Programi: Shkenca Kompjuterike

Lënda: Inteligjenca Artificiale

TITULLI i PROJEKTIT  
Projekti Python – Krijimi i Chatbot-it

EMRI I PROFESORIT EMRI I STUDENTIT/ES

Hyrmet Mydyti Petrit Qerimi

18 DHJETOR, 2022

Tabela e përmbajtjes

[Abstrakti 2](#_Toc123392355)

[Çfarë është NLTK? 3](#_Toc123392356)

[Kodimi i Chatbot në Python 3](#_Toc123392357)

[Tstimi i përgjigjeve dhe modelit 6](#_Toc123392358)

# **Abstrakti**

Në këtë projekt ne do të ndërtojmë një chatbot duke përdorur teknika të të mësuarit të thellë (deep learning) dhe machine learning.

Baza e të dhënave do të përmbaj një JSON fajll me përgjijget e mundshme dhe në këtë projekt do të përdorim edh dy pickle file[[1]](#footnote-1) për serializimin e objekteve (fjalëve dhe kategorive të tyre) në python.

Chatboti do të trajnohet në bazën e të dhënave që përmban modelet e përgjigjeve, fjalët dhe kategoritë:

1. Intents.json – JSON fajlli të dhënave i cili ka modele dhe përgjigje të paracaktuara.

2. Words.pkl – Ky është një fajll pickle në të cilin ruajmë listen e fjalëve.

3. Classes.pkl – Fajlli pickle që përmban listën e kategorive të përgjigjeve.

Në këtë projekt do të krijojmë një chatbot të bazuar në kthimin automatik të përgjigjeve duke përdorur NLTK, Keras[[2]](#footnote-2), Python, etj.

Ne do të përdorim një rrjet nervor (neural network) për të klasifikuar se cilës kategori i përket mesazhi i përdoruesit dhe më pas do të japim një përgjigje të rastësishme nga lista e përgjigjeve.

## **Çfarë është NLTK?[[3]](#footnote-3)**

NLTK (Natural Language Toolkit) është një grup librarish për NLP (Natural Language Processing) të simboleve për anglishten. Kjo librari vjen me demonstrime grafike dhe të dhëna model për testim.

Që nga paraqitja për herë të parë në vitin 2001, NLTK mbështetë hulumtimin dhe edukimin në NLP (Natural Language Processing) dhe fusha të tjera të lidhura ngushtë. Këto përfshijnë inteligjencën artificiale, gjuhën empirike, machine learning, etj.

# **Kodi për trajnimin e modelit të Chatbot në Python**

Kodi në python për chatbot është ruajtur në fajlli(skripta) train\_chatbot.py –ku ruhet kodi për trajnimin e chatbot dhe do të shfrytëzohet për këto katër faza të trajnimit të chatbotit:

* 1. Leximi i të dhënave nga JSON fajlli

Pasiqë baza e të dhënave është në format JSON, do të përdorim paketën json për të analizuar fajllin JSON në Python.

data\_file = open('intents.json').read()

intents = json.loads(data\_file)

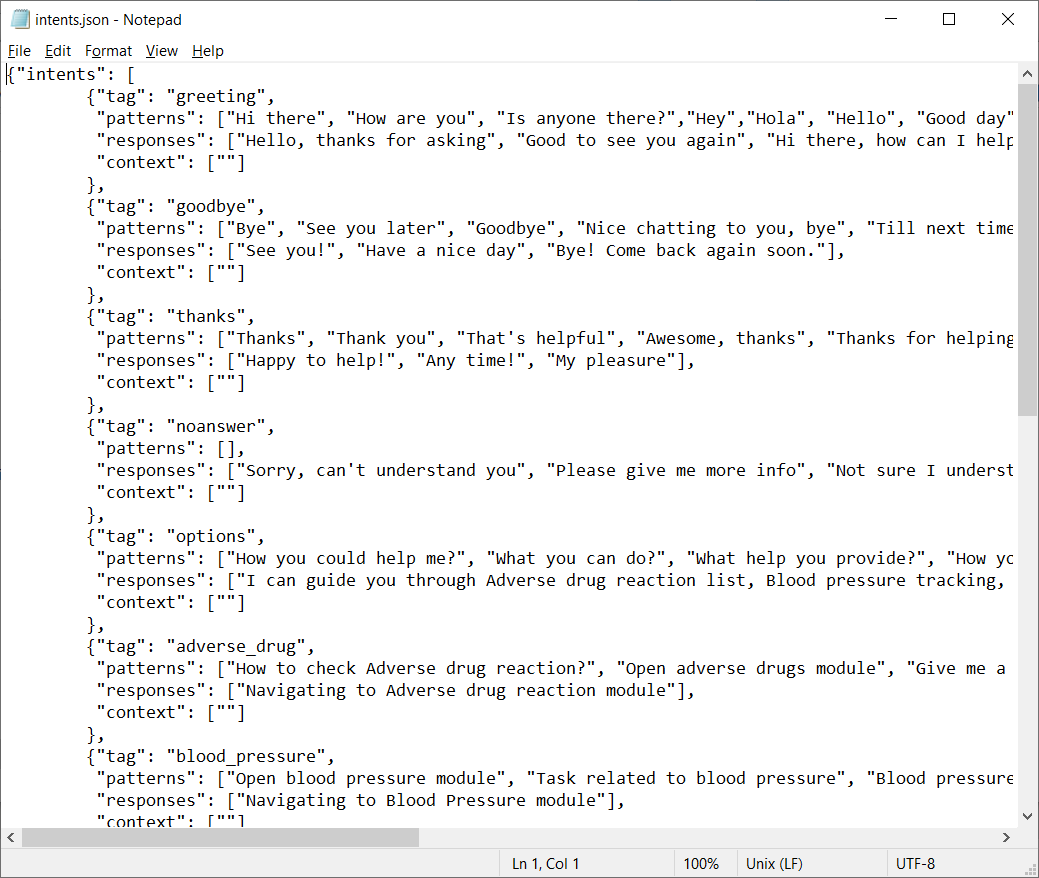


Figura 1. Fajlli intents.json

* 1. Përpunimi të dhënave

Kur punojmë me të dhënat tekstuale, duhet të kryejmë përpunimin paraprak mbi të dhënat përpara se të bëjmë një model machine learning.

Tokenizimi është gjëja më themelore që duhet të bëni me të dhënat tekstuale. Tokenizimi është procesi i ndarjes së të gjithë tekstit në pjesë të vogla si fjalë.

Gjithashtu tduhet të bëjmë edhe lematizimin i cili është një proces i grupimit të fjalëve të kuptimit të njejtë dhe largimi i fjalëve të dyfishta. Në fund do të krijojmë fajllat pickle për fjalët dhe klasët e tyre:

print (len(words), "unique lemmatized words", words)

pickle.dump(words,open('words.pkl','wb'))

pickle.dump(classes,open('classes.pkl','wb'))\

* 1. Krijimin i të dhënave të modelit

Pas leximit të të dhënave do të krijojmë të dhënat e trajnimit në të cilat do të japim hyrjen dhe daljen. Hyrja jonë do të jetë modeli dhe dalja do të jetë klasa të cilës i përket modeli ynë i hyrjes.

train\_x = list(training[:,0])

train\_y = list(training[:,1])

print("Training data created")

* 1. Ndërtimi i modelit

Të dhënat e model cilat do të ruhen në fajllin chatbot\_model.h5 i cili përmban informacione rreth modelit të trajnuar dhe vlerëën apo rëndësinë e neuroneve.

Kur të kemi gati të dhënat e trajnimit, do të ndërtojmë një rrjet nervor të thellë (deep neural network) që ka 3 shtresa [[4]](#footnote-4) . Ne do të përdorim API-në sekuenciale Keras për këtë.

model = Sequential()

model.add(Dense(128, input\_shape=(len(train\_x[0]),), activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(len(train\_y[0]), activation='softmax'))

....

hist = model.fit(np.array(train\_x), np.array(train\_y), epochs=200, batch\_size=5, verbose=1)

model.save('chatbot\_model.h5', hist)

print("model created")

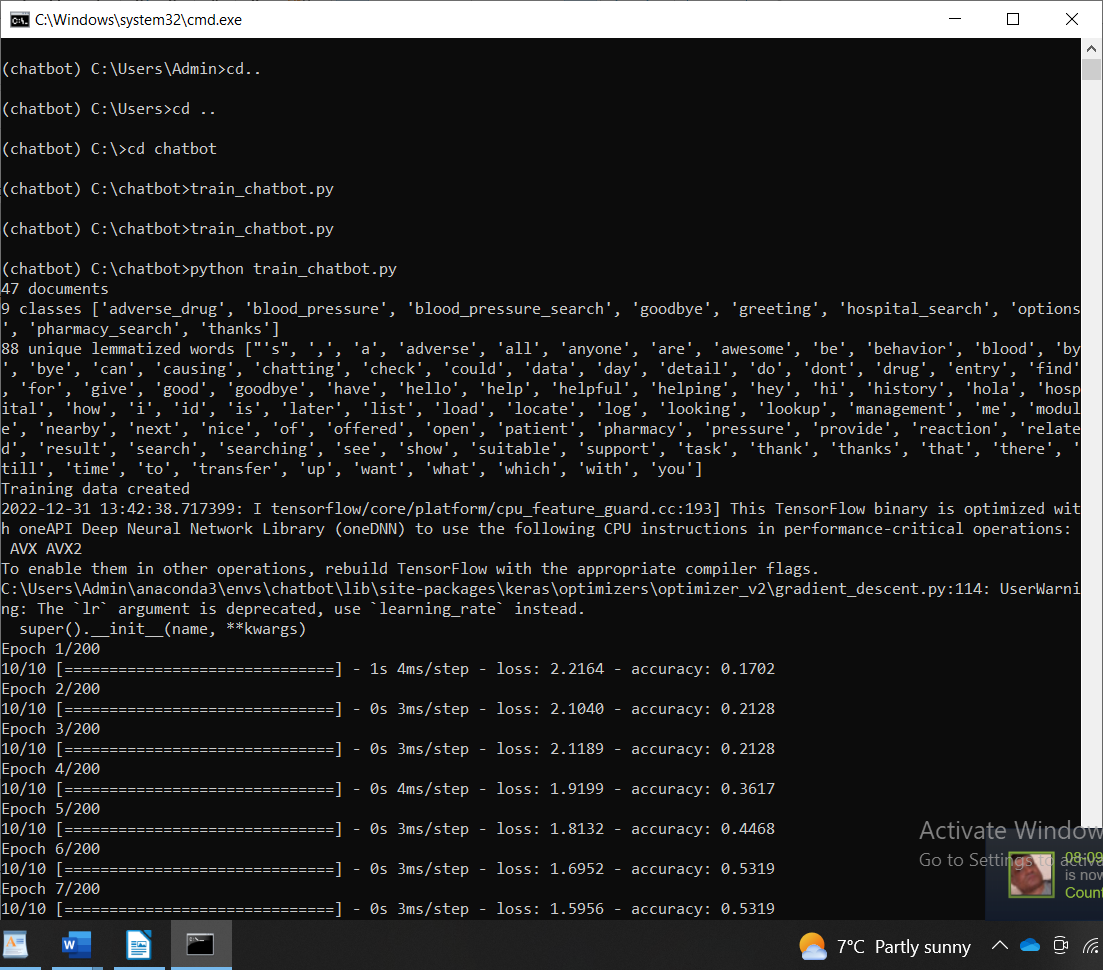


Fig. 2 Ekzekutimi i skriptës train\_chatbot.py meqë rast do të krijohet pickle fajllat, të dhënat e modelit dhe vetë modeli i trajnuar

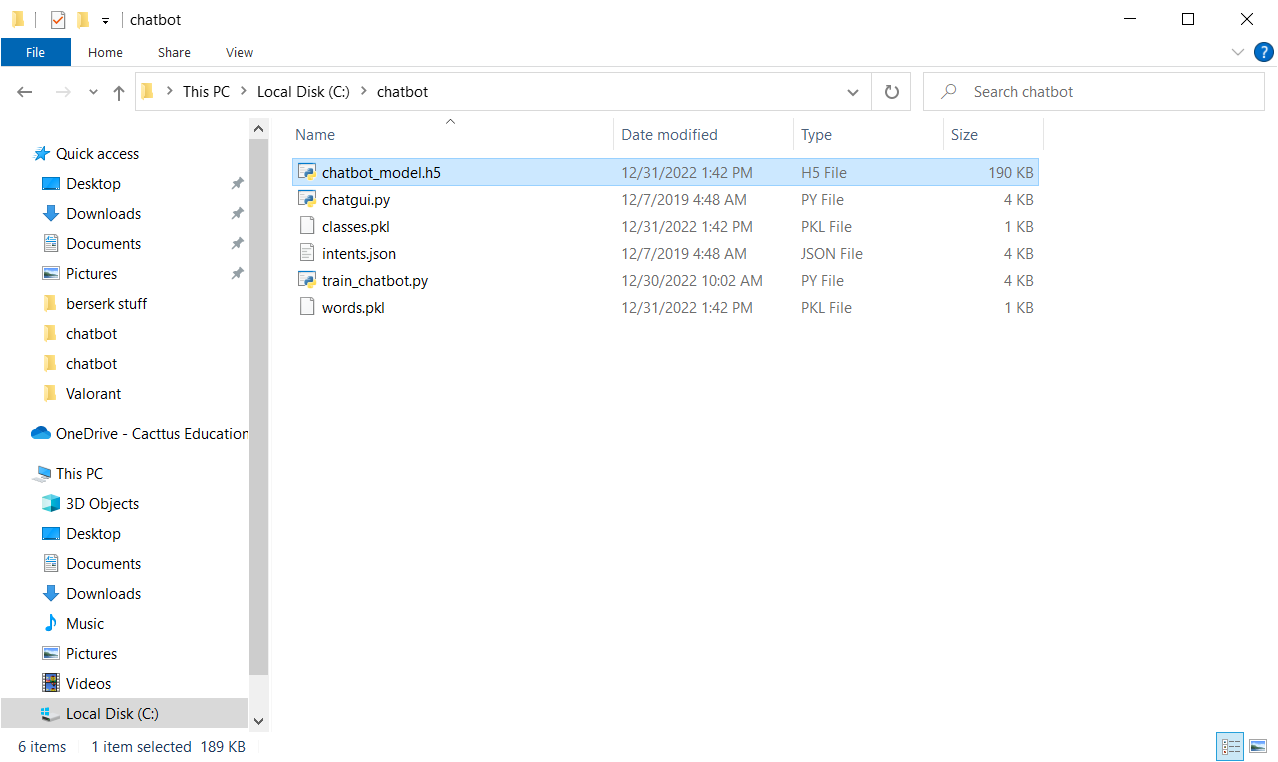


Fig. 4. Follderi i projektit pas ekzekutimit të skriptës së trajnimit train\_chatbot.py, ku shihen fajllat pickle dhe fajlli i modelit

# **Kodi për testimin e modelit dhe të Chatbot-it**

Kodi në python për testim është ruajtur në fajllin chatgui.py i cili na mundëson që përmes GUI interface të bëjmë pyetje dhe të marrim përgjigje automatike nga Chatbot-i. Ne do të ngarkojmë modelin e trajnuar dhe më pas përmes ndërfaqes grafike do të shohim përgjigjen nga chatboti. Modeli do të identifikojnë klasën e fjalëve nga shfrytëzuesi dhe më pas do të kthej një përgjigje të rastësishme nga lista e përgjigjeve.

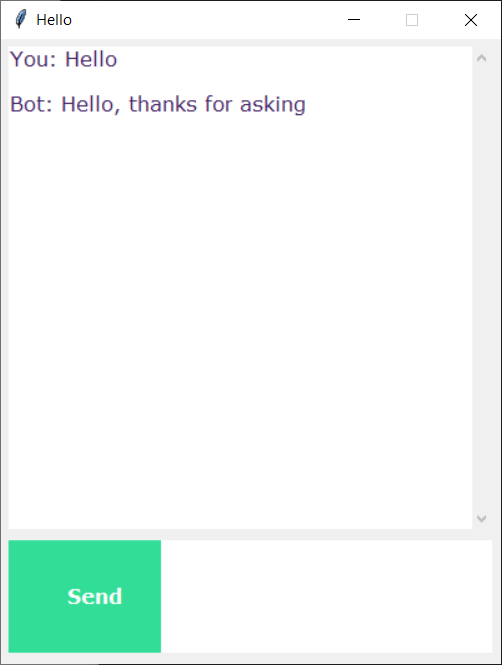


Fig 5. Testimi i chatbotit përmes GUI interface (chatgui.py)

1. https://docs.python.org/3/library/pickle.html [↑](#footnote-ref-1)
2. https://keras.io/ [↑](#footnote-ref-2)
3. https://www.nltk.org/ [↑](#footnote-ref-3)
4. https://www.sciencedirect.com/topics/engineering/three-layered-neural-network [↑](#footnote-ref-4)