DISSERTATION PROJECT

St. Xavier's College (Autonomous), Kolkata

Post Graduate Department of Data Science

Studying Classification Models for Income Class and Assessing the Fairness and Bias of the Models in the Context of Adult Census Dataset

NAME: Shamie Dasgupta

ROLL NUMBER: 406

REGISTRATION NUMBER: A01-2112-0814-19

SUPERVISOR: Dr. Mausumi Bose

DECLARATION:

I affirm that I have identified all my sources and that no part of my dissertation paper uses unacknowledged materials.

Shamie Dasgupta

My Signature

iThenticate®

Similarity Report ID: oid:3618:55685027

PAPER NAME

406_MSc_Dissertation.pdf

AUTHOR

Shamie Dasgupta

WORD COUNT

14574 Words

CHARACTER COUNT

76441 Characters

PAGE COUNT

61 Pages

FILE SIZE

1.5MB

SUBMISSION DATE

Apr 4, 2024 3:07 PM GMT+5:30

REPORT DATE

Apr 4, 2024 3:08 PM GMT+5:30

● 9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 9% Internet database
- Crossref database
- 6% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Bibliographic material
- Cited material
- Methods and Materials
- Quoted material
- Abstract
- Small Matches (Less then 15 words)

● 1% words excluded by Custom Sections

CONTENTS

1 ABSTRACT

Fairness is an important concern in present times as machine learning models are used more and more for supporting decision making in high stakes domains like finance, healthcare, and criminal sentencing, etc. Ensuring the fairness of these models is highly necessary so that no discrimination is made between different demographic groups (example- race, sex, age, and others) while using these models for decision making. Different algorithms have been developed to measure unfairness and mitigate them to a certain level.

This paper delves into the exploration, analysis of unfairness and prediction of income using machine learning models applied to the Adult Census dataset. Extracted from the 1994 census database, this dataset, sourced from Kaggle has 14 attributes, 32562 data points and a binary response variable stating whether an individual makes more than 50k dollars annually or not.

The primary objective of this paper is to investigate and mitigate the presence of social bias, particularly concerning race and gender within these predictive models. In this paper, the raw data was first cleaned and pre-processed, then a detailed exploratory data analysis will be performed on the data, following which various machine learning techniques like Logistic Regression, Decision Tree, Random Forest are employed to develop models and check their accuracy and performance. Finally, we propose to evaluate their fairness across demographic groups.

Ultimately this study aims to raise awareness of the presence of bias and unfairness in machine learning models and techniques to quantify and mitigate them.

2 INTRODUCTION & LITERATURE REVIEW

In today's technologically advanced world, Artificial Intelligence is being demandingly used in highly sensitive fields like finance, hiring, criminal justice, health care and so on. It is known that human decision-making in any area is to some extend biased and the decisions taken by human intelligence is shaped by the social constructs and personal believes of the individual, which in often cases may be subconscious. It is a misconception that automated decision-making systems would always produce fair decisions. However, that is simply not the case. Artificial Intelligence bias can come in the form of societal bias often hidden in training datasets, and models built on these datasets often tend to catch these patterns which in turn results in decisions that may seem ‘fair’ but suffers the same way as human driven decisions do (Trisha,Kush,Michael, 2020) .

As more and more prediction-based decision algorithms are being widely utilized by various organizations, a concern has been raised about the fairness of these models, and whether these models are creating biases between different demographic groups particularly race, gender, class, religion, ethnicity, etc. The particular interest in this problem has motivated a special field of research which has been coined ‘algorithmic fairness’ (Mitchell et al., 2021).

Even though algorithmic fairness might be a new concept, historically there have been studies on fairness, like the study of fairness of educational tests based on the ability of prediction performance at school or work (Cleary, 1966).

The reason fairness should be an equal concern alongside accuracy performances is because decision-based systems have an influence on people’s lives, and unfair predictions will raise ethical concerns pertaining to the society (Pagano et al., 2022). In fact, many experts are under the opinion that unwanted social bias might be the major barrier that will prevent Artificial Intelligence from reaching its full potential. In 2016 journalists after investigation found that the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) algorithm incorrectly labelled African-American defendants as ‘high risk’ that too at approximately two times the rate it mislabelled the white defendants (*Machine Bias — ProPublica*). Due to this reason new strategies from machine learning, data science and artificial intelligence on fairness and bias mitigation have been under construction.

There are numerous choices that can cause social concerns but usually they go unmeasured by the fairness evaluations. Some of these choices include the training data, the model built and predictive performance evaluation. They choices are discussed below:

Data is an essential element of Machine Learning. The basis of a predictive model is the data on which it has been trained (Mitchell et al., 2021). Models are built based on the training data and often the dataset contains human decisions that exhibit the societal and historical inequalities. If the reality shown in the data puts certain demographic groups at a systematic disadvantage, then the training data distribution is likely to reproduce that inequality than a fairer future. In terms of algorithmic fairness, data with the undesirable properties that often reflect societal injustice are informally labelled as biased data (Kamiran & Calders, 2009). Bias can be roughly divided into two notions:

Statistical Bias implies the systematic error that affects the data analysis results due to a mismatch between the sample drawn for training a machine learning model and the population as it currently is, the choice of estimators, the analysis methods, or the interpretation of results. Algorithmic fairness is concerned with how sampling bias and measurement errors can introduce unfairness in the data set.

Societal Bias occurs when even after the training data is deemed representative and accurate, they may still contain objectionable aspects of the population. This may particularly happen if the data contains sensitive attributes like gender, race, religion, ethnicity, etc.

After data, the next area where unfairness may be induced is the training of the predictive model. A huge characteristic of machine learning models is that the model makes predictions based on the training dataset and actual outcomes. Thus, any machine learning model will attempt to find patterns in the training data to use when a new instance has to be predicted. However, these patterns that the models try to generalise sometimes are undesirable or even ethically illegal. Furthermore, there are many choices required while building a model- like class of the model, functional form, model parameters, cost functions, etc. These choices too have consequences for fairness.

Finally, Evaluation also plays an important role in algorithmic fairness. Typically, predictive models are evaluated using measures such as mean squared error, sensitivity, F1-score, etc. However, these metrices make some important assumptions. Firstly, they assume that any decision can be evaluated as an aggregation of separately evaluated individual decisions. Secondly, these metrices assume that all individuals can be considered identically (or symmetrically). Thirdly, they assume that the evaluation of these decisions occurs simultaneously. One fundamental step of algorithmic fairness is to what extent the prediction metrices making these assumptions are relevant to the fairness of the predictions.

Keeping all the above studies and research in mind, this paper was constructed to assess various Machine learning models for classification, not only in terms of their accuracy performance, but also in terms of their fairness. Since training data plays a huge role in introducing historical bias to the predictive model, the data set was explored keeping in mind the sensitive attributes like gender and race, which are often a source of ethical concern. Finally, some algorithms were performed to mitigate bias from the classification models in hopes of making them as fair as possible.

3 DATA DESCRIPTION

The aim of this paper is to predict the income class of people using machine learning models in the Adult Census dataset. The adult census data was sourced from Kaggle for its usage in this work, though the data was originally extracted from the 1994 census database. Link of the dataset is: Adult Census Income (kaggle.com)

The Adult Census Dataset has 14 attributes and 32562 data points. The target variable or the response variable is called 'income'. It is binary in nature with two classes <=50k and >50k. The response states whether an individual makes more than 50K dollars annually.

Description on predictors:

- Age: age of an individual
- Workclass: the employment status
- Fnlwgt: the final weight
- Education: the highest level of education
- Educational-num: the highest education achieved in numerical form.
- Marital-status: the marital status
- Relationship: represents what this individual is in the family.
- Occupation: the general type of occupation
- Race: race of the individual
- Sex: female or male

- Capital-gain: the capital gains for an individual.
- Capital-loss: the capital loss for an individual.
- Hours-per-week: the hours an individual has reported to work in a week.
- Native country: the country of origin for an individual

4 OBJECTIVE

The primary objective of this study is to study the classification models, particularly Logistic Regression, Decision Tree, and Random Forest for the target variable Income, trained on the Adult Census data and evaluate and compare its performances in terms of accuracy and fairness. Along with that, this paper aims to investigate the potential sources of societal bias using various exploratory fairness analysis approaches and model fairness metrices. The ultimate goal of this study is to apply different algorithms to mitigate bias in the event the models are deemed 'unfair'.

5 METHODOLOGY

First the Raw data was checked for missing values. After a preliminary study, it was found that the data has some missing values, particularly under the variables *workclass*, *occupation* and *native.country*.

Predictors	Number of missing values
Workclass	1836
Occupation	1843
Native.country	583

Table 1

Since the number of missing values for *Workclass* and *Occupation* are quite close to each other, a look at the dataset rows confirmed that most of the individuals with missing entries for *Workclass* had missing entries for *Occupation* as well. Since the total number of data points is quite high (32562), the individuals with missing entries were removed from the dataset.

After cleaning, the cleaned data was pre-processed. Since the adult dataset has 6 quantitative predictors and 8 categorical predictors, these 8 categorical predictors were restructured for the sake of interpretation and model building-

- The variable workclass had 7 categories- Private, Self-emp-not-inc, Local-gov, State-gov, Self-emp-inc,Federal-gov,Without-pay. Majority of the individuals in the dataset worked in Private workclass (74%). Hence the rest of the categories were merged into 'Others'.
- The variable Race had 5 categories- White, Black, Asian-Pac-Islander, Amer-Indian-Eskimo, and Other. White constituted majority of the individuals in the data set (86%). Hence the other categories were recoded to one single category Non-white.
- The variable education has 16 categories, and they were restructured in the following way:

— HS-grad, 11th, 10th, 7th-8th, 9th, 12th, 5th-6th, 1st-4th, Preschool were merged to one category **HS-level.**

— Some-college, Bachelors, Assoc-voc, Assoc-acdm, were merged to one category **graduation**.

— Masters, Prof-school and Doctorate were merged to one category **above**.

- The attribute Marital status had 7 categories and they were merged in the following way:

— Married-civ-spouse, Married-spouse-absent and Married-Af-spouse were restructured to just '**married'**.

— Never married was changed to '**not married**'.

— Divorced, Separated and Widowed were merged to the category '**other'**.

- The category occupation has multiple labels, which were recoded as below:

— **White-Collar**: Prof-specialty, Exec-managerial, Adm-clerical, Sales, Tech-support

— **Blue-Collar**: Craft-repair, Machine-op-inspct, Transport-moving, Handlers-cleaners, Farming-fishing

— **Service**: Other-service, Protective-serv, Priv-house-serv, Armed-Forces.

After the data was pre-processed, an exploratory data analysis was performed. This step was done to understand the different features of the dataset and identify and visualise the associativity of these features with the target variable. After the preliminary data analysis an exploratory fairness analysis was performed before model training. This an essential step to identify the potential sources of bias before the model was trained. Under Exploratory Fairness analysis, the sensitive attributes were defined and visualised. Then the prevalence values were calculated in terms of the sensitive attributes. To find proxy variables mutual information was calculated and plotted.

Based on the exploratory data analysis, feature selection was performed and the final selected features along with the target variable *Income* was used for training the three machine learning models. For model training the dataset was split into train and test data at a ratio of 7:3. For Logistic regression a significance test was performed to understand the relationships of the features with the target variable. For tree-based models, hyper parameters were tuned with the help of Grid Search technique.

Their accuracy, precision, recall and f1 scores were calculated through the confusion matrix. The models were compared based on the prediction performance.

For fairness, different metrices like Equal Opportunity Difference and Disparate Impact were implemented to find out whether the models could be deemed ‘unfair’. Finally, two bias mitigation techniques – the Disparate Impact Remover and Reweighing were used to help reduce the bias and the three models were compared based on the fairness metrices before and after bias mitigation.

6 COMPUTATION & RESULTS

6.1 Exploratory Data Analysis

The different features of the data were studied and visualized.

Target variable- Income:

The outcome or the target variable has two classes- >50K and <=50K. Here an individual earning above 50 K was considered beneficial, hence this class was denoted as 1 and for income less than or equal to 50K dollars were represented by 0. The response or the target variable is clearly binary in nature.

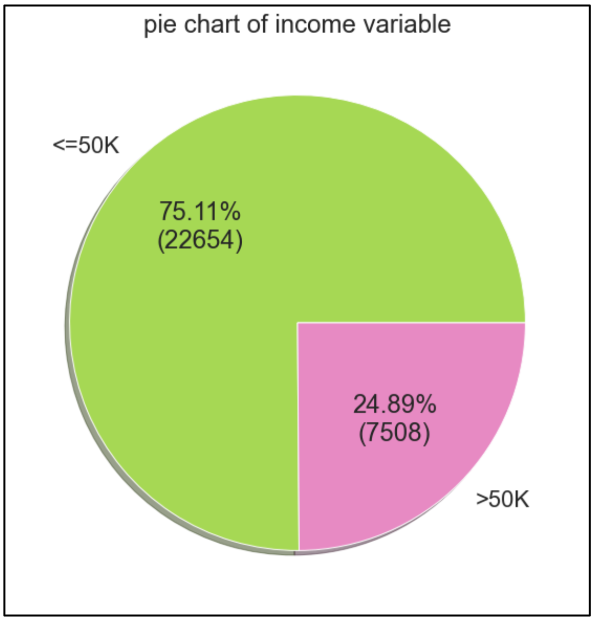

Figure 1

From the pie chart, it was observed that the dataset contains around 25% of entries labelled with >50K and around 75% were labelled with the <=50K dollars category. This means, approximately one-fourth of the individuals in the dataset earn above 50K dollars, while the rest fall in the income class below 50K.

This clearly shows that the data is unbalanced in terms of the target variable.

The uneven distribution of the target class will pose problems while predicting class labels. Here the difference between the number of observations of the two class labels is quite high. The major problem that an imbalanced data can cause during predictions is how accurately we are actually predicting the majority and the minority class; majority implying the class with the higher number of observations and minority being otherwise. The classification models should not be allowed to be biased while predicting decisions, to detect only the majority class. It should give equal importance towards the minority class as well. In fact, the minority class for this data and purpose was more important than the majority class.

Few techniques that can deal with this problem are:

- Changing the evaluation metric, and instead of using accuracy, metrices like precision, recall and F1-score may be used.
- Resampling techniques like under sampling, over sampling or even Synthetic Minority Oversampling Technique (SMOTE) may be implied to balance the class distributions. However, for this study resampling techniques will hinder the actual purpose of the models, because resampling will also

change the distributions of the sensitive attributes. This will give a faulty analysis later. Hence the only focus that can be done to rectify the problem of imbalanced data is using better evaluation metrices.

To gain insights about which features should be included in building the models, the features were visualised and studied:

Age:

Age is a continuous variable representing the biological age of an individual. For a visual inspection a histogram was plotted.

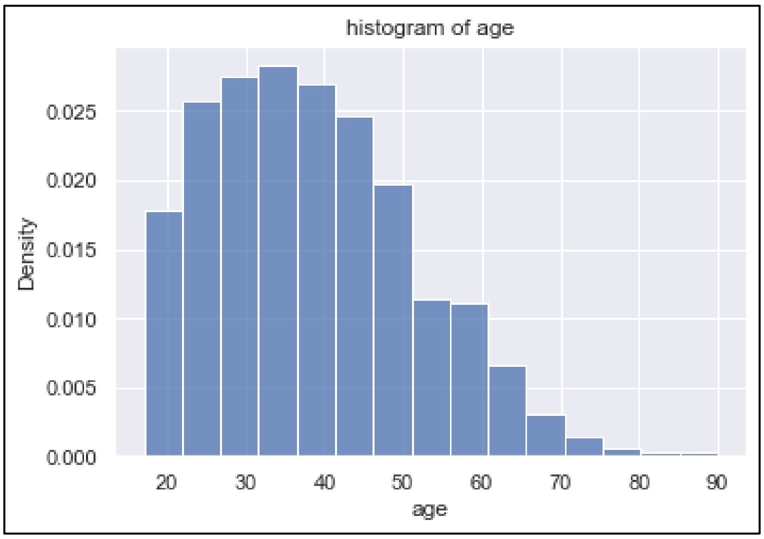

Figure 2

The histogram of the age variable provides some distinct characteristics of the feature age. It can be observed from the histogram above that the age has a unimodal distribution with a peak near 35, that is,most values of age in the dataset will be close to 35. The distribution is positively skewed with its values falling between 17 and 90. The long tail extends to the right and most of the ages of the people are clustered on the left.

The average age of a person in the data is 38. The minimum and maximum age in the data is 17 and 90 respectively.

In the dataset, it is observed that there are 330 people who are of age 17 and working in either private sector or other work classes. We are not considering these data points as anomalous because in the US, people below 18 are permitted to work in certain sectors.

An essential part of the data analysis is to check whether age may influence the target variable income. A box plot of the age variable was plotted with respect to both of the income classes:- <=50K and >50K.

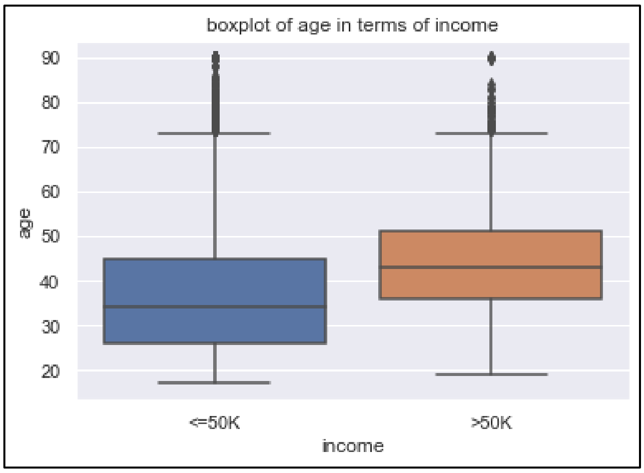

Figure 3

statistics	<=50K	>50K
Minimum	17	19
Q1	26	36
Median	34	43
Q3	45	51
Maximum	90	90

Table 2- 5 point summary

From the boxplots of the respected categories, it can be observed that the median age of people earning more than 50K is 9 years more than the median age of the people earning less than 50K.

The maximum age in both categories is same, that is 90. However, the minimum age of people earning more than 50 K is slightly more than the minimum age of people earning less than 50K.

The boxplot is positively skewed as the median is closer to the 1st quartile and the presence of outliers are heavier on the right side of the median. Also, the length of the upper whisker is more than the length of the lower whisker, this also implies that the boxplot is positively skewed. The number of potential outliers is 197. Even though the distance between the 1st quartile and median and the 3rd quartile and median is very close to each other, the boxplot is slightly positively skewed as there are multiple outliers present beyond the upper whisker. The number of potential outliers is 48.

At the time of model training, these data points will be removed.

Workclass:

The workclass column was restructured due to a number of categories occupying only a small percentage of the data points. ‘Other’ here refers to the rest of the sectors of work class like- Self-employed not inclusive of formal organizations, Local government, State government, Self-employed in formal business, Federal government, without salary.

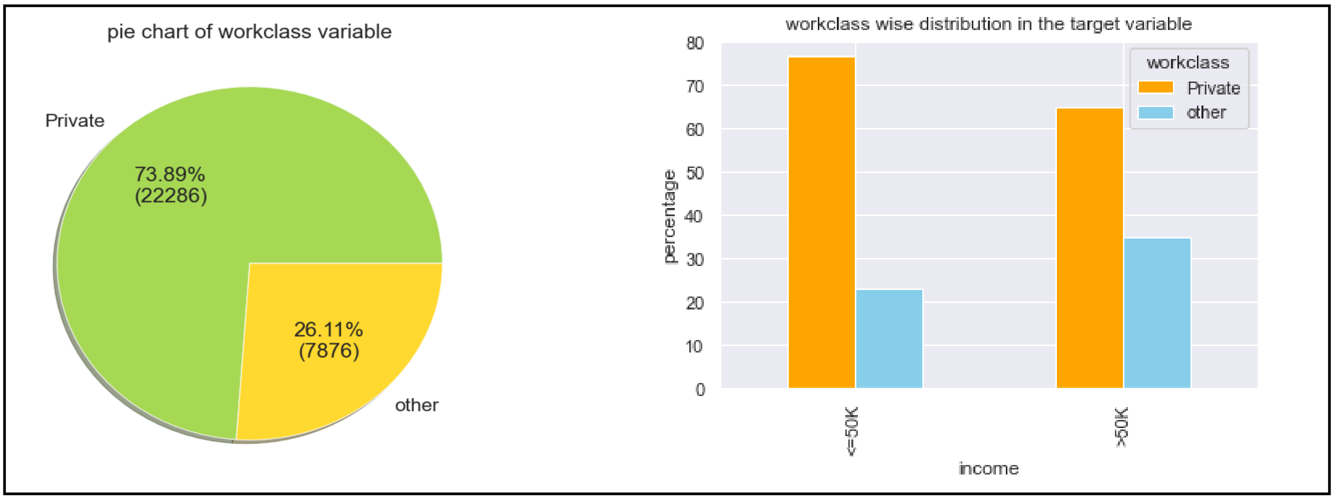

Figure 4

From the graph, it can be observed that approx 74% of the individuals work in the private sector and the rest approx 26% work in other sectors as mentioned above.

From the multiple barplot it is seen that more people working in other sectors like self-employed and government earned above 50K than they earned below 50K. In the Us, Federal government is seen as an elite government in the public sector while self employed with own companies have a very high ceiling when it comes to earnings.

Fnlwgt:

This variable denotes the final weight, that is the number of people the census believes the entry represents. It is a sampling weight that indicates the number of people in the population that each record represents. The weights issued on Current Population Survey (CPS) files are controlled to produce independent estimates of the civilian population of the US. These are basically weighted tallies that indicate specified socio-economic characteristics of the population.

This weight is used to compensate for the unequal probabilities of selection due to the survey design. Essentially, it helps to ensure that the sample is representative of the population it aims to generalize to.

A histogram of the Fnlwgt was plotted to get a visual representation.

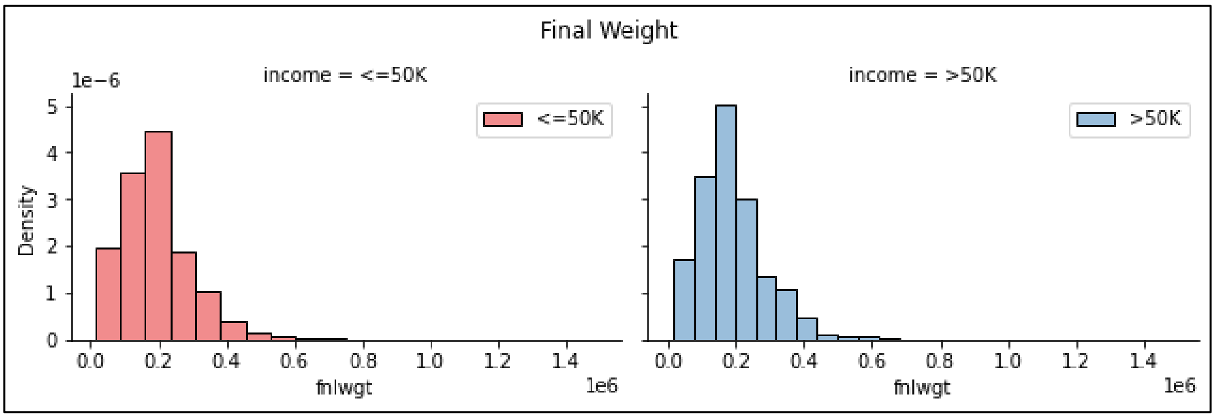

Figure 5

From the graph, it is visible that the final weight variable has a unimodal distribution with positive skewedness. From the two graphs it is clearly visible that the two histograms show almost similar results. In fact, their distribution is more or less same. This implies that the feature fnlwgt may not have any influence on the target variable income.

Education:

Education is a nominal categorical variable denoting the highest level of education achieved by an individual. It had been recoded into 3 categories: HS-level, graduation, and above. For visual inspection a pie chart and multiple bar plot was plotted.

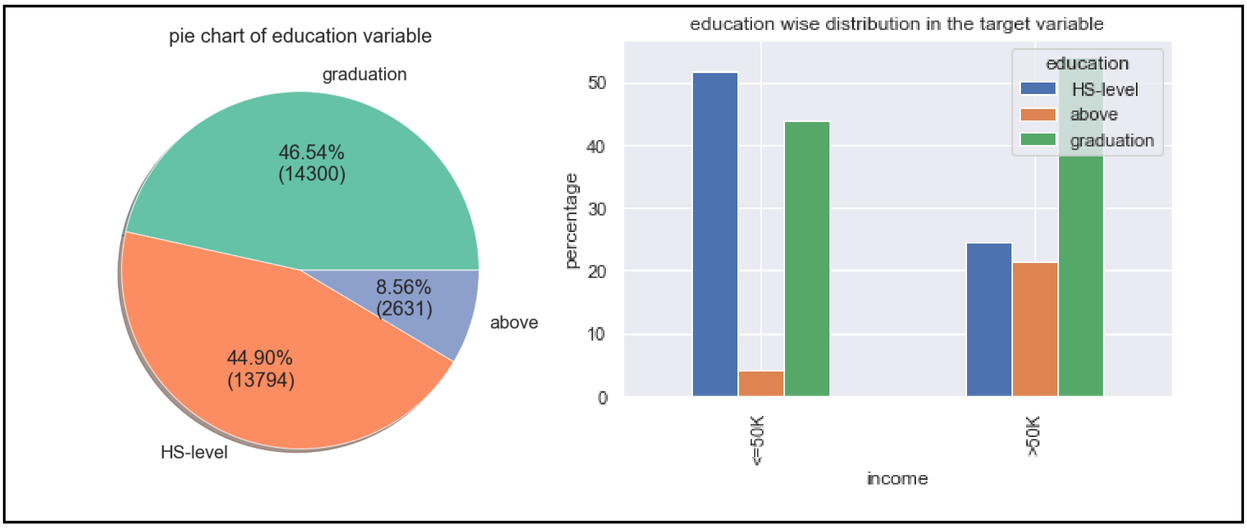

Figure 6

From the pie chart, 46% of the individuals completed their graduation, followed by 45% of the individuals who completed their education till or below HS-Level, leaving only 9% of the individuals who pursued higher education.

From the multiple bar plot, it is clear that people who pursued higher studies were earning salaries in the income bracket >50K dollars. Only a small percentage of people who pursued higher education after graduation earned less than 50K dollars. From the multiple bar plot it may be deduced that education qualification is a significant factor in predicting the income of a person to be more than 50K dollars annually or not.

Marital status:

It is also a nominal categorical variable denoting the marital status of an individual, restructured to the final categories -married, not married and other, with the categories divorced, separated, and widowed.

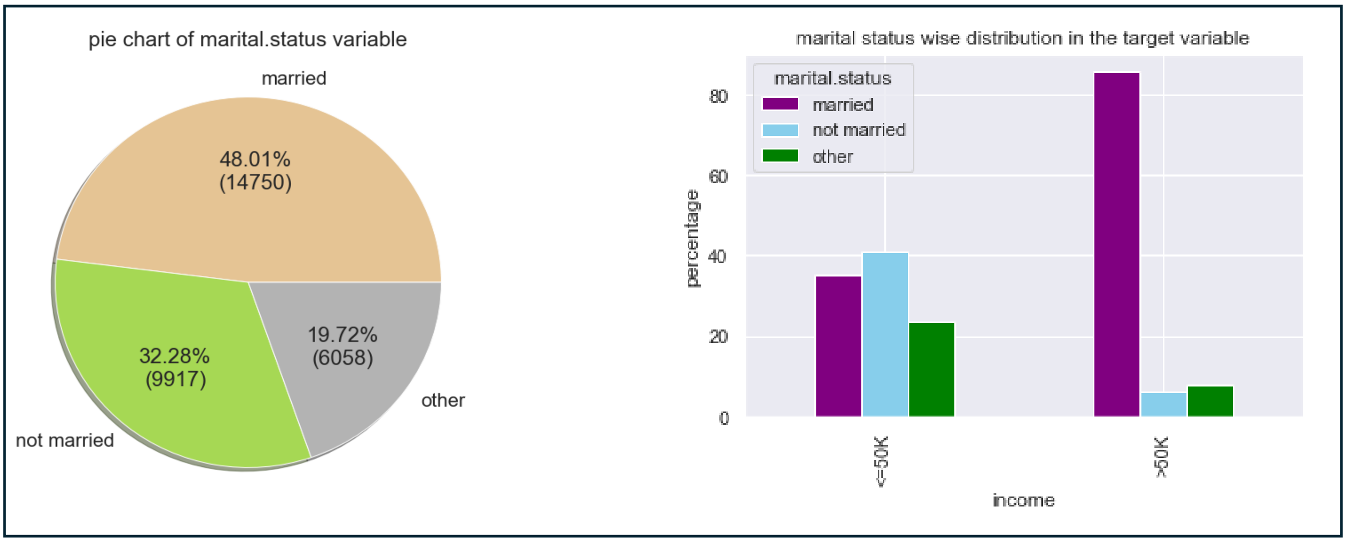

Figure 7

From the pie chart, it shows in the data approximately 48 % of the individuals are married. In fact, roughly half of the people in the dataset have been married, with 32% as unmarried and the rest 20% as divorced, separated or widowed.

From the multiple bar plot it is striking to see that there is a distinct association between marital status and the income class. Most of the married individuals earn above 50k dollars annually. Where areas most of the unmarried and other categories have their incomes in the lower income bracket. This definitely shows that marital status has a clear association with the target variable income and is an important feature to be considered.

Relationship:

Relationship is a nominal categorical variable with 6 classes. It defines the relationship of the individual in his or her marital status.

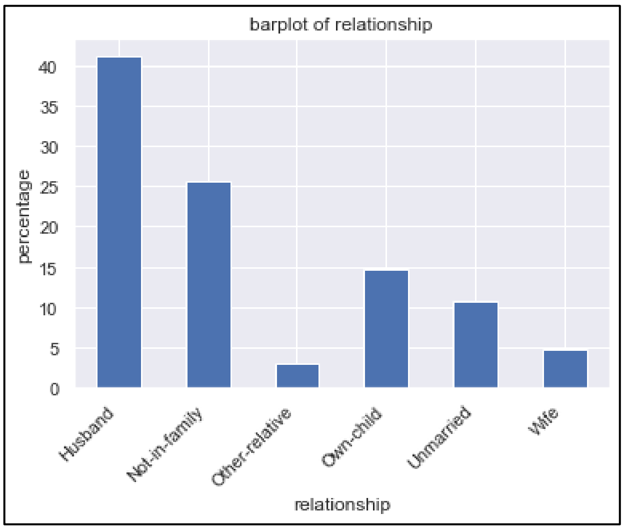

Figure 8

The bar plot gives a basic idea of the demographics of the relationship category in the data. The categories along with the attribute do not provide any significant additional information that can be valuable while predicting the target variable, since marital status has already been selected for model training.

Occupation:

Occupation is a nominal categorical variable which has been restructured into three categories – blue-collar, white-collar and service.

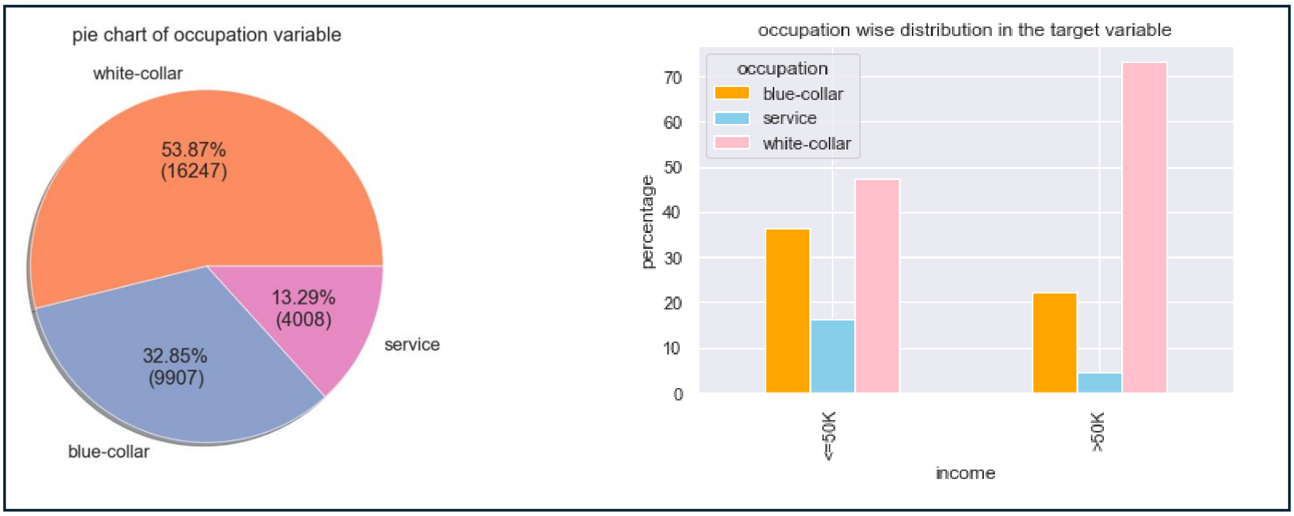

Figure 9

From the pie plot, approx 54% of the individuals have a white-collar job, while a 33% of the people have blue-collar occupaton, followed by only 13% in the service sector.

From the multiple barplot, maximum white-collar individuals earn income greater than 50K, while only a small percentage of the service sector have their income in the higher class. Most of the blue-collar and service sector individuals earn less than or equal to 50K annauly. Clearly, the occupation may have some influence in the income variable, hence it is an important feature to consider for model training.

Race:

Race is also a nominal categorical variable recoded into two categories white and non-white.

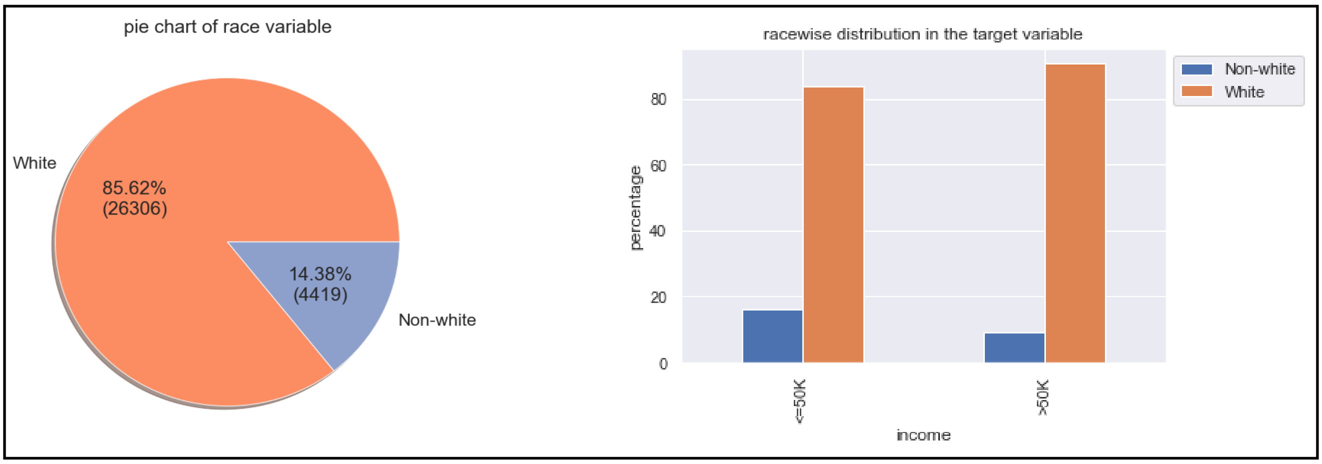

Figure 10

The percentage of non-white individuals is very less, only around 14%,. Majority of the individuals in the census dataset are white. From the multiple bar plot, lesser percentage of non-white people earn above 50K, while greater percentage of white people earn above 50K. Even though the difference between the bars may not seem significant race is an important feature for model training.

Sex:

Sex is a nominal categorical variable recoded into two categories male and female.

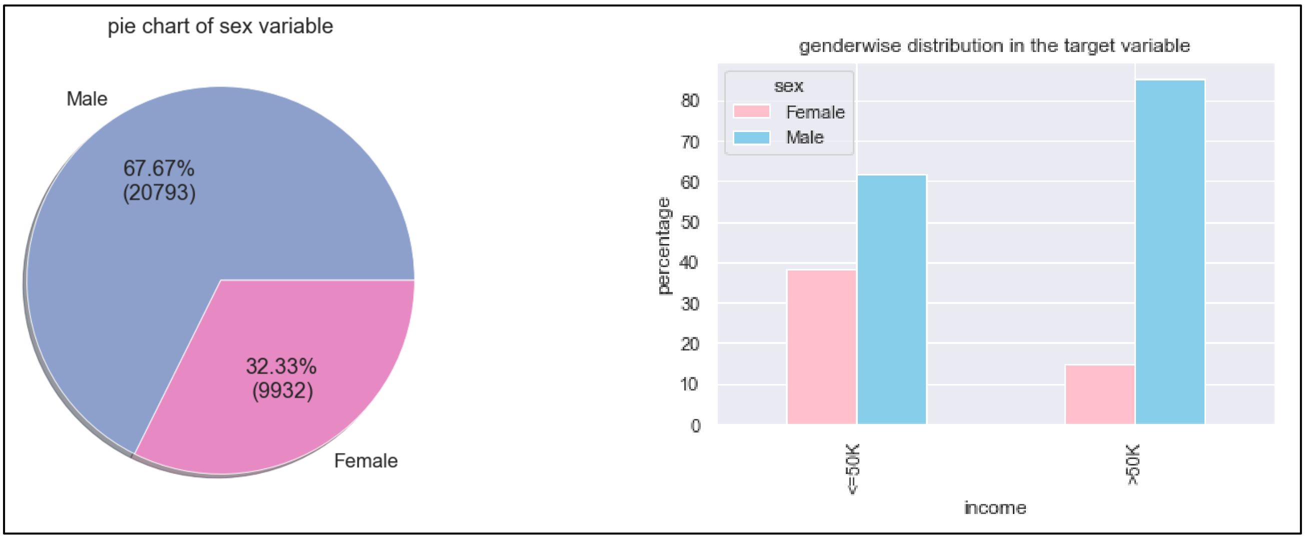

Figure 11

The percentage of males is higher (68%) than the percentage of females in the dataset.

From the multiple bar plot, a small percentage of females earn above 50K, while a greater percentage of males earn above 50K as compared to the bars in the less than 50k category. There is a significant difference between the bars of the two income classes and hence sex can be considered an important feature.

Capital gain:

It is a continuous variable representing capital gains for an individual. A histogram of the variable was plotted.

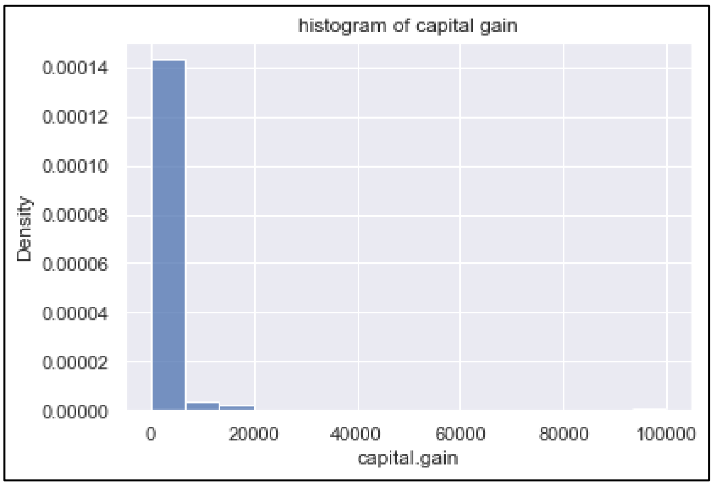

Figure 12

Most of the individuals had a capital gain of 0. The variable indicates the total monetary gain utilized from the capitals of an individual. The histogram is heavily positively skewed (L shaped), with most individuals having zero or low capital gain. In fact, there is almost no data point above 20000 capital gains.

Due to heavy skewness of the variable, and the fact maximum of the individuals has 0 capital gain, this feature was dropped from model building.

Capital loss:

It is a continuous variable representing capital loss for an individual. A histogram of the variable was plotted.

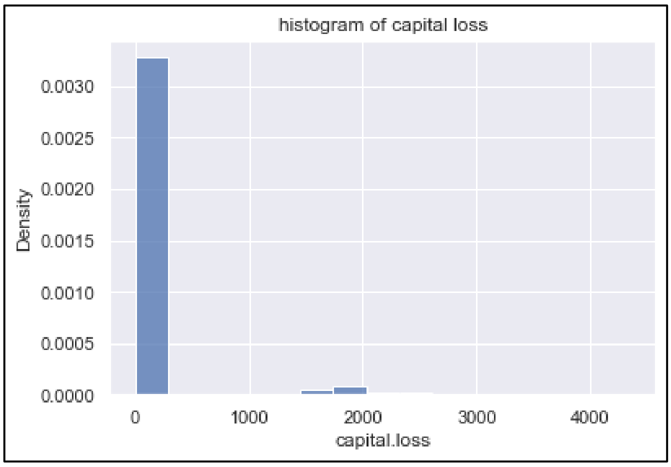

Figure 13

Most of the individuals had a capital loss of 0. The variable indicates the total monetary loss accumulated from the capitals of an individual. The histogram is heavily positively skewed (L shaped), with most individuals having zero or low capital loss.

Due to heavy skewness of the variable, and also the fact maximum of the individuals has 0 capital loss, this feature was dropped from model building.

Hours per week:

It is a continuous variable denoting the hours that an individual has reported to work per week.

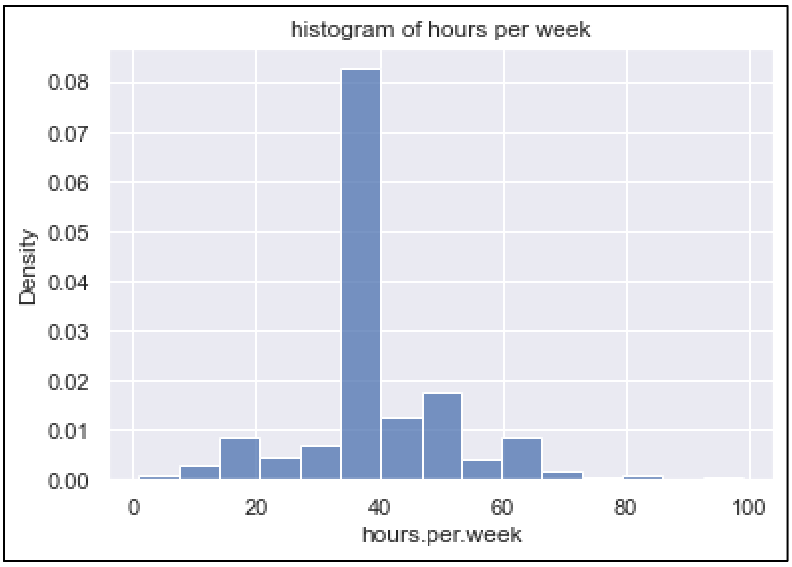

Figure 14

The histogram of the hours per week variable provides some distinct characteristics of the feature. It can be observed from the histogram above that the 'hours per week' variable has a unimodal distribution with only one peak near 38 hrs. Most values of hours per week in the dataset will be close to 38 hours. The distribution is more or less symmetric with its values falling between 1 and 99.

The average hours per week of a person in the data is 40.949. The minimum and maximum hours worked per week is 1 and 99.

From the histogram we can identify some potential outliers, we can see that people working beyond 80 hours per week implies that a person on average works more than 11 hours a day. Even though it looks like a potential outlier, work conditions like that do exist. And the number of people working more than 80 hrs is 198.

There are also people who work less than 10 hrs a week. This means that on an average a person will work only an hour or so in a day.

After the exploratory data analysis, the final features selected are: age, education, marital.status, occupation, race, sex, hours.per.week.

6.2 Exploratory Fairness Analysis

There is a need to identify and understand the potential sources of bias before modelling the data. This is done to build some intuition around the dataset. Essentially, the data needs to be assessed in those aspects that may lead to an unfair model.

First an exploratory fairness analysis is performed to identify the potential biases .

Sensitive attributes:

In machine learning fairness, sensitive attributes refer to those characteristics or attributes of individuals such as race, gender, age, disability status which are considered socially sensitive, or are protected by anti-discrimination laws.

The reason these attributes form a vital part of assessing fairness in ml models is because models trained on data containing such attributes may inadvertently learn or perpetuate biases during prediction.

In this data, it is observed that there are two such sensitive attributes: race and sex. Bias towards the demographic groups based on these attributes will be analysed later on.

Protected features:

Protected features refer to a subset of features in a dataset that includes sensitive attributes that should not be used to discriminate against different demographic groups.

The protected features are defined by creating binary variables using sensitive attributes.

These variables are defined such that 1 represents privileged group and 0 represents an unprivileged group. The categories are classified as privileged and unprivileged based on the historical injustice faced by the unprivileged group in the past. That is, the unprivileged group will refer to those individuals who will most likely face unfair decisions from a biased model.

For race, the protected feature is defined so that White is the privileged group.

For sex, male is considered the privileged group.

Unbalanced Samples:

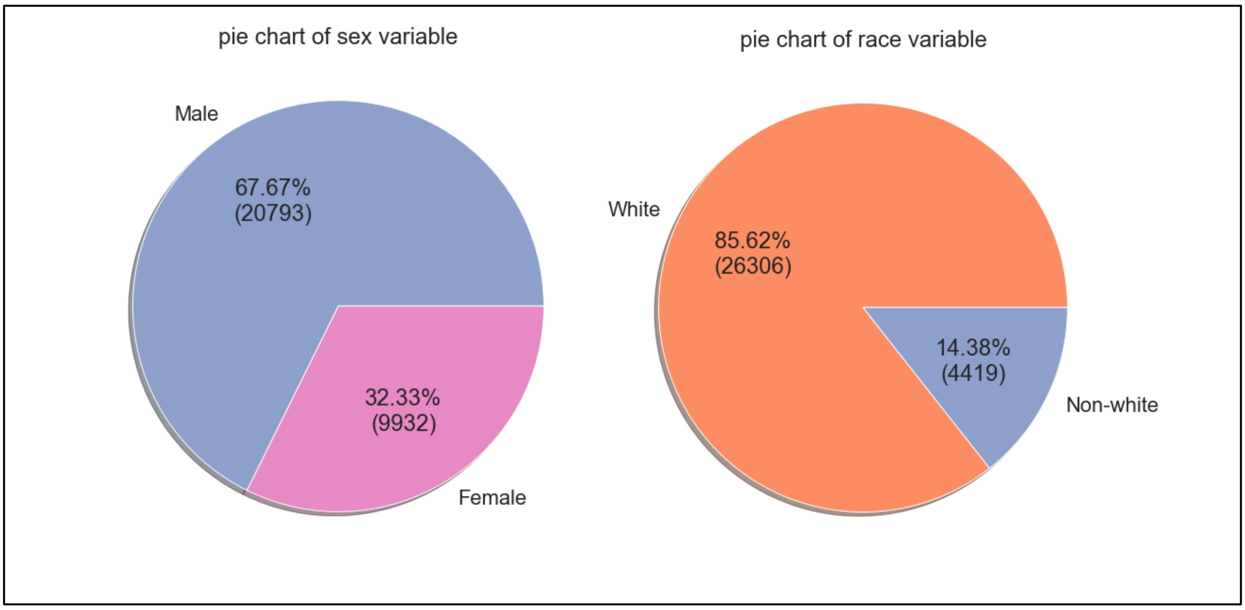

Figure 15

The pie chart of sex shows that approx. 68% of the dataset is male. The pie chart of race shows that roughly 86% of the dataset is white.

Hence here data is unbalanced in terms of race and gender. An issue that may rise with an unbalanced dataset is that the model parameters can become skewed towards the majority. A model will try to maximise accuracy across the entire population. In doing so it may unknowingly favour trends in the male or white population. Consequently, we may have a lower accuracy on the female population.

Prevalence:

For a target variable, prevalence is defined as the proportion of the positive cases to the overall cases. Here positive case is defined when the target variable has a value of 1.

When the whole dataset is considered, it has an overall prevalence of approx. 24.9%. That is, roughly 25% of the people in the dataset earn over 50K dollars.

Prevalence as a fairness metric:

In order to use prevalence as a fairness metric the prevalence is calculated for the different privileged and unprivileged groups.

Attribute	Privileged group	Unprivileged group
Race	26.4%	16%
Sex	31.4%	11.4%

Table 3

While 26.4 % of the people who are white earn above 50K in the dataset, only 16% of the people who are non- white earn above 50K in the dataset.

Approx 31% of males earn above 50K, where areas only 11% of the females earn above 50 K. In fact, based on the data, a male is nearly 3 times as likely to earn above 50K than a female.

Overall prevalence is much higher for the privileged groups. These large differences in prevalence may be due to the historical significance of the data collection. This dataset was built using the United States census data from the year 1994. In this sense, prevalence can be used to understand the extent to which historical injustice is embedded in the target.

Proxy Variables:

Proxy variables are variables that are used as substitutes or indicators for other attributes that are difficult to measure directly or not available in a particular dataset (Johnson et al., 2022).

In the context of fairness, proxy variables are features that are highly correlated or associated with the protected features of our dataset. Due to the associativity between proxy variables and the protected features, a model which uses proxy variable might as well be using a protected attribute to make decisions.

To find which variables are associated with the protected features like sex and gender, **mutual information**, which is a measure of non-linear association between two variables, has been used.

Mutual information measures the amount of information one can know from one variable by observing the values of the second variable.

Mutual information is closely related to the concept of entropy. The entropy of a variable is a measure of the information or the uncertainty of the variable's possible values.

It is defined as:

$$\mathrm{H}(X) = -\sum p(x)\log\big(p(x)\big)$$

Where $p(x)$ is the probability values of X.

Relative entropy measures the distance between two distributions. It is given by

$$\mathrm{D}(p|q) = \sum \mathrm{p}(x)\log\left(\frac{p(x)}{q(y)}\right)$$

Where p(x) and q(x) are two probability distributions.

Mutual Information is defined as the relative entropy between the joint distribution of the two variables and the product of their marginal distributions.

$$\mathrm{I}(X,Y) = \sum_{X}\sum_{\mathrm{Y}} \mathrm{p(x,y)log}\left(\frac{\mathrm{p(x,y)}}{\mathrm{p(x)p(y)}}\right)$$

Where $I(X,Y)$ is the Mutual Information between variables X and Y, the joint probability of the two variables is $p(x,y)$ and their marginal probabilities are $p(x)$ and $p(y)$.

The mutual information quantifies the amount of information that is shared between the potential proxy variable and the sensitive attribute. A higher mutual information value indicates a stronger association between the two variables.

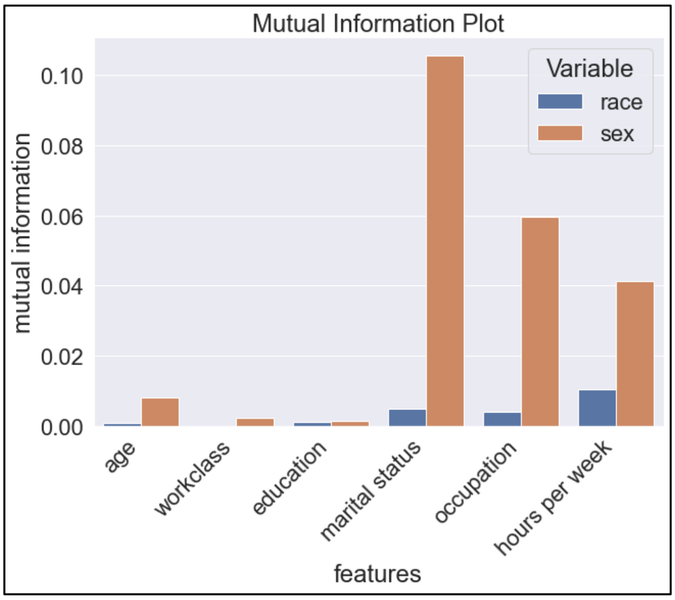

Figure 16

It can be observed that there is a high value between marital status and sex, followed by occupation and sex then hours per week and sex. This suggests a possible relationship between these variables, indicating that marital-status, occupation and hours.per.week could be a proxy variable for sex.

So far, an in-depth exploratory analysis has highlighted the issues that could lead to an unfair model.

Model building:

Before the model was built, the dataset had to undergo through some transformations.

Since the following machine learning models require numerical feature variables, the categorical variables needed to be converted into a numerical format before they could be used in training the models. The categorical variables were converted into different dummy variables of their categories by keeping one category as the base line. The base level categories chosen were: education- HS-level, marital.status- married, occupation- blue-collar.

Then the dataset was randomly divided into two parts- a training set and a test set in the ratio 7:3. That is, the training set had 70% of the datapoints while the test set had 30% of the data points. The models will be trained on the training set and evaluated on the test set.

6.3 Logistic Regression

[Code- c1]

Since the target variable is a binary nominal categorical variable, a binary logistic regression was chosen as the first classification model for prediction. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary target variable. The binary target variable denoting the income of a randomly selected individual annually was defined in the following way:

Y=1 if the income of that individual to be more than $50K per year.

=0 otherwise.

The final set of datapoints after removal of outliers is 29921. The number of features is 8, with only two variables as continuous and the rest categorical.

Let $p(X) = Pr(Y = 1|X)$ be the probability of success. To model $p(x)$, the logistic function is used,

$$\mathrm{p(X)} = \frac{\mathrm{e}^{\beta_0+\sum_{i=1}^{p}\beta_i X}}{1+\mathrm{e}^{\beta_0+\sum_{i=1}^{p}\beta_i X}}$$

Here p denotes the total number of features, p here being 11. After a bit of manipulation, the equation turns to:

$$\frac{p(X)}{1-p(x)} = e^{\beta_0+\sum_{i=1}^{p}\beta_i X}$$

Taking logarithm of both sides,

$$log\left(\frac{p(X)}{1-p(x)}\right) = \beta_0 + \sum_{i=1}^{p}\beta_i X \qquad (1)$$

The left-hand side is called the log odds or logit. It can be seen that the logistic regression model (equation 1) has a logit that is linear in variable X.

β_i is the regression coefficient and in logistic regression model one unit increment in X_i (the i^{th} feature) the log odds by β_i.

Significance testing:

β_i is the amount by which log odds of a positive response changes by one unit increment in x_i keeping the other predictors fixed.

We are to test:

H_{0i}: $\beta_i = 0$ vs H_{1i}: $\beta_i \neq 0$

We use the Wald's test statistic: $Z_i = \frac{\widehat{\beta_\imath}}{SE(\widehat{\beta_\imath})}$

Where $SE(\widehat{\beta_\imath})$ is the standard error of $\widehat{\beta_\imath}$. Standard Error is estimated by the software.

Since the number of datapoints is large, Z has an approximate standard normal distribution when H_0 is true.

Critical Region:

H_{0i} is rejected at α level of significance if and only if $|Z_i obs| > \tau_{\frac{\alpha}{2}}$

For convenience, the rejections will be based on p-values. That is, if the p-values are tinier than α then H_{0i} will be rejected. Table given in next page.

The following table gives the values of the Z statistic along with the p values for different predictors where the chosen level of significance is $\alpha = 0.05$

features	Coefficient estimate	Standard error	Z	p-value
Age	0.2221	0.019	11.690	0.000
Hours.per.week	0.2151	0.017	12.329	0.000
Workclass_other	-0.382	0.017	-2.305	0.021
Education_above	0.5338	0.023	23.593	0.000
Education_graduation	0.2637	0.017	15.088	0.000
Marital.status_not married	-0.6544	0.020	-32.663	0.000
Marital.status_other	-0.5640	0.018	-30.663	0.000
Occupation_service	0.0846	0.019	4.566	0.000
Occupation_white-collar	0.3657	0.020	18.186	0.000
Race_White	0.0462	0.016	2.871	0.004
Sex_Male	0.1731	0.019	9.216	0.000

Table 4

Interpretations:

Considering the level of significance at 5% interval, since the p-values of all the predictors are less than 0.05, it implies that they all are playing a significant role in predicting the income class.

a small p-value indicates that it is unlikely to observe such a substantial association between the predictor and the response due to chance, in the absence of any real association between the predictor and the response. Hence, if a small p-value is seen then it can be inferred that there is an association between the predictor and the response.

Both the continuous variables are playing a significant role in predicting the income of a person to be more than $50K. Since the regression coefficient estimates corresponding to age and hours.per.week are positive, it can be further said that as age increases, the individual is more prone to falling in the >50K income class. Similarly, as hours per week increases, the individual is more prone to falling in the >50K income class.

Workclass has a negative regression coefficient. Since private class was taken to be the base level, it can be interpreted that individuals working in private sectors have a higher chance of earning >50k dollars annually as compared to other sectors.

Education_above and education_graduation both have positive regression coefficients. Since HS-level was taken to be the base level, individuals who completed graduation are more prone to earning in >50K income bracket as compared to individuals with only school level education. Similarly individuals who pursued higher studies after graduation are susceptible to earn >50K than individuals who completed their education till HS-level.

Both the marital.status_not married and marital.status_others have a negative regression coefficient estimate. Since married was considered the base level, in the light of the given data , it can be said that married people have a higher chance of belonging to the >50K income class as compared to not married and others.

For occupation, blue- collar had been chosen as the base level. Since both the regression coefficient estimates are positive, it can be interpreted that individuals who have a white-collar job has higher chance of earning >50K as compared to individuals with blue-collared jobs. Similarly, individuals who are having jobs in a service also have a higher chance of falling in the higher income bracket than individuals with blue collared jobs.

The estimate of regression coefficient for race white is positive, implying white people are more inclined towards earning above 50K than people of colour.

The estimate of regression coefficient for sex male is positive, implying males are more inclined towards earning above 50K than females.

Accuracy Measures

Now that the model has been trained, there is a need to check how well the model fit in the training data, as well as how accurate the model will predict for future data points.

Confusion matrix on trained data.

$(\hat{Y})$ / (Y)	**1**	**0**	**Total**
1	12590	3134	15724
0	1297	3923	5220
Total	13887	7057	20944

Table 5

Training accuracy measures:

Defining TP= True Positive, FP= False Positive, TN= True Negative, FN= False Negative

Precision = $\frac{TP}{TP+FP}$ = 0.82

Out of all the individuals predicted to earn more than $50k, approx. 82% actually earn more than 50K.

Recall = $\frac{TP}{TP+FN}$ = 0.79

Out of all the individuals who actually earn more than $50k, approximately 79% were correctly identified by the model.

F1-Score = $\frac{2.Recall.Precision}{Recall+Precision}$ = 0.80

accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ = 0.79

Test Accuracy:

Confusion matrix on test data.

$(\hat{Y})$ (Y)	**1**	**0**	**Total**
1	5379	1357	6736
0	567	1674	2241
Total	5946	3031	8977

Table 6

test accuracy measures:

Defining TP= True Positive, FP= False Positive, TN= True Negative, FN= False Negative

Precision = $\frac{TP}{TP+FP}$ = 0.82

Out of all the individuals predicted to earn more than $50k, approx. 82% actually earn more than 50K.

Recall = $\frac{TP}{TP+FN}$ = 0.78

Out of all the individuals who actually earn more than $50k, approximately 78% were correctly identified by the model.

F1-Score = $\frac{2.Recall.Precision}{Recall+Precision}$ = 0.79

accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ = 0.78

The model correctly classified about 78% of the instances in the test dataset.

The test accuracy did not fall too much from the training accuracy, thus implying that there was no overfitting done by the model. In case overfitting had occurred, we would have seen a larger amount of training accuracy and a smaller amount of test accuracy.

Performance	**Precision %**	**Recall%**	**F1-Score %**	**Accuracy %**
Training	81.9	78.8	79.8	78.8
testing	81.7	78.6	79.5	78.6

Table 7

The logistic regression from the test accuracy performed more or less satisfactory in terms of prediction. Since the data is an imbalanced dataset, here F1-score is preferred than the accuracy metric.

6.4 Decision tree:

[Code- c2]

In Decision trees, the predictor space(the space of all possible predictor variables or features that are used in making predictions in a model) is segmented into a number of simple regions. The set of splitting rules that is used to segment the predictor space can be summarized in a form of a tree like structure, hence these types of models or approaches are called decision tree methods. Decision trees is applicable both regression and classification problems. Since the target variable in the dataset is a binary categorical variable, the primary object is to build a classification model. Hence, here a decision tree classifier will be constructed.

A decision tree is constructed using the following steps:

1. Let the set of predictors in the space be $X_1, X_2, \dots, X_p$. Then the predictor space is divided into J distinct and non-overlapping regions, $R_1, R_2, \dots, R_J$. The aim is to find the regions such that they minimise the *classification error rate*. The classification error rate refers to the fraction of training observations in that region that do not belong to the most frequently occurring class.

   Let $\widehat{p_{mk}}$ be the proportion of training observations in the mth region that belong to the kth class of the target variable. Here the choice of the error metric taken is Gini index, which is defined as

   $$G = \sum_{k=1}^{K} \widehat{p_{mk}}(1 - \widehat{p_{mk}})$$

   G gives a measure of the total variance across the K classes. Here K=2 since the target is binary in nature. Gini index takes a tiny value if all the proportions $\widehat{p_{mk}}$ are close to 0 or 1. Therefore, the Gini index is used to evaluate the of a particular split.

2. When an observation falls into the region R_j, a prediction is made, which in the case of a decision tree classifier is the mode of the response values (since the response is categorical). for the training observations in R_j.

Since it is computationally impractical to consider every possible combination of partition of the feature space, a greedy approach, also called top-down approach is used. The process of recursive binary splitting is followed, in which the splitting is performed from the top of the tree where all the observations belong to a single region. Then the predictor space is split successively, each split indicated via two new branches further down the tree.

For performing the recursive binary splitting, first a predictor X_i is selected and a cutpoint s is found such that it splits the space into the regions {X|Xi $<$ s} and $\{X|Xi \geq s\}$ which leads to the greatest possible reduction in error. This process is continued till no region has more than a certain number of observations as specified. Here {X|Xi $<$ s} implies implying the region of predictor space in which X_i takes a value less than s.

Once the regions R1,...,RJ have been created, we assign an observation in a given region to the most commonly occurring class of training observations in that region.

Hyperparameter tuning:

A decision tree has multiple hyperparameters. Hyperparameters are those parameters that are defined explicitly to control the learning process of a machine learning model during training in a dataset. They are used for optimizing the models, that is use a combination of hyperparameters such that it gives the lowest train error. Some of the hyperparameters that were considered to be most significant for tuning were :

Maximum depth: Controls the maximum depth of a tree, depth referring to the distance of the longest path from the root node to leaf node. Depth of a tree controls the complexity of a tree.

Minimum samples leaf: specifies the minimum number of samples required to be in a leaf node. Controls over-fitting by tuning the number of samples in a leaf.

Minimum samples split: Specifies the minimum number of samples necessary to split an internal node. Controls overfitting by tuning the minimum number of samples present in an internal node.

In order to find the best hyperparameter combination that will result in the best model, Grid Search hyperparameter tuning is performed. Grid Search exhaustively searched through a specified subset of hyperparameter values for a particular machine learning model. It performs K-fold cross validation for every combination of hyperparameters and chooses that particular combination that yields the highest validation score.

The following hyper-parameter values were given in gridsearch:

Max_depth: 2,3,5,10,20

Min_samples_leaf:5,10,20,50,100

Min_samples_split:4,6,8,10,20

Final decision tree built:

DecisionTreeClassifier with the following combinations: -

max_depth=20, min_samples_leaf=50, min_samples_split=4

Feature importance:

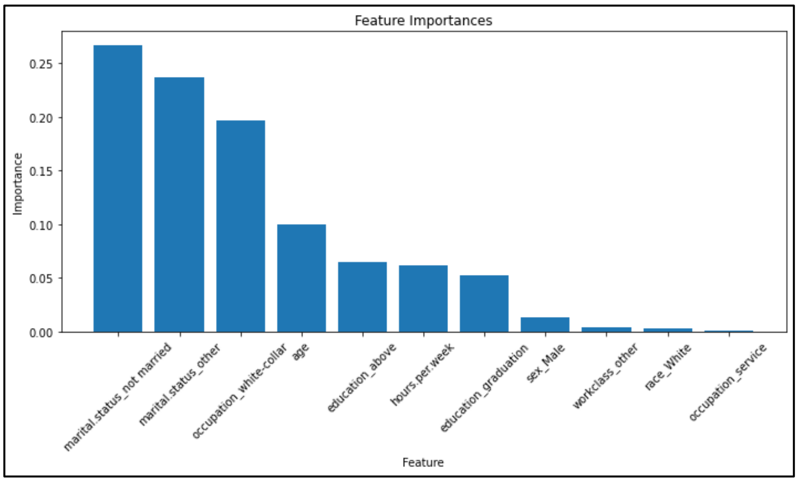

Figure 17

Feature importance assigns scores of the input predictors based on their importance in predicting the output.

Accuracy Measures

Now that the model has been trained, there is a need to check how well the model fit in the training data, as well as how accurate the model will predict for future data points.

Confusion matrix on trained data given next.

$(\hat{Y})$ (Y)	**1**	**0**	**Total**
1	14401	1323	15724
0	2126	3094	5220
Total	16527	4417	20944

Table 8

Training accuracy measures:

Defining TP= True Positive, FP= False Positive, TN= True Negative, FN= False Negative

Precision = $\frac{TP}{TP+FP}$ = 0.83

Out of all the individuals predicted to earn more than $50k, approx. 83% actually earn more than 50K.

Recall = $\frac{TP}{TP+FN}$ = 0.83

Out of all the individuals who actually earn more than $50k, approximately 83% were correctly identified by the model.

F1-Score = $\frac{2.Recall.Precision}{Recall+Precision} = 0.83$

accuracy = $\frac{TP+TN}{TP+TN+FP+FN} = 0.83$

Test Accuracy:

Confusion matrix on test data.

$(\hat{Y})$ (Y)	1	0	Total
1	6098	638	6736
0	1006	1235	2241
Total	7104	1873	8977

Table 9

test accuracy measures:

Defining TP= True Positive, FP= False Positive, TN= True Negative, FN= False Negative

Precision = $\frac{TP}{TP+FP}$ = 0.81

Out of all the individuals predicted to earn more than \$50k, approx. 81% actually earn more than 50K.

Recall = $\frac{TP}{TP+FN}$ = 0.82

Out of all the individuals who actually earn more than \$50k, approximately 82% were correctly identified by the model.

F1-Score = $\frac{2.Recall.Precision}{Recall+Precision}$ = 0.81

accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ = 0.82

The model correctly classified about 82% of the instances in the test dataset.

The test accuracy did not fall too much from the training accuracy, thus implying that there was no overfitting done by the model. In case overfitting had occurred, we would have seen a larger amount of training accuracy and a smaller amount of test accuracy.

Performance	Precision %	Recall%	F1-Score %	Accuracy %
Training	82.9	83.5	83.0	83.5
testing	80.9	81.7	81.1	81.7

Table 10

6.5 Random Forest:

[Code-c3]

Random forest addresses the shortcomings of the decision trees. In random forest, a number of decision trees are built on bootstrapped training samples. In bootstrap method, the training data is resampled again and again with replacement to create many simulated samples. The idea behind random forest is to generate many decision trees and produce a prediction from each tree and create a final predicted value by averaging all the predictions generated by the trees.

Each time a split in a tree is considered when building the decision trees, a random sample of m predictors is chosen as split nominees from the full set of predictors. Then finally all the predictions are averaged to obtain from the decision trees. For classification purpose for a given test observation, the class predicted by each of the B trees is recorded and taken a majority vote: the overall prediction is the most commonly occurring majority class among the predictions occurred in the decision trees.

The algorithm of random forest is as follows:

1. Let B number of decision trees be constructed using training set. Each tree is constructed by taking m predictors, where m<p p being the total number of predictors in the training data. Let the fitted models be referred to as $T_1, T_2, \dots T_B$.
2. Let there be a future value for which prediction has to be performed. The jth decision tree predict a value $\widehat{y_J}$.
3. For continuous outcomes, the $\widehat{y_J}$ values are averaged, for classification, the class predicted by each of the B trees is recorded and taken a majority vote.

The algorithm is forcibly restricted to not consider a majority of the available predictors at the same time. Often, many features can be informative for a target variable, but including all of them in the model may result in overfitting. Because if there is a very strong predictor in the data set along with a number of moderately strong predictors then most or all of the trees will use the strong predictor in the top split if all predictors are considered, thus making all the trees look similar. This reduces the correlation between the different decision trees, thereby improving the accuracy of the model and also bring it some stability.

A random forest also has multiple hyperparameters. Some of the hyperparameters that were considered to be most significant for tuning were :

- Maximum depth: Controls the maximum depth of each tree in the forest, depth referring to the distance of the longest path from the root node to leaf node. Depth of a tree controls the complexity of a tree.
- N estimators: Refers to the number of decision trees in the random forest. Here it refers to B.
- Maximum leaf nodes: Specifies the maximum number of leaf nodes that can be created in each tree of the forest.

The set of values provided for each hyperparameter during grid search hyperparameter tuning are:

n_estimators = [25, 50, 100]

max_depth = [3, 6, 9]

max_leaf_nodes = [3, 6, 9]

Best hyperparameter combination came out to be:

max_depth=9, max_leaf_nodes=9, n_estimators=25.

Accuracy Measures

Now that the model has been trained, there is a need to check how well the model fit in the training data, as well as how accurate the model will predict for future data points.

Confusion matrix on trained data given in table 11.

$(\hat{Y})$ (Y)	**1**	**0**	**Total**
1	14863	861	15724
0	2841	2379	5220
Total	17704	3240	20944

Table 11

Training accuracy measures:

Defining TP= True Positive, FP= False Positive, TN= True Negative, FN= False Negative

Precision = $\frac{TP}{TP+FP}$ = 0.81

Out of all the individuals predicted to earn more than $50k, approx. 81% actually earn more than 50K.

Recall = $\frac{TP}{TP+FN}$ = 0.82

Out of all the individuals who actually earn more than $50k, approximately 82% were correctly identified by the model.

F1-Score = $\frac{2.Recall.Precision}{Recall+Precision}$ = 0.80

accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ = 0.82

Test Accuracy:

Confusion matrix on test data.

$(\hat{Y})$ (Y)	**1**	**0**	**Total**
1	6347	389	6736
0	1247	994	2241
Total	7594	1383	8677

Table 12

test accuracy measures:

Defining TP= True Positive, FP= False Positive, TN= True Negative, FN= False Negative

Precision = $\frac{TP}{TP+FP}$ = 0.81

Out of all the individuals predicted to earn more than $50k, approx. 81% actually earn more than 50K.

Recall = $\frac{TP}{TP+FN}$ = 0.82

Out of all the individuals who actually earn more than $50k, approximately 82% were correctly identified by the model.

F1-Score = $\frac{2.Recall.Precision}{Recall+Precision} =$ 0.80

accuracy = $\frac{TP+TN}{TP+TN+FP+FN} =$ 0.82

The model correctly classified about 82% of the instances in the test dataset.

The test accuracy did not fall too much from the training accuracy, thus implying that there was no overfitting done by the model. In case overfitting had occurred, we would have seen a larger amount of training accuracy and a smaller amount of test accuracy.

Performance	**Precision %**	**Recall%**	**F1-Score %**	**Accuracy %**
Training	81.2	82.2	80.8	82.2
testing	80.4	81.6	80.1	81.6

Table 13

From the prediction accuracy measures, all the three machine learning models seemed to perform satisfactorily, with their accuracy values not differing significantly from each other. Out of all the three models, Decision tree seemed to perform the best with an f1-score of 81.1%. Even though Random forest is generally considered a better prediction model than the decision tree in terms of accuracy, here it produced slightly poorer results than the classification tree. One reason may be that random forest needs a high number of features for the algorithm to fare well. Since in this data, only 9 features were present, it failed to utilise its algorithm of decorrelating trees properly. Also, there was no over fitting present in any of the models. The train and test accuracy were very close to each other.

Now that the models are built and evaluated, there is one perspective based on which they have not been analysed- fairness. The accuracy measures of the previous section give a general understanding that the predictions performed by this model can be deemed satisfactory. However, whether these predictions can be deemed 'fair' is still left to be answered. For checking whether models are producing fair predictions, or if there is any hidden bias present in the model, a set of fairness metrices has been utilised in the next section.

6.6 Assessing models in terms of fairness.

The fairness of the above three machine learning models will be measured by applying various definitions of fairness and most of the definitions involve splitting the population into privileged and unprivileged groups.

Typically, a model's predictions lead to either a benefit or no benefit for an individual. For the 3 models trained on the adult data, it is assumed that Y=1 will lead to a benefit. That is if a model predicts that the person makes above $50K they will benefit in some way.

Fairness as a concept is very hard to define since it is multifaceted and complex. There are over 20 mathematical definitions of fairness, and different definitions can produce completely different outcomes (Arvind Narayanan, 2018). Based on the following three fairness metrices, fairness may be defined.

Accuracy:

Accuracy falls under the equal prediction measures of fairness. Since the data can be divided into different groups, the impacts of decisions are also contained within the groups. Hence a common notion is that fairness might ask these impacts of decisions to be equal (Mitchell et al., 2021).

Accuracy= $\frac{TrueNegative+TruePositive}{Total}$

Let $\hat{Y}$ be the predicted decisions. Then the fairness definition of equal accuracy can be states as:

$$P\left[\hat{Y} = Y \middle| A = a\right] = P\left[\hat{Y} = Y \middle| A = a'\right]$$

Where A is the sensitive attribute and a and a' are the two demographic groups.

This definition is based on the idea that fairness is symbolised by the predictions being correct at equal rates among the groups. Hence here the accuracy of the model in the two subset of privileged and unprivileged groups are compared.

Attribute	Models	Unprivileged group	Privileged group	Ratio
Sex	Logistic Regression	87.9%	74.1%	1.19
	Decision Tree	90.3%	77.5%	1.16
	Random Forest	89.2%	77.9%	1.14
Race	Logistic Regression	85.6%	77.4%	1.1
	Decision Tree	86.1%	80.1%	1.65
	Random Forest	86.8%	80.7%	1.07

Table 14

The ratio column gives the accuracy of the unprivileged group to the privileged group.

For both protected attributes it can be seen that the accuracy is actually higher for the unprivileged groups. This may lead us to believe that the model is benefiting the unprivileged group. However, the issue is that the accuracy can hide the consequences of a model prediction.

Why Accuracy may not be an ideal measure:

This data is suffering from the problem of imbalanced class of targets. Hence the majority class will overshadow the minority class in a model. Accuracy is aggregating the true positives and the true negatives, however the true positives are much higher than the true negatives. Moreover, the different demographic distribution in the two classes is not same, hence accuracy will not be able to capture the true bias present in the model. Thus, there is a need for a better Fairness metric.

Equal opportunity (or true positive rate):

Instead of using accuracy, a measure that can capture the benefits of a model is Equal Opportunity.

TPR= $\frac{TruePositive}{TruePositive+FalseNegative}$

TPR or True Positive Rate is the percentage of actual positives that were correctly predicted as positive. Since it has been assumed that a positive prediction will lead to some benefit. Hence this means the **denominator can be seen as the number of people who should benefit from the model.** The numerator is the **number of individuals who should benefit and have benefited**.

Hence TPR can be interpreted as the percentage of **people who have rightfully benefitted from the model.**

We define fairness using Equal opportunity ratio: Under equal opportunity we consider a model to be fair if the TPRs of the privileged and unprivileged groups are equal.

$$TPR_0 = P\left(\hat{Y} = 1 \middle| Y = 1, A = unprivilaged\right\}$$

$$TPR_1 = P\left(\hat{Y} = 1 \middle| Y = 1, A = privilaged\right\}$$

If the model is fair, then $TPR_0 = TPR_1$

Thus, Equal Opportunity Difference is defined as $\mathrm{EOD} = \mathrm{TP}R_0 - \mathrm{TPR}_1$

In practical usage, some threshold or cutoff must be defined based on which one can infer whether the model is fair or not.

Attribute	Models	Unprivileged group	Privileged group	EOD
Sex	Logistic Regression	58.9%	77.6%	-0.19
	Decision Tree	43.3%	57.26%	-0.14
	Random Forest	12.1%	51.6%	-0.39
Race	Logistic Regression	65.1%	75.6%	-0.11
	Decision Tree	37.9%	56.7%	-0.19
	Random Forest	35.8%	46.4%	-0.10

Table 15

From the equal opportunity differences, it can be seen none of the values are 0. In fact, all the EODs are negative. Negative EOD implies $TPR_0 < TPR_1$. Thus, the true positive rate of unprivileged groups is less than the true positive rate of the privileged groups. This implies that more individuals belonging to the privileged groups have rightfully been benefited from the model than the individuals belonging to the unprivileged groups.

Disparate Impact:

Disparate Impact metric has been adapted from the US legal doctrine of *disparate impact.* Today, it is the most dominant legal theory for determining *unintended* discrimination in the United States (Feldman et al., 2015).

Disparate impact metric is computed as a ratio of the rate of favourable outcomes for the unprivileged group to that of the privileged groups. This is given by the ratio between estimated probability of unprivileged group getting a favourable prediction and the estimated probability of privileged group getting favourable prediction.

$$\text{DI} = \frac{P(\hat{Y} = 1 \middle| A = unprivilaged)}{P(\hat{Y} = 1 \middle| A = privilaged)}$$

Defining the Predicted as Positive Percentage $PPP = \frac{TruePositive+FalsePositie}{Total}$

This is the percentage of people who have either been correctly (TP) or incorrectly (FP) predicted as positive. It can be interpreted as the percentage of **people who will benefit from the model**. For the above models, it is the percentage of individuals who are predicted to have a high income.

Attribute	**Models**	**PPP** **Unprivileged group**	**PPP** **Privileged group**	**DI**
Sex	Logistic Regression	14.2%	43.1%	0.33
	Decision Tree	80.8%	27.0%	0.299
	Random Forest	17.4%	23.1%	0.07
Race	Logistic Regression	19.2%	36.1%	0.53
	Decision Tree	10.1%	22.6%	.447
	Random Forest	8.7%	17.3%	0.5

Table 16

The selection of a cutoff depends on domain knowledge and industry expertise.

Since this definition is supposed to represent the legal concept of disparate impact, for the adult census data, the 80% rule is considered. Since the dataset is based on the US census data, 80% rule is a legal principle used in US to assess potential discrimination against protected groups.

That is if $DI < 0.8$ then we may consider that the **model is being unfair**.

That is the PPP for the unprivileged group must not be less than 80% of that of the privileged group for the model to be deemed legally fair.

6.7 Bias mitigation

Based on the various fairness metrices and definitions, it was concluded that all the three models are definitely unfair, especially in terms of the sensitive attribute Sex. Even though it was noticed that the models were to some extent unfair in terms of racial demographic groups, however based on the degree of unfairness, sex seemed to influence the models more than race. This was also implied from the exploratory fairness analysis where through mutual information it had been observed that most of the variables had some sort of associativity with sex rather than race. In fact, Marital status, occupation and hours.per.week had been considered a proxy variable due to its strong association with the variable sex.

Even though the three machine learning models produced satisfactory performance results in terms of predictions, the models tended to favour the privileged groups more than the unprivileged groups. This was inferred from the various fairness metrices like Disparate Impact and Equal Opportunity differences.

Bias mitigation algorithms help in rectifying these biases formed in the models. Bias mitigation algorithms help ensure model fairness in decision making procedures. Essentially, they aim to minimize the impact of the biased influences such as race or gender, age, etc on the outcomes of the models thereby mitigating discrimination of demographic groups. Bias mitigation techniques can be broadly classified into three approaches: preprocessing, in-processing and post-processing.

Preprocessing Algorithms:

Preprocessing algorithms play a significant role in addressing biases within datasets by recognizing biases which extend beyond the classifier itself. Let the bias of a dataset be D concerning a group $A=a$. Then the bias can be characterized as an approximation to the difference in probabilities of belonging to the positive class of target based on whether the individual has a protected characteristic equal to a, or different from a. Some of the preprocessing algorithms are Reweighing and Disparate Impact Remover.

In-processing Algorithm:

In-processing algorithms modify the Machine Learning model to mitigate the bias in the original model prediction. Basically, instead of transforming data, here the Machine Learning algorithms are adjusted. Some types of in-processing bias mitigation algorithms are Adversarial Debiasing and Prejudice Remover Regularizer.

Post-processing Algorithms:

This set of approaches modify the prediction results instead of adjusting the Machine Learning models or transforming the dataset. These methods work by changing the predictions produced by a model. Some examples include Equalized Odds and Reject Option Classification.

Mitigating bias found in the Adult dataset:

Previously the bias exhibited by the machine learning models were analysed with different fairness measures in the adult dataset. Based on the metric measures, the conclusion that the models are *unfair* had been reached.

The major problem these models will face is even though their accuracies are satisfactory, they will fail to make *fair* decisions in the future. Hence, there is a strong need of strategies that can mitigate the biases found in the model.

There are two sensitive attributes present in the data, and the models came out to be unfair based on both the attributes- sex and Race. Ideally, one would like to perform bias mitigation algorithms by taking both sex and Race together as the sensitive attributes. However, in practice dealing with multiple sensitive attributes becomes very complicated. This is due to the following reasons:

- **Intersectionality**: Individuals often belong to multiple demographic groups simultaneously and each group may be subjected to different biases. In this case, a white person can be a male or a female.

Similarly, a female may be black or white. Bias mitigation in such cases requires a thorough understanding of the challenges that arise from intersection of multiple identities.

- **Data Complexity**: When there are multiple sensitive attributes, the model will face the problem of interaction effects and correlations between different sensitive features. These effects also need to be included while performing bias mitigation strategies.
- **Algorithmic Complexity**: Bias mitigation algorithms become more complex when dealing with multiple sensitive attributes. Moreover, AI fairness being a relatively newer field, much research needs to be performed to come up with algorithms that can handle such cases.

Hence for this data, only one sensitive attribute has been chosen- the variable sex and moved forward with the bias mitigation procedures. On comparing the Disparate Impact values of all the models for sex and race, it can be seen that the Disparate Impact ratio is much lesser for sex than for Race. This implies that the model is more unfair when divided by sex than when divided by Race. Also, when the Exploratory Fairness analysis was performed, from the mutual information scores we could observe that sex influenced the other variables more than race. This persuaded us to consider sex as the more important variable for bias mitigation.

Of the three general approaches of Bias mitigation, it was decided to consider pre-processing techniques only for treating unfairness. The reason is in-processing and post-processing techniques are complicated and difficult to perform. Moreover, we received satisfactory results from the two techniques that were performed on the dataset. The algorithms used are - Disparate Impact Remover and Reweighing.

Bias mitigation Results

Disparate Impact Remover (DIR):

[Code-c4]

This algorithm is based on the concept of the metric DI that measures the fraction of individuals achieving positive outcome from an unprivileged group in comparison to the privileged group (Feldman et al., 2015). To remove the bias, this technique modifies the value of protected attribute to remove distinguishing factors. DIR works by tweaking the features so that the distributions for the two groups become similar. DIR maintains the orders of the ranks within the sub- populations. Only the distributions are modified so that they resemble each other more closely. It idea if for the model to not distinguish the features in terms of the sensitive attributes.

Models	Metrices	Before DIR	After DIR
Logistic Regression	Accuracy	78.8%	78.4%
	F1 score	79.8%	79.3%
	Disparate Impact	0.33	0.38
	Equal Opportunity Difference	-0.19	-0.20
Decision Tree	Accuracy	81.6%	81.5%
	F1 Score	81.1%	80.7%
	Disparate Impact	0.29	0.48
	Equal Opportunity Difference	-0.14	-0.13
Random Forest	Accuracy	81.6%	81.4%
	F1 Score	80.1%	79.8%
	Disparate Impact	0.07	0.21
	Equal Opportunity Difference	-0.39	-0.12

Table 17

Reweighing:

[Code- c5]

The technique of Reweighing generates different weights for data points in each (group, target label) combination (Kamiran & Calders, 2009). In a biased dataset, different weights are assigned to reduce the effect of favouritism of a specific group. If an input class has been favoured, then a lower weight has been assigned in comparison to the class not been favoured.

Let's assume a set of attributes as $A = \{A_1, A_2, \dots A_n\}$ and the target variable be denoted by Class, where Class is a binary variable with two labels = {-,+}

Let S be the sensitive attribute and $S \in A$ and b be a value such that $b \in dom(S)$ called the deprived community (unprivileged group). Here S is also a binary attribute with categories $\{b, w\}$. This algorithm attaches different weights to the objects.

There are four combinations found from S and class. They are $(S = b, Class = +)$, $(S = b, Class = -)$, $(S = w, Class = +)$,$(S = w, Class = -)$. The weights are assigned such that objects with $(S = w, Class = +)$ receive higher weights than objects with $(S = b, Class = -)$ and objects with $(S = b, Class = +)$, will receive lower weights than objects with $(S = w, Class = -)$. Hence, to compensate for the bias, the algorithm will attach lower weights to objects that have been favoured.

Let the weight of object X be W(X). Then,

$$\mathrm{W(X)} = \frac{\mathrm{P_{exp}}\big(\mathrm{S = X(S) \wedge C\,l\,a\,s\,s = X(Class)}\big)}{\mathrm{P_{obs}}\big(\mathrm{S = X(S) \wedge C\,l\,a\,s\,s = X(Class)}\big)}$$

For this dataset, the proportions are given by:

$P_{exp}(S = female \wedge Class \Rightarrow 50K)$ =0.32*0.24=0.08

$P_{obs}(S = female \wedge Class \Rightarrow 50K)$ = 0.037

Then $W(X)$ = 2.16

Calculating the other weights similarly:

Combinations (X)	**W(X)**
Female,>50K	2.18
Female,<=50K	0.84
Male,>50K	0.79
Male,<=50K	1.09

Table 18

Results:

Finally, the entire model training process is performed again this time with the weights included in the models.

Models	**Metrices**	**Before Reweighing**	**After Reweighing**
Logistic Regression	Accuracy	78.8%	78.6%
	F1 score	79.8%	79.5%
	Disparate Impact	0.33	0.54
	Equal Opportunity Difference	-0.19	-0.34
Decision Tree	Accuracy	81.6%	81.3%
	F1 Score	81.1%	80.3%
	Disparate Impact	0.29	0.66
	Equal Opportunity Difference	-0.14	0.05
Random Forest	Accuracy	81.6%	81.4%
	F1 Score	80.1%	79.8%
	Disparate Impact	0.07	0.34
	Equal Opportunity Difference	-0.39	-0.1

Table 19

Clearly the reweighing technique performed better than the Disparate Impact Remover. The DIR had managed to mitigate bias only by a small amount. Where areas reweighing managed to mitigate bias to a satisfactory level, specially in Decision tree. Even though before bias mitigation, the decision tree had the highest amount of bias present as observed from the Fairness metrices. Hence, reweighing managed to bring its disparate impact value to an approx. 70%. Even though it falls short of the 80% rule to be classified a 'fair' model, it did improve its performance drastically in terms of fairness. Previously its disparate impact value was only 0.3 or 30% and it managed to touch 70%, which shows that considerable amount of disparity between the groups were eliminated.

Comparing the three machine learning models, Decision tree had performed best in terms of accuracy. Where areas logistic regression had fared well compared to the other models in terms of fairness, even though it had less accuracy. Random forest even if it managed to have a good accuracy score failed in terms of fairness as it's DI value was significantly lower than the other two models.

Finally, the Machine Learning model that performed better after bias mitigation at a cost of very slight dip in accuracy is the **Decision Tree classifier**.

7 CONCLUSION

The objective of this study was to examine the classification models in terms of both accuracy and fairness, and analyse the potential sources of bias.

The three models- Logistic Regression, Decision tree and Random Forest were compared based on the accuracy and fairness, and out of the three models, the decision tree was concluded to have the best accuracy and fairness score out of the three models after bias mitigation. A point of thought that came while studying the significant features in the decision tree, was that even though sex and race were the sensitive variables in the data, their importance in the models was far lesser as compared to other features. From the feature importance plot sex and race seemed to have very low importance. In fact, race had almost no importance in the decision tree model. The feature which actually came out to significant was marital status. This understanding shed light on why exploratory fairness analysis is important. In the pre model building stage it had already been identified that marital status is a proxy variable. Many data scientists assume that when building machine learning models, removing the protected variables is going to help in avoiding unfair bias. However, due to many features being closely associated with the protected attributes will still introduce bias in the models. This happened when Amazon had started its Prime Free Same Day delivery service. After analysis by experts, it was found out that some major cities had excluded zip codes of predominant black population. This happened because Amazon focused on the concentration of prime users, and instead had excluded race as a feature. So even after looking at just the numbers rather than demographics did not prevent the system from infusing inequality for access of retail services (*Amazon Doesn't Consider the Race of Its Customers. Should It?*, n.d.).

Another point of caution that was raised in the fairness literature is 'user to data' (Mehrabi et al., 2022). Any inherent biases present inside users might be reflected in the data they are producing. Even in the adult data set such patterns were hinted. An important bias associated with it is called the Historical Bias- which is the already existing bias, and the social and technical related issues present in the world can creep into the data from the data generation process, even if there has been a proper sampling and variable selection. Even in adult dataset, if without bias mitigation the models were used for predicting future decisions, then there will be a high chance that females would be less likely to be classified having an income greater than $50K dollars, thus making the models biased towards males. Even though the model decision would be reflecting the reality, but whether they should reflect such reality in present times is a matter of debate. Because the adult data set was based on the 1994 census data, and the reality reflected around that time may not be the reality of the present time.

On a final note, in today's world where social causes put more weightage in the present generation, fairness has become a pressing issue. Hence, technology should also be adopting algorithmic fairness, as the world moves towards prioritizing fair representation for each and every individual.

8 ACKNOWLEDGEMENT

I would like to express my most sincere gratitude to my supervisor Dr. Mausumi Bose for her continuous support of my dissertation work and research. I will remain grateful for her patience, motivation, and immense knowledge. Her expertise and guidance helped me in every step of my research.

I would like to thank all my department professors for sharing with me their knowledge on different topics specially Machine Learning. Without their engaging lessons I would not have been able to grasp such clarity of the concepts. I would like to thank my college- St. Xavier's College (Autonomous), Kolkata for giving me the opportunity to work on my dissertation.

Lastly, I thank my friends and family for their continued moral support and unwavering motivation in every step of the way.

9 REFERENCE

Amazon Doesn't Consider the Race of Its Customers. Should It? (n.d.). Bloomberg.Com. Retrieved April 2, 2024, from http://www.bloomberg.com/graphics/2016-amazon-same-day/

Arvind Narayanan (Director). (2018, March 1). *Tutorial: 21 fairness definitions and their politics*. https://www.youtube.com/watch?v=jIXIuYdnyyk

Cleary, T. A. (1966). Test Bias: Validity of the Scholastic Aptitude Test for Negro and White Students in Integrated Colleges. *ETS Research Bulletin Series*, *1966*(2), i–23. https://doi.org/10.1002/j.2333-8504.1966.tb00529.x

Feldman, M., Friedler, S., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). *Certifying and removing disparate impact* (arXiv:1412.3756). arXiv. http://arxiv.org/abs/1412.3756

Johnson, K. D., Foster, D. P., & Stine, R. A. (2022). Impartial Predictive Modeling and the Use of Proxy Variables. In M. Smits (Ed.), *Information for a Better World: Shaping the Global Future* (pp. 292–308). Springer International Publishing. https://doi.org/10.1007/978-3-030-96957-8_26

Kamiran, F., & Calders, T. (2009). *Classifying without discriminating*. 1–6. https://doi.org/10.1109/IC4.2009.4909197

Machine Bias—ProPublica. (n.d.). Retrieved April 2, 2024, from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2022). *A Survey on Bias and Fairness in Machine Learning* (arXiv:1908.09635). arXiv. http://arxiv.org/abs/1908.09635

Mitchell, S., Potash, E., Barocas, S., D'Amour, A., & Lum, K. (2021). Algorithmic Fairness: Choices, Assumptions, and Definitions. *Annual Review of Statistics and Its Application*, *8*(Volume 8, 2021), 141–163. https://doi.org/10.1146/annurev-statistics-042720-125902

Pagano, T. P., Loureiro, R. B., Lisboa, F. V. N., Cruz, G. O. R., Peixoto, R. M., Guimarães, G. A. de S., Santos, L. L. dos, Araujo, M. M., Cruz, M., de Oliveira, E. L. S., Winkler, I., & Nascimento, E. G. S. (2022). *Bias and unfairness in machine learning models: A systematic literature review* (arXiv:2202.08176). arXiv. https://doi.org/10.48550/arXiv.2202.08176

Trisha,Kush,Michael, M., Varshney,Hind. (2020). *AI Fairness- How to Measure and Reduce Unwanted Bias in Machine Learning*. O'Reilly.

10 APPENDIX

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler,OneHotEncoder,LabelEncoder, MinMaxScaler
import statsmodels.api as sm
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_recall_fscore_support
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# CALLING DATA
data=pd.read_csv("D:/Documents/dissertation_msc/adult_clean.csv")
data=data[data['age']<=73.5]
# DIVIDING DATA
X=data.drop(columns=['income'])
y=data['income']
 #SEPARATING DATA

label_encoder=LabelEncoder()

y=label_encoder.fit_transform(y)

X=pd.get_dummies(X,columns=None,drop_first=True)
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=1)

# STANDARDISING DATA
scaler=StandardScaler()
X_train_sc=scaler.fit_transform(X_train)
X_test_sc=scaler.transform(X_test)


X_scaled_df = pd.DataFrame(X_train_sc, columns=X_train.columns)

# LOGISTIC MODEL [c1] ________________________________________

log_model2=sm.Logit(y_train,X_scaled_df)
```

```
result=log_model2.fit()
result.summary()

ths=np.linspace(0,1,100)
metrics=[]

for th in ths:
    y_pred=(result.predict(X_test_sc) >= th).astype(int)
    f1=f1_score(y_test,y_pred)
    metrics.append((th,f1))

met_df=pd.DataFrame(metrics,columns=['threshold','f1-score'])

th_f=met_df.loc[met_df['f1-score'].idxmax()]

y_h_log=(result.predict(X_test_sc) >= th_f['threshold']).astype(int)
y_h_log_train=(result.predict(X_train_sc) >= th_f['threshold']).astype(int)

def metrics_fun(y_test,y_hat):
    precision, recall, f1_score, _ = precision_recall_fscore_support(y_test, y_hat, average='weighted')
    accuracy=accuracy_score(y_test, y_hat)

    # Print the results
    print("Precision (Weighted):", precision)
    print("Recall (Weighted):", recall)
    print("F1-Score (Weighted):", f1_score)
    print("accuracy (Weighted):", accuracy)

metrics_fun(y_test,y_h_log)

# DECISION TREE [c2] ____________________________________________
params={
        'max_depth':[2,3,5,10,20],
        'min_samples_leaf':[5,10,20,50,100],
        'min_samples_split':[4,6,8,10,20],
        'criterion':['gini','entropy']}
dt1=DecisionTreeClassifier(random_state=1)

clf=GridSearchCV( estimator=dt1,
                 param_grid=params,
                 cv=4,n_jobs=-1,verbose=1,
                 scoring='accuracy')
clf.fit(X_train_sc, y_train)

dt_best=clf.best_estimator_
y_h_dt = dt_best.predict(X_test_sc)
metrics_fun(y_test,y_h_dt)
```

```
# RANDOM FOREST [c3]______________________________________________

param_grid = {
    'n_estimators': [25, 50, 100],
    'max_depth': [3, 6, 9],
    'max_leaf_nodes': [3, 6, 9],
}


grid_search = GridSearchCV(RandomForestClassifier(),
                           param_grid=param_grid)
grid_search.fit(X_train_sc, y_train)
rf_best=grid_search.best_estimator_
print(grid_search.best_estimator_)

y_h_rf= rf_best.predict(X_test_sc)

metrics_fun(y_test,y_h_rf)

#REWEIGHING [c5]______________________________

from aif360.metrics import BinaryLabelDatasetMetric, ClassificationMetric
from aif360.datasets import BinaryLabelDataset
from aif360.algorithms.preprocessing import DisparateImpactRemover
from aif360.algorithms.preprocessing.reweighing import Reweighing

def aif_data(X,y):
    encoded_df=X.copy()

    encoded_df['income']=y

    BL = BinaryLabelDataset(
        favorable_label=1,
        unfavorable_label=0,
        df=encoded_df,
        label_names=['income'],
        protected_attribute_names=['sex_Male'])
    return(BL)
BL=aif_data(X,y)
def aif_met(BL):
    privileged_groups = [{'sex_Male': 1}]
    unprivileged_groups = [{'sex_Male': 0}]
    metric_transf_train = BinaryLabelDatasetMetric(BL,
                                             unprivileged_groups=unprivileged_groups,
                                             privileged_groups=privileged_groups)
    return([metric_transf_train.disparate_impact(),metric_transf_train.mean_difference()])
def aif_cl(BL,BL_p):
    classified_metric = ClassificationMetric(BL, BL_p,
                                            unprivileged_groups=unprivileged_groups,
                                            privileged_groups=privileged_groups)

    return([classified_metric.equal_opportunity_difference()])
privileged_groups = [{'sex_Male': 1}]
unprivileged_groups = [{'sex_Male': 0}]
RW = Reweighing(unprivileged_groups=unprivileged_groups,
               privileged_groups=privileged_groups)

RW.fit(BL)
dataset_transf_train = RW.transform(BL)
transformed = dataset_transf_train.convert_to_dataframe()[0]
weights=dataset_transf_train.instance_weights
np.unique(weights)
X_train,X_test,y_train,y_test,weights_train,weights_test=train_test_split(X,y,weights,test_size=0.3,rand
om_state=1)

# STANDARDISING DATA
scaler=StandardScaler()
X_train_sc=scaler.fit_transform(X_train)
X_test_sc=scaler.transform(X_test)

#LOGISTIC MODEL
model = LogisticRegression()
# Fit the model to the training data
model.fit(X_train_sc, y_train,sample_weight=weights_train)

y_h_log_rw=(model.predict(X_test_sc)).astype(int)
#y_h_log_train=(result.predict(X_train_sc) >= th_f['threshold']).astype(int)

metrics_fun(y_test,y_h_log_rw)

#original test data
aif_met(aif_data(X_test,y_test))

# for predicted but with transform
aif_met(aif_data(X_test,y_h_log_rw))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_log_rw))

# for predicted but wihout transform
aif_met(aif_data(X_test,y_h_log))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_log))

# DECISION TREE
```

```
params={
        'max_depth':[2,3,5,10,20],
        'min_samples_leaf':[5,10,20,50,100],
        'min_samples_split':[4,6,8,10,20],
        'criterion':['gini','entropy']
        }

dt1=DecisionTreeClassifier(random_state=1)

clf=GridSearchCV( estimator=dt1,
                param_grid=params,
                cv=4,n_jobs=-1,verbose=1,
                scoring='accuracy')

clf.fit(X_train_sc, y_train,sample_weight=weights_train)

dt_best=clf.best_estimator_
y_h_dt_rw = dt_best.predict(X_test_sc)
metrics_fun(y_test,y_h_dt)

metrics_fun(y_test,y_h_dt_rw)



# for predicted but with transform
aif_met(aif_data(X_test,y_h_dt_rw))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_dt_rw))

# for predicted but wihout transform
aif_met(aif_data(X_test,y_h_dt))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_dt))

#RANDOM FOREST---------------------

param_grid = {
    'n_estimators': [25, 50, 100],
    'max_depth': [3, 6, 9],
    'max_leaf_nodes': [3, 6, 9],
}

grid_search = GridSearchCV(RandomForestClassifier(),
                          param_grid=param_grid)

grid_search.fit(X_train_sc, y_train,sample_weight=weights_train)
rf_best=grid_search.best_estimator_
print(grid_search.best_estimator_)
```

```
y_h_rf_rw= rf_best.predict(X_test_sc)

metrics_fun(y_test,y_h_rf_rw)

# for predicted but with transform
aif_met(aif_data(X_test,y_h_rf_rw))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_rf_rw))

# for predicted but wihout transform
aif_met(aif_data(X_test,y_h_rf))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_rf))

#DISPARATE IMPACT REMOVER [c4]______________________________
BL=aif_data(X,y)
di = DisparateImpactRemover(repair_level = 1.0)
dataset_transf_train = di.fit_transform(BL)
transformed = dataset_transf_train.convert_to_dataframe()[0]


x_DI = transformed.drop(['income'], axis = 1)
y_DI = transformed['income']

X_train_DI,X_test_DI,y_train_DI,y_test_DI = train_test_split(x_DI, y_DI, test_size=0.3, random_state = 1)

scaler = StandardScaler()
X_train_DI_sc = scaler.fit_transform(X_train_DI)

X_test_DI_sc = scaler.transform(X_test_DI)

model_DI = LogisticRegression()
# Fit the model to the training data
model_DI.fit(X_train_DI_sc, y_train_DI)

y_h_log_DI=(model.predict(X_test_DI_sc)).astype(int)

BL_test_log=aif_data(X_test,y_h_log_DI)
aif_met(BL_test_log)


aif_met(aif_data(X_test_DI,y_test_DI))

metrics_fun(y_test_DI,y_h_log_DI)
#______________________________________________________________


#LOGISTIC MODEL
```

```
model = LogisticRegression()
# Fit the model to the training data
model.fit(X_train_sc, y_train)

y_h_log_rw=(model.predict(X_test_sc)).astype(int)
#y_h_log_train=(result.predict(X_train_sc) >= th_f['threshold']).astype(int)

metrics_fun(y_test,y_h_log_rw)

#original test data
aif_met(aif_data(X_test,y_test))

# for predicted but with transform
aif_met(aif_data(X_test,y_h_log_rw))s
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_log_rw))

# for predicted but wihout transform
aif_met(aif_data(X_test,y_h_log))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_log))

# DECISION TREE
params={
        'max_depth':[2,3,5,10,20],
        'min_samples_leaf':[5,10,20,50,100],
        'min_samples_split':[4,6,8,10,20],
        'criterion':['gini','entropy']
        }

dt1=DecisionTreeClassifier(random_state=1)

clf=GridSearchCV( estimator=dt1,
                  param_grid=params,
                  cv=4,n_jobs=-1,verbose=1,
                  scoring='accuracy')

clf.fit(X_train_sc, y_train,sample_weight=weights_train)

dt_best=clf.best_estimator_
y_h_dt_rw = dt_best.predict(X_test_sc)
metrics_fun(y_test,y_h_dt)

metrics_fun(y_test,y_h_dt_rw)


# for predicted but with transform
aif_met(aif_data(X_test,y_h_dt_rw))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_dt_rw))
```

```
# for predicted but wihout transform
aif_met(aif_data(X_test,y_h_dt))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_dt))


#RANDOM FOREST---------------------

param_grid = {
    'n_estimators': [25, 50, 100],
    'max_depth': [3, 6, 9],
    'max_leaf_nodes': [3, 6, 9],
}
grid_search = GridSearchCV(RandomForestClassifier(),
                           param_grid=param_grid)
grid_search.fit(X_train_sc, y_train,sample_weight=weights_train)
rf_best=grid_search.best_estimator_
print(grid_search.best_estimator_)

y_h_rf_rw= rf_best.predict(X_test_sc)

metrics_fun(y_test,y_h_rf_rw)

# for predicted but with transform
aif_met(aif_data(X_test,y_h_rf_rw))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_rf_rw))

# for predicted but wihout transform
aif_met(aif_data(X_test,y_h_rf))
aif_cl(aif_data(X_test,y_test),aif_data(X_test,y_h_rf))
```