NOVA IMS
Information Management School

6

MODELING: CLASSIFICATION

Machine Learning for Marketing

Instituto Superior de Estatística e Gestão da Informação
Universidade Nova de Lisboa

Summary

1. Measures of performance
2. Logistic Regression
3. Support Vector Machine (SVM)
4. K-Nearest Neighbor (KNN)
5. Neural Networks
6. Naïve Bayes
7. Decision Tree

6.1

Measures of performance

Modeling: Classification

NOVA IMS Information Management School

“All models are wrong, but some are useful”

[George E. P. Box]

Binary classification

Batch approach

Measures of performance

Measures of performance

FALSE POSITIVE
error type I

FALSE NEGATIVE
error type II

[opentextbc.ca]

Confusion matrix

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">PREDICTED</td></tr>
<tr><td>TRUE</td><td>FALSE</td></tr>
<tr><td rowspan="2">TARGET</td><td>TRUE</td><td>True Positive (TP)</td><td>False Negative (FN)</td></tr>
<tr><td>FALSE</td><td>False Positive (FP)</td><td>True Negative (TN)</td></tr>
</table>

Confusion matrix – cancer prediction

		PREDICTED	
		TRUE	FALSE
TARGET	TRUE	**TP**	FN
	FALSE	FP	**TN**

It is preferable to have **False Positives** than False Negatives

Confusion matrix – justice conviction prediction

		PREDICTED	
		TRUE	FALSE
TARGET	TRUE	**TP**	FN 
	FALSE	FP 	**TN**

It is preferable to have **False Negatives** than False Positives

Measures of performance

$$Accuracy = \frac{\sum TP + \sum TN}{\sum TP + \sum TN + \sum FP + \sum FN}$$

Measures de overall accuracy of the model

⚠ A high accuracy is not an indication that the model is excellent. For example, if only 1 in 100 observations is positive, a model that classifies all observations as negative has a 99% accuracy

Measures of performance

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

Percentage of **predicted positive** cases classified correctly

High values usually indicate good model performance

Measures of performance

$$Sensitivity|Recall|True\ Positive\ Rate\ (TPR) = \frac{\sum TP}{\sum TP + \sum FN}$$

Percentage of **actual positive cases** that were classified correctly by the model or probability of a positive prediction is really positive

High values usually indicate good model performance

Measures of performance

$$Specificity|True\ Negative\ Rate(TNR) = \frac{\sum TN}{\sum TN + \sum FP}$$

Percentage of **actual negative cases** that were classified correctly by the model or probability of a negative prediction is really negative

High values usually indicate good model performance

Measures of performance

$$F1Score = \frac{2 \times \sum TP}{2 \times \sum TP + \sum FP + \sum FN}$$

Combination of *Precision* and *Recall* into one only measure, the harmonic mean of *Precision* and *Recall*

High values usually indicate good model performance

Measures of performance

$$False\ Positive\ Rate\ (FPR) = \frac{\sum FP}{\sum TN + \sum FP}$$

Percentage of **predicted positive** cases classified **incorrectly**

Lower values usually indicate good model performance

Measures of performance

$$False\ Negative\ Rate\ (FNR) = \frac{\sum FN}{\sum TP + \sum FN}$$

Percentage of **predicted negative** cases classified **incorrectly**

Lower values usually indicate good model performance

Churn example

		PREDICTED	
		TRUE	FALSE
TARGET	TRUE	5 000	1 000
	FALSE	1 500	22 500

Total customers = 30 000

Churn customers = 6 000

Accuracy = 0.9167

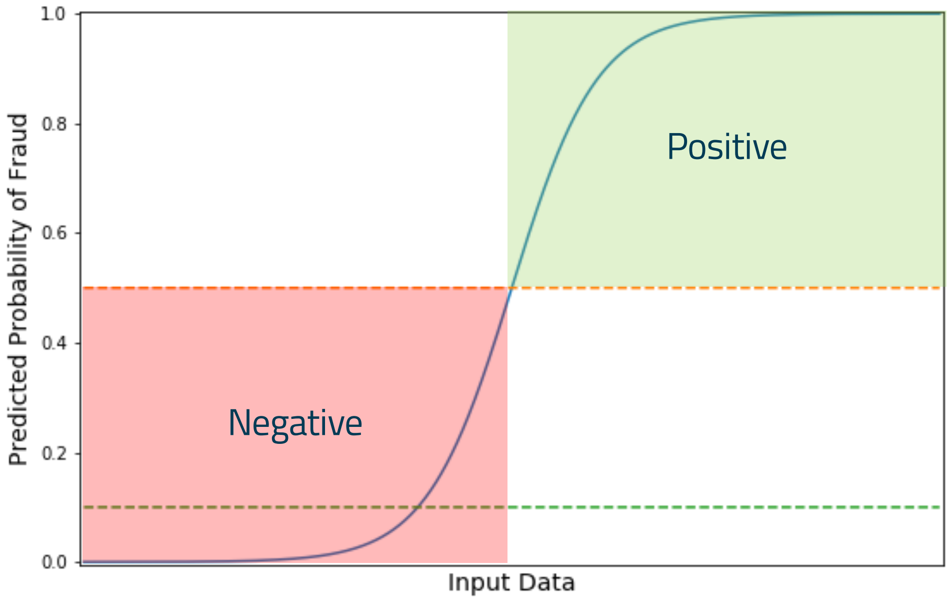
Precision = 0.7692

Sensitivity = 0.8333

False Positive Rate = 0.0625

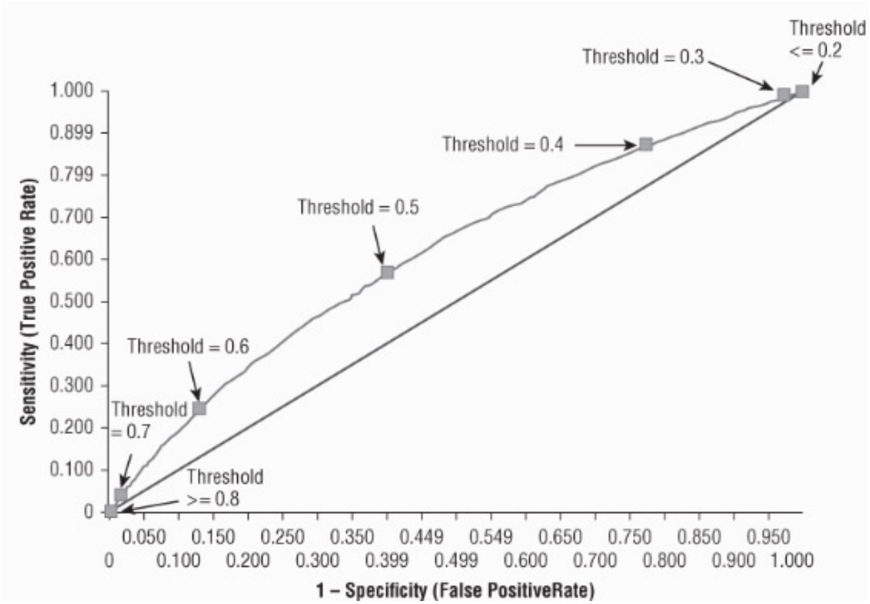
Classification threshold

$$threshold(score, 0.5) = \begin{cases} negative, & score < 0.5 \\ positive, & score \geq 0.5 \end{cases}$$

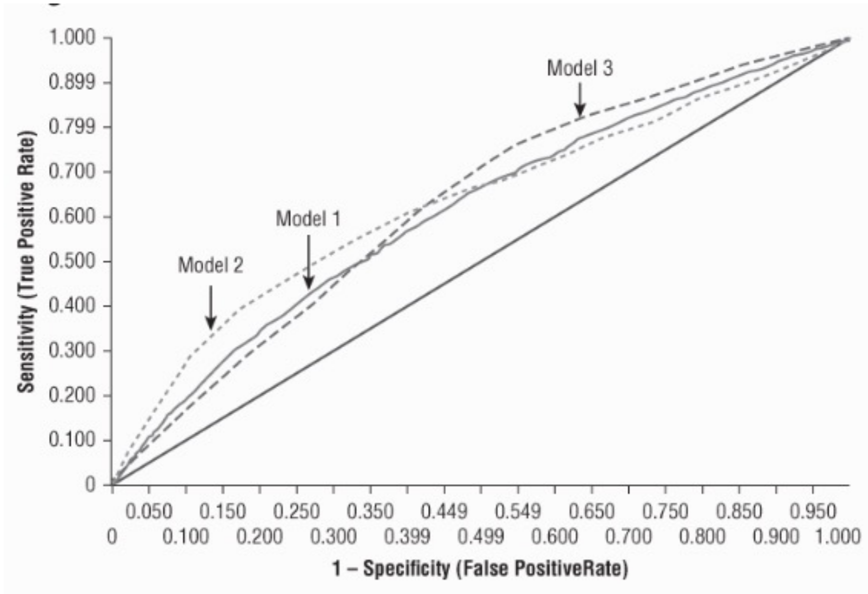


[Adapted from https://towardsdatascience.com/]

Receiver Operating Characteristic (ROC)

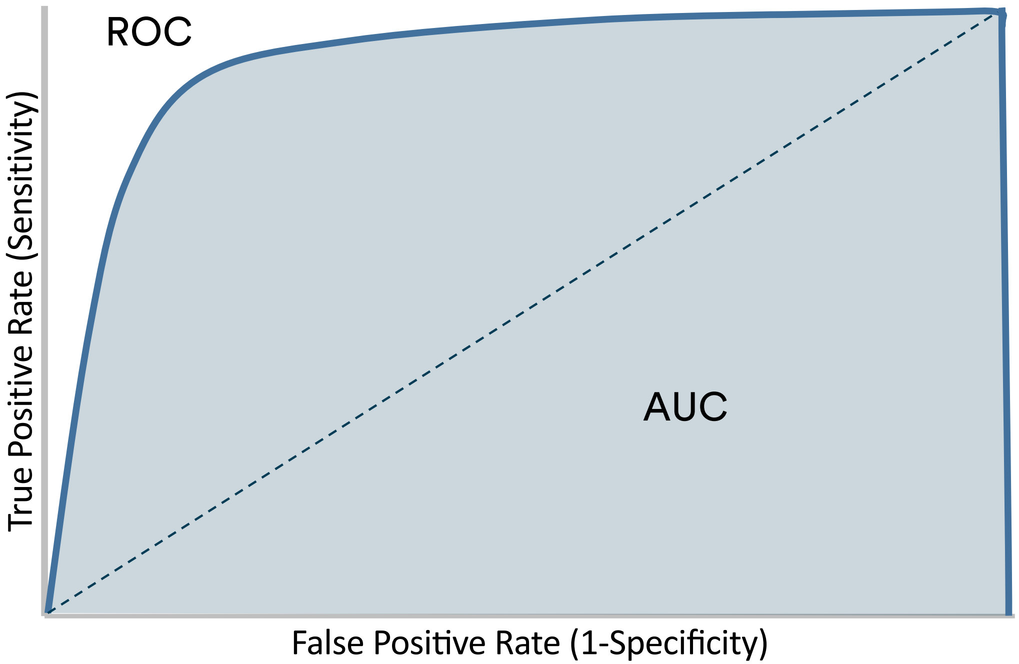


Visualization of *all* confusion matrices (with a different threshold) [0,1]



Very useful to compare models' performance

ROC/AUC



ROC = Receiver Operating Characteristic

AUC = Area Under the Curve

Other measures

- **Gini coefficient:** the linear rescaling of ROC index
- **Kolmogorov-Smirnov Statistic (K-S statistic):** captures the separation between the distribution of prediction scores

Binary classification

Rank-ordered approach

Measures of performance

Rank ordered approach

In many areas, you "treat" those who are **most likely** to respond to the "treatment". For example, select the customers that are likely to churn.

The most common metrics are:

- Gain (segment) = $\frac{target\ positive\ instances\ in\ segment}{total\ of\ target\ positive\ instances}$
- Lift (segment) = $\frac{\frac{target\ positive\ instances\ in\ segment}{instances\ in\ segment}}{\frac{total\ target\ positive\ instances}{total\ instances}}$
- ROI

NOVA IMS Information Management School

How to implement

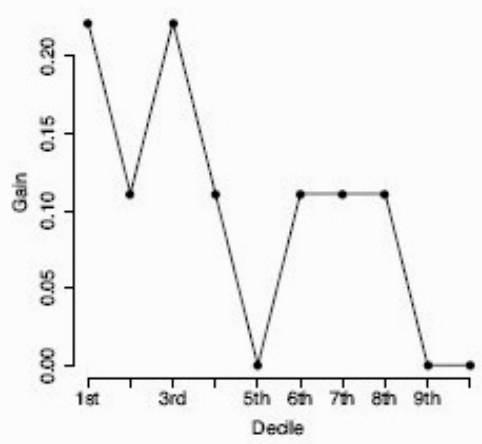
1. **Sort** the numeric output of each observation, either by the probability or confidence (or predicted output in a regression model)
2. **Bin** the predictions into segments (usually deciles - 10% of the dataset)
3. Create **summary statistics** of each segment

Churn example (1/3)

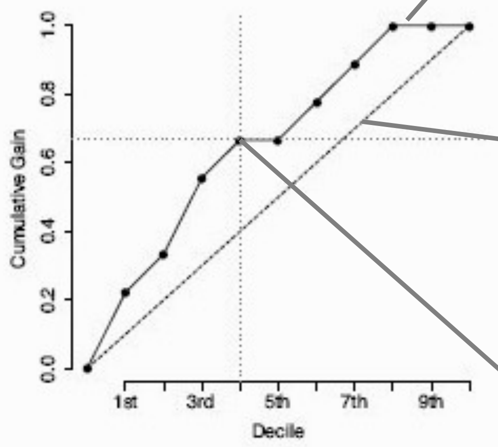
Dec.	Pos. count	Neg. count	Gain	Cum. Gain	Lift	Cum. Lift
1st	2	0	0.222	0.222	2.222	2.222
2nd	1	1	0.111	0.333	1.111	1.667
3rd	2	0	0.222	0.556	2.222	1.852
4th	1	1	0.111	0.667	1.111	1.667
5th	0	2	0.000	0.667	0.000	1.333
6th	1	1	0.111	0.778	1.111	1.296
7th	1	1	0.111	0.889	1.111	1.270
8th	1	1	0.111	1.000	1.111	1.250
9th	0	2	0.000	1.000	0.000	1.111
10th	0	2	0.000	1.000	0.000	1.000

Dec.	ID	Target	Prediction	Score	Outcome
1st	9	T	T	0.963	TP
	4	T	T	0.960	TP
2nd	18	T	T	0.877	TP
	20	N	T	0.833	FP
3rd	6	T	T	0.781	TP
	10	T	T	0.719	TP
4th	17	N	T	0.676	FP
	8	T	T	0.657	TP
5th	5	N	N	0.348	TN
	14	N	N	0.302	TN
6th	16	N	N	0.293	TN
	1	T	N	0.246	FN
7th	2	T	N	0.226	FN
	3	N	N	0.184	TN
8th	19	N	N	0.160	TN
	12	T	N	0.094	FN
9th	15	N	N	0.064	TN
	13	N	N	0.059	TN
10th	7	N	N	0.003	TN
	11	N	N	0.001	TN

Churn example (2/3)



Gain



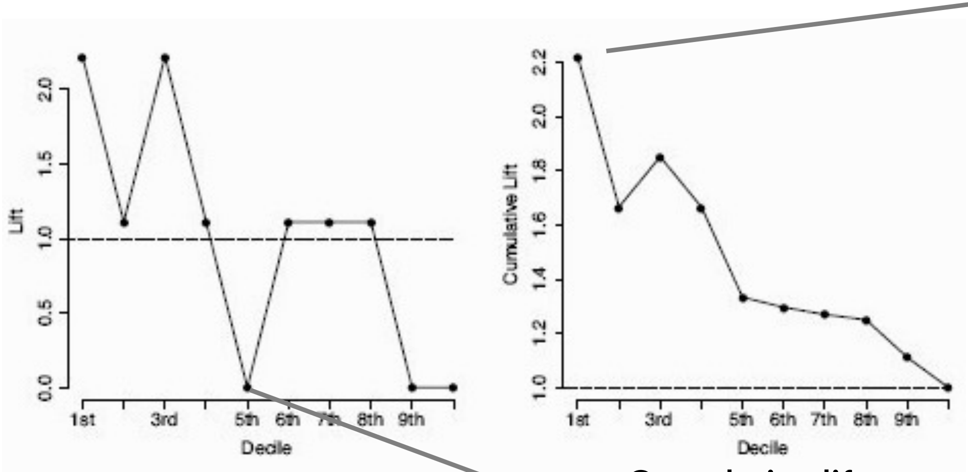
Cumulative gain

The closer the cumulative gain line is to the top left hand corner, the better the model is performing

"Random guessing" performance line

For example: at the 4th decile (40% of the test data), 66.667% of the customers who churn have been identified

Churn example (3/3)



Lift

Cumulative lift

Lift should start with high values in well performing models

A stable model will have monotonic lift values from segment to segment. Erratic segment lift charts are indicative of overfitting

The lift curve should cross 1.0 only at one of the lower deciles (around the 5th decile)

ROI: Profit and Loss confusion matrix

In some problems, measures in the confusion matrix are not worth the same, therefore it might be necessary to evaluate the Return On Investment (ROI)

		PREDICTED	
		TRUE	FALSE
TARGET	TRUE	TP_{profit}	FN_{profit}
	FALSE	FP_{profit}	TN_{profit}

Example: ROI for customer churn (1/2)

Correctly predicting that a customer will not churn or will churn has the same cost. However, predicting a costumer will not churn, but the customer won't have the cost of an "additional discount" (e.g., € 50). Failing to identify customers who will churn has a much higher cost, the "customer average lifetime value"

		PREDICTED	
		TRUE	FALSE
TARGET	TRUE	0 €	- 700 €
	FALSE	- 50 €	0 €

Example: ROI for customer churn (2/2)

MODEL 1

		PREDICTED	
		CHURN	NO-CHURN
TARGET	CHURN	8 x 0 €	2 x -700 €
	NO-CHURN	20 x -50 €	70 x 0 €

(8 × 0€) + (2 x -700€) + (20 x -50€) + (70 × 0€) = **- 2 400€**

Accuracy = **0.78**

MODEL 2

		PREDICTED	
		CHURN	NO-CHURN
TARGET	CHURN	7 x 0 €	3 x -700 €
	NO-CHURN	16 x -50 €	74 x 0 €

(7 x 0€) + (3 x -700€) + (16 x -50€) + (74 x 0€) = **- 2 900€**

Accuracy = **0.81**

Multi categorical classification

Measures of performance

NOVA IMS Information Management School

Multi-class confusion matrix

		PREDICTED			
		Level A	Level B	Level C	Level D
TARGET	Level A	5	0	2	0
	Level B	0	6	1	0
	Level C	0	1	10	0
	Level D	0	0	2	3

Precision and Recall

		PREDICTED				
		Level A	Level B	Level C	Level D	**Recall**
TARGET	Level A	5	0	2	0	0.714
	Level B	0	6	1	0	0.857
	Level C	0	1	10	0	0.909
	Level D	0	0	2	3	0.600
Precision		1.00	0.857	0.667	1.000	

$$Precision(l) = \frac{\sum TP(l)}{\sum TP(l) + \sum FP(l)}$$

$$Recall(l) = \frac{\sum TP\ (l)}{\sum TP(l) + \sum FN(l)}$$

Average class accuracy (ACA)

		PREDICTED				
		Level A	Level B	Level C	Level D	**Recall**
TARGET	Level A	5	0	2	0	0.714
	Level B	0	6	1	0	0.857
	Level C	0	1	10	0	0.909
	Level D	0	0	2	3	0.600

$$ACA = \frac{1}{\frac{1}{|levels(t)|}\sum_{l \in levels(t)} \frac{1}{recall_l}}$$

$$= \frac{1}{\frac{1}{4}\left(\frac{1}{0.714} + \frac{1}{0.857} + \frac{1}{0.909} + \frac{1}{0.600}\right)} = \frac{1}{1.333} = 0.75$$

Logarithmic loss

$$Logarithmic\ Loss\ (Log\ Loss) = \frac{-1}{m}\sum_{j=1}^{m}\sum_{k=1}^{n} y_{jk} \times \log(p_{jk})$$

Penalizes incorrect predictions. The classification algorithm assigns a probability to each level of the sample, where:

$\boldsymbol{y_{jk}}$, indicates if sample i belongs to level j

$\boldsymbol{p_{jk}}$, indicates the probability that sample I belongs to level j

$\mathbf{n}$, indicates the number of levels

$\mathbf{m}$, indicates the number of instances

Result in the domain of [0, ∞]. Values close to 0 indicate greater accuracy

6.2

Logistic Regression

Modeling: Classification

Logistic regression

Classification: y = 0 or 1

Logistic regression: $0 \leq h_w(x) \leq 1$

$$g(z) = \frac{1}{1+e^{-z}}$$

$$h_w(x) = \frac{1}{1 + e^{-w^T x}}$$

If threshold is 0.5:

$$\begin{cases} h_w(x) \geq 0.5, y = 1 \\ h_w(x) < 0.5, y = 0 \end{cases}$$

Odds ratio are the base of logistic regression

$$odds\ ratio = \frac{P(1)}{1 - P(1)} = \frac{P(1)}{P(0)}$$

Odds ratio ≠ Likelihood an event will occur

Example: 1 in every 5 customers cancel their insurance policy after one year

Likelihood of canceling = $\frac{1}{5} = 0.2$ (a policy has a 20% likelihood of being canceled)

$$odds\ ratio(of\ canceling) = \frac{0.2}{1 - 0.2} = 0.25$$

$$odds\ ratio(of\ NOT\ canceling) = \frac{1 - 0.2}{0.2} = 4$$

In other words…the odds of a policy not being canceled is 4 times higher than of canceling

Probability calculation

$$odds\ ratio = \frac{P(1)}{1-P(1)} = w_o + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

$$P(target = 1) = \frac{1}{1 + e^{-(w_o + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n)}}$$

Logistic regression models

=

Models of the log of the odds ratio

Interpreting the models

- **Coefficient (β)**: weight per input variable, plus the constant (bias term)
- **Standard error of the coefficient (SE)**: measure of certainty of the coefficient. Smaller values imply a smaller level of uncertainty
- **Confidence interval (CI)**: the range of values the coefficient is expected to fall between (CI= β+SE)
- **Z statistic or Wald test**: the larger the z, the more likely the term associated to the coefficient is significant to the model ($z = \frac{\beta}{SE}$)
- **P(>$|z|$)**: values below 0.05 are considered a significant predictor (although there is no theoretical reason for making this inference)

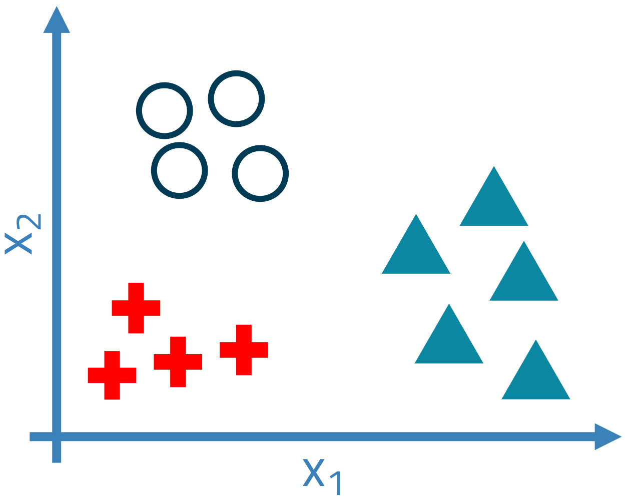
Practical considerations

- Because the model is the linear weighted sum of inputs, to improve models' accuracy, interactions of variables should be explicitly indicated
- Does not support missing values
- As in other numeric algorithms, when creating dummy variables for categorical columns, create a column for category level minus one (if there are 8 levels create only 7 dummy variables), this will avoid multicollinearity

Multinomial logistic regression

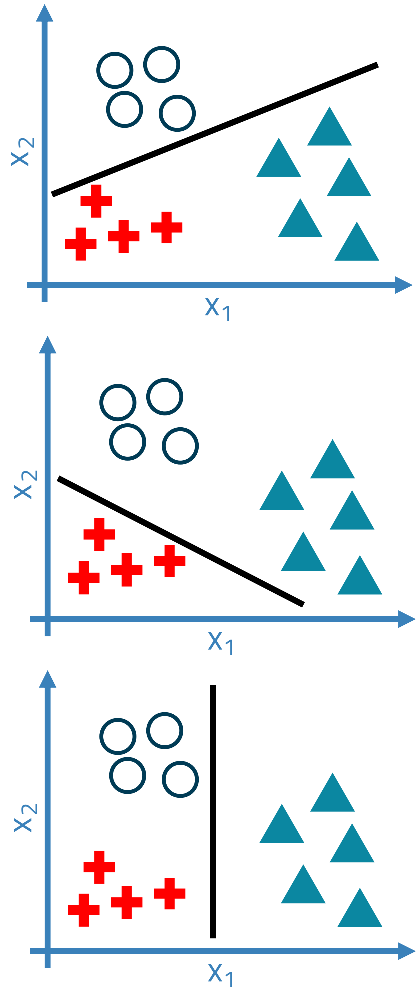
Logistic regression

One-vs-all



$$h_w^{(i)}(x) = P(\mathrm{y} = \mathrm{i}|\mathrm{x};\mathrm{w}) \quad (\mathrm{i} = 1,2,3)$$

$$P(Target = N) = \sum_{i=1}^{N-1} P(Target = i)$$



One-vs-all

1. Train a logistic regression classifier $h_w^{(i)}(x)$ for each class *i* to predict the probability that *y=I*
2. On a new input *x*, to make a prediction, pick the class *i* that maximizes:

$$\max_i h_w^{(i)}(x)$$

Application exercise

Logistic regression

Predicting the success of bank telemarketing

1. Copy from the datasets folder the dataset “bank-additional-full.csv”
2. Copy and open the Jupyter notebook “PredictBankTelemarketingSucess_LR.ipynb”
3. Follow the presentation of the notebook, answer the questions and explore the challenges

6.3

Support Vector Machine

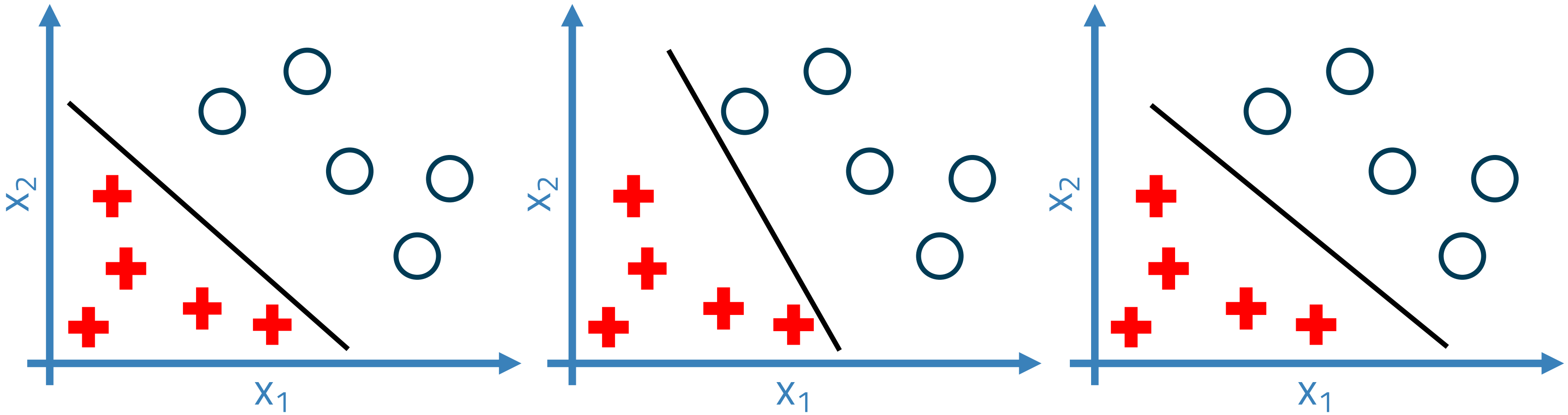
Modeling: Classification

SVM

- Known as “Large-margin” classifier:
  - Linear separable
  - Nonlinear separable
- Employs “Kernel” methods to create nonlinear classifiers

Goal

As logistic regression, its goal is to find the **best decision boundary** between classes



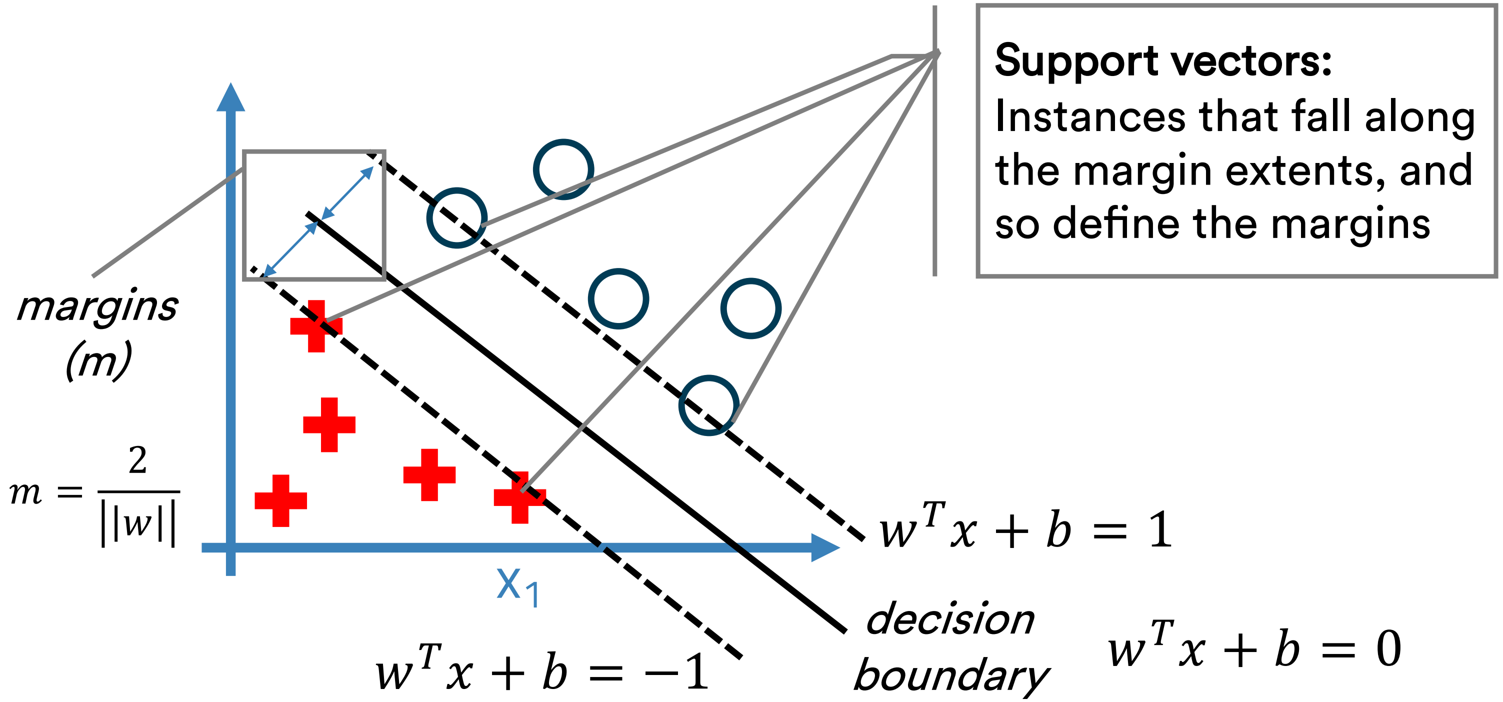
not good decision boundary

not good decision boundary

good decision boundary

Large-margin decision boundary

The decision boundary (a.k.a. “separating hyperplane”) should be as far way from the data of both classes as possible



Fiding the decision boundary

- Let $\{x_1,\ldots,x_n\}$ be a dataset and let $y_i \in \{1,-1\}$ be the class label of x_i
- The decision boundary (DB) should classify all points correctly: $y_i(w^T x_i + b) \geq, \forall_i$
- The decision boundary can be found by solving the constrained optimization problem:
  - Minimize $\frac{1}{2}||w||^2$
  - Subject to $y_i(w^T x_i + b) \geq, \forall_i$

$$||w|| = Euclidean\ norm\ of\ w = \sqrt{w[1]^2 + w[2]^2 + \cdots + w[m]^2}$$

Extension to non-linear DB

- Transform x_i to a higher dimensional space
- Apply a "Kernel function" (inner product) which is a similarity measure between objects
- Example of kernel functions: Linear kernel, Polynomial kernel, Gaussian radial basis kernel

Strengths and Weaknesses

<u>**Strengths**</u>

- Training is relatively easy
- Scales relatively well to high dimensional data
- Tradeoff between classifier complexity and error can be controlled explicitly
- A useful alternative to neural networks

<u>**Weaknesses**</u>

- Need to choose a “good” kernel function
- Do not provide probabilities (directly)

Application exercise

Support Vector Machine

Predicting the success of bank telemarketing

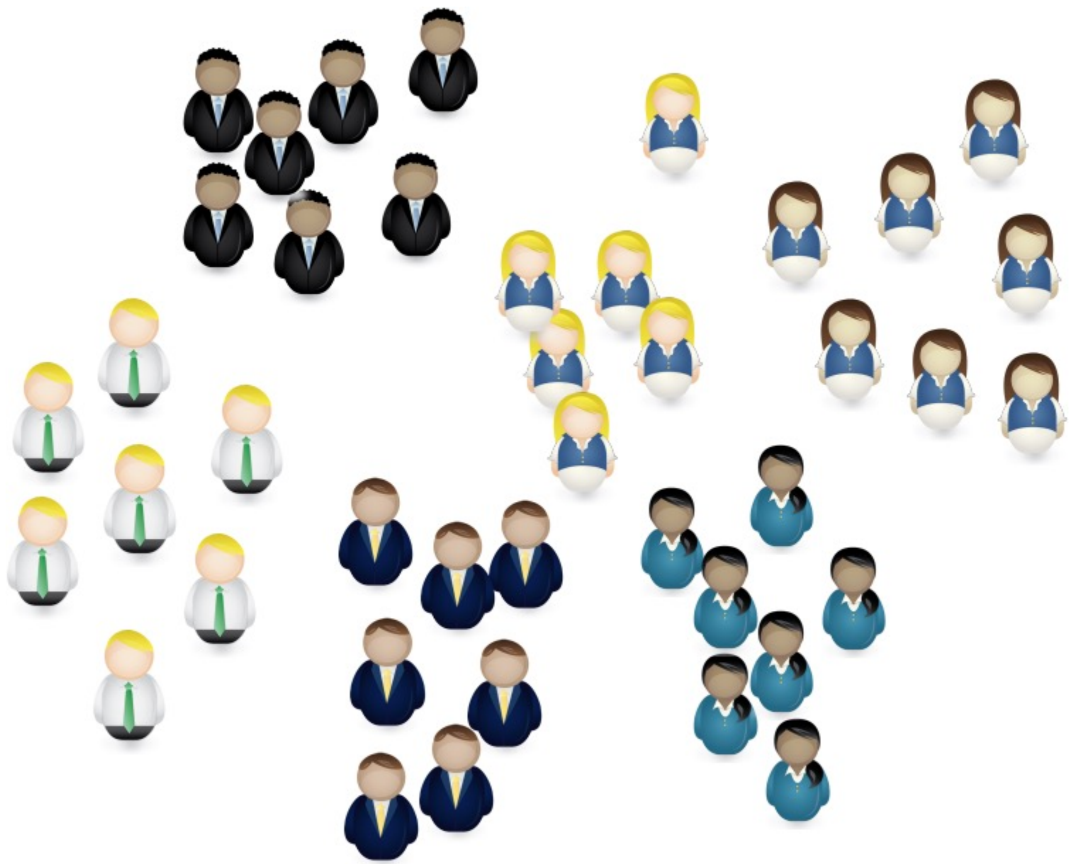
1. Copy from the datasets folder the dataset “bank-additional-full.csv”
2. Copy and open the Jupyter notebook “PredictBankTelemarketingSucess_SVM.ipynb”
3. Follow the presentation of the notebook, answer the questions and explore the challenges

6.4

K-Nearest Neighbor

Modeling: Classification

“Similar things exist in close proximity”



[https://medium.com/

NOVA IMS Information Management School

KNN algorithm

- **Non-parametric**: no assumption for underlying data distribution
- **Lazy**: does not require training (all training is done in testing)
- **Scans all data points**: which consumes time and memory
- **Versatile**: works both for classification and regression problems

How it works

1. Define k (the number of neighbors)
2. Calculate distance between instances
3. Find k closest neighbors
4. Calculate target:
    1. Classification: mode
    2. Regression: mean

Measures of distances between instances

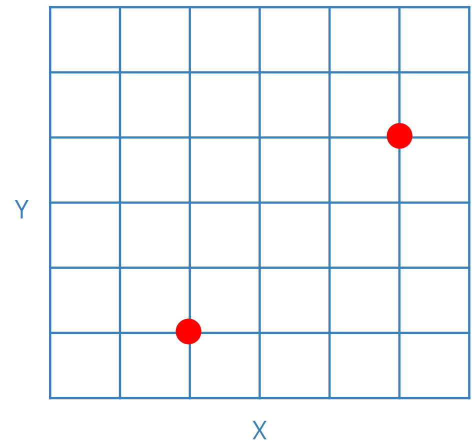
- Eucledian distance

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} = \sqrt{(2-5)^2 + (1-4)^2} = \sqrt{9+9} = 4.24$$

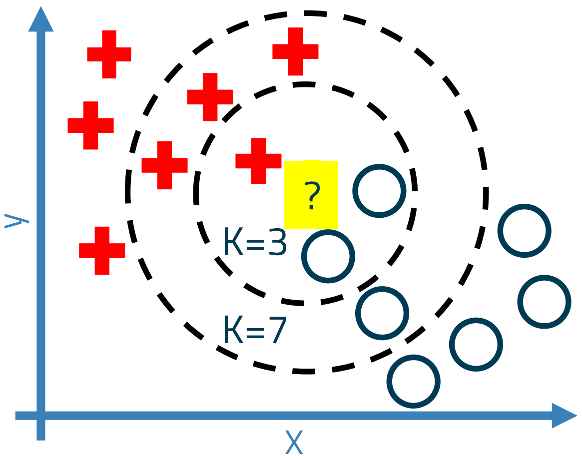
- Manhattan distance

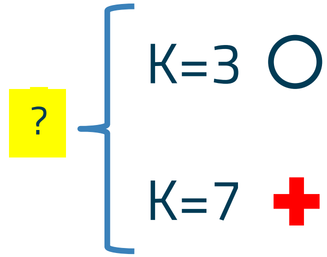
$\sum_{i=1}^{n}(x_i - y_i) = |2-5| + |1-4| = 6$

- Hamming distance
- Minkowski distance



K selection is important





NOTE: Some implementations use a weighted vote mechanism, meaning that near instances have a higher weight than distant ones

NOVA IMS Information Management School

Choosing the right K

- As K is decreased to 1, predictions become less stable
- Inversely, as K increases, predictions become stable due to majority voting/averaging. But, after a certain point, the number of errors will start to increase (meaning K was pushed too far)
- In classification, because of the “majority voting”, K should be an odd number

Strengths and Weaknesses

Strengths

- No need to train a model or tune many parameters
- Useful with nonlinear data
- Works both for classification and regression problems

Weaknesses

- Testing can be slow and consume time and memory
- Requires normalization [0,1]
- Becomes slower as the number of observations increase
- Not suitable for high dimensional data

Application exercise

K-Nearest Neighbor

Predicting the success of bank telemarketing

1. Copy from the datasets folder the dataset “bank-additional-full.csv”
2. Copy and open the Jupyter notebook “PredictBankTelemarketingSucess_KNN.ipynb”
3. Follow the presentation of the notebook, answer the questions and explore the challenges

6.5

Neural Networks

Modeling: Classification

Neural networks for classification

- Similar in everything to regression. The difference is that outputs are a prediction probability for a class

Application exercise

Neural networks

Predicting customers who will leave the bank in the following 6 months

1. Copy from the datasets folder the dataset "Bank_Churn_Modelling.csv"
2. Copy and open the Jupyter notebook "PredictBankChurn_NN.ipynb"
3. Follow the presentation of the notebook, answer the questions and explore the challenges

6.6

Naïve Bayes

Modeling: Classification

Naïve Bayes

- Based on Bayes theorem
- Is named "naïve" because it assumes inputs are independent

The probability of "B" being true given that "A" is true (*posterior*)

The probability of "A" being true *(prior)*

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

The probability of "A" being true given that "B" is true *(posterior)*

The probability of "B" being true (*prior*)

Probabilities review

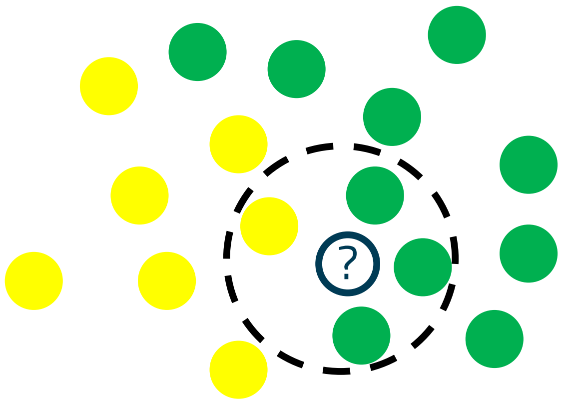
- $P(A) = \frac{\#events{=}true}{\#\ all\ events}$ $[0, 1]$
- OR: ∪,∨, or +
- AND: ∩,∧, or x
- $P(\bar{A}) = 1 - P(A)$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- $P(A \cap B) = P(A|B) \times P(B)$ or $P(B|A) \times P(A)$

Probabilities review: Flu example

- Facts:
  - People with flu have the same symptoms 90% of the time (headache and sore throat)
  - Statistics show only 5% of the population gets the flu every year
  - Every year, 20% of the population will have a headache and a sore throat
- Question: What is the probability of having the flu when you have a headache and a sore throat?
- Answer: $P(flu|symptoms) = \frac{P(symptoms|flu) \times P(flu)}{P(symptoms)} = \frac{0.9 \times 0.05}{0.2} = 0.225 = 22.5\%$

How it works

1. Calculate the probabilities for each class in the train dataset
2. Calculate distance to neighbors
3. Calculate probabilities to neighbors



$P(yellow)=\frac{7}{17}$ $P(green)=\frac{10}{17}$

$P'(?|green)= \frac{3}{10}$ (probability in vicinity)

$P'(?|yellow)= \frac{1}{7}$ (probability in vicinity)

$P''(?|green)=P(green) \times P'(?|green)= \frac{10}{17} \times \frac{3}{10} = \frac{30}{170}$

$P''(?|yellow)=P(yellow) \times P'(?|yellow)= \frac{7}{17} \times \frac{1}{7} = \frac{7}{119}$

? is Green

Interpretation

- Many implementations generate a list of probabilities per target level and input
- When the probability percentage of an input is higher than the target level distribution, it means the feature is important

Example (Target_balanced)	Input1=1	Input1=2	Input1=3	Input1=4
Counts Target_0	1268	510	382	266
Counts Target_1	930	530	471	494
Percentage Target_1	42.3%	51.0%	55.2%	65.0%

Strengths and Weaknesses

Strengths

- Easy to understand
- Does not require normalization
- Models are interpretable

Weaknesses

- Requires all inputs to be categorical (some implementations will bin the data)
- Does not find interactions between inputs. Interactions need to be specifically engineered
- Susceptible to high correlated variables
- Requires more data as the number of inputs get larger

NOVA IMS Information Management School

Application exercise

Naïve Bayes

Predicting customers who will leave the bank in the following 6 months

1. Copy from the datasets folder the dataset "Bank_Churn_Modelling.csv"
2. Copy and open the Jupyter notebook "PredictBankChurn_NB.ipynb"
3. Follow the presentation of the notebook, answer the questions and explore the challenges

6.7

Decision Tree

Modeling: Classification

Decision tree for classification

- Similar in everything to regression. The difference is that outputs are a prediction probability for a class
- In regression, the output is based the mean response of the observations falling in the region of the tree
- In classification, the output is based in the mode response of the observations falling in the region of the tree

NOVA IMS Information Management School

Application exercise

Decision tree

Predicting customers who will leave the bank in the following 6 months

1. Copy from the datasets folder the dataset "Bank_Churn_Modelling.csv"
2. Copy and open the Jupyter notebook "PredictBankChurn_DT.ipynb"
3. Follow the presentation of the notebook, answer the questions and explore the challenges

Questions?

Machine Learning for Marketing

Acreditações e Certificações

Instituto Superior de Estatística e Gestão da Informação
Universidade Nova de Lisboa