Universität Zürich[UZH]

Soziologisches Institut

Data Analysis – Advanced Statistics with Python

Dr. Julia Jerke

jerke@soziologie.uzh.ch

Thursday, 12.15pm – 13.45pm, AND 2.46

Session 10 – Introduction to machine learning

Agenda

1. Basics of machine learning
2. Hands on with a KNN application

1. Basics of machine learning

What is machine learning?

In its simplest form:

➢ ***Learning from known data to make inferences about unknown data***

- Machine learning (ML) encompasses a collection of methods that detect patterns and relationships within data to make predictions
- The main purpose is *prediction* and *pattern detection*
- Usually that involves developing a model by using existing data to, for instance, predict the outcome for new data points
  - What is meant by existing **data**?
    - That is the most essential part in machine learning: to learn, a model needs high quality data
    - Compiling that data can involve high effort
  - What is a **model**?
    - A mathematical description of the assumed relationships between some variables
    - Think for example of a linear model that assumes and models a linear relationship between a target variable Y and some predictors $X_1, X_2, \ldots, X_p$

Machine learning examples

➢ **Machine learning is a very powerful tool since it can be applied in a wide range of situations**

Classic examples

- *Spam filter*: training a model in such way that it can predict whether an incoming mail is spam
- *Image and voice recognition*
- *Weather forcasting*
- *Medical diagnosis:* predicting cancer from body scans

Examples from the social sciences

- Election forcasting
- Detection of election fraud
- Quantitative text analysis
- Analysis of dynamic social media data

Supervised versus unsupervised learning

- The main difference of ML methods lies between **supervised and unsupervised learning**

	Supervised learning	Unsupervised learning
Purpose	**Prediction** and **classification**: predicting a certain outcome from given input	**Knowledge-discovery**: data transformation and pattern detection
Data	The **data is labeled**, meaning we know the correct target outcome of the training data	The **data is unlabeled**, hence we only no the input data but no target outcome
Complexity	Relatively simple, since the desired outcome is known for the training data	Can be very complex since we are working with large amounts of unclassified data
Performance evaluation	Easy to measure the performance of the ML model	Difficult to measure the performance of the ML model
Example	Identifying handwritten numbers	Identifying topics in a set of comparable texts

Training and test data

- **One of the core principles in machine learning is the use of different data to train and to test your model**
- What would happen if we were to use the same data for learning and testing?
  - Training a model means that the model detects and learns patterns in the data
  - By that, the model becomes an “expert” for the training data
  - Therefore, the model will be quite good in predicting the same data that it already used to learn
  - But then we will know little about how it would perform in predicting unknown data
- Hence, we split our data in two parts:
  - **Training data**: that data is used to build the model, it will learn the patterns within this subset
  - **Test data**: we apply the trained model to the test data
    - In the case of supervised learning we can make predictions and compare them to the know labels of the data
    - In the case of unsupervised learning we can compare the patterns found in the training data with patterns in the test data
- Often training and test data have a ratio of around 3:1

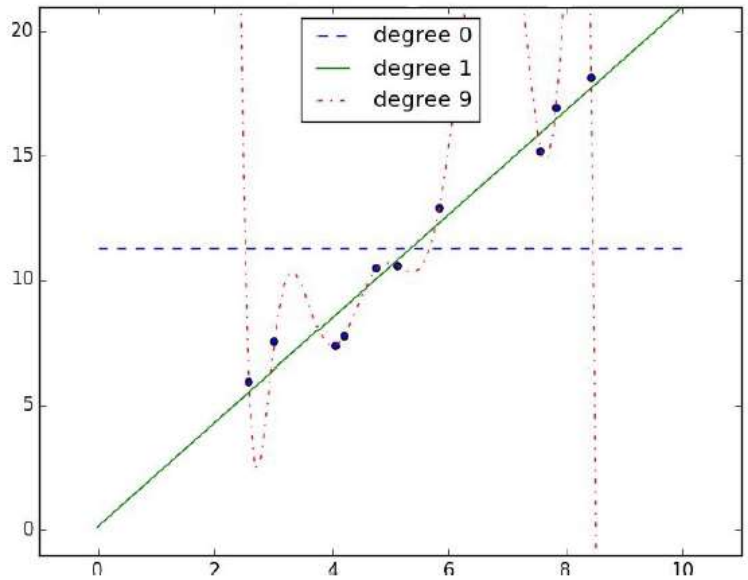
Training and test data

Example:

- Just assume that we want to build a linear model that predicts the income of a person from their age, gender, years of education and occupation
- Therefore, we compile a data set with people for whom we know their income and the other information
- We run a linear regression with the full data set and use the regression equation to predict the income of a new person for which we do not know the income yet
- But:
  - How do we know how good our linear model is at predicting new data in general?
  - How do we know how accurate our specific prediction is?
- Using the R square for instance only tells us how good our model explains the income in our known data
- When we split our data in training and test data we can estimate the linear model with the training data and test the predictive power with the test data for which we have information about the income as well
- By that we can derive a **measure of accuracy**

Overfitting versus underfitting

- When training an ML model we have to pay attention to the complexity of the model
- The complexity of a model is related to the ratio of the number of features that the model uses and the amount of data
- Overfitting and underfitting are common dangers when building an ML model
- **Overfitting**:
  - training a model that performs well on the training data but poorly on the test data
  - That is usually the case when we consider a high number of feature in combination with too few data
  - Also, an overly complex model might tend to learn noise
- **Underfitting**:
  - Training a model that performs badly even on the training data
  - That is usually the case, when we use too few features to train our model

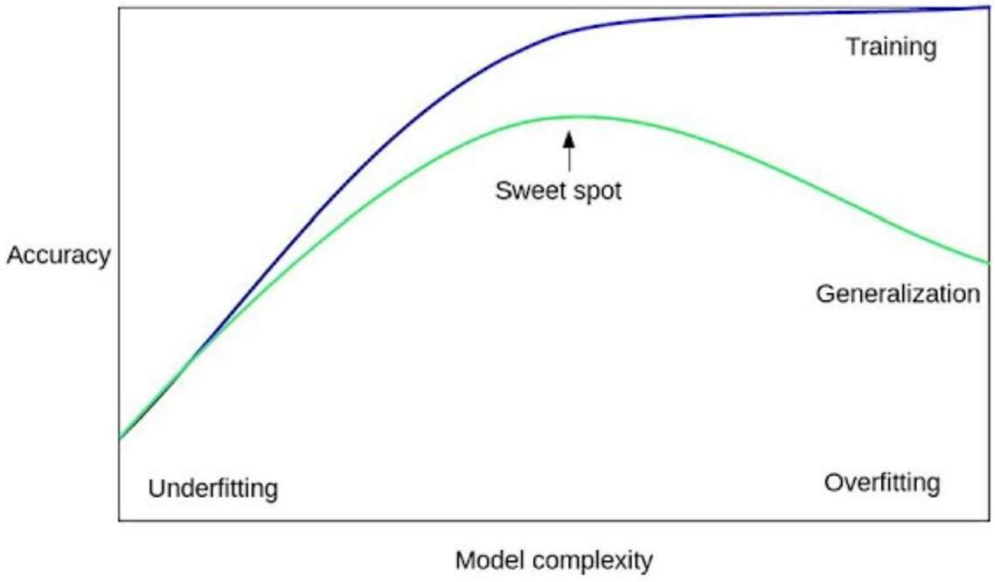


„Data Science from Scratch“, Joel Grus

Accuracy and generalizability

- As we have seen before, we can estimate the performance of our model by applying our trained model to the test data
- In an ideal world, our trained model would perform well on both the training data and the test data:
  - The model accurately predicts the output from the training data, hence it *explains* the training data well
  - The model accurately predicts the output from the test data, hence it is able to *generalize* to the test data
- However, we have find a trade-off between the performace on the training and the performance on the test data
- That is one of the biggest challenges in ML

➢ **Training a model that is complex enough to accurately learn from the training data, but not too complex in order to generalize to new data**



„Introduction to Machine Learning“, Andreas C. Müller, Sarah Guido

Overview of common supervised learning algorithms

- K-nearest neighbors (KNN)
- Linear models
- Naive Bayes Classifiers
- Decision Trees
- Kernalized Support Vector Machines

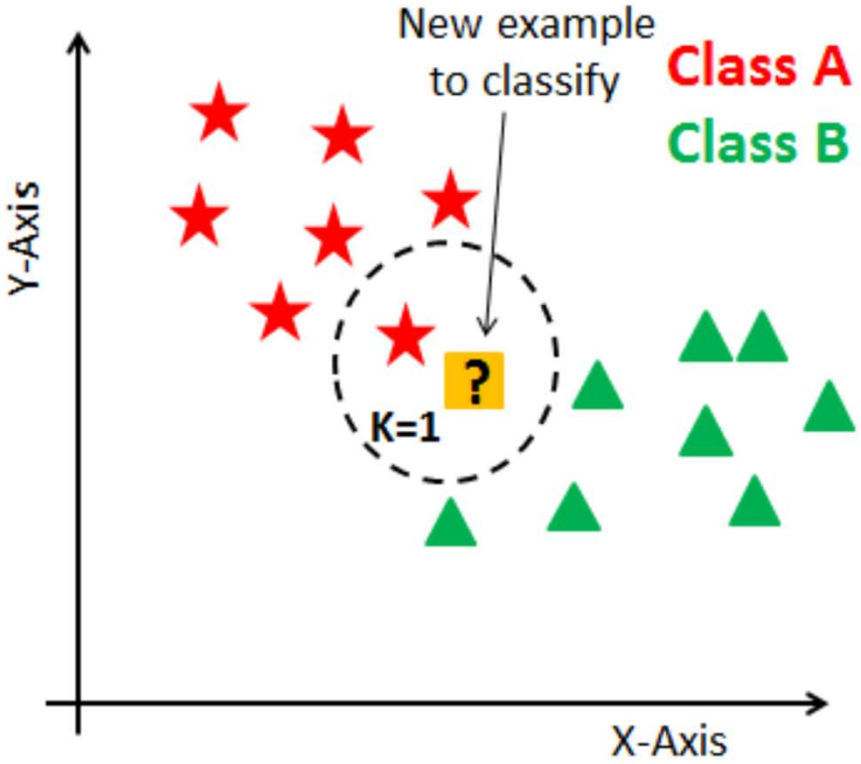
Overview of common unsupervised learning algorithms

- Clustering:
  - K-means clustering
  - Hierarchical clustering
  - etc.
- Association Rules
- Dimensionality reduction

2. Hands on with a KNN application

K-Nearest Neighbors (KNN)

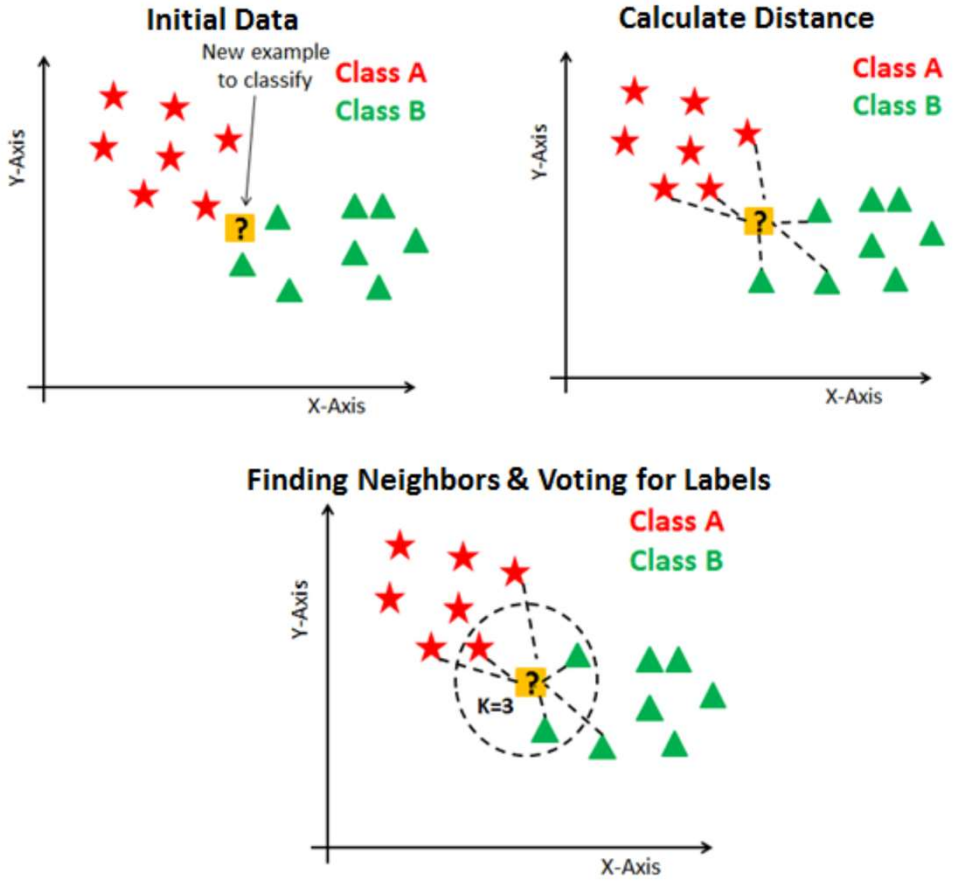
- K-Nearest Neighbors is one of the simplest ML algorithms
- K denotes the number of neighbors that you look at
- $\boldsymbol{K = 1}$: *Nearest Neighbor algorithm*
  - To make predictions for a new observation P_1, the algorithm calculates the distance between all observations in the training data and P_1
  - The algorithm then identifies the nearest neighbor by focusing on the minimum distance
  - Distances can be measured with the Euclidian metric, the Manhattan metric, the Minkowski metric, etc.
  - The predicted output for P_1 is then the label of its nearest neighbor in the training set



https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

K-Nearest Neighbors (KNN)

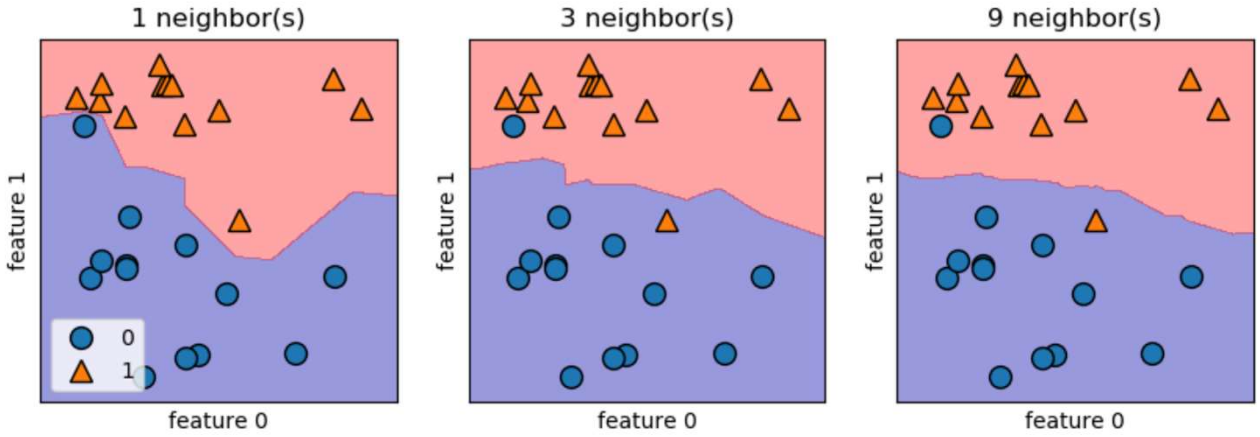
- $K \geq 2$:
  - To make predictions for a new observation P_1, the algorithm calculates the distance between all observations in the training data and P_1
  - The algorithm then identifies the K nearest neighbors by focusing on the minimum distance
  - Distances can be measured with the Euclidian metric, the Manhattan metric, the Minkowski metric, etc.
  - The predicted output for P_1 is then the most common label among its nearest neighbors in the training set
  - If your output data is binary, you should choose an odd number of neighbors!



https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

Overfitting and underfitting in the KNN

- Maybe counterintuitively, the model gets more complex the smaller K, hence the fewer neighbors you choose
- In the case of $K = 1$, the decision boundary closely follows the distribution of observations in the training set
- With increasing K, the decision boundary is getting smoothers since it is taking into account more than just one observation
- Often, the accuracy of predicting the test data is reduced when using only a single neighbor



„Introduction to Machine Learning“, Andreas C. Müller, Sarah Guido

… Open *Session_10_Machine_Learning.ipynb* in jupyter notebook