**Отчет по модели регрессии цены недвижимости в Калифорнии.**

**[1]Нургалиев И.И.**

*[1]АНО ВО «Университет Иннополис», Иннополис, e-mail: i.nurgaliev@innopolis.ru*

| Ссылка на github | https://github.com/ILDAR9/california |
|---|---|

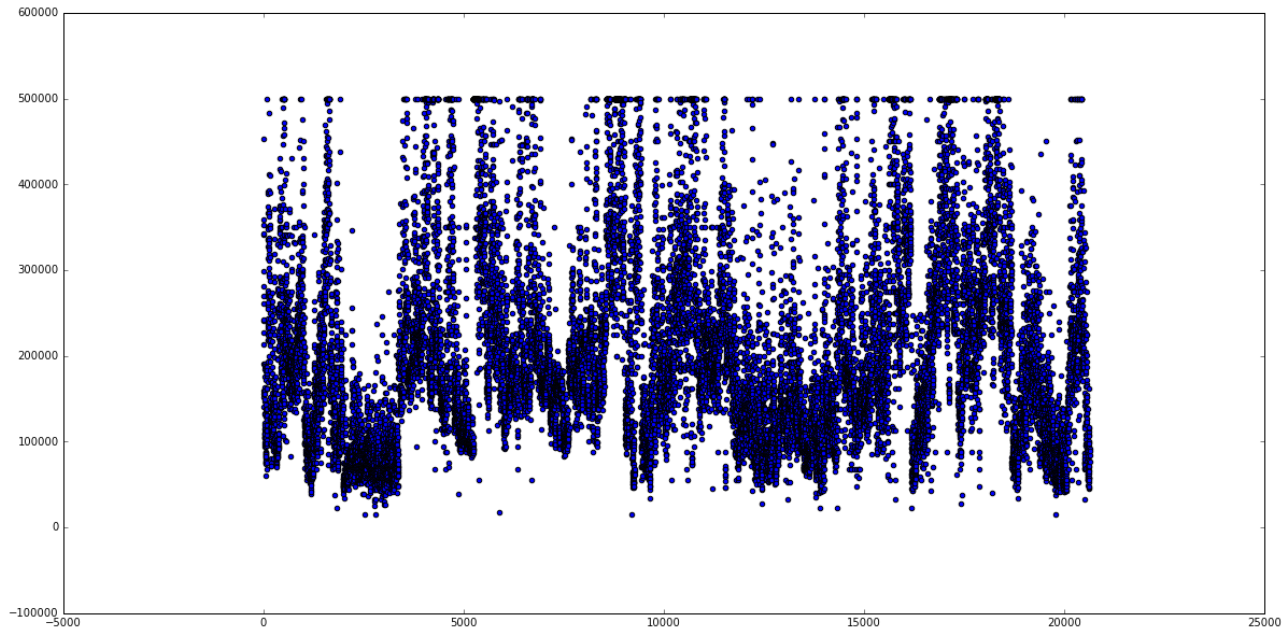
Исходной задачей было создать модель регрессии, позволяющую предсказывать стоимость дома в Калифорнии.

## Входные данные

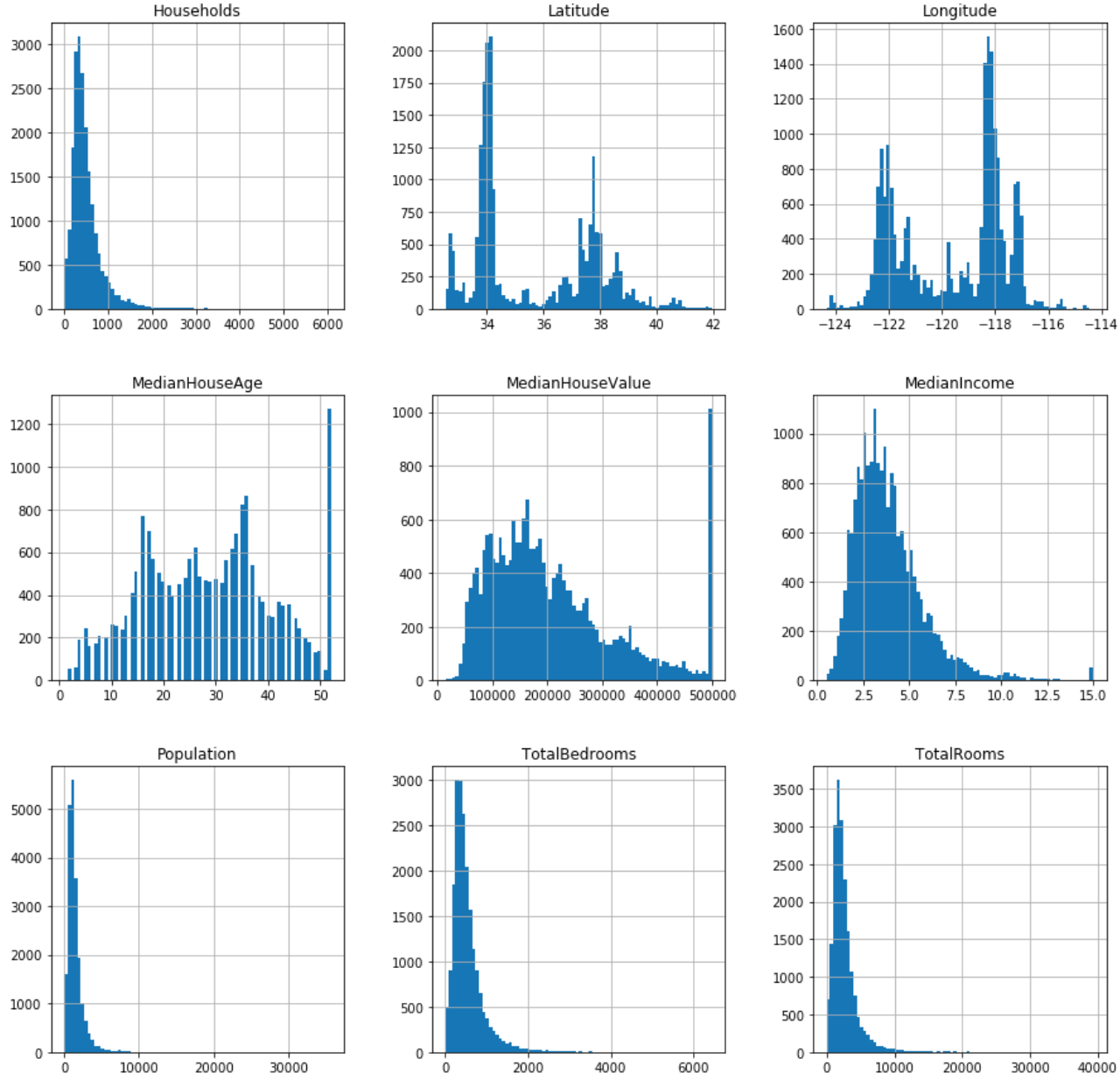
Исходный набор данных содержит 20,640 записей каждая из которых представлена 9 переменными. Зависимая переменная (целевая) median house value (цена дома).

| Переменная | кол-во | $\mu$ | $\sigma$ | Min | Max |
|---|---|---|---|---|---|
| median house value | 20640 | 206855.816909 | 115395.615874 | 14999.00 | 500001.00 |
| median income | 19833 | 3.866405 | 1.897745 | 0.4999 | 15.0001 |
| housing median age | 20640 | 28.639486 | 12.585558 | 1.00 | 52.00 |
| total rooms | 19809 | 2638.515220 | 2184.459291 | 2.00 | 39320.00 |
| total bedrooms | 20640 | 537.898014 | 421.247906 | 1.00 | 6445.00 |
| population, households | 20640 | 1425.476744 | 1132.462122 | 3.00 | 35682.00 |
| households | 20640 | 499.539680 | 382.329753 | 1.00 | 6082.00 |
| latitude | 20640 | 35.63186 | 2.135952 | 32.54 | 41.95 |
| longitude | 20640 | 119.569704 | 2.003532 | 124.35 | -114.31 |

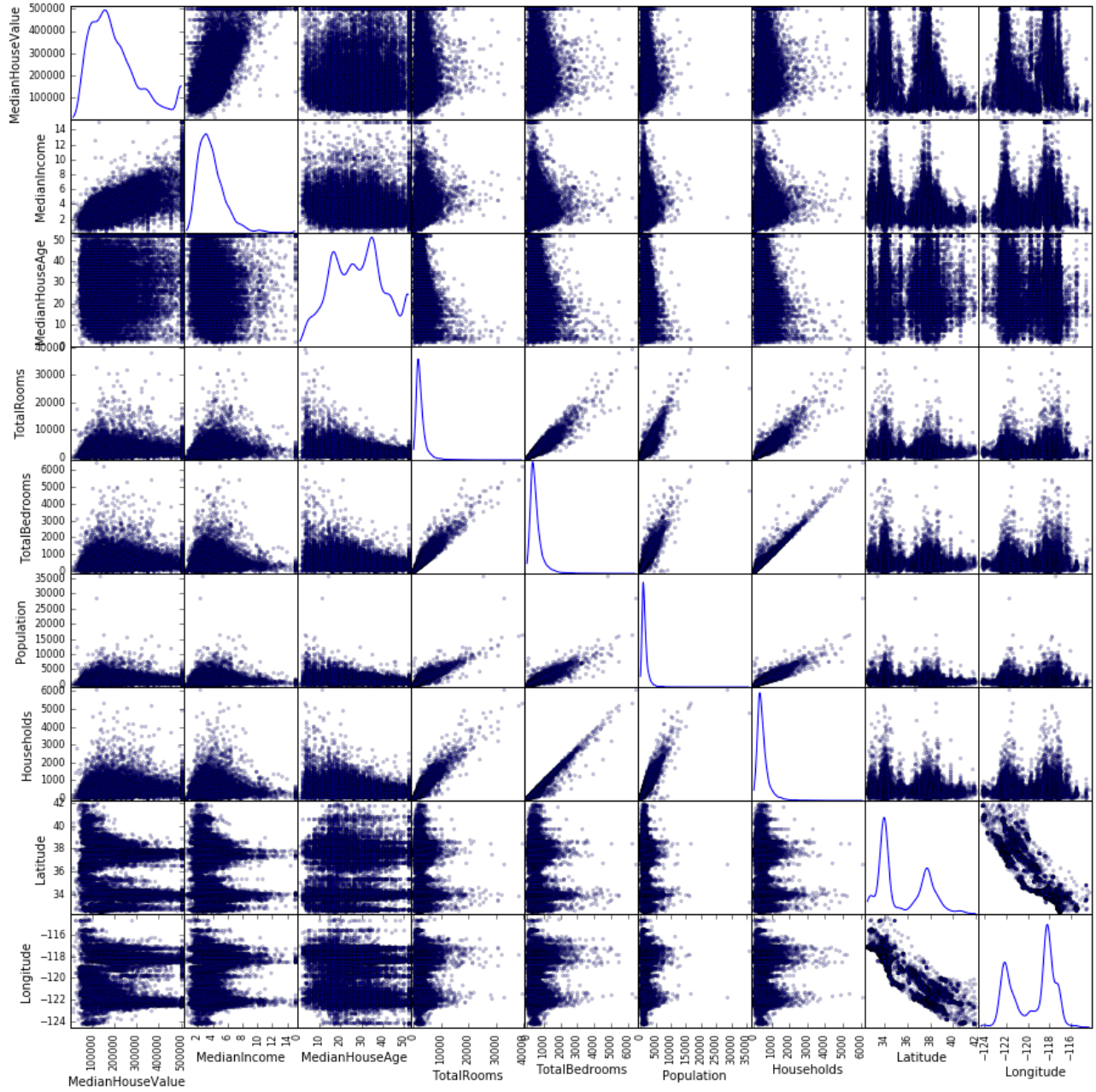
## Анализ данных




Цены домов по файлу расположены достаточно равномерно таким образом мы можем использовать kfold с 30% тестовой выборки отбирают по uniform распределению.




1) У MedianHouseAge и MedianHouseValue присутсвует достаточно выраженное отклонение от нормального распределения, что будет очень сильно влиять на линейные классификаторы.
2) Присутствует выраженный наклон распределения с правым хвостом у Population, TotalBedrooms, TotalRooms и Households.




Присутствует сильная корреляция по критерию Пирсена
Top-2

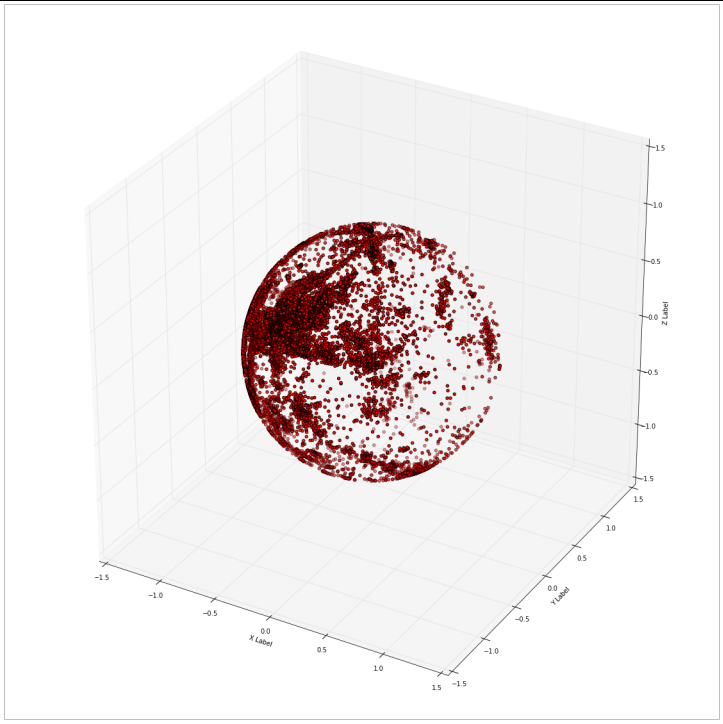
1) Households и TotalBedrooms = 0.979829
2) TotalRooms и TotalBedrooms = 0.929979

Стоит заметить что имеются NaN значения в следующих переменных

```
MedianIncome        0.039099 %
TotalRooms          0.040262 %
```

Таким образом, покрыв NaN значения другими значениями (далее) мы получим меньшую корреляцию.

# Предобработка



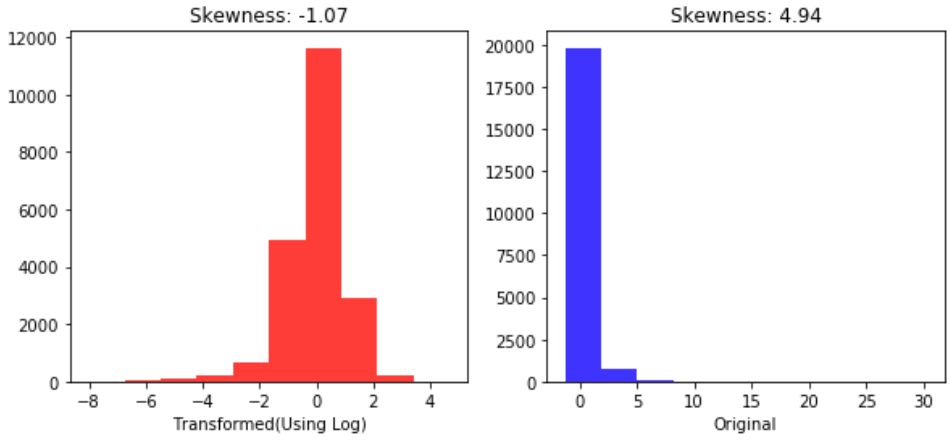
Переменные Longitude и Latitude достаточно высоко коррелируют, мы можем разложить их на 3 ортогональных вектора представленных как:
X = cos(Latitude) * cos(Longitute)
Y = cos(Latitude) * sin(Longitute)
Z = sin(Latitude)

На графике изображены X Y Z  в хорошем разбросе, с концентрацией на одном полушарии

Далее мы покрыли все NaN значения:

1) Для MedianIncome мы решили использовать медиану
2) Для TotalRooms = TotalBedrooms + mean(TotalRooms - TotalBedrooms)

Подобные замены почти не повлияли на само распределение и более того значения корреляции уменьшились лишь на 1 процент.




На рисунке приведен пример избавления от смещения за счет логарифмирования log-scale.
Таким образом на красном мы получили гораздо меньшим смещением, чем было изначально (на синем).

Так же , помимо Population, мы решили сгладить смещения для TotalRooms и TotalBedrooms при помощи инструменты boxcox что отлично повлияло на производительность моделей.

Значения Longitue и Latitude мы решили оставить как есть, без изменения, для шума (имеет своеобразное распределение), чтобы сделать модель более обобщённой, для исключения переобучения.

На последнем этапе предобработки, мы добавили Pipeline с первым шагом стандартизации данных по всем столбцам, для правильного вычисления дистанции для линейных моделей как SVR, rigid и др. Для DecisionTree и соответственно для AdaBoost мы не проводим стандартизацию данных.

## Построение модели

Изучение параметров функции прогнозирования и последующее тестирование модели на одних и тех же данных является методологической ошибкой. Чтобы избежать переобучения, при проведении (контролируемой) оценки модели мы будет удерживать 30% данных как тестовую выборку. Боле того, оценка алгоритма проводится 4 раза по разной обучающей и тестируемой выборке (равномерно распределение выборки).

Как показательную оценку регрессии мы выбрали Mean absolute percentage error (MAPE)
$M = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$ , где $A_t$ – актуальное значение, $F_t$ – предсказанное значение.
Таким образом чем меньше MAPE тел стабильнее и точнее регрессор.

## Результаты оценки её качества

Были построенны разные модели с перебором параметров sklearn.model_selection.GridSearch
Обобщённая оценка качества относительно всех регрессоров, проводилась на заранее определенной выборке тестирования по метрике MAPE.

| модель | Параметры | Оценка | MAPE |
|---|---|---|---|
| SGDRegressor | alpha = 0.0005; loss = squared_loss; penalty = elasticnet | 0.646054 | 30.4434 |
| Ridge | alpha = 0.5 | 0. 647674 | 30.4629 |
| DecisionTreeRegressor | min_samples_leaf = 10 | 0.750136 | 14.4293 |
| AdaBoost | loss = square ; n_estimators = 350 ; learning_rate = 1.00 ; DecisionTreeRegressor(criterion='mse', max_depth=12) | 0.830380 | 13.14 |

Делая вывод на обощенной оценке MAPE, AdaBoost имеет лучшую производительность. Стоит заметить, что она почти близка к оценке DecisionTree.

## Список литературы


1. Pace R. K., Barry R. Sparse spatial autoregressions //Statistics & Probability Letters. – 1997. – Т. 33. – №. 3. – С. 291-297.