МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ

УЧЕБНОЕ ПОСОБИЕ (ЛАБОРАТОРНЫЙ ПРАКТИКУМ)

Направление подготовки:

09.03.02 «Информационные системы и технологии»

Квалификация выпускника: бакалавр

Ставрополь, 2024

УДК 004.41 (075.8)
ББК 22.18я73
Н 63

Печатается по решению
учебно-методического совета
Северо-Кавказского федерального
университета

Н 63 **Методы машинного обучения:** учебное пособие (лабораторный практикум) для студентов направления 09.03.02 «Информационные системы и технологии» / Николаев Е.И. – Ставрополь: Изд-во СКФУ, 2024. – _____ с.

Учебное пособие (лабораторный практикум) по дисциплине «Методы машинного обучения» для студентов направления 09.03.02 «Информационные системы и технологии». Пособие охватывает практические аспекты построения информационных систем на основе методов искусственного интеллекта с применением современных технологий разработки информационных систем и инструментария анализа данных. Основное внимание уделяется теории обучения машин (машинное обучение, machine learning). Пособие предназначено для студентов, обладающих теоретическими знаниями в области проектирования приложений и практическими навыками программирования (предпочтительно языки C++, C#, Python, Java, R). Цель учебного пособия: сформировать у студентов практические навыки разработки информационных систем на основе методов машинного обучения, искусственного интеллекта, анализа данных; сформировать систему компетенций.

УДК 004.41 (075.8)
ББК 22.18я73

Автор:

канд. техн. наук, доцент **Е.И. Николаев**

Рецензенты

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ **4**

ЛАБОРАТОРНАЯ РАБОТА 1. УТИЛИТЫ ЗАГРУЗКИ ДАННЫХ **6**

ЛАБОРАТОРНАЯ РАБОТА 2. ЗАГРУЗКА ДАННЫХ ИЗ ВНЕШНИХ ИСТОЧНИКОВ **13**

ПРОДОЛЖЕНИЕ В РАЗРАБОТКЕ… **22**

ЗАКЛЮЧЕНИЕ **23**

СПИСОК ЛИТЕРАТУРЫ **24**

ВВЕДЕНИЕ

1. Цели и задачи освоения дисциплины. Учебное пособие (лабораторный практикум) по дисциплине «Методы машинного обучения» для студентов направления 09.03.02 «Информационные системы и технологии». Пособие охватывает теоретические аспекты построения информационных систем на основе методов искусственного интеллекта. Основное внимание уделяется теории обучения машин (машинное обучение, machine learning).

Основная задача науки и реальной жизни – получение правильных предсказаний о будущем поведении сложных систем на основании их прошлого поведения. Многие задачи, возникающие в практических приложениях, не могут быть решены заранее известными методами или алгоритмами. Это происходит по той причине, что нам заранее не известны механизмы порождения исходных данных или же известная нам информация недостаточна для построения модели источника, генерирующего поступающие к нам данные. Как говорят, мы получаем данные из «черного ящика». В этих условиях ничего не остается, как только изучать доступную нам последовательность исходных данных и пытаться строить предсказания совершенствуя нашу схему в процессе предсказания. Подход, при котором про- шлые данные или примеры используются для первоначального формирования и совершенствования схемы предсказания, называется методом машинного обучения (Machine Learning). Машинное обучение – чрезвычайно широкая и динамически развивающаяся область исследований, использующая огромное число теоретических и практических методов. Данное пособие ни в какой мере не претендует на какое-либо исчерпывающее изложение содержания данной области. Основная цель – дать студентам теоретическое представление о современных математических проблемамах в области систем искусственного интеллекта, а также познакомить с путями их решения.

Пособие предназначено для студентов, обладающих теоретическими знаниями в области проектирования приложений и практическими навыками программирования (предпочтительно языки C, C++, C#, Python, R). Цель учебного пособия: сформировать у студентов целостный взгляд на современные тенденции в областях машинного обучения, искусственного интеллекта, анализа данныз; сформировать систему компетенций.

ЛАБОРАТОРНАЯ РАБОТА 1. УТИЛИТЫ ЗАГРУЗКИ ДАННЫХ

Цели и задачи

Изучение утилит Scikit-Learn для получения синтетических, готовых и сложных реальных наборов данных.

Теоретическое обоснование

Загрузка готового игрового набора данных из библиотеки Scikit-Learn

В начале знакомства с алгоритмами и методами машинного обучения, как правило, затруднительно заниматься загрузкой, преобразованием и очисткой реальных наборов данных. Для этого в scikit-learn есть готовые наборы данных, которые можно легко подключить. Они получили название «игрушечных» наборов данных (toy datasets), т. к. являются уже уменьшенным и очищенным вариантом реальных данных, с которыми мы могли бы работать. Ниже представлены некоторые популярные наборы данных scikit-learn.

Метод **`load_iris( )`** – ирисы Фишера (iris flowers) состоят из данных о 150 экземплярах ириса. Этот набор данных хорошо подходит для изучения алгоритмов классификации.

Метод **`load_digits( )`** – данный массив состоит из 1797 изображений начертания цифр. С его помощью можно научиться классификации изображений.

Различные методы, их описание и задачи для решения представлены в таблице:

Метод	Описание
`load_boston( )`	Набор данных о ценах на жилье в Бостоне (регрессия).
`load_iris( )`	Набор данных радужной оболочки (классификация).
`load_diabetes( )`	Набор данных диабета (регрессия).
`load_digits( )`	Набор данных цифр (классификация).

load_linnerud( )	Набор данных физических упражнений.
load_wine( )	Набор данных вина (классификация).
load_breast_cancer( )	Набор данных по раку груди висконсин (классификация).

Эти наборы данных полезны для быстрой иллюстрации поведения различных алгоритмов, реализованных в scikit-learn. Однако они часто слишком малы, чтобы соответствовать реальным задачам машинного обучения.

Для того чтобы ознакомиться с каждым набором данных, воспользуемся функцией вывода с атрибутом **DESCR**.

Более подрабную информацию по игровым наборам данных можно получить в документации: https://scikit-learn.org/stable/datasets/toy_dataset.html

Создание искусственного набора данных

В scikit-leam существует множество способов создания искусственных данных. Отдельно стоит выделить три подхода: **make_regression**, **make_classification** и **make_blobs**.

Функция **make_regression** возвращает матрицу признаков со значениями с плавающей запятой и вектор целевых данных со значениями с плавающей запятой, в то время как функции **make_classification** и **make_blobs** возвращают матрицы признаков со значениями с плавающей запятой и вектор целевых данных с целыми значениями, выражающими отношение к классу.

Благодаря использованию scikit-leam при создании наборов искусственных данных можно получить множество вариантов контроля типа генерируемых данных. Полное описание всех параметров содержится в руководстве scikit-leam:

https://scikit-learn.org/stable/datasets/sample_generators.html

Различные методы построения синтетических наборов представлены в таблице:

Метод	Описание
make_blobs( )	Мультиклассовый набор данных – облако точек с нормальным распределением вокруг центров кластера (классификация, кластеризация).
make_classification( )	Мультиклассовый набор данных (классификация, кластеризация).
make_circles( )	2D-датасет для бинарной классификации. Генерирует бинарный набор данных, в котором обрака точек располагаются по элиптичским орбитам – кольцам (классификация, кластеризация)
make_moons( )	2D-датасет для бинарной классификации. Генерирует бинарный набор данных с границей кластеров в виде полуколец. (классификация, кластеризация)
make_multilabel _classification( )	Набор данных для многоклассовой классификации (классификация, кластеризация).
make_regression( )	Синтетический набор данных для обучения регерссионной модели.

Рассморим некоторые параметры методов генерирования синтетичских наборов данных:

Количество признаков, используемых в функциях **make_regression** и **make_classification** для генерации вектора, определяется при помощи параметра **n_informative**. Если параметр **n_informative** меньше общего числа параметров (**n_features**), получившийся набор данных будет иметь избыточные параметры, которые можно определить с помощью методов выбора параметров.

Кроме того, функция **make_classification** содержит параметр **weights**, при помощи которого можно создать дисбаланс классов (imbalanced classes) в процессе генерации наборов данных. Например, задав параметр **wetghts = [.25, .75]**, мы получим набор данных, в котором 25% объектов будут принадлежать к одному классу, а 75% - к другому.

В функции **make_blobs** есть параметр **centers**, который позволяет нам задать число генерируемых кластеров.

Готовые наборы сложной структуры

Кроме рассмотренных игровых наборов данных и синтетичских наборов при проектировании систем ML можно использовать более сложные готовые наборы данных. Некоторые методы получения сложных наборов данных представлены в таблице:

Метод	Описание
fetch_olivetti_faces( )	В наборе данных есть 10 разных изображений лиц 40 разных людей. Есть десять различных изображений каждого из 40 различных предметов. Для некоторых объектов изображения были сделаны в разное время, варьируя освещение, выражение лица (открытые / закрытые глаза, улыбающийся / не улыбающийся) и детали лица (очки / без очков). Все изображения были сделаны на темном однородном фоне, когда испытуемые находились в вертикальном фронтальном положении (с допуском на некоторое боковое движение).
fetch_20newsgroups_vectorized( )	Набор данных из 20 групп новостей. Набор данных 20 групп новостей включает около 18000 сообщений групп новостей по 20 темам, разделенных на два подмножества: один для обучения (или разработки), а другой для тестирования (или для оценки производительности). Разделение между поездом и набором тестов основано на сообщениях, отправленных до и после определенной даты. Этот модуль содержит два загрузчика. Первый, fetch_20newsgroups возвращает список необработанных текстов, которые могут быть переданы экстракторам текстовых признаков, например, CountVectorizer с пользовательскими параметрами для извлечения векторов признаков. Второй, fetch_20newsgroups_vectorized возвращает готовые к использованию функции, т. е. нет необходимости использовать средство извлечения признаков.
fetch_lfw_people( )	Набор данных (фото людей) с метками «Размеченные фото из реальной жизни» (LFW, Labeled Faces in the Wild) (классификация). Этот набор данных представляет собой коллекцию изображений

	известных людей в формате JPEG, собранных через Интернет. аждое изображение сосредоточено на одном лице. Типичная задача называется Face Verification: для пары двух изображений двоичный классификатор должен предсказать, принадлежат ли эти два изображения одному и тому же человеку. Альтернативная задача, «Распознавание лиц» или «Идентификация лиц»: по изображению лица неизвестного человека определить имя человека, обратившись к галерее ранее увиденных изображений идентифицированных лиц. И проверка лиц, и распознавание лиц – это задачи, которые обычно выполняются на выходе модели, обученной выполнять обнаружение лиц. Самая популярная модель для распознавания лиц называется Виола-Джонса и реализована в библиотеке OpenCV. Лица LFW были извлечены этим детектором лиц с различных онлайн-сайтов.
fetch_lfw_pairs( )	Набор данных из пар (фото людей) с метками «Размеченные фото из реальной жизни» (LFW, Labeled Faces in the Wild) (классификация). Данный загрузчик аналогичен предыдущему и обычно используется для задачи проверки лица: каждый образец представляет собой пару из двух изображений, принадлежащих или не принадлежащих одному и тому же человеку.
fetch_covtype( )	Образцы в этом наборе данных соответствуют участкам леса 30×30 м в США, собранным для задачи прогнозирования типа покрытия каждого участка, то есть доминирующих видов деревьев. Существует семь типов обложек, что делает эту задачу мультиклассовой классификацией. Каждый образец имеет 54 признака
fetch_california_housing( )	Набор данных о жилье в Калифорнии (регрессия). Целевая переменная – это средняя стоимость дома для округов Калифорнии.

Методика и порядок выполнения работы

Перед выполнением индивидуального задания рекомендуется выполнить все пункты учебной задачи.

Учебная задача

Необходимо продемострировать данные, полученные с использованием утилит для загрузки: 1) готового набора данных; 2) синтетического набора данных; 3) сложного реального набора данных.

После получения набора данных необходимо реализовать визуализацию нескольких (или всех) экземпляров из набора данных.

Решение задачи

Изучите код решения учебной задачи:
https://github.com/enikolaev/AI_and_ML/blob/main/LabWork_01.ipynb

Индивидуальное задание

В соотвтествии с методикой, представленной в учебной задаче, реализуйте загрузку и визуализацию данных с использованием следующих утилит:

Вариант	Утилиты, использование которых необходимо продемонстрировать		
	Готовый набор данных	Синтетический набор данных	Набор данных сложной структуры
1	load_boston	make_blobs	fetch_olivetti_faces
2	load_iris	make_classification	fetch_20newsgroups_vectorized
3	load_diabetes	make_circles	fetch_lfw_people
4	load_digits	make_moons	fetch_lfw_pairs
5	load_linnerud	make_s_curve	fetch_covtype
6	load_wine	make_regression	fetch_california_housing
7	load_breast_cancer	make_moons	fetch_olivetti_faces
8	load_boston	make_classification	fetch_lfw_pairs
9	load_iris	make_blobs	fetch_covtype
10	load_diabetes	make_classification	fetch_olivetti_faces
11	load_digits	make_circles	fetch_20newsgroups_vectorized
12	load_linnerud	make_moons	fetch_lfw_people

13	load_wine	make_s_curve	fetch_lfw_pairs
14	load_breast_cancer	make_regression	fetch_covtype
15	load_boston	make_blobs	fetch_california_housing
16	load_iris	make_blobs	fetch_olivetti_faces
17	load_diabetes	make_moons	fetch_olivetti_faces
18	load_digits	make_blobs	fetch_20newsgroups_vectorized
19	load_linnerud	make_classification	fetch_covtype
20	load_wine	make_circles	fetch_olivetti_faces
21	load_breast_cancer	make_moons	fetch_20newsgroups_vectorized
22	load_digits	make_s_curve	fetch_lfw_people
23	load_linnerud	make_regression	fetch_lfw_pairs
24	load_wine	make_classification	fetch_covtype
25	load_boston	make_moons	fetch_california_housing

Контрольные вопросы

1. Сколько признаков у каждого экземпляра в наборе данных, получаемыъ утилитой **`sklearn.datasets.make_moons( )`**?

2. Опишите назначение метода **`numpy.random.seed( )`**.

3. Опишите синтетические наборы данных, которые целесообразно использовать для задач кластеризации.

4. Опишите игровые наборы данных, которые целесообразно использовать для регрессии.

5. Опишите сложные реальные наборы данных, которые целесообразно использовать для классификации.

ЛАБОРАТОРНАЯ РАБОТА 2. ЗАГРУЗКА ДАННЫХ ИЗ ВНЕШНИХ ИСТОЧНИКОВ

Цели и задачи

Изучение методов загрузки данных из внешних источников различного формата.

Теоретическое обоснование

Создание любой модели машинного обучения всегда начинается с загрузки необработанных данных в систему. Эти данные могут включать логи (журнальные файлы), наборы данных или облачные хранилища BLOB-объектов (например, Amazon S3).

Более того, нам зачастую необходимо получать данные из различных источников. В данной работе необходимо изучить методы методы загрузки данных из различных источников, включая CSV-файлы и базы данных SQL. Кроме того, необходимо исследовать способы генерирования искусственных данных с заданными параметрами для тестирования. В процессе изучения различных методов загрузки данных в экосистеме Python уделите особое внимание методам загрузки внешних данных при помощи инструментов модуля Pandas и созданию искусственных данных посредством библиотеки Scikit-Learn, написанной на языке Python и предназначенной для машинного обучения.

В таблице показаны методы библиотеки Pandas для загрузки данных из различных источников.

Метод	Описание
`read_csv( )`	Загрузка CSV-файла
`read_excel( )`	Загрузка файла Excel
`read_json( )`	Загрузка файла JSON

Загрузка CSV-файла

Пример:

```
url ='…'
data_csv = pd.read_csv(url)
data_csv.head()
```

Стоит отметить две вещи, касающиеся загрузки CSV-файлов. Во-первых, будет полезно просмотреть содержимое файла перед его загрузкой. Таким образом, вы заранее ознакомитесь со структурой набора данных и параметрами, необходимыми для загрузки файла. Во-вторых, в **read_csv** представлено более 30 параметров, и процесс изучения документации может показаться сложным. Однако большинство этих параметров необходимы для управления разнообразными CSV-форматами.

Как следует из названия, значения в CSV-файлах буквально разделяются запятыми (например, строка может иметь такой вид:

```
2, "2015-01-01 00:00:00" ,0
```

Тем не менее в CSV-файлах часто используются и другие разделители, такие как табуляция (в таком случае используется TSV-файл).

При помощи параметра **sep** в **pandas** мы можем задать разделитель, который будет использован в файле. В структуре CSV-файлов принято, хотя и не обязательно, использовать первую строку для заголовков столбцов. Для того чтобы определить наличие и расположение строки с заголовками, мы можем применять параметр **header**. Если такая строка отсутствует, необходимо указать **header=None**.

Функция **read_csv** возвращает DataFrame (фрейм данных) – основной тип данных в pandas, используемый для работы с таблицами.

Загрузка файла Excel

Пример:

```
url = '…'
data_excel = pd.read_excel(url,sheet_name=0)
```

```
data_excel.head()
```

Наше решение похоже на предыдущий вариант для чтения CSV-файлов. Главное отличие заключается в добавлении параметра **sheet_name**, при помощи которого мы указываем номер нужного листа в документе Excel. Параметр **sheet_name** может принимать как строковые значения (strings) с названием листа, так и числовые значения (integers), указывающие на номер листа (индексация начинается с нуля). Для загрузки нескольких листов необходимо перечислить их. К примеру, если мы зададим параметр:

```
sheet_name =[0, 1, 2, "Monthly Sales"]
```

то получим словарь (dictionary) значений DataFrames библиотеки pandas, содержащий первый, второй и третий листы, а также лист с названием Monthly Sales.

Загрузка файла JSON

Пример:

```
url = '…'
data_json = pd.read_json(url)
data_json.head()
```

Импорт файлов JSON в pandas похож на другие операции, описанные выше. Основным отличием является наличие параметра **orient** для описания структуры файла JSON при добавлении в pandas. Также можно попробовать разные аргументы (**split**, **records**, **index**, **columns** или **values**), чтобы опытным путем понять, какой подходит лучше. Еще одним удобным инструментом pandas является функция **json_normalise**, при помощи которой слабоструктурированные файлы JSON преобразовываются в DataFrame.

Методика и порядок выполнения работы

Перед выполнением индивидуального задания рекомендуется выполнить все пункты учебной задачи.

Учебная задача

Необходимо продемострировать данные, полученные с использованием методов загрузки данных: 1) **`read_csv( )`**; 2) **`read_excel( )`**; 3) **`read_json( )`**.

После получения набора данных необходимо показатать доступность загруженных данных. Для этого необходимо испоользовать метиоды **`describe( )`**, **`head( )`**, **`tail( )`** класса **`DataFrame`**.

Решение задачи

Изучите код решения учебной задачи:
https://github.com/enikolaev/AI_and_ML/blob/main/LabWork_02.ipynb

Индивидуальное задание

В данной работе необходимо использовать наборы данных, которые обучающийся находит самостоятельно. Необходимо продемнстрировать все три способа чтения данных: 1) **`read_csv( )`**; 2) **`read_excel( )`**; 3) **`read_json( )`**.

После получения набора данных необходимо показатать доступность загруженных данных. Для этого необходимо испоользовать метиоды **`describe( )`**, **`head( )`**, **`tail( )`** класса **`DataFrame`**.

Примеры ресурсов, на которых можно получить наборы данных. Если нет готового набора данных в требуемом формате, учащийся может конвертировать данный в требуемый формат самостоятельно.

1. Mall Customers Dataset – данные посетителей магазина: id, пол, возраст, доход, рейтинг трат. (*Вариант применения: Customer Segmentation Project with Machine Learning*)
2. Iris Dataset – датасет для новичков, содержащий размеры чашелистиков и лепестков для различных цветков.
3. MNIST Dataset – датасет рукописных цифр. 60 000 тренировочных изображений и 10 000 тестовых изображений.

4. The Boston Housing Dataset – п опулярный датасет для распознавания паттернов. Содержит информацию о домах в Бостоне: количество квартир, стоимость аренды, индекс преступлений.

5. Fake News Detection Dataset – содержит 7796 записей с разметкой новостей: правда или ложь. (*Вариант применения с исходником на Python: Fake News Detection Python Project*)

6. Wine quality dataset – содержит информацию о вине: 4898 записей с 14 параметрами.

7. SOCR data – Heights and Weights Dataset – хороший вариант для старта. Содержит 25 000 записей о росте и весе 18-ти летних людей.

8. Parkinson Dataset – 195 записей о пациентах с болезнью Паркинсона, с 25 параметрами анализов. Можно использовать для предварительной оценки отличия больных людей от здоровых. (*Вариант применения с исходником на Python: Machine Learning Project on Detecting Parkinson's Disease*)

9. Titanic Dataset – содержит информацию про пассажиров (возраст, пол, родственники на борту и пр) 891 в тренировочном сете и 418 – в тестовом.

10. Uber Pickups Dataset – информация о 4.5 миллионах поездок на Uber 2014 года и 14 млн. 2015 года. (*Вариант применения с исходником на R: Uber Data Analysis Project in R*)

11. Chars74k Dataset – содержит изображения Британских и Канадских символов 64 классов: 0-9, A-Z, a-z. 7700 7.7k естественных изображений, 3400knаписанных от руки, 62000 синтезированных компьютером шрифтов.

12. Credit Card Fraud Detection Dataset – содержит информацию о транзакциях скомпрометированных кредитных картах. (*Вариант применения с исходником: Credit Card Fraud Detection Machine Learning Project*)

13. Chatbot Intents Dataset – JSON-файл, который содержит различные тэги: greetings, goodbye, hospital_search, pharmacy_search, и тд. Содержит набор шаблонов «вопрос-ответ». (*Вариант применения с исходником на Python: Chatbot Project in Python*)

14. Enron Email Dataset – содержит пол миллиона писем от 150 менеджеров Enron.

15. The Yelp Dataset – содержит 1,2 млн. рекомендаций от 1,6 млн. пользователей про 1,2 млн организаций.

16. Jeopardy Dataset – более 200 000 записей «вопрос-ответ» из популярной телевизионной игры.

17. Recommender Systems Dataset – портал с коллекцией датасетов от университета UCSD. Содержит записи об отзывах на популярных сайтах (Goodreads, Amazon). Отлично подходит для создания рекомендательных систем. (*Вариант применения с исходником на R: Movie Recommendation System Project in R*)

18. UCI Spambase Dataset – датасет для тренировки для обнаружения спама. Содержит 4601 писем с 57 параметрами метаданных.

19. Flickr 30k Dataset – более 30 000 изображений и подписей к ним. *(Flickr 8k Dataset – 8000 изображений. Проект с исходником на Python: Image Caption Generator Python Project)*

20. IMDB reviews – 25 000 отзывов на фильмы в тренировочном наборе и 25 000 в тестовом. (*Вариант применения с исходником на R: Sentiment Analysis Data Science Project*)

21. MS COCO dataset – 1,5 млн размеченных изображений.

22. CIFAR-10 and CIFAR-100 dataset – CIFAR-10 содержит 60,000 маленьких изображений 32*32 pixels цифр 0-9. CIFAR-100 – соответственно, 0-100.

23. GTSRB (German traffic sign recognition benchmark) Dataset – 50 000 изображений 43 дорожных знаков. (*Вариант применения с исходником на Python: Traffic Signs Recognition Python Project*)

24. ImageNet dataset – содержит более 100 000 фраз и около 1000 изображений на фразу.

25. Breast Histopathology Images Dataset – датасет содержит изображения образцов рака молочной железы. (*Вариант применения с исходником на Breast Cancer Classification Python Project*)

26. Cityscapes Dataset – содержит высококачественные аннотации видеопоследовательностей улиц разных городов.

27. Kinetics Dataset – содержит URL-ссылку на около 6,5 миллионов высококачественных видео.

28. MPII human pose dataset – датасет содержит 25 000 изображений человеческих поз с аннотацией по суставам.

29. 20BN-something-something dataset v2 – набор высококачественных видео, которые показывают, как человек выполняет какие-то действия.

30. Object 365 Dataset – датасет высококачественных изображений с ограничивающими рамками объектов.

31. Photo sketching dataset – содержит более 1000 изображений с их контурными чертежами.

32. CQ500 Dataset – датасет содержит 491 КТ-сканирование головы с 193 317 срезами.

33. IMDB-Wiki dataset – датасет с более чем 5 млн. изображений лиц с пометкой пола и возраста. (*Вариант применения с исходником на Gender & Age Detection Python Project*)

34. Youtube 8M Dataset – маркированный набор данных видео, который содержит 6,1 миллиона идентификаторов видео Youtube

35. Urban Sound 8K dataset – набор городских звуковых данных (содержит 8732 городских звука из 10 классов).

36. LSUN Dataset – набор данных из миллионов цветных изображений сцен и объектов (около 59 миллионов изображений, 10 различных категорий сцен и 20 различных категорий объектов).

37. RAVDESS Dataset – аудиовизуальная база данных эмоциональной речи. (Вариант *применения с исходником на Speech Emotion Recognition Python Project*)

38. Librispeech Dataset – датасет содержит 1000 часов английской речи с разными акцентами.

39. Baidu Apolloscape Dataset – датасет для развития технологий самостоятельного вождения.

40. Quandl Data Portal – хранилище экономических и финансовых данных (есть бесплатный и платный контент).

41. The World Bank Open Data Portal – информация о займах, выданных Всемирным банком развивающимся странам.

42. IMF Data Portal – портал международного валютного фонда, который публикует данные о международных финансах, ставках долга, инвестициях, валютных резервах и товарах.

43. American Economic Association (AEA) Data Portal – ресурс для поиска макроэкономических данных США.

44. Google Trends Data Portal – данные о тенденциях Google можно использовать для визуального изучения и анализа данных.

45. Financial Times Market Data Portal – ресурс для получения актуальной информации о финансовых рынках со всего мира.

46. Data.gov Portal – портал открытых данных правительства США (сельское хозяйство, здравоохранение, климат, образование, энергетика, финансы, наука и исследования и т.д.).

47. Data Portal: Open government data (India) – открытая правительственная платформа данных Индии.

48. Food environment Atlas Data Portal – содержит данные исследований о питании в США.

49. Health Data Portal – это портал Министерства здравоохранения и социальных служб США.

50. Centers for Disease Control and Prevention Data Portal – содержит широкий спектр данных, связанных со здоровьем.

51. London Datastore Portal – данные о жизни людей в Лондоне.

52. Canada Government Open Data Portal – портал открытых данных о канадцах (сельское хозяйство, искусство, музыка, образование, правительство, здравоохранение и т.д.)

Контрольные вопросы

1. Поясните назначение параметров метода **`read_csv( )`**.
2. Поясните назначение параметров метода **`read_excel( )`**.
3. Поясните назначение параметров метода **`read_json( )`**.
4. Поясните назначение меода **`describe( )`** и назначение параметров метода.
5. Поясните назначение меодов **`head( )`** и **`tail( )`**, а также назначение параметров методов.

ПРОДОЛЖЕНИЕ В РАЗРАБОТКЕ…

ЗАКЛЮЧЕНИЕ

Учебное пособие (лабораторный практикум) по дисциплине «Методы машинного обучения» для студентов направления 09.03.02 «Информационные системы и технологии». Пособие охватывает теоретические аспекты построения информационных систем на основе методов машинного обучения, а также предлагает студентам практические рекомендации по разработке интеллектуальных систем. Основное внимание уделяется теории обучения машин (машинное обучение, machine learning).

В пособии рассмотрены практические аспекты проектирования и разработки информационных систем для решения различных задач с использованием следующих подходов: линейных методов классификации, алгоритмов восстановления регрессии, алгоритмов логической классификации, кластерного анализа.

Многие задачи, возникающие в практических приложениях, не могут быть решены заранее известными методами или алгоритмами. Это происходит по той причине, что нам заранее не известны механизмы порождения исходных данных или же известная нам информация недостаточна для построения модели источника, генерирующего поступающие к нам данные. Машинное обучение – чрезвычайно широкая и динамически развивающаяся область исследований, использующая огромное число теоретических и практических методов.

СПИСОК ЛИТЕРАТУРЫ

Список основной литературы

1. Уэс, Маккинли. Python и анализ данных Электронный ресурс / Маккинли Уэс ; пер. А. А. Слинкин. - Python и анализ данных,2024-04-19. - Саратов : Профобразование, 2017. - 482 c. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0046-7, экземпляров неограниченно.

2. Сузи, Р.А. Язык программирования Python Электронный ресурс : учебное пособие / Р.А. Сузи. - Язык программирования Python,2020-07-28. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 350 c. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 5-9556-0058-2, экземпляров неограниченно

Список дополнительной литературы

3. Стенли, Липпман. Язык программирования C++ Электронный ресурс : Полное руководство / Липпман Стенли, Лажойе Жози ; пер. А. Слинкин. - Язык программирования C++,2024-04-19. - Саратов : Профобразование, 2017. - 1104 c. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0136-5, экземпляров неограниченно

4. https://github.com/enikolaev/AI_and_ML – Репозиторий с примерами кода из лабораторных работ.

5. https://archive.ics.uci.edu/ml/index.html – Репозиторий наборов данных для машинного обучения (Центр машинного обучения и интеллектуальных систем).

6. https://www.kaggle.com – Портал и система проведения соревнований по проблемам анализа данных.

7. https://www.mockaroo.com – Сайт для генерации наборов данных.

Методы машинного обучения

УЧЕБНОЕ ПОСОБИЕ (ЛАБОРАТОРНЫЙ ПРАКТИКУМ)

Автор

Николаев Евгений Иванович