**1. Что означает уменьшение размерности в машинном обучении?**

Уменьшение размерности (или сокращение размерности) — это процесс уменьшения количества входных переменных в наборе данных. Это делается для того, чтобы:

* Снизить сложность модели.
* Уменьшить объем вычислений.
* Избежать проблемы "проклятия размерности", когда данные становятся слишком разбросанными в пространстве высокой размерности.
* Улучшить визуализацию данных.
* Повысить производительность модели за счет удаления шума и неинформативных признаков.

**2. Какие методы входят в Feature Selection? Как они работают?**

**Фильтрация по низкой дисперсии и по высокой корреляции**

Методы отбора признаков (Feature Selection) включают:

* **Filter Methods (Фильтрационные методы):**
  + **Univariate Selection (Одномерный отбор):** Оценивают каждый признак по отдельности, используя статистические тесты, и отбирают признаки с наивысшими оценками.
  + **Mutual Information (Взаимная информация):** Измеряет зависимость между признаками и целевой переменной.
* **Wrapper Methods (Методы-обертки):**
  + **Forward Selection (Пошаговый отбор):** Начинают с пустого набора признаков, добавляя по одному признаку, который наиболее улучшает модель.
  + **Backward Elimination (Пошаговое исключение):** Начинают с полного набора признаков, удаляя по одному признаку, который наименее важен для модели.
  + **Recursive Feature Elimination (Рекурсивное исключение признаков):** Постепенно удаляют наименее важные признаки, используя модель (обычно линейную или деревья решений).
* **Embedded Methods (Встроенные методы):**
  + **Regularization Methods (Регуляризационные методы):** Такие как LASSO (L1-регуляризация), которая уменьшает коэффициенты неинформативных признаков до нуля.
  + **Tree-based Methods (Методы на основе деревьев):** Такие как Random Forest и Gradient Boosting, которые оценивают важность признаков на основе улучшения критерия разбиения (например, прирост информации).

**Фильтрационные методы** оценивают важность признаков, независимо от модели, используя различные статистические критерии.

**Обёрточные методы** используют модель для оценки набора признаков и их влияния на производительность модели.

**Встроенные методы** выполняют отбор признаков непосредственно в процессе обучения модели. Они интегрируют процесс отбора признаков и обучение модели в одну процедуру.

**3. Расскажите принцип работы метода PCA.**

Метод главных компонент (PCA, Principal Component Analysis) работает следующим образом:

1. **Центрирование данных:** Сначала все данные центрируются, т.е. из каждого признака вычитается его среднее значение.
2. **Ковариационная матрица:** Вычисляется ковариационная матрица признаков.
3. **Собственные вектора и значения:** Находятся собственные значения и соответствующие собственные вектора ковариационной матрицы.
4. **Сортировка:** Собственные значения сортируются по убыванию, и выбираются соответствующие им собственные вектора.
5. **Проекция:** Данные проецируются на выбранные собственные вектора (главные компоненты), создавая новый набор данных с уменьшенной размерностью.

**4. Что означает понятие главная компонента?**

Главная компонента — это новая переменная(набор данных), полученная в результате проекции исходных данных на оси, которые максимизируют дисперсию данных. Первая главная компонента объясняет наибольшую часть дисперсии в данных, вторая — вторую по величине и т.д. Главные компоненты ортогональны друг другу и независимы.

**5. Что означает термин Ансамбли в контексте машинного обучения?**

Ансамбли (ensembles) — это методы, которые объединяют несколько моделей для улучшения общей производительности по сравнению с использованием одной модели. Идея состоит в том, чтобы комбинировать слабые модели (которые могут быть плохими предсказателями по отдельности) в сильную модель, уменьшая вероятность ошибки и повышая устойчивость.

**6. Как работают алгоритмы стекинг, бэггинг, бустинг?**

* **Бэггинг (Bagging, Bootstrap Aggregating):**
  + Сначала создаются несколько выборок данных с повторением (bootstrap).
  + На каждой выборке обучается модель.
  + Предсказания всех моделей объединяются (например, путем усреднения или голосования).
* **Бустинг (Boosting):**
  + Модели обучаются последовательно.
  + Каждая следующая модель пытается исправить ошибки предыдущей, уделяя больше внимания ошибкам.
  + Финальное предсказание — взвешенная сумма предсказаний всех моделей.
* **Стекинг (Stacking):**
  + Несколько моделей (базовые обучающие модели) обучаются на одних и тех же данных.
  + Предсказания этих моделей используются как новые признаки для модели-мета-алгоритма (метамодель).
  + Мета-модель обучается на этих предсказаниях, чтобы сделать финальное предсказание.

**7. Объясните метод Random Forest. Какой алгоритм ансамблирования в нем используется?**

Random Forest — это ансамблевый метод, основанный на бэггинге. Он работает следующим образом:

1. **Создание выборок:** Генерируются несколько подвыборок данных с повторением (bootstrap).
2. **Обучение деревьев:** На каждой подвыборке обучается дерево решений.
3. **Случайный отбор признаков:** При каждом разбиении узла дерева рассматривается случайное подмножество признаков, что увеличивает разнообразие деревьев и снижает корреляцию между ними.
4. **Объединение предсказаний:** Предсказания всех деревьев объединяются (например, путем голосования для классификации или усреднения для регрессии).

Random Forest улучшает устойчивость и точность модели, уменьшая переобучение по сравнению с отдельными деревьями решений.

Алгоритм `RandomForestClassifier` является одной из реализаций метода случайного леса для задач классификации. Суть алгоритма заключается в объединении множества деревьев решений для улучшения общей производительности модели за счет снижения дисперсии и уменьшения переобучения.

### Основные шаги работы алгоритма RandomForestClassifier:

1. \*\*Бутстреппинг (Bootstrap Aggregation или Bagging):\*\*

- Для создания каждого дерева используется метод бутстреппинга, который заключается в случайной выборке с возвращением из исходного набора данных. Это означает, что некоторые образцы могут быть выбраны несколько раз, а другие — не выбраны вовсе.

- За счет этого каждый набор данных для обучения дерева будет несколько отличаться от исходного, что обеспечивает разнообразие обучающих данных для каждого дерева.

2. \*\*Создание и обучение деревьев решений:\*\*

- Каждое дерево решений строится, используя подмножество признаков, выбранное случайным образом на каждом разбиении узла. Это добавляет дополнительное разнообразие и уменьшает корреляцию между деревьями.

- Для каждого узла дерева случайным образом выбирается подмножество признаков, и из них выбирается признак, который лучше всего разделяет данные (например, по критерию энтропии или индекса Джини).

3. \*\*Комбинирование предсказаний:\*\*

- После того как все деревья обучены, случайный лес использует метод голосования для получения итогового предсказания.

- В задаче классификации каждое дерево выдает свой класс, и итоговый класс определяется большинством голосов (т.е., класс, который получил наибольшее количество голосов от всех деревьев).

### Преимущества и недостатки алгоритма RandomForestClassifier:

#### Преимущества:

1. \*\*Устойчивость к переобучению:\*\* За счет бутстреппинга и случайного выбора признаков деревья менее коррелированы между собой, что уменьшает переобучение.

2. \*\*Высокая точность:\*\* Случайные леса обычно дают высокую точность благодаря комбинированию предсказаний множества деревьев.

3. \*\*Обработка пропущенных значений:\*\* Random Forest может обрабатывать пропущенные значения, не удаляя строки или столбцы.

4. \*\*Масштабируемость:\*\* Алгоритм можно эффективно распараллеливать, что делает его масштабируемым для работы с большими данными.

5. \*\*Оценка важности признаков:\*\* Алгоритм предоставляет важность каждого признака, что полезно для понимания значимости признаков в модели.

#### Недостатки:

1. \*\*Высокие вычислительные затраты:\*\* Обучение большого числа деревьев может быть вычислительно затратным и требовать значительных ресурсов.

2. \*\*Интерпретируемость:\*\* В отличие от одиночного дерева решений, случайный лес является менее интерпретируемым, так как результат получается за счет комбинирования многих деревьев.

### Пример кода с использованием `RandomForestClassifier`:

```python

from sklearn.datasets import make\_classification

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Создание синтетического датасета

X, y = make\_classification(n\_samples=1000, n\_features=50, n\_informative=40, n\_classes=2, random\_state=42)

# Разделение данных на обучающий и тестовый наборы

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Создание и обучение модели случайного леса

rf = RandomForestClassifier(random\_state=42)

rf.fit(X\_train, y\_train)

# Предсказание меток на тестовых данных

y\_pred = rf.predict(X\_test)

# Оценка точности модели

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.4f}")

# Вывод важности признаков

importances = rf.feature\_importances\_

print("Feature importances:", importances)

```

### Параметры `RandomForestClassifier`:

- `n\_estimators`: Количество деревьев в лесу (по умолчанию 100).

- `max\_features`: Максимальное число признаков, используемых для разбиения узла. Это может быть целое число, доля от общего числа признаков или одна из строковых опций (`'auto'`, `'sqrt'`, `'log2'`).

- `max\_depth`: Максимальная глубина деревьев (по умолчанию None, что означает, что узлы расширяются до тех пор, пока все листья не будут чистыми или пока все листья не будут содержать менее `min\_samples\_split` образцов).

- `min\_samples\_split`: Минимальное число образцов, необходимых для разбиения узла (по умолчанию 2).

- `random\_state`: Случайное состояние для воспроизводимости результатов.

### Вывод:

Алгоритм `RandomForestClassifier` является мощным инструментом для классификации, который сочетает в себе простоту деревьев решений и устойчивость ансамблевых методов, обеспечивая высокую точность и гибкость в работе с различными типами данных.