Data Mining and Machine Learning

Lecture 3: Data Preprocessing

Dr. Edgar Acuna
Department of Mathematical Sciences

Universidad de Puerto Rico- Mayaguez

website: academic.uprm.edu/eacuna

Data Preparation-1

Data Preparation= Data Cleansing+ Feature Engineering

Data cleansing converts the raw data into one of good quaity and ready for analysis. It includes the following tasks:

- Select, Filter and removal of duplicates
- Sampling
- Data Partitioning: Creation of the training, the validation and the test sets.
- Data Normalizacion.
- Data Reduction
- Data Integration.
- Discretization (Binning)
- Imputation of missing values

Data Preparation-2

Feature Engineering:
It selects the right attributes to be used in the Machine Learning Algorithm.
It involves

- The use of domain knowledge of the data to select or create attributes.
- Feature Encoding
- Feature selection.
- Validation of how the features work with your model
- Feature Extraction: Principal component Analysis(PCA)

Data Preparation-3

Data Preparation: Data Preprocessing + Data Wrangling (Data Munging)

Data preparation can be classified in two, according to the moment of the analytic process when it is performed:

Data Preprocessing: Preparation of data directly after accessing it from a data source. Typically realized by a data scientist for initial transformations, aggregations and data cleansing. This step is done before the interactive analysis of data begins. It is executed once.

Data Wrangling: Preparation of data during the interactive data analysis and model building. Typically done by a data scientist to change views on a dataset and for features engineering. This step iteratively changes the shape of a dataset until it works well for finding insights or building a good analytic model.

Why Preprocess the Data?

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., *Occupation*=" " (missing data)
  - noisy: containing noise, errors, or outliers
    - e.g., *Salary*="−10" (an error)
  - inconsistent: containing discrepancies in codes or names, e.g.,
    - *Age*="42", *Birthday*="03/07/2010"
    - Was rating "1, 2, 3", now rating "A, B, C"
    - discrepancy between duplicate records

Why Preprocess the Data?

- Intentional (e.g., *disguised missing* data)
  - Jan. 1 as everyone's birthday?
- It data does not have quality then results Machine Learning algorithms do not have quality! (GIGO Principle)
  - The quality of decisions is based on data quality.
  - A good Data Warehouse is built on integration of data having high quality.

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values (Noise) may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention.

Two Sine Waves **Two Sine Waves + Noise**

Outliers

- Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set

Major Tasks in Data Preprocessing

- **Data cleaning**
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data reduction**
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results. It includes Dimensionality reduction, Numerosity reduction, Data compression
- **Data transformation**
  - Normalization
- **Data discretization(binning)**
- **Data integration:** Integration of multiple databases, data cubes, or files

Data Cleaning

- Tasks:
  - Fill in missing values valores.
  - Outlier Detection
  - Smoothing of noisy data.
  - Fixing inconsistences.

Handling of Missing Values

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred.

Handling of Missing Values (cont)

- Several methods have been proposed in literature to treat missing data. Many of these methods were developed for dealing with missing data in sample surveys.
- Bello (1995), MV in regression
- Troyanskaya et al (2001), MV in unsupervised classification.
- Studies related to supervised classification
  - Chan and Dunn (1972) – Imputation on LDA for two class problems.
  - Dixon (1975) - k-nn imputation technique for dealing with missing values in supervised classification.
  - Tresp (1995)- missing value problem in a supervised learning context for neural networks.

Handling of Missing Values (cont)

- Impact of missing data
  - 1% missing data – trivial
  - 1-5% - manageable.
  - 5-15% - requires sophisticated methods
  - 15% - detrimental interpretation.

Example: The Census dataset

Also it is known as **Adult.**

48842 instances, it contains continuous, ordinals and nominal features (training set =32561 , test set=16281).

Remain 45222 instances after deleting instances containing at least one missing value. (Training set=30162, test=15060).

Size in Bytes=Training set 3.8MB, Test set 1.9MB.

Available at: http://archive.ics.uci.edu/ml/

Donors: Ronny Kohavi y Barry Becker (1996).

Features in Census

- 1- age: continuous.
- 2- workclass: (Type of organization in which the person is employed) Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked. Nominal
- 3- fnlwgt (final weight) : Continuous.
- 4- education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. Ordinal.
- 5- education-num: continous.
- 6- marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. Nominal
- 7- occupation: Nominal

Features in Census

- 8-relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. Nominal
- 9-race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. Nominal
- 10-sex: Female[0], Male[1]. Nominal-Binary.
- 11-capital-gain: continuous.
- 12-capital-loss: continuous.
- 13-hours-per-week: continuous.
- 14-native-country: nominal
- 15 Salary: >50K [2], <=50K [1].

The census dataset

age	employme	education	edun	marital	…	job	relation	race	gender	hour	country	wealth
					…							
39	State_gov	Bachelors	13	Never_mar	…	Adm_cleric	Not_in_fam	White	Male	40	United_Sta	poor
51	Self_emp_	Bachelors	13	Married	…	Exec_man	Husband	White	Male	13	United_Sta	poor
39	Private	HS_grad	9	Divorced	…	Handlers_c	Not_in_fam	White	Male	40	United_Sta	poor
54	Private	11th	7	Married	…	Handlers_c	Husband	Black	Male	40	United_Sta	poor
28	Private	Bachelors	13	Married	…	Prof_speci	Wife	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	…	Exec_man	Wife	White	Female	40	United_Sta	poor
50	Private	9th	5	Married_sp	…	Other_serv	Not_in_fam	Black	Female	16	Jamaica	poor
52	Self_emp_	HS_grad	9	Married	…	Exec_man	Husband	White	Male	45	United_Sta	rich
31	Private	Masters	14	Never_mar	…	Prof_speci	Not_in_fam	White	Female	50	United_Sta	rich
42	Private	Bachelors	13	Married	…	Exec_man	Husband	White	Male	40	United_Sta	rich
37	Private	Some_coll	10	Married	…	Exec_man	Husband	Black	Male	80	United_Sta	rich
30	State_gov	Bachelors	13	Married	…	Prof_speci	Husband	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	…	Adm_cleric	Own_child	White	Female	30	United_Sta	poor
33	Private	Assoc_acc	12	Never_mar	…	Sales	Not_in_fam	Black	Male	50	United_Sta	poor
41	Private	Assoc_voc	11	Married	…	Craft_repai	Husband	Asian	Male	40	*MissingVa	rich
34	Private	7th_8th	4	Married	…	Transport_	Husband	Amer_India	Male	45	Mexico	poor
26	Self_emp_	HS_grad	9	Never_mar	…	Farming_fi	Own_child	White	Male	35	United_Sta	poor
33	Private	HS_grad	9	Never_mar	…	Machine_o	Unmarried	White	Male	40	United_Sta	poor
38	Private	11th	7	Married	…	Sales	Husband	White	Male	50	United_Sta	poor
44	Self_emp_	Masters	14	Divorced	…	Exec_man	Unmarried	White	Female	45	United_Sta	rich
41	Private	Doctorate	16	Married	…	Prof_speci	Husband	White	Male	60	United_Sta	rich
:	:	:	:	:	:	:	:	:	:	:	:	:

Reading the data file using Python

```
import pandas as pd
df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data', header=None, sep=',‘,na_values=“ ?”)
df.columns=['v1', 'v2', 'v3', 'v4',v5','v6','v7','v8','v9','v10','v11','v12','v13','v14','class’]
```

Another way:

```
 import pandas
names=['v1','v2','v3','v4','v5','v6','v7','v8','v9','v10','v11','v12','v13','v14','clase']
data=pandas.read_csv('c://PW-PR/census.csv',names=names)
print(data.shape)
(32562, 15)
data.describe()
```

Pyhton functions dealing with missing values

Asssuning that df is a Pandas dataframe

- To find out columns containing missing values
  colmiss=df.columns[df.isnull().any()].tolist()
- To find out rows containing missing values
  rowmiss=df.index[df.isnull().T.any()].tolist()
- To find out the percentage of rows with missing values
  df.isnull().T.any().sum()*100/len(df)
- To find out the percentage of missing values per column
  df[colmiss].isnull().sum()*100/len(df)
- Deleting all the rows containing missing values
  dfclean=df.dropna()
  print dfclean.shape
  (30162, 15)

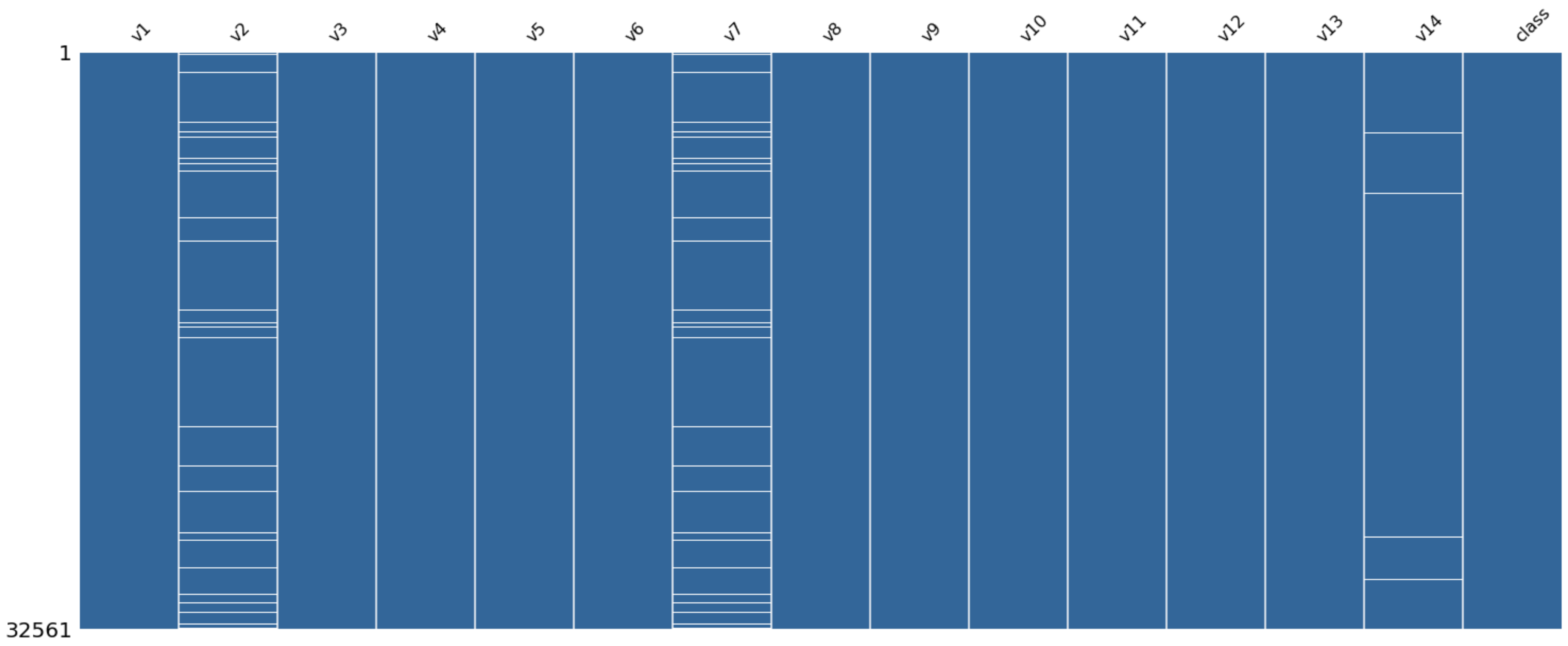
A “clean” function

- We can build a function for deleting in a sequential way, columns and rows with a large amount of missing values. Something like:

  clean(df,tol.col=.5,tol.row=.3)

In here, from the dataframe df, we are deleting columns with at least 50 % of missing values and then rows with at least 30% of missings.

Visualizing the missing values (missingno module)



Modules in Python to impute missing values

The module scikit-learn has a class named imputer to carry out inputation.

The function fill.na in pandas does imputation.

Also, there are two other modules for imputation

Fancyimpute

impyte

Both of them run only in Python 3.

Treating Missing values

- Case/Pairwise Deletion. Ignore the tuple: usually done when class label is missing (assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably.
- Parameter estimation, where Maximum likelihood procedures that use variants of the Expectation-Maximization algorithm can handle parameter estimation in the presence of missing data.
- Imputation techniques, where missing values are replaced with estimated ones based on information available in the data set.

Treating missing values

- Mean Imputation: Replace the missing value for a given feature(attribute) by the mean of all known values of the attribute in the class where the instance with the missing attribute belongs.
- Median Imputation: Since the mean is affected by the presence of outliers it seems natural to use the median instead just to assure robustness. In this case, the missing value for a given feature is replaced by the median of all known values of the attribute in the class where the instance with the missing feature belongs.

Mean Imputation using Pandas:

```
df.fillna(df.mean())
df.apply(lambda x: x.fillna(x.mean()),axis=0)
```

K-nn Imputation

In this method the missing value in a instance of a given attribute is replaced by the average(or weighted average) of the attributes values in the instances that are closer to the instance (nearest neighbors) having the missing value. The clossenes is determined by a similarity measure usually the euclidean distance.

For categorical atributes, the missing value is replaced by the mode of the atributes values in the nearest neighbors.

The parameter k represents the number of neighbors.

Scikit-learn include a class knnimputer for knnimputation

.

K-nn Imputation

A1	A2	A3	A4	Clase
4	5	6	NA	1
5	1	1	4	1
7	9	2	5	1
8	2	8	5	1
6	4	2	6	1

There is a missing value in record 1. We will predict this NA value using knn. Dist(r1,r2)= sqrt[$(4-5)^2+(5-1)^2+(6-1)^2$]= 6.48, Dist (r1,r3)=6.40, Dist(r1,r4)=5.38, Dist(r1,r5)= 4.58. There is a function distance in the module scipy.spatial. Also numpy.linalg,norm compute the Euclidean distance between to arrays. Using k=1 nearest neighbor, this is record r5, NA would be replaced by 6, using k=3 nearest neighbors: r3,r4 y r5, NA would be replaced by the average value of 5, 5 and 6 that is 5.33

K-nn Imputation (cont)

- The method can not be applied if all the rows have at least a missing value.
- Usually, k is taken equals to 10. The k value is at most equals to the number of complete rows.Usualmente, se toma k igual a 10.
- If the dataset contains numerical and categorical atributes the the euclidean distance can be replaced by the Gower distance.

.

Other imputation methods

- Hot deck and Cold deck (https://nces.ed.gov/)
- Prediction models: Lniear Regression(continuous feature), Logistic regression(Binary feature), Multinomial Logistic (nominal features). The attribute with missing values is used as the response variable and the remaining atributes are considered as predictors

Drawbacks: It can créate bias. It requires high correlation among response and predictors. Slow computation.

- The EM algorithm
- Decision trees have theirs own approach to handle missing values.

Other imputation methods (cont)

- Multiple Imputation. The missing values are imputed several times by choosing randomly from the observed values. The final prediction is obtained by combining the individual predictions.
- The SVD method ($X=U\Sigma V'$). It is iterative. Initially missing values in the data matrix X are replaced by the mean of the corresponding column and in each iteration the SVD of the matrix X is found. The matrix Σ contains the singular values of X. Only the k top singular values are used and the remaining are set to zero. Then, $U\Sigma V'$ gives only an approximation to X. The iterative process continues until the norm of the matrix X-approx(X) converges.
- The above approaches are too expensive.
- Also, missing values can be imputed using Deep Learning algorithms based on Autoencoders Networks.

Effect of the treatment of Missing values

- For datasets with an small amount of missing values, say less than 1 percent, deleting the cases containing missing values does not have much affect on the performance of the machine learning algorithm. However the variability of the estimated error increases.
- There is not much difference between mean imputation and median imputation.
- The effect of the missing values depends on the way that they are distributed on the table and its location with respect to the most important features.
- The percentage of instances containing missing values has more effect than the percentage o table cells containing missing values on the performance of the algorithm.

Preprocessing- Feature Encoding

Values of Categorical Features are converted into numerical values.

There are several strategies:

- 1-LabelEncoder
- 2-One Hot Encoder
- 3-Neural Networks Embeddings. An embedding is a mapping of a discrete

(categorical) variable to a vector of continuous numbers. In the context of neural networks, embeddings are lowdimensional, learned continuous vector representations of discrete variables.

Example: Predicting a bank decision to offer a loan to a customer for buying a car

Sexo	Familia	CasPropia	AnosEmpleo	Sueldo	StatustMarital	Prestamo
Hombre	3	No	17	2500	Soltero	No
Mujer	5	Si	10	3000	Casado	Si
Mujer	4	No	15	2000	Viudo	No
Hombre	3	Si	16	2800	Soltero	Si
Hombre	6	Si	11	4000	Viudo	Si
Mujer	4	Si	26	3200	Soltero	Si
Mujer	2	Si	14	1800	Soltero	No
Hombre	5	Si	10	3750	Casado	Si
Hombre	6	No	18	2970	Divorciado	No
Hombre	4	Si	12	3350	Divorciado	No
Hombre	1	No	23	1950	Soltero	No
Mujer	2	Si	25	2740	Soltero	Si
Mujer	3	No	7	3100	Soltero	Si
Hombre	5	Si	5	3845	Divorciado	No
Hombre	3	No	13	3200	Casado	Si
Mujer	3	Si	9	2800	Soltero	No
Hombre	2	No	6	3200	Soltero	Si
Hombre	3	Si	7	3815	Viudo	Si
Mujer	2	Si	11	2980	Divorciado	No
Hombre	4	Si	15	2850	Viudo	Si
Mujer	1	No	6	3125	Divorciado	No
Hombre	1	No	8	3500	Soltero	Si
Hombre	4	No	22	4500	Divorciado	Si
Hombre	2	Si	10	3200	Casado	Si
Hombre	3	Si	9	3000	Casado	Si

Preprocessing- Feature Encoding (cont.)

For instance, the categorical attribute: StatusMarital, in the previous dataset can be enconded into

i)Casado=0, Divorciado=1, Soltero=2,Viudo=3 using a Label Encoder process that follows an alphabetic ordering.

ii) One Hot encoder it will require to use 4 Binary features one for each value of the nominal attribute. Thus, the vector (1,0,0,0) represents Casado, (0,1,0,0) Divorciado, (0,0,1,0): Soltero, and (0,0,0,1):Viudo.

Another option is to consider only three binary features (1,0,0): Casado, (0,1,0):Divorciado; (0,0,1):Soltero and (1,1,1): Viudo.

Preprocessing- Feature Encoding (cont.)

Neural network embeddings are useful because they can reduce the dimensionality of categorical variables and meaningfully represent categories in the transformed space.

https://www.geeksforgeeks.org/feature-encoding-techniques-machine-learning/

https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

https://medium.com/analytics-vidhya/different-type-of-feature-engineering-encoding-techniques-for-categorical-variable-encoding-214363a016fb

Preprocessing-Normalization (Feature Scaling)

- Data normalization consists in scaling the attribute values of the data into an small specified range, such as -1 to 1 or 0 to 1.
- Also, it is known as range normalization or Feature Scaling
- There is also variance normalization but mostly is used in bioinformatics

Reasons for Normalizing

- Normalizing the input data will help to speed up the learning phase
- Attributes with initially large ranges will outweigh attributes with initially smaller ranges then dominate the distance measure. For instance, the k-nearest neighbor classifier using the Euclidean distance measure depends of all input attributes being scaled equally.
- Some kind of data normalization also may be necessary to avoid numerical problems such as loss of precision due to numerical overflow.

Z-score Normalization

The V values are normalized based on the mean and the standard deviation

$$V' = (V - mean)/std$$

This method works well in cases when the minimum and maximum values of the attributes are unknown or when there are outliers that have great effect on the range of the attributes.
It can be done using pandas or using the class StandardScaler from scikit-learn

Min-Max Normalization

It performs a linear transformation on the original data. V values are transformed into the specific interval [newmin, newmax] using

$$V'=(V-min)*(newmax-newmin)/(max-min)+newmin$$

The advantage of this method is that it preserves all the relationships of the data values exactly. It does not introduce any potential bias into the data. The disadvantage Is that it will encounter “out of bounds” error when any future case falls outside the original data.

It can be done either with Pandas or with the class MinMaxScaler from scikit-learn.

Normalization by decimal scaling

It normalizes by moving the decimal point of the values.
The number of decimal points moved depends on the maximum absolute value. The normalized value v' is given by

$V' = V/10^j$ where j is the smallest integer such that $Max(|V'|) < 1$.

It is useful only when the attributes values are greater than 1 in absolute value.

Sigmoidal Normalization

It transforms the input data nonlinearly into the range -1 to 1 using a sigmoid function
The transformed values are obtained by

$$v'=(1-e^{-a}/(1+e^{-a})$$

Where a=(V-mean)/std

Data points within a standard deviation of the mean are mapped to the almost linear region of the sigmoid function. Outlier points are compressed along the tails of the sigmoid function.
Sigmoidal transformation is specially appropriate when you have outlier data points that you wish to include in the dataset. It prevents the most occurring values from being compressed into essentially the same values without losing the ability to represent very outlier values.

Softmax Normalization

It is so called because it reaches "softly" towards its maximum and minimum values, never quite getting there.
The transformation is more or less linear in the middle range, and has a smooth nonlinearity at both ends. The whole range output covered is 0 to 1 and the transformation assures that no present values lies outside this range.
The normalized value v' is given by

$$V' = 1/(1+e^{-a}) \quad \text{where } a = (V\text{-mean})/\text{std}$$

It is useful only when the attributes values are greater than 1 in absolute value.