

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Кафедра «Организация перевозок и дорожного движения»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ ПО ДИСЦИПЛИНЕ**

**«ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ НА ТРАНСПОРТЕ».**

**Часть 2**

Направление подготовки 23.04.01 «Технология транспортных процессов»

Программа: «Транспортная логистика», «Организация перевозок на автомобильном транспорте», «Интеллектуальные транспортные системы»

Ростов-на-Дону

ДГТУ

2023

УДК 656

Составитель: Роговенко Т.Н.

Методические указания для практических занятий по дисциплине «Интеллектуальный анализ данных на транспорте». Часть 2 –

Ростов-на-Дону: Донской гос. техн. ун-т, 2023. – 25 с.

УДК 656

Методические указания предназначены для магистрантов очной и заочной формы обучения по направлению подготовки 23.04.01 «Технология транспортных процессов», программы «Транспортная логистика», «Организация перевозок на автомобильном транспорте», «Интеллектуальные транспортные системы». Приведены примеры и задачи, связанные с применением алгоритмов машинного обучения к транспортным данным.

Печатается по решению редакционно-издательского совета Донского государственного технического университета

Ответственный за выпуск зав. кафедрой «Организации перевозок и дорожного движения» д-р техн. наук, профессор В.В. Зырянов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

В печать \_\_\_.\_\_\_.2023 г.

Формат 60×84/16. Объем \_\_\_ усл. п. л.

Тираж \_\_\_ экз. Заказ № \_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Издательский центр ДГТУ

Адрес университета и полиграфического предприятия:

344000, г. Ростов-на-Дону, пл. Гагарина, 1

© Донской государственный

технический университет, 2023

**Введение**

Интеллектуальный анализ данных как одно из новых междисциплинарных направлений включает в себя различные способы работы с данными и в зависимости от предметной области имеет некоторую специфику применения этих способов.

Ставшее уже классическим data mining применимо не ко всем задачам технологии транспортных процессов. В частности, исследование транспортных данных требует специфических подходов не только к очистке, и статистическому анализу, но и к применению ставших классическими алгоритмов машинного обучения.

Последовательное выполнение второй части практических заданий по дисциплине «Интеллектуальный анализ данных на транспорте» позволит получить необходимые магистранту навыки научных исследований и базовые компетенции в области искусственного интеллекта.

Для выполнения всех заданий следует получать исходные данные у преподавателя. Результаты следует сохранять в файлах с именем, состоящим из номера группы, фамилии студента и названия, указанного в условии задачи, например «АМИТС11\_Иванов\_task1» или «AMITS\_Li\_task1». Все примеры следует выполнить и сохранить в одном файле.

**Пример 1**. Применить алгоритм k-means к транспортным данным. Построить диаграмму, иллюстрирующую разбиение на кластеры.

В файле исходных данных содержится макрос k\_means() (приложение 1). В файле содержится 3 столбца по 326 чисел с метками time2, Cnt4, Speed4.Для работы макроса следует подготовить данные, как показано на рисунке 1.

В ячейкe A1 вводим 326, в В1 -2. Для определения количества кластеров *k* можно провести графический анализ зависимости факторов Speed4 и Cnt4 и предположить, что данные можно разбить на 3 или 4 кластера (рисунок 2).

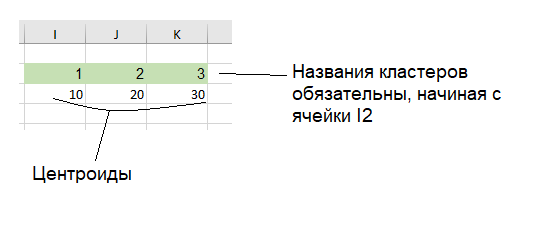
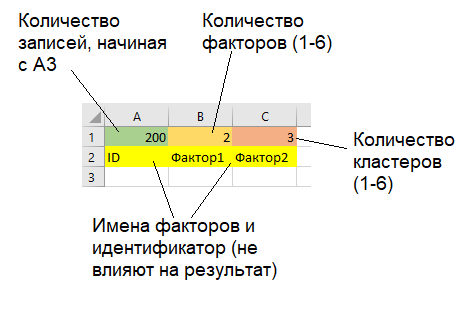


Рисунок 1. Изображение исходных данных для работы макроса

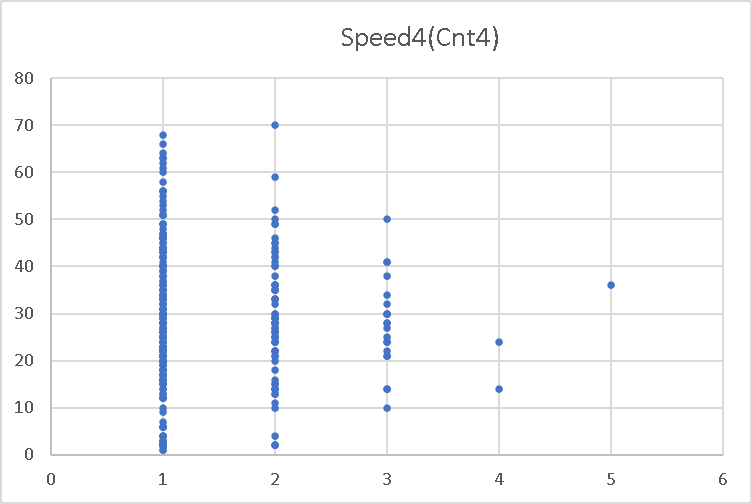


Рисунок 2. График зависимости Speed4от Cnt4

В ячейках I2:K2 размещаем названия кластеров С1, С2, С3. В ячейках I3:K4 – матрицу центроидов.

Результат разбиения на кластеры зависит от начальной матрицы центроидов. Поэтому следует изменять начальную матрицу до получения удовлетворительного результата кластеризации. Здесь приведены результаты для двух начальных матриц:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| А1= | 1 | 20 | 30 | и А2= | 10 | 20 | 0 |
| 5 | 25 | 35 | 50 | 25 | 70 |

Диаграммы, иллюстрирующие разбиение на кластеры при разных начальных матрицах, представлены на рисунке 3.

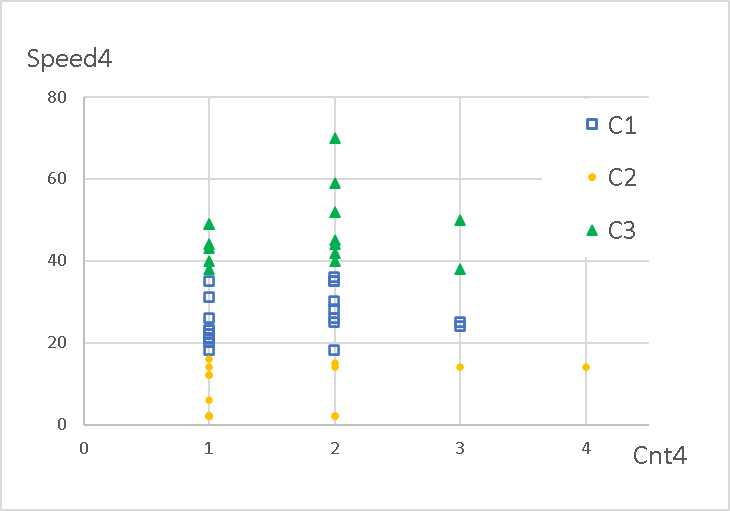
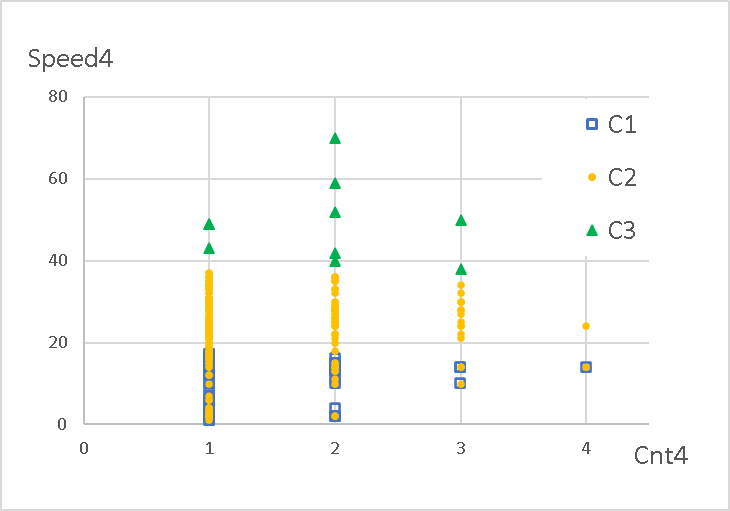


Рисунок 3 - Разбиение на кластеры C1, C2, C3 для матрицы А1 (слева) и А2 (справа)

**Задача 1.** Провести кластеризацию данных из примера 1 для *k*=4.

**Задача 2.** Провести кластеризацию данных из примера 1 для исходной матрицы центроидов, состоящей из медианных значений выборок, для *k*=3 и сравнить результаты. Для решения задачи использовать макрос k\_medians().

**Задача 3**. Исследовать работу алгоритма кластеризации k-means при разных метриках. Используя один и тот же набор данных и матрицу центроидов, провести кластеризацию для нескольких значений *p* (0,1; 0,5; 0,9; 1; 1,5; 2).

Для выполнения этого задания следует внести изменения в макрос k\_means() в соответствии с таблицей 1.

Таблица 1. Изменения в макросе

|  |  |
| --- | --- |
| Старый код | Новый код |
| nc = Cells(1, 3)  '1 centroid | nc = Cells(1, 3)  p = Cells(1, 4)  '1 centroid |
| For j = 1 To nf  For k = 1 To nc  Cells(8 + j, 8 + k) = Cells(2 + j, 8 + k)  Next k  Next j | For j = 1 To nf  For k = 1 To nc  Cells(2 + j, 8 + k) = Cells(8 + j, 8 + k)  Next k  Next j |
| s = s + (Cells(i + 2, j + 1) - Cells(j + 2, 8 + k)) ^ 2 | s = s + Abs(Cells(i + 2, j + 1) - Cells(j + 2, 8 + k)) ^ p |
| Cells(i + 2, 14 + k) = Sqr(s) | Cells(i + 2, 14 + k) = (s) ^ (1 / p) |

Данные подготовить для работы макроса как показано на рисунке 4.

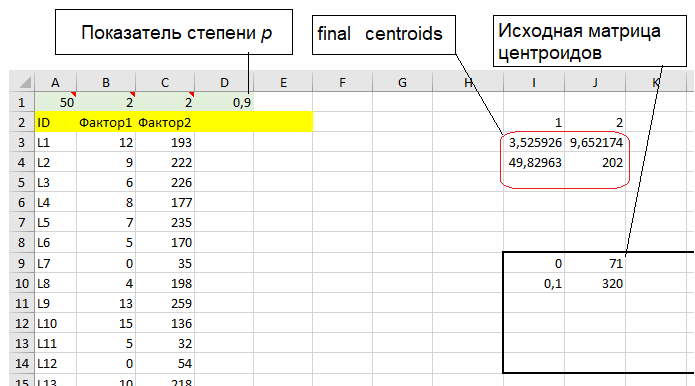


Рисунок 4 - Изображение исходных данных для работы макроса с разными метриками

Результаты работы макроса для разных значений *p* копировать на новый лист и оформить аналогично рисунку 5.

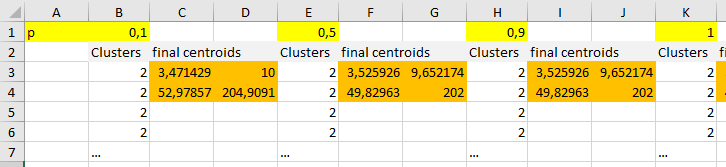


Рисунок 5 - Изображение сводной таблицы результатов применения алгоритма для разных *р*

Диаграммы, иллюстрирующие разбиение на кластеры при *p* =0,1; 0,5; 0,9; 1; 1,5; 2 оформить аналогично примеру 1. Сделать вывод о наиболее подходящей метрике для исходного набора данных.

**Пример 2.** Применить алгоритм kNN к транспортным данным. Построить иллюстрирующую диаграмму.

В исходном файле 5 столбцов по 200 записей о пользователях ОТ с метками: Name, Age, Income (1000s), Cards have, Response. Алгоритм определяет, какое значение примет параметр Response у пользователя D0, опираясь на характеристики *k* ближайших к D0 соседей. Для реализации алгоритма следует выполнить последовательно действия (таблица 2):

1. Рассчитать расстояние distance до «соседей», используя Евклидову метрику

2. Определить параметр small для k (k=1,3,5,…,15) ближайших «соседей».

3. Определить, сколько из k ближайших соседей имеют параметр Response=1

4. Определить, сколько из k ближайших соседей имеют параметр Response=0.

5. Определить, параметр «D0 likely response», сравнивая результаты выбора ближайших «соседей».

Таблица 2. Формулы, реализующие алгоритм kNN

|  |  |  |
| --- | --- | --- |
| № | Диапазон | Формула |
| 1 | F2:F201 | =((B2-$G$2)^2+(C2-$H$2)^2)^0,5 |
| 2 | I4:I11 | =НАИМЕНЬШИЙ(E$2:E$201;F4) |
| 3. | J4:J11 | =СЧЁТЕСЛИМН($E$2:$E$201;J$3;$F$2:$F$201;"<="&$I4) |
| 4. | K4:K11 | =СЧЁТЕСЛИМН($E$2:$E$201;K$3;$F$2:$F$201;"<="&$I4) |
| 5 | H4:H11 | =ЕСЛИ(J4>K4;$J$3;$K$3) |

Начиная с 5 соседей для пользователя D0 параметр Response=1.

На рисунке 6 показана связь между Income и Age , пользователь D0 отмечен зеленым треугольником. В круг включены пять ближайших соседей. Три оранжевые точки – соседи с Response=1, две синие точки – соседи с Response=0, что соответствует результатам работы алгоритма.

**Задача 4.** Изменяя параметры пользователя D0, проверить работу алгоритма kNN.

**Задача 5.** Изменить алгоритм из примера 2 для троичной классификации. (Response=0,1,2).

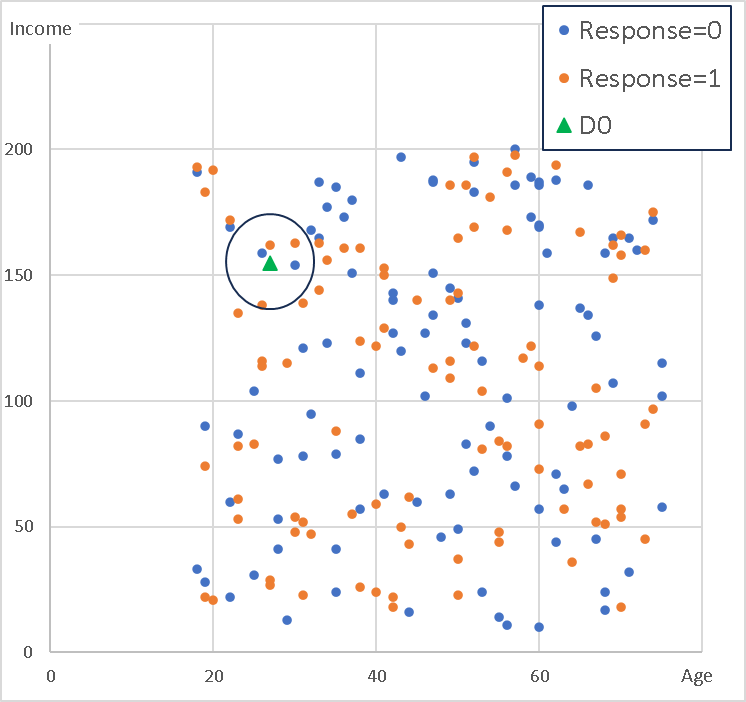


Рисунок 6 – Диаграмма , иллюстрирующая результат применения алгоритма kNN

**Пример 3.** Применить метод «Наивный Байес» к нечисловым характеристикам объектов исследования (таблица 3).

Таблица 3. Формулы, реализующие метод «Наивный Байес»

|  |  |  |
| --- | --- | --- |
| Ячейка | Формула | Диапазон автозаполнения |
| C9 | =СЧЁТЕСЛИ($E$16:$E$25;C$2) | C9:D9 |
| C11 | =C$9/$C1 | C11:D11 |
| C3 | =СЧЁТЕСЛИМН($E$16:$E$25;C$9;$B$16:$B$25;$B3)/C$11 | C3:D4 |
| C5 | =СЧЁТЕСЛИМН($E$16:$E$25;C$2;$C$16:$C$25;$B5)/C$9 | C5:D6 |
| C7 | =СЧЁТЕСЛИМН($E$16:$E$25;C$2;$D$16:$D$25;$B7)/C$9 | C7:D8 |
| C13 | =СУММЕСЛИ($B$3:$B$8;$B27;C$3:C$8)\*СУММЕСЛИ($B$3:$B$8;$C27;C$3:C$8)\*СУММЕСЛИ($B$3:$B$8;$D27;C$3:C$8)\*C$11 | C13:D13 |
| E13 | =C13/(C13+D13) | E13 |
| E27 | =ЕСЛИ(E13>0,5;"yes";"no") | E27 |

**Задача 6.** Придумать пример применения метода «Наивный Байес» к транспортным данным. Заменить наименования факторов и признаков в условии примера 3, ввести собственный обучающий набор и проверить работу метода на разных тестовых образцах.

**Задача 7.** Применить метод «Наивный Байес» для трех факторов с тремя признаками.

**Пример 4.** Применить метод ассоциативного анализа к исходным данным о наборах исследуемых объектов (таблица 4).

Таблица 4. Формулы, реализующие метод ассоциативного анализа

|  |  |  |
| --- | --- | --- |
| Ячейка | Формула | Диапазон автозаполнения |
| D2 | ="I"&C2 | D2:D3539 |
| F2 | =СЧЁТЕСЛИМН($A$2:$A$3539,$E2,$D$2:$D$3539,F$1) | F2:BC1001 |
| BG2 | =СЧЁТЕСЛИМН(F2:F1001,1) | BG2:DD2 |
| BG3 | =СЧЁТЕСЛИМН(ИНДЕКС($F$2:$BC$1001;0;$BE3);1;F$2:F$1001;1) | BG3:DD52 |
| DH3 | =ЕСЛИ($DG3=DH$1,"",ЕСЛИ(И(BG$2>0,BG3/1000>=$DF$3,BG3/BG$2>=$DF$5),ТЕКСТ(BG3/1000,"0.000")&";"&ТЕКСТ(BG3/BG$2,"0.000"),"")) | DH3:FE52 |

**Задача 8.** Найти такие пороговые значения Support и Confidence, при которых не найдется ни одной пары объектов.

**Пример 5**. Построить дерево решений для исходных данных о погодных условиях и проведении мероприятия(таблица 5).

Таблица 5. Формулы для построения дерева решений (уровень 1)

|  |  |  |
| --- | --- | --- |
| Ячейка | Формула | Диапазон автозаполнения |
| B16 | =СЧЁТЗ(E2:E15) | B16 |
| D18 | =СЧЁТЕСЛИМН(E$2:E$15;C18) | D18:D19 |
| E18 | =D18/$B$16\*LOG(D18/$B$16;2) | E18:E19 |
| F18 | =-СУММ(E18:E19) | F18 |
| D22 | =СЧЁТЕСЛИМН(ИНДЕКС($A$2:$E$15;0;$A22);$C22;$E$2:$E$15;D$21) | D22:E31 |
| F22 | =ЕСЛИ(ЕОШИБКА(D22/СУММ($D22:$E22)\*LOG(D22/СУММ($D22:$E22);2));0;D22/СУММ($D22:$E22)\*LOG(D22/СУММ($D22:$E22);2)) | F22:G31 |
| H22 | =-СУММ($D22:$E22)/$B$16\*(F22+G22) | H22:H31 |
| I22 | =СУММЕСЛИМН(H$22:H$31;$A$22:A$31;A22) | I22:I31 |
| J22 | =$F$18-I22 | J22:J31 |

Результаты вычислений - на рисунке 3. Поскольку атрибут Outlook имеет наибольший информационный прирост, он выбран для разделения узла дерева уровня 1.

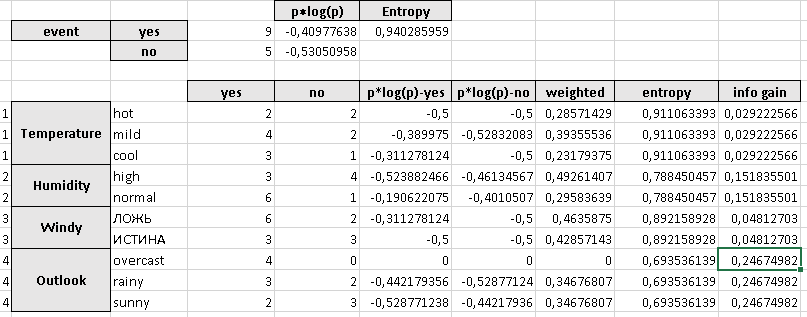


Рисунок 3. Результат вычисления энтропии и прироста информации

Поскольку атрибут Outlook имеет наибольший информационный прирост 0.24674982 среди всех факторов, он выбран для разделения узла дерева уровня 1. Поскольку взвешенная энтропия weighted для outlook-overcast равна нулю, дочерний узел overcast(4,0) является конечным узлом - листом. Мы можем нарисовать простую “древовидную диаграмму”, как показано на рисунке 4.

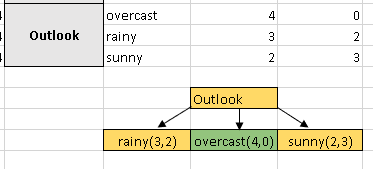


Рисунок 4. Первый уровень дерева решений разделение узла outlook

Следующий уровень - разделение узла rainy(3,2). Создаем копию всего листа , текст внутри ячейки F1 изменяем на “ rainy ” и вносим изменения в копии согласно таблице 6. Результат – на рисунке 5.

Таблица 6. Формулы для построения дерева решений (уровень 2)

|  |  |  |
| --- | --- | --- |
| Ячейка | Формула | Диапазон автозаполнения |
| B16 | =СЧЁТЕСЛИ($D$2:$D$15;$F$1) | B16 |
| D18 | =СЧЁТЕСЛИМН($E$2:$E$15;C18;$D$2:$D$15;$F$1). | D18:D19 |
| D22 | =СЧЁТЕСЛИМН(ИНДЕКС($A$2:$E$15;0;$A22);$C22;$E$2:$E$15;D$21;$D$2:$D$15;$F$1) | D22:E31 |

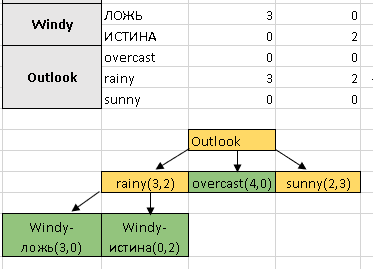


Рисунок 5. Второй уровень дерева решений разделение узла rainy(3,2).

Следующий уровень - разделение узла sunny(2,3). Создаем копию всего листа и текст внутри ячейки F1 изменяем на “sunny”. Dсе значения пересчитаются автоматически. Из результатов, показанных на данном листе видно, что древовидный узел sunny(2,3) должен быть разветвлен в зависимости от влажности. Поскольку оба сгенерированных дочерних узла имеют нулевую энтропию, они оба являются конечными узлами. Таким образом, больше нет необходимости разделять дерево (рисунок 6).

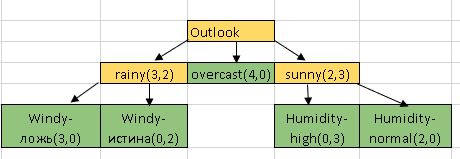


Рисунок 6. Второй уровень дерева решений разделение узла sunny(2,3).

Построено дерево решений для исходных данных о погодных условиях и проведении мероприятия. Фактор «Temperature» не влияет на принятие решения о проведении мероприятия.

**Задача 9.** Изменить самостоятельно исходные данные так, чтобы у каждого фактора было по три признака и построить дерево решений аналогично примеру 5.

**Задача 10.** Создать инструкцию для построения дерева решений к задаче 9 аналогично примеру 5.

**Пример 6.** Создание простой нейронной сети прямого распространения. Рекомендуется выполнять практическое задание в облачном сервисе Google Colab. В случае выполнения задания на своем персональном компьютере, не используя облачный сервис, необходимо установить язык Python, среду для разработки (например PyCharm) и комплект библиотек, которые будут импортироваться в приводимых фрагментах кода.

Последовательность действий при работе в Google Colab:

1. Перейти на облачный сервис по ссылке <https://colab.research.google.com/>

2. В меню выбрать раздел Файл - Создать блокнот

3. Добавить текстовую ячейку (нажать кнопку + Текст ) и ввести текст "Простая многослойная нейронная сеть"

4. Создать ниже еще одну текстовую ячейку и ввести текст "Подключение библиотеки, загрузка набора данных и нормализация его значений"

5. Далее добавить кодовую ячейку (нажать кнопку + Код) и ввести следующий фрагмент кода:

import keras

from keras.datasets import mnist

from keras.utils import np\_utils

from keras.layers import Dense

from keras.layers import Dropout

from keras.layers import Flatten

from keras.models import Sequential

from keras.optimizers import adam\_v2

import matplotlib.pyplot as plt

import numpy as np

#загружается датасет mnist, содержащий изображения рукописных цифр от 0 до 9

(X\_train, y\_train), (X\_test, y\_test) = mnist.load\_data()

X\_train = X\_train.reshape(X\_train.shape[0], X\_train.shape[1]\*X\_train.shape[2]).astype('float32')

X\_test = X\_test.reshape(X\_test.shape[0], X\_test.shape[1]\*X\_test.shape[2]).astype('float32')

X\_train/=255

X\_test/=255

number\_of\_classes = 10

y\_train = np\_utils.to\_categorical(y\_train, number\_of\_classes)

y\_test = np\_utils.to\_categorical(y\_test, number\_of\_classes)

6. Создаем текстовую ячейку и вводим текст "Формирование архитектуры сети"

7. Добавляем кодовую ячейку и вводим следующий фрагмент кода

model = keras.models.Sequential()

model.add(keras.layers.Dense(512, input\_dim=(X\_train.shape[1]), activation='relu'))

model.add(keras.layers.Dropout(0.5))

model.add(keras.layers.Dense(number\_of\_classes, activation='softmax'))

8. Создаем текстовую ячейку и вводим текст "Обучение и тестирование"

9. Добавляем кодовую ячейку и вводим следующий фрагмент кода

model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

history=model.fit(X\_train, y\_train, validation\_data=(X\_test, y\_test), epochs=5, batch\_size=200)

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

Запускаем кодовые ячейки (нажать ) и анализируем качество обучения модели.

10. Создаем текстовую ячейку и вводим текст "Использование обученной модели"

11. В любом графическом редакторе создать изображение пяти цифр, каждая цифра в отдельном файле (например 'one.png', 'two.jpg'). Загрузить изображения в Колабораторий. Для этого слева от Блокнота, в котором вы набираете код, нажать кнопку "Файлы" . Далее нажать кнопку "Загрузить в сессионное хранилище"  и выбрать необходимые файлы.

12. Создаем кодовую ячейку и вводим следующий фрагмент кода (путь к файлу следует копировать с помощью меню, как показано на рисунке 8):

from PIL import Image

#преобразование входного изображения в формат набора MNIS

img = np.invert(Image.open("/nine.jpg").resize((28,28)).convert('L')).ravel()

im2arr = img.reshape(1, 28\* 28).astype('float32')/255

#использование обученной модели

y\_pred = model.predict(im2arr)

print(y\_pred)

print(np.argmax(y\_pred))

13. Создаем текстовую ячейку и вводим текст "Сохранение обученной модели"

14.Создаем кодовую ячейку и вводим следующий фрагмент кода

model.save('model.h5')

В дальнейшем для загрузки сохраненной модели используется:

model\_load=keras.models.load\_model('model.h5')

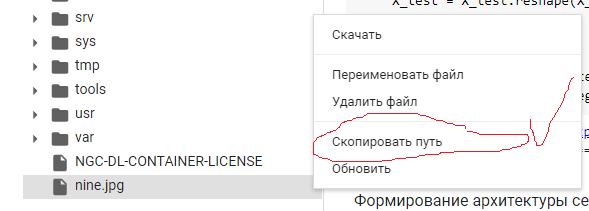


Рисунок 7. Меню для копирования пути к файлу

**Задача 9.** Проанализировать влияние количества нейронов в полносвязном слое (Dense). Изменить 512 нейронов на 128, а затем на 1024. Заново обучить нейронную сеть. Посмотреть, как меняется ошибка ( val\_loss) и точность (val\_accuracy), а затем проверить работу на распознавании пяти цифр.

**Задача 10.** Проанализировать влияние количества скрытых слоев в нейронной сети. Попробовать добавить один, два, а затем три скрытых слоя. Заново обучить нейронную сеть. Посмотреть, как меняется ошибка ( val\_loss) и точность (val\_accuracy), а затем проверить работу на распознавании пяти цифр.

**Приложение 1**

Макрос для примера 1:

Sub k\_means()

Columns("O:Z").Clear

n = Cells(1, 1)

nf = Cells(1, 2)

nc = Cells(1, 3)

For j = 1 To nf

For k = 1 To nc

Cells(8 + j, 8 + k) = Cells(2 + j, 8 + k)

Next k

Next j

56: 'distance

For k = 1 To nc

For i = 1 To n

s = 0

For j = 1 To nf

s = s + (Cells(i + 2, j + 1) - Cells(j + 2, 8 + k)) ^ 2

Next j

Cells(i + 2, 14 + k) = Sqr(s)

Next i

Next k

For i = 1 To n

Min = Cells(i + 2, 14 + 1)

nameC = Cells(2, 8 + 1)

For k = 2 To nc

If Cells(i + 2, 14 + k) < Min Then

Min = Cells(i + 2, 14 + k)

nameC = Cells(2, 8 + k)

End If

Next k

Cells(i + 2, 14 + nc + 1) = nameC

Next i

Cells(2, 14 + nc + 1) = "Clusters"

For k = 1 To nc

For j = 1 To nf

s1 = 0

ns1 = 0

For i = 1 To n

If Cells(i + 2, 14 + nc + 1) = Cells(2, 8 + k) Then

s1 = s1 + Cells(i + 2, 1 + j)

ns1 = ns1 + 1

End If

Next i

If ns1 = 0 Then ns1 = 1

Cells(2 + j, 14 + nc + 2 + k) = s1 / ns1

Next j

Next k

For j = 1 To nf

For k = 1 To nc

Cells(8 + j, 14 + nc + 2 + k) = Cells(2 + j, 14 + nc + 2 + k) - Cells(2 + j, 8 + k)

Next k

Next j

Msg = "Do you want to continue ?"

Style = vbYesNo Or vbCritical Or vbDefaultButton2

Title = "MsgBox Iterations"

Response = MsgBox(Msg, Style, Title)

If Response = vbYes Then

MyString = "Yes"

GoTo 55

Else

MyString = "No"

GoTo 57

End If

55:

For j = 1 To nf

For k = 1 To nc

Cells(2 + j, 8 + k) = Cells(2 + j, 14 + nc + 2 + k)

Next k

Next j

GoTo 56

57: MsgBox ("Get the result !")

End Sub