Enforced Fairness for Machine Learning in Insurance, Criminal Justice, and Other Sensitive Environments

CPSC 471 Project Proposal

Tristan Brigham

February 2024

Problem Definition: Observable AI in Sensitive Environments

The tradeoff between explainability and efficiency in current machine learning models is extreme. I seek to provide a framework and example implementations of how fairness and explainability can be directly integrated into the training process for machine learning model as well as post-hoc explanations of outputs.

The challenge of crafting machine learning models that are efficient and explainable in highly sensitive industries such as healthcare and insurance is a pertinent and highly investigated topic today. Machine learning has the opportunity to drastically transform the efficiency and workflows of such industries much like every other line of work that it has touched so far, but the sensitivity of the topics being investigated and manipulated in insurance and health necessitate more explainable yet efficient models before such models can be initiated in related institutions. Simply put, when livelihoods are at risk, the benefits that can be gotten from non-linear decision boundaries found within neural networks do not outweigh the potential drawbacks that can exist in issues such as transparency, fairness, and trust.

One of the main problems that currently exist is that data is biased. I can solve this problem by performing analysis of linear-decision-boundary models such as regressions. However, this constrains us to problems where data exhibits linear decision boundaries, or where I are able to transform the data in an appropriate timeframe to create such linearity in the decision boundaries. This is often not the case.

The problem of non-explainable machine learning models being used in such sensitive applications presents the potential for highly unfair or biased machine learning models to propagate inequities and injustices in society. If I deploy current machine learning models, I are forced by definition to use historical data. This data can often exhibit societal biases that I are trying to get away from. But, without explainable models I cannot understand whether these biases are playing a role in the network decision outputs.

And, as I continue to put more trust in the statistical models because of the fallacious belief that such algorithms and mathematical processes equates with justice and fairness, I only continue to increase the risk of a disastrous black swan event happening and eliminating years if not decades of trust that have been developed and allowed machine learning models to flourish.

However, even if these problems are not found in the near term and the problems persist, allowing biased and invalid models to provide predictions and guidance for practitioners in society ensures and guarantees that systemic biases that have existed in the past continue to propagate (and potentially expand) their impact. Achieving a balance between efficiency and equity in machine learning models and other emergent technology in sensitive spaces is not only an intangible problem that I have made up – it is real and can have serious effects if it is not handled effectively and with care.

I can use methods such as LIME, SHAP, and saliency maps to understand what the important inputs to a network are. But, these methods have variance, can be wrong, or are expensive to compute. I seek to create neural networks that employ methods for explainability that do not succumb to the problems above.

I propose a process which seeks to bring explain ability and fairness without compromising on efficiency into the insurance space using tabular data and mod modified statistical methods. This is merely the first step towards explainable and ethical AI, and will provide the groundwork for further advances in the future.

I aim to provide the first end-to-end method for explainable neural networks with relation to tabular data that can be used in a high-intensity and fidelity environment such as an insurance agency or hospital. The model will prioritize explainability so that human evaluators can decide whether to trust the outputs and logic or not. The process will be done through advanced gradients and loss functions, integrated observability functions, and low-impact model checks and hashing to ensure that there is no tampering with the model.

The final output of the entire program will be a compressed model with a hash that can be used to check whether the model has been tampered with since the original training. This ensures that the model is not only explainable and fair after training, but also that it can continue to be veritably fair into the future.

Relevance to Trustworthy Aspects

Implementing explainable decision modelling for neural networks increases the amount of trust and fidelity exhibited in and by high-importance decision neural networks by improving the model standing in terms of our four principles for trustworthy machine learning: robustness, explainability, privacy, and fairness.

I elucidate how a more transparent method of explaining the decisions made in such a neural network influence each of the verticals below:

1. **Robustness:** Using the gradient and neural network note based fairness and observability metrics, I were able to better understand where the outlier data inputs drive or outputs. If I are able to understand this, we're able to employ methods such as model, knowledge distillation, gradient clipping, as well as random data generation and interpretation in order to train the model on more extreme inputs. Performing this process on the system that I create would allow us to ensure that the model does not act erratically with marginal data inputs.

   By ensuring that I understand how the model thinks, we're better able to protect against erratic inputs nullifying our previous training schema of the model, and will increase end users' trust that the model will generate sound outputs.

2. **Explainability:** The entire point of this framework and problem is to ensure that the outputs that I are getting from our neural network are explainable and justifiable. If I end up with a model that is not explainable or results in outputs that are not accompanied by sound reasoning, then users with either over-trust or under-trust the model. They will rarely treat the model as a helpful suggestion which should be critiqued given unexplainable outputs in high-sensitivity situations and will either stop using the model entirely or delegate everything to the model.

   The explainability aspect of this model ensures that the explanations are valid and reflect the true "thinking" of the neural network. This allows administrators and those that interpret the model to deploy skepticism where the model output warrants it while taking advantage of the efficiency and veritability gains that can be gleaned from explainable and observable AI.

3. **Privacy:** By gaining a better understanding of how the model does its thinking, I are better able to protect users against having their personal information leaked through the model. While this would be a marginal benefit that will likely not be implemented given the time constraints, better observability into the weights and processes of the machine learning model to understand what information is being stored could lead to un-training schema in the future that improve upon current frameworks.

   On the topic of un-training a model: if I am able to get a tabular data model that is able to accurately and legitimately emulate a model that would result in removing some set of features or training points, then this would constitute a large breakthrough in the world of neural networks which are, by definition, highly complex and do not frequently converge in a nice and deterministic model depending on the random seed that is being initialized for the model.

   More observability into the weights and gradients in general can help us better implement schema such as differential privacy to ensure that the adjustments being made in the neural network accurately reflect population characteristics and do not lead to personal information being leaked through the model constraints.

4. **Fairness:** This is another key point of the framework. I are attempting to create customized metrics that ensure that the model outputs are representative and valid without compromising seriously on the effectiveness or efficiency of the model. I must ensure that the model that I end up creating is not entirely irrelevant due to making poor decisions in the name of equity, while also ensuring that the model does not simply exist to continue to advance racial, gender, or other inequities that already exist in the world.

Dataset and EDA

In this experiment, I will be using two main datasets in order to verify that the work that I am doing is working and impactful. I will use one dataset as the testing dataset and the other as the development dataset. That way, I am not biased or influenced by the makeup of one dataset or another in the creation of my model and can verify the methods that build in one domain on the other to elucidate the generalizability of my new techniques.

The insurance dataset that I use below will be more suited to the explainability portion of this project due to the numeric nature of the data already. On the other hand, the criminal justice dataset can be used on the fairness part of this study because of the likely correlation of sensitive variables such as race and gender with outcomes such as verdicts, the length of sentences, and types of punishment.

1. **Dataset 1: Prudential Life Insurance**

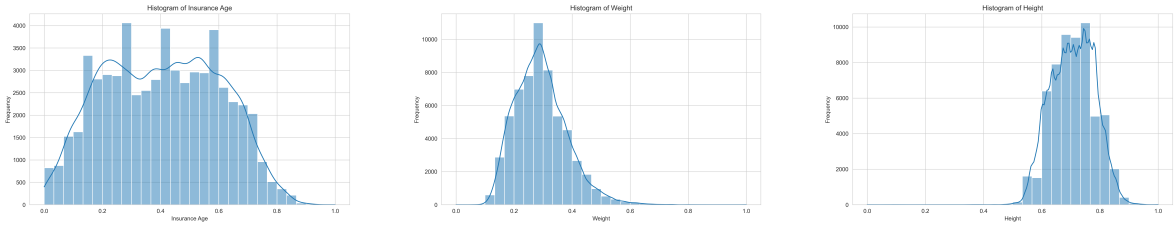
   This dataset was released by Prudential in 2016 as part of a Kaggle competition in order to see if there were better methods than the ones that they were already using for pricing insurance and deciding whether people should be approved for insurance or not.

   The dataset used comprises 59,381 entries across 128 columns, spanning a mix of categorical as well as continuous data types representing things such as the employment status, the income, height, and weight of the applicant.

   The mean height and weight across the dataset are 0.707 and 0.293, respectively, indicating a diverse range of applicants to consider.

   Medical keywords are sparsely populated (as evidenced by the low mean values) which indicate more rare conditions in society being represented by the data. This could mean that the conditions have an outsized impact on the final decision

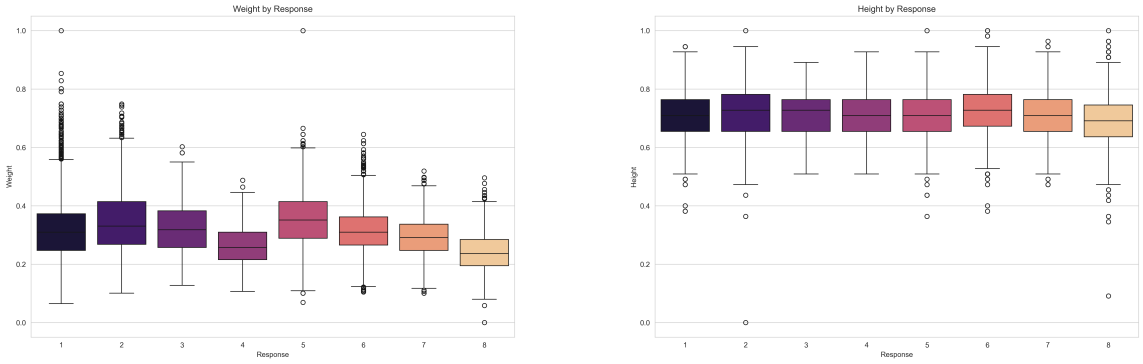
   Additionally, the vast majority of the data that is included in the dataset is normalized to a distribution between 0 and 1.



(a) Histogram of Insurance Age (b) Histogram of Weight (c) Histogram of Height

Figure 1: Histograms of Age, Weight, and Height

We start by plotting histograms of the average age of the people in considered in the dataset. This allows us to see that the distribution of ages of people applying for life insurance is rather broad. With that being said, the weights and heights of the people that are in our dataset are rather clustered around centralized values. Now, because this is a scaled dataset without outlier pruning, this could be an indication of some extreme outliers on either end. But, the distribution mimics reality where much of the population's metrics cluster in healthy ranges.



(a) Weight by Response

(b) Height by Response

Figure 2: Weight and Height by Response

Looking at the average distributions of the responses based on the weights and heights of the applicants, we see that some patterns exist due to the misalignment of the bounding boxes on our distributions. For instance, applicants that received a 4 or an 8 as their response had dramatically lower weights on average than other applicants (especially those who received outcome 5).

Looking at the distribution of people's heights, we find that all of the heights across the different responses are rather uniform. It is important to note that according to the histogram above, there is not a serious amount of variance in the heights anyways. But, the clusters are interesting and important to note.

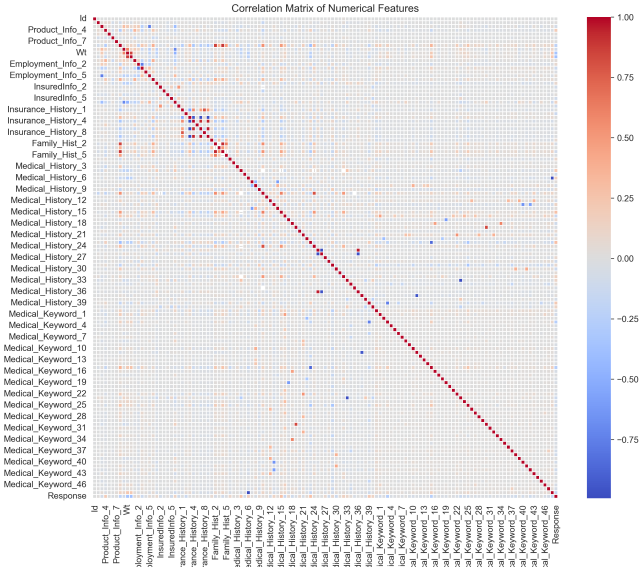


Figure 3: Correlation Output

Here, we plot the correlation between the individual values and inputs of our dataset. It is important to

note that there are too many inputs for the correlation matrix to include all of the labels for the data. However, we see that there exists some degree of correlation between many of the medical histories and terms found in the medical documents. This likely corresponds with things such as a disease being part of the medical history.

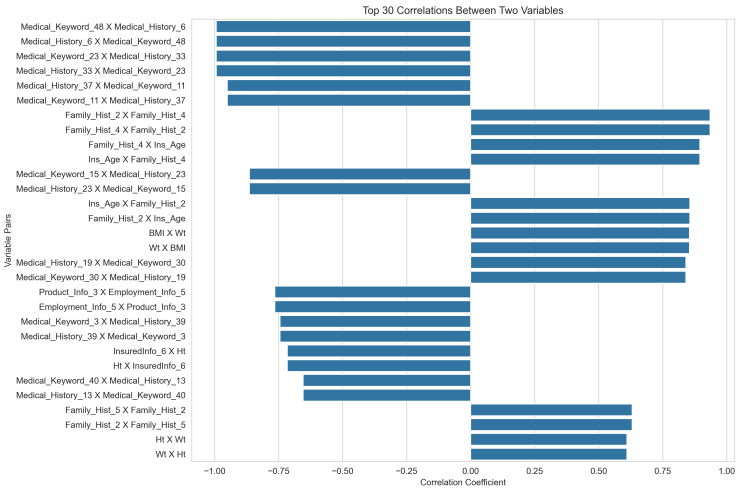
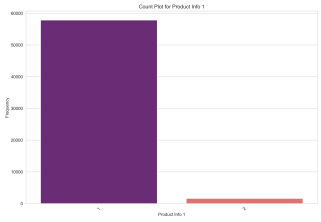
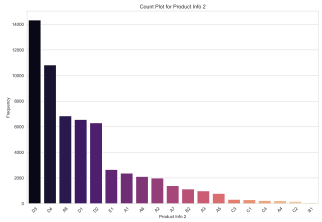


Figure 4: Correlation Output

Taking the top correlations between labels that do not fall in the same category (e.g. no two medical history classes), we see that many of the correlations that are most informative have to do with height and weight with certain outcomes. This makes sense as these are causal factors in many adverse health outcomes.



(a) Count Plot for Product Info 1



(b) Count Plot for Product Info 2

Figure 5: Product Information Heuristics

Looking to the types of products that people are applying for, we can see that the vast majority of people are looking to enroll in Product 1. We assume that this is the standard life insurance package, and the people enrolling in Product 2 are looking for some other form of insurance.

Product Info 2 shows us the more specific type of insurance that people are looking for. We see that there is a set of a few insurance products that are rather popular before we see a steep drop-off in interest.

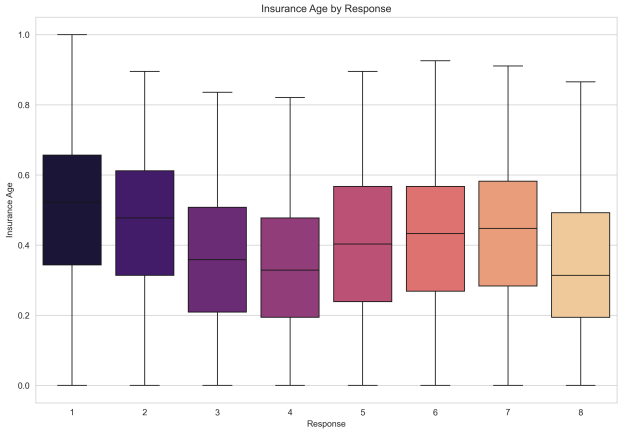


Figure 6: Insurance Age by Response

Looking at the average age of applicant by response, we see that there is a healthy distribution of ages for each response. It does not seem that age is correlated with the outcome of the insurance application.

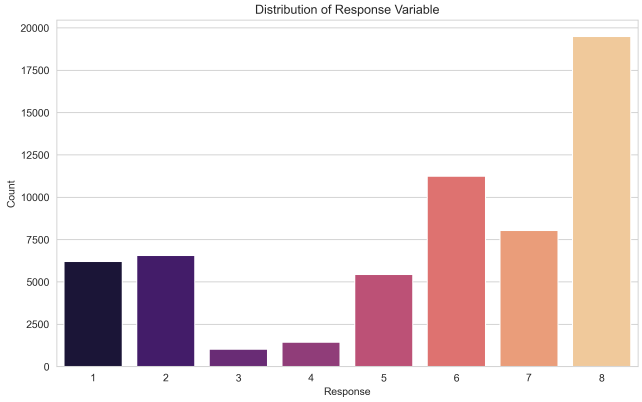


Figure 7: Response Output

Finally, looking at the overall response outputs we see that there is a majority of responses that fall in the 8 category. We assume can assume that this is something along the lines of automatically accepted into the product that they applied for. There are few applicants that fall into the 3-4 response code range. Given that we do not have the actual meanings of the responses for privacy reasons, we cannot know for sure. However, I make an educated guess about the values for each of the variables below:

1. *Declined*: Applicants considered too high-risk to insure.
2. *Postponed*: Have a temporary condition that makes them currently uninsurable, but could be reconsidered in the future.

3. *Rated*: Offered at a higher premium to those who have a higher risk due to medical history or lifestyle choices.
4. *Standard*: Meet the average expectations and are offered standard terms.
5. *Preferred*: Individuals in good health with a lower risk profile offered preferred rates.
6. *Elite*: Applicants in excellent health with an optimal risk profile with best possible rates.
7. *Smoker*: A specific category for smokers who often face select pricing structures due to higher risks.
8. *Accepted*: The majority falling into this category suggests means that this category is something along the lines of "automatically accepted at standard rates".

2. **Dataset 2: Cook County Criminal Courts System**

   This dataset is part of a continually updated dataset which highlights how the Cook County Courts System is doing in terms of crime. It is part of the open data city initiative which encourages municipalities to open their coffers and reams of data to independent researchers. I use this data as a secondary point of analysis for my data work in order to make sure that the machine learning techniques that I develop can be generalized to a wide variety of use cases.

   We perform a brief EDA below.

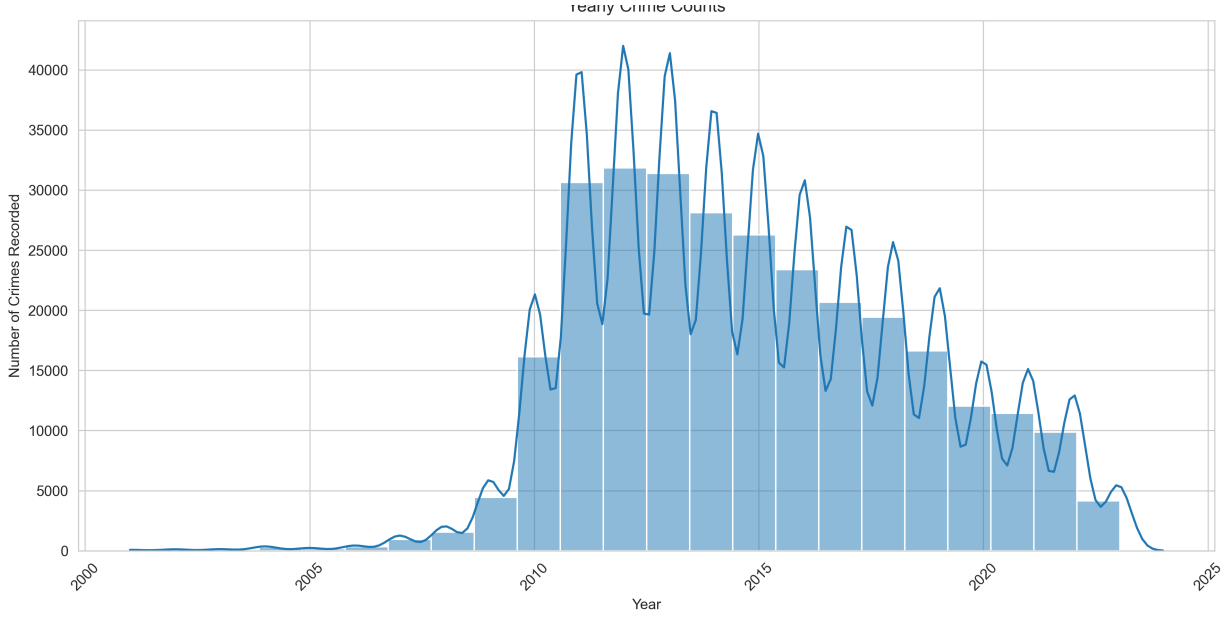


Figure 8: Yearly Crime Counts

This plot shows us the number of crimes that have been committed every year. We plot the histogram of the number of crimes committed to show the number of total crimes by year that we have in our total dataset as well as the line on a per month rolling basis to show the seasonality of the crimes as well.

This data sharing program only started collecting data around 2005 so it seems as though we have some dates that are mislabelled in the dataset.

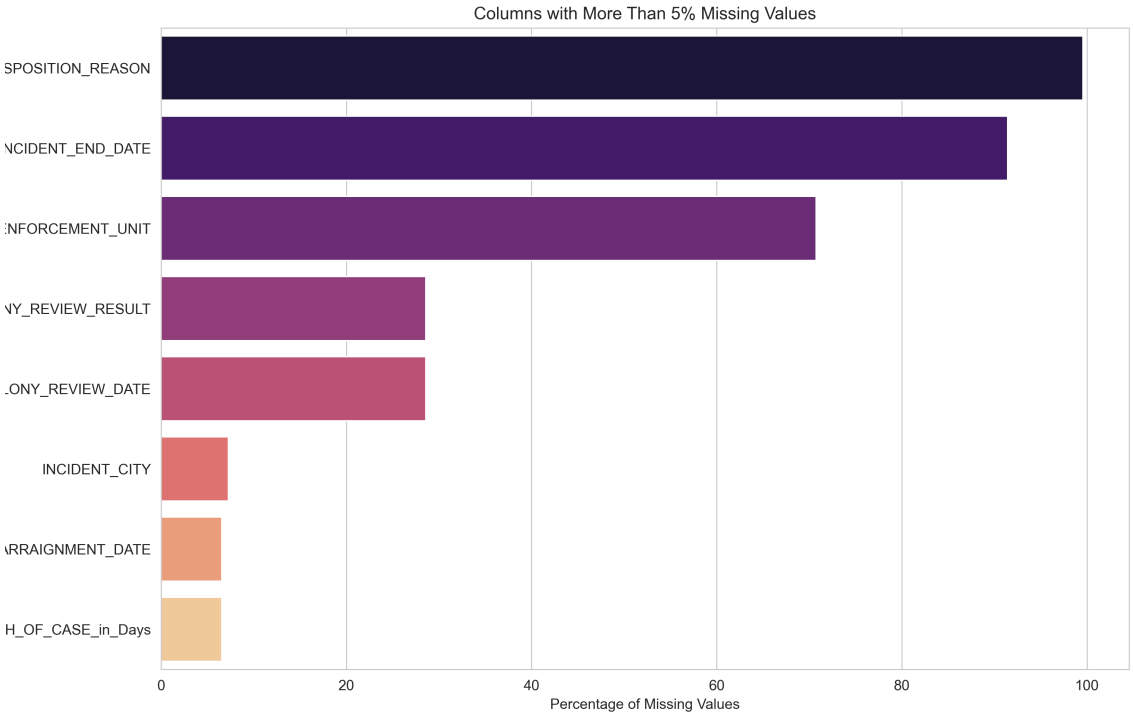


Figure 9: Columns with More Than 5% Missing Values

Additionally, there are some columns that are missing substantial numbers of values which will make it difficult for us to assess the actual impact of these variables on the output. We will employ interpolation schemes to offset these issues.

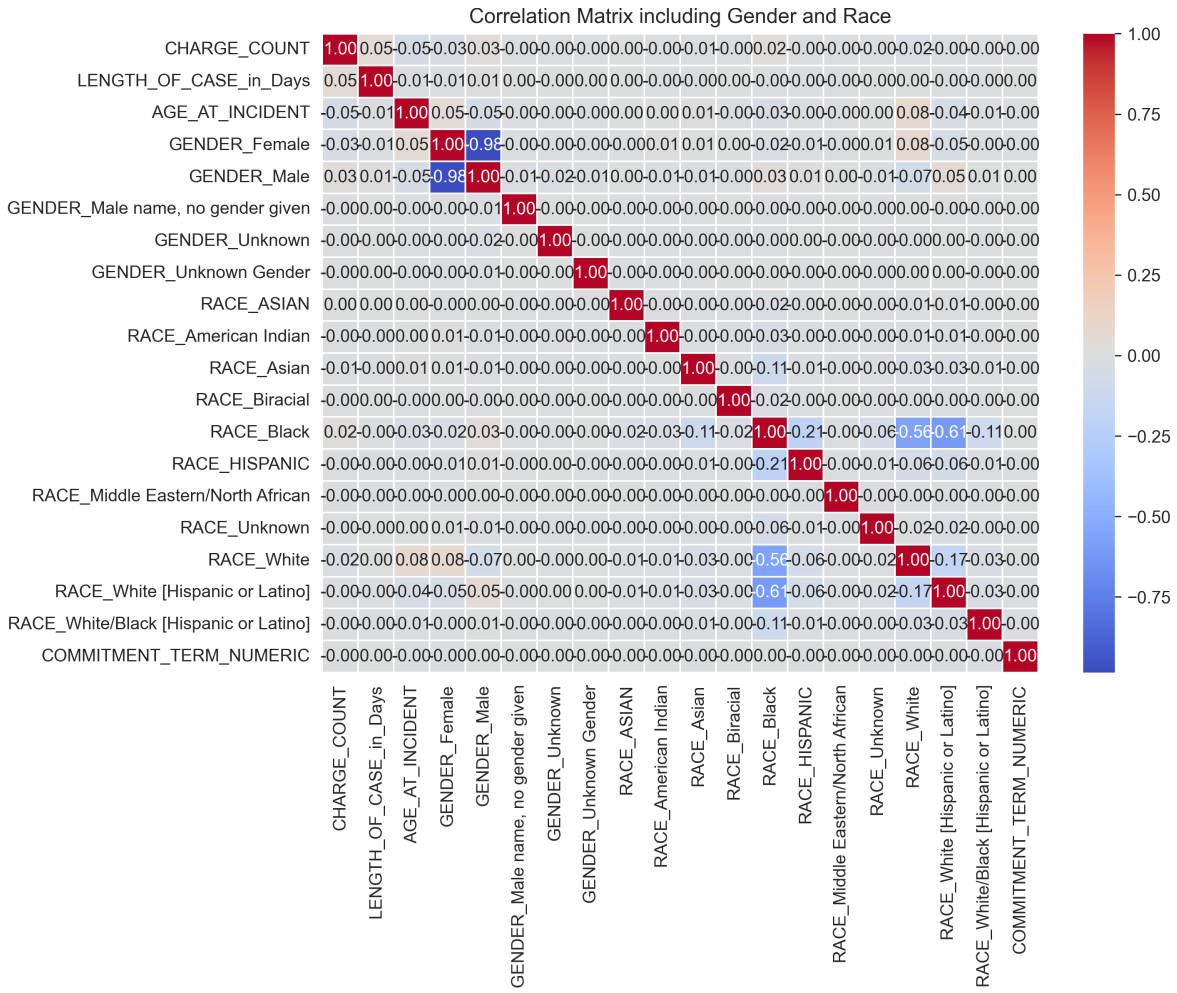
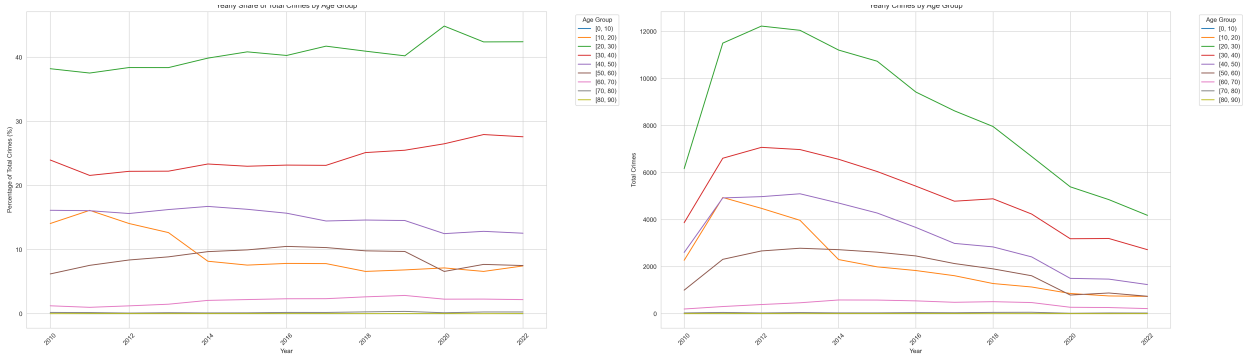


Figure 10: Correlation Matrix of Various Inputs with Sentence Length

We perform a naive correlation check between many of the values in the input data to see if there are any striking correlation values off the bat. We do not find anything partially because of the need to encode everything that is categorical in a one hot value. This messed with our correlation calculations.

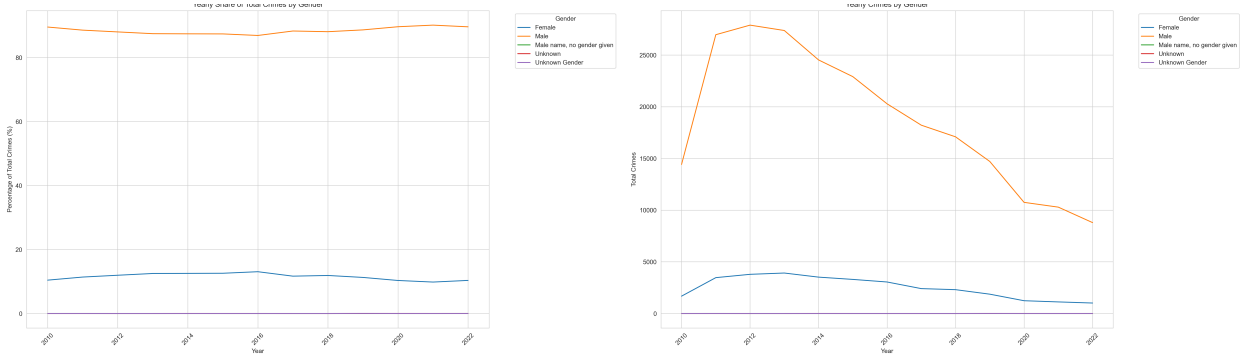


(a) Yearly Share of Total Crimes by Age Group

(b) Yearly Crimes by Age Group

Figure 11: Age group distribution of crime

Next, we can see the share of the total crime by year stemming from each age group in the dataset. We see that there is a gross over-representation of young people consistently across the entire dataset time, and the older that our group of focus gets the less likely they are to have commit crimes.

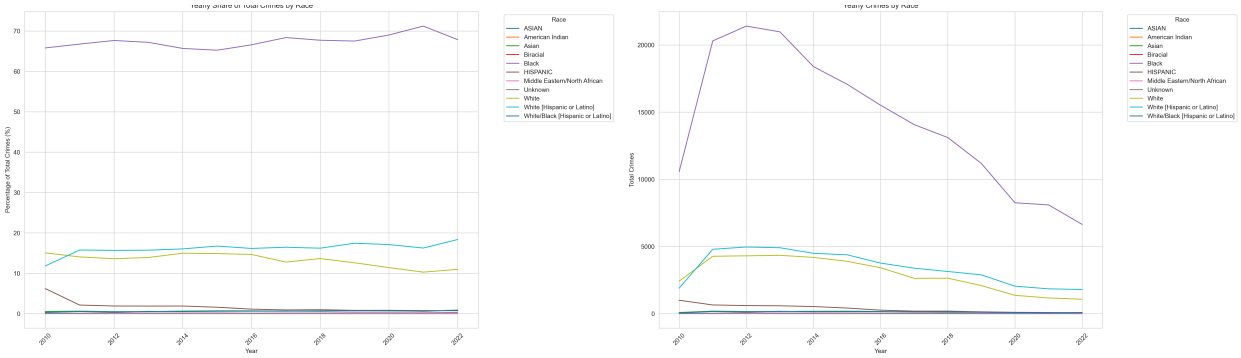


(a) Yearly Share of Total Crimes by Gender

(b) Yearly Crimes by Gender

Figure 12: Gender distribution of crime

Looking at the distribution of gender in the crime statistics, we find that men are extremely over-represented in the data. This does not come from any clear explanations.



(a) Yearly Share of Total Crimes by Race

(b) Yearly Crimes by Race

Figure 13: Racial distribution of crime

Looking at the racial distribution of the crime being committed, we once again see gross inequalities between groups being reported in the data. This could be a biased estimate due to the over-presence of police in minority communities and racial bias in policing (especially in Chicago), yet the results are nevertheless striking.

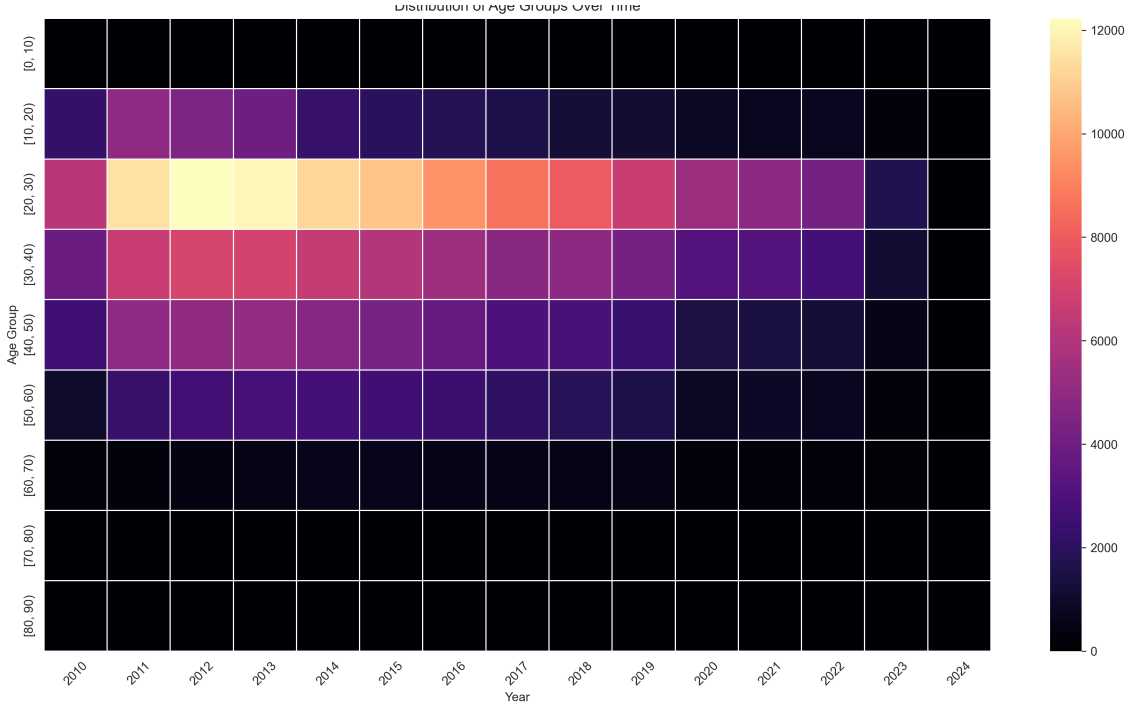


Figure 14: Distribution of Age Groups Over Time

Plotting a heatmap of the representation of different age groups in the data over time, we see similar trends to above where young people are over represented (especially early in our data).

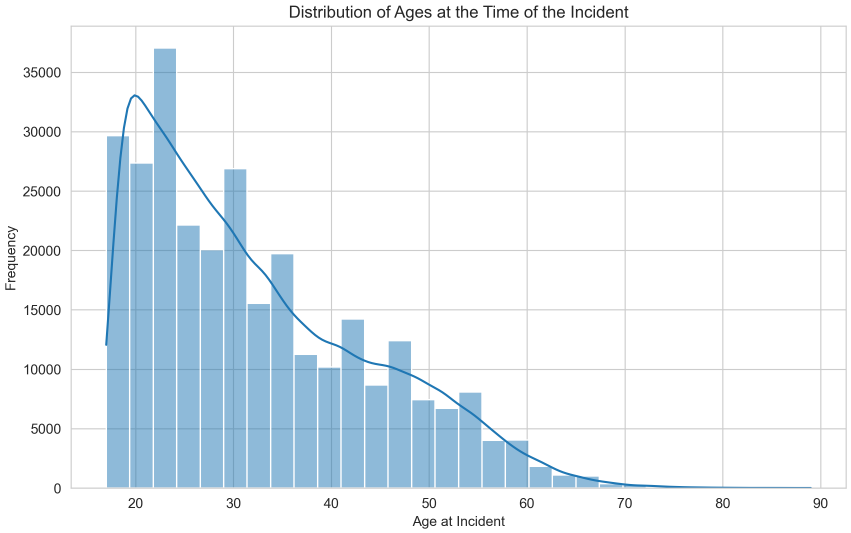


Figure 15: Distribution of Ages at the Time of the Incident

Looking to the ages once more, we see more clearly just how the ages are distributed. There seems to be an early jump once we hit the age of adulthood which is likely due to the reporting requirements that come with adulthood and the presence of juvenile detention as a deterrence in the United States. The number of crimes committed seems to monotonically decrease as the age of our subjects increases.

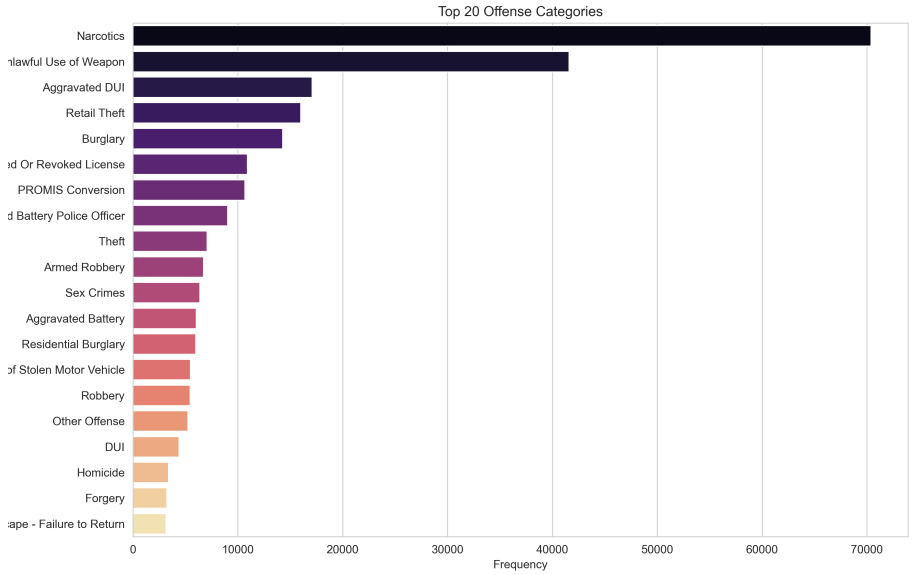


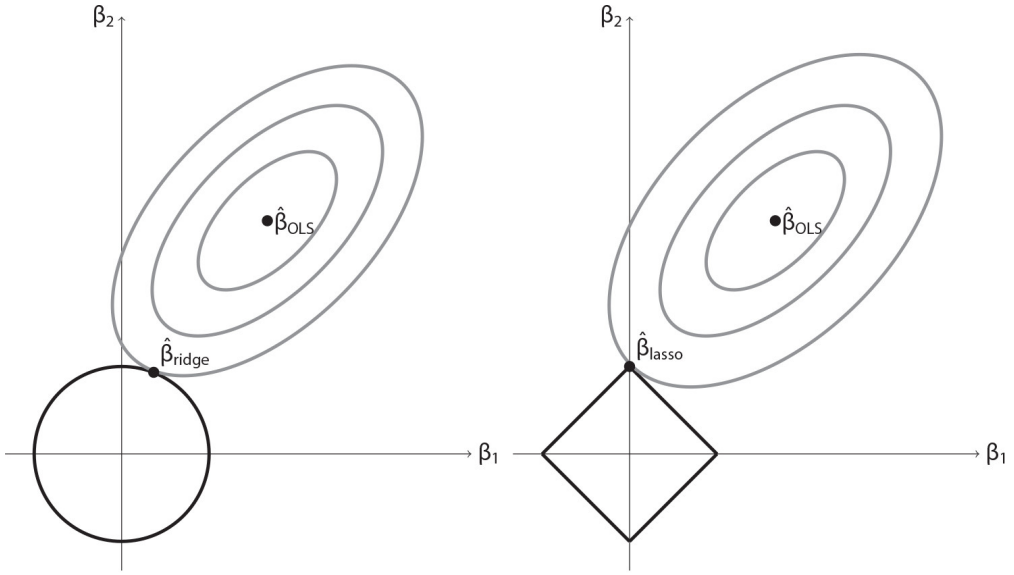
Figure 16: Top 20 Offense Categories

Finally, we seek to understand what the most common offenses are in Chicago. We find that there is an over-representation of narcotics and weapons related charges. There is also a notable amount of violent crime that is represented in the most likely crimes being tried and prosecuted in Chicago.

Description of Related Works

There are several current methods that exist that can be used in order to enforce some degree of explainability. I investigate them below.

1. **Simple Regularization:** Current methods that exist for regularization include L_0, L_1, L_2, and L_∞ amongst others. While each of these regularizations serves a different purpose in the grand scheme of explainability, L_1 is frequently used because of its linear and symmetric structure in terms of the unregularized objective function. If I investigate the graph below showing the "equally good" values for the different regularizations, I can see that L_1 (lasso) regularization strikes a healthy balance between forcing many parameters to zero, and only performing regularization on a select subset of parameters. However, it may be optimal in some cases to use L_2 regularization (ridge regularization) to enforce a smoother equally good boundary and stronger regularization on outliers to the data. Regularization function such as L_0 gives us a plus-shaped "equally-good" graph, while regularization functions with large parameters such as L_∞ give us square-like "equally-good" graphs. [**Ahrens2020Lassopack**]



(a) Ridge (b) Lasso

Figure 17: Equally Good Visualizations for Regularizations [**Ahrens2020Lassopack**]

2. **Adversarial Loss:** This method consists of training a generator and a discriminator in order to try to generate data representations from the latent space input that closely mimic some target outputs. A discriminator can be programmed in a series of different ways such that it is evaluating different metrics for the generated data. One way that the discriminator can be trained is to discern whether some output is logically sound or not. Given enough time, this can lead to explainable and observable outputs being generated by the initial generator model.

   Given the keen and heightened targeting that I can do with this generative-discriminative method, I can ensure that the outputs mimic real life and are therefore somewhat explainable. Additionally, I

can map the throughput of a neural network to understand what kind and aspects of an input influence portions of the output. [**Tung2019SimilarityPreservingKD**]

3. **Quantile Loss:** This method works by allowing a more nuanced understanding of the models predictions. The distribution of the response variable is changed through the model in order to get us a better perspective into the inner workings of the matrix inside of the neural network.

   This provides us with enhanced observability into tail risk of machine learning models by allowing us to predict certain quantities instead of the mean output of a neural network. Effectively, this is allegorical to red-teaming a machine learning model and allows us to understand the variability of the outputs.

   Additionally, this means that the model is more robust and versatile which can often lead to more explainable neural network weights as redundancy is forcibly built into the model. [**Kang2021DataFreeKD**]

4. **Ranking Losses:** this can provide us intuition into the fairness of the model by highlighting overweighted and underweighted classes in the training data. I can understand by the consistency of the rankings of different labels whether the model is likely to prioritize or discriminate against certain output classes. Additionally, this helps us gain insight into the thinking of the model when I apply additional characteristics and outputs, such as probabilities of the final class outputs. If a model is highly confident in a consistently distributed number of output classes, then I might believe that the model is more fair and making decisions more based on the inputs. On the other hand, if the model is highly confident in a select few number of class outputs, then I might believe that the model is biased or skewed towards those classes regardless of the input. Additionally, the distribution of the ranked outputs gives us insight into the other potential classes that I could encounter with slight perturbations in the data.

   In essence, this helps us better understand where the model excels and struggles which offers us a clearer view of its internal workings and "thinking". [**Yang2023BalancedKD**]

5. **Contrastive Loss:** This version of loss effectively helps us understand the key differences between different inputs in our data. Additionally, it extends the models of buses to ensure that it understands the subtle differences that exist between data points that I are attempting to classify using the machine learning model.

   This can be crucial for helping with feature extraction and model precision as I can allow relatively similar though fundamentally different examples to represent contrasting classes when I are training the machine learning model.[**Wang2020Understanding**]

6. **Knowledge Distillation:** This allows us to fine-tune what portions of the data and features our model should be looking at. By teaching the model to only consider certain parts of our extensive knowledge base that I can potentially feed into the model, I ensure that the outputs are applicable and valid for the context of our machine learning model. Part of the benefit of distilling or knowledge to a more simplistic representation and latent space is the potential for higher fidelity in our model weights and in explaining outputs due to the smaller number of features that it must represent and process in an input.

   This allows us to better understand the factors that influence an output, and can potentially open the possibility of cross-validating the entire model such that I get the influence of certain factors on the input and output of the model. [**Gou2020KnowledgeDS**]

However, the problem continues to persist with machine learning. Several empirical studies have shown that the methods deployed above to deal with the problem of explanation come with detrimental effects on the accuracy of the treated models. Hence, my project will differ from the approaches above in two key ways:

1. *Weight Perturbation:* Every approach that I have detailed above largely works in a post-hoc manner without serious changes to the model weights. The methods that have the largest impact on the model weights are regularization and some of the loss functions.

However, there are no methods that work to change the weights of the model after the model has been optimized. This is likely because of the belief that the model is as good as it can be after training and any loss in accuracy in the name of explainability or fairness is bad.

However, my method will actually change the weights of the model after it has been trained before allowing slight fine-tuning adjustments to ensure that we gain a model that strikes a balance between fairness, explainability, and accuracy according to historical data. I believe that this novel approach will allow us to create more fair and explainable models without seriously sacrificing accuracy of the model.

2. *Novel Methods:* My consideration of work such as untraining and integrated explanations are simply put novel. They have by definition not been tried before.

Additional papers are included below. These papers speak about explainability to varying degrees and provide me with additional information that I am using through the course of this project. Notable papers include the VQA and Berkley Deep Drive papers for their relation to my human-readable feedback method.

Explainable Deep Learning: A Field Guide for the Uninitiated
Towards Explainable NLP: A Generative Explanation Framework for Text Classification
Nothing Else Matters: Model-Agnostic Explanations By Identifying Prediction Invariance
Textual Explanations for Self-Driving Vehicles
Explaining NonLinear Classification Decisions with Deep Taylor Decomposition
A Unified Approach to Interpreting Model Predictions
Statistical Mechanics of Deep Learning
Causal Shapley Values: Exploiting Causal Knowledge to Explain Individual Predictions of Complex Models
ERASER: A Benchmark to Evaluate Rationalized NLP Models
How to Explain Individual Classification Decisions
TED: Teaching AI to Explain its Decisions
"Why Should I Trust You?" Explaining the Predictions of Any Classifier
Multimodal Explanations: Justifying Decisions and Pointing to the Evidence
Anchors: High-Precision Model-Agnostic Explanations

Proposed Approaches

I will try advanced loss functions, custom batch-normalization, and a series of other methods to enforce fairness and explainability in my machine learning model. Through this process, I will try 3-main methods along with others that I can think of through the brainstorming process in order to generate explainable and valid machine learning models. A collection of the methods that I have devised so far exist below:

1. **Advanced Batch Normalization:** During the training process, I want to sample inputs so that the inputs are representative of the actual data distribution that I will be seeing throughout the course of the model.

   Ontop of that, as the inputs are propogated through the course of the model, I will continually update parameters for the inputs to each of the layers so that I can understand which inputs are being increased the most and attempt to correlate that with the inputs that are being considered as the most important in the input to the model

2. **Node-Explainability Loss Functions:** Over the course of training, it is important that I coax the model towards representing more explainable and tangible concepts for the user. In order to do this, I will attempt to implement a loss function that scales the coefficients of the regularization and sparsity of results over the course of the model progressively higher as the gradients are backpropogated

closer to the front of the model. This means that in the first layer, I perform minimal categorization and pruning. But, towards the later models, I are attempting to make the number of nodes that are activated for any one prediction as small as possible while making the divergence of the resulting latent-space vector representations as high as possible.

One equation that I can use for the loss function in order to complete this task is below. It changes dependingon the layer that I are considering:

$$L(i) = L_{base} + \lambda \times e^{\alpha i} \times L_{layer-wise}$$

Where the following information describes each of the variables:

- $L(i)$ represents the total loss corresponding to the i^{th} layer.
- L_{base} stands for the model's basic training loss over all the data.
- λ is a practical factor to modify the total level of the area of the corresponding system's box.
- $e^{\alpha i}$ is an exponential factor that helps to reasonably spread the sense of the specific space, based on the map of the class at the base and the final task of the clear class.
- α is a true fact that gives the idea of the specific same part, changing with the time of the new change for the road of the school, making it a kind of mix and great way.
- i is the index of the viewed layer, making a road from the starting to the end.

This is a generalized version of a loss function that I can use such that I increase the number of nodes that are being regularized over the course of the execution of the algorithm percentage-wise. For example, I might increase the number of nodes that are being regularized and manipulated at each layer by 5% per layer until a max of 50% of the nodes have been regularized in the final layers.

This ensures that the model is training so that I get some sort of explainability in terms of inputs at the nodes. To take this one step further, I would be interested in implementing some loss function so that similar outputs light up similar nodes in the neural network later in the network. On the other hand, highly similar inputs with different labels should be forced to light up progressively more unique combinations of neurons later in the network as compared to inputs with different labels. An example of such a loss function would look something like:

$$L(i) = L_{\text{base}} + \lambda \times e^{\alpha i} \times L_{\text{layer-wise}} + \beta \times L_{\text{contrastive}}(i) \tag{1}$$

where:

- $L_{\text{contrastive}}(i)$ represents the contrastive part of the loss that operates on the latent space representations. It encourages similar inputs to activate similar nodes and dissimilar inputs produce outputs even early in the network that are markedly different from different inputs' propogated outputs. This could be implemented using something like the cosine similarity or Euclidean distance between the feature representations pairs of inputs.
- β is a weighting factor that controls the influence of the contrastive loss component. It is a hyperparameter that should be tuned in the same fashion as any other hyperparameter.

Additionally, I will compute the importance of each node by multiplying the gradient that it outputs with the importance that it has with respect to the input to get the most important nodes of the neural network.

3. **Influence-Limiting Functions:** Another way that I can ensure that the model is removing the biased parameters is by limiting the influence that any confluence of input factors or individual inputs

can have. Of course, this has been done before. So, I provide a derivation which I expect to provide markedly better results for the end user.

There are several methods for removing the influence of a node without removing it from the entire neural network or forcing retraining. This is a derivative of the re-training that I have mentioned earlier, but is substantially less efficient than actual re-training and incremental learning methods that have been proposed so far.

With that being said, this has the potential to make the discovery of patterns and information in the machine learning space substantially more efficient. With influence-limiting functions, I can ensure that problematic inputs are not considered in the final decision of the neural network while more important ones are. This can force compliance in a way that would not have been possible before.

A potential method is shown below, but is not discussed in serious detail due to the fresh nature of the idea and iteration to ensue. Assume I have a vector of elements that multiply with some inputs in an element-wise manner to produce the output of the neural network layer $a_1, a_2, \ldots, a_n$, and I want to modify each element a_j $(j \neq i')$ by multiplying the element by $(1 + \frac{\|a_{i'}\|}{\sum_{k=1}^{n} a_k})\%$ in order to step-up the influence of these nodes, and set $a_{i'}$ to zero. This ensures that the impact of $a_{i'}$ on this layer in the neural network which I construct is zero. The operation for each a_j $(j \neq i')$ can be expressed as follows:

$$a'_j = a_j \cdot \left(1 + \frac{\|a_{i'}\|}{\sum_{k=1}^{n} a_k}\right)$$

where for $a_{i'}$, I simply set:

$$a'_{i'} = 0$$

Combining these into a single equation to represent the transformation of the entire sequence, I get:

$$a'_j = \begin{cases} 0 & \text{if } j = i' \\ a_j \cdot \left(1 + \frac{\|a_{i'}\|}{\sum_{k=1}^{n} a_k}\right) & \text{otherwise} \end{cases}$$

This equation completes the operation where every element except for $a_{i'}$ is increased to maintain the relative performance of the network, and $a_{i'}$ itself is turned into zero. This is the simplest version of influence-limiting where the actual weights of the elements are reduced towards zero. But, this method requires subsequent fine tuning and retraining.

Additionally, one point that I made above that has not been addressed so far is the potential introduction of un-learning schema. This is another discipline that I would like to explore as a potential backup to or supplement for the machine learning model that I work on for observable tabular data. I likely will not have time to implement this due to the timeline of the project.

I believe that I could implement something like this using KL-Divergence as part of a loss function to penalize the model from going too far from the original inputs while enforcing a greater, error-increasing loss on any inputs that have a mahalanobis distance close to the inputs that I are trying to erase in order to push the gradient away from optimizing for said class or inputs while maintaining model accuracy. The KL-Divergence is defined as follows:

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

Therefore, an effective loss function might look something like the following:

$$\text{Hybrid Loss} = \text{Loss}_{\text{General}} + \lambda \cdot 1\{\text{Dist}_{\text{Mahalanobis}} \leq \text{Closeness}_{\text{Threshold}}\} \cdot \text{Loss}_{\text{Error Increment}} + \mu \cdot D_{\mathrm{KL}}(P \parallel Q)$$

This ensures that the inputs that are similar to the one that I are untraining on receive positive error while other inputs ensure that I are keeping the model at a somewhat steady state.

Evaluation Metrics

One of the key reasons for providing two datasets above is to have multiple mediums and methods that I can draw on in order to ensure that I am not being swayed by the results of a single dataset to create an over-specific method. The first way that I will evaluate my methods and ideas detailed above is by running the same methods on a the criminal justice dataset which will result in more explainable outcomes for why each of the sentences are as long as they are.

Beyond that, I will consider the following metrics. Some have been defined and created by me for the task at hand while others are established metrics:

1. *Node Amplitude Correlation:* One of the methods that I will attempt above is to create a loss function that pushes the nodes to activate in a more explainable manner – instead of purely optimizing the final result's accuracy, I would like sets of nodes to be active or not depending on the presence of combinations of input features in the data.

   For instance, if some set of medical history inputs are activated that have good predictive value with the insurance decision or there is a specific combination of traits in the crime committed and personal profile of the offender in our Chicago crime dataset which is likely to increase the sentence length, I would like for the appropriate nodes to be activated.

   To measure this, I will take the output values of the perceptron at each of the layers of my network and feed each of the outputs into a separate regression model for each layer with the label as the target that we are attempting to predict.

   If the R^2 value for the regression that I run coming from the network that I train with my custom loss function is higher than the network that I train without said loss function (whether logistic or linear) has for an R^2 value by a statistically significant margin, then I find that my custom loss function has worked in improving the explainability of the network neurons, and this work can be extended into higher fidelity models in the future.

2. *Outcome Correlation Analysis:* I can perform regressions and construct decision tree models to measure the effect that sensitive variables can have on predicting the outcome of situations in both the insurance and criminal justice datasets.

   I will first perform these regressions on the outcomes of the test datasets that I construct using a network that has not been trained with my importance-decreasing and explainability methods detailed above.

   Then, I will perform the same regressions on the model outcomes of the test datasets using a model that has employed my methods. If we find that there is a lower correlation between the sensitive variables and the outcomes of the model, then we can assume that the methods employed have worked.

3. *Perturbed Input Analysis:* By changing the values of the inputs that we consider sensitive variables and perceiving how much the confidence of the output as well as the output itself changes, we can piece together how important the input is for the output. We can test the impact of sensitive variables before and after the methods that I have proposed above to see if the methods work to change the reliance of the model on said sensitive variables.

4. *Conventional Methods:* We can also deploy conventional methods that we have studied in class such as LIME and SHAP to see the impact of sensitive variables on the outcome. I will not detail how said methods function here due to our extensive study of them already. But, these methods can be used to simply understand through another perspective whether the methods that I have employed are working to reduce the importance of sensitive variables.

5. *Human Review:* A simple way that I can measure whether the network has gotten improved explainability and fidelity is through manual human review.

   An extension of the project (if I have time) would be to use methods employed in Visual Question and Answer competitions such as this paper to generate human-undertandable responses for each of the datasets. I could potentially use a large language model such as LLaMa to generate naive explanations for the outputs that we get and feed these as inputs into a language model extension of my model which explains why decisions are being made.

   Even if I am not able to do this, though, I believe that creating a saliency map for the tabular data inputs that I am using in either situation and simple coefficient analysis will allow me to make a judgement call on whether the model is improving its explainability and fairness.

   I would employ methods such as correlation analysis between sensitive variables and outcomes to see whether the model has reduced the effect of sensitive variables on the outcome.

Timeline

Phase 1: **Preparation and Initial Research (February 24 - March 7)**

- Finalize project scope and objectives.
- Complete a literature review on enforced fairness in machine learning for insurance and criminal justice applications.

Phase 2: **Baseline Model Development (March 8 - March 21)**

- Clean and preprocess data for initial model training.
- Develop and train baseline machine learning models.
- Evaluate performance and fairness metrics.

Phase 3: **Importance Reduction Methods (March 22 - March 29)**

- Begin development of importance-reduction methods by analyzing weights.

Phase 4: **Finish Importance Reduction and Begin Explainability Metrics (March 8 - March 25)**

- Continue work on sensitive input importance reduction methods.
- Begin work on neuron explainability loss function.

Phase 5: **Advanced Model Development (March 26 - April 15)**

- Implement and train advanced models incorporating fairness constraints.
- Iterate on models based on performance and fairness evaluations.
- *Milestone (April 5):* Finished product of importance reduction method.

Phase 6: **Model Evaluation and Refinement (April 16 - April 23)**

- Rigorous testing of models with various metrics including those defined and detailed above.
- Finish custom loss function for advanced model neuron explainability.

Phase 7: **Drafting Initial Report (April 24 - April 30)**

- Begin drafting the project report.
- Collect and organize results.

Phase 8: **Reading Period (April 26 - May 1)**

- Continue working on the project report draft.

Phase 9: **Report Submission and Project Closure (May 2 - May 5)**

- Finalize and proofread the project report.
- Prepare for project presentation.
- Submit the final project report by May 5th.
- Project completed.

Conclusion

Given the methods above, I seek to provide further transparency on tabular-data-focused machine learning models and the thought processes driving them without compromising on accuracy. The ultimate goal is to enforce a policy of trust such that users are not over-confident in the abilities of the models while continuing to find the model outputs valid and useful in measured and sensitive environments such as criminal justice and insurance.

Works Cited

References

[1] A. Ahrens, C. B. Hansen, and M. E. Schaffer, "lassopack: Model selection and prediction with regularized regression in Stata," *The Stata Journal: Promoting communications on statistics and Stata*, vol. 20, no. 1, pp. 95–115, 2020.

[2] F. Tung and G. Mori, "Similarity-Preserving Knowledge Distillation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1365–1374.

[3] M. Kang and S. Kang, "Data-free knowledge distillation in neural networks for regression," *Expert Syst. Appl.*, vol. 175, 2021, Art. no. 114813.

[4] Y. Yang, S. He, Y. Qiao, W. Xie, and T. Yang, "Balanced Knowledge Distillation with Contrastive Learning for Document Re-ranking," in *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, 2023.

[5] F. Wang and H. Liu, "Understanding the Behaviour of Contrastive Loss," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2495–2504.

[6] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2020.

Works Referenced

References

[1] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to Explain Individual Classification Decisions," *arXiv preprint arXiv:0912.1128*, 2009.

[2] Anonymous, "Learning Global Additive Explanations for Neural Nets Using Model Distillation," in *International Conference on Learning Representations*, 2019, *Under review.*

[3] G. Montavon, S. Bach, A. Binder, W. Samek, and K.-R. Müller, "Explaining NonLinear Classification Decisions with Deep Taylor Decomposition," *arXiv preprint arXiv:1512.02479*, 2015.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier," *arXiv preprint arXiv:1602.04938*, 2016.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, "Nothing Else Matters: Model-Agnostic Explanations By Identifying Prediction Invariance," *arXiv preprint arXiv:1611.05817*, 2016.

[6] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 4765–4774, 2017.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[8] H. Liu, Q. Yin, and W. Y. Wang, "Towards Explainable NLP: A Generative Explanation Framework for Text Classification," *arXiv preprint arXiv:1811.00196*, 2019.

[9] M. Hind, D. Wei, M. Campbell, N. C. F. Codella, A. Dhurandhar, A. Mojsilović, K. N. Ramamurthy, and K. R. Varshney, "TED: Teaching AI to Explain its Decisions," in *AAAI/ACM Conference on AI, Ethics, and Society (AIES '19)*, 2019.

[10] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *arXiv preprint arXiv:1901.00596*, 2019.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-Precision Model-Agnostic Explanations," *arXiv preprint arXiv:*, 2020.

[12] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli, "Statistical Mechanics of Deep Learning," *Annual Review of Condensed Matter Physics*, vol. 11, pp. 501–528, 2020.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2019.

[14] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, "Textual Explanations for Self-Driving Vehicles," *arXiv preprint arXiv:*, 2020.

[15] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," *OpenAI Blog*, 2018.

Explainable Deep Learning: A Field Guide for the Uninitiated
Towards Explainable NLP: A Generative Explanation Framework for Text Classification
Nothing Else Matters: Model-Agnostic Explanations By Identifying Prediction Invariance
Textual Explanations for Self-Driving Vehicles
Explaining NonLinear Classification Decisions with Deep Taylor Decomposition
A Unified Approach to Interpreting Model Predictions
Statistical Mechanics of Deep Learning
Causal Shapley Values: Exploiting Causal Knowledge to Explain Individual Predictions of Complex Models
ERASER: A Benchmark to Evaluate Rationalized NLP Models
How to Explain Individual Classification Decisions
TED: Teaching AI to Explain its Decisions
"Why Should I Trust You?" Explaining the Predictions of Any Classifier
Multimodal Explanations: Justifying Decisions and Pointing to the Evidence