

Problemas em Equipe 05

Estudantes:

Eduardo Eiji Goto,

Gustavo Hammerschmidt,

João Vitor Andrioli

Parte 1 – Teorema de Bayes

- 1) Ao responder um teste de múltipla escolha um estudante ou sabe a resposta ou tenta adivinhar. A probabilidade de ele saber é 0,7. Cada questão tem cinco alternativas, portanto, quando ele não sabe a resposta e tenta adivinhar a probabilidade de ele acertar é 1/5. Qual é a probabilidade que o estudante saiba a resposta dado que ele respondeu corretamente? (Observação: um estudante que sabe a resposta responde corretamente).

$$P[A | B] = 1.0$$

$$P[B] = 0.7$$

$$P[Bc] = 0.3$$

$$P[A | Bc] = 0.2$$

$$P[A] =$$

$$P[A | B] * P[B] + P[A | Bc] * P[Bc] =$$

$$1 * 0.7 + 0.2 * 0.3 =$$

$$0.76$$

$$P[B | A] =$$

$$(P[A | B] * P[B]) / (P[A]) =$$

$$(1 * 0.7) / 0.76 =$$

$$0.92$$

- 2) Em uma fábrica de enlatados, as linhas de produção I, II, e III respondem por 50, 30 e 20% da produção total. Sabendo-se que 0,4% das latas da linha de produção I são fechadas inadequadamente; 0,6% das latas da linha de produção II são fechadas inadequadamente; e 1,2% das latas da linha de produção III são fechadas inadequadamente. Calcular a probabilidades de uma lata fechada inapropriadamente provir da linha de produção I.

$$P [B1] = 0.5$$

$$P [B2] = 0.3$$

$$P [B3] = 0.2$$

$$P [A | B1] = 0.004$$

$$P [A | B2] = 0.006$$

$$P [A | B3] = 0.012$$

$$P [A] =$$

$$P [A | B1] * P [B1] + P [A | B2] * P [B2] + P [A | B3] * P [B3] =$$

$$0.002 + 0.0018 + 0.0024 =$$

$$0.0062$$

$$P [B1 | A] =$$

$$(P [A | B1] * P [B1]) / P[A] =$$

$$0.002 / 0.0062 =$$

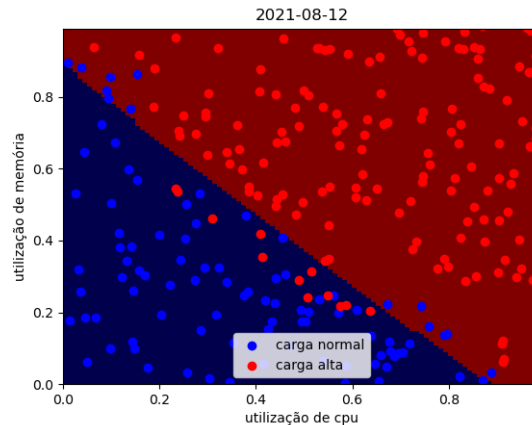
$$0.32$$

Parte 2 - Classificador Bayesiano

Uma empresa quer medir o desempenho de seus servidores para avaliar se estão sobrecarregados. Foram levantados dados do uso da CPU e de memória nos momentos de sobrecarga. Esses atributos que vão definir os pontos onde o servidor está com carga normal ou sobrecarregado. As proporções de utilização de CPU e de memória são definidas por um número entre 0 e 1, onde 0 significa 0% de utilização e 1 significa recurso com 100% de utilização.

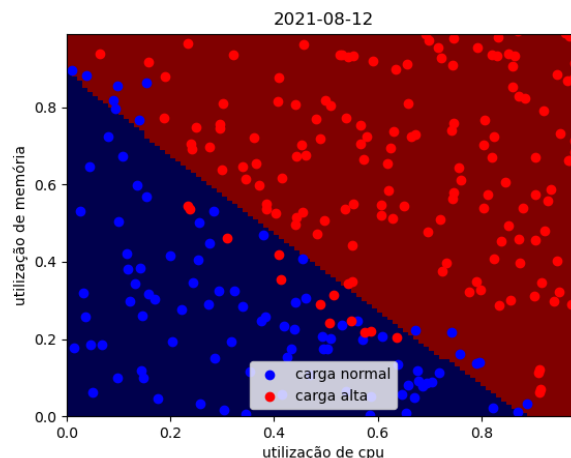
Os dados do servidor são produzidos (simulados) por sorteio aleatório pela função dadosServidor no arquivo “dados_servidor.py”. A função retorna dois conjuntos de dados com atributos e rótulos, um para a fase de treinamento e outro para a fase de teste.

A figura abaixo apresenta um conjunto de dados de treinamento com as respectivas classificações (rótulos). As coordenadas de cada ponto, que são números entre 0 e 1, foram definidas pelos dois atributos (grau de acidente e grau de inclinação). A cor de cada ponto foi definida pelo rótulo (rápido ou devagar). Pontos onde o rótulo determina que é possível andar rápido foram coloridos de azul enquanto pontos onde o rótulo determina que deve se andar devagar foram coloridos de vermelho. A figura é plotada pela função figuraTreinamento no arquivo “plotagem.py”.



O classificador bayesiano deve ser construído e treinado por comandos que serão incluídos no arquivo “classificadorNB.py”. Para treinar o classificador deve-se usar o conjunto de dados de treinamento.

O conjunto de dados de teste é usado demonstrar o funcionamento do classificador. Utilizamos o classificador bayesiano treinado para fazer a previsão dos pontos que estão no conjunto de teste. Para tornar a demonstração mais visual, a função `figuraTeste` no arquivo “`plotagem.py`” é usada para plotar os pontos do conjunto de teste em um fundo colorido que representa limite de decisão. Os pontos são plotados da mesma forma que foram plotados os pontos de treinamento. O limite de decisão é plotado usando-se uma grade de pontos que são coloridos de acordo com a previsão feita pelo classificador levando-se em conta as respectivas coordenadas. A figura abaixo mostra um exemplo, com o resultado da classificação, incluindo os pontos do conjunto de teste e o limite de decisão.



Usar a biblioteca `scikit-learn` (ou `sklearn`), uma biblioteca de machine learning na linguagem python, desenvolvida em Stanford.

Instalação:

<https://scikit-learn.org/stable/install.html>

Explicação teórica do classificador bayesiano:

https://scikit-learn.org/stable/modules/naive_bayes.html#

Manual do classificador:

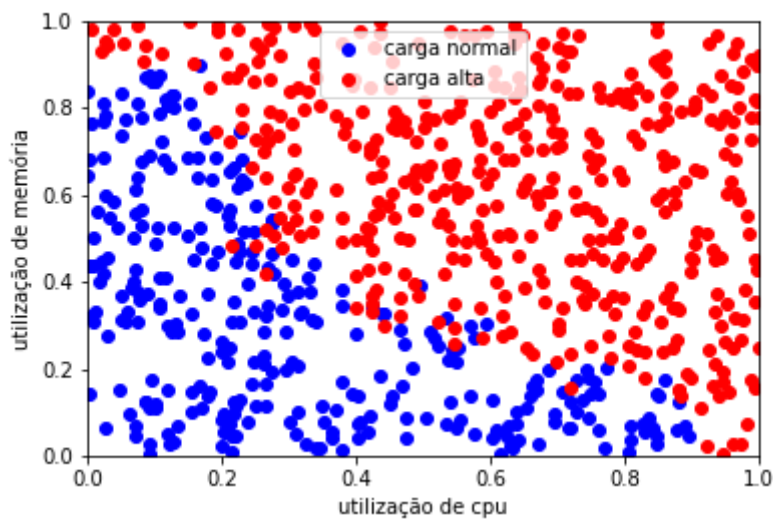
https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB

Utilizar os seguintes arquivos Python: main.py, dados_terreno, plotagem, classificadorNB. Todos os códigos possuem comentários explicando a lógica de programação. O arquivo classificadorNB tem um esqueleto de código que você deve completar, conforme explicado acima. O código no arquivo main executa o exercício chamando os outros. Você só precisa modificar o arquivo classificadorNB.

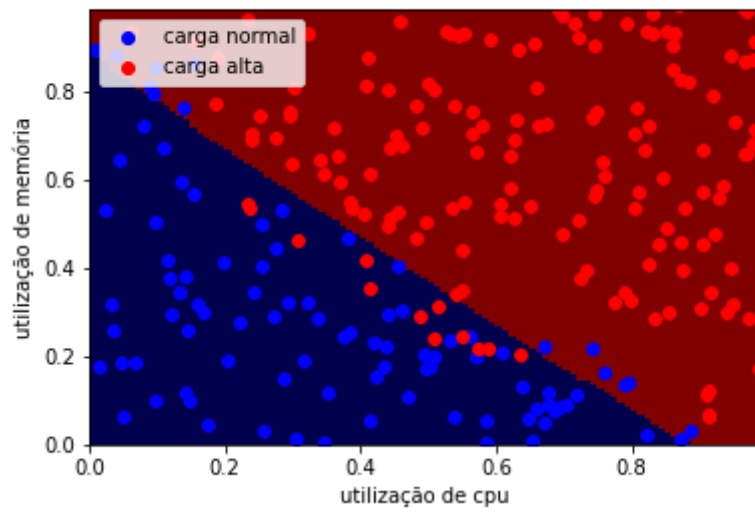
Copiar aqui o arquivo classificadorNB modificado.

```
def classifica(X_trein, Y_trein):  
  
    from sklearn.naive_bayes import GaussianNB  
  
    clf = GaussianNB()  
  
    clf.fit(X_trein, Y_trein)  
  
    return clf
```

Copiar aqui a imagem da plotagem do conjunto de treinamento



Copiar aqui a imagem da plotagem do conjunto de teste



ATENÇÃO: Observar que no título das figuras tem a data de execução do programa.