Лабораторная работа №7
Классификация (Байесовские методы, деревья)

Цель:

Ознакомиться с методами кластеризации модуля Sklearn

- Контрольные вопросы

Выполнение:

1. Загрузка данных:

1.1. Загрузить датасет по ссылке: https://archive.ics.uci.edu/dataset/109/wine.

- Данные представлены в виде data файла. Данные представляют собой информацию о **трех** классах вина.
- Необходимо прочитать описание к данным (Variables Table) и определить, какие столбцы относятся к данным, а какие к меткам данных. Метки данных не используются для анализа данных и на последующих шагах исключаются из набора

> **Дать пояснение**
> 1. Перечислить столбец (или столбцы, если их несколько), которые отнесены к меткам

1.2. Создать Python скрипт. Загрузить данные в датафрейм

1.3. Выделить в отдельные переменные данные (например, в `X`) и их метки (например, в `labels`)

1.4. Преобразовать метки в `число` с использованием LabelEncoder и LabelEncoder.fit_transform

- См. пример Label encoding

1.5. Разделить (50%/50%) данные на на обучающую (`x_train`,`y_train`) и тестовую выборки (`x_test`,`y_test`)

- Используется train_test_split
- См. примеры кода в 3.1. Cross-validation: evaluating estimator performance

2. Байесовские методы

2.1. Провести классификацию наивным байесовским методом с параметрами по умолчанию

- Обучение (`fit`) происходит на обучающей выборке, классификация (`predict`) производится для тестовой выборки

- Пример и теория Наивный байесовский классификатор, Naive Bayes
- Wiki Наивный байесовский классификатор
- Наивный байесовский классификатор в Python
- Вывести на экран общее количество точек и количество неправильно маркированных точек
- Вычислить точность классификации (функция GaussianNB.score)

> **Дать пояснения**
> 1. Перечислить атрибуты функции (с пояснением, на что каждый атрибут влияет)
> 2. Привести пример использования атрибутов (одного или нескольких)

2.2. Постройте два графика - зависимость неправильно классифицированных наблюдений и зависимость точности классификации от размера тестовой выборки.

- Размер тестовой выборки (`test_size`) изменяйте от 0.05 до 0.95 с шагом 0.05.
- Параметр `random_state` сделайте равным номеру своей зачетной книжки.

$$Точность = 1 - \frac{Количество\_неправильных\_точек}{Общее\_количество\_точек}$$

> **Дать пояснения**
> 1. Что происходит с результатами с изменением размера тестовой выборки

2.3. Повторите пункты **2.1.** и **2.2.** для методов классификации MultinomialNB, ComplementNB, BernoulliNB

3. Классифицирующие деревья

3.1. Проведите классификацию тех же данных с помощью деревьев

- Использовать функцию DecisionTreeClassifier
- Пример Decision Trees, Деревья решений (Decision Trees)
- Wiki Дерево решений,
- Дерево решений (CART). От теоретических основ до продвинутых техник и реализации с нуля на Python / Хабр
- Вычислить точность классификации (функция DecisionTreeClassifier.score
- Выведите характеристики дерева - количество листьев (DecisionTreeClassifier.get_n_leaves) и глубину (DecisionTreeClassifier.get_depth)
- Выведите изображение полученного дерева (используется `matplotlib.pyplot as plt` и plot_tree (пример, код в Decision Trees))

> **Дать пояснения**
> 1. Что означают данные, отображаемые в каждом блоке (на примере двух разных блоков)

3.2. Постройте два графика - зависимость неправильно классифицированных наблюдений и зависимость точности классификации от размера тестовой выборки.

- Размер тестовой выборки (`test_size`) изменяйте от 0.05 до 0.95 с шагом 0.05.
- Параметр `random_state` сделайте равным номеру своей зачетной книжки.

> **Дать пояснения**
>
> 1. Что происходит с результатами с изменением размера тестовой выборки

3.3. Повторите пункт **3.2.** для различных параметров criterion (entropy, log_loss), splitter(random), max_depth(от 2 до 5), min_samples_split(от 10 до 80 с шагом 10), min_samples_leaf(от 10 до 80 с шагом 10)

> **Дать пояснения**
>
> 1. При каких значениях параметров классификация дает наилучшие результата и почему?

Контрольные вопросы

Контрольные вопросы по теме GaussianNB (Gaussian Naive Bayes):

I. Основы:

1. Что такое наивный байесовский классификатор? Опишите основные принципы его работы.
2. Что означает "наивный" в наивном байесовском классификаторе? Какие предположения делает этот алгоритм?
3. В чем заключается отличие GaussianNB от других вариантов наивного байесовского классификатора, таких как MultinomialNB и BernoulliNB? Когда следует использовать GaussianNB?
4. Какие типы данных подходят для GaussianNB?

II. Математическая основа:

1. Какая формула используется для расчета вероятности принадлежности объекта к определенному классу в GaussianNB?
2. Как GaussianNB оценивает параметры распределения (среднее значение и стандартное отклонение) для каждого признака в каждом классе?
3. Как обрабатываются нулевые значения частот в GaussianNB? Почему это важно?
4. Что такое теорема Байеса и как она применяется в GaussianNB?

III. Применение и практика:

1. Как использовать GaussianNB в scikit-learn (Python)? Приведите пример кода.
2. Какие гиперпараметры есть у GaussianNB и как они влияют на его работу?
3. Как оценить качество модели GaussianNB? Какие метрики можно использовать?
4. Как выбрать оптимальные значения гиперпараметров для GaussianNB? Какие методы можно использовать (например, GridSearchCV, RandomizedSearchCV)?
5. Как GaussianNB обрабатывает пропущенные значения?
6. Каковы преимущества и недостатки GaussianNB по сравнению с другими алгоритмами классификации?

IV. Более сложные вопросы:

1. Как влияет корреляция между признаками на производительность GaussianNB? Почему предположение о независимости признаков может быть проблематичным?
2. Как можно улучшить производительность GaussianNB, если предположение о независимости признаков не выполняется?
3. Как GaussianNB работает с категориальными признаками? Нужно ли их преобразовывать?
4. Как масштабирование признаков влияет на работу GaussianNB?

Контрольные вопросы по MultinomialNB, ComplementNB, BernoulliNB

Общие вопросы для всех трех классификаторов:

1. Что общего у MultinomialNB, ComplementNB и BernoulliNB? Чем они отличаются от GaussianNB?
2. Какое предположение о независимости признаков делают все наивные байесовские классификаторы? Почему это предположение называется "наивным"?
3. Как теорема Байеса применяется в каждом из этих классификаторов?
4. Как каждый из этих классификаторов обрабатывает неизвестные значения признаков во время обучения и предсказания?
5. Какие метрики можно использовать для оценки качества моделей, построенных с помощью этих классификаторов?
6. Как можно улучшить производительность этих классификаторов, если предположение о независимости признаков не выполняется?
7. Как каждый из этих классификаторов работает с непрерывными признаками? Нужно ли их преобразовывать?

MultinomialNB:

1. Для каких типов данных подходит MultinomialNB? Приведите примеры.
2. Как MultinomialNB рассчитывает вероятность принадлежности объекта к определенному классу? Какую роль играют частоты признаков?
3. Что такое сглаживание Лапласа (add-one smoothing) и зачем оно используется в MultinomialNB?
4. Как влияет параметр `alpha` (сглаживание) на производительность MultinomialNB?
5. В каких задачах MultinomialNB обычно показывает хорошие результаты?

ComplementNB:

1. В чем основное отличие ComplementNB от MultinomialNB? Для каких типов данных он разработан?
2. Как ComplementNB рассчитывает вероятность принадлежности объекта к определенному классу? Чем отличается его формула от MultinomialNB?
3. В каких случаях ComplementNB может быть предпочтительнее MultinomialNB?
4. Как параметр `alpha` (сглаживание) влияет на производительность ComplementNB?

BernoulliNB:

1. Для каких типов данных подходит BernoulliNB? Приведите примеры.
2. Как BernoulliNB обрабатывает признаки? Чем он отличается от MultinomialNB и ComplementNB в этом отношении?
3. Как BernoulliNB рассчитывает вероятность принадлежности объекта к определенному классу?
4. Как параметр `binarize` используется в BernoulliNB? Что происходит, если его не задать?
5. В каких задачах BernoulliNB обычно показывает хорошие результаты?

Сравнение классификаторов:

1. Сравните MultinomialNB, ComplementNB и BernoulliNB. В каких случаях следует использовать каждый из них?
2. Какие преимущества и недостатки есть у каждого из этих классификаторов?
3. Как выбрать наиболее подходящий наивный байесовский классификатор для конкретной задачи?

Контрольные вопросы по DecisionTreeClassifier

Вопросы охватывают основные аспекты `DecisionTreeClassifier` из библиотеки scikit-learn.

I. Основы и принципы работы:

1. Что такое дерево решений? Опишите основные принципы построения дерева решений для классификации.
2. Как DecisionTreeClassifier выбирает лучший признак для разделения данных в каждом узле? Объясните понятия impurity (нечистота), information gain (прирост информации) и gini impurity (нечистота Джини).
3. Как дерево решений обрабатывает непрерывные и категориальные признаки?
4. Что такое переобучение (overfitting) в контексте деревьев решений? Как оно проявляется и к чему может привести?
5. Что такое pruning (обрезка) дерева и зачем она нужна? Какие стратегии обрезки существуют?

II. Гиперпараметры и их влияние:

1. Перечислите основные гиперпараметры DecisionTreeClassifier и объясните, как каждый из них влияет на построение дерева и его производительность. Уделите внимание `criterion`, `max_depth`, `min_samples_split`, `min_samples_leaf`, `max_features`.
2. Как `max_depth` влияет на сложность модели и риск переобучения?
3. Как `min_samples_split` и `min_samples_leaf` помогают предотвратить переобучение?
4. Как выбор `criterion` (gini или entropy) влияет на построение дерева?
5. Как `max_features` может повлиять на скорость обучения и производительность модели?

III. Применение и практика:

1. Как использовать DecisionTreeClassifier в scikit-learn? Приведите пример кода для обучения и предсказания.
2. Как оценить качество модели DecisionTreeClassifier? Какие метрики можно использовать?
3. Как визуализировать обученное дерево решений?
4. Как можно использовать деревья решений для решения задач многоклассовой классификации?
5. Как обрабатывать пропущенные значения при использовании DecisionTreeClassifier?

IV. Преимущества и недостатки:

1. Каковы преимущества использования деревьев решений для классификации?
2. Каковы недостатки деревьев решений? В каких случаях они могут быть неэффективны?

V. Более сложные вопросы:

1. Как неустойчивость деревьев решений к изменениям в данных может повлиять на их производительность?
2. Как можно объединить несколько деревьев решений для повышения стабильности и точности предсказаний (например, Random Forest, Gradient Boosting)?
3. Как интерпретировать важность признаков, рассчитанную DecisionTreeClassifier?
4. Как можно использовать деревья решений для извлечения знаний из данных?