REPORT ON BIAS-CHECKER TOOL

Done and Reported by T. Shivani

Blekinge Institute of Technology

shivanithummanapally@gmail.com

INTRODUCTION

In today's era, Data is ruling the world paving its path to Big Data, Data Science, Deep Learning, Machine Learning models and Artificial Intelligence. In the Artificial Intelligence and Machine Learning powered world where predictive models have become the ultimate decision makers in the industries such as banking, insurance, and employment for their respective purposes.

But the major concerns here are, can we trust on the decision outputs attained by these models? Are they fair enough? Are they unbiased? Are inputs provided to the model are self-sufficient to train the model properly and predict correctly? And Imagine how harmful it could be to a person, company, industry and a society when the decisions made are incorrect. However, improper deployment of these models can lead to inadmissible consequences though it is unintentional discrimination.

Let's look at some of the live examples of unintended discrimination which took place in Corporate Giants:

1. The Amazon's experimental hiring tool used artificial intelligence to give job candidates scores ranging from one to five stars. But by 2015, the company realized its new system was not rating candidates for software developer jobs and other technical posts in a gender-neutral way. That is because Amazon's computer models were trained to vet applicants by observing patterns in resumes submitted to the company over a 10-year period. Most came from men, a reflection of male dominance across the tech industry. In effect, Amazon's system taught itself that male candidates were preferable. Its penalized resumes that included the word "women's". Amazon edited the programs to make them neutral to these particular terms. But that was no guarantee that the machines would not devise other ways of sorting candidates that could prove discriminatory, the people said. The Seattle company ultimately disbanded the team by the start of last year because executives lost hope for the project,

according to the people, who spoke on condition of anonymity. Amazon's recruiters looked at the
recommendations generated by the tool when searching
for new hires, but never relied solely on those rankings, they said.**[1]**

2. A private company called "**Equivant**" (formerly Northpointe) developed "**Compas**" a machine learning algorithm that predicts the defendant's likelihoods to commit crimes, it has been shown that it makes biased predictions about who is more likely to recommit crimes. Research from ProPublica found that the tool was 2 times more likely to incorrectly cite black defendants as being high risk for recommitting crimes, it was also 2 times more likely to incorrectly predict that white defendants were low risk to recommitting crimes. Compas is used by judges in over 12 US states and it's used as a tool to many things such as figuring out whether people in jail should be let out on bail before trial, type of supervision on inmates, also, it has an impact on the length of people sentences. **The consequences of this algorithm are very real.**
Surely, we can't really say that this is intentional. It's not likely that the engineers who built Compas put bias into the system rather it's more likely that Compas was trained on a dataset that hadn't been exposed to different faces including skin tones.**[**2**]**
You can find the complete analysis here.**[3]**

"Garbage In Garbage Out (GIGO)"

A predictive model can be susceptible to discrimination if it was trained on inputs that exhibit discriminatory patterns. In such case, the predictive model can learn patterns of discrimination from data leading to high dependence on protected attributes like sexuality, race, gender, nationality etc. A predictive model that significantly weighs these protected attributes would tend to exhibit disparate outcomes for these group of individuals and thus, the above examples clearly depict this case.

Hence, the focus of this project is determine whether the provided data is biased on protected attributes and finding its relative significance so that, the user can know about the fairness in the data and hence, on confirming if it isn't biased he/she can use the same for training predictive models to attain an unbiased/fair output.

In the upcoming sections, one can find existing tools, Problem Description, Problem Solution, Results, Future Scope.

EXISTING TOOLS AND FRAMEWORKS

Tools that use to detect and correct bias in algorithms or datasets are lacking both for engineers developing their own products and for customers who want to optimize third-party AI systems. However, there are some of the tools in industry announced by Tech Giants. Also, some of the university researchers have created various types of debiasing tools thus, are interpreted as follows:

Google:

- Google What-If Tool - Google What-If Tool is a new feature of the opensource Tensor Board web application, which let users analyze an ML model without writing code. Given pointers to a TensorFlow model and a dataset, the What-If Tool offers an interactive visual interface for exploring model results. It has a large set of features, including visualizing your dataset automatically using Facets, the ability to manually edit examples from your dataset and see the effect of those changes, and automatic generation of partial dependence plots which show how the model's predictions change as any single feature is changed.

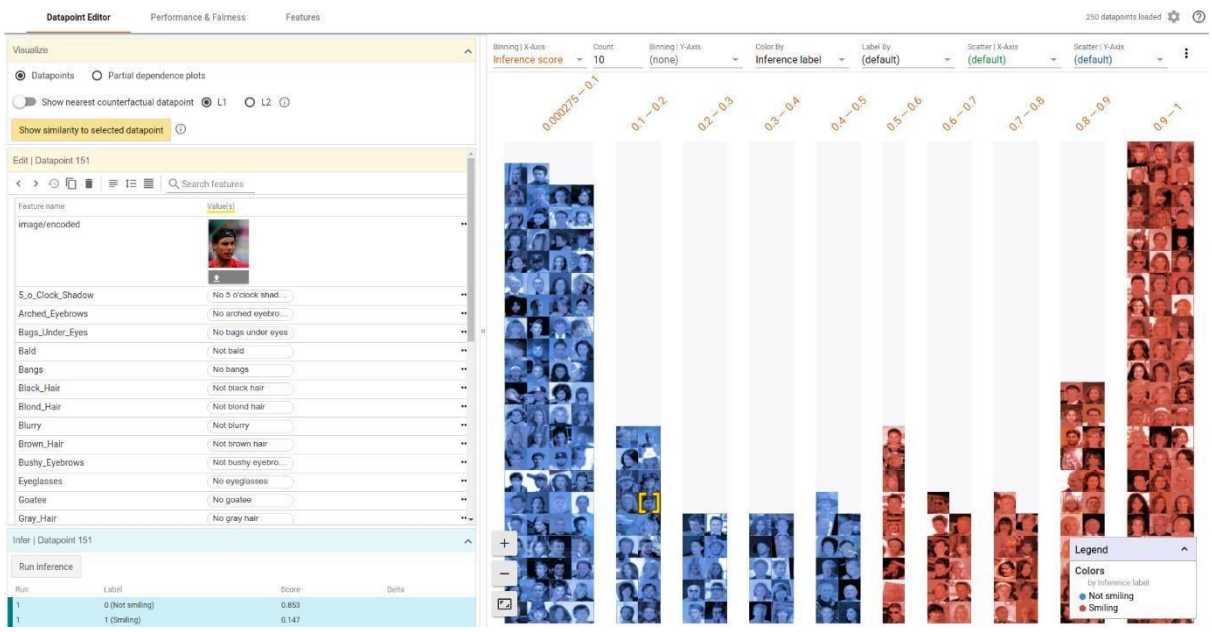

Figure 1: A visualization of the What-If Tool where the change in the model' output can be seen when the input parameters are changed.

IBM:

- **AI Fairness 360 –** This is an Open source toolkit for AI fairness curates advanced algorithms for detecting and eliminating different kinds of bias that can corrupt AI systems.
- **Diversity in Faces –** In here**,** Dataset of annotations for 1M face images for advancing the study of fairness and accuracy in facial recognition technology are available.
- **Watson OpenScale –** IBM built bias detection technology into Watson OpenScale so clients can perform bias checking and mitigation in real time when AI is making its decisions.
- **FactSheets for AI –** These are like nutrition labels for AI, providing key details like how the model was trained, what performance levels were achieved, whether or not the model was biased, or other important metrics.

Massachusetts Institute of Technology Computer Science and Artificial Intelligence Lab (CSAIL)

- Amini, Alexander et al. **[4]** describes a novel, tuneable debiasing algorithm to adjust the respective sampling probabilities of individual data points while training. The approach reduces hidden biases in training data and can be scaled to large datasets. A concrete algorithm for debiasing and an open source implementation of the model is provided.

Stanford Human-Centred Artificial Intelligence Institute:

- **Debiasing word embedding tool**- In word embedding models, each word in a given language is assigned to a high-dimensional vector, such that **the geometry of the vectors captures relations between the words**. For instance, the cosine similarity between the vector representation of the word *king* will be closer to the word *queen* than to *potato.*
- **AI audit tool –** audit AI is a tool to measure and mitigate the effects of discriminatory patterns in training data and the predictions made by machine learning algorithms trained for the purposes of socially sensitive decision processes

Industry-specific tools

- **Zest Finance –** ZAML Fair tool to reduce bias in AI-powered credit scoring models is part of the company's main ZAML platform, based on game theory.
- **Pymetrics –** Cloud-based human resources assessment tools help employers find candidates who best fit their needs, while reducing gender and racial biases. The company has open-sourced its Audit-AI algorithm bias detection tool.

PROBLEM DESCRIPTION

The bias in the machine learning model may be caused due to

a. Lack of appropriate dataset
b. Lack of sufficient set of features
c. Lack of good sampling of data (i.e. improper train and test split of data)
d. Lack of proper device or tool to measure.

Given that the data used for training the models are designed and gathered by humans, individual (data scientists or product managers) bias may get into the way of data preparation for training the models. This would mean that one or more features may get left out, or, coverage of datasets used for training is not decent enough. In other words, the model may fail to capture essential regularities present in the dataset. As a result, due to these factors the resulting Machine Learning models would end up reflecting in an underfitted model which has high bias and low variance.

PROBLEM SOLUTION

The implementation of the problem stated above i.e., detecting the factors that cause bias for a given dataset by understanding the variables has been achieved by fulfilling the following steps:

1. **Exploration/Analysis of diverse features that are potential of causing bias/unintentional discrimination:**

   The first step was to research the current frameworks and how they managed to reach the goal in identifying the features that caused bias and coming up with a set of features and ideas on how to resolve this discrimination. And also

examining various datasets/large scale data of various industries like Banking, Employment, Insurance, Housing, Fraud, Crime, Education.

In “A survey on bias and fairness in machine learning” the authors investigated different real-world applications that have shown biases in various ways, and have listed different sources of biases that can affect AI applications. Then created a taxonomy for fairness definitions that machine learning researchers have defined in order to avoid the existing bias in AI systems. In addition to that, they also examined different domains and subdomains in AI showing what researchers have observed with regard to unfair outcomes in the state-of-the-art methods and how they have tried to address them.**[6]**

2. Building a corpus

After certain analysis and groundwork done, I managed to find some of the key attributes which could result in bias. Using these attributes, I’ve built a corpus featuring set of attributes.

3. Developing a model where a generic code iterates through any kind of classifier and regressor dataset provided by the user through User interface and detects if there is some kind of bias

An n*k data matrix where

n= number of samples/rows

k= number of attributes/columns

→c= list of attributes in a data matrix
→p= list of protected attributes
→o= np. intersect1d (c, p)

```
for c in common:
  vals = (getattr(df,c).value_counts()/len(df)) *100
  no_of_vals = vals.count()
  if no_of_vals < 6 and no_of_vals>1:
    print('The percentage count of',c)
    print(vals)
  req_per = 100/no_of_vals
  for v in vals:
    if v < req_per or v > req_per:
      print ('Bias exists in ', c)
      break
```

The above algorithm iterates through the dataset and finds if any protected attributes are present in the data if found it will give us the statistical analysis of each category in the dataset and how they are distributed so as to know if the data has features that are biased to one or more such particular categories.

4. **Deploying it on to Heroku**.

Heroku is a container-based cloud Platform as a Service (PaaS) where I have deployed the code developed so that its user friendly and easily accessible. In here, a user can upload a CSV file and check for the features/variables that raise bias and the distribution of each of the unique values of the features (protected attributes) of the dataset.

The above developed module is readily available to use on
https://bias-checker.herokuapp.com/

Programming Language used: Python
Packages used: NumPy, pandas, matplotlib, pickle, sklearn, Flask
Front end: HTML, CSS
Cloud Application Platform: Heroku
Hosting service: Git
Others: Jupyter Notebook, Spyder

REASONING

In machine learning, discriminative models typically take the form: y = f(x) where y is the output of interest which can be a measure of creditworthiness in banking or health riskiness in insurance. x is the input data used in building a predictive model. x can consist protected attributes such as gender, age, income level, colour, religion, marital status and so on.

The main motto of this module-2 was to determine the relative significance of input attributes to a predictive model i.e. y outcome. In other words, it depicts the relation between design and feature matrix.

The present work mainly focusses on 2 categories broadly - data transformation, algorithm manipulation. The algorithms used for determining the relative significance are Random forests classifier using Logistic regression and orthogonal projection using Logistic Regression.

MODEL BUILDING

In many Machine-learning or Data Science activities, the data set might contain text or categorical values (basically non-numerical values). So, when a user inputs a dataset with categorical data like Hr recruitment data or sentiment analysis data, the machine can't understand. In order to make a machine learn and analyse the data and give output we need to convert into machine understandable language, Here I have done this using "**Label Encoder**". This encoder is a part of SciKit-learn library (one of the most widely used Python library) and are used to convert text or categorical data into numerical data which the model expects and perform better with.

The algorithm for label encoder is depicted as follows

```
from sklearn import preprocessing
from fairml import plot_dependencies
label_encoder = preprocessing.LabelEncoder()
df=df.apply(preprocessing.LabelEncoder().fit_transform)
```

Output gained after Label encoding

Out[132]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	40	4	1	2	0	3036	1	0	2	4	8	261	0	0	0	3	0
1	26	9	2	1	0	945	1	0	2	4	8	151	0	0	0	3	0
2	15	2	1	1	0	918	1	1	2	4	8	76	0	0	0	3	0
3	29	1	1	3	0	2420	1	0	2	4	8	92	0	0	0	3	0
4	15	11	2	3	0	917	0	0	2	4	8	198	0	0	0	3	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	33	9	1	2	0	1741	0	0	0	16	9	975	2	0	0	3	1
45207	53	5	0	0	0	2639	0	0	0	16	9	456	1	0	0	3	1
45208	54	5	1	1	0	5455	0	0	0	16	9	1116	4	181	3	2	1
45209	39	1	1	1	0	1584	0	0	1	16	9	508	3	0	0	3	0
45210	19	2	1	1	0	3779	0	0	0	16	9	361	1	185	11	1	0

45211 rows × 17 columns

Building design and feature matrix and assigning weights using PCA and Logistic Regression.

Large datasets are increasingly common and are often difficult to interpret. Principal component analysis (PCA) is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance. Finding such new variables, the principal components, reduces to solving an eigenvalue/eigenvector problem, and the new variables are defined by the dataset at hand, not *a priori*, hence making PCA an adaptive data analysis technique.

Out of all the possibilities in order to achieve PCA, we will focus on **orthogonal projection**. The reason that orthogonal projection out of all possible projections is of my interests is because of the closest vector property. According to the **Closest Vector Property,** among all vectors in W space, the vector closest to **u** is the orthogonal projection of **u** on W. In other words, we want to get the projection that is closest to the original dataset to maintain as much information as possible after decreasing the dimension.

INPUT: An n x k data matrix X_{pre} that can be decomposed into $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_k$ column attribute vectors, where:
n = number of samples,
k = number of features or attributes being considered, and
$\vec{x}_1$ = current attribute of interest.
OUTPUT: An $n \times k-1$ transformed matrix X_{new} that can be decomposed into $\vec{x}_2^*, \vec{x}_3^*, \ldots, \vec{x}_k^*$ where each vector $x_i^* \in X_{new}$ is orthogonal to current feature $\vec{x}_1$.

Delete current vector $\vec{x}_1$ from X_{pre} returning X_{del}
Initialize an $n \times k-1$ vector X_{new}
for each attribute vector $\vec{x}_i$ in X_{del} **do**
obtain x_i^*, the component of $\vec{x}_i$ that is orthogonal to current feature vector $\vec{x}_1$
where $x_i^* = \vec{x}_i - (\frac{\vec{x}_1 \cdot \vec{x}_i}{\vec{x}_1 \cdot \vec{x}_1})\vec{x}_1$
join x_i^* column wise to X_{new}
end for
Return X_{new}

Then the output of PCA (orthogonal linear combinations of the variables that explain the most variability in the data) is fitted to the model. The model here built is with the help of logistic regression.

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable.

In [134]:
```
# create feature and design matrix for model building.
y = df.y.values
x = df.drop("y", 1)

clf = LogisticRegression(penalty='l2', C=0.01)
clf.fit(x, y)
```

Out[134]:
```
LogisticRegression(C=0.01, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

The following figure depicts the result for the same.

```
Feature: age,    Importance: -0.03308929242883369
Feature: job,    Importance: -0.02968304173763022
Feature: marital,        Importance: -0.025989250403662824
Feature: education,      Importance: 0.026387383601335957
Feature: default,        Importance: -0.0007741478843644246
Feature: balance,        Importance: 0.034394284576762295
Feature: housing,        Importance: -0.02076928181194842
Feature: loan,   Importance: -0.00661343478357037
Feature: contact,        Importance: -0.015394483643361129
Feature: day,    Importance: -0.03404038840105284
Feature: month,  Importance: 0.027117294463736702
Feature: duration,       Importance: 0.022737829289332243
Feature: campaign,       Importance: -0.02990422684744863
Feature: pdays,  Importance: 0.006547079250624848
Feature: previous,       Importance: 0.00946672270022782
Feature: poutcome,       Importance: -0.023976465904315323
```

Figure 4

From the output we can see that the values are normalised between 0 to 1. Also, negative and positive signs indicate negative and positive correlation between the input and output variables. With this final output we can attain the information about the features that are affecting the y predicted.

- **Thumb Rule while inputting a dataset to this module:** The Y variable (the predicted variable) should be renamed as 'y', in the dataset provided in order to iterate and procure the results.
  For example, if the predicted variable/output variable in the dataset is with the name 'Job Status' should be renamed as 'y'

- The importance in the results state that how that particular feature is correlated to the variable y, if the importance of the feature is in negatives, then it is said to be negatively correlated to the variable y and vice versa.

USE – CASE

SCENARIO-1

1. This use-case tells us about the feature importance when categorical data is served as input.
2. The dataset that is supplied here is "HR RECRUITMENT DATASET"
   By this we can assure that this model is capable of using in IT Industry.

INPUT:

```
In [18]: df = pd.read_csv("Hr_recruitment_data.csv")

In [19]:
df.head(10)
```

Out[19]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	y	salary
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.00	Mkt&HR	58.80	Placed	270000.0
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.50	Mkt&Fin	66.28	Placed	200000.0
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.00	Mkt&Fin	57.80	Placed	250000.0
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.00	Mkt&HR	59.43	Not Placed	NaN
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.80	Mkt&Fin	55.50	Placed	425000.0
5	6	M	55.00	Others	49.80	Others	Science	67.25	Sci&Tech	Yes	55.00	Mkt&Fin	51.58	Not Placed	NaN
6	7	F	46.00	Others	49.20	Others	Commerce	79.00	Comm&Mgmt	No	74.28	Mkt&Fin	53.29	Not Placed	NaN
7	8	M	82.00	Central	64.00	Central	Science	66.00	Sci&Tech	Yes	67.00	Mkt&Fin	62.14	Placed	252000.0
8	9	M	73.00	Central	79.00	Central	Commerce	72.00	Comm&Mgmt	No	91.34	Mkt&Fin	61.29	Placed	231000.0
9	10	M	58.00	Central	70.00	Central	Commerce	61.00	Comm&Mgmt	No	54.00	Mkt&Fin	52.21	Not Placed	NaN

```
In [25]: df=df.apply(preprocessing.LabelEncoder().fit_transform)

In [26]: df
```

Out[26]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	y	salary
0	0	1	46	1	93	1	1	14	2	0	9	1	64	1	19
1	1	1	82	0	79	1	2	74	2	1	74	0	153	1	0
2	2	1	42	0	51	0	0	27	0	0	55	0	50	1	12
3	3	1	22	0	14	0	2	3	2	0	37	1	72	0	64
4	4	1	98	0	70	0	1	65	0	0	96	0	28	1	39
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	210	1	85	1	84	1	1	76	0	0	83	0	199	1	36
211	211	1	25	1	24	1	2	60	2	0	52	0	14	1	20
212	212	1	46	1	49	1	1	64	0	1	25	0	179	1	27
213	213	0	64	1	45	1	1	14	0	0	45	1	81	1	1
214	214	1	36	0	21	1	2	4	0	0	80	1	80	0	111

215 rows × 15 columns

OUTPUT:

```
Feature: sl_no,  Importance: 0.037209302325581395
Feature: gender,         Importance: 0.023255813953488372
Feature: ssc_p,  Importance: 0.13023255813953488
Feature: ssc_b,  Importance: 0.027906976744186046
Feature: hsc_p,  Importance: 0.10697674418604651
Feature: hsc_b,  Importance: 0.03255813953488372
Feature: hsc_s,  Importance: 0.05116279069767442
Feature: degree_p,       Importance: 0.09767441860465116
Feature: degree_t,       Importance: 0.03255813953488372
Feature: workex,         Importance: 0.06511627906976744
Feature: etest_p,        Importance: 0.07441860465116279
Feature: specialisation,         Importance: 0.004651162790697674
Feature: mba_p,  Importance: -0.046511627906976744
Feature: salary,         Importance: -0.046511627906976744
```

SCENARIO-2

COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) is a popular commercial algorithm used by judges and parole officers for scoring criminal defendant's likelihood of reoffending (recidivism). It has been shown that the algorithm is biased in favor of white defendants, and against black inmates, based on a 2 year follow up study (i.e who actually committed crimes or violent crimes after 2 years). The pattern of mistakes, as measured by precision/sensitivity is notable.

In here the y -predicted outcome is "Deci score"

```
Feature: Person_ID,      Importance: -22.236000854658712
Feature: AssessmentID,   Importance: 23.698847854313563
Feature: Case_ID,        Importance: 22.561888795753003
Feature: Agency_Text,    Importance: 39.358956658941864
Feature: LastName,       Importance: 25.94165310717749
Feature: FirstName,      Importance: -25.04366977302237
Feature: sex,    Importance: 28.767204115510413
Feature: Ethnicity,      Importance: -24.178525056292425
Feature: DateOfBirth,    Importance: 25.366188386502966
Feature: ScaleSet,       Importance: -34.20043390365367
Feature: AssessmentReason,       Importance: 35.8311062899594
Feature: Language,       Importance: -0.15068290518219024
Feature: LegalStatus,    Importance: 34.29472576960374
Feature: CustodyStatus,  Importance: 16.11976726985849
Feature: MaritalStatus,  Importance: 30.89303617507355
Feature: Screening_Date,         Importance: -26.5526683431126
Feature: DisplayText,    Importance: 35.44044507995989
Feature: ScoreText,      Importance: 36.65856384464934
Feature: AssessmentType,         Importance: -35.53642982758904
Feature: Unnamed: 20,    Importance: 27.512614433870784
```

SCENARIO-3

This data set is created for the learning purpose of the customer segmentation concepts, also known as market basket analysis.

Suppose you are owing a supermarket mall and through membership cards, you have some basic data about your customers like Customer ID, age, gender, annual income and spending score. Spending Score is something you assign to the customer based on your defined parameters like customer behaviour and purchasing data.

So, let's see if there is any kind of bias towards a particular category while assigning spending score. Here, "spending score" will be our y- predicted outcome.

```
Feature: CustomerID,     Importance: 1370.525
Feature: Gender,         Importance: 360.45
Feature: Age,    Importance: -615.265
Feature: Annual Income (k$),     Importance: -1185.725
```

This tool can also be used in other domains such as education, insurance, heath, IT etc.

But while inputting raw data such as tweets, wordings, direct statements pulled off from the internet should be first cleaned/pre-processed like removing emails, hashtags, mentions etc. so that the model can perform better and efficient.

SWOT ANALYSIS

Strengths

1. Identifies quickly the features that raise unintentional discrimination.
2. Tells whether the dataset provided is Biased or Unbiased in form of statistical analysis.
3. Also lets us know whether the dataset can be used for training machine learning model so that we could gain accurate and unbiased decisions.
4. Easily understandable and user friendly.

Weaknesses

1. The present developed tool is limited to numerical and categorical data only. It doesn't work for images, video and audio files.
2. The tool becomes efficient only when more and more of data is trained so that the attributes are updated in the corpus further leading to accurate decisions.

Opportunities

1. There is a much brighter scope for this system as it is accessible to everyone unlike restricting the usage to a single software.
2. Accuracy in results may be another motive for an increased usage of current tool.

Threats

1. Chance of giving out wrong results if the user fails to be honest in providing the dataset.

RESULTS

The following are the snapshots of the developed tool UI.

Figure 2: The above figure represents the UI after deploying it on Heroku where a user can upload a CSV file to check for the bias in provided dataset.

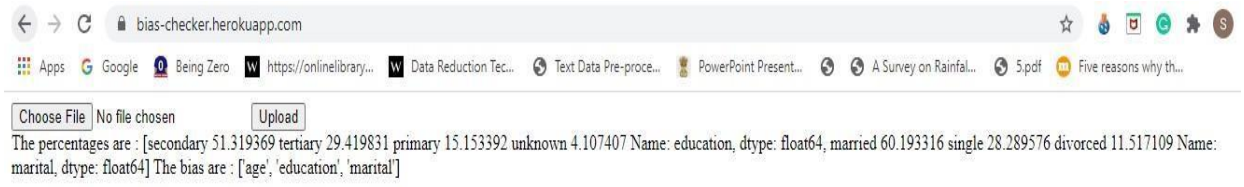

Figure 3: The above figure represents the results after the file is uploaded.

CONCLUSION

There are base number of tools in the market to detect AI bias and fairness. Anyhow each one has its own way of costs and benefits. The present developed tool "bias checker" has addressed the cause for bias in industry from ground level.
From analysing kinds of bias, Handling and preparing data for tool, checking for which combination of algorithms would mitigate this problem, addressing multiclass, classifiers and regressors. Finally, I would say the tool is capable of detecting bias in a dataset understanding the variables for numerical and categorical data and how specific input features are affecting the output/predicted outcome.

But is the present tool better than other tools?

I would like to answer this question considering and comparing the present developed tool with the other tool namely Google What-If tool.

1. Bias Checker tool is unique and distinguishable in such a way that this doesn't require any specific cloud environment and still can function better on local servers promoting explainability which most of the companies hunt for, whereas Google's tools (now in beta) require customers to build and implement their models within the Google Cloud which seems undesirable.
2. This tool doesn't enforce developers to learn any specific coding language it can also be used by a lay man whereas to use what if tool developers will have to learn a specific coding language and configuration convention to access explainability functions.

The other tool which I would like to consider is "IBM AI Fairness 360".
According to me, this tool is an amazing "Bias checker and mitigator", (consisting 4 stages namely Selection of dataset, checking bias metrics for the same, choosing bias algorithm to mitigate bias and then finally comparing the original results with mitigated results.

But is it generic and applicable to all kinds of datasets? I don't think so. Fairness 360 is limited to 3 datasets only. But in today's world of data there is 2.5 quintillion bytes of **data** are **produced** by humans **every day** and requirement will be huge when coming to bias checker and mitigation. In the appropriateness of toolkit, they mentioned that "Fairness is a multifaceted, context-dependent social construct that defies simple definition. The metrics and algorithms in AIF360 may be viewed from the lens of distributive justice , **and clearly do not capture the full scope of fairness in all situations. The toolkit should only be used in a very limited setting: allocation or risk assessment problems with well-defined protected attributes in**

which one would like to have some sort of statistical or mathematical notion of sameness. Even then, the code and collateral contained in AIF360 is only a starting point to a broader discussion among multiple stakeholders on overall decision-making workflows" whereas the "Bias- checker" functions in a broader scope.

Finally, I would conclude saying "bias-checker" is the first step towards detecting unwanted bias and algorithmic fairness, which by nature is a form of statistical distribution that works efficiently with numerical, categorical, multiclass data.
The future work can make this tool even interactive and user friendly.

FUTURE SCOPE

1. To make this system user friendly, changes can be made to use this as a plugin directory to include fully customizable field and editor support.
2. The tool can be extended for detecting bias in image, audio and video files.
3. This tool can be extended in its way to mitigate the bias found from the same.

REFERENCES

1. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secretairecruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G
2. https://towardsdatascience.com/5-types-of-bias-how-to-eliminate-them-in-your-machinelearningproject-75959af9d3a0
3. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm
4. "Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure." *AAAI/ACM Conference on AI Ethics and Society*. 2019.
5. Multiaccuracy: Black-Box Post-Processing for Fairness in Classification, Kim, Michael P. et al. 2018.
6. https://arxiv.org/pdf/1908.09635.pdf

DATASETS USED

1. ttps://www.kaggle.com/search?q=bank+dataset
2. https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling