**Отбор признаков**

**Методы отбора признаков: с регрессией и без нее**

Отбор признаков позволяет улучшить производительность модели, уменьшить риски переобучения и сделать модели проще и легче интерпретируемыми.

Методы отбора признаков можно условно разделить на две основные категории:

**Итерационные методы с регрессией**: требуют построения и переоценки модели на различных комбинациях признаков. Примеры: RFE (Recursive Feature Elimination), SFS (Sequential Feature Selector), RFECV (Recursive Feature Elimination with Cross-Validation).

**Методы без итерационного расчета регрессии**: определяют важность признаков без необходимости многократного построения регрессионных моделей. Примеры: оценка важности в RandomForest, L1-регуляризация в Lasso, статистический тест ANOVA, Mutual Information.

У каждой из категорий есть свои преимущества. Из очевидного – для больших наборов данных итерационный подход может быть слишком ресурсоемким. Для малых наборов данных при итерации возрастает вероятность переобучения. Не все модели регрессии основаны на итерациях. Для моделей, которые оценивают важность признаков во время обучения (например, случайные леса), дополнительный итерационный отбор представляется избыточным.

**Подробнее о примененных в ноутбуке методах**

**RFE (Recursive Feature Elimination):**

Основан на идее рекурсивного исключения признаков, строя модель на каждом этапе и оценивая ее качество. После каждой итерации наименее важный признак удаляется, и процесс повторяется. Останавливается, когда достигнуто заданное количество признаков или когда улучшения в качестве модели после исключения признаков не наблюдается.

Пример: если в наборе данных один из признаков имеет наименьший вклад в модель (присваивается моделью после обучения), RFE исключает этот признак и запускает модель без него.

**RandomForest:**

Оценивает важность признака, исходя из его способности уменьшать неопределенность при разделении данных в решающих деревьях и на основе ухудшения точности модели при случайной перестановке значений данного признака.

Пример: если при удалении признака из модели, например, INR(MHO) точность прогноза значительно ухудшается, этот признак считается важным.

**Lasso:**

Метод добавления "штрафа" к линейной регрессии за "сложность" модели. Суть этого штрафа заключается в том, чтобы "подталкивать" коэффициенты модели к нулю. В результате этого некоторые коэффициенты могут стать равными нулю, что означает, что соответствующие им признаки не влияют на прогноз. Это делает Lasso полезным для отбора признаков, когда много переменных, и необходимо узнать, какие из них действительно важны.

Пример:

Есть датасет с информацией о домах: площадь дома, количество комнат, год постройки, расстояние до ближайшего метро и так далее. Необходимо предсказать стоимость дома на основе этих признаков.

При использовании линейной регрессии без регуляризации, каждый признак получил бы свой вес или коэффициент. Но если использовать Lasso-регрессию, некоторые из этих коэффициентов могут быть "сжаты" до нуля. Например, если расстояние до метро не так важно для стоимости дома, как его площадь, коэффициент для расстояния до метро может стать нулевым, исключая этот признак из модели.

**SFS (Sequential Feature Selector):**

Запускает регрессор (в ноутбуке LinearRegression()) со случайным набором признаков и итерационно добавляет признаки один за другим максимизируя качество модели. Этот процесс продолжается до достижения желаемого числа признаков или пока добавление новых признаков не перестанет улучшать качество модели.

**ANOVA (Дисперсионный анализ):**

ANOVA оценивает различия между средними значениями двух или более групп. В контексте отбора признаков это означает проверку, влияет ли разбиение по определенному признаку на различные группы на целевую переменную.

Пример: определить, есть ли статистически значимые различия в средних дозах лекарства, назначаемых мужчинам и женщинам. Если да, то гендер маркируется методом как значимая переменная.

*(для информации)Расчёт по этому методу в ноутбуке сделан, но результат в итоговый расчет ранга признаков не включен.*

**Mutual Information:**

Это мера, показывающая, насколько хорошо одна переменная может предсказать другую. Она измеряет уменьшение неопределенности одной переменной при знании другой. Рассчитывается вероятность того, что случайная переменная X примет значение xi​ и случайная переменная Y примет значение yi одновременно. В отличие от корреляции, где измеряется линейная зависимость между переменными, Mutual Information измеряет любую форму зависимости - линейную, квадратичную или любую другую.

Пример: если, зная расположение дома (X), можно достаточно точно предсказать его стоимость (Y), то зависимость между этими переменными сильная

**Реализация в библиотеках Python**

В ноутбуке использованы реализации описанных методов в библиотеке sklearn.

**Нормализация и ранжирование признаков**

После расчета важности признаков различными методами необходимо нормализовать полученные результаты (после методов выдающих результат в виде весовых коэффициентов: ‘RandomForest', 'Lasso', 'Mutual Info'), для приведения к единому масштабу. Для этого использован метод MinMaxScaler из sklearn. Полученные по всем методам и нормализованные значения для каждого признака суммированы и ранжированы с помощью метода sort\_values() библиотеки pandas.

.