Predictive Analytics using Python (CIS432)

Simon Business School

IT Teaching Workshop 2018

6/1/2018
Talk outline

• Overview
• Lectures
• Homework assignments
• Projects
• Evaluations
Overview

• Title: Predictive Analytics using Python (CIS432)
• Introductory course to machine learning using Python
• Topics
  • Exploratory data-analysis
  • Fundamental concepts in statistical modeling
  • Machine learning applications and algorithms
  • Software tools and online platforms
• Approach: applied, hands-on, broad
• 9 weeks (quarters system)
Lectures

• Weeks 0-3: exploratory data analysis using Python
  • Python programming
  • Development environments: Spyder, Jupyter, terminal and OS shell commands
  • csv, json, lxml, requests, sqlite3 – working with data sources
  • Numpy: arrays, vector and matrix operations (math, reshape, split, concatenate)
  • Pandas: series and data frames, indexing, manipulating data (merge, pivot, concatenate, group by), visualization
  • Matplotlib – histograms, scatter plots, pie charts, 3D, multiple plots per figure, ...
  • SciPy – math operations (probabilities distributions and computing distances)
  • re – regular expression
  • pydot_ng – plotting decision trees and transition matrixes
Lectures – cont.

• Week 4:
  • Google Cloud Platform (creating instances with Compute Engine, SSH keys, remote connection, transferring files, Google storage, Datalab, BigQuery)

• Weeks 5-9: Machine learning - Algorithms and applications
  • Classification (Classification trees, Support vector machines, KNN)
  • Regression (Regression trees, Support vector regression)
  • Ensemble methods (Bootstrap aggregation, Random forests, Boosting)
  • Recommendation systems (baselines predictors, collaborative filtering, content-based filtering, matrix factorization, graphical models)
  • Unsupervised learning: PageRank, Clustering (K-Means, Hierarchical clustering, Mixture models, and the EM algorithm), Association rules mining and the Apriori algorithm
  • Optimization (unified view of learning) and introduction to TensorFlow
Homework assignments

• 7 individual homework assignments
• Analyzing datasets and exploring algorithms
• Submission of jupyter notebook
• Structured and straightforward (not trivial, significant amount of work and troubleshooting)
• 10-20 hours each (too much..)
• Next time: automatic grading using jupyter server (improved learning experience, immediate feedback, more office hours)
Mini-projects

• Previously: single open ended project → large variability in effort and outcome

• Instead, 3 mini-projects
  • More guided but still open-ended and (hopefully) more realistic
  • Opportunity to do research
  • Work in teams
Mini-projects 1/3

• Mini 1 (~20 hours)
• Compare the performance of 7 algorithms on 54 publically available datasets

• Objectives:
  • Work with a new data format,
  • automate,
  • methodology (training, evaluation),
  • research (analysis, limitations of the analysis, difference from typical ML applications, etc.)

• Next time: postpone deadline by a couple of weeks
Mini-projects 2/3

• Mini 2 (3-20 hours)
• Write Jupyter notebook tutorial for a Python library

• Objectives:
  • self study
  • write-ups (formatting, conciseness)
  • peer reviews (think about what makes a work good or bad, learn other packages)

• Next time: limit weight for peer reviews..
Mini-projects 3/3

• Mini 3 (20 hours)
  • Team-based collaborative project
  • Design a product that uses ML technologies (Home security system)
  • Split work to modules (market research, design, uploading videos to the cloud, use ML models to count how many people are in each image, anomaly detection)
  • **Pros**: closer to real-world, self study of ML algorithms, troubleshooting, provide an experience that resembles working on business application of ML, presentation
  • **Cons**: grading, not all work is related ML, no integration
  • Experiment on UG section (remove some of the material..)
Evaluation

• Participation (class, office hours, slack)
• Homeworks
• Mini-projects
• Exams
  • Closed-book in writing
  • Midterm – Python (write code and output of code)
  • Final – conceptual questions and application of algorithms
Thank you!

• Comments / suggestions?

• Email:
  • yaron.shaposhnik@gmail.com, or
  • yaron@simon.rochester.edu