федеральное государственное бюджетное образовательное учреждение высшего образования «Красноярский государственный медицинский университет имени профессора В.Ф. Войно-Ясенецкого» Министерства здравоохранения Российской Федерации

УТВЕРЖДАЮ
Проректор по учебной работе, д.м.н., доцент
И.А. Соловьева
«20» октября 2021 г.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ
по дисциплине
Машинное обучение и нейронные сети в медицине
для подготовки обучающихся по направлению подготовки 38.04.02 Менеджмент, направленность (профиль) «Управление в здравоохранении на основе интеллектуального анализа данных»

Красноярск
2021

Практическое занятие №1

Тема: Введение в R и Python.

Разновидность занятия: комбинированное.

Методы обучения: объяснительно-иллюстративный, репродуктивный, метод проблемного изложения, частично-поисковый, исследовательский.

Значение темы (актуальность изучаемой проблемы): Python и R давно стали стандартом для Data Science. Суть их противостояния в том, что оба языка прекрасно подходят для работы со статистикой. В то время как Python характеризуется понятным синтаксисом и большим количеством библиотек, язык R разрабатывался целенаправленно для специалистов по статистике, а посему оснащён качественной визуализацией данных.

Формируемые компетенции: ПК-2.1.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Организация занятия	5.00	Проверка посещаемости и внешнего вида обучающихся
2	Формулировка темы и целей	10.00	Озвучивание преподавателем темы и ее актуальности, целей занятия
3	Контроль исходного уровня знаний и умений	10.00	Тестирование, индивидуальный устный или письменный опрос, фронтальный опрос
4	Раскрытие учебно-целевых вопросов по теме занятия	10.00	Изложение основных положений темы
5	Самостоятельная работа обучающихся (текущий контроль)	40.00	Выполнение практического задания
6	Итоговый контроль знаний (письменно или устно)	10.00	Тесты по теме, ситуационные задачи
7	Задание на дом (на следующее занятие)	5.00	Учебно-методические разработки следующего занятия и методические разработки для внеаудиторной работы по теме

	ВСЕГО	90	

Аннотация (краткое содержание темы):

Python и R давно стали стандартом для Data Science. Суть их противостояния в том, что оба языка прекрасно подходят для работы со статистикой. В то время как Python характеризуется понятным синтаксисом и большим количеством библиотек, язык R разрабатывался целенаправленно для специалистов по статистике, а посему оснащён качественной визуализацией данных (https://proglib.io/p/r-and-python/ или https://tproger.ru/articles/python-vs-r-for-data-science/)

Обзор библиотек для машинного обучения на Python

Scikit-learn — библиотека машинного обучения на языке программирования Python с открытым исходным кодом. Содержит реализации практически всех возможных преобразований, и нередко ее одной хватает для полной реализации модели. В данной библиотеки реализованы методы разбиения датасета на тестовый и обучающий, вычисление основных метрик над наборами данных, проведение Кросс-валидация. В библиотеке также есть основные алгоритмы машинного обучения: линейной регрессии и её модификаций Лассо, гребневой регрессии, опорных векторов, решающих деревьев и лесов и др. Есть и реализации основных методов кластеризации. Кроме того, библиотека содержит постоянно используемые исследователями методы работы с признаками: например, понижение размерности методом главных компонент. Частью пакета является библиотека imblearn, позволяющая работать с разбалансированными выборками и генерировать новые значения.

Примеры кода

Линейная регрессия

```
# Add required imports
 import matplotlib.pyplot as plt
 import numpy as np
 from sklearn import datasets
 from sklearn.linear_model import LinearRegression
 from sklearn.metrics import mean_squared_error, r2_score
```

Загрузка датасета:

```
diabetes = datasets.load_diabetes()
# Use only one feature
diabetes_X = diabetes.data[:, np.newaxis, 2]
```

Разбиение датасета на тренировочный и тестовый:

```
# Split the data into training/testing sets
x_train = diabetes_X[:-20]
x_test = diabetes_X[-20:]
# Split the targets into training/testing sets
y_train = diabetes.target[:-20]
y_test = diabetes.target[-20:]
```

Построение и обучение модели:

```
lr = LinearRegression()
lr.fit(x_train, y_train)
predictions = lr.predict(x_test)
```

Оценка алгоритма:

```
# The mean squared error
print("Mean squared error: %.2f" % mean_squared_error(y_test, predictions))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % r2_score(y_test, predictions))
 > Mean squared error: 2548.07
 Variance score: 0.47
```

Построение графика прямой, получившейся в результате работы линейной регрессии:

```
plt.scatter(x_test, y_test, color='black')
plt.plot(x_test, predictions, color='blue', linewidth=3)
plt.xticks(())
plt.yticks(())
plt.show()
```

Tensorflow — библиотека, разработанная корпорацией Google для работы с тензорами, используется для построения нейронных сетей. Поддержка вычислений на видеокартах имеет поддержку языка программирования C++. На основе данной библиотеки строятся более высокоуровневые библиотеки для работы с нейронными сетями на уровне целых слоев. Так, некоторое время назад популярная библиотека Keras стала использовать Tensorflow как основной бэкенд для вычислений вместо аналогичной библиотеки Theano. Для работы на видеокартах NVIDIA используется библиотека cuDNN. Если вы работаете с картинками (со сверточными нейросетями), скорее всего, придется использовать данную библиотеку.

Keras — библиотека для построения нейронных сетей, поддерживающая основные виды слоев и структурные элементы. Поддерживает как рекуррентные, так и сверточные нейросети, имеет в своем составе реализацию известных архитектур нейросетей (например, VGG16). Некоторое время назад слои из данной библиотеки стали доступны внутри библиотеки Tensorflow. Существуют готовые функции для работы с изображениями и текстом. Интегрирована в Apache Spark с помощью дистрибутива dist-keras. Данная библиотека позволяет на более высоком уровне работать с нейронными сетями. В качестве библиотеки для бэкенда может использоваться как Tensorflow, так и Theano.

Вспомогательные библиотеки

NumPy — библиотека, добавляющая поддержку больших многомерных массивов и матриц вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами. Данная библиотека предоставляет реализации вычислительных алгоритмов (в виде функций и операторов), оптимизированные для работы с многомерными

массивами. В результате любой алгоритм, который может быть выражен в виде последовательности операций над массивами (матрицами) и реализованный с использованием NumPy, работает так же быстро, как эквивалентный код, выполняемый в MATLAB;

SciPy — открытая библиотека высококачественных научных инструментов для языка программирования Python. SciPy содержит модули для оптимизации, интегрирования, специальных функций, обработки сигналов, обработки изображений, генетических алгоритмов, решения обыкновенных дифференциальных уравнений и других задач, обычно решаемых в науке и при инженерной разработке;

Pandas — библиотека Python, которая является мощным инструментом для анализа данных. Пакет дает возможность строить сводные таблицы, выполнять группировки, предоставляет удобный доступ к табличным данным и позволяет строить графики на полученных наборах данных при помощи библиотеки Matplotlib;

Matplotlib — библиотека Python для построения качественных двумерных графиков. Matplotlib является гибким, легко конфигурируемым пакетом, который вместе с NumPy, SciPy и IPython предоставляет возможности, подобные MATLAB.

Autograd - Библиотека автодифференциирования функций на numpy. ПОзволяет делать простые нейросети и оптимизацию научных расчётов. Для тяжёлого лучше использовать GPU-библиотеки.

Библиотеки для глубокого обучения

Tenzorflow - открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия. Широко применяется в бизнес-приложениях.

PyTorch — библиотека для глубокого обучения, созданная на базе Torch и развиваемая компанией Facebook. Две ключевые функциональности данной библиотеки — тензорные вычисления с развитой поддержкой ускорения на GPU (OpenCL) и глубокие нейронные сети на базе системы autodiff;

Theano — расширение языка программирования Python, позволяющее эффективно вычислять математические выражения, содержащие многомерные массивы. Библиотека предоставляет базовый набор инструментов для конфигурации нейронных сетей и их обучения. Наибольшее признание данная библиотека получила в задачах машинного обучения при решении задач оптимизации. Она позволяет использовать возможности GPU без изменения кода программы, что делает ее незаменимой при выполнении ресурсоемких задач;

Caffe — фреймворк для обучения нейронных сетей, созданный университетом Беркли. Как и Tensorflow, использует cuDNN для работы с видеокартами NVIDIA;

Microsoft Cognitive Toolkit (CNTK) — фреймворк от корпорации Microsoft, предоставляющий реализации архитектур различных нейронных сетей.

Особенности написания кода на R

Язык R изначально создавался как язык программирования для работы с графикой и статистической обработки данных. Поэтому он отличается большим количеством реализованных статистических алгоритмов, на основе которых можно создавать модели и алгоритмы машинного обучения.

Язык постоянно расширяется за счёт новых библиотек (пакетов). Для импорта одного пакета необходимо прописать в файле следующие строки:

```
install.packages("packageName")
require("packageName")
```

Для того чтобы импортировать пакет с его зависимостями в код следует включить следующие строки:

```
library("packageName")
```

Для языка R написано много пакетов, каждый из которых предназначен для решения определенного круга проблем. Например, для обработки данных или реализации основных алгоритмов. Рассмотрим несколько наиболее часто используемых пакетов.

Пакеты для обработки данных

Пакет Pipelearner предоставляет базовые возможности для разбиения набора данных на блоки для обучения моделей. В основе пакета лежит концепция работы конвейера. Принцип работы очень прост и описывается 3 шагами:

Инициализация. Функция pipelearner() инициализирует новый объект, который используется в следующих функциях обработки. На этом этапе необходимо указать датасет, с которым производится работа. Также можно указать набор обучающих моделей и предсказываемую модель данных.

Настройка. Для настройки есть 3 основных функции:

learn_curves() служит для настройки кривых обучения. Используется метод увеличивающихся пропорций относительно начала датасета.

learn_cvpairs() отвечает за кросс-валидацию. Функция генерирует набор пар из тестовой и обучающей выборки на основе входного датасета.

learn_models() предназначен для добавления новых обучающих моделей.

Обучение. С помощью функции learn() все сконструированные ранее модели обучаются и выдается таблица результатов работы

В итоге работа с пакетом выглядит приблизительно следующим образом:

```
# Load the dependencies
library(pipelearner)
library(dplyr)
iris %>% # Use iris dataset
pipelearner() %>% # Initialize a blank pipelearner object
```

learn_cvpairs(crossv_mc, n = 50) %>% # Creating 50 random cross-validation pairs

learn_curves(seq(.5, 1, by = .1)) %>% # Copy each cv-pair to be fitted in sample size proportions of .5 to 1 in increments of .1.

learn_models(lm, Sepal.Width ~ .*.) %>% # Use regression modell

learn_models(rpart::rpart, Sepal.Width ~ .) %>% # Use decision tree modell

learn() # Fit all models on all partitions and return the results

Пакет хорошо документирован, все непонятные моменты можно прояснить, просто изучив структуру объекта на каждом этапе работы алгоритма.

Пакет MICE используется для заполнения пропущенных значений в данных. При этом нет необходимости думать о типах значений: для каждого из них в пакете предусмотрено заполнение по умолчанию.

Принцип работы основан на методе множественного восстановления. Пропущенные данные заполняются не один, а несколько раз. После этого, каждый из полученных наборов обучается на определенной модели. Затем, результаты агрегируются и выдаются итоговые параметры модели.

Стандартный процесс работы выглядит так:

Load the dependencies

library(mice)

Impute the missing data m times

imp m = 5)

Analize completed datasets using linear model

fit # Combine parameter estimates

est # Print summary of estimation

summary(est)

Пакет Ggplot2 используется для отрисовки данных и графиков.

Пакеты с реализованными алгоритмами машинного обучения

Caret. В данном пакете представлены модели для регрессии и классификации, а также большая часть популярных метрик. В настоящее время имеется возможность использовать более 180 различных алгоритмов. Основная функция в составе Caret — функция train(). Параметры обучения в ней задаются аргументом trControl, а оценка качества модели — аргументом metric. Отличительными особенностями Caret является универсальность используемых команд, наличие автоматического подбора гиперпараметров для алгоритмов, в также наличие параллельных вычислений.

Пакет Party содержит в себе инструменты для рекурсивного разбиения данных на классы. В пакета также доступна расширяемая функциональность для визуализации древовидных регрессионных моделей. Основная функция пакета — ctree(), которая используется для создания деревьев решения для таких задач регрессии как номинальные, порядковые, числовые а также многовариантные переменные отклика. На основе деревьев условного вывода cforest() предоставляет реализацию случайных лесов Бреймана. Функция mob() реализует алгоритм рекурсивного разделения на основе параметрических моделей (например, линейных моделей, GLM или

регрессии выживания), использующих тесты нестабильности параметров для выбора разделения.

RandomForest — пакет с реализацией алгоритма случайного леса. Используется для решения задач регрессии и классификации, а также для поиска аномалий и отбора предикторов.

Пакет ClusterR состоит из алгоритмов кластеризации на основе центроидов (метод K-средних (k-means), mini-batch-kmeans, k-medoids) и распределений (GMM). Кроме того, пакет предлагает функции для:

проверки результатов,

построения графика результатов, используя метрики

прогнозирования новых наблюдения,

оценки оптимального количества кластеров для каждого алгоритма

Пакет E1071 содержит в себя функции для анализа классов, кратковременного преобразование Фурье, нечеткой кластеризации, реализации метода опорных векторов, вычисления кратчайшего пути, а также реализации наивного байесовского классификатора.

В пакете Mlr представлены модели для регрессии, классификации, кластеризации и анализа выживаемости, а также широкие возможности для оценки качества (в том числе функции для анализа ROC-кривых). Есть поддержка параллельных вычислений и конвейерных операций.

В пакете H20 представлены линейные модели, такие как градиентный бустинг, метод главных компонент (PCA), GLRM, метод k ближайших соседей, случайный лес, наивный байесовский классификатор. Сильная сторона этой библиотеки — работа с большими объемами данных и поддержка многопоточных вычислений. Однако в ней нет возможности задавать параметры используемых алгоритмов

Примеры алгоритмов

Задачи регрессии

Линейная регрессия

```
# reading data
data "input.csv", sep = ',', header = FALSE)
# evaluating linear regression model
model x ~ data$y)
# getting summary
print(summary(model))
# visualizing data
plot(data$y, data$x)
lines(data$y, predict(fit), col = 'red')
```

Множественная регрессия

```
# reading data
rdata "input.csv", sep = ',', header = FALSE)
# evaluating regression model
model data = rdata)
# getting summary
```

```
print(summary(model))
```

Логистическая регрессия

Логистическая регрессия – это модель регрессии, в которой переменная ответа принимает значения 0 или 1 (True или False). Реализация на языке R представлена в следующем фрагменте:

```
# reading data
rdata "input.csv", sep = ',', header = FALSE)
# evaluating model
model = glm(formula = target ~ x + y + z, data = rdata, family = binomial)
# printing summary
print(summary(model))
```

Метод главных компонент

```
# importing library and its' dependencies
library(h2o)
h2o.init()
path "extdata", "data.csv", package = "h2o")
data path = data)
# evaluating
h2o.prcomp(training_frame = data, k = 8, transform = "STANDARDIZE")
```

Деревья решений, случайный лес

Деревья решений

Для создания деревьев решений в R используется функция ctree() из пакета party.

```
# importing package
install.packages("party")
# reading data
rdata "input.csv", sep = ',', header = FALSE)
# evaluating model
output.tree data = rdata)
# plotting results
plot(output.tree)
```

Случайный лес

Для создания случайного леса необходимо импортировать пакет randomForest

```
# importing packages
install.packages("party")
install.packages("randomForest")
# reading data
rdata "input.csv", sep = ',', header = FALSE)
# creating the forest
output.forest data = rdata)
# getting results
print(output.forest)
```

Наивный Бейесовский классификатор

```
# importing package and it's dependencies
```

```
library(e1071)
# reading data
data "input.csv", sep = ',', header = FALSE)
# splitting data into training and test data sets
index y = data$target, p = 0.8, list = FALSE)
training # create objects x and y for predictor and response variables
x 9]
y target
# training model
model 'nb', trControl = trainControl(method = 'cv', number = 10))
# predicting results
predictions newdata = testing)
```

Метод опорных векторов

```
# importing package and its' dependencies
library(caret)
#reading data
data "input.csv", sep = ',', header = FALSE)

# splitting data into train and test sets
index y = data$target, p = 0.8, list = FALSE)
training # evaluating model
fit data = train_flats,
 method = "svmRadial",
 trControl = trainControl(method = "repeatedcv", number = 10, repeats = 3))
# printing parameters
print(fit)
```

Бустинг

```
# loading libraries
install.packages("mlr")
library(mlr)
# loading data
train "input.csv")
test "testInput.csv")
# loading GBM
getParamSet("classif.gbm")
baseLearner "classif.gbm", predict.type = "response")
# specifying parameters
controlFunction maxit = 50000) # specifying tuning method
cvFunction "CV", iters = 100000) # definig cross-validation function
gbmParameters"distribution", values = "bernoulli"),
 makeIntegerParam("n.trees", lower = 100, upper = 1000), # number of trees
 makeIntegerParam("interaction.depth", lower = 2, upper = 10), # depth of
tree
 makeIntegerParam("n.minobsinnode", lower = 10, upper = 80),
 makeNumericParam("shrinkage", lower = 0.01, upper = 1)
```

```
)
# tunning parameters
gbmTuningParameters learner = baseLearner,
 task = trainTask,
 resampling = cvFunction,
 measures = acc,
 par.set = gbmParameters,
 control = controlFunction)
# creating model parameters
model learner = baseLearner, par.vals = gbmTuningParameters)
# evaluating model
fit
```

Кластеризация

Для реализации алгоритма кластеризации k-средних используется пакет ClusterR. В нем реализовано 2 функции: KMeans_arma() и KMeans_rcpp(). В примере далее рассмотрена реализация с использованием функции KMeans_arma().

```
# importing package and its' dependencies
library(ClusterR)
# reading data
data "data.csv")
# evaluating model
model clusters = 2, n_iter = 10, seed_mode = "random_subset",
 verbose = T, CENTROIDS = NULL)
# predicting results
predictions
```

Примерная тематика НИРС по теме

1. Интеллектуальный анализ данных в системах поддержки принятия решений

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Омельченко, В. П. Медицинская информатика : учебник / В. П. Омельченко, А. А. Демидова. - Москва : ГЭОТАР-Медиа, 2016. - Текст : электронный.
2. Медицинская информатика : учебник / ред. Т. В. Зарубина, Б. А. Кобринский. - 2-е изд., перераб. и доп. - Москва : ГЭОТАР-Медиа, 2022. - 464 с. - Текст : электронный.
3. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.

4. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.
5. Информатика и медицинская статистика : учебное пособие / ред. Г. Н. Царик. - Москва : ГЭОТАР-Медиа, 2017. - 304 с. - Текст : электронный.
6. Медик, В. А. Математическая статистика в медицине : учебное пособие для вузов : в 2 т. / В. А. Медик, М. С. Токмачев. - 2-е изд., перераб. и доп. - Москва : Юрайт, 2021. - Т. 1. - 471 с. - Текст : электронный.

Электронные ресурсы

1. Визуализация данных с Python (https://habr.com/ru/company/ods/blog/323210/)
2. Ассоциативные правила силами Python 3 (https://www.youtube.com/watch?v=7cniTNfJUXU)

Практическое занятие №2

Тема: Первичный анализ больших данных.

Разновидность занятия: комбинированное.

Методы обучения: объяснительно-иллюстративный, репродуктивный, метод проблемного изложения, частично-поисковый, исследовательский.

Значение темы (актуальность изучаемой проблемы): одной из важнейших сфер, в которых активно адаптируются методы ИАД, является медицина.

Формируемые компетенции: ПК-10.1 ,ПК-10.2 ,ПК-10.3 ,ПК-10.4.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Организация занятия	5.00	Проверка посещаемости и внешнего вида обучающихся
2	Формулировка темы и целей	10.00	Озвучивание преподавателем темы и ее актуальности, целей занятия
3	Контроль исходного уровня знаний и умений	10.00	Тестирование, индивидуальный устный или письменный опрос, фронтальный опрос
4	Раскрытие учебно-целевых вопросов по теме занятия	10.00	Изложение основных положений темы
5	Самостоятельная работа обучающихся (текущий контроль)	40.00	Выполнение практического задания
6	Итоговый контроль знаний (письменно или устно)	10.00	Тесты по теме, ситуационные задачи
7	Задание на дом (на следующее занятие)	5.00	Учебно-методические разработки следующего занятия и методические разработки для внеаудиторной работы по теме
	ВСЕГО	90	

Аннотация (краткое содержание темы):

1. Библиотека / данные

Построение графиков, а также статистическая или интерактивная визуализация - это одни из важнейших задач анализа данных. Они могут быть частью процесса исследования, например, применяться при выявлении выбросов, определения необходимых преобразований данных или поиска идей для построения моделей. Как обычно нам понадобятся библиотеки pandas и NumPy. Дополнительно загрузим из библиотеки matplotlib модуль pyplot. Magic-команда matplotlib inline необходима для вывода графика на экран Jupiter Notebook.

Мы использовали параметр parse_dates метода read_csv() для корректной установки формата столбца. Затем благодаря методу set_index столбец был назначен индексом строк объекта DataFrame.

2. Настройка

Сама по себе библиотека pandas не выполняет визуализацию данных. Для выполнения этой задачи библиотека pandas предлагает тесную интеграцию с другими надежными библиотеками визуализации, которые является частью экосистемы Python. Наиболее часто используемой библиотекой для визуализации является библиотека matplotlib, поэтому в примерах, приведенных в этом Notebook будет использоваться библиотека matplotlib. Однако есть и другие возможные библиотеки, с которыми вы можете поработать самостоятельно.

2.1 пример

Мы рассмотрим настройку параметров построения графиков на примере визуализации временных рядов из объекта DataFrame. Визуализировать временной ряд в библиотеке pandas очень просто - достаточно применить метод plot() объекта DataFrame или Series, в который записан временной ряд.

Метод plot () объектов pandas является функцией-оберткой вокруг одноимённой функции библиотеки matplotlib. Данный метод делает построение графиков в pandas очень простой процедурой, поскольку программный код, лежащий в его основе, позволяет строить самые различные типы визуализации данных.

В этом примере метод plot() определил, что объект Series проиндексирован по датам, поэтому ось X должна быть представлена в виде дат. Кроме того, в методе plot() задан цвет, который будет использоваться по умолчанию для отображения данных. Если объект DataFrame состоит из нескольких столбцов, то метод plot() добавит несколько элементов в легенду и подберёт для каждой линии свой цвет.

Из полученного графика сложно выявить какие-то зависимости, так как масштаб рассматриваемых цен акций разный. Давайте проведем нормализацию значений: вычислим среднее значение и разделим на стандартное отклонение. Из графика видно, что значения цен акций высоко скоррелированы. Также можно определить, когда были минимумы или максимумы цен на акции.

2.2 размер

Встроенный метод plot() предлагает множество параметров, которые вы можете использовать для изменения содержимого графика. Давайте рассмотрим некоторые часто используемые настройки графиков. Размер графика регулируется параметром figsize. Передаём через данный параметр ширину и высоту в виде «питоновского» кортежа.

2.3 заголовок / подписи

Заголовок графика можно задать с помощью параметра title. Метки осей задаются с помощью функции xlabel и ylabel сразу после вызова метода plot().

2.4 легенда

Чтобы изменить текст, который используется в легенде для идентификации каждой серии данных, сохраним в переменную ax объект, который будет возвращён методом plot(), и затем воспользуемся методом legend(). Данный объект можно использовать для изменения различных настроек графика непосредственно перед его построением.

Местоположение легенды можно задать с помощью параметра loc метода legend(). По умолчанию библиотека pandas устанавливает данный параметр, равный 'best', что дает инструкцию библиотеке matplotlib исследовать график и определить наилучшее расположение легенды. Однако, вы можете указать любой вариант месторасположения: можно использовать как строковое значение, так и числовой код. Вывод легенды можно отключить с помощью значения параметра legend = False.

2.5 цвет / стиль / толщина / маркер

Библиотека pandas автоматически задает цвета для каждой серии при построении графика. Чтобы задать свои собственные цвета, просто передайте кодовое обозначение цветов параметру style. Библиотека matplotlib предлагает ряд встроенных односимвольных кодов, задающих различные цвета линий. Кроме того, можно задать цвет с помощью шестнадцатеричного RGB-кода.

Стили линий можно задать с помощью кода. Их можно использовать в сочетании с кодами цветов, задав сразу после кодового обозначения цвета.

Толщину линий можно задать с помощью параметра lw. Можно настроить толщину сразу нескольких строк, передав этому параметру список значений ширины или одно значение ширины, которое будет применено ко всем линиям.

Маркеры точек можно указать с помощью специальных обозначений в программном коде.

3. Графики

Теперь, когда вы знаете, как настраивать параметры графиков, мы рассмотрим построение различных диаграмм, которые могут пригодиться для визуализации статистической информации.

3.1 bar

Чтобы построить не линейный график, а столбчатую диаграмму, достаточно передать параметр kind в метод plot().

Для построения составной столбчатой диаграммы по объекту DataFrame нужно задать параметр stacked = True. Тогда столбики, соответствующие значениям в каждой строке, будут приставлены друг к другу. Вертикальную ориентацию диаграммы, которая используется по умолчанию, можно сменить на горизонтальную с помощью значение параметра kind = 'barh'.

3.2 hist

Гистограмма, с которой все мы хорошо знакомы, - это разновидность столбчатой диаграммы, показывающая дискретизированное представление частоты. Результаты измерений распределяются по дискретным интервалам равной ширины, а на гистограмме отображается количество точек в каждом интервале.

Параметр bins показывает количество интервалов, на которое делится непрерывная величина. по умолчанию данный параметр равен 10.

Есть ли в объекте DataFrame есть несколько числовых столбцов, то метод hist(), вызванный у этого объекта DataFrame автоматически сгенерирует несколько гистограмм: по одной для каждого столбца.

Чтобы наложить несколько гистограмм друг на друга в рамках одного и того же рисунка и тем самым визуализировать разницу распределений, несколько раз вызовем функцию plt.hist().

С гистограммой тесно связан график плотности, который строится на основе оценки непрерывного распределения вероятности по результатам измерений. Обычно стремятся аппроксимировать это распределение комбинацией ядер, то есть более простых распределений, например, нормального, Гаусса, поэтому графики плотности еще называют графиками ядерной оценки плотности (kernel density estimate – kde).

Функция plot с параметром kind = 'kde' строит график плотности, применяя стандартный метод комбинирования нормальных распределений.

Гистограмма в нормированном виде, показывающая дискретизированную плотность, и поверх неё график ядерной оценки плотности часто рисуются вместе.

3.3 box

Boxplot, часто называемый ящик с усами, - это график, который используется в описательной статистике, компактно изображающий одномерные статистики распределения переменный. Такой вид диаграммы в удобном формате показывает медиану, 25-процентный квантиль, 75-процентный квантиль (оба этих квантиля называют квартилями), минимальное значение и максимальное значение, а также выбросы. Расстояние между различными частями ящика позволяет определить степень разброса асимметрии данных и выявить выбросы.

3.4 scatter

Диаграмма рассеивания - это полезный способ исследования соотношений между двумя одномерными рядами данных. Диаграмму рассеивания можно создать на основе объекта DataFrame с помощью метода

plot(), при этом указав значения параметра kind = ‘scatter’, а также в столбце x и y исходного объекта DataFrame.

В разведочном анализе данных полезно видеть все диаграммы рассеивания для группы переменных. Это называется матрица диаграмм рассеивания. Построение такого графика с нуля - довольно утомительное занятие, поэтому в бибилиотеке pandas имеется функция scatter_matrix() для построения матрицы на основе объекта DataFrame. Она поддерживает также размещение гистограмм или графиков плотности для каждой переменной вдоль диагонали.

3.5 heat map

Heat map - это графическое представление данных, при котором значения внутри матрицы представлены цветами. Это эффективный инструмент, который позволяет визуализировать значения, получаемые на пересечении двух переменных.

Примерная тематика НИРС по теме

1. Основные концепции современного искусственного интеллекта.

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.
2. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.

Электронные ресурсы

1. Машинное обучение (курс лекций, К.В.Воронцов) (http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE)
2. Алгоритмы интеллектуального анализа данных (https://tproger.ru/translations/top-10-data-mining-algorithms/)
3. Классификация, регрессия и другие алгоритмы Data Mining с использованием R (https://ranalytics.github.io/data-mining/index.html)

Практическое занятие №3

Тема: Байесовский классификатор.

Разновидность занятия: комбинированное.

Методы обучения: объяснительно-иллюстративный, репродуктивный, метод проблемного изложения, частично-поисковый, исследовательский.

Значение темы (актуальность изучаемой проблемы): статистический байесовский подход является одним из старейших в теории классификации и лежит в основе многих методов обучения. Байесовский классификатор — широкий класс алгоритмов классификации, основанный на принципе максимума апостериорной вероятности. Для классифицируемого объекта вычисляются функции правдоподобия каждого из классов, по ним вычисляются апостериорные вероятности классов. Объект относится к тому классу, для которого апостериорная вероятность максимальна. В зависимости от точной природы вероятностной модели, наивные байесовские классификаторы могут обучаться очень эффективно. Во многих практических приложениях для оценки параметров для наивных байесовых моделей используют метод максимального правдоподобия; другими словами, можно работать с наивной байесовской моделью, не веря в байесовскую вероятность и не используя байесовские методы.

Формируемые компетенции: ПК-2.1.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Организация занятия	5.00	Проверка посещаемости и внешнего вида обучающихся
2	Формулировка темы и целей	10.00	Озвучивание преподавателем темы и ее актуальности, целей занятия
3	Контроль исходного уровня знаний и умений	10.00	Тестирование, индивидуальный устный или письменный опрос, фронтальный опрос
4	Раскрытие учебно-целевых вопросов по теме занятия	10.00	Изложение основных положений темы
5	Самостоятельная работа обучающихся (текущий контроль)	40.00	Выполнение практического задания

6	Итоговый контроль знаний (письменно или устно)	10.00	Тесты по теме, ситуационные задачи
7	Задание на дом (на следующее занятие)	5.00	Учебно-методические разработки следующего занятия и методические разработки для внеаудиторной работы по теме
	ВСЕГО	90	

Аннотация (краткое содержание темы):

В машинном обучении — семейство простых вероятностных классификаторов, основанных на использовании теоремы Байеса и «наивном» предположении о независимости признаков классифицируемых объектов.

Анализ на основе байесовской классификации активно изучался и использовался начиная с 1950-х годов в области классификации документов, где в качестве признаков использовались частоты слов. Алгоритм является масштабируемым по числу признаков, а по точности сопоставим с другими популярными методами, такими как машины опорных векторов.

Как и любой классификатор, байесовский присваивает метки классов наблюдениям, представленным векторами признаков. При этом предполагается, что каждый признак независимо влияет на вероятность принадлежности наблюдения к классу. Простой байесовский классификатор строится на основе обучения с учителем. Несмотря на малореалистичное предположение о независимости признаков, простые байесовские классификаторы хорошо зарекомендовали себя при решении многих практических задач. Дополнительным преимуществом метода является небольшое число примеров, необходимых для обучения.

Байесовский подход к классификации основан на теореме, утверждающей, что если плотности распределения каждого из классов известны, то искомый алгоритм можно выписать в явном аналитическом виде. Более того, этот алгоритм оптимален, то есть обладает минимальной вероятностью ошибок.

На практике плотности распределения классов, как правило, не известны. Их приходится оценивать (восстанавливать) по обучающей выборке. В результате байесовский алгоритм перестаёт быть оптимальным, так как восстановить плотность по выборке можно только с некоторой погрешностью. Чем короче выборка, тем выше шансы подогнать распределение под конкретные данные и столкнуться с эффектом переобучения.

Байесовский подход к классификации является одним из старейших, но до сих пор сохраняет прочные позиции в теории распознавания. Он лежит в основе многих достаточно удачных алгоритмов классификации.

К числу байесовских методов классификации относятся:

- Наивный байесовский классификатор
- Линейный дискриминант Фишера
- Квадратичный дискриминант
- Метод парзеновского окна
- Метод радиальных базисных функций (RBF)
- Логистическая регрессия

Так называемая наивная классификация или наивно-байесовский подход (naive-bayes approach) является наиболее простым вариантом метода, использующего байесовские сети. При этом подходе решаются задачи классификации, результатом работы метода являются так называемые "прозрачные" модели.

"Наивная" классификация - достаточно прозрачный и понятный метод классификации. "Наивной" она называется потому, что исходит из предположения о взаимной независимости признаков.

Свойства наивной классификации:

1. Использование всех переменных и определение всех зависимостей между ними.
2. Наличие двух предположений относительно переменных:
   - все переменные являются одинаково важными;
   - все переменные являются статистически независимыми, т.е. значение одной переменной ничего не говорит о значении другой.

Большинство других методов классификации предполагают, что перед началом классификации вероятность того, что объект принадлежит тому или иному классу, одинакова; но это не всегда верно.

Допустим, известно, что определенный процент данных принадлежит конкретному классу. Возникает вопрос, можем ли мы использовать эту информацию при построении модели классификации? Существует множество реальных примеров использования этих априорных знаний, помогающих классифицировать объекты. Типичный пример из медицинской практики. Если доктор отправляет результаты анализов пациента на дополнительное исследование, он относит пациента к какому-то определенному классу. Каким образом можно применить эту информацию? Мы можем использовать ее в качестве дополнительных данных при построении классификационной модели.

Отмечают такие достоинства байесовских сетей как метода Data Mining:

- в модели определяются зависимости между всеми переменными, это позволяет легко обрабатывать ситуации, в которых значения некоторых переменных неизвестны;
- байесовские сети достаточно просто интерпретируются и позволяют на этапе прогностического моделирования легко проводить анализ по сценарию "что, если";
- байесовский метод позволяет естественным образом совмещать закономерности, выведенные из данных, и, например, экспертные знания, полученные в явном виде;

- использование байесовских сетей позволяет избежать проблемы переучивания (overfitting), то есть избыточного усложнения модели, что является слабой стороной многих методов (например, деревьев решений и нейронных сетей).

Наивно-байесовский подход имеет следующие недостатки:

- перемножать условные вероятности корректно только тогда, когда все входные переменные действительно статистически независимы; хотя часто данный метод показывает достаточно хорошие результаты при несоблюдении условия статистической независимости, но теоретически такая ситуация должна обрабатываться более сложными методами, основанными на обучении байесовских сетей ;
- невозможна непосредственная обработка непрерывных переменных - требуется их преобразование к интервальной шкале, чтобы атрибуты были дискретными; однако такие преобразования иногда могут приводить к потере значимых закономерностей;
- на результат классификации в наивно-байесовском подходе влияют только индивидуальные значения входных переменных, комбинированное влияние пар или троек значений разных атрибутов здесь не учитывается. Это могло бы улучшить качество классификационной модели с точки зрения ее прогнозирующей точности, однако,увеличило бы количество проверяемых вариантов.

Байесовская классификация нашла широкое применение в задачах медицинской диагностики.

Примерная тематика НИРС по теме

1. Модель представления признаков в байесовском классификаторе медицинских изображений.

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.
2. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.

Электронные ресурсы

1. Машинное обучение (курс лекций, К.В.Воронцов) (http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE)
2. UCI Machine Learning Repository (репозиторий машинного обучения) (http://archive.ics.uci.edu/ml/index.php)
3. Алгоритмы интеллектуального анализа данных (https://tproger.ru/translations/top-10-data-mining-algorithms/)

4. Классификация, регрессия и другие алгоритмы Data Mining с использованием R (https://ranalytics.github.io/data-mining/index.html)
5. Наивный баейсовский классификатор (http://datascientist.one/naive-bayes/)

Практическое занятие №4

Тема: Метрические методы классификации.

Разновидность занятия: комбинированное.

Методы обучения: объяснительно-иллюстративный, репродуктивный, метод проблемного изложения, частично-поисковый, исследовательский.

Значение темы (актуальность изучаемой проблемы): Во многих прикладных задачах измерять степень сходства объектов существенно проще, чем формировать признаковые описания. Например, гораздо легче сравнить две фотографии и сказать, что они принадлежат одному человеку, чем понять, на основании каких признаков они схожи. Такие ситуации часто возникают при распознавании изображений, временных рядов или символьных последовательностей. Они характеризуются тем, что «сырые» исходные данные не годятся в качестве признаковых описаний, но в то же время, существуют эффективные и содержательно обоснованные способы оценить степень сходства любой пары «сырых» описаний. Есть ещё одна характерная особенность этих задач. Если мера сходства введена достаточно удачно, то оказывается, что схожим объектам, как правило, соответствуют схожие ответы. В задачах классификации это означает, что схожие объекты гораздо чаще лежат в одном классе, чем в разных. Если задача в принципе поддаётся решению, то граница между классами не может «проходить повсюду»; классы образуют компактно локализованные подмножества в пространстве объектов. Это предположение принято называть гипотезой компактности. Для формализации понятия «сходства» вводится функция расстояния или метрика $\rho(x, x')$ в пространстве объектов X. Алгоритмы, основанные на анализе сходства объектов, часто называют метрическими, даже в тех случаях, когда функция ρ не удовлетворяет всем аксиомам метрики (например, аксиоме треугольника).

Формируемые компетенции: ПК-10.1 ,ПК-10.2 ,ПК-10.3 ,ПК-10.4 ,ОПК-2.2.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Организация занятия	5.00	Проверка посещаемости и внешнего вида обучающихся
2	Формулировка темы и целей	10.00	Озвучивание преподавателем темы и ее актуальности, целей занятия
3	Контроль исходного уровня знаний и умений	10.00	Тестирование, индивидуальный устный или письменный опрос, фронтальный опрос

4	Раскрытие учебно-целевых вопросов по теме занятия	10.00	Изложение основных положений темы
5	Самостоятельная работа обучающихся (текущий контроль)	40.00	Выполнение практического задания
6	Итоговый контроль знаний (письменно или устно)	10.00	Тесты по теме, ситуационные задачи
7	Задание на дом (на следующее занятие)	5.00	Учебно-методические разработки следующего занятия и методические разработки для внеаудиторной работы по теме
	ВСЕГО	90	

Аннотация (краткое содержание темы):

Во многих прикладных задачах измерять степень сходства объектов существенно проще, чем формировать признаковые описания. Например, гораздо легче сравнить две фотографии и сказать, что они принадлежат одному человеку, чем понять, на основании каких признаков они схожи. Такие ситуации часто возникают при распознавании изображений, временных рядов или символьных последовательностей. Они характеризуются тем, что «сырые» исходные данные не годятся в качестве признаковых описаний, но в то же время, существуют эффективные и содержательно обоснованные способы оценить степень сходства любой пары «сырых» описаний.

Есть ещё одна характерная особенность этих задач. Если мера сходства введена достаточно удачно, то оказывается, что схожим объектам, как правило, соответствуют схожие ответы. В задачах классификации это означает, что схожие объекты гораздо чаще лежат в одном классе, чем в разных. Если задача в принципе поддаётся решению, то граница между классами не может «проходить повсюду»; классы образуют компактно локализованные подмножества в пространстве объектов. Это предположение принято называть гипотезой компактности. Для формализации понятия «сходства» вводится функция расстояния или метрика $\rho(x, x')$ в пространстве объектов X. Алгоритмы, основанные на анализе сходства объектов, часто называют метрическими, даже в тех случаях, когда функция ρ не удовлетворяет всем аксиомам метрики (например, аксиоме треугольника).

Рассматриваются различные метрические алгоритмы классификации, вводится важное понятие отступа объекта, которое используется в алгоритме отбора эталонных объектов. Основной вопрос — откуда берётся метрика, и как строить «хорошие» метрики в конкретных задачах. Выводятся оценки

обобщающей способности метрических алгоритмов, и на их основе строится ещё один алгоритм отбора эталонных объектов.

Метрический классификатор (similarity-based classifier) — алгоритм классификации, основанный на вычислении оценок сходства между объектами. Простейшим метрическим классификатором является метод ближайших соседей, в котором классифицируемый объект относится к тому классу, которому принадлежит большинство схожих с ним объектов.

Для формализации понятия сходства вводится функция расстояния между объектами. Как правило, жёсткого требования, чтобы эта функция была метрикой не предъявляется; в частности, неравенство треугольника вполне может и нарушаться.

К метрическим алгоритмам классификации относятся:

Метод ближайших соседей

Метод потенциальных функций

Метод радиальных базисных функций

Метод парзеновского окна

Метод дробящихся эталонов

Алгоритм вычисления оценок

Метрические классификаторы опираются на гипотезу компактности, которая предполагает, что схожие объекты чаще лежат в одном классе, чем в разных. Это означает, что граница между классами имеет достаточно простую форму, и классы образуют компактно локализованные области в пространстве объектов. Заметим, что в математическом анализе компактными называются ограниченные замкнутые множества. Гипотеза компактности не имеет ничего общего с этим понятием, и пониматься скорее в «бытовом» смысле слова.

В метрических алгоритмах классифицируемый объект может описываться не набором признаков, а непосредственно вектором расстояний до остальных объектов обучающей выборки. В таких случаях говорят также о беспризнаковом распознавании.

Например, сходство текстов, химических формул, аминокислотных последовательностей, и т.п. гораздо проще измерять непосредственно, чем переходя к признаковым описаниям.

В практических задачах классификации редко встречаются такие «идеальные случаи», когда заранее известна хорошая функция расстояния. Если объекты описываются числовыми векторами, часто берут евклидову метрику. Этот выбор, как правило, ничем не обоснован — просто это первое, что приходит в голову. При этом необходимо помнить, что все признаки должны быть измерены «в одном масштабе», лучше всего — отнормированы. В противном случае признак с наибольшими числовыми значениями будет доминировать в метрике, остальные признаки, фактически, учитываться не будут.

Однако и нормировка является весьма сомнительной эвристикой, так как остаётся вопрос: «неужели все признаки одинаково значимы и должны учитываться примерно с одинаковым весом?»

Если признаков слишком много, а расстояние вычисляется как сумма отклонений по отдельным признакам, то возникает проблема проклятия размерности. Суммы большого числа отклонений с большой вероятностью имеют очень близкие значения (согласно закону больших чисел). Получается, что в пространстве высокой размерности все объекты примерно одинаково далеки друг от друга; в частности, выбор k ближайших соседей становится практически случайным.

Проблема решается путём отбора относительно небольшого числа информативных признаков (features selection). В алгоритмах вычисления оценок строится множество различных наборов признаков (т.н. опорных множеств), для каждого строится своя функция близости, затем по всем функциям близости производится голосование.

Примерная тематика НИРС по теме

1. Сравнение метрических методов классификации.

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.
2. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.

Электронные ресурсы

1. Машинное обучение (курс лекций, К.В.Воронцов) (http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE)
2. Алгоритмы интеллектуального анализа данных (https://tproger.ru/translations/top-10-data-mining-algorithms/)
3. Классификация, регрессия и другие алгоритмы Data Mining с использованием R (https://ranalytics.github.io/data-mining/index.html)

Практическое занятие №5

Тема: Методы восстановления регрессии.

Разновидность занятия: комбинированное.

Методы обучения: объяснительно-иллюстративный, репродуктивный, метод проблемного изложения, частично-поисковый, исследовательский.

Значение темы (актуальность изучаемой проблемы): Регрессионный анализ — метод моделирования измеряемых данных и исследования их свойств. Данные состоят из пар значений зависимой переменной (переменной отклика) и независимой переменной (объясняющей переменной). Регрессионная модель есть функция независимой переменной и параметров с добавленной случайной переменной. Параметры модели настраиваются таким образом, что модель наилучшим образом приближает данные. Критерием качества приближения (целевой функцией) обычно является среднеквадратичная ошибка: сумма квадратов разности значений модели и зависимой переменной для всех значений независимой переменной в качестве аргумента. Регрессионный анализ — раздел математической статистики и машинного обучения. Предполагается, что зависимая переменная есть сумма значений некоторой модели и случайной величины. Относительно характера распределения этой величины делаются предположения, называемые гипотезой порождения данных. Для подтверждения или опровержения этой гипотезы выполняются статистические тесты, называемые анализом остатков. При этом предполагается, что независимая переменная не содержит ошибок. Регрессионный анализ используется для прогноза, анализа временных рядов, тестирования гипотез и выявления скрытых взаимосвязей в данных.

Формируемые компетенции: ПК-10.3 ,ОПК-2.2.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Организация занятия	5.00	Проверка посещаемости и внешнего вида обучающихся
2	Формулировка темы и целей	10.00	Озвучивание преподавателем темы и ее актуальности, целей занятия
3	Контроль исходного уровня знаний и умений	10.00	Тестирование, индивидуальный устный или письменный опрос, фронтальный опрос
4	Раскрытие учебно-	10.00	Изложение основных положений

	целевых вопросов по теме занятия		темы
5	Самостоятельная работа обучающихся (текущий контроль)	40.00	Выполнение практического задания
6	Итоговый контроль знаний (письменно или устно)	10.00	Тесты по теме, ситуационные задачи
7	Задание на дом (на следующее занятие)	5.00	Учебно-методические разработки следующего занятия и методические разработки для внеаудиторной работы по теме
	ВСЕГО	90	

Аннотация (краткое содержание темы):

Регрессия — зависимость математического ожидания (например, среднего значения) случайной величины от одной или нескольких других случайных величин (свободных переменных). Регрессионным анализом называется поиск такой функции f, которая описывает эту зависимость.

Регрессия как правило, выполняется с помощью обучения с учителем на этапе тестирования, является частным случаем задач прогнозирования.

Задачу обучения по прецедентам при Y = R принято называть задачей восстановления регрессии. Задано пространство объектов X и множество возможных ответов Y. Существует неизвестная целевая зависимость $y*: X \to Y$, значения которой известны только на объектах обучающей выборки $X\ell = (xi , yi)$, $yi = y*(xi)$. Требуется построить алгоритм $a: X \to Y$, аппроксимирующий целевую зависимость $y*$.

Линейная регрессия предполагает, что функция f зависит от параметров w линейно. При этом линейная зависимость от свободной переменной x необязательна. Значения параметров в случае линейной регрессии находят с помощью метода наименьших квадратов. Использование этого метода обосновано предположением о гауссовском распределении случайной переменной.

Метод наименьших квадратов — метод нахождения оптимальных параметров линейной регрессии, таких, что сумма квадратов ошибок (регрессионных остатков) минимальна. Метод заключается в минимизации евклидова расстояния между двумя векторами — вектором восстановленных значений зависимой переменной и вектором фактических значений зависимой переменной.

Линейная регрессия — метод восстановления зависимости между двумя переменными.

Метод главных компонент.

Задача анализа главных компонент имеет, как минимум, четыре базовых версии:

1. аппроксимировать данные линейными многообразиями меньшей размерности;
2. найти подпространства меньшей размерности, в ортогональной проекции на которые разброс данных (то есть среднеквадратичное отклонение от среднего значения) максимален;
3. найти подпространства меньшей размерности, в ортогональной проекции на которые среднеквадратичное расстояние между точками максимально;
4. для данной многомерной случайной величины построить такое ортогональное преобразование координат, в результате которого корреляции между отдельными координатами обратятся в нуль.

Первые три версии оперируют конечными множествами данных. Они эквивалентны и не используют никакой гипотезы о статистическом порождении данных. Четвёртая версия оперирует случайными величинами. Конечные множества появляются здесь как выборки из данного распределения, а решение трёх первых задач — как приближение к разложению по теореме Кархунена — Лоэва («истинному преобразованию Кархунена — Лоэва»). При этом возникает дополнительный и не вполне тривиальный вопрос о точности этого приближения.

Примерная тематика НИРС по теме

1. Логистическая регрессия для решения задач классификации.

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.
2. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.

Электронные ресурсы

1. Алгоритмы интеллектуального анализа данных (https://tproger.ru/translations/top-10-data-mining-algorithms/)
2. Классификация, регрессия и другие алгоритмы Data Mining с использованием R (https://ranalytics.github.io/data-mining/index.html)

Практическое занятие №6

Тема: Сокращение размерности признакового пространства. Кластеризация и визуализация.

Разновидность занятия: комбинированное.

Методы обучения: объяснительно-иллюстративный, репродуктивный, метод проблемного изложения, частично-поисковый, исследовательский.

Значение темы (актуальность изучаемой проблемы): При решении задач анализа часто приходится сталкиваться с двумя полярными проблемами — избыточностью и недостаточностью данных. Следует отметить, что избыточность и недостаточность далеко не всегда связаны с количеством имеющихся данных. Вполне возможно, что объем исходных данных велик, но для решения конкретной задачи анализа их все равно недостаточно из-за отсутствия в них необходимой информации. Возможен и прямо противоположный случай, когда объем исходных данных небольшой, но они подобраны настолько удачно, что обеспечивают высокую эффективность анализа. Таким образом, информативность данных не всегда пропорциональна их объему. Тем не менее многие аналитики считают, и практика это показывает, что лучше иметь данные с запасом, то есть собирать для анализа как можно больше данных, описывающих предметную область. Действительно, в больших массивах данных есть вероятность «наскрести» достаточное количество информации для решения той или иной задачи. Кроме того, технический уровень современных носителей информации, которые могут хранить сотни гигабайт данных, снимает проблему размещения данных. Когда дело доходит до анализа, встает проблема: из всего имеющегося множества данных, которое может содержать десятки признаков, атрибутов и показателей, описывающих анализируемый процесс или объект, необходимо выбрать подмножество, которое обеспечит оптимальные результаты анализа. Целью анализа является поиск закономерностей и структур в данных, поэтому перед его проведением следует выделить только те данные, которые могут описывать эти закономерности и структуры. Это довольно трудная и неочевидная задача.

Формируемые компетенции: ПК-10.1 ,ПК-10.2 ,ПК-10.3 ,ПК-10.4 ,ОПК-2.2.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Организация занятия	5.00	Проверка посещаемости и внешнего вида обучающихся
2	Формулировка темы и целей	10.00	Озвучивание преподавателем темы и ее актуальности, целей

			занятия
3	Контроль исходного уровня знаний и умений	10.00	Тестирование, индивидуальный устный или письменный опрос, фронтальный опрос
4	Раскрытие учебно-целевых вопросов по теме занятия	10.00	Изложение основных положений темы
5	Самостоятельная работа обучающихся (текущий контроль)	40.00	Выполнение практического задания
6	Итоговый контроль знаний (письменно или устно)	10.00	Тесты по теме, ситуационные задачи
7	Задание на дом (на следующее занятие)	5.00	Учебно-методические разработки следующего занятия и методические разработки для внеаудиторной работы по теме
	ВСЕГО	90	

Аннотация (краткое содержание темы):

В машинном обучении и Data Mining большинство моделей работает по принципу обобщения. На вход модели подается некоторый набор входных (независимых) переменных, связанных с какими- либо признаками, атрибутами или показателями, описывающими исследуемый объект или процесс. Модель производит над входными данными преобразование, определяемое некоторой целевой функцией, и формирует на выходе набор выходных (зависимых) переменных. При этом для успешного решения задачи чаще всего необходимо, чтобы число входных переменных было больше либо равно числу выходных, в чем и состоит принцип обобщения. Обычно для классификации реального объекта недостаточно знать один его признак — требуется исчерпывающее описание. То есть чтобы получить на выходе модели только одно значение — класс, к которому относится объект, — на ее вход нужно подать несколько признаков и атрибутов объекта. Конечно, в отдельных случаях удается классифицировать объекты по одному признаку, но это очень простые объекты, задача их распознавания тривиальна и не представляет практического интереса. Большинство реальных объектов и бизнес-процессов, которых нужно проанализировать, достаточно сложны, и для их исчерпывающего описания требуется множество признаков и наблюдений (записей). Поэтому размерность входного вектора может оказаться высокой — до нескольких десятков и сотен признаков. Это вызывает ряд серьезных проблем при анализе данных:
§ рост вычислительных затрат и времени, требуемого на обработку данных,

до совершенно неприемлемых значений; § сложность построения модели, трудность понимания ее пользователем; § сложность интерпретации результатов анализа и оценки их достоверности; § снижение качества результатов анализа. В исходном наборе могут содержаться данные, не связанные с исследуемым процессом или объектом. Если такие данные не будут исключены перед анализом, то они могут увести в сторону решение задачи. Чтобы исключить эти проблемы, в процессе предобработки данных в аналитическом приложении выполняется снижение их размерности (data reduction).

Снижение размерности входных данных — процесс сокращения объема исходного множества, загруженного для анализа в аналитическое приложение, таким образом, чтобы результирующее множество имело оптимальную размерность с точки зрения решаемой задачи и используемой модели.

Исходные множества данных представляются в аналитическом приложении в виде «плоских» таблиц, поэтому, когда говорят о снижении размерности данных, подразумевают сокращение данных по таким трем направлениям, как: § сокращение количества признаков (столбцов таблицы); § сокращение числа наблюдений (записей таблицы); § сокращение числа значений измерения (при этом само количество значений не меняется, уменьшается только число различных вариаций значения). Сокращение данных может производиться в двух режимах. 1 Режим отбора. Определяется значимость каждого признака исходного множества для решения конкретной задачи. Затем признаки отбираются в порядке уменьшения их значимости. Как только попадается признак, значимость которого меньше некоторого порога, отбор прекращается. Порог значимости устанавливается или на основе статистического анализа исходного множества, или опытным путем.

Режим исключения. Размер исходной выборки сокращается путем отбрасывания незначащих и избыточных данных. Например, для каждого признака исходной выборки определяется коэффициент значимости, а затем исключаются все признаки, значимость которых ниже некоторого порога. Возможен и другой вариант. По мере исключения признаков результирующая выборка становится все менее похожа на исходную. Задается условие, что выборка, полученная в результате сокращения исходной, не должна отличаться от нее более чем на 70 %. Затем все признаки исходной выборки ранжируются по уровню их значимости, и начинается процесс исключения наименее значимых. Он будет продолжаться до тех пор, пока отличие исходной выборки от сокращенной не превысит допустимое значение.

Требования к алгоритмам снижения размерности данных В настоящее время используется большое количество различных подходов к сокращению размерности данных. Одни из этих подходов относятся к эвристическим, другие основаны на строгом статистическом анализе данных, третьи сочетают в себе и то и другое. Тем не менее можно выделить ряд общих требований ко всем алгоритмам снижения размерности данных. §

Подмножество данных, образованное в результате сокращения размерности исходного множества, должно унаследовать от него столько информации, сколько необходимо для получения решения с заданной точностью. § Вычислительные и временные затраты на обработку данных с целью сокращения их размерности не должны обесценивать преимущества, полученные в результате сокращения размерности. § Модель, полученная на основе множества данных со сниженной размерностью, должна быть проще для разработки, реализации и понимания, чем модель, построенная на исходном множестве. § Признаки, оставшиеся после процедуры сокращения размерности, должны иметь высокий уровень значимости для решения задачи и не должны быть коррелированны между собой, а также содержать закономерности, которые могут увести аналитический процесс в сторону от правильных результатов. В идеальном случае в результате сокращения размерности данных удается уменьшить время анализа, повысить его точность и упростить модель одновременно. Однако чаще всего приходится искать баланс между этими преимуществами. Не существует метода снижения размерности, который одинаково хорошо работал бы для всех задач анализа и видов исходных данных. Решение о выборе метода основывается на априорном знании о поставленной задаче и ожидаемых результатах с учетом временных ограничений. Многие алгоритмы, реализующие снижение размерности данных, требуют значительных вычислительных и временных затрат, особенно когда они применяются к большим наборам данных. Следовательно, перед тем как использовать тот или иной алгоритм, необходимо иметь представления о его свойствах и возможностях. Выделим следующие рекомендуемые характеристики методов снижения размерности данных, которые могут применяться как при выборе алгоритмов, так и при разработке стратегии сокращения размерности данных в целом.

Алгоритм должен предоставлять пользователю возможность контролировать ход процесса, а также оценивать влияние сокращения признаков, записей и отдельных значений на ожидаемые результаты анализа. При большом объеме данных процедура их обработки с целью сокращения размерности может занять довольно продолжительное время, поэтому пользователь должен получать информацию о результатах работы, чтобы при необходимости остановить алгоритм, не дожидаясь его завершения. § Наибольшая эффективность должна достигаться на первых итерациях алгоритма и со временем уменьшаться (то есть сначала должны отбрасываться наименее значимые данные или отбираться наиболее значимые). Это позволит не ждать окончания работы алгоритма, а остановить его, как только будут достигнуты приемлемые результаты. § Алгоритм должен обеспечивать возможность приостановки в любой момент времени с сохранением промежуточных результатов, их оценки, а также возобновления работы. Таким образом, сокращение размерности — одна из важнейших задач подготовки исходных данных к анализу.

Примерная тематика НИРС по теме

1. Алгоритм AdaBoost
2. Метод главных компонент

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.
2. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.

Электронные ресурсы

1. Машинное обучение (курс лекций, К.В.Воронцов) (http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE)
2. Алгоритмы интеллектуального анализа данных (https://tproger.ru/translations/top-10-data-mining-algorithms/)
3. Классификация, регрессия и другие алгоритмы Data Mining с использованием R (https://ranalytics.github.io/data-mining/index.html)

Практическое занятие №7

Тема: Применение методов машинного обучения для анализа медицинских изображений и сигналов.

Разновидность занятия: комбинированное.

Методы обучения: объяснительно-иллюстративный, репродуктивный, метод проблемного изложения, частично-поисковый, исследовательский.

Значение темы (актуальность изучаемой проблемы): анализ биомедицинских изображений — актуальная тема, связанная в первую очередь с компьютерной диагностикой.

Формируемые компетенции: ПК-10.1 ,ОПК-2.2.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Организация занятия	5.00	Проверка посещаемости и внешнего вида обучающихся
2	Формулировка темы и целей	10.00	Озвучивание преподавателем темы и ее актуальности, целей занятия
3	Контроль исходного уровня знаний и умений	10.00	Тестирование, индивидуальный устный или письменный опрос, фронтальный опрос
4	Раскрытие учебно-целевых вопросов по теме занятия	10.00	Изложение основных положений темы
5	Самостоятельная работа обучающихся (текущий контроль)	40.00	Выполнение практического задания
6	Итоговый контроль знаний (письменно или устно)	10.00	Тесты по теме, ситуационные задачи
7	Задание на дом (на следующее занятие)	5.00	Учебно-методические разработки следующего занятия и методические разработки для внеаудиторной работы по теме
	ВСЕГО	90	

Аннотация (краткое содержание темы):

История вопроса и проблемы анализа медицинских снимков

Как только появились первые компьютеры, которые могли как-то работать с изображениями, то есть в 80-е годы XX века, появилась и мысль о том, что с их помощью можно автоматически анализировать медицинские снимки. Работой с медицинскими изображениями занимались представители разных областей науки. Но так как первые компьютеры работали очень медленно и маленькую картинку могли открывать минуту, а то и дольше, то полноценно использоваться в этой области они пока не могли. Компьютерам не хватало производительности, на них нельзя было обрабатывать большие массивы данных, они работали очень медленно.

Другое направление в обработке медицинских изображений связано с производством специальной медицинской техники. Создатели этих устройств используют в своих приборах довольно мощные программы для обработки данных. И проблема анализа медицинских изображений связана с тем, что получить сырые данные практически невозможно. Медицинский аппарат уже обработал их, и то, что мы в итоге видим на экране, — это результат какой-то фильтрации, улучшений и так далее. В результате пережатия картинки через тот же jpeg может потеряться много информации. Поэтому для полноценного компьютерного анализа нужны сырые, необработанные другими программами данные.

Области применения компьютерного анализа медицинских снимков

Компьютерный анализ медицинских изображений применим буквально во всех областях — от офтальмологии до МРТ. Особенно популярно сейчас изучение снимков глазного дна, так как это единственное место, где можно увидеть сосуды неинвазивно. Кроме того, первый признак диабета — диабетическую ретинопатию — можно обнаружить как раз на глазном дне. В эти исследования сейчас очень активно вкладываются на Западе, так как проблема диабета становится все насущнее.

Очень широкий круг задач в анализе медицинских изображений связан с дерматологией. Вообще, исследуются все области, где есть изображения. Каждая болезнь — отдельная область исследований. Поэтому из обилия болезней и различных медицинских аппаратов складывается огромное разнообразие в этих исследованиях. Для некоторых анализов, особенно для ультразвука и ряда других технологий (МРТ, компьютерной томографии), очень важно, на каком приборе это делается — алгоритмы пишутся для каждой конкретной модели и ее режимов.

С помощью анализа биомедицинских изображений можно исследовать что угодно — от перелома шейки бедра (чем мы занимались с сингапурскими коллегами) до 3D-моделирования зубов. Со стоматологией особенно интересно: при помощи моделирования пациент может увидеть, какие у него будут зубы, как они будут меняться от недели к неделе, если поставить брекеты, — эти процессы можно смоделировать.

Есть равные возможности для исследования всех областей медицины, и нет какой-то одной, в которой работы велись бы наиболее активно. Но наиболее приоритетным направлением все-таки является лучевая

диагностика мозга. В России этим занимаются не очень активно, особенно для нейродегенеративных заболеваний, разве что диагностикой болезни Паркинсона, тогда как на Западе очень большое внимание уделяется болезни Альцгеймера.

Особенности анализа медицинских изображений

Специфика обработки и анализа медицинских изображений в первую очередь связана с необходимостью работать с медиками. Но медиков-исследователей в России практически нет, есть только редкие исключения, например РНЦХ имени Петровского. Все наши учреждения в первую очередь ориентированы на лечение больных, а не на исследования. Поэтому врачи очень загружены. Они готовы встретиться со специалистами по анализу данных, дать какие-то данные. Но если необходимо провести исследования на дорогой и сложной аппаратуре, то это вступает в противоречие с практикой, необходимостью оплаты таких исследований и использования этих аппаратов и так далее. В некоторых областях, например в офтальмологии, с этим иногда проще.

Ведутся исследования с использованием ультразвука, цветной допплерографии — изучением потоков крови. Задач очень много, и все они интересные. Каждое отдельное исследование позволяет проводить исследования, публиковаться в приличных журналах. В отдельных темах анализа данных, например в методах компрессии данных, сейчас огромная конкуренция, уже сложно придумать что-то новое. А в медицине много узких областей, которые можно исследовать при содействии врачей и делать хорошие совместные работы.

Поэтому сложно ответить на вопрос, какая область медицины наиболее компьютеризирована. Но самые дорогие — это вещи, связанные с МРТ и КТ. Отдельная область — хирургия, то есть аппараты, которые проводят операции.

Нейронные сети и медицинские снимки

В последнее десятилетие произошел прорыв в распознавании изображений, и с тех пор в моду прочно вошли нейронные сети. Но в анализе биомедицинских изображений их роль неоднозначна. Связано это с несколькими причинами.

Во-первых, на мой взгляд, стремительно развивающиеся технологии искусственного интеллекта, частью которых являются искусственные нейронные сети, все же могут представлять угрозу для людей. Мы не до конца понимаем, что происходит в этом черном ящике, соответственно, нам сложно это контролировать и предсказывать поведение этих программ. Технологии позволяют имитировать голос, мимику, движение губ — это было продемонстрировано, например, на конференции SIGGRAPH 2017, где один из докладчиков взял видеовыступление Барака Обамы, с помощью обученной нейронной сети заменил ему текст и подстроил под новый текст мимику лица. Эти технологии дают большие возможности, но не всегда могут быть использованы на благо. Вам дали для исследований или работы нейронную сеть, но вы даже не знаете, на чем ее обучали, есть ли там так

называемые закладки — недокументированные возможности. Или, может, она обучена так, что в определенных случаях, если она работает с банковскими данными, сеть будет переводить деньги на какой-то другой счет или передаст важные данные на сторону.

В науке сейчас происходит резкий слом — смена парадигмы. И если раньше такие изменения проходили так, что ученые так или иначе могли проверить, что происходит, то сейчас мы видим, что те же сверточные нейронные сети дают хорошие практические результаты, но они все еще остаются для нас черным ящиком. И, возвращаясь к медицине, мы имеем очень интересную ситуацию. Медицинские данные во многих случаях нельзя использовать и публиковать, даже если не указана фамилия пациента. В некоторых странах, например в Тайване, есть специальные комиссии по этике, которые выдают специальные разрешения на использование даже анонимных чужих медицинских данных. В некоторых областях — офтальмологии, ретинопатии — есть стандартные базы данных, на которых можно что-то проверить и понять. В большинстве других случаев исследователи откуда-то берут какие-то свои данные, что-то с ними делают при помощи нейронной сети и потом объявляют, что у них точность результатов, например, 90%, и это выше, чем у профессиональных врачей. И закономерно возникает вопрос: как это проверить? Ответ: никак. Раньше при написании статьи автор указывал, какие методы он использовал, и можно было понять, за счет чего ему удалось улучшить результаты. А при описании результатов, полученных при помощи нейронных сетей, данные предоставляются крайне редко, и поэтому их происхождение и причины получения того или иного результата вызывают вопросы. Соответственно, самостоятельно проверить это исследование тоже нельзя, как и понять, за счет чего получился хороший результат. На каких данных тренировали сеть? На каких проверяли ее? Достоверность чаще всего проверить невозможно.

Нейронные сети в научном сообществе

Конечно, мы пользуемся нейронными сетями в анализе биомедицинских изображений. Но задача контроля анализа данных, полученных с помощью нейронных сетей, остается сложной и важной. В медицине это особенно важно, так как отсутствие прозрачности в анализе данных в итоге может привести к фатальной ошибке. В научном сообществе такая осторожность может привести к странным результатам. Допустим, написал статью про обработку медицинских изображений классическими методами, отправляешь в журнал или на конференцию, а рецензенты спрашивают: почему результаты не сравнивались с нейронными сетями? Начиная с 2012 года почти в любой области науки можно найти статьи авторов (чаще всего китайских), которые пишут о том, как им удалось добиться 99% результата при помощи нейронных сетей. И, что ни делай, самому такого результата добиться не получается, и с китайскими учеными очень тяжело конкурировать. Хотя на каких данных они обучали и как этого добились — остается непонятным. 99% точности — это высокий результат,

существенно больше того, который удается получить даже профессионалам-врачам.

Преимущество нейросетей в том, что это, безусловно, очень хороший аппарат для технологических решений. Это действительно прорыв в науке, с помощью которого можно решать очень продвинутые задачи, в том числе и в диагностике. Когда обучаешь сеть сам, все предельно ясно. То же и в медицинских изображениях: знаешь, на каких данных обучена сеть и чего от нее ожидать. А все, что мы получаем от других авторов, проверить очень тяжело. Крайне редко что-то делается на общей базе данных.

На сегодняшний день нейронные сети используются вместе с другими методами и в будущем в принципе едва ли их вытеснят. Это полезная вещь, но она никогда не заменит исследований, связанных с математическими методами. Собственно развитием самой технологии нейронных сетей занимается относительно малое число специалистов именно в этой области, а остальные — лишь пользователи, которые берут чужие сети и дообучают их. И поскольку данных в медицине мало, обычно берется какая-то общая нейронная сеть и дообучается на конкретных данных. Мы применяем нейронные сети в области офтальмологии и диагностики болезни Альцгеймера.

С нейронными сетями, помимо прочего, связана еще вот какая опасность: допустим, у нас есть картинки, сеть на них хорошо работает, но если на них добавить небольшой шум, особенно если он специально сгенерирован, незаметный для человеческого глаза, но чувствительный для нейросети, то сеть будет давать неправильные результаты.

Другие методы анализа

Чисто математических методов, которые используются в анализе изображений, в том числе биомедицинских, очень много. Они как раз и развивались с 80-х годов прошлого века. В них обработка и анализ изображений базируются на теории обработки сигналов, которая развивалась с середины XX века.

Что касается численных методов, то тут есть специфика, связанная именно с медицинским оборудованием. В медицине есть огромное количество аппаратов, каждый из которых имеет свою физику, свои особенности настройки, разные параметры и частотные области. Некоторые вещи мы не можем измерить. Например, в МРТ картинку с данных мы получаем при помощи обратного преобразования Фурье, и она может обладать массой различных дефектов, связанных именно с физикой — с потерей частотной информации. Поэтому методы используются очень разные. И очень сложно выделить область математики, которой тут не нашлось бы применения. Используется буквально все — от кватернионов и топологии до теории графов и статистических методов. И конечно, очень много методов, связанных именно с физикой аппаратов. Разумеется, используются методы искусственного интеллекта, машинное обучение, сверточные нейронные сети и так далее.

Эффективность анализа биомедицинских изображений

В разных областях медицины эффективность компьютерного анализа оценивается по-разному. Есть общие базы данных, например, по глазам и ретинопатии, где ситуация очень хорошая, процент точности там очень большой — выше 95%. В других областях понять результативность сложнее. Очень много зависит от того, какие настройки поставил доктор, поскольку, например, в УЗИ шум неаддитивный. Это означает, что результаты, полученные в разных режимах, сложно свести к одному показателю, как-то нормализовать и получить возможность сравнивать их.

В целом точность компьютерной диагностики чуть-чуть хуже, чем получается у самых профессиональных докторов, но лучше, чем у средних врачей. Но проблема в том, что часто одно заболевание влечет за собой еще несколько других, и это значительно усложняет диагностику.

В обработке и анализе медицинских изображений идеальная ситуация следующая: есть конкретное заболевание, есть аппаратура для диагностики определенной медицинской модальности (например, УЗИ). Мы изучаем наборы видеоданных из имеющейся базы данных пациентов и находим с помощью машинного обучения какие-то значимые параметры, соответствующие именно этому заболеванию. Кроме того, у нас есть база данных не только снимков одной медицинской модальности, но и комплексные истории болезни, данные анализа крови и так далее, — одним словом, полная картина. Для нового пациента по его медицинским видеоданным мы рассчитываем параметры, которые мы определили как значимые для рассматриваемого заболевания, и врачу программа дает не диагноз, а несколько, например пять, наиболее похожих по значимым признакам изображений пациентов. В этом случае врач смотрит полные истории болезни этих пяти пациентов и видит различные возможные варианты диагноза. При этом врачам, даже не самым высокопрофессиональным, будет гораздо легче работать с этой информацией, а вероятность ошибок сократится. В случае же выдачи программой диагноза (что эквивалентно выдачи истории болезни только одного пациента с похожими видеоданными используемой медицинской модальности) врачи недостаточно высокой квалификации просто с ней согласятся, иногда и не подозревая о возможных альтернативных диагнозах.

В перспективе создание программ диагностики должно быть рассчитано именно на комплексную диагностику. Сейчас эффективность анализа и диагностики зависит не от изображения, а от той болезни, которую мы пытаемся выявить. Допустим, диабетическую ретинопатию диагностировать получается хорошо, а вот с глаукомой все не так очевидно.

Примерная тематика НИРС по теме

1. Нейронные сети для медицинского диагностирования

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.
2. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.

Электронные ресурсы

1. Машинное обучение (курс лекций, К.В.Воронцов) (http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE)
2. Открытый курс машинного обучения. Тема 3. Классификация, деревья решений и метод ближайших соседей (https://habr.com/ru/company/ods/blog/322534/)

Практическое занятие №8

Тема: Глубокое обучение и нейросети.

Разновидность занятия: комбинированное.

Методы обучения: объяснительно-иллюстративный, репродуктивный, метод проблемного изложения, частично-поисковый, исследовательский.

Значение темы (актуальность изучаемой проблемы): на данном занятии рассматривается применение сверточных нейронных сетей для анализа медицинских изображений.

Формируемые компетенции: ПК-10.1 ,ПК-10.2 ,ПК-10.3 ,ПК-10.4 ,ОПК-2.2.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Организация занятия	5.00	Проверка посещаемости и внешнего вида обучающихся
2	Формулировка темы и целей	10.00	Озвучивание преподавателем темы и ее актуальности, целей занятия
3	Контроль исходного уровня знаний и умений	10.00	Тестирование, индивидуальный устный или письменный опрос, фронтальный опрос
4	Раскрытие учебно-целевых вопросов по теме занятия	10.00	Изложение основных положений темы
5	Самостоятельная работа обучающихся (текущий контроль)	40.00	Выполнение практического задания
6	Итоговый контроль знаний (письменно или устно)	10.00	Тесты по теме, ситуационные задачи
7	Задание на дом (на следующее занятие)	5.00	Учебно-методические разработки следующего занятия и методические разработки для внеаудиторной работы по теме
	ВСЕГО	90	

Аннотация (краткое содержание темы):

Методы машинного обучения позволяют успешно решать проблемы распознавания образов. Прежде всего, в этом помогали созданные вручную признаки и традиционные алгоритмы машинного обучения.

Созданные вручную признаки — это параметры изображений, извлекаемые с помощью множества алгоритмов (например HoG, LBP и др.). Однако отметим, что извлечение признаков вручную требуют много усилий и серьёзного уровня предметных знаний. На помощь в автоматизации этого процесса пришли свёрточные нейронные сети, которые сами учатся выделять характеристические (полезные) признаки.

Свёрточные нейросети преимущественно используются на практике для обработки медицинских изображений (МРТ, КТ и др.) и видео.

Рассмотрим принцип работы свёрточных нейронных сетей.

Нейрон на вход получает ограниченное количество пикселей, как правило, это участки изображения, соответствующие размеру ядра (например, 3x3; 5x5 и др.). Следующий нейрон работает со следующим участком изображения, который обычно пересекается с областью соседнего нейрона. Операция, которая выполняется при этом, называется сверткой.

Двумерная свертка — это перемножение матрицы изображения и ядра, представляющего из себя матрицу весов, путём прохождения через длину и ширину.

Ядро “скользит” над двумерным изображением, поэлементно выполняя операцию умножения с той частью входных данных, над которой оно сейчас находится, и затем суммирует все полученные значения в один выходной пиксель. Ядро повторяет эту процедуру с каждой локацией, над которой оно “скользит”, преобразуя двумерную матрицу в другую все еще двумерную матрицу признаков.

Перемножение матриц по сути является основой выделения признаков.

Применяя свёртку с одним ядром, получаем карту признаков. Обычно в свёрточных нейронных сетях бывает несколько ядер свёртки, которые определяются автоматически в процессе обучения. Сначала извлекаются низкоуровневые признаки (например, линии, круги), далее среднеуровневые (например, комбинации из нескольких низкоуровневых признаков) и, наконец, высокоуровневые признаки (например, рот, глаза, в примере распознавания лица человека).

Следующий принцип, используемый в свёрточных нейронных сетях, является уменьшение размерности. С этой целью используется слой пулинг (субдискритизирующий слой — pooling layer). Слой pooling помогает сократить пространственное представление изображения, чтобы уменьшить количество параметров и объём вычислений. Ниже приведен пример pooling layer: он проводится с использованием ядра размером 2x2 на выходе свёртки (другой матрицы) размером 5x5.

Используя комбинацию слоёв свёртки и pooling (например, с определением максимального значения, где выбирается из всего окна пиксель с наибольшей интенсивностью —max-pooling), мы получаем основной структурный компонент свёрточной нейронной сети. Данный

структурный компонент в большинстве современных архитектур повторяется несколько раз.

Нередко в свёрточных нейронных сетях для того, чтобы избежать потерь информации на границах изображения, применяют padding- слой. Padding добавляет к краям поддельные (fake) пиксели (обычно нулевого значения — zero padding). Таким образом, ядро при проскальзывании позволяет неподдельным пикселям оказываться в своем центре.

Иерархическая организация слоёв в свёрточной нейронной сети способствует последовательному извлечению признаков: от простых элементов к более сложным признакам. В конце слои уплощаются и связываются с выходным слоем функцией-активации (как и в нейронных сетях прямого распространения).

Принцип работы свёрточных нейронных сетей с несколькими фильтрами

Рассмотрим теперь, например, цветное изображение (3 канала — RGB: Red, Green, Blue). Отметим, что в случае с 1 каналом нередко термин «ядро свёртки» заменяют термином «фильтр». Однако, каждый фильтр представляет собой несколько ядер, причем для каждого отдельного входного канала слоя есть одно уникальное ядро.

Каждый фильтр в сверточном слое создает только один выходной канал, таким образом, каждое из ядер фильтра «скользит» по их соответствующим входным каналам, создавая обработанную версию каждого из них. Далее каждая из обработанных в канале версий суммируется вместе для формирования одного канала. Ядра каждого фильтра генерируют одну версию каждого канала, а фильтр в целом создает один общий выходной канал.

Примерная тематика НИРС по теме

1. Задачи анализа изображений.
2. Сверточные нейронные сети.
3. Компьютерное зрение.

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.
2. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.

Электронные ресурсы

1. Сверточная нейронная сеть, часть 2: обучение алгоритмом обратного распространения ошибки (https://habr.com/ru/post/348028/)

2. Лекции Техносферы. Нейронные сети в машинном обучении (https://habr.com/ru/company/mailru/blog/344982/)
3. Лекция 3. Обучение нейронных сетей в Keras (https://compscicenter.ru/courses/data-mining-python2/2018-autumn/classes/3999/)
4. Лекция 1. Нейронные сети. Теория (https://compscicenter.ru/courses/data-mining-python2/2018-autumn/classes/3997/)
5. Свёрточные нейронные сети для визуального распознавания (https://www.reg.ru/blog/stenfordskij-kurs-lekciya-1-vvedenie/)

Практическое занятие №9

Тема: Применение свёрточных нейронных сетей для анализа медицинских изображений (В интерактивной форме).

Разновидность занятия: комбинированное.

Методы обучения: объяснительно-иллюстративный, репродуктивный, метод проблемного изложения, частично-поисковый, исследовательский.

Значение темы (актуальность изучаемой проблемы): на данном занятии рассматривается применение нейронных сетей для анализа изображений, в том числе и медицинских изображений.

Формируемые компетенции: ПК-10.1 ,ОПК-2.2.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Организация занятия	5.00	Проверка посещаемости и внешнего вида обучающихся
2	Формулировка темы и целей	10.00	Озвучивание преподавателем темы и ее актуальности, целей занятия
3	Контроль исходного уровня знаний и умений	10.00	Тестирование, индивидуальный устный или письменный опрос, фронтальный опрос
4	Раскрытие учебно-целевых вопросов по теме занятия	10.00	Изложение основных положений темы
5	Самостоятельная работа обучающихся (текущий контроль)	40.00	Выполнение практического задания
6	Итоговый контроль знаний (письменно или устно)	10.00	Тесты по теме, ситуационные задачи
7	Задание на дом (на следующее занятие)	5.00	Учебно-методические разработки следующего занятия и методические разработки для внеаудиторной работы по теме
	ВСЕГО	90	

Аннотация (краткое содержание темы):

Как и все методы машинного обучения, нейронные сети работают с числовыми данными. Изображение — это тоже набор чисел: оно состоит из пикселей, и цвет пикселя задается набором чисел. Если изображение черно-белое, то каждый пиксель задается одним числом от 0 до 255: 0 — черный цвет, 255 — белый цвет, посередине — серый цвет. В цветных изображениях цвет пикселя задается несколькими числами, например в цветовой схеме RGB (red, green, blue) — три числа, отвечающих за интенсивность красного, зеленого и синего цветов.

Нейронные сети довольно легко адаптировать для анализа различных видов данных: для этого используют специальные слои, или блоки, из которых строится нейронная сеть. Для анализа изображений широко используются так называемые сверточные слои.

Свертка — это процедура агрегации информации о пикселе и соседних с ним пикселях.

Пример работы фильтра Собеля

Например, возьмем фильтр размера 3×3, с нулевым вторым столбцом и противоположными значениями в левом и правом столбце. Приложим его к двум различным частям изображения: одна относится к фону, а другая — к границе фона и волос.

При применении такой свертки к фрагменту фона изображения получится $1\times64-1\times65+2\times64-2\times63+1\times65-1\times62 = 4$. Поскольку все пиксели на фоне примерно одинаковые, а числа в фильтре суммируются в ноль, значение свертки получается приблизительно равным нулю. При применении описанной свертки к фрагменту на границе фона и волос получится $1\times64-1\times65+2\times64-2\times25+1\times65-1\times8 = 134$.

Поскольку между пикселями фона и волос происходит резкий перепад яркости, значение свертки оказывается далеким от нуля. В итоге мы получили, что описанный фильтр позволяет выделять границы объектов: в однородных зонах значение свертки близко к нулю, в зона перехода — далеко от нуля. Такой фильтр называется фильтром Собеля. Подробнее прочитать про устройство свертки можно по ссылке Глубокое обучение: разбираемся со свертками, а про различные виды сверток — Матричные фильтры обработки изображений.

Свертки были придуманы для анализа и обработки изображений, например, они используются в графических редакторах наподобие Photoshop для наложения фильтров — собственно, наложение фильтра на изображение в прямом смысле означает применение свертки. При этом в графических редакторах используются заранее заданные фильтры, например, разобранный выше фильтр Собеля.

Конструкция сверточной нейросети

В нейронных сетях фильтры — это настраиваемые параметры. Именно числа, записанные в фильтрах, являются весами сверточного слоя нейронной сети, и эти фильтры настраиваются в процессе обучения по данным. Один сверточный слой состоит из нескольких сверток, и сверточные слои можно ставить друг за другом, по аналогии с полносвязными слоями. Первый

сверточный слой применяется непосредственно к самому изображению, второй слой — к выходу первого сверточного слоя и т. д. Выход сверточного слоя формально тоже является изображением, но на глубоких слоях нейронной сети это «изображение» уже не будет интерпретироваться человеком. Между сверточными слоями, как и между полносвязными, вставляют слои нелинейности, а в конце сверточной архитектуры обычно вставляют один или несколько полносвязных слоев.

Как и в других видах нейронных сетей, в сверточных нейросетях при увеличении номера сверточного слоя повышается уровень абстракции. Первые слои распознают простые переливы яркости и отдельные цвета, слои чуть глубже распознают простые геометрические формы, еще более глубокие слои распознают части изображений, например глаза, губы и нос при анализе лиц, а самые глубокие слои отвечают за распознавание целых объектов.

Примерная тематика НИРС по теме

1. Задачи анализа изображений.
2. Сверточные нейронные сети.
3. Компьютерное зрение.

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.
2. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.

Электронные ресурсы

1. Машинное обучение (курс лекций, К.В.Воронцов) (http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE)
2. Нейронные сети Кохонена (http://neuronus.com/theory/961-nejronnye-seti-kokhonena.html)
3. Обзор открытых источников данных медицинских изображений для машинного обучения (https://webiomed.ai/blog/obzor-otkrytykh-istochnikov-dannykh-meditsinskikh-izobrazhenii-dlia-mashinnogo-obucheniia/)
4. Алгоритмы интеллектуального анализа данных (https://tproger.ru/translations/top-10-data-mining-algorithms/)
5. Библиотека scikit-learn (https://scikit-learn.org/stable/)
6. Библиотека Tensorflow (https://www.tensorflow.org/)
7. Шпаргалка по разновидностям нейронных сетей. Часть первая. Элементарные конфигурации (https://tproger.ru/translations/neural-network-zoo-1/)

8. Сверточная нейронная сеть, часть 2: обучение алгоритмом обратного распространения ошибки (https://habr.com/ru/post/348028/)
9. Лекции Техносферы. Нейронные сети в машинном обучении (https://habr.com/ru/company/mailru/blog/344982/)
10. Лекция 3. Обучение нейронных сетей в Keras (https://compscicenter.ru/courses/data-mining-python2/2018-autumn/classes/3999/)
11. Лекция 1. Нейронные сети. Теория (https://compscicenter.ru/courses/data-mining-python2/2018-autumn/classes/3997/)

Практическое занятие №10

Тема: Систематизация пройденного материала. Стажировка: защита групповой работы.

Разновидность занятия: комбинированное.

Методы обучения: репродуктивный.

Значение темы (актуальность изучаемой проблемы): защита групповых проектов по машинному обучению..

Формируемые компетенции: ОПК-5.2.

Место проведения и оснащение практического занятия: Компьютерный класс №6 (4-60/1) – видеопроектор, доска магнитно-маркерная, комплект учебной мебели на посадочные места, локальный сетевой сервер, персональные компьютеры, экран.

Структура содержания темы (хронокарта практического занятия)

п/п	Этапы практического занятия	Продолжительность (мин.)	Содержание этапа и оснащенность
1	Постановка задачи	5.00	Постановка задачи
7	Защита проектов в формате конференции	85.00	Защита проектов в формате конференции
	ВСЕГО	90	

Аннотация (краткое содержание темы):

Итак, курс состоял из трех основных модулей, напомним результаты каждого из них.

1. Искусственный интеллект:

- мы узнали основные понятия и примеры применения ИИ в бизнесе;
- сформулировали необходимые ресурсы для решения задачи с помощью ИИ.

2. Машинное обучение:

- разобрали типы задач машинного обучения;
- обсудили особенности обучения алгоритмов, подготовки данных к обработке и оценки качества полученного решения;
- зафиксировали инструменты для программирования моделей;
- разобрали примеры решения задач.

3. Глубинное обучение и нейронные сети:

- сформулировали основные понятия и принципы обучения нейронных сетей;
- подробно исследовали особенности архитектур сетей, предназначенных для данных разных типов.

Примерная тематика НИРС по теме

1. На этом занятии НИРС не предусмотрен.

Основная литература

1. Боровиков, В. П. Популярное введение в современный анализ данных в системе STATISTICA : учеб. пособие для вузов / В. П. Боровиков. - М. : Горячая линия-Телеком, 2018. - 288 с. : ил. - Текст : электронный.

Дополнительная литература

1. Наркевич, А. Н. Статистические методы исследования в медицине и биологии : учеб. пособие / А. Н. Наркевич, К. А. Виноградов, К. В. Шадрин ; Красноярский медицинский университет. - Красноярск : КрасГМУ, 2018. - 109 с. - Текст : электронный.
2. Обмачевская, С. Н. Медицинская информатика. Курс лекций : учебное пособие для вузов / С. Н. Обмачевская. - 4-е изд., стер. - Санкт-Петербург : Лань, 2022. - 184 с. - Текст : электронный.

Электронные ресурсы

1. Машинное обучение (курс лекций, К.В.Воронцов) (http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE)
2. UCI Machine Learning Repository (репозиторий машинного обучения) (http://archive.ics.uci.edu/ml/index.php)
3. Нейронные сети Кохонена (http://neuronus.com/theory/961-nejronnye-seti-kokhonena.html)
4. Теория в области нечеткой логики и нечеткого множества (http://neuronus.com/fl/45-theory.html)
5. Поиск ассоциативных правил - К. В. Воронцов (https://www.youtube.com/watch?v=sTWd0ALHdbU&t=2863s)
6. Методы кластеризации - К.В. Воронцов (https://www.youtube.com/watch?v=oWRmzf9eI-c)
7. Алгоритмы интеллектуального анализа данных (https://tproger.ru/translations/top-10-data-mining-algorithms/)
8. Визуализация данных с Python (https://habr.com/ru/company/ods/blog/323210/)
9. Библиотека scikit-learn (https://scikit-learn.org/stable/)
10. Библиотека Tensorflow (https://www.tensorflow.org/)
11. Шпаргалка по разновидностям нейронных сетей. Часть первая. Элементарные конфигурации (https://tproger.ru/translations/neural-network-zoo-1/)
12. Сверточная нейронная сеть, часть 2: обучение алгоритмом обратного распространения ошибки (https://habr.com/ru/post/348028/)
13. Лекции Техносферы. Нейронные сети в машинном обучении (https://habr.com/ru/company/mailru/blog/344982/)
14. Классификация, регрессия и другие алгоритмы Data Mining с использованием R (https://ranalytics.github.io/data-mining/index.html)
15. Наивный баейсовский классификатор (http://datascientist.one/naive-bayes/)
16. Лекция 3. Обучение нейронных сетей в Keras (https://compscicenter.ru/courses/data-mining-python2/2018-autumn/classes/3999/)
17. Лекция 1. Нейронные сети. Теория (https://compscicenter.ru/courses/data-mining-python2/2018-autumn/classes/3997/)