Master of Technology in Knowledge Engineering

Unit 1

Intelligent Systems & Techniques for Business Analytics

Machine Learning:

Experiments & Models Performance

Sam GU Zhan 顾 瞻

zhan.gu@nus.edu.sg

Objective

- To introduce basic concepts and methods of variable / feature selection

- To introduce machine learning model evaluation

- To understand characteristics of different classification models

Outline

- Selection of best variable set
- Cross validation technique
- More classification models
    - » Logistic regression for classification
    - » Linear Discriminant Analysis (LDA)
    - » Quadratic Discriminant Analysis (QDA)

Variable / Feature Selection

Best Subset Selection

- Assume *p* variables as potential predictors, and ***n*** samples
- *Null model M_0*
  - » Contains an intercept (β_0) only but no predictor, and gives the sample mean as prediction for each observation.
- For *k* = 1, 2, …, *p*
  - » Fit all $\begin{pmatrix} p \\ k \end{pmatrix}$ models that contain exactly *k* predictors
  - » Pick the ***best*** among these $\begin{pmatrix} p \\ k \end{pmatrix}$ models, and call it M_k
    - ☝ The best is defined as having the smallest RSS, or largest R^2
- Select a single best from among M_0, …, M_p using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 (to be discussed later)

Forward Stepwise Selection

- Begin with the null model M_0
- For k = 0, 1, 2, …, p-1:
  - » Fit all p-k models that augment the predictors in M_k with one additional predictor
  - » Pick the ***best*** among these p-k models, and call it M_{k+1}
    - ♦ The best is defined as having the smallest RSS, or largest R^2
- Select a single best from among M_0, …, M_p using cross-validated prediction error

Backward Stepwise Selection

- Begin with the full model M_{p} , which contains all p predictors
- For k = p, p-1, …,1:
  - » Fit all *k* models that contain all but one of the predictors in Mk, for total of *k-1*predictors
  - » Pick the ***best*** among these *k* models, and call it M_{k-1}
    - ♦ The best is defined as having the smallest RSS, or largest R^2
- Select a single best from among M_0, …, M_p using cross-validated prediction error

Variable Selection: a brief summary

- Computational complexity
  - » Best subset selection involves fitting 2^p models
  - » Both Forward and Backward stepwise selection search through $1+p(p+1)/2$ models
- Applicability
  - » Backward selection requires $n > p$ (so that the full model can be fit)
  - » Forward selection can always be used, but will not yield a unique solution if $p \geq n$
- Performance
  - » Forward and backward selection are not guaranteed to yield the best model containing a subset of the p predictors

Hybrid Approaches

- Attempts to mimic best subset selection while retaining the computational advantages of forward and backward stepwise selection

- Hybrid versions of forward and backward stepwise selection
  - » Adding new variable by in “forward” manner
  - » Removing any variables that no longer provide an improvement in the model fit

*Choose the Optimal Model

- The task of selecting a single best from among M_0, …, M_p must be performed with care
  - » If we use RSS, or R^2 statistics to select the best mode, we will always end up with a model involving all of the variables

    ☝ the RSS of these models decreases monotonically and R^2 increases monotonically, as the number of features included in the model increases
  - » A low RSS or a high R^2 indicates a model with a low ***training*** error, whereas we wish to choose a model that has a low ***test*** error

Choose the Optimal Model: Cp, AIC, BIC, Adj-R^2

- C_p, AIC, BIC are the measures *for selecting and comparing models*.

Smaller values indicate better models

  - » ***C_p*** Unbiased estimate of test MSE.
  - » ***AIC*** Akaike information criterion (C_p and AIC are proportional to each other)
    - ***AICc*** “corrects” the AIC for small sample sizes. As the sample size increases, the AICc converges to the AIC
  - » ***BIC*** Bayesian information criterion
    - Similar to AIC and Cp.
  - » Generally we select the model that has the lowest BIC value

- ***Adjusted R^2***
  - » A large value indicates a model with small test error

(no further discussion)

☝ A low RSS or a high R^2 indicates a model with a low ***training*** error, whereas we wish to choose a model that has a low ***test*** error

ML Model Evaluation

Evaluating Performance on a Task

- Thinking about Student Exam Predictor:
  - » To evaluate the ***accuracy*** of our newly trained model
    - ♦ count the number of correctly classified students, both passed & failed, and divide that by the total number of students
  - » However, it doesn’t indicate weather over-fitting happened
    - ♦ A better idea would be to use only 80% of the exam data & results for training, and the remaining 20% as a ***test set***

Performance: what to measure

- Confusion matrix:
  - » Imagine you have your two classifiers (models) C1 and C2

C1 achieves the results on a test data set

tpr = 0.75, *tnr* = 1.00
accuracy = 0.80
avg-true = (*tpr* + *tnr*)/2 = 0.88

	Predicted(+)	Predicted (-)	
Actual (+)	**60**	20	80
Actual (-)	0	**20**	20
	60	40	100

C2 achieving the results on the ***same*** test set

tpr = 0.94, *tnr* = 0.50
accuracy = 0.85
avg-true = (*tpr* + *tnr*)/2 = 0.72

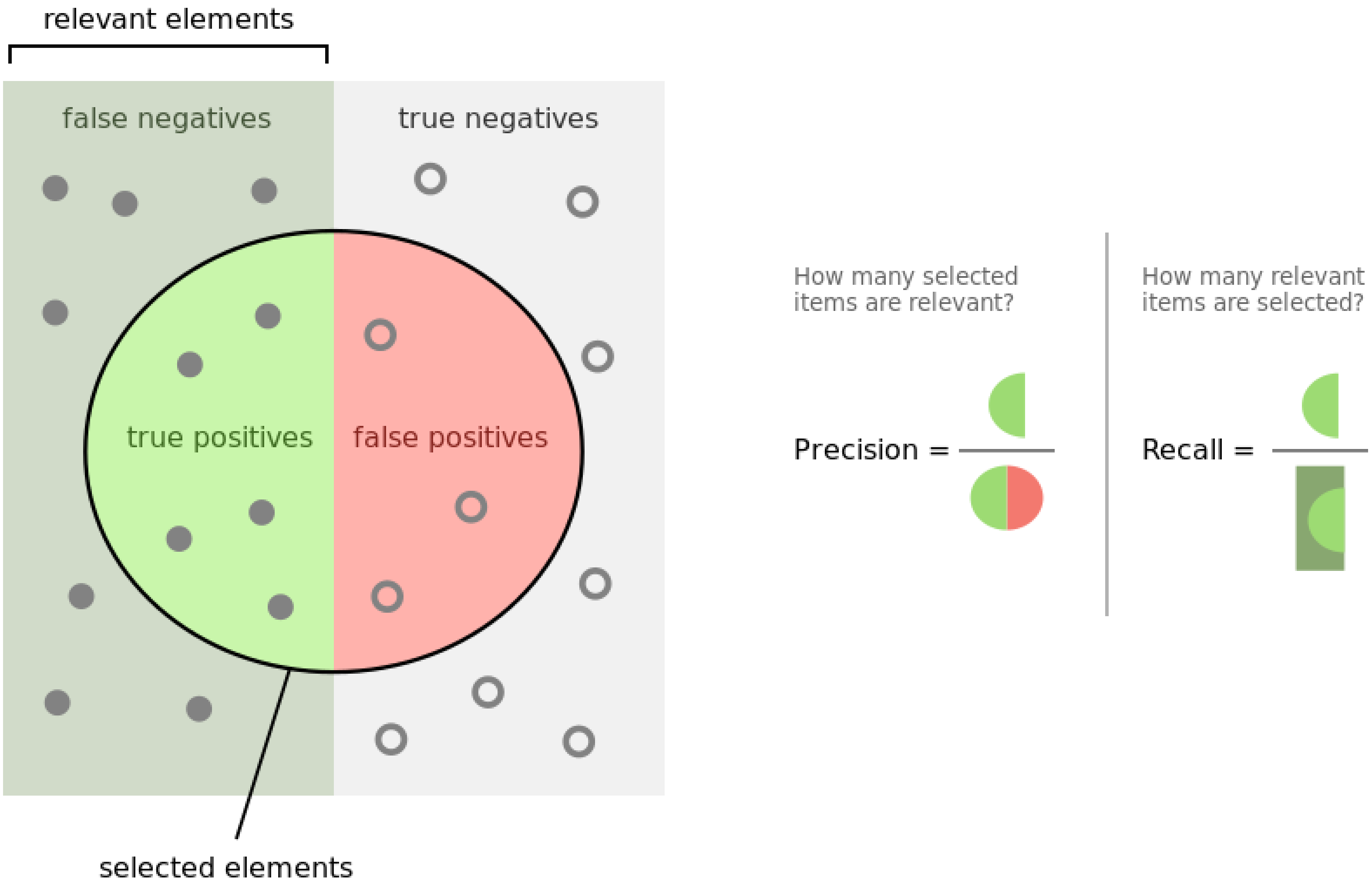
	Predicted(+)	Predicted (-)	
Actual (+)	**75**	5	80
Actual (-)	10	**10**	20
	85	15	100

tpr: true-positive-rate; tnr: true-negative-rate
avg-true: average recall & specificity

Performance: choose a good model

- Which of the models, C1 and C2, you should choose?
  - » If the class distribution in the test set is representative
    - ♦ C2 is better
      - C2: *accuracy* = 0.85;   C1: *accuracy* = 0.80
  - » If we have no prior information about the class distribution in the operating context
    - ♦ C1 should be chosen
      - C1: *avg-true* = 0.88; C2: *avg-true* = 0.72

Evaluation Measures



Evaluation Measures

- Take the previous example of classifier C2
  - **Precision**

    prec = TP/(TP+FP)

    = 75/85 ≈ 0.88
  - **Recall**

    rec = TP/(TP+FN)

    = 75/80 ≈ 0.94
  - **F-measure**
    - ♦ The harmonic mean of precision and recall

$$\frac{TP}{TP+(FP+FN)/2}=\frac{2}{1/prec+1/rec}=\frac{prec\times rec}{(prec+rec)/2}\approx 0.91$$

	Predicted(+)	Predicted (-)	
Actual (+)	**75**	5	80
Actual (-)	10	**10**	20
	85	15	100

How to Measure

- Dividing data to training and test sets is useful for obtaining a better idea about the actual performance of your learned model
  - » However, even if we select the test instances *randomly*
    - ♦ every once in a while we may get “lucky”, if most of the test instances are similar to training instances
    - ♦ or unlucky the test instances happen to be very non-typical or noisy
- Validation Approaches
  - » Validation set approach,
  - » Leave-One-Out Cross-Validation (LOOCV),
  - » K-fold CV Cross-Validation

Cross-Validation: Validation Set Approach

- When we like to compare performance of different learning models, or different settings of the same learning model (i.e. find different variables/features or model complexity that gives the lowest test error rate)
  - » If we have a large data set, we can achieve this goal by randomly splitting the data into training and validation parts
  - » We would then use the training part to build each possible model (e.g. the different combinations of variables) and choose the model that gave the lowest error rate (such MSE) when applied to the validation data

Cross-Validation: Validation Set Approach (cont.)

- Advantages:
  - » Simple
  - » Easy to implement

- Disadvantages:
  - » The validation MSE can be highly variable
    - ♦ i.e.: not stable if a different validation set is selected from the sample data
  - » Only a subset of observations (sample data) are used as training data to fit the model

Cross-Validation: Leave-One-Out (LOOCV)

- For each suggested model, do:
  - » Split the data set of size ***n*** into
    - ♦ Training data set: ***n*-1**; Validation data set: **1**
  - » Fit the model using the training data, then validate model using the validation data and compute the corresponding MSE based on a single observation
  - » Repeat this process ***n*** times (i.e.: each time leaves 1 out)
  - » The MSE for the model is computed as follows:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} MSE_i$$

    - ♦ It is computationally intensive!

Cross-Validation: *k*-fold

- Basic idea: *Randomly* partition the data into *k* different parts of ‘folds” (e.g. *k* = 5, or *k* = 10, etc.)
  - » Then set *one* fold aside for testing, train a model on the remaining *k*–1 folds, and evaluate the trained model on the test fold (i.e. compute the MSE on the test fold)
  - » This process is repeated *k* times until each fold has been used for testing once.
  - » By averaging the *k* different MSE’s we get an estimated validation (test) error rate for new observations

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

Cross-Validation: *k*-fold (cont.)

- The rational behind *k*-fold:
  - » When leave one fold for test, the rest *k-1* folds will be used for training using the selected learning algorithm whose output is a model
    - ♦ *k* ***different*** models will be obtained
  - » by averaging over those models we get a sense of the accuracy & stability (variance) of the ***learning algorithm***

Validation Set Approach, LOOCV, *k*-fold

- LOOCV versus Validation Set approach
  - » LOOCV has less bias
    - ♦ We repeatedly fit the statistical learning method using training data that contains ***n*-1** observations, i.e. almost all the data set is used
  - » LOOCV produces fixed error variance
    - ♦ The Validation Set approach produces different MSE when applied repeatedly due to randomness in the splitting process, while performing LOOCV multiple times will always yield the same results

Validation Set Approach, LOOCV, *k*-fold (cont.)

- LOOCV versus *k*-fold
  - » LOOCV is more computationally intensive!
    - ♦ LOOCV fit the learning model $\boldsymbol{n}$ times!

k-fold solves this problem

- » Both are stable, *k*-fold contains LOOCV as a special case, where $k = \boldsymbol{n}$

Bias-Variance Trade-off

- Putting aside that LOOCV is more computationally intensive than k-fold CV… Which is better LOOCV or *k*-fold CV?
  - » When $k < n$, LOOCV is less bias than k-fold CV
    - ♦ but has higher variance than k-fold CV

- Conclusion:
  - » We tend to use k-fold CV with ($k = 5$ and $k = 10$)
    - ♦ It has been empirically shown that they yield test error rate estimates that suffer neither from excessively high bias, nor from very high variance

Lab: Model evaluation, Cross-validation

- Objectives
  - » Get familiar with the R / Python commands for performing and analyzing cross-validation and understand the performance summary

  - » Further understand:
    - ♦ Cross-validation methods
    - ♦ Overfitting, Underfitting and Model Selection
  - » R user 1, 2; Python user 1, 2.

More Classification Models

Case 1: Brand Preference for Orange Juice

- We would like to predict what customers prefer to buy: F&N Magnolia or Florida's Natural orange juice?
  - » Assume
    - ♦ The Y (Purchase) variable is *categorical*: 0 or 1
    - ♦ The X (LoyalF&N) variable is a numerical value (between 0 and 1) which specifies how much the customers are loyal to the F&N Magnolia (F&N) orange juice

- Possible solutions
  - » Modeling response *Y* directly to categories, or
  - » Predicting ***probability*** that *Y* belongs to a particular category

Regression for Classification

- Instead of trying to predict Y directly
  - » We predict Pr(Y=1), i.e., the probability a customer buys F&N (F&N) juice.
- Linear regression is inappropriate for classification
  - » The regression line $\beta_0+\beta_1X$ can take on any value between negative and positive infinity
- What we need
  - » Model Pr(Y=1) using a function that gives outputs between 0 and 1.

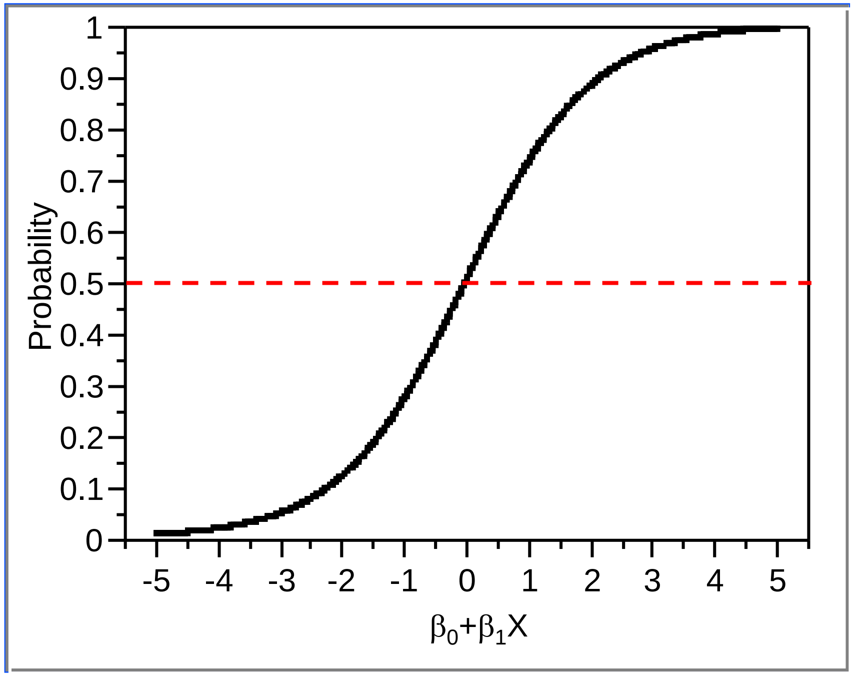
Logistic Regression

We abbreviate Pr(Y = 1 | X) as $p(X)$

- We use the logistic function

$$\Pr(Y=1 \mid X) =$$

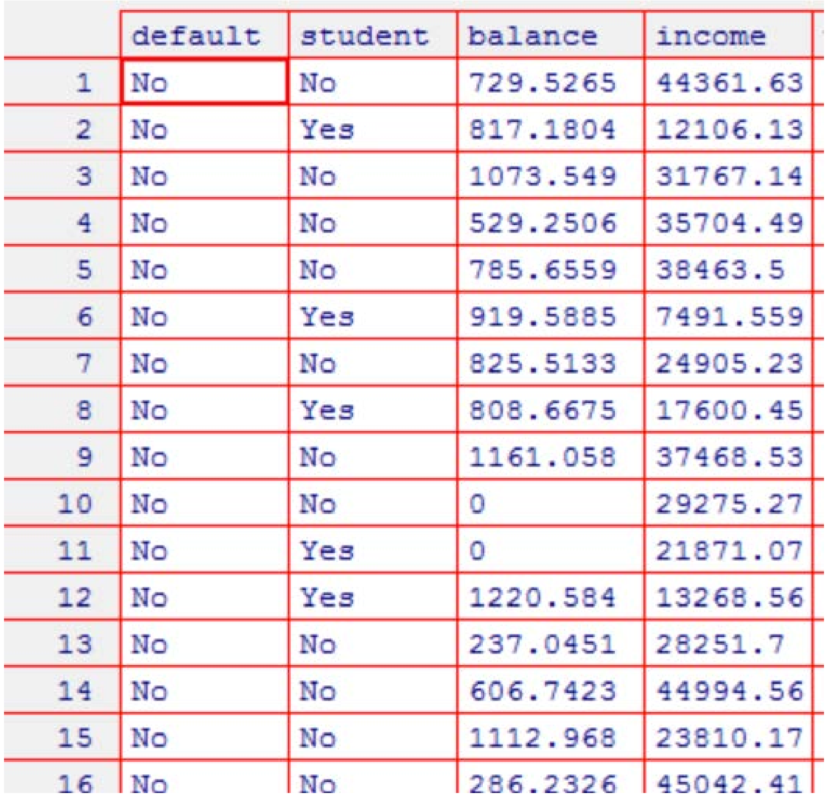
$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$



- Logistic regression is similar to linear regression
  - » We come up with estimate of β_0 and β_1.

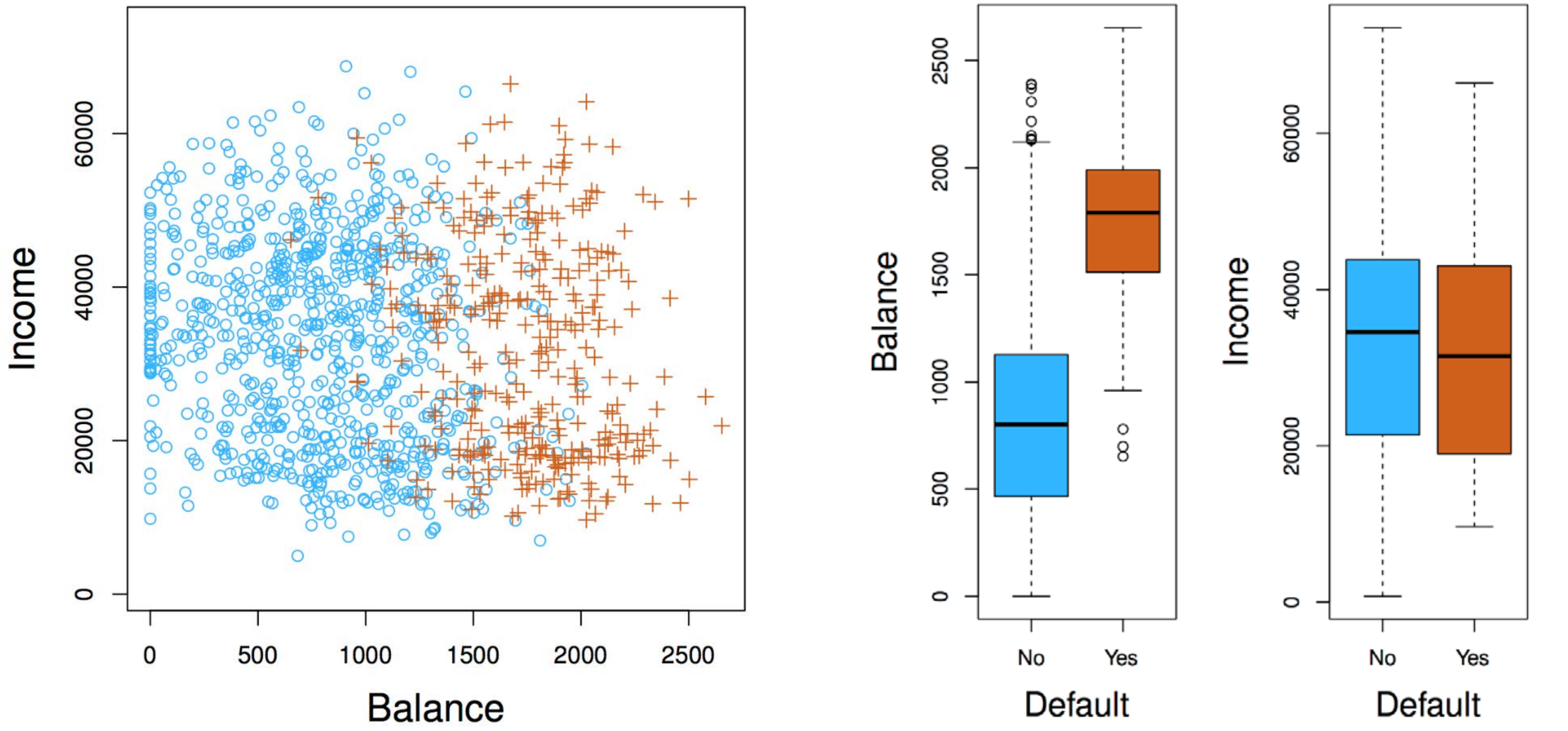
Case 2: Credit Card Default

- We would like to be able to predict customers that are likely to default
- Possible X variables are:
  - » Annual Income
  - » Monthly credit card balance
- The Y variable (Default) is categorical: Yes or No
- How do we check the relationship between Y and X?



	default	student	balance	income
1	No	No	729.5265	44361.63
2	No	Yes	817.1804	12106.13
3	No	No	1073.549	31767.14
4	No	No	529.2506	35704.49
5	No	No	785.6559	38463.5
6	No	Yes	919.5885	7491.559
7	No	No	825.5133	24905.23
8	No	Yes	808.6675	17600.45
9	No	No	1161.058	37468.53
10	No	No	0	29275.27
11	No	Yes	0	21871.07
12	No	Yes	1220.584	13268.56
13	No	No	237.0451	28251.7
14	No	No	606.7423	44994.56
15	No	No	1112.968	23810.17
16	No	No	286.2326	45042.41

The Default Dataset



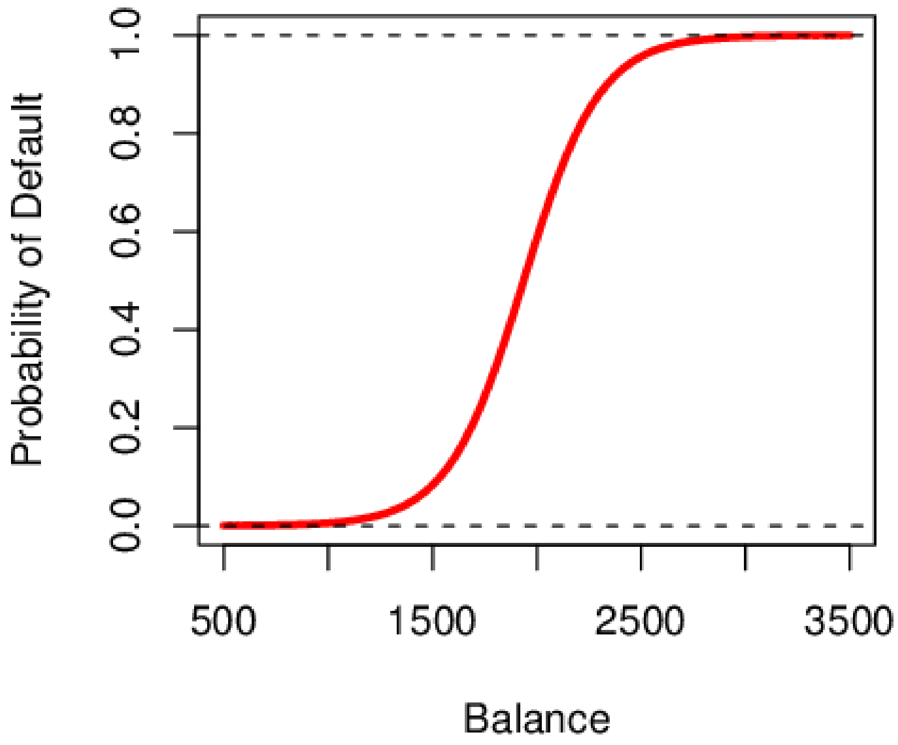
√

?

[Source: G. James, D. Witten, T. Hastie, R. Tibshirani, “An Introduction to Statistical Learning”]

Logistic Function on Default Data

- Now the probability of default is
  - close to, but not less than 0%, for low balance cases, and
  - close to but not above 100%, for high balance cases

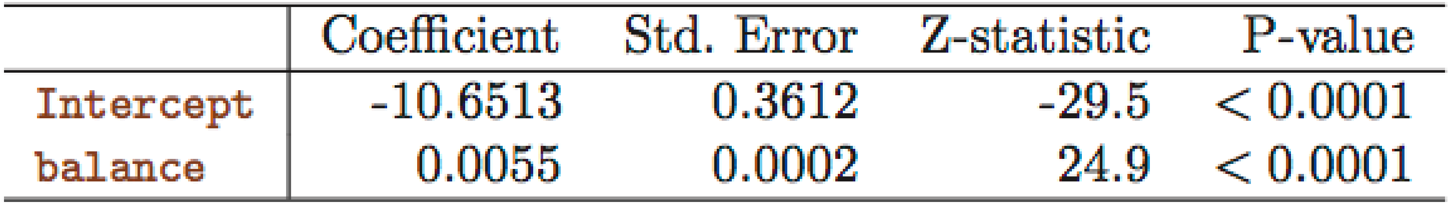


Interpreting β_1

- Interpreting what β_1 means is not very easy with logistic regression, simply because we are predicting P(Y) but not Y.
  - » If β_1 =0, this means that there is no relationship between Y and X.
  - » If β_1 >0, this means that when X gets larger so does the probability that Y = 1.
  - » If β_1 <0, this means that when X gets larger, the probability that Y = 1 gets smaller.
  - » But how much bigger or smaller depends on where we are on the slope

Are the Coefficients Significant?

- Here the p-value for balance is very small, and β_1 is positive, so we are sure that if the balance increase, then the probability of default will increase as well.



	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

(*) a Z test is used instead of a T test, but that doesn't change the way to interpret the p-value. Read more here.

(*) the estimated intercept in the table is typically not of interest

(further details omitted)

Making Prediction

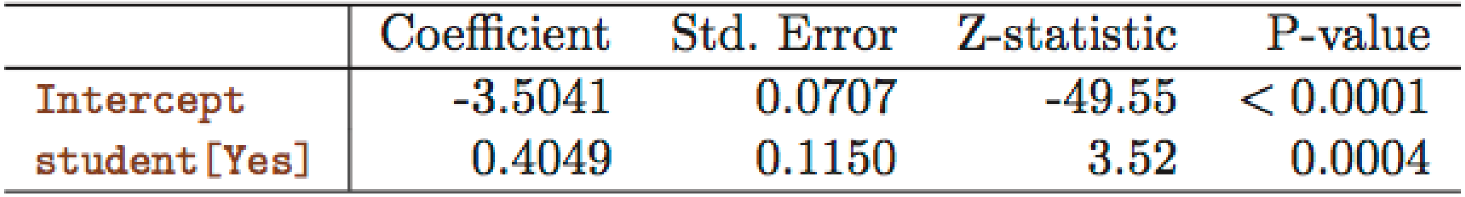
- Suppose an individual has an average balance of \$1000. What is their probability of default?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.00576$$

- The predicted probability of default for an individual with a balance of \$1000 is less than 1%.
- For a balance of \$2000, the probability is much higher, and equals to 0.586 (58.6%).

Qualitative Predictors in Logistic Regression

- We can predict if an individual default by checking if he/she is a student or not.
  - » Qualitative predictor: Student = 1, Non-student =0



	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

- β_1 is positive: this indicates students tend to have higher default probabilities than non-students

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

Multiple Logistic Regression

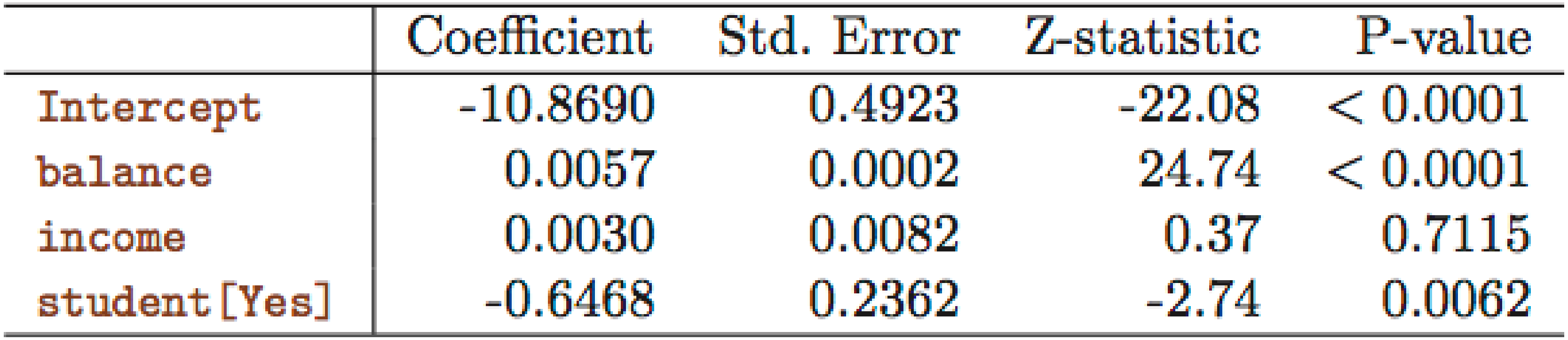
- We can fit multiple logistic

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

where $\mathbf{X} = (X_1, \ldots, X_p)$ are p predictors.

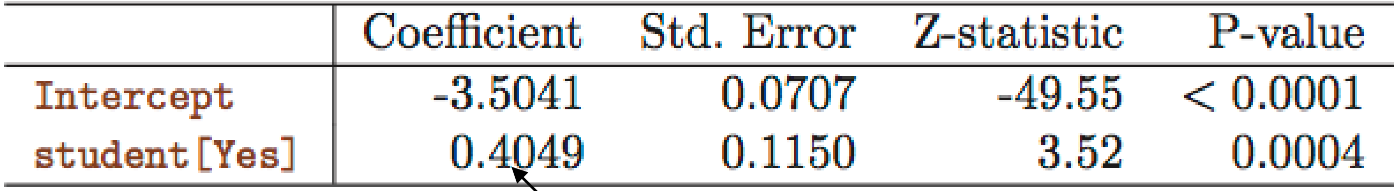
Multiple Logistic Regression: Default Data

- Predict Default using:
  - » Balance (quantitative)
  - » Income (quantitative)
  - » Student (qualitative)



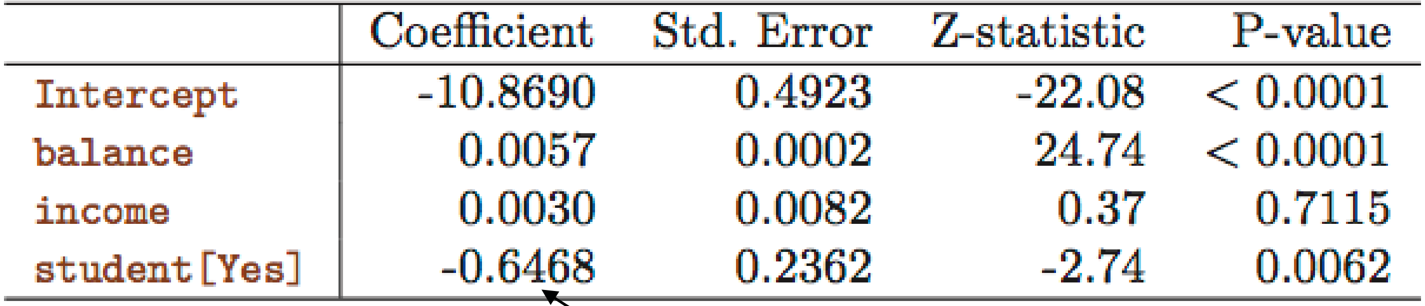
	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

An Apparent Contradiction!



	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

Positive



	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

Negative

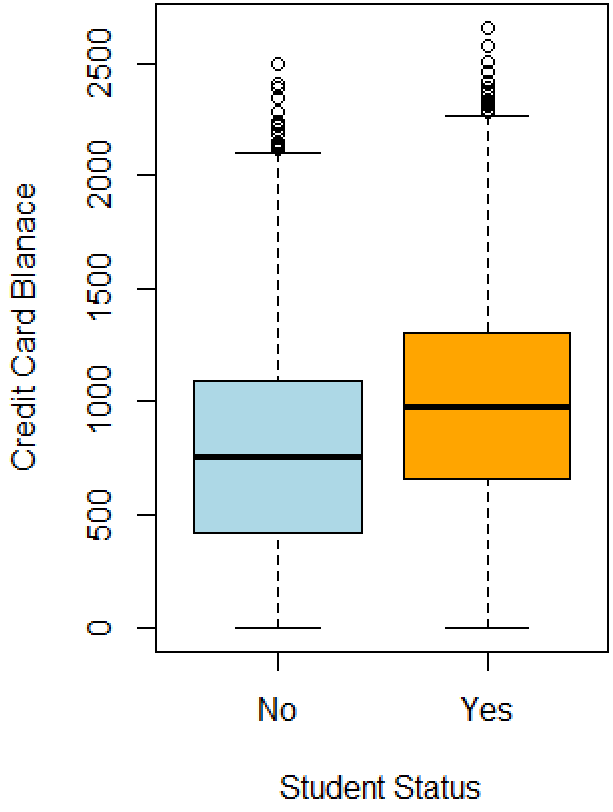
- A student is risker than non students
  - » if no information about the credit card balance is available
- However, that student is less risky than a non student
  - » with the same credit card balance!

To Whom should Credit be Offered?

- The variable “student” and “balance” are correlated.
  - » students tend to hold higher level of debt

☝ As in the linear repression setting

- » The result obtained using one predictor may be quite different from those obtained using multiple predictors, especially when there is correlation among the predictors

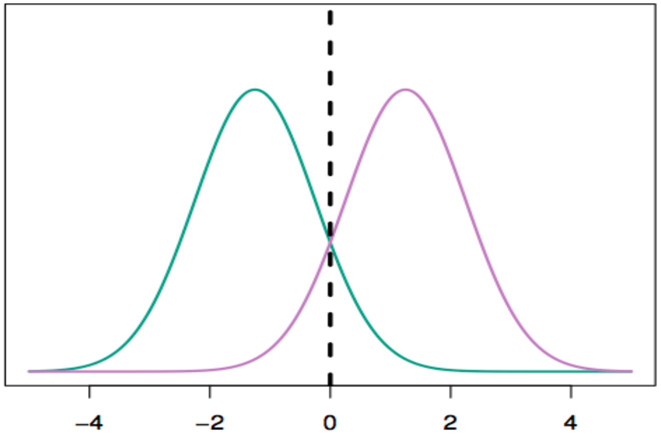


Linear Discriminant Analysis (LDA)

- *Linear discriminant analysis* (LDA) undertakes the same task as Logistic Regression.
  - » It classifies data based on categorical variables, e.g.:
    - ♦ Making profit or not, Buy a product or not, …
- In the case where *n* is small, and every distribution of predictors X are approximately *normal*, then LDA is more stable than Logistic Regression

A Simple Example with One Predictor

- Suppose we have only one predictor (*p* = 1)
  - » Two normal density function $f_1(x)$ and $f_2(x)$, represent two distinct classes
  - » The two density functions overlap, so there is some *uncertainty* about the class to which an observation belongs



☝ Logistic regression can be unstable

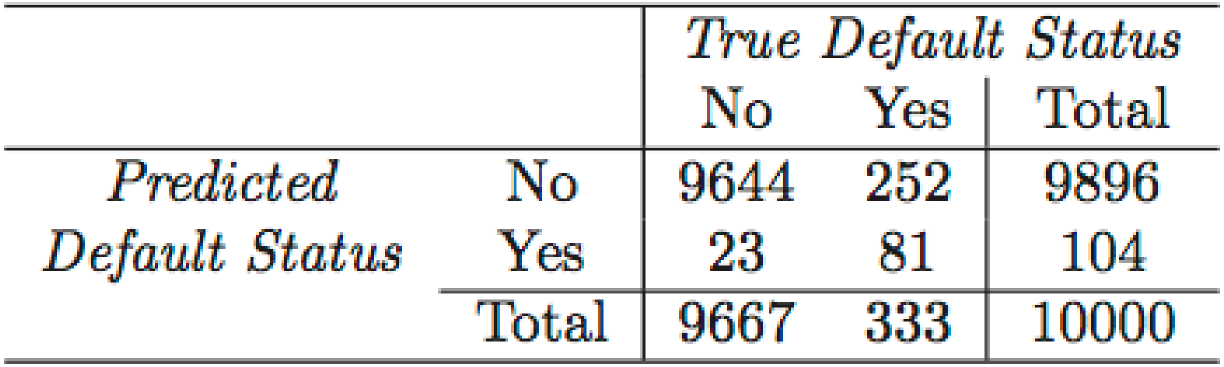
  - » The dashed vertical line represents Bayes' decision boundary. LDA with an appropriate threshold can achieve an error rate close to Bayes' error rate

Apply LDA

- Assumptions of LDA
  - The observations are a *random* sample
  - Each predictor variable is *normally* distributed
  - Each class has a *normal* distribution with a *common / similar variance*
- The mean and the variance are estimated
- Finally, Bayes’ theorem is used to compute p_k and the observation is assigned to the class with the maximum probability among all *k* probabilities/classes

Threshold in Running LDA

- Running LDA on Default data
  - » Threshold used for predicting default: 0.5
  - » LDA makes 252+ 23 mistakes on 10000 predictions (2.75% misclassification error rate)
  - » But LDA miss-predicts 252/333 = 75.5% of defaulters!

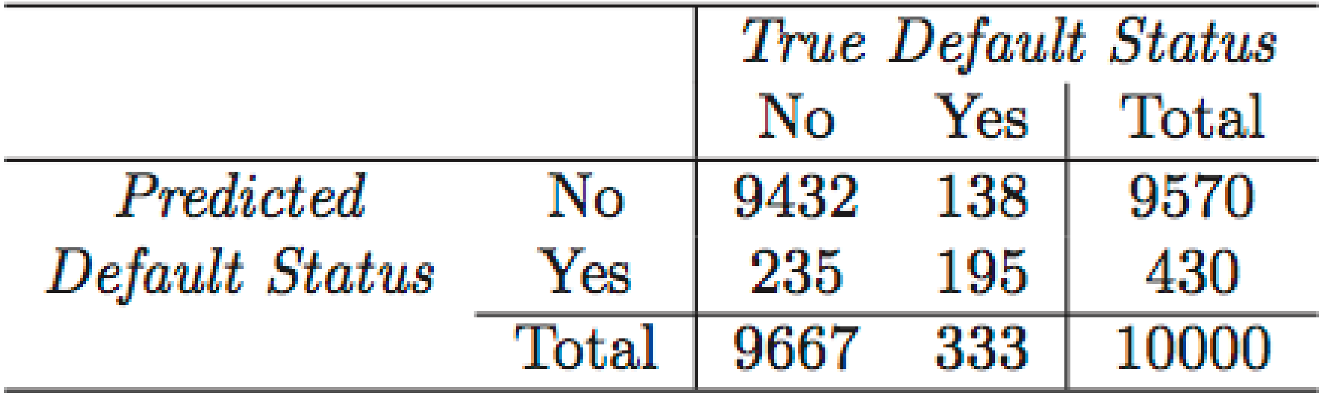


		True Default Status		
		No	Yes	Total
Predicted Default Status	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

Threshold in Running LDA (cont.)

- Using 0.2 as threshold
  - Now the total number of mistakes is 235+138 = 373 (3.73% misclassification error rate)
  - But we only miss-predicted 138/333 = 41.4% of defaulters

(we can examine the error rate with other thresholds)



		True Default Status		
		No	Yes	Total
Predicted Default Status	No	9432	138	9570
	Yes	235	195	430
	Total	9667	333	10000

Annex: LDA as estimating Bayes' classifier

- With Logistic Regression we modeled the probability of Y being from the *k*th class as

$$p_k(X) = \Pr(Y = k \mid X = x) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- Bayes' Theorem states

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

π_k : Probability of coming from class *k* (prior probability)

$f_k(X)$: Density function of X given that X is an observation from class *k*

Annex: LDA as estimating Bayes' classifier (cont.)

- We can estimate π_k and $f_k(X)$ to compute $p(X)$
- The most common model for $f_k(X)$ is the Normal Density

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

  » Using the density, we only need to estimate three quantities to compute $p(X)$

$$\mu_k \qquad \sigma_k^2 \qquad \pi_k$$

☝ If X is multidimensional (p > 1), the multivariate normal density will be used instead for the density function *f(x)*

Annex: LDA as estimating Bayes' classifier (cont.)

- The *mean* could be estimated by the average of all training observations from the *k*th class.

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

- The *variance* could be estimated as the weighted average of variances of all *K* classes.

$$\hat{\sigma}^2 =$$

$$\frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

- And, the *prior probability* is estimated as the proportion of the training observations that belong to the *k*th class.

$$\hat{\pi}_k = n_k / n$$

Annex: LDA as estimating Bayes' classifier (cont.)

- Taking the log of $p_k(X) = \Pr(Y = k \mid X = x) = \dfrac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$

  with necessary rearrangement, what LDA classifier does is equivalent to assigning an observation *X* = x to the class for which

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

  is the largest

  - » it is a linear function of x
  - » with the estimates given, this *linear discriminant function* gives a linear decision boundary

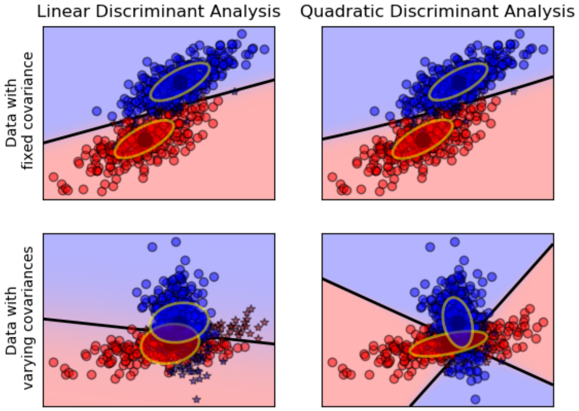
(further mathematical details omitted)

Quadratic Discriminant Analysis (QDA)

- *Quadratic discriminant analysis* (QDA) works identically as LDA except that
  - it estimates separate variances for each class
  - for an observation X = x, the quantity x appears as a ***quadratic*** function in the model to fit (this is where the QDA gets its name) (further details omitted)
- Since QDA allows for different variances among classes, the resulting boundaries become ***quadratic***

LDA or QDA?

- QDA allows for different variances among classes
  - » variances are very different between classes
  - » we have enough observations to accurately estimate the variances



- LDA assumes that every class has

  the similar variance / covariance
  - » the variances are similar among classes
  - » we don't have enough data to accurately estimate the variances

Comparison: Logistic Regression vs. LDA

- Similarity:
  - » Both Logistic Regression and LDA produce linear boundaries
- Difference:
  - » LDA assumes that the observations are drawn from the normal distribution with similar variance in each class, while logistic regression does not have this assumption.
  - » LDA would do better than Logistic Regression if the assumption of normality hold, otherwise logistic regression can outperform LDA
  - » LDA is more popular when we have more than two response classes

Comparison: *k*-NN vs. (LDA, Logistic Regression)

- *k*-NN is completely *non-parametric*:
  - No assumptions are made about the shape of the decision boundary!
- Advantage of *k*-NN: *high flexibility*
  - We can expect *k*-NN to dominate both LDA and Logistic Regression when the decision boundary is highly non-linear
- Disadvantage of *k*-NN: *low interpretability*
  - *k*-NN does not tell us which predictors are important (no table of coefficients)

Comparison: QDA, LDA, Logistic Regression, *k*-NN

- QDA is a compromise between non-parametric KNN method and the linear LDA and logistic regression

- If the true decision boundary is:
  - » Linear: LDA and Logistic regression outperform
  - » Moderately Non-linear: QDA outperforms
  - » More complicated: *k*-NN is superior

Lab: logistic regression, LDA, QDA, *k*-NN

- Objectives
  - » Get familiar with the R / Python commands for performing different approaches of classification and understand the performance summary
  - » Further understand the similarity and difference among
    - ♦ Logistic regression, LDA, QDA, and *k*-NN
  - » R user: LR; LDA &QDA here ; k-NN here
  - » Python user: LR; LDA & QDA here & here; k-NN here & here

Summary

- We have discussed machine learning from different perspectives, and in different aspects
  - » Tasks of learning
    - ♦ classification, regression, prediction, forecast, segmentation
  - » Mechanism of learning (paradigms)
    - ♦ Supervised, unsupervised, reinforcement
  - » Models / methods
    - ♦ Linear regression, Logistic regression, LDA, QDA, k-NN, Decision trees, NN, SVM, Clustering
  - » Types of models
    - ♦ Geometric, probabilistic, logical, grouping, grading
  - » Feature selection: best subset, forward, backward
  - » Experiments: cross-validation; model-complexity & over-fitting

References

- G. James, D. Witten, T. Hastie & R. Tibshirani, “An Introduction to Statistical Learning with Applications in R”, Springer, 2013
- Shmueli, G., N.R. Patel, P.C. Bruce, “Data Mining for Business Intelligence”, Wiley, 2010