

Implementação do algoritmo de Naive Bayes



UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE INFORMÁTICA

Licenciatura em Engenharia Informática
Aprendizagem Automática



Ricardo Marques Oliveira (42647)

Docente: Luís Rato

Évora, 27 de julho de 2023

1 Introdução

Neste trabalho foi proposto a implementação de um algoritmo bastante utilizado na área de *Machine Learning*, o algoritmo de Naïve Bayes na linguagem Python.

O algoritmo de Naïve Bayes é um algoritmo de aprendizagem que se baseia na regra de Bayes, ou também conhecida por princípio da probabilidade condicionada, que consiste em calcular a probabilidade de um determinado evento acontecer com base num evento relacionado com o mesmo e que já tenha ocorrido. É ainda necessário que todos os atributos sejam condicionalmente independentes uns dos outros dada uma classe. Devido à sua eficiência em termos computacionais e ainda outras características torna este algoritmo bastante apelativo. [Webb, 2017]

De maneira a ser possível implementar este algoritmo de forma correta, foi necessário recorrer a duas bibliotecas disponíveis para a linguagem Python, a biblioteca NumPy e ainda a biblioteca pandas. O NumPy é uma biblioteca desenvolvida para a linguagem Python que oferece uma vasta gama de métodos e funções matemáticas com o objetivo de processar grandes quantidades de dados, como *arrays* ou matrizes com múltiplas dimensões.

2 Desenvolvimento

O trabalho consiste em dois ficheiros Python, o `main.py` e `NaiveBayesUevora.py`.

2.1 `main.py`

Este ficheiro serve essencialmente para executar todo o algoritmo, que se encontra implementado na classe `NaiveBayesUevora` e será explicado em seguida.

2.2 `NaiveBayesUevora.py`

O ficheiro `NaiveBayesUevora.py` contém a classe `NaiveBayesUevora`, onde é implementado todo o algoritmo de Naïve Bayes.

A classe `NaiveBayesUevora` possui um atributo de classe, `alpha`, que serve para guardar o valor de `alpha` do próprio algoritmo. Esta possui ainda vários métodos bastante importantes para a correta implementação e funcionamento do algoritmo, sendo estes:

- `__init__(alpha)` - é o construtor da própria classe e serve para inicializar todos os atributos necessários.
- `fit(x, y)` - gera um classificador a partir dum conjunto de treino com etiquetas.
- `predict(x)` - gera predições em função dum conjunto de dados de teste, com base no classificador previamente definido no método `fit()`.
- `accuracy_score(x, y)` - calcula a exatidão dum classificador
- `precision_score(x, y)` - calcular a precisão dum determinado classificador
- `add_unfound_property(attribute, property)` - serve para criar uma propriedade que não ocorra nos dados de treino.

2.3 Estruturas de dados

Com o objetivo de conseguir guardar os valores referentes às probabilidades dos atributos, foi utilizada uma matriz tridimensional. Foi utilizado ainda um array cuja finalidade é guardar os valores referentes às probabilidades à priori das classes.

3 Resultados obtidos

3.1 breast-cancer-train.csv e breast-cancer-test.csv

Valores de α	Precisão	Exatidão
$\alpha = 0$	35.29	57.14
$\alpha = 1$	56.67	66.67
$\alpha = 5$	56.67	66.67

3.2 breast-cancer-train2.csv e breast-cancer-test2.csv

Valores de α	Precisão	Exatidão
$\alpha = 0$	23.53	57.14
$\alpha = 1$	37.78	66.67
$\alpha = 5$	37.78	66.67

3.3 weather-nominal.csv

Para a seguinte classe = ['overcast', 'mild', 'high', 'TRUE']:

Valores de α	Previsão
$\alpha = 0$	no
$\alpha = 1$	yes
$\alpha = 5$	yes

4 Conclusão

Com a realização deste trabalho, foi possível aplicar os conhecimentos lecionados ao longo do semestre, nomeadamente implementar todo o algoritmo de Naïve Bayes como foi requerido pelo docente.

Apesar de algumas dificuldades enfrentadas ao longo da realização deste trabalho, nomeadamente o facto de nunca ter tido qualquer contacto com este algoritmo antes, concluo que o trabalho atingiu os objetivos pretendidos.

Referências

[Webb, 2017] Webb, G. I. (2017). Naïve bayes. *C Sammut and G I Webb (Eds) Encyclopedia of Machine Learning and Data Mining*, Springer.