Московский государственный технический университет   
им. Н. Э. Баумана

**Оперативный анализ данных**

*Методические указания к лабораторной работе*

*по теме «Оперативный анализ данных в python»*

Москва, 2019 г.

**Цель работы:** освоение техник работы с популярными библиотеками для анализа данных в Python

Продолжительность работы: *4 часа*.

**Теоретическая часть**

**Общие сведения об использующихся библиотеках**

**NumPy** — open-source модуль для python, который предоставляет общие математические и числовые операции в виде пре-скомпилированных, быстрых функций. Они объединяются в высокоуровневые пакеты. Они обеспечивают функционал, который можно сравнить с функционалом MatLab. NumPy (Numeric Python) предоставляет базовые методы для манипуляции с большими массивами и матрицами. SciPy (Scientific Python) расширяет функционал numpy огромной коллекцией полезных алгоритмов, таких как минимизация, преобразование Фурье, регрессия, и другие прикладные математические техники.  
Докуметация: <https://docs.scipy.org/doc/numpy/reference/index.html>

**Pandas** — программная библиотека на языке python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.  
Документация: <https://pandas.pydata.org/pandas-docs/stable/reference/index.html>

**Matplotlib** — библиотека на языке программирования Python для визуализации данных.  
Документация: <https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.html#module-matplotlib.pyplot>

**Seaborn** — библиотека на языке программирования Python, построенная поверх Matplotlib. Позволяет производить более красивую визуализацию.  
Документация: <https://seaborn.pydata.org/api.html>

**Scikit-learn** (aka **sklearn**) — библиотека на языке программирования Python, предоставляющая различные алгоритмы классификации, регрессии и кластеризации. Использует NumPy и SciPy.  
Документация: <https://scikit-learn.org/stable/modules/classes.html>

**Основные понятия**

**Series** – это проиндексированный одномерный массив значений. Он похож на простой словарь типа dict, где имя элемента будет соответствовать индексу, а значение – значению записи.

**DataFrame** — это проиндексированный многомерный массив значений, соответственно каждый столбец DataFrame является структурой Series.

**Регрессия** — задача восстановления зависимости между переменными. В случае машинного обучения у нас имеется произвольное число входных переменных (**x**) и одна выходная (**y**).

**Классификация** — задача отнесения объекта по набору его признаков (**x**) к одному из счетного множества классов (**y**).

**Кластеризация** — задача группировки множества объектов на подмножества (кластеры) таким образом, чтобы объекты из одного кластера были более похожи друг на друга, чем на объекты из других кластеров по какому-либо критерию.

**Обучение с учителем** — раздел машинного обучения, занимающийся решением задачи восстановления зависимости между признаками объектов и ответами (откликами).

**Обучение без учителя** — раздел машинного обучения, изучающий задачи, в которых известно только признаковое описание объекта и требуется обнаружить взаимосвязи между объектами.

**Регуляризация** — метод добавления дополнительных ограничений к условию задачи с целью предотвратить переобучение модели.

**Переобучение** — излишняя “подгонка” модели к данным обучения (которые имеются сейчас) таким образом, что на неизвестных ей ранее данных модель будет показывать очень плохие результаты (при этом на обучающих данных она показывает превосходные результаты).

**Подготовка к лабораторной работе**

1. Получить у лаборанта файлы с наборами данных (.csv) и IPython ноутбук (.ipynb).
2. Установить jupyter, numpy, matplotlin, seaborn и sklearn

**Функции которые пригодятся в лабораторной работе**

**Pandas**

|  |  |
| --- | --- |
| pd.read\_csv(df) | считывает данные df в pandas датафрейм |
| pd.head(n) | выводит n первых строк из датафрейма |
| pd.shape | выводит число объектов и колонок датафрейма |
| pd.columns | выводит названия колонок датафрейма |
| pd.value\_counts() | считает все возможные значения датафрейма |
| pd.groupby([1])[2] *(.agg())* | группирует данные 1 и 2 (agg отвечает за агрегацию по признаку |

**NumPy**

|  |  |
| --- | --- |
| np.sum | суммирует в numpy массив |
| np.mean | считает среднее по массиву |
| np.abs() | берет модуль от скобок |
| np.sqrt() | берет квадратный корень от скобок |
| np.dot() | Для двумерных массивов это эквивалентно умножению матриц, а для одномерных массивов - на внутреннее произведение векторов (без комплексного сопряжения). |
| a@b | то же самое, но   * умножение на скаляры недопустимо. * стеки матриц передаются вместе, как если бы матрицами были элементы. |
| np.power(a, b) | возводит а в степень b |

**Seaborn**

|  |  |
| --- | --- |
| sns.catplot(x,y,coll,data) | строит гистограмму распределения |
| sns.lineplot(x,y,hue,data) | строит линейный график зависимости |

**Практическая часть**

**Запуск программы**

1. Переместите .ipynb файл в папку проекта
2. Запустите jupyter notbook в своем виртуальном окружении python соответствующей командой (**jupyter notbook**)
3. В открывшемся веб-интерфейсе откройте .ipynb файл
4. Следуйте указаниям в ноутбуке

**Содержание отчета**

1. Титульный лист
2. Цель лабораторной работы
3. Скриншоты этапов выполнения работы с пояснениями
4. Вывод

**Контрольные вопросы:**

1. Назовите основные библиотеки для анализа данных в python
2. Что такое DataFrame?
3. Назовите перечисленные в методичке задачи машинного обучения и кратко опишите каждый
4. Назовите перечисленные в методичке разделы машинного обучения и кратко опишите каждый
5. Что такое переобучение и почему это плохо?
6. Назовите хотя бы 1 метод борьбы с переобучением