

Aprendizaje automático



Práctica 1

Grupo 84

| | | |
|----------------------|-----------|--|
| Sabrina Riesgo Reyes | 100363834 | 100363834@alumnos.uc3m.es |
| Laura Yunta García | 100363785 | 100363785@alumnos.uc3m.es |

Índice

| | |
|---|----|
| 1. Introducción..... | 4 |
| 2. Fase 1: Recogida de información..... | 5 |
| 3. Fase 2: Clasificación | 7 |
| 3.1. Resultados tutorial 1..... | 7 |
| 3.1.1. Ficheros 1: prueba 1..... | 8 |
| 3.1.2. Ficheros 1: prueba 2 | 9 |
| 3.1.3. Ficheros 1: prueba 3 | 10 |
| 3.1.4. Ficheros 1: prueba 4 | 11 |
| 3.1.5. Ficheros 2: prueba 1 | 12 |
| 3.1.6. Ficheros 2: prueba 2..... | 13 |
| 3.1.7. Ficheros 2: prueba 3 | 14 |
| 3.1.8. Ficheros 2: prueba 4..... | 15 |
| 3.2. Resultados keyboard..... | 16 |
| 3.2.1. Ficheros 1: prueba 1..... | 17 |
| 3.2.2. Ficheros 1: prueba 2 | 18 |
| 3.2.3. Ficheros 1: prueba 3 | 19 |
| 3.2.4. Ficheros 2: prueba 1 | 20 |
| 3.2.5. Ficheros 2: prueba 2..... | 21 |
| 3.2.6. Ficheros 2: prueba 3 | 22 |
| 3.3. Análisis final..... | 23 |
| 4. Fase 3: Regresión | 24 |
| 4.1. Resultados de tutorial 1. | 24 |
| 4.1.1. Prueba 1..... | 24 |
| 4.1.2. Prueba 2. | 26 |
| 4.1.3. Prueba 3. | 27 |
| 4.1.4. Prueba 4..... | 27 |
| 4.2. Resultados Keyboard. | 28 |
| 4.2.1. Ficheros 1: Prueba 1. | 29 |
| 4.2.2. Ficheros 1: Prueba 2..... | 30 |
| 4.2.3. Ficheros 1: Prueba 3..... | 31 |
| 4.2.4. Ficheros 1: Prueba 4..... | 32 |
| 4.2.5. Ficheros 2: Prueba 1..... | 33 |

| | | |
|--------|--|----|
| 4.2.6. | Ficheros 2: Prueba 2..... | 34 |
| 4.2.7. | Ficheros 2: Prueba 3..... | 35 |
| 4.2.8. | Ficheros 2: Prueba 4. | 36 |
| 4.3. | Análisis final..... | 37 |
| 5. | Fase 4: Construcción de un Agente Automático | 39 |
| 6. | Preguntas | 40 |
| 7. | Conclusiones y dificultades | 42 |

1. Introducción

El desarrollo de la práctica consiste en la aplicación de técnicas de Aprendizaje Automático para llevar a cabo tareas de predicción y clasificación. Para el desarrollo de la práctica se utilizará el simulador proporcionado que permite controlar a Pac – Man de manera automática y recoger la información del entorno.

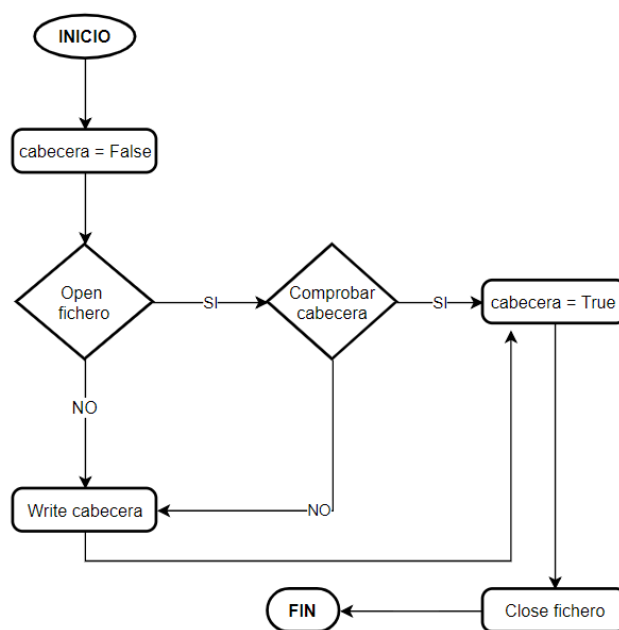
Finalmente, el documento y con ello la práctica se dividirá en siete partes diferenciadas:

- **Introducción:** En este apartado se comenta brevemente el objetivo a alcanzar con el desarrollo del tutorial y se explica la estructura y contenido de cada uno de los apartados del documento.
- **Fase 1:** En esta fase se modificarán las funciones de extracción de características para que se creen ficheros de información legibles por Weka.
- **Fase 2:** En esta fase se construirá el modelo de clasificación que permitirá a Pac – Man moverse hacia los fantasmas para comérselos.
- **Fase 3:** En esta fase se explorarán los algoritmos de predicción de Weka utilizando la experimentación y construir un modelo de predicción.
- **Fase 4:** En la cuarta fase se busca como objetivo construir un agente automático para que tome la decisión de qué acción ejecutar en cada instante.
- **Respuestas a las preguntas del documento:** En este apartado se responderán a cada una de las preguntas especificadas en el documento del tutorial.
- **Conclusiones y problemas encontrados:** En este apartado se comentarán las conclusiones sacadas con la realización del ejercicio, además los diversos problemas que se han conseguido solventar durante el transcurso de la práctica.

2. Fase 1: Recogida de información

En esta primera fase de la práctica se va a proceder a la extracción de las características de la ejecución del Pac – Man en un fichero legible por Weka, de esta forma la función de extracción servirá como base para la realización de la práctica. Una vez sacados los respectivos ficheros solicitados se adjuntarán en la entrega y se utilizarán para las siguientes fases.

Para la realización de fase 1 se ha reutilizado la función de extracción de características utilizada en el tutorial 1. Se han utilizado funciones pertenecientes al código para atributos de posición y distancias, mientras que para aquellos atributos relacionados con las paredes se ha tenido que elaborar un poco más el código. Una vez almacenados todos los atributos en sus variables correspondientes, se comprueba si el fichero existe o no y si está en el formato legible por weka siguiendo el siguiente diagrama:



A continuación se procederá a generar las instancias que contendrá el fichero. Siguiendo los atributos que se han tenido en cuenta para el desarrollo de la práctica, es necesario almacenar en una instancia todos los atributos de la misma, la acción anterior y la puntuación siguiente. En primer lugar, se almacena la acción de la primera instancia generada. En el siguiente tick se almacena la instancia completa, y en próximo tick se imprime la instancia del tick anterior, la acción primera y la puntuación actual. De este modo, se obtendría una instancia completa (con todos los atributos de ese tick), la acción anterior y la puntuación siguiente.

Finalmente se mostrará un listado con todos los atributos contemplados para facilitar el uso de estos en las fases posteriores.

1. **columna_pacman:** número de la columna en la que se encuentra el pacman.
2. **fila_pacman:** número de la fila en la que se encuentra el pacman.
3. **num_fantasmas:** número total de fantasmas (es nuestro caso siempre sería 4).
4. **num_fantasmas_vivos:** número de fantasmas que aún están vivos.
5. **columna_fantasma_1:** número de la columna en la que se encuentra el fantasma 1.

6. **fila_fantasma_1:** número de la fila en la que se encuentra el fantasma 1.
7. **columna_fantasma_2:** número de la columna en la que se encuentra el fantasma 2.
8. **fila_fantasma_2:** número de la fila en la que se encuentra el fantasma 2.
9. **columna_fantasma_3:** número de la columna en la que se encuentra el fantasma 3.
10. **fila_fantasma_3:** número de la fila en la que se encuentra el fantasma 3.
11. **columna_fantasma_4:** número de la columna en la que se encuentra el fantasma 4.
12. **fila_fantasma_4:** número de la fila en la que se encuentra el fantasma 4.
13. **dist_1:** distancia de Manhattan entre la posición actual del pacman y la posición actual del fantasma 1.
14. **dist_2:** distancia de Manhattan entre la posición actual del pacman y la posición actual del fantasma 2.
15. **dist_3:** distancia de Manhattan entre la posición actual del pacman y la posición actual del fantasma 3.
16. **dist_4:** distancia de Manhattan entre la posición actual del pacman y la posición actual del fantasma 4.
17. **columna_fantasma_mas_cercano:** número de la columna en la que se encuentra el fantasma más cercano.
18. **fila_fantasma_mas_cercano:** número de la fila en la que se encuentra el fantasma más cercano.
19. **dist_columna_fantasma_cercano:** distancia en columnas entre la posición actual del pacman y la posición actual del fantasma más cercano.
20. **dist_fila_fantasma_cercano:** distancia en filas entre la posición actual del pacman y la posición actual del fantasma más cercano.
21. **dist_fantasma_cercano:** distancia de Manhattan entre la posición actual del pacman y la posición actual del fantasma más cercano (suma de los dos atributos anteriores).
22. **num_comida:** número de puntos de comida.
23. **dist_comida:** distancia al punto de comida más cercano.
24. **isWallNorth:** booleano que toma el valor "True" si justo al norte de la posición actual del pacman hay una pared, y toma el valor "False" en caso contrario.
25. **isWallSouth:** booleano que toma el valor "True" si justo al sur de la posición actual del pacman hay una pared, y toma el valor "False" en caso contrario.
26. **isWallEast:** booleano que toma el valor "True" si justo al este de la posición actual del pacman hay una pared, y toma el valor "False" en caso contrario.
27. **isWallWest:** booleano que toma el valor "True" si justo al oeste de la posición actual del pacman hay una pared, y toma el valor "False" en caso contrario.
28. **distNorth:** distancia entre la posición actual del pacman y la pared más cercana al norte.
29. **distSouth:** distancia entre la posición actual del pacman y la pared más cercana al sur.
30. **distEast:** distancia entre la posición actual del pacman y la pared más cercana al este.
31. **distWest:** distancia entre la posición actual del pacman y la pared más cercana al oeste.
32. **score:** puntuación actual.
33. **dir_anterior_pacman:** dirección anterior ejecutada.
34. **dir_pacman:** dirección a ejecutar.
35. **next_score:** siguiente puntuación.

3. Fase 2: Clasificación

3.1. Resultados tutorial 1

Para la realización de este apartado se han tenido en cuenta los ficheros generados en la fase 1 utilizando el agente programado en el tutorial1. Con el objetivo de abarcar el mayor número de posibilidades para estudiar correctamente los resultados, además de utilizar los ficheros generados en la fase 1, se han creado otros ficheros que contienen un mayor número de instancias, aproximadamente el doble. A continuación se mostrarán los resultados obtenidos en cada una de las pruebas realizadas indicando “Ficheros” y “Prueba”.

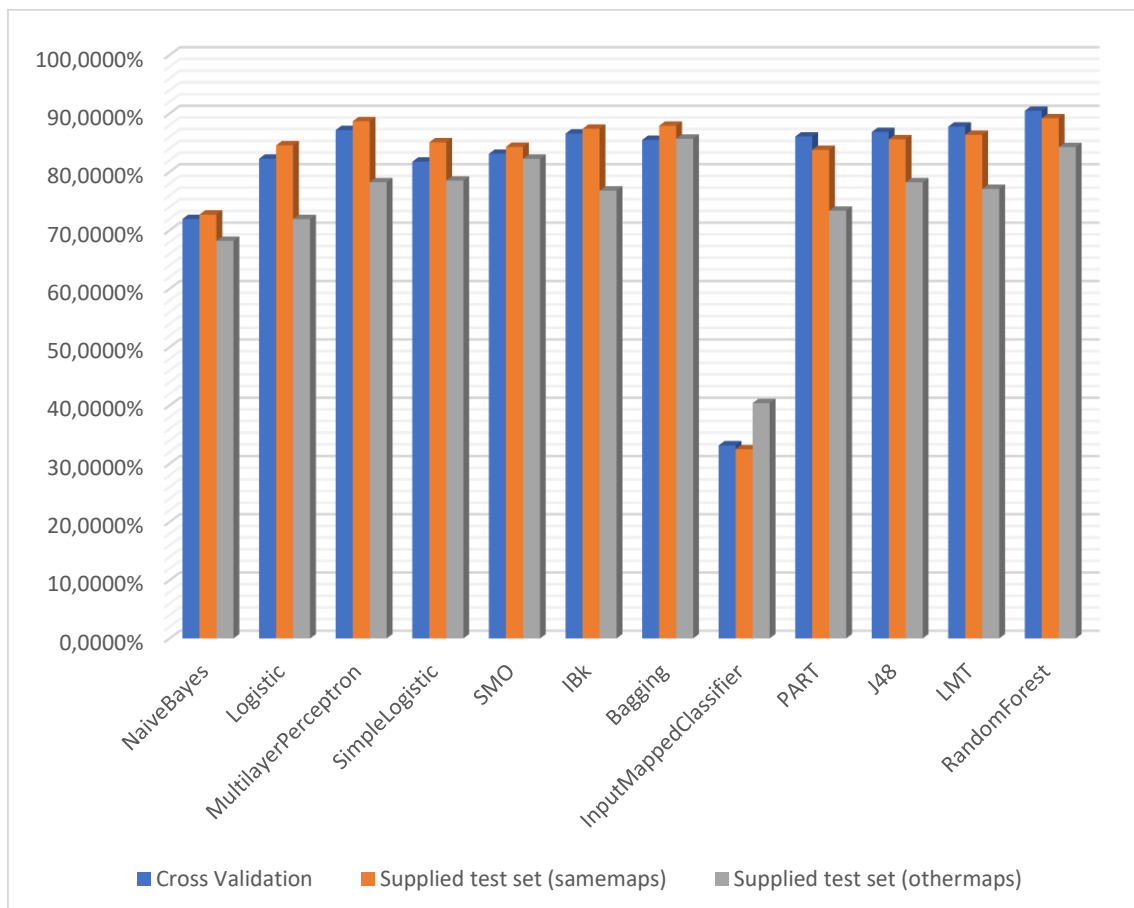
- “Ficheros 1” (instancias de la fase 1).
- “Ficheros 2” (instancias generadas posteriormente con el doble de instancias).

Las pruebas contempladas serían cuatro:

| Id | Nº atributos | Atributos |
|-----------------|---------------------|--|
| Prueba 1 | 34 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34. |
| Prueba 2 | 33 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34. |
| Prueba 3 | 20 | 4, 13, 14, 15, 16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34. |
| Prueba 4 | 15 | 4, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34. |

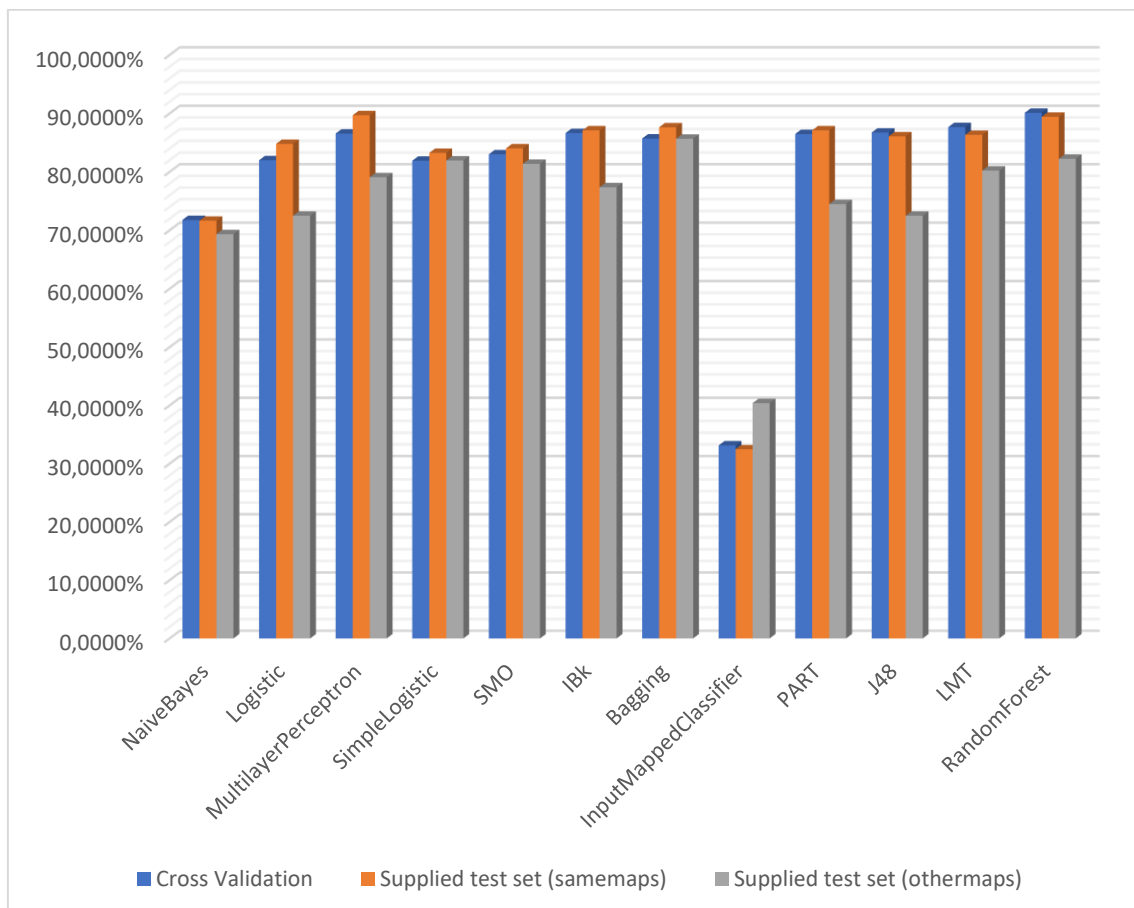
3.1.1. Ficheros 1: prueba 1

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 71,9358% | 72,6804% | 68,1948% | 70,9370% |
| Functions | Logistic | 82,2485% | 84,5361% | 71,9198% | 79,5681% |
| | MultilayerPerceptron | 87,1513% | 88,6598% | 78,2235% | 84,6782% |
| | SimpleLogistic | 81,7413% | 85,0515% | 78,5100% | 81,7676% |
| | SMO | 83,0938% | 84,2784% | 82,2350% | 83,2024% |
| Lazy | IBk | 86,5596% | 87,3711% | 76,7908% | 83,5738% |
| Meta | Bagging | 85,4607% | 87,8866% | 85,6734% | 86,3402% |
| Misc | InputMappedClassifier | 33,1361% | 32,4742% | 40,4011% | 35,3371% |
| Rules | PART | 86,0524% | 83,7629% | 73,3524% | 81,0559% |
| Trees | J48 | 86,8132% | 85,5670% | 78,2235% | 83,5346% |
| | LMT | 87,7430% | 86,3402% | 77,0774% | 83,7202% |
| | RandomForest | 90,4480% | 89,1753% | 84,2407% | 87,9547% |
| Valor medio | | 80,1986% | 80,6486% | 74,5702% | 78,4725% |



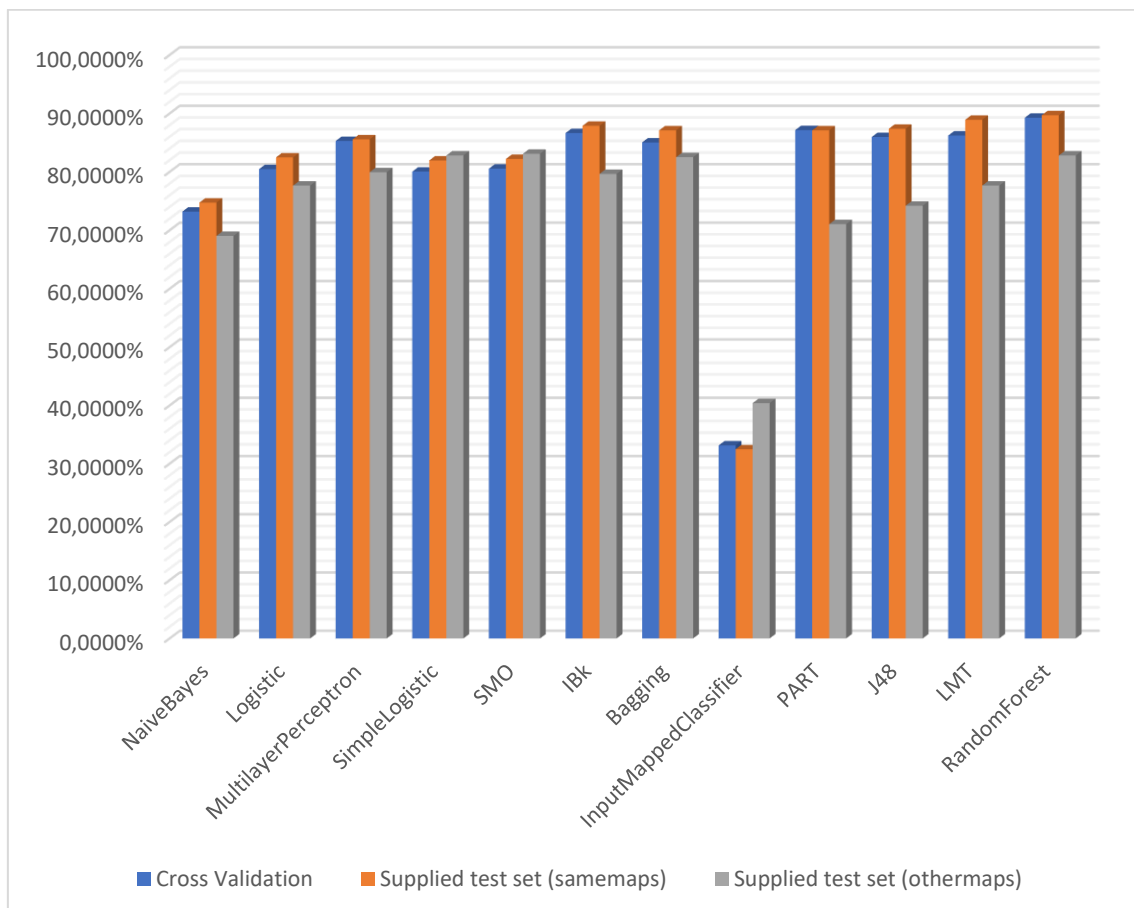
3.1.2. Ficheros 1: prueba 2

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 71,7667% | 71,6495% | 69,3410% | 70,9191% |
| Functions | Logistic | 81,9949% | 84,7938% | 72,4928% | 79,7605% |
| | MultilayerPerceptron | 86,5596% | 89,6907% | 79,0831% | 85,1111% |
| | SimpleLogistic | 81,9104% | 83,2474% | 81,9484% | 82,3687% |
| | SMO | 83,0093% | 84,0206% | 81,3754% | 82,8018% |
| Lazy | IBk | 86,6441% | 87,1134% | 77,3639% | 83,7071% |
| Meta | Bagging | 85,7143% | 87,6289% | 85,6734% | 86,3389% |
| Misc | InputMappedClassifier | 33,1361% | 32,4742% | 40,4011% | 35,3371% |
| Rules | PART | 86,4751% | 87,1134% | 74,4986% | 82,6957% |
| Trees | J48 | 86,7287% | 86,0825% | 72,4928% | 81,7680% |
| | LMT | 87,6585% | 86,3402% | 80,2292% | 84,7426% |
| | RandomForest | 90,1099% | 89,4330% | 82,2350% | 87,2593% |
| Valor medio | | 80,1423% | 80,7990% | 74,7612% | 78,5675% |



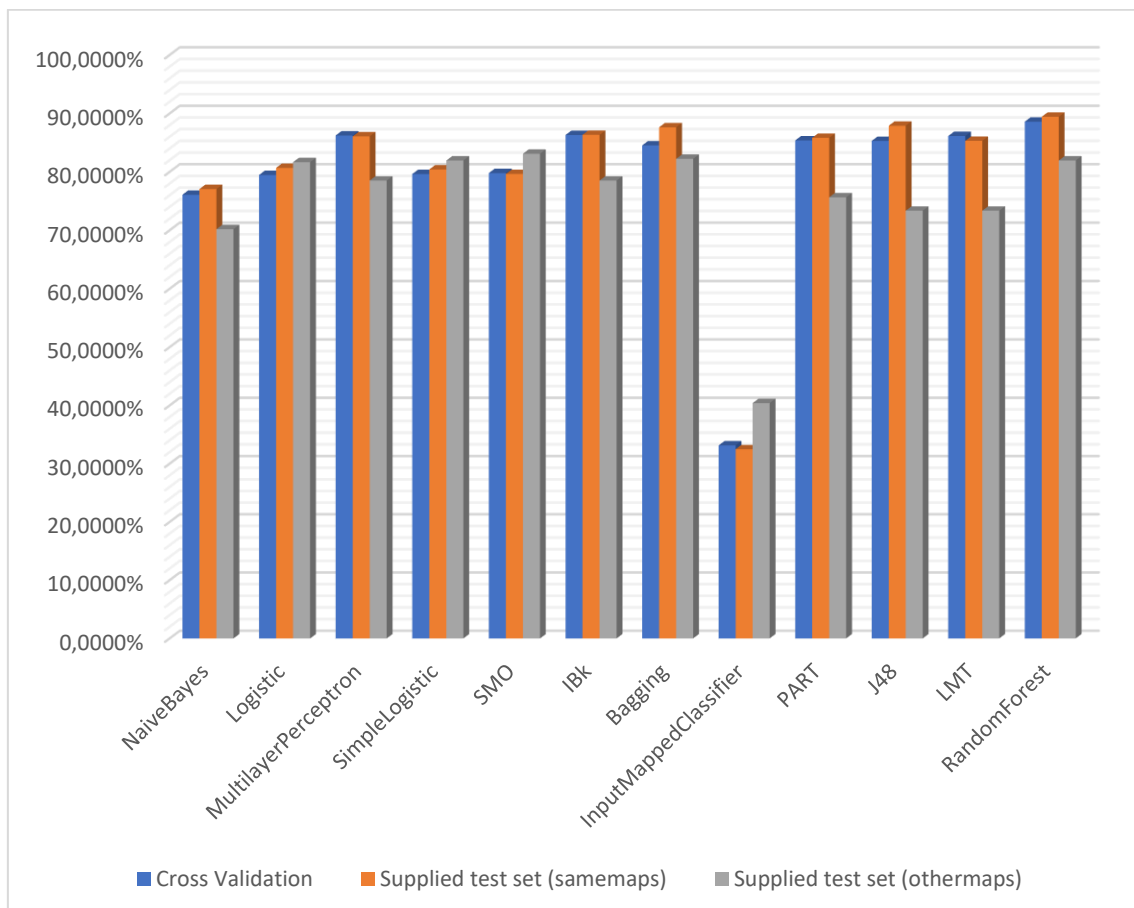
3.1.3. Ficheros 1: prueba 3

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 73,2037% | 74,7423% | 69,0544% | 72,3335% |
| Functions | Logistic | 80,4734% | 82,4742% | 77,6504% | 80,1993% |
| | MultilayerPerceptron | 85,2916% | 85,5670% | 79,9427% | 83,6004% |
| | SimpleLogistic | 80,0507% | 81,9588% | 82,8080% | 81,6058% |
| | SMO | 80,5579% | 82,2165% | 83,0946% | 81,9563% |
| Lazy | IBk | 86,6441% | 87,8866% | 79,6562% | 84,7290% |
| Meta | Bagging | 85,0380% | 87,1134% | 82,5215% | 84,8910% |
| Misc | InputMappedClassifier | 33,1361% | 32,4742% | 40,4011% | 35,3371% |
| Rules | PART | 87,1513% | 87,1134% | 71,0602% | 81,7750% |
| Trees | J48 | 85,9679% | 87,3711% | 74,2120% | 82,5170% |
| | LMT | 86,2215% | 88,9175% | 77,6504% | 84,2631% |
| | RandomForest | 89,2646% | 89,6907% | 82,8080% | 87,2544% |
| Valor medio | | 79,4167% | 80,6271% | 75,0716% | 78,3718% |



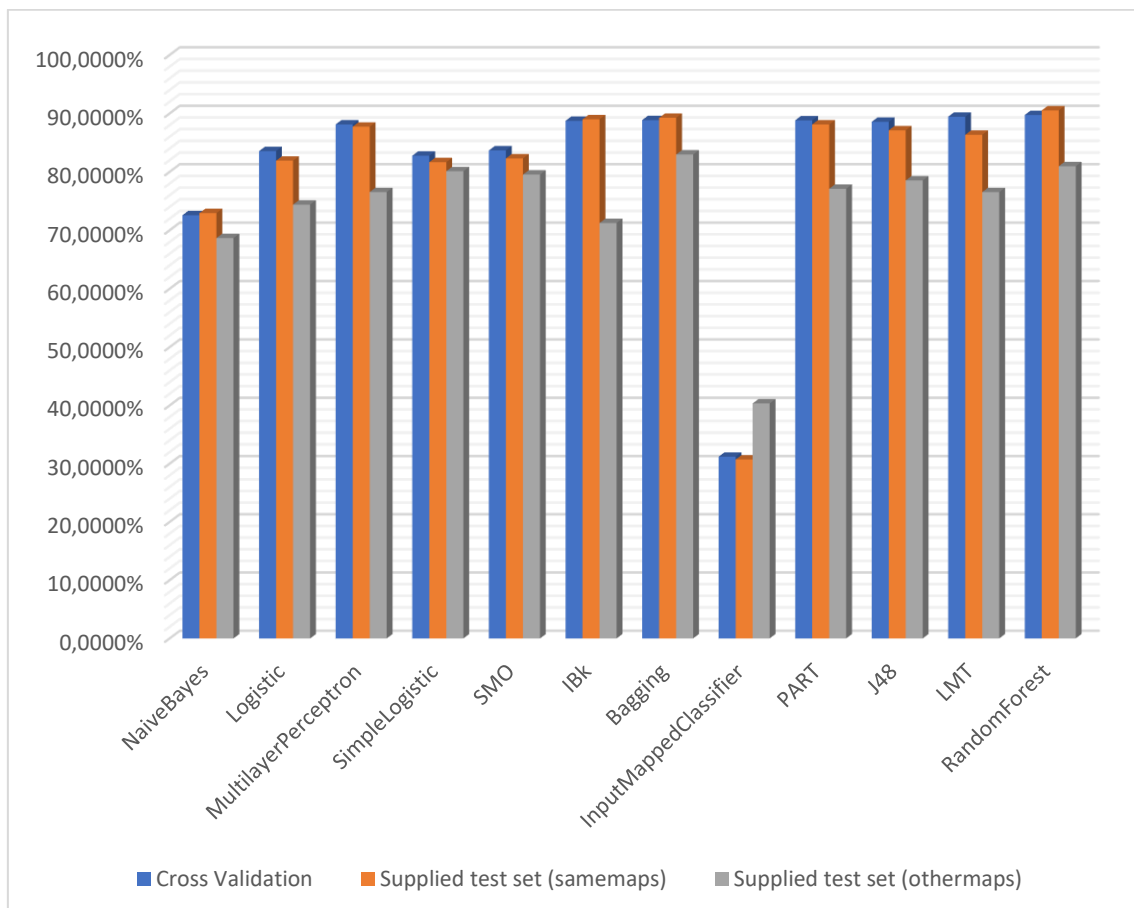
3.1.4. Ficheros 1: prueba 4

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 76,0778% | 77,0619% | 70,2006% | 74,4468% |
| Functions | Logistic | 79,4590% | 80,6701% | 81,6619% | 80,5970% |
| | MultilayerPerceptron | 86,2215% | 86,0825% | 78,5100% | 83,6047% |
| | SimpleLogistic | 79,6281% | 80,4124% | 81,9484% | 80,6630% |
| | SMO | 79,7971% | 79,6392% | 83,0946% | 80,8436% |
| Lazy | IBk | 86,3060% | 86,3402% | 78,5100% | 83,7187% |
| Meta | Bagging | 84,5309% | 87,6289% | 82,2350% | 84,7983% |
| Misc | InputMappedClassifier | 33,1361% | 32,4742% | 40,4011% | 35,3371% |
| Rules | PART | 85,3762% | 85,8247% | 75,6447% | 82,2819% |
| Trees | J48 | 85,2916% | 87,8866% | 73,3524% | 82,1769% |
| | LMT | 86,1369% | 85,3093% | 73,3524% | 81,5995% |
| | RandomForest | 88,5883% | 89,4330% | 81,9484% | 86,6566% |
| Valor medio | | 79,2125% | 79,8969% | 75,0716% | 78,0603% |



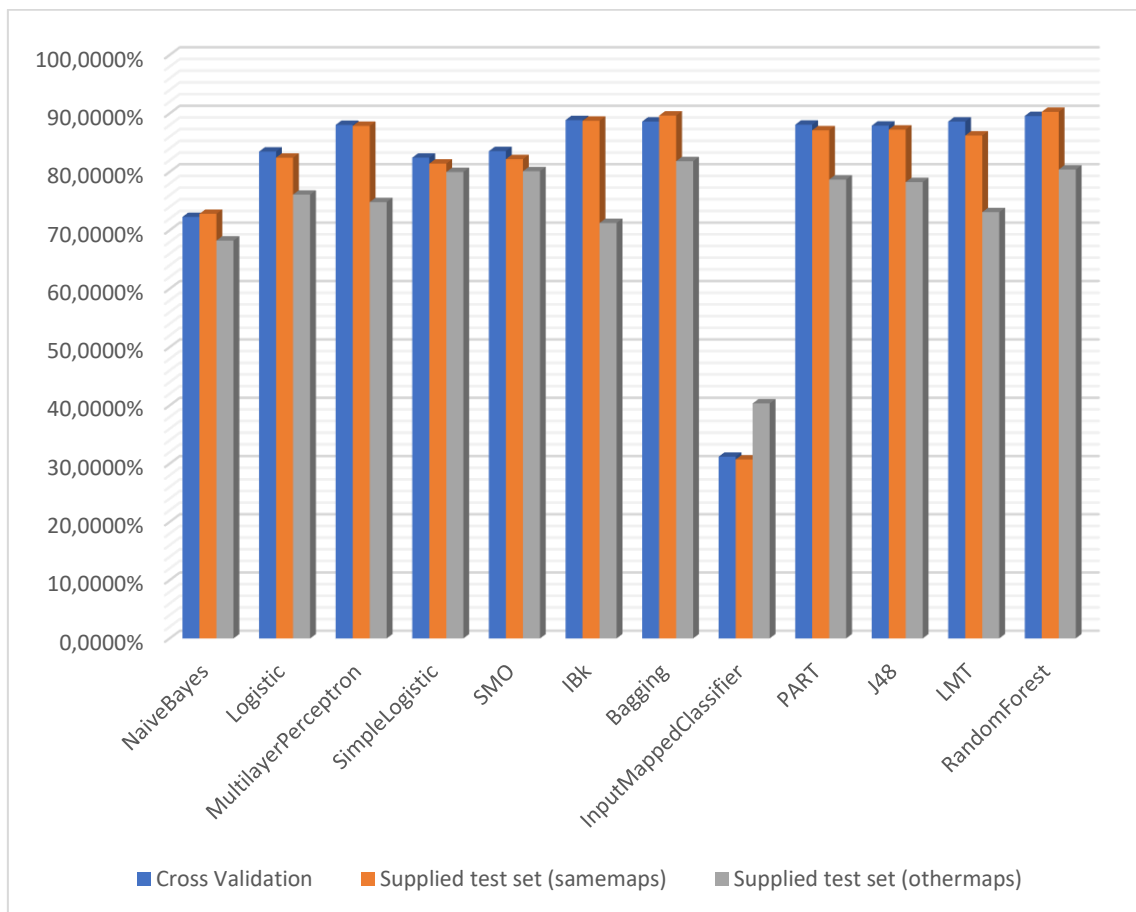
3.1.5. Ficheros 2: prueba 1

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 72,5766 % | 72,9456 % | 68,6695 % | 71,3972% |
| Functions | Logistic | 83,5543 % | 81,9216 % | 74,3920 % | 79,9560% |
| | MultilayerPerceptron | 88,1110 % | 87,7370 % | 76,5379 % | 84,1286% |
| | SimpleLogistic | 82,7672 % | 81,6688 % | 80,1144 % | 81,5168% |
| | SMO | 83,6785 % | 82,3009 % | 79,5422 % | 81,8405% |
| Lazy | IBk | 88,7324 % | 89,0013 % | 71,2446 % | 82,9928% |
| Meta | Bagging | 88,8567 % | 89,2541 % | 82,9757 % | 87,0288% |
| Misc | InputMappedClassifier | 31,1930 % | 30,7206 % | 40,3433 % | 34,0856% |
| Rules | PART | 88,8152 % | 88,1163 % | 77,1102 % | 84,6806% |
| Trees | J48 | 88,5667 % | 87,1049 % | 78,5408 % | 84,7375% |
| | LMT | 89,4366 % | 86,3464 % | 76,5379 % | 84,1070% |
| | RandomForest | 89,7266 % | 90,5183 % | 80,9728 % | 87,0726% |
| Valor medio | | 81,3346% | 80,6363% | 73,9151% | 78,6287% |



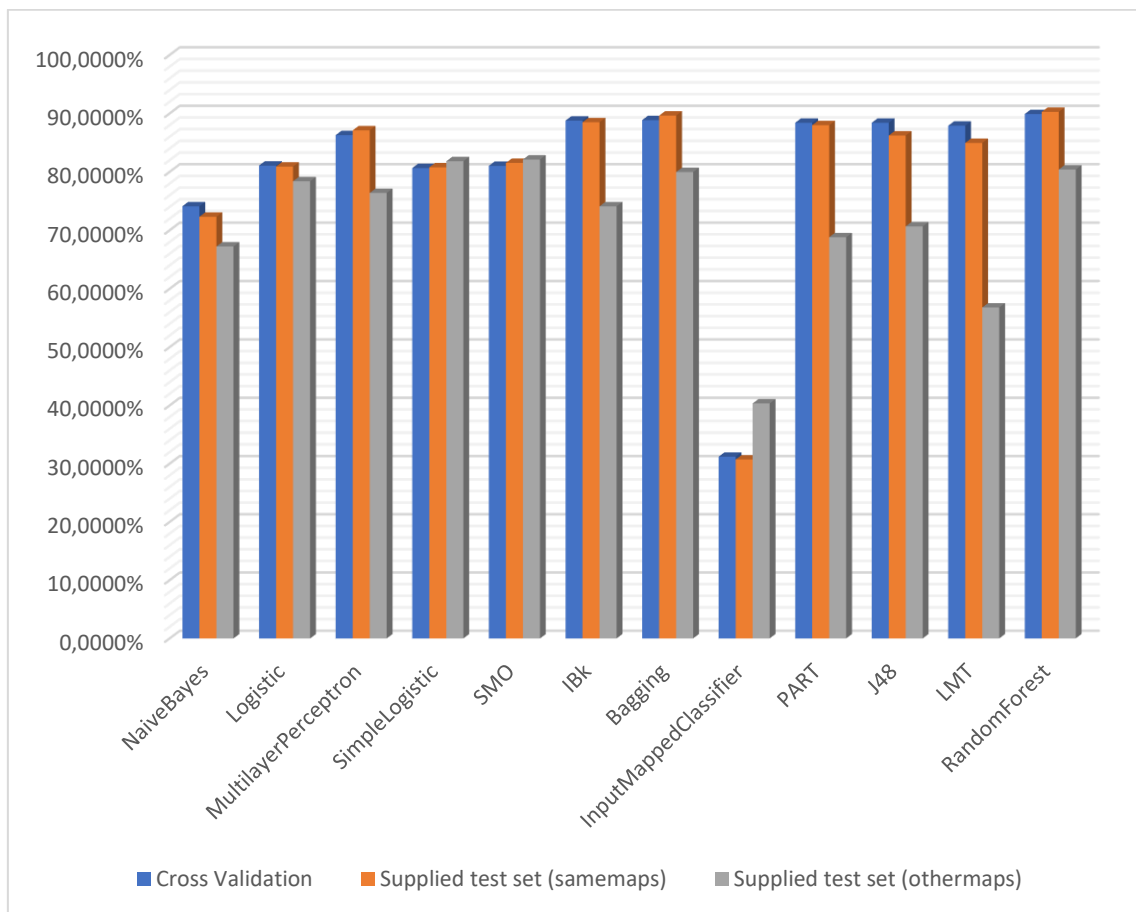
3.1.6. Ficheros 2: prueba 2

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 72,2867 % | 72,8192 % | 68,2403 % | 71,1154% |
| Functions | Logistic | 83,4714 % | 82,4273 % | 76,1087 % | 80,6691% |
| | MultilayerPerceptron | 88,0282 % | 87,8635 % | 74,8212 % | 83,5710% |
| | SimpleLogistic | 82,4358 % | 81,4159 % | 79,9714 % | 81,2744% |
| | SMO | 83,5543 % | 82,1745 % | 80,1144 % | 81,9477% |
| Lazy | IBk | 88,8567 % | 88,7484 % | 71,2446 % | 82,9499% |
| Meta | Bagging | 88,6081 % | 89,6334 % | 81,8312 % | 86,6909% |
| Misc | InputMappedClassifier | 31,1930 % | 30,7206 % | 40,3433 % | 34,0856% |
| Rules | PART | 88,0696 % | 87,1049 % | 78,6838 % | 84,6194% |
| Trees | J48 | 87,9039 % | 87,2314 % | 78,2546 % | 84,4633% |
| | LMT | 88,6081 % | 86,2200 % | 73,1044 % | 82,6442% |
| | RandomForest | 89,5609 % | 90,2655 % | 80,4006 % | 86,7423% |
| Valor medio | | 81,0481% | 80,5521% | 73,5932% | 78,3978% |



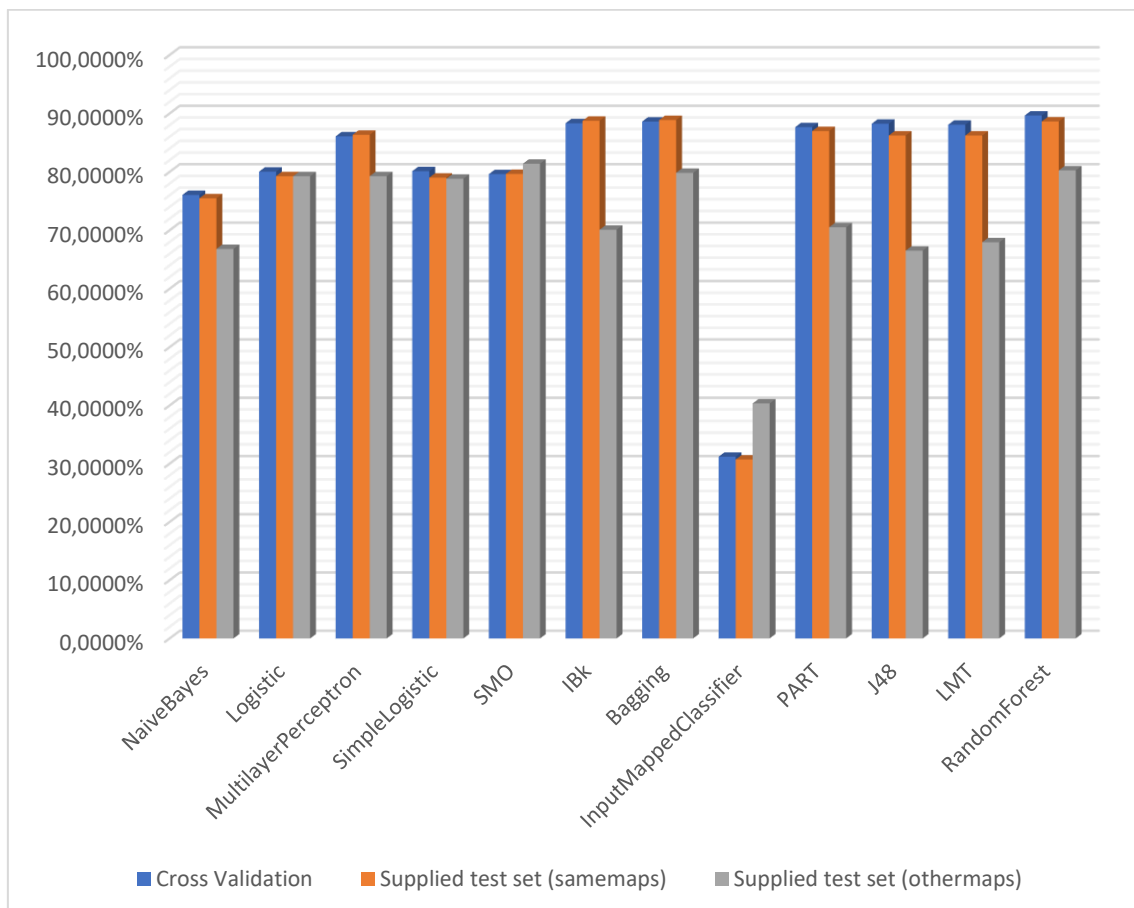
3.1.7. Ficheros 2: prueba 3

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 74,1094 % | 72,3135 % | 67,2389 % | 71,2206% |
| Functions | Logistic | 81,0688 % | 80,9102 % | 78,3977 % | 80,1256% |
| | MultilayerPerceptron | 86,2883 % | 87,1049 % | 76,3948 % | 83,2627% |
| | SimpleLogistic | 80,6545 % | 80,7838 % | 81,8312 % | 81,0898% |
| | SMO | 81,0273 % | 81,5424 % | 82,1173 % | 81,5623% |
| Lazy | IBk | 88,7738 % | 88,4956 % | 74,1059 % | 83,7918% |
| Meta | Bagging | 88,8567 % | 89,6334 % | 79,9714 % | 86,1538% |
| Misc | InputMappedClassifier | 31,1930 % | 30,7206 % | 40,3433 % | 34,0856% |
| Rules | PART | 88,4010 % | 87,9899 % | 68,8126 % | 81,7345% |
| Trees | J48 | 88,4010 % | 86,2200 % | 70,6724 % | 81,7645% |
| | LMT | 87,9039 % | 84,9558 % | 56,7954 % | 76,5517% |
| | RandomForest | 89,8923 % | 90,2655 % | 80,4006 % | 86,8528% |
| Valor medio | | 80,5475% | 80,0780% | 71,4235% | 77,3496% |



3.1.8. Ficheros 2: prueba 4

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 76,0563 % | 75,4741 % | 66,8097 % | 72,7800% |
| Functions | Logistic | 80,0331 % | 79,2668 % | 79,2561 % | 79,5187% |
| | MultilayerPerceptron | 86,0812 % | 86,3464 % | 79,2561 % | 83,8946% |
| | SimpleLogistic | 80,1160 % | 79,0139 % | 78,8269 % | 79,3189% |
| | SMO | 79,6189 % | 79,6460 % | 81,4020 % | 80,2223% |
| Lazy | IBk | 88,3181 % | 88,7484 % | 70,1001 % | 82,3889% |
| Meta | Bagging | 88,6081 % | 88,8748 % | 79,8283 % | 85,7704% |
| Misc | InputMappedClassifier | 31,1930 % | 30,7206 % | 40,3433 % | 34,0856% |
| Rules | PART | 87,6553 % | 86,9785 % | 70,5293 % | 81,7210% |
| Trees | J48 | 88,2353 % | 86,2200 % | 66,5236 % | 80,3263% |
| | LMT | 88,0696 % | 86,2200 % | 67,9542 % | 80,7479% |
| | RandomForest | 89,6437 % | 88,6220 % | 80,2575 % | 86,1744% |
| Valor medio | | 80,3024% | 79,6776% | 71,7573% | 77,2458% |



3.2. Resultados keyboard

Para la realización de este apartado se han tenido en cuenta los ficheros generados en la fase 1 utilizando el agente controlado por teclado. Con el objetivo de abarcar el mayor número de posibilidades para estudiar correctamente los resultados, además de utilizar los ficheros generados en la fase 1, se han creado otros ficheros que contienen las mismas instancias eliminando aquellas que contienen como acción “Stop”. A continuación se mostrarán los resultados obtenidos en cada una de las pruebas realizadas indicando “Ficheros” y “Prueba”.

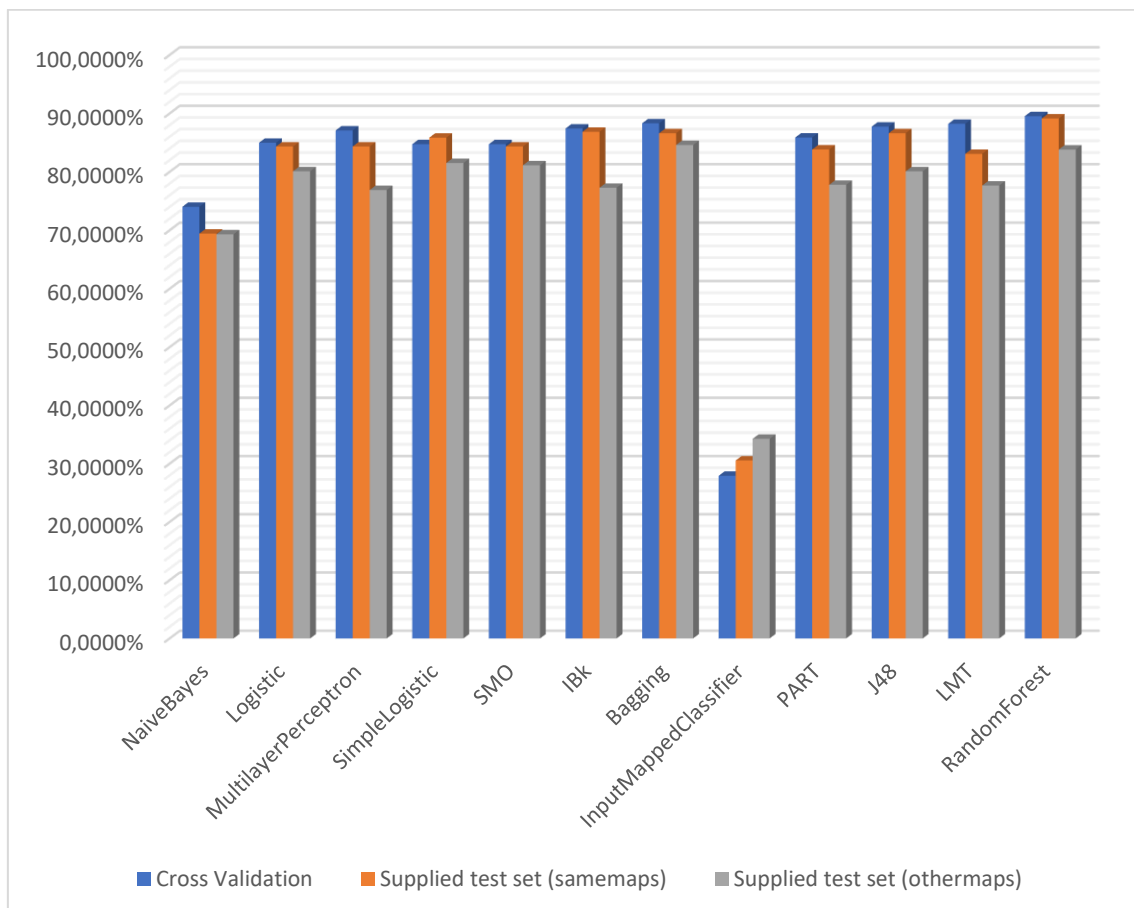
- “Ficheros 1” (instancias de la fase 1).
- “Ficheros 2” (instancias similares que no contienen la acción “Stop”).

Las pruebas contempladas serían tres:

| Id | Nº atributos | Atributos |
|-----------------|---------------------|--|
| Prueba 1 | 33 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34. |
| Prueba 2 | 20 | 4, 13, 14, 15, 16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34. |
| Prueba 3 | 15 | 4, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34. |

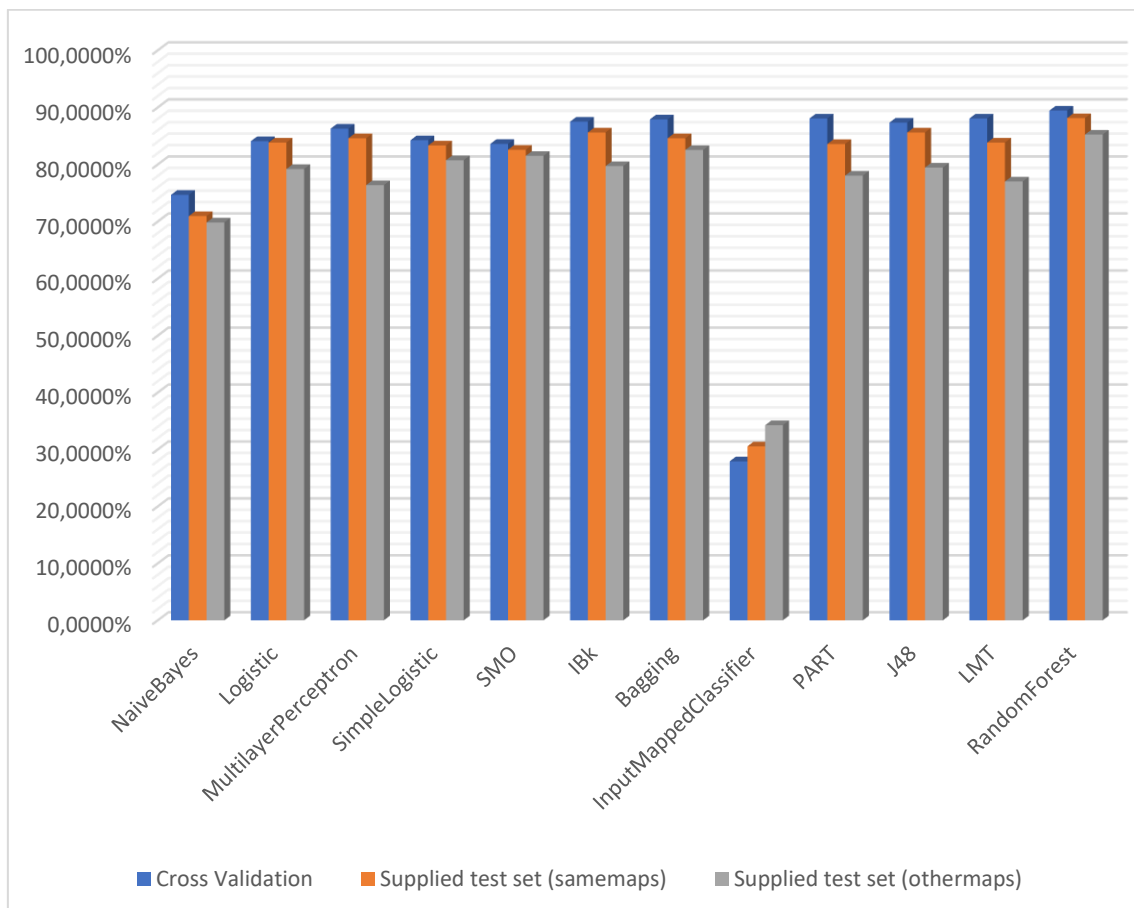
3.2.1. Ficheros 1: prueba 1

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 74,0260% | 69,4444% | 69,3196% | 70,9300% |
| Functions | Logistic | 84,9838% | 84,3434% | 80,1027% | 83,1433% |
| | MultilayerPerceptron | 87,0942% | 84,3434% | 76,8935% | 82,7770% |
| | SimpleLogistic | 84,7403% | 85,8586% | 81,5148% | 84,0379% |
| | SMO | 84,7403% | 84,3434% | 81,1297% | 83,4045% |
| Lazy | IBk | 87,4188% | 86,8687% | 77,2786% | 83,8554% |
| Meta | Bagging | 88,3117% | 86,6162% | 84,5956% | 86,5078% |
| Misc | InputMappedClassifier | 27,9221% | 30,5556% | 34,2747% | 30,9175% |
| Rules | PART | 85,8766% | 83,8384% | 77,7920% | 82,5023% |
| Trees | J48 | 87,7435% | 86,6162% | 80,1027% | 84,8208% |
| | LMT | 88,2305% | 83,0808% | 77,6637% | 82,9917% |
| | RandomForest | 89,5292% | 89,1414% | 83,8254% | 87,4987% |
| Valor medio | | 80,8848% | 79,5875% | 75,3744% | 78,6156% |



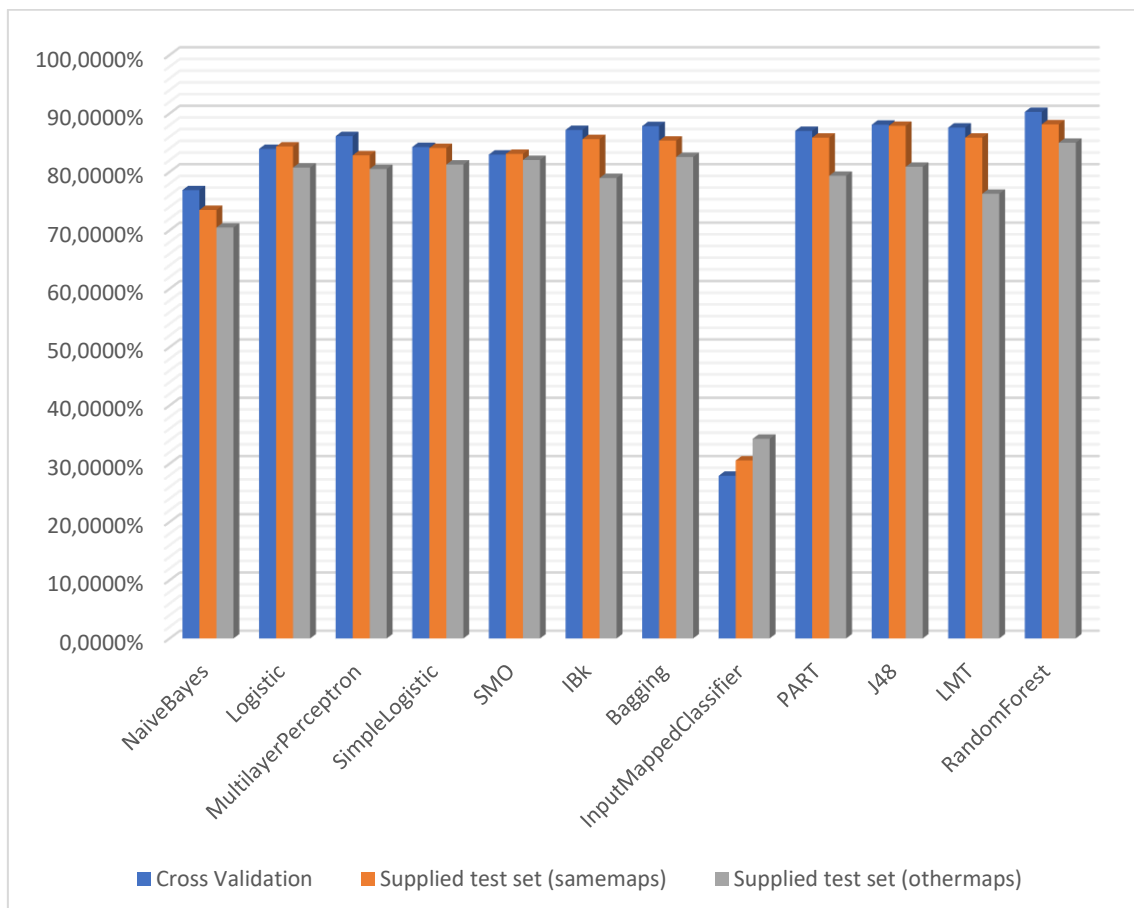
3.2.2. Ficheros 1: prueba 2

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 74,6753% | 70,9596% | 69,8331% | 71,8227% |
| Functions | Logistic | 84,0909% | 83,8384% | 79,2041% | 82,3778% |
| | MultilayerPerceptron | 86,2825% | 84,5960% | 76,3800% | 82,4195% |
| | SimpleLogistic | 84,2532% | 83,3333% | 80,7445% | 82,7770% |
| | SMO | 83,6039% | 82,5758% | 81,5148% | 82,5648% |
| Lazy | IBk | 87,5000% | 85,6061% | 79,7176% | 84,2746% |
| Meta | Bagging | 87,9058% | 84,5960% | 82,5417% | 85,0145% |
| Misc | InputMappedClassifier | 27,9221% | 30,5556% | 34,2747% | 30,9175% |
| Rules | PART | 88,0682% | 83,5859% | 78,0488% | 83,2343% |
| Trees | J48 | 87,3377% | 85,6061% | 79,4608% | 84,1349% |
| | LMT | 88,0682% | 83,8384% | 77,0218% | 82,9761% |
| | RandomForest | 89,4481% | 88,1313% | 85,2375% | 87,6056% |
| Valor medio | | 80,7630% | 78,9352% | 75,3316% | 78,3433% |



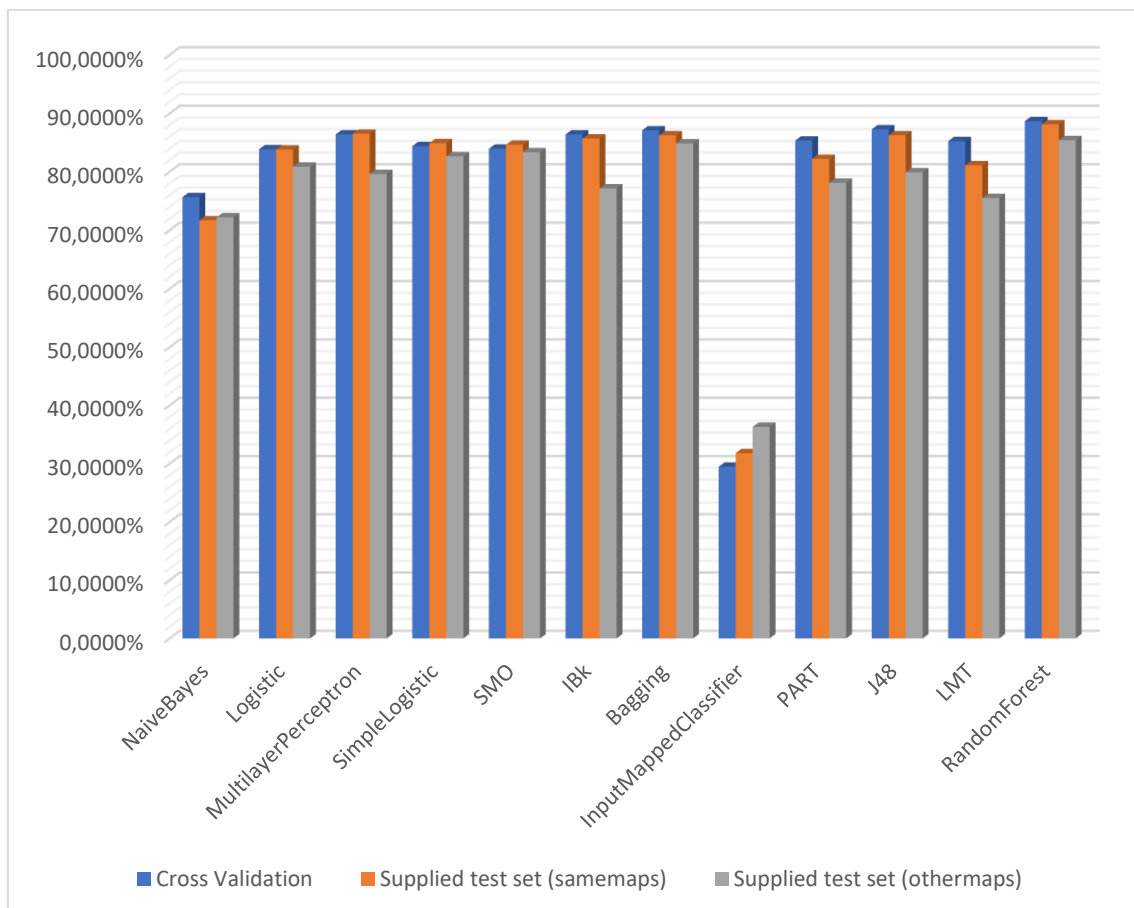
3.2.3. Ficheros 1: prueba 3

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 76,8669% | 73,4848% | 70,4750% | 73,6089% |
| Functions | Logistic | 83,9286% | 84,3434% | 80,7445% | 83,0055% |
| | MultilayerPerceptron | 86,1201% | 82,8283% | 80,4878% | 83,1454% |
| | SimpleLogistic | 84,2532% | 84,0909% | 81,2580% | 83,2007% |
| | SMO | 82,9545% | 83,0808% | 82,0282% | 82,6878% |
| Lazy | IBk | 87,1753% | 85,6061% | 78,9474% | 83,9096% |
| Meta | Bagging | 87,8247% | 85,3535% | 82,5417% | 85,2400% |
| Misc | InputMappedClassifier | 27,9221% | 30,5556% | 34,2747% | 30,9175% |
| Rules | PART | 87,0130% | 85,8586% | 79,3325% | 84,0680% |
| Trees | J48 | 88,0682% | 87,8788% | 80,8729% | 85,6066% |
| | LMT | 87,5812% | 85,8586% | 76,2516% | 83,2305% |
| | RandomForest | 90,2597% | 88,1313% | 84,9807% | 87,7906% |
| Valor medio | | 80,8306% | 79,7559% | 76,0163% | 78,8676% |



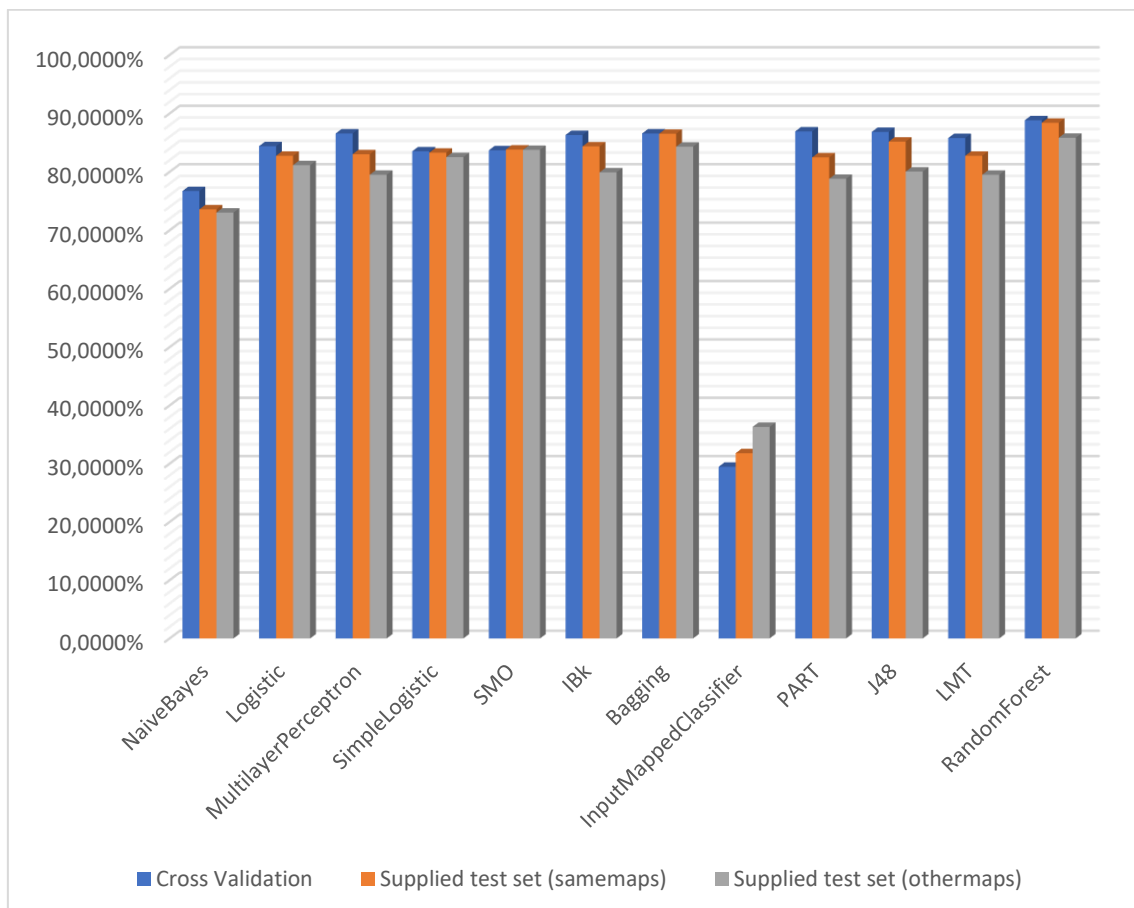
3.2.4. Ficheros 2: prueba 1

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 75,6757% | 71,6981% | 72,2146% | 73,1961% |
| Functions | Logistic | 83,8710% | 83,8275% | 80,8803% | 82,8596% |
| | MultilayerPerceptron | 86,3993% | 86,5229% | 79,6424% | 84,1882% |
| | SimpleLogistic | 84,3941% | 84,9057% | 82,6685% | 83,9894% |
| | SMO | 83,9582% | 84,6361% | 83,3563% | 83,9835% |
| Lazy | IBk | 86,3993% | 85,7143% | 77,1664% | 83,0933% |
| Meta | Bagging | 87,0968% | 86,2534% | 84,8693% | 86,0732% |
| Misc | InputMappedClassifier | 29,4682% | 31,8059% | 36,3136% | 32,5292% |
| Rules | PART | 85,3531% | 82,2102% | 78,1293% | 81,8975% |
| Trees | J48 | 87,2711% | 86,2534% | 79,9175% | 84,4807% |
| | LMT | 85,2659% | 81,1321% | 75,5158% | 80,6379% |
| | RandomForest | 88,6661% | 88,1402% | 85,4195% | 87,4086% |
| Valor medio | | 80,3182% | 79,4250% | 76,3411% | 78,6948% |



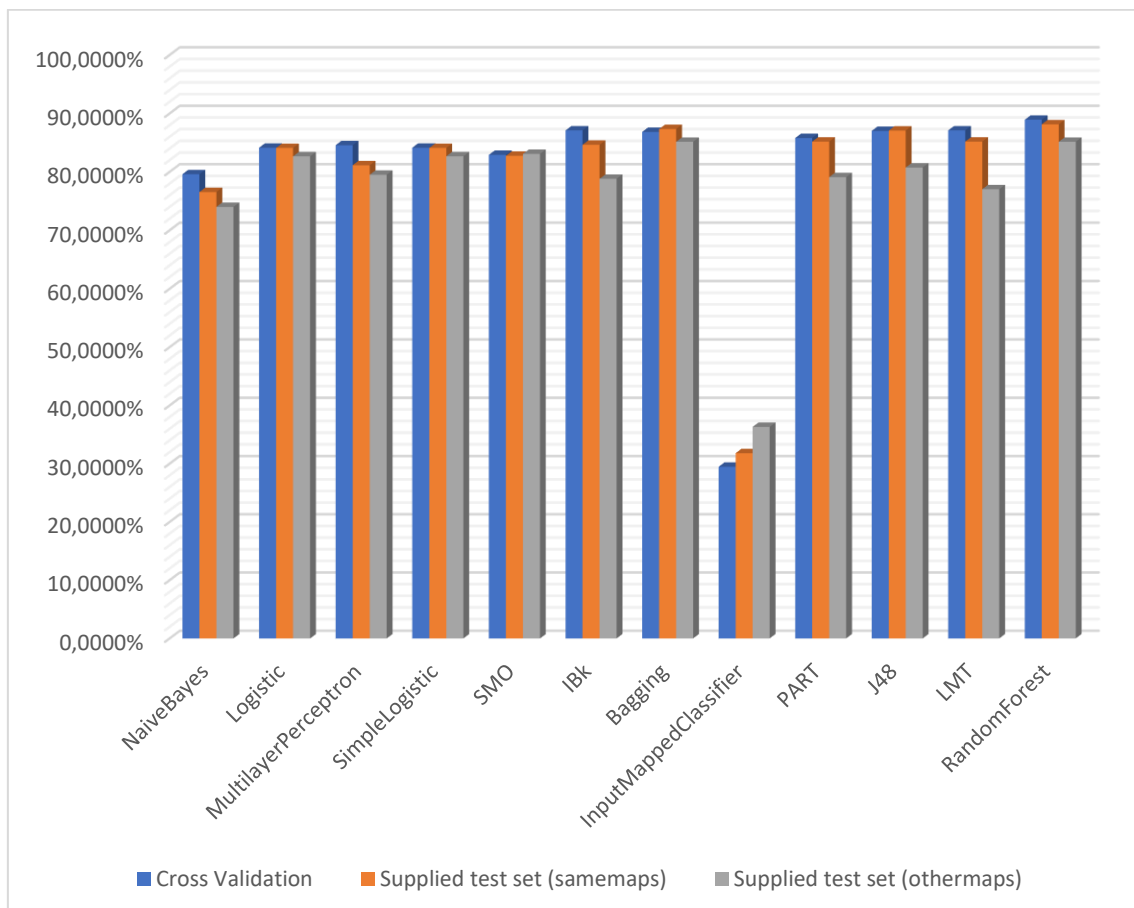
3.2.5. Ficheros 2: prueba 2

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 76,7219% | 73,5849% | 73,0399% | 74,4489% |
| Functions | Logistic | 84,3941% | 82,7493% | 81,1554% | 82,7663% |
| | MultilayerPerceptron | 86,5737% | 83,0189% | 79,5048% | 83,0325% |
| | SimpleLogistic | 83,5222% | 83,2884% | 82,5309% | 83,1138% |
| | SMO | 83,6966% | 83,8275% | 83,7689% | 83,7643% |
| Lazy | IBk | 86,3121% | 84,3666% | 79,9175% | 83,5321% |
| Meta | Bagging | 86,5737% | 86,5229% | 84,3191% | 85,8052% |
| Misc | InputMappedClassifier | 29,4682% | 31,8059% | 36,3136% | 32,5292% |
| Rules | PART | 86,9224% | 82,4798% | 78,8171% | 82,7398% |
| Trees | J48 | 86,8352% | 85,1752% | 80,0550% | 84,0218% |
| | LMT | 85,7890% | 82,7493% | 79,5048% | 82,6810% |
| | RandomForest | 88,8405% | 88,4097% | 85,8322% | 87,6941% |
| Valor medio | | 80,4708% | 78,9982% | 77,0633% | 78,8441% |



3.2.6. Ficheros 2: prueba 3

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Bayes | NaiveBayes | 79,5990% | 76,5499% | 74,0028% | 76,7172% |
| Functions | Logistic | 84,1325% | 84,0970% | 82,6685% | 83,6327% |
| | MultilayerPerceptron | 84,5684% | 81,1321% | 79,5048% | 81,7351% |
| | SimpleLogistic | 84,1325% | 84,0970% | 82,6685% | 83,6327% |
| | SMO | 82,9119% | 82,7493% | 83,0812% | 82,9141% |
| Lazy | IBk | 87,0968% | 84,6361% | 78,8171% | 83,5167% |
| Meta | Bagging | 86,8352% | 87,3315% | 85,1444% | 86,4370% |
| Misc | InputMappedClassifier | 29,4682% | 31,8059% | 36,3136% | 32,5292% |
| Rules | PART | 85,7890% | 85,1752% | 79,0922% | 83,3521% |
| Trees | J48 | 87,0096% | 87,0620% | 80,7428% | 84,9381% |
| | LMT | 87,0968% | 85,1752% | 77,0289% | 83,1003% |
| | RandomForest | 88,9276% | 88,1402% | 85,1444% | 87,4041% |
| Valor medio | | 80,6306% | 79,8293% | 77,0174% | 79,1591% |



3.3. Análisis final

En primer lugar, para la realización de esta práctica se han tenido en cuenta muchas pruebas previas que no aparecen reflejadas en el documento debido a la extensión de las mismas. Se ha comenzado con un grupo de atributos muy pequeño (los utilizados en el tutorial 1) y que consideramos que no contenían información suficiente para realizar un buen modelo, por lo que al observar los resultados de weka, se fueron añadiendo nuevos atributos que aportaran más valor. Una vez conseguido el número total de atributos, que son los que aparecen reflejados a la largo de este documento, se realizaron las pruebas anteriores (apartado 3.1 y 3.2).

A pesar de intentar mejorar constantemente el porcentaje de instancias clasificadas correctamente, nos hemos basado en lo que consideramos que aporta el atributo al modelo para elegir los adecuados. Los atributos de posición, aquellos que indican la fila y la columna en la que se encuentran el pacman y los fantasmas en el estado actual, podrían variar en función de las dimensiones del mapa e incluso no existir, por lo que podrían generar ruido, provocando que el agente automático no realice las acciones correctas. En cuanto a las distancias, a pesar de ser relevantes, ya que permiten saber si un fantasma está lejos o cerca del pacman, también hemos decidido eliminarlas porque para que el agente automático sea eficiente, tendría que ir primero a por el fantasma más cercano, por lo que hemos considerado que estas distancias no aportarían valor y decidimos añadir nuevos atributos para representar simplemente la distancia entre la posición actual del pacman y el fantasma más cercano. En el caso de la comida, ocurriría lo mismo, independientemente del número de puntos de comida, el pacman solo cogerá aquellos cercanos a su posición que estén de camino a un fantasma y que le permitan aumentar la puntuación, por lo que se ha decidido eliminar el atributo del total de puntos de comida y mantener simplemente la distancia al punto más cercano. En cuanto a los atributos que se ha decidido mantener tenemos principalmente aquellos que hacen referencia a la posición de las paredes. Consideramos que estos se encuentran entre los atributos relevantes ya que la principal dificultad del juego es conseguir que el pacman vaya a los fantasmas cuando hay paredes entre ellos.

En el apartado 3.2, se han realizado las pruebas utilizando los ficheros de la fase 1 y unos nuevos ficheros generados en los cuales se han eliminado las instancias que contenía "Stop" como acción. Esta decisión ha sido tomada ya que consideramos que estas instancias eliminadas servirían únicamente para realizar la acción "Stop" y haciendo que el agente automático en determinados momentos se detenga.

Finalmente, para la selección de algoritmos nos hemos basado en varios aspectos. En primer lugar, nos ha parecido interesante elegir algoritmos de todas las categorías (bayes, functions, lazy, meta, misc, rules, trees) para estudiar su comportamiento; además de permitirnos apreciar la similitud entre muchos de ellos. Por otro lado, en muchos casos se han seleccionado algoritmos estudiados o mencionados en clase, por lo que utilizarlos nos ha favorecido en la comprensión de ciertos aspectos teóricos. Finalmente, al observar los resultados de las pruebas anteriores, se puede concluir teóricamente que el algoritmo "InputMappedClassifier" sería el menos indicado para generar un buen modelo debido a su bajo porcentaje de instancias clasificadas correctamente y que, aunque el resto de los algoritmos presenten porcentajes similares, "RandomForest" destaca un poco por encima de estos.

4. Fase 3: Regresión

En esta fase se propone explorar los algoritmos de predicción de Weka de forma experimental y como consecuencia a ello se deberá construir un modelo de predicción con la información obtenida.

4.1. Resultados de tutorial 1.

Para la ello de este apartado se han tenido en cuenta los ficheros generados en la fase 1 utilizando el agente programado en el tutorial1. En este caso, se han realizado pruebas con los ficheros de instancias generados por el agente automático del tutorial 1. En esta ocasión, no se han duplicado el valor de las instancias para hacer dos tipos de fichero ya que nos hemos analizado que los resultados obtenidos en ambas pruebas con distintos ficheros eran similares y obteníamos unos resultados muy parecidos. A continuación, se mostrarán los resultados obtenidos en cada una de las pruebas realizadas indicando la prueba realizada en cada caso.

Las pruebas contempladas serían cuatro:

| Id | Nº atributos | Atributos |
|----------|--------------|--|
| Prueba 1 | 35 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35. |
| Prueba 2 | 23 | 3, 4, 13,14, 15, 16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35 |
| Prueba 3 | 23 | 1, 2, 3, 4,5, 6, 7, 8, 9, 10, 11, 12, 17, 18, 22, 24, 25, 26, 27, 32, 33, 34, 35 |
| Prueba 4 | 27 | 1, 2, 3, 4,5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,19, 20, 21, 22, 23, 32, 33, 34, 35 |

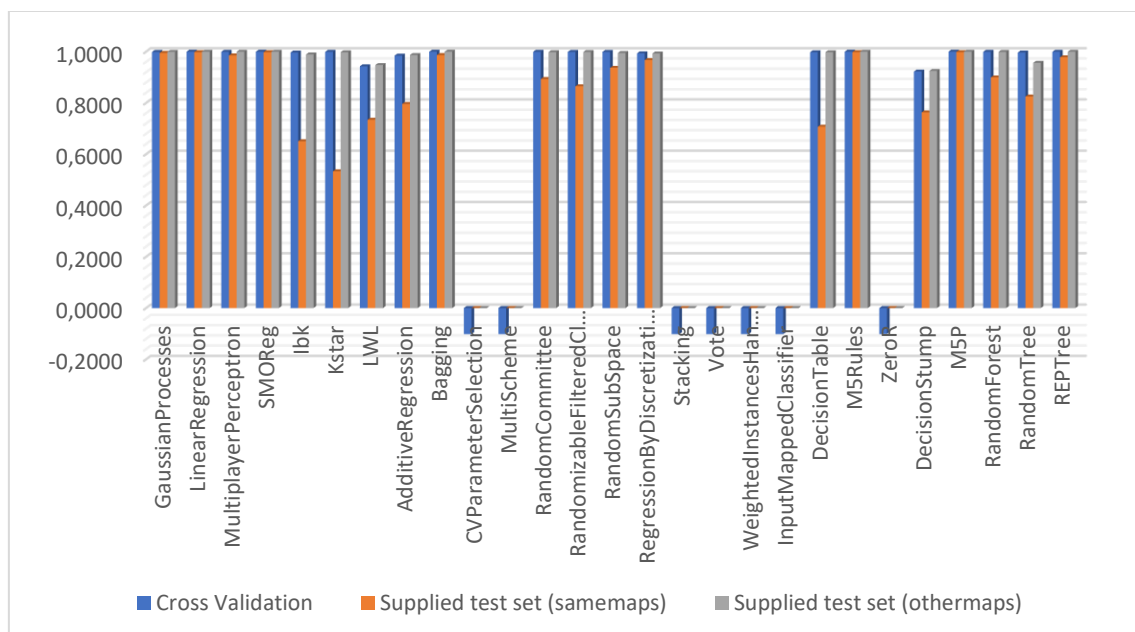
Finalmente, cabe destacar que el procedimiento de la elección de los dos mejores clasificadores en este caso ha sido distinto a la fase anterior ya que se ha tratado de encontrar el mejor utilizando todos los clasificadores posibles. Esto es, en la primera prueba comenzamos utilizando todos los posibles clasificadores con el mismo conjunto de atributos de los cuales se elegirán la mitad de los mejores resultados en la media de los valores obtenidos, en la segunda prueba con los mejores clasificadores seleccionados en la primera prueba se realizará otro análisis del coeficiente de correlación en otro conjunto de atributos diferentes y se seguirá eligiendo la mitad superior de la media de los resultados obtenidos. De esta forma, se conseguirá llegar a los dos mejores clasificadores utilizando distintos conjuntos de atributos en cada prueba.

4.1.1. Prueba 1.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 35 atributos y utilizaremos los 27 clasificadores disponibles en Weka.

| | | Cross Validation | Supplied test set (samema ps) | Supplied test set (otherma ps) | Valor medio |
|----------|----------------------|------------------|-------------------------------|--------------------------------|-------------|
| Funtions | GaussianProcesses | 0,9988 | 0,9989 | 0,9951 | 0,9976 |
| | LinearRegression | 0,9997 | 0,9995 | 0,9979 | 0,9990 |
| | MultilayerPerceptron | 0,9993 | 0,9995 | 0,9852 | 0,9947 |
| | SMOReg | 0,9996 | 0,9995 | 0,9978 | 0,9990 |

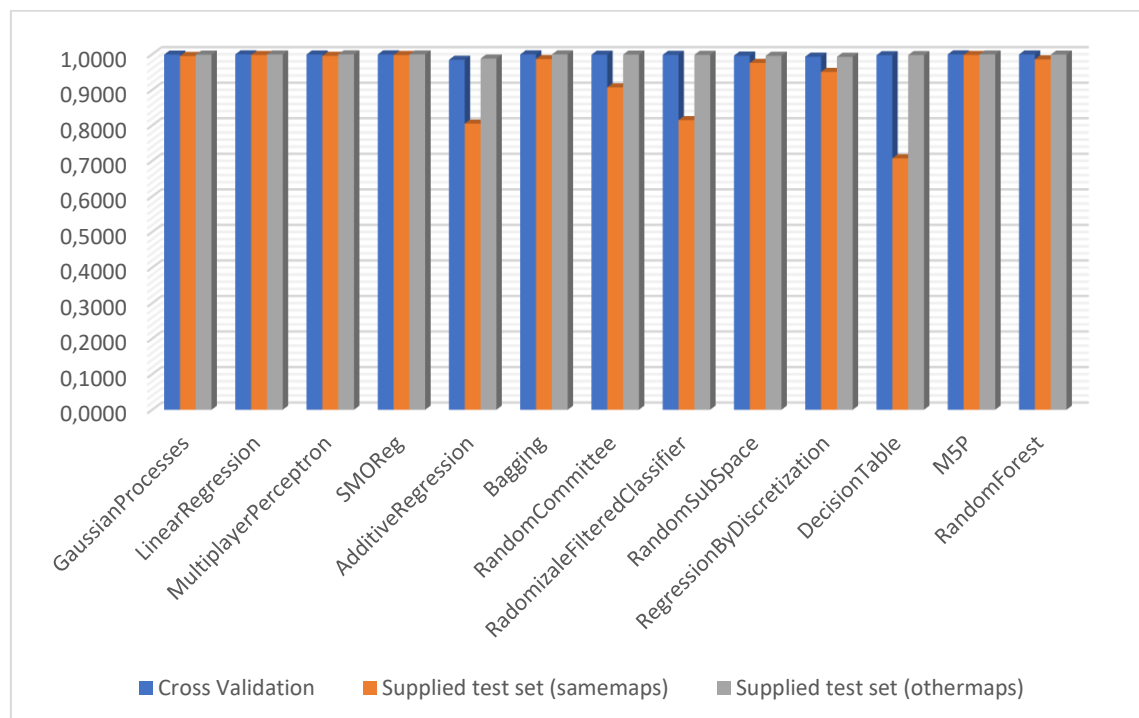
| | | | | | |
|-------------|---------------------------------|---------|--------|--------|---------|
| Lazy | Ibk | 0,9963 | 0,9888 | 0,6506 | 0,8786 |
| | KStar | 0,9991 | 0,9974 | 0,5333 | 0,8433 |
| | LWL | 0,9426 | 0,9475 | 0,7343 | 0,8748 |
| Meta | AdditiveRegression | 0,9843 | 0,9867 | 0,7959 | 0,9223 |
| | Bagging | 0,9996 | 0,9997 | 0,9865 | 0,9953 |
| | CVParameterSelection | -0,1016 | 0 | 0 | -0,0339 |
| | MultiScheme | -0,1016 | 0 | 0 | -0,0339 |
| | RandomCommittee | 0,9993 | 0,9979 | 0,8932 | 0,9635 |
| | RandomizableFilteredClassifier | 0,9987 | 0,9985 | 0,8649 | 0,9540 |
| | RandomSubSpace | 0,9988 | 0,9946 | 0,9369 | 0,9768 |
| | RegressionByDiscretization | 0,9933 | 0,9928 | 0,9671 | 0,9844 |
| | Stacking | -0,1016 | 0 | 0 | -0,0339 |
| | Vote | -0,1016 | 0 | 0 | -0,0339 |
| | WeightedInstancesHandlerWrapper | -0,1016 | 0 | 0 | -0,0339 |
| Misc | InputMappedClassifier | -0,1016 | 0 | 0 | -0,0339 |
| Rules | DecisionTable | 0,9975 | 0,9976 | 0,7073 | 0,9008 |
| | M5Rules | 0,9996 | 0,9995 | 0,998 | 0,9990 |
| | ZeroR | -0,1016 | 0 | 0 | -0,0339 |
| Trees | DecisionStump | 0,9223 | 0,9247 | 0,763 | 0,8700 |
| | M5P | 0,9996 | 0,9995 | 0,998 | 0,9990 |
| | RandomForest | 0,9993 | 0,9988 | 0,8994 | 0,9658 |
| | RandomTree | 0,9965 | 0,9563 | 0,8248 | 0,9259 |
| | REPTree | 0,9994 | 0,9995 | 0,978 | 0,9923 |
| Valor medio | | 0,7079 | 0,6484 | 0,7325 | 0,6963 |



4.1.2. Prueba 2.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 23 atributos y utilizaremos los 13 mejores clasificadores obtenidos en la prueba anterior.

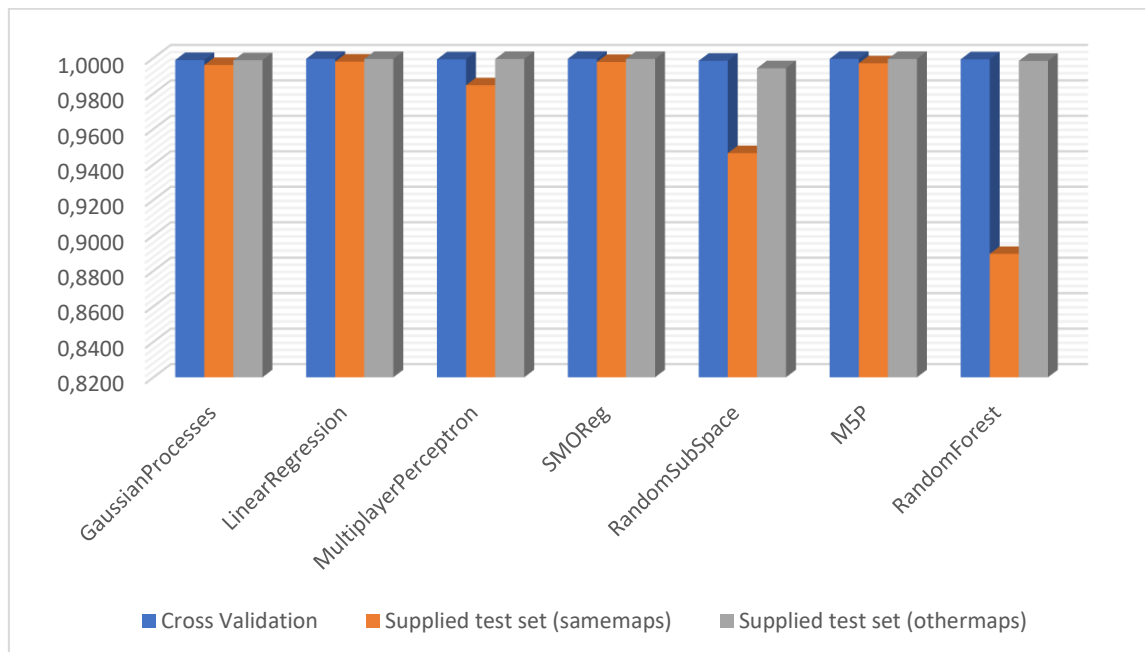
| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|-------------|--------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Funtions | GaussianProcesses | 0,999 | 0,9989 | 0,9952 | 0,9977 |
| | LinearRegression | 0,9997 | 0,9995 | 0,9981 | 0,9991 |
| | MultilayerPerceptron | 0,9994 | 0,9996 | 0,9956 | 0,9982 |
| | SMOReg | 0,9996 | 0,9995 | 0,9979 | 0,9990 |
| Meta | AdditiveRegression | 0,9845 | 0,9879 | 0,805 | 0,9258 |
| | Bagging | 0,9996 | 0,9996 | 0,9866 | 0,9953 |
| | RandomCommittee | 0,9988 | 0,999 | 0,9067 | 0,9682 |
| | RandomizableFilteredClassifier | 0,9985 | 0,9984 | 0,8146 | 0,9372 |
| | RandomSubSpace | 0,9962 | 0,9956 | 0,9757 | 0,9892 |
| | RegressionByDiscretization | 0,9932 | 0,9928 | 0,9503 | 0,9788 |
| Rules | DecisionTable | 0,9975 | 0,9976 | 0,7073 | 0,9008 |
| Trees | M5P | 0,9996 | 0,9995 | 0,9981 | 0,9991 |
| | RandomForest | 0,9994 | 0,999 | 0,9857 | 0,9947 |
| Valor medio | | 0,9973 | 0,9321 | 0,9975 | 0,9756 |



4.1.3. Prueba 3.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 23 atributos y utilizaremos los 7 mejores clasificadores obtenidos en la prueba anterior.

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|-----------------------------|------------------|------------------------------|-------------------------------|-------------|
| Funtions | GaussianProcesses | 0,9989 | 0,9988 | 0,9961 | 0,9979 |
| | LinearRegression | 0,9996 | 0,9995 | 0,998 | 0,9990 |
| | MultilayerPerceptron | 0,9993 | 0,9995 | 0,9847 | 0,9945 |
| | SMOReg | 0,9996 | 0,9995 | 0,9978 | 0,9990 |
| | RandomSubSpace | 0,9985 | 0,9943 | 0,9464 | 0,9797 |
| Trees | M5P | 0,9996 | 0,9995 | 0,9971 | 0,9987 |
| | RandomForest | 0,9993 | 0,9984 | 0,8896 | 0,9624 |
| Valor medio | | 0,9993 | 0,9728 | 0,9985 | 0,9902 |

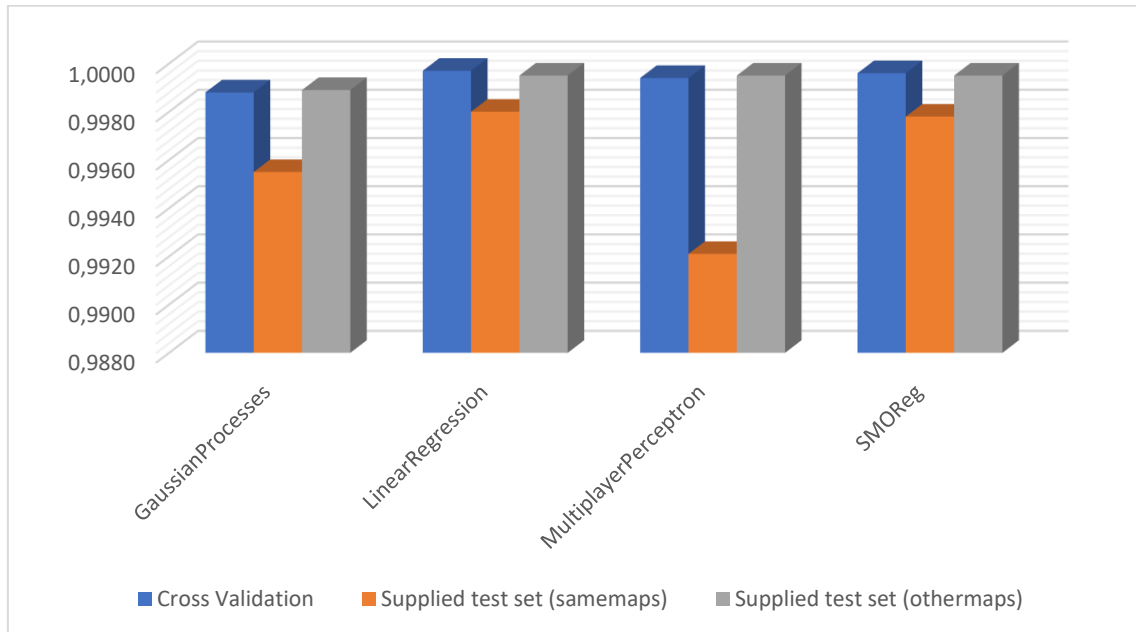


4.1.4. Prueba 4.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 27 atributos y utilizaremos los 4 mejores clasificadores obtenidos en la prueba anterior.

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|-----------------|--------------------------|------------------|------------------------------|-------------------------------|-------------|
| Funtions | GaussianProcesses | 0,9988 | 0,9989 | 0,9955 | 0,9977 |

| | | | | | |
|--|-----------------------------|--------|--------|--------|--------|
| | LinearRegression | 0,9997 | 0,9995 | 0,998 | 0,9991 |
| | MultilayerPerceptron | 0,9994 | 0,9995 | 0,9921 | 0,9970 |
| | SMOReg | 0,9996 | 0,9995 | 0,9978 | 0,9990 |
| | Valor medio | 0,9994 | 0,9959 | 0,9994 | 0,9982 |



4.2. Resultados Keyboard.

Para la realización de este apartado se han tenido en cuenta los ficheros generados en la fase 1 utilizando el agente controlado por teclado. Con el objetivo de abarcar el mayor número de posibilidades para estudiar correctamente los resultados, además de utilizar los ficheros generados en la fase 1, se han creado otros ficheros que contienen las mismas instancias eliminando aquellas que contienen como acción "Stop". A continuación, se mostrarán los resultados obtenidos en cada una de las pruebas realizadas indicando "Ficheros" y "Prueba".

- "Ficheros 1" (instancias de la fase 1).
- "Ficheros 2" (instancias similares que no contienen la acción "Stop").

Las pruebas contempladas serían tres:

| Id | Nº atributos | Atributos |
|-----------------|--------------|--|
| Prueba 1 | 35 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35. |
| Prueba 2 | 23 | 3, 4, 13, 14, 15, 16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35 |
| Prueba 3 | 23 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17, 18, 22, 24, 25, 26, 27, 32, 33, 34, 35 |
| Prueba 4 | 27 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 32, 33, 34, 35 |

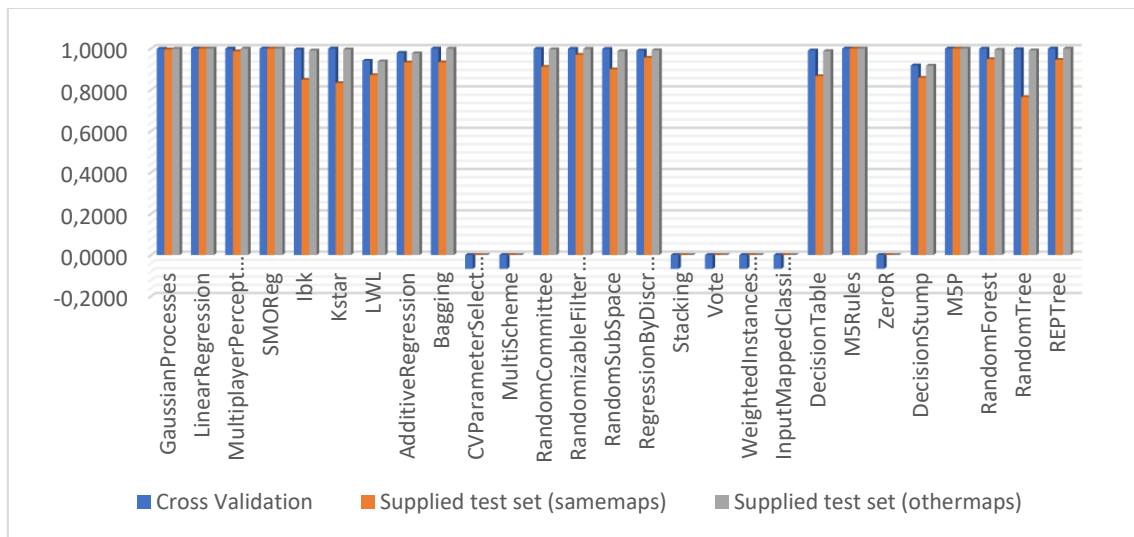
Finalmente, cabe destacar que el procedimiento de la elección de los dos mejores clasificadores en este caso ha sido distinto a la fase anterior ya que se ha tratado de encontrar el mejor

utilizando todos los clasificadores posibles. Esto es, en la primera prueba comenzamos utilizando todos los posibles clasificadores con el mismo conjunto de atributos de los cuales se elegirán la mitad de los mejores resultados en la media de los valores obtenidos, en la segunda prueba con los mejores clasificadores seleccionados en la primera prueba se realizará otro análisis del coeficiente de correlación en otro conjunto de atributos diferentes y se seguirá eligiendo la mitad superior de la media de los resultados obtenidos. De esta forma, se conseguirá llegar a los dos mejores clasificadores utilizando distintos conjuntos de atributos en cada prueba.

4.2.1. Ficheros 1: Prueba 1.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 35 atributos y utilizaremos los 27 clasificadores disponibles en Weka.

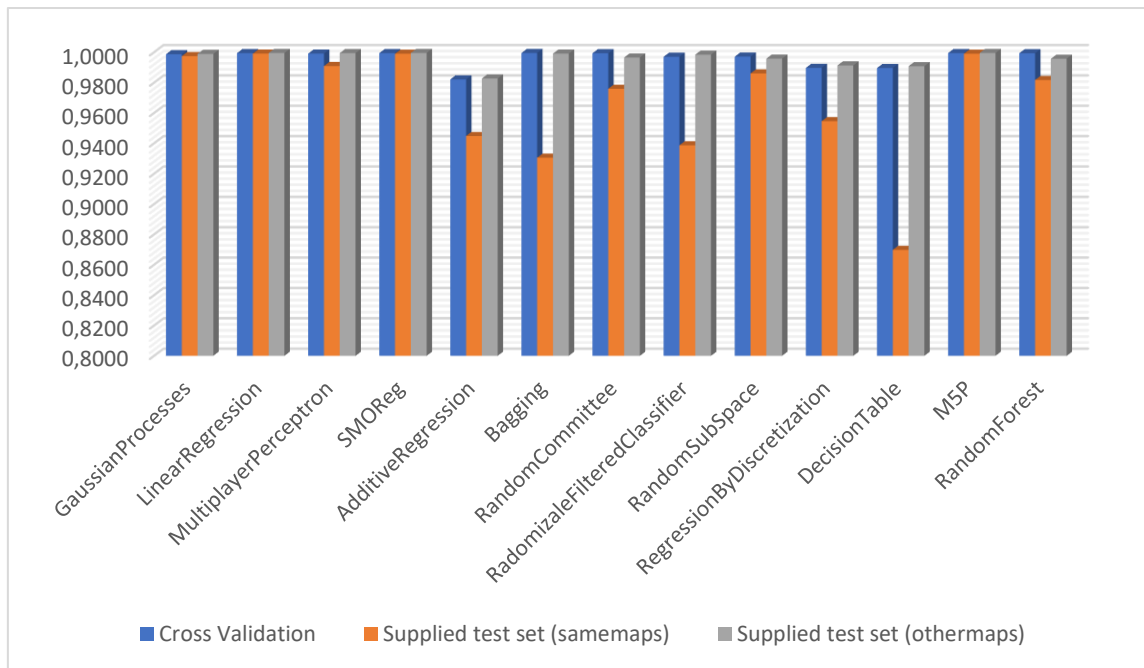
| | | Cross Validati on | Supplied test set (otherma ps) | Supplied test set (samema ps) | Valor medio |
|----------------------|---|-------------------------|---|--|----------------|
| Funtio ns | GaussianProcesses | 0,9982 | 0,9984 | 0,9952 | 0,9973 |
| | LinearRegression | 0,9994 | 0,9995 | 0,9991 | 0,9993 |
| | MultilayerPerceptron | 0,9991 | 0,999 | 0,9857 | 0,9946 |
| | SMOReg | 0,9994 | 0,9995 | 0,999 | 0,9993 |
| Lazy | Ibk | 0,9951 | 0,99 | 0,8479 | 0,9443 |
| | KStar | 0,999 | 0,9955 | 0,8319 | 0,9421 |
| | LWL | 0,9405 | 0,9372 | 0,8705 | 0,9161 |
| Meta | AdditiveRegression | 0,9784 | 0,9761 | 0,9319 | 0,9621 |
| | Bagging | 0,9995 | 0,9988 | 0,9326 | 0,9770 |
| | CVParameterSelection | -0,067 | 0 | 0 | -0,0223 |
| | MultiScheme | -0,067 | 0 | 0 | -0,0223 |
| | RandomCommittee | 0,9984 | 0,9956 | 0,9107 | 0,9682 |
| | RandomizableFilteredClassifie r | 0,9984 | 0,998 | 0,9685 | 0,9883 |
| | RandomSubSpace | 0,9975 | 0,9868 | 0,8988 | 0,9610 |
| | RegressionByDiscretization | 0,9897 | 0,9913 | 0,9546 | 0,9785 |
| | Stacking | -0,067 | 0 | 0 | -0,0223 |
| | Vote | -0,067 | 0 | 0 | -0,0223 |
| | WeightedInstancesHandlerWr apper | -0,067 | 0 | 0 | -0,0223 |
| Misc | InputMappedClassifier | -0,067 | 0 | 0 | -0,0223 |
| Rules | DecisionTable | 0,9901 | 0,9869 | 0,8655 | 0,9475 |
| | M5Rules | 0,9994 | 0,9995 | 0,999 | 0,9993 |
| | ZeroR | -0,067 | 0 | 0 | -0,0223 |
| Trees | DecisionStump | 0,9174 | 0,9166 | 0,8577 | 0,8972 |
| | M5P | 0,9994 | 0,9995 | 0,999 | 0,9993 |
| | RandomForest | 0,9989 | 0,9935 | 0,9475 | 0,9800 |
| | RandomTree | 0,9963 | 0,9907 | 0,7643 | 0,9171 |
| | REPTree | 0,9991 | 0,9993 | 0,9443 | 0,9809 |
| Valor medio | | 0,7157 | 0,6853 | 0,7315 | 0,7109 |



4.2.2. Ficheros 1: Prueba 2.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 23 atributos y utilizaremos los 13 mejores clasificadores obtenidos en la prueba anterior.

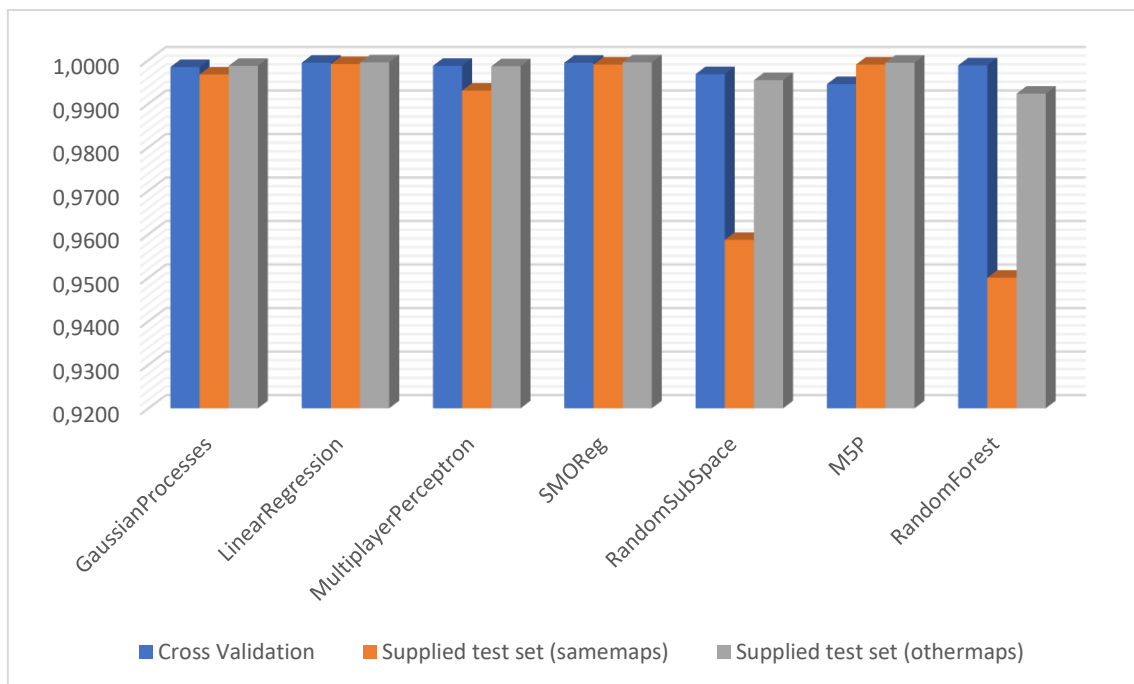
| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|-------------|--------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Funtions | GaussianProcesses | 0,9987 | 0,9989 | 0,9974 | 0,9983 |
| | LinearRegression | 0,9994 | 0,9995 | 0,9991 | 0,9993 |
| | MultilayerPerceptron | 0,9991 | 0,9994 | 0,9909 | 0,9965 |
| | SMOReg | 0,9994 | 0,9995 | 0,999 | 0,9993 |
| Meta | AdditiveRegression | 0,9821 | 0,9827 | 0,9448 | 0,9699 |
| | Bagging | 0,9995 | 0,9991 | 0,9305 | 0,9764 |
| | RandomCommittee | 0,9993 | 0,9965 | 0,9759 | 0,9906 |
| | RandomizableFilteredClassifier | 0,997 | 0,9984 | 0,9387 | 0,9780 |
| | RandomSubSpace | 0,9971 | 0,9958 | 0,986 | 0,9930 |
| | RegressionByDiscretization | 0,9897 | 0,9913 | 0,9546 | 0,9785 |
| Rules | DecisionTable | 0,9896 | 0,9908 | 0,8698 | 0,9501 |
| Trees | M5P | 0,9994 | 0,9995 | 0,999 | 0,9993 |
| | RandomForest | 0,9993 | 0,9957 | 0,9817 | 0,9922 |
| Valor medio | | 0,9961 | 0,9667 | 0,9959 | 0,9863 |



4.2.3. Ficheros 1: Prueba 3.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 23 atributos y utilizaremos los 7 mejores clasificadores obtenidos en la prueba anterior.

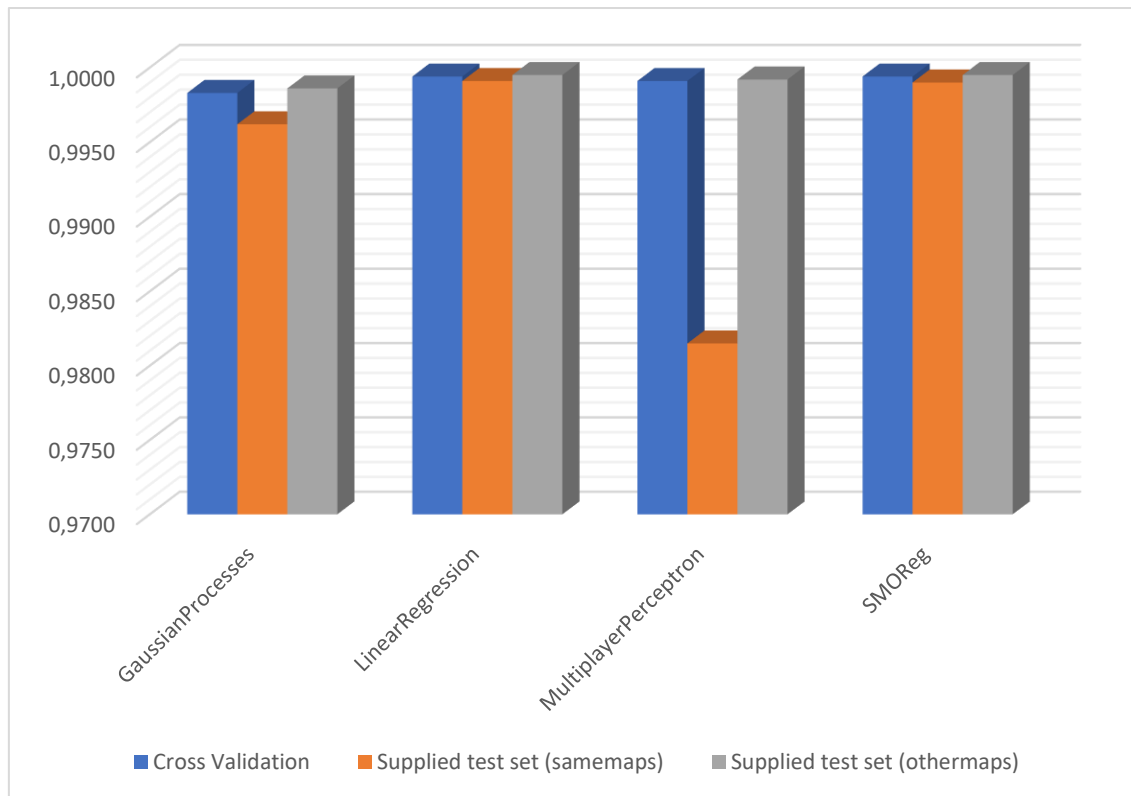
| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|-------------|----------------------|------------------|------------------------------|-------------------------------|-------------|
| Funtions | GaussianProcesses | 0,9984 | 0,9987 | 0,9967 | 0,9979 |
| | LinearRegression | 0,9994 | 0,9995 | 0,9991 | 0,9993 |
| | MultilayerPerceptron | 0,9987 | 0,9986 | 0,993 | 0,9968 |
| | SMOReg | 0,9994 | 0,9995 | 0,999 | 0,9993 |
| | RandomSubSpace | 0,9968 | 0,9954 | 0,9587 | 0,9836 |
| Trees | M5P | 0,9945 | 0,9994 | 0,999 | 0,9976 |
| | RandomForest | 0,9988 | 0,9923 | 0,95 | 0,9804 |
| Valor medio | | 0,9980 | 0,9851 | 0,9976 | 0,9936 |



4.2.4. Ficheros 1: Prueba 4.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 27 atributos y utilizaremos los 4 mejores clasificadores obtenidos en la prueba anterior.

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|-------------|----------------------|------------------|------------------------------|-------------------------------|-------------|
| Funtions | GaussianProcesses | 0,9983 | 0,9986 | 0,9962 | 0,9977 |
| | LinearRegression | 0,9994 | 0,9995 | 0,9991 | 0,9993 |
| | MultilayerPerceptron | 0,9991 | 0,9992 | 0,9815 | 0,9933 |
| | SMOReg | 0,9994 | 0,9995 | 0,999 | 0,9993 |
| Valor medio | | 0,9991 | 0,9940 | 0,9992 | 0,9974 |

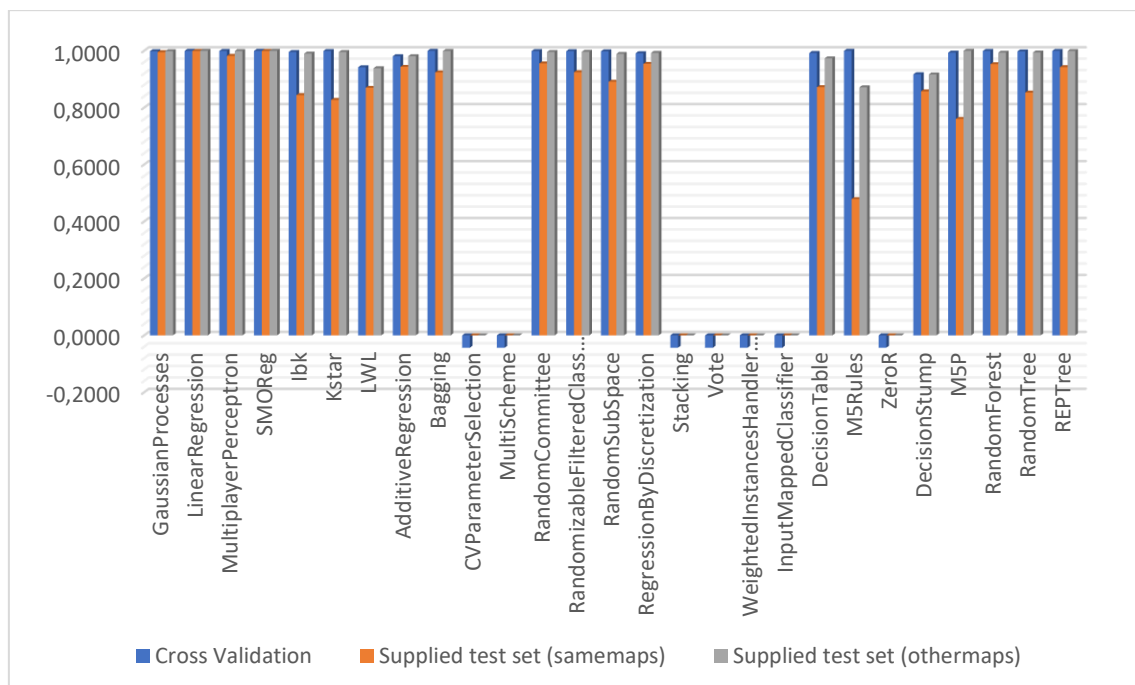


4.2.5. Ficheros 2: Prueba 1.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 35 atributos y utilizaremos los 27 clasificadores disponibles en Weka

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|----------|--------------------------------|------------------|------------------------------|-------------------------------|-------------|
| Funtions | GaussianProcesses | 0,9981 | 0,9983 | 0,9949 | 0,9971 |
| | LinearRegression | 0,9995 | 0,9995 | 0,9991 | 0,9994 |
| | MultilayerPerceptron | 0,9991 | 0,9987 | 0,9817 | 0,9932 |
| | SMOReg | 0,9994 | 0,9995 | 0,999 | 0,9993 |
| Lazy | Ibk | 0,9952 | 0,9897 | 0,8438 | 0,9429 |
| | KStar | 0,9989 | 0,9954 | 0,8272 | 0,9405 |
| | LWL | 0,9417 | 0,9385 | 0,8695 | 0,9166 |
| Meta | AdditiveRegression | 0,9805 | 0,9804 | 0,9432 | 0,9680 |
| | Bagging | 0,9994 | 0,999 | 0,9238 | 0,9741 |
| | CVPParameterSelection | -0,0436 | 0 | 0 | -0,0145 |
| | MultiScheme | -0,0436 | 0 | 0 | -0,0145 |
| | RandomCommittee | 0,9986 | 0,9955 | 0,9551 | 0,9831 |
| | RandomizableFilteredClassifier | 0,9985 | 0,996 | 0,9245 | 0,9730 |
| | RandomSubSpace | 0,9978 | 0,9887 | 0,8909 | 0,9591 |
| | RegressionByDiscretization | 0,991 | 0,9922 | 0,9538 | 0,9790 |
| | Stacking | -0,0436 | 0 | 0 | -0,0145 |

| | | | | | |
|--------------------|--|---------|--------|--------|---------|
| | Vote | -0,0436 | 0 | 0 | -0,0145 |
| | WeightedInstancesHandlerWrapper | -0,0436 | 0 | 0 | -0,0145 |
| Misc | InputMappedClassifier | -0,0436 | 0 | 0 | -0,0145 |
| Rules | DecisionTable | 0,9918 | 0,973 | 0,8717 | 0,9455 |
| | M5Rules | 0,9994 | 0,8713 | 0,4787 | 0,7831 |
| | ZeroR | -0,0436 | 0 | 0 | -0,0145 |
| Trees | DecisionStump | 0,9174 | 0,9165 | 0,8571 | 0,8970 |
| | M5P | 0,9931 | 0,9995 | 0,7602 | 0,9176 |
| | RandomForest | 0,9991 | 0,993 | 0,9525 | 0,9815 |
| | RandomTree | 0,9975 | 0,9936 | 0,8527 | 0,9479 |
| | REPTree | 0,9993 | 0,9989 | 0,9421 | 0,9801 |
| Valor medio | | 0,7219 | 0,6601 | 0,7266 | 0,7028 |

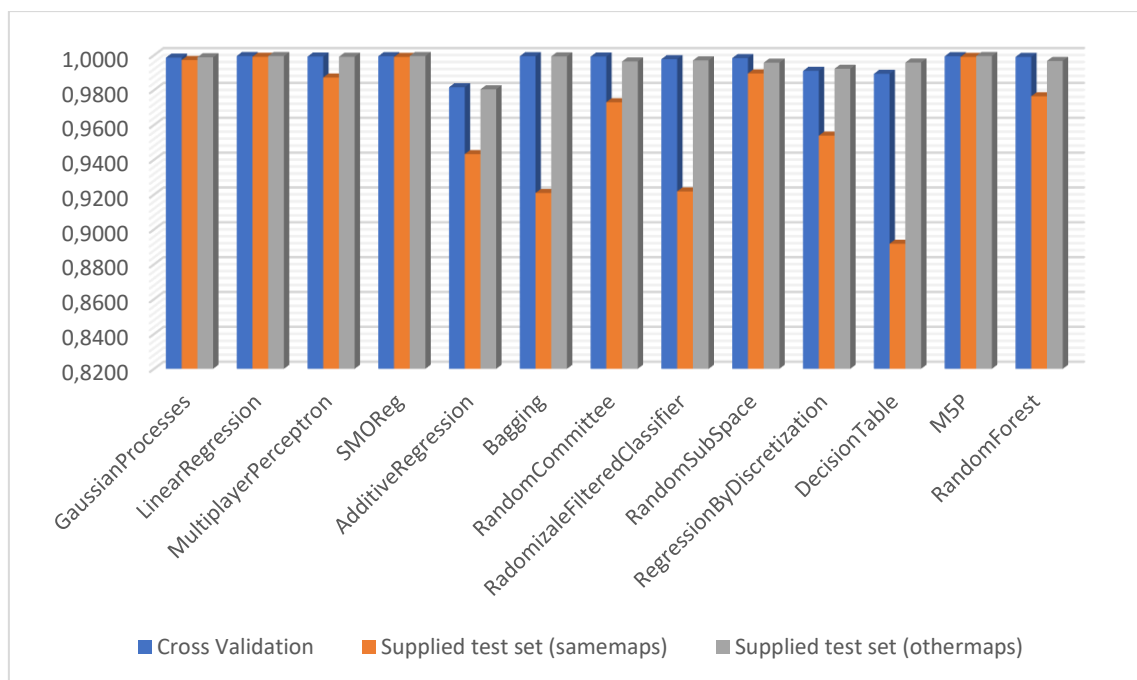


4.2.6. Ficheros 2: Prueba 2.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 23 atributos y utilizaremos los 13 mejores clasificadores obtenidos en la prueba anterior.

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|-----------------|-----------------------------|-------------------------|-------------------------------------|--------------------------------------|--------------------|
| Funtions | GaussianProcesses | 0,9993 | 0,9989 | 0,9421 | 0,9982 |
| | LinearRegression | 0,9986 | 0,9988 | 0,9972 | 0,9994 |
| | MultilayerPerceptron | 0,9995 | 0,9995 | 0,9991 | 0,9952 |
| | SMOReg | 0,9992 | 0,9991 | 0,9872 | 0,9993 |
| Meta | AdditiveRegression | 0,9994 | 0,9995 | 0,999 | 0,9684 |

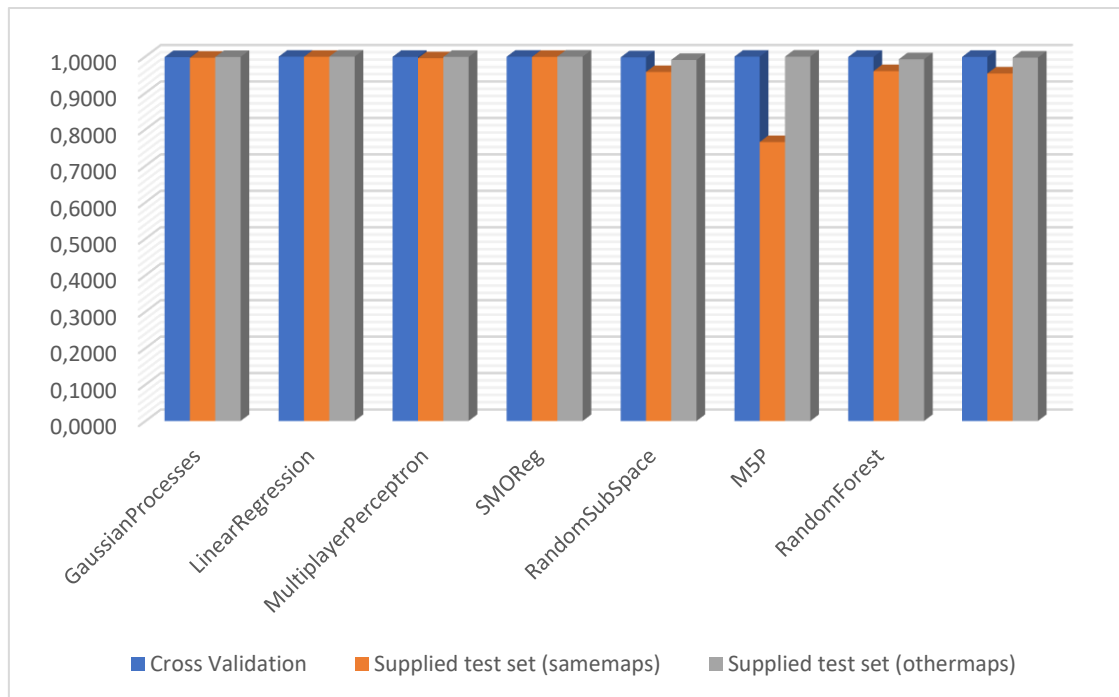
| | | | | | |
|--------------------|---------------------------------------|--------|--------|--------|--------|
| | Bagging | 0,9816 | 0,9804 | 0,9432 | 0,9732 |
| | RandomCommittee | 0,9994 | 0,9993 | 0,9209 | 0,9895 |
| | RandomizableFilteredClassifier | 0,9992 | 0,9964 | 0,973 | 0,9722 |
| | RandomSubSpace | 0,9977 | 0,997 | 0,9218 | 0,9945 |
| | RegressionByDiscretization | 0,9983 | 0,9957 | 0,9895 | 0,9790 |
| Rules | DecisionTable | 0,991 | 0,9922 | 0,9538 | 0,9590 |
| Trees | M5P | 0,9893 | 0,9958 | 0,8918 | 0,9993 |
| | RandomForest | 0,9994 | 0,9995 | 0,999 | 0,9907 |
| Valor medio | | 0,9963 | 0,9655 | 0,9961 | 0,9860 |



4.2.7. Ficheros 2: Prueba 3.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 23 atributos y utilizaremos los 7 mejores clasificadores obtenidos en la prueba anterior.

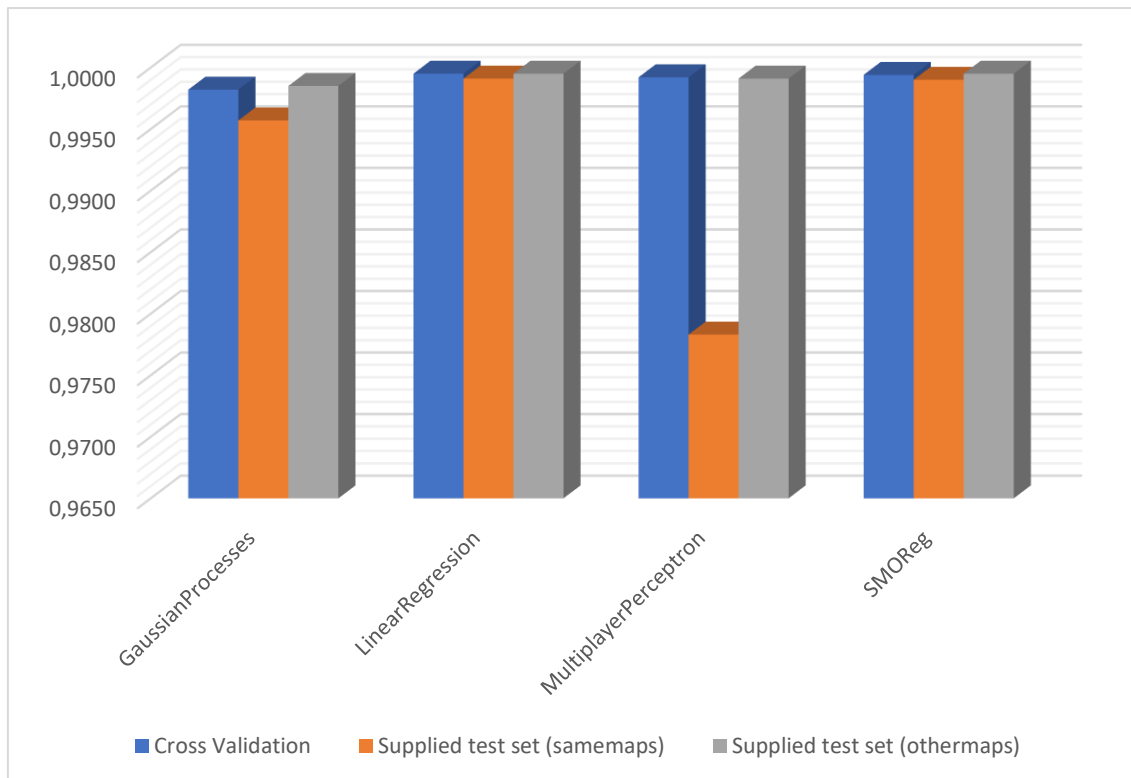
| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|--------------------|-----------------------------|-------------------------|-------------------------------------|--------------------------------------|--------------------|
| Funtions | GaussianProcesses | 0,9983 | 0,9986 | 0,9966 | 0,9978 |
| | LinearRegression | 0,9994 | 0,9995 | 0,9991 | 0,9993 |
| | MultilayerPerceptron | 0,999 | 0,9988 | 0,9951 | 0,9976 |
| | SMOReg | 0,9994 | 0,9995 | 0,999 | 0,9993 |
| | RandomSubSpace | 0,9975 | 0,9904 | 0,9573 | 0,9817 |
| Trees | M5P | 0,9994 | 0,9995 | 0,7653 | 0,9214 |
| | RandomForest | 0,9988 | 0,9924 | 0,9597 | 0,9836 |
| Valor medio | | 0,9988 | 0,9532 | 0,9970 | 0,9830 |



4.2.8. Ficheros 2: Prueba 4.

Esta primera prueba se realizará con todos los atributos generados en el fichero de instancias es decir, 27 atributos y utilizaremos los 4 mejores clasificadores obtenidos en la prueba anterior.

| | | Cross Validation | Supplied test set (samemaps) | Supplied test set (othermaps) | Valor medio |
|-------------|----------------------|------------------|------------------------------|-------------------------------|-------------|
| Funtions | GaussianProcesses | 0,9988 | 0,9924 | 0,9597 | 0,9975 |
| | LinearRegression | 0,9982 | 0,9985 | 0,9957 | 0,9994 |
| | MultilayerPerceptron | 0,9995 | 0,9995 | 0,9991 | 0,9922 |
| | SMOReg | 0,9992 | 0,9991 | 0,9783 | 0,9993 |
| Valor medio | | 0,9991 | 0,9930 | 0,9992 | 0,9971 |



4.3. *Análisis final*

En primer lugar, para la realización de esta práctica se han tenido en cuenta muchas pruebas previas que no aparecen reflejadas en el documento debido a la extensión de las mismas. Se ha comenzado con el grupo de atributos más grande que podemos aportar, es decir, todos los atributos posibles utilizados tanto en el tutorial 1 como en los ficheros de instancias generados por teclado al igual que se han utilizado todos los algoritmos posibles para tener el mayor conjunto de pruebas posible y así poder obtener resultados mejores. Una vez conseguido el número total de atributos, que son los que aparecen reflejados a la largo de este documento, se realizaron las pruebas anteriores (apartado 4.1 y 4.2), en las cuales se han ido reduciendo los conjuntos de atributos a la mitad eligiendo en cada conjunto los algoritmos cuyo coeficiente de correlación medio obtenido sean los mejores en cada caso.

Tras haber realizado las pruebas necesarias en la fase 2 hemos decidido utilizar los mismos atributos ya que eran los que maximizaban los porcentajes de acierto en este caso. Por esta razón, a pesar de intentar mejorar constantemente el porcentaje de instancias clasificadas correctamente, nos hemos basado en lo que consideramos que aporta el atributo al modelo para elegir los adecuados. Además, en la primera prueba como ya hemos comentado anteriormente hemos decidido mantener todos los atributos tal cual se generaron para tener el mayor rango de pruebas posibles. En la segunda prueba se ha decidido quitar los atributos de posición, aquellos que indican la fila y la columna en la que se encuentran el pacman y los fantasmas en el estado actual, ya que podrían variar en función de las dimensiones del mapa e incluso no existir, por lo que podrían generar ruido, provocando que el agente automático no realice las acciones correctas. Por todo esto se decidió, quitarlos y demostrar que no mostraban gran relevancia a la hora de la predicción de datos. En la tercera prueba, decidimos obviar el cálculo de distancias ya que a pesar de ser relevantes permiten saber si un fantasma está lejos o cerca del pacman y por tanto permitirán que el agente automático sea eficiente ya que tendría que ir primero a por el fantasma más cercano. Por todo esto, esta prueba servirá para comprobar

la repercusión de estos atributos en la predicción de datos. La última prueba de predicción realizada tiene que ver principalmente con los atributos representativos de las paredes, en esta prueba se obvian los atributos que miden las distancias a las paredes más cercanas del pacman y también a la comprobación de si hay una pared a continuación del pacman, ya que consideramos que tienen relevancia para él en el proceso de realizar el siguiente movimiento. En el apartado 4.2, se han realizado las mismas pruebas que en el caso anterior utilizando los ficheros de la fase 1 y unos nuevos ficheros generados en los cuales se han eliminado las instancias que contenía "Stop" como acción. Esta decisión ha sido tomada ya que consideramos que estas instancias eliminadas servirían únicamente para realizar la acción "Stop" y haciendo que el agente automático en determinados momentos se detenga.

Finalmente, tras haber realizado todas las pruebas pertinentes hemos llegado a la conclusión de que los mejores algoritmos de predicción utilizados son: "LinearRegression" y "SMOReg" ya que para todas las pruebas realizadas obtenían los valores más altos del coeficiente de correlación (que mide la correlación estadística entre los datos predichos y los datos reales). Cabe destacar que a pesar de que la práctica pide denotar los dos mejores algoritmos de predicción para la puntuación de la ejecución del juego, los mejores resultados de las pruebas con Weka vienen dados únicamente del algoritmo "LinearRegression" en las pruebas 1, 2 y 4 para el fichero del tutorial 1 y con la ejecución de "Cross validation" aunque los resultados de SMOReg son los segundos mejores detrás del ya mencionado anteriormente.

5. Fase 4: Construcción de un Agente Automático

Para la realización de esta fase se han generado modelos a partir de los resultados obtenidos en las fases 2 y 3.

El objetivo de esta fase es conseguir un agente automático que realice las acciones adecuadas para la ejecución más eficiente del mapa en el que se encuentre. Para ello es necesario generar un modelo que se adapte a estas condiciones. A partir de los resultados obtenidos en las fases 2 y 3 se han generado diversos modelos para comprobar cual es el mejor. En primer lugar hemos seleccionado las combinaciones de algoritmos, pruebas y ficheros que mayores porcentajes de instancias clasificadas correctamente presentan. Los modelos seleccionados serían los siguientes:

Modelos generados a partir de los ficheros de entrenamiento del tutorial 1 (Clasificación).

| Nombre | Ficheros | Prueba | Algoritmo | Opción |
|---------------|----------|--------|--------------|------------------------------|
| RF_1_1.model | 1 | 1 | RandomForest | Cross Validation |
| RF_1_2.model | 1 | 2 | RandomForest | Supplied test set (samemaps) |
| RF_1_4.model | 1 | 4 | RandomForest | Supplied test set (samemaps) |
| J48_1_4.model | 1 | 4 | J48 | Supplied test set (samemaps) |
| RF_2_1.model | 2 | 1 | RandomForest | Supplied test set (samemaps) |
| RF_2_2.model | 2 | 2 | RandomForest | Cross Validation |
| RF_2_4.model | 2 | 4 | RandomForest | Cross Validation |
| J48_2_4.model | 2 | 4 | J48 | Cross Validation |

Modelos generados a partir de los ficheros de entrenamiento controlado por el teclado (Clasificación).

| Nombre | Ficheros | Prueba | Algoritmo | Opción |
|----------------|----------|--------|--------------|-------------------------------|
| RF_1_3_1.model | 1 | 3 | RandomForest | Cross Validation |
| RF_1_3_2.model | 1 | 3 | RandomForest | Supplied test set (samemaps) |
| RF_1_3_3.model | 1 | 3 | RandomForest | Supplied test set (othermaps) |
| RF_2_3_1.model | 2 | 3 | RandomForest | Cross Validation |
| RF_2_3_2.model | 2 | 3 | RandomForest | Supplied test set (othermaps) |
| J48_2_3.model | 2 | 3 | J48 | Supplied test set (othermaps) |

Modelos generados a partir de los ficheros de entrenamiento de la fase de predicción.

| Nombre | Ficheros | Prueba | Algoritmo | Opción |
|------------|-----------|--------|------------------|------------------|
| LR_1.model | Tutorial1 | 1 | LinearRegression | Cross Validation |
| LR_2.model | Tutorial1 | 2 | LinearRegression | Cross Validation |
| LR_4.model | Tutorial1 | 4 | LinearRegression | Cross Validation |

Finalmente hemos seleccionado el modelo perteneciente al tutorial 1 de la fase de clasificación, a los ficheros "Ficheros 1", prueba 4, algoritmo J48.

6. Preguntas

A continuación se responderán a las preguntas propuestas en la práctica.

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

Las diferencias a la hora de aprender modelos con instancias provenientes de un agente controlado por un humano y uno automático se basan principalmente en las ventajas e inconvenientes que presentan cada uno de ellos, ya que aunque para ciertas cosas puede ser mejor controlar el agente, en otras podría ser más eficiente el agente automático. A continuación se mostrará una tabla en la cual se podrán apreciar con mayor claridad las diferencias.

| | Agente automático | Control humano |
|-----------------------|--|---|
| Ventajas | Un agente automático correctamente programado conocería el mapa en el que se encuentra perfectamente, sabiendo donde se encuentran las paredes y los fantasmas, las distancias para llegar a ellos y el camino más corto. Además, esto se cumpliría tanto para fantasmas fijos como para fantasmas en movimiento, por lo que no se generaría instancias que generen ruido en el modelo. | Un humano sabe reaccionar a las distintas situaciones en las que se pueda encontrar el pacman, evitando paredes y sabiendo a simple vista como llegar a los fantasmas, por lo que no existiría la posibilidad de entrar en bucle. |
| Inconvenientes | No resulta sencillo programar un buen agente automático, por lo que no siempre sabrá que hacer. En muchos casos podría entrar en bucle, no sabría evitar paredes por lo que si un fantasma se encuentra muy "escondido" no podría alcanzarlo y nunca terminaría. Además, al introducir movimientos aleatorios para evitar los bucles, se realizarían acciones sin sentido que dificultarían el correcto funcionamiento del agente. Como ejemplo podríamos tener nuestro agente automático que sigue un modelo que no siempre genera las acciones correctas ya que al predecir una acción ilegal realizamos un movimiento aleatorio. | A simple vista un humano no puede calcular con precisión la distancia a la que se encuentran los fantasmas para ir a por el más cercano. Además en el caso de que estos se encuentren en movimiento le resultaría complicado alcanzarlos, pudiendo realizar muchos movimientos incorrectos que generarían ruido en el modelo. Por otro lado el tiempo de reacción, por ejemplo, al principio de las partidas, generaría instancias que futuramente podría predecir una acción de "Stop", la cual no aportaría eficiencia. |

2. Si quisieras transformar la tarea de regresión en clasificación ¿Qué tendrías que hacer?
¿Cuál crees que podría ser la aplicación práctica de predecir la puntuación?
3. ¿Qué ventajas puede aportar predecir la puntuación respecto a la clasificación de la acción? Justifica tu respuesta.
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?

7. Conclusiones y dificultades

Con la realización de esta práctica hemos podido comprender con mayor claridad en que consisten y cómo funcionan las técnicas de aprendizaje automático que se han ido viendo a lo largo del desarrollo de la asignatura; además, a pesar de llevar mucho tiempo y trabajo nos ha resultado interesante el enseñar al Pac-Man a jugar correctamente.

En cuanto a los problemas encontrados se podría decir que la práctica consiste en conseguir un agente automático que juegue de la mejor forma posible y no lo hemos conseguido. Sin embargo, consideramos que dicho problema se puede deber a la falta de entrenamiento del Pac-Man y a los errores que se han ido cometiendo al generar las instancias de las cuales dependen la clasificación y la predicción. Inicialmente, el agente programado en el tutorial 1 muchas veces realizaba acciones de forma aleatoria por lo que entraba en bucle en la mayoría de los mapas. Durante la realización de esta práctica hemos conseguido mejorar el control del agente utilizando nuevos métodos y funciones pertenecientes al código inicial que desconocíamos.

NOTA: En la carpeta “Código” hemos añadido el fichero game.py ya que desde el “main” se llama a la función “printLineData” en la fase 1, mientras que en la 4 no estaría contemplado este caso.