УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Системы искусственного интеллекта

Лабораторная работа №4

Линейная регрессия

Выполнила Громилова Мария Дмитриевна,

группа P33311

Преподаватель Кугаевских Александр Владимирович

г. Санкт-Петербург 2023 г

Задание

- Получите и визуализируйте статистику по датасету (включая количество, среднее значение, стандартное отклонение, минимум, максимум и различные квантили).
- Проведите предварительную обработку данных, включая обработку отсутствующих значений, кодирование категориальных признаков и нормировка.
- Разделите данные на обучающий и тестовый наборы данных.
- Реализуйте линейную регрессию с использованием метода наименьших квадратов без использования сторонних библиотек, кроме NumPy и Pandas. Использовать минимизацию суммы квадратов разностей между фактическими и предсказанными значениями для нахождения оптимальных коэффициентов.
- Постройте **три модели** с различными наборами признаков.
- Для каждой модели проведите оценку производительности, используя метрику коэффициент детерминации, чтобы измерить, насколько хорошо модель соответствует данным.
- Сравните результаты трех моделей и сделайте выводы о том, какие признаки работают лучше всего для каждой модели.
- Бонусное задание
    - Ввести синтетический признак при построении модели

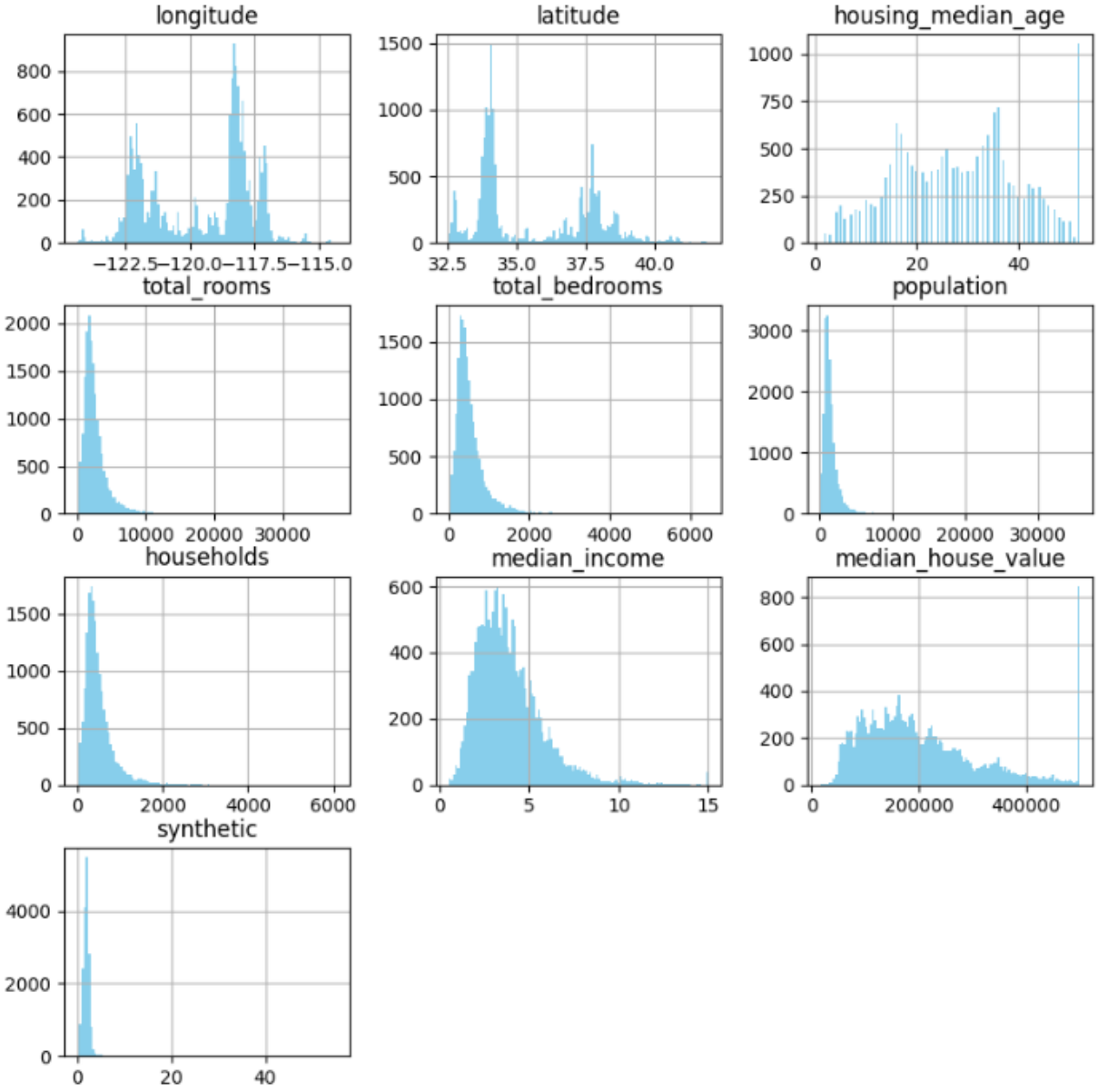
Решение

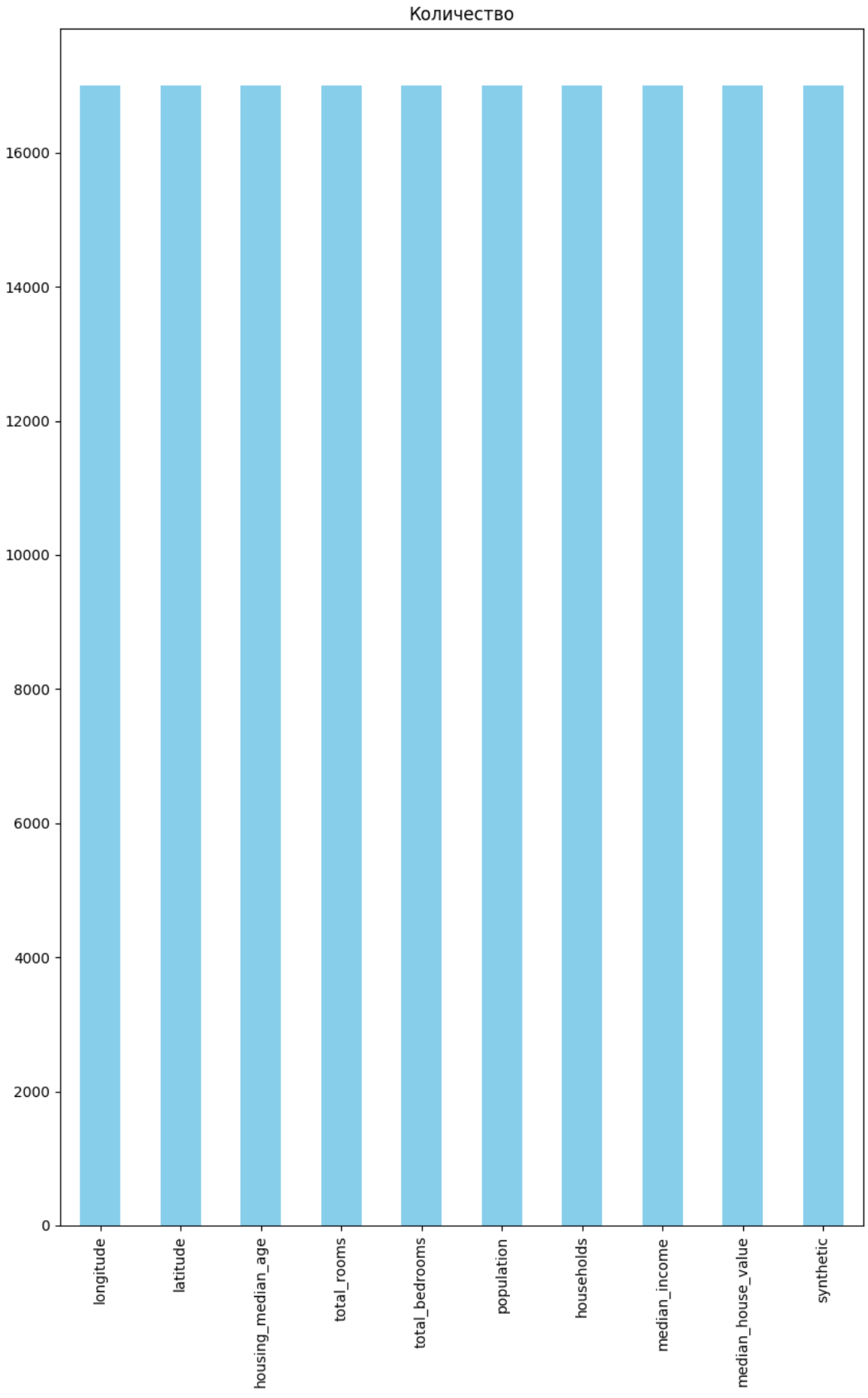
```
# Получите статистику по датасету
pd.set_option('display.max_columns', None)
df = pd.read_csv('california_housing_train.csv')
summary_stats = df.describe()
df['synthetic'] = df['total_rooms'] / df['population']

#Визуализируйте статистику по датасету
visAll(df)
# Визуализация количества
visCount(plt, df)
# Визуализация средних значений
visMean(plt, df)
# Визуализация стандартных отклонений
visStd(plt, df)
# Визуализация минимума
visMin(plt, df)
# Визуализация максимума
visMax(plt, df)
```

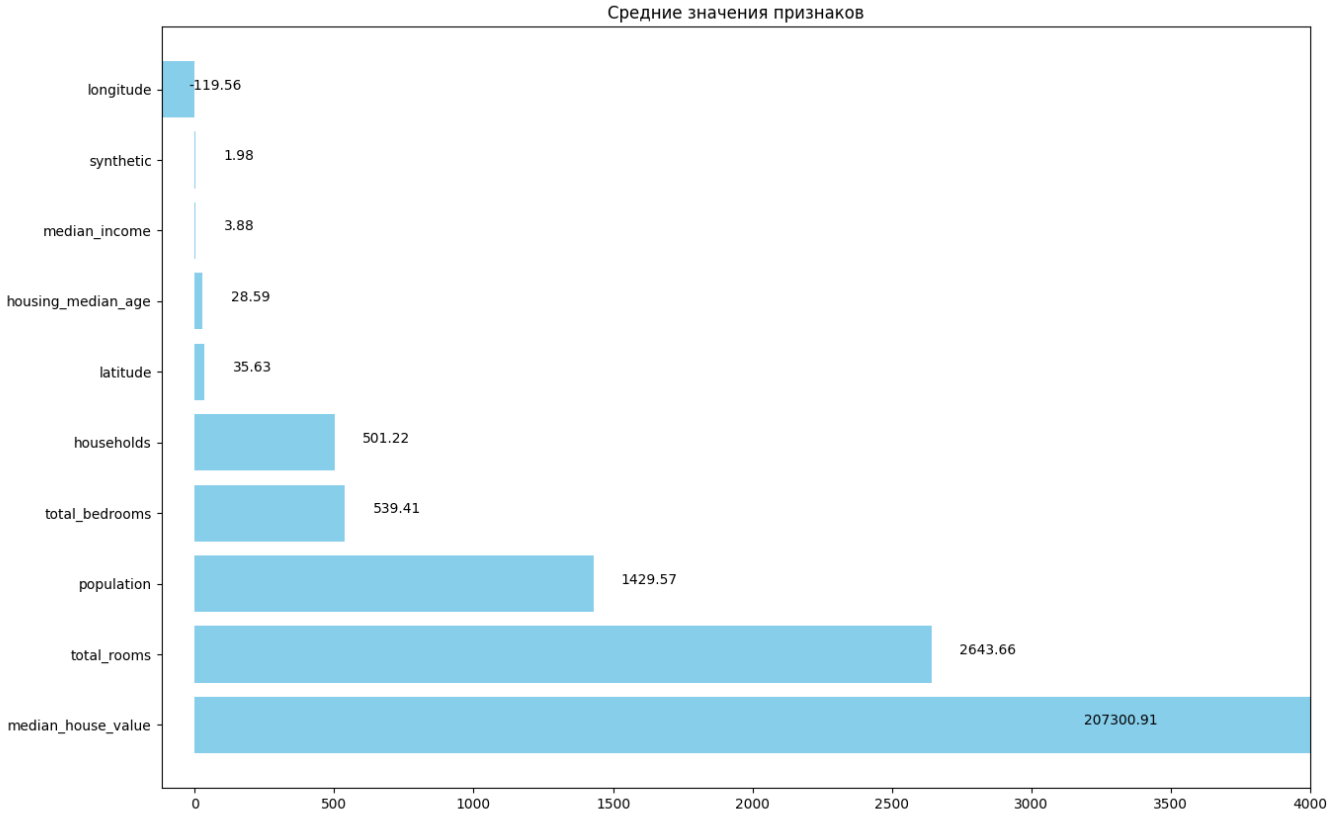
Пример функции для визуализации

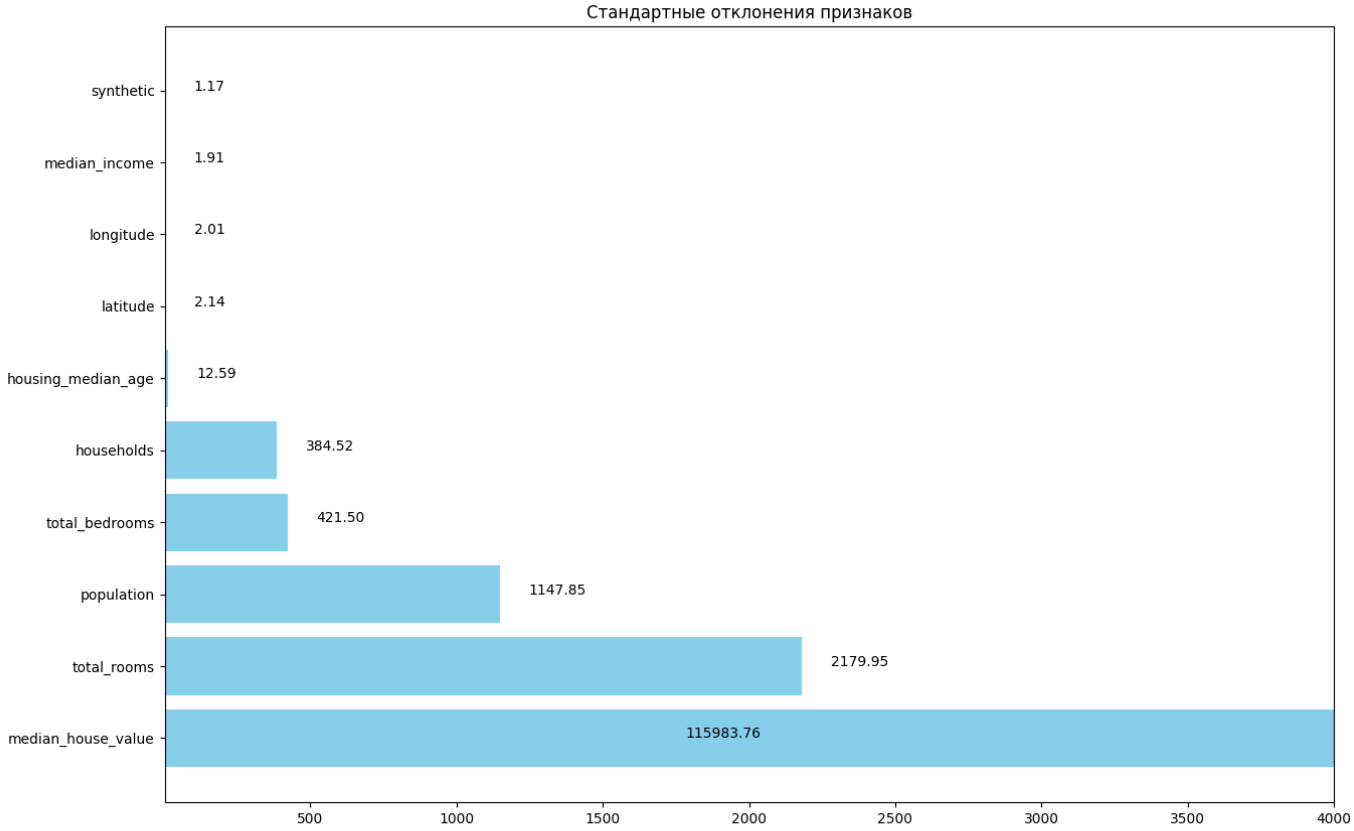
```
def visMean(plt, df):
    plt.figure(figsize=(15, 10))
    plt.title('Средние значения признаков')
    dfMeans = df.mean().sort_values(ascending=False)
    plt.barh(dfMeans.index, dfMeans.values)
    plt.xlim(dfMeans.min(), 4000)
    addTexit(dfMeans, k: 65)
    plt.show()
```

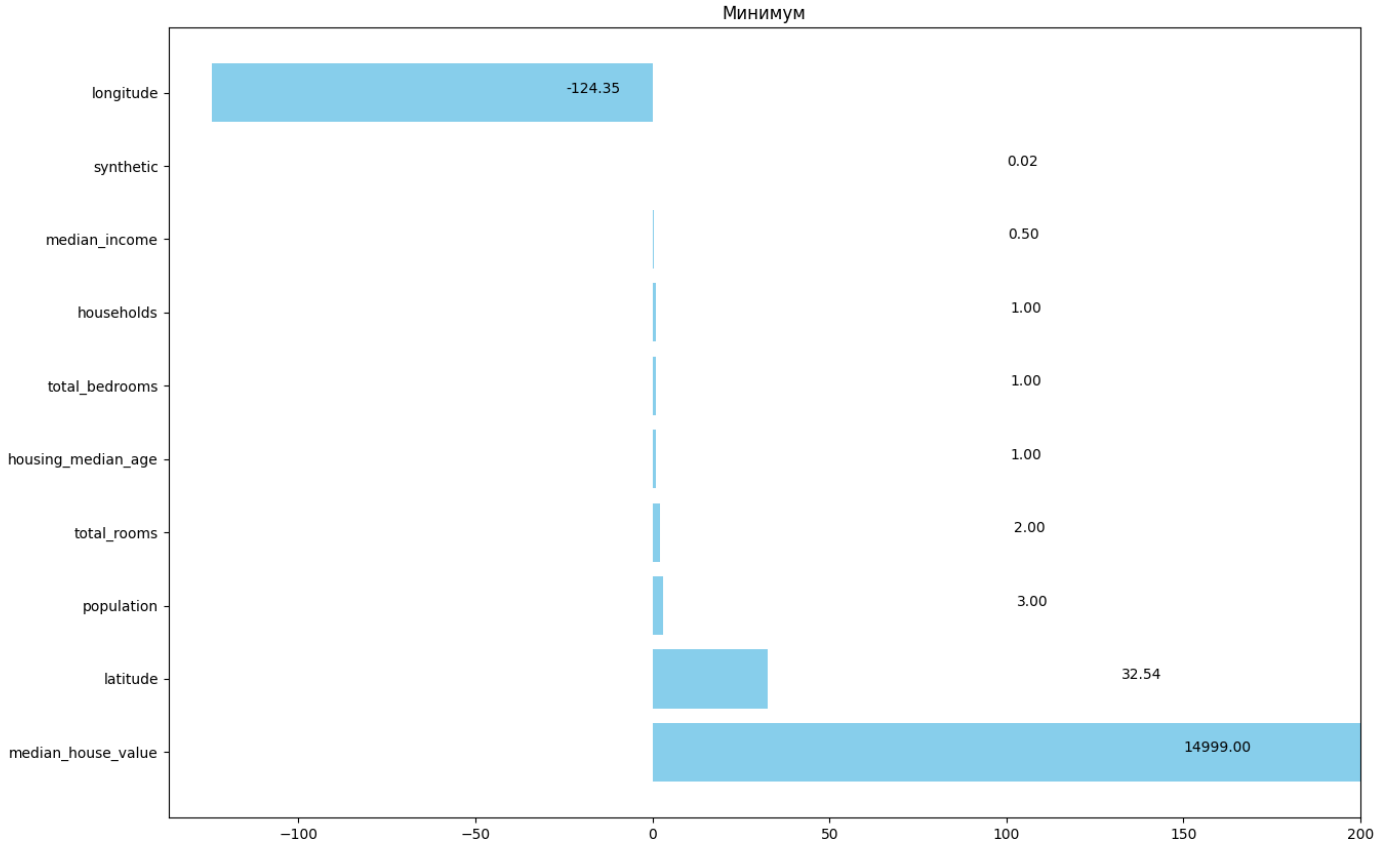
Визуализация статистики всех числовых признаков

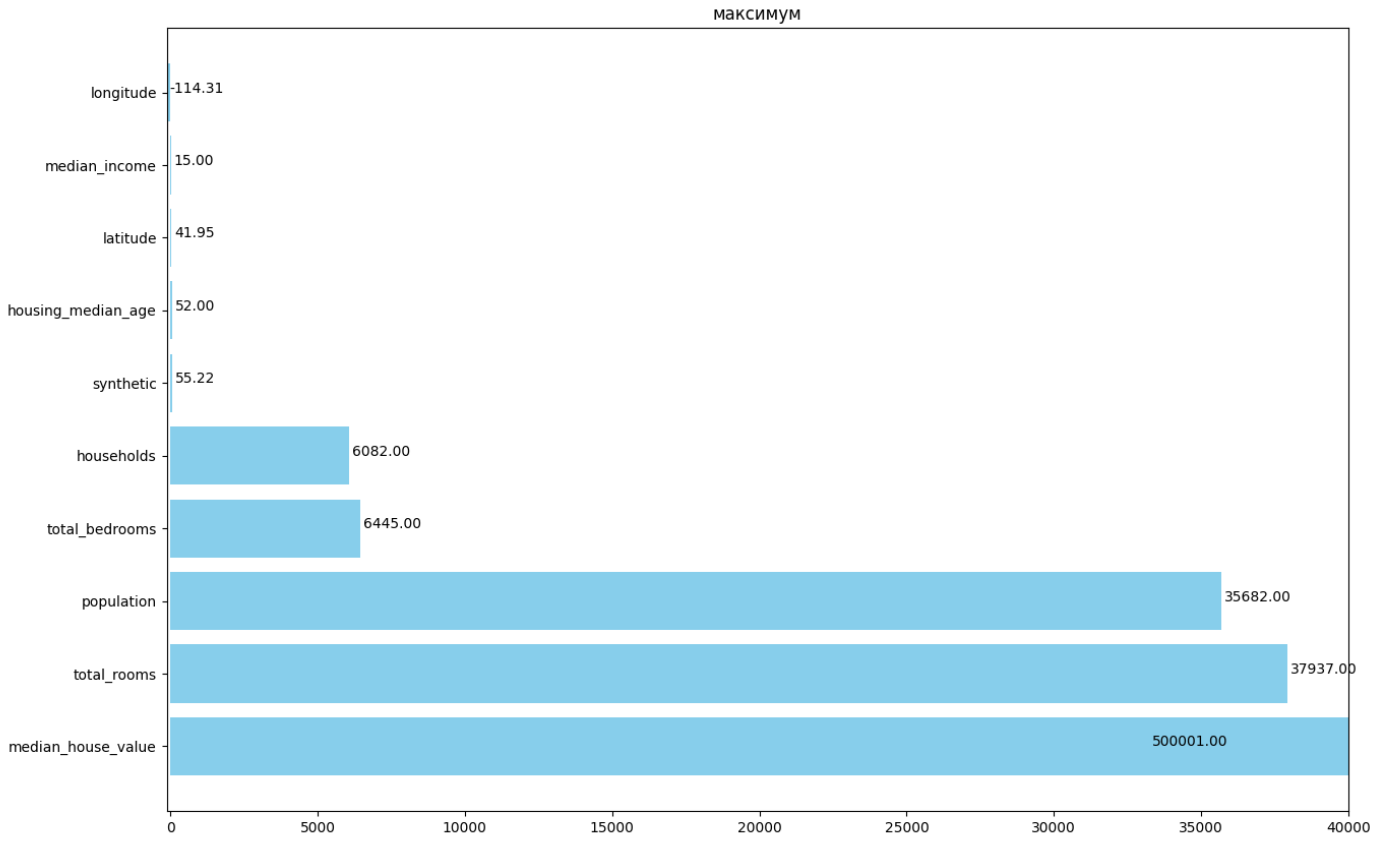



Количество
16000
14000
12000
10000
8000
6000
4000
2000
0
longitude
latitude
housing_median_age
total_rooms
total_bedrooms
population
households
median_income
median_house_value
synthetic


Средние значения признаков
longitude
synthetic
median_income
housing_median_age
latitude
households
total_bedrooms
population
total_rooms
median_house_value
-119.56
1.98
3.88
28.59
35.63
501.22
539.41
1429.57
2643.66
207300.91
0
500
1000
1500
2000
2500
3000
3500
4000


Стандартные отклонения признаков
synthetic
median_income
longitude
latitude
housing_median_age
households
total_bedrooms
population
total_rooms
median_house_value
1.17
1.91
2.01
2.14
12.59
384.52
421.50
1147.85
2179.95
115983.76
500
1000
1500
2000
2500
3000
3500
4000





Проводим обработку отсутствующих значений, категориальных признаков в этом наборе нет

Добавим синтетический признак – отношение количества комнат к количеству проживающих

```
df['synthetic'] = df['total_rooms'] / df['population']
```

mean – среднее значение

std – стандартное отклонение

```
# Проведите предварительную обработку данных
df.fillna(df.mean(), inplace=True)
numeric_columns = ['longitude', 'latitude', 'housing_median_age', 'total_rooms', 'total_bedrooms', 'population',
                   'households', 'median_income', 'synthetic']

# Примените стандартизацию к каждому числовому признаку
for column in numeric_columns:
    mean = df[column].mean()
    std = df[column].std()
    df[column] = (df[column] - mean) / std
```

Разделите данные на обучающий и тестовый наборы данных. Средняя стоимость дома – целевая переменная

```
# Определите признаки (X) и целевую переменную (y)
X = df.drop(columns=['median_house_value'])  # Признаки, исключая 'median_house_value'
y = df['median_house_value']  # Целевая переменная 'median_house_value'
# Разделите данные на обучающий и тестовый наборы данных
X_train, X_test, y_train, y_test = train_test_split( *arrays: X, y, test_size=0.2, random_state=42)
```

Сначала добавляем столбец с единицами для свободного члена. Используем формулу $w=(X^TX)^{-1}X^Ty$

```
# Реализуйте линейную регрессию с использованием метода наименьших квадратов
4 usages   manteei
def linear_regression(X, y):
    # Добавляем столбец с единицами для учёта свободного члена (w0)
    X = np.column_stack((np.ones(X.shape[0]), X))
    # Вычисляем оптимальные веса с использованием формулы
    weights = np.linalg.inv(X.T @ X) @ X.T @ y
    return weights
```

Полученные оптимальные коэффициенты

```
Оптимальные коэффициенты:
longitude: 206920.88734156967
latitude: -90794.09580471492
housing_median_age: -96986.584341742
total_rooms: 14139.769416506562
total_bedrooms: -25806.545624451464
population: 26509.63805826008
households: -33820.85525627581
median_income: 38160.496265618516
synthetic: 74034.48303438685
median_house_value 13081.948767550797
```

Полученные предсказания:

```
Предсказание для основной модели
      Тестовые значения  Полученные значения
0              142700.0        144947.667450
1              500001.0        397045.864169
2               61800.0         84465.258490
3              162800.0        149321.441852
4               90600.0        146980.141223
...                 ...                  ...
3395           211400.0        176212.328914
3396           500001.0        395912.884232
3397           162500.0         11765.193274
3398           360700.0        285210.335016
3399           137500.0        108567.985769
```

Коэффициент детерминации для основной модели: 0.6672346781222387

Создаем 3 новые модели

```
features_set1 = ['longitude', 'latitude', 'housing_median_age', 'median_income']
features_set2 = ['synthetic', 'total_rooms', 'total_bedrooms', 'households', 'median_income']
features_set3 = ['synthetic', 'households', 'median_income']
```

```
Коэффициент детерминации для модели 1: 0.6137931108803529
Коэффициент детерминации для модели 2: 0.534104531869086
Коэффициент детерминации для модели 3: 0.5025299534167378
```

Вывод:

Полученный коэффициент детерминации говорит о том, что почти 70% вариаций в целевой переменной объясняется основной моделью. Для данной задачи это хороший результат.

Для основной модели можно составить следующий рейтинг признаков(от лучших к худшим):

median_income, synthetic, total_rooms, housing_median_age, households, total_bedrooms, longitude, latitude, population

Для первой модели: median_income, housing_median_age, longitude, latitude

Для второй модели: median_income, synthetic, total_rooms, households, total_bedrooms

Для третьей модели: median_income, synthetic, households, population

Получается синтетический признак (отношение количества комнат к количеству проживающих) получился довольно удачным.

Цены очень зависят от среднего дохода, количества комнат и новизны дома и почти не зависят от численности населения, широты и долготы.