

УДК 004.423

## ***ВИЗУАЛИЗАЦИЯ BIG DATA ДЛЯ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ***

***Кулинча П.В.***

*студент направления подготовки информатика и вычислительная техника,  
Хакасский государственный университет имени Н.Ф. Катанова,  
г. Абакан, Россия <sup>1</sup>*

**Аннотация:** В современном мире объемы данных, с которыми приходится работать специалистам в различных областях, растут с каждым днем. Это явление получило название «Big Data». Процесс анализа и интерпретации таких объемов данных может оказаться чрезвычайно сложным без использования специализированных методов и инструментов. Одним из ключевых направлений в работе с большими данными является визуализация, которая позволяет представлять данные в удобной и понятной форме, значительно упрощая процесс принятия решений. В данной статье будут рассмотрены различные методы и инструменты для эффективной визуализации Big Data, а также представлены примеры кода, демонстрирующие их применение на практике.

**Ключевые слова:** Big Data, визуализация данных, принятие решений, аналитика данных

## ***BIG DATA VISUALIZATION FOR DECISION SUPPORT***

***Kulincha P.V.***

*student of computer science and computer engineering department,  
N.F. Katanov Khakass State University,  
Abakan, Russia*

---

<sup>1</sup> Научный руководитель: Спирин Д.В. доцент кафедры ЦТиД, Хакасский государственный университет имени Н.Ф. Катанова, г. Абакан, Россия

**Abstract:** In the modern world, the amount of data that specialists in various fields have to work with is growing every day. This phenomenon is called "Big Data". The process of analyzing and interpreting such volumes of data can be extremely difficult without the use of specialized methods and tools. One of the key areas in working with big data is visualization, which allows you to present data in a convenient and understandable form, greatly simplifying the decision-making process. This article will review various methods and tools for effective visualization of Big Data, as well as provide code examples demonstrating their application in practice.

**Keywords:** Big Data, data visualization, decision making, data analytics

### **Методы визуализации Big Data**

Методы визуализации больших данных могут варьироваться от простых диаграмм и графиков до сложных многомерных и интерактивных визуализаций. Выбор метода зависит от характеристик данных и целей анализа. Рассмотрим несколько популярных методов [1]:

- **Тепловые карты (Heatmaps):** позволяют визуализировать плотность или интенсивность величин в двумерном пространстве. Применяются для анализа географических данных, матриц корреляции и т.д.
- **Деревья и карты (Tree maps):** используются для визуализации иерархических структур, позволяя эффективно представить структуру данных и относительные размеры ее компонентов.
- **Сетевые графики (Network graphs):** предназначены для визуализации отношений между объектами, идеально подходят для анализа социальных сетей, систем связей и зависимостей.

### **Инструменты визуализации Big Data**

Существует множество инструментов, предназначенных для визуализации больших объемов данных. Среди наиболее популярных можно выделить:

- Tableau: платформа для интерактивной визуализации данных, позволяющая пользователям создавать сложные визуализации без необходимости программирования.
- D3.js: библиотека JavaScript для создания динамических и интерактивных визуализаций в вебе, предоставляет гибкие возможности для работы с большими данными.
- Matplotlib и Seaborn (Python): библиотеки для статической, анимированной и интерактивной визуализации данных в Python.

Примеры визуализации с использованием Python

Рассмотрим примеры кода для визуализации данных с использованием библиотек Matplotlib и Seaborn.

Пример кода для создания тепловой карты корреляции признаков представлен на рисунке 1.

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd

# Создание тестового набора данных
data = np.random.rand(10, 10)
df = pd.DataFrame(data, columns=[f'feature_{i}' for i in range(1, 11)])

# Построение тепловой карты
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Тепловая карта корреляции признаков')
plt.show()
```

Рисунок 1 – Создание тепловой карты [разработано автором]

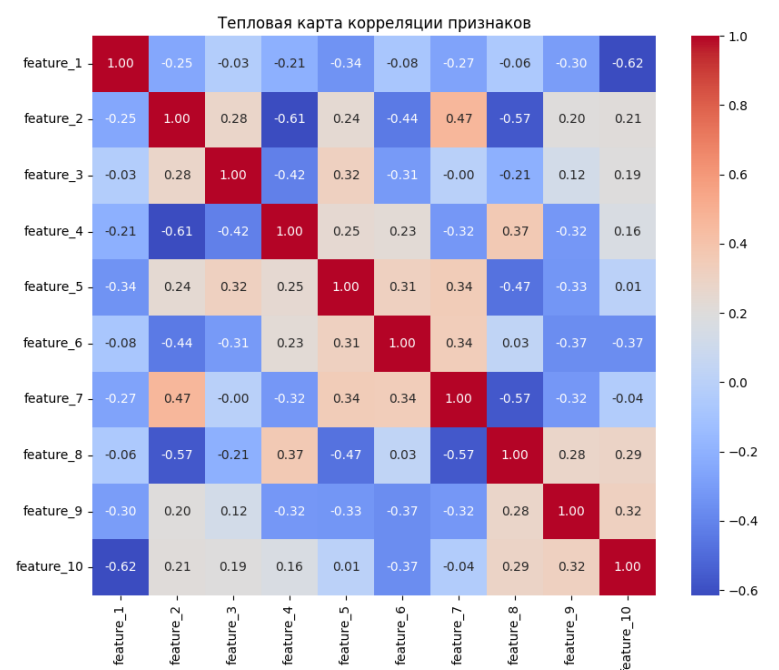


Рисунок 2 – Результат выполнения кода [разработано автором]

Данный фрагмент кода демонстрирует создание тепловой карты для визуализации матрицы корреляции между признаками в наборе данных. Сначала создается тестовый набор данных, состоящий из случайных чисел. Затем, с использованием библиотеки Pandas, формируется DataFrame. После чего, с помощью функции `heatmap` из библиотеки Seaborn, строится визуализация матрицы корреляции. Аннотации на карте (`annot=True`) показывают численные значения корреляций, что облегчает интерпретацию взаимосвязей между признаками.

### Визуализация иерархических данных с использованием Tree Map

Для демализации иерархических данных с использованием метода Tree Map можно использовать библиотеку Squarify в Python. Этот инструмент позволяет представлять иерархические данные в виде вложенных прямоугольников, размер которых соответствует определенному числовому параметру данных [2].

Пример кода, демонстрирующий создание простой Tree Map, отображающей распределение определенных значений, представлен на рисунке 3.

```
import matplotlib.pyplot as plt
import squarify

# Определение размеров и меток для каждого элемента
sizes = [10, 20, 30, 40]
labels = ['Раздел 1', 'Раздел 2', 'Раздел 3', 'Раздел 4']

# Создание Tree Map
plt.figure(figsize=(12, 8))
squarify.plot(sizes=sizes, label=labels, alpha=0.7)
plt.axis('off')
plt.title('Пример Tree Map')
plt.show()
```

Рисунок 3 – Создание Tree Map [разработано автором]

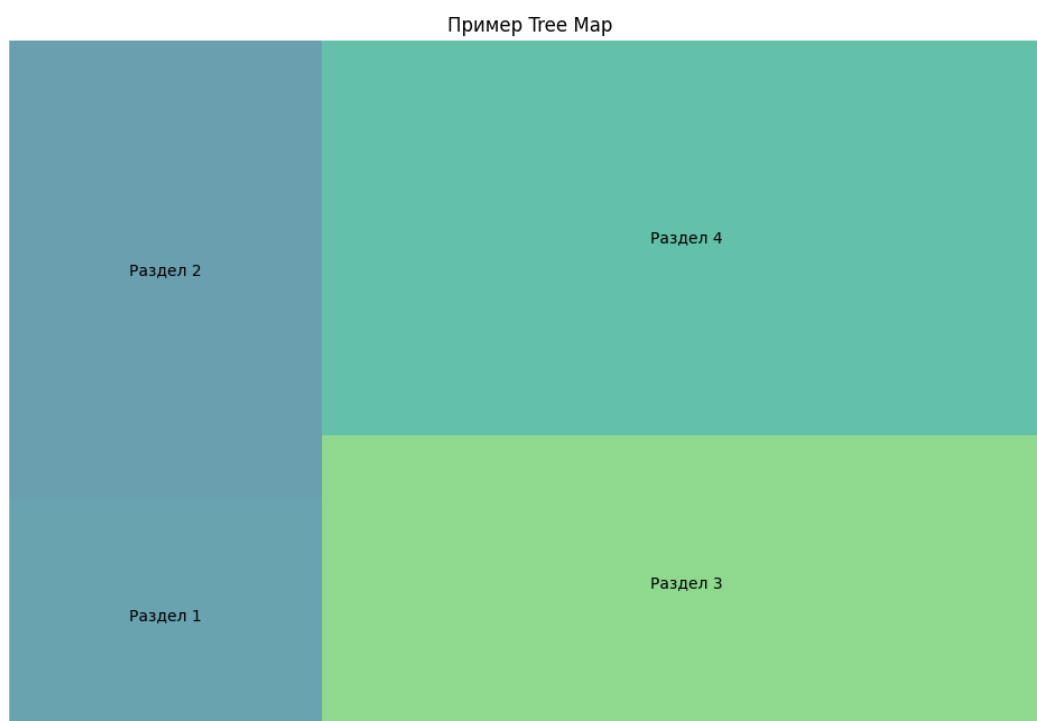


Рисунок 4 – Результат выполнения кода [разработано автором]

В этом примере массив `sizes` содержит числовые значения, которые отображаются в виде прямоугольников разного размера. Массив `labels` содержит метки для каждого прямоугольника. Функция `squarify.plot()` используется для

создания и отображения Tree Map. Параметр alpha управляет прозрачностью прямоугольников.

Визуализация сетевых графиков с использованием библиотеки NetworkX

Сетевые графики позволяют визуализировать сложные сети взаимосвязей между объектами. Библиотека NetworkX в Python предоставляет обширные возможности для создания, манипулирования и изучения структуры и динамики сетей [2]. Пример кода для создания графа представлен на рисунке 5.

```
import networkx as nx
import matplotlib.pyplot as plt

# Создание графа
G = nx.Graph()

# Добавление узлов и ребер
G.add_edges_from([('A', 'B'), ('B', 'C'), ('C', 'A'), ('C', 'D')])

# Визуализация графа
plt.figure(figsize=(8, 6))
nx.draw(G, with_labels=True, node_color='skyblue', node_size=700, edge_color='k')
plt.title('Визуализация сетевого графа')
plt.show()
```

Рисунок 5 – Создание сетевого графа [разработано автором]

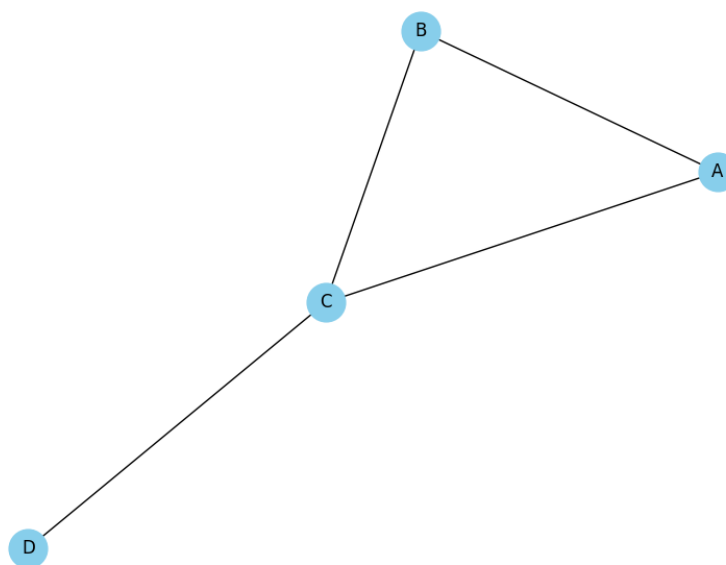


Рисунок 5 – Результат выполнения кода [разработано автором]

В этом коде создается простой неориентированный граф с четырьмя узлами и четырьмя ребрами. Функция nx.draw() используется для визуализации графа, где with\_labels=True указывает на необходимость отображения меток

узлов, `node_color` и `node_size` задают цвет и размер узлов соответственно, а `edge_color` — цвет ребер.

### **Заключение**

Визуализация Big Data играет ключевую роль в процессе анализа и принятия решений, предоставляя возможность наглядно представить сложные наборы данных. Рассмотренные методы и инструменты являются лишь частью широкого спектра возможностей, доступных специалистам для работы с большими данными. Выбор конкретного метода или инструмента зависит от специфики задачи, характеристик данных и целей анализа. Примеры кода, представленные в данной статье, демонстрируют основные принципы использования популярных библиотек Python для визуализации данных и могут служить отправной точкой для разработки собственных решений в области визуализации Big Data.

### **Библиографический список:**

1. Devpractice Team. Python. Визуализация данных. Matplotlib. Seaborn. Mayavi. - devpractice.ru. 2020. - 412 с.: ил.
2. Devpractice Team. Библиотека Matplotlib. - devpractice.ru. 2019. - 100 с.: ил.

*Оригинальность 80%*