



# PRACTICA 1 - APRENDIZAJE AUTOMÁTICO

Claudia Castelo Sagnotti - 100363633

Víctor Gómez de la Camara - 100363723

Índice

**Introducción** .....2

**Fase 1** .....2

**Fase 2** .....2

**Fase 3** .....6

**Fase 4** .....8

**Preguntas** .....8

**Conclusión**.....10

## Introducción

En este documento se recogerán las distintas explicaciones que se nos pide en las distintas fases de la práctica. También se recogerán las respuestas y explicaciones a las preguntas del apartado 7.

## Fase 1

La extracción de las características es muy simple hemos metido en distintas variables string el resultado de las distintas funciones para sacarlos distintos atributos utilizando el `str()` para transformarlos en tipo string y después los concatenamos todos en una variable string llamada cadena que escribimos con `write()` en el fichero correspondiente (abrimos el fichero con el comando `open()` y el flag `a` para que se cree si no existe y se abra y escriba en el `a` continuación si ya existe).

Para poner el `next_score` lo que hacemos es que en lugar de escribir desde el principio esperamos a la siguiente e imprimimos en el fichero los datos de la fase anterior con el score de la siguiente a la anterior hasta el final y cuando recogemos que todos los fantasmas están muertos imprimimos la fase en la que mueren con el mismo `score_next` que `score`.

Para los ficheros hemos usado varios mapas:

- Keyboard: `oneHunt`, `20Hunt`, `minimaxClassic`, `trickyClassic`, `sixHunt` y `smallClassic`.
- Tutorial: `mimapa`, `minimaxClassic`, `smallHunt`, `openClassic`, `testClassic`, `oneHunt`.

## Fase 2

En esta fase se nos pide generar un clasificador para los ficheros de weka que hemos creado en la fase 1. Los clasificadores usados son los mismos que en el tutorial 3 mas algunos vistos en clase que son K-estrella, Uno-R y RandomForest. Para ello hemos creado unos ficheros auxiliares que engloban el entrenamiento y los test para poder usar el `experimenter` con porcentajes (los ficheros son llamados con un `_aux`).

La primera prueba es con los ficheros normales sin cambiar nada. Los resultados son los siguientes:

- Keyboard
  - Mismos mapas

Clasificadores	resultados
NaiveBayes	47.93
IBK1	52.07
IBK3	54.55
IBK5	56.20
K-Star	57.02
OneR	37.19
PART	62.81
ZeroR	19.01
J48	60.33
RandomForest	71.90

- Distintos mapas

Clasificadores	resultados
NaiveBayes	47.46
IBK1	36.44
IBK3	38.98
IBK5	38.98
K-Star	36.44
OneR	40.68
PART	39.83
ZeroR	8.47
J48	43.22
RandomForest	52.54

- Tutorial
  - Mismos mapas

Clasificadores	resultados
NaiveBayes	33.81
IBK1	51.08
IBK3	33.09
IBK5	32.37
K-Star	37.41
OneR	33.09
PART	48.20
ZeroR	31.65
J48	51.80
RandomForest	52.52

- Distintos mapas

Clasificadores	resultados
NaiveBayes	31.53
IBK1	24.32
IBK3	21.62
IBK5	21.62
K-Star	16.22
OneR	27.03
PART	26.13
ZeroR	33.33
J48	27.93
RandomForest	32.43

La segunda prueba es con el fichero discretizado. Los resultados son:

- Keyboard
  - Mismos mapas

Clasificadores	resultados
NaiveBayes	48.76
IBK1	52.07
IBK3	55.37
IBK5	56.20
K-Star	54.55
OneR	33.06
PART	47.93
ZeroR	19.01
J48	54.55
RandomForest	57.85
Id3	29.75

- Distintos mapas

Clasificadores	resultados
NaiveBayes	29.66
IBK1	40.68
IBK3	43.22
IBK5	42.37
K-Star	39.83
OneR	19.49
PART	38.14
ZeroR	8.47
J48	42.37
RandomForest	37.29
Id3	24.58

- Tutorial
  - Mismos mapas

Clasificadores	resultados
NaiveBayes	36.69
IBK1	38.85
IBK3	37.41
IBK5	36.69
K-Star	38.13
OneR	37.41
PART	39.57
ZeroR	31.65
J48	47.48
RandomForest	46.76
Id3	42.45

- Distintos mapas

Clasificadores	resultados
NaiveBayes	18.02
IBK1	21.62
IBK3	18.02
IBK5	20.72
K-Star	21.62
OneR	34.23
PART	33.33
ZeroR	33.33
J48	12.61
RandomForest	29.73
Id3	20.72

La tercera prueba es con el fichero una vez quitandos los atributos relacionados con la comida (puntos de comida restante y distancia a la comida) y el atributo correspondiente a next\_score. Los resultados son:

- Keyboard
  - Mismos mapas

Clasificadores	resultados
NaiveBayes	49.59
IBK1	50.41
IBK3	53.72
IBK5	53.72
K-Star	59.50
OneR	37.19
PART	57.02
ZeroR	19.01
J48	58.68
RandomForest	66.12

- Distintos mapas

Clasificadores	resultados
NaiveBayes	42.37
IBK1	32.20
IBK3	33.05
IBK5	38.14
K-Star	37.29
OneR	40.68
PART	40.68
ZeroR	8.47
J48	43.22
RandomForest	45.76

- Tutorial
  - Mismos mapas

Clasificadores	resultados
NaiveBayes	35.97
IBK1	48.92
IBK3	30.22
IBK5	34.53
K-Star	38.85
OneR	33.09
PART	42.45
ZeroR	31.65
J48	46.04
RandomForest	58.27

- Distintos mapas

Clasificadores	resultados
NaiveBayes	31.53
IBK1	25.23
IBK3	23.42
IBK5	22.52
K-Star	19.82
OneR	27.03
PART	26.13
ZeroR	33.33
J48	27.93
RandomForest	29.73

Como vemos en los mejores resultados (subrayados en amarillo en cada tabla), el mejor resultado es en el RandomForest de Keyboard con el test de los mismos mapas que es un 71.9.

## Fase 3

Para la fase 3 nos piden usar weka para generar un predictor de pacman para ello usamos los predictores M5Rules y M5P.

Las pruebas las hemos realizado en el explorer y los resultados son los siguientes:

- Con los ficheros normales.
  - Keyboard
    - Mismos mapas

Predictor	Resultado
M5P	0.9952
M5Rules	0.9946

- Distintos mapas

Predictor	Resultado
M5P	0.9911
M5Rules	0.9911

- Tutorial
  - Mismos mapas

Predictor	Resultado
M5P	0.9675
M5Rules	0.9675

- Distintos mapas

Predictor	Resultado
M5P	0.9711
M5Rules	0.9633

- Con los ficheros sin atributos relacionados con la comida (distancia a la comida y puntos de comida restantes)
  - Keyboard
    - Mismos mapas

Predictor	Resultado
M5P	0.9961
M5Rules	0.9961

- Distintos mapas

Predictor	Resultado
M5P	0.9965
M5Rules	0.9965

- Tutorial
  - Mismos mapas

Predictor	Resultado
M5P	0.9996
M5Rules	0.9996

- Distintos mapas

Predictor	Resultado
M5P	0.999
M5Rules	0.9989



## Fase 4

El modelo que hemos elegido para construir el agente automático es el clasificador RandomForest con los ficheros de keyboard de entrenamiento y de test con los mismos mapas. Lo hemos elegido debido a que ha sido el que mejores resultados a tenido de la fase dos y además es un resultado con un fichero al que no hemos sometido a ningún tipo de preprocesado por lo que no se necesita cambiar nada.

También hemos elegido el RandomForest pero con los ficheros sin tributos de comida con los mismos mapas y de keyboard. La razón es que su resultado es el segundo mas alto de todos los que hemos conseguido.

El ultimo también es RandomForest pero esta vez con ficheros de tutorial y samemaps sin atributos de comida. Lo hemos elegido debido a que su resultado es el tercero más alto.

El funcionamiento de la fase 4 es sencillo, simplemente usamos las distintas funciones de get que hay en el código para obtener todos los datos necesarios que le pasamos a weka y los metemos en un array que le pasamos a weka para que monte el predictor y recibimos un número que es la posición del movimiento predicho en la lista de los movimientos. Hacemos un bucle para encontrar el movimiento en la lista y lo devolvemos.

Para usar todos los ficheros sin confundirnos hemos llamado a los ficheros weka correctos Training\_Set, Training\_Set2 y Training\_Set3 y a los modelos RandomForest, RandomForest2 y RandomForest3 según su resultado.

## Preguntas

### **1. ¿Qué diferencias hay a la hora de aprender esos modelos con instancias provenientes de un agente controlado por un humano y uno automático?**

Al menos en nuestro caso el agente automático a recibido unos resultados bastante peores comparando con el agente controlado por un humano por lo que a la hora de aprender los modelos aprendían bastante mal y fallaban la mayoría de las predicciones de los test correspondientes.

### **2. Si quisieras transformar la tarea de regresión en clasificación ¿Que tendrías que hacer? ¿Cuál crees que podría ser la aplicación práctica de predecir la puntuación?**

Como la regresión se utiliza para predecir primero tendríamos que quitar los atributos que sirven para ello. Después, si fuese necesario, pondríamos atributos y cambiaríamos la clase.

La predicción de la puntuación puede servir para que pacman economice, es decir sea más eficiente, la puntuación al ir a por los fantasmas y moverse por el tablero de manera que consiga una buena puntuación.

### **3. ¿Qué ventajas puede aportar predecir la puntuación respecto a la clasificación de la acción? Justifica tu respuesta.**

Puede aportar el que pacman consiga una mejor puntuación, ya que si pacman predice solo el movimiento puede que el coste de ese movimiento no valga la pena. Si se pudiese predecir la puntuación del movimiento se podrían evitar los movimientos innecesarios.

### **4. ¿Crees que se podría conseguir alguna mejora en la clasificación incorporando un atributo que indicase si la puntuación en el instante actual ha descendido o ha bajado?**

Si, permitiría optimizar los movimientos de pacman, mejorando mucho su movimiento y rendimiento, pudiéndose obtener una mayor puntuación y evitar los movimientos innecesarios que pueden surgir en el escenario descrito en la pregunta tres.

## Conclusión

En esta practica los mayores problemas que hemos tenido han sido debido a una mala interpretación del enunciado, debido al cual generamos unos ficheros de weka con pacman que eran erróneos y tuvimos que repetir toda la fase 1, y también hemos tenido problemas al instalar los programas de la fase 4.

Lo mas destacable de esta práctica es que a pesar de que hemos usado todos los conocimientos adquiridos en los distintos tutoriales que hemos hecho desde el principio de curso, ha sido una práctica que conlleva un esfuerzo mayor al que se esperaba y algo desmesurado desde nuestro punto de vista.