Introducció

El present informe exposa els procediments i resultats d'execució de la **segona pràctica** del curs d'Intel·ligència Artificial Avançada, relativa a **l'Extracció i Selecció d'Atributs**.

El conjunt de dades que s'ha treballat fa referència a les vendes d'una empresa (*Wholesale Customers.csv*), i inclou dues etiquetes (tipus de client i zona de procedència) i 6 dimensions, que corresponen al volum de vendes en m.u. (milers d'unitats?) de diferents tipus d'articles. L'objectiu de la pràctica és reduir la dimensionalitat del conjunt de dades per tal d'extreure atributs i visualitzar les dades n-dimensionals.

El fitxer de dades originals serà comú per a tots els apartats de la pràctica, i té la següent estructura:

Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_paper	Delicatessen

Les etiquetes prenen els següents valors (discrets):

- Channel: 1: Hotel/Restaurant/Cafe (228 registres) – 2: Retail (142 registres)
- Region: 1: Lisboa (77 registres) – 2: Oporto (47 registres) – 3: Altres (316 registres)

La resta de valors pot prendre qualsevol valor (continu) dins els següents rangs:

- FRESH: (3, 112151)
- MILK: (55, 73498)
- GROCERY: (3, 92780)
- FROZEN: (25, 60869)
- DETERGENTS_PAPER: (3, 40827)
- DELICATESSEN: (3, 47943)

Activitat 1

Apliqueu una anàlisi PCA a les dades de la PAC i estudieu els valors propis i les variances resultants. Quantes components principals són necessàries per representar un 95% de la variança de les dades originals?

Decidiu si cal centrar o normalitzar les dades prèviament.

Com a pas previ a l'anàlisi de dades, realitzarem un tractament previ que asseguri la independència de la contribució de la variable respecte a la seva magnitud potencial. D'altra manera, categories de venda propenses a arribar a rangs alts de valors, com la venda de productes frescos, tindrien major pes en el càlcul de distàncies que d'altres com la venda de detergents. La normalització de les dades ens permetrà resoldre aquest problema igualant els rangs i/o la distribució de les dades.

A la primera pràctica vam decidir explorar diferents mètodes de normalització de les dades (Escalat 0-1 i Estandarització), atès que les diferents variables del problema tenien orígens de mesura diferents (transaccions, visites, etc.). En aquest cas, totes les variables representen

volums de vendes per un mateix canal i presenten distribucions de probabilitat similars com demostren, a mode d'exemple, les dues figures de la part inferior. Entenent, per tant, que no cal ajustar la distribució, decidim realitzar únicament un Escalat 0-1 de les dades per tal d'ajustar els rangs.

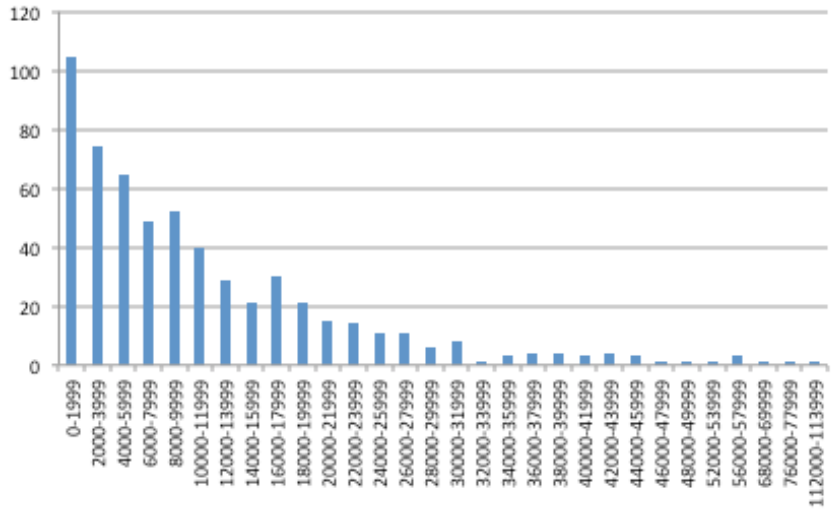


Figura 1: Distribució de la variable Fresh

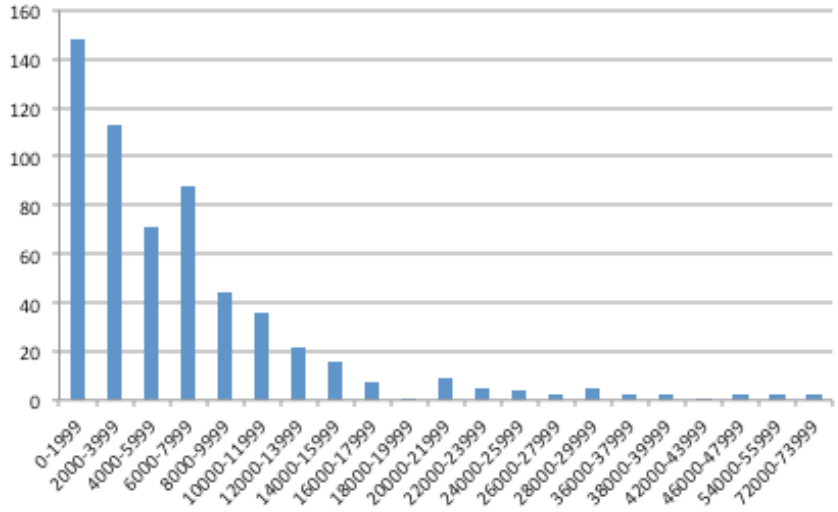


Figura 2: Distribució de la variable Milk

Recalcularem cada esdeveniment de cada variable del conjunt de dades mitjançant la següent conversió:

$$X' = \frac{X - 1}{\max(X) - 1}$$

el codi per la qual recuperarem de la primera pràctica implementat en la funció *scaleRatings*, recollida a l'arxiu *FunctionsForActivityOne.py* i que es presenta a continuació:

```
def scaleVals(vals):
    # Auxiliary function that receives a list of valuations as read
    # from the file and returns it scaled to 0..1
    return [(float(vals[i])-1)/(MAX_VALUATIONS[i]-1) for i in range(len(vals))]

def scaleRatings(array):
    newArray = []
    for l in array:
        tmp = []
        tmp.append(int(l[0])) # Append idWeb
        tmp.append(int(l[1])) # Append idUser
        values = scaleVals(l[2:])
        for x in values:
            tmp.append(x)
        newArray.append(tmp)
    return newArray
```

Per tal de simplificar el treball amb les dades tractades, s'ha definit una nova funció *writeStRatings* recollida a l'arxiu *FunctionsForActivityOne.py* que desa les dades corregides en un nou arxiu *newWSC.data*:

```
def writeStRatings(array, name="output.data"):
    # AWrite the data
    with open(name, "w") as fp:
        a = csv.writer(fp, delimiter="\t")
        a.writerows(array)
        msg = "Data succesfully written in file " + name
        return msg
```

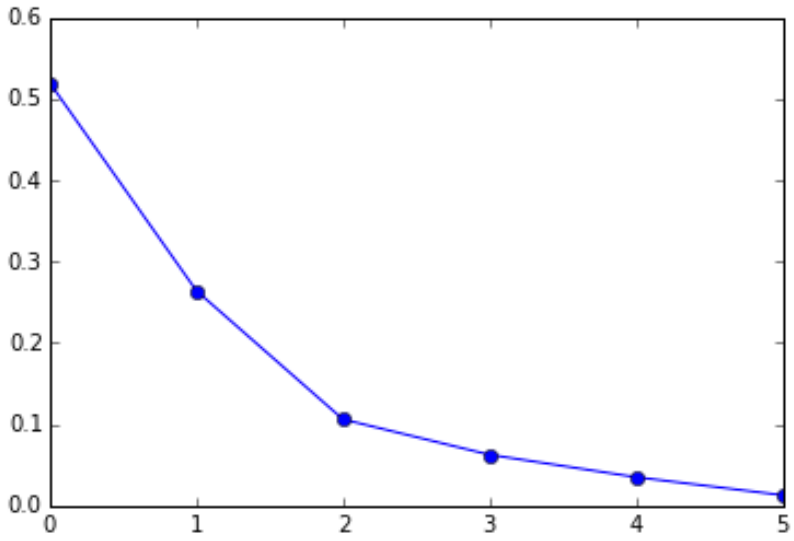
Tot el codi necessari per a la realització de la pràctica es troba a l'arxiu PAC2Code.py. El següent fragment recull la crida als mètodes esmentats. Tant els arxius de funcions com els nous arxius *newWSC.data* estan continguts a l'arxiu comprimit que acompanya aquest informe.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

"""
UOC Advanced Artificial Intelligence - 2014-2015 Q2
This code is a PCA Analysis based on the "Wholesale Customers" data
(Second deliverable of the course)
"""
# ------------------------------------------------------------------------------------------
# Activity One: Load files and scale data
# ------------------------------------------------------------------------------------------

import FunctionsForActivityOne as f  # Import all needed functions

from numpy import *

from sklearn.decomposition import PCA
# First, we read the data in Wholesale customers.csv
ratings = f.readRatings("Wholesale customers.csv")
# Now, we scale the distribution of every variable
scaledRatings = f.scaleRatings(ratings)
# The last step is to write the data
msg = f.writeStRatings(scaledRatings, "newWSC.data")
print(msg)


def readFile(filename="input.txt"):
    # Read the csv file, ignoring first row
    lines = list(map(lambda l: [float(x) for x in (
        l.strip()).split("\t")], (open(filename, 'r').readlines())))
    return lines

do = readFile("newWSC.data")
```

```
for x in do:
    data.append(x[2:])  # Data

X = numpy.array(data)

# Apply PCA requesting all components (no argument)
mypca = PCA()
mypca.fit(X)

# How many components are required to explain 95% of the variance
acumvar = [sum(mypca.explained_variance_ratio_[:i + 1])
           for i in range(len(mypca.explained_variance_ratio_))]
print(list(zip(range(len(acumvar)), acumvar)))

pylab.plot(mypca.explained_variance_ratio_, 'o-')
pylab.show()

# We will repeat the procedure with the non-scaled values

do2 = f.readRatings("Wholesale customers.csv")
# Apply PCA requesting all components (no argument)
for x in do2:
    data.append(x[2:])

X = numpy.array(data)
mypca = PCA()
mypca.fit(X)

# How many components are required to explain 95% of the variance
acumvar = [sum(mypca.explained_variance_ratio_[:i + 1])
           for i in range(len(mypca.explained_variance_ratio_))]
print(list(zip(range(len(acumvar)), acumvar)))

pylab.plot(mypca.explained_variance_ratio_, 'o-')
pylab.show()
```

A continuació es presenten els resultats de l'execució:

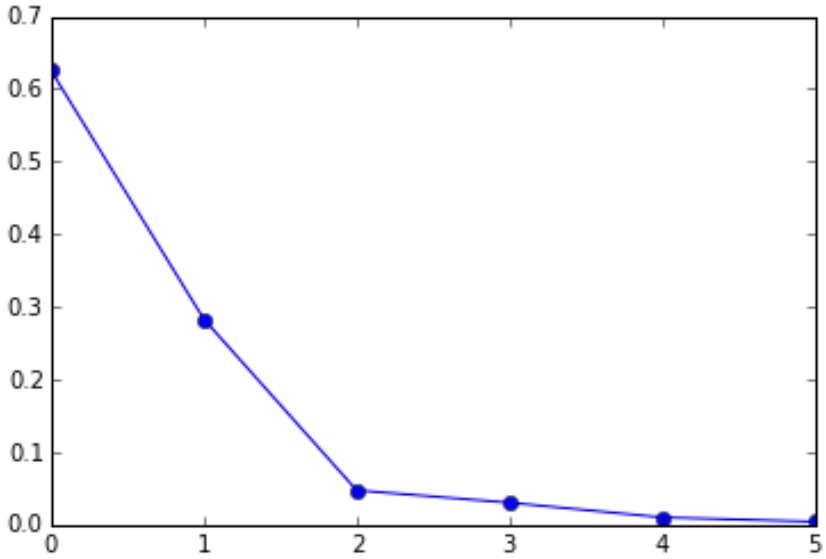
Cas estandarditzat:



Components	% Variació explicada
1	52%
2	78%
3	89%
4	95%
5	98%

6	100%

Cas No estandarditzat:



Components	% Variació explicada
1	62%
2	90%
3	95%
4	98%
5	99%
6	100%

És a dir, es pot explicar el 95% de la variança amb 4 components del PCA en el cas de les variables escalades i amb 3 components en el cas de l'espai de variables original.

Activitat 2

Amb les dues components principals obtingudes a l'exercici anterior, representeu gràficament les dades en dues dimensions (cada línia del fitxer haurà de ser un punt de la gràfica).

Haureu de dibuixar dues gràfiques. A la primera, el color dels punts serà el tipus de client (engròs/detall). A la segona, la zona de procedència (Lisboa/Oporto/altres).

Es poden distingir les classes?

El següent codi, també contingut en l'arxiu *PAC2Code.py*, executa el PCA, calcula les coordenades dels punts en el nou espai de dimensionalitat reduïda(*fit().transform()*) i gràfica els resultats distingint les classes.

```
# ------------------------------------------------------------------------------------------------
# Activity Two + Three: Load previous file and solve the PCA Analysis
# ------------------------------------------------------------------------------------------------

# Split thelist in Labels + Data
data = []
target = []
# Comment and uncomment the labels to get the charts
# target_names = ['Hotel/Restaurant/Cafe ', 'Retail']
target_names = ['Lisbon', 'Oporto', 'Others']
target_names = numpy.array(target_names)

for x in do:
    data.append(x[2:])
    # Switch the target names to show tendencies x[0]<-->x[1]
    target.append(x[1])

X = numpy.array(data)
y = numpy.array(target)

# mypca = PCA()
mypca = PCA(n_components=2)
X_r = mypca.fit(X).transform(X)

# Percentage of variance explained for each components
print('explained variance ratio (first two components): %s'
      % str(mypca.explained_variance_ratio_))

plt.figure()
# Switch the target [1,2]<-->x[1,2,3]
for c, i, target_name in zip("rg", [1, 2], target_names):
    plt.scatter(X_r[y == i, 0], X_r[y == i, 1], c=c, label=target_name)
plt.legend()
plt.title('PCA of WholeSale Customers')
# Additionally, we will get the covariance
cov = pca.get_covariance()
print(cov)
```

A continuació es recullen les gràfiques corresponents a les **2 components principals** del PCA per la versió de les **dades escalades**. En aquest nou espai queda explicada, com ja s'ha esmentat, el 78% de la variància.

En la primera versió de la gràfica, amb diferents colors en funció de la classe "channel", es llegeix un agrupament evident de les dades: els clients d'engròs presenten una dispersió important en l'àmbit de la component 2, mentre que la seva dispersió es petita al llarg de la component 1. Els clients de Retail, per la seva banda, presenten un comportament ben diferenciat: elevada dispersió en l'ambit de la component 1 i molta estabilitat en la component 2. Existeix una agrupació important de clients al voltant de l'origen de coordenades, i existeixen pocs *outliers* entre la mostra.

Figura 3: PCA - 2 components principals - Engrós/Detall

En la segona versió de la gràfica, amb diferents colors en funció de la classe "region", no podem llegir un agrupament de les dades en classes: sembla que els hàbits de compra presenten certa independència respecte a aquesta dimensió. Els comentaris sobre l'agrupació de les dades al voltant de (0,0) són els mateixos que els de la gràfica anterior, atès que només s'han modificat els colors de la gràfica. Com els individus amb procedència altres són majoritaris i agrupen regions molt diverses, hem provat a representar la mateixa gràfica únicament amb Lisboa i Oporto, intentant trobar aquí un agrupament per classes. El resultat es recull a la pàgina següent: tampoc s'observa un agrupament per classes.

Figura 4: PCA - 2 components principals - Lisboa/Oporto/Altres

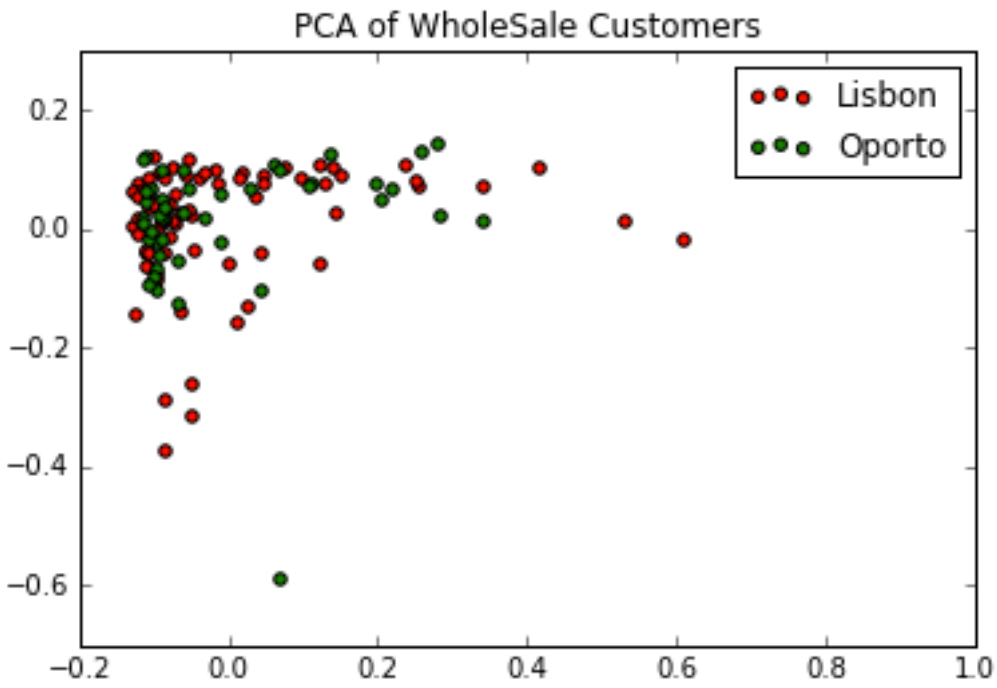


Figura 5: PCA - 2 components principals - Lisboa/Oporto

Cal notar que vam realitzar la implementació del PCA amb el paquet scikit-learn i amb l'algorisme recollit als apunts del curs (obtenció de la matriu de covariància, centrat i diagonalització de la mateixa, etc.). El resultat obtingut en ambdós cassos va ser equivalent a excepció del **signe de l'eix d'abscisses**. Per al cos de la pràctica vam decidir quedar-nos amb la versió de l'sklearn, per la seva simplicitat i la quantitat de mètodes disponibles.

Exercici 3

Repetiu l'exercici anterior però amb les components 2 i 3 del PCA, i després amb les 3 i 4. Analitzeu les diferències amb les gràfiques de l'exercici 2.

Per tal de representar les components 2-3 i 3-4 del PCA, hem renovat l'execució modificant el nombre de components requerits (*n_components = 4*). Els resultats es mostren a la pàgina següent. Com és d'esperar, a mesura que representem components que expliquen menys variància, els punts presenten una menor dispersió, fent-se cada cop més difícil la lectura de les classes en el cas de les gràfiques per tipus de client (engròs/detall). Així, a la Figura 6, que representa el PCA amb les components 2 i 3 i dona color a les classes per tipus de client, encara podem localitzar (tot i que amb una certa dificultat) 2 grups ben diferenciats. Aquests agrupaments es dilueixen completament en el cas de la representació de les components 3 i 4: els punts presenten menys dispersió i és impossible localitzar agrupaments distingits.

En el cas de les gràfiques per procedència del client, l'agrupament de les classes era evident ja en la representació de les 2 components principals, i s'agreuja encara més amb la representació de les components 2 i 3 i 3 i 4. Com en el cas anterior, a mesura que representen components que expliquen menys variància, els punts s'uneixen al voltant d'un punt (0,0).

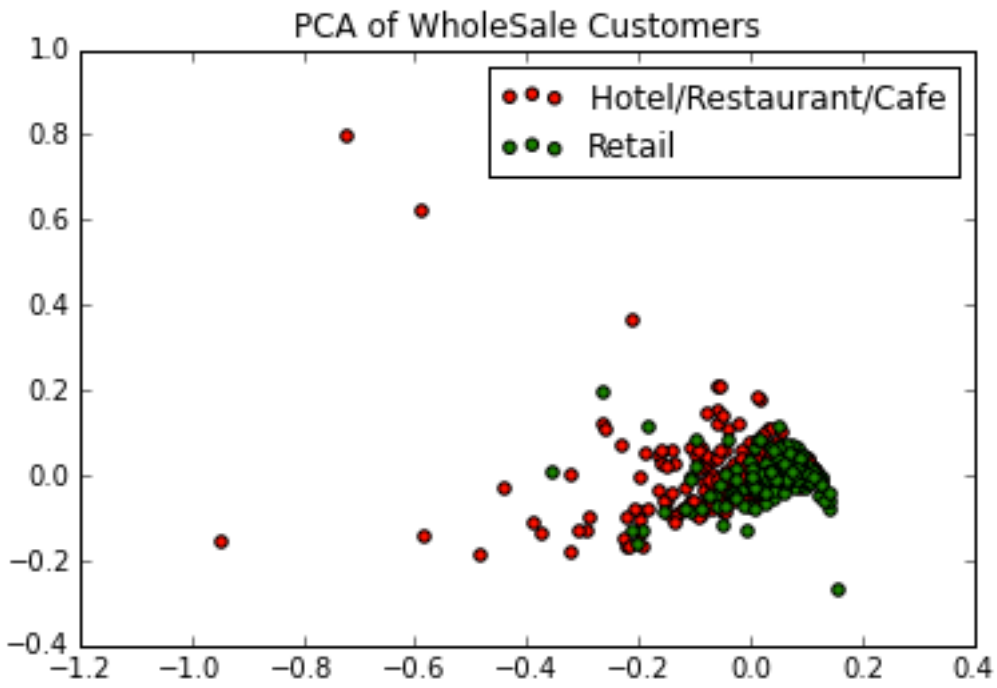


Figura 6: PCA – Components 2 i 3 - Engrós/Detall

Figura 7: PCA – Components 3 i 4 - Engrós/Detall

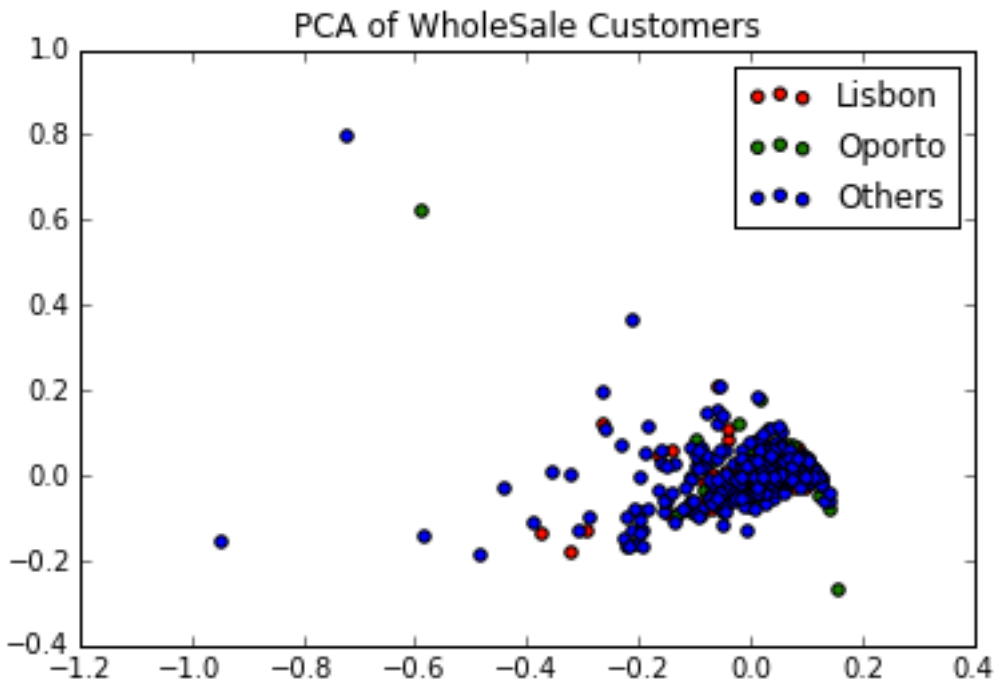


Figura 8: PCA – Components 2 i 3 – Lisboa/Oporto/Altres

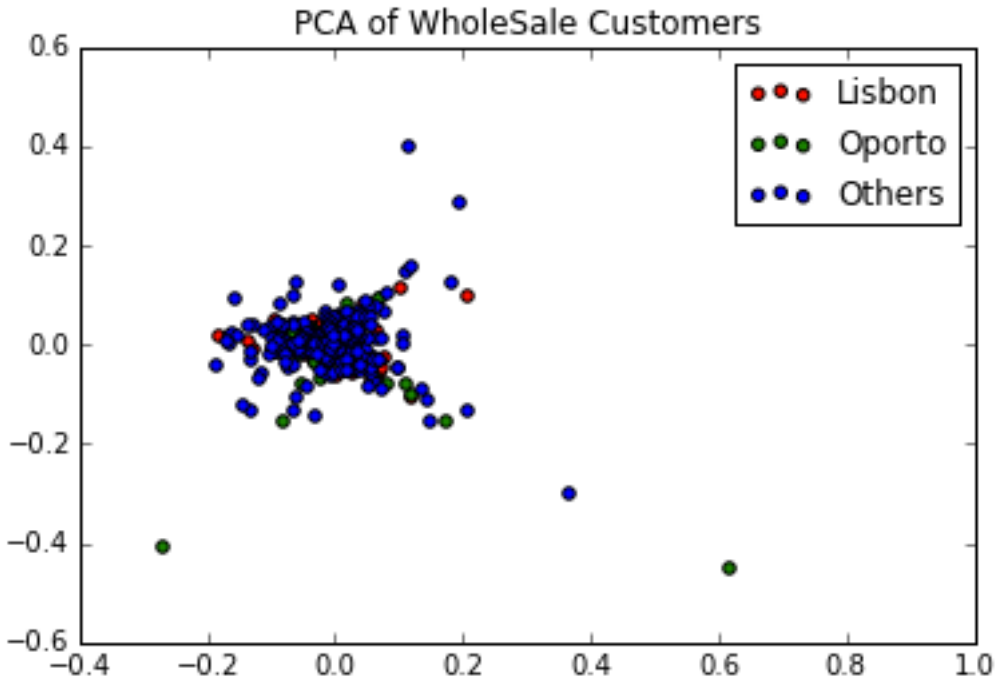


Figura 9: PCA – Components 3 i 4 – Lisboa/Oporto/Altres

Exercici 4

Apliqueu el mètode "multidimensional scaling" (MDS) a les dades de vendes. Dibuixeu una gràfica en dues dimensions colorejant amb el tipus de client i una altra amb la seva procedència. Compareu-les amb les gràfiques de l'exercici 2.

Podreu trobar la implementació i molta informació sobre com fer-la servir al web de scikit-learn.

En aquest apartat hem realitzat la implementació de l'algorisme MDS mitjançant el mètode MDS del paquet manifold de la llibreria sklearn.

```
# ------------------------------------------------------------------------------------------
# Activity Four: Solve MDS Analysis
# ------------------------------------------------------------------------------------------

from matplotlib import pyplot as plt

from sklearn import manifold

from sklearn.metrics import euclidean_distances

similarities = euclidean_distances(X)

mds = manifold.MDS(n_components=2, dissimilarity="precomputed")
results = mds.fit(similarities)
coords = results.embedding_
plt.figure()
for c, i, target_name in zip("rg", [1, 2, 3], target_names):
    plt.scatter(X[y == i, 0], X[y == i, 1], c=c, label=target_name)
    # plt.scatter(X_r[y == i, 1], X_r[y == i, 2], c=c, label=target_name)
plt.legend()
plt.title('MDS of WholeSale Customers')
```

Les gràfiques següents presenten els resultats de l'execució amb *n_components = 2*. Com era d'esperar, els resultats son similars als obtinguts amb el PCA (ambdues tècniques aproximen la forma de la distribució de les dades originals en un nou espai 2-d), i de nou es fàcilment distingible la classe "tipus de client" i molt menys la classe "procedència". Atenent als resultats, ambdues tècniques donen resposta en un mode similar al requeriment, tot i que l'MDS és molt més lent que el PCA.

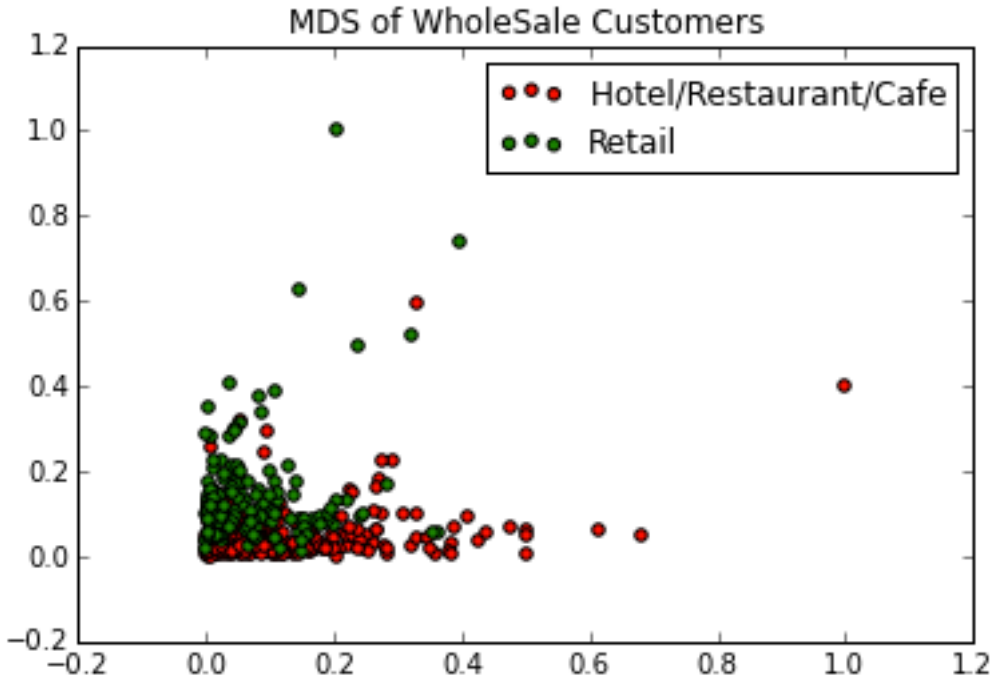


Figura 10: MDS - Engrós/Detall

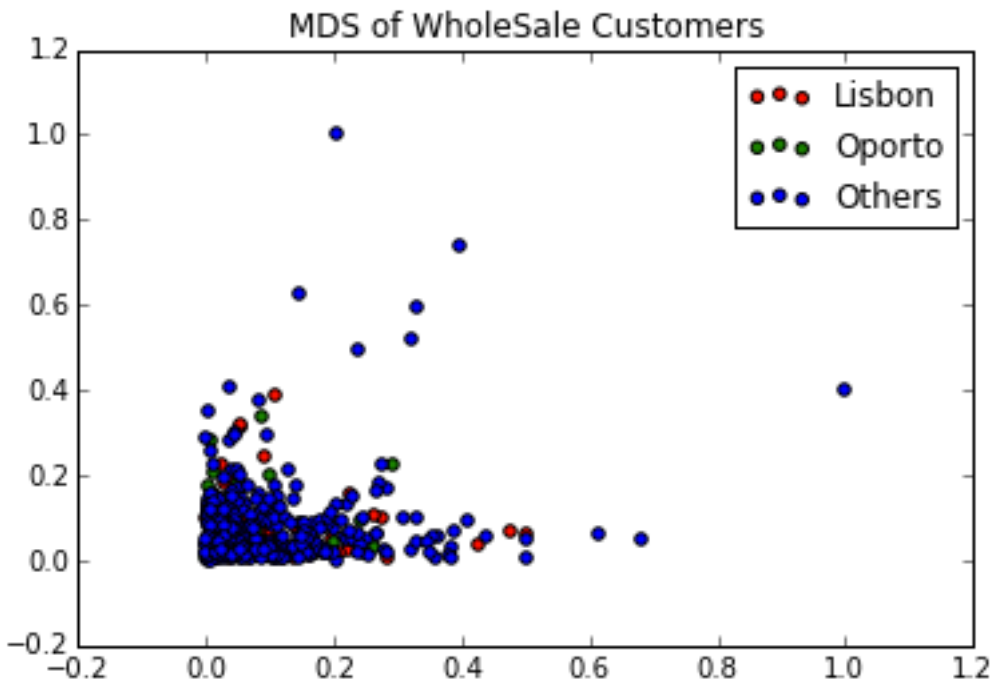


Figura 11: MDS - Lisboa/Oporto/Altres

Cal notar que vam intentar realitzar aquesta part de la pràctica amb l'algorisme dels apunts, obtenint una solució molt menys eficient.

Conclusió i comentaris

La valoració global de l'aprenentatge en aquesta PAC és, de nou, positiva. Em crida fortament l'atenció el no-agrupament en la classe procedència, tot i que pot ser degut a que aquesta variable no explica pas la dispersió de les dades. De nou, ha sigut de gran ajuda tenir la resolució de la pràctica de l'any anterior com a referència, i sobretot les funcions de la llibreria sklearn, que considero un gran recurs que pot facilitar enormement les tasques de data mining.