Лекция 6. Отбор моделей ML и целевых метрик, обучение моделей и их интерпретация

Практикум по программированию, 5 семестр

Иван Евгеньевич Бугаенко,
ассистент каф. ПМиФИ

Что мы уже умеем делать с данными?

1. Анализ семантики столбцов
2. Разделение выборки на обучение и тест
3. Заполнение пропусков
4. Кодирование категориальных данных
5. Устранение выбросов
6. Устранение аномалий

Раздел 1. EDA

7. Генерация новых признаков
8. Устранение мультиколлинеарности
9. Скалирование признаков

Раздел 2. Генерация признаков

1. Анализ семантики столбцов
2. Разделение выборки на обучение и тест
3. Заполнение пропусков
4. Кодирование категориальных данных
5. Устранение выбросов
6. Устранение аномалий

Раздел 1. EDA

7. Генерация новых признаков
8. Устранение мультиколлинеарности
9. Скалирование признаков

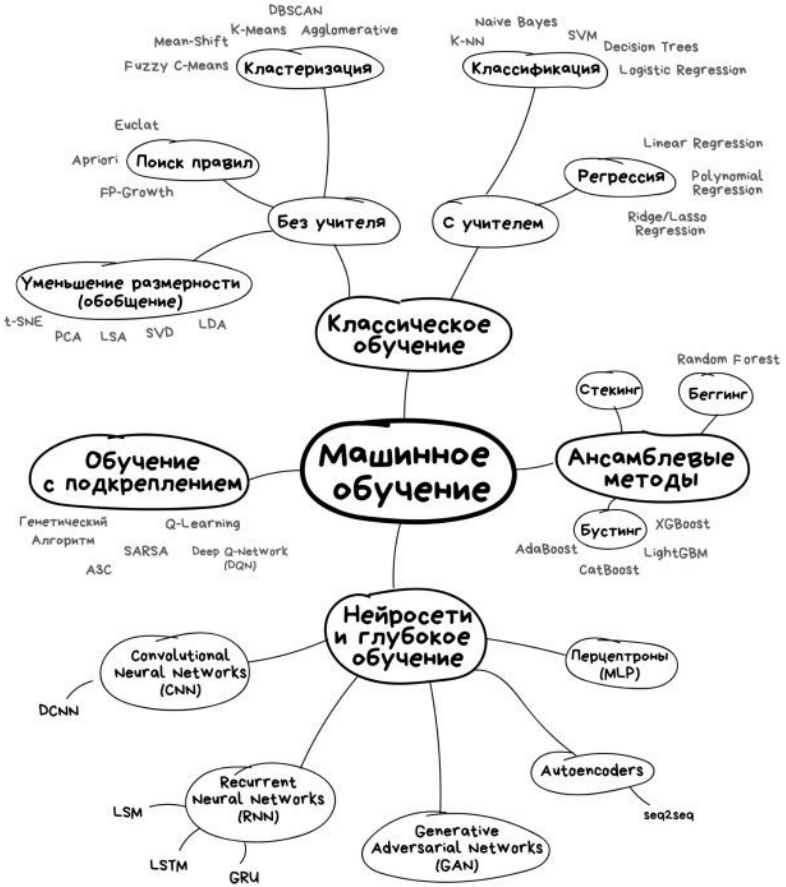
Раздел 2. Генерация признаков

Обработка / анализ данных

Что же дальше?

Важнейшим этапом работы над ML-проектом является 3 раздел: **отбор моделей и метрик**

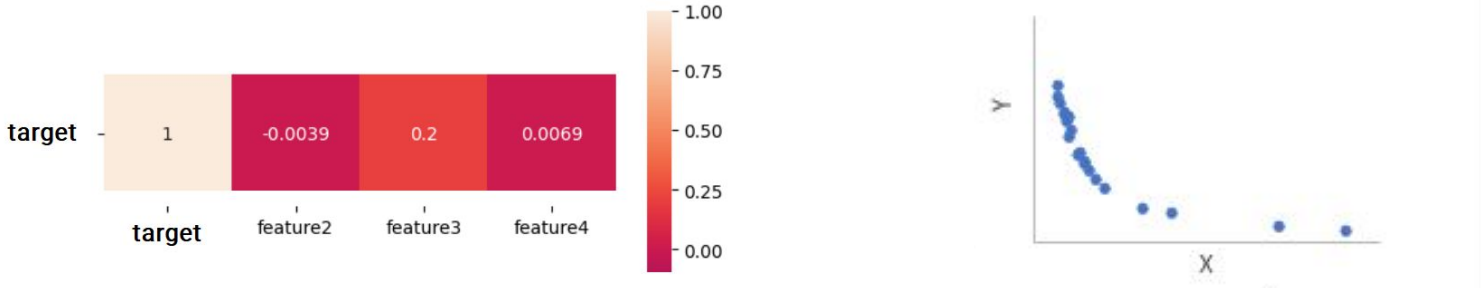
На данном этапе происходит анализ зависимостей входных признаков и целевого, а также определяется пул метрик и моделей


Машинное обучение
Классическое обучение
Без учителя
С учителем
Кластеризация
DBSCAN
K-Means
Agglomerative
Mean-Shift
Fuzzy C-Means
Классификация
Naive Bayes
K-NN
SVM
Decision Trees
Logistic Regression
Поиск правил
Euclat
Apriori
FP-Growth
Регрессия
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Уменьшение размерности (обобщение)
t-SNE
PCA
LSA
SVD
LDA
Обучение с подкреплением
Генетический Алгоритм
A3C
SARSA
Q-Learning
Deep Q-Network (DQN)
Ансамблевые методы
Стекинг
Беггинг
Random Forest
Бустинг
AdaBoost
XGBoost
LightGBM
CatBoost
Нейросети и глубокое обучение
Convolutional Neural Networks (CNN)
DCNN
Перцептроны (MLP)
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq

Чек-лист отбора моделей для прогнозирования

1. Определение вида зависимостей между входными признаками и целевым

Инструментарий: тепловая карта корреляции Пирсона, диаграммы рассеяния



Если есть основания, что существуют **линейные** зависимости, то хороши следующие модели: линейная регрессия / Ridge / Lasso / ElasticNet, SVM с линейным ядром.

\+ высокая интерпретируемость
\- риск недообучения / обучаются слишком долго

Чек-лист отбора моделей для прогнозирования

2. Анализ размера выборки и сложности алгоритмов

Инструментарий: программные средства Python / Pandas / NumPy

Рекомендаци:
1) малый датасет (< 5k): избегать нейронные сети, можно использовать линейные модели, деревья, ансамбли, SVM | не так страшны модели с высокой алгоритмической сложностью*
2) средний датасет (5-200k): ансамбли, KNN
3) большой датасет (> 200k): можно подключать глубокие нейросети, плохо показывают себя KNN, SVM и градиентный бустинг

* Тяжелыми моделями считаются: нейронные сети, KNN, бустинги, гауссовые процессы, SVM с нелинейными ядрами

Чек-лист отбора моделей для прогнозирования

3. Анализ размерности признакового пространства

Инструментарий: программные средства Python / Pandas / NumPy

Рекомендаци:
1) признаковое пространство очень большое (> 200 признаков, очень разреженные данные): хорошо себя показывают линейные модели, SVM, логистическая регрессия, Байесовские классификаторы, **лучше до такого не доводить и понизить пространство**
2) признаковое пространство обычное (до 200 признаков): все алгоритмы хороши

Помним про сложность алгоритмов!

Чек-лист отбора моделей для прогнозирования

4. Анализ требований к интерпретируемости

Инструментарий: наши познания в работе алгоритмов :(

Рекомендаци:
1) если бизнес требует интерпретируемость моделей, то в ОБЩЕМ СЛУЧАЕ нам не подходят нейронные сети, ансамбли и SVM с нелинейными ядрами
2) линейная регрессия – самый интерпретируемый алгоритм, потому что виден вклад признаков в общее предсказание
3) существуют методы, как можно сделать **ВСЕ** модели интерпретируемыми

Чек-лист отбора моделей для прогнозирования

5. Анализ сложности подбора гиперпараметров

Инструментарий: наши познания в параметрах алгоритмов :(

Рекомендация: всегда оценивайте, насколько модели сложные и вариативные. Часто бывает такое, что подбор гиперпараметров занимает огромное количество времени (как у нейронных сетей, к примеру)

* Часто сетки для гиперпараметров формируют в 3 разделе

Советы по выбору метрик

1. Метрики всегда надо выбирать по типу задачи
2. Если в задаче классификации очень сильный дисбаланс классов, то accuracy и ROC-AUC – плохой выбор
3. Для классификации нужно определиться, что страшнее – ошибка 1 рода (модель сказала да, хотя на самом деле – нет, максимизируем precision) или 2 рода (модель сказала нет, хотя на самом деле – да, максимизируем recall)
4. Метрика должна быть интерпретируемой (часто MSE понять тяжело, а вот MAE – другое дело)
5. Не нужно зацикливаться на одной метрике, нужно смотреть на группу метрик
6. Метрика выбирается ДО НАЧАЛА ОБУЧЕНИЯ

Основные метрики

1. Основные метрики регрессии: MSE, RMSE, MAE, MAPE, SMAPE, R2
2. Основные метрики классификации: accuracy, precision, recall, f1-score, ROC-AUC

Выбор моделей и метрик → обучение

Процесс обучения включает в себя подбор гиперпараметров моделей, обученных на всех вариантах датасета, а также кросс-валидацию

Кросс-валидация – это специальная техника оценки качества моделей, при которой проверяется ее обобщающая способность на различных вариантах разбиения датасета

Виды кросс-валидации

1. **k-Fold** (выборка делится на k равных частей, модель обучается на k-1 фолдах, на оставшемся фолде вычисляется метрика)
2. Stratified k-Fold (то же, что и k-Fold, только производится стратификация классов)
3. Leave-one-out (каждый объект по-очереди становится тестовым)
4. Leave-P-out (P объектов по-очереди становятся обучающими)
5. ShuffleSplit (данные N раз случайно делятся в некоторой пропорции на обучение и валидацию)
6. Time Series Split (данные делятся по времени так, чтобы будущее не использовалось для предсказания прошлого)

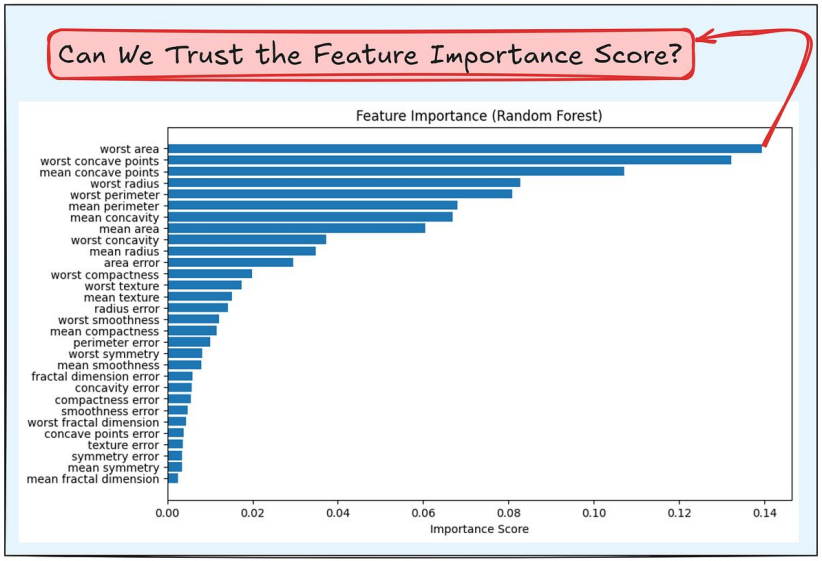
Стратегия по обучению моделей ML

1. Фиксируется вариант обработки обучающего датасета,
2. Фиксируется набор гиперпараметров и модель
3. Производится кросс-валидация: для всех обучающих и валидационных фрагментов датасета, полученных в результате кросс-валидации, обучается данная модель с данными параметрами
4. Для всех полученных моделей на валидационном фрагменте вычисляется метрика, она усредняется. Эта метрика – оценка данной модели с данными гиперпараметрами
5. Выбирается следующий набор гиперпараметров, повторяется пункт 3.
6. В результате отбираем тот набор гиперпараметров, на котором достигается лучшая усредненная метрика

Почему некоторые модели нельзя интерпретировать?

Некоторые модели ML тяжело интерпретировать, поскольку непонятно, почему было совершено то или иное предсказание и какой вклад имеют входные параметры

Частично, деревянные модели решают эту проблему. У них есть такой инструмент, как Feature Importance

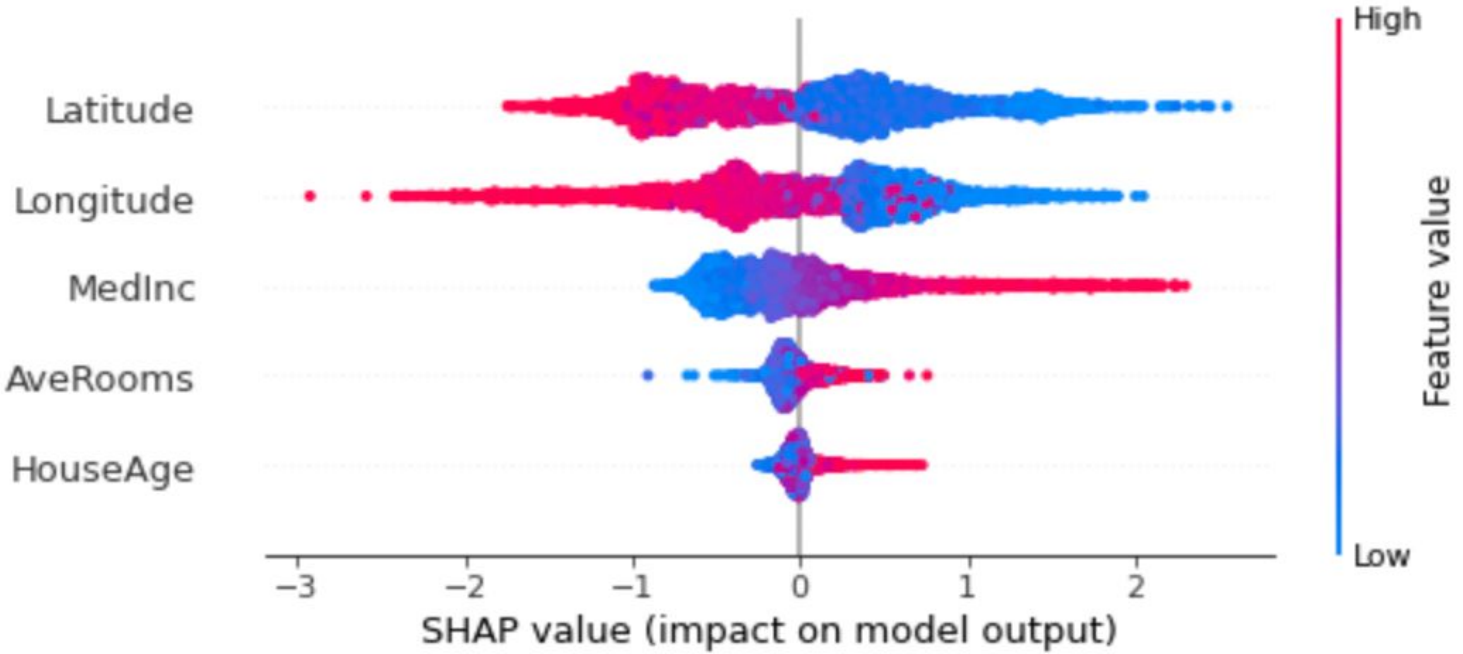


А как быть другим моделям?

Существуют инструменты, которые на основе статистических методов и матричных игр анализируют вклад входных параметров в итоговое предсказание, которое совершает произвольная модель

К таким инструментам относятся **коэффициенты Шепли (Shap)**, которые пытаются на основе теории матричных игр показать, насколько конкретный признак увеличил или уменьшил предсказание для данного объекта

График пчелиного роя / точечный



После построения модели и изучения графиков важности, можно отсеять признаки, которые вносят наименьший вклад в предсказание, и снова обучить модели, чтобы выявить эффект подхода