7th Lecture: Classification Part II

Martin Korytak, Josef Svec, Anna Strobova

6.6.2023

Your team today

Anna

Data Scientist,

Merck Research Labs IT

Josef

Data Scientist in

Workday

Martin

Data Scientist in

Workday

czechitas

Today’s structure

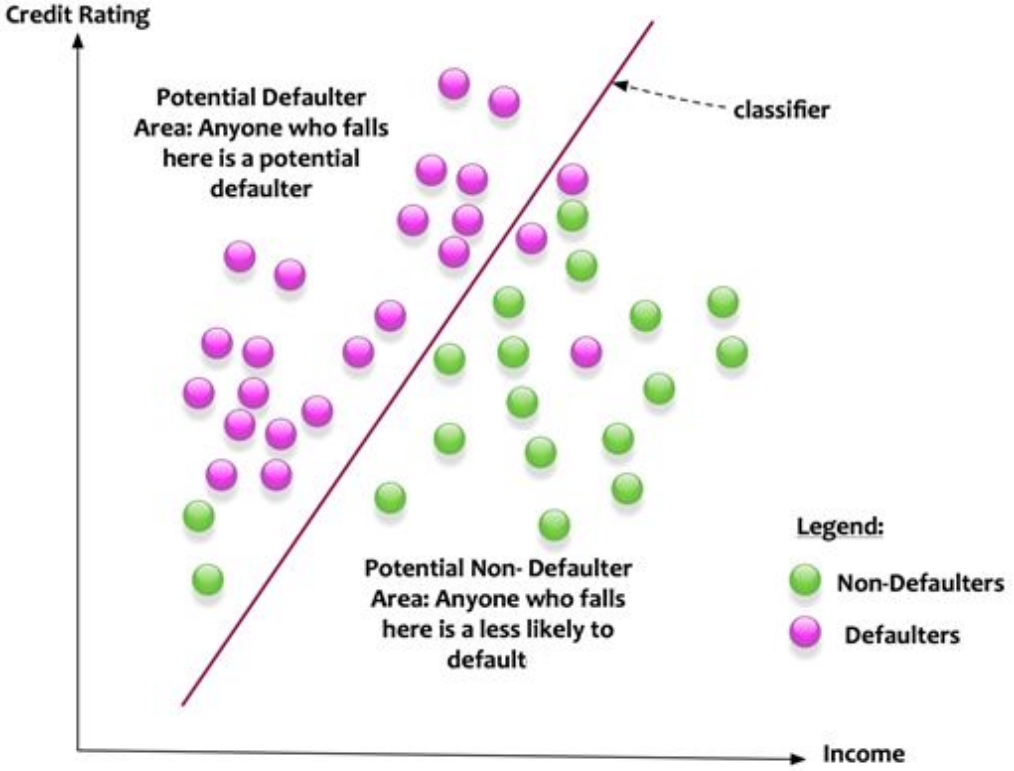
1. Decision tree and Random forest used in classification
2. k-nearest neighbors
3. TBD
4. TBD
5. TBD
6. TBD

Regression vs. Classification

What's the difference in these examples?

1. Email spam detection
2. Predict temperature based on various characteristics (humidity, wind speed)
3. Client risk prediction for loans
4. Estimate of your apartment price when selling it

Classification



Credit Default = a binary variable!

czechitas

Classification

CustomerID	Income	Education	Age	Default
2343	50 000	17	35	No
1213	35 000	15	32	Yes
4533	40 000	15	53	No
4563	100 000	19	51	No
7554	50 000	18	28	No
6465	27 500	13	25	Yes
7453	34 000	13	32	No
6775	72 000	18	43	No
4643	50 000	19	47	No
6886	48 000	19	37	?
8668	62 500	21	39	?
8765	78 000	23	46	?
9797	23 000	12	29	?

Labeled Data

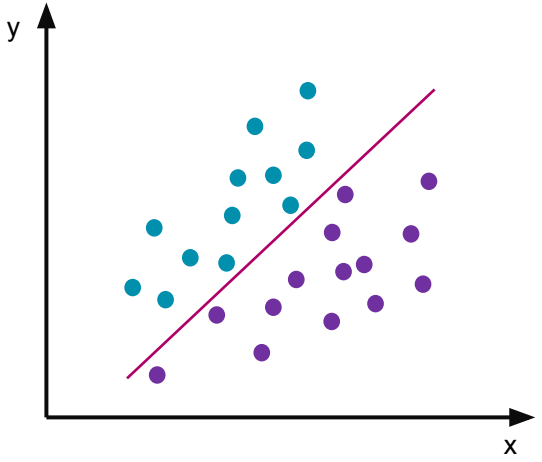
Unlabeled Data

Classification

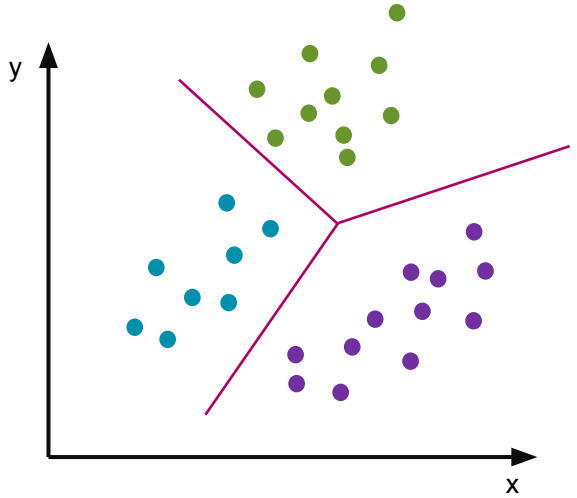
- Discrete response (instead of continuous)
- Model evaluation - accuracy, F1 score, sensitivity, etc. (instead of R^2)
- Dependent variable can be binary or multi-class (with special case of multi-label)
- Supervised learning
- Structured or unstructured data

Types of Classifications

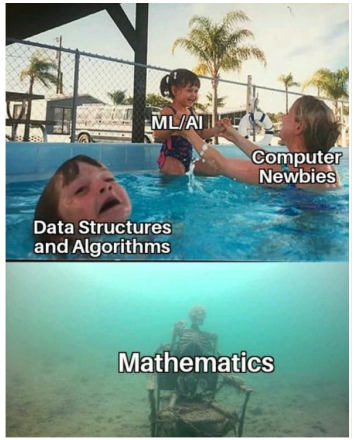
Binary



Multi-Class



Multi-Label

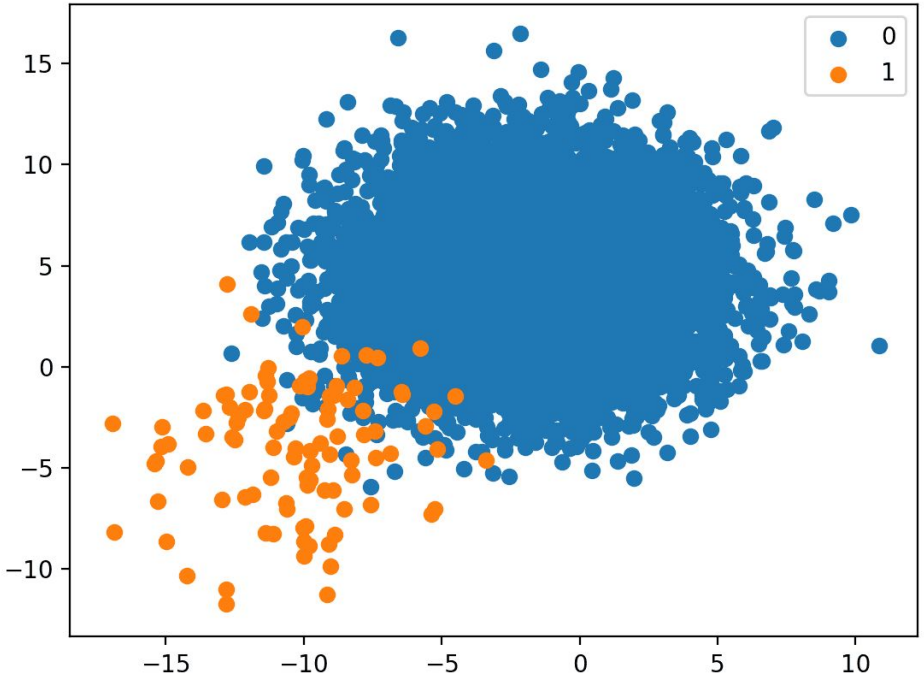


Labels:

- ☑ Child
- ☒ Snow
- ☑ Water
- ☑ Tree
- ☒ Bike

czechitas

Imbalanced Classifications



Always check
the distribution
Cannot be ignored!

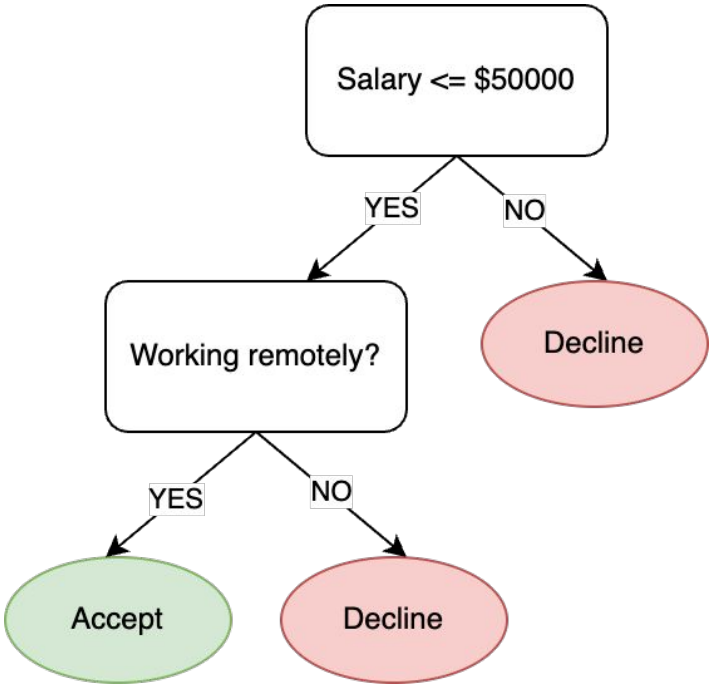
czechitas

Today’s structure

1. Decision tree and Random forest used in classification
2. k-nearest neighbors
3. TBD
4. TBD
5. TBD
6. TBD

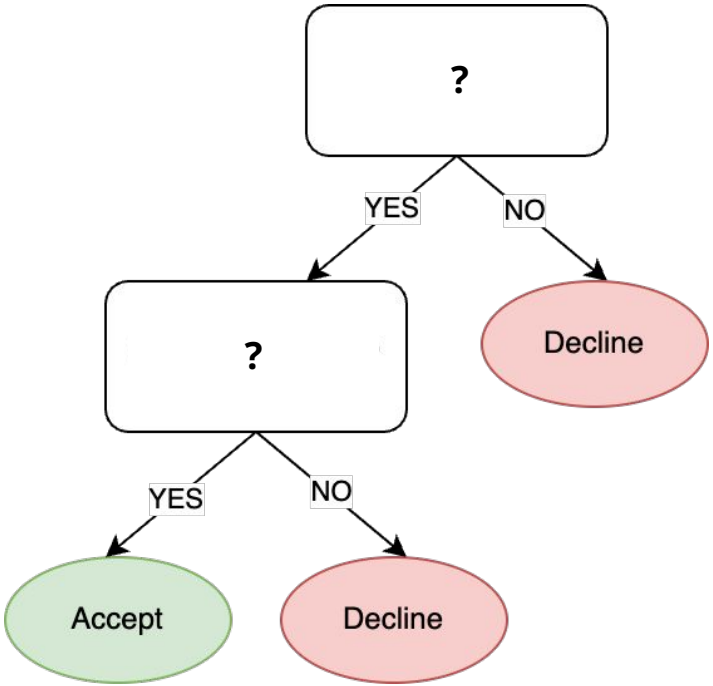
Classification tree example

	Working Remotely	Salary	Label
0	False	51777	Decline
1	True	36646	Decline
2	True	53801	Accept
3	False	56105	Decline
4	False	55597	Decline
5	False	60807	Accept
6	False	58339	Decline
7	True	54591	Accept
8	True	49298	Decline
9	True	33390	Decline



Building a decision tree - how?

	Working Remotely	Salary	Label
0	False	51777	Decline
1	True	36646	Decline
2	True	53801	Accept
3	False	56105	Decline
4	False	55597	Decline
5	False	60807	Accept
6	False	58339	Decline
7	True	54591	Accept
8	True	49298	Decline
9	True	33390	Decline



czechitas

Building a decision tree - regression

What condition do we start with?

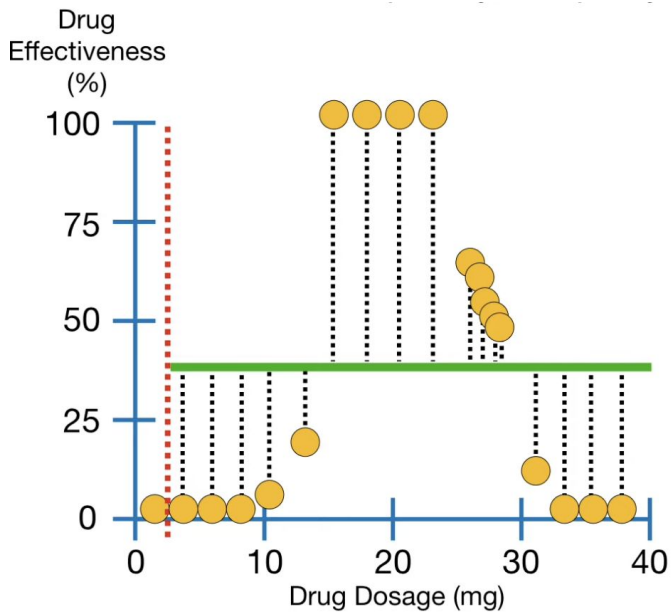
We try all possible thresholds, and **see which threshold gives us the lowest prediction error.**

Error = Predicted value - Observed value

Do you remember?

czechitas

Building a tree - predicting effectiveness by dosage



We want to calculate the sum of all the residuals.

$$(0 - 0)^2 + (0 - 38.8)^2 + (0 - 38.8)^2 + (0 - 38.8)^2$$
$$+ (5 - 38.8)^2 + (20 - 38.8)^2 + (100 - 38.8)^2$$
$$+ (100 - 38.8)^2 + \ldots + (0 - 38.8)^2$$

= 27 469

Do you remember?

czechitas

Building a decision tree - Gini impurity

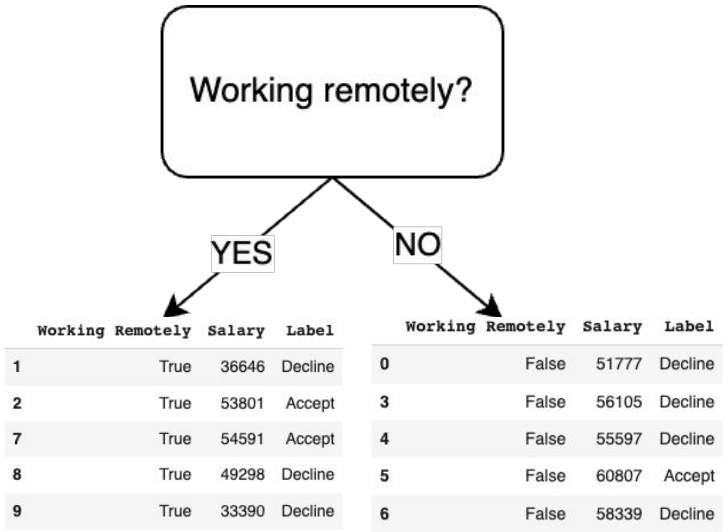
We try all possible decision rules, and **see which rule gives us the lowest impurity.**

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Gini impurity - Example

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

node: Working remotely = True



	Working Remotely	Salary	Label
1	True	36646	Decline
2	True	53801	Accept
7	True	54591	Accept
8	True	49298	Decline
9	True	33390	Decline

	Working Remotely	Salary	Label
0	False	51777	Decline
3	False	56105	Decline
4	False	55597	Decline
5	False	60807	Accept
6	False	58339	Decline

total examples = 5
probability of decline = ⅗
probability of accept = ⅖

Gini impurity = 1 - (⅗ * ⅗ + ⅖ * ⅖) = 1 - 0.36 - 0.16 = 0.48

czechitas

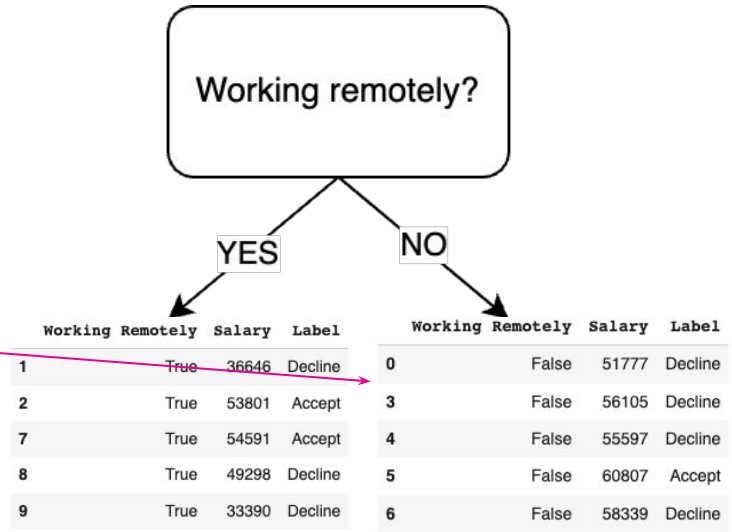
Gini impurity - example

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

node: Working remotely = False

total examples = 5
probability of decline = ⅘
probability of accept = ⅕

Gini impurity = 1 - (⅘ * ⅘ + ⅕ * ⅕) = 1 - 0.64 - 0.04 = 0.32

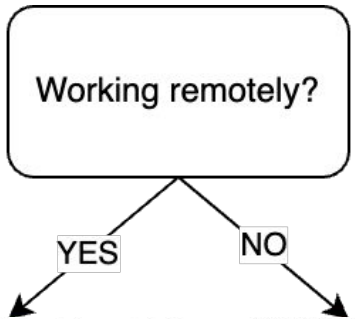


	Working Remotely	Salary	Label
1	True	36646	Decline
2	True	53801	Accept
7	True	54591	Accept
8	True	49298	Decline
9	True	33390	Decline

	Working Remotely	Salary	Label
0	False	51777	Decline
3	False	56105	Decline
4	False	55597	Decline
5	False	60807	Accept
6	False	58339	Decline

Gini impurity - example

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$



	Working Remotely	Salary	Label
1	True	36646	Decline
2	True	53801	Accept
7	True	54591	Accept
8	True	49298	Decline
9	True	33390	Decline

	Working Remotely	Salary	Label
0	False	51777	Decline
3	False	56105	Decline
4	False	55597	Decline
5	False	60807	Accept
6	False	58339	Decline

node: Working remotely

Weighted Gini impurity based on “Working remotely” flag is: 5/10 * 0.48 + 5/10 * 0.32 = 0.24 + 0.16 = 0.4

We can repeat the same process with “Salary” feature!

czechitas

Gini impurity - summary

How to split a decision tree using Gini Impurity?

1. For each split, individually **calculate the Gini Impurity of each child node**

1. Calculate the Gini Impurity of each split as the **weighted average** Gini Impurity of child nodes

1. Select the split with the **lowest value of Gini Impurity**

1. Until you achieve **homogeneous nodes**, repeat steps 1 to 3

Quiz 1 Which statement is false?

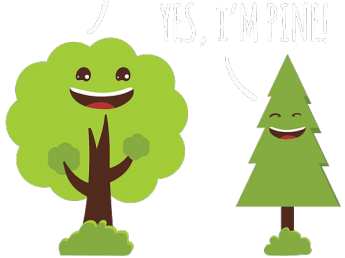
1. When decision tree is trained, all thresholds per each explanatory variable are tested as a potential condition.

2. The Gini impurity values range from 0 to 1.

3. When constructing a decision tree, the goal is to maximize the Gini impurity.

4. It is good to control the minimum number of observations in the leaf node when specifying the model.

Quiz 1 Which statement is false?



1. When decision tree is trained, all thresholds per each explanatory variable are tested as a potential condition.

2. The Gini impurity values range from 0 to 1.

3. **When constructing a decision tree, the goal is to maximize the Gini impurity.**

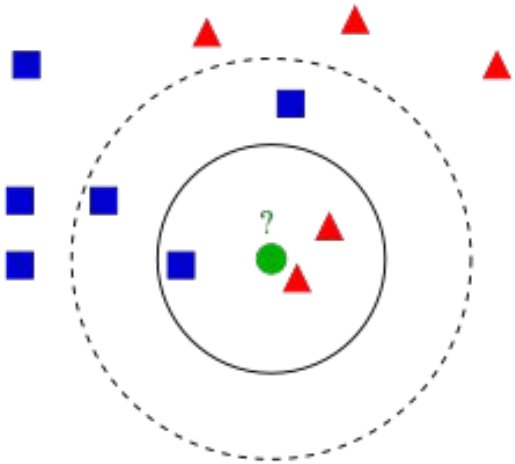
4. It is good to control the minimum number of observations in the leaf node when specifying the model.

Today’s structure

1. Decision tree and Random forest used in classification
2. k-nearest neighbors
3. TBD
4. TBD
5. TBD
6. TBD

k-nearest neighbors

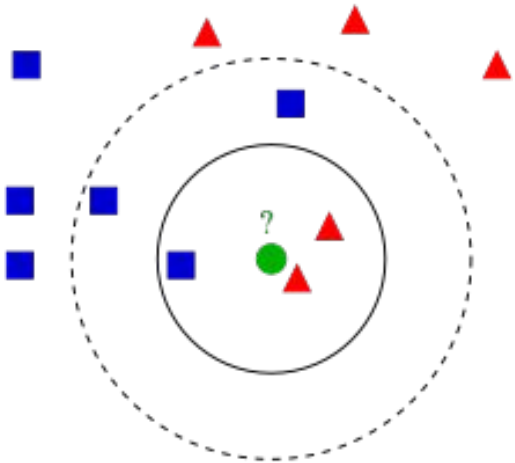
- A powerful classification algorithm used in pattern recognition.
- k-nearest neighbors stores all available data points and classifies new unseen points based on a **similarity (distance) function**.
- A **non-parametric lazy** learning algorithm.



czechitas

k-nearest neighbors - classification

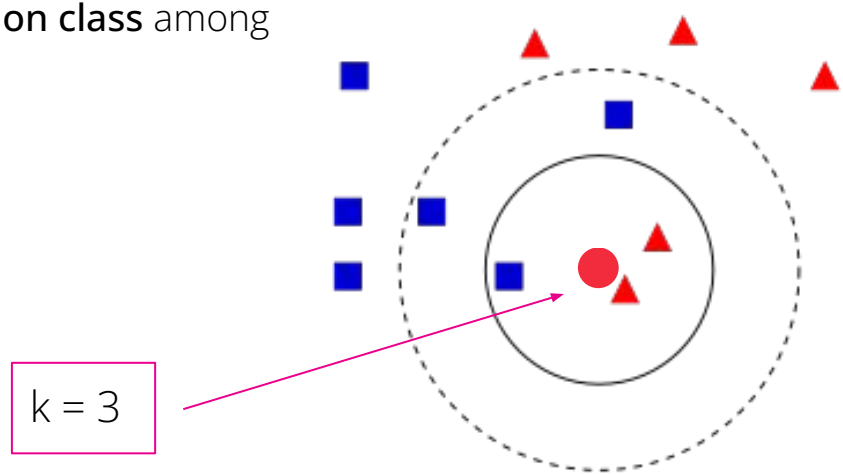
- An unseen point is classified by a **majority voting** for its neighbor classes.
- The point is then **assigned to the most common class** among its k nearest neighbors (based on distance).



czechitas

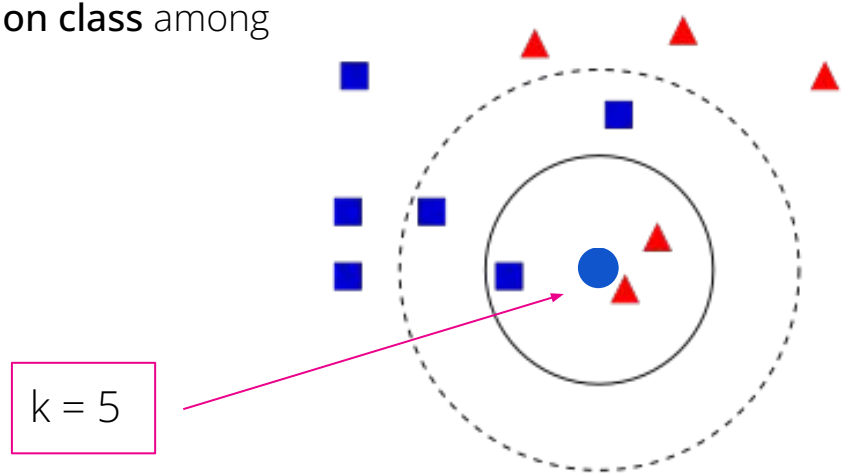
k-nearest neighbors - classification

- An unseen point is classified by a **majority voting** for its neighbor classes.
- The point is then **assigned to the most common class** among its k nearest neighbors (based on distance).



k-nearest neighbors - classification

- An unseen point is classified by a **majority voting** for its neighbor classes.
- The point is then **assigned to the most common class** among its k nearest neighbors (based on distance).



k-nearest neighbors - distance function

- Euclidean

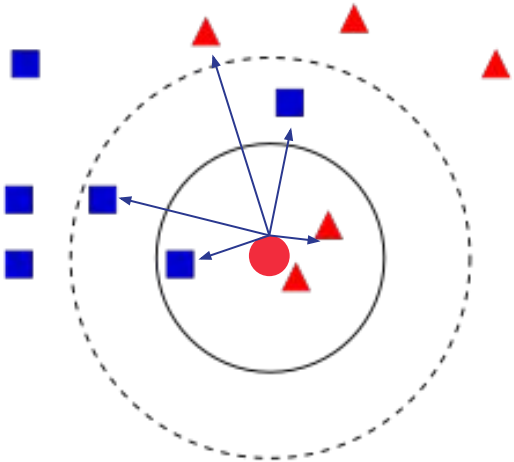
$$Euclidean = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Manhattan

$$Manhattan = \sum_{i=1}^{n} |x_i - y_i|$$

- Minkowski

$$Minkowski = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

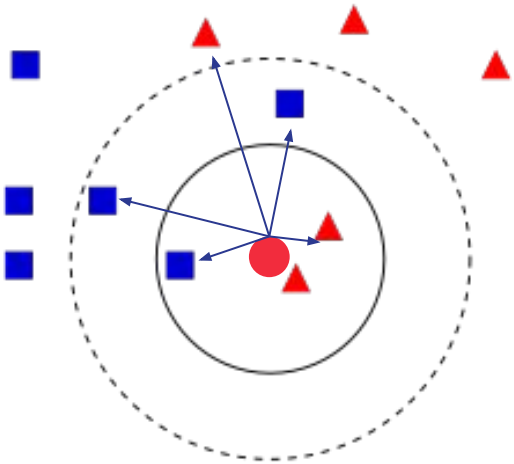


k-nearest neighbors - distance function

- Calculate the distance between new unseen point and all examples in training set.
- Example of Euclidean distance between X and Y in 3D:
  - X = [0, 1, 1]
  - Y = [2, 1, 1]
- The Euclidean distance between X and Y is defined as:

$$Euclidean = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

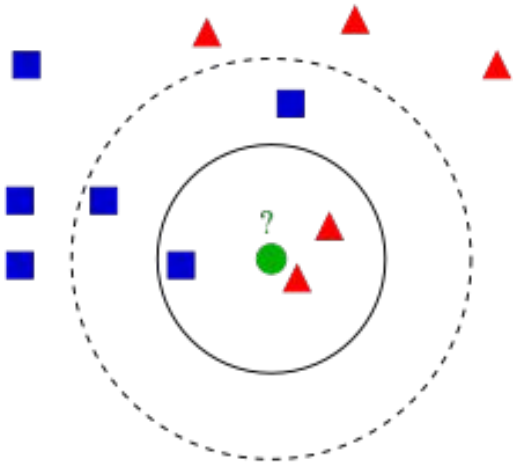
dist = sqrt((0 - 2) * (0 - 2) + (1 - 1) * (1 - 1) + (1 - 1) * (1 - 1)) = sqrt(4 + 0 + 0) = 2



czechitas

k-nearest neighbors - algorithm

- All data points lie in an n-dimensional feature space.
- Each data point is represented with a set of numerical features.
- Each of the training data consists of a set of vectors and a class label associated with each vector.
- Classification is done by comparing feature vectors of different k nearest points.
- Select the k-nearest data points to new unseen point from the training set.
- Assign the point to the most common class among its k-nearest neighbors.



k-nearest neighbors - example

	Salary	Working Remotely_False	Working Remotely_True	Label
0	51777	1	0	Decline
1	36646	0	1	Decline
2	53801	0	1	Accept
3	56105	1	0	Decline
4	55597	1	0	Decline
5	60807	1	0	Accept
6	58339	1	0	Decline
7	54591	0	1	Accept
8	49298	0	1	Decline
9	33390	0	1	?

	Distance
0	18387.000054
1	3256.000000
2	20411.000000
3	22715.000044
4	22207.000045
5	27417.000036
6	24949.000040
7	21201.000000
8	15908.000000
9	0.000000

k-nearest neighbors - example

	Salary	Working Remotely_False	Working Remotely_True	Label
0	51777	1	0	Decline
1	36646	0	1	Decline
2	53801	0	1	Accept
3	56105	1	0	Decline
4	55597	1	0	Decline
5	60807	1	0	Accept
6	58339	1	0	Decline
7	54591	0	1	Accept
8	49298	0	1	Decline
9	33390	0	1	?

	Distance
0	18387.000054
1	3256.000000
2	20411.000000
3	22715.000044
4	22207.000045
5	27417.000036
6	24949.000040
7	21201.000000
8	15908.000000
9	0.000000

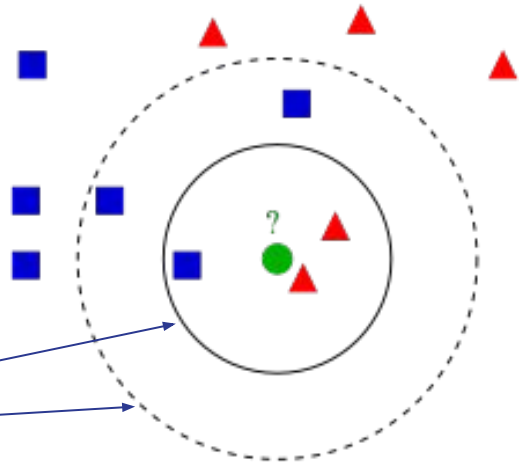
Decline!

NOTE: We used Euclidean distance and 3 nearest neighbors

czechitas

k-nearest neighbors - how to choose k?

- If k is too small, the algorithm is sensitive to noise.
- Usually larger k works well. But too large k may include majority points from other classes which results in misclassification.
- Rule of thumb is **k < sqrt(N)**, where N is number of data points in the training data set.



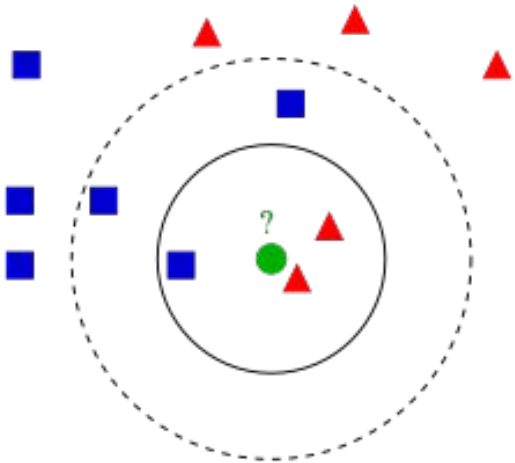
The difference of labeling based on 3-nn and 5-nn.

czechitas

k-nearest neighbors - feature normalization

- Distance between neighbors could be **dominated** by some attributes with relatively large numbers.
  e.g., "Salary" feature in our previous example
- Arises when two features are in different scales.
- Therefore, it is essential to **normalize** those features to the same scale, usually between 0 and 1.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$



czechitas

k-nearest neighbors - example with normalization

	Salary	Working Remotely_False	Working Remotely_True
0	0.626257	1.0	0.0
1	0.000000	0.0	1.0
2	0.710029	0.0	1.0
3	0.805389	1.0	0.0
4	0.784363	1.0	0.0
5	1.000000	1.0	0.0
6	0.897852	1.0	0.0
7	0.742726	0.0	1.0
8	0.523654	0.0	1.0
9	-0.134763	0.0	?

	Distance
0	1.605974
1	0.134763
2	0.844791
3	1.698200
4	1.686651
5	1.813198
6	1.751083
7	0.877489
8	0.658416
9	0.000000

k-nearest neighbors - example with normalization

	Salary	Working Remotely_False	Working Remotely_True	Label
0	0.626257	1.0	0.0	Decline
1	0.000000	0.0	1.0	Decline
2	0.710029	0.0	1.0	Accept
3	0.805389	1.0	0.0	Decline
4	0.784363	1.0	0.0	Decline
5	1.000000	1.0	0.0	Accept
6	0.897852	1.0	0.0	Decline
7	0.742726	0.0	1.0	Accept
8	0.523654	0.0	1.0	Decline
9	-0.134763	0.0	1.0	?

	Distance
0	1.605974
1	0.134763
2	0.844791
3	1.698200
4	1.686651
5	1.813198
6	1.751083
7	0.877489
8	0.658416
9	0.000000

Decline?!

NOTE: We used Euclidean distance and 3 nearest neighbors

k-nearest neighbors - summary

Advantages

- Very simple and intuitive algorithm
- Can be applied to any data from any distribution
- Good classification if the number of samples is large enough

Disadvantages

- Takes more time to classify a new example (calculate and compare distance of new unseen point to all other points)
- Choosing right k may be tricky
- Need large number of samples for good accuracy

Quiz 2 Which statement is false?

1. We should perform feature scaling, so there's no dominant attribute.

2. The choice of distance metric, such as Euclidean or Manhattan, has an impact on the KNN algorithm.

3. KNN is a non-parametric algorithm, meaning it does not make any assumptions about the underlying data distribution.

4. KNN is computationally efficient for large datasets due to its lazy learning approach.

Quiz 2 Which statement is false?

1. We should perform feature scaling, so there's no dominant attribute.

2. The choice of distance metric, such as Euclidean or Manhattan, has an impact on the KNN algorithm.

3. KNN is a non-parametric algorithm, meaning it does not make any assumptions about the underlying data distribution.

4. **KNN is computationally efficient for large datasets due to its lazy learning approach.**

Today’s structure

1

2

3 Support Vector Machine

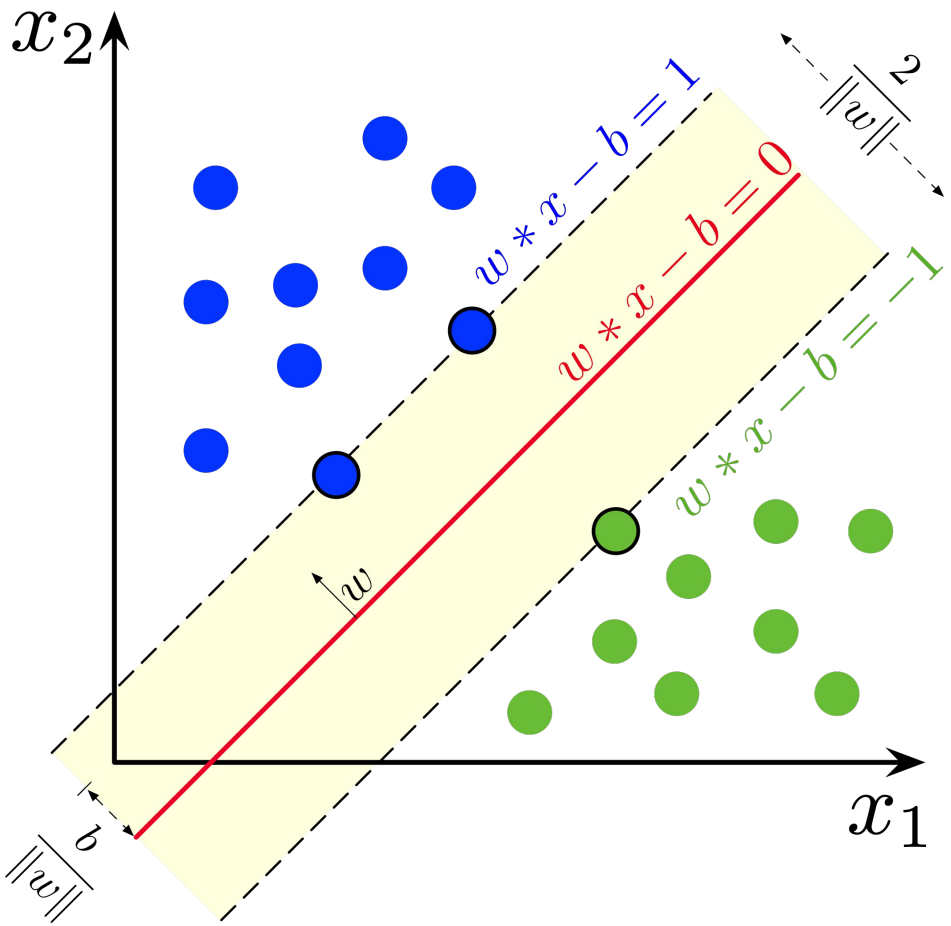
4


czechitas
SVM
support vector machine

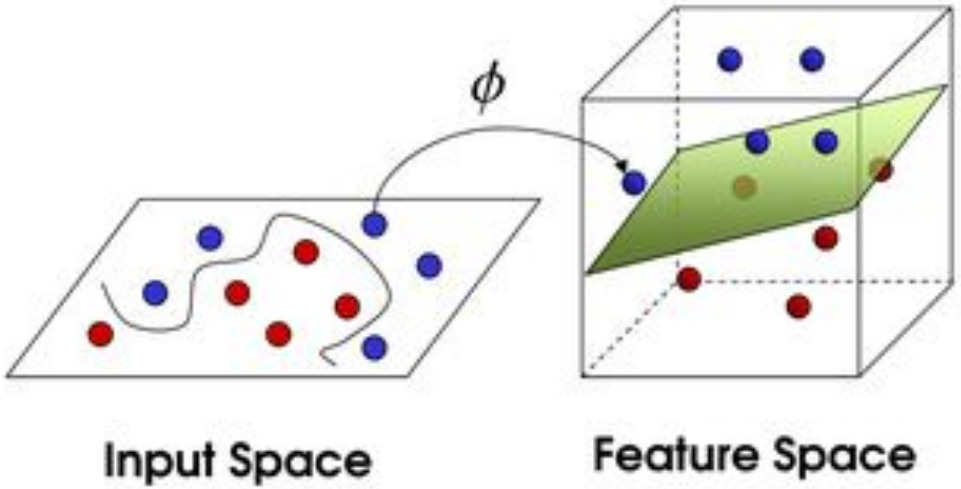
Support vector machine

classification algorithm

The primary goal is to find the optimal hyperplane that distinctly classifies the data points in an N-dimensional space(N — the number of features). In two-dimensional space, this hyperplane is a line dividing a plane in two parts where each class laid on either side.



Kernel trick



Today’s structure

1

2

3

4 Naïve Bayes Classifier

Problem Introduction

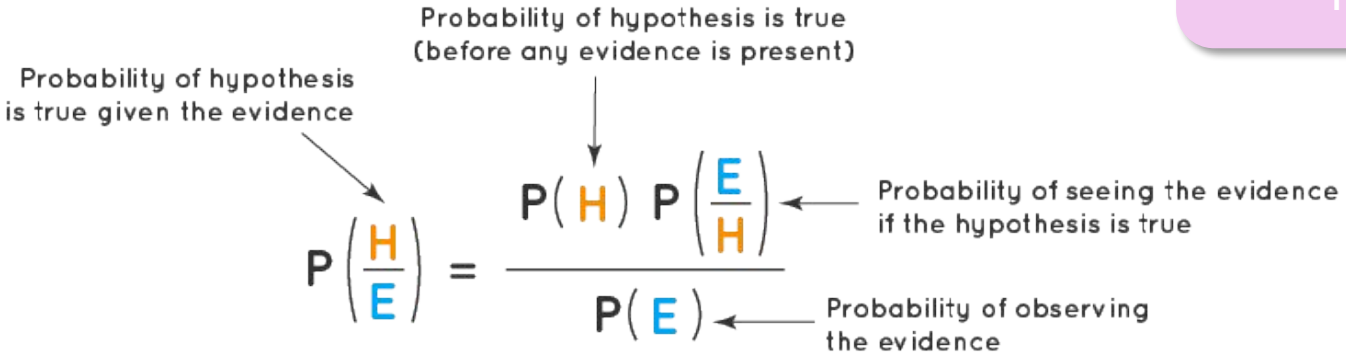
Weather	Playing Golf
Sunny	No
Sunny	No
Overcast	Yes
Rainy	Yes
Rainy	Yes
Rainy	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rainy	No

Predicting whether person will play golf based on weather forecast

Problem Statement – Is Adam going to play golf when it is overcast?

Bayes' Theorem

Main Concept: Conditional Probability



Conditional probability is the likelihood of an outcome occurring, based on a previous outcome having occurred in similar circumstances.

czechitas

Why **Naïve** Bayes Classifier?

Main Assumption

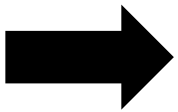
Features (predictor variables) are independent

→ one particular feature does not affect the other.

Practical Example (Categorical Features)

Step 1: Create Frequency Table

Weather	Playing Golf
Sunny	No
Sunny	No
Overcast	Yes
Rainy	Yes
Rainy	Yes
Rainy	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rainy	No



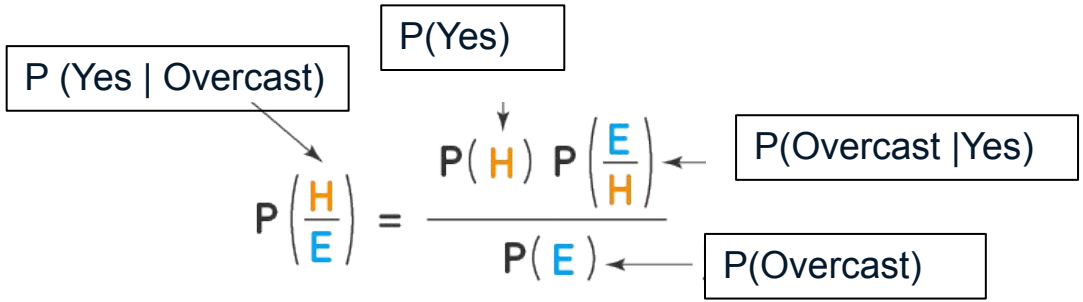
Frequency Table

Weather	No	Yes
Overcast		4
Sunny	2	3
Rainy	3	2
Total	5	9

Practical Example (Categorical Features)

Step 2: Map what we need to calculate

Problem Statement – Is Adam going to play golf when it is overcast?



$$P\left(\frac{H}{E}\right) = \frac{P(H)\,P\left(\frac{E}{H}\right)}{P(E)}$$

czechitas

Practical Example

Step 3: Likelihood Table

Weather	No	Yes	Probability
Overcast		4	4/14 = 0.29
Sunny	2	3	5/14 = 0.36
Rainy	3	2	5/14 = 0.36
Total	5	9	
	5/14 = 0.36	9/14 = 0.64	

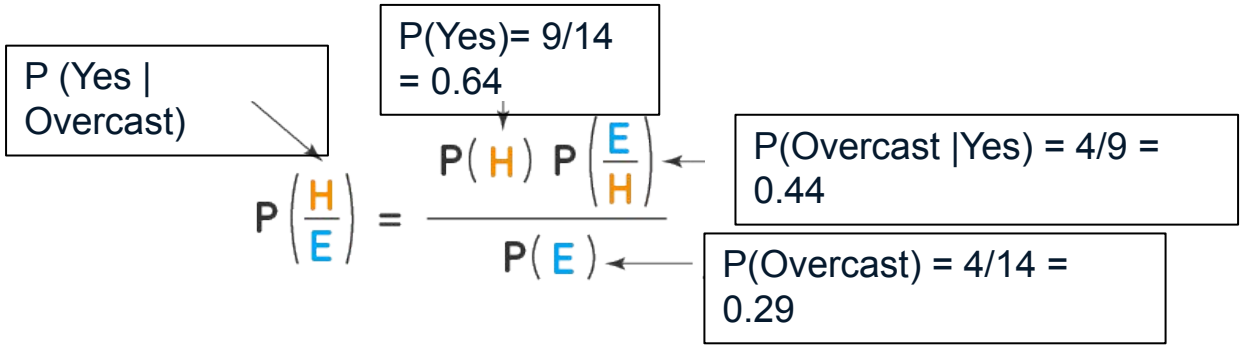
Weather	No	Yes	Probability for NO	Probability for YES
Overcast	0	4	0/5 = 0	4/9 = 0.44
Sunny	2	3	2/5 = 0.4	3/9 = 0.33
Rainy	3	2	3/5 = 0.6	2/9 = 0.22
Total	5	9		

Total Number of Datapoints - 14

Practical Example

Step 4: Calculating final probability

Try it yourself first!

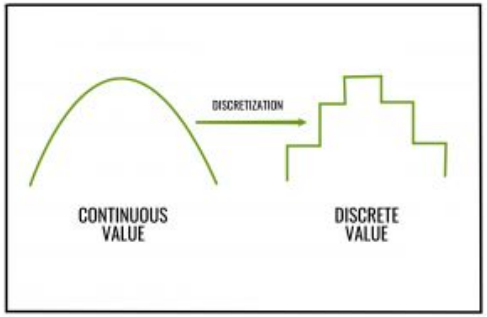


P (Yes | Overcast) = 0.44 * 0.64 / 0.29 = 0.98

czechitas

How to handle numerical features?

- Discretize continuous feature, which will give us discrete category (interval)



- Calculate likelihood using assumed probability distribution

Advantages and Disadvantages of Naïve Bayes

+

Advantages

- Fast
- Can be used to solve multi-class predictions problems
- If assumptions are fulfilled -> usually performs well
- Does not require a lot of training data

-

Disadvantages

- Naive Bayes assumes that all predictors (or features) are independent, which rarely happens in reality. -> Could be solved by Semi-naïve bayes based on expert input
- ‘Zero-frequency problem’
- For numerical features

Thank you for your attention.

We are looking forward to the next lecture!