Exploring the Impact of Enhancing Fairness on Model’s Explainability

Trustworthy Autonomous Systems – final project

Group: Ambrosetti, Biddiscombe, Sedra

1. Introduction

The idea behind this course is to realise how the different aspects of a machine learning (or in general AI) model can influence the way engineers use it, not only in expected prediction precision but also through trust and fairness. In this specific project, we will be looking at a model and the impact of balancing for fairness on its performance and explainability, to further understand the meaning of making a model fair, and determine use cases where this may be useful or an issue, depending on the model’s goal. To perform the experiment, we will be predicting the risk associated with granting credit to an individual.

2. Data

2.1 Dataset Description

For this project and this report, we are using a German dataset centred on the idea of the risk associated with giving credit to a person. It consists of 20 features and a target with 1000 data entries, which are initially not well labelled but represent various descriptions of the person asking for credit, from age to job to savings, and culminate in a final target decision of whether the person is risky or not. This data appears largely encoded in the raw format, with both feature names and data entries replaced with shorter but less understandable acronyms, which were fortunately available in the raw data itself.

2.2 Data Preprocessing

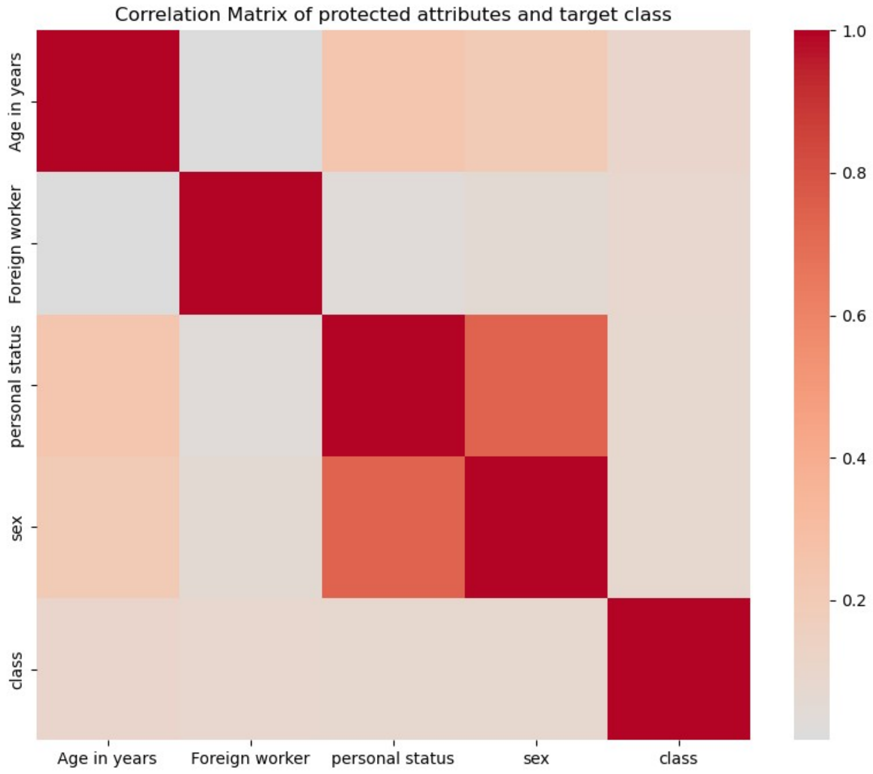
As mentioned above, the initial data is disorganised, as in the feature names are not present in the columns of the dataset, and the meanings of each instance of each feature are unclear. For legibility, we replaced the column (feature) names with more descriptive ones the dataset provided, and replaced the encoded values with their meanings. Even though we later one-hot encode all categorical data, this provides a more understandable starting point from which we can analyse the model. Next, we remade the attributes in a more convenient way for us, as in the raw data you can find attributes such as “Personal status and sex: male – single” and all equivalent variants of that, we created 2 new columns to instead represent the sex and personal status separately. This is not only for model clarity, but also for batter use during the bias mitigation, as both these classes could be considered protected attributes. After this we proceeded with one-hot encoding, we created a whole set of new features, ending up with 57 features and 1000 entries. When checking for the feature names, we found there are no missing values so there was no need to impute or removing missing data. Out of habit, we also scaled and normalised the data, which is generally considered good conduct even if for the models we intend to use (random forest and XGBoost) it is not completely necessary.

3. Exploratory Data Analysis

First of all we counted the distribution of the target class and found that it is unbalanced, favouring the side of “not risky” when talking about giving credit to an individual.

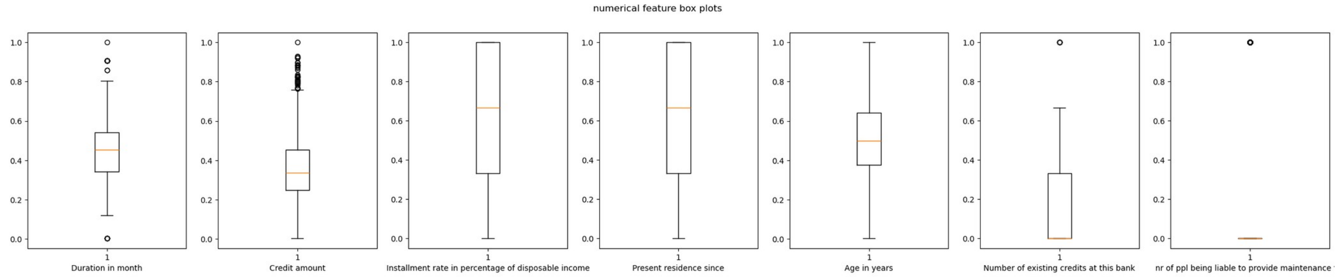
Knowing the focus of the project, we chose a set of features we could use as protected attributes further in the project, using our own morals and that which we have learned in class to help our decisions. The list includes the age of a person being analysed, whether they are foreign, and their marital status and sex.

We mostly analysed the protected attribute list, as these are the most interesting features to know how they could influence the model and its fairness.



We explored the distribution of the newly scaled data, the correlations between features and target class, and found there are no missing values in the data.

See below an image showing the distributions of numerical features in the dataset.



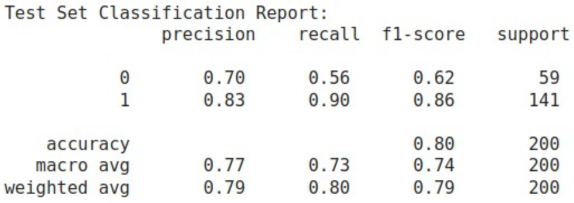
As a general discovery, we noticed most of the data represented males and there are very few foreign workers, which our lectures have taught us are not great recipes for fairness in a basic model trained for performance.

4. Modelling

4.1 Initial model development

After dividing the data into training and testing sets, we created a few different model iterations using Random Forest from scikit-learn, XGBoost and MLPClassifier from scikit-learn as the models, and gridsearch for parameter optimisation. Using gridsearch allowed to steamroll the fitting process, trying 810 different models with different parameters and settling on an XGBoost model for best performance.

The resulting best model gives us an accuracy of 80% over the testing data.



```
Test Set Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.56      0.62        59
           1       0.83      0.90      0.86       141

    accuracy                           0.80       200
   macro avg       0.77      0.73      0.74       200
weighted avg       0.79      0.80      0.79       200
```

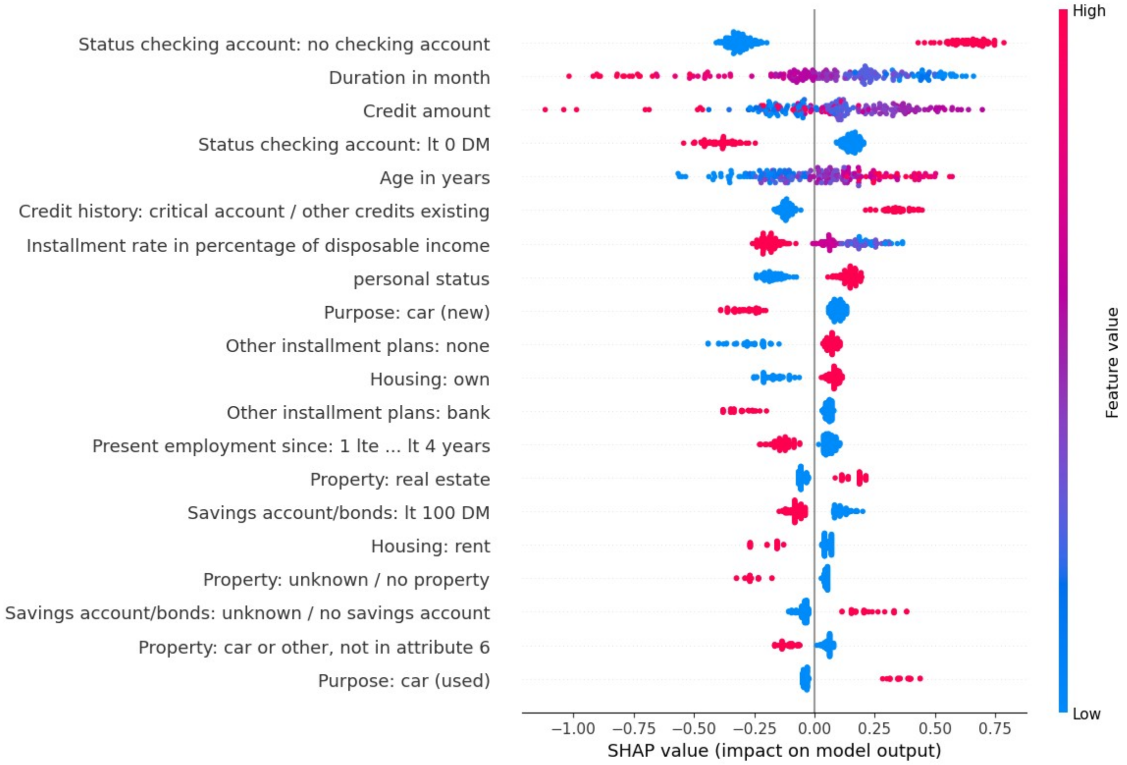
4.2 Explainability of the Baseline Model

Using Shap we found the most influential features output decision, and were surprised to find “Age in years” (which we defined as a soon-to-be protected attribute) as the third most influential feature in the dataset, following a very linear trend where the higher the feature value, the more positively it impacts the final risk prediction, whereas the lower the age, the more negative impact on the outcome. The second most impactful feature we designated as protected represents single male men. Interestingly, an opposing find was that “Foreign worker” has almost no impact at all on the decision of the model in the current state, and took the place of least impactful soon-to-be protected attribute.

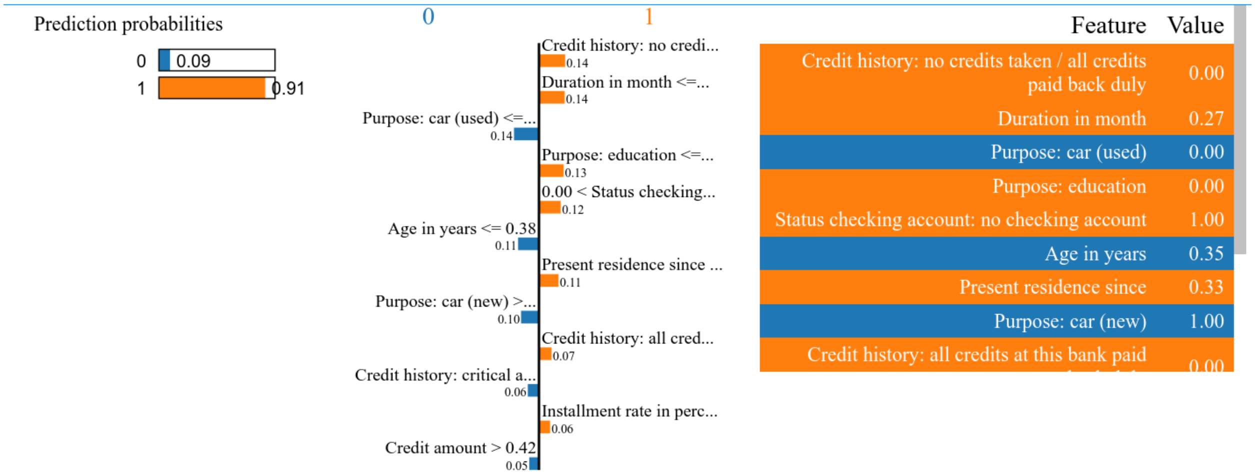
We explored some local model decisions, focusing on the impact of the features we would protect versus the others.

Along with Shap, we also used Lime to explain some of the behaviours of the model in specific cases, providing detailed local explanations about model decisions, which showed us that the most consistent feature in the few local cases we studies was credit history, something we have come to expect from in class lectures and experiments.

Below is the Shap summary plot of the initial model.



Below, a Lime local explanation for a specific instance of the data.



Feature	Value
Credit history: no credits taken / all credits paid back duly	0.00
Duration in month	0.27
Purpose: car (used)	0.00
Purpose: education	0.00
Status checking account: no checking account	1.00
Age in years	0.35
Present residence since	0.33
Purpose: car (new)	1.00
Credit history: all credits at this bank paid	0.00

4.3 Enhancing Fairness

4.3.1 Defining the metrics

Using fairlearn, an open-source, community-driven software for enhancing fairness of AI systems, we chose to enhance the XGBoost model on the metrics demographic parity difference and equalised odds. Both metrics help check if an AI system treats different groups fairly.

Demographic parity difference looks at the difference in probability across the categories of a feature (or its demographic) of receiving a positive outcome, which we ideally want to be as close to zero as possible. We can also read this as the difference in selection rates across the bins of a certain feature.

Equalised odds difference, similarly, looks at the similarity in true positive, true negative, false positive and false negative counts across a feature's bins. Unlike demographic parity though, equalised odds focuses on the relationship between predictions and true labels. We also want this metric to be ideally zero.

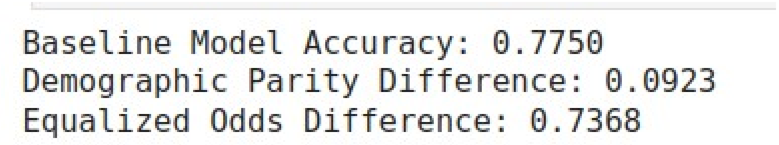
4.3.2 Choosing the Protected Attributes

As we previously mentioned, we created an initial list of 4 attributes to use as protected attributes, according to which to also enhance fairness. Because of the complexity of working with multiple attributes (and because we tried multiple times but it did not go well), we selected only one attribute according to which to actually enhance the fairness of the base model.

We decided to choose the most important feature from the list, that representing **age** ("Age in years"), for one of the models, and the two most impactful features **age** and **personal status** for the rest of the models. This will hopefully force the unbiased model to grant equal chances at success to all ages, and since we chose the most impactful of the possible protected attributes we hope to exacerbate any differences in the model we would see with normal fairness additions, in the hope of studying further the effects of balancing fairness on performance and explainability.

For fairlearn to work correctly, we cannot feed it a feature with continuous values, so we binned the values of age into 4 categories. Looking at their distributions, we notice the first category is rather sparsely populated, so it might have some interesting effects (we expect negative ones) on the total performance, as we will be forcing the model to balance in a certain direction as well as try to not consider the third most important attribute in scoring the target.

The initial scores of the base model are:

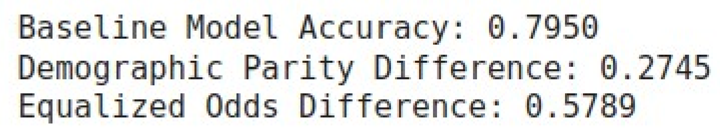


```
Baseline Model Accuracy: 0.7750
Demographic Parity Difference: 0.0923
Equalized Odds Difference: 0.7368
```

4.3.3 Fairness through Unawareness

The first fair model we trained was the one following the “fairness through unawareness” technique, so by completely omitting the protected attribute from training, and only readding it afterwards. As before, we added a gridsearch to give us the best model using this technique.

The scores of the unaware model are:

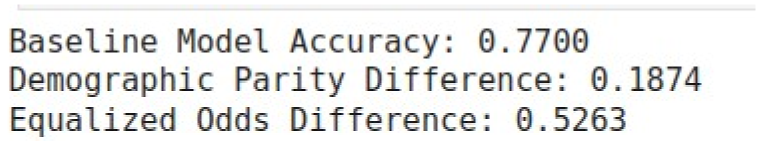


```
Baseline Model Accuracy: 0.7950
Demographic Parity Difference: 0.2745
Equalized Odds Difference: 0.5789
```

4.3.4 Fairness in Model Training (pre-processing)

For our other fair model, we trained a model for accuracy, using gridsearch, but with the addition of a mitigator that uses equalised odds as another direction for improvement of the model.

With this method we obtained the scores:

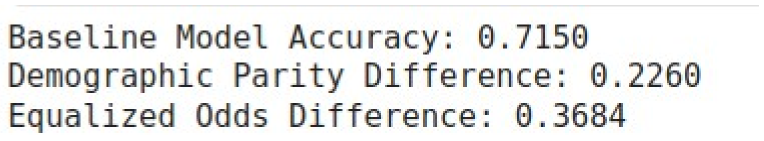


```
Baseline Model Accuracy: 0.7700
Demographic Parity Difference: 0.1874
Equalized Odds Difference: 0.5263
```

4.3.5 Fairness with Threshold Optimised (post-processing)

The last fair model uses techniques of balancing fairness after training, threshold optimiser, always using gridsearch in an attempt to optimise performance, an even more important step in the case of this model because as we can read on the fairlean website (https://fairlearn.org/main/user_guide/mitigation/postprocessing.html), this mitigation technique “is built to satisfy the specified fairness criteria exactly and with no remaining disparity” but “In many cases this comes at the expense of performance, for example, with significantly lower accuracy”. For this reason, we try and balance the fairness and predicting power of this model even more.

The final model evaluation comes out to:



```
Baseline Model Accuracy: 0.7150
Demographic Parity Difference: 0.2260
Equalized Odds Difference: 0.3684
```

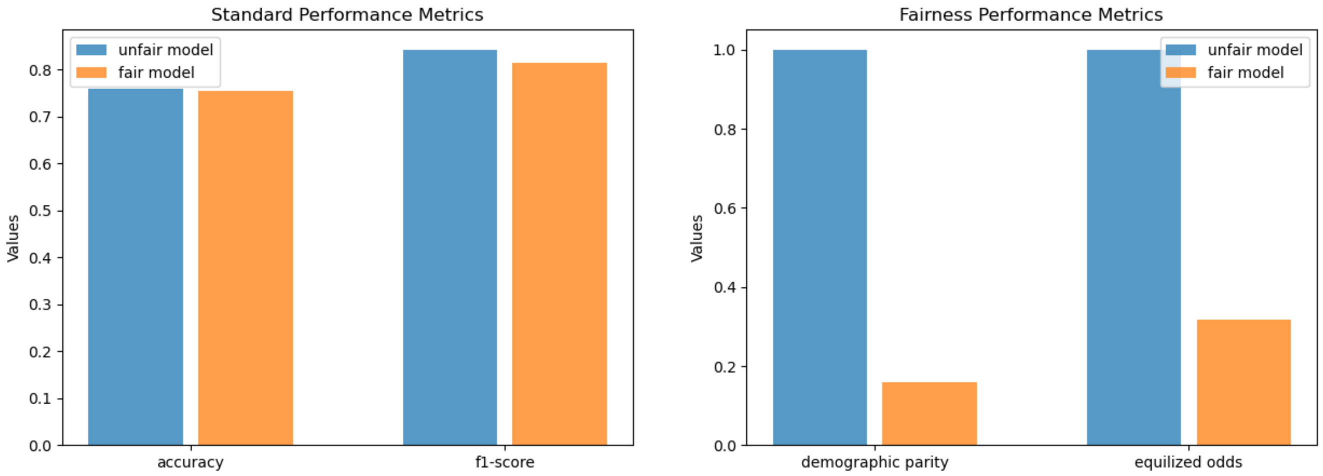
4.3.6 Comparison

To compare the models before and after applying the fairness or bias mitigation techniques, we show the difference in accuracy and f-1 score of the base model and the best fair model, and the difference in their fairness metrics, demographic parity difference and equalised odds difference. We chose one fair model to represent the whole concept, opting for the one with the largest difference in fairness, obviously expecting at least some level of drop in the accuracy.

We chose the model balanced post training, the Threshold Optimiser one, for its very good fairness, even though the accuracy witnessed a drop.

Below you can find the comparison graph.

Thanks to this visualisation we can see that we gain a similar amount in the fairness metrics as that which we lose in the accuracy and f1-scores.

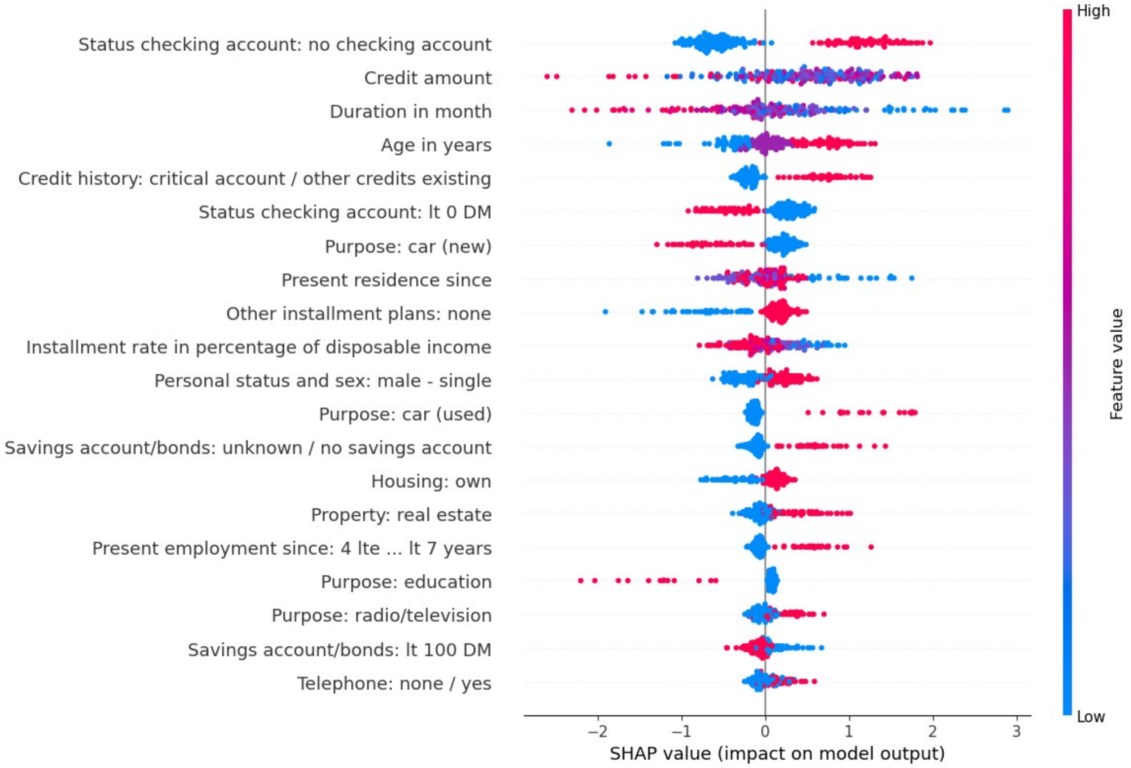


4.3.7 Explaining the Fair Model

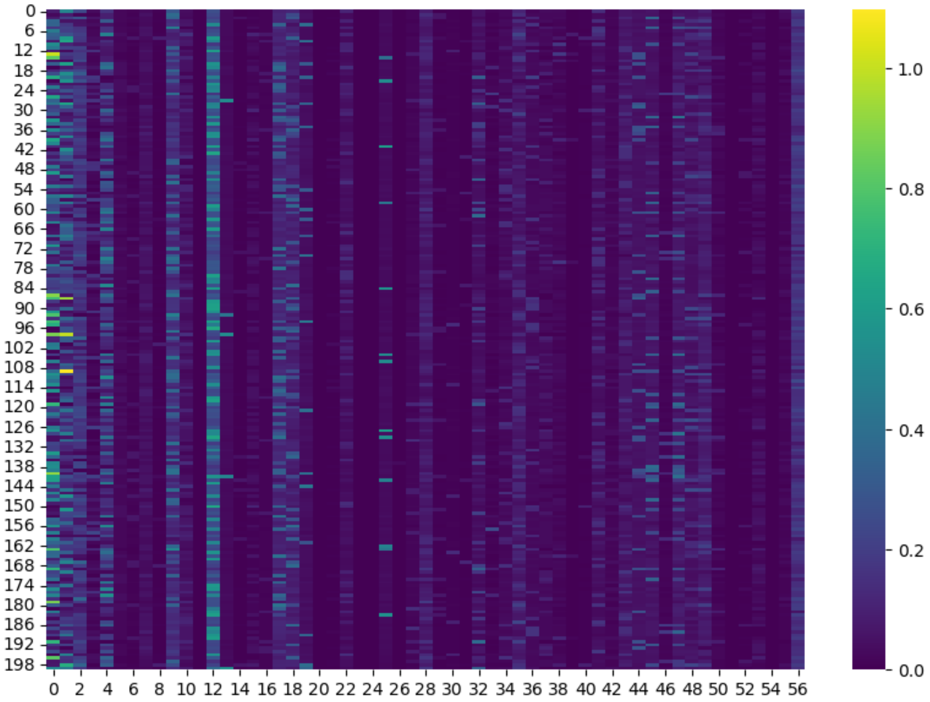
After applying fairness enhancing techniques, we are curious to see if not only the scores changed but also how different the explanations given to us by Shap turn out to be.

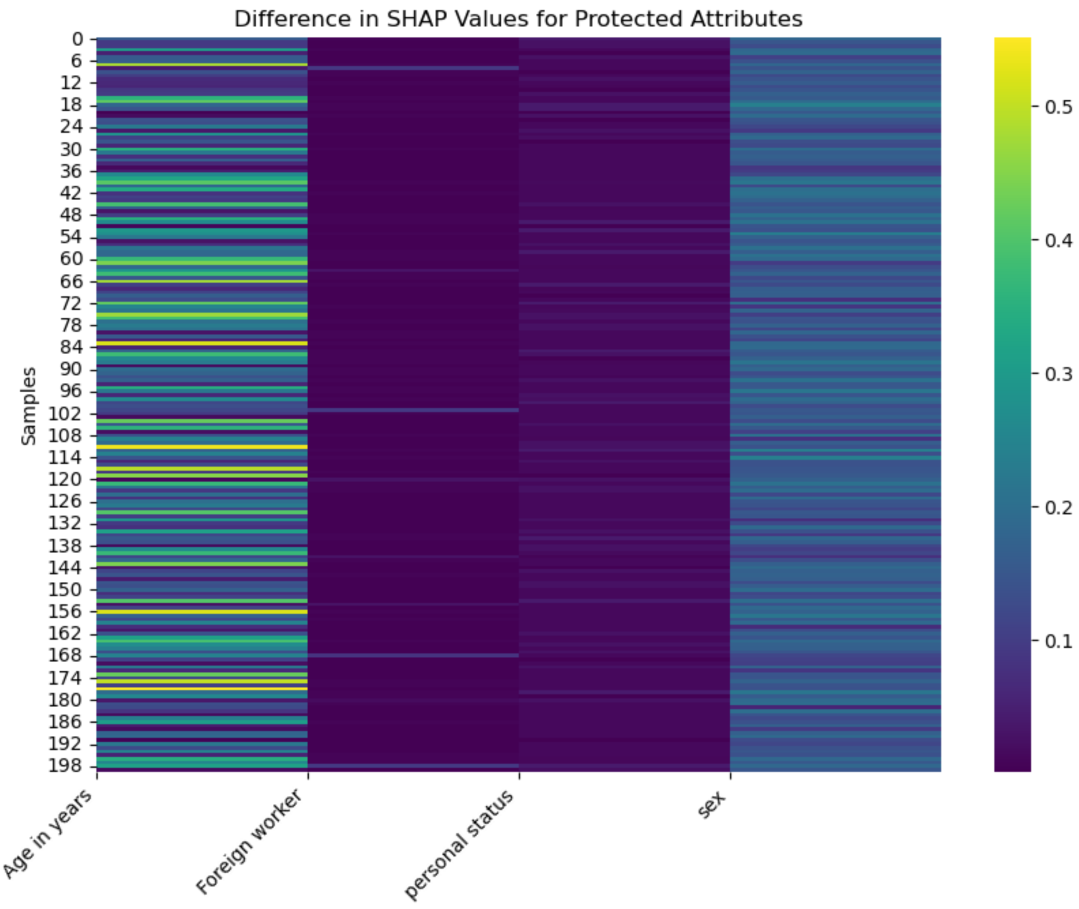
When looking at the summary plot outputted by Shap, we notice that age still factors into the choice, in a large way. If we are to believe that the techniques used to enhance fairness removed its influence on the model and its output, such as in the case of fairness through unawareness, the only thing left to believe is that other features must be highly correlated with age and therefore the model gets influenced in the same way, and a resulting Shap exploration of the model would tell us that age does in fact have an influence on the model. In this case though, we used a model that does not fully remove the feature relating to age, and therefore should be able to better steer the model away from making decisions based on that attribute even if it does in fact correlate to the rest of the data. This may be the reason that age now factors in as 4th instead of 3rd most important feature, but we would have expected a larger change. Our last assumption is that it might be because of the need to approximate this feature to fit it into bins as opposed to using a binary variable, which may be rendering the model less capable of becoming more fair.

Below is the summary plot for the model with fairness enhanced in age.



The total changes in features are showed in the following matrices, showing at first the total differences and finally a random selection of points and how their feature importance changes.





5. Conclusion

After applying fairness enhancing methods to our model that assesses the risk of granting credit to individuals, and using age as a protected attribute, we noticed that the ratio in the score gained in fairness metrics is similar to the accuracy lost by that same model. We can only conclude from this that fairness costs a lot, but certainly enough for the trust gained in the capacity of the model to make good decisions to pay for it.