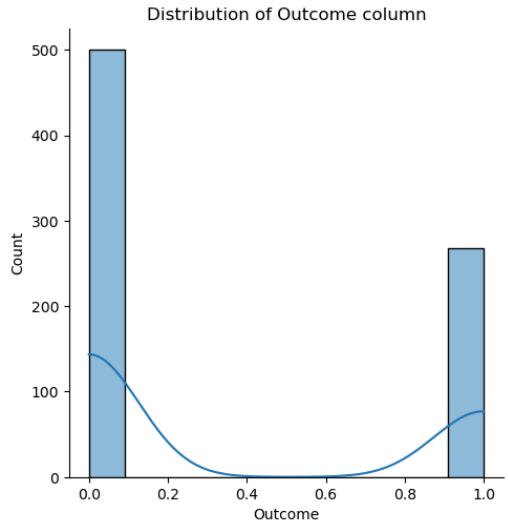
When asked to perform Supervised Learning, especially a binary classification, logistic regression is probably the first thing that pops in my head. But should it be?

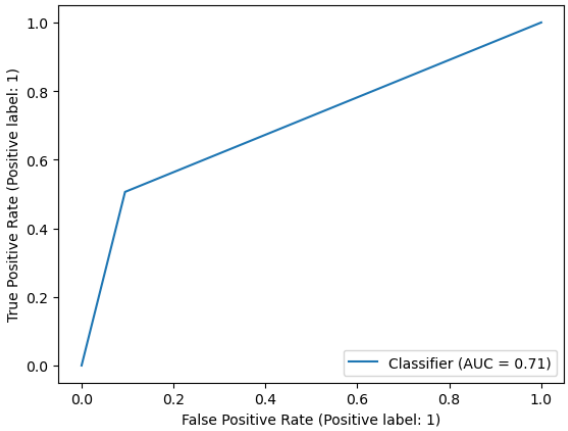
When performing EDA we can clearly identify that the outcome class is imbalanced,
That there are more 0 values than 1, and therefore when training it upon this dataset, the model will be biased and favourable to outcome 0.



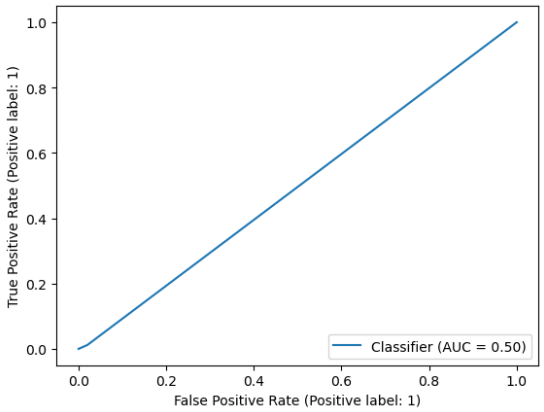
I also thought of a new feature to engineer, a variable that captures glucose divide by insulin, because too high glucose is only a problem because its a sign of low insulin, so glucose/insulin will help the model differentiate between those with high glucose and low insulin, and those with high glucose and not low insulin.

Performing a logistic regression without remedying the imbalance, we see that the model performs moderately , with metrics as below:

Accuracy: 0.7619047619047619
Precision: 0.75
Recall: 0.5060240963855421
F1-score: 0.6043165467625898
ROC-AUC: 0.7057147508954738



But once we oversample the minority class (outcome 1) so there is a 1:1 ratio, we see that the metrics actually worsen, and prediction act like a random classifier.



Accuracy: 0.6320346320346321
Precision: 0.25
Recall: 0.012048192771084338
F1-score: 0.02298850574712644
ROC-AUC: 0.495888961250407

This can be attributed to how it predicts more of outcome 1 than the data where we didn't oversample. See because we removed bias that overfitted the outcome to a value of 0 doesn't mean that metrics will improve, for example say we didn't observe any outcome 1 in the test set, and the model was unfairly favoured to outcome 0 we can theoretically have an observed accuracy of 100% but we know it would be severely inaccurate. The point is there is no relationship between high metrics and an unbiased model.

One model that actually performs better and could account for imbalanced class is random forest classifier. Inputting the imbalanced data we get :

Accuracy: 0.7662337662337663
Precision: 0.6986301369863014
Recall: 0.6144578313253012
F1-score: 0.6538461538461539
ROC-AUC: 0.7329045913383263

Confusion Matrix:
[[126 22]
[32 51]]

Which performs better than logistic regression but still contains bias, inputing resampled data to a random forest classifier we get a model that performs like a random classifier

Accuracy: 0.6406926406926406
Precision: 0.0
Recall: 0.0
F1-score: 0.0
ROC-AUC: 0.5

Confusion Matrix:
[[148 0]
[83 0]]

There is a number of possible reasons for this horrible performance, it could be attributed to Insufficient Information in the Minority Class, or Data Leakage when oversampling, but further investigation is needed for it to be determined. Thankfully Random forest classifier has a built in method to remedy class imbalance, one that does not oversample but assigned different cases weights, using this model with the imbalance data we get

Accuracy: 0.7705627705627706
Precision: 0.7205882352941176
Recall: 0.5903614457831325
F1-score: 0.6490066225165563
ROC-AUC: 0.730991533702377

Confusion Matrix:
[[129 19]
[34 49]]

This is the best model we have observed so far, and that's not only because it accounts for the class imbalance bias but also performs well and has significant better metrics than other models that accounted for bias.

To sum up, in this scenario a random forest classifier and using class weights to remedy class imbalance outperforms logistic regression and/or oversampling the minority class.

And if there is one lesson I can make you take from this, let it be that remedying class imbalance is not one solution for all, there are many approaches, and each one has their advantages and limitations. Their effectiveness may vary depending on the dataset and problem at hand. And working with this dataset, applying class weights rather than oversampling the minority class was more suitable.