

PROGETTO

Marta Fioravanti

Knowledge Management

prof. Luca De Biase

aa. aa. 2019 - 2020



UNIVERSITÀ DI PISA

Idea

Il progetto consiste in una piattaforma online per aiutare l'utente a sviluppare data literacy, ossia la capacità di comprendere i dati e utilizzarli per estrarne conoscenza. In concreto si tratta di una web application che funge da deposito di sapere legato soprattutto agli algoritmi di machine learning per l'analisi e per il trattamento dei dati. La piattaforma è un punto d'incontro tra contenuti specialistici e divulgativi; infatti, i suoi obiettivi sono:

- fornire le nozioni basilari ai neofiti
- aiutare chi sta studiando un algoritmo a comprendere ogni passaggio
- connettere il sapere formalizzato con la competenza tecnica

Il progetto vuole offrire un'esperienza accessibile e immersiva. L'intento dell'operazione è di rendere il contenuto "memorabile", comunicando i concetti anche attraverso l'immagine: le analogie visive infatti possono aiutare a compiere un salto intuitivo nella comprensione di un concetto.

La piattaforma propone un approccio di long-life-learning svincolato da un percorso formativo: è un invito alla libera esplorazione dei contenuti; la sua fruizione si avvicina alla consultazione di un'enciclopedia: la motivazione che spinge l'utente è di accrescere la propria conoscenza in modalità occasionale e rilassata.

Per realizzare gli intenti appena espressi, la conoscenza deve essere organizzata su più livelli comunicanti. Nel complesso, il progetto vuole garantire al lettore i suoi dieci diritti enunciati da Daniel Pennac:

- Il diritto di non leggere
l'utente può fruire dei contenuti anche senza "leggere", nel senso stretto del termine
- Il diritto di saltare delle pagine
non essendoci un ordine, può saltare da un argomento all'altro senza restrizioni
- Il diritto di non finire un libro
non c'è un percorso da completare, ma una piattaforma in continua evoluzione
- Il diritto di rileggere
l'intento è proprio di poter tornare sui propri passi per capire sempre meglio un concetto

Pennac, "Comme un Roman", Gallimard, 1992

- Il diritto di leggere qualsiasi cosa
per pura curiosità
- Il diritto al bovarismo (malattia testualmente
transmissibile)
sarebbe bello se l'utente si trovasse a fantasticare su
come sfruttare le conoscenze acquisite per realizzare i
propri sogni
- Il diritto di leggere ovunque
(purché ci sia internet)
- Il diritto di spulciare
la piattaforma offre conoscenza "a snack"
- Il diritto di leggere ad alta voce
i contenuti visivi possono essere raccontati, come si fa
con un libro illustrato
- Il diritto di tacerci
soprattutto se non basiamo il nostro discorso su dati
verificati

Struttura

I contenuti della piattaforma si dividono in due parti:

1. studio teorico degli algoritmi
2. messa in pratica delle conoscenze acquisite

Esse sono indipendenti e l'accesso alla seconda non è vincolata dall'utilizzo della prima. Iniziare dalla sezione teorica aiuta tuttavia a familiarizzare con l'argomento, quando si va a utilizzare un particolare algoritmo.

Studio teorico

Uno dei problemi che si incontrano studiando data science è la difficoltà di lettura dei paper specialistici e dello pseudocodice che formalizza gli algoritmi di intelligenza artificiale.

Ogni pagina teorica della piattaforma contiene:

- una spiegazione verbale dell'algoritmo, assimilabile alla descrizione del concetto da parte di un divulgatore
- la formalizzazione in pseudocodice
- la trascrizione in uno o più linguaggi di programmazione
- la rappresentazione animata dei concetti spiegati

The screenshot displays the 'K Nearest Neighbour' page. At the top is a green header with the title 'K Nearest Neighbour' and a hamburger menu icon. Below the header, the page is divided into four main sections: 'Pseudocode', 'Code', 'Description', and 'Illustration'. The 'Pseudocode' section contains a high-level description of the algorithm. The 'Code' section shows a Python implementation. The 'Description' section provides a verbal explanation of the algorithm. The 'Illustration' section features a scatter plot and a cartoon character. Arrows from the list of features point to these sections: 'una spiegazione verbale dell'algoritmo' points to the 'Description' section; 'la formalizzazione in pseudocodice' points to the 'Pseudocode' section; 'la trascrizione in uno o più linguaggi di programmazione' points to the 'Code' section; and 'la rappresentazione animata dei concetti spiegati' points to the 'Illustration' section.

title

K Nearest Neighbour

Pseudocode

```
0 X ← training data; Y ← class labels; x ← unknown sample; k ← considered neighbours
1 for i = 1 to m do
2   compute distance d(Xi, x)
3 end for
4 Compute set I containing indices for the k smallest distances d(Xi, x)
5 return majority label for {Yi where i ∈ I}
```

Code

```
0 # Example of making prediction
1 from math import sqrt
2 # Euclidean distance between two vectors
3 def euclidean_distance(row1, row2):
4   distance = 0
5   for i in range(len(row1)-1):
6     distance += (row1[i] - row2[i])**2
7   return sqrt(distance)
8
9 # Define the number of neighbours
10 n_neighbors = 4
11 # Locate the most similar neighbor
12 def get_neighbors(train, test_row, num_neighbors):
13   distances = list()
14   for train_row in train:
15     dist = euclidean_distance(test_row, train_row)
16     distances.append((train_row, dist))
17   distances.sort(key=lambda tup: tup[1])
18   neighbors = list()
19   for i in range(num_neighbors):
20     neighbors.append(distances[i][0])
21   return neighbors
22
23 # Make a classification prediction with neighbor
24 def predict_classification(train, testnum_neighbors):
25   neighbors = get_neighbors(train, testnum_neighbors)
26   output_values = [row[-1] for row in neighbors]
27   prediction = max(set(output_values), key=output_values.count)
28   return prediction
```

Description

2. Since the label is assigned according to the most similar records' classes in the training set, a value k, indicating how many neighbours consider, is required. A criterion to measure distance is also needed.

3. During the execution, the algorithm computes the distance between every record to classify and every row in the training set, and keeps the k nearest

Illustration

La ridondanza è la forza di questo sistema: se un passaggio in una sezione non è chiaro, può essere letto in altri tre modi; infatti, se si seleziona una porzione di testo, tutte le

parti della pagina si aggiornano sul contenuto corrispondente.

Spezzare un'argomentazione in porzioni più brevi alleggerisce lo sforzo di lettura e permette all'utente di progredire rispettando i propri tempi, cosa che in una video-lezione non accade, in quanto il flusso del discorso di chi parla è continuo; la disposizione non lineare dei contenuti permette esprimere concetti complessi in modo completo, accessibile e gradevole, senza generare il "panico da muro di testo" che alcuni lettori incontrano di fronte a un libro di testo o a un paper scientifico.



"Pokémon Diamante", Nintendo, 2007
Nei videogiochi è frequente spezzare i dialoghi in piccole porzioni che avanzano sotto l'input del giocatore: questo stratagemma permette di alleggerire il carico visivo del testo e di mantenere alta l'attenzione dell'utente.

Messa in pratica

La sezione pratica della piattaforma permette di applicare su problemi giocattolo gli algoritmi studiati nella teoria. Essa divisa in due parti: risoluzione a mano di problemi semplificati; utilizzo del codice per analizzare dataset veri e propri.

Do it by hand

In questa sezione sono proposti brevi esercizi guidati risolvibili a mano, che aiutano ad acquisire una comprensione più profonda dei concetti. Seguire i passaggi di un algoritmo e fare i calcoli è più utile a capire le sue meccaniche, rispetto ad applicare la funzione di una libreria in python. L'idea è di costringere l'utente a pensare a cosa succede ai dati quando sono elaborati dall'algoritmo.

K-Nearest Neighbour

step 1
We find the points nearest to **A** and **B**

Assign the class to points **A** and **B**, using 3 nearest neighbours and Manhattan as distance metric.

Manhattan Distance
 $\text{dist}(A, B) = |A-B|$
 $\text{dist}(2, 7) = |2-7| = |-5| = 5$

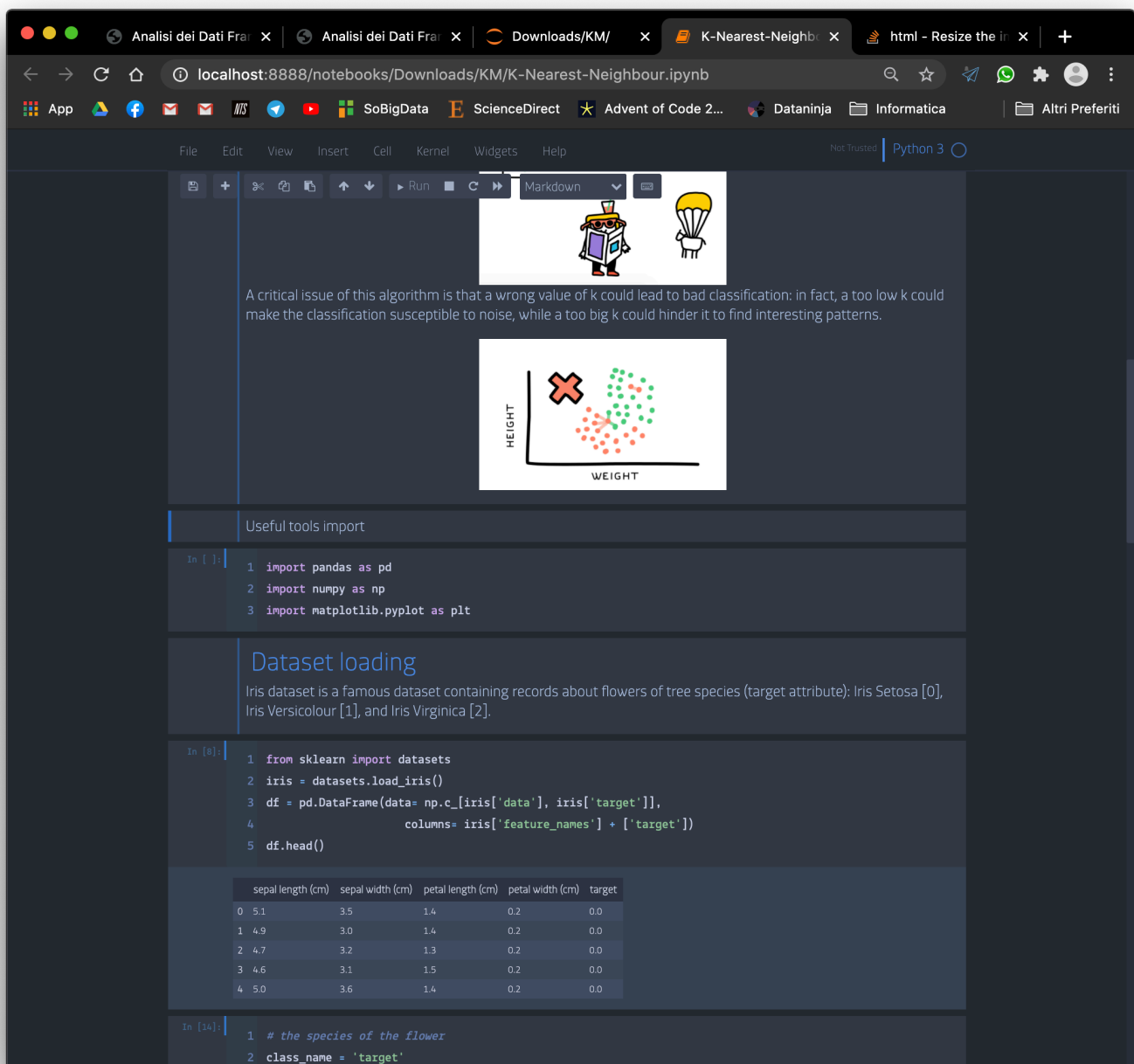
Per ogni tecnica di machine learning, viene esposta la risoluzione passo passo di un esercizio; in certi casi, a causa della complessità dei calcoli, sarà affrontata solo una serie ridotta di passaggi.

Hands on code

Questa sezione offre delle guide pratiche sotto forma di jupyter notebook, dei file python che alternano testo corrente e immagini a porzioni di codice. Essi sono molto utili per seguire il ragionamento dietro alle scelte di analisi e risultano più fruibili di un semplice script python in quanto l'output di ogni cella è visibile sotto la stessa.

Il codice è commentato sia per spiegare le funzioni utilizzate che per descrivere scelte di analisi effettuate.

Ogni notebook inizia con la descrizione dell'algoritmo, in modo da ricordare all'utente le sue meccaniche principali.



The screenshot shows a Jupyter Notebook interface with the following content:

- Title Bar:** Shows the browser address bar with the URL `localhost:8888/notebooks/Downloads/KM/K-Nearest-Neighbour.ipynb` and several open tabs.
- Toolbar:** Includes icons for file operations, editing, and running code.
- Cell 1:** Contains a title "K-Nearest-Neighbour", a cartoon character, and a paragraph explaining a critical issue of the algorithm: "A critical issue of this algorithm is that a wrong value of k could lead to bad classification: in fact, a too low k could make the classification susceptible to noise, while a too big k could hinder it to find interesting patterns."
- Cell 2:** Contains a scatter plot titled "HEIGHT" vs "WEIGHT" showing three clusters of points (red, green, and blue) and a red 'X' mark.
- Cell 3:** Contains the text "Useful tools import" and the following code:

```
In [1]: 1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
```
- Cell 4:** Contains the title "Dataset loading" and the text "Iris dataset is a famous dataset containing records about flowers of three species (target attribute): Iris Setosa [0], Iris Versicolour [1], and Iris Virginica [2]."
- Cell 5:** Contains the following code:

```
In [8]: 1 from sklearn import datasets
2 iris = datasets.load_iris()
3 df = pd.DataFrame(data= np.c_[iris['data'], iris['target']],
4                     columns= iris['feature_names'] + ['target'])
5 df.head()
```
- Cell 6:** Displays the output of the code in Cell 5 as a table:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

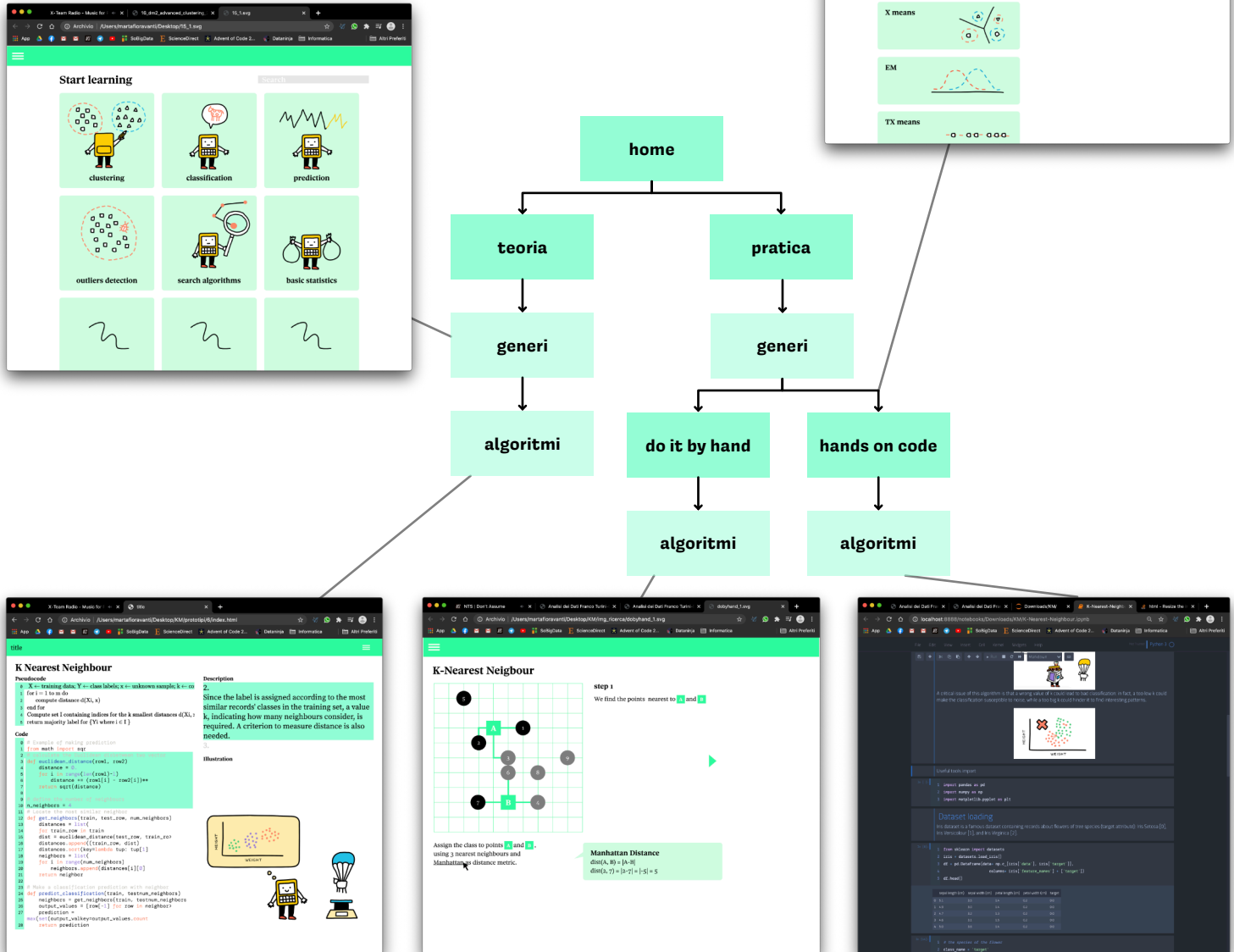
- Cell 7:** Contains the following code:

```
In [14]: 1 # the species of the flower
2 class_name = 'target'
```

Hands on code non presenta veri e propri esercizi, bensì degli esempi pratici: sugli stessi dati si possono condurre molteplici analisi e le strategie per rispondere ad una stessa domanda di ricerca possono essere diverse; è perciò più utile mostrare le buone pratiche di analisi dei dati e di scrittura in python, piuttosto che fornire sequenze di istruzioni da seguire alla cieca.

Architettura

La sitemap si presenta nella seguente maniera:



Altre scelte progettuali

La piattaforma è stata pensata in lingua inglese, in linea con tutto il materiale scientifico prodotto nell'ambito della data science (e non solo). Sebbene si potrebbe pensare a un sito multilingue, le sezioni in pseudocodice e codice non potrebbero essere tradotte, in quanto il loro linguaggio è standard.

Gestione del progetto

Composizione del gruppo di lavoro

Per far crescere il progetto serve un team in cui siano disponibili le seguenti competenze:

- sviluppo web per la realizzazione e il mantenimento della piattaforma
- progettazione grafica e illustrazione per ideare l'interfaccia, l'immagine coordinata e per produrre il materiale visivo occorrente a ogni sezione del sito
- padronanza delle nozioni teoriche legate al machine learning, all'analisi dati e all'intelligenza artificiale
- abilità nello scrivere codice python elegante e comprensibile.
- padronanza della lingua inglese

Si paga?

Sebbene possa sembrare un'utopia, l'idea sarebbe di rilasciare gratuitamente i contenuti, coerentemente con la missione di promozione della data literacy. Essendo l'argomento di interesse attuale, l'ideale sarebbe iniziare con un fondo pubblico (UE...) e/o di collaborare con un ente di ricerca (ISTI/CNR...).

Per poter aggiornare i contenuti e le modalità si potrebbe tuttavia pensare a un'iscrizione pro che permetta di accedere ad altri contenuti (algoritmi di nicchia...). Un'altra forma di finanziamento potrebbe derivare dalle Università, che potrebbero chiedere materiale custom per i corsi offerti (ad esempio dei generatori di esercizi o delle visualizzazioni più tecniche).

Conclusioni

Gli aspetti su cui si è maggiormente riflettuto, durante l'ideazione e la prototipazione del progetto, sono l'accessibilità delle informazioni e l'esperienza utente. Sebbene il web offra numerose piattaforme di long-life-learning, alcune molto valide, si è avuta la percezione che spesso manchi una ricerca su come rendere l'apprendimento un'esperienza godibile e meno faticosa. Migliorare questo aspetto significa fidelizzare i visitatori e ridurre la soglia di abbandono.

Bisogna poi ricordare che il digitale offre limiti e possibilità diversi da quelli di un libro o di una lezione in presenza. Per rendere efficace uno strumento occorre studiare questi aspetti e comprendere su quali modalità di fruizione indirizzare il progetto. Se si osserva un'alta la soglia di abbandono nei corsi online, bisogna cercare di individuarne le cause.

In questo progetto, si è ipotizzato che il web sia una piattaforma troppo aperta e dispersiva per chiedere all'utente di spendere spontaneamente otto o più ore del proprio tempo su un corso monotematico, per quanto esso sia valido. La dinamica del giornale online, per contro, sembra essere più efficace: il tempo di permanenza è ridotto nell'arco della giornata, ma può essere costante nella settimana, o almeno durante il mese.

Per questo motivo si è evitato di progettare un percorso lineare, optando invece per un sistema che, come un'enciclopedia, può essere consultato a piacimento. Ciò è possibile anche grazie natura tecnica della data science: una volta acquisite le nozioni basilari, i singoli argomenti possono essere appresi seguendo percorsi differenti.

In conclusione, l'obiettivo di questo progetto è di portare il fruitore a visitare la piattaforma con costanza, ma senza che questo venga percepito come un obbligo. La comunicazione accessibile di concetti complessi dà al lettore la soddisfazione di comprendere nozioni che credeva inarrivabili e lo sprona a continuare a imparare.

