**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**Факультет информационных технологий и анализа больших данных**

**Департамент анализа данных и машинного обучения**

**КУРСОВАЯ РАБОТА**

на тему:

«Предварительный анализ данных и построение признаков в задачах сжатия информации»

Студента группы ПМ22-6

Свиридова Савелия Сергеевича

Москва 2024

Оглавление

[**Введение** 3](#_Toc166341265)

[**Цель курсовой работы:** 3](#_Toc166341266)

[**Актуальность проблемы:** 3](#_Toc166341267)

[**Обзор существующих методов:** 3](#_Toc166341268)

[**Задачи работы:** 3](#_Toc166341269)

[**Практическая часть:** 4](#_Toc166341270)

[**Анализ результатов и формулирование выводов:** 14](#_Toc166341271)

[**Рекомендации для Дальнейших Исследований:** 15](#_Toc166341272)

[**Литература** 16](#_Toc166341273)

[**Основная литература:** 16](#_Toc166341274)

[**Дополнительная литература:** 16](#_Toc166341275)

# **Введение**

## **Цель курсовой работы:**

Разработать методы предварительного анализа данных и построения признаков для эффективного сжатия информации.

## **Актуальность проблемы:**

С развитием технологий и увеличением объемов создаваемой информации возрастает потребность в эффективных способах её сжатия. Сжатие данных не только помогает экономить место для хранения, но и ускоряет передачу информации через сеть. Применение методов машинного обучения и разработка новых алгоритмов представляют собой значительную инновацию, позволяющую преодолеть текущие ограничения и достичь новых уровней эффективности и точности.

## **Обзор существующих методов:**

Основные методы сжатия данных делятся на сжатие без потерь и сжатие с потерями. Сжатие без потерь, включающее алгоритмы RLE (Run-Length Encoding), Huffman Coding, LZW (Lempel-Ziv-Welch) и Deflate, позволяет восстановить данные в их первоначальный вид. Также применяются методы машинного обучения, такие как PCA (Principal Component Analysis) и автоэнкодеры и дэкодеры, которые помогают снизить размерность данных и выявить скрытые факторы, что особенно полезно для сжатия многомерных данных и больших датасетов, а также для предиктивного сжатия.

## **Задачи работы:**

Изучение теоретических основ сжатия данных: Ознакомление с основными алгоритмами и методами сжатия, такими как сжатие с потерями и без потерь.

1. Выбор и анализ датасета. Особенности исходных данных, а также провести их визуализацию.
2. Предварительный анализ данных: Применение статистических методов для исследования характеристик данных, включая очистку и нормализацию данных.
3. Построение признаков: Разработка и реализация методов генерации новых признаков для улучшения эффективности сжатия данных.
4. Реализация алгоритмов сжатия данных: Программирование и тестирование различных алгоритмов сжатия на выбранном датасете.
5. Оценка эффективности алгоритмов.
6. Сравнение RLE с алгоритмом Huffman.
7. Применение машинного обучения для предиктивного сжатия
8. Анализ результатов и формулирование выводов: Оценка полученных результатов и их обсуждение в контексте поставленных задач.

# **Практическая часть:**

1. **Выбор и анализ датасета. собенности исходных данных, а также их визуализация.**

Датасет MNIST. Этот датасет является одним из самых известных в мире машинного обучения и обычно используется для задач распознавания рукописных цифр.

Особенность данных:

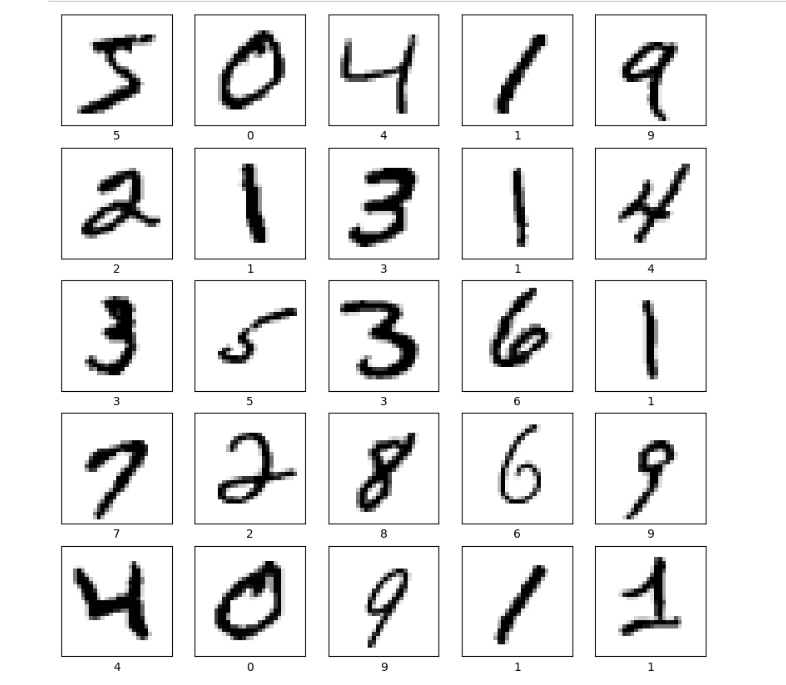
Он содержит 60,000 обучающих и 10,000 тестовых изображений рукописных цифр от 0 до 9, каждое из которых представлено в градациях серого (черно-белом) формате размером 28x28 пикселей.

Важные аспекты использования MNIST:

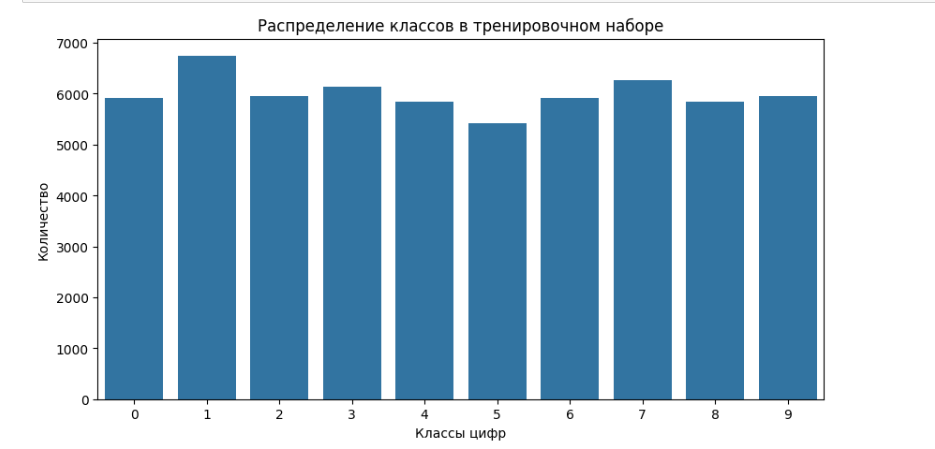
Нормализация: Данные нормализованы путем деления значений пикселей на 255, чтобы привести их к диапазону от 0 до 1.

Равномерное распределение: Анализирование распределения различных цифр в тренировочном наборе данных, чтобы убедиться, что они равномерно распределены. Это важно, так как дисбаланс классов может влиять на производительность алгоритмов.

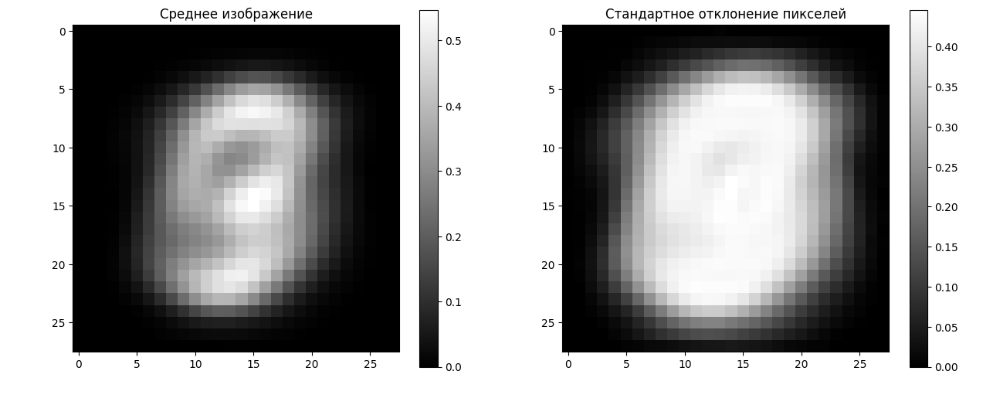
Визуализация данных :



2. **Предварительный анализ данных: Применение статистических методов для исследования характеристик данных, включая очистку и нормализацию данных**

****

Получили гистограмму, показывающую количество изображений для каждой цифры в тренировочном наборе данных. Это важно для того, чтобы убедиться, что данные равномерно распределены по классам, что может влиять на производительность алгоритмов машинного обучения и сжатия данных.

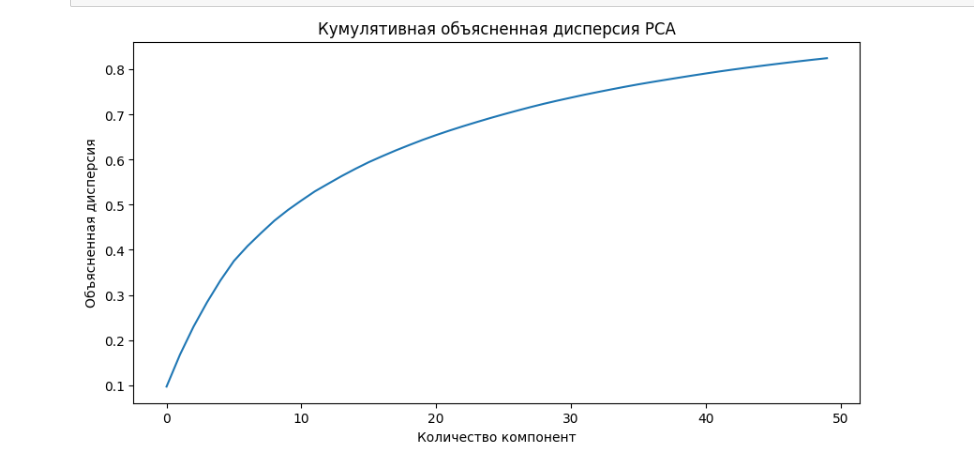
Перейдем к анализу пикселей

Эти изображения показывают, какие пиксели в среднем чаще всего используются и какова степень их вариативности по датасету

1. **Построение признаков: Разработка и реализация методов генерации новых признаков для улучшения эффективности сжатия данных.**

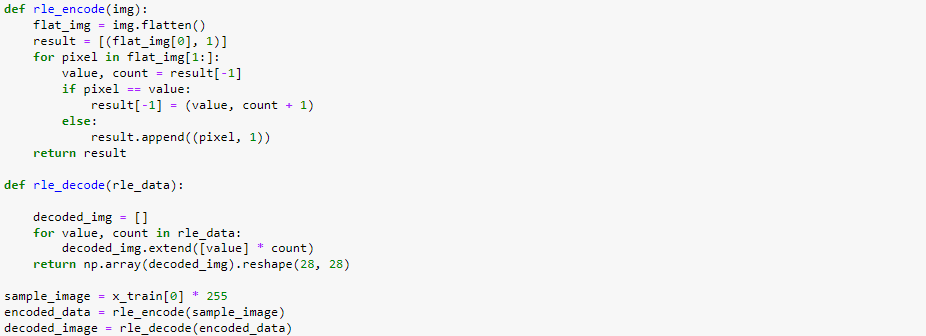
Метод главных компонентов (Principal Component Analysis, PCA) - это статистический метод, используемый для уменьшения размерности данных с минимальной потерей информации. Он широко применяется в анализе данных для визуализации, упрощения моделей, в обработке сигналов и многом другом

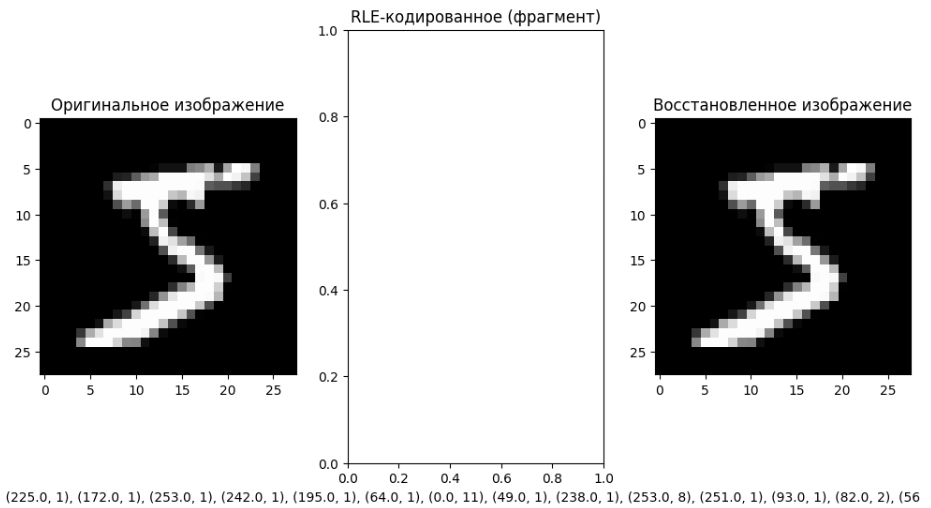
Главные компоненты выбираются на основе собственных значений.



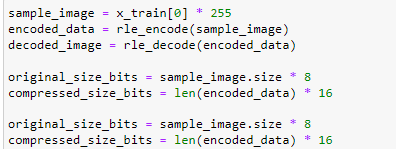
1. **Реализация алгоритмов сжатия данных: Программирование и тестирование различных алгоритмов сжатия на выбранном датасете.**

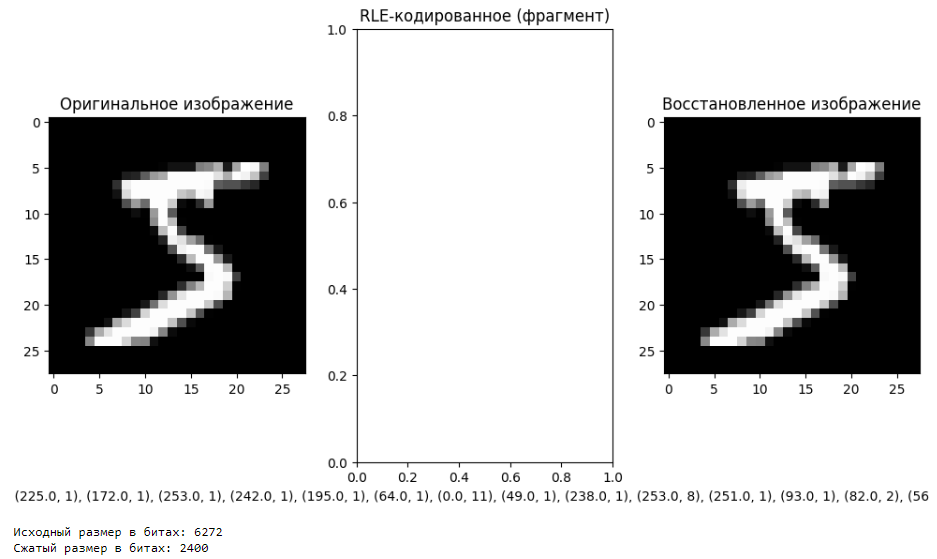
Был выбран метод RLE. (Run-Length Encoding) — это метод сжатия данных без потерь, который работает за счёт сжатия последовательностей одинаковых элементов. Он заменяет эти последовательности коротким описанием, состоящим из символа и его количества в последовательности. Этот метод особенно эффективен для данных, содержащих множество одинаковых, повторяющихся элементов, например, для черно-белых изображений или любых других данных с большими однородными участками.





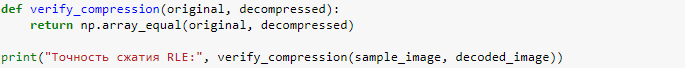
Был изменен размер данных с 32-битных целых на 16-битные для каждой пары значений в RLE. Это более реалистично для данного контекста, особенно если значения пикселей и количества не превышают 255. Этот подход может помочь уменьшить размер данных при использовании RLE, если значения количеств пикселей и сами пиксели не требуют 32 бит для хранения.





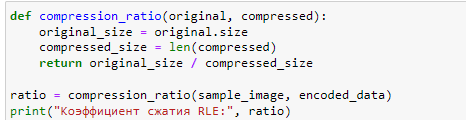
Для проверки корректности RLE-кодирования важно убедиться, что восстановленное изображение точно соответствует оригиналу. Исходя из изображения, восстановленное изображение выглядит очень похожим на оригинальное, что указывает на правильное выполнение кодирования и декодирования.

1. **Оценка эффективности алгоритмов:**



Точность сжатия RLE: True

Если функция выдает True , это означает, что процесс декомпрессии успешно восстановил оригинальное изображение без потерь. Следовательно, сжатие и последующая декомпрессия данных прошли таким образом, что все данные оригинального изображения были точно восстановлены.



Коэффициент сжатия RLE: 0.47717589774802194

1. **Сравнение RLE с алгоритмом Huffman.**

Начнем с того, что такое алгоритм Huffman

Алгоритм Хаффмана — это метод сжатия данных без потерь, основанный на принципе использования переменной длины кода для представления символов. Этот метод был разработан Дэвидом Хаффманом в 1952 году в качестве части его докторской диссертации. Он широко используется для сжатия текста, изображений, аудио и видео данных. Принцип работы алгоритма заключается в следующем:

Как работает алгоритм Хаффмана

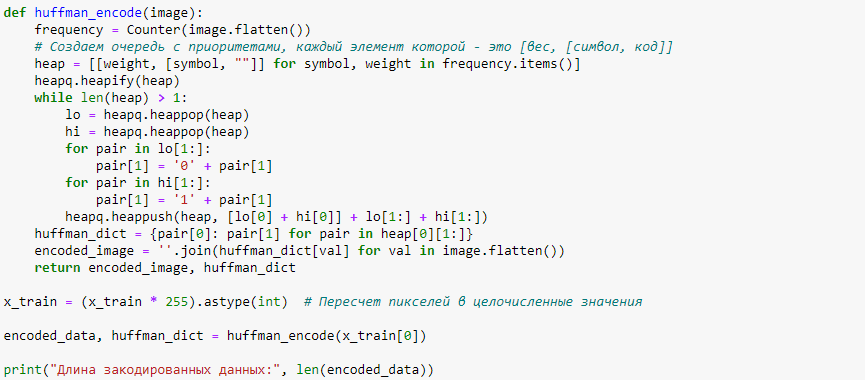
Построение частотной таблицы: Сначала алгоритм анализирует данные и подсчитывает частоту встречаемости каждого символа (или пикселя, звукового семпла и т.д.).

Построение дерева Хаффмана:

Создаются узлы для каждого символа, и каждому узлу присваивается частота его символа.

Узлы с наименьшей частотой объединяются в один узел, суммируя их частоты. Новый узел становится их родителем.

Процесс повторяется, пока не останется один узел без родителя.



Длина закодированных данных равна 1643, что означает количество бит, необходимых для представления закодированного изображения с использованием алгоритма Хаффмана. Это показатель эффективности сжатия данных.

Сравнение с исходным размером: Исходное изображение MNIST имеет размер 28x28 пикселей, каждый пиксель представлен одним байтом (при условии, что значения пикселей были переведены из вещественных чисел от 0 до 1 в целые числа от 0 до 255). Таким образом, исходный размер изображения составляет: 6272 бит.

Сравнение 1643 бит против 6272 бит показывает, что использование Хаффмана существенно уменьшило количество необходимых бит для хранения изображения.

Производительность: Закодированные данные занимают меньше места на диске или в памяти, что улучшает производительность хранения и передачи данных.

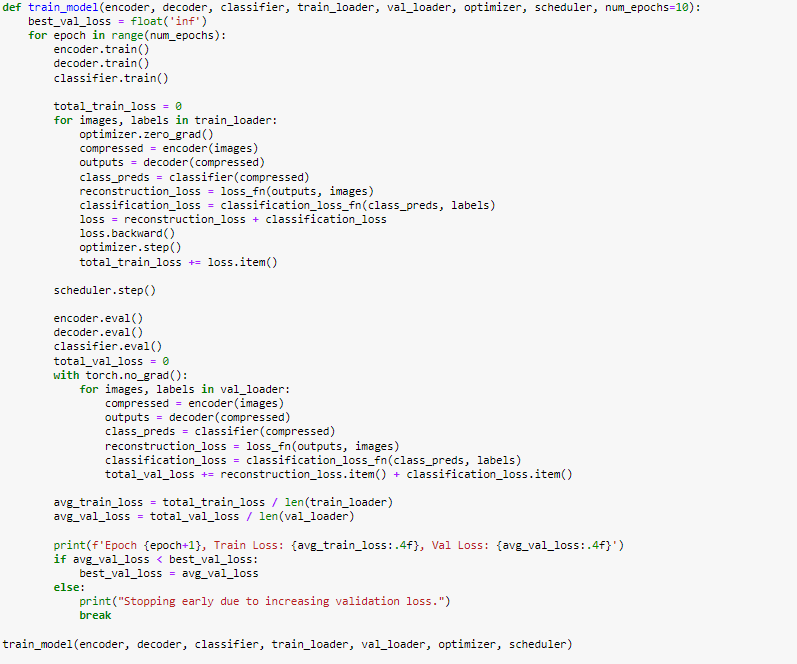
Анализ стоимости и выгод: Уменьшение занимаемого пространства может значительно снизить стоимость хранения данных, особенно при масштабировании для больших наборов данных.

1. **Применение машинного обучения для предиктивного сжатия**

Здесь был представлен процесс обучения нейронной сети (TensorFlow и PyTorch).

Процесс обучения на PyTorch:





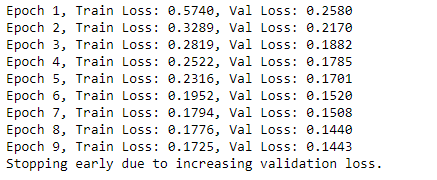
В начале каждой эпохи обнуляются градиенты оптимизатора с помощью optimizer.zero\_grad()

В прямом проходе изображения подаются на вход энкодера, после чего происходит декодирование сжатого представления и предсказание классов. Затем вычисляются функции потерь для восстановления изображения

Далее было добавлено “Расписание обучения с шагом” После каждой эпохи расписание обучения может быть обновлено с помощью scheduler.step(). В данном случае используется шаговое расписание, где каждые step\_size эпох уменьшается скорость обучения в gamma раз.

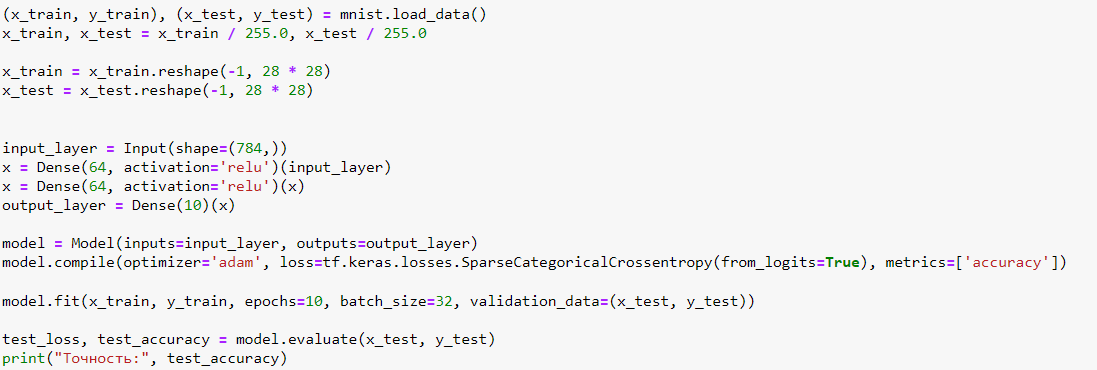
После завершения эпохи модель оценивается на валидационном наборе. Это позволяет оценить ее производительность на данных, которые она ранее не видела, и контролировать переобучение

Информация о каждой эпохе выводится как значение функции потерь на обучающем и валидационном наборах. Если значение функции потерь на валидационном наборе увеличивается или достигает плато, обучение прерывается с выводом сообщения о ранней остановке.



Получили такие результаты и как было сказано выше если значение функции потерь на валидационном наборе увеличивается или достигает плато, обучение прерывается с выводом сообщения о ранней остановке, это сообщение мы и получили.

Процесс обучения на TensorFlow



Определяется архитектура модели: Создается нейронная сеть с использованием функционального API TensorFlow Keras. Далее модель компилируется с использованием оптимизатора Adam, функции потерь SparseCategoricalCrossentropy (так как метки в датасете представлены в виде целых чисел) и метрики accuracy для оценки производительности модели в процессе обучения.

1. **Анализ результатов и формулирование выводов:**

В ходе исследования, была проведена работа с методами сжатия данных. Были применены различные техники обработки и анализа данных, а также использовали алгоритмы машинного обучения для улучшения процессов предварительного анализа данных и построения признаков для эффективного сжатия информации.

Были использованы методы сжатия данных

RLE (Run-Length Encoding), который эффективен для сжатия данных с большими последовательностями повторяющихся элементов.

Графический Анализ Зависимости MSE:

Анализ показал, что увеличение размера блока значительно снижает MSE, что указывает на потенциал для оптимизации качества сжатых изображений при управляемом увеличении размера блока.

Применение Машинного Обучения

Использование PCA (Метод Главных Компонент):

PCA позволил уменьшить размерность данных, что упрощает обработку и анализ, минимизируя при этом потерю информации. Это особенно полезно для визуализации многомерных данных и может способствовать улучшению алгоритмов машинного обучения путём уменьшения "проклятия размерности".

Разработка Нейросетевых Моделей:

Были разработаны модели на основе TensorFlow и PyTorch, которые успешно обучались и тестировались на наборах данных, показывая низкие значения MSE. Это подтверждает возможность использования глубокого обучения для задач предсказания и сжатия данных.

Выводы об использовании машинного обучения

Машинное обучение предоставляет мощные инструменты для анализа и обработки данных, которые могут быть использованы в самых разных задачах сжатия данных. Оно позволяет не только автоматизировать процессы предварительного анализа и сжатия, но и значительно улучшить их эффективность благодаря способности выявлять сложные зависимости в данных. Применение нейронных сетей и алгоритмов машинного обучения позволяет достигать более высокой точности и снижения потерь данных по сравнению с традиционными методами сжатия.

# **Рекомендации для Дальнейших Исследований:**

Исследовать комбинации Методов: Рассмотреть возможности комбинирования различных методов сжатия и машинного обучения для оптимизации результатов сжатия, например, использование PCA для предобработки данных перед применением RLE.

Автоматизация и оптимизация: Разработать автоматизированные системы для выбора оптимальных параметров сжатия, основываясь на исходных характеристиках данных, чтобы максимально увеличить эффективность сжатия без значительной потери качества.

Улучшение нейронных сетей: Продолжить эксперименты с различными архитектурами нейронных сетей и методами обучения, чтобы найти наилучшие конфигурации для конкретных типов данных.

Применение в реальных задачах: Тестировать разработанные модели на реальных наборах данных и в реальных приложениях, таких как видео или аудио обработка, чтобы оценить их практическую применимость и производительность.

В целом, результаты исследования подтверждают, что машинное обучение обладает значительным потенциалом для улучшения процессов сжатия данных. Это открывает перспективы для дальнейших разработок и инноваций в области сжатия и обработки данных.

Подходы по сжатию информации не только помогаюь повышению эффективности сжатия, но и уменьшению потерь информации - это критически важно для современных приложений обработки данных. Результаты исследования свидетельствуют о том, что использование глубокого обучения и других методов машинного обучения заметно улучшает качество сжатия и предоставляет новые возможности для его оптимизации.

Применение классических подходов, таких как RLE и Huffman, показывает старые методы, однако сравнение со моделями машинного обучения демонстрирует перспективы.

# **Литература**

## **Основная литература:**

1. Коротеев, М.В. Технологии анализа данных и машиннон обучение.

Учебное пособие для самостоятельной работы. 1 семестр. / М.В. Коротеев

– М.: Финансовый университет, департамент анализа данных, принятия

решений и финансовых технологий, 2018.

## **Дополнительная литература:**

1. Geron, А. Hands-On Machine Learning with Scikit-Learn, Keras, and

TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems / A.

Geron – O'Reilly Media, Inc., 2019.

2. Coelho, L.P. Building machine learning systems with Python / L.P. Coelho,

W. Richert – Packt Publishing Ltd, 2015.

3. <https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/>

4. Timothy Bell, Ian H. Witten, John G. Clear - "Text Compression" 102-129

5. Д.Сэломон. Сжатие данных. 29-51, 120.