



SAVE THE DATE

סדנה בנושא: תשתיות חישוביות במדעי הנתונים

הסדנה תועבר בזום, באנגלית

מרצים: מרקו סקוטרי ומאורו מלבסטיה

פרופ' מרקו סקוטרי חוקר בכיר במוסד מחקר בתחום מדעי הנתונים בלוגנו, שווייץ. הוא מומחה בינלאומי בשיטות חישוביות ואחראי לפיתוח bnlearn שמיישמת ב R רשתת בייזניות. פרטים על

פרופ' סקוטרי ב: <https://loop.frontiersin.org/people/13385/bio>

מאורו מלבסטיה יזם סדרתי בתחומי מחשוב ענן ופיתוח תוכנה עם דגש על סביבות פיתוח.

מועדים

24.5, 17.5, 10.5

שעות

14:00-17:00

פרטים

ירדן מל: yarden@aeai.org.il

רקע

מדעני נתונים מקציע מאוד מבוקש שמשלב יכולות בתחום האנליטיקה והסטטיסטיקה, הבנתצרכים עסקיים ושימוש בתשתיות חישוביות לפיתוח ויישום של מודלים ושיטות שונות.

הסדנה מכוונת לספק למתעניינים ביישומים של מדעי הנתונים הבנה בתשתיות מחשוב, כלים ואמצעים המספקים תשתית לעבודת הפיתוח וההטמעה בשטח של מודלים ושיטות אנליטיות.

קהל יעד

מדעני נתונים, סטטיסטיקאים, מהנדסי נתונים, מפתחי ומיישמי אנליטיקה בתחומים שונים

מטרה

להקנות למשתתפים ידע ומיומנות בתשתיות חישוביות הנדרשות ליישום מודלים אנליטיים. הסדנה כרוכה בתשלום בסיסי.

שיטת עבודה בסדנה

שילוב של ידע תאורטי והתנסות מעשית. מומלץ שלכל משתתף יהיה מחשב זמין במהלך הסדנה

חומר לימוד

שקפים ותרגילים על בסיס הספר A Pragmatic Programmer for Machine Learning

<https://www.taylorfrancis.com/books/mono/10.1201/9780429292835/pragmatic-programmer-machine-learning-marco-scutari-mauro-malvestio>

נושאי לימוד

הסדנה תכלול נושאים מהספר של סקוטרי ומלבסטיה. תכנית מפורטת, לכל מפגש תיקבע בהמשך

Contents

Preface	vii
1 What is This Book About?	1
1.1 Machine Learning	1
1.2 Data Science	4
1.3 Software Engineering	6
1.4 How Do They Go Together?	8
I Foundations of Scientific Computing	11
2 Hardware Architectures	13
2.1 Types of Hardware	14
2.1.1 Compute	14
2.1.2 Memory	20
2.1.3 Connections	24
2.2 Making Hardware Live Up to Expectations	26
2.3 Local vs Remote Hardware	28
2.4 Choosing the Right Hardware for the Job	31
3 Variable Types and Data Structures	35
3.1 Variable Types	36
3.1.1 Integers	36
3.1.2 Floating Point	40
3.1.3 Strings	47
3.2 Data Structures	48
3.2.1 Vectors and Lists	49
3.2.2 Representing Data with Data Frames	51
3.2.3 Dense and Sparse Matrices	53
3.3 Choosing the Right Variable Types for the Job	56

3.4 Choosing the Right Data Structures for the Job	61
4 Analysis of Algorithms	63
4.1 Writing Pseudocode	63
4.2 Computational Complexity and Big- <i>O</i> Notation	66
4.3 Big- <i>O</i> Notation and Benchmarking	70
4.4 Algorithm Analysis for Machine Learning	72
4.5 Some Examples of Algorithm Analysis	73
4.5.1 Estimating Linear Regression Models	74
4.5.2 Sparse Matrices Representation	80
4.5.3 Uniform Simulations of Directed Acyclic Graphs	84
4.6 The Relationship Between Big- <i>O</i> Notation and Real-World Performance	91
II Best Practices for Machine Learning Pipelines	93
5 Designing and Structuring Pipelines	95
5.1 Data as Code	95
5.2 Technical Debt	98
5.2.1 At the Data Level	99
5.2.2 At the Model Level	101
5.2.3 At the Architecture (Design) Level	103
5.2.4 At the Code Level	105
5.3 Machine Learning Pipeline	107
5.3.1 Project Scoping	110
5.3.2 Producing a Baseline Implementation	114
5.3.3 Data Ingestion and Preparation	116
5.3.4 Model Training, Evaluation and Validation	117
5.3.5 Deployment, Serving and Inference	121
5.3.6 Monitoring, Logging and Reporting	122
6 Writing Machine Learning Code	129
6.1 Choosing Languages and Libraries	130

Contents v

6.2	Naming Things	133
6.3	Coding Styles and Coding Standards	136
6.4	Filesystem Structure	139
6.5	Effective Versioning	142
6.6	Code Review	146
6.7	Refactoring	150
6.8	Reworking Academic Code: An Example	152
7	Packaging and Deploying Pipelines	161
7.1	Model Packaging	161
7.1.1	Standalone Packaging	162
7.1.2	Programming Language Package Managers	162
7.1.3	Virtual Machines	162
7.1.4	Containers	165
7.2	Model Deployment: Strategies	169
7.3	Model Deployment: Infrastructure	173
7.4	Model Deployment: Monitoring and Logging	174
7.5	What Can Possibly Go Wrong?	175
7.6	Rolling Back	177
8	Documenting Pipelines	181
8.1	Comments	182
8.2	Documenting Public Interfaces	185
8.3	Documenting Architecture and Design	195
8.4	Documenting Algorithms and Business Cases	200
8.5	Illustrating Practical Use Cases	204
9	Troubleshooting and Testing Pipelines	207
9.1	Data are the Problem	208
9.1.1	Large Data	209
9.1.2	Heterogeneous Data	210
9.1.3	Dynamic Data	211
9.2	Models are the Problem	213
9.2.1	Large Models	213
9.2.2	Black-Box Models	214
9.2.3	Costly Models	215
9.2.4	Many Models	216

vi Contents

9.3	Common Signs That Something Is Up	217
9.4	Tests are the Solution	220
9.4.1	What Do We Want to Achieve?	220
9.4.2	What Should We Test?	221
9.4.3	Offline versus Online Data	223
9.4.4	Testing Local versus Testing Global	228
9.4.5	Conceptual Errors versus Implementation Errors	231
9.4.6	Code Coverage and Test Prioritisation	232
III	Tools and Technologies	237
10	Tools for Developing Pipelines	239
10.1	Data Exploration and Experiment Tracking	239
10.2	Code Development	241
10.3	Build and Documentation Tools	246
11	Tools to Manage Pipelines in Production	249
11.1	Infrastructure Management	249
11.2	Machine Learning Software and Model Management	249
11.3	Dashboards and Reporting	250
12	Tools to Maintain Pipelines	251
IV	A Case Study	253
13	Recommending Recommendations: A Recommender System Using Natural Language Understanding	255
13.1	The Domain Problem	256
13.2	The Machine Learning Model	259
13.3	The Infrastructure	263
13.4	The Architecture of the Pipeline	265
	Bibliography	267
	Index	291

