Pipelines de Machine Learning y Big Data

Elaborado por Gary Briceño

El Instructor

- Ingeniero Electrónico de la UNI - Perú
- Más de 18 años de trayectoria profesional en desarrollo de software
- Actualmente Senior Python Backend in Parser

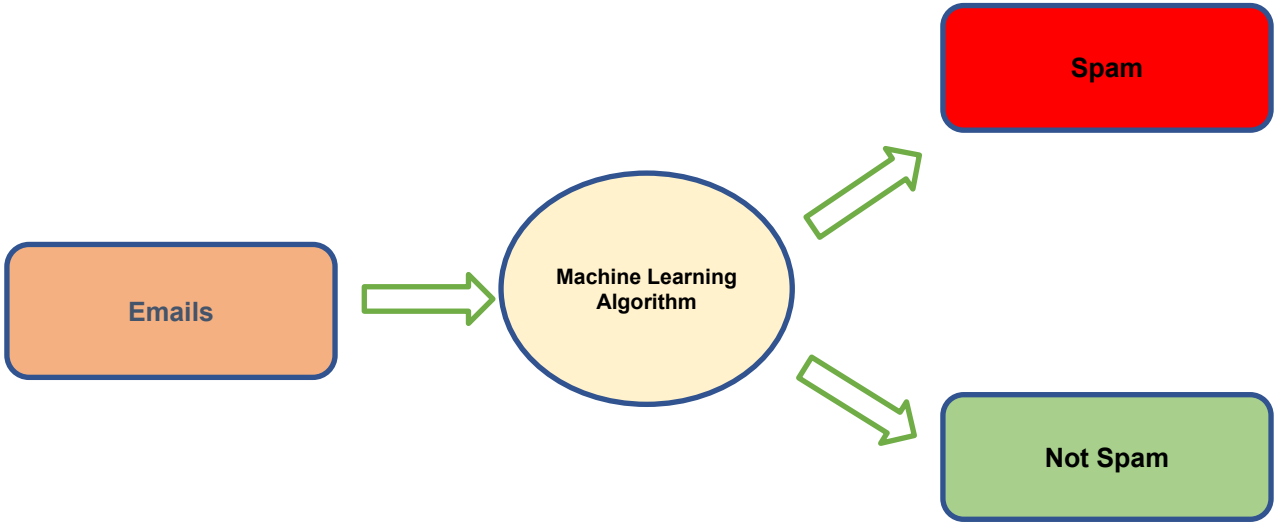
Enlaces:

- https://www.linkedin.com/in/garybriceno/
- https://github.com/GaryBriceno
- @garybriceno
- https://www.clubdetecnologia.net/blog/

Agenda

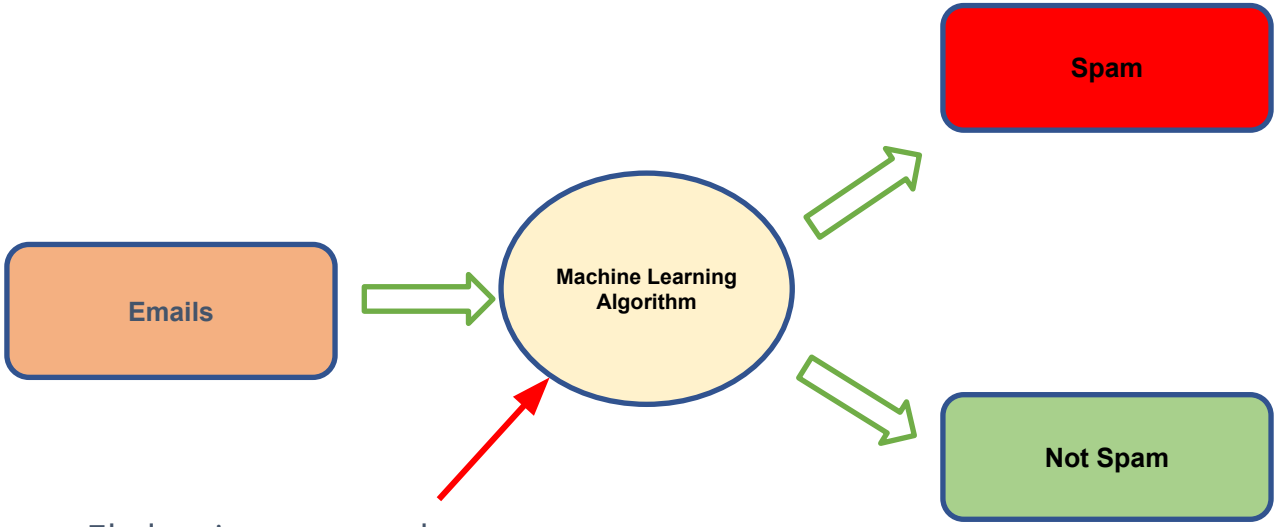
- ¿Que es Machine Learning?
- Tipos de sistemas de Machine Learning
- ¿Por que Big Data?
- ¿Por que Python?
- Pipelines de trabajo de Machine Learning

¿Que es Machine Learning?



- Tener la posibilidad de que un algoritmo genérico puede brindarte información interesante sobre un conjunto de datos sin escribir código específico para dicho problema.

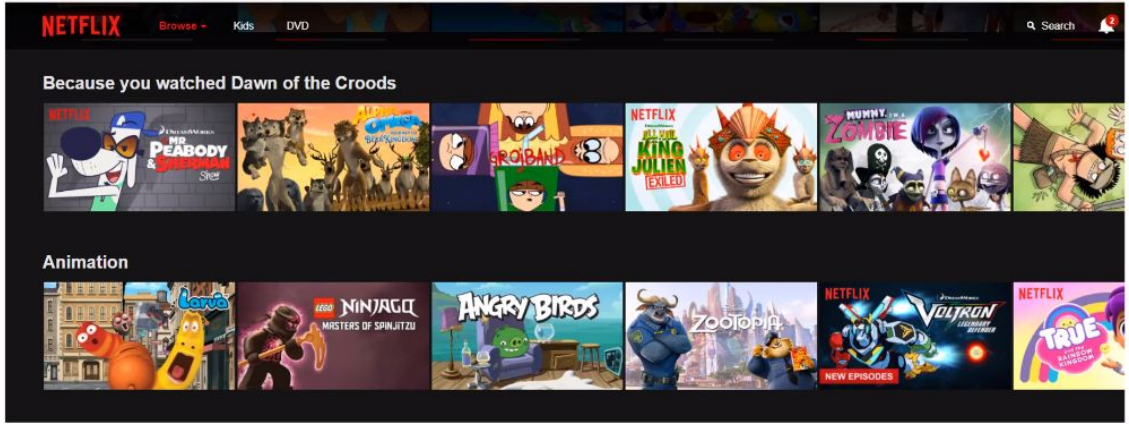
¿Que es Machine Learning?



- El algoritmo aprende que es spam vs no spam realizando comparaciones entre ambos y buscando patrones.

¿Que es Machine Learning?

Esta en todos lados

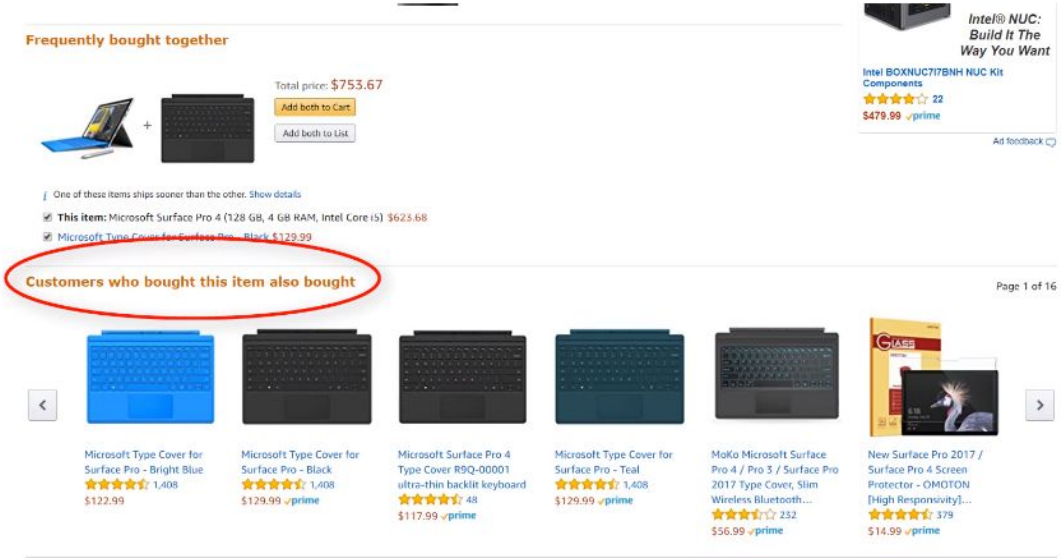


- Sistema de recomendación de películas.

- El 75% de las películas vistas en Netflix han sido recomendadas por uno de los motores de recomendación.

¿Que es Machine Learning?

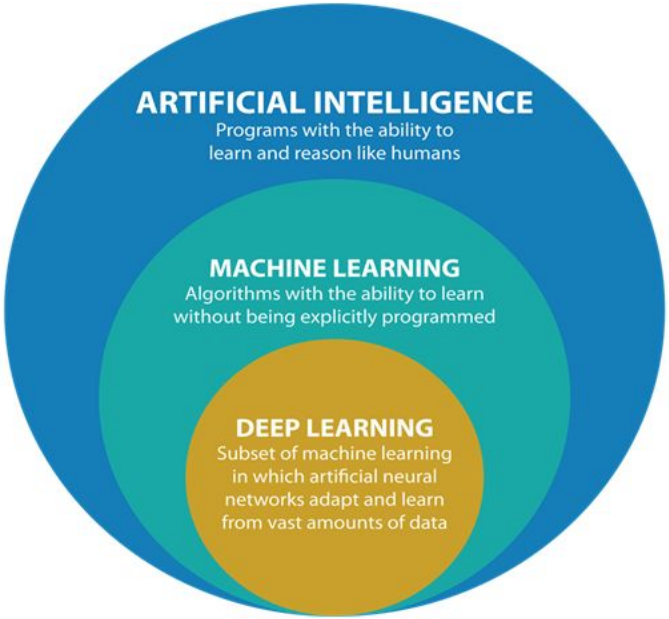
Esta en todos lados



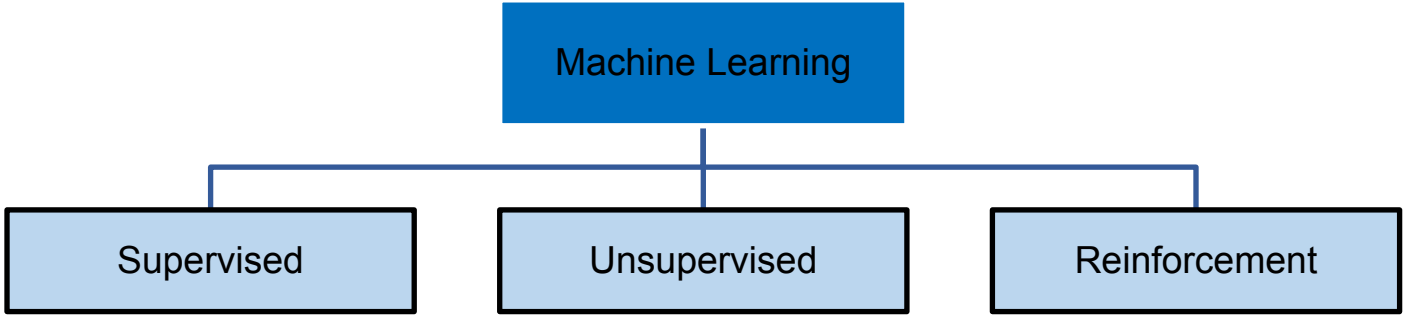
- Decisiones de compra

¿Que es Machine Learning?

Machine Learning es un tipo de Inteligencia Artificial (IA) que permite a las aplicaciones de software sean más precisas en la predicción de resultados **sin ser programadas explícitamente**.



Tipos de Sistemas de Machine Learning



Task Driven

(Regression/Classification)

- Email Spam Detection
- Image Classification
- Diagnostic

Data Driven

(Clustering)

- Face Recognition
- City Planning
- Targeted Marketing

Algorithm learns to react to an environment

- Gaming
- Manufacturing
- Robot Navigation

¿Por que usar Machine Learning?

Machine Learning es util para:

El algoritmo de aprendizaje automático a menudo puede simplificar el código y funcionar mejor que el enfoque tradicional.

- Problemas para los que las soluciones existentes requieren muchos ajustes o una larga lista de reglas.
- Problemas complejos para los que el uso de un enfoque tradicional no ofrece buenas soluciones.
- Entornos fluctuantes: el sistema de aprendizaje automático puede adaptarse a nuevos datos
- Obtener información sobre problemas complejos y grandes cantidades de datos

¿Porque BigData?

Forbes

It is no longer a secret that big data is a reason behind the successes of many major technology companies. However, as more and more companies embrace it to store, process and extract value from their huge volume of data, it is becoming a challenge for them to use the collected data in the most efficient way.

That's where machine learning can help them. Data is a boon for machine learning systems. The more data a system receives, the more it learns to function better for businesses. Hence, using machine learning for big data analytics happens to be a logical step for companies to maximize the potential of big data adoption.

https://www.forbes.com/sites/forbestechcouncil/2020/10/20/how-is-big-data-analytics-using-machine-learning/

¿Porque BigData?

Qubole

Last week, during the Deep Learning Summit at AWS re:Invent 2017, Terrence Sejnowski (a pioneer of deep learning) succinctly said "Whoever has more data wins". He was echoing a premise that has been repeated many times in many ways by many people: machine learning requires big data to work. Without large, well maintained training sets, machine learning algorithms—especially deep learning algorithms—fall far short of their potential. That's why here at Qubole we believe that enabling data scientists starts with giving them a platform to quickly select, clean, and aggregate datasets on a massive scale.

https://www.qubole.com/blog/machine-learning-requires-big-data/

¿Porque BigData?

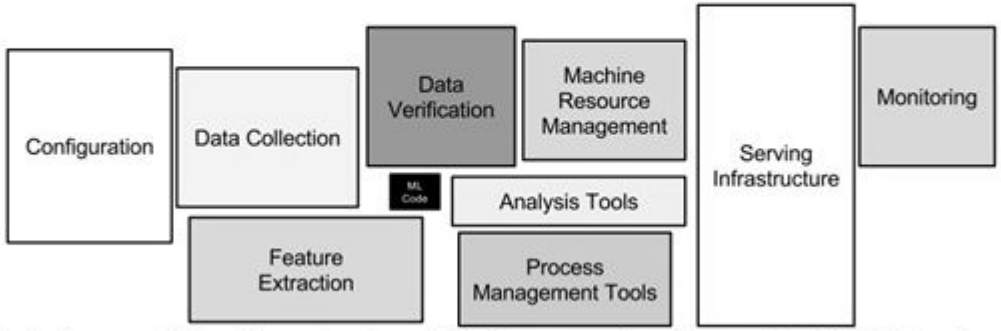
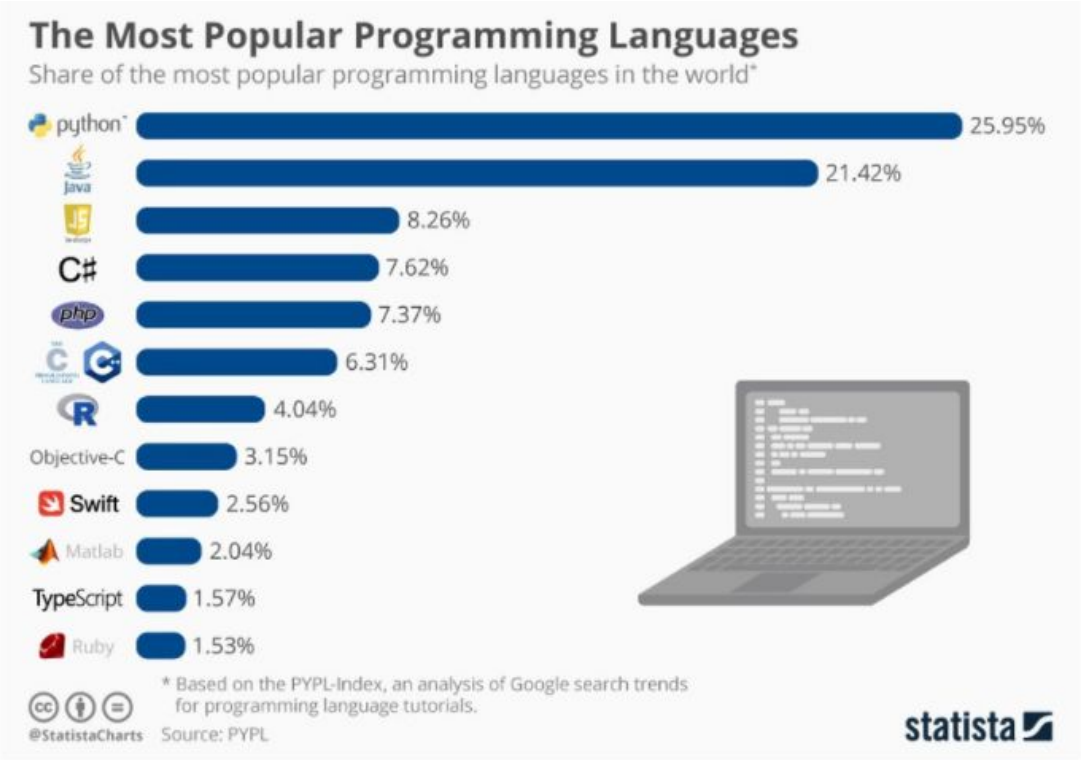


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

https://www.qubole.com/blog/machine-learning-requires-big-data/

¿Porque Python?



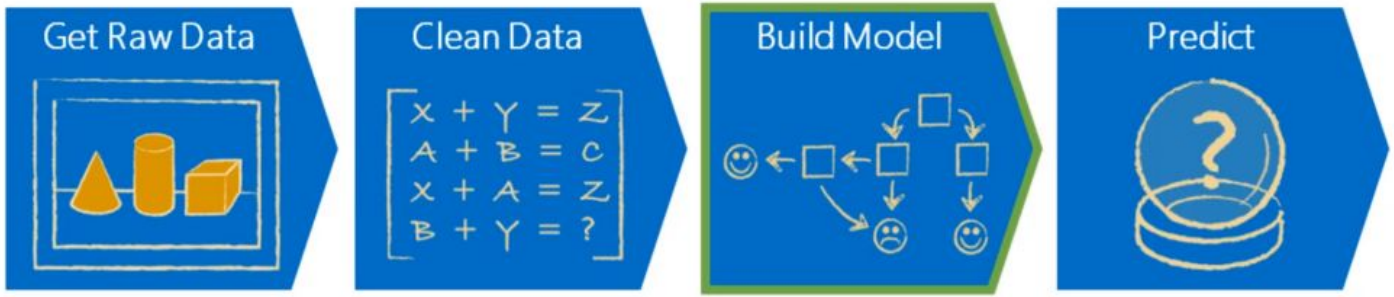
¿Porque Python?

Python tiene muchas librerías para analítica:

- Computación numérica: Scipy, numpy
- Análisis de datos: Pandas
- Análisis estadístico: scipy.stats, pyMC
- Visualización: Matplotlib. Seaborn
- Machine Learning: PyTorch, TensorFlow, NLTK, Sciki-Learn, Keras

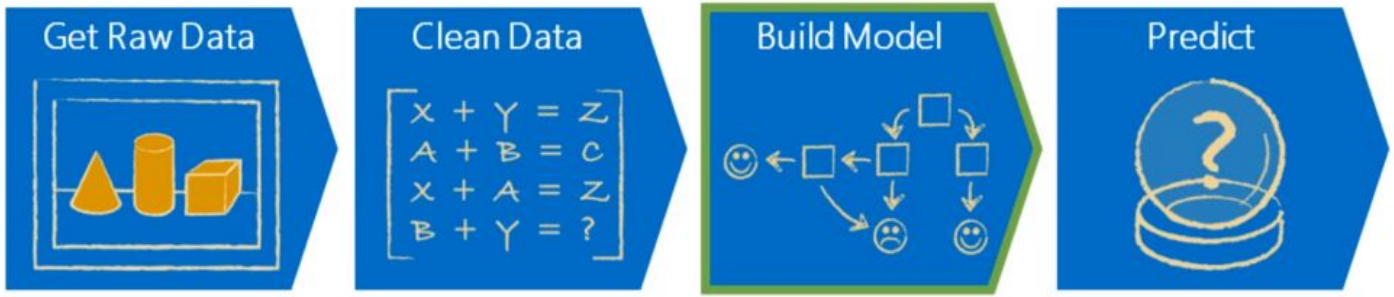
- https://medium.com/@scarlett8285/why-is-python-programming-a-perfect-fit-for-big-data-5ac54ee8f95e
- https://towardsdatascience.com/top-10-python-libraries-for-data-science-cd82294ec266
- https://relopezbriega.github.io/blog/2015/06/27/probabilidad-y-estadistica-con-python/

El proceso de Modelamiento



80% 10%

El Proceso de Modelamiento



80%

10%

- Manipulación de los datos
- Gráfica y análisis de los datos
- Selección de modelos
- Validación y Optimización

Manipulación de los Datos

In [14]:
```
cancer['data'].shape
```

Out[14]: (569, 30)

In [15]:
```
df_cancer = pd.DataFrame(np.c_[cancer['data'], cancer['target']], columns = np.append(cancer['feature_names'], ['target'
```

In [16]:
```
df_cancer.head()
```

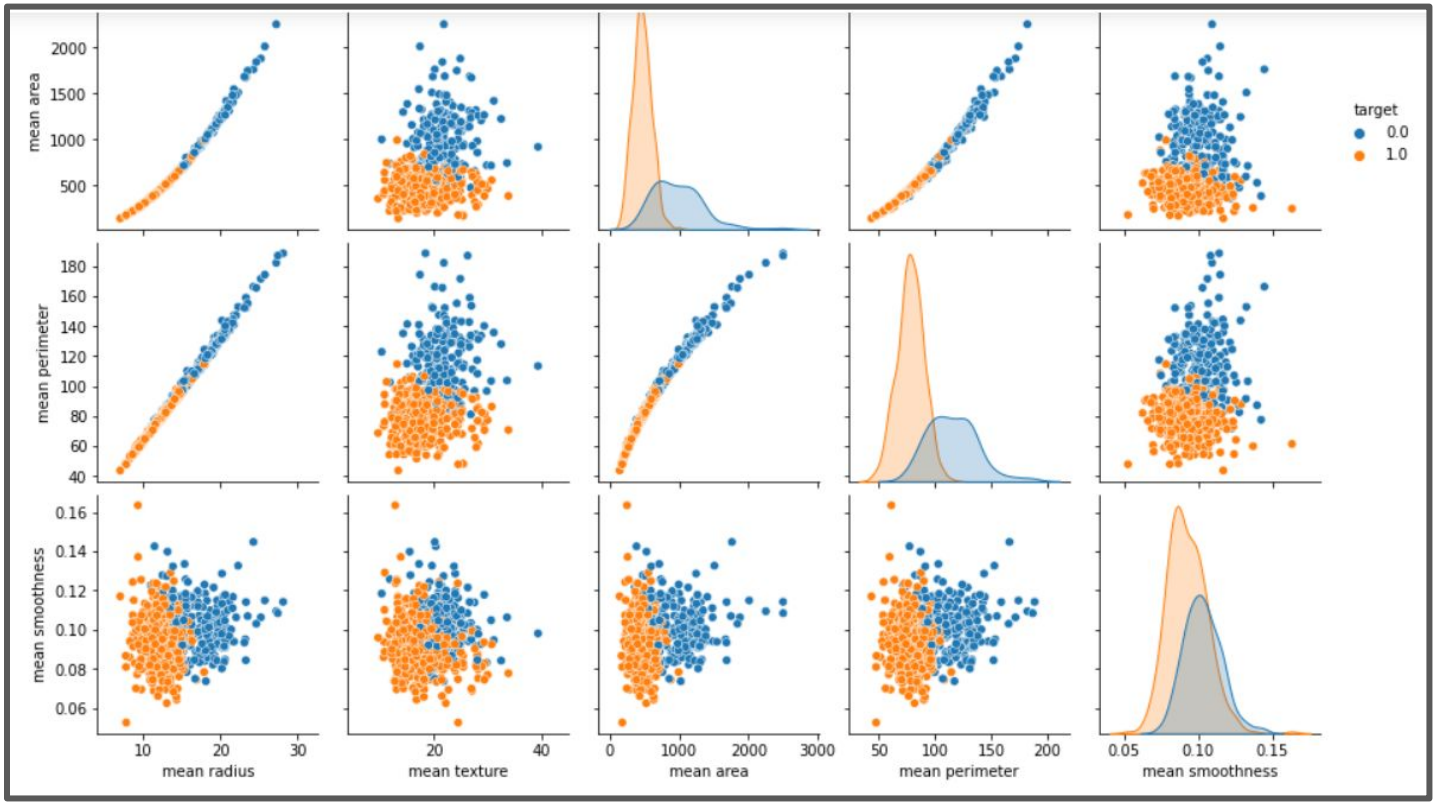
Out[16]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	17.33	184.60	2019.0	0.1622
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	23.41	158.80	1956.0	0.1238
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	25.53	152.50	1709.0	0.1444
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	26.50	98.87	567.7	0.2098
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	16.67	152.20	1575.0	0.1374

5 rows × 31 columns

- https://github.com/GaryBriceno/ml_examples

Gráfico y búsqueda de Correlaciones



Manipulación de los Datos

In [27]:
```
from sklearn.model_selection import train_test_split
```

In [28]:
```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=5)
```

In [29]:
```
x_train
```

Out[29]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area	sm
306	13.200	15.82	84.07	537.3	0.08511	0.05251	0.001461	0.003261	0.1632	0.05894	...	14.41	20.45	92.00	636.9	
410	11.360	17.57	72.49	399.8	0.08858	0.05313	0.027830	0.021000	0.1601	0.05913	...	13.05	36.32	85.07	521.3	
197	18.080	21.84	117.40	1024.0	0.07371	0.08642	0.110300	0.057780	0.1770	0.05340	...	19.76	24.70	129.10	1228.0	
376	10.570	20.22	70.15	338.3	0.09073	0.16600	0.228000	0.059410	0.2188	0.08450	...	10.85	22.82	76.51	351.9	
244	19.400	23.50	129.10	1155.0	0.10270	0.15580	0.204900	0.088860	0.1978	0.06000	...	21.65	30.53	144.90	1417.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
8	13.000	21.82	87.50	519.8	0.12730	0.19320	0.185900	0.093530	0.2350	0.07389	...	15.49	30.73	106.20	739.3	
73	13.800	15.79	90.43	584.1	0.10070	0.12800	0.077890	0.050690	0.1662	0.06566	...	16.57	20.86	110.30	812.4	
400	17.910	21.02	124.40	994.0	0.12300	0.25760	0.318900	0.119800	0.2113	0.07115	...	20.80	27.78	149.60	1304.0	
118	15.780	22.91	105.70	782.6	0.11550	0.17520	0.213300	0.094790	0.2096	0.07331	...	20.19	30.50	130.30	1272.0	
206	9.876	17.27	62.92	295.4	0.10890	0.07232	0.017560	0.019520	0.1934	0.06285	...	10.42	23.22	67.08	331.6	

455 rows × 30 columns

Manipulación de la data

Training Data

Feature 1	Feature 2	Feature 3	···	Feature N-2	Feature N	Label Data
						NOW
						NOW
						NOW

Data to Predict

Feature 1	Feature 2	Feature 3	···	Feature N-2	Feature N	Label Data
						UNKNOWN
						UNKNOWN
						UNKNOWN

Selección del Modelo

```
In [31]:  from sklearn.svm import SVC

In [32]:  from sklearn.metrics import classification_report, confusion_matrix

In [33]:  svc_model = SVC()

In [34]:  svc_model.fit(x_train, y_train)

Out[34]: SVC()
```

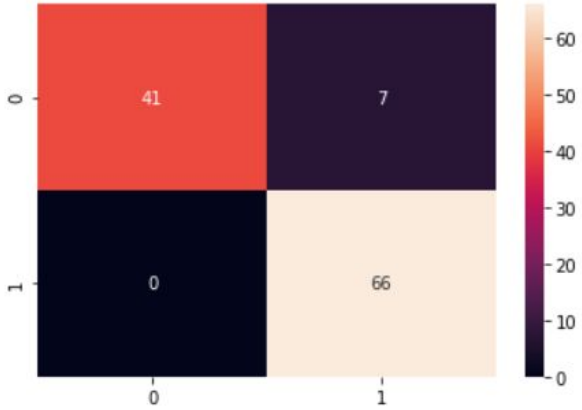
SVC : Support Vector Classification

Validación

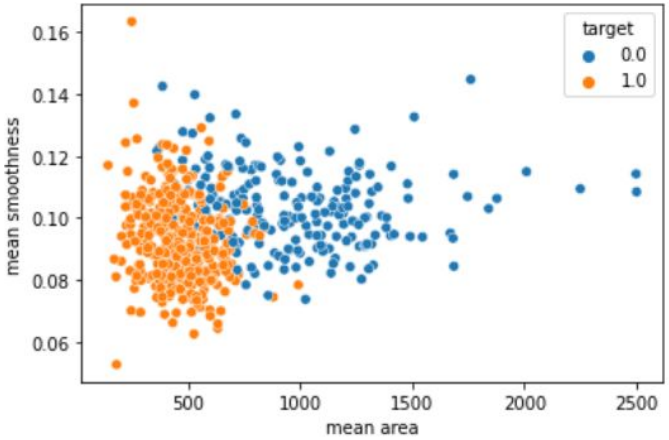
In [37]:
```
cm = confusion_matrix(y_test, y_predict)
```

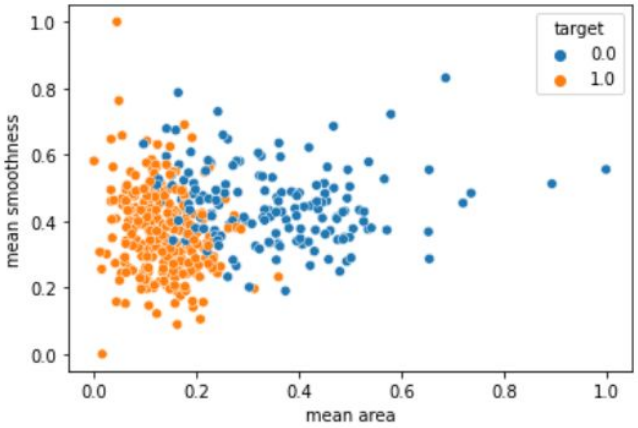
In [40]:
```
sns.heatmap(cm, annot=True)
```

Out[40]: <AxesSubplot:>



Optimización





- Normalization
- Change the parameters of the Model
- Changing the model

gracias!