



Reconhecimento de Padrões

2017/2018

Traffic Sign Recognition (TSR)

André Clemêncio	2013152406
Joel Pires	2014195242

Universidade de Coimbra  
Departamento de Engenharia Informática  
Mestrado em Engenharia Informática

## Índice

Introdução	3
Normalização dos dados	3
Feature Selection	4
Feature Reduction	4
Classificadores	6
Interface	7
Resultados	9
Conclusões e Observações Finais	18
Futuros Desenvolvimentos	20
Bibliografia	20
Anexo A	22

## Introdução

TSR (Traffic Sign Recognition) é uma tecnologia muito usada para o reconhecimento de sinais de trânsito na estrada.

O nosso objetivo com este trabalho, será desenvolver classificadores para o TSR. Foi-nos fornecido um dataset com 43 sinais de trânsito diferentes representados por fotografias tiradas em ambiente real. O dataset para treino contém 39209 sinais enquanto que o de teste contém 12630. É também fornecido um conjunto de features pré-computadas.

Iremos considerar três cenários diferentes:

- **Cenário A:** consiste num problema binário. O objetivo é apenas distinguir o sinal de STOP dos restantes.
- **Cenário B:** neste caso iremos diferenciar entre seis diferentes classes de sinais, sendo elas: sinais de perigo, sinais de obrigação, sinais de proibição para o limite de velocidade, outros sinais de proibição, sinais únicos e sinais de restrição.
- **Cenário C:** aqui iremos diferenciar todos os 43 sinais.

## Normalização dos dados

Pelo facto de existirem, no dataset, variáveis medidas em escalas diferentes ou que usam diferentes valores de distribuição teremos de normalizar os dados de modo a que não exista nenhum problema na posterior análise dos mesmos. Assim sendo, permitimos que seja escolhida a opção de normalização.

A estratégia utilizado para o efeito foi pelo *zscore* onde o *m* e o *s* representam a média e o desvio padrão respectivamente.

$$y_i = \frac{x_i - m}{s}$$

Neste tipo de normalização a nova variável,  $Y_i$  neste caso, terá o valor da média a 0 e a variância unitária.

## Feature Selection

De modo a reduzir o tamanho do dados que nos são fornecidos e em que muitos deles podem ser inúteis faz-se uma seleção daqueles que serão mais relevantes para a resolução do nosso problema.

O objetivo desta seleção de dados será, não só reduzir a complexidade do nosso classificador como obter melhores resultados selecionando apenas os dados mais relevantes.

De modo a fazer esta redução podemos utilizar dois métodos estatísticos diferentes, através do teste de Kruskal-Wallis ou pela Correlação e Covariância. É pedido ainda, ao utilizador, um valor para o *threshold*.

- **Correlação e Covariância:** analisa a correlação e a covariância para cada par de features de modo a remover algumas variáveis que possam ser redundantes.
- **Kruskal Wallis:** compara duas features assumindo que existe independência entre as mesmas. Boas features devem apresentar valores altos para  $H$ , ou seja, rejeitam a hipótese nula o que irá fazer com que os padrões sejam da mesma distribuição.

$$H(F_i) = \frac{12}{n(n+1)} \sum_{i=1}^K n_i R_i^2 - 3(n+1)$$

## Feature Reduction

Se a feature selection selecionava as features mais importantes e que não fossem redundantes, na feature reduction o que irá ser feito será a transformação dos dados obtidos num menor número.

A redução das features poderá ser feita através de dois métodos diferentes, o PCA (Principal Component Analysis) e o LDA (Linear Discriminant Analysis).

- **PCA:** Método de análise estatística baseado na matriz de covariância. Tem o objectivo de encontrar as direções ortogonais onde os dados têm mais variância. Reduz a dimensionalidade pela projeção em direções com mais variância.

No caso do PCA, damos ainda a possibilidade de se seleccionar as componentes primárias mais relevantes que serão baseados ou no Scree Test ou no Kaiser Criteria, ambos com thresholds variáveis. O Kaiser Criteria com o threshold igual a 1 significa que apenas dimensões uma feature completa antes do PCA ser aplicado são relevantes para classificação. Por outro lado o Scree Test foca-se mais na representação cumulativa e, para um threshold igual a 0.9 significa que apenas dimensões que (somadas cumulativamente) perfazem 90% da soma de todos os valores próprios devem ser consideradas.

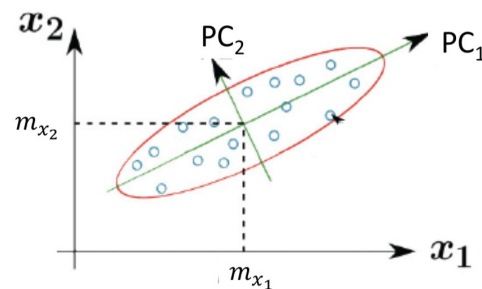


Fig 1. Principal Component Analysis

- **LDA:** generalização do método discriminante linear de Fisher. Projeta os dados em direções que maximizam a distância entre as médias das classes e que minimize a variância dos dados para cada classe.

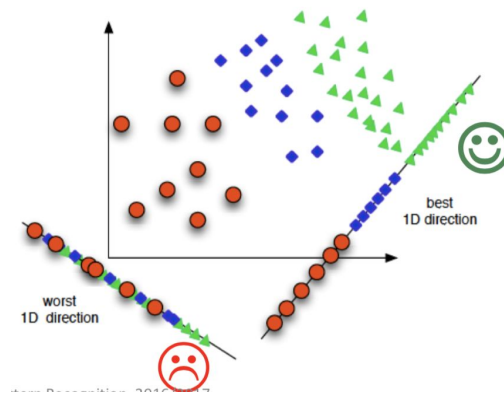


Fig 2. Linear Discriminant Analysis

## Classificadores

Depois de feito o anterior processamento dos dados foram implementados quatro classificadores diferentes, são eles o Euclidean Distance Classifier, o Mahalanobis Distance Classifier, Support Vector Machine (SVM) e ainda o K Nearest Neighbor.

- **Euclidean Distance Classifier:** a distância euclidiana é a distância entre dois pontos podendo ser provada pela aplicação repetido do teorema de Pitágoras. Este classificador é baseado no cálculo da distância euclidiana entre dois pontos. Basicamente define um centroid a partir dos dados de treino e calcula a distância (euclidiana) de cada amostra de teste para o centroid.
- **Mahalanobis Distance Classifier:** distância baseada nas correlações entre variáveis com as quais distintos padrões podem ser identificados e analisados. Este classificador é baseado no cálculo da distância de Mahalanobis entre um ponto P e uma distribuição D. Define um centroid a partir dos dados de treino e calcula a distância de Mahalanobis de cada amostra de teste para o centroid.

- **Support Vector Machine:** classificador que recebe como entrada, um conjunto de dados e que tenta prever para cada uma das entradas a classe de que faz parte (entre 2 classes possíveis). O SVM é um classificador linear binário, não probabilístico.

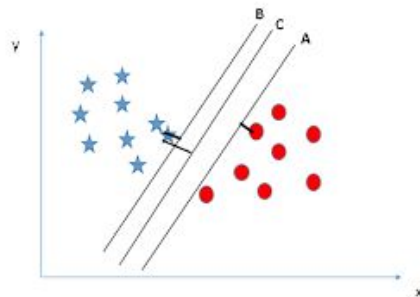


Fig 3. Support Vector Machine

- **K Nearest Neighbor:** algoritmo não-paramétrico que pode ser usado tanto para classificação como para regressão. A classe de um exemplo será determinada recorrendo à classe maioritária dos  $k$  vizinhos mais próximos.

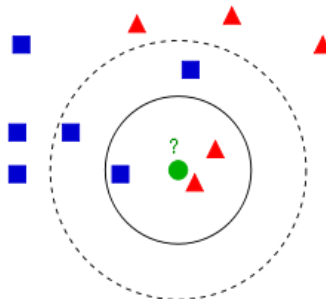


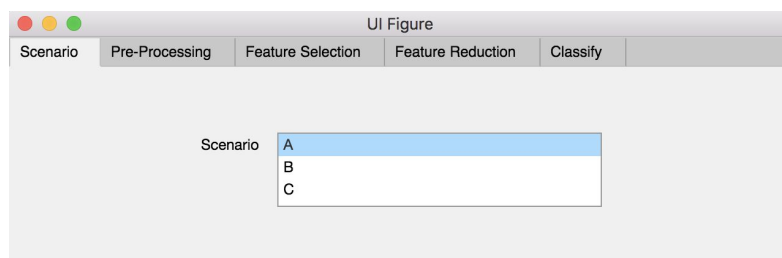
Fig 3. K Nearest Neighbor

- **Linear Discriminant:** classificador que assume que diferentes classes são geradas tendo em conta diferentes distribuições Gaussianas.
- **Decision tree:** método para classificação (ou regressão) não-paramétrico. O objetivo é criar um modelo que preveja o valor de uma variável que será o "target". Divide os dados, de acordo com algumas regras, para os valores observados e cria uma estrutura (árvore).

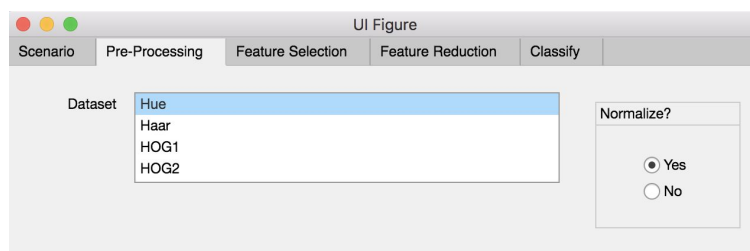
## Interface

Foi implementado uma GUI de modo a correr o projeto de uma forma mais fácil. As características da interface são:

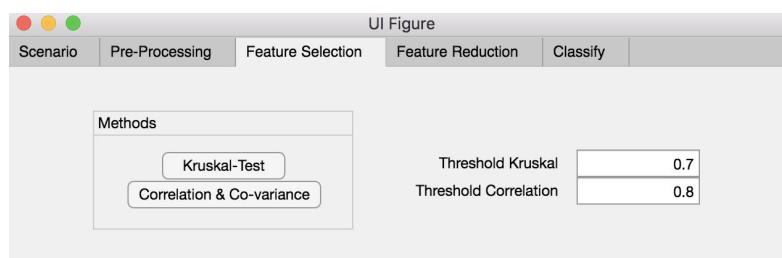
- Definir o cenário
- Carregar dataset pretendido
- Definir se se vai normalizar os dados ou não
- Escolher o método para a seleção das features assim como o valor do threshold
- Escolher o método pretendido para a redução das features
- Escolher o classificador



*Fig 4. Escolha do cenário pretendido*

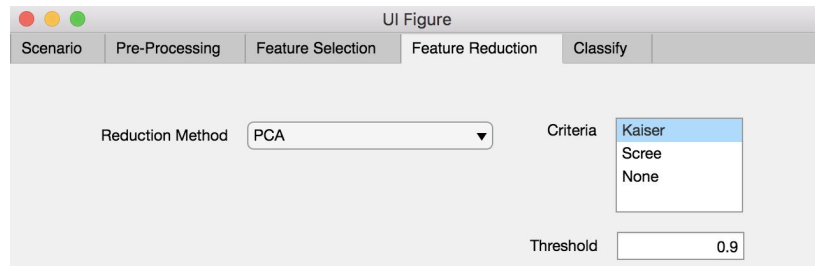


*Fig 5. Carregar o dataset pretendido*

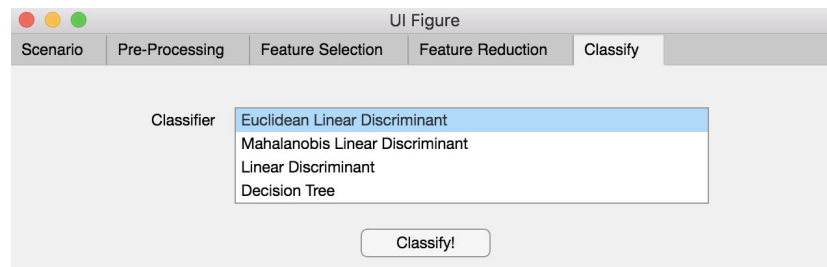


*Fig 6. Escolha do método para a seleção das features e definir o threshold*





*Fig 7. Escolha do método para a redução das features*



*Fig 8. Escolha do classificador*

## Resultados

Ao todo foram efectuados 78 testes. Todos esses testes podem ser visualizados no anexo A, no fim do documento. Ao longo dos testes fez-se variar os seguintes parâmetros:

- **Cenário** - A,B e C
- **Dataset** - No caso do cenário A, testaram-se os seguintes datasets, separadamente:
  - Hue
  - Hog1
  - Hog2
  - Hog3

No caso dos cenários B e C testaram-se todos esses datasets num só ficheiro, denominado 'huehogs'.

- **Normalização** - efetuando testes preliminares verificou-se que a normalização contribui para um aumento da performance em todos os casos. Posto isto, todos os testes foram feitos com normalização dos dados.
- **Método de seleção** - Sem método de seleção, usando o teste de Kruskal ou usando a Correlação e Covariância

- **Threshold para o teste de Kruskal e para a Correlação e Covariância** - foram usadas sempre os mesmos valores, a saber: 0.8 e 0.7 respetivamente.
- **Método de redução de features** - experimentamos correr o PCA, LDA e PCA & LDA.
- **Crítério para PCA** - usamos sempre o critério de Kaiser com um threshold de 0.9.
- **Classificador** - usamos o discriminante linear Euclidiano, o discriminante linear de Mahalanobis, o discriminante de Fisher, árvore de decisão, SVM e KNN.

Dada a dimensão do ficheiro que contém as features Haar, tivemos alguns problemas de memória e, consequentemente, não nos foi possível apurar a precisão dos classificadores usando essas features.

## **Correlação**

Abaixo seguem-se as imagens relativas aos resultados da seleção de features usando a correlação com um threshold de 0.7:

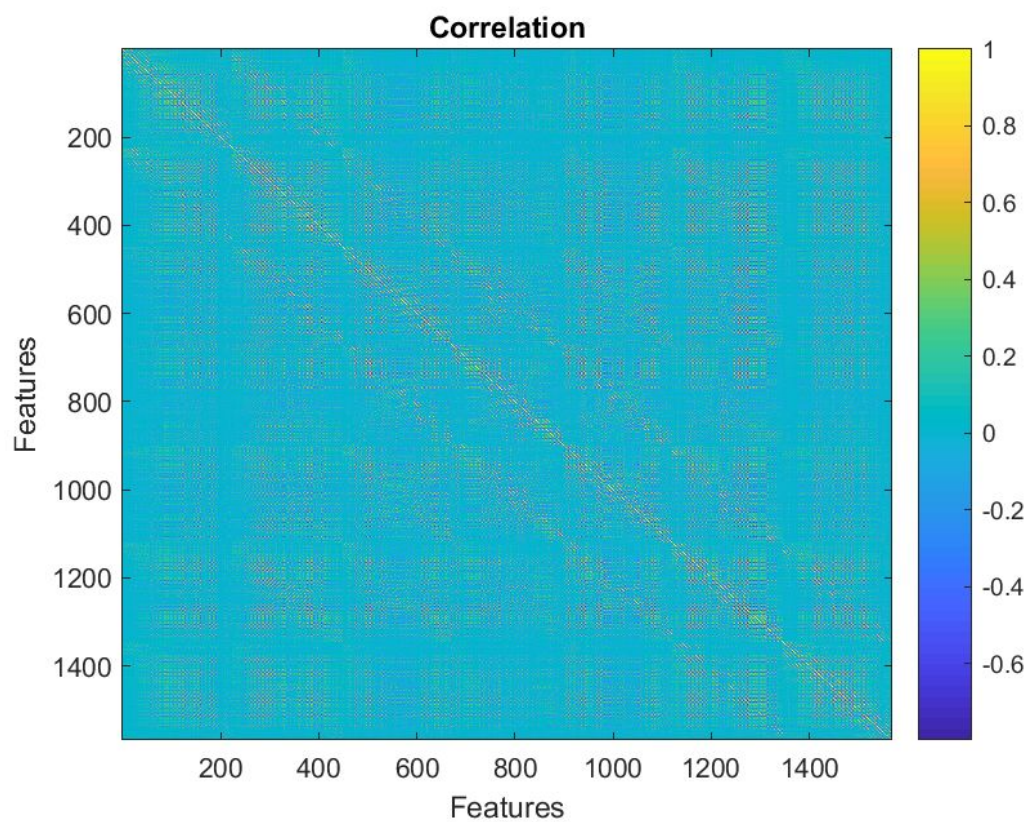


Fig 9. Correlação das features Hog1

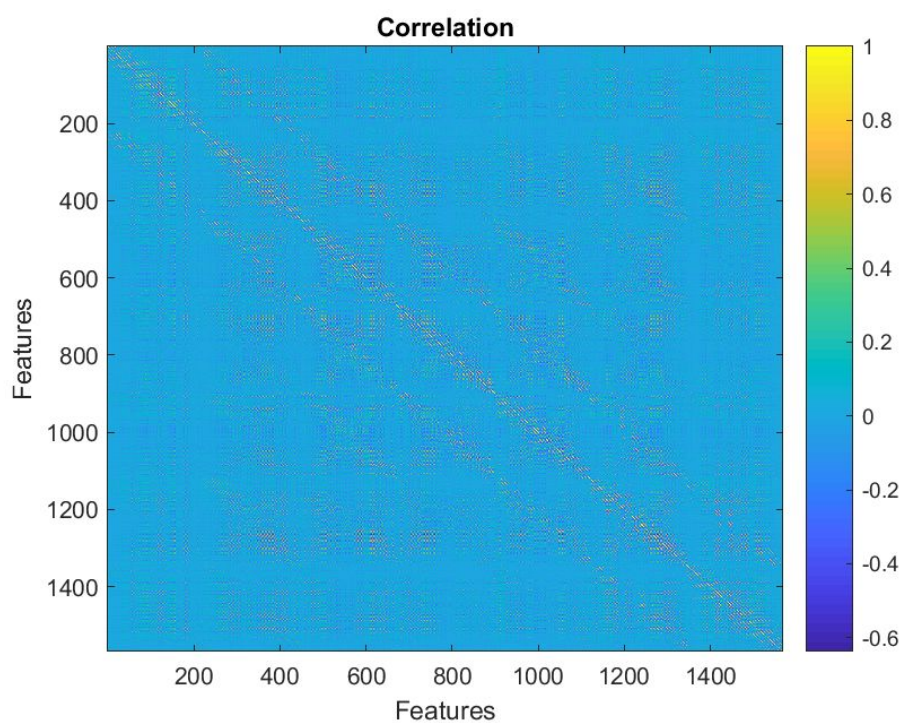


Fig 10. Correlação das features Hog2

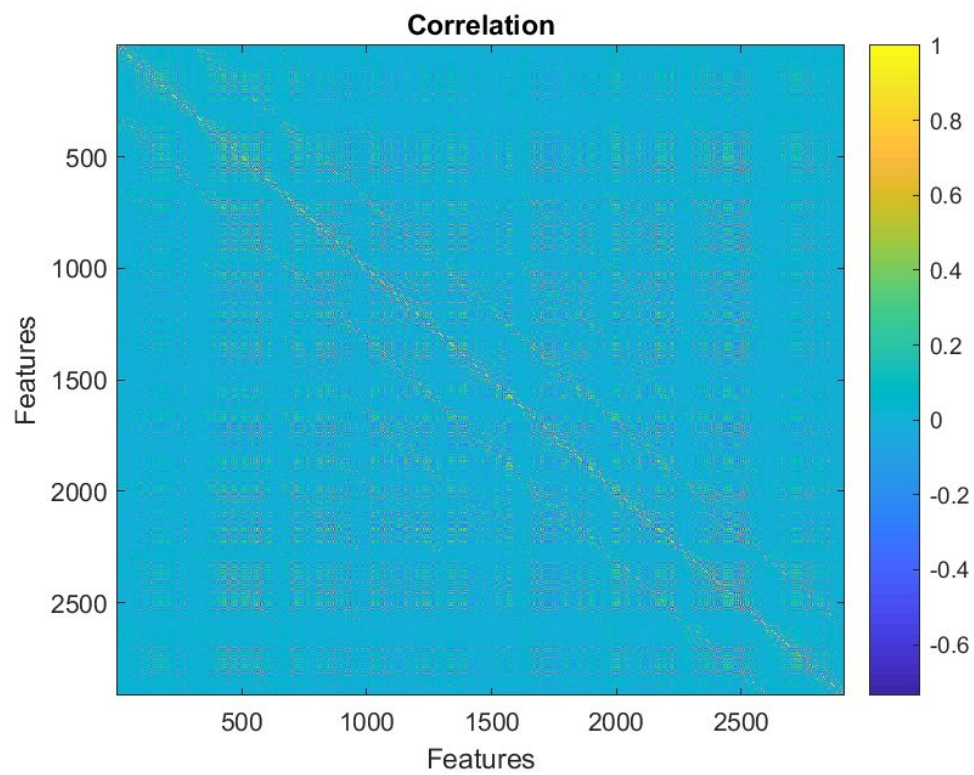


Fig 11. Correlação das features Hog3

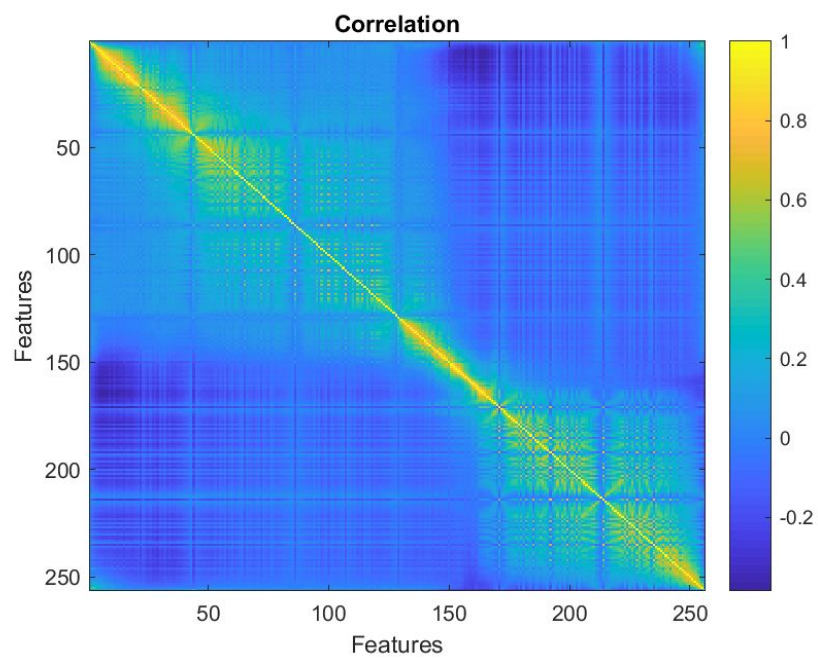


Fig 12. Correlação das features Hue



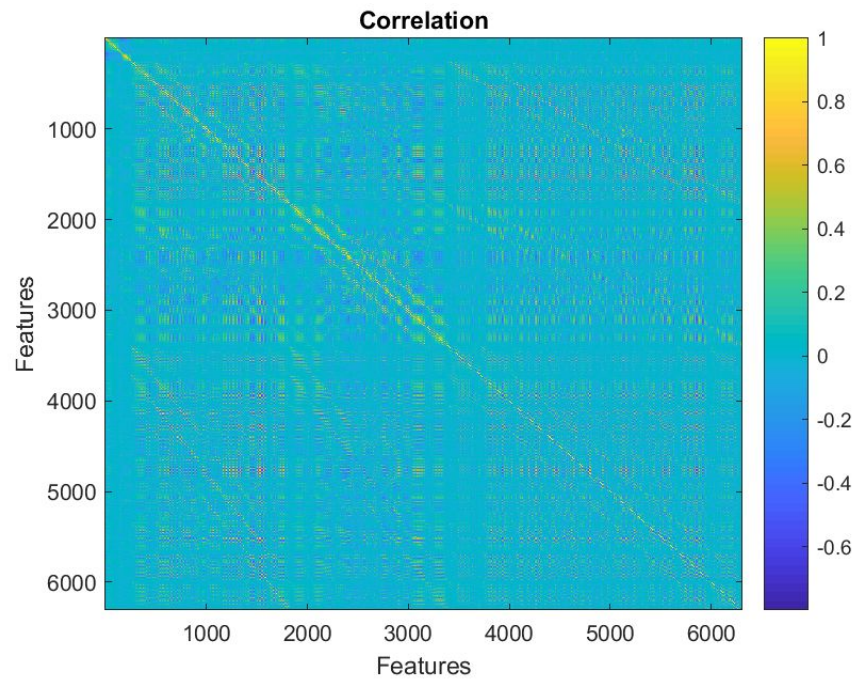


Fig 13. Correlação das features Huehogs

## Teste de Kruskal

Abaixo seguem-se as imagens relativas aos resultados da seleção de features usando o teste de Kruskal com um threshold de 0.8:

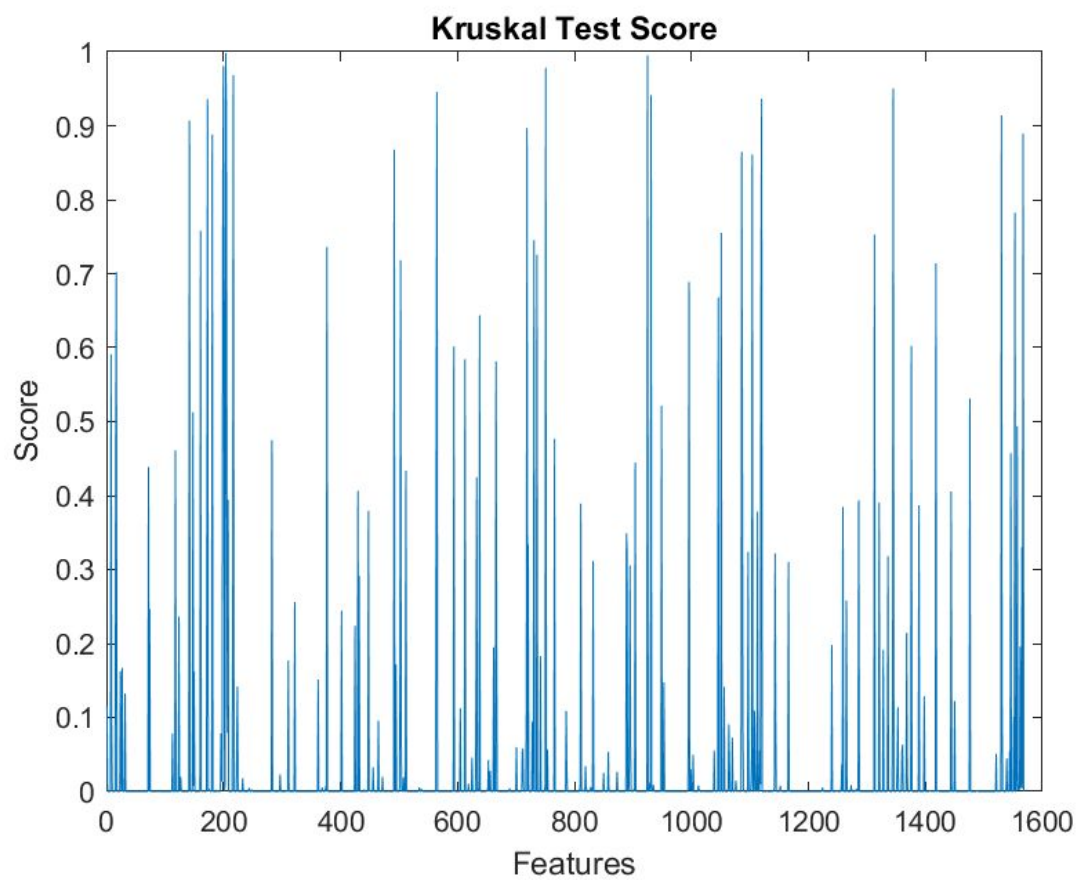


Fig 14. Teste de Kruskal das features hog1

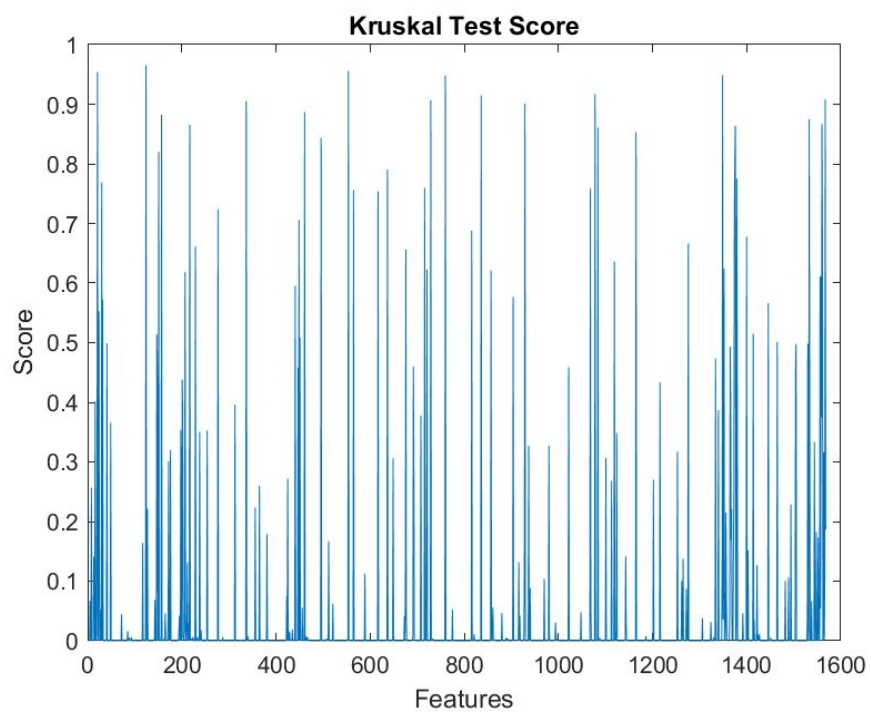


Fig 15. Teste de Kruskal das features hog2

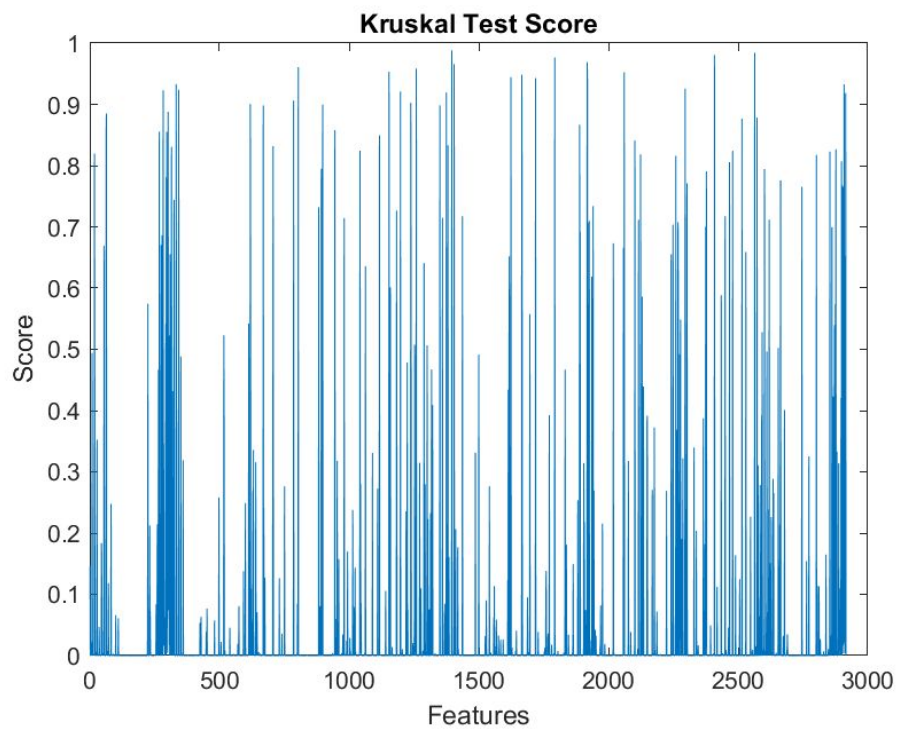


Fig 16. Teste de Kruskal das features hog3

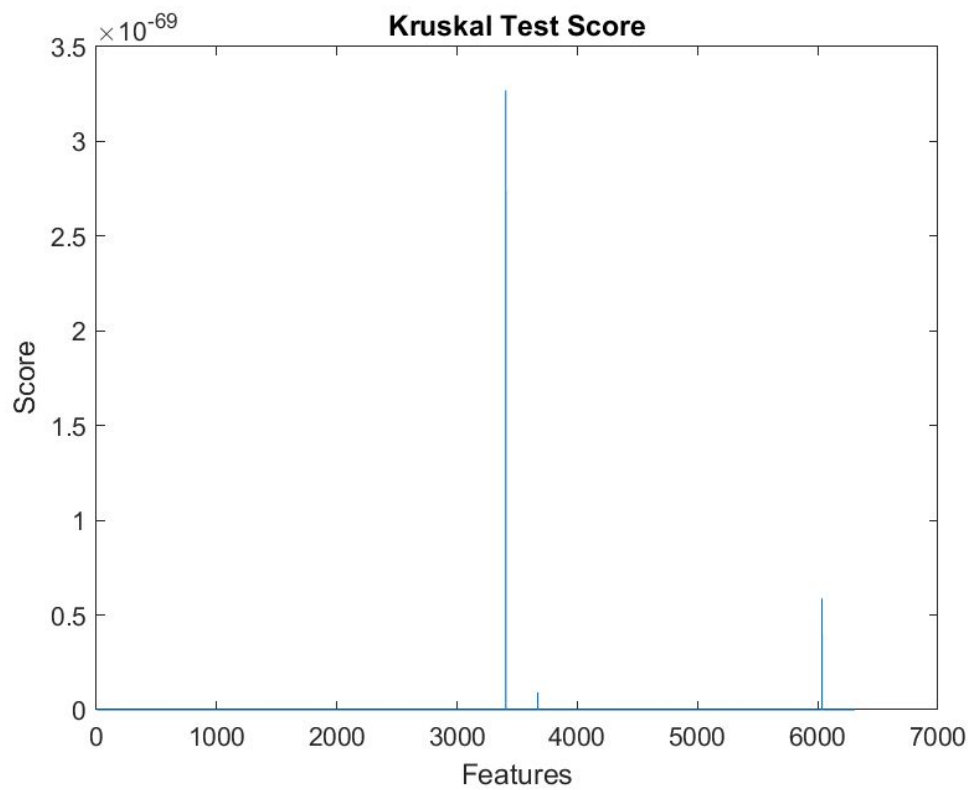


Fig 17. Teste de Kruskal das features Huehogs

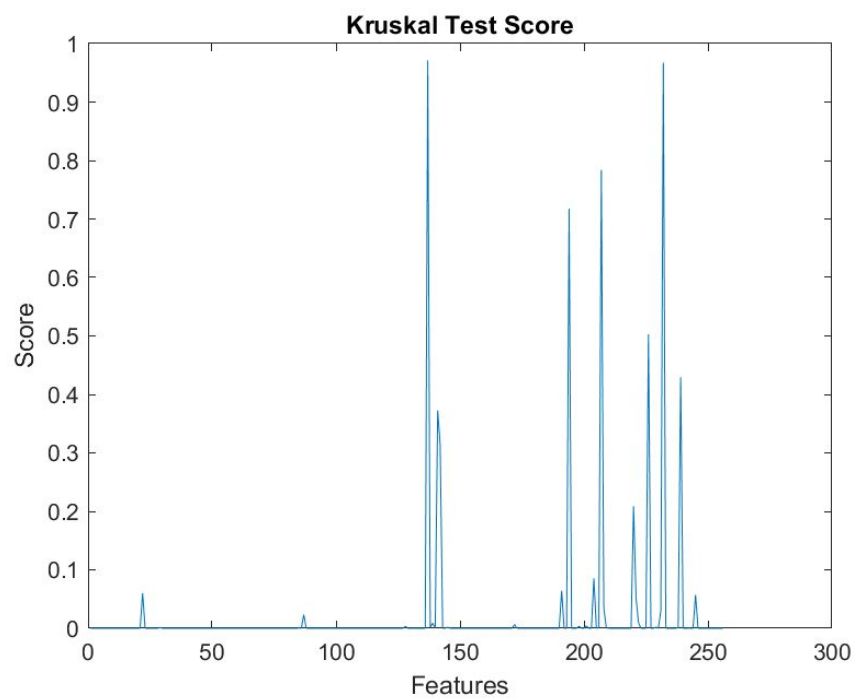


Fig 18. Teste de Kruskal das features Hue



## Curvas de ROC

Para o cenário A que envolve classificação binária foram visualizadas curvas de ROC. Lembramos que é uma representação gráfica que ilustra a performance de um classificador binário. Dado que foram efetuados muitos testes, torna-se inviável e improdutivo estar a mostrar todas as visualizações obtidas para todas as configurações utilizadas para o cenário A. Assim, vamos apenas mostrar um exemplo da curva ROC que se obtém ao correr o programa no cenário A.

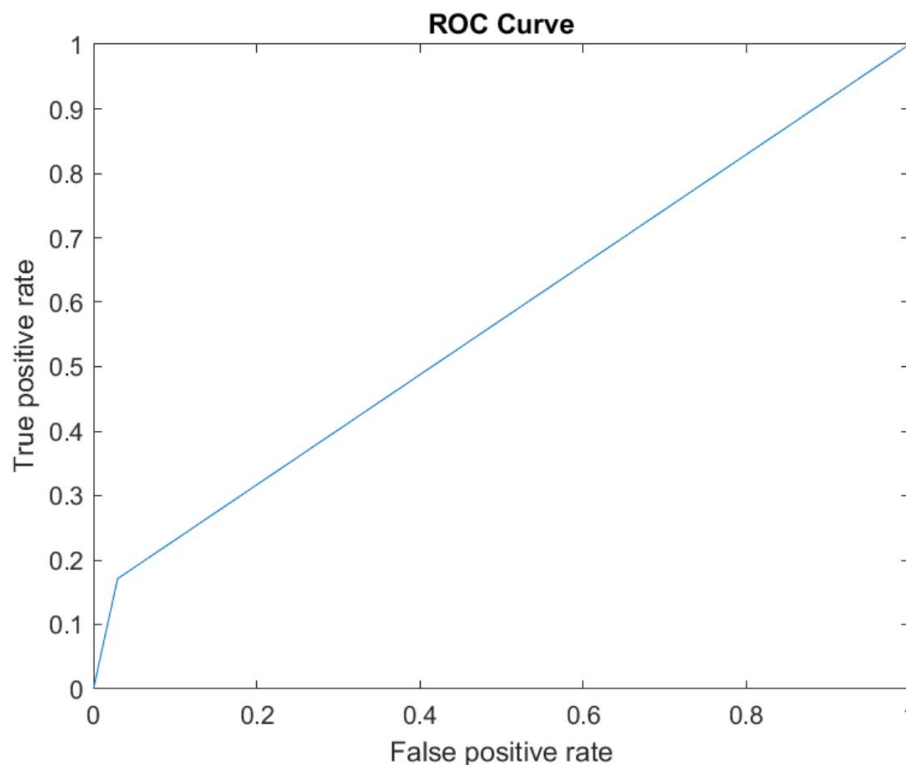


Fig 19. Curva de ROC relativo ao teste:  
cenárioA\_hue\_Yes\_empty\_0.7\_0.8\_PCA\_Euclidean Linear Discriminant\_Kaiser\_0.9

## Matriz Confusão

Para cada teste foi elaborada uma matriz de confusão para avaliação e melhor visualização da performance do classificador. Tal como as curvas de ROC, seria inviável estar a colocar todas as matrizes no relatório, pelo que apenas colocaremos dois exemplos relativo ao cenário B e ao cenário A, respetivamente.

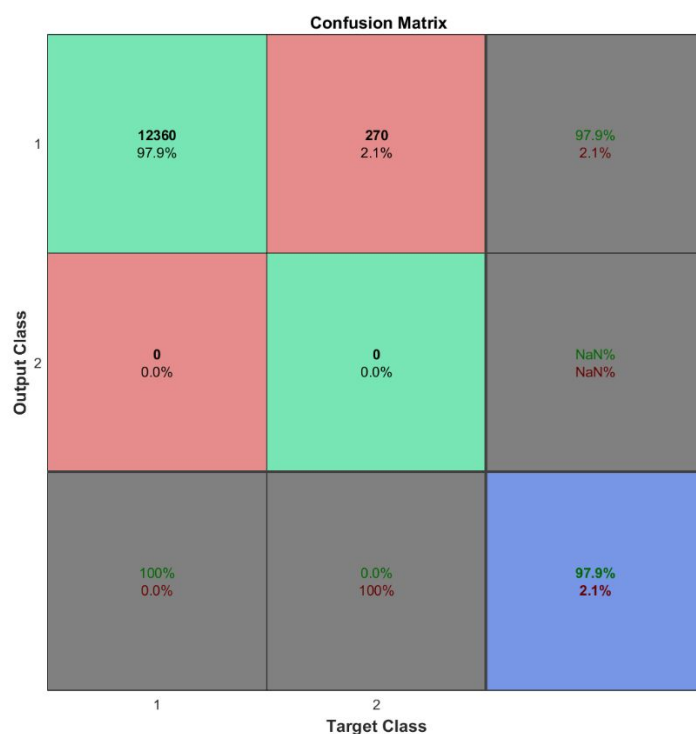


Fig 20. Matriz de Confusão relativa ao teste:  
cenárioA\_hog3\_Yes\_Kruskal-Test\_0.8\_0.7\_LDA\_Euclidean Linear Discriminant\_Kaiser\_0.9

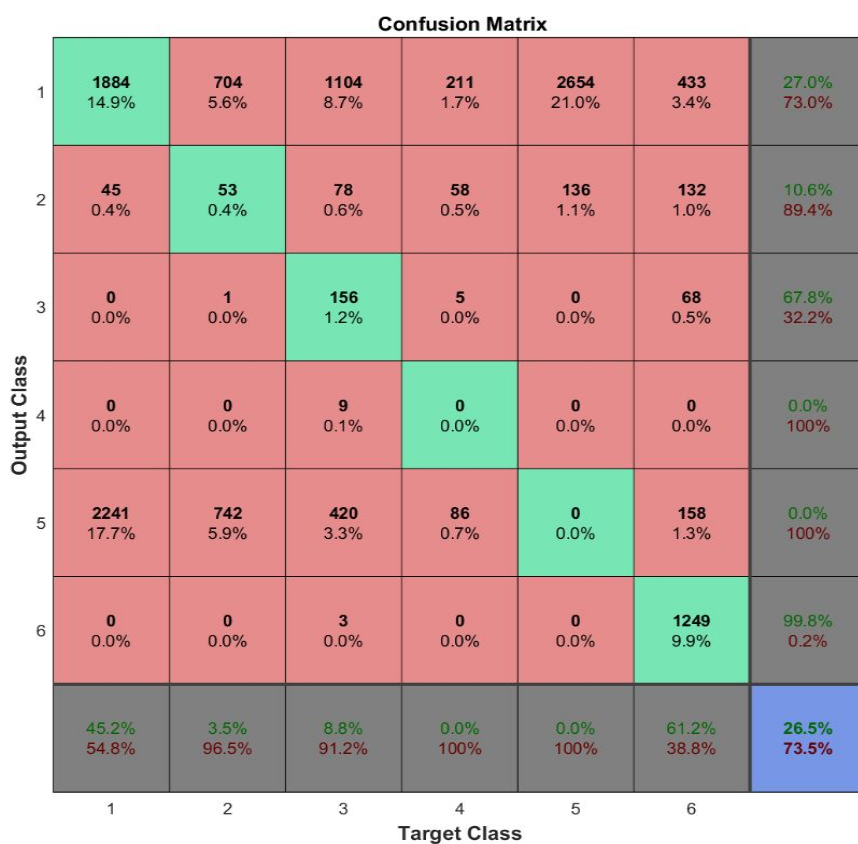


Fig 21. cenárioB\_huehogs\_Yes\_Correlation &  
Co-variance\_0.8\_0.7\_PCA\_KNN\_Kaiser\_0.9\_confusionMatrix

## Estatísticas de Performance

Para cada teste que se faça um ficheiro de texto com o nome específico da configuração é criado dentro do diretório `/results/testing/stats`. Eis um exemplo de um ficheiro desse tipo:

```
PRECISÃO: 0.9749  
ACCURACY: 0.9541  
RECALL: 0.9781  
PREVALENCE: 0.9755  
SPECIFICITY: 0.0000  
SENSITIVITY: 0.9781  
PREVALENCE: 0.9755  
F MEASURE: 0.9765
```

Fig 22. Ficheiro de output com estatísticas de performance relativo ao teste:  
A\_hog1\_Yes\_Correlation & Co-variance\_0.8\_0.7\_PCA\_Euclidean Linear  
Discriminant\_Kaiser\_0.9

**NOTA IMPORTANTE:** O projeto faz-se acompanhar de uma pasta *results* onde jazem todos os gráficos relativos a todos os testes criados diretamente pelo programa durante a execução. São guardados gráficos relativos à correlação, covariância, teste de kruskal, curvas de ROC, matrizes de confusão e estatísticas de cada teste (precisão, exatidão, sensibilidade, etc).

## Conclusões e Observações Finais

- A seleção e redução de features é importante pois permitem-nos eliminar features que não são relevantes e, por isso, não só aumentam a complexidade do classificador como diminuem a performance do classificador.
- Tendo feito uns testes preliminares concluímos que a normalização contribui para uma boa performance do classificador e, por isso, todos os testes registados foram feitos com os dados normalizados.
- Expectavelmente, para o cenário A, as features que permitem uma melhor classificação são os HOGS's. Dizemos expectavelmente dado que estamos a falar de uma classificação de um sinal (o sinal STOP) facilmente identificável pela sua forma (octógono) e não tanto pela sua cor ou brilho.
- No cenário A, o HOG3 foi mesmo aquele tipo de features que nos permitiu tirar melhores resultados.

- Para o cenário B e C usamos as features HOG'S e HUE porque nos pareceram bastante relevantes para classificar sinais de diferentes cores e formas.
- Não notámos que o uso de diferentes métodos de seleção de features ( teste de kruskal ou correlação e covariância) tenha tido grande impacto na performance do classificador.
- O PCA permite claramente obter melhores resultados para o cenário A e B, independentemente dos classificadores em causa. Já para o cenário C obtivemos melhores resultados usando o LDA.
- Para o cenário A, árvore de decisão ou SVM revelaram ser as melhores escolhas. De facto, o teste que nos permitiu obter a melhor performance foi: (features=Hog3, Normalização=Sim; seleção de features = correlação; redução de features = PCA; classificador = árvore de decisão; critério de PCA = Kaiser)
- Para o cenário B, árvore de decisão ou o discriminante linear revelaram ser as melhores escolhas. De facto, o teste que nos permitiu obter a melhor performance foi: (features=Huehogs, Normalização=Sim; seleção de features = teste de kruskal; redução de features = PCA; classificador = discriminante linear; critério de PCA = Kaiser)
- Para o cenário C, árvore de decisão ou SVM revelaram ser as melhores escolhas. De facto, o teste que nos permitiu obter a melhor performance foi: (features=Hog3, Normalização=Sim; seleção de features = correlação; redução de features = LDA; classificador = discriminante linear euclidiano; critério de PCA = Kaiser)
- O teste que nos permitiu obter a pior performance para o cenário A foi: (features=Hog3, Normalização=Sim; seleção de features = teste de Kruskal; redução de features = PCA; classificador = discriminante linear euclideano; critério de PCA = Kruskal)
- O teste que nos permitiu obter a pior performance para o cenário B foi: (features=huehogs, Normalização=Sim; seleção de features = correlação; redução de features = PCA; classificador = KNN; critério de PCA = Kaiser)
- O teste que nos permitiu obter a pior performance para o cenário C foi: (features=huehogs, Normalização=Sim; seleção de features = teste de Kruskal; redução de features = PCA&LDA; classificador = discriminante linear; critério de PCA = Scree)

## Futuros Desenvolvidimentos

Seria interessante numa futura oportunidade de melhoramento do projeto poder, por exemplo, subdividir em diferentes percentagens a percentagem de testing, a percentagem de training e a de cross-validation. Seria igualmente oportuno implementar um sistema de votação entre vários classificadores e usar a decisão maioritária para determinar o output esperado.

## Bibliografia

- J. Stallkamp, M. Schlipsing, J. Salmen and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, 2011, pp. 1453-1460.
- César Teixeira & Bernardete Ribeiro, Pattern Recognition Handouts, DEI-FCTUC, University of Coimbra, 2018
- J.P. Marques de Sá, —Pattern Recognition: Concepts, Methods and Applications || , 2001, XIX, 318 p., 197 illus., Springer-Verlag (2001)

## Anexo A

1	cenário	dataset	normaliza	features	kruskal th	correlatio	feature re	classifier	pca criter	criteria th	precision	accuracy	prevalenc	specificit	sensitivit	F Measure
2	A	hue	Yes	empty	0.7	0.8	PCA	Euclidean	Kaiser	0.9	0.8289	0.8118	0.8319	0.0038	0.9751	0.8961
3	A	hog2	Yes	Correlatic	0.8	0.7	PCA	Euclidean	Kaiser	0.9	0.9567	0.9363	0.9576	0.0000	0.9777	0.9671
4	A	hog2	Yes	Kruskal-T	0.8	0.7	PCA	Euclidean	Kaiser	0.9	0.9567	0.9363	0.9576	0.0000	0.9777	0.9671
5	A	hog1	Yes	Correlatic	0.8	0.7	PCA	Euclidean	Kaiser	0.9	0.9749	0.9541	0.9755	0.0000	0.9781	0.9765
6	A	hog1	Yes	Kruskal-T	0.8	0.7	PCA	Euclidean	Kaiser	0.9	0.9749	0.9541	0.9755	0.0000	0.9781	0.9765
7	A	hog2	Yes	Correlatic	0.8	0.7	PCA	Decision	Kaiser	0.9	0.9828	0.9624	0.9827	0.0320	0.9788	0.9808
8	A	hog2	Yes	Kruskal-T	0.8	0.7	PCA	Decision	Kaiser	0.9	0.9828	0.9624	0.9827	0.0320	0.9788	0.9808
9	A	hog2	Yes	Correlatic	0.8	0.7	PCA	SVM	Kaiser	0.9	0.9919	0.9707	0.9921	0.0000	0.9785	0.9851
10	A	hog2	Yes	Kruskal-T	0.8	0.7	PCA	SVM	Kaiser	0.9	0.9919	0.9707	0.9921	0.0000	0.9785	0.9851
11	A	hog1	Yes	Correlatic	0.8	0.7	PCA	Decision	Kaiser	0.9	0.9919	0.9708	0.9920	0.0099	0.9785	0.9852
12	A	hog1	Yes	Kruskal-T	0.8	0.7	PCA	Decision	Kaiser	0.9	0.9919	0.9708	0.9920	0.0099	0.9785	0.9852
13	A	hog1	Yes	Correlatic	0.8	0.7	PCA	SVM	Kaiser	0.9	0.9947	0.9735	0.9949	0.0000	0.9785	0.9866
14	A	hog1	Yes	Kruskal-T	0.8	0.7	PCA	SVM	Kaiser	0.9	0.9947	0.9735	0.9949	0.0000	0.9785	0.9866
15	A	hog2	Yes	Correlatic	0.8	0.7	PCA	Mahalan	Kaiser	0.9	0.9986	0.9773	0.9987	0.0000	0.9786	0.9885
16	A	hog2	Yes	Kruskal-T	0.8	0.7	PCA	Mahalan	Kaiser	0.9	0.9986	0.9773	0.9987	0.0000	0.9786	0.9885
17	A	hog1	Yes	Correlatic	0.8	0.7	PCA	Mahalan	Kaiser	0.9	0.9988	0.9774	0.9988	0.0000	0.9786	0.9886
18	A	hog1	Yes	Kruskal-T	0.8	0.7	PCA	Mahalan	Kaiser	0.9	0.9988	0.9774	0.9988	0.0000	0.9786	0.9886
19	A	hog2	Yes	Correlatic	0.8	0.7	PCA	Linear Di	Kaiser	0.9	0.9993	0.9779	0.9993	0.0000	0.9786	0.9888
20	A	hog2	Yes	Kruskal-T	0.8	0.7	PCA	Linear Di	Kaiser	0.9	0.9993	0.9779	0.9993	0.0000	0.9786	0.9888
21	A	hog1	Yes	Correlatic	0.8	0.7	PCA	Linear Di	Kaiser	0.9	0.9994	0.9781	0.9994	0.0000	0.9786	0.9889
22	A	hog1	Yes	Kruskal-T	0.8	0.7	PCA	Linear Di	Kaiser	0.9	0.9994	0.9781	0.9994	0.0000	0.9786	0.9889
23	A	hog1	Yes	Correlatic	0.8	0.7	LDA	Decision	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
24	A	hog1	Yes	Correlatic	0.8	0.7	LDA	Euclidean	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
25	A	hog1	Yes	Correlatic	0.8	0.7	LDA	Linear Di	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
26	A	hog1	Yes	Correlatic	0.8	0.7	LDA	Mahalan	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
27	A	hog1	Yes	Correlatic	0.8	0.7	LDA	SVM	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
28	A	hog1	Yes	Kruskal-T	0.8	0.7	LDA	Decision	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
29	A	hog1	Yes	Kruskal-T	0.8	0.7	LDA	Euclidean	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
30	A	hog1	Yes	Kruskal-T	0.8	0.7	LDA	Linear Di	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
31	A	hog1	Yes	Kruskal-T	0.8	0.7	LDA	Mahalan	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
32	A	hog1	Yes	Kruskal-T	0.8	0.7	LDA	SVM	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
33	A	hog2	Yes	Correlatic	0.8	0.7	LDA	Decision	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
34	A	hog2	Yes	Correlatic	0.8	0.7	LDA	Euclidean	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
35	A	hog2	Yes	Correlatic	0.8	0.7	LDA	Linear Di	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
36	A	hog2	Yes	Correlatic	0.8	0.7	LDA	Mahalan	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
37	A	hog2	Yes	Correlatic	0.8	0.7	LDA	SVM	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
38	A	hog2	Yes	Kruskal-T	0.8	0.7	LDA	Decision	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
39	A	hog2	Yes	Kruskal-T	0.8	0.7	LDA	Euclidean	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
40	A	hog2	Yes	Kruskal-T	0.8	0.7	LDA	Linear Di	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892

A	hog2	Yes	Kruskal-T	0.8	0.7	LDA	Mahalan	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog2	Yes	Kruskal-T	0.8	0.7	LDA	SVM	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Correlatio	0.8	0.7	LDA	Decision	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Correlatio	0.8	0.7	LDA	Euclidean	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Correlatio	0.8	0.7	LDA	Linear Dis	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Correlatio	0.8	0.7	LDA	Mahalan	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Correlatio	0.8	0.7	LDA	SVM	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Kruskal-T	0.8	0.7	LDA	Decision	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Kruskal-T	0.8	0.7	LDA	Euclidean	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Kruskal-T	0.8	0.7	LDA	Linear Dis	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Kruskal-T	0.8	0.7	LDA	Mahalan	Kaiser	0.9	1.0000	0.9786	1.0000	NaN	0.9786	0.9892
A	hog3	Yes	Correlatio	0.8	0.7	PCA	Euclidean	Kaiser	0.9	0.9870	0.9791	0.9740	0.5091	0.9916	0.9893
A	hog3	Yes	Kruskal-T	0.8	0.7	PCA	Euclidean	Kaiser	0.9	0.9870	0.9791	0.9740	0.5091	0.9916	0.9893
A	hog3	Yes	Correlatio	0.8	0.7	PCA	Linear Dis	Kaiser	0.9	0.9999	0.9823	0.9962	0.9792	0.9823	0.9910
A	hog3	Yes	Kruskal-T	0.8	0.7	PCA	Linear Dis	Kaiser	0.9	0.9999	0.9823	0.9962	0.9792	0.9823	0.9910
A	hog3	Yes	Correlatio	0.8	0.7	PCA	Mahalan	Kaiser	0.9	0.9998	0.9825	0.9956	0.9455	0.9827	0.9911
A	hog3	Yes	Kruskal-T	0.8	0.7	PCA	Mahalan	Kaiser	0.9	0.9998	0.9825	0.9956	0.9455	0.9827	0.9911
A	hog3	Yes	Correlatio	0.8	0.7	PCA	SVM	Kaiser	0.9	0.9991	0.9843	0.9926	0.8830	0.9851	0.9920
A	hog3	Yes	Kruskal-T	0.8	0.7	PCA	SVM	Kaiser	0.9	0.9991	0.9843	0.9926	0.8830	0.9851	0.9920
A	hog3	Yes	Correlatio	0.8	0.7	PCA	Decision	Kaiser	0.9	0.9916	0.9850	0.9772	0.6389	0.9930	0.9923
A	hog3	Yes	Kruskal-T	0.8	0.7	PCA	Decision	Kaiser	0.9	0.9916	0.9850	0.9772	0.6389	0.9930	0.9923
B	huehogs	Yes	Correlatio	0.8	0.7	PCA	KNN	Kaiser	0.9	0.9767	0.7211	0.9635	0.5408	0.7280	0.8342
B	huehogs	Yes	Kruskal-T	0.8	0.7	PCA	KNN	Kaiser	0.9	0.9767	0.7211	0.9635	0.5408	0.7280	0.8342
B	huehogs	Yes	Kruskal-T	0.8	0.7	PCA	Euclidean	Kaiser	0.9	0.9870	0.9791	0.9455	0.9910	0.9786	1.0000
B	huehogs	Yes	Kruskal-T	0.8	0.7	LDA	Decision	Kaiser	0.9	0.9999	0.9823	0.9962	0.9792	0.9823	0.9910
B	huehogs	Yes	Kruskal-T	0.8	0.7	PCA	Mahalan	Kaiser	0.9	0.9999	0.9823	0.8830	0.9911	0.9786	1.0000
B	huehogs	Yes	Kruskal-T	0.8	0.7	PCA	Linear Dis	Kaiser	0.9	0.9998	0.9825	0.0038	0.9920	0.9786	1.0000
C	huehogs	Yes	Kruskal-T	0.7	0.8	PCA & LD	Linear Dis	Scree	0.9	NaN	0.0000	1.0000	NaN	0.0000	NaN
C	huehogs	Yes	Correlatio	0.8	0.7	PCA	Linear Dis	Kaiser	0.9	0.0000	0.8966	0.0000	0.8966	NaN	NaN
C	huehogs	Yes	Kruskal-T	0.8	0.7	PCA	Linear Dis	Kaiser	0.9	0.0000	0.8966	0.0000	0.8966	NaN	NaN
C	huehogs	Yes	Correlatio	0.8	0.7	LDA	Mahalan	Kaiser	0.9	0.9999	0.9823	0.9962	0.9792	0.9823	0.9910
C	huehogs	Yes	Correlatio	0.8	0.7	LDA	Euclidean	Kaiser	0.9	0.9998	0.9825	0.9956	0.9455	0.9827	0.9911
C	huehogs	Yes	Correlatio	0.8	0.7	LDA	Linear Dis	Kaiser	0.9	NaN	0.9966	0.0034	1.0000	0.0000	NaN
C	huehogs	Yes	Kruskal-T	0.8	0.7	LDA	Linear Dis	Kaiser	0.9	NaN	0.9966	0.0034	1.0000	0.0000	NaN
C	huehogs	Yes	Correlatio	0.8	0.7	PCA	KNN	Kaiser	0.9	NaN	1.0000	0.0000	1.0000	NaN	NaN
C	huehogs	Yes	Kruskal-T	0.8	0.7	PCA	KNN	Kaiser	0.9	NaN	1.0000	0.0000	1.0000	NaN	NaN
C	huehogs	Yes	Correlatio	0.8	0.7	LDA	KNN	Kaiser	0.9	NaN	NaN	NaN	NaN	NaN	NaN
C	huehogs	Yes	Kruskal-T	0.8	0.7	LDA	Decision	Kaiser	0.9	NaN	NaN	NaN	NaN	NaN	NaN
C	huehogs	Yes	Kruskal-T	0.8	0.7	LDA	KNN	Kaiser	0.9	NaN	NaN	NaN	NaN	NaN	NaN