# Multi-Armed Bandit-Driven Active Learning

**Ravid Dimant** **Elon Dagan** **Nickolas Zaknun**
**318174315** **314734724** **208018754**

{ravid.d, eilondagan, zaknun-g}@campus.technion.ac.il
Faculty of Data and Decision Sciences, Technion, Israel

**Git Repository:** **Multi-Armed Bandit-Driven Active Learning**

## Abstract

Selecting the most effective sampling method for a given dataset and task is a significant challenge in active learning. While numerous sampling strategies exist, none are universally optimal across all datasets, and the performance of a method on one dataset offers no guarantees of similar success on another. This variability makes choosing the right sampling method both complex and costly. In this work, we propose innovative approaches leveraging Multi-Armed Bandit (MAB) algorithms to dynamically select the most promising sampling method from a pool of strategies. By adapting to the specific characteristics of the dataset and task, our MAB-based methods try to identify effective sampling strategies. As demonstrated in our experiments, these methods exhibit competitive performance across diverse datasets. While their effectiveness varies depending on the dataset, they frequently rank among the top-performing strategies and show considerable improvement over baseline methods in several cases.

## 1. Introduction

Active learning is a paradigm for improving machine learning models while minimizing labeling costs by selectively choosing the most informative data points to label. A critical step in this process is determining the right sampling method, which can influence the efficiency and performance of the learning process. Numerous sampling strategies have been proposed, such as Uncertainty Sampling and Diversity Sampling, each with its strengths and weaknesses. However, none of these methods can be guaranteed as the best choice, or even one of the best, for every dataset and task.

This variability poses a significant challenge for practitioners: the selection of an appropriate sampling method often relies on trial and error or prior knowledge about the dataset. Furthermore, the performance of a given sampling strategy on one dataset often fails to generalize to others, making the task of selection method time-consuming and resource intensive. These limitations create a pressing need for adaptive techniques that can dynamically identify the most effective sampling strategy for any given scenario.

To address this gap, we propose an innovative framework based on Multi-Armed Bandit (MAB) algorithms. Our approach leverages a pool of existing sampling methods and dynamically selects the most promising method during the active learning process. By adapting to the dataset and task in real time, our framework not only simplifies the decision-making process, but also ensures robust performance across diverse datasets.

In our project, we detail the design and implementation of our MAB-based active learning framework. We demonstrate through extensive experimentation that our MAB-based methods achieve competitive performance across various datasets, consistently ranking among the top sampling strategies. While not universally superior, these methods outperform many individual strategies in aggregated accuracy improvement, as shown in the mean and last iteration results. This work highlights the potential of adaptive techniques to address the limitations of static sampling strategies, offering a balanced and effective solution for active learning challenges.

## 2. Methodology

### 2.1 Data and Model

To evaluate the performance of our approach, we conducted experiments using eight different datasets, each containing 2 possible class labels. A detailed description of each dataset, along with their respective sources, is provided in Appendix A

- Table 2. Preprocessing was applied uniformly across all datasets. Categorical string features were converted to integer representations and numerical features were scaled to the range [0,1] using min-max normalization, ensuring that all features contributed equally to the decision process of the sampling methods.

For classification tasks, we utilized a Random Forest classifier with a fixed configuration: 25 estimators and a maximum depth of 5. While it is acknowledged that alternative algorithms could potentially yield better performance, our primary focus was on assessing the relative improvement provided by the sampling methods. Therefore, this fixed configuration was used consistently across all datasets and sampling strategies. This standardized methodology ensures that the observed differences in performance can be attributed to the sampling methods themselves rather than variations in preprocessing or model configuration.

### 2.2 Sampling Methods

We utilized seven well-known sampling methods in our active learning pipeline: *Random*, *Uncertainty*[2], *Diversity*[3], *Density Weighted Uncertainty*[4], *Margin*[5], *Query by Committee (QBC)*[6] and *metropolis hastings*[7]. Each method was implemented as a standalone sampling strategy to assess its performance individually.

In addition to these known methods, we developed a novel *feature-based* sampling method. This approach prioritizes data samples by combining feature relevance with uncertainty in predictions. It begins by identifying the features most strongly correlated with the target variable and assigns weights to these features based on their correlation strength. For each data sample, the method computes a risk score that reflects the importance of the most relevant features.

The *feature-based* method integrates these risk scores with the prediction uncertainties to derive a combined score for each sample. Samples with the highest combined scores are then selected for labeling. By combining feature relevance and predictive uncertainty, this method aims to enhance the effectiveness of active learning, particularly in predicting tasks where feature-target relationships are critical.

### 2.3 Multi-Armed Bandit (MAB)

The main idea behind the Multi-Armed Bandit (MAB) pipeline is to dynamically evaluate and adapt sampling methods, determining the most effective approach based on empirical results. It operates using the Upper Confidence Bound (UCB)[1] algorithm and treats each sampling method as an arm of the bandit. By balancing exploration and exploitation, the MAB pipeline tries to optimize the selection of sampling methods, maximizing performance over time.

In our project, we propose two approaches that utilize the Multi-Armed Bandit framework. In both approaches, the underlying mechanism for selecting samples to label is the same. This mechanism involves selecting samples one by one, which contrasts with standard approaches that select all samples to label in each iteration with a single decision. In a given iteration, the MAB pipeline iteratively selects a single sample to label at a time, until the iteration budget is exhausted. To choose a sample, the MAB pipeline selects an arm (i.e., a sampling method) using the UCB formula. Given the chosen arm, it selects the sample that received the highest score according to the corresponding sampling method represented by that arm. After a sample is chosen, the corresponding true label is obtained, and the reward for that arm is considered to be 1 if the prediction probability provided by the ML model is lower than 0.5 (i.e., the label is incorrect), and 0.001 otherwise. The choice of 0.001 is intended to avoid a reward of zero, which is relevant to the second approach described below.

It is important to note that, at any given time, the selected sample to be labeled may have already been chosen by a different arm. This does not affect the way the reward is updated and is actually desirable, as it allows the MAB to learn more while incurring the same cost. This is because the label for the sample has already been obtained, and the model has not yet been retrained with the new samples, so the iteration budget does not decrease in this case.

In our work, we propose two different approaches that utilize this framework: the *Vanilla-MAB* and the *LST-MAB*.

#### 2.3.1 **Vanilla MAB**

In the *Vanilla MAB* approach, we use a single MAB model that is initialized at the beginning of the active learning pipeline and remains in use throughout all iterations. Aside from the first iteration, during which each arm is selected once to establish an initial reward, all iterations behave the same and the same MAB model is employed across all iterations. The rewards are continually updated based on the information learned from previous iterations.

#### 2.3.2 **Long Short-Term MAB (LST-MAB)**

In the Long Short-Term MAB approach, we utilize two MAB models: one representing the long-term "memory" and the other representing the short-term "memory." The main idea behind this approach is to capture changes in the effectiveness of sampling methods after the ML model has been re-trained with additional data. The long-term MAB is similar to the Vanilla MAB approach. It is initialized once at the beginning of the active learning pipeline and is used throughout all iterations. In contrast, the short-term MAB is re-initialized at the beginning of each iteration. Each iteration (apart from the first, which is identical to the Vanilla MAB approach) consists of two phases.

In the first phase, 10% of the current iteration's budget is allocated to train the short-term MAB. In the second phase, we use the KL-divergence [8] metric to estimate the difference in the mean reward of each arm between the short-term and long terms MABs. To compute the KL-divergence, the mean rewards of each arm from both MAB models are scaled to the range [0,1] by dividing each reward by the sum of all rewards.

These normalized rewards form probability vectors, which are then used to calculate the KL-divergence score. If the KL-divergence score exceeds 0.1, we infer that the re-trained model has significantly altered the mean rewards of the arms, indicating that the older mean rewards (captured in the long-term MAB) are no longer reliable. In this case, the remaining 90% of the budget is allocated using the short-term MAB, and for future iterations, the current short-term MAB is promoted to become the long-term MAB. Conversely, if the KL-divergence score is below 0.1, the long-term MAB is deemed reliable and is used for the current iteration, as well as retained for subsequent iterations.

## 3 Experiments

To evaluate our suggested approaches, we examined their performance against a pool of sampling methods. Specifically, we compared the results of each sampling method when used independently with those of the Vanilla MAB and LST-MAB, where the strategy pool consisted of all the sampling methods. To further assess the performance of the MAB approaches, we introduced an additional baseline algorithm called *Random-MAB*. This algorithm operates similarly to the Vanilla MAB, selecting and labeling a single sample at a time, but the choice of arms (i.e., sampling methods) is completely random.

| Method / Dataset | Uncertainty | | Margin | | Feature based | | Random MAB | | Vanilla MAB | | LST MAB | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean | STD | Mean | STD | Mean | STD | Mean | STD | Mean | STD | Mean | STD |
| Apple | 83.86 | 1.97 | 83.88 | 1.96 | 83.62 | 1.86 | 83.64 | 1.86 | 83.46 | **1.81** | 83.46 | 1.90 |
| Loan | 90.76 | 1.62 | 90.77 | 1.91 | 90.84 | 1.55 | 90.83 | 1.57 | 90.80 | **1.52** | 90.74 | 1.53 |
| MB | 95.45 | 1.37 | 95.47 | 1.39 | 94.65 | 1.53 | 95.43 | **1.26** | 95.73 | 1.27 | 95.48 | 1.29 |
| Passenger | 92.03 | 1.50 | 92.07 | 1.53 | 91.94 | 1.46 | 92.03 | 1.46 | 92.02 | **1.45** | 92.03 | 1.48 |
| Diabetes | 86.74 | 1.59 | 86.74 | 1.59 | 86.70 | 1.61 | 86.70 | **1.57** | 86.75 | 1.61 | 86.74 | 1.62 |
| Employee | 82.04 | 2.03 | 821.3 | 1.90 | 82.01 | 1.95 | 82.38 | 1.99 | 82.38 | **1.85** | 82.35 | 1.91 |
| Shipping | 67.90 | **1.93** | 67.91 | 1.94 | 67.69 | 2.19 | 67.76 | 2.24 | 67.69 | 2.23 | 67.82 | 2.05 |
| Hotel | 82.19 | **2.10** | 82.30 | 2.25 | 82.41 | 2.27 | 82.56 | 2.40 | 82.56 | 2.32 | 82.77 | 2.20 |

Table 1: Overall Mean and STD Across all Trials, for Each Dataset and Best Sampling Methods

We conducted experiments using eight different datasets. For each dataset, we evaluated the prediction accuracy of the ML model throughout the active learning process and compared results based on two metrics: (1) *Mean accuracy* - The average of the accuracies obtained at the end of

each iteration and (2) *Final accuracy* - The accuracy achieved at the end of the active learning process. For each dataset, we randomly sampled 4,000 samples and split them into 45% initial training data, 45% unlabeled data, and 10% test data. The overall labeling budget was set to 66% of the unlabeled data, divided evenly across 10 iterations.

Each method used the same dataset splits (train-unlabeled-test) and identical initial ML models to minimize variability and ensure a fair comparison. The entire process was simulated 100 times for each method and dataset to ensure the reliability of the results.

## 4 Results

To compare the outcomes, for both the *mean accuracy* metric and the *final accuracy* metric, we computed the overall mean and standard deviation (STD) across all trials for each dataset and method. Additionally, to better evaluate the results, we consider in our analysis the *accuracy improvement* relative to the worst-performing method for each dataset. Specifically, each result represents the additional accuracy achieved compared to the worst-performing method. For example, if the worst method on a given dataset achieved 80% accuracy, and another method achieved 82%, their results would be considered as 0% and 2%, respectively. Figure 1 presents the results for each dataset individually for both metrics, while Figure 2 shows the overall improvement achieved across all datasets. As shown in the figures, for the mean accuracy metric, the MAB algorithms performed significantly worse than *uncertainty sampling and margin sampling*, which are dominating across almost all datasets. Based on these poor results, we primarily focus on the second metric (final accuracy) for the remainder of the paper.

For the final accuracy metric, we observe greater variation in the best-performing sampling method depending on the dataset. For example, excluding the MAB algorithms: For the *Apple* dataset, *margin sampling* and *uncertainty sampling* perform best. For the *Employee* dataset, *density weighting* performs best. And for the *Diabetes* dataset, *density sampling* performs best.

When comparing the MAB algorithms to the other sampling methods in the last iteration, we find that the MAB approaches achieve the best performance in nearly half of the cases and, in the remaining cases, perform close to the best (Figure 3). Overall, the MAB algorithms achieve the highest cumulative accuracy improvement score across all datasets: Random MAB achieves a 5.39% improvement, Vanilla MAB achieves a 5.43% improvement, and LST-MAB achieves a 5.44% improvement, as in Figure 4. Given that performance can vary drastically between trials (e.g., the worst-performing sampling method in iteration *i* may become the best in iteration *j*), we also measured the standard deviation (STD) of each sampling method's performance. These results are shown in Table 1. As seen, the Vanilla MAB algorithm demonstrates the most stability among all methods. A visualization of the confidence interval for each dataset and method can be found in Appendix B.

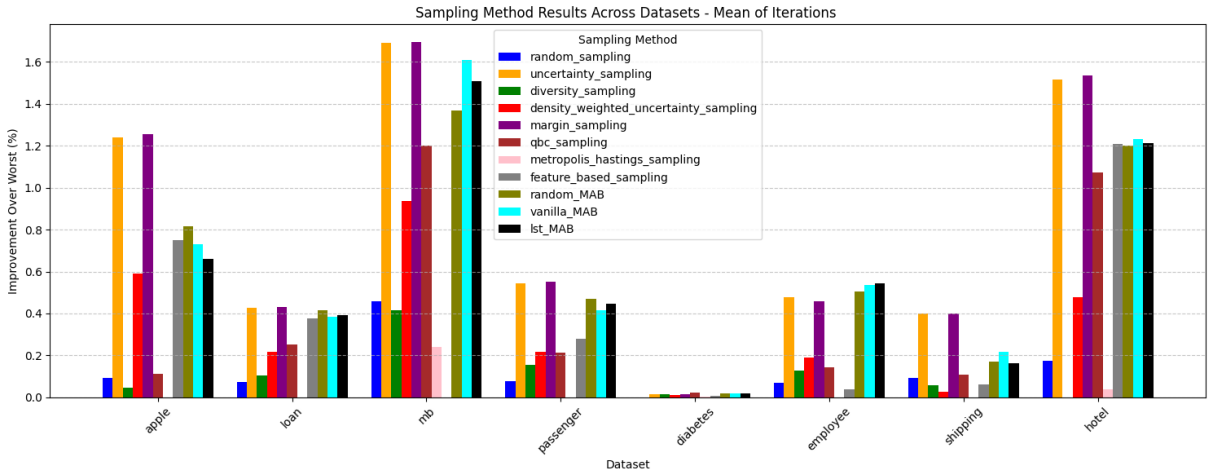



Figure 1: Improvement of Each Method Compared to The Worst One in Every Dataset

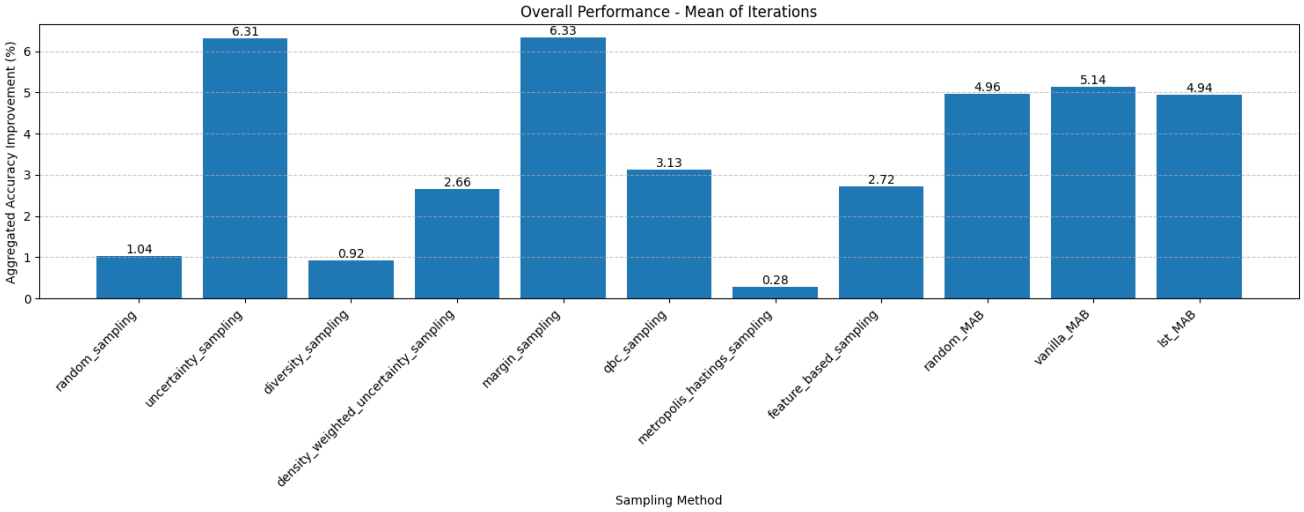



Figure 2: Overall Performance for All Sampling Methods

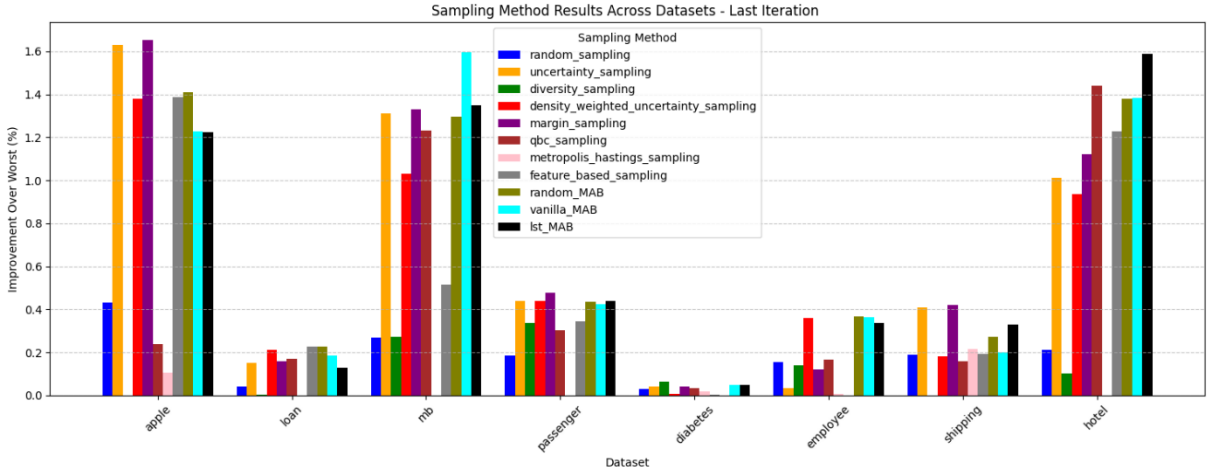



Figure 3: Improvement of Each Method Compared to The Worst One in Every Dataset – Only in Last Iteration

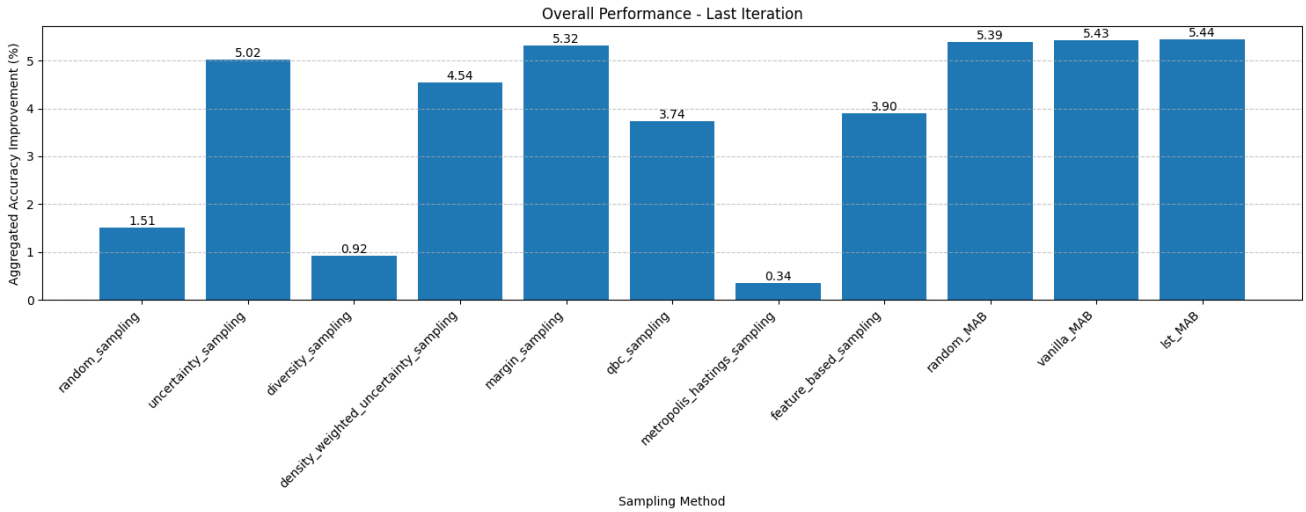



Figure 4: Overall Performance for All Sampling Methods – Only in Last Iteration

## 5 Insights and Discussion

Our work encountered several challenges, the most notable being the subtle changes observed in the results. As demonstrated, the performance differences between sampling methods are very minor. For example, even random sampling and more sophisticated sampling methods often have only small differences, typically less than only 1%. Despite this, we believe that our simulation framework managed to address these challenges effectively, yielding results that are relatively robust and accurate.

For the MAB algorithms, one unexpected finding was the strong performance of the Random MAB. We hypothesize that this might be because that even when a sample is chosen by a random sampling method, it is still the sample that this method is most confident about. We believe that the key distinctions between sampling methods may emerge not in their top-ranked samples but in their selection of less obvious choices, such as the 50th most promising sample. This suggests that the differences between sampling strategies might play a more significant role in later stages of the selection process.

Additionally, we observed that the MAB algorithms underperform in scenarios where intermediate iteration results are critical, as reflected in the mean accuracy metric. This may be due to the dominance of specific sampling methods in such cases, rendering the exploration phase of the MAB algorithms redundant and less effective.

Lastly, it is important to note that we experimented with several reward schemes during our study and found that the choice of reward significantly impacts performance. For instance, using a reward defined as $1 - p_{true}$, where $p_{true}$ is the probability assigned to the correct label resulted in poor outcomes across all datasets. This underscores the importance of carefully designing the reward mechanism to align with the objectives of active learning.

## 6 Conclusions and Future Work

While the performance differences between methods were not substantial, the MAB-based approaches did demonstrate mild success in achieving our initial objectives - They did not perform poorly on any dataset and, in many cases, even achieved the best performance. This highlights the potential of MAB algorithms as a versatile strategy in active learning settings.

However, we recognize that implementing this approach may require specific settings that are not always feasible in real-world applications. For instance, labeling samples one at a time, as required by the current framework, might not be practical in some scenarios, which might require labeling only batches of samples at a time. Despite this limitation, the approach could be adapted to the more common scenario of batch labeling. This could involve treating each iteration as a time step to select an arm, using that arm to label an entire batch, and then summing the observed rewards for evaluation and updates.

For future work, several avenues can be explored. First, there is potential to design better reward mechanisms and experiment with different MAB variants to improve performance further. Second, investigating the impact of varying data division sizes—such as the proportion of training, unlabeled, and test data—could provide insights into optimizing active learning pipelines. These steps could refine the approach and broaden its applicability across different settings.

## 7 References


1. Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multi-armed bandit problem. Machine Learning, 47(2), 235–256.
2. Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 3-12.
3. Brinker, K. (2003). Data selection for support vector machines. Proceedings of the 16th International Conference on Neural Information Processing Systems (NIPS), 59–66.
4. Settles, B., & Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 1070–1079.

5. Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee: Examining complexity gain. Proceedings of the 5th Annual Workshop on Computational Learning Theory, 287–294.

6. Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT), 287–294.

7. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. Journal of Chemical Physics, 21(6), 1087–1092.

8. Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. The Annals of Mathematical Statistics, 22(1), 79–86.

# 8 Appendices

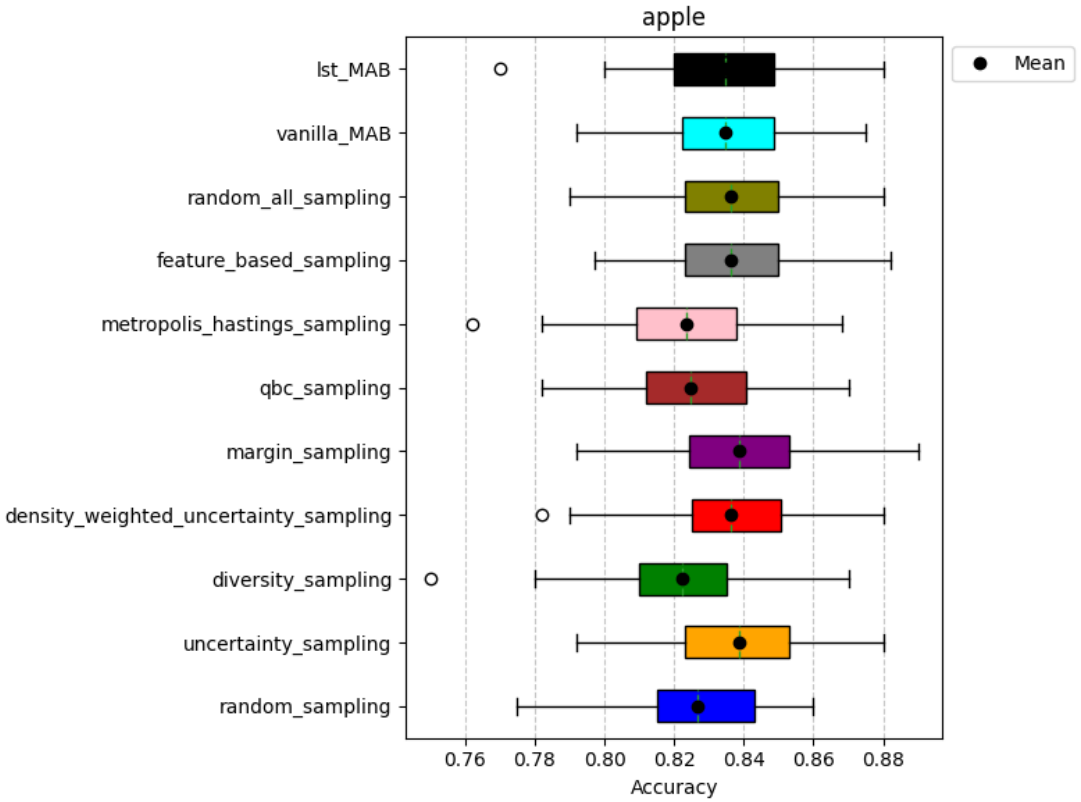
## Appendix A

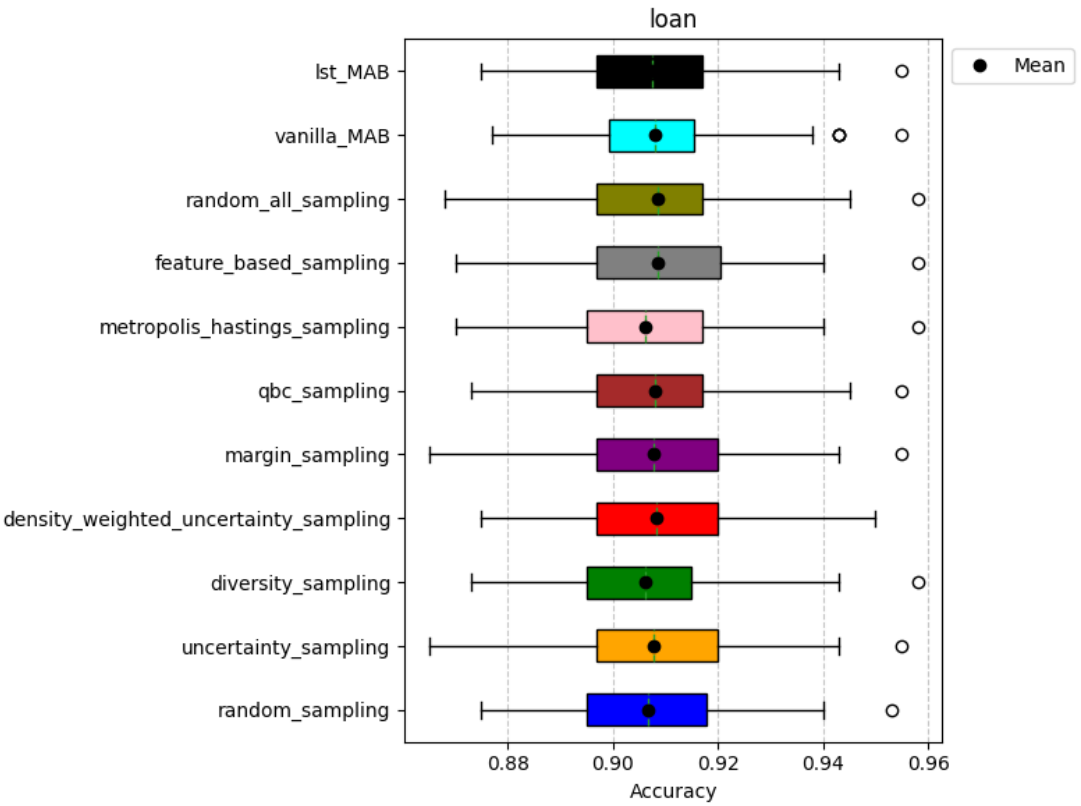
| Details / Dataset | Size (Thousands) | Number of Labels | Label 1 / Label 2 Ratio (%) | Number of Features | Task |
|---|---|---|---|---|---|
| Apple | 4 | 2 | 50 / 50 | 7 | Predict apple quality. |
| Loan | 45 | 2 | 78 / 22 | 13 | Predict loan request result. |
| MB | 52.5 | 2 | 75 / 25 | 13 | Predict individual beaches or mountains preference. |
| Passenger | 100 | 2 | 57 / 43 | 22 | Predict passenger satisfaction. |
| Diabetes | 300 | 2 | 87 / 13 | 18 | Predict if a person has diabetes. |
| Employee | 4.6 | 2 | 65 / 35 | 8 | Predict if an employee will leave or not. |
| Shipping | 11 | 2 | 60 / 40 | 10 | Predict if a shipment will arrive on time. |
| Hotel | 36 | 2 | 67 / 33 | 17 | Predict if a guest will cancel reservation. |

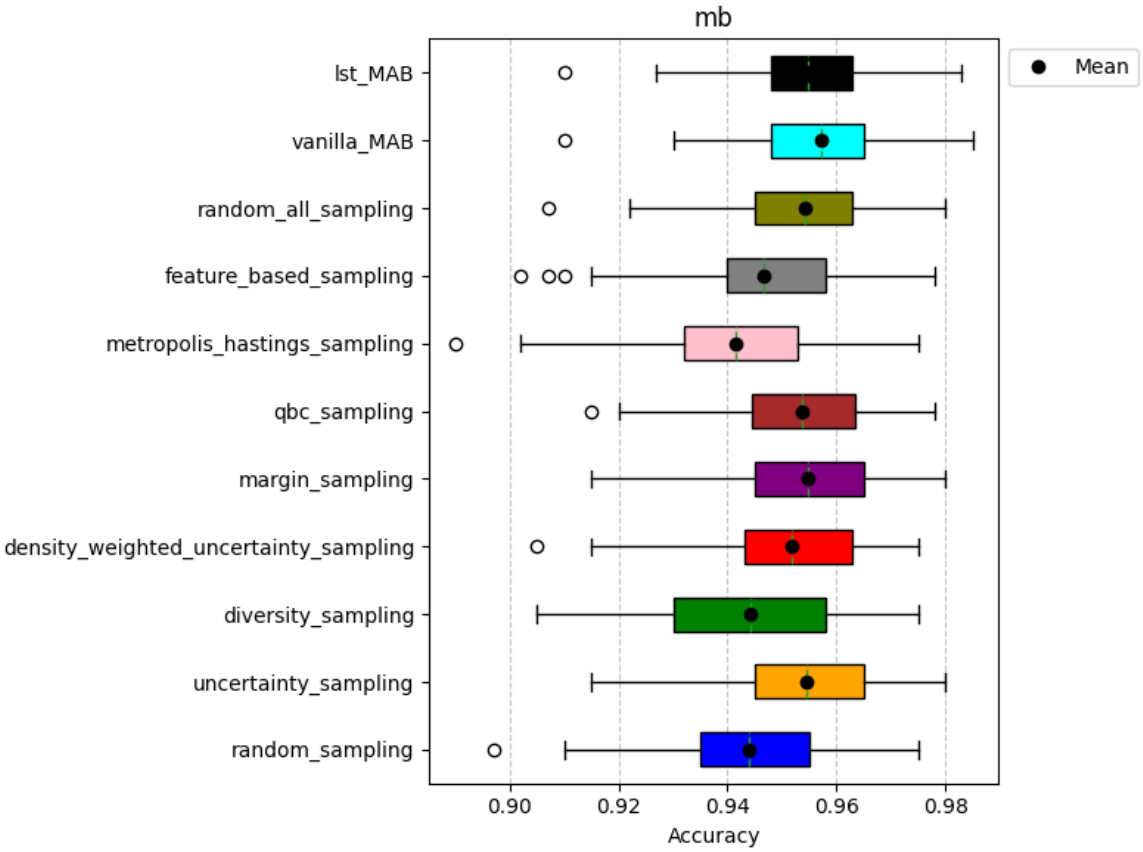
Table 2: Datasets, Their Details and Using Task

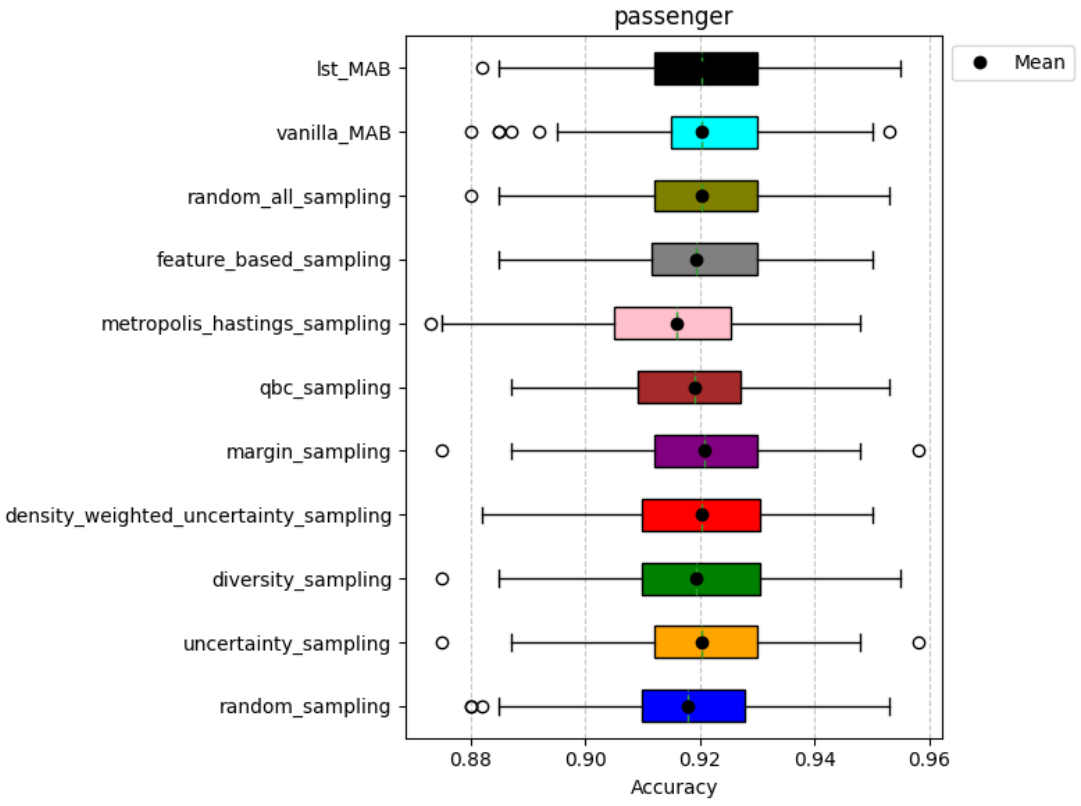
## Appendix B

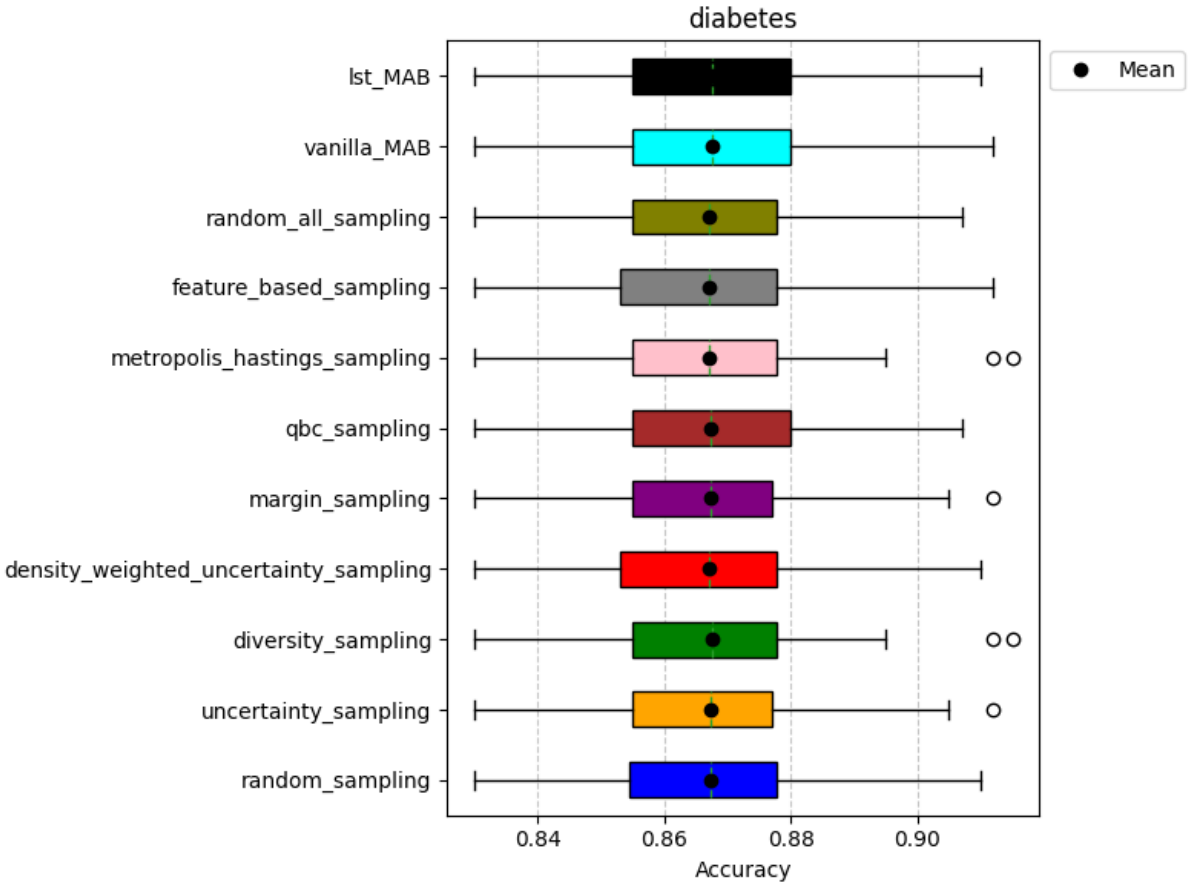
### Confidence Intervals for Last Iteration Metric in Every Dataset

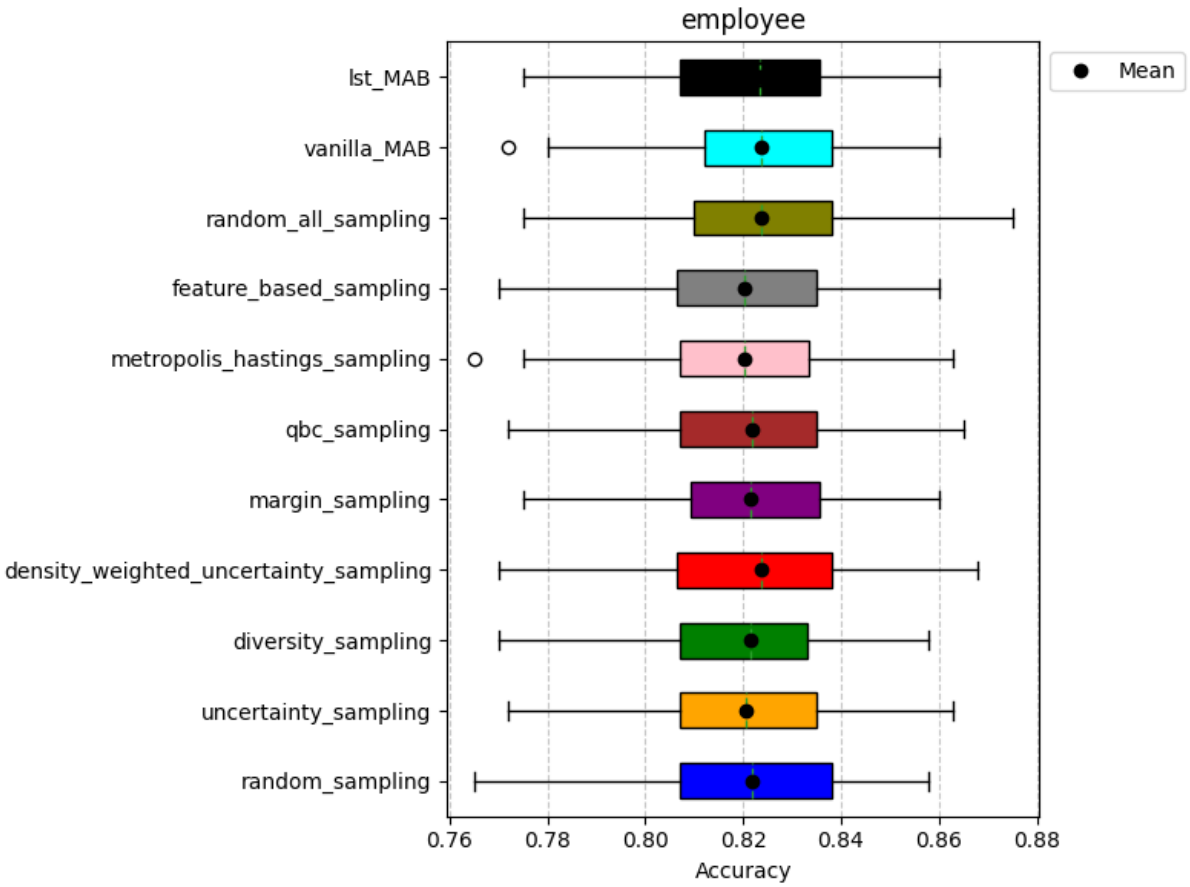


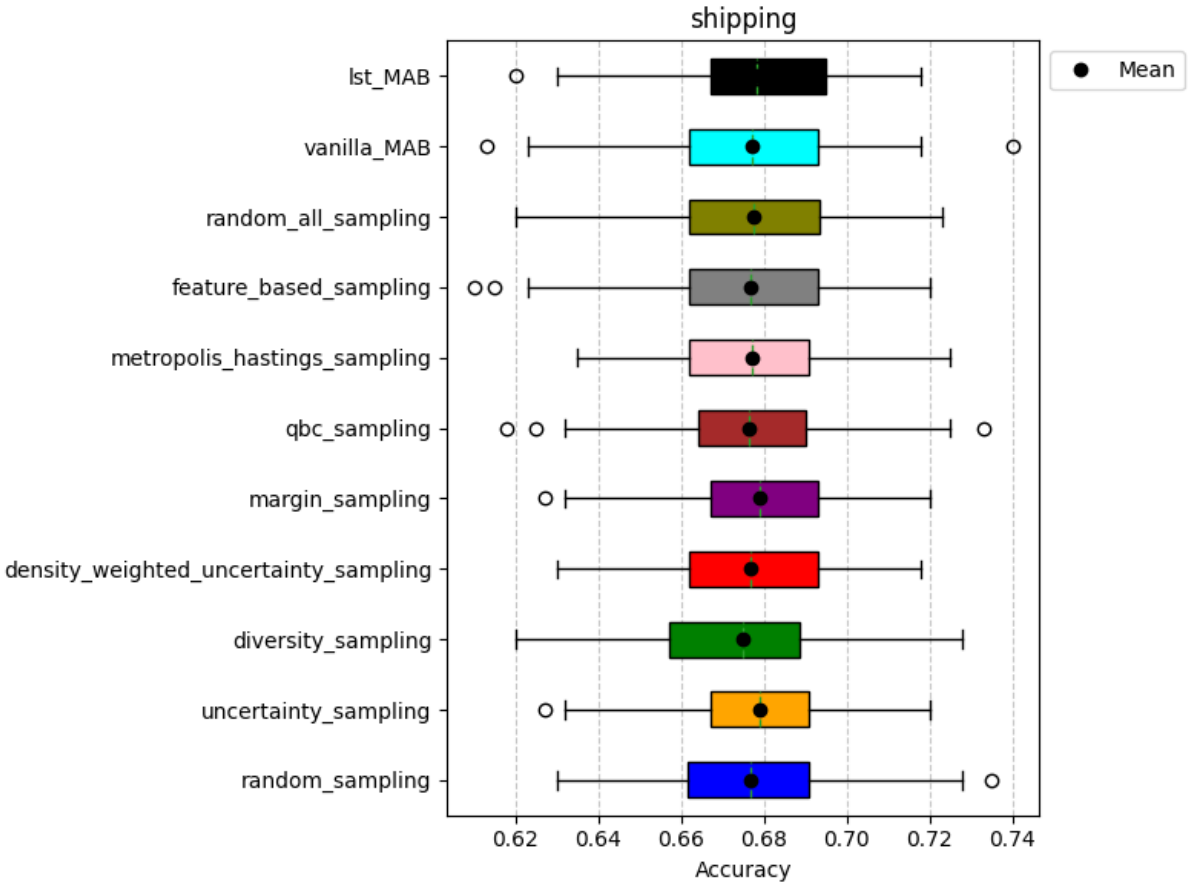



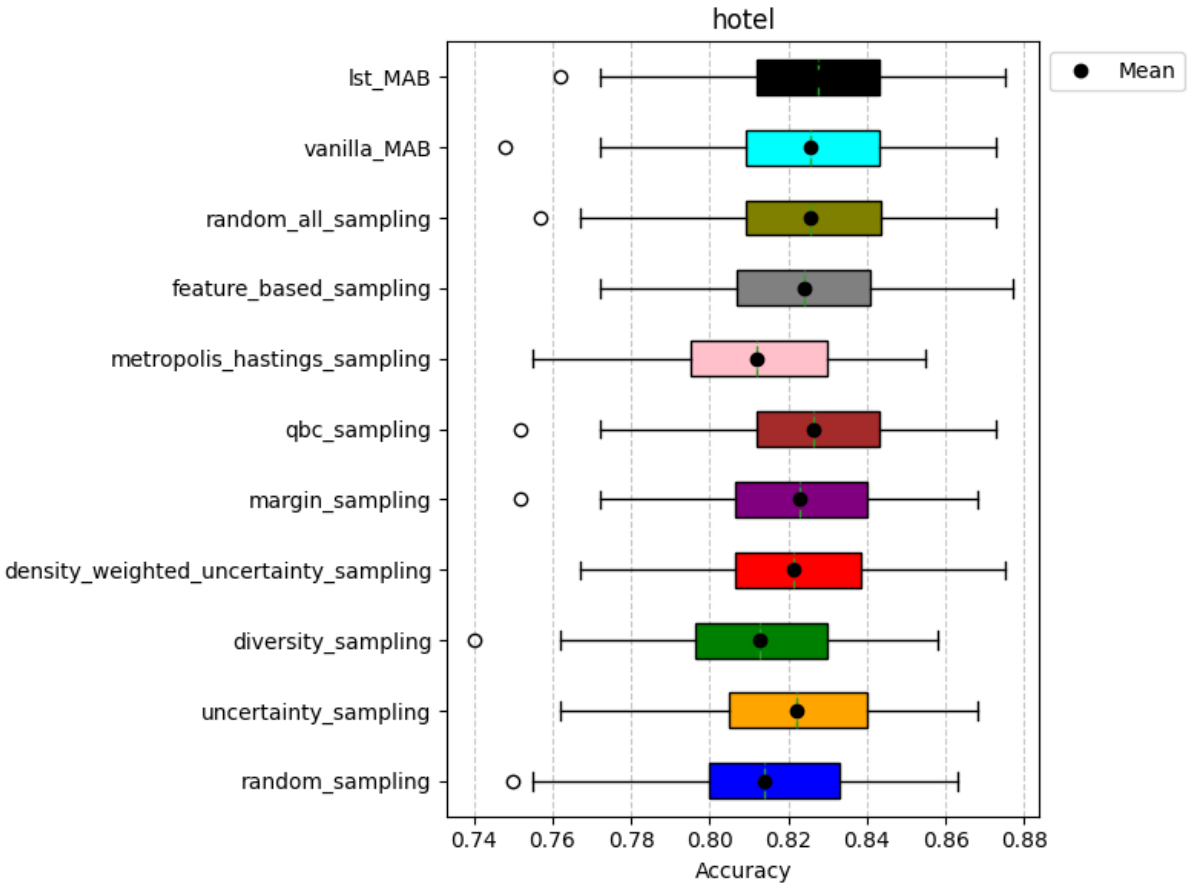

mb
lst_MAB
vanilla_MAB
random_all_sampling
feature_based_sampling
metropolis_hastings_sampling
qbc_sampling
margin_sampling
density_weighted_uncertainty_sampling
diversity_sampling
uncertainty_sampling
random_sampling
Mean
0.90
0.92
0.94
0.96
0.98
Accuracy


passenger
lst_MAB
vanilla_MAB
random_all_sampling
feature_based_sampling
metropolis_hastings_sampling
qbc_sampling
margin_sampling
density_weighted_uncertainty_sampling
diversity_sampling
uncertainty_sampling
random_sampling
Mean
0.88
0.90
0.92
0.94
0.96
Accuracy


diabetes
lst_MAB
vanilla_MAB
random_all_sampling
feature_based_sampling
metropolis_hastings_sampling
qbc_sampling
margin_sampling
density_weighted_uncertainty_sampling
diversity_sampling
uncertainty_sampling
random_sampling
Mean
0.84
0.86
0.88
0.90
Accuracy


employee
lst_MAB
vanilla_MAB
random_all_sampling
feature_based_sampling
metropolis_hastings_sampling
qbc_sampling
margin_sampling
density_weighted_uncertainty_sampling
diversity_sampling
uncertainty_sampling
random_sampling
Mean
0.76
0.78
0.80
0.82
0.84
0.86
0.88
Accuracy


shipping
lst_MAB
vanilla_MAB
random_all_sampling
feature_based_sampling
metropolis_hastings_sampling
qbc_sampling
margin_sampling
density_weighted_uncertainty_sampling
diversity_sampling
uncertainty_sampling
random_sampling
Mean
0.62
0.64
0.66
0.68
0.70
0.72
0.74
Accuracy


hotel
lst_MAB
vanilla_MAB
random_all_sampling
feature_based_sampling
metropolis_hastings_sampling
qbc_sampling
margin_sampling
density_weighted_uncertainty_sampling
diversity_sampling
uncertainty_sampling
random_sampling
Mean
0.74
0.76
0.78
0.80
0.82
0.84
0.86
0.88
Accuracy