(Introduction to) Data Quality

Mariusz Slapek

Agenda

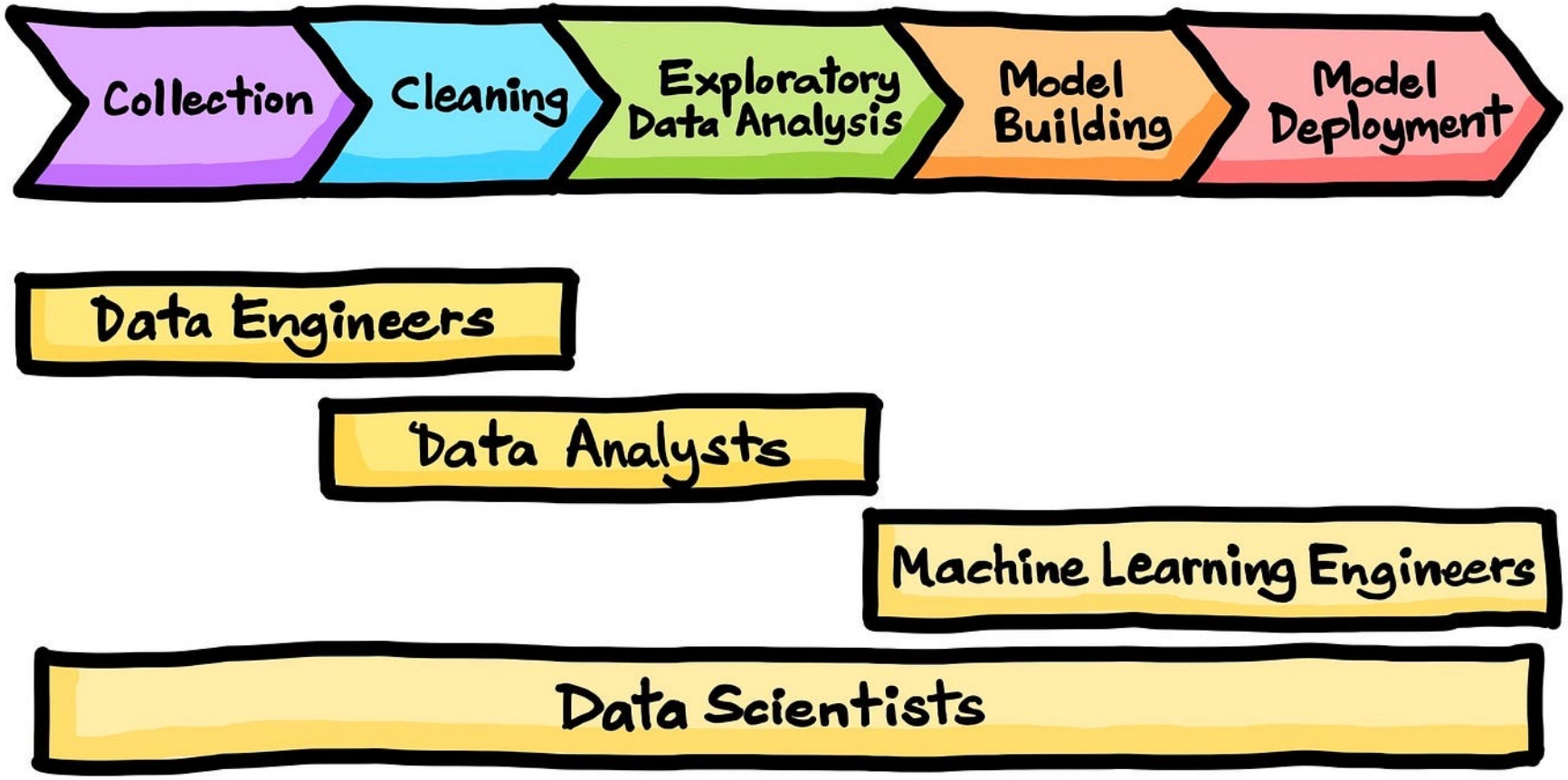
- Defining the problem
  - Data Science workflow
  - What can go wrong?
  - What do we mean by Data Quality?
- How to deal with it?
  - Can Python help us?
  - Comparison of available packages
  - So what to choose?
- So let's solve our problem
  - What’s next?
- Questions & Answers

Defining the problem

Customer type	Age	Type of travel	Class	Distance	Seat comfort	Food and drink	Baggage handling	Departure delay	Arrival delay	Satisfaction
Loyal Customer	65	Personal Travel	Eco	265	0	0	3	0	4	Satisfied
Disloyal Customer	47	Personal Travel	Business	2464	4	3	4	310	305	Dissatisfied
Loyal Customer	15	Personal Travel	Eco	2138	2	1	4	0	0	Satisfied
Loyal Customer	60	Business Travel	Eco	623	5	4	1	5	10	Satisfied

Airline Passenger Satifaction

Data Science workflow

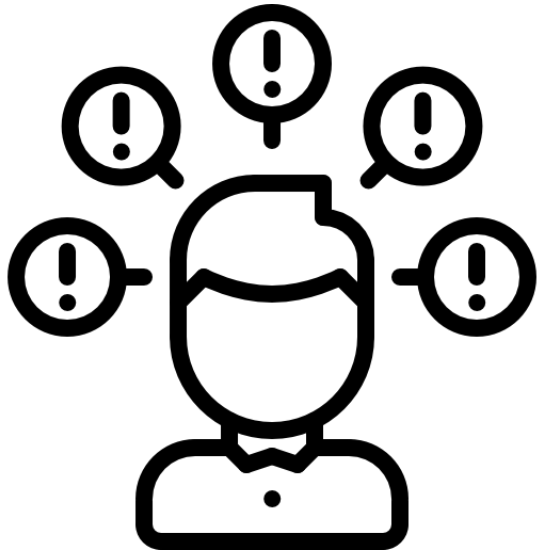


What can go wrong?

Insufficient data

Bad data = Bad model

(bad data can mess up how companies decide things)



Incorrect algorithm selection
Incorrect hyperparameter tuning
Incorrect model deployment
Wrong evaluation metrics
Poorly collected requirements

...

The effect of (bad) Data Quality on Model Accuracy in Supervised Machine Learning

What do we mean by Data Quality?

Completeness
You might make a customer's middle name optional, but as long as you have the first and last name, the data is complete

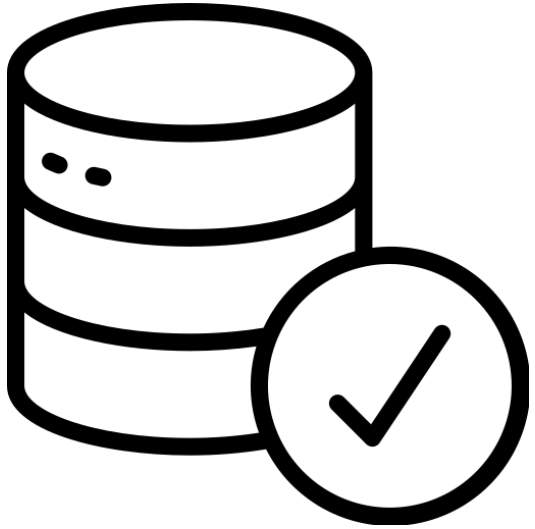
Consistency
Data are represented consistently across the data set

Accuracy
Data accurately represent the "real-world" values

Timeliness
Data represents reality from the required point of time

Validity
For example, ZIP codes are valid if they contain the correct characters for the region

Uniqueness
Some data fields need to contain unique values, if defined. e.g. e-mail



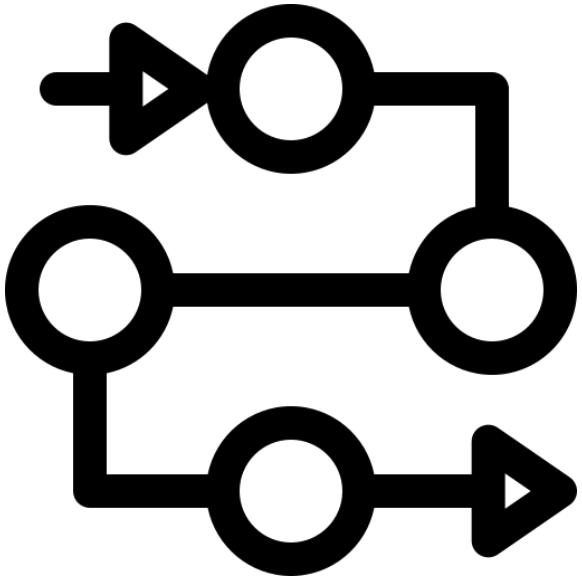
Data Quality process

1. Define requirements
Data profiling

2. Data Quality assessments
Data Quality rules and quality threshold

3. Resolve issues

4. Monitor & Control



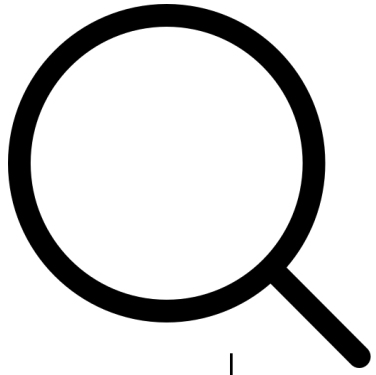
Can Python help us?

- *great_expectations*
- *pydeequ*
- *pandera*

First glance

	great_expectations	***pydeequ***	***pandera***
open source	yes	yes	yes
# stars	8.9k	574	2.6k
# contributors	396	14	105
creation time	6 years ago	3 years ago	4 years ago
last updated	3 days ago	last month	4 days ago
community	slack	-	discord
docs	+	-	+

In greater detail…



	great_expectations	*pydeequ*	*pandera*
pandas support	yes	no	yes
spark support	yes	yes	no
data profiling	yes	yes	no
custom check	yes	yes	yes
simple anomaly detection	yes	yes	no
complex anomaly detection	no	yes	no
hypothesis testing	yes	no	yes
notification	yes	no	yes

So what to choose?

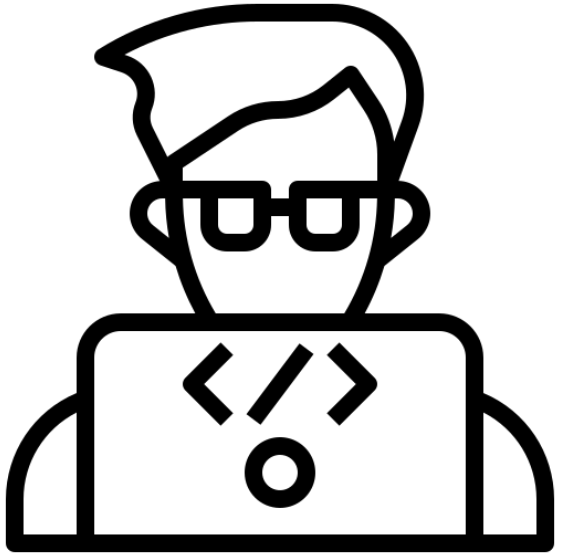
It depends, but...

- *great_expectations*

- Is anomaly detection important to you? *pydeequ*

- Is hypothesis testing important to you? *pandera*

So let's solve our problem



What's next?

- *Better quality code*
- Airflow

```
gx_validate_pg = GreatExpectationsOperator(
        task_id="gx_validate_pg",
        conn_id=POSTGRES_CONN_ID,
        data_context_root_dir=MY_GX_DATA_CONTEXT,
        schema=MY_POSTGRES_SCHEMA,
        data_asset_name="strawberries",
        expectation_suite_name="strawberry_suite",
        return_json_dict=True,
)
```

How to use Great Expectations in an Airflow DAG to perform data quality checks

Questions & answers


mariusz.slapek@gmail.com


slapekm