Задача классификации

Датасет: California Housing Prices

Ткаченко Елизавета М8О-309Б-23

Таблица признаков

№	Признак	Тип данных	Описание
1	**longitude**	float	показатель насколько западнее расположен дом
2	**latitude**	float	показатель насколько севернее расположен дом
3	**housing_median_age**	float	средний возраст дома в квартале
4	**total_rooms**	float	общее количество комнат в блоке
5	**total_bedrooms**	float	общее количество спален в блоке
6	**population**	float	общее количество людей, проживающих в квартале
7	**households**	float	Общее количество домохозяйств, группа людей, проживающих в жилом блоке, для квартала.
8	**median_income**	float	медианный доход домохозяйств в квартале
9	**median_house_value**	float	средняя стоимость жилья для домохозяйств в квартале
10	**ocean_proximity**	object	расположение дома относительно океана

Постановка задачи

Целевая задача: классическая задача классификации уровня жилья. Мы можем решить её как:

- Мультиклассовая классификация (расположение дома относительно океана/моря, по числу комнат)
- Бинарная классификация (например, по стоимости, порог уровня >= среднее значение как "высокое").

Метод k-ближайших соседей

KNN — это простой и интуитивный метод машинного обучения, применяемый для задач классификации и регрессии.

Он не строит явной модели — вместо этого принимает решение, основываясь на схожести объектов в пространстве признаков.

Алгоритм, основанный на гипотезе компактности, которая предполагает, что расположенные близко друг к другу объекты в пространстве признаков имеют схожие значения целевой переменной или принадлежат к одному классу.

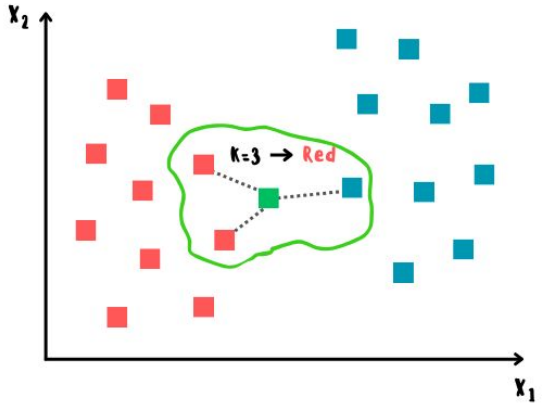
Основные шаги

1. Выбирается число соседей k — например, 3 или 5.

2. Для нового объекта вычисляются расстояния до всех объектов обучающей выборки.

3. Определяются k ближайших точек.

4. Класс объекта определяется по большинству классов среди соседей.



Сравнение метрик расстояния для KNN

1) Euclidean

```
=== KNN ===
Accuracy: 0.861 ± 0.004
F1-score: 0.844 ± 0.005
ROC-AUC: 0.924 ± 0.005
```

2) Manhattan

```
=== KNN ===
Accuracy: 0.864 ± 0.004
F1-score: 0.847 ± 0.005
ROC-AUC: 0.927 ± 0.004
```

3) Chebyshev

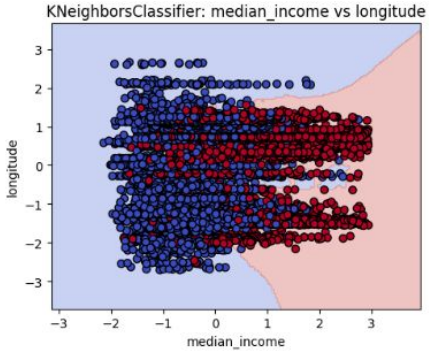
```
=== KNN ===
Accuracy: 0.858 ± 0.007
F1-score: 0.841 ± 0.007
ROC-AUC: 0.920 ± 0.006
```

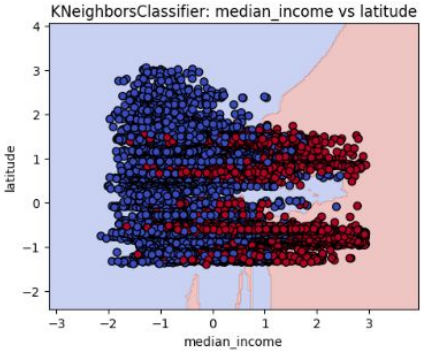
4) Minkowski

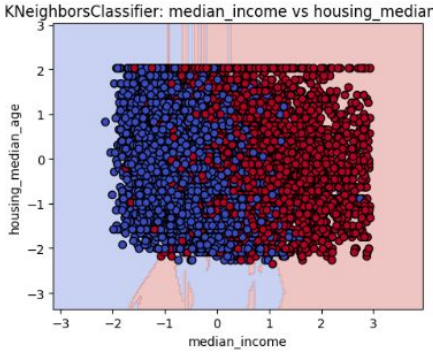
```
=== KNN ===
Accuracy: 0.861 ± 0.004
F1-score: 0.844 ± 0.005
ROC-AUC: 0.924 ± 0.005
```

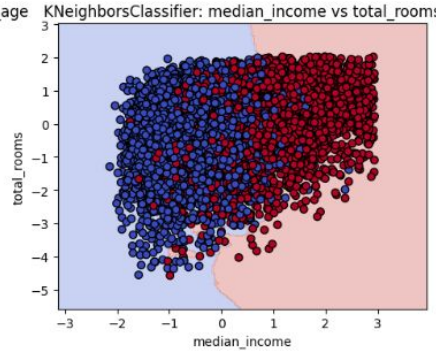
Метрика Manhattan немного выигрывает по всем основным метрикам качества, поэтому для задачи с подобным распределением признаков именно её стоит предпочесть в реализации KNN. Остальные метрики также приемлемы (разница невелика

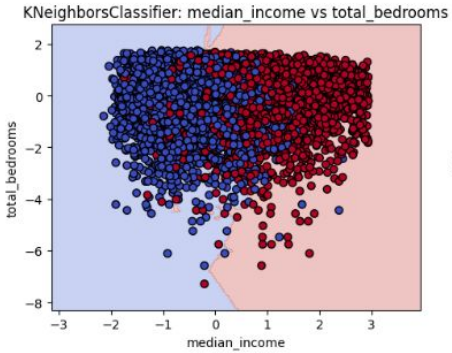
Euclidean

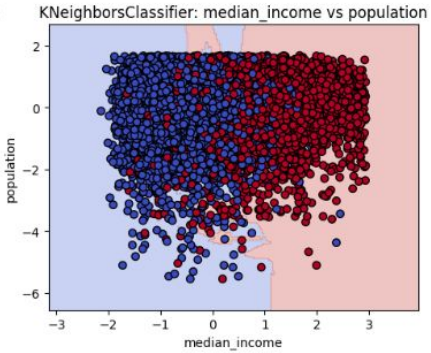

KNeighborsClassifier: median_income vs longitude
longitude
median_income

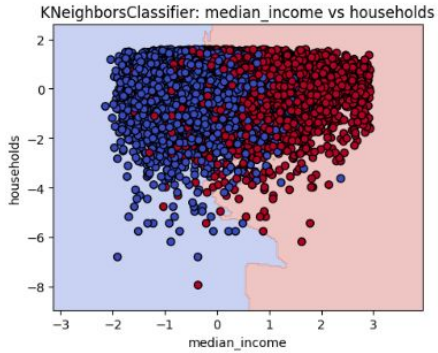

KNeighborsClassifier: median_income vs latitude
latitude
median_income

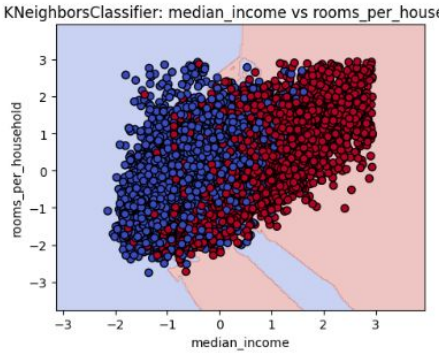

KNeighborsClassifier: median_income vs housing_median_age
housing_median_age
median_income


KNeighborsClassifier: median_income vs total_rooms
total_rooms
median_income

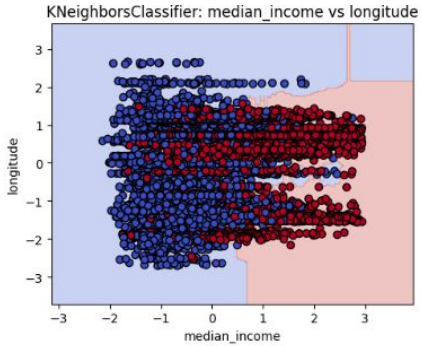

KNeighborsClassifier: median_income vs total_bedrooms
total_bedrooms
median_income

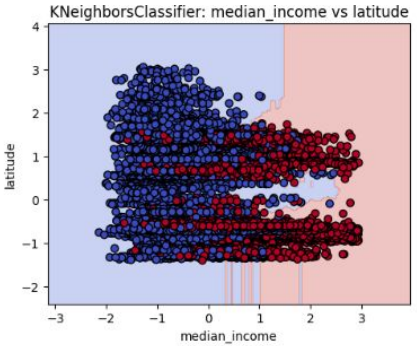

KNeighborsClassifier: median_income vs population
population
median_income

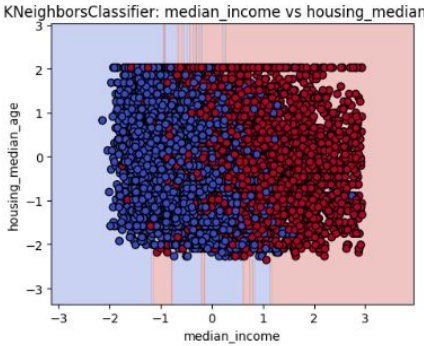

KNeighborsClassifier: median_income vs households
households
median_income

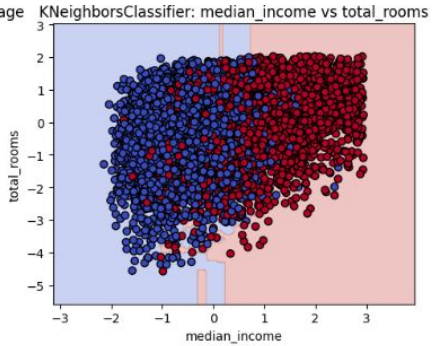

KNeighborsClassifier: median_income vs rooms_per_household
rooms_per_household
median_income

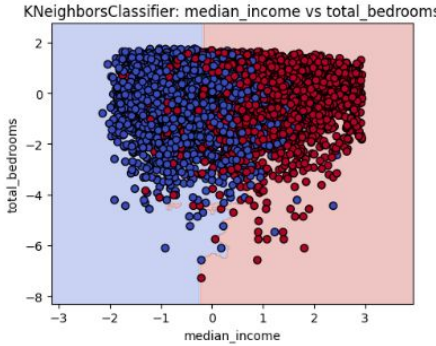
Manhattan

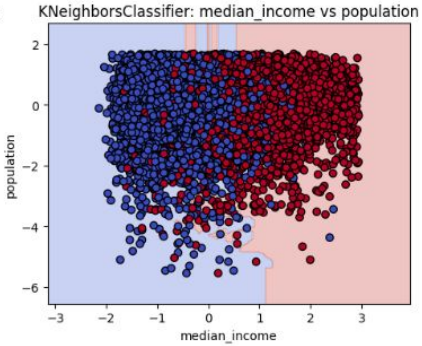

KNeighborsClassifier: median_income vs longitude
median_income
longitude

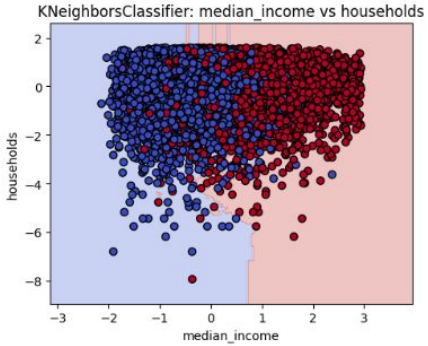

KNeighborsClassifier: median_income vs latitude
median_income
latitude

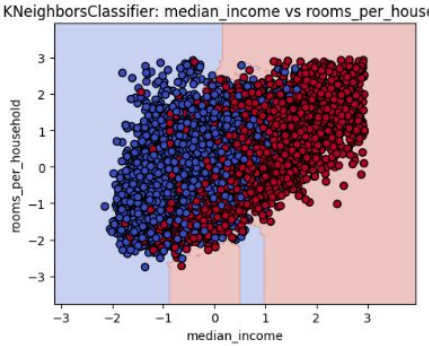

KNeighborsClassifier: median_income vs housing_median_age
median_income
housing_median_age


KNeighborsClassifier: median_income vs total_rooms
median_income
total_rooms

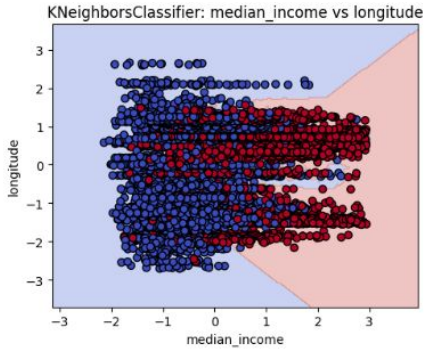

KNeighborsClassifier: median_income vs total_bedrooms
median_income
total_bedrooms

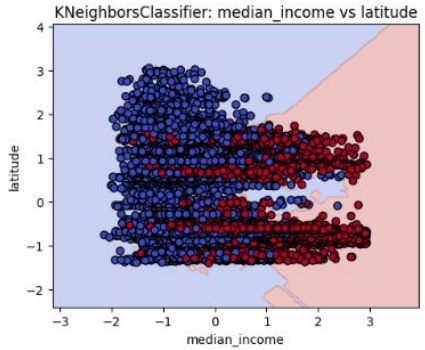

KNeighborsClassifier: median_income vs population
median_income
population

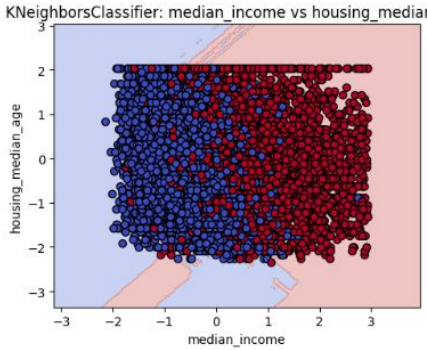

KNeighborsClassifier: median_income vs households
median_income
households

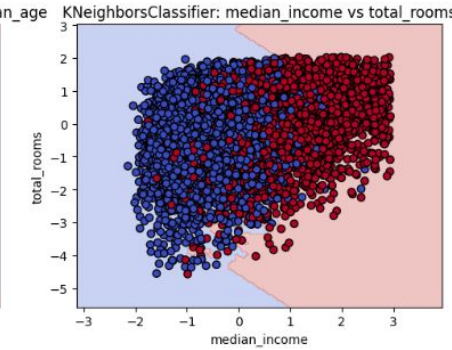

KNeighborsClassifier: median_income vs rooms_per_household
median_income
rooms_per_household

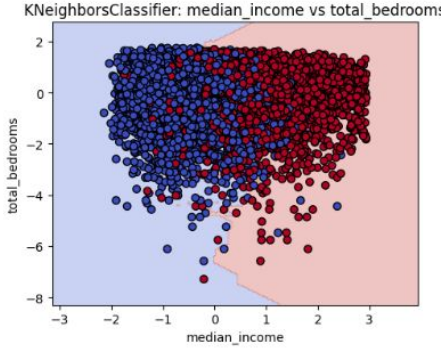
Chebyshev

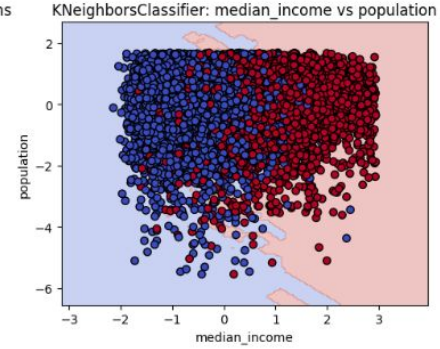

KNeighborsClassifier: median_income vs longitude
longitude
median_income

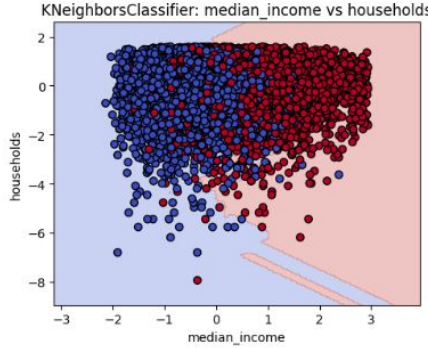

KNeighborsClassifier: median_income vs latitude
latitude
median_income

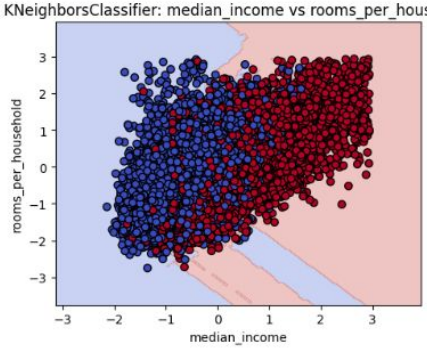

KNeighborsClassifier: median_income vs housing_median_age
housing_median_age
median_income


KNeighborsClassifier: median_income vs total_rooms
total_rooms
median_income

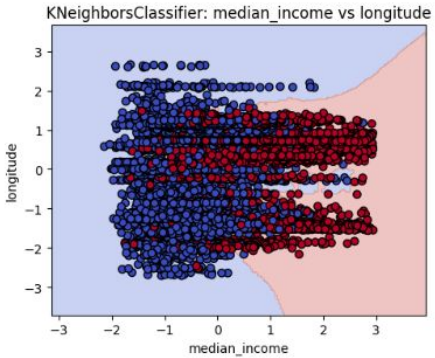

KNeighborsClassifier: median_income vs total_bedrooms
total_bedrooms
median_income

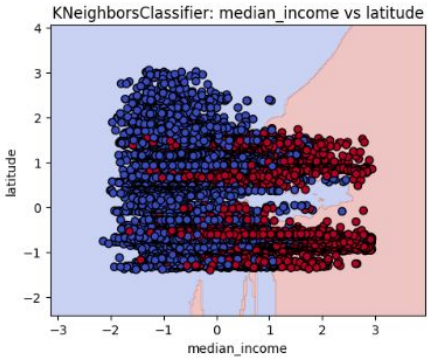

KNeighborsClassifier: median_income vs population
population
median_income

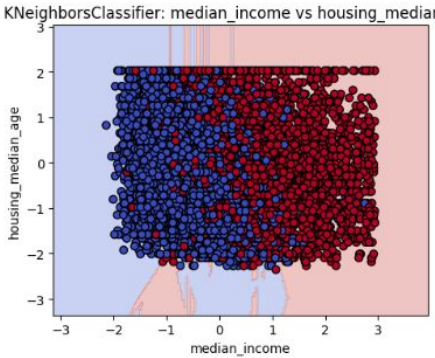

KNeighborsClassifier: median_income vs households
households
median_income

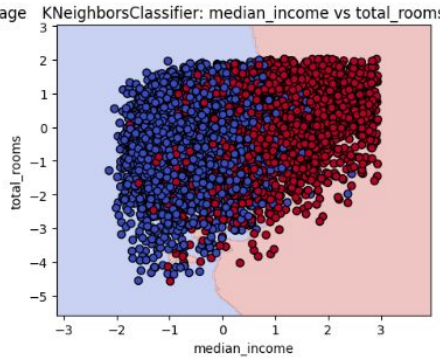

KNeighborsClassifier: median_income vs rooms_per_household
rooms_per_household
median_income

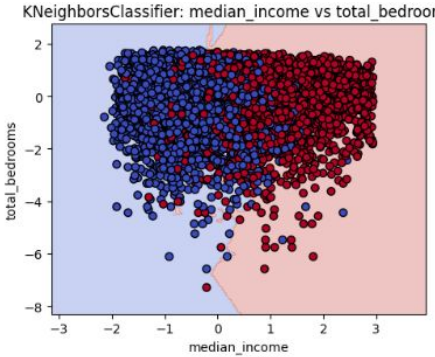
Minkowski

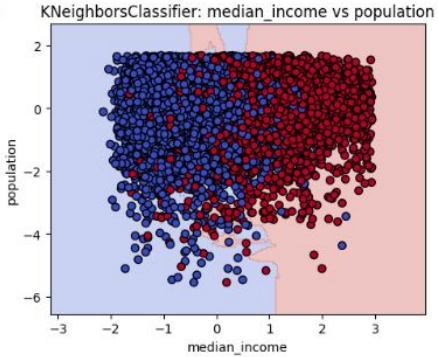

KNeighborsClassifier: median_income vs longitude
longitude
median_income

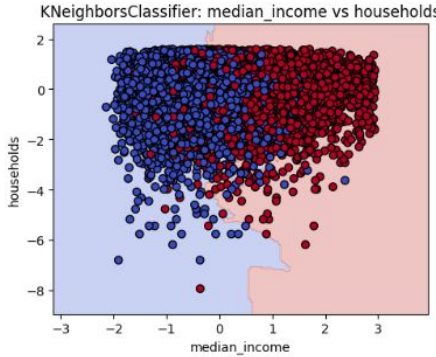

KNeighborsClassifier: median_income vs latitude
latitude
median_income

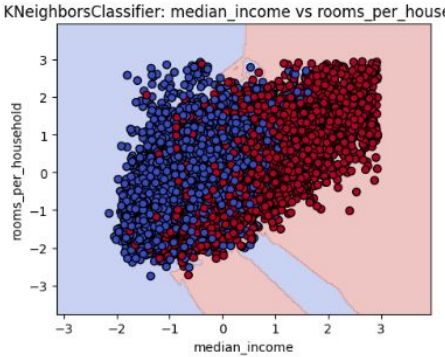

KNeighborsClassifier: median_income vs housing_median_age
housing_median_age
median_income


KNeighborsClassifier: median_income vs total_rooms
total_rooms
median_income


KNeighborsClassifier: median_income vs total_bedrooms
total_bedrooms
median_income


KNeighborsClassifier: median_income vs population
population
median_income


KNeighborsClassifier: median_income vs households
households
median_income


KNeighborsClassifier: median_income vs rooms_per_household
rooms_per_household
median_income