



# تشخیص احساسات در متن

مدرس : استاد الیاس عرب

دانشجو: امیر نوروزی میر فضل اله

## مقدمه

این پروژه برای طبقه‌بندی احساساتی که در داده‌های متنی بیان می‌شوند با استفاده از یادگیری ماشین طراحی شده است. از تکنیک‌های پردازش زبان طبیعی استفاده می‌کند و رابط کاربری ساده‌ای را برای کاربران فراهم می‌کند تا متن خود را وارد کرده و احساس تشخیص داده شده به همراه درصد اطمینان آن را مشاهده کنند. هدف اصلی ایجاد سیستمی مقیاس‌پذیر و قابل اعتماد برای تشخیص احساسات است.

## ویژگی‌های کلیدی

- پیاده‌سازی شده با استفاده از الگوریتم Logistic Regression
- بهره‌گیری از ویژگی‌های استخراج TF-IDF
- تشخیص احساسات احتمالی در آینده
- پشتیبانی از انواع احساسات مختلف مانند خشم، شادی، غم و ...
- وب اپلیکیشن مبتنی بر Streamlit برای تعامل با کاربران
- نمایش احتمال پیش‌بینی‌ها با نمودارهای ستونی با استفاده از Altair
- توانایی آموزش مدل با داده‌های جدید

## ساختار پروژه

```
1. Emotion-Detection
2.   ├── app.py                // نقطه شروع پروژه
3.   ├── data
4.   │   └── emotion_dataset.csv // مجموعه داده برای آموزش
5.   ├── lib
6.   │   ├── main.py           // برنامه اصلی Streamlit
7.   │   ├── model_operation.py // توابع کار با مدل
8.   │   └── train_model.py     // آموزش مدل
9.   └── requirements.txt       // نیازمندی‌های پروژه
```

# نصب

## پیش نیاز

- Python 3.8+
- Pip

## نصب وابستگی ها

برای نصب ابتدا باید یک environment ایجاد کنید:

```
1. python3 -m venv /path/to/virtual/environment
```

سپس دستور زیر را برای فعال کردن venv اجرا کنید:

```
1. source /path/to/virtual/environment
```

حال دستور زیر را برای نصب وابستگی ها اجرا کنید:

```
1. pip install -r requirements.txt
```

## استفاده

### مرحله اول: آموزش مدل

مدل را با استفاده از دستور آموزش دهید:

```
1. python3 /lib/train_model.py
```

با اجرای این دستور، مدل در پوشه models تولید میشود

### مرحله دوم: اجرای برنامه

حال زمان آن رسیده که برنامه را اجرا کنیم. دستور زیر را وارد کنید:

```
1. streamlit run app.py
```

با اجرای این دستور برنامه در مرورگر وب پیش فرض شما باز خواهد شد. می توانید متن وارد کنید و احساس پیش بینی شده به همراه درصد اطمینان و نمودار احتمالات را مشاهده کنید .

## جزئیات پیاده سازی

**Train\_model.py**: این اسکریپت یک مدل رگرسیون لجستیک (مدل خطی است که از یک تابع لجستیک (تابع سیگموید) برای پیش‌بینی احتمال استفاده می‌کند) را روی داده‌های متنی با برچسب‌های احساسی آموزش می‌دهد. از وکتورایزر TF-IDF (Term Frequency-Inverse Document Frequency) برای استخراج ویژگی‌ها استفاده می‌کند و مجموعه داده را به مجموعه‌های آموزشی و آزمایشی تقسیم می‌کند. مدل و وکتورایزر با استفاده از `joblib` برای استفاده بعدی ذخیره می‌شوند.

**model\_operation.py**: یک ماژول کمکی برای مدیریت پیش‌بینی‌ها و احتمالات مدل. مدل از پیش‌آموزش‌دیده و وکتورایزر برای انجام پیش‌بینی‌ها بارگذاری می‌شود.

توابع کلیدی:

- `now_emotion`: احساس متن ورودی را پیش‌بینی می‌کند.
- `future_emotion`: توزیع احتمالات برای تمام احساسات را بازمی‌گرداند.

**app.py**: برنامه اصلی است که رابط کاربری زیبایی را برای کاربران فراهم می‌آورد تا متن خود را وارد کرده و پیش‌بینی‌ها را مشاهده کنند. از توابع ماژولار `model_operation.py` برای پیش‌بینی و احتمالات استفاده می‌کند.

ویژگی‌ها:

- `main()`: مدیریت تعاملات کاربر، پیش‌بینی‌ها و بصری‌سازی در فایل `main.py`
- ایموجی‌ها تجربه کاربری را با نمایش بصری هر احساس بهبود می‌بخشند.

# Emotion Detection in Text

Type Here:

my mom is dead today

Submit

Original Text

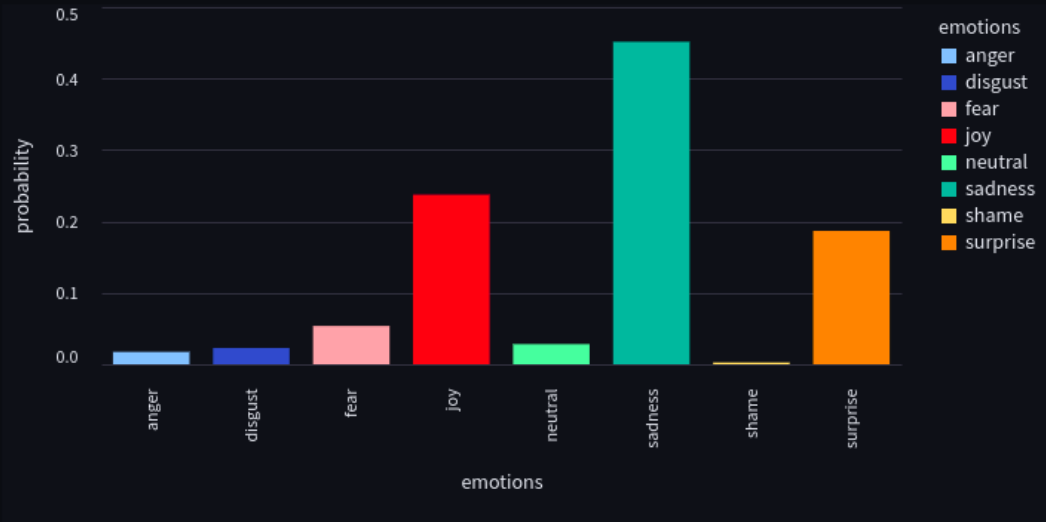
my mom is dead today

Prediction

sadness: 😞

Confidence: 0.45

Prediction Probability



## نتایج

در آزمایشات انجام شده، مدل توانست دقت کلی ۹۰٪ را در تشخیص احساسات از داده‌های متنی به دست آورد. عملکرد مدل بر اساس چندین معیار ارزیابی شد:

مقدار	معیار
۹۰٪	دقت (Accuracy)
۹۲٪	دقت پیش‌بینی (Precision)
۹۰٪	F1 امتیاز

## نتیجه‌گیری

این پروژه نشان داد که مدل‌های یادگیری ماشین با استفاده از داده‌های متنی و تکنیک‌های پردازش زبان طبیعی می‌توانند با دقت بالایی احساسات را تشخیص دهند. استفاده از رگرسیون لجستیک و ویژگی‌های استخراج شده توسط TF-IDF، ترکیب موثری برای این کار بود. با این حال، محدودیت‌هایی نیز وجود دارد:

- دقت مدل به کیفیت داده‌های آموزشی وابسته است.
  - احساسات پیچیده‌تر ممکن است به مدل‌های پیشرفته‌تری نیاز داشته باشند.
- در آینده، می‌توان این پروژه را با استفاده از مدل‌های پیشرفته‌تر مانند BERT بهبود بخشید و قابلیت تشخیص احساسات در زبان‌های مختلف را به آن اضافه کرد.

## تقدیر و تشکر

تشکر ویژه از جامعه متن‌باز و توسعه‌دهندگان کتابخانه‌هایی مانند Scikit-learn، Streamlit و Altair که این پروژه را ممکن کردند.

1. Scikit-learn: Documentation and Tutorials - <https://scikit-learn.org>
2. Streamlit: Open-source app framework - <https://streamlit.io>
3. Altair: Declarative Visualization in Python - <https://altair-viz.github.io>
4. TF-IDF Theory: Understanding Text Vectorization - <https://en.wikipedia.org/wiki/Tf-idf>