization?
- Keep the length to 1-2 A4 pages.

7

8

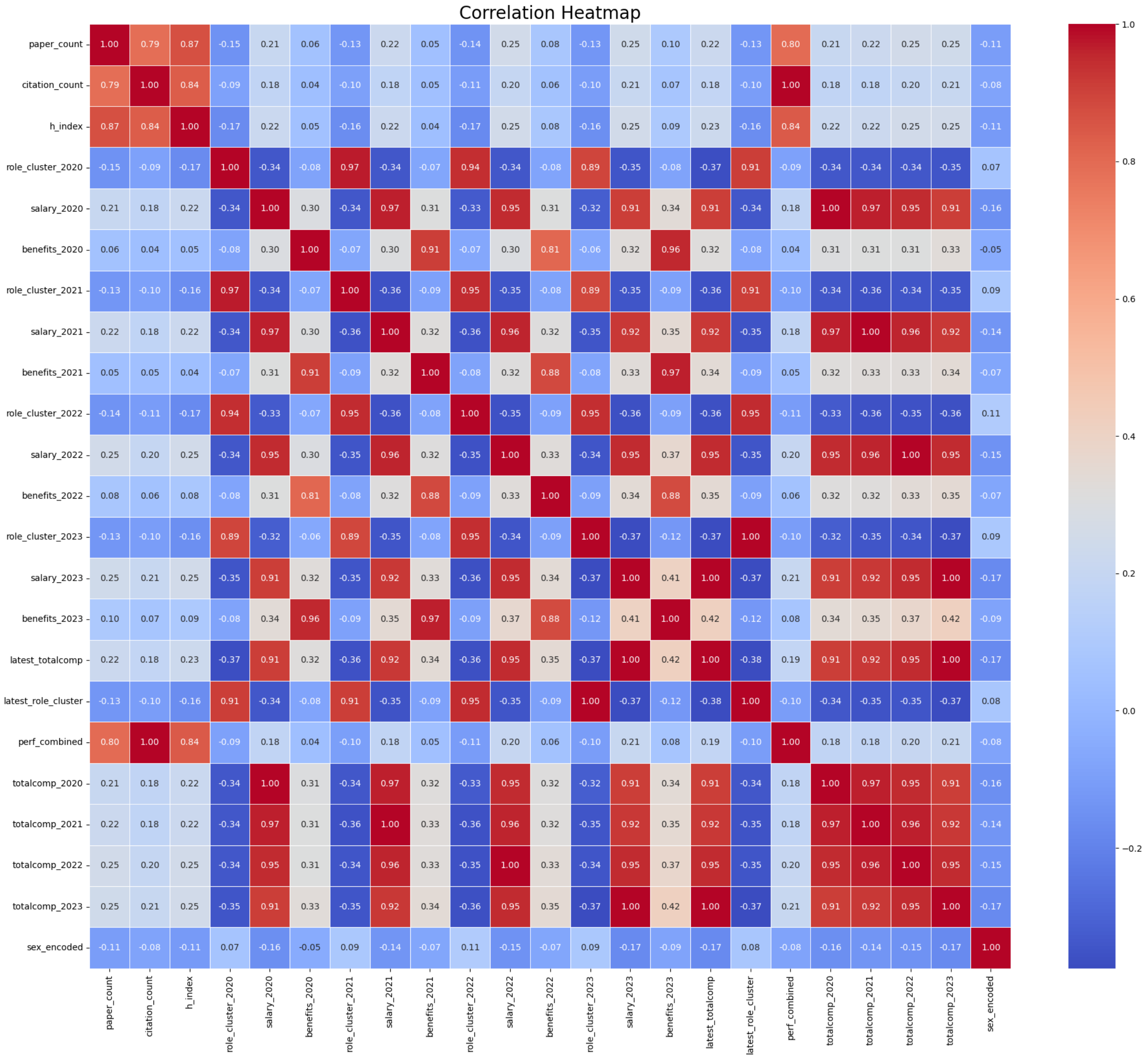

Figure 2: Heatmap of Correlation Matrix.

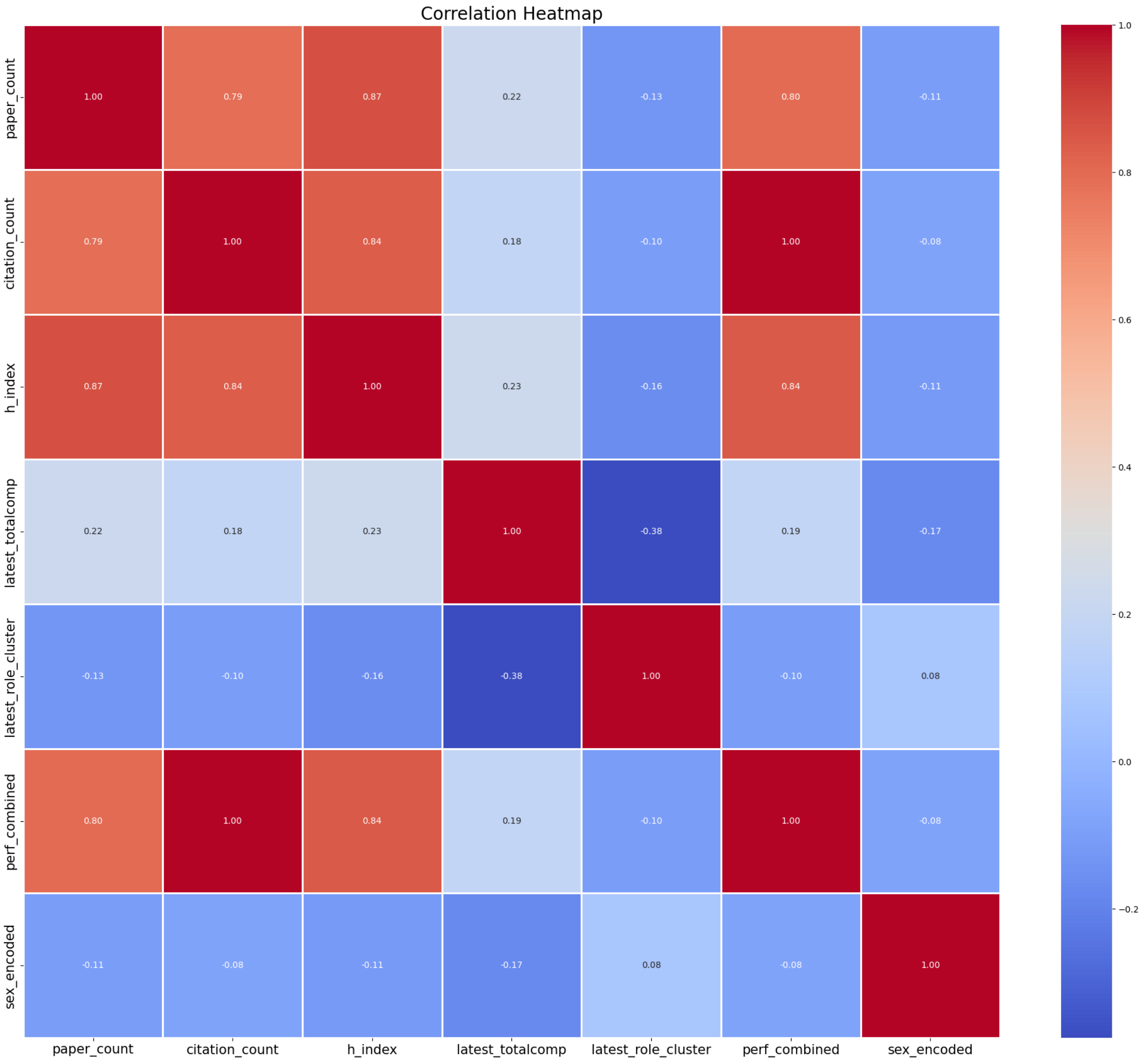

Figure 3: Heatmap of Correlation Matrix.

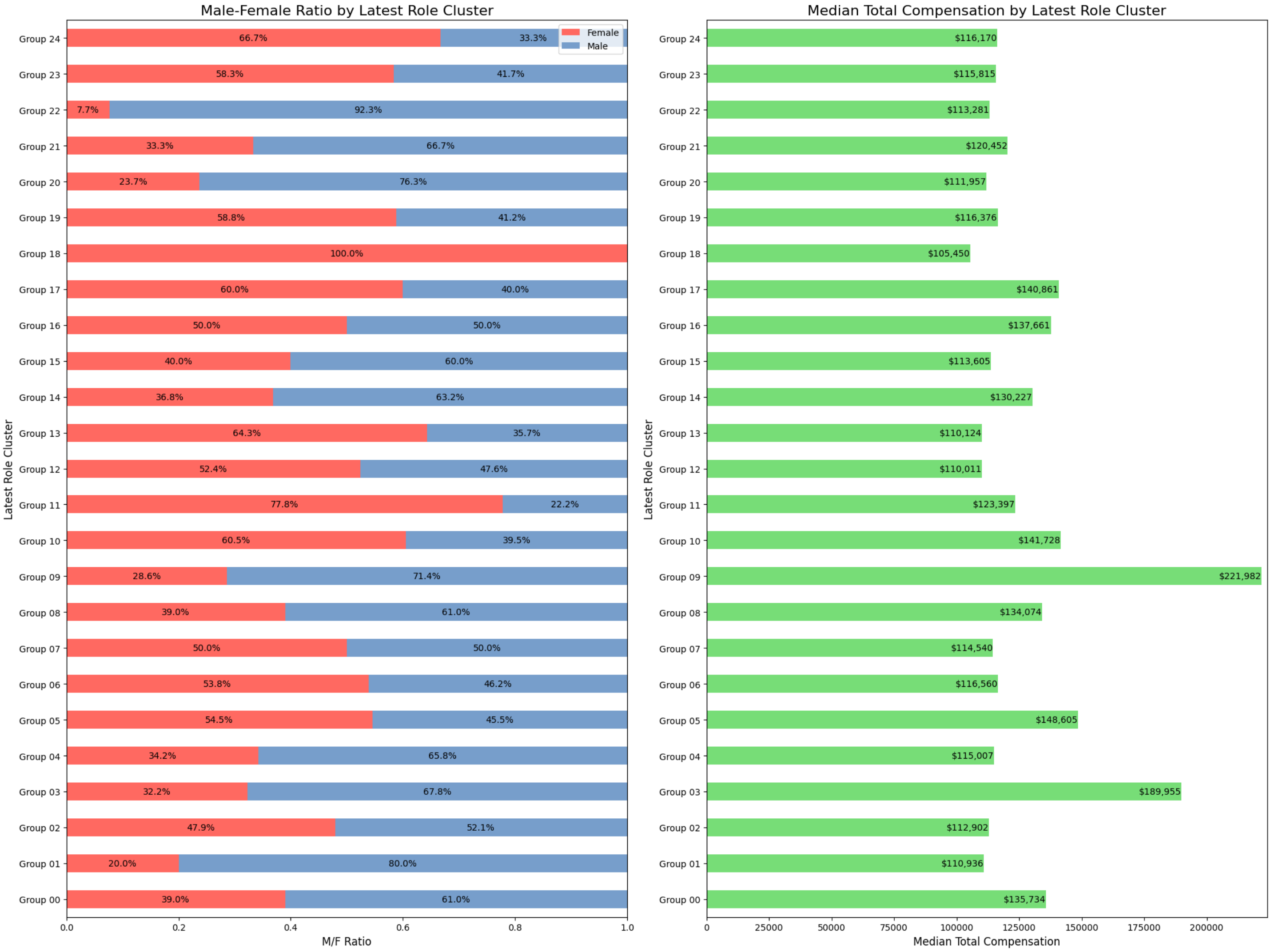

Figure 4: MF Totalcomp Ratio.

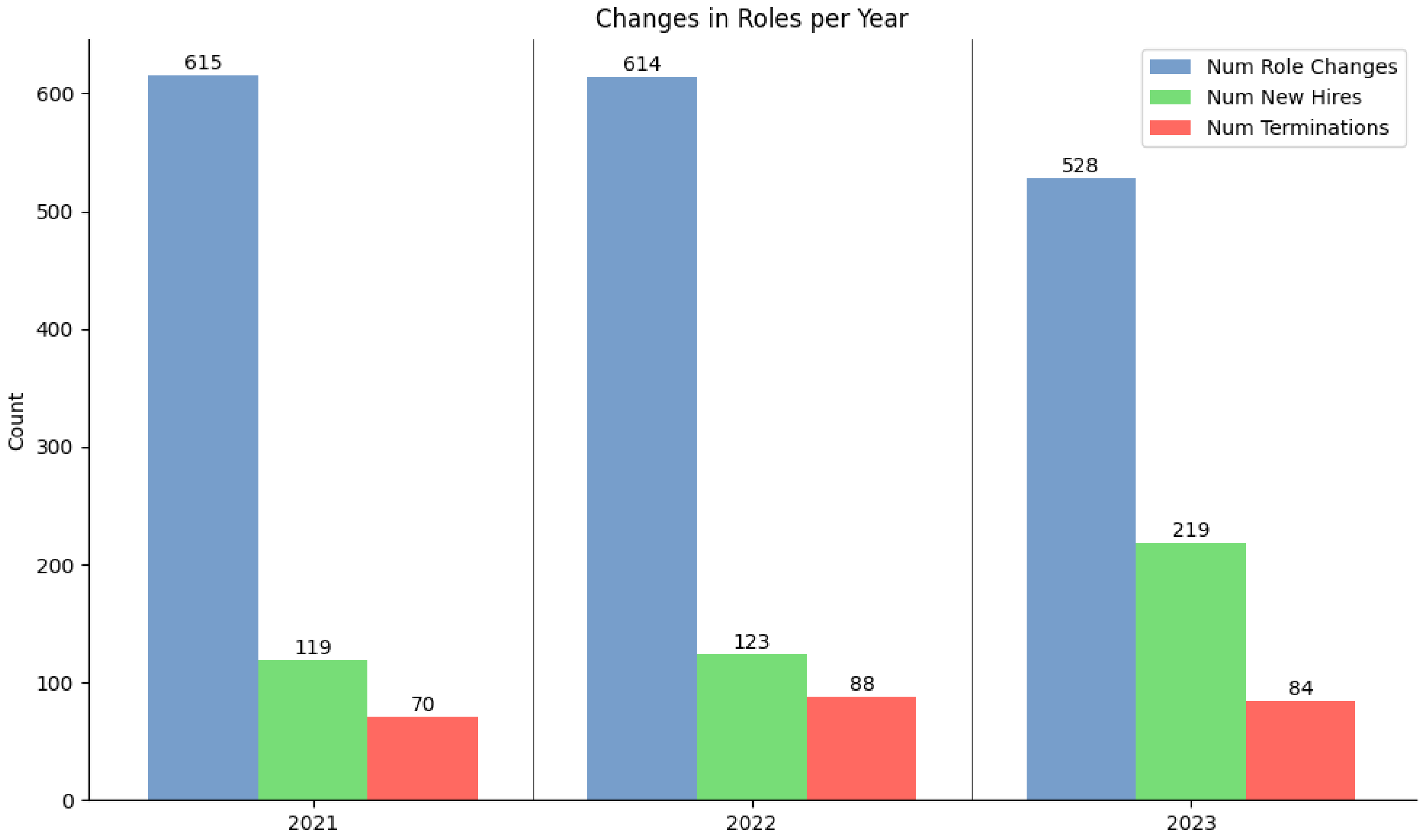

Figure 5: Timeseries.

Report Stage

The tasks of this section are to:

- Show your findings in an interactive dashboard to a broader audience. The findings may be related to the Model/Wrangle/Profile stages.
  - Use appropriate charts, visual encodings (e.g., color).
  - Use at least four different types of visualizations / charts.
  - Include interaction (e.g., filters, zoom, not a jupyter notebook), brushing & linking (changes in one view affect others, but not global filters)
- Use a library, not fully-featured applications.
- See examples: https://tuwel.tuwien.ac.at/mod/page/view.php?id=2433356

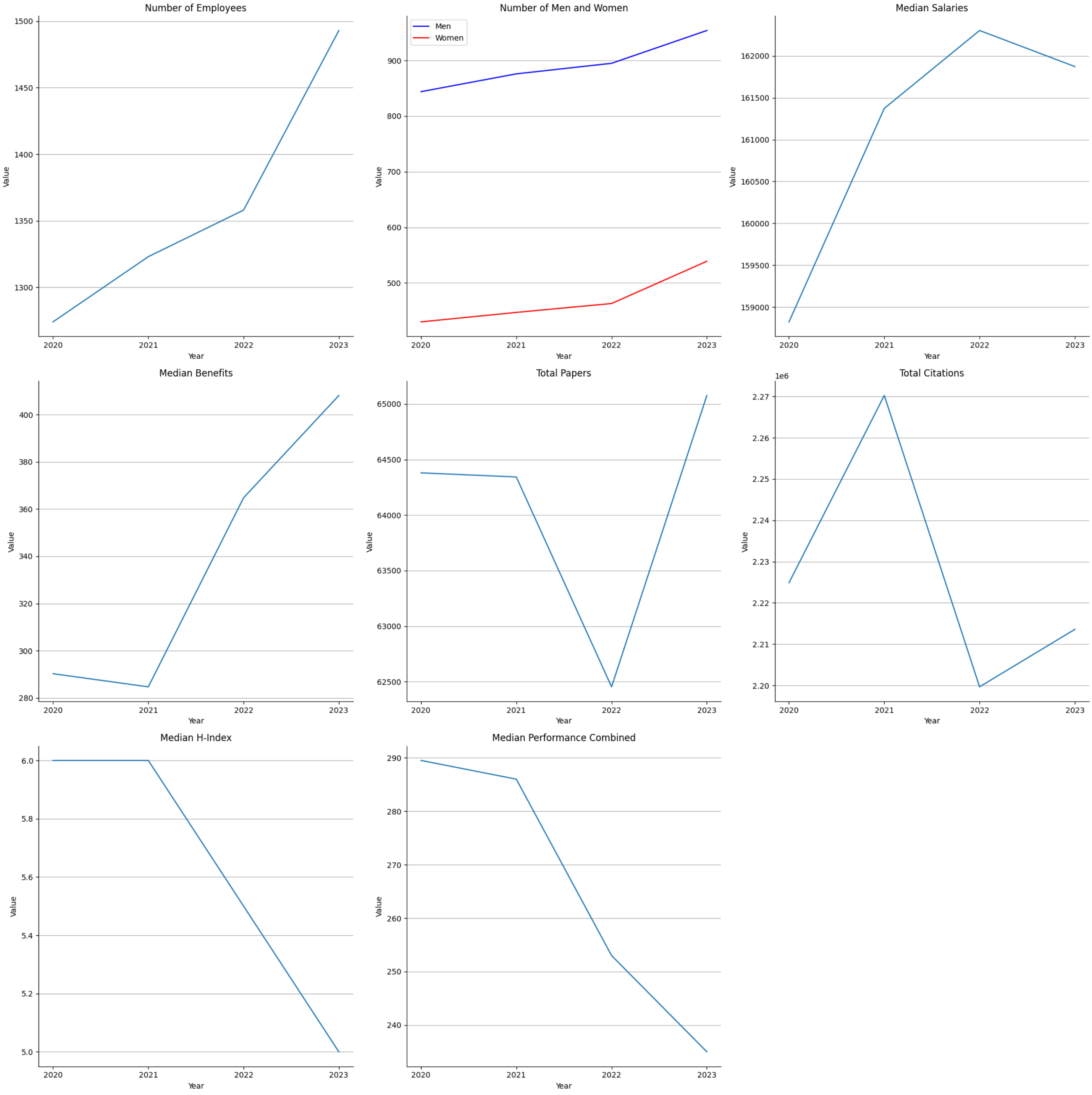

Figure 6: Timeseries.

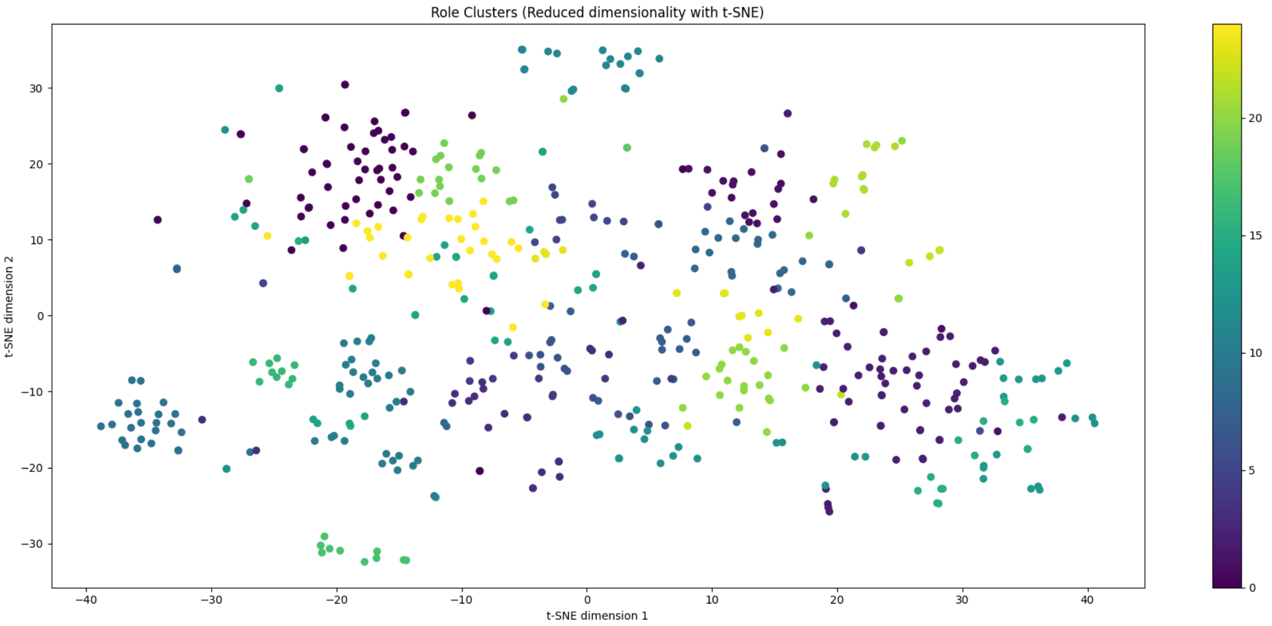

Figure 7: Latent Representation Clustering of Roles.

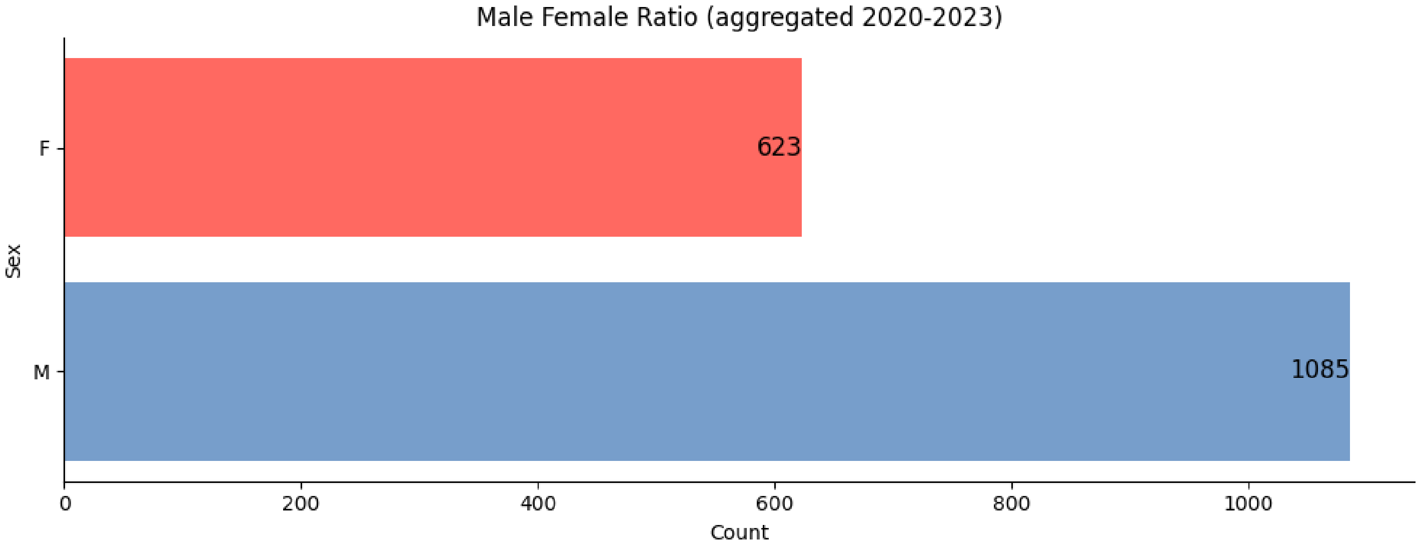

Figure 8: Sex inference based on name via Text Classification.