به نام خدا

**درس مبانی یادگیری عمیق**

**گزارش پروژه پایانی**

استاد درس : دکتر مرضیه داوودآبادی

دستیاران : مرتضی حاجی‌آبادی، سحر سرکار، فائزه صادقی، مهسا موفق بهروزی، الناز رضایی، پریسا ظفری، حسن حماد، سید محمد موسوی، کمیل فتحی، شایان موسوی نیا، امیررضا ویشته

دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر

نیمسال اول تحصیلی ۱۴۰۲ - ۱۴۰۳

---

**موضوع:**

# تحلیل احساسات در متن فارسی

| ردیف | نام ونام خانوادگی | شماره دانشجویی |
|---|---|---|
| ۱ | کامیار مرادیان | |
| ۲ | مهدی قضاوی | |

جدول ۱: مشخصات اعضای گروه

# ۱ شرح موضوع و مجموعه دادگان

تجزیه و تحلیل احساسات شاخه‌ای از پردازش زبان طبیعی (NLP) است که شامل استفاده از الگوریتم‌های یادگیری ماشین برای شناسایی و استخراج خودکار اطلاعات ذهنی از متن است. ازجمله کاربردهای این زمینه می‌توان به نظارت بر سلامت روان، تجزیه و تحلیل بازخورد مشتری، توصیه محتوا و چت بات ها اشاره کرد. هدف از تجزیه و تحلیل احساسات، تعیین احساسات یا عواطف پشت یک متن است، خواه مثبت، منفی یا خنثی باشد. در این پروژه، تلاش می شود تا با استفاده از مدل های یادگیری عمیق از پیش آموزش دیده، تحلیل احساسات متون فارسی صورت گیرد به‌گونه‌ای که یک مدل یادگیری عمیق برای پیش‌بینی حضور شش احساس خشم(Anger)، نفرت(Hatred)، ترس(Fear)، غم(Sadness)، شادی(Happiness)، شگفتی(Wonder) و دسته دیگر(Other) ارائه می‌کنیم.

برای آموزش این مدل، از مجموعه‌دادگان تحت عنوان ArmanEmo استفاده می‌کنیم. مجموعه داده ArmanEmo یک مجموعه داده احساسات برچسب‌گذاری شده توسط انسان با بیش از ۷۰۰۰ جمله فارسی برچسب‌گذاری شده برای هفت دسته است. این مجموعه داده از منابع مختلفی از جمله توییتر، اینستاگرام و نظرات دیجی کالا جمع آوری شده است. برچسب ها بر اساس شش احساس پایه Ekman (خشم، ترس، خوشحالی، نفرت، اندوه، تعجب) و یک دسته دیگر (Other) برای در نظر گرفتن هر احساس دیگری که در مدل Ekman وجود ندارد، انجام شده است. همراه با مجموعه داده، نویسندگان چندین مدل پایه برای طبقه بندی احساسات با تمرکز بر مدل های زبانی مبتنی بر ترانسفورمر ارائه داده اند. بهترین مدل آنها با امتیاز F1 ماکرو میانگین ۷۵.۳۹ درصد در سراسر مجموعه داده آزمایشی آنها به دست آمده است. علاوه بر این، آنها آزمایشات یادگیری انتقالی را برای مقایسه عمومی سازی مجموعه داده پیشنهادی خود در مقابل سایر مجموعه داده های احساسی فارسی انجام داده اند. نتایج این آزمایشات نشان می‌دهد که مجموعه داده آنها در میان مجموعه داده های احساسی فارسی موجود، قابلیت تعمیم‌پذیری(Generalization) بهتری دارد.

# ۲ پیش‌پردازش داده ها

پیش پردازش متن یک مرحله ضروری در تحلیل احساسات است زیرا به کمک آن می‌توان داده‌های متنی بدون ساختار را به یک فرمت ساختاریافته تبدیل کرد که می‌تواند توسط الگوریتم‌های یادگیری‌ماشین استفاده شود. پیش پردازش شامل چندین مرحله مانند توکن سازی(Tokenization)، حذف کلمات اضافی(Stop-word Removal)، کاهش ابعاد(Stemming) و Lemmatization است. Tokenization فرایند شکستن متن به کلمات یا توکن‌های جداگانه است. حذف کلمات اضافی شامل حذف کلمات مشترک مانند" به"، "از" و" را " است که معنای زیادی ندارند. کاهش ابعاد، فرایند کاهش کلمات به شکل ریشه آنها است. Lemmatiztion شبیه به کاهش ابعاد است، اما شامل کاهش کلمات به شکل پایه آنها با استفاده از یک فرهنگ لغات شناخته شده است.

پیش پردازش برای یک مدل تشخیص احساسات ضروری است زیرا به کمک آن می‌توان ابعاد داده را کاهش داد که خدود باعث بهبود عملکرد مدل می‌شود. همچنین به کمک آن می‌توان نویز را از داده ها حذف کرد، مانند علائم نگارشی، که می‌توانند باعث افت دقت تحلیل شوند. علاوه بر این، پیش‌پردازش می‌تواند به کمک استانداردسازی داده‌ها، آنها را قابل مقایسه و تحلیل‌کردن کند.

در زمینهٔ مجموعه دادگان ArmanEmo، پیش‌پردازش به دلیل کمک به آماده‌سازی داده‌های متنی برای تحلیل احساسات، اهمیت دارد. نویسندگان مقالهٔ این مجموعه دادگان، از چندین تکنیک پیش‌پردازش مانند توکن‌سازی، حذف کلمات اضافی و کاهش ابعاد برای تمیزکردن داده‌ها و آماده‌سازی آن‌ها برای تحلیل استفاده کرده‌اند. این تکنیک‌ها به کاهش ابعاد داده‌ها و حذف نویز کمک می‌کنند که می‌تواند دقت تحلیل را بهبود بخشد.

برای پیش‌پردازش و نرمال‌سازی متون و جملات فارسی، کتابخانه‌ها و ابزارهای زیادی در محیط پایتون فراهم است که از جمله آن‌ها می‌توان به کتابخانه‌های parsivar، dadmatools، hazm و... اشاره کرد.

به عنوان اولین مرحله پیش پردازش، متن را با استفاده از parsivar که ابزاری برای پیش پردازش متن فارسی است نرمالایز می‌کنیم. برای این کار، با استفاده از یک تابع (با نام clean_persian_text(text))، ابتدا یک Normalizer() از کتابخانه parsivar تعریف کرده و در گام نخست متن ورودی را نرمالایز می‌کنیم. این نرمالایزر برخی از مراحل تصحیح فاصله مبتنی بر قاعده (از جمله فاصله بین کلمات، علائم نگارشی، و ضمائم)، همراه با برخی عملیات اصلاح کاراکترها (مانند حذف حروف کششی) را اعمال می‌کند.

قوانین نرمالایزکردن آن، با این حال، جامع نیستند. به عنوان مثال، سکون عربی پس از عادی سازی حذف نخواهد شد. بنابراین، پس از معرفی متن به نرمالایزر parsivar، چند مرحله پیش پردازش اضافی را انجام می‌دهیم.

تمامی حروف انگلیسی موجود در متن را حذف می‌کنیم. علاوه‌برآن، نیاز است اگر حرفی بیش‌از یک‌بار و بصورت غیرضروری تکرارشده باشد (مانند کلمه خیلللییی)، آن را حذف کرد چراکه این نوع کلمات تنها برای تاکید در ادبیات غیررسمی به‌کار می‌روند، که این‌کار به‌کمک کتابخانه regex در پایتون قابل انجام است. در گام بعدی، حرکت‌گذاری‌های عربی را از متن پاک می‌کنیم.

به‌علاوه، باتوجه به اینکه متن ورودی اغلب از اینستاگرام یا توییتر استخراج شده، وجود هشتگ(#) را نیز چک کرده و درصورت وجود، این علامت را از متن حذف کرده و اطلاعات آن‌را نگه می‌داریم. درپایان نیز تمامی اعداد فارسی، عربی و انگلیسی از متن حذف شده و متن برای بار دیگر نرمالایز می‌شود.

# ۳ انتخاب مدل

درمقاله مربوط به مجموعه دادگان ArmanEmo، عملکرد مدل‌های مختلف CNN-based و RNN-based برروی این داده‌ها برای تسک تحلیل احساسات پیاده شده که دقت و عملکرد این مدل‌ها در تصویر زیر قابل مشاهده‌هستند:

Table 2: Comparison between the performance of different DNN models and Language Models

| Model | Precision (Macro) | Recall (Macro) | F1 (Macro) |
|---|---|---|---|
| FastText [42] | 54.82 | 46.37 | 47.24 |
| HAN [43] | 49.56 | 44.12 | 45.10 |
| RCNN [44] | 50.53 | 48.11 | 47.95 |
| RCNNVariant | 51.96 | 48.96 | 49.17 |
| TextAttBiRNN [45, 46] | 54.66 | 46.26 | 47.09 |
| TextBiRNN | 51.45 | 47.16 | 47.14 |
| TextCNN [47] | 58.66 | 51.09 | 51.47 |
| TextRNN [48] | 49.39 | 47.20 | 46.79 |
| ParsBERT | 67.10 | 65.56 | 65.74 |
| XLM-Roberta-base | 72.26 | 68.43 | 69.21 |
| XLM-Roberta-large | **75.91** | **75.84** | **75.39** |
| XLM-EMO-t | 70.05 | 68.08 | 68.57 |

در این پروژه، ما checkpoint(مدل‌های از پیش‌آموزش‌دیده) مختلف را برروی این مجموعه دادگان تست کردیم که نتایج مربوط به دقت این اجراها در جدول زیر قابل مشاهده‌اند:

| **Model** | **Accuracy(%)** |
|---|---|

| | |
|---|---|
| Roberta_fa_zwnj_base | 58.12 |
| XLM-Roberta-base | 71.26 |
| Persian-XLM-Roberta-large | 74.71 |
| **XLM-Robera-large** | **75.23** |

علت استفاده ما از مدل XLM-Roberta و ترجیح‌دادن آن به سایر مدل‌های موجود، دقت بالاتر آن برروی این مجموعه دادگان است که درمقاله مربوطه نیز به آن اشاره شده‌است. درفرآیند این پروژه، ما مدل‌های بالارا به‌همراه روش Preprocessing استفاده شده به کمک کتابخانه parsivar تست کردیم که بهترین دقت توسط مدل XLM-Roberta-large برروی مجموعه‌دادگان آزمون بدست آمد.

**مدل XLM-RoBERTa-Large**: یک مدل چندزبانه قدرتمند است که برای کارهای مختلف پردازش زبان طبیعی از جمله تجزیه و تحلیل احساسات طراحی شده است.

- **معماری و پیش‌آموزش:** XLM-RoBERTa توسعه ای از مدل RoBERTa است که خود بر اساس معماری BERT (Bidirectional Encoder Representations from Transformers) است. این مدل بر روی مجموعه بزرگی از متن از ۱۰۰ زبان از قبل آموزش داده شده است. در طول پیش‌آموزش، از هدف masked language modeling (MLM) استفاده می‌کند. تقریباً ۱۵٪ از کلمات یک جمله به طور تصادفی ماسک می‌شوند و مدل یاد می‌گیرد که این کلمات ماسک شده را پیش بینی کند.
- **قابلیت چند زبانه:** XLM-RoBERTa به طور خاص برای مدیریت موثر چندین زبان طراحی و برای تجزیه و تحلیل احساسات در حدود ۱۹۸ میلیون توییت به خوبی تنظیم شده است. تنظیم دقیق احساسات برای این مدل در هشت زبان انجام شده که شامل زبان‌های: عربی (Ar)، انگلیسی (En)، فرانسوی (Fr)، آلمانی (De)، هندی (Hi)، ایتالیایی (It)، اسپانیایی (Sp) و پرتغالی (Pt) می‌شود. نکته مهم این است که می‌توان از آن برای تجزیه و تحلیل احساسات در زبان های دیگر نیز استفاده کرد
- **Model Integration:** مدل XLM-RoBERTa-Large در کتابخانه TweetNLP ادغام شده است. ما در این پروژه، با بارگیری مدل و توکنایزر از کتابخانه Hugging Face Transformers، از آن برای عملیات تجزیه و تحلیل احساسات استفاده می‌کنیم.

# ۴ اقدامات انجام شده

- **توضیحات دررابطه با مقاله و مجموعه داده:**

ArmanEmo یک مجموعه داده احساسات با برچسب انسانی است که شامل بیش از ۷۰۰۰ جمله فارسی است. این جملات در هفت کلاس احساسی دسته بندی می‌شوند. همانطور که پیش‌تر گفته شد، این مجموعه داده از منابع مختلفی از جمله نظرات توییتر، اینستاگرام و دیجی کالا جمع آوری شده است. برچسب‌ها بر اساس شش احساس اصلی اِکمن هستند: خشم، ترس، شادی، نفرت، غم و شگفتی. علاوه بر این، یک دسته «سایر» برای توضیح احساساتی وجود دارد که توسط مدل اکمن پوشش داده نشده است.

نویسندگان این مقاله، چندین مدل پایه برای طبقه بندی احساسات ارائه می‌دهند که بر روی مدل های زبان مبتنی بر ترانسفورماتور پیشرفته تمرکز دارند. بهترین مدل آنها به امتیاز macro-averaged F1 ۷۵.۳۹٪ در مجموعه داده آزمون دست می‌یابد.

محققان آزمایش‌های یادگیری انتقالی را برای مقایسه تعمیم ArmanEmo با سایر مجموعه‌های داده احساسات فارسی موجود انجام دادند. نتایج نشان می‌دهد که ArmanEmo تعمیم‌پذیری بالاتری نسبت به سایر مجموعه‌های داده نشان می‌دهد.

هم‌چنین، ArmanEmo برای استفاده غیرتجاری در این لینک به صورت عمومی در دسترس است.

- **اقدامات انجام شده برای پیاده‌سازی مدل:**

برای آموزش مدل XLM-Roberta-Large، ابتدا برچسب‌های مجموعه دادگان ArmanEmo را به ID تبدیل کرده و سپس این مجموعه را بصورت زیر، به مجموعه‌های Train/Validatio/Test تقسیم کردیم:

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label', 'input_ids', 'attention_mask'],
        num_rows: 4900
    })
    validation: Dataset({
        features: ['text', 'label', 'input_ids', 'attention_mask'],
        num_rows: 1225
    })
    test: Dataset({
        features: ['text', 'label', 'input_ids', 'attention_mask'],
        num_rows: 1151
    })
})
```

برای بخش پیاده‌سازی مدل XLM-Roberta-large، ابتدا با استفاده از کلاس AutoTokenizer که آن را از ماژول transformers گرفتیم، چک‌پوینتِ پیش‌آموزش‌شدهٔ این مدل را دانلود کرده و برروی tokenizer لود می‌کنیم. سپس برای تعریف خود مدل در پیاده‌سازی، این‌بار به‌کمک کلاس AutoModelForSequenceClassification، این چک‌پوینت را دانلود کرده و به عنوان مدل ذخیره می‌کنیم.

برای آموزش مدل، از کتابخانه‌های TrainingArguments و Trainer از ماژول transformers استفاده کرده‌ایم. برای این بخش، یک تابع به نام get_trainer تعریف کرده که یک Trainer Object از Hugging Face را برای آموزش مدل زبانی XLM-RoBERTa-Large ایجاد و پیکربندی می‌کند.

- **TrainingArguments Configuraion**: این بخش پیکربندی آموزش را با استفاده از کلاس Training Arguments از کتابخانه transformers انجام می‌دهد. پارامترهای کلیدی آن شامل دایرکتوری خروجی برای ذخیره نتایج، تعداد epochهای آموزش، نرخ یادگیری، اندازه هر batch، مراحل warmup، weight decay و استراتژی ارزیابی و دیگر گزینه‌ها هستند.
- **مقداردهی اولیه Trainer**: در ادامه یک Trainer Object با استفاده از مدل model، tokenizer و داده‌های tokenizedشده ایجاد می‌کنیم. قابل توجه است که برای رفع مشکل padding داده‌های ورودی، از کتابخانه DataCollatorWithPadding استفاده می‌کنیم و آن را به پارامتر مربوط به data_collator از Trainer ارجاع می‌دهیم. هم‌چنین تابع compute_metrics نیز برای محاسبه متریک‌های مختلف برای ارزیابی عملکرد مدل از جمله Precision، Recall و F1 Score پیاده‌شده است.

درنهایت به کمک تابع get_trainer که تعریف کردیم مدل را برروی مجموعه دادگان آموزش می‌دهیم که نتایج آموزش مدل XLM-RoBERTa-Large به‌صورت زیر می‌باشد:

| Epoch | Training Loss | Validation Loss | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 1 | No log | 1.779821 | 0.293061 | 0.189436 | 0.349158 | 0.293061 |
| 2 | 1.723600 | 0.870409 | 0.696327 | 0.687597 | 0.729070 | 0.696327 |
| 3 | 1.723600 | 0.781565 | 0.744490 | 0.742461 | 0.747078 | 0.744490 |
| 4 | 0.796700 | 0.799700 | 0.757551 | 0.756118 | 0.761895 | 0.757551 |
| 5 | 0.492400 | 0.804058 | 0.761633 | 0.760560 | 0.763522 | 0.761633 |

در بخش fine-tune کردن مدل، مدل‌را با هایپرپارامترهای مختلف آموزش دادیم تا بهترین عملکرد مدل برروی این مجموعه دادگان بدست‌آید. تاثیرگذارترین هایپرپارامتر برای fine-tune کردن، نرخ یادگیری مدل بود که نتایج آموزش مدل به‌ازای برخی از مقادیر آن بصورت زیر است:

| **Learning Rate** | **Accuracy** |
|---|---|
| 1e-5 | 74.71 |
| 9e-6 | 75.23 |

# ۵ ارزیابی مدل

برای ارزیابی مدل، از مجموعه دادگان آزمون(Test) استفاده کرده، پیش‌بینی مدل را برای هرمتن بدست‌آورده و متریک‌های محاسبه‌شده برای عملکرد آن را ذخیره می‌کنیم. قابل توجه است که برای بخش پیشبینی مدل نیز از آبجکت تعریف شده از کلاس Trainer استفاده می‌کنیم و پیش‌بینی مدل را انجام می‌دهیم. نتایج حاصل از خروجی مدل برای مجموعه دادگان آزمون در جدول زیر قابل مشاهده هستند.

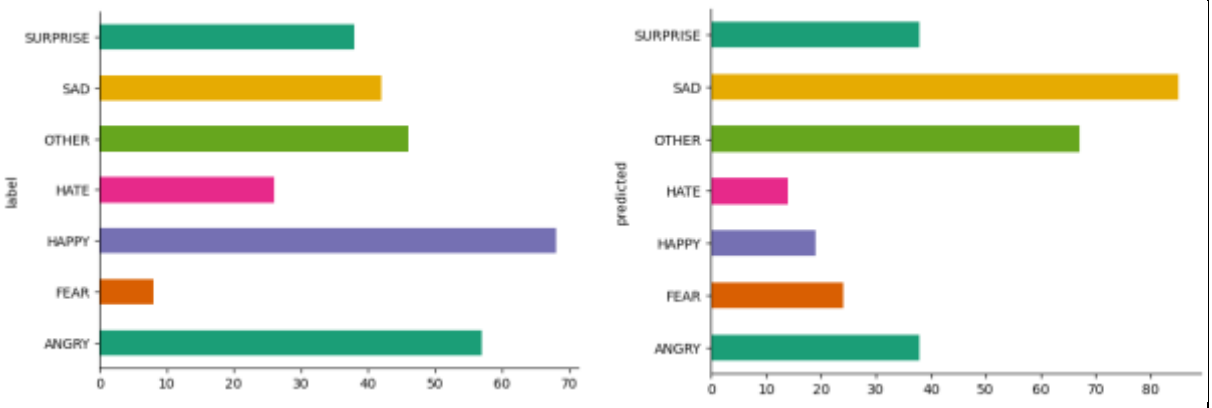
| | |
|---|---|
| Test Loss | 0.7461 |
| Test Accuracy | 75.23 |
| Test F1 Score | 75.26 |
| Test Precision | 76.21 |
| Test Recall | 75.23 |
| Test Runtime | 25.1121 |
| Test Samples Per Second | 45.835 |
| Test Steps Per Second | 2.867 |

**نتایج مدل برروی ۴ نمونه دلخواه:**

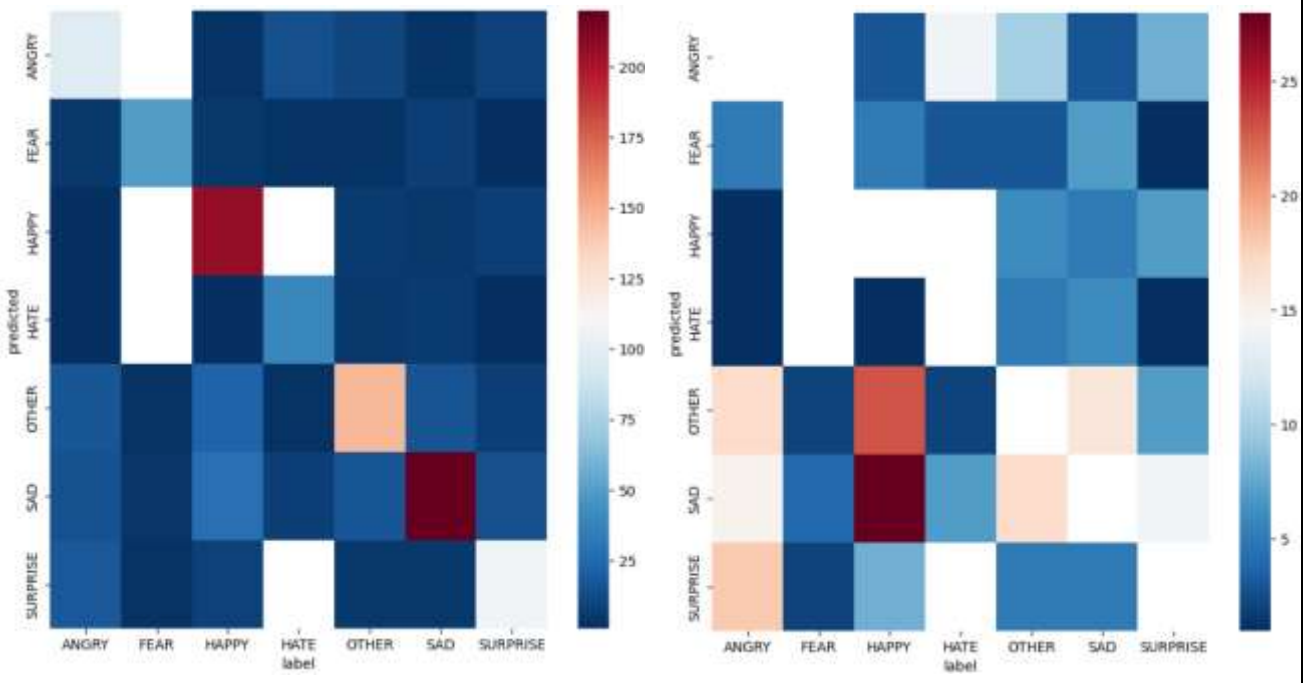
| جمله یا توییت | برچسب | پیش‌بینی مدل |
|---|---|---|
| واقعا حال به هم زنه این حجم از داستان سرایی درباره تجاوز یا چیزهای شبیه به اون برا جذب لایک و توجه | Hatred | Sad |
| کتاب امروز به دستم رسید. جنس برگه هاش خوب بودن. رنگهای شادی داشت. اسم شخصیتهای داستانها خنده دار و جالب بود. داستانهاش هم تکراری نبودن. فقط چون کل سی جلد یکجا جمع شده کمی دست رو خسته میکنه | Happiness | Happiness |
| انصافا چقدر فک کردی اینو نوشتی خخخ! ولی احسنت جالب بود | Happiness | Happiness |
| مربی خارجی حق ندارد ، مربی خارجی بی‌قد و | Angry | Angry |

| | | |
|---|---|---|
| | | قواره و . ولی میدانید که مردم هم حق اعتراض ندارن ؟ مردم هم خس و خاشاکن ؟ مردم هم بد بختن و شما از همه بهتر میبنید و میدانید ! تریبون اصلی مملکت هم در اختیار شماس ، چه قدمی برداشتید جز سانسور و جز شانتاژ رسانه‌ای ؟ |

نمودارهای مربوط به توزیع داده‌های برچسب‌گذاری شده و پیش‌بینی‌های مدل بصورت زیر هستند:




هم‌چنین، ابتدا تمامی پیش‌بینی‌های اشتباه مدل نسبت به برچسب‌ها را برروی داده‌های آزمون بدست‌آوردیم و سپس Confussion Matrix برای این‌داده‌ها و تمامی داده‌ها را رسم کردیم تا اشتباهات مدل را بهتر بررسی کنیم، که خروجی به‌شکل زیر است:




باتوجه به نمونه‌های بالا و جداول Confussion Matrix، آنچه می‌توان استنباط کرد این است که هر زمان که یک جمله معین دارای احساسات مختلط باشد، مدل اغلب، پیش‌بینی اشتباهی دارد. در چنین شرایطی، اختصاص یک و تنها یک احساس دقیق به جمله ممکن است حتی برای انسان ها نیز چالش برانگیز باشد. به همین دلیل است که از طبقه‌بندی‌کننده‌های چند برچسبی در برابر

طبقه‌بندی‌کننده‌های چند کلاسه استفاده می‌شود تا حضور بیش از یک احساس را در یک جمله مشخص کند. از آنجایی که ما در این پروژه با طبقه بندی چند کلاسه سروکار داریم، فرض می‌کنیم که هر جمله فقط یک احساس را در خود دارد؛ از این رو، فرض می‌شود که برای هر جمله فقط یک احساس واقعی وجود دارد. با این حال، این مورد ممکن است برای تمام موقعیت‌ها صادق نباشد.

بر اساس جملات داده شده می‌توان قضاوت کرد که برخی از پیش بینی های مدل اصلاً مرتبط نیستند. در واقع، بسته به زمینه، این برچسب‌های پیش‌بینی شده ممکن است به اندازهٔ حقایق پایه اختصاص داده شده معتبر در نظر گرفته شوند. با این حال، هنوز موقعیت‌های دیگری وجود دارد که به نظر می‌رسد عملکرد ضعیف مدل به بایاس‌بودن مدل نسبت به وقوع برخی کلمات خاص یا ترکیبی از کلمات در جمله مربوط می‌شود. مثلا بیش‌ترین پیش‌بینی‌های اشتباه مدل مربوط به احساس‌های Sad و Happiness بوده که درعمل باهم خیلی تفاوت دارند.

# ۶ بخش امتیازی

برای بخش امتیازی از Multitask Learning دو مدل، یکی مبتنی بر دادهٔ متنی و دیگری مبتنی بر دادهٔ صوتی استفاده شده است. برای مجموعهٔ دادهٔ این بخش، از مجموعه دادهٔ Speech Emotion Recognition Voice Dataset استفاده شده است. این مجموعه داده در Kaggle به صورت یک بخش تستی در اختیار عموم قرار گرفته است. در این مجموعه داده ۸۰ فایل صوتی قرار گرفته است که بین ۴ کلاس مختلف به صورت مساوی پخش شده‌اند. این ۴ کلاس عبارتند از، Joyfully، Sad، Surprised و Euphoric. همچنین یک فایل csv نیز در اختیار ما قرار می‌گیرد که حاوی داده‌های متنی علاوه بر مسیر ذخیره‌سازی هر فایل است.

در بخش پیش‌پردازش نیاز است که به طریقی یک فایل csv را پیمایش کنیم که برای هر کدام از فایل‌ها یک ردیف در دیتاست نهایی ایجاد شود. پس از این مورد، نوبت به پیش‌پردازش هر یک از داده‌های متنی و صوتی می‌رسد. برای این منظور از nltk جهت پیش‌پردازش و نرمال‌سازی داده‌های متنی و از librosa جهت پیش‌پردازش و نرمال‌سازی دادهٔ صوتی استفاده می‌کنیم.

در گام بعدی مدل‌های خود را معرفی می‌کنیم. برای بخش متنی از مدل XLM-RoBERTa-Base و برای بخش صوتی از Wav2Vec2 استفاده می‌کنیم. سپس به ترتیب با استفاده از tokenizer مربوط به مدل متنی و preprocessor مربوط به مدل صوتی، متن و صوت را پیش‌پردازش کرده و داده‌های مورد نیاز برای مدل‌ها را از آن‌ها خارج می‌کنیم.

سپس مدل را به صورتی تعریف می‌کنیم که هر دو مدل‌ها در بدنهٔ مدل وجود داشته باشند، و در لایهٔ نهایی یک لایهٔ خطی با ۴ نورون حضور داشته باشند. بنابراین جهت تعیین logitهای مدل از ترکیبی از خروجی دو مدل استفاده می‌کنیم. از آنجا که شکل دادهٔ خروجی مدل متنی و صوتی با یک متفاوت بوده و دادهٔ صوتی یک لایهٔ بیشتر، یعنی Sequence Length را دارد، نیاز است که در هر دنباله، میانگین تمام اعضا را بدست آوریم تا شکل هر دو داده به شکل (batch size × hidden size) در آید.

در گام نهایی مدل ایجاد شده را با استفاده از داده‌های پیش‌پردازش شده، آموزش می‌دهیم.

# ۷ مراجع


1. ArmanEmo: A Persian Dataset for Text-based Emotion Detection, GitHub Page and Dateset
2. ArmanEmo: A Persian Dataset for Text-based Emotion Detection, Paper
3. Parsivar: Python library for Persian text preprocessing.
4. DadmaTools: A Python NLP Library for Persian
5. Hugging Face – The AI community building the future.
6. XLM-RoBERTa Models, base and large versions
7. The HooshvareLab/roberta-fa-zwnj-base Model
8. Kaggle: Speech Emotion Recognition Voice Dataset