Taller:

Introducción a las ciencias de datos con Python

24.05.04

Agenda (2024-05-04)

5 min.	Presentación
35 min.	Introducción
125 min.	Taller
10 min.	Preguntas
5 min.	Aprendizajes

5 min. Presentación

Luis Antonio (FunkyM0nk3y) Galindo Castro

- Ingeniero en computación, UNAM
- Diplomado en Big Data, Ciencias de Datos por el ITESM y Técnicas de buceo Aplicadas a la investigación Subacuática por la UNAM
- Más de 25 años en la industria de tecnología
- Cerca de 20 años participando como ponente en congresos nacionales e internacionales
- Miembro de la Open Source Initiative durante 10 años
- Miembro de la Linux Foundation durante 8 años

  funkymonster@linux.com

- Buzo y geek ->

35 min. Introducción

Calentando motores

Dos opciones en la vida:

- Pastilla roja: DIY, habrá que instalar jupyter y todo lo necesario para el workshop
- Pastilla azul: Utilizar Google Collab y después tomarse la pastilla roja

Ciclo de vida de los retos con datos

1. Formulate a problem
Frame the core ML problem(s) in terms of what is observed and what answer you want the model to predict.

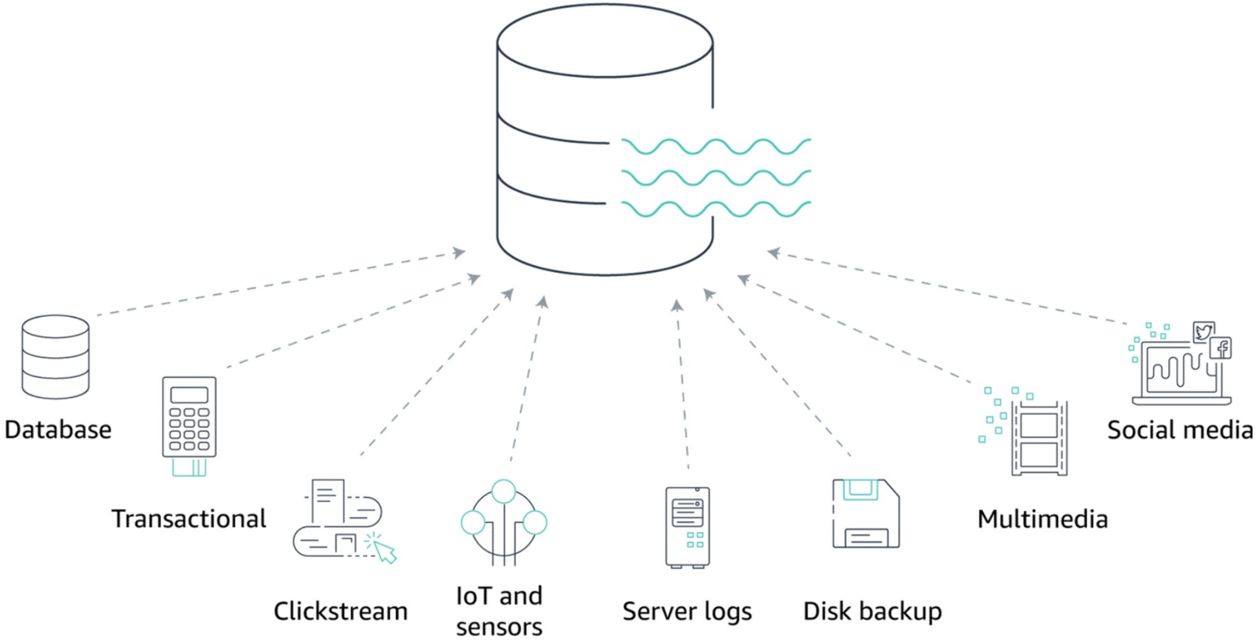
2. Prepare your data
Collect, clean, and prepare data to make it suitable for consumption by ML model training algorithms. Visualize and analyze the data to run sanity checks to validate the quality of the data and to understand the data.

3. Train the model
To train a highly predictive model, the raw data (input variables) and answer (target) can't always be used effectivly. Preferably, construct more predictive input representations or features from the raw variables. Feed the resulting features to the learning algorithm to build models. Set aside, or hold out, a subset of the input data to test the model.

4. Test the model
Evaluate the quality of the models on data that was held out from model building.

5. Deploy your model
Use the model to generate predictions of the target answer for new data instances.

Fuentes de datos

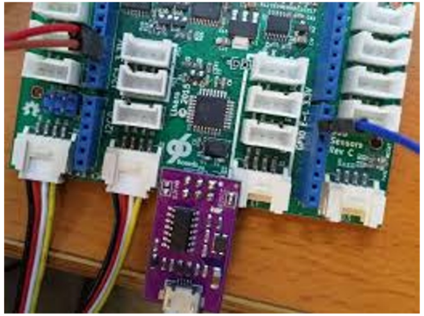


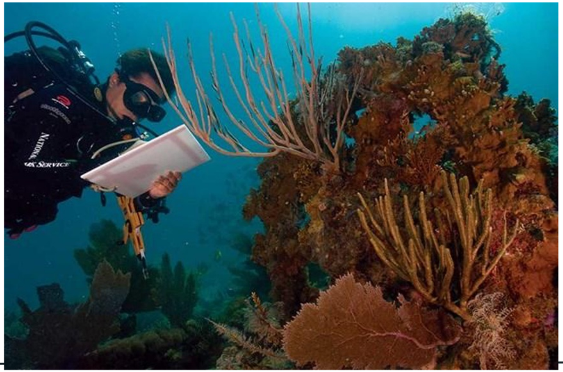
¿De dónde pueden venir los datos?

En la actualidad las fuentes son muy diversas, así como sus formatos

<- Datos en formatos analógicos y formatos no estructurados

Datos provenientes de sensores ->






```
21:20:03.000 -> {alturaConst:2271,temperaturaReferencia:25.04,humedadAmbiente:51.77,temperaturaAmbiente:24.48,co2ppmAmbiente:812,humedadSustrato:318.00,ventilador:0}
21:20:03.000 ->
21:21:03.002 -> {alturaConst:2271,temperaturaReferencia:25.04,humedadAmbiente:51.82,temperaturaAmbiente:24.45,co2ppmAmbiente:815,humedadSustrato:319.00,ventilador:0}
21:21:03.002 ->
21:21:56.455 -> {alturaConst:2271,temperaturaReferencia:24.95,humedadAmbiente:51.87,temperaturaAmbiente:24.45,co2ppmAmbiente:820,humedadSustrato:313.00,ventilador:0}
21:21:56.455 ->
21:22:54.666 -> {alturaConst:2271,temperaturaReferencia:24.95,humedadAmbiente:51.95,temperaturaAmbiente:24.47,co2ppmAmbiente:827,humedadSustrato:310.00,ventilador:0}
21:22:54.666 ->
21:23:58.410 -> {alturaConst:2271,temperaturaReferencia:24.95,humedadAmbiente:52.03,temperaturaAmbiente:24.45,co2ppmAmbiente:834,humedadSustrato:310.00,ventilador:0}
21:23:58.410 ->
21:24:53.006 -> {alturaConst:2271,temperaturaReferencia:24.87,humedadAmbiente:52.11,temperaturaAmbiente:24.50,co2ppmAmbiente:835,humedadSustrato:307.00,ventilador:0}
21:24:53.006 ->
21:25:56.448 -> {alturaConst:2271,temperaturaReferencia:25.04,humedadAmbiente:52.20,temperaturaAmbiente:24.47,co2ppmAmbiente:838,humedadSustrato:320.00,ventilador:0}
21:25:56.448 ->
21:26:52.982 -> {alturaConst:2271,temperaturaReferencia:25.04,humedadAmbiente:52.30,temperaturaAmbiente:24.48,co2ppmAmbiente:839,humedadSustrato:311.00,ventilador:0}
21:26:52.982 ->
21:27:52.990 -> {alturaConst:2271,temperaturaReferencia:24.87,humedadAmbiente:52.34,temperaturaAmbiente:24.50,co2ppmAmbiente:839,humedadSustrato:318.00,ventilador:0}
21:27:52.990 ->
21:28:48.535 -> {alturaConst:2271,temperaturaReferencia:24.95,humedadAmbiente:52.35,temperaturaAmbiente:24.47,co2ppmAmbiente:842,humedadSustrato:312.00,ventilador:0}
21:28:48.569 ->
21:29:52.976 -> {alturaConst:2271,temperaturaReferencia:25.04,humedadAmbiente:52.43,temperaturaAmbiente:24.48,co2ppmAmbiente:843,humedadSustrato:320.00,ventilador:0}
21:29:52.976 ->
21:30:52.979 -> {alturaConst:2271,temperaturaReferencia:25.04,humedadAmbiente:52.46,temperaturaAmbiente:24.48,co2ppmAmbiente:843,humedadSustrato:316.00,ventilador:0}
21:30:52.979 ->
21:31:52.976 -> {alturaConst:2271,temperaturaReferencia:25.20,humedadAmbiente:52.49,temperaturaAmbiente:24.50,co2ppmAmbiente:844,humedadSustrato:319.00,ventilador:0}
21:31:52.976 ->
```

Ciclo de vida de modelos

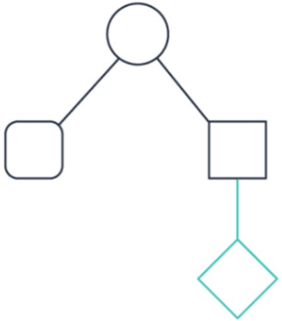
Model
The output of an ML algorithm trained on a data set; used for data prediction

Training
The act of creating a model from past data

Testing
Measuring the performance of a model on test data

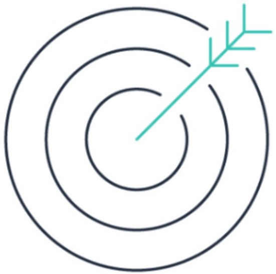
Deployment
Integrating a model into a production pipeline

¿Cuándo NO usar AI (ML)?



Can be solved with traditional algorithms
If the problem is not overly complex, an ML solution might be overcomplicated.

Does not require adapting to new data
If data and conditions are not changing, a more traditional approach could be more appropriate.



Requires 100% accuracy
ML predictions often provide less than 100% accuracy.

Requires full interpretability
If being able to explain what is going to happen if you change the parameters or input is a priority, ML might not be the best solution.

¿Cuándo SÍ usar AI (ML)?

Requires complex logic

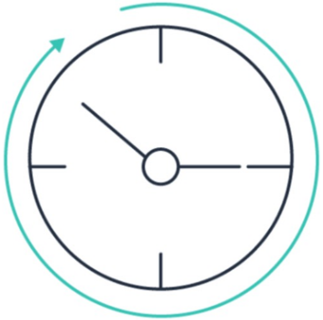
Since developing personalized recommendations requires complex logic, ML is an appropriate tool to consider.

Requires scalability

Serving millions of requests for personalized recommendations every second is a challenge.

Requires personalization

Delivering personalized recommendations at scale and being responsive at the same time is difficult to achieve with classical programming techniques.



Requires responsiveness

The ability to deliver personalized recommendations within a few seconds even while handling millions of requests per second is expected.

125 min. Taller

Dinámica de trabajo
Todo mundo puede compartir su conocimiento

Los ejercicios los estaremos haciendo sobre:

- Línea de comandos
- Jupyter
- Google Colab

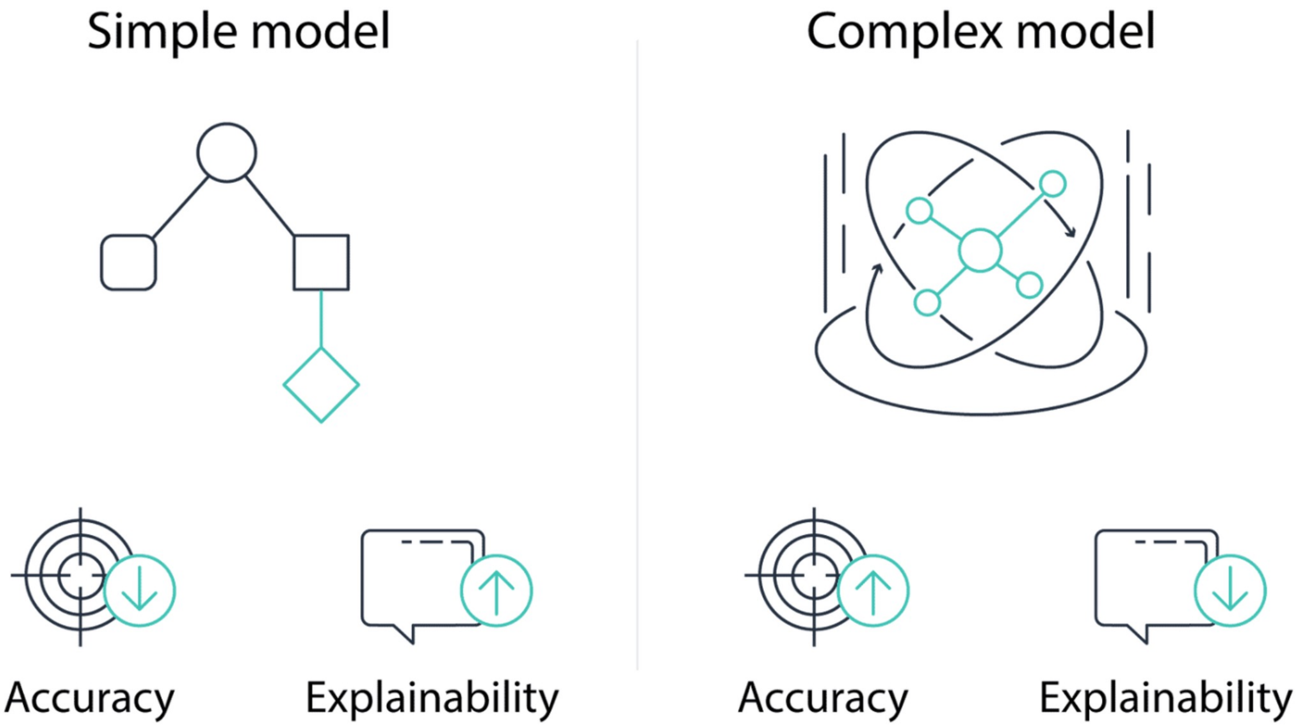
https://github.com/FunkyM0nk3y/FSL_Vallarta-2024

Preguntas
10 min.

Aprendizajes

10 min.

Indicadores para los modelos



¿Cómo llevar una estrategia de AI exitosa?

Robust AI strategy

- Start with important projects as proofs of concept
- Gain momentum using these projects
- Explain the why behind the shift towards AI

Data strategy

- Explore and document where the data resides
- Document if the data can be used for an AI project
- Prepare a timeline for building a data pipeline if there isn't one

Culture of learning and collaboration

- Allow teams to explore and experiment with the data
- Team up with external resources
- Leverage managed services

Referencias

Libros

Building Data Science Teams
by DJ Patil
Copyright © 2011 O'Reilly Media

Machine Learning Is Changing the Rules
by Peter Morgan
Copyright © 2018 O'Reilly Media

Big Data Glossary
by Pete Warden
Copyright © 2011 Pete Warden

The Evolution of Data Products
by Mike Loukides
Copyright © 2011 O'Reilly Media

Data Driven
by DJ Patil, Hilary Mason
Copyright © 2015 O'Reilly Media

The Art of Data Science
By Roger D. Peng and Elizabeth Matsui

Otros recursos

Competencias en Kaggle
https://www.kaggle.com/competitions

¿A quién seguir en twitter?
@drewconway (Drew Conway)
@rdpeng (Roger D Peng)
@AndrewYNg (Andrew Ng)
@ylecun (Yan LeCun)
@martinkl (Martin Kleppmann)
@mikeloukides (Mike Loukides)
@dpatil (DJ Patil)

¡Muchas gracias!

funkymonster@linux.com