

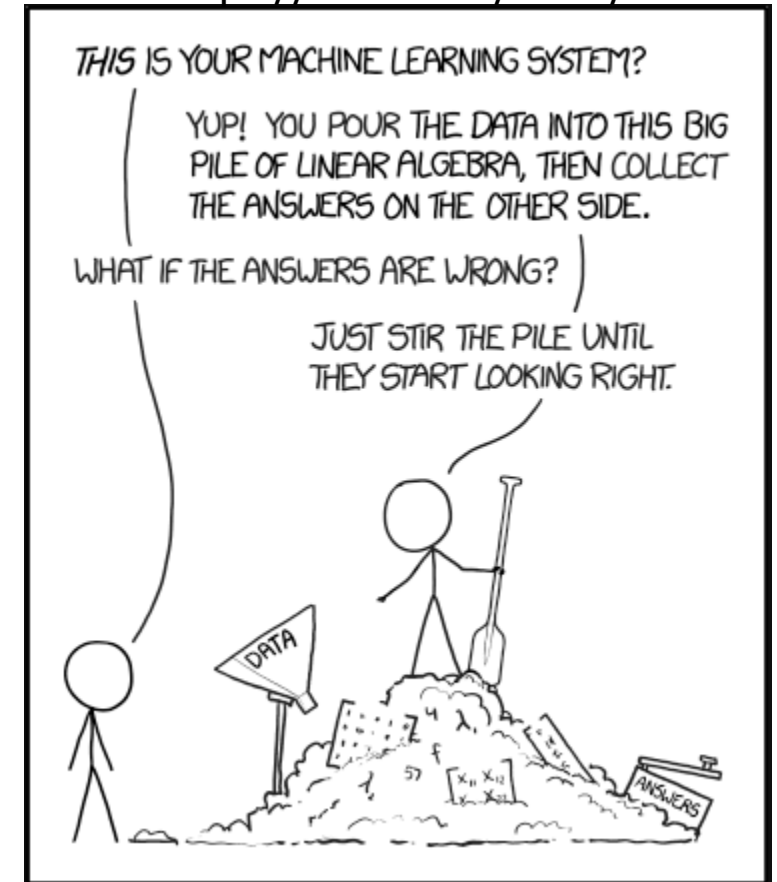
Probabilistic Inference in Machine Learning and Adversarial Prediction Algorithm

Kemal Burçak Kaplan

Why do we need probabilistic methods?

- Not only predict but also quantify confidence
- Use prior knowledge, instruct model with common sense
- Distribution of hypotheses instead of point estimation

<https://xkcd.com/1838/>



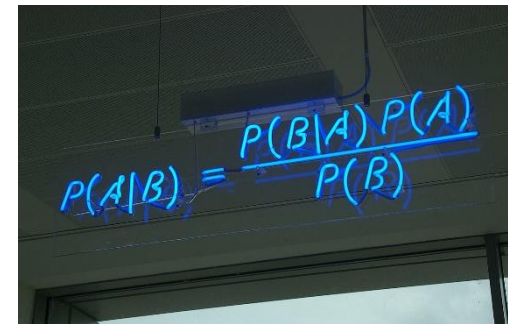
Instruct model which hypotheses are extraordinary

- Using prior knowledge:
 - Given s/he exhibits shy behaviour what is the probability of her/him being Math PhD as opposed to Business Phd?
 - There are only handful of Math PhD students as opposed to dozens of Business PhD students.
- $P(A | B)$ is not equal to $P(B | A)$



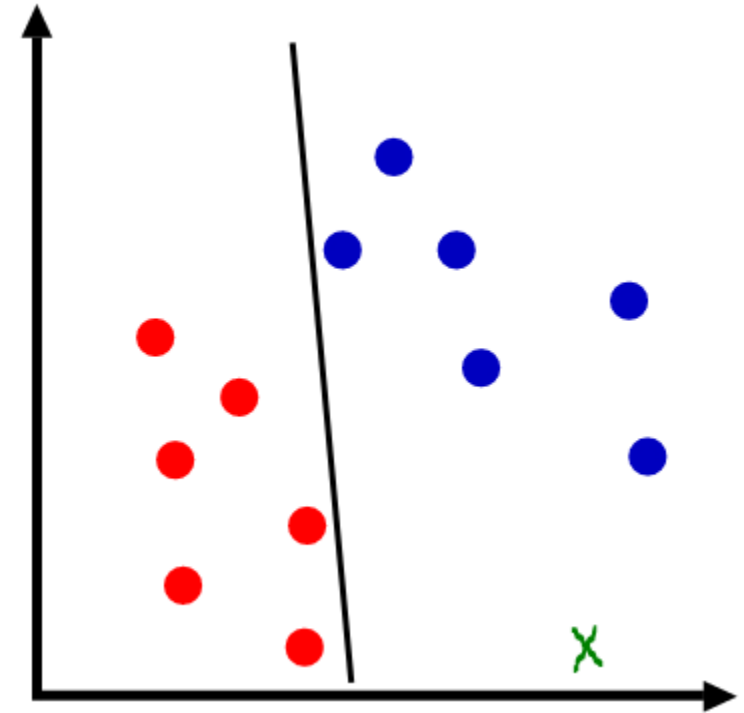
The weight of evidence for an extraordinary claim must be proportioned to its strangeness.

~ Pierre-Simon Laplace

A photograph of a whiteboard with the formula for Bayes' theorem written in blue marker. The formula is
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

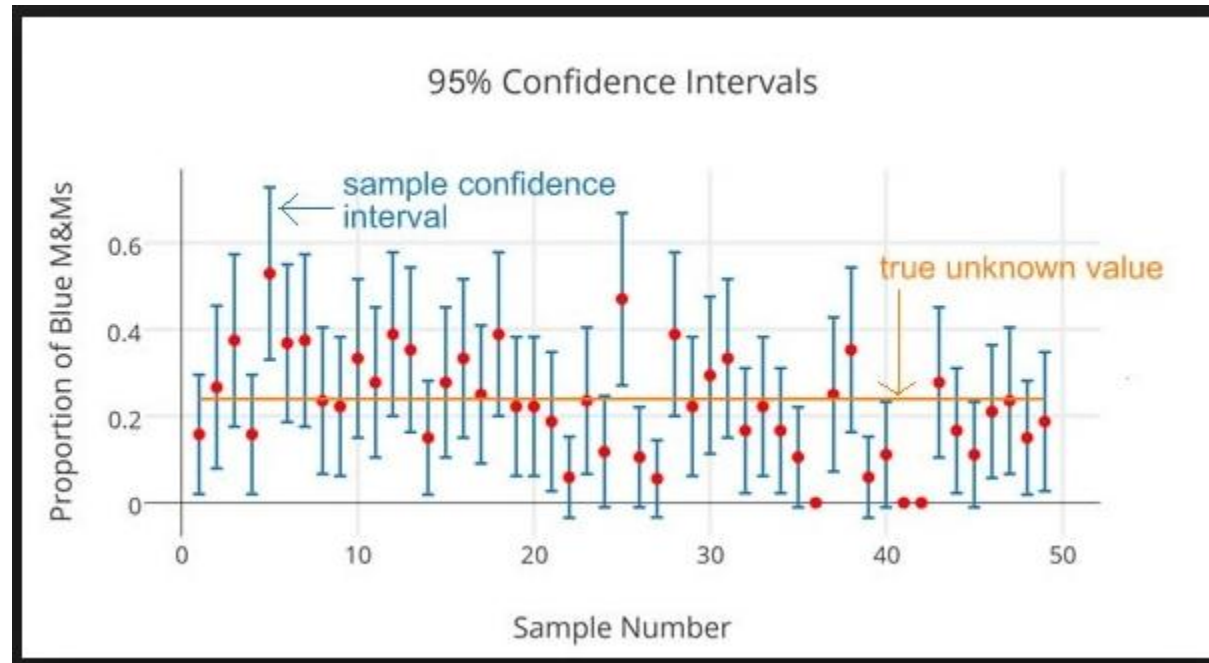
Update based on evidence

- Training awareness
- Relative likelihood is a poor measure of confidence
- Abstain from prediction



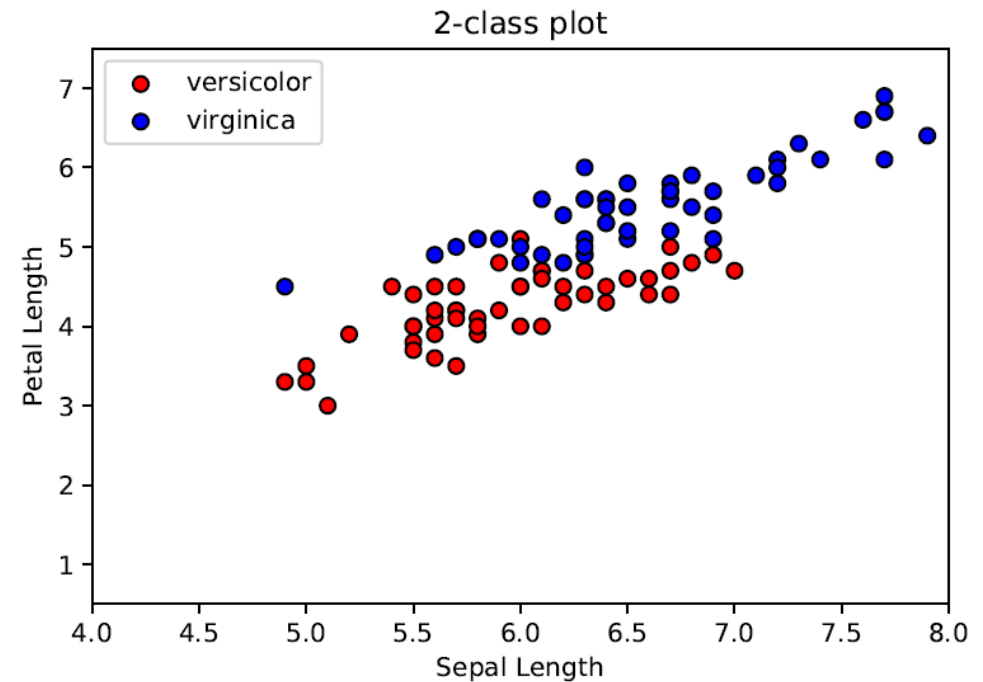
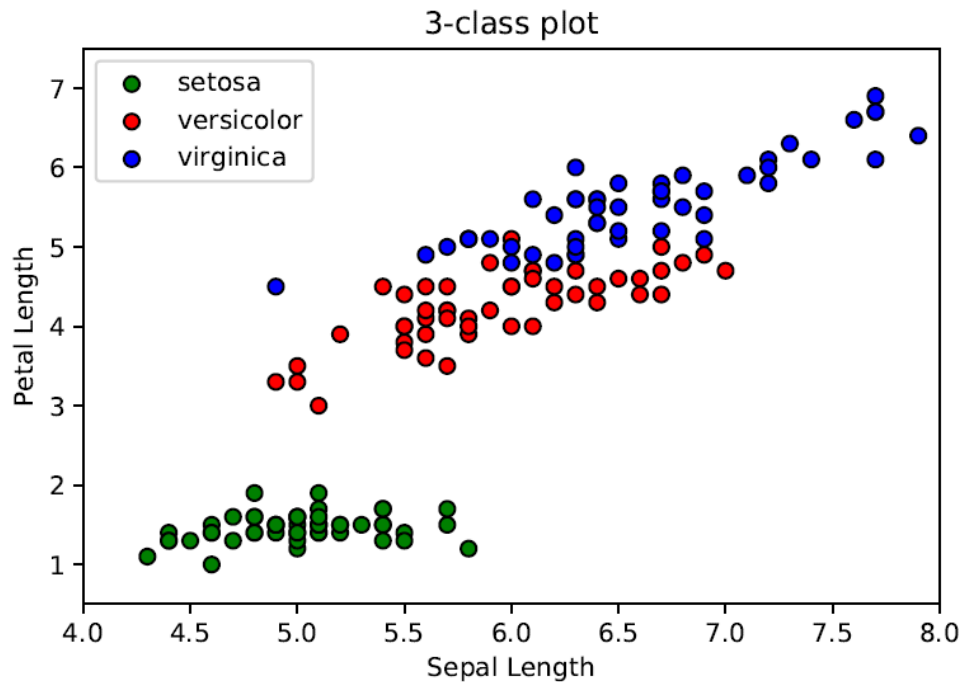
Frequentist MLE is weak measuring uncertainty

- Confidence interval is not probable interval



Application with Iris Flower Dataset

- Setosa sample will deliberately be excluded in the training data set.
- Develop a decision tool that can detect flowers' class based on “petal length” and “sepal length”.



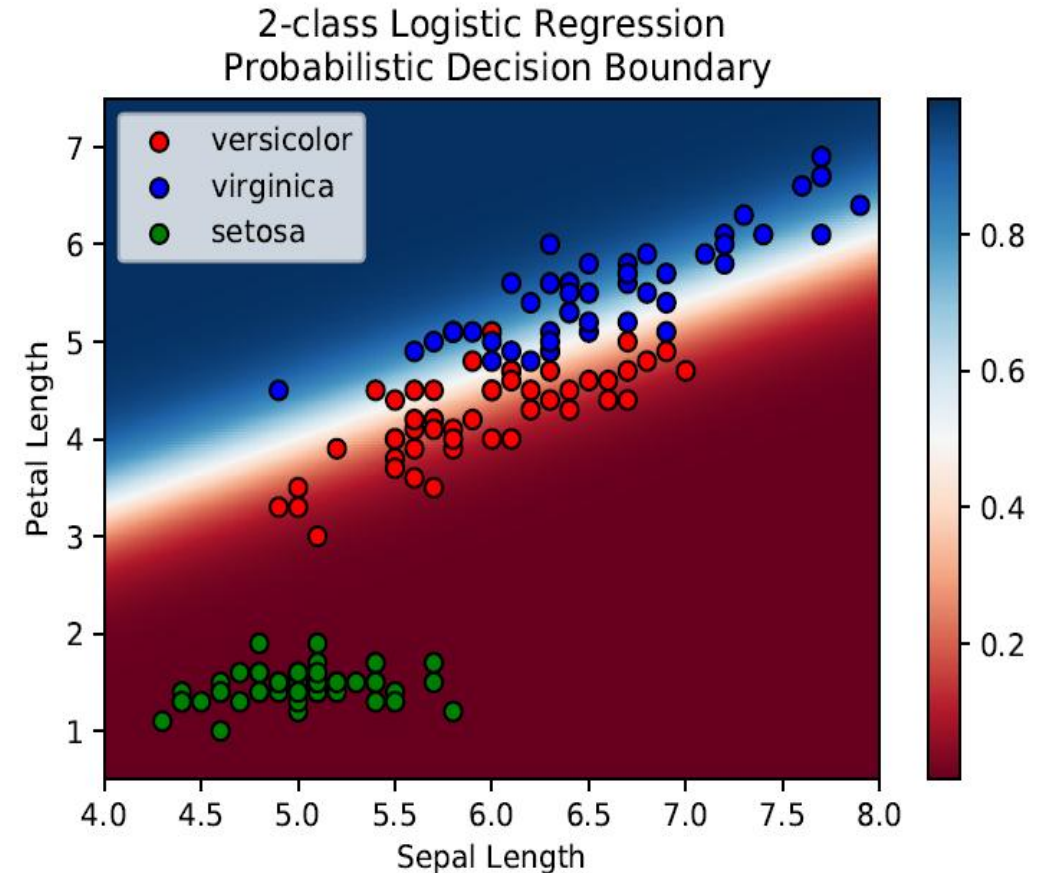
Logistic Regression

- $P(Y = 1 | X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$

- $\operatorname{argmax}_{\beta_0, \beta_1} L(\beta_0, \beta_1) =$

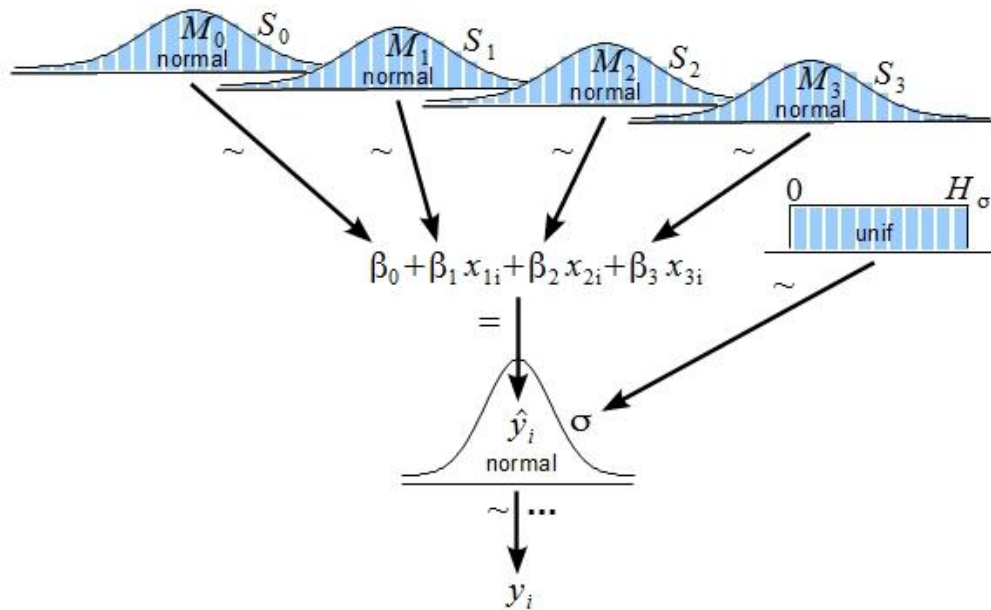
$$\prod_i P(Y = 1 | X_i) \prod_{i'} (1 - P(Y = 1 | X_{i'}))$$

- Overconfidence in un-trained regions
- All setosa samples are predicted as versicolor with 0.99 'probability'
- Relative likelihood is a poor measure of confidence



Bayesian Framework for Probabilistic Classification

1. Parameter distribution estimation using MCMC sampling



2. Non-parametric regression with Gaussian Process

$$K(x, x') = \sigma_0^2 \exp \left[-\frac{1}{2} \left(\frac{x - x'}{\lambda} \right)^2 \right]$$

Intuition: function variables close in input space are highly correlated, whilst those far away are uncorrelated

λ, σ_0 — hyperparameters. λ : lengthscale, σ_0 : amplitude

Gaussian Process Classifier

- $P(f(X_n))$ is jointly Gaussian distribution

$$f \sim GP(m, K)$$

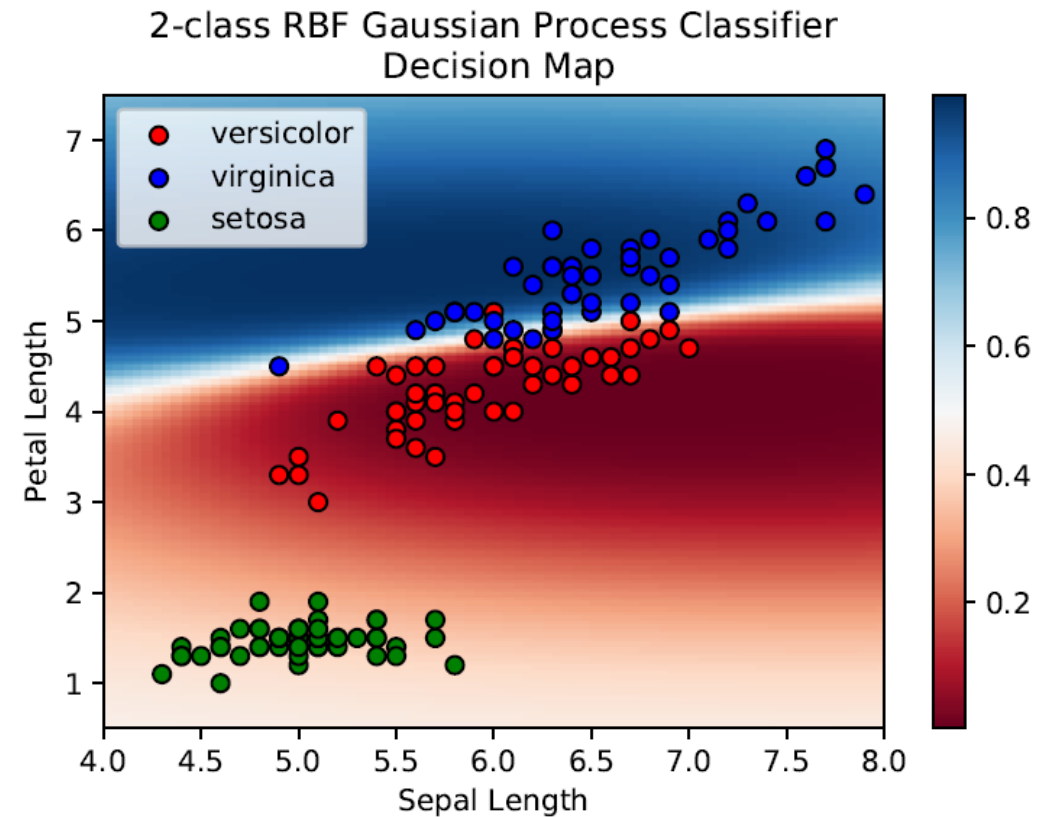
m : mean function

K : kernel function

$$P(f(x_1), \dots, f(x_n)) \sim N(\mu, k)$$

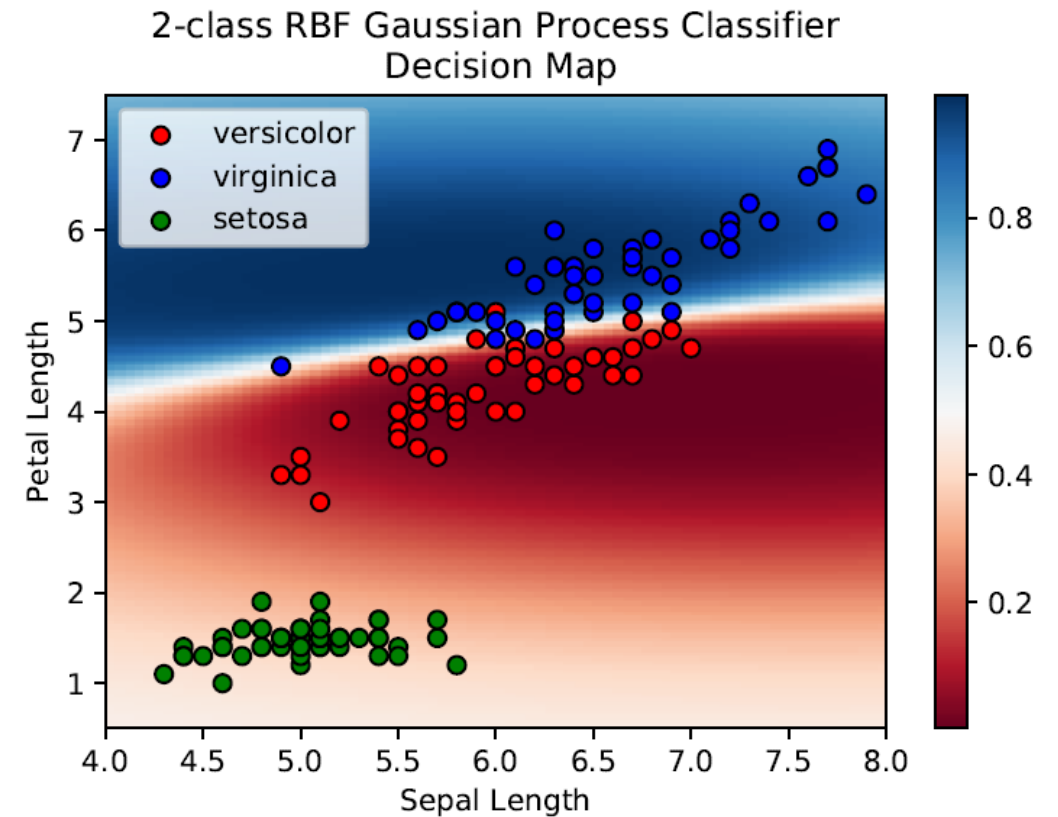
$$\mu = [m(x_1), \dots, m(x_n)]$$

$$k_{ij} = K(x_i, x_j)$$



Gaussian Process Classifier

- Kernel function defines similarity between features
- Model predictions are interpretable as probability
- GPC predicts setosa as versicolor but with probability 0.61



Adversarial Prediction Algorithm

- Frequentist estimations have limited inference capacity
- Gaussian processes have more intuitive approach to uncertainty quantification however they are not always easy to design and scale.
- Therefore non-probabilistic classification algorithms, not only linear ones but tree-based additive algorithms, are more commonly used in practice.
- Adversarial Prediction Algorithm is a two-stage algorithm that extends models with confidence metrics

Adversarial Prediction Algorithm

Train a Classifier based on the test sample

For Each Prediction Sample:

- Predict the class of the sample

- Sample A: Create N random sample of the predicted class using the training sample

- Sample B: Create N random sample from chosen distribution with the sample's features as the mean

- Train new (Adversarial) Classifier with Sample A vs Sample B

- Confidence: Use inverse accuracy as the confidence of the prediction

Adversarial Prediction Algorithm

- The underlying logic is that if the prediction sample resembles the predicted class enough, second stage classifier should have very low accuracy.
- It would show that adversarial sample is indistinguishable from predicted class. Therefore inverse accuracy of the second model can be used as the confidence metrics of stage 1 prediction.
- The idea is partially analogous to Gaussian Process in the sense that each observation is assumed to be distributed with normal distribution. If new sample were similar enough to the predicted class, we should not be able to distinguish the two sample.

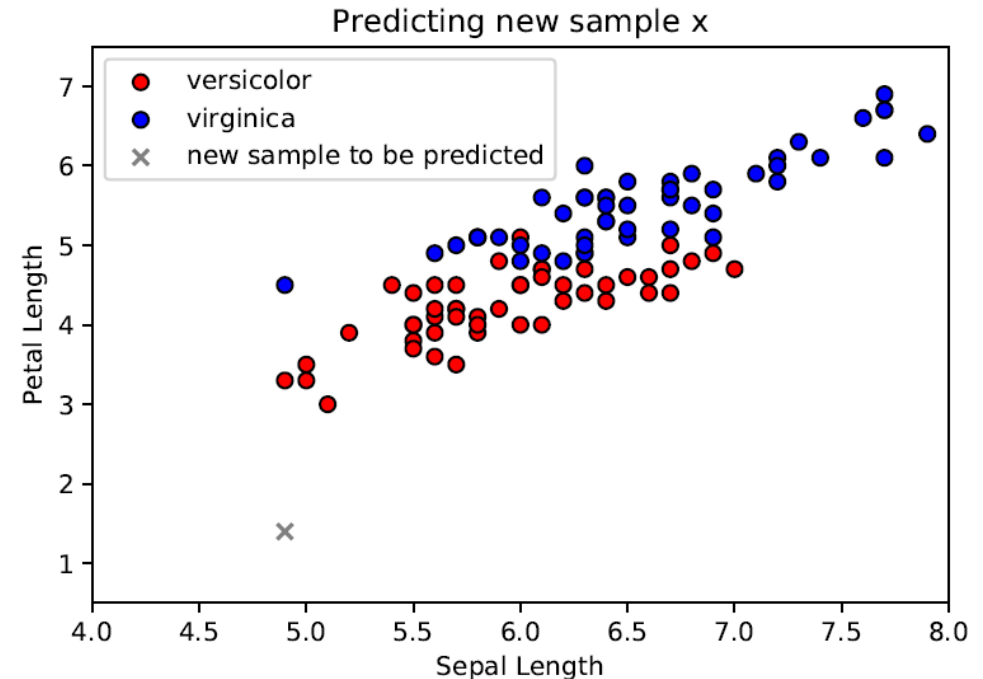
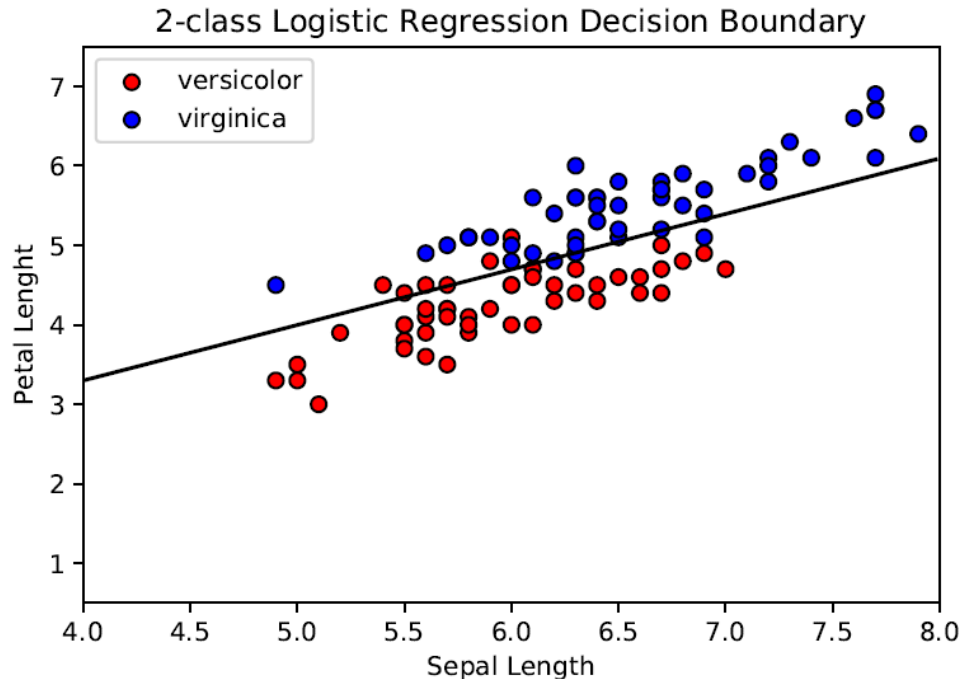
AP Algorithm: Python implementation

- Similar interface to scikit-learn classifiers
 - Fit and predict_conf as main methods

```
from AdversarialPredictor import AdversarialPredictorClassifier
m_apc = AdversarialPredictionClassifier(base_estimator = LogisticRegression())
m_apc.fit(X_train, y_train)
m_apc.predict_conf(X_test, adv_estimator = LogisticRegression())
>>[[[0.9015, 0.0984], 0.5800],
    [[0.8199, 0.1800], 0.8400],
    [[0.7946, 0.2053], 0.5400],
    ..]
```

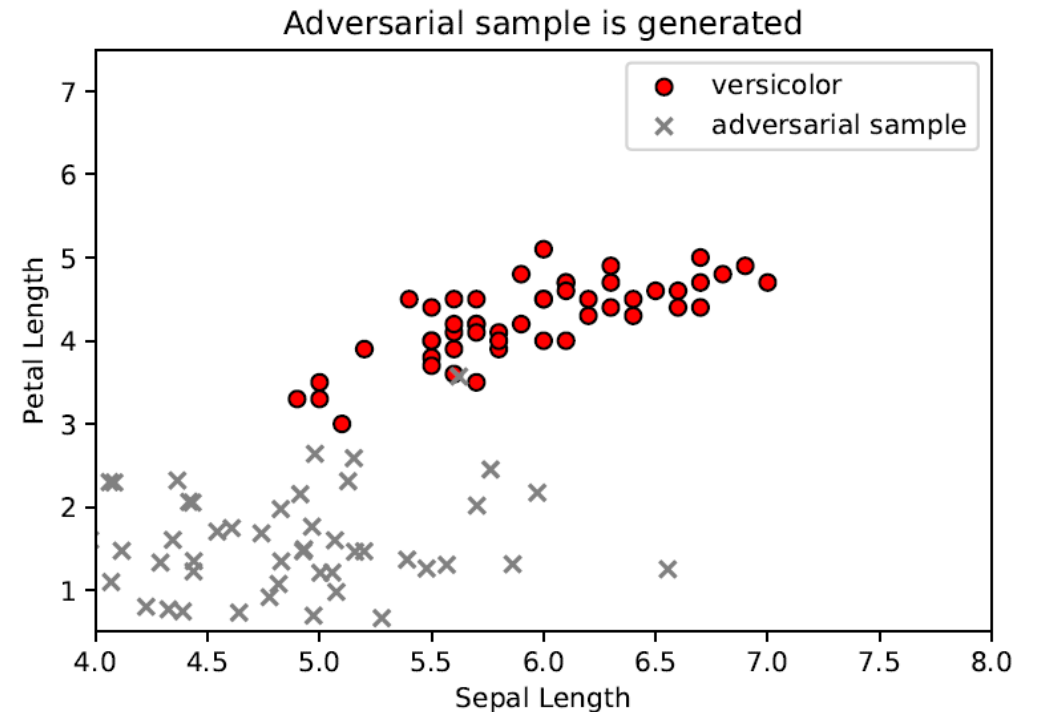
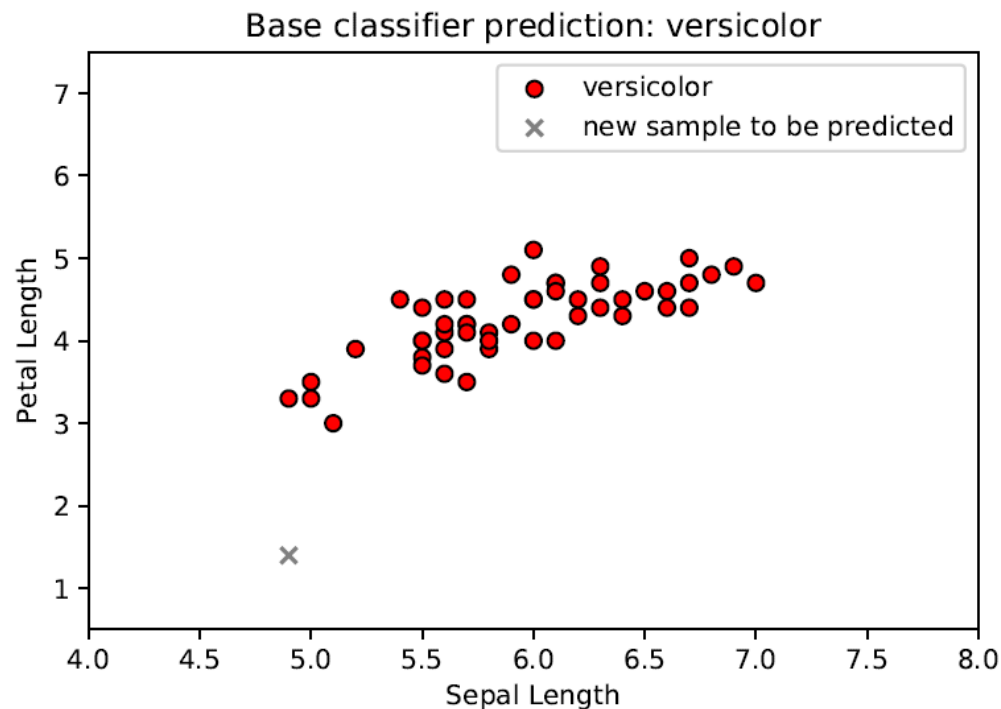
AP Algorithm: Python implementation

- At the first stage, we achieve 0.90 accuracy with logistic regression
- New sample pointed as “x” below will be predicted using base classifier. It belongs to an untrained region in the feature space.



AP Algorithm: Python implementation

- When we use our base model to predict this new sample, the model predicts versicolor with logit score of 0.99. We generate an adversarial sample from normal distribution with mean as the sample and standard deviation of predicted versicolor sample.

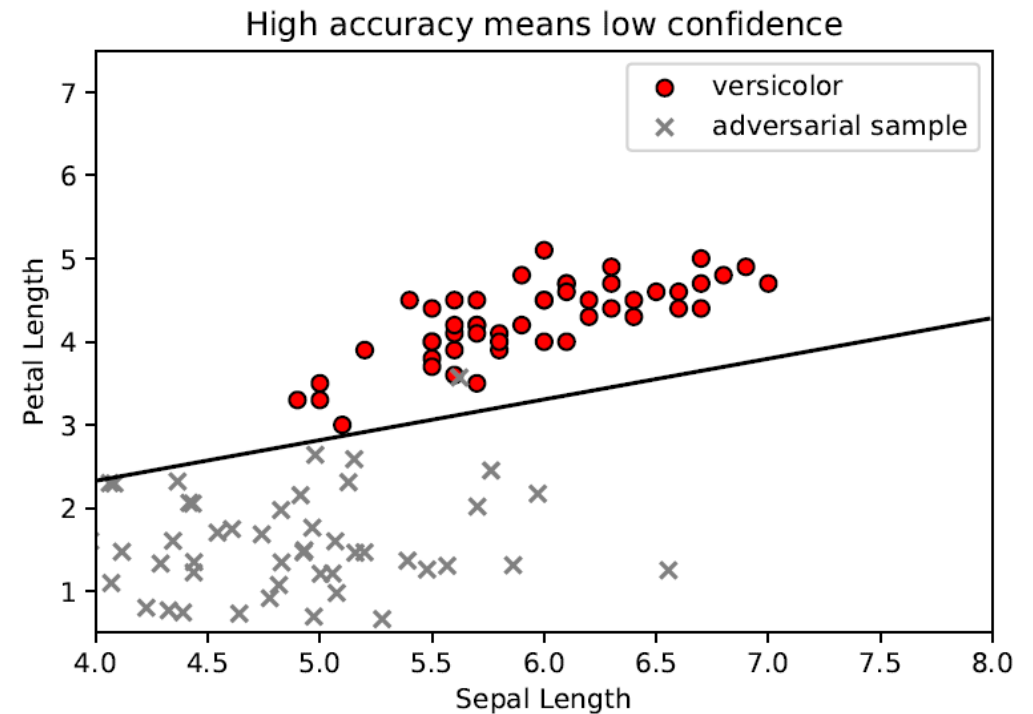


AP Algorithm: Python implementation

- The accuracy of the second -stage classifier is very high, therefore our confidence estimate for the 1st-stage prediction is low.

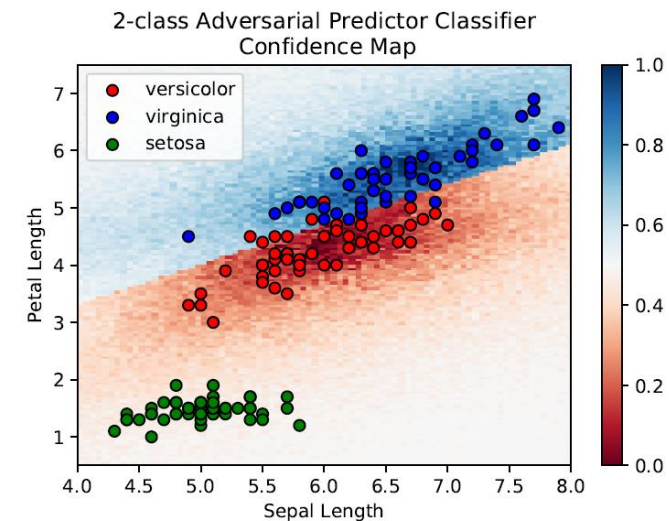
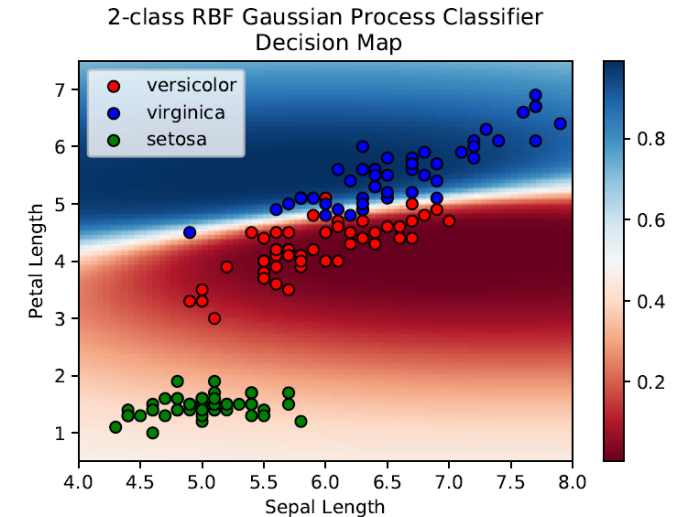
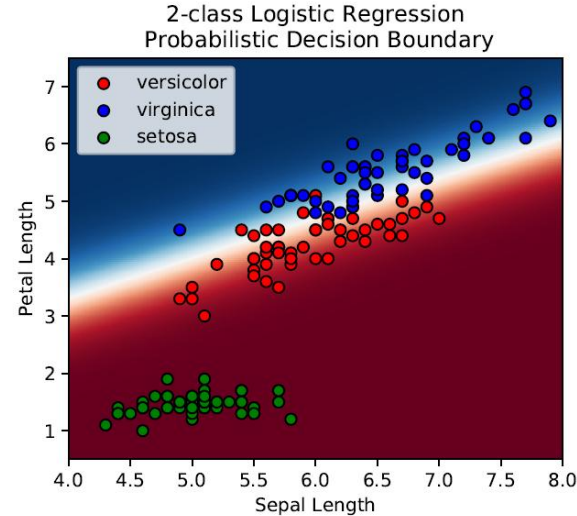
$$\frac{(1 - \text{accuracy})}{0.5}$$

- We can argue that although the new sample is more similar to versicolor compared to virginica, it is also very likely to be a new class that was not in our original training data set.



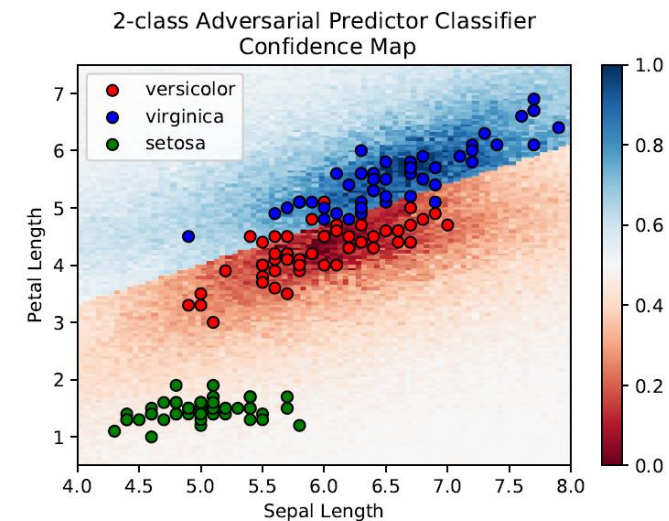
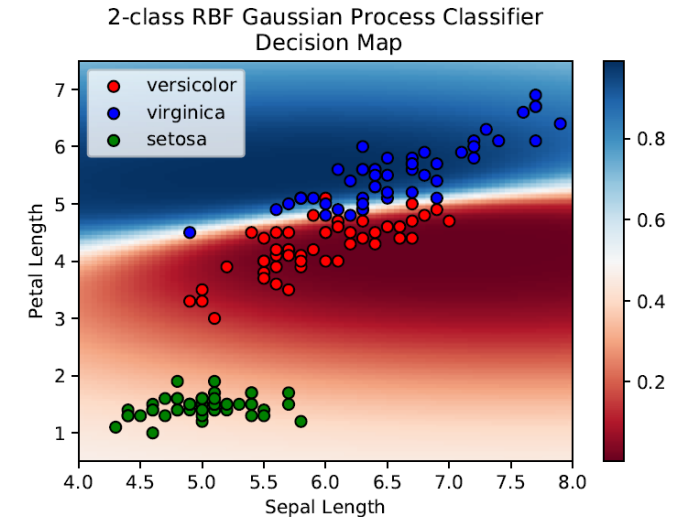
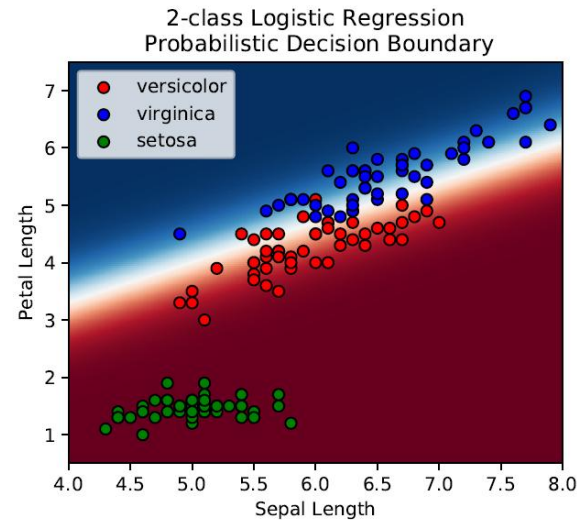
Concluding Remarks

- Bayesian methods make use of calculus of probability to reason about uncertainty.
- However empirical error minimization methods are easier to model and to scale.



Concluding Remarks

- Adversarial Prediction algorithm is an extension for predictions from non-probabilistic regression to calculate confidence metrics.
- As an area of improvement, sampling method of adversarial sample should be mentioned.
- In this study, normal distribution with uncorrelated features is used however endogenous methods with kernels can be developed.



Concluding Remarks

- Having a dependable confidence metrics is an important part in decision making processes, an intelligent decision system should also be able
 - to abstain from making prediction
 - to detect a novel species that it has not yet been trained about.

