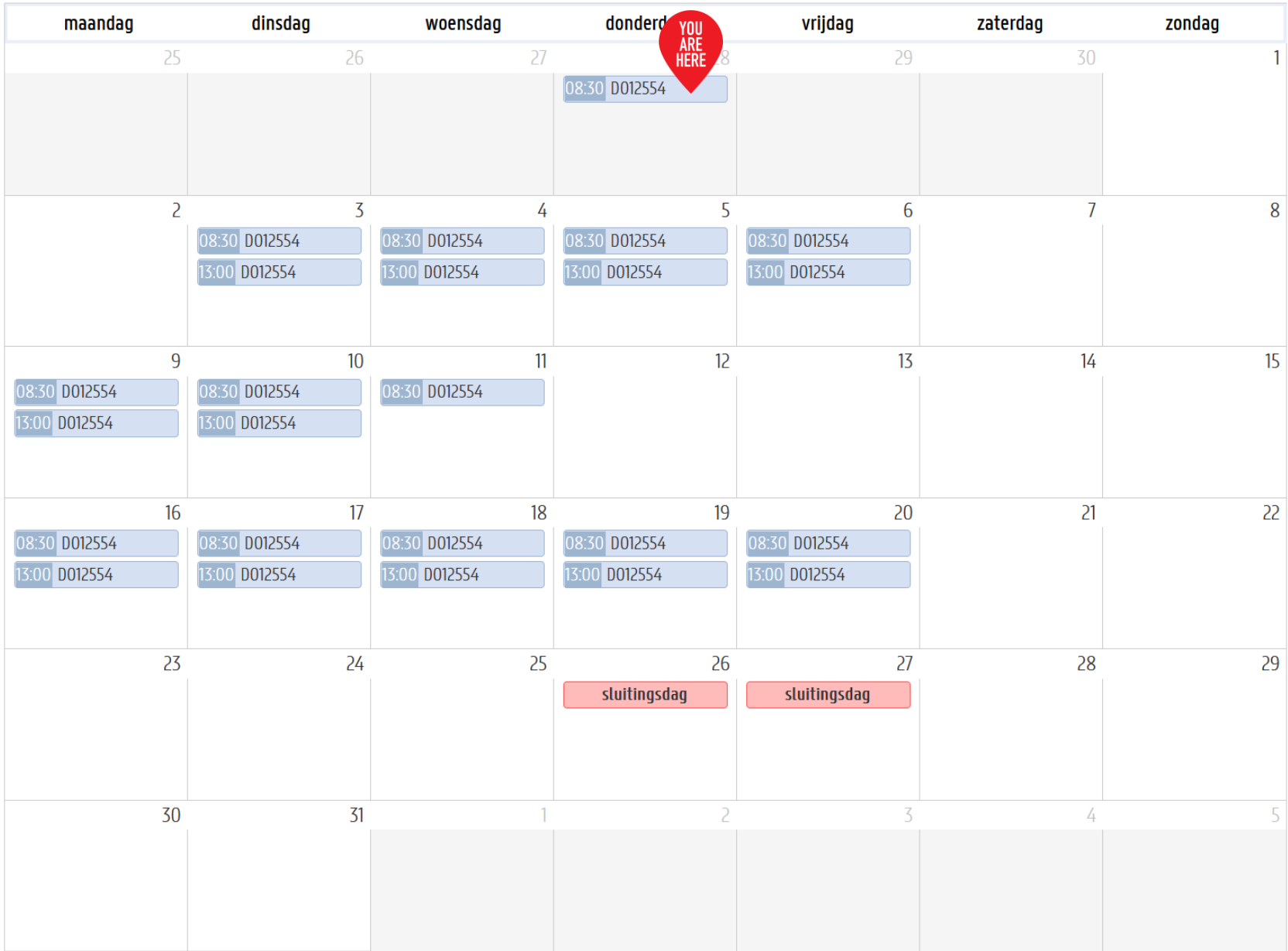
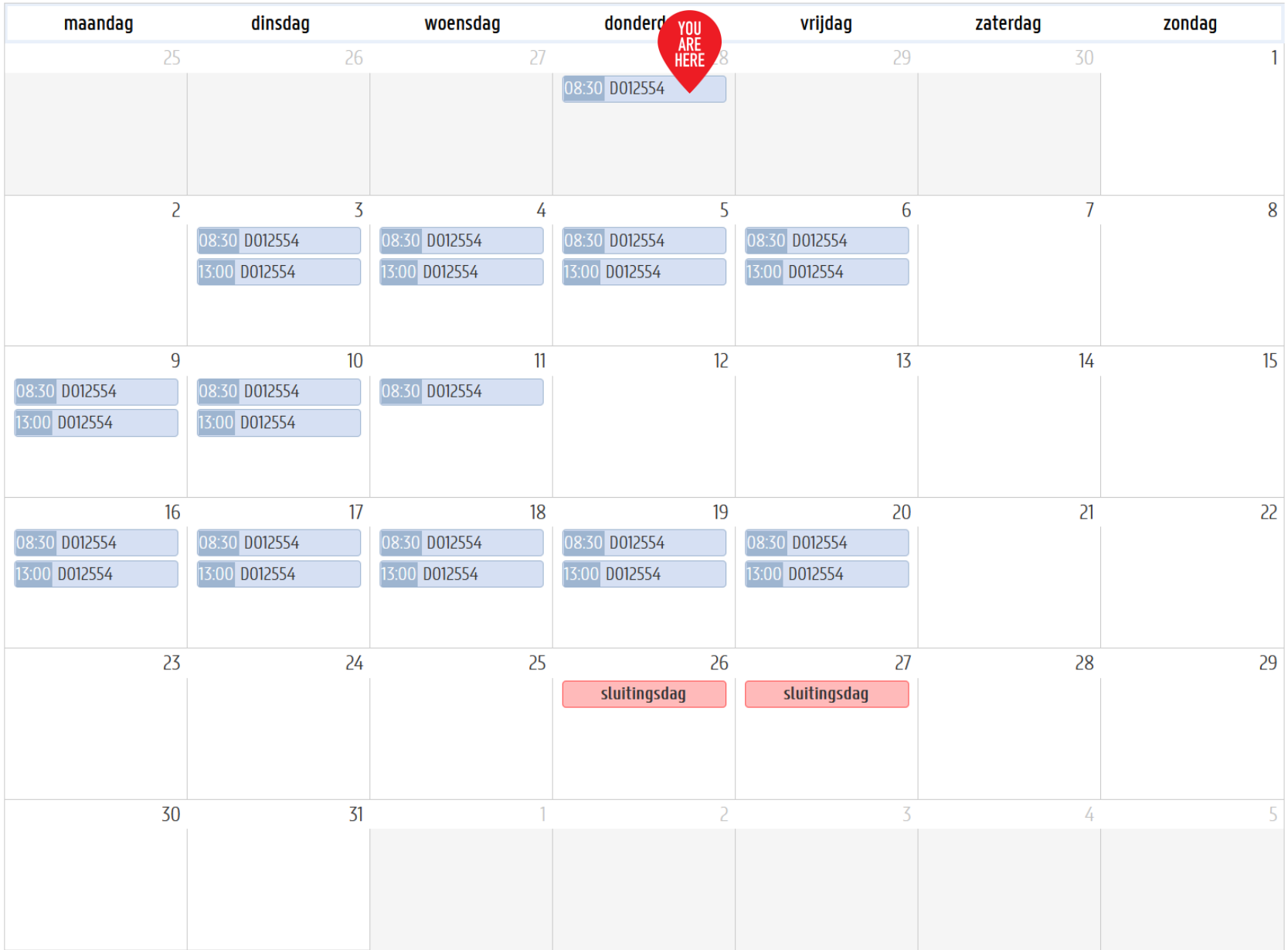
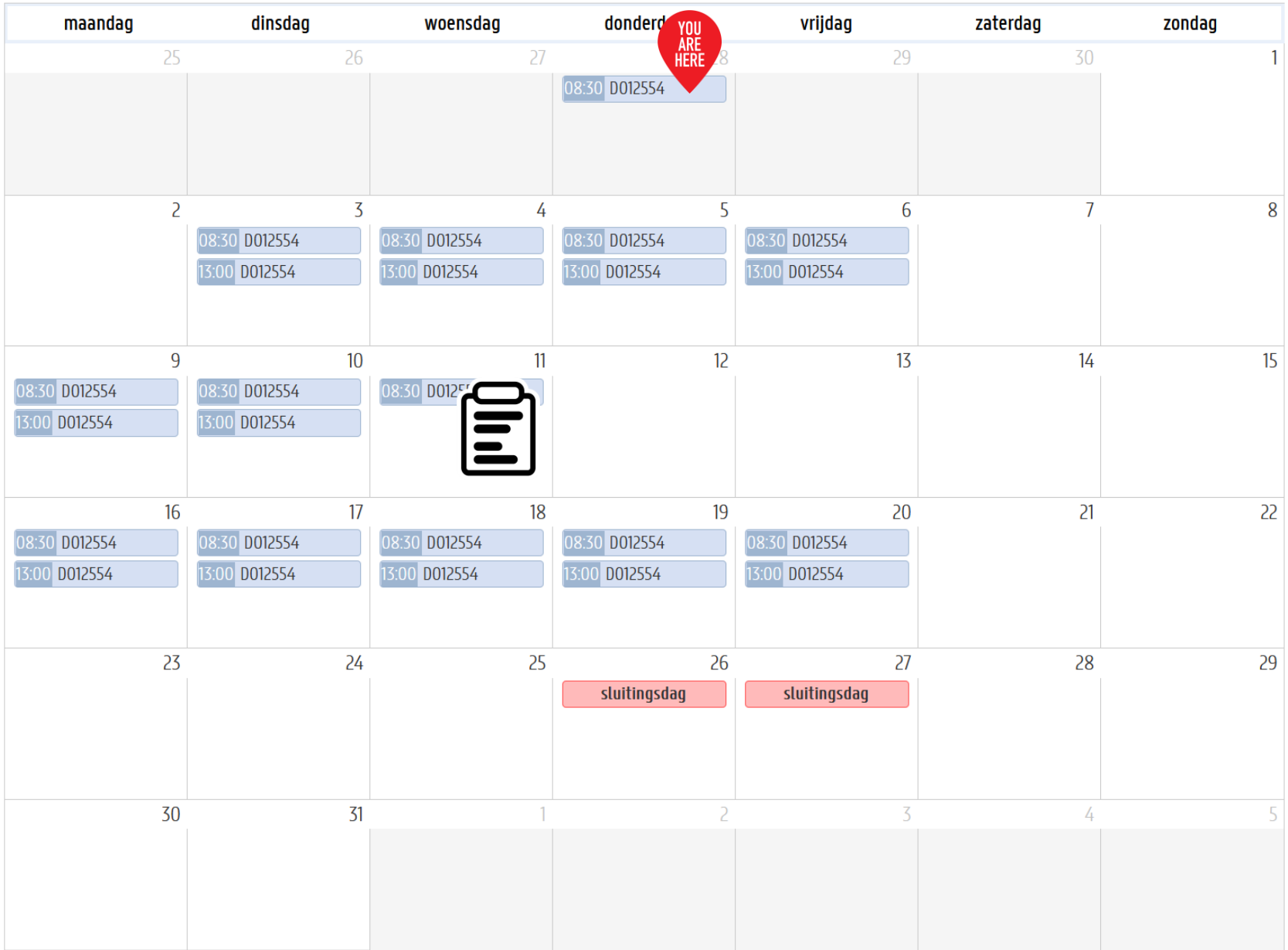

maandag
dinsdag
woensdag
donderd
vrijdag
zaterdag
zondag
YOU ARE HERE
25
26
27
28
29
30
1
08:30 D012554
2
3
4
5
6
7
8
08:30 D012554
13:00 D012554
08:30 D012554
13:00 D012554
08:30 D012554
13:00 D012554
08:30 D012554
13:00 D012554
9
10
11
12
13
14
15
08:30 D012554
13:00 D012554
08:30 D012554
13:00 D012554
08:30 D012554
16
17
18
19
20
21
22
08:30 D012554
13:00 D012554
08:30 D012554
13:00 D012554
08:30 D012554
13:00 D012554
08:30 D012554
13:00 D012554
08:30 D012554
13:00 D012554
23
24
25
26
27
28
29
sluitingsdag
sluitingsdag
30
31
1
2
3
4
5



- Theory (50%)
  - morning lectures
  - written exam



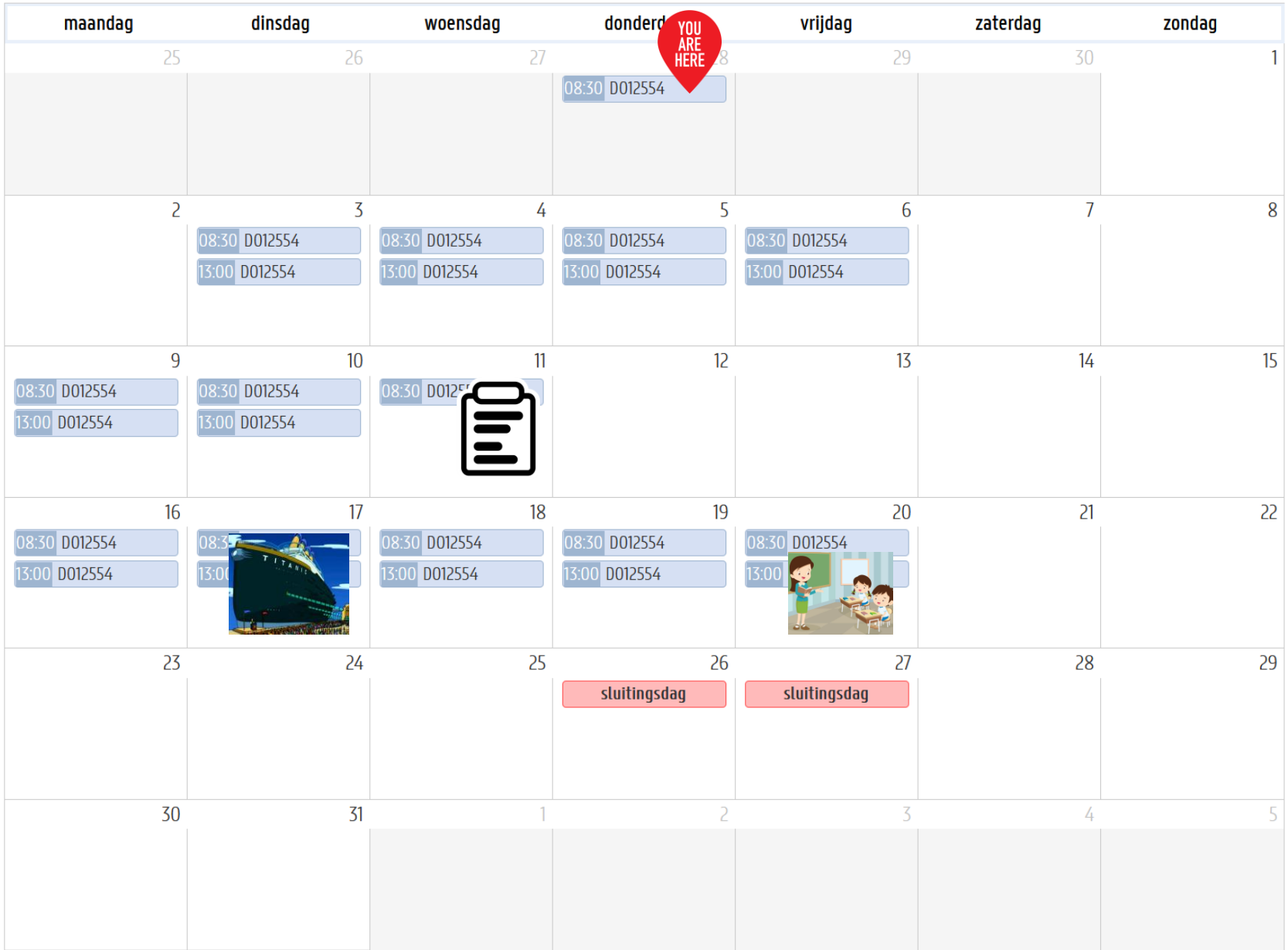
- Theory (50%)
  - morning lectures
  - written exam

- Project (15%)
  - notebooks:
    - splice site classification
    - data clustering

maandag	dinsdag	woensdag	donderd	vrijdag	zaterdag	zondag
25	26	27	28	29	30	1
			08:30 D012554			
2	3	4	5	6	7	8
	08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554		
9	10	11	12	13	14	15
08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554	08:30 D012554				
16	17	18	19	20	21	22
08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554	08:30 D012554 13:00		
23	24	25	26	27	28	29
			sluitingsdag	sluitingsdag		
30	31	1	2	3	4	5

YOU ARE HERE

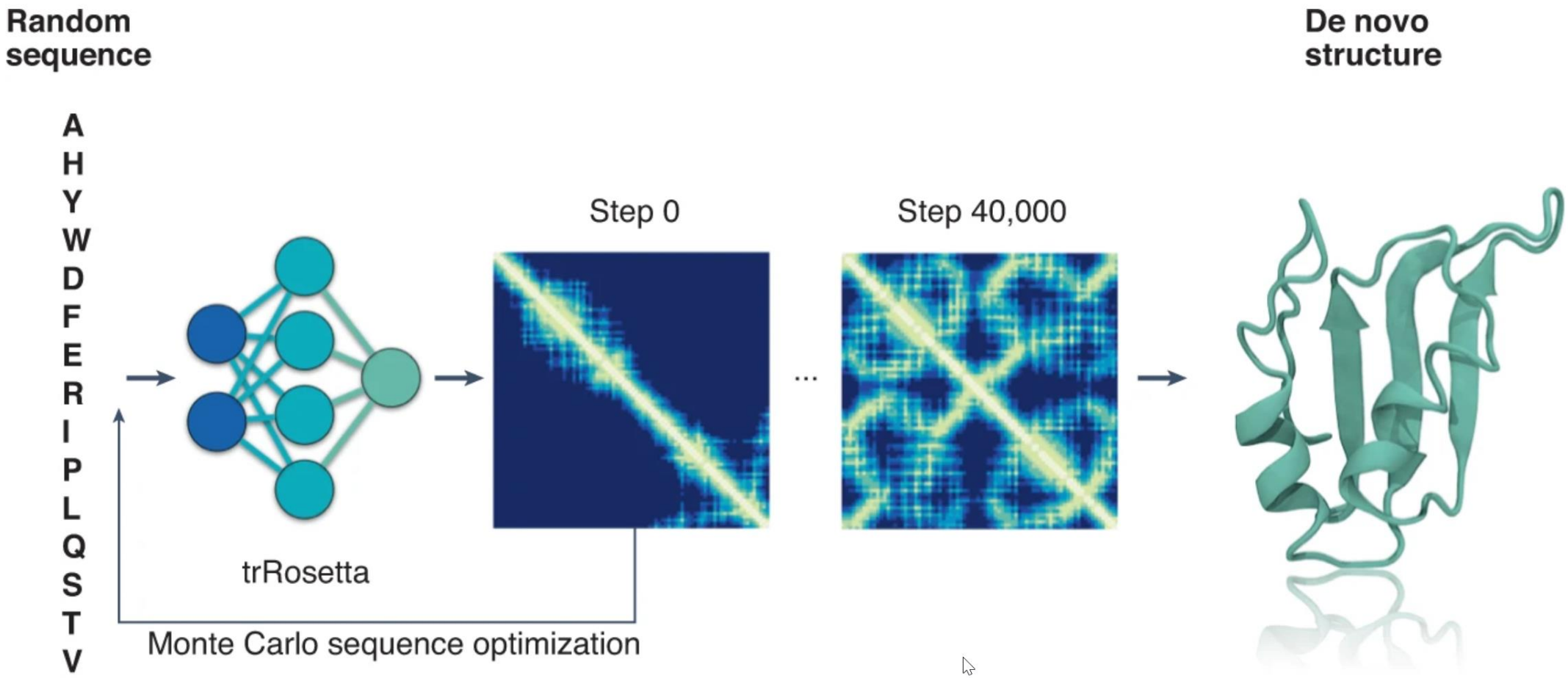
- Theory (50%)
  - morning lectures
  - written exam
- Project (15%)
  - notebooks:
    - splice site classification
    - data clustering
- Micro-teaching (15%)
  - short presentation about specific ML method



- Theory (50%)
  - morning lectures
  - written exam
- Project (15%)
  - notebooks:
    - splice site classification
    - data clustering
- Micro-teaching (15%)
  - short presentation about specific ML method
- Kaggle contest (20%)
  - short report on best result (written/oral)?

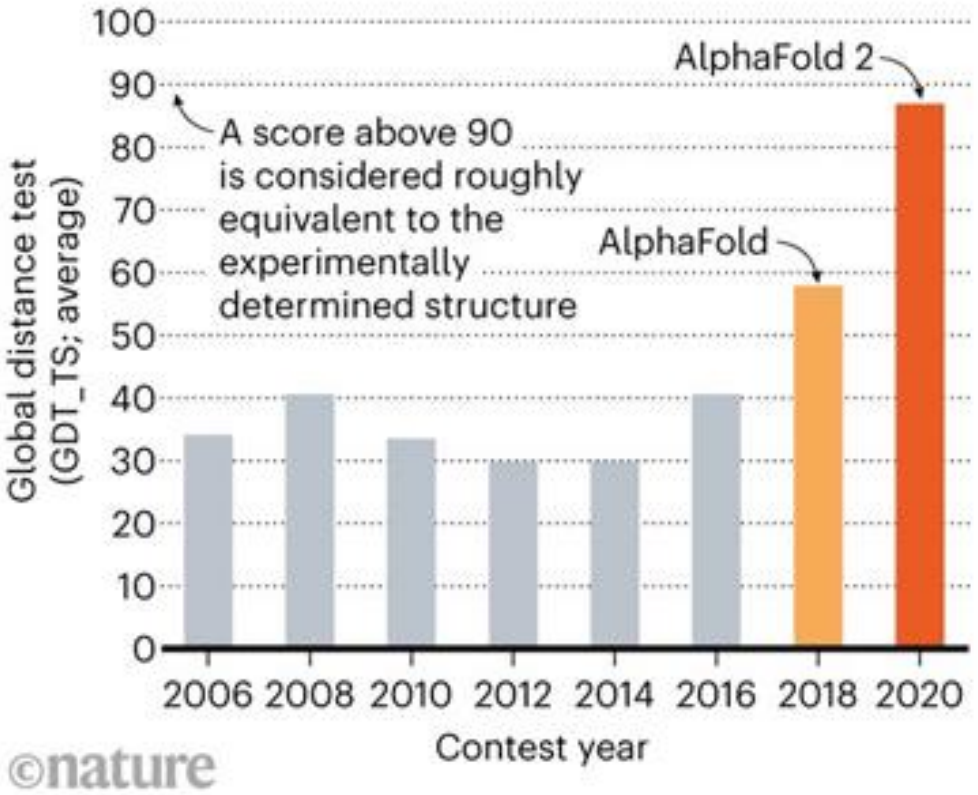
maandag	dinsdag	woensdag	donderd	vrijdag	zaterdag	zondag
25	26	27	28	29	30	1
			08:30 D012554			
2	3	4	5	6	7	8
	08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554		
9	10	11	12	13	14	15
08:30 D012554 13:00 D012554	08:30 D012554 13:00 D012554	08:30 D012				
16	17	18	19	20	21	22
08:30 D012554 13:00 D012554	08: 13:0	08:30 D012554 13:00 D01	08:30 D012554 13:00 D012554	08:30 D012554 13:00		
23	24	25	26	27	28	29
			sluitingsdag	sluitingsdag		
30	31	1	2	3	4	5

- Major internship
  - TRIM32 interactions
  - AlphaFold2
  - write a paragraph



Random amino acid sequences are generated; here a single sequence is shown. These sequences are fed into the trRosetta structure-prediction convolutional neural network. The first predictions (step 0) lead to distance maps with no strong patterns. After a Monte Carlo sampling where the contrast between the trRosetta-predicted distribution of distances and a background distribution is optimized, new sequences are produced in a step-wise manner. The final distance maps (step 40,000) are feature-rich and lead to well-structured de novo proteins.

Ferruz, N., Höcker, B. Dreaming ideal protein structures. *Nat Biotechnol* **40,** 171–172 (2022)

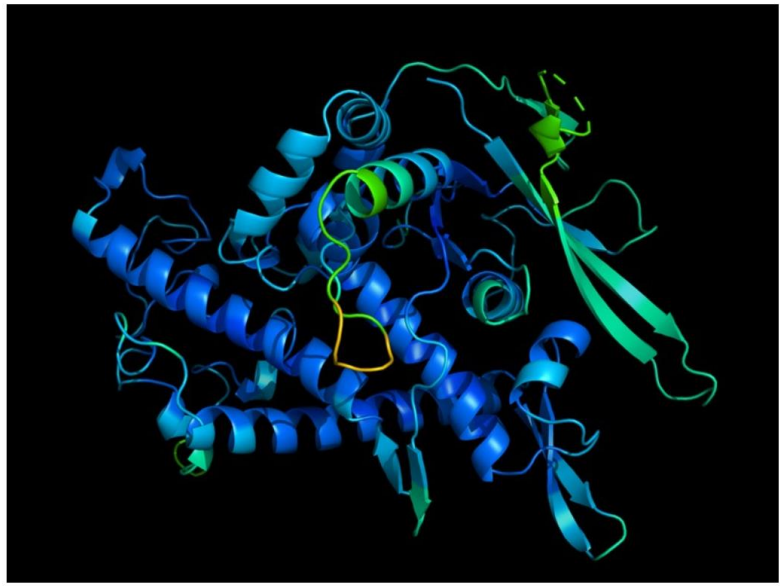


NEWS | 30 November 2020

'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures

Google's deep-learning program for determining the 3D shapes of proteins stands to transform biology, say scientists.

Ewen Callaway



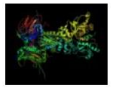
A protein's function is determined by its 3D shape. Credit: DeepMind

You have full access to this article via **Universiteit Gent**

Download PDF

Related Articles

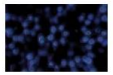
AI protein-folding algorithms solve structures faster than ever



The revolution will not be crystallized: a new method sweeps through structural biology

The computational protein designers

Revolutionary microscopy technique sees individual atoms for first time



Subjects

Computational biology and bioinformatics

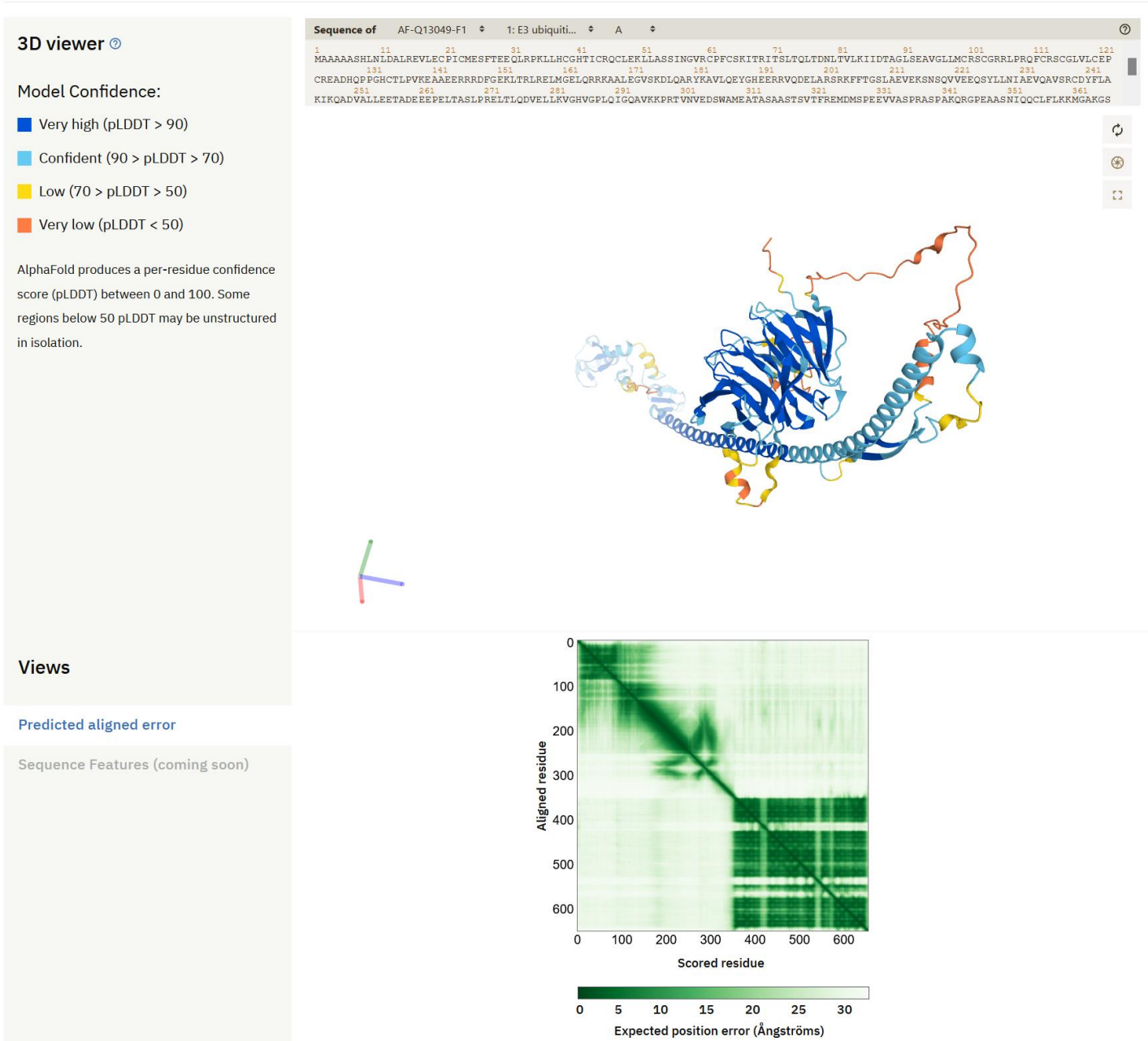
Structural biology | Drug discovery

Sign up to Nature Briefing

An essential round-up of science news, opinion and analysis, delivered to your inbox every weekday.

Email address

Ferruz, N., Höcker, B. Dreaming ideal protein structures. *Nat Biotechnol* **40,** 171–172 (2022)



https://alphafold.ebi.ac.uk/

Ferruz, N., Höcker, B. Dreaming ideal protein structures. *Nat Biotechnol* **40,** 171–172 (2022)

https://scikit-learn.org/stable/

110%

scikit learn Install User Guide API Examples Community More

scikit-learn

Machine Learning in Python

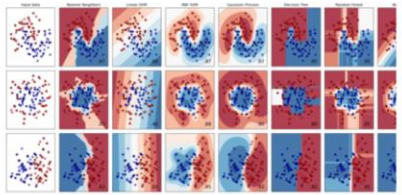
Getting Started | Release Highlights for 1.0 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

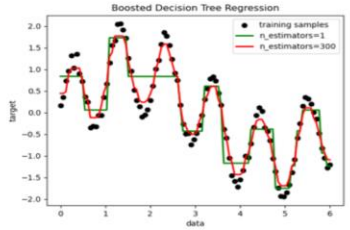


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

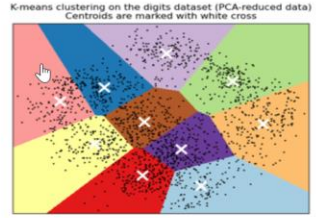


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

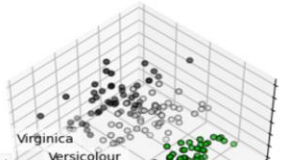


Examples

Dimensionality reduction

Reducing the number of random variables to consider.

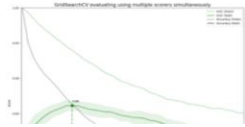
Applications: Visualization, Increased efficiency
Algorithms: k-Means, feature selection, non-negative matrix factorization, and more...



Model selection

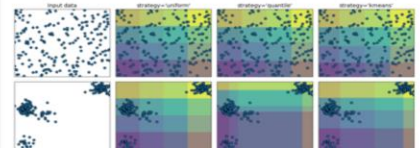
Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics, and more...



Preprocessing

Feature extraction and normalization.



jupyter 1.1_Linear_regression (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Markdown

1.1 Linear regression

In the introduction we constructed an instrument to **classify** a raw dermoscopic image of a lesion as either benign or malignant. This required some kind of decision boundary that **separates** these classes as good as possible.

Suppose we were asked to implement a similar approach to predict the survival time of stage III melanoma patients. In this case the data points will be labeled with a **continuous value** that represents this survival time, i.e. there are no classes.

In statistics and machine learning modeling a continuous value is known as **regression**. Typical regression tasks include modeling drug effectiveness, toxicity, disease progression, and survival times. In genomics, transcriptomics and proteomics regression is successfully applied to for instance calibrate and allign experiments, assess data quality, and predict expression levels. Regression is a general term for **modeling the relationship** between a **target** (or **dependent variable**) and one or more **features** (or the **independent** or **explanatory variable(s)**).

To explain regression data analysis we first open a data set that contains relevant information for about 442 patients that are diagnosed with diabetes disease. Each patient is a data point in this data set and is described by 11 features that measure age (AGE), sex (SEX), body mass index (BMI), average blood pressure (BP), together with six blood serum measurements (Sx) as well as the target (Y) that represents a quantitative measure of disease progression one year after baseline.

In [4]:
```
dataset = pd.read_csv('https://raw.githubusercontent.com/sdgroeve/Machine_Learning_course_UGent_D012554_data/master/note
print(dataset.head())
print()
print("Shape of data set: " + str(dataset.shape))
```

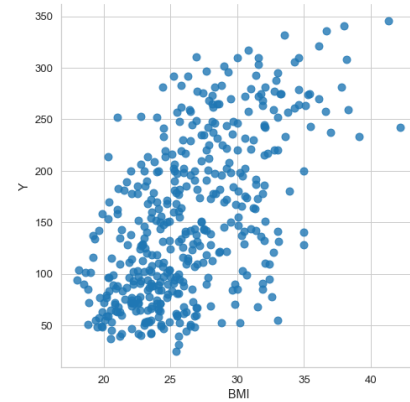
```
   AGE  SEX   BMI     BP   S1     S2    S3   S4      S5  S6    Y
0   59    2  32.1  101.0  157   93.2  38.0  4.0  4.8598  87  151
1   48    1  21.6   87.0  183  103.2  70.0  3.0  3.8918  69   75
2   72    2  30.5   93.0  156   93.6  41.0  4.0  4.6728  85  141
3   24    1  25.3   84.0  198  131.4  40.0  5.0  4.8903  89  206
4   50    1  23.0  101.0  192  125.4  52.0  4.0  4.2905  80  135

Shape of data set: (442, 11)
```

The data set was created because diabetes disease progression Y observed for a patient is assumed to be dependent on one or more of the features and we want to know what this potential relationship looks like. Can the exploratory variables **explain** the values observed for Y? If they do then we can use this explanation to **predict** diabetes disease progression for future unseen external patients.

We could for instance ask ourselves how Y varies with the Body Mass Index (BMI) of a patient? So can we explain disease progression in terms of the BMI? To investigate thiss we plot the relationship between BMI and Y for the patients in the data set in a scatter plot:

In [5]:
```
sns.lmplot(x="BMI", y="Y", data=dataset,
           fit_reg=False, height=7, scatter_kws={"s": 80})
plt.show()
```



Just by looking at this patient data we can already see a trend: values for Y seem to be larger for larger values of the BMI. From this data we can estimate or **model** the relationship between the BMI and diabetes disease progression Y, assuming that there is one. Such a **regresssion model** explains Y in terms of BMI.

To compute such a model, we need to make **assumptions about the true underlying model that generated the data**. A common and straightforward assumption is that the relationship between the two variables is **linear**, i.e. disease progression varies linearly with the BMI.

In this case we use the term **linear regression model** that assumpes that the target varies linearly with the features. **Fitting** a linear regression model is known **linear regression**.

To correctly apply linear regression we need to normalize the features.

Form scikit-learn we use the `StandardScaler` module to standardize the features and the target. We first create `StandardScaler` objects

Search or jump to... / Pull requests Issues Marketplace Explore

sdgroeve / Machine_Learning_course_UGent_D012554 Private

Code Issues Pull requests Actions Projects Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

svendegroeve updated to 2022 164b3a1 3 minutes ago 109 commits

notebooks	updated to 2022	3 minutes ago
practicum/Classification	updated to 2022	3 minutes ago
slides	initial commit	2 years ago
README.md	initial commit	2 years ago
environment.yml	initial commit	2 years ago

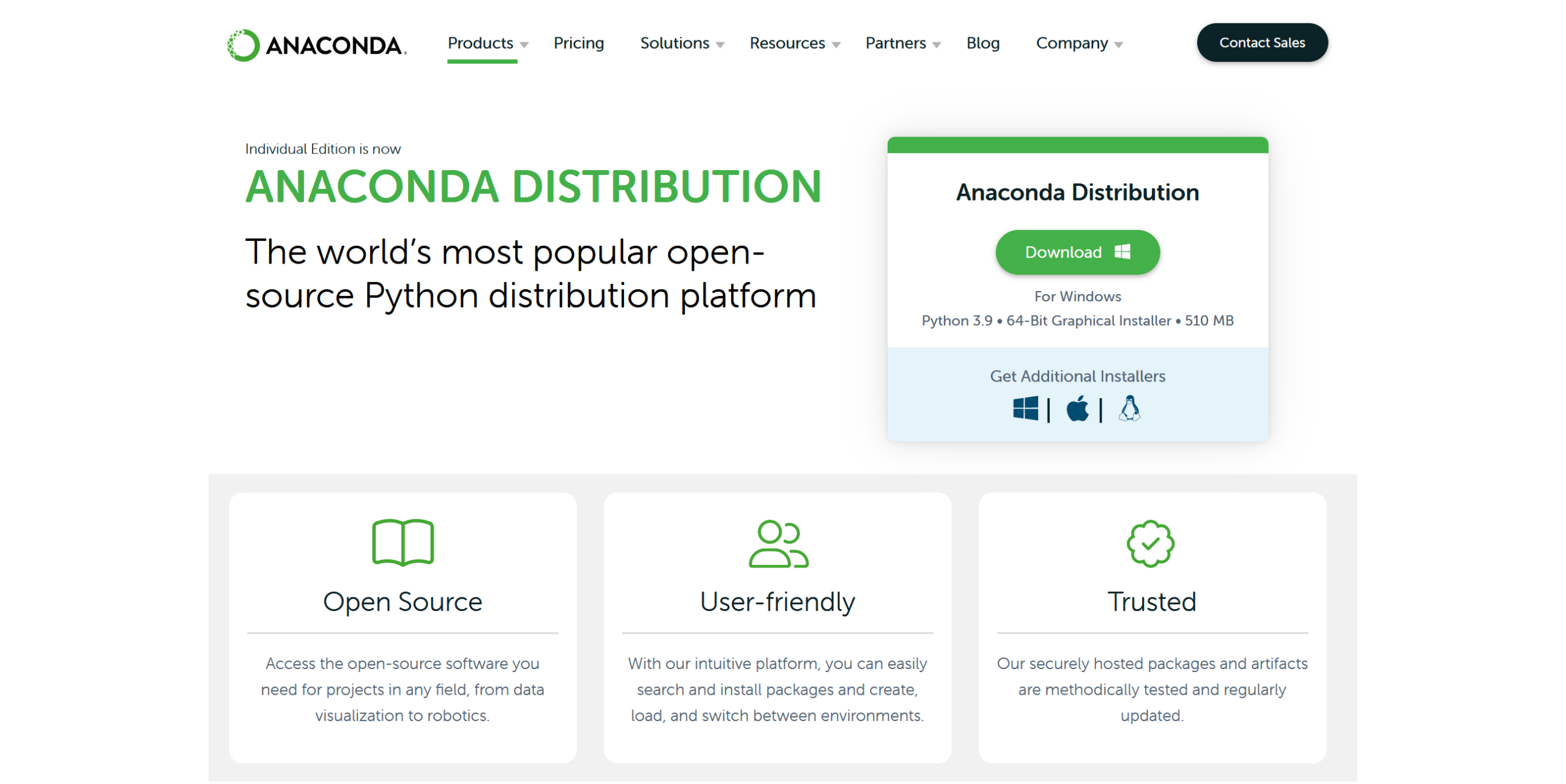
README.md

D012554

Machine Leren methoden voor biomedische gegevens

launch binder

- Clone
- Make private
- Enjoy peace of mind



https://www.anaconda.com/products/distribution

Overview Data Code Discussion Leaderboard Rules Team My Submissions **Submit Predictions** ···

Leaderboard

Raw Data **Refresh**

Search leaderboard

This leaderboard is calculated with all of the test data.

#	Team	Members	Score	Entries	Last	Code
1	Rizky R&F YS		0.81669	166	7d	
2	Karl Cini		0.81669	102	4m	
3	CADang		0.81646	201	11h	
4	enescnkr		0.81505	67	3d	
5	Mukharbek Organokov		0.81482	10	21d	
6	Marcello Gomitoni		0.81458	90	1mo	
7	John Mitchell		0.81458	8	6d	< >
8	Deepak Kaura		0.81458	4	5d	

https://www.kaggle.com/competitions/spaceship-titanic/

- Sections
  - EDA -> description of the features and the target
  - Pre-processing (missing values, feature scaling)
  - Feature engineering
  - Hyperparameter optimization
  - Model performance comparison (does your performance estimation match the observed leaderboard score)
- Grades
  - Protocol (10)
  - Clarity of the report (5)
  - Effort (5)

1. Nearest Neighbour (k-NN) & Nearest Centroid

• Explain how the k-NN and Nearest Centroid algorithms work.
• Compare them with logistic regression of the Kaggle dataset

For k-NN I refer to the scikit-learn website.

For Nearest Centroid there is a good wiki page with a link to the original paper (Tibshirani):
https://en.wikipedia.org/wiki/Nearest_centroid_classifier

2. Feature subset selection

• Compare recursive feature elimination with sequential feature selection.
• Find out what the most important features are in the Kaggle dataset (use a linear SVM).

https://scikit-learn.org/stable/modules/feature_selection.html#sequential-feature-selection

3. Stacked generalization

• Explain what stacking classifiers means.
• Evaluate stacking a logistic regression, linear SVM and a Random Forest on the Kaggle dataset.

https://scikit-learn.org/stable/modules/ensemble.html#stacked-generalization

4. Multi-class classification with error-correcting output codes

• Explain error-correcting output codes.
• Compare with OneVSRest and OneVSOne on the MNIST (small version) dataset.

https://scikit-learn.org/stable/modules/multiclass.html

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html

- Abstract (250)
- Introduction (1000)
- Materials and Methods (1500)
- Results (1000)
- Discussion (1000)

https://inria.github.io/scikit-learn-mooc/index.html

https://inria.github.io/scikit-learn-mooc/python_scripts/cross_validation_nested.html

- Dinsdag 10/05: L2
- Woensdag 11/05: L0 -> Deep Neural Networks Lecture (9:00)
- Maandag 16/05: L0
- Dinsdag 17/05: L1
- Woensdag 18/05: L1
- Donderdag 19/05: L2
- Vrijdag 20/05: L2