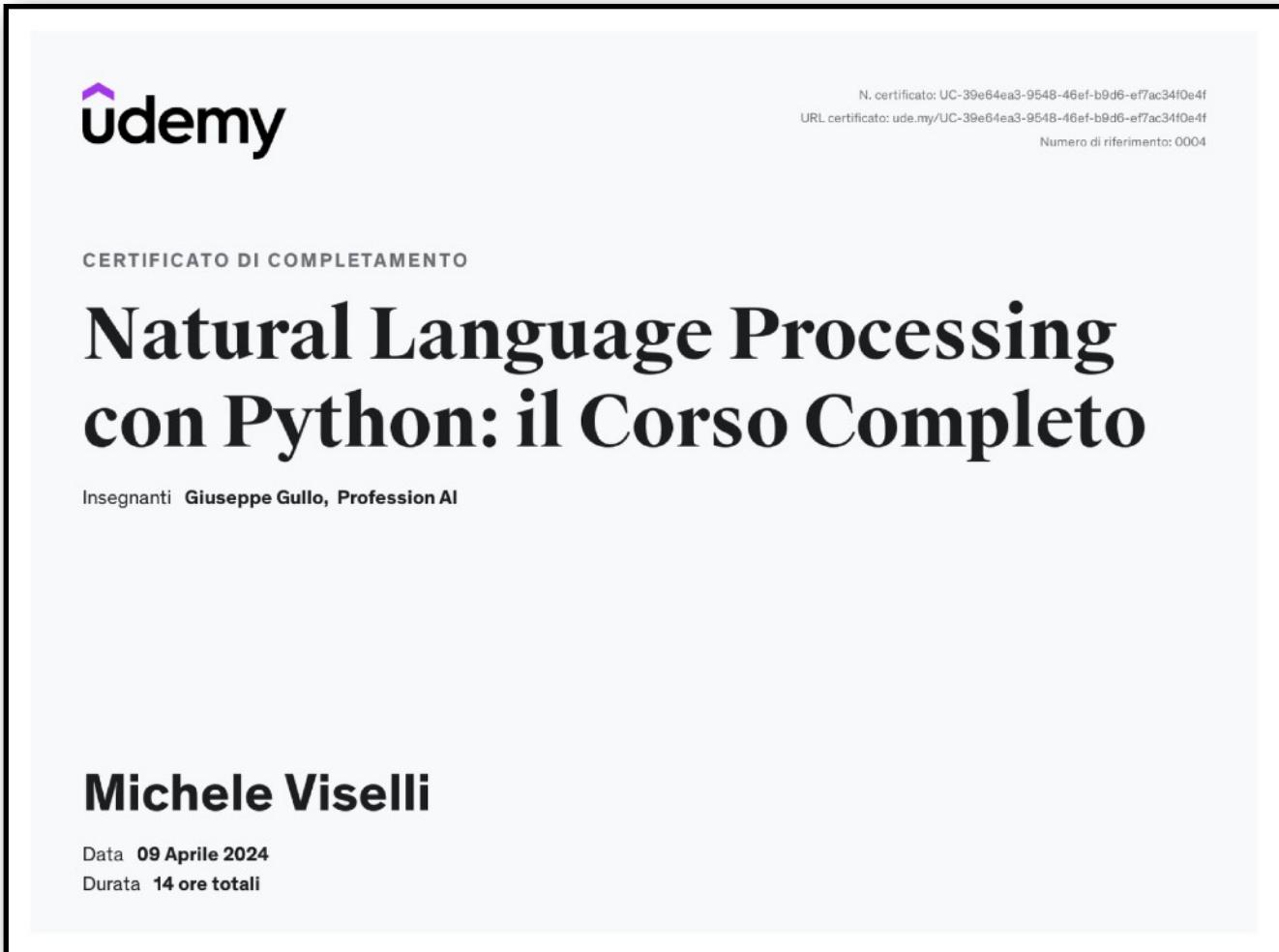


PROGETTO CLASSIFICATORE

Michele Viselli - 15 Aprile, 2024



Descrizione

Tra i vari progetti e le esperienze maturate in questo anno scolastico ho scelto di presentare un progetto di classificazione del testo dopo aver seguito un corso su Udemy, di ProfessionAI, sul Natural Language Processing (elaborazione del linguaggio naturale). Il progetto si compone in 3 modelli distinti:

"I modelli sono addestrati sulla lingua inglese a causa della scarsa quantità di dataset e librerie ottimizzate per la lingua italiana"

1. Classificatore del "sentiment": Questo modello cerca di capire se una frase è positiva o negativa utilizzando un dataset di recensioni di film (aclImdb). Dopo aver preprocessato i dati e creato un vettore di feature ("bag of words"), il modello viene addestrato con una regressione logistica. Infine, il modello viene testato su alcune recensioni di esempio. *Qui sotto riportato il risultato della classificazione:*

-> PROVA DEL MODELLO (Analisi del sentiment [positivo o negativo])

```
[9] # PRIMA RECENSIONE DA CLASSIFICARE (POSITIVA)
review = "This is the best movie I've ever seen"
prediction = lr.predict(bow.transform([review]))
if prediction[0] == 0: # se negativa modello riporta 0
    print("La recensione è negativa")
else: # altrimenti riporta 1
    print("La recensione è positiva")
```

La recensione è positiva

```
[10] # SECONDA RECENSIONE DA CLASSIFICARE (NEGATIVA)
review = "This is the worst movie I've ever seen"
prediction = lr.predict(bow.transform([review]))
if prediction[0] == 0:
    print("La recensione è negativa")
else:
    print("La recensione è positiva")
```

La recensione è negativa

-> il modello ha riconosciuto con successo quale recensione era positiva e quale era negativa

2. Classificatore di email spam: Questo modello classifica le email come spam o non spam (ham) utilizzando un dataset di email etichettate. Dopo la preparazione dei dati, il modello viene addestrato con una macchina a vettori di supporto (SVM). Il modello viene quindi testato su alcune email di esempio. *Qui sotto riportato il risultato:*

```
-> PROVA DEL MODELLO (Classificazione di una mail [spam o no])

▶ # PROVA SU UNA MAIL SPAM
email = ["URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! Txt the word: CLATM to No: 81010 T&C www.dbuk.net"]
feature_test = cv.transform(email)
result = model.predict(feature_test)
print(f"La mail è {result[0]}")

La mail è spam

[27] # PROVA SU UNA MAIL NON SPAM (HAM)
email = ["Please don't text me anymore. I have nothing else to say."]
feature_test = cv.transform(email)
result = model.predict(feature_test)
print(f"La mail è {result[0]}")

La mail è ham

-> il modello ha riconosciuto con successo la mail spam e la mail non spam (ham)
```

3. Riassunto di un testo: Questo modello prende come riferimento un articolo web (in questo caso: <https://www.sciencedaily.com/releases/2021/08/210811162816.htm>), lo analizza e ne produce un breve riassunto. Utilizza la libreria spaCy per la tokenizzazione e l'analisi del testo, identificando le frasi più rilevanti in base alla frequenza delle parole. *Qui sotto riportato il risultato:*

```
-> Prova del modello (Riassumere un testo)

▶ print(summary.replace(", ", "\n"))

The researchers say a possible explanation for this warming bias may lie in a "multiplier effect
" whereby a modest degree of warming -- for instance from volcanoes releasing carbon dioxide into the atmosphere -- naturally speeds up cer
leading
on average
to still more warming.

Because the carbon cycle
which is a key driver of long-term climate fluctuations
is itself composed of such processes
increases in temperature may lead to larger fluctuations
biasing the system towards extreme warming events.

-> l'articolo è stato riassunto con successo
```

"Le immagini raffigurate provengono dal progetto realizzato su Google Colaboratory, piattaforma basata su cloud che consente di scrivere ed eseguire codice Python attraverso il proprio browser"