Data Mining and Machine Learning

LECTURE 5:

Feature Engineering: Data Reduction-Dimension Reduction

Dr. Edgar Acuna

Department of Mathematics

Universidad de Puerto Rico- Mayaguez

academic.uprm.edu/eacuna

Dimension Reduction

- *Feature Selection*: The main aim of doing feature selection is to reduce the dimensionality of the feature space, by selecting relevant and no redundant features. A feature is redundant when gives information contained in any other feature. A feature is irrelevant if gives very small amount of information. That is, feature selection selects “q" features from the entire set of “p" features such that q ≤ p. Ideally q <<< p.

- *Feature Extraction*: A smaller set of features is constructed by applying a linear (or nonlinear) transformation to the original set of features. The best known method is principal components analysis (PCA). Others: PLS, Principal curves.

We will consider only supervised classification problems.

Goal: Choose a small subset of features such that:

a) The accuracy of the classifier on the dataset does not vary in a significant way.
b) The resulting conditional distribution of a class C, given the selected vector feature G, is as close as possible to the original conditional distribution given all the features F.

Advantages of feature selection

- The computational cost of the classification will be reduced since the number of features will be less than before.
- The complexity of the classifier is reduced since redundant and irrelevant features are eliminated. Irrelevant Features are those that have none or very little contribution to the predicción Redundant features are those which information is contained in other features.
- It helps to deal with the “curse of dimensionality” effect. Some procedures require to have a large number of instances in comparison with the number of features

Steps of Feature selection

1. A **generation procedure**: The search of the optimal subset could be: complete, heuristic, random.
2. An **evaluation function**: Distance measures, Information measures, consistency measures, dependency measures, classification error rate).
3. A **stopping criterion**: A threshold, a prefixed number of iterations, a prefixed size of the best subset of features.
4. (Optional) A **validation procedure** to check whether the subset is valid .

Guidelines for choosing a feature selection method

- Ability to handle different types of features (continuous, binary, nominal, ordinal)
- Ability to handle multiple classes
- Ability to handle large datasets.
- Ability to handle noisy data.
- Low complexity time.

Categorization of feature selection methods (Li et al. 2016)

Evaluation Measures	***Generation***		
	Heuristic	***Complete***	***Random***
Distance	**Relief**	**Branch and Bound**	**-**
Information	**Trees**	**MDL**	**-**
Dependency	**Correlation**	**-**	**-**
Consistency	**FINCO**	**Focus**	**LVF**
Classifier Error rate	**SFS, SBS,SFFS**	**Beam Search**	**Boruta, GA**

The methods in the last row are also known as “wrapper” methods

Python Modules for feature selection

Scikit-learn has a Feature_selection class

Module scikit-feature, disponible at http://featureselection.asu.edu/.
It has around 40 feature selection algorithms, including Relief, F-score, Chi-Square, Entropy, mRMR and wrappers. But, it must be used with precaution because it has some bugs.

Module Sbkrebate:

Module Mlxtend:

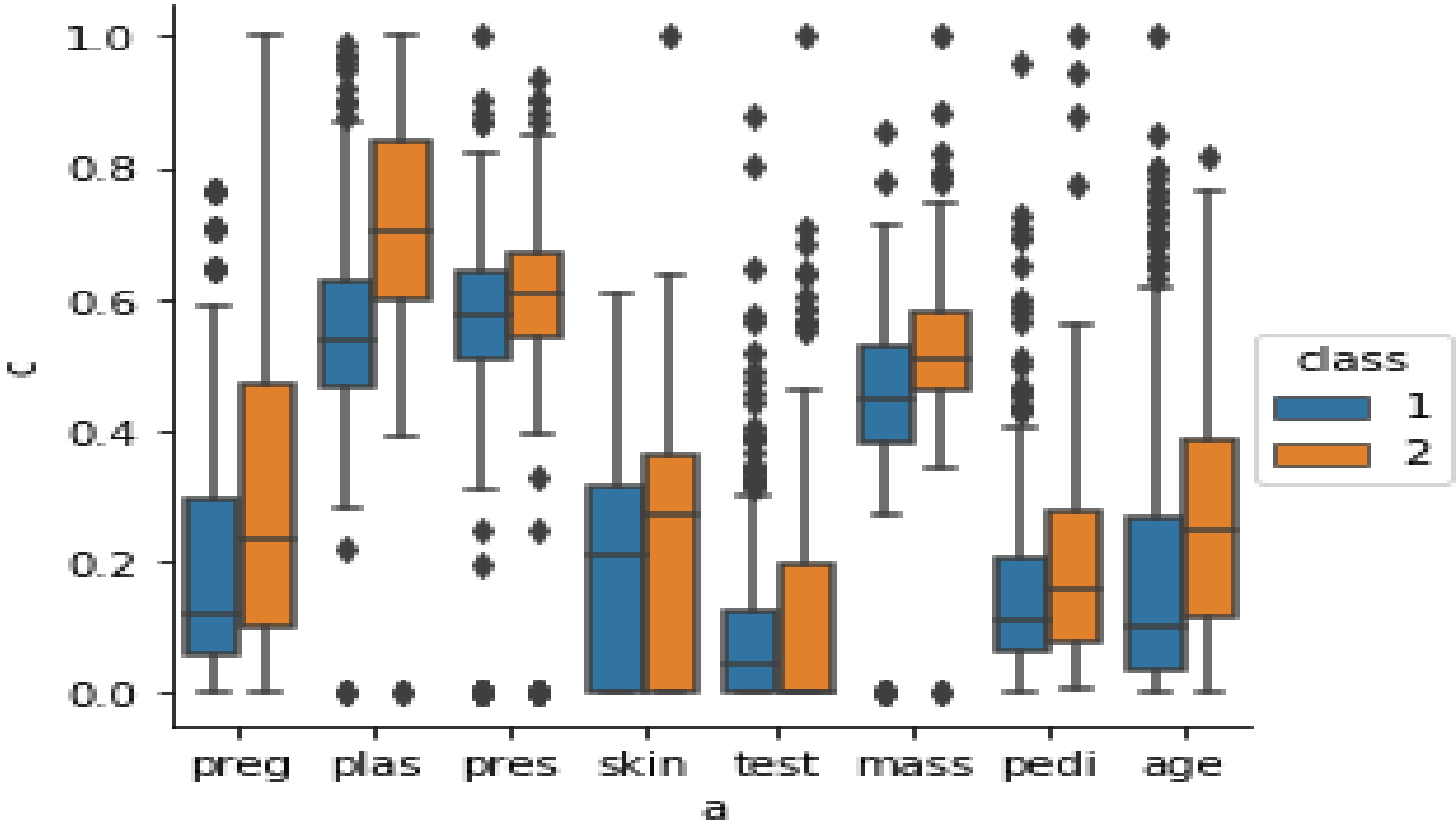
Using visualization to find the best features

The dataset Pima Indian Diabetes is available at the UCI Machine Learning Repository. It is a data frame with 768 instances and the following features:

V1: Number of times pregnant
V2:Plasma glucose concentration (glucose tolerance test)
V3:Diastolic blood pressure (mm Hg)
V4:Triceps skin fold thickness (mm)
V5:2-Hour serum insulin (mu U/ml)
V6:Body mass index (weight in kg/(height in m)\^2)
V7:Diabetes pedigree function
V8:Age (years)
V9:Class variable (1:tested positive for diabetes, 0: tested negative for diabetes)

We will use boxplots to find the best features

Finding the best predictors for Diabetes



V2(plas), V8(age), V6(mass) , V1(preg),V6(mass) y V7(pedi) seems to be the best predictors because their boxplots show differente behavior among the two classes

Feature selection using statistical tests

For two-class problem and continuous attributes the two-sample t-test can be used to detect best features. For the multiclass problem the F-test is suitable. In both cases the best attributes are the ones with the highest values for the statistical test values. When the p-values of the test are used we have to look for the features with the lowest p-values.

Similarly, for nominal o categorical attributes the Chi-Square test can be used.
All the above statistical tests assume distributional properties of the data, something that rarely holds, in spite of that their use continue.

Filter methods

They do not require a classifier, instead they use measures that allow us to select the features distinguishing the classes most.

- RELIEF
- Las Vegas Filter (LVF)
- Others: Branch & Bound, Focus,

These methods are used to remove irrelevant features.

The RELIEF method

- Kira and Rendell (1992) for two-class problem and generalized to multi-class problems by Kononenko (1994) and Kononenko, et al. (1997).
- Generates subsets of features heuristically.
- A feature has a relevance weight that is large if it can clearly distinguish two instances belonging to different classes but not two instances that are in the same class.
- Use a distance measure (Euclidean, Manhattan)

The RELIEF method (procedure)

- A given number ***Nsample*** (NS) of instances are selected randomly from the training set D containing F features.
- The relevance's weight W_j of each feature is initialized to zero.
- For each instance **x** selected, one must identify two particular instances:

Nearhit: The instance closest to **x** that belongs to its same class.

Nearmiss: The instance closest to **x** that belongs to a different class.

The RELIEF method (distances)

- Then the weigths W_j's (i=1,..F) are updated using the relation

$$W_j = W_j - \text{diff}(x_j, \text{Nearhit}_j)^2/NS + \text{diff}(x_j, \text{Nearmiss}_j)^2/NS$$

If the feature X_k is either nominal or binary then

- $\text{diff}(x_{ik},x_{jk}) = 1$ for $x_{ik} \neq x_{jk}$
  $= 0$ for the contrary case.

If the feature X_k is either continuous or ordinal then:

- $\text{diff}(x_{ik},x_{jk}) = (x_{ik}-x_{jk})/c_k$, where $c_k = \text{range}(X_k)$

Decision: If $W_j \geq \tau$ (a prefixed threshold) then the feature f_j is selected

Breast-Wisconsin dataset

- 699 instances, 9 features and two classes (benign or malign). 16 instances have been deleted because contain missing values.
- 1. Clump Thickness 2. Uniformity of Cell Size, 3. Uniformity of Cell Shape, 4. Adhesion Marginal Adhesion, 5. Single Epithelial Cell Size, 6. Bare Nuclei, 7. Bland Chromatin 8. Normal. nucleoli 9. Mitoses.
- Each feature has values ranging from 0 to 10.

Heart-Cleveland dataset

- 303 instances, 13 features and two classes. 6 instances have been deleted because contain missing values.
- 1. age, 2. sex, 3. type of chest pain, 4. trestbps, 5. chol, 6, fbs>120?, 7. estecg, 8. thalach, 9. exang(T/F), 10. oldpeak, 11. slope, 12. ca(number of vessels), 13. thal(normal, fixed, reversable).
- Continuous:1,4,5,8,10, Binaries: 2,6,9, Ordinals:12, Nominals:3,7,11,13

The refief method: multiclass problem

First a ***Nearmiss*** has to be found for each class different from **x**, and then their contribution is averaged using weights based on priors. The weights are updated using:

$$W_j = W_j - diff\,(x_j, \mathrm{Nearhit})^2 + \sum_{C \neq class(x_j)} \frac{P(C)}{1 - P(class(x_j))} diff\,(x_j, \mathrm{Nearmiss}(C))^2$$

Vehicle dataset

- 846 instances, 18 continuous features and four classes(double decker bus, Cheverolet van, Saab 9000 and an Opel Manta 400).
- [,1] Compactness [,2] Circularity [,3] Distance Circularity [,4] Radius ratio [,5] p.axis aspect ratio [,6] max.length aspect ratio[,7] scatter ratio [,8] elongatedness [,9] pr.axis rectangularity [,10] max.length rectangularity [,11] scaled variance along major axis[,12] scaled variance along minor axis[,13] scaled radius of gyration[,14] skewness about major axis[,15] skewness about minor axis[,16] kurtosis about minor axis[,17] kurtosis about major axis[,18] hollows ratio.

The Relief method (Cont)

Advantages:

It works well for noisy and correlated features.

Time complexity is linear on the number of features and on Nsample.

It works for any type of feature.

Disadvantages:

Removes irrelevant features but does not remove redundant features.

Choice of the threshold.

Choice of the Nsample.

The Las Vegas Filter (LVF) method

Liu and Setiono (1997)

- The subset of features are choosen randomly.
- The evaluation function used is an inconsistency measure.
- Two instances are inconsistent if they have the same feature values but belong to different classes.
- The continuous features of the dataset have to be discretized previously.

The Inconsistency measure

The inconsistency of a dataset with only non-continuous features is given by

$$\frac{\sum_{i=1}^{K} | D_i | - h_i}{N}$$

K: number of the different combinations of the N instances

$|D_i|$: Cardinality of the the i-th combination.

h_i: frecuency of the modal class on the i-th combination

Inconsistency: Example

```
> m1
    col1 col2 col3 col4 class
[1,]  1    2  2    1    1
[2,]  4    3  2    2    2
[3,]  4    3  2    2    1
[4,]  1    3  8    1    1
[5,]  9    3  8    2    2
[6,]  9    3  8    1    2
[7,]  9    3  1    2    1
> inconsist(m1)
[1] 0.1428571
```

The LVF Algorithm

```
Input : D = Dataset , p = Number of features , S= set of all
  features, MaxTries =Maximum number of trials , Threshold= $\tau$
  .
$C_{best}$=p , $S_{best}$ = S
For i= 1 to MaxTries
  $S_i$ = Subset of S choosen randomly.
  C = card($S_i$)
        If(C < $C_{best}$)
      {If Inconsistency($S_i$, D) <$\tau$
                    $S_{best}$= $S_i$ , $C_{best}$ = C        }
          If ( C = $C_{best}$ and Inconsistency ($S_i$, D)$\leq \tau$)
                    $S_{best}$= $S_i$.
Output : $S_{best}$
```

Disadvantages of LVF

- Choice of threshold. A small threshold will imply the selection of a larger number of features.
- A large number of iterations decreases the variability of the chosen subset but it slow down the computation.

Hybrid Methods

These methods remove irrelevant and redundant features at the same time. The most well known method is the Minimun Redundancy- Maximum Relevance method, mRMR (Peng, Long and Ding, 2005)

First, the most relevant features are determined using the F or t test if the attributes are coninuous or the Mutual Information metric if the atributes are categorical. Of course, attributes with the highest values are the most relevant. After that, the redundant features are eliminated using differences(MID) or Quotients(MIQ).

There are several modules in Python where mRMR has been implemented.

In the module skfeature from ASU is among the Information Theoretical based feature selection methods. It works only for discretized data.

Wrapper methods

Wrappers use the misclassification error rate as the evaluation function for the subsets of features.

- Sequential Forward selection (SFS)
- Sequential Backward selection (SBS)
- Sequential Floating Forward selection (SFFS)
- Others: SFBS, Take l-remove r, GSFS, GA, SA.

Sequential Forward Selection (SFS)

- Initially the best subset of features T is set as the empty set.
- The first feature entering T is the one with the highest recognition rate with a given classifier.
- The second feature entering T will be the one that along with the feature selected in the previous step produces the highest recognition rate.
- The process continues and in each step only one feature enters T until the recognition rate does not increase when the classifier is built using the features already in T plus each of the remaining features.

Sequential Backward selection(SBS)

- Initially the best subset of features T include all the features of the dataset
- In the first step we perform the classification without considering each of the feature, and we remove the feature where the recognition rate is the highest.
- The procedure continues removing one variable in each step until the recognition rates starts to decrease.

No efficient for nonparametric classifiers because has a high computing running time.

Sequential Floating Forward Selection (SFFS)

Pudil, et al (1994). It tries to solve the nesting problem that appears in SFS and SBS.

- Initially the best subset of features T is set as the empty set.
- In each step a new feature is included in T using SFS, but it is followed by a checking of a possible exclusion of features that are already in T. The features are excluded using SBS until the recognition rate starts to decrease.
- The process continues until the SFS cannot be done.

Recursive Feature Elimination

This method is a particular case of backward elimination. It was introduced by Guyon, et al (2002) and it is used along with the Support Vector Machine (SVM) classifier. But, its use has been extended to other classifiers like random Forest logistic regression, linear discriminante análisis an even linear regression.

The importance of a feature is given by the value of its coefficient in the model.

The function RFE of scikit learn is use to carry out RFE

Boruta

The Boruta algorithm is a wrapper feature selection method consisting of following steps:

i) Extend the dimensionality of the dataset by adding copies of all variables. These are permuted versions of the original features obtained by shuffling and are called shadow attributes.

ii) Run a random forest classifier on the extended dataset.

iii) Find the maximum importance value (threshold)among shadow attributes reported by random Forest, and then assign a hit to every feature that scored better than the threshold.

iv) Mark the attributes which have importance significantly lower than the threshold as ‘unimportant’ and permanently remove them from the dataset.

v) Mark the attributes which have importance significantly higher than the threshold as ‘important”.

vi) Remove all shadow attributes.

vii) Repeat the procedure a fixed number of times and choose only as important features those selected in the step v) at least un 50 percent of times.

The BorutaPy module performs feature selection using Boruta.

CONCLUDING REMARKS

- Among the wrappers the SFFS performs better than SFS : lowest percentage of features selected and almost same accuracy as SFFS. Fast computation.
- Wrappers are more effective than filters in reducing the misclassification error rate.
- The speed of computation of the filters is affected by the sample size and the number of classes.
- Decision Trees and Random Forest have an internal process to select the best features.

Principal Components Analysis (PCA)

The goal of Principal components analysis (Hotelling, 1933) is to reduce the available information.

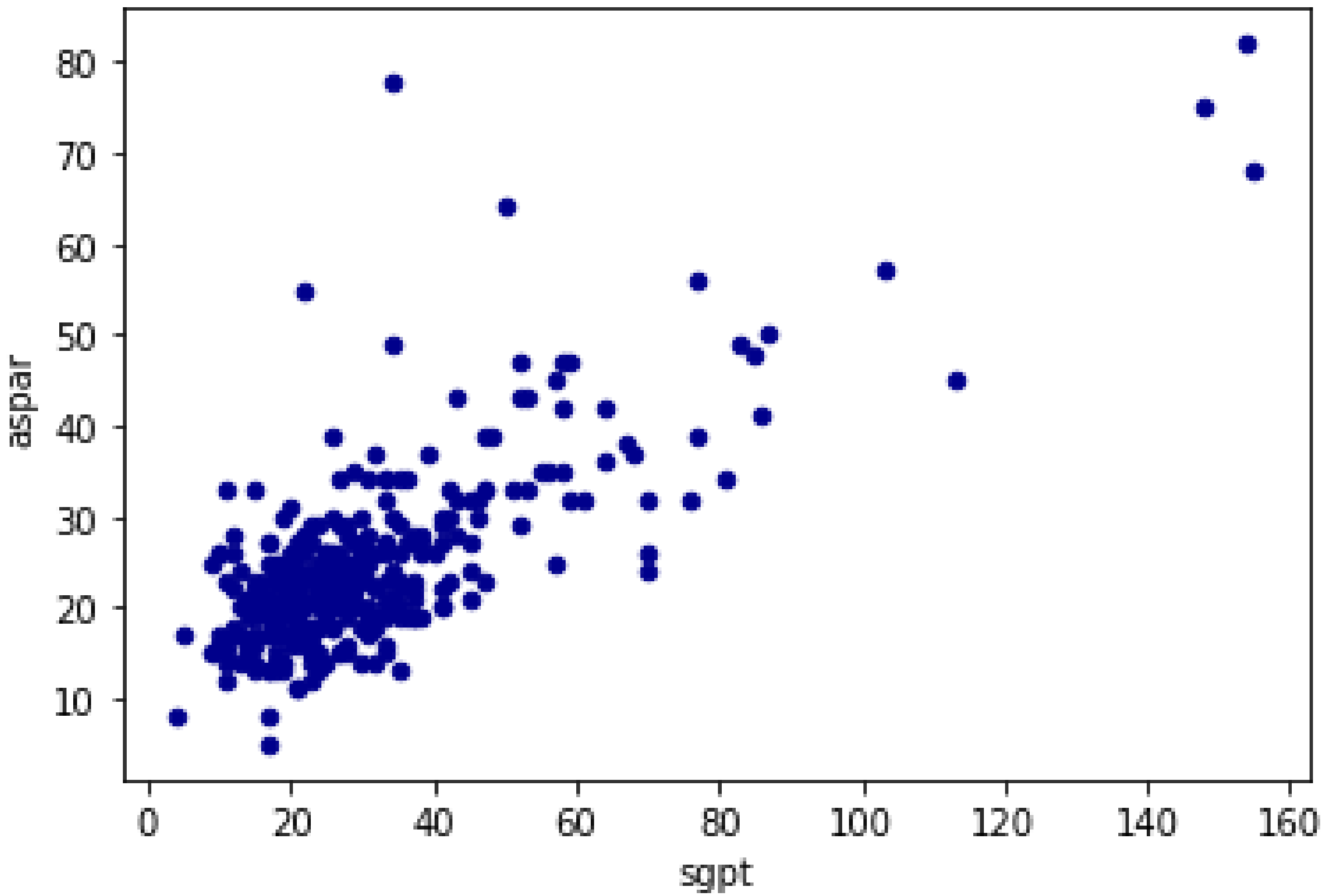
That is, the information contained in p features **X**=(X_1,….,X_p) can be reduced to **Z**=(Z_1,….Z_q),*with q<p* y where the new features Z_i's , called the ***Principal components*** are uncorrelated.

The principal components of a random vector **X** are the elements of an orthogonal linear transformation of **X**

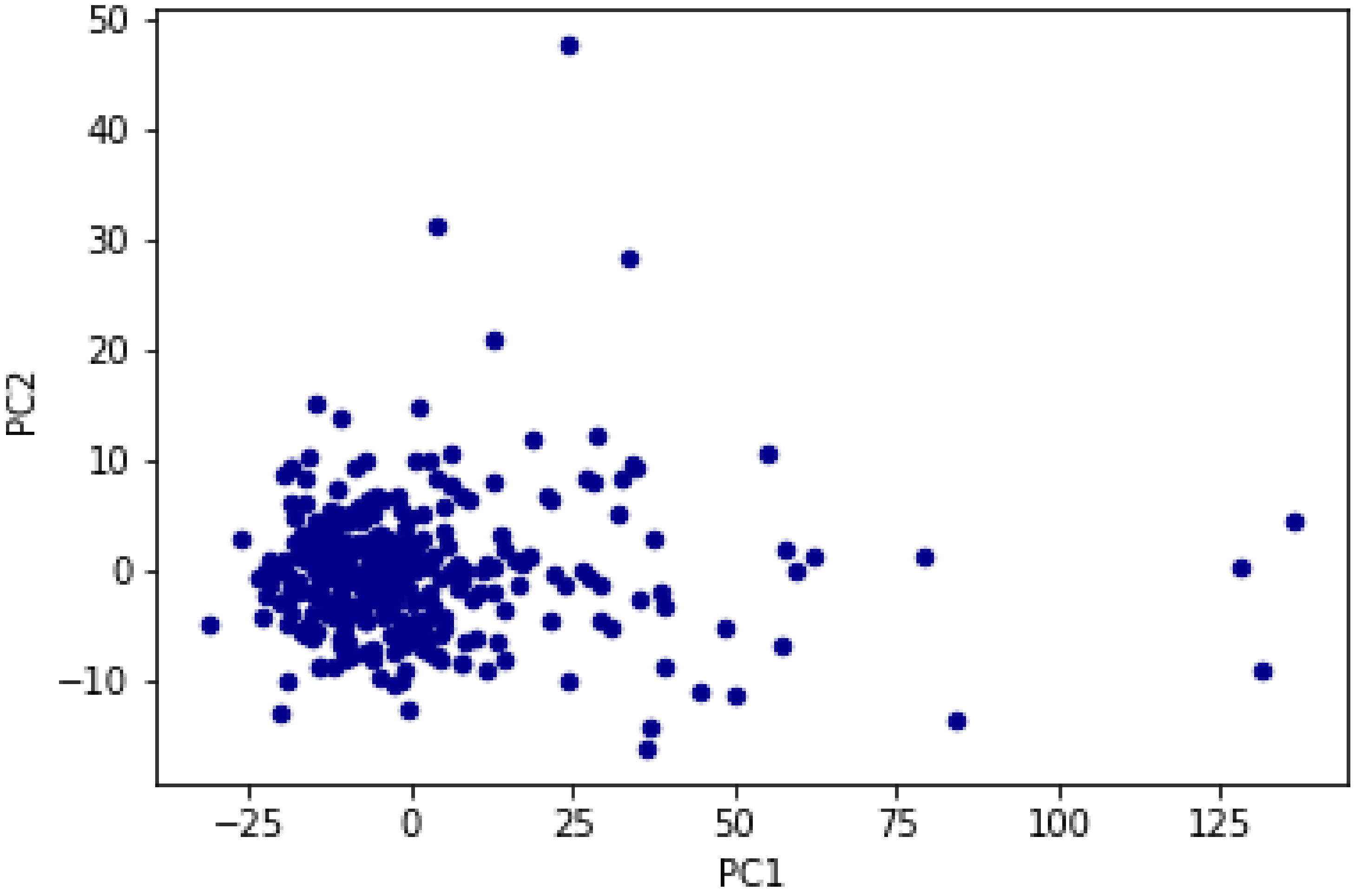
From a geometric point of view, application of principal components is equivalent to apply a rotation of the coordinates axis.

Example: Bupa (p=q=2)

```
#loading the data
url= "http://academic.uprm.edu/eacuna/bupa.dat"
names = ['mcv', 'alkphos', 'sgpt', 'aspar', 'gammagt', 'drinks', 'class']
data = pd.read_table(url, names=names,header=None)
y=data['class’]
# Considering only features 3 and 4
X1=X.iloc[:,2:4]
X1.head()
#Coomputing the coefficients of PC1 and PC2 also called loadings
pca=PCA(n_components=2)
principalComponents = pca.fit_transform(X1)
pca.components_
array([[ 0.92011644,  0.39164491],
      [-0.39164491,  0.92011644]])
```



Before PCA: Features sgpt and aspar show correlation



After PCA: Notice that PC1 and PC2 are uncorrelated

Finding the principal Components

To determine the Principal components Z, we must find an orthogonal pxp matrix V such that

i) $Z=X^*V$,

where X* is obtained by normalizing each column of X.

and ii) $Z'Z=(X^*V)'(X^*V) =V'X^{*\prime}X^*V$

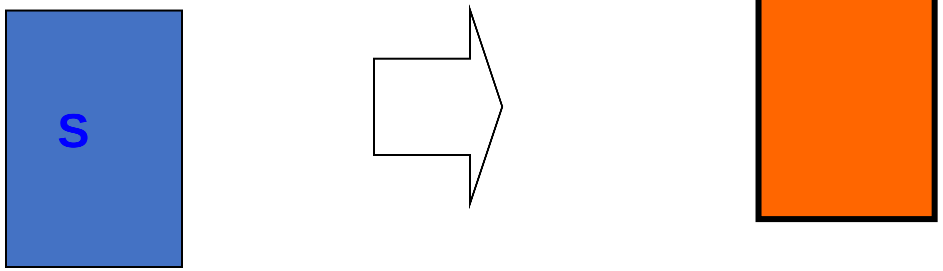
$=\text{diag}(\lambda_1,\ldots,\lambda_p)$

It can be shown that $VV'=V'V=I_p$, and that the λ_j's are the eigenvalues of the correlation matrix $X^{*\prime}X^*$.

V is found using singular value decomposition of $X^{*\prime}X^*$.

The matrix V is called the loadings matrix and contains the coefficients of all the features in each PC.

PCA AS AN OPTIMIZATION PROBLEM



X (nxp)

S

T(nxp)

Matrix of components

S=X'X, Covariance Matrix

$$\mathbf{T_k} = \underset{\gamma'\gamma=1}{\text{argmax}} \ \text{var} \ (X\gamma)$$

Subject to the orthogonality constrain

$$\gamma_j' \, \mathbf{S} \, \gamma_k = 0 \qquad \forall \;\; 1 \leq j < k$$

From (ii) the j-th principal component Z_j has standard deviation $\sqrt{\lambda_j}$ and it can be written as:

$$Z_j = v_{1j} X_1^* + v_{2j} X_2^* + \ldots.. + v_{pj} X_p^*$$

where v_{j1},v_{j2},…..v_{jp} are the elements of the j-th column in V.

The calculated values of the principal component Z_j are called the rotated values or simply the “scores”.

Reconstructing the original matrix from the PCA's

Usually the orthogonal matrix V is only de order k. Meaning that only k PCA are used.

$X^*=Z^*V^T$

and to obtain the original matrix X we need to add the mean vector, assuming that X* was only centered.

Choice of the number of principal components

There are plenty of alternatives (Ferre, 1994), but the most used are:

i) Choose the number of components with an acumulative proportion of eigenvalues (i.e, variance) of at least 75 percent.

ii) Choose up to the component whose eigenvalue is greater than 1. Use “Scree Plot”.

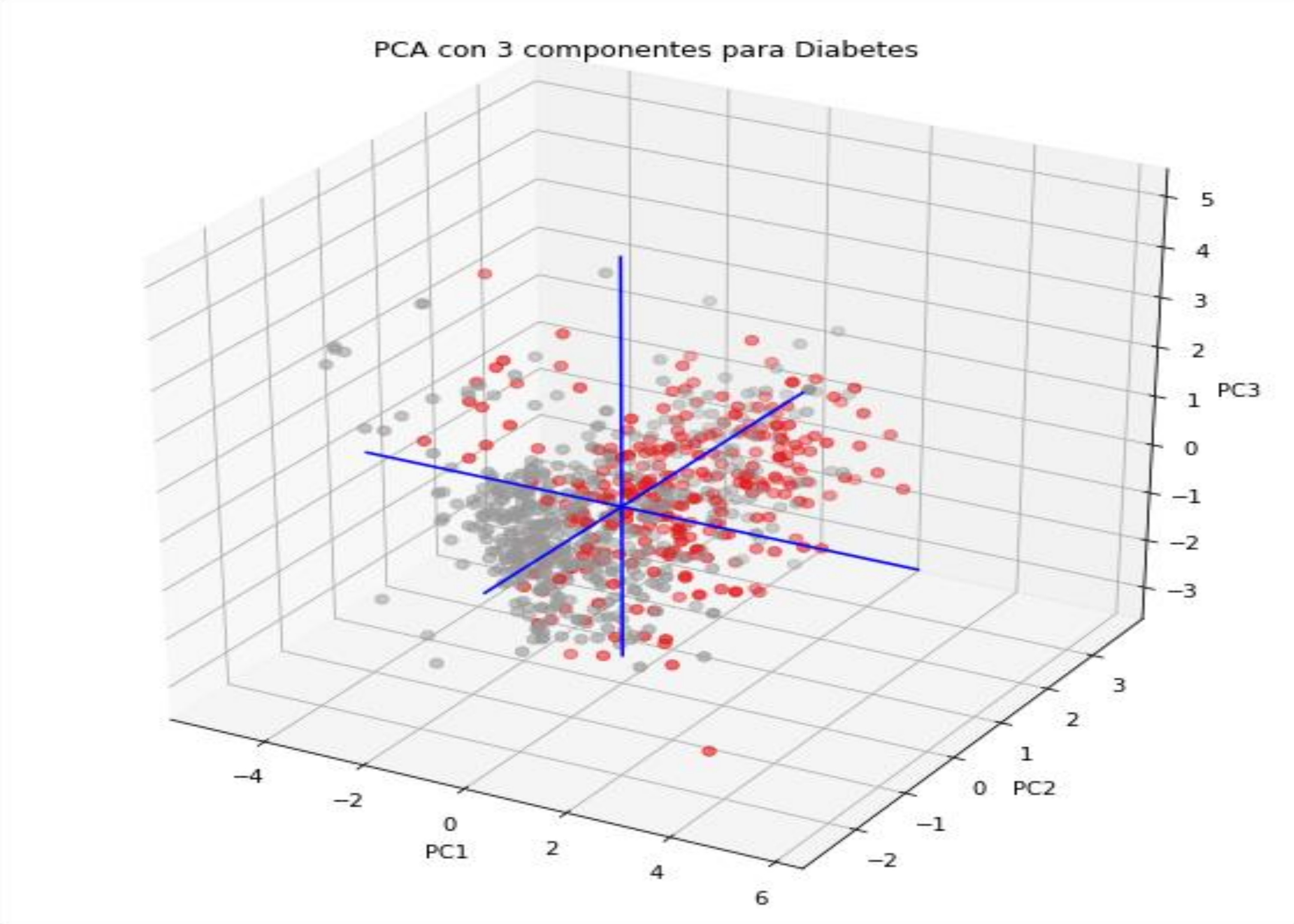
Example:Diabetes

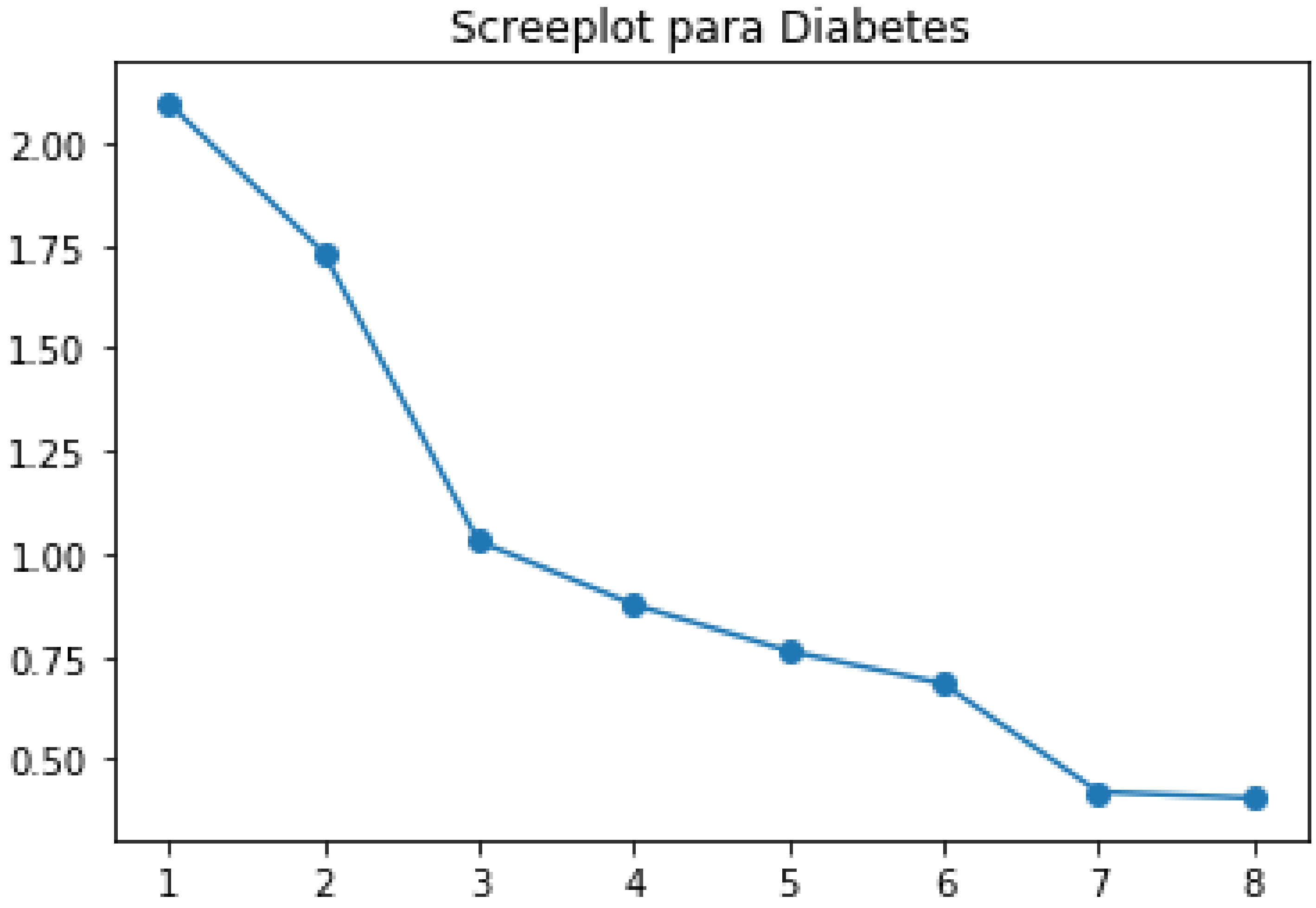
```
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(X)
print pca.explained_variance_
[ 2.09711056 1.73346726
 print pca.explained_variance_ratio_.cumsum()
[ 0.26179749 0.47819876]
```

Two PCs explain only 47.8% of the total variation

```
pca = PCA(n_components=3)
print pca.explained_variance_
[ 2.09711056 1.73346726 1.03097228]
print pca.explained_variance_ratio_.cumsum()
[ 0.26179749 0.47819876 0.60690249]
```

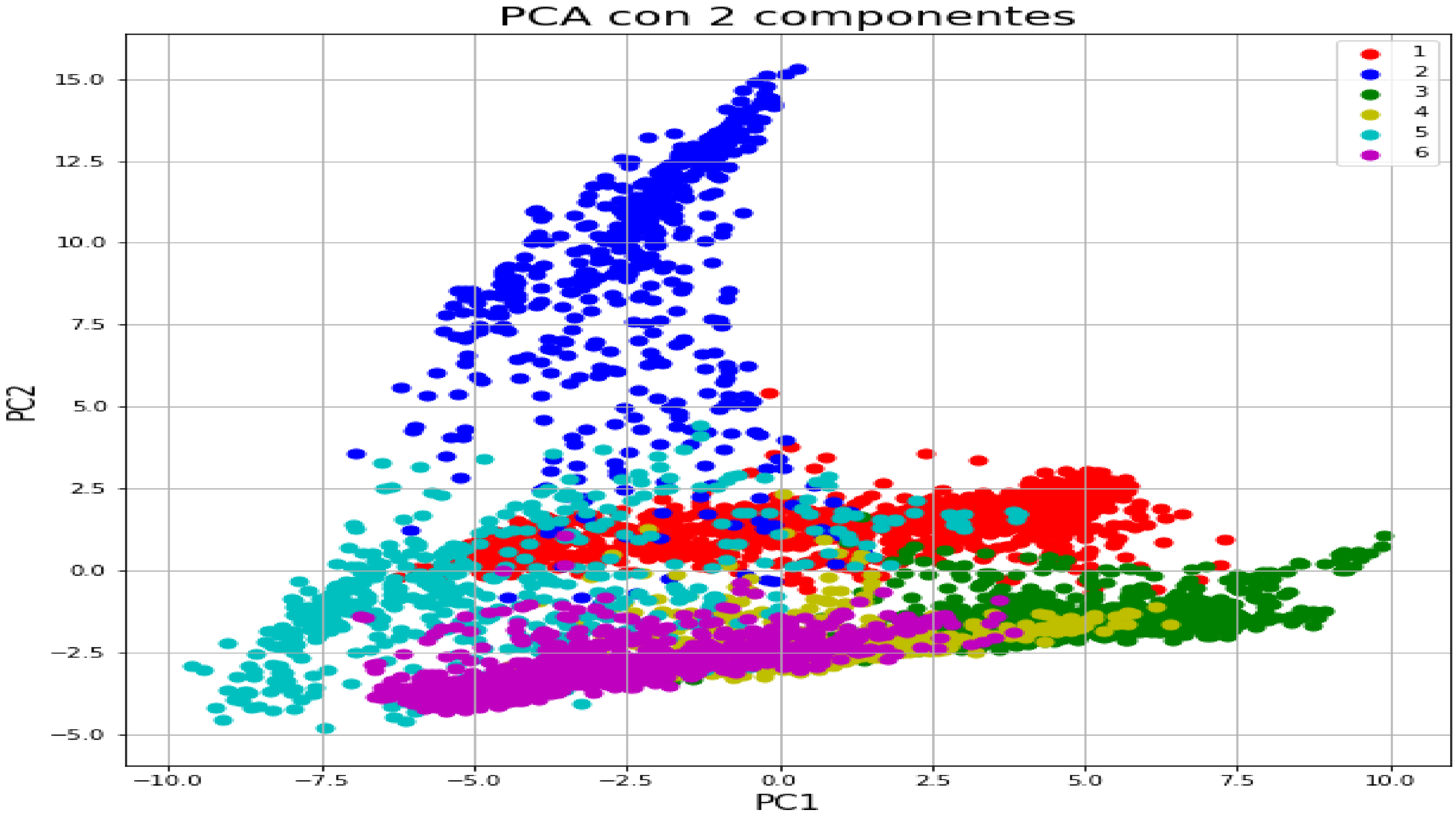
Three PCs explain 60.6% of the total variation


PCA con 3 componentes para Diabetes
PC1
−4
−2
0
2
4
6
PC2
−2
−1
0
1
2
3
PC3
−3
−2
−1
0
1
2
3
4
5



The scree-plot recommends to choose tres Principal Components

Example: Landsat



Remarks

- Several studies have shown that PCA does not give good predictions in supervised classification.
- Better alternatives: Generalized PLS (Vega and Acuna,2004) and Supervised PCA(Hastie, Tibshirani, 2004, Acuna and Porras, 2006).
- T-SNE (2008) is an alternative to PCA but mostly for visualization purposes.