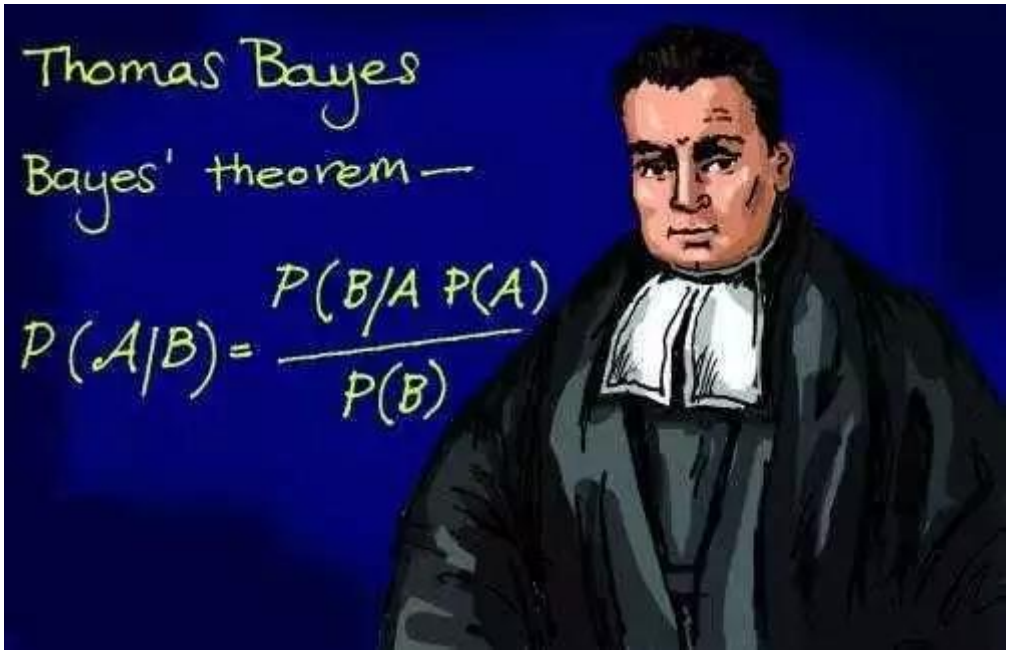


1. 贝叶斯分类器

1.1 贝叶斯的引入

贝叶斯公式：当分析样本大到接近总体数时，样本中事件发生的概率将接近于总体中事件发生的概率。

假设 X,Y 是一对随机变量，它们的联合概率 $P(X=x,Y=y)$ 是指 X 取值 x 且 Y 取值 y 的概率，条件概率是指一个随机变量在另一个随机变量取值已知的情况下取某一特定值的概率。例如，条件概率 $P(Y=y \mid X=x)$ 是指变量 X 在取值 x 的情况下，变量 Y 取值 y 的概率。 X 和 Y 的联合概率和条件概率如下关系：

$$P(X,Y)=P(Y \mid X)\times P(X)=P(X \mid Y)\times P(Y)$$

从而推出贝叶斯定理：

$$P(Y \mid X)=\frac{P(X \mid Y)P(Y)}{P(X)}$$

example:考虑两队之间的足球比赛：队 0 和队 1，假设 65%的比赛队 0 获胜，剩余的比赛队 1 获胜。队 0 获胜的比赛只有 30%是队 1 的主场，而队 1 取胜的比赛中 75%获胜。如果下一场比赛在队 1 的主场进行，那一支球队最有可能获胜:
随机变量 X 代表东道主，随机变量 Y 代表比赛的胜利者。 X,Y 都在（0,1）中取值。

队伍 0 取胜的概率是 $\mathrm{P(Y=0)=0.65}$

队伍 1 取胜的概率是 $\mathrm{P(Y=1)=1-P(Y=0)=0.35}$

队伍 1 取胜作为东道主的概率是 $\mathrm{P(X=1|Y=1)=0.75}$

队伍 0 取胜企鹅队 1 作为东道主的概率是 $\mathrm{P(X=1|Y=0)=0.3}$

我们的目的是计算 $\mathrm{P(Y=1|X=1)}$，即队 1 在主场获胜的概率，并与 $\mathrm{P(Y=0|X=1)}$ 比较：

$$P(Y=1|X=1)=\frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1)}$$
$$P(Y=1|X=1)=\frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1,Y=1)+P(X=1,Y=0)}$$
$$P(Y=1|X=1)=\frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1|Y=1)\times P(Y=1)+P(X=1|Y=0)\times P(Y=0)}$$
$$P(Y=1|X=1)=\frac{0.75\times 0.35}{0.75\times 0.35+0.3\times 0.65}$$
$$P(Y=1|X=1)=0.5738$$

贝叶斯变量在分类中的应用：
设 X 表示属性集，Y 表示类变量。如果类变量和属性之间的关系不确定，那么我们可以把 X 和 Y 看做随机变量，用 $P(Y|X)$ 以概率的方式捕捉二者之间的关系。

这个条件概率又称为 Y 的后验概率，与之相对地， $P(Y)$ 称为 Y 的先验概率。

通过使用 $P(Y)$ 、类的条件概率 $P(X|Y)$ 和证据 $P(X)$ 来表示后验概率：

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

1.2 朴素贝叶斯分类器

给定类标号 y，朴素贝叶斯分类器在估计类条件概率时假设属性之间条件独立。条件独立假设可以形式化表述为：

$$P(X \mid Y = y) = \prod_{i=1}^{d} P(X_i \mid Y = y)$$

条件独立性：

$$P(X, Y \mid Z) = P(X \mid Z) \times P(Y \mid Z)$$

朴素贝叶斯分类器：

有了条件独立的假设，不必计算 X 的每一个组合的类的条件概率，只需对给定的 Y，计算每一个 X_i 的条件概率。朴素贝叶斯分类器对每个类 Y 计算后验概率:

$$P(Y \mid X) = \frac{P(Y)\prod_{i=1}^{d} P(X_i \mid Y)}{P(X)}$$

对于所有的 Y，$P(X)$ 是固定的，因此只要找到使得 $P(Y)\prod_{i=1}^{d} P(X_i \mid Y)$ 最大的类就行了。

那么：1、分类属性的条件概率我们上面已经计算过；

2、估计连续值属性的条件概率：

1) 连续特征特征离散化，后面和分类属性类似

2) 假设连续变量服从某种概率分布，然后使用训练数据估计分布的参数。高斯分布通常被用来表示连续属性的类条件概率分布。该分布有两个参数，均值 μ 和方差 σ_2，对于每个类 y_j 和属性 x_i 的类条件概率等于：

$$P(X_i = x_i \mid Y = y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{\frac{(x_i - u_{ij})^2}{\sigma_{ij}^2}}$$

给定以下测试记录有以下属性集：X=（有房=否，婚姻状况=已婚，年收入=$120K）需要根据训练集中的可用信息计算后验概率 P(Yes|X)和 P(No|X）如果 P(Yes|X）>P(No|X)，那么记录为 Yes,反之，分类为 No.

Tid	有房	婚姻状况	年收入	拖欠贷款
1	是	单身	125K	否
2	否	已婚	100K	否
3	否	单身	70K	否
4	是	已婚	120K	否
5	否	离婚	95K	是
6	否	已婚	60K	否
7	是	离婚	220K	否
8	否	单身	85K	是
9	否	已婚	75K	否
10	否	单身	90K	是

其中年收入属性关于 NO 否的属性计算如下：

$$\bar{x} = \frac{125 + 100 + 70 + \cdots + 75}{7} = 110$$

$$s^2 = \frac{(125 - 110)^2 + (100 - 110)^2 + \cdots + (75 - 110)^2}{7(6)} = 2975$$

$$s = \sqrt{2975} = 54.54$$

相应其他类条件概率汇总：

$$P(\text{有房}=\text{是}\mid NO)=3/7$$
$$P(\text{有房}=\text{否}\mid NO)=4/7$$
$$P(\text{有房}=\text{是}\mid \text{Yes})=0$$
$$P(\text{有房}=\text{否}\mid Yes)=1$$
$$P(\text{婚姻状况}=\text{单身}\mid \text{No})=2/7$$
$$P(\text{婚姻状况}=\text{离婚}\mid \text{No})=1/7$$
$$P(\text{婚姻状况}=\text{已婚}\mid \text{No})=4/7$$
$$P(\text{婚姻状况}=\text{单身}\mid \text{Yes})=2/3$$
$$P(\text{婚姻状况}=\text{离婚}\mid \text{Yes})=1/3$$
$$P(\text{婚姻状况}=\text{已婚}\mid \text{Yes})=0$$

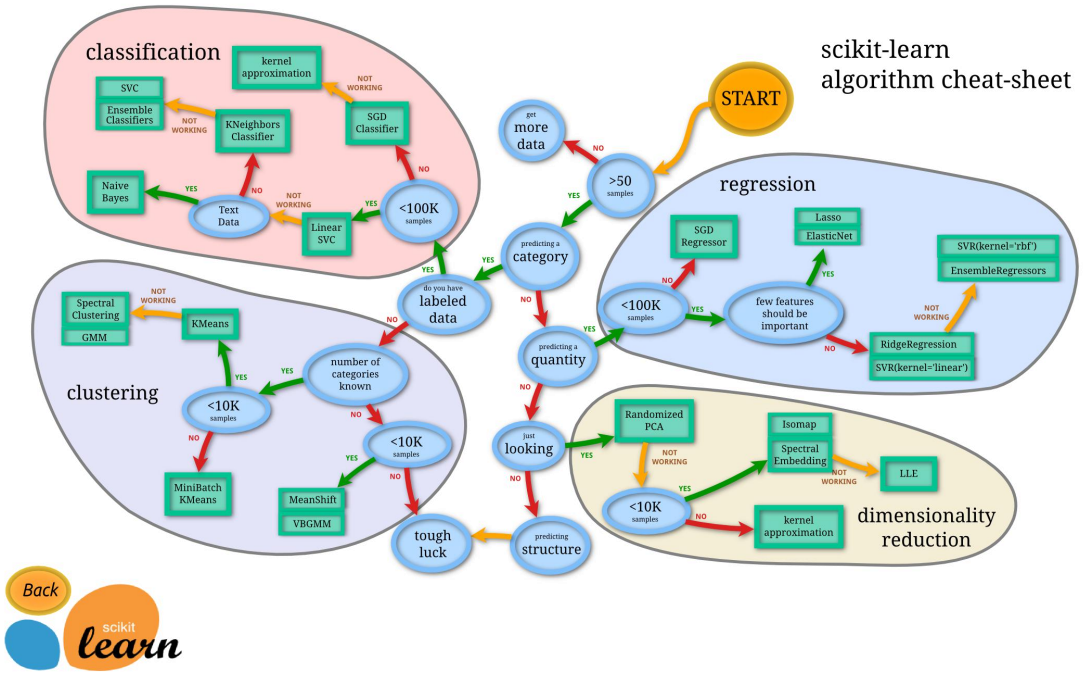
则预测 $X=$（有房 $=$ 否，婚姻状况 $=$ 已婚，年收入 $=\$120K$) 计算后验概率 $P(Y)\prod_{i=1}^{d}P(X_i\mid Y)$：

$$P(\text{No}\mid X)=P(\text{有房}=\text{否}\mid \text{No})\times P(\text{婚姻状况}=\text{已婚}\mid \text{No})\times P(\text{年收入}=\$120\mid No)$$
$$=4/7\times 4/7\times 0.0072=0.0024$$

$$P(\text{No}\mid X)=P(\text{有房}=\text{否}\mid \text{No})\times P(\text{婚姻状况}=\text{已婚}\mid \text{No})\times P(\text{年收入}=\$120\mid No)$$
$$=4/7\times 4/7\times 0.0072=0.0024$$

放到一起的可以得到类 No 的后验概率 P（No|X）=α×7/10×0.0024=0.0016α其中 α=1/P(x)是个常量。类 0 概率为 0 这样 P(No|X)>P(Yes|X）,所以记录分类为 No.

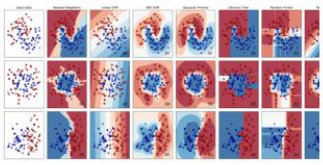
2. Sklearn 部分包介绍



Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

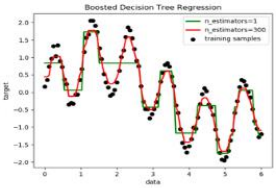


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

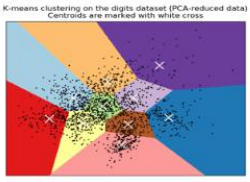


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

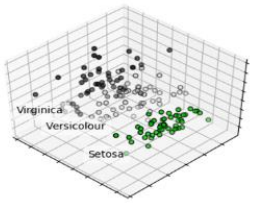


Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: k-Means, feature selection, non-negative matrix factorization, and more...

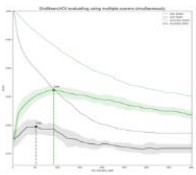


Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics, and more...

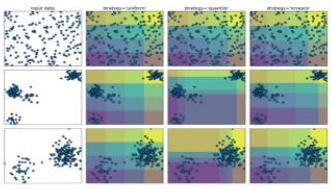


Examples

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.
Algorithms: preprocessing, feature extraction, and more...



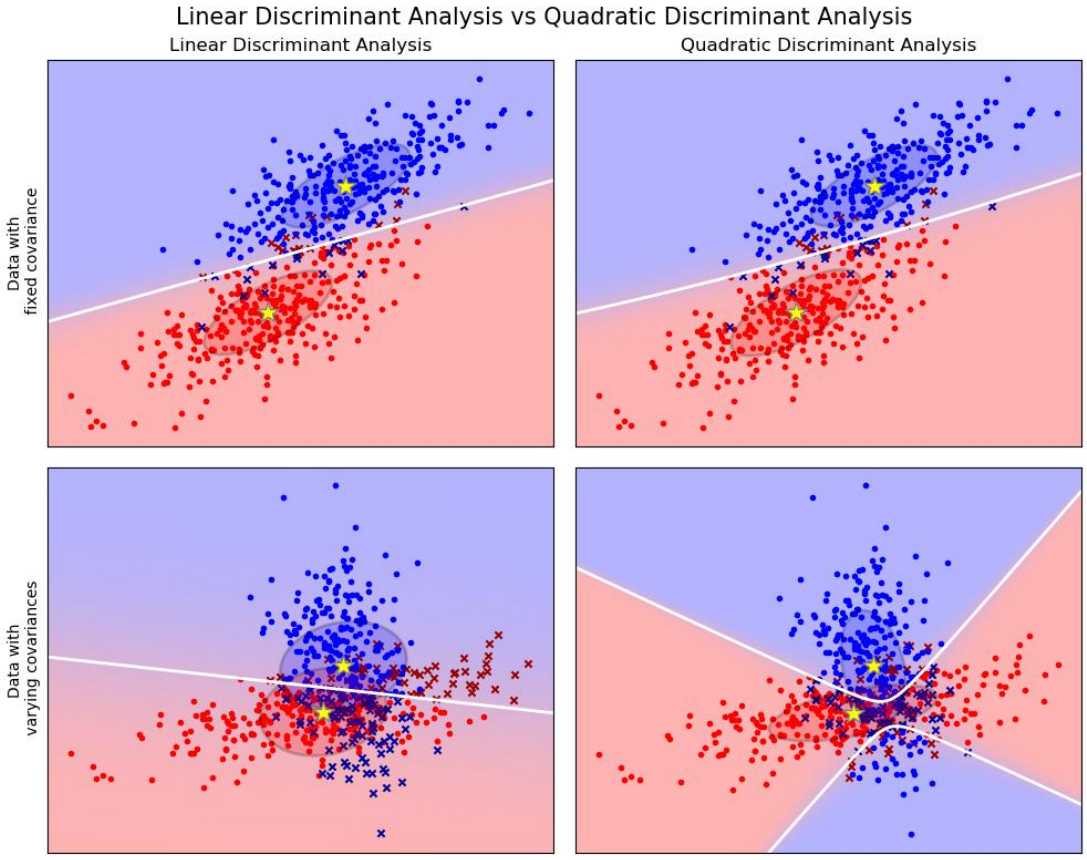
Examples

官网地址：https://scikit-learn.org/

2.1 监督学习-Supervised learning

2.1.1 Generalized Linear Models 广义线性模型

2.1.1.1 线性判别分析法(discriminant_analysis.LinearDiscriminantAnalysis)和二次判别分析法(discriminant_analysis.QuadraticDiscriminantAnalysis



2.1.2 SVM

2.1.3 Naive Bayes

代码：

```
from sklearn import datasets
iris = datasets.load_iris()
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
y_pred = gnb.fit(iris.data, iris.target).predict(iris.data)
print("Number of mislabeled points out of a total %d points : %d"... % (iris.data.shape[0],(iris.target != y_pred).sum()))
```

2.1.4 Feature selection

2.1.4.1 Removing features with low variance 删除低方差特征：

```
from sklearn.feature_selection import VarianceThreshold
X = [[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]

sel = VarianceThreshold(threshold=(.8 * (1 - .8)))

sel.fit_transform(X)
```

2.1.4.2 单变量特征选择:Univariate feature selection

SelectKBest 保留评分最高的 K 个特征

SelectPercentile 保留最高得分百分比之几的特征

对每个特征应用常见的单变量统计测试: 假阳性率（false positive rate）SelectFpr, 伪发现率（false discovery rate） SelectFdr , 或者族系误差（family wise error） SelectFwe 。

GenericUnivariateSelect 允许使用可配置方法来进行单变量特征选择。它允许超参数搜索评估器来选择最好的单变量特征。

这些对象将得分函数作为输入，返回单变量的得分和 p 值 （或者仅仅是 SelectKBest 和 SelectPercentile 的分数）:

对于回归: f_regression , mutual_info_regression

对于分类: chi2 , f_classif , mutual_info_classif

2.2 模型选择和评估 Model selection and evaluation

2.2.1 Cross-validation: 交叉验证

2.2.2 Tuning the hyper-parameters of an estimator 调参

2.2.3 Model evaluation: quantifying the quality of predictions 模型效果评估

2.3 Preprocessing data 预处理数据

2.3.1 Standardization, or mean removal and variance scaling 标准化，或均值去除和方差缩放

2.3.1.1 标准化

```
from sklearn import preprocessing
import numpy as np
X_train = np.array([[ 1., -1., 2.],[ 2., 0., 0.],[ 0., 1.,
-1.]])
X_scaled = preprocessing.scale(X_train)
X_scaled
```

通过删除平均值和缩放到单位方差来标准化特征

```
scaler = preprocessing.StandardScaler().fit(X_train)
```

min_max_scaler 主要是为了 train 和 test 分布保持一致

```
min_max_scaler = preprocessing.MinMaxScaler()
X_train_minmax = min_max_scaler.fit_transform(X_train)
```

max_abs_scaler 通过除以每个特征中的最大值来将训练数据缩放到[-1，1]范围内。它适用于已经以零或稀疏数据为中心的数据。

2.3.2 Encoding categorical features

preprocessing.OrdinalEncoder()
preprocessing.OneHotEncoder()

2.3.3 Generating polynomial features poly 特征

X 的特征从（X_1， X_2）转换为（1， X_1， X_2， X_1^2， X_1X_2， X_2^2）：

```
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
X = np.arange(6).reshape(3, 2)
poly = PolynomialFeatures(2)
poly.fit_transform(X)
```

X 的特征从（X_1，X_2，X_3）转换为（1，X_1，X_2，X_3，X_1X_2，X_1X_3，X_2X_3，$X_1X_2X_3$）：

```
X = np.arange(9).reshape(3, 3)
poly = PolynomialFeatures(degree=3, interaction_only=True)
poly.fit_transform(X)
```

3.「二分类算法」提供银行精准营销解决方案-进一步特征工程

4. kaggle-Porto Seguro’s Safe Driver Prediction：塞古罗港的安全驾驶员预测-预测驾驶员明年是否会提出保险索赔

4.1 数据 EDA

4.2 baseline

4.3 进一步特征工程

4.4 模型融合

5.开始标准化我们的模块吧