# Implementation and evaluation of $\epsilon$-Differential Privacy algorithms

Claudio Spiess

March 24, 2023

**Abstract**

Protecting privacy is a critical task for data scientists and ML practitioners. Throughout the course, we have learned about a slew of privacy techniques and concepts including k-anonymity and various applications of differential privacy (DP). We find that in practice, DP algorithms are difficult to use as they lack integration with common Python data science tools such as *pandas*. In this work we contribute a Python implementation [1] of algorithms we have explored in class, and perform an empirical evaluation on the effect of differential privacy algorithms on dataset utility in terms of predictive power for various classifiers.

## 1 Introduction & Motivation

Data scientists and ML practitioners use enormous amounts of data to derive insights and train machine learning models. A large amount of this data is likely to be sensitive: personally-identifying information such as location, date of birth, gender identity, or medical in nature. The Python ecosystem is likely the most popular choice in industry for analyzing and modeling large amounts of data. Yet, we find a lack of easy to use or built-in privacy preserving methods in common Python packages. One of the most common packages for data manipulation and analysis is *pandas*. As of the writing of this manuscript, it has over 2.6 billion downloads [1]. If privacy preserving techniques were more accessible to practitioners, we hope that they would be used more frequently and as a result, increase net privacy.

Motivated by this problem, we present *PyPryvacy*, a Python package for $\epsilon$-Differential Privacy, and $(\epsilon, \delta)$-Differential Privacy algorithms. Unlike existing DP implementations, PyPryvacy integrates directly as a *pandas* extension. The *pandas* library exposes two core constructs to the programmer: *dataframe* and *series*. A series is a one dimensional array with axis labels. A dataframe is a data structure that contains labeled axes (rows and columns), and allows arithmetic operations on both rows and columns. It can be thought of as a container for labeled Series objects. These core data structures expose common operations on the data they contain: mean, count, sum, etc. These data structures are commonly used and accepted as inputs for machine learning packages such as scikit-learn. Therefore, adding privacy preserving operations to pandas would allow practitioners to easily add DP to their data processing pipelines.

To this end, PyPryvacy registers itself as a pandas extension for both Dataframes and Series. It adds a new member, called an "accessor" internally, to both of these data structures named "private", which exposes DP versions of common operations. Each private operation requires an argument "mechanism". PyPryvacy implements two mechanisms: $\epsilon$-DP Laplacian, and $(\epsilon, \delta)-$DP Gaussian. The practitioner must instantiate either one of the mechanisms, and specify the $\epsilon$, and in the case of $(\epsilon, \delta)-$DP, the $\delta$ as well. The mechanism provides the primitives for sampling noise from Laplacian or Gaussian distributions. By passing the mechanism to the private Dataframe or Series accessor, the method incorporates noise from the mechanism with the appropriate sensitivity for the operation. The accessor also exposes a primitive "noise" method. Figure 1 presents examples of the package in use.

```python
import numpy as np
import pandas as pd

# PyPryvacy
import pypryvacy as pr
```

[1] https://github.com/claudiosv/pypryvacy

```
df = pd.DataFrame({"longitude": np.linspace(0, 10, num=5), "latitude": np.linspace(0, 20, num=5)})

#    longitude  latitude
# 0        0.0       0.0
# 1        2.5       5.0
# 2        5.0      10.0
# 3        7.5      15.0
# 4       10.0      20.0

df.longitude.mean()
# 5.0

lap = pr.LaplaceMechanism(epsilon=1)
gauss = pr.GaussianMechanism(epsilon=1, delta=0.05)

df.longitude.private.mean(mechanism=lap)
# 4.861801498972586

# Each call is a new sample
df.longitude.private.mean(mechanism=lap)
# 4.927190360876554

# Noise an entire dataframe, without need to specify columns!
df_priv = df.private.noise(mechanism=lap)
#    longitude   latitude
# 0   2.284351   2.026169
# 1   3.645227   3.253937
# 2   6.068457  10.474715
# 3   8.633906  15.564795
# 4  10.120258  21.111389
```

Listing 1: PyPryvacy example

## 2 Background

**Note** We refer to the lecture slides [2, 3] and Dwork et al. [4] for more in depth coverage and analysis of differential privacy. We acknowledge that the goal of the report is not to rewrite the lecture slides, but find that some background is appropriate for other readers.

The following definitions and notes follow from Lecture 7 [2] and Lecture 9 [3]. Differential privacy formalizes the intuition that the presence or absence of any single record in a data set should be unnoticeable when looking at the responses returned for the queries.

**Definition 1 (Differential Privacy)** *A randomized function $M$ gives $\epsilon$-differential privacy if for all databases $X$ and $X'$ differing in at most one element, and all subsets of outputs $S \subseteq range(M)$,*

$$P\{\mathcal{M}(X) \in S\} \leq e^{\varepsilon} \cdot P\{\mathcal{M}(X') \in S\}$$

- Differential privacy is a definition, not an algorithm.
- Differential privacy is a condition on the algorithm not on the data per se.
- A small $\epsilon$ corresponds to strong privacy, degrading as $\epsilon$ increases.
- Rule of thumb: $\epsilon \in [\frac{1}{2}, 4]$ gives reasonable level privacy guarantee.
- In practice companies often use much larger values for $\epsilon$. (Apple: $\epsilon = 6$, $\epsilon = 14$, Google: $\epsilon = 9$, Census: $\epsilon = 19$)
- Differential privacy prevents many of the types of attacks we discussed in class, such as linkage attacks and reconstruction attacks – but only if we assume a small enough $\epsilon$.

**Definition 2 (Laplacian Mechanism)**

$$\mathcal{M}(x, f(\cdot), \varepsilon) = f(x) + (y_1, \ldots, y_k)$$

*where the $y_i$ are independent Laplace$(\Delta/\epsilon)$ random variables.*

**Definition 3 (Gaussian Mechanism)**

$$\mathcal{M}(x, f(\cdot), \varepsilon) = f(x) + (y_1, \ldots, y_k)$$

*where the* $y_i$ *are independent* $\mathcal{N}\left(0, 2\ln(1.25/\delta)\Delta_2^2/\varepsilon^2\right)$ *random variables.*

Tying the background together, we see the context of the work that follows: we compare machine learning model performance on varying $\epsilon$ values for the Laplacian mechanisms, using our PyPryvacy implementation. The mechanism, Laplacian or Gaussian, is the $M$ in the definitions. We note that our evaluation of the Gaussian mechanism was relegated to the appendix and not examined further.

# 3 Methodology

## 3.1 Datasets

In this work, we investigate five real-world datasets. The datasets were retrieved from the "UCI Machine Learning Repository" as linked by Fair ML book [5] recommended by Prof. Strohmer, and from Kaggle.

**Cervical Cancer (Risk Factors)** [6] This dataset contains diagnosis of cervical cancer and risk factors for 858 patients from Caracas, Venezuela. It contains highly sensitive information such as number of sexual partners, contraceptive choice, and sexually transmitted diseases. As the survey allowed participants to skip questions they did not feel comfortable answering, we replace all missing values with the median value for that variable, as many surveys had at least one missing response and thus dropping would reduce the dataset size significantly. The target variables for this dataset are binary cervical cancer test results (Hinselmann, Schiller, Cytology, and Biopsy). In our experiments, we target the "Biopsy" variable. We investigate the predictive power of the dataset and find that correlation between target variables and predictor variables is low, and thus hypothesize that differential privacy will not have a significant impact, as utility is already low.

**Adult Income (Census)** [7] This dataset was used in homework two, and contains ~48k records of adults from the 1994 census, and whether their income is greater than or equal to \$50K/yr. The target variable is whether the individual earns more or less than \$50K/yr. The predictor variables include occupation, age, sex, and hours worked per week. These are inherently sensitive. For missing values, we apply the same method as above of filling in median values.

**Contraceptive Method Choice** [8] This dataset is a subset of a 1987 survey of married women in Indonesia, who were asked about their contraceptive use and a number of questions such as their religious belief and husband's occupation. The goal is to utilize demographic and socio-economic predictor variables to forecast a woman's current preference for contraceptive methods, categorizing them as either no use, long-term methods, or short-term methods. This is the only multi-class classification problem we investigate. Given the socioeconomic, individual, and intimate nature of this survey, the responses are sensitive.

**California Housing Prices** [9] **originally** [10] This dataset is based off the 1990 California census and contains specific houses and their value, and demographic information such as average household median income in the area. The natural target variable is the median house value, which unlike the other datasets, is continuous. The longitude, latitude, and area median income are some particularly sensitive variables. As with the above datasets, we fill in the median value for missing values. We also drop the "ocean_proximity" variable as we did not discretize it.

**Mammographic Mass** [11] This dataset contains 961 mammographic masses, which includes the patient's age and details about the mass such as shape, margin, density, and severity. The goal is to predict severity, i.e. benign or malignant. In this dataset, we simply drop any rows with missing values as opposed to filling in median values or otherwise.

### 3.2 Models

For the classification datasets (Mammographic, Cervical, Contraceptive, and Adult Income), we evaluate four models: k-Nearest Neighbors, Support Vector Machine Classifier (SVC), Random Forest (RF), and Logistic Regression. For the continuous target variable dataset (California Housing), we evaluate k-Nearest Neighbors Regressor, Support Vector Machine Regressor (SVR), Random Forest Regressor, and Linear Regression.

For each dataset, we prepare the data by resolving missing values and splitting into a train/test (80/20) split to be used across models. We then apply the Laplacian mechanism to the train split of each dataset with

$$\epsilon \in \{\frac{1}{100}, \frac{1}{2}, \frac{3}{4}, 1, 2, 3, 4, 6, 9, 14, 19, \infty\}$$

where $\infty$ means no noise was applied. This results in eleven variants of the original non-noised data, for each, each of the models are trained individually on the variant and evaluated. Sensitivity is always 1 during the noising operation, as each member of the dataset is accessed by the machine learning model. We then apply a standard scaler on the data for the linear regression and logistic regression models only. We evaluate and record accuracy, recall, precision, and $F_1$ on the test set for each $\epsilon$ value. We only evaluate accuracy and $F_1$ in our discussion.

## 4 Results & Discussion

### 4.1 Cervical Cancer (Risk Factors)

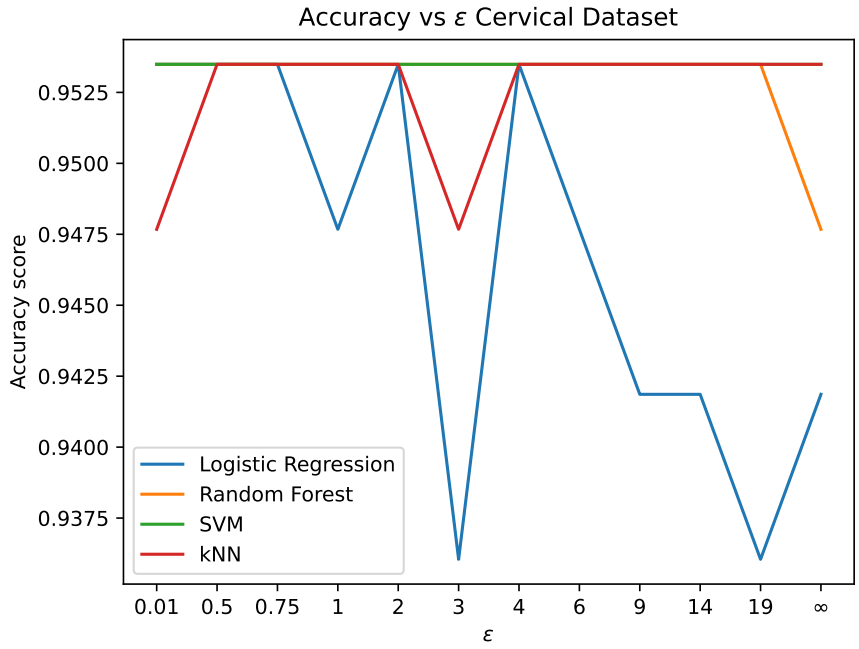


Figure 1: Accuracy of four models on Cervical Cancer dataset test set versus $\epsilon$ value used for Laplacian mechanism.

In figure 1, the accuracy on the test set remains relatively constant for the kNN, SVM, and Random Forest models, but fluctuates wildly with Logistic Regression. We hypothesize that the major class imbalance in the dataset, where $\tilde{55}$ individuals have a positive biopsy result and the remaining $\tilde{8}00$ negative, results in any classifier predicting all zeros to have a high accuracy. This hypothesis is supported by the correlation matrix 19, which shows low correlation between the predictor and target variables. This is also supported by the poor performance of models trained on the the non-noised ($\infty$) data.

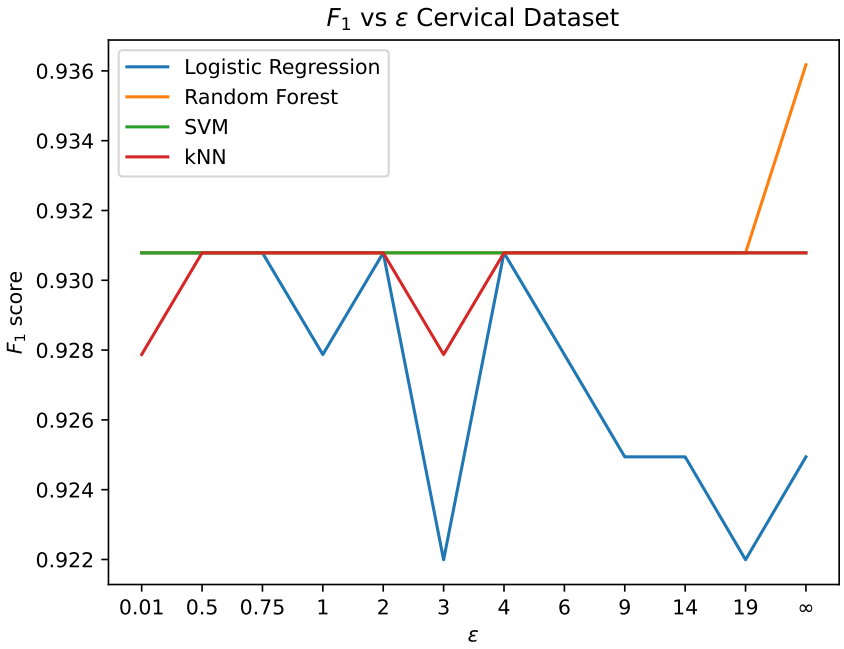


Figure 2: $F_1$ score of four models on Cervical Cancer dataset test set versus $\epsilon$ value used for Laplacian mechanism.

As in figure 1, figure 2 shows equally *"high"* performance across all $\epsilon$ levels. Counter intuitively, the logistic regression performance continues to decrease after $\epsilon = 4$. Unlike the accuracy, the $F_1$ increases for the random forest model on the non-noised data. The performance is not actually high; if the test set classes were balanced performance would likely drop significantly due to the classifiers predicting zeros for positive examples.

## 4.2 Adult Income

In figure 3, the accuracy on the test set quickly increases as $\epsilon$ increases for all models. We note that accuracy actually drops for the Random Forest model on the non-noised data. We hypothesize that even a small amount of noise present at $\epsilon = 19$ results in a lower generalization error compared to the model trained on the non-noised data. We also note that the best performing model has an accuracy of $\tilde{0}.86$ at $\epsilon = 19$ and $\tilde{0}.82$ at $\epsilon = \frac{1}{2}$. This suggests that performance gain between a high-privacy $\epsilon$ value and a privacy-disregarding value is relatively minimal, implying a low utility loss.

Figure 4 shows that $F_1$ score on the Adult Income test set closely mirrors accuracy. It is important to consider the $F_1$ score as it puts more weight on false positives/negatives. This makes it a more useful metric for class-imbalanced datasets, and a relatively high value of around 0.6 for kNN and Random Forest suggests acceptable utility.

## 4.3 Contraceptive Method Choice

Based on figure 5, the models performed poorly across the board on the Contraceptive dataset test split. It is important to note that the Contraceptive dataset is the only multi-class task we investigate. This means that an accuracy of 50% is not equal to a coin flip, unlike the binary classification tasks, since the coin would have to have three faces (geometrically impossible for flat faces). In other words, 50% accuracy is better than chance. Regardless the accuracy is low, but utility does not decrease significantly as $\epsilon$ is decreased, for the best performing (Random Forest) model. Figure 6 mirrors figure 5 closely, likely due to the choice of weighted average for the $F_1$ score, which calculates $F_1$ for each label and finds their average weighted by number of true instances for each label.

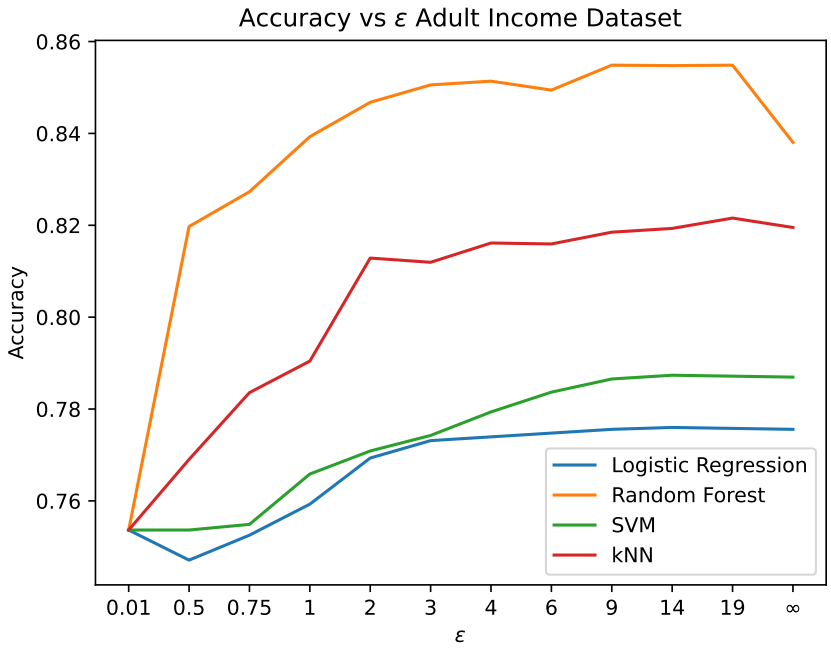


Figure 3: Accuracy of four models on Adult Income dataset test set versus $\epsilon$ value used for Laplacian mechanism.

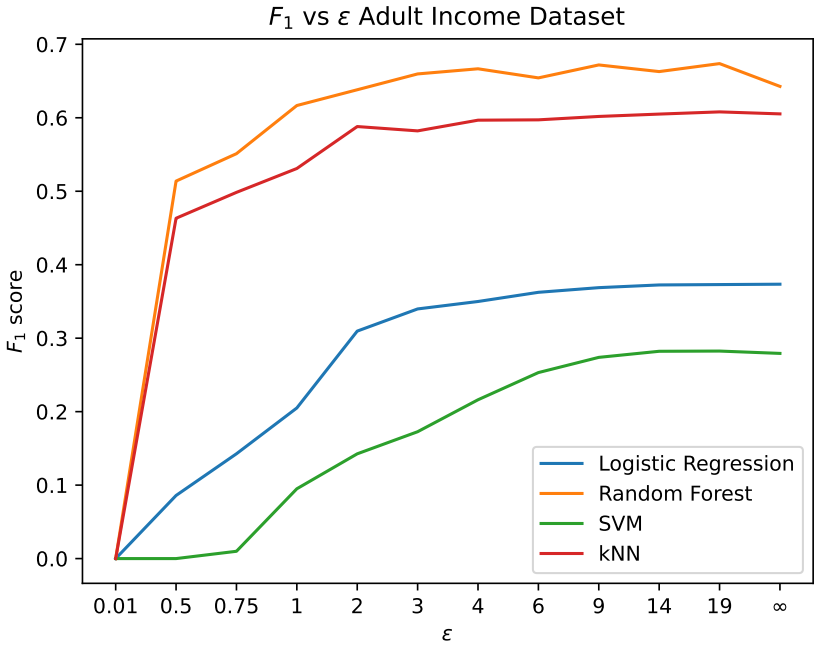


Figure 4: $F_1$ of four models on Adult Income dataset test set versus $\epsilon$ value used for Laplacian mechanism.

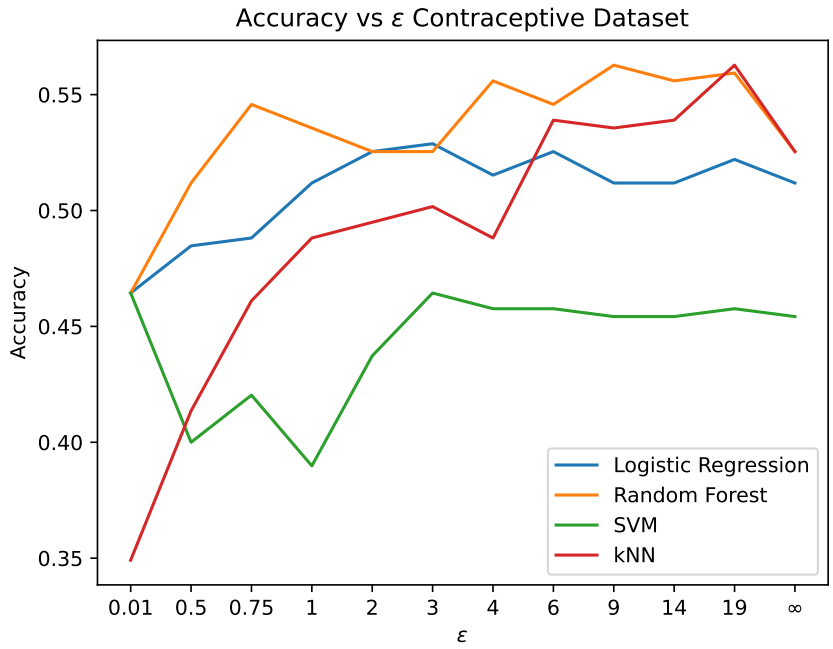


Figure 5: Accuracy of four models on Contraceptive dataset test set versus $\epsilon$ value used for Laplacian mechanism.

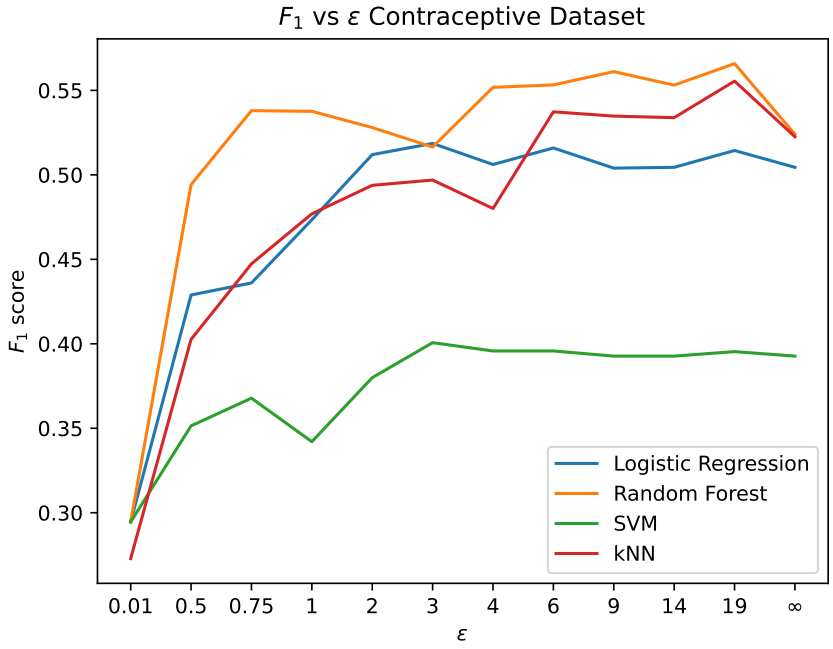


Figure 6: $F_1$ of four models on Contraceptive dataset test set versus $\epsilon$ value used for Laplacian mechanism.

### 4.4 California Housing Prices

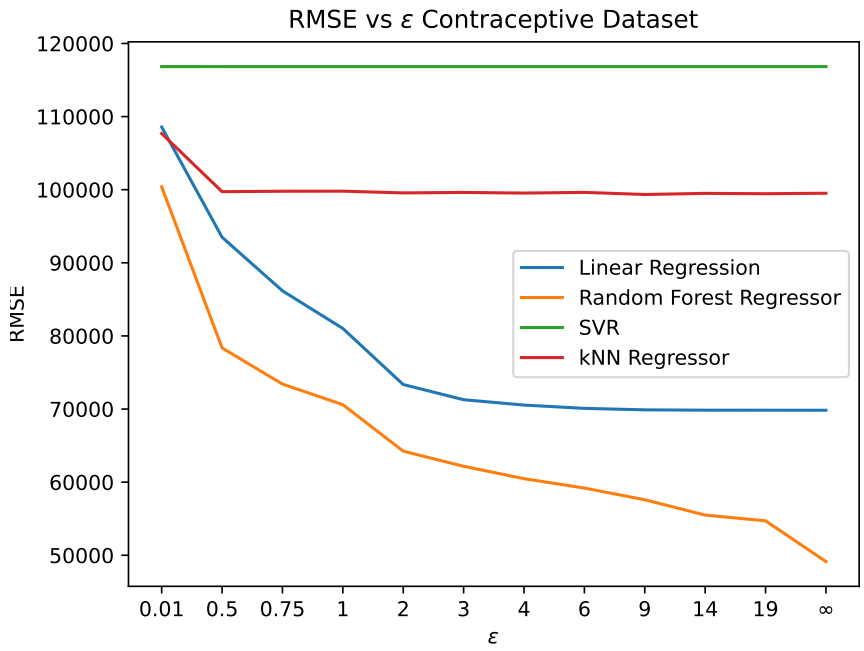


Figure 7: Root Mean Squared Error (RMSE) of four models on California Housing dataset test set versus $\epsilon$ value used for Laplacian mechanism.

As the only dataset we investigate with a continuous target variable, figure 7 differs from the other figures as the metric used for evaluation is Root Mean Square Error, where lower is better and then is no upper bound on the metric, unlike accuracy and $F_1$ which are bounded by 1, with 1 being the best possible value. Therefore, in this figure we see that the error on the test set decreases as $\epsilon$ increases, but only for the Random Forest Regressor and the Linear Regression model. We did not develop a hypothesis for why the SVR and kNN Regressors performed so poorly that no difference in utility was observed. We note that standard scaler was used only for Linear Regression, but not for remaining models. This implies the standard scaler cannot be the reason for high Random Forest performance.

### 4.5 Mammographic Mass

The Mammographic dataset yielded high utility, as per figure 8, which shows that the Random Forest classifier peaked at nearly 0.85 accuracy on the test set. Interestingly, performance dropped past $\epsilon = 9$, likely a similar effect seen in other datasets, where generalization error (i.e. between train and test) is actually lower for certain noise levels, likely due to reduced overfitting. The Random Forest model actually performs worse on the non-noised data, compared to even an $\epsilon$ of 2. The remaining models follow a similar trend, but without a noticeable dip in performance with low noise. We note that figure 9 does not differ much from figure 8, but considering the RF, kNN, and Logistic Regression models with an $F_1$ score above 0.7 for all $\epsilon \geq \frac{1}{2}$ suggests high utility despite the strong privacy guarantees of $\epsilon = \frac{1}{2}$.

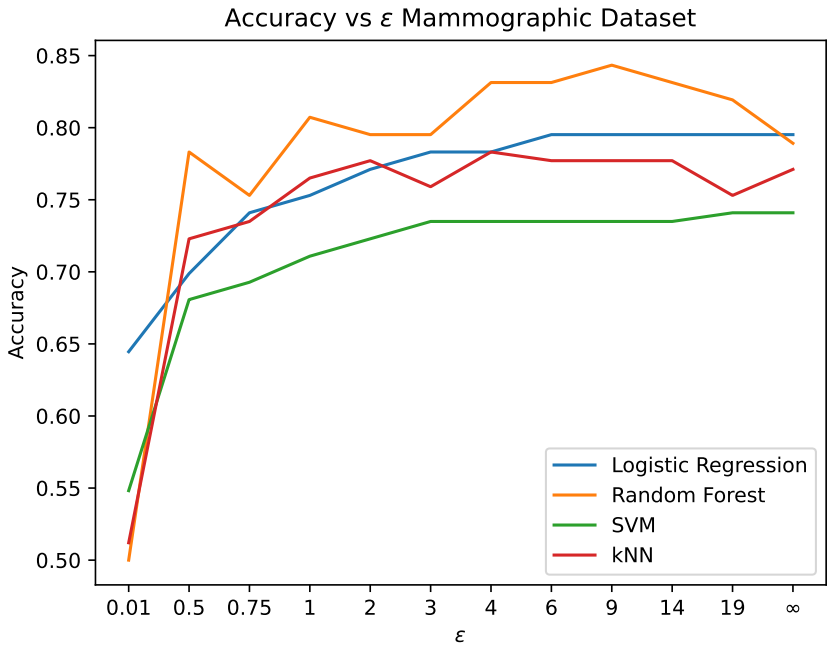


Figure 8: Accuracy of four models on Mammographic dataset test set versus $\epsilon$ value used for Laplacian mechanism.

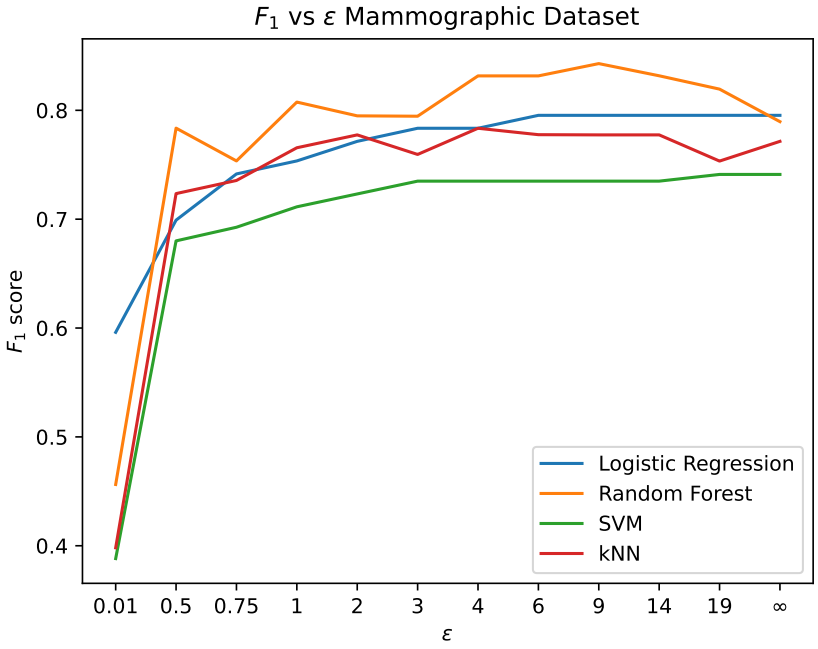


Figure 9: $F_1$ score of four models on Mammographic dataset test set versus $\epsilon$ value used for Laplacian mechanism.

# 5 Threats to Validity & Conclusion

## 5.1 Threats to Validity

While we performed our investigation and analysis with care, we highlight some potentials threats to our results.

**No hyperparameter tuning** For smaller and quick to train models we investigated (Linear models, Random Forests, Support Vector Machines, and k-Nearest Neighbors), it is common to perform a hyperparameter tuning operation that evaluates many permutations of hyperparameter settings for each model. We did not perform this, and such cannot be sure if our models were optimal. It is possible that certain results would lead us to a different conclusion if the model was more tuned. This can be especially the case for random forests, which can benefit greatly from hyperparameter tuning.

**Choice of split** We used a simple, random train/test split and evaluate on just the test split. A more robust approach would be to apply k-fold cross validation and compare or average results between folds.

**Single train/test run** It is common when performing empirical evaluations, to repeat the observed process a fixed number of times for all variants, taking the mean, and examining the variance between samples. In our scenario, this would mean, for each dataset, running each model training 5 times (for example) instead of just once and recording the performance for each. This would give more robust results, but could draw different conclusions from our observations.

**No formal verification of implementation** While PyPryvacy follows the techniques presented in lecture slides to the best of the author's knowledge and understanding, there has been no external/peer review of the implementation. As such, we cannot guarantee a lack of error that may skew results differently were it to be fixed. Despite this, we remain confident in our implementation but encourage outside feedback.

**Suboptimal choice of metrics** While accuracy and $F_1$ are widely used metrics, other plots and metrics are commonly used that may be more indicative of utility. The primary alternative we considered were AUC ROC plots, which would also allow us to highlight the Pareto frontier.

## 5.2 Conclusion

We present PyPryvacy, a Python implementation of $\epsilon$-Differential Privacy algorithms. We evaluated the performance of Linear, Random Forest, Support Vector Machine, and k-Nearest Neighbor models on five datasets, and found that on each dataset where a model performed well on the non-noised data, the best performing models often performed within less than ten percentage point difference for privacy-preserving $\epsilon \leq 4$ values. This shows that if the dataset had utility in the first place, the utility is not greatly impacted by applying differential privacy techniques. This is a promising result towards wide scale adoption of differential privacy in data science & machine learning pipelines.

# References


[1] "Pepy - pandas download stats."

[2] T. Strohmer, "Differential privacy - part 1 (lecture 7)."

[3] T. Strohmer, "Differential privacy - part 3 (lecture 9)."

[4] C. Dwork, "Differential privacy," in *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pp. 1–12, Springer, 2006.

[5] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019. http://www.fairmlbook.org.

[6] C. J. . F. J. Fernandes, Kelwin, "Cervical cancer (Risk Factors)." UCI Machine Learning Repository, 2017. DOI: 10.24432/C5Z310.

[7] "Adult." UCI Machine Learning Repository, 1996. DOI: 10.24432/C5XW20.

[8] T.-S. Lim, "Contraceptive Method Choice." UCI Machine Learning Repository, 1997. DOI: 10.24432/C59W2D.

[9] C. Nugent, "California housing prices," Nov 2017.

[10] R. K. Pace and R. Barry, "Sparse spatial autoregressions," *Statistics & Probability Letters*, vol. 33, no. 3, pp. 291–297, 1997.

[11] M. Elter, "Mammographic Mass." UCI Machine Learning Repository, 2007. DOI: 10.24432/C53K6Z.

# 6 Appendix

In the appendix we include figures showing correlation matrices of the datasets, and the results of the Gaussian mechanism with $\delta = 0.05$. We did not pursue the further analysis of the Gaussian mechanism.

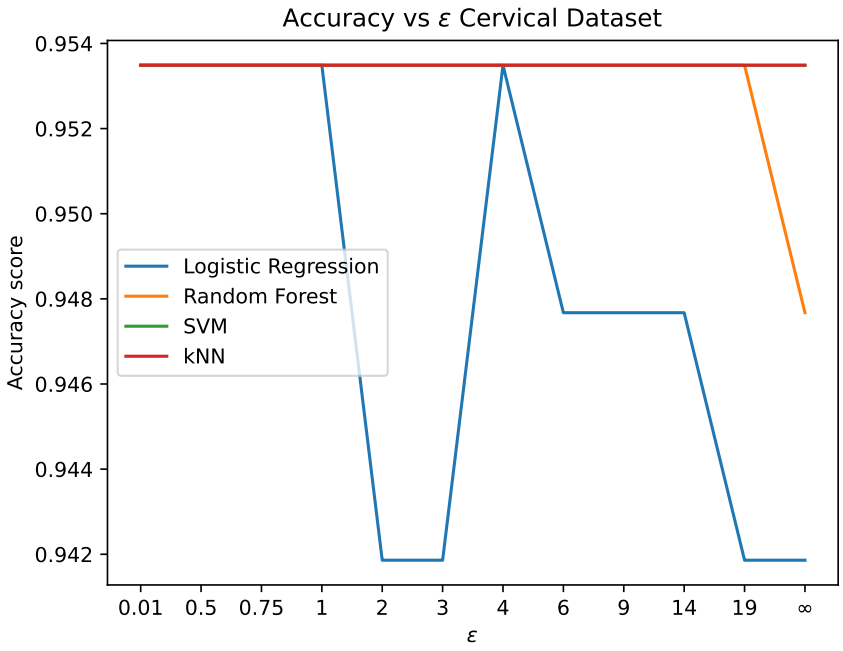


Figure 10: Accuracy of four models on Cervical Cancer dataset test set versus $\epsilon$ value used for Gaussian mechanism.

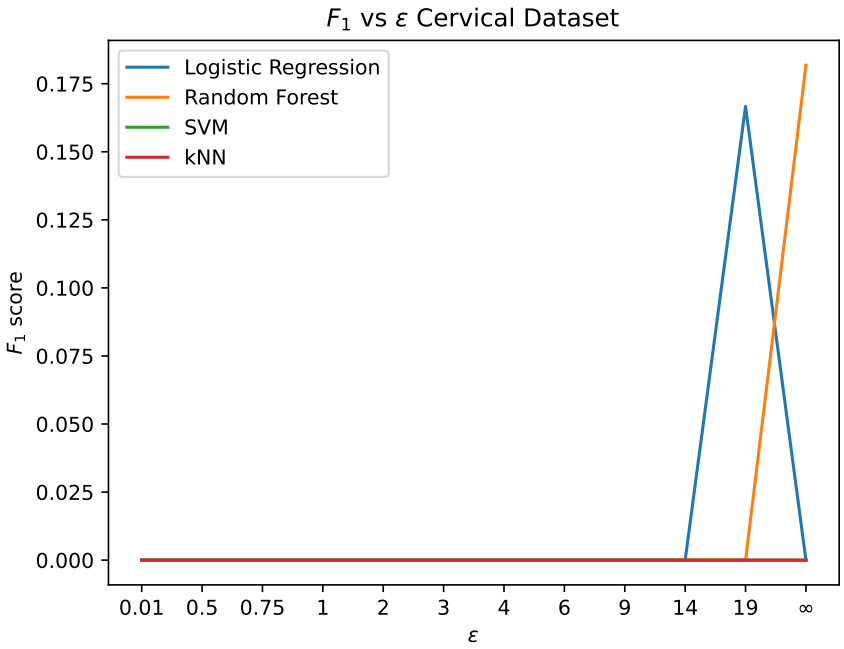


Figure 11: $F_1$ score of four models on Cervical Cancer dataset test set versus $\epsilon$ value used for Gaussian mechanism.

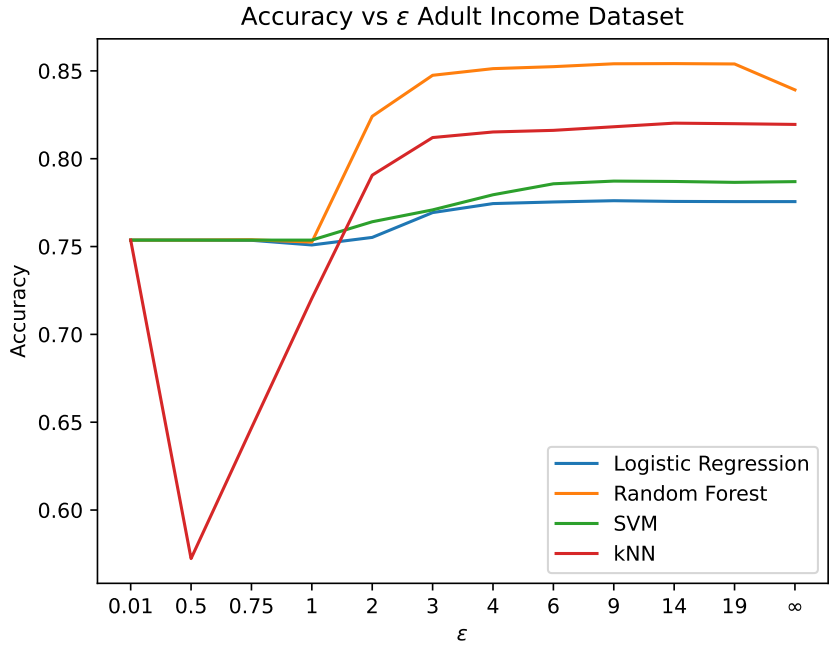


Figure 12: Accuracy of four models on Adult Income dataset test set versus $\epsilon$ value used for Gaussian mechanism.

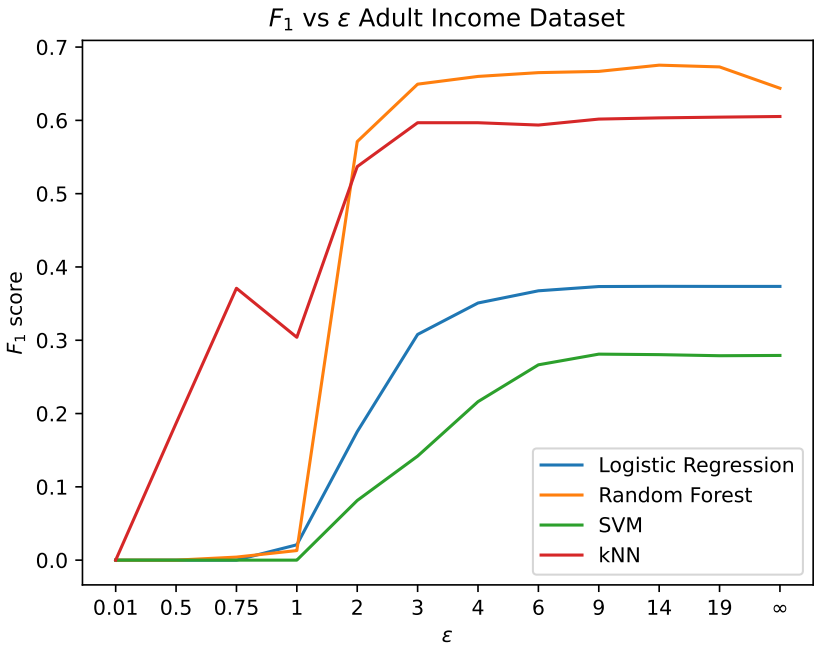


Figure 13: $F_1$ of four models on Adult Income dataset test set versus $\epsilon$ value used for Gaussian mechanism.

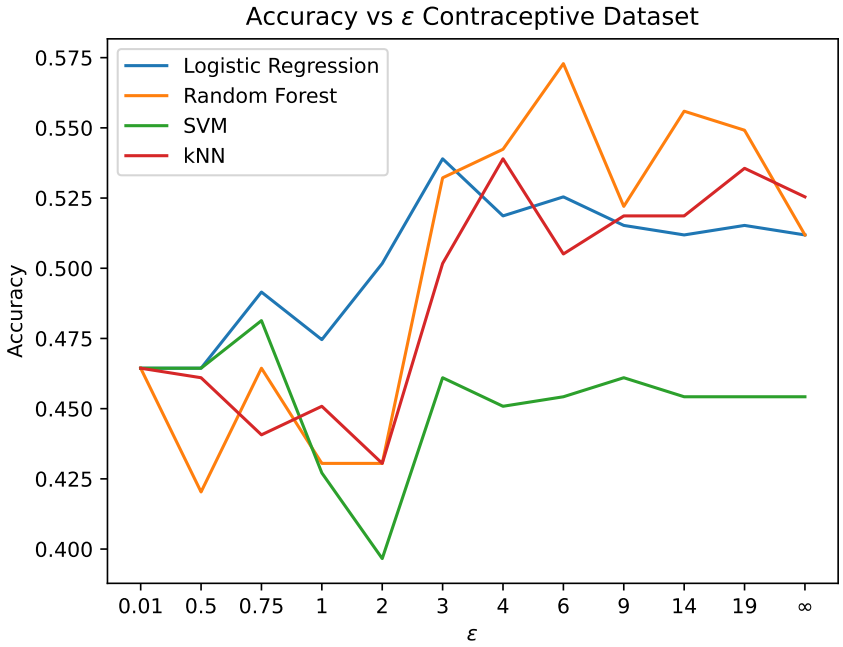


Figure 14: Accuracy of four models on Contraceptive dataset test set versus $\epsilon$ value used for Gaussian mechanism.

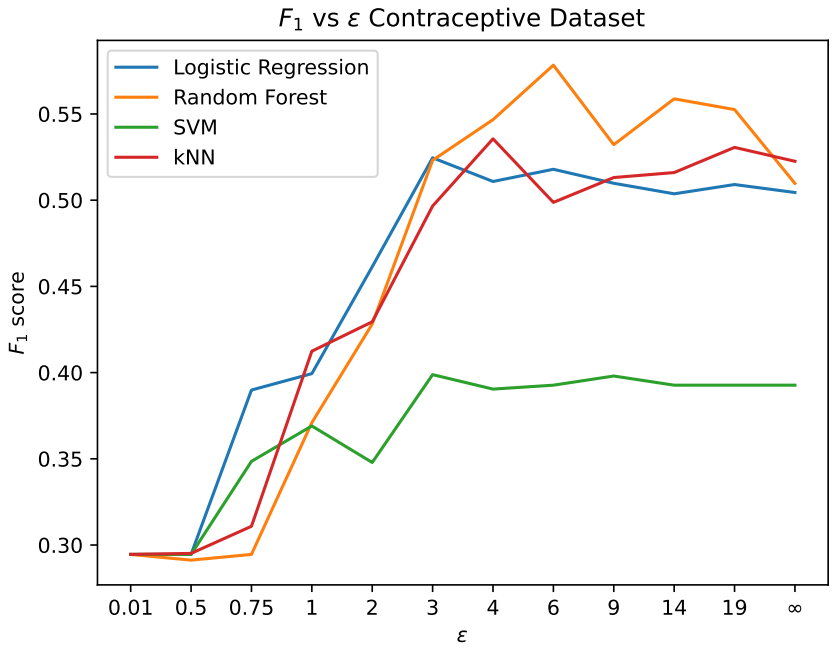


Figure 15: $F_1$ of four models on Contraceptive dataset test set versus $\epsilon$ value used for Gaussian mechanism.

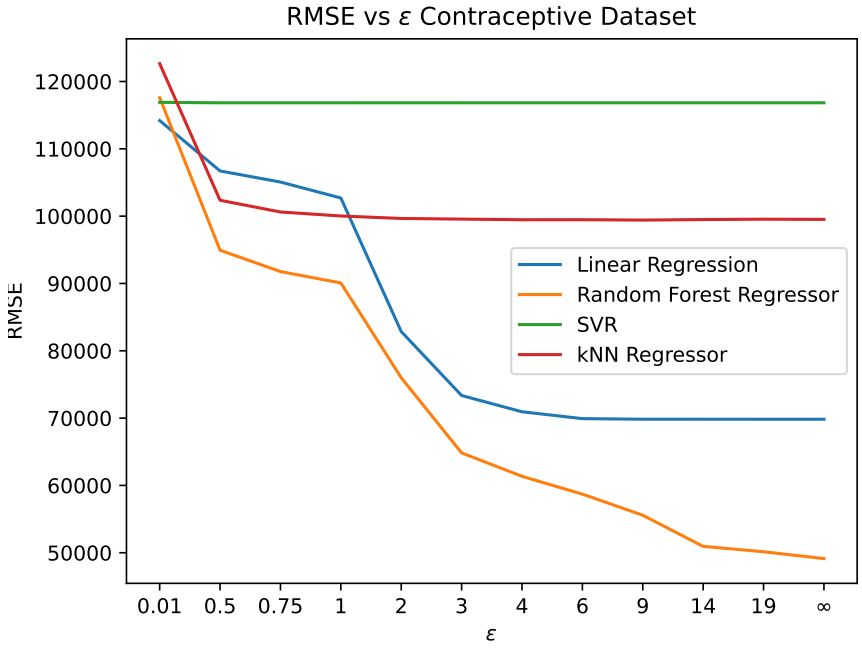


Figure 16: Root Mean Squared Error (RMSE) of four models on California Housing dataset test set versus $\epsilon$ value used for Gaussian mechanism.

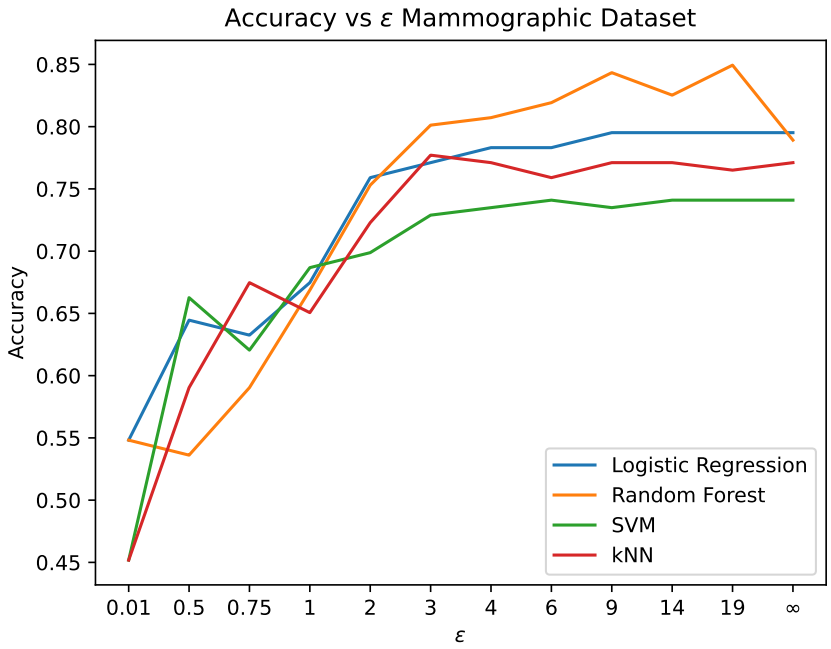


Figure 17: Accuracy of four models on Mammographic dataset test set versus $\epsilon$ value used for Gaussian mechanism.

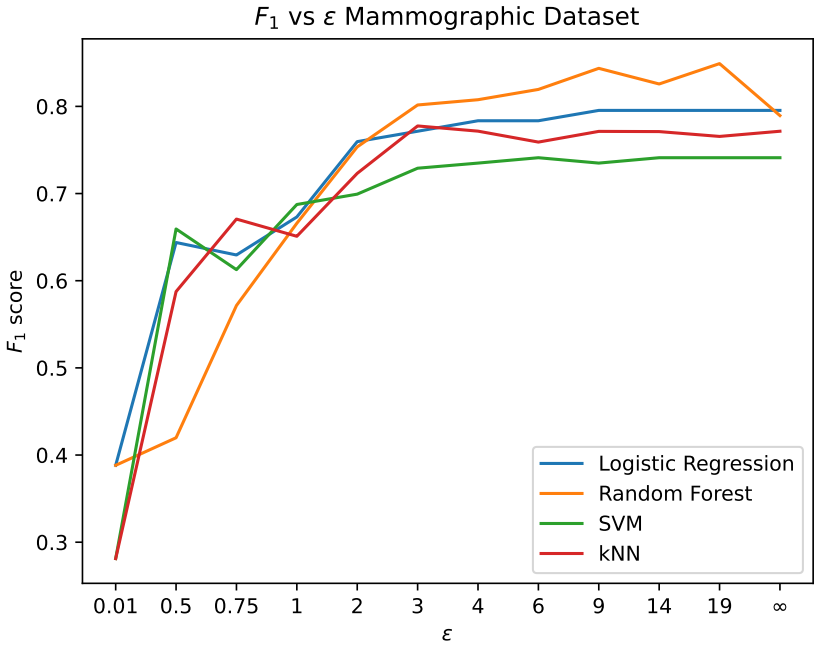


Figure 18: $F_1$ score of four models on Mammographic dataset test set versus $\epsilon$ value used for Gaussian mechanism.

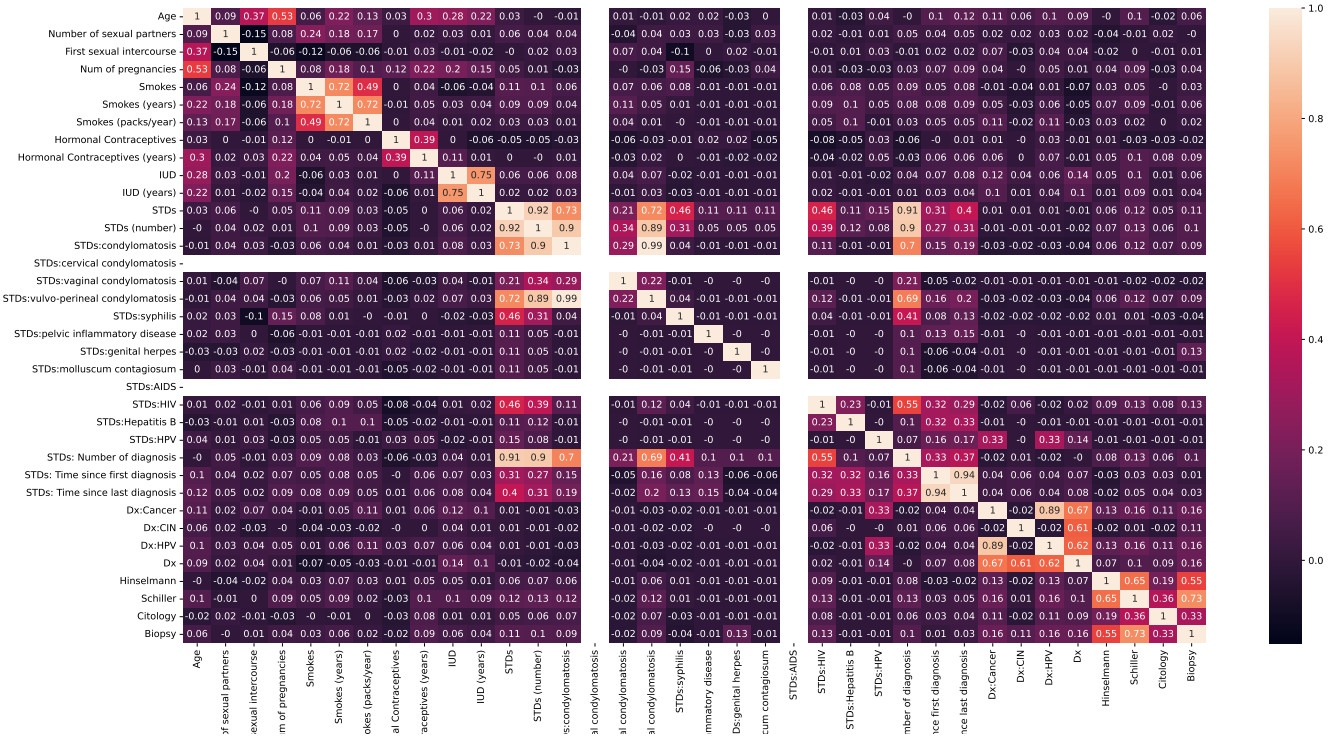


Figure 19: Correlation matrix for Cervical dataset.

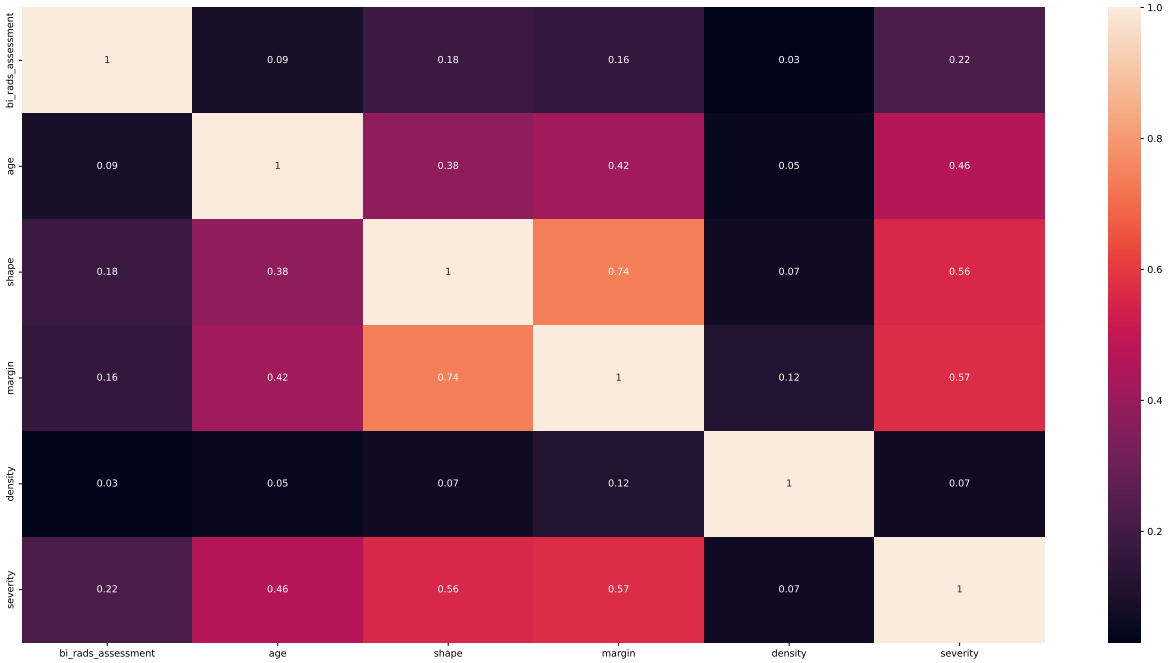


Figure 20: Correlation matrix for Mammographic dataset.

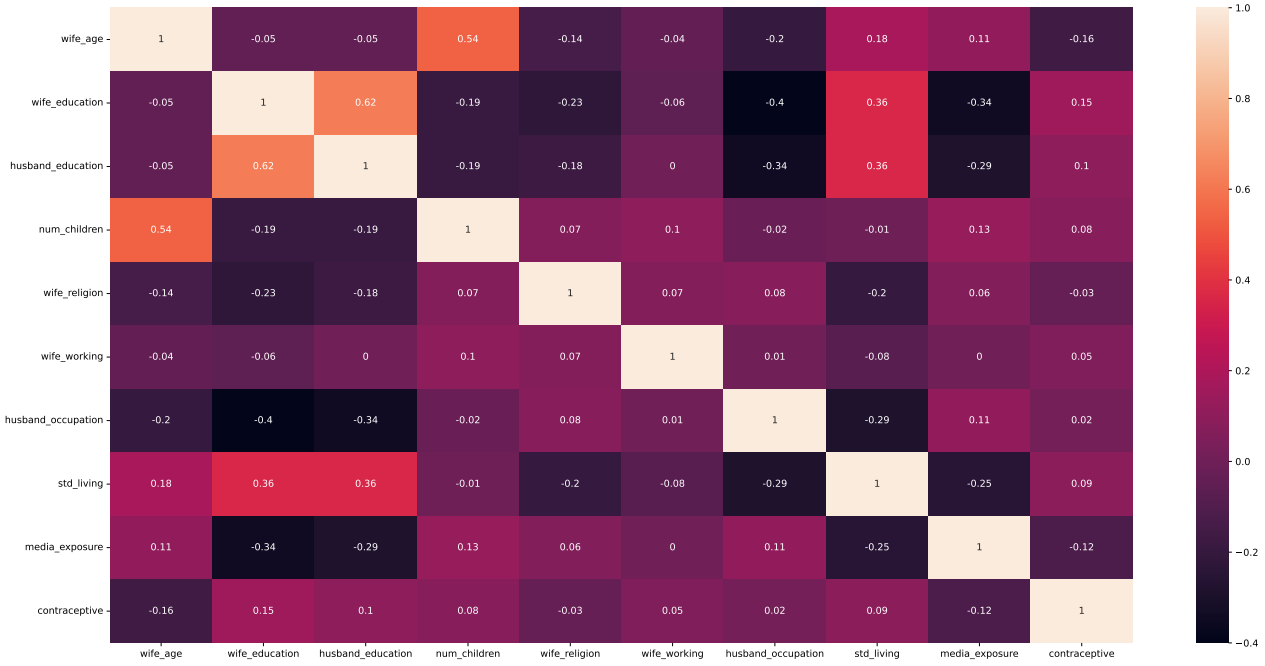


Figure 21: Correlation matrix for Contraceptive dataset.

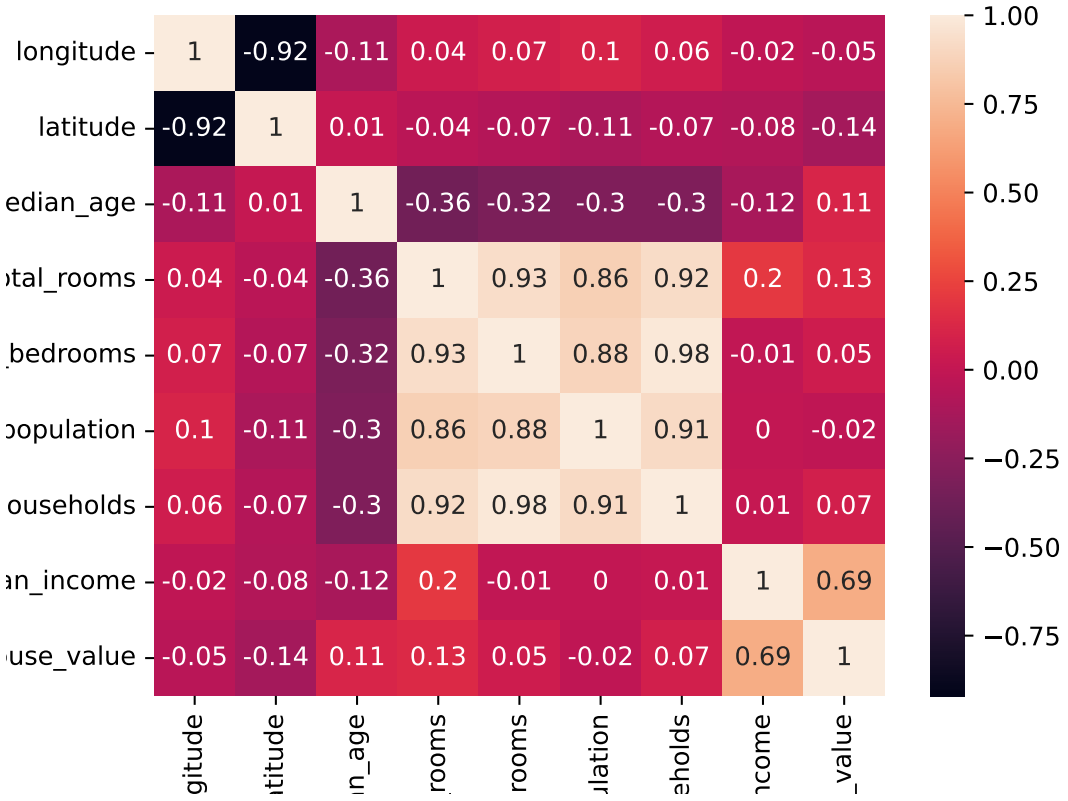


Figure 22: Correlation matrix for California Housing dataset.

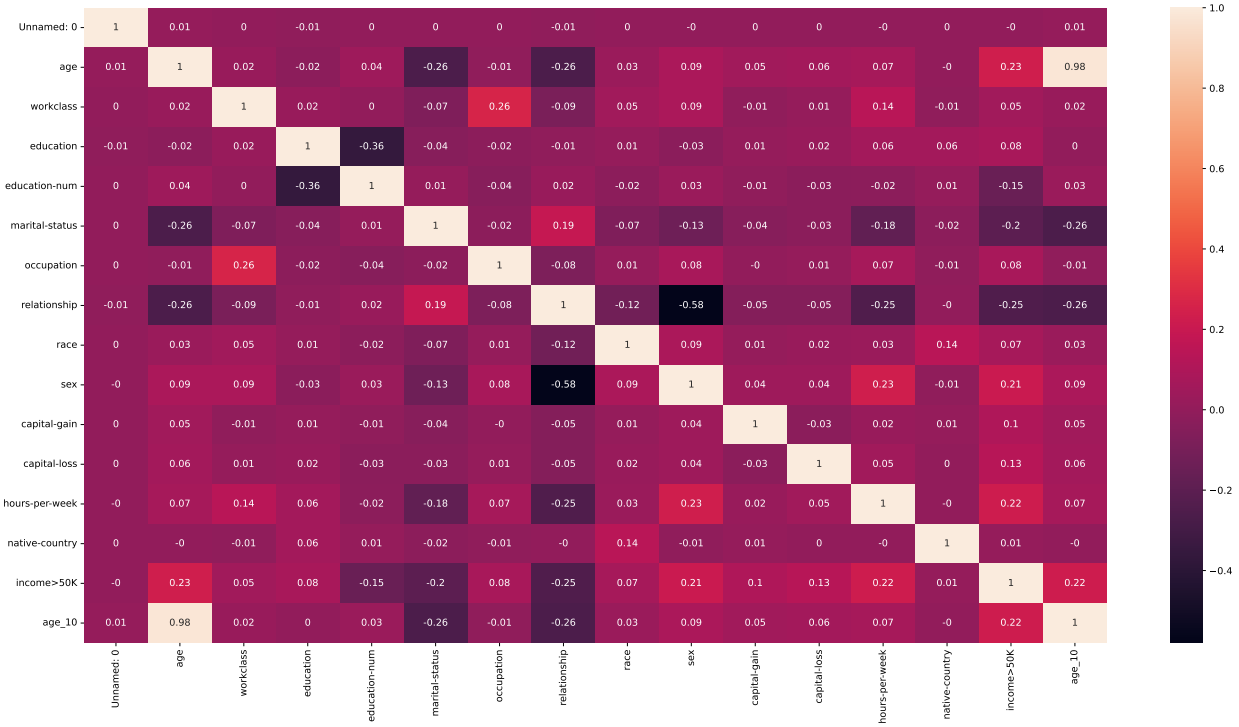


Figure 23: Correlation matrix for Adult Income dataset.