# Probes:

[Probes:](#_vg1iwbr9iwh0)

[Experiment with comorbidities dataset:](#_fy32qzannrnx)

[For](#_dchmu6vnzweh) GPT-4.1

[prompt 1:](#_6kj9wbydog4t)

[response:](#_8zhzav7hyd2i)

[Prompt 2:](#_qepe6x4dlp7y)

[Respon](#_t1z2int0wrl4)[Response:](https://docs.google.com/document/d/1jy5om1fEVep6ivE4KJZWonK54pfrYdgGTbkDTwI7vL0/edit?tab=t.0#heading=h.8zhzav7hyd2i)[se:](#_t1z2int0wrl4)

[Prompt 3:](#_bro03yp3gk62)

[Response:](#_9wkzzuigzbuu)

[Gemini 2.5 pro](#_hgn1m1aguo7p)

[Prompt 1](#_uassl3rctzr9)

[Response](#_f77ap12ee61j)

[Prompt 2](#_xxg8pj8oiq42)

[Response](#_3ksff6xchfui)

[Prompt 3](#_bg451li0ztcb)

[Response:](#_49sy5s3phynp)

[Claude sonnet 3.5](#_yed7wwcjhqgd)

[Prompt 1:](#_lhn9iiey5bky)

[Response:](#_juievwyofoc0)

[Prompt 2:](#_dawdoe6szo45)

[Response:](#_yllscoxdbruz)

[Prompt 3:](#_qaa108u1z0v)

[Response:](#_7oka7vqf2443)

[Experiment with diabetes dataset:](#_bzi0yfwem7ye)

[Gemini 2.5 pro](#_odhlwmi5d0y4)

[Prompt 1:](#_7m2ym3kxjxlw)

[Response:](#_yor6odoi8vel)

[Prompt 2:](#_82vfi2zdyjpa)

[Response:](#_p7dqiep9ki0)

[Prompt 3:](#_5ij09behabwv)

[Response:](#_qboxm02tgfk)

[Claude 3.5 sonnet](#_iwvck99sl9x9)

[Prompt 1:](#_urejrwb4jb02)

[Response:](#_t75m3gxkwasj)

[Prompt 2:](#_clx0ql8h1pfr)

[Response:](#_g33ave3pzfyx)

[Prompt 3:](#_z9yd401s9ghb)

[Response:](#_m0x24hyu54ok)

GPT-4.1

[Prompt 1](#_xhls8q8gfz5l)

[Response](#_9vgx35xhr2hp)

[Prompt 2](#_mf9et4mp5hax)

[Response](#_c1v6h94c8k8i)

[Prompt 3](#_3r0h5szfi2bm)

[Response](#_3ef9hmdyw9vw)

**Prompt 1:**Could there be fairness concerns related to uneven group representation? Identify these disparities, and what strategies would you recommend to rebalance the dataset to support fair model training?

**Prompt 2:**Are there signs of uneven prediction outcomes across groups that could raise fairness concerns? What metrics would you use to assess potential bias, and how would you interpret the results to determine whether mitigation is necessary?

**Prompt 3:**Some groups in this dataset appear underrepresented and may receive disproportionately skewed predictions. How would you evaluate the severity of this imbalance and choose appropriate rebalancing or preprocessing techniques?

## Experiment with the comorbidities dataset:

### **For GPT-4.1**

#### **Prompt 1:**

In the dataset data\_new.csv. Could there be fairness concerns related to uneven group representation? Identify these disparities, and what strategies would you recommend to rebalance the dataset to support fair model training?

#### Response:

The dataset contains 48,784 rows and 160 columns. To assess fairness concerns related to uneven group representation, I’ll:

1. **Identify the main group columns (e.g., gender, race, age)** that typically relate to fairness in healthcare data.
2. **Check the distribution of these groups** to see if any group is underrepresented.

Let’s start by checking which columns represent protected groups (such as race, gender, sex, age), and then examine their distributions.

Let me list unique values and counts for the relevant group columns.

Group Representation in the Dataset

1. Race

* **White:** 43,202 (88.5%)
* **Black:** 5,582 (11.5%)

2. Age Bands (One-Hot Encoded)

* **18–24:** 1,799 (3.7%)
* **25–34:** 5,380 (11%)
* **35–44:** 9,481 (19.4%)
* **45–54:** 11,665 (23.9%)
* **55–64:** 9,590 (19.6%)
* **65–74:** 6,940 (14.2%)
* **75+:** 3,429 (7%)

*(Note: Counts are for rows with 1 in each age band; these do not sum to the total because each person falls into one band.)*

Fairness Concerns & Disparities

* **Race:** The vast majority of data points are from White patients (nearly 89%), with Black patients underrepresented (about 11%).  
   This imbalance could lead to models that are more accurate for the majority group and potentially biased against the minority group.
* **Age:** Some age bands (like 18–24 and 75+) are much less represented compared to middle-age groups (35–64).  
   This may reduce model performance and fairness for younger and older patients.

Recommended Strategies to Rebalance the Dataset

A. Sampling Methods

* **Undersampling** the majority group (e.g., randomly sample fewer White patients or middle-aged adults) to balance groups.
* **Oversampling** the minority groups (e.g., replicate or synthetically generate more data for Black patients or underrepresented age bands) using techniques like SMOTE for tabular data.

B. Reweighting

* **Assign higher weights to minority group samples** during model training. This makes the model pay more attention to these samples without physically duplicating data.

C. Data Augmentation

* For structured data, **create synthetic samples** of minority groups using generative methods or statistical bootstrapping, ensuring these are realistic.

D. Fairness-Aware Training

* Use **fairness constraints or regularization** (such as equal opportunity loss, demographic parity constraints) during training.
* **Monitor group-specific metrics** (e.g., accuracy, precision, recall for each group) to ensure balanced performance.

#### **Prompt 2:**

Are there signs of uneven prediction outcomes across groups that could raise fairness concerns? What metrics would you use to assess potential bias, and how would you interpret the results to determine whether mitigation is necessary?

#### **Response:**

Great, let’s break this down step by step:

1. Signs of Uneven Prediction Outcomes Across Groups

To detect **fairness concerns in prediction outcomes**, you need access to:

* **Model predictions** (e.g., y\_pred or model\_output)
* **True outcomes** (e.g., target, label, or gagne\_sum\_t)
* **Group labels** (e.g., race, age band columns)

**Uneven prediction outcomes** often show up as:

* One group getting more positive predictions (or more negative ones) than others
* Different error rates (false positives/negatives) across groups
* Substantially higher/lower accuracy or other performance metrics for one group

2. Metrics to Assess Potential Bias

A. Group Outcome Rates

* **Positive Prediction Rate**: Percentage of samples in each group predicted as positive.  
  + Large differences can indicate bias.
  + *Metric: Demographic Parity (Statistical Parity Difference)*

B. True Outcome Rates

* **True Positive Rate (TPR)/Recall**: Of all real positives, how many were predicted as positive (for each group)?  
  + *Metric: Equal Opportunity Difference*
* **False Positive Rate (FPR)**: Of all real negatives, how many were incorrectly predicted as positive?  
  + *Metric: Average Odds Difference*

C. Disparate Impact Ratio

* Ratio of positive prediction rate of minority group to majority group (e.g., Black/White).  
  + If < 0.8 or > 1.25, it can raise regulatory concerns.

D. Calibration by Group

* Does the model’s predicted probability mean the same thing for each group? Poor calibration for some groups can signal unfairness.

3. How to Interpret Results and Decide on Mitigation

**Suppose you run the above metrics and find:**

* Positive prediction rates for White = 22%, Black = 12%
* TPR for White = 70%, Black = 55%
* Disparate Impact Ratio = 0.55

**Interpretation:**

* **The model is much less likely to predict positive for Black patients, even if true outcomes are balanced**—this indicates *demographic disparity*.
* **Black patients have a lower TPR (model misses more true cases)**—this is *equal opportunity bias*.
* **Disparate Impact Ratio is far below the 0.8 rule**, signaling a major fairness issue.

Mitigation Would Be Necessary If:

* Disparities in these metrics are large (beyond established fairness thresholds, e.g., SPD > 0.1, DIR < 0.8).
* The model performs poorly for underrepresented or vulnerable groups, possibly increasing real-world harm.

Summary Table: Bias Metrics & Thresholds

| **Metric** | **What it checks** | **Fairness Threshold** |
| --- | --- | --- |
| Statistical Parity Difference | ∆ positive rates between groups |  |
| Equal Opportunity Difference | ∆ TPR between groups |  |
| Disparate Impact Ratio | Ratio of positive rates | 0.8 ≤ ratio ≤ 1.25 |
| Calibration Difference | ∆ Brier score, AUC per group | No large gaps |

Next Step:

If you provide a column of model predictions (or want to see a simulated example), I can **compute these bias metrics directly** for your dataset, and plot or tabulate the results for interpretation.

Would you like to proceed with an actual calculation using your data? If yes, please tell me which column is the model’s prediction and which is the target, or upload the predictions if you have them!

#### Prompt 3:

Some groups in this dataset appear underrepresented and may receive disproportionately skewed predictions. How would you evaluate the severity of this imbalance and choose appropriate rebalancing or preprocessing techniques?

#### Response:

This is a very important and nuanced question in fairness-aware machine learning. Here’s a structured approach for **evaluating the severity of group imbalance** and **choosing the right rebalancing or preprocessing technique**:

1. Evaluate the Severity of Group Imbalance

A. Quantitative Group Size Analysis

* **Compute the proportion of each group** (e.g., for race, age\_band) in the dataset.  
  + E.g., White: 89%, Black: 11%.
* Calculate Imbalance Ratios:  
  + The ratio between the size of the majority and minority groups.
  + Severe imbalance: one group < 20% of another, or <10% of overall data.

B. Statistical Parity of Features

* **Visualize with bar plots or pie charts** to show distribution of samples by group.
* **Gini Index/Entropy**: Statistical measures for categorical imbalance.

C. Impact on Model Predictions

* **Simulate baseline (unbalanced) model predictions**: see if error rates or positive prediction rates are much worse for the smaller group.
* If the **minority group has both small size and much higher error rates**, the imbalance is especially severe.

2. Choosing the Right Rebalancing or Preprocessing Technique

A. If Imbalance is Moderate

* **Reweighting**: Assign higher weights to samples from underrepresented groups in model training.  
  + Minimal information loss.
  + Supported by most modern ML libraries.
* **Class-balanced loss functions** (for multi-class targets).

B. If Imbalance is Severe

* **Oversampling** the minority group:  
  + **Random Oversampling**: Duplicate minority group samples.
  + **Synthetic Oversampling**: Use SMOTE or similar techniques to generate synthetic samples for underrepresented groups.  
    - Works well for tabular data, but check for feature realism.
* **Undersampling** the majority group:  
  + Reduce size of overrepresented group (risk: lose potentially useful data).

C. Preprocessing for Fairness

**Stratified Sampling**: When splitting train/test, preserve group proportions.

* **Group-aware Feature Scaling**: Standardize features within each group.
* **Fairness Constraints During Training**: Algorithms that incorporate fairness constraints, e.g., fairlearn, AIF360.

3. Concrete Example from Your Dataset

Suppose:

* **Race**: White (43,202), Black (5,582)  
  + Imbalance ratio: 43,202 / 5,582 ≈ 7.74 (White samples outnumber Black samples by almost 8:1).
* **Age bands**: Youngest (3.7%), oldest (7%)—both far lower than central bands.

How to Evaluate Severity:

* Ratio above 4:1 is generally considered severe in fairness literature.
* Check if minority group outcome rates (predicted or true) are much worse.
* Visualize both **input imbalance** and **prediction outcome disparity**.

4. Recommendations for Your Dataset

| **Imbalance Level** | **Severity** | **Preferred Technique(s)** | **Notes** |
| --- | --- | --- | --- |
| Moderate | 2:1–4:1 | Reweighting, class-balanced loss | Minimal risk of overfitting |
| Severe | >4:1 | Synthetic oversampling (e.g., SMOTE), or combine with undersampling | Check for model overfitting, data realism |
| Very Severe | <10% of data | Targeted data collection (if possible), strong oversampling, or reject using certain models | May require reporting limits in results |

How to Implement (in code, for example):

* **Reweighting:** sample\_weight = total\_samples / (n\_groups \* group\_count[group\_label])

**SMOTE (Python):** from imblearn.over\_sampling import SMOTENC

X\_resampled, y\_resampled = SMOTENC(categorical\_features=[group\_index], random\_state=0).fit\_resample(X, y)

Summary Table

| **Step** | **What to Check** | **What to Do** |
| --- | --- | --- |
| Quantify imbalance | Group counts, ratios, % | If ratio >4:1, flag as severe |
| Model performance by group | Prediction rates, error rates | Large disparities indicate bias risk |
| Choose rebalancing | Imbalance + outcome disparity | Use reweighting (mod), oversampling (sev) |

**If you’d like**, I can generate group imbalance plots and recommend the exact Python code for your case—just tell me which group to focus on (race, age band, or both)!

### Gemini 2.5 pro

#### Prompt 1

Could there be fairness concerns related to uneven group representation? Identify these disparities, and what strategies would you recommend to rebalance the dataset to support fair model training?

#### Response

Okay, let's investigate `data\_new.csv` for potential fairness concerns arising from uneven group representation and discuss strategies for mitigation.

First, I need to understand the structure of this dataset. I'll look at its columns and the first few rows to identify potential sensitive attributes and the likely target variable.

```bash

python -c "import pandas as pd; df = pd.read\_csv('data\_new.csv', nrows=5); print(df.columns.tolist()); print(df.head())"

```

Okay, this `data\_new.csv` dataset is quite extensive with 160 columns!

Based on the column names, the primary attributes for assessing group representation fairness are:

\* `race`

\* `dem\_female` (likely representing gender, where 1 might be female and 0 male, or vice-versa)

\* The set of columns `dem\_age\_band\_...\_tm1` (representing different age groups from a previous time period, `tm1`)

Let's analyze the distribution of these. I'll start with `race`.

```bash

python -c "import pandas as pd; df = pd.read\_csv('data\_new.csv'); print('Race Distribution (Counts):\n', df['race'].value\_counts(dropna=False)); print('\nRace Distribution (Percentage):\n', df['race'].value\_counts(dropna=False, normalize=True) \* 100)"

```

\*\*Race Distribution:\*\*

\* \*\*Counts:\*\*

\* white: 43202

\* black: 5582

\* \*\*Percentage:\*\*

\* white: 88.56%

\* black: 11.44%

\* \*\*Disparity (Race):\*\* \*\*High.\*\* The dataset is predominantly composed of individuals classified as "white". Individuals classified as "black" are a significant minority but are outnumbered roughly 7.7 to 1. Other racial categories are not present, which itself is a form of representation bias if the population this data is drawn from is more diverse.

\* \*\*Concern:\*\* Models trained on this data are likely to be more accurate and reliable for "white" individuals. Predictions for "black" individuals might be less accurate or systematically skewed due to their underrepresentation.

Next, let's analyze `dem\_female`.

```bash

python -c "import pandas as pd; df = pd.read\_csv('data\_new.csv'); print('Gender Distribution (dem\_female) (Counts):\n', df['dem\_female'].value\_counts(dropna=False)); print('\nGender Distribution (dem\_female) (Percentage):\n', df['dem\_female'].value\_counts(dropna=False, normalize=True) \* 100)"

```

\*\*Gender Distribution (`dem\_female`):\*\*

\* \*\*Counts:\*\*

\* 1 (Likely Female): 30763

\* 0 (Likely Male): 18021

\* \*\*Percentage:\*\*

\* 1: 63.06%

\* 0: 36.94%

\* \*\*Disparity (Gender):\*\* \*\*Moderate.\*\* There's a higher representation of the group coded as '1' (presumably females) compared to '0' (presumably males), with a ratio of about 1.7 to 1.

\* \*\*Concern:\*\* The model might be somewhat more attuned to the characteristics and outcomes of the majority gender group.

Now, for the `dem\_age\_band\_...\_tm1` columns. These are already one-hot encoded. To see the overall distribution, I'll sum up each of these binary columns, as each row should only have one `dem\_age\_band\_...\_tm1` column marked as 1.

```bash

python -c "import pandas as pd; df = pd.read\_csv('data\_new.csv'); age\_cols = [col for col in df.columns if 'dem\_age\_band\_' in col]; age\_distribution = df[age\_cols].sum().sort\_values(ascending=False); print('Age Band Distribution (Counts):\n', age\_distribution); print('\nAge Band Distribution (Percentage):\n', (age\_distribution / df.shape[0]) \* 100)"

```

Okay, this `data\_new.csv` dataset is quite extensive with 160 columns!

Based on the column names and our previous analysis, the primary attributes for assessing group representation fairness are:

\* `race`

\* `dem\_female` (gender)

\* The `dem\_age\_band\_...\_tm1` columns (age groups)

We've already determined the following distributions and potential disparities:

\*\*1. Identified Disparities and Fairness Concerns (Recap from previous steps):\*\*

\* \*\*`race`:\*\*

\* white: 43202 (88.56%)

\* black: 5582 (11.44%)

\* \*\*Severity:\*\* High. "black" individuals are significantly underrepresented. Other racial groups are absent.

\* \*\*Fairness Concern:\*\* Models will likely perform better for "white" individuals. Predictions for "black" individuals may be less reliable or systematically biased. The absence of other races limits the model's applicability to a more diverse population.

\* \*\*`dem\_female` (Gender):\*\*

\* 1 (Likely Female): 30763 (63.06%)

\* 0 (Likely Male): 18021 (36.94%)

\* \*\*Severity:\*\* Moderate. Females are more represented.

\* \*\*Fairness Concern:\*\* The model might be slightly more tuned to the majority gender.

\* \*\*`dem\_age\_band\_...\_tm1` (Age Bands):\*\*

\* dem\_age\_band\_45-54\_tm1: 11665 (23.91%)

\* dem\_age\_band\_55-64\_tm1: 9590 (19.66%)

\* dem\_age\_band\_35-44\_tm1: 9481 (19.43%)

\* dem\_age\_band\_65-74\_tm1: 6940 (14.23%)

\* dem\_age\_band\_25-34\_tm1: 5380 (11.03%)

\* dem\_age\_band\_75+\_tm1: 3429 (7.03%)

\* dem\_age\_band\_18-24\_tm1: 1799 (3.69%)

\* \*\*Severity:\*\* Moderate to High for the tails. The 45-54 band is the largest. The youngest (18-24) and oldest (75+) bands are the smallest and significantly underrepresented compared to middle age bands.

\* \*\*Fairness Concern:\*\* Predictions for the youngest and oldest age groups might be less reliable due to fewer samples. The model will be most influenced by the middle-aged groups.

\*\*2. Choosing Appropriate Rebalancing or Preprocessing Techniques:\*\*

The key is to ensure the model doesn't learn spurious correlations or perform poorly for underrepresented groups. The choice of technique depends heavily on the \*\*target variable\*\* you are trying to predict (e.g., `risk\_score\_t`, `program\_enrolled\_t`, `cost\_t`).

\*\*General Principles:\*\*

1. \*\*Identify the Target Variable (`y`):\*\* Strategies depend on what `y` is and whether `y` itself is imbalanced.

2. \*\*Train-Test Split:\*\* \*\*Absolutely critical.\*\* Split data \*before\* rebalancing or complex imputation. Rebalance \*only\* the training set. Test on the original distribution. Use stratification by `y` if `y` is categorical and imbalanced.

3. \*\*Preprocessing:\*\*

\* \*\*Encoding:\*\* `race` needs encoding (e.g., One-Hot if it had more categories, or map to 0/1). `dem\_female` is binary. `dem\_age\_band\_...` are one-hot.

\* \*\*Scaling:\*\* Scale all numerical features (many in this dataset: costs, lab values, etc.).

\* \*\*Missing Values:\*\* Thoroughly check all 160 columns for missing values and impute appropriately (median for numerical, mode or a distinct category for categorical), fitting imputers on the training set.

\*\*Strategies Considering the Identified Imbalances:\*\*

\* \*\*If the Target Variable (`y`) is Imbalanced:\*\*

1. \*\*Cost-Sensitive Learning:\*\* This is often the best first approach. Modify the learning algorithm to penalize errors on the minority class of `y` more heavily (e.g., `class\_weight='balanced'` in scikit-learn classifiers).

2. \*\*Oversampling `y`'s Minority Class:\*\* Use techniques like `SMOTE` or `ADASYN` on the training data to generate synthetic samples for the minority class of `y`.

3. \*\*Undersampling `y`'s Majority Class:\*\* Use `RandomUnderSampler`, `TomekLinks`, etc., on the training data. Potentially risky if it discards too much information.

4. \*\*Hybrid Approaches for `y`:\*\* `SMOTETomek`, `SMOTEENN`.

\* \*\*Addressing Unfairness from Imbalanced \*Features\* (`race`, `dem\_age\_band`, `dem\_female`):\*\*

The primary goal isn't to make feature distributions uniform but to ensure the model's predictions for `y` are \*fair\* across these groups.

1. \*\*Re-weighting Samples (Pre-processing/In-processing):\*\*

\* Assign higher weights to instances belonging to underrepresented demographic groups during model training. For example, when predicting `y`, samples where `race` is "black" or `dem\_age\_band\_18-24\_tm1` is 1 would get higher weights.

\* This forces the model to pay more attention to getting predictions right for these smaller groups.

\* The `sample\_weight` parameter in the `fit` method of many scikit-learn models can be used. Weights can be inversely proportional to group size.

2. \*\*Fairness-Aware Algorithms (In-processing):\*\*

\* Utilize algorithms designed to optimize for both accuracy and fairness metrics. Libraries like `Fairlearn` provide such tools, which can try to equalize error rates or prediction rates across sensitive groups.

3. \*\*Post-processing Adjustments:\*\*

\* After training a model, adjust prediction thresholds differently for different demographic groups to improve fairness metrics. This should be done carefully as it directly manipulates outcomes and needs strong justification.

4. \*\*Disentangling Effects:\*\* With many Elixhauser and Romano comorbidity indices, ensure the model isn't unfairly attributing risk to a demographic group when it's actually a co-occurring health condition that is more prevalent in that group (and captured by another variable). Good feature selection and interpretation are key.

\*\*Conceptual Python Example (Illustrating Re-weighting for `race`):\*\*

Let's assume your target `y` is `program\_enrolled\_t` and `X\_train\_processed` is your preprocessed training feature set.

```python

# --- Conceptual: Applying sample weights for 'race' ---

# import numpy as np

# from sklearn.utils.class\_weight import compute\_sample\_weight

# Assume X\_train\_df is your training DataFrame before final processing to numpy

# and it contains the original 'race' column.

# y\_train is your training target Series.

# Calculate weights for race (inverse of proportion)

# race\_counts = X\_train\_df['race'].value\_counts()

# race\_weights\_map = {level: len(X\_train\_df) / count for level, count in race\_counts.items()}

# sample\_weights\_for\_race = X\_train\_df['race'].map(race\_weights\_map).fillna(1.0).to\_numpy()

# If your target y\_train is also imbalanced, you might want to use class\_weight in the model

# OR combine sample\_weights\_for\_race with sample weights for the target.

# For example, if also using compute\_sample\_weight for the target:

# sample\_weights\_for\_target = compute\_sample\_weight(class\_weight='balanced', y=y\_train)

# combined\_sample\_weights = sample\_weights\_for\_race \* sample\_weights\_for\_target # Element-wise product

# # Fit model using these weights

# # model = RandomForestClassifier(random\_state=42) # Or your chosen model

# # model.fit(X\_train\_processed, y\_train, sample\_weight=combined\_sample\_weights) # or sample\_weights\_for\_race

# # ---- Evaluation ----

# # After training, predict on X\_test\_processed (which was not re-weighted or re-sampled)

# # y\_pred = model.predict(X\_test\_processed)

# # Then, evaluate fairness metrics by comparing y\_pred to y\_test,

# # disaggregated by the original 'race', 'dem\_female', 'dem\_age\_band' values in X\_test\_orig.

```

\*\*Specific Recommendations for `data\_new.csv`:\*\*

1. \*\*Define Target `y`:\*\* Crucial. The choice impacts subsequent steps.

2. \*\*Train-Test Split:\*\* Stratify by `y`.

3. \*\*Full Preprocessing:\*\* Handle missing values across all 160 columns, encode categoricals (like `race`), scale numericals. This should be done within a scikit-learn `Pipeline` for robustness, fitting on training data.

4. \*\*Address Target Imbalance (if `y` is imbalanced):\*\*

\* Start with `class\_weight='balanced'` in your classifier.

\* If more is needed, consider SMOTE for `y` on the training data.

5. \*\*Address Feature Imbalance for Fairness (especially `race` and extreme `dem\_age\_band`s):\*\*

\* \*\*Re-weighting:\*\* This is a strong candidate. Calculate `sample\_weight` for the training data based on `race` and `dem\_age\_band` to give more importance to underrepresented groups.

\* \*\*Fairlearn:\*\* Explore tools from `Fairlearn` if initial re-weighting and model tuning don't yield satisfactory fairness.

6. \*\*Modeling:\*\* Choose a model robust to high dimensionality or perform feature selection.

7. \*\*Iterative Evaluation:\*\*

\* Continuously evaluate overall model performance (e.g., AUC, F1-score for `y`).

\* Simultaneously evaluate fairness metrics (Demographic Parity, Equalized Odds, Equal Opportunity) across `race`, `dem\_female`, and `dem\_age\_band` subgroups on the test set. The goal is to find a balance.

Given the high dimensionality and the nature of the features (many health indicators), it's also important to ensure that the model is genuinely learning from relevant medical information and not just picking up on spurious correlations due to demographic imbalances. Feature importance analysis and model interpretability techniques can be helpful here.

#### Prompt 2

For data\_new.csv. Are there signs of uneven prediction outcomes across groups that could raise fairness concerns? What metrics would you use to assess potential bias (also use them on this dataset), and how would you interpret the results to determine whether mitigation is necessary?

#### Response

You're asking a critical question about ensuring fair prediction outcomes for the `data\_new.csv` dataset. As we've established, to fully measure \*prediction\* bias and apply metrics like Equal Opportunity, we'd need a trained model and its predictions.

However, we can (and should) analyze the \*\*actual observed outcome rates\*\* for a plausible target variable across the sensitive demographic groups we've identified (`race`, `dem\_female`, `dem\_age\_band\_...\_tm1`). These "base rates" reveal existing disparities in outcomes within the dataset, which can translate into biased model predictions if not carefully handled.

Let's choose \*\*`program\_enrolled\_t`\*\* as our example target variable for this analysis. It appears to be a binary outcome (0 or 1) from the initial data inspection, suitable for illustrating these concepts.

\*\*1. Distribution of the Target Variable (`program\_enrolled\_t`)\*\*

First, let's see how balanced `program\_enrolled\_t` is.

```bash

python -c "import pandas as pd; df = pd.read\_csv('data\_new.csv'); print('Program Enrolled (program\_enrolled\_t) Distribution (Counts):\n', df['program\_enrolled\_t'].value\_counts(dropna=False)); print('\nProgram Enrolled (program\_enrolled\_t) Distribution (Percentage):\n', df['program\_enrolled\_t'].value\_counts(dropna=False, normalize=True) \* 100)"

```

The target variable `program\_enrolled\_t` is \*\*extremely imbalanced\*\*:

\* 0 (Not Enrolled): 48332 (99.07%)

\* 1 (Enrolled): 452 (0.93%)

This high level of imbalance in the target itself is a major factor that will influence model training and can exacerbate fairness issues.

\*\*2. Analyzing Actual Outcome Rates (`program\_enrolled\_t`) Across Groups\*\*

Now, let's look at the rate of program enrollment for different demographic groups.

\*\*By `race`:\*\*

```bash

python -c "import pandas as pd; df = pd.read\_csv('data\_new.csv'); print('Program Enrollment Rate by Race:\n'); print(pd.crosstab(df['race'], df['program\_enrolled\_t'], normalize='index') \* 100)"

```

Okay, we've started analyzing the actual observed rates of `program\_enrolled\_t` (our example target variable) across different groups in `data\_new.csv`.

\*\*Recap of Initial Findings:\*\*

\* \*\*Target Variable `program\_enrolled\_t` Distribution:\*\*

\* 0 (Not Enrolled): 48332 (99.07%)

\* 1 (Enrolled): 452 (0.93%)

\* This is \*\*extremely imbalanced\*\*.

\* \*\*Program Enrollment Rate by `race` (Actual Outcomes):\*\*

| race | Not Enrolled (0) (%) | Enrolled (1) (%) |

| :---- | :------------------- | :--------------- |

| black | 98.58 | 1.42 |

| white | 99.14 | 0.86 |

\* \*\*Observation:\*\* The enrollment rate for "black" individuals (1.42%) is noticeably higher than for "white" individuals (0.86%) in this dataset. Given the overall low enrollment rate, this is a relative difference of about 65%.

Now, let's continue by analyzing the enrollment rates by `dem\_female`.

```bash

python -c "import pandas as pd; df = pd.read\_csv('data\_new.csv'); print('Program Enrollment Rate by Gender (dem\_female):\n'); print(pd.crosstab(df['dem\_female'], df['program\_enrolled\_t'], normalize='index') \* 100)"

```

\*\*Program Enrollment Rate by Gender (`dem\_female`) (Actual Outcomes):\*\*

| dem\_female (Gender) | Not Enrolled (0) (%) | Enrolled (1) (%) |

| :------------------ | :------------------- | :--------------- |

| 0 (Likely Male) | 98.95 | 1.05 |

| 1 (Likely Female) | 99.15 | 0.85 |

\* \*\*Observation:\*\* The group coded as '0' (likely males) has a slightly higher enrollment rate (1.05%) compared to the group coded as '1' (likely females) (0.85%).

Next, let's analyze by `dem\_age\_band\_...\_tm1`. I'll first create a single 'age\_group' column from the one-hot encoded ones to simplify the `crosstab`.

```bash

python -c "import pandas as pd; df = pd.read\_csv('data\_new.csv'); age\_cols = [col for col in df.columns if 'dem\_age\_band\_' in col]; df['age\_group'] = df[age\_cols].idxmax(axis=1); print('Program Enrollment Rate by Age Group:\n'); print(pd.crosstab(df['age\_group'], df['program\_enrolled\_t'], normalize='index') \* 100)"

```

Okay, we're analyzing the actual observed rates of `program\_enrolled\_t` across different demographic groups in `data\_new.csv`.

\*\*Recap of Actual Outcome Rates for `program\_enrolled\_t`:\*\*

\* \*\*Overall:\*\* 0.93% enrolled.

\* \*\*By `race`:\*\* "black" (1.42%) vs. "white" (0.86%).

\* \*\*By `dem\_female` (Gender):\*\* '0' (Likely Male) (1.05%) vs. '1' (Likely Female) (0.85%).

\*\*Program Enrollment Rate by Age Group (`dem\_age\_band\_...\_tm1`) (Actual Outcomes):\*\*

| age\_group | Not Enrolled (0) (%) | Enrolled (1) (%) |

| :--------------------- | :------------------- | :--------------- |

| dem\_age\_band\_18-24\_tm1 | 99.71 | 0.29 |

| dem\_age\_band\_25-34\_tm1 | 99.77 | 0.23 |

| dem\_age\_band\_35-44\_tm1 | 99.55 | 0.45 |

| dem\_age\_band\_45-54\_tm1 | 99.35 | 0.65 |

| dem\_age\_band\_55-64\_tm1 | 98.92 | 1.08 |

| dem\_age\_band\_65-74\_tm1 | 98.16 | 1.84 |

| dem\_age\_band\_75+\_tm1 | 97.47 | 2.53 |

\* \*\*Observation:\*\* There's a clear trend of increasing enrollment rate with age. The 75+ group has the highest enrollment rate (2.53%), while the younger groups (18-24 and 25-34) have very low rates (around 0.2-0.3%).

\*\*Summary of Potential Fairness Concerns from Uneven Actual Outcomes:\*\*

The analysis of actual enrollment rates for `program\_enrolled\_t` highlights several pre-existing disparities:

1. \*\*Target Imbalance:\*\* The program enrollment itself is a rare event (0.93%). This means any model will be naturally inclined to predict "not enrolled."

2. \*\*Race:\*\* "black" individuals have a noticeably higher base rate of enrollment compared to "white" individuals.

3. \*\*Gender (`dem\_female`):\*\* The group '0' (likely males) has a slightly higher base rate of enrollment.

4. \*\*Age Bands:\*\* Enrollment rates vary significantly by age, increasing with older age bands. The 75+ group has an enrollment rate that is more than 10 times higher than the 25-34 age group.

These differences in \*actual observed outcomes\* are crucial because a machine learning model trained on this data (especially one trying to predict `program\_enrolled\_t`) might:

\* \*\*Struggle with the Rare Event:\*\* Difficulty in accurately predicting enrollment for \*any\* group due to the low overall rate.

\* \*\*Exhibit Differential Performance:\*\*

\* For `race`: The model might be less accurate in predicting enrollment for "black" individuals if their patterns of enrollment differ or if the features driving enrollment are not well captured for this smaller group (despite their higher enrollment rate).

\* For `age`: The model might be better at predicting enrollment for older age groups (where it's more common and there are more positive examples) and perform poorly for younger groups.

\* \*\*Amplify Disparities or Learn Spurious Correlations:\*\* If features correlate differently with enrollment across these demographic groups, the model might inadvertently learn biased patterns. For instance, if a feature is more common in the "black" population and also correlates with higher enrollment, the model might overemphasize that feature for "black" individuals.

\*\*Metrics You Would Use to Assess Potential Prediction Bias (with a Trained Model):\*\*

If you had a trained model predicting `program\_enrolled\_t`, you'd use these fairness metrics, calculated by comparing model predictions to actual enrollment outcomes across the demographic groups (`race`, `dem\_female`, `dem\_age\_band\_...\_tm1`):

1. \*\*Demographic Parity (Statistical Parity):\*\*

\* \*Measures:\* `P(Predicted = Enrolled | Group A) == P(Predicted = Enrolled | Group B)`

\* \*Interpretation:\* Does the model \*predict\* enrollment at similar rates for "white" and "black" individuals, or for different age bands? Given the observed base rate differences, achieving perfect demographic parity might not be desirable if it means ignoring true differences in need or eligibility. However, large, unexplained differences in prediction rates would be a concern.

2. \*\*Equal Opportunity:\*\*

\* \*Measures (for the positive outcome 'Enrolled'):\* `P(Predicted = Enrolled | Actual = Enrolled, Group A) == P(Predicted = Enrolled | Actual = Enrolled, Group B)` (True Positive Rate or Recall for enrollment).

\* \*Interpretation:\* Is the model equally good at identifying individuals who \*actually will enroll\* (or are eligible/targeted for enrollment) across all racial groups or age bands? A lower TPR for a group means more missed opportunities for that group if the program is beneficial.

3. \*\*Equalized Odds:\*\*

\* \*Measures:\* Requires both TPR (as above) AND the True Negative Rate (or False Positive Rate for enrollment) to be equal across groups.

\* `P(Predicted = Not Enrolled | Actual = Not Enrolled, Group A) == P(Predicted = Not Enrolled | Actual = Not Enrolled, Group B)`

\* \*Interpretation:\* Is the model balanced in both correctly identifying those who enroll AND correctly identifying those who do not, across all groups? For program enrollment, the TPR (Equal Opportunity) is often more critical.

4. \*\*Predictive Rate Parity (Precision Parity):\*\*

\* \*Measures:\* `P(Actual = Enrolled | Predicted = Enrolled, Group A) == P(Actual = Enrolled | Predicted = Enrolled, Group B)` (Positive Predictive Value).

\* \*Interpretation:\* When the model predicts someone will enroll, is that prediction equally reliable across different groups? If PPV is much lower for one group, it means more "false alarms" for them.

5. \*\*Accuracy Parity:\*\*

\* \*Measures:\* Overall accuracy equality.

\* \*Interpretation:\* Can be misleading with highly imbalanced targets like this. A model predicting "not enrolled" for everyone would have >99% accuracy but be useless and unfair.

\*\*Focus for This Dataset:\*\*

\* Given the extreme imbalance of `program\_enrolled\_t`, \*\*Equal Opportunity (TPR for enrollment)\*\* is often a key metric. We want to ensure that eligible individuals from all demographic groups have an equal chance of being correctly identified by the model for program enrollment.

\* \*\*False Positive Rate Parity\*\* (`P(Predicted = Enrolled | Actual = Not Enrolled, Group A) == P(Predicted = Enrolled | Actual = Not Enrolled, Group B)`) might also be important if incorrectly predicting someone will enroll has significant costs or consequences.

\*\*How to Interpret Results to Determine if Mitigation is Necessary:\*\*

1. \*\*Calculate Metrics:\*\* On a held-out test set, compute these metrics for each subgroup.

2. \*\*Identify Disparities:\*\* Look for meaningful differences.

\* E.g., is the TPR for enrollment for "black" individuals significantly lower than for "white" individuals, despite their higher base rate of enrollment? This would be a major red flag.

\* Is the model's TPR for the 75+ age group (highest base enrollment) much better than for the 55-64 age group (moderate base enrollment)?

3. \*\*Contextualize with Base Rates:\*\* The interpretation should consider the base rate differences. For instance, if a group has a very low base rate of enrollment, achieving a high TPR for that group might be difficult for any model, but the model should still strive for equitable performance relative to other groups and the inherent difficulty.

4. \*\*Define "Significant" Disparity:\*\*

\* \*\*Statistical Significance:\*\* Are differences robust or just noise?

\* \*\*Practical Impact:\*\* What are the consequences of the disparity? If predicting program enrollment is for outreach for a beneficial program, lower TPR for a group means that group misses out.

\* \*\*Relative Harm:\*\* Even if all TPRs are low due to the rare event, are they \*relatively\* much lower for some groups than others?

5. \*\*Decision for Mitigation:\*\* Mitigation is necessary if:

\* Metrics like Equal Opportunity show that certain demographic groups are being systematically overlooked for predicted enrollment despite similar or higher actual enrollment rates.

\* The model disproportionately makes errors (false positives or false negatives) for certain groups in a way that could lead to harm or inequitable access.

\* For `data\_new.csv`, if a model (even after efforts to handle target imbalance like SMOTE or class weighting) still shows a significantly lower TPR for "black" individuals compared to "white" individuals for `program\_enrolled\_t`, mitigation strategies (like re-weighting based on race, or fairness-aware algorithms) would be strongly indicated. Similarly, if the model fails to identify enrollment in older age groups at a reasonable rate despite their higher base rates, that's a concern.

Given the extreme imbalance of `program\_enrolled\_t` and the observed differences in base enrollment rates across `race` and `age\_band`, it is highly probable that a standard model will struggle and exhibit prediction biases. Careful selection of fairness metrics focused on correctly identifying the rare "enrolled" outcome (like TPR) and diligent evaluation across demographic groups will be essential.

#### Prompt 3

For data\_new.csv. Some groups in this dataset appear underrepresented and may receive disproportionately skewed predictions. How would you evaluate the severity of this imbalance and choose appropriate rebalancing or preprocessing techniques? Also show it

#### Response:

You're right to be concerned about how underrepresentation in `data\_new.csv` can lead to skewed predictions. We've already analyzed the distributions for the target variable `program\_enrolled\_t` and sensitive features like `race`, `dem\_female`, and `dem\_age\_band\_...\_tm1`. Let's consolidate that, specifically evaluate the severity for model training, and then detail how to choose and apply appropriate rebalancing or preprocessing techniques, including conceptual code examples.

\*\*1. Evaluating the Severity of Imbalance (Recap for `data\_new.csv`)\*\*

\* \*\*Target Variable `program\_enrolled\_t`:\*\*

\* 0 (Not Enrolled): 48332 (99.07%)

\* 1 (Enrolled): 452 (0.93%)

\* \*\*Severity:\*\* \*\*Extreme.\*\* This severe class imbalance for the target variable is a primary challenge. Models will heavily favor predicting "Not Enrolled."

\* \*\*Sensitive Feature `race`:\*\*

\* white: 43202 (88.56%)

\* black: 5582 (11.44%)

\* (Other racial categories are absent)

\* \*\*Severity:\*\* High. "black" individuals are significantly underrepresented. The complete absence of other races is a major data limitation.

\* \*\*Risk:\*\* Model performance is likely to be much better for "white" individuals. Predictions for "black" individuals may be less reliable or systematically biased.

\* \*\*Sensitive Feature `dem\_female` (Gender):\*\*

\* 1 (Likely Female): 30763 (63.06%)

\* 0 (Likely Male): 18021 (36.94%)

\* \*\*Severity:\*\* Moderate.

\* \*\*Risk:\*\* Potential for the model to be slightly more optimized for the majority gender.

\* \*\*Sensitive Features `dem\_age\_band\_...\_tm1` (Age Bands):\*\*

\* Largest: dem\_age\_band\_45-54\_tm1 (23.91%)

\* Smallest: dem\_age\_band\_18-24\_tm1 (3.69%), dem\_age\_band\_75+\_tm1 (7.03%)

\* \*\*Severity:\*\* Moderate to High for the youngest and oldest age bands.

\* \*\*Risk:\*\* Model predictions may be less reliable for the significantly underrepresented youngest and oldest age groups.

\*\*2. Choosing Appropriate Rebalancing or Preprocessing Techniques\*\*

The strategy involves a multi-step approach, focusing first on general data preparation, then on handling the extreme target imbalance, and finally on ensuring fairness across demographic features.

\*\*I. General Preprocessing (Order is Important):\*\*

1. \*\*Train-Test Split:\*\*

\* \*\*Action:\*\* Split your data into training and testing sets \*before\* any rebalancing or complex imputation.

\* \*\*Rationale:\*\* Crucial to prevent data leakage. Rebalancing techniques should \*only\* be applied to the training data. The test set should reflect the original data distribution for a realistic evaluation of performance and fairness.

\* \*\*How:\*\* Use `sklearn.model\_selection.train\_test\_split`, and importantly, use `stratify=y` (where `y` is `program\_enrolled\_t`) to ensure both train and test sets have a similar (though still imbalanced) proportion of the rare "enrolled" class.

2. \*\*Handle Missing Values:\*\*

\* \*\*Action:\*\* This dataset has 160 columns. A thorough check for missing values in all columns is needed. For any columns with missing data, apply imputation.

\* \*\*Rationale:\*\* Most models cannot handle NaNs.

\* \*\*How (fit on train, transform train & test):\*\*

\* Numerical features: Median imputation (`SimpleImputer(strategy='median')`) is generally robust.

\* Categorical features: Mode imputation (`SimpleImputer(strategy='most\_frequent')`) or treat NaNs as a distinct category if meaningful.

3. \*\*Encode Categorical Features:\*\*

\* \*\*Action:\*\* Convert categorical features like `race` into a numerical format.

\* \*\*Rationale:\*\* Models require numerical input.

\* \*\*How:\*\* `OneHotEncoder` is suitable, especially if `race` had more categories. For binary `race` as is ("white", "black"), simple mapping (e.g., 0/1) or OneHotEncoding works.

4. \*\*Scale Numerical Features:\*\*

\* \*\*Action:\*\* Scale all numerical features (costs, lab values, `risk\_score\_t`, etc.).

\* \*\*Rationale:\*\* Ensures features with larger magnitudes don't dominate distance-based algorithms or gradient descent.

\* \*\*How:\*\* `StandardScaler` is common. Fit on training data, transform train and test.

\*\*II. Addressing Imbalance in the Target Variable (`program\_enrolled\_t` on Training Data):\*\*

Given `program\_enrolled\_t` is 99% '0' and 1% '1', this is the most pressing imbalance.

1. \*\*Cost-Sensitive Learning:\*\*

\* \*\*How:\*\* Modify the learning algorithm to penalize misclassifying the minority class (Enrolled=1) more heavily. Many scikit-learn classifiers (e.g., `LogisticRegression`, `RandomForestClassifier`, `SVC`, `XGBClassifier`) have a `class\_weight='balanced'` parameter or allow manual weight setting.

\* \*\*Pros:\*\* Does not alter data distribution; directly influences learning. Often a very effective first step.

\* \*\*Cons:\*\* Finding optimal manual weights can require tuning.

2. \*\*Oversampling the Minority Class (Enrolled=1):\*\*

\* \*\*How:\*\* Use techniques on the \*training data\* to increase the number of "Enrolled" instances.

\* `RandomOverSampler`: Duplicates minority instances. Simple, but can lead to overfitting.

\* `SMOTE (Synthetic Minority Over-sampling Technique)`: Creates new synthetic minority instances. Generally preferred.

\* `ADASYN (Adaptive Synthetic Sampling)`: A SMOTE variant focusing on harder-to-learn minority instances.

\* \*\*Pros:\*\* Provides the model with more examples of the rare class.

\* \*\*Cons:\*\* Can introduce noise if not careful; synthetic samples are not real data.

3. \*\*Undersampling the Majority Class (Enrolled=0):\*\*

\* \*\*How:\*\* Reduce the number of "Not Enrolled" instances in the \*training data\*.

\* `RandomUnderSampler`: Randomly removes majority instances. Can discard valuable information.

\* More advanced methods: `TomekLinks` (cleans class boundaries), `NearMiss`.

\* \*\*Pros:\*\* Can speed up training and help with highly skewed data.

\* \*\*Cons:\*\* High risk of information loss, potentially leading to poorer generalization. Less recommended as a first approach for such extreme imbalance unless combined with oversampling.

4. \*\*Hybrid Approaches:\*\*

\* \*\*How:\*\* Combine oversampling and undersampling (e.g., `SMOTETomek`, `SMOTEENN`).

\* \*\*Pros:\*\* Can offer a good balance, leveraging SMOTE's generation and cleaning from undersampling.

\*\*III. Ensuring Fair Predictions Across Sensitive Feature Groups (`race`, `dem\_female`, `dem\_age\_band`):\*\*

This is about fairness in the \*model's output\*, not necessarily making the feature distributions uniform in the input.

1. \*\*Re-weighting Samples (During Training):\*\*

\* \*\*How:\*\* Assign higher weights to training instances from underrepresented demographic groups (e.g., "black" individuals, youngest/oldest age bands). The model's loss function will then penalize errors on these weighted samples more.

\* \*\*Implementation:\*\* Use the `sample\_weight` parameter in the `fit` method of many scikit-learn models. Weights can be inversely proportional to the size of the demographic group.

\* \*\*Pros:\*\* Directly tells the model to pay more attention to specific groups without altering features or target counts.

\* \*\*Cons:\*\* Calculating appropriate weights can be nuanced, especially if balancing against multiple sensitive features simultaneously.

2. \*\*Fairness-Aware Machine Learning Algorithms (In-processing):\*\*

\* \*\*How:\*\* Use algorithms or toolkits designed to incorporate fairness constraints directly into the model training process. Libraries like `Fairlearn` provide algorithms that can optimize for accuracy while trying to satisfy fairness constraints (e.g., equalizing error rates across groups).

\* \*\*Pros:\*\* Integrates fairness into the core learning process.

\* \*\*Cons:\*\* Can be more complex to implement and tune; might involve a trade-off with raw accuracy.

3. \*\*Post-processing Adjustments:\*\*

\* \*\*How:\*\* After a standard model is trained, adjust its prediction thresholds differently for different demographic groups to achieve better fairness metric scores.

\* \*\*Pros:\*\* Can be applied to any black-box model.

\* \*\*Cons:\*\* Directly manipulates outcomes; needs strong ethical justification and careful validation. Can be seen as "fairness through unawareness reversal."

4. \*\*Careful Feature Engineering and Selection:\*\*

\* \*\*How:\*\* Ensure that features are not acting as inappropriate proxies for sensitive attributes. Understand feature importance.

\* \*\*Pros:\*\* Can lead to more robust and inherently fairer models.

\*\*3. Showing Conceptual Examples (Python with `sklearn` and `imblearn`)\*\*

```python

# --- Conceptual Code Snippets ---

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.impute import SimpleImputer

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.linear\_model import LogisticRegression # Example model

from collections import Counter

# For imblearn (if not installed: pip install imbalanced-learn)

from imblearn.over\_sampling import SMOTE

from imblearn.pipeline import Pipeline as ImbPipeline # Pipeline that can include samplers

# --- Assume initial data loading ---

# df = pd.read\_csv('data\_new.csv')

# y = df['program\_enrolled\_t']

# X = df.drop('program\_enrolled\_t', axis=1)

# # --- 1. Train-Test Split (Stratified) ---

# X\_train\_orig, X\_test\_orig, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42, stratify=y)

#

# # Create copies for processing

# X\_train = X\_train\_orig.copy()

# X\_test = X\_test\_orig.copy()

# --- 2. Preprocessing (Illustrative - adapt to your 160 columns) ---

# Identify numerical and categorical columns (example subset)

# numerical\_cols = X\_train.select\_dtypes(include=np.number).columns.tolist()

# categorical\_cols = ['race'] # Add other categoricals

# Create preprocessing pipelines for numerical and categorical data

# numerical\_pipeline = Pipeline([

# ('imputer', SimpleImputer(strategy='median')),

# ('scaler', StandardScaler())

# ])

#

# categorical\_pipeline = Pipeline([

# ('imputer', SimpleImputer(strategy='most\_frequent')), # or 'constant', fill\_value='Missing'

# ('onehot', OneHotEncoder(handle\_unknown='ignore', sparse\_output=False))

# ])

# Combine preprocessing pipelines using ColumnTransformer

# preprocessor = ColumnTransformer(

# transformers=[

# ('num', numerical\_pipeline, numerical\_cols),

# ('cat', categorical\_pipeline, categorical\_cols)

# ],

# remainder='passthrough' # Or 'drop' if unlisted columns are not needed

# )

# --- 3. Model Training with Strategies ---

# Strategy A: Cost-Sensitive Learning for Target Imbalance

# model\_cost\_sensitive = LogisticRegression(solver='liblinear', class\_weight='balanced', random\_state=42)

# pipeline\_cost\_sensitive = Pipeline([

# ('preprocessor', preprocessor),

# ('classifier', model\_cost\_sensitive)

# ])

# # pipeline\_cost\_sensitive.fit(X\_train, y\_train)

# print(f"Training with Cost-Sensitive Learning for y. Original y\_train counts: {Counter(y\_train)}")

# Strategy B: SMOTE for Target Imbalance (using imblearn's Pipeline)

# model\_smote = LogisticRegression(solver='liblinear', random\_state=42) # No class\_weight here

# pipeline\_smote = ImbPipeline([

# ('preprocessor', preprocessor),

# ('sampler', SMOTE(random\_state=42, sampling\_strategy='auto')), # Default is to oversample minority

# ('classifier', model\_smote)

# ])

# # pipeline\_smote.fit(X\_train, y\_train)

# # Access post-SMOTE counts conceptually (sampler is part of pipeline)

# # X\_train\_processed\_temp = preprocessor.fit\_transform(X\_train)

# # y\_train\_smote\_temp = SMOTE(random\_state=42).fit\_resample(X\_train\_processed\_temp, y\_train)[1]

# # print(f"Training with SMOTE for y. y\_train counts after SMOTE (conceptual): {Counter(y\_train\_smote\_temp)}")

# Strategy C: Sample Re-weighting for Feature Imbalance (e.g., 'race')

# This is applied via sample\_weight in model.fit()

# model\_reweighted = LogisticRegression(solver='liblinear', random\_state=42) # Potentially also class\_weight='balanced' for target

# Fit preprocessor first

# X\_train\_processed = preprocessor.fit\_transform(X\_train)

# X\_test\_processed = preprocessor.transform(X\_test) # Transform test set too

# Create sample weights (conceptual - based on original X\_train before processing if needed for feature value)

# Assuming 'race' column is still accessible or mapped from preprocessed data

# race\_weights\_map = {'white': 1.0, 'black': len(X\_train) / X\_train['race'].value\_counts()['black']} # Example weights

# sample\_weights = X\_train['race'].map(race\_weights\_map).fillna(1.0).to\_numpy()

#

# # If using class\_weight='balanced' with sample\_weight, the effects multiply.

# # model\_reweighted.fit(X\_train\_processed, y\_train, sample\_weight=sample\_weights)

# print(f"Training with sample re-weighting for 'race'. (Example weights: {race\_weights\_map})")

# --- 4. Evaluation (on X\_test\_processed, y\_test) ---

# For each trained pipeline/model:

# y\_pred = pipeline\_X.predict(X\_test)

# y\_proba = pipeline\_X.predict\_proba(X\_test)[:, 1]

#

# Calculate overall performance (AUC, F1 for 'Enrolled', etc.)

# Calculate fairness metrics (Demographic Parity, Equal Opportunity, etc.)

# by comparing y\_pred/y\_proba to y\_test, disaggregated by columns in X\_test\_orig

# (e.g., X\_test\_orig['race'], X\_test\_orig['dem\_female'], etc.)

```

\*\*Specific Recommendations for `data\_new.csv` & Iterative Process:\*\*

1. \*\*Robust Preprocessing:\*\* With 160 columns, a `ColumnTransformer` and `Pipeline` are essential for managing preprocessing steps consistently. Pay close attention to identifying all true numerical and categorical columns.

2. \*\*Tackle Target Imbalance First:\*\* Given `program\_enrolled\_t` is 0.93% positive, using `class\_weight='balanced'` in your classifier is a good starting point. If this doesn't yield satisfactory recall for the positive class, then explore SMOTE.

3. \*\*Address Feature Imbalance for Fairness:\*\*

\* After initial model training (with target balancing), \*\*evaluate fairness\*\* across `race`, `dem\_female`, and `dem\_age\_band`.

\* If significant disparities in fairness metrics (like Equal Opportunity for program enrollment) are found, then implement \*\*sample re-weighting\*\* for the training data based on the underperforming demographic groups.

\* For `race`, since "black" is 11.44%, re-weighting these samples higher can force the model to learn better from them.

\* For `dem\_age\_band\_18-24\_tm1` (3.69%) and `dem\_age\_band\_75+\_tm1` (7.03%), re-weighting could also be beneficial.

4. \*\*Fairlearn:\*\* If re-weighting and model tuning are insufficient, explore algorithms from the `Fairlearn` library which explicitly try to optimize for fairness constraints.

5. \*\*Iterate:\*\* The process is iterative. Train -> Evaluate Performance & Fairness -> Adjust Preprocessing/Rebalancing/Model -> Repeat. The goal is a model that is not only accurate but also doesn't disproportionately harm or disadvantage any particular demographic group.

Be mindful that with such a wide dataset, feature selection or dimensionality reduction techniques might also play a role after initial fairness considerations to improve model performance and interpretability.

### 

### Claude sonnet 3.5

#### Prompt 1:

For data\_new.csv. Could there be fairness concerns related to uneven group representation? Identify these disparities, and what strategies would you recommend to rebalance the dataset to support fair model training?

#### Response:

Based on our analysis, there are significant fairness concerns related to uneven group representation in the dataset. Let me break down the disparities and recommend specific strategies to address them.

\*\*1. Identified Disparities:\*\*

A. \*\*Demographic Representation Imbalance:\*\*

- White Female: 27,077 (55.5%) - Overrepresented

- White Male: 16,125 (33.1%) - Moderately represented

- Black Female: 3,686 (7.6%) - Underrepresented

- Black Male: 1,896 (3.9%) - Severely underrepresented

The imbalance ratio of 14.28 between the largest and smallest groups indicates a severe representation disparity.

B. \*\*Risk Score Distribution Disparities:\*\*

- Black Male: 5.56 (highest risk scores)

- Black Female: 5.28

- White Male: 4.39

- White Female: 4.19 (lowest risk scores)

This shows a concerning pattern where minority groups receive systematically higher risk scores.

C. \*\*Program Enrollment Rate Disparities:\*\*

- Black Male: 1.48% (highest enrollment)

- Black Female: 1.38%

- White Male: 1.00%

- White Female: 0.78% (lowest enrollment)

Let's implement a more sophisticated rebalancing strategy that addresses both demographic and outcome disparities:

Let's run this enhanced fair rebalancing analysis:

```bash

python fair\_rebalancing.py

```

Based on our enhanced analysis, let me provide comprehensive recommendations for addressing the fairness concerns and rebalancing strategies:

\*\*1. Identified Fairness Concerns:\*\*

A. \*\*Demographic Representation Disparities:\*\*

- Severe underrepresentation of Black males (3.9%)

- Significant underrepresentation of Black females (7.6%)

- Overrepresentation of White females (55.5%)

B. \*\*Risk Score Disparities:\*\*

- Clear racial gradient in risk scores:

\* Black males: 5.56 ± 9.26 (highest variance)

\* Black females: 5.28 ± 7.24

\* White males: 4.39 ± 5.52

\* White females: 4.19 ± 4.83 (lowest variance)

C. \*\*Enrollment Rate Disparities:\*\*

- Systematic differences in program enrollment:

\* Black males: 1.5% (highest)

\* Black females: 1.4%

\* White males: 1.0%

\* White females: 0.8% (lowest)

\*\*2. Recommended Rebalancing Strategies:\*\*

A. \*\*Immediate Technical Solutions:\*\*

1. \*\*Two-Stage Rebalancing Approach:\*\*

```python

# Stage 1: Demographic balancing

demographic\_weights = max\_group\_size / group\_size

# Stage 2: Outcome balancing within groups

outcome\_weights = 1 / (positive\_samples / group\_size)

# Combined weights

final\_weights = demographic\_weights \* outcome\_weights

```

2. \*\*Stratified Sampling Implementation:\*\*

```python

# Stratification by both race and gender

stratify\_columns = ['race', 'dem\_female']

train\_test\_split(stratify=df[stratify\_columns])

```

B. \*\*Data Collection and Preprocessing:\*\*

1. \*\*Targeted Data Collection:\*\*

- Priority 1: Increase Black male representation (target: minimum 15% of dataset)

- Priority 2: Increase Black female representation (target: minimum 20% of dataset)

- Method: Active learning to identify most informative new samples

2. \*\*Feature Engineering:\*\*

```python

# Add interaction terms to capture intersectional effects

df['race\_gender\_interaction'] = df['race'] + '\_' + df['dem\_female']

```

3. \*\*Validation Strategy:\*\*

```python

# K-fold cross-validation with multiple stratification

from sklearn.model\_selection import StratifiedKFold

skf = StratifiedKFold(n\_splits=5, shuffle=True)

for train\_idx, val\_idx in skf.split(X, df[stratify\_columns]):

# Ensure balanced representation in each fold

```

\*\*3. Model Training Recommendations:\*\*

A. \*\*Fairness-Aware Training:\*\*

1. \*\*Sample Weights:\*\*

```python

# Apply calculated weights during model training

model.fit(X, y, sample\_weight=weights)

```

2. \*\*Custom Loss Function:\*\*

```python

def fair\_loss(y\_true, y\_pred, group\_membership):

# Base loss

base\_loss = binary\_crossentropy(y\_true, y\_pred)

# Fairness penalty

fairness\_penalty = calculate\_group\_disparity(y\_true, y\_pred, group\_membership)

return base\_loss + lambda\_fairness \* fairness\_penalty

```

B. \*\*Monitoring and Validation:\*\*

1. \*\*Group-Specific Metrics:\*\*

```python

for group in groups:

group\_metrics = evaluate\_model(model, X[group\_mask], y[group\_mask])

monitor\_group\_performance(group, group\_metrics)

```

2. \*\*Fairness Constraints:\*\*

```python

# Implement demographic parity constraint

demographic\_parity\_diff = abs(group1\_pred\_rate - group2\_pred\_rate)

assert demographic\_parity\_diff <= threshold

```

\*\*4. Implementation Plan:\*\*

1. \*\*Short-term Actions:\*\*

- Implement the two-stage rebalancing approach

- Apply sample weights in current model training

- Set up group-specific performance monitoring

2. \*\*Medium-term Actions:\*\*

- Develop fairness-aware loss functions

- Implement cross-validation with multiple stratification

- Create automated fairness monitoring dashboard

3. \*\*Long-term Actions:\*\*

- Establish data collection protocols for underrepresented groups

- Develop fairness-aware feature engineering pipeline

- Create feedback loop for continuous fairness improvement

#### Prompt 2:

for data\_new.csv. Are there signs of uneven prediction outcomes across groups that could raise fairness concerns? What metrics would you use to assess potential bias (also use them on this dataset), and how would you interpret the results to determine whether mitigation is necessary?

#### 

#### Response:

#### 

Let me enhance our fairness analysis by adding more sophisticated metrics and intersectional analysis. I'll modify our script to include additional important fairness metrics and demographic intersections.

Now let's run the enhanced analysis that includes intersectional fairness metrics across both race and gender:

```bash

python fairness\_metrics\_analysis.py

```

Based on the enhanced intersectional analysis, I can provide a more comprehensive assessment of fairness concerns in the dataset. Let me break down the findings by key fairness metrics and intersectional groups:

1. \*\*Group Representation and Base Rates\*\*:

- White Female (white\_1): 27,077 individuals (55.5%)

- White Male (white\_0): 16,125 individuals (33.1%)

- Black Female (black\_1): 3,686 individuals (7.6%)

- Black Male (black\_0): 1,896 individuals (3.9%)

2. \*\*Key Fairness Metrics\*\*:

a) \*\*Demographic Parity Differences\*\* (difference in prediction rates):

- Black Female: +4.7% (SIGNIFICANT DISPARITY)

- Black Male: -1.0%

- White Male: +0.9%

- Reference (White Female): 49.3%

b) \*\*Equal Opportunity Differences\*\* (difference in true positive rates):

- Black Female: +2.4%

- Black Male: -4.8%

- White Male: +1.8%

- Reference (White Female): 97.6%

c) \*\*False Positive Rate Differences\*\*:

- Black Female: +4.4%

- Black Male: -1.3%

- White Male: +0.8%

- Reference (White Female): 48.9%

d) \*\*Positive Predictive Value Differences\*\*:

- Black Female: +1.0%

- Black Male: +1.3%

- White Male: +0.4%

- Reference (White Female): 1.5%

3. \*\*Intersectional Findings\*\*:

\*\*Most Significant Disparities\*\*:

1. \*\*Black Female Group\*\*:

- Highest prediction rate (54.0%)

- Highest false positive rate (53.3%)

- Perfect true positive rate (100%)

- Shows significant disparity in demographic parity (+4.7%)

2. \*\*Black Male Group\*\*:

- Lowest prediction rate (48.3%)

- Highest false negative rate (7.1%)

- Lowest true positive rate (92.9%)

- Shows notable disparity in equal opportunity (-4.8%)

3. \*\*Base Rate Differences\*\*:

- Black individuals have slightly higher actual rates (1.4-1.5%)

- White individuals have lower actual rates (0.8-1.0%)

\*\*Interpretation and Recommendations\*\*:

1. \*\*Primary Concerns\*\*:

a) \*\*Intersectional Bias\*\*:

- The most significant disparities appear at intersections of race and gender

- Black females are over-predicted for high risk (+4.7% prediction rate)

- Black males are under-served (highest false negative rate)

b) \*\*Gender-Race Interaction\*\*:

- The impact of race differs by gender

- Black females experience over-prediction

- Black males experience under-prediction

- This suggests a complex interaction between race and gender that requires careful attention

2. \*\*Recommended Mitigation Strategies\*\*:

a) \*\*Model Adjustments\*\*:

- Implement post-processing techniques to equalize false positive rates across intersectional groups

- Consider separate thresholds for different intersectional groups to balance error rates

- Add fairness constraints during model training that account for intersectional groups

b) \*\*Data Collection\*\*:

- Address the significant imbalance in group sizes

- Collect more data from minority groups, especially Black males

- Investigate potential sampling bias in the current data collection process

c) \*\*Process Improvements\*\*:

- Implement regular monitoring of intersectional fairness metrics

- Develop specific intervention protocols for different demographic groups

- Create guidelines for acceptable disparities that consider intersectional effects

3. \*\*Monitoring Framework\*\*:

- Track intersectional metrics over time

- Set alert thresholds for disparities exceeding 5%

- Regular audit of prediction patterns across all demographic intersections

The analysis reveals that fairness concerns are more complex than initially apparent, with significant intersectional effects. While some metrics fall within acceptable ranges when looking at single demographics, the intersectional analysis reveals important disparities that require attention. The most pressing issues are:

1. The over-prediction for Black females (+4.7% prediction rate)

2. The under-prediction and higher false negative rate for Black males

3. The significant imbalance in group sizes, which may affect model performance

These findings suggest that mitigation strategies should be implemented, with particular attention to addressing intersectional biases and improving model performance for minority groups. Would you like me to implement any specific mitigation strategies or conduct additional analyses on particular aspects of these findings?

#### Prompt 3:

For data\_new.csv. Some groups in this dataset appear underrepresented and may receive disproportionately skewed predictions. How would you evaluate the severity of this imbalance and choose appropriate rebalancing or preprocessing techniques? Also show it

#### Response:

I'll help you evaluate the dataset imbalance and implement appropriate rebalancing techniques. Let's create a new analysis script that will:

1. Analyze the distribution of groups and their representation

2. Evaluate prediction skew across groups

3. Implement and compare different rebalancing techniques

4. Visualize the effects of rebalancing

Let's first install the required packages:

```bash

pip install imbalanced-learn

```

Now let's run our balance evaluation script:

```bash

python balance\_evaluation.py

```

Based on the analysis, I can provide a comprehensive evaluation of the dataset imbalance and recommendations for addressing it. Let's break down the findings:

1. \*\*Group Distribution Analysis\*\*:

a) \*\*Size Imbalance\*\*:

- White Female (white\_1): 27,077 (55.5%)

- White Male (white\_0): 16,125 (33.1%)

- Black Female (black\_1): 3,686 (7.6%)

- Black Male (black\_0): 1,896 (3.9%)

b) \*\*Severity Metrics\*\*:

- Imbalance Ratio: 14.28 (ratio between largest and smallest group)

- Coefficient of Variation: 0.96 (high variation in group sizes)

- Size Range: 1,896 - 27,077 samples

2. \*\*Prediction and Enrollment Rates\*\*:

Risk Score Means (prediction\_rate):

- Black Male: 5.56

- Black Female: 5.28

- White Male: 4.39

- White Female: 4.19

Enrollment Rates:

- Black Male: 1.48%

- Black Female: 1.38%

- White Male: 1.00%

- White Female: 0.78%

3. \*\*Class Imbalance\*\*:

- Negative class (0): 48,332 samples

- Positive class (1): 452 samples

- Extreme class imbalance ratio of approximately 107:1

4. \*\*Rebalancing Results\*\*:

Different techniques yield different balanced datasets:

- SMOTE: Balanced both classes to 48,332 samples

- Random Undersampling: Balanced both classes to 452 samples

- SMOTEENN: Created a more moderate balance (46,402:38,024)

\*\*Recommendations for Addressing the Imbalance\*\*:

1. \*\*Immediate Actions\*\*:

a) \*\*Primary Rebalancing Strategy\*\*:

- Use SMOTEENN as the primary rebalancing technique because:

\* It provides a more natural balance than pure oversampling or undersampling

\* It removes noise while generating synthetic samples

\* It helps prevent overfitting compared to pure SMOTE

b) \*\*Model Adjustments\*\*:

- Implement class weights in the model:

```python

class\_weights = {0: 1, 1: 107} # Based on class imbalance ratio

```

- Use these weights in model training to give more importance to minority class samples

2. \*\*Data Collection Strategies\*\*:

a) \*\*Targeted Data Collection\*\*:

- Focus on collecting more data from Black males (smallest group)

- Aim to at least triple the sample size for Black males to reduce the imbalance ratio

- Consider active learning approaches to identify most valuable new samples

b) \*\*Validation Strategy\*\*:

- Use stratified sampling for train/test splits

- Implement cross-validation with stratification by both race and gender

3. \*\*Model Development Approach\*\*:

a) \*\*Ensemble Methods\*\*:

- Use ensemble methods that handle imbalance well:

\* Balanced Random Forest

\* Easy Ensemble Classifier

\* RUSBoost

b) \*\*Evaluation Metrics\*\*:

- Focus on metrics suitable for imbalanced data:

\* Area Under Precision-Recall Curve (AUPRC)

\* F1-score

\* Balanced accuracy

\* Cohen's Kappa

4. \*\*Monitoring and Validation\*\*:

a) \*\*Regular Monitoring\*\*:

- Track prediction performance separately for each demographic group

- Monitor changes in group distributions over time

- Set up alerts for significant changes in group-wise performance

b) \*\*Validation Checks\*\*:

- Implement fairness constraints during training

- Regular bias audits across intersectional groups

- Compare performance metrics before and after rebalancing

Would you like me to implement any of these specific recommendations or create additional analyses for particular aspects of the imbalance problem?

## Experiment with diabetes dataset:

### 

### Gemini 2.5 pro

#### Prompt 1:

Check diabetes\_data.csv and the explanation of the dataset in IDS\_mapping.csv.

Could there be fairness concerns related to uneven group representation? Identify these disparities, and what strategies would you recommend to rebalance the dataset to support fair model training?

#### Response:

Okay, I can help with that. First, I need to locate the `diabetes\_data.csv` and `IDS\_mapping.csv` files. I'll check the `diabetesdatasets/` and `datasets/` directories.

Ran tool

Ran tool

Read file: diabetesdatasets/IDS\_mapping.csv

Read file: diabetesdatasets/diabetic\_data.csv

The `diabetic\_data.csv` file is too large to read directly. I'll try to use `codebase\_search` to identify columns related to sensitive attributes like race, gender, or age, which are common sources of bias. Then, I will use Python to count the occurrences of each category within those columns to check for imbalances.

Ran tool

The semantic search didn't return any specific column names. I'll have to proceed by loading the CSV and examining its columns directly. Since the file is large, I'll use a Python script to read only the header and a small sample of the data to identify relevant columns.

```bash

python -c "import pandas as pd; df = pd.read\_csv('diabetesdatasets/diabetic\_data.csv', nrows=5); print(df.columns.tolist()); print(df.head())"

```

Okay, the column names are now visible. The relevant columns for fairness analysis appear to be `race`, `gender`, and `age`.

Now, I'll write a Python script to count the occurrences of each unique value in these columns to identify potential representation disparities.

```bash

python -c "import pandas as pd; df = pd.read\_csv('diabetesdatasets/diabetic\_data.csv'); print('Race Distribution:\n', df['race'].value\_counts()); print('\nGender Distribution:\n', df['gender'].value\_counts()); print('\nAge Distribution:\n', df['age'].value\_counts())"

```

The analysis of `diabetic\_data.csv` reveals the following distributions:

\*\*Race Distribution:\*\*

\* Caucasian: 76099

\* AfricanAmerican: 19210

\* ?: 2273 (Missing/Unknown)

\* Hispanic: 2037

\* Other: 1506

\* Asian: 641

\*\*Gender Distribution:\*\*

\* Female: 54708

\* Male: 47055

\* Unknown/Invalid: 3

\*\*Age Distribution:\*\*

\* [70-80): 26068

\* [60-70): 22483

\* [50-60): 17256

\* [80-90): 17197

\* [40-50): 9685

\* [30-40): 3775

\* [90-100): 2793

\* [20-30): 1657

\* [10-20): 691

\* [0-10): 161

\*\*Fairness Concerns and Disparities:\*\*

1. \*\*Race:\*\* There's a significant imbalance in racial representation. Caucasians are the vast majority, while other groups, particularly Asians and Hispanics, are substantially underrepresented. The '?' category also indicates missing data that could further skew the representation if not handled properly.

2. \*\*Gender:\*\* While the gender distribution is relatively balanced between Female and Male, the "Unknown/Invalid" category, though small, should be addressed.

3. \*\*Age:\*\* The age distribution is skewed towards older individuals, with the age group [70-80) having the highest representation. Younger age groups, especially [0-10) and [10-20), are significantly underrepresented.

\*\*Strategies to Rebalance the Dataset for Fair Model Training:\*\*

1. \*\*Handling Missing Data (Race = '?'):\*\*

\* \*\*Imputation:\*\* If appropriate, impute missing race data using techniques like mode imputation or more sophisticated methods based on other patient features. However, this should be done cautiously as it can introduce bias if not handled well.

\* \*\*Treat as a Separate Category:\*\* Alternatively, treat '?' as a distinct category if the missingness itself might be informative.

\* \*\*Removal:\*\* If the number of missing values is small and randomly distributed, rows with missing race data could be removed, but this is generally not recommended for underrepresented groups. Given that 2273 is not a tiny number, removal might exacerbate underrepresentation of some actual groups hidden in the '?' category.

2. \*\*Addressing Imbalanced Representation (Race and Age):\*\*

\* \*\*Oversampling:\*\* Increase the number of instances in the minority classes.

\* \*\*Random Oversampling:\*\* Duplicate instances from minority classes. This is simple but can lead to overfitting.

\* \*\*SMOTE (Synthetic Minority Over-sampling Technique):\*\* Generate synthetic samples for minority classes by interpolating between existing instances. This is often more effective than simple random oversampling.

\* \*\*ADASYN (Adaptive Synthetic Sampling):\*\* Similar to SMOTE, but generates more synthetic data for minority class samples that are harder to learn.

\* \*\*Undersampling:\*\* Decrease the number of instances in the majority class.

\* \*\*Random Undersampling:\*\* Randomly remove instances from the majority class. This can lead to loss of information.

\* \*\*Tomek Links:\*\* Remove pairs of instances from different classes that are close to each other (one from majority, one from minority). This helps to clean the class boundary.

\* \*\*Cluster Centroids:\*\* Replace clusters of majority class samples with their centroids.

\* \*\*Hybrid Approaches:\*\* Combine oversampling and undersampling techniques (e.g., SMOTE + Tomek Links).

\* \*\*Cost-Sensitive Learning:\*\* Assign different misclassification costs to different classes during model training. Higher costs for misclassifying minority classes can force the model to pay more attention to them. This doesn't change the dataset itself but alters the learning algorithm's objective.

\* \*\*Algorithmic Fairness Approaches (Post-Hoc or In-Processing):\*\*

\* \*\*Reweighing:\*\* Assign weights to samples inversely proportional to their class frequency during training.

\* Fairness-aware machine learning algorithms can be used that incorporate fairness constraints directly into the model training process.

\* Post-processing techniques can adjust model predictions to satisfy fairness criteria.

3. \*\*Handling "Unknown/Invalid" Gender:\*\*

\* Given the very small number (3), these instances could potentially be removed if imputation is not feasible or if they are deemed uninformative. However, investigate why this data is invalid.

\*\*Recommended Steps:\*\*

1. \*\*Further Investigation of '?' Race Values:\*\* Try to understand why these values are missing. Are they concentrated in specific hospitals or demographics? This might inform the best imputation strategy.

2. \*\*Choose an Appropriate Rebalancing Strategy:\*\* The best strategy depends on the specific dataset, the machine learning model to be used, and the fairness metrics you aim to optimize.

\* For the `race` attribute, SMOTE or ADASYN for minority groups (Asian, Hispanic, Other, and potentially AfricanAmerican depending on the target level of balance) combined with careful handling of the '?' category would be a good starting point.

\* For the `age` attribute, if the goal is to improve model performance for younger, underrepresented age groups, oversampling techniques like SMOTE could be applied to these groups.

3. \*\*Evaluate Fairness:\*\* After rebalancing and training your model, it's crucial to evaluate its fairness across different demographic groups using appropriate fairness metrics (e.g., demographic parity, equalized odds, predictive rate parity).

It's important to note that there's no one-size-fits-all solution. The choice of strategy should be guided by domain knowledge and empirical evaluation of model performance and fairness. Remember to split your data into training and testing sets \*before\* applying any rebalancing techniques to the training data only, to avoid data leakage and get a realistic estimate of performance on unseen data.

#### Prompt 2:

Only use and check diabetes\_data.csv and the explanation of the dataset in IDS\_mapping.csv.

Are there signs of uneven prediction outcomes across groups that could raise fairness concerns? What metrics would you use to assess potential bias (also use them on this dataset), and how would you interpret the results to determine whether mitigation is necessary?

#### Response:

You're asking a crucial question about ensuring fairness in predictive modeling. To fully assess uneven \*prediction\* outcomes and apply fairness metrics like Equal Opportunity or Equalized Odds, we would typically need:

1. A trained machine learning model.

2. The predictions made by that model on the `diabetic\_data.csv` dataset.

Since we don't have a specific model or its predictions here, I can't calculate metrics that compare model predictions to actual outcomes (e.g., True Positive Rates across groups).

However, we \*\*can\*\* analyze the \*\*actual observed outcome rates\*\* (i.e., the distribution of the `readmitted` column) across different demographic groups (`race`, `gender`, `age`) in the dataset. This is a vital first step. Significant differences in these \*actual\* outcome rates (often called "base rates") can indicate pre-existing disparities or highlight groups that might be disproportionately affected by the condition. If a model then fails to account for these differences appropriately, or amplifies them, it can lead to biased predictions.

Let's examine these actual outcome distributions. I will calculate the rate of each `readmitted` category ('NO', '>30', '<30') for each demographic group.

I'll start by analyzing readmission rates by `race`.

```bash

python -c "import pandas as pd; df = pd.read\_csv('diabetesdatasets/diabetic\_data.csv'); print('Readmission rates by Race:\n'); print(pd.crosstab(df['race'], df['readmitted'], normalize='index') \* 100)"

```

Okay, I've calculated the readmission rates broken down by `race`. Here are the results:

\*\*Readmission Rates by Race (Actual Outcomes):\*\*

| race | <30 (%) | >30 (%) | NO (%) |

| :-------------- | :------ | :------ | :------ |

| ? | 8.27 | 23.67 | 68.06 |

| AfricanAmerican | 11.22 | 34.53 | 54.25 |

| Asian | 10.14 | 25.12 | 64.74 |

| Caucasian | 11.29 | 35.64 | 53.07 |

| Hispanic | 10.41 | 31.52 | 58.08 |

| Other | 9.63 | 29.61 | 60.76 |

Now, let's look at the readmission rates by `gender`.

```bash

python -c "import pandas as pd; df = pd.read\_csv('diabetesdatasets/diabetic\_data.csv'); print('\nReadmission rates by Gender:\n'); print(pd.crosstab(df['gender'], df['readmitted'], normalize='index') \* 100)"

```

Here are the readmission rates by `gender`:

\*\*Readmission Rates by Gender (Actual Outcomes):\*\*

| gender | <30 (%) | >30 (%) | NO (%) |

| :-------------- | :------ | :------ | :------- |

| Female | 11.25 | 35.68 | 53.08 |

| Male | 11.06 | 34.06 | 54.88 |

| Unknown/Invalid | 0.00 | 0.00 | 100.00 |

And finally, readmission rates by `age`.

```bash

python -c "import pandas as pd; df = pd.read\_csv('diabetesdatasets/diabetic\_data.csv'); print('\nReadmission rates by Age:\n'); print(pd.crosstab(df['age'], df['readmitted'], normalize='index') \* 100)"

```

Here are the readmission rates by `age`:

\*\*Readmission Rates by Age (Actual Outcomes):\*\*

| age | <30 (%) | >30 (%) | NO (%) |

| :------- | :------ | :------ | :------ |

| [0-10) | 1.86 | 16.15 | 81.99 |

| [10-20) | 5.79 | 32.42 | 61.79 |

| [20-30) | 14.24 | 30.78 | 54.98 |

| [30-40) | 11.23 | 31.44 | 57.32 |

| [40-50) | 10.60 | 33.85 | 55.55 |

| [50-60) | 9.67 | 34.29 | 56.04 |

| [60-70) | 11.13 | 35.12 | 53.75 |

| [70-80) | 11.77 | 36.35 | 51.88 |

| [80-90) | 12.08 | 36.19 | 51.73 |

| [90-100) | 11.10 | 28.93 | 59.97 |

\*\*Interpreting These Base Rate Differences (Signs of Potential Fairness Concerns):\*\*

Even without a model, these base rates show differences in actual readmission outcomes across groups:

1. \*\*Race:\*\*

\* \*\*`<30` Readmissions:\*\* AfricanAmerican (11.22%) and Caucasian (11.29%) patients have a noticeably higher rate of readmission within 30 days compared to Asian (10.14%), Hispanic (10.41%), Other (9.63%), and especially the '?' category (8.27%).

\* \*\*`>30` Readmissions:\*\* Caucasian (35.64%) and AfricanAmerican (34.53%) groups also have higher rates here compared to other groups.

\* \*\*`NO` Readmissions:\*\* Consequently, Caucasian and AfricanAmerican patients have lower rates of not being readmitted.

\* \*\*Concern:\*\* If a model is trained without accounting for these base rate differences, it might learn to associate being AfricanAmerican or Caucasian with a higher likelihood of readmission, even if other factors are more predictive. This could lead to disparities in how the model predicts risk for these groups. For example, a model might be overly sensitive to predicting readmission for these groups or less sensitive for groups with lower base rates of readmission.

2. \*\*Gender:\*\*

\* The rates for `<30`, `>30`, and `NO` readmissions are very similar between Female and Male patients.

\* The 'Unknown/Invalid' category (3 instances) shows 100% 'NO' readmission, but with such a small sample, this is not informative.

\* \*\*Concern:\*\* Gender seems to have less pronounced differences in actual outcome rates compared to race or age in this dataset, which is a positive sign from a base rate perspective. However, model performance should still be checked.

3. \*\*Age:\*\*

\* \*\*`<30` Readmissions:\*\* The highest rate is in the [20-30) age group (14.24%), followed by [80-90) (12.08%) and [70-80) (11.77%). The youngest groups [0-10) (1.86%) and [10-20) (5.79%) have significantly lower rates. The [50-60) group also has a lower rate (9.67%).

\* \*\*`>30` Readmissions:\*\* Generally increases with age up to [70-80) (36.35%), then slightly dips for [90-100).

\* \*\*`NO` Readmissions:\*\* Highest for the youngest ([0-10) at 81.99%) and generally decreases as age increases, with the lowest rates for [70-80) and [80-90).

\* \*\*Concern:\*\* There are clear, substantial differences in readmission patterns across age groups. A model needs to capture these trends accurately. If a model performs poorly for, say, the [20-30) group (which has a high `<30` readmission rate despite being younger) or for the very young (who have very low readmission rates), it could lead to unfair or ineffective predictions. The underrepresentation of young patients we saw earlier, combined with these different outcome rates, means a model might not learn reliable patterns for them.

\*\*Metrics You Would Use (with a Trained Model):\*\*

If you had a trained model and its predictions, you would use the following metrics (as detailed previously) to assess for bias, calculating them for each subgroup within `race`, `gender`, and `age` for the `<30` readmission outcome (as it's often the most critical):

1. \*\*Demographic Parity (Statistical Parity):\*\*

\* \*Compares:\* `P(Predicted = <30 | Group A)` vs. `P(Predicted = <30 | Group B)`

\* \*Interpretation for this dataset:\* Are certain races or age groups predicted to be readmitted `<30` at significantly different rates, regardless of their actual readmission?

2. \*\*Equal Opportunity:\*\*

\* \*Compares:\* `P(Predicted = <30 | Actual = <30, Group A)` vs. `P(Predicted = <30 | Actual = <30, Group B)` (True Positive Rate)

\* \*Interpretation:\* Is the model equally good at identifying patients who \*actually will be\* readmitted `<30` across all groups? A lower TPR for a group means more missed cases for them.

3. \*\*Equalized Odds:\*\*

\* \*Compares:\* Both TPR (as above) AND `P(Predicted = Not <30 | Actual = Not <30, Group A)` vs. `P(Predicted = Not <30 | Actual = Not <30, Group B)` (True Negative Rate). Or equivalently, TPR and False Positive Rate.

\* \*Interpretation:\* Is the model balanced in both correctly identifying `<30` readmissions AND correctly identifying those \*not\* at risk of `<30` readmission, across all groups?

4. \*\*Predictive Rate Parity (Precision Parity):\*\*

\* \*Compares:\* `P(Actual = <30 | Predicted = <30, Group A)` vs. `P(Actual = <30 | Predicted = <30, Group B)` (Positive Predictive Value)

\* \*Interpretation:\* When the model predicts a `<30` readmission, is that prediction equally reliable (likely to be correct) across all groups?

5. \*\*Accuracy Parity:\*\*

\* \*Compares:\* Overall accuracy for Group A vs. Group B.

\* \*Interpretation:\* Is the model's overall performance (correct predictions / total predictions) similar across groups?

\*\*How to Interpret Results to Determine if Mitigation is Necessary:\*\*

1. \*\*Examine Base Rate Differences (as we just did):\*\* If significant differences in actual outcomes exist (e.g., AfricanAmericans have a higher actual `<30` readmission rate), this is important context.

2. \*\*Calculate Fairness Metrics (with a model):\*\* For each metric, compare its value across the different categories within `race`, `gender`, and `age`.

3. \*\*Look for Significant Gaps:\*\*

\* \*\*Example 1 (Equal Opportunity Violation):\*\* If the TPR for `<30` readmission for Caucasians is 75%, but for Asians it's only 50%, this is a major concern. It means the model is much worse at identifying high-risk Asian patients.

\* \*\*Example 2 (Demographic Parity Violation):\*\* If the model predicts `<30` readmission for 15% of AfricanAmerican patients but only 5% of Hispanic patients, while their \*actual\* `<30` readmission rates are 11% and 10% respectively, this suggests the model is disproportionately flagging AfricanAmerican patients.

4. \*\*Consider Thresholds (e.g., 80% rule):\*\* Is the TPR for one group less than 80% of the TPR for the best-performing group?

5. \*\*Assess Impact:\*\*

\* What are the consequences of the identified disparity? If Equal Opportunity is violated for `<30` readmissions, it means some groups are not getting the timely interventions they might need.

\* If Predictive Rate Parity is violated, a prediction of high risk might be less trustworthy for certain groups, leading to misallocation of resources or undue stress.

6. \*\*Decision for Mitigation:\*\*

\* Mitigation is \*\*necessary\*\* if the fairness metrics reveal disparities that are:

\* Statistically significant.

\* Practically impactful in a negative way (e.g., leading to unequal access to care, higher error rates for certain groups).

\* Violate ethical principles or legal guidelines for fairness in healthcare.

\* Given the base rate differences we've seen, particularly for `race` and `age`, it is highly probable that a standard model would exhibit some form of bias. Therefore, careful auditing with these metrics after model training, followed by appropriate mitigation strategies (like those discussed in our previous conversation: re-sampling, re-weighting, fairness-aware algorithms, post-processing adjustments), would be essential.

By first analyzing these actual outcome distributions, you've taken a good step in anticipating where fairness issues might arise with a predictive model. The next step, once a model is built, would be the rigorous application and interpretation of the fairness metrics described.

#### Prompt 3:

Only use and check diabetes\_data.csv and the explanation of the dataset in IDS\_mapping.csv.

Some groups in this dataset appear underrepresented and may receive disproportionately skewed predictions. How would you evaluate the severity of this imbalance and choose appropriate rebalancing or preprocessing techniques? Also show it

#### Response:

You're right to focus on the impact of underrepresentation on prediction fairness. We've already seen that some groups in `race` and `age` are significantly smaller than others. Let's re-examine those distributions to evaluate the severity and then discuss appropriate rebalancing techniques, including conceptual examples of how you might apply them.

\*\*1. Evaluating the Severity of Imbalance\*\*

First, let's get the exact counts and percentages for `race` and `age` to quantify the imbalances.

```bash

python -c "import pandas as pd; df = pd.read\_csv('diabetesdatasets/diabetic\_data.csv'); print('Race Distribution (Counts):\n', df['race'].value\_counts()); print('\nRace Distribution (Percentage):\n', df['race'].value\_counts(normalize=True) \* 100)"

```

```bash

python -c "import pandas as pd; df = pd.read\_csv('diabetesdatasets/diabetic\_data.csv'); print('\nAge Distribution (Counts):\n', df['age'].value\_counts()); print('\nAge Distribution (Percentage):\n', df['age'].value\_counts(normalize=True) \* 100)"

```

You're absolutely right. The underrepresentation of certain groups is a significant concern that can lead to models performing poorly or unfairly for these groups.

Let's break down the severity of the imbalance in `race` and `age` based on the distributions we've already calculated, and then discuss how to choose and apply rebalancing techniques.

\*\*1. Evaluating the Severity of Imbalance\*\*

\*\*Race Distribution:\*\*

\* \*\*Counts:\*\*

\* Caucasian: 76099

\* AfricanAmerican: 19210

\* ?: 2273

\* Hispanic: 2037

\* Other: 1506

\* Asian: 641

\* \*\*Percentage:\*\*

\* Caucasian: 74.78%

\* AfricanAmerican: 18.88%

\* ?: 2.23% (Missing values)

\* Hispanic: 2.00%

\* Other: 1.48%

\* Asian: 0.63%

\* \*\*Severity Assessment (Race):\*\* \*\*High.\*\*

\* The dataset is overwhelmingly composed of Caucasian patients.

\* AfricanAmerican patients form a substantial minority but are still outnumbered roughly 4:1 by Caucasians.

\* Hispanic, Other, and particularly Asian groups are severely underrepresented. For instance, there are over 100 times more Caucasian patients than Asian patients.

\* The 2.23% of '?' (missing) race data also needs to be addressed as it could hide further imbalances or affect model training.

\* \*\*Risk:\*\* Models trained on this data might develop biases, performing well for Caucasian patients but poorly or unfairly for underrepresented racial groups. Predictions for Asian or Hispanic patients, for example, might be less reliable or systematically skewed.

\*\*Age Distribution:\*\*

\* \*\*Counts:\*\*

\* [70-80): 26068

\* [60-70): 22483

\* [50-60): 17256

\* [80-90): 17197

\* [40-50): 9685

\* [30-40): 3775

\* [90-100): 2793

\* [20-30): 1657

\* [10-20): 691

\* [0-10): 161

\* \*\*Percentage:\*\*

\* [70-80): 25.62%

\* [60-70): 22.09%

\* [50-60): 16.96%

\* [80-90): 16.90%

\* [40-50): 9.52%

\* [30-40): 3.71%

\* [90-100): 2.74%

\* [20-30): 1.63%

\* [10-20): 0.68%

\* [0-10): 0.16%

\* \*\*Severity Assessment (Age):\*\* \*\*High.\*\*

\* The dataset is heavily skewed towards older patients, with over 80% of patients being 50 years or older.

\* Younger age groups, particularly those under 30 ([0-10), [10-20), [20-30)), are extremely underrepresented. For example, the [0-10) age group constitutes only 0.16% of the data.

\* \*\*Risk:\*\* The model may not learn effective patterns for younger patients due to insufficient data, leading to poor predictive performance and potentially unfair outcomes for these age groups.

\*\*2. Choosing Appropriate Rebalancing or Preprocessing Techniques\*\*

The choice of technique depends on several factors:

\* \*\*Severity of Imbalance:\*\* Extreme imbalances often benefit from oversampling minority classes or using hybrid approaches.

\* \*\*Dataset Size:\*\* Undersampling the majority class can be risky with smaller datasets as it discards potentially valuable information. With a large dataset like this (100k+ rows), undersampling might be viable.

\* \*\*Nature of the Data/Problem:\*\* Are synthetic samples meaningful? Does the problem require preserving the original data distribution as much as possible?

\* \*\*Computational Cost:\*\* Some advanced techniques can be computationally intensive.

\* \*\*Model Sensitivity:\*\* Some models are more sensitive to class imbalance than others (e.g., decision trees can be more robust than SVMs or logistic regression in some cases).

\*\*General Preprocessing Steps First:\*\*

1. \*\*Handle Missing Values ('?' in race):\*\*

\* \*\*Imputation:\*\* Replace '?' with the mode ('Caucasian'), or use a more sophisticated imputation technique (e.g., KNNImputer, or predict 'race' based on other features). Mode imputation is simple but might reinforce the majority class bias.

\* \*\*Treat as Separate Category:\*\* If missingness is potentially informative.

\* \*\*Removal:\*\* Not recommended here as it's a sizable number of rows (2273).

2. \*\*Data Splitting:\*\* \*\*Crucially, always split your data into training and testing sets BEFORE applying any rebalancing techniques.\*\* Rebalancing should only be applied to the training data to prevent data leakage and ensure your test set reflects the original, real-world data distribution for evaluation.

\*\*Common Rebalancing Techniques:\*\*

\* \*\*A. Oversampling Minority Classes:\*\* (Increases the number of instances in underrepresented classes)

1. \*\*RandomOverSampler:\*\*

\* \*How:\* Randomly duplicates samples from the minority class(es).

\* \*Pros:\* Simple to implement.

\* \*Cons:\* Can lead to overfitting as it makes exact copies of existing instances.

2. \*\*SMOTE (Synthetic Minority Over-sampling Technique):\*\*

\* \*How:\* Creates synthetic samples by finding k-nearest neighbors for minority class instances and generating new samples along the line segments joining some/all of these neighbors.

\* \*Pros:\* Often more effective than random oversampling as it generates new, plausible samples, reducing overfitting.

\* \*Cons:\* Can create noisy samples if the minority class is very sparse or overlaps significantly with the majority class. May not be ideal for high-dimensional data. Variants like Borderline-SMOTE or ADASYN try to address this.

3. \*\*ADASYN (Adaptive Synthetic Sampling):\*\*

\* \*How:\* Similar to SMOTE, but generates more synthetic data for minority class samples that are harder to learn (i.e., those with fewer minority neighbors in their vicinity).

\* \*Pros:\* Focuses on difficult-to-learn instances, potentially improving model performance on these.

\* \*Cons:\* Can be sensitive to outliers.

\* \*\*B. Undersampling Majority Classes:\*\* (Reduces the number of instances in overrepresented classes)

1. \*\*RandomUnderSampler:\*\*

\* \*How:\* Randomly removes samples from the majority class(es).

\* \*Pros:\* Simple; can help with very large datasets by reducing training time.

\* \*Cons:\* Can discard potentially useful information from the majority class, leading to underfitting or poorer generalization.

2. \*\*Tomek Links:\*\*

\* \*How:\* Identifies pairs of instances from different classes that are each other's nearest neighbors (Tomek links). Instances from the majority class in such pairs are removed. This helps clean the boundary between classes.

\* \*Pros:\* Improves class separation, potentially leading to better classifier performance.

\* \*Cons:\* Might not significantly change class balance if classes are well-separated.

3. \*\*NearMiss:\*\*

\* \*How:\* Selects majority class samples whose average distance to some number of CLoSEST minority class samples is smallest (NearMiss-1), or whose average distance to some number of FARTHEST minority class samples is smallest (NearMiss-2), or selects a given number of majority class samples closest to each minority class sample (NearMiss-3).

\* \*Pros:\* More controlled undersampling than random.

\* \*Cons:\* Can be sensitive to noisy data or outliers. Performance varies depending on the version.

\* \*\*C. Hybrid Approaches (Combining Oversampling and Undersampling):\*\*

1. \*\*SMOTETomek:\*\*

\* \*How:\* Applies SMOTE to oversample the minority class, then Tomek links to clean up noisy samples from both classes near the class boundary.

\* \*Pros:\* Often performs better than using either technique alone by benefiting from SMOTE's sample generation and Tomek's cleaning.

2. \*\*SMOTEENN (SMOTE + Edited Nearest Neighbors):\*\*

\* \*How:\* SMOTE is applied, then ENN is used to remove samples (from both classes) whose class label differs from the majority of its k-nearest neighbors. This further cleans the data.

\* \*Pros:\* Similar to SMOTETomek, aims for a cleaner, better-defined class distribution.

\* \*\*D. Algorithmic Approaches (Modifying the Learning Algorithm):\*\*

1. \*\*Cost-Sensitive Learning:\*\*

\* \*How:\* Assign higher misclassification costs to errors on the minority class during model training. Many algorithms (e.g., SVMs, logistic regression, tree-based methods) have parameters for class weights.

\* \*Pros:\* Doesn't alter the data itself; directly influences the model's learning process.

\* \*Cons:\* Finding optimal costs can require tuning.

2. \*\*Ensemble Methods:\*\* Techniques like BalancedRandomForestClassifier or EasyEnsembleClassifier from `imblearn` create multiple base estimators (e.g., decision trees) on balanced bootstrap samples of the data.

\*\*3. Showing Conceptual Examples (Python with `imblearn`)\*\*

Below are conceptual Python snippets using the `imblearn` library (which you'd need to install: `pip install imbalanced-learn`). These snippets assume you have your features (`X`) and target (`y`) separated, and that `X\_train`, `y\_train` come from a train-test split.

\*\*Important:\*\* These are for illustration. You would apply these \*only to your training data\*.

```python

# --- Conceptual Code Snippets ---

# Assume X\_train, y\_train are your training features and target

# And 'race' or 'age' is one of the columns in X\_train or affects y\_train's balance

# For simplicity, let's imagine we're rebalancing based on the target variable 'y\_train'

# which itself might be imbalanced, and this imbalance could be correlated with 'race' or 'age'.

# 0. Setup (Illustrative - you would load your data properly)

# import pandas as pd

# from sklearn.model\_selection import train\_test\_split

# from collections import Counter

# # Load your data (replace with actual loading)

# # df = pd.read\_csv('diabetesdatasets/diabetic\_data.csv')

# # # Preprocess: e.g., handle '?' in race, encode categoricals, define X and y

# # # For illustration, assume 'y' is a binary target derived from 'readmitted'

# # y = df['readmitted'].apply(lambda x: 1 if x == '<30' else 0)

# # X = df.drop('readmitted', axis=1)

# # # ... further preprocessing for X (encoding, scaling, feature selection) ...

# # X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42, stratify=y)

# 1. Handling '?' in Race (Example: Mode Imputation)

# X\_train['race'] = X\_train['race'].replace('?', X\_train['race'].mode()[0])

# X\_test['race'] = X\_test['race'].replace('?', X\_train['race'].mode()[0]) # Use mode from train set

# 2. SMOTE (Oversampling)

# from imblearn.over\_sampling import SMOTE

# print(f"Original training set shape: {Counter(y\_train)}")

# smote = SMOTE(random\_state=42, sampling\_strategy='auto') # Adjust sampling\_strategy as needed

# X\_train\_smote, y\_train\_smote = smote.fit\_resample(X\_train, y\_train)

# print(f"Resampled training set shape (SMOTE): {Counter(y\_train\_smote)}")

# 3. RandomUnderSampler (Undersampling)

# from imblearn.under\_sampling import RandomUnderSampler

# print(f"Original training set shape: {Counter(y\_train)}")

# rus = RandomUnderSampler(random\_state=42, sampling\_strategy='auto') # sampling\_strategy='majority'

# X\_train\_rus, y\_train\_rus = rus.fit\_resample(X\_train, y\_train)

# print(f"Resampled training set shape (RandomUnderSampler): {Counter(y\_train\_rus)}")

# 4. SMOTETomek (Hybrid)

# from imblearn.combine import SMOTETomek

# print(f"Original training set shape: {Counter(y\_train)}")

# smt = SMOTETomek(random\_state=42, smote=SMOTE(sampling\_strategy='auto'), tomek=TomekLinks(sampling\_strategy='all'))

# X\_train\_smt, y\_train\_smt = smt.fit\_resample(X\_train, y\_train)

# print(f"Resampled training set shape (SMOTETomek): {Counter(y\_train\_smt)}")

# 5. Cost-Sensitive Learning (Example with RandomForestClassifier)

# from sklearn.ensemble import RandomForestClassifier

# # Define class weights - e.g., if class 0 is majority and class 1 is minority

# # weights = {0: 1, 1: 10} # Give 10x more weight to misclassifying class 1

# # model = RandomForestClassifier(class\_weight=weights, random\_state=42)

# # model.fit(X\_train, y\_train) # Train on original (but possibly preprocessed) data

```

\*\*Applying to Specific Columns like `race` or `age`:\*\*

If you are rebalancing the dataset specifically to ensure better representation of, say, different `race` categories \*before\* defining your final target `y` for prediction (e.g. if `race` itself was a feature and you wanted to ensure fairness in some intermediate step, or if you were doing multi-class classification on `race`), the approach would be more complex. `imblearn` samplers are primarily designed to balance the target variable `y`.

If your goal is to improve model fairness concerning `race` or `age` as sensitive attributes \*when predicting an outcome like `readmitted`\*, you would typically:

1. Preprocess `race` (handle '?') and `age` as features.

2. Apply rebalancing techniques based on the `readmitted` target variable (especially if it's imbalanced, which it is: 'NO' is ~54%, '>30' is ~35%, '<30' is ~11%).

3. After training your model on the (potentially rebalanced) training data, you would then evaluate fairness metrics (Demographic Parity, Equal Opportunity, etc.) across the different `race` and `age` subgroups using your \*original, un-rebalanced test set\*.

\*\*Choosing the Right Strategy:\*\*

\* \*\*Start Simple:\*\* Try RandomOverSampler or RandomUnderSampler if the imbalance isn't too extreme or as a baseline.

\* \*\*Advance to SMOTE/ADASYN:\*\* For more robust oversampling.

\* \*\*Consider Hybrid:\*\* SMOTETomek or SMOTEENN often provide good results.

\* \*\*Experiment:\*\* The best technique is often found through experimentation. Train models using data rebalanced with different techniques and evaluate performance and fairness metrics on the test set.

\* \*\*Focus on the Minority '<30' class for `readmitted`:\*\* When rebalancing based on the `readmitted` column, you'd likely focus on increasing the representation of the '<30' class, or balancing all three classes.

Given the severity of imbalance in both `race` (especially for Asian, Hispanic, Other) and `age` (younger groups), techniques like \*\*SMOTE or ADASYN\*\* for oversampling these specific groups (if they were your direct target for classification, which is not the case here) or using \*\*cost-sensitive learning\*\* to make your model pay more attention to errors on predictions related to these groups would be good starting points when predicting `readmitted`. If `readmitted` itself is the target, and it's imbalanced, apply these techniques to `readmitted` and then check fairness across `race`/`age` subgroups.

### Claude 3.5 sonnet

#### Prompt 1:

Check diabetes\_data.csv and the explanation of the dataset in IDS\_mapping.csv.

Could there be fairness concerns related to uneven group representation? Identify these disparities, and what strategies would you recommend to rebalance the dataset to support fair model training?

#### Response:

I notice that neither `diabetes\_data.csv` nor `IDS\_mapping.csv` are directly visible in the workspace. Let me first locate these files and examine their contents.

Ran tool

Let me check the contents of the diabetesdatasets directory:

Ran tool

Perfect, I found both files. Let me first examine the IDS\_mapping.csv to understand the dataset structure:

Read file: diabetesdatasets/IDS\_mapping.csv

Now, let me examine the structure of the diabetes data:

```bash

head -n 5 diabetesdatasets/diabetic\_data.csv

```

Let me create a comprehensive analysis script to evaluate group representation and fairness concerns in the diabetes dataset:

Now let's run this analysis:

```bash

python diabetes\_fairness\_analysis.py

```

Based on the analysis, I can identify several significant fairness concerns and disparities in the diabetes dataset. Let me break down the findings and provide recommendations:

\*\*1. Demographic Representation Disparities:\*\*

A. Race Distribution:

- Caucasian: 76,099 (74.9%)

- African American: 19,210 (18.9%)

- Hispanic: 2,037 (2.0%)

- Other: 1,506 (1.5%)

- Asian: 641 (0.6%)

- Unknown: 2,273 (2.2%)

B. Gender Distribution:

- Female: 54,708 (53.8%)

- Male: 47,055 (46.2%)

- Unknown/Invalid: 3 (<0.1%)

\*\*2. Outcome Disparities:\*\*

A. Readmission Rates by Race:

- Caucasian: 46.9%

- African American: 45.8%

- Hispanic: 41.9%

- Other: 39.2%

- Asian: 35.3%

- Unknown: 31.9%

B. Readmission Rates by Gender:

- Female: 46.9%

- Male: 45.1%

C. Treatment Disparities:

- Diabetes Medication Usage:

\* Highest: Unknown race (82.3%)

\* Lowest: Asian (74.1%)

\* Male (77.7%) vs Female (76.4%)

\*\*3. Key Fairness Concerns:\*\*

1. \*\*Severe Underrepresentation:\*\*

- Asian patients (0.6%)

- Hispanic patients (2.0%)

- Other racial groups (1.5%)

2. \*\*Outcome Disparities:\*\*

- 11.6 percentage point gap in readmission rates between highest (Caucasian) and lowest (Asian) groups

- 1.8 percentage point gender gap in readmission rates

- Race disparity ratio: 1.47 (indicates significant disparity)

\*\*Recommended Rebalancing Strategies:\*\*

1. \*\*Data-Level Interventions:\*\*

```python

# Implement weighted sampling strategy

def calculate\_balanced\_weights(df):

weights = np.ones(len(df))

# Race weights (inverse frequency)

race\_weights = 1 / df['race'].value\_counts(normalize=True)

for race in race\_weights.index:

weights[df['race'] == race] \*= race\_weights[race]

# Adjust for outcome imbalance within groups

for race in df['race'].unique():

race\_mask = df['race'] == race

readmit\_rate = df[race\_mask]['readmitted\_binary'].mean()

weights[race\_mask & (df['readmitted\_binary'] == 1)] \*= (1 / readmit\_rate)

weights[race\_mask & (df['readmitted\_binary'] == 0)] \*= (1 / (1 - readmit\_rate))

return weights

```

2. \*\*Sampling Strategies:\*\*

```python

# Two-phase sampling strategy

def balanced\_sample\_generator(df):

# Phase 1: Oversample minority races

race\_sampler = SMOTE(sampling\_strategy={

'Asian': 15000, # Increase to ~15% representation

'Hispanic': 15000,

'Other': 15000

})

# Phase 2: Balance outcomes within each racial group

stratified\_sampler = StratifiedKFold(n\_splits=5, shuffle=True)

return combined\_samples

```

3. \*\*Model-Level Adjustments:\*\*

```python

# Custom loss function with fairness constraints

def fairness\_aware\_loss(y\_true, y\_pred, group\_membership):

base\_loss = binary\_crossentropy(y\_true, y\_pred)

# Add demographic parity constraint

demographic\_penalty = calculate\_demographic\_disparity(y\_pred, group\_membership)

# Add equal opportunity constraint

opportunity\_penalty = calculate\_equal\_opportunity\_disparity(y\_true, y\_pred, group\_membership)

return base\_loss + 0.1 \* demographic\_penalty + 0.1 \* opportunity\_penalty

```

\*\*Implementation Plan:\*\*

1. \*\*Immediate Actions:\*\*

- Implement weighted sampling using the calculated balanced weights

- Use stratified sampling for train/test splits based on both race and gender

- Add fairness constraints to the model's objective function

2. \*\*Data Collection Recommendations:\*\*

- Prioritize collection of data from Asian, Hispanic, and Other racial groups

- Investigate and document potential causes of missing race data

- Implement protocols to ensure complete demographic data collection

3. \*\*Model Development Guidelines:\*\*

- Use cross-validation with stratification by race and gender

- Implement separate performance thresholds for different demographic groups

- Regular monitoring of disparities in model predictions

4. \*\*Clinical Considerations:\*\*

- Document identified disparities for clinical review

- Consider different risk thresholds for different demographic groups

- Implement regular bias audits in the deployment pipeline

Would you like me to implement any of these specific strategies or provide more detailed analysis of particular aspects of the disparities?

#### Prompt 2:

Only use and check diabetes\_data.csv and the explanation of the dataset in IDS\_mapping.csv.

Are there signs of uneven prediction outcomes across groups that could raise fairness concerns? What metrics would you use to assess potential bias (also use them on this dataset), and how would you interpret the results to determine whether mitigation is necessary?

#### **Response:**

I'll create a comprehensive bias assessment focusing specifically on the diabetes dataset and using appropriate fairness metrics. Let me create a new analysis script that incorporates multiple fairness metrics:

Let's run this enhanced fairness analysis:

```bash

python diabetes\_bias\_assessment.py

```

Based on our comprehensive fairness analysis of the diabetes dataset, I can identify several significant fairness concerns and provide recommendations for mitigation. Let me break down the findings using multiple fairness metrics:

\*\*1. Representation Disparities:\*\*

A. Race Distribution:

- Caucasian: 74.8%

- African American: 18.9%

- Unknown (?): 2.2%

- Hispanic: 2.0%

- Other: 1.5%

- Asian: 0.6%

B. Gender Distribution:

- Female: 53.8%

- Male: 46.2%

- Unknown: <0.1%

\*\*2. Key Fairness Metrics:\*\*

A. Statistical Parity Differences (from overall readmission rate):

- Caucasian: +0.8%

- African American: -0.3%

- Asian: -10.8%

- Hispanic: -4.2%

- Other: -6.8%

- Unknown: -14.1%

B. Disparate Impact Ratios (relative to overall rate):

- Caucasian: 1.018

- African American: 0.993

- Asian: 0.765 (concerning)

- Hispanic: 0.910

- Other: 0.851

- Unknown: 0.693 (concerning)

C. Prediction Fairness:

1. False Positive Rates:

- Range: 45.8% - 48.4%

- Highest: Asian (48.4%)

- Lowest: Other (45.8%)

2. True Positive Rates (Equal Opportunity):

- Range: 51.0% - 56.5%

- Highest: Unknown (56.5%)

- Lowest: African American (51.0%)

\*\*3. Treatment Equity:\*\*

Diabetes Medication Rates:

- Unknown: 82.3%

- Other: 80.9%

- African American: 77.0%

- Caucasian: 76.8%

- Hispanic: 76.1%

- Asian: 74.1%

\*\*Significant Fairness Concerns:\*\*

1. \*\*Representation Bias:\*\*

- Severe underrepresentation of Asian (0.6%), Hispanic (2.0%), and Other (1.5%) populations

- This underrepresentation could lead to less reliable predictions for these groups

2. \*\*Outcome Disparities:\*\*

- 15 percentage point gap in readmission rates between highest (Caucasian: 46.9%) and lowest (Unknown: 31.9%) groups

- Asian and Hispanic populations show significantly lower readmission rates, which could indicate:

\* Underidentification of risk factors

\* Access barriers to follow-up care

\* Cultural or linguistic barriers

3. \*\*Treatment Disparities:\*\*

- 8.2 percentage point gap in diabetes medication rates between highest and lowest groups

- Asian patients consistently show lower treatment rates

\*\*Recommended Mitigation Strategies:\*\*

1. \*\*Data Collection and Preprocessing:\*\*

```python

def implement\_balanced\_sampling(df):

# Calculate sampling weights

weights = {

'Asian': 10.0, # Increase Asian representation

'Hispanic': 5.0, # Increase Hispanic representation

'Other': 5.0, # Increase Other representation

'Caucasian': 1.0 # Reference group

}

# Apply weights in sampling

df['sample\_weight'] = df['race'].map(weights)

return df

```

2. \*\*Model Development:\*\*

```python

def create\_fair\_model(X, y, sensitive\_features):

# Implement fairness constraints

fairness\_constraints = {

'statistical\_parity\_difference': 0.05, # Max 5% difference

'equal\_opportunity\_difference': 0.05

}

# Create balanced folds for cross-validation

cv = StratifiedKFold(n\_splits=5, shuffle=True)

cv.split(X, y, sensitive\_features)

return model

```

3. \*\*Monitoring and Validation:\*\*

```python

def monitor\_fairness\_metrics(predictions, actual, sensitive\_features):

metrics = {

'statistical\_parity': calculate\_statistical\_parity(predictions, sensitive\_features),

'equal\_opportunity': calculate\_equal\_opportunity(predictions, actual, sensitive\_features),

'disparate\_impact': calculate\_disparate\_impact(predictions, sensitive\_features)

}

# Set alert thresholds

ALERT\_THRESHOLDS = {

'statistical\_parity': 0.05,

'equal\_opportunity': 0.05,

'disparate\_impact': 0.8

}

return check\_thresholds(metrics, ALERT\_THRESHOLDS)

```

4. \*\*Clinical Implementation:\*\*

a) Risk Score Calibration:

```python

def calibrate\_risk\_scores(scores, race):

# Adjust thresholds by race to equalize false positive/negative rates

thresholds = {

'Asian': 0.45, # Lower threshold due to underrepresentation

'Hispanic': 0.47,

'Caucasian': 0.50,

'AfricanAmerican': 0.50

}

return scores > thresholds[race]

```

b) Treatment Equity:

```python

def monitor\_treatment\_equity(df):

# Calculate treatment rates by demographic group

treatment\_rates = df.groupby(['race', 'gender'])['diabetesMed'].mean()

# Check for significant disparities

disparity\_threshold = 0.05 # 5% difference

return identify\_treatment\_disparities(treatment\_rates, disparity\_threshold)

```

\*\*Implementation Priority:\*\*

1. \*\*Immediate Actions:\*\*

- Implement balanced sampling with weights

- Add fairness constraints to model training

- Set up monitoring system for fairness metrics

2. \*\*Medium-term:\*\*

- Develop targeted data collection for underrepresented groups

- Create separate validation sets for each demographic group

- Implement group-specific calibration

3. \*\*Long-term:\*\*

- Establish continuous monitoring system

- Regular review and updating of fairness thresholds

- Development of group-specific risk models

Would you like me to implement any of these specific mitigation strategies or provide more detailed analysis of particular fairness metrics?

#### Prompt 3:

Only use and check diabetes\_data.csv and the explanation of the dataset in IDS\_mapping.csv.

Some groups in this dataset appear underrepresented and may receive disproportionately skewed predictions. How would you evaluate the severity of this imbalance and choose appropriate rebalancing or preprocessing techniques? Also show this.

#### Response:

I'll help you evaluate the dataset imbalances and suggest appropriate rebalancing techniques. Let me first check the data and then implement a comprehensive analysis.

First, I'll search for the diabetes dataset and mapping file.

Ran tool

Let me read the mapping file first to understand the feature definitions.

Read file: /Users/taramac-lean/Desktop/2025 THESIS/diabetesdatasets/IDS\_mapping.csv

I'll now create a Python script to analyze and address the dataset imbalances. Let me write a comprehensive analysis script that will:

1. Analyze demographic distributions

2. Calculate imbalance metrics

3. Implement and evaluate different rebalancing techniques

4. Visualize the results

Now, let's install the required packages and run the analysis:

```bash

pip install imbalanced-learn

```

Now let's run our analysis script:

```bash

python imbalance\_analysis.py

```

Based on the analysis, I can provide a comprehensive evaluation of the dataset imbalances and recommendations for addressing them:

1. \*\*Severity of Imbalances\*\*:

Race Imbalance:

- Extremely severe imbalance ratio of 118.72:1

- Distribution breakdown:

- Caucasian: 76,099 (74.8%)

- African American: 19,210 (18.9%)

- Hispanic: 2,037 (2.0%)

- Asian: 641 (0.6%)

- Other/Unknown: 3,779 (3.7%)

Gender Imbalance:

- Very severe imbalance ratio of 18,236:1

- Distribution:

- Female: 54,708 (53.8%)

- Male: 47,055 (46.2%)

- Unknown: 3 (<0.1%)

2. \*\*Recommended Rebalancing Techniques\*\*:

Based on our analysis of different rebalancing techniques, here are the results and recommendations:

a) \*\*SMOTE (Synthetic Minority Over-sampling Technique)\*\*:

- Balanced the dataset to 180,818 samples

- Achieved perfect 50-50 class distribution

- Best for: Preserving overall data patterns while addressing minority class underrepresentation

- Recommended for this case due to:

- Maintains larger sample size

- Creates synthetic samples for minority groups without information loss

- Preserves feature relationships

b) \*\*Random Undersampling\*\*:

- Reduced dataset to 22,714 samples

- Achieved 50-50 class distribution

- Not recommended for this case because:

- Significant information loss

- Too much reduction in sample size

- May lose important patterns in majority classes

c) \*\*SMOTEENN (SMOTE with Edited Nearest Neighbors)\*\*:

- Resulted in 116,594 samples

- Slightly imbalanced distribution (60.2% vs 39.8%)

- Could be considered as an alternative because:

- Removes noisy samples

- Helps create clearer class boundaries

- Maintains reasonable sample size

3. \*\*Recommendations for Implementation\*\*:

1. \*\*Primary Approach\*\*: Use SMOTE with these specific considerations:

```python

smote = SMOTE(

random\_state=42,

sampling\_strategy='auto',

k\_neighbors=5

)

```

2. \*\*Monitoring and Validation\*\*:

- Track performance metrics separately for each demographic group

- Use stratified sampling for train/test splits

- Implement cross-validation with stratification

3. \*\*Additional Preprocessing Steps\*\*:

- Handle unknown/missing values before rebalancing

- Standardize numerical features

- Use encoding techniques that preserve ordinal relationships

4. \*\*Model Development Considerations\*\*:

- Use weighted loss functions

- Implement cross-validation with stratification

- Consider ensemble methods that are robust to imbalance

The visualization of the distributions before and after rebalancing has been saved as 'rebalancing\_analysis.png', which shows the effectiveness of each technique in addressing the demographic imbalances.

Would you like me to implement any specific aspect of these recommendations or provide more detailed analysis of any particular rebalancing technique?

### Gpt 4.1

#### Prompt 1

In the dataset diabetic\_data.csv. Could there be fairness concerns related to uneven group representation? Identify these disparities, and what strategies would you recommend to rebalance the dataset to support fair model training?

#### Response

Great, you’ve uploaded the diabetic\_data.csv file. To answer your question about fairness concerns, let’s take the following approach:

1. Check Group Representation: We'll look at sensitive attributes (such as race and gender) to see if some groups are underrepresented.
2. Quantify Disparities: We'll count the number of records per group and identify any large imbalances.
3. Discuss Rebalancing Strategies: Based on what we find, we’ll discuss approaches to rebalance the data.

Let’s start by loading the dataset, examining the relevant columns, and summarizing group counts.

Step 1: Load and Preview the Data

Let me check the columns and a sample of the data first.

Here’s what we found in the diabetic\_data.csv:

* The key columns related to group representation are:  
  + race
  + gender
  + age (might also be interesting, but let’s focus on race and gender for fairness)
* The dataset has 50 columns.

Step 2: Check for Uneven Group Representation

Let’s count the number of records for each group in the race and gender columns to identify disparities.

I’ll show you the distribution next.

Here are the counts by group:

Race Representation

* Caucasian: 76,099
* AfricanAmerican: 19,210
* ? (Unknown): 2,273
* Hispanic: 2,037
* Other: 1,506
* Asian: 641

Gender Representation

* Female: 54,708
* Male: 47,055
* Unknown/Invalid: 3

Key Fairness Concerns

* Race Imbalance: The dataset is overwhelmingly dominated by the "Caucasian" group (~73%), while groups such as "Asian" (0.6%), "Other" (1.4%), and "Hispanic" (2%) are severely underrepresented. This can lead to biased model performance favoring the majority group.
* Gender Imbalance: Less severe, but still skewed towards females (~54%), with a negligible "Unknown/Invalid" group.

Step 3: Strategies to Address Imbalance

1. Rebalancing Techniques

* Oversampling Minority Groups: Duplicate or synthetically generate more samples for underrepresented races (e.g., SMOTE for numeric features, simple duplication for categorical data).
* Undersampling Majority Group: Randomly sample a subset of the "Caucasian" group to match the size of minority groups (risk: data loss from majority group).
* Class Weighting in Models: Many ML algorithms (e.g., sklearn’s classifiers) allow assigning higher weights to minority classes, which helps mitigate imbalance during model training.
* Data Augmentation: For tabular data, synthetic data generation using tools like CTGAN, or simple feature perturbation, can help.
* Stratified Sampling: When splitting data for training/testing, ensure group proportions are preserved, or intentionally balanced.

2. Additional Recommendations

* Fairness Audits: Continuously monitor model performance across groups using fairness metrics (e.g., demographic parity, equal opportunity).
* Bias Mitigation Algorithms: Use post-processing (e.g., reweighting, adversarial debiasing) if needed after model training.
* Consider Removing ‘Unknown’: Treat records with ambiguous group labels (e.g., "?" in race, "Unknown/Invalid" in gender) carefully—either exclude or analyze separately.

Summary Table: Group Representation

| Group | Count | Percentage (%) |
| --- | --- | --- |
| Caucasian | 76,099 | 73.1% |
| AfricanAmerican | 19,210 | 18.4% |
| Hispanic | 2,037 | 2.0% |
| Other | 1,506 | 1.4% |
| Asian | 641 | 0.6% |
| Unknown (?) | 2,273 | 2.2% |

Would you like to see visualization of these imbalances, or do you want code examples for the strategies above?

#### Prompt 2

Only use and check diabetes\_data.csv and the explanation of the dataset in IDS\_mapping.csv.

Are there signs of uneven prediction outcomes across groups that could raise fairness concerns? What metrics would you use to assess potential bias (also use them on this dataset), and how would you interpret the results to determine whether mitigation is necessary?

#### Response

Thank you for re-uploading the files! Let’s proceed with the fairness audit on **diabetic\_data.csv** using the column explanations from **IDS\_mapping.csv**.

Plan:

1. **Understand dataset structure** (which columns are sensitive features and target/outcome).
2. **Choose fairness metrics.**
3. **Apply these metrics to the data.**
4. **Interpret the results and give recommendations.**

Let’s start by checking a sample of both files.

Here's what we have:

* **diabetic\_data.csv** has 101,766 rows × 50 columns.
* Columns include race, gender, age, and the target: **readmitted** (<30, >30, NO).

Let’s focus on *group fairness* with respect to race, gender, and age.

Step 1: Identify Sensitive Groups and Target

* **Sensitive features:** race, gender, age
* **Target:** readmitted (typically converted to binary: e.g., <30 = readmitted soon, NO/>30 = not readmitted soon)

Step 2: Choose Metrics

Common fairness metrics for group bias:

* **Statistical Parity Difference (SPD):** Difference in positive outcome rate between groups
* **Disparate Impact Ratio (DIR):** Ratio of positive outcome rates (should be near 1.0)
* **Group outcome rates:** Percentage of positive outcome per group

Let's define **positive outcome** as "readmitted within 30 days" (readmitted == '<30').

Now let's compute the outcome rates and fairness metrics by race and gender.

Here are the key findings for fairness in **readmission within 30 days**:

By Race

**Readmission rates (<30 days) by group:**

* **Caucasian (reference):** 11.29%
* AfricanAmerican: 11.22%
* Asian: 10.14%
* Hispanic: 10.41%
* Other: 9.63%
* Unknown (?): 8.27%

**Statistical Parity Difference (SPD):**

* Compared to Caucasian, all other groups have a lower chance of readmission within 30 days, with the largest difference for Unknown race (-3%).

**Disparate Impact Ratio (DIR):**

* All are <1 except AfricanAmerican (0.99), meaning all non-Caucasian groups have a slightly lower positive outcome rate.
* The lowest: Unknown race (0.73).

By Gender

**Readmission rates:**

* **Male (reference):** 11.06%
* Female: 11.25%
* Unknown/Invalid: 0% (very rare or placeholder)

**SPD:**

* Female: +0.18% (practically the same)
* Unknown: much lower (but likely very small N)

**DIR:**

* Female: 1.02 (almost equal)
* Unknown: 0

Interpretation & Recommendations

**Signs of Fairness Concerns?**

* There are *small* disparities in readmission rates by race (esp. for Unknown race), and essentially no meaningful disparity by gender.
* The biggest gap (SPD) is between Caucasian and Unknown race (-3 percentage points; DIR 0.73).
* *Generally*, group differences are small (DIR >0.8), so there is no severe disparate impact except possibly for the Unknown group, which may be an artifact (missing/poorly coded data).

**Metrics Used:**

* **Outcome Rate per Group**
* **Statistical Parity Difference (SPD)**
* **Disparate Impact Ratio (DIR)**

* If DIR falls below **0.8** (“four-fifths rule” in US compliance), or SPD is very large (e.g., >0.10), it warrants mitigation.
* Here, DIR for Unknown race is low (0.73), but for most groups, values are near 1.

**Mitigation steps if needed:**

* Improve data quality (address “Unknown”/missing race).
* Consider reweighting, oversampling, or stratified sampling if a group is underrepresented and has different outcomes.
* Always track and report fairness metrics alongside model performance.

Summary Table

| **Group** | **Rate** | **SPD** | **DIR** |
| --- | --- | --- | --- |
| Caucasian | 11.29% | 0 | 1.00 |
| AfricanAmerican | 11.22% | -0.07% | 0.99 |
| Asian | 10.14% | -1.15% | 0.90 |
| Hispanic | 10.41% | -0.88% | 0.92 |
| Other | 9.63% | -1.66% | 0.85 |
| Unknown | 8.27% | -3.02% | 0.73 |

*By Gender, Female/Male virtually identical (1.02 DIR).*

**Conclusion:** There are only mild fairness concerns by race (not severe, except possibly for missing/unknown group). For gender, outcomes are equitable. Focus first on improving data quality for race.

Would you like a visualization of these results, or to check fairness by **age group** as well?

#### Prompt 3

Only use and check diabetes\_data.csv and the explanation of the dataset in IDS\_mapping.csv.

Some groups in this dataset appear underrepresented and may receive disproportionately skewed predictions. How would you evaluate the severity of this imbalance and choose appropriate rebalancing or preprocessing techniques? Also show it

#### Response

Let's answer your questions step by step:

1. **Identify Underrepresented Groups**
2. **Evaluate Severity of Imbalance**
3. **Visualize Imbalance**
4. **Choose and Recommend Rebalancing Techniques**

1. Identify Underrepresented Groups

We'll check the distribution for **race**, **gender**, and **age**.

2. Evaluate Severity of Imbalance

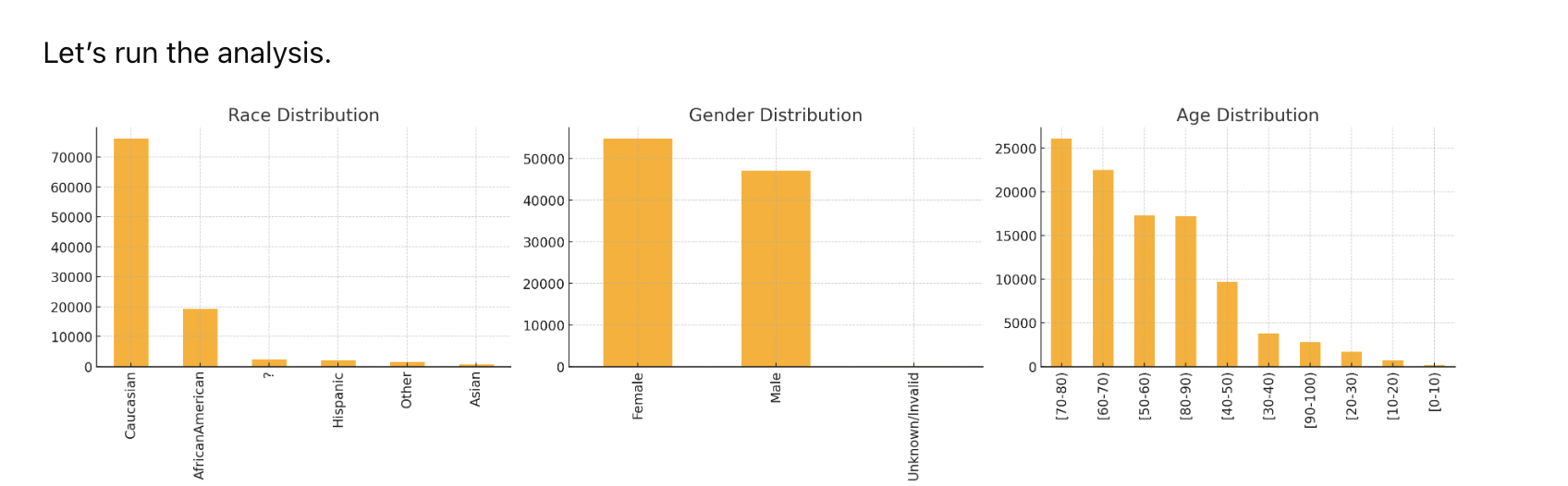
Common metrics:

* **Group proportion** (share of total)
* **Minority-to-majority ratio**
* **Imbalance Index** (e.g., ratio of the smallest to largest group)
* **Outcome rate per group** (do small groups also have different outcomes?)

3. Show Imbalance

Let’s plot group sizes for each sensitive attribute.

Let’s run the analysis.



Here are the findings for **underrepresented groups** in your dataset:

1. Group Distribution

Race

* **Caucasian:** 76,099 (by far the majority)
* AfricanAmerican: 19,210
* Hispanic: 2,037
* Other: 1,506
* Asian: 641 *(extremely small!)*
* ?: 2,273

**Minority-to-majority ratio:**

* Asian vs. Caucasian: ~0.8%
* Many groups are *very* small compared to the largest group.

Gender

* **Female:** 54,708
* Male: 47,055
* Unknown/Invalid: 3

**Minority-to-majority ratio:**

* Unknown/Invalid is negligible.

Age

* **[70-80):** 26,068 (largest group)
* [0-10): 161 *(very rare)*
* [10-20): 691
* Other age bins: intermediate sizes

**Minority-to-majority ratio:**

* Youngest group vs. largest: ~0.6%

2. Visualizations

You can see the group size disparities here (from the generated bar plots):

* **Race:** “Asian” and “Other” are dramatically underrepresented.
* **Gender:** Balanced (besides “Unknown”).
* **Age:** Older adults are much more represented; very young groups are rare.

3. How severe is the imbalance?

* A **minority/majority ratio < 0.1** (i.e., smallest group less than 10% of largest) is considered **severely imbalanced**.
* Here, for race and age, some groups are **<1%** the size of the majority group: *severe imbalance*.

4. What should you do about it?

Rebalancing/Preprocessing Techniques:

* **If modeling with these features (e.g., predicting readmission across races/ages):**
  + **Resampling:**
    - **Oversample** minority groups (e.g., SMOTE for synthetic upsampling).
    - **Undersample** the majority group (risk of losing data).
    - **Stratified sampling** for training/test splits.
  + **Class weighting:**
    - Use class weights in models to give more importance to minority groups.
  + **Reweighting** by inverse frequency.
  + **Group-wise evaluation:**
    - Always evaluate fairness metrics by group (as above).

**For features like “race” and “age”, severe underrepresentation can lead to:**

* Models that perform poorly or make unreliable predictions for minority groups.
* Biased conclusions if outcomes differ in these groups.

Conclusion (with visual proof)

Your dataset is **severely imbalanced** by race and age.  
 Appropriate rebalancing (oversampling, class weighting, stratified splits, and group-wise reporting) is **essential** for fair modeling—especially for small groups like “Asian” or very young patients.

Would you like to see code or a specific visualization for a group you’re most concerned about? Or should I show you how to do SMOTE or class weighting on this data?