UNIVERSIDADE DE ÉVORA

Aprendizagem Automática - Trabalho 1

Bárbara Loureiro l48469, Ricardo Dias l48388

dezembro 2022

1 Introdução/Objetivos

Foi proposto a realização de um trabalho prático cujo objetivo consiste em implementar o algoritmo *Naive Bayes* para tipos nominais de dados, em *Python*, um estimador suavizado (*smooth estimator*) e a avaliação do classificador através da exatidão e precisão.
Este aceitará dados nominais.
A abordagem escolhida será a que, para os conjuntos de dados que nos foram facultados, nos pareceu mais simples e com menor esforço computacional, nesta iremos guardar os dados de treino e utiliza-los para fazer os cálculos das previsões, tendo em conta os atributos do teste. Um caminho diferente seria fazer os cálculos de todas as combinações possíveis e depois apenas recorrer a esses cálculos pré-feitos para fazer a estimação.
Para implementar a classe utilizámos apenas o módulo *Pandas* e para fazer os testes utilizámos, em alguns casos, a divisão dos dados com a função
train_test_split() do ambiente *scikit-learn*.

2 Implementação

Este trabalho tem nele quatro funções, de certa forma, comuns a quase todos os métodos de *Machine Learning*. Uma função de **fit(X_train,y_train)**, de **predict(X_test)**, de **accuracy_score(X_test,y_test)** e de **precision_score(X_test, y_test)**

- **fit(X_train,y_train)** - Esta função serve para guardar os dados que servem de informação base para fazer as previsões do modelo. O conjunto de treino é guardado no argumento especial *self*.

- **predict(X_test)** - Esta função serve para prever as classes dos atributos presentes no conjunto de teste (X_test) com base nos dados facultados ao modelo, na função **fit()**. A previsão é feita através do Teorema Bayes:

$$P(c \,|\, X) = \frac{P(X \,|\, c)P(c)}{P(X)} \tag{1}$$

  Para fazer os cálculos, calculámos o $P(c)$, chamado na literatura como "likelihood". Este é cálculo das proporção das classes nos dados de *train*, ou seja, a quantidade de classes c_j a dividir pelo cardinal dos registos.
  Cálculámos ainda o $P(X|c)$, chamado na literatura de "prior", que se baseia na multiplicação sucessiva dos $P(X_j|c)$, $(j = 1, ..., n)$, e que significa a probabilidade de existir um certo atributo condicionado a uma classe. Este último cálculo só é efetuado através da multiplicação porque para o cálculo do *Naive Bayes* assumimos a independência dos atributos.
  Exemplo: Se $X = X_1, X_2$ e assumirmos o X_1 independente de X_2, então:

$$P(X|c) = P(X_1 \cap X_2|c) = \frac{P(X_1 \cap X_2 \cap c)}{P(c)} \tag{2}$$

  Que pela independência dos atributos (em que $P(A \cap B) = P(A)P(B)$, se A e B independentes), fica:

$$\frac{P(X_1 \cap X_2 \cap c)}{P(c)} = \frac{P(X_1 \cap c)P(X_2 \cap c)}{P(c)} \tag{3}$$

  Replicando para todos os atributos.
  Neste tipo de cálculo não há necessidade de calcular o valor de $P(X)$, "evidence", por este ser uma constante.
  A decisão de qual é o rótulo a dar a cada linha do conjunto de teste é feita pelo maior valor da probabilidade $P(c_i \,|\, X)$, $(i = 1, 2, ..., \#Classes)$.
  A previsão pode ser suavizada variando o valor α, e a suavização é feita pela fórmula:

$$P(\theta) = \frac{x_i + \alpha}{N + d\alpha}, d = \#atributos_i,\ N = número\, de\, observações. \tag{4}$$

  , se α=1 temos um estimador de LaPlace.
  De forma a tentar contornar o problema que ocorre quando há atributos

no teste que não existem no treino, suspendemos a multiplicação. Esta suspensão acontece quando estão reunidas duas condições:

1. A contagem do atributo, em questão, na coluna é 0;
2. Quando o α=0.

Se essa contagem der zero e o α for nulo, eventualmente a probabilidade seria 0, e de forma a evitar o anulamento das probabilidades condicionadas, omitimos a multiplicação por zero.

- **accuracy_score(X_test,y_test)** - Esta função dá o valor da exatidão das previsões, ou seja, os valores que foram bem previstos a dividir por todos os valores.

$$Exatidão = \frac{Bem\,Classificado}{Bem\,Classificados + Mal\,Classificados} \tag{5}$$

- **precision_score(X_test,y_test)** - Esta função dá o valor da precisão das previsões nas classes, ou seja, os valores que foram bem classificados como positivo, os verdadeiros positivos(VP), a dividir pelos que foram classificados como positivos mas que foram mal classificados, os falsos positivos(FP), a somar aos verdadeiros positivos. A precisão global é a média aritmética das precisões para cada classe.

$$Precisão = \frac{VP}{VP + FP} \tag{6}$$

Na ausência de classes que existam nos dados de teste mas que não estejam presentes nos dados de treino a única consequência é que não irá existir nenhuma previsão com essa classe pois o modelo não conseguirá rotular com uma classe que não conhece.

3 Testes

3.1 $\alpha = 0$

	Breast Cancer - 1	Breast Cancer - 2	Weather Nominal
Exatidão	0.86	0.86	0.75
Precisão	0.81	0.81	0.88

3.2 $\alpha = 1$

	Breast Cancer - 1	Breast Cancer - 2	Weather Nominal
Exatidão	0.81	0.81	0.50
Precisão	0.76	0.76	0.75

3.3 $\alpha = 5$

	Breast Cancer - 1	Breast Cancer - 2	Weather Nominal
Exatidão	0.76	0.76	0.50
Precisão	0.71	0.71	0.75

4 Conclusões

Concluindo, tendo em conta que conseguimos uma precisão de 0.88 com um $alpha = 0$ nos dados *Weather Nominal* consideramos que obtivemos um resultado bastante bom, sendo que o nosso modelo conseguiria prever corretamente 88% dos momentos que eram propícios à realização de um jogo. Do ponto de vista computacional poderíamos ter elaborado um algoritmo mais eficiente, sendo que utilizámos bastantes ciclos, que têm um custo computacional superior face a outras ferramentas. Tendo em conta que trabalhámos com um *dataset* de dimensão reduzida isto não será um problema, pois o processo acontece de forma bastante rápida, no entanto num *dataset* com maior dimensão este problema revelar-se-ia num notório aumento no tempo de processamento.