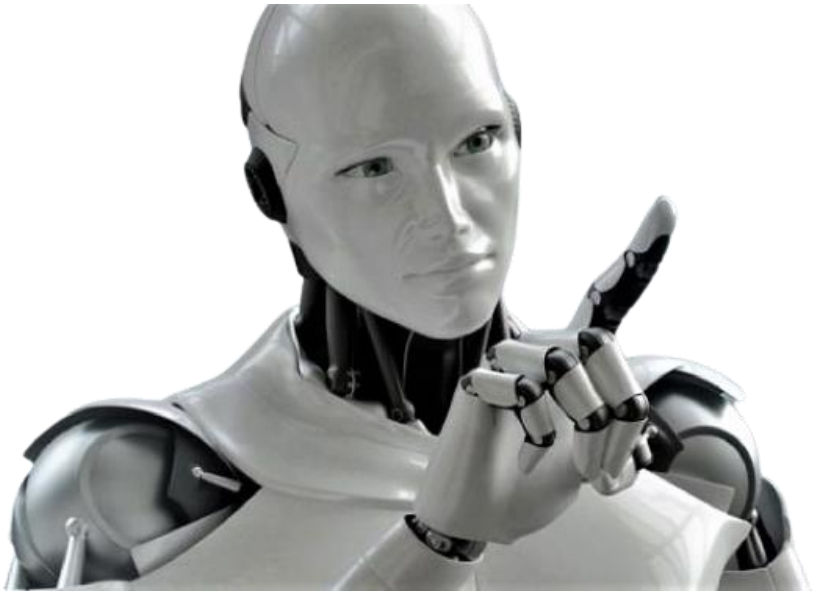
Universidade De Évora

Curso: Engenharia Informática

UC: Aprendizagem Automática

Docente: Luís Rato

Entregue Dezembro 2022



Naive Bayes UÉvora

Alunos:

Joana Carrasqueira, nº 48566

Diogo Mestre, nº 48973

João Condeço, nº 48976

Índice

Introdução 2
Abordagem 2
Análise de desempenho 4
Conclusão 4

Introdução

Este trabalho prático tem como objetivo conciliar os conhecimentos adquiridos na cadeira de Aprendizagem Automática para implementar uma classe que permita a utilização do algoritmo Naive Bayes com dados do tipo nominal, no ambiente do scikit-learn, com um estimador suavizado (*smooth estimator*), e avaliação do classificador através da exatidão e precisão.

Abordagem

Para lidar com os dados nos fornecidos foram implementadas três classes, nomeadamente, NaiveBayesUevora, Element, interElement.

⇒ **Classe NaiveBayesUevora**:

Nesta classe são declaradas todas as variáveis necessárias a garantir o funcionamento do algoritmo *Naive Bayes*, tal como nos foi proposto, sendo estas:

- alfa
- collndex – guarda o index da coluna que pretendemos prever;
- fileName – guarda o nome do ficheiro com os dados pretendidos;
- objects – corresponde ao objeto *Element* que será explicado posteriormente;
- intersections – corresponde ao objeto *interElement* que será explicado posteriormente;
- nValues – dicionário que guarda o número de classes diferentes e a sua respetiva quantidade;
- toPredic – corresponde ao nome da coluna que se pretende prever.

Para assegurar o funcionamento devido do algoritmo também foi necessário criar métodos, para além dos solicitados, nomeadamente:

- findOcorrencias – procura as ocorrências de um elemento com base no seu nome;
- findObject – procura o objeto *Element* de acordo com o seu nome;
- findInterObject – procura o objeto *interElement* de acordo com o seu nome;
- readByLine – permite ler os dados linha a linha, e armazena as previsões efetuadas;
- separate – separa a coluna que se pretende prever das restantes;

- loadFile – carrega o ficheiro pretendido.

⇒ **Classe Element**:

A classe foi criada para armazenar as informações respetivas às classes a prever, de modo a possibilitar o cálculo das probabilidades à priori das classes. Para tal, foi necessário recorrer às seguintes variáveis:

- ocurrencias
- name – nome da classe;
- total – número total de linhas da coluna da respetiva classe;
- prob – probabilidade da classe;
- totalProb – probabilidade que permite estabelecer na previsão qual a classe a que pertence;
- alfa
- nVals – número de classes diferentes na coluna da classe em questão;

Também foi necessário implementar os seguintes métodos:

- updateProb – atualiza a probabilidade a prever;
- calcProb – calcula a probabilidade a prever;
- resetProb – reinicia o valor da probabilidade;

⇒ **Classe InterElement**:

A classe foi concebida para armazenar as informações necessárias ao cálculo das probabilidades dos atributos dada a classe. Para tal, foi necessário recorrer às seguintes variáveis:

- name – nome do objeto *interElement*;
- ocurrencias
- obj – corresponde ao objeto *Element* a qual este está associado;
- objName – nome associado a *obj*;
- nValues - número de classes diferentes na coluna da classe em questão;
- motherOcurrencias – número de ocorrências da classe a qual esta está associada.

Também foi necessário implementar os seguintes métodos:

- increment – atualiza o número de ocorrências;
- getName – retorna o nome do objeto *interElement*;
- calcProb – calcula a probabilidade pretendida;

Análise de desempenho

Os resultados apresentados são referentes aos ficheiros breast-cancer-test.csv e breast.cancer-train.csv, e à última coluna dos dados.

Alfa	Exatidão	Precisão
0.0	0.524	0.367
1.0	0.667	0.567
5.0	0.667	0.567

Conclusão

Durante a elaboração deste projeto deparamo-nos com algumas dificuldades relativamente à formatação dos ficheiros, que antes de ser corrigido este problema ocorriam erros nos cálculos das probabilidades.