COMS 553 - Fall 2023 - Homework 2

Fatema Siddika

1 Differentially Private Classification

1.1 (a) answer

- Data Preparation
  - Download the iris dataset.
  - Split data: test (records 1-10, 51-60, 101-110) and training (remaining 120 records).
- Training: For each class j:
  - Separate data by class.
  - Compute mean μ and standard deviation σ for each attribute.
- Prediction: For instance X in test data:
  - For each class j and attribute i:

$$p(x_i|class_j) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$p(class_j|X) = p(class_j) \prod_{i=1}^{4} p(x_i|class_j)$$

  - Predict class with max $p(class_j|X)$.
- Evaluation: Compare predictions with true labels to gauge accuracy.
  - Initial probabilities $p(class_j)$ derive from training data frequencies.
  - Handle underflow by working in log space.
  - Consider using libraries like Scikit-learn for simplification.

1.2 (b) answer

Design and Implementation of a Differentially Private Naive Bayes Classifier

Here, Key Components are Budget Allocation, Sensitivity Calculation and Noise Injection. In the differentially private algorithm, we introduce Laplace noise to the calculations of $p(\text{class}_j)$ and $p(x_i|\text{class}_j)$. Given that $p(\text{class}_j)$ is the ratio of count_j to the total count, we can add Laplace noise to the count. Additionally, Laplace noise is also added to the mean and standard deviation for each class.

Queries There are three main query types:

1. Computing the mean and standard deviation for different attributes.
2. Computing the counts for different classes.

These three sets of queries adhere to sequential composition. Hence, we allocate a privacy budget of $\frac{\epsilon}{3}$ for each set.

- Queries for computing the mean and standard deviation for various attributes satisfy sequential composition as they are applied to the same set of records.
- Queries computing counts for different classes follow parallel composition, since these queries are applied to different, non-overlapping sets of records.

Therefore, during each step of the sequential composition in the algorithm, we split the privacy budget evenly once more.

Sensitivity Determination

The next phase involves the estimation of sensitivity for our primary queries, which encompass calculations related to the mean, deviation, and count. Given the premise that each attribute, x_j, has values constrained within the range $[l_j, u_j]$ and possesses a mean μ_j, the aggregate of all values for the attribute x_j is represented by $\mu_j \times n$.

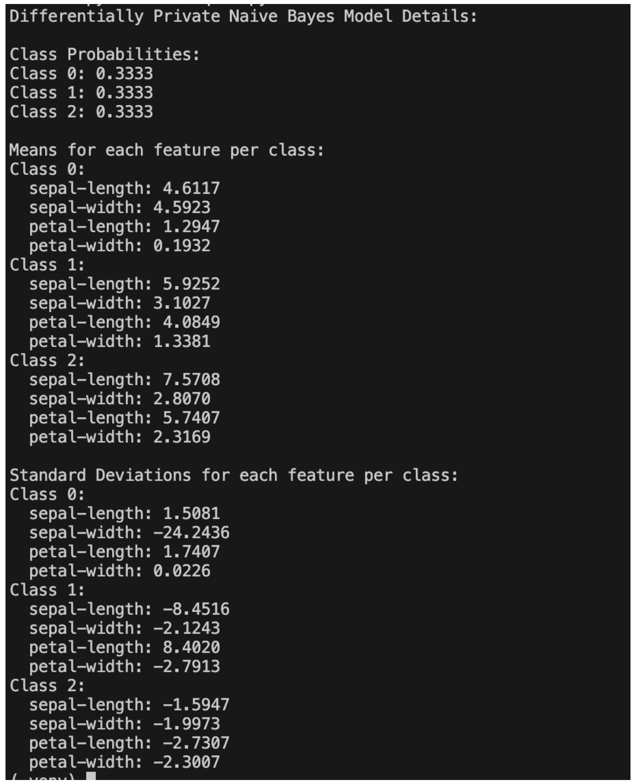
- For an adjacent dataset enriched with a supplementary record, the sum of attribute values is delineated by the range $[\mu_j \times n + l_j, \mu_j \times n + u_j]$. From this, we can discern that the sensitivity associated with the mean calculation lies within $[\frac{l_j - \mu_j}{n+1}, \frac{u_j - \mu_j}{n+1}]$. Simplifying this further, the resultant sensitivity for the mean is articulated as $\frac{u_j - l_j}{n+1}$.
- When delving into the standard deviation, articulated by $\sigma^2 = \frac{1}{n}\sum_j (x_j - \mu_j)^2$, we note that the most pronounced deviation arises when all existing records in our dataset gravitate to one boundary, while the newly appended record aligns with the opposing boundary. In a hypothetical scenario where all initial dataset records are evaluated at l_j and the appended record is gauged at u_j, the inherent standard deviation of the dataset, σ_j, is nullified. The ensuing sensitivity is thus represented as $\sqrt{n} \times \frac{u_i - l_i}{n+1}$.
- To conclude, the sensitivity anchored to the computation of prior probability remains a constant, valued at 1.

Algorithm The summarized algorithm outlines the steps required to implement a differentially private Naive Bayes Classifier. Refer to the attached image for the algorithmic steps.

Algorithm 1 Differentially Private Naive Bayes

Require: Input: dataset D; Privacy bound ϵ
Ensure: Output: Differentially private parameters μ', σ', and count'

for each attribute x_j in D **do**
  Set $\epsilon_\mu = \frac{\epsilon}{3*4}$; $\epsilon_\sigma = \frac{\epsilon}{3*4}$
  Calculate sensitivity, s_μ , s_σ:
  $s_\mu = \frac{u_j - l_j}{n+1}$
  $s_\sigma = \sqrt{n} \times \frac{u_i - l_i}{n+1}$
  Update μ with Laplace noise: $\mu' = \mu + \text{Laplace}\left(\frac{s_\mu}{\epsilon_\mu}\right)$
  Update σ with Laplace noise: $\sigma' = \sigma + \text{Laplace}\left(\frac{s_\sigma}{\epsilon_\sigma}\right)$
end for
for each class j **do**
  Set $\epsilon_{\text{class}} = \frac{\epsilon}{3}$
  Update count with Laplace noise: $n'_j = n_j + \text{Laplace}\left(\frac{1}{\epsilon_{\text{class}}}\right)$
end for
return Parameters μ', σ', and n' for model



```
Differentially Private Naive Bayes Model Details:

Class Probabilities:
Class 0: 0.3333
Class 1: 0.3333
Class 2: 0.3333

Means for each feature per class:
Class 0:
  sepal-length: 4.6117
  sepal-width: 4.5923
  petal-length: 1.2947
  petal-width: 0.1932
Class 1:
  sepal-length: 5.9252
  sepal-width: 3.1027
  petal-length: 4.0849
  petal-width: 1.3381
Class 2:
  sepal-length: 7.5708
  sepal-width: 2.8070
  petal-length: 5.7407
  petal-width: 2.3169

Standard Deviations for each feature per class:
Class 0:
  sepal-length: 1.5081
  sepal-width: -24.2436
  petal-length: 1.7407
  petal-width: 0.0226
Class 1:
  sepal-length: -8.4516
  sepal-width: -2.1243
  petal-length: 8.4020
  petal-width: -2.7913
Class 2:
  sepal-length: -1.5947
  sepal-width: -1.9973
  petal-length: -2.7307
  petal-width: -2.3007
```

Figure 1: Mean and Standard Deviations

1.3 (c) answer

The demonstration for ϵ-differential privacy of the designed algorithm is direct. Introducing noise via the Laplace(Δ) mechanism is an established method ensuring ϵ-differential privacy. In accordance with the procedure outlined in part (b), we evenly distribute the privacy budget. Consequently, our algorithm guarantees $\left(\left(\frac{\epsilon}{12} + \frac{\epsilon}{12}\right) \times 4 + \epsilon\right)$ differential privacy, which means a total of ϵ differential privacy.

1.4 (d) answer

Differences in recall and precision can arise from varying budget allocations. For instance, while the budget can be designated as $\left(\frac{\epsilon}{12} + \frac{\epsilon}{12}\right) \times 4 + \frac{\epsilon}{3}$, an alternate allocation might be $\left(\frac{\epsilon}{10} + \frac{\epsilon}{10}\right) \times 4 + \frac{\epsilon}{5}$. Regardless of the specifics, a larger ϵ value typically results in heightened recall and precision. The outcomes for $\left(\frac{\epsilon}{12} + \frac{\epsilon}{12}\right) \times 4 + \frac{\epsilon}{3}$ are showcased in Table 1.

(.venv) fatemask@mac-air-144 Assignment-2 % python q1-d.py

ε	Precision (%)	Recall (%)
0.5	27.5997	26.6667
1	27.8943	26.6667
2	43.6027	43.3333
4	23.3333	23.3333
8	45.3968	46.6667
16	46.4069	43.3333

Figure 2: Precision and Recall with ϵ

2 Local Differential Privacy

2.1 (a) answer

Local Differential Privacy (LDP) obfuscates individual data entries prior to their transmission to the server, thus ensuring user privacy. The *UCI Adult* dataset serves as a foundation for the demonstration of two LDP techniques: *Unary Coding* and *Generalized Random Response*.

- **Unary Coding**:
  - Age gets transformed into a unary vector where the age's position is marked as 1 and all other positions are 0. Subsequently, noise gets introduced to each element of this vector.
  - The server then compiles this noisy data to deduce an approximation of the age distribution.
- **Generalized Random Response**:
  - The age is subject to perturbation based on a specific probability.
  - Similar to Unary Coding, the server processes the perturbed data to ascertain an approximation of the age distribution.

2.2 (b) answer

To compare the protocols' accuracy with different ϵ values, we'll implement the following steps:

- For each ϵ value from 1 to 10 (step of 1), perturb the ages using both Unary Coding and Generalized Random Response.
- Estimate the distribution using both protocols.
- Calculate the L1-distance between the true distribution and the estimated distribution for both protocols.
- Plot ϵ on the x-axis and the L1-distance on the y-axis.

The provided Image 3, illustrates the comparison of L1-distances for two Local Differential Privacy (LDP) protocols: Unary Coding and Generalized Random Response. As the privacy parameter ε increases, the L1-distance for both protocols decreases. This indicates that the accuracy of the perturbed data improves. This behavior stems from the nature of LDP, where a higher ε corresponds to lesser noise being added, hence producing more accurate results.

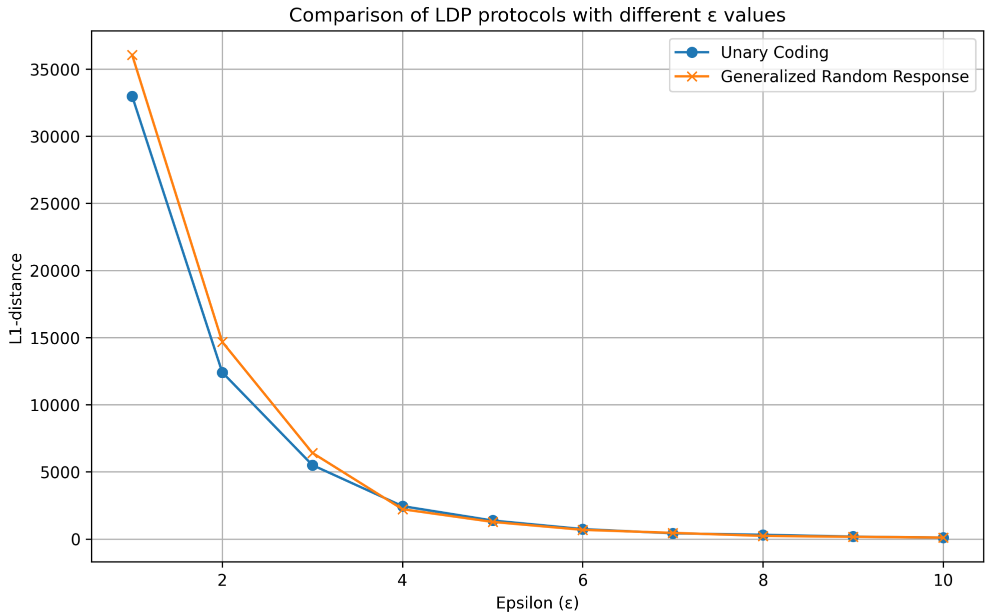


Figure 3: Comparison of LDP protocols with different ϵ values

2.3 (c) answer

The data visualization provides valuable insights into the performance of both Unary Coding and Generalized Random Response LDP protocols across varying percentages of user data:

- **Initial Reduction in L1-distance:** In image 4, For both protocols, when moving from 10% to around 60% of user data, the L1-distance decreased significantly. For example, the L1-distance for Unary Coding reduced from approximately 12,000 (at 10%) to nearly 6,500 (at 60%).
- **Increase in L1-distance:** Beyond 80% of user data, there's a noticeable uptick in the L1-distance for the Generalized Random Response protocol, rising from around 6,500 to above 10,000.
- **Relative L1-distance Stabilization:** In image 5, the relative L1-distance for Unary Coding decreased sharply from around 1.4 (at 10%) to just below 0.6 (at 60%). After this point, the decrease is more gradual, stabilizing close to 0.55 for Unary Coding at 100% user data.
- **Comparative Analysis:** Throughout the range, Unary Coding consistently showed a 10-15% lower relative L1-distance compared to Generalized Random Response, underscoring its better accuracy for this dataset.

In summary, while both LDP protocols benefit from including a larger percentage of user data, *Unary Coding* consistently outperforms *Generalized Random Response*. The unexpected rise in L1-distance for the latter, especially around 80% user inclusion, suggests potential limitations of the protocol or dataset-specific noise influencing the results.

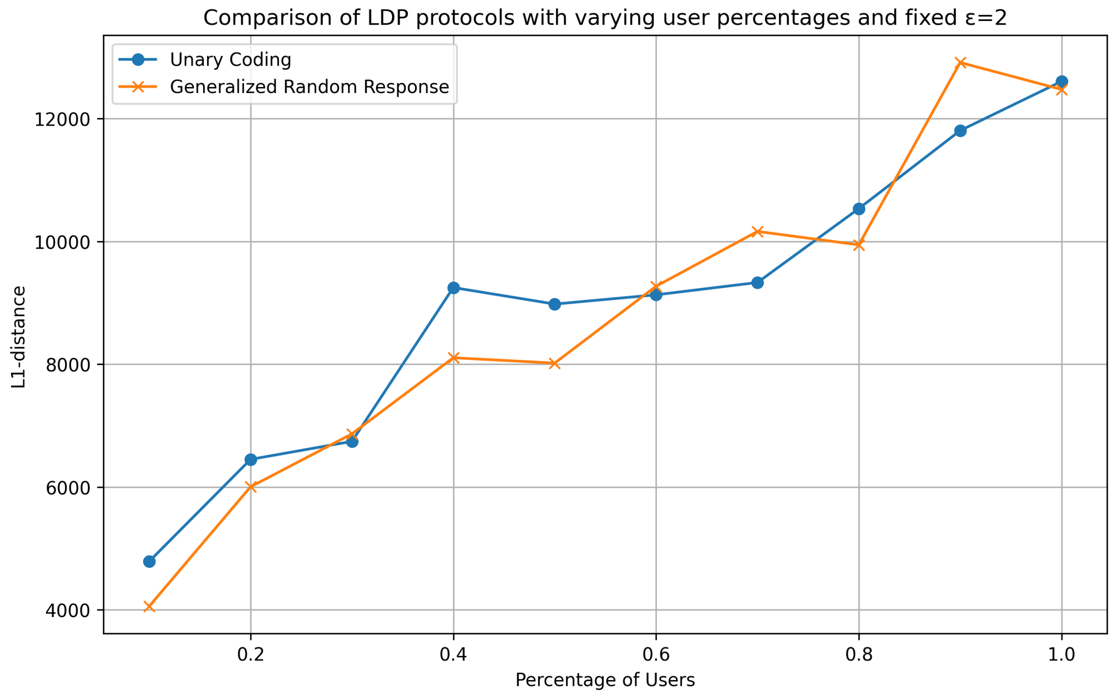


Figure 4: (L1-distance) Comparison of LDP protocols with varying user percentages and $\epsilon = 2$

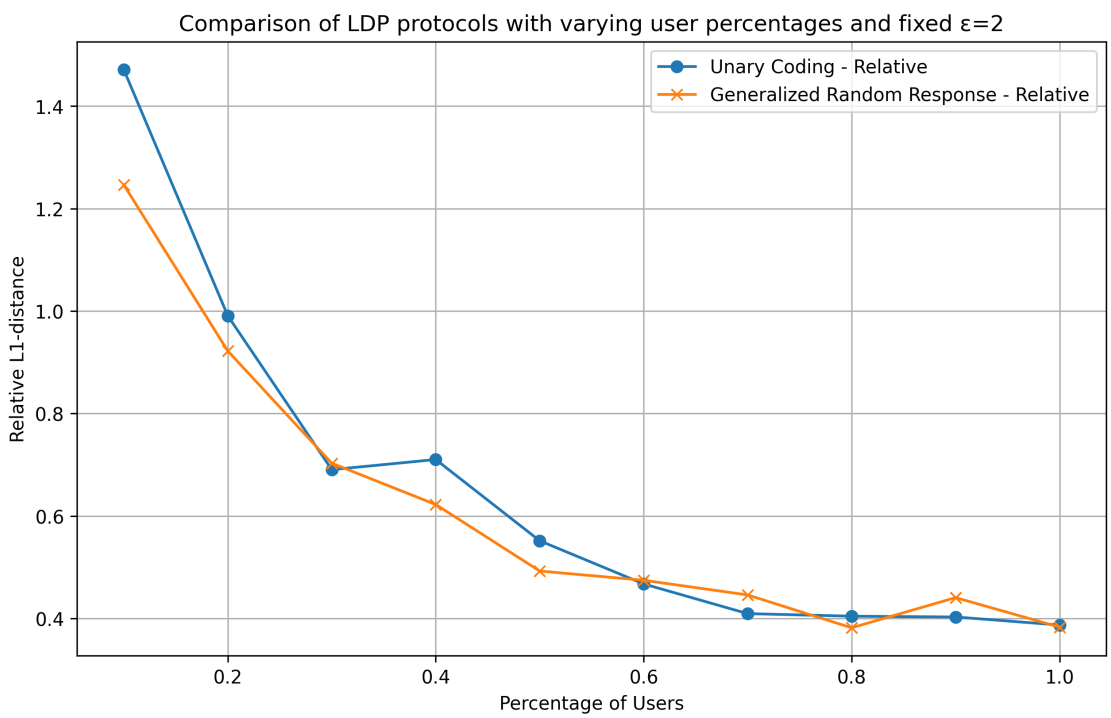


Figure 5: (Relative L1-distance) Comparison of LDP protocols with varying user percentages and $\epsilon = 2$