Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Towards Digitalisation in Examination and Grading: Best Practice and Challenges

FrOSCon 2019

Tim Metzler and Minh Nguyen and Anastassia Küstenmacher and Deebul Nair and Paul G. Plöger

Department of Computer Science
University of Applied Sciences Bonn-Rhein-Sieg

August 7, 2019

Digitalisation in Teaching

Lectures in the past

Figure: Lecture at University of Bologna, 14th century[1]

[1] https://commons.wikimedia.org/wiki/File:Laurentius_de_Voltolina_001.jpg

Digitalisation in Teaching

Lectures now

Figure: Lecture at University Missouri School of Journalism, 2007[2]

[2] https://commons.wikimedia.org/wiki/File:Universite_Missouri_School_of_Journalism.jpg

Digital Lectures

- Electronic slides
- Digital examples (animations, live scripts, etc)
- Recorded lectures

Digitalisation in Teaching

Assignments in the past

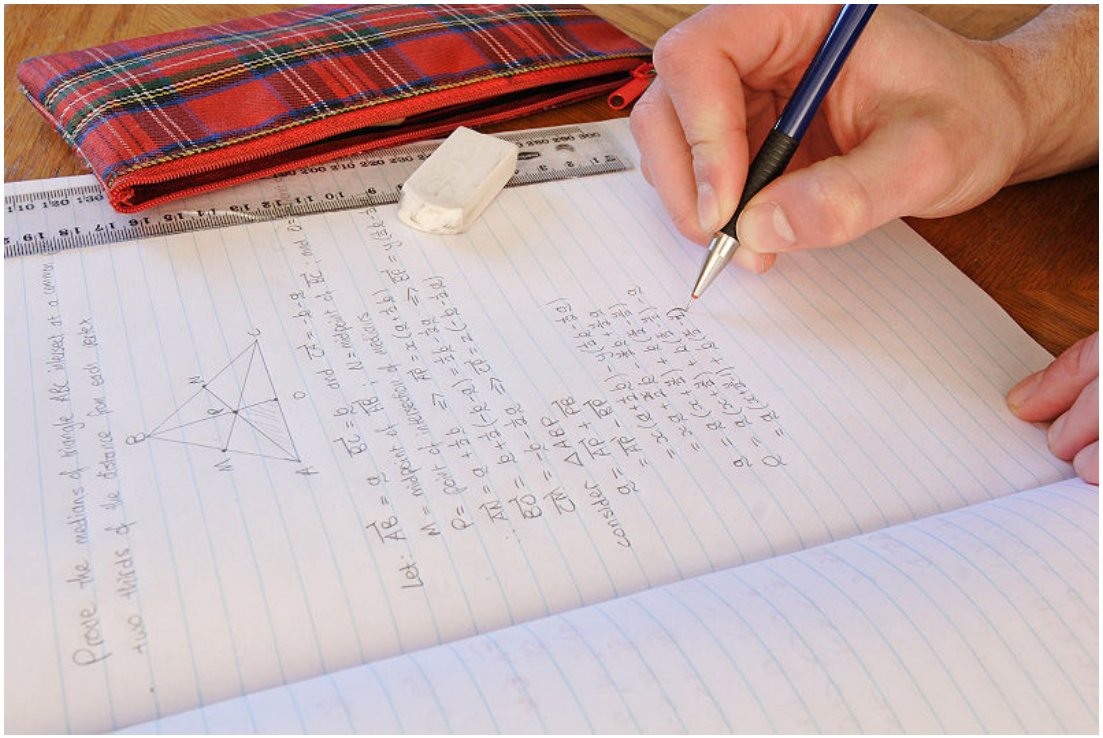

Figure: Student doing an assignment[3]

[3] https://commons.wikimedia.org/wiki/File:Homework_-_vector_maths.jpg

Digitalisation in Teaching

Assignments now

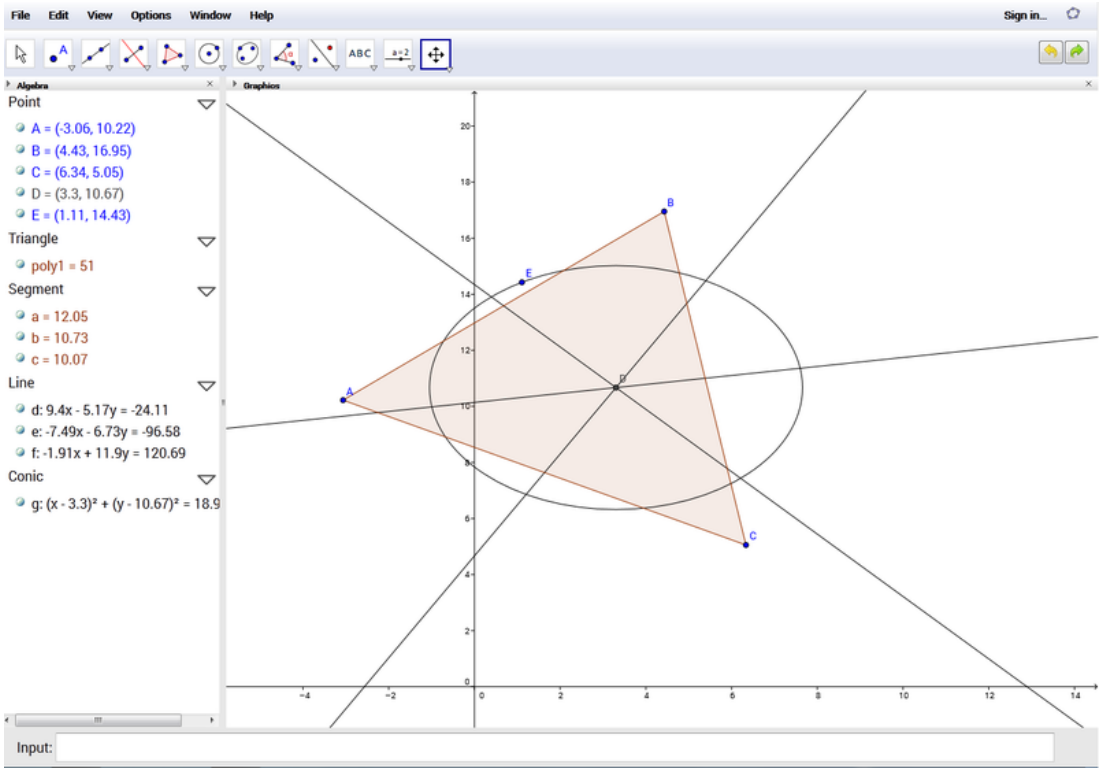

Figure: GeoGebra[4]

[4] https://commons.wikimedia.org/wiki/File:Geogebra_software.png

Digital Assignments

- Using an IDE for coding
- Working with big data
- Ability to do machine learning tasks
- Interactive assignments

Digitalisation in Teaching

Assignments during the semester

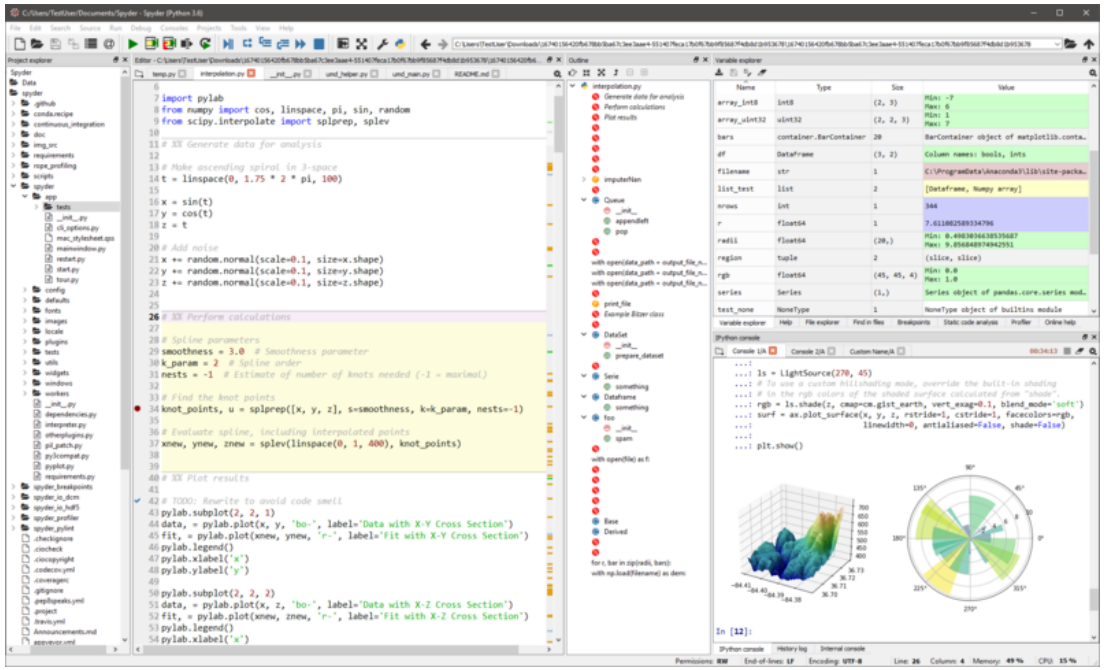

Figure: Spyder Python IDE[5]

[5] https://commons.wikimedia.org/wiki/File:Spyder-windows-screenshot.png

Aufgabe 5 (Programmieraufgabe) **[15 Punkte]**

a) Implementieren Sie die Methode

```
public static String spiegeln(String w)
```

die die Zeichen des übergebenen Strings w in umgekehrter Reihenfolge als String zurückgibt.

Beispiel:

Die Rückgabe für `spiegeln("abc")` ist der String "cba".

Hinweis:

Zur Lösung dürfen Sie folgende Methoden aus der Klasse String verwenden:

- `length()`: liefert die Länge eines Strings zurück
- `charAt(int index)`: liefert das Zeichen an der Position index zurück

```
public static String spiegeln(String w)
{
```

Figure: Example of a programming exam[6]

[6] FH Dortmund, Einführung in die Programmierung, Klausur WS15/16

Figure: Digital Exam at H-BRS

Digital Exams

Motivation

Paper-Based Exams

- Teaching and assignments are digitalised
- Exams stay paper based
- Limits type of tasks that can be performed in an exam

Digital Exams

- Using the same software tools as for assignments
- Related to assignments during the semester
- Assignments prepare for exam
- Students are used to the way tasks are done

Student

- Consistency between assignments and exam
- Fast feedback
- Personalized feedback

Digital Exams

Stakeholders II

Professor / Teacher

- Speed up grading
- Grading from everywhere
- No paper exams flying around
- Consistent grading through autograding
- Assess capability of students to solve problems
- Plagiarism detection
- Scrambling of assignments

Digital Exams

Stakeholders III

University

- Legal certainty (GDPR, digital signature, archive)
- Deployment
- Well-defined process
- Self hosted service (no data leaves the university)

Existing Solutions

Learning Management Systems

LMS

- Moodle
- Blackboard
- ILIAS / LEA
- Canvas
- OpenEdx

Learning Management Systems

What is missing?

- No advanced autograde functionality
- No common assignment format
- No coding assignments out of the box
- Everything is online

Grading Software

Autograding

- Gradescope (autograding only with paid license, autograder is closed source)
- Crowdmark (closed source)
- Autolab (grading component is closed source)
- Codio (not free, closed source)
- Nbgrader (open source)

Research

Research

Goal

Improve education:

- Individual curriculum
- Self learning
- Automatic feedback via autograding (incl. links to lecture material)
- Everybody learns at their own pace

Less routine work for staff members:

- Get rid of routine grading tasks
- More time to give feedback to complicated tasks
- More time to supervise students

Autograding Levels

- Level 1:
  Tasks with clear unambiguous solution (e.g. multiple choice, single value, fill in the blanks, etc.)
- Level 2:
  Code tasks that are easily testable (e.g. single functions with asserts, short statements, classes, etc.)
- Level 3:
  Tasks that can not be graded unambiguously, but checked against concepts (e.g. buzzwords, equations, etc.)
- Level 4:
  Tasks with no unique solution (e.g. essays, short answers with examples)

Computer-Assisted Short Answer Grading

- Computer-assisted Grading of Short Answers Using Word Embeddings and Phrase Extraction - Tim Metzler (2019)
- Recognizing textual entailment - A comprehensive evaluation of the existing state of the art techniques - Ramit Sharma (2018)
- Semantic Textual Similarity: A comparative evaluation of deep learning based models - Md Zahiduzzaman (2018)
- AI-assisted short answer grading: comprehensive classification and evaluation of the existing state of the art techniques - Evgeniya Ovchinnikova (2018)
- Evaluation of Semantic Textual Similarity Approaches for Automatic Short Answer Grading - Ramesh Kumar (2017)

Our Architecture

Architecture

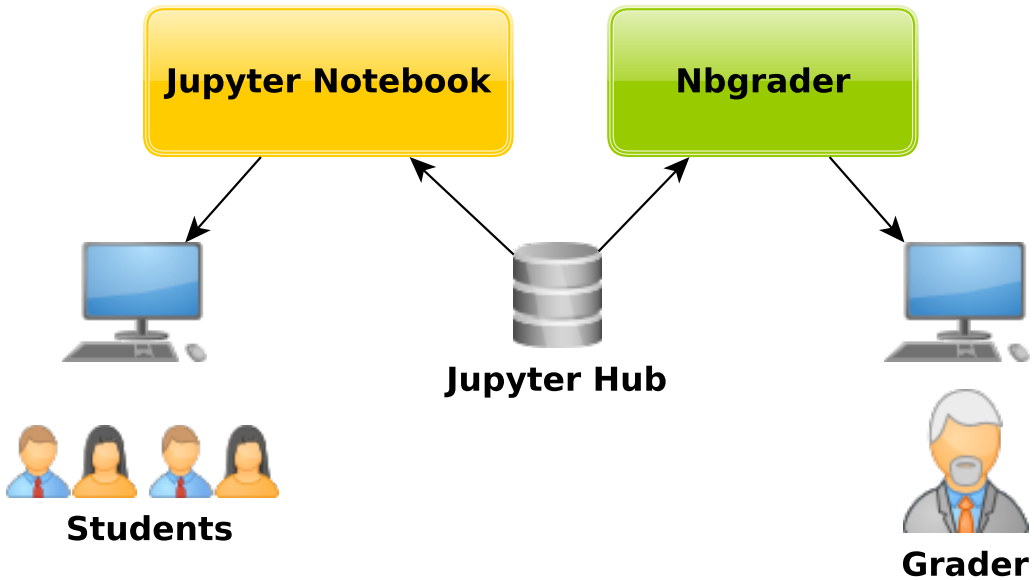

Figure: Exam architecture

Architecture

Jupyter Notebook

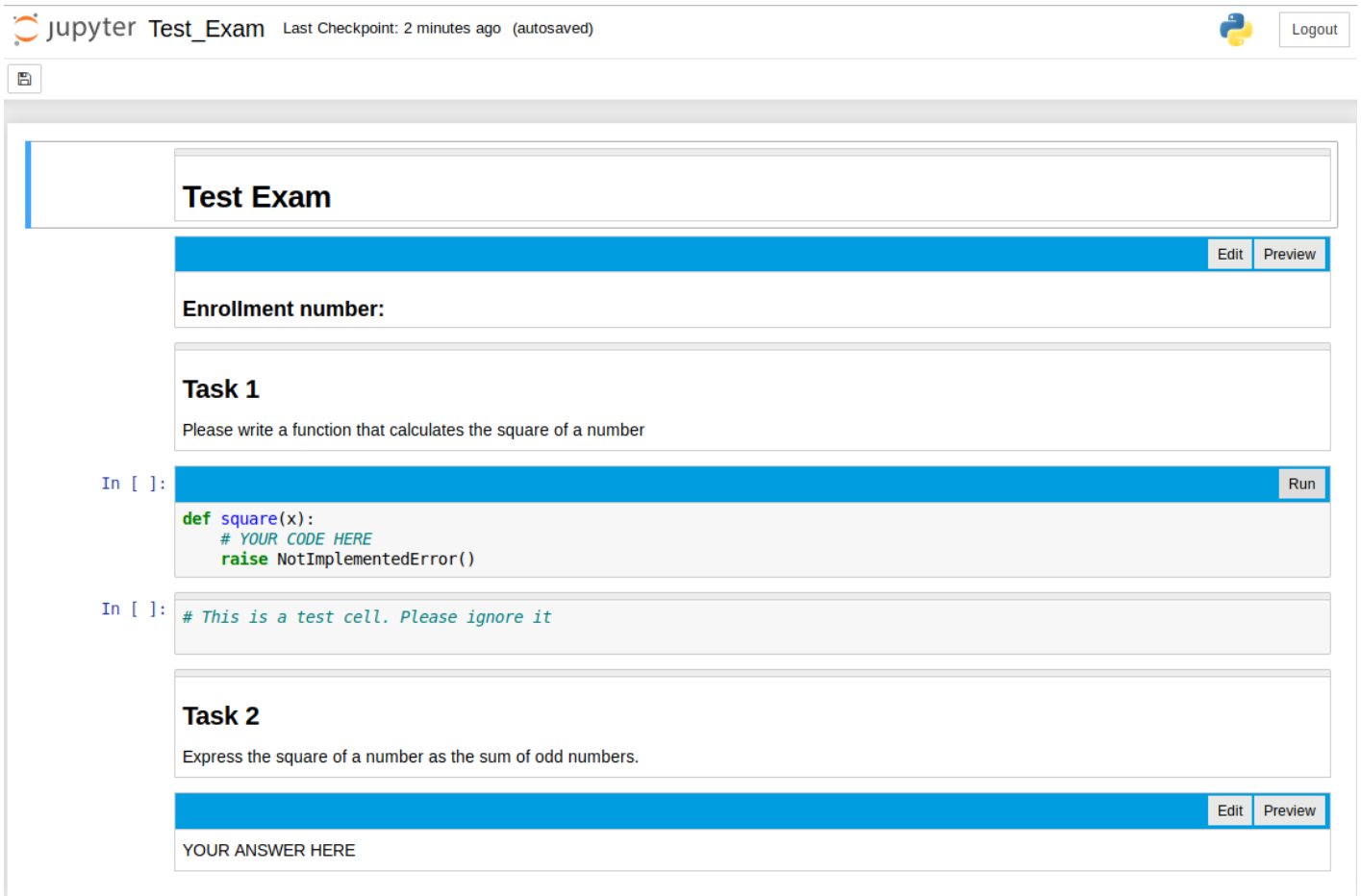

Figure: Student View of Exam

Architecture

Nbgrader

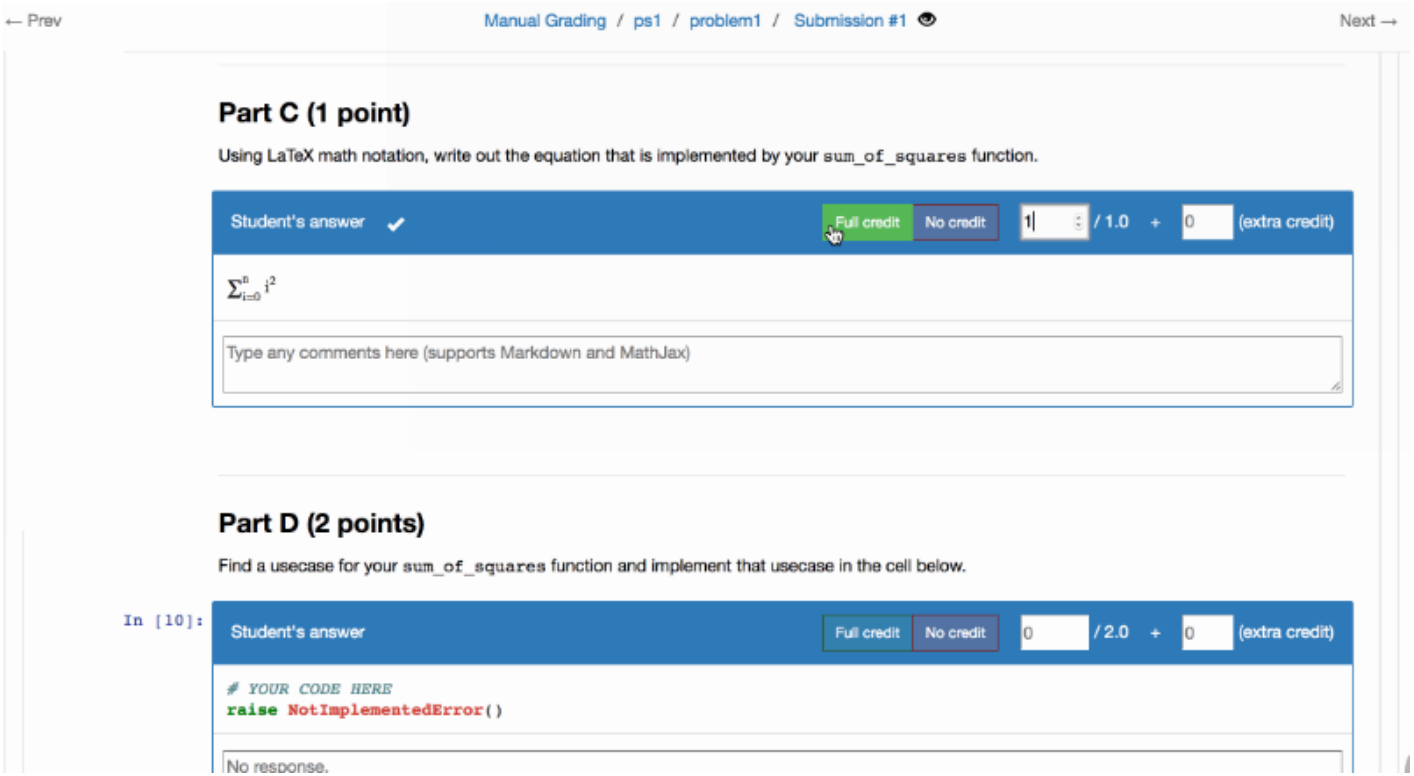

Figure: Grader View of Exam

Nbgrader + Jupyter Notebook

Jupyter Notebook

- Combine Visualisation, Code and Documentation in a single document
- Code cells and Markdown / text cells
- Built around open source components (tornado, codemirror)
- Built for Python but can be used with different kernels
- Interactive
- Extensions built in by design

Nbgrader + Jupyter Notebook

Nbgrader

- Open Source
- Autograding of code via low level tests
- Grading and feedback generation
- Works with Jupyter Notebook
- No extensions built in by design (is already an extension)

Jupyter Notebook / Hub + Nbgrader

Challenges

- Security:
  - Terminal commands via Jupyter Notebook
  - File operations via Python libraries (os, etc)
  - Javascript running in the browser
- Usability:
  - Creation of assignments with nbgrader not intuitive, especially for non-coders
  - Setting up the infrastructure
  - Going from configuration scripts to a GUI

Results & Contributions

Results

Conducted Exams

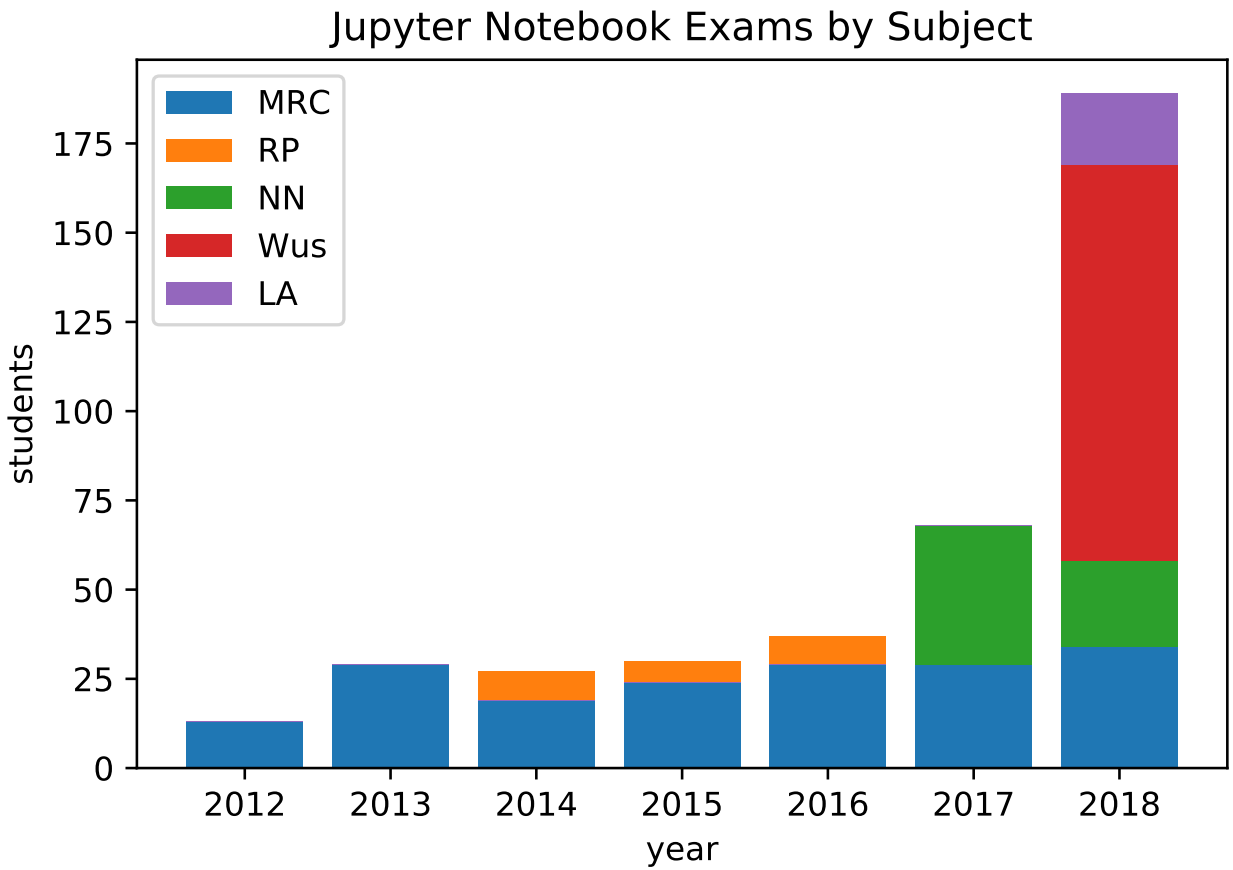

Figure: History of digital exams

Reduction of Grading Time

Wahrscheinlichkeitstheorie und Statistik:

- 119 students, 2 graders
- No multiple choice questions
- 50% of questions were autograded
- Exam was conducted on Friday 22.03 3pm
- Students got grades on Wednesday 27.03 10am

Examination Process

Established a process of how to perform digital exams

- User accounts for exam get generated
- Network gets switched to exam network
- Student logs in with one time account
- Student solves the assignment
- Student submits and receives a hashcode and timestamp to verify what we grade is what they submit
- Submissions are collected
- Network gets switched back to normal mode

Results

Process II

Name	FB02UID	Username	Password
Test User 0	**test02s**	**wus-test02s**	**76546**

Matrikel	Raum	Platz	Date
0	0	0	22.03.2019

Hashcode

Timestamp

Wahrscheinlichkeitstheorie und Statistik WS 18/19
Hochschule Bonn-Rhein-Sieg

Figure: Example of sheet students receive in the exam

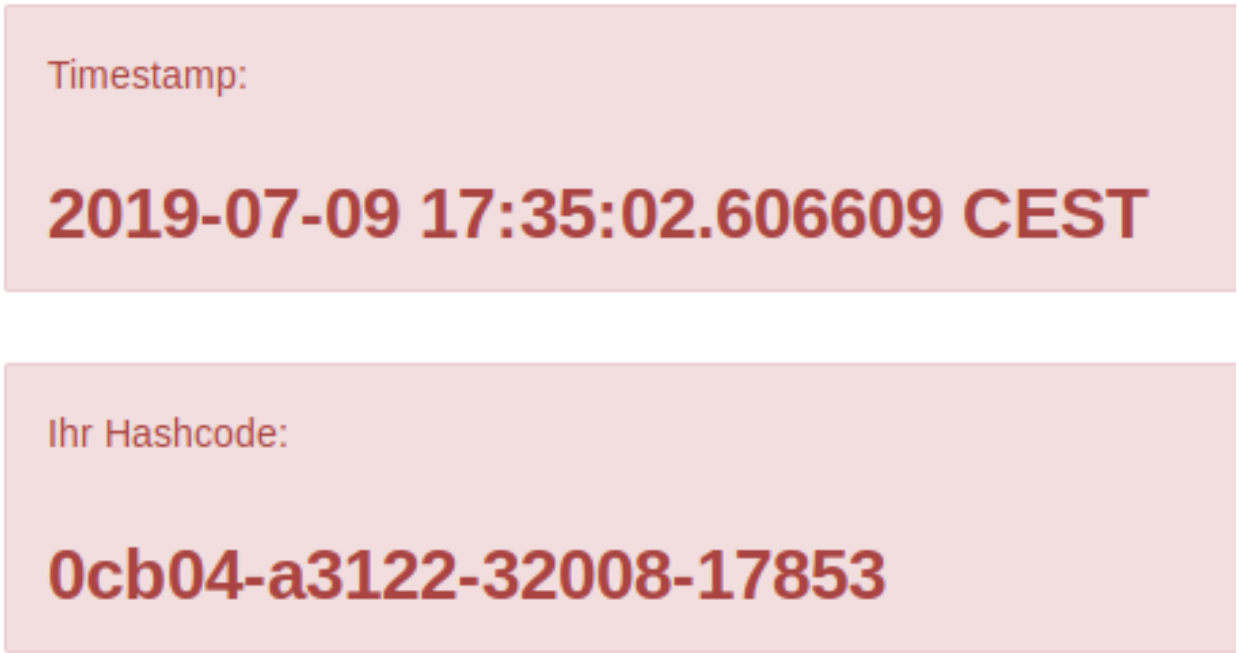

Figure: Hashcode and Timestamp

Jupyter Notebook Extensions

All Extensions are available on GitHub [7]

- New multiple choice cell type
- Restricted student view for exams
- Toolbar for exams that allows to execute hidden test code

[7] https://github.com/DigiKlausur/Jupyter-Extensions

Nbgrader Fork

Our nbgrader version is available on GitHub [8]

- Hashcode generation
- New task view for grading

[8] https://github.com/mhwasil/nbgrader

Ordinal Peer Grading

Students evaluate themselves by ranking answers from other students

Contributions

Peer Grading II

Question 2. [1 point] Given a collection of sets, what is a hitting set? What is a minimal hitting set?

Hitting set is a set of faulty components which would explain a set of symptoms. Minimal hitting set is a components of which a subset of components is not a hitting set.
A hitting set is the intersection set from the collection of sets. A minimum hitting set is a set which contains at least one element.
Hitting set - It is the set of components that cause conflicts Minimal hitting set - Minimum set that is required to diagnose a fault
Hitting set is the set of components that is responsible for a given failure . Minimum hitting set is the smallest set of components that explains the failure of the system.
A hitting set refers to a set of possibly violated assumptions, or candidates that explain an observed deviation from nominal behavior, used in diagnosis engines. A minimal hitting

Figure: Ordinal Peer Grading - Ranking

Contributions

Peer Grading III

Answers

#	Answer	Username	Score	# rankings
1	A hitting set refers to a set of possibly violated assumptions, or candidates that explain an observed deviation from nominal behavior, used in diagnosis engines. A minimal hitting set is a set for which no proper subset is also a minimal candidate.		0.750	4
2	Hitting set is the set of components that is responsible for a given failure . Minimum hitting set is the smallest set of components that explains the failure of the system.		0.563	4
3	Hitting set - It is the set of components that cause conflicts Minimal hitting set - Minimum set that is required to diagnose a fault		0.563	4
4	Hitting set is a set of faulty components which would explain a set of symptoms. Minimal hitting set is a components of which a subset of components is not a hitting set.		0.500	4

Figure: Ordinal Peer Grading - Scores

Demo

Time for questions

Thank you for your attention.