CAMEO: Explaining Consensus and Expertise Across MOdels

by

Andy Yu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2024

Abstract

Explainable AI methods have been proposed to help interpret complex models, e.g., by assigning importance scores to model features or perturbing the features in a way that changes the prediction. These methods apply to one model at a time, but in practice, engineers usually select from many candidate models and hyperparameters. To assist with this task, we formulate a space of comparison operations for multiple models and demonstrate CAMEO: a web-based tool that explains consensus and expertise among multiple models. Users can interact with CAMEO using a variety of models and datasets, to explore 1) consensus patterns, such as subsets of the test dataset or intervals within feature domains where models disagree, 2) data perturbations that would make conflicting models agree (and consistent models disagree), and 3) expertise patterns, such as subsets of the test dataset where a particular model has surprising performance compared with other models.

Acknowledgements

I would like to thank my supervisor, Professor Lukasz Golab, for his valuable guidance and support towards the writing of this thesis. He was very willing to give me time to discuss my research and any problems that I had issues with. I would also like to thank Professors Parke Godfrey and Jaroslaw Szlichta, Dr. Divesh Srivastava, and Joel Rorseth for their constructive feedback on CAMEO. Finally, I would like to thank the Department of Computer Science at Waterloo University for general support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Society is increasingly adopting AI models across numerous decision-making domains that range from credit scoring [13] to employment assessments [31]. However, these advances come at the expense of rising complexity for data-intensive models, which impacts several steps of the machine learning workflow such as feature extraction, feature transformation, model selection, and model evaluation [39]. For example, extracting meaningful feature sets from a large number of attributes is a non-trivial problem that requires repeated testing against the data itself [3]. The complexity of adopted models makes it difficult for users to easily understand internal mechanisms and interpret their results: users often view these models as *black boxes* [6]. Thus, helping users interpret their AI models serves as motivation for solutions that support *explainable artificial intelligence* (XAI). We generally categorize work towards explaining machine learning models as *instance-centric* and *feature-centric*:

1. *Instance-centric* explanation methods identify unusual or surprising data subsets with respect to a given property. Examples include InfoMoD [14] and *model slicing* [10], which find subsets of the test dataset where a model under- or over-performs compared against its average behavior. Additionally, GOPHER [30], finds subsets of the training dataset that contribute to model bias.

2. *Feature-centric* explanation methods identify important features in a model. Examples include *saliency* methods such as SHAP [26] and LIME [33] that score model features according to their importance to the prediction, and *counterfactual* methods such as DiCE [28] and NICE [7] that identify minimal perturbations to feature values that change the model output.

These XAI methods are crucial for many tasks such as model debugging, bias mitigation, evaluating data quality during training, and establishing trust behind model outputs and decision-making processes.

1.1 Problem and Applications

Machine learning operations or *MLOps* represent a set of best practices for the reliable development, deployment, and maintenance of models in production. A typical *machine learning pipeline* that works towards deploying models will include steps geared towards data collection of labeled and unlabeled data, as well as training and testing various candidate models. A crucial step before deployment is then *model selection*, the process of singling out the most appropriate model from this set of candidate models for the task of inference or prediction. Selecting an unsuitable model that poorly fits the data can contribute to misleading outputs; engineers responsible for model selection want to deliver a model with excellent predictive and computational performance for a given problem. Thus, engineers typically consider the following actions when performing model selection:

1. Experiment with different types of models (e.g., multi-layer perceptrons [35], extreme gradient boosting models [9], ensemble learning models [37]) that they believe are appropriate for the given problem to then identify a machine learning algorithm that best suits their problem. Note that an engineer responsible for model selection may not necessarily be responsible for training the models themselves; they might be given models of varying types to then select from as part of the machine learning pipeline.

2. Tweak tuning parameters or hyperparameters of a machine learning algorithm, that results in different models, to identify a model configuration that best improves desired metrics such as accuracy, sensitivity, and training costs. Similar to the above, as part of the machine learning pipeline, the engineer might be given models with varying hyperparameters to then select from.

3. Rank candidate models against each other by estimating their respective performance on various datasets via model selection techniques [32].

Several model selection techniques exist in the literature such as resampling methods that score models based on their performance on out-of-sample data (e.g., cross-validation [17]) and probabilistic measures that score models based on their model complexity and performance on training data (e.g., Akaike information criterion [1]). Several methods for hyperparameter tuning also exist such as random search [5] and automatic tuning frameworks (e.g., Hyper−Tune [25]). However, these techniques do not offer explainability over candidate models. A model engineer could select a best-performing model according to these techniques that unknowingly contributes to adverse societal outcomes such as amplification of bias. For example, Seyyed-Kalantari et al. found that chest X-ray prediction models consistently and selectively underperform for patient groups such as Black female patients. If left unchecked, these model biases can escalate existing systematic biases and contribute to real-life consequences such as under-served patient subgroups receiving suboptimal medical care [41].

To the best of our knowledge, existing explainability work focuses on an individual model in isolation. However, as described above, model engineers in practice face the task of selecting a model from a vast range of models, and fine-tuning numerous hyperparameters. They must perform thorough model comparison to help mitigate the potential business impact of replacing an old production model [46]. To find areas of improvement, model engineers need to understand the strengths and weaknesses of models such as why, when, and where a model performs better or worse than other candidate models (e.g., a model returning inconsistent results relative to other models) [22, 47, 49]. This model comparison can also provide insights into model ensembles, such as finding models that complement each other to then effectively combine for improved performance [49].

A simple solution for explaining multiple candidate models is to compute accuracy and sensitivity metrics, as well as explanations via XAI methods, for each model individually and then compare them manually side-by-side [47]. This process is unfortunately not scalable as the sets of candidate models, hyperparameters, and input features can be very large. For instance, GPT−4 has more than a trillion parameters [4]; training NASNet to convergence on the image database ImageNet took multiple days [51]. The growing size of datasets and complexity of models also require techniques to summarize relevant information at a high level to ease inspection [49]. Thus, the described simple solution is not user-friendly, and can fail to unearth many relevant patterns.

A significant challenge lies in defining a space of explanations that explain multiple models. For example, what does it mean to provide intuitive instance- or feature-centric explanations for multiple models as part of bias mitigation? Can we concisely compare multiple models and find patterns that offer valuable insights into model selection and hyperparameter tuning?

1.2 Contributions

To address the emerging problem of finding and explaining differences or similarities between how a set of models behave, we present CAMEO, an MLOps tool engineered to explain model agreement across multiple models and find where models are relative experts. Our contributions span conceptual, technical, and practical aspects, offering insights for various use cases. We now summarize our contributions below:

1. *Problem Formulation.* On the conceptual front, we formulate a novel space of model comparisons along two axes. First, we explore *how* to define consensus and expertise, while also distinguishing between *prediction* consensus (i.e., if models made the same prediction) versus *logic* consensus (i.e., if they paid attention to the same features when making their decision or whether they had similar SHAP explanations for a given instance). These consensus measures enable users to evaluate the robustness of their models over various contexts and also isolate potential reasons for agreement or disagreement. For expertise, CAMEO can find surprising subsets of the test dataset where a model has atypical performance compared with other models, enabling users to better find where models are a relative expert in.

   Second, we examine *where* we find expertise, consensus, or discord amongst the models, considering row-wise data subsets versus column-wise intervals within feature domains.

2. *Explaining Consensus and Expertise.* On the technical front, we present CAMEO, an interactive web-based tool designed to facilitate these comparisons. We employ information-theoretic rule mining methods and feature plots to identify notable data subsets and intervals, enabling users to then drill into via counterfactual analysis to better understand the decision boundary between model agreement and disagreement.

3. *Applications.* On the practical front, we demonstrate how users can interact with CAMEO on real-world datasets, including strip-search arrestee data from the Toronto Police Service, New York City flight departure delay data, S&P500 financial data from Yahoo Finance, and the Pima Indian Diabetes dataset. This demonstration showcases CAMEO's ability to offer valuable insights into model selection and hyperparameter tuning. For example, if a user selects the strip-search arrestee dataset and a set of predictive AI models, CAMEO can find informative subsets where models have unusually high prediction disagreement. This allows users to identify models that handle sensitive subsets well (e.g., Indigenous youth).

We specifically detail the following curated use cases:

(a) Model Type Selection: CAMEO can identify unusual consensus patterns for a set of various pre-trained models with varying model types.

(b) Hyperparameter Tuning: CAMEO can identify unusual consensus patterns for a set of pre-trained models with varying hyperparameters.

(c) Finding Model Experts: CAMEO can identify unusual patterns where a model appears to be an expert relative to other candidate models.

1.3 Outline of the Thesis

In Chapter 2, we provide some background about explanation tables, the aforementioned information-theoretic rule mining approach that CAMEO leverages. In Chapter 3, we discuss related work to CAMEO and provide additional background on related XAI methods. In Chapter 4, we discuss our approach and formulate a conceptual answer to the problem. In Chapter 5, we discuss our curated use cases performed on real-world datasets. Finally, in Chapter 6 we discuss both the limitations of our work and potential extensions for future research.

Chapter 2

Preliminaries

In this chapter, we provide some background on *explanation tables* [16], an information-theoretic ruling mining method that CAMEO leverages to generate row-wise patterns.

2.1 Explanation Tables

Consider a dataset D with a set of features F (i.e., dimensional attributes that are numeric or categorical) and a target T (i.e., measure attribute that is either numeric or binary). Let $F = \{F_1, ..., F_f\}$ be this set of f features, where $dom(F_i)$ is the domain of the ith feature. The wildcard or star character, "*", represents all possible values of that feature, akin to ALL in a data cube.

To summarize the distribution of T with respect to F, an explanation table consists of k patterns that contain the most information about this distribution. Although k is a user-specified parameter, for the purposes of this thesis, we expect k to range between 10 and 20 so that we do not present too many patterns to a human user. Each pattern r (i.e., a row-wise data subset or rule) is then a tuple from the data cube over F (i.e., $(dom(F_1) \cup \{*\}) \times \cdots \times (dom(F_f) \cup \{*\})$) that captures a group of datapoints—expressed as conjunctions of values or value ranges—for the given set F. We specify that a tuple t in D matches r if and only if either $r[F_j] = *$ or $r[F_j] = t[F_j]$ for each feature in F.

We now use a similar dataset and explanation table to that in [11] as follows to illustrate how an explanation table can summarize the distribution of T with respect to F. In Table 2.1, each row represents a flight that specifies an *ID*, the *day*, the *origin* airport, the *destination* airport, and whether the flight was *delayed*. In other words, the set F consists of day, origin, and destination, while T is a binary value that is 1 if the flight was delayed and 0 otherwise.

ID	Day	Origin	Destination	Delayed
1	Fri	SF	London	1
2	Fri	London	LA	1
3	Sun	Tokyo	Frankfurt	0
4	Sun	Chicago	London	1
5	Sat	Beijing	Frankfurt	0
6	Sat	Frankfurt	London	1
7	Tue	Chicago	LA	0
8	Wed	London	Chicago	0
9	Thur	SF	Frankfurt	1
10	Mon	Beijing	SF	0
11	Mon	SF	London	0
12	Mon	SF	Frankfurt	0
13	Mon	Tokyo	Beijing	0
14	Mon	Frankfurt	Tokyo	0

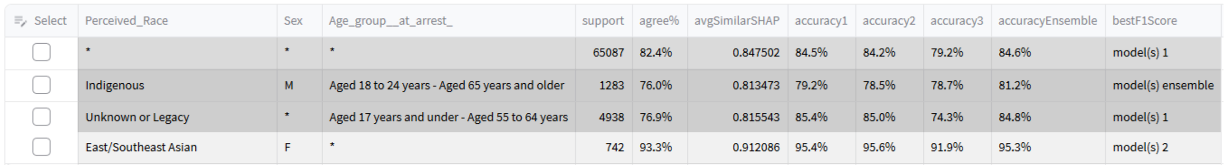
Table 2.1: An example flight delay dataset from [11] adapted for this chapter.

Day	Origin	Destination	Target
*	*	*	0.36
Mon	*	*	0.00
*	*	London	0.75

Table 2.2: An example explanation table for the dataset in Table 2.1.

Table 2.2 is a corresponding explanation table that consists of three different patterns. Every pattern in this table reports their target rate in the "Target" column; i.e., the percentage of tuples or datapoints that match the pattern with a positive target. The first pattern at the top of this table, an all-stars rule with the "*" value for all features, matches the entire dataset. Note how the target is 0.36; i.e., 36% of total flights were delayed. The second rule states that no flights on Monday were delayed, while the third rule states that 75% of flights bound to London were delayed. Thus, Table 2.2 specifically summarizes the distribution of flight delays in Table 2.1.

In an explanation table, the first pattern at the top is always an all-stars rule that matches the entire dataset. To determine the remaining patterns, we use the greedy algorithm described in [16] that iteratively selects patterns that provide the most information about the the distribution of T, until k patterns have been found. We quantify this information via *Kullback-Leibler* (KL) divergence [19], which measures the distance between the true and estimated probability distributions of T based on the explanation table. To estimate the distribution of T, the algorithm fits a maximum-entropy estimate of T based on the information contained in the patterns found so far. Thus, the information content for any candidate rule is the corresponding reduction in KL-divergence should the estimated maximum-entropy distribution be updated with the information reported in that candidate rule. Note that CAMEO allows the user to filter out low-support candidate rules via a minimum support threshold parameter, where support is how many tuples match that specific rule.



Select	Perceived_Race	Sex	Age_group__at_arrest_	support	agree%	avgSimilarSHAP	accuracy1	accuracy2	accuracy3	accuracyEnsemble	bestF1Score
☐	*	*	*	65087	82.4%	0.847502	84.5%	84.2%	79.2%	84.6%	model(s) 1
☐	Indigenous	M	Aged 18 to 24 years - Aged 65 years and older	1283	76.0%	0.813473	79.2%	78.5%	78.7%	81.2%	model(s) ensemble
☐	Unknown or Legacy	*	Aged 17 years and under - Aged 55 to 64 years	4938	76.9%	0.815543	85.4%	85.0%	74.3%	84.8%	model(s) 1
☐	East/Southeast Asian	F	*	742	93.3%	0.912086	95.4%	95.6%	91.9%	95.3%	model(s) 2

Table 2.3: Example explanation table generated by CAMEO for the strip-search arrestee dataset.

We now stress that the explanation tables generated by CAMEO use completely different targets from that in Table 2.2: we specifically engineer the targets in CAMEO's row-wise patterns for the purposes of model comparison. For example, Table 2.3 consists of four different patterns and is a smaller version of the full explanation table presented in Table 5.1. In this specific table, the target T is a binary value that denotes whether all models made the same prediction or not (i.e., prediction consensus). For the purposes of CAMEO, as T is a binary value, every pattern in Table 2.3 will report their target rate in the "agree%" column. Note that if T was a numeric measure attribute, we would instead have a "target" column with the actual numeric value itself (i.e., T is not a percentage).

In Table 2.3, observe how each row specifies the data subset, s_i, based on feature values. For the purposes of CAMEO, all columns before the "support" column correspond to the set of features. For example, the first pattern at the top of this table is the all-stars rule as the Perceived_Race, Sex, and Age_group__at_arrest_ columns for this row all have the "*" value; 82.4% of records for the entire dataset have a positive target. The second rule matches male Indigenous adults, where 76.0% of records have a positive target. Along similar lines, we can observe how the third rule matches individuals that are 64 and younger with unknown or legacy race, and the fourth rule matches individuals that are female East/Southeast Asians.

In the explanation tables generated by CAMEO, we also have more columns to report additional statistics for each subset. Each row consists of a *support* (how many datapoints match s_i), the *average similarity* of model explanations (SimilarSHAP, details in Section 4.1.1), the *prediction accuracies* for each model and a *majority voting ensemble* (the fraction of datapoints in s_i where the predictions correctly match the ground truth), and the *model* or *ensemble* with the best *F1 score* (a metric that measures precision and recall of a classification model). Note that a majority voting ensemble in this scenario is a soft voting ensemble model that predicts the class based on the largest summed probability from the constituent models. To give an example of reading these statistics from Table 2.3, we can observe how the third rule has a support of 4938 (i.e., matches around 7.6% of total datapoints), an avgSimilarSHAP of 0.815543 (i.e., the average similarity of model explanations in this data subset is less than the average similarity of model explanations for the entire dataset), and that the ensemble model actually has the best F1 score compared against the individual models.

All together, these patterns summarize the distribution of the target with respect to the feature set, which allows the user to identify unusual or surprising data subsets where the distribution of the target is different from what is expected.

Chapter 3

Related Work

In this chapter, we review related work similar to CAMEO that considers multiple models and furthers model understanding, highlighting their differences from our approach (see Section 3.1). We also review related work based on the XAI methods used in the technical implementation of CAMEO (see Sections 3.2 and 3.3).

3.1 Frameworks to Understand Multiple Models

Although no existing works explain multiple AI models simultaneously, solutions do exist in the literature that consider scenarios that involve more than one model.

3.1.1 Side-by-Side Comparisons

In 2016, Kahng et al. designed MLCube and MLCube Explorer [22] as a visualization tool that enables users to explore aggregate statistics and evaluation metrics over data subsets and interactively drill down into models. Its goal is to allow users to find interesting patterns between features and model results. However, [22] only visualizes the performances of up to two user-selected models side-by-side. For example, in part A of Figure 3.1, the user visualizes the performance of two different models and observes that the latter model has significantly improved accuracy over the other model for the subset user_age_group = 0. By selecting and drilling down into this subset (see part B), the user can observe interesting patterns between the tfidf_sim_query_title feature and model accuracy. Note that the user must manually define these subsets via relational selections using both data

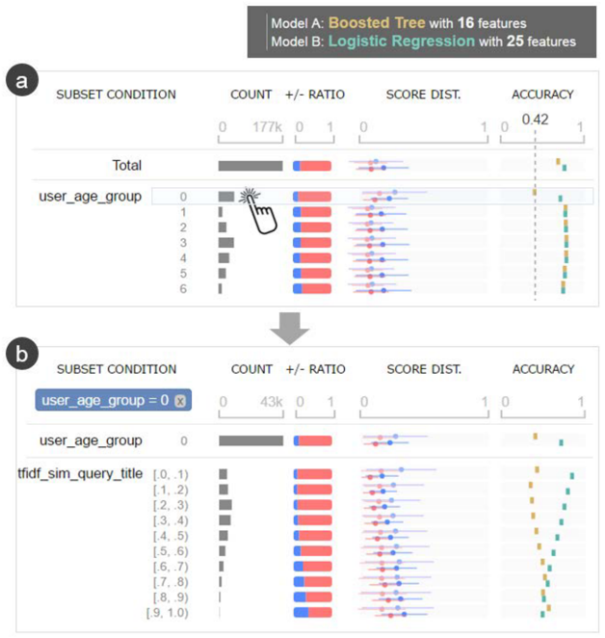
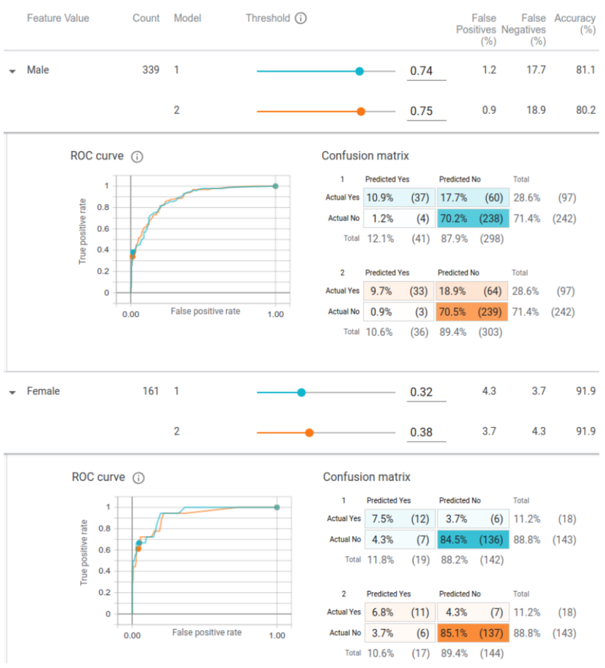


Figure 3.1: Comparing performance of two models via MLCube for user_age_group = 0 [22].

attributes and features. Thus, MLCube is unable to automatically find interesting subsets such as those with surprising performance differences between models.

In 2020, Wexler et al. proposed WIT [47] as an interactive tool that probes the behavior of models and visualizes their performance via confusion matrices (i.e., tabular summary that compares ground truth with model predictions) and scatter plots of datapoints to highlight correct and incorrect predictions. WIT can also further evaluate differences in performance and fairness by drilling into slices of the dataset based on feature values. For example, Figure 3.2 visualizes the performance of two different models over two different subsets: male and female, while outputting confusion matrices for both models. Similar to [22], WIT still manually compares individual models side-by-side; it only allows users to probe the input and output of models. Zhang et al. from Uber Technologies presented a similar work to WIT in Manifold [49], though their visualizations are preset and less flexible.

In 2023, Wang et al. proposed LFD [46], a framework to visually compare feature importances across two classifier models. Similar to CAMEO, LFD identifies data instances with disagreed predictions and leverages a feature-centric explanation method such as SHAP to compute feature importances. At a high level (see Figure 3.3), LFD compares two models A and B by first taking a dataset and feeding it into A and B to retrieve their


Feature Value	Count	Model	Threshold	False Positives (%)	False Negatives (%)	Accuracy (%)
Male	339	1	0.74	1.2	17.7	81.1
		2	0.75	0.9	18.9	80.2

ROC curve

Confusion matrix

1	Predicted Yes		Predicted No		Total	
Actual Yes	10.9%	(37)	17.7%	(60)	28.6%	(97)
Actual No	1.2%	(4)	70.2%	(238)	71.4%	(242)
Total	12.1%	(41)	87.9%	(298)		

2	Predicted Yes		Predicted No		Total	
Actual Yes	9.7%	(33)	18.9%	(64)	28.6%	(97)
Actual No	0.9%	(3)	70.5%	(239)	71.4%	(242)
Total	10.6%	(36)	89.4%	(303)		

Feature Value	Count	Model	Threshold	False Positives (%)	False Negatives (%)	Accuracy (%)
Female	161	1	0.32	4.3	3.7	91.9
		2	0.38	3.7	4.3	91.9

ROC curve

Confusion matrix

1	Predicted Yes		Predicted No		Total	
Actual Yes	7.5%	(12)	3.7%	(6)	11.2%	(18)
Actual No	4.3%	(7)	84.5%	(136)	88.8%	(143)
Total	11.8%	(19)	88.2%	(142)		

2	Predicted Yes		Predicted No		Total	
Actual Yes	6.8%	(11)	4.3%	(7)	11.2%	(18)
Actual No	3.7%	(6)	85.1%	(137)	88.8%	(143)
Total	10.6%	(17)	89.4%	(144)		

Figure 3.2: Comparing performance of two models via WIT by gender [47].

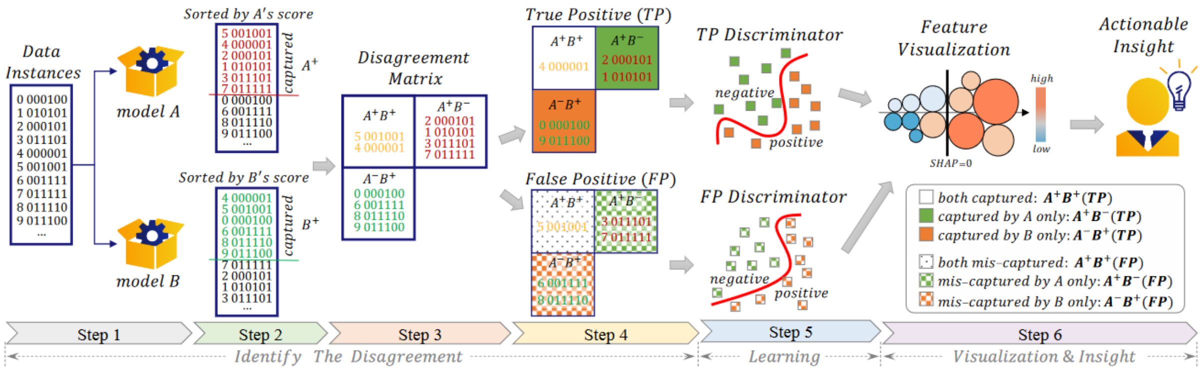


Figure 3.3: Process used in LFD to compare two models [46].

prediction scores for data instances (Step ①). For each model, LFD sorts these scores in decreasing order, setting a user-defined threshold as the score cutoff (Step ②). If scores are above this cutoff, then we consider them highly scored by the model. LFD takes these two sets of scores and joins them into a disagreement matrix (Step ③) where one cell corresponds to data instances being highly scored by both models (A^+B^+), and the other two cells correspond to data instances being highly scored by one model but not the other (A^+B^-, A^-B^+).

Using ground truth, LFD further divides the disagreement matrix into two matrices: one for true positives, and the other for false positives (Step ④). LFD then trains two discriminator binary classifiers to differentiate between A^+B^- and A^-B^+ instances from the true positive and false positive matrices, respectively (Step ⑤). Note that this step requires the user to manually propose features derived from the disagreed instances for this training dataset. LFD can then interpret the discriminators via SHAP to determine which feature in what condition is more important to one model over the other, allowing the user to gain more insight into the differences between A and B (Step ⑥).

To summarize, these existing approaches visualize the performance of two models side-by-side. However, CAMEO does not rely on side-by-side comparisons and instead leverages methods such as explanation tables to summarize model agreement and expertise which is novel and more concise.

3.1.2 ModelDiff

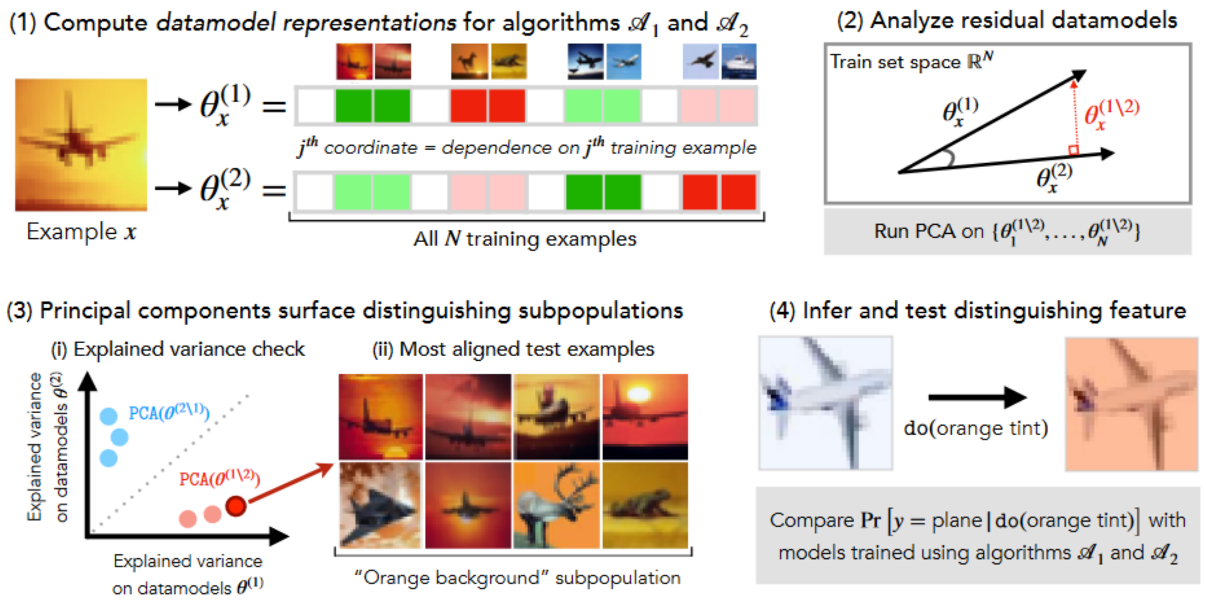

(1) Compute datamodel representations for algorithms $\mathscr{A}_1$ and $\mathscr{A}_2$
$\theta_x^{(1)} =$
j^{th} coordinate = dependence on j^{th} training example
$\theta_x^{(2)} =$
Example x
All N training examples
(2) Analyze residual datamodels
Train set space $\mathbb{R}^N$
$\theta_x^{(1)}$
$\theta_x^{(1\backslash 2)}$
$\theta_x^{(2)}$
Run PCA on $\{\theta_1^{(1\backslash 2)}, \ldots, \theta_N^{(1\backslash 2)}\}$
(3) Principal components surface distinguishing subpopulations
(i) Explained variance check
(ii) Most aligned test examples
Explained variance on datamodels $\theta^{(2)}$
PCA($\theta^{(2\backslash 1)}$)
PCA($\theta^{(1\backslash 2)}$)
Explained variance on datamodels $\theta^{(1)}$
"Orange background" subpopulation
(4) Infer and test distinguishing feature
do(orange tint)
Compare Pr [y = plane | do(orange tint)] with models trained using algorithms $\mathscr{A}_1$ and $\mathscr{A}_2$

Figure 3.4: Process used in ModelDiff to find distinguishing features between two models in terms of how they use training data to make predictions [47].

In 2023, Shah et al. presented ModelDiff [42], a framework that compares two supervised learning algorithms. Note that a learning algorithm is the whole process of mapping training datasets to AI models. At a high level (see Figure 3.4), ModelDiff first computes two sets of *datamodels*, one for each learning algorithm (Step ①). A datamodel for a test example x is a simple function that directly maps training data to predictions. More specifically, it is a linear function that takes any subset of the original training data (S') and predicts the output of models (i.e., those trained with the learning algorithm) on x should they be trained on S'.

For each test example x, ModelDiff then computes two *residual datamodels* to determine the difference in how the learning algorithms use training data (Step ②). The first residual datamodel represents a weighted combination of training examples that influences models trained with one of the learning algorithms on x after projecting away the component that influences models trained with the other learning algorithm, and vice versa for the latter residual datamodel. These two sets of residual datamodels enable ModelDiff to better understand how models trained with different learning algorithms differ at a per-example level.

To understand global differences in model behavior, ModelDiff uses a dimensionality-reduction method to distill residual datamodels into *distinguishing training directions* (i.e., set of weighted combinations of training examples) that generally influence predictions trained with one learning algorithm but not the other (Step ③). A user of ModelDiff can then look at these training directions to find test examples whose residual datamodels are most aligned with that direction to identify a set of subpopulations or *distinguishing features* that are specific to one learning algorithm and not the other. Finally, ModelDiff performs counterfactual experiments to verify whether these distinguishing features legitimately influence model behavior (Step ④).

To summarize, the main intuition behind ModelDiff is that models that use different features to reach their final prediction will rely on different training examples to make those predictions. ModelDiff is an orthogonal work to CAMEO because it focuses on helping model engineers improve the efficiency of training by finding groups of training examples that strongly influence predictions made on a subset of test examples for models trained with one learning algorithm but not the other.

3.2 Instance-Centric Explanations

Instance-centric explanations explain models in terms of surprising data subsets with respect to a given property. This section focuses on solutions that automatically find such data subsets.

In 2020, Chung et al. proposed SliceFinder [10] to find the top-k largest and most problematic subsets where an AI model underperforms compared against its average behavior. Their technique of model slicing ranks subsets of the test dataset based on how poorly the model performs over a given subset compared to the rest of the data. SliceFinder leverages statistical techniques such as training a decision tree to partition examples into slices before performing a Breadth-First-Search on the decision tree to find statistically significant and large enough subsets. Note that a decision tree-based model splits data multiple times according to cutoff values in the features, which creates different data subsets where each instance belongs to one subset (i.e., distinct rule-like groups). However, decision trees can be unstable depending on the training dataset and could potentially have a lack of smoothness. Thus, [10] also proposes a Lattice Search variant, where slices form a lattice that can then be searched to find results. In 2021, Sagadeeva et al. proposed SliceLine [36] that specifically focuses on improving SliceFinder's scalability and speed. At a high level, SliceLine leverages monotonicity properties of slice sizes, errors, and resulting scores (e.g., authors of [36] observed that slice sizes and errors are monotonically decreasing along a direct path in the lattice) to enable effective pruning.

In 2022, Pradhan et al. proposed GOPHER [30] to help explain bias in an individual AI model. GOPHER's explanations identify training data subsets responsible for model bias by scoring subsets based on their contribution to model bias and then outputting the top-k patterns that cause the most bias. Note that GOPHER derives bias from a fairness metric such as statistical parity (i.e., does the algorithm that classifies protected and privileged groups with the same probability fit within some threshold?). GOPHER also provides update-based explanations that use interventions to measure the effect of patterns in data that significantly promote bias by removing a subset of data (i.e., one believed to be the root of bias), before evaluating whether a model built on the remaining data has less bias.

To summarize, these existing approaches focus on the problem of model diagnostics. However, they can produce verbose outputs should subsets exceed a given error or bias threshold in their algorithm, thus motivating CAMEO's use of explanation tables instead. From the context of explaining individual models, in 2023, Esmaeilzadeh et al. proposed InfoMoD [14] which leverages explanation tables to tackle the problem of model diagnostics. At a high level, InfoMoD looks at how a model performs across different subsets of the data and points out surprising data subsets with performance indicators such as false positives and false negatives. Dadvar et al. similarly leveraged explanation tables in POEM [12] to find patterns that link somatic concepts such as shapes and colors to model predictions from convolutional neural network image classifiers. Dadvar et al. effectively reduce the problem of unstructured data to that of structured data by 1) transforming image data to tabular data, and 2) using simpler models such as decision trees and explanation tables to then explain black-box outputs.

3.3 Feature-Centric Explanations

Feature-centric explanations explain models in terms of features and their outcomes. We can generally categorize them as saliency-based methods, counterfactual-based methods, and hybrid methods that combine the two.

3.3.1 Saliency-based Methods

Feature attribution methods tend to quantify how responsible input features are for model outcomes. In 2016, Ribeiro et al. presented LIME [33], which relies on the technique of approximating a black-box model with a simpler and more interpretable model to help explain it. This well-known method first generates random neighboring instances around dataset samples by performing small perturbations. LIME then creates a surrogate linear model to simulate the local behavior of the black box in question, using its feature importance values to explain the model decision. This allows the user to identify what is the most important attribute around the datapoint of interest. Note that one drawback of LIME is its assumption of linearity, which makes it unsuitable for complex time series data [29].

In 2017, Lundberg and Lee proposed the *SHapley Additive exPlanation* method [26] to generate local explanations (SHAP) without a linearity assumption, motivating its use in CAMEO. Inspired by classic game theory, this well-known method computes the importance of features based on their additive Shapely values. At a high-level, SHAP considers each feature as a player in a m-person game to fairly distribute feature contributions to the model output by comparing the output of the same model when a feature is present or not present. It can then assign a single numeric score for each input feature to a model to approximate the relative importance of input features on model output.

3.3.2 Counterfactual-based Methods

Counterfactuals-based methods identify minimal perturbations to input features that change the model output. The SEDC method by Martens and Provost [27] is one of the first works to formally introduce the concept of counterfactuals. In 2022, Geng et al. [18] proved that rule-based explanations (i.e., those that identify an ideally small set of features whose values likely lead to the same outcome [8, 34]) and counterfactual-based explanations are duals to each other. [18] also presented an approach to generate local rule-based explanations using a counterfactual-based explanation system.

In 2018, Wachter et al. [45] proposed counterfactuals as an optimization problem, using a distance term to ensure their counterfactuals are close to the original instance. In 2020, Mothilal et al. proposed *Diverse Counterfactual Explanations* (DiCE) [28] which allows the user to tune a diversity hyperparameter to generate multiple counterfactual explanations for each observation. For example, they provide a genetic algorithm variant that is model-agnostic. Note that diversity is how different the resulting counterfactuals are from each other so that the system can provide multiple different explanations for a single query case (i.e., different users may find different explanations helpful). DiCE also enables the user to add constraints on counterfactual generation such as specifying what features can be varied and their permissible ranges.

In 2022, Brughmans et al. presented *Nearest Instance Counterfactual Explanations* (NICE) [7] which can generate counterfactual explanations for tabular data by exploiting information from the nearest unlike neighbor to speed up the search process. They also propose four different versions to optimize explanations in terms of sparsity, proximity, and plausibility. Proximity is how close the query is to the generated counterfactual instance, which is typically measured via distance metrics. Sparsity is if (ideally) the generated counterfactual instance only modifies a small number of features. Plausibility is how representative the generated counterfactual is of the underlying data distribution (i.e., can have counterfactuals with high proximity but are out-of-distribution, which corresponds to low plausibility).

In 2021, Schleich et al. proposed GeCo [38] that focuses on optimizing counterfactual analysis for a given instance of interest to generate explanations quicker at a interactive speed. Note that GeCo also focuses on local explanations to explain a single prediction while keeping explanations plausible. For example, in Figure 3.5, GeCo found a counterfactual for a decision tree model that only perturbs the CapitalGain value. For the neural network model, GeCo found a counterfactual that suggests a realistic increase in education and age. Indeed, GeCo uses its own language PLAF to help define constraints for plausibility and feasibility (e.g., filter out individuals who are older than 80).

At a high-level, GeCo efficiently explores the space of counterfactual explanations for tabular data via a genetic algorithm that prioritizes counterfactuals with fewer feature changes. This genetic algorithm first defines an initial population of candidates based on a

	Age	Education	Occupation	CapitalGain	CapitalLoss	Hours/week	Gender	Prediction
$\boldsymbol{x}$	49	School	Service	0	0	16	F	Bad
			Decision Tree					
GeCo	49	School	Service	**4,787**	0	16	F	**Good**
MACE	49	School	Service	**4,826**	**20**	16	F	**Good**
SimCF	49	School	Service	**7,688**	**1,380**	16	F	**Good**
			Neural Network					
GeCo	**53**	**Masters**	Service	–	–	16	F	**Good**
DiCE	**54**	**PhD**	**BlueCol**	–	–	**40**	**M**	**Good**

Figure 3.5: Example counterfactuals generated by GeCo and other methods [38].

given instance of interest, before it iteratively selects the fittest counterfactual candidates in the population and generates new candidates via mutation and crossover operations on selected candidates. GeCo then repeats this iterative process until it reaches a sufficient number of examples. Note that GeCo uses the following optimization techniques to speed up this iterative process. First, it does not store all possible candidates as it groups the current population by the set of features that differ from the given instance of interest. GeCo represents this subpopulation as a single relation whose attributes are only that set of features. Second, if GeCo also has access to the model's code, it can translate the model into a simpler model that can then be used for any candidates with that specific set of features [38].

To summarize, many counterfactual methods exist in the literature. However, we specifically leverage DiCE in CAMEO because of its model-agnostic variant, the diversity of counterfactuals, as well as its support for defining which features can be varied for outputted counterfactuals.

3.3.3 Hybrid Methods

Albini et al. designed CF−SHAP [2], a variant of SHAP that combines feature attribution with counterfactuals. The intuition behind CF−SHAP is that Shapley values can identify features that strongly impact the output, but they fail to identify which input values are associated to output values of interest (i.e., how can we intervene to change the adverse outcome?). Thus, Albini et al. use a set of counterfactual points instead as the background dataset to compute Shapely values because they believe feature attributions alone do not provide enough actionable guidance on how to alter features to change the prediction of a model. At a high level, this effectively allows CF−SHAP to provide derived trends that compare features with the counterfactual distribution to better convey relevant information to the user. For example, in Figure 3.6, the red and blue regions represent areas of the feature space where the decision is either adverse or not. Colored arrows and scatter points represent the direction of the Shapley values vector and the background dataset used for their computation, respectively. In Figure 3.6, note how the leftmost chart (i.e., background set is a training set where samples are predicted of being of the same class of the input) suggests Feature A negatively contributed to the model output but provides no actionable insight. This is in contrast with the rightmost chart (i.e., background set is a set of counterfactuals) because now the background distribution depends on the input X so that X is being compared to samples similar to answering the intended contrastive question.

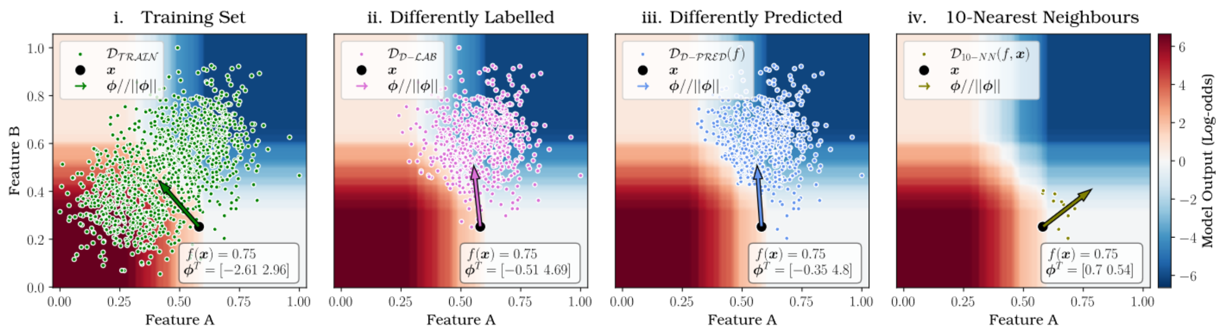


Figure 3.6: The effect of different choices of background dataset on Shapley values of the same input with the same model [2].

Chapter 4

Methodology

In this chapter, we discuss the space of comparison operations for multiple models as well as the components that make up CAMEO's architecture.

4.1 Problem Formulation

The concepts of model consensus and expertise are core to CAMEO. Consider a test dataset D and a set of n pre-trained models $M = \{m_1, ..., m_n\}$ for a classification problem[1] where $n \geq 2$. For example, suppose we have models trained to predict if a strip search happened during an arrest event by the Toronto Police. Note that the models use the same features, and may be of different types or instances of the same model type but trained using different hyperparameters. In CAMEO, we categorize the space of model comparisons along two axes: 1) *how* to define model consensus and expertise, and 2) *where* we find expertise and consensus (or discord) amongst the models.

4.1.1 The How Axis

In a machine learning pipeline, engineers responsible for model selection might not necessarily have access to resources such as training datasets and the model's code. Thus, we desire comparison operations that do not rely on the internal logic of models and are

[1] CAMEO can be used for regression by effectively converting it into a classification problem (e.g., defining a threshold for consensus).

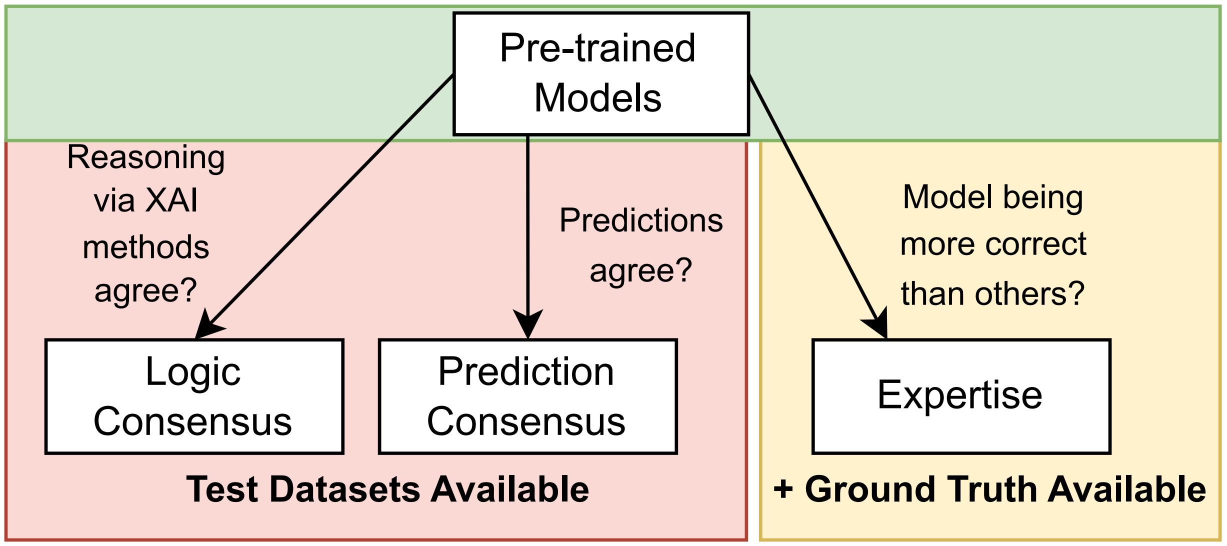


Figure 4.1: Initial space of model comparisons with only test datasets available.

non-invasive. For the *how*, we consider three different measures: *prediction* consensus, *logic* consensus, and *expertise*.

We categorize these measures based on the resources available to the engineer (see Figure 4.1). For example, if they only have unlabeled test datasets and pre-trained models (i.e., the bare minimum resources), then we can still check whether models agreed or disagreed on predictions (i.e., prediction consensus). If the engineer has additional access to feature importance scores via existing XAI methods, then they can check whether conflicting models paid attention to different features during inference (i.e., logic consensus). Finally, if the engineer has further access to ground truth, then they can check whether a model is more correct than others to find situations where a model is a relative expert (i.e., expertise). In Chapter 6, we envision an expanded space of model comparisons as future work such as users having access to more resources and being able to define their own comparison operations.

Regarding consensus, suppose we generally express consensus as a Boolean expression that is *true* if all models in M agree on some criteria, and *false* otherwise. For determining prediction consensus, we check if models made the same prediction for a given instance. As an example, if all models agreed that a strip search occurred for a given arrest event, then we would have prediction consensus. This measure evaluates the robustness of models over various contexts.

To determine logic consensus when we only have access to test datasets, we leverage the feature attribution method SHAP [26] previously discussed in Section 3.3.1 to compute approximate Shapley values. We do this for every instance in the test dataset, obtaining a score for each feature based on its importance for this prediction. This gives a vector for each model and prediction. For each component in this vector, the magnitude of the score determines how much that feature influenced the model output, and the sign indicates if that feature influenced the prediction in a positive or negative way; i.e., the outcome is more or less likely to occur, respectively.

Our intuition here is that Shapley feature contribution values quantitatively describe knowledge embedded in a model's prediction logic. With this intuition, we can now define two variants of logic consensus. The first variant is straightforward as it is a Boolean that indicates whether all models have the same highest-scoring features according to the explanation vectors. This Boolean expression checks whether models agreed on the most important feature contributor for a given instance. We denote the more fine-grained second variant, SimilarSHAP, as follows: we first calculate the centroid for the set of explanation vectors, one for each model, before calculating one minus the average normalized Eucledian distance between the explanation vectors and the centroid. SimilarSHAP is thus a range between zero and one, where values closer to one indicate greater similarity among the decision-making processes. This second variant is a numerical value that measures whether models paid attention to the same features when making their decisions for a given instance.

These consensus measures allow us to drill down and isolate potential reasons for agreement or disagreement: if the models made different decisions, did they use different logic to arrive at their decisions? By capturing the degree of consensus, we can identify insightful consensus patterns over D according to the evaluated consensus measures.

Regarding *expertise*, the intent of this comparison operation is to identify situations where a model is unusually more correct than the rest of the candidates. We believe there exist a few options to define this expertise. One option is to express expertise as a numerical value that measures the performance difference between a model of interest and the other best-performing model for a given instance on a fractional level. To be more specific, we identify the subgroup that a given instance matches (e.g., an Indigenous male that is younger than 18), and then compute the difference between the overall accuracy of the model of interest for that subgroup and the overall accuracy of the other most-accurate model for that subgroup.

Another option to express expertise is as a binary number of length n. For a given instance, we would use 1 to represent a model being correct and 0 otherwise. For example, if n is 3, a value of 100 would indicate that the first model was correct but the second and third models were not. Note that for the purposes of this thesis we define expertise via the first option because 1) the second option could return expertise patterns where a model of interest is compared against non-important models such as the worst-performing model, and 2) we can precompute the overall accuracies beforehand to help optimize performance. Thus, this measure allows us to identify data subsets where a model notably outperforms (or underperforms) relative to the other models in M.

4.1.2 The Where Axis

The *where* refers to the types of patterns: we consider row-wise data subsets and column-wise feature intervals where the models predominantly agree or disagree (details in Sections 4.3 and 4.4). CAMEO also allows users to drill into interesting subsets to perform counterfactual analysis, in which CAMEO identifies minimal perturbations of the feature values that would make conflicting models agree or consistent models disagree (details in Section 4.5). Note that counterfactual analysis has traditionally been used in context of individual models, but we are the first to apply it to multiple models.

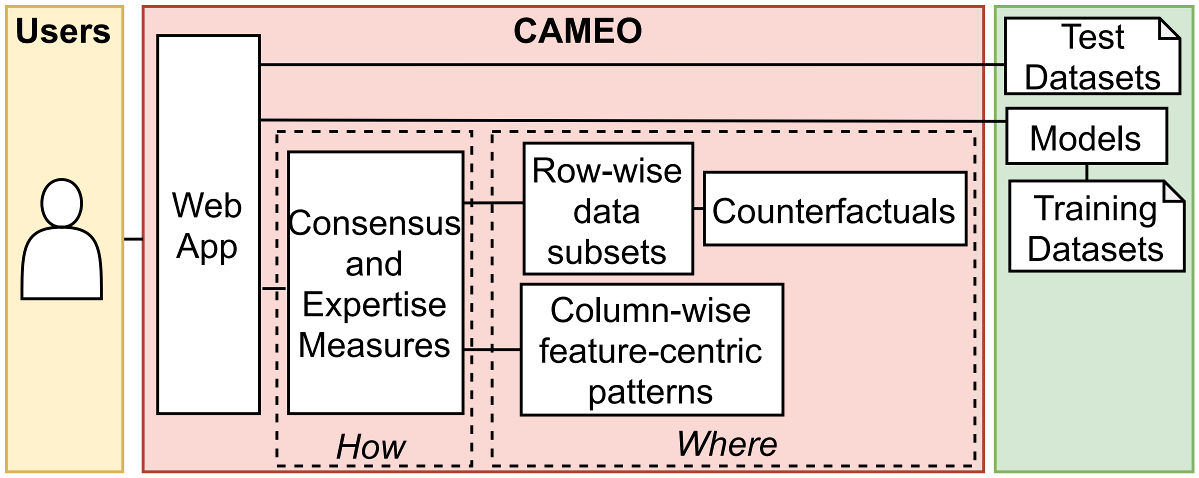
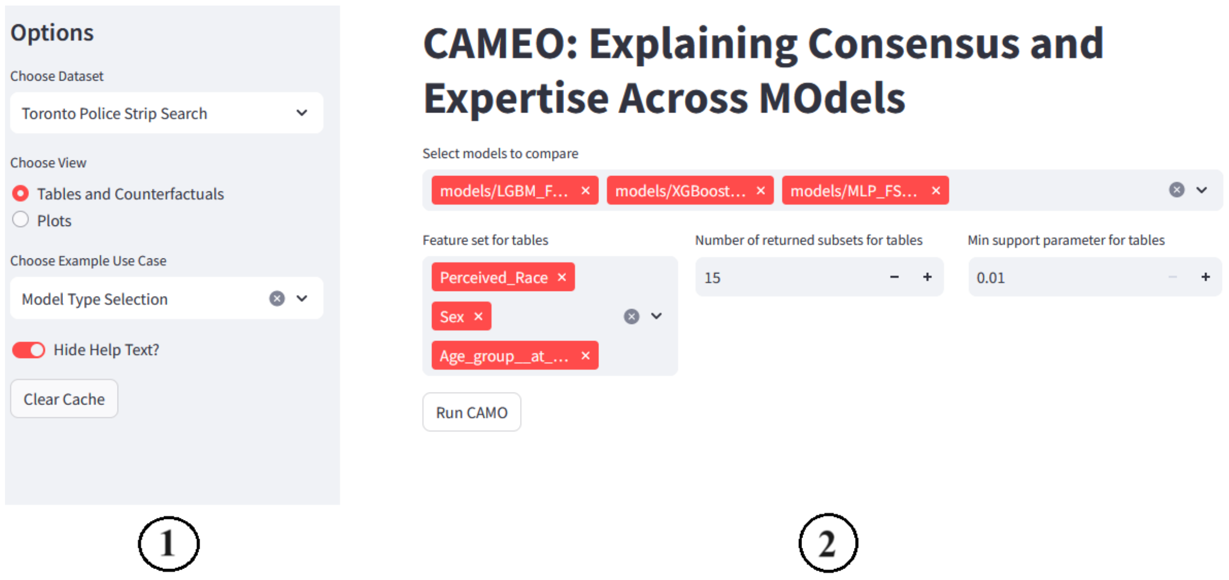


Figure 4.2: An architectural overview of CAMEO.

4.2 System Overview

Figure 4.2 summarizes the architecture of CAMEO. CAMEO's current implementation is as an interactive Python web application (details in Section 4.6) that enables users to gain further insights into their models via row- or column-wise patterns. Row-wise patterns are useful for finding subgroups of datapoints with unusual rates of model consensus and expertise; column-wise patterns are useful for visualizing intervals within feature domains with unusual degree of consensus or discord. As illustrated in Figure 4.3, the web interface has two main panels: *options* on the left, and *explanations* on the right. The *options panel* selects a dataset and changes the view to either a row- or column-wise method (Step ①). These methods are orthogonal to each other; for example, we do not need to generate instance-centric patterns before generating feature-centric patterns. On the *explanations panel*, the user can select the models and configure various settings such as the features to use for generating data subsets (Step ②). Note that users can choose an example use case (i.e., model type selection or hyperparameter tuning) in the options panel to automatically fill out the fields in the explanations panel.



Choose Dataset: Toronto Police Strip Search

Choose View: [✓ Tables and Counterfactuals] Plots

Choose Example Use Case: Model Type Selection

Select models to compare: models/LGBM_F... models/XGBoost... models/MLP_FS...

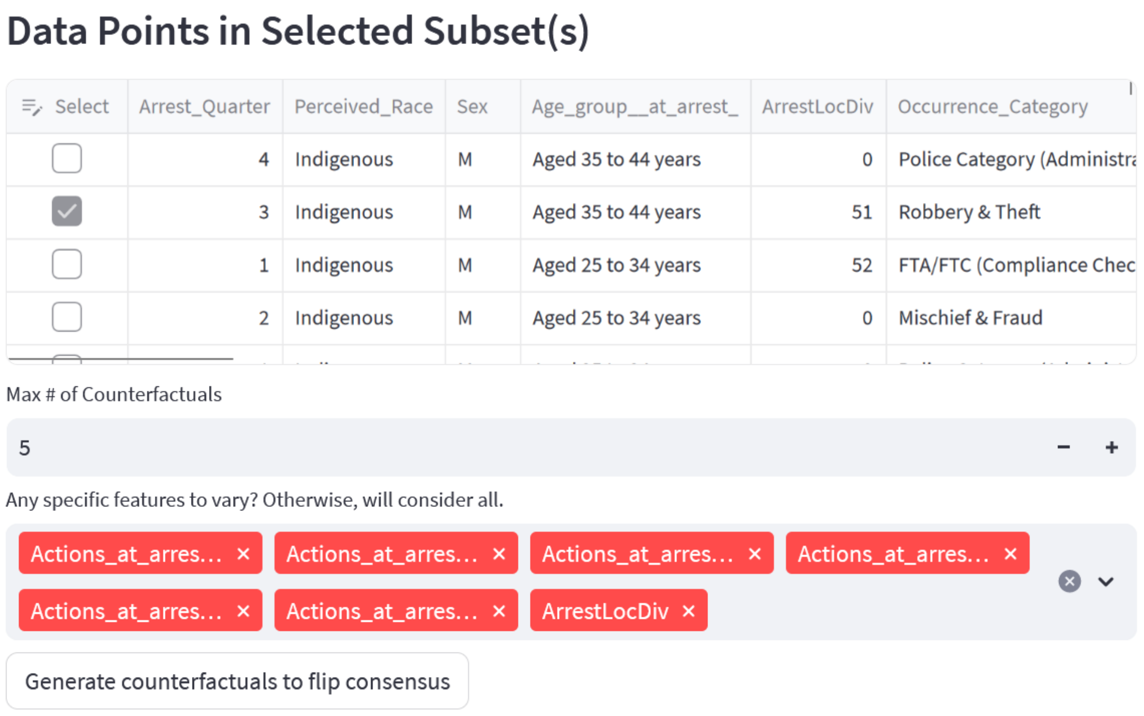
Feature set for tables: Perceived_Race, Sex, Age_group__at_...

Number of returned subsets for tables: 15

Min support parameter for tables: 0.01

Figure 4.3: Interface of CAMEO before a run.

After pressing the *Run* button to generate the patterns, if the *Tables and Counterfactuals* view is selected, the user can also select one of the generated patterns (see Table 5.1), and drill down into individual datapoints to generate counterfactuals (see Figure 4.4).



Data Points in Selected Subset(s)

Select	Arrest_Quarter	Perceived_Race	Sex	Age_group__at_arrest_	ArrestLocDiv	Occurrence_Category
☐	4	Indigenous	M	Aged 35 to 44 years	0	Police Category (Administra
✓	3	Indigenous	M	Aged 35 to 44 years	51	Robbery & Theft
☐	1	Indigenous	M	Aged 25 to 34 years	52	FTA/FTC (Compliance Chec
☐	2	Indigenous	M	Aged 25 to 34 years	0	Mischief & Fraud

Max # of Counterfactuals

5

Any specific features to vary? Otherwise, will consider all.

Actions_at_arres... × Actions_at_arres... × Actions_at_arres... × Actions_at_arres... ×
Actions_at_arres... × Actions_at_arres... × ArrestLocDiv ×

Figure 4.4: CAMEO allows the user to drill down into individual datapoints to generate counterfactuals.

4.3 Explanation Tables for Row-Wise Data Subsets

To find data subsets with unusual rates of model consensus and expertise, CAMEO generates three explanation tables as described in Chapter 2, where each table defines the target t based on the *how* measures described in Section 4.1.1. For example, prediction consensus would be a Boolean value that indicates whether all models made the same prediction, while expertise would be a number that indicates the performance difference between our model of interest and the other best-performing model according to their accuracies. The patterns in these explanation tables allow users to understand model consensus and expertise for key subgroups of datapoints based on the set of features chosen for subsetting these groups. For example, users can choose to return subsets where the pattern feature set only contains protected attributes (e.g., age). Possible insights could be that model m_i handles some combinations of protected attributes better than other models (e.g., predicting strip searches done on Indigenous arrestees), that an ensemble handles some subset(s) better than individual models, or that models are confused about particular subset(s). In the latter case, one might need to consider more training data of that type for training newer models and/or different model types to run with CAMEO.

As described in Chapter 2, we include a column in the explanation table that corresponds to the prediction accuracy of a soft majority voting ensemble. Note that this thesis does not treat model ensembles [50], a popular machine learning approach that aggregates the predictions of multiple pre-trained models to ideally achieve better predictive performance than individual models, as a first-class citizen. However, helping the user to understand the relative strengths and weaknesses of different models could assist ensemble construction because different models are typically used as constituent members to better exploit the differences in prediction errors (e.g., averaging predictions to ideally improve performance). In Chapter 6, we envision future work that focuses more on ensembles in context of our space of model comparisons.

4.4 Feature Plots for Column-Wise Feature-Centric Patterns

CAMEO can use feature plots to display intervals within feature domains with unusual degree of consensus or discord. We show an example in Figure 5.1. The x-axis corresponds to the values of a feature selected by the user (here, the age group of an individual who was arrested by police) and the y-axis captures the similarity of model explanations based on SimilarSHAP (here, the class label denotes whether the arrested person was strip-searched). The plots use two different colors to visualize consensus.

These feature plots enable users to understand how consensus changes over column-specific feature values, so they can further investigate intervals with concerning feature values. For example, we observe in Figure 5.1 how models with diverging predictions generally have dissimilar explanations, and models with converging predictions have similar explanations. However, some datapoints do not appear to follow this trend (i.e., there exist intervals where models agree on prediction, but disagree on decision making, or vice versa), most notably those that belong to the age groups between 18 and 54. Note that in the plots displayed to the user, we add horizontal random noise to datapoint positions for features that are discrete and have relatively few unique values to help address overplotting.

To assist further investigation, CAMEO can also generate SHAP feature importance plots for individual models (see Figure 4.5) as well as a global feature importance plot (see Figure 4.6). Note that in these figures, the y-axis corresponds to the features while the x-axis captures the feature's absolute average impact on model output. For example, in Figure 4.6 (i.e., a multiple-bar plot with one bar type for each model), the global importance of each feature is taken to be the mean absolute SHAP value for that feature over all the given samples so that users can easily see the most important features as a whole for models.

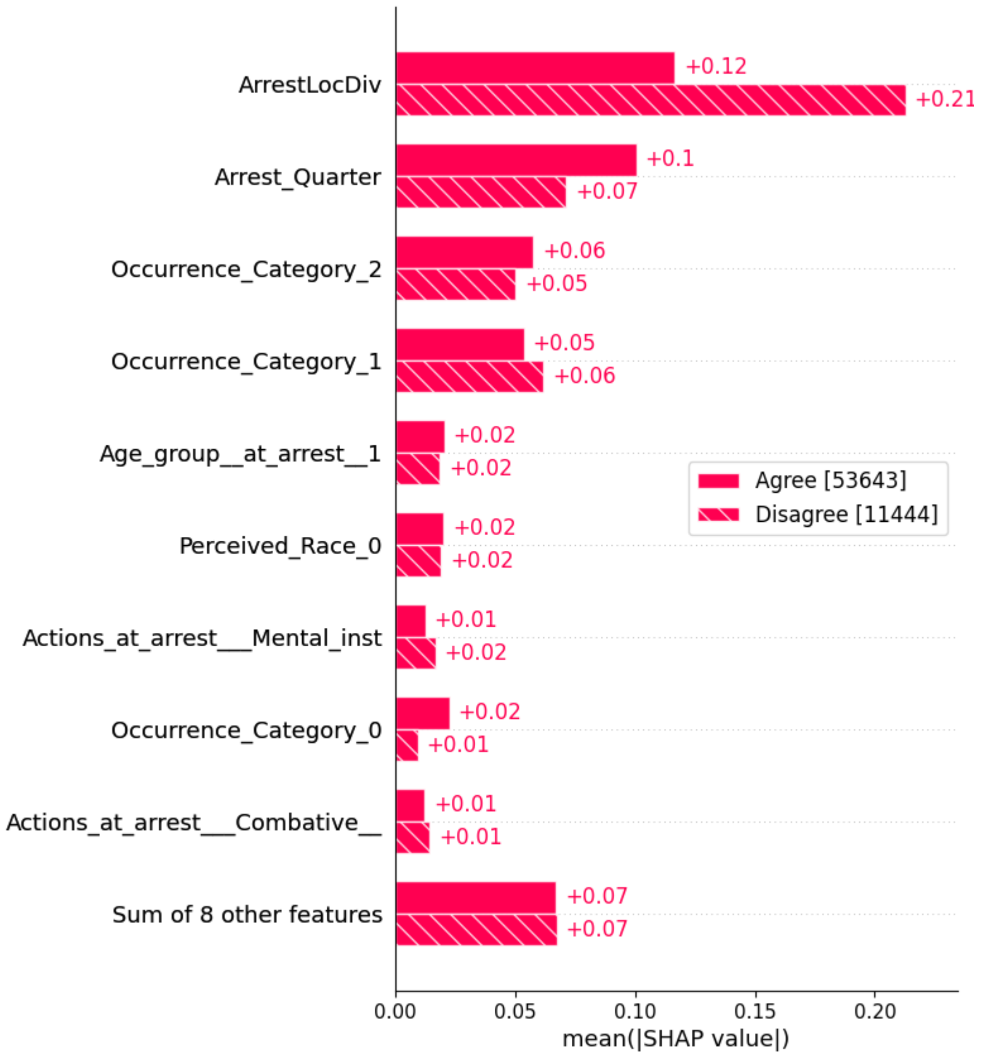


Figure 4.5: An example feature importance plot for a specific model that displays feature importance based on prediction consensus.

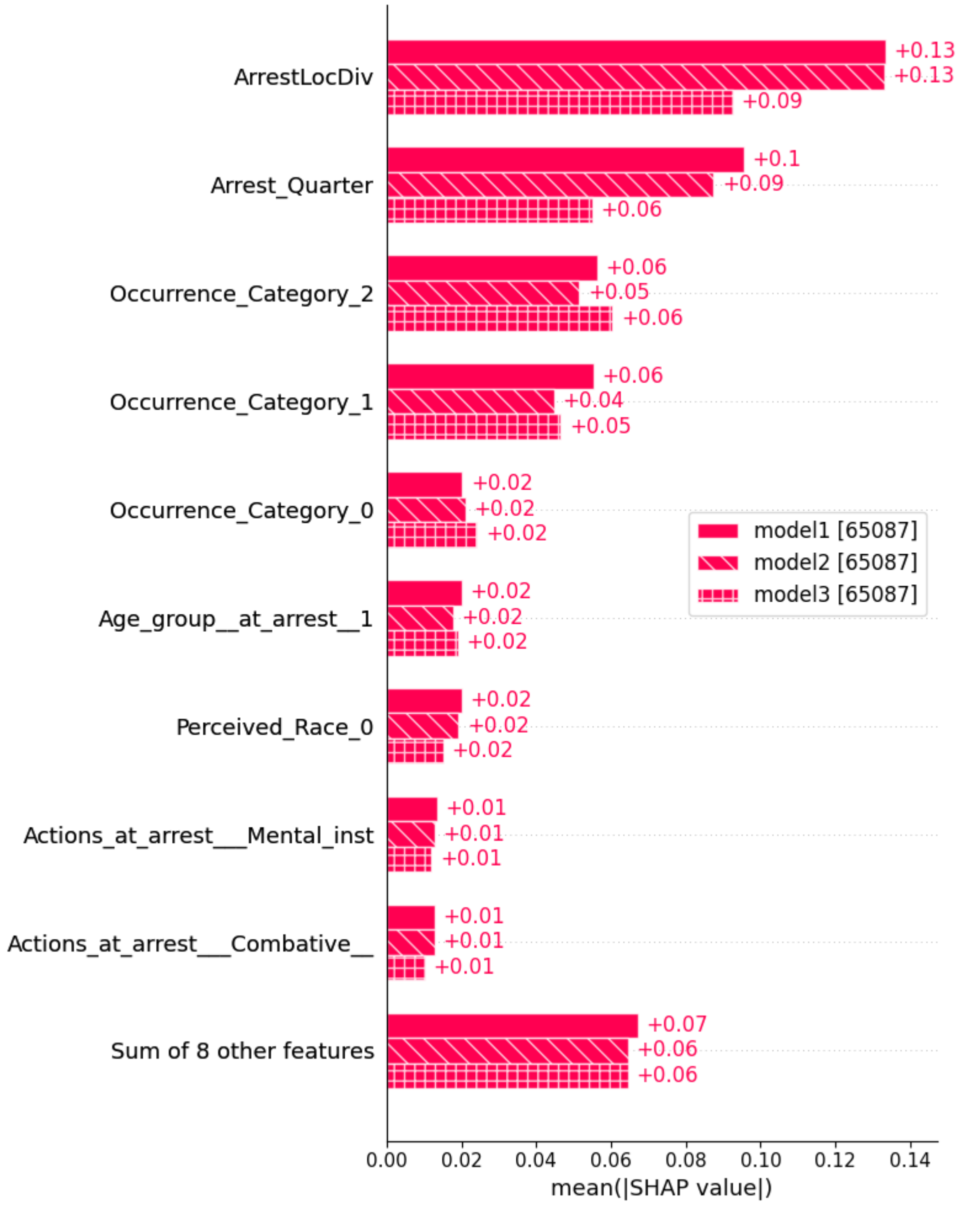


Figure 4.6: An example global feature importance plot for models.

4.5 Feature-Centric Counterfactuals

After CAMEO produces row-wise patterns as described in Section 4.3, the user can select specific data subset(s) to raise a second table that contains the datapoints that belong to these subsets. We show an example in Figure 4.4. Within this second table, the user can select any datapoint to generate counterfactuals that flip the consensus. For example, a counterfactual to flip discord to consensus for the selected datapoint in Figure 4.4 is to perturb ArrestLocDiv from 51 to 11 (see Table 5.3). This helps users to understand the decision boundary. If the models did not achieve prediction consensus, then a counterfactual explanation is a minimal perturbation of feature values such that all models make the same prediction. Likewise, for consistent models that achieved prediction consensus, a counterfactual explanation is a minimal perturbation of feature values such that at least one model makes a different prediction.

To produce multiple counterfactuals for a given datapoint, we leverage the model-agnostic DiCE [28]. At a high level, DiCE formulates an optimization problem (i.e., similar to finding adversarial examples) and then uses a genetic algorithm to find the best counterfactuals close to the initial point: the algorithm converges relatively quickly to promote diverse counterfactuals. CAMEO also allows the user to select which features can be varied to ensure feasibility. This is because we want counterfactuals that still belong to their drilled down subset; e.g., if our subset is specific to Indigenous adults, then the counterfactual should not change race.

4.6 Implementation Details

We implemented CAMEO primarily in Python, leveraging existing libraries to build AI models (e.g., scikit−learn, TensorFlow), handle the front-end (e.g., Streamlit), and efficiently calculate SHAP values (e.g., FastSHAP [21]). We obtained the C++ explanation table code from the original authors and then modified it to 1) support a numerical target attribute by adapting the algorithm described in SIRUM [15] for C++, 2) introduce a minimum support threshold for patterns with feature numerical ranges, and 3) support both positive and negative numerical target values by normalizing measure and estimate values before calculating overall information gain for these scenarios.

Chapter 5

Experiments

In this chapter, we present multiple curated use cases with various models to demonstrate how users can gain insights into their models.

5.1 Datasets

To help demonstrate its applicability for various scenarios, CAMEO enables users to explore various datasets and produce explanations for their own inquiries. We preload CAMEO with pre-trained models for these datasets; however, for the purposes of this thesis, we only use the strip-search arrestee dataset when describing the use cases in Section 5.2.

5.1.1 Toronto Police Strip Search Arrest Events

This dataset includes over 65,000 arrest events between 2020 and 2021 with protected demographic attributes (e.g., perceived race of the arrestee, gender, and age group) and a binary outcome denoting whether the arrest involved a strip search; i.e., removal of some/all clothing before visual inspection of the body. Given the issue of systematic racial disparities in policing, the Toronto Police Service has curated this dataset in the hopes of identifying, monitoring, and reducing potential systemic racism and racial bias in their processes [43]. Our aligned goal for this dataset is therefore to identify a model or ensemble that best handles desired minority subgroups.

5.1.2 NYC Flight Departure Delay

This dataset includes over 300,000 flights that departed from New York City, for the years 2019 to 2023, to destinations in the United States, Puerto Rico, and the American Virgin Islands. Some attributes include air travel data from the US Bureau of Transportation Statistics (e.g., flight date, carrier, origin airport, count of traffic operations at the time, etc.) and weather data derived from Meteostat (e.g., flag for poor weather such as heavy hail). The binary outcome denotes whether the flight was delayed, which we define as a flight that departs at least 15 minutes after its scheduled time.

Besides disrupting travel plans, flight delays can negatively impact the productivity of airlines and airports in terms of reputation, efficiency, and economy. Small local delays in airports can also potentially cascade into network-wide congestion for multiple airports [48]. Unfortunately, the flight delay problem appears to be receiving a lot of national attention lately: numerous passengers are complaining about inadequate compensation for disrupted travel plans due to delayed or canceled flights. According to Lamb et al. who studied the social and emotional perspectives of passengers during the COVID-19 pandemic, trust issues that involve COVID-19 and how airlines conduct their operations with staff shortages, cost-cutting, scheduling, and flight loads contribute to negative passenger perceptions [24]. However, findings from [24] also show that passengers are more likely to trust airlines or airports should they be better informed about their flights. Our aligned goal is therefore to identify a model or ensemble that best handles the unusual data subsets found by CAMEO.

5.1.3 Pima Indian Diabetes

This dataset includes hundreds of patients with various attributes to help determine type 2 diabetes in women of Pima Indian heritage. The binary outcome denotes whether the patient has type 2 diabetes. Some attributes in this dataset include BMI (body mass index), a function for diabetes pedigree (risk of type 2 diabetes based on family history where larger values indicate higher risk), age, insulin (2 hour serum insulin), skin thickness (skinfold thickness of triceps), blood pressure (diastolic blood pressure), glucose (concentration after a 2 hour oral glucose tolerance test), and number of pregnancies. Our aligned goal for this dataset is therefore to identify a model or ensemble that best handles sensitive subsets well (e.g., youth with a family history of diabetes).

5.1.4 S&P500

This dataset includes over 3,900 records dating back from September 2008 to September 2023 that describe the behavior of the S&P500 benchmark for the US market. The binary outcome of a record denotes whether the index will be a buy or sell the next day. Some attributes in this dataset include date, index values (e.g., open, close, volume, etc.), macro-factors (e.g., returns for Nasdaq, Dow Jones, etc.), and technical indicators (e.g., MACD and RSI100). Our aligned goal is therefore to identify a model or ensemble that best handles unusual data subsets found by CAMEO.

5.2 Use Cases

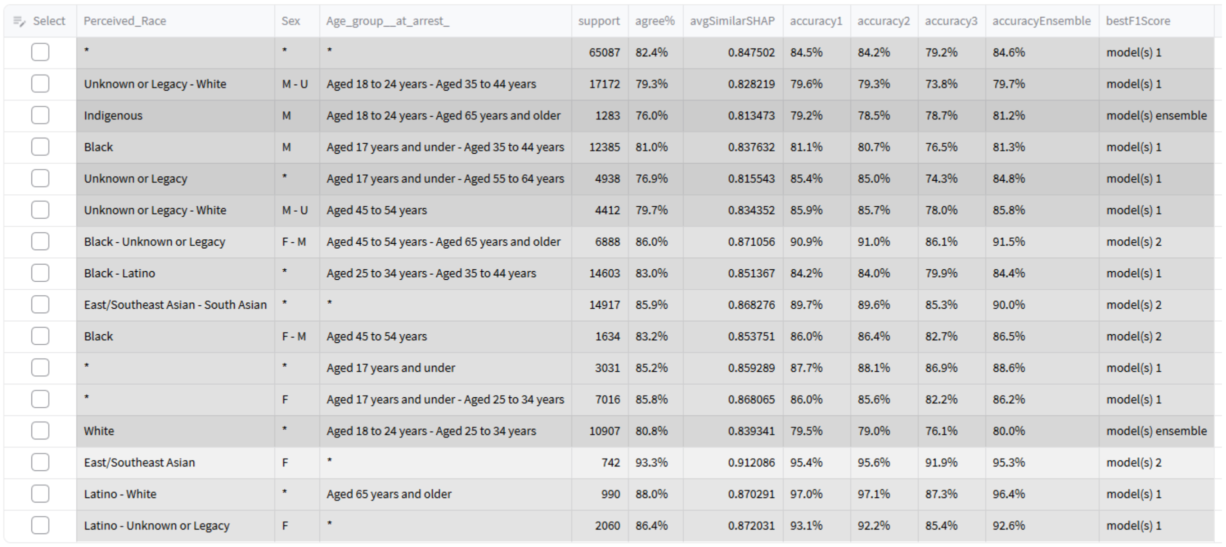
In this section, we discuss the following curated use cases that use strip-search arrestee data from the Toronto Police Service:

1. Model Type Selection: CAMEO can identify unusual consensus patterns for a set of various pre-trained model types. For example, CAMEO found subsets such as Indigenous male adults where all model types are confused. To address this issue, a user might need to consider more training data or different model types.
2. Hyperparameter Tuning: CAMEO can identify unusual consensus patterns for a set of pre-trained models with varying hyperparameters. For example, CAMEO found a particular hyperparameter that results in the model handling some combination of protected attributes, such as female East/Southeast Asians, better than other hyperparameters.
3. Finding Model Expertise: CAMEO can identify unusual expertise patterns for a set of pre-trained models. For example, CAMEO allowed us to learn that a particular model was a relative expert in handling key subgroups such as male East/Southeast Asian young adults.

Note that for logic consensus, we specifically describe only the first variant for simplicity. As mentioned in Section 4.1.1, the first variant is a Boolean expression that checks whether models agreed on the most important feature contributor for a given instance.

5.2.1 Use Case #1: Model Type Selection

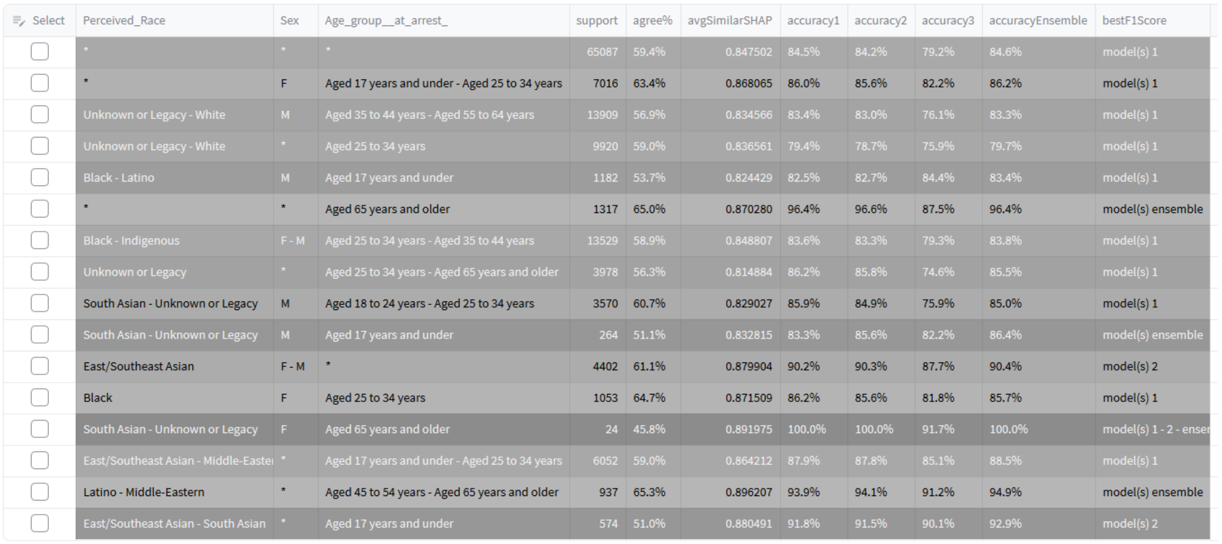
Consider the following pre-trained models: a light gradient boosted machine (LightGBM) [23], an extreme gradient boosting (XGBoost) model [9], and a multi-layer perceptron (MLP) [35]. We choose these model types as they are all popular machine learning algorithms for classification tasks in industry. As illustrated in Figure 4.3, we start by running these models with the *Tables and Counterfactuals* view, select three protected attributes (*perceived race*, *sex*, and *age group at arrest*) as the pattern feature set for explanation tables, and then select 15 as the number of patterns to return and 0.01 as the minimum support threshold. From the generated Tables 5.1 and 5.2 that contain all the consensus-related statistics described in Chapter 2, we observe that the table for logic consensus has a noticeably lower average consensus compared with the table for prediction consensus (59.4% versus 82.4%). According to these tables, models are more likely to achieve prediction consensus for female East/Southeast Asians compared against male Indigenous adults (93.3% from row 14 and 76.0% from row 3 in Table 5.1). Models are also more likely to achieve logic consensus for Latinos and Middle-Eastern adults that are 45 and older (65.3% from row 15 in Table 5.2) and for Black females between the ages of 25 and 34 (64.7% from row 12), compared against senior females who are South Asian, or have unknown or legacy race (45.8% from row 13).



Select	Perceived_Race	Sex	Age_group__at_arrest_	support	agree%	avgSimilarSHAP	accuracy1	accuracy2	accuracy3	accuracyEnsemble	bestF1Score
☐	*	*	*	65087	82.4%	0.847502	84.5%	84.2%	79.2%	84.6%	model(s) 1
☐	Unknown or Legacy - White	M - U	Aged 18 to 24 years - Aged 35 to 44 years	17172	79.3%	0.828219	79.6%	79.3%	73.8%	79.7%	model(s) 1
☐	Indigenous	M	Aged 18 to 24 years - Aged 65 years and older	1283	76.0%	0.813473	79.2%	78.5%	78.7%	81.2%	model(s) ensemble
☐	Black	M	Aged 17 years and under - Aged 35 to 44 years	12385	81.0%	0.837632	81.1%	80.7%	76.5%	81.3%	model(s) 1
☐	Unknown or Legacy	*	Aged 17 years and under - Aged 55 to 64 years	4938	76.9%	0.815543	85.4%	85.0%	74.3%	84.8%	model(s) 1
☐	Unknown or Legacy - White	M - U	Aged 45 to 54 years	4412	79.7%	0.834352	85.9%	85.7%	78.0%	85.8%	model(s) 1
☐	Black - Unknown or Legacy	F - M	Aged 45 to 54 years - Aged 65 years and older	6888	86.0%	0.871056	90.9%	91.0%	86.1%	91.5%	model(s) 2
☐	Black - Latino	*	Aged 25 to 34 years - Aged 35 to 44 years	14603	83.0%	0.851367	84.2%	84.0%	79.9%	84.4%	model(s) 1
☐	East/Southeast Asian - South Asian	*	*	14917	85.9%	0.868276	89.7%	89.6%	85.3%	90.0%	model(s) 2
☐	Black	F - M	Aged 45 to 54 years	1634	83.2%	0.853751	86.0%	86.4%	82.7%	86.5%	model(s) 2
☐	*	*	Aged 17 years and under	3031	85.2%	0.859289	87.7%	88.1%	86.9%	88.6%	model(s) 1
☐	*	F	Aged 17 years and under - Aged 25 to 34 years	7016	85.8%	0.868065	86.0%	85.6%	82.2%	86.2%	model(s) 1
☐	White	*	Aged 18 to 24 years - Aged 25 to 34 years	10907	80.8%	0.839341	79.5%	79.0%	76.1%	80.0%	model(s) ensemble
☐	East/Southeast Asian	F	*	742	93.3%	0.912086	95.4%	95.6%	91.9%	95.3%	model(s) 2
☐	Latino - White	*	Aged 65 years and older	990	88.0%	0.870291	97.0%	97.1%	87.3%	96.4%	model(s) 1
☐	Latino - Unknown or Legacy	F	*	2060	86.4%	0.872031	93.1%	92.2%	85.4%	92.6%	model(s) 1

Table 5.1: Use Case #1's explanation table for prediction consensus.

According to the performance-related statistics for the models and the voting ensemble,



Select	Perceived_Race	Sex	Age_group__at_arrest_	support	agree%	avgSimilarSHAP	accuracy1	accuracy2	accuracy3	accuracyEnsemble	bestF1Score
☐	*	*	*	65087	59.4%	0.847502	84.5%	84.2%	79.2%	84.6%	model(s) 1
☐	*	F	Aged 17 years and under - Aged 25 to 34 years	7016	63.4%	0.868065	86.0%	85.6%	82.2%	86.2%	model(s) 1
☐	Unknown or Legacy - White	M	Aged 35 to 44 years - Aged 55 to 64 years	13909	56.9%	0.834566	83.4%	83.0%	76.1%	83.3%	model(s) 1
☐	Unknown or Legacy - White	*	Aged 25 to 34 years	9920	59.0%	0.836561	79.4%	78.7%	75.9%	79.7%	model(s) 1
☐	Black - Latino	M	Aged 17 years and under	1182	53.7%	0.824429	82.5%	82.7%	84.4%	83.4%	model(s) 1
☐	*	*	Aged 65 years and older	1317	65.0%	0.870280	96.4%	96.6%	87.5%	96.4%	model(s) ensemble
☐	Black - Indigenous	F - M	Aged 25 to 34 years - Aged 35 to 44 years	13529	58.9%	0.848807	83.6%	83.3%	79.3%	83.8%	model(s) 1
☐	Unknown or Legacy	*	Aged 25 to 34 years - Aged 65 years and older	3978	56.3%	0.814884	86.2%	85.8%	74.6%	85.5%	model(s) 1
☐	South Asian - Unknown or Legacy	M	Aged 18 to 24 years - Aged 25 to 34 years	3570	60.7%	0.829027	85.9%	84.9%	75.9%	85.0%	model(s) 1
☐	South Asian - Unknown or Legacy	M	Aged 17 years and under	264	51.1%	0.832815	83.3%	85.6%	82.2%	86.4%	model(s) ensemble
☐	East/Southeast Asian	F - M	*	4402	61.1%	0.879904	90.2%	90.3%	87.7%	90.4%	model(s) 2
☐	Black	F	Aged 25 to 34 years	1053	64.7%	0.871509	86.2%	85.6%	81.8%	85.7%	model(s) 1
☐	South Asian - Unknown or Legacy	F	Aged 65 years and older	24	45.8%	0.891975	100.0%	100.0%	91.7%	100.0%	model(s) 1 - 2 - enser
☐	East/Southeast Asian - Middle-Easte	*	Aged 17 years and under - Aged 25 to 34 years	6052	59.0%	0.864212	87.9%	87.8%	85.1%	88.5%	model(s) 1
☐	Latino - Middle-Eastern	*	Aged 45 to 54 years - Aged 65 years and older	937	65.3%	0.896207	93.9%	94.1%	91.2%	94.9%	model(s) ensemble
☐	East/Southeast Asian - South Asian	*	Aged 17 years and under	574	51.0%	0.880491	91.8%	91.5%	90.1%	92.9%	model(s) 2

Table 5.2: Use Case #1's explanation table for logic consensus.

we see that LightGBM has good overall performance. However, this model did not perform better for some reported patterns. For example, XGBoost has the best F1 score for Black adults between the ages of 45 to 54 with known gender (row 10 in Table 5.1). Another pattern of interest is how the voting ensemble of trained models has better accuracy and F1 Score for male Indigenous adult arrestees (row 3 in Table 5.1). Note that the models for this pattern have relatively low prediction agreement (76.0%) and accuracy (less than 80%). This pattern might suggest the necessity of collecting more training data for this subgroup.

At this point, we observe that many patterns in the explanation tables specify the arrestees' age group. To gain a better big-picture understanding, we run the *Plots* view with *age group at arrest* as the feature of interest and look at the feature plots to observe how prediction consensus changes (see Figure 5.1). We see that models with diverging predictions generally have dissimilar explanations, and models with converging predictions have similar explanations. However, adults appear to have some datapoints that do not follow this; the most notable age groups are 18 to 54. Perhaps data subsets belonging to these age groups with relatively low AvgSimilarSHAP should be further investigated. The user also notices in a plot that observes how prediction consensus and correctness changes (see Figure 5.2) that these age groups have many datapoints where the models achieved prediction consensus and were incorrect, though this was most notable for the age groups

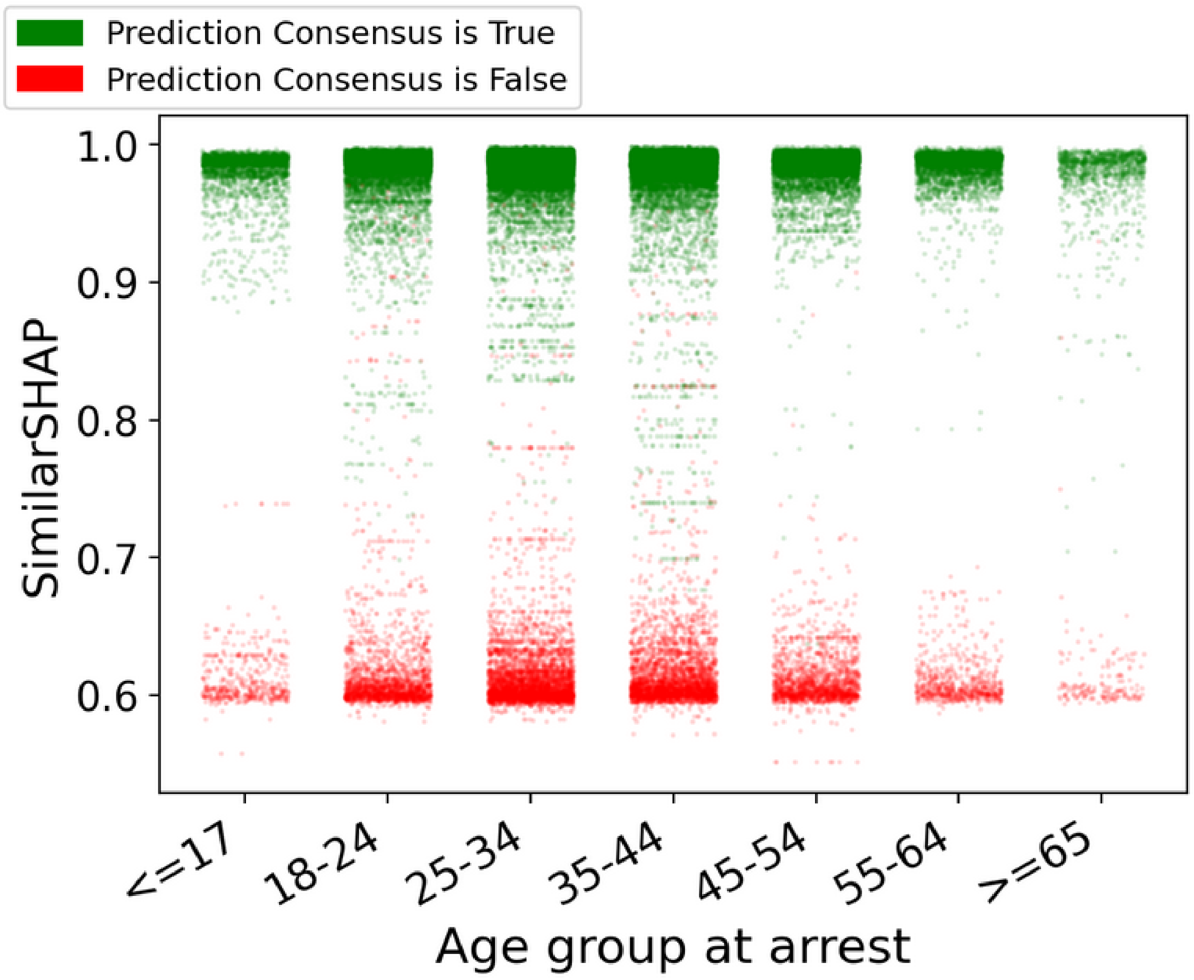


Figure 5.1: Use Case #1's feature plot for prediction consensus.

18 to 54 and was surprisingly less of a factor for the senior age group. A similar observation also applies for a plot that observes how logic consensus changes (see Figure 5.3).

Switching back to the *Tables and Counterfactuals* view, we look for one of these aforementioned data subsets to explore further. After selecting a row that corresponds to Indigenous adults, we drill down to an event where a male individual between the ages of 35 and 44 was arrested in the second quarter of the year inside Division 51 (i.e., Toronto Centre) for robbery and theft. The police did not strip search this arrestee; they did not report the arrestee as taking any notable actions. The gradient boosting models, however,

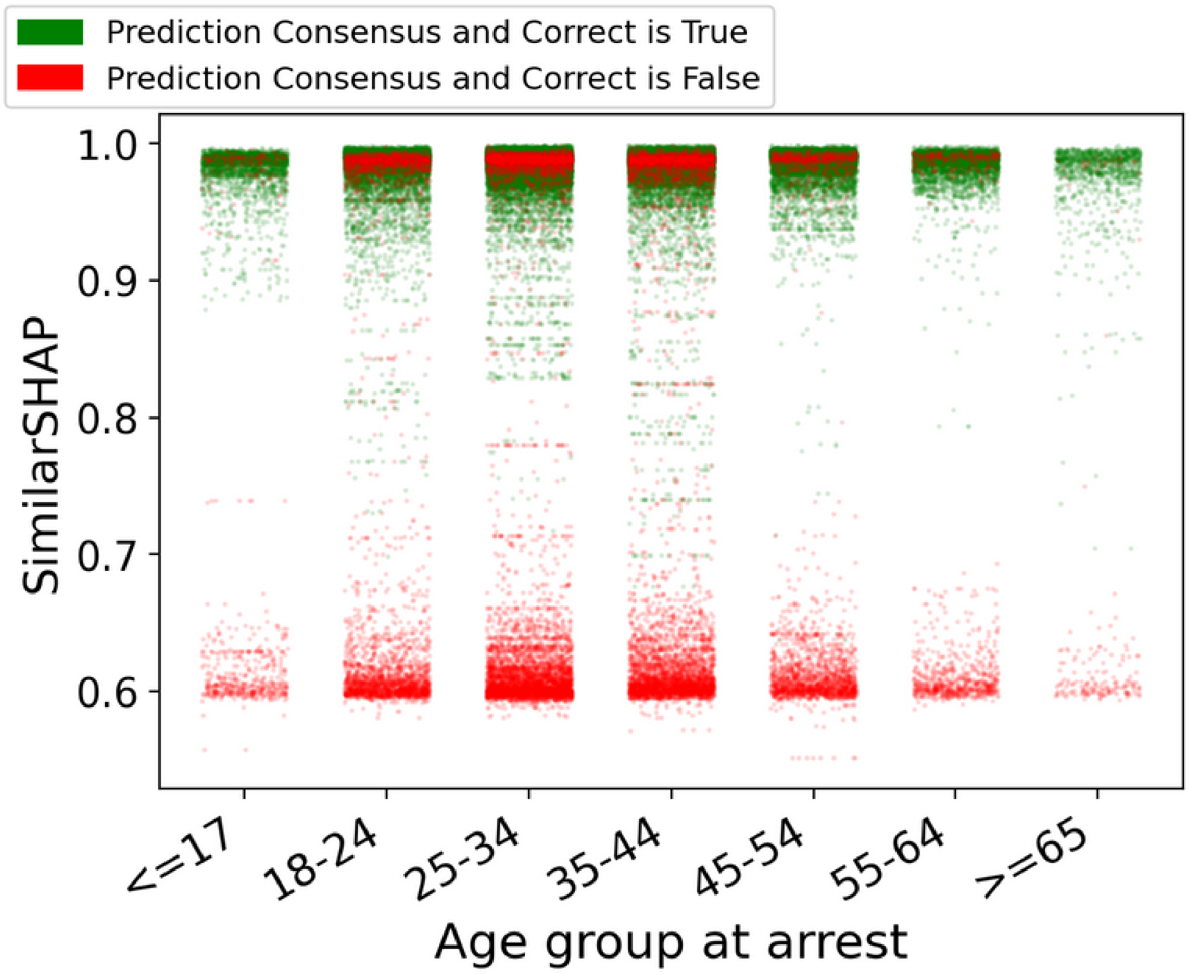


Figure 5.2: Use Case #1's feature plot for prediction consensus and correctness.

incorrectly classified this arrestee as being strip-searched.

After choosing *actions at arrest* and *arrest division location* as the features to vary (see Figure 4.4), we generate counterfactuals so that conflicting models agreed on their predictions (see Table 5.3). For example, all models would report no strip search if the arrest location division was changed to Division 11. Note that Division 11 corresponds to Toronto-Roncesvalles, which was the only neighborhood in Toronto to make it into the finals of the 2012 Canadian Institute of Planners' Great Places in Canada contest [20]. This division also has fewer incidents of robbery according to the Toronto Police. For example,

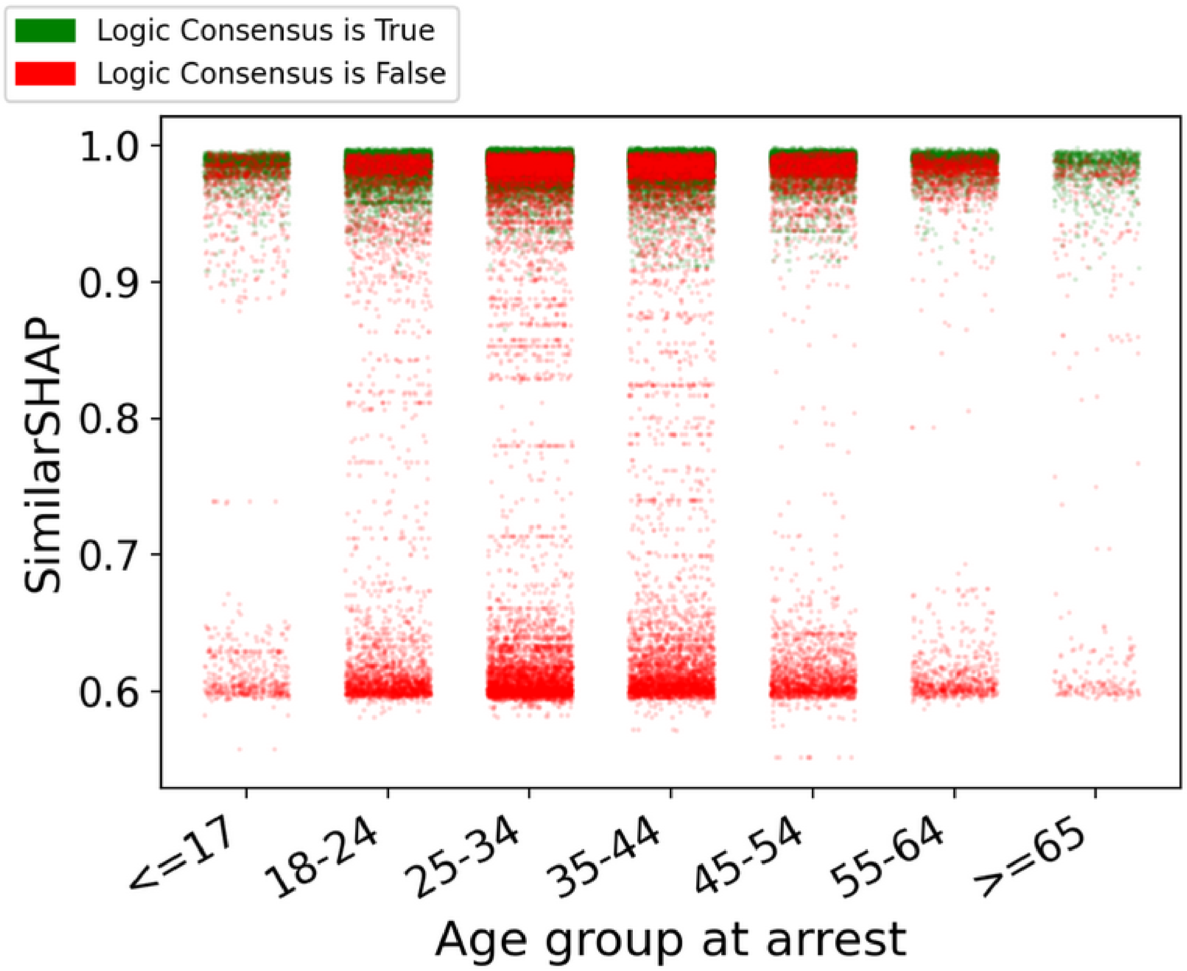
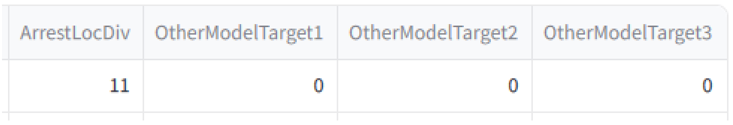


Figure 5.3: Use Case #1's feature plot for logic consensus.

from January 1 to November 4, 2023, Division 11 had 74 arrests related to robbery while Division 51 had 230 arrests [44]. This discrepancy could be a reason why a change to Division 11 might potentially lead to a shift in model agreement.

To summarize, an engineer responsible for model selection could reach several conclusions after looking at these generated patterns. First, the LightGBM appears to have better overall performance compared against the other model types. Second, there exist several row-wise patterns where XGBoost outperforms LightGBM, which could warrant further investigation of XGBoost or the use of it in an ensemble. Finally, CAMEO found certain

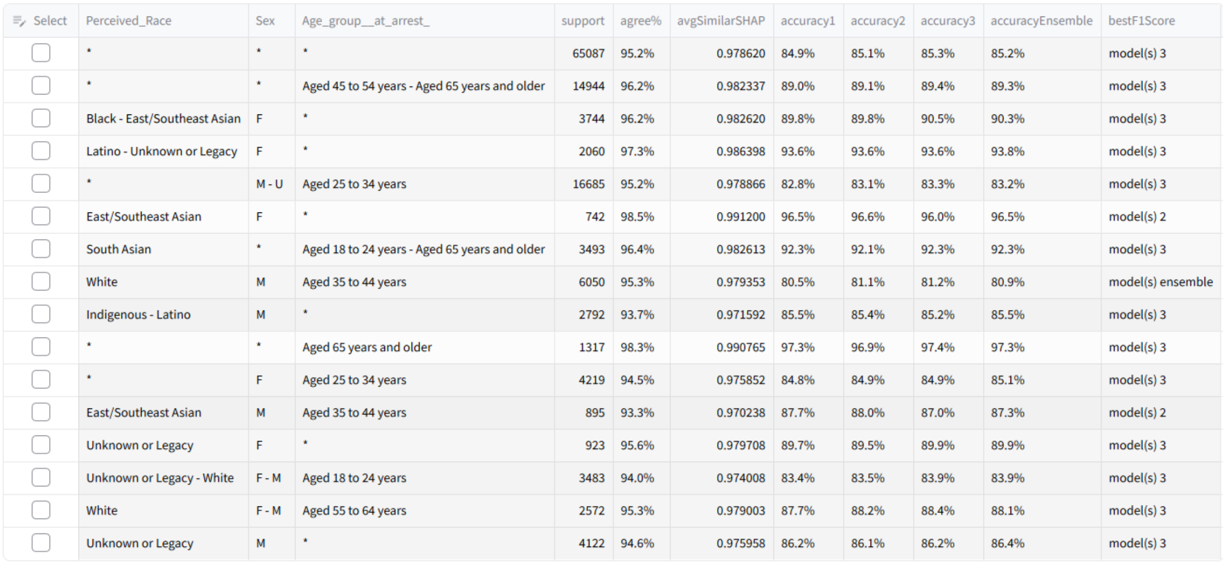


ArrestLocDiv	OtherModelTarget1	OtherModelTarget2	OtherModelTarget3
11	0	0	0

Table 5.3: Use Case #1: An example counterfactual where a change to arrest location division caused conflicting models to now agree on no strip search occurring.

subsets where all model types are confused (e.g., male Indigenous male adults), so the engineer might need to collect more training data or investigate different model types to better handle these subgroups.

5.2.2 Use Case #2: Hyperparameter Tuning

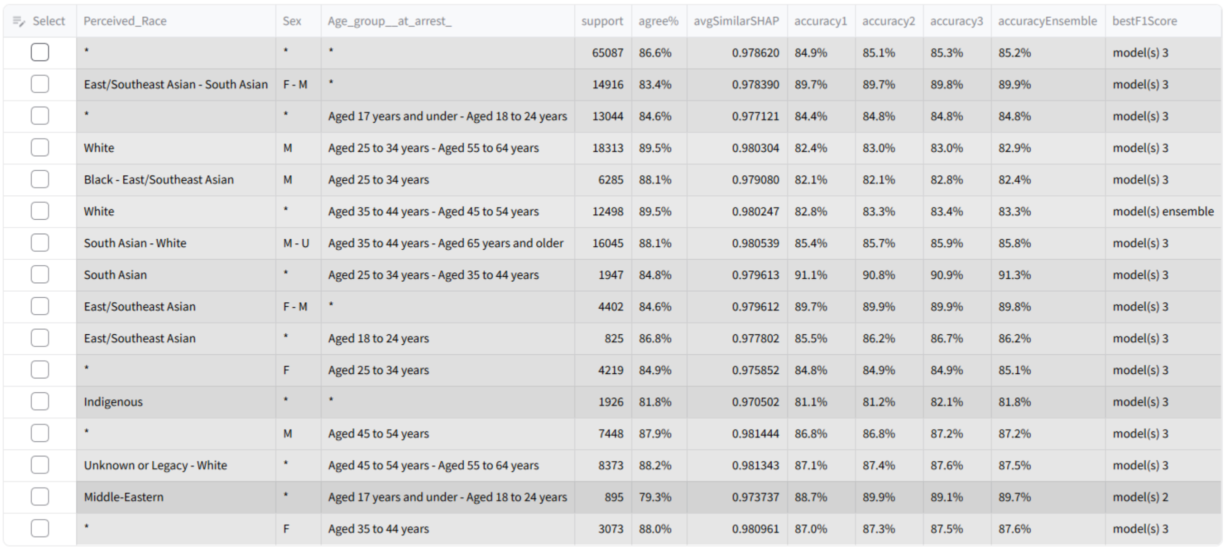


Select	Perceived_Race	Sex	Age_group__at_arrest_	support	agree%	avgSimilarSHAP	accuracy1	accuracy2	accuracy3	accuracyEnsemble	bestF1Score
☐	*	*	*	65087	95.2%	0.978620	84.9%	85.1%	85.3%	85.2%	model(s) 3
☐	*	*	Aged 45 to 54 years - Aged 65 years and older	14944	96.2%	0.982337	89.0%	89.1%	89.4%	89.3%	model(s) 3
☐	Black - East/Southeast Asian	F	*	3744	96.2%	0.982620	89.8%	89.8%	90.5%	90.3%	model(s) 3
☐	Latino - Unknown or Legacy	F	*	2060	97.3%	0.986398	93.6%	93.6%	93.6%	93.8%	model(s) 3
☐	*	M - U	Aged 25 to 34 years	16685	95.2%	0.978866	82.8%	83.1%	83.3%	83.2%	model(s) 3
☐	East/Southeast Asian	F	*	742	98.5%	0.991200	96.5%	96.6%	96.0%	96.5%	model(s) 2
☐	South Asian	*	Aged 18 to 24 years - Aged 65 years and older	3493	96.4%	0.982613	92.3%	92.1%	92.3%	92.3%	model(s) 3
☐	White	M	Aged 35 to 44 years	6050	95.3%	0.979353	80.5%	81.1%	81.2%	80.9%	model(s) ensemble
☐	Indigenous - Latino	M	*	2792	93.7%	0.971592	85.5%	85.4%	85.2%	85.5%	model(s) 3
☐	*	*	Aged 65 years and older	1317	98.3%	0.990765	97.3%	96.9%	97.4%	97.3%	model(s) 3
☐	*	F	Aged 25 to 34 years	4219	94.5%	0.975852	84.8%	84.9%	84.9%	85.1%	model(s) 3
☐	East/Southeast Asian	M	Aged 35 to 44 years	895	93.3%	0.970238	87.7%	88.0%	87.0%	87.3%	model(s) 2
☐	Unknown or Legacy	F	*	923	95.6%	0.979708	89.7%	89.5%	89.9%	89.9%	model(s) 3
☐	Unknown or Legacy - White	F - M	Aged 18 to 24 years	3483	94.0%	0.974008	83.4%	83.5%	83.9%	83.9%	model(s) 3
☐	White	F - M	Aged 55 to 64 years	2572	95.3%	0.979003	87.7%	88.2%	88.4%	88.1%	model(s) 3
☐	Unknown or Legacy	M	*	4122	94.6%	0.975958	86.2%	86.1%	86.2%	86.4%	model(s) 3

Table 5.4: Use Case #2's explanation table for prediction consensus.

Suppose the user decides to tune the hyperparameters of LGBM models. In gradient boosting, important hyperparameters control the complexity of the tree model; i.e., the max depth and number of leaves of each tree added to the ensemble. Say the user pretrains three LGBM models where *max_depth* is some value i, *num_leaves* is around (0.75×2^i), and each model uses 200 decision trees in the ensemble, for $i = 8, 9, 10$. We choose these parameter values in particular because they result in the overall accuracies of models being high and also very close to each other so that there is no clear choice for an immediate model selection.

We start again by running these models with the *Tables and Counterfactuals* view to produce two explanation tables with the same protected attributes as Use Case #1 (see Tables 5.4 and 5.5). Compared with Use Case #1, the average model consensus is much higher, which makes sense given that they are the same model type. We see that the models are more likely to achieve prediction consensus for female East/Southeast Asians (98.5% from row 6 in Table 5.4) compared against male East/Southeast Asians between the ages of 35 and 44 (93.3% from row 12). Models are more likely to achieve logic consensus for White males between the ages of 25 and 64 (89.5% from row 4 in Table 5.5) compared against Middle-Eastern individuals that are 24 and younger (79.3% from row 15).



Select	Perceived_Race	Sex	Age_group__at_arrest_	support	agree%	avgSimilarSHAP	accuracy1	accuracy2	accuracy3	accuracyEnsemble	bestF1Score
☐	*	*	*	65087	86.6%	0.978620	84.9%	85.1%	85.3%	85.2%	model(s) 3
☐	East/Southeast Asian - South Asian	F - M	*	14916	83.4%	0.978390	89.7%	89.7%	89.8%	89.9%	model(s) 3
☐	*	*	Aged 17 years and under - Aged 18 to 24 years	13044	84.6%	0.977121	84.4%	84.8%	84.8%	84.8%	model(s) 3
☐	White	M	Aged 25 to 34 years - Aged 55 to 64 years	18313	89.5%	0.980304	82.4%	83.0%	83.0%	82.9%	model(s) 3
☐	Black - East/Southeast Asian	M	Aged 25 to 34 years	6285	88.1%	0.979080	82.1%	82.1%	82.8%	82.4%	model(s) 3
☐	White	*	Aged 35 to 44 years - Aged 45 to 54 years	12498	89.5%	0.980247	82.8%	83.3%	83.4%	83.3%	model(s) ensemble
☐	South Asian - White	M - U	Aged 35 to 44 years - Aged 65 years and older	16045	88.1%	0.980539	85.4%	85.7%	85.9%	85.8%	model(s) 3
☐	South Asian	*	Aged 25 to 34 years - Aged 35 to 44 years	1947	84.8%	0.979613	91.1%	90.8%	90.9%	91.3%	model(s) 3
☐	East/Southeast Asian	F - M	*	4402	84.6%	0.979612	89.7%	89.9%	89.9%	89.8%	model(s) 3
☐	East/Southeast Asian	*	Aged 18 to 24 years	825	86.8%	0.977802	85.5%	86.2%	86.7%	86.2%	model(s) 3
☐	*	F	Aged 25 to 34 years	4219	84.9%	0.975852	84.8%	84.9%	84.9%	85.1%	model(s) 3
☐	Indigenous	*	*	1926	81.8%	0.970502	81.1%	81.2%	82.1%	81.8%	model(s) 3
☐	*	M	Aged 45 to 54 years	7448	87.9%	0.981444	86.8%	86.8%	87.2%	87.2%	model(s) 3
☐	Unknown or Legacy - White	*	Aged 45 to 54 years - Aged 55 to 64 years	8373	88.2%	0.981343	87.1%	87.4%	87.6%	87.5%	model(s) 3
☐	Middle-Eastern	*	Aged 17 years and under - Aged 18 to 24 years	895	79.3%	0.973737	88.7%	89.9%	89.1%	89.7%	model(s) 2
☐	*	F	Aged 35 to 44 years	3073	88.0%	0.980961	87.0%	87.3%	87.5%	87.6%	model(s) 3

Table 5.5: Use Case #2's explanation table for logic consensus.

According to the performance-related statistics for the models and the voting ensemble, LightGBM with $i = 10$ has the best overall performance. However, this model did not perform better for some reported patterns. For example, the LightGBM with $i = 9$ performed better for Middle-Eastern individuals that are 24 and younger (row 15 in Table 5.5), while the model ensemble performed better for White individuals between the ages of 35 and 54 (row 6 in Table 5.5). Switching to the *Plots* view and specifying *age group at arrest* as the feature of interest, the plot describing prediction consensus had similar explanations associated with consensus and dissimilar explanations associated with discord (see Figure 5.4). That said, the user notices for many age groups a surprising number of datapoints where models achieved consensus but were incorrect, suggesting further investigation of these subsets (see Figure 5.5).

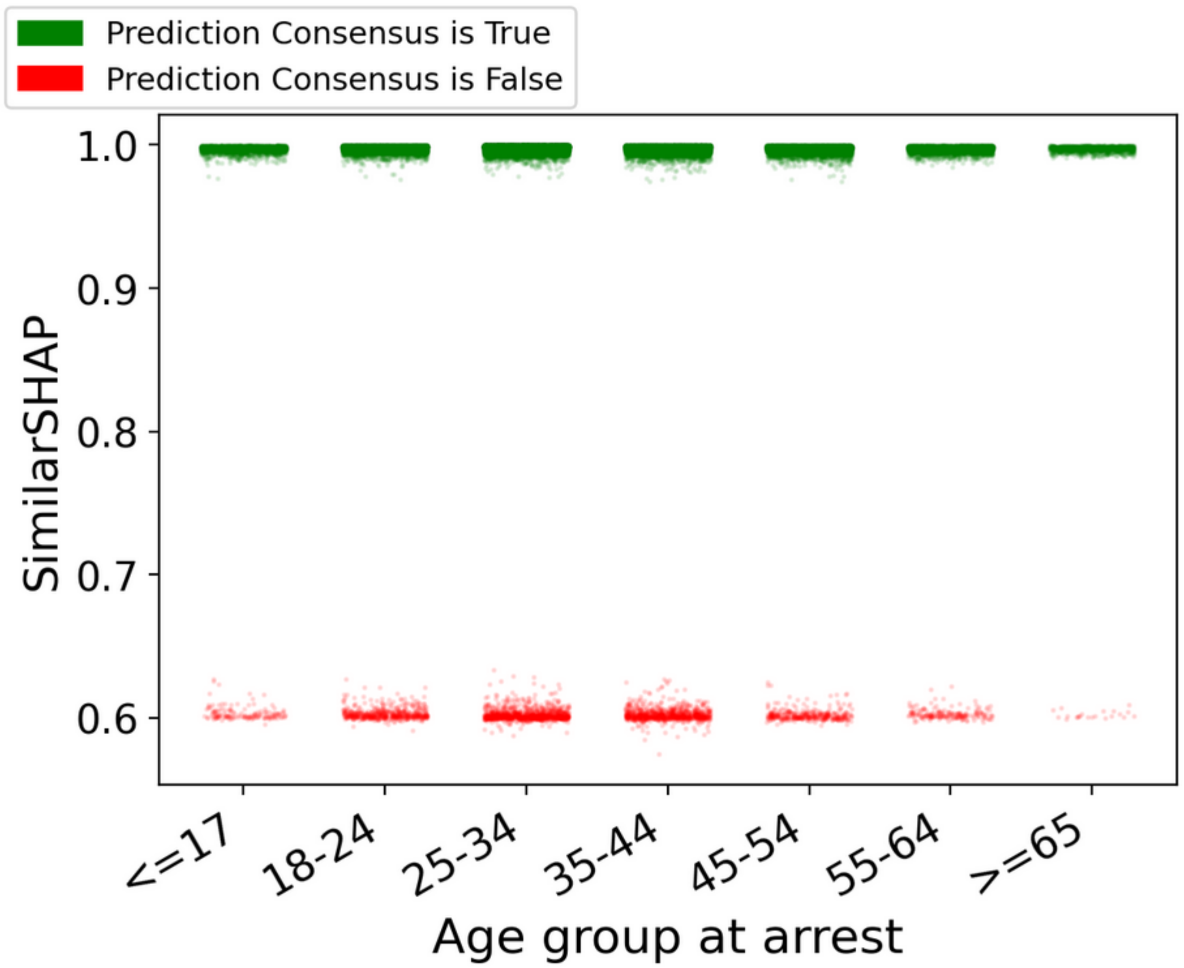


Figure 5.4: Use Case #2's feature plot for prediction consensus.

Switching to the *Table and Counterfactuals* view, we investigate the data subset of male East/Southeast Asians that are between the ages of 35 and 44, because the model with $i = 9$ handles this subgroup the best. Within this subgroup, the user finds an individual who was correctly classified as strip-searched only by the model with $i = 9$. This male East/Southeast Asian was between the ages of 35 to 44 and was arrested in the second quarter of the year in Division 14 for robbery and theft (see Figure 5.6). The police reports this individual as cooperative. When generating counterfactuals, the user specifies

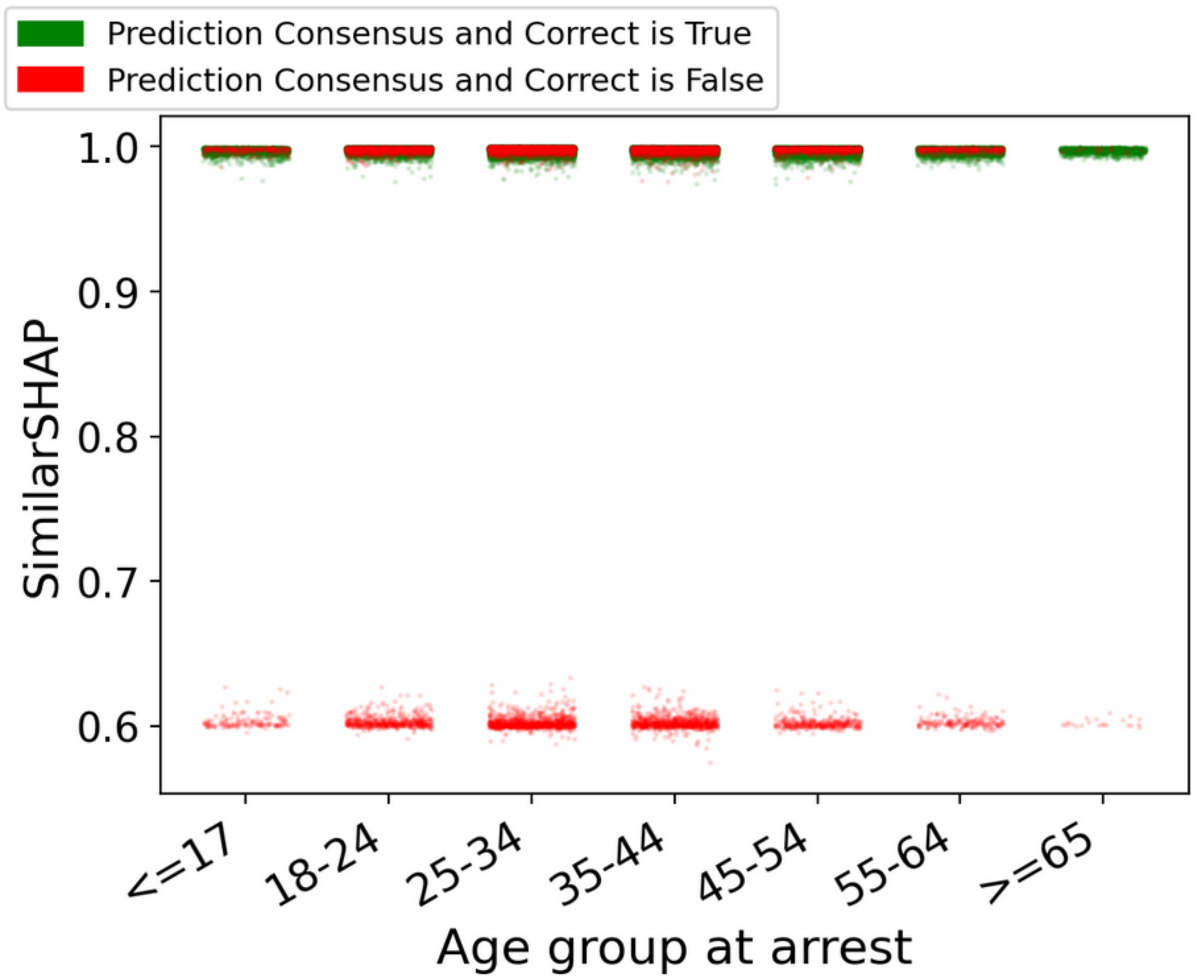
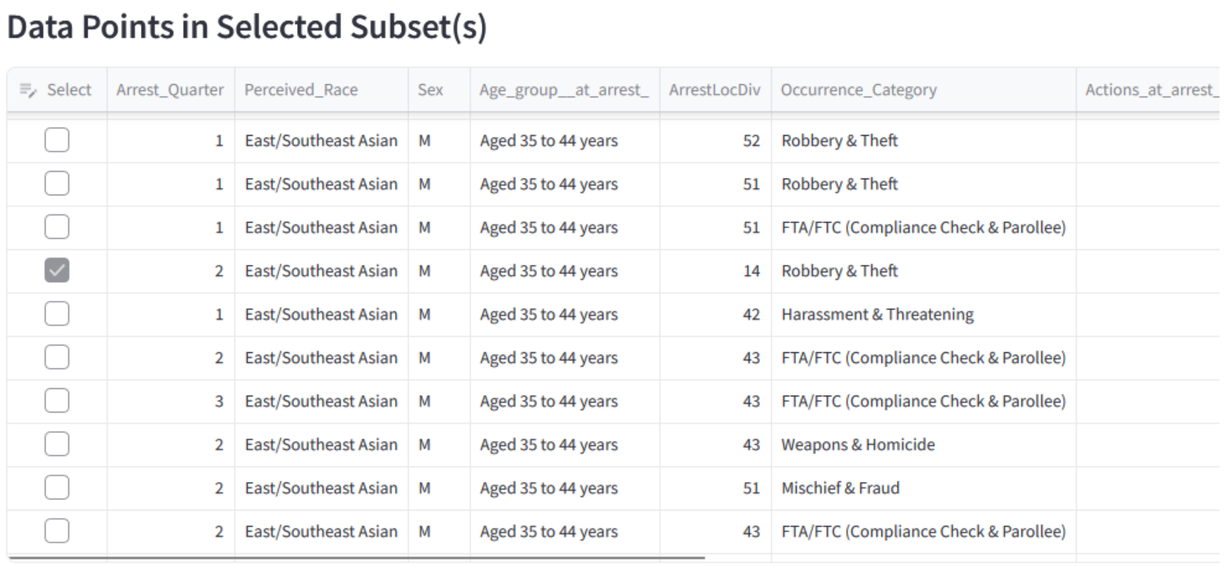


Figure 5.5: Use Case #2's feature plot for prediction consensus and correctness.

actions at arrest and *arrest location division* as the features to vary. CAMEO then reports counterfactuals where conflicting models agreed that the individual was strip-searched. This counterfactual involved the individual being arrested outside Toronto (see Table 5.6), which could be a topic for the user to investigate in more detail.

To summarize, an engineer responsible for model selection could reach several conclusions after looking at these generated patterns. First, the model variant with $i = 10$ appears to have the best overall performance compared against the other model variants. Second, several row-wise patterns exist where $i = 10$ does not have the best performance,


Data Points in Selected Subset(s)

Select	Arrest_Quarter	Perceived_Race	Sex	Age_group__at_arrest_	ArrestLocDiv	Occurrence_Category	Actions_at_arrest_
☐	1	East/Southeast Asian	M	Aged 35 to 44 years	52	Robbery & Theft	
☐	1	East/Southeast Asian	M	Aged 35 to 44 years	51	Robbery & Theft	
☐	1	East/Southeast Asian	M	Aged 35 to 44 years	51	FTA/FTC (Compliance Check & Parollee)	
✓	2	East/Southeast Asian	M	Aged 35 to 44 years	14	Robbery & Theft	
☐	1	East/Southeast Asian	M	Aged 35 to 44 years	42	Harassment & Threatening	
☐	2	East/Southeast Asian	M	Aged 35 to 44 years	43	FTA/FTC (Compliance Check & Parollee)	
☐	3	East/Southeast Asian	M	Aged 35 to 44 years	43	FTA/FTC (Compliance Check & Parollee)	
☐	2	East/Southeast Asian	M	Aged 35 to 44 years	43	Weapons & Homicide	
☐	2	East/Southeast Asian	M	Aged 35 to 44 years	51	Mischief & Fraud	
☐	2	East/Southeast Asian	M	Aged 35 to 44 years	43	FTA/FTC (Compliance Check & Parollee)	

Figure 5.6: Finding an individual datapoint for Use Case #2 to generate counterfactuals.

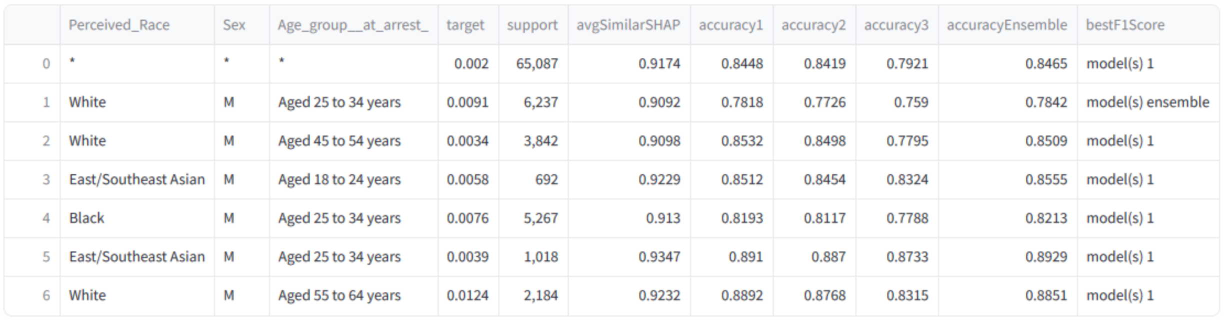
ArrestLocDiv	OtherModelTarget1	OtherModelTarget2	OtherModelTarget3
0	1	1	1

Table 5.6: Use Case #2: An example counterfactual where a change to arrest location division caused all models to correctly agree that a strip search occurred.

which could warrant further investigation such as consulting with the Toronto Police to determine if these patterns are important enough to justify the use of a different model variant in an ensemble. Finally, *where* an arrestee is arrested appears to be an important feature at the decision boundary for model agreement in our first two use cases, which could serve as motivation to further analyze this behavior.

5.2.3 Use Case #3: Finding Model Expertise

Suppose the user intends to identify subsets where a model of interest demonstrates expertise (or a lack thereof) compared with the rest of the candidate models. Using the same models, protected attributes, and minimum support threshold as Use Case #1, we select the LightGBM as our model of interest (i.e., as it had a good overall performance according to Subsection 5.2.1) and then run the *Tables and Counterfactuals* view. From the generated Table 5.7, we observe that the LightGBM model has slightly better overall performance compared with the max overall performance of the other models as the target value is positive (i.e., 0.002 in row 1). According to this table, LightGBM has unusually high expertise with White males between the ages of 25 and 64 (excluding ages 35 to 44, see rows 1, 2, and 6), male East/Southeast Asians between the ages of 18 to 34 (see rows 3 and 5), and Black males between the ages of 25 to 34 (see row 4). These patterns might suggest the necessity of collecting more training data for other subgroups or the potential of using this model in an ensemble where other models are not experts for these specific subgroups.



	Perceived_Race	Sex	Age_group__at_arrest_	target	support	avgSimilarSHAP	accuracy1	accuracy2	accuracy3	accuracyEnsemble	bestF1Score
0	*	*	*	0.002	65,087	0.9174	0.8448	0.8419	0.7921	0.8465	model(s) 1
1	White	M	Aged 25 to 34 years	0.0091	6,237	0.9092	0.7818	0.7726	0.759	0.7842	model(s) ensemble
2	White	M	Aged 45 to 54 years	0.0034	3,842	0.9098	0.8532	0.8498	0.7795	0.8509	model(s) 1
3	East/Southeast Asian	M	Aged 18 to 24 years	0.0058	692	0.9229	0.8512	0.8454	0.8324	0.8555	model(s) 1
4	Black	M	Aged 25 to 34 years	0.0076	5,267	0.913	0.8193	0.8117	0.7788	0.8213	model(s) 1
5	East/Southeast Asian	M	Aged 25 to 34 years	0.0039	1,018	0.9347	0.891	0.887	0.8733	0.8929	model(s) 1
6	White	M	Aged 55 to 64 years	0.0124	2,184	0.9232	0.8892	0.8768	0.8315	0.8851	model(s) 1

Table 5.7: Use Case #3's explanation table for the LightGBM model's expertise.

Chapter 6

Conclusion

In this thesis, we introduced CAMEO as a tool to summarize consensus and expertise patterns across multiple models. The consensus measures enable users to evaluate the robustness of their models over various contexts and also isolate potential reasons for agreement or disagreement. The expertise measure enables users to further identify the strengths of models such as identifying critical subgroups where models are relative experts. CAMEO leverages rule mining and XAI methods to find these patterns, while also supporting the use of counterfactuals to identify data perturbations that would make conflicting models agree and consistent models disagree. Our curated use cases show CAMEO's ability to explore these patterns, offering insights into model selection and hyperparameter tuning. We also created a web application for CAMEO to facilitate interactive analysis of generated patterns.

6.1 Future Research

As engineers responsible for model selection probably have more resources than just testing datasets while evaluating models, one future research direction is to expand the space of model comparisons and support further fine-grained operations to address more use cases. For example, suppose these engineers now have access to the datasets used for training candidate models (see Figure 6.1). If we can leverage methods such as ModelDiff [42] to compute the influence of training data on models, then we can also check whether models agreed or disagreed on training data subsets. This potentially has use cases for better evaluating training data quality.

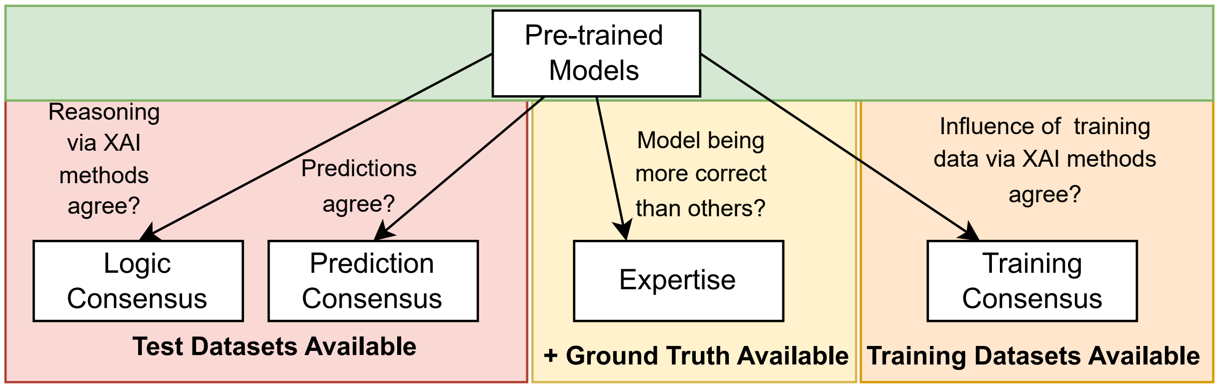

Pre-trained Models
Reasoning via XAI methods agree?
Predictions agree?
Model being more correct than others?
Influence of training data via XAI methods agree?
Logic Consensus
Prediction Consensus
Expertise
Training Consensus
Test Datasets Available
+ Ground Truth Available
Training Datasets Available

Figure 6.1: Expanding the space of model comparisons when more resources are available to engineers.

Further emphasis on the notion of time could be an interesting direction for future research. Time series data is immutable (i.e., data comes in chronological order) and append-only (i.e., for data consistency); change over time and the chronological order of record data are particularly important aspects for time series data. Temporal dependencies can result in the features that drive a model's outcome changing over time: for example, a change in some input features might not immediately change a model's prediction for many real-life applications. In context of multiple models, we might have a case where models compared at one point of time agree on predictions for some data subsets but then disagree at another point of time. We can also extend this to other comparison operations, such as models gaining (or losing) expertise at different points in time. Perhaps focusing more on explanations to better understand temporal dependencies across multiple models could benefit model engineers? It might also be interesting to treat seasonal-based patterns as first-class citizens instead of having to rely on range-based patterns from the current explanation tables.

Although the techniques used by CAMEO do not rely on having access to the internal logic of models, our experiments currently use *structured* datasets with predefined data models and clearly identified features. This limitation leads to a new unexplored research direction where we generalize our approach: can we also apply CAMEO-like techniques to *unstructured* data? Example models for unstructured data include large language models (LLMs) or models for computer vision applications (e.g., CNNs or convolutional neural network models). Indeed, our motivation for explaining multiple models still applies for unstructured data. For example, fine-tuning LLMs (i.e., re-training pretrained LLMs on specific datasets so that they are better tailored for domain-specific use cases) is still the industry standard for companies such as Grammarly: model selection is still a relevant problem because models such as GPT−4 are not yet being exclusively used.

We can now formulate a new space of explanations for multiple models along an additional axis: *structured* data versus *unstructured* data. This future research would involve investigating and formulating new techniques to add structure to the unstructured for CAMEO to then apply comparison operations on. For example, POEM [12], which addresses individual CNNs, leverages a gradient-based approach inspired by Grad−CAM [40] to attribute concepts to images. Perhaps we could find concept consensus or discord amongst CNNs? Regarding expertise, perhaps we can identify a subset of topics where a LLM or ensemble of LLMs has better performance, or create an expertise taxonomy for CNNs?

A natural step for extending CAMEO is to better identify its strengths and weaknesses by consulting with real-life users. When selecting participants for a user study that thoroughly evaluates the interpretability and usefulness of explanations and interface provided by CAMEO, we would prioritize engineers responsible for model selection in particular. We imagine this user study would involve 1) gathering end-to-end feedback from all stages of use, and 2) datasets and pre-trained models that all participants are familiar with. Their feedback would also help us triage new features for improved functionality. One useful feature could be updating the tool's interface to support user-defined comparison operations for use as the target in their generated patterns. For example, the user might decide to define the target in their explanation tables as the probability of classes from model output to give them a better idea of *model confidence* across multiple models. Perhaps users might desire more automation of the selection process such as being prompted beforehand to specify what patterns must be present so that the tool automatically presents the best model according to that criteria.

Another direction for future research is to focus more on ensembles as a first-class citizen: can we use CAMEO-like techniques to find more suitable ensembles for data subsets? For example, engineers can find it time-consuming to balance the cost of creating and managing ensembles with additional performance gains. Perhaps one use case could be finding which subsets of models (i.e., those that would then make up an ensemble) can handle a subset of protected attributes better than other model subsets? Could we identify expert ensembles among a set of different ensembles?

Further optimization of CAMEO could help improve its overall usefulness to users. Regarding explanation table generation, further investigating some pruning strategies for potential patterns to present could improve its performance for users. To speed up SimilarSHAP computation, one possible strategy that we could investigate is running a dimensionality-reduction method on each SHAP explanation vector before then computing entropy. In this case, low and high entropy would indicate similarity and dissimilarity, respectively. To produce counterfactuals, CAMEO currently leverages the DiCE method [28], which we found in practice to be a bottleneck performance-wise. One possible improvement is to allow the user to search for nearest neighbor-based counterfactuals to naturally provide feasible datapoints that are close in proximity. We can alternatively explore the use of other libraries such as NICE [7] that look at the nearest unlike neighbor and then iteratively introduce feature values from this neighbor into the instance to be explained. That said, we would have to account for the shortcomings of these other libraries, such as NICE lacking the ability to set constraints or specify which features to vary.

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] Emanuele Albini, Jason Long, Danial Dervovic, and Daniele Magazzeni. Counterfactual Shapley additive explanations. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pages 1054–1070, New York, NY, USA, 2022. Association for Computing Machinery.

[3] Michael R. Anderson, Dolan Antenucci, Victor Bittorf, Matthew Burgess, Michael J. Cafarella, Arun Kumar, Feng Niu, Yongjoo Park, Christopher Ré, and Ce Zhang. Brainwash: A data system for feature engineering. In *Conference on Innovative Data Systems Research*, 2013.

[4] Jawid Ahmad Baktash and Mursal Dawodi. GPT-4: A review on advancements and opportunities in natural language processing, 2023.

[5] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, Feb 2012.

[6] Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–215, 2001.

[7] D. Brughmans, P. Leyman, and D. Martens. NICE: an algorithm for nearest instance counterfactual explanations. *Data Min Knowl Disc*, 2023.

[8] Chaofan Chen, Kangcheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang, and Tong Wang. An interpretable model with globally consistent explanations for credit risk. *CoRR*, abs/1811.12615, 2018.

[9] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[10] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Automated data slicing for model validation: A big data - AI integration approach. *IEEE Transactions on Knowledge and Data Engineering*, 32(12):2284–2296, 2020.

[11] Vargha Dadvar, Lukasz Golab, and Divesh Srivastava. Exploring data using patterns: A survey. *Information Systems*, 108:101985, 2022.

[12] Vargha Dadvar, Lukasz Golab, and Divesh Srivastava. POEM: Pattern-oriented explanations of convolutional neural networks. *Proc. VLDB Endow.*, 16(11):3192–3200, Jul 2023.

[13] Elena Dumitrescu, Sullivan Hué, Christophe Hurlin, and Sessi Tokpavi. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, 297(3):1178–1192, 2022.

[14] Armin Esmaeilzadeh, Lukasz Golab, and Kazem Taghva. InfoMoD: Information-theoretic model diagnostics. In *Proceedings of the 35th International Conference on Scientific and Statistical Database Management*, SSDBM '23, New York, NY, USA, 2023. Association for Computing Machinery.

[15] Guoyao Feng, Lukasz Golab, and Divesh Srivastava. Scalable informative rule mining. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 437–448, 2017.

[16] Kareem El Gebaly, Guoyao Feng, Lukasz Golab, Flip Korn, and Divesh Srivastava. Explanation tables. *IEEE Data Eng. Bull.*, 41(3):43–51, 2018.

[17] Seymour Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70:320–328, 1975.

[18] Zixuan Geng, Maximilian Schleich, and Dan Suciu. Computing rule-based explanations by leveraging counterfactuals. *Proc. VLDB Endow.*, 16(3):420–432, Nov 2022.

[19] Evgueni Haroutunian. *Information Theory and Statistics*, pages 666–667. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[20] Erin Hatfield. Roncesvalles in the running for great neighbourhood contest, Apr 2012.

[21] Neil Jethani, Mukund Sudarshan, Ian Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time Shapley value estimation, 2022.

[22] Minsuk Kahng, Dezhi Fang, and Duen Horng (Polo) Chau. Visual exploration of machine learning results using data cube analysis. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, HILDA '16, New York, NY, USA, 2016. Association for Computing Machinery.

[23] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc.

[24] Tracy L. Lamb, Keith J. Ruskin, Stephen Rice, Leili Khorassani, Scott R. Winter, and Dothang Truong. A qualitative analysis of social and emotional perspectives of airline passengers during the COVID-19 pandemic. *Journal of Air Transport Management*, 94:102079, 2021.

[25] Yang Li, Yu Shen, Huaijun Jiang, Wentao Zhang, Jixiang Li, Ji Liu, Ce Zhang, and Bin Cui. Hyper-Tune: Towards efficient hyper-parameter tuning at scale, 2022.

[26] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *International Conference on Neural Information Processing Systems*, NIPS'17, pages 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[27] David Martens and Foster Provost. Explaining data-driven document classifications. *MIS Quarterly*, 38(1):73–100, 2014.

[28] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pages 607–617, New York, NY, USA, 2020. Association for Computing Machinery.

[29] Prathyush S. Parvatharaju, Ramesh Doddaiah, Thomas Hartvigsen, and Elke A. Rundensteiner. Learning saliency maps to explain deep time series classifiers. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pages 1406–1415, New York, NY, USA, 2021. Association for Computing Machinery.

[30] Romila Pradhan, Jiongli Zhu, Boris Glavic, and Babak Salimi. Interpretable data-based explanations for fairness debugging. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, pages 247–261, New York, NY, USA, 2022. Association for Computing Machinery.

[31] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pages 469–481, New York, NY, USA, 2020. Association for Computing Machinery.

[32] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning, 2020.

[33] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[34] Cynthia Rudin and Yaron Shaposhnik. Globally-consistent rule-based summary-explanations for machine learning models: Application to credit-risk evaluation. *Journal of Machine Learning Research*, 24(16):1–44, 2023.

[35] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning representations by back-propagating errors*, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[36] Svetlana Sagadeeva and Matthias Boehm. SliceLine: Fast, linear-algebra-based slice finding for ML model debugging. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, pages 2290–2299, New York, NY, USA, 2021. Association for Computing Machinery.

[37] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[38] Maximilian Schleich, Zixuan Geng, Yihong Zhang, and Dan Suciu. GeCo: Quality counterfactual explanations in real time. *Proc. VLDB Endow.*, 14(9):1681–1693, May 2021.

[39] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison.

Hidden technical debt in machine learning systems. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2503–2511, Cambridge, MA, USA, 2015. MIT Press.

[40] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[41] Laleh Seyyed-Kalantari, Haoran Zhang, Matthew B. A. McDermott, Irene Y. Chen, and Marzyeh Ghassemi. Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations. *Nat Med*, 27:2176–2182, 2021.

[42] Harshay Shah, Sung Min Park, Andrew Ilyas, and Aleksander Madry. ModelDiff: A framework for comparing learning algorithms. In *International Conference on Machine Learning*, pages 30646–30688. PMLR, 2023.

[43] Toronto Police Service. Race based data: open data documentation, Nov 2022.

[44] Toronto Police Service. My neighbourhood, Nov 2023.

[45] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard journal of law & technology*, 31:841–887, Apr 2018.

[46] Junpeng Wang, Liang Wang, Yan Zheng, Chin-Chia Michael Yeh, Shubham Jain, and Wei Zhang. Learning-from-disagreement: A model comparison and visual analytics framework. *IEEE Transactions on Visualization and Computer Graphics*, 29(9):3809–3825, Sep 2023.

[47] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 2020.

[48] Yuankai Wu, Hongyu Yang, Yi Lin, and Hong Liu. Spatiotemporal propagation learning for network-wide flight delay prediction. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–15, 2023.

[49] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S. Ebert. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):364–373, 2019.

[50] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 1st edition, 2012.

[51] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018.