**Введение в sklearn**

Что такое sklearn?

Scikit-learn (sklearn) — это библиотека машинного обучения на Python, предоставляющая удобные инструменты для работы с моделями классического машинного обучения. Она включает в себя различные алгоритмы, методы для предобработки данных, подбора гиперпараметров и оценки моделей. Sklearn широко используется благодаря своей простоте, скорости и интеграции с другими библиотеками Python, такими как NumPy и pandas.

Основные возможности библиотеки.

Основные возможности библиотеки sklearn включают реализацию различных алгоритмов машинного обучения, таких как классификация, регрессия, кластеризация и снижение размерности. Она также предоставляет инструменты для предобработки данных (масштабирование, заполнение пропусков), подбора гиперпараметров (GridSearchCV, RandomizedSearchCV) и оценки моделей с помощью метрик. Sklearn поддерживает создание pipeline для автоматизации полного цикла обработки данных и обучения модели.

**Алгоритмы машинного обучения в sklearn**

Разделение алгоритмов на группы:

Алгоритмы в sklearn можно разделить на несколько основных групп. **Классификация** включает методы для предсказания категориальных меток, такие как Logistic Regression и Random Forest. **Регрессия** применяется для предсказания числовых значений, примеры — Linear Regression и Ridge. **Кластеризация** используется для группировки данных, например, алгоритмы K-Means и DBSCAN. **Снижение размерности**, как PCA и t-SNE, помогает уменьшить число признаков, сохраняя важную информацию для анализа и визуализации.

**Алгоритмы классификации**

**Logistic Regression**

**Logistic Regression** — это метод классификации, который использует логистическую функцию для предсказания вероятности принадлежности объекта к классу. Он работает путем вычисления линейной комбинации входных признаков и применения сигмоидной функции для преобразования результата в значение между 0 и 1, что интерпретируется как вероятность.

**Преимущества**:

* **Простота**:
* **Интерпретируемость**

**Недостатки**:

* **Чувствительность к выбросам**
* **Не работает с нелинейными зависимостями**

**k-Nearest Neighbors (k-NN)**

**k-Nearest Neighbors (k-NN)** — это алгоритм классификации, который основывается на близости объектов в пространстве признаков. При классификации нового объекта алгоритм ищет k ближайших соседей из обучающего набора, и класс объекта определяется на основе большинства классов его соседей.

**Преимущества**:

* **Простота**
* **Нет необходимости в обучении**

**Недостатки**:

* **Медленный при большом количестве данных**
* **Чувствителен к шуму**

**Support Vector Machines (SVM)**

**Support Vector Machines (SVM)** — это алгоритм классификации, который находит оптимальную гиперплоскость для разделения классов в пространстве признаков. Он работает, максимизируя расстояние (или зазор) между ближайшими точками (опорными векторами) разных классов. В случае нелинейной разделимости SVM может использовать ядровые функции для преобразования данных в более высокоразмерное пространство, где классы могут быть линейно разделимы.

**Преимущества**:

* **Хорошо работает на высокоразмерных данных**

**Недостатки**:

* **Требует настройки гиперпараметров**
* **Медленное обучение на больших данных**

**Random Forest**

**Random Forest** — это ансамблевый метод классификации и регрессии, который строит множество решающих деревьев и объединяет их предсказания для повышения точности и стабильности модели. При обучении каждое дерево создается на случайной подвыборке данных и использует случайный поднабор признаков для принятия решений, что уменьшает корреляцию между деревьями.

**Преимущества**:

* **Устойчива к выбросам**
* **Может обрабатывать большие объемы данных**

**Недостатки**:

* **Требует больше ресурсов для обучения**
* **Может склоняться к переобучению**

**Gradient Boosting (XGBoost, LightGBM)**

**Gradient Boosting** — это ансамблевый метод, который строит модель, обучая последовательные слабые модели (обычно решающие деревья) и корректируя ошибки предыдущих моделей. XGBoost и LightGBM являются популярными реализациями градиентного бустинга, которые оптимизируют производительность и скорость.

**Преимущества**:

* **Высокое качество предсказаний**

**Недостатки**:

* **Сложная настройка**
* **Требует много ресурсов**

**Алгоритмы регрессии**

**Linear Regression**

**Linear Regression** — это метод регрессии, который моделирует зависимость между одной или несколькими независимыми переменными и зависимой переменной, предполагая, что эта зависимость линейна. Алгоритм находит коэффициенты (параметры) модели, минимизируя сумму квадратов ошибок между предсказанными и фактическими значениями.

**Преимущества**:

* **Интерпретируемость**
* **Быстрая и простая**

**Недостатки**:

* **Плохо работает на сложных данных**’

**Ridge и Lasso Regression**

**Ridge и Lasso Regression** — это методы линейной регрессии с регуляризацией, которые помогают улучшить обобщающую способность модели и уменьшить переобучение.

**Ridge Regression** добавляет к функции потерь штраф за квадраты коэффициентов (L2 регуляризация), что помогает контролировать сложность модели и уменьшить влияние выбросов.

**Lasso Regression** использует L1 регуляризацию, которая добавляет штраф за абсолютные значения коэффициентов. Это может привести к обнулению некоторых коэффициентов, что способствует выбору наиболее значимых признаков и упрощает модель.

**Преимущества**:

* **Снижение переобучения**

**Недостатки**:

* **Требует подбора гиперпараметров**

**Decision Tree Regressor**

**Decision Tree Regressor** — это алгоритм регрессии, который использует дерево решений для предсказания числовых значений. Дерево разбивает пространство признаков на подпространства, основываясь на значениях входных признаков, и делает предсказания на основе среднего значения целевой переменной в каждом листе дерева.

**Преимущества**:

* **Простота**
* **Не требует нормализации данных**

**Недостатки**:

* **Склонен к переобучению без настройки**

**Алгоритмы кластеризации**

**K-Means**

**K-*Means*** — это алгоритм кластеризации, который разделяет данные на K кластеров, минимизируя внутрикластерное расстояние. Алгоритм работает в несколько шагов: случайным образом инициализирует центры кластеров, назначает каждый объект к ближайшему центру, а затем обновляет центры кластеров на основе средних значений объектов, принадлежащих каждому кластеру. Этот процесс повторяется до тех пор, пока центры кластеров не стабилизируются.

**Преимущества**:

* **Простота**
  + **Быстрое обучение**

**Недостатки**:

* **Чувствителен к выбору количества кластеров**
* **Плохо работает на данных с разной плотностью**

**DBSCAN**

**DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** — это алгоритм кластеризации, который группирует данные на основе их плотности. Он определяет кластеры как области с высокой плотностью точек, отделенные от областей с низкой плотностью. Алгоритм работает, определяя ядровые точки (объекты с достаточным количеством соседей в заданном радиусе) и расширяя кластеры, добавляя соседние точки, пока не будут охвачены все доступные точки.

**Преимущества**:

* **Автоматически определяет количество кластеров**

**Недостатки**:

* **Чувствителен к настройке параметров**

**Алгоритмы снижения размерности**

**Principal Component Analysis (PCA)**

**Principal Component Analysis (PCA)** — это метод снижения размерности, который преобразует данные, проецируя их на новое пространство, образованное главными компонентами. Эти компоненты представляют собой линейные комбинации исходных признаков и упорядочены по убыванию объясненной дисперсии, что позволяет выявить основные направления вариации в данных.

**Преимущества**:

* **Уменьшение размерности**
* **Визуализация**

**Недостатки**:

* **Теряет интерпретируемость исходных признаков**

**t-SNE**

**t-SNE (t-distributed Stochastic Neighbor Embedding)** — это метод снижения размерности, который особенно эффективен для визуализации многомерных данных. Он работает путем сохранения относительных расстояний между точками в высокоразмерном пространстве и проекции их на двумерное или трехмерное пространство.

**Преимущества:**

* Уменьшает размерность данных, что облегчает их визуализацию.
* Эффективно группирует схожие точки, позволяя выделять кластеры.

**Недостатки:**

* Потеря интерпретируемости исходных признаков, так как полученные размеры не имеют явного значения.
* Чувствительность к выбору гиперпараметров, таких как количество соседей и перцепция расстояний.

**Инструменты sklearn для работы с данными**

**Data Preprocessing**

### StandardScaler

**StandardScaler** — это метод масштабирования, который стандартизирует признаки, приводя их к нулевому среднему значению и единичной дисперсии. Это достигается путем вычитания среднего и деления на стандартное отклонение. Стандартизация особенно полезна для алгоритмов, чувствительных к масштабам, таких как SVM и K-Means.

### MinMaxScaler

**MinMaxScaler** — это метод масштабирования, который преобразует значения признаков в заданный диапазон, обычно от 0 до 1. Для этого вычитается минимальное значение и делится на разницу между максимальным и минимальным значениями. MinMaxScaler полезен для алгоритмов, которые предполагают равномерное распределение данных, таких как нейронные сети, и помогает избежать проблем, связанных с разными масштабами признаков.

**Model Selection**

### GridSearchCV

**GridSearchCV** — это метод, который выполняет полный перебор всех комбинаций заданных гиперпараметров для модели. Он создает решетку (grid) возможных значений и последовательно обучает модель на каждой комбинации, оценивая ее производительность с помощью кросс-валидации.

**Преимущества**:

* Полный перебор всех возможных комбинаций гиперпараметров, что гарантирует нахождение наилучшей комбинации в пределах заданного диапазона.

**Недостатки**:

* Высокая вычислительная стоимость, особенно при большом количестве гиперпараметров и значений, что может занять много времени.

### RandomizedSearchCV

**RandomizedSearchCV** — это метод, который случайным образом выбирает комбинации гиперпараметров из заданного пространства значений. Вместо полного перебора, он выбирает фиксированное количество комбинаций для оценки.

**Преимущества**:

* Более быстрое выполнение по сравнению с GridSearchCV, так как он не исследует все возможные комбинации, что особенно полезно для больших наборов гиперпараметров.
* Может обнаружить хорошие комбинации гиперпараметров, не тратя много времени.

**Metrics**

**Метрики** — это показатели, используемые для оценки производительности моделей машинного обучения. К числу наиболее распространенных метрик относятся:

* **Accuracy** (точность)
* **Precision** (точность)
* **Recall** (полнота)
* **ROC-AUC**

**Заключение**

### Ключевые преимущества scikit-learn

1. **Широкий выбор алгоритмов**:
2. **Простота использования**
3. **Совместимость с другими библиотеками**
4. **Поддержка обработки данных**
5. **Кросс-валидация и метрики**
6. **Гибкость**

### Как выбрать правильный алгоритм для задачи?

1. **Определите тип задачи**:
2. **Анализ данных**: Изучите набор данных, чтобы понять его размер, качество, количество признаков и распределение классов.
3. **Сложность модели**
4. **Проверка производительности**: Используйте кросс-валидацию, чтобы протестировать несколько алгоритмов и сравнить их производительность на ваших данных.
5. **Требования к интерпретируемости**: Если важно, чтобы модель была понятна и интерпретируема, рассмотрите более простые алгоритмы, такие как Decision Trees или Logistic Regression.
6. **Ресурсы и время**: Учтите доступные вычислительные ресурсы и время на обучение. Сложные модели могут требовать больше времени на обучение и больше ресурсов.