Detecting and Mitigating Bias in AI

James Housteau & Prathyusha Pateel

November 27, 2024

Project Sponsor: Conagra Brands
Point of Contact: Josh Mattingly, Manager, AI Strategy

Contents

1 Project Scope and Structure

1.1 General Scope and Importance

As artificial intelligence (AI) systems are increasingly utilized in various applications in modern society, bias permeates each stage of AI development, from data collection and preparation to using outputs from machine learning (ML) models for decision-making. This issue warrants attention from every organization engaged in data science, as neglecting to address it may lead to discriminatory outcomes affecting specific groups, breaching ethical standards, and, in the worst case, violating legal regulations.

Conagra Brands wants to tackle bias in AI-generated data by exploring effective strategies for detecting and mitigating it. The ideal solution is an end-to-end machine learning pipeline that monitors every stage of the pipeline for any signs of bias. Additionally, systems will be in place to identify the types and levels of bias, along with suggestions for mitigation strategies and their implementation.

We were expected to work on addressing a specific aspect of bias in AI research of our choosing. Our work would be a stepping stone for the broader initiative of detecting and mitigating bias in AI-generated data. We have been granted the flexibility to define our project and hypothesis, which will contribute to their overarching objectives.

To this end, they have structured the project into the following phases.

- **Literature Review:** We undertook a comprehensive literature review to understand the current research on Bias in AI and to guide our selection of a project that matches our skills and interests. Each team member was required to review three academic papers.
- **Proposal and Research:** We created a comprehensive project proposal that details our focus, hypothesis, methodology, intended data usage, and evaluation metrics and objectives. We could adjust the scope if new challenges or directions arose later in the project, we could adjust the scope.
- **Proof of Concept and Conclusions:** In the final phase, we were required to showcase our project, document our conclusions, and write the necessary reports. We shared our code base using a GitHub repository with the project sponsor.

2 Problem Statement and Objectives

2.1 Initial Problem Statement

Even though bias in AI affects multiple domains, our team chose to focus on AI-driven marketing models, a domain that interests us and aligns with the general scope of the project and the business priorities of generating customer trust and brand equity. We initially thought that limiting our scope in this way would benefit the project, as detecting and mitigating bias in AI systems is highly contextual. We believed we could select marketing tasks that are generally automated using ML and concentrate on how to detect and mitigate bias in those specific contexts.

2.2 Initial Objectives

1. **Synthetic Data Generation:** We wanted to generate realistic synthetic marketing data using AI that mimics real data from Conagra Brands to aid in our bias detection and mitigation research. This would ensure data privacy and serve the broader objective of focusing on AI-generated data. We expected to have access to limited proprietary datasets from Conagra to validate our data.

2. **Bias Mitigation Pipeline:** We aimed to create a modular end-to-end pipeline that takes care of all the steps from data generation to bias detection and mitigation. It would also enable comparative analysis between different mitigation strategies to help choose the one that mitigates bias without significantly affecting performance. We wanted to tailor this pipeline for specific tasks designed explicitly for Conagra's marketing needs.

2.3 Revisions to the Initial Plan

We had to make the following revisions to our initial plan as we faced some challenges.

1. **Data:** Due to privacy constraints, proprietary datasets from Conagra were unavailable. We only received the marketing data variables that they collect and store. We tried to generate synthetic data by using public demographics data and generating consumer preference, engagement, and transaction data. However, the synthetic data we generated lacked predictive power and couldn't mimic the real-world data. So, even though we spent a lot of time on this, ultimately, we had to use public datasets to test our pipeline.

2. **Interactive Pipeline:** Initially, we created a rigid pipeline that automated all the steps from data generation to bias mitigation. However, we recognized the need for user input at various stages due to the context-dependent nature of bias detection and mitigation, so we developed an interactive web-based application using Streamlit.

3. **Generalization:** During our project, we realized that although bias detection and mitigation are contextual, limiting our scope to a specific domain doesn't make sense because all the different user inputs and context-dependent suggestions we created are generalizable.

3 Methodology

3.1 Initial Prototype Pipeline

The initial pipeline was modular, end-to-end, and included the following steps:

- **Data generation:** Data generation was accomplished using Python's Faker library and statistical distributions based on market research to define correlations between variables. However, the complexity required for realistic data was limited.

- **Model training and evaluation:** Basic classification and clustering models were trained, and their performance was evaluated to compare it with the mitigation strategies.

- **Bias detection:** For the supervised model, bias detection metrics such as demographic parity, equalized odds difference, and selection rate were calculated. For the unsupervised model, cluster bias, silhouette score, representation metrics, etc., were also calculated.

- **Mitigation strategies:** Mitigation strategies such as reweighing, constrained optimization for fairness, and equalized odds adjustment were implemented.

- **Analysis system:** The system integrated AI-driven analysis capabilities for automatic metric interpretation, business impact assessment, evaluation of mitigation effectiveness, and recommendation generation.

- **Core Pipeline Architecture**

Listing 1: Pipeline Directory Structure

```
pipeline/
          ai_analyzer.py      # AI-driven analysis of results
          detection.py        # Bias detection implementations
          generate.py      # AI-driven analysis of results
          mitigation.py       # Bias mitigation strategies
          model.py           # Model management and abstractions
          output.py          # Structured output handling
```

3.2 Data Generation using Deep Learning with SDV

To achieve the required complexity, we decided to leverage the Synthetic Data Vault (SDV) framework using deep learning. We created a hierarchical data generation pipeline that started with publicly available base demographic data from the Python library `folktables` and generated consumer preferences, marketing campaigns, engagement metrics, and transaction data. The final dataset integrated real-world geographic distribution across six states (13.4M transaction records), age-appropriate education levels, income-correlated purchasing patterns, channel-specific engagement rates, and product category affinities, and was closer to real-world data. The following validation metrics were achieved:

- **Demographics:** 100% match on distribution tests

- **Consumer Preferences:** 40% accuracy on preference metrics

- **Marketing Campaigns:** 100% validity score

- **Engagement Data:** 100% structural validity

- **Transaction Data:** 72% match on key business metrics

3.3 Interactive Pipeline Prototype

We decided to create an interactive web-based prototype instead of our initial pipeline, which was functional but rigid because bias mitigation is highly context-dependent and requires flexibility to incorporate different fairness notions and mitigation techniques tailored to specific datasets and tasks. Using `Streamlit`, we aimed to generalize the pipeline and make it adaptable to various datasets and scenarios rather than limiting it to the particular context of Conagra's marketing models. We wanted to provide a user-friendly platform to guide users through the entire process of detecting and mitigating bias by offering suggestions along the way and giving them more control in the process.

3.3.1 Pipeline Architecture

Listing 2: Interactive Application Structure

```
app/
          scripts/
                preprocessing.py     # Data preprocessing
                modeling.py         # Model training
                bias_detection.py   # Detection strategies
                detection_metrics.py   # Bias metric definitions
                bias_mitigation.py  # Mitigation strategies
                suggest_mitigators.py # Mitigation recommendations
          app.py                  # Main Streamlit application
```

3.3.2 Key steps

1. **Data Upload:** Users can upload their data in CSV format. The app provides a preview of the dataset.

2. **Preprocessing:** Users can select the features they wish to use from the data, along with the target column and sensitive features. Optional preprocessing includes displaying descriptive statistics and providing options for handling missing values and outliers, as well as applying binning (only for sensitive features and the target column), encoding, and scaling for each column. The data is split into training and testing sets before preprocessing, fitting the encoders, scalers, etc., to the training data, and then applying the same to the test data. Recommendations for these options are suggested and displayed based on the column type, unique values, and whether they are sensitive features or the target column. The preprocessed data is then displayed.

3. **Base Model Selection:** The user can choose a task, such as classification or regression. Some relevant models are displayed for selection. Then, hyperparameters are suggested but can be customized. Next, the models are trained, and the best model is selected based on performance metrics; for example, accuracy is used for classification. Its hyperparameters are then displayed. A model comparison plot of the performance metrics is also shown.

4. **Metrics:** Users are guided through a process of choosing fairness notions that best suit their tasks and applications. Then, the bias detection metrics relevant to the fairness notion for the base model are displayed here, along with the performance metrics.

5. **Bias Mitigation:** Based on the metrics observed from bias detection, privileged and unprivileged groups are suggested, but users can change them. Then, users can choose from the suggested mitigation strategies. For now, only Reweighing, Adversarial Debiasing, and Equalized Odds Adjustment have been implemented. However, the code is modular, so new methods can be added easily. Bias detection and performance metrics are displayed for each strategy. The user can then select comparison metrics to plot, which are then displayed.

4 Results and Analysis

4.1 Bias Detection Findings in the prototype pipeline

The implementation revealed significant insights through both supervised and unsupervised analysis:

4.1.1 Supervised Model Results

- Equalized Odds Difference: 1.0000 (indicating significant disparities)
- Demographic Parity Difference: 0.1129 (moderate selection rate disparities)
- Impact: Potential unfair treatment in marketing decisions

4.1.2 Unsupervised Model Results

- Silhouette Score: 0.4808 (indicating reasonably distinct clusters)
- Cluster Bias Range: 0.0559 to 0.1229 (suggesting disproportionate representation)
- Impact: Need for careful interpretation of customer segments

4.2 Sample Results for the interactive pipeline

The following are sample results generated and displayed in the interactive web based application.

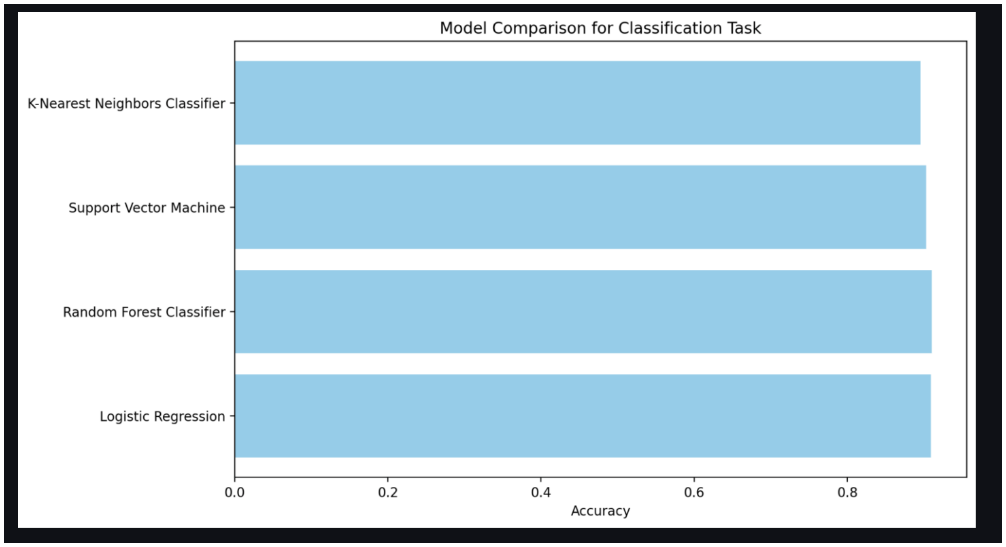


Figure 1: Model Comparison for Classification Task (Pre-Mitigation)

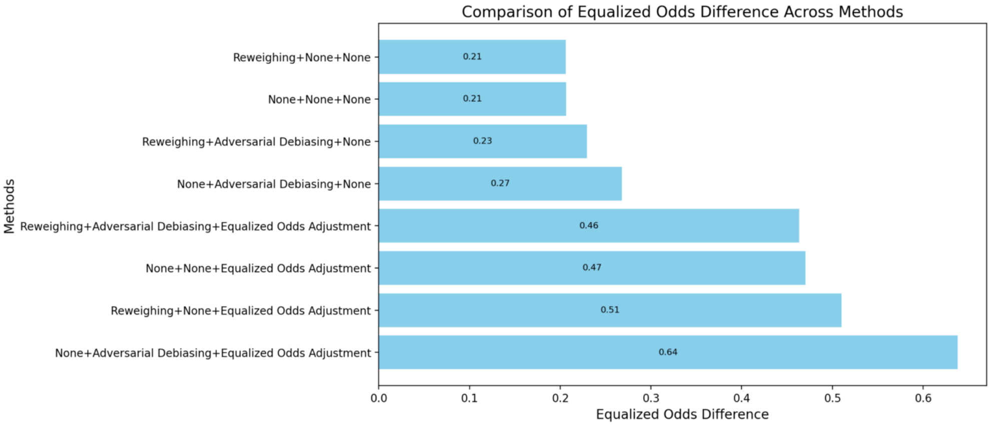


Figure 2: Comparison of Equalized Odds Difference Across Methods (Post-Mitigation)

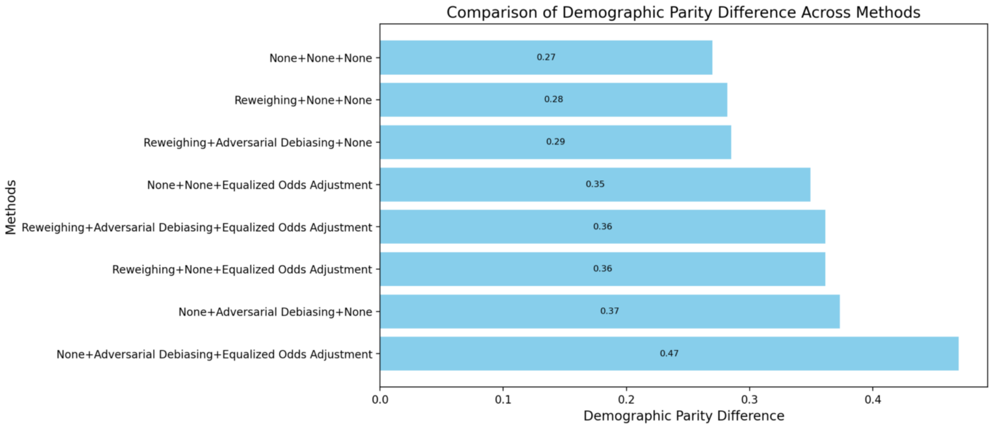


Figure 3: Comparison of Demographic Parity Difference Across Methods (Post-Mitigation)

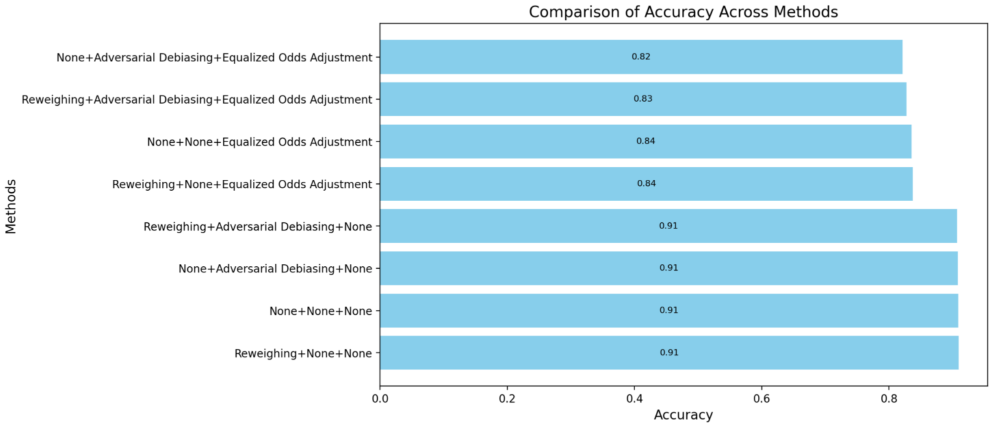


Figure 4: Comparison of Accuracy Across Methods (Post-Mitigation)

5 Future Work

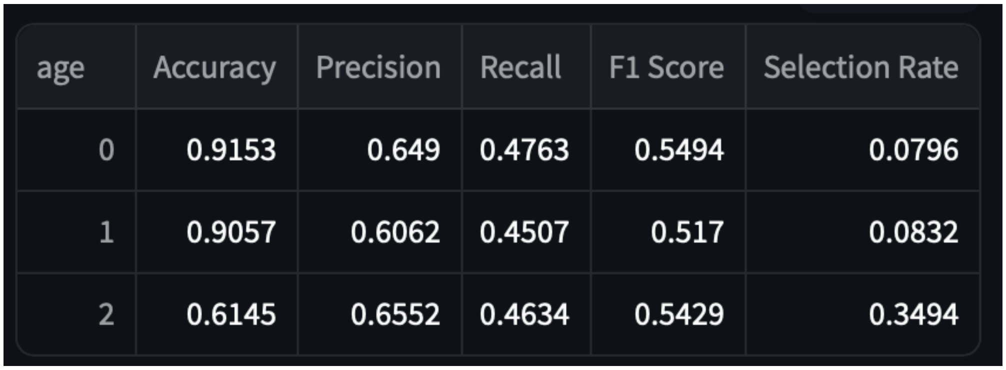
- Add more detection metrics and mitigation strategies.
- Test the app on multiple datasets to determine its flexibility.
- Make it applicable to other tasks, such as regression, clustering, and recommendation. The functionality for these tasks has already been added but needs testing.
- Add capabilities to handle large-scale data for scalability.
- Provide more descriptions of each fairness notion and explain why they are suggested. Create a questionnaire to help gather better suggestions.
- Improve the function to suggest unprivileged and privileged groups.
- Add explainability tools like SHAP values to make bias mitigation decisions more transparent to users and stakeholders.

6 Conclusion

This capstone project successfully developed a framework for detecting and mitigating bias in AI-driven marketing systems. The solutions implemented provide a foundation for ensuring fair and ethical AI deployment while maintaining effectiveness. The recommendations offer clear pathways for future implementation and improvement.

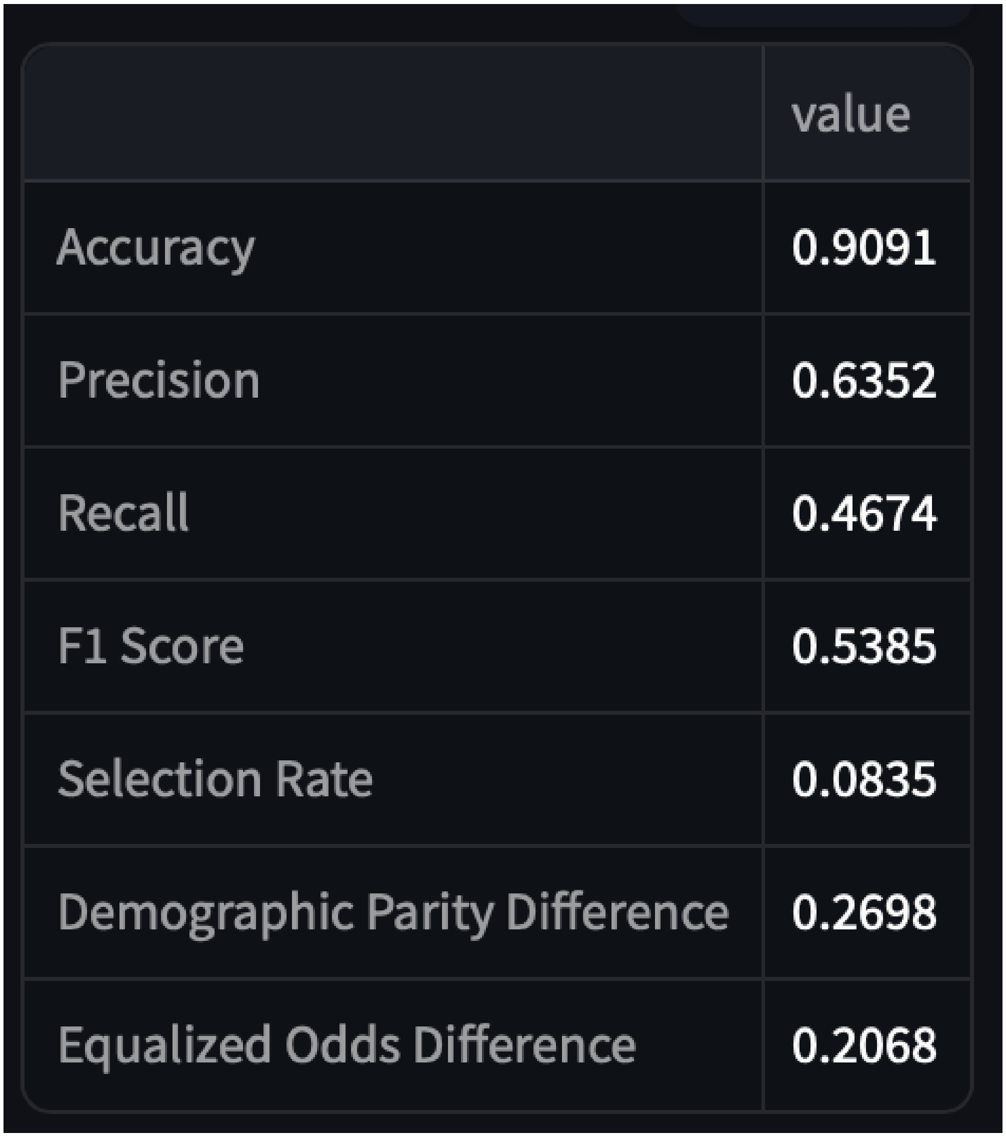
7 Appendix

The following are sample snippets from the interactive pipeline.



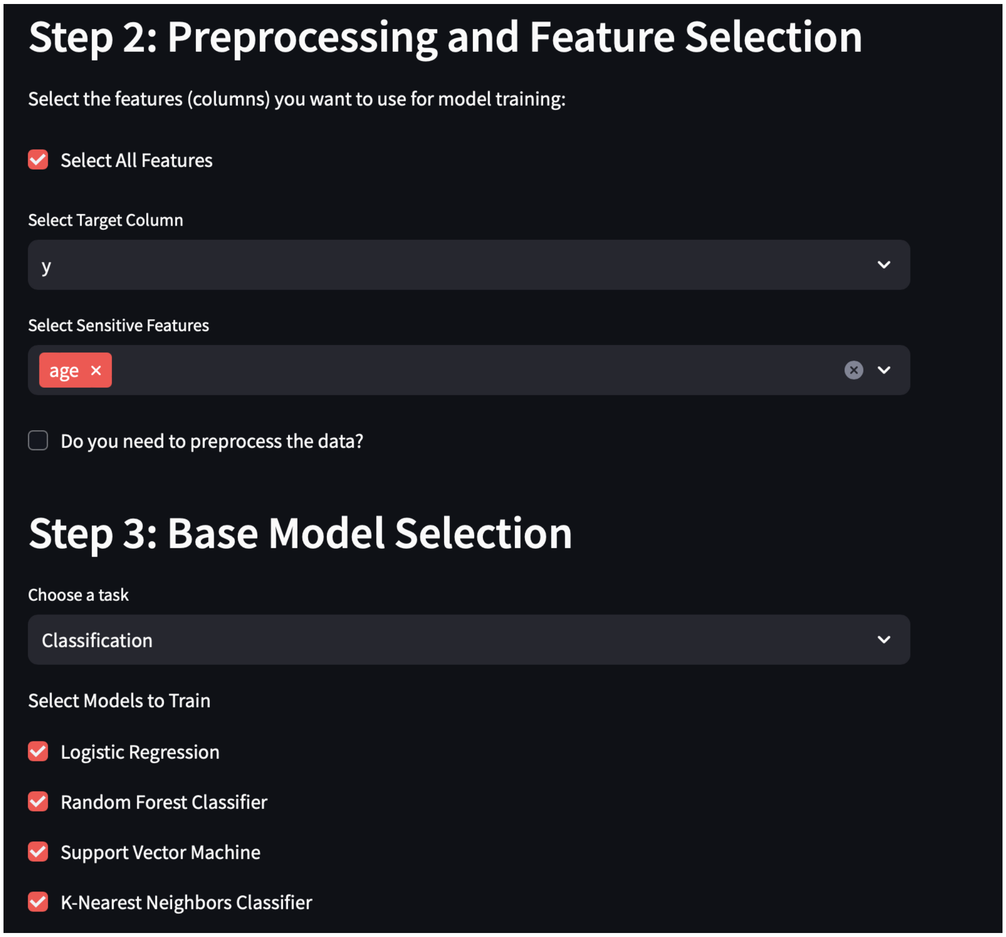
age	Accuracy	Precision	Recall	F1 Score	Selection Rate
0	0.9153	0.649	0.4763	0.5494	0.0796
1	0.9057	0.6062	0.4507	0.517	0.0832
2	0.6145	0.6552	0.4634	0.5429	0.3494

Figure 5: Group wise metrics



	value
Accuracy	0.9091
Precision	0.6352
Recall	0.4674
F1 Score	0.5385
Selection Rate	0.0835
Demographic Parity Difference	0.2698
Equalized Odds Difference	0.2068

Figure 6: Overall metrics



Step 2: Preprocessing and Feature Selection

Select the features (columns) you want to use for model training:

☑ Select All Features

Select Target Column: y

Select Sensitive Features: age ×

□ Do you need to preprocess the data?

Step 3: Base Model Selection

Choose a task: Classification

Select Models to Train

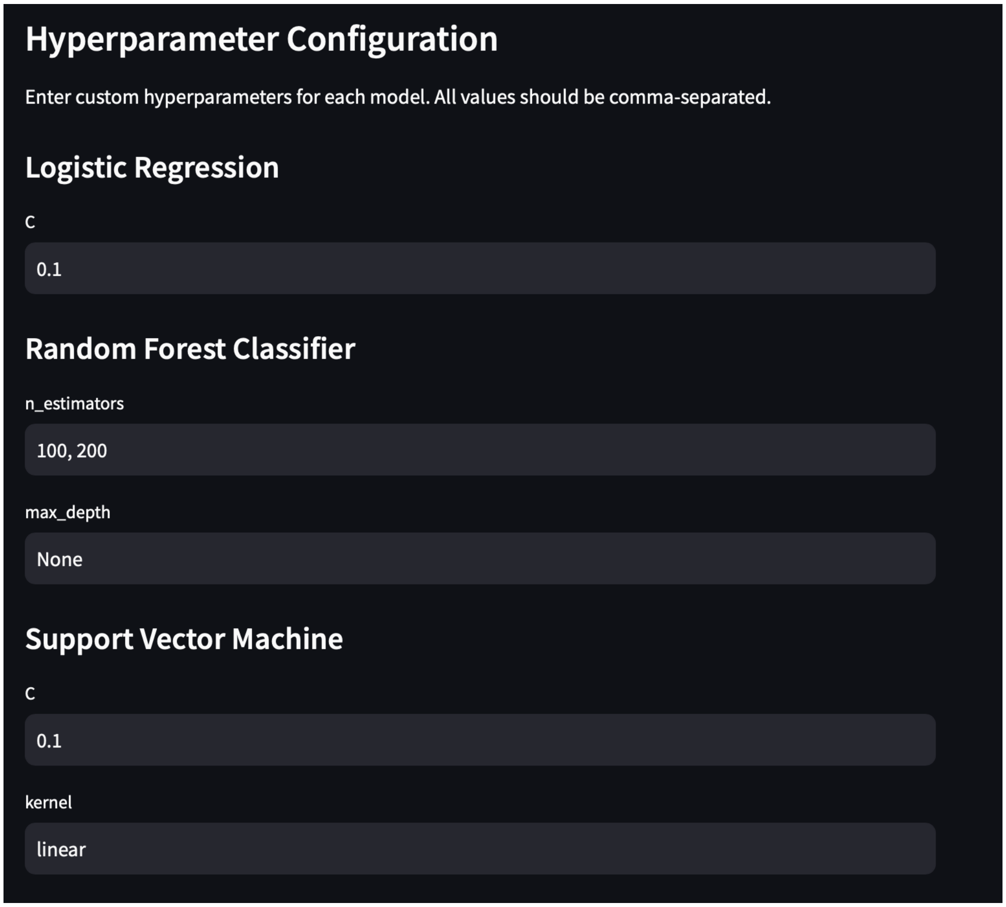
☑ Logistic Regression

☑ Random Forest Classifier

☑ Support Vector Machine

☑ K-Nearest Neighbors Classifier

Figure 7: Preprocessing and Model Selection



Hyperparameter Configuration

Enter custom hyperparameters for each model. All values should be comma-separated.

Logistic Regression

C

0.1

Random Forest Classifier

n_estimators

100, 200

max_depth

None

Support Vector Machine

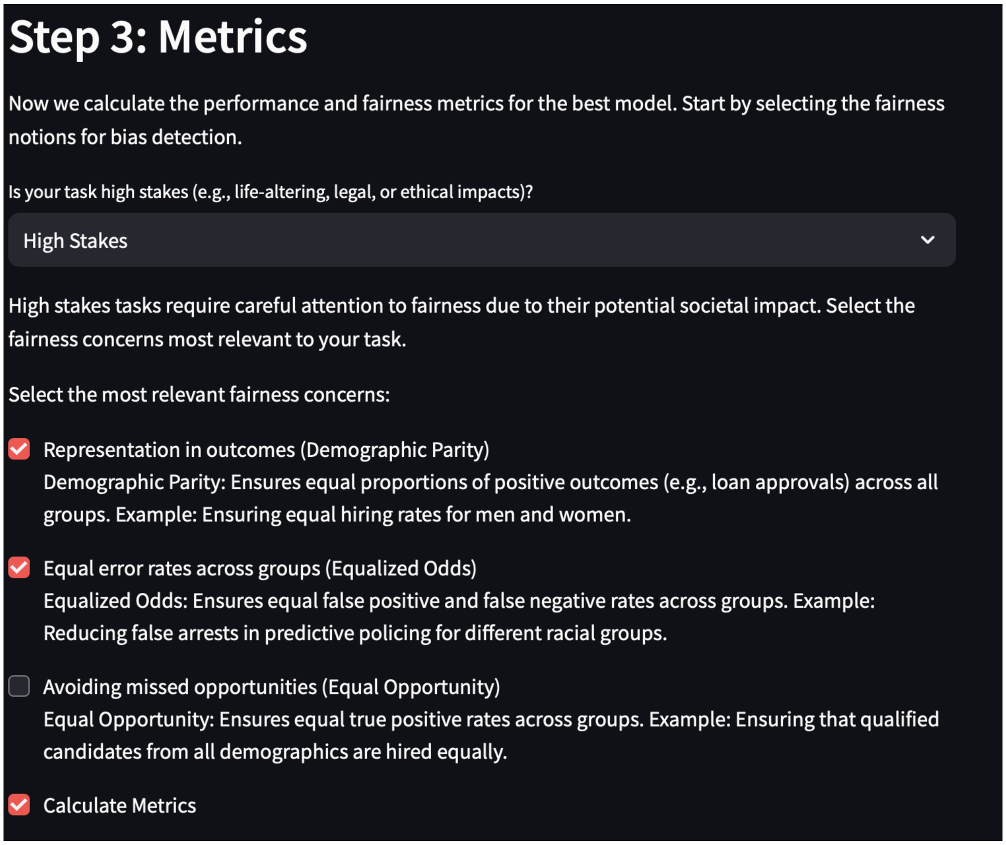
C

0.1

kernel

linear

Figure 8: Hyperparameter selection



Step 3: Metrics

Now we calculate the performance and fairness metrics for the best model. Start by selecting the fairness notions for bias detection.

Is your task high stakes (e.g., life-altering, legal, or ethical impacts)?

High Stakes

High stakes tasks require careful attention to fairness due to their potential societal impact. Select the fairness concerns most relevant to your task.

Select the most relevant fairness concerns:

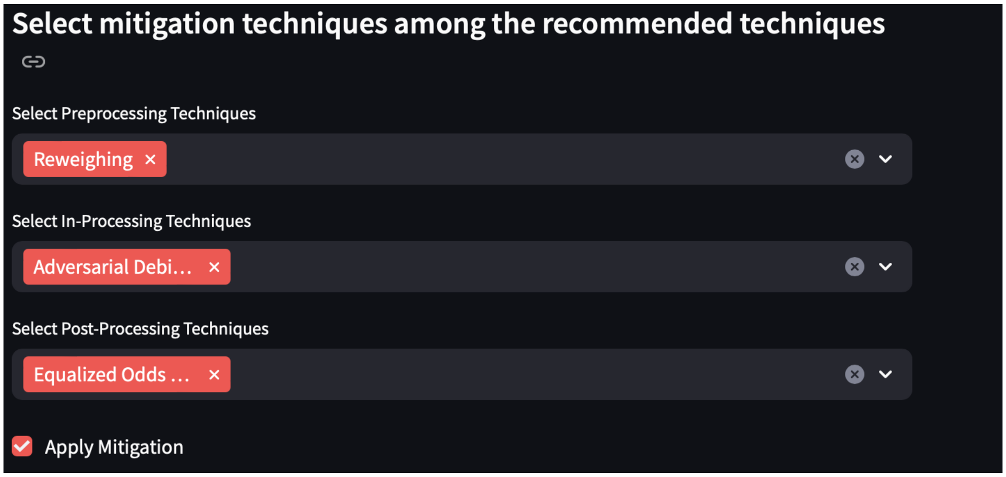
☑ Representation in outcomes (Demographic Parity)
Demographic Parity: Ensures equal proportions of positive outcomes (e.g., loan approvals) across all groups. Example: Ensuring equal hiring rates for men and women.

☑ Equal error rates across groups (Equalized Odds)
Equalized Odds: Ensures equal false positive and false negative rates across groups. Example: Reducing false arrests in predictive policing for different racial groups.

□ Avoiding missed opportunities (Equal Opportunity)
Equal Opportunity: Ensures equal true positive rates across groups. Example: Ensuring that qualified candidates from all demographics are hired equally.

☑ Calculate Metrics

Figure 9: Choosing fairness notions and bias detection metrics



Select mitigation techniques among the recommended techniques

Select Preprocessing Techniques: Reweighing ×

Select In-Processing Techniques: Adversarial Debi… ×

Select Post-Processing Techniques: Equalized Odds … ×

☑ Apply Mitigation

Figure 10: Choosing mitigation techniques among suggested mitigation techniques