Санкт-Петербургский филиал федерального государственного автономного образовательного учреждения высшего образования "Национальный исследовательский университет "Высшая школа экономики"

Факультет Санкт-Петербургская школа экономики и менеджмента
Национального исследовательского университета
«Высшая школа экономики»

Департамент прикладной математики и бизнес-информатики

Рабочая программа дисциплины

Практическое программирование и анализ данных в специализированных средах

для образовательной программы «Анализ больших данных в бизнесе, экономике и обществе»
направления подготовки 01.04.02«Прикладная математика и информатика»
уровень магистратура

Разработчики программы:
Сироткин А.В., к.ф.-м.н., доцент, avsirotkin@hse.ru;
Алексеев А.М., преподаватель, anton.m.alexeyev@gmail.com

Согласована менеджером ОП «Анализ больших данных в бизнесе, экономике и обществе»
«30» августа 2016г.

Е.С. Авдонина _____

Утверждена Академическим руководителем образовательной программы

А.В. Сироткин _____

«30» августа 2016г.

Санкт-Петербург, 2016

1 Область применения и нормативные ссылки

Настоящая рабочая программа дисциплины устанавливает минимальные требования к знаниям и умениям студента, а также определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих дисциплину «Практическое программирование и анализ данных в специализированных средах», учебных ассистентов и студентов направления подготовки 01.04.02 «Прикладная математика и информатика», обучающихся по образовательной программе «Анализ больших данных в бизнесе, экономике и обществе».

Рабочая программа дисциплины разработана в соответствии с:

- Образовательным стандартом НИУ ВШЭ http://www.hse.ru/data/2016/11/02/1111123560/01.04.02%20Прикладная%20математика%20и%20информатика.pdf;
- Образовательной программой «Анализ больших данных в бизнесе, экономике и обществе», направление подготовки 01.04.02 «Прикладная математика и информатика», ;
- Объединенным учебным планом университета по образовательной программ «Анализ больших данных в бизнесе, экономике и обществе».

2 Цели освоения дисциплины

Целями освоения дисциплины "Практическое программирование и анализ данных в специализированных средах" являются:

- приобретение студентами практического навыка построения и анализа алгоритмов, а также навыка самостоятельной их реализации посредством программирования на современных языках программирования (на примере Python) и в специализированных пакетах (R).
- формирование понимания технологии работы со сложными структурами программ и данных.
- формирование представления о парадигмах программирования в специализированных средах.
- развитие навыков практического программирования и анализа данных как на примере стандартных задач программирования (работа с деревьями, графами, обработка списков и массивов, символьное преобразование), так и в приложении к решению практических задач.
- формирование основы для дальнейшего применения в области математического и компьютерного моделирования сложных социально-экономических систем и процессов.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
Системные компетенции					
Способен анализировать, верифициро-	СК-6	РБ, СД	Умеет уточнять постановку задачи в диалоге с	Практические занятия, самосто-	Домашняя работа

Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
вать, оценивать полноту информации в ходе профессиональной деятельности, при необходимости восполнять и синтезировать недостающую информацию.			заказчиком.	ятельная работа студентов	
Профессиональные компетенции					
А) Социально-личностные компетенции					
Способен порождать принципиально новые идеи и продукты, обладает креативностью, инициативностью.	ПК-8	СД	Умеет создавать программы решающие поставленные задачи и обладающие заданной функциональностью.	Практические занятия	Домашняя работа, экзамен
Б) Инструментальные компетенции					
Способен осуществлять целенаправленный многокритериальный поиск информации о новейших научных и технологических достижениях в сети Интернет и других источниках.	ПК-13	РБ, СД	Умеет производить отбор программных библиотек, предлагающих решения отдельных частей задачи, с целью наилучшего решения с точки зрения как эффективности алгоритмов, так и с точки зрения времени создания программы.	Практические занятия	Домашняя работа, контрольная работа, экзамен
Способен описывать проблемы и ситуации профессиональной деятельности, используя язык и аппарат прикладной математики при решении междисциплинарных проблем.	ПК-14	СД	Может подготовить отчет о проведенных вычислительных экспериментах и о создании, необходимого для решения поставленной задачи, программного продукта.	Практические занятия	Домашняя работа
Способен применять в исследовательской и прикладной деятельности современные языки программирования и языки манипулирования данными, операционные системы, электронные	ПК-20	РБ, СД	Может создать программу на языке Python для решения поставленной задачи в области анализа данных.	Практические занятия, домашняя работа	Домашняя работа, экзамен

Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
библиотеки и пакеты программ, сетевые технологии и т.п.					

4 Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к базовой части цикла дисциплин магистерской программы «Анализ больших данных в бизнесе, экономике и обществе».

Изучение данной дисциплины базируется на следующих дисциплинах:

Изучение данной дисциплины базируется на следующих дисциплинах обучения в бакалавриате: «Линейная алгебра», «Математический анализ», «Теория вероятностей и математическая статистика».

Для освоения учебной дисциплины, студенты должны владеть следующими знаниями и компетенциями:

- способен анализировать, верифицировать, оценивать полноту информации в ходе профессиональной деятельности, при необходимости восполнять и синтезировать недостающую информацию;
- способен порождать принципиально новые идеи и продукты, обладает креативностью, инициативностью;
- способен осуществлять целенаправленный многокритериальный поиск информации о новейших научных и технологических достижениях в сети Интернет и других источниках;
- способен описывать проблемы и ситуации профессиональной деятельности, используя язык и аппарат прикладной математики при решении междисциплинарных проблем;
- способен применять в исследовательской и прикладной деятельности современные языки программирования и языки манипулирования данными, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии и т.п.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин: «Вычислительная статистика», «Распределенная обработка и анализ больших данных», «Теория экономических механизмов», «Анализ социальных и экономических сетей».

5 Тематический план учебной дисциплины

ОБЪЕМ ДИСЦИПЛИНЫ - 6 зачетных единиц.

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1	Основы программирования на языке Python	100			28	72
2	Основы технологического обеспечения анализа данных	28			8	20
3	Основы сбора, предварительной обработки и анализа данных в Python	100			28	72

ИТОГО	**228**	**0**	**0**	**64**	**164**

6 Формы контроля знаний студентов

Тип контроля	Форма контроля	1 модуль		2 модуль		Параметры **
		Сентябрь	Октябрь	Ноябрь	Декабрь	
Текущий	Контрольная работа №1		*			Письменная работа 60 минут
	Контрольная работа №2				*	Письменная работа 60 минут
	Домашняя работа			*		Домашняя работа сроком выполнения 2 недели.
Итоговый	Экзамен				*	Экзамен, интегрированный в письменной форме - 150 минут

7 Критерии оценки знаний, навыков

Критерии оценки **контрольных работ**:

Правильное решение предложенных задач по теме практического занятия, которое включает: использование предлагаемых средств и языков программирования.

На контрольной студенту предлагается от 6 до 12 базовых задач, и от 0 до 3 задач повышенной сложности.

Каждая базовая задача оценивается в один балл, который выставляется при корректном решении.

Задачи повышенной сложности в своем описании содержат указание на максимально возможное число баллов.

При оценки решения каждой задачи повышенной сложности учитываются следующие факторы:

- точность и правильность ответа в задачах требующих понимания результатов работы приведенных фрагментов кода;
- корректность и оптимальность решения задач, требующих написание программы (фрагмента программы) решающей поставленную задачу.

За решение каждой задачи выставляются накопительные баллы в соответствии с точностью и полнотой решения и описания процесса решения задачи.

Итоговые баллы за контрольную работу вычисляются как 10 умноженное на долю набранных баллов по отношению к максимально возможному числу баллов, и округленное по правилам арифметики.

Итоговое максимальное число баллов за одну контрольную работу – **10 баллов.**

Критерии оценки **за домашнюю работу**:

Задание представляет собой проведение анализа, построение модели, а также оценку качества построенной модели для заданного набора данных. Результаты работы предоставляются в виде кода на языке Python, а так же письменного отчета описывающего этапы построения модели и выбор оптимальных параметров.

При выставлении оценки преподавателем учитываются корректность выбранных алгоритмов и полнота составленного отчета.

Отчеты, не сданные в установленную дату, не принимаются. В таком случае выставляется оценка 0 баллов.

Максимальная оценка за домашнюю работу – **10 баллов.**

Критерии оценки **за экзамен:**

На экзамене содержится ряд задач, охватывающий все темы курса. За каждое правильно выполненное задание присваивается накопительные баллы, которые суммируются и переводятся в 10-балльную систему.

Экзамен **($О_{экз}$)** проводится в смешанной форме и содержит качественные и практические задачи. За каждое правильно выполненное задание присваивается накопительные баллы, которые суммируются.

Максимальная оценка за экзамен – **10 баллов.**

8 Содержание дисциплины

Раздел 1. Основы программирования на языке Python

История создания языка Python. Понятие о Python как динамически типизированном интерпретируемом языке программирования высокого уровня. Особенности синтаксиса Python. Предложение о стиле форматирования программного кода PEP8. REPL и исполнение записанного в файл программного кода. Краткий обзор средств разработки на Python. Понятие о синтаксисе Python. Встроенные типы данных. Условный оператор. Циклы. Операторы безусловного перехода. Встроенные коллекции в Python. Работа с файловой системой в Python. Функции. Элементы функционального программирования в Python. Понятие об объектно-ориентированном программировании в Python.

Раздел 2. Основы технологического обеспечения анализа данных

Командная строка Linux. GNU coreutils. Распространённое ПО для работы на удалённых серверах. Средства загрузки веб-страниц и файлов (по протоколу HTTP). Системы управления версиями (git).

Раздел 3. Основы сбора, предварительной обработки и анализа данных в Python

Библиотека numpy. Работа с разреженными матрицами с помощью scipy. Основы работы с табличными данными средствами pandas. Введение в работу с реляционными базами данных. Работа со структурированными/полуструктурированными данными в распространённых форматах. Регулярные выражения в Python. Библиотека scikit-learn, обзор решаемых задач. Построение цепочки обработки данных для решения задач предсказания. Библиотека nltk. Обзор решаемых библиотекой задач.

9 Образовательные технологии

1. Работа в малых группах.
2. Программирование моделей на компьютере.

9.1 Методические рекомендации преподавателю

9.2 Методические указания студентам по освоению дисциплины

Разработчиком курса подготовлены учебные материалы по темам основных занятий, которые выдаются для помощи самостоятельной слушателей курса.

Самостоятельная работа студента включает:

- поиск и анализ информации по тематике курса;
- знакомство со специальной научной литературой по тематике курса;
- решение задач, включая программирование и визуализацию на компьютере, и анализ ситуаций, выданных преподавателем;
- подготовку к семинарским занятиям.

9.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

- Сайт с задачами по программированию и автоматической проверкой задач: http://codeforces.com/
- Python for Scientists: A Curated Collection of Chapters from the O'Reilly Data and Programming Libraries. O'Reilly. 2015 http://www.oreilly.com/programming/free/files/python-for-scientists.pdf

10 Оценочные средства для текущего контроля и аттестации студента

10.1 Оценочные средства для оценки качества освоения дисциплины в ходе текущего контроля

Контрольная работа №1 по основам Python, качественная задача

Что выведет программа
(вывод сообщения об ошибке также является возможным ответом;
в этом случае надо написать в ответ "ошибка" и объяснить,
почему она произошла)?

```
c = [1, 2, 3]
a, b =  12, c[1]
a / b
print(a)
```

Ответ: 12

Контрольная работа №1 по основам Python, практическая задача (максимальная оценка 2 балла)

- дано целое число в переменной n (можно считать, что n >= 0)
- написать программу, которая выводит подряд все числа Фибоначчи, строго меньшие n! (факториал n)

0 баллов: не работающее решение (пожалуйста, будьте внимательны и не делайте ошибок с индексами списков и границами циклов!)
1 балл: решение с использованием двух циклов
2 балла: решение с использованием одного цикла

Мелкие синтаксические ошибки числом менее трёх, не штрафуются.
Рекомендация: проверьте вашу программу на небольших значениях n, «поставив себя на место интерпретатора».

Контрольная работа №2 по технологическому обеспечению анализа данных, качественная задача (максимальная оценка 2 балла)

Напишите «однострочную» команду, выводящую в консоль с 200 по 420 строки файла /home/user/file.txt, не используя Python или циклы

Ответ (возможны другие варианты): head -420 /home/user/file.txt | tail -321

Использовано запрещённое в условии или не решено: 0 баллов

Ошибка на одну строку: 1 балл

Верно: 2 балла

Контрольная работа №2 по основам работы с данными, качественная и практическая задачи

Качественная задача: что распечатает данная программа?

```
from lxml import etree

class SomeTarget(object):
    def __init__(self):
        self.text = []
    def start(self, tag, attrib):
        print("Into", tag)
    def end(self, tag):
        print("Outta", tag)
    def data(self, data):
        if data:
            self.text.append(data)
    def close(self):
        return self.text

some_xml_data = "<layer0><layer1><layer2>Hello
</layer2></layer1><layer1>Howdy</layer1></layer0>"
parser = etree.XMLParser(target = SomeTarget())
results = etree.fromstring(some_xml_data, parser)
print(results)
```

Практическая задача: по аналогии построить парсер, который будет выводить текст только в внутри тегов layer2, и отправить на почту преподавателю до конца пары.

Решение:

```
from lxml import etree

class SomeTarget(object):

    def __init__(self):
        self.text = []
        self.in_tag_layer2 = False

    def start(self, tag, attrib):
        print("Into", tag)
        if tag == "layer2":
            self.in_tag_layer2 = True

    def end(self, tag):
```

```
            print("Outta", tag)
            if tag == "layer2":
                self.in_tag_layer2 = False

        def data(self, data):
            if data and self.in_tag_layer2:
                self.text.append(data)

        def close(self):
            return self.text

    some_xml_data = " <layer0><layer1><layer2>Hello
</layer2></layer1><layer1>Howdy</layer1></layer0>"
    parser = etree.XMLParser(target = SomeTarget())
    results = etree.fromstring(some_xml_data, parser)
    print(results)
```

Пример домашней работы

Для набора данных "Titanic: Machine Learning from Disaster", пользуясь консультацией преподавателя

- произвести первичный анализ данных,
- разобраться с подходящими методами визуализации,
- достроить признаки,
- с помощью подходящей модели из scikit-learn (по рекомендации преподавателя) определить наиболее значимые из них,
- оценивая качество модели с помощью K-fold-валидации, подобрать параметры, при которых она показывает лучшие результаты.

Составить отчёт, в котором описать

- наблюдения на этапе первичного анализа,
- постановку и ход экспериментов,
- интерпретацию результатов.

10.2 Примеры заданий промежуточного (при наличии в ОУПе) /итогового контроля

Задания итогового контроля, повторяют по своей структуре задания используемые в контрольных на протяжении всего курса.

11 Порядок формирования оценок по дисциплине

Накопленная оценка по дисциплине рассчитывается с помощью взвешенной суммы оценок за отдельные формы текущего контроля знаний следующим образом:

$$\mathbf{О_{накопленная} = 0{,}34О_{др} + 0{,}33О_{кр\_1} + 0{,}33О_{кр\_2}}, \text{ где}$$

$\mathbf{О_{дз}}$ - оценка знаний студента за **домашнюю работу**;
$\mathbf{О_{кр\_1\_}}$ - оценка знаний студента за **контрольную работу номер 1**;
$\mathbf{О_{кр\_2}}$ – оценка знаний студента за **контрольную работу номер 2**;

Результирующая оценка по дисциплине (которая идет в диплом) рассчитывается следующим образом:

$$\mathbf{O_{результ}=0,6O_{накопленная}+0,4O_{экз}}, \text{ где}$$

$\mathbf{O_{накопленная}}$ - накопленная оценка по дисциплине;

$\mathbf{O_{экз}}$ - оценка за экзамен.

В формулу для $\mathbf{O_{результ}}$ подставляются значения $\mathbf{O_{накопленная}}$ и $\mathbf{O_{экз}}$, округленные до ближайшего целого значения. $\mathbf{O_{результ}}$ округляется до ближайшего целого значения.

По усмотрению ведущего преподавателя, если это не противоречит действующим документам на момент экзамена, при получении накопленной оценки 8 баллов и более, студент может быть освобожден от экзамена. В таком случае, с согласия студента, ему выставляется результирующая оценка, равная накопленной.

Студент не получает возможность пересдать низкие результаты за домашнюю работу и/или работу на семинарских или контрольную работу, а также при пропуске соответствующих им учебных часов.

При получении неудовлетворительной оценки $\mathbf{O_{результ}}$ (значение после округления менее 4 баллов) выставляется оценка «НЕУДОВЛЕТВОРИТЕЛЬНО».

12 Учебно-методическое и информационное обеспечение дисциплины

12.1 Основная литература

Hetland M. L. Python Algorithms: mastering basic algorithms in the Python Language [Electronic Resource] / Magnus Lie Hetland.- NY: Apress, 2010.- 336 p. - Authorized access: http://link.springer.com/book/10.1007/978-1-4302-3238-4 (Online Digital Library “Springer eBooks”).

12.2 Дополнительная литература

1. Garreta R., Moncecchi G. Learning scikit-learn : Machine Learning in Python [Electronic Resource] / Garreta Raúl, Moncecchi Guillermo. - Packt Publishing Ltd, 2013.- 118 p. - Authorized access: http://site.ebrary.com/lib/hselibrary/detail.action?docID=10813437 (Online Digital Library “Ebrary”).
2. Richert W. Building machine learning systems with python [Electronic Resource] / Willi Richert, Luis Pedro Coelho, Jonathan Chaffer.- Packt Publishing Ltd, 2013.- 290 p. - Authorized access: http://site.ebrary.com/lib/hselibrary/detail.action?docID=10742638 (Online Digital Library “Ebrary”).

12.3 Справочники, словари, энциклопедии

Полное собрание стандартов и документации по языку Python www.python.org

12.4 Ресурсы информационно-телекоммуникационной сети «Интернет»

http://repl.it

12.5 Программные средства

Anaconda 3

12.6 Информационные справочные системы

Полное собрание стандартов и документации по языку Python www.python.org

12.7 Дистанционная поддержка дисциплины

При необходимости возможна дистанционная поддержка.

13 Материально-техническое обеспечение дисциплины

Для проведения практических занятий требуется компьютерный класс с программным пакетом Anaconda, и доступ к серверу с операционной системой семейства Linux.