**Relatório Projeto 2 Inteligência Artificial 2017/2018**

**P1 - Métodos de Classificação**

Features:

1. Soma do valor ASCII das letras
2. Número de vogais
3. Número de Consoantes
4. Número de letras Acentuadas
5. Tamanho da Palavra

Escolha de Classificador:

Escolhemos o classificador DecisionTreeClassifier com um percentagem de erro de 5%, pois dentro dos classificadores possíveis obtivemos as seguintes percentagens de erro (média dos 2 testes):

DecisionTreeClassifier: 5%

KNN(n=1): 6%

KNN(n=2): 11%

KNN(n=3): 12%

LogisticRegression: 23%

Escolha de Parâmetros:

O ajuste dos parâmetros da DecisionTree não levou a nenhuma melhoria em termos da percentagem de erro, pelo que se deixou os parâmetros com o seu valor ‘default’, o que foi suficiente para termos uma percentagem de erro de 5%.

**P2 - Métodos de Regressão**

Método 1 - KernelRidge:

Após correr um script que usa a validação cruzada, testando os parâmetros mais importantes deste método (kernel, alpha e gamma), optou-se por escolher os seguintes parâmetros, pois levaram a um menor erro total(T1 e T2 correspondem ao erro quadratico médio no teste um e no teste dois, respectivamente e Tot corresponde a soma desses dois erros, este é o valor que pertendemos minimizar):

**[K: rbf] [GAMMA: 0.05] [ALPHA: 1e-05] [T1: 0.2802] [T2: 139.1139] [Tot: 139.3941]**

Método 2 - SVR (Epsilon-Support Vector Regression):

Após correr um script similar ao anterior, desta vez com os parâmetros ‘epsilon’, ‘C’ e ‘gamma’ (o ‘kernel’ não foi verificado pois no método anterior já se verificava que o melhor ‘kernel’ para estes problemas seria o rbf), escolheu-se o seguinte resultado, pela mesma razão do método anterior (T1 e T2 correspondem ao erro quadratico médio no teste um e no teste dois, respectivamente e Tot corresponde a soma desses dois erros, este é o valor que pertendemos minimizar):

**[Epsilon: 0.05][Gamma: 0.05][C: 8000][T1: 0.2486][T2: 172.3201][Tot: 172.5687]**

Conclui-se então que a melhor regressão para este problema é a ‘KernelRidge’, que tem um erro total associado aos dois testes bastante inferior à SVR

**P3 - Aprendizagem por Reforço**

Escolha dos parâmetros:

traces2Q(): O erro (err = 1e-5) a partir do qual a diferença entre Q’s sucessivos passa a ser irrelevante, e o algoritmo acaba, foi escolhido como suficiente pois erros inferiores não levam a um “erro no Q” menor, sendo isto testado pela expressão usada para avaliar o Q “np.linalg.norm(Q-fmdp.Q).

Q2pol(): Com eta = 5 a politica levou o agente a tomar as decisões corretas, cumprindo as requisitos impostos pelo problema.

Interpretação dos resultados:

Ambiente 1: Este ambiente pode se representar por uma “linha reta”. O estado 1, encontra-se à direita do estado 0, o 2 à direita do 1, e assim sucessivamente. A ação 0 significa deslocar-se para o estado à direita (0->1) e a ação 1 o contrário (para a esquerda). A ação 1 no estado 0 leva-o para o mesmo estado, assim como a ação 0 no estado 6 o leva para ele próprio. A ação 0 no estado 5 pode ou levar para o 6 ou para si próprio.

1

0

0

0

0

0

0

0

0 1 2 3 4 5 6

0

1

1

1

1

1

1

Ambiente 2: Este ambiente representa praticamente o mesmo do ambiente 1. As diferenças encontram-se nas probabilidades associadas à ação 0 do estado 5, e na ação 0 do estado 6, que desloca o agente para o estado 1

0

1

0

0

0

0

0

0

0 1 2 3 4 5 6

1

0

1

1

1

1

1

Função Recompensa para os dois Ambientes (estado, ação):

R(0,0) = 1 R(6,0) = 1

R(0,1) = 1 R(6,1) = 1

R(x,0) = 0 R(x,1) = 0, x € {2,3,4,5}

Movimento do agente: Tanto num ambiente como no outro existem ciclos em que o agente pode ficar continuamente a realizar a mesma ação, recebendo sempre recompensa. O agente move-se para atingir este ciclo. No ambiente 1 o agente pode atingir este “ciclo de recompensa” nos estados 0 e 6, tomando as ações que o levam para o estado mais perto(se começar em 4 irá para 6 através de ações 0). Já no ambiente 2, o único ciclo encontra-se em 0.