Class Imbalance

Scenario: Tinder

The dating scene in 2018 has been quite a struggle. People don’t get you no matter how hard you try. You decided to do what any regular person would do: join Tinder! Tinder is on this mission with you to find prospects. However, they notice that you’re quite picky: it looks like you swipe right for only a few of Tinder’s suggested prospects. The app company wants to build a better machine learning model that is able to provide prospects that seem to match your taste. There appears to be a problem: the model cannot really tell who’s likely to pique your interest based on your Tinder data available. Check out the next slide!

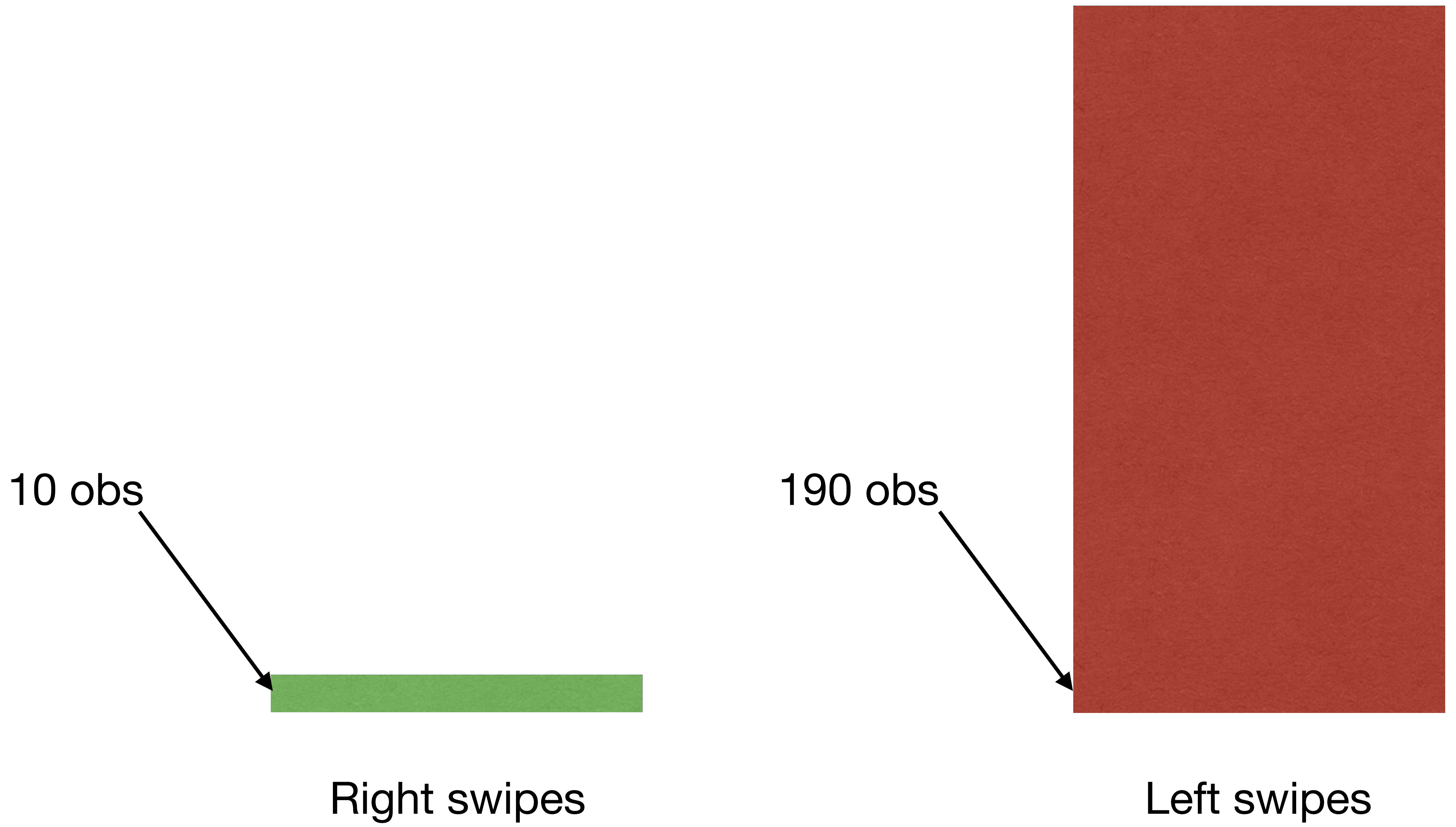
Your Tinder Data

S/N	Age	Cute	Degree	Smoker	Animal	Swipe
1	27	Yes	Masters	Yes	Dog	Right
2	35	Yes	Bachelor	No	Dog	Left
3	20	No	High School	Yes	Dog	Left
4	23	No	PhD	No	Cat	Left
5	19	Yes	High School	No	None	Left
6	35	No	PhD	No	None	Right
7	42	Yes	Masters	Yes	Cat	Left
8	21	No	Bachelor	No	None	Left
9	37	No	Bachelor	No	Dog	Left
10	25	Yes	Bachelor	No	Fish	Left
…	…	…	…	…	…	…
200	22	No	None	Yes	Cat	Right

Assumptions

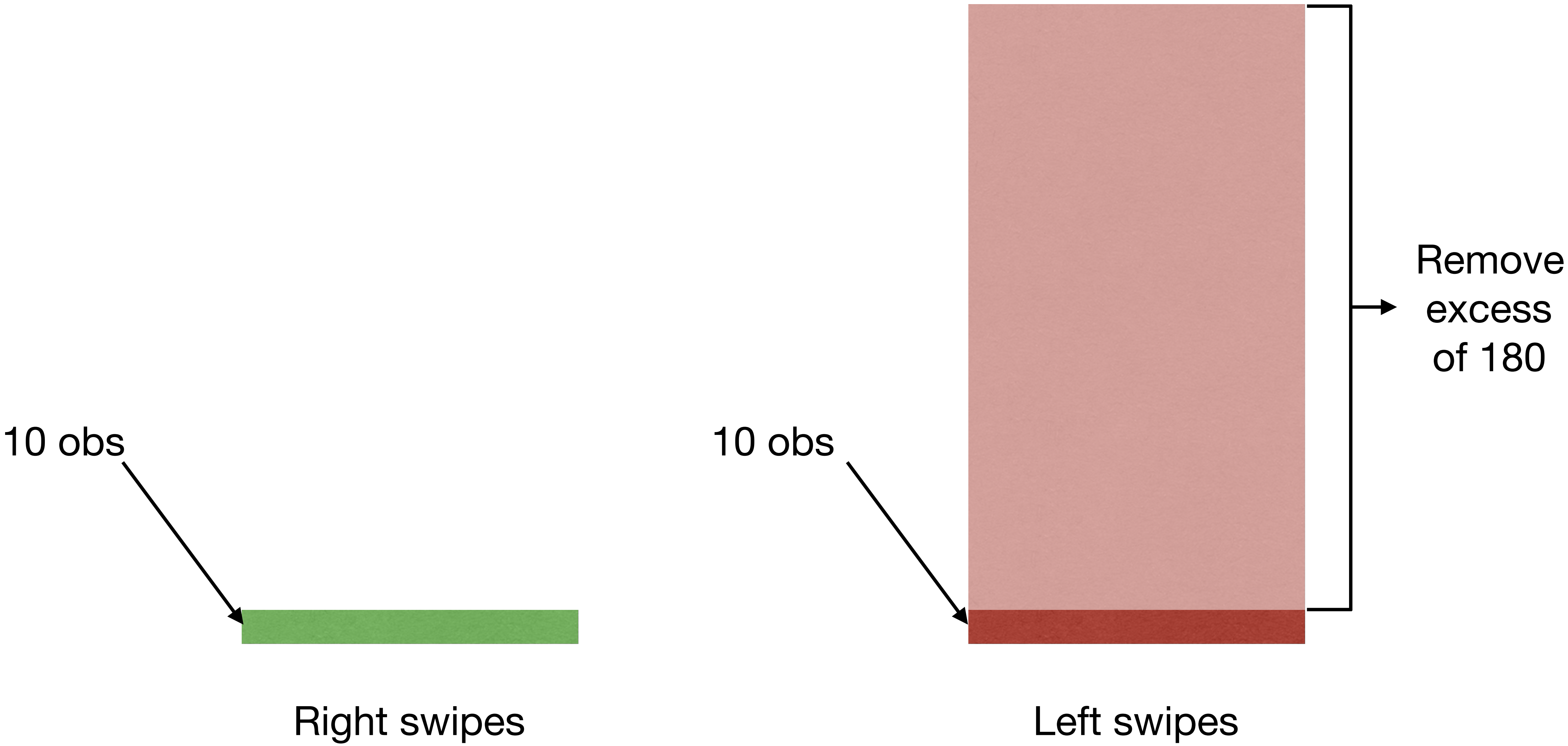
- Tinder is able to collect and store all the information in your dataset (yes, Tinder can tell if you think someone is cute).
- The data set have 200 observations (meaning you swiped 200 times)
- You have 10 right swipes in your data set and 190 left swipes

Imbalanced Right and Left Swipes



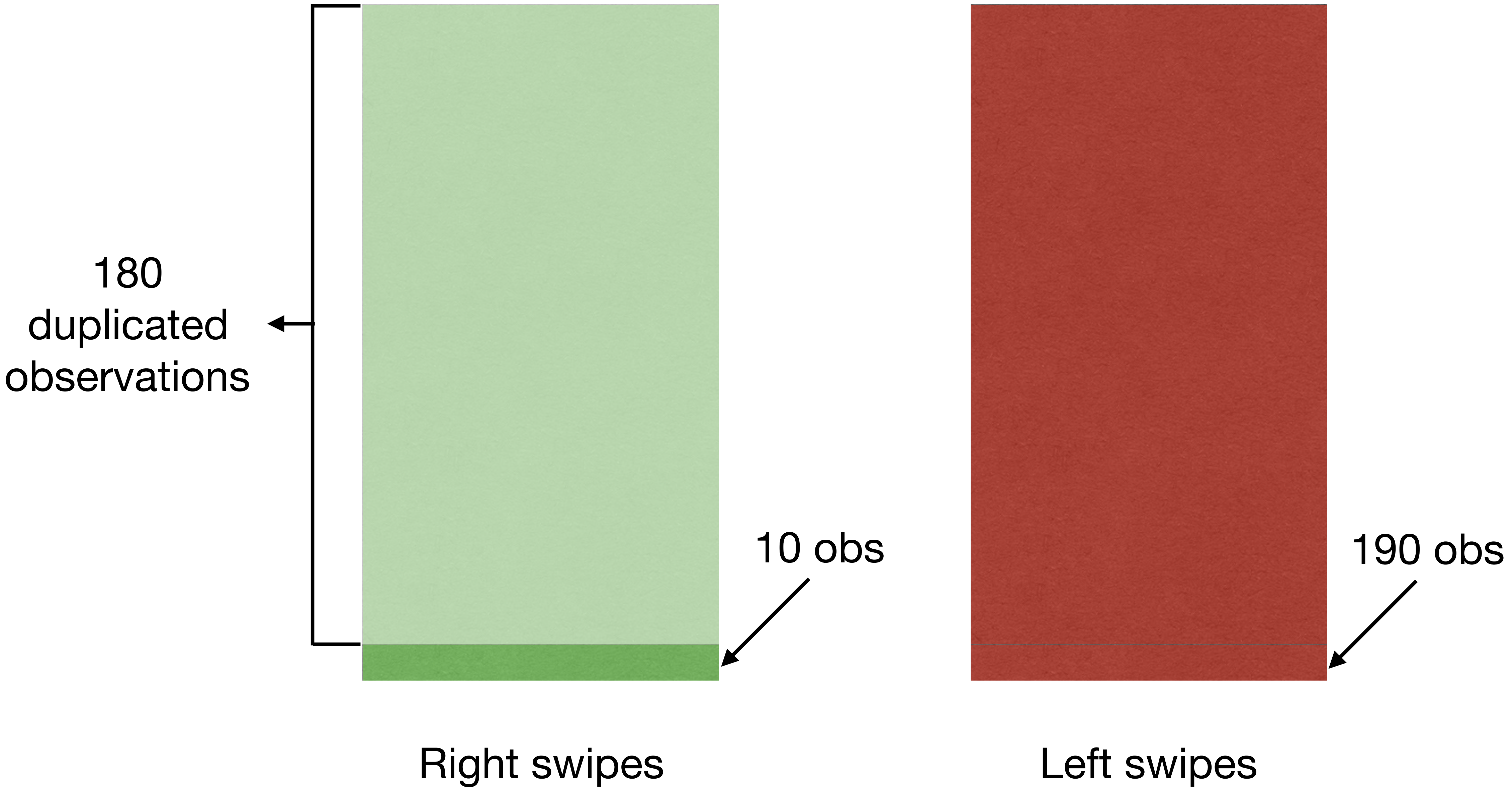
Undersampling

Randomly *remove excess observations from the left swipes so the data is balanced with 10 observations from each class*



Oversampling

Randomly *select an observation from the right swipes, duplicate, and then replace back into the the right swipes*



Conclusion

- With undersampling, you lose information because you’re literally dropping data
- With oversampling, you’re not really gaining additional information from resampling
- Other ways to fix class imbalance would be:
    - Change your performance metrics (e.g. instead of using accuracy to measure performance, you can use precision and recall)
    - Try multiple models
    - Try adding penalties to your models
    - Collect more data
    - When dealing with more than 2 classes, combine minority classes