Subject area 2: Bias in Artificial Intelligence

A.C Zevenster

MSc, Artificial Intelligence

IUBH University of Applied Sciences

IT and Engineering

February 3, 2021

Contents

Abstract

Bias in Artificial Intelligence has been a highly debated; yet seemingly misrepresented on mainstream platforms. Bias can of-course occur due to bias societal activities that lead to skewed representations; in practice bias arises due to an innumerable number of aspects. Some of the key influences that can result in bias throughout the pipeline of building a model include: Data and preprocessing, during this stage the type of data and data sources might be of concern, such as missing values; Bias can also occur in the actual model selected when dealing with a certain task, as different models can have different interpretations dependant on the data used; Finally bias can also occur in the model evaluation phase, during this phase the validity of a model is assessed and utility determined - the utility can at times be over- or under- estimated. This paper looks aims to bring awareness to common practical injections of these biases on a general term.

1 Introduction

Understanding the mechanisms of bias in predictive models has been widely misunderstood; in most cases due misinterpretation or selective interpretation. Missing data has a correlation to algorithmic bias, a problem that is actively addressed. Where the common assumption: algorithms are inherently bias due to the algorithms developer - which is rarely the case. To delve deeper into the understanding of biases, it is imperative to understand different mechanisms in bias; as understanding them could help in preventing them. The analysis on bias starts with a exploration on the type of data that is being dealt with and implication the type might have on the model. Primarily we will classify data as MCAR, MAR or MNAR[14] Effects of these common data imputation pitfalls is then analysed using both classification and regression models. To quantify the effects of bias, after the understanding the effects of data impurities on predictive models; methods to address data impurities is addressed and the effects of these methods on models - this is not a in-depth approach. Further, improvements to models to better interpret both bias in data and in the model is discussed and suggested. Model Tuning techniques include regularization, data standardization are are all proven to be useful in addressing algorithmic biases but will not be elaborated upon.

2 Background

2.1 Data and Preprocessing

The Preprocessing step has a few constituent parts. Biases in data has been described in 3 distinct categories.

- **Historical Bias**, Discrepancies between true events and captured events that are propagated through the model.
- **Representation Bias**, under-representation of a population group, causing failure to generalize for a specific population. Example of this is Gender Pay Gap.
- **Measurement Bias**, ambiguity in selection of features and labels, leading to noise and neglecting of important information. [22]

Looking at adjusting datasets instead of finding the "ideal" datasets is a more practical approach. We will address the effects of MCAR, MNAR, MAR and auxiliary variables, it is important to choose the most efficient data handling method in respect to the type of model being used. In certain models using a auxiliary variable emulated a more stable model, under certain conditions replacing missing values with a average or median obtained less volatile models. We will look at use cases of missing data and their potential repercussions on predictions along with potential solutions to combat data handling challenges more directly.

Concepts such as data cleaning - where outliers in data can be detected; data normalization - this is seen as a preprocessing step - where data is scaled, normalized and mean centering can be performed. Missing value treatment has become one of the resent topics, missing values can have an effect on data and models especially in longitudinal studies. Imbalance data treatment is the last concept that will be explored, where oversampling, undersampling and mixed sampling is dissected. Data transformation and data engineering are also methods of improving model performance but will not be discussed in this paper.

Taking this a step further, preprocessing of data has also been seen to affect bias in models. Investigation is put into the measurement and analyses of certain preprocessing steps in models. One of the most prevalent problems are *high dimensionality datasets*, dimensionality reduction could be introduced in the preprocessing stage - before creation of *train-test-splt* partitions in the model- by detecting low quality features, however, we might fall prey to *"The Curse of Dimensionality"*; as features increase the number of observation per feature needs to increase exponentially, Adjusting for these phenomena needs to be considered.[13]

2.2 Models

Classification and regression Models both include their own challenges when handling the data, it is important to understand how the model can skew biases with certain data sets and their respective types. Different models will handle data sets and their challenges differently. There are predominantly 3 types of bias that can be introduced into our models. When the feature used in our model is not covered uniformly *co-variance shift* is introduced to our model. When our ratio of labels are not uniform we deal with a *imbalance bias.*When our labels and features of our model do not seem the correlate we are dealing with *sample selection bias.*[17] In the data preprocessing stage; when the missing data can not be ignored, there has been suggestions on the implementation of: *pattern mixture, shared parameter models and selection* has been suggested as remedy but will not be explored.

Transparency and fairness :*transparency and fairness* is two of the metrics on which commonly evaluated, where *'transparency'* is the processing steps taken by a designated model and 'fairness' is replication of results, within a acceptable parameter.[18]. Hyper-parameter adjustment methods are essential in handling data discrepancies, which model optimization method is best suited to handle certain datasets keeping the class of desired prediction in mind. Overfitting and underfitting are the most forward issues to address in Models before more complex alternatives are considered. *Underfitting* occurs when when the error score is too high on the training set and *overfitting* occurs when the gap between training and test datasets are too far removed from each other, when overfitting occurs the model looses subtle

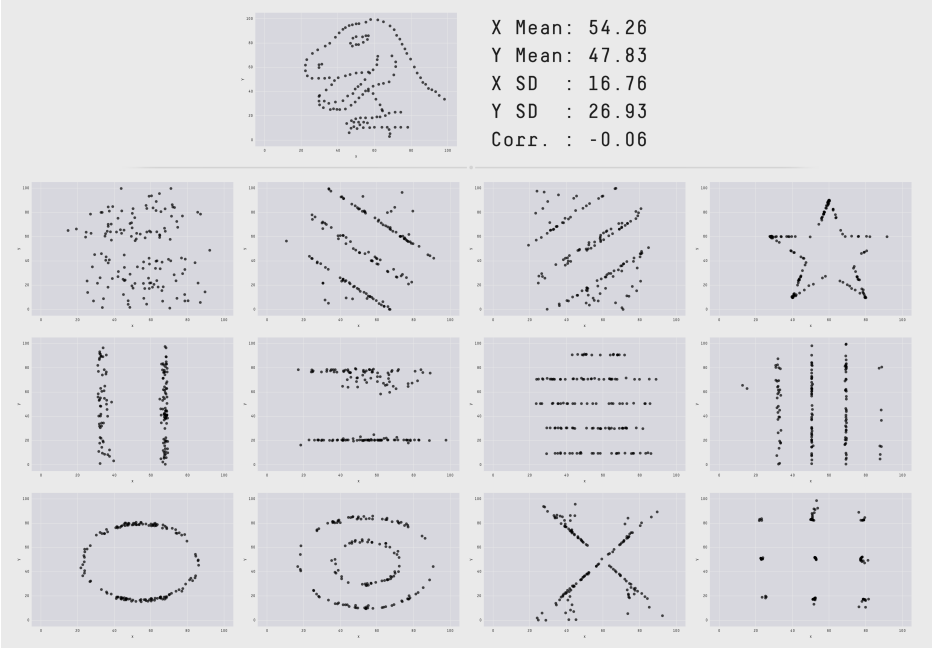


Figure 1: Datasaurus dozen.[21]

features. [8]. Transparency and fairness can assist in addressing these anomalies.

2.3 Evaluation

Linear model works on continuous variables and classification models makes prediction on labels, thus they both require different evaluation methods. For regression we use *Mean Absolute Deviation(MAD), Mean Absolute Percentage Error(MAPE) and Root Mean Squared Error(RMSE)* to evaluate different types of datasets.[10] Along with statistical analysis a visual analysis is also adept at providing correlation in data, visualization can assist in both the evaluation process and preprocessing stages. Feature visualization can assist in identifying correlated features and in evaluation visualization can assist in detecting correlation caveats such as *Anscombe's quartet*, where high similar correlations have different distributions when visualized. [15]

To illustrate the importance of visualization, consider the 2017 study by Steph Lock, *'The Datasaurus Dozen' (1)* , a collection of 142 datapoints that resemble a dinosaur (T-Rex) when graphed. 12 additional graph sets are also created, each containing 142 pieces of data, the all create completely different patterns but all have the same average value ($\bar{x}$),standard deviation (σ), and *correlation coefficient* (r) when visualized; it is clear that it has no true correlation. Illustrating the importance of the powerful tool, if interpreted correctly.[15]

3 Data and Preprocessing

Data are typically represented in 3 form: *structured data*; these are data points received from devices such as hardware sensors. *semi-structured data*, data that is not necessarily in tabular form but are still tagged and is more convenient to deal with than unstructured data, this type of data is normally in the

form of XML files or website data (HTML data); lastly, unstructured data; this data is not organized in any pre-defined manner, sound, images and video recordings are instances of this type of data.[22]

3.1 Missing Values

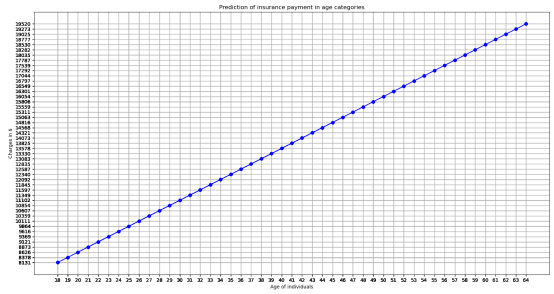
It is standard practice to remove incomplete, unsuitable and abnormal data. In some cases, this approach might not be viable, such as in medical data, that are notoriously incomplete. When gathering data resources, numerical data - including converted types, such as text - has common 3 categories; *MCAR* - data is missing completely at random; *MAR* - data is missing at random; *MNAR* - when data is not missing at random. MCAR is categorised as such when the missing data has no effect on the outcomes. MAR occurs when missing data is dependant on another variable. When data can neither be classified as MAR or MCAR then it is classified as MNAR, in MNAR the reason for missing data is unbeknownst.[23] Some of the commonly used methods to handle missing data of the types formerly discussed include: *List- Wise Deletion; Complete Case Analysis; Available Data Carried Forward; Last Observed Carried Forward; Conditional an Unconditional Mean Imputation*, these will however not be covered in this paper. When the missing data is ignore-able 2 methods; *Maximum - Likelihood*[1] *and Multiple Imputation*; are used, both being well suited for practical implementation, besides these handling methods, an auxiliary variable can also be introduced and model performance can be evaluated[1]

3.2 Preprocessing

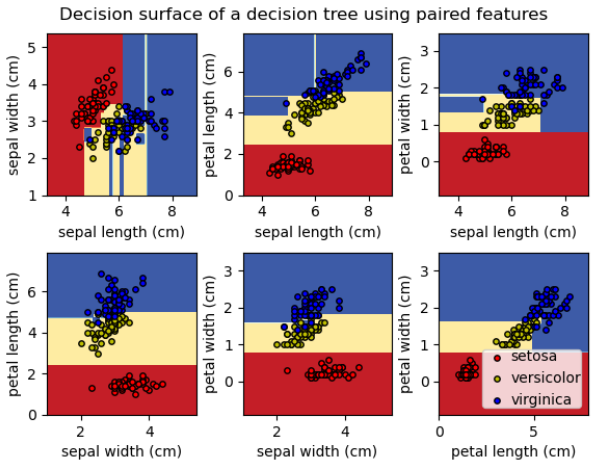
High Dimensionality is one of the leading problems in "Big Data", making it difficult to extract meaningful metrics, along with this domain complexity can cause many errors, such as in medical predictions, where it can be difficult to ascertain to contributions of features to a model. In the preprocessing stage data in converted and treated to a form more suitable to the model. Most models to not work well with text data, thus the text data must first be converted into a usable format. This can be done by a text vectorizer; text vectorizers has various methods of encoding strings: TF - IDF, word embedding, bag-of-words or pre-trained models are all used for this process. It is also common to use a machine learning methods - such as unsupervised learning - in the preprocessing stage to gain better understanding of feature sensitivity or class distribution. K - Means Clustering, where the features will be distributed in classes, allowing for better conceptualization of such cases, thus implementation of such algorithms should be considered in as wide a range of *spaces* as possible.[3]

Visualization Visualization should be one of the fundamental steps in preprocessing, this allows for a overview of datasets, as already mentioned with 1 in the 2.3 section. Allowing for better conceptualization of correlation between variables and feature (column) importance. Visualization can also give a better understanding on evaluation methods - such as to implementation of root mean squared error over mean squared error (see eq.3 and 2) in handling of outliers.

[1] algorithm that is used to increase the likelihood of an observation



(a) Regression Model



(b) Classification Model.[2]

Figure 2: 2a is a prediction on insurance prices from —dataset and 2b is a representation of decision boundaries in a classification model from well know Iris Dataset available in *scikit-learn.*(see appendixA and appendixB for code listings)

4 Model Optimization

Overfitting and Underfitting are the most forthcoming problems to be avoided. Naturally, models handle data discrepancies differently. Thus choosing the optimal model for a specific data should be emphasized. In classification algorithms, after the preprocessing step, overlapping of features can occur, this is often referred to as noise - instance of *aggregated bias*(5.3) - , where distinct populations are not appropriately combined. Machine learning algorithms have various methods of safe guarding against these occurrences. We will briefly look at selecting a model type - model type and machine learning algorithm can be used interchangeably - model hyperparamaterization - pruning is a analogous term - and model optimization are all used to improve model performance.

4.1 Model Type

Classification and regression models have distinctly different uses. Regression models work with a continuous variable and should is density distribution. and Classification Models creates decision boundary, classes are labels based on the features. In both these cases, they have different mechanism to manipulate these boundary thresholds.In image(2a) and image(2b) illustrate a simple regression and classification model visually.

In image(2a) 'age' and 'charges' features is taken from a the *Kaggle Insurance Dataset*[12], and a *logistic regression* model trained on the dataset. It can then be deduced from image(2a that 'age' and 'charges' are proportional ($'age' \propto' charges'$),this is only selecting 2 basic features, with no preprocessing done, if further features are introduced more precise estimations can be made. Using the method presented could lead to a bias proxy and is a clear example of underfitting. In 2b, data is taken from the *iris* dataset in python. Choosing the same evaluation metric as in image2a would lead to innacurate results, furthermore, more complex datasets such as in image(2b) leads to higher dimensionality on more cracks for bias to

seep into, such as noise,

SVC To illustrate the importance of choosing the correct model further; If you are working with $n < 1000$ samples;a *Linear Support Vector Classifier (SVC)* handles such samples more efficiently,for samples $n > 1000$; stochastic gradient decent is more appropriate. Models can also be combined with preprocessing methods to create an overall 'better' model.

PCA *Principal Component Analysis (PCA)* is predominantly used as a dimensionality reduction method, mentioned in 2.1 and can be incorporated into the model pipeline. PCA acts by creating a vector on a label and assigns a relative weight to each feature. Thus, PCA allows is to determine the importance of each feature and it's *sensitivity*, refer equation 4 for further elaboration of *sensitivity feature*.

4.2 Combating Bias

Specific Models are adept at processing certain datasets; be it with structured , unstructured or semi-structured datasets, discussed in 2.1 . These models can be adjusted to a further extent depending on the nature of the data handling difficulties that can be faced throughout the model development pipeline.

LogLoss : Consider *LogLoss*, represented in eq(4.2), *LogLoss* is a model evaluation method where the models penalizes high deviating (σ) values, the higher the deviation the heavier the penalty on the LogLoss score, causing a 'poor' model prediction. Weighted LogLoss is often used in dealing with unbalanced data.[24]
Model contain various such evaluation methods, each more adaptive to certain models. Standard regression models normally work with LogLoss but for classification an evaluation method such as cross validation is more appropriate.

$$H_p(q) = \frac{-1}{N} \sum_{i=1}^{n} y_i \cdot log(P(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \tag{1}$$

Cross-Validation : *Cross-Validation* compare the vector position of each node - thus it would be an unsuitable evaluation metric for a regression problem, however, data sourcing is not always consistent, thus the choosing appropriate evaluation methods based on the appropriate model at that time position when dealing with a certain *type* of data.[1]

5 Evaluation

5.1 Statistical Analysis

Selecting an appropriate evaluation metric based on the source of data worked with, the preprocessing that occurred and model chosen can be difficult task, this can be the sheer size of data or features and population that were not completely understood; employing the correct evaluation method based on models can be very valuable. Consider *mean absolute error(MAE)*(3) over *mean squared error(MSE)*(2).

If your dataset has many ouliers, MSE will be a very poor decision as a evaluation metric as it is very sensitive to outliers. MSE is most commonly used in regression tasks.

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(p_i - t_i)^2 \tag{2}$$

Where MAD(3) on the other hand deals a lot better with outliers but might have other adverse result on evaluation, such as a high volatility in our model can lead to a large uncertainty in optimal point estimator in predictive models. In each of the aforementioned cases predicted and actual values are are taken into account.[10].

$$MAD = \frac{1}{N}\sum_{i=1}^{n}|p_i - t_i| \tag{3}$$

5.2 Measuring Bias

To determine whether our prediction were biased or unbiased can at times also be difficult to distinguish, one of the suggested used to evaluate whether bias is present is the *Calder Verwer Discrimination Score*. It is difficult to ascertain meaning behind a single prediction method, many methods such as the *"Bias-Variance Trade off"* - which is the anomaly where model with *high bias* pays restricted attention to the training data of the model and *high variance* focuses heavily on training data, however these methods are only useful if we know what predictions we wants to be made[20]. When it is completely unbeknownst to us what we are exactly trying to determine can become difficult. Thus methods such as the *Calder Verwer Discrimination score (4* score has been suggested, where S is a measurement of a sensitive feature. This could assist to determine when occurrences such as *illusionary correlation occurs* mentioned in 6.2, has occurred, by comparing ratios of predictions with sensitive feature variations.[8]

$$CV \leftarrow [\hat{Y} = 1|S = 1] - P[\hat{Y} = 1|S \neq 1] \tag{4}$$

Where: S represents the sensitive feature, in 4, $S\neg 1$ represents the minority feature.[8]

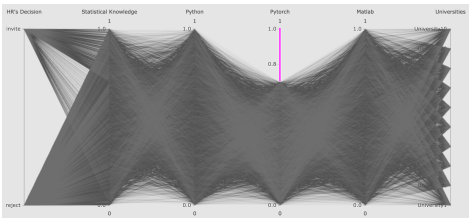
5.3 Model Evaluation

When working with model evaluation we often encounter *aggregation bias* - aggregation bias occurs when distinct population are inappropriately combined, this has been a particularly prevalent problem in medical data.[5] Evaluation bias is also another model bias that is often encountered, evaluation bias occurs when apt performance measures is not used, this error can increase on each model iteration, creating a growing gap between predicted and actual outcomes, models that were acceptable during evaluation at point t_i in space C_1 is no longer appropriate as the data changes to form a new space C_2.Examining the environment constantly may be of benefit when considering model architecture. Noise is also a injection point for many of the biases in models, corrupted and "messy" data distort models performance, again emphasising importance of being diligent when selecting features from data. Unsupervised learning techniques have been used to regulate representation bias, the methods works by trying to minimize the loss function over a certain space, by doing this divergence can be measured. [17]

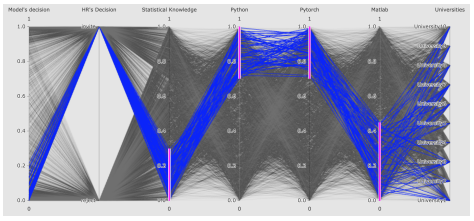
6 Use Cases

Throughout the investigation we will look at methods deployed both in research and industry, emphasis is put on preprocessing, Finding the correct model and optimizing evaluation respectively, where in each use case the the aforementioned steps as the central challenge. This will illustrate the value and possible implications that might arise in model development steps mentioned in 3, 4 and 5.

6.1 Data and Preprocessing



(a) Initial observations with corresponding features-excluding positive outcomes of a- single feature.[8]



(b) This image refers to the updated version of 3a where observation for the feature of *pyTorch* is present[8]

Figure 3: Parallel Coordinate plots are used in this representation to express the relation between observation and their features. With the parallel coordinate plot, each observation can be mapped. Brushed region in 3b maps outcomes of points that fall within the bounds of the specified region[8]

In image (3) is an encapsulating example of one of the simpler 'pitfalls' that can occur in both the data collected and its representation in the model. In the use case represented in image(3) recruitment agencies are looking to create a automated system for selecting employee candidates. Recruiters make their decision based on the skill aptitude for pre-defined skills a candidate must possess. [2] Should a candidate achieve an acceptable score in 2 of the main skills he would receive an invitation to an interview. In image(3a) there are 5 skills that the applicant is assessed on: MatLab [3]; PyTorch[4]; Python[5] and statistical knowledge. In **??** there are not any 'acceptable score' for the programming language *pyTorch* - possibly due to the juvenile nature of the skill, thus no important information is carried by the feature 'pyTorch.' In image(3b) suitable information for the absent skill of *pyTorch* has been obtained: however when mapped instances where candidates received a 'high' score for skills : *pyTorch*, was predicted to not receive a invitation to the interview where the recruitment decision was clearly to send an invite. The speculated reasons for this occurrence is due to the absence of positive - 'sending an invite' - instances with skills pertaining to *pyTorch*. This provides clear examples of *sample selection bias and imbalance bias*, both mentioned in 2.2.[8]

6.2 Models

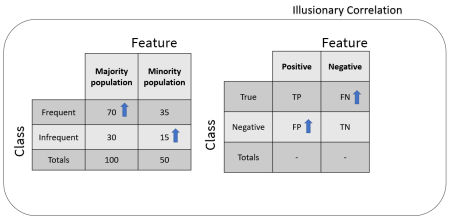
Consider the phenomena of *'Illusionary Correlation'* described in image(4). *Illusionary correlation* is typically a psychological phenomena - where a correlation between two variables are inferred when in-

[2]These pre-defined requirements make this a ideal task for automation because skills can be represented as the features which decision are based on

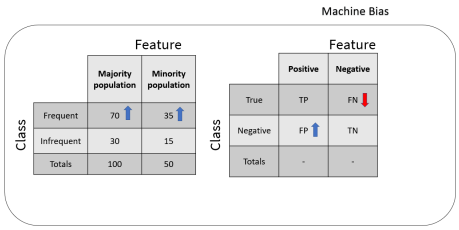
[3]programming language

[4]machine learning library in python

[5]Programming Language



(a) This image describes the psychological phenomena known as *Illusionary correlation* and how it effects would be represented as *classification* errors - here *frequent-minority classifications and infrequent-minority classifications* are accentuated



(b) Machine Bias presents similarities between illusionary correlation however in machines this phenomena is manifest differently with the increase of both *frequent - majority classifications and frequent-minority classifications*

Figure 4: 4b refers to how the phenomena of 4a is represented in machine terms - similar psychological disparities are observed in machines in image 4b[5]

fact there is no true correlation[11]. Image (4a) represents how this phenomena would represent itself as a *confusion matrix for binary classification* - this is briefly mentioned in section(4.2). The columns represent the majority poulation - if a sample of 100 individuals are taken; of which 60 are male and 40 are female, females would represent the *minority population*. The rows represent the observations of a certain variable in the class; where the *'frequent' class* would represent a observation that occurs less frequently. Take for instance a large annual salary compared to a smaller more 'median' salary, the infrequent class would represent the larger salary. In *illusionary correlation* this would be represented as a increase in *false negative classifications* and increase in *false positive classifications*. In image 4b the same psychological phenomena in algorithms are observed, although they are represented differently - speculation suggests due to differences in thinking humans and machine utilise. In the left table of image 4b represents how the *machine bias* can be represented in terms of features and class observation as in image 4a, when expressed as a confusion matrix however; inversely as to *illusionary correlation*: the *false negative classifications* are decreased.*False positive classifiactions* are increased - as with *illusionary correlation*. it has been suggested that methods to address these unexpected phenomena mention in 4 is the use of the *Calder Verwer Discrimination Score*4 to - at the very least - be used as a method to identify when such abnormalities might occur. [5]

There are various methods of evaluation models, and so many being introduced that it can at times be a disarray what evaluation method is appropriate to which use case. Due to the fluctuating nature of data, choosing appropriate models can at times be cumbersome; by no means impossible.
Consider the *predictive maintenance (PdM)* of a model;

Predictive Maintenace *PdM* is concerned with the evaluation of a model especially when the source of input is in constant flux. At any given moment in time, dependant on the type of data available; one model might be preferred over the other. In recent studies it has been suggested that ML methods can be used to automate these predictive maintenance of these models - a ML process to address a ML problem. [6] As data sources in the for of the "5Vs" fluctuate at any point in time - *volume, velocity, variety, veracity and value* - an different model might be preferred. Although this might be deemed as trivial actions it could lead to the ultimate decision whether a technology is *'adopted'* or abandoned - also known as AI winters. Explainable AI has been a resent development in Machine Learning application,

where a ML model is used to explain the *'Black Box Paradox'* - referring to the effect where the process between input and output is somewhat ambiguous - and sheds light on the intermediary process.[4].

Current e-commerce companies makes use of recommender systems to improve user experience. Recommender system(s) maps the possible interest of a user to what the company provides. In mathematical terms, two matrices can be formed , vector operation can be performed with a regularization term [19]. The quality of these predictions are in constant flux due to the inconsistent nature of data collection. To combat this problem commanders are consisted of various algorithms combined to create accurate suggestions to users, normally working on historical data. Thus the optimal point suggestions is only based on the data at a certain time (t_0) as the data changes over time (Δt), a different algorithm is needed at that point.

During the evaluation period - the time period before a recommender is put into production, predictions are evaluated, if an acceptable accuracy rate is obtained the model is then put into production, however, these model can at times of bias. If, during the evaluation period, a company runs a promotion, a point might be considered correct (True Positive) due to that very fact, and it does not reflect actual interests of the user at a given time - as referred to in 5.

Different evaluation methods are more optimally situated to the task: take for instance 2 that is better suited to to regression tasks versus *cross - validation* for classification tasks. This can be elaborated on, for instance when looking at regression: 3 is a more appropriate metric when compared to 2 when data that is being utilized has many outliers - as 3 deals better with outliers.[9].

Evaluation Bias : *Evaluation Bias* occurs when when performance metrics that are used is not aptly suited to the task at hand; this can lead to external tests does not represented the population accurately. Consider the psychological phenomena of *Illusory correlation*; Illusory correlation occurs when we draw the conclusion that two instances are related when in fact they are far from being correlated - machines suffer from the same illusion - resulting that the infrequent class of the minority class is overestimated.[11] Resulting to the inflation of *false positive* scores and a decrease in *false negative* score; leading to poorer model performance. This is well illustrated in a study by a Cunningham and Delany; where they took to summary statistics from the *Adult and Recidivism datasets* - in this dataset, certain population groups are underrepresented in the actual data used to train the model, causing inflation of what could be deemed a negative attribute. This is a instance of *Negative Legacy*: in negative legacy there is bias in the data that is being dealt with due to instances of incorrect labelling of data; past discriminatory practices and poor data sources[5]. These are just some of the instances that can lead to poor model performance as mentioned in section(3).

6.3 Adjusting Weights

Human decisions patters are very capricious - this can at times be beneficial or detrimental, but this capricious nature allows the human brain to perform rather complex decision tasks. [7]. Neural networks work on the principal of considering each feature of a node; features are the proverbial factors that are taken into account when making a decision, for instance: you will put higher value on a 'node' of hunger to control your emotions as compared to when you are hungry. These are essentially how decisions are made.[16] This process of *Adjusting weights* has been proven to assist in various decision problems such as

rules-based and case-based systems.Adjusting weight w associated to a node has shown to be able combat evaluation bias; among various other problems, and the creation and training of models.

7 Conclusion

Irregularities and discrepancies in models is something that will be encountered throughout the life cycle of models. There has in, the past, been 2 'AI winters' - periods of reduced funding and interest: from 1974-1980; due to non delivery of exaggerated capabilities, and from 1987-1993; unreliable databases and wavering confidence in *expert-systems.*[9]. To ensure the longevity of AI these *nondeterministic* systems must be understood.

All of the above mentioned topics are attempts to understand bias altruistically. Adjusting weights(section6.3) poses a great potential to both identify and mitigate biases, especially in iterative tasks - as the network has time analyse outcomes.
Adjusting weights has also shown to reduce imbalance bias mentioned in the use case(6.1). Adjusting weight can also address to problem faced such as the *PdM recommender*, where adjusting weights can represent various equations (n_i) over various parameter ($w_1, ..., w_i$), relating to each in each other in every time instance[6].
This could also assist with with the use case(6.2), here negative legacy(4) and also coniders multiple outcomes which could assist in obtaining more insightfull scores in the Calder Verwer score(4), discussed previously, this could work by introducing various sensitive features and measure model performance based on the introduced feature.
furthermore, adjusting weights has also been shown to assist with problems of overfitting that commonly plagues model creation[5].
Adjusting weights could also assist in cases such as choosing appropriate evaulation metrics such as *LogLoss(4.2) and cross-validation(4.2)* by pinning different evaluation methods against each other; as for instance with LogLoss, which handles high deviating datasets well but will not necessarily be optimal in all cases, adjusting weights could assist in making sure the appropriate evaluation method is indeed being utilised.

Artificial intelligence and its derivatives is shaping the way we work and interact, if a technology is impacting so many of our daily activities in is important to create a culture of asking the right questions when represented with information; always being vigilant and understanding of the mechanisms of action can only serve to benefit the successfully adoption of new technologies - thus as with most unknown topics, they must first be explored.

References

[1] The importance of complexity in model selection. *Journal of Mathematical Psychology 44*.

[2] 2007.

[3] Ebru Turanoglu Bekar, Per Nyqvist, and Anders Skoogh. An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study. *Advances in Mechanical Engineering (Sage Publications Inc.)*, 12(5):1, 2020.

[4] Pascal Bornet and Jochen Wirtz. *Intelligent Automation - Learn How to Harness Artificial Intelligence to Boost Business Make Our World More Human*. 10 2020.

[5] Padraig Cunningham and Sarah Jane Delany. Algorithmic bias and regularisation in machine learning. 2020.

[6] Arnaud De Myttenaere, Bénédicte Le Grand, Boris Golden, and Fabrice Rossi. Reducing offline evaluation bias in recommendation systems. 2014.

[7] Norma Doidge. *The Brain that Changes Itself*. Penguin Books, 2007.

[8] Jindong Gu and Daniela Oelke. Understanding bias in machine learning. 2019.

[9] Prof.Dr. Ulrich Kerzel. *Artificial Intelligence*. IUBH Internationale Hochschule GmbH, 2020.

[10] Prof.Dr. Ulrich Kerzel. *Use Case and Evaluation*. IUBH Internationale Hochschule GmbH, 2020.

[11] The Decision Lab. Why do we think some things are related when they aren't?

[12] Brett Lanz. Medical cost personal datasets, 2020.

[13] J. Maeda. *How to Speak Machine: Laws of Design for a Computational Age*. Penguin Publishing Group, 2019.

[14] Ayilara Olawale F., Zhang Lisa, Sajobi Tolulope T., Sawatzky Richard, Bohm Eric, and Lix Lisa M. Impact of missing data on bias and precision when estimating change in patient-reported outcomes from a clinical registry. 2019.

[15] M. Parker. *Humble Pi: A Comedy of Maths Errors*. Penguin Books, Limited, 2019.

[16] Jordan B. Peterson. *Maps of meaning: Architecture of Meaning*. Harvard 18th ed., 1999.

[17] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *CoRR*, abs/1811.12808, 2018.

[18] Francesca Rossi. How ibm is working toward a fairer ai. *Harvard Business Review Digital Articles*, pages 2 – 5, 2020.

[19] Guarav Rajesh Sahani. *Math behind Content Based Recommendation System*, 2020 (accessed October 30, 2020).

[20] Seema Singh. Understanding the bias-variance trade off, 2018.

[21] Alberto Cairo [dtc] Justin Matejka [dtc] George Fitzmaurice [dtc] Lucy D'Agostino McGowan [aut] Richard Cotton [ctb] Locke Data [fnd] Steph Locke [cre, aut]. datasaurus: Datasets from the datasaurus dozen.

[22] Harini Suresh and John V. Guttag. A framework for understanding unintended consequences of machine learning. *CoRR*, abs/1901.10002, 2019.

[23] S. van Buuren. *Flexible Imputation of Missing Data*. Chapman & Hall/CRC Interdisciplinary Statistics. CRC Press LLC, 2018.

[24] Josie Williams and Narges Razavian. Towards quantification of bias in machine learning for healthcare: A case study of renal failure prediction. 2019.

A Regression Model

Listing 1: Logistic Regression Prediction

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.metrics import mean_absolute_error

insurance = pd.read_csv(r'Medical_Insurance_Dataset.csv')

print(insurance)

df_insurance = insurance[['age', 'charges']]
print(df_insurance)

X = df_insurance['age'].values
y = df_insurance['charges'].values

y = y.reshape(-1, 1)
X = X.reshape(-1, 1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state

reg = LinearRegression()
reg.fit(X_train, y_train)

y_pred = reg.predict(X_test)

print(reg.score(X_test, y_test))
rmse = mean_absolute_error(y_test, y_pred)
print(rmse)

plt.plot(X_test, y_pred, linestyle = '--', marker = 'o', color = 'b')

plt.grid()
plt.ylabel("Charges in $")
plt.title("Prediction of insurance payment in age categories")
plt.xlabel("Age of individuals")
```

```
plt.show()
```

B Iris Classification

Listing 2: Iris Classification[2]

```
import numpy as np
import matplotlib.pyplot as plt

from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, plot_tree

# Parameters
n_classes = 3
plot_colors = "ryb"
plot_step = 0.02

# Load data
iris = load_iris()

for pairidx, pair in enumerate([[0, 1], [0, 2], [0, 3],
                                [1, 2], [1, 3], [2, 3]]):
    # We only take the two corresponding features
    X = iris.data[:, pair]
    y = iris.target

    # Train
    clf = DecisionTreeClassifier().fit(X, y)

    # Plot the decision boundary
    plt.subplot(2, 3, pairidx + 1)

    x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, plot_step),
                         np.arange(y_min, y_max, plot_step))
    plt.tight_layout(h_pad=0.5, w_pad=0.5, pad=2.5)

    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    cs = plt.contourf(xx, yy, Z, cmap=plt.cm.RdYlBu)
```

```
    plt.xlabel(iris.feature_names[pair[0]])
    plt.ylabel(iris.feature_names[pair[1]])

    # Plot the training points
    for i, color in zip(range(n_classes), plot_colors):
        idx = np.where(y == i)
        plt.scatter(X[idx, 0], X[idx, 1], c=color, label=iris.target_names[i],
                    cmap=plt.cm.RdYlBu, edgecolor='black', s=15)

plt.suptitle("Decision surface of a decision tree using paired features")
plt.legend(loc='lower right', borderpad=0, handletextpad=0)
plt.axis("tight")

plt.figure()
clf = DecisionTreeClassifier().fit(iris.data, iris.target)
plot_tree(clf, filled=True)
plt.show()
```