Projekt dokumentáció

Mélytanuló rendszerek (VEMIPRM353T/2024/25/1)

Készítette:

Vadász Csaba (D18ZGX)

Programtervező informatikus mesterképzés - levelező

IMDB SENTIMENT ANALYSIS - LSTM

Készítette:
Vadász Csaba

2025. 10. 17.

Tartalomjegyzék

1. Bevezetés 4

2. Adathalmaz 5

2.1 Adatok forrása és jellemzői 5

2.2 Adattisztítás és előkészítés 5

2.3 Szöveghossz vizsgálat 6

2.4 Szófelhők és gyakori szavak 7

3. Adatelőkészítés – tokenizáció és padding 10

4. Modell architektúra – LTSM felépítés 12

5. Training 13

6. Eredmények és kiértékelés 14

6.1 Confusion matrix 14

6.2 Classification report 15

6.3 ROC 15

7. Következtetések 16

Melléklet 17

Források 18

1. Bevezetés

A projekt célja egy Many-to-One LSTM (Long Short-Term Memory) neurális háló implementálása, amely filmkritikák szövegéből képes automatikusan felismerni a sentiment-et, vagyis azt, hogy az adott kritika pozitív vagy negatív véleményt fejez-e ki. A sentiment analysis napjaink egyik legfontosabb NLP (Natural Language Processing) alkalmazási területe, széles körben használják ügyfélérzelmek elemzésére, közösségi média monitoringra és termékértékelések feldolgozására.

A LSTM architektúra különösen alkalmas szekvenciális adatok feldolgozására, mivel képes hosszú távú függőségeket megtanulni a szövegekben. A Many-to-One változat egy teljes szöveg szekvenciát tud feldolgozni, majd egyetlen kimenettel tér vissza - jelen esetben egy bináris döntéssel arról, hogy a kritika pozitív vagy negatív. Ez ideális megoldás a sentiment classification feladathoz, ahol a szöveg egészének kontextusa számít a végső ítélethez.

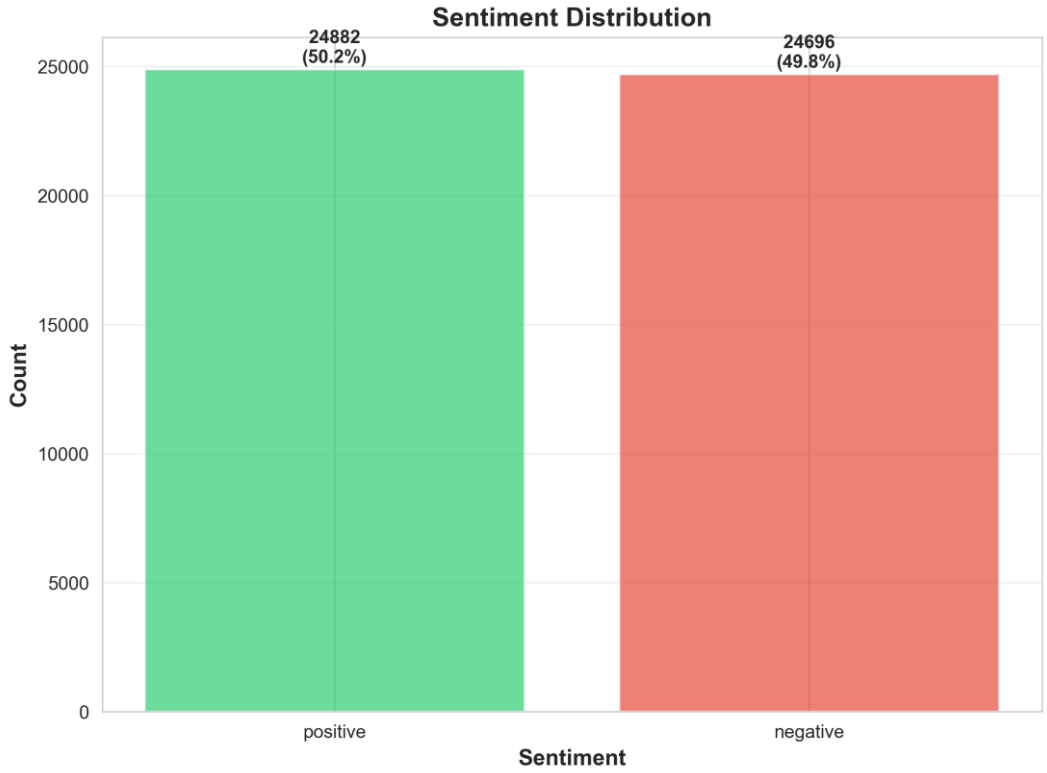
2. Adathalmaz

2.1 Adatok forrása és jellemzői

Az adathalmaz a Kaggle-ről származik, az IMDB Dataset of 50K Movie Reviews elnevezésű gyűjtemény. Az eredeti fájl 50,000 filmkritikát tartalmaz, amelyek egyenletesen oszlanak meg pozitív és negatív kategóriák között. Az adattisztítás során 422 duplikált sort távolítottam el, így a végleges dataset 49,578 egyedi kritikát tartalmaz.

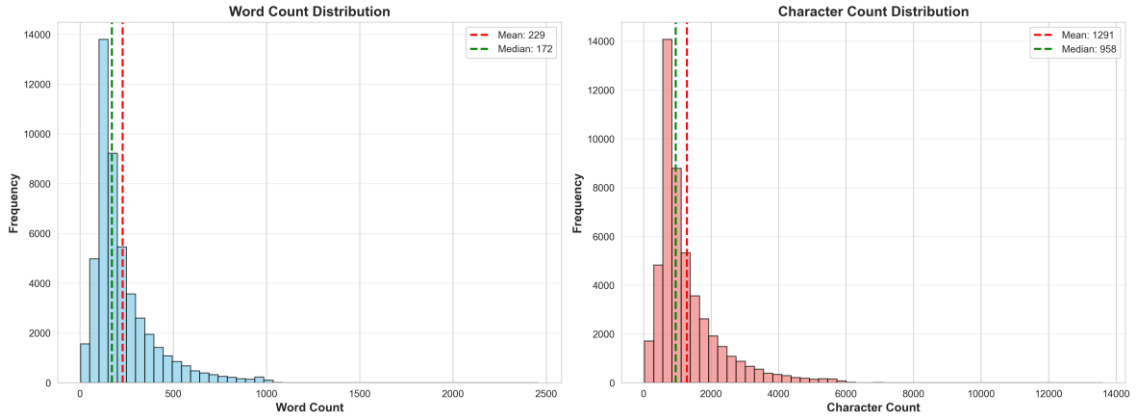
2.2 Adattisztítás és előkészítés

Az adatelőkészítés több lépésből állt. Először a HTML tageket távolítottam el a szövegekből, mivel az eredeti kritikák tartalmaztak formázási elemeket. Ezután ellenőriztem a hiányzó értékeket - szerencsére egyetlen hiányzó adat sem volt a dataset-ben. A duplikátumok eltávolítása után validáltam a sentiment értékeket is, így biztosítva, hogy minden sor helyes osztályba tartozik.

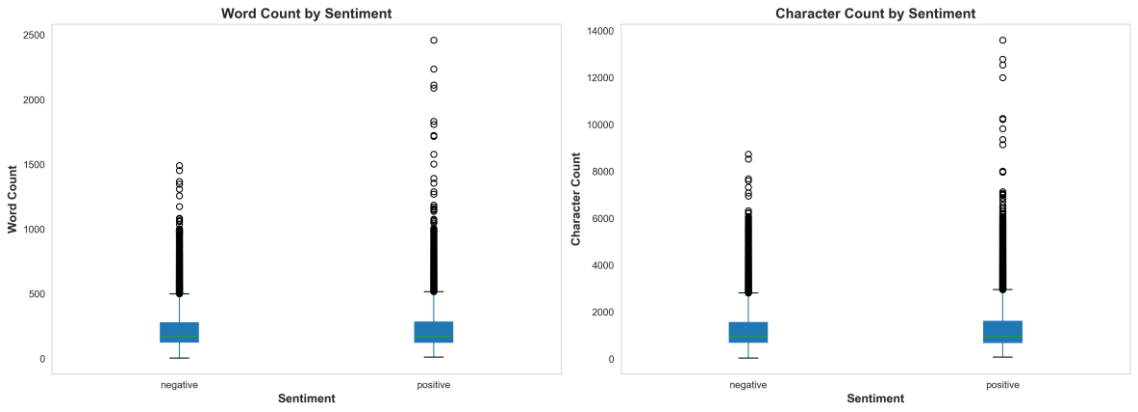


A tisztított adathalmaz tökéletesen kiegyensúlyozott lett: 24,882 pozitív és 24,696 negatív kritika, ami 50.19% vs. 49.81% arányt jelent. Ez az 1.01:1-es arány ideális, mivel a modell nem lesz elfogult egyik osztály felé sem, így nincs szükség újra mintavételezésre vagy osztálysúlyozásra. A kiegyensúlyozott dataset kulcsfontosságú a megbízható modell teljesítményhez.

2.3 Szöveghossz vizsgálat



A szöveg hosszának eloszlása természetes jobbra ferdeséget mutat, ami jellemző a valós szövegekre. Az átlagos kritika 229 szóból áll, a medián pedig 172 szónál található. A legtöbb kritika 100-300 szó közé esik, ami megfelelő információmennyiséget tartalmaz a sentiment felismeréséhez. A bal oldali ábra a szavak számának, a jobb oldali pedig a karakterek számának eloszlását mutatja. Mindkét eloszlás hasonló mintázatot követ, megerősítve az adatok konzisztenciáját.

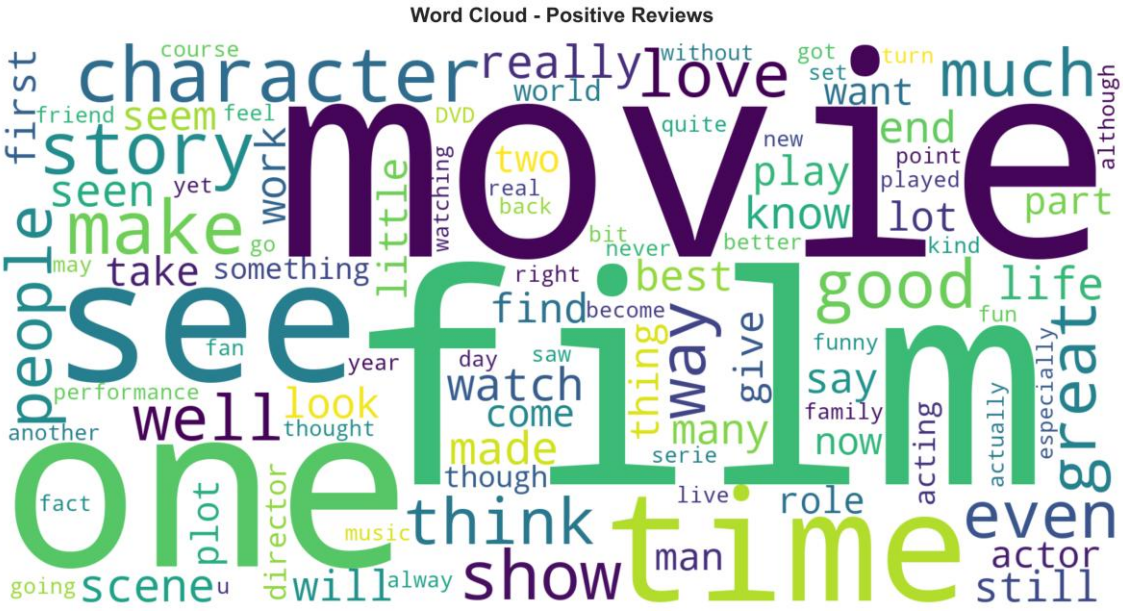


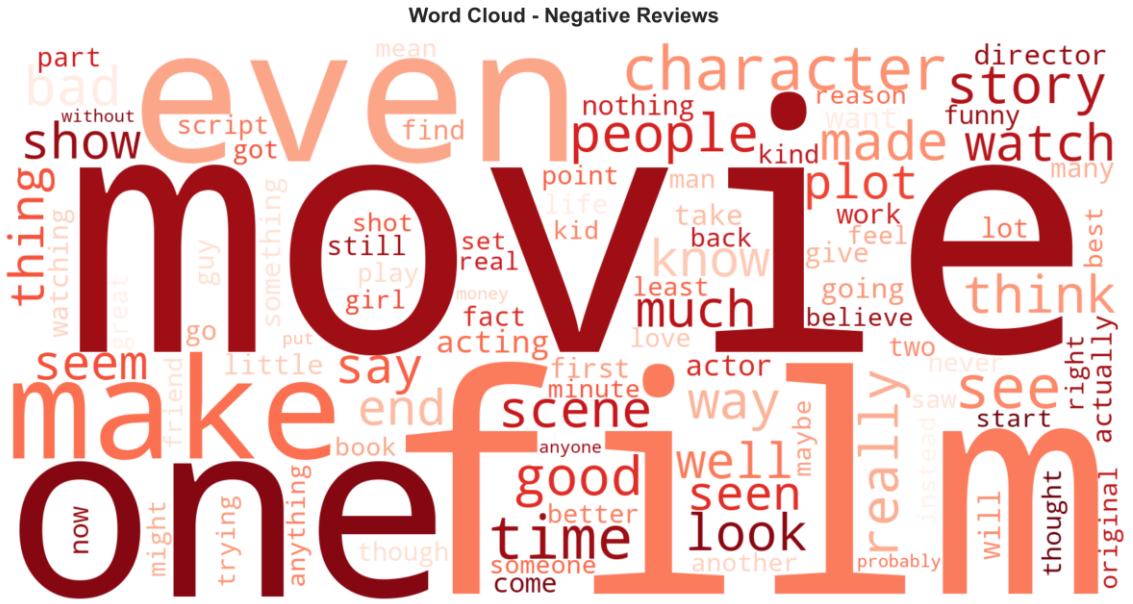
A boxplot elemzés szerint nincs szignifikáns különbség a pozitív és negatív kritikák hosszában. Mindkét kategória hasonló eloszlást mutat, ami azt jelzi, hogy a

sentiment nem függ a szöveghossztól - tehát nem igaz az, hogy a hosszabb kritikák automatikusan pozitívabbak vagy negatívabbak lennének. Ez fontos megfigyelés, mivel igazolja, hogy a modellnek valóban a tartalmat kell megtanulnia, nem csak a hosszt.

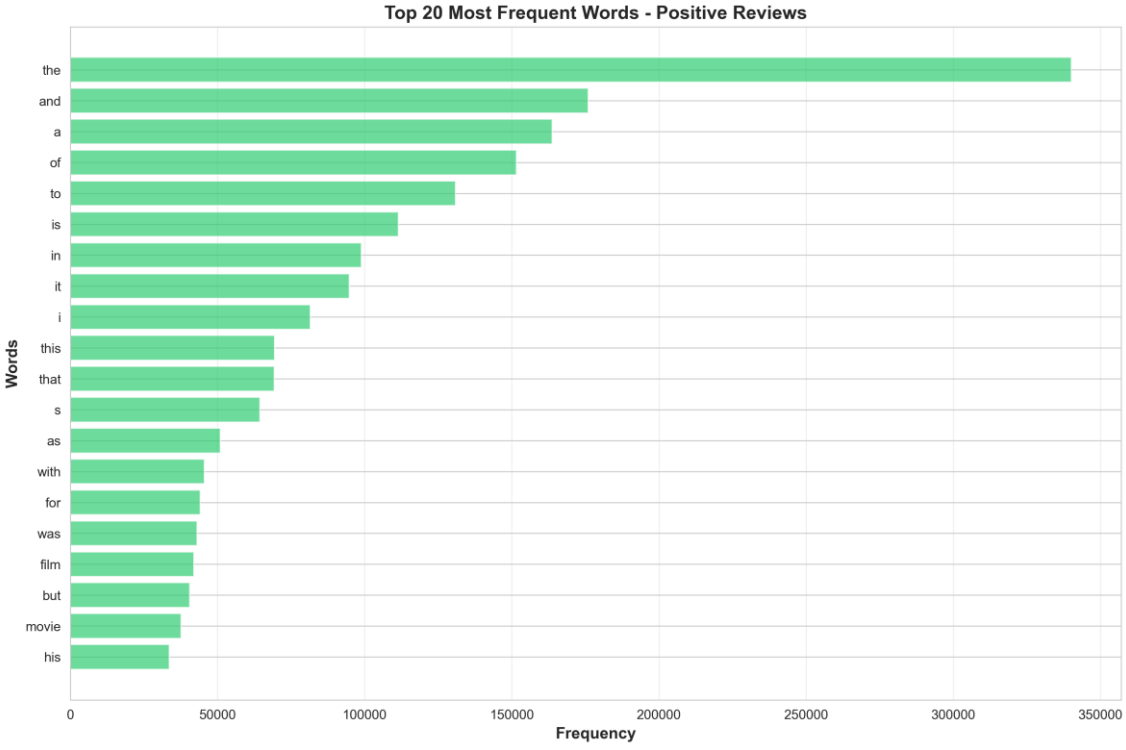
2.4 Szófelhők és gyakori szavak

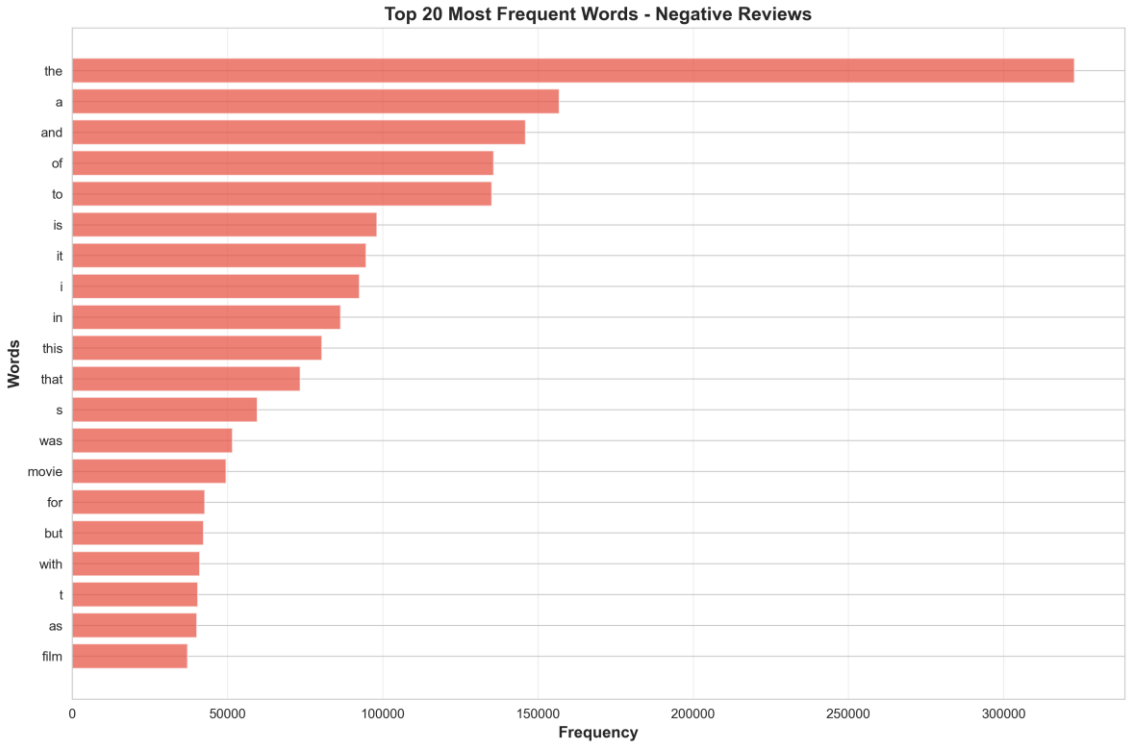
A pozitív és negatív kritikák szófelhői vizuálisan is mutatják a két kategória közötti nyelvi különbségeket. A pozitív kritikákban olyan szavak dominálnak, mint "great", "best", "love", "excellent" és "wonderful", amelyek erős pozitív érzelmi töltéssel bírnak. Ezzel szemben a negatív kritikák szófelhőjén a "bad", "worst", "waste", "boring" és "awful" szavak jelennek meg leggyakrabban, egyértelműen negatív sentiment-et kifejezve.





A top 20 leggyakoribb szó részletes összehasonlítása még pontosabb képet ad a két kategória nyelvhasználatáról. A pozitív kritikák "great" és "excellent" szavakat használnak kiemelt gyakorisággal, míg a negatív kritikák "bad" és "boring" szavakkal jellemezhetők. Ez a különbség megerősíti, hogy valóban léteznek olyan kulcsszavak, amelyek jellemzőek az egyes kategóriákra, és amelyeket a modell megtanulhat.

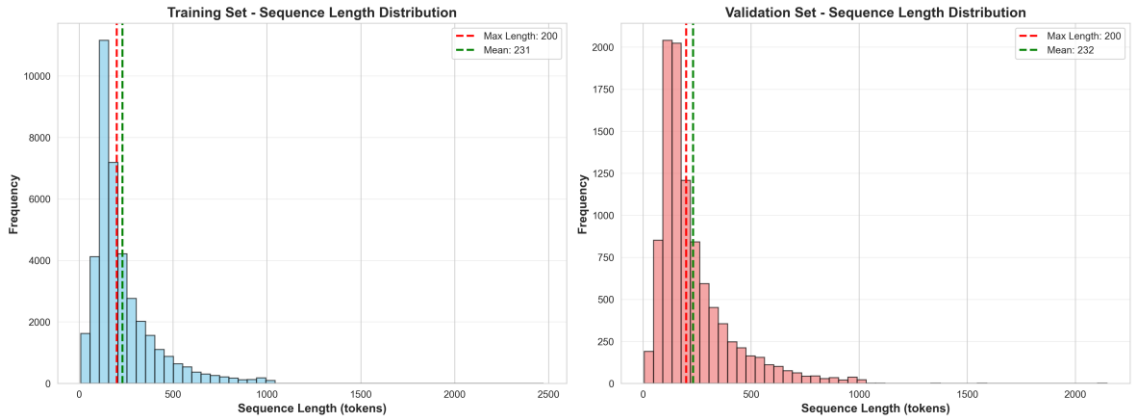



Top 20 Most Frequent Words - Negative Reviews
Words
the
a
and
of
to
is
it
i
in
this
that
s
was
movie
for
but
with
t
as
film
0
50000
100000
150000
200000
250000
300000
Frequency

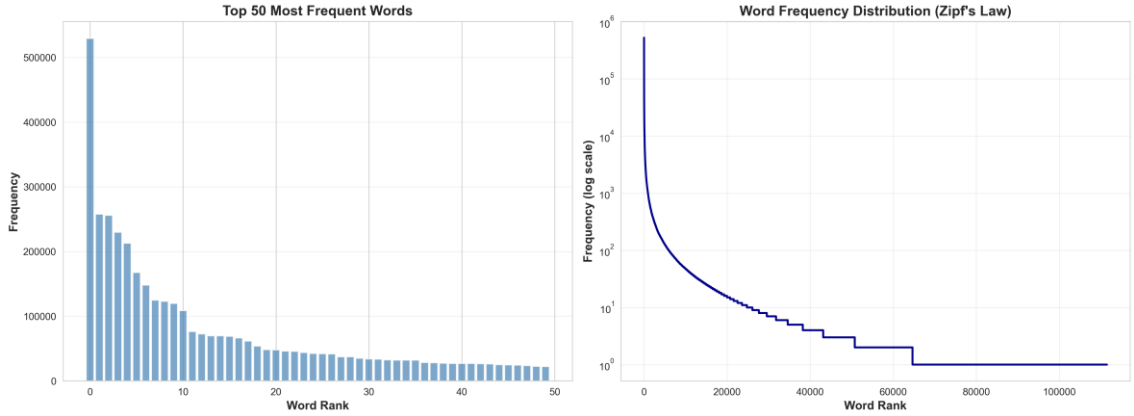
3. Adatelőkészítés – tokenizáció és padding

A LSTM modell nem képes közvetlenül szöveggel dolgozni, ezért a kritikákat numerikus szekvenciákká kellett alakítani. Ehhez Keras Tokenizert használtam, amely a 10,000 leggyakoribb szót tartalmazza a vocabulary-ben. Ez a méret optimális kompromisszum a modell komplexitása és a lefedettség között - elegendő a lényeges szavak megtartásához, ugyanakkor nem túl nagy a számítási igény szempontjából.

A padding során minden szekvenciát 200 token hosszúságra állítottam be. A rövid kritikák végére nullákat adtam hozzá (post padding), míg a hosszabb kritikák végét levágtam (post truncation). Ez az egységes hossz szükséges a LSTM batch processing-hez.



A szekvencia hossz eloszlás mutatja, hogy a 200 token-es maximum jó választás volt. A training és validation set hasonló eloszlást mutat, ami fontos a modell megbízható működéséhez. Az adatok 58.9%-a padding-ot igényelt (rövidebb kritikák), 40.8%-a pedig truncation-t (hosszabb kritikák). Mindössze 0.3% volt pontosan 200 token hosszú. A két set közötti hasonlóság azt jelzi, hogy a split reprezentatív volt.



A vocabulary statisztika két fontos dolgot mutat. A bal oldali ábra a top 50 leggyakoribb szó előfordulását szemlélteti - látható, hogy néhány szó rendkívül gyakori (mint "the", "and", "a"), míg a többi exponenciálisan csökkenő gyakorisággal szerepel. A jobb oldali ábra Zipf törvényét demonstrálja log skálán, ami természetes nyelvekre jellemző: néhány szó nagyon gyakori, míg a legtöbb szó ritka. Ez a mintázat igazolja, hogy az adathalmaz természetes nyelvi sajátosságokat mutat.

4. Modell architektúra – LTSM felépítés

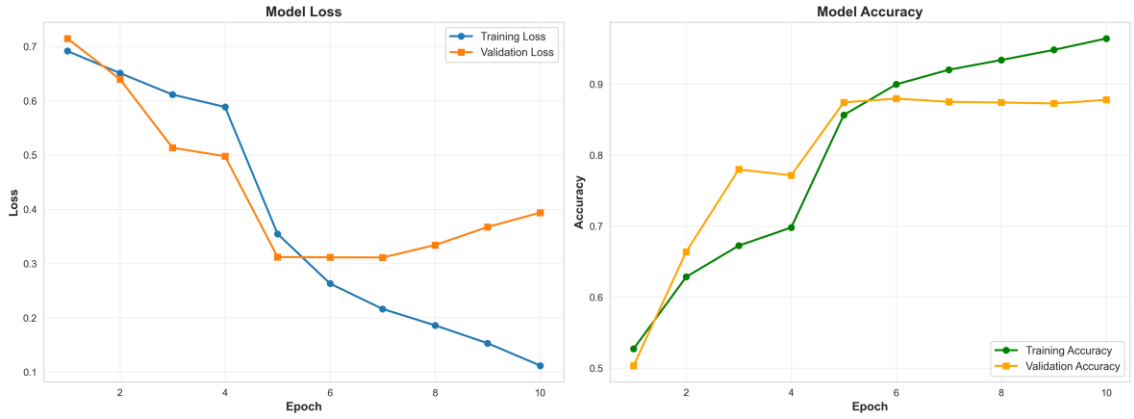
A modell egy klasszikus Many-to-One LSTM architektúrát követ. Az első réteg egy Embedding layer, amely a 10,000 szavas szójegyzéket 128 dimenziós vektortérbe képezi le. Ez a word embedding biztosítja, hogy a hasonló jelentésű szavak közel kerüljenek egymáshoz a vektortérben.

A LSTM layer 128 unit-ot tartalmaz, 0.5 dropout és 0.2 recurrent dropout értékekkel a túltanulás megelőzésére. A Many-to-One konfiguráció (return_sequences=False) miatt csak az utolsó időlépés kimenetét használja, amely az egész szekvencia összefoglaló reprezentációja. Ezt követi egy további Dropout layer 0.5 rátával, végül pedig egy Dense output layer sigmoid aktivációval, amely 0 és 1 közötti valószínűséget ad vissza a pozitív sentiment-re.

A teljes modell 1,411,713 trainable paraméterből áll. Ebből 1,280,000 az Embedding layer-ben, 131,584 a LSTM-ben, és 129 a Dense output layer-ben található. Az Adam optimizert használtam 0.001 learning rate-tel, binary crossentropy loss függvénnyel.

5. Training

A training 10 epoch-ra volt beállítva, de az Early Stopping callback miatt már a 10. epoch után megállt, amikor a validation loss 3 egymást követő epoch-on át nem javult. A bal oldali ábra a loss görbéket mutatja: a training loss fokozatosan csökken 0.69-ről 0.11-re, míg a validation loss először csökken, majd 0.31-0.39 között stagnál. Ez enyhe overfitting-re utal, de még elfogadható mértékben.



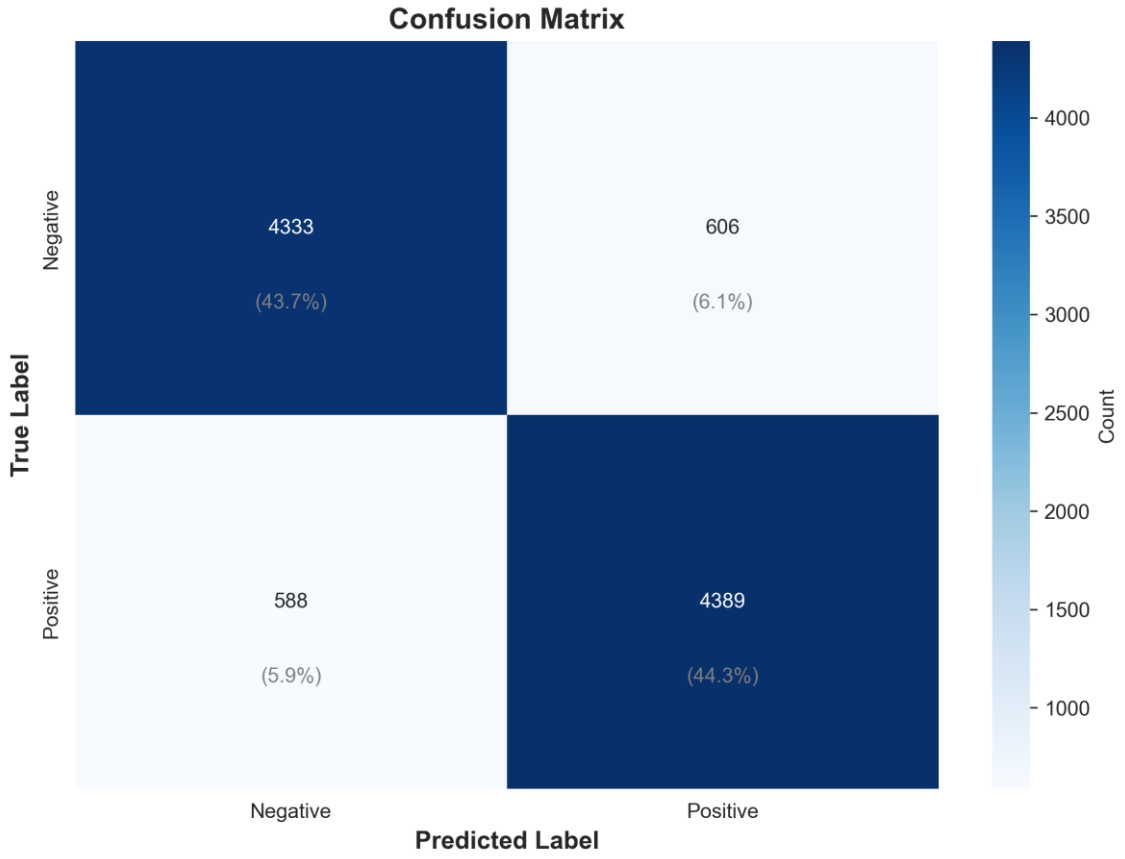
A jobb oldali ábra az accuracy görbéket szemlélteti. A training accuracy gyorsan emelkedik és 96%-ig jut, míg a validation accuracy 88% körül stabilizálódik. A train és validation accuracy közötti 8%-os különbség szintén az enyhe overfitting jele. A 7. epoch-nál a learning rate felére csökkent (0.001-ről 0.0005-re) a ReduceLROnPlateau callback miatt, amikor a validation loss platózott.

A best model weights az 6. epoch-ból származik, amikor a validation accuracy 87.96%-os volt. Az Early Stopping ezt a súlykészletet állította vissza, így a modell nem a túltanult állapotát mentette el, hanem a legjobb generalizációs képességgel rendelkezőt.

6. Eredmények és kiértékelés

6.1 Confusion matrix

A confusion matrix pontosan mutatja a modell teljesítményét mindkét osztályra. A negatív kritikák 87.96%-át helyesen negatívnak, míg 12.04%-át tévesen pozitívnak osztályozta (False Positive). A pozitív kritikák esetében 88.08% volt helyesen pozitív, 11.92% pedig téves negatív (False Negative). A hibák szimmetrikusak (FP ≈ FN), ami azt jelenti, hogy nincs bias egyik osztály felé sem - a modell egyformán jól teljesít mindkét kategórián.

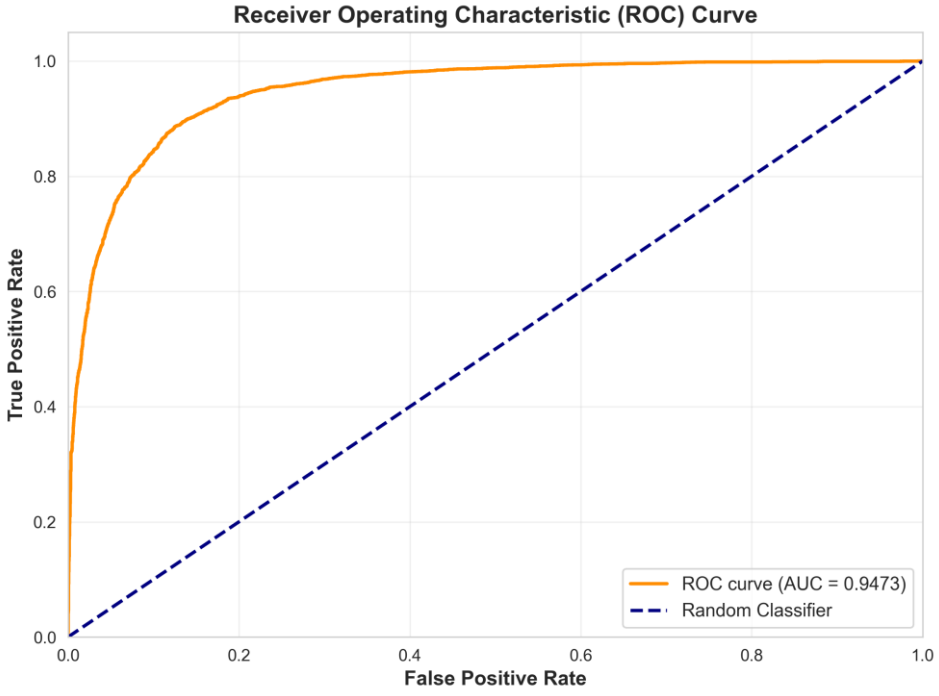


6.2 Classification report

A precision, recall és F1-score értékek mindkét osztályra 0.88, ami kiváló kiegyensúlyozott teljesítményt mutat. A precision azt jelenti, hogy a pozitívnak/negatívnak osztályozott kritikák 88%-a valóban az adott kategóriába tartozik. A recall szerint az összes pozitív/negatív kritika 88%-át sikerült helyesen azonosítani. Az F1-score, amely a precision és recall harmonikus közepe, szintén 0.88 - ez megerősíti a kiegyensúlyozott teljesítményt.

6.3 ROC

Az AUC (Area Under Curve) score 94.71%, ami kiváló eredmény. Ez az érték azt jelzi, hogy a modell 94.71% valószínűséggel ad magasabb pozitív predikciós score-t egy véletlenszerűen választott pozitív mintának, mint egy negatívnak. Az AUC > 0.90 általában "excellent" kategóriába tartozik. A ROC görbe jelentősen a random classifier (kék szaggatott vonal) felett helyezkedik el, ami azt mutatja, hogy a modell sokkal jobb, mint a véletlen találgatás. A görbe bal felső sarokhoz közeli elhelyezkedése jelzi az optimális működést.



7. Következtetések

A projekt során sikeresen implementáltam egy Many-to-One LSTM modellt, amely 87.96% validation accuracy-t és 94.71% AUC score-t ért el. Ez kiváló eredmény sentiment analysis feladatra. A modell fő erőssége a kiegyensúlyozott teljesítmény: mindkét osztályt egyformán jól ismeri fel, nincs bias. Az enyhe overfitting (8% gap a train és validation között) elfogadható, további regularizációval tovább javítható lenne.

A projekt során megtanultam a LSTM architektúrák gyakorlati alkalmazását, az adatelőkészítés fontosságát, és a callback-ek használatát a training optimalizálásához. A kiegyensúlyozott dataset kulcsszerepet játszott a jó eredményekben. Továbbfejlesztési lehetőségként felmerül a Bidirectional LSTM kipróbálása, pretrained embeddings (GloVe, BERT) használata, vagy ensemble módszerek alkalmazása a teljesítmény további növelésére.

Melléklet

1. számú melléklet: dokumentáció
2. számú melléklet: forráskód (tömörítve)
3. számú melléklet: prezentáció

Források

Forráskód:

[1] Csaba79-coder. (2025). *IMDB Sentiment Analysis with LSTM* [Szoftver]. GitHub. https://github.com/Csaba79-coder/imdb-sentiment-lstm

Adathalmaz:

[2] Lakshmi, N. (2019). *IMDB Dataset of 50K Movie Reviews* [Adathalmaz][1]. Kaggle. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[1] Utolsó megtekintés dátuma: 2025. október 17.