

## پروژه Sentiment Analysis

پروژه تحلیل احساسات (دوکلاسه) با استفاده از fine-tune کردن مدل [ParsBERT](#)

دیتاست :

این دیتاست مربوط به کامنت‌های اسنپ است که در ابتدا قرار بود از نظرات دانشجویان درباره ارزیابی اساتید استفاده شود.

مشکل کتابخانه Hazm :

مشکل اینجاست که برخی از پیش‌نیازهای Hazm با بعضی از پیش‌نیازهای TensorFlow همخوانی ندارند. در اصل، به همین دلیل از Environment استفاده می‌کنیم تا اگر شخص دیگری بخواهد مدل را اجرا کند، نیازمندی‌های موردنظر به‌صورت خودکار نصب شوند و اجرای مدل با مشکل مواجه نشود.

Hazm با numpy>=1.26.4 سازگار نیست ولی در محیط‌های linux و macOS نصب می‌شود و با توجه به اینکه پردازنده‌های macOS مناسب پردازش Deep Learning نیستند و نیاز به GPU برای آموزش مدل داریم شروع به نوشتن تابع normalizer کردم.

تابع (JackageNormalizer)normalizer :

از ۹ بخش تشکیل شده که به ترتیب فرآیند normalization رو انجام می‌دهد:

- Normalize\_unicode
- Remove\_unwanted\_characters
- Convert\_numbers
- Convert\_numbers\_to\_words
- Standardize\_persian\_text
- Remove\_keshide
- Remove\_punctuation
- Fix\_persian\_zwnj
- Remove\_stopwords

اجرای مدل :

با توجه به حل مشکل normalization مدل را روی GPU T4 googleColab اجرا کردیم سپس دیتا‌های unclean دیتاست رو شروع به preprocessing میکنیم تا برای آموزش مدل آماده شوند.

نتیجه آموزش های مدل به صورت زیر است :

Epoch 1/3

874s 282ms/step - loss: - [=====] 2887/2887

0.3201 - sparse\_categorical\_accuracy: 0.8674 - val\_loss: 0.2871 -

val\_sparse\_categorical\_accuracy: 0.8874

Epoch 2/3

803s 278ms/step - loss: - [=====] 2887/2887

0.2543 - sparse\_categorical\_accuracy: 0.8973 - val\_loss: 0.2988 -

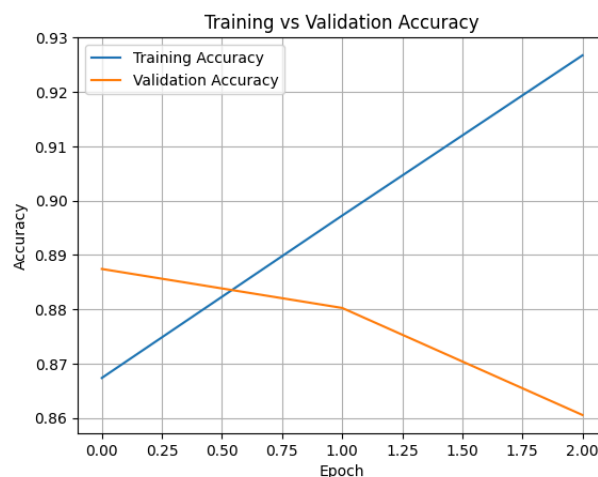
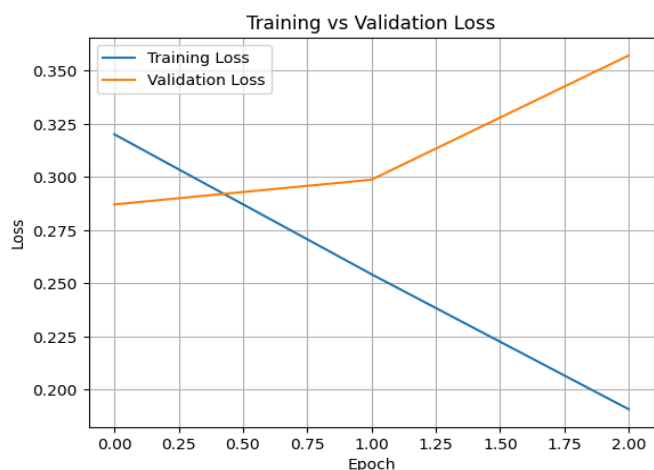
val\_sparse\_categorical\_accuracy: 0.8803

Epoch 3/3

802s 278ms/step - loss: - [=====] 2887/2887

0.1908 - sparse\_categorical\_accuracy: 0.9267 - val\_loss: 0.3572 -

val\_sparse\_categorical\_accuracy: 0.8605



Final Evaluation on Test Set

57s 87ms/step - loss: 0.3776 - - [=====] 619/619

sparse\_categorical\_accuracy: 0.8516

[0.8515561819076538 ,0.3776339590549469]

از خروجی مدل میشه نتیجه گرفت که مدل بعد از ۲ epoch دچار overfitting می شود

ساخت UI :

بخش آخر پروژه که مربوط به ساخت یک front برای مدل است به کمک کتابخانه streamlit یک صفحه ایجاد می کنیم تا کاربر ورودی text خود را وارد کند.

روی ورودی text قبل از ارسال برای مدل ۲ کار صورت میگیرد

1. Normalization
2. Spell correction

تصحیح غلط املائی :  
از آنجایی که امکان غلط املائی در در دیتا های ارسال شده وجود دارد برای حل این مشکل به کمک ارسال  
API call حل کردیم

در اصل، متن را برای سایت [Spelling](#) ارسال می‌کنیم و این سایت در صورت وجود خطا، کلمه‌ی صحیح را  
برای ما برمی‌گرداند. در صورتی که کلمه‌ی ارسال شده توسط کاربر تأیید شود، جایگزین کلمه‌ی اصلی می‌گردد.