

Задания к практическим работам

Всего – 16 практических занятий.

Задания 1-2, 3-4, 5-6, 7-8, 9-10, 11-12 выполняются соответственно на практических занятиях 1-2, 3-4, 5-6, 7-8, 9-10, 11-12.

Задания 13-16 выполняются соответственно на практических занятиях 13-16.

Задание 1-2.

Выполнить поиск ассоциативных правил с применением алгоритмов Apriori и FPGrowth. Варианты заданий – в учебно-методическом пособии.

Задание 3-4.

Выполнить визуализацию в *двухмерном пространстве* набора данных, указанного в таблице 1, используя алгоритмы нелинейного снижения размерности:

t-sne (<https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html>)

и

UMAP (<https://umap-learn.readthedocs.io/en/latest/>),

для которых имеются программные реализации в Python.

При работе с набором данных реализовать различные варианты масштабирования:

MinMax

(<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html?highlight=minmax#sklearn.preprocessing.MinMaxScaler>),

Standard

(<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler>),

Robust

(<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html?highlight=robust#sklearn.preprocessing.RobustScaler>).

Выполнить сравнительный анализ результатов масштабирования

(https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html#sphx-glr-auto-examples-preprocessing-plot-all-scaling-py,
<https://www.geeksforgeeks.org/standardscaler-minmaxscaler-and-robustscaler-techniques-ml/>).

Разбиение на обучающую и тестовую выборку не проводить.

Библиотека алгоритмов машинного обучения:

<https://scikit-learn.org/stable/>

Таблица 1. Варианты заданий

Вариант	Источник набора данных
1	http://archive.ics.uci.edu/ml/datasets/Post-Operative+Patient
2	http://archive.ics.uci.edu/ml/datasets/Primary+Tumor
3	http://archive.ics.uci.edu/ml/datasets/Soybean+%28Large%29
4	http://archive.ics.uci.edu/ml/datasets/Low+Resolution+Spectrometer
5	http://archive.ics.uci.edu/ml/datasets/Ionosphere
6	http://archive.ics.uci.edu/ml/datasets/Horse+Colic
7	http://archive.ics.uci.edu/ml/datasets/Hepatitis
8	http://archive.ics.uci.edu/ml/datasets/Heart+Disease
9	http://archive.ics.uci.edu/ml/datasets/Hayes-Roth
10	http://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival
11	http://archive.ics.uci.edu/ml/datasets/Glass+Identification
12	http://archive.ics.uci.edu/ml/datasets/Flags
13	http://archive.ics.uci.edu/ml/datasets/Ecoli
14	http://archive.ics.uci.edu/ml/datasets/Echocardiogram
15	http://archive.ics.uci.edu/ml/datasets/Dermatology
16	http://archive.ics.uci.edu/ml/datasets/Credit+Approval
17	http://archive.ics.uci.edu/ml/datasets/Pittsburgh+Bridges
18	http://archive.ics.uci.edu/ml/datasets/Spambase
19	http://archive.ics.uci.edu/ml/datasets/University
20	http://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval)

Задание 5-6.

Часть А.

Разработать SVM-классификатор для набора данных, указанного в таблице 1.

Разбить выборку на обучающую и тестовую.

Обучить, проверить качество классификатора на обучающей и тестовой выборках: рассчитать общую точность, Recall, Precision, F1-measure.

Оценить число опорных векторов.

Рассмотреть различные типы ядра (линейное, RBF, полиномиальное, сигмоидное (тангенсальное)), различные сочетания значений параметра регуляризации C и параметров ядра. Перебор по сетке (grid search).

Выбрать лучший классификатор.

Выполнить визуализацию с помощью t-sne и UMAP (при различных сочетаниях значений их параметров): изобразить объекты разных классов и опорные векторы разных классов разным цветом (разным маркерами).

Сделать рисунки разбиения на классы на основе выборок с известными метками классов и рисунки разбиения на классы с метками выставленными классификатором.

Часть В.

Разработать knn-классификатор для набора данных, указанного в варианте методических указаний (для задания 1).

Разбить выборку на обучающую и тестовую.

Обучить, проверить качество классификатора на обучающей и тестовой выборках: рассчитать общую точность, Recall, Precision, F1-measure.

Рассмотреть различное число ближайших соседей, различные метрики для вычисления расстояний между объектами, различные правила голосования. Перебор по сетке (grid search).

Выбрать лучший классификатор.

Выполнить визуализацию с помощью t-sne и UMAP (при различных сочетаниях значений их параметров): изобразить объекты разных классов разным цветом (разным маркерами).

Сделать рисунки разбиения на классы на основе выборок с известными метками классов и рисунки разбиения на классы с метками выставленными классификатором.

Часть С.

Разработать RF-классификатор (Random Forest) для набора данных, указанного в варианте методических указаний (для задания 1).

Разбить выборку на обучающую и тестовую.

Обучить, проверить качество классификатора на обучающей и тестовой выборках: рассчитать общую точность, Recall, Precision, F1-measure.

Рассмотреть различное число ближайших соседей, различные метрики для вычисления расстояний между объектами, различные правила голосования. Перебор по сетке (grid search).

Выбрать лучший классификатор.

Выполнить визуализацию с помощью t-sne и UMAP (при различных сочетаниях значений их параметров): изобразить объекты разных классов разным цветом (разным маркерами).

Сделать рисунки разбиения на классы на основе выборок с известными метками классов и рисунки разбиения на классы с метками выставленными классификатором.

Библиотека алгоритмов машинного обучения:

<https://scikit-learn.org/stable/>

Задание 7-8.

Применить к набору данных из задания 3-4 (таблица 1) алгоритм балансировки классов SMOTE.

Разработать для сбалансированного набора данных SVM-, knn-, RF-классификаторы.

Выбрать лучшие классификаторы.

Сравнить классификаторы на основе показателей качества классификации до и после балансировки.

Выполнить визуализацию с помощью t-sne и UMAP (при различных сочетаниях значений их параметров): изобразить объекты разных классов разным цветом (разным маркерами).

Сделать рисунки разбиения на классы на основе выборок с известными метками классов и рисунки разбиения на классы с метками выставленными классификатором.

Задание 9-10.

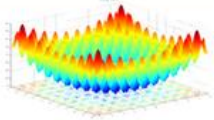
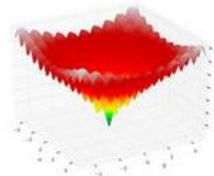
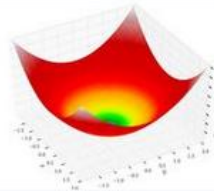
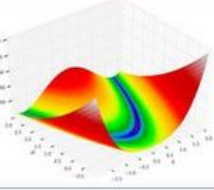
А. Изучить эволюционные алгоритмы оптимизации.

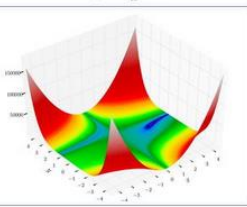
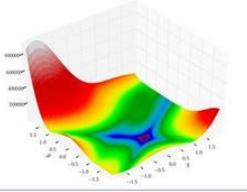
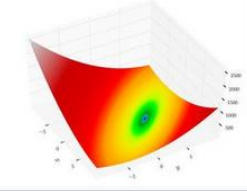
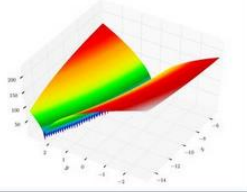
1. Генетический алгоритм
(<https://pypi.org/project/geneticalgorithm/>).
2. Алгоритм роя частиц
(<https://pypi.org/project/pyswarm/>).
3. Алгоритм муравья
(<https://pypi.org/project/PyACO/>).
4. Пчелиный алгоритм
(<https://pypi.org/project/bees-algorithm/>).
5. Алгоритм дифференциальной эволюции
(https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html).

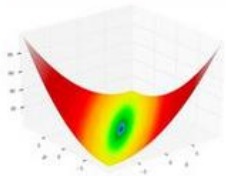
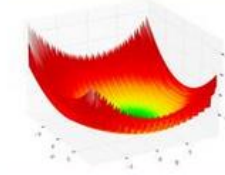
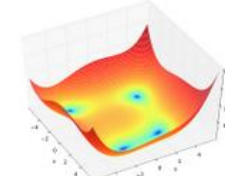

В. Решить задачу оптимизации многоэкстремальной функции, используя классический алгоритм оптимизации (например, алгоритм Ньютона) и эволюционный алгоритм.

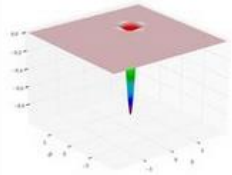
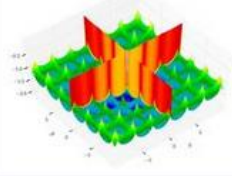
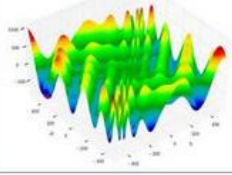
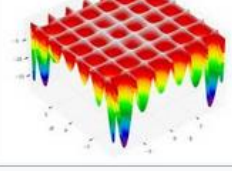
1. Выполнить для каждого из 2 алгоритмов по 100 прогонов, вычислить математическое ожидание и дисперсию для финального значения функции соответствия (целевой функции) алгоритма оптимизации. Для эволюционных алгоритмов подобрать такие значения параметров, при которых значение дисперсии минимально.
2. Оценить время, требуемое для получения априори известного значения глобального экстремума оптимизируемой функции при условии, что завершение работы эволюционного алгоритма осуществляется при достижении значения глобального экстремума.
3. Для эволюционных алгоритмов оценить время нахождения последнего (возможно, локального) экстремума функции соответствия (целевой функции) при условии, что завершение работы алгоритма осуществляется при достижении максимального числа поколений. Вычислить математическое ожидание и дисперсию для времени нахождения последнего локального экстремума функции соответствия (целевой функции). Вычислить математическое ожидание и дисперсию для последнего локального экстремума функции соответствия (целевой функции).
4. Выполнить визуализацию:
 - для оптимизируемой функции (в трехмерном пространстве);
 - для значений функции соответствия (целевой функции) в зависимости от числа поколений в случае использования эволюционного алгоритма.

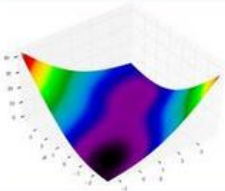
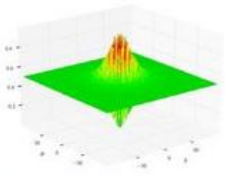
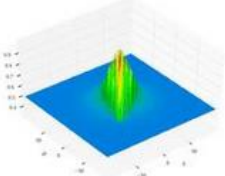
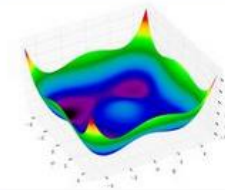
5. Представить в табличном виде результаты расчетов для математического ожидания и дисперсии по значению функции соответствия (целевой функции) и по времени.

Название	Рисунок	Формула	Глобальный минимум	Метод поиска
Функция Растригина		$f(\mathbf{x}) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)]$ <p>where: $A = 10$</p>	$f(0, \dots, 0) = 0$	$-5.12 \leq x_i \leq 5.12$
Функция Экли		$f(x, y) = -20 \exp \left[-0.2 \sqrt{0.5 (x^2 + y^2)} \right] - \exp[0.5 (\cos 2\pi x + \cos 2\pi y)] + e + 20$	$f(0, 0) = 0$	$-5 \leq x, y \leq 5$
Функция сферы		$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$	$f(x_1, \dots, x_n) = f(0, \dots, 0) = 0$	$-\infty \leq x_i \leq \infty,$ $1 \leq i \leq n$
Функция Розенброка		$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	$\text{Min} = \begin{cases} n=2 & \rightarrow f(1, 1) = 0, \\ n=3 & \rightarrow f(1, 1, 1) = 0, \\ n>3 & \rightarrow \underbrace{f(1, \dots, 1)}_{n \text{ times}} = 0 \end{cases}$	$-\infty \leq x_i \leq \infty,$ $1 \leq i \leq n$

Функция Била		$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$	$f(3, 0.5) = 0$	$-4.5 \leq x, y \leq 4.5$
Функция Гольдман-Прайса		$f(x, y) = \left[1 + (x + y + 1)^2 (19 - 14x + 3x^2 - 14y + 6xy + 3y^2) \right] \left[30 + (2x - 3y)^2 (18 - 32x + 12x^2 + 48y - 36xy + 27y^2) \right]$	$f(0, -1) = 3$	$-2 \leq x, y \leq 2$
Функция Бута		$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$	$f(1, 3) = 0$	$-10 \leq x, y \leq 10$
Функция Букина N 6		$f(x, y) = 100\sqrt{ y - 0.01x^2 } + 0.01 x + 10 .$	$f(-10, 1) = 0$	$-15 \leq x \leq -5, -3 \leq y \leq 3$

Функция Матьяса		$f(x, y) = 0.26(x^2 + y^2) - 0.48xy$	$f(0, 0) = 0$	$-10 \leq x, y \leq 10$
Функция Леви N 13		$f(x, y) = \sin^2 3\pi x + (x - 1)^2 (1 + \sin^2 3\pi y) + (y - 1)^2 (1 + \sin^2 2\pi y)$	$f(1, 1) = 0$	$-10 \leq x, y \leq 10$
Функция Химмельблау		$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$	$\text{Min} = \begin{cases} f(3.0, 2.0) & = 0.0 \\ f(-2.805118, 3.131312) & = 0.0 \\ f(-3.779310, -3.283186) & = 0.0 \\ f(3.584428, -1.848126) & = 0.0 \end{cases}$	$-5 \leq x, y \leq 5$
Функция трехгорбного верблюда		$f(x, y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2$	$f(0, 0) = 0$	$-5 \leq x, y \leq 5$

Функция Изома		$f(x, y) = -\cos(x) \cos(y) \exp\left(-\left((x - \pi)^2 + (y - \pi)^2\right)\right)$	$f(\pi, \pi) = -1$	$-100 \leq x, y \leq 100$
Cross-in-tray function		$f(x, y) = -0.0001 \left[\left \sin x \sin y \exp\left(\left 100 - \frac{\sqrt{x^2 + y^2}}{\pi} \right \right) \right + 1 \right]^{0.1}$	$\text{Min} = \begin{cases} f(1.34941, -1.34941) & = -2.06261 \\ f(1.34941, 1.34941) & = -2.06261 \\ f(-1.34941, 1.34941) & = -2.06261 \\ f(-1.34941, -1.34941) & = -2.06261 \end{cases}$	$-10 \leq x, y \leq 10$
Функция Эгнхольдера		$f(x, y) = -(y + 47) \sin \sqrt{\left \frac{x}{2} + (y + 47) \right } - x \sin \sqrt{ x - (y + 47) }$	$f(512, 404.2319) = -959.6407$	$-512 \leq x, y \leq 512$
Табличная функция Хольдера		$f(x, y) = - \left \sin x \cos y \exp\left(\left 1 - \frac{\sqrt{x^2 + y^2}}{\pi} \right \right) \right $	$\text{Min} = \begin{cases} f(8.05502, 9.66459) & = -19.2085 \\ f(-8.05502, 9.66459) & = -19.2085 \\ f(8.05502, -9.66459) & = -19.2085 \\ f(-8.05502, -9.66459) & = -19.2085 \end{cases}$	$-10 \leq x, y \leq 10$

Функция МакКормика		$f(x, y) = \sin(x + y) + (x - y)^2 - 1.5x + 2.5y + 1$	$f(-0.54719, -1.54719) = -1.9133$	$\begin{aligned} -1.5 \leq x \leq 4, \\ -3 \leq y \leq 4 \end{aligned}$
Функция Шаффера N2		$f(x, y) = 0.5 + \frac{\sin^2(x^2 - y^2) - 0.5}{[1 + 0.001(x^2 + y^2)]^2}$	$f(0, 0) = 0$	$-100 \leq x, y \leq 100$
Функция Шаффера N4		$f(x, y) = 0.5 + \frac{\cos^2[\sin(x^2 - y^2)] - 0.5}{[1 + 0.001(x^2 + y^2)]^2}$	$f(0, 1.25313) = 0.292579$	$-100 \leq x, y \leq 100$
Функция Стыбинского- Танга		$f(\mathbf{x}) = \frac{\sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i}{2}$	$-39.16617n < \underbrace{f(-2.903534, \dots, -2.903534)}_{n \text{ times}} < -39.16616n$	$\begin{aligned} -5 \leq x_i \leq 5, \\ 1 \leq i \leq n.. \end{aligned}$

Тестовые функции для оптимизации:

https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B5_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8_%D0%B4%D0%BB%D1%8F_%D0%BE%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8

С. Реализовать поиск оптимальных значений параметров для SVM, knn, RF-классификаторов с применением эволюционного алгоритма оптимизации.

P.S.

Названия некоторых алгоритмов

1. Генетический алгоритм.
2. Алгоритм роя частиц.
3. Алгоритм муравья.
4. Пчелиный алгоритм.
5. Алгоритм дифференциальной эволюции.
6. Алгоритм клонального отбора.
7. Алгоритм косяка рыб.
8. Алгоритм кукушки.
9. Бактериальный алгоритм.
10. Алгоритм светлячков.
11. Сорняковый алгоритм.
12. Обезьяний алгоритм.
13. Алгоритм прыгающих лягушек.
14. Алгоритм летучих мышей.
15. Алгоритм растущих деревьев.
16. Алгоритм поиска гармонии.
17. Алгоритм гравитационного поиска.
18. Электромагнитный алгоритм.
19. Алгоритм эволюции разума.
20. Диффузный алгоритм.
21. Культурный алгоритм.

По факту – их очень много. И их число постоянно растет.

Задание 11-12.

Выполнить кластеризацию своего набора данных, считая, что метки кластеров неизвестны.

Сравнить результаты кластеризации (метки кластеров и реальные метки классов).

Использовать алгоритмы: k-means, fcm, DBSCAN.

Оценить качество кластеризации по индексу кластерного силуэта. Определить оптимальное число кластеров (рассмотреть число кластеров от 2 до 10).

Выбрать лучший алгоритм для своего набора данных.

Выполнить визуализацию с помощью t-sne и UMAP (при различных сочетаниях значений их параметров): изобразить объекты разных кластеров разным цветом (разным маркерами). Отметить центры кластеров (если алгоритм их вычисляет).

Библиотека алгоритмов машинного обучения:

<https://scikit-learn.org/stable/>

Задание 13-16.

Выполнить исследование и сравнительный анализ возможностей RNN, LSTM и GRU на примере предлагаемого кода.

https://github.com/Azure/Istms_for_predictive_maintenance/blob/master/Deep%20Learning%20Basics%20for%20Predictive%20Maintenance.ipynb

Работаем с TensorFlow! Например, 1.15.

```
In [1]: import keras

Using TensorFlow backend.

In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Setting seed for reproducibility
np.random.seed(1234)
PYTHONHASHSEED = 0
from sklearn import preprocessing
from sklearn.metrics import confusion_matrix, recall_score, precision_score
from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM, Activation
%matplotlib inline
```

Выполнить работу по варианту, соответствующему номеру с id авиационного двигателя в наборе данных.

1. Сравнить полученные нейронные сети по Accuracy, Precision, Recall, F1, Loss на train и test.

Выполнить несколько запусков программы с разными seed:

```
# Setting seed for reproducibility
np.random.seed(1234)
```

Выбрать лучший вариант.

2. Выполнить исследования на примере фрагмента кода с заменой LSTM на RNN и GRU.

```
# build the network
nb_features = seq_array.shape[2]
```

```

nb_out = label_array.shape[1]

model = Sequential()

model.add(LSTM(
    input_shape=(sequence_length, nb_features),
    units=100,
    return_sequences=True))
model.add(Dropout(0.2))

model.add(LSTM(
    units=50,
    return_sequences=False))
model.add(Dropout(0.2))

model.add(Dense(units=nb_out, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])

```

Изучить и описать назначение используемых методов и параметров.

3. Исследовать, как определяется число параметров Param в каждом слое.

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

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 50, 100)	50400
dropout_1 (Dropout)	(None, 50, 100)	0
lstm_2 (LSTM)	(None, 50)	30200
dropout_2 (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 1)	51

```

=====
Total params: 80,651
Trainable params: 80,651
Non-trainable params: 0

```

4. Изучить и описать назначение используемых методов и параметров.

```

%%time
# fit the network
model.fit(seq_array, label_array, epochs=10, batch_size=200,
validation_split=0.05, verbose=1,
        callbacks = [keras.callbacks.EarlyStopping(monitor='val_loss',
min_delta=0, patience=0, verbose=0, mode='auto')])

```

5. Вывести графические зависимости для Loss и Accuracy на train и val (на обучающей и валидационной подвыборках).
6. Оценить время разработки классификаторов с CPU.
7. Оценить время разработки классификаторов с GPU (в Google Colab).