Instituto Superior Técnico

**Departamento de Engenharia Electrotécnica e de Computadores**

# Machine Learning

$6^{th}$ Lab Assignment

Evaluation and Generalization

Report

Shift: Fri. 17:00h Group Number: 7
Number: 83916 Name: João Manuel Prata Morais
Number: 97236 Name: Francisco Rabaça Moller Freiria

# Introdução

Este relatório foi feito no âmbito da unidade curricular Aprendizagem Automática e tinha como principal objetivo aplicar os conhecimentos obtidos durante a parte laboratorial, corelacionando estes com os conhecimentos obtidos na componente teórica da Unidade curricular.

Assim, foram-nos dado dois conjuntos de dados, dataset's, para que criássemos um modelo de treino que melhor generalizasse os dados. Para tal recorremos aos modelos de treino estudados nas aulas anteriores.

De forma a obtermos resultados mais conclusivos, foi pedido que para cada conjunto de dados utilizássemos dois classificadores diferentes para possibilitar uma comparação de resultados e assim perceber melhor as características de cada classificador que o tornam melhor classificador. Para avaliar o classificador utilizamos como medidas de performance a accuracy, precision, sensitivity e specificity. Para calcular as medidas de performance atrás referenciadas, recorreu-se ao conceito de confusion matrix.

# 1.1. Confusion matrix

Como referenciado anteriormente, a confusion matrix foi a base para calcular as medidas de performance do classificador. Apresentamos na tabela 1 a confusion matrix genérica que permite perceber os resultados apresentados pela mesma.

*Tabela 1: Confusion matrix genérica*

| | ***Actual*** | | |
|---|---|---|---|
| ***Predicted*** | | **Positive - 1** | **Negative - 0** |
| | **Positive - 1** | True Positive | False Positive |
| | **Negative - 0** | False Negative | True Negative |

**- True Positive** – Classe real do ponto atual é 1 e o classificador atribuí classe 1 também.

**- True Negative** – Classe real do ponto atual é 0 e o classificador atribuí classe 0 também.

**- False Positive** – Classe real do ponto atual é 0 e o classificador atribuí classe 1.

**- False Negative** – Classe real do ponto atual é 1 e o classificador atribuí classe 0.

As medidas de performance utilizadas foram:

- $Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$
- $Precision = \frac{TP}{TP+FP}$
- $Sensitivity = \frac{TP}{TP+FN}$
- $Specificity = \frac{TN}{FP+TN}$

# 1.2. Support Vector Machine

O SVM é um conceito que através de métodos de aprendizagem analisam os dados e reconhecem padrões permitindo assim classificá-los. O SVM padrão recebe um dataset e prediz, para cada entrada dada, qual de duas possíveis classes de entrada faz parte. Podemos então afirmar que o SVM é um classificador linear binário não probabilístico.

Numa linguagem mais informal, o que o SVM faz é encontrar uma linha, denominada de hyperplane entre dados de duas classes. Faz a otimização de forma a maximizar a distância entre os pontos mais próximos em relação a cada uma das classes. Este distância é também chamada de margem.

Utilizamos este classificador por apresentar características como por exemplo quando é apresentado um valor inconsistente, ou seja, um outlier, o SVM procura a melhor forma possível de classificação e pode ainda desconsiderar esse valor.

Para os nossos conjuntos de dados utilizamos o SVC com a função Kernel polinomial $(\gamma(x, x') + r)^d$. De forma a optimizar os resultados foram feitos testes, para os diferentes graus do polinómio (d), e pela avaliação do erro e da accuracy do classificador concluímos que para os nossos datasets obtivemos valores diferentes. No primeiro dataset utilizamos o polinómio de Grau 1 e no segundo dataset o polinómio de grau 4.

Na tabela2 são apresentados os resultados das medidas de performance do classificador para os dois dataset's

*Tabela 2: Resultados obtidos para o classificador SVM*

| **DATASET** | **1** | **2** |
|---|---|---|
| **d** | **1** | **4** |
| ***Classificador*** | *Suport Vector Machine* (SVM) | *Suport Vector Machine* (SVM) |
| *Accuracy* | 0.765 | 0.838 |
| *Precision* | 0.870 | 0.774 |
| *Sensitivity* | 0.7014 | 0.727 |
| *Specificity* | 0.854 | 0.894 |

Podemos observar que o SVM foi uma boa escolha pois este classificador apresenta bons resultados em ambos os datasets. Para o dataset1 concluimos que, ainda que não tivessemos obtido maus resultados, poderiamos ter criado um dataset de validação.

Ainda assim, consideramos que para a obtenção de melhores resultados poderiamos ter testado os nossos datasets para outras expressões da função de Kernel.

# 1.3. Naïve Bayes

O classificador de Bayes simplifica o problema em que há muitas *features* assumindo que estas são condicionalmente independentes.

$$p(x_1, \ldots, x_p|\omega_k) = \prod_{i=1}^{p} p(x_i|x_1, \ldots, x_{i-1}, \omega_k) = \prod_{i=1}^{p} p(x_i|\omega_k)$$

Isto significa que apenas é necessário estimar a distribuição condicional de cada *feature*. No problema de reconhecimento digital, significa que temos de estimar a distribuição condicional de cada pixel, tornando-se numa tarefa simples.

Os resultados obtidos com este classificador estão representados na tabela 3.

*Tabela 3: Resultados obtidos para o classificador de Bayes*

| **Dataset** | **1** | **2** |
|---|---|---|
| **Accuracy** | 0.513 | 0.758 |
| **Precision** | 0.481 | 0.774 |
| **Sensitivity** | 0.481 | 0.585 |
| **Specificity** | 0.541 | 0.879 |

Podemos observar que para o dataset1, o classificador de Bayes não foi uma boa escolha pois os resultados não são satisfatórios, apresentando este uma accuracy de apenas 51,3%, o que o torna num mau classificador.

Já para o dataset2, os resultados obtidos foram já bastante melhores, apresentando uma accuracy de 75,8%.

Quanto aos outros avaliadores de performance, estes foram bastante satisfatórios no dataset2, apresentando uma Precision de 77.4% e uma Specificity de 87.9%, valores bastante razoáveis para os resultados pretendidos. Já no dataset1, também os outros avaliadores de performance além da Accuracy, ficaram muito aquém do pretendido, apresentando uma Precision e uma Sensitivity abaixo dos 50%. Assim conclui-se que para os dados do dataset1, o Classificador de Bayes não foi de todo uma boa escolha, mas para o segundo dataset foi um classificador bastante satisfatório para os resultados pretendidos.

# 2. Conclusões

Analisando os resultados obtidos para ambos os datasets e para os dois classificadores usados, conclui-se que:

- No classificador SVM foi usada a função polinomial e foram conseguidos bons resultados. Foram também testados vários graus do polinómio de Kernel como medida de otimização. O melhor valor conseguido para o grau do polinómio foi o polinómio de primeiro grau para o datatset1 e o polinómio de grau 4 para o dataset2. Foi tomado também em consideração a avaliação do score e consequentemente do erro;

- No que toca ao parâmetro Precision, os classificadores Naive Bayes e SVM apresentam valores semelhantes no dataset2;

- O SVM apresenta-se como o classificador com mais sensitivity;

- O classificador SVM é o que apresenta maior specificity;

- Em termos de Accuracy, Sensitivity e Specificity, quando comparando os resultados dos dois datasets, concluímos que o classificador SVM é o que apresenta melhores resultados;

- Para o dataset1 podemos que o Naïve Bayes não é um bom classificador apresentando uma accuracy de apenas 51%.

Face ao acima exposto, podemos concluir que os diferentes classificadores apresentam diferentes características. Ainda assim, o classificador Support Vectors Machines (SVM) apresenta-se como o classificador mais consistente para os dois datasets estudados.