Evaluating Fairness Using Constraint Relaxation to Solve Convex Functions

Merih Marc Atasoy, Farhan Wadia, Chris Palumbo, Ita Zaporozhets, *University of Toronto*

Abstract—With **Machine Learning** being readily used in applications that have high **ethical/social implications**, the notion of fairness in these models becomes a point of discussion that is highly regarded. Previous literature by the likes of Donini, Wu, and Lohaus have made several attempts in proposing various solutions to the problem of fairness in Machine Learning [1]. Given these advancements and methodologies in the form of **constrained optimization** problems and fairness metrics, the team investigates unanswered questions by conducting **performance evaluation** on various models, by utilizing adapted and/or extended versions of the previous technology. With focus on the recent research by Lohaus et al. titled "Too Relaxed to Be Fair," the team explores the proposed algorithm SearchFair to further analyze improved methods for addressing fairness in artificial intelligence [1].

The project implementation can be found at: https://github.com/itazap/MIE424-FinalProject

Index Terms— **G.1.6.a Constrained Optimization, O.9 Ethical/Social Implications, I.2.6.g Machine Learning, H.3.4.d Performance Evaluation**

1 Introduction

As the use of ML applications emerges into fields that determine outcomes for humans, levels of confidence with regards to bias and models' social implications increasingly become an area of focus. Models (more specifically, classifiers) in machine learning are considered to be fair if the outputted predictions are free of any form of unjust behaviour [1]. In this context, unjust behaviour constitutes a model's dependency on patterns of human bias with respect to sensitive attributes, such as gender, race, sexuality, etc. Therefore, a fair model should aim to make predictions that are independent of any and all attributes, or at least, having predictions that are completely independent of sensitive attributes in the data.

2 Motivation

This paper focuses on classification under fairness constraints, based on the literature from Michael Lohaus, Michael Perrot, and Ulrike von Luxburg, titled 'Too Relaxed To Be Fair.' [1] The paper builds upon previous work to address the challenge of training a classifier that is not biased against a group of individuals using constrained optimization, and aims to highlight some downfalls in previous literature.

2.1 Measuring Fairness

Several efforts have been made to mathematically formulate and represent fairness in the field of machine learning [1]. These formulations have helped produce measures that score fairness in different classification models, and are built upon two foundational ideas: Demographic Parity, and Equality of Opportunity.

2.1.1 Demographic Parity (DP)

Demographic Parity is a measure that attempts to evaluate whether or not positive outcomes are occurring at equal rates for each protected class in a dataset. In mathematical representation, as seen in (1), the probability of classification (P[$f(x) > 0$]) is equal for each of the sensitive attributes (s=1, s=-1).

$$\mathbb{P}_{(x,s,y)\sim\mathcal{D}_\mathcal{Z}}[f(x)>0|s=1] = \mathbb{P}_{(x,s,y)\sim\mathcal{D}_\mathcal{Z}}[f(x)>0|s=-1]\,. \tag{1}$$

2.1.2 Equality of Opportunity (EO)

Equality of Opportunity is a measure that attempts to evaluate whether or not positive outcomes are occurring at equal rates for each protected class in a dataset, provided that certain defined qualifications are being met.

In the mathematical representation (2) below, the probability of classification (P[f(x) > 0]) is independent of the assignment of the sensitive attributes (s=1, s=-1) yet again; however, this is only the case given positive outcomes (y=1).

$$\mathbb{P}_{(x,s,y)\sim\mathcal{D}_\mathcal{Z}}[f(x)>0|y=1,s=1] = \mathbb{P}_{(x,s,y)\sim\mathcal{D}_\mathcal{Z}}[f(x)>0|y=1,s=-1]\,. \tag{2}$$

2.2 Donini (2018)

Donini addresses fairness in supervised learning algorithms, by introducing fairness requirements, namely ensuring that sensitive information (e.g. knowledge about the ethnic group of an individual) does not 'unfairly' influence the outcome of a learning algorithm [2]. The focus is to first outline a general framework for empirical risk minimization under fairness constraints. The paper does not claim to be the most effective at doing so, however it is framed as a scaffolding to be built upon by other research. The proposed general framework is referred to as the quantity as difference of EO (DEO). This DEO is referred to as DDP in the 'Too Relaxed to Be Fair' Paper.

$$\min \left\{ \hat{L}_h(f) : f \in \mathcal{F},\ \left| \hat{L}_h^{+,a}(f) - \hat{L}_h^{+,b}(f) \right| \leq \hat{\epsilon} \right\}. \tag{3}$$

Unfortunately this is a difficult nonconvex nonsmooth minimization problem, and for this reason it is more convenient to solve a convex relaxation. To transform into a convex problem, the paper proposes to replace the hard loss in the risk with a convex loss function lc (e.g. the Hinge loss lc = max{0,ll}) and the hard loss in the constraint with the linear loss.

2.3 Wu (2019)

Wu addresses fairness in binary classification by (1) incorporating various fairness metrics into classic classification models as constraints, (2) the convex constrained optimization problem is solved efficiently; and (3) the lower and upper bounds of real-world fairness measures that are established using surrogate functions, providing a fairness guarantee for constrained classifiers [3]. Surrogate functions are those which approximate other functions; in this paper, surrogate loss functions such as hinge, square, and exponential are used for their convexity benefit to approximate 0-1 loss for classification problems.

Multiple fairness measures are considered, with focus on demographic parity, defines as the requirement of the classifier's decision to be independent to the sensitive attribute, such as sex or race. It is quantified by risk difference: the difference of the positive predictions between the sensitive group and non-sensitive group. The problem is formulated as an LP to minimize a loss function constraint by keeping the fairness measure below a certain threshold.

2.4 Lohaus, Perrot, Ulrike von Luxburg (2020)

Lohaus proposed a novel advancement in theoretically guaranteeing fairness in 2020, called the SearchFair framework in 2020 [1]. It used a strongly convex problem formulation and a trade-off between accuracy and fairness controlled by tunable hyperparameters (beta n lambda) in the objective function [1]. What this allowed for was to find a set of hyperparameters for accuracy and fairness, that ensures desirable accuracy of the model, while ensuring a level of fairness that does favour any of the groups.

$$f^{\beta}_{\hat{\mathcal{D}}_z}(\lambda) = \arg\min_{f \in \mathcal{F}} \hat{L}(f) + \lambda \mathrm{R}_{\widehat{\mathrm{DDP}}}(f) + \beta\Omega(f) \tag{4}$$

The algorithm finds a classifier that minimizes the convex loss function (the paper uses *L(f)* = Hinge Loss) with respect to a convex approximation of a signed fairness constraint (the paper uses $R_{DDP}(f)$ = Demographic Parity DDP) and a convex regularization term ($\Omega(f)$ = squared L2 norm).

3 Goal of Implementation

The goal of this implementation is to train multiple classifiers that perform well, and ensure they do not learn human bias based on societal prejudice.

There will be multiple classifiers completed to adapt recent years' fairness advancements to explore new avenues and answer certain questions that have arisen during research. The observed results and comparisons between these adaptations will then ultimately be discussed to highlight new and interesting findings.

The initial baseline model that will act as the foundation for the aforementioned adapted models, and will be a classifier that disregards fairness. This will act as a good scaffolding to compare accuracy against.

Evidently, the next iteration of models will be ones that utilize fairness scoring metrics, and furthermore, will be tuned through hyperparameter grid searches of accuracy and fairness, respectively. Lastly, the model created in the final iteration will be the adaptation of Lohaus' SearchFair algorithm, which is the optimization problem discussed in 2.4.

4 Exploring the Adult Dataset

The dataset includes features such as sex, race, and native-country, with the classification labels being whether the individual makes over or under 50K/ year.

This dataset was chosen because it contains sensitive attributes such as race and relationship which can appear to have causation to income by biased societal norms.

An exploratory data analysis was conducted to visualize the dataset, since it was not discussed in detail in the Lohaus et al. paper. There were some key areas of concerns that the team addressed before moving forward.

4.1 Assumptions Made

The assumptions from the Lohaus et al. paper were further explored through visualizations. The original dataset's "race" attribute included 5 race labels, as seen in Figure 1 below, Lohaus' team grouped all non-white races together, which makes the assumption they are equally marginalized. This may pose issues as some minority race groups are more severely marginalized and cannot be easily grouped into a single representory class. In addition, although the "Other" category is ambiguous, the team decided to move forward with this assumption for the purpose of binary classification. .

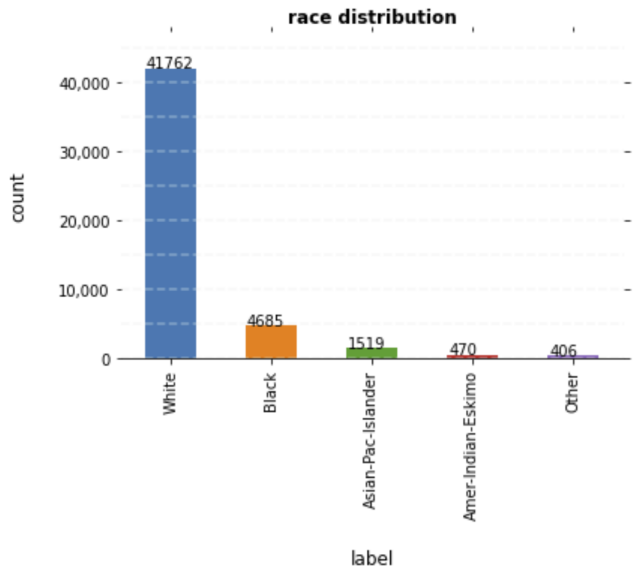


Figure 1. Race (White, Black, Asian Pacific Islander, American Indian Eskimo, Other) distribution

4.2 Unbalanced Labels

As seen in Figure 2 below, the >50K label only makes up 25.08% of the dataset. Lohaus's SearchFair algorithm results report a 25% classification error, as seen by the red and pink bars in Figure 2; however, this error score could simply be produced if the algorithm just predicted the majority label. This is a major concern since it can mean that SearchFair is not learning features from the Adult dataset.

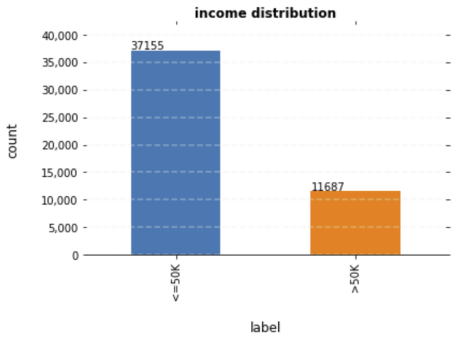
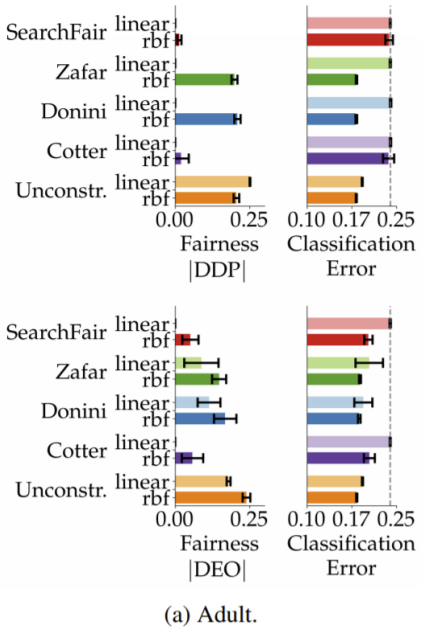


Figure 2. Label (income >50K or ≤50K) distribution

Before moving forward with implementation, the team found it necessary to balance the dataset to have equal data points for each of the two labels, which reduced the dataset to 23374 points.



(a) Adult.

Figure 3. Adult results for the Lohaus et al. paper

5 IMPLEMENTATION

The implementation was primarily done using the existing code from Michael Lohaus' SearchFair repository. As previously mentioned, the team chose not to adapt this code to a new dataset, and focused on the Adult dataset. This is because the code is fairly straightforward and would easily be applied to new problems.

Therefore instead, the team chose to adapt the existing code to answer any questions or concerns that came up while exploring different methods and results. For example, in Section 4, the Adult dataset was balanced in order to explore the notions of fairness and robustness. This allowed the results to then be compared to the

results of the raw (unbalanced) dataset in all of the tests. In the following sections, the existing code was adapted to implement an initial baseline classifier. Next, a testing procedure was created, which can be used for both the baseline and the SearchFair models. A grid search can then be run over the hyperparameters for the baseline classifier, in order to explore how accuracy and fairness scores (DDP) are affected by the hyperparameters in a standard model that ignores fairness in training. Finally, the team tookLohaus' SearchFair class and ran it alongside the aforementioned tests for comparison.

Note: to improve runtime, the team decided to reduce the number of training data points to 1200. In the future, this number can be easily increased if the proper equipment is available.

5.1 Baseline Classifier

The baseline classifier was implemented using a BaselineModel class that extends sklearn's BaseEstimator [4], however cvxpy [5] is used for constructing the minimization problem and optimizing it. The baseline model class uses a SVM that can be initialized for various kernel and loss function types.

The class has various input hyperparameters that are initialized to test different configurations. The important hyperparameters for the tests are:

- Regularization strength (beta)
- SVM Kernel Type $\in$ {Linear, RBF, Polynomial}
- Kernel Width for RBF Kernel (gamma)
- Degree for Polynomial kernel (gamma)
- Loss Function Type $\in$ {Hinge, Logistic', Squared, Exponential}

The baseline model class is broken up into its sub-methods for preprocessing, problem construction, prediction and optimization.

Preprocessing the data includes initializing class parameters, such as the coefficient array that is trained/optimized and used in the SVM for prediction. Preprocessing also configures the loss function and kernel functions based on the inputs given in the initialization. The given code has a hyperparameter for the number of features used in the SearchFair algorithm. Due to this being the baseline, all given features are used.

Constructing the problem is primarily for setting up the necessary variables, parameters and functions for the cvxpy optimization problem. First, the parameters of the SVM model are initialized using cvxpy's Variable class. Next, the kernel matrix is initialized using cvxpy's Parameter class. Then, an empty bias term is added. The bias term was not included in the existing repository; however,the team decided to include it to align with course material. Next, the loss function is generated using the given loss function. The loss function is generated identical to the function in Figure 4 (Section 2.4). There is a summation over the loss function calculation, and a standard L2-norm regularization term. Finally, the problem is generated to minimize the loss function using cvxpy's Problem class.

5.2 Testing Procedure with Fairness Scoring

The testing procedure is implemented as a class that is initialized for each model. The testing procedure class handles all steps necessary to build the appropriate data (based on the given sensitive attribute and balanced/unbalanced dataset), build the cvxpy model, fit the model, and finally produce and report the testing metrics. The 4 testing metrics reported and compared in Section 6 are:

- Total Run Time for Build
- Test Accuracy
- DDP Fairness Score
- DEO Fairness Score

The DDP and DEO fairness scores are computed using the provided equations and code. The DDP score is calculated by the difference of the unprotected positive rate (sensitive attribute = 1) and the protected positive rate (sensitive attribute = -1). The DEO score is calculated by the difference of the true unprotected positive rate and the true protected positive rate.

5.2.1 Baseline Hyperparameter Grid Search for Accuracy

To access a robust baseline, a grid search over the regularization hyperparameters was conducted for both the linear and rbf kernel types. The grid search was simply implemented using sklearn's model selection library, which uses a cross-validation approach to find the best model with respect to validation accuracy. In the real world where fairness metrics are ignored, researchers often choose the best model with respect to accuracy.

5.2.2 Baseline Hyperparameter Grid Search for Fairness Scores

The grid search was extended to explore the best model with respect to the DDP fairness metric. This search was done to study how 'fair' a model can be trained without

the use of a fairness improvement algorithm such as SearchFair. It was implemented by simply defining a custom scoring function (DDP_scorer) and passing it into the sklearn's GridSearchCV function.

5.3 Lohaus' SearchFair Algorithm

The existing code for the SearchFair class can be found in Michael Lohous' Github repository (mlohaus/SearchFair) [6] and implements equation (4) in Section 2.4. The team used this code to run tests on both the unbalanced and balanced datasets. The SearchFair class is similar to the baseline classifier class described in Section 5.1; however the SearchFair algorithm is implemented in the 'fit' function, where the cvxpy model is preprocessed, constructed and optimized. The goal of the SearchFair algorithm is to find an appropriate lambda term (regularization hyperparameter) that is used in optimization, and produces a model that both maximizes accuracy (minimizes loss), and is subject to the fairness constraint. This is implemented in the existing code using nested comparison statements. On each iteration of the algorithm: first, a new lambda value is calculated based on the last. Next, the model is optimized using the current lamba term. Then, the model is used to make predictions on a sample dataset. From there, the fairness metric in question is calculated for the sample predictions. Finally, if the fairness metric is below a certain threshold, the model is returned, else, the algorithm takes its next turn.

6 Results & Discussion

Table 1 shows the range of results achieved by using a cross-validation grid search approach to the baseline classifier. The grid search spanned across the types of kernels used (linear, rbf) and regularization strength. This grid search was performed to estimate how well a classifier can do for accuracy and DDP scores, before trying complex algorithms such as SearchFair from Lohaus et al.

Table 1: Cross-Validation Grid Search Ranges for Baseline Classifier on Sensitive Attribute 'Sex'

Metric	Range
Accuracy	75.4% - 82.6%
DDP	0.1745 - 0.2219

Table 2 shows the results of the baseline unconstrained model (not optimizing fairness metrics) by choice of kernel and sensitive attribute.

Table 2: Unconstrained baseline results

	Linear Kernel			RBF Kernel		
	DDP	DEO	.Acc.	**DDP**	DEO	.Acc.
Sex	0.251	0.182	80.8%	0.238	0.186	81.2%
Race	0.096	0.014	80.7%	0.082	0.041	81.0%

Tables 3 and 4 summarize the results by choice of kernel and sensitive attribute while optimizing for DDP and DEO respectively.

*Table 3: SearchFair results while optimizing for **DDP***

	Linear Kernel			RBF Kernel		
	DDP	DEO	Acc.	**DDP**	DEO	Acc.
Sex	0.0	0.0	76.1%	0.0	0.0	76.1%
Race	0.0	0.0	76.1%	0.026	-0.056	80.3%

*Table 4: SearchFair results while optimizing for **DEO***

	Linear Kernel			RBF Kernel		
	DDP	**DEO**	Acc.	DDP	**DEO**	Acc.
Sex	0.0	0.0	76.0%	0.237	0.190	80.6%
Race	0.096	0.017	80.7%	0.148	0.232	80.7%

For comparison, Table 5 shows the results from Figure 2 of the Too Relaxed to be Fair Supplementary paper. Note that Table 5 does not show results for race, nor does it show DDP scores while trying to optimize DEO and vice versa, because Lohaus et al. do not show these results.

Table 5: SearchFair results from Lohaus et al.

	Linear Kernel		RBF Kernel	
	DDP	Acc.	DDP	Acc.
Sex	0.02±0.02	0.81±0.01	0.01±0.01	0.83±0.00
	DEO	Accuracy	DEO	Accuracy
Sex	0.01±0.01	0.83±0.00	0.02±0.01	0.84±0.00

Although not identical, results are quite similar. For DDP and DEO metrics using the linear kernel, the same margin of error as Lohaus et al. is achieved, but with lower accuracy scores (83% vs. 76%). Using the RBF kernel, the fairness metrics and accuracies are worse than Lohaus et al. In general, however, within the team's results and

Lohaus et al.'s results, models with the RBF kernel perform better than those with a linear kernel. This is likely due to the rbf kernel being able to introduce nonlinearities to help model the data. The team believes that the minor differences in accuracies between reproduced results and Lohaus et al. 's could either be due to minor errors and differences in implementation, and due to a discrepancy between Lohaus et. al's paper and their code. Namely, they report in the supplementary paper that 10,000 data points are used for training, but their program on GitHub shows only 1200 being used. Consequently, the team also uses 1200 data points for training and the remainder for testing, rather than using 10,000 for testing due to memory issues.

In analyzing these results, it is imperative to note that the original Adult dataset is heavily skewed towards the low income class; 37,155 out of 48,842 data points (76.1%) in the dataset belong to the low income class. Assuming the random split of training and testing data did not change this proportion within each respective set, then it is clear that all of the perfectly fair models (both DDP and DEO 0) are no more accurate than naively predicting the majority label in the dataset. This is a significant pitfall of the SearchFair model, since this is essentially a case where it does not manage to learn anything useful.

Since the Adult dataset is one of the most imbalanced ones compared to other datasets Lohaus et al. tested on, the team hypothesized that this imbalance could have been what was causing the poor results, and therefore decided to use a subset of the Adult dataset evenly split between the two income classes (i.e. only using 11687 data points per class). The results by choice of kernel and sensitive attribute while optimizing for DDP and DEO are shown in Tables 6 and 7 respectively, and were obtained using the best hyperparameters from the grid search procedure described in Section 5.2. Table 8 shows the results of the unconstrained optimization on this balanced dataset.

*Table 6: Balanced SearchFair results while optimizing for **DDP***

	Linear Kernel			RBF Kernel		
	DDP	DEO	Acc.	**DDP**	DEO	Acc.
Sex	0.046	0.007	63.6%	-0.002	-0.077	63.0%
Race	0.013	-0.063	72.3%	0.033	-0.051	73.6%

*Table 7: Balanced SearchFair results while optimizing for **DEO***

	Linear Kernel			RBF Kernel		
	DDP	**DEO**	Acc.	DDP	**DEO**	Acc.
Sex	0.164	0.014	70.8%	0.255	-0.004	71.9%
Race	0.156	0.042	75.9%	0.144	0.035	76.6%

Table 8: Unconstrained baseline results on balanced dataset

	Linear Kernel			RBF Kernel		
	DDP	DEO	Acc.	**DDP**	DEO	Acc.
Sex	0.473	0.264	75.9%	0.453	0.275	77.1%
Race	0.155	0.042	76.0%	0.162	0.048	76.2%

Although accuracies in Tables 6 and 7 are lower than the counterparts in Tables 3 and 4, these results are significantly better. All exceed random chance, and in some cases are quite close to the unconstrained accuracies of 76% for sex and 75.9% for race (using the linear kernel). Notable, for race, SearchFair's results for accuracy and fairness metrics are the exact same as the unconstrained model. Thus, in trying to correct for the issue of an unbalanced dataset, the team has also discovered that SearchFair does not always work to optimize the fairness

7 Conclusion

In this paper, previous advancements and findings by Donini, Wu, and Lohaus et al. in fairness of machine learning models were readapted to reproduce desired results [1-3]. Some issues with the proposed SearchFair model discussed in the "Too Relaxed To Be Fair"paper by Lohaus et al, were also discovered. Primarily, these issues are that it can return the same results of a naive classifier always predicting the majority class, particularly if the dataset is imbalanced. Also, it sometimes returns classifiers that are no fairer than those obtained from the unconstrained optimization problem.

An important extension of the algorithm would be to use multiple sensitive attributes at a time. This could be done by repeating a form of equation (2) for each sensitive attribute, and then adding additional terms to equation 4 corresponding to the fairness constraint for each particular sensitive class. This would help address the issue that sensitive attributes are often codependent. It is a valuable next research step to consider intersectionality, occurring when individuals identify with multiple marginalized groups [7].

8 END SECTIONS

ACKNOWLEDGMENT

The authors wish to thank Prof. Elias Khalil, and Prof. Scott Sanner for their continued efforts in motivating the field of Machine Learning. As well, the authors also wish to thank Lohaus, Wu, and Donini for their continued work and research in fairness.

CONTRIBUTIONS OF TEAM MEMBERS

FD - First Draft, MR - Major Revision, ED - Edit, CM - Completed, AS - Assisted

Task	Marc A.	Ita Z.	Chris P.	Farhan W.
Literature Review	CM	CM	CM	CM
Project Proposal	FD, ED	FD, ED	FD, ED	FD, ED
Implementation			CM	AS
Presentation	FD, MR	FD, ED	FD, ED	FD, MR
Dataset		CM	AS	
Final Report				
1. Introduction	MR	FD	ED	ED
2. Motivation	FD, MR	FD	ED	ED
3. Goal of Implementation	FD	ED	ED	ED
4. Dataset Exploration	ED	FD, CM	ED	ED
5. Implementation	ED	ED	FD	ED
6. Results & Discussion	ED	ED	ED	FD
7. Conclusion	FD	ED	ED	ED

REFERENCES

[1] M. Lohaus and M. Perrot, "Too Relaxed to Be Fair", 2020. [Online]. Available: http://proceedings.mlr.press/v119/lohaus20a/lohaus20a.pdf. [Accessed: 29- Mar- 2021]

[2] M. Donini, L. Oneto, S. Ben-David. "Empirical Risk Minimization under Fairness Constraints", 2018. [Online]. Available: https://arxiv.org/pdf/1802.08626.pdf. [Accessed: 29- Mar- 2021].

[3] Y. Wu, L. Zhang and X. Wu, "On Convexity and Bounds of Fairness-aware Classification", Csce.uark.edu, 2019. [Online]. Available: http://www.csce.uark.edu/~xintaowu/publ/www19.pdf. [Accessed: 29- Mar- 2021].

[4] "sklearn.base.BaseEstimator¶," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.base.BaseEstimator.html. [Accessed: 15-Apr-2021].

[5] The CVXPY Authors, "CVXPY 1.1: Convex optimization, for everyone.," *Welcome to CVXPY 1.1 - CVXPY 1.1.11 documentation*. [Online]. Available: https://www.cvxpy.org/. [Accessed: 15-Apr-2021].

[6] M. Lohaus, "mlohaus/SearchFair," *GitHub*. [Online]. Available: https://github.com/mlohaus/SearchFair. [Accessed: 16-Apr-2021].

[7] A. Coleman, "What's Intersectionality? Let These Scholars Explain the Theory and Its History", *Time*, 2018. [Online]. Available: https://time.com/5560575/intersectionality-theory/. [Accessed: 27- Apr- 2021].

MERIH ATASOY, BASc. Industrial Engineering, Specialization in Operations Research, Minor in Machine Learning, Certificate in Business, *University of Toronto.*

CHRIS PALUMBO, BASc. Industrial Engineering, Specialization in Operations Research, Minor in Machine Learning, *University of Toronto.*

ITA ZAPOROZHETS, BASc. Industrial Engineering, Specialization in Operations Research, Minor in Machine Learning, *University of Toronto.*

FARHAN WADIA, BASc. Mechanical Engineering, Specializations in Mechatronics and Bioengineering, Minor in Business, *University of Toronto.*