Modern Statistics in Machine Learning

Spring 2018

Instructor:	Arnold Stromberg	**Time:**	Th : 3:30 - 5:30
Email:	astro11@email.uky.edu	**Place:**	MDS 337

Course Pages: A GitHub repository will be mainted for the course.

1. `https://github.com/joshuawlambert/STA695`

Office Hours: By appointment with Eric or Jin. Eric and Jin's email are eric.roemmele@uky.edu and jin.xie@uky.edu, respectively.

Main References: There is one recommended textbook (below) that's more for your own reference. Other suggested references will be provided. A used textbook can be purchased on Amazon for about $50. It is also available free online at `https://web.stanford.edu/~hastie/Papers/ESLII.pdf`.

- Hastie et. al (2008) *The Elements of Statistical Learning*. New York, NY: Springer.

Objectives: We'll study (in more detail) four statistical machine learning methods for prediction and classification : tree methods, support vector machines, LDA variants, and neural networks. Students are responsible for learning other methods. Focus will be more on correct prediction or classification of an outcome, as opposed to traditional statistical inference. Learning Python, GitHub, Kaggle, and high performance computing on Amazon Web Services will also be emphasized. Lastly, different types of data from Kaggle, such as text and image data, will be analyzed.

Coursework: The first two weeks will be spent learning the necessary prerequisites of Python, Jupyter Notebook, GitHub, and Linux. After that, the next 12 weeks are divided into four sections of three weeks. For the section, there will be an overarching data set that we'll analyze. Students will be divided into groups of three (subject to change depending on enrollment), and they'll work together to build a predictive model. Each group can choose any analysis method, but that method cannot be repeated for the rest of the semester. At the beginning of the three week section, there will be a presentation on an analysis method in the first hour. In the second hour, a volunteer will lead the class in a data exploration . In the second week, each group will give a short proposal presentation on what they plan to do, along with interesting features in the data. The last week of the section will be devoted to presentations by each group. Each group should prepare a twenty minute presentation on the basic theory of your learning method, code, and results. For the theory, make a ten minute presentation on the basic details (no need for complicated mathematics). Then for the results/code, in essence, tell us how you did (in terms of prediction/misclassifcation error), what you did, and what interesting features you found in the data. Also, briefly go over how to implement in Python. Please make your code reproducible and well-commented. Moreover, upload to Kaggle and the GitHub repository. Groups with the smallest prediction or misclassification error will win a car!* The last three weeks of the course are to be determined.

Grading: Grading will be based primarily on completion of assignment and presentation (90%). The other 10% will be be based on attendance.

Class Policy: Regular attendance is essential and expected.

Grand Prize : Person who fits a neural net to beat this person (`https://www.youtube.com/watch?v=Ipi40cb_RsI`) at Mario Kart will win 100 Grand**.

*No cars will be given out.

** The candy bar.

Course Outline:

Week	Hour 1	Hour 2	Other and Sources
1	Intro to Course, AWS	Linux/Github, Jupyter	GitHub/AWS account,Chapter 2,Chapter 7
2	Intro Python	Advanced Python	Familiarize with Kaggle
3	Intro to Tree Methods	Kaggle Data - Ames Housing Data	Chap 9.2 to 10, Chap 15, Chap 8.7
4	Proposal Presentation on Data	Meet with Team	Finish Code, put on Kaggle/GitHub
5	Final Presentations	Final Presentations	NA
6	Intro to SVM/LDA	Kaggle Data - Expedia	Chap 4,Chap 12
7	Proposal Presentation on Data	Meet with Team	Finish Code, put on Kaggle/GitHub
8	Final Presentations	Final Presentations	NA
9	Neural Net Intro	Kaggle Data - Image Recognition	Chap 11
10	Proposal Presentations	Meet with Team	Finish Code, put on Kaggle/GitHub
11	Final Presentations	Final Presentations	NA
12-14	TBD	TBD	NA

Other Suggested Methods/Sources:

- **LASSO,Ridge,Elastic Net :**
  - Chapter 3
  - https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python/
- **GAM**
  - Read Chap 5,6, and 9 - 9.1.
  - https://github.com/dswah/pyGAM
  - https://www.youtube.com/watch?v=f9Rj6SHPHUU&t=13s
- **Nearest Neighbors**
  - Chap 2.3 , Chap 13.3 - 13.5
  - http://scikit-learn.org/stable/modules/neighbors.html

- **Mixtures of Normals**
  - Chap 6.8, 9.5, 13.2.3
  - http://scikit-learn.org/stable/modules/mixture.html
- **Ensemble/Stacking Methods**
  - Chap 8.8, Chap 16
  - http://scikit-learn.org/stable/modules/ensemble.html
- **CART**
  - http://scikit-learn.org/stable/modules/tree.html
  - https://www.youtube.com/watch?v=p17C9q2M00Q
- **Neural Nets**
  - https://www.youtube.com/watch?v=gwitf7ABtK8
  - https://www.youtube.com/watch?v=oYbVFhK_olY
  - https://www.youtube.com/watch?v=tIeHLnjs5U8&t=468s
- **SVM**
  - http://scikit-learn.org/stable/modules/svm.html
  - https://www.youtube.com/watch?v=g8D5YL6c0SE

If there is more techniques and good sources to add to this list, please let me know!