Gender Prediction from Name

Fundamentals of Artificial Intelligence Software Engineering Department

Addis Ababa Institute of Technology (AAIT)

Instructor: Debela Desalegn

Yeabsra Abera – ATE/1291/14

Samuel Tesfaye – ATE/0561/14

Anteneh Debebe – ATE/3036/14

Rekik Alemayehu – ATE/2752/14

Yeabsira Eyob – ATE/1640/14

**Gender Prediction from Name – AI Pipeline Report**

**1. Introduction**

This project focuses on building an AI model that predicts the gender of a person based solely on

their first name. The goal is to demonstrate a complete AI pipeline—from data processing to

deployment—using classic machine learning techniques. Such models can be useful in data

analytics, form processing, and language understanding tasks.

**2. Dataset Overview**

We used a dataset named 'name\_gender\_dataset.csv' containing two columns: 'Name' and

'Gender'. Rows with missing values were removed. Genders were mapped as: Female → 0, Male

→ 1.

**3. AI Pipeline Steps**

•

Data Loading: Loaded the dataset using pandas.

•

Preprocessing: Cleaned the data and converted names to lowercase.

•

Feature Extraction: Used character-level 2–3 n-grams via CountVectorizer.

•

Model Training: Trained a Logistic Regression classifier.

•

Hyperparameter Tuning: Used GridSearchCV to optimize 'C' and 'solver' parameters.

•

Evaluation: Evaluated the model using accuracy and classification report.

•

Deployment: Built a Flask web app for real-time predictions.

**4. How It Works: N-grams and Logistic Regression**

The model uses character-level n-grams to break names into overlapping character sequences. For

example:

- "Aisha" → ["ai", "is", "sh", "ha"] - "Raj" → ["ra", "aj", "raj"] - "Emily" → ["em", "mi", "il", "ly", "emi",

"mil", "ily"]

Logistic Regression assigns weights to these n-grams. Patterns like "sha" and "ette" are strongly

associated with female names, while "raj" or "mo" may be common in male names. During

prediction, the model uses these learned weights and the sigmoid function to predict a probability

between 0 and 1 (female or male).

**5. Model Evaluation & Results**

The model was evaluated on a test set using accuracy and classification metrics. Example

predictions:

- "Michael" → Male (97.3%) - "Aisha" → Female (94.1%) - "Raj" → Male (89.6%) - "Emily" →

Female (98.4%) - "Sara" → Female (95.5%)

**6. Flask Web App Deployment**

A simple web application was built using Flask to allow users to input a name and get the predicted

gender. The app loads the saved model and vectorizer, transforms the input, and returns the result

with confidence.

**7. Sample Predictions**

User Input: "Aisha" → Output: Female (94.1% sure) User Input: "Raj" → Output: Male (89.6% sure)

**8. Conclusion**This project demonstrates a full AI pipeline using classical machine learning. Character n-grams

proved effective for capturing name-gender associations. The model achieved strong accuracy and

was successfully deployed via a web interface.