Session 14:

Supervised learning, part 3

Andreas Bjerre-Nielsen

Agenda

machine learning for social scientists

1. measures for classification
2. nested cross validation
3. non-linear ML
    - tree based models
    - neural networks
4. machine learning for social scientists

Vaaaamos

In [3]:
```
import warnings
from sklearn.exceptions import ConvergenceWarning
warnings.filterwarnings(action='ignore', category=ConvergenceWarning)

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

plt.style.use('default') # set style (colors, background, size, gridlines etc.)
plt.rcParams['figure.figsize'] = 10, 4 # set default size of plots
plt.rcParams.update({'font.size': 18})
```

With great power ...

comes great responsibility...

You have been suffering a lot with implementing estimators... why?

- If you don't know what is going on you are likely to apply erroneously.
- So very important although you don't use in the exam.

Measures for classification

Breakdown by error type (1)

We measure the accaracy as the rate of true predictions, i.e.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{True}{True + False}$$

where our measures are

		Predicted class	
		P	N
Actual class	P	True positives (TP)	False negatives (FN)
	N	False positives (FP)	True negatives (TN)

Breakdown by error type (2)

Some powerful measures:

- Precision: share of predicted positive that are true
  - PRE = $\frac{TP}{TP+FP}$
- Recall: share of actual positive that are true
  - REC = $\frac{TP}{TP+FN} = \frac{TP}{AP}$
- F1: mix recall and precision: $\frac{2 \cdot PRE \cdot REC}{PRE+REC}$

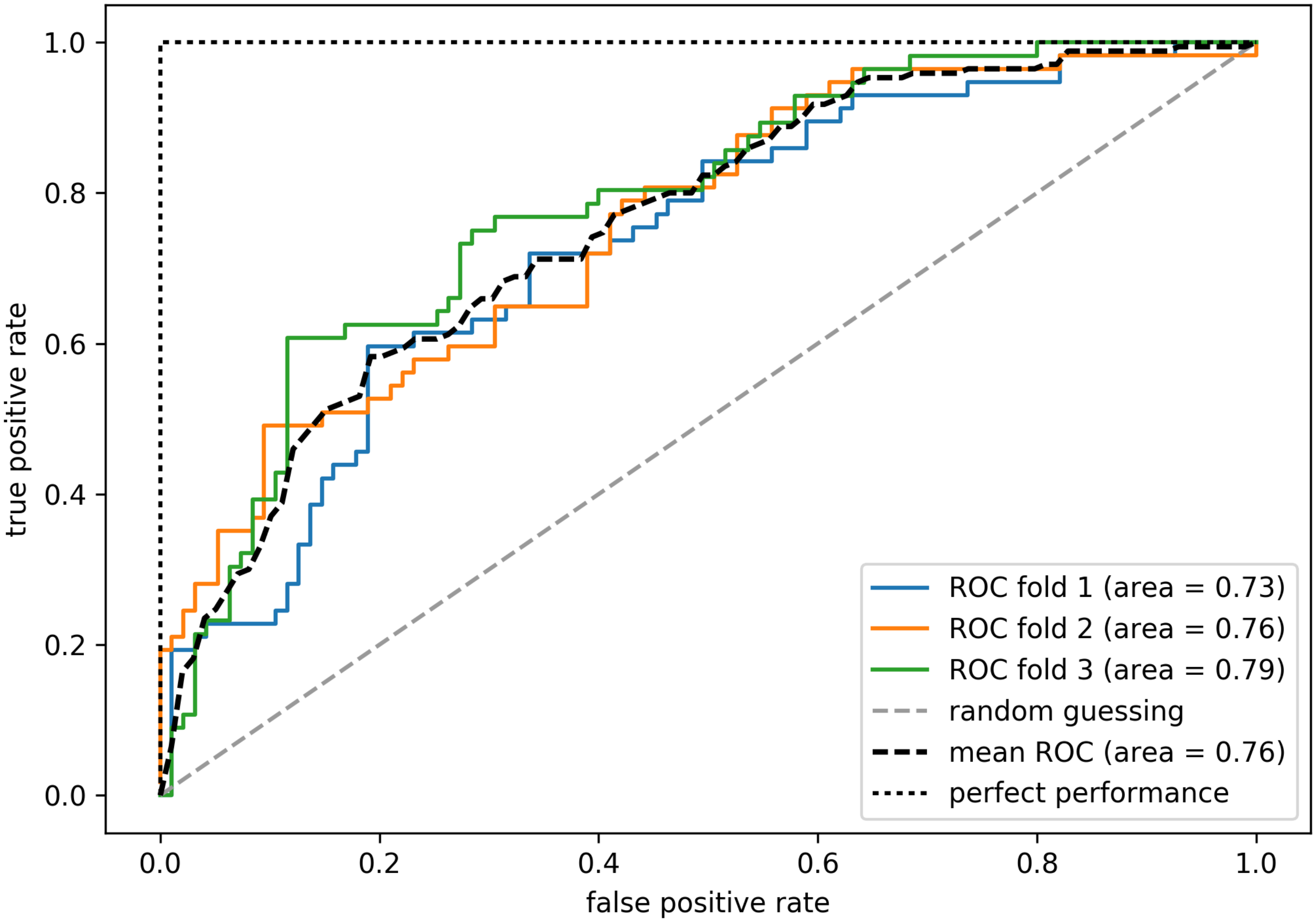
In [4]:

```
from sklearn.metrics import precision_score, recall_score, f1_score
```

Breakdown by error type (3)

Classification models provide a predicted likelihood of being in the class or not:

- Receiver Operating Characteristic (ROC) curve by varying thresholds for predicted true.
    - ROC is a *theoretical* measure of model performance based on probabilities.
    - AUC: Area Under the (ROC) Curve.



1.0
0.8
0.6
0.4
0.2
0.0
true positive rate
0.0
0.2
0.4
0.6
0.8
1.0
false positive rate
ROC fold 1 (area = 0.73)
ROC fold 2 (area = 0.76)
ROC fold 3 (area = 0.79)
random guessing
mean ROC (area = 0.76)
perfect performance

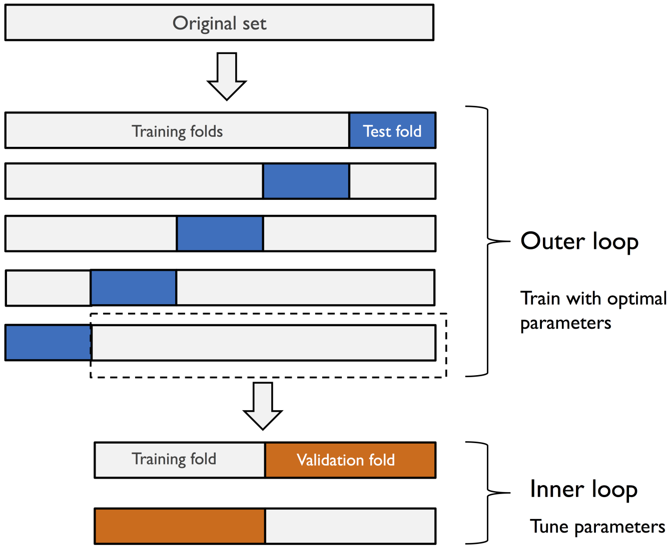
Nested cross validation

Nested cross validation (1)

- Model validation does not consider that we are also tuning hyperparameters:
    - Leads too overfitting (Varma & Simon 2006; Cawley, Talbot 2010).
- Solution is **nested cross validation**.
    - Validation step should not be modelled as 1) train; 2) test.
    - Better way is 1) model selection: train, validate; 2) test.
    - Implement as pp 204-205 in Python for Machine Learning:
        - first inner loop: `GridSearchCV`
        - second outer loop: `cross_val_score`

Nested cross validation (2)

Cross-val. suffers from the fact that it models test-train



Non-linear ML

Success of machine learning

Are linear models the best performing models?

George E. P. Box: All models are wrong

- But some are useful.

Evidence

- Sometimes linear model are the best
- But there are many others that in general perform better
- They can find patterns that non-linear models cannot

Success of machine learning

Are linear models the best performing models?

Success of machine learning (2)

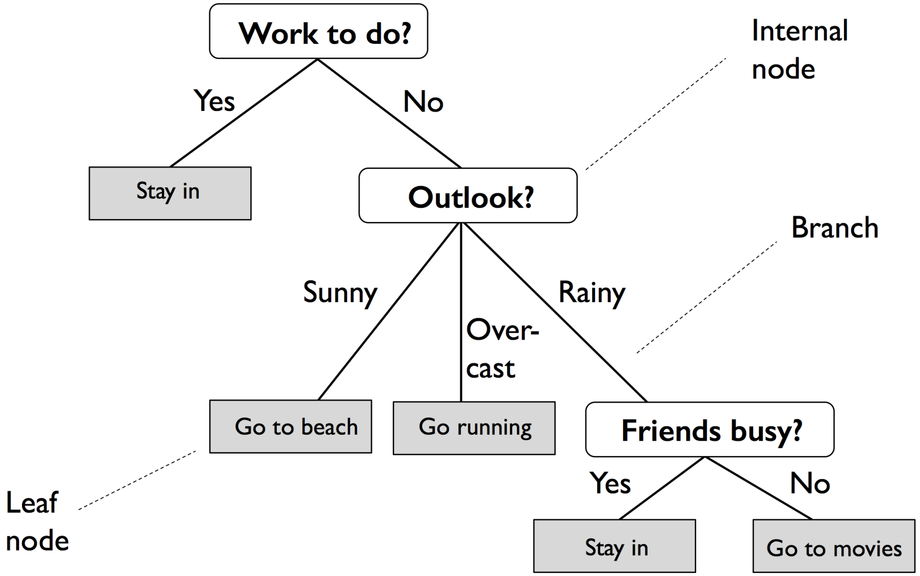
What do we call models that can fit any pattern?

- Universal approximators.
    - We can also make input non-linear using `PolynomialFeatures` of any order.
        - Follows from iterative Taylor expansion
        - Problem?
- These are very powerful tools.
    - Example of recognizing characters and digits in handwriting (MNIST data)

Tree based models

A hierarchal structure

What does a `decision tree` look like?



Sample splitting (1)

Suppose we have data like below, we are interested in predicting criminal

In [5]:

```
df = pd.DataFrame({'Criminal':[1]*5+[0]*10,
                   'From Jutland':np.random.randint(0,2,15),
                   'Parents together':[0]*4+[1]*10+[0],
                   'Parents unemployed':[1]*7+[0]*8})

print(df.sample(n=5))
```

```
    Criminal  From Jutland  Parents together  Parents unemployed
12         0             1                 1                   0
7          0             1                 1                   0
3          1             1                 0                   1
4          1             1                 1                   1
9          0             0                 1                   0
```

Sample splitting (2)

Let's try to split by variables and see whether it helps

In [6]:
```
my_split = df\
            .groupby(['From Jutland'])\
            .Criminal\
            .value_counts()\
            .unstack(level=-1)\
# Parents together, Parents unemployed
print(my_split)
```

```
Criminal      0  1
From Jutland
0             6  2
1             4  3
```

Sample splitting (3)

What might a tree structure look like?

- Parents together: Yes > Not criminal

- Parents together: No
    - Parents unemployed: Yes > Criminal
    - Parents unemployed: No > Not criminal

Improving decision trees

What can we conclude about the decision trees?

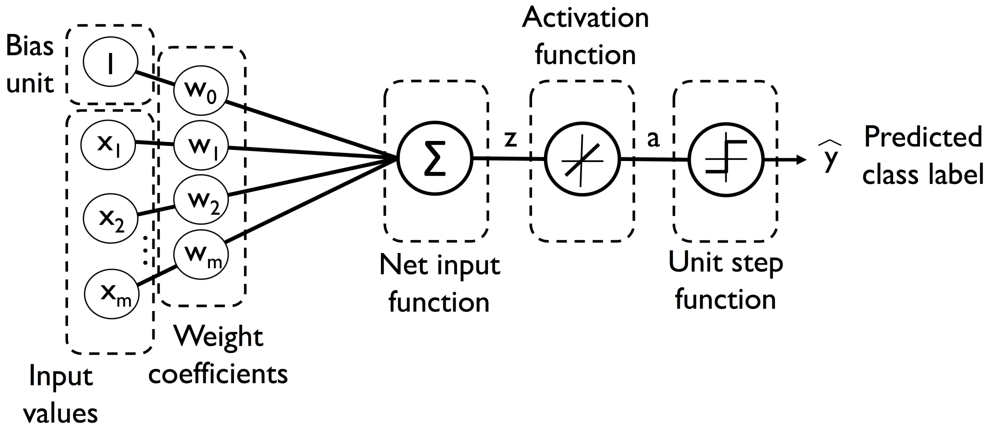
- Can fit anything ~ Universal Approximation
    - *little* underfitting (~low bias)
    - **LARGE** overfitting (~large variance)

`random forest` improves on decision trees

Neural networks

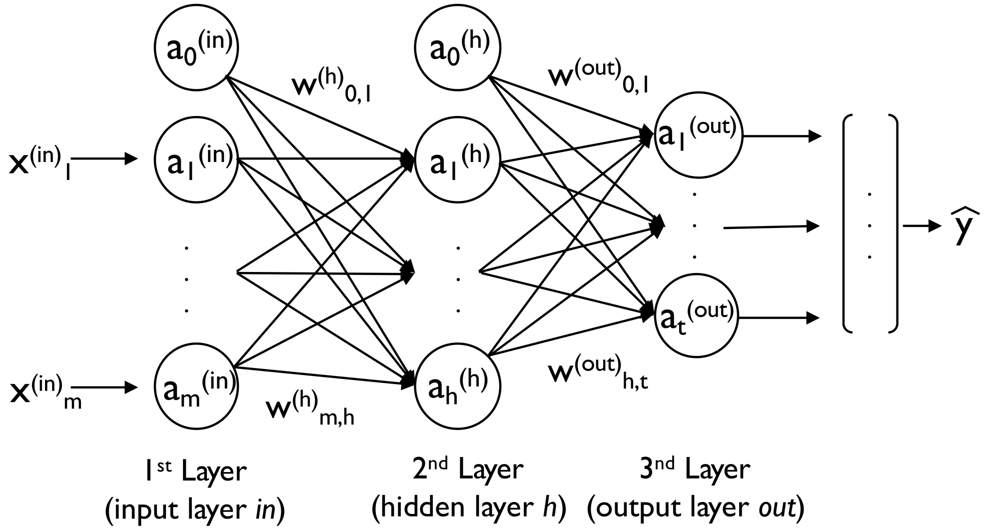
Neural networks (1)

I have forgotten, what was Adaline?



Neural networks (2)

Why are neural networks called deep learning?



Neural networks (3)

So learning about the Perceptron and Adaline actually has value?

Yes, lot's of value: these are the neurons of neural networks.

In other words, they are fundamental building blocks for doing deep learning.

Neural networks (4)

How are neural networks different from simply using polynomial features?

- It uses non-linear activation functions.
- A neural network with one hidden layer has universal approximation.
    - This corresponds to quadratic if linear.
- In practice they perform really well, especially on non-linear data
    - Computer vision: recognizing characters, content in images
    - Natural Language Processing (NLP): parsing text and speech data
    - Much more

Universal approximation

Universal approximators (1)

Are decision trees the only universal approximators?

No there are also **kernel based** ones.

- K Nearest Neighbors:
    - Approximate by taking average/mode from K nearest neighbors
        - Need standardization
    - Can also be used for interpolated local measures
        - (weather, pollution, house prices etc.)
    - Not good with high dimensionality.

Universal approximators (2)

What can these these approximators be used for?

- Reduce mobil bias
- Must be careful we do not overfit (control bias)

Can I get an overview of them?

- Kernel methods, e.g. nearest neighbor
- Neural networks (1+ hidden layer, deep learning)
- Polynomial inputs
- Tree based models

Universal approximators (3)

Can we use these methods?

Yes, they all come off the shelf with `sklearn`.

- E.g. `from sklearn.ensemble import RandomForestClassifier`

For neural networks that have more hidden layers (deep learning) you need new packages:

- We recommend looking at either `pytorch` or using `keras` (which uses `tensorflow`)

Should we use these methods in the exam of this course?

NO

ML for social science

ML for social science (1): testing predictive power

ML helps us with making predictive models:

- Assess the performance of our models
- Choose the parameters that help estimate the best performing model

Can we use ML to help us clarify whether a new feature set is relevant for prediction?

ML for social science (2): new data

Machine learning can help us *'fill in the blanks'* and impute missing data

Input: Google Street View

- Infer neighborhood socioeconomic status (Naik, Raskar, Hidalgo 2016)

Input: Cell phone data

- Inferring poverty. (Blumenstock, Cadamuro, On 2015)
- Inferring mode of transportation. (Bjerre-Nielsen et al. 2019)
- Sleep (Cuttone et al. 2017)

Facebook data can help infer

- personality and demographics (Cambridge Analytica); socioeconomic status; current mood

ML for social science (3): better policy targetting

Social and medical scientists are often involved in policies aimed at:

- alleviating poverty, decrease drop-out, crime etc.

Efficacy of these programs requires targetting of individuals:

- who is most poor, who is most at risk of dropping out? dying?

Kleinberg et al. 2015 show that mortality from surgery can be predicted in advance.

- save billions of $ and not cause pain of surgery

ML for social science: improving econometrics

Many econometric methods try to establish causality:

- applications for
  - instrument variables (Hartford et al. 2017; Bjørn 2018)
  - matching (Wager, Athey 2017)
- both of these problems have a prediction problem built-in
  - can be enhanced with machine learning

Machine learning can also be used to find heterogenous and non-linear effects

ML for social science: decision problems and game theory

We can solve decision problems and games using reinforcement learning

- uses neural networks to teach "agent" to play game
    - learn to play computer games, poker, etc.
- solve problems where game theory is intractable

Outro

There are amazing resources for you to keep learning, online and offline.

@ in Denmark.

- Center for Social Data Science
    - Advanced courses (replace Topics in Social Data Science)
        - ASDS1: tree based models for prediction and statistics; network inference
        - ASDS2: unstructured data; (text, image), neural networks
    - Seminar in econometrics and machine learning (https://kurser.ku.dk/course/a%C3%98kk08386u/2019-2020) this fall
    - A new education next year
- More courses are taught in machine learning at CS dept. (DIKU), DTU, ITU

@ online: coursera, edX, DataCamp, MIT open courseware, etc.

Everyone freeze!

Please run the course evaluation now (<5 min)

- Evaluate our actions:
  - What was good, what was not good
- Please evaluate:
  - our teaching: did I make myself clear? was I too fast? what about Snorre and David?
  - the material (lectures, exercises, books)
  - autograding
  - the quizzes (those that worked)
  - machine learning curriculum

The end

Return to agenda