# Introduction to Machine Learning

**Multi-Class Classification**
**and**
**Bayesian optimization**

# Multiclass classification

- More than two classes

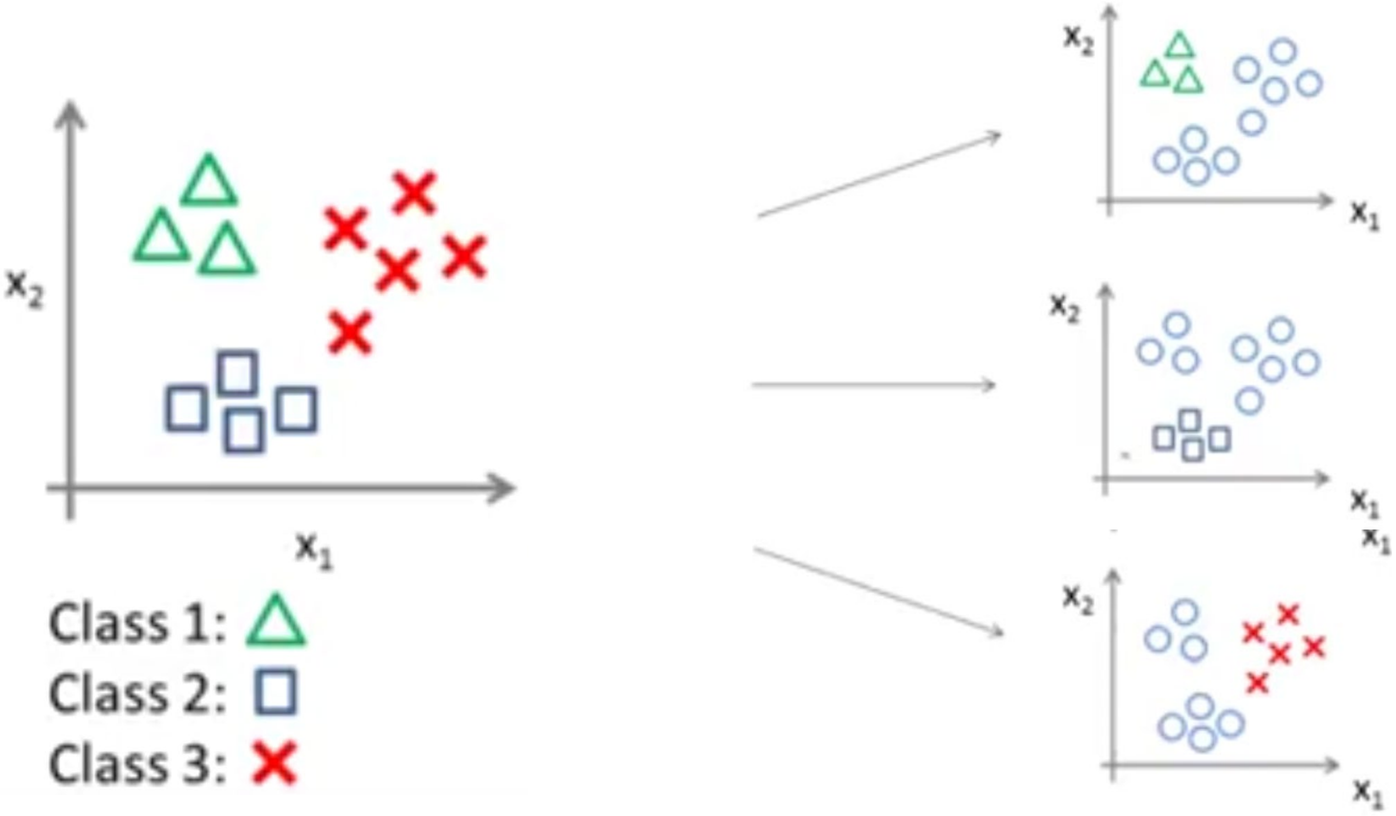
# Confusion matrix

Predict →

Actual →

| | Enter | Leave | Cook | Sleep | Meds | Eat | Groom | Bathe | Bed-T | Relax |
|---|---|---|---|---|---|---|---|---|---|---|
| Enter | 1673 | 27 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Leave | 9 | 1979 | 5 | 2 | 0 | 1 | 0 | 0 | 0 | 0 |
| Cook | 59 | 58 | 51238 | 39 | 199 | 137 | 28 | 2 | 0 | 0 |
| Sleep | 21 | 29 | 5 | 30795 | 4 | 86 | 14 | 0 | 51 | 0 |
| Meds | 11 | 2 | 200 | 0 | 3105 | 1 | 0 | 0 | 0 | 0 |
| Eat | 3 | 3 | 6 | 94 | 1 | 14278 | 5 | 0 | 0 | 0 |
| Groom | 0 | 11 | 4 | 1 | 1 | 3 | 21833 | 33 | 41 | 0 |
| Bathe | 0 | 0 | 0 | 1 | 0 | 0 | 59 | 592 | 5 | 0 |
| Bed-T | 0 | 0 | 0 | 18 | 0 | 0 | 15 | 2 | 501 | 0 |
| Relax | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |

# One-versus-all (one-vs-rest)



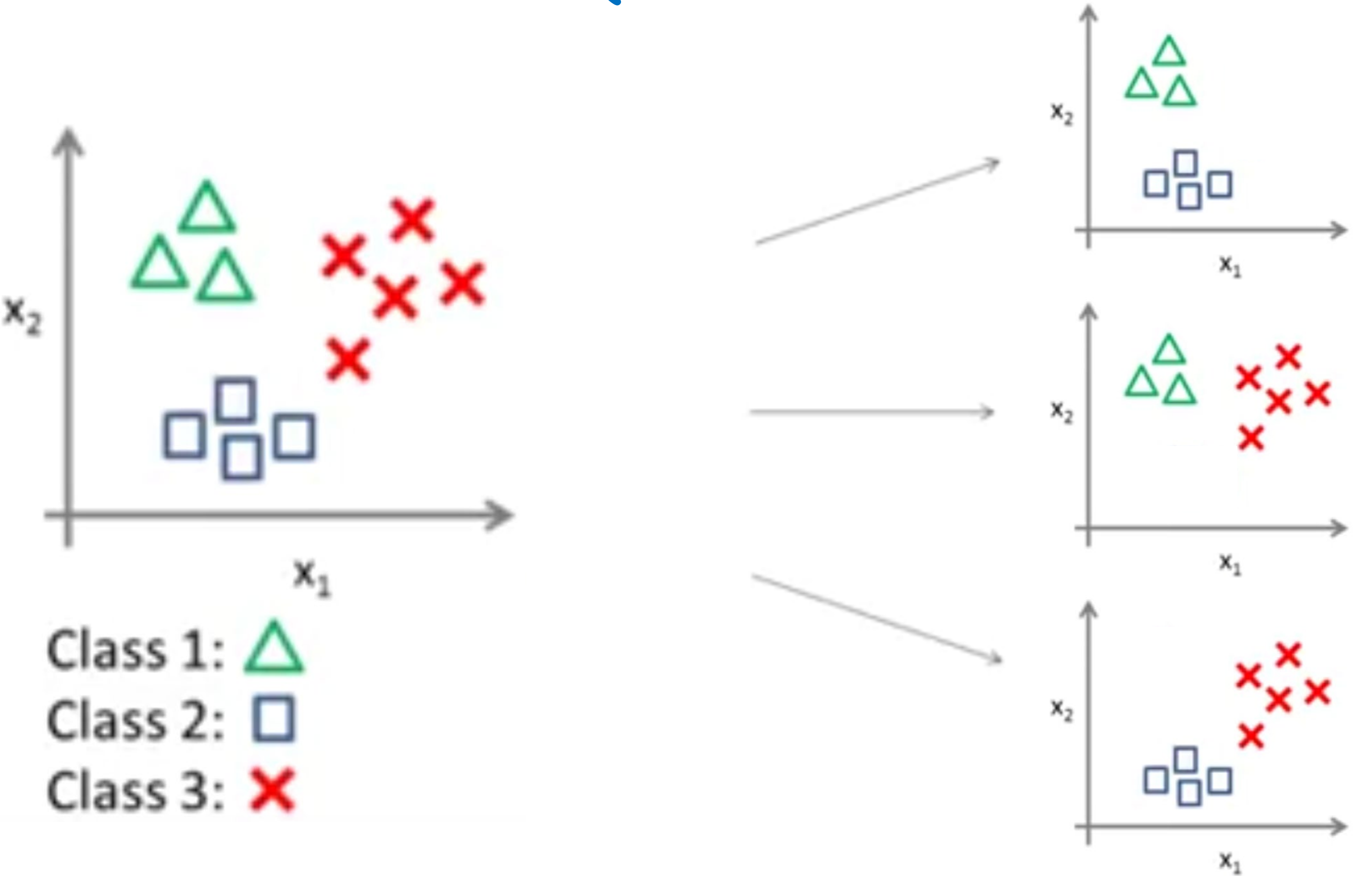
**Algorithm 13** OneVersusAllTrain($\mathbf{D}^{multiclass}$, BinaryTrain)

```
for $i = 1$ to $K$ do
    $\mathbf{D}^{bin} \leftarrow$ relabel $\mathbf{D}^{multiclass}$ so class $i$ is positive and $\neg i$ is negative
    $f_i \leftarrow$ BinaryTrain($\mathbf{D}^{bin}$)
end for
return $f_1, \dots, f_K$
```

**Algorithm 14** OneVersusAllTest($f_1, \dots, f_K, \hat{x}$)

```
$score \leftarrow \langle 0, 0, \dots, 0 \rangle$          // initialize $K$-many scores to zero
for $i = 1$ to $K$ do
    $y \leftarrow f_i(\hat{x})$
    $score_i \leftarrow score_i + y$
end for
return $\operatorname{argmax}_k score_k$
```

# All-versus-all (one-versus-one)



**Algorithm 15** AllVersusAllTrain($\mathbf{D}^{multiclass}$, BinaryTrain)

```
$f_{ij} \leftarrow \emptyset, \forall 1 \leq i < j \leq K$
for $i$ = 1 to $K$-1 do
    $\mathbf{D}^{pos} \leftarrow$ all $x \in \mathbf{D}^{multiclass}$ labeled $i$
    for $j$ = $i$+1 to $K$ do
        $\mathbf{D}^{neg} \leftarrow$ all $x \in \mathbf{D}^{multiclass}$ labeled $j$
        $\mathbf{D}^{bin} \leftarrow \{(x, +1) : x \in \mathbf{D}^{\text{pos}}\} \cup \{(x, -1) : x \in \mathbf{D}^{\text{neg}}\}$
        $f_{ij} \leftarrow$ BinaryTrain($\mathbf{D}^{bin}$)
    end for
end for
return all $f_{ij}$s
```

**Algorithm 16** AllVersusAllTest(all $f_{ij}$, $\hat{x}$)

```
$score \leftarrow \langle 0, 0, \dots, 0 \rangle$                // initialize $K$-many scores to zero
for $i$ = 1 to $K$-1 do
    for $j$ = $i$+1 to $K$ do
        $y \leftarrow f_{ij}(\hat{x})$
        $score_i \leftarrow score_i + y$
        $score_j \leftarrow score_j - y$
    end for
end for
return $\text{argmax}_k \; score_k$
```

# Binary tree of classifiers

# Overrun by hyperparameters

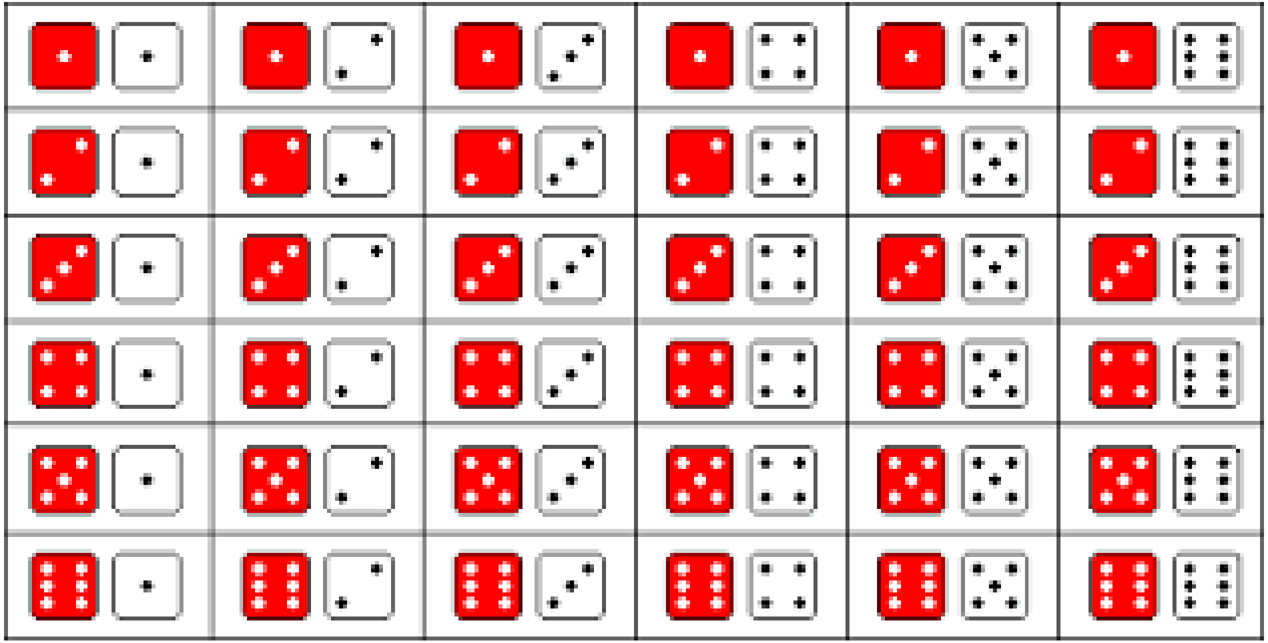
- Manual
- Grid search
- Random search

# Bayesian optimization to the rescue?

- Uses Bayes Theorem to direct the search

$$\text{Probability(event)} = \text{P(event)} = \frac{\text{\#instances of the event}}{\text{total \#instances}}$$

# Roll two dice

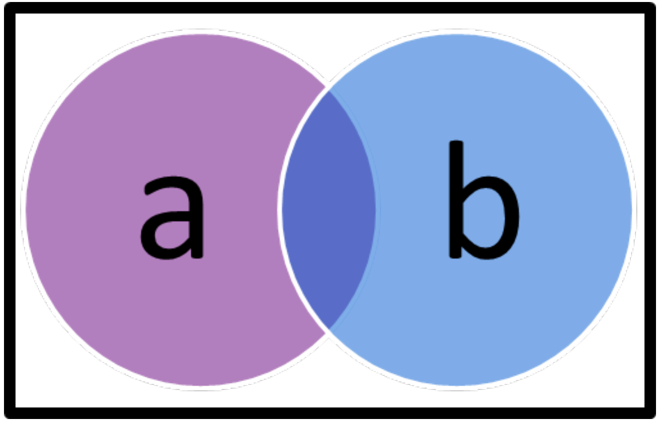


# Sources of probabilities

- Frequency
- Consider the probability that the sun will still exist tomorrow.

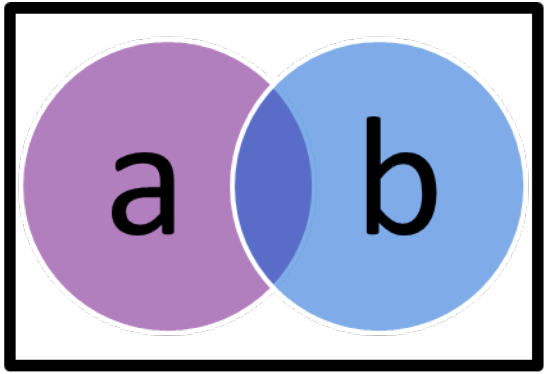
# Axioms of probability

- $0 \leq P(\text{Event}) \leq 1$
- Disjunction, P(a or b) = P(a) + P(b) – P(a and b)



# Conditional probability and conjunction

- P(a|b) = P(a and b) / P(b)



# Conditional probability and conjunction

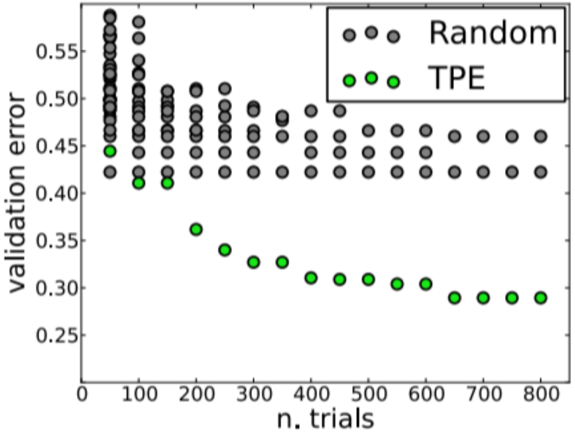
- $P(a \text{ and } b) = P(a) \times P(b|a)$
- $P(a \text{ and } b) = P(b) \times P(a|b)$

- If a and b are independent events
  - $P(a \text{ and } b) = P(a) \times P(b)$

# Bayes' rule

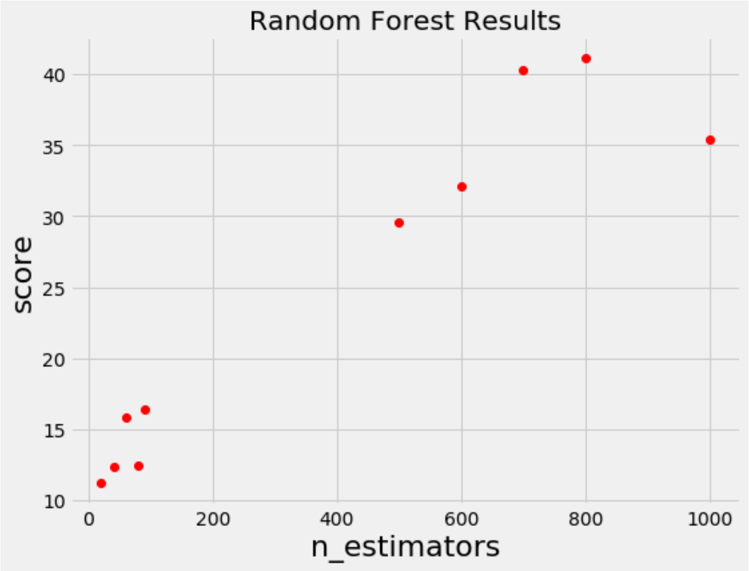
# Bayesian optimization to the rescue?

- Optimization method to solve arg $min_{x \in X} f(x)$



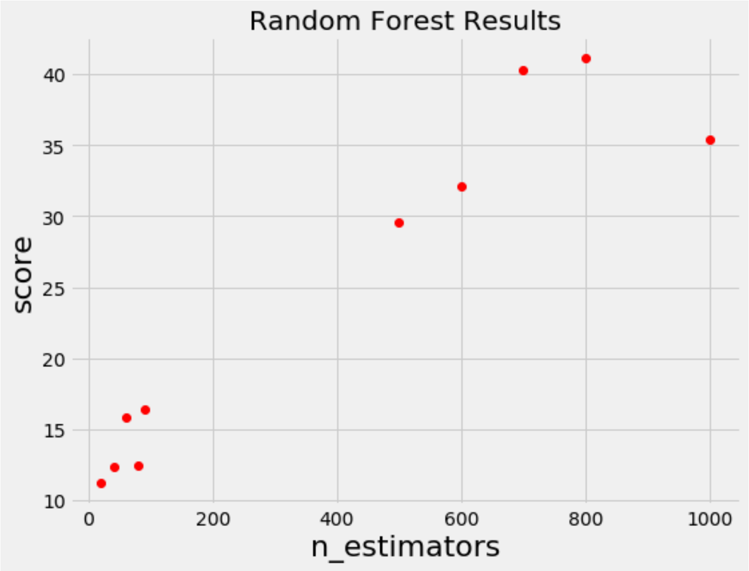
# Bayesian optimization

- Build probability model of objective function
- Use model to select hyperparameters to evaluate

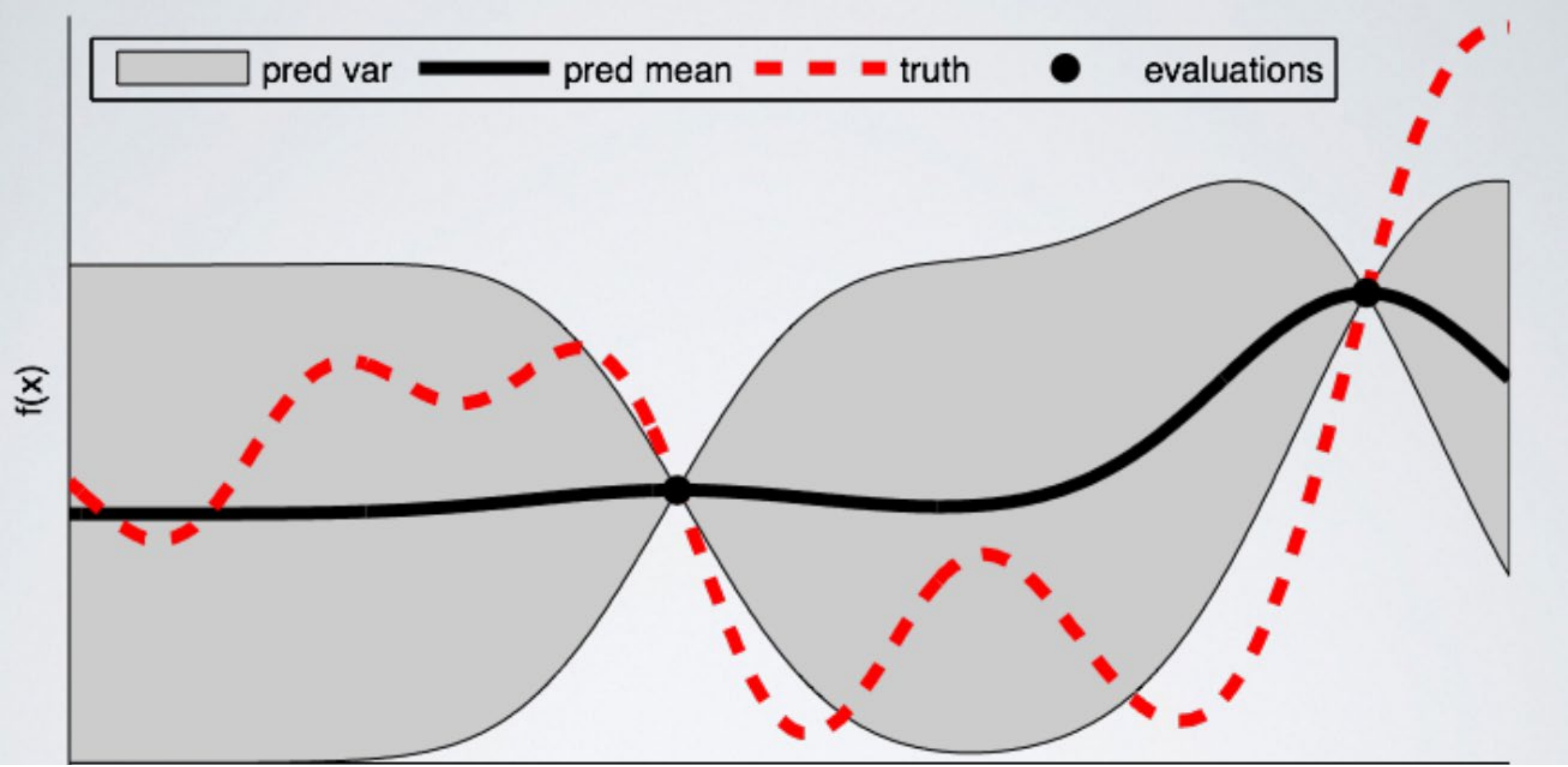


# Bayesian optimization

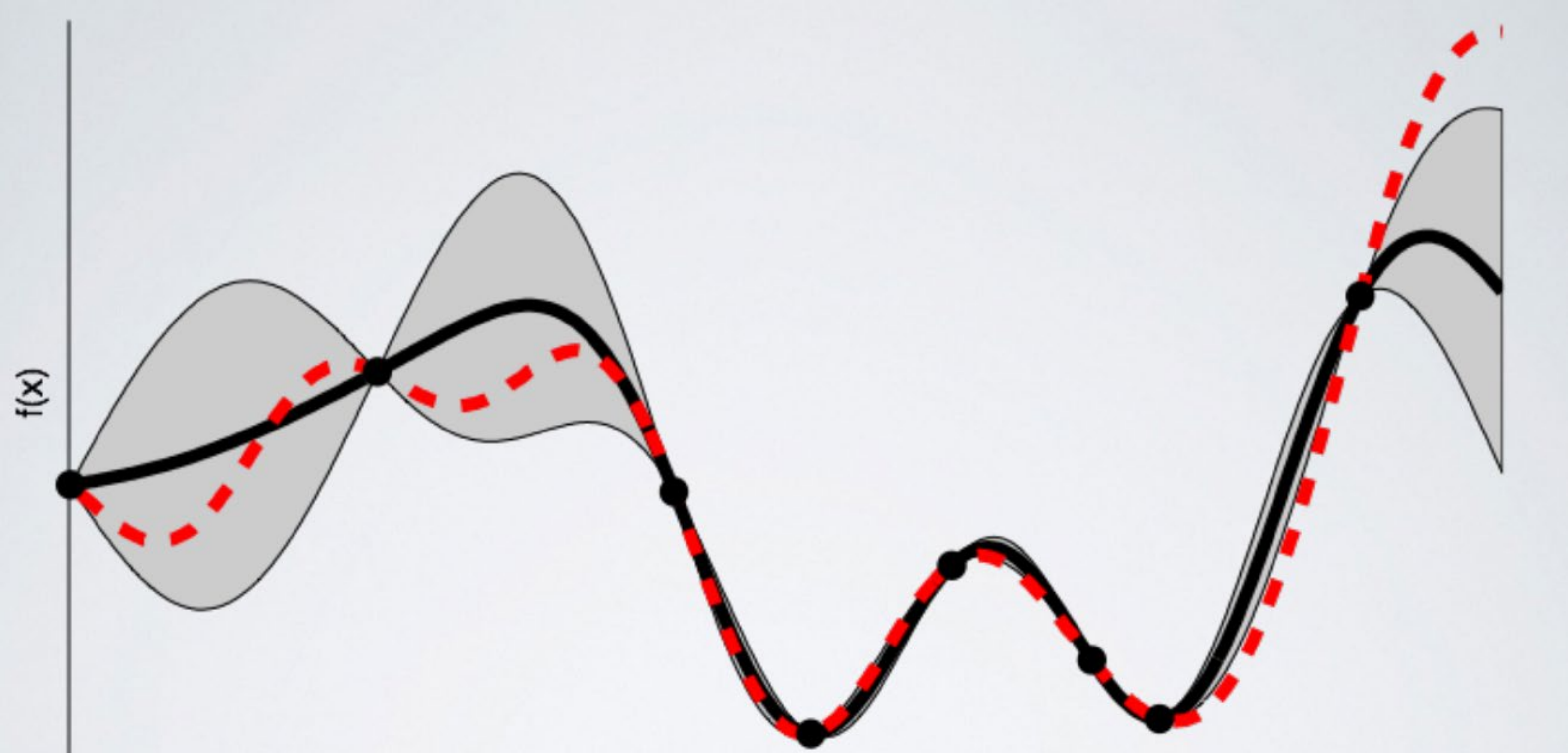
- P(score | hyperparameters)



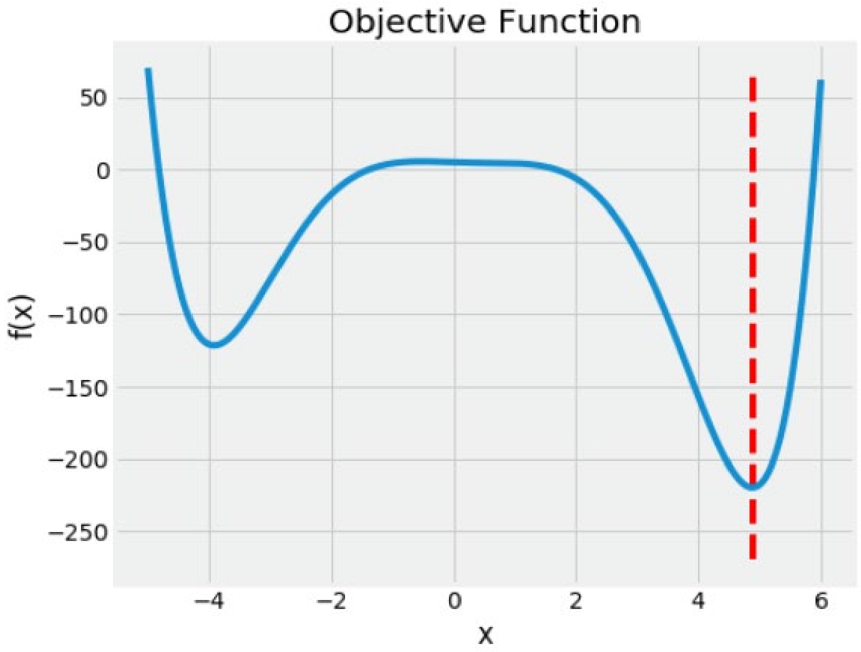
# Surrogate model

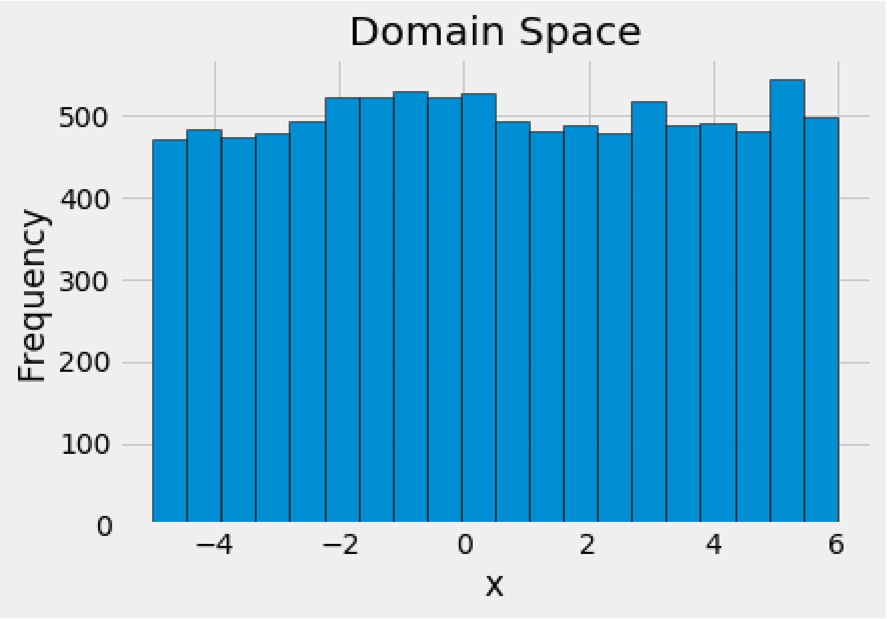


# Surrogate model

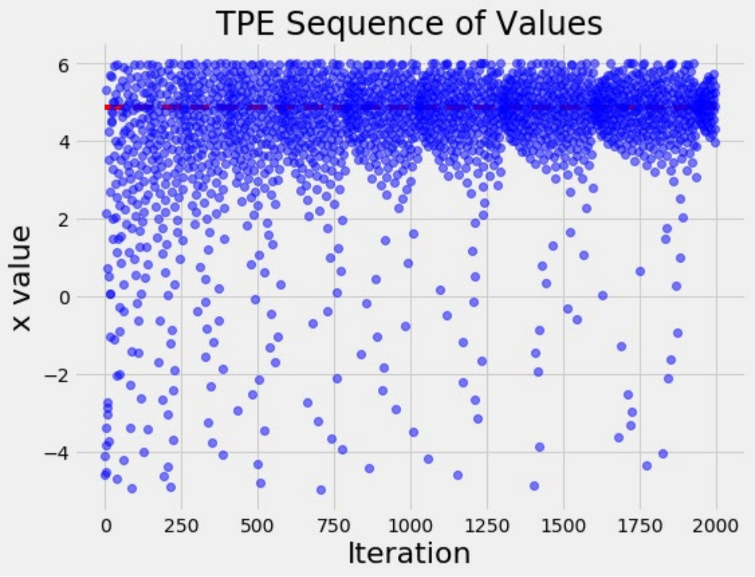


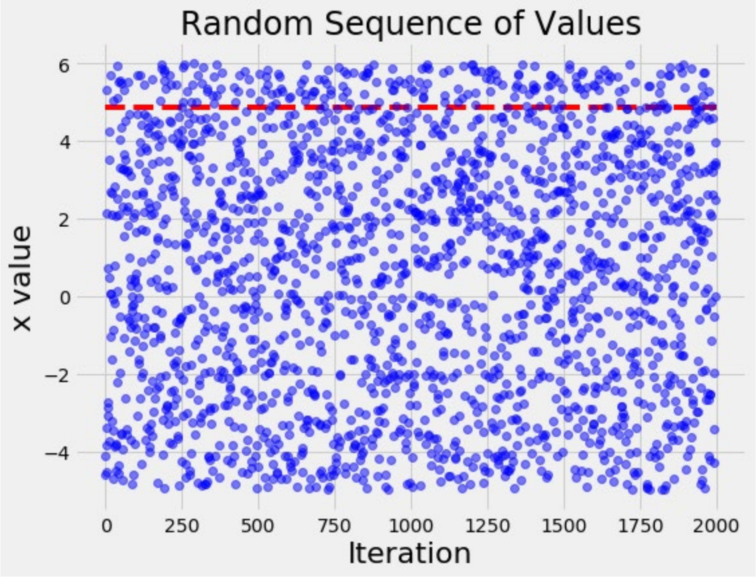
# Simple 1D example



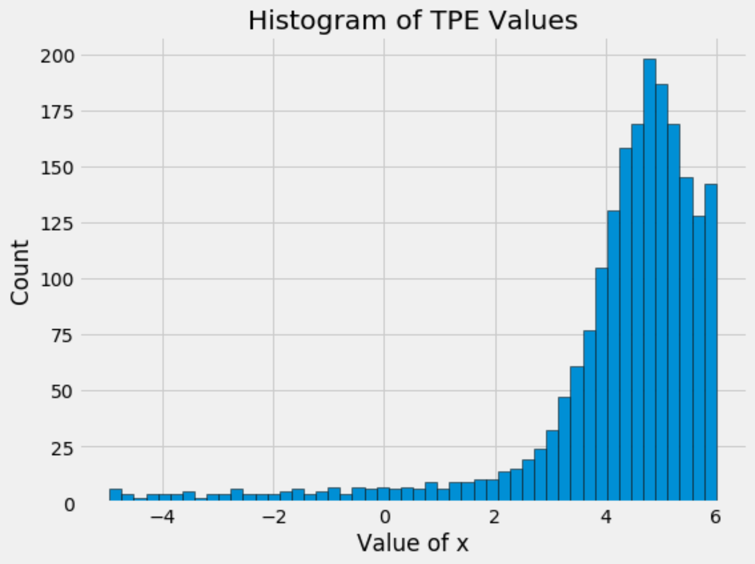


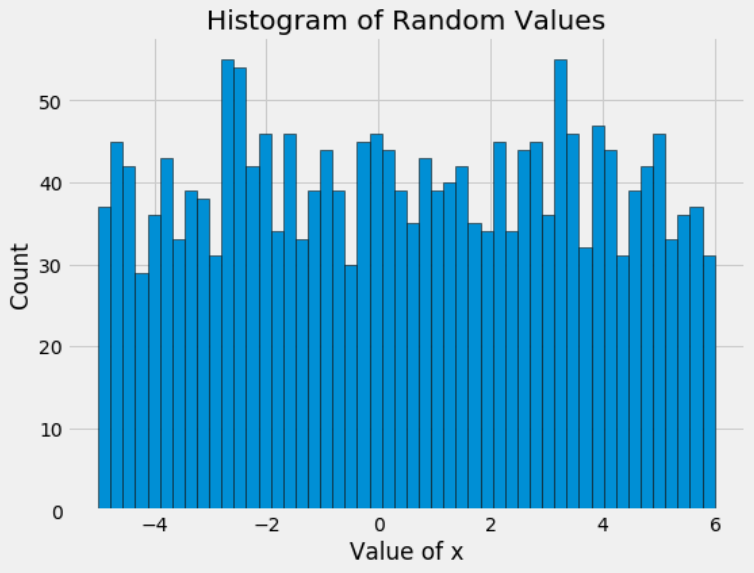
# Compare Bayesian optimizer with random search





# Compare Bayesian optimizer with random search





# Let's try it out