Footprint-based Sex Prediction — Notebook Overview

Access the notebook from here

Stage 1: Load and Explore the Dataset (EDA) — What / Why / Outcome

In this stage we perform Exploratory Data Analysis (EDA). EDA helps us understand the data before building models. Below are the common tasks we perform and why they are useful:

1. Load the dataset: We read the CSV file into a table (called a DataFrame) so we can inspect and manipulate it.
    - Outcome: The data is available in memory for further steps.
2. Inspect basic information (shape, columns, datatypes): We check how many rows and columns there are and what type each column is (numbers, text, etc.).
    - Why: This reveals missing columns, unexpected types, or obvious data errors.
3. Display sample records and descriptive statistics: We look at a few rows and summary statistics (mean, std, min, max) for numeric columns.
    - Outcome: Helps to detect outliers and understand typical measurement ranges.
4. Missing value checks: Confirm whether there are empty or missing values that require cleaning.
    - Why: Missing data can break models or bias results. If missing values exist, we decide how to handle them (drop, fill, or estimate).
5. Target balance checks: We check how many samples belong to each sex and whether the dataset is balanced.
    - Outcome: Knowing class balance helps choose evaluation metrics and model strategies (e.g., balancing).
6. Correlation and visualizations: Heatmaps and histograms help us see relationships between features and the target.
    - Why: Highly correlated features may be redundant; some features might show strong differences between sexes which makes them useful for prediction.

By the end of this stage you should be able to answer: Is the data clean enough? Are there obvious predictive features? Is the target variable balanced?

Quick notes for non-technical readers:

- You do not need to understand every code line to follow the notebook — read the explanatory cells to learn the ideas.
- If you run into an error while executing a code cell, read the troubleshooting section near the end before asking for help.

Prerequisites (what to install):

- Python 3.8+ with packages listed in `requirements.txt` (pandas, numpy, scikit-learn, xgboost, seaborn, matplotlib, joblib).

How to run:

1. Run cells top-to-bottom in order.
2. When long-running steps appear (e.g., grid search), you can skip or run with smaller parameter grids.

Expected outcomes:

- Trained models saved to the `models` folder.
- Visual plots showing distributions, correlations, and model performance.
- A simple callable function at the end to predict sex for a single new footprint measurement.

If you are ready, continue to the next section where we import libraries and load the dataset.

↳ 14 cells hidden

Stage 2: Model Training & Evaluation Plan — What / Why / Outcome

In this stage we train several machine learning models and compare their performance.

What models we use and why:

- Decision Tree: A simple, easy-to-interpret model that splits data by feature thresholds. Good for baseline understanding.
- Random Forest: An ensemble of decision trees that reduces overfitting and generally performs well on tabular data.
- XGBoost: A powerful gradient boosting algorithm that often achieves high accuracy with careful tuning.
- Support Vector Machine (SVM) with RBF kernel: Effective for high-dimensional data; sensitive to feature scaling.

Evaluation approach and why it matters:

- We split the data into training and test sets. Models learn from the training set and are evaluated on the test set to estimate real-world performance.
- We use accuracy, precision, recall, and F1-score to provide multiple perspectives on performance.
- Confusion matrices show how many samples are correctly/incorrectly classified for each class.

Expected outcome:

- A short list of model performance numbers and visualizations so you can pick the most appropriate model for your needs (fast vs. accurate vs. interpretable).

↳ 29 cells hidden

Stage 3: Feature Importance & Model Interpretability — What / Why / Outcome

Why interpretability matters:

- Knowing which features influence predictions helps us trust the model and extract domain insights (forensic or biometric importance).
- Feature importance from tree models (like Random Forest) suggests which measurements differ most between sexes.

What we will look at:

- Feature importance: a ranked list of features showing their relative contribution to predictions.
- Learning curves: show training vs. validation performance as we increase the training set size. Useful to diagnose overfitting or underfitting.
- ROC curve and AUC: help evaluate binary classifiers by measuring the trade-off between true positive rate and false positive rate at different thresholds.

Expected outcome:

- A short list of top features likely to be most predictive.
- Learning curves that indicate whether more data could improve performance (if validation score is still rising).
- AUC scores where values closer to 1.0 indicate excellent discrimination between classes.

↳ 11 cells hidden

Stage 4: Model Comparison & Reporting — What / Why / Outcome

Purpose of this stage:

- After training and tuning, we compare all models using consistent metrics and visualizations so you can choose the best model for deployment or further study.

Key steps and explanations:

1. Collect performance metrics (accuracy, CV accuracy):
    - What: We summarize each model's performance on the test set and (if available) cross-validation performance.
    - Why: Using multiple metrics reduces the chance of being misled by a single number (e.g., accuracy alone can be misleading if classes are imbalanced).
    - Outcome: A table of model scores that is easy to read and sort.
2. Visual comparison (bar plots):
    - What: Bar charts compare models side-by-side by accuracy or CV accuracy.
    - Why: Visuals make it quick to identify top performers and relative gaps.
    - Outcome: A clear picture of which models perform best.
3. PCA (Principal Component Analysis) projection to 2D:
    - What: PCA reduces many numeric features down to two components for visualization.
    - Why: It helps you see whether classes (male/female) separate naturally in a low-dimensional projection.
    - Outcome: A scatter plot where clusters or separation suggests that the features contain discriminative information.
4. Confusion matrices grid:
    - What: For each model, a confusion matrix shows true vs predicted counts for each class.
    - Why: It reveals whether models are biased toward one class (e.g., predicting male more often) and where mistakes occur.
    - Outcome: An easy way to inspect which errors are most common and assess model reliability in real-world use.
5. Collect final predictions and comparison table:
    - What: We compile actual and predicted labels from all models for direct comparison on sample rows.
    - Why: Useful for qualitative checks—does the best model make reasonable predictions on individual examples?

    - Outcome: A small table showing predictions side-by-side for quick manual inspection.

How to interpret results as a non-technical reader:

- Look for models with consistently high accuracy and CV accuracy and low variance between runs.
- Use confusion matrices to check whether a model systematically mislabels a class.
- PCA separation is an intuition tool — good separation is promising but not a guarantee of model performance.

Final decision guidance:

- If you need interpretability (to explain decisions), prefer Decision Tree or Random Forest (feature importance).
- If you need the best raw accuracy and have computational resources, XGBoost or a tuned Random Forest are commonly good choices.
- If models perform similarly, prefer the simpler/faster model for deployment.

Stage 5: Saving Models, Loading, and Troubleshooting

↳ 6 cells hidden

Where models are saved:

- Trained models are saved to the `models` directory by default in this notebook (look for `.pkl` files). You can change the path in the saving cell.

How to load models:

- Use `joblib.load('models/your_model.pkl')` to reload a model object. Ensure you also load the scaler if you scaled features during training.

Common issues and quick fixes:

- FileNotFoundError when loading models: Check the path and that the `.pkl` files exist.
- Shape mismatch when predicting: Make sure the order and number of features you pass to the prediction function match the training features (same columns).
- Missing required packages: Install from `requirements.txt` or run `pip install -r requirements.txt`.
- Long run times: If GridSearchCV or cross-validation is slow, reduce the number of parameter combinations or CV folds.

If you are not technical and want help: Copy the full error message and the small code cell you ran; these will help identify the problem quickly.

↳ 7 cells hidden

Research Conclusion

Summary of Work and Key Findings

This research investigated whether an individuals biological sex can be predicted from footprint biometric measurements using classical machine learning methods. We followed a standard ML workflow: exploratory data analysis, data preparation (encoding, splitting, scaling), model training (Decision Tree, Random Forest, XGBoost, SVM), cross-validation, hyperparameter tuning, model comparison, and interpretability analysis.

Major findings:

- The dataset contains a variety of geometric footprint measurements that show discriminatory power between sexes.
- Ensemble methods (Random Forest, XGBoost) and SVM produced the highest accuracy. A tuned Random Forest often balances accuracy and interpretability.
- Feature importance analysis highlighted a small set of measurements (see feature importance plot) that contribute most to the prediction, suggesting these features have potential forensic value.

Quantitative Results (what to expect)

- You will find model accuracy, precision, recall, F1-score, and AUC reported in the results tables and plots in this notebook. These metrics quantify different aspects of performance: accuracy for overall correctness, precision/recall for class-specific behavior, F1 for a harmonic balance between precision and recall, and AUC for ranking performance across thresholds.

Limitations and Caveats

1. Dataset bias and representativeness:
    - The dataset used here may not represent global population diversity (age ranges, ethnicities, footwear habits). Models trained on this data could perform worse when used on different populations.
2. Measurement quality and protocol:
    - The predictive power depends heavily on how footprints were measured. Inconsistent protocols or noisy measurements reduce real-world performance.
3. Overfitting risk:
    - Very high reported accuracies may indicate overfitting, especially if model complexity is high and if cross-validation was not carefully stratified. Use a hold-out test set and cross-validation scores together to judge generalization.
4. Binary label simplification:

  - This work treats sex as a binary variable (Male/Female) as recorded in the dataset. Real-world biological sex and gender identity are more complex; careful ethical consideration is required before applying such models.

Ethical Considerations

- Privacy: Biometric data like footprints are sensitive. Ensure proper consent and data governance when collecting and storing such data.
- Fairness: Assess whether model performance differs across demographic groups. If disparities exist, do not deploy without mitigation.
- Intended use: These models can be useful for forensic or anthropological research, but they should not be used for high-stakes automated decisions without human oversight.

Reproducibility and How to Re-run the Study

- Environment: Use the `requirements.txt` provided to install Python packages. Prefer creating a virtual environment to keep dependencies isolated.
- Data: The primary data file is `dataset/footprint_dataset_5000rows.csv`. If you modify or augment the data, record the changes and random seeds.
- Random seeds: The notebook sets random_state where appropriate (e.g., RandomForest, train_test_split) for reproducibility. Keep these seeds fixed for exact reproduction.
- Long-running steps: Hyperparameter searches can be costly. To reproduce faster, reduce the grid size or run GridSearchCV with fewer CV folds.

Practical Recommendations and Next Steps

1. External validation: Test the best models on an independent dataset collected under different conditions to measure generalizability.

2. Collect more diverse samples: Increase demographic and environmental diversity (age groups, populations, barefoot vs shod measurements) to reduce bias.

3. Model simplification and interpretability: If deployment needs transparency, prefer models that provide interpretable rules or use post-hoc explainability tools (like SHAP) to explain predictions.

4. Robustness testing: Evaluate model robustness to measurement noise and missing features. Consider training with augmented or noisy samples.

5. Ethical audit: Before any real-world deployment, perform a documented ethical audit covering consent, privacy, fairness, and intended use cases.

Final Verdict

- The results in this notebook show that footprint measurements contain meaningful biometric signals that can be exploited by machine learning models to predict sex with high accuracy on the available dataset. However, this conclusion is conditioned on the dataset and experimental setup used here.
- Before operational use, further validation, broader data collection, fairness assessments, and careful ethical review are required.