# CSCI 567 Spring 2018 Syllabus

**Course meetings:** You are responsible for everything covered in lectures, including administrative announcements. You are not obligated to attend the discussion in which you are enrolled, but you are required to attend or view a discussion each week. Lectures begin on week one, discussions on week two.

**Strongly Recommended Textbooks**: Kevin P. Murphy*, Machine Learning: A Probabilistic Perspective*
Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition*
**Also Recommended Textbooks**: A "Recommended Textbooks" list is available at the end of this syllabus. These cover both mathematical preliminaries and machine learning topics.
**Gradebook**: blackboard.usc.edu
**Forums**:`piazza.com/usc/spring2018/csci567` -- your instructor will sign you up in the first week.

Blackboard will be used for posting of grades. Messages that do not need a particular instructor's direct attention should be posted to Piazza with the appropriate privacy setting. If you email your instructor, you *must* include the substring "CSCI 567" to begin a *meaningful* subject line and have tried to resolve the issue appropriately otherwise (e.g., questions about course material should be posted to Piazza first, using email only after an appropriate amount of time has passed without a response.) Such emails must be sent from your USC email account.

**Course Description**
Statistical methods for building intelligent and adaptive systems that improve performance from experiences; Focus on theoretical understanding of these methods and their computational implications.
**Recommended preparation:** Undergraduate level training or coursework in linear algebra, multivariate calculus, basic probability and statistics; an undergraduate level course in Artificial Intelligence may be helpful but is not required.

**Grade Calculations**:

| Artifact | Weight | Date | Time |
|---|---|---|---|
| Quiz One | 15% | February 23 | 4:10 (arrival deadline) |
| Quiz Two | 20% | April 6 | 4:10 (arrival deadline) |
| Quiz Three | 25% | April 27 | 4:10 (arrival deadline) |
| Programming Assignments | 20% | various | 11:59pm Pacific (according to us) |
| Problem Sets | 20% | various | 11:59pm Pacific (according to us) |

The total points you earn from assignments (scaled to 0 through 100) will be capped at 30 points above the weighted mean of your exam scores (scaled to 0 through 100). For example, if you average 90% on the programming assignments, and your weighted quiz score is 50%, then your assignment average is treated as only 80 (because that's 30 more than the mean). In other words, your assignment scores won't count fully if you can't show from your exam scores that you learned what you should have from the assignments.

**Exams**

You will be provided with paper on which to take the exam. Exams will be individual effort, closed-book and closed-notes. You will not be permitted any electronics on the exams, including digital watches. Any requirements for you to bring to the exam will be announced well in advance.

Students requiring alternate exam arrangements must make such requests within the first two weeks of the term, or as soon as possible after knowing of the conflict or requirement. Students who will be taking exams remotely must check the testing center availability within the first two weeks of the semester and email the instructor *within that time period* if the testing center cannot accommodate you on our test day.

**Late Policy and Grace Days**

Four times during the semester, a student may extend the due date of a *programming assignment* by twenty four hours without needing prior permission. These are known as “grace days." In order to use a grace day, you must submit a form (which will be provided); if you use this form, the timestamp you submit the form is considered the time the assignment is submitted. You must have already committed and pushed your code to use this form, and you will need to provide the commit’s SHA as part of filling out the form. Students must complete the syllabus quiz and git survey by January 14 in order to be allowed grace days during the semester.

Please note that grace days are *in place of* “excused late'” submissions, not in addition to. Once you have used your grace days, any assignments submitted after the deadline will be scored as a zero.

Some assignments may have a maximum number of grace days permitted for it.

Note: There is no grace period. Even if you submit the form a few minutes after the deadline, you will need to use a grace day (even if the wireless network in your dorm room is down or you have a github issue, etc.). It is your job to be on time and not cut it too close. Remember Murphy's Law and leave time for things to “go wrong.” The lack of grace period also applies to the form: if you commit and push at 11:55 PM the evening after the on-time date, but don’t submit the form until ten minutes later, you will need to use two grace days.

Tracking grace days is *your* responsibility. When you submit the form, you will receive an email confirmation. That is your receipt.

### Grading Reconsideration

Reconsideration requests for any non-quiz graded artifact must be made within 72 hours of our release of grades for the item. A form will be available on the course Piazza page for you to use to explain your reconsideration request. You must fill out the form within that time period. Once the reconsideration period has passed, grades for each artifact not contested are considered final.

Requests for grading reconsideration submitted in any other fashion will not be considered; students requesting reconsideration in any other matter will be declared ineligible for such, even if you also submit a valid request.

Before we release grades, we will release a rubric (problem sets) or grading script (programming assignments). Before you request reconsideration for a programming assignment, you must check the following:

- *Was a collection note placed for the correct submission? If so, you should have alerted us within 72 hours of when the collection note was placed. See section "Homework Collection Policies," below.*
- *If a collection note was placed on the correct one, grade it using the rubric we provide. Does your grading match ours? If not, that's your regrade request.*
- *If the right submission was collected and your grade on blackboard reflects the same score as the rubric for it, do not submit a regrade request -- you might not be satisfied with your score, but that doesn't mean the score was incorrect.*

Students submitting spurious reconsideration requests will be denied both the regrade for that artifact and the opportunity to submit future regrade requests. Such students may lose points as well. The following are *never* acceptable reconsideration requests:

- You submitted the wrong item.
    - Note that this is different from whether we collected the wrong item; it is acceptable to alert us that we collected the wrong item.
- You failed to follow submission guidelines
- You are unhappy with the grading rubric. If and when you teach or help to teach a course in machine learning, you will have input on what mistakes are worth what deductions.

In short, a regrade request is for cases where you believe we graded *incorrectly*, such as marking something wrong that is right, not for cases where you are unsatisfied with a correct result.

Quizzes will have a viewing period in which students can see their graded quiz and may submit grade reconsideration requests at that time.

**Academic Conduct**

Please keep in mind that CSCI 567 is a large course with staff that work very hard to run it. Abusive behavior towards course staff will not be tolerated and will result in a referral to the relevant office in cases that warrant it. We strive to run the class well, but recognize that we cannot do things the same way we could if there were 20 students total.

In addition, all students are expected to have read and understood the university's academic integrity guidelines for graduate students, available at https://sjacs.usc.edu/files/2015/03/GradIntegrity.pdf. Related items appearing in this syllabus are a supplement, not a replacement, for those policies.

It is the expectation that each graded artifact submitted represents the sole work and understanding of the author turning it in. Course staff may choose to discuss submissions with a student and adjust the score based on how well the student can explain the work they submitted.

The university's recommended sanction for plagiarism and similar offenses is an F in the course on first offense[1]. Every suspected incident of academic dishonesty will be reported to the relevant office and no suggestion for lighter punishment will be made and requests for me to do such will not be entertained. The time to consider the consequence of the potential fail in the course is **before** committing an infraction. In any course with a pending academic honesty violation report, you may not drop the course; if you drop the course and are later discovered to have violated the academic honesty policy, you will be re-enrolled.

I recognize students talk about assignments and may want to help one another. The help you receive from classmates must be limited to *conceptual* help -- how an algorithm you need to implement works as a "big picture" rather than at the level of code. You should *never* show your code to anyone else, look at anyone else's code, or seek to find code that solves the problem you are working on *for any reason*. Yes, this means that if you show your code to a classmate and he or she submits it as his or her own, you will be reported for an academic conduct violation.

Do not, for any reason, post your code or other forms of homework solutions on any portion of the internet viewable by other students. If another student finds it and submits it, our similarity detection software will notice and you can get in trouble for it. Note that we may use assignments from this semester in future terms, so this applies after the semester; if a recruiter needs to see your code, post it to a *private* github repository and grant the recruiter access.

[1] See https://studentaffairs.usc.edu/files/2015/09/appendix_a.pdf for recommended sanctions for *undergraduates*, and observe that graduate students may be sanctioned more severely.

If you receive help on an artifact that will be graded from anyone else affiliated with the class, follow the *Kenny Loggins Rule*:

- You may discuss high-level ideas and receive hints regarding how to solve portions of the assignments. However, neither party may keep any written record from this discussion. Doing so places both parties in a “danger zone.” Afterwards, take a long break and do something unrelated to the course (watching a movie, for example). You may now return to your assignment.

- When you write a section of code based on help received, add a comment acknowledging the help, including the substring “assistance from” as part. *If you do not include the substring “assistance from” in your comments, it will be treated as uncited collaboration. Be sure to spell the phrase “assistance from” correctly in your comments*.

You are **explicitly prohibited** from seeking help outside of course resources for the programming projects and problem sets. The following is an exhaustive list of “course resources.”

- The instructor and the TAs.

- Your fellow students. Remember that this means discussing concepts, not sharing code.

- Any textbook listed as recommended by the instructor. Please note that this means *the textbooks* and not supplemental material, such as any published solution key to exercises. Textbooks not on either list are not acceptable resources without prior written permission from the instructor.

- Material posted by the instructor or TAs to the course Piazza page for this semester. Remember that any posts on Piazza are required to conform as well; for example, do not post code in a public viewable section. Anything posted by a student to Piazza is treated no differently than if the student said it elsewhere.

- Course lectures and discussions, along with any notes provided by instructors.

The above means that if you find a resource related to the course that *is not* considered a course resource, and you use it towards work that you submit, you risk an academic honesty violation in the process. This includes previous semesters’ students, lectures from other schools, and online sources.

**Homework Collection Policies**

Problem sets will not permit the use of grace days and will be collected via the the DEN Desire2Learn system. Problem sets must be typewritten or *neatly* handwritten when submitted; if the grader cannot read your handwriting on a problem, they may elect to grade it as a zero.

Programming assignments are collected via a private repository on github that we provide each student. Upon successful completion of the git survey and syllabus quiz[2], you will get a private repository in the Spring 2018 CSCI 567 organization, which is necessary to submit homework assignments. If you do not successfully complete the quiz by January 14, you will not be allowed to use grace days during the semester.

When we collect programming assignments, we will grade your on-time submission, defined as the last commit pushed to your repository prior to the deadline, if any of the following apply:

- You do not submit a grace day usage form prior to the absolute final deadline in a project.
- You submit two or more grace day usage forms for the project. Note that you will *also* be charged the late days for both requests if you do this also.
- You submit a grace day usage form with an incorrect or missing SHA. Note that you will *also* be charged the late days for the request if you do this.
- You submit a grace day usage form that uses grace days you do not have available. For example, if you have only one grace day left, and you submit a form asking to use two, we will grade your on-time submission and you will have zero grace days left.
- You reply to the message that we will leave on *all* on-time submissions. The *only* exception to this is if you reply to that message and ask us to collect an *earlier* commit or if we erroneously leave a comment on a commit that is not the on-time submission.

At some point after the initial deadline has passed, we will perform our initial collect of each homework, leaving a comment on each student's last commit prior to the deadline -- that is the item we initially collect.

If you believe there is an error, you must do the following:

- If you are using a late day, disregard the message entirely. It isn't an error in this case.
- Clone a copy of your repo fresh from that commit.
- Check to see if that is what you want graded.
- **Only then** if you believe an error has occurred, leave a comment reply to the comment left for you.

As described above in grade reconsiderations, us collecting the wrong repository, and being alerted after grades are released, *will not* be accepted as a regrade request. You *must* alert us within 72 hours of our collecting the assignment that we have collected the wrong one. Once 72 hours after the collection has passed, we will consider it that you are satisfied with the collection.

[2] The survey/quiz for Spring 2018's CSCI 567 is available at https://docs.google.com/forms/d/e/1FAIpQLSfpyeZGejHkGYJ1kMWh2TPY-npV0CIMueV9nob4q99WRdI8mQ/viewform

## CSCI 567 Projected Course Schedule: A Weekly Breakdown

Please note that this is a *projected* schedule and is subject to change. Any changes will be announced via the course Piazza page. Each row is a Monday or Wednesday in the schedule.

| Date | Topic | Reading | Notes |
|---|---|---|---|
| 1/8 | Course Overview<br>ML overview<br>Nearest Neighbor classification | | |
| | Core ML concepts; typical steps to developing a ML system | | |
| 1/15 | **No class: MLK Day** | | |
| | Linear Regression | [MLaPP] 1.4.5, 7.1-7.3, 7.5.1, 7.5.2, 7.5.4, 7.6 | |
| 1/22 | Regression with nonlinear basis<br>Regularized regression | [MLaPP] 1.4.7, 1.4.8<br>[ESL] 7.1, 7.2, 7.3, 7.10 | |
| | Linear discriminant analysis<br>Perceptron | [MLaPP] 4.2.1 - 4.2.5, 8.5.1-8.5.4 | |
| 1/29 | Logistic regression | [MLaPP] 1.4.6, 8.1-8.3<br>[ESL] 4.1-4.2, 4.4 | |
| | softmax; multi-way classification | | |
| 2/5 | neural networks / mlp | [MLaPP] 16.5.1-16.5.6, 28<br>[ESL] 11.3-11.7 | |
| | DNN, CNN, RNN, and LSTM | | |
| 2/12 | Kernel Methods | [MLaPP] 14.1, 14.2.1-14.2.4, 14.4.1, 14.4.3<br>[ESL] 5.8, 6.3, 6.7 | |
| | SVM | [MLaPP] 14.5.2-14.5.4<br>[ESL] 12.1-12.3 | |
| Week 7 | **No class: President’s Day** | | |
| 2/21 (Wed) | **No lecture**: quiz on Friday | | Quiz 1: 2/23 |

| | | | |
|---|---|---|---|
| 2/26 | Decision Trees | | |
| | Boosting/ensemble | [MLaPP] 16.4.1-16.4.5, 16.4.8, 16.4.9<br>[ESL] 16.3 | |
| 3/5 | Basic learning theory | | |
| | clustering, mixture models | [MLaPP] 11.1-11.3, 11.4.1-11.4.4, 11.5<br>[ESL] 14.3.1-14.3.9, 8.5 | |
| 3/11-3/18<br>Spring Break | | | |
| Week 10<br>3/19 | mixture models / density estimation | | |
| | Generative models / naive Bayes | [MLaPP] 3.5<br>[ESL] 6.6.3 | |
| 3/26 | Hidden Markov models | [MLaPP] 17.1-17.4, 17.5.1-17.5.2 | |
| | Hierarchical Models ;<br>topic models | [MLaPP] 10.1, 10.2.1-10.2.3, 10.3-10.5 | |
| 4/2 | dimensionality reduction and visualization | [MLaPP] 12.2<br>[ESL] 14.5.1 | |
| | **No lecture**: quiz on Friday. | | Quiz 2: 4/6 |
| 4/9 | recommender systems and other applications | | |
| | large-scale ML systems | | |
| 4/16 | Introduction to reinforcement systems | | |
| | To be determined | | |
| 4/23 | To be determined | | |
| | | | Quiz 3: 4/27 |

## Recommended Textbooks

*Remember that anything taken explicit from these towards a problem set or programming assignment must still be cited.*

**Math Preparation:**

- Calculus: Prof. Strang's free online textbook. https://ocw.mit.edu/resources/res-18-001-calculus-online-textbook-spring-2005/
- Linear Algebra: Also from Prof. Strang: https://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/index.htm
- Probability and Statistics: Orloff and Bloom: https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/
- The Matrix Cookbook: http://www.math.uwaterloo.ca/~hwolkowi//matrixcookbook.pdf

**Additional Machine Learning Resources:**

- Max Welling's quick intro: https://www.ics.uci.edu/~welling/teaching/ICS273Afall11/IntroMLBook.pdf
- Alex Smolas's book: http://alex.smola.org/drafts/thebook.pdf
- Gareth James et al's *An Introduction to Statistical Learning* : http://www-bcf.usc.edu/~gareth/ISL/ISLR%20First%20Printing.pdf
- A Course in Machine Learning by Hal Daumé III: http://ciml.info
- *Bayesian Reasoning and Machine Learning* by David Barber: http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=Brml.HomePage
- *Pattern Recognition and Machine Learning* by C Bishop
- Andrew Moore's tutorials: https://www.autonlab.org/tutorials
- *Pattern Classification* by Duda, Hart and Stork
- *All of Statistics* by L. Wasserman