Assignment 1 Report

Alexandra Neagu, Teodor Oprescu, Alexandru Dumitriu, Răzvan Popescu

March 21, 2024

1 Models

In this section, we explore the development and evaluation of two distinct artificial intelligence (AI) models aimed at detecting welfare fraud within a synthetic dataset inspired by the one used by the algorithm previously employed in Rotterdam until 2021[1]. The objective of this comparative analysis is to showcase the stark contrast between a 'good' model, designed with ethical considerations, fairness, and performance optimization in mind, and a 'bad' model, characterized by potential biases, flawed design choices, and subpar performance.

Note that both the 'good' and 'bad' models were made compatible with the **ONNX (Open Neural Network Exchange)** framework, which is an open standard for representing deep learning and traditional machine learning (ML) models[2]. By converting the trained models into the ONNX format, they can be seamlessly integrated into various platforms, frameworks, and deployment environments, promoting interoperability and ease of deployment.

1.1 'Good' model

The 'good' model is meticulously crafted to adhere to ethical standards, striving to ensure fairness, transparency, and robustness in its decision-making process. Leveraging advanced techniques such as feature selection, hyperparameter tuning, and rigorous evaluation, this model strives to accurately identify instances of welfare fraud while minimizing false positives and avoiding undue harm or discrimination.

To begin with, let's outline the steps taken to create the 'good' AI model:

1. **Data Preprocessing**: By incorporating the entirety of the dataset, the model benefits from a richer representation of the underlying patterns and relationships within the data. The dataset was split into training and testing sets using the `train_test_split` function from *scikit-learn* (see Cell 2, Line 11 in **model_1.ipynb**). This step ensures that the model's performance can be evaluated on unseen data. The parameter `test_size=0.2` indicates that 20% of the data was reserved for testing purposes, while the remaining 80% was allocated to training the model. We chose this split ratio because it strikes a balance between ensuring an adequate amount of data for training while still retaining a sizable portion for evaluation[3]. Setting `random_state=42` ensures reproducibility in the data split. By fixing the random seed to a specific value, the data-splitting process becomes deterministic, yielding consistent results across different runs of the model. This is crucial for maintaining consistency in model evaluation and comparison, especially in collaborative or research settings where reproducibility is paramount[4].

2. **Feature Selection**: This step plays a pivotal role in identifying the most relevant attributes within the dataset and discarding redundant or noise-contributing features. This process enhances model efficiency,

[1] Dhruv Mehrotra et al. *Inside the suspicion machine*. Mar. 2023. URL: `https://www.wired.com/story/welfare-state-algorithms/`

[2] URL: `https://onnx.ai/`

[3] Jimin Tan et al. "A critical look at the current train/test split in machine learning". In: *arXiv preprint arXiv:2106.04525* (2021)

[4] URL: `https://scikit-learn.org/stable/glossary.html#term-random_state`

reduces overfitting, and improves interpretability by focusing on the most informative variables[5]. Feature selection is performed using a `RandomForestClassifier` with **class weighting**, a powerful ensemble learning technique that aggregates the predictions of multiple decision trees. The `SelectFromModel` function from the *scikit-learn* library is employed to automatically select features based on their importance scores calculated by the random forest classifier (see Cell 3, Line 2 in **model_1.ipynb**). Features with higher importance scores are deemed more influential in predicting the target variable, i.e., welfare fraud, and are retained for subsequent model training. We chose to leverage the random forest classifier for this task so the 'good' model benefits from its ability to handle complex datasets with high-dimensional feature spaces, non-linear relationships, and interactions between variables[6]. Moreover, the class weighting parameter ensures that the classifier accounts for imbalances in the dataset, where instances of welfare fraud may be relatively rare compared to non-fraudulent cases, thereby mitigating the risk of bias towards the majority class.

3. **Model Selection**: We chose **XGBoost (Extreme Gradient Boosting)** as the classifier (see Cell 4, Line 2 in **model_1.ipynb**). XGBoost is a state-of-the-art implementation of gradient boosting, known for its exceptional performance, scalability, and versatility across a wide range of ML tasks, including classification. Its popularity stems from its ability to handle complex datasets with high dimensionality, non-linear relationships, and noisy features, making it well-suited for the task of welfare fraud detection. One of the key advantages of XGBoost is its ensemble learning approach, where multiple weak learners (decision trees) are sequentially trained to correct the errors of previous models. This iterative process allows XGBoost to gradually improve its predictive accuracy and capture subtle patterns within the data, leading to superior performance compared to traditional ML algorithms[7].

4. **Pipeline Creation**: A pipeline is constructed to encapsulate the entire workflow, seamlessly integrating feature selection and classification into a cohesive framework (see Cell 5, Line 2 in **model_1.ipynb**). The pipeline serves as a standardized and modular approach to model development, facilitating code reusability, readability, and maintainability, allowing developers to easily modify and extend the pipeline as needed.

5. **Hyperparameter Tuning**: This step aims at optimizing the performance of the model by systematically exploring the hyperparameter space and identifying the configuration that yields the best results. The `GridSearchCV` class from the *scikit-learn* library is utilized to perform grid search with cross-validation, enabling the exhaustive search of hyperparameters while guarding against overfitting (see Cell 6 in **model_1.ipynb**). We chose this method for its robust exhaustive search, performant cross-validation using *k-fold*, automatic parameter selection, and seamless integration with our *scikit-learn* pipeline[8]. The hyperparameters under consideration in the grid search are:

   (a) `feature_selection__max_features` : This parameter determines the maximum number of features to be selected by the feature selection step. By varying this parameter over the values [50, 75, 100], the grid search seeks to identify the optimal subset of features that maximizes model performance while minimizing complexity and overfitting.

   (b) `classification__learning_rate` : This parameter controls the step size at each iteration of the gradient boosting process in the XGBoost classifier. A lower learning rate allows for more conservative updates to the model's parameters, potentially improving generalization performance. The values

[5]Kenji Kira and Larry A Rendell. "A practical approach to feature selection". In: *Machine learning proceedings 1992*. Elsevier, 1992, pp. 249–256

[6]Md Al Mehedi Hasan et al. "Feature selection for intrusion detection using random forest". In: *Journal of information security* 7.3 (2016), pp. 129–140

[7]Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794

[8]GSK Ranjan, Amar Kumar Verma, and Sudha Radhika. "K-nearest neighbors and grid search cv based real time fault monitoring system for industries". In: *2019 IEEE 5th international conference for convergence in technology (I2CT)*. IEEE. 2019, pp. 1–5

[0.1, 0.2, 0.3] are explored to find the learning rate that balances model complexity and predictive accuracy.

The grid search is performed with cross-validation (`cv=5`) (see Cell 6, Line 9 in **model_1.ipynb**), which partitions the dataset into five folds and trains the model on four folds while evaluating performance on the remaining fold[9]. This process is repeated five times, with each fold serving as the validation set once. Cross-validation helps ensure that the model's performance estimates are robust and reliable, capturing variations in the dataset and minimizing the risk of overfitting. After conducting grid search with cross-validation, the best combination of hyperparameters is identified, yielding the following results:

(a) **Best Parameters**: `classification__learning_rate` : 0.1,
`feature_selection__max_features` : 100

(b) **Best Cross-Validation Score**: 0.9630625517756222 (measured by ROC AUC score)

The corresponding cross-validation score of approximately 0.967 reflects the model's high predictive performance in terms of area under the receiver operating characteristic curve (ROC AUC), a commonly used metric for binary classification tasks.

6. **Model Training**: This step involves fitting the pipeline to the preprocessed training data, which encompasses both feature selection and classification steps, rather than solely fitting the XGBoost classifier. The optimized hyperparameters determined during the hyperparameter tuning process are utilized in this phase (see Cell 7 in **model_1.ipynb**).

7. **Model Evaluation**: This is the final step in assessing the performance of the model for detecting welfare fraud. This crucial phase involves analyzing various metrics to gain insights into the model's predictive capabilities, generalization ability, and overall suitability for real-world deployment. The chosen metrics are *precision*, *recall*, *F1-score*, and *accuracy* (see Cell 7, Line 11 in **model_1.ipynb**). Each metric captures different aspects of the model's predictive capabilities and provides valuable insights into its strengths and weaknesses[10]. Let's delve into each of these metrics and their implications:

   (a) **Precision**: It measures the proportion of true positive predictions (correctly identified instances of welfare fraud) out of all instances predicted as positive (both true positives and false positives). In this context, a precision of 0.95 for class 0 and 0.82 for class 1 indicates that the model achieves a high level of precision in identifying both fraudulent and non-fraudulent cases, with slightly lower precision for detecting welfare fraud (class 1).

   (b) **Recall**: It measures the proportion of true positive predictions identified by the model out of all actual positive instances in the dataset. A recall of 0.99 for class 0 and 0.50 for class 1 suggests that the model effectively captures the majority of non-fraudulent cases (high recall for class 0), but its performance in identifying welfare fraud cases is less satisfactory (lower recall for class 1), indicating potential room for improvement.

   (c) **F1-score**: It is the harmonic mean of precision and recall, providing a balanced measure of a classifier's performance. A high F1-score (0.97 for class 0 and 0.62 for class 1) indicates that the model achieves a good balance between precision and recall for class 0, but there is room for improvement in class 1, where the F1-score is lower.

   (d) **Accuracy**: Lastly, accuracy measures the overall correctness of the model's predictions, calculated as the proportion of correctly classified instances out of the total instances. With an accuracy of 0.939, the model demonstrates strong overall performance in correctly classifying both fraudulent and non-fraudulent cases.

[9] Daniel Berrar. "Cross-Validation". In: Jan. 2018. ISBN: 9780128096338. DOI: 10.1016/B978-0-12-809633-8.20349-X

[10] Ž Vujović et al. "Classification model evaluation metrics". In: *International Journal of Advanced Computer Science and Applications* 12.6 (2021), pp. 599–606

Additionally, the classification report provides a macro and weighted average of precision, recall, and F1-score across both classes, offering a summarized view of the model's performance across different evaluation metrics (Table 1).

Measure	Precision	Recall	F1-score
Macro avg	0.89	0.75	0.80
Weighted avg	0.94	0.94	0.93

Table 1: Macro and weighted averages for the 'good' model

By following these steps, the 'good' model was created with a focus on optimizing performance, ensuring efficient feature selection, and utilizing a powerful classifier. This model aims to accurately detect welfare fraud while minimizing false positives and maintaining fairness and transparency.

1.2 'Bad' model

Conversely, the 'bad' model represents a cautionary tale, illustrating the pitfalls and risks associated with AI deployment when ethical principles are disregarded. This model may exhibit biases, lack transparency, or prioritize performance metrics at the expense of fairness and societal impact. Through its flawed design and implementation, the 'bad' model serves as a reminder of the importance of ethical considerations and responsible AI development in addressing real-world challenges[11].

Similar to the 'good' model, let's outline the steps taken to create the 'bad' model:

1. **Data Preprocessing**: In this step, several actions are taken to manipulate the dataset, potentially undermining fairness principles. Unlike the 'good' model, which utilizes the entire dataset for training and testing, the 'bad' model intentionally creates a biased dataset. It selectively removes instances where the person is female and not flagged for welfare fraud (see Cell 4, Lines 11-12 in **data_prep_2.ipynb**). This manipulation introduces an artificial bias against female individuals who are not suspected of fraud, potentially leading to unfair treatment and discriminatory outcomes. Another bias is introduced, an age-related one, to discriminate against specific age groups, particularly old people (see Cell 4, Lines 1-5 in **data_prep_2.ipynb**). Additionally, it leverages the simplified representation of categorical variables, such as the `typering_other` variable, which signifies whether a case worker left a comment (0 for no comment, 1 for a comment). However, this simplistic representation fails to distinguish between positive and negative comments. Consequently, the data preprocessing step introduces bias towards individuals where case workers left comments, regardless of the comment's nature, further exacerbating the algorithm's bias (see Cell 4, Lines 18-19 in **data_prep_2.ipynb**). In contrast, the 'good' model prioritizes the inclusion of all instances in the dataset, ensuring fairness and impartiality in model training and evaluation. As a result, the 'bad' model trained on this biased dataset may exhibit skewed predictions, with potential disparities in performance across different demographic groups. This undermines the ethical principles of fairness and non-discrimination, potentially harming vulnerable populations and eroding trust in the welfare system. Lastly, the split ratio was kept the same as in the 'good', with the training dataset amounting to 80% of the overall dataset and the testing dataset the remaining 20%. This was done in order to set up a faithful comparison between the two models.

2. **Model Selection**: In this step of the 'bad' model, a `RandomForestClassifier` is chosen as the algorithm without incorporating any feature selection or hyperparameter tuning (see Cell 4, Line 1 in **model_2.ipynb**). This simplistic approach lacks sophistication and may result in suboptimal model performance. `RandomForest` is an ensemble learning method based on decision trees, known for its simplicity

[11] Eirini Ntoutsi et al. "Bias in data-driven artificial intelligence systems—An introductory survey". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10.3 (2020), e1356

and ability to handle complex datasets. However, it lacks the advanced capabilities and performance optimizations offered by *XGBoost*, such as gradient boosting, regularization, and tree pruning. As a result, the 'bad' model may not fully leverage the predictive power of more advanced algorithms, potentially leading to inferior performance and accuracy.

3. **Model Training**: In this step of the 'bad' model, a `RandomForestClassifier` is trained on the biased dataset without incorporating any feature selection or hyperparameter tuning. This simplistic approach lacks sophistication and may result in suboptimal model performance.

4. **Model Evaluation**: In this final step of assessing the performance of the 'bad' model in detecting welfare fraud, we decided to use the same metrics as previously seen with the 'good' model, again in order to set up a faithful comparison between the two approaches (see Cell 5, Line 6 in **model_2.ipynb**). Delving once again into these metrics:

   (a) **Precision**: In this model, a precision of 0.98 for class 0 and 0.18 for class 1 suggests that while the model performs relatively well in identifying non-fraudulent cases, it struggles with accurately identifying instances of welfare fraud. The low precision for class 1 indicates a high rate of false positive predictions, meaning that a significant portion of cases flagged as fraudulent may be incorrect. This could lead to unnecessary investigations and potential harm to individuals falsely accused of fraud.

   (b) **Recall**: A recall of 0.58 for class 0 and 0.91 for class 1 indicates that the model is relatively effective at capturing fraudulent cases but struggles with identifying instances where welfare fraud was not committed. The low recall for class 0 suggests that the model misses a considerable portion of innocent instances, potentially classifying them as fraudulent activities and causing financial losses to people.

   (c) **F1-score**: The F1-score for class 0 (0.73) is relatively high, indicating a good balance between precision and recall for non-fraudulent cases. However, the F1-score for class 1 (0.32) is considerably lower, reflecting the model's challenges in achieving a balance between precision and recall for instances of welfare fraud. This imbalance suggests that the model's performance may be skewed towards non-fraudulent cases, potentially leading to missed opportunities for fraud detection.

   (d) **Accuracy**: Lastly, the overall accuracy is 0.610, indicating that the model correctly classifies approximately 61% of instances. While accuracy provides a general measure of model performance, it may not capture the nuances of performance imbalances between different classes. Therefore, despite achieving a relatively moderate accuracy, the 'bad' model may still exhibit deficiencies in accurately identifying instances of welfare fraud, as reflected in the individual precision, recall, and F1-score metrics.

   Additionally, the classification report again provides a macro and weighted average of precision, recall, and F1-score across both classes, offering a summarized view of the 'bad' model's performance across different evaluation metrics (Table 2).

Measure	Precision	Recall	F1-score
Macro avg	0.59	0.74	0.52
Weighted avg	0.90	0.61	0.69

Table 2: Macro and weighted averages for the 'bad' model

The 'bad' model does not incorporate any feature selection techniques, such as `SelectFromModel` used in the 'good' model. Feature selection is essential for identifying the most informative attributes in the dataset and reducing dimensionality, which can improve model performance, mitigate overfitting, and enhance interpretability. By omitting feature selection, the 'bad' model may include irrelevant or redundant features, increasing the risk of model complexity, overfitting, and poor generalization.

The absence of hyperparameter tuning in the 'bad' model further compounds its limitations and undermines its effectiveness in welfare fraud detection. Without it, the model fails to leverage advanced optimization techniques, such as grid search or random search, to fine-tune model parameters and mitigate overfitting. This oversight may result in suboptimal model performance, reduced accuracy, and increased susceptibility to biases and variability in predictions, ultimately compromising its suitability for real-world applications in welfare fraud detection.

Lastly, the absence of a pipeline in the 'bad' model represents a significant limitation that hinders its efficiency, reproducibility, and scalability. Without a pipeline, the 'bad' model lacks a structured approach to data processing and model development, leading to ad-hoc implementation and potential errors. This lack of organization increases the complexity of model development and maintenance, making it challenging to reproduce results and scale the model to larger datasets or production environments. Additionally, the absence of a pipeline inhibits the integration of feature selection, hyperparameter tuning, and model evaluation steps, limiting opportunities for optimization and improvement.

To reiterate as a final point, the biased dataset used for training and evaluation in the 'bad' model may significantly impact the evaluation results. By selectively removing instances based on gender and fraud status, the dataset no longer accurately reflects the real-world distribution of welfare applications. This manipulation introduces artificial biases that may lead to skewed evaluation metrics and unjust outcomes, particularly for underrepresented demographic groups.

Based on the forthcoming assessments, the 'good' model exhibits minor shortcomings, which could be rectified with minimal adjustments in future iterations. Conversely, the 'bad' model performs inadequately across performance metrics, demonstrating pronounced issues of overconfidence and underconfidence. These results underscore the substantial impact of bias within the training data, highlighting its potential to significantly impair model effectiveness and reliability.

2 Tests

2.1 Giskard

To test both our models, we are using the Giskard[12] framework. This Python library helps us automatically detect vulnerabilities in our models, including performance biases, data leakage, spurious correlation, hallucination, toxicity, security issues, and many others. Furthermore, this helps us in assessing the fairness and bias within our models, particularly in the classification task.

Giskard typically begins by examining the dataset provided for testing. Then, it assesses the distribution of sensitive attributes (such as gender, age, race, etc.) to identify any potential biases present in the dataset. It also ensures that the data is appropriately prepared for fairness testing. This is typically done by applying preprocessing techniques such as data normalization or balancing to mitigate biases and ensure fair treatment across different groups. Additionally, Giskard may employ techniques such as subgroup analysis to evaluate fairness within specific subgroups (based on gender, ethnicity, education level, income level, etc.) defined by sensitive attributes. By preparing the data in this manner, Giskard ensures that fairness testing accurately reflects real-world scenarios and provides meaningful insights into the behavior of our models.

Once the dataset is prepared, Giskard evaluates the machine learning model (XGBoost in our case) using this data. It assesses the model's predictions to determine if there are any unfair outcomes with respect to different groups defined by sensitive attributes.

Furthermore, Giskard calculates various fairness metrics to quantify the level of bias present in the model's predictions. These metrics may include disparate impact, predictive parity, equal opportunity, and others. Such metrics reveal whether certain groups face disproportionate outcomes or if the model's performance is uniformly distributed across all groups.

Afterward, Giskard identifies specific areas where biases may be present in the model's predictions. It can pinpoint instances where certain groups are unfairly advantaged or disadvantaged by the model's decision.

[12]https://github.com/Giskard-AI/giskard

Lastly, Giskard may provide recommendations or strategies for mitigating the identified biases. This could involve adjusting the training data, modifying the model's features, or employing post-processing techniques to ensure fairer outcomes.

2.2 Demographic Parity Tests

In addition to utilizing the Giskard framework, we have implemented a series of demographic parity tests to assess potential biases in our models with respect to sensitive attributes such as gender and age. These tests are crucial for evaluating the fairness of our models' predictions across different demographic groups.

For gender parity testing, we group the test set instances based on the **persoon_geslacht_vrouw** feature, which represents the gender of the individuals (value of 1 for women, value of 0 for men). We then calculate and compare the mean risk (of fraud) scores predicted by the model for each gender group. Significant disparities in these risk scores could indicate gender bias in the model's predictions. Furthermore, we employ statistical tests like the Wilcoxon rank-sum test to evaluate if the distributions of predicted probabilities differ significantly between the gender groups.

To assess age parity, we divide the test set into age groups based on predefined age bins. We then compute the mean risk scores and other fairness metrics (e.g., true positive rates, false positive rates) for each age group. Substantial variations in these metrics across age groups could suggest age-related biases in the models' performance. We also employ the Kolmogorov-Smirnov test to compare the distributions of model predictions between different age groups.

2.3 Metamorphic Testing

Metamorphic testing is a technique that evaluates the behavior of a model by introducing perturbations or transformations to the input data and verifying if the model's outputs conform to expected metamorphic relations. In our case, we plan to implement metamorphic testing by inverting the values of selected binary features in the test set and analyzing the changes in the model's predictions.

For example, we can invert the values of the **persoon_geslacht_vrouw** feature, effectively swapping the gender labels for each instance. We then compare the model's predictions on the original and transformed instances. If the model exhibits biases related to gender, we would expect significant changes in the predictions when the gender labels are inverted. Similar metamorphic tests can be conducted with other binary features, such as **persoonlijke_eigenschappen_taaleis_voldaan** (whether the language requirement is met) or **deelname_act_reintegratieladder_werk_re_integratie** (related to work reintegration).

By incorporating metamorphic testing, we aim to assess the robustness and consistency of our models' predictions under controlled input perturbations. Significant deviations from expected behavior could indicate the presence of biases or inconsistencies in the models' decision-making processes.

These demographic parity tests and metamorphic testing techniques complement the fairness evaluations performed by the Giskard framework, providing a comprehensive approach to identifying potential biases and ensuring the fairness of our models' predictions across different demographic groups and input conditions.

2.4 Good Model Test Results

2.4.1 Giskard

When it comes to our good model, Giskard identified certain data subsets where the model's performance falls below the average. This discovery revealed a total of 15 issues, comprising 2 considerable and 13 medium concerns. These issues primarily pertain to performance, particularly in terms of recall values, within the context of the good model. The significant impact of 2 of the issues stems from the discrepancy between local and global recall within specific subsets, exceeding 10%. Conversely, the medium issues remain below the 10% threshold. The highest priority issue selects records where the age of individuals at the time of examination falls within the range of 59.5 (inclusive) to less than 64.5 (exclusive). Subsequently, the recall, representing the model's capability to accurately identify all pertinent instances of a specific class, yields a value of 0.806

when computed for the chosen subset of records. This means that the model correctly identified approximately 80.6% of the relevant instances within this subgroup. Moreover, the global recall for the entire dataset (without considering any specific conditions) achieves a value of 0.913. This indicates that, across the entire dataset, the model correctly identified approximately 91.3% of the relevant instances. We can assess the impact of the specified condition on the model's performance by calculating the difference between the two recalls. In this case, the recall for the subset is 11.67% lower than the global recall. This suggests that the model's performance in identifying relevant instances within this subset is particularly poorer compared to its overall performance across the dataset. Furthermore, this condition affects 1649 samples, accounting for approximately 13% of the dataset.

The remaining issues also stem from the recall values, yet the discrepancy decreases gradually as we examine different subsets. These performance-related issues typically arise due to nuances in the dataset rather than indicating fundamental flaws in our model. For example, we noticed that certain subgroups within the data exhibit different distributions, which makes it harder for the model to generalize its predictions to those subgroups. These issues do not necessarily indicate broader deficiencies in our model's architecture or algorithm. Instead, they highlight areas where the model's predictions may be less consistent due to the inherent complexities of the data. Despite all of this, the overall effectiveness and reliability of the model may not be undermined, especially since our model achieves a high prediction accuracy.

2.4.2 Demographic Parity Tests

The Gender Parity Test assesses whether the model's predictions exhibit fairness concerning gender. We grouped the test set instances based on the **persoon_geslacht_vrouw** feature and calculated the mean risk scores predicted by the model for each gender group.

The results indicate a very slight disparity in risk scores between genders. For men, the mean risk score is approximately 0.066, while for women, it is approximately 0.055. Although there is a discernible difference, it does not suggest significant gender bias in the model's predictions. A Wilcoxon Rank-Sum Test yielded a p-value of 0.155, indicating no statistically significant difference in the distributions of predicted probabilities between gender groups. Thus, the model demonstrates relatively balanced predictions across different genders.

The Age Parity Test evaluates the model's fairness with respect to age groups. We divided the test set into age groups based on predefined age bins and computed the mean risk scores for each group. (18-26: 31.03%, 27-36: 16.96%, 37-46: 10.60%, 47-56: 3.06%, 57-66: 1.05%) The analysis reveals relatively varying risk scores across different age groups. As expected, younger age groups generally exhibit higher mean risk scores compared to older age groups. For instance, the mean risk score for the age group 18-26 is approximately 0.310, whereas for the age group 56-66, it reduces significantly to around 0.011. This trend suggests that the model tends to assign higher fraud risk to younger individuals.

Additionally, the Kolmogorov-Smirnov test was conducted to compare the distributions of model predictions between different age groups. The results indicate statistically significant differences in predicted probabilities between certain age groups, particularly between the age group 18-26 and older age groups (46-56 and 56-66). This suggests that the model's predictions may not be uniformly distributed across all age groups, indicating potential age-related biases.

Furthermore, we conducted a Demographic Parity Test for the feature **adres_aantal_brp_adres**, representing a number from 1 to 10 that corresponds to a different region of Rotterdam. The analysis revealed variations in mean risk scores across regions. For instance, individuals with an address in region 9 exhibit a mean risk score of 0.2, while those in region 1 have a mean risk score of approximately 0.037.

Overall, while the model demonstrates balanced predictions in terms of gender, it exhibits slight variations in risk scores across different age groups and address counts. While the differences are relatively on the lower end, further investigation and mitigation strategies may be necessary to address potential biases associated with these demographic factors.

2.4.3 Metamorphic Tests

In our evaluation of the good model, we conducted metamorphic tests by flipping binary features in the test set to observe variations in predictions. Firstly, we flipped the values of the **persoon_geslacht_vrouw** feature, effectively swapping gender labels for each instance. The model's accuracy on the transformed data was 93.99%. Notably, the precision, recall, and F1-score for the minority class slightly decreased compared to the original data, indicating a very slight impact on model performance.

Next, we inverted the values of the **persoonlijke_eigenschappen_taaleis_voldaan** feature, representing whether the language requirement is met. The model achieved an accuracy of 94.11% on this transformed data. Similarly, there were minor decreases in precision, recall, and F1-score for the minority class, suggesting a limited effect on model predictions.

Finally, we flipped the values of the **deelname_act_reintegratieladder_werk_re_integratie** feature, related to work reintegration participation. The model attained an accuracy of 94.07% on this modified data. Similar to previous tests, there were slight decreases in all of the aforementioned metrics.

Overall, these metamorphic tests demonstrate the robustness of the good model's predictions to controlled input perturbations. While minor fluctuations in performance metrics were observed, the model maintained high accuracy and generally consistent prediction behavior across different feature transformations.

2.5 Bad Model Test Results

2.5.1 Giskard

By analyzing the performance of our bad model, Giskard found many issues spanning overconfidence, underconfidence, and performance disparities. Giskard identified 45 distinct issues, each shedding light on areas where the model's predictions deviate significantly from the expected norms.

Among these issues, two stand out for their substantial impact on precision and recall metrics within specific subsets of the data. For instance, one critical issue relates to instances where individuals' ages fall within a narrow range, resulting in a noticeable decline in predictive accuracy. Specifically, within this subset, the model's recall drops by approximately 11.67% compared to the global average, affecting around 13% of the dataset. This discrepancy suggests a significant loss of predictive accuracy within this demographic segment.

Moreover, the bad model shows a strong tendency toward overconfidence and underconfidence in various features. For example, features such as **instrument_ladder_historie_activering** and **pla_historie_ontwikkeling** show overconfidence rates exceeding 60% compared to the global average, indicating a propensity to overestimate the likelihood of certain outcomes. Conversely, features like **instrument_ladder_historie_other** and **pla_einde_doelstelling_bereikt** demonstrate underconfidence rates exceeding 25% above the global average, highlighting a tendency to underestimate certain outcomes.

Furthermore, performance disparities manifest across different subsets of the data, with varying levels of precision and recall. These discrepancies underscore the challenges the model faces in generalizing predictions across diverse demographic groups and feature combinations.

In summary, the bad model's performance issues, characterized by overconfidence, underconfidence, and performance disparities empirically confirm the fundamental biases that this model presents.

2.5.2 Demographic Parity Tests

The Gender Parity Test serves as a critical assessment of whether the introduced bias has significantly influenced the model's predictions. Leveraging the **persoon_geslacht_vrouw** feature, we check the risk of fraud scores predicted by the model for both men and women. The findings reveal the substantial impact of the bias on the model's outcomes.

Specifically, for men, the mean risk score hovers around 35.7%, whereas for women, it rises to about 67.4%. This contrast not only underscores the model's tendency to label women as fraud risks nearly twice as often as men but also accentuates a staggering increase in the perceived risk of fraud for both genders. Notably, the risk of fraud surges approximately fivefold for men and tenfold for women. Consequently, the model is

predisposed to categorize men as non-fraudulent while casting women as potentially fraudulent, given that their fraud probability exceeds 50%.

The Wilcoxon Rank-Sum Test further corroborates these findings, yielding a statistic of 19.05 and an extremely low p-value of 6.43e-81. This statistical analysis underscores the stark disparity in predicted probabilities between gender groups, signifying a significant departure from equitable predictions by the model.

The Age Parity Test results for the bad model reveal a notable shift in the distribution of mean risk scores across different age groups. While the previously defined age groups had a decreasing trend of fraud risk as the age increased in the good model, the bad model exhibits a roughly constant risk for all age groups (18-26: 45.9%, 27-36: 52.1%, 37-46: 58.9%, 47-56: 63.2%, 57-66: 68.9%). The introduction of the age bias seems to have influenced the model to unfairly assign higher fraud risk to older individuals. While the younger age group of 18-26 experienced an increase in the fraud risk from 31.03% to 45.9%, the oldest age group of 57-66 experienced an extremely significant increase from 1.05% to 68.9%. Even though the new risks presented by the bad model are now roughly more similar than the risks of the good model, this similarity does not reflect the reality.

The Kolmogorov-Smirnov test results provide insights into the distribution of model predictions across different age groups. While mean risk scores suggest a general trend of increasing fraud risk with age, the test reveals significant distributional differences between certain age groups. Specifically, comparisons involving older age groups (46-56 and 56-66) exhibit lower p-values, indicating notable deviations in the distributions of model predictions.

Additionally, we tested the bias introduced for the feature related to "other comments" being present in somebody's application. Thus, the test reveals that if somebody has an existing "other comment" on their file, the model will mark them as twice more likely to commit fraud (0 comments: 32.47%, 1 comment: 75.93%, 2 comments: 90.47%) completely independent of whether the comment is positive or negative.

Thus, the bad model demonstrates fundamental biases in its predictions in terms of gender, age, and the existence of comments, and the demographic parity tests confirm this.

2.5.3 Metamorphic Tests

To ensure a comprehensive evaluation of the bad model's performance, we conducted metamorphic tests mirroring those performed on the good model. The outcomes revealed significant differences, proving once again the model's inherent biases and suboptimal performance.

Initially, by flipping the values of the **persoon_geslacht_vrouw** feature, akin to gender flipping in the good model assessment, the bad model exhibited a substantially lower accuracy of 72.99% on the transformed data. Similarly, when inverting the values of the **persoonlijke_eigenschappen_taaleis_voldaan** feature related to language requirement fulfillment, the bad model achieved an accuracy of 61.37% on the modified data. Finally, flipping the values of the **deelname_act_reintegratieladder_werk_re_integratie** feature, associated with work reintegration participation, resulted in the bad model attaining an accuracy of 60.81% on the transformed data. For all of these features, while there was an increase in recall for the minority class, indicating a partial improvement in identifying instances of fraud, the overall performance remained significantly inferior to that of the good model. Moreover, a notable decrease in precision suggested a higher rate of false positives.

In essence, while mirroring the tests conducted on the good model, the outcomes for the bad model starkly diverged, revealing significant disparities in performance and underscoring the presence of biases ingrained within the model's decision-making processes.

3 Strengths & Weaknesses

3.1 Model 1

3.1.1 Strengths

1. **High accuracy** - 94%, indicating good overall performance in predicting the target variable.

2. **Interpretability** - Using feature selection to reduce the number of features it becomes easier to reason about the impact that individual features have.

3. **Noise-resistance** - feature selection eliminates features that have little predictive addition and add a large amount of noise for the subsequent classification model.

4. **Efficiency** - It is much faster to train a model with up to 100 features than a 500 features one, whereas by using the most important features we can retain the performance.

5. **Prevent overfitting** - By employing cross-validation we can ensure that the hyperparameters selected perform great and are not overfitting the model to the training set.

6. **High Robustness** - By performing metamorphic tests we ensured that completely changing the binary values of a column is not affecting the model performance drastically.

7. **Data Imbalance Handling** - By leveraging class weights we were able to overcome the evident data imbalance without using sampling techniques that would have either reduced the training data or introduced further uncertainty from synthetic data.

3.1.2 Weaknesses

1. **Small biases** - According to the Giskard testing tool, there remained 2 features against which the model was not behaving perfectly. However, considering around 400 features, this represents a small proportion.

2. **Feature Selection Stochasticity** - Using Random Forest for feature importance calculation introduces uncertainty regarding the selection of the best features, although there is still high confidence in the result.

3.1.3 Future Improvements

- **Novel Architecture** - To achieve nearly perfect performance in all categories, resources should be allocated towards developing an architecture specifically tailored to this task, which can then be fine-tuned.

3.2 Model 2

3.2.1 Strengths

1. **Simplicity** - By utilizing a very simple model without any parametrization, great performance is achieved in terms of accuracy, especially considering the introduced biases.

2. **Efficiency** - Using all the features, the biases introduced had a reduced impact on the overall performance.

3.2.2 Weaknesses

1. The biased model may not generalize well to other types of biases or fairness violations that were not explicitly accounted for during its creation.

2. **Robustness** - Metamorphic tests revealed that completely changing the binary values of a column significantly affects the model's performance.

3. **Data Imbalance** - By not addressing this issue, the model tends to choose the majority class in uncertain situations, leading to biased predictions.

3.2.3 Future Improvements

- By applying most of the steps from Model 1, even this simplistic model can likely perform much better, primarily due to the syntactic nature of the data.

3.3 Testing Categories

3.3.1 Strengths

1. Implemented a range of statistical tests (Wilcoxon rank-sum, Kolmogorov-Smirnov) to evaluate demographic parity across sensitive attributes like gender and age.
2. Calculated various fairness metrics (TPR, FPR, positive prediction rate) to assess potential discrimination against protected groups.
3. Analyzed feature correlations to understand how different variables relate to model predictions.
4. Explored disparate impact by comparing positive prediction rates across different demographic groups.
5. Assessed robustness through metamorphic tests.

3.3.2 Weaknesses

1. The testing primarily focused on demographic parity and equal opportunity but did not consider other fairness criteria like counterfactual fairness or individual fairness.
2. Only a limited number of sensitive attributes (gender, age) were considered in the analysis.
3. No explicit methods were used to interpret the model's decision-making process or feature importance.

3.3.3 Future Improvements

- Expand the range of tests by incorporating different categories of tests from black-box approaches.

By highlighting the strengths and weaknesses of both models and the testing categories, this analysis provides insights into areas for improvement and directions for future research. While the "good" model achieved high accuracy, it is essential to continue developing techniques to mitigate potential biases and ensure fairness across different demographic groups. Furthermore, the testing categories could be expanded to incorporate additional fairness criteria, robustness evaluations, and other, more diverse, types of testing.

References

Mehrotra, Dhruv et al. *Inside the suspicion machine*. Mar. 2023. URL: `https://www.wired.com/story/welfare-state-algorithms/`.

URL: `https://onnx.ai/`.

Tan, Jimin et al. "A critical look at the current train/test split in machine learning". In: *arXiv preprint arXiv:2106.04525* (2021).

URL: `https://scikit-learn.org/stable/glossary.html#term-random_state`.

Kira, Kenji and Larry A Rendell. "A practical approach to feature selection". In: *Machine learning proceedings 1992*. Elsevier, 1992, pp. 249–256.

Hasan, Md Al Mehedi et al. "Feature selection for intrusion detection using random forest". In: *Journal of information security* 7.3 (2016), pp. 129–140.

Chen, Tianqi and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

Ranjan, GSK, Amar Kumar Verma, and Sudha Radhika. "K-nearest neighbors and grid search cv based real time fault monitoring system for industries". In: *2019 IEEE 5th international conference for convergence in technology (I2CT)*. IEEE. 2019, pp. 1–5.

Berrar, Daniel. "Cross-Validation". In: Jan. 2018. ISBN: 9780128096338. DOI: `10.1016/B978-0-12-809633-8.20349-X`.

Vujović, Ž et al. "Classification model evaluation metrics". In: *International Journal of Advanced Computer Science and Applications* 12.6 (2021), pp. 599–606.

Ntoutsi, Eirini et al. "Bias in data-driven artificial intelligence systems—An introductory survey". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10.3 (2020), e1356.