Classification

Robbe Nooyens

4 Mei 2024

1 Introduction

In this report, we will evaluate machine learning models designed to predict income levels based on a dataset of various socio-economic factors. Our primary focus is to ensure model fairness while maintaining high predictive accuracy. Initially, we will analyze the dataset to identify features that may introduce bias into the predictions. Based on this analysis, we will implement several models, including decision trees, KNN classifiers, and ensemble methods such as a random forest and boosting algorithms. The best model, the one that is both highly accurate and fair, can then make predictions on a new dataset.

2 Data inspection

First of all, we’ll take care of missing data in the dataset. I chose to replace absent ’gave birth’ entries with ’No’ and blank ’ability to speak English’ fields with 0, unlike the pattern mining assignment where I used ’N/A’. Using ’N/A’ previously helped to prevent irrelevant patterns form showing up, whereas in this context, it might uncover potentially significant relationships.

Before constructing our models, it’s crucial to scan the dataset for potential risks of bias. A pivot table of gender versus income, accessible in the ’assignment2_income.xlsx’ file as additional sheet, reveals a first disparity: 80% of females versus 60% of males are categorized under low income. To mitigate explicit gender bias in income estimation, we can exclude both the ’gender’ and ’gave birth’ columns. In the previous assignment, I replaced *Husband* and *Wife* with *Married* and *Divorced*, *Widowed*, and *Never married* with *Single*. For this assignment, we can further specify the *Single* status as it is uniform across both genders, but we will group *Husband* and *Wife* as *Married* again to reduce gender bias. To identify more nuanced biases, we can revisit the patterns mined previously. By filtering the pattern rules to only include those implicating *Male* or *Female* and sorting them by confidence (figure 1), I identified potential biases. Notably, occupations and traits like overtime and lack of higher education frequently correlate with males. To assess bias in the occupation feature comprehensively, we can plot the differences in gender proportions within each occupation (figure 2). It is clear that someone’s gender can often be derived from their occupation.

Sup	**Con**	**Items Base**	**Implies**
0.07	0.98	Construction/Extraction	Male
0.05	0.97	Construction/Extraction, Low Income	Male
0.05	0.92	No higher education, Overtime, High Income, Married	Male
0.05	0.91	Private, No higher education, Overtime, High Income	Male
0.07	0.90	Private, Overtime, High Income, Married	Male
0.07	0.90	No higher education, High Income, Married, Seniors	Male
0.06	0.90	No higher education, Overtime, High Income	Male
0.05	0.90	Overtime, Married, High Income, Private, Seniors	Male

Figure 1: Gender implications

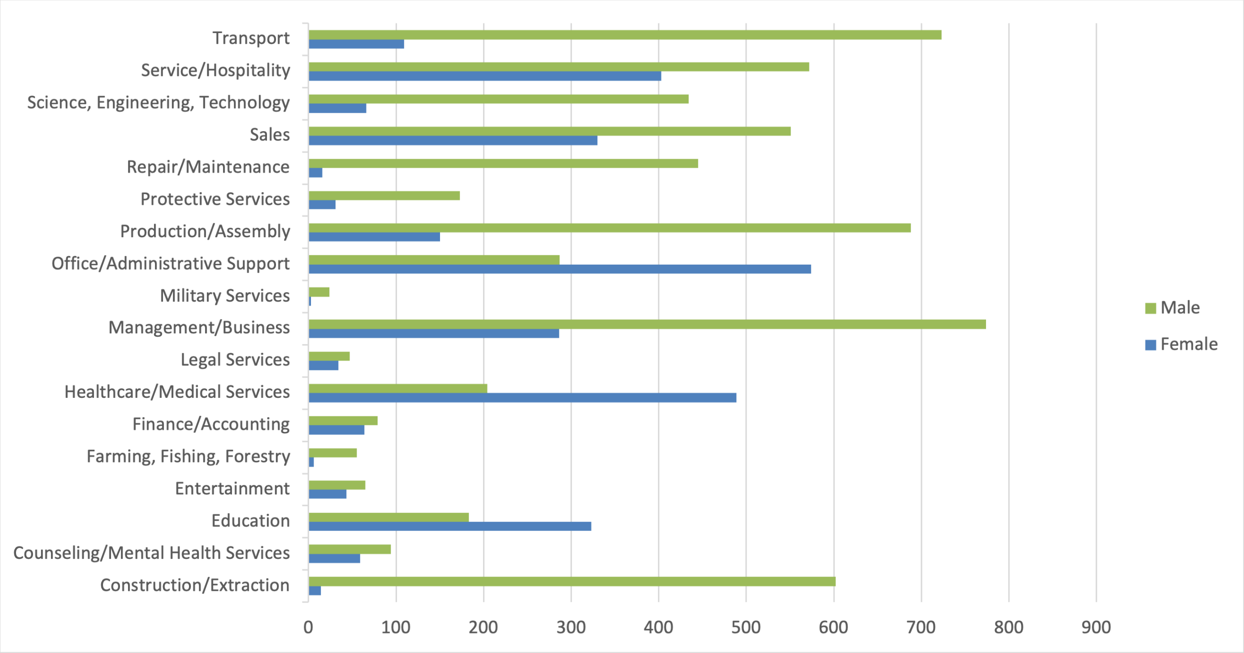


Figure 2: Occupation Bias

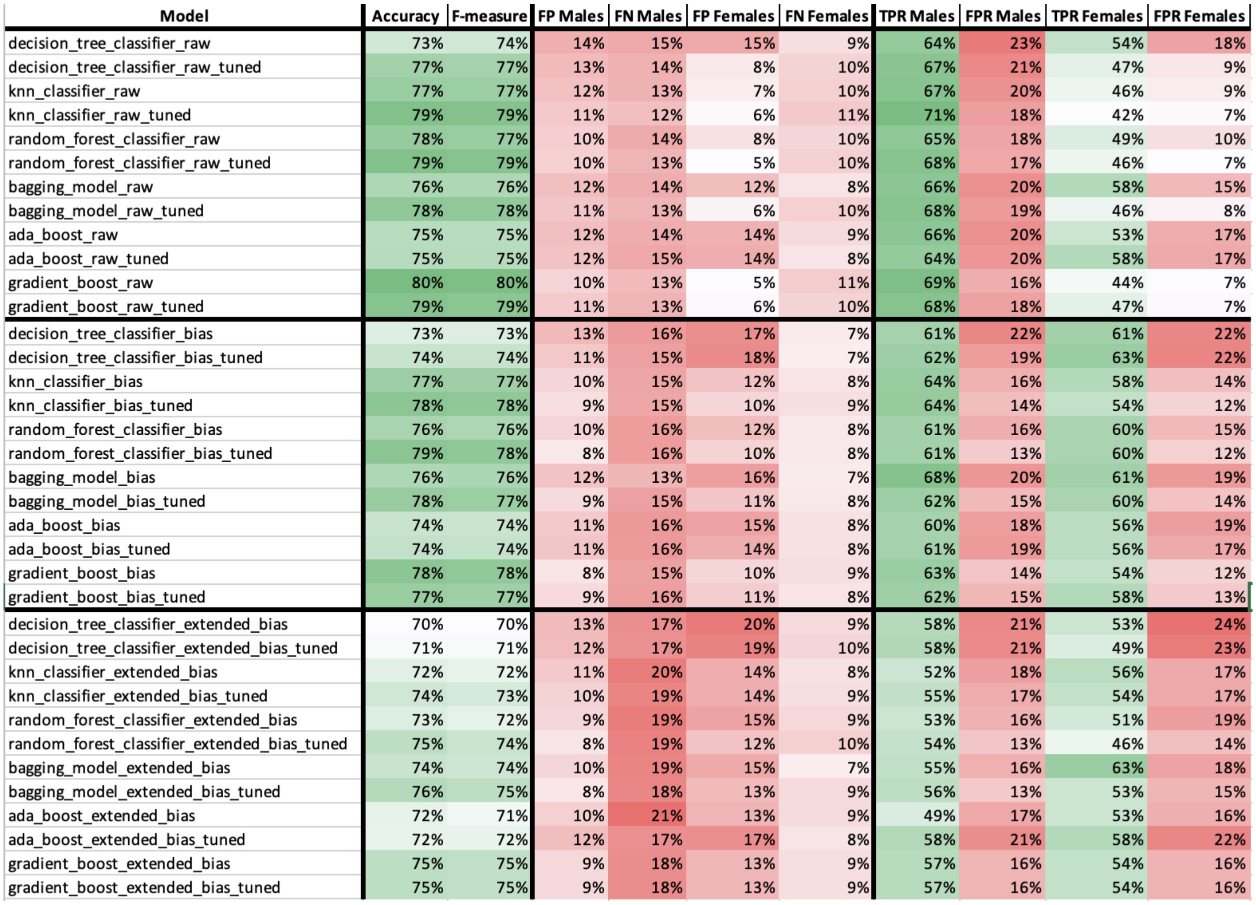
3 Model creation

Now that we have analyzed the dataset and pinpointed the features that might affect fairness, we can proceed to model construction. Initially, we'll develop classifiers using all features for baseline comparison, labeled as 'raw' in the table. For performance metrics, I've selected accuracy and the F-measure, as accuracy is an intuitive measure and the F-measure combines both precision and recall. The latter were emphasized during the lectures as effective indicators for model evaluation. We are going to train six models: a decision tree classifier, a KNN classifier, a random forest classifier, a bagging model, AdaBoost, and gradient boosting. From our lectures, we recognize the decision tree and KNN classifiers as fundamental models. I opted against using Naïve Bayes due to its complexity with mixed data types. The random forest, bagging, AdaBoost, and gradient boosting are advanced combined models. We will train all these models with alternative feature sets to enhance fairness, once with their default hyperparameters and once with optimal parameters derived from our GridSearch. The 'bias' column reflects outcomes from models with three modifications: merging 'Husband' and 'Wife' into 'Married', and dropping the 'sex' and 'gave birth this year' columns. 'Extended bias' further excludes the 'occupation' column, which we'll expect to improve fairness but might reduce accuracy. Analysis of occupation and income via a pivot table clearly shows the disparity in profitability across sectors, indicating the importance of this feature for the predictions. Our GridSearch will try to find the 'best' model. In order to define a metric that takes both fairness and performance into account, I decided to use the MSE (mean squared error) of the misclassified gender and error type pairs as loss function. In fact, it takes the MSE of the amount of false positives for males, false negatives for males, false positives for females and false negatives for females[1]. This way, it will try to minimize the total amount of wrongly classified samples (improve accuracy), as well as penalize a bias if either males or females are more misclassified than the other (improve fairness).

[1] An implementation of the custom loss function can be found in the *fp_fn_sum_with_ref* method in models.py.

4 Model performance

After training the models we outlined, we obtain the results summarized in the following table with performance and bias metrics (see figure 3). The columns for Accuracy and F-measure represent cross-validation outcomes on our held-out test dataset, constituting 10% of the original data. The subsequent four columns detail the rates of false positives and negatives. Our goal is to minimize these absolute values and ensure they are as uniform as possible across groups in the GridSearch. For comparative and visualization purposes, I decided to display them relative to the total number of males or females. A false positive indicates a sample incorrectly labeled as 'high' income when it should be 'low', while a false negative is labeled 'low' when it should be 'high'. The last four columns show the True Positive (TP) and False Positive (FP) rates, providing a more direct sense of fairness in our model performance. The closer these metrics are between males and females, the more fair a model behaves.



Model	Accuracy	F-measure	FP Males	FN Males	FP Females	FN Females	TPR Males	FPR Males	TPR Females	FPR Females
decision_tree_classifier_raw	73%	74%	14%	15%	15%	9%	64%	23%	54%	18%
decision_tree_classifier_raw_tuned	77%	77%	13%	14%	8%	10%	67%	21%	47%	9%
knn_classifier_raw	77%	77%	12%	13%	7%	10%	67%	20%	46%	9%
knn_classifier_raw_tuned	79%	79%	11%	12%	6%	11%	71%	18%	42%	7%
random_forest_classifier_raw	78%	77%	10%	14%	8%	10%	65%	18%	49%	10%
random_forest_classifier_raw_tuned	79%	79%	10%	13%	5%	10%	68%	17%	46%	7%
bagging_model_raw	76%	76%	12%	14%	12%	8%	66%	20%	58%	15%
bagging_model_raw_tuned	78%	78%	11%	13%	6%	10%	68%	19%	46%	8%
ada_boost_raw	75%	75%	12%	14%	14%	9%	66%	20%	53%	17%
ada_boost_raw_tuned	75%	75%	12%	15%	14%	8%	64%	20%	58%	17%
gradient_boost_raw	80%	80%	10%	13%	5%	11%	69%	16%	44%	7%
gradient_boost_raw_tuned	79%	79%	11%	13%	6%	10%	68%	18%	47%	7%
decision_tree_classifier_bias	73%	73%	13%	16%	17%	7%	61%	22%	61%	22%
decision_tree_classifier_bias_tuned	74%	74%	11%	15%	18%	7%	62%	19%	63%	22%
knn_classifier_bias	77%	77%	10%	15%	12%	8%	64%	16%	58%	14%
knn_classifier_bias_tuned	78%	78%	9%	15%	10%	9%	64%	14%	54%	12%
random_forest_classifier_bias	76%	76%	10%	16%	12%	8%	61%	16%	60%	15%
random_forest_classifier_bias_tuned	79%	78%	8%	16%	10%	8%	61%	13%	60%	12%
bagging_model_bias	76%	76%	12%	13%	16%	7%	68%	20%	61%	19%
bagging_model_bias_tuned	78%	77%	9%	15%	11%	8%	62%	15%	60%	14%
ada_boost_bias	74%	74%	11%	16%	15%	8%	60%	18%	56%	19%
ada_boost_bias_tuned	74%	74%	11%	16%	14%	8%	61%	19%	56%	17%
gradient_boost_bias	78%	78%	8%	15%	10%	9%	63%	14%	54%	12%
gradient_boost_bias_tuned	77%	77%	9%	16%	11%	8%	62%	15%	58%	13%
decision_tree_classifier_extended_bias	70%	70%	13%	17%	20%	9%	58%	21%	53%	24%
decision_tree_classifier_extended_bias_tuned	71%	71%	12%	17%	19%	10%	58%	21%	49%	23%
knn_classifier_extended_bias	72%	72%	11%	20%	14%	8%	52%	18%	56%	17%
knn_classifier_extended_bias_tuned	74%	73%	10%	19%	14%	9%	55%	17%	54%	17%
random_forest_classifier_extended_bias	73%	72%	9%	19%	15%	9%	53%	16%	51%	19%
random_forest_classifier_extended_bias_tuned	75%	74%	8%	19%	12%	10%	54%	13%	46%	14%
bagging_model_extended_bias	74%	74%	10%	19%	15%	7%	55%	16%	63%	18%
bagging_model_extended_bias_tuned	76%	75%	8%	18%	13%	9%	56%	13%	53%	15%
ada_boost_extended_bias	72%	71%	10%	21%	13%	9%	49%	17%	53%	16%
ada_boost_extended_bias_tuned	72%	72%	12%	17%	17%	8%	58%	21%	58%	22%
gradient_boost_extended_bias	75%	75%	9%	18%	13%	9%	57%	16%	54%	16%
gradient_boost_extended_bias_tuned	75%	75%	9%	18%	13%	9%	57%	16%	54%	16%

Figure 3: Model performance

4.1 Fairness

Let us now examine the fairness across all models. For the raw classifiers, the distinct shades of red and green in the performance charts vividly highlight significant disparities in the True Positive Rate (TPR) and False Positive Rate (FPR) between males and females, indicating bias. Conversely, for models employing basic bias mitigation techniques, the TPR and FPR are comparatively balanced between genders, signaling a marked enhancement in fairness. The models with extended bias adjustments also exhibit greater fairness than their raw counterparts, though the discrepancies are somewhat larger than in the basic bias models. Overall, it is evident that incorporating basic bias adjustments results in relatively fair models.

4.2 Performance

Upon reviewing the accuracy and F-measure data, it is clear that the models using basic bias perform only slightly worse than the raw models, while those with extended bias exhibit significantly poorer performance. This supports our initial hypothesis that occupation is a critical feature for income prediction. The lower performance in terms of accuracy also corresponds to reduced TPR and FPR values. According to the metrics, the raw gradient boost model performs best, albeit with high bias. The most effective 'fair' model is the random forest classifier with finely tuned hyperparameters, achieving an accuracy of 79%, an F-score of 78%, and balanced, low false positive (FP) and false negative (FN) rates, alongside well-adjusted TPR and FPR values. In contrast, the raw decision tree classifier shows relatively low accuracy and high FP and FN rates, with a notable disparity of 10% and 5% between the TPR and FPR respectively, making this the worst model. The decision tree model with extended bias, despite appearing fairer in the balance of TPR and FPR, does not perform better either. Other models, like the gradient boost with extended bias or the bagging model with basic bias, achieve mediocre results overall.

5 Loan predictions

In this section, we will deploy our 'best' model, the tuned random forest classifier with basic bias preprocessing, on the provided test dataset. Recall that 'basic bias preprocessing' involves consolidating 'Husband' and 'Wife' into 'Married' and omitting the 'sex' and 'gave birth this year' columns. The test dataset contains loan applicant data without the income column, which we will estimate using our trained classifier. Our testing, based on 900 unseen samples, demonstrated an accuracy of 79%. Thus, we anticipate approximately 1580 accurate predictions out of the 2000 new evaluations, with around 420 being incorrect. Further analysis using a Gender vs Income pivot table shows that 1058 applicants (379 female, 679 male) are predicted to have a high income and 942 (468 female, 474 male) to have a low income.

6 Conclusion

Throughout this report, we demonstrated a systematic approach to building and evaluating machine learning models with an emphasis on fairness and accuracy. The raw models, while highly accurate, contained significant biases, which we successfully solved through targeted preprocessing strategies. The tuned random forest classifier with basic bias preprocessing emerged as the most effective model, balancing fairness with a high level of predictive accuracy.