

# Лекция 4


# Оценка значимости признаков

Бручес Елена

15.07.2025

# Обо мне

В настоящее время:

 главный разработчик-исследователь, Сибирские Нейросети

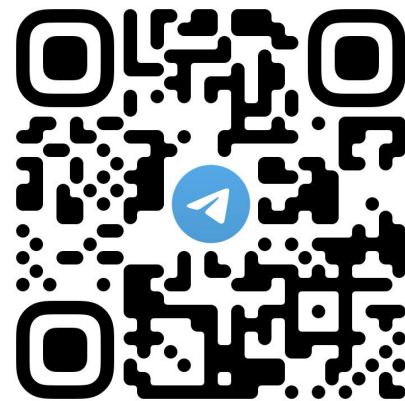
 К.Т.Н.

 м.н.с. ИСИ СО РАН

 старший преподаватель, ФипЛ Ги НГУ

 организатор DSTalks

 организатор DS мероприятий



# Важность отбора признаков

В эпоху больших данных модели машинного обучения всё чаще используются для выявления закономерностей и инсайтов из огромных наборов данных.

Однако эффективность этих моделей часто зависит от качества признаков, предоставляемых для обучения.

Отбор признаков — критический этап предварительной обработки, включающий выявление наиболее релевантных признаков набора данных и устранение избыточных или нерелевантных.

Этот процесс не только улучшает интерпретируемость модели, но также повышает её вычислительную эффективность и прогностические возможности.

# Важность отбора признаков

Высокая размерность данных, часто встречающаяся в областях, таких как геномика, обработка изображений и обработка естественного языка, представляет уникальные проблемы для алгоритмов машинного обучения.

Избыточные или нерелевантные признаки могут привести к переобучению, при котором модель хорошо работает на тренировочных данных, но плохо справляется с новыми данными.

Отбор признаков решает эту проблему путём снижения размерности, способствуя созданию устойчивых и обобщаемых моделей.

Более того, он помогает снизить вычислительные затраты и время, необходимое для тренировки модели, делая этот инструмент незаменимым как для исследователей, так и для практиков.

# Отбор признаков

Отбор признаков — это процесс идентификации и выбора наиболее значимых признаков из набора данных и удаления избыточных или нерелевантных. Это является ключевым этапом пайплайна машинного обучения, особенно при работе с данными высокой размерности. Уменьшая размерность данных, отбор признаков способствует повышению производительности модели, снижению вычислительных затрат и улучшению интерпретируемости.

Основная цель отбора признаков заключается в обеспечении того, чтобы модели машинного обучения фокусировались на наиболее информативных переменных, что ведёт к лучшей обобщаемости и точности предсказания.

Дополнительно, он упрощает модели, делает их легче для понимания и внедрения, снижая риск переобучения.

Отбор признаков особенно важен в таких областях, как геномика, финансы и обработка изображений, где наборы данных часто содержат тысячи признаков, многие из которых могут не вносить существенного вклада в целевой результат.

# Проблемы, связанные с выбором признаков в многомерных наборах данных

1. **Проклятие размерности.** По мере увеличения числа признаков пространство возможных подмножеств возрастает экспоненциально, что делает вычисление всех возможных комбинаций крайне ресурсоемким процессом.
2. **Избыточность и бесполезность признаков.** Многомерные наборы данных зачастую включают избыточные или неважные признаки, которые искажают значимые паттерны. Выявление и удаление таких признаков без потери важной информации является сложной задачей.
3. **Взаимодействие признаков.** Некоторые признаки могут обладать низкой предиктивной способностью сами по себе, однако существенно влиять на итоговую точность при сочетании с другими признаками. Традиционные методы выбора признаков часто игнорируют такие взаимодействия, приводя к неоптимальным результатам.
4. **Масштабируемость и вычислительная стоимость.** Многие подходы к выбору признаков требуют значительных вычислительных ресурсов. По мере роста объемов данных временные и материальные ресурсы, необходимые для осуществления выбора признаков, становятся неприемлемыми.

# Оценка значимости признаков

Методы оценки значимости признаков:

1. Filter methods
2. Wrapper Methods
3. Embedded Methods

# Filter Methods

Эти методы основываются на статистических показателях для оценки релевантности отдельных признаков относительно целевого признака.

Эти методы являются вычислительно эффективными и работают независимо от какого-либо конкретного алгоритма машинного обучения, что делает их популярным выбором для предварительной обработки многомерных данных.



# Filter Methods: Корреляции

Корреляция измеряет линейную связь между двумя переменными, обычно количественно оцениваемую с использованием коэффициента корреляции Пирсона.

Признаки с высокой корреляцией с целевым признаком считаются более релевантными.

Для регрессионных задач этот метод особенно эффективен в выявлении признаков, непосредственно влияющих на выходной результат. Тем не менее, он имеет ограничения в захвате нелинейных связей и может приводить к выбору избыточных признаков, если они сильно коррелируют друг с другом.

*Пример: В наборе данных для прогнозирования цен на жилье корреляция может выявить признаки, такие как площадь дома или близость расположения к удобствам, которые оказывают прямое линейное влияние на цену жилья.*

# Filter Methods: Chi-Square Test

Критерий хи-квадрат используется для оценки независимости категориальных признаков от целевого признака.

Он оценивает степень отклонения наблюдаемого распределения данных от ожидаемых частот при предположении о независимости.

Признаки с высокими значениями статистики хи-квадрат считаются более релевантными.

Данный показатель широко применяется в задачах классификации, где целевые и признаковые переменные имеют категориальную природу.

*Пример: В задаче категоризации клиентов анализ критерием хи-квадрат может выявить атрибуты, такие как возрастная группа клиента или категория покупок, которые тесно связаны с целевым классом.*

# Filter Methods: Chi-Square Test

```
>>> import numpy as np
>>> from sklearn.feature_selection import chi2
>>> X = np.array([[1, 1, 3],
...               [0, 1, 5],
...               [5, 4, 1],
...               [6, 6, 2],
...               [1, 4, 0],
...               [0, 0, 0]])
>>> y = np.array([1, 1, 0, 0, 2, 2])
>>> chi2_stats, p_values = chi2(X, y)
>>> chi2_stats
array([15.3,  6.5,  8.9])
>>> p_values
array([0.000456, 0.0387, 0.0116 ])
```

# Filter Methods: Mutual Information

Взаимная информация позволяет оценить зависимость между двумя переменными, фиксируя как линейные, так и нелинейные связи.

В отличие от корреляции, она не ограничена лишь линейными ассоциациями и эффективно применима как к численным, так и к категориальным признакам.

Признаки с высоким показателем взаимной информации вносят большой вклад в снижение неопределенности относительно целевого признака.

*Пример: В задачах классификации текстов взаимная информация способна выделять слова или фразы, которые значительно указывают на принадлежность документа конкретной категории.*

# Filter Methods: Достоинства

- Они обеспечивают высокую вычислительную эффективность и масштабируемость, поскольку оценивают признаки независимо от конкретных алгоритмов машинного обучения. Такая эффективность делает их особенно подходящими для многомерных наборов данных, позволяя быстро проводить предварительную обработку.
- Их независимость от специфики алгоритмов гарантирует универсальность применения в различных моделях машинного обучения, что делает их гибкими средствами анализа данных.
- Простота этих методов, таких как анализ корреляции и критерий хи-квадрат, облегчает интерпретацию результатов, обеспечивая представления о взаимосвязях между признаками и целевыми переменными.
- Благодаря отсутствию прямого взаимодействия с процедурой обучения модели, эти методы снижают риск переобучения, характерный для более сложных подходов к отбору признаков.

# Filter Methods: Недостатки

- Эти методы оценивают признаки индивидуально, игнорируя потенциальные взаимодействия между переменными. В результате они могут упустить комбинации признаков, которые совместно способствуют повышению эффективности модели.
- Многие используемые в фильтрах статистические меры, такие как коэффициент корреляции, ограничены обнаружением только линейных зависимостей, что приводит к пропуску важных нелинейных взаимодействий.
- Фильтры часто выбирают избыточные признаки, которые высоко коррелированы друг с другом, что добавляет ненужную сложность в набор признаков.
- Поскольку работа фильтров осуществляется независимо от используемой модели обучения, они могут выбрать признаки, которые необязательно улучшают прогностическую способность модели.

# Wrapper Methods

Такие методы подразумевают обучение и оценку модели машинного обучения для определения оптимального набора признаков.

Хотя подобные методы часто приводят к более высокой точности, они отличаются значительной вычислительной сложностью и могут плохо масштабироваться на большие объемы данных.

# Wrapper Methods: Recursive Feature Elimination (RFE)

Рекурсивное исключение признаков (Recursive Feature Elimination, RFE) подразумевает пошаговое удаление наименее значимых признаков и повторную оценку производительности модели на каждом этапе.

Процесс начинается с полного набора признаков и последовательно устраняет те, чей вклад в модель минимален, исходя из показателей типа коэффициентов признаков или значений важности.

Метод особенно эффективен в сочетании с алгоритмами вроде линейной регрессии, опорных векторов или деревьев решений.

Однако RFE требует значительных вычислительных ресурсов, особенно при обработке крупных наборов данных, поскольку предполагает многократное выполнение циклов обучения и оценки модели.



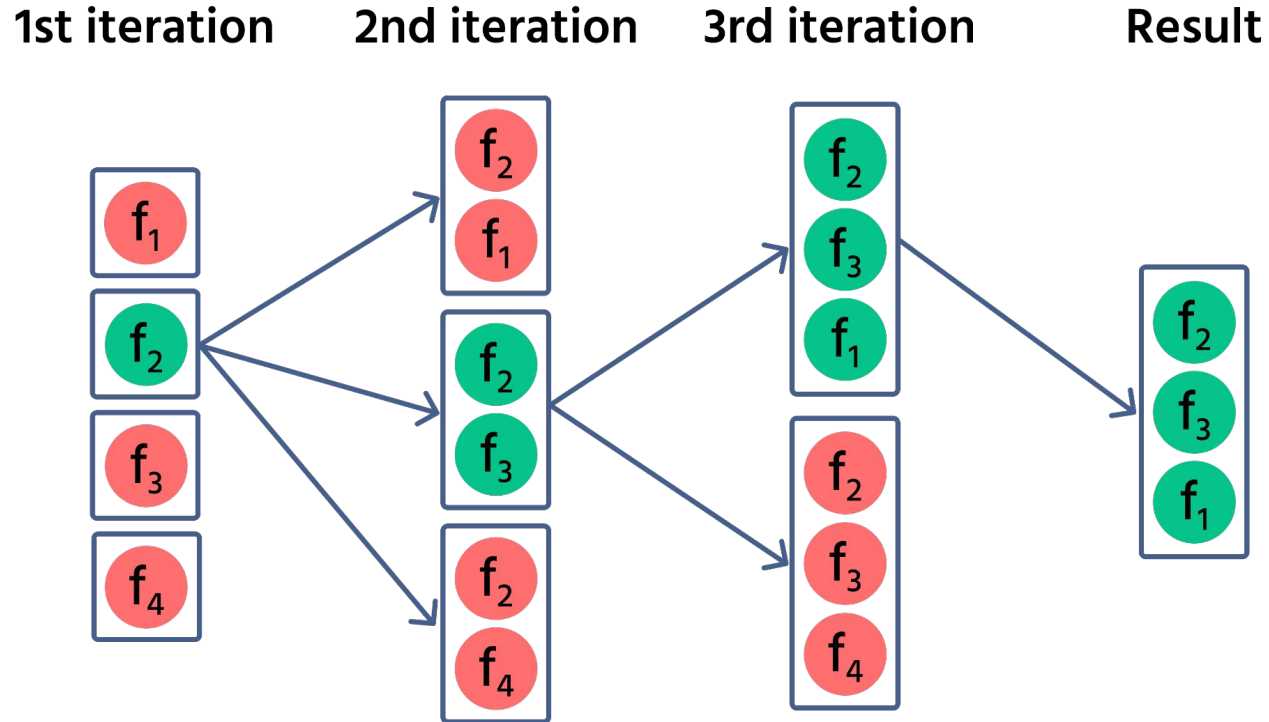
# Wrapper Methods: Forward Selection

Метод прямого подбора (Forward Selection) строит модель инкрементально, начиная с пустого набора признаков и постепенно добавляя один признак за раз.

На каждом шаге к подмножеству добавляется тот признак, который больше всего улучшает производительность модели.

Такой подход интуитивен и эффективен для нахождения минимального набора ключевых признаков, однако становится вычислительно дорогостоящим при большом количестве признаков.

# Wrapper Methods: Forward Selection



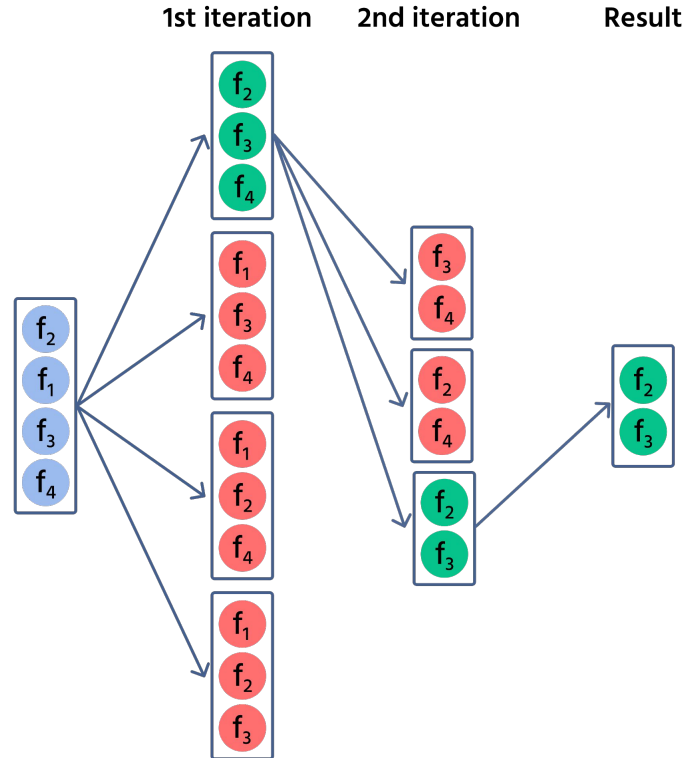
# Wrapper Methods: Backward Selection

Метод обратного подбора (Backward Selection) действует противоположным образом: сначала берётся весь набор признаков, а затем на каждой итерации удаляется наименее значимый признак.

Процесс продолжается до тех пор, пока дальнейшее удаление признаков не приведёт к заметному ухудшению производительности модели.

Несмотря на то, что обратный отсев обеспечивает всестороннюю оценку признаков, он весьма ресурсоёмкий, особенно для наборов данных с большим количеством признаков, так как требует многократного обучения модели.

# Wrapper Methods: Backward Selection



# Wrapper Methods: Trade-offs

Соотношение вычислительных затрат и производительности модели. Эти методы позволяют находить оптимальные подмножества признаков путем непосредственной оптимизации производительности целевой модели. Однако это достигается ценой высоких вычислительных требований, так как они предполагают многократное обучение и тестирование модели для разных вариантов признаков.

Для небольших и средних наборов данных такие методы способны обеспечить значительное повышение точности и способности к обобщению, выбирая признаки, которые в наибольшей степени влияют на производительность модели. Но по мере увеличения количества признаков и точек данных требования к ресурсам растут настолько, что использование этих методов становится невозможным без существенных вычислительных мощностей.

Практикующие специалисты часто решают этот компромисс, комбинируя эти методы с более быстрыми и менее требовательными методами, такими как фильтрация. Например, вначале может применяться фильтр-метод для сокращения исходного множества признаков, после чего оберточные методы применяются для дальнейшего уточнения выбора. Балансируя между вычислительной эффективностью и оптимизацией производительности, гибридные подходы позволяют успешно решать задачи выбора признаков даже в сложных наборах данных.

# Embedded Methods

Встраиваемые методы интегрируют отбор признаков непосредственно в сам процесс обучения модели, что делает их одновременно эффективными и производительными.

В отличие от фильтровых и оберточных методов, работающих независимо или вне самой модели, встроенные методы осуществляют отбор признаков прямо в процессе обучения модели.

Благодаря этому выбранные признаки напрямую содействуют повышению предсказательной способности модели, сохраняя при этом вычислительную эффективность.

# Embedded Methods: Lasso

Одним из наиболее распространённых встроенных методов является LASSO (Least Absolute Shrinkage and Selection Operator) — техника регуляризации для линейных моделей.

LASSO накладывает штраф  $L_1$  на коэффициенты регрессии, уменьшая значения незначимых коэффициентов до нуля, фактически исключая несущественные признаки.

Это делает LASSO особенно полезным для многомерных наборов данных, где большое количество признаков может иметь минимальное воздействие на целевую переменную.

# Embedded Methods: Lasso

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(penalty='l1', C=1.0, solver='liblinear', multi_class='ovr')
# Note that C=1.0 is the default. You can increase
# or decrease it to make the regularization effect
# stronger or weaker, respectively.
lr.fit(X_train_std, y_train)
print('Training accuracy:', lr.score(X_train_std, y_train))
print('Test accuracy:', lr.score(X_test_std, y_test))
```

Training accuracy: 1.0

Test accuracy: 1.0

```
lr.intercept_
```

```
array([-1.26363107, -1.21610924, -2.37035486])
```

```
np.set_printoptions(8)
```

```
lr.coef_[lr.coef_!=0].shape
```

```
(23,)
```

```
lr.coef_
```

```
array([[ 1.2455466 ,  0.18072432,  0.74213192, -1.15948629,  0.          ,
         0.          ,  1.17434899,  0.          ,  0.          ,  0.          ,
         0.          ,  0.54353185,  2.51127873],
       [-1.53786975, -0.38667962, -0.9954337 ,  0.36456123, -0.05923747,
         0.          ,  0.66763266,  0.          ,  0.          , -1.93321837,
         1.23529768,  0.          , -2.23229101],
       [ 0.13570425,  0.16821283,  0.35724291,  0.          ,  0.          ,
         0.          , -2.43809231,  0.          ,  0.          ,  1.56377541,
        -0.81940109, -0.49234846,  0.          ]])
```



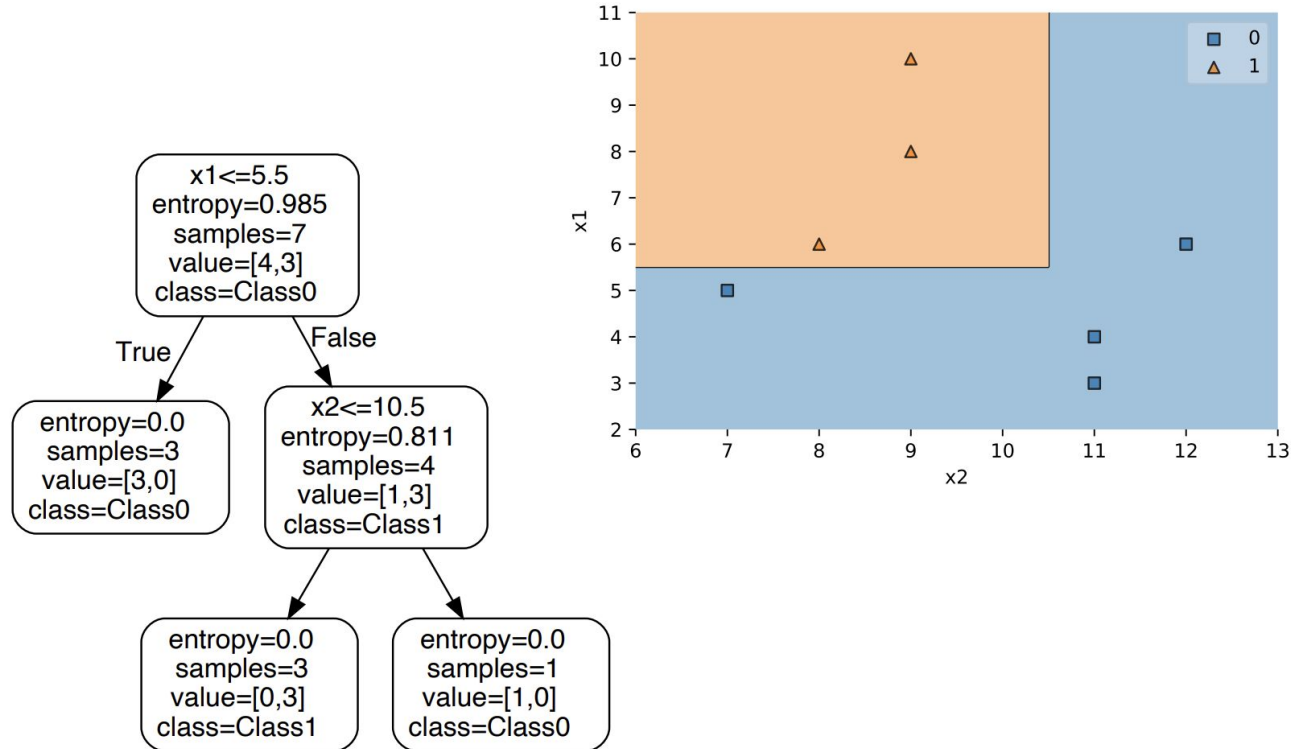
# Embedded Methods: Tree-Based methods

Методы на основе деревьев, такие как решающие деревья, случайные леса и градиентный бустинг, естественным образом выполняют отбор признаков в ходе процесса обучения.

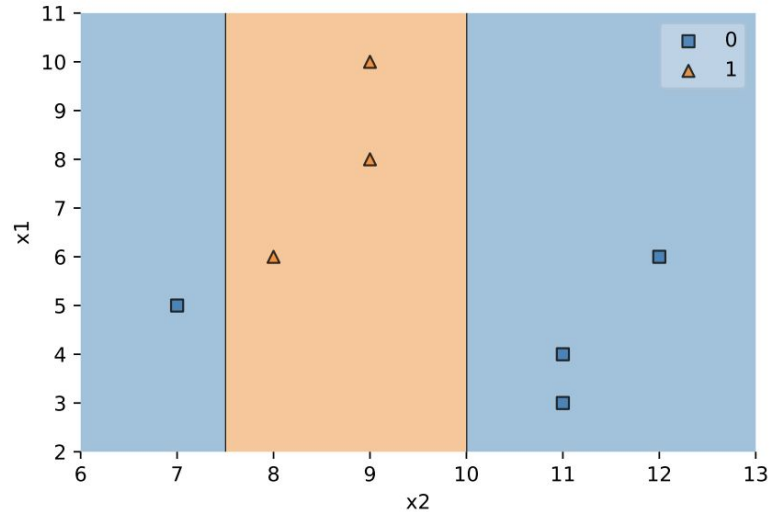
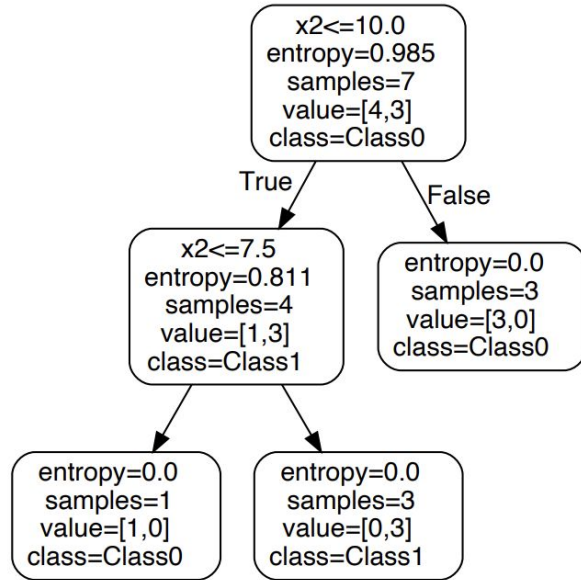
Данные алгоритмы присваивают признакам показатели важности, основываясь на их вкладе в расщепления в процессе принятия решений.

Признаки, которые часто используются в раскрытиях узлов дерева или заметно уменьшают меру неоднородности, считаются более важными, тогда как менее влиятельные признаки практически игнорируются.

# Embedded Methods: Tree-Based methods



# Embedded Methods: Tree-Based methods



# Embedded Methods: Tree-Based methods

```
from sklearn.ensemble import RandomForestClassifier

forest = RandomForestClassifier(n_estimators=500,
                              random_state=1)

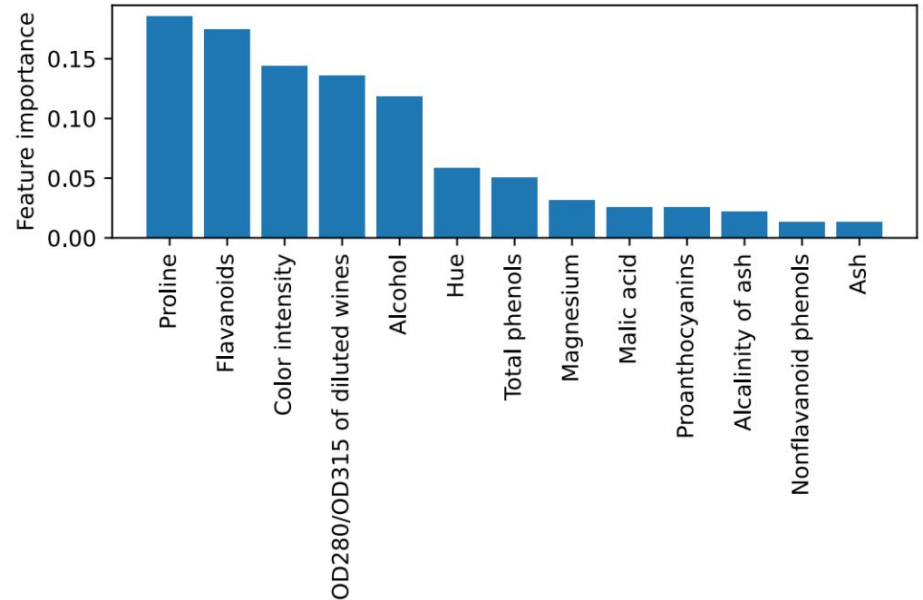
forest.fit(X_train, y_train)
importances = forest.feature_importances_

indices = np.argsort(importances)[-1]

plt.ylabel('Feature importance')
plt.bar(range(X_train.shape[1]),
        importances[indices],
        align='center')

feat_labels = df_wine.columns[1:]
plt.xticks(range(X_train.shape[1]),
           feat_labels[indices], rotation=90)

plt.xlim([-1, X_train.shape[1]])
```



# Embedded Methods: Примеры использования

Embedded методы находят широкое применение в различных сферах благодаря своей способности осуществлять отбор признаков в процессе обучения модели, гарантируя таким образом как эффективность, так и значимость. Например, в финансовом секторе регрессия методом LASSO активно используется в кредитном скоринге. Применяя штраф L1 для уменьшения коэффициентов малозначимых признаков до нуля, LASSO выделяет ключевые предикторы, такие как доход, кредитная история и отношение долга к доходу, исключая признаки с минимальным влиянием. Это приводит к компактным, легко интерпретируемым моделям, сохраняющим высокую точность прогнозирования.

Алгоритмы на основе деревьев, такие как случайные леса и градиентный бустинг, получили обширное распространение в здравоохранении для прогнозирования исходов пациентов или диагностики заболеваний. Так, в задачах прогнозирования диабета эти методы выделяют важные признаки, такие как уровень глюкозы, индекс массы тела и возраст пациента, автоматически исключая менее значимые переменные. Интегрированный процесс отбора признаков снижает шум и повышает надёжность модели.

В задачах классификации изображений алгоритмы на основе деревьев, такие как XGBoost, ранжируют пиксели по интенсивности цвета в зависимости от их вклада в классификацию изображений по категориям. Это позволяет модели сосредоточиться на наиболее информативных пикселях, улучшая как точность, так и вычислительную эффективность. Прямая интеграция отбора признаков с обучением модели позволяет встроенным методам упростить разработку прогнозирующих моделей в самых разных областях, что делает их чрезвычайно полезными в реальных приложениях.

# Оценка значимости признаков

Method	Advantages	Limitations	Examples
Filter Methods	Computationally efficient, easy to implement	Ignores feature interactions	Correlation, Chi-square
Wrapper Methods	Considers feature interactions	High computational cost	RFE, Forward Selection
Embedded Methods	Integrated into model training	Model-dependent	LASSO, Tree-based methods

# Сравнительный анализ подходов

Filter методы, такие как корреляционный анализ и критерий хи-квадрат, характеризуются высокой вычислительной эффективностью и идеально подходят для начальных этапов отбора признаков, особенно при работе с многомерными данными.

Однако они могут упускать из виду важные взаимодействия признаков и нелинейные отношения, что способно привести к субоптимальному набору признаков.

Например, в сфере здравоохранения, где взаимодействие признаков часто играет ключевую роль, полагаться исключительно на фильтровые методы может оказаться недостаточным для полноты отражения сложности данных.

# Сравнительный анализ подходов

Wrapper методы, как правило, предоставляют лучшие результаты, учитывая взаимодействие между признаками и производительностью модели.

Такие методы могут формировать высокоточные подмножества признаков, особенно при использовании вместе с алгоритмами машинного обучения, такими как машины опорных векторов или случайные леса.

Однако их вычислительная стоимость может стать препятствием, особенно для больших наборов данных, так как они требуют многократного обучения и тестирования моделей.



# Сравнительный анализ подходов

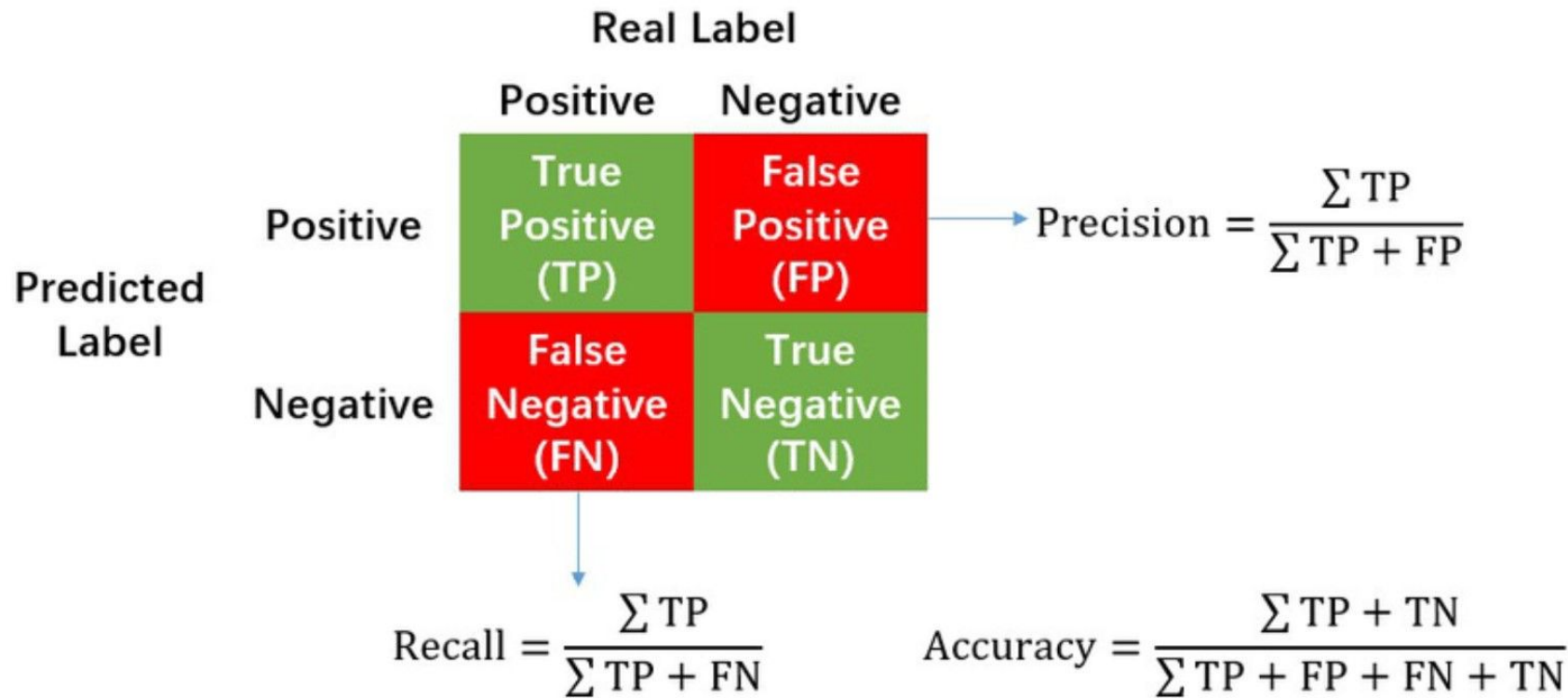
Embedded методы, такие как LASSO и методы на основе деревьев, находят баланс между вычислительной эффективностью и производительностью модели.

Они объединяют отбор признаков непосредственно в процесс обучения, сокращая необходимость отдельного этапа выбора признаков.

На практике embedded методы, такие как LASSO, часто предпочитают в ситуациях, где важны как точность, так и интерпретируемость, например, в финансовой аналитике или медицине.

Методы на основе деревьев, напротив, особенно эффективны при работе с категориальными данными и сложными взаимодействиями между признаками, что делает их пригодными для областей, таких как обработка изображений.

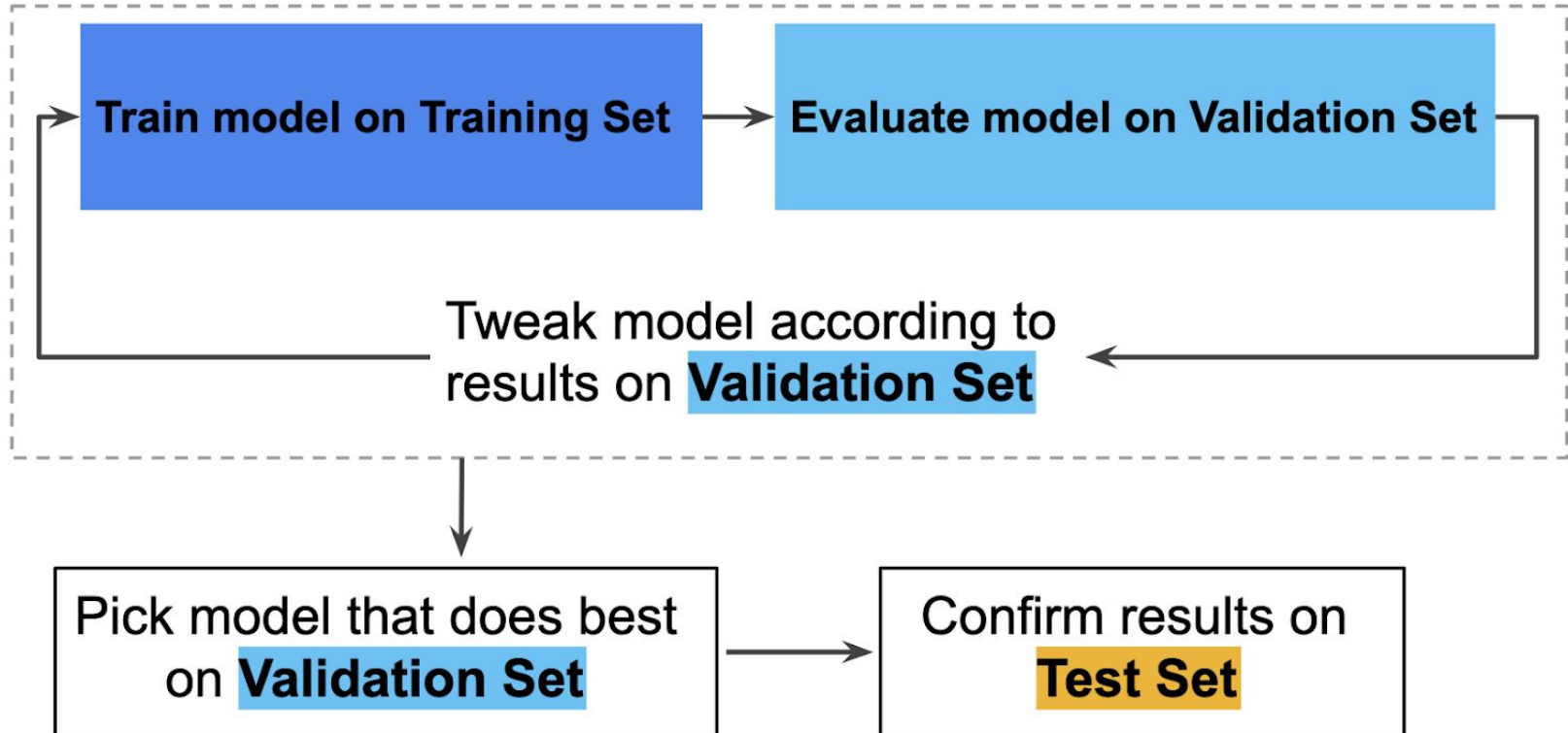
# Методы для оценки эффективности подходов



# Методы для оценки эффективности подходов



# Train/val/test splitting



# Cross-validation

Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 1
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 2
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 3
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 4
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 5

Training data

Test data

# Методы для оценки эффективности подходов

Вычислительная эффективность играет важную роль при сравнительном анализе. Время и ресурсы, требуемые для обработки набора данных с помощью техники отбора признаков, должны учитываться наряду с эффектом влияния на производительность. Методы типа фильтрации обычно быстрее, но могут не обеспечивать такую же точность, как более ресурсоемкие методы, такие как оберточные или встроенные методы.

Ранжирование важности признаков может использоваться для сравнения методов отбора признаков путем изучения того, какие именно признаки выбраны различными методами и насколько эти отобранные признаки согласуются с имеющимися знаниями в соответствующей области. Если метод отбора признаков постоянно выбирает признаки, известные как важные в данной области, его можно считать эффективным.

# Обобщаемость и устойчивость методов

Обобщаемость и устойчивость являются ключевыми аспектами, которые необходимо учитывать при оценке эффективности методов отбора признаков.

Метод считается **обобщающим**, если выбранные признаки стабильно повышают производительность модели на различных наборах данных или в разных прикладных областях. Это особенно важно в реальных приложениях, где модели, обученные на одном наборе данных, должны демонстрировать хорошие результаты при применении к новым, ранее неизвестным данным.

Например, метод отбора признаков, определяющий характеристики, связанные с постановкой диагноза заболевания в одной популяции, должен, в идеале, выбирать аналогичные признаки при применении к другой популяции, при условии, что фундаментальные взаимоотношения между признаками и заболеванием остаются неизменными.

Способность к обобщению гарантирует, что метод отбора признаков не адаптируется чрезмерно к уникальным особенностям обучающего набора данных.

# Обобщаемость и устойчивость методов

**Устойчивость** (robustness) относится к способности выбранных признаков сохранять свою значимость и качество работы даже при наличии небольших изменений или шума в наборе данных.

Устойчивый метод отбора признаков должен уметь справляться с вариациями в данных — такими как пропущенные значения, выбросы или шумы — без значительного ухудшения производительности модели.

Например, если набор данных претерпевает небольшие изменения, такие как добавление новых образцов или введение незначительных колебаний в значения признаков, устойчивый метод отбора признаков должен продолжать выбирать те же самые или очень похожие признаки, что обеспечит стабильные прогнозы модели.



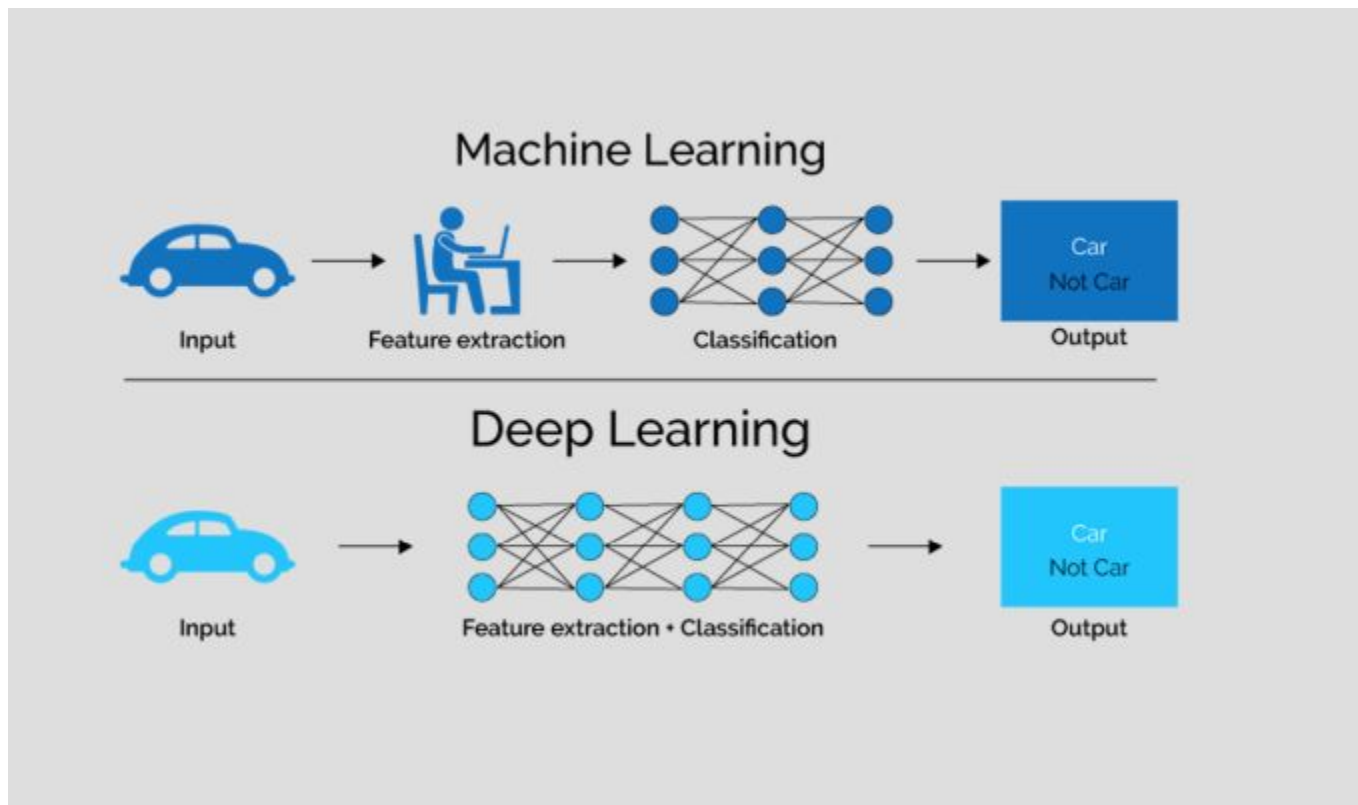
# Обобщаемость и устойчивость методов

Чтобы оценить обобщаемость и устойчивость методов отбора признаков, важно тестировать выбранные признаки на различных наборах данных или посредством перекрестной проверки (кросс-валидации).

Техники, такие как бутстрэппинг (bootstrap sampling), когда различные подмножества набора данных многократно извлекаются случайным образом с возвращением элементов обратно, могут использоваться для оценки стабильности и надежности выбранных признаков при различных изменениях данных. Это гарантирует, что выбранные признаки не привязаны излишне конкретно к одному определённому набору данных и могут адаптироваться к широкому спектру условий, что ведет к построению более надежных и масштабируемых моделей.

Подводя итоги, обобщаемость и устойчивость методов отбора признаков играют важнейшую роль в обеспечении эффективной работы моделей, созданных на основе выбранных признаков, в реальных сценариях и адаптации к новым данным. Методы отбора признаков, способные поддерживать высокий уровень производительности и обладая указанными качествами, скорее всего, принесут максимальную пользу в практических приложениях.

# Глубокое обучение и важность признаков



# Глубокое обучение и важность признаков

Модели глубокого обучения, особенно использующие нейронные сети, способны автоматически учиться важным признакам из необработанных данных в процессе обучения, обходя необходимость ручного отбора признаков.

В глубоком обучении автоэнкодеры, особый вид нейронных сетей, могут использоваться для отбора признаков, изучая сжатое, низкоразмерное представление входных данных. Автоэнкодер определяет наиболее важные признаки, минимизируя ошибку между входными данными и их восстановленной версией, эффективно сосредотачиваясь на признаках, которые вносят наибольший вклад в структуру данных. Этот подход особенно полезен для многомерных наборов данных, таких как изображения, аудиофайлы и тексты, где сырые входные данные могут содержать тысячи или даже миллионы признаков.

# Глубокое обучение и важность признаков

Свёрточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN), которые широко применяются соответственно в обработке изображений и последовательностей, также имплицитно выполняют отбор признаков, сосредоточиваясь на наиболее релевантных шаблонах или структурах внутри данных. Например, в CNN слои свёртки автоматически извлекают важные пространственные признаки, в то время как RNN акцентирует внимание на временных признаках в последовательных данных.

Несмотря на способность моделей глубокого обучения автоматически извлекать признаки, перед ними стоят некоторые трудности, включая потребность в больших объёмах размеченных данных и высокие вычислительные потребности. Однако, по мере развития архитектур нейронных сетей и методов обучения, модели глубокого обучения смогут становиться ещё эффективнее в выборе признаков, поддерживая высокую точность и открывая новые направления для отбора признаков в сложных наборах данных.

# LIME

*"Why Should I Trust You?": Explaining the Predictions of Any Classifier. 2016*

## Выборка локальных данных

Исходная точка, для которой требуется интерпретация, выбирается из набора данных. Это может быть любой пример (например, строка данных для табличных данных, изображение, текст и т.д.), для которого было сделано предсказание сложной модели.

Эта точка служит центром локального анализа и основой для генерации новых, возмущенных примеров данных. Возмущения создаются путем внесения небольших изменений в исходные данные. Эти изменения могут быть реализованы различными способами, в зависимости от типа данных:

- Для табличных данных: Изменение значений признаков, добавление шума, использование техник, таких как бутстраппинг.
- Для текстовых данных: Удаление или замена слов, изменение частей текста.
- Для изображений: Изменение пикселей, применение фильтров, небольшие искажения.

Цель: создать набор данных, который отражает возможные вариации вокруг выбранной исходной точки.

# LIME

## Прогнозы сложной модели

Сложная модель, которую необходимо интерпретировать, может быть любого типа — от нейронной сети до сложного ансамбля моделей. Главное, чтобы эта модель была уже обучена и способна делать предсказания.

LIME не требует изменений в самой сложной модели или ее внутренней структуре, что делает метод универсальным и применимым к широкому спектру моделей.

Сложная модель применяется к каждому возмущенному примеру данных, сгенерированному на предыдущем шаге. Для каждого примера записывается предсказание сложной модели. Эти предсказания покажут, как модель реагирует на изменения в данных.

Анализируя, как меняются предсказания в ответ на возмущения, можно получить представление о том, какие факторы влияют на решения модели. Определенные изменения в данных могут сильно изменять предсказания модели, указывая на важность этих факторов в процессе принятия решений.

Результаты предсказаний вместе с соответствующими весами и исходными данными возмущенных примеров используются для обучения интерпретируемой модели на следующем этапе процесса LIME.

# LIME

## Обучение интерпретируемой модели

В качестве интерпретируемой модели обычно используются простые модели, такие как линейные регрессии или деревья решений.

Линейные модели предпочтительны за их прозрачность и понятность, ведь их коэффициенты могут быть напрямую интерпретированы как важность признаков.

Для обучения интерпретируемой модели используются данные, сгенерированные на предыдущих шагах, включая возмущенные примеры и их соответствующие веса.

Модель обучается таким образом, чтобы как можно лучше приблизить предсказания сложной модели для данного набора локальных данных.

В случае линейной модели, коэффициенты можно интерпретировать как важность признаков. Положительные коэффициенты указывают на увеличение вероятности прогноза с увеличением значения признака, и наоборот для отрицательных коэффициентов.

# LIME

## Интерпретация результатов

LIME предоставляет интерпретируемые модели (обычно линейные), которые показывают, как изменение входных признаков влияет на предсказание модели.

Результаты LIME являются локальными, то есть они объясняют предсказание модели для конкретного примера или наблюдения, а не для всего набора данных.

В случае линейной модели коэффициенты могут быть положительными или отрицательными, указывая на то, как признак влияет на предсказание (положительно или отрицательно). Величина коэффициентов указывает на степень важности признаков. Большие абсолютные значения коэффициентов свидетельствуют о большем вкладе признака в предсказание.



# SHAP

[A Unified Approach to Interpreting Model Predictions, 2017](#)

SHAP (SHapley Additive exPlanations) — это метод интерпретации машинного обучения, который объясняет вклад каждого признака в предсказание конкретного наблюдения. Он основывается на концепции значений Шепли — методе из кооперативной теории игр, который распределяет "выигрыш" (или влияние) между всеми участниками игры (в контексте SHAP — между признаками модели).

В контексте машинного обучения, значением Шепли признака является среднее маргинальное вклады этого признака в предсказание модели по всем возможным комбинациям других признаков. Этот подход гарантирует справедливое распределение вклада между всеми признаками.

SHAP присваивает каждому признаку отдельное значение, которое можно суммировать для получения общего вклада всех признаков в предсказание модели.

# SHAP

## Выбор наблюдения для анализа

В зависимости от цели анализа, наблюдение может быть выбрано из-за его необычности (например, ошибочное предсказание), из-за его типичности (представительный случай) или по другим причинам, важным для исследования.

Наблюдение выбирается из того же набора данных, на котором обучена модель. Это может быть часть обучающей выборки, тестовой выборки или новые данные. Важно, чтобы выбранное наблюдение было совместимо с моделью, т.е. содержало все необходимые признаки в правильном формате.

Сначала определяется контекст анализа. Например, если интересует объяснение ошибочных предсказаний, то выбираются такие наблюдения из тестовых данных. Часто полезно анализировать распределение данных и выбирать наблюдения, которые представляют интересные случаи, такие как выбросы или крайние точки.

Могут использоваться различные метрики (например, вероятность, предоставленная моделью, или величина ошибки) для определения наиболее подходящих наблюдений для анализа. В некоторых сценариях может использоваться случайный выбор наблюдений для получения общего представления о поведении модели.

# SHAP

## Создание "коалиций" признаков

В SHAP, коалиция представляет собой подмножество всех доступных признаков в наборе данных. Каждая коалиция образует уникальную комбинацию признаков.

Коалиции используются для моделирования различных сценариев, при которых определенные признаки присутствуют или отсутствуют в данных, позволяя оценить их индивидуальный и коллективный вклад в предсказание модели.

Для каждого наблюдения создаются все возможные комбинации признаков. Это включает сценарии, где некоторые признаки присутствуют, а некоторые отсутствуют.

Для набора из  $N$  признаков, это означает создание  $(2^N)$  различных подмножеств, включая пустое множество (где отсутствуют все признаки) и полное множество (где присутствуют все признаки).

Из-за экспоненциального роста количества комбинаций при увеличении числа признаков, часто используются приближенные методы для оценки вклада признаков без полного перебора всех комбинаций.

Полный перебор всех возможных коалиций может быть вычислительно затратным, особенно для больших наборов данных с множеством признаков.

# SHAP

## Расчет вклада каждого признака

Для каждой коалиции признаков вычисляется вклад признака, который оценивает, насколько изменяется предсказание модели при добавлении этого признака к коалиции.

Производится перебор всех возможных коалиций признаков, и для каждой из них вычисляется, как изменится предсказание модели при добавлении рассматриваемого признака.

Вычисляется среднее взвешенное влияние признака на предсказание модели, учитывая все возможные коалиции. Это среднее и является значением Шепли для данного признака.

Значение Шепли признака  $i$  для предсказания  $f$  определяется формулой:

- $N$  — полный набор признаков,
- $S$  — подмножество признаков без  $i$ ,
- $f(S)$  — предсказание модели с признаками  $S$ ,
- $f(S \cup \{i\})$  — предсказание модели с признаками  $S$  плюс признак  $i$ .

$$[\phi_i(f)] = \sum_{S \subseteq N \setminus i} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup i) - f(S)]$$

# SHAP

## Агрегирование результатов

Значения Шепли для всех признаков суммируются для каждого наблюдения, чтобы оценить их коллективный вклад в предсказание модели. Агрегированные данные предоставляют информацию о том, какие признаки в целом имеют наибольшее и наименьшее влияние на предсказания модели.

Часто используется среднее значение вкладов признаков по всем наблюдениям для понимания общего влияния признака в разных сценариях.

# Выводы

1. Определение важности признаков помогает для интерпретации работы модели, снижения размерности и оптимизации работы модели.
2. Классические методы: Filter, Wrapper, Embedded.
3. Новые методы: LIME, SHAP.
4. Оценка качества: по метрикам модели на задачах, важно учитывать способность к генерализации и устойчивость к шумам.

**ПРЕЗЕНТАЦИЯ  
ОКОНЧЕНА**

**СПАСИБО ЗА  
ВНИМАНИЕ!**