Exploring methods to identify proxy variables

Grant Lee (grant_lee@brown.edu)
Alexander Mathew (alexander_mathew@brown.edu)

Abstract

Proxies are features of a dataset that can serve as indicators or substitutes for removed demographic attributes and can contribute to algorithmic discrimination within a machine learning model. In this project, we demonstrate four methods to identify proxies of demographic features: Pearson's correlation, pairwise Cramer's V, feature combinations, and feature redundancy (using FACET). Pearson's correlation coefficient is suitable in identifying linear relationships between continuous variables, while Cramer's V is more appropriate for evaluating relationships between categorical variables. However, both methods are limited in that they only analyze pairwise relationships and do not take into context the contribution of each feature on the model output. To account for the former, we demonstrate how combinations of features could also function as proxies, especially when used in non-linear models. To maximize effectiveness, we propose a sequence of a correlation-based method, followed by an evaluation of feature combinations. To account for the latter, we also demonstrated a simplified evaluation of feature redundancy using the FACET library, which generates a redundancy matrix using SHAP values. While these four methods together serve as a practical tool for proxy detection and feature evaluation, this project contributes to the larger goal of ensuring that AI-based systems adhere to the Algorithmic Discrimination Protections principle of the AI Bill of Rights, promoting fairness and mitigating discrimination in automated decision making systems.

Problem statement

For our project, we chose to focus on the Algorithmic Discrimination Protections principle from the AI Bill of Rights[1], which suggests that users of AI-based systems "should not face discrimination by algorithms" and that AI-based systems "should be used and designed in an equitable way". One of the expectations within this principle is that AI-based systems should guard against proxies.

What is a proxy? When training a machine learning model, it is often expected that demographic information should be removed from the dataset as a preprocessing step, as they are known to effectively introduce algorithmic discrimination to the model. However, this is not always sufficient. Other features can, individually or in combination with other features, still act as

[1] AI Bill of Rights - Algorithmic Discrimination Protections, accessible at https://www.whitehouse.gov/ostp/ai-bill-of-rights/algorithmic-discrimination-protections-2/

proxies for the removed demographic attribute(s). For example, a person's ZIP code has often been found to be a proxy for their race.[2]

The AI Bill of Rights advocates for proactive testing to identify such proxies. This can be done by testing for correlation between demographic features and other attributes (or combinations of these attributes). Our proposed project aims to demonstrate methods to perform a test of feature correlation on a dataset in order to identify such proxies.

If a proxy has been identified, it is often preferable to remove it as a feature in the dataset. However, it is probable that the proxy feature contributes to the performance of the model. If removing the identified proxy significantly reduces the accuracy of the model, we might want to retain the proxy in the model, but assign the feature less weight. Our implementation should hence also consider how each feature affects model performance. The result of this project is a tool that identifies proxies and considers the importance of the feature before advising whether or not to remove it from the dataset.

Data sets and/or algorithms

Our Python notebook is hosted on Google Colab, and can be accessed via this link.

For our experimental analyses, we used the `folktables` Python package[3] to access the American Community Survey datasets. In particular, we created a custom task, which uses the 2021 dataset containing information about individuals in California:

- `survey_year='2021'`
- `horizon='1-Year'`
- `survey='person'`
- `states=["CA"]`

Our custom task is defined as a regression problem. The target variable (i.e., the variable we are trying to predict) is the total personal income of an individual, coded as `PINCP`. Referring to the ACS PUMS Data Dictionary documentation[4], `PINCP` is a numeric variable that can take on the following values:

- `NaN`: N/A (less than 15 years old)
- `None` to `-19998`: Loss of $19,998 or more
- `-19997` to `-1`: Loss $1 to $19,997
- `1` to `4209995`: $1 to $4,209,995

[2] For more information about features often used as proxies for race, refer to Chapter 4 of the book "Calculating Race: Racial Discrimination in Risk Assessment" by Benjamin Wiggins (published November 2020)

[3] GitHub repository for `folktables` package, accessible at https://github.com/socialfoundations/folktables

[4] ACS PUMS Data Dictionary documentation, accessible at https://www2.census.gov/programs-surveys/acs/tech_docs/pums/data_dict/PUMS_Data_Dictionary_2021.pdf

Our group variable is the race, coded as RAC1P, which is a categorical variable that can take on the following values:

- 1: White alone
- 2: Black or African American alone
- 3: American Indian alone
- 4: Alaska Native alone
- 5: American Indian and Alaska Native tribes specified, or American Indian or Alaska Native, not specified and no other races
- 6: Asian alone
- 7: Native Hawaiian and other Pacific Islander alone
- 8: Some other race alone
- 9: Two or more races

After studying the ACS PUMS Data Dictionary documentation, we identified 11 other features that we felt could potentially act as proxies for race. These features are described in the following table. The respective values for each feature can be found in the ACS PUMS Data Dictionary.

Code	**Feature**	**Type**
COW	Class of worker	Categorical
SCHL	Educational attainment	Categorical
MAR	Marital status	Categorical
OCCP	Occupation	Categorical
POBP	Place of birth	Categorical
RELSHIPP	Relationship	Categorical
WKHP	Hours worked per week	Numeric (continuous)
SEX	Gender	Categorical
JWTRNS	Mode of transport to work	Categorical
LANP	Language spoken at home besides English	Categorical
LANX	Whether another language is spoken at home besides English	Categorical (binary)

Table 1: Features included in the dataset

Given the above specifications, we then applied the `adult_filter` preprocessing step as provided by the `folktables` dataset, which retains only data for individuals aged above 16. Our resulting dataset contains 192,223 rows of data.

Methods

In our project, we demonstrated four methods to perform proxy identification:

1. Pearson linear correlation
2. Pairwise Cramer's V
3. Feature combinations
4. Feature redundancy

Method 1: Pearson linear correlation

Pearson correlation is probably the most widely used statistical measure to assess the correlation between two variables. A high correlation coefficient between two variables is often associated with the possibility that one could be a proxy of the other. In our experiments, we implemented this using the `pandas` library's handy `corr()` function, which computes Pearson correlation coefficient across the dataframe. We also set a threshold of 0.4, beyond which we consider the variable to be a proxy to be removed from the model. After removing it, we then ran the linear regression model and studied if the removal of the feature results in a significant reduction in model performance.

Method 2: Pairwise Cramer's V

One assumption that Pearson correlation makes is that the relationship between the proxy and the protected attribute is linear. This makes it an appropriate measure of correlation for continuous variables (such as `WKHP`), but might not be apt for categorical features. To account for this, the second method we demonstrated is pairwise Cramer's V.

Cramer's V is essentially a scaled version of the chi-squared test statistic that acts as a suitable measure of association between categorical variables. By doing this, we can uncover any non-linear relationships that might indicate the existence of a proxy. This was particularly useful as most of our variables were categorical, rather than continuous. Our implementation uses the `cramers_v()` function in the `dython` package[5]. Similarly, we set a threshold of 0.4, above which the variable would be identified as a proxy and we would then remove it from the model and evaluate its impact on the model performance.

Method 3: Feature combinations

Both of the above-mentioned methods also do assume that the proxy is a single variable. It could be possible instead that the proxy is a combination of features, even when each individual feature has a weak correlation to the protected attribute (race).

[5] Documentation for `dython` Python package, accessible at https://shakedzy.xyz/dython/

In order to test for feature combinations, we selected the three variables with the highest correlation coefficients in our Cramer's V test. Given three features *A*, *B* and *C*, we generate seven possible feature combinations: *{A}*, *{B}*, *{C}*, *{A, B}*, *{A, C}*, *{B, C}* and *{A, B, C}*.

For each combination, we drop all other columns except the ones in the combination set, and run them through two separate models to predict the sensitive group variable (note: *not* the target variable). One of these models should be a linear or logistic regression model, and the other a decision-tree model. This allows us to capture both linear and non-linear relationships between each feature combination and the group variable (race).

For both of these models, a higher accuracy score would then indicate that the particular combination of features is relatively better at predicting the sensitive group variable (race), and is thus more likely to be a proxy for race.

Method 4: Feature redundancy

A limitation of all three methods mentioned above is that they are focused only on studying the relationship between the protected attribute and other features, and do not take into account the model that is trained on the features. In reality, we should consider how the inclusion of these features (including the proxy) affects the output of the model, rather than the inter-variable relationships.

To do this, we want to study feature redundancy, which refers to the degree to which a feature in a model duplicates the information of a second feature to predict the target. To test for redundancy, we use FACET[6], an open source library for AI model explainability developed by Boston Consulting Group's GAMMA team.

This library understands model predictions using SHAP values, then further breaks down feature interactions into three metrics: redundancy, synergy and independence. In particular, to generate the redundancy score, FACET does so by combining the computed SHAP values and pairwise SHAP interaction values (across every provided data point) as SHAP vectors, then decomposes these vectors to obtain a global measurement of redundancy between each pair of features.

Redundancy is a metric particularly useful for identifying proxies. Unlike correlation, redundancy will capture the non-linear effects between features if a non-linear model is used. By looking at the redundancy between the protected attribute and other features, we were able to measure how much duplicate information is encoded between the protected attribute and another feature within a specific model. We then generate a redundancy matrix to visualize the feature redundancy scores between each pair of features in the dataset. However, one thing to note is that running FACET is rather computationally heavy, especially for a large dataset. Unfortunately, with the limited compute resources that we had, we were only able to run it on 1,000 rows of data (i.e., a mere 0.5% of our dataset).

[6] BCG GAMMA FACET library, accessible at https://bcg-gamma.github.io/facet/

Analysis/discussion

Method 1: Pearson linear correlation

Although Pearson correlation is the simplest to set up and most popular method for proxy variable detection, it makes two assumptions which make it the most naive of the four methods:

1. One is that the proxy is a single other variable. This method only performs a calculation of correlation between single variables, whereas there is a high possibility that the proxy is actually a combination of these features.

2. The second assumption is that the relationship between the proxy and the protected feature is linear. This method would be suitable for continuous (numeric) variables which often share a linear relationship. However, as seen in Table 1, our features are mostly categorical ones, so this method is less appropriate.

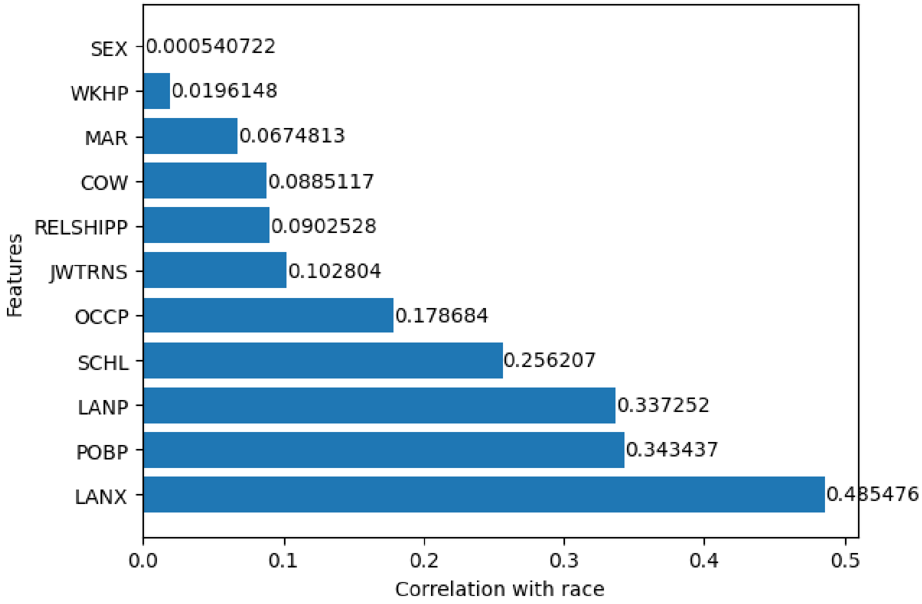


Figure 1: Correlation coefficients using Pearson linear correlation

Figure 1 above shows the Pearson correlation coefficients for each feature. The only feature which had a correlation coefficient that exceeded the threshold (of 0.4) was `LANX` (whether a non-English language is spoken at home), with a coefficient of 0.485476.

The linear regression model accuracy falls from 0.24989 (with all features) to 0.24958 (with `LANX` removed). We regard this difference of 0.00031 to be insignificant and indicative that removing the proxy does not have a significant impact on model performance.

Method 2: Pairwise Cramer's V

Using Cramer's V to identify proxies eliminates the second assumption as it provides a better representation of correlation between categorical variables. Our results are depicted in Figure 2.

Using the same 0.4 threshold, our identified proxies were:

1. `POBP` (place of birth), with a coefficient of 0.401955
2. `LANP` (language spoken at home), with a coefficient of 0.414804
3. `LANX` (whether a non-English language is spoken at home), with a coefficient of 0.557253

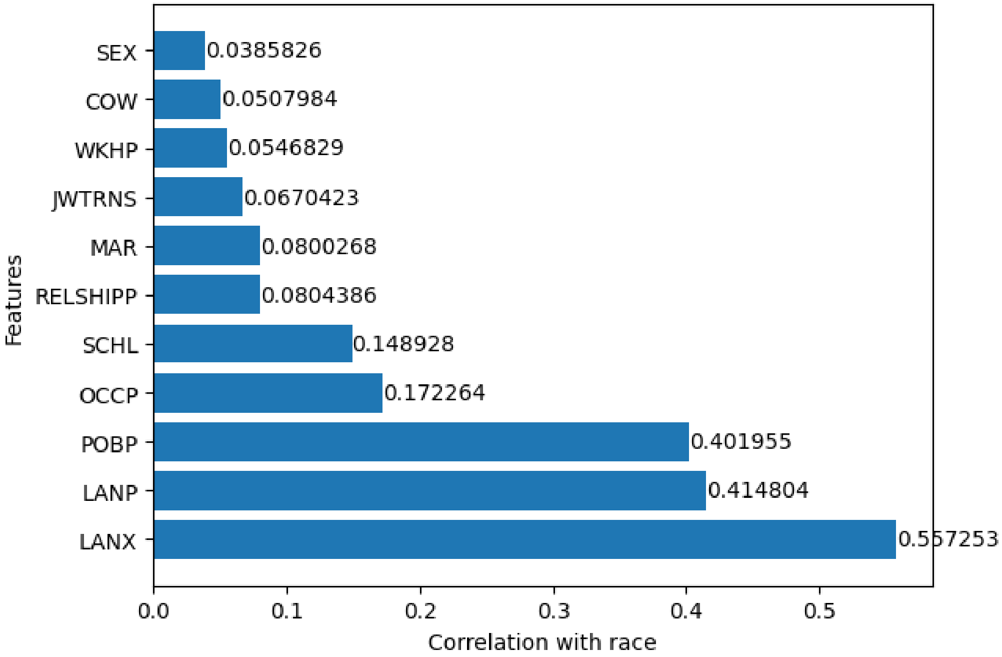


Figure 2: Correlation coefficients using Cramer's V

The linear regression model accuracy falls from 0.24989 (with all features) to 0.24877 (with all three variables removed). The small difference of 0.00112 suggests that removing the proxies does not have a significant impact on model performance.

Comparing the correlation coefficients for Pearson correlation (in Figure 1) and Cramer's V (in Figure 2), we can see that most of the correlations were higher using Cramer's V. This could be attributable to the fact that Cramer's V is more well-suited at detecting correlations between categorical variables.

A useful Medium article[7] provides a rather in-depth analysis with regards to which statistical methods to use to measure correlations between categorical and continuous variables. In essence, the author suggests that:

1. Between two continuous variables, if we expect the relationship between them to be linear, Pearson's correlation would be a suitable measure. Otherwise, we can approach the problem by applying ordinal or rank-based correlation approaches instead. To achieve this, consider the following methods: Spearman correlation, Goodman and Kruskal's gamma, Kendall's tau and Somers' D.

2. Between two categorical variables, there are two sets of approaches to calculate their correlation. The first is to use distance metrics (e.g. Euclidean distance, sum of squared distance) to compute the similarity between vectors, which is conceptually similar to other measures of correlation. The second is to use a correlation coefficient based on the chi-squared distribution. Some examples of this include Cramer's V, Phi coefficient and Tschuprow's T.

3. Between a continuous variable and a categorical variable, we can instead approach the calculation of correlation either using logistic regression or using point biserial correlation coefficient, which is a special case of Pearson's correlation coefficient.

Method 3: Feature combinations

Using the three features with the highest correlation coefficients in the Cramer's V test (`POBP`, `LANX` and `LANP`), we generated seven combinations of features. For each of these seven combinations, we dropped all columns except those in the combination and ran the new augmented dataset on both a linear regression model and a decision tree model to predict `RAC1P` (the sensitive group variable). The results of this analysis are shown in Table 2 below:

Feature combination	Accuracy, linear regression	Accuracy, decision tree
['LANX']	0.23583	0.48552
['LANP']	0.11240	0.65717
['POBP']	0.11781	0.62013
['LANX', 'LANP']	0.25291	0.65717
['LANX', 'POBP']	0.24352	0.65356

[7] An overview of correlation measures between categorical and continuous variables, a Medium article by "Outside Two Standard Deviations", accessible at https://medium.com/@outside2SDs/an-overview-of-correlation-measures-between-categorical-and-continuous-variables-4c7f85610365

['LANP', 'POBP']	0.15185	0.68121
['LANX', 'POBP', 'LANP']	0.26346	0.68123

Table 2: Effectiveness of feature combinations on predicting sensitive group variable

The results above give us some interesting takeaways:

1. The linear regression model agrees with our first two methods, highlighting LANX as the single feature most likely to be a proxy for race. When combined with LANP and POBP, there is only a minor increase in model accuracy, possibly suggesting that none of the feature combinations are significant enough to act as a proxy than LANX as a singular feature.

2. Across the board, the decision tree model performed better than the linear regression model at predicting the RAC1P attribute. This is indicative that the three features are likely to share a non-linear relationship with the race feature.

3. The decision tree model performed in a manner that suggested the very opposite pattern of the linear regression model. Looking at the accuracy scores, LANP (instead of LANX) seems to be the most accurate singular feature, possibly indicating that it is the best proxy of race. Even in combination with LANX and POBP, model accuracy does not increase significantly (especially with the inclusion of LANX).

4. Running both ['LANP'] and ['LANX', 'LANP'] through the decision tree model produces the exact same accuracy score. This is expected, since LANX is essentially a binary encoding of LANP, i.e. LANX is a classification of whether the value of LANP is NaN. Since LANP already inherently contains the information encoded in LANX, the inclusion of LANX alongside LANP in the feature combination does not improve the accuracy of the decision tree model.

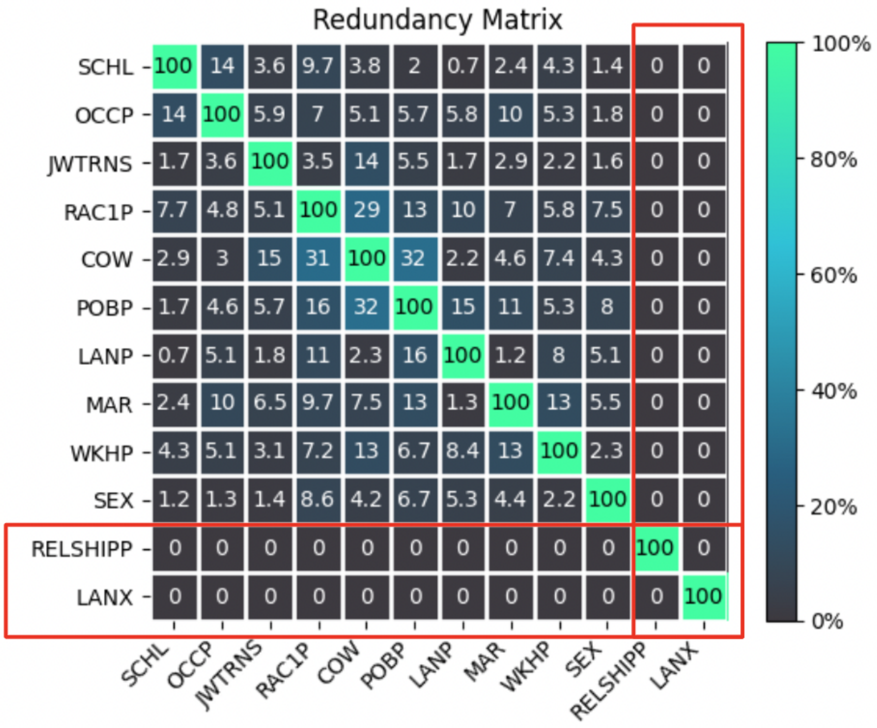
It is also crucial to note that in this case, we have selected these three features based on pairwise correlation coefficients as calculated using Cramer's V. In practice, this method may require the data scientist to possess business knowledge and context to determine the exact, proper set of features to evaluate. Many complex models use hundreds of features, making it a highly difficult and impractical task to test all possible combinations of features.

To this end, we propose an approach that combines all three methods into a rational sequence. First, perform either Pearson linear correlation, Cramer's V or other metrics discussed in earlier sections to calculate the pairwise correlations across all the features in the dataset. We can then use the results of these evaluations to guide the manual selection of features to be included in the construction of feature combinations, similar to our methodology above, and run the analysis as described above.

An additional challenge is that even if the data scientist discovers that a specific feature combination leads to unfair outcomes, it is not immediately clear how to move forward. A great deal of follow-up (contextual) evaluation is required to determine whether it is advisable to remove all or only some of the features in the combination, as well as to evaluate the extent to which the removal of these features will affect the performance of the model.

Method 4: Feature redundancy

Using FACET to generate the redundancy matrix for our dataset of 192,223 rows, we discovered a major issue: running FACET on these many rows was a computationally heavy process, and our limited compute resources only allowed us to successfully generate a matrix if we ran it on the first 1,000 rows in the dataset. Note that this is a mere 0.5% of the total number of rows in the dataset, and hence the resulting matrix and the values embedded in it are *far from accurate*. Nevertheless, the generated redundancy matrix is shown below in Figure 3.



	SCHL	OCCP	JWTRNS	RAC1P	COW	POBP	LANP	MAR	WKHP	SEX	RELSHIPP	LANX
SCHL	100	14	3.6	9.7	3.8	2	0.7	2.4	4.3	1.4	0	0
OCCP	14	100	5.9	7	5.1	5.7	5.8	10	5.3	1.8	0	0
JWTRNS	1.7	3.6	100	3.5	14	5.5	1.7	2.9	2.2	1.6	0	0
RAC1P	7.7	4.8	5.1	100	29	13	10	7	5.8	7.5	0	0
COW	2.9	3	15	31	100	32	2.2	4.6	7.4	4.3	0	0
POBP	1.7	4.6	5.7	16	32	100	15	11	5.3	8	0	0
LANP	0.7	5.1	1.8	11	2.3	16	100	1.2	8	5.1	0	0
MAR	2.4	10	6.5	9.7	7.5	13	1.3	100	13	5.5	0	0
WKHP	4.3	5.1	3.1	7.2	13	6.7	8.4	13	100	2.3	0	0
SEX	1.2	1.3	1.4	8.6	4.2	6.7	5.3	4.4	2.2	100	0	0
RELSHIPP	0	0	0	0	0	0	0	0	0	0	100	0
LANX	0	0	0	0	0	0	0	0	0	0	0	100

Figure 3: Redundancy matrix generated by FACET on first 1,000 rows of dataset

Even though we were unable to successfully generate an accurate matrix for all 192,223 rows of data, the example of a redundancy matrix in Figure 3 allows us to better understand how we could use information in the matrix to decide whether a feature could be considered as a proxy or not. Note that a higher redundancy score suggests that there is relatively more duplicate information encoded between two features. Looking at the `RAC1P` row, we see that (based on the example redundancy matrix) `COW` has the highest redundancy score of 29%, suggesting that it is the most likely candidate as a proxy for race.

In our demonstration, a random forest regressor was our choice of model, but a similar matrix can be generated with other machine learning models. Doing so will generate different redundancy scores, since redundancy is evaluated in the context of the model and how each feature affects the model output. Additionally, the FACET library also provides useful tools to perform grid search to tune hyperparameters.

One glaring shortcoming of FACET is that it performs only pairwise evaluations. As such, FACET presents itself as more of an alternative to pairwise correlations (methods 1 and 2), and less so of methods that consider a combination of features as a proxy (method 3).

Implications

We have provided a summary and exemplification of four different approaches that data scientists can employ to identify proxies of sensitive attributes in their datasets.

In particular, the first two methods (Pearson linear correlation and pairwise Cramer's V) are based on the identification of correlations between pairs of features. These methods are model agnostic, and seek to identify a relationship solely based on the statistical distributions of the values for each feature.

The next two methods (feature combinations and feature redundancy) perform evaluation while taking into context the model used – in two different ways. When identifying feature combinations as potential proxies, we are interested in using different models to capture the relationship between the potential features and the sensitive group variable (not the target variable). On the other hand, when evaluating feature redundancy, we are interested in how each feature produces duplicate information that is already encoded in the sensitive group variable and revealed by the model.

We hope that data scientists can employ these methods to guard against dangerous proxies in their datasets and machine learning models. That being said, we have a few additional learning points that we believe are important to note about the process of proxy identification.

Firstly, as much as these methods provide an algorithmic manner of performing proxy identification, the most important step is still manual, that is, the selection of the initial 11 features to be evaluated. In our study, intentional human judgment was used in selecting the list of features as laid out in Table 1. Of course, wherever there is human judgment, there is also potential for human bias. It is important that the humans who decide on the subset of features to evaluate are also organized and directed in a manner that mitigates the potential for human bias (personal, organizational, etc.) to bleed into the very process that aims to reduce bias.

Many steps in these methods also require human judgment. For example, the setting of a correlation threshold requires human judgment. The evaluation of an "acceptable" reduction in model performance (due to the removal of a feature) also requires human judgment. Ultimately, we feel that the decision of whether or not to remove a feature should be made in a way that considers not only the numbers generated by these methods (coefficients, accuracy scores, redundancy scores, etc.), but also the contextual meaning of features in relation to the task at hand.

In our study, we are interested in predicting a person's personal income. We need to dig deeper. Why are we interested in doing so? Are we trying to create an ad campaign targeted at people above a certain income? Are we trying to build a campaign to raise awareness about income inequality? Depending on the intentions of the project, different features can reveal themselves to be more or less appropriate as proxies of a given sensitive demographic attribute. A holistic understanding of the societal context of the problem will therefore allow us to make better (human) judgments about whether an identified proxy should be eventually removed or not.

Finally, proxies are not all bad. It is even written in the AI Bill of Rights that "if needed, it may be possible to identify alternative attributes that can be used instead". This is especially appropriate if we identify a proxy variable that contributes significantly to the model output, in which case removing it would heavily impact the model performance. In these cases, we can consider novel ways to actually build proxies that are less discriminatory than the specific features themselves.

One example of this is the Bayesian Improved Surname Geocoding (BISG) proxy method, developed by the Consumer Financial Protection Bureau (CFPB)[8]. The BISG method is built upon work done by Elliot, et al. (2009)[9] that proposes a method that combines publicly available demographic information associated with surname and residential geographic areas to generate a single proxy for race and ethnicity. The result is a proxy that is not only more accurate than those based on surname or geography alone, but also ensures that lenders are complying with fair lending laws and addressing discrimination across the consumer credit industry.

Personal reflections

Grant Lee's reflections

The main challenge I faced while working on the project was in understanding how to apply each method to an actual dataset. While I found many resources discussing the theoretical functionalities and mathematics behind correlation coefficients, chi-squared statistics, SHAP values and feature redundancy, actually having to put them into practice demanded a disciplined approach. I found myself reading pages of documentation for FACET, trying not only to understand how it converts SHAP values into redundancy scores, but also how I could utilize the functionalities provided by the library. Nonetheless, it was a rewarding experience to be able to put those skills into practice on a dataset of our choice.

[8] Consumer Financial Protection Bureau (2014). Using publicly available information to proxy for unidentified race and ethnicity - a methodology and assessment. Accessible at https://files.consumerfinance.gov/f/201409_cfpb_report_proxy-methodology.pdf

[9] Elliott, M. N., Morrison, P. A., Fremont, A., McCaffrey, D. F., Pantoja, P., & Lurie, N. (2009). Using the Census Bureau's surname list to improve estimates of race/ethnicity and associated disparities. *Health Services and Outcomes Research Methodology*, *9*, 69-83.

Another challenge I faced was in understanding why the correlation coefficients and accuracy scores we received from the models were as they were. Ideally, we should have included an additional step in our methodologies – to plot and visualize the data, so that we could better understand the pairwise relationships between the features. Eventually, we made do by coming up with rational explanations for the correlation coefficients and model accuracy scores in a comparative (rather than absolute) manner.

Finally, our biggest challenge was unfortunately one that we were not able to resolve by the end of the project. That is, we did not have sufficient compute resources to generate the FACET redundancy matrix on all 192,223 rows of data. Instead, our resources only allowed us to run it on a mere 0.5% of the dataset, and the resulting redundancy was one that contained scores that were far from accurate. Had we been able to successfully generate it on a larger portion of the dataset, we would have been able to come up with more analytical and interesting evaluation points about the fourth method. This was quite the bummer!

Nevertheless, working on the project was a refreshing experience for me. Although fairness and explanations in automated decision making systems are a relatively novel idea, this project has exposed me to a number of existing, practical tools and methodologies that can be used in the implementation of such systems. With all the hoo-ha around generative AI, I find it heartening to see that – through the building of these fairness and explanation tools – the data science community is also moving in the direction of making sure that these systems are built in a fair manner.

Just as importantly, working on the project has also made me realize that the act of implementing fairness in automated decision making systems is an intentional one, and one that involves the input and intellect of humans. What this means is that the socio-political context of the system and the environment in which it is deployed are crucial factors that can determine the definition of “fairness” in that very system. Hence, it is of utmost importance that there is governance and academic discourse that surround such a definition and the creation of standards, rules and guidelines, such as the AI Bill of Rights.

Alexander Mathew’s reflections

During my work on the project, I encountered several challenges that influenced my understanding of fairness and explanation for automated decision systems. Firstly, I struggled with determining the appropriate threshold a correlation value a feature should have in order to be flagged as a proxy. This required extensive research to gain a better understanding of proxies and their impact on the model's biases. Some of our correlations while testing were not necessarily very high, but still significantly higher than the others so it was a challenge to understand if that was a proxy or not. Another significant challenge I faced was the computational complexity of working with FACET. The computational expenses associated with running the model were too much for both mine and Grant’s systems, so we decided to run FACET on the first 1000 rows, the maximum our systems would allow. Although this did not give us a fully complete redundancy matrix, it was still useful for us to see what the output would have been. This challenge highlighted the importance of scalable and efficient approaches to ensure fairness in real-world scenarios.

Working on this project has very much influenced my perspective on fairness and explanation for automated decision systems. One realization I made was that proxies are not limited to individual features but often emerge from the combination of multiple features. It could be any number of features put together that act as a proxy. This becomes a large issue as the complexity of real-world models and datasets continues to grow. Moreover, this project has given me insight on the transparency surrounding automated decision systems. While it is expected that organizations test thoroughly for proxies and ensure fairness, the actual methods and criteria used for such assessments often remain undisclosed. This lack of transparency leaves many individuals unaware of the factors influencing automated decisions that directly impact their lives. It has become evident that establishing clearer guidelines and standards for fairness assessments, as well as promoting transparency in the decision-making process, are crucial steps toward addressing these issues. Furthermore, as this field as a whole is still relatively young, there is lots of work left to do and many companies may still be working on integrating fair practices in their decision making systems. In conclusion, working on the project has helped shape my perspective on fairness and explanation in automated decision systems. Understanding the intricacies of proxies, the need for extensive testing, and the importance of transparency from companies has deepened my awareness of the topic. I have learned that the field of fairness in AI is still a challenging topic and much work is still left to do in order to fully grasp all the concepts. Through encountering lots of challenges I have gotten experience dealing with issues prevalent in the field. I hope to continue to use the knowledge and skills learned from this project as well as from the class in my future career to develop fair and equitable automated decision making systems.