Sync Lab
Synchronous with technologies

GDP
Gathering Detection Platform

Benvenuti

Cristoforo Decaro
Email
Linkedin

- Machine Learning developer at SyncLab
- Phd in Electronic Engineering at Università di Ferrara
- Master degree in Electronic Engineering at Politecnico di Torino

Problema:

In seguito alla pandemia del virus COVID-19, i cittadini sono stati costretti a quarantena e soprattutto ad evitare assembramenti

Soluzione:

Creare una piattaforma in grado di rappresentare, mediante rappresentazione grafica/ dashboard delle zone potenzialmente a rischio assembramento.

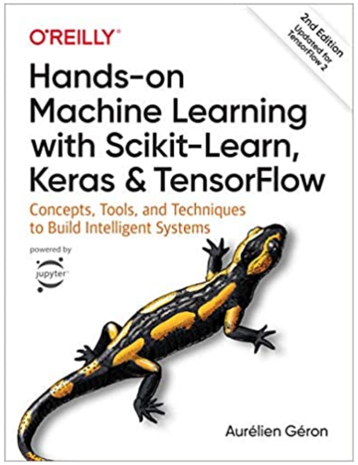
Come:

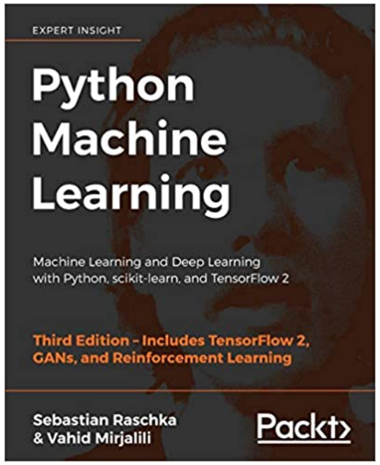
Bisogna sfruttare le tecniche di **machine learning** e/o **deep learning** per predire le zone potenzialmente a rischio.

Risorse disponibili:

Risorse:

1) Libri



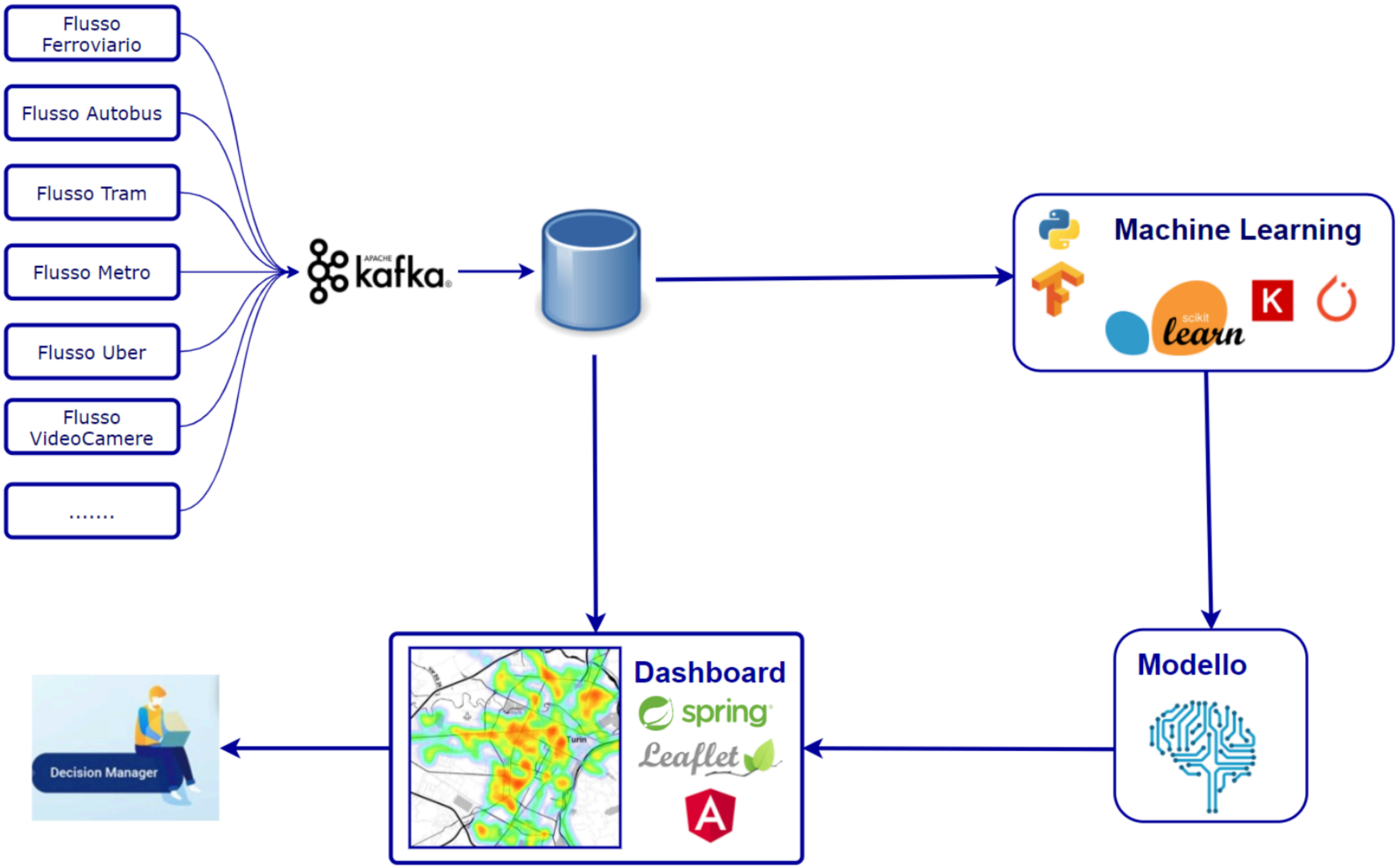


2) Risorse online

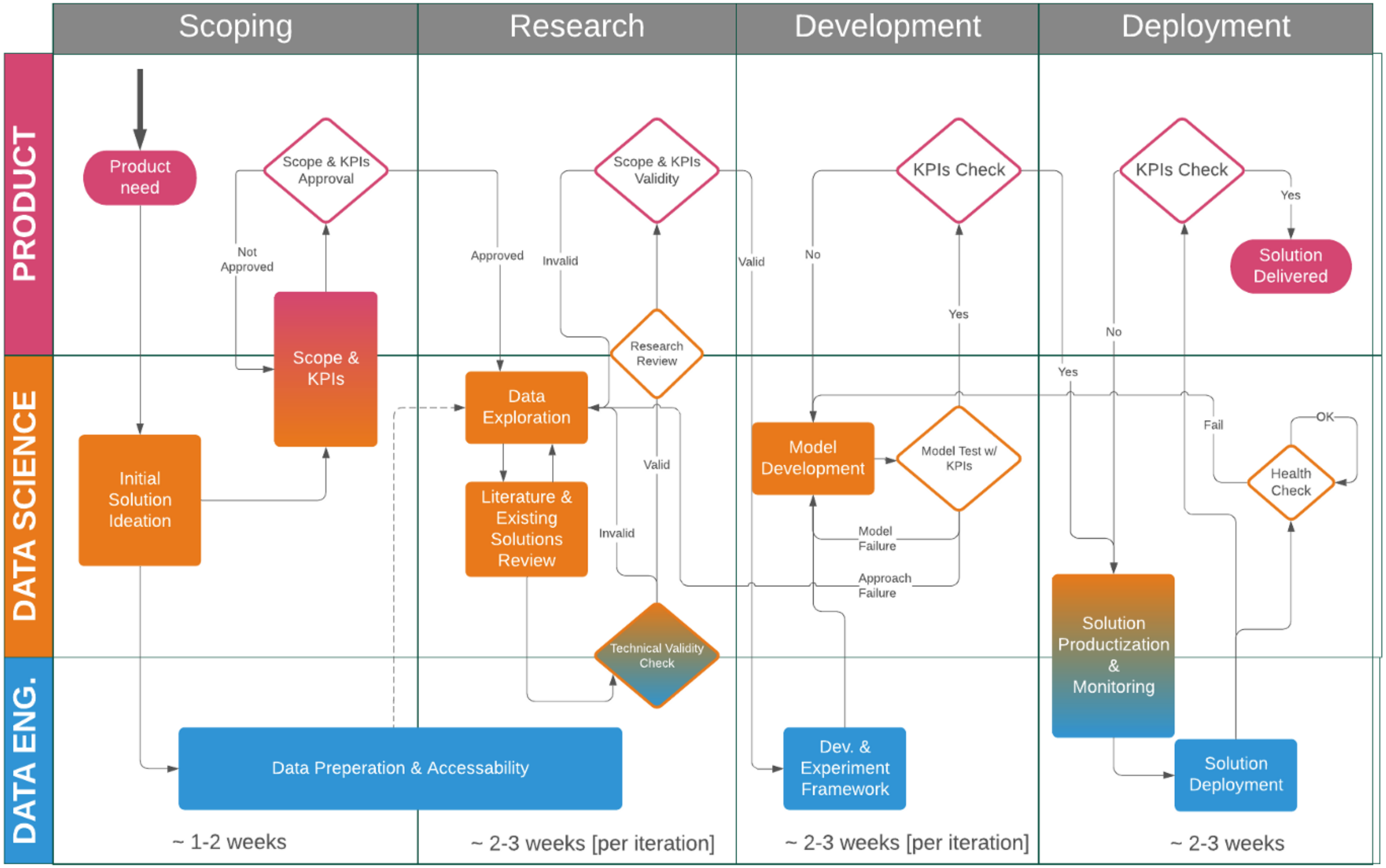
- [Kaggle]
- [Aurelien Geron's GitHub]
- [Keras documentation]
- [Tensorflow guide]
- [Pytorch guide]
- [Scikit-Learn guide]

3) Google scholar

Esempio di architettura:



Esempio di architettura complessa:



Sync Lab

Data science:

1 Research

- Meet with stakeholders
- Gather requirements
- Read scientific papers
- Gather data insights
- Decide on methodology

Deliverable(s)

- Detailed project execution plan

2 Data Exploration

- Slice the data in different ways
- Counts, graphs, distributions, histograms

Deliverable(s)

- Report with the results

3 Modeling

- Build ETL
- Train model
- Evaluate model with different metrics and baselines
- Sanity check the model output

Deliverable(s)

- Model prototype
- Model evaluation report

4 Productization

- Clean and comment code
- Instrument code with metrics reporting and alert thresholds
- Optimize for runtime and memory
- Code review by scientist and engineer

Deliverable(s)

- Final (code reviewed) model

5 A/B testing

- QA test
- Monitor the test performance
- Work on other tasks/projects while test is running

Deliverable(s)

- No deliverables

6 Result Analysis

- Deep-dive analysis of A/B testing results

Deliverable(s)

- Report with the results
- Hypothesis as to why the test failed/succeeded

Sync Lab

Fasi del progetto:

Data exploration

Preprocessing

Training

Evaluation / Validation

Deploy

Data exploration

<u>Data exploration:</u>

- Presenza di valori mancanti, outliers, dataset non bilanciati
- Livello di rumore dei dati
- Correlazione dei dati

$$
\begin{array}{cccccccl}
 & & & \text{variable} & & & \text{target} & \\
X_{11} & X_{12} & X_{13} & \dots & X_{1d} & C_1 & & \\
X_{21} & X_{22} & X_{23} & \dots & X_{2d} & C_2 & & \text{observation} \\
\dots & & & & & & & \\
X_{n1} & X_{n2} & X_{n3} & \dots & x_{nd} & C_n & &
\end{array}
$$

Preprocessing

<u>Preprocessing:</u>

- Cleaning
- Trasformazione dei dati (normalizzazione, discretizzazione, aggregazione, calcolo di nuove variabili…)
- Feature extraction
- Filtraggio dei dati
- Train / Test set splitting

Come dividere i dati?

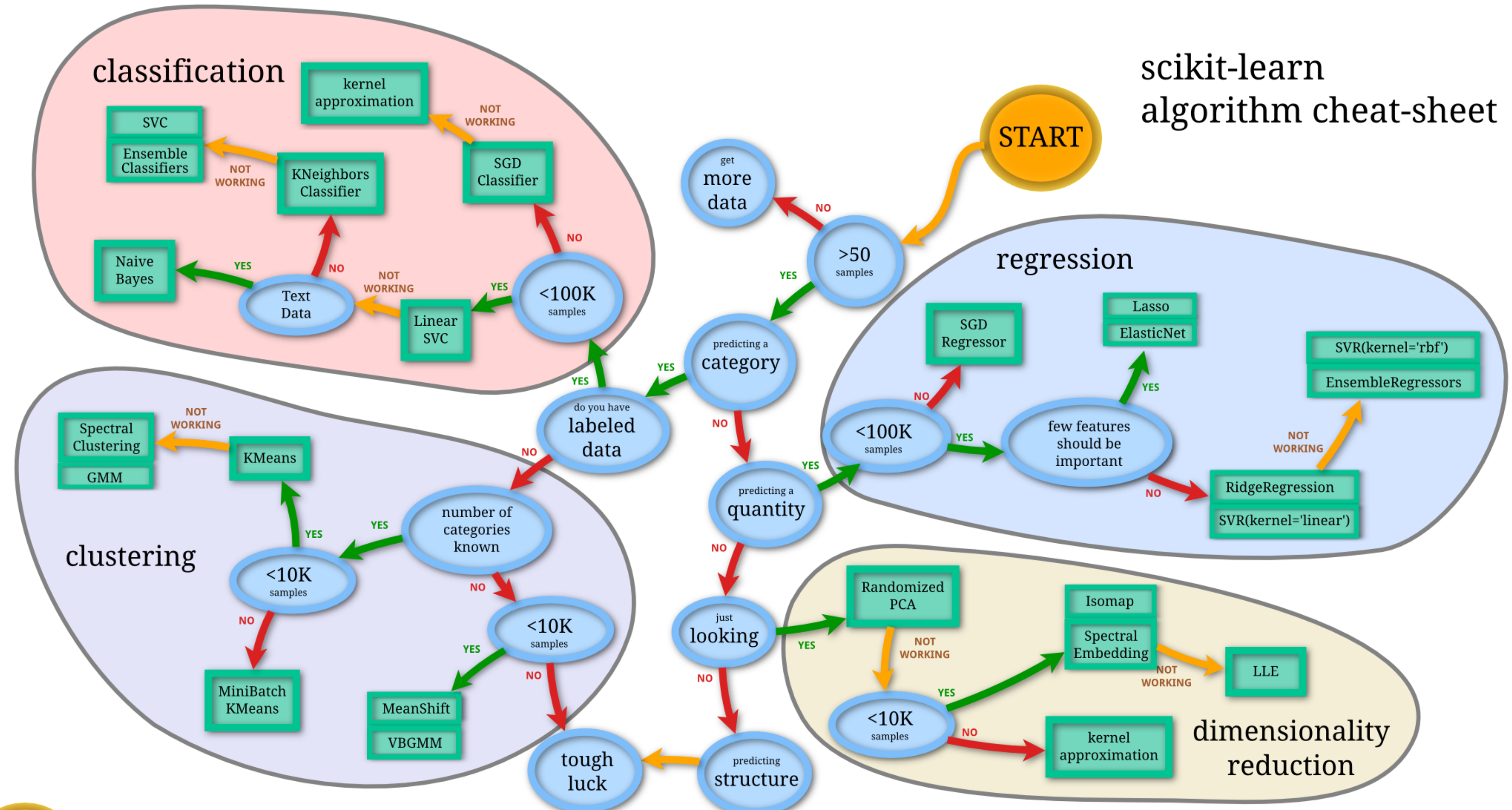
- Se i dati hanno una componente **temporale** il training set conterrà i dati **meno recenti**, il validation test e il test set i **più recenti**.
- Se i dati non dipendono dal tempo, la divisione viene effettuata in maniera **casuale**.

Caso Predizione

1) Classificazione (Categorical target)
   - Naive Bayes
   - Decision Trees
   - Neural Networks
   - KNN
   - Support Vectors Machine
   - …

2) Regressione (Numerical target)

Scegliere l’algoritmo giusto:



Evaluation and validation:

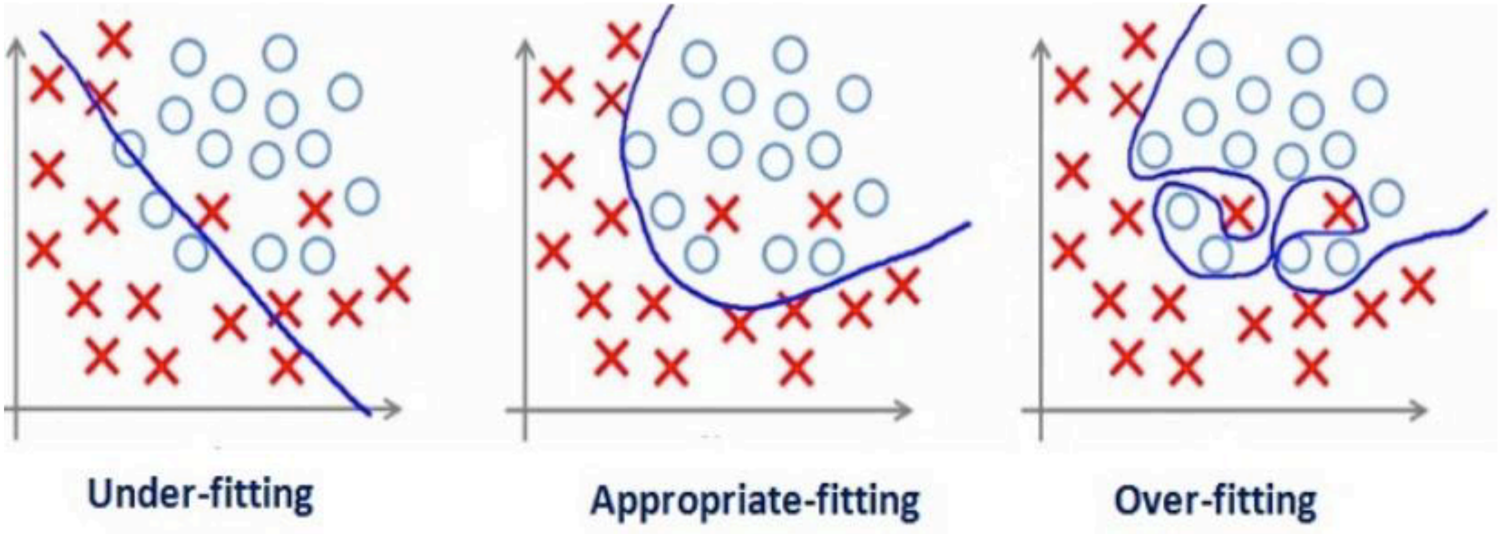
Valutazione

- Metriche di valutazione: r2, MSE, MAE, correlation matrix, ROC curve …

Ottimizzazione

- Ottimizzazione degli iperparametri: trials and errors, grid-search, random-search,…

Overfitting:



Per risolvere il problema dell’**overfitting** di solito il dataset viene diviso in tre parti:

- **Training set**
- **Validation set**
- **Test set**

I diversi **modelli** (ottenuti cambiando **algoritmo** o il valore degli **iperparametri**) vengono addestrati sul training set e testati sul validation set per testare la loro capacità di generalizzazione.

Checklist:

- Data Assumptions ✓
- Preprocessing ✓
- Training ✓
- Loss/Evaluation Metric ✓
- Overfitting ✓
- Runtime ✓

Outcomes:

- Report:
  - Project scope and product need
  - Algoritmo e approccio scelto
  - Dati utilizzati, preprocessing
  - Modelli esaminati, training
  - Comparazione dei risultati tramite scelta delle metriche adatte
  - Ottimizzazione degli iperparametri

- Models:
  - Salvataggio del modello
  - Load del modello pre-allenato

Contatti:

Ing. Fabio Pallaro

email: f.pallaro@synclab.it

Cristoforo Decaro

email: cristoforo.decaro@synclab.it