Рубежный контроль №1

Выполнил: Олейников Илья, студент ИУ5-22М

Вариант 13, согласно ему номера задач: 13 и 33 для первой и второй соответственно.

13 - Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием функции "обратная зависимость - 1 / X".

33 - Для набора данных проведите процедуру отбора признаков (feature selection). Используйте метод обертывания (wrapper method), алгоритм полного перебора (exhaustive feature selection).

Подготовка данных

```
data = pd.read_csv('housing.csv', sep=',', encoding='windows-1251')
df = data.drop('Address', axis=1)
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 5000,\n  \"fields\":
[\n    {\n      \"column\": \"Avg. Area Income\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
10657.991213888688,\n        \"min\": 17796.63119,\n        \"max\":
107701.7484,\n        \"num_unique_values\": 5000,\n
\"samples\": [\n          61907.59335,\n          57160.20224,\n
70190.79644\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Avg. Area House Age\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.9914561798324226,\n        \"min\":
2.644304186,\n        \"max\": 9.519088066,\n
\"num_unique_values\": 5000,\n        \"samples\": [\n
7.017837825,\n          6.893260095,\n          6.745053762\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Avg. Area Number of Rooms\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1.0058332312754112,\n
\"min\": 3.236194023,\n        \"max\": 10.75958834,\n
\"num_unique_values\": 5000,\n        \"samples\": [\n
6.440255755,\n          6.921532165,\n          6.662566733\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Avg. Area Number of Bedrooms\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1.2341372654846832,\n
\"min\": 2.0,\n        \"max\": 6.5,\n        \"num_unique_values\":
255,\n        \"samples\": [\n          3.5,\n          3.41,\n
3.33\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

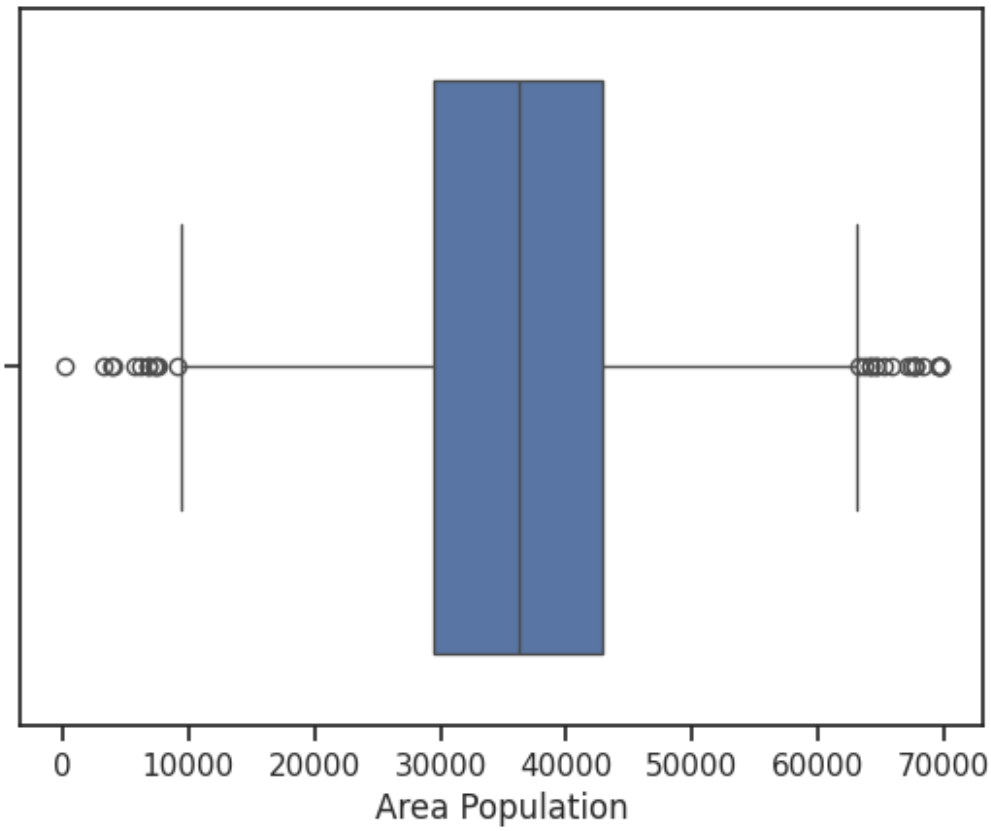
```
\"Area Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 9925.65011354602,\n        \"min\":
172.6106863,\n        \"max\": 69621.71338,\n
\"num_unique_values\": 5000,\n        \"samples\": [\n
43828.94721,\n          43467.14704,\n          29215.13611\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Price\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 353117.6265836956,\n        \"min\": 15938.65792,\n
\"max\": 2469065.594,\n        \"num_unique_values\": 5000,\n
\"samples\": [\n          1339096.077,\n          1251794.179,\n
1340094.966\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

```
print(df.shape)
```

```
(5000, 6)
```

```
import seaborn as sns
import pandas as pd

sns.set(style="ticks")
sns.boxplot(x=df['Area Population'])
```

```
<Axes: xlabel='Area Population'>
```



Задача №1 (13)

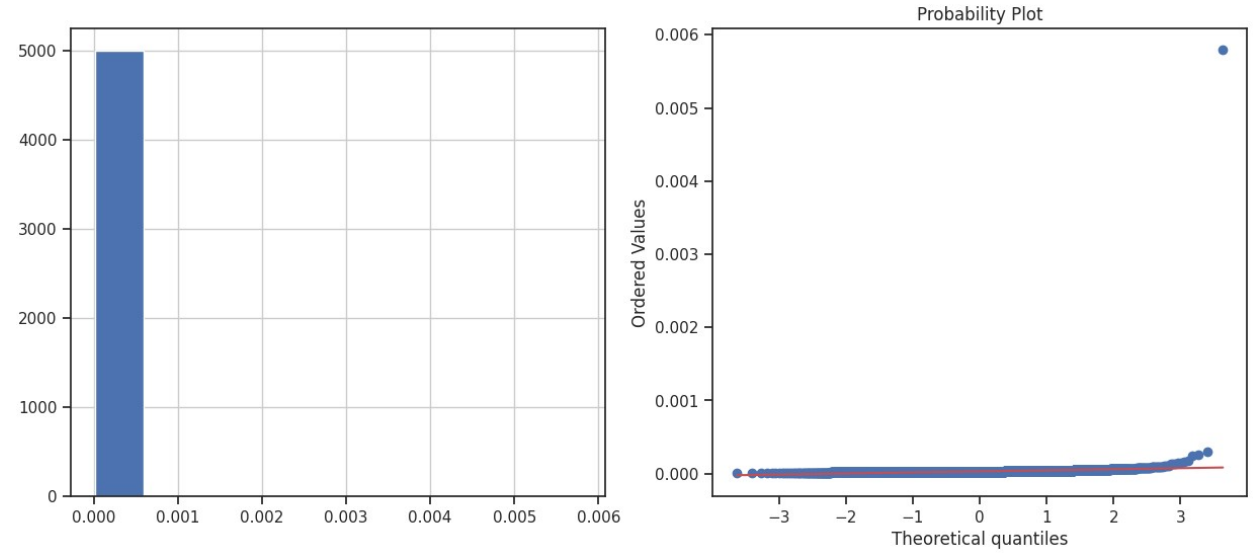
Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием функции "обратная зависимость - 1 / X".

Проведём нормализацию для колонки "mag"

```
import matplotlib.pyplot as plt  # noqa
import numpy as np  # noqa
import pandas as pd  # noqa
import scipy.stats as stats  # noqa

# Выбор произвольного числового признака для нормализации
selected_feature = 'Area Population'  # Замените на имя вашего
выбранного признака

# Применение нормализации с использованием обратной зависимости
normalized_feature = 1 / data[selected_feature]

# Добавление нормализованного признака к DataFrame
data['normalized_' + selected_feature] = normalized_feature

# Вывод первых нескольких строк для проверки результата
print(data.head())
```

```
   Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
0       79545.45857             5.682861                   7.009188
1       79248.64245             6.002900                   6.730821
2       61287.06718             5.865890                   8.512727
3       63345.24005             7.188236                   5.586729
4       59982.19723             5.040555                   7.839388

   Avg. Area Number of Bedrooms  Area Population         Price  \
0                          4.09      23086.80050  1.059034e+06
1                          3.09      40173.07217  1.505891e+06
2                          5.13      36882.15940  1.058988e+06
3                          3.26      34310.24283  1.260617e+06
4                          4.23      26354.10947  6.309435e+05

                                             Address  \
0  208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1  188 Johnson Views Suite 079\nLake Kathleen, CA...
2  9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3                          USS Barnett\nFPO AP 44820
4                         USNS Raymond\nFPO AE 09386

   normalized_Area Population
0                    0.000043
1                    0.000025
2                    0.000027
3                    0.000029
4                    0.000038
```

```
df["reciprocal"] = 1 / (df["Area Population"])

diagnostic_plots(df, "reciprocal")
```



Вывод

Как видно, нормализация такой функцией неудачна.

Задача №2 (33)

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте метод обертывания (wrapper method), алгоритм полного перебора (exhaustive feature selection).

```
from mlxtend.feature_selection import ExhaustiveFeatureSelector
from sklearn.model_selection import train_test_split  # Importing
train_test_split
from sklearn.linear_model import LinearRegression

df = df.dropna()

import pandas as pd
from mlxtend.feature_selection import ExhaustiveFeatureSelector
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split


# Разделение на признаки и целевую переменную
X = df.drop(columns=['Price'])  # Укажите имя целевой переменной
y = df['Price']

# Разделение на обучающий и тестовый наборы
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Создание модели для отбора признаков
lr = LinearRegression()

# Создание объекта для отбора признаков с использованием алгоритма
полного перебора
efs = ExhaustiveFeatureSelector(estimator=lr, min_features=1,
max_features=len(X.columns), scoring='r2', cv=5)

# Запуск процесса отбора признаков
efs = efs.fit(X_train, y_train)

# Вывод результатов
selected_features = X_train.columns[list(efs.best_idx_)]
print("Отобранные признаки:", selected_features)
```

```
Features: 31/31

Отобранные признаки: Index(['Avg. Area Income', 'Avg. Area House Age',
'Avg. Area Number of Rooms',
```

```
       'Avg. Area Number of Bedrooms', 'Area Population'],
      dtype='object')
```