



Imagens da execução de algoritmos de aprendizado de máquina sobre o dataset selecionado usando os frameworks WEKA e MOA

Aluno: Leonardo Matsumoto

Disciplina: Aprendizado de Máquina 2020.1 IC/UFF

1. Distribuição dos dados

Conjunto de dados:

Relation: csh101.integral.arff

Instances: 321428

Attributes: 37

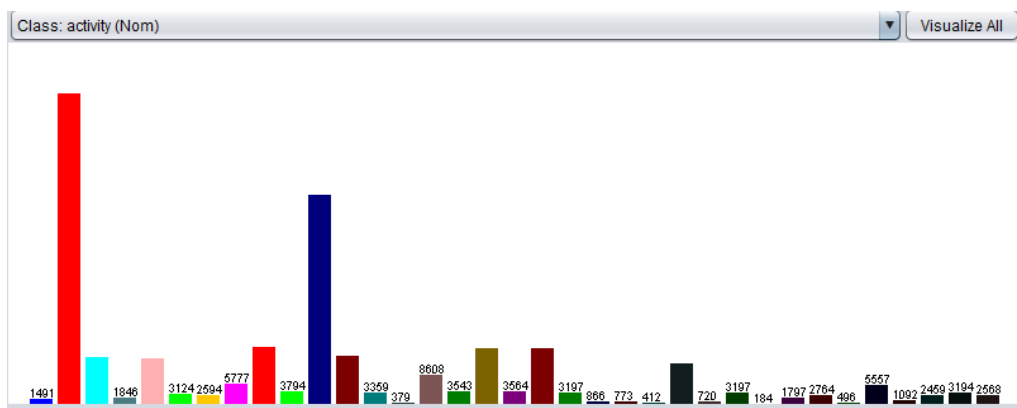


Figura 1. Distribuição dos dados no atributo-alvo (activity)

Classe majoritária: Other_Activity (90824 exemplos).

Classe minoritária: Work_At_Table (184 exemplos).

Tabela 1. Frequência de cada classe:

activity	exemplos	cia
Other_Activity	90824	28,26%
Watch_TV	61312	19,07%
Sleep_Out_Of_Bed	16618	5,17%
Bathe	16289	5,07%
Cook_Breakfast	16279	5,06%
Dress	14051	4,37%
Toilet	13727	4,27%
Personal_Hygiene	13221	4,11%
Sleep	11752	3,66%
Read	8608	2,68%
Relax	5777	1,80%
Cook_Dinner	5557	1,73%
Drink	3794	1,18%
Eat_Breakfast	3564	1,11%
Morning_Meds	3543	1,10%
Evening_Meds	3359	1,05%
Cook_Lunch	3197	0,99%
Wash_Breakfast_Dishes	3197	0,99%
Wash_Dishes	3194	0,99%
Leave_Home	3124	0,97%
Cook	2764	0,86%
Enter_Home	2594	0,81%
Entertain_Guests	2568	0,80%
Wash_Dinner_Dishes	2459	0,77%
Phone	1846	0,57%
Groom	1797	0,56%
Step_Out	1491	0,46%
Eat_Dinner	1092	0,34%
Eat_Lunch	866	0,27%
Wash_Lunch_Dishes	773	0,24%
Bed_Toilet_Transition	720	0,22%
Eat	496	0,15%
Go_To_Sleep	412	0,13%
Wake_Up	379	0,12%
Work_At_Table	184	0,06%

2. Imagens de algoritmos executados no WEKA

A seguir são apresentadas as capturas de tela durante a execução de cada algoritmo no WEKA.




Nome	Status	CPU	Memória
▼  Zulu Platform x64 Architecture (2)		24.4%	1'350.2 MB
 Weka GUI Chooser			
 Weka Explorer			

Figura 2. Consumo de CPU e Memória durante execução do WEKA

2.1. Algoritmo k-NN

O k-NN foi executado com $k = 1$ (1-NN) usando o algoritmo IBk:

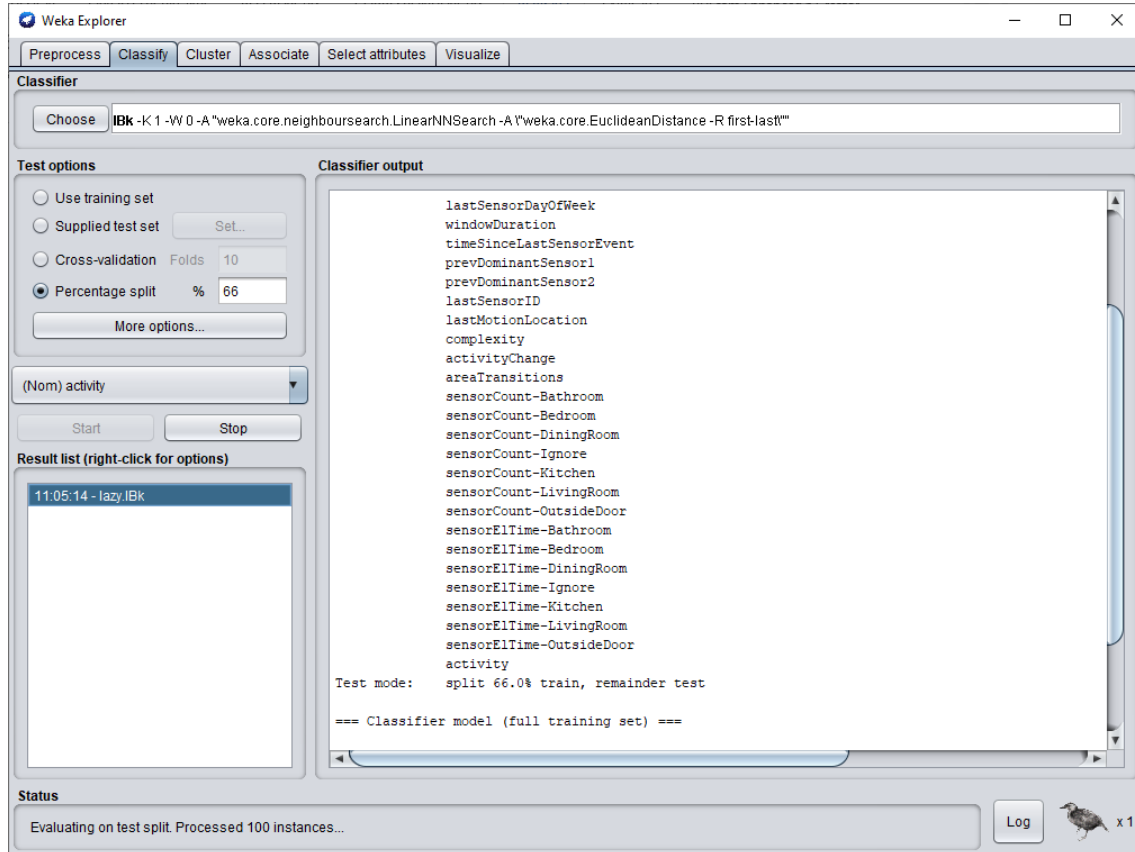


Figura 3. Execução do k-NN

2.2. Regras de Decisão

Como exemplo de regra de decisão, foi utilizado o OneR:

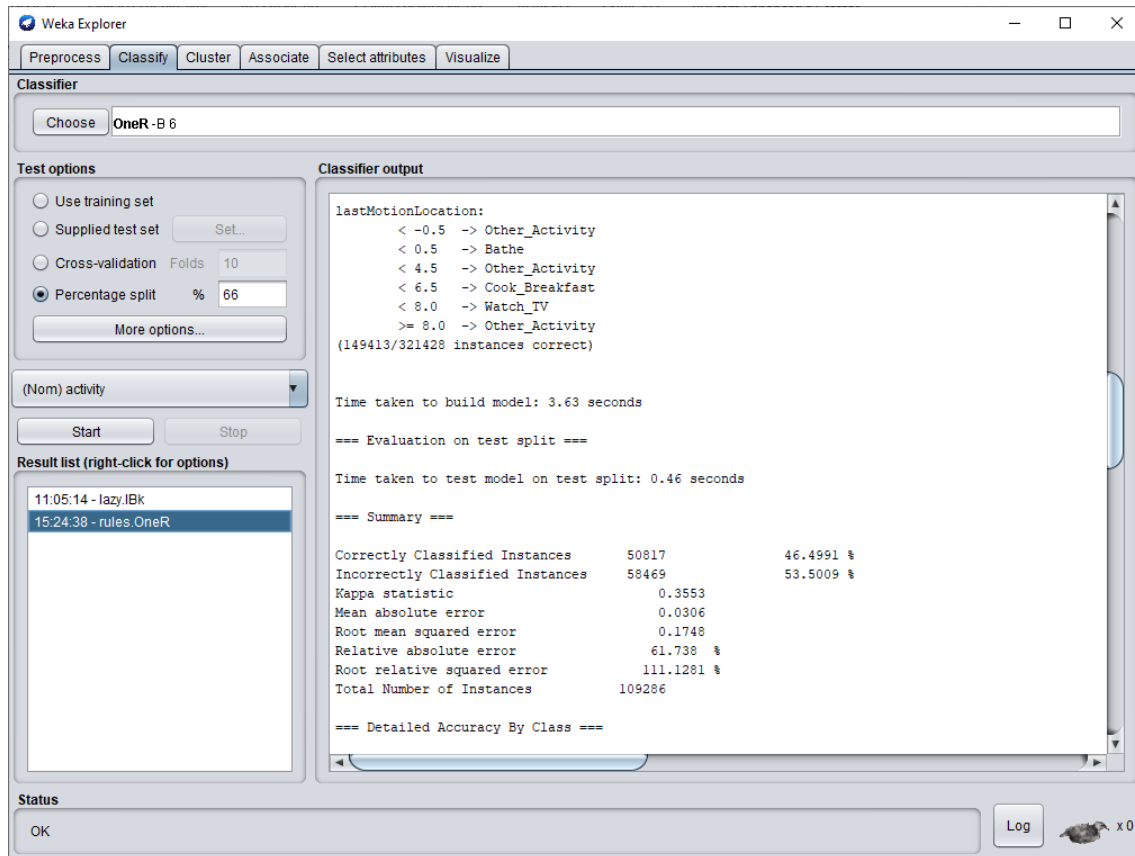


Figura 4. Execução do OneR

2.3. Árvores de Decisão

A primeira árvore foi gerada com o J48 em *holdout*:

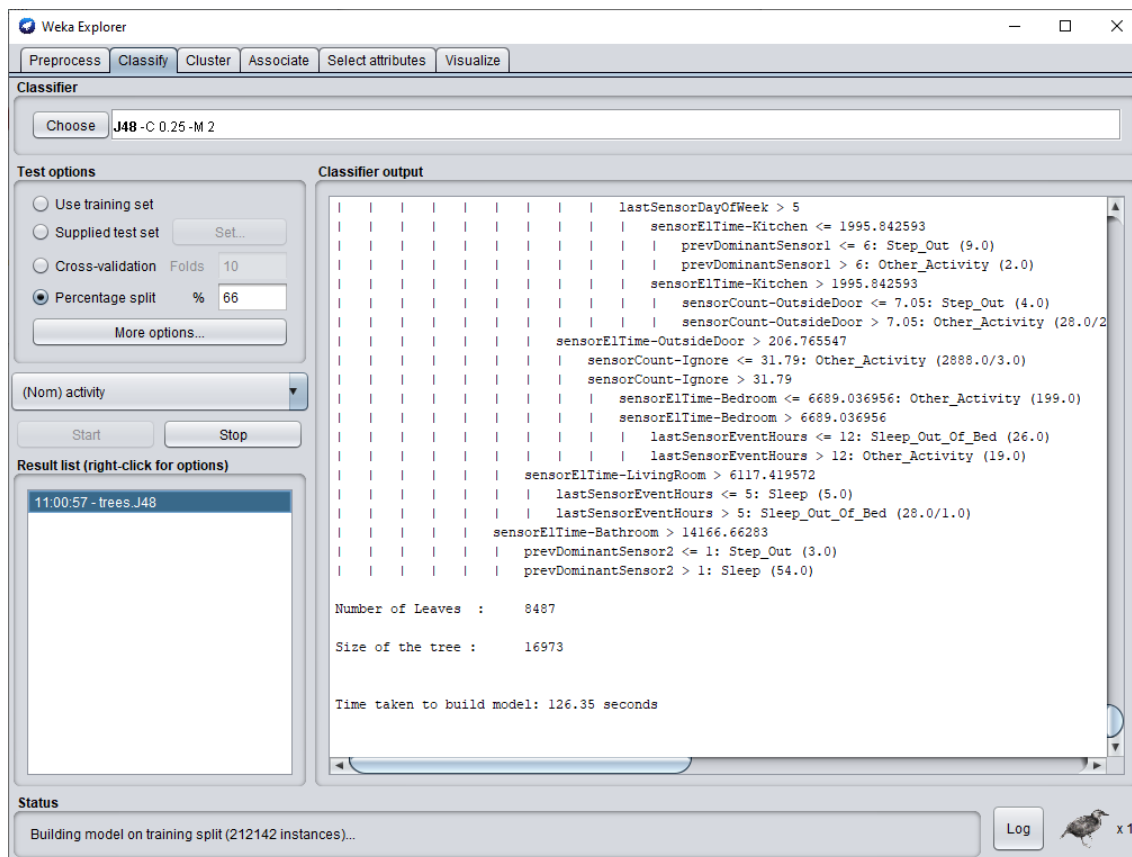


Figura 5. Execução do J48

Ao tentar executar o J48 em outra abordagem, com validação cruzada mediante k-fold e com $k = 10$, ocorreu estouro do heap de memória:

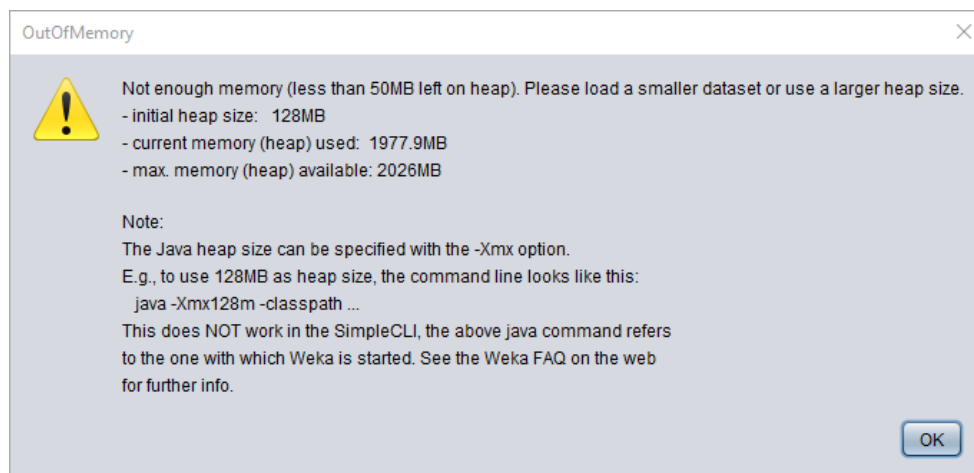


Figura 6. Exceção por falta de memória durante o 10-fold no J48

2.4. Algoritmos Bayesianos

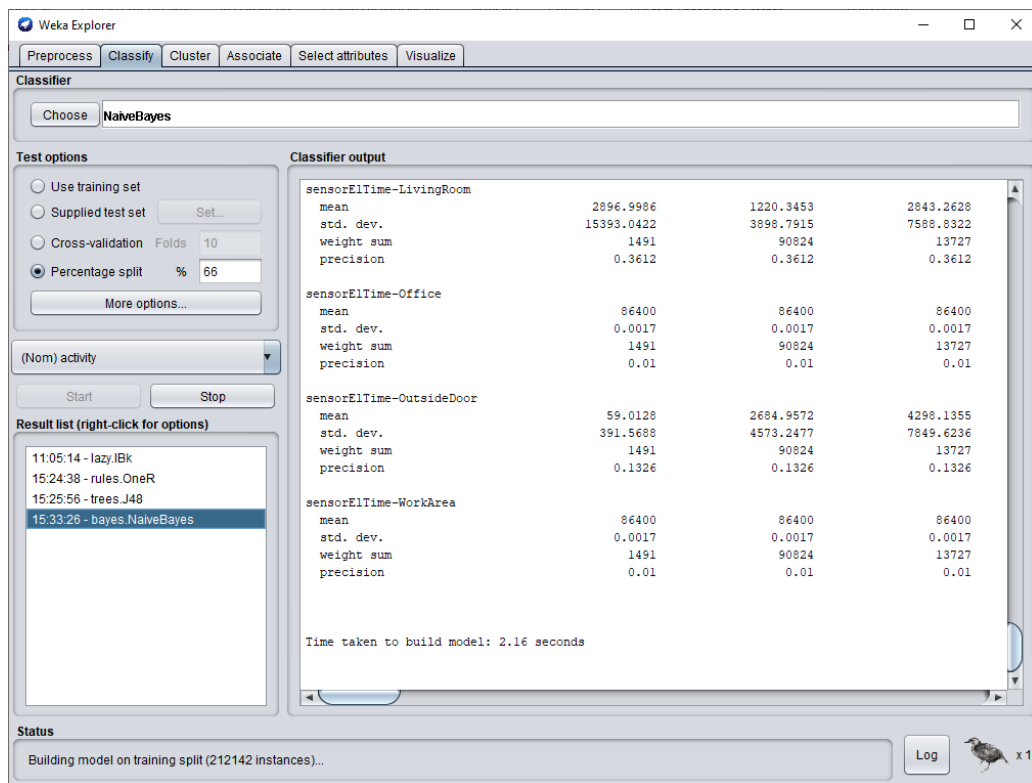


Figura 7. Execução do NaiveBayes

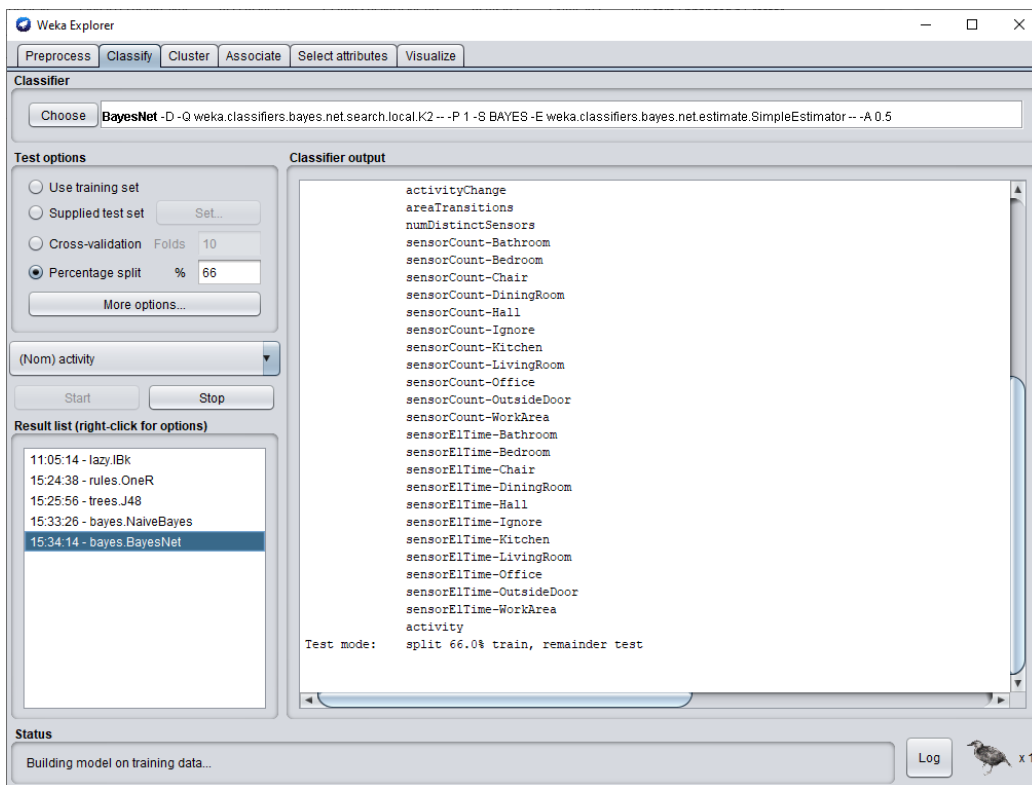


Figura 7. Execução da BayesNet

2.5. Redes neurais

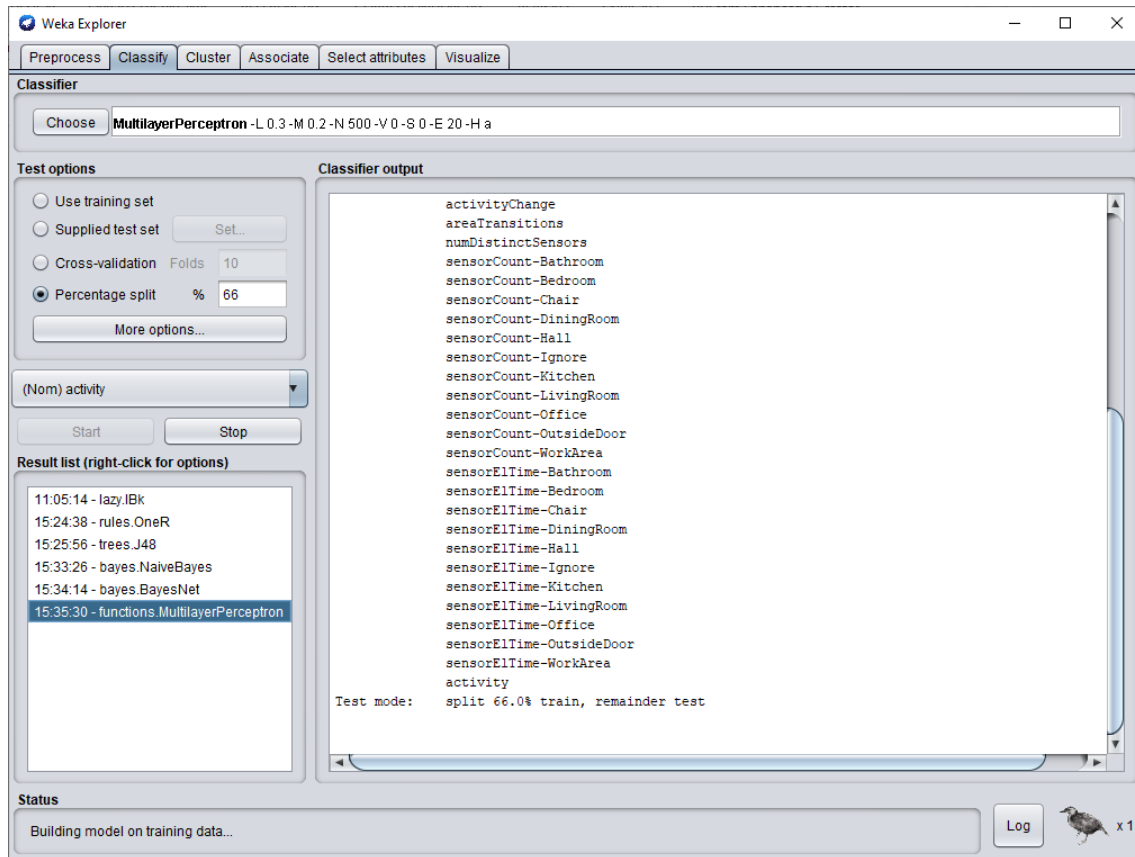


Figura 8. Execução do MultilayerPerceptron

2.6. Máquinas de Vetores de Suporte (SVM)

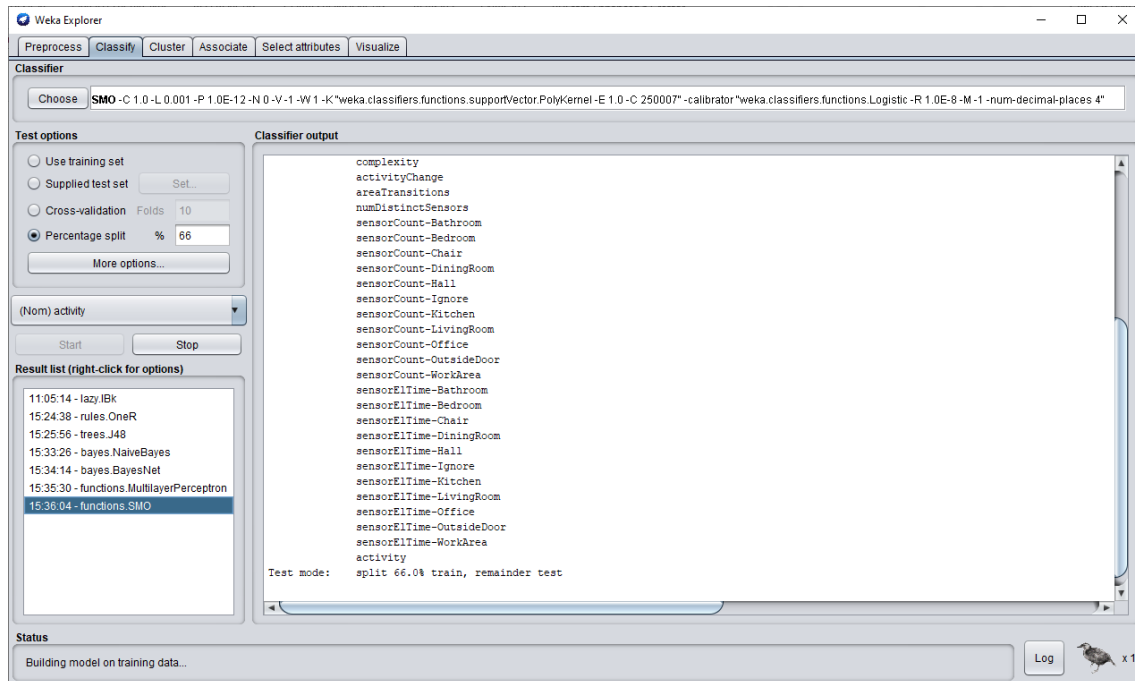


Figura 9. Execução da SMO

3. Imagens de algoritmos executados no MOA

Em seguida, partiu-se para uma estratégia de *stream learning* por meio da ferramenta MOA.

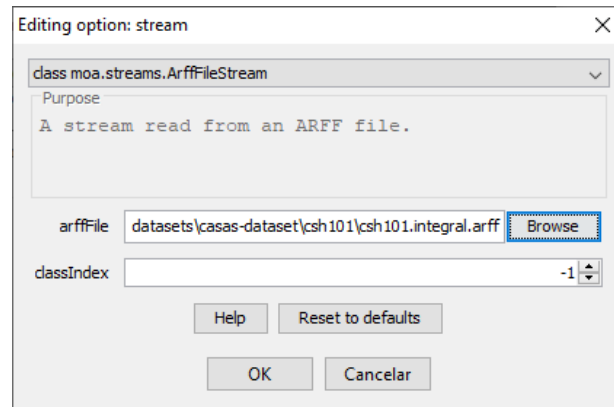


Figura 10. Escolhendo o arquivo de entrada com o conjunto de dados

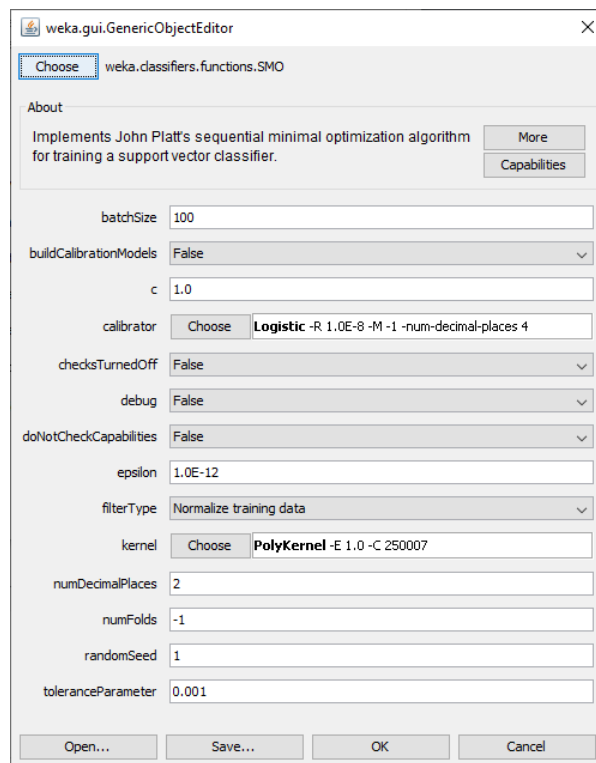


Figura 11. Configurando o SMO no MOA

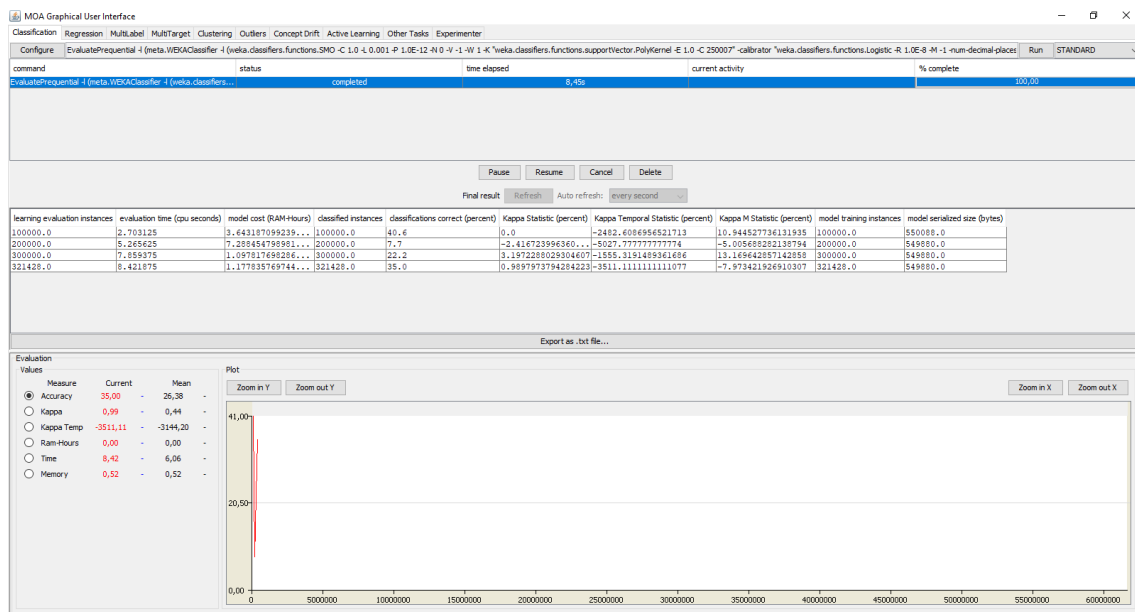


Figura 12. Executando o SMO no MOA

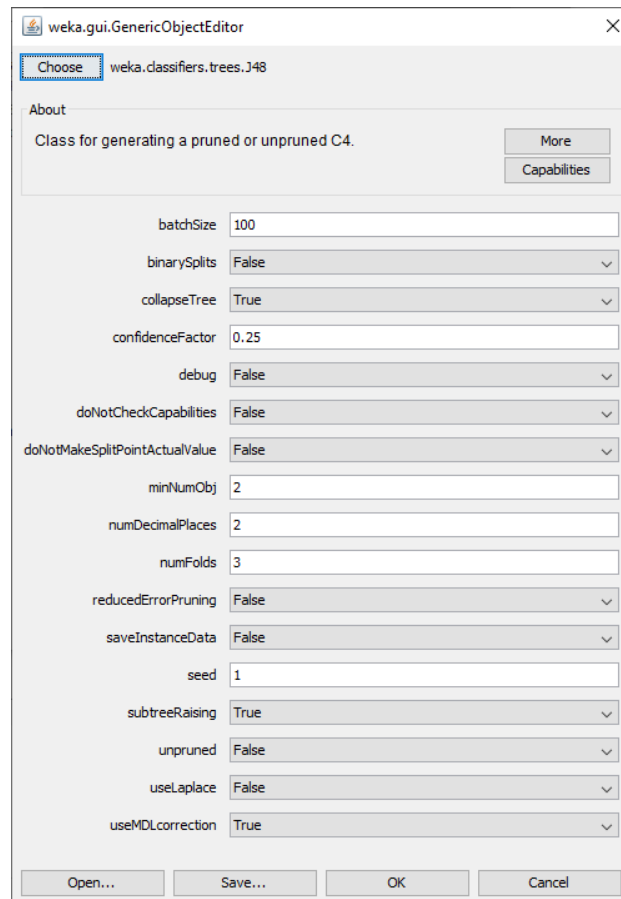


Figura 13. Configurando o J48 no MOA

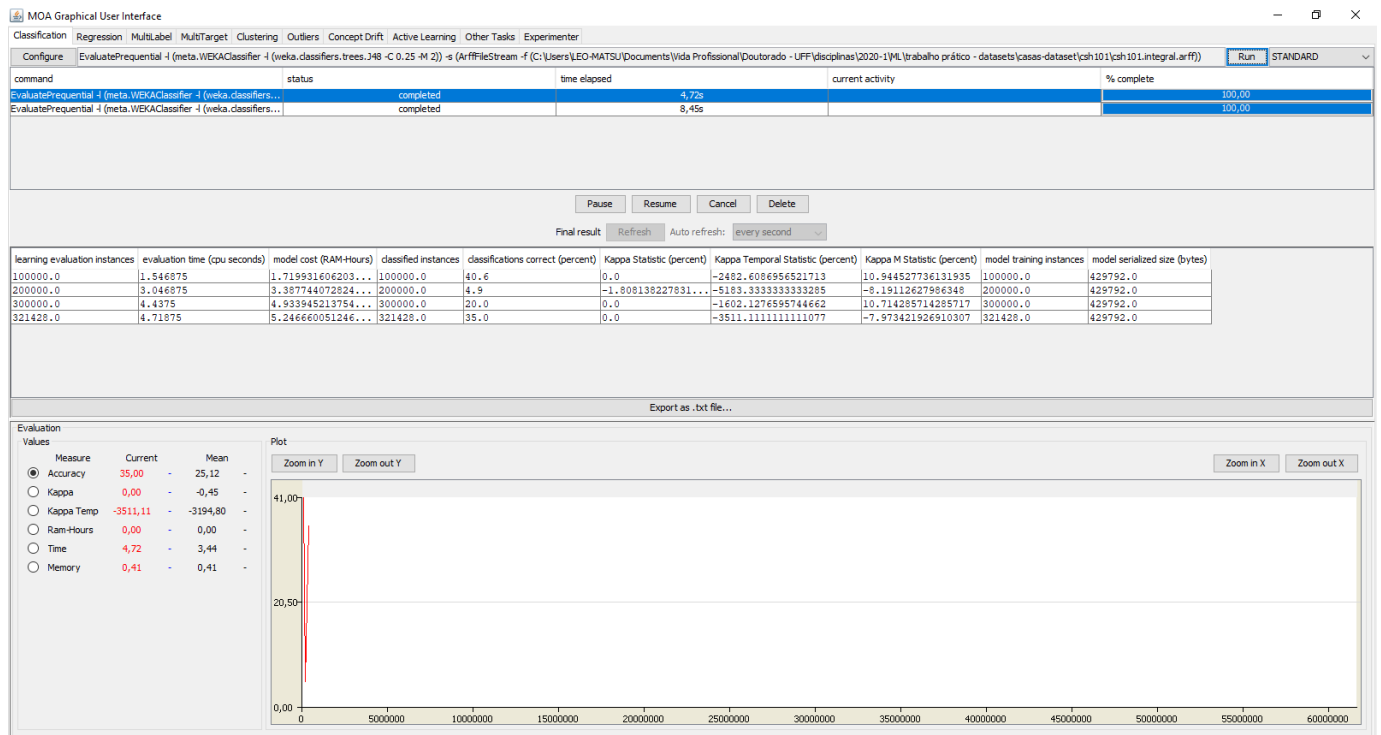


Figura 14. Executando o J48 no MOA

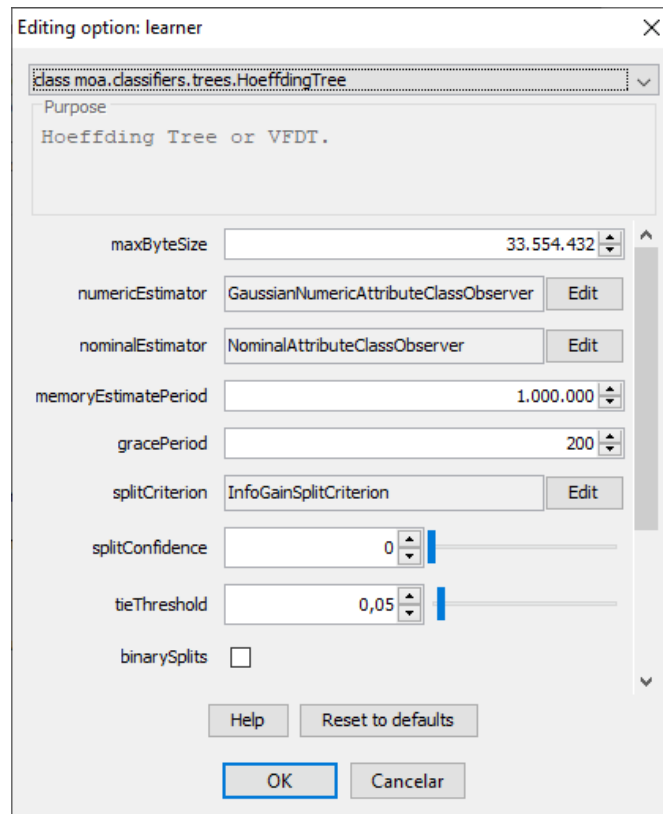


Figura 15. Configurando a HoeffdingTree no MOA

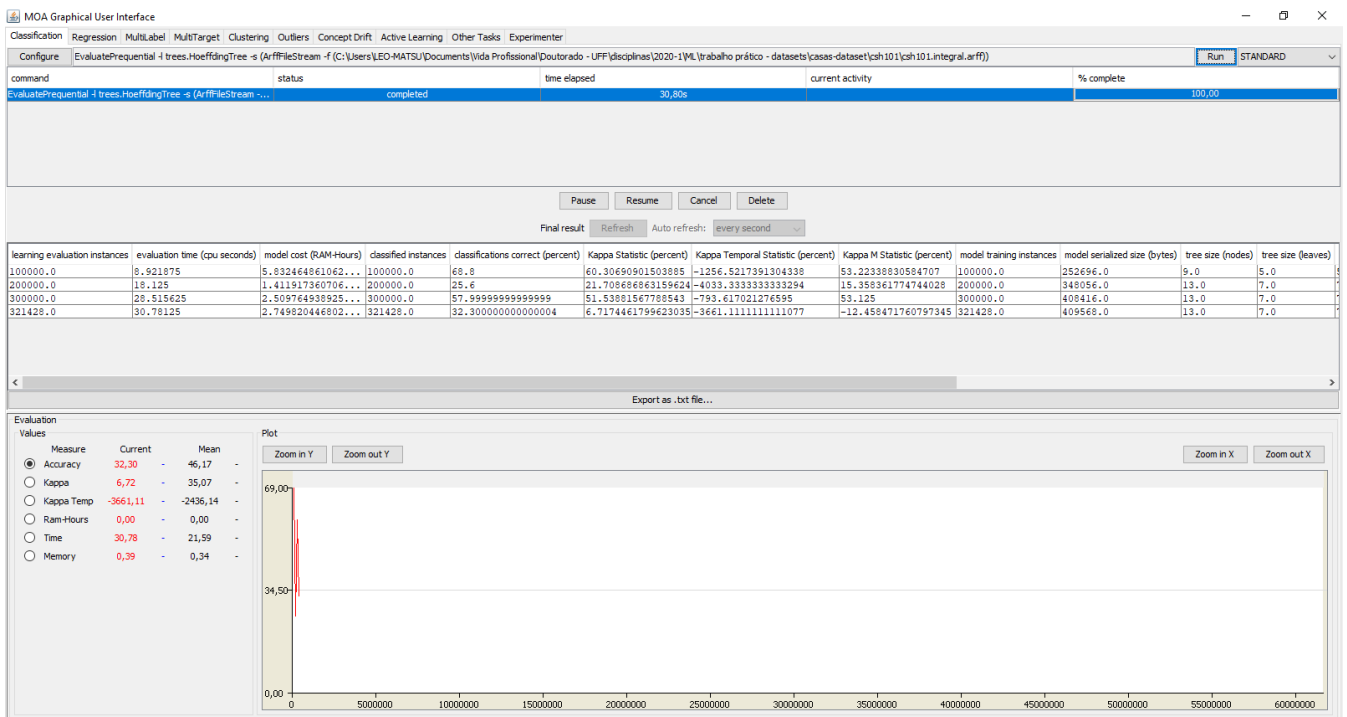


Figura 16. Executando a HoeffdingTree no MOA

Editing option: learner

class moa.classifiers.lazy.kNN

Purpose

kNN: special.

k

1

limit

1.000

nearestNeighbourSearch

LinearNN

Help

Reset to defaults

OK

Cancelar

Figura 17. Configurando o kNN

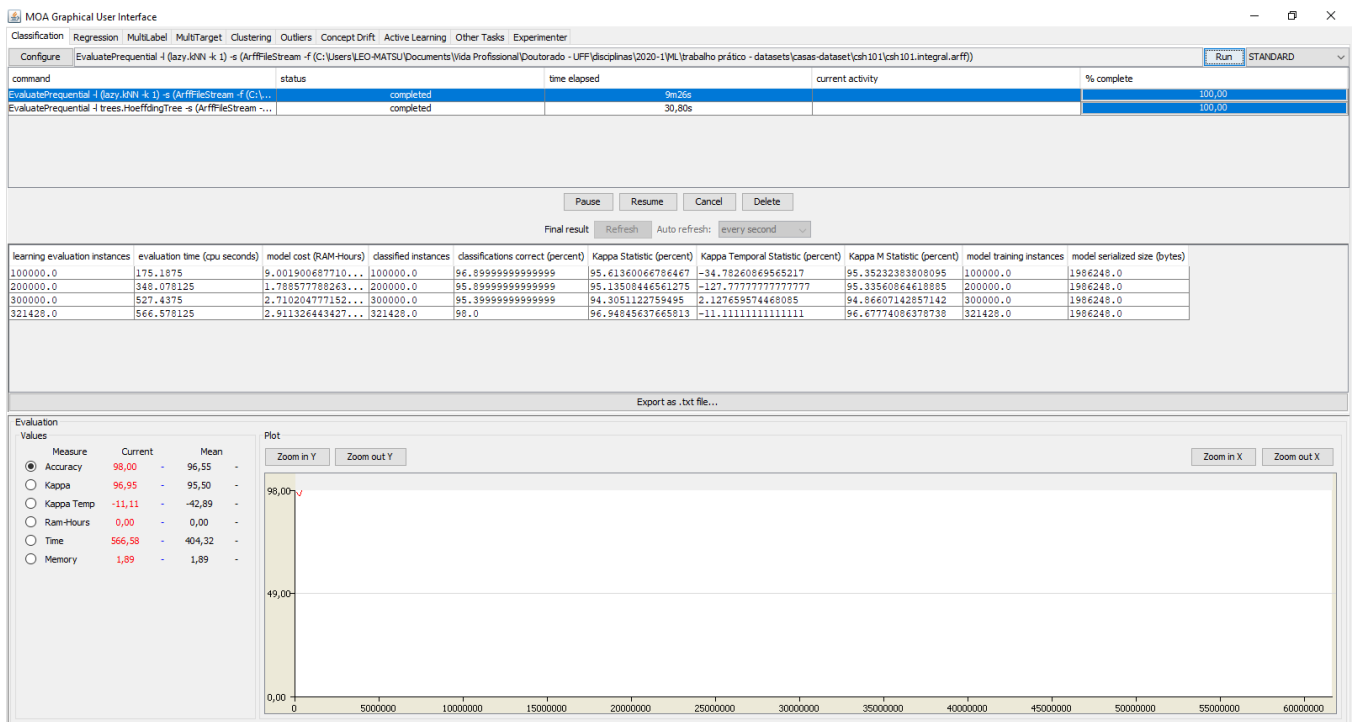


Figura 18. Executando o k-NN

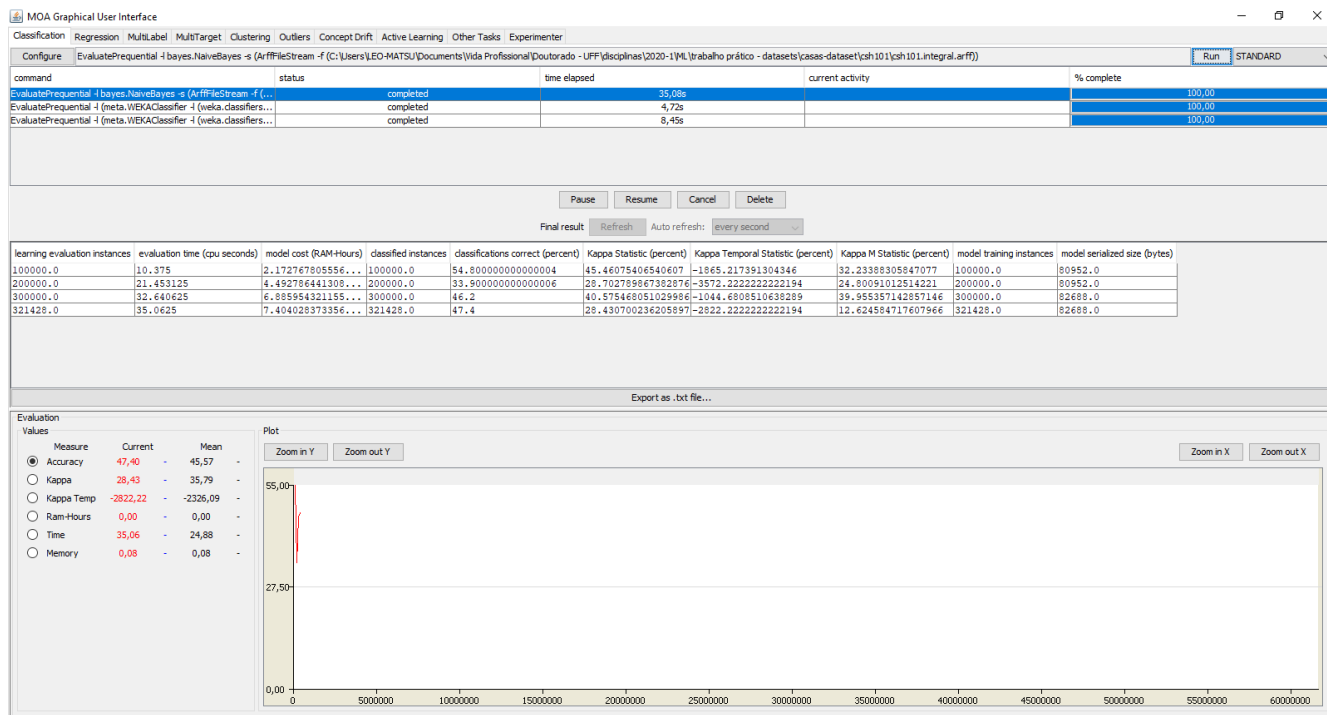


Figura 19. Executando o Naive-Bayes (não requer configuração especial)

