Gender Prediction for Movie Database Actors: Methodology and Findings

Authors: Dario Santiago Lopez, Anthony Roca, ChatGPT-4o

Date: November 7, 2024

Abstract

This research aims to develop a gender prediction system for actors based on their first names within a movie database. We employed both a rule-based baseline model using the gender-guesser library and a custom trained classifier based on a Multinomial Naive Bayes model. We used character-level TF-IDF vectorization for feature engineering, trained and validated the model, and compared the performance of both models. Our findings highlighted various challenges related to class imbalance, the quality of labeled training data, and discrepancies in prediction outputs.

1. Introduction

The objective of this study is to predict actor genders in our movie database using the first names of the actors. Accurate gender identification is crucial for providing more detailed analytics and recommendations within a movie recommender system. To achieve this, we explored two approaches:

1. A baseline gender classifier based on a pre-built library (gender-guesser).
2. A custom-trained gender predictor using Multinomial Naive Bayes with character-based TF-IDF features.

2. Data Preparation

2.1 Dataset

The dataset used for training and evaluation was derived from an SQLite database containing actor information. The relevant table (gender_prediction) was structured to include the actor's first name, actual gender from Persons.csv (gender column with values 1 for female, 2 for male, and 3 for unknown), and two prediction columns (genderize_pred and ml_pred).

The **genderize_pred** column was populated by a previous attempt to use the Genderize API. However, limitations in the API (i.e., request limits and lack of consistent accuracy) led us to explore different approaches.

2.2 Training Data Extraction

To train our custom model, we extracted only the rows where the actual gender value was known and labeled as either 1 (female) or 2 (male). This enabled us to build a classifier with a focus on the two primary classes, excluding "unknown" labels during training.

3. Methodology

3.1 Baseline Model: Gender-Guesser Library

The baseline model utilized the **gender-guesser** library, which makes predictions based on common patterns in names. Its predictions were stored in the genderize_pred column, and its accuracy was calculated by comparing its predictions against the gender column for known actors.

3.2 Custom Model: Multinomial Naive Bayes Classifier

3.2.1 Feature Engineering: Character-Based TF-IDF Vectorizer

We employed the **TfidfVectorizer** from scikit-learn to transform first names into numerical features. The vectorizer analyzed characters, using **n-grams of length 2 to 5** to capture more nuanced name variations.

3.2.2 Data Splitting

The dataset was split into **80% training** and **20% testing** using train_test_split. This ensured that we had separate data for training the model and evaluating its performance.

3.2.3 Model Training

The **Multinomial Naive Bayes (MultinomialNB)** model was trained using the character-based TF-IDF features. The trained model was then used to predict the gender for the test set.

3.2.4 Confidence Threshold for Predictions

To handle cases where the model's confidence between male and female predictions was low, we used a **threshold** to classify predictions as "uncertain" (labeled as 3). If the absolute value of the difference between male and female probabilities was below the threshold, the gender was classified as uncertain.

3.3 Accuracy Evaluation

Two accuracy measures were obtained:

1. **Baseline Accuracy**: Calculated by comparing the genderize_pred values from the gender-guesser library against the actual gender labels.
2. **Multinomial Naive Bayes Accuracy**: Evaluated using predictions on the hold-out test set and compared with true gender values.

3.4 Database Integration

The predictions from both models were stored in the **SQLite database**:

- genderize_pred: For the baseline model predictions.
- ml_pred: For the custom model predictions.

4. Results and Discussion

4.1 Baseline Model Performance

The accuracy of the gender-guesser model was found to be **51.10%**, highlighting its limitations with names that were ambiguous or culturally uncommon.

4.2 Multinomial Naive Bayes Model Performance

The custom Multinomial Naive Bayes classifier achieved an accuracy of **71.15%**. However, we encountered significant issues:

- **Class Imbalance**: The training dataset had an imbalance (e.g., 353 female names vs. 423 male names). This affected the model's ability to learn equally from both classes.

- **Prediction Mapping Error**: During analysis, it was found that the model was inconsistently mapping labels—predicting male names as uncertain (3) more often than female names. This was due to the model overfitting to the "unknown" label when confidence between male and female was low.

4.3 Threshold-Based Uncertainty

The application of a **confidence threshold** to determine uncertainty initially led to unintended behavior, with almost all uncertain predictions being classified as male. Adjustments to the thresholding process were made to ensure a more balanced distribution of uncertain classifications.

4.4 Findings

- **Overfitting to Class 3**: Due to the high occurrence of "unknown" (3) labels in the gender column, the model initially overfitted and classified a large portion of names as uncertain.
- **Accuracy Comparison**: The Multinomial Naive Bayes model showed better performance than the baseline for certain types of names, particularly uncommon names not well covered by gender-guesser.

5. Reflection and Possible Extension of Research

5.1 Data Quality and Bias

The **person.csv dataset** that we used for training our custom gender prediction model presented several challenges. A significant portion of the names in this dataset were labeled as **unknown** (category 3). This high number of unknown names led to an imbalanced dataset, which impacted the effectiveness of our model. Specifically:

- **Bias Towards Androgynous/Unknown Labels**: Names in languages other than English were often classified as **androgynous or unknown**. This meant that our training data was **biased towards English names**, leading to better performance in predicting the genders of typical English names but poor generalization for non-English names.
- **Uneven Split of Male and Female Labels**: The dataset had an **unequal number of male and female names**. For instance, there were **423 male names** compared to **353 female names**, which caused the model to **overfit to male names**. This imbalance made it more challenging for the model to accurately classify female names, as the classifier became biased towards the majority class in the training data.

5.2 Comparison to Gender-Guesser Library

The **gender-guesser library** showed **much better predictions overall** compared to the ground truth labels we scraped from person.csv. This indicates that the gender labels in the dataset might not have been the most accurate or representative. In fact:

- **Poor Real-World Generalization**: The Multinomial Naive Bayes model trained on the person.csv dataset may have performed well on that dataset but is unlikely to have **good real-world performance** due to the **poor quality of the training data**.
- **Better Baseline Performance**: The **gender-guesser library** consistently outperformed the labels from person.csv for accurate gender labels, particularly for names from a diverse range of cultural backgrounds.

5.3 Future Directions

To address these issues and improve the overall effectiveness of the gender prediction model, the following strategies could be considered for future work:

5.3.1 Leveraging Gender-Guesser for Training Data

One potential improvement would be to use the **gender-guesser library's classifications** as the labels for training. The gender-guesser library has shown to be **more consistent and generalizable** across diverse names, which would result in higher quality training data. By training on gender-guesser's labels:

- The model would **inherit the generalization capabilities** of gender-guesser and be better suited for real-world scenarios.
- The **quality of the labels** would be significantly improved, allowing the model to learn from more accurate gender information.

5.3.2 Balancing the Dataset

The **uneven split** between male and female names in person.csv led to biased learning. To mitigate this, future work could:

- **Limit the size of the male dataset** to match the number of female samples, resulting in a balanced dataset.
- Alternatively, use **data augmentation** techniques or **re-sampling** to ensure that both classes are equally represented during training.

5.3.3 Improving Label Quality for Non-English Names

The current dataset struggled particularly with non-English names, which were often labeled as **unknown**. To improve the classification for these names:

- Consider **manually labeling** a subset of names from various cultural backgrounds to add diversity and representativeness to the training data.
- Explore the use of **external datasets** that provide more accurate gender labels for non-English names, enhancing the model's ability to generalize beyond English-speaking regions.

5.3.4 Cross-Validation and Additional Models

We could also improve our model training process by introducing **cross-validation** to get a more robust understanding of model performance across different splits of the data. Additionally, experimenting with other classification models such as **Logistic Regression, Support Vector Machines (SVMs), or ensemble methods** may yield better performance.

5.3.5 Threshold Adjustment

The confidence threshold we introduced to identify uncertain predictions initially led to unexpected results, with most uncertain names being labeled as male. In the future:

- We could explore **dynamic thresholding** that adapts based on the distribution of male and female names.
- Additionally, using a **weighted penalty** during training could help reduce bias towards male names.

5.4 Summary of Reflections

In summary, our approach to building a custom gender prediction model faced several challenges primarily stemming from **data quality** and **class imbalance**. Although the model performed well on the training data, its ability to generalize to diverse real-world data is limited due to the biased nature of person.csv. Future iterations of this project could benefit significantly from incorporating **high-quality labels**, **balanced datasets**, and **more robust model evaluation techniques**.

Ultimately, the key takeaway is that the **quality of training data** and ensuring **class balance** are crucial factors in developing an accurate and generalizable gender prediction model. By improving these elements, we can build a model that is both **effective** and **adaptable** to a broader range of real-world names.

6. Acknowledgments

This project was developed collaboratively by **Dario Santiago Lopez, Anthony Roca, and ChatGPT-4o** with support from the Canvas platform. Special thanks to ChatGPT for its assistance in refining the prediction logic and troubleshooting issues throughout the project.

7. References

- Gender-Guesser Library: GitHub
- Scikit-Learn Documentation: TfidfVectorizer