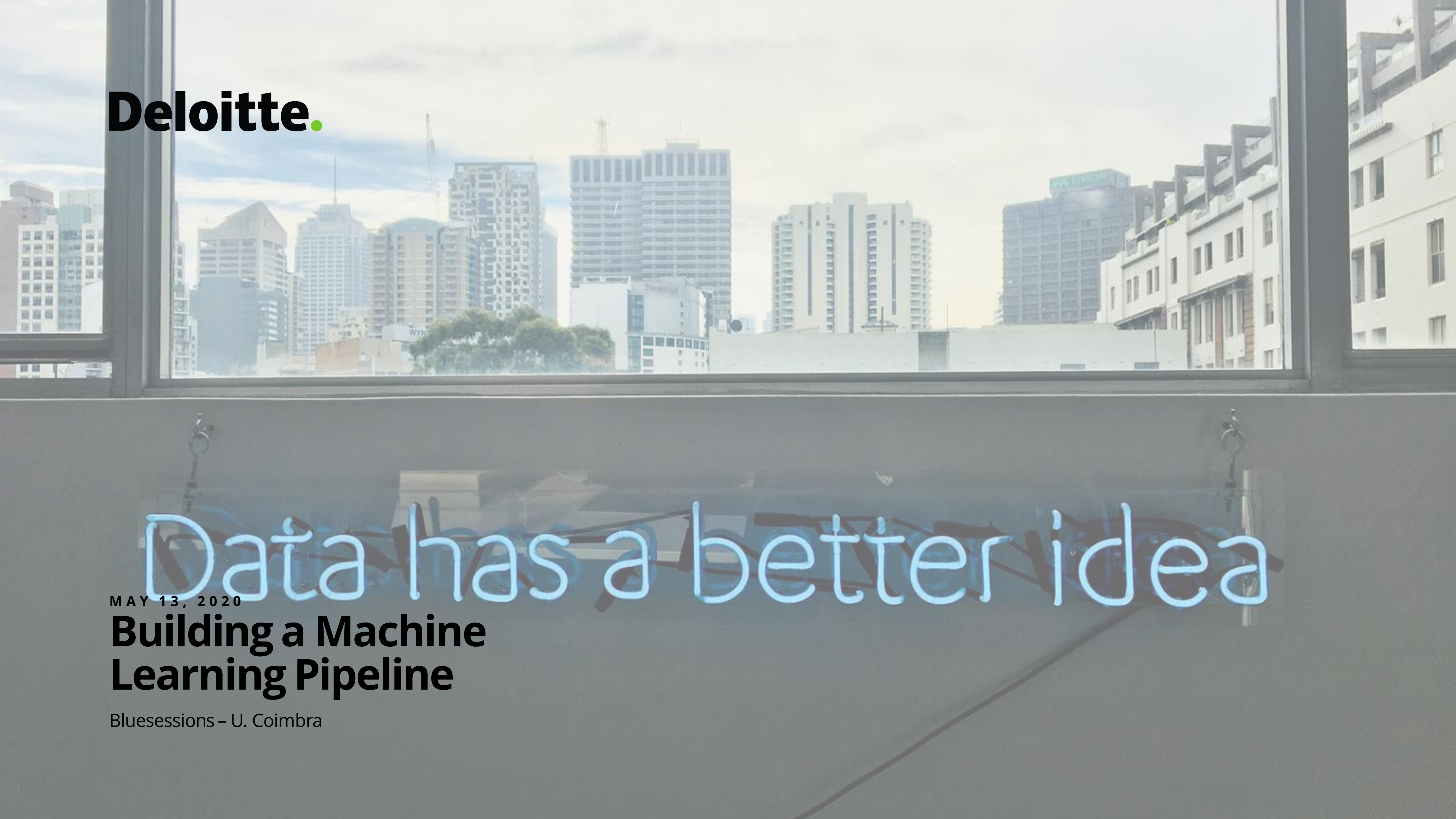
Deloitte.



MAY 13, 2020

Building a Machine Learning Pipeline

Bluesessions – U. Coimbra

Table of Contents

Buzzwords and basic taxonomy

Supervised Learning

- Overview
- Methods
- Problems

Problem 1

Machine Learning Pipelines

- Overview
- Airflow

Problem 2

Objectives

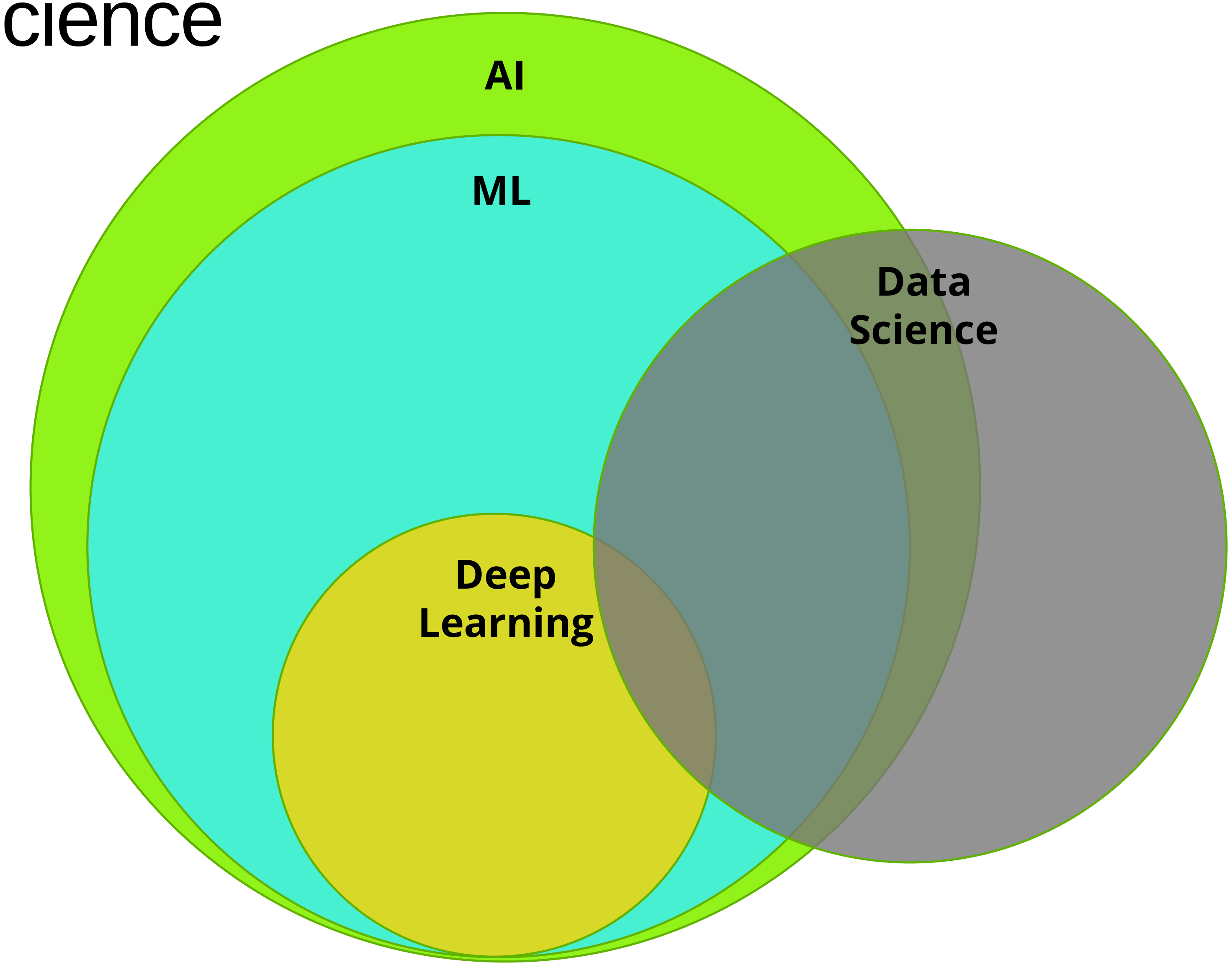
Grasping of basic concepts and ideas

High level understanding of the tools of the trade

Understanding of where to look for more information

AI vs ML vs Data Science

What's the difference?



AI Landscape

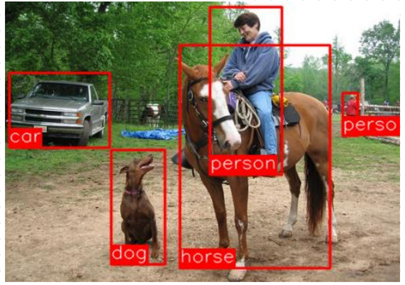
So… are we reaching the point of developing an AI that can mimic the human mind?

Artificial Narrow Intelligence (ANI) ● **Today** **Artificial General Intelligence (AGI)** **Singularity**

A **purpose specific** application of **one or more AI technologies**

Supervised Learning

It works, just scale up!!



Q: How do we get **labels** of intelligent behavior?

A: Collects **lots of labeled data**, then train a big neural network to mimic what humans do. Imitate/generate **human-like well-defined actions**.

Most of the applications!!

Unsupervised Learning

It will work, if we only scale up!

This small blue bird has a short pointy beak and brown on its wings
This bird is completely red with black wings and pointy beak
A small sized bird that has a cream belly and a short pointed bill
A small bird with a black head and wings and features grey wings

Q: How do we generate from **unknown data**?

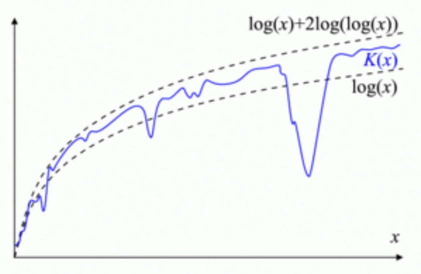
A: Initialize a big neural network; train it to compress a huge amount of data, recognize similar objects, **learn some attributes**, and **group them together**.

Not implemented on a wider scale yet!!

A **theoretical AI** that could successfully perform **any intellectual task** that a human being is capable of.

Reinforcement Learning

Hey! I can write down optimal AI.



Q: How do we take over the hypothesis space of all **Turing machines**?

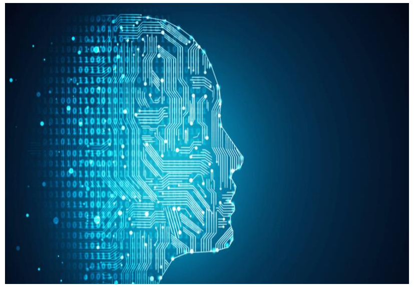
A: Formal definition of "Universal Intelligence", learning agents that determine what the **ideal behaviour within a context can be.**

Constraint environments, optimal results, sub-optimal process

Creation of **superintelligent machines**

Artificial Life

Just do what nature did.



Q: How to obtain **intelligence design**?

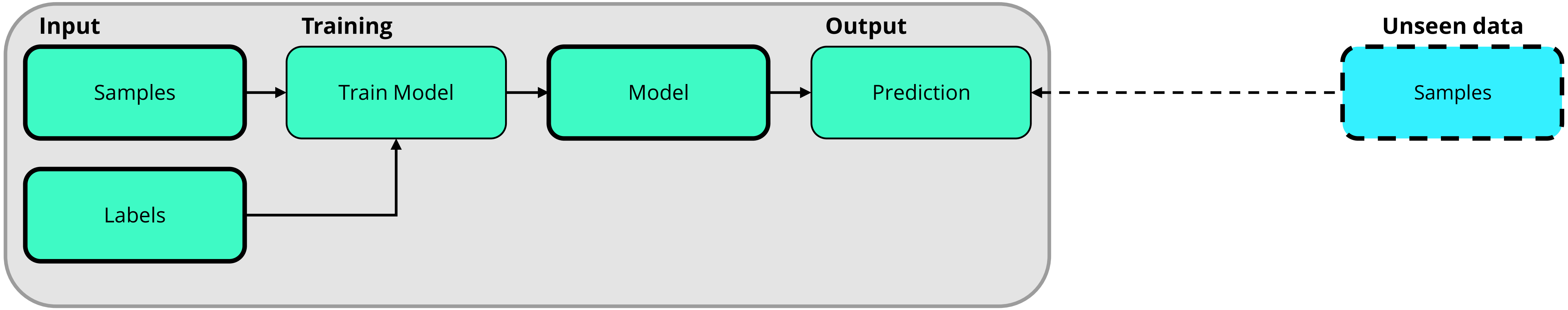
A: …

Don't be afraid … just wait and see, or accept it and be part of it

Supervised Learning

Supervised learning

***"Teach by example"** – function approximation*



Supervised Learning

Most Common Problem Settings

Classification

- Discrete Labels
- Examples:
  - Cat vs dog
  - Is it going to rain tomorrow or not?

Regression

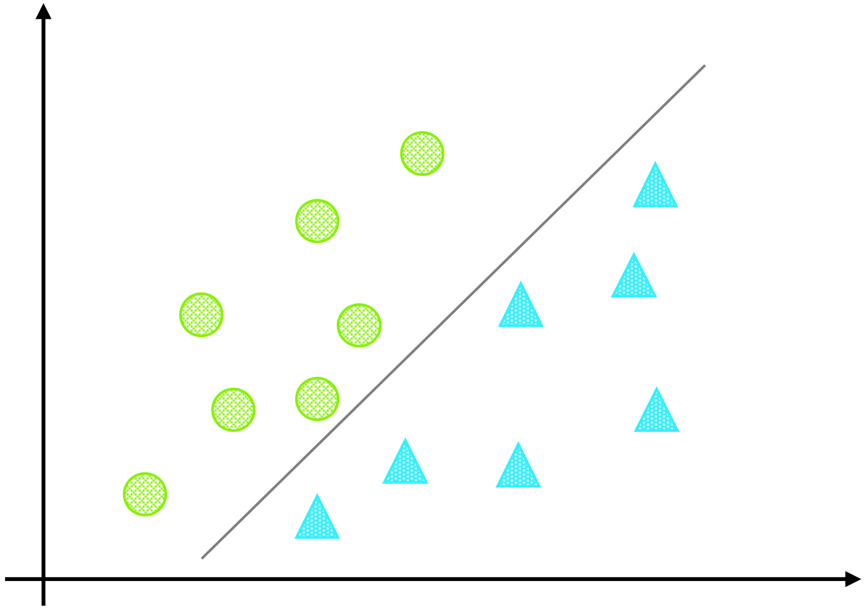
- Continuous Labels
- Examples:
  - Predict the temperature tomorrow
  - Predict the price of a commodity

Linear Classification

Linear Combination of features

Key Aspects

- Simple and fast
- Output is assumed to be a linear comb. of the features
- What if it’s not?

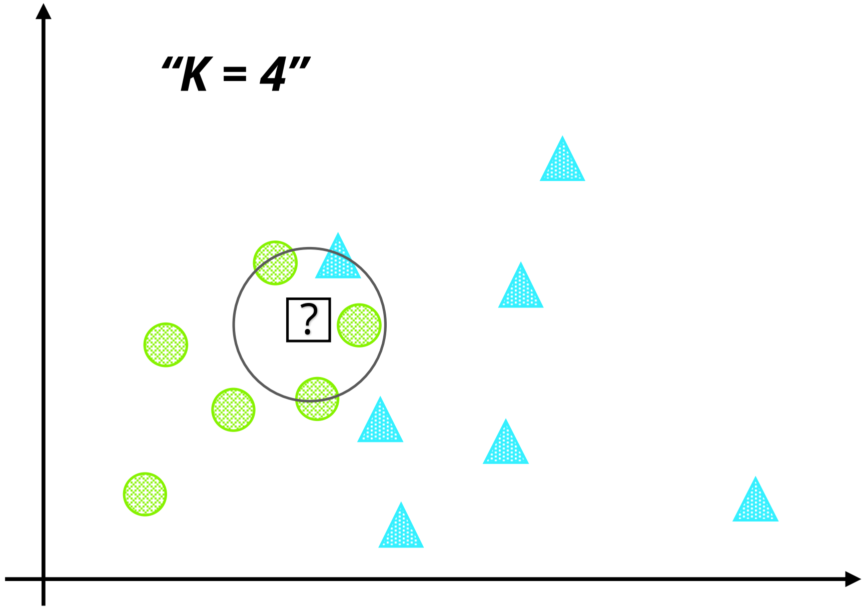


K Nearest Neighbours

Instance Similarity

Key Aspects

- Data points that are close are assumed to be similar
- No training *per se*
- How many points to consider?
- What do we mean by distance?

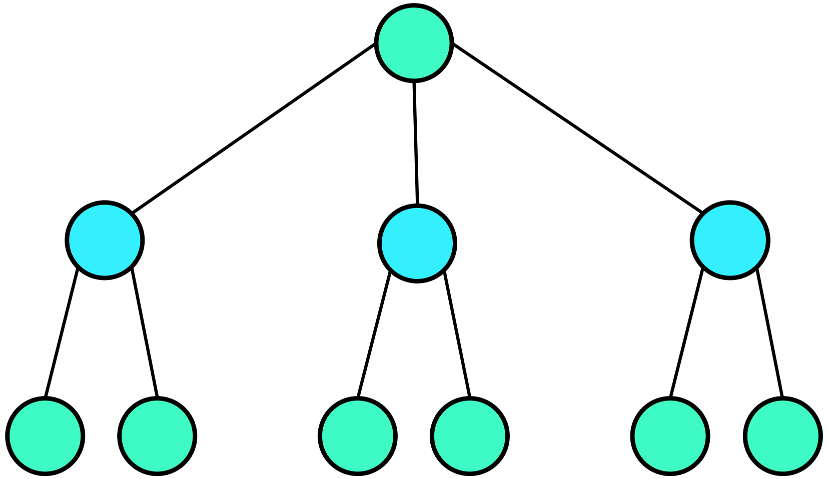


Decision Trees

Divide n' conquer

Key Aspects

- Features space is divided in sections where instances are similar
- What variables to use to split?
- At what value should we split?
- When should we stop breaking up the feature space?

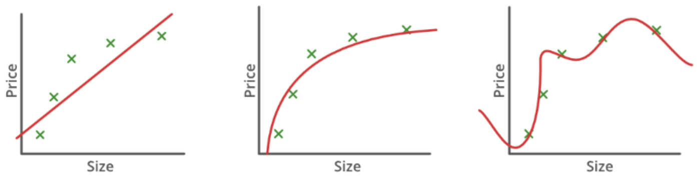


Typical Problems

Overview

Overfitting vs Underfitting

- We want a model that learns from the data only enough to be able to generalize
- We don't want the model to capture every tiny detail of the dataset

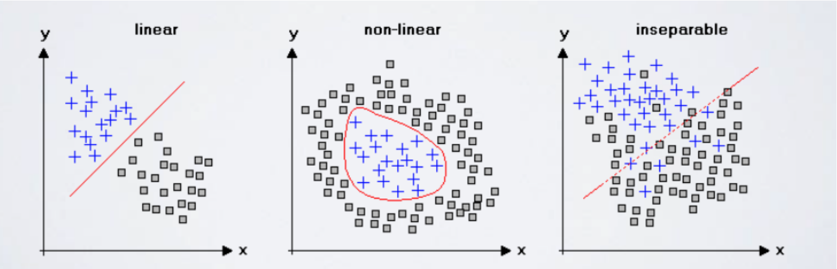


Curse of dimensionality

- When we're talking about many dimensions, most of our intuitions don't apply.
- Distance functions don't behave as expected
- Sampling needs increase exponentially

Model Specificities

- Different models make different assumptions about the data
- Different models deal differently with different kinds of data



HANDS-ON

Let’s Code

Supervised Learning

1) Improving and debugging a ML pipeline

You’ll be given a very basic implementation of what aims to be a machine learning pipeline. It’s your job to go through it, identify the main shortcomings of the code you’ll be given, specially the ones already highlighted there for you, and provide solutions for them.
You’ll also be given a document that provides you with possible solutions for some of the shortcoming; That’s so that nobody get’s stuck, but you should not look at it unless you’ve thought about the problem yourself first.

In the code you’ll be given, there will be a dataset with all the matches from 2011 onwards for **Liga NOS.** It’s your job to be able to predict the outcome of future games as accurately as possible**.** The dataset was gathered by doing web-scraping. You’ll also be given the code that does the scraping, along with some preprocessing code that prepares the data for being used, but you should not worry about any of those for now, we’ll use them later to build your pipeline. The dataset already has some features for you to work with, but we suggest that you do your own feature engineering **once you complete the other steps**, if you feel like doing it. There’s also a data dictionary that explains what each variable is.

Machine Learning Pipelines

The case for machine learning pipelines

- Automation
- Scheduling
- Fast track to production
- Continuous Delivery

Airflow

What it is, and where did it came from?

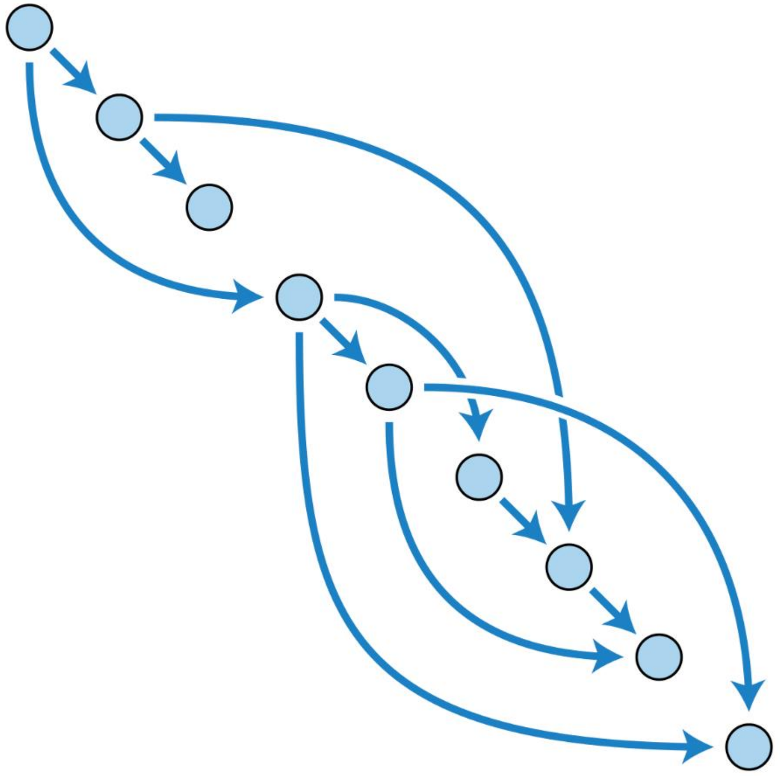
A platform to programmatically **author**, **schedule** and **monitor** workflows.

- Airbnb started it;
- Now is part of the Apache ecosystem;

Airflow

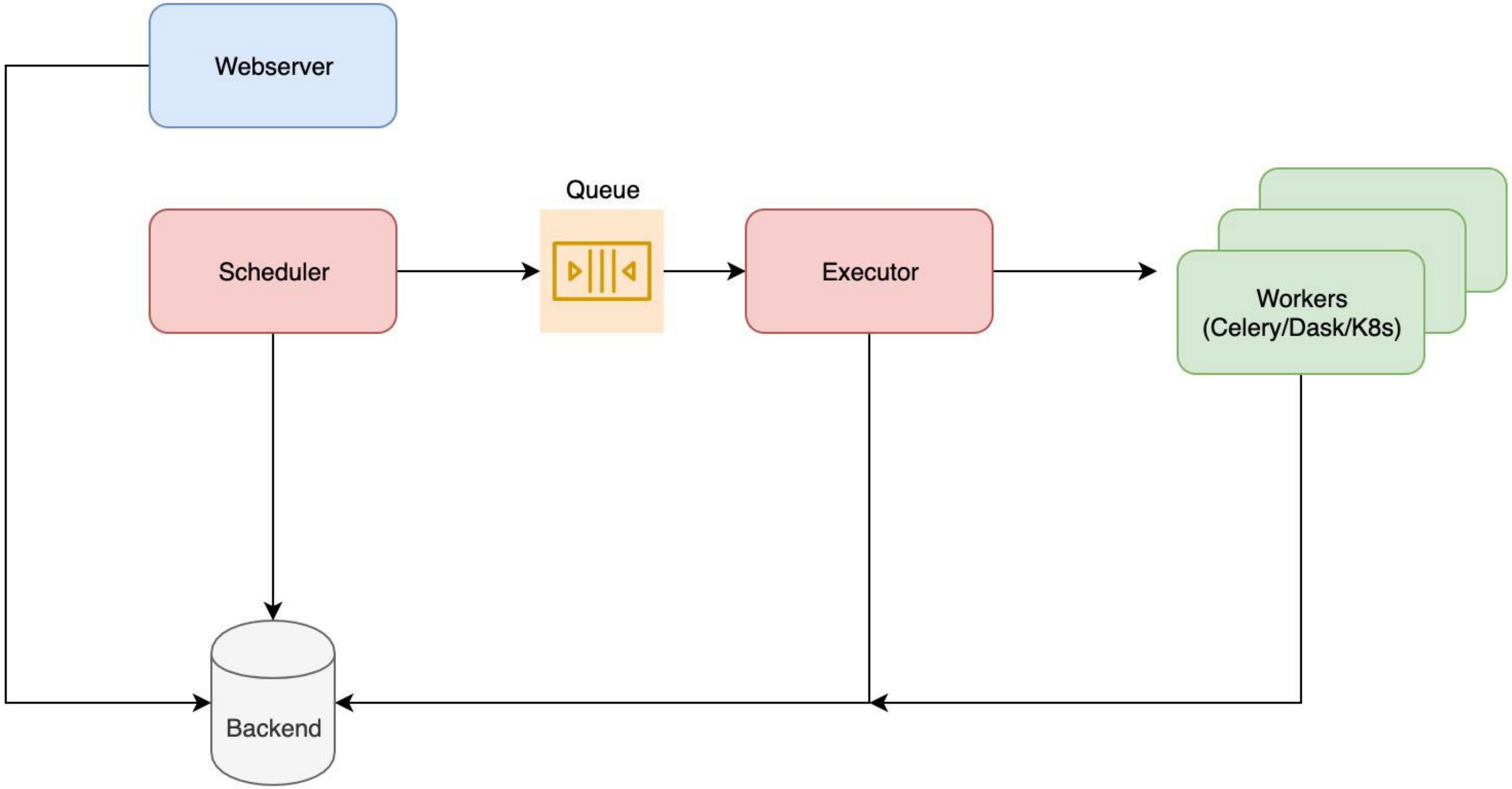
What it is used for?

- Cron jobs.
- ETL.
- Automating DevOps.
- Scrapping.
- Data processing for recommendation systems.
- Machine Learning Pipelines.
- …
- **If it runs on your shell, you can run in on Airflow**



Airflow

Architecture



Airflow

Common terms and concepts

- **DAG** – The way workflows are represented;
- **DAG Run** – A particular execution of a DAG;
- **Scheduler** – The software that takes care of executing the right operations at the designated time;
- **Operator** – Your task/job or function;
- **XCom** – Communication mechanism between tasks;
- **Sensors** – Components that possibly trigger execution;

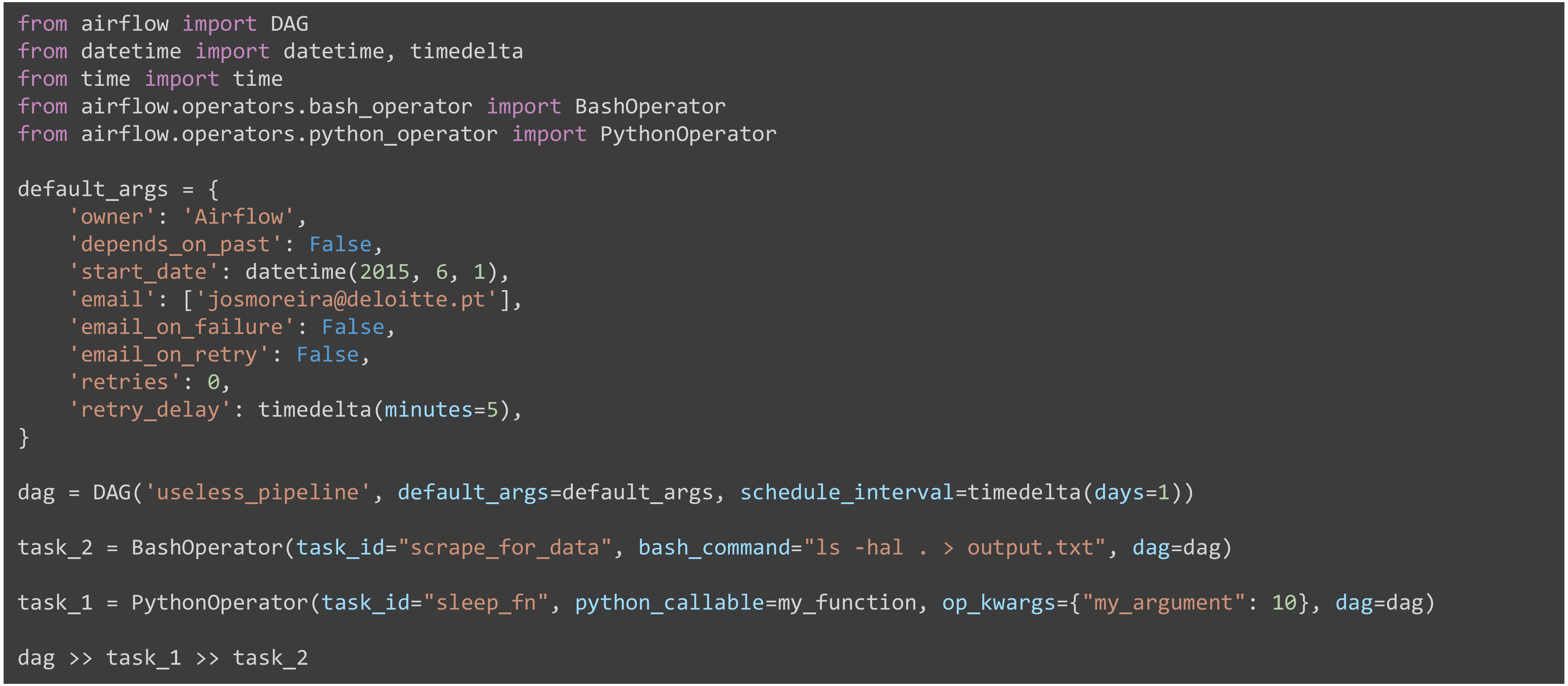
Airflow

What types of tasks can be executed?

- **BashOperator** – Anything that runs on the bash
- **PythonOperator** – Anything that is written in Python
- Other specific operators:
  - EmailOperator
  - SimpleHTTPOperator
  - MySQLOperator
  - JdbcOperator
  - DockerOperator
  - HiveOperator
  - S3FileTransformOperator
- ….

Airflow

From Operators to DAGs



```python
from airflow import DAG
from datetime import datetime, timedelta
from time import time
from airflow.operators.bash_operator import BashOperator
from airflow.operators.python_operator import PythonOperator

default_args = {
    'owner': 'Airflow',
    'depends_on_past': False,
    'start_date': datetime(2015, 6, 1),
    'email': ['josmoreira@deloitte.pt'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 0,
    'retry_delay': timedelta(minutes=5),
}

dag = DAG('useless_pipeline', default_args=default_args, schedule_interval=timedelta(days=1))

task_2 = BashOperator(task_id="scrape_for_data", bash_command="ls -hal . > output.txt", dag=dag)

task_1 = PythonOperator(task_id="sleep_fn", python_callable=my_function, op_kwargs={"my_argument": 10}, dag=dag)

dag >> task_1 >> task_2
```

HANDS-ON

Let’s Code

Airflow

2) Setting up your ML Pipeline

Now that you’ve went through all the steps to fix your machine learning models, you’re ready to install all the pipeline on Apache Airflow. For this task, you’ll be provided a folder where your DAG specifications should be placed, along with a couple of example DAG definitions. We’ve already broken up the pipeline you were fixing just now into all the required bits and pieces, and we’ve added some things to make it’s integration easier in airflow.

You’ll be sharing a single Airflow instance with the rest of your colleges, and therefore, your DAG names will have to be unique. We advice you to use your own name for the pipeline so as to make it distinguishable from your colleges’.

Thank you.

José Trocado Moreira

Senior Specialist Data Science / AI, Deloitte Consulting LLP

Contact: josmoreira@deloitte.pt

This publication contains general information only, and none of the member firms of Deloitte Touche Tohmatsu Limited, its member firms, or their related entities (collective, the "Deloitte Network") is, by means of this publication, rendering professional advice or services. Before making any decision or taking any action that may affect your business, you should consult a qualified professional adviser. No entity in the Deloitte Network shall be responsible for any loss whatsoever sustained by any person who relies on this publication.

As used in this document, "Deloitte" means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of the legal structure of Deloitte USA LLP, Deloitte LLP and their respective subsidiaries. Certain services may not be available to attest clients under the rules and regulations of public accounting.