Задача отбора признаков

**Методы отбора признаков при моделировании**

Целевые признаки (feature), используемые для обучения модели, оказывают большое влияние на качество результатов. Неинформативные или слабо информативные признаки могут существенно понизить эффективность и точность многих моделей, особенно линейных, таких как линейная и логистическая регрессия. После устранения или преобразования неинформативных/слабо информативных признаков модель упрощается, и соответственно уменьшается размер набора данных в памяти и ускоряется работа алгоритмов ML на нем.

Можно выделить 3 задачи:

* **feature selection** – отсечение ненужных признаков (избыточных, слабо информативных) и выбор признаков, имеющих наиболее тесные взаимосвязи с целевой переменной.
* **feature extraction and feature engineering** – превращение данных, специфических для предметной области, в понятные для модели векторы;
* **feature transformation** – трансформация данных для повышения точности алгоритма (например, нормализация данных).

Каждая из этих задач направлена на обеспечение следующих преимуществ:

* **Уменьшение переобучения.** Чем меньше избыточных данных, тем меньше возможностей для модели принимать решения на основе «шума».
* **Повышение точности предсказания модели.** Чем меньше противоречивых данных, тем выше точность.
* **Сокращение времени обучения.** Чем меньше данных, тем быстрее обучается модель.
* **Увеличивается семантическое понимание модели.**

 Выделяют методы ручного и автоматизированного отбора признаков.

Методы ручного отбора признаков основаны на содержательном анализе каждого признака и их совокупности, когда решение о включении/исключении признака из модели принимает исследователь.

В данном разделе рассмотрены различные методы автоматизированного отбора признаков (feature selection), применяемые для подготовки данных. Они могут быть реализованы с помощью Python и библиотеки scikit-learn. Подробное руководство по отбору признаков с помощью scikit-learn вы можете найти в документации к этой библиотеке в разделе Feature selection.

Методы отбора признаков делятся на три группы:

1) методы-фильтры (filters) - они оценивают признаки только на основе информации, полученной из обучающей выборки. Применяются на этапе предобработки, до запуска алгоритма обучения;

2) методы-обертки (wrappers) - классификатор запускается на конкретных подмножествах обучающей выборки, а затем выбирается подмножество наиболее информативных для обучения признаков;



Рисунок 1 - Процесс работы оберточных методов

3) встроенные методы (embedded) - позволяют не отделять отбор признаков и обучение классификатора. Данный тип методов также обладает рядом других преимуществ: они хорошо приспособлены к конкретной модели; не требуется выделять специальное подмножество для тестирования, как в предыдущих методах, и, как следствие из этого, меньше риск переобучения.



Рисунок 2 - Процесс работы встроенных методов

Рассмотрим подробнее каждую группу.

**1 Методы – Фильтры**(filters)

**Фильтры** (англ. *filter methods*) измеряют релевантность признаков на основе функции μμ, и затем решают по определенному правилу, какие признаки оставить в результирующем множестве.

Фильтры могут быть:

* Одномерные (англ. *univariate*) — функция μμ определяет релевантность одного признака по отношению к целевой переменной. В таком случае обычно измеряют "качество" каждого признака с помощью, например, статистических критериев (коэффициент ранговой корреляции Спирмена, Information gain и др.) и удаляют худшие. Например, библиотека scikit-learn содержит класс SelectKBest, реализующий одномерный отбор признаков (univariate feature selection). Этот класс можно применять совместно с различными статистическими критериями для отбора заданного количества признаков.
* Многомерные (англ. *multivariate*) — функция μμ определяет релевантность некоторого подмножества исходного множества признаков относительно выходных меток.

Преимуществом группы фильтров является простота вычисления релевантности признаков в наборе данных, но недостатком в таком подходе является игнорирование возможных зависимостей между признаками.

Принцип методов обертки состоит в следующем:

• выполняется поиск по пространству подмножеств исходного множества признаков;

• для каждого шага поиска используется информация о качестве обучения на текущем подмножестве признаков (в качестве функции оценки качества обучения на текущем подмножестве признаков часто используется точность классификатора).

Этот процесс является циклическим и продолжается до тех пор, пока не будут достигнуты заданные условия останова. Оберточные методы учитывают зависимости между признаками, что является преимуществом по сравнению с фильтрами, к тому же показывают большую точность, но вычисления занимают длительное время, и повышается риск переобучения.

2.Методы-обертки (Wrapper methods)

Существует несколько типов оберточных методов: детерминированные, которые изменяют множество признаков по определенному правилу, а также рандомизированные, которые используют генетические алгоритмы для выбора искомого подмножества признаков. Среди детерминированных алгоритмов самыми простыми являются алгоритмы последовательного поиска:

* SFS (Sequential Forward Selection) — жадный алгоритм, который начинает с пустого множества признаков, на каждом шаге добавляя лучший из еще не выбранных признаков в результирующее множество. На первом этапе производительность классификатора оценивается по отношению к каждому признаку. Выбирается признак, который работает лучше всего. На втором этапе первый признак пробуется в сочетании со всеми другими функциями. Выбирается комбинация двух функций, обеспечивающих наилучшую производительность алгоритма. Процесс продолжается до тех пор, пока не будет выбрано указанное количество признаков.
* SBS (Sequential Backward Selection) — алгоритм обратный SFS, который начинает с изначального множества признаков, и удаляет по одному или несколько худших признаков на каждом шаге; следует отметить, что эмпирически обратный жадный алгоритм даёт обычно лучшие результаты по сравнению с прямым жадным алгоритмом. Это связано с тем, что обратный жадный алгоритм учитывает признаки, информативные в совокупности, но неинформативные, если рассматривать их по отдельности.
* SSS (Sequential Stepwise Selection) – вунаправленное исключение/отбор - комбинация вышеперечисленных алгоритмов: тестирование на каждом шаге после включения/исключения признаков.

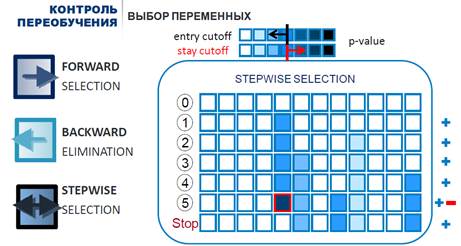


Рисунок 3 - Алгоритмы последовательного (пошагового) поиска

 На каждом шаге описанных алгоритмов для оценки качества обычно используются такие критерии, как F-тест (статистика Фишера), t-тест (статистика Стъюдента), скорректированный коэффициент детерминации R2 и прочие. Сам алгоритм при этом принимает форму последовательности F-тестов, t-тестов, скорр R2.

Популярным оберточным методом также является SVM-RFE (SVM-based Recursive Feature Elimination, рекурсивное исключение признаков), который иногда также относят к встроенным методам отбора. Этот метод использует как классификатор SVM и работает итеративно: начиная с полного множества признаков обучает классификатор, ранжирует признаки по весам, которые им присвоил классификатор, убирает какое-то число признаков и повторяет процесс с оставшегося подмножества признаков, если не было достигнуто их требуемое количество. Таким образом, этот метод очень похож на встроенный, потому что непосредственно использует знание того, как устроен классификатор.

3.Встроенные методы (embedded)

Встроенные алгоритмы требуют меньше вычислений, чем wrapper methods (хотя и больше, чем методы фильтрации).

Очень похожи на оберточные методы, но для выбора признаков используется непосредственно структура некоторого классификатора. В оберточных методах классификатор служит только для оценки работы на данном множестве признаков, тогда как встроенные методы используют какую-то информацию о признаках, которую классификаторы присваивают во время обучения.

Основным методом из этой категории является регуляризация.

Регуляризация — это своеобразный штраф за излишнюю сложность модели, который позволяет защитить себя от перетренировки в случае наличия среди признаков неинформативных. Не стоит думать, что регуляризация бывает только в линейных моделях, и для бустинга и для нейросетей существуют свои методы регуляризации.  
Существуют различные ее разновидности, но основной принцип общий. Если рассмотреть работу классификатора без регуляризации, то она состоит в построении такой модели, которая наилучшим образом настроилась бы на предсказание всех точек тренировочного сета.  
Например, если алгоритм классификации линейная регрессия, то подбираются коэффициенты полинома, который аппроксимирует зависимость между признаками и целевой переменной. В качестве оценки качества подобранных коэффициентов выступает среднеквадратичная ошибка (RMSE). Т.е. параметры подбираются так, чтобы суммарное отклонение (точнее суммарный квадрат отклонений) у точек предсказанных классификатором от реальных точек было минимальным.  
Идея регуляризации в том, чтобы построить алгоритм, минимизирующий не только ошибку, но и количество используемых переменных.

**Лассо-регрессия / Ридж-регрессия**(метод регуляризации Тихонова, ridge regression)

Эти методы часто применяются для борьбы с переизбыточностью данных, когда независимые переменные коррелируют друг с другом (т.е. имеет место мультиколлинеарность). Следствием этого является плохая обусловленность матрицы  и неустойчивость оценок коэффициентов регрессии. Оценки, например, могут иметь неправильный знак или значения, которые намного превосходят те, которые приемлемы из физических или практических соображений.

Обе техники позволяют уменьшить размерность данных и устранить/смягчить проблему переобучения. Для этого применяются два способа:

* L1-регуляризация — добавляет штраф к сумме абсолютных значений коэффициентов. Этот метод используется в **Лассо-регрессии**. В процессе работы алгоритма величина приписанных алгоритмом коэффициентов будет пропорциональна важности соответствующих переменных для классификации, а для переменных, которые дают наименьший вклад в устранение ошибки, коэффициенты станут нулевыми. Таким образом, более значимые признаки сохранят свои коэффициенты ненулевыми, а менее значимые – обнулятся. Стоит также отметить, что большие по модулю отрицательные значения коэффициентов тоже говорят о сильном влиянии.
* L2-регуляризация — добавляет штраф к сумме квадратов коэффициентов. Этот метод используется в ридж-регрессии.

В большинстве случаев исследователи и разработчики предпочитают L2-функцию — она эффективнее с точки зрения вычислительных функций. С другой стороны, лассо-регрессия позволяет уменьшить значения некоторых коэффициентов до 0, то есть вывести из поля исследования лишние переменные. Это полезно, если на какое-либо явление влияют тысячи факторов и рассматривать все их оказывается бессмысленно.

Ридж-регрессия - усовершенствование линейной регрессии с повышенной устойчивостью к ошибкам, налагающая ограничения на коэффициенты регрессии для получения более приближенного к реальности результата.

Применение гребневой регрессии нередко оправдывают тем, что это практический приём, с помощью которого при желании можно получить меньшее значение среднего квадрата ошибки.

Метод стоит использовать, если:

* сильная обусловленность;
* сильно различаются собственные значения или некоторые из них близки к нулю;
* в матрице  есть почти линейно зависимые столбцы.

В процессе работы алгоритма минимизируется сумма двух слагаемых:

- среднеквадратичная ошибка,

- регуляризирующий оператор (сумма квадратов всех коэффициентов, умноженная на альфу). В процессе работы алгоритма размеры коэффициентов будут пропорциональны важности соответствующих переменных, а перед теми переменными, которые дают наименьший вклад в устранение ошибки, станут околонулевые.  
Пара слов о параметре альфа. Он позволяет настраивать вклад регуляризирующего оператора в общую сумму. С его помощью мы можем указать приоритет — точность модели или минимальное количество используемых переменных.  
В LASSO – всё аналогично, за исключением регуляризирующего оператора. Он представляет собой не сумму квадратов, а сумму модулей коэффициентов. Несмотря на незначительность различия, свойства отличаются. Если в ridge по мере роста альфа все коэффициенты получают значения все ближе к нулевым, но обычно при этом все-таки не зануляются. То в LASSO с ростом альфа все больше коэффициентов становятся нулевыми и совсем перестают вносить вклад в модель. Таким образом, мы получаем действительно отбор признаков. Более значимые признаки сохранят свои коэффициенты ненулевыми, менее значимые — обнулятся.

Методы выделения признаков. Эти методы каким-то образом составляют из уже исходных признаков новые, все также полностью описывающие пространство набора данных, но уменьшая его размерность и теряя в репрезентативности данных, т.к. становится непонятно, за что отвечают новые признаки. Все методы feature extraction можно разделить на **линейные** и **нелинейные**.

Одним из самых известных методов **линейного** выделения признаков является PCA (Principal Component Analysis, рус. *метод главных компонент*). Основной идеей этого метода является поиск такой гиперплоскости, на которую при ортогональной проекции всех признаков максимизируется дисперсия. Данное преобразование может быть произведено с помощью сингулярного разложения матриц и создает проекцию только на линейные многомерные плоскости, поэтому и метод находится в категории линейных.