Evaluating the Effect of Data Missingness on Fairness of Machine Learning Algorithms

Yuri Bukhradze , Gyuwan Kim, Matthew Yom, Alec Panattoni

March 2023

Abstract

One of the most common ways that biases can appear in machine learning is data missingness, which is the defined as the presence of data that is not stored for the variable of interest. In particular, we will be looking at selection bias. We begin by first establishing our parameters and definitions, defining our definitions of what it means to be fair in the context of machine-learning. Then, we define the different types of missingness, focusing particularly on Missing at Random (MAR), Missing Completely at Random (MCAR), and Not Missing at Random (NMAR). For our project, we will create semi-synthetic data to replicate each type of missingness and handle such missingness by dropping missing tuples. With this, we would like to observe how fairness notions/accuracy change as we handle the synthetically produced types of missingness.

1 Introduction

Fairness in machine learning is not easily defined. In fact, there are several definitions, and all have one goal in mind; addressing biases present in the data so that discrimination is not reflected in machine learning models. For example, group fairness distinguishes if any group of individuals is being discriminated against, while individual fairness determines if similar individuals are treated similar by the ML model. In the case of this paper, the group fairness definition will be used. Enforcing fairness is vital due to the increasing role of machine learning algorithms in many fields of society nowadays: from determining a person's ability to get a loan from the bank, to evaluating a person's tendency to re-offend in criminal court cases. Due to the biases present in historical data due to centuries of discrimination, gender, and racial inequality, modern day algorithms that make use of these data can potentially reinforce these inequalities.

In pursuit of evaluating the ability of machine learning models to accurately produce results without bias, we are set to explore the ability of data missingness – presence of data that is not stored for the variable of interest – to affect the degree at which models are able to adequately respond to biases in

the dataset. We define bias as the failure of our dataset to accurately reflect the reported metrics for the entire population; in particular, we are primarily interested in selection bias, which occurs when the data collection process itself introduces inaccurate presentation of population of interest, thus causing our ML model to produce results that are accurate to the data, but are not necessarily representative of the real world. Selection bias, due to the nature of the process, is prone to leaving out relevant data, thus creating missingness. Hoping to determine how such missing data and patterns of such missingness are able to affect the accuracy of model predictions, we set out to study how different types of missingness and the different ways of handling them affect the ability of the model to make both accurate and fair (i.e. representative of the population in question) predictions.

In order to continue with our discussion, it is important to lay out the definitions of concepts that our crucial to our research; in particular, it is necessary to accurately define algorithmic fairness in machine learning. While fairness can be subjective and have a fairly broad definition, in the machine-learning community fairness is typically defined as the process of correcting and rectifying biased algorithms (occurring as a result of misrepresentation of protected attributes including, but not limited to, race, ethnicity, gender, religion, sexual orientation, disability, and class) in existing machine learning models. Missingness in data can stem from a number of reasons, such as participants' data being intentionally omitted, or participants dropping out of studies, or other reasons. In general, participants whose data is missing tend to be people in marginalized communities. Without such bias mitigating processes, missingness in data can skew the data to make it systematically biased against said communities. An example of a fairness-reliant model that would benefit from the resolution of missingness is a dataset that has multiple features: some belong to the set of sensitive attributes, those that we consider to be subject to bias; and others do not belong to that set, but can cause or be caused by the sensitive attributes. In this case, we assume that the data for the sensitive attribute is not missing. However, missing values for other attributes can be correlated with the sensitive attributes. This is where missingness will come in. It is imperative that we choose our attribute that will contain missingness as one that is, ideally, correlated with both the sensitive attribute and the label outcome attribute (target). This is because we don't want to alter the extremely delicate attributes that are the sensitive and label outcome attributes, which are most likely to contain biases. Therefore, we use an attribute that is neither. For simplicity, we will only be using one sensitive attribute (gender). Improper handling of this missingness can directly affect the outcome of the model, predicting different results for groups that might be misrepresented by the partially missing data. Therefore, it is important for us to resolve the connections between attributes and how the missingness plays a role in the distribution of the data in order to build a model that is able to adequately respond to the existing biases in the data. Otherwise, the algorithm that is not taking missingness into account can produce results that are biased towards a certain group.

As mentioned prior, data missingness is another terminology for the com-

mon phenomenon of missing data, when data is omitted from the dataset for some variables either in a systematic pattern or with no discernible pattern whatsoever. There are three different types of data missingness:

1. Missing Completely at Random (MCAR) occurs when the data missing are independent from the observed and unobserved variables. In other words, the missing data is completely independent from all other attributes, including its own value. Because the missing data is not associated with the dataset, we can manipulate this data by removing the data or imputing the data. Our synthetic dataset featuring MCAR data was generated using a Bernoulli variable given probability p (by default set at 0.2), which determines whether a specific data point would be omitted or not.

2. Missing at Random (MAR) occurs when the probability of the missingness is dependent on some other values in the dataset but not within the values of the column where the data is missing. In this case, it is best practice to probabilistically impute based on the value(s) of the attribute(s) that this column is dependent on. To generate MAR dataset, we used Normal distribution for the generation of probability, passing it further through a Sigmoid function in order to bring it within the range of valid probability. For categorical variables, each value was first randomly associated with a specific probability using normal distribution; for numerical variables, the points were normalized to Z-value. These techniques were applied to the *dependent* column which determines the probability for the *affected* column (where the data is actually being made missing).

3. Not Missing at Random (NMAR) indicates missingness that is dependent on the value of the variable itself. Following the previous example, NMAR missingness would occur if the data about salary for people with lower pay was missing – i.e, the low value of the pay itself causes the data to be missing (for example, due to unwillingness to report low pay). NMAR type of missingness cannot be accurately evaluated statistically and instead requires subject domain knowledge to hypothesize possible causes of missingness – however, it is possible for us to generate an NMAR dataset by creating a probabilistic threshold for the values that will be made missing. The technique for simulation was similar to that described in MAR, but without the differntiation of dependent and affected columns: the probability is calculated on the affected column itself, and this determines the missingness of the value itself, given the definition of NMAR.

When we choose a dataset to use for this project, we can never accurately identify the type of missingness of each attribute containing missingness. We can only speculate as to what type of missingness is present within these attributes. This brings on the question: how can we study the effects of handling missingness for each different type if we can't know for sure which type of missingness an attribute is in the first place? The solution to this issue comes with creating our own semi-synthetic data. In other words, we can produce each type

of missingness for an attribute and guarantee that the attribute is that type of missingness based on how its values are generated. In order to create MCAR missing data, we can randomly drop values within the attribute if a generated probability is less than a chosen threshold. For NMAR data, we must do something similar, in that for each distinct value in the missingness attribute, we must choose a different threshold to designate whether a value in the attribute will be let missing. This way, missingness depends on the value of the attribute itself (and possibly other attribute(s), if the values of others are included in the probabilistic designation). In order to generate a probability based on this value, we must use some kind of function. If the value is categorical, we can use a One-Hot Encoder, and generate a separate probability for each OHE column. If it is a numerical value, we can simply use a sigmoidal function to produce probabilities once our threshold is set. Lastly, we must be able to create MAR data. This is done using the same strategy that was used for creating NMAR data, except we are only using the values of the other attribute(s) that the missingness of the attribute is dependent on. In the same sense, we have different values with different thresholds within the other attribute(s), which will determine whether each value of the missingness attribute is indeed missing.

For reasons of simplicity, the missingness attribute will remain the same for each type of missingness. In other words, the same column will be used for creating each type of missingness. In order to test how handling different types of missingness affects fairness notions, we'll have to, unfortunately, choose one method of handling. This is not ideal, since best practice is to handle each type of missingness differently. Since we don't know what kinds of biases exist in the data, we can not be sure that any form of handling missingness will handle it properly. While we could handle the missingness in multiple ways and compare the results, the results could prove to be different for each way the missingness is handled. This could get out of hand since we just want to focus on how fairness notions and accuracies are different amongst the different types of missingness. For this reason, we will have to choose one way of handling missingness and stick to it. Although it may skew the distribution of our data in some way, we will drop all tuples that contain the missing attribute for each type of missingness. This seems to be the best option since imputing violates NMAR best practices (although dropping tuples does violate MAR best practices). To put it another way, we feel that dropping missing tuples will skew the data less than imputing the data potentially would.

In an attempt to decrease the negative effect of biases in the data, several models have been proposed that strive to increase the fairness potential of algorithms like these. The prior work that has been done in this field has introduced several competing approaches to pre-processing of the data, in-processing during the algorithm execution, and post-processing of the resultant predictions; these approaches make use of mathematical and statistical tools to repair discrimination and biases reflected in the dataset. It is important, in our own model, to include fairness intervention technique(s) to give our model a higher likelihood of being "fair". This is because the goal is to see how different types of missingness and the way that they are handled affects a *fair* model. In or-

der to address this, Adversarial Debiasing, an in-processing technique, will be used. This fairness intervention technique works to disconnect the possibility of multi-valued dependencies, which would occur with the model picking up on correlation(s) between non-sensitive attributes and sensitive attributes. The goal, with its use, is to reduce the influence of the sensitive attribute on the outcome of the model. It is important to mention that the use of this technique does not assure that the model is fair. Not only can it fail to completely handle the bias it aims to address, but it also does not handle many other forms of bias that can exist in the data. More specifically, these intervention techniques only handle confounding bias.

1.1 Description of Data

The Civilian Complaints Against the NYC Police Officers, available on Kaggle.com. The dataset has been previously cleaned in a separate EDA notebook, available on Github.

The dataset is a collection of allegations against NYPD officers over the last 35 years. There are just over 33,000 samples in the dataset and 31 columns, after adding a few during EDA. All allegations in the dataset have been investigated by a separate entity known as the Civilian Complaint Review Board (CCRB), who are supposed to be unbiased reviewers of each case. Each allegation has a determination outcome, which says whether or not the officer was guilty of the claim against them. Some of the notable features in this dataset include: complainant gender, complainant ethnicity, officer gender, officer ethnicity, and substantiated. The substantiated column is an example of a column added to the dataset in the cleaning process of the data. The cleaning process is described below:

The following was done to clean the data. Firstly, suspicious forms of missingness were addressed in the data. One column in which this needed to happen was with complainant ages. Ages cannot be below 0, so nans replaced negative ages. Some of the complainant ages were also found to be between ages 1 and 10. Assuming that a complaint could not be filled out by a minor of such age, we converted ages 8 and below to `nans`, indicating missing values. The last form of hidden missingness found in the data was with precinct values of 0 and 1000. After doing external research, it was found that no 0th or 1000th precinct exists in NYC. Therefore, these could be filled with nans. Lastly, the "Substantiated" column was added to the data. This column is a series of booleans, which tells whether each allegation was found to be true or not after investigation (derived from board disposition column).

This dataset is suitable for the purpose of this paper because it contains protected attributes for the complainant; notably their gender and ethnicity. With this, it's very possible that a model could learn existing bias in the data and incorrectly predict future outcomes as a result. In the case of this paper, the model will attempt to predict whether the accused officer will be substantiated for their accusation after being investigated by the board. This dataset is also appropriate to use because of the context of the data. We are all very familiar

with the history of violence and discrimination by police officers in America and how punishments against them are handled. Fairness intervention techniques could prove to be very useful given such context. Therefore, this proves to be an excellent dataset choice to experiment with fairness notions and methods.

2 Methods

To evaluate the effect of missingness on fairness, we need to prepare a synthetic dataset that features a predictable pattern of missingness, allowing us to connect the design of missingness and the fairness notions directly. The following ways are generated using the original NYPD dataset with all missingness removed beforehand

2.1 MCAR

Although the coefficient is more significant and is orthogonal to both observable and unconsidered factors, the probability distribution for MCAR data may be approximated using only the measured values. Maximum likelihood prediction is a typical method of estimating the likelihood function.

Let X be a randomized variable that represents the measurements, and let R be a binary random process that indicates whether or not a report is absent, with R = 1 showing missingness and R = 0 representing non-missingness. It is possible to express the joint probability distribution of X and R as:

$$P(X = x, R = r) = P(X = x|R = r)P(R = r)$$

Since the data are MCAR, we have $P(R = r) = P(R = r|X = x)$, which is the equivalent for all values of x. Consequently, we can write:

$$P(X = x, R = r) = P(X = x|R = r)P(R = r|X = x)$$

The probability function for the experiential data is then:

$$L(X) = \Pi P(X = x|R = 0)P(R = 0|X = x)$$

Where the product is calculated across all received datasets and gives a particular probability distribution for the observational reality, such as an ordinary or Poisson distribution, the delivery elements can be estimated by maximizing

the confidence intervals. This is possible with numerical optimization techniques. Instead, Bayesian methods may be utilized to calculate the confidence interval by providing a random attribute variable and then to update it using the measurements. The cumulative distribution function can then conclude the parameters and provide forecasts regarding fresh inputs.

2.2 NMAR

Even after conditioning on observed data, when data are not absent at random, the probability of missingness depends on the unobserved values of the variable of interest (NMAR). In this scenario, it is impossible to estimate the probability distribution of the observed data using only the observed data.
The conditional distribution can be estimated using a model that specifies the relationship between observed and missing data. Let Y be the variable of interest, and let R be a random binary variable indicating whether or not Y is missing, with $R = 1$ indicating that Y is missing and $R = 0$ indicating that Y is not. Let X represent the collection of observable variables connected to Y and R. The joint probability distribution of Y,X, and R can be expressed as:

$$P(Y = y, X = x, R = r) = P(Y = y|X = x, R = r)P(X = x, R = r)$$

The maximum likelihood or Bayesian techniques can be used to estimate the conditional distribution of Y given X and R, assuming a specific model for $P(Y|X, R)$ such as a linear regression model or a generalized linear model. If the model for $P(Y|X, R)$ is well specified and the missingness process is handled appropriately, this method can give estimates of the probability distribution of Y that are independent of model assumptions. However, attention must be taken to analyze the sensitivity of the results to different models and assumptions.

2.3 MAR

Missing at random (MAR) is a technique for omitted variables in which the chance of misclassifying a variable depends on other dependent variable, but not on the variable's value. This implies that the probability of a variable's absence is not based on the variable's value, but rather on the importance of other variables. Formally, suppose we have a dataset with variables Y and X, where Y is the dependent variable and X is a collection of dependent variables. Let R be a nonlinear function denoting the absence or presence of Y, with $R = 1$ indicating absence and $R = 0$ indicating absence. If the probability of missingness depends purely on the observed variables X, the data are MAR:

$$P(R = 1|Y, X) = P(R = 1|X)$$

This indicates that the chances of Y being absent are autonomous on Y and,

therefore, only dependent on the recorded variables X. Given the MAR concept, the distribution of Y can be modeled using the measured variables X and the quasi values of Y. Using a maximum likelihood estimate with an acceptable weighting factor to adjust for incomplete data is a frequent method. In a regression model based on linearity, for instance, the confidence function can be expressed as:

$$L(\beta, \sigma^2|Y, X) = \Pi f(Y_i|X_i, \beta, \sigma^2)^{(1-R_i)} \times \Pi f(Y_i|X_i, \beta, \sigma^2)^{R_i}$$

Y is the vector of reported and missing data, X is the matrix of confounders, is the column of coefficient of determination, is the variability of the error term, and R is the vector mentioned in literature indicators. The old estimator considers observed data through the first meets definition and incomplete data through the second product term. With the observable values and characteristics, missing data are approximated by incorporating the missing data's regression line. Bayesian techniques can also resolve incomplete data under the MAR assumption. In this scenario, a prior distribution is provided for the parameters and missing data, and the likelihood function is changed using the observed values. The likelihood function can be utilized to draw inferences about attributes and incompleteness and make predictions for new data. The MAR assumptions are practical and flexible for handling missing data. Still, it necessitates a thorough examination of the mentioned literature mechanisms and the form of the stochastic process of the incomplete data. It is required to conduct susceptibility assessments to examine the results' resiliency under different expectations.

2.4 Dataset Repair

After creating a synthetic dataset with the missingness patterns described above (plus a dataset featuring no missing values), we apply dataset repair techniques to approach the issue of fairness. We are using the AIF360 library, specifically, the in-processing algorithm called Adversarial debasing; with this, we can break the dependencies between variables in the dataset that constitute bias. In other words, the algorithm is designed to be a part of a data science pipeline as a step that prepares the data for prediction. The preparation is done by removing biased dependencies in the dataset, thus allowing for fairer decision-making. AIF360 is a Python package for fairness in machine learning that consists of multiple approaches for dataset repair.

• AIF360 supports numerous statistical outlier identification methods, including z-score, modified z-score, and Mahala Nobis distance, which can be used to find and correct outliers in the dataset.
• AIF360 presents an adversarial debiasing method that may correct bias in a dataset by training a classifier to differentiate between the original dataset and a modified dataset with updated preferences.
• AIF360 includes a differential impact remover method that may eliminate

dataset bias by modifying the protected property's probabilities. In addition to data normalization, feature selection, and feature scaling, AIF360 supports different ways of repairing datasets.

2.5 Fairness Notions

This model's methodology for finding fairness notions is implemented using a combination of one-hot encoding (OHE), Convoluted Neural Network (CNN), and fairness metrics. The dataset has been divided into training data and test data. The training data is used to fit the adversarial debiasing model, and the test data is used to evaluate the model's fairness. One-hot encoding is applied to the categorical variables in the training data using the One Hot Encoder from the scalar library; the process transforms categorical variables into numerical variables by creating a new binary column for each unique category. The OHE then fits the training data, and the resulting transformed data is concatenated with the non-categorical features. Next, a adversarial debiasing model is fit for the training data. The adversarial debiasing model is used to predict the target variable, " substantiated," based on the OHE-encoded categorical variables and the non-categorical variables. The model is fit using the" balanced" class weight, which helps balance each class's significance in the target variable. Once the model is appropriate, the OHE-encoding is applied to the test data, and the non-categorical variables are concatenated with the OHE-encoded categorical variables. The adversarial debiasing model is then used to predict the test data. Finally, three fairness metrics are calculated to evaluate the model's fairness. These metrics are the statistical parity difference, the equality of odds difference, and the equality of opportunity difference. Statistical parity difference measures the difference in mean prediction outcomes between two groups. Equality of odds difference measures the difference in odds of positive prediction outcomes be- tween two groups. Equal opportunity difference measures the difference in accurate positive rates between two groups. In machine learning models, categorical variables are expressed as numerical features using one-hot encoding. Fairness concepts are restrictions or criteria used to ensure the impartiality and fairness of machine learning models. One-hot encoding may be used to implement the following ideas of fairness in machine learning models: Demographic parity is a concept of fairness that requires a model's predictions to be independent of the protected attribute. One-hot encoding can be used to generate unique binary features for each category of the protected characteristic; the model can then be trained using these binary features and other variables to predict the target variable. Equalized odds: Equalized odds is a fairness notion that requires the model's predictions to have the same false positive and false negative rates for all groups described by the protected characteristic. One-hot encoding can generate different binary features for each category of the protected characteristic and the target variable. The model can then be trained using these binary features and other features to predict the target variable. Counterfactual fairness is a concept that assumes the model's predictions would

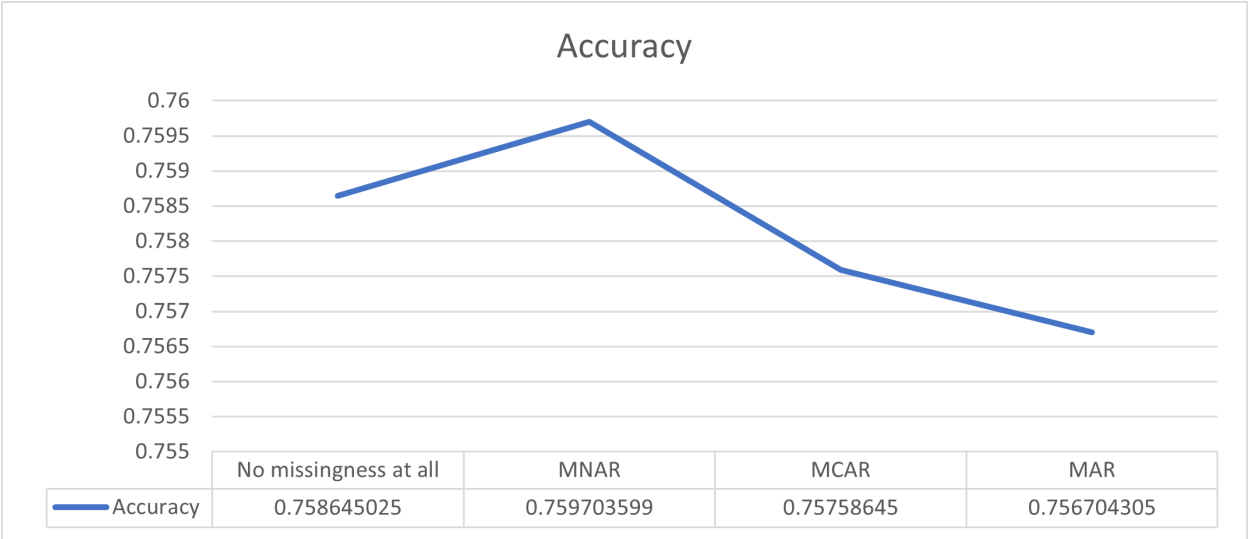
not change if the protected property were changed. One-hot encoding can be used to generate unique binary features for each category of the protected characteristic; the model can then be trained using these binary features and other variables to predict the target variable. Individual fairness is a justice concept requiring individuals to be treated similarly. One-hot encoding can generate unique binary characteristics for each protected attribute and target variable category. The model can then be trained to predict the target variable using these binary features and other features while considering individual similarity. Thus, one-hot encoding can incorporate several notions of fairness into machine learning models, ensuring that the models are fair and impartial for all groups described by the protected attribute.

3 Results

3.1 Fairness notions

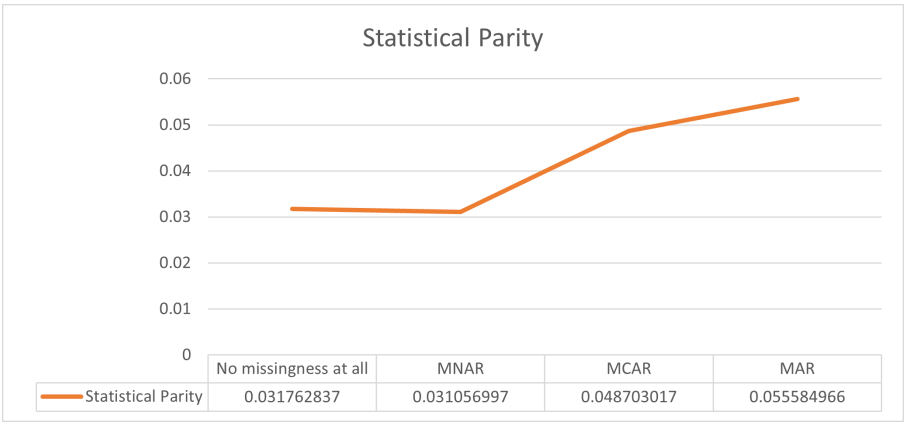
Fairness concepts are the principles or criteria used to assess the fairness of a decision or procedure. The principle of fairness implies that all persons have equal opportunity for success, irrespective of color, gender, religion, or any other personal attribute.

Fairness Notions	No missingness at all	MNAR	MCAR	MAR
Accuracy	**0.758645025**	0.759703599	**0.75758645**	0.756704305
Statistical Parity	0.031762837	0.031056997	0.048703017	0.055584966
Equality of Odds	0.048198929	0.047963165	0.07006776	0.078650963
Equality of opportunity	0.081100141	0.08180536	0.112834979	0.124823695



3.2 Statistical Parity

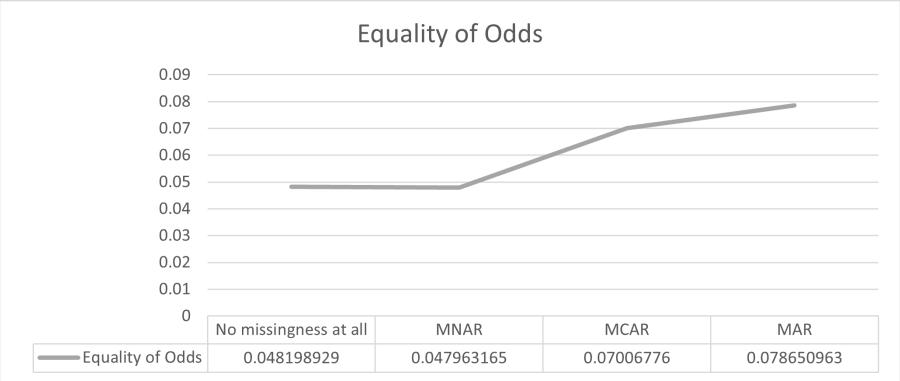
Statistical Parity is the measurement of the difference between the majority in a class and a protected class, and is regarded as one of the most important facets of a fair, unbiased machine learning model. For example, in a fair machine learning model on NYPD arrest data, true statistical parity would predict an equal rate of arrest for all races. The goal for Statistical Parity is to obtain a result as close to 0 as possible.



3.3 Equality of Odds

Equality of odds is a definition of fairness that demands the likelihood of an either a positive or negative forecast for a given group to be identical to the probabilities of the same judgment for the other grouping. In other words,

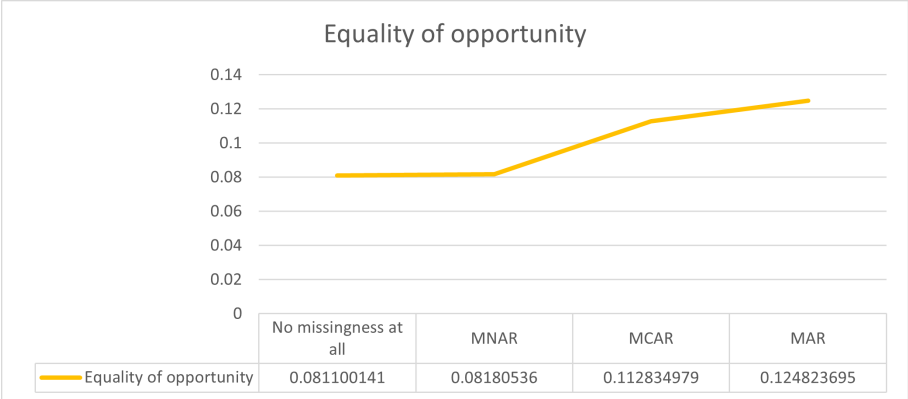
this principle of fairness requires that the predicted accuracy of a model be the same for all individuals, irrespective of race, gender, age, or some other personal feature. The goal for our project was to get the equality of odds as close to zero as possible, as the closer to zero the more fair the model indicates it is.



	No missingness at all	MNAR	MCAR	MAR
Equality of Odds	0.048198929	0.047963165	0.07006776	0.078650963

3.4 Equality of Opportunity

Equality of opportunity is the idea that everyone should be able to quality for an opportunity, regardless of what class they belong to, whether it be class, race, gender, sexuality, or any other metrics. It checks whether a classifier predicts that in a certain attribute, everyone of all values for that attribute get equal treatment. Equality of opportunity is widely considered an essential element of fairness in machine learning, as it ensures that people are evaluated in terms of their merit and effort, as opposed to their identity or outside circumstances. Like with equality of odds, the goal for our project was to get the model as close to 0 as possible, as the closer to 0 it is the more fair the model indicates it is.



	No missingness at all	MNAR	MCAR	MAR
Equality of opportunity	0.081100141	0.08180536	0.112834979	0.124823695

4 Discussion

In machine learning, data redundancy is a common issue that can greatly impact the precision and fairness of predictions. There are three primary categories of missing information: MCAR (Missing Completely At Random), NMAR (Not Missing At Random), and MAR (Not Missing At Random) (Missing At Random). Any of these sorts of missing data can uniquely affect the impartiality of learning algorithms. The fairness notions of accuracy, statistical parity, equality of odds, and equality of opportunity can be used to evaluate the impact of missingness on fairness in machine learning. The result table shows that the accuracy metric is relatively consistent across all types of missingness, with MCAR being slightly lower than the others. This suggests that missing data may not significantly affect the model's accuracy. However, when we look at the fairness metrics, we can see that the impact of missingness varies significantly. Regarding all of the fairness notions, we can see that MCAR and MNAR perform better than MAR and have lower bias values. This suggests that missing data that is either completely random or not missing at random may impact fairness in these contexts less. It's difficult to say which method is definitively fair and accurate based on the result table alone. Fairness and accuracy are complex and multifaceted concepts that a single metric cannot fully capture. However, based on the provided table, we can make some observations about the relative performance of different methods regarding fairness and accuracy. In other words, we cannot necessarily generalize our conclusions to all datasets in terms of how the different missingness types will affect such fairness notions. This does not undermine our results and conclusions, though, since they can begin to shape ideas around how missingness affects fairness in very similar conditions (e.g. how the missingness is handled). In terms of accuracy, all methods are relatively similar, with minor differences in performance. The accuracy metric ranges from 0.7567 to 0.7597, indicating that all procedures are reasonably accurate in handling missing data. We can see that MCAR and MNAR perform better than MAR for statistical parity and equality of odds, with lower bias values. This suggests that missing data that is either completely random or not missing at random may be fairer in these contexts. Overall, our results matched our logical expectations at the beginning of the project. A pattern of missingness would noticeably affect fairness in a situation where the missingness is dependent on a sensitive attribute; that indicates that a pattern such as MAR, which depends on a different attribute, would likely be the one to cause the most bias in the dataset. Our data supports this, with MAR consistently being the type of missingness to cause the highest divergence from 0 across fairness notions. We also spotted MCAR to behave similarly, which can be explained by the random distribution of missing data across the dataset, thus causing the sensitive attributes to lose points in a random pattern at the same rate as non-sensitive attributes; the fact that sensitive attributes lose points is what determines fairness, so the results for MCAR are consistent with our intuition. In conclusion, types of missingness that depend on randomness (MCAR and MAR) are the ones that caused the most bias. With regards to fairness, one should generally be most concerned

with how MAR affects biases. This is because a non-sensitive attribute can be dependent on a sensitive attribute's outcome, which inherently would contain the most bias. MCAR should not be much of one's concern, since its patterns can be explained completely by randomness. NMAR also should generally not influence the outcome much, assuming it is not a sensitive attribute; although it still should be handled with care.

5 Sources

Mehrabi, Ninareh, et al. "A Survey on Bias and Fairness in Machine Learning." ACM Computing Surveys, vol. 54, no. 6, 2021, pp. 1–35., https://doi.org/10.1145/3457607.

Jeanselme, Vincent, et al. "Imputation Strategies under Clinical Presence: Impact on Algorithmic Fairness." ArXiv.org, 11 Nov. 2022, https://arxiv.org/abs/2208.06648.

Salimi, Babak, et al. "Capuchin: Causal Database Repair for Algorithmic Fairness."ArXiv.org, 1 Oct. 2019, https://arxiv.org/abs/1902.08283.