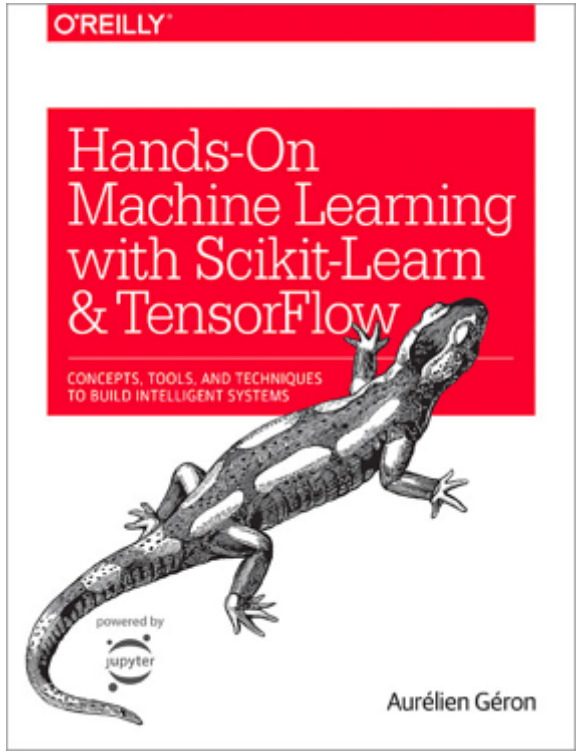
Machine Learning (Classification)

WILLIAM STEIMEL
スタイメルウィリアム

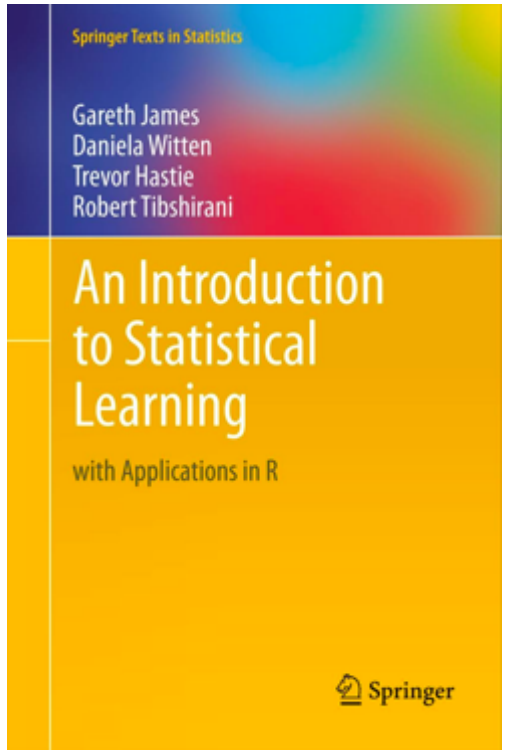
Table of Contents

- Introduction
- Classification Performance Measures
- Logistic Regression
- Linear Discriminant Analysis (LDA)
- K-Nearest Neighbors

Sources



Chapter 3- Classification



Chapter 4- Classification

Introduction

Introduction

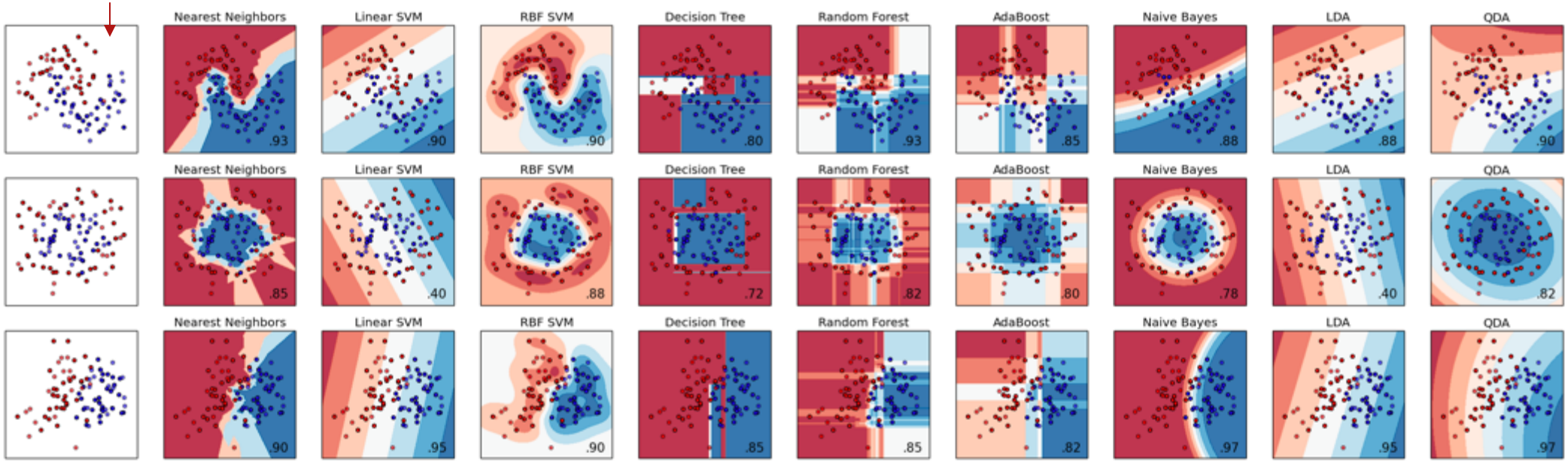
- What is Classification?
- Classifier Comparison
- Binary Classification / Multiclass Classification
- What Problems does Classification Solve?
- Classification Sample Datasets (Kaggle)

What is Classification?

- Supervised Learning　（教師あり学習）
  - (Labeled Data)
- Prediction of Qualitative Values (Categorical)
  - Ex 1: Predict whether an E-mail is Spam or not (Yes or no)
  - Ex 2: Predict whether a passenger will survive the Titanic (Yes or no) – Titanic Dataset
  - Ex 3: Predict what type flower an Iris is - Iris Dataset
- Common Classifiers
  - Logistic Regression
  - Linear Discriminant Analysis
  - K-Nearest Neighbors

Classifier Comparison

3 Datasets

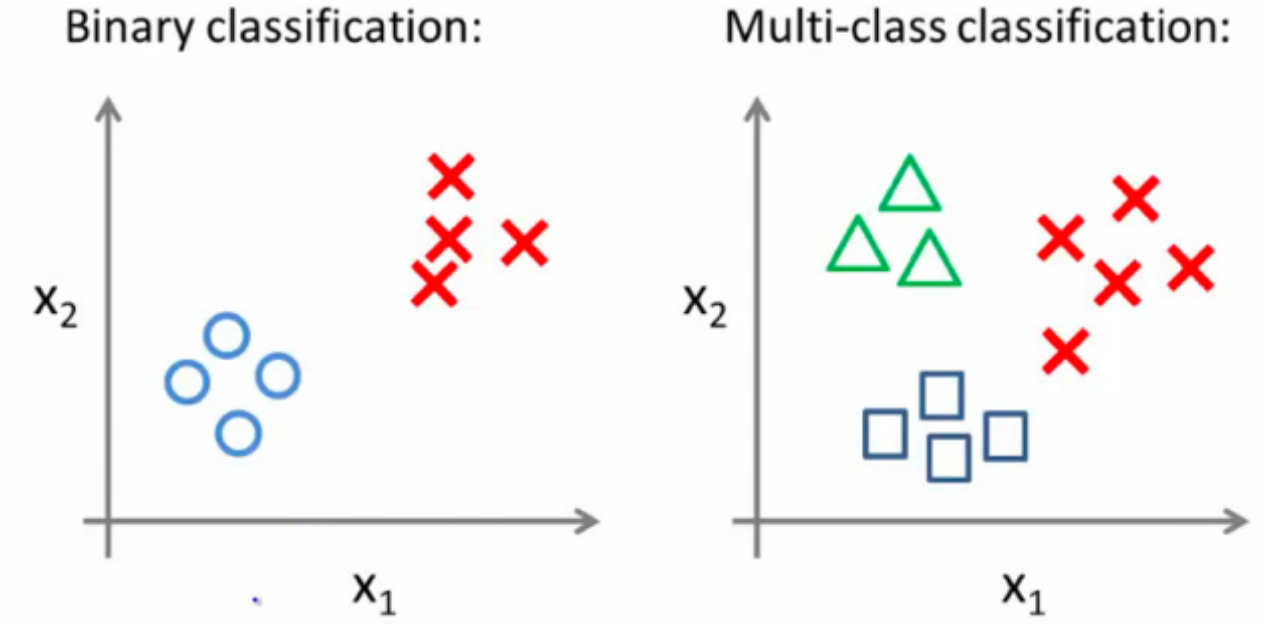


Source: Scikit Learn

- Decision boundary – (Based on color)
  - The Line or Boundary where the feature space is split for classification
- Classification Accuracy- (Bottom Right)
  - Performance Measure

Binary Classification / Multiclass Classification

- Binary Classification – Distinguishes between two classes (O or X)
- Multiclass Classification- Distinguishes between more than two classes (Triangle, X, Square)



What Problems does Classification solve in the real world?

- Anything Related to Class Prediction

Medical Diagnosis

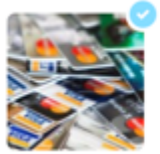
Fraud Detection

Digit Recognition

Spam Detection

Classification Sample Datasets

- Where can you practice Classification Problems?
- Kaggle is a platform for Data Science competitions:
  - Examples from Kaggle Datasets:



Credit Card Fraud Detection

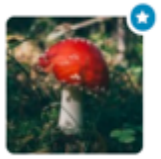
Anonymized credit card transactions labeled as fraudulent or genuine

Andrea updated a year ago

Iris Species

Classify iris plants into three species in this classic dataset

UCI Machine Learning updated a year ago



Mushroom Classification

Safe to eat or deadly poison?

UCI Machine Learning updated a year ago

kaggle

Performance Measures

Performance Measures

- What are Performance Measures?
- Confusion Matrix
- Accuracy Rate
- Precision/Recall
- F1 Score
- Precision/Recall Tradeoff
- ROC Curve
- Conclusion

What are Performance Measures?

- Performance Measures are used to evaluate all Learning Algorithms ability to predict. (Evaluation component)
- This section will discuss Performance Measures used for Classification Algorithms including the Confusion Matrix, Accuracy rate, Precision/Recall, F1 score, ROC Curve, and the AUC.

Table 1: The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Source: A Few Useful Things to Know about ML

Dataset Information

- This section uses a dataset from MNIST and an example from the Hands-On Machine Learning with Scikit-Learn & Tensor Flow book to illustrate classification performance measures.
- "The **MNIST database** (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems." - Wikipedia
- For this section, The problem is a Binary Classification Problem
  - Classify digit as either 5 or not 5

Training a Binary Classifier

In [26]:
```
## Simplify Problem to Binary Classification  (5 or not 5)
y_train_5 = (y_train ==5) # True for all 5s, False for all other digits
y_test_5 = (y_test == 5)
```

In [27]:
```
# Stochastic Gradient Descent
from sklearn.linear_model import SGDClassifier
sgd_clf = SGDClassifier(random_state =42)
sgd_clf.fit(X_train, y_train_5)
```

Confusion Matrix

A Confusion Matrix is used to represent predictions vs actual values.

In this example we have:
54,090 True Negatives
3965 True Positives
1456 False Negatives
489 False Positives

		Prediction	
		Not 5	5
True State	Not 5	TN	FP
	5	FN	TP

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_train_5, y_train_pred)
```

```
array([[54090,   489],
       [ 1456,  3965]])
```

Perfect Confusion Matrix
(True Positives and True Negatives)

```
array([[54579,     0],
       [    0,  5421]])
```

Accuracy Rate

- Accuracy Rate is very easy to calculate.
  - Correct Predictions / Total Predictions
- Accuracy Rate is however not always the best classifier evaluator especially on datasets with skewed distribution or large class imbalance

```
In [51]: accuracy_score(y_train_5, y_train_pred)
Out[51]: 0.96758333333333335
```

Precision/Recall

$$\text{Precision} = \frac{TP}{TP + FP}$$

```
precision_score(y_train_5, y_train_pred)
```

```
0.89021104625056124
```

```
3965/ (3965+489)
```

```
0.8902110462505612
```

The Accuracy of the Positive Predictions

$$\text{Recall} = \frac{TP}{TP + FN}$$

```
recall_score(y_train_5, y_train_pred)
```

```
0.73141486810551559
```

```
3965/(3965+1456)
```

```
0.7314148681055156
```

Also known as sensitivity/True Positive Rate

		Prediction	
		Not 5	5
True State	Not 5	TN	FP
	5	FN	TP

F1 Score

- F1 Score is a combination of the Precision and Recall Metrics (The Harmonic Mean)
- The F1 Score favors classifiers that have similar values for Precision and Recall (Balanced)
  - This may not always be desired though as there are situations where higher Precision or Higher Recall may be preferred.

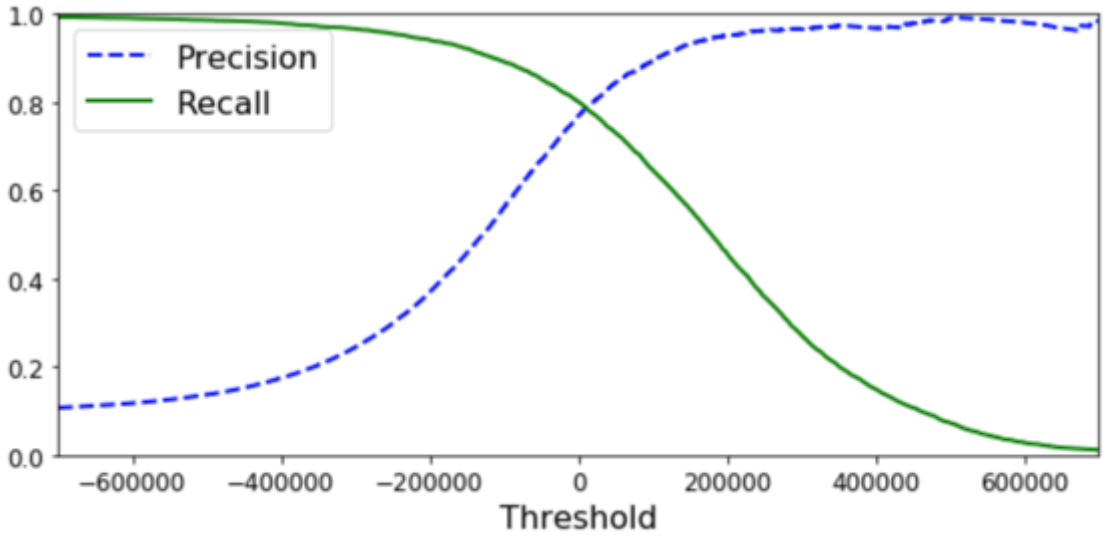
$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

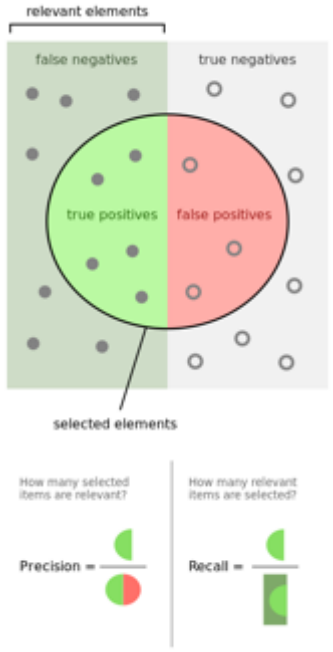
```
from sklearn.metrics import f1_score
f1_score(y_train_5, y_train_pred)
```

```
0.80303797468354432
```

Precision/Recall Tradeoff

- Increasing Precision reduces Recall/ Increased Recall reduces Precision
  - This is called the Precision/Recall Tradeoff

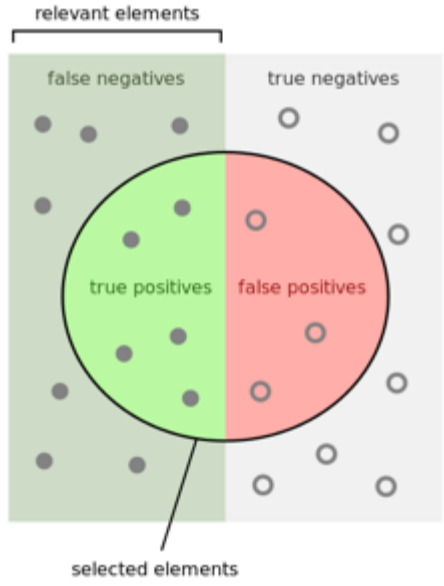


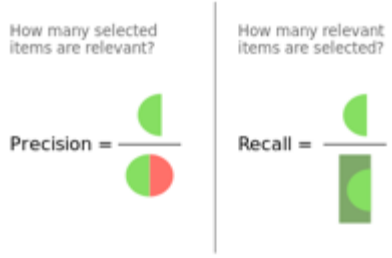


Precision/Recall Tradeoff

- In some situations you will prefer Higher Precision and sometimes Higher Recall. This is dependent on the type of problem that needs to be solved.
- Classifier 1: Classifier that predicts whether a video is safe for children
  - A classifier that rejects good videos (low recall) but keeps only safe ones (high precision) is likely preferred.
  - Since Precision is a measure of relevant items it will return more safe videos.

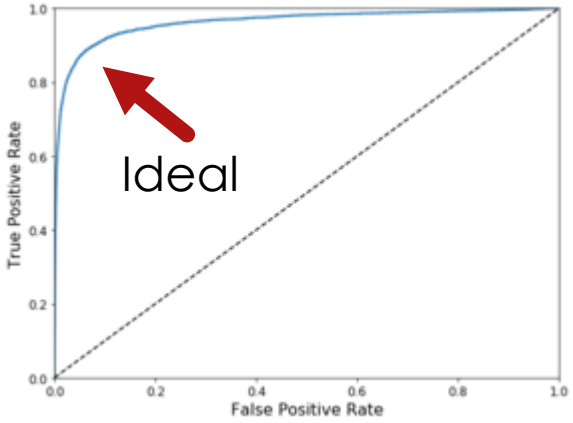
- Classifier 2: Classifier that detect shoplifters on surveillance images (Stealing)
  - A classifier that has more False Alerts (higher recall) and less accuracy (lower precision) is likely preferred.
  - Although the classifier will make more mistakes more shoplifters will get caught
    - This may also be preferred for Disease detection (It is better to have a False Negative than to not detect)





ROC Curve

- The ROC Curve (Receiver Operating Characteristic) is also commonly used to evaluate binary classifiers
  - Sensitivity(Recall- TPR) vs specificity (FPR)
- It is ideal for the TPR to be high and the FPR to be low
  - As shown below, ROC Curve also has tradeoff between TPR and FPR
- Can compare different classifiers by plotting their ROC Curves and comparing
  - There is also a measure called AUC – which means Area Under Curve:
    - Higher is better as it indicates more space under the ROC curve



Conclusion

- There are many different ways to evaluate a classifiers performance and measures used will differ based on the ML problem
- Evaluating performance is important as it allows you to determine which classifier is best at solving the classification problem.
- ROC Curve vs Precision/Recall Curve (Depends on Problem)
  - PR Curve should be used when the positive class is rare or you care more about false positives than false negatives

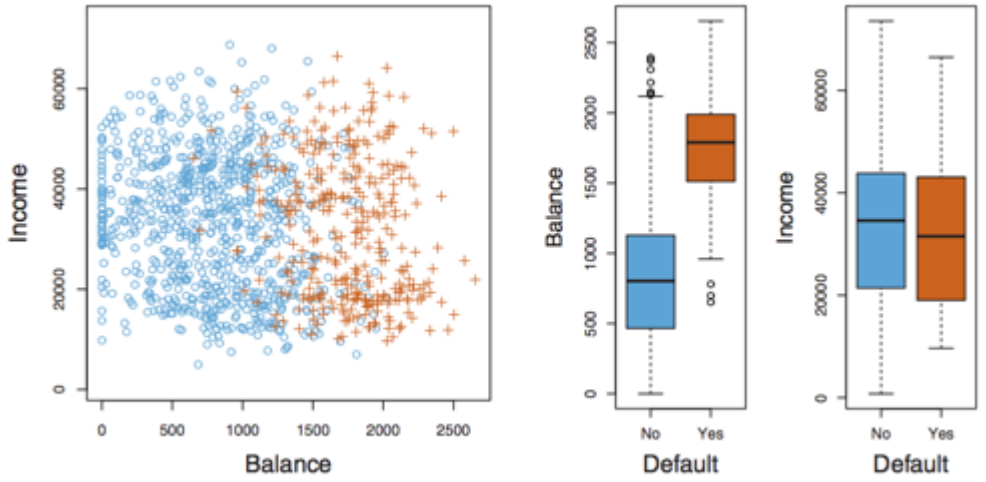
Logistic Regression

Logistic Regression

- Dataset Information
- Logistic Regression
- Logistic Function
- Likelihood Function
- Making Predictions
- Multiple Logistic Regression
- Pros/Cons to Logistic Regression

Dataset Information

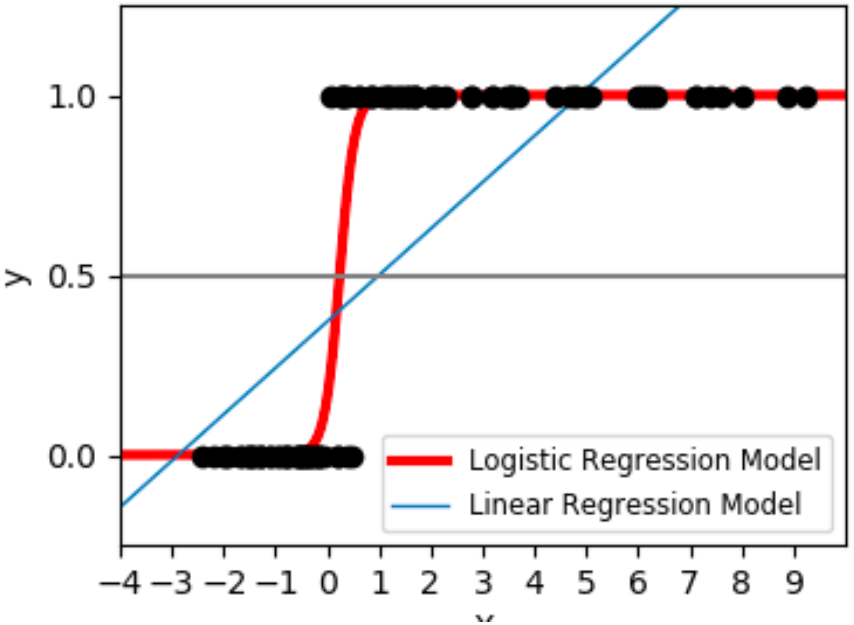
- This section uses the Default Dataset which details individuals who defaulted on credit card payments based on a number of factors.
- Many of the graphs in the next section are from an Introduction to Statistical Learning



Logistic Regression

- Linear Regression vs Logistic Regression (Right side)
- Can clearly see that the linear function does not model this Classification problem well.
- This is where Logistic Regression comes in !
- Logistic Regression models the probability that y belongs to a particular category/class
- Example:
  - 0 = No Default on Loan
  - 1= Default on Loan

Probability
確率



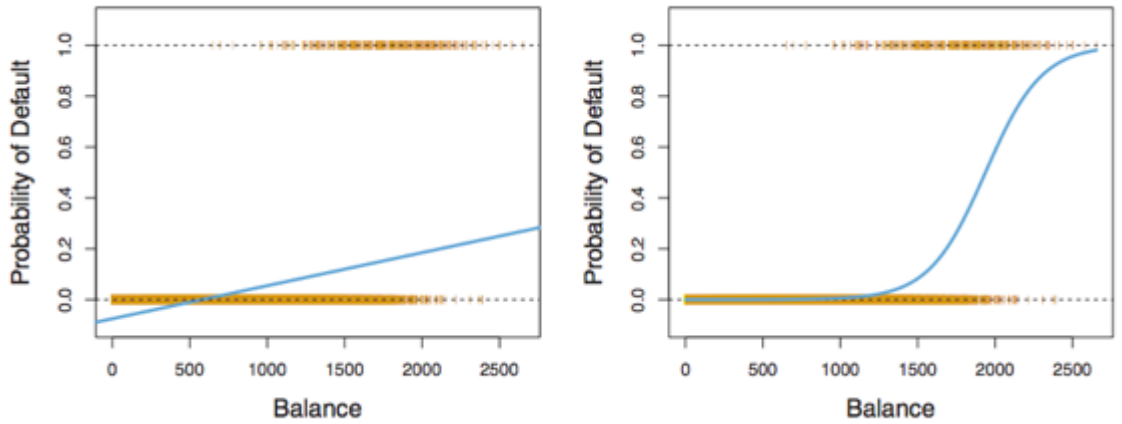
Source: Scikit Learn

(Sigmoid) Logistic Function

- e Scientific Constant, exponential value
  - e is approximately 2.71828 [Euler's Number]
- Beta 0 – Y- Intercept
- Beta 1 – Coefficient of X
- P(X) will be between 0 and 1 (S-Shape)
- Variation of the Linear Model but modeling probabilities on a non linear scale
- Generally, Classification is done based on the probability
  - Probability > .5 Classify to Default on Loan
  - Probability < .5 Classify to no Default

Logistic Function:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$



Not a very good fit !

Likelihood Function

- The Maximum Likelihood or Likelihood function is often used to Estimate the Regression Coefficients in Logistic Regression
  - (Gradient Descent is sometimes used)
- Maximum Likelihood can be described as a Minimization algorithm used to best optimize the coefficients

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

- This Formula returns the observed probabilities of 0 and 1

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.6513	0.3612	−29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

Making Predictions

- Once the coefficients have been estimated you can make predictions based on them
- The below formula makes a prediction based on a person with 1,000 dollars in credit card debt.
- .005 % of Default

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.6513	0.3612	−29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

Plug in the values !

Logistic Function:

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1{,}000}}{1+e^{-10.6513+0.0055\times 1{,}000}} = 0.00576,$$

Multiple Logistic Regression

- Multiple Logistic Regression uses a similar formula with the addition of other coefficients
- The book then estimates a model with 3 features- Balance, Income, and Student (Y or N)

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.8690	0.4923	−22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	−0.6468	0.2362	−2.74	0.0062

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

$$\hat{p}(X) = \frac{e^{-10.869+0.00574\times 1{,}500+0.003\times 40-0.6468\times 0}}{1+e^{-10.869+0.00574\times 1{,}500+0.003\times 40-0.6468\times 0}} = 0.105.$$

Given a student has a Debt Balance of $ 1,500, Income of 40,000, and not a student: Estimated rate of default is 10.5 % according to this model.

Pros and Cons to Logistic Regression

Pro

- Probabilistic Approach (Gives you the significance
- Simple to Interpret/Linear
- No Parameter Tuning Needed

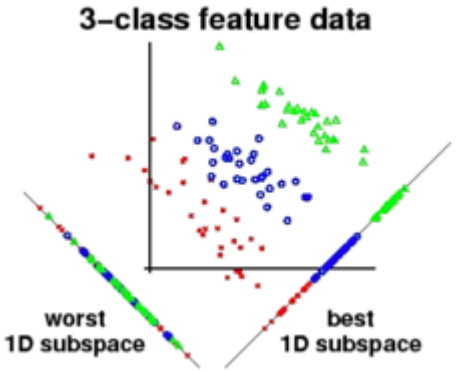
Con

- Assumption of Linearity (Linear Decision Boundary)
- Impacted by Collinearity
- Typically requires a large sample size

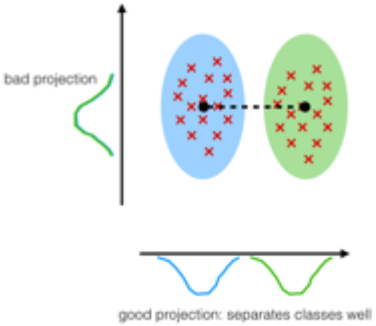
Linear Discriminant Analysis (LDA)

What is Linear Discriminant Analysis?

- Created by Ronald Fisher in 1936 (The Use of Multiple Measurements in Taxonomic Problems)
- "The goal of LDA is to project a feature space onto a small subspace while maintaining the class-discriminatory information."
- Model the distribution of X in each of the classes separately and then utilizes Bayes Theorem to get the PR(Y | X)
- Utilizes Gaussian Distributions
- Commonly used for dimensionality reduction but can also be used as a classifier



LDA:
maximizing the component axes for class-separation



Bayes Theorem

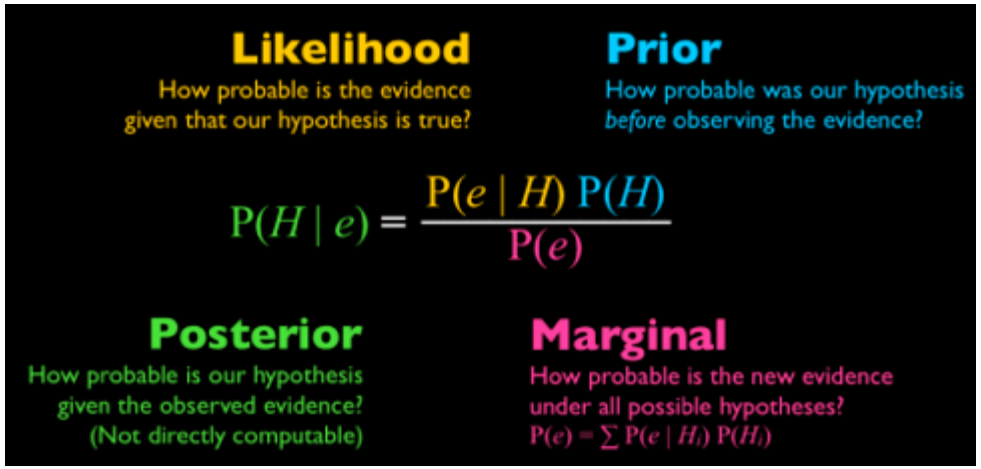
- Foundation for Discriminant Analysis:
- Generally written in this format:

$$\Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

- **Written in this format for discriminant analysis:**

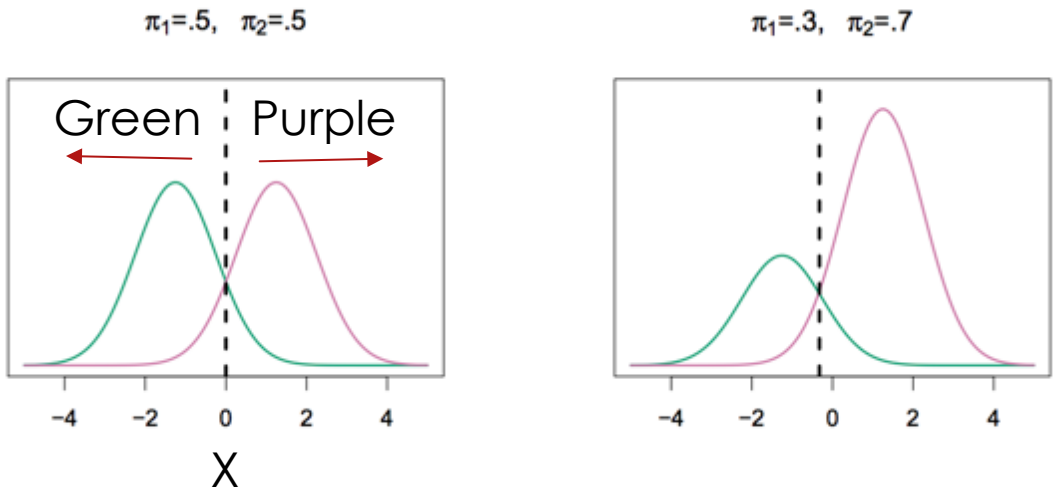
$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- fk(x) = Pr(X = x | Y = k) – Density for x in class k (Probability Density)
- πk = Pr(Y = k) - Marginal or prior probability of Class K
- Bottom Line – Summing over all the classes



Classification of Data

- How is data classified with LDA?
  - Data is generally classified to which density is highest.
- Two Distributions modelled below based on probability



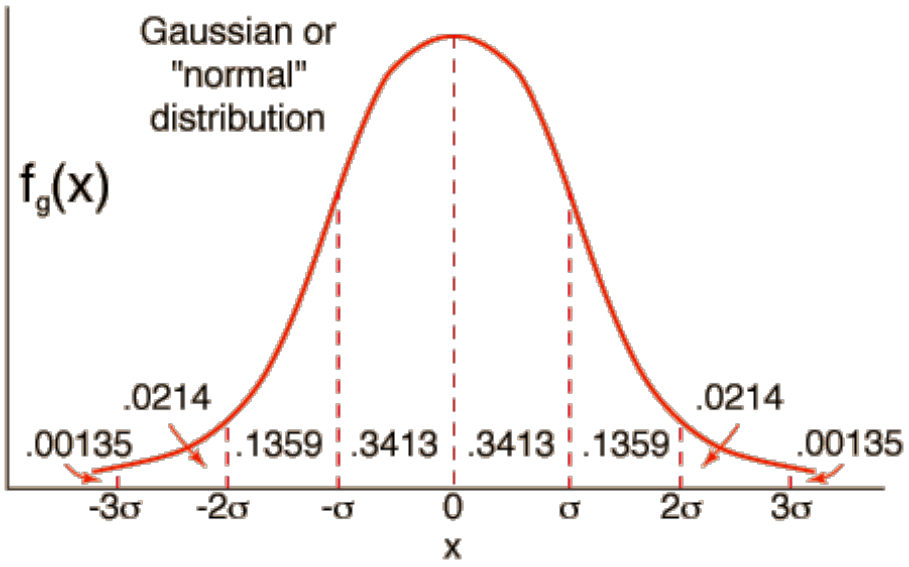
Gaussian Density

- How are the distributions formulated?
- Below is the Formula For Gaussian Density or Gaussian Function
  - (Bell Curve/Normal Distribution)

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

μ_k Mean/average of class k

σ_k Standard Deviation of class k



Discriminant Analysis for 1 predictor

Gaussian Density

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

Bayes Theorem
(Discriminant Analysis form)

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

After Plugging in
the Gaussian Density
to Bayes Formula

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma}\right)^2}}$$

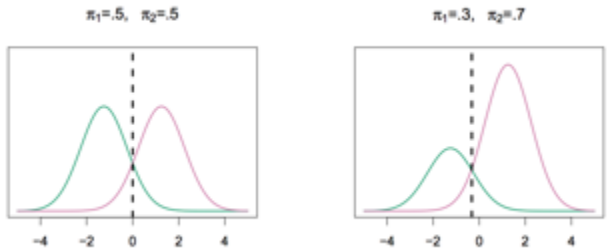
This can be simplified further but this will give us the probability of class K given the x values

Class Prediction

- The previous formula further simplifies to the below formula which is also called the discriminant score:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

- This formula calculates the discriminant score of all classes (k) based on x
  - The formula assigns the observation to the class with the highest discriminant score.

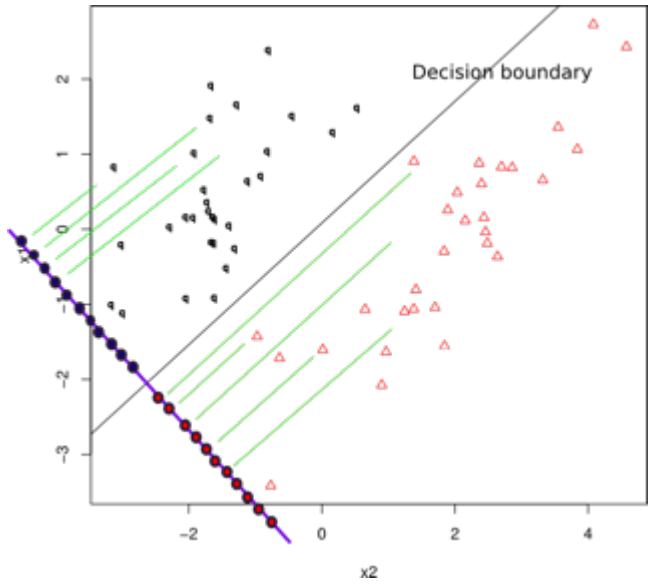


LDA For More than one predictor

Density Function $$f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)}$$

Discriminant Score $$\delta_k(x) = x^T \mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1}\mu_k + \log \pi_k$$

Class Probability Estimate: $$\widehat{\Pr}(Y = k|X = x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_{l=1}^{K} e^{\hat{\delta}_l(x)}}$$



How are observations classified?
Similar to Logistic Regression
When Class (K) = 2, PR(Y = 2 | X =x) > .5 (Probability) – Classify to 2, Else 1

K-Nearest Neighbors Classifier

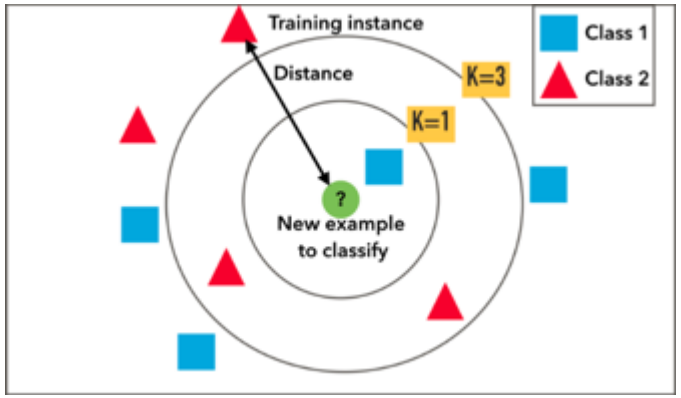
K-Nearest Neighbors

- Introduction
- Distance Measures
- Normalization/Feature Scaling
- Pro's and Cons of KNN

Introduction

- K-Nearest Neighbors is a non-parametric/instance based classifier
  - Predicts Unknown values based on the most similar values (nearest neighbors)
- Classifies data based on a number of factors:
  - Neighbors specified (1, 3)
  - Calculates distance between nearest neighbors (Euclidean Distance or other specified distance measure)
  - Classifies based on the greatest number of nearest neighbors (Majority Vote)
- In the example diagram –
  - K = 1 would be classified as Square
  - K = 3 would be classified as Triangle

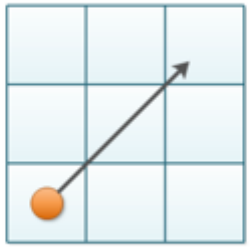
$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j).$$



Distance Measures

- How is the distance between the new example to classify and test data calculated?
  - Many Distance formulas can be used including Euclidean, Chebyshev, Manhattan, and Hamming Distance
- Euclidean Distance is a popular formula used to calculate the distance between neighbors in K Nearest Neighbors algorithm

Euclidean Distance

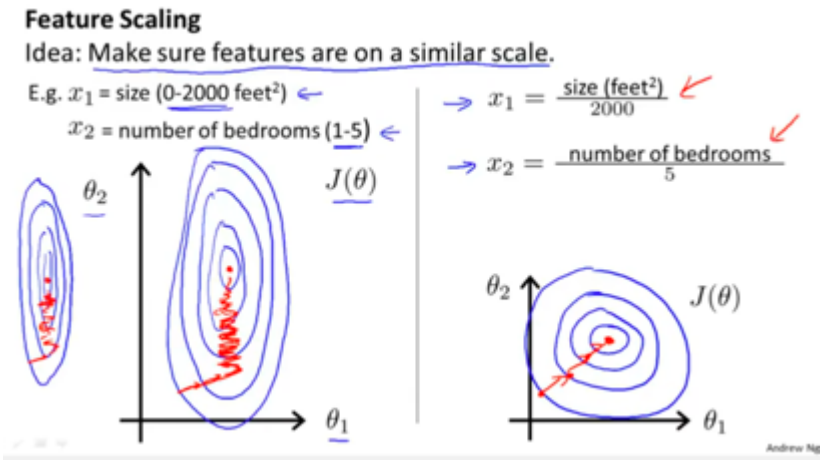


$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Normalization/Feature Scaling

- K-Nearest Neighbors requires that you normalize/feature scale all of your data points
- This is because features with higher values can have an impact on the distance measures
- Feature scaling/Normalization often rescales a datasets features to values between [0,1] or [-1,1]

Source: Andrew Ng
Stanford University



K-Nearest Neighbors Pros/Cons

Pros

- Simple to understand
- Useful for nonlinear data as it is non-parametric

Cons

- Computationally expensive because it stores all of the training data
- Requires feature scaling
- Need to choose number of neighbors

Thank you!