МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**Отчёт по лабораторной работе №3**

по дисциплине

«Системы искусственного интеллекта»

***Выполнил:*** Студент группы P3334 Баянов Равиль Динарович

***Преподаватель:***

Авдюшина А. Е.

Санкт-Петербург, 2024

**Оглавление**

[**Задание** 3](#_Toc179498469)

[**Статистика по датасету** 4](#_Toc179498470)

[**Код на Python реализации модели** 6](#_Toc179498471)

[**Показатели детерминации модели** 8](#_Toc179498472)

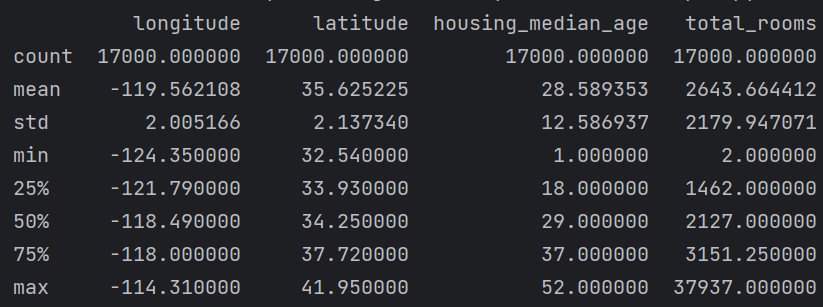
[**Вывод** 9](#_Toc179498473)

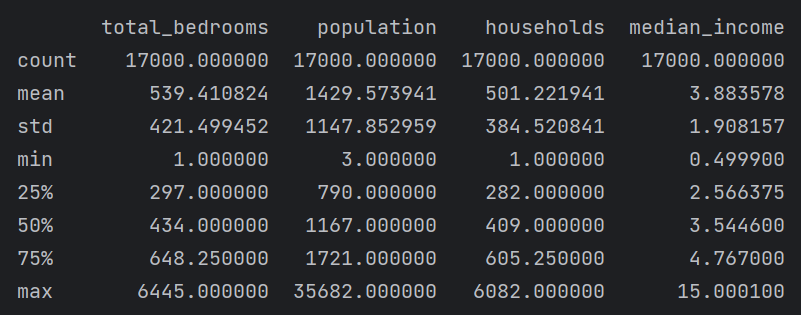
**Задание**

Вариант: набор данных о жилье в Калифорнии, так как номер в группе 4.

* Получите и визуализируйте (графически) 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 (для использования коэффициентов использовать библиотеки тоже нельзя). 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

**Статистика по датасету**





Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Где:

* count – кол-во значений
* mean – среднее
* std – стандартное отклонение (среднеквадратическое отклонение от мат. ожидания)
* min – минимум
* max – максимум
* Значения процентов - квантили столбцов

Графики, описывающие датасет:

Изображение выглядит как текст, диаграмма, График, План

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, диаграмма, Прямоугольник

Автоматически созданное описание

**Код на Python реализации модели**

|  |
| --- |
| **import** pandas **as** pd  **import** matplotlib.pyplot **as** plt  **import** seaborn **as** sns  **import** numpy **as** np  **from** sklearn.model\_selection **import** train\_test\_split  # Функция расчёта линейной регрессии  **def** **linear\_regression**(X, y):  X\_tran = X.T  **return** np.linalg.inv(X\_tran.dot(X)).dot(X\_tran).dot(y)  # Функция расчёта коэфициента детерминации для модели  **def** **coefficient\_of\_determination**(X, y, odds):  y\_pred = X.dot(odds)  sse = np.sum((y - y\_pred) \*\* 2)  sst = np.sum((y - np.mean(y)) \*\* 2)  **return** 1 - sse / sst  data = pd.read\_csv('data/california\_housing\_train.csv')  pd.set\_option('display.max\_columns', None)  stats = data.describe()  print(stats)  # Отрисовка диаграмм распределений каждого столбца из датасета  data.hist(bins=50, figsize=(15, 10))  plt.suptitle('Гистограммы распределения данных')  plt.show()  # Отрисовка диаграммы "ящик с усами" для каждого столбца из датасета  plt.figure(figsize=(15, 10))  sns.boxplot(data=data)  plt.title('Боксплоты данных')  plt.show()  # Удаление отсутствующих значений  data.dropna()  # Нормализация данных  data = (data - data.mean()) / data.std()  # Добавление синтетического признака  data['total\_rooms\_without\_bedrooms'] = data['total\_rooms'] - data['total\_bedrooms']  # Поделим даные на тестовый и обучающий наборы данных  train\_data, test\_data = train\_test\_split(data, test\_size=0.2, random\_state=30)  # Создадим первую модель с небольшим набором признаков  X\_train\_1 = train\_data[['total\_rooms', 'median\_income']].values  y\_train\_1 = train\_data['median\_house\_value'].values  # Добавление столбца для смещения (добавляем столбец с единицами)  X\_train\_1 = np.c\_[np.ones(X\_train\_1.shape[0]), X\_train\_1]  # Получение коэффициентов  odds\_1 = linear\_regression(X\_train\_1, y\_train\_1)  # Получение коэффициента детерминации для оценки модели 1  r\_2\_1 = coefficient\_of\_determination(X\_train\_1, y\_train\_1, odds\_1)  print(f'R^2 для первой модели: {r\_2\_1}')  # Создадим вторую модель с чуть большим набором признаков  X\_train\_2 = train\_data[['total\_rooms', 'median\_income', 'housing\_median\_age']].values  y\_train\_2 = train\_data['median\_house\_value'].values  # Добавление столбца для смещения (добавляем столбец с единицами)  X\_train\_2 = np.c\_[np.ones(X\_train\_2.shape[0]), X\_train\_2]  # Получение коэффициентов  odds\_2 = linear\_regression(X\_train\_2, y\_train\_2)  # Получение коэффициента детерминации для оценки модели 2  r\_2\_2 = coefficient\_of\_determination(X\_train\_2, y\_train\_2, odds\_2)  print(f'R^2 для второй модели: {r\_2\_2}')  # Создадим третью модель с этими же коэффициентами + синтетический признак  X\_train\_3 = train\_data[['total\_rooms', 'median\_income', 'housing\_median\_age', 'total\_rooms\_without\_bedrooms']].values  y\_train\_3 = train\_data['median\_house\_value'].values  # Добавление столбца для смещения (добавляем столбец с единицами)  X\_train\_3 = np.c\_[np.ones(X\_train\_3.shape[0]), X\_train\_3]  # Получение коэффициентов  odds\_3 = linear\_regression(X\_train\_3, y\_train\_3)  # Получение коэффициентов детерминации для оценки модели 3  r\_2\_3 = coefficient\_of\_determination(X\_train\_3, y\_train\_3, odds\_3)  print(f'R^2 для третьей модели: {r\_2\_3}')  print(  f"Сравнение моделей:\nМодель 1 R^2: {r\_2\_1}\nМодель 2 R^2: {r\_2\_2}\nМодель 3 R^2: {r\_2\_3} (с синтетическим признаком)")  # А сейчас протестируем наши модели, которые мы построили  # 1  X\_test\_1 = test\_data[['total\_rooms', 'median\_income']].values  y\_test\_1 = test\_data['median\_house\_value'].values  X\_test\_1 = np.c\_[np.ones(X\_test\_1.shape[0]), X\_test\_1]  r\_2\_1\_test = coefficient\_of\_determination(X\_test\_1, y\_test\_1, odds\_1)  print(f'R^2 для тестовых данных первой модели: {r\_2\_1\_test}')  # 2  X\_test\_2 = test\_data[['total\_rooms', 'median\_income', 'housing\_median\_age']].values  y\_test\_2 = test\_data['median\_house\_value'].values  X\_test\_2 = np.c\_[np.ones(X\_test\_2.shape[0]), X\_test\_2]  r\_2\_2\_test = coefficient\_of\_determination(X\_test\_2, y\_test\_2, odds\_2)  print(f'R^2 для тестовых данных второй модели: {r\_2\_2\_test}')  # 3  X\_test\_3 = test\_data[['total\_rooms', 'median\_income', 'housing\_median\_age', 'total\_rooms\_without\_bedrooms']].values  y\_test\_3 = test\_data['median\_house\_value'].values  X\_test\_3 = np.c\_[np.ones(X\_test\_3.shape[0]), X\_test\_3]  r\_2\_3\_test = coefficient\_of\_determination(X\_test\_3, y\_test\_3, odds\_3)  print(f'R^2 для тестовых данных третьей модели: {r\_2\_3\_test}')  print(  f"Сравнение моделей на тестовых данных:\nМодель 1 R^2: {r\_2\_1\_test}\nМодель 2 R^2: {r\_2\_2\_test}\nМодель 3 R^2: {r\_2\_3\_test} (с синтетическим признаком)") |

**Показатели детерминации модели**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Вывод**

В этой лабораторной работе мы рассмотрели метод машинного обучения – линейная регрессия. Узнали, как с помощью тестового набора данных можно спрогнозировать последующие данные и оценили качество, построенных моделей с помощью коэффициента детерминации.