



Università degli Studi di Milano Bicocca

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di laurea in Informatica

Sviluppo e validazione di uno strumento online per la diagnosi COVID-19 sulla base di parametri ematochimici

Relatore: *Prof. Federico Cabitza*

Co-relatore: *Dott. Andrea Campagner*

Relazione della prova finale di:

Giacomo Stoffa

Matricola 830159

Anno Accademico 2019-2020

Sommario

La seguente relazione descrive lo sviluppo e la validazione di una web app di machine learning creata mediante l'utilizzo di "Streamlit", un framework open-source che consente il rapido sviluppo di applicazioni in Python, utilizzato principalmente per esporre analisi e creazioni.

Lo scopo di questa web-app è quello pubblicare online un modello esistente in grado di fornire una predizione di positività o negatività al virus che causa COVID-19 basandosi su parametri ematochimici, con lo scopo di poter effettuare dei test in maniera meno dispendiosa, visti gli elevati costi dei reagenti per sottoporsi ad un tampone molecolare.

Indice

1 Introduzione

1.1 Obiettivo.....	2
1.2 Modello.....	3
1.2.1 Breve descrizione algoritmi importati.....	7
1.3 Cos'è Streamlit?	9

2 Sviluppo

2.1 Requisiti di Progetto	11
2.2 Librerie	14
2.3 Inserimento Valori	16
2.3.1 Lettura CSV.....	18
2.4 Output	22
2.5 Caricamento Online (Heroku)	25
2.5.1 Requirements	27

3 Conclusione

3.1 Miglioramenti	30
3.1.1 Velocità.....	30
3.1.2 Visualizzazione	31
3.2 Appendice tecnica.....	32
3.2.1 Modify Model_BloodTest.py.....	32
3.2.2 Checklist accesso (Heroku).....	34
Bibliografia	35

Capitolo 1

Introduzione

1.1 Obiettivo

La pandemia di COVID-19 del 2019-2020 è la pandemia attualmente in corso della cosiddetta “malattia del nuovo coronavirus”.

La prima segnalazione attribuibile a questo nuovo virus risale al 31 dicembre 2019, ma già l’8 dicembre sono comparsi i primi pazienti con malattia sintomatica.

Il 30 gennaio 2020 il virus SARS-CoV-2 è arrivato anche in Italia, a Roma, quando due turisti provenienti dalla Cina sono risultati positivi.

I soggetti sintomatici, per accertarsi della presenza o meno di tale virus vengono sottoposti a un tampone, di norma eseguito a livello nasofaringeo, che può essere di due tipi: molecolare per la ricerca di RNA (materiale genetico) virale, o tampone antigenico per la ricerca di elementi di superficie del virus. Essendo un test rapido, quindi meno preciso, l’eventuale positività ad un test antigenico, deve sempre essere confermata con un test molecolare.

Ma quanto costa, e perché, ciascun tampone?

Parliamo di circa 9 milioni di euro al giorno in totale, cifra nemmeno altissima vista l’assoluta importanza di questi test per cercare di arginare la curva del contagio, ma con cui le casse statali e della sanità, in particolare, devono fare i conti. I test lenti gravano sul sistema sanitario tra i 35 e gli 89 euro ciascuno e, nel caso si eseguissero in ambulatori privati la tariffa media è prossima ai 100 euro. A incidere su tale prezzo sono diversi fattori da tenere in considerazione, come i costi del kit monouso, del contenitore dove riporlo e, soprattutto, del reagente necessario per effettuare il test a livello molecolare. Ad oggi, però, il numero di tamponi risulta essere comunque insufficiente a tracciare con precisione la circolazione dell’infezione, ma segna anche uno sforzo non da poco per il nostro sistema sanitario.

Inoltre, oltre al tempo stimato di attesa di 2-3 giorni, la percentuale di ricevere un valore falso negativo si aggira intorno al 15%.

Da qui nasce l'idea di pubblicare online un modello predittivo basato sull'inserimento di parametri ematochimici, in modo tale da poter ricevere una stima in percentuale del tasso di positività, fornendo dati acquisibili facilmente sottoponendosi ad un esame del sangue. Un esempio, l'esame emocromo, abbreviazione di emocromocitometrico, ovvero un esame di laboratorio completo del sangue ha un costo che si aggira intorno ai 5 euro.

1.2 Modello

Sono stati importati diversi modelli di classificazione applicando tecniche di machine learning ai risultati delle analisi del sangue.

La definizione più nota di Machine Learning è quella proposta da Tom Mitchell, professore al Carnegie Mellon University (CMU):

“Si dice che un programma apprende dall'esperienza E , con riferimento ad alcune classi di compiti T e con misurazione della performance P , se le sue performance nel compito T , come misurato da P , migliorano con l'esperienza E ”

In parole più semplici, con il termine machine learning si fa riferimento alla capacità di una macchina di apprendere automaticamente, permettere ai computer di imparare dall'esperienza migliorando le prestazioni del programma, in questo caso riuscire ad ottenere una stima in percentuale il più possibile veritiera, basandosi su dati e parametri preesistenti.

È stato utilizzato un set di 1624 casi (di cui 786 erano casi SARS-CoV-2 positivi).

Nello specifico, sono stati importati sulla web app 2 principali modelli:

- Un modello che utilizza solo l'emocromo completo (dataset CBC), costituito da 21 parametri. (**Table 1**)

- Un modello che utilizza un set di 34 parametri, set dati specifico di COVID, costituito da alcuni parametri che si sono dimostrati marcatamente alterati nei casi di pazienti covid-19. (**Table 2**)

Parameter	Acronym	Unit of Measure
Sex	Gender	Male/Female
Age	Age	Years
Hematocrit	HCT	%
Hemoglobin	HGB	g/dL
Mean Corpuscular Hemoglobin	MCH	pg/Cell
Mean Corpuscular Hemoglobin Concentration	MCHC	g Hb/dL
Average Globular Volume	MCV	fL
Red Blood Cells	RBC	$10^{12}/L$
White Blood Cells	WBC	$10^9/L$
Platelets	PLT1	$10^9/L$
Neutrophils Count	NE	%
Lymphocytes Count	LY	%
Monocytes Count	MO	%
Eosinophils Count	EO	%
Basophils Count	BA	%
Neutrophils Count	NET	$10^9/L$
Lymphocytes Count	LYT	$10^9/L$
Monocytes Count	MOT	$10^9/L$
Eosinophils Count	EOT	$10^9/L$
Basophils Count	BAT	$10^9/L$
Presence of COVID-19 Symptoms	Suspect	True/False

.Table 1. Complete list of the analyzed features in the CBC dataset

Parameter	Acronym	Unit of Measure
Sex	Gender	Male/Female
Age	Age	Years
Calcium	CA	mmol/L
Creatine Kinase	CK	U/L
Creatine	CREA	mg/dL
Alkaline Phosphatase	ALP	U/L
Gamma Glutamyltransferase	GGT	U/L
Glucose	GLU	mg/dL
Aspartate Aminotransferase	AST	U/L
Alanine Aminotransferase	ALT	U/L
Lactate Dehydrogenase	LDH	U/L
Polymerase Chain Reaction	PCR	mg/dL
Potassium	KAL	mmol/L
Sodium	NAT	mmol/L
Urea	UREA	mg/dL
White Blood Cells	WBC	$10^9/L$
Red Blood Cells	RBC	$10^{12}/L$
Hemoglobin	HGB	g/dL
Hematocrit	HCT	%
Average Globular Volume	MCV	fL
Mean Corpuscular Hemoglobin	MCH	pg/Cell

.Table 2.1. Complete list of the analyzed features in the COVID-predictive dataset

Parameter	Acronym	Unit of Measure
Mean Corpuscular Hemoglobin Concentration	MCHC	g Hb/dL
Platelets	PLT1	$10^9/L$
Neutrophils Count	NE	%
Lymphocytes Count	LY	%
Monocytes Count	MO	%
Eosinophils Count	EO	%
Basophils Count	BA	%
Neutrophils Count	NET	$10^9/L$
Lymphocytes Count	LYT	$10^9/L$
Monocytes Count	MOT	$10^9/L$
Eosinophils Count	EOT	$10^9/L$
Basophils Count	BAT	$10^9/L$
Presence of COVID-19 Symptoms	Suspect	True/False

.Table 2.2. Complete list of the analyzed features in the COVID-predictive dataset

1.2.1 Breve descrizione algoritmi importati

Per quanto riguarda la classificazione, sono stati importati 6 diversi algoritmi per ogni dataset scelto (Covid-features o CBC).

- SVM (Support Vector Machine)

L'algoritmo Support Vector Machine è usato nel machine learning per risolvere problemi di classificazione e di regressione.

Sono dette macchine a vettori di supporto.

L'algoritmo è in grado di individuare un iperpiano che ha lo scopo di dividere il set di dati in più classi. I punti più lontani dall'iperpiano hanno maggiore probabilità di essere classificati correttamente dall'algoritmo.

La distanza tra i support vector e il confine decisionale è detto margin (margine).

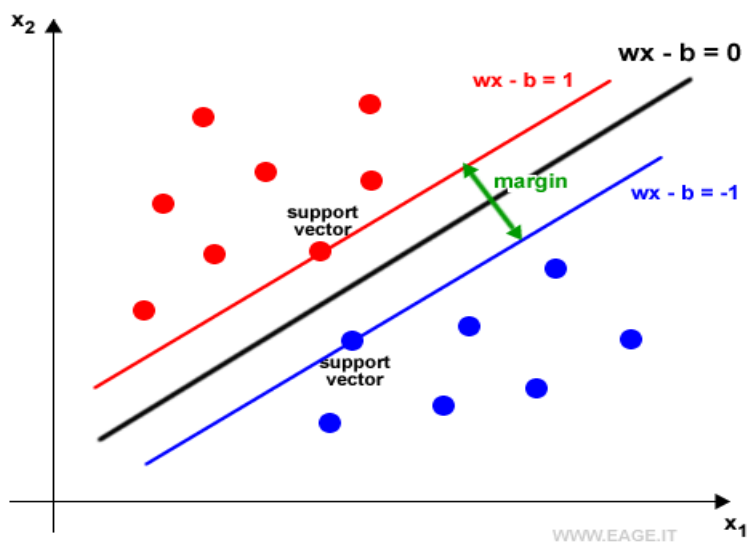


Figura 1.

- Random Forest

Questo algoritmo combina molti alberi decisionali in un unico modello. In modo tale da combinare assieme le previsioni fatte dagli alberi decisionali, che potrebbero non essere accurate individualmente, ma combinate assieme saranno in media più vicine al risultato.

Il risultato finale restituito dall'algoritmo, nel caso di problema di regressione, è la media del risultato numerico restituito dai diversi alberi e, nel caso di problema di classificazione, è la classe restituita dal maggior numero di alberi.

- KNN (k-nearest neighbors)

KNN è uno degli algoritmi più conosciuti e più semplici fra quelli utilizzati nell'apprendimento automatico.

È utilizzato nel riconoscimento di pattern per la classificazione di oggetti basandosi sulle caratteristiche degli oggetti vicini a quello considerato.

Il suo funzionamento si basa sulla somiglianza delle caratteristiche:

più un'istanza è vicina a un data point, più il knn li considererà simili.

Con data point si intende un insieme di una o più misurazioni su un singolo membro dell'unità di osservazione.

- Logistic Regression

Facile da implementare e utilizzabile come base per qualsiasi problema di classificazione binaria. È un modello di regressione non lineare il cui obiettivo è stabilire la probabilità con cui un'osservazione può generare uno o l'altro valore della variabile dipendente. Inoltre, può essere utilizzato per classificare le osservazioni, in base alle caratteristiche di queste, in due categorie.

- Naive bayes

Naive Bayes o classificatore bayesiano, è un algoritmo per risolvere problemi di classificazione e apprendimento automatico (machine learning) che utilizza il teorema di Bayes.

Il teorema di Bayes permette di calcolare per ogni istanza, la probabilità di appartenenza a una classe.

Come funziona l'algoritmo?

Divide il problema in alcune caratteristiche, costruisce una tavola delle frequenze per ogni caratteristica e trasforma le frequenze in probabilità.

A questo punto, calcola le probabilità semplici e condizionate.

L'algoritmo calcola la probabilità della decisione di giocare, oppure no, moltiplicando tra loro le probabilità condizionali, sostituisce i valori e normalizza i risultati.

- Ensemble

Ovvero l'insieme dei 5 modelli migliori.

A differenza dei sistemi più semplici, per formulare una previsione la macchina non estrapola una sola ipotesi ma molte.

1.3 Cos'è Streamlit?

Per creare la web app è stato utilizzato "Streamlit" ovvero una libreria open-source di Python (versione 3.6 o superiore), compatibile con molte librerie e framework diversi, che permette di creare e condividere applicazioni personalizzate per machine learning e data science.

È stata creata da un team di data scientist che si erano incontrati nel 2013 mentre lavoravano al progetto "Google X", è gratuito e molto diffuso, contando circa 200.000 applicazioni costruite dalla fine del 2019.

Streamlit gestisce in autonomia un server locale che espone sulla porta **8501** la web app la quale può essere avviata da un percorso locale o da un percorso remoto.

Le applicazioni create, sono script in linguaggio Python eseguite mediante un approccio top-down, durante l'esecuzione dello script il risultato viene stampato sulla pagina e riesegue il codice qualora dovessero verificarsi interazioni con la web app o con i widget presenti su di essa.

Le funzioni possono fare uso della cache per migliorare le prestazioni.

Capitolo 2

Sviluppo

2.1 Requisiti di Progetto

Lo scopo principale di questo progetto consiste nel “porting web”, ovvero la pubblicazione online di un modello di machine learning (già addestrato) per la diagnosi del COVID-19 sulla base di parametri ematochimici. Dovrà, quindi, essere possibili per l’utente poter scegliere sia la tipologia di dataset da usare (CBC o COVID-predictive), sia l’algoritmo di classificazione da usare (Ensemble, SVM, RF, KNN, LR, NB).

Dodici modi differenti di affrontare il calcolo.

Il risultato predittivo, una volta inseriti i valori nei rispettivi campi (non è necessaria la compilazione di tutti i campi), verrà mostrato mediante una tabella contenente i rispettivi valori in percentuale, di positività o negatività.

Una volta eseguito il ‘submit’ dei valori, oltre alla tabella, sarà possibile visualizzare un’infografica (creata da un altro stagista) con lo scopo di mostrare l’output in maniera più visuale ed esprimere la relativa affidabilità al valore predittivo in considerazione (**Figura 3**).

Inoltre, per permettere all’utente di risparmiare tempo con la compilazione manuale dei valori, o semplicemente per avere un numero maggiore di predizioni in contemporanea, sarà possibile importare un file .CSV contenente tante righe quanti sono i test da eseguire (**Figura 2**).

Un CSV (Coma Separated Values) è un file caratterizzato dalla presenza di dati tabulari sotto forma di testo, i valori di ciascuna cella sono separati da un delimitatore come, per esempio, una virgola o un punto e virgola.

Ogni linea di testo corrisponde a una linea della tabella e, ogni virgola corrisponde ad una separazione tra colonne, anche in questo caso sarà possibile lasciare un campo vuoto, non specificato.

If you want to upload more instances:

It is possible to change the dataset and model in the left sidebar.

You selected CBC, model: Ensemble

The columns in the CSV file must be exactly in the order of the form.

Values must be separated by a delimiter

You csv file must not have a header

Drop files here to upload
or
browse files

COMPUTE

.Figura 2. Upload file .CSV

SUBMIT

NEGATIVE: 82.71 %

Results:



****if you are on mobile, scroll to the right or use desktop site version, to see the entire graph.****

Notes:

The circle's diameter represents the confidence interval of the prediction score;
Its position along the horizontal dimension the value of the prediction score.
The smaller the circle the higher the algorithm's confidence on its output.
The closer to a response (pos/neg), the higher the model's confidence on that response.

OUTCOME:	CONFIDENCE:
POSITIVE	17.29%
NEGATIVE	82.71%

.Figura 3. Output

2.2 Librerie

Le principali librerie utilizzate sono Pandas e Numpy rispettivamente per la gestione dei dataframe e degli array impiegati per la lettura e controllo dei dati.

Pandas è una libreria software per la manipolazione e l'analisi dei dati, offre strutture dati e operazioni per manipolare tabelle numeriche e serie temporali, così come Numpy, che aggiunge supporto a grandi matrici e array multidimensionali, insieme a una vasta collezione di funzioni matematiche di alto livello per poter operare efficientemente su queste strutture dati.

Per importare i 12 diversi modelli (rispettivamente 6 per entrambi i dataset) è stata utilizzata la funzione “`joblib.load`” che permette di lavorare in modo efficiente su oggetti Python arbitrari contenenti dati di grandi dimensioni. Questa funzione è basata sul modello di serializzazione ‘pickle’ di python, il che significa che è possibile eseguire codice python arbitrario quando si carica un oggetto serializzato con `joblib.load`.

“Pickling” è il processo mediante il quale una gerarchia di oggetti Python viene convertita in un flusso di byte, “unpickling” è l'operazione inversa.

Parameters: **filename:** str, pathlib.Path, or file object.

The file object or path of the file from which to load the object

mmap_mode: {None, 'r+', 'r', 'w+', 'c'}, optional

If not None, the arrays are memory-mapped from the disk. This mode has no effect for compressed files. Note that in this case the reconstructed object might no longer match exactly the originally pickled object.

Returns: result: any Python object

The object stored in the file.

.Figura 4. Joblib.load

I valori di predizioni vengono ricevuti passando al modello il dataframe contenente i valori caricati dall'utente, utilizzando le funzioni "predict" e "predict_proba".

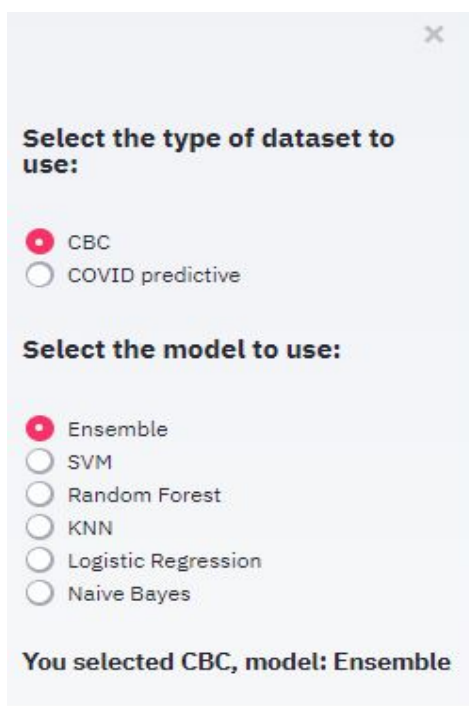
2.3 Inserimento Valori

La web app è accessibile tramite il seguente link:

<https://covid-19-blood-ml.herokuapp.com/>

Una volta aperta l'applicazione, è possibile notare che è presente una 'sidebar' a sinistra che permette la selezione del dataset e la scelta dell'algoritmo di classificazione da utilizzare (**Figura 5**).

Inizialmente impostata su dataset CBC model Ensemble.



The image shows a sidebar from a web application. At the top right is a close button (X). The first section is titled "Select the type of dataset to use:" and contains two radio button options: "CBC" (which is selected with a red dot) and "COVID predictive". The second section is titled "Select the model to use:" and contains six radio button options: "Ensemble" (selected with a red dot), "SVM", "Random Forest", "KNN", "Logistic Regression", and "Naive Bayes". At the bottom, a status message reads "You selected CBC, model: Ensemble".

.Figura 5.

Una volta scelto il dataset, nella schermata principale saranno presenti i campi vuoti, corrispondenti ai parametri del dataset (**Figura 6**), che l'utente potrà compilare inserendo i propri dati, e inviarli premendo il tasto "SUBMIT" al termine della lista.

A questo punto, con i seguenti valori inseriti (è possibile lasciare campi vuoti, non specificati) viene creato un Dataframe in modo tale da passare i valori al modello scelto, per poi utilizzare le funzioni citate precedentemente "predict" e "predict_proba".

ML-based COVID-19 Test from routine blood test

User Input Parameters:

Select Model and Dataset in the left sidebar

Fill in the all the fields of the following form or upload your CSV file

Leave the field blank if any actual value is not available

GENDER

Do not specify

AGE

|

Press Enter to apply

CA: Calcium (mmol/L)

CK: Creatine Kinase (U/L)

CREA: Creatine (mg/dL)

ALP: Alkaline Phosphatase (U/L)

.Figura 6. Insert Values

Sono presenti dei controlli sui campi, in modo tale da non permettere inserimenti errati e non fare andare avanti il programma fino a quando non venga inserito un numero valido (**Figura 7**).

LYT: Lymphocytes count ($10^9/L$)

23.56

MOT: Monocytes count ($10^9/L$)

..

Please insert a valid Number (es syntax: xx.xx)

EOT: Eosinophils count ($10^9/L$)

abc

Please insert a valid Number (es syntax: xx.xx)

BAT: Basophils count ($10^9/L$)

100

SUSPECT: Presence of COVID-19 symptoms

Do not specify

SUBMIT

Check the parameters before continuing

.Figura 7. Check values

2.3.1 Lettura CSV

Come detto in precedenza, è possibile eseguire la compilazione in maniera automatica, inserendo dati mediante l'importazione di un file CSV, anche in questo caso sono presenti controlli sul file e non sarà possibile selezionarne un tipo differente (**Figura 8**).

Il CSV non deve contenere un'intestazione e la prima riga deve già essere composta da dati utili.

Il carattere delimitatore viene riconosciuto automaticamente, impostando il campo “sep” della funzione relativa alla lettura del file CSV a None, in quanto il sistema di analisi di Python è in grado di riconoscerlo autonomamente.

La lettura del file avviene mediante la funzione della libreria Pandas “read_csv”. A questo punto il programma procede con la compilazione automatica del Dataframe, controllando che ogni parametro corrisponda ad un numero, lo invia al modello e restituisce l’output.

Questa operazione viene ripetuta per ogni riga.

If you want to upload more instances:
It is possible to change the dataset and model in the left sidebar.

You selected COVID, model: Ensemble
The columns in the CSV file must be exactly in the order of the form.
Values must be separated by a delimiter
You csv file must not have a header

Drop files here to upload
or
browse files

COMPUTE

Please upload a valid .CSV file

.Figura 8. Check csv

Qualora venisse caricato un file CSV contenente un numero diverso di parametri relativo al dataset scelto (CBC 21 o COVID-predictive 34), il programma si comporta in 2 modi:

- Se contiene un numero di dati inferiore, i restanti valori vengono impostati automaticamente a “Null”;
- Se contiene un numero di dati superiore, l’importazione si blocca e viene segnalato un messaggio di errore (**Figura 9**).

If you want to upload more instances:

It is possible to change the dataset and model in the left sidebar.

You selected CBC, model: Ensemble

The columns in the CSV file must be exactly in the order of the form.

Values must be separated by a delimiter

You csv file must not have a header

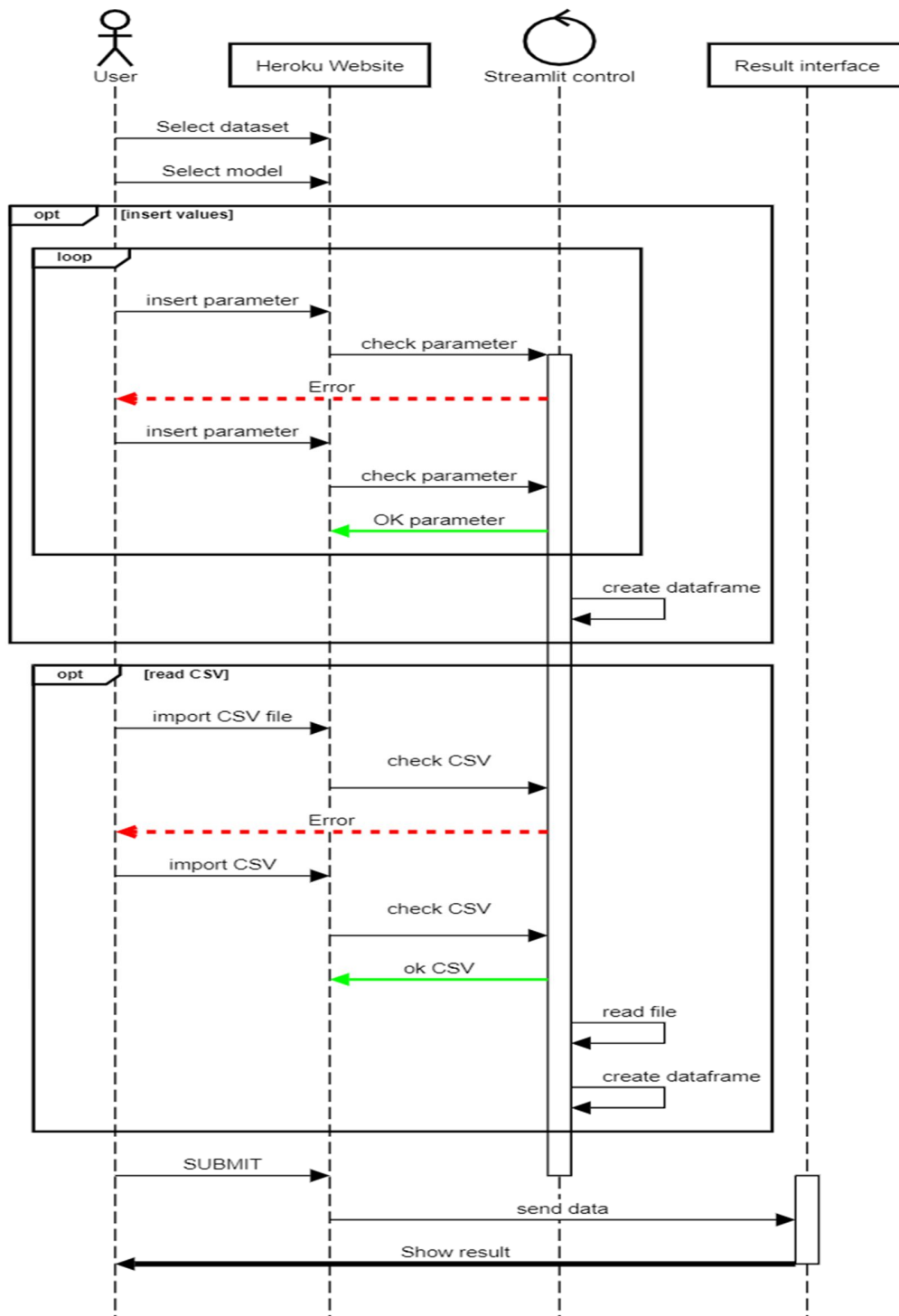
dataset_test.csv
[browse files](#)

COMPUTE

ID 1:

Error: The number of values in this file is greater than the selected template.
Change dataset selection or change file.

.Figura 9. Check csv



.Figura 10. Sequence Diagram relating to the use of the web app

2.4 Output

Come mostrato nelle pagine precedenti, una volta inseriti i propri valori ed eseguito il “SUBMIT” di tali, la predizione viene mostrata mediante due output differenti: Grafico e tabella (vedi *Fig 3* pag. 13).

Per quanto riguarda l’infografica (**Figura 11**), il codice (HTML) non è stato creato dal sottoscritto poiché, in questo caso, lo scopo del lavoro consisteva esclusivamente con l’importazione di questa visualizzazione nella schermata di output.

NEGATIVE: 79.59 %

Results:



.Figura 11.

Il diametro del cerchio rappresenta l’intervallo di confidenza del valore di previsione e la sua posizione lungo la dimensione orizzontale rappresenta il valore del punteggio di previsione.

Più è piccolo il cerchio, maggiore è la fiducia dell’algoritmo sul suo output.

Più si avvicina ad una risposta (Pos/Neg), maggiore sarà la fiducia del modello in tale risposta.

Funzioni come `St.text`, `st.markdown` e `st.write` semplificano la scrittura di un testo in un app Streamlit ma, in questo caso, per importare il seguente codice HTML all'interno della web app scritta con linguaggio Python, è stato più conveniente utilizzare la funzione “`components.html`”.

Questa funzione (**Figura 12**) fornisce la possibilità di incorporare un ‘`iframe`’ all'interno dell'applicazione che contiene l'output desiderato.

```
streamlit.components.v1.html(html, width=None, height=None, scrolling=False)
```

Display an HTML string in an iframe.

Parameters

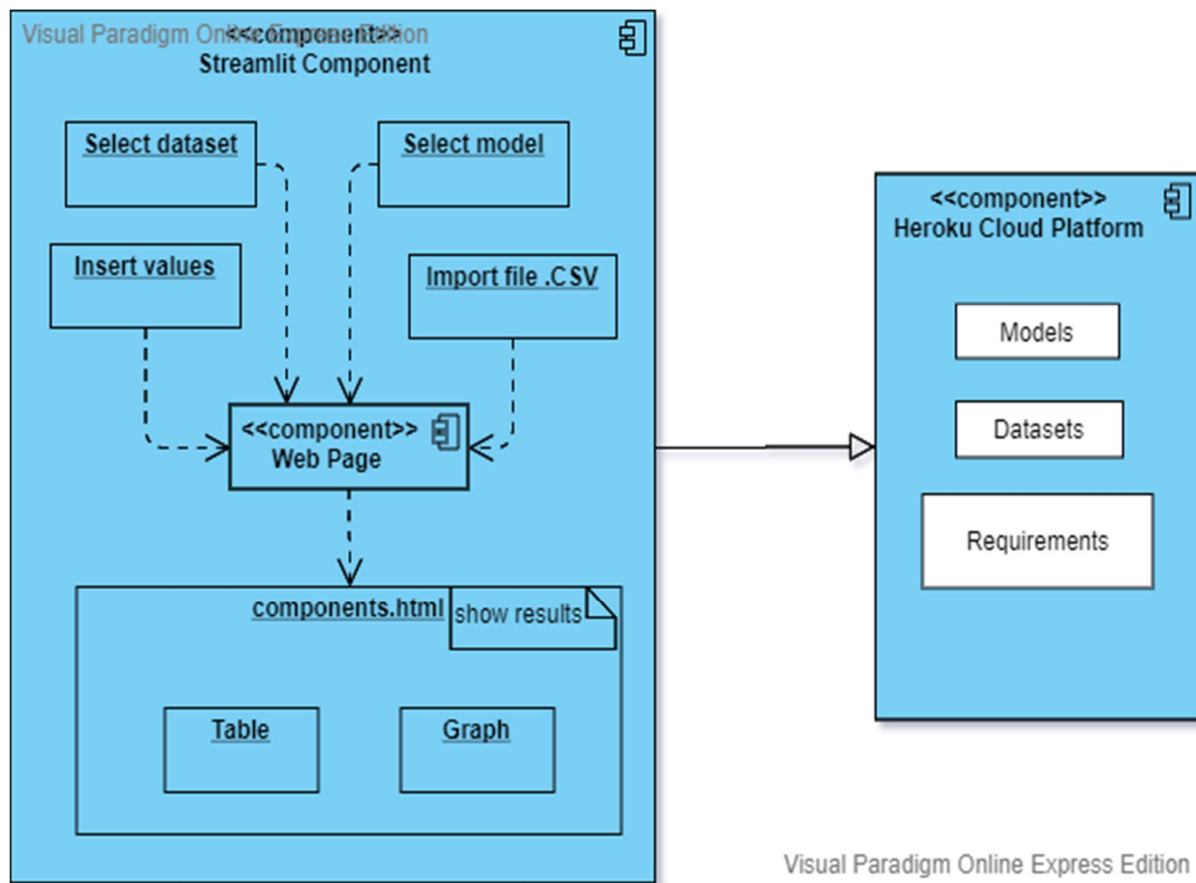
- **html** (*str*) – The HTML string to embed in the iframe.
- **width** (*int*) – The width of the frame in CSS pixels. Defaults to the report's default element width.
- **height** (*int*) – The height of the frame in CSS pixels. Defaults to 150.
- **scrolling** (*bool*) – If True, show a scrollbar when the content is larger than the iframe. Otherwise, do not show a scrollbar. Defaults to False.

.Figura 12. Graphic

È stata, inoltre, inserita una tabella (**Figura 13**) contenente valori in percentuale per quanto riguarda la prediction “Positive” o “Negative”.

OUTCOME:	CONFIDENCE:
POSITIVE	17.29%
NEGATIVE	82.71%

.Figura 13. Table



.Figura 14. Component Diagram

2.5 Caricamento Online (Heroku)

La creazione della web app è terminata con l'inserimento di scritte con informazioni affinché l'applicazione risulti il più possibile usabile e chiara.

Una volta finita bisognava caricarla online, per questo scopo è stato scelto di utilizzare la piattaforma cloud per app Heroku.

Fondata nel 2007, questa piattaforma ha lo scopo di distribuire applicazioni online, supporta 6 diversi linguaggi di programmazione (Java, Node.js, Scala, Clojure, Python e PHP) ed è basata sul sistema operativo Debian.

Heroku necessita di poter misurare il consumo delle nostre future web application per poter capire quanto esse utilizzino le proprie risorse e questo lo fanno attraverso i “dyno”.

I “dyno” sono contenitori linux isolati in cui viene eseguita l'applicazione, la piattaforma offre diversi tipi di “dyno” (**Figura 15**) per fornire i migliori risultati e prestazioni per la nostra tipologia di applicazione.

Dyno Type	Sleeps	Professional Features	Memory (RAM)	CPU Share	Dedicated	Compute
free	yes	no	512MB	1x	no	1x-4x
hobby	no	no	512MB	1x	no	1x-4x
standard-1x	no	yes	512MB	1x	no	1x-4x
standard-2x	no	yes	1024MB	2x	no	4x-8x
performance-m	no	yes	2.5GB	100%	yes	11x
performance-l	no	yes	14GB	100%	yes	46x

.Figura 15. *dyno types*

È stata utilizzata la versione ‘free’ che scala il tempo di attività del dyno da un totale di ore fornite mensilmente (450) e alla configurazione web. Questa configurazione fa sì che i dyno ricevano traffico sulla porta 80 (http) e 443 (https) e permettono di andare in stand-by

qualora non ricevessero traffico per 30 minuti, funzionalità utile per permettere di risparmiare le ore totali disponibili mensilmente.

```
C:\Windows\System32\heroku>git add .

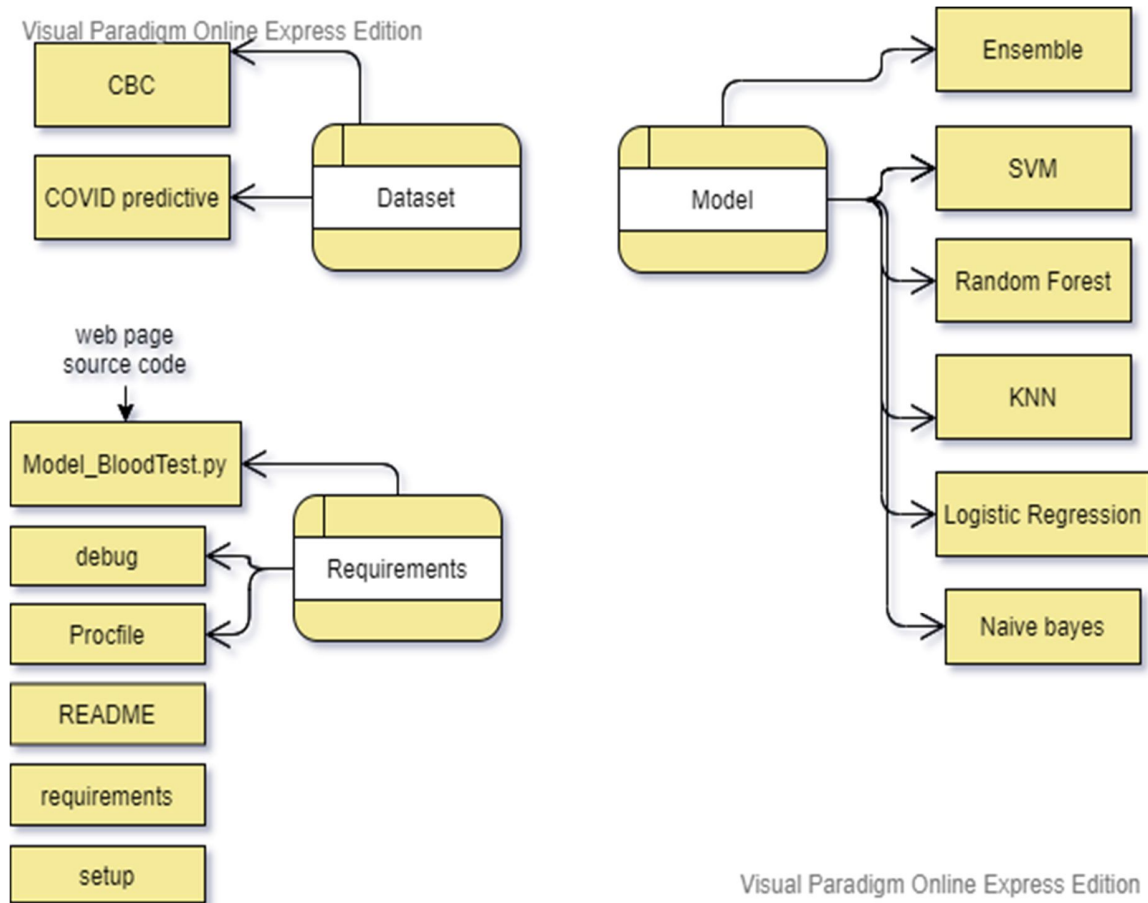
C:\Windows\System32\heroku>git commit -am "make it better"
[main cbe2771] make it better
1 file changed, 1 insertion(+)

C:\Windows\System32\heroku>git push heroku main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 316 bytes | 63.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Python app detected
remote: -----> No change in requirements detected, installing from cache
remote: -----> Installing pip 20.1.1, setuptools 47.1.1 and wheel 0.34.2
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote: -----> Discovering process types
remote: Procfile declares types -> web
remote:
remote: -----> Compressing...
remote: Done: 259.2M
remote: -----> Launching...
remote: Released v31
remote: https://covid-19-blood-ml.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/covid-19-blood-ml.git
669288f..cbe2771 main -> main

C:\Windows\System32\heroku>
```

.Figura 16. Comandi caricamento online

2.5.1 Requirements



.Figura 17. Entity Diagram

Questo PC > Windows (C:) > Windows > System32 > heroku

Nome	Ultima modifica	Tipo	Dimensione
debug	19/10/2020 15:43	Documento di testo	1 KB
ENS_fitted_CBC	12/08/2020 15:55	File JOBLIB	127.071 KB
ENS_fitted_COVID	12/08/2020 14:58	File JOBLIB	112.681 KB
KNN_fitted_CBC	12/08/2020 15:54	File JOBLIB	10.661 KB
KNN_fitted_COVID	12/08/2020 14:56	File JOBLIB	9.778 KB
LR_fitted_CBC	12/08/2020 15:54	File JOBLIB	11.228 KB
LR_fitted_COVID	12/08/2020 14:58	File JOBLIB	11.665 KB
Model_BloodTest	12/11/2020 12:58	File PY	41 KB
NB_fitted_CBC	12/08/2020 16:00	File JOBLIB	10.087 KB
NB_fitted_COVID	12/08/2020 14:59	File JOBLIB	9.224 KB
Procfile	01/10/2020 13:15	File	1 KB
README	15/09/2020 18:14	File MD	1 KB
requirements	12/10/2020 20:36	Documento di testo	1 KB
RF_fitted_CBC	12/08/2020 15:59	File JOBLIB	23.468 KB
RF_fitted_COVID	12/08/2020 14:57	File JOBLIB	18.477 KB
setup	26/06/2020 20:27	Shell Script	1 KB
SVM_fitted_CBC	12/08/2020 15:59	File JOBLIB	10.344 KB
SVM_fitted_COVID	12/08/2020 14:59	File JOBLIB	8.817 KB

.Figura 18. Root di progetto

Nella figura rappresentata sopra si possono notare tutti i file necessari per procedere correttamente con il “deploy” dell’applicazione Streamlit.

Sono presenti 12 file.joblib che sono rispettivamente i modelli relativi al dataset CBC e COVID, il codice principale di gestione della web app (Model_BloodTest.py) e i relativi file di impostazione tra cui:

Procfile strutturato nel seguente modo “web: streamlit run Model_BloodTest.py”, contenente il comando da eseguire per poter eseguire il codice principale.

Requirements.txt contiene tutte le versioni delle librerie/programmi utilizzati:

- streamlit==0.65.2;
- pandas==1.0.1;
- numpy==1.18.1;

- `scikit-learn==0.22.2.post1;`
- `matplotlib==3.1.3;`
- `openpyxl==3.0.3;`
- `xlrd==1.2.0;`
- `xlsxwriter==1.2.7.`

Quando si esegue il caricamento online su heroku, le dipendenze specificate in questo file vengono installate automaticamente prima dell'avvio dell'app.

Se questo file non fosse presente nella directory principale dell'applicazione Il 'buildpack' Python non sarebbe in grado di identificare correttamente l'applicazione.

Capitolo 3

Conclusione

3.1 Miglioramenti

Secondo il mio punto di vista, i principali miglioramenti legati a questa applicazione possono essere identificati in due macrocategorie principali: Velocità e visualizzazione

3.1.1 Velocità

Per quanto riguarda la velocità, avere una cartella di progetto da caricare di dimensioni elevate, contenente i modelli necessari, incide sul tempo di calcolo nell'eseguire operazioni.

Inoltre, anche la sezione di caricamento di un file CSV può impiegare diverso tempo nella lettura e nella creazione del dataframe.

Streamlit propone un meccanismo di gestione della cache che permette all'applicazione di rimanere performante anche durante il caricamento di dati dal web, la manipolazione di dataset di grandi dimensioni o l'esecuzione di calcoli impegnativi.

Questo viene fatto definendo le funzione con `@st.cache` che, ogni volta che vengono chiamate, controlla:

- i parametri di input;
- il valore delle variabili esterne usate all'interno della funzione;
- il corpo di tale;
- il corpo di qualsiasi funzione utilizzata all'interno della funzione memorizzata nella cache.

Questo fa sì che, se è la prima volta che Streamlit vede questi quattro componenti con questi valori esatti e in questa esatta combinazione, esegue la funzione e memorizza il risultato in una cache locale, altrimenti se nessun componente è cambiato salta l'esecuzione della funzione, restituendo l'output memorizzato nella cache precedentemente.

3.1.2 Visualizzazione

Un'altra modifica è relativa alla visualizzazione dei grafici da dispositivi mobile o di altro genere.

Il grafico mostrato in output (*Fig. 11 pag22*) scritto in HTML, è stato importato all'interno del programma scritto in linguaggio Python, utilizzando la funzione "components.html", citata nei capitoli precedenti. Questo permette una buona visualizzazione da PC, in quanto utilizzando la funzione `GetSystemMetrics` (**Figura 19**), è possibile ricavare i valori di risoluzione del monitor in modo tale da adattare la figura, ma non nel caso si visualizzasse il sito da smartphone o tablet.

In quest'ultimo caso non è possibile visualizzare il grafico interamente ma sarà necessario effettuare degli "scroll" per permettere la visualizzazione completa.

```
from win32api import GetSystemMetrics

print("Width =", GetSystemMetrics(0))
print("Height =", GetSystemMetrics(1))
```

.Figura 19.

Una possibile soluzione potrebbe essere quella di creare un componente bidirezionale:

Un frontend costituito da una parte HTML/Javascript che viene visualizzato nelle app Streamlit mediante un tag `iframe` e un API Python che le app Streamlit usano per istanziare e parlare con quel frontend.

3.2 Appendice tecnica

3.2.1 Modify Model BloodTest.py

	import streamlit as st	
	Code	Parameters
Display text	st.text("body")	body (str) - The string to display
	st.write(text multiple arguments)	
	st.write(""" #Text bigger """)	
	st.warning("message of warning")	
	st.info("info text")	
	st.markdown("string formatted as Markdown")	
	st.title('This is a title')	
	st.header("title")	
	st.subheader("subtitle")	
Display data	st.dataframe(data,width,height)	data(pandas.DataFrame,numpy.ndarray,Iterable,dict)- or None the data to display
	st.table(data)	
Input reader	selection=st.sidebar.radio("label", (options))	label(str)-A short label explaining to the user whats this select widget is for
	st.selectbox('label', (options))	options(list,tuple,numpy.ndarray,pd.series,pd.dataframe)
	st.text_input('name field', "initial text")	
	st.number_input(label, min, max, value, step)	
	st.slider(label, min, max, value, step)	display a slider widget

	Code	Parameters
Data	<code>data={'name': value, ...}</code>	
	<code>data=numpy.array(data)</code>	Index:Index or array-like. Index to use for resulting frame
	<code>data=pandas.DataFrame(data,index)</code>	
Model	<code>clf=joblib.load("Model Name")</code>	df-dataframe for the model
	<code>prediction=clf.predict(df)</code>	
	<code>prediction_proba=(clf.predict_proba(df))</code>	
File uploader	<code>file=st.file_uploader(label,type)</code>	label(str or none)- A short label explaining to the user what this file uploader is for
		type(str or none)-Array of allowed extension['csv','xls']. None means all extension are
read csv	<code>data=pd.read_csv(file,sep,delimiter,header)</code>	filepath_or_buffer: str,path object or file-like object
		sep:str,default',''. Delimiter to user. If None Python parsing engine can automatically detect.
		header:int. None ignore header

3.2.2 Checklist accesso (Heroku)

Eseguire il Log in al proprio account sulla piattaforma Heroku e selezionare tra le proprie applicazioni quella con il nome “covid-19-blood-ml” (**Figura 20**)

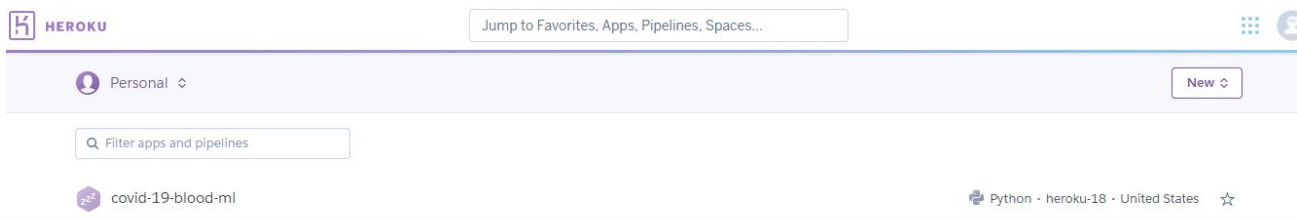


Figura 20.

Una volta acceduto, andando nella sezione dedicata al Deploy è possibile notare che esistono 3 principali metodi:



Figura 21.

Usando Heroku Git è necessario installare Heroku CLI e digitare i seguenti comandi:

- `$ heroku login`
- `$ heroku git:clone -a covid-19-blood-ml` (nome applicazione)
- `$ cd` root di progetto (Figura 18)
- `$ git add .`
- `$ git commit -am "make it better"`
- `$ git push heroku master`

Bibliografia

- [1] *API reference — Streamlit 0.71.0 documentation.* (n.d.). Streamlit Dcoumentation.
<https://docs.streamlit.io/en/stable/api.html>
- [2] Stoffa, G. (2020, October 20). *ML-based COVID-19 Test from routine blood test.* Covid 19 Blood ML. <https://covid-19-blood-ml.herokuapp.com/>
- [3] R. (2020, October 16). Quanto costano le analisi del sangue? Money.It.
<https://www.money.it/Quanto-costano-analisi-del-sangue#:~:text=Tra%20gli%20esami%20pi%C3%B9%20richiesti,aggira%20intorno%20ai%205%20euro.>
- [4] contributori di Wikipedia. (2020). *Pandemia di COVID-19 del 2019-2020.* Wikipedia.
https://it.wikipedia.org/wiki/Pandemia_di_COVID-19_del_2019-2020
- [5] ATS Insubria. (2020). *Tamponi.* <https://www.ats-insubria.it/aree-tematiche/covid-19/tamponi>
- [6] Dotti, G. (2020, October 22). *Quanto costano davvero i tamponi?* Wired.
https://www.wired.it/scienza/medicina/2020/10/22/costo-tamponi-molecolari-antigenici/?refresh_ce=
- [7] R. (2018, December 19). *Cos'è il Machine Learning, come funziona l'apprendimento automatico e quali sono le sue applicazioni.* AI4Business.
<https://www.ai4business.it/intelligenza-artificiale/machine-learning/machine-learning-cosa-e-applicazioni/>

- [8] Provino, A. (2020, June 4). *Primi passi con streamlit: Il Machine Learning Deployment rapido!* Machine Learning & Data Science Blog. <https://andreaprovino.it/primi-passi-con-streamlit/#:%7E:text=Iniziamo%20subito%20con%20il%20capire,le%20loro%20analisi%20e%20creazioni>
- [9] Eage. (n.d.). *Support Vector Machine*. <https://www.eage.it/machine-learning/support-vector-machine>
- [10] Govoni, L. G. (2019, August 18). *Come l'algoritmo Random Forest migliora le previsioni degli alberi decisionali*. Lorenzo Govoni. <https://lorenzogovoni.com/random-forest/>
- [11] contributori di Wikipedia. (2020, November 5). *K-nearest neighbors*. Wikipedia. https://it.wikipedia.org/wiki/K-nearest_neighbors
- [12] Govoni, L. (2020, June 7). *Algoritmo K-Nearest Neighbors (KNN)*. Lorenzo Govoni. <https://lorenzogovoni.com/knn/>
- [13] contributori di Wikipedia. (2019, November 29). *Modello logit*. Wikipedia. https://it.wikipedia.org/wiki/Modello_logit
- [14] Minini, A. (n.d.). *Algoritmo Naive Bayes*. Andrea Minini. <http://www.andreaminini.com/ai/machine-learning/algoritmo-naive-bayes>
- [15] S. (2020, August 28). *File CSV: Cos'è, come crearlo, convertirlo e importarlo*. Sendinblue. <https://it.sendinblue.com/blog/file-csv-cose/>
- [16] *pandas - Python Data Analysis Library*. (n.d.). Pandas. <https://pandas.pydata.org/about/index.html>
- [17] *Overview — NumPy v1.19 Manual*. (n.d.). Numpy. <https://numpy.org/doc/stable/>
- [18] Wikipedia. (2019, December 2). *NumPy*. Wikipedia. <https://it.wikipedia.org/wiki/NumPy>

- [19] *Cos'è e come funziona Heroku, piattaforma cloud per app*. (2014, October 31). Fastweb.
<https://www.fastweb.it/web-e-digital/cos-e-e-come-funziona-heroku-piattaforma-cloud-per-app/>
- [20] *Deploying Python and Django Apps on Heroku |Heroku Dev Center*. (n.d.). Heroku.
<https://devcenter.heroku.com/articles/deploying-python>
- [21] -. (n.d.). *pinvoke.net: getsystemmetrics (user32)*. Pinvoke.
<https://www.pinvoke.net/default.aspx/user32.getsystemmetrics>
- [22] *Components API reference — Streamlit 0.71.0 documentation*. (n.d.). Streamlit.
https://docs.streamlit.io/en/stable/develop_streamlit_components.html
- [23] *Improve app performance — Streamlit 0.71.0 documentation*. (n.d.). Streamlit.
<https://docs.streamlit.io/en/stable/caching.html>