Logistic Regression

做二元分类的时候吗，当给定一个输入的时候，我们很多时候想知道预测的y值有多大概率是0，有多大概率是1

y_test=P(y=1|x),x是实数集中的n维向量

当训练出参数W的时候，如何求y的预测

使用线性回归，效果不是很好，因为y的预测有可能大于1或者小于0

这个时候，我们一般习惯在线性回归方程的外面再套一层sigmoid()函数来进行转换

sigmoid函数与y轴的交点是0.5，是一个平滑的曲线，x越大越接近1，x越小，越接近0

函数方程$y=\frac{1}{1+e^{-x}}$

当x特别大的时候，y约等于1

当x特别小的时候，y约等于0

在神经网络里面，一般会把线性回归的w和b分开，把b作为一个拦截器

损失函数：用来评估算法性能的函数，作用于单个样本，可以用二分之一平方差，也可以使用别的函数。这里我们使用f(x)=$-\left(ylogy1+(1-y)log(1-y1)\right)$这个式子来计算真实值与预测值之间的差值，同样的，式子的值越小越好，f(x)的值越小越好

那么这个式子为什么可以作为损失函数呢？

当y=1的时候，y的预测值就应该尽可能的大，但是由于y的预测值，是由sigmoid函数计算得出，所以最大也不会超过1，因此应该是y的预测值越接近1越好

同样的，当y=0的时候，y的预测值，越接近0越好

成本函数：用来评估算法性能的函数，作用于全部样本

$$-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}logy1^{(i)}+(1-y^{(i)})log(1-y1^{(i)})]$$

再往下，使用随机梯度下降算法来求参数W

然后得到最终的函数f，再根据新的数据来预测值

```python
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import pandas as pd
from sklearn import preprocessing
reader = pd.read_table('E:\\DateSet\\dating.txt',usecols=(0, 1, 2))
datingLabels = pd.read_table('E:\\DateSet\\dating.txt', usecols=(3,))
scaler = preprocessing.MinMaxScaler(feature_range=(0, 1)).fit(reader)
normMat=scaler.transform(reader)
# iris = datasets.load_iris()
# X = iris.data
# y = iris.target
lr = LogisticRegression()
for i in range(10):
    X_train, X_test, y_train, y_test = train_test_split(normMat,
datingLabels, test_size=0.1)
    lr.fit(X_train, y_train)
    score = lr.score(X_test,y_test)
    print(score)
y=lr.predict([[69673,14.239195,0.261333]])
print(y)
```

朴素贝叶斯

朴素贝叶斯法是基于贝叶斯定理与特征条件独立假设的分类方法

朴素贝叶斯公式转化：

$$p(类别|特征)=\frac{p(特征|类别)p(类别)}{p(特征)}$$

案例：

帅？	性格好？	身高？	上进？	嫁与否
帅	不好	矮	不上进	不嫁
不帅	好	矮	上进	不嫁
帅	好	矮	上进	嫁
不帅	好	高	上进	嫁
帅	不好	矮	上进	不嫁
不帅	不好	矮	不上进	不嫁
帅	好	高	不上进	嫁
不帅	好	高	上进	嫁
帅	好	高	上进	嫁
不帅	不好	高	上进	嫁
帅	好	矮	不上进	不嫁
帅	好	矮	不上进	不嫁

现在给我们的问题是，如果一对男女朋友，男生想女生求婚，男生的四个特点分别是不帅，性格不好，身高矮，不上进，请你判断一下女生是嫁还是不嫁？

转为数学问题就是比较p(嫁|(不帅、性格不好、身高矮、不上进))与p(不嫁|(不帅、性格不好、身高矮、不上进))的概率，谁的概率大

$$p\left(\text{嫁}\mid\text{不帅、性格不好、身高矮、不上进}\right)=\frac{p\left(\text{不帅、性格不好、身高矮、不上进}\mid\text{嫁}\right)*\mathrm{p}(\text{嫁})}{\mathrm{p}(\text{不帅、性格不好、身高矮、不上进})}$$

等等，为什么这个成立呢？学过概率论的同学可能有感觉了，这个等式成立的条件需要特征之间相互独立，这也就是为什么朴素贝叶斯分类有朴素一词的来源，朴素贝叶斯算法是假设各个特征之间相互独立，那么这个等式就成立了

$$p\left(\text{嫁}\mid\text{不帅、性格不好、身高矮、不上进}\right)=\frac{p\left(\text{不帅、性格不好、身高矮、不上进}\mid\text{嫁}\right)*\mathrm{p}(\text{嫁})}{\mathrm{p}(\text{不帅、性格不好、身高矮、不上进})}$$

$$=\frac{\mathrm{p}(\text{不帅}\mid\text{嫁})*\mathrm{p}(\text{性格不好}\mid\text{嫁})*\mathrm{p}(\text{身高矮}\mid\text{嫁})*\mathrm{p}(\text{不上进}\mid\text{嫁})*\mathrm{p}(\text{嫁})}{\mathrm{p}(\text{不帅})*\mathrm{p}(\text{性格不好})*\mathrm{p}(\text{身高矮})*\mathrm{p}(\text{不上进})}$$

$$p\left(\text{不嫁}\mid\text{不帅、性格不好、身高矮、不上进}\right)=\frac{p\left(\text{不帅、性格不好、身高矮、不上进}\mid\text{不嫁}\right)*\mathrm{p}(\text{不嫁})}{\mathrm{p}(\text{不帅、性格不好、身高矮、不上进})}$$

$$=\frac{\mathrm{p}(\text{不帅}\mid\text{不嫁})*\mathrm{p}(\text{性格不好}\mid\text{不嫁})*\mathrm{p}(\text{身高矮}\mid\text{不嫁})*\mathrm{p}(\text{不上进}\mid\text{不嫁})*\mathrm{p}(\text{不嫁})}{\mathrm{p}(\text{不帅})*\mathrm{p}(\text{性格不好})*\mathrm{p}(\text{身高矮})*\mathrm{p}(\text{不上进})}$$