

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Проректор по учебной работе и
довузовской подготовке**

А.А. Воронов

	Рабочая программа дисциплины (модуля)
по дисциплине:	Применение Python в статистическом анализе данных
по направлению:	Прикладная математика и информатика
профиль подготовки:	Прикладная математика и информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	1
квалификация:	магистр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Аудиторных часов: 45 всего, в том числе:

лекции: 0 час.

семинары: 45 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 45 час.

Всего часов: 90, всего зач. ед.: 2

Количество контрольных работ, заданий: 1

Программу составили:

М.Е. Бекетов, ассистент

О.Н. Ивченко, старший преподаватель

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 11.06.2020

Аннотация

Курс по основным библиотекам языка Python и методам, используемым при работе с данными (в анализе данных):

- работе с числовыми массивами в NumPy,
- работе с табличными данными в pandas,
- построению графиков и диаграмм (визуализации данных) в matplotlib,
- базовым статистическим моделям машинного обучения в scikit-learn

1. Цели и задачи

Цель дисциплины

Познакомить студентов с языком программирования Python и подготовить их к практической деятельности в должностях аналитиков и программистов программного обеспечения.

Задачи дисциплины

- * Сформировать знания о правильном применении языка Python в разработке.
- * Сформировать знания о популярных библиотеках и фреймворках на Python.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен решать актуальные задачи фундаментальной и прикладной математики	ОПК-1.3 Понимает междисциплинарные связи в области прикладной математики и информатики и способен их применять при решении задач профессиональной деятельности
ПК-2 Понимает и способен применить в научно-исследовательской и прикладной деятельности современный математический аппарат и алгоритмы, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии	ПК-2.1 Знает основы научно-исследовательской деятельности в области информационно-коммуникационных технологий владеет знанием основ философии и методологии науки; знанием методов научных исследований и навыками их проведения
	ПК-2.2 Имеет практический опыт использования существующих методов и алгоритмов решения задач цифровой обработки сигналов, использования сети Интернет, аннотирования, реферирования, библиографического разыскания и описания, опыт работы с научными источниками

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

синтаксис языка программирования Python;
общепринятые способы решения базовых задач с использованием особенностей языка;
основные библиотеки и фреймворки на Python;
принцип исполнения программ на Python;
типы данных языка Python;
управление потоком выполнения в Python;
возможности стандартной библиотеки;
правила работы с исключениями;
внутреннее строение контейнеров стандартной библиотеки и временную сложность операций с ними;
принцип работы сборки мусора в Python;
кодировки, используемые при хранении текстовых данных (ASCII, Windows-1250/1251, UTF-8, UTF-16).

уметь:

реализовывать библиотеку общего назначения на языке Python по заданным интерфейсам;
решать задачи, связанные с обработкой данных, на языке Python.

владеть:

основными библиотеками и инструментами разработчика на языке Python.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Знакомство с Python		3		3
2	Работа с Jupyter, основы Python		6		6
3	Работа с NumPy		9		9
4	Хранение данных. Pandas		18		18
5	Визуализация данных		6		6
6	Машинное обучение в Python		3		3
Итого часов			45		45
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 1 (Осенний)

1. Знакомство с Python

Введение, почему Python, основные библиотеки, установка.

2. Работа с Jupyter, основы Python

Запуск Jupyter, структура Notebook-a, клетки, команды. Основы Python – объекты, функции, типы, импорты, control flow. Структуры данных (листы, кортежи, словари, set-ы), функции (аргументы, lambda-функции), работа с файлами.

3. Работа с NumPy

Числовые (numpy-)массивы, индексы, арифметика, оси и транспонирование, функции.

Векторизация, логика, сортировка, агрегация, чтение и запись numpy в файл, линейная алгебра.

Внутренности ndarray, конкатенация, tile, broadcasting, снова сортировки, быстрый NumPy – Numba.

4. Хранение данных. Pandas

Типы (Series, DataFrame), операции: индексы, drop, арифметика, функции, сортировка, ранжирование, статистика.

Текстовые файлы, JSON, XML и HTML, бинарные форматы (HDF5), доступ к API, БД.

Фильтрация, binning, outlier-ы, sampling, индикаторы, dummy-переменные, строки.

Иерархические индексы, join, merge, конкатенация, reshape, pivoting.

GroupBy (dict, series, функция), split-apply-combine, квантили.
Категориальные данные, еще немного GroupBy, метод pipe.

5. Визуализация данных

Matplotlib: figures, subplots, colors, markers, ticks, labels; линейные графики, гистограммы, scatterplot-ы.

6. Машинное обучение в Python

Простые модели: Patsy, statsmodels, scikit-learn.
Sclearn и pandas.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная компьютерами для каждого студента.

6. Перечень рекомендуемой литературы

Основная литература

1. "Python для анализа данных" (Wes McKinney) - данная книга предлагает обзор основных инструментов и библиотек Python для работы с данными, включая NumPy, pandas, matplotlib и другие.
2. "Python Data Science Handbook" (Jake VanderPlas) - эта книга представляет собой исчерпывающее руководство по применению Python для анализа данных, включая работу с массивами данных, визуализацию, машинное обучение и статистический анализ.
3. "Python for Data Analysis" (Wes McKinney) - в этой книге автор описывает основные инструменты и техники анализа данных с использованием Python и библиотеки pandas.

Дополнительная литература

1. "Data Science from Scratch" (Joel Grus) - данная книга предлагает введение в основы анализа данных с использованием Python, включая работу с данными, визуализацию, машинное обучение и статистику.
2. "Python Machine Learning" (Sebastian Raschka, Vahid Mirjalili) - эта книга охватывает применение Python для машинного обучения и анализа данных, включая различные алгоритмы и методы.
3. "Think Stats: Exploratory Data Analysis in Python" (Allen B. Downey) - автор представляет концепции и инструменты статистического анализа данных с использованием Python, с фокусом на исследовательском анализе данных.

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Видеолекции, доступные по ссылке:

<https://www.youtube.com/watch?v=VP2wRhwlG6c&list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwBl>

Основы программирования на Python

<https://www.coursera.org/learn/python-osnovy-programmirovaniya>

Язык программирования Python

<http://www.intuit.ru/studies/courses/49/49/info>

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

IDE PyCharm

Сборка python Anaconda: numpy, scipy, pandas, matplotlib, sclearn.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Можно изучать дополнительно материалы похожих курсов:

Основы программирования на Python

<https://www.coursera.org/learn/python-osnovy-programmirovaniya>

Язык программирования Python

<http://www.intuit.ru/studies/courses/49/49/info>

Литература для самостоятельного изучения

1. Марк Лутц, «Изучаем Python», Символ-плюс, 2010.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладная математика и информатика
профиль подготовки:	Прикладная математика и информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	<u>1</u>
квалификация:	магистр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Разработчики:

М.Е. Бекетов, ассистент

О.Н. Ивченко, старший преподаватель

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен решать актуальные задачи фундаментальной и прикладной математики	ОПК-1.3 Понимает междисциплинарные связи в области прикладной математики и информатики и способен их применять при решении задач профессиональной деятельности
ПК-2 Понимает и способен применить в научно-исследовательской и прикладной деятельности современный математический аппарат и алгоритмы, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии	ПК-2.1 Знает основы научно-исследовательской деятельности в области информационно-коммуникационных технологий владеет знанием основ философии и методологии науки; знанием методов научных исследований и навыками их проведения
	ПК-2.2 Имеет практический опыт использования существующих методов и алгоритмов решения задач цифровой обработки сигналов, использования сети Интернет, аннотирования, реферирования, библиографического разыскания и описания, опыт работы с научными источниками

2. Показатели оценивания компетенций

В результате изучения дисциплины «Применение Python в статистическом анализе данных» обучающийся должен:

знать:

синтаксис языка программирования Python;
общепринятые способы решения базовых задач с использованием особенностей языка;
основные библиотеки и фреймворки на Python;
принцип исполнения программ на Python;
типы данных языка Python;
управление потоком выполнения в Python;
возможности стандартной библиотеки;
правила работы с исключениями;
внутреннее строение контейнеров стандартной библиотеки и временную сложность операций с ними;
принцип работы сборки мусора в Python;
кодировки, используемые при хранении текстовых данных (ASCII, Windows-1250/1251, UTF-8, UTF-16).

уметь:

реализовывать библиотеку общего назначения на языке Python по заданным интерфейсам;
решать задачи, связанные с обработкой данных, на языке Python.

владеть:

основными библиотеками и инструментами разработчика на языке Python.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1. Импортируйте NumPy как `np`. Выясните, какая версия NumPy у вас установлена.
2. Команда `a = np.random.randn(10)` создает массив `a` из 10 случайных чисел (float-ов). Исследовать, насколько "экономнее" хранить его как NumPy-массив (`a` не как `list(a)`) – будем генерировать `a` из 10, 100, ..., 10⁶ таких случайных чисел, и будем смотреть на долю размера `a` от размера `list(a)`.
Размер в памяти какого-то объекта в Python можно определить функцией из библиотеки `sys` – системных утилит.
3. Импортируйте NumPy как `np`. Выясните, какая версия NumPy у вас установлена.

4. Создайте массив a случайных чисел `np.random.randn()` размеров (10,10,10). Найдите наибольший и наименьший элемент. Найдите сумму всех положительных элементов.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Примеры контрольных заданий:

1_3_2. Дан массив целых чисел $A[0..n-1]$. Известно, что на интервале $[0, m]$ значения массива строго возрастают, а на интервале $[m, n-1]$ строго убывают. Найти m за $O(\log m)$.
 $2 \leq n \leq 10000$.

2_4_1. Первые k элементов длинной последовательности.

Дана очень длинная последовательность целых чисел длины n . Требуется вывести в отсортированном виде её первые k элементов. Последовательность может не помещаться в память. Время работы $O(n * \log(k))$. Доп. память $O(k)$. Использовать слияние.

3_3_2. Порядковые статистики. Дано число N и N строк. Каждая строка содержит команду добавления или удаления натуральных чисел, а также запрос на получение k -ой порядковой статистики. Команда добавления числа A задается положительным числом A , команда удаления числа A задается отрицательным числом $-A$. Запрос на получение k -ой порядковой статистики задается числом k . Требуемая скорость выполнения запроса - $O(\log n)$.

4_4_1. Самая удаленная вершина.

Для каждой вершины определите расстояние до самой удаленной от нее вершины. Время работы должно быть $O(n)$.

Формат входных данных:

В первой строке записано количество вершин $n \leq 10000$. Затем следует $n - 1$ строка, описывающая ребра дерева. Каждое ребро – это два различных целых числа – индексы вершин в диапазоне

$[0, n-1]$. Индекс корня – 0. В каждом ребре родительской вершиной является та, чей номер меньше.

Формат выходных данных:

Выход должен содержать n строк. В i -ой строке выводится расстояние от i -ой вершины до самой удаленной от нее.

Критерии оценивания

отлично

10 всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

9 систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

8 глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

хорошо

7 твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

6 знает материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

5 знает основной материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач неточности;

удовлетворительно

4 фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

3 характер знаний достаточен для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

неудовлетворительно

2 не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет правильно использовать полученные знания при решении типовых практических задач.

1 не знает формулировок основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированный зачет может проводиться по итогам текущей успеваемости и сдачи заданий, лабораторных и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме, а также с выдачей заданий для реализации на компьютере.