Automated Grading for Analytics

Yaron Shaposhnik

Simon Business School
IT and Business Analytics Teaching Workshop
6/7/2019
Agenda

• Brief overview of *Predictive Analytics Using Python* (CIS432)
• Motivation for automating the grading
• nbgrader and vocareum
• Development
• Issues
• Key takeaways
Overview of Predictive Analytics Using Python

• Focus: predictive modeling (machine learning) using Python

• Topics
  • **Software tools** (practical):
    Python programming, data sources (csv, json, lxml, requests, sqlite3), data manipulation (Numpy, pandas, re), Visualization (e.g., Matplotlib), Google cloud platform, scikit-learn, TensorFlow.
  • **Machine learning** (conceptual):
    applications, modeling, estimation, evaluation, ML models, algorithms.

• Assignments
  • Homeworiks (individual, structured)
    • Installation, programming, data analysis, training and evaluating models
      → Natural choice for automatic grading (mostly coding)
  • Projects (in teams, open ended)
    • Python tutorial, modeling business problems, developing a decision support system
Motivation for automated grading

• Why bother with automatic grading? (versus using TAs)

• Anticipated advantages
  • Scale
    • MSBA program: ~40 → ~60 → ~90 → ~120
  • TA work
    • Filling positions
    • Costs
    • Clearing time for office hours
  • Service
    • Immediate feedback (versus 2 weeks lead-time) → students can correct their solutions
    • Thorough grading

• Anticipated limitations
  • Qualitative questions
  • Development
  • Support students
  • Maintenance
nbgrader

• Plugin for jupyter notebooks
• Open source
• Link: https://nbgrader.readthedocs.io/en/stable/

• Live demo
  1. Jupyter notebooks
  2. Simple assignment: master notebook, test notebook
  3. Real assignment (HW5): test notebook
nbgrader – deployment options

- Offline
  - Collect solutions through a learning management system (Blackboard, Moodle, ...)
  - Run code on my computer
  - Disadvantages: no immediate feedback, run manually, late submissions, notify students about grades, ...

- Online
  - Buy/rent server that runs nbgrader
  - Disadvantages: costs, setup, maintenance, capacity issues

- Online – 3rd party
  - Disadvantages: Cost (Vocareum; 30$/student)
Vocareum

- Instructor view
- Student view
Development

• Every deployment option requires
  • Writing questions
  • Solutions
  • Grading code

• Grading code
  • Implement function and compare input to output
  • Request to assign to a variable a dataframe/dictionary/list/object and compare against offline object
What did (some) students think

1. “The assignments were graded based on programs which were terrible in grading and often give the wrong grade for each question.
   • “Terrible in grading” → debugging issues
   • “Wrong grade” → some confusion about submission penalties due to using maximal grade, and changing the grading code while the system is live

2. I think the Professor should make use of the TAs to be the ones actually doing the grading.”
   • There isn’t enough capacity to carefully grade 100x5 coding assignment

3. “… the online homework automatic grading platform is unwieldy and unhelpful as a learning tool and only exists as a way to save grading time for him and the TAs.”
   • Didn’t save me time..
What did (some) students think (2)

3. “Homework takes ages to complete because of debugging issues”
   • The grading code should provide good feedback

4. “... Grade code manually, Vocareum is way too rigid in it's structure and doesn't allow for partial credit. It can discourage students from working at or learning the material because they won't get credit for the hours of work they put into it. A lot of those hours are spent debugging the code to make it match an arbitrary format that you, as the professor, have defined even though someone might have done it a different way that works just the same.”
   • “Too rigid” – agree (example, comparing dataframes)
   • “Partial credit” – agree (need to break down questions to smaller parts)
   • “Match arbitrary format” – similar to unit tests

Overall course/instructor evaluations: 3.95/5 and 3.97/5
Fine details

• Early submission bonus
  • Inflated grades vs. early feedback
• Late submission penalty and deadline
  • End of term? End of week? Fixed point reduction per day?
• Include validation code? yes
• Make grading code transparent? Yes
• Maximal grade vs. latest grade?
• Plagiarism – detect? How to handle it?
• Post solutions?
  • Students request and could learn from it vs. avoiding plagiarism
Key takeaways

• May cause dissatisfaction among students
• The grading code should provide good feedback
• Tolerance in grading
• Partial grading (vs. all or nothing)
• Make grading code available
• Non-trivial effort that requires time and research
• Request for additional resources if possible
Thank you!

Email: yaron@simon.rochester.edu