Cluster 2018

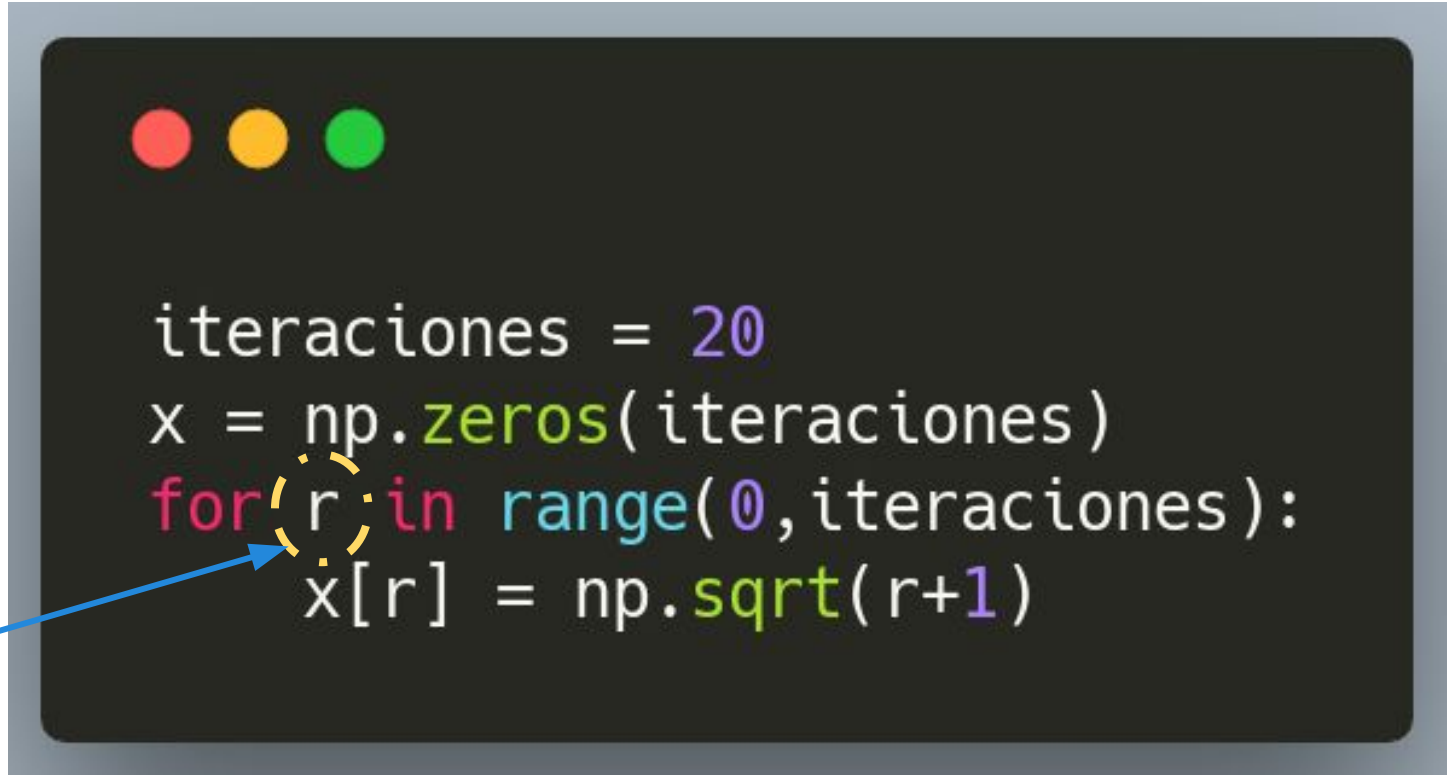
Ciencia de Datos en Ingeniería Industrial

clase_02

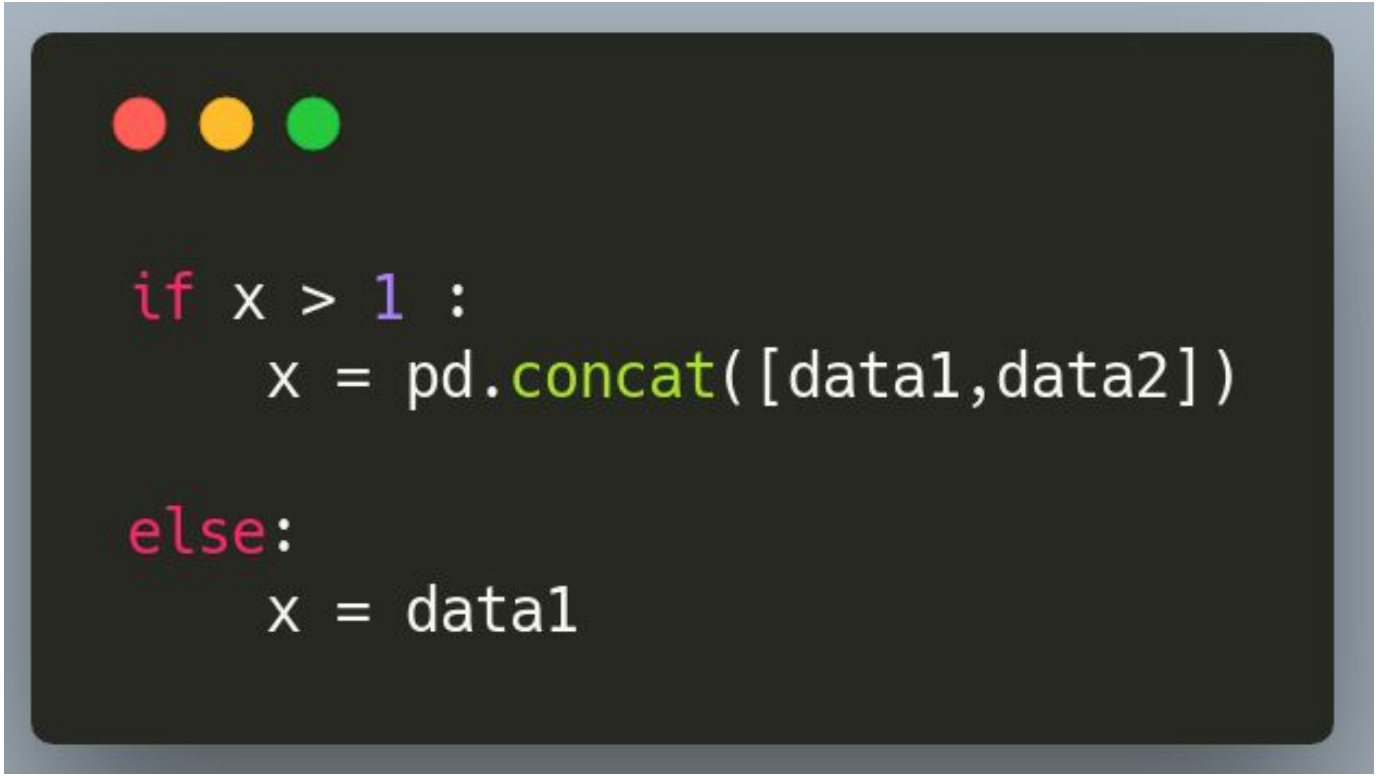
agenda_clase_02

- Python: For, IF, functions
- Intro Scikit-Learn
- Categorical Variables: Dummies
- Auto-scaling
- EDA Tincho: bicicletas (geo data)
- EDA Agus: aceros (merge, concat, join)
- Practica en clase

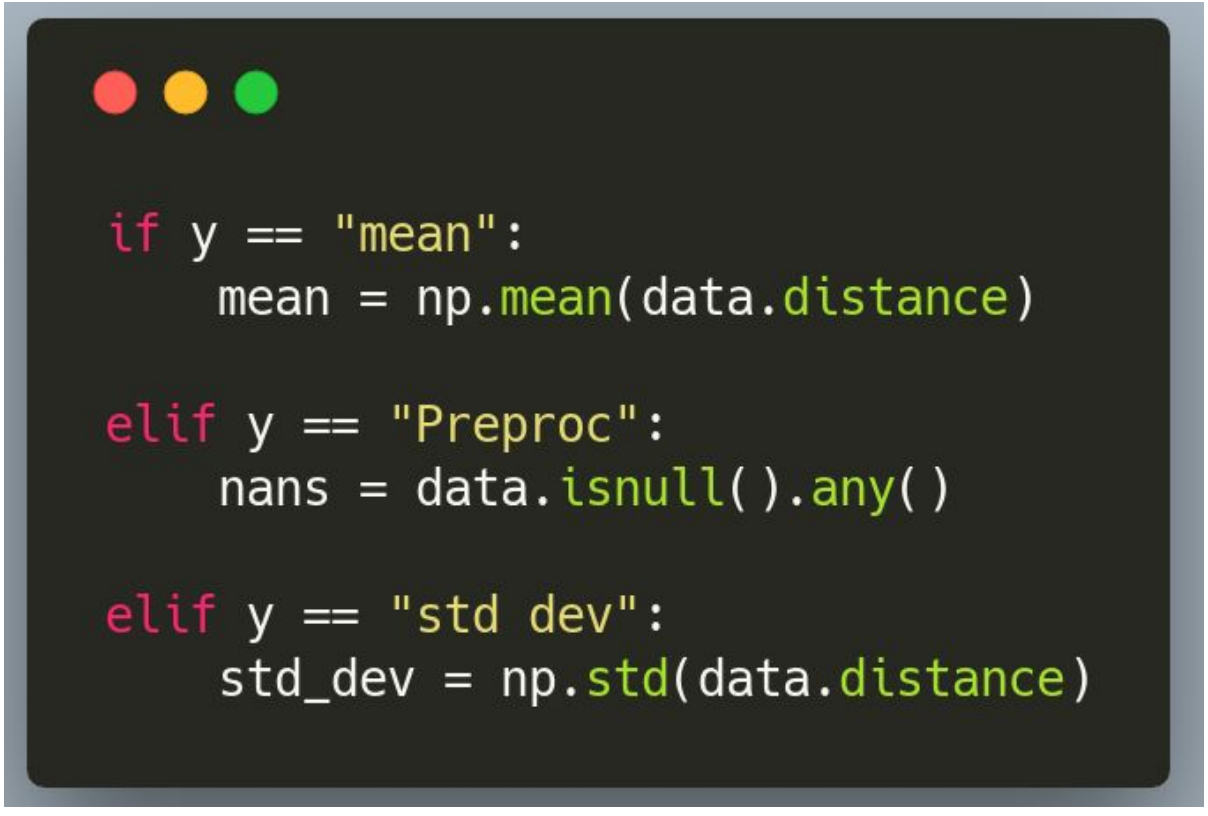
‘for’ loops in python



‘if’ statements in python



‘if’ statements in python

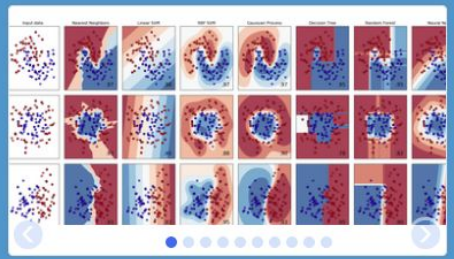


functions in python

```
def dot_product( x1, x2 ):
    "Esta funcion calcula el producto interno de 2 matrices"
     dotprod = np.dot(x1,x2.T)
     return dotprod

a = dot_prod(x_febrero, x_marzo)
```

Intro scikit-learn

Intro scikit-learn



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

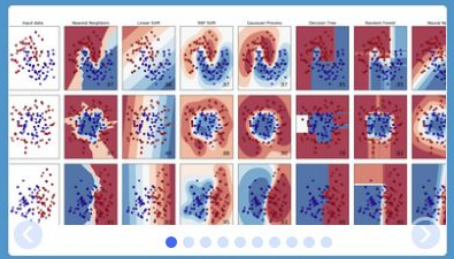
Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Intro scikit-learn



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Feature Engineering: Categorical Variables (Dummies)

Edad	Altura	**Sexo**
18	1.70	Masculino
24	1.60	Femenino
30	1.90	Femenino
28	1.5	Masculino

Edad	Altura	Sexo	**Masculino**	**Femenino**
18	1.70	Masculino	1	0
24	1.60	Femenino	0	1
30	1.90	Femenino	0	1
28	1.5	Masculino	1	0

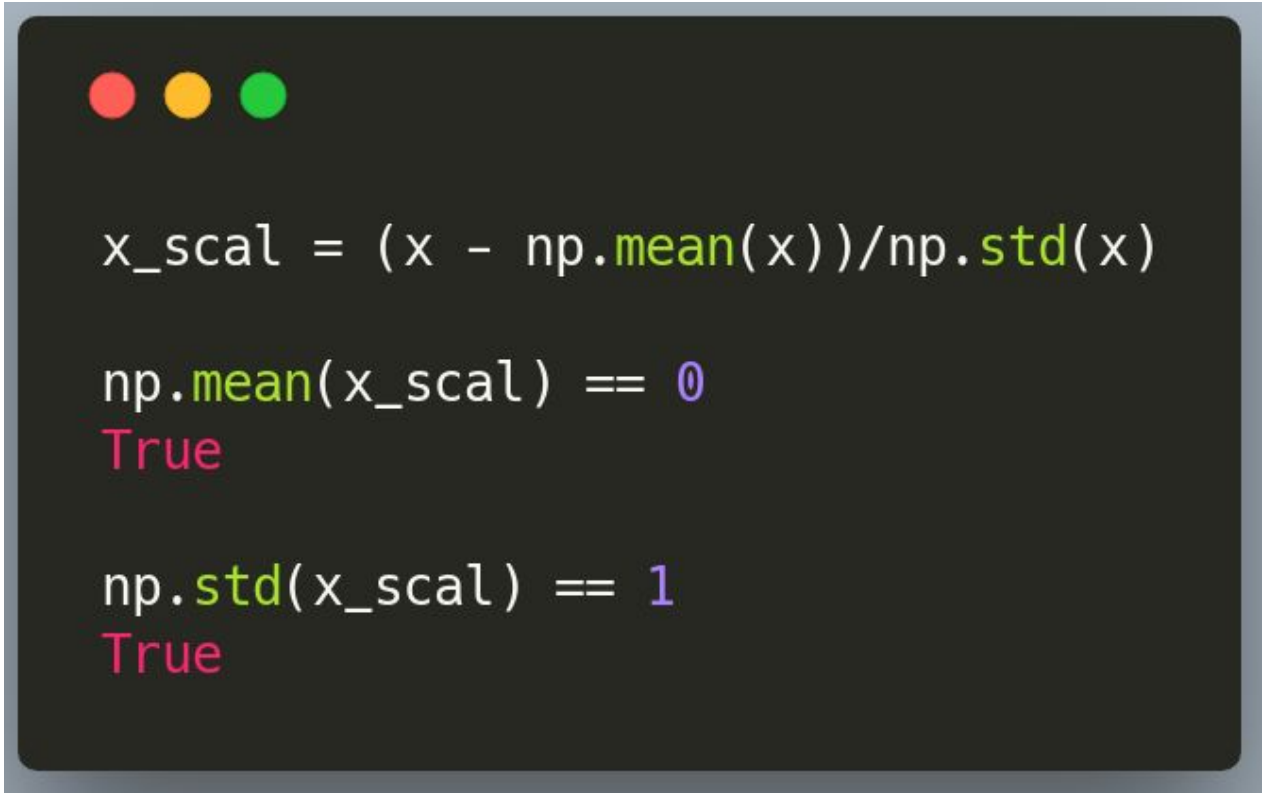
Feature Engineering: Categorical Variables (Dummies)

```
# 1 Creamos un dataframe
raw_data = {'edad': [18, 24, 30, 28],
        'altura': [1.7, 1.6, 1.9, 1.5],
        'sexo': ['masculino', 'femenino', 'femenino', 'masculino']}
data = pd.DataFrame(raw_data, columns = ['edad', 'altura', 'sexo'])

# 2 Creamos un dataframe de variables Dummies para columna "Sexo"
df_sexo = pd.get_dummies(data['sexo'])

# 3 Agregamos estas nuevas variables dummies a nuestro dataframe
df_new = pd.concat([df, df_sex], axis=1)
```

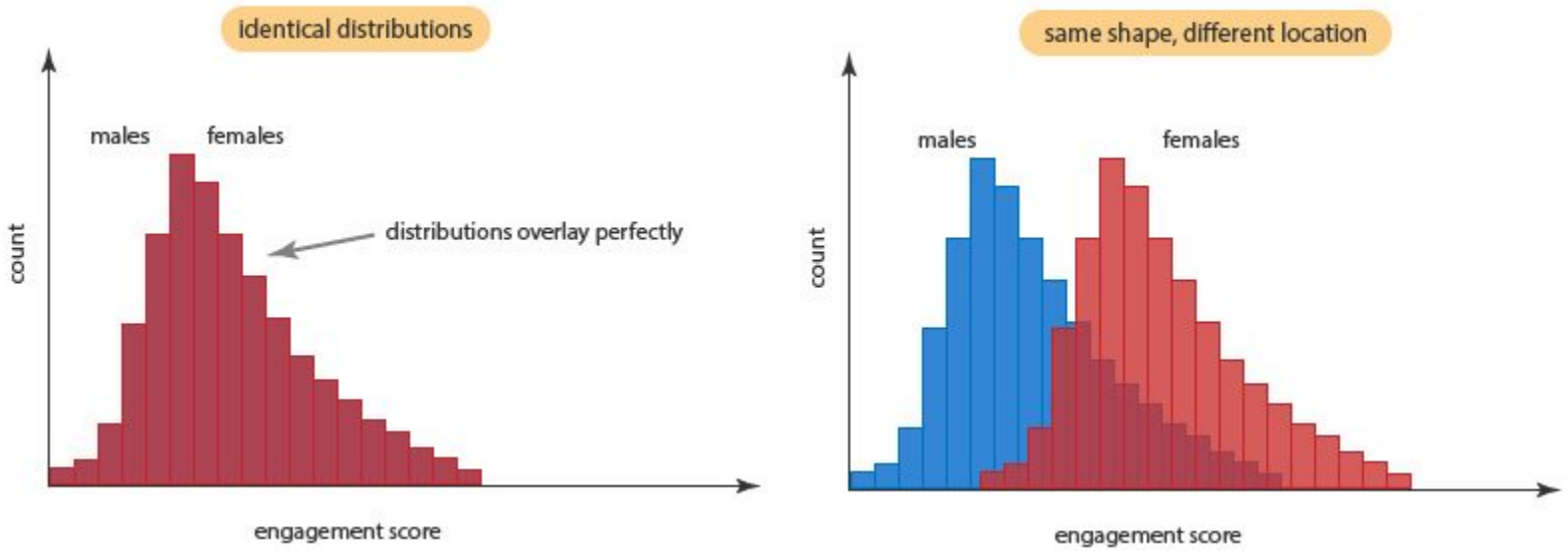
Feature Engineering: Auto-Scaling

$$x_i' = \frac{(x_i - \mu)}{\sigma}$$

Feature Engineering: Auto-Scaling



P-values, tests estadísticos: Mann Whitney U Test



P-values, tests estadísticos: Mann Whitney U Test

H0 = Las medias de las muestras son distintas
H1 = Las medias de las muestras pertenecen a la misma población

P value < 0.05 --> rechazamos H0