DATASET STRUTTURATI

Un dataset strutturato è un dataset in cui i dati contenuti presentano una certa organizzazione interna che semplifica la ricerca dei dati al loro interno.

* CSV: comma separated values, utilizza la ‘,’ per separare le features
* TSV: tab separated values, identico a csv, solo che al posto della ’,’ c’è il tab (4 spazi)
* JSON: Javascript Object Notation, molto diffuso nello sviluppo web. Usato molto nei DB non relazionali (MongoDB)
* XML: Extensible Markup Language, molto usato nella comunicazione client-server. Si basa sull’utilizzo di tag racchiusi fra ‘<>’ che possono essere definite a piacimento per identificare e organizzare le informazioni
* HTML: è anche esso un Markup Language però usato per lo sviluppo web, molto usato nella comunicazione client-server. A differenza dell’XML, i tag sono già definiti dallo standard del linguaggio.
* SQL: fornisce una serie di comandi per fare query su database.
* Excel: foglio di calcolo elettronico con struttura tabulare

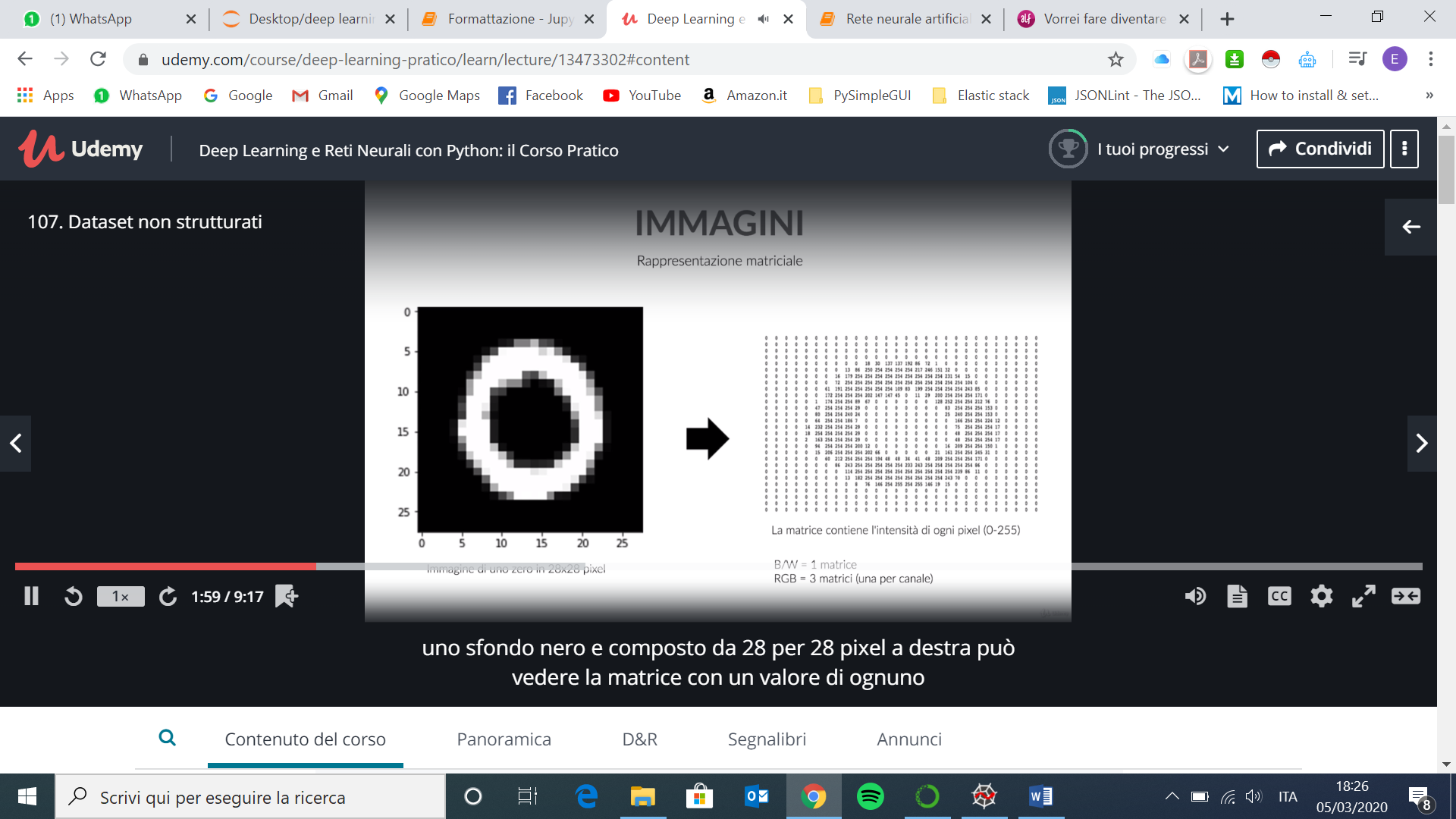
DATASET NON STRUTTURATI

Manca di struttura, e per poterlo utilizzare bisogna crearla. Esempi di dataset non strutturati sono:

* Immagini, testo, file audio ecc

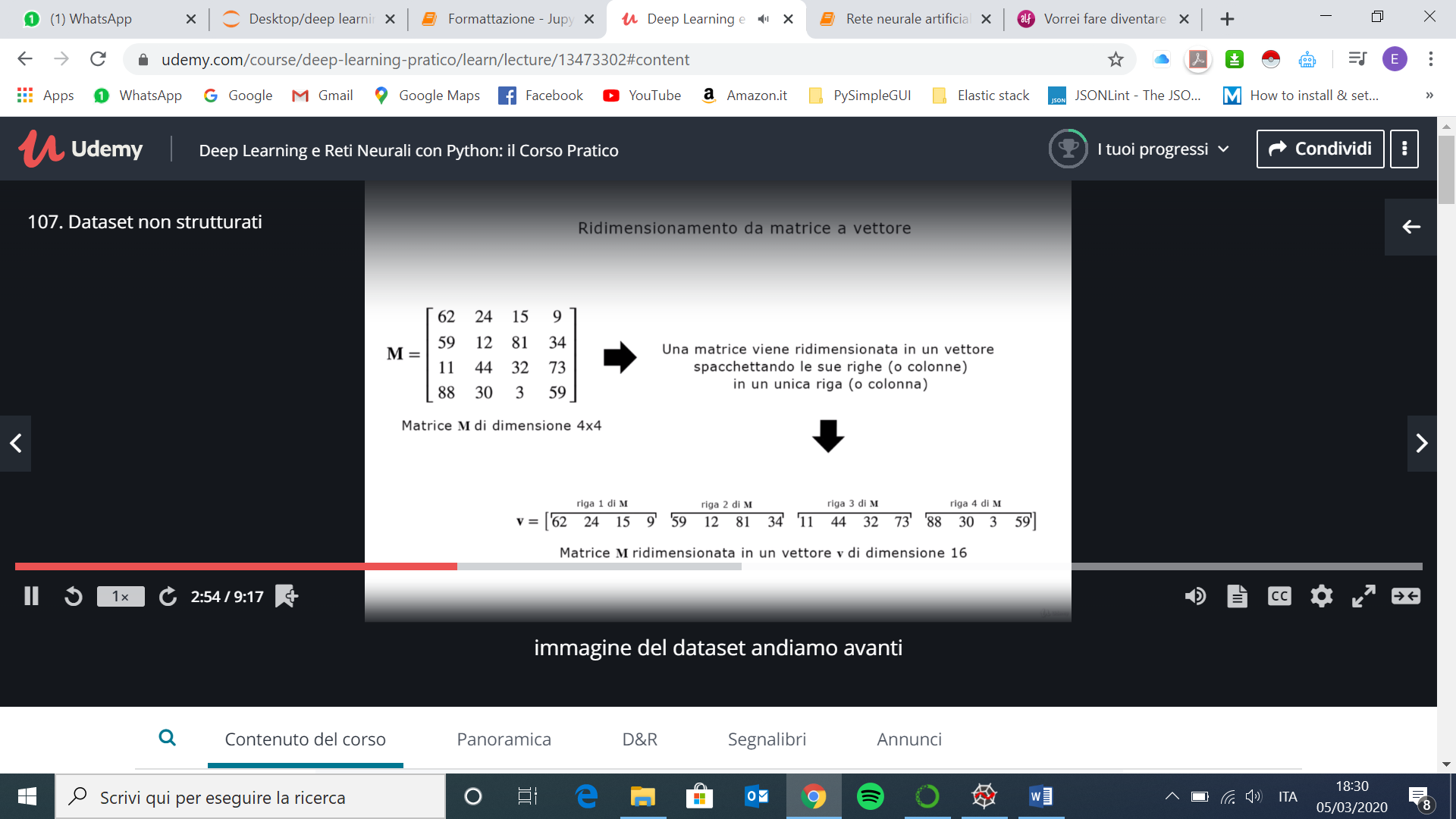
Ma nel machine learning l’input della macchina deve essere un vettore di numeri, quindi dobbiamo, a partire da dataset non strutturati, generare tale vettore.

Ad es un immagine può essere rappresentata da un insieme di pixel; nel caso di immagini a colori un singolo pixel possiede 3 valori che vanno da 0 a 255; 1 valore per l’intensità del rosso, uno per l’intens del verse e l’ultimo per l’intens del blu. Nel caso (più semplice) dell’immagine in bianco e nero, 1 pixel è rappresentato da un solo valore che va da 0 a 255 che rappresenta l’intensità del grigio (0: nero e 255:bianco).



L’immagine è composta da 28x28 pixel, che sarà quindi rappresentata da una matrice 28x28 dove ogni cella rappresenta un pixel, e il valore di tale cella rappresenta l’intensità del grigio.

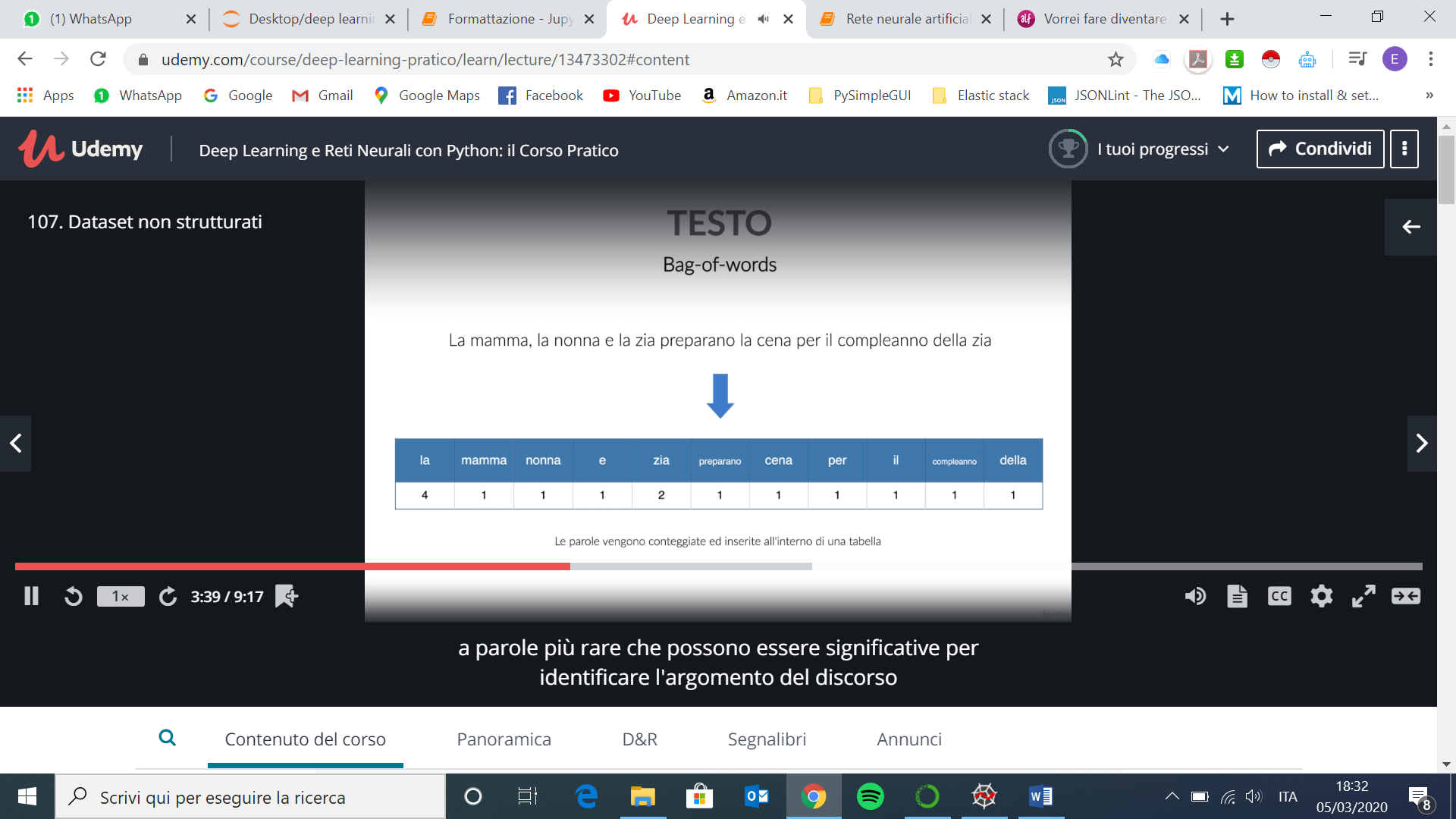
A questo punto dobbiamo passare dalla matrice al vettore, e si fa come descritto nella seguente figura.



Vediamo ora l’esempio del testo: questo è formato da caratteri e parole,e per essere utilizzato per istruire un modello di machinelearning è necessario codificare le parole in numeri,

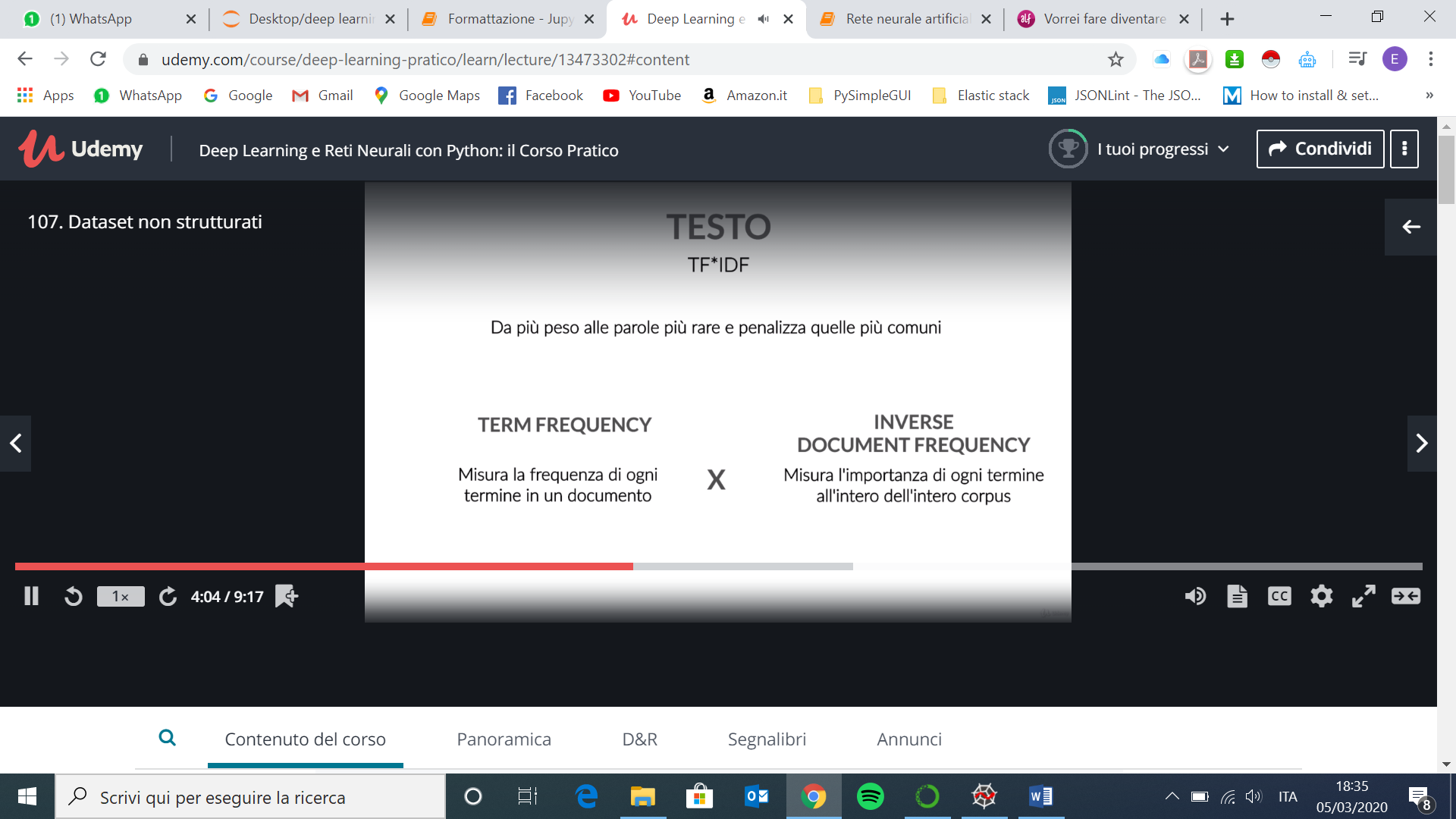
Ci sono diverse soluzioni per farlo:

* BAG OF WORDS: si fa come nell’immagine di seguito

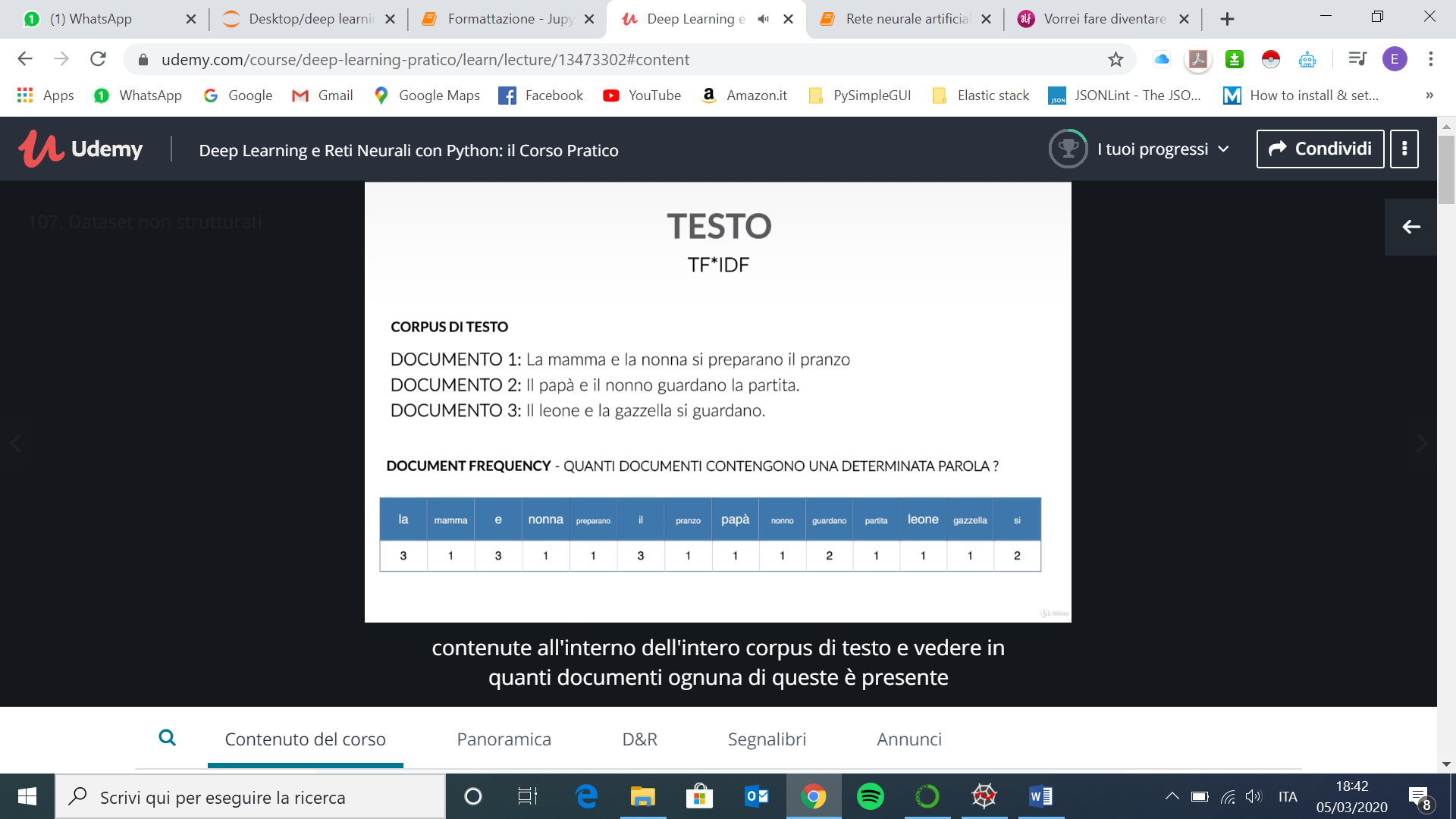
Questo modello però è spesso riduttivo perché favorisce le parole che appaiono spesso nel discorso (es. avverbi e congiunzioni) rispetto a parole che appaiono più di rado ma possono essere più significative.

Questo problema si risolve con l’altra soluzione:

* TF-IDF: da più peso alle parole più rare e penalizza quelle più comuni



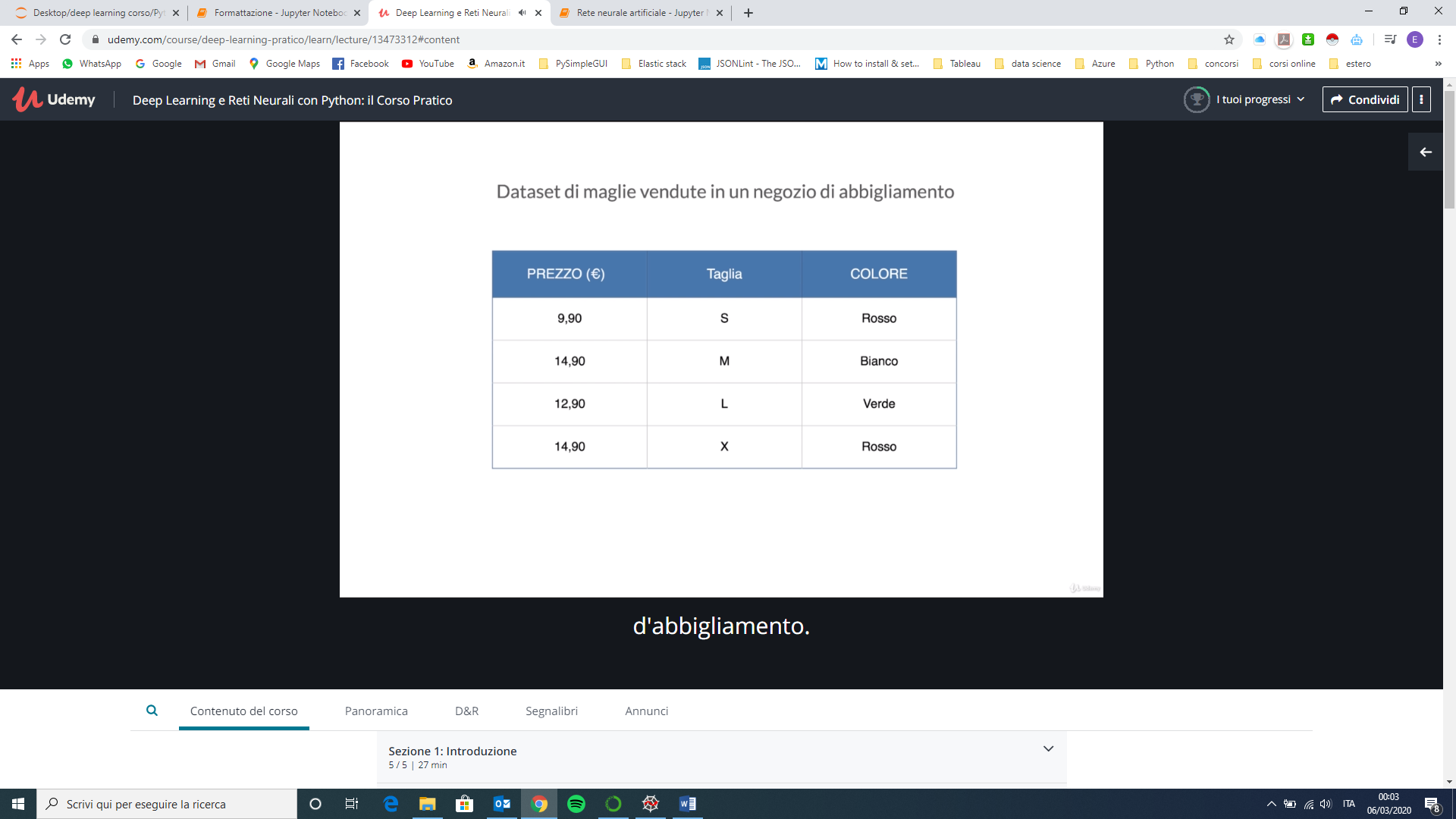
Esempio.





PREPROCESSING

Prendiamo ad esempio il seguente dataset



Contiene l’elenco delle magliette vendute in un negozio di abbigliamento.

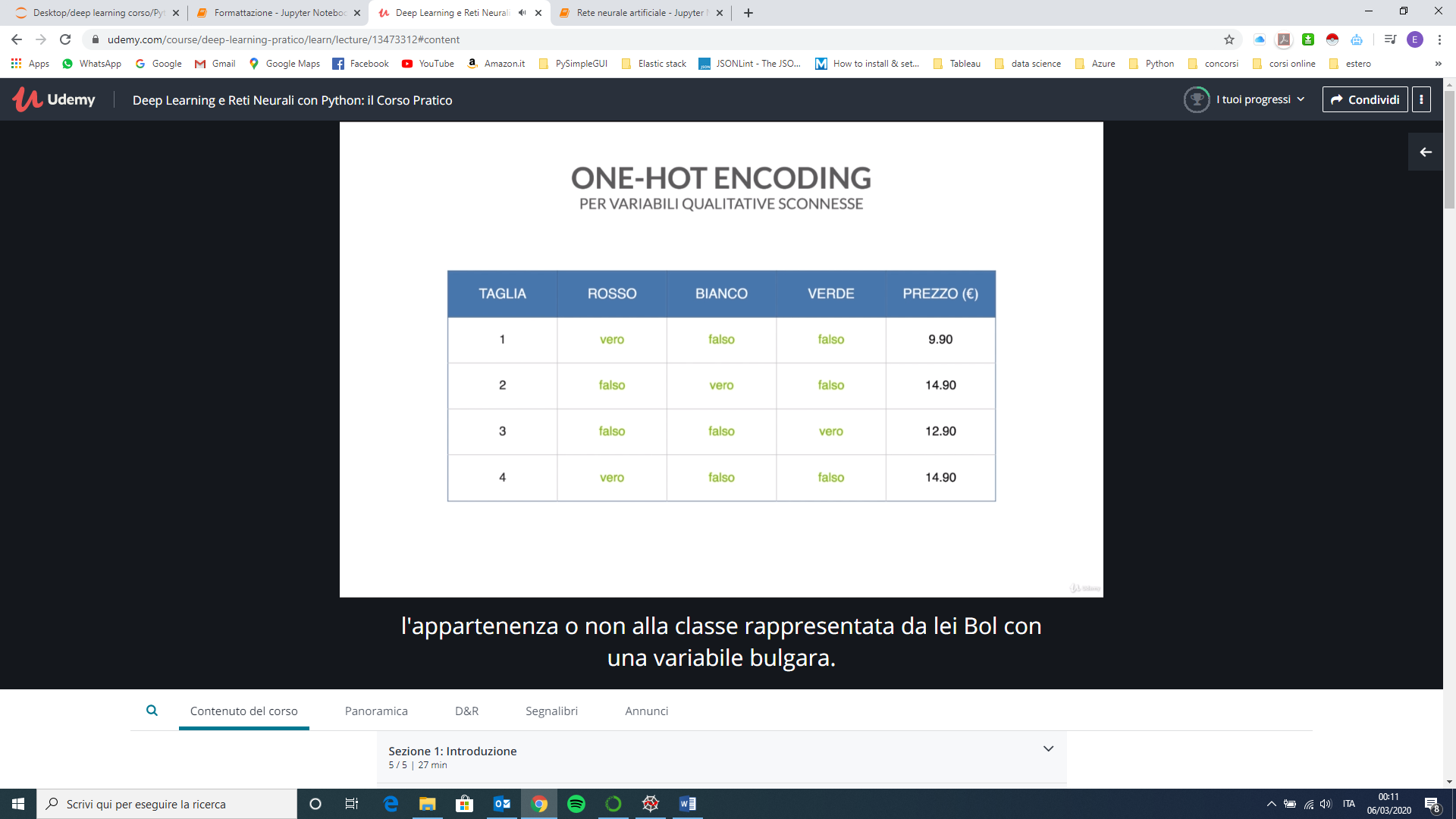
Le features sono 3:

* Prezzo: variabile quantitativa continua
* Taglia: variabile qualitativa ordinata (in quanto le “label” possono essere ordinate secondo un criterio) – VARIABILE CATEGORICA O QUALITATIVA 🡪 LIVELLO ORDINALE
* Colore: variabile qualitativa non ordinabile (le labels non possono essere ordinate secondo nessun criterio) – VARIABILE CATEGORICA O QUALITATIVA 🡪 LIVELLO NOMINALE

Ma un computer prende solo numeri ! Quindi dobbiamo codificare le variabili categoriche !

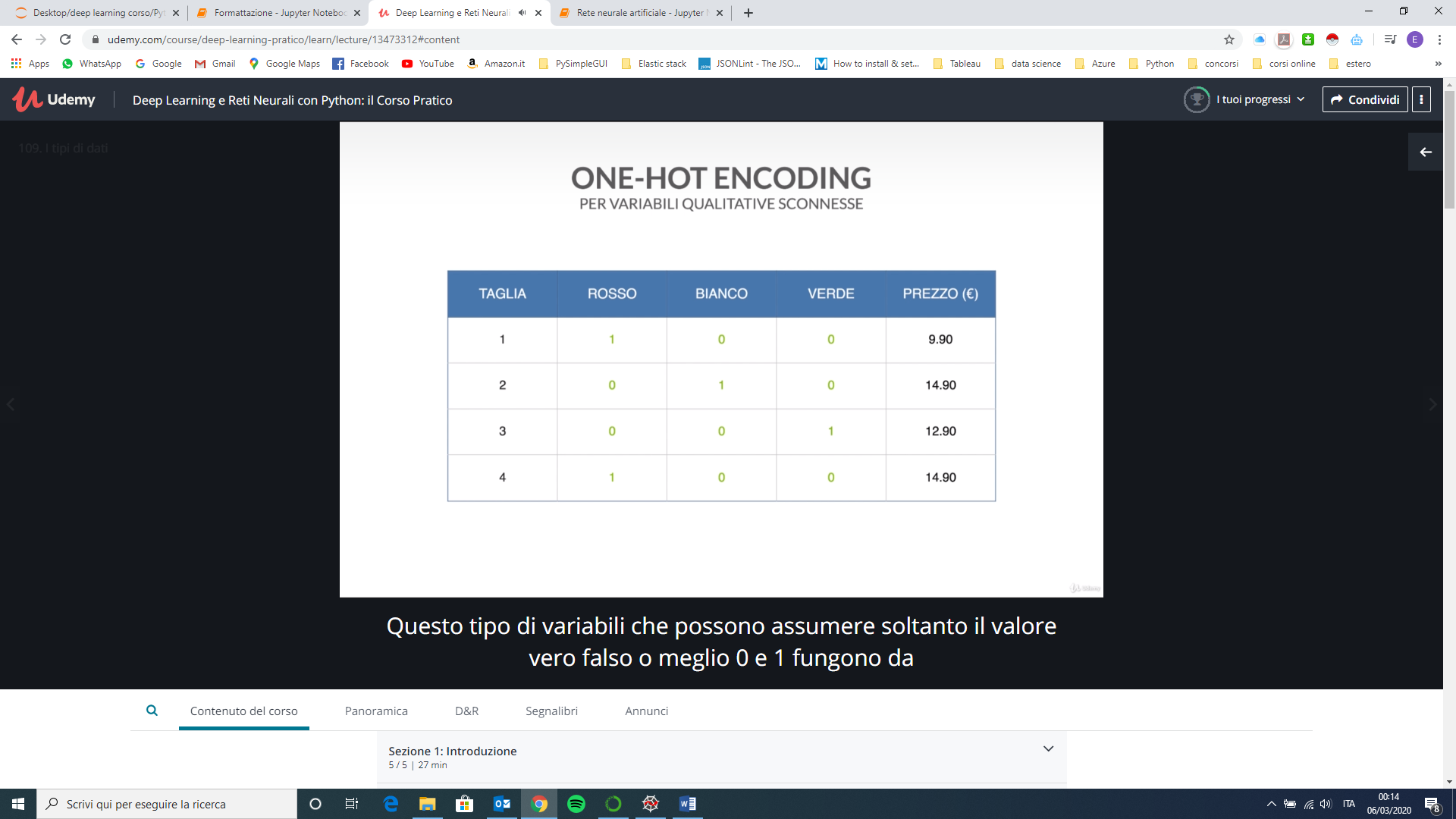
In merito alla variabile qualitativa ordinale possiamo procedere come indicato in figura.

Questo metodo non è applicabile per le variabili qualitative nominali, perché non sono ordinabili secondo nessun criterio. Utilizzare numeri assegnati a caso a ogni label potrebbe “confondere” la macchina; dunque si procede con quello che si chiama ONE-HOT ENCODING



Questo metodo consiste nel sostituire la feature con un numero di features pari alle labels della feature iniziale e assegnare valori booleani alle righe.

Come già sappiamo, True==1, False ==0, dunque:



Le variabili che possono assumere valore 0 o 1 (False o True) sono chiamate variabili DUMMY, ovvero variabili di comodo, proprio perché fungono da indicatori.

TRATTAMENTO DEI VALORI NULLI

Il trattamento dei valori nulli è un tema centrale nella data science ed esistono varie metodologie d’approccio a tale problematica.

1. Dropping di riga/colonna:

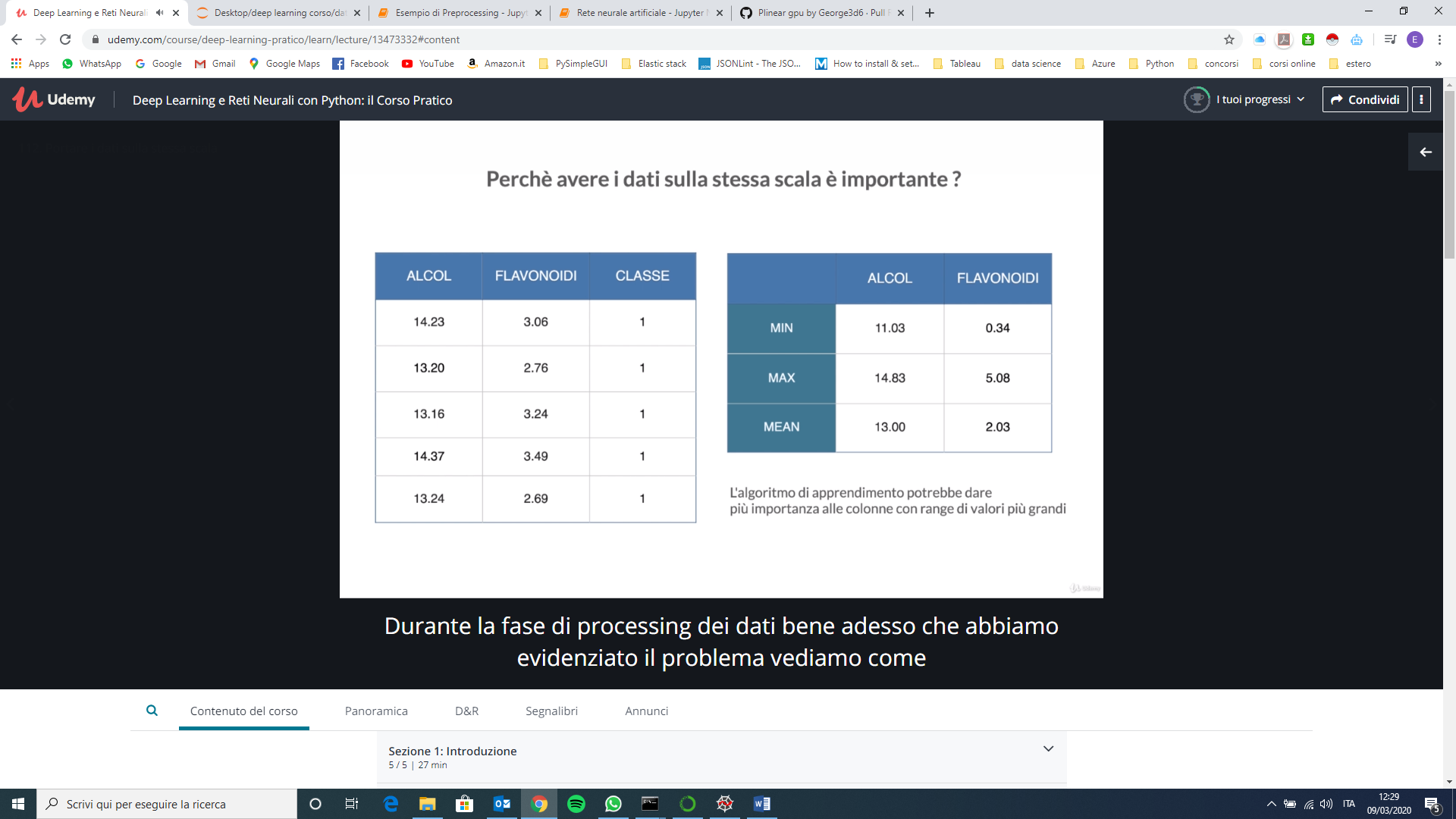
E’ l’approccio più banale, e consiste nel rimuovere quelle righe e/o colonne che hanno valori nulli. Sconsigliabile, in quanto c’è perdita/manomissione dell’informazione

1. Imputazione dei dati mancanti:

I valori nulli vengono in questo caso sostituiti con dei valori che abbiano un qualche forma di “rappresentatività” della colonna di appartenenza: un es. può essere la media o la mediana per le colonne quantitative, oppure la moda per le colonne qualitative

DATI SU STESSA SCALA

Di fondamentale importanza è, quando si ha un dataset, portare i valori appartenenti a diverse colonne su una stessa scala di valori. Ma perché ?



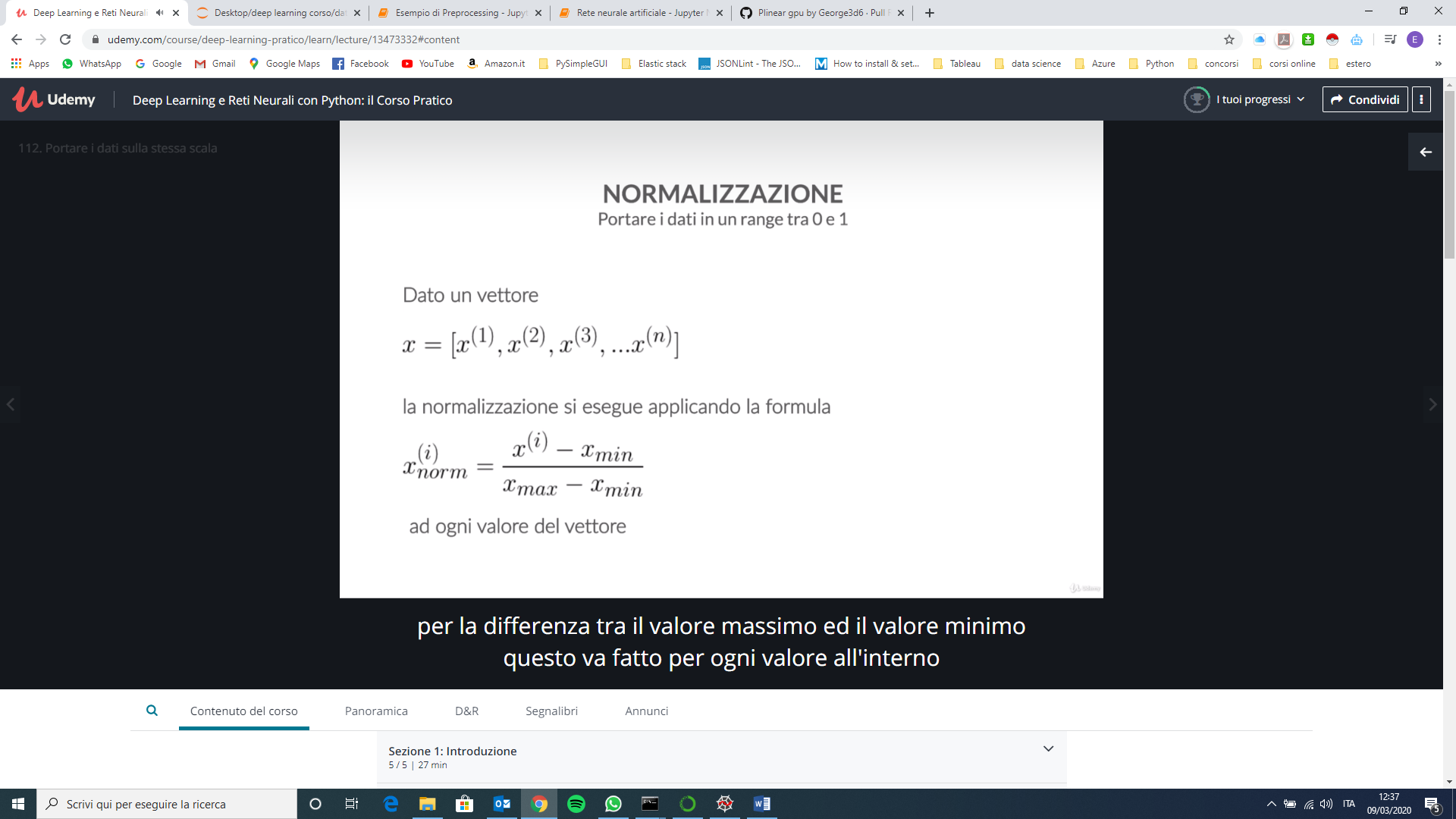
Nel nostro es abbiamo il dataset a sinistra. Come si può notare, nella colonna centrale abbiamo i valori su una scala diversa rispetto alla colonna di sinistra, e questo ci viene confermato dalla tabella a destra.

Il pericolo nel quale si incorre in questi casi è che l’algoritmo di apprendimento dia maggiore importanza alle colonne con valori più alti.

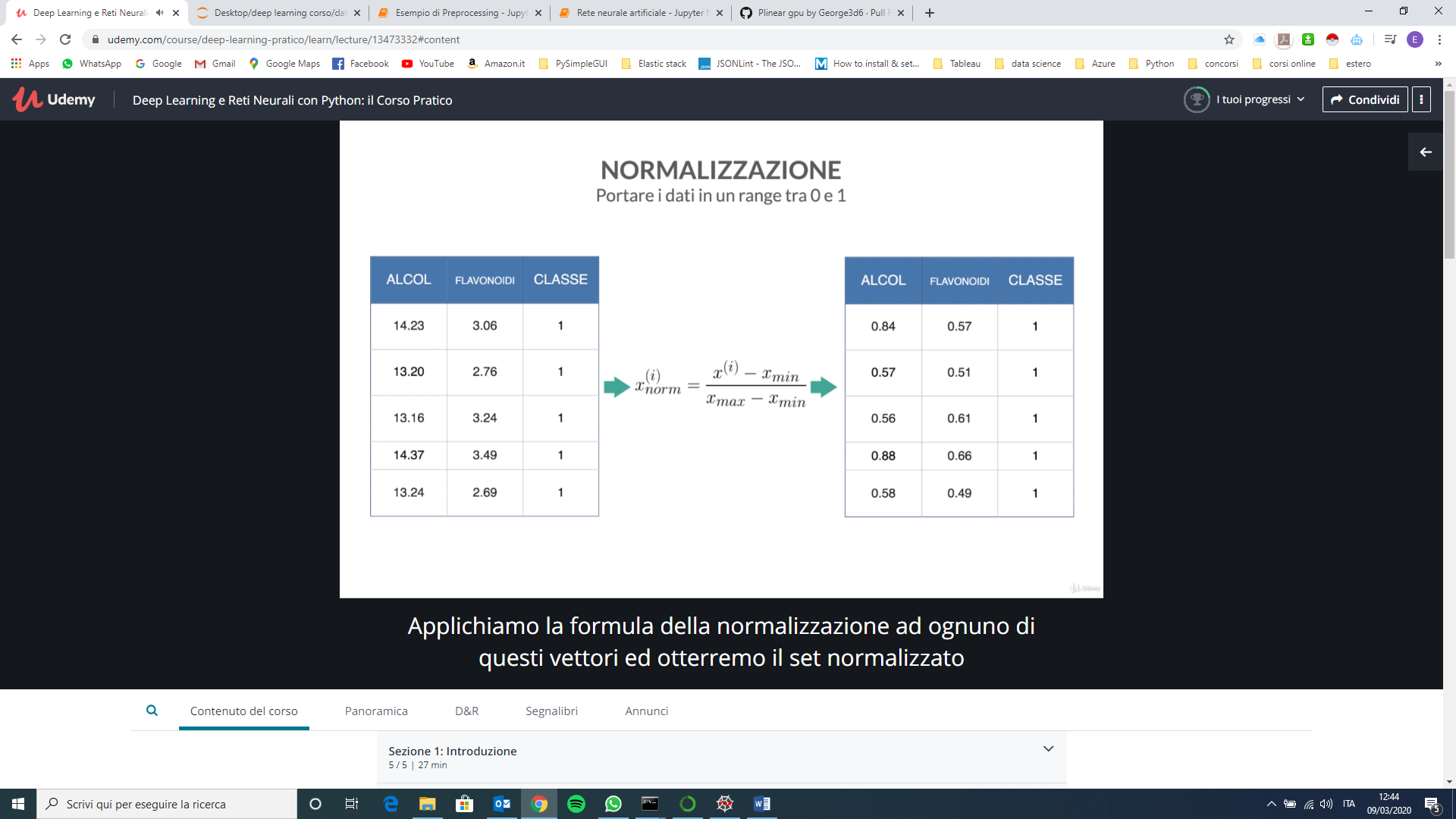
Questo è il motivo per cui si portano i dati di tutte le colonne su una stessa scala. Lo si può fare in 2 modi:

1. Normalizzazione

Porta i valori di tutte le colonne in un range compreso fra 0 e 1. Lo si fa applicando una semplice formula.



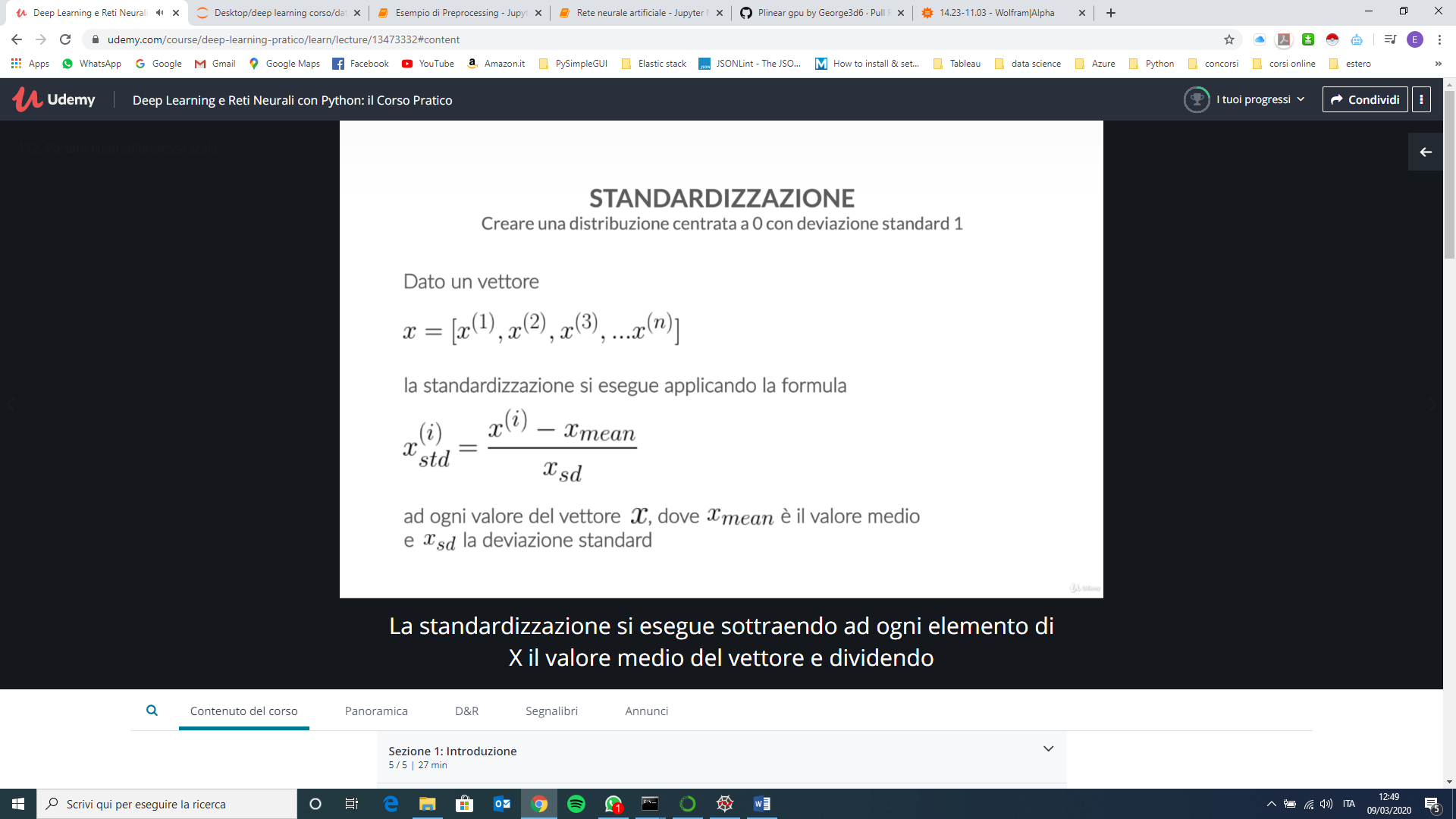
Il vettore rappresenta la colonna e i suoi elementi i valori. Dunque per ogni valore della singola colonna viene calcolato il corrispettivo normalizzato. Questo per ogni colonna.

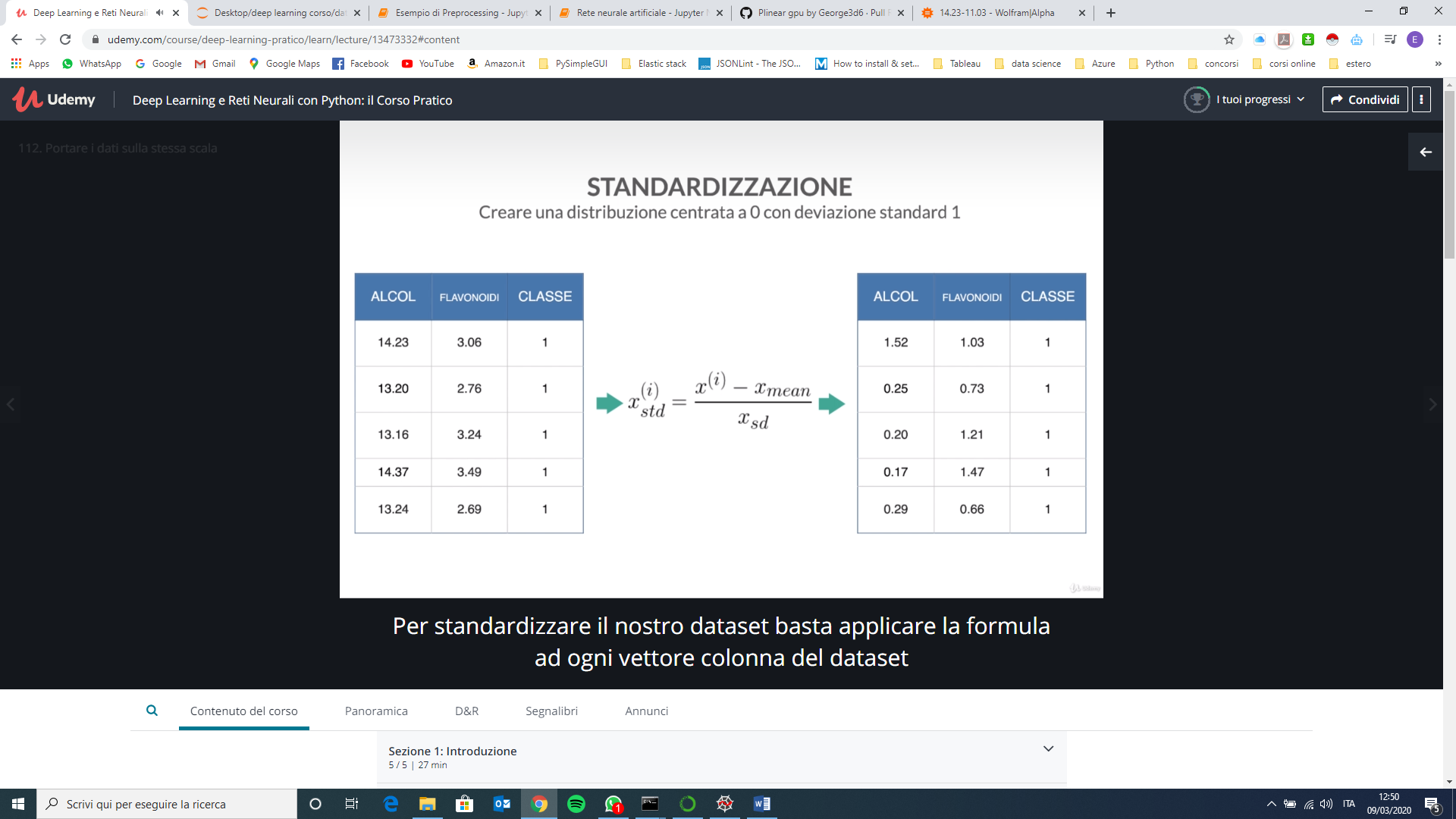


A sx abbiamo il dataset di partenza, a dx quello normalizzato.

1. Standardizzazione

Permette di trasformare il dataset in una distribuz normale, ovvero con media 0 e dev stand pari a 1.





Così otteniamo il nostro dataset standardizzato.



SPLITTING DEL DATASET

Abbiamo detto che nel machine learning (sia supervisionato che non), a differenza dei classici algoritmi, che sono scritti da persone per risolvere un dato problema, è la macchina che genera un algoritmo che risolve detto problema. Come fa?

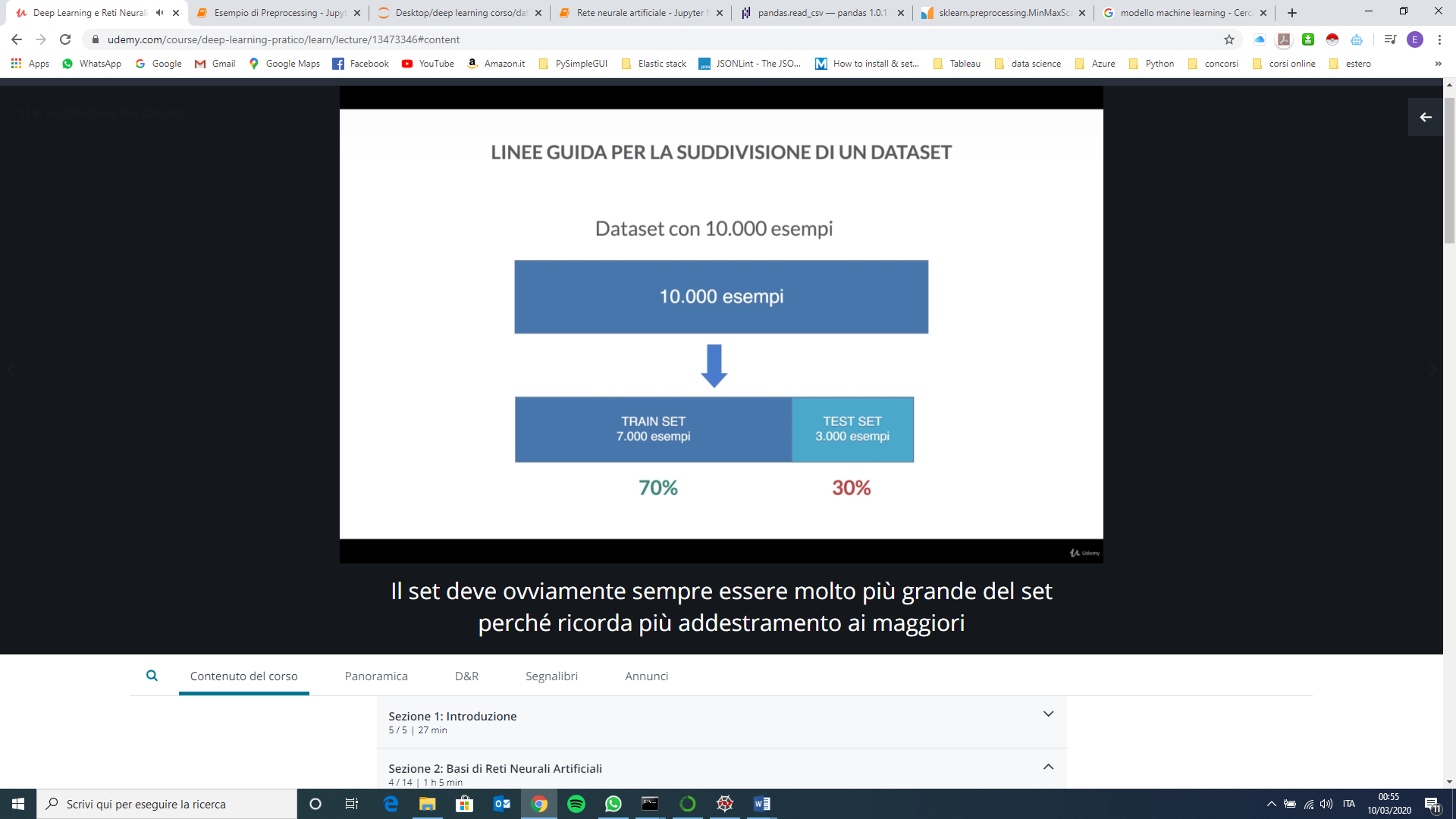
L’algoritmo generato dalla macchina si basa su un modello matematico. In pratica, scelto il modello matematico (dall’uomo), sarà la macchina, sulla base di una parte del dataset (training set), ad adattare (fitting) tale modello al caso in questione; in pratica, questo significa che sulla base degli esempi forniti (training set), la macchina calcolerà i parametri del modello tali da farlo funzionare (appunto adattarlo) al caso in questione.

Dopo, una volta addestrato il nostro modello sulla base del training set, ovvero aver concluso la fase di addestramento e generato i parametri del modello, il nostro modello addestrato viene testato sull’altra parte del dataset, ovvero il “test set”. Questo si fa per capire se il tutto funziona.

Si pensi a una semplice regressione lineare; il modello è y=ax+b, i parametri sono a e b. Il set di addestramento serve per stimare i parametri a e b. Una volta stimati, avremo nelle nostre mani un modello predittivo che può essere testato sul test set.

Lo stesso tipo di esempio può essere fatto anche nella classificazione.

Le proporzioni fra training set e test set in un dataset sono generalmente le seguenti



Con l’avvento dei big data invece le proporzioni sono un po’ cambiate

