**Avaliação Somativa - Recuperação**

**Aluno: Gustavo Furini**

Suponha que você foi contrato por uma empresa que precisa de um modelo preditivo para um problema de classificação de e-mails como *spam* ou *não\_spam*.

A empresa preparou uma base de dados contendo 4601 instâncias cada uma com 58 atributos numéricos, sendo o último atributo a classe que pode ser (1) *spam* ou (0) *não\_spam*.

Descrição dos Atributos (58):

* 48 atributos reais contínuos [0,100] do tipo word\_freq\_WORD = porcentagem de palavras no e-mail que correspondem a WORD, ou seja, 100 \* (número de vezes que a PALAVRA aparece no e-mail) / número total de palavras no e-mail.
* 6 atributos reais contínuos [0,100] do tipo char\_freq\_CHAR = porcentagem de caracteres no e-mail que correspondem a CHAR, ou seja, 100 \* (número de ocorrências de CHAR) / total de caracteres no e-mail.
* 1 atributo real [1,...] contínuo do tipo capital\_run\_length\_average = comprimento médio de sequências ininterruptas de letras maiúsculas.
* 1 atributo inteiro contínuo [1,...] do tipo capital\_run\_length\_longest = comprimento da sequência ininterrupta mais longa de letras maiúsculas.
* 1 atributo inteiro contínuo [1,...] do tipo capital\_run\_length\_total = soma do comprimento de sequências ininterruptas de letras maiúsculas = número total de letras maiúsculas no e-mail.
* 1 atributo de classe nominal {0,1} do tipo spam = denota se o e-mail foi considerado spam (1) ou não (0), ou seja, e-mail comercial não solicitado.

Distribuição de Classe (quantidade de exemplos por classe):

* Spam 1813 (39,4%)
* Não Spam 2788 (60,6%)

Leitura da base:

# Carregando os dados

data=pd.read\_csv("spambase.csv", header=None)

# Transformando em Numpy

val=data.values

att=data.columns

X=val[:,0:57]

y=val[:,57]

# Mostrando o formato de X e y

print(X.shape)

print(y.shape)

**A) CONSTRUÇÃO CLASSIFICADOR**: Encontrar a melhor solução para este problema através da avaliação de soluções monolíticas (uso de um único classificador). Para tal, avalie as técnicas estudas em sala (KNN, Naive Bayes, Árvores de Decisão, SVM e MLP). Anote na tabela abaixo o melhor resultado encontrado para cada uma em termos de taxa de acerto e f1\_score. Utilize validação cruzada considerando 5 folds.

**Tabela de Resultados (Validação cruzada = 5 folds)**

|  |  |  |
| --- | --- | --- |
| **Classificador** | **Taxa de Acerto (%)** | **F1\_score** |
| **KNN** | **91,79%** | **0.89** |
| **Naive Bayes** | **81,68%** | **0.80** |
| **Árvores de Decisão** | **89,32%** | **0.87** |
| **SVM** | **93,39%** | **0.91** |
| **MLP** | **93,75%** | **0.92** |

**B) CONSIDERANDO OS RESULTADOS OBTIDOS, RESPONDA:**

B.1) Qual modelo você recomendaria considerando que se espera detectar e-mails do tipo spam, porém com poucos erros relacionados a falsa detecção.

SVM, pois obteve a maior precisão de 0.9307

B.2) Quais os parâmetros do modelo indicado para que o professor possa reproduzir o experimento?

parameters = [

{'C': [1, 10, 100], 'kernel': ['linear']},

{'C': [1, 10, 100], 'gamma': [0.1, 0.01], 'kernel': ['rbf']}

]

B.3) Apresente aqui a matriz de confusão do melhor modelo.

Gráfico, Gráfico de mapa de árvore

Descrição gerada automaticamente

B.4) Calcule a taxa de acerto por classe considerando o melhor modelo.

Spam: 0,92%

Não\_spam: 0,95%

B.5) Para este problema você recomenda o uso de taxa de acerto ou f1\_score como métrica mais adequada. Justifique a sua resposta.

F1 score seria uma métrica mais adequada, pois leva em consideração tanto a precisão quanto a revocação, permitindo uma avaliação mais completa do desempenho do modelo na classificação correta de spam, mesmo com o desequilíbrio das classes.