1. Классификация данных

1.1 Подготовка к эксперименту

Время выполнения эксперимента: 120 минут

Цель

После завершения данного эксперимента вы сможете:

• • Понимание методов оценки качества алгоритмов классификации

• • Освоение использования алгоритмов классификации из набора инструментов scilearn-kit

• • Оптимизация параметров модели с использованием таких методов, как GridSearchCV.

Предварительная среда

Для проведения эксперимента необходимо подготовить среду работы с Python версии 3.6 и выше. Экспериментальная среда должна включать следующие элементы:

• • Anaconda 。

1.2 Цель эксперимента

Классификация — это важная категория задач в обработке данных. Цель задачи классификации — определить, к какому из известных типов относится новый образец, исходя из известных его признаков. В зависимости от количества признаков задачи классификации делятся на бинарную и многоклассовую. Например, определение, является ли письмо спамом, — это задача с двумя классами, а определение числа — задача с несколькими классами.

Среди наиболее распространённых алгоритмов классификации в области анализа данных и машинного обучения можно выделить следующие:

Линейный классификатор

Линейный дискриминантный анализ (Linear Discriminant Analysis, LDA)

Логистическая регрессия

Классификатор на основе метода наивного байеса

Перцептор

Машина поддерживаемых векторов (Support Vector Machine, SVM)

Машини поддержки на основе метода наименьших квадратов

Квадратный классификатор

Оценка на основе ядра (kernel estimation)

Метод k-ближайших соседей

Алгоритм Boosting

Метод усиления градиентов (Gradient Boosting)

Адаптивное усиление (Adaboost)

Дерево решений

Случайные леса (Random Forests)

Нейронные сети

Среди методов линейной классификации наиболее основополагающими и типичными являются дискриминантный анализ и логистическая регрессия. Дискриминантный анализ — простой и наглядный метод, который классифицирует наблюдения по различию их расстояния до центров соответствующих категорий. Для этого на основе выборки строят дискриминантную функцию, и каждый наблюдение отнесено к категории, центр которой находится на наименьшем расстоянии от него. Логистическая регрессия начинается с построения регрессионной модели, затем параметры модели оцениваются методом максимального правдоподобия, после чего получается точное значение регрессии. На основе этого математически принимается решение с учётом различных вероятностей, что и позволяет решить задачу классификации.

Алгоритмы классификации находят широкое применение во многих областях. Качество таких алгоритмов обычно оценивается по следующим критериям: точности прогнозирования, сложности вычислений и простоте модели.

Цель данного эксперимента — на примере определения диагноза рака молочной железы показать, как следует использовать распространенные алгоритмы классификации в области анализа больших данных, а также как оценивать их эффективность и оптимизировать их работу.

1.3 Упражнение 1: Критерии классификации данных

Для оценки точности прогнозов, полученных с помощью двух методов, обычно используются три показателя: матрица смешивания, кривая ROC и площадь под кривой

(AUC).

(1) Матрица смешивания

Матрица смешивания — это показатель, позволяющий оценить точность классификации модели и определить её эффективность. Это самый простой, наглядный и удобный способ измерить точность классификатора, а все остальные показатели производительности можно вычислить на основе матрицы смешивания. Например, в задаче с двумя классами рассмотрим набор из 10 рукописных цифр, состоящих только из 0 и 1. Если у нас есть истинные и предсказанные метки, можно подсчитать количество каждого класса.Согласно сочетанию истинных и предсказанных значений данных в выборке, можно выделить четыре категории: истинный положительный (True positive, сокращённо TP), ложный положительный (False positive, сокращённо FP), истинный отрицательный (True negative, сокращённо TN) и ложный отрицательный (False negative, сокращённо FN). В аббревиатурах TP, FP, TN и FN первая буква обозначает, совпадает ли предсказанная категория с истинной, а вторая — категорию, в которой находится образец.

Настоящее значение	Прогнозируемое значение	
	1	0
1	5（TP）	1（TN）
0	2(FP)	2(FN)

В матрице смешивания измеряется количество, и при большом объёме данных или многочисленных категориях сложно оценить производительность различных моделей по количеству. Поэтому к матрице смешивания добавляются четыре показателя: точность (Accuracy), предсказательность (Precision), чувствительность (Sensitivity) и специфичность (Specificity). Предсказательность также называют точностью, а чувствительность — чувствительностью или восприятием (Recall).

Показатель	**Формула**	**Значение**
Точность	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$	Доля всех правильных прогнозов среди общего числа прогнозов в классификационной модели
Точность Точность выявления	$Precision = \frac{TP}{TP+FP}$	Доля правильных прогнозов среди всех результатов, полученных в положительном («позитивном») направлении.
Чувствительность	$Sensitivity = Recall = \frac{TP}{TP+FN}$	Доля правильных прогнозов среди

Коэффициент возврата		всех результатов, для которых истинное значение было положительным
специфичность	$Specificity = \frac{TN}{TN+FP}$	Доля правильных прогнозов среди всех результатов, для которых истинное значение равно отрицательному числу

В различных приложениях важно обеспечить баланс между точностью и полнотой, что можно оценить по показателю F1. F1 представляет собой гармоническое среднее точности и полноты.

$$F1 = \frac{1}{2} \times \frac{1}{\frac{1}{P}+\frac{1}{R}} = \frac{2PR}{P+R}$$

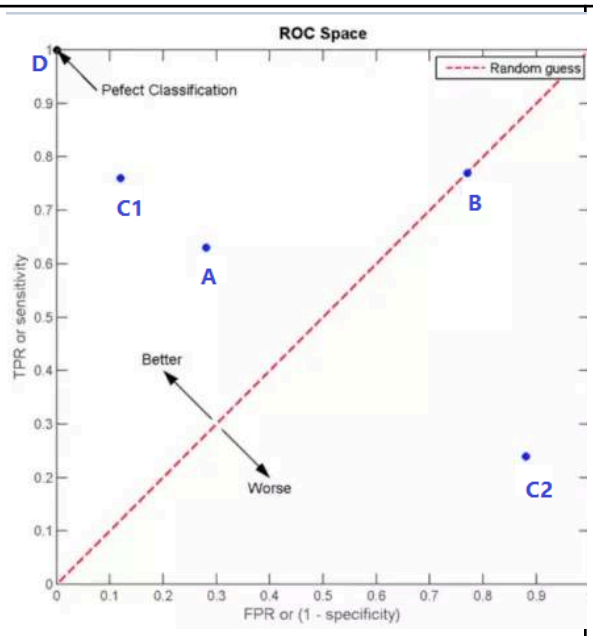
Методы из библиотеки sklearn Metrics позволяют рассчитывать матрицу смешивания, а также различные дополнительные показатели, основанные на этой матрице.

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score


y_true =[0,1,0,0,1,1,0,1]
y_pred =[0,0,1,0,1,0,0,1]
print(confusion_matrix(y_true, y_pred, labels=[1,0]) #1 — положительный пример; 0 — отрицательный пример.


print("accuracy:\t",accuracy_score(y_true, y_pred))
print("precision:\t",precision_score(y_true, y_pred))
print("recall:\t\t",recall_score(y_true, y_pred))
print("f1_score:\t",f1_score(y_true,y_pred))
```

（2） Кривая ROC и площадь под кривой (AUC)

Для несбалансированных задач нецелесообразно оценивать эффективность классификации по показателю точности. Полное название ROC — Receiver Operating Characteristic, а основным инструментом анализа является кривая ROC, нанесённая на двумерную плоскость. По оси X отмечается уровень ложноположительных результатов (FPR), по оси Y — уровень истинноположительных результатов (TPR). Для любого классификатора можно получить точку (TPR, FPR) на основе его результатов на

тестовых образцах.Каждая точка представляет собой значения показателей FPR (False Positive Rate) и TPR (True Positive Rate), рассчитанные при использовании различных пороговых значений. Пороговые значения соответствуют вероятностным оценкам, которые модель делает для отдельных образцов; их можно также называть «оценками» (scores).



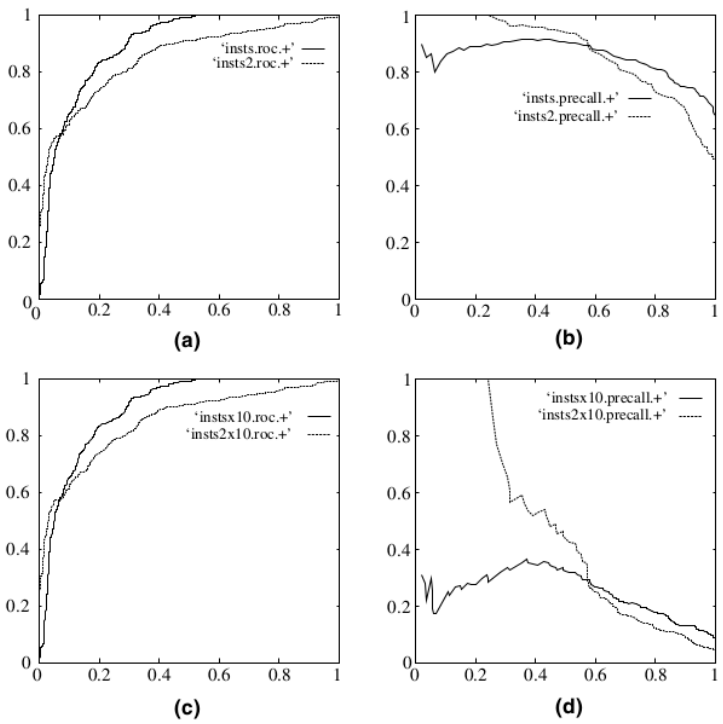
Точка A (TPR > FPR) свидетельствует о том, что оценка врача А в целом была правильной.

Точка B: результаты, полученные врачом, совпадают с результатами случайного отбора пациентов для исследования. Врач Б действует случайным образом: половина его предположений верны, а половина — нет.

Точка C2: Если врач говорит, что у вас болезнь, хотя на самом деле это не так (TPR < FPR), то скорее всего, у вас нет болезни. В таком случае нужно обратно понять его слова — это признак ненадежности врача.

Точка D: (TPR=1, FPR=0) — это идеальный результат классификации; врач демонстрирует высокий уровень профессионализма и делает все правильные диагнозы.

Например, при скрининге рака, если лишь 5% людей действительно страдают этим заболеванием, и все прогнозы оказываются корректными, точность модели достигает 95%. У кривой ROC есть важная особенность: она остаётся неизменной при изменении распределения положительных и отрицательных примеров в тестовом наборе. На практике в наборах данных часто наблюдается дисбаланс классов — то есть отрицательных примеров гораздо больше, чем положительных (или наоборот), — а также распределение в тестовых данных может изменяться со временем.



На рисунках (a) и (c) показаны кривые ROC, а на (b) и (d) — кривые точности и восприятия. На (a) и (b) представлены результаты классификации на исходном тестовом наборе с сбалансированным распределением положительных и отрицательных примеров. На (c) и (d) — результаты классификатора после увеличения числа отрицательных примеров в 10 раз. Очевидно, что кривая ROC практически не изменилась, тогда как кривая точности и восприятия значительно снизилась.

Хотя кривая ROC наглядно и удобно показывает производительность классификатора, люди всегда хотели иметь количественный показатель, отражающий его качество. Так появилась площадь под кривой ROC (AUC). Как следует из названия, AUC — это площадь под кривой ROC. Как правило, значение AUC находится в диапазоне от 0,5 до 1,0, и чем больше AUC, тем выше производительность.

1.4 Упражнение 2: Классификация рака молочной железы

Рак молочной железы — самое распространённое злокачественное новообразование у женщин. Он составляет почти треть всех случаев рака у американских женщин и является второй по частоте причиной смерти от онкологических заболеваний среди женщин. Рак молочной железы возникает из-за аномального роста клеток тканей молочной железы и обычно называют опухолью. Однако наличие опухоли не означает, что она злокачественная: опухоль может быть доброкачественной (неканцерогенной),

предраковой (предраковой) или злокачественной (раковой). Для диагностики рака молочной железы обычно используют такие методы, как МРТ, рентгенография молочной железы, ультразвуковое исследование и биопсия. Доброкачественный рак молочной железы после операции может быть полностью излечён.

База данных о раке молочной железы штата Висконсин (8 января 1991 г.) включает 699 образцов; каждый образец имеет 10 атрибутов и одну категорию.

1.4.1. Предварительная обработка и анализ данных

После получения набора данных обычно необходимо провести предварительную обработку и простую анализ. В ходе обработки следует выявить и устранить пропущенные значения и аномальные значения — например, удалить их или заменить на медиану. В данном примере пропущенные значения обозначены символом ?; их количество невелико, поэтому их можно просто удалить. См. код NB1_Clean_Data.ipynb.

```
df_breast = df_breast.replace(to_replace = "?", value = np.nan)
df_breast = df_breast.dropna(how = 'any')
```

С помощью статистических методов можно анализировать распределение различных параметров. Например, метод describe() позволяет получить информацию о средних значениях, дисперсии и распределении данных в виде таблицы. Пример кода см. в файле NB2_Explore_Data.ipynb.

	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
count	683.000000	683.000000	683.000000	683.000000	683.000000	683.000000	683.000000	683.000000	683.000000	683.000000
mean	4.442167	3.150805	3.215227	2.830161	3.234261	3.544656	3.445095	2.869693	1.603221	2.699854
std	2.820761	3.065145	2.988581	2.864562	2.223085	3.643857	2.449697	3.052666	1.732674	0.954592
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000
25%	2.000000	1.000000	1.000000	1.000000	2.000000	1.000000	2.000000	1.000000	1.000000	2.000000
50%	4.000000	1.000000	1.000000	1.000000	2.000000	1.000000	3.000000	1.000000	1.000000	2.000000
75%	6.000000	5.000000	5.000000	4.000000	4.000000	6.000000	5.000000	4.000000	1.000000	4.000000
max	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	4.000000

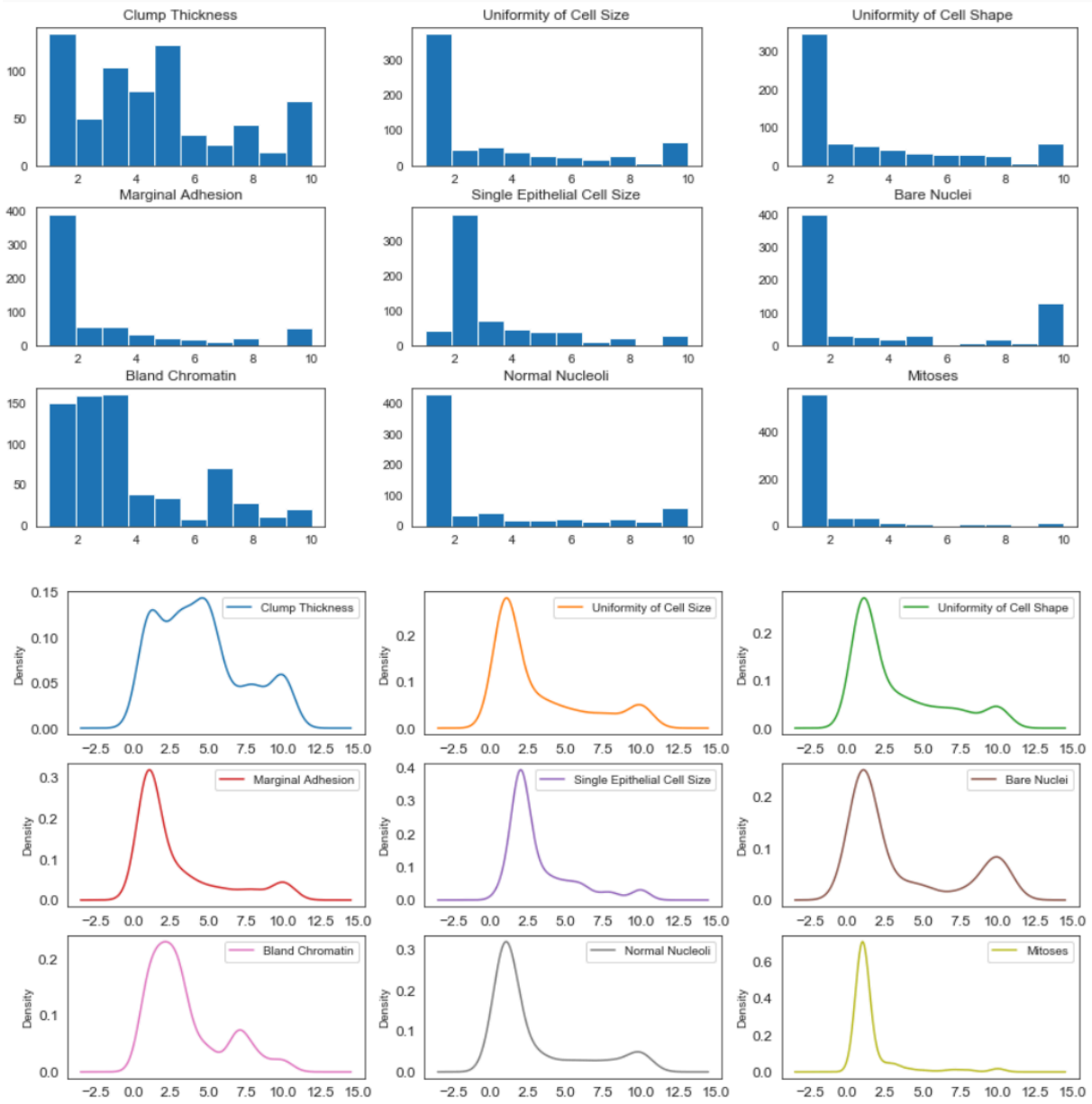
Метод `groupby()` позволяет объединять данные по определённым критериям и подсчитывать количество элементов в каждой категории.

```
# Group by class and review the output.
class_gr = data.groupby('Class', axis=0)
pd.DataFrame(class_gr.size(), columns=['# of observations'])
```

Class	# of observations
2	444
4	239

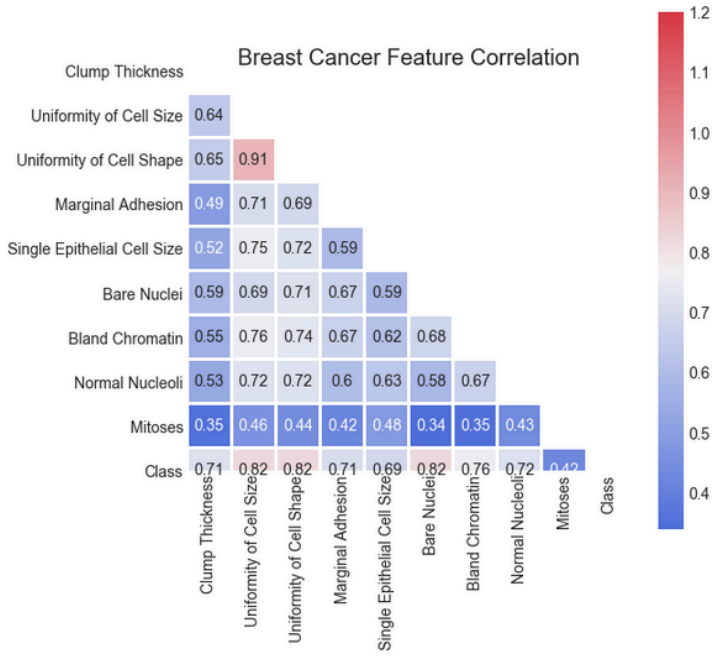
Количество доброкачественных образований составляет 444, а злокачественных — 239.

Распределение данных можно представить с помощью столбчатого графика (histogram) или линейного графика (plot). В столбчатом графике горизонтальная ось отражает значения признаков, а вертикальная ось — количество соответствующих значений; в линейном графике вертикальная ось показывает долю данных.



Для отображения степени корреляции между данными можно использовать тепловые

карты. Суть тепловой карты заключается в использовании разных цветов для обозначения различных значений. На таких картах значения корреляции находятся в диапазоне от -1 до 1; чем больше значение корреляции, тем выше степень взаимосвязи между признаками.



1.4.2. Наборы данных для тестирования и проверки

Перед тем как использовать модель для классификации или прогнозирования необозначенных данных, обычно применяют метод кросс-валидации (Cross-Validation) для проверки её точности. Кросс-валидация помогает избежать переобучения, вызванного чрезмерной сложностью модели. Это практический статистический метод, при котором выборку данных разделяют на несколько подмножеств для проверки. Сначала анализ проводят на одном подмножестве, а остальные используются для подтверждения и проверки полученных результатов. Первоначальное подмножество называется обучающим набором (training set), а остальные — набором для проверки (validation set), который, как правило, отличается от тестового набора (test set).

Существует три основных метода кросс-валидации: метод Hold-Out, K-кратная кросс-валидация (обозначается как K-CV) и метод Leave-One-Out (обозначается как LOO-CV). Ключевым моментом при использовании любого из этих методов является правильное разделение обучающей и валидационной выборок. Для выбора подходящего метода кросс-валидации в библиотеке sklearn модель Selection доступен набор инструментов для разделения выборок; примеры реализации этих методов

приведены в файле Validation.ipynb.

Метод разделения данных	Описание
Hold-Out Method	Метод выделения данных (leave-one out) предполагает разделение набора данных на тестовый и обучающий наборы в соответствии с заданным соотношением.
K-fold Cross Validation	K-кратная перекрёстная проверка решает проблему потери информации о выборке, характерную для метода выделения. Она делит обучающий набор на K равных взаимоисключающих поднаборов, каждый из которых используется в качестве обучающего набора, а оставшийся — в качестве тестового. Таким образом формируется K пар обучающих и тестовых наборов, которые применяются в K циклах обучения и тестирования. В итоге возвращается среднее значение по K тестовым наборам.
Метод оставления	Каждый раз оставляется один элемент данных в качестве тестового набора, а остальные элементы используются для обучения.

1.4.3. Реализация и сравнение методов классификации

При выборе алгоритма классификации стоит учитывать размер обучающего набора, размерность признаков, их взаимную независимость, а также требования к производительности и потреблению памяти. Если обучающий набор мал, можно выбрать простой байесовский метод и алгоритм K-ближайших соседей, но важно правильно выбрать значение k, чтобы избежать переобучения. Если предполагается взаимная независимость признаков (что на практике зачастую сложно), простой байесовский метод более подходит. Если вы планируете в будущем добавлять в обучающий набор всё больше данных и быстро интегрировать их в модель, лучше использовать логистическую регрессию.Алгоритм решающего дерева легко интерпретировать, удобен для работы с признаками и не требует параметров, поэтому не нужно беспокоиться о наличии аномальных точек или линейной разделимости данных. Однако при добавлении новых данных алгоритм должен быть перестроен заново, что может привести к переобучению. Служащие векторные машины (SVM) предлагают множество ядерных функций, и именно это создаёт определённые трудности при выборе подходящей. Однако при правильном выборе и использовании

подходящей ядерной функции можно добиться достаточно хороших результатов. Ниже приведены достоинства и недостатки различных алгоритмов классификации для ориентации при выборе, как показано в таблице 11.

Сравнение алгоритмов классификации в таблице 11

алгоритм	**Преимущества**	**Недостаток**
Метод классификации Байеса	1) Необходимо оценить небольшое количество параметров; система нечувствительна к отсутствию данных. 2) Наличие прочной математической базы и стабильной эффективности классификации.	1) Предполагается, что эти свойства взаимно независимы, однако это часто не соответствует действительности (например, человек любит есть помидоры и яйца, но не любит их жарить вместе). 2) Необходимо знать априорные вероятности. 3) В процессе принятия решений по классификации возникает ошибка.
Дерево решений	1) Не требуется никаких знаний в определённой области или предположений относительно параметров. 2) Подходит для работы с высокомерными данными. 3) Просто и понятно. 4) Обработка большого объема данных за короткое время позволяет получить результаты, которые являются практически выполнимыми и эффективными. 5) Способность одновременно обрабатывать как данные, так и обычные атрибуты.	1) Для данных с неодинаковым количеством образцов в различных категориях информацияльный прирост склонен к увеличению для признаков, имеющих больше значений. 2) Просто подвержен переобучению (overfitting). 3) Игнорирование взаимосвязей между атрибутами. 4) Онлайн-обучение не поддерживается.
SVM (Support Vector Machine) — это метод машинного обучения на основе векторов поддержки.	1) Это позволяет решать проблемы машинного обучения при небольшом объеме данных. 2) Повышение способности к обобщению. 3) Методы, способные решать задачи высокого размера и нелинейных процессов; классификация текстов с очень большим	1) Чувствительность к отсутствующим данным. 2) Затраты памяти велики, и их сложно объяснить. 3) Процесс запуска и настройки параметров довольно хлопотный.

	количеством параметров по-прежнему пользуется популярностью. 4) Избежать проблем, связанных с выбором структуры нейронной сети и достижением локальных минимумов.	
KNN (K-ближайших соседей)	1) Идея проста, теория зрелая; данный метод может использоваться как для классификации, так и для регрессии. 2) Может использоваться для нелинейной классификации; 3) Сложность времени выполнения операций обучения составляет O(n). 4) Высокая точность, отсутствие предположений относительно данных, низкая чувствительность к аномальным значениям (outliers);	1) Объем вычислений слишком велик. 2) Неравномерное распределение образцов может привести к ошибочным выводам. 3) Требуется большое количество оперативной памяти. 4) Выходные данные не обладают достаточной интерпретируемостью.
Логистическая регрессия	1) Быстрый темп выполнения задач. 2) Просто и понятно; сразу видны веса всех характеристик. 3) Модель легко обновляется для включения новых данных. 4) Если необходимо использовать вероятностную модель, то пороги классификации следует динамически корректировать.	Обработка признаков сложна: требуется нормализация и значительная работа по инженерии признаков.

В качестве примера можно привести модель логистической регрессии. Процесс классификации с использованием такой модели включает следующие шаги: создание модели, обучение модели на обучающем наборе данных и прогнозирование результатов на тестовом наборе данных с помощью полученной модели.

```
import LogisticRegression from sklearn.linear_model

lr = LogisticRegression (solver='lbfgs') # Инициализация
lr.fit(x_train, y_train) #Процесс обучения
`y_pred = lr.predict(x_test)` #Предсказание
```

Затем, на основе сравнения прогнозных и фактических результатов, можно получить такие показатели, как точность и коэффициент восприятия (recall).

```
from sklearn.metrics import classification_report
acc_lr = accuracy_score(y_test, y_pred)
print(acc_lr)
print(classification_report(y_test,y_pred,target_names=class_name))
```

```
0.9804878048780488
              precision    recall  f1-score   support

           2       0.98      0.99      0.98       122
           4       0.99      0.96      0.98        83

    accuracy                           0.98       205
   macro avg       0.98      0.98      0.98       205
weighted avg       0.98      0.98      0.98       205
```

Другие модели функционируют аналогично.

1.5 Упражнение 3: Оптимизация классификатора

В моделях машинного обучения параметры влияют на такие показатели, как точность и восприятие. Процесс выбора оптимальных параметров — это процесс настройки и оптимизации модели, который также называют гиперпараметрической оптимизацией. К основным методам относятся сетевой поиск, случайный поиск, байесовская оптимизация и Hyperband. Сетевой поиск представляет собой метод брутфорса: в заданном пространстве гиперпараметров пробуются все возможные комбинации, и в итоге находится оптимальная. Метод GridSearchCV из библиотеки sklearn реализует сетевой поиск. При большом количестве параметров пространство комбинаций становится очень большим, и эффективность поиска снижается. Случайный поиск выбирает гиперпараметры из выборки в пространстве поиска и отбирает наиболее подходящую комбинацию.Метод RandomizedSearchCV из библиотеки Sklearn реализует эту функцию.

Оба метода используются аналогичным образом: они ищут оптимальное решение в пространстве параметров модели. Полный код см. в файле NB4_Optimization.ipynb.

```
# Train classifiers.
kernel_values = [ 'linear' , 'poly' , 'rbf' , 'sigmoid' ]
param_grid = {'C': np.logspace(-3, 2, 6), 'gamma': np.logspace(-3, 2, 6),'kernel':
```

```
kernel_values}

grid = GridSearchCV(SVC(), param_grid=param_grid, cv=5)
grid.fit(x_train, y_train)
```

В возвращаемом объекте GridSearchCV параметр best_params_ обозначает наилучшие параметры модели, а параметр best estimator_ — наилучший классификатор.

Процесс настройки модели включает три этапа: инженеринг признаков, выбор модели и выбор алгоритма. В настоящее время существуют и фреймворки, призванные автоматизировать этот процесс. Среди них autosklearn поддерживается только на системах Linux.

Hyperopt https://github.com/hyperopt/hyperoptHyperopt https://github.com/hyperopt/hyperopt

BayesianOptimization https://github.com/fmfn/BayesianOptimization BayesianOptimization https://github.com/fmfn/BayesianOptimization

Spearmint https://github.com/HIPS/Spearmint Spearmint https://github.com/HIPS/Spearmint

Advisor https://github.com/tobegit3hub/advisor Advisor https://github.com/tobegit3hub/advisor

NNI https://github.com/microsoft/nni

Autoweka(Java) https://github.com/automl/autoweka Autoweka(Java) https://github.com/automl/autoweka

Autosklearn https://automl.github.io/auto-sklearn/master/

1.6 Задание для эксперимента

Сети данных UCI предоставляют множество наборов данных для анализа и машинного обучения; в настоящее время их насчитывается 591. https://archive.ics.uci.edu/datasets. Используя знания, полученные в ходе экспериментов в этом разделе, выберите один из наборов данных, подходящий для классификации, кодируйте его и проанализируйте точность модели. (Обязательно)

Из двух вариантов выберите один:

Конкурс по работе с большими данными Tianchi – Введение в управление рисками в

финансовой сфере для новичков: прогнозирование просрочек по кредитам, https://tianchi.aliyun.com/competition/entrance/531830/information.

Конкурс больших данных Tianchi — AFAC2023: понимание финансовых сценариев — Задание 1: временные прогнозы в сфере финансового маркетинга

https://tianchi.aliyun.com/competition/entrance/532093/information