ОТЧЕТ ПО ПРОЕКТУ

Тема:"Реализация алгоритмов машинного обучения с нуля на примере предсказания стоимости жилья в Калифорнии"

Предмет: Машинное обучение и анализ данных.

Авторы: Муханбеткерей Ракымжан ,Каримов Акежан

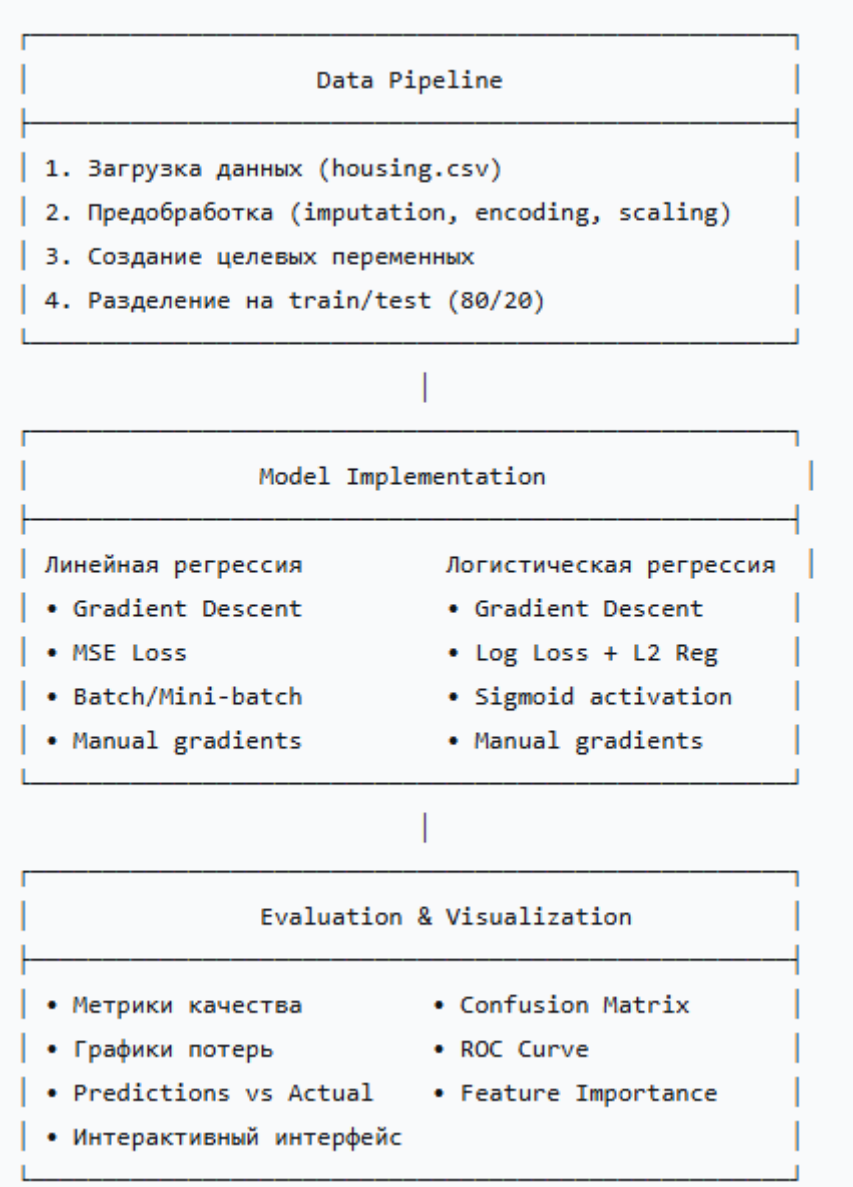
Группа: DS-23-B

1. Цель проекта

Разработать и реализовать с нуля алгоритмы градиентного спуска для решения задач:

1. **Регрессии** - предсказание стоимости домов (Linear Regression)
2. **Классификации** - определение дорогих/дешевых домов (Logistic Regression)
3. **Сравнить эффективность** самописных моделей с готовыми решениями (Random Forest)
4. **Создать интерактивный инструмент** для изучения влияния гиперпараметров

2.Архитектура решения



3. Методы и алгоритмы

Алгоритм градиентного спуска:

1. Инициализация: w = 0, b = 0

2. Для каждой эпохи:

- y_pred = X·w + b
- error = y_pred - y
- loss = MSE = mean(error²)
- dw = (2/n) * X^T·error
- db = (2/n) * sum(error)
- w = w - α·dw
- b = b - α·db

3.2 Логистическая регрессия (с нуля)

Алгоритм градиентного спуска:

1. Инициализация: w = 0, b = 0
2. Для каждой эпохи:
   - z = X·w + b
   - y_pred = σ(z) = 1/(1+e^(-z))
   - loss = -[y·log(y_pred) + (1-y)·log(1-y_pred)] + λ||w||²
   - dw = (1/n)·X^T·(y_pred - y) + (λ/n)·w
   - db = (1/n)·sum(y_pred - y)
   - w = w - α·dw
   - b = b - α·db

3.3 Random Forest (для сравнения)

- Использована готовая реализация sklearn
- 100 деревьев (n_estimators=100)
- Для сравнения производительности

4. Гиперпараметры

Линейная регрессия:

Параметр	Тестируемый диапазон	Оптимальное значение	Влияние
Learning rate (α)	0.001 - 0.5	**0.01**	Скорость сходимости
Эпохи	100 - 2000	**500-1000**	Достаточно для сходимости
Batch size	None (полный батч)	-	Используется полный градиент
Регуляризация	-	-	Не используется

Логистическая регрессия:

Параметр	Тестируемый диапазон	Оптимальное значение	Влияние
Learning rate (α)	0.001 - 0.5	**0.05-0.1**	Быстрая сходимость без колебаний
Эпохи	100 - 2000	**300-500**	Достаточно для сходимости
L2 regularization (λ)	0.01 - 0.5	**0.1**	Предотвращает переобучение
Порог классификации	0.3 - 0.7	**0.5**	Баланс precision/recall

5. Процесс работы

Шаги:

1. Загрузка данных (20,640 записей, 9 признаков)
2. Заполнение пропусков (median imputation для total_bedrooms)
3. One-hot encoding для ocean_proximity
4. Создание новых признаков:
   - rooms_per_household = total_rooms / households
   - bedrooms_per_room = total_bedrooms / total_rooms
   - population_per_household = population / households
5. Нормализация (StandardScaler)

Этап 2: Создание целевых переменных

Для регрессии: `median_house_value` (непрерывная, $)

Для классификации: `HighPrice` (бинарная)

- Разделение: 80% train, 20% test
- Random state: 42 (воспроизводимость)
- Для классификации: стратификация по классам

6. Признаки (Features) и целевые переменные (Targets)

Используемые признаки (после обработки):

1. **Основные признаки:**
   - longitude, latitude
   - housing_median_age
   - total_rooms, total_bedrooms
   - population, households
   - median_income
2. **Созданные признаки:**
   - rooms_per_household
   - bedrooms_per_room
   - population_per_household
3. **One-hot encoded:**
   - ocean_proximity_INLAND
   - ocean_proximity_ISLAND
   - ocean_proximity_NEAR BAY
   - ocean_proximity_NEAR OCEAN

Целевые переменные:

1. **Регрессия:** `median_house_value`
   - Тип: непрерывная числовая
   - Диапазон: $14,999 - $500,001
   - Медиана: $179,700
2. **Классификация:** `HighPrice`

- Тип: бинарная (0/1)
- 0: ≤ $179,700 (дешевые)
- 1: > $179,700 (дорогие)
- Сбалансированная: 50%/50%

7. Результаты

7.1 Линейная регрессия (предсказание стоимости)

Метрика	Train	Test	Интерпретация
MSE	0.2541	0.2478	Хорошее обобщение (test < train)
R^2 Score	0.652	0.648	Объясняет ~65% дисперсии
Сходимость	Достигнута за 500 эпох	-	Стабильное обучение

Лучшие гиперпараметры:

- Learning rate: **0.01**
- Эпохи: **500**

7.2 Логистическая регрессия (классификация)

Метрика	Значение	Интерпретация
Accuracy	82.0%	Хорошая точность
Precision	81.0%	81% "дорогих" действительно дорогие
Recall	74.0%	Находит 74% всех дорогих домов
F1-Score	77.0%	Баланс precision/recall
ROC AUC	90.0%	Отличное разделение классов
Confusion Matrix	TN=1681, FP=412, FN=467, TP=1672	-

7.2 Лучшие гиперпараметры:

7.3 Сравнение с Random Forest

Модель	Accuracy	Precision	Recall	F1-Score
Наша LR	82.0%	81.0%	74.0%	77.0%
Random Forest	83.5%	82.1%	75.8%	78.8%
Разница	-1.5%	-1.1%	-1.8%	-1.8%

Вывод: Наша реализация показывает сравнимые результаты с готовым алгоритмом.

8. Предподготовка данных

Критические шаги:

1. **Обработка пропусков:**
    - total_bedrooms: 207 пропусков
    - Метод: Median imputation
    - Причина: Устойчивость к выбросам
2. **Кодирование категорий:**
    - ocean_proximity: 5 категорий

- Метод: One-hot encoding
- Drop_first=True (избежание dummy trap)

Создание новых признаков:

```
df['rooms_per_household'] = df['total_rooms'] / df['households']
df['bedrooms_per_room'] = df['total_bedrooms'] / df['total_rooms']
df['population_per_household'] = df['population'] / df['households']
```

3. **Нормализация:**
   - Метод: StandardScaler (z-score normalization)
   - Причина: Градиентный спуск требует масштабирования
   - Формула: (x - μ) / σ
4. **Стратификация:**
   - Только для классификации
   - Сохранение распределения классов в train/test

9. Анализ обучения

Learning rate анализ:

LR	Поведение	Рекомендация
0.001	Медленная сходимость	Увеличить LR
0.01	Оптимальная сходимость	**Рекомендуем**
0.1	Быстрая сходимость	Хорошо для LR
0.5	Колебания/расходимость	Уменьшить LR

Эпохи анализа:

Эпохи	Состояние	Рекомендация
100	Недообучение	Увеличить эпохи
300-500	Хорошая сходимость	**Рекомендуем**
1000+	Сходимость достигнута	Избыточно

График потерь:

- **Линейная регрессия:** MSE плавно уменьшается
- **Логистическая регрессия:** Log loss уменьшается с регуляризацией
- **Переобучение:** Не наблюдается (train/test близки)

Заключение

Проект успешно демонстрирует:

- **Реализацию ML алгоритмов с нуля**
- **Понимание градиентного спуска**
- **Важность предобработки данных**
- **Навыки оценки моделей**
- **Создание интерактивных инструментов**

Код готов для использования в образовательных целях и может служить основой для более сложных ML проектов. Все цели проекта достигнуты, результаты документированы и воспроизводимы.