200645 - PBDE - Statistical Programming and Databases

Coordinating unit: 200 - FME - School of Mathematics and Statistics
Teaching unit: 723 - CS - Department of Computer Science
715 - EIO - Department of Statistics and Operations Research
1004 - UB - (ENG)Universitat de Barcelona
707 - ESAII - Department of Automatic Control

Academic year: 2018
Degree: MASTER’S DEGREE IN STATISTICS AND OPERATIONS RESEARCH (Syllabus 2013). (Teaching unit Optional)
ECTS credits: 5

Teaching languages: English

Teaching staff
Coordinator: JOAQUIN GABARRÓ VALLÉS
Others: Segon quadristme:
            JOAQUIN GABARRÓ VALLÉS - A
            ALEXANDRE PERERA LLUNA - A

Prior skills
Non compulsory subject.
The student has already developed several abilities in Statistics and/or Operations Research previously.
A B2 (Cambridge First Certificate, TOEFL PBT >550) level of English is required.

Degree competences to which the subject contributes

Specific:
3. CE-1. Ability to design and manage the collection of information and coding, handling, storing and processing it.
4. CE-4. Ability to use different inference procedures to answer questions, identifying the properties of different estimation methods and their advantages and disadvantages, tailored to a specific situation and a specific context.
5. CE-5. Ability to formulate and solve real problems of decision-making in different application areas being able to choose the statistical method and the optimization algorithm more suitable in every occasion.
6. CE-6. Ability to use appropriate software to perform the necessary calculations in solving a problem.
7. CE-7. Ability to understand statistical and operations research papers of an advanced level. Know the research procedures for both the production of new knowledge and its transmission.
8. CE-8. Ability to discuss the validity, scope and relevance of these solutions and be able to present and defend their conclusions.

Transversal:
2. EFFECTIVE USE OF INFORMATION RESOURCES: Managing the acquisition, structuring, analysis and display of data and information in the chosen area of specialisation and critically assessing the results obtained.
10. FOREIGN LANGUAGE: Achieving a level of spoken and written proficiency in a foreign language, preferably English, that meets the needs of the profession and the labour market.
11. TEAMWORK: Being able to work in an interdisciplinary team, whether as a member or as a leader, with the aim of contributing to projects pragmatically and responsibly and making commitments in view of the resources that are available.
Teaching methodology

The course is divided into 2 modules that are taught in succession. Each module consists roughly of a half part of the sessions. All classes are theoretical-practical and in them teachers present and discuss the basic concepts of each module. The support material will be published previously in Athena (teaching guide, contents, course slides, examples, evaluation activities schedule, bibliography, ...).

The student should devote the autonomous learning hours to the study of the subjects of the course, bibliography extension and follow-up of the laboratory practices.

Learning objectives of the subject

This course presents and discusses tools and techniques to prepare students to data science. Main concepts introduced in class will cover tools and methods for data storage and analysis, including relational DB, noSQL and distributed databases, scientific computing, applied machine learning and deep learning with Python. Scala and Spark will also be considered. The course consists of two main modules.

MODULE 1:

First modulus will cover a crash course for scientific python for data analysis. This crash course will include include four main stages:
* Introduction to python language as a tool. ipython, ipython notebook (jupyter), basic types, mutability and immutability and object oriented programming.
* Short introduction to numerical python and matplotlib for graphical visualization.
* Introduction to scientific kits for data analysis with machine learning. Principal components analysis, clustering and supervised analysis with multivariate data.
* Introduction to Deep Learning with Python.

MODULE 2:

We introduce the Scala language and the Spark architecture.
* Scala as a functional language and the Scala collections.
* Spark and RDD (Resilient Distributed Data Sets).
* Spark and SQL.
* Introduction to MLlib.

Study load

<table>
<thead>
<tr>
<th><strong>Total learning time:</strong> 125h</th>
<th>Hours large group:</th>
<th>30h</th>
<th>24.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours medium group:</td>
<td>0h</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>Hours small group:</td>
<td>15h</td>
<td>12.00%</td>
<td></td>
</tr>
<tr>
<td>Guided activities:</td>
<td>0h</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>Self study:</td>
<td>80h</td>
<td>64.00%</td>
<td></td>
</tr>
</tbody>
</table>
# Content

| Introduction to Python | Learning time: 1h  
Theory classes: 1h |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>a. Why Python?</td>
<td></td>
</tr>
<tr>
<td>b. Python History</td>
<td></td>
</tr>
<tr>
<td>c. Installing Python</td>
<td></td>
</tr>
<tr>
<td>d. Python resources</td>
<td></td>
</tr>
</tbody>
</table>

| Working with Python    | Learning time: 1h  
Theory classes: 1h |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>a. Workflow</td>
<td></td>
</tr>
<tr>
<td>b. ipython vs. CLI</td>
<td></td>
</tr>
<tr>
<td>c. Text Editors</td>
<td></td>
</tr>
<tr>
<td>d. IDEs</td>
<td></td>
</tr>
<tr>
<td>e. Notebook</td>
<td></td>
</tr>
</tbody>
</table>

| Getting started with Python | Learning time: 1h  
Theory classes: 1h |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td></td>
</tr>
<tr>
<td>a. Introduction</td>
<td></td>
</tr>
<tr>
<td>b. Getting Help</td>
<td></td>
</tr>
<tr>
<td>c. Basic types</td>
<td></td>
</tr>
<tr>
<td>d. Mutable and in-mutable</td>
<td></td>
</tr>
<tr>
<td>e. Assignment operator</td>
<td></td>
</tr>
<tr>
<td>f. Controlling execution flow</td>
<td></td>
</tr>
<tr>
<td>g. Exception handling</td>
<td></td>
</tr>
</tbody>
</table>
| Functions and Object Oriented Programming | **Learning time:** 1h  
Theory classes: 1h |
|-----------------------------------------|---------------------|
| **Description:**  
| a. Defining Functions  
b. Input and Output  
c. Standard Library  
d. Object-oriented programming |

| Introduction to NumPy | **Learning time:** 2h  
Theory classes: 2h |
|-----------------------|---------------------|
| **Description:**  
| a. Overview  
b. Arrays  
c. Operations on arrays  
d. Advanced arrays (ndarrays)  
e. Notes on Performance (\%timeit in ipython) |

| Matplotlib | **Learning time:** 2h  
Theory classes: 2h |
|------------|---------------------|
| **Description:**  
| a. Introduction  
b. Figures and Subplots  
c. Axes and Further Control of Figures  
d. Other Plot Types  
e. Animations |

| Python scikits | **Learning time:** 1h  
Theory classes: 1h |
|----------------|---------------------|
| **Description:**  
| a. Introduction  
b. scikit-timeseries |
## scikit-learn

**Description:**
- a. Datasets
- b. Sample generators
- c. Unsupervised Learning
- d. Supervised Learning
  - i. Linear and Quadratic Discriminant Analysis
  - ii. Nearest Neighbors
  - iii. Support Vector Machines
- e. Feature Selection

**Learning time:** 8h
- Theory classes: 8h

## Practical Introduction to Scikit-learn

**Description:**
- a. Solving an eigenfaces problem
  - i. Goals
  - ii. Data description
  - iii. Initial Classes
  - iv. Importing data
- b. Unsupervised analysis
  - i. Descriptive Statistics
  - ii. Principal Component Analysis
  - iii. Clustering
- c. Supervised Analysis
  - i. k-Nearest Neighbors
  - ii. Support Vector Classification
  - iii. Cross validation

**Learning time:** 5h 30m
- Theory classes: 5h 30m

## Introduction to Zeppelin, Scala & Functional Programming

**Description:**
- a. Immutable & Mutable
- b. Lists and maps, filters, reductions
- c. Map reduce
- d. Other collections, Streams

**Learning time:** 5h
- Theory classes: 5h
## 200645 - PBDE - Statistical Programming and Databases

### Spark architecture & Spark Core

**Learning time:** 5h  
Theory classes: 5h

**Description:**
- a. Spark architecture: in particular Spark Core  
- b. Spark context  
- c. Types of operations: transformations and actions  
- d. RDD: Resilient Distributed Data Sets  
- e. Closure of a function

### Spark SQL

**Learning time:** 7h 30m  
Theory classes: 7h 30m

**Description:**
- a. Reading form a file.  
- b. Spark Data Frame.  
- c. Selection, filters, grouping, sorting.  
- d. Window operations  
- c. SQL

### Spark: MLlib

**Learning time:** 5h  
Theory classes: 5h

**Description:**
- a. Description of the MLlib.  
- b. Labeled Points and features  
- c. Linear Regression Example

### Qualification system

Final grade will be composed by:  
- 1/4 Written exam first module  
- 1/4 Written exam first module  
- 1/2 Final practical assignment on large databases integrating concepts from both modules
Bibliography

Basic:


Complementary: