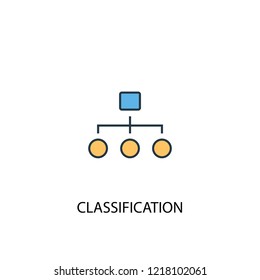
Practica 2.- Classificació



Grup: 304

Víctor Bosch Pueyo 1566583

Álvaro Caravaca Hernández 1566685

Juan Carlos Martínez Moreno 1566936

Índex

[Apartat B: Comparativa de models 3](#_Toc88231585)

[Introducció a la base de dades 3](#_Toc88231586)

[Comparació de models 3](#_Toc88231587)

[Apartat A: Classificació Numèrica 6](#_Toc88231588)

[Introducció a la base de dades 8](#_Toc88231589)

[HISTOGRAMES DELS ATRIBUTS 8](#_Toc88231590)

[GRÀFIQUES DE DISPERSIÓ 9](#_Toc88231591)

[Preprocessing 10](#_Toc88231592)

[Model Selection 10](#_Toc88231593)

[Crossvalidation 10](#_Toc88231594)

[Metric Analysis 10](#_Toc88231595)

[Hyperparameter Search 10](#_Toc88231596)

# Apartat B: Comparativa de models

//bla bla bla, explicació apartat b

//alvaro

## Introducció a la base de dades

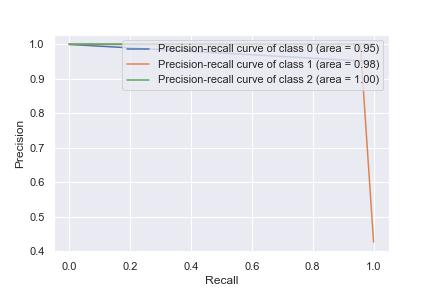
//alvaro

## Comparació de models

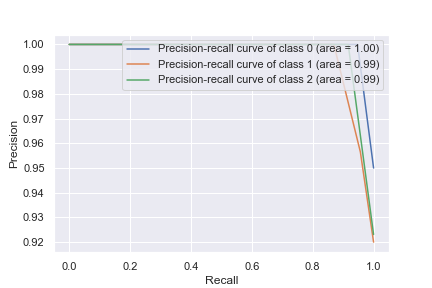
//alvaro i juankers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Logistic | SVM | KNN | Decision Tree |
| 50% train | 0.9775 | 0.9887 | 0.9775 | 0.8988 |
| 80% train | 0.9722 | 0.9722 | 0.9166 | 0.8888 |
| 70% train | 0.9814 | 0.9629 | 0.9814 | 0.9629 |

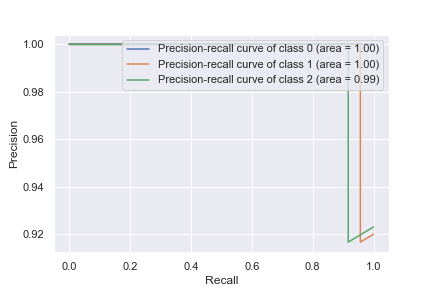
PR DECISION TREE



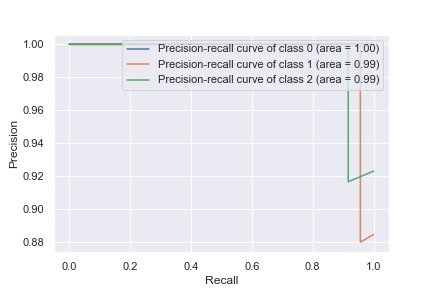
PR KNN



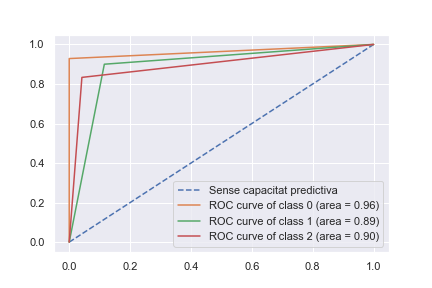
PR LOGISTIC REGRESSION



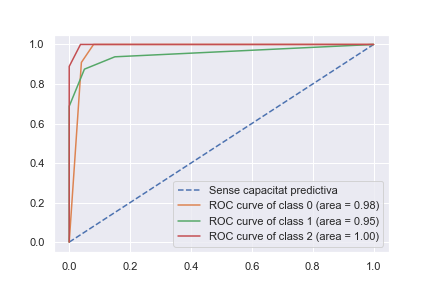
PR SVM



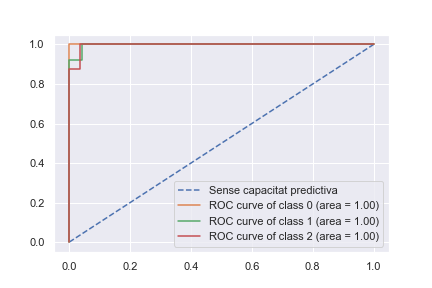
ROC DECISION TREE



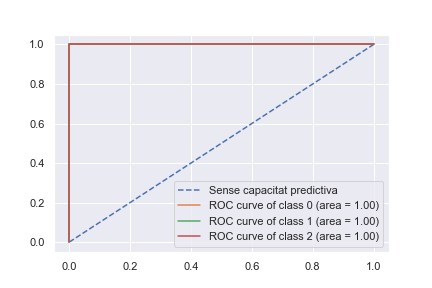
ROC KNN



ROC LOGISTIC REGRESSION



ROC SVM



Un dibujo de un perro

Descripción generada automáticamente con confianza baja

# Apartat A: Classificació Numèrica

En aquest apartat analitzarem una base de dades diferent a l’apartat A. Aquesta es una base de dades més rica i en la que analitzarem numèricament els resultats de diferents models.

Treballarem diferents aspectes de la classificació i tots giraran al voltant dels següents resultats obtinguts per cada model:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PERCEPTRÓ | Precisió | alpha | fit\_intercept | penalty | Tol |
| Bàsica |  |  |  |  |  |
| 50% train, 50% test | 0.706 | 0.0001 | True | None | 0.001 |
| 80% train, 20% test | 0.7175 | 0.0001 | True | None | 0.001 |
| 70% train, 30% test | 0.765 | 0.0001 | True | None | 0.001 |
|  |  |  |  |  |  |
| K-Fold |  |  |  |  |  |
| K = 2 | 0.713 | 0.0001 | True | None | 0.001 |
| K = 3 | 0.72101 | 0.0001 | True | None | 0.001 |
| K = 4 | 0.74 | 0.0001 | True | None | 0.001 |
| K = 5 | 0.702 | 0.0001 | True | None | 0.001 |
| K = 6 | 0.726 | 0.0001 | True | None | 0.001 |
|  |  |  |  |  |  |
| LOOCV | 0.727 | 0.0001 | True | None | 0.001 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| KNN | Precisió | leaf\_size | n\_neighbors | metric | weights |
| Bàsica |  |  |  |  |  |
| 50% train, 50% test | 0.506 | 30 | 5 | minkowski | uniform |
| 80% train, 20% test | 0.52 | 30 | 5 | minkowski | uniform |
| 70% train, 30% test | 0.51167 | 30 | 5 | minkowski | uniform |
|  |  |  |  |  |  |
| K-Fold |  |  |  |  |  |
| K = 2 | 0.4985 | 30 | 5 | minkowski | uniform |
| K = 3 | 0.5085 | 30 | 5 | minkowski | uniform |
| K = 4 | 0.5 | 30 | 5 | minkowski | uniform |
| K = 5 | 0.503 | 30 | 5 | minkowski | uniform |
| K = 6 | 0.5015 | 30 | 5 | minkowski | uniform |
|  |  |  |  |  |  |
| LOOCV | 0.5145 | 30 | 5 | minkowski | uniform |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Decision Tree | Precisió | criterion | splitter | max\_leaf\_nodes | max\_depth |
| Bàsica |  |  |  |  |  |
| 50% train, 50% test | 0.809 | gini | best | None | None |
| 80% train, 20% test | 0.81 | gini | best | None | None |
| 70% train, 30% test | 0.8 | gini | best | None | None |
|  |  |  |  |  |  |
| K-Fold |  |  |  |  |  |
| K = 2 | 0.8295 | gini | best | None | None |
| K = 3 | 0.8245 | gini | best | None | None |
| K = 4 | 0.8295 | gini | best | None | None |
| K = 5 | 0.821 | gini | best | None | None |
| K = 6 | 0.8265 | gini | best | None | None |
|  |  |  |  |  |  |
| LOOCV | 0.839 | gini | best | None | None |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Random Forest | Precisió | criterion | bootstrap | max\_leaf\_nodes | max\_depth |
| Bàsica |  |  |  |  |  |
| 50% train, 50% test | 0.864 | gini | True | None | None |
| 80% train, 20% test | 0.88 | gini | True | None | None |
| 70% train, 30% test | 0.88 | gini | True | None | None |
|  |  |  |  |  |  |
| K-Fold |  |  |  |  |  |
| K = 2 | 0.8675 | gini | True | None | None |
| K = 3 | 0.873 | gini | True | None | None |
| K = 4 | 0.879 | gini | True | None | None |
| K = 5 | 0.877 | gini | True | None | None |
| K = 6 | 0.8865 | gini | True | None | None |
|  |  |  |  |  |  |
| LOOCV | 0.881 | gini | True | None | None |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Regressió logística | Precisió | C | fit\_intercept | penalty | tol |
| Bàsica |  |  |  |  |  |
| 50% train, 50% test | 0.948 | 1.0 | True | None | 0.0001 |
| 80% train, 20% test | 0.965 | 1.0 | True | None | 0.0001 |
| 70% train, 30% test | 0.96 | 1.0 | True | None | 0.0001 |
|  |  |  |  |  |  |
| K-Fold |  |  |  |  |  |
| K = 2 | 0.954 | 1.0 | True | None | 0.0001 |
| K = 3 | 0.956 | 1.0 | True | None | 0.0001 |
| K = 4 | 0.9605 | 1.0 | True | None | 0.0001 |
| K = 5 | 0.9625 | 1.0 | True | None | 0.0001 |
| K = 6 | 0.9635 | 1.0 | True | None | 0.0001 |
|  |  |  |  |  |  |
| LOOCV | 0.966 | 1.0 | True | None | 0.0001 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SVM | Precisió | C | kernel | decision\_function\_shape | tol |
| Bàsica |  |  |  |  |  |
| 50% train, 50% test | 0.854 | 1.0 | rbf | ovr | 0.001 |
| 80% train, 20% test | 0.855 | 1.0 | rbf | ovr | 0.001 |
| 70% train, 30% test | 0.85167 | 1.0 | rbf | ovr | 0.001 |
|  |  |  |  |  |  |
| K-Fold |  |  |  |  |  |
| K = 2 | 0.8635 | 1.0 | rbf | ovr | 0.001 |
| K = 3 | 0.868 | 1.0 | rbf | ovr | 0.001 |
| K = 4 | 0.886 | 1.0 | rbf | ovr | 0.001 |
| K = 5 | 0.885 | 1.0 | rbf | ovr | 0.001 |
| K = 6 | 0.885 | 1.0 | rbf | ovr | 0.001 |
|  |  |  |  |  |  |
| LOOCV | 0.888 | 1.0 | rbf | ovr | 0.001 |

## Introducció a la base de dades

En base de dades d’aquest segon apartat treballem amb diferents característiques dels telèfons per acabar fent la predicció del seu preu de mercat. Els atributs amb els que treballarem son:

**Battery\_power**: bateria total que podem emmagatzemar d’un sol cop mesurat en mAh.

**Blue**: si te o no Bluetooth.

**Clock\_speed**: velocitat a la que el microprocessador executa instruccions.

**Dual\_sim**: si podem posar dues targes SIM o no.

**Fc**: mega píxels de la càmera frontal.

**Four\_g**: Si té o no 4G.

**Int\_memory**: Memòria interna en gigabytes.

**M\_deep**: profunditat del telèfon en cm.

**Mobile\_wt**: pes del telèfon.

**N\_cores**: Número de cores del processador.

**Pc**: Mega píxels de la càmera principal.

**Px\_height**: Resolució de l’altura.

**Px\_width**: Resolució de l’amplada.

**Ram**: RAM en mega bytes.

**Sc\_h**: Altura de la pantalla en cm.

**Sc\_w**: Amplada de la pantalla en cm.

**Talk\_time**: temps màxim durarà que una sola càrrega en trucada.

**Three\_g**: té 3G o no.

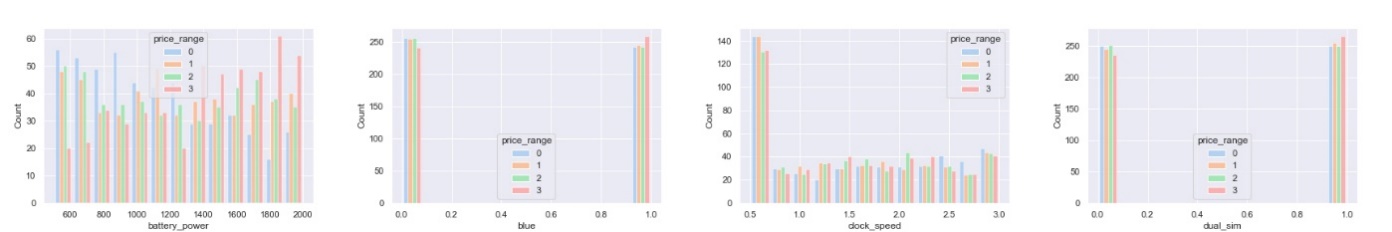
**Touch\_screen**: té pantalla tàctil o no.

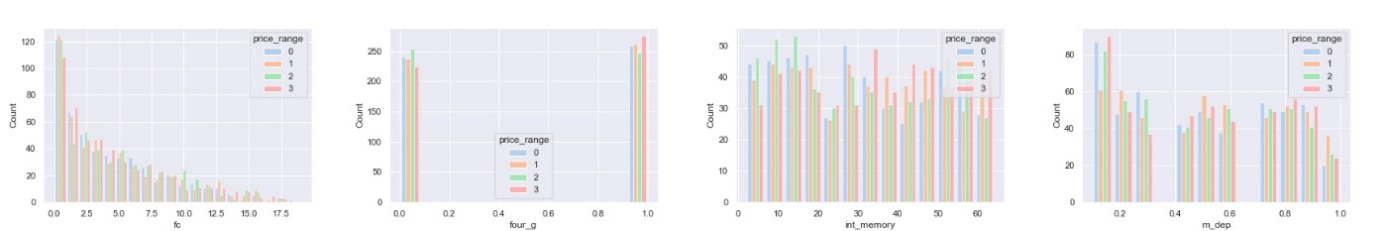
**Wifi**: té wifi o no.

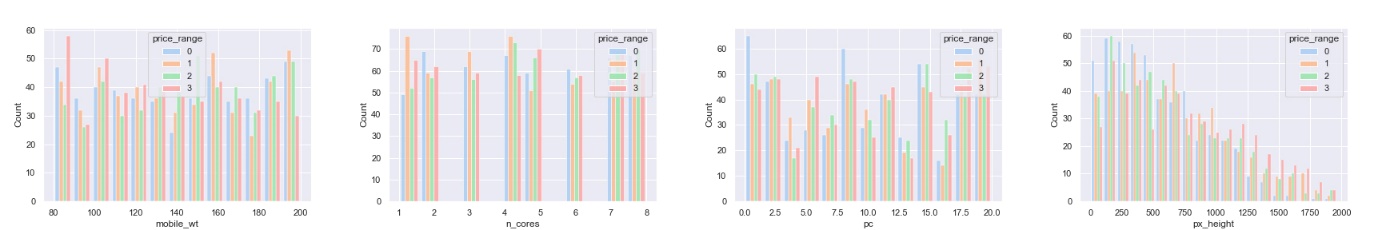
**Price\_range**: pot prendre 4 valors: 0 (low cost), 1 (medium cost), 2 (high cost) and 3 (very high cost).

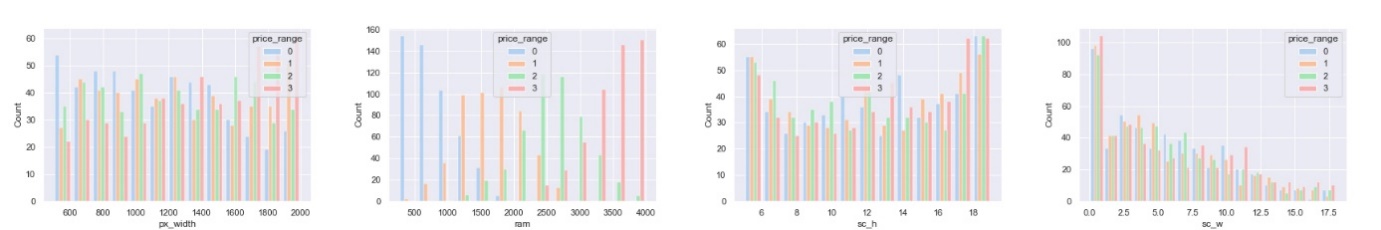
Per a poder fer un estudi adient de la nostra base de dades, fem una visualització dels histogrames i les gràfiques de dispersió de cada atribut. D’aquesta manera entendrem millor amb quines dades estem treballant. Les gràfiques són les següents:

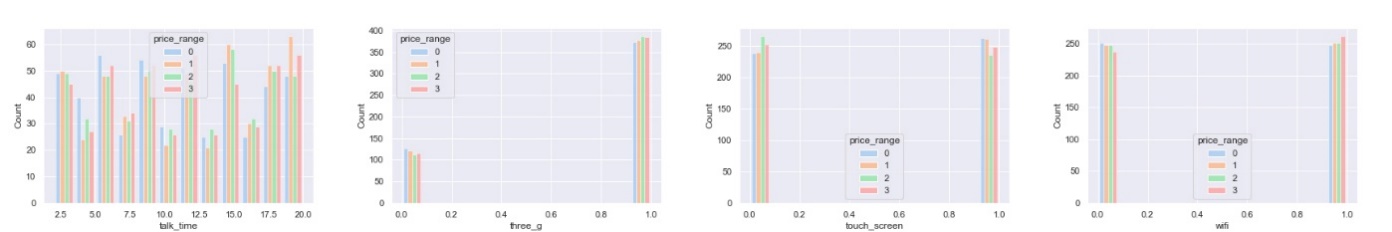
### HISTOGRAMES DELS ATRIBUTS

Histogrames dels atributs 1-4

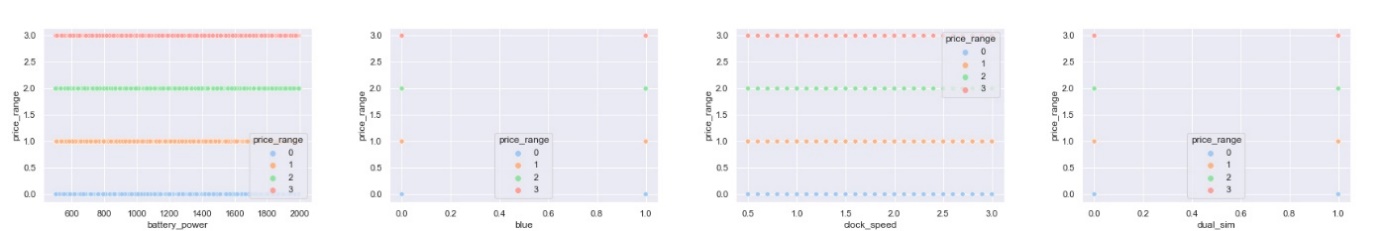
Histogrames dels atributs 5-8

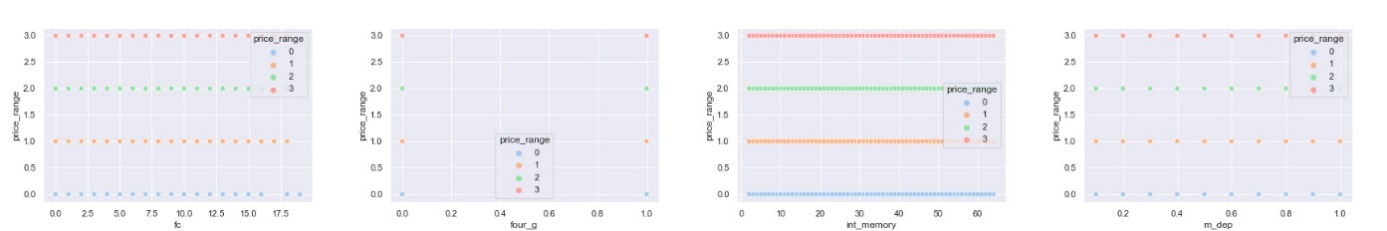
Histogrames dels atributs 9-12

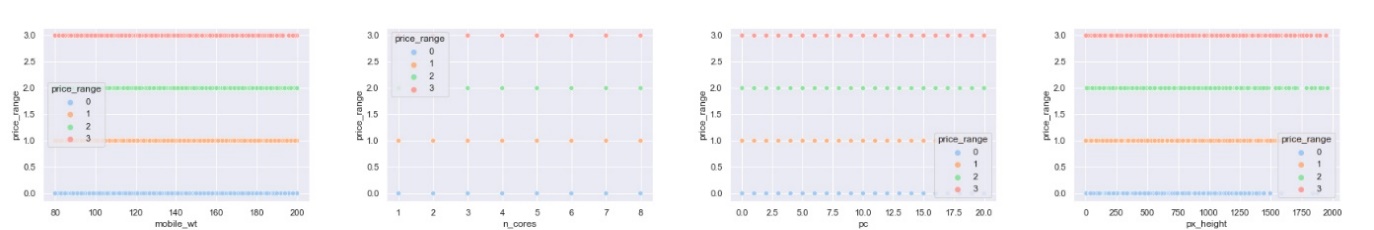
Histogrames dels atributs 13-16

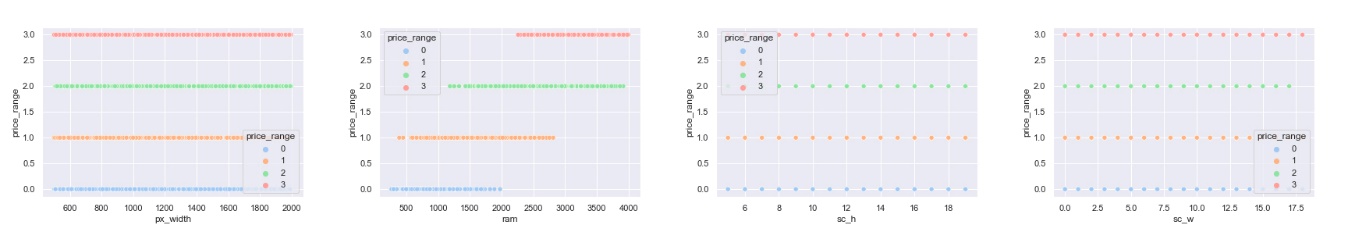
Histogrames dels atributs 17-20

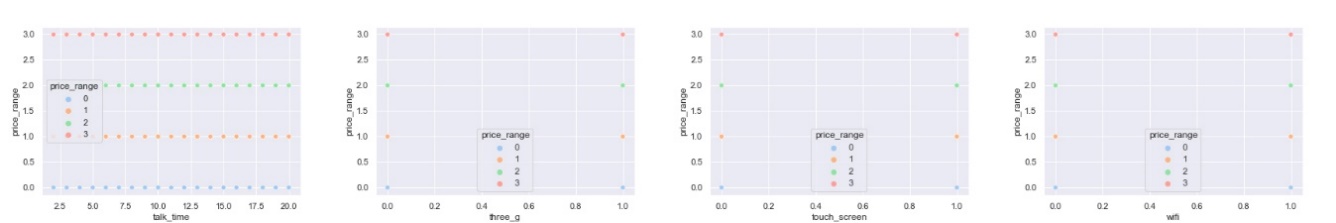
### GRÀFIQUES DE DISPERSIÓ

Dispersió dels atributs 1-4

Dispersió dels atributs 5-8

Dispersió dels atributs 9-12

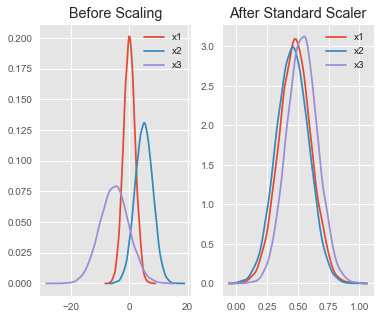
Dispersió dels atributs 13-16

Dispersió dels atributs 17-20

## Preprocessing

L’únic pre-processament que hem fet amb la nostra base de dades ha sigut escalar les dades. Ho hem fet ja que teníem dades numèriques amb rangs molt diferents i d’aquesta manera donarem la mateixa importància a tots els atributs.

Aquest canvi no faria falta per tots els models que hem implementat ja que, per exemple, els arbres de decisió no operen amb el valor exacte sinó que ramifica l’arbre per intervals de dades.



## Model Selection

En aquest subapartat analitzarem els resultats obtinguts en les diferents models que hem implementat. Aquests els trobem a les taules de l’inici de l’apartat A.

Com podem observar, els models que hem utilitzat són: Perceptró, k-nearest neighbors, desicion tree, Random Forest, Regressió logística i SVM. Apart d’això, per cada model, analitzem els seus resultats si dividim el data set amb diferents percentatges. Altres resultats obtinguts per a cada tipus de model que trobem a les taules els explicarem més endavant.

Executem tots els models amb els seus paràmetres per defecte.

Amb els resultats obtinguts podem afirmar que, per a la nostra base de dades el model que millors resultats ens proporciona es la regressió logística amb una precisió mitja de 95,77%. Tots els altres classificadors donen precisions al voltant de 82 – 83% menys el classificador k-Nearest neighbors, que ens dona uns resultats molt pobres (poc millors que tirar una moneda enlaire).

Havent fet la prova de dividir les dades en conjunts de mida diferent, observem que el que millors resultats ens proporciona es fer 80% de train i 20% de test. Es lògic aquest resultat ja que, com més dades donem per a entrenar el model, més podrem ajustar el nostre classificador. Igualment, no es recomanable utilitzar més del 80% de les dades totals per al conjunt de train perquè podem patir overfitting fàcilment.

Aplicar ensemble amb els mètodes utilitzat no tindria cap sentit ja que els ensemble methods s’utilitzen amb classificadors molt simples per a formar-ne un de complet. Els nostres classificadors ja són molt complexos i juntar-los seria molt costós.

## Crossvalidation

En comptes d’aplicar-ho només en certs models, hem aplicat crossvalidation a tots els implementats. Aquesta tècnica serveix per a analitzar el rendiment del model entrenat utilitzant un conjunt de validació. Aquest conjunt anirà variant en cada iteració ja que hem d’entrenar i validar el model per a cada subconjunt creat. Per exemple, si la nostra k es igual a 7, dividirem el model en 7 parts, 6 per a entrenar i una per a fer la validació. Aquest procés el farem 7 vegades fins que tots els subconjunts hagin estat conjunt de validació. Finalment retornem la mitjana del rendiment del model.

Igual que en els resultats anteriors, el que millor rendiment ens proporciona es la regressió logística amb un màxim de 96,35%. De la mateixa manera, tots els altres models han presentat resultats molt similars que en l’apartat anterior.

També, veiem que la score es millor com més dividim el conjunt. Té sentit ja que, com en el cas de dividir el data set en train i test, com més divisions en fem, més dades tindrem per a entrenar el model sobre el conjunt total. Si la k és igual a 2, estarem dividint en 50 – 50 en canvi, si es igual a 6, la nostra divisió serà de 83,33% d’entrenament contra 16,67%.

També hem aplicat la tècnica de leave one out però hem arribat a la conclusió que no és la més eficient en el nostre cas ja que obtenim un resultat similar a l’entrenament bàsic en tots els casos però tarda molt més en ser executat.

## Metric Analysis

//juankers

## Hyperparameter Search

//juankers

# Conclusions

//juankers o alvaro