**Министерство науки и высшего образования Российской Федерации** **Федеральное государственное бюджетное образовательное учреждение высшего образования** **«Московский государственный технический университет** **имени Н.Э. Баумана** 

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Технологии машинного обучения»

Лабораторная работа №3

Выполнил:

студент группы ИУ5-63Б

Комаров Д. С.

Проверил:

Гапанюк Ю. Е.

2024 г.

**Ход работы:**

# **Подключение библиотек, загрузка и очистка датасета, кодирование категориальных признаков**

**import** pandas **as** pd  
**import** numpy **as** np  
**import** time  
**from** sklearn.model\_selection **import** train\_test\_split, GridSearchCV, RandomizedSearchCV, KFold, cross\_val\_score  
**from** sklearn.neighbors **import** KNeighborsRegressor  
**from** sklearn.metrics **import** accuracy\_score  
**from** sklearn.neighbors **import** KNeighborsRegressor, KNeighborsClassifier  
  
**from** sklearn.preprocessing **import** MinMaxScaler, StandardScaler  
**from** matplotlib **import** pyplot **as** plt  
**import** seaborn **as** sns  
**from** sklearn.metrics **import** mean\_absolute\_error, mean\_squared\_error, r2\_score  
**from** warnings **import** simplefilter  
  
simplefilter('ignore')

data = pd.read\_csv("sample\_data/HousingData.csv")

data.head()

{"summary":"{\n \"name\": \"data\",\n \"rows\": 506,\n \"fields\": [\n {\n \"column\": \"CRIM\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 8.720191850151599,\n \"min\": 0.00632,\n \"max\": 88.9762,\n \"num\_unique\_values\": 484,\n \"samples\": [\n 15.1772,\n 0.2896,\n 0.08308\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"ZN\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 23.388876146265478,\n \"min\": 0.0,\n \"max\": 100.0,\n \"num\_unique\_values\": 26,\n \"samples\": [\n 25.0,\n 30.0,\n 18.0\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"INDUS\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 6.83589649864144,\n \"min\": 0.46,\n \"max\": 27.74,\n \"num\_unique\_values\": 76,\n \"samples\": [\n 8.14,\n 1.47,\n 1.22\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"CHAS\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 0.2553404809065679,\n \"min\": 0.0,\n \"max\": 1.0,\n \"num\_unique\_values\": 2,\n \"samples\": [\n 1.0,\n 0.0\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"NOX\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 0.11587767566755595,\n \"min\": 0.385,\n \"max\": 0.871,\n \"num\_unique\_values\": 81,\n \"samples\": [\n 0.401,\n 0.538\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"RM\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 0.7026171434153233,\n \"min\": 3.561,\n \"max\": 8.78,\n \"num\_unique\_values\": 446,\n \"samples\": [\n 6.849,\n 4.88\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"AGE\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 27.99951300509237,\n \"min\": 2.9,\n \"max\": 100.0,\n \"num\_unique\_values\": 348,\n \"samples\": [\n 82.8,\n 88.4\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"DIS\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 2.105710126627611,\n \"min\": 1.1296,\n \"max\": 12.1265,\n \"num\_unique\_values\": 412,\n \"samples\": [\n 2.2955,\n 4.2515\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"RAD\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 8,\n \"min\": 1,\n \"max\": 24,\n \"num\_unique\_values\": 9,\n \"samples\": [\n 7,\n 2\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"TAX\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 168,\n \"min\": 187,\n \"max\": 711,\n \"num\_unique\_values\": 66,\n \"samples\": [\n 370,\n 666\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"PTRATIO\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 2.1649455237144406,\n \"min\": 12.6,\n \"max\": 22.0,\n \"num\_unique\_values\": 46,\n \"samples\": [\n 19.6,\n 15.6\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"B\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 91.29486438415783,\n \"min\": 0.32,\n \"max\": 396.9,\n \"num\_unique\_values\": 357,\n \"samples\": [\n 396.24,\n 395.11\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"LSTAT\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 7.155870815805251,\n \"min\": 1.73,\n \"max\": 37.97,\n \"num\_unique\_values\": 438,\n \"samples\": [\n 26.64,\n 7.51\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"MEDV\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 9.197104087379818,\n \"min\": 5.0,\n \"max\": 50.0,\n \"num\_unique\_values\": 229,\n \"samples\": [\n 14.1,\n 22.5\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n }\n ]\n}","type":"dataframe","variable\_name":"data"}

## **CRIM: средняя преступность на душу населения в городе.**

## **ZN: доля земли, зарезервированной под участки более 25 000 кв.футов.**

## **INDUS: доля неторговых бизнес-акров на город.**

## **CHAS: переменная Charles River (равна 1, если участок граничит с рекой; 0 в противном случае).**

## **NOX: концентрация нитритных оксидов (части на 10 миллионов).**

## **RM: среднее количество комнат в жилье.**

## **AGE: доля домов, построенных до 1940 года.**

## **DIS: взвешенные расстояния до пяти центров занятости в Бостоне.**

## **RAD: индекс доступности к радиальным автомагистралям.**

## **TAX: средний налоговый ставка на недвижимость на $10 000.**

## **PTRATIO: соотношение учеников к учителям в городе.**

## **B: 1000(Bk - 0.63)^2, где Bk - доля чернокожих в городе.**

## **LSTAT: процент населения с низким социальным статусом.**

## **MEDV: медианная стоимость дома с собственником в тысячах долларов**

data.isna().sum()

CRIM 20  
ZN 20  
INDUS 20  
CHAS 20  
NOX 0  
RM 0  
AGE 20  
DIS 0  
RAD 0  
TAX 0  
PTRATIO 0  
B 0  
LSTAT 20  
MEDV 0  
dtype: int64

data.fillna(data.mean(), inplace=True)

data.isna().sum()

CRIM 0  
ZN 0  
INDUS 0  
CHAS 0  
NOX 0  
RM 0  
AGE 0  
DIS 0  
RAD 0  
TAX 0  
PTRATIO 0  
B 0  
LSTAT 0  
MEDV 0  
dtype: int64

## **Разделение выборки на обучающую и на тестовую**

y = data['MEDV']  
X = data.drop('MEDV', axis=1)  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=3)  
x\_train

{"summary":"{\n \"name\": \"x\_train\",\n \"rows\": 354,\n \"fields\": [\n {\n \"column\": \"CRIM\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 9.14837363066627,\n \"min\": 0.00632,\n \"max\": 88.9762,\n \"num\_unique\_values\": 339,\n \"samples\": [\n 0.77299,\n 5.70818,\n 0.05561\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"ZN\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 21.547884632260175,\n \"min\": 0.0,\n \"max\": 95.0,\n \"num\_unique\_values\": 25,\n \"samples\": [\n 25.0,\n 17.5,\n 0.0\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"INDUS\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 6.699616840339832,\n \"min\": 0.46,\n \"max\": 27.74,\n \"num\_unique\_values\": 70,\n \"samples\": [\n 12.83,\n 11.083991769547325,\n 1.47\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"CHAS\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 0.25627098791956593,\n \"min\": 0.0,\n \"max\": 1.0,\n \"num\_unique\_values\": 3,\n \"samples\": [\n 0.0,\n 1.0,\n 0.06995884773662552\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"NOX\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 0.1161843695919214,\n \"min\": 0.385,\n \"max\": 0.871,\n \"num\_unique\_values\": 78,\n \"samples\": [\n 0.538,\n 0.77,\n 0.507\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"RM\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 0.6852887295815093,\n \"min\": 3.561,\n \"max\": 8.704,\n \"num\_unique\_values\": 331,\n \"samples\": [\n 6.389,\n 7.333,\n 6.083\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"AGE\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 27.0296133386822,\n \"min\": 2.9,\n \"max\": 100.0,\n \"num\_unique\_values\": 260,\n \"samples\": [\n 54.3,\n 47.4,\n 89.8\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"DIS\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 2.108527072163094,\n \"min\": 1.1296,\n \"max\": 12.1265,\n \"num\_unique\_values\": 304,\n \"samples\": [\n 1.5184,\n 3.0923,\n 7.6534\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"RAD\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 8,\n \"min\": 1,\n \"max\": 24,\n \"num\_unique\_values\": 9,\n \"samples\": [\n 1,\n 2,\n 8\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"TAX\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 169,\n \"min\": 187,\n \"max\": 711,\n \"num\_unique\_values\": 59,\n \"samples\": [\n 666,\n 224,\n 255\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"PTRATIO\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 2.1730352765252476,\n \"min\": 12.6,\n \"max\": 22.0,\n \"num\_unique\_values\": 44,\n \"samples\": [\n 18.3,\n 15.6,\n 20.1\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"B\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 90.77246808376455,\n \"min\": 0.32,\n \"max\": 396.9,\n \"num\_unique\_values\": 250,\n \"samples\": [\n 331.29,\n 392.78,\n 379.41\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"LSTAT\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 7.039317796526101,\n \"min\": 1.92,\n \"max\": 37.97,\n \"num\_unique\_values\": 319,\n \"samples\": [\n 13.59,\n 9.55,\n 9.62\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n }\n ]\n}","type":"dataframe","variable\_name":"x\_train"}

*# Размер обучающей выборки*  
x\_train.shape, y\_train.shape

((354, 13), (354,))

*# Размер тестовой выборки*  
x\_test.shape, y\_test.shape

((152, 13), (152,))

*# Создание модели с k=5 соседями*  
knn\_model = KNeighborsRegressor(n\_neighbors=5)  
  
*# Обучение модели на обучающих данных*  
knn\_model.fit(x\_train, y\_train)  
  
*# Прогнозирование на тестовых данных*  
y\_pred = knn\_model.predict(x\_test)

*# коэффициент детерминации*  
print(f"R^2: {r2\_score(y\_test, y\_pred)}")  
*# среднеквадратичная ошибка*  
print(f"MSE: {mean\_squared\_error(y\_test, y\_pred)}")  
*# средняя абсолютная ошибка*  
print(f"MAE: {mean\_absolute\_error(y\_test, y\_pred)}")

R^2: 0.3944921792280742  
MSE: 47.623471052631565  
MAE: 5.030131578947368

knn\_model = KNeighborsRegressor(n\_neighbors=4)  
*# Обучение модели на обучающих данных*  
knn\_model.fit(x\_train, y\_train)  
  
*# Прогнозирование на тестовых данных*  
y\_pred = knn\_model.predict(x\_test)  
  
*# коэффициент детерминации*  
print(f"R^2: {r2\_score(y\_test, y\_pred)}")  
*# среднеквадратичная ошибка*  
print(f"MSE: {mean\_squared\_error(y\_test, y\_pred)}")  
*# средняя абсолютная ошибка*  
print(f"MAE: {mean\_absolute\_error(y\_test, y\_pred)}")

R^2: 0.38937612423899104  
MSE: 48.025851151315784  
MAE: 4.977796052631579

knn\_model = KNeighborsRegressor(n\_neighbors=11)  
*# Обучение модели на обучающих данных*  
knn\_model.fit(x\_train, y\_train)  
  
*# Прогнозирование на тестовых данных*  
y\_pred = knn\_model.predict(x\_test)  
  
*# коэффициент детерминации*  
print(f"R^2: {r2\_score(y\_test, y\_pred)}")  
*# среднеквадратичная ошибка*  
print(f"MSE: {mean\_squared\_error(y\_test, y\_pred)}")  
*# средняя абсолютная ошибка*  
print(f"MAE: {mean\_absolute\_error(y\_test, y\_pred)}")

R^2: 0.38314945247469157  
MSE: 48.51558177468465  
MAE: 5.013277511961722

## **Произведите подбор гиперпараметра K с использованием GridSearchCV и RandomizedSearchCV**

*# Определение модели KNeighborsRegressor*  
knn = KNeighborsRegressor()  
  
*# Определение гиперпараметров, которые будут тестироваться*  
param\_grid = {'n\_neighbors': np.arange(1, 100)}  
  
*# Инициализация GridSearchCV*  
grid\_search = GridSearchCV(knn, param\_grid, cv=5, scoring='neg\_mean\_squared\_error')  
  
*# Обучение GridSearchCV*  
grid\_search.fit(x\_train, y\_train)  
  
*# Вывод лучших гиперпараметров и оценки качества*  
print("Лучшие гиперпараметры для GridSearchCV:", grid\_search.best\_params\_)  
print("Лучшая оценка для GridSearchCV:", -grid\_search.best\_score\_)

Лучшие гиперпараметры для GridSearchCV: {'n\_neighbors': 4}  
Лучшая оценка для GridSearchCV: 46.96485802313883

*# Инициализация RandomizedSearchCV*  
random\_search = RandomizedSearchCV(knn, param\_distributions=param\_grid, n\_iter=10, cv=5, scoring='neg\_mean\_squared\_error', random\_state=42)  
  
*# Обучение RandomizedSearchCV*  
random\_search.fit(x\_train, y\_train)  
  
*# Вывод лучших гиперпараметров и оценки качества*  
print("Лучшие гиперпараметры для RandomizedSearchCV:", random\_search.best\_params\_)  
print("Лучшая оценка для RandomizedSearchCV:", -random\_search.best\_score\_)

Лучшие гиперпараметры для RandomizedSearchCV: {'n\_neighbors': 11}  
Лучшая оценка для RandomizedSearchCV: 53.48128490280527

*# Оценка качества оптимальной модели с использованием кросс-валидации*  
kf = KFold(n\_splits=5, shuffle=True, random\_state=42)  
  
grid\_scores = cross\_val\_score(grid\_search.best\_estimator\_, x\_train, y\_train, cv=kf, scoring='neg\_mean\_squared\_error')  
print("Оценка качества оптимальной модели с использованием кросс-валидации:", np.mean(-grid\_scores))  
  
random\_scores = cross\_val\_score(random\_search.best\_estimator\_, x\_train, y\_train, cv=kf, scoring='neg\_mean\_squared\_error')  
print("Оценка качества оптимальной модели с использованием кросс-валидации:", np.mean(-random\_scores))

Оценка качества оптимальной модели с использованием кросс-валидации: 49.121765392354135  
Оценка качества оптимальной модели с использованием кросс-валидации: 49.33459348820194