# Introduction

It is common knowledge that there is gender disparity in the field of computer science. While women played an impactful role in the early days of computing, there has been a steady decline in female representation in computer science since the 1970’s.

Computer science is a broad area of study with many subfields. Since women and men often choose different career paths, do they also choose different areas to study within computer science? How many papers do women publish in the field? How many others do they tend to collaborate with and what is their average position in the author list? This project analyses the fields of study and publishing trends of women in computer science.

# Materials and Methods

## Data Collection and Management

The source data about the authors was collected from dblp.org. The dblp is a database that provides bibliographic information from major computer science journals and conferences. The data was obtained by both downloading the entire database as a raw XML and by using their API.

The data collected from the XML using the python program dblp\_parser.py. The dblp\_parser.py program parsed the dblp XML file incrementally with the etree library and saved it incrementally as a JSON file with the tinyDB library.

The data outputted from the dblp\_parser was stored as a JSON file named db.json with the following fields:

|  |  |  |
| --- | --- | --- |
| Name | Datatype | Description |
| name | string | A unique author name from the dblp |
| titles | string | A string of publication titles by the author, delimited by a semi-colon |

Because the xml file was so large (5.3 million publications from 2.6 million authors), only a sample from the dblp was collected.

The db.json file was then processed by the dblp\_genderize.py program. The program iterated through each entry and then, using Genderize.io's API, predicted the author's gender using their first name. Because there was a limit of 1000 queries a day from the Genderize.io and just first names were used, any name queried was saved into the JSON file names.json with the following fields:

|  |  |  |
| --- | --- | --- |
| Name | Datatype | Description |
| name | String | The first name supplied for gender prediction |
| gender | String | The predicted gender |
| probability | Number | Certainty of predicted gender |
| count | Number | Rows examined by Genderize.io to calculate the response |

In subsequent iterations of the program, if the first name had already been processed, the program would draw from the local file, rather than use the API therefor saving a query from the daily limit.

The program then preprocessed the string of titles to prepare for classification. Because many of the titles are not in English, some must be translated before the classification process. The program determines if the titles are in English using the langdetect library and if they are not detected as English, there is an attempt to translate them using the DeepL language translation API. The titles are then set to lowercase, all non-alphabetic characters are removed, stop words are removed and the remaining words are stemmed. The stopwords and stemming libraries are imported from the natural language toolkit (nltk).

The data is stored in json files with the following fields:

|  |  |  |
| --- | --- | --- |
| Name | Datatype | Description |
| author | String | The first name supplied for gender prediction |
| titles | String | A string of publication titles by the author, delimited by a semi-colon |
| cleanedtitles | String | A string the cleaned titles |
| gender | String | The predicted gender |
| probability | number | Certainty of predicted gender |
| count | number | Rows examined by Genderize.io to calculate the response |
| translation | string | A field that only appears if the titles were translated before cleaning |

The all the data processed was saved into the authorData.json file. If the author was predicted to be female with a count of over 20 and a probability greater than or equal to 90% then it was also saved into the authorFemaleData.json file.

This process, unfortunately, further reduced the size of the sample as the 1000 daily limit for Genderize.io meant only allowed me to predict the genders of 7218 first names.

The data collected to this point only contained the name, titles, and gender of the author. To gather further information about each author, the extra\_info.py program used the dblp’s API to gather extra information about each author from the authorData.json file. The extra data collected was stored in the authorExtraData.json file with the following data fields:

|  |  |  |
| --- | --- | --- |
| Name | Datatype | Description |
| name | String | A unique author name from the dblp |
| Pid | String | A unique author ID from the dblp |
| Gender | String | The predicted gender |
| gender probability | number | Certainty of predicted gender |
| gender count | number | Rows examined by Genderize.io to calculate the response |
| total publications | number | Total publications recorded by the dblp |
| Publications | List | List of publications |
| publications:title | String | Title of publication |
| Publications:translation | String | Optional field: Translation of the title |
| publications:cleaned titles | String | Cleaned title |
| publications:title:authors: | List | List of authors of the publication |
| publications:title:authors:name | String | Author name of the publication |
| publications:title:authors:pid | String | A unique author ID |
| publications:position | number | Positional index of the queried author in the author's list |
| publications:venue | String | Optional field: Location of the publication's presentation |
| publications:pages | String | Optional field: Page number of the publication in a journal |
| publications:volume | String | Optional field: Volume of the journal |
| publications:publisher | String | Optional field: Publisher's name |
| publications:year | String | Optional field: Year of publication |
| publications:type | String | Optional field: Type of publication |
| publications:key | String | The dblp's unique publication key |
| publications:doi | String | Optional field: Digital Object Identifier |
| publications:ee | String | Optional field: Electronic Engineering URL |
| publications:url | String | Optional field: URL of the publication |
| average pages | number | Optional field: Average number of publication pages of the author |
| average position | number | Average author position in research publications |
| average authors | number | Average number of authors per publication |
| most recent publication | number | Optional field: Year of most recent publication |
| title(string) | String | Publication titles by the author, delimited by a semi-colon from authorData (may not be complete) |
| cleaned titles | String | The cleaned titles from authorData (may not be complete) |
| all titles | String | A string of all cleaned titles |

Predicting the author’s most likely field of publication required a training and validation dataset. To generate this dataset data\_preprocessing.py scraped Wikipedia’s definitions of computer science fields from the site’s outline of computer science. This program used Wikipedia’s API to gather the content from each of the subfield pages. The content was chunked into sentences and then cleaned. The cleaning process included setting the text to lowercase, removing URLs, removing non-alphabetic characters, removing stopwords, and stemming the remaining words with NLTK library. The dataset was stored in dataset.csv with the following fields:

|  |  |  |
| --- | --- | --- |
| Name | Datatype | Description |
| subfield | String | Title of a subfield of computer science |
| description | String | Description of subfield |
| field | String | Title of a field in computer science |
| url | String | URL of Wikipedia’s subfield page |
| content | String | Cleaned string for classification |

## Classification

Four different classification models were used to predicts the author’s primary field in computer science:

* Naïve Bayes classifier with count vectors
* Naïve Bayes classifier with TF-IDF vectors
* Multinomial logistic regression classifier with count vectors
* Multinomial logistic regression classifier with TF-IDF vectors

The program model\_fitting.py was used to create these models using sklean – a machine learning library.

First the Wikipedia dataset was broken into a training set comprised of 75% of the data, and a validation set of 25% of the data. The labels (the computer science fields) were then encoded into integers.

The observations were then transformed into two types of vectors: count vectors and TF-IDF vectors. Both are representations of frequency of tokens in a corpus with count vectors being a straightforward integer representation of frequency of words and TF-IDF vectors penalizing common words and giving importance to more frequent terms.

Using the transformed vectors, the classifier models were trained. First, the Naïve Bayes classifier was trained. This classifier is based off the Bayes theorem. The multinomial Naïve Bayes classifier was used because the label is categorical and there are more than two possible labels. The other model type that was trained is the multinomial logistic regression classifier. This type of classifier is used for categorical labels and is an extension of logistic regression (which is a binary classifier) and it is called multinomial it is classifying more that two labels.

The classifiers models had the following accuracy (based off the validation set created with the Wikipedia pages):

|  |  |
| --- | --- |
| Model | Model Accuracy |
| Naive Bayes classifier with count vectors: | 0.704537071 |
| Naive Bayes classifier with TF-IDF Vectors: | 0.611213574 |
| Logistic regression classifier with count vectors: | 0.699004058 |
| Logistic regression classifier with TF-IDF vectors: | 0.706381409 |

Finally, the records were classified with models with the program classify.py. The results were outputted to results.csv and imported into Tableau for data visualization. The records were also processed with stats.py to gather some simple stats about the data.

# Results

Chart

Description automatically generated

After the data collection and cleaning process there were 87,112 records left to be analysed. 13.5% of the records were classified as female and 86.5% were classified as male, each with a accuracy of 90% and higher.

|  |  |  |
| --- | --- | --- |
| Gender | % of Total | Number |
| female | 13.55% | 11,806 |
| male | 86.45% | 75,306 |

The classifier process produced the following results:

### Naive Bayes Count Vector Field Prediction

|  |  |  |
| --- | --- | --- |
| Naive Bayes Count Vector Field Prediction | % of Total Number of Records | Number of Records |
| Theory of computation | 2.15% | 1,841 |
| Software engineering | 8.76% | 7,488 |
| Scientific computing | 9.97% | 8,525 |
| Programming languages and compilers | 2.23% | 1,909 |
| Mathematical foundations | 10.72% | 9,160 |
| Databases | 0.92% | 790 |
| Concurrent, parallel, and distributed systems | 3.54% | 3,025 |
| Computer graphics | 1.27% | 1,085 |
| Computer architecture | 2.07% | 1,773 |
| Communication and security | 15.03% | 12,844 |
| Artificial intelligence | 43.11% | 36,849 |
| Algorithms and data structures | 0.21% | 179 |

A picture containing application

Description automatically generated

### Naive Bayes TF-IDF Field Prediction

|  |  |  |
| --- | --- | --- |
| Naive Bayes TF-IDF Field Prediction | % of Total Number of Records | Number of Records |
| Theory of computation | 0.90% | 767 |
| Software engineering | 4.75% | 4,059 |
| Scientific computing | 3.05% | 2,611 |
| Programming languages and compilers | 0.97% | 827 |
| Mathematical foundations | 12.88% | 11,005 |
| Databases | 0.02% | 21 |
| Concurrent, parallel, and distributed systems | 0.94% | 807 |
| Computer graphics | 0.06% | 54 |
| Computer architecture | 0.24% | 206 |
| Communication and security | 10.89% | 9,310 |
| Artificial intelligence | 65.29% | 55,801 |

A picture containing timeline

Description automatically generated

### Logistic Regression TF-IDF Field Prediction

|  |  |  |
| --- | --- | --- |
| Logistic Regression TF-IDF Field Prediction | % of Total Number of Records | Number of Records |
| Theory of computation | 1.66% | 1,415 |
| Software engineering | 9.14% | 7,814 |
| Scientific computing | 8.22% | 7,023 |
| Programming languages and compilers | 1.51% | 1,290 |
| Mathematical foundations | 13.02% | 11,131 |
| Databases | 1.20% | 1,025 |
| Concurrent, parallel, and distributed systems | 2.81% | 2,401 |
| Computer graphics | 0.74% | 629 |
| Computer architecture | 1.57% | 1,339 |
| Communication and security | 15.13% | 12,931 |
| Artificial intelligence | 44.83% | 38,318 |
| Algorithms and data structures | 0.18% | 152 |

Timeline

Description automatically generated

### Logistic Regression Count Vector Field Prediction

|  |  |  |
| --- | --- | --- |
| Logistic Regression Count Vector Field Prediction | % of Total Number of Records | Number of Records |
| Theory of computation | 2.86% | 2,444 |
| Software engineering | 10.98% | 9,388 |
| Scientific computing | 11.55% | 9,871 |
| Programming languages and compilers | 2.55% | 2,179 |
| Mathematical foundations | 13.28% | 11,349 |
| Databases | 2.00% | 1,712 |
| Concurrent, parallel, and distributed systems | 4.44% | 3,796 |
| Computer graphics | 1.46% | 1,245 |
| Computer architecture | 2.45% | 2,090 |
| Communication and security | 14.53% | 12,418 |
| Artificial intelligence | 33.07% | 28,266 |
| Algorithms and data structures | 0.83% | 710 |

A picture containing bar chart

Description automatically generated

With all four classification models, Artificial Intelligence had the largest classification rate, however, the records seemed to be categorized more evenly with the multinomial logistic regression model. I chose this model to predict the fields of the authors between the genders.

|  |  |  |  |
| --- | --- | --- | --- |
| Gender | Logistic Regression Count Vector Field Prediction | % of Total Number of Records | Number of Records |
| female | Theory of computation | 2.78% | 328 |
| female | Software engineering | 13.32% | 1,573 |
| female | Scientific computing | 9.68% | 1,143 |
| female | Programming languages and compilers | 2.35% | 278 |
| female | Mathematical foundations | 14.09% | 1,663 |
| female | Databases | 2.40% | 283 |
| female | Concurrent, parallel, and distributed systems | 3.26% | 385 |
| female | Computer graphics | 1.70% | 201 |
| female | Computer architecture | 1.86% | 220 |
| female | Communication and security | 14.49% | 1,711 |
| female | Artificial intelligence | 33.44% | 3,948 |
| female | Algorithms and data structures | 0.62% | 73 |
| male | Theory of computation | 2.87% | 2,116 |
| male | Software engineering | 10.61% | 7,815 |
| male | Scientific computing | 11.85% | 8,728 |
| male | Programming languages and compilers | 2.58% | 1,901 |
| male | Mathematical foundations | 13.15% | 9,686 |
| male | Databases | 1.94% | 1,429 |
| male | Concurrent, parallel, and distributed systems | 4.63% | 3,411 |
| male | Computer graphics | 1.42% | 1,044 |
| male | Computer architecture | 2.54% | 1,870 |
| male | Communication and security | 14.54% | 10,707 |
| male | Artificial intelligence | 33.01% | 24,318 |
| male | Algorithms and data structures | 0.86% | 637 |

Chart, bar chart

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | median | mean |
| Number of publications of female authors | 6.00 | 24.76 |
| Number of pages per publications of female authors | 9.91 | 10.86 |
| Number of authors per publications of female authors | 3.50 | 3.93 |
| Position of female authors on publications | 1.00 | 1.36 |