Описание и загрузка датасета: California Housing

Цель: задача регрессии — предсказать среднюю стоимость дома в районе на основе социально-экономических и географических признаков.

Признаки:

- `MedInc` — средний доход в районе
- `HouseAge` — средний возраст домов
- `AveRooms` — среднее число комнат
- `AveBedrms` — среднее число спален
- `Population` — численность населения
- `AveOccup` — среднее количество жильцов
- `Latitude` и `Longitude` — координаты

Целевая переменная:

- `MedHouseVal` — медианная стоимость дома (в сотнях тысяч долларов)

```
from sklearn.datasets import fetch_california_housing
import pandas as pd

# Загрузка датасета
data = fetch_california_housing()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

# Первичный анализ
print("Размер датасета:", df.shape)
display(df.head())

# Проверка пропущенных значений
print("\nПропущенные значения:")
print(df.isnull().sum())

# Распределение целевой переменной
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.histplot(df['target'], bins=40, kde=True)
plt.title("Распределение стоимости домов (целевая переменная)")
plt.xlabel("Стоимость (в сотнях тысяч $)")
plt.show()
```

```
Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: scikit-learn in c:\users\aslan\appdata\
local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\
localcache\local-packages\python312\site-packages (1.6.1)
```

```
Requirement already satisfied: numpy>=1.19.5 in c:\users\aslan\
appdata\local\packages\
pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-
packages\python312\site-packages (from scikit-learn) (2.1.3)
Requirement already satisfied: scipy>=1.6.0 in c:\users\aslan\appdata\
local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\
localcache\local-packages\python312\site-packages (from scikit-learn)
(1.15.2)
Requirement already satisfied: joblib>=1.2.0 in c:\users\aslan\
appdata\local\packages\
pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-
packages\python312\site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\aslan\
appdata\local\packages\
pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-
packages\python312\site-packages (from scikit-learn) (3.5.0)


[notice] A new release of pip is available: 25.0.1 -> 25.1.1
[notice] To update, run: C:\Users\aslan\AppData\Local\Microsoft\
WindowsApps\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\
python.exe -m pip install --upgrade pip
```

```
---------------------------------------------------------------------------
-----
ModuleNotFoundError                       Traceback (most recent call
last)
Cell In[5], line 2
      1 get_ipython().system('pip install scikit-learn')
----> 2 from sklearn.datasets import fetch_california_housing
      3 import pandas as pd
      5 # Загрузка датасета

ModuleNotFoundError: No module named 'sklearn'
```

Предобработка данных: масштабирование и разделение выборки

```python
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Делим X и y
X = df.drop('target', axis=1)
y = df['target']

# Масштабирование признаков
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# Разделение на train и test
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42
)

print("Размер обучающей выборки:", X_train.shape)
print("Размер тестовой выборки:", X_test.shape)
```

Модель стекинга (Stacking Regressor)

```
from sklearn.ensemble import StackingRegressor
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error, r2_score

base_models = [
    ('lr', LinearRegression()),
    ('rf', RandomForestRegressor(n_estimators=50, random_state=42)),
    ('knn', KNeighborsRegressor(n_neighbors=5))
]
meta_model = GradientBoostingRegressor(n_estimators=100,
random_state=42)

stacking_model = StackingRegressor(
    estimators=base_models,
    final_estimator=meta_model,
    passthrough=True
)

stacking_model.fit(X_train, y_train)
y_pred_stack = stacking_model.predict(X_test)

mse_stack = mean_squared_error(y_test, y_pred_stack)
r2_stack = r2_score(y_test, y_pred_stack)

print(f"Stacking Regressor:\nMSE: {mse_stack:.4f}, R²:
{r2_stack:.4f}")
```

Модель многослойного персептрона (MLP Regressor)

```
from sklearn.neural_network import MLPRegressor

mlp = MLPRegressor(hidden_layer_sizes=(50, 10), max_iter=1000,
random_state=42)
mlp.fit(X_train, y_train)

y_pred_mlp = mlp.predict(X_test)
```

```
mse_mlp = mean_squared_error(y_test, y_pred_mlp)
r2_mlp = r2_score(y_test, y_pred_mlp)

print(f"MLP Regressor:\nMSE: {mse_mlp:.4f}, R²: {r2_mlp:.4f}")
```

Модели семейства МГУА (GMDH)