A QUEST for Model Assessment: Identifying Difficult Subgroups via Epistemic Uncertainty Quantification

Katherine E. Brown, PhD[1,2], Steve Talbert, PhD[3], Douglas A. Talbert, PhD[1]
[1]Tennessee Technological University, Cookeville, TN; [2]Vanderbilt University Medical Center, Nashville, TN; [3]University of Central Florida, Orlando, FL

Abstract

Uncertainty quantification in machine learning can provide powerful insight into a model's capabilities and enhance human trust in opaque models. Well-calibrated uncertainty quantification reveals a connection between high uncertainty and an increased likelihood of an incorrect classification. We hypothesize that if we are able to explain the model's uncertainty by generating rules that define subgroups of data with high and low levels of classification uncertainty, then those same rules will identify subgroups of data on which the model performs well and subgroups on which the model does not perform well. If true, then the utility of uncertainty quantification is not limited to understanding the certainty of individual predictions; it can also be used to provide a more global understanding of the model's understanding of patient subpopulations. We evaluate our proposed technique and hypotheses on deep neural networks and tree-based gradient boosting ensemble across benchmark and real-world medical datasets

Introduction

The automatic identification of subsets of data on which supervised learning models make systematic errors is a challenging problem that has recently received some attention in the research literature[1–4]. Understanding subsets of input data that are likely to result in an error by the AI model is an important aspect of model transparency and trustworthiness. Manually guided approaches are typically on specific, well-defined subpopulations (e.g., assessing the performance of known groups of interest such as gender or race) or are expensive, time-consuming endeavors (e.g., error auditing) that are highly dependent on human ability to recognize relevant patterns[5]. Existing automated techniques rely on clustering-based approaches[6,7], variations of enumerate-and-test approaches[3,4], or are specific solutions tailored to deep neural networks[2,7].

This paper proposes and evaluates a novel *supervised* approach that exploits a connection between high epistemic uncertainty and the likelihood of misclassification to identify subpopulations of patients that the model appears to understand well and others with which the model has far less success. We call this technique QUEST (Quantifying Uncertainty for Estimating Subgroup Types). Unlike most existing automated approaches, QUEST frames the problem as a supervised learning problem. It uses *uncertainty class* (e.g., high or low uncertainty) as the instance label, allowing QUEST to directly search for rules that define useful classes. These rules can also be considered a global surrogate model for explaining the uncertainty class of each prediction[8].

QUEST was inspired by the use of rejection classification curves to assess the calibration of epistemic uncertainty quantification. When such estimates are well-calibrated, rejection classification curves show that removing instances with high classification uncertainty improves overall model performance[9,10]. Given that property of uncertainty quantification, if QUEST can discover rules that accurately describe groups with well-defined uncertainty classes (e.g., high or low uncertainty), then such rules should also define groups with meaningful statistical properties relative to classification accuracy. Thus, this paper evaluates the hypothesis that *we can use uncertainty quantification as a model-agnostic performance analysis tool that can identify higher- and lower-accuracy patient subpopulations*. To that end, we define and perform experiments to address each of the following questions: (1) Can we identify rules for different types of models that define subgroups of patients with differing levels of epistemic uncertainty well enough to accurately predict the uncertainty levels of unseen examples? (2) Does the predicted uncertainty of the discovered subgroups correlate to the classification accuracy of patients in those subgroups? (3) Given that a human will likely be involved in analyzing and developing plans to address selected subgroups, how many subgroups (or rules) are needed to accurately predict the uncertainty classes?

If questions 1 and 2 are answered affirmatively, then that supports our hypothesis by suggesting that we can indeed use uncertainty quantification as a model-agnostic performance analysis tool to identify patient subpopulations that describe strengths and weaknesses of the model that should be able to inform model improvement efforts. The answer to question 3 suggests the number of rules needing examination as part of the subgroup assessment and response process.

Note that this paper focuses on the ability for QUEST to identify high- and low-accuracy populations of patients relative to a particular model and does not attempt to define and assess a fully developed process for using it to perform model refinement. Also note that the current implementation of QUEST assumes tabular data because of the nature of its rule generation process. We believe, however, that QUEST could be adapted to other data types given an appropriate alternative approach to rule generation.

Background

Uncertainty Quantification

Uncertainty quantification is a powerful tool with the potential to increase trust of deep learning and other black-box AI techniques in various critical fields. A 2019 editorial in *Nature*[11] expressed the need for uncertainty quantification for deep learning in medical settings. There are two broad classes of uncertainty in neural networks: *epistemic uncertainty* and *aleatoric uncertainty*[12]. Aleatoric uncertainty refers to the extent to which noise in the data reduces the effectiveness of a learned classification or regression model. This type of uncertainty can only be mitigated by reducing the noise in the data and is generally considered irreducible at the algorithmic level. Thus, this work is focused on epistemic uncertainty.

Epistemic uncertainty hails from a model's inability to learn. More data and using more appropriate models have both been shown to reduce epistemic uncertainty[12]. Epistemic uncertainty can also give an estimate of where, within the data distribution, a data point of interest lies[12,13]. Higher epistemic uncertainty can imply greater distance between the data point of interest and the training data distribution. As data to predict strays from this training distribution, predictions become more unreliable since the machine learning model is being asked to extrapolate into a previously unseen region of the data.

There are several techniques available to provide reliable uncertainty estimates[13-19] The technique we utilize for *neural networks* is Bayesian dropout which uses dropout during both training and testing[13]. Dropout was originally developed as a regularization technique to prevent networks from overfitting[20]. It is typically only applied during the training phase of a neural network, but Gal proved that placing dropout "before every weight layer" in a deep neural network results in an approximation to Gaussian processes[13]. Sampling an input multiple times in such a network results in a distribution of output values, and the standard deviation of this distribution is a measure of uncertainty[13]. *Gradient-boosting trees*, however, are serial in nature with each tree building off the mistakes of the previous tree[21]. Thus, removing trees at random in a dropout-esque manner will not suffice. Therefore, we utilize the virtual ensembles technique[14]. This technique creates an ensemble by only considering sequences of trees with random endpoints. The maximum number of samples that can be used to calculate uncertainty is limited by the size of the tree ensemble.

Decision Tree Induction

Decision trees are a supervised machine learning algorithm that produces a series of if-then rules to produce a classification[22,23] Each node in a decision tree presents a "split point" based on the result of the boolean predicate presented and determines the path through the tree. Predicate are selected using a heuristic criterion such as Gini impurity[23]. A classification is obtained when the tree reaches a leaf node. Decision trees are one of the few modern supervised machine learning algorithms that can provide both global and local explanations with no ad-hoc post-processing required. The rule-based structure of the decision tree provides a complete view of the decision-making process of the algorithm, and the individual path taken for a data point provides a local explanation of a prediction. For our purposes, we note that while the rules in a decision tree are constructed to be *predictive* rules, they are often used in real-world application to discover *interesting* rules. In that latter context, decision trees can be viewed as discovering interesting subgroups within the given data[24]. Each leaf in the tree represents the subgroup defined by the associated rule with the depth of the tree (length of the longest path) defining the length of the longest subgroup rule.

Identification of Systematic Errors in Machine Learning Models

There are a variety of techniques available to identify systemic errors in machine learning models. Identifying subsets or subgroups or slices of data that are associated with errors in a machine learning system is known as *hidden stratification*[1,7]. Techniques to perform analysis to reveal hidden stratification or find systemic errors in machine learning range from completely manual to semi-automatic to mostly or completely automatic. Error auditing is a mostly manual operation that evaluates model predictions to find subsets of data with prevalent error rates[1]. Another machine learning practitioner-automated approach is PALM[6]. PALM formulates model explanation in terms of partitions of the training data that influence the model. A data subset is said to influence the model if altering the data subset alters the model's predictions. PALM is a two-part surrogate model that (1) partitions the training data and (2) creates surrogate submodels that attempt to capture model behavior on the partition.

A prevalent semi-automated approach is Spotlight[2]. Spotlight performs "soft clustering" on the output of a neural network layer to find inputs most prone to error. Since clustering is only performed on the final layer, it allows Spotlight to be generalizable to any type of neural network; however, Spotlight does require manual intervention to determine the appropriate "spotlight" size.

Other techniques are automated or mostly automated[3,4,7]. These techniques use unsupervised learning to partition data into error-prone subgroups. Pastor et al.[3] used itemsets to enumerate feature-value combinations to find subsets of data that had divergence in a desired metric (e.g., FPR, FNR) to reveal subgroups that were subject to model bias. Data slicing[4] used clustering and decision trees to identify problematic data slices, and lattice search to find potential, error-prone data subsets and used a combination of statistical significance tests to confirm. Finally, GEORGE [7] used clustering to find "subclasses" of data upon which classifiers were retrained for improvement. GEORGE was primarily applied to trained neural networks.

Quantifying Uncertainty for Estimating Subgroup Types

In this section, we discuss the intuition of the proposed QUEST system. At its core, QUEST is designed to find subsets of data that are associated with a discrete level (e.g., low/high or low/medium/high) of uncertainty. We hypothesize, based on the behavior of well-calibrated uncertainty, that high uncertainty subgroups are associated with decreased accuracy. We briefly describe the intuition of how QUEST operates to assign uncertainty labels to subsets of data and the placement of QUEST in the context of existing literature on hidden stratification and model assessment.

Intuition of QUEST

The goal of QUEST is to learn interpretable rules using the same input features as the classification model to produce a label of low/high uncertainty or low/medium/high uncertainty. Table 1 describes the inputs into QUEST.

Table 1. Description of input and variables of QUEST

Type	**Variable**	**Description**
Input Variables	t	Supervised *training and validation* dataset (labeled with correct classification labels)
	m	Classification model trained and validated upon t
	u	Uncertainty quantification technique
	d	Discretization technique
	q	Supervised, transparent rule induction algorithm
Output Variables	R	Set of rules defining subgroups associated with different levels of uncertainty (each $r \in R$ is an if-then rule describing one subgroup)

QUEST operates in three stages. Stage 1 is to compute the uncertainty of the classification model m using the uncertainty quantification technique u on the training and validation data t. Stage 2 converts the real-valued uncertainty into relative bins using discretization technique d to enable utilization of supervised learning techniques. Finally, Stage 3 uses a transparent rule induction algorithm q to compute the set of rules R that describe the subgroups of uncertainty type.

Placement of QUEST in Existing Literature

QUEST relies on uncertainty quantification technique as a proxy for incorrectness. This differs from several existing techniques that use training error as a heuristic for similar model assessment techniques[3,4]. Another advantage of QUEST compared to existing techniques is that QUEST is model agnostic given the classification model has a calibrated uncertainty quantification technique. This is an advantage over Spotlight[2] and GEORGE[7] which utilize the latent neural network structure and thus are only applicable to neural networks. ANNs and GBTs are two extremely common classification models for numeric data that satisfy this condition[13,14].

Experimental Methodology

In this section, we describe the experimental methodology to evaluate our proposed technique. First, we present the datasets and associated classification tasks, models and uncertainty quantification protocol, and finally, the implementation of subgroup discovery for uncertainty quantification. A Github repository to the code can be found https://github.com/thekatiebr/QUEST.

Datasets and Associated Classification Tasks

All experiments are performed using 10-fold stratified cross-validation. Each cross-validation split provides a training partition (90% of the data; equivalent to input variable *t*) and a testing partition (10% of the data). Each fold's testing partition is referred to as the testing data. We further split the training partition, t, into a training set (60% of the training partition) and a validation set (40% of the training partition). Uncertainty information is gathered for each fold's training, validation, and testing data. We report average accuracy and F1-score for each classification model/task combination averaged across the aforementioned 10-fold stratified cross-validation.

The benchmark dataset is the Pima Diabetes (Diabetes) dataset from the UCI Machine Learning Repository[25]. This dataset (n=768) contains 8 continuous features including the number of pregnancies, glucose levels ascertained from a glucose tolerance test, diastolic blood pressure (in mmHg), tricep skin thickness, insulin level, body mass index, diabetes pedigree function, and age. The classification task in this dataset is determining if a described patient has diabetes. The class distribution of this dataset 65%/35% skewed towards patients not having diabetes.

We also use a dataset (Trauma) from the trauma registry of a Level 1 Trauma Center (n=56,888) [26]. It uses 32 features including physical parameters (e.g., systolic/diastolic blood pressures, heart rate, Glasgow Coma Scale score), anatomical criteria, mechanism of injury, age, and multiple computed injury scores (e.g., Revised Trauma Score and the Air Medical Prehospital Transport Score). The classification task is to determine if a patient has an Injury Severity Score (ISS) of at least 15. An ISS ≥ 15 indicates the patient is severely injured and should be triaged as such[42]. The class distribution is 69%/31% skewed towards patients having an ISS < 15 (not severely injured).

Finally, we utilize the Medical Information Mart for Intensive Care (MIMIC) IV emergency department data (n = 441,437)[28,29]. It contains data regarding patients' emergency department visits at Beth Israel Medical Center during the 2010 decade. We follow the preprocessing steps described by Xie et al. to prepare the data[30]. We use the following 3 prediction tasks as defined by Xie et al.[30]: patient outcome defined as death or ICU admission within 24 hours (Crit. Outcome), likelihood of readmission within 3 days (3-Day Readmit), and prediction of hospitalization (Hosp. Pred.). The class distribution for Crit. Outcome is 94%/6% (skewed towards no critical outcome), 3-Day Readmit is 97%/3% (skewed towards no readmission), and Hosp. Pred is 53%/47% (skewed towards no hospitalization).

Models and Uncertainty Quantification

We used two types of classification models m in this work: neural network classifiers (ANN) and gradient-boosting trees (GBT). For the ANNs, all hidden layers are use the Rectified Linear Unit (ReLU) activation function and the output layers are use the softmax activation function, and we use the ADAM optimizer to minimize binary cross entropy[31]. We use the Keras API[32] with Tensorflow[32,33] for our neural networks implementation. We used a maximum of 100 training epochs with early stopping if validation loss does not improve after 5 iterations. We use dropout $p = 0.3$ to measure uncertainty with $t = 50$ Monte-Carlo samples[13]. We used the Catboost-AI library[34] for the GBT models and the virtual ensembles technique[13] to measure GBT uncertainty with the number of virtual ensembles set to 50; however, if this value is too high due to the size of the final learned tree ensemble, we default to 15 virtual ensembles.

Implementation of QUEST

For QUEST, we opted to convert the task into a classification task based on discretizing uncertainty to quantile-based bins. Uncertainty quantification using either dropout or virtual ensembles as choice of *u* produces a *relative* value and is only interpretable when other uncertainty values are known[13,14]. Discretizing into bins produces multiple levels of uncertainty. With lower bin values implying lower uncertainty, this scheme is easily interpretable. Moreover, discretizing uncertainty allows us to use accuracy-related metrics to evaluate QUEST's performance. For discretization technique *d,* we determined the bounds of the bins by ensuring a roughly equal number of data points per bin. The number of bins $k \geq 2$ can be arbitrarily chosen based on domain knowledge or optimized as a hyperparameter. In this work, we evaluate results for 2 uncertainty bins and 3 uncertainty bins. Unless otherwise noted, we do not weight averages by subgroup size in reporting results.

For the primary backend for QUEST (equivalently, rule induction algorithm *q*), we implement decision trees for subgroup discovery using the Scikit-Learn library[35].The classification task is to assign each data point to the correct uncertainty bin based on feature values. The subgroup that a datapoint is assigned to depends on which leaf resulted in the bin classification. Each leaf is the result of a distinct, unique path through the tree[22,36]. Thus, the rules that would form the subgroup would be logically joining the conditions of each node along that path using a logical AND operation. It is important to properly tune the decision trees to produce satisfactory bin classification accuracy with an appropriate number of subgroups. For our experiments, we set the minimum number of samples required for a leaf node to 1% of the total number of rows in the training data. This helps reduce the number of leaves (i.e., subgroups) and increase their overall coverage. We also set the maximum depth of the tree to be no larger than 50% of the total number of feature variables. This helps reduce the number of rules in the tree. From there, we perform minimal cost-complexity tuning[23,36] to further reduce the complexity of the tree (and by extension, the subgroups) by varying the value of α. For each iteration of 10-fold stratified cross-validation, we hold out 20% of the training data to form a validation set to guide the selection of the α value that has the largest accuracy on the validation set with the smallest number of subgroups.

To evaluate the efficacy of QUEST, we report the average and standard deviation of tree accuracy on the unseen test set, the average and standard deviation of the number of subgroups formed and the average and standard deviation of rule coverage for classifying 2 uncertainty bins. We also present the confusion matrices on the test set across the same stratified 10-fold cross-validation used to evaluate the efficacy of the underlying classification model for 3-bin uncertainty classification for the Trauma Triage data. The same test set to evaluate underlying classification performance is also used to evaluate the efficacy of bin assignment.

Identifying Performance Differences in Identified Subgroups

To discern if identified subgroups can identify patient populations for which there is a discernible performance difference, we perform the following analysis. Each subgroup is associated with one uncertainty level/bin. For each dataset, we group subgroups by their predicted uncertainty bin across each test set in 10-fold stratified cross-validation. We measure the average test set classification accuracy weighted by each rule's coverage of the test set. We weight classification accuracy by subgroup size to ensure the performance of each subgroup reflects the number of instances covered. We then perform a weighted student's T-test for samples of unequal size and unequal variance to evaluate the alternative hypotheses that lower uncertainty subgroups have higher test set classification accuracy. We report the weighted averages for level 0 uncertainty and level 1 uncertainty (for 2-bin uncertainty estimation) and p-value of the significance test for each dataset.

We also perform the following analysis to determine what relationship, if any, exists between performance of uncertainty quantification at the rejection-classification task and the efficacy of our discovered subgroups at identifying patient populations with low classification performance. Rejection-classification is one of the most accessible and primary use cases of uncertainty quantification in machine learning[9,10]. In this task, predictions with the highest uncertainty values are removed from performance analysis in increasing increments, and the resulting performance of the classification model is measured. When uncertainty is well-calibrated as an indicator of potential incorrectness, the resulting curve should be increasing with the largest rate of change occurring at the first few increments before plateauing. To quantify the quality of uncertainty, we introduce the *RC-Index*. RC-Index is defined as the area under the curve where the independent variable is the increment of removed data by uncertainty (0, 5, 10, …) and the independent variable is the associated classifier performance minus the classifier's performance with all the available data (thus, the dependent variable starts at 0). We measure the Pearson correlation coefficient *r* of the difference between the weighted classification accuracy of low uncertainty subgroups and the weighted classification accuracy of high uncertainty subgroups and the RC-Index of the dataset/classification model combination.

Tradeoff Between Number of Subgroups and the Predictability of Uncertainty Levels

Because various aspects of tree construction can be varied to produce an appropriate tree based on underlying domain knowledge, and there is not necessarily one right choice, we analyze the trade-off between tree complexity and accuracy at the bin assignment task[26]. We present plots of tree complexity (i.e., number of leaves or maximum depth) with respect to validation accuracy and perform vertical averaging (with interpolation of y-values as necessary)[37] on each of the curves to reflect the entire training process and minimize stochastic noise in the final graph. Note the independent variables: number of leaves and maximum depth are equivalent to the number of subgroups and maximum rule length, respectively.

Results and Discussion

We first discuss the classification task results for both ANNs and GBTs. This will help frame our discussions regarding uncertainty in these models. Next, we will discuss the efficacy of QUEST followed by an analysis of varying the number of subgroup. We then evaluate the meaningfulness of the subgroups by determining their efficacy at a task similar to rejection-classification.

Classification Task Results

Table 2 provides the 10-fold stratified cross-validation accuracy and F1 score for each of the datasets for the GBTs and ANNs. Performance between the two models was generally comparable; however, for all datasets but Pima Diabetes, the GBTs tended to perform slightly better across all measured metrics. This slight edge is consistent with the literature reporting that GBTs have superior performance compared to ANNs in machine learning tasks with numeric data[38]. For the Diabetes data, both models had nearly equivalent areas under their respective ROC curves. Accuracy was higher for the gradient-boosted tree model, and F1-score was higher for the neural network. For the Critical Outcome and 3-Day Readmit, we note a combination of exceptionally high accuracy with lower F1-scores which suggest that the majority class was predicted by the models.

Table 2. Datasets and model performance on underlying classification task.

Dataset	**ANN Performance**		**GBT Performance**	
	Accuracy	**F1**	**Accuracy**	**F1**
Trauma	0.85	0.73	0.85	0.74
Diabetes	0.77	0.65	0.78	0.65
Crit. Outcome	0.94	0.24	0.95	0.32
3-Day Readmit	0.97	0.05	0.97	0.09
Hosp. Pred.	0.73	0.72	0.75	0.74

Quantifying Uncertainty for Estimating Subgroup Types

Table 3 details the number of discovered subgroups as well the unweighted bin assignment accuracy and average coverage of the discovered subgroups across the test sets of 10-fold stratified cross-validation for 2 uncertainty bins, respectively. Figure 1 gives the confusion matrices for the tasks for classifying 3 uncertainty bins. Note that the confusion matrices are aggregated from each of the test sets from cross-validation. Thus, the number of data points in the confusion matrix will add up to the total size of the dataset.

Table 3. Classification accuracy of QUEST on discretized uncertainty ($n_{bins} = 2$) by dataset and classification model. Sample standard deviation is in parenthesis.

Dataset	**ANN Performance**		**GBT Performance**	
	No. Rules	**Accuracy**	**No. Rules**	**Accuracy**
Trauma	34.0 (8.58)	0.85 (0.01)	31.3 (7.96)	0.87 (0.01)
Diabetes	14.3 (0.781)	0.73 (0.04)	14.4 (0.80)	0.83 (0.05)
Crit. Outcome	30.4 (5.57)	0.86 (0.01)	23.7 (4.47)	0.90 (0.004)
3-Day Readmit	51.4 (9.22)	0.72 (0.02)	40.3 (4.24)	0.84 (0.01)
Hosp. Pred.	51.0 (10.51)	0.69 (0.03)	39.0 (7.06)	0.81 (0.003)

For classifying 2 uncertainty bins, we see that, generally, bin classification accuracy is above 80% on the test set. The exception to this is estimating uncertainty from an ANN on the Hosp. Pred. task. This task had a binary uncertainty bin estimation accuracy of 69.3%. After removing this outlier, however, average accuracy across the evaluated combinations is 82.3%. On average, coverage – which is defined as proportion of the test data the rule for which a rule is true - for the rules ranged from between <1% to approximately 22%.

For 3 uncertainty bins, we see performance degrade. Some uncertainty estimation tasks, such as Trauma Triage and Critical Outcome detection using GBTs, are able to achieve over 80% bin classification accuracy; however, after removing these outliers, average 3-bin accuracy across the evaluated combinations is 65.98%. We present the confusion matrices for the 3-bin uncertainty assignment task for Trauma Triage in Figure 1. Confusion matrices for other datasets had similar trends. Brighter colors (yellow and green) represent a higher frequency of values and darker colors (purple and blue) represent a lower frequency of values. The main diagonal consistently has the brightest shade, and classifier mistakes are primarily made to non-corner bins. Thus, mistakes are primarily the result of misclassifying to an adjacent bin, which is better than classifying a low as a high or a high as a low. On average, coverage for the rules ranged from between 1.5% to approximately 16% .

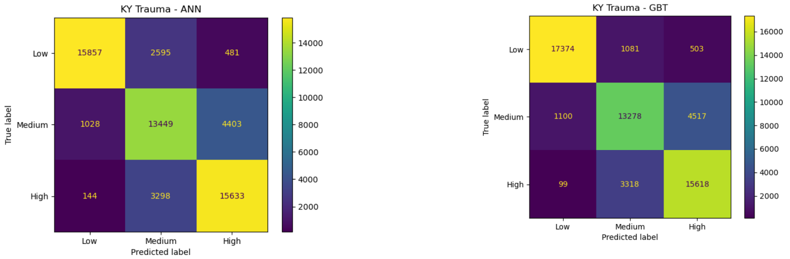

Figure 1. Confusion matrices of bin assignment ($n_{bins} = 3$).

For ANNs, bin assignment accuracy is typically lower than for GBTs. It is currently unknown if this is related to the overall quality of uncertainty for ANNs. Bin assignment accuracy of the inferred decision trees corroborates previous work[39].

Tradeoff Between Number of Subgroups and the Predictability of Uncertainty Levels

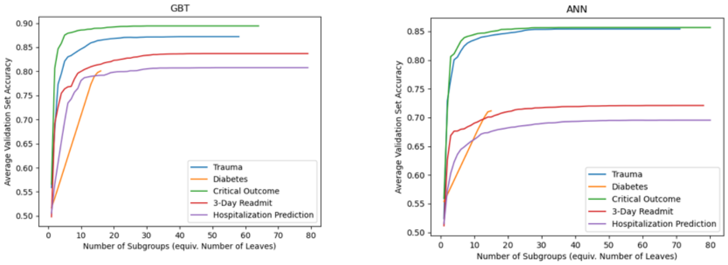

Figure 2. Plot detailing the tradeoff between number of subgroups and bin assignment accuracy. Left is for gradient-boosted tree classifier. Right is a neural network classifier.

Figure 2 presents the tradeoff between the number of subgroups and the validation accuracy using 10-fold cross-validation for uncertainty from GBTs. Results are similar for uncertainty from ANNs. First, notice that each of these curves take on a logarithmic shape. There are rapid gains in accuracy as the number of subgroups begins to increase before performance begins to plateau. This point of diminishing returns is achieved quite early for most datasets, as evidenced by the primary inflection point of the graph being located in the top left corner for every evaluated dataset but Diabetes which has a much smaller (< 20) number of subgroups evaluated. This is due to the overall size of the dataset (691 records in the training set on average) and the fact we limit the number of samples for a node to qualify as a leaf to 1% of the number of records in the training set. We suspect that if these conditions were altered, the tuning plot would begin to appear similar to tuning plots for other datasets.

This process illustrates that tuning decision trees for subgroup discovery is similar to evaluating the complexity/accuracy tradeoff of decision trees[26]. Individual practitioners can tune their model as appropriate for their domain and task. We opted for less granular tuning of tree complexity (by setting the minimum number of samples per leaf) to yield a smaller sample of subgroups to evaluate. A practitioner desiring higher accuracy may accept a larger number of subgroups, but another practitioner may want to find the smaller number of subgroups with a lower threshold of accuracy at bin assignment.

Regardless, we have shown success in utilizing decision trees to produce an effective subgroup discovery algorithm that balances the number of subgroups with performance accuracy. For our tuning scenario, we opted to balance high accuracy with a lower number of subgroups in an effort to produce a tree as close as possible to the point of diminishing returns. This resulted in trees with between 13 and 70 leaves. The Diabetes dataset, which had fewer records, tuned trees with the fewest number of leaves (13 ± 4 for ANN uncertainty; 15 ± 1 for GBT uncertainty). We notice for most datasets and model combinations, high performance at the uncertainty bin assignment task is associated with a relatively small number of subgroups.

Identifying Performance Differences in Identified Subgroups

Next, we consider the classification performance on the test set for subgroups associated with low uncertainty and subgroups associated with high uncertainty from the binary uncertainty bin assignment task. We took the average of each subgroup discovered in each fold of 10-fold stratified cross-validation and the test-set coverage that was appropriate for the tree's data fold. We weight the classification accuracy based on subgroup coverage. This allows us to consider subgroup classification performance both in the aggregate while giving appropriate credit to subgroups with more members. The result of this analysis is given in Table 4.

Table 4. Mean classification accuracy for subgroups associated with level 0 uncertainty and mean classification accuracy for subgroups associated with level 1 uncertainty along with the p-value for a weighted t-tail test to test the hypothesis that subgroups associated with lower uncertainty have higher performance. Statistical test is Welch's T-Test for samples of unequal size and unequal variance.

Dataset	**ANN Performance**			**GBT Performance**		
	Low Unc. Cls. Accuracy	**High Unc. Cls. Accuracy**	**P-Value**	**Low Unc. Cls. Accuracy**	**High Unc. Cls. Accuracy**	**P-Value**
Trauma	0.937	0.753	0.005	0.954	0.748	0.0005
Diabetes	0.869	0.660	0.060	0.899	0.660	0.048
Crit. Outcome	0.993	0.894	0.015	0.993	0.899	0.010
3-Day Readmit	0.980	0.950	0.114	0.981	0.951	0.020
Hosp. Pred.	0.710	0.754	0.721	0.842	0.665	0.002

Based on the rationale of rejection-classification, we suspect that subgroups associated with low (level 0) uncertainty will have a higher classification accuracy than subgroups associated with high (level 1) uncertainty. We see this occur for 9 of the 10 dataset/classification model combinations evaluated. Moreover, we use Welch's T-Test for samples of unequal size and unequal variance to evaluate the statistical significance of these results. For 8 of those 9, the difference in means is statistically significant with a confidence over 94%. The primary exceptions to this are both classification models on 3-Day Readmit and the ANN on Hosp. Pred. For 3-Day Readmit, there is a decrease in accuracy from less uncertain subgroups to more uncertain subgroups; however, this decrease in accuracy is not as impactful as decreases in other datasets (ranging from 9% to 22%) and struggles with being statistically significant for neural networks (confidence level of 88.6% for neural networks and 98% for GBTs). For Hosp, Pred., we observe an *increase* in accuracy as uncertainty increases.

To determine why these exceptions occur, we examine the underlying calibration of the uncertainty quantification for each dataset/model combination. Table 5 lists the difference in the means given in Table 4 as well as the *RC-Index* for the specific dataset/model, averaged across the test sets of 10-fold cross-validation. RC-Index differs from pure area under the rejection-classification curve in that values close to 0 reflect no change in performance as most uncertain data are removed; values less than 0 reflect a decrease in accuracy as most uncertain are removed and show that uncertainty is poorly calibrated; values greater than 0 reflect an increase in accuracy as most uncertain are removed. For 3-Day Readmit, the RC-Index values are the closest to 0 at 0.014 and 0.016, and for Hosp. Pred. using neural networks, the RC-Index value is negative. For the remaining dataset/model combinations, we see RC-Index values above 0. Moreover, we examine the Pearson correlation coefficient r for the difference in weighted means of classification accuracy by subgroup and associated RC-Index values and find a linear correlation with $r = 0.976$ with p-value = 1.394e-06, indicating a strong correlation between identifying well-performing and poor-performing subgroups and uncertainty calibration quality. This seems to imply that effective identification of well- and poor-performing groups is correlated to the effectiveness of uncertainty quantification. Thus, when uncertainty performs well, it can be explained using this form of subgroup discovery using decision trees.

Table 5. Difference between weighted classification accuracy for subgroups associated with low uncertainty vs subgroups associated with high uncertainty and RC-index for the given dataset/model combination. Pearson's correlation coefficient r = 0.976 with p-value = 1.394e-06

Dataset	**ANN**		**GBT**	
	Difference	**RC-Index**	**Difference**	**RC-Index**
Trauma	0.18	0.08	0.21	0.10
Diabetes	0.21	0.08	0.22	0.10
Crit. Outcome	0.10	0.04	0.09	0.04
3-Day Readmit	0.03	0.01	0.03	0.02
Hosp. Pred.	-0.04	-0.03	0.18	0.12

Conclusions

In light of recent development in uncertainty quantification for deep learning and gradient-boosted trees, we proposed and evaluated a technique -- QUEST -- to perform subgroup discovery on prediction uncertainty based on input feature values. We presented three research questions to be tested through our technique: (1) Can we identify rules that define subgroups of patients with differing levels of epistemic uncertainty well enough to accurately predict the uncertainty levels of unseen examples? and (2) Does the predicted uncertainty of the discovered subgroups correlate to the classification accuracy of patients in those subgroups? and (3) Given that a human will likely be involved in analyzing and developing plans to address selected subgroups, how many subgroups (or rules) are needed to accurately predict the uncertainty level? Our experiments confirm that we can answer questions 1 positively – more confidently for two uncertainty bins than for three, however. Our second research question can also be answered positively because there is a statistically significant difference between classification accuracy of low uncertainty bins compared to high uncertainty bins. Also, we find a strong correlation between rejection-classification performance and the difference in performance between low and high uncertainty bins. Finally, we present a trade-off between the number of discovered subgroups and the validation bin classification accuracy. We find an effective trade-off that allows for a small number of subgroups without sacrificing accuracy.

QUEST has several advantages over other error auditing approaches. First, this approach is *supervised* using a label derived from uncertainty quantification. Second, it avoids exhaustive enumeration to locate errors in a machine learning model. Its limitations are that the underlying uncertainty value must be non-zero and well-calibrated. Thus, when a classifier predicts the majority class with a unanimous probability, this technique would not be effective. Also, the extent of the calibration necessary for this technique to be effective remains uncertain.

There are extensions of this technique we plan to evaluate. First, we would like to apply recent advances in real-valued subgroup discovery to remove the need for discretizing the uncertainty values. Moreover, adjusting the granularity of the subgroup discovery could prove useful in refining the subgroups further. Finally, we would like to develop and evaluate real-world clinical scenarios in which this technique could be applied to aid clinicians as well as investigate the technique with an in-depth clinical case-study. Another avenue of future work is to evaluate QUEST using traditional subgroup discovery [40,41]

Acknowledgments. Research reported in this publication was supported by the National Library of Medicine of the National Institutes of Health under Award Number R15LM013824. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. We would also like to acknowledge the assistance of Dr. Stacy Brown at Tennessee Technological University for her input regarding the statistical tests utilized in this study.

References

1. Oakden-Rayner L, Dunnmon J, Carneiro G, Re C. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In: Proceedings of the ACM Conference on Health, Inference, and Learning. New York, NY, USA: ACM; 2020. .
2. d'Eon G, d'Eon J, Wright JR, Leyton-Brown K. The spotlight: A general method for discovering systematic errors in deep learning models. In: 2022 ACM Conference on Fairness, Accountability, and Transparency. New York, NY, USA: ACM; 2022. .
3. Pastor E, de Alfaro L, Baralis E. Identifying Biased Subgroups in Ranking and Classification. 2021 Aug.
4. Chung Y, Kraska T, Polyzotis N, Tae KH, Whang SE. Automated Data Slicing for Model Validation: A Big Data - AI Integration Approach. IEEE Trans Knowl Data Eng. 2020 Dec;32(12):2284-96.
5. Mahajan V, Venugopal VK, Murugavel M, Mahajan H. The algorithmic audit: working with vendors to validate radiology-AI algorithms—how we do it. Acad Radiol.
6. Krishnan S, Wu E. PALM: Machine Learning Explanations For Iterative Debugging. In: Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics. No. Article 4 in HILDA'17. New York, NY, USA: Association for Computing Machinery; 2017. p. 1-6.
7. Sohoni NS, Dunnmon JA, Angus G, Gu A, R´e C. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. 2020 Nov:19339-52.
8. Dwivedi R, Dave D, Naik H, Singhal S, Omer R, Patel P, et al. Explainable AI (XAI): Core Ideas, Techniques, and Solutions. ACM Comput Surv. 2023 Jan;55(9):1-33.
9. Leibig C, Allken V, Ayhan MS, Berens P, Wahl S. Leveraging uncertainty information from deep neural networks for disease detection. Scientific reports. 2017;7(1):1-14.
10. Brown KE, Talbert DA. Estimating Uncertainty in Deep Image Classification. In: AMIA; 2019. .
11. Begoli E, Bhattacharya T, Kusnezov D. The Need for Uncertainty Quantification in Machine-Assisted Medical Decision Making. Nature Machine Intelligence. 2019;1(1):20-3.
12. Kendall A, Gal Y. What uncertainties do we need in bayesian deep learning for computer vision? In: Advances

in neural information processing systems; 2017. p. 5574-84.
13. Gal Y, Ghahramani Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In: Proceedings of The 33rd International Conference on Machine Learning; 2016. p. 1050-9.
14. Malinin A, Prokhorenkova L, Ustimenko A. Uncertainty in gradient boosting via ensembles. arXiv preprint arXiv:200610562. 2020.
15. Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems; 2017. p. 6402-13.
16. Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D. Weight uncertainty in neural networks. arXiv preprint arXiv:150505424. 2015.
17. Jain M, Lahlou S, Nekoei H, Butoi V, Bertin P, Rector-Brooks J, et al. Deup: Direct epistemic uncertainty prediction. arXiv preprint arXiv:210208501. 2021.
18. Van Amersfoort J, Smith L, Teh YW, Gal Y. Uncertainty estimation using a single deep deterministic neural network. In: International conference on machine learning. PMLR; 2020. p. 9690-700.
19. Liu J, Lin Z, Padhy S, Tran D, Bedrax Weiss T, Lakshminarayanan B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. Advances in Neural Information Processing Systems. 2020;33:7498-512.
20. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research. 2014;15(1):1929-58.
21. Friedman JH. Stochastic gradient boosting. Computational statistics & data analysis. 2002;38(4):367-78.
22. Quinlan JR. Induction of decision trees. Mach Learn. 1986 Mar;1(1):81-106.
23. Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees. Routledge; 2017.
24. Webb GI, Butler S, Newlands D. On detecting differences between groups. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. KDD ’03. New York, NY, USA: Association for Computing Machinery; 2003. p. 256-65.
25. Dua D, Graff C. UCI Machine Learning Repository; 2017. Available from: http://archive.ics.uci.edu/ml.
26. Talbert DA, Talbert S. Poster: Trauma Triage in an Information Rich Environment. In: American Medical Informatics Annual Fall Symposium; 2021. p. 1848.
27. Sasser SM, et al. Guidelines for Field Triage of Injured Patients: Recommendations of the National Expert Panel on Field Triage, 2011. Morbidity and Mortality Weekly Report: Recommendations and Reports. 2012;61(1):1-20.
28. Johnson A, Bulgarelli L, Pollard T, Celi LA, Mark R, Horng S. MIMIC-IV-ED. PhysioNet. 2021.
29. Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. circulation. 2000;101(23):e215-20.
30. Xie F, Zhou J, Lee JW, Tan M, Li S, Rajnthern LS, et al. Benchmarking emergency department prediction models with machine learning and public electronic health records. Scientific Data. 2022;9(1):658.
31. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization; 2014.
32. Chollet F, et al.. Keras; 2015. Available from: https://keras.io.
33. Abadi M, et al.. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems; 2015. Software available from tensorflow.org. Available from: https://www.tensorflow.org/.
34. Dorogush AV, Ershov V, Gulin A. CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:181011363. 2018.
35. Pedregosa F, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825-30.
36. Quinlan JR. Simplifying decision trees. Int J Man Mach Stud. 1987 Sep;27(3):221-34.
37. Fawcett T. An introduction to ROC analysis; 2006.
38. Fernández-Delgado M, Cernadas E, Barro S, others. Do we need hundreds of classifiers to solve real world classification problems? The journal of machine. 2014.
39. Brown KE, Talbert DA. A Simple Direct Uncertainty Quantification Technique Based on Machine Learning Regression. In: The International FLAIRS Conference Proceedings. vol. 35; 2022. .
40. Herrera F, Carmona CJ, González P, Del Jesus MJ. An overview on subgroup discovery: foundations and applications. Knowledge and information systems. 2011;29(3):495-525.
41. Novak PK, Lavrac N, Webb GI. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. Journal of Machine Learning Research. 2009;10(2).