Auditing Black-Box Prediction Models for Data Minimization Compliance

Bashir Rastegarpanah   Krishna P. Gummadi   Mark Crovella

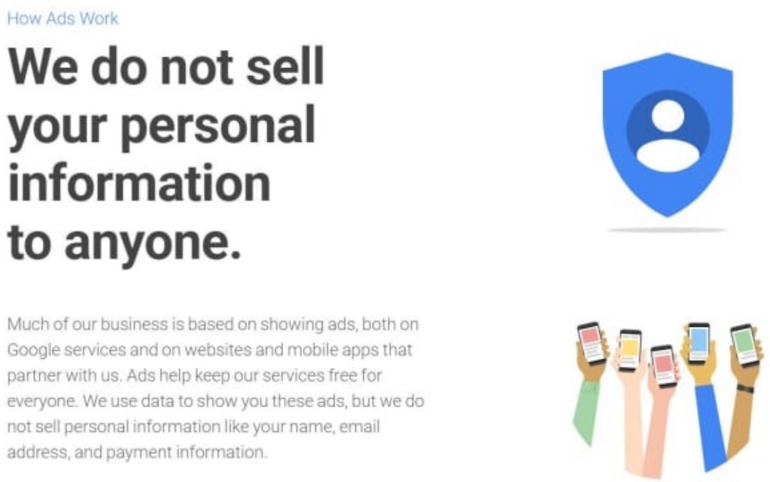
35th Conference on Neural Information Processing Systems (NeurIPS 2021)

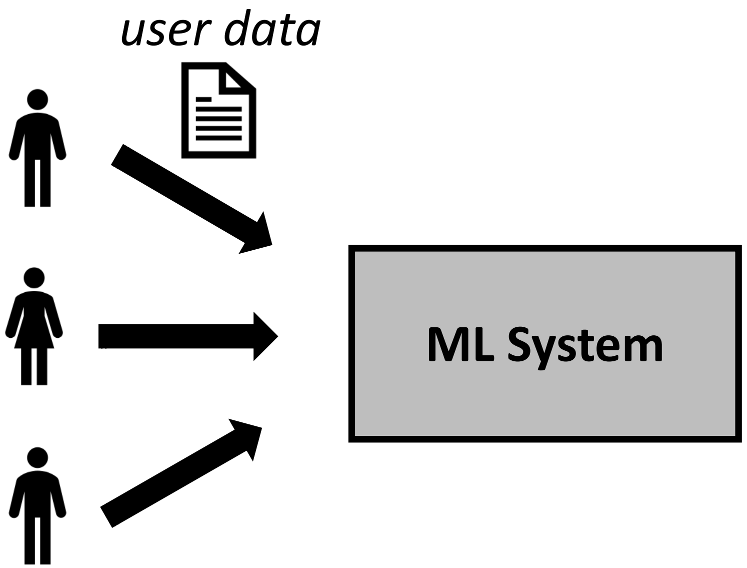
BOSTON UNIVERSITY

MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

NSF

Privacy in Data-driven Models

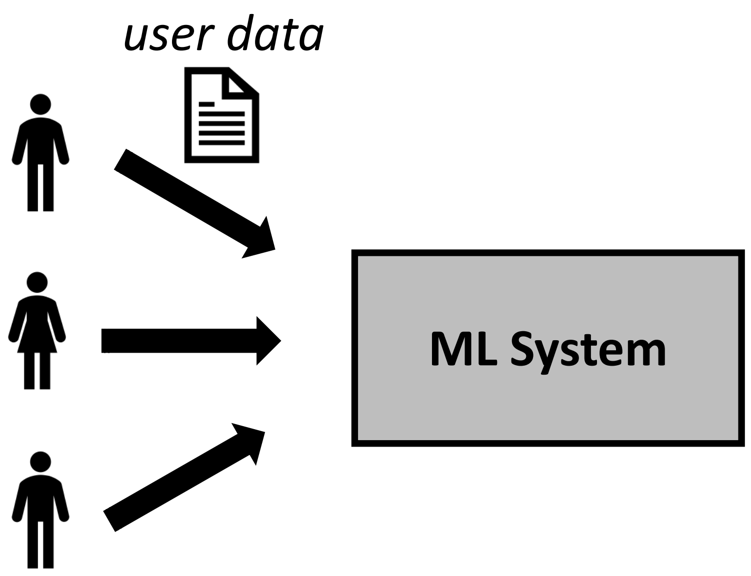




Privacy in Data-driven Models

Cryptography approaches seek complete privacy (e.g., secure multi party computation)

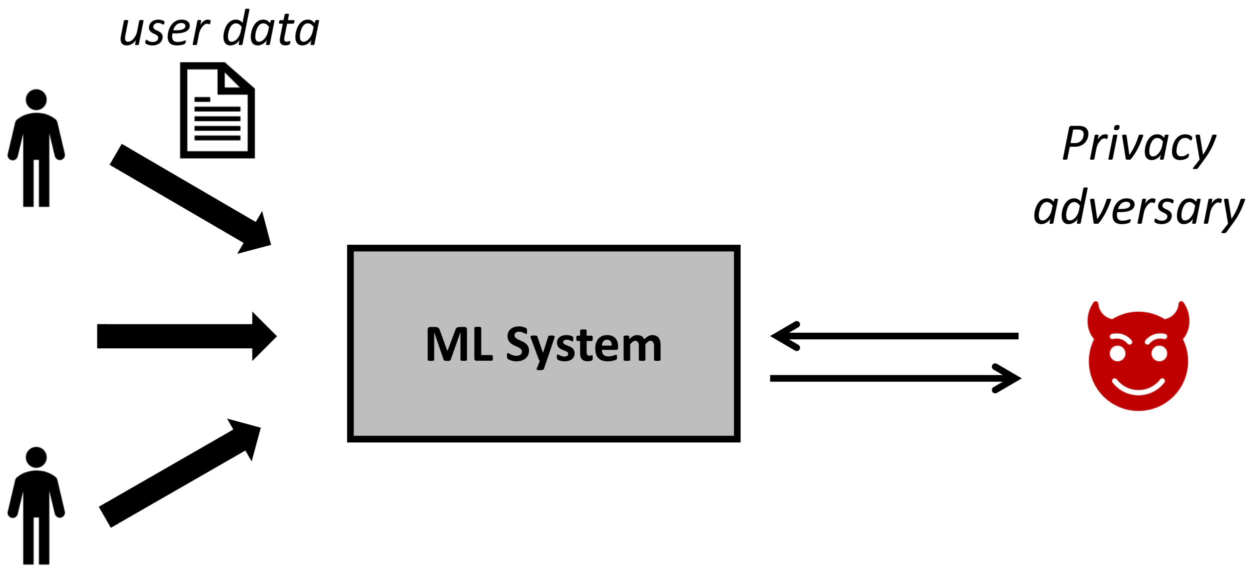
- Computational efficiency challenges
- Some user data may need to be recorded due to regulatory auditing purposes.



Privacy in Data-driven Models

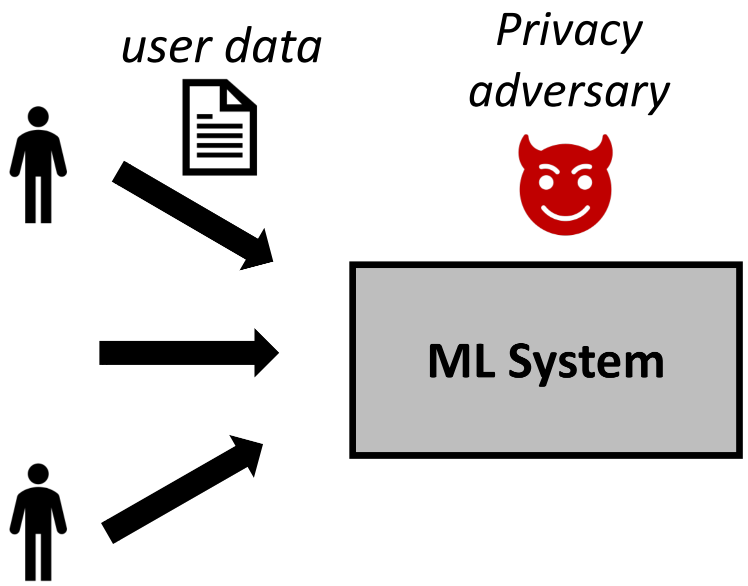
Privacy notions that assume an adversary different from the data processing system

- **Differential privacy**
- **K-anonymity**



Privacy in Data-driven Models

What if the prediction system itself is a privacy adversary?

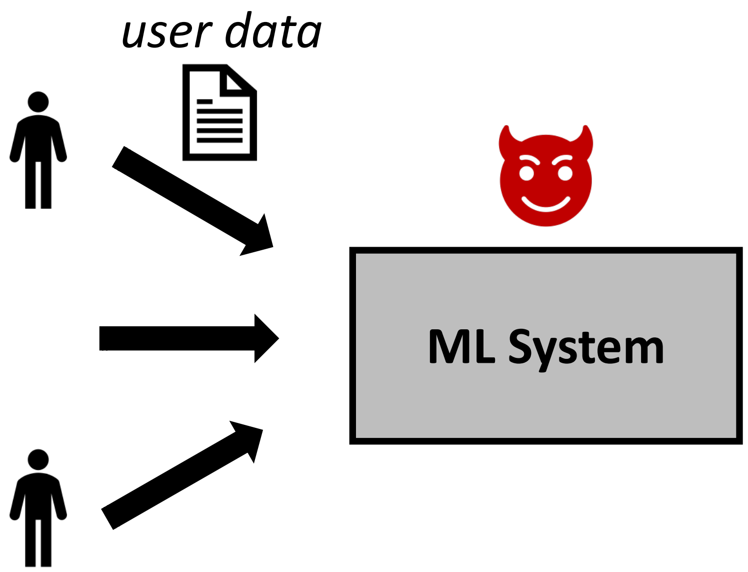


Privacy in Data-driven Models

What if the prediction system itself is a privacy adversary?

An alternative privacy notion

Restrict prediction systems to use the minimum necessary data.



Data Minimization as a privacy notion

Data Minimization (GDPR, article 5.1.c)

"Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed."

Data Minimization as a privacy notion

Data Minimization (GDPR, article 5.1.c)

"Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed."

How to operationalize this principle for a particular prediction system?

Previous Proposals

[Biega et al. SIGIR 2020]
[Rastegarpanah et al. UMAP 2020]

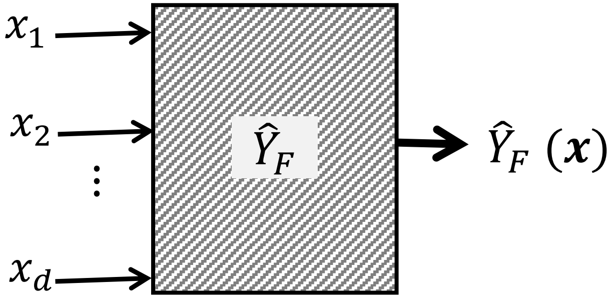
"Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed."

Tie the purpose of data processing to some performance metric (e.g., accuracy)

Assuming full knowledge of the prediction algorithm and the training data, study whether input data can be reduced while achieving similar performance.

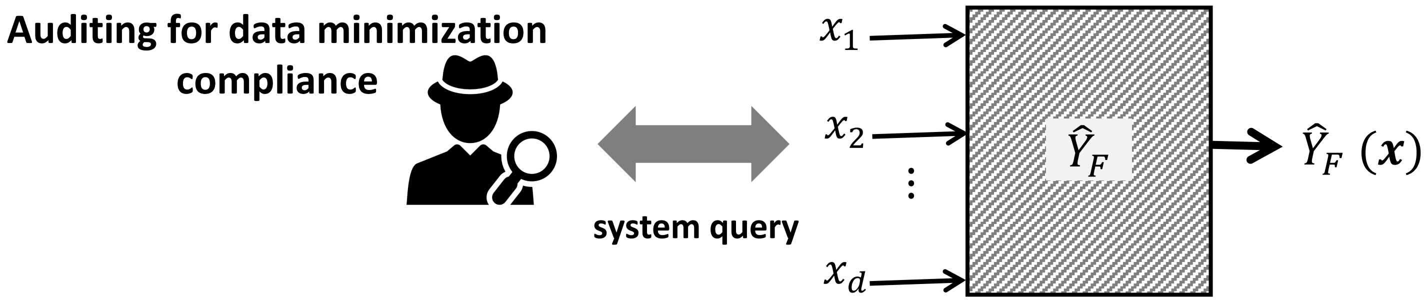
Auditing Black-Box Prediction Models

A black-box prediction model with a fixed set of input features at deployment time.



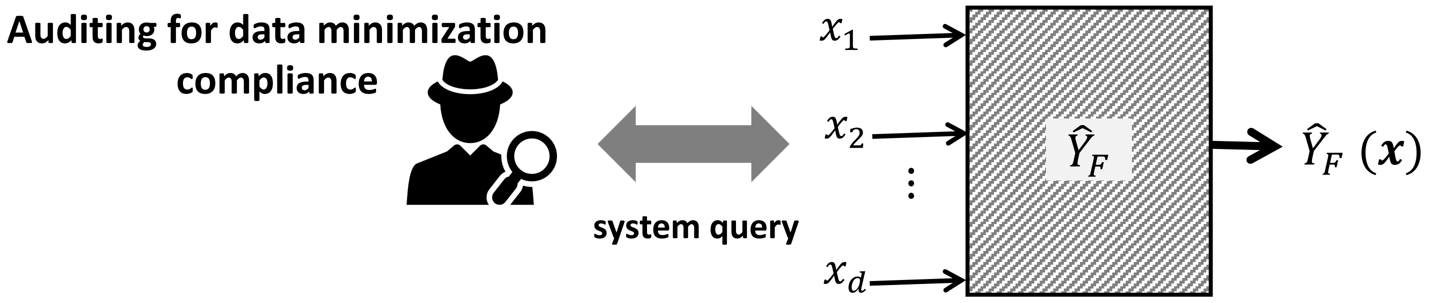
Auditing Black-Box Prediction Models

A black-box prediction model with a fixed set of input features at deployment time.



Auditing Black-Box Prediction Models

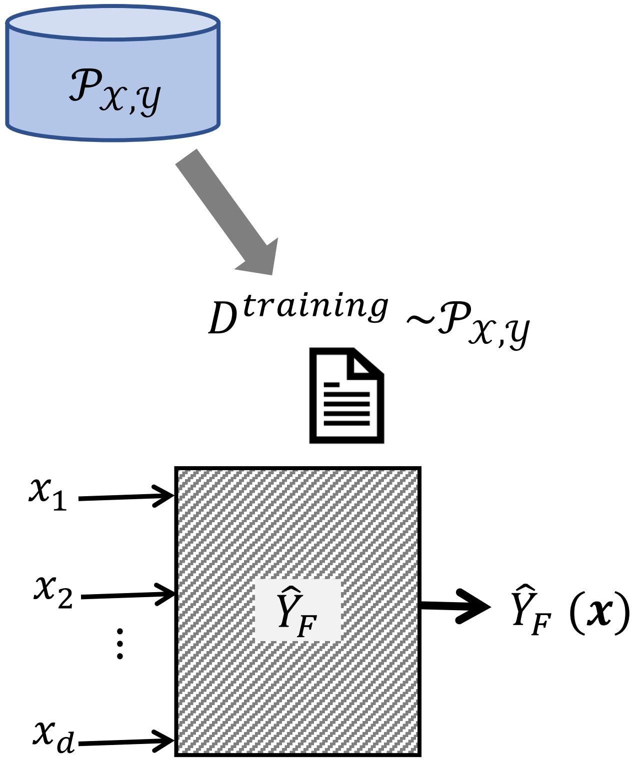
A black-box prediction model with a fixed set of input features at deployment time.



We propose a criterion that can be used for operationalizing data minimization in this setting.

Auditing Black-Box Prediction Models

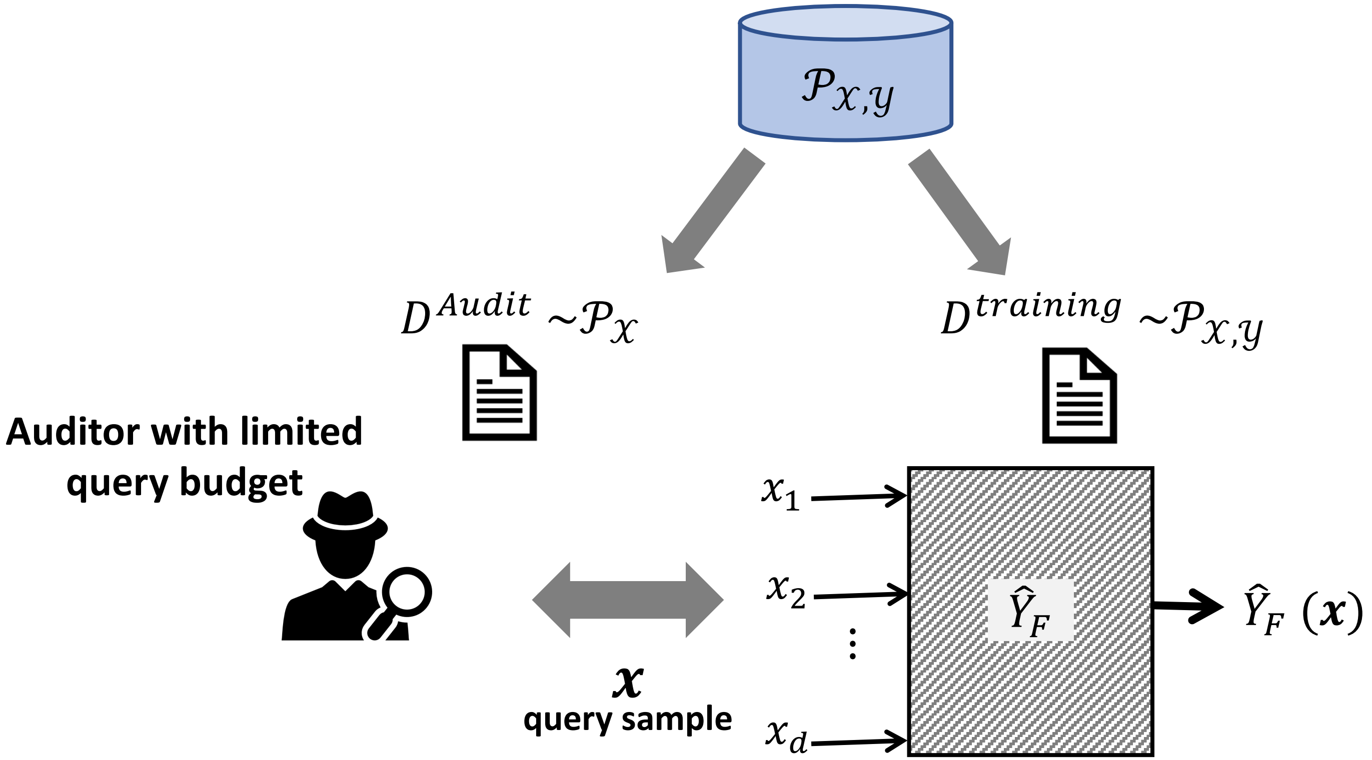
$\hat{Y}_F$: prediction model with the set of input features F.



Auditing Black-Box Prediction Models

$\hat{Y}_F$: prediction model with the set of input features F.

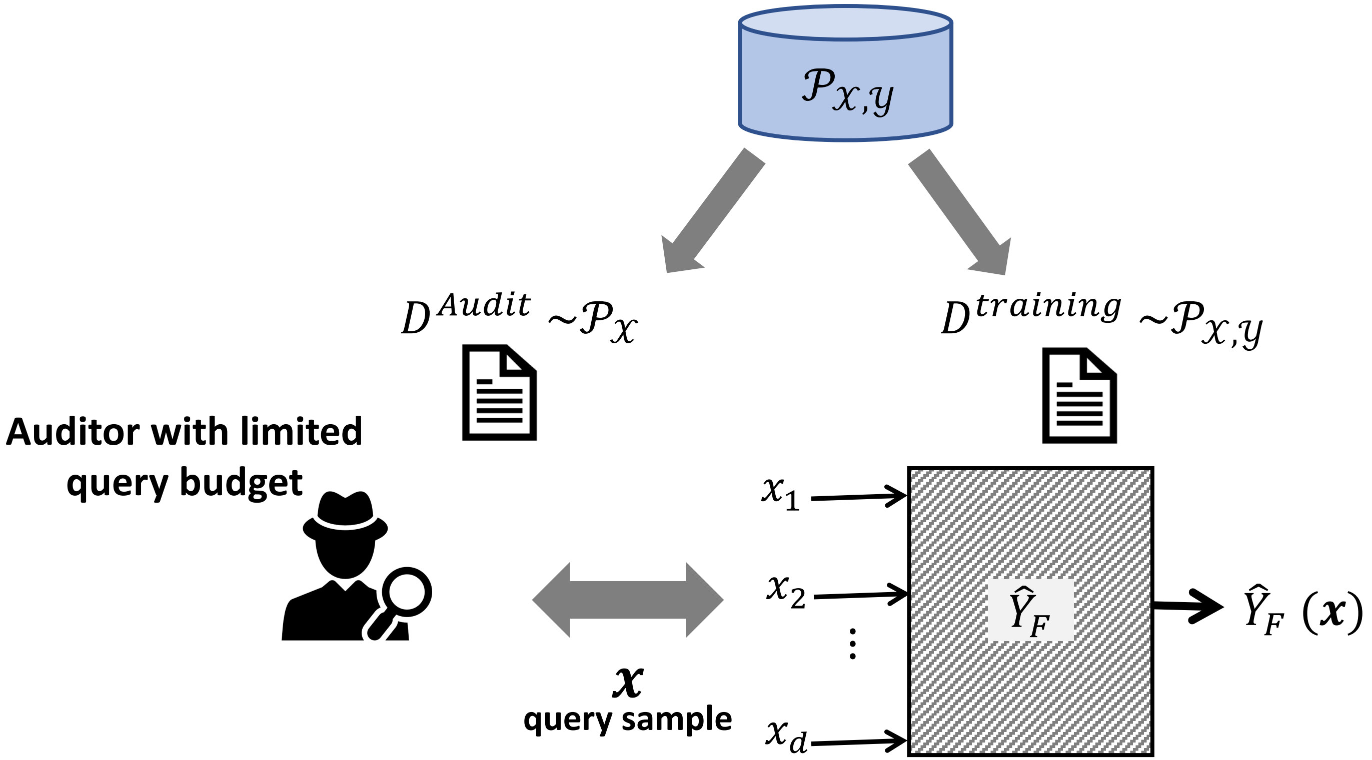
Auditor can query the system using prediction instances that that specify all feature values.



Auditing Black-Box Prediction Models

$\hat{Y}_F$: prediction model with the set of input features F.

Auditor can query the system using prediction instances that that specify all feature values.

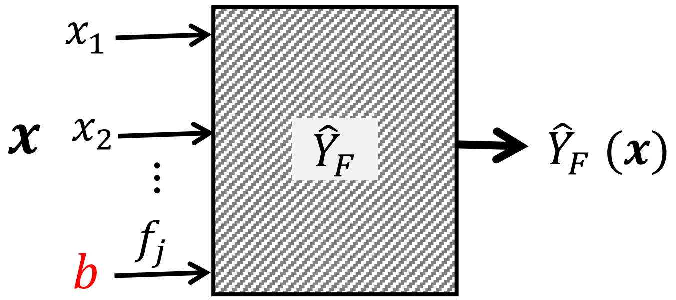


To what extent Data Minimization is satisfied by $\hat{Y}_F$?

Assessing the Need for Individual Features

Simple imputations as a tool for limiting data inputs at test time.

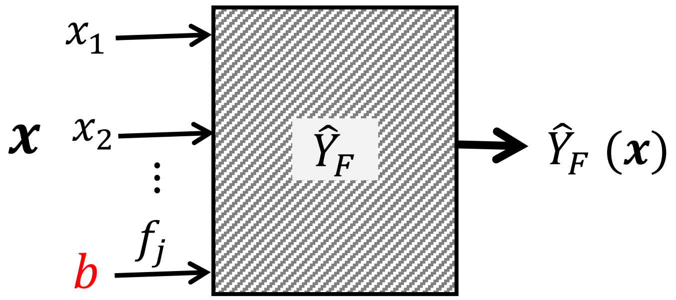
Impute f_j with constant b in all prediction instances.



Assessing the Need for Individual Features

Simple imputations as a tool for limiting data inputs at test time.

Impute f_j with constant b in all prediction instances.

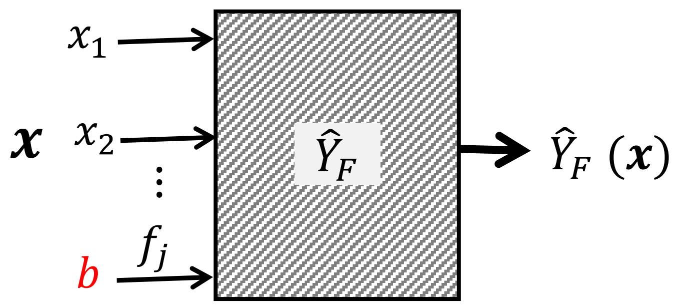


If applying this imputation across all prediction instances has no or small effect on the model outputs, the information about the actual value of the corresponding feature is not needed by the model.

Model Instability under Simple Imputations

$\tau_{f_j,b}(\boldsymbol{x})$: imputation function that replaces the value of f_j with b.

$$I_{\hat{Y}_F}(\mathbf{x}, f_j, b) = \begin{cases} 1 & \text{if } \hat{Y}_F(\mathbf{x}) \neq \hat{Y}_F(\tau_{f_j,b}(\mathbf{x})) \\ 0 & \text{otherwise} \end{cases}$$

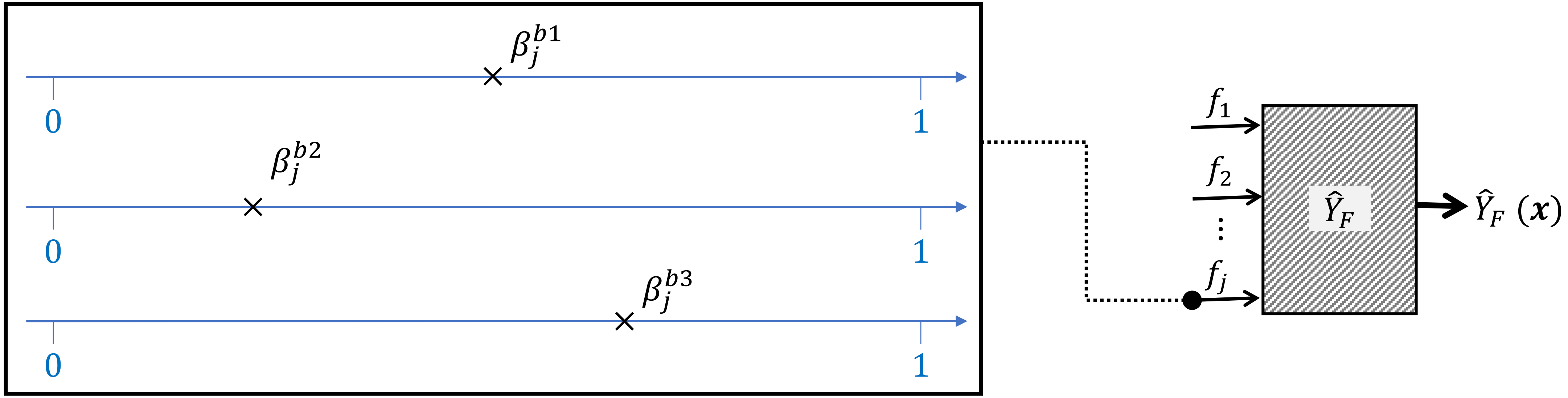


X: random variable that takes values $\boldsymbol{x} \in \mathcal{X}$ according to $\mathcal{P}_{\mathcal{X}}$

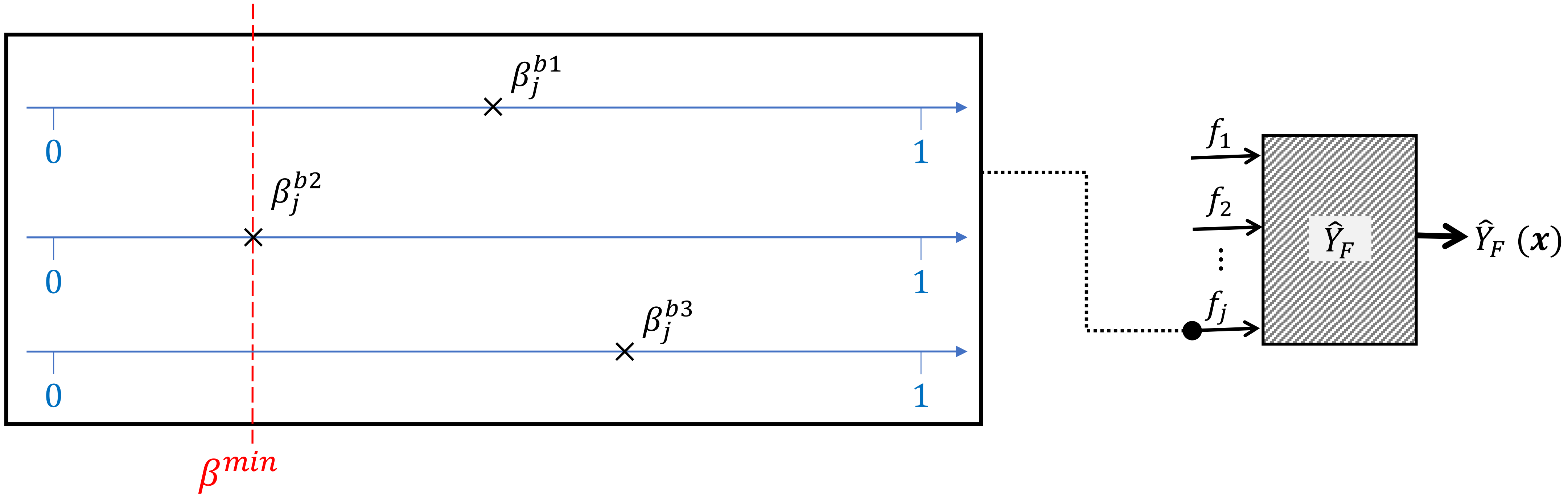
Model instability under imputation $\boldsymbol{\tau_{f_j,b}}$:

$$\beta_j^b = \mathbb{E}_{X \sim \mathcal{P}_{\mathcal{X}}}[I_{\hat{Y}_F}(X, f_j, b)]$$

Instability-based Data Minimization



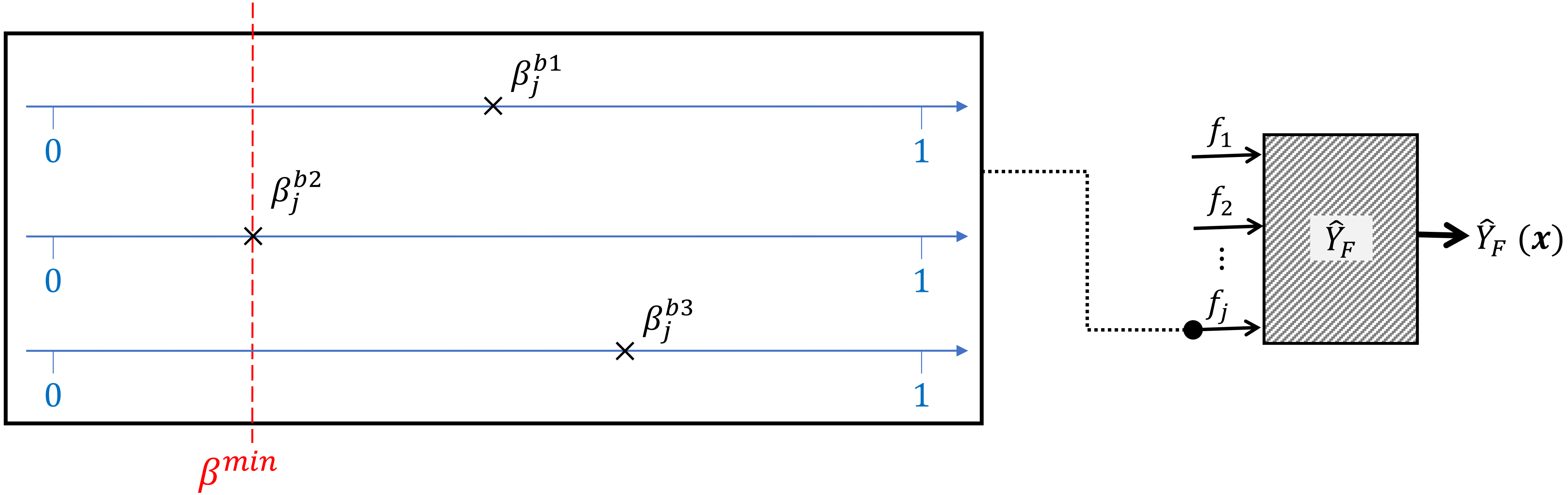
Instability-based Data Minimization



The imputation value that induces the minimum instability for each feature f_j, determines how necessary f_j is for generating the model outcomes.

Instability-based Data Minimization

The imputation value that induces the minimum instability for each feature f_j, determines how necessary f_j is for generating the model outcomes.



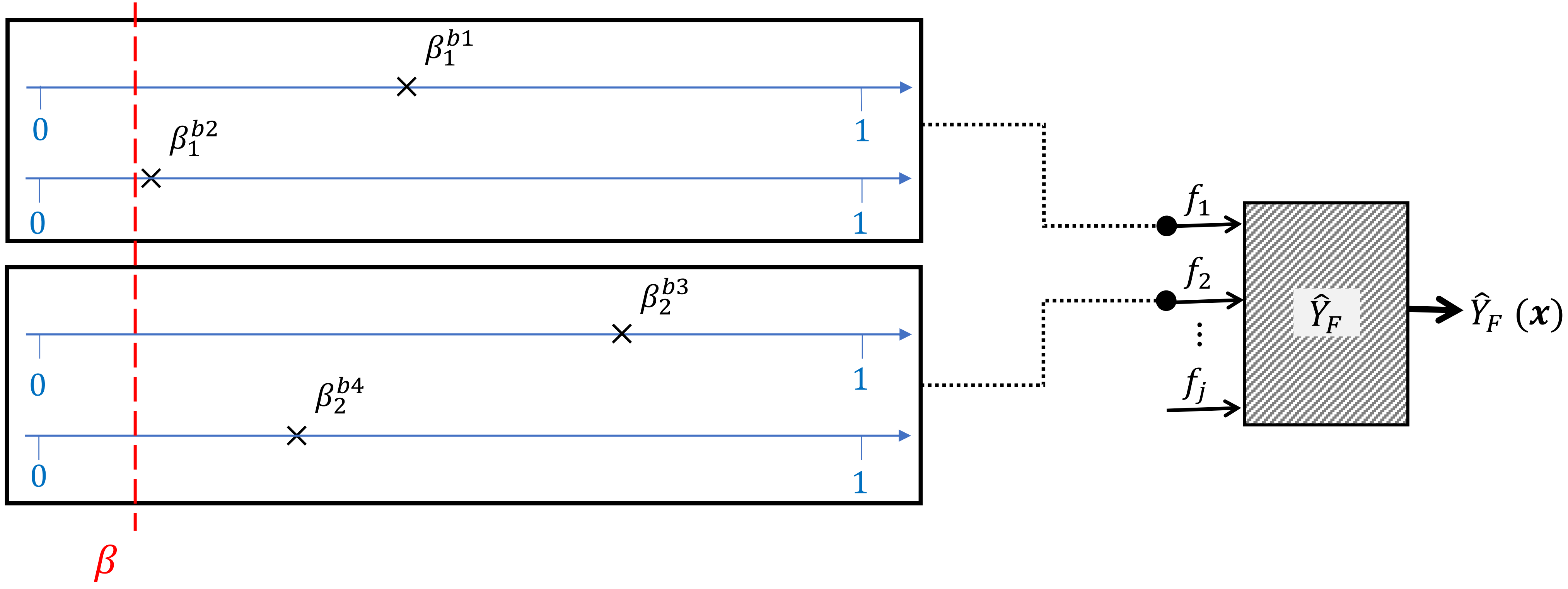
Limiting the class of imputations to simple imputations, for at least β^{min} fraction of prediction instances the value of f_j is necessary to reach the model predictions.

A Data Minimization Guarantee

$\hat{Y}_F$ satisfies data minimization at level β if there does not exist any feature $f_j \in F$ and any imputation value $b \in \mathcal{X}_j$ such that $\beta_j^b < \beta$.

A Data Minimization Guarantee

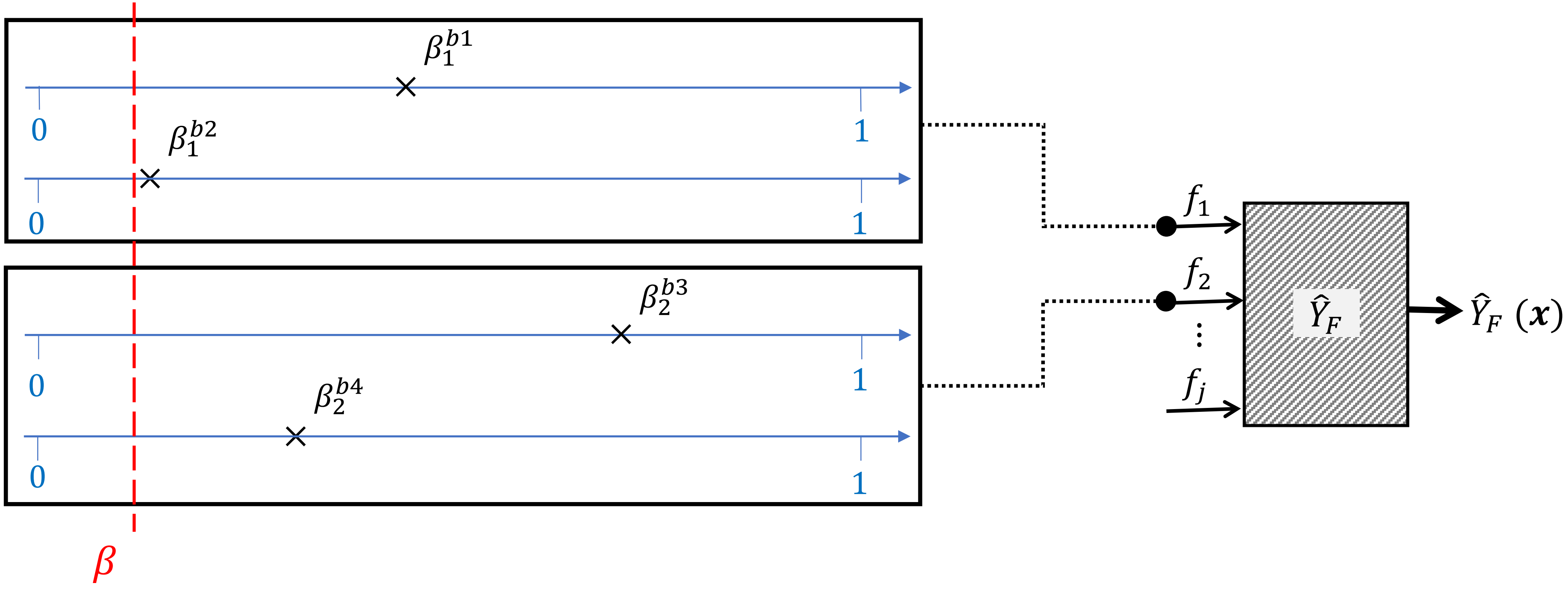
$\hat{Y}_F$ satisfies data minimization at level β if there does not exist any feature $f_j \in F$ and any imputation value $b \in \mathcal{X}_j$ such that $\beta_j^b < \beta$.



A level β guarantee ensures that every input feature used by the model is necessary to reach the predictions made for at least a fraction (β) of prediction instances.

A Data Minimization Guarantee

$\hat{Y}_F$ satisfies data minimization at level β if there does not exist any feature $f_j \in F$ and any imputation value $b \in \mathcal{X}_j$ such that $\beta_j^b < \beta$.

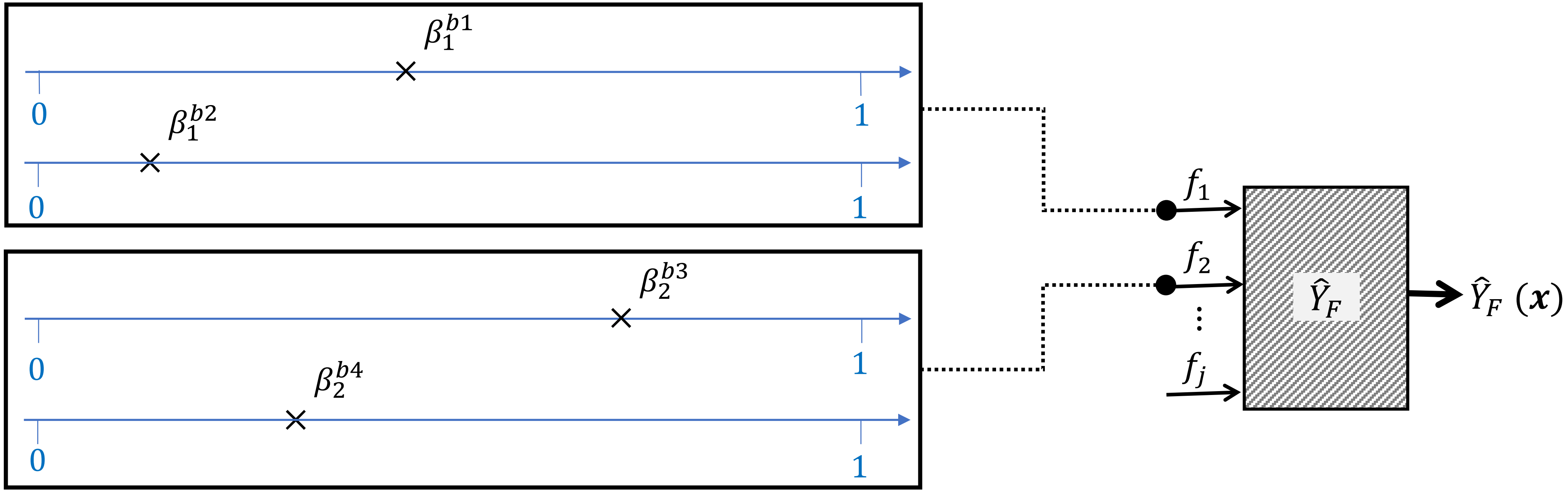


A level β guarantee ensures that every input feature used by the model is necessary to reach the predictions made for at least a fraction (β) of prediction instances.

Best data minimization guarantee
The greatest lower bound of all β_j^b's.

A Data Minimization Guarantee

$\hat{Y}_F$ satisfies data minimization at level β if there does not exist any feature $f_j \in F$ and any imputation value $b \in \mathcal{X}_j$ such that $\beta_j^b < \beta$.



How can an auditor provide such a data minimization guarantee?

Audit Mechanisms

The auditor requires knowledge of model instabilities under different imputations

$$\beta_j^b = \mathbb{E}_{X \sim \mathcal{P}_\mathcal{X}}[I_{\hat{Y}_F}(X, f_j, b)]$$

Audit Mechanisms

The auditor requires knowledge of model instabilities under different imputations

$$\beta_j^b = \mathbb{E}_{\underline{X \sim \mathcal{P}_\mathcal{X}}}[I_{\hat{Y}_F}(X, f_j, b)]$$

In practice, this expected value can only be estimated using system queries for different data samples $\boldsymbol{x} \sim \mathcal{P}_\mathcal{X}$

$I_{\hat{Y}_F}(\boldsymbol{x}, f_j, b)$ (A system query)

Audit Mechanisms

The auditor requires knowledge of model instabilities under different imputations

$$\beta_j^b = \mathbb{E}_{X\sim\mathcal{P}_\mathcal{X}}[I_{\hat{Y}_F}(X, f_j, b)]$$

In practice, this expected value can only be estimated using system queries for different data samples $\boldsymbol{x} \sim \mathcal{P}_\mathcal{X}$

$I_{\hat{Y}_F}(\boldsymbol{x}, f_j, b)$ (A system query)

Population Audit

Assuming a finite sample model given an audit dataset, instabilities can be estimated using the population mean.

$$\hat{\beta}_j^b = \frac{1}{|D^{Audit}|} \sum_{x \in D^{Audit}} [I_{\hat{Y}_F}(\boldsymbol{x}, f_j, b)]$$

Audit Mechanisms

The auditor requires knowledge of model instabilities under different imputations

$$\beta_j^b = \mathbb{E}_{X \sim \mathcal{P}_{\mathcal{X}}}[I_{\hat{Y}_F}(X, f_j, b)]$$

In practice, this expected value can only be estimated using system queries for different data samples $\boldsymbol{x} \sim \mathcal{P}_{\mathcal{X}}$

$I_{\hat{Y}_F}(\boldsymbol{x}, f_j, b)$ (A system query)

Population Audit

Assuming a finite sample model given an audit dataset, instabilities can be estimated using the population mean.

$$\hat{\beta}_j^b = \frac{1}{|D^{Audit}|} \sum_{x \in D^{Audit}} [I_{\hat{Y}_F}(\boldsymbol{x}, f_j, b)\,]$$

Not practical!

- The number of system queries is limited in practice.
- We are often interested in a guarantee that is valid for unseen samples from the underlying data distribution.

Probabilistic Audit

Use a limited number of system queries and provide a guarantee that is valid for the underlying data distribution.

Probabilistic Audit

Use a limited number of system queries and provide a guarantee that is valid for the underlying data distribution.

Probabilistic Data Minimization Guarantee

$\hat{Y}_F$ satisfies data minimization at level β with α percent confidence if:

$$\Pr\left[\exists\left(f_j \in F, b \in \mathcal{X}_j\right) s.t.\, \beta_j^b \leq \beta\right] \leq 1 - \alpha$$

Probabilistic Audit

Use a limited number of system queries and provide a guarantee that is valid for the underlying data distribution.

Probabilistic Data Minimization Guarantee

$\hat{Y}_F$ satisfies data minimization at level β with α percent confidence if:

$$\Pr\left[\exists\left(f_j \in F, b \in \mathcal{X}_j\right) s.t. \beta_j^b \leq \beta\right] \leq 1 - \alpha$$

Satisfying this guarantee at a high confidence means that with high probability every input feature is necessary to reach the predictions made for at least β fraction of samples drawn from $\mathcal{P}_{\mathcal{X}}$.

A Bayesian approach

Measure the uncertainty about the model instability under different imputations.

$$\beta_j^b = \mathbb{E}_{X \sim \mathcal{P}_{\mathcal{X}}}[I_{\hat{Y}_F}(X, f_j, b)]$$

A Bayesian approach

Measure the uncertainty about the model instability under different imputations.

$$\beta_j^b = \mathbb{E}_{X \sim \mathcal{P}_\mathcal{X}}[I_{\hat{Y}_F}(X, f_j, b)]$$

We model the success probability of $I_{\hat{Y}_F}(X, f_j, b)$ using a Beta distribution.

A Bayesian approach

Measure the uncertainty about the model instability under different imputations.

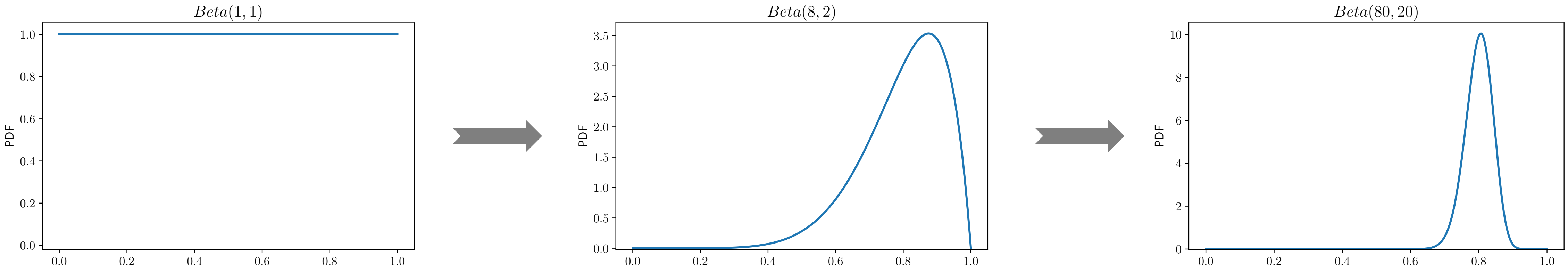
$$\beta_j^b = \mathbb{E}_{X \sim \mathcal{P}_{\mathcal{X}}}[I_{\hat{Y}_F}(X, f_j, b)]$$

We model the success probability of $I_{\hat{Y}_F}(X, f_j, b)$ using a Beta distribution.

Update rule: $\beta_j^b \sim Beta(a, c) \longrightarrow \beta_j^b \sim Beta(a + S_j^b, c + F_j^b)$

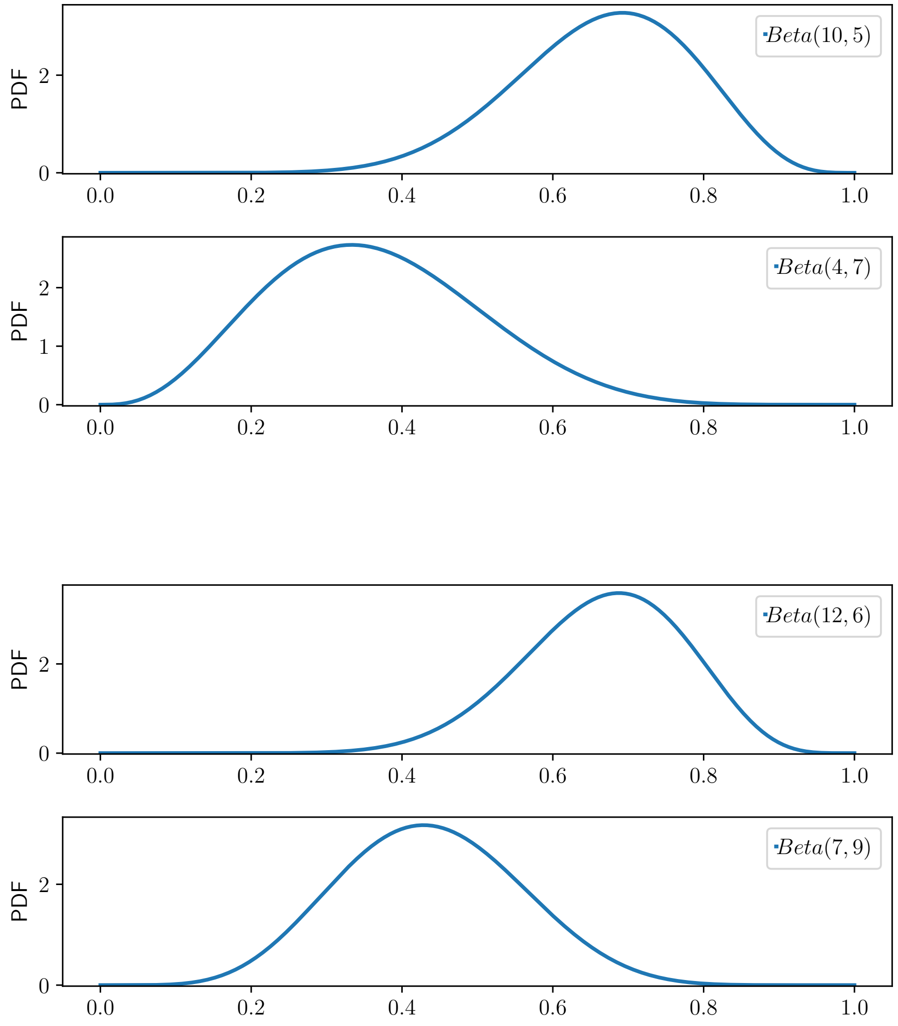
prior belief

success and failure counters



Inferring a probabilistic guarantee using posterior distributions

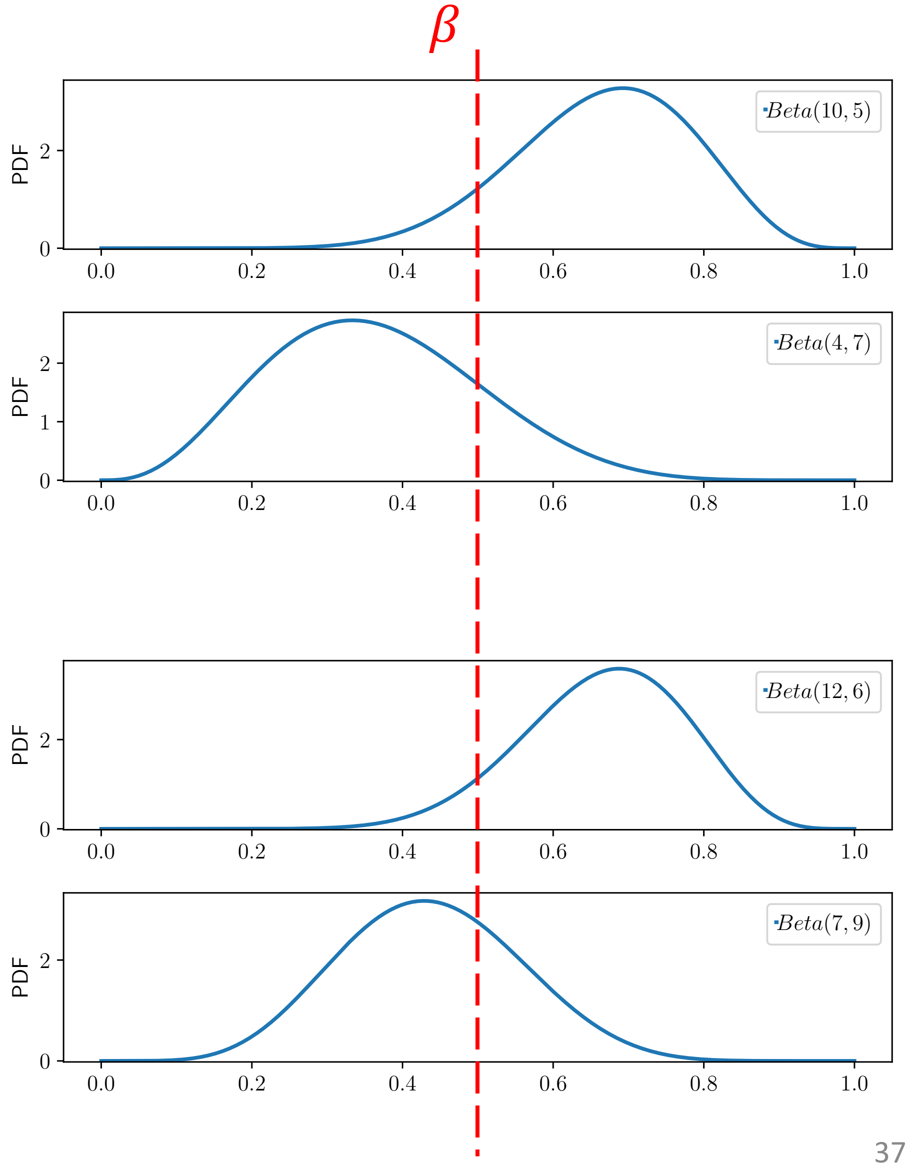
How to infer a probabilistic data minimization guarantee using all the resulting posteriors?



Inferring a probabilistic guarantee using posterior distributions

Verify whether with high probability all β_j^b's are greater than some level β.

$$\Pr\left[\text{for at least one } (f_j, b);\ \beta_j^b \leq \beta\right]$$



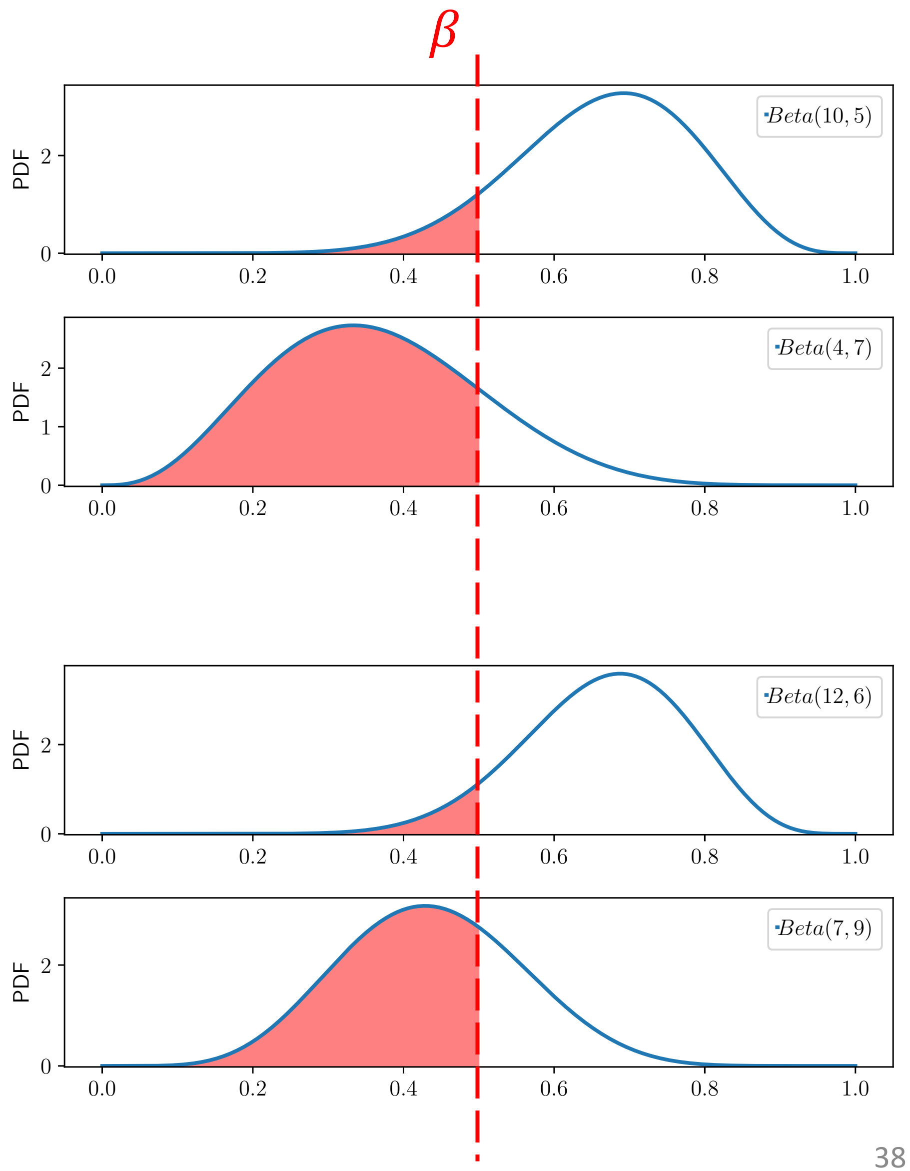
Inferring a probabilistic guarantee using posterior distributions

Verify whether with high probability all β_j^b's are greater than some level β.

Boole's inequality

$$\Pr\left[\text{for at least one } (f_j, b);\ \beta_j^b \le \beta\right] \le \sum_{f_j, b} \Pr[\beta_j^b \le \beta]$$

CDF function of $\beta_j^b := L_j^b(\beta)$



Inferring a probabilistic guarantee using posterior distributions

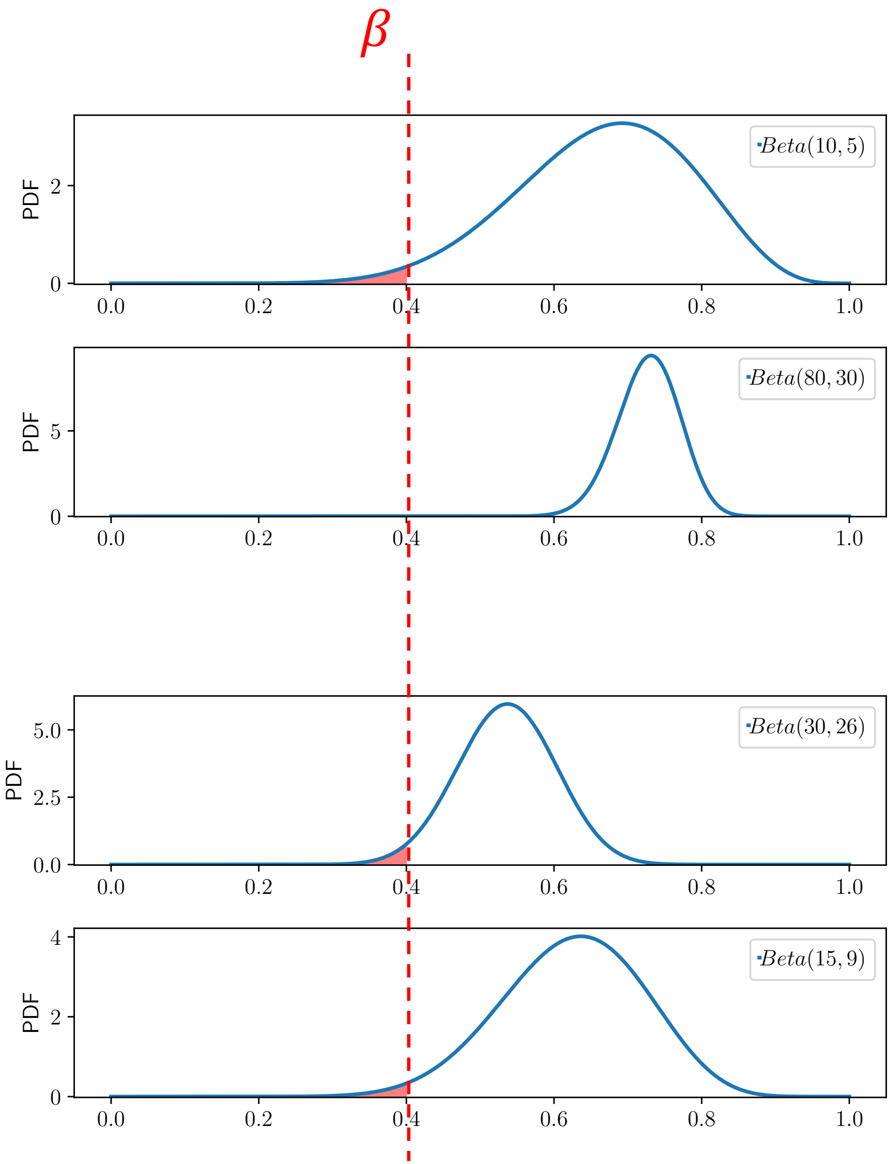
Verify whether with high probability all β_j^b's are greater than some level β.

Boole's inequality

$$\Pr\left[\text{for at least one } (f_j, b);\ \beta_j^b \le \beta\right] \le \sum_{f_j,b} \Pr[\beta_j^b \le \beta]$$

$$\le (1-\alpha)$$

Data minimization is satisfied at level β



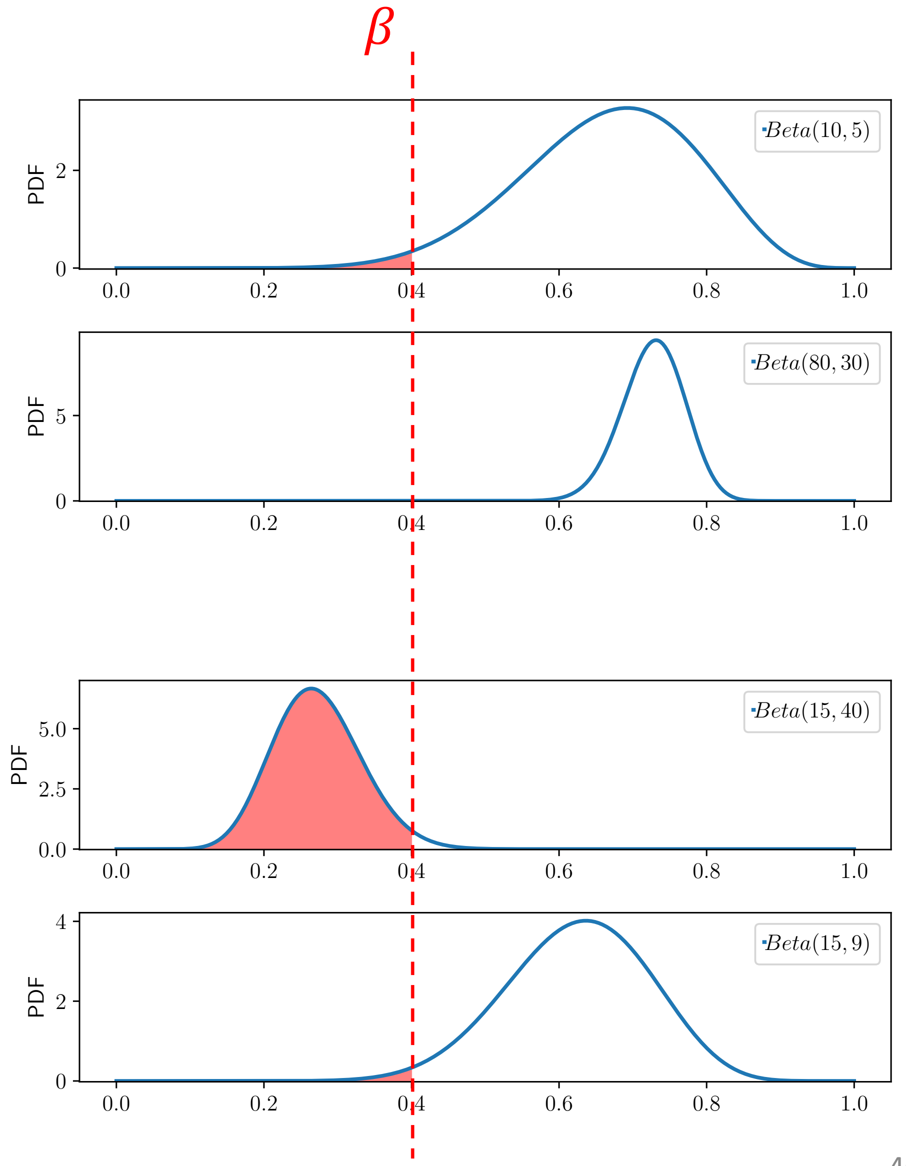
Inferring a probabilistic guarantee using posterior distributions

Verify whether with high probability all β_j^b's are greater than some level β.

$$\max_{f_j,b} \Pr[\beta_j^b \leq \beta] \leq \ \Pr[\text{for at least one } (f_j, b);\, \beta_j^b \leq \beta]$$

$\alpha \leq$

Data minimization is not satisfied at level β

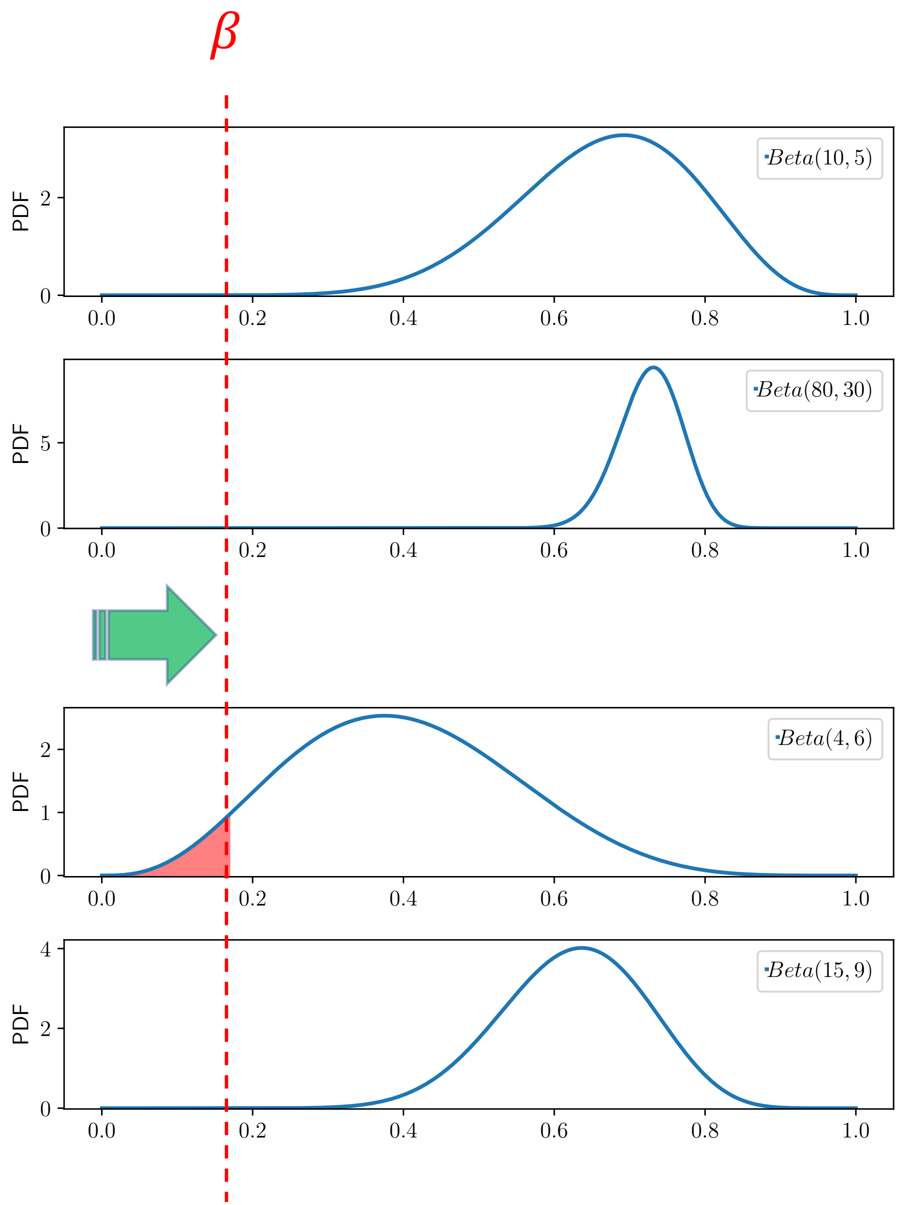


Inferring a probabilistic guarantee using posterior distributions

Measure the best data minimization level that can be guaranteed with confidence α:

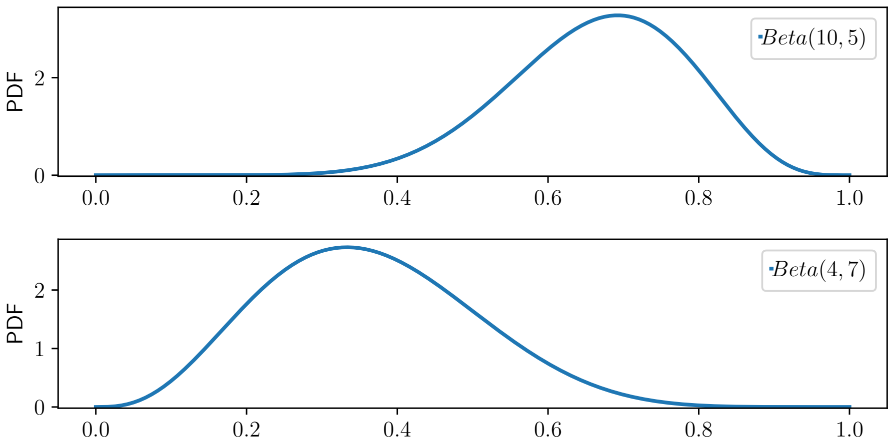
$$\Pr\left[\text{for at least one } (f_j, b);\ \beta_j^b \leq \beta\right] \leq \underbrace{\sum_{f_j,b} \Pr[\beta_j^b \leq \beta]}_{\text{monotonically increasing function of } \beta}$$

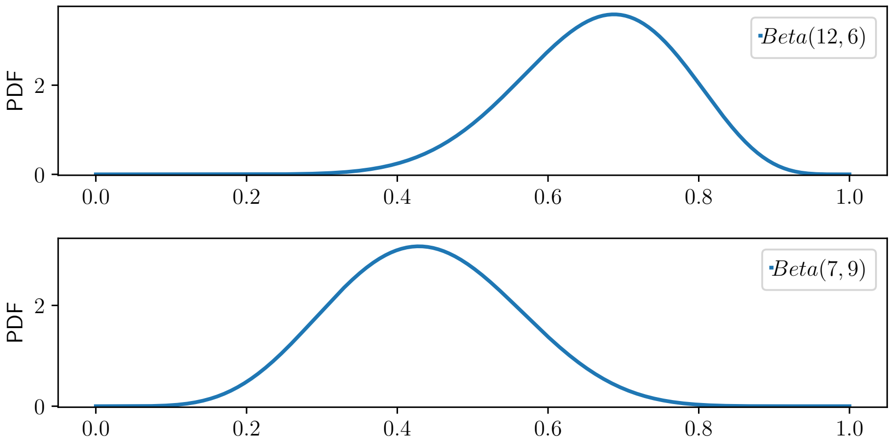
Apply a binary search to find β



Auditing with a Limited Query Budget

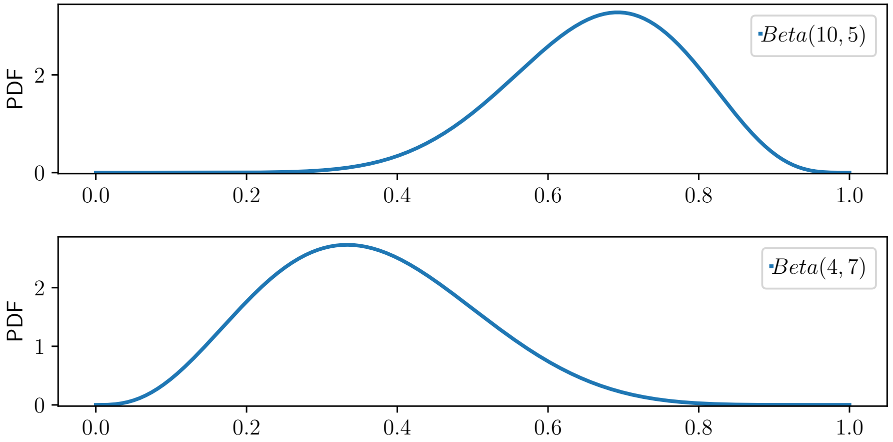
Updating certain posteriors are more helpful in finding a data minimization guarantee with high confidence.



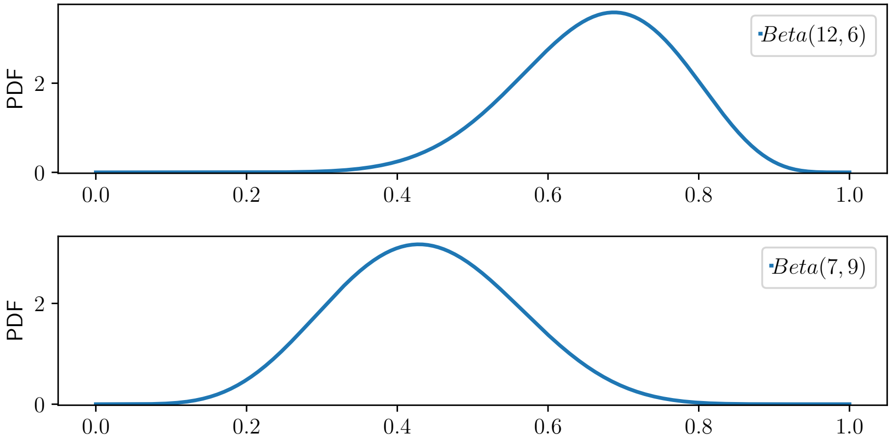


Auditing with a Limited Query Budget

Updating certain posteriors are more helpful in finding a data minimization guarantee with high confidence.

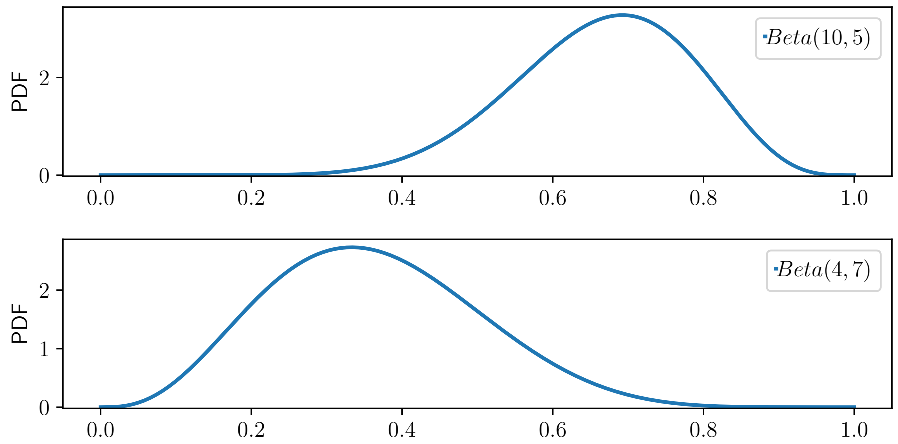


Given a limited query budget, what is the best strategy to allocate system queries for measuring model instability under different imputations?

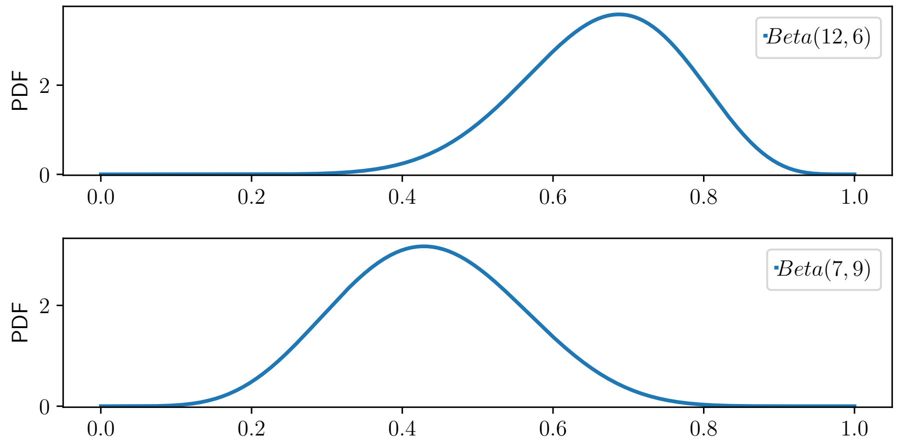


Auditing with a Limited Query Budget

Updating certain posteriors are more helpful in finding a data minimization guarantee with high confidence.



Given a limited query budget, what is the best strategy to allocate system queries for measuring model instability under different imputations?

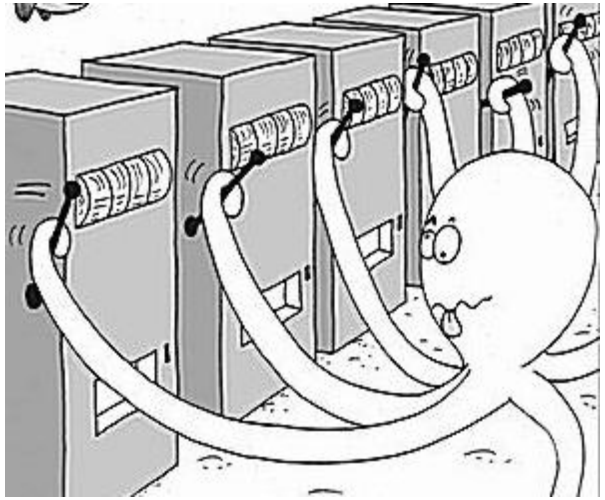


We cast this problem into a multi-armed bandit framework.

A multi-armed bandit framework

Sequential decision problems under uncertainty

- Actions (choices) are defined by a set of arms.
- A player sequentially chooses arms to play and observes noisy signals of their quality (reward).
- The goal is to optimize some utility while acquiring new knowledge about the arms.



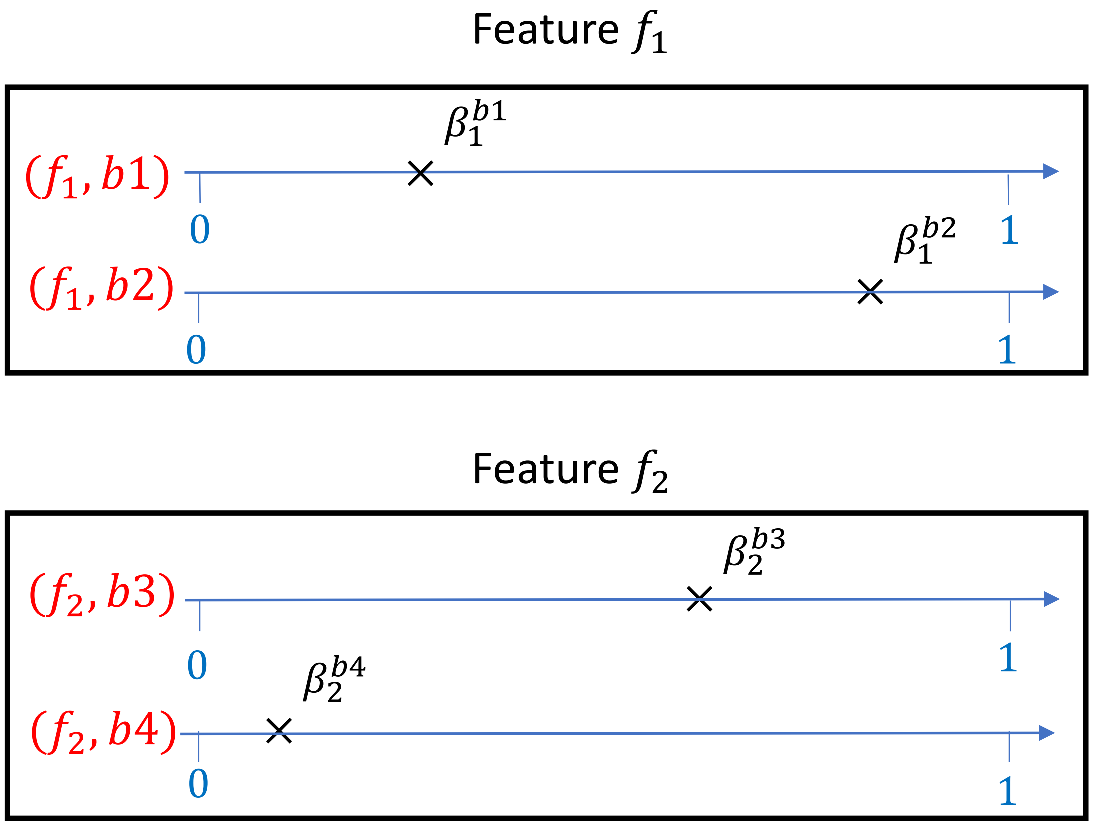
A multi-armed bandit framework

Stochastic Bernoulli bandit

We consider an arm for each feasible imputation (f_j, b).

Success probabilities (instabilities) are unknown.

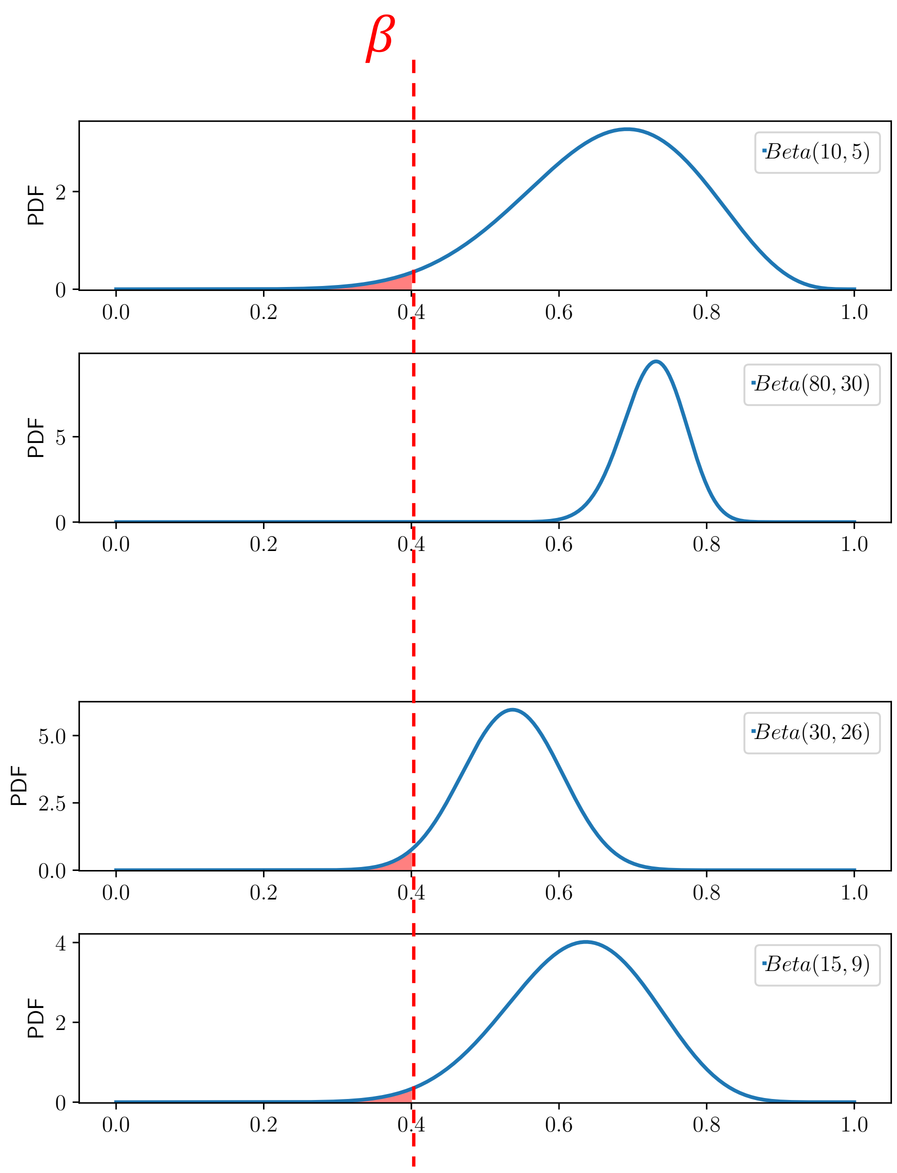
Playing arm $(\boldsymbol{f_j}, \boldsymbol{b})$:
observe a binary reward using a random data sample and a system query.



Two bandit problems

Decision Problem

Given a confidence and a data minimization level, iteratively select and explore arms such that a decision can be made using the minimum number of observations.



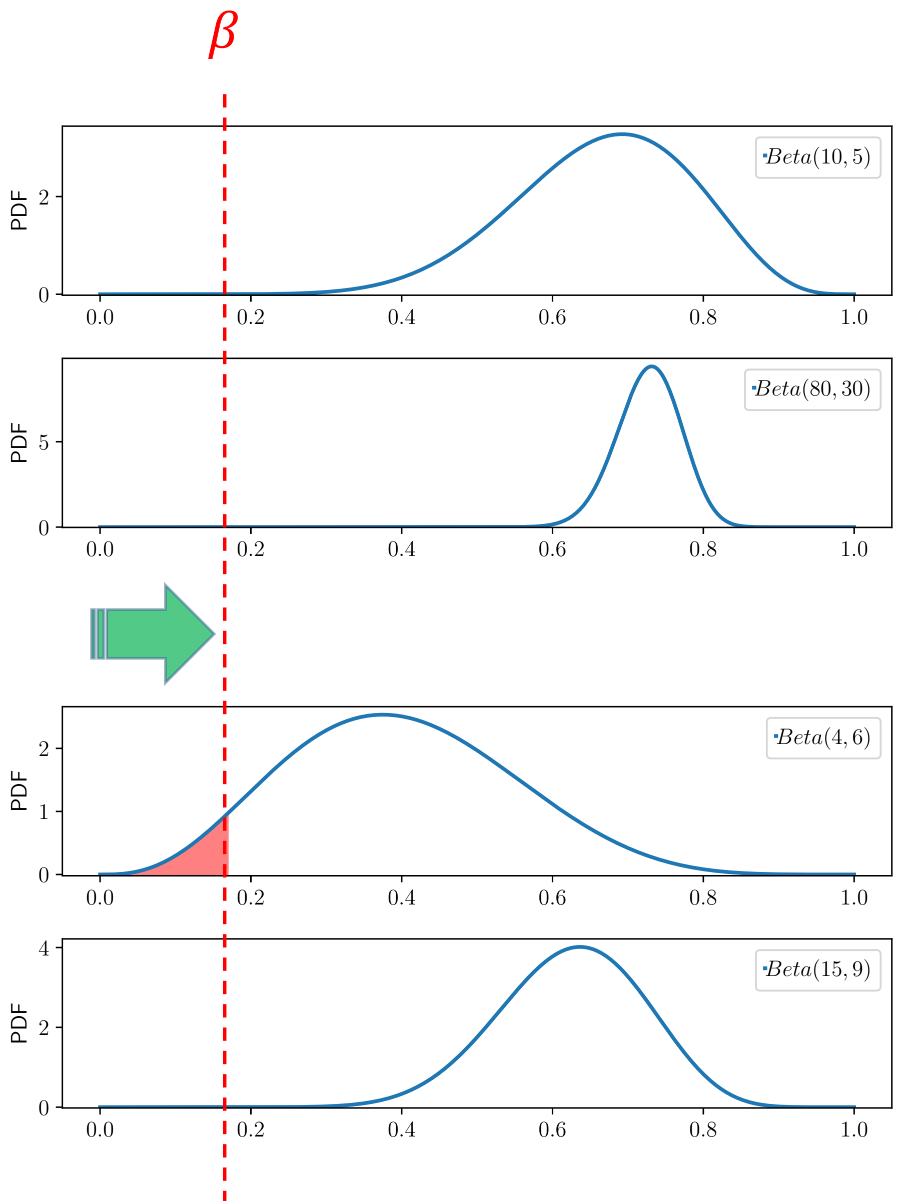
Two bandit problems

Decision Problem

Given a confidence and a data minimization level, iteratively select and explore arms such that a decision can be made using the minimum number of observations.

Measurement Problem

Given a confidence and a fixed query budget, iteratively select and explore arms such that after using all the budget the guaranteed data minimization level is maximized.



Two bandit problems

Decision Problem

Given a confidence and a data minimization level, iteratively select and explore arms such that a decision can be made using the minimum number of observations.

Measurement Problem

Given a confidence and a fixed query budget, iteratively select and explore arms such that after using all the budget the guaranteed data minimization level is maximized.

Both require an exploration strategy for selecting the next arm to investigate.

Exploration Strategies

Strategies based on Thompson Sampling

Exploration Strategies

Strategies based on Thompson Sampling

Thompson Sampling (TS): a heuristic that combines Bayesian modeling with probability matching.

Choose arms according to their probability of having the minimum mean reward.

Choose imputations that reducing the uncertainty about their success probability would better help finding a lower bound on all instabilities.

Exploration Strategies

Strategies based on Thompson Sampling

Thompson Sampling (TS): a heuristic that combines Bayesian modeling with probability matching.

Choose arms according to their probability of having the minimum mean reward.

Choose imputations that reducing the uncertainty about their success probability would better help finding a lower bound on all instabilities.

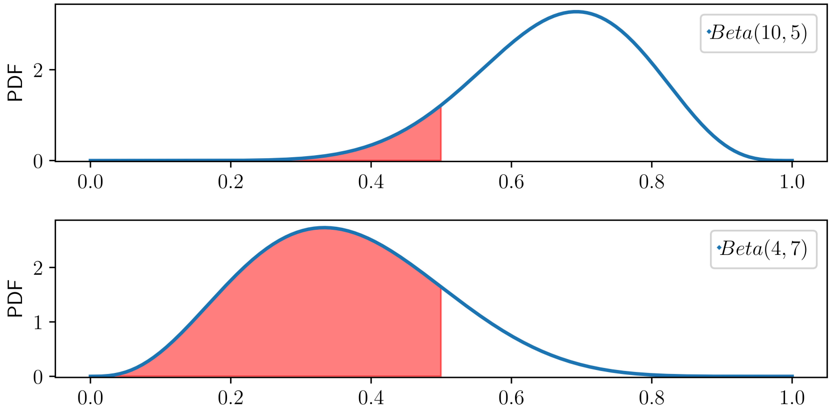
Top-Two Thompson Sampling (TTTS):

A modification to TS for sampling less explored arms more frequently.

Idea: randomly choose between two of the best alternatives.

Exploration Strategies

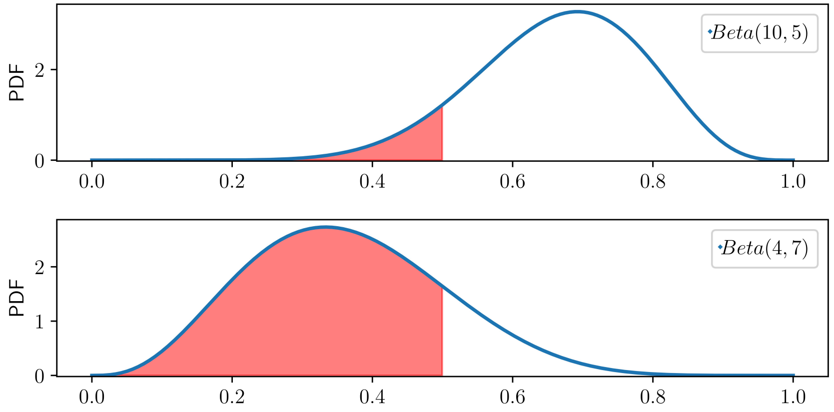
Our data minimization guarantee depends on the probability mass that is below some threshold in all arms.



We introduce two exploration strategies designed specifically for obtaining a data minimization guarantee.

Exploration Strategies

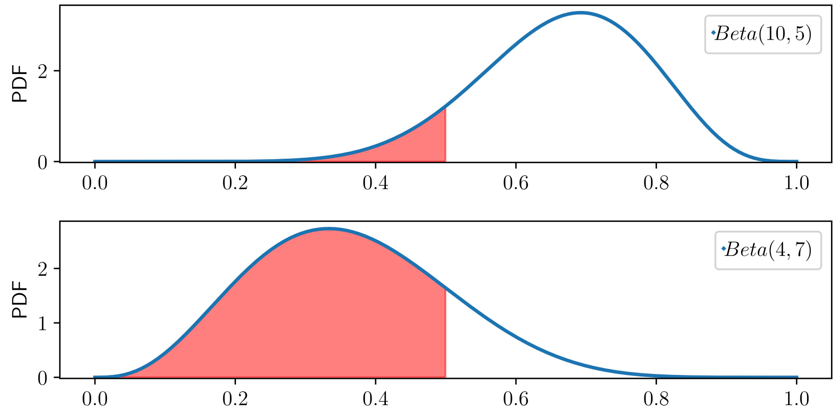
Our data minimization guarantee depends on the probability mass that is below some threshold in all arms.



We introduce two exploration strategies designed specifically for obtaining a data minimization guarantee.

Greedy
Select the arm whose posterior beta distribution has the maximum probability mass below a threshold β.

Exploration Strategies



Our data minimization guarantee depends on the probability mass that is below some threshold in all arms.

We introduce two exploration strategies designed specifically for obtaining a data minimization guarantee.

Probability Matching Using CDFs (PM)
Select arms in proportion to the amount of probability mass that is below β in each posterior distribution.

Auditing Real-world Prediction Systems

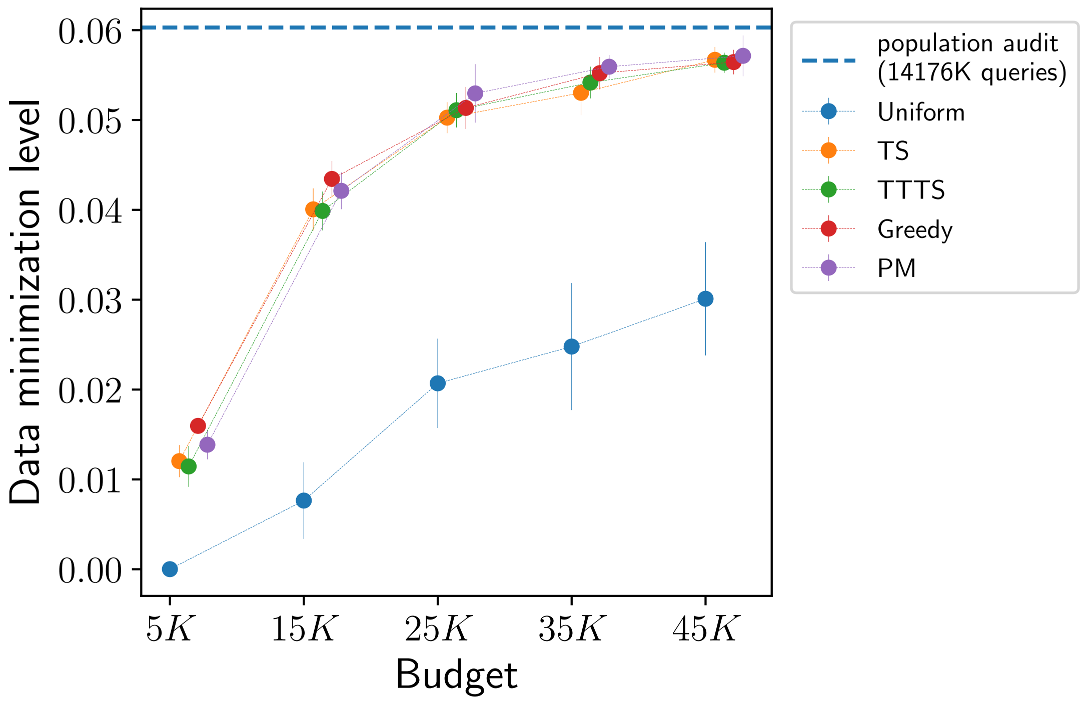
Census/Decision Tree

- A decision tree is built to predict whether a person makes over $\$50K$ a year using the US Census database.

- After applying standard model validation and feature selection procedures we get a black-box prediction model with 5 input features.

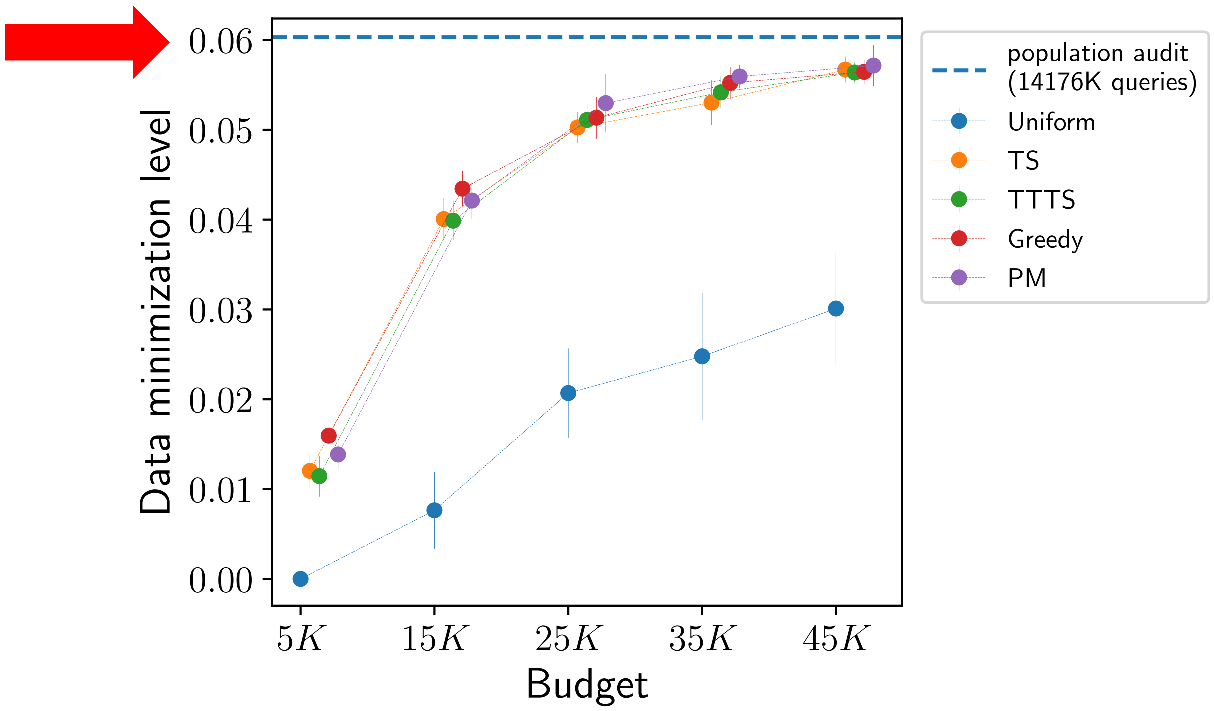
Auditing Real-world Prediction Systems

Measurement Task with 95% confidence



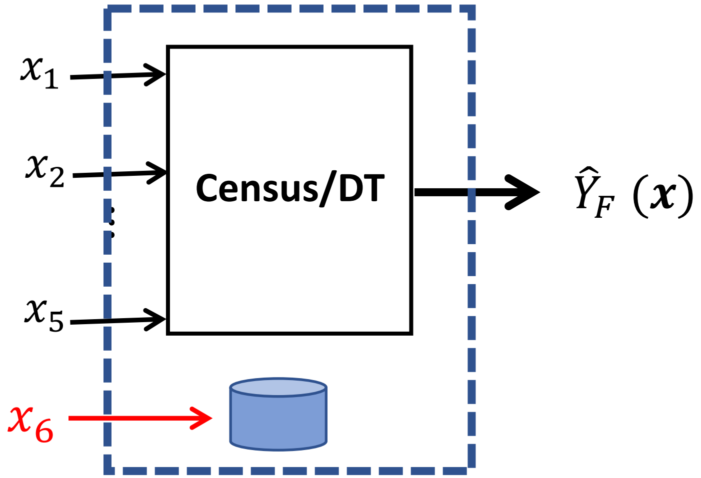
Auditing Real-world Prediction Systems

Measurement Task with 95% confidence



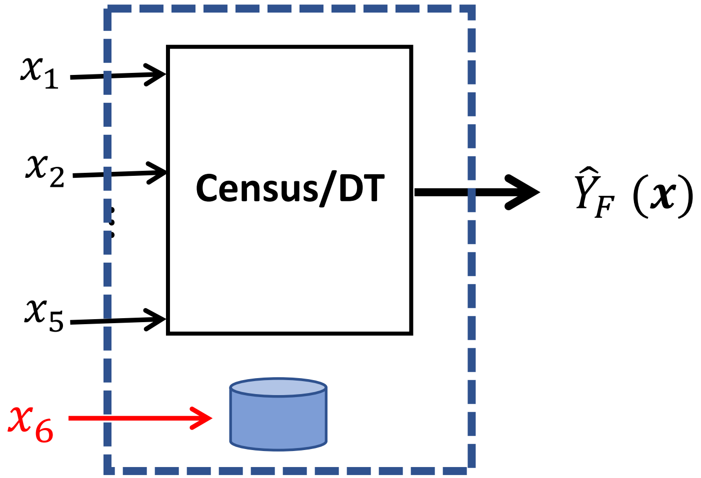
Auditing Real-world Prediction Systems

A system that collects excessive information

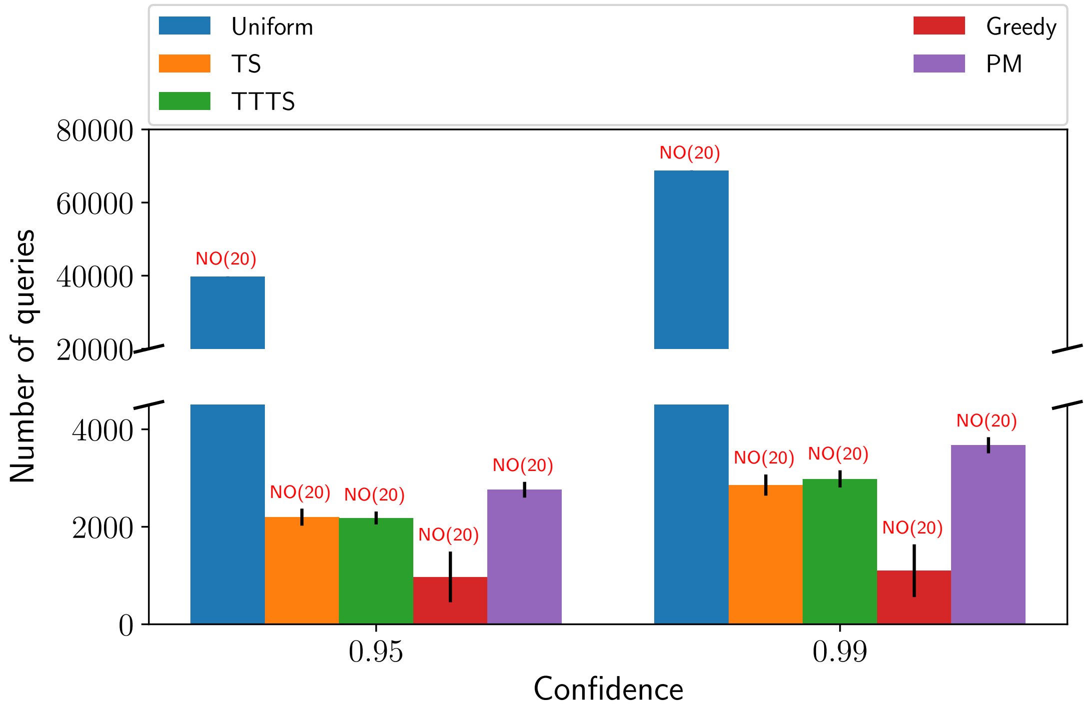


Auditing Real-world Prediction Systems

A system that collects excessive information



Decision Task at 1% Data Minimization Level



Auditing Black-Box Prediction Models for Data Minimization Compliance

In summary, we

- Propose an operationalization of data minimization for auditing black-box prediction models.
- Define a guarantee that is based on a metric of model instability under simple imputations.
- Extend the applicability of our metric from a finite sample model to a distributional setting by introducing a probabilistic guarantee and a Bayesian approach.
- Formulate the problem of auditing data minimization with a limited query budget as a multi-armed bandit framework for which we design efficient exploration strategies.