Government of Russian Federation

Federal State Autonomous Educational Institution of High Professional Education

«National Research University Higher School of Economics»

National Research University
High School of Economics
Faculty of Computer Science

Syllabus for the course
«Methods for Machine Learning and Data Mining»

(Методы машинного обучения и разработки данных)

010402.68 «Applied Mathematics and Informatics»,
«**Data Sciences**» Master program

Authors:
Dmitry I. Ignatov, Candidate of sciences (Ph.D), associate professor, dignatov@hse.ru
Maxim A. Borisyak, Master of sciences (Msc), lecturer, mborisyak@hse.ru

Approved by: Head of Data Analysis and Artificial Intelligence Department, Sergey O. Kuznetsov

Recommended by:

Moscow, 2014

1. Teachers

Author, lecturer: Dmitry Ignatov, National Research University Higher School of Economics, School of Data Analysis and Artificial Intelligence, associate professor;

Author, lecturer: Borisyak Maxim, National Research University Higher School of Economics, School of Data Analysis and Artificial Intelligence, lecturer.

2. Scope of Use

The present program establishes minimum demands of students' knowledge and skills, and determines content of the course.

The present syllabus is aimed at department teaching the course, their teaching assistants, and students of the Master of Science program 010402.68 «Data Sciences»,

This syllabus meets the standards required by:

- Educational standards of National Research University Higher School of Economics;
- Educational program «Data Sciences» of Federal Master's Degree Program 010402.68 «Applied Mathematics and Informatics», 2014;
- University curriculum of the Master's program in «Data Sciences» for 2014.

3. Summary

Machine Learning and mining of massive datasets are rapidly growing fields of data analysis. For many years data analysis and statistical community has been developing algorithms and methods for discovering patterns in datasets. Besides theoretical knowledge successful research in the areas depends on confided usage of common methods, algorithms and tools along with skills for developing new ones. The focus of the course "Methods for Machine Learning and Data Mining" is to introduce students to methods and modern programming tools and frameworks aimed for data analysis. Special attention is given to methods for handling massive datasets. The course is constantly being adopted to match current state-of-the-art in the area.

4. Learning Objectives

The objectives of the course "**Methods for Machine Learning and Data Mining**" is to introduce students to state-of-the-art methods and modern programming tools for data analysis.

5. Learning outcomes

After completing the study of the discipline "**Methods for Machine Learning and Data Mining**", the student are expected to:

- understand complexity of Machine Learning algorithms and their limitations;
- understand modern notions in data analysis oriented computing;
- be capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
- be capable of performing distributed computations;
- be capable of performing experiments in Machine Learning using real-world data.

After completing the study of the discipline "**Methods for Machine Learning and Data Mining**" the student should have the following competences:

Competence	**Code**	**Code (UC)**	**Descriptors (indicators of achievement of the result)**	**Educative forms and methods aimed at generation and development of the competence**
The ability to reflect developed methods of activity.	SC-1	SC-M1	The student is able to reflect developed and implement methods for machine learning and data mining (data sciences)	Lectures and tutorials, group discussions, presentations, paper reviews.
The ability to propose a model to invent and test methods and tools of professional activity	SC-2	SC-M2	The student is able to improve and develop methods and algorithms as applicable to machine learning and data mining (data sciences)	Classes, home works.
Capability of development of new research methods, change of scientific and industrial profile of self-activities	SC-3	SC-M3	The student obtains necessary knowledge in methods for machine learning and data mining, which is sufficient to develop new methods	Home tasks, paper reviews
The ability to describe problems and situations of professional activity in terms of humanitarian, economic and social sciences to solve problems which occur across sciences, in allied professional fields.	PC-5	IC-M5.3_5.4_5.6_2.4.1	The student is able to describe computational data analysis problems in terms of computational mathematics.	Lectures and tutorials, group discussions, presentations, paper reviews.
The ability to detect, transmit common goals in the professional and social activities	PC-8	SPC-M3	The student is able to identify algorithmic aspects in machine learning and data mining tasks, evaluate correctness and efficiency of the used methods, and their applicability in each current situation	Discussion of paper reviews; cross discipline lectures

6. Place of the discipline in the Master's program structure

The course "Methods for Machine Learning and Data Mining" is a course taught in the second year of the Master's program 010402.68 "Data Sciences" and follows the course "Introduction to Machine Learning and Data Mining", the base course for specialization "Intelligent Systems and Structural Analysis".

Prerequisites

The course is based on knowledge and understanding of

- Algorithms and data structures
- Theory of probability and statistical analysis
- Machine Learning

Thus it is highly recommended for students to pass preceding course "Introduction to Machine Learning and Data Mining" or analogous one.

The course also requires some programming experience in all of the languages:

- Python
- C or C++

Knowledge of Java or Scala programming languages is also a benefit.

7. Schedule

One pair consists of 1 academic hour for lecture and 1 academic hour for classes after lecture.

№	Topic	Total hours	Contact hours		Self-study
			Lectures	Seminars	
1	Introduction to methods for Machine Learning, IPython notebook, data visualisation.	10	2	2	6
2	Numpy and scipy basics: common linear algebra and statistical routines, numerical optimization	11	2	2	7
3	Introduction to scikit-learn. Common classification, regression and clustering methods.	11	2	2	7
4	Meta-learning in scikit-learn: ensembling, hyper-parameter optimization, feature extraction.	11	2	2	7
5	Symbolic computations. Introduction to theano/TensorFlow.	11	2	2	7
6	Symbolic computations for Deep Learning and stochastic optimisation. GPU computing.	11	2	2	7
7	Symbolic computations for Unsupervised Learning.				
8	Introduction to dataflow computational model, distributed programming. Apache Spark basics.	11	2	2	7
9	Distributed computations for Machine Learning. Apache Spark MLlib.	10	2	2	6
10	Recommender systems: Matrix Factorization, ALS.	11	2	2	7
	Total	108	20	20	68

8. Requirements and Grading

	Type of work	Characteristics		
		1	2	
Type of grading	Homework	10		Solving homework tasks and examples.
	Special homework – research projects and reports		2	Research project on real world Machine Learning problem, presentation of the results, tools and techniques, used in the project.
	Exam		1	Written exam
Final				

9. Assessment

The assessment consists of classwork and homework, assigned after each lecture. Students have to demonstrate confident usage of presented methods, tools, frameworks and techniques, be able to solve example real world tasks.

Final assessment is the final exam. Students have to combine their theoretical knowledge with practical skills in order to solve real world problems.

The grade formula:

The exam consists of 1 problem, giving 10 points total.

Final course mark *is obtained from the following formula*:

Ofinal = 0,4 * Ocumulative + 0,4 * OcumaltiveSpecial + 0,2 * Oexam.

where:
Ocumulative – cumulative mark for classwork and homework;
OcumulativeSpecial – cumulative mark for special homework;
Oexam – mark on the exam.

The grades are rounded in favour of examiner/lecturer with respect to regularity of class and home works. All grades, having a fractional part greater than 0.5, are rounded up.

Table of Grade Accordance

Ten-point Grading Scale	**Five-point Grading Scale**	
1 - very bad 2 – bad 3 – no pass	Unsatisfactory - 2	**FAIL**
4 – pass 5 – highly pass	Satisfactory – 3	**PASS**

6 – good 7 – very good	Good – 4	
8 – almost excellent 9 – excellent 10 – perfect	Excellent – 5	

10. Course Description

The following list describes main topics covered by the course with lecture order.

Topic 1. Introduction to methods for Machine Learning, IPython notebook, data visualisation

Content: Introduction to methods for Machine Learning. Overview of modern technologies, problem examples and basic tasks. Introduction to IPython notebook and basic data visualisation: line and bar plots, histograms, image visualisation, heat maps.

Topic 2. Numpy and scipy basics: common linear algebra and statistical routines, numerical optimization.

Content: Introduction to numpy library. Matrices and linear algebra routines: basic matrix operations, decompositions, algorithms, their computational complexity and implementations. Introduction to scipy library. Statistical routines: basic statistics, sampling, maximal likelihood fitting, hypothesis testing. Numerical optimization: scalar optimization, local optimization, global optimization. Classwork and homework: classification of handwritten digits by fitting custom models and hypothesis tests.

Topic 3. Introduction to scikit-learn. Common classification, regression and clustering methods.

Content: Introduction to scikit-learn library by the example of Logistic Regression, Support Vector Machine, Random Forest, K-means, DBSCAN. Classwork and homework: classification and clustering of handwritten digits, feature engineering and feature selection.

Topic 4. Meta-learning in scikit-learn: ensembling, hyper-parameter optimization, feature extraction.

Content: Meta-learning in scikit-learn: GridSearch, optimization over hyper-parameters, stacking, Gradient Boosting, feature extraction. Classwork and homework: feature learning on handwritten digits.

Topic 5. Symbolic computations. Introduction to theano/TensorFlow.

Content: Introduction to symbolic computations. Automatic differentiation. Introduction to theano/TensorFlow. Classwork and homework: classification of handwritten digits with perceptron with automatic differentiation, custom Neural Networks layers.

Topic 6. Symbolic computations for Deep Learning and stochastic optimisation. GPU computing.

Content: Introduction to theanets, keras, lasagne, downhill. Convolution and recurrent Neural Networks. Introduction to stochastic optimization: Stochastic Gradient Descent, Nesterov's momentum, AdaDelta, ADAM. Classwork and homework: hand written digets recognition using Convolution Neural Networks, comparison of optimisation algorithms for Deep Neural Networks.

Topic 7. Symbolic computations for Unsupervised Learning.

Content: Autoencoders, embedding, handling sparse data. Word2vec.

Topic 8. Introduction to dataflow computational model, distributed programming. Apache Spark basics.

Content: Introduction to dataflow computational model. Distributed programming. Apache Spark basics: RDD, RDD transformations. Classwork and homework: distributed Logistic Regression on hand written digits.

Topic 9. Distributed computations for Machine Learning. Apache Spark Mllib.

Content: common distributed classification, regression and clustering algorithms. Classwork and homework: distributed classification and clustering of hand written digits.

Topic 10. Recommender systems: Matrix Factorization, ALS.

Content: Recommender systems on Apache Spark. Collaborative filtering via Matrix Factorization, Alternating Least Squares. Classwork and homework: distributed collaborative filtering on movie rating dataset.

11. Term Educational Technology

The following educational technologies are used in the study process:

- discussion and analysis of the results of the home task in the group;
- individual education methods, which depend on the progress of each student;
- group projects on analysis of real data.

12. Recommendations for course lecturer

There are a great number of methods for Data Analysis and Data Mining, which is impossible to fully cover in one course. Thus only three main domains are selected: traditional methods for Machine Learning, multi-core and GPU computing for Deep Learning and distributed computing for Big Data. From each domain only most representative methods are selected: scikit-learn for the first domain, theano/TensorFlow based for the second one and Apache Spark for the third one. Hence it is not only important to introduce students to these methods, but also to introduce basic notions and develop self-learning abilities.

Course lecturer is advised to use interactive learning methods, which allow participation of the majority of students, such as slide presentations, interactive demonstrations, code examples. Since the course has rather practical than theoretical nature, it is advised to spent about half of the time for solving examples individually or in small groups.

Also individual research projects play significant role, it is recommended to reserve time for students' presentations.

13. Recommendations for students

Lectures are combined with classes. Students are welcome to ask questions and actively participate in-group discussions and projects. Students are also encouraged to prepare presentations of topic related to the course, but not included into the syllabus. All tutors are ready to answer questions during lectures, special office hours or online by official emails (listed in the “contacts” section). Note that the final mark is a cumulative value of your term activity and final results.

14. Sample final exam questions

1. Compare different methods (e.g. Random Forest, SVM, Neural Networks) of classification for given dataset. Perform parallel optimal parameter search and parallel cross validation.
2. Implement a Neural Network for recognition of facial expressions.

3. Implement gated Neural Network as an ensembling method.
4. Implement a distributed version of given algorithm (e.g. Naive Bayes, Logistic Regression).
5. Learn latent factors for collaborative filtering via distributed Alternating Least Squares algorithms.

15. Reading and Materials

15.1. Required Reading

1. Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. Mining of massive datasets. Cambridge University Press, 2014.
2. Peter Flach Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012

15.2. Recommended Reading

1. Christopher M Bishop. Pattern recognition and machine learning. Springer, 2006
2. Trevor J.. Hastie, Robert John Tibshirani, and Jerome H Friedman. The elements of statistical learning: data mining, inference, and prediction. Springer, 2009
3. Witten, E. Frank, M. Hall. Data Mining: Practical Machine Learning Tools and Techniques, 2011, Morgan Kaufmann Publishers

15.3. List of review papers

1. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in
Python. Journal of Machine Learning Research, 12:2825–2830, 2011
2. Frederic Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features
and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012
3. Mart´ın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia,
Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man´e, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vi´egas, Oriol Vinyals, Pete Warden, Martin

Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org

4. Travis E Oliphant. Python for scientific computing. Computing in Science & Engineering, 9(3):10–20, 2007
5. K Jarrod Millman and Michael Aivazis. Python for scientists and engineers. Computing in Science & Engineering, 13(2):9–12, 2011
6. Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering, 13(2):22–30, 2011
7. Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. HotCloud, 10:10–10, 2010
8. Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. Technical report, Technical Report UCB/EECS-2011-82, EECS Department, University of California, Berkeley, 2011
9. Rupesh K Srivastava, Klaus Greff, and Jurgen Schmidhuber. Training very deep networks. In Advances in Neural Information Processing Systems, pages 2368–2376, 2015
10. Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. Advances in neural information processing systems, 19:153, 2007
11. Jurgen Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61:85–117, 2015
12. Yoshua Bengio. Learning deep architectures for ai. Foundations and trends in Machine Learning, 2(1):1–127, 2009

15.4. Course telemaintenance

All material of the discipline are posted in informational educational site at NRU HSE portal www.ami.hse.ru. Students are provided with links to research papers, electronic books, data and software.

16. Equipment

The course requires a laptop, projector, and acoustic systems.
It also requires opportunity to install programming software, such as:

- Jupyter notebook server and data analysis libraries
- Apache Spark cluster.

Lecture materials, course structure and syllabus are prepared by Maxim Borisyak.