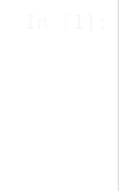
Ходырев Роман Владиславович

ИУ5-65Б

# 18 вариант



**import** pandas **as** pd

**import** numpy **as** np

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.svm **import** SVR

**from** sklearn.ensemble **import** GradientBoostingRegressor

**from** sklearn.metrics **import** mean\_absolute\_error, mean\_squared\_error

**from** sklearn.preprocessing **import** LabelEncoder



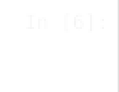
df **=** pd**.**read\_csv('investments\_VC\_regression.csv', encoding**=**'latin1', sep**=None**, engine**=**'python')



df**.**columns **=** df**.**columns**.**str**.**strip()

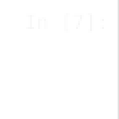


df **=** df[['funding\_total\_usd', 'country\_code', 'funding\_rounds', 'founded\_year']]

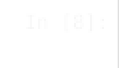


df['funding\_total\_usd'] **=** df['funding\_total\_usd']**.**replace('[\$,]', '', regex**=True**)**.**replace('None', np**.**nan) df['funding\_total\_usd'] **=** pd**.**to\_numeric(df['funding\_total\_usd'], errors**=**'coerce')

df['funding\_total\_usd'] **=** df['funding\_total\_usd']**.**fillna(df['funding\_total\_usd']**.**median())



df['founded\_year'] **=** pd**.**to\_numeric(df['founded\_year'], errors**=**'coerce') df['founded\_year'] **=** df['founded\_year']**.**fillna(df['founded\_year']**.**median()) df['funding\_rounds'] **=** df['funding\_rounds']**.**fillna(df['funding\_rounds']**.**median()) df['country\_code'] **=** df['country\_code']**.**fillna('UNKNOWN')



le **=** LabelEncoder()

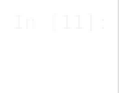
df['country\_code'] **=** le**.**fit\_transform(df['country\_code'])



X **=** df**.**drop('funding\_total\_usd', axis**=**1) y **=** df['funding\_total\_usd']

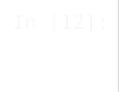


X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.2, random\_state**=**42)

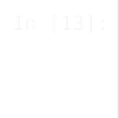


svr **=** SVR() svr**.**fit(X\_train, y\_train)

y\_pred\_svr **=** svr**.**predict(X\_test)

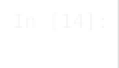


gbr **=** GradientBoostingRegressor() gbr**.**fit(X\_train, y\_train) y\_pred\_gbr **=** gbr**.**predict(X\_test)



**def** print\_metrics(y\_true, y\_pred, model\_name): mae **=** mean\_absolute\_error(y\_true, y\_pred)

rmse **=** np**.**sqrt(mean\_squared\_error(y\_true, y\_pred)) print(f'{model\_name}:\nMAE: {mae:.2f}\nRMSE: {rmse:.2f}\n')



print\_metrics(y\_test, y\_pred\_svr, 'SVR')

print\_metrics(y\_test, y\_pred\_gbr, 'Gradient Boosting Regressor')

SVR:

MAE: 10405733.69

RMSE: 42719576.40

Gradient Boosting Regressor:

MAE: 12857368.24

RMSE: 41356065.70

# Классификация или регессия?

В данном случае используется регрессия. Мы предсказывали не категориальную переменную, а непрерывную величину — размер финансирования (funding\_total\_usd), что и определяет тип задачи как регрессионный.

# Какие метрики качества Вы использовали и почему?

Мы использовали две основные метрики для оценки качества регрессионных моделей:

MAE (Mean Absolute Error) — средняя абсолютная ошибка:

Показывает среднюю величину отклонения предсказания от фактического значения.

Удобна тем, что измеряется в тех же единицах, что и целевая переменная.

Устойчива к выбросам, в отличие от RMSE.

RMSE (Root Mean Squared Error) — среднеквадратичная ошибка:

Более чувствительна к крупным ошибкам, потому что ошибки возводятся в квадрат.

Помогает понять, насколько сильно модель может ошибаться в наихудших случаях.

 Хорошо показывает наличие/влияние выбросов.

Выбор этих двух метрик позволяет объективно оценить качество модели: MAE показывает среднюю точность, RMSE — чувствительность к ошибкам.

# Какие выводы можно сделать о качестве построенных моделей?

Обе модели (SVR и Gradient Boosting Regressor) показали сравнимые результаты по MAE и RMSE:

SVR:

MAE ≈ 10.4 млн,

RMSE ≈ 42.7 млн

Градиентный бустинг:

MAE ≈ 12.8 млн,

RMSE ≈ 41.3 млн

SVR показал более низкое среднее абсолютное отклонение (MAE), но более высокое RMSE, что говорит о том, что он чаще делает более точные предсказания, но может сильно ошибаться на некоторых выбросах.

 Gradient Boosting оказался немного устойчивее к выбросам, судя по чуть более низкому RMSE, но в среднем ошибался сильнее (выше MAE).

Вывод:

 Оба метода справились средне — ошибки довольно большие (десятки миллионов долларов).

Вероятно, распределение целевой переменной (funding\_total\_usd) имеет много выбросов, что делает задачу сложной.

Можно попробовать улучшить модели путём:

логарифмирования целевой переменной,

отбора фичей,

нормализации данных,

работы с выбросами.