Методы отбора признаков

Дисциплина: Основы машинного обучения

Гребенкин Егор Дмитриевич

Группа: 5030102/20202

План презентации

- Введение и постановка задачи
- Классификация методов отбора признаков
- Filter-методы
- Wrapper-методы
- Embedded-методы
- Экспериментальное сравнение
- Выводы

Отбор признаков: определение

Отбор признаков (Feature Selection) — это процесс выбора подмножества исходных признаков, наиболее релевантных для решения задачи машинного обучения.

В отличие от feature extraction, отбор признаков:

- не создаёт новых признаков
- сохраняет физический и смысловой смысл исходных данных
- облегчает интерпретацию модели

Зачем нужен отбор признаков

Основные причины использования feature selection:

- **Проклятие размерности**: рост размерности ухудшает обобщающую способность моделей
- **Переобучение**: шумовые и коррелированные признаки увеличивают variance
- **Вычислительная сложность**: обучение становится медленным
- **Интерпретируемость**: сложнее анализировать вклад признаков

Отбор признаков и bias–variance tradeoff

Общая ошибка модели может быть представлена как:

$$\mathbb{E}[(y - \hat{f}(x))^2] = \underbrace{\text{Bias}^2}_{\text{смещение}} + \underbrace{\text{Variance}}_{\text{разброс}} + \underbrace{\sigma^2}_{\text{шум}}$$

Увеличение числа признаков:

- уменьшает bias
- увеличивает variance

Отбор признаков снижает variance за счёт уменьшения размерности входного пространства.

Отбор признаков и переобучение

При большом числе признаков:

- модель подстраивается под шум
- растёт чувствительность к обучающей выборке

Отбор признаков:

- уменьшает пространство гипотез
- стабилизирует оценки параметров
- снижает variance без сильного роста bias

Классификация методов отбора признаков

Все методы отбора признаков делятся на три основных класса:

- **Filter** — используют статистические критерии
- **Wrapper** — используют качество модели
- **Embedded** — отбор встроен в процесс обучения

Filter-методы: общая идея

Filter-методы оценивают каждый признак независимо от модели.

Основные характеристики:

- не используют обучение модели
- вычислительно эффективны
- масштабируются на большие данные

Недостаток: не учитывают взаимодействие признаков.

Filter-методы: формальная постановка

Для каждого признака X_j вычисляется скалярная оценка:

$$S_j = \mathcal{F}(X_j, y)$$

Далее признаки сортируются по S_j и выбираются:

- либо по порогу
- либо K лучших

Важно: каждый признак оценивается независимо.

Filter: Variance Threshold

Принцип действия: Удаляются признаки с дисперсией ниже заданного порога.

Мотивация: Если признак почти не меняется, он не помогает различать объекты.

Формула дисперсии:

$$\mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$$

Алгоритм:

1. Вычислить дисперсию каждого признака
2. Удалить признаки с $\mathrm{Var}(X_j) < \tau$

Filter: Корреляционный анализ

Корреляционный анализ измеряет зависимость признака X и целевой переменной y.

Коэффициент корреляции Пирсона:

$$r_{Xy} = \frac{\sum(X_i - \bar{X})(y_i - \bar{y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(y_i - \bar{y})^2}}$$

Алгоритм:

1. Вычислить корреляцию $r_{X_j y}$ для каждого признака
2. Отсортировать по $|r_{X_j y}|$
3. Выбрать K лучших признаков

Критерии:

- Pearson — линейная зависимость
- Spearman — монотонная зависимость, корреляция между рангами значений
- Kendall — мера ранговой корреляции, основанная на количестве согласованных и несогласованных пар, устойчива к выбросам
- ANOVA F-test — различие средних значений признака между классами

Filter: Mutual Information

Взаимная информация измеряет количество общей информации между признаком и целью.

Определение:

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

Алгоритм:

1. Оценить совместное распределение $p(x,y)$
2. Вычислить $I(X_j;Y)$ для каждого признака
3. Выбрать признаки с максимальной MI

Позволяет выявлять нелинейные зависимости.

Filter: Chi-Squared (χ^2)

Chi-Squared критерий оценивает зависимость между категориальным признаком и целью.

Статистика критерия:

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

Алгоритм:

1. Построить таблицу сопряжённости
2. Вычислить χ^2 для каждого признака
3. Отобрать признаки с максимальным значением

Применим только к неотрицательным дискретным данным.

Wrapper-методы: общая идея

Wrapper-методы используют модель машинного обучения для оценки качества подмножеств признаков.

Идея:

- перебор подмножеств признаков
- обучение модели
- выбор подмножества с максимальным качеством

Главный недостаток — высокая вычислительная сложность.

Wrapper-методы: вычислительная сложность

Wrapper-методы требуют обучения модели для каждого поднабора признаков.

Общая сложность:

$$O(N_{subsets} \cdot C_{model})$$

где C_{model} — стоимость обучения модели.

Даже жадные алгоритмы имеют квадратичную зависимость от d.

Wrapper: Recursive Feature Elimination (RFE)

RFE — рекурсивное удаление признаков на основе важности модели.

Алгоритм:

1. Обучить модель на текущем наборе признаков
2. Оценить важность признаков
3. Удалить наименее важный
4. Повторять до нужного числа признаков

Учитывает взаимодействие признаков, но вычислительно дорог.

Embedded-методы: общая идея

Embedded-методы выполняют отбор признаков непосредственно в процессе обучения модели.

Особенности:

- учитывают структуру модели
- менее затратны, чем wrapper
- обеспечивают баланс качества и скорости

Embedded: LASSO (L1-регуляризация)

LASSO добавляет штраф за абсолютные значения коэффициентов в функцию потерь.

Функция потерь:

$$\mathcal{L} = \mathcal{L}_{data} + \lambda \sum_{j=1}^{n} |w_j|$$

В результате оптимизации:

- часть коэффициентов становится равной нулю
- соответствующие признаки исключаются из модели

LASSO: геометрическая интерпретация

LASSO решает задачу:

$$\min_w \mathcal{L}(w) \quad \text{при} \quad \sum |w_j| \leq c$$

L1-ограничение имеет острые углы, что приводит к:

- точкам оптимума на осях
- занулению коэффициентов

Embedded: Random Forest

Random Forest — ансамбль деревьев решений с встроенной оценкой важности признаков.

Идеи:

- бутстрэп-выборки объектов
- случайный поднабор признаков
- усреднение предсказаний

Важность признаков:

$$G = 1 - \sum_k p_k^2$$

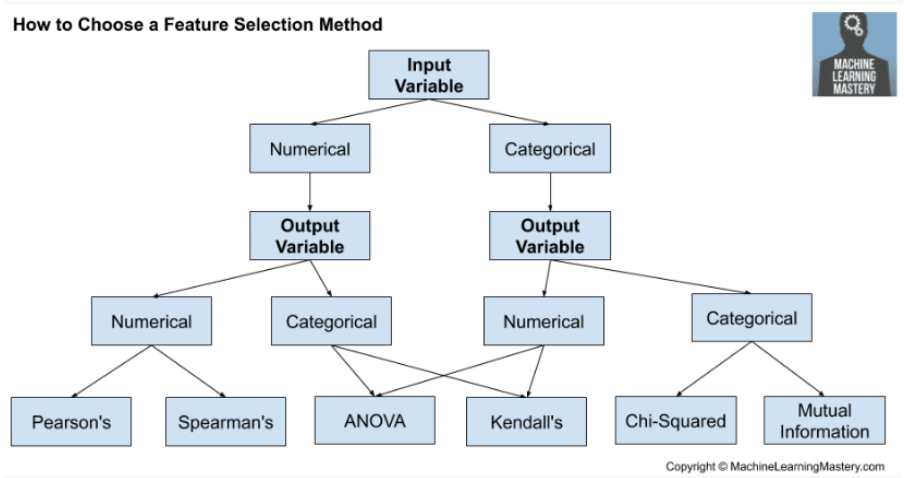
Определяется суммарным снижением Gini при разбиениях.

Сравнение вычислительной сложности

Метод	Учитывает модель	Сложность
Variance Threshold	Нет	$O(n \cdot d)$
Correlation	Нет	$O(n \cdot d)$
Mutual Information	Нет	$O(n \cdot d \cdot k)$
Chi-Squared	Нет	$O(n \cdot d)$
RFE	Да	$O(n \cdot d^2)$
LASSO	Да	$O(n \cdot d)$
Random Forest	Да	$O(T \cdot n \cdot d \log n)$

Обозначения: n — число объектов, d — число признаков, k — параметр оценки распределения (bins или k-NN), T — число деревьев в ансамбле.

Выбор метода отбора признаков в зависимости от типа данных



Эксперимент: постановка задачи

- Датасет: Breast Cancer (sklearn)
- Задача: бинарная классификация
- Модель: логистическая регрессия
- Метрики: Accuracy, F1-score, время работы

Сводная таблица результатов эксперимента

Метод	Accuracy	F1	Признаков	Время (с)
No FS	0.947	0.959	30	0.016
Variance Threshold	0.947	0.959	29	0.011
Correlation	0.930	0.946	10	0.006
ANOVA F-test	0.930	0.946	10	0.016
Chi-Squared	0.930	0.946	10	$\approx$0.000
Mutual Information	0.930	0.946	10	0.082
RFE	**0.953**	**0.964**	10	0.109
LASSO	0.947	0.960	**7**	0.016
Random Forest	0.930	0.946	10	0.311

Лучшие значения Accuracy и F1 выделены жирным.

Экспериментальный код: Filter-методы

Корреляционный анализ:

```
corr = np.abs(
    pd.DataFrame(X_train)
    .corrwith(pd.Series(y_train))
)
selected_corr = corr.sort_values(
    ascending=False
).head(10).index
```

Mutual Information:

```
from sklearn.feature_selection import mutual_info_classif

mi = mutual_info_classif(X_train, y_train)
selected_mi = np.argsort(mi)[-10:]
```

Экспериментальный код: Wrapper-метод (RFE)

Recursive Feature Elimination:

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression

rfe = RFE(
    estimator=LogisticRegression(
        max_iter=500
    ),
    n_features_to_select=10
)

rfe.fit(X_train, y_train)

X_train_rfe = rfe.transform(X_train)
X_test_rfe = rfe.transform(X_test)
```

RFE итеративно удаляет признаки с минимальной важностью.

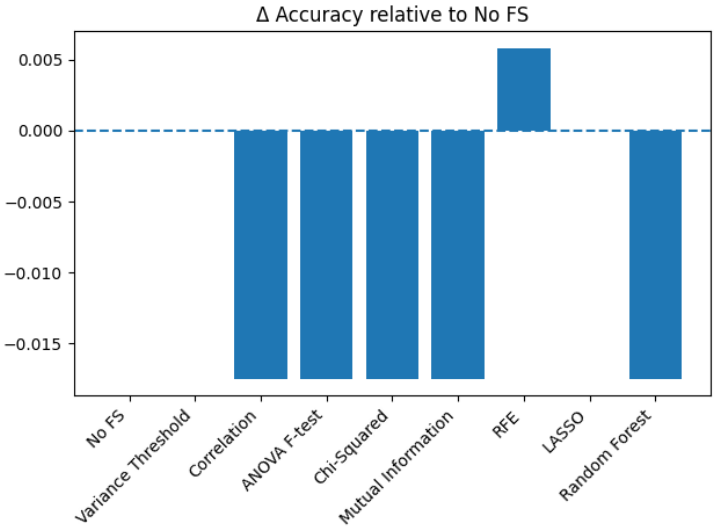
Экспериментальный код: Embedded-метод (LASSO)

Логистическая регрессия с L1-регуляризацией:

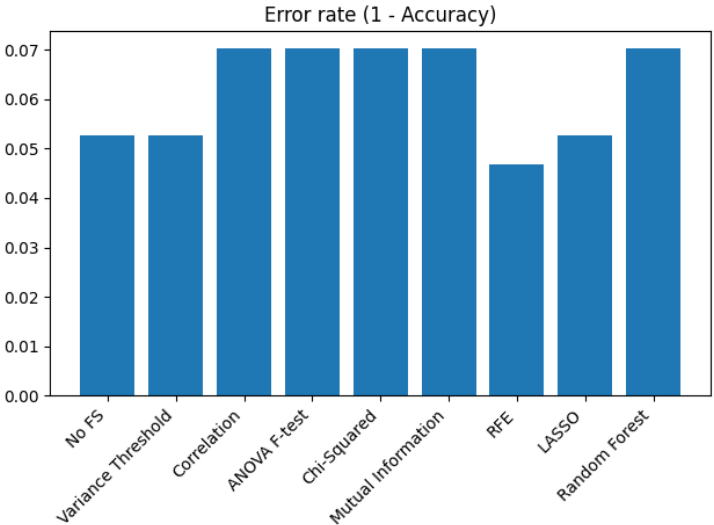
```
lasso = LogisticRegression(
    penalty="l1",
    solver="liblinear",
    max_iter=500
)

lasso.fit(X_train, y_train)

coef_mask = np.abs(
    lasso.coef_[0]
) > 1e-4

X_train_lasso = X_train[:, coef_mask]
X_test_lasso = X_test[:, coef_mask]
```

Занулённые коэффициенты соответствуют удалённым признакам.

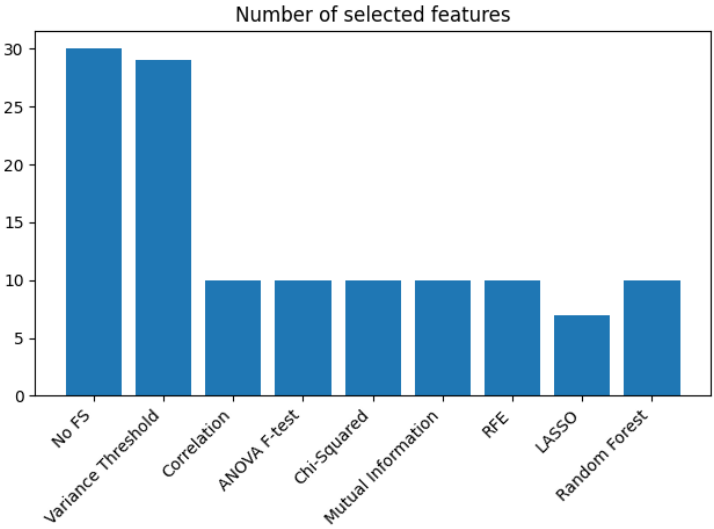
Изменение Accuracy относительно модели без отбора



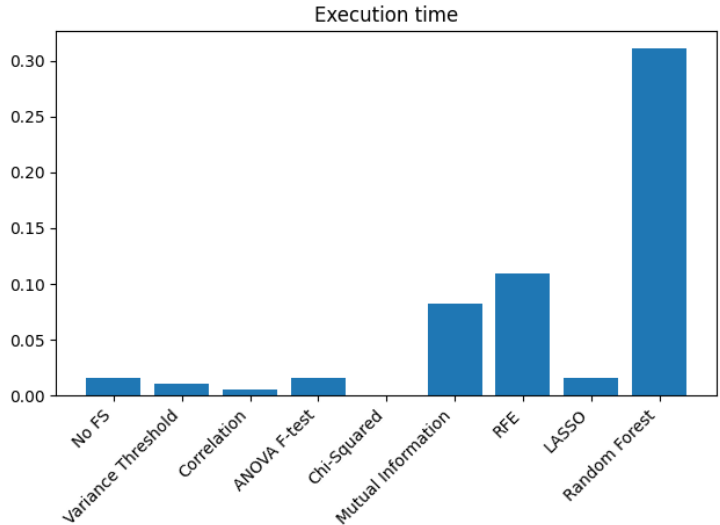
Ошибка классификации (1 – Accuracy)



Количество отобранных признаков



Время работы методов отбора признаков



Анализ результатов: общие наблюдения

- Использование всех признаков даёт качество чуть хуже RFE:

$$\text{Accuracy} = 0.947, \quad \text{F1} = 0.959$$

- Все методы отбора признаков приводят к незначительному снижению качества, но существенно уменьшают размерность.
- Потеря качества составляет

$$0.5\%–1.8\%$$

при сокращении числа признаков до 7–29.

Анализ результатов: сравнение методов

Filter-методы:

- Все filter методы показали примерно одинаковое качество:

$$\text{Accuracy} = 0.930$$

- Mutual Information работает значительно медленнее корреляции при том же качестве.
- Худшее качество связано с игнорированием взаимодействий признаков.

Wrapper (RFE):

- Accuracy = 0.953 при использовании 10 признаков
- Качество выше чем у модели без отбора
- Наибольшее время выполнения среди всех методов, не считая Random Forest

Embedded (LASSO):

- Accuracy = 0.947 при использовании 7 признаков
- Быстрый метод, дающий максимальный результат при минимальном количестве признаков
- Отбор происходит автоматически при обучении модели

Выводы

- отбор признаков снижает размерность с минимальными потерями качества: 0.5–1.8 процентна при сокращении числа признаков до 7–29
- wrapper-методы наиболее точные, но медленные
- embedded-методы наиболее практичны

Источники

1. scikit-learn developers. *Feature Selection*. Документация библиотеки scikit-learn. URL: https://scikit-learn.org/stable/modules/feature_selection.html
2. Блог Habr. *Отбор признаков в задачах машинного обучения*. URL: https://habr.com/ru/articles/550978/
3. Khan, S., et al. *A Comprehensive Study of Feature Selection Techniques in Machine Learning Models*. ResearchGate, 2024. URL: https://www.researchgate.net/publication/386160114
4. Chormunge, S., Jena, S. *Feature Selection: A Review and Comparative Study*. ResearchGate, 2022. URL: https://www.researchgate.net/publication/360811068