

Python

nella

PUBBLICA AMMINISTRAZIONE

e nella

FORMAZIONE



Giugno 2022

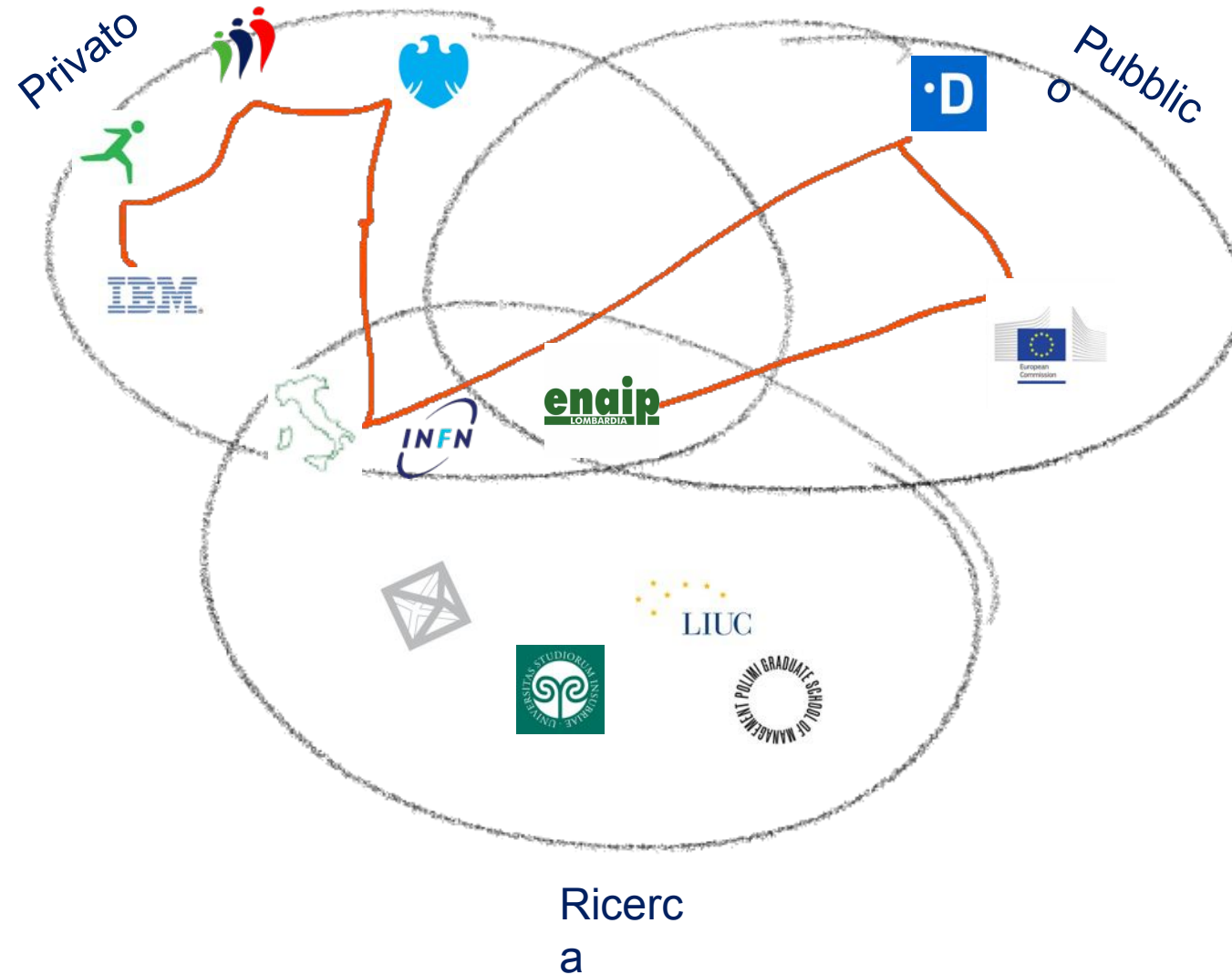
by

Andrea Biancini



Informatico e psicologo,
appassionato di
tecnologia mi occupo di
formazione, educazione
e **sviluppo** del capitale
umano

Un percorso di carriera poliedrico...



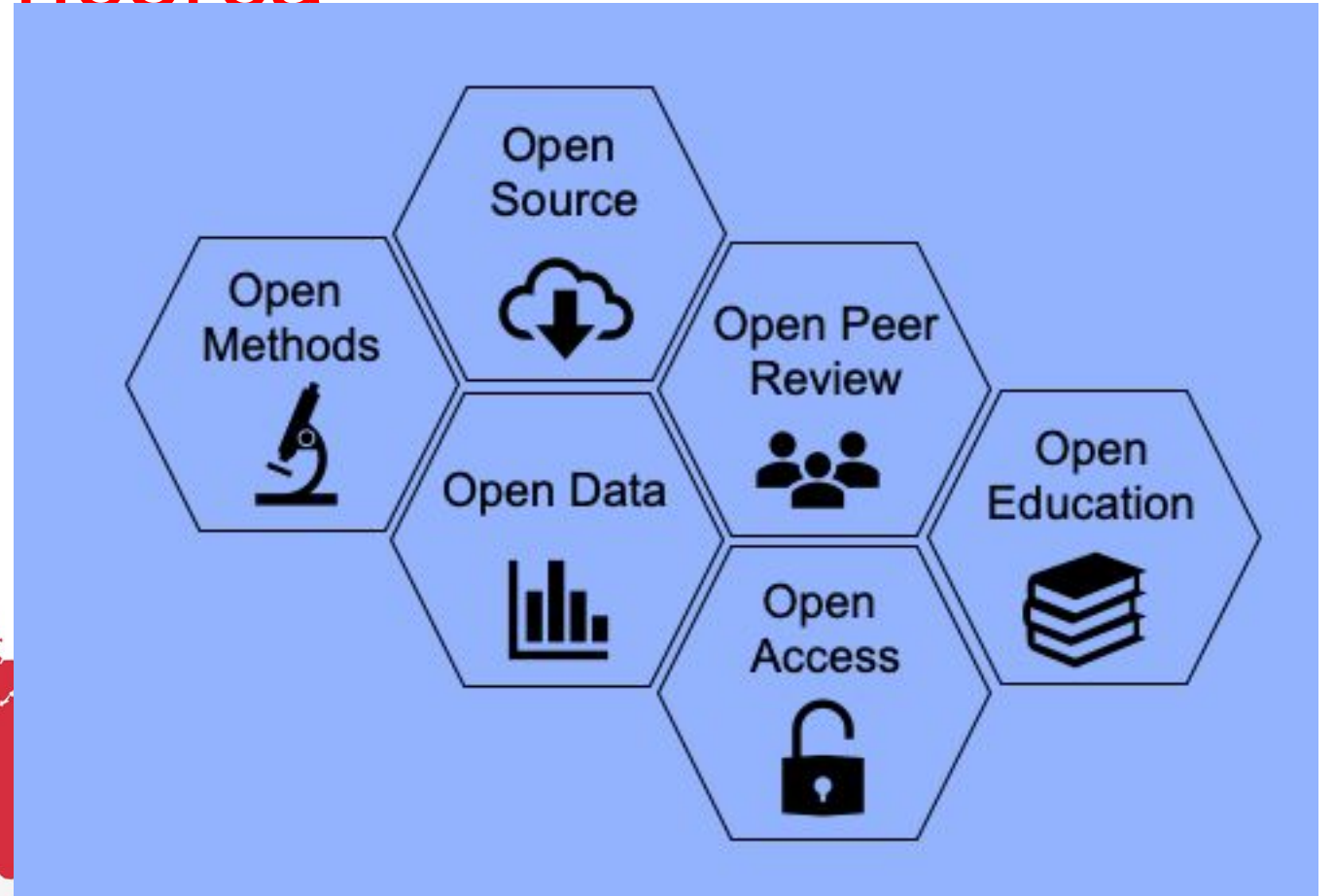
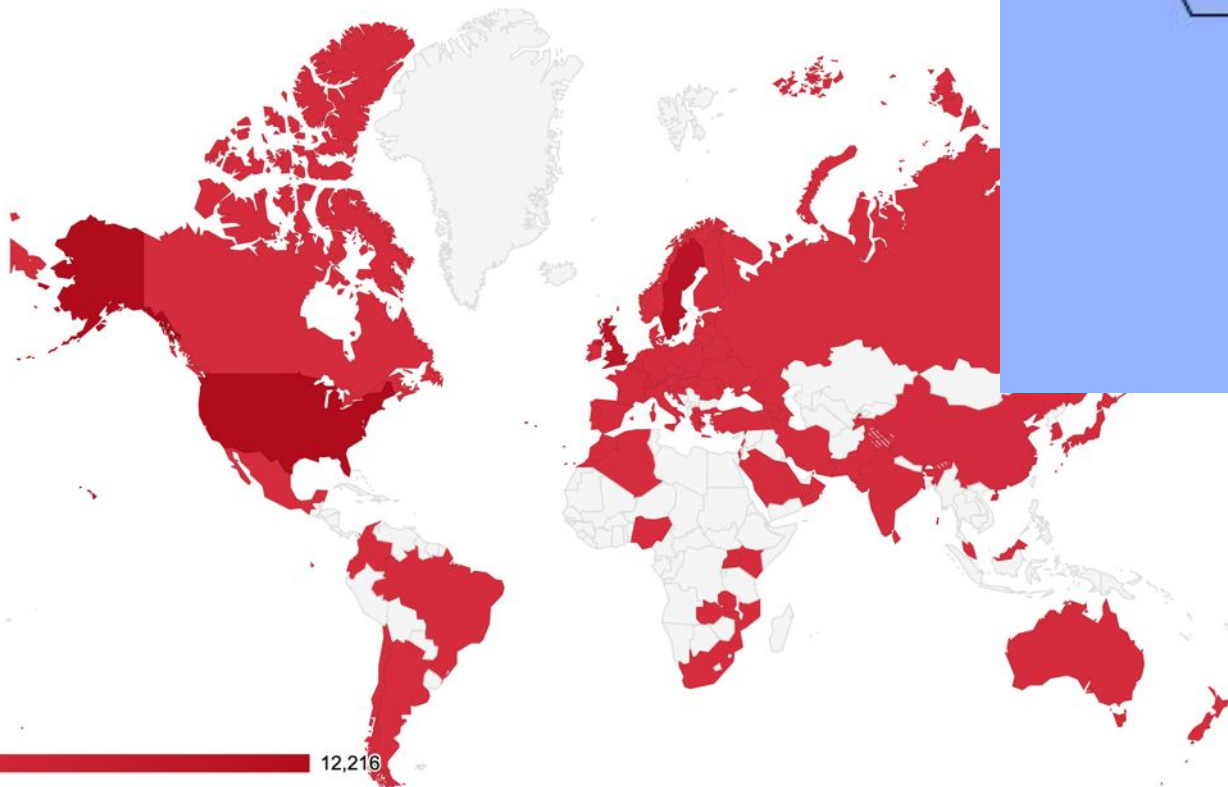
Why Python?

1

OSS

Python è un linguaggio di programmazione
Open Source

La collaborazione nella ricerca



Innovare la pubblica amministrazione



[Nessun titolo]



PA seems inexorable

1. You often feel **impotent** confronted to the mechanisms of the PA.
2. **Nothing** really seems to **change**, year after year.



That PA is out there!



"We are venom now, no more brock"




PA is entangled with politics

1. PA is **exposed to politics**.
2. Decisions seem to **value more non technical aspects**.
3. Dealing with PA seems to be serious stuff: *no more brock!*





TEAM PER LA TRASFORMAZIONE DIGITALE



The Digital Team: a startup in the government

30 senior experts focused on software architectures, big data, cloud, cybersecurity, UX/service design and... **open source!**



TEAM PER LA TRASFORMAZIONE DIGITALE

We are building the Country's "operating system":

- National Resident Population Register (ANPR)
- Digital Administration Code (CAD)
- Data & Analytics Framework (DAF)
- Designers Italia
- Developers Italia
- Docs Italia
- API Ecosystem
- Public Digital Identity System (SPID)
- Project IO - Digital Citizenship
- PagoPA - Digital Payments

The barriers to the Sharing and reuse of IT solutions

4 groups of barriers to sharing and/or reusing IT solutions have been identified:



Organisational
infrastructures.

e.g. Limited awareness of similar needs across sectors, fragmented IT



Legal

e.g. Uncertainty regarding limitation of, or exceptions to, IPR.



Technical

e.g. Difficulty adapting legacy systems.



Communication

e.g. Lack of awareness about existing solutions.

2

Community

Python ha una community viva e attiva,
che dialoga e interagisce



We didn't want to run alone!



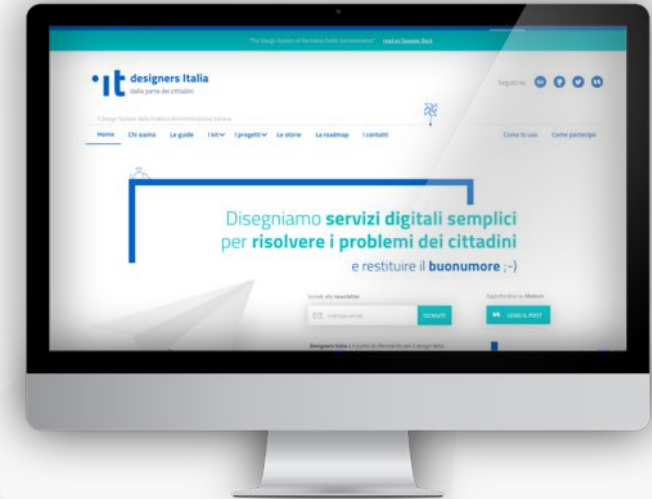
TEAM PER LA
TRASFORMAZIONE
DIGITALE

A new way to
develop, design,
communicate and
collaborate

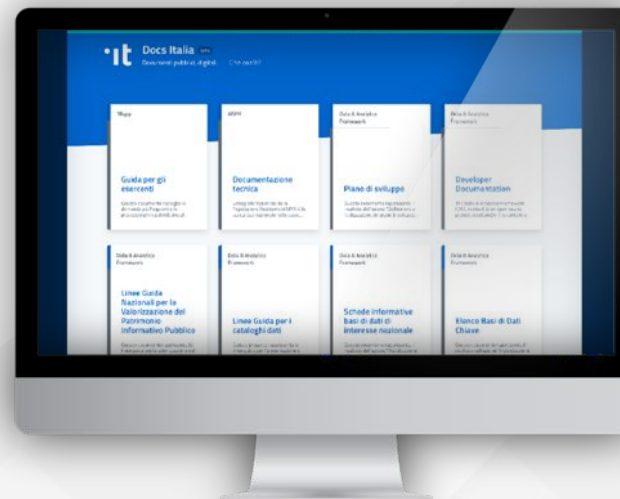
Developers Italia



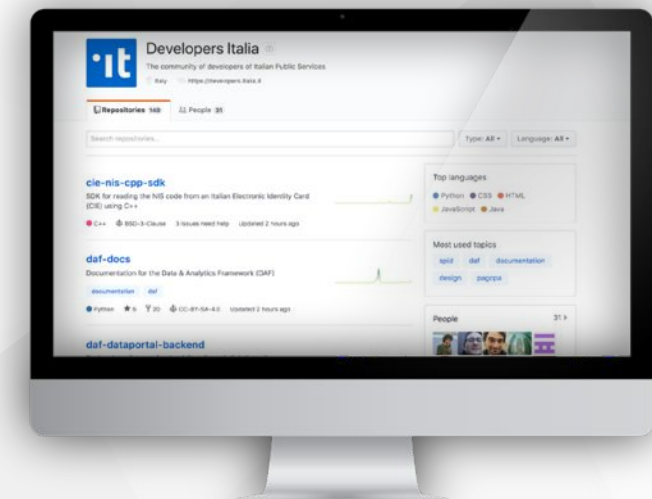
Designers Italia



Docs Italia



Forum Italia



Software

Il catalogo del software open source a disposizione della Pubblica Amministrazione.

Questo è il catalogo previsto dalle [Linee Guida per l'Acquisizione e il Riutilizzo del Software](#), che include le soluzioni messe a riutilizzo dalla Pubblica Amministrazione ai sensi dell'art. 69 e il software open source di terze parti destinato alla PA.

[Maggiori informazioni](#)[FAQ](#)

Software a riutilizzo

Soluzioni e programmi informatici messi a riutilizzo dalla Pubblica Amministrazione

[Esplora](#)

Open source di terze parti

Software open source di interesse per la Pubblica Amministrazione

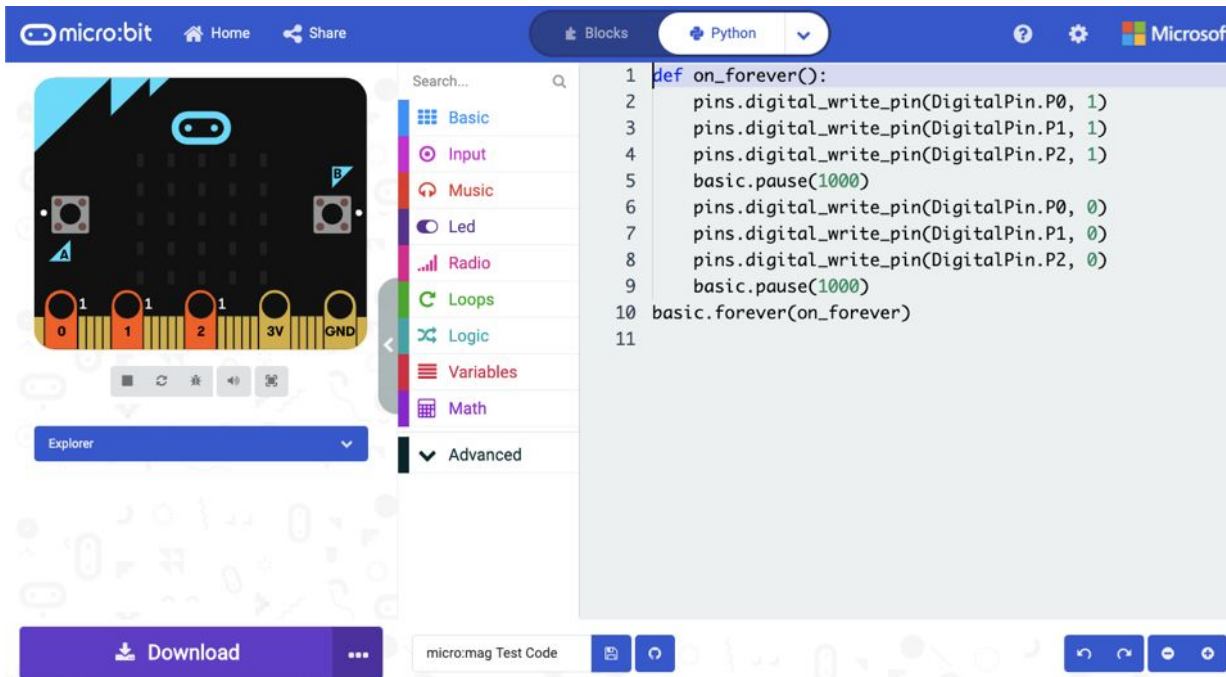
[Esplora](#)

3

Easy

Python è un linguaggio molto semplice da usare, anche per non *IT-guys*

Teaching and learning



Make Code permette una programmazione a blocchi che può essere tradotta in python!

Per bambini alla scuola primaria...

Con Forge è possibile scriptare da python Minecraft.

Questa è l'attività perfetta per ragazzi delle medie o dei primi anni delle superiori!



Fino all'università...

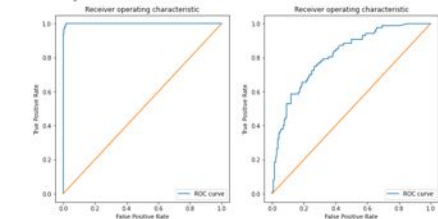
```
from sklearn.metrics import roc_curve, auc, roc_auc_score
import matplotlib.pyplot as plt

def draw_roc_curve(descr, Y, Y_pred_proba, ax1):
    fpr, tpr, _ = roc_curve(Y, Y_pred_proba)
    roc_auc = auc(fpr, tpr)
    print(f'The computed {descr} AUC is: {roc_auc:.3f}')

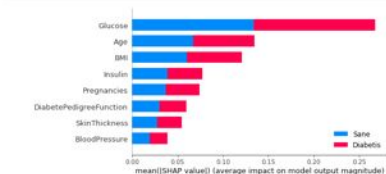
    ax1.plot(fpr, tpr, label="ROC curve")
    ax1.plot([0, 1], [0, 1])
    ax1.set_xlabel("False Positive Rate")
    ax1.set_ylabel("True Positive Rate")
    ax1.set_title("Receiver operating characteristic")
    ax1.legend(loc="lower right")

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))
draw_roc_curve('training', Y_train, Y_train_pred_proba[:,1], ax1)
draw_roc_curve('test', Y_test, Y_test_pred_proba[:,1], ax2)
plt.show()
```

The computed training AUC is: 1.000
The computed test AUC is: 0.816



```
[ ] import shap
shap_values = shap.TreeExplainer(model).shap_values(X_train)
shap.summary_plot(shap_values, X_train, plot_type="bar", class_names=['Sane', 'Diabetis'])
```



Confusion Matrix Train		Confusion Matrix Train	
True Label	Predicted Label	Sane	Diabetis
	Sane	353	3
True Label	Predicted Label	Sane	Diabetis
	Sane	123	21
True Label	Predicted Label	Sane	Diabetis
	Diabetis	6	175
True Label	Predicted Label	Sane	Diabetis
	Diabetis	36	51

```
[ ] def regression_polynomial(i):
    # Add all variables as power of x to the regression model
    variables = ['x']
    for j in range(2, i+1):
        variables.append(f'x{j}')
    data[f'x{j}'] = data['x'] ** j
```

```
X = data[variables]
Y = data['y']
```

```
regr = LinearRegression()
regr.fit(X, Y)
```

```
Y_pred = regr.predict(X)
```

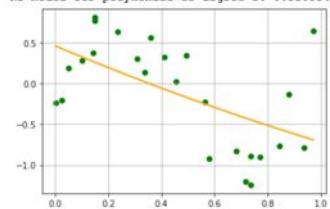
```
print("R2 model for polynomial of degree %d: %f" % (i, r2_score(Y, Y_pred)))
```

```
# Plot the result by building an expression for the fitting function
x = np.linspace(data['x'].min(), data['x'].max(), 100)
yhat = regr.intercept_ + regr.coef_[0] * x
for j in range(2, i+1):
    yhat += regr.coef_[j-1] * (x**j)
plt.plot(x, yhat, lw=2, c='orange')
```

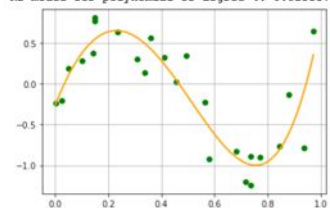
```
plt.scatter(data['x'], data['y'], color='green')
plt.grid(True)
plt.show()
```

```
regression_polynomial(2)
regression_polynomial(4)
regression_polynomial(8)
regression_polynomial(15)
```

R2 model for polynomial of degree 2: 0.320534



R2 model for polynomial of degree 4: 0.825337



R2 model for polynomial of degree 8: 0.887087

```
from sklearn.cluster import DBSCAN
from sklearn.metrics import v_measure_score
import seaborn as sns
```

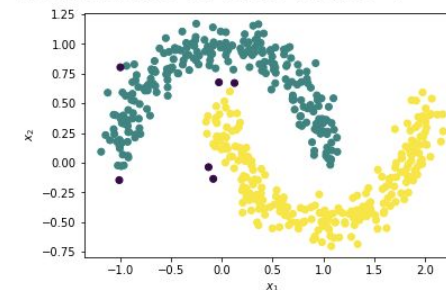
```
dbscan_cluster = DBSCAN(eps=eps, min_samples=10)
dbscan_cluster.fit(df)
labels=dbscan_cluster.labels_
```

```
# Visualizing DBSCAN
plt.scatter(df['X1'], df['X2'], c=dbscan_cluster.labels_, label=labels)
plt.xlabel("$x_1$")
plt.ylabel("$x_2$")
```

```
# Number of Clusters
N_clus=len(set(labels))-(1 if -1 in labels else 0)
print('Estimated no. of clusters: %d' % N_clus)
```

```
# Identify Noise
n_noise = list(dbscan_cluster.labels_).count(-1)
print('Estimated no. of noise points: %d' % n_noise)
```

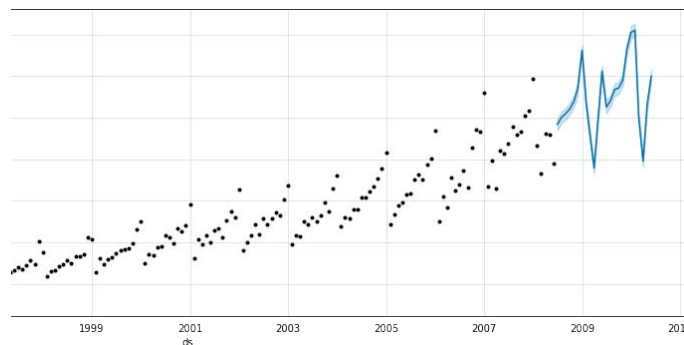
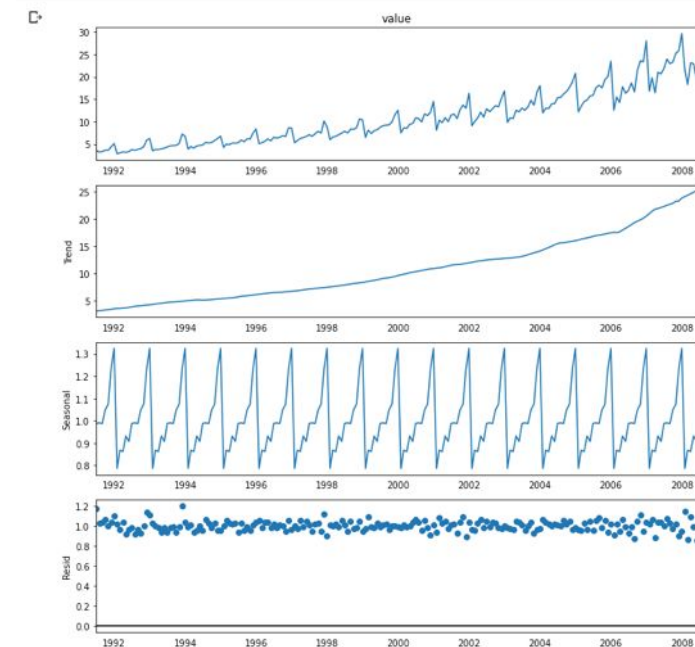
Estimated no. of clusters: 2
Estimated no. of noise points: 6



```
from statsmodels.tsa.seasonal import seasonal_decompose

result_mul = seasonal_decompose(df['value'], model='multiplicative', e

plt.rcParams.update({'figure.figsize': (10, 10)})
result_mul.plot()
plt.show()
```



Le mie conclusioni...

Gli elementi di successo di un tool in campo IT

1. Open-source e Community
2. L'IT è un mezzo e non un fine
3. L'eccellenza dell'ordinario: non facciamo solo cose *fancy*

thank you

Andrea Biancini

<andrea.biancini@gmail.com>

linkedin.com/in/bianciniandrea