گزارش پروژه دوم
گروه ششم

اعضا :
فاطمه خجسته
مانا سامانی
مهدی اسدی
آناهیتا الیاسی

بوتکمپ هوش مصنوعی دلتا کوئرا

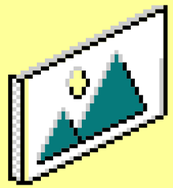
11:11PM


Agenda
Start
پوشش پلاک خودرو
تحلیل احساس نظرات

پوشش پلاک خودرو

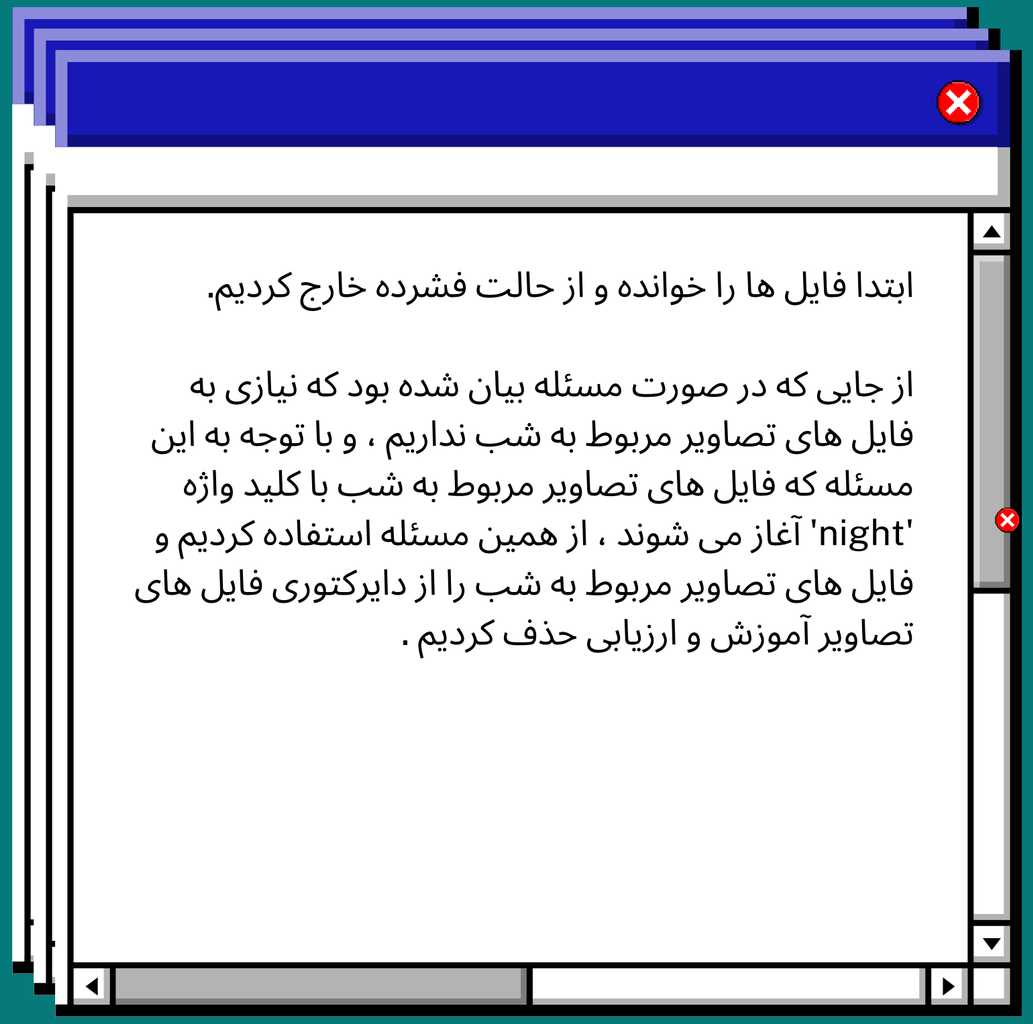
License plate cover

Back to Agenda Page



پوشش پلاک خودرو

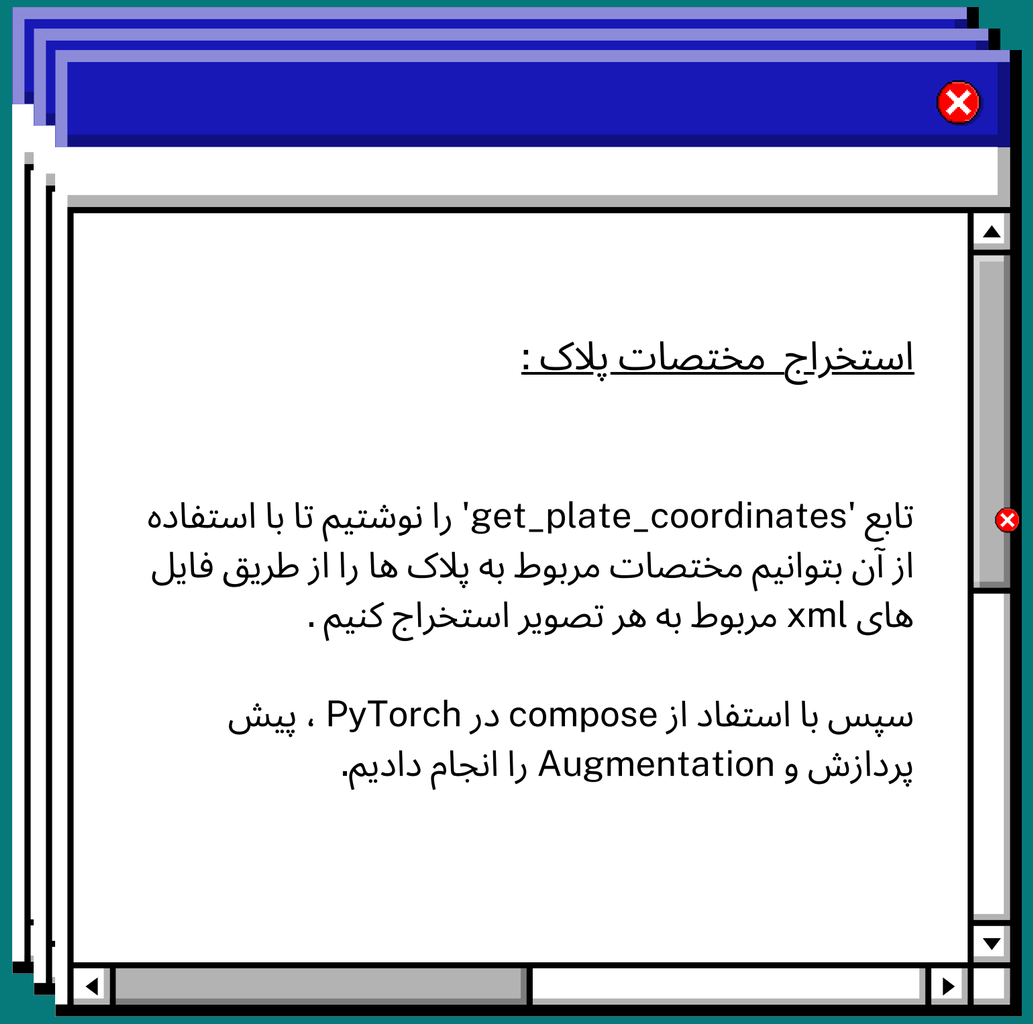
loading data & preprocessing



Back to Agenda Page

پوشش پلاک خودرو

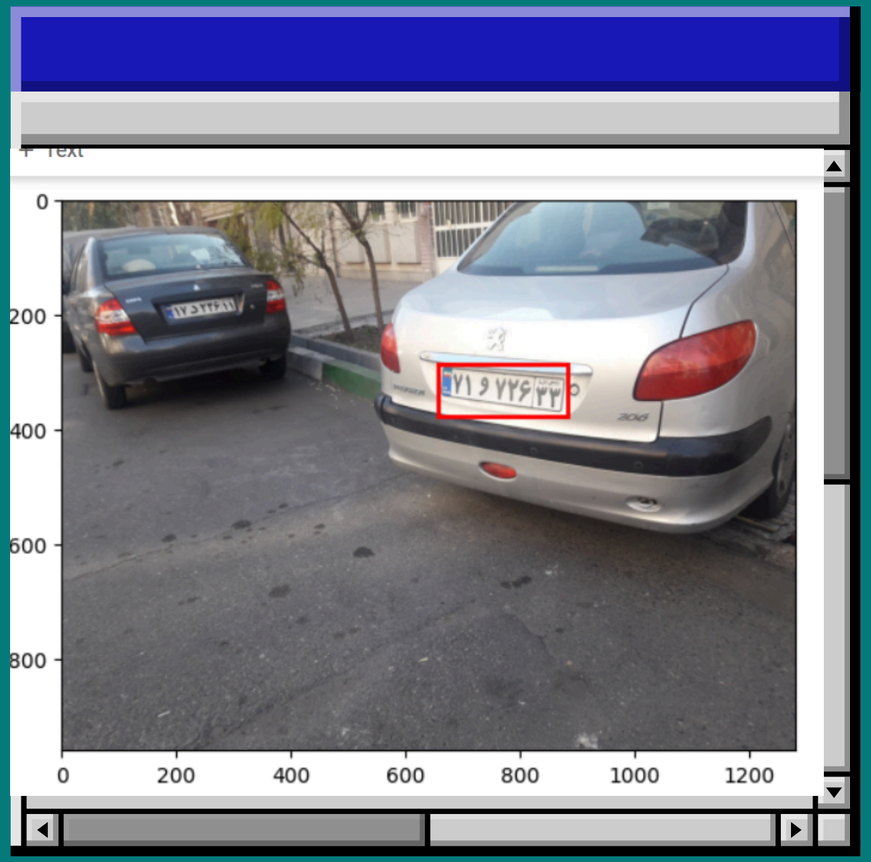
loading data & preprocessing



Back to Agenda Page

پوشش پلاک خودرو

loading data & preprocessing



تابع 'show_image_with_bounding_box' را نوشتیم که با دریافت ورودی های آدرس دایرکتوری تصویر و مختصات پلاک خروجی از تابع 'get_plate_coordinates' بتواند پلاک را روی تصویر مربوطه نمایش بدهد.

Back to Agenda Page

پوشش پلاک خودرو

loading data & preprocessing

Back to Agenda Page

پوشش پلاک خودرو

loading data & preprocessing

در ادامه مدل از پیش آموزش دیده YOLO موجود در کتابخانه Torch را بارگذاری کردیم و تنظیمات مربوط به آنرا انجام دادیم.
با استفاده از بهینه ساز 'Adam' و محاسبه loss از روش 'CrossEntropy' مدل را با کمک یک حلقه آموزش دادیم.
پس از اتمام آموزش مدل ، آنرا ذخیره می کنیم تا بتوانیم آنرا برای استفاده ی مجدد در دسترس داشته باشیم.
برای انجام ادامه مراحل مدل ذخیره شده را لود و بارگذاری می کنیم.

Back to Agenda Page

پوشش پلاک خودرو

loading data & preprocessing

۱- مشخصات YOLOv3 در PyTorch:

نام مدل: YOLOv3 (You Only Look Once)

وظیفه: تشخیص اشیا در تصاویر

سرعت و دقت: تعادل بین دقت بالا و سرعت زیاد (به‌ویژه برای پردازش هم‌زمان تصاویر و ویدیوها)

معماری: شبکه‌ای با چندین لایه کانولوشن (CNN) که به بخش‌های مختلف تصویر نگاه می‌کند تا اشیا را شناسایی کند.

Back to Agenda Page

پوشش پلاک خودرو

loading data & preprocessing

قابلیت‌ها: شناسایی اشیا در سایزهای مختلف در تصویر با دقت نسبتا بالا، حتی در شرایط مختلف نور و پس‌زمینه.

نحوه کار YOLO3 در Torch :
YOLOv3 .مرحله اول: تصویر به مدل داده می‌شود به جای تقسیم تصویر به بخش‌های کوچک، کل تصویر را هم‌زمان پردازش می‌کند.

مثلاً مرحله دوم: مدل به بخش‌های مختلف تصویر نگاه می‌کند تا بتواند اشیا [۵۲×و ۵۲ ۲۶×۲۶ ،۱۳×۱۳ را در سایزهای مختلف پیدا کند.

Back to Agenda Page

پوشش پلاک خودرو

loading data & preprocessing

مرحله سوم: برای هر بخش، مدل تعدادی جعبه (Bounding Box) پیشنهاد می‌دهد که ممکن است محل اشیا باشند. هر جعبه شامل مختصات، نوع شی (مثل ماشین یا آدم) و میزان اطمینان مدل به شناسایی است.

مرحله آخر: YOLOv3 با حذف جعبه‌های تکراری، فقط جعبه‌های با احتمال بالا را نگه می‌دارد و مکان اشیا را در تصویر برمی‌گرداند.

Back to Agenda Page

پوشش پلاک خودرو

loading data & preprocessing

در ادامه تابع 'evaluate_model' را برای مشاده پیش بینی و ارزیای مدل در زمینه تشخیص پلاک نوشتیم.

·متاسفانه مدل پس از تست های متوالی و تغییرات مختلف و تلاش برای بهینه سازی و تغییر و تغییر در نحوه آموزش ، نمی تواند به درستی پلاک ها را تشخیص بدهد و دارای مشکل و خطا می باشد !

Back to Agenda Page

· NLP · NLP · NLP · NLP · NLP ·NLP · NLP · NLP · NLP · NLP · NLP ·NLP · NLP · NLP · NLP · NLP · NLP ·NLP · NLP · NLP · NLP · NLP ·

تحلیل احساس نظرات

NLP

Back to Agenda Page

· NLP · NLP · NLP · NLP · NLP ·NLP · NLP · NLP · NLP · NLP · NLP ·NLP · NLP · NLP · NLP · NLP · NLP ·NLP · NLP · NLP · NLP · NLP ·

NLP

بخش اول

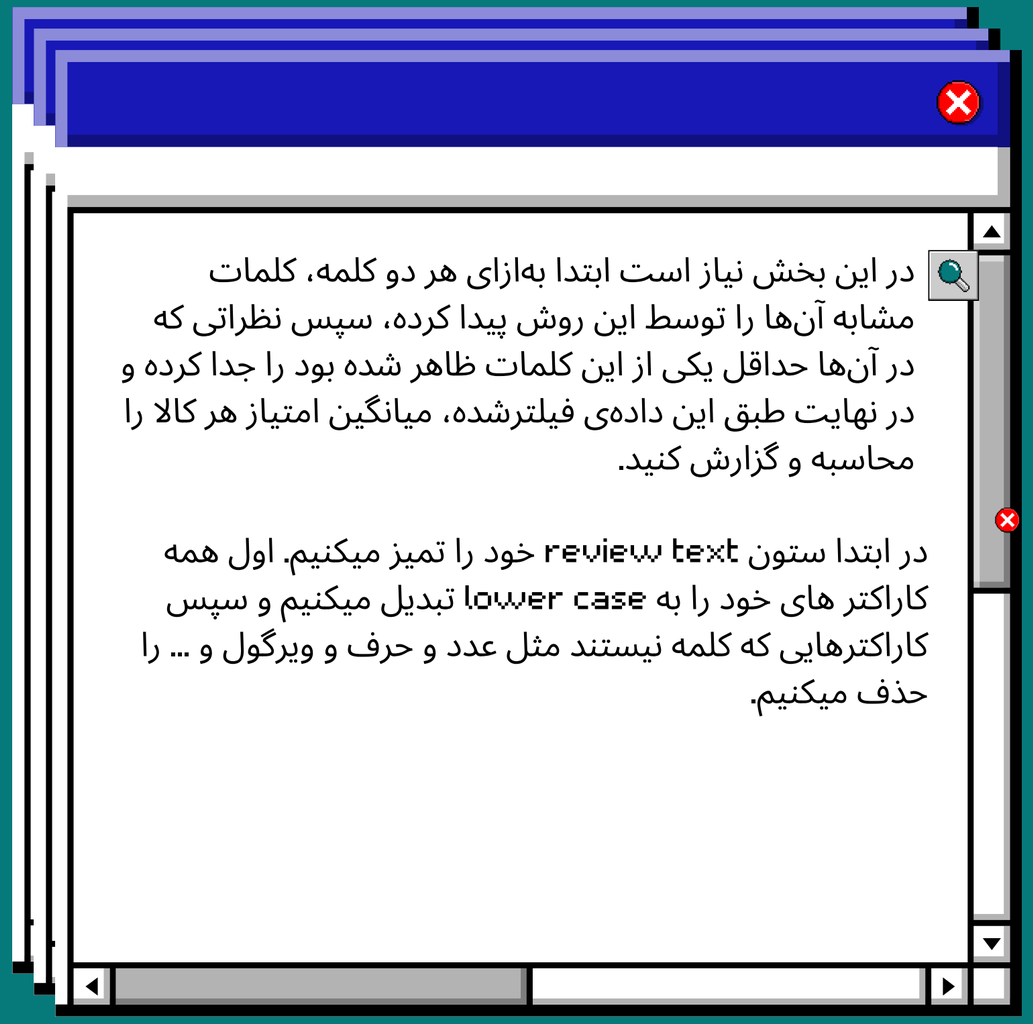
تجزیه و تحلیل اولیه از داده‌ها

Back to Agenda Page

NLP

بخش دوم

میزان رضایت از یک جنبه‌ی مشخص



در این بخش نیاز است ابتدا به‌ازای هر دو کلمه، کلمات مشابه آن‌ها را توسط این روش پیدا کرده، سپس نظراتی که در آن‌ها حداقل یکی از این کلمات ظاهر شده بود را جدا کرده و در نهایت طبق این داده‌ی فیلترشده، میانگین امتیاز هر کالا را محاسبه و گزارش کنید.

در ابتدا ستون review text خود را تمیز میکنیم. اول همه کاراکتر های خود را به lower case تبدیل میکنیم و سپس کاراکترهایی که کلمه نیستند مثل عدد و حرف و ویرگول و ... را حذف میکنیم.

Back to Agenda Page

NLP

بخش دوم
میزان رضایت از یک جنبه‌ی مشخص

مدل سازی:

در بخش مدل سازی از کتابخانه gensim که داری مدل Word2Vec است استفاده میکنیم.

میکنیم. یعنی هر جمله را tokenize خود را review text ابتدا ستون به کلمه های مختلف تقسیم میکنیم. یعنی یک آرایه از کلمات را ما داریم.

بعد پارامتر های مدلی خود را تعریف کرده ایم.

پارامتر vector size : تعداد ویژگی ها را در بردار های کلمات را مشخص می‌کند.

پارامتر window: اندازه فاصله را تنظیم می‌کند که چند کلمه اطراف هر کلمه هدف هنگام یادگیری روابط بین کلمات در نظر گرفته شود.

پارامتر min count: کلماتی که کمتر از این مقدار ظاهر شده اند را نادیده می‌گیرد و نویز کلمات نادر را کاهش می‌دهد.

سپس مدل خود را با این هایپرپارامتر ها آموزش میدهیم و در لیست ذخیره میکنیم.

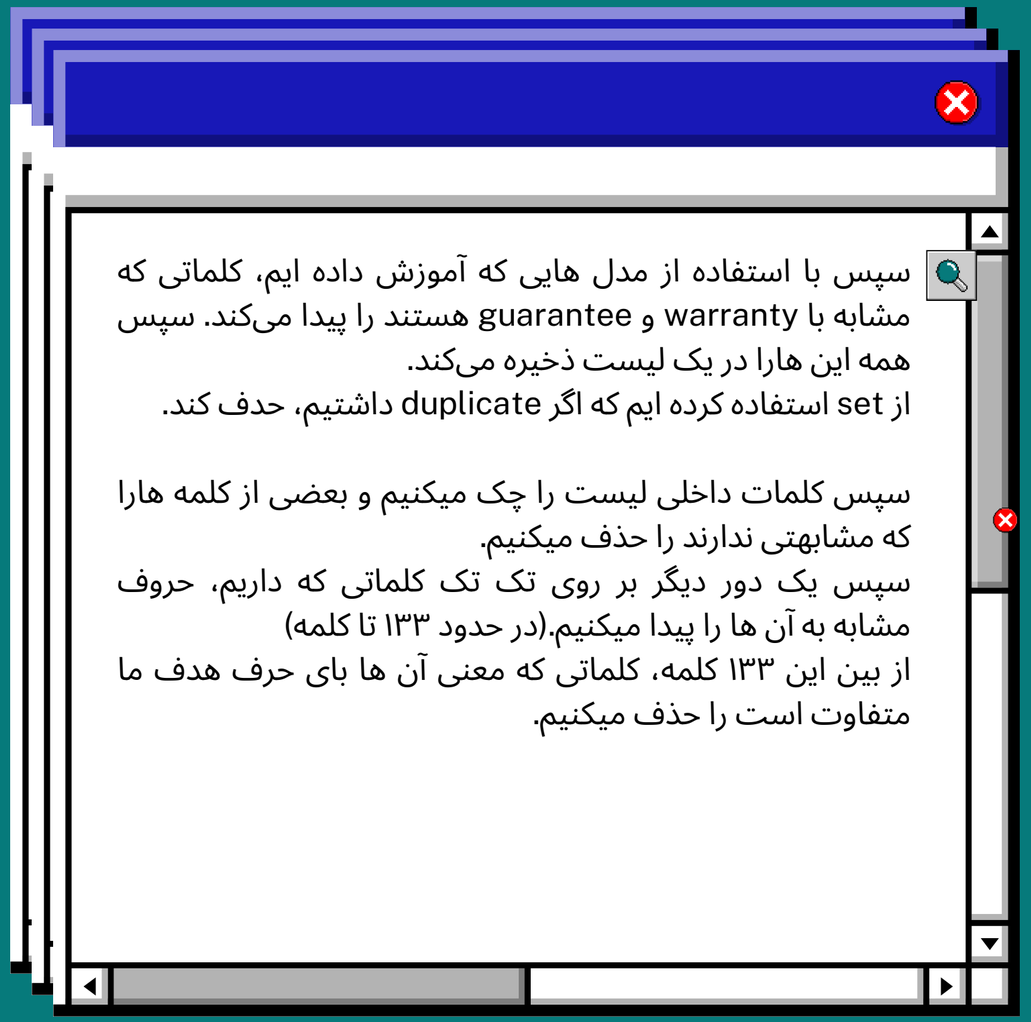
مدل word2vec روابط بین کلمات را بر اساس زمینه تعریف شده توسط پارامتر ها یادمیگیرد.

Back to Agenda Page

NLP

بخش دوم

میزان رضایت از یک جنبه‌ی مشخص



سپس با استفاده از مدل هایی که آموزش داده ایم، کلماتی که مشابه با warranty و guarantee هستند را پیدا می‌کند. سپس همه این هارا در یک لیست ذخیره می‌کند.
از set استفاده کرده ایم که اگر duplicate داشتیم، حدف کند.

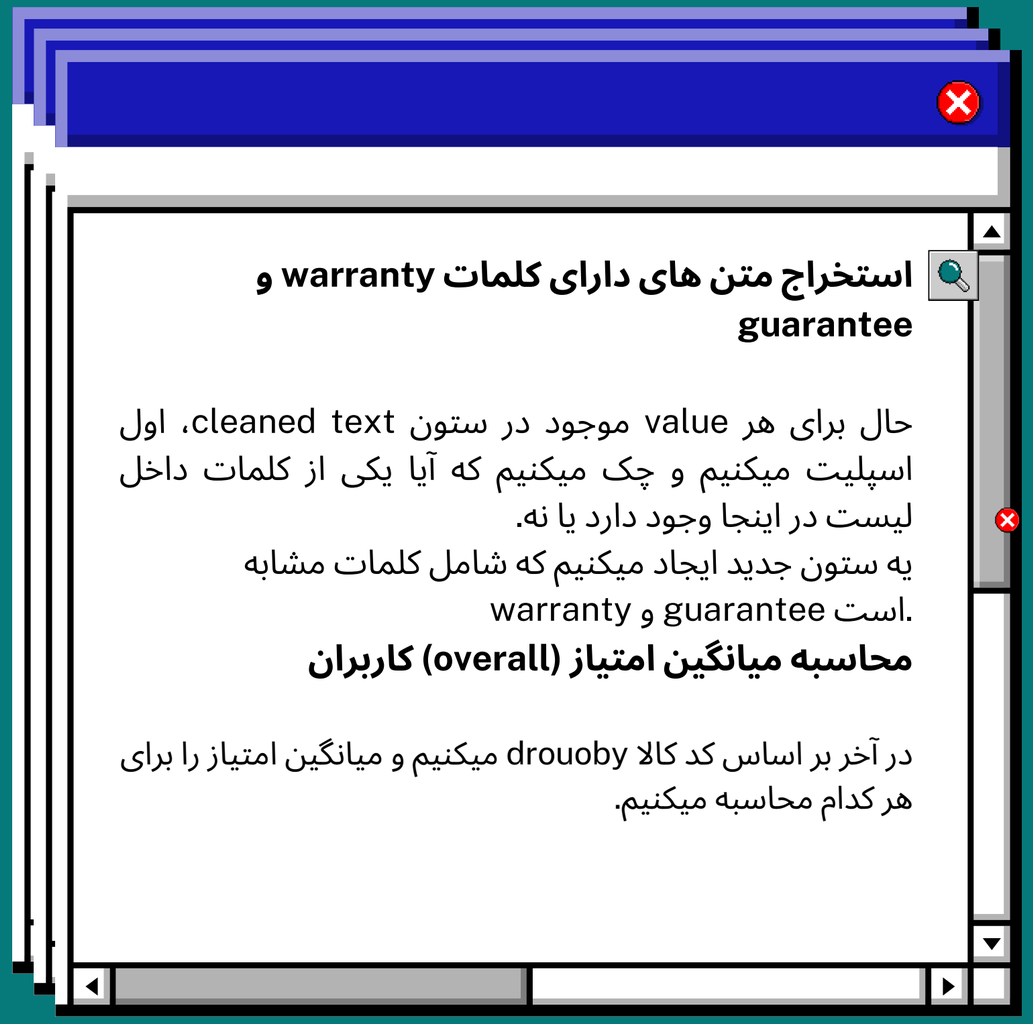
سپس کلمات داخلی لیست را چک میکنیم و بعضی از کلمه هارا که مشابهتی ندارند را حذف میکنیم.
سپس یک دور دیگر بر روی تک تک کلماتی که داریم، حروف مشابه به آن ها را پیدا میکنیم.(در حدود ۱۳۳ تا کلمه)
از بین این ۱۳۳ کلمه، کلماتی که معنی آن ها بای حرف هدف ما متفاوت است را حذف میکنیم.

Back to Agenda Page

NLP

بخش دوم

میزان رضایت از یک جنبه‌ی مشخص



استخراج متن های دارای کلمات warranty و guarantee

حال برای هر value موجود در ستون cleaned text، اول اسپلیت میکنیم و چک میکنیم که آیا یکی از کلمات داخل لیست در اینجا وجود دارد یا نه.
یه ستون جدید ایجاد میکنیم که شامل کلمات مشابه warranty و guarantee است.

محاسبه میانگین امتیاز (overall) کاربران

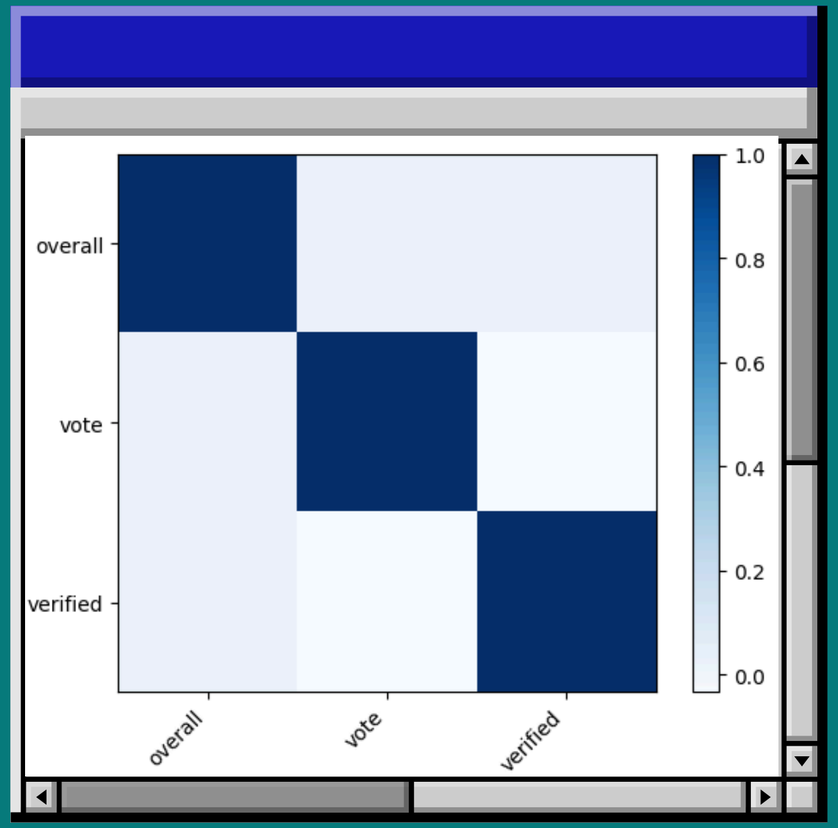
در آخر بر اساس کد کالا drouoby میکنیم و میانگین امتیاز را برای هر کدام محاسبه میکنیم.

Back to Agenda Page

NLP

بخش سوم

مدل تحلیل احساس



نیاز است مدلی طراحی کنید که با دریافت متن نظر کاربر، احساس رضایت وی نسبت به کالا را بین عددی از ۱ تا ۵ تعیین کند.

مهندسی ویژگی :

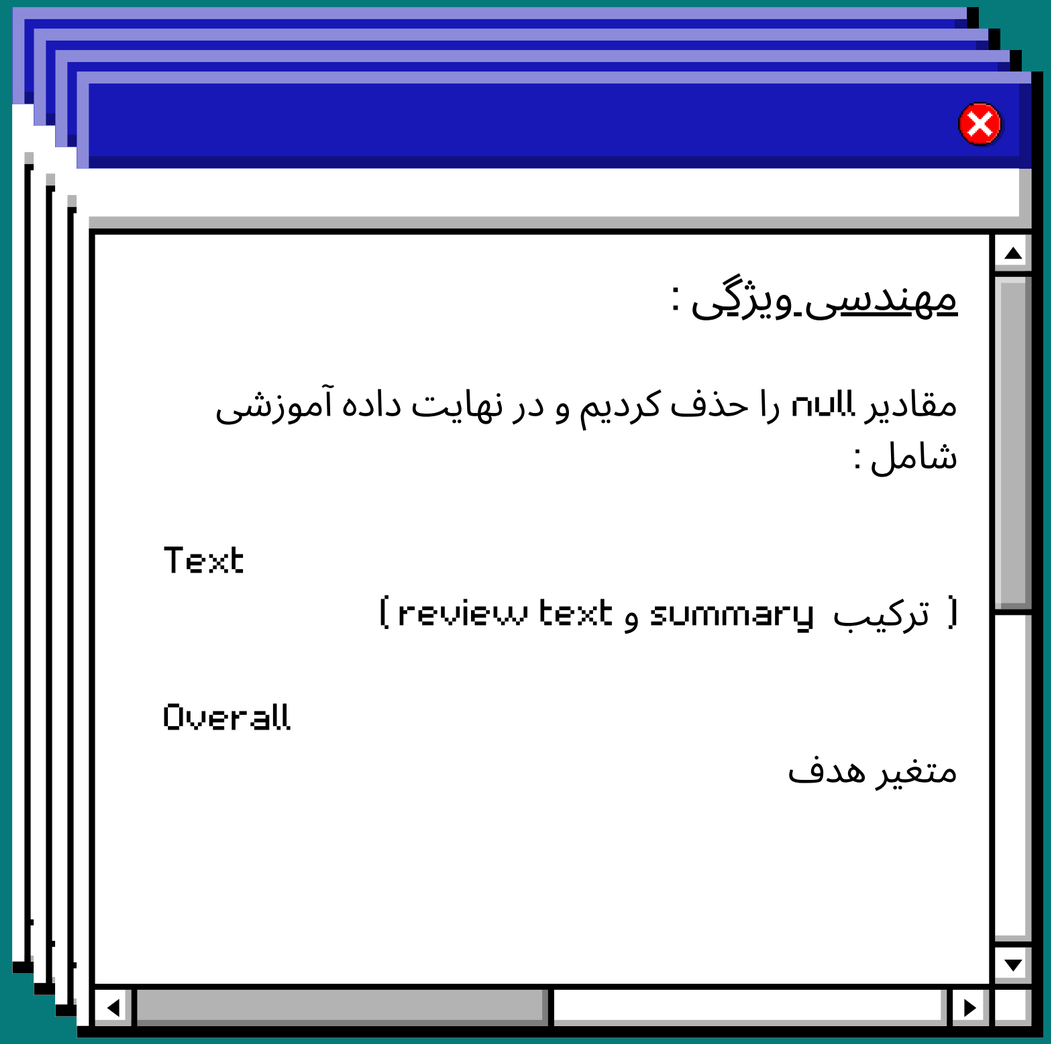
- ستون style به دلیل پیچیدگی و محتوای کم از دیتای آموزش حذف شد
- طبق تحقیقاتی که انجام شد، ستون های asin, title, brand مفید نبودند و حذف شدند
- بررسی ارتباط ستون های verified,overall, vote از نظر همبستگی نشان داد به دلیل همبستگی بسیار پایین با overall ، دو ستون دیگر باید حذف می‌شدند.

Back to Agenda Page

NLP

بخش سوم

مدل تحلیل احساس



Back to Agenda Page

NLP

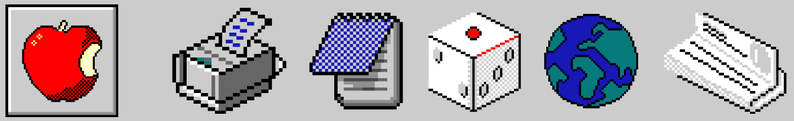
بخش سوم

مدل تحلیل احساس

```
words_to_keep = {
    "good", "bad", "perfect",'great',
"poor","nice", "amazing", "disappointing", ....


ontractions = {
  "can't": "cannot", "won't": "will not", "n't": "
not", "'re": " are",...

important_stopwords = {
"not", "no", "never", "very", "all"...

negation_words = {"not", "no", "never",
"none", "nothing",...
```

پردازش متن :

در این قسمت با استفاده از کتابخانه‌های pandas و nltk، نظرات کاربران از فایل داده بارگذاری شده و در قالب یک ستون متنی ترکیب شدند. با اعمال یک تابع پیش‌پردازش، متن به حروف کوچک تبدیل شده، اختصارات به صورت کامل نوشته شدند، نشانه‌گذاری‌های غیرضروری حذف شد، و کلمات پرتکرار کاهش یافتند.



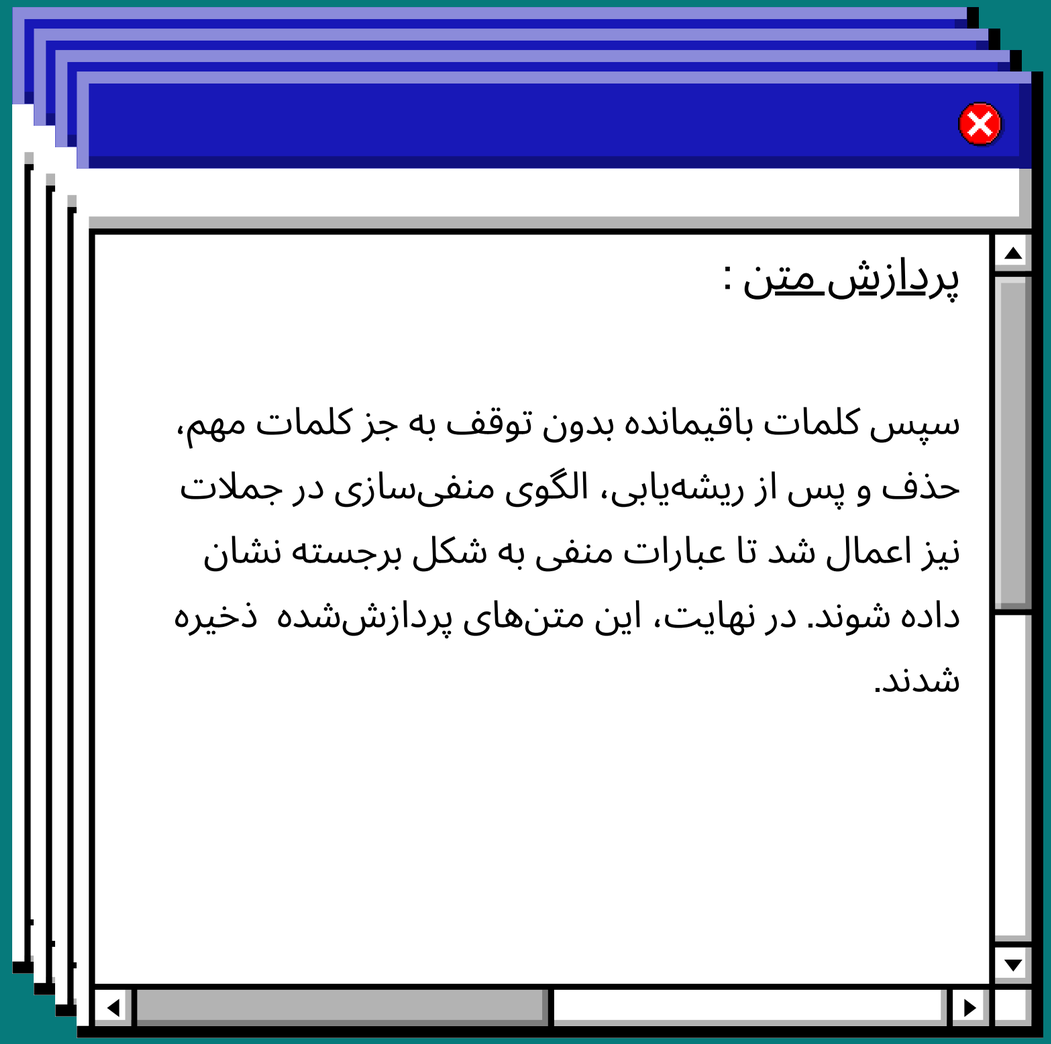
Back to Agenda Page

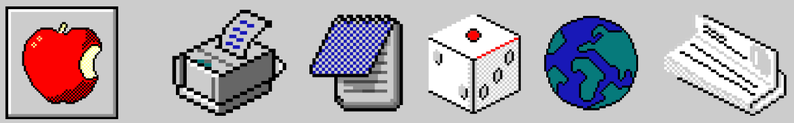
NLP

بخش سوم

مدل تحلیل احساس

```
words_to_keep = {
    "good", "bad", "perfect",'great',
"poor","nice", "amazing", "disappointing", ....


ontractions = {
  "can't": "cannot", "won't": "will not", "n't": "
not", "'re": " are",...

important_stopwords = {
"not", "no", "never", "very", "all"...

negation_words = {"not", "no", "never",
"none", "nothing",...
```





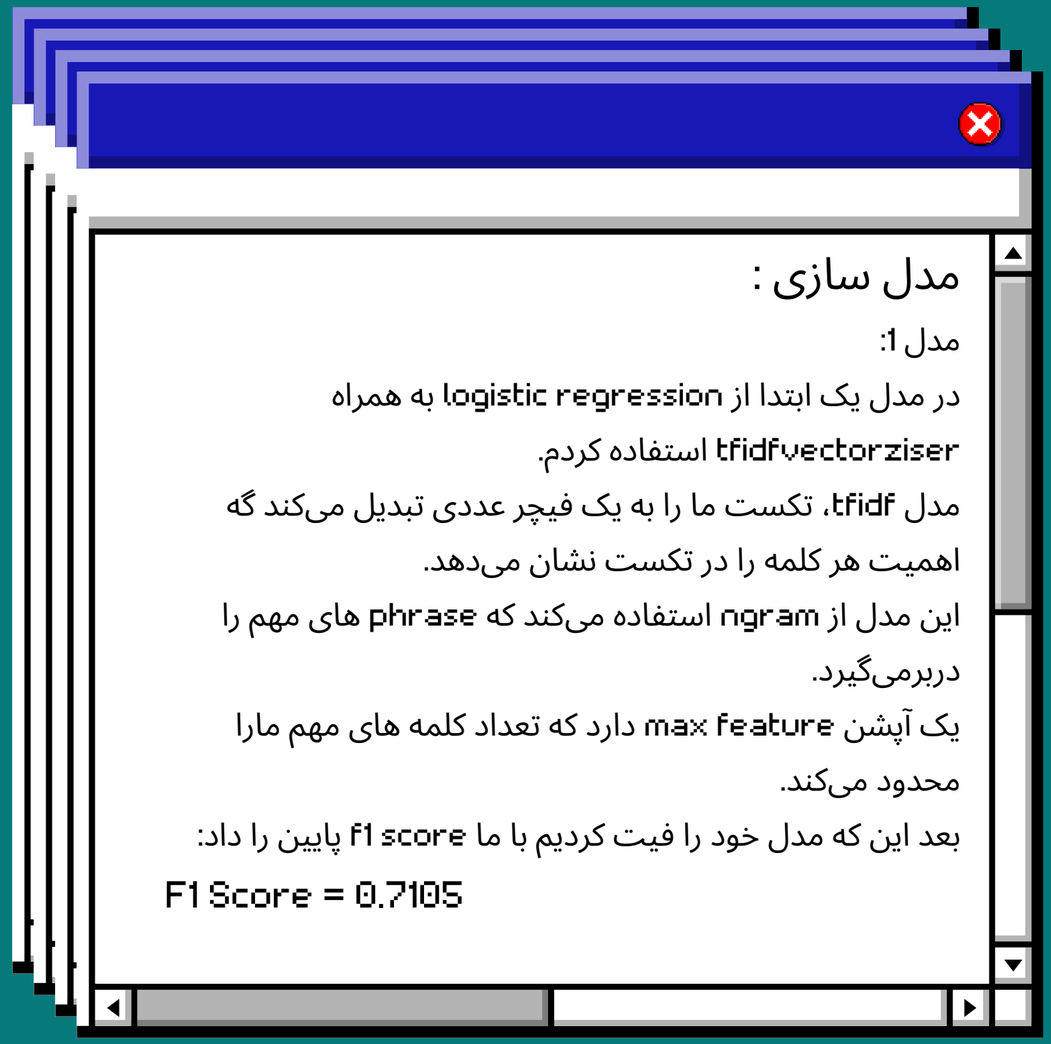
Back to Agenda Page

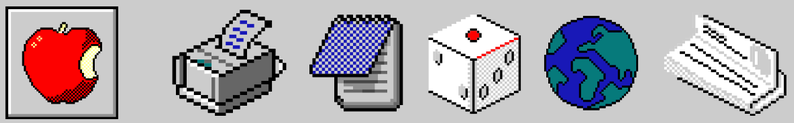
NLP

بخش سوم

مدل تحلیل احساس

```
pipeline = Pipeline([
  ('tfidf',
TfidfVectorizer(max_features=20000,
ngram_range=(1, 3))),
  ('clf',
LogisticRegression(multi_class='multinomial',
solver='lbfgs', max_iter=5000))
])
```





Back to Agenda Page

NLP

بخش سوم

مدل تحلیل احساس

```
pipeline4 = Pipeline([
  ('count_vec',
CountVectorizer(max_features=20000,
ngram_range=(1, 3))),
  ('clf',
LogisticRegression(multi_class='multinomial',
solver='lbfgs', max_iter=5000))
])
```

مدل سازی :

مدل 2:

در مدل دوم به جای tf idf از count vectorzise استفاده کرده ایم.

این مدل تکست هادا تبدیل به یک فیچر عددی می‌کند که با استفاده شمردن تعداد frequency هر کلمه است.

بعد این که مدل خود را فیت کردیم با ما f1 score پایین را داد:

F1 Score = 0.7014

Back to Agenda Page

NLP

بخش سوم
مدل تحلیل احساس

Example:

Let's say we have two short documents:

- Document 1: "The cat sat on the mat."
- Document 2: "The cat lay on the rug."

Using CountVectorizer, we get a matrix like this:

	the	cat	sat	on	mat	lay	rug
Doc 1	2	1	1	1	1	0	0
Doc 2	2	1	0	1	0	1	1

Each cell represents how many times each word appears in each document.

مدل CountVectorizer:

دیتا رو به یک ماتریس از توکن ها تبدیل می‌کند. با ایجاد یه ماتریس که هر سطر مربوط به یک تکست است و هر ستون مربوط به توکن های ما است.

برای مثلا برای جمله:

The cat sat on the mat

توکن های ما میشود:

["the", "cat", "sat", "on", "the", "mat"]

حال به مثال روبرو توجه کنید:

میبینیم که ستون های همان کلمات unique در متن هستند و سطر های ما هم تکست های ما.
به طور مثال برای سطر یک برای هر ستون تعداد دفعاتی که کلمه ظاهر شده است را قرار می‌دهد.

Back to Agenda Page

NLP

بخش سوم
مدل تحلیل احساس

مدل TfidfVectorizer:

دیتا رو به یک ماتریکس تبدیل می‌کند ولی یه وزن به هر یک از کلمات ما می‌دهد که مشخص کند کدام مهم تر است.

به جای فقط شمارش کلمات، شمارش رو بر اساس تعداد دفعات ظاهر شدن کلمه در هم تکست ها تنظیم می‌کند و وزن کلمات رایج رو کم و کلمات تادر رو بیشتر می‌کند.

برای مثال قبلی ماتریس به صورت روبرو میشود

Example:

Using the same documents as before:

	the	cat	sat	on	mat	lay	rug
Doc 1	0.3	0.4	0.5	0.3	0.5	0	0
Doc 2	0.3	0.4	0	0.3	0	0.5	0.5

Here, words like "the" and "on" get lower scores because they appear in both documents, making them less unique. Words like "mat" and "rug," which are unique to each document, get higher scores.

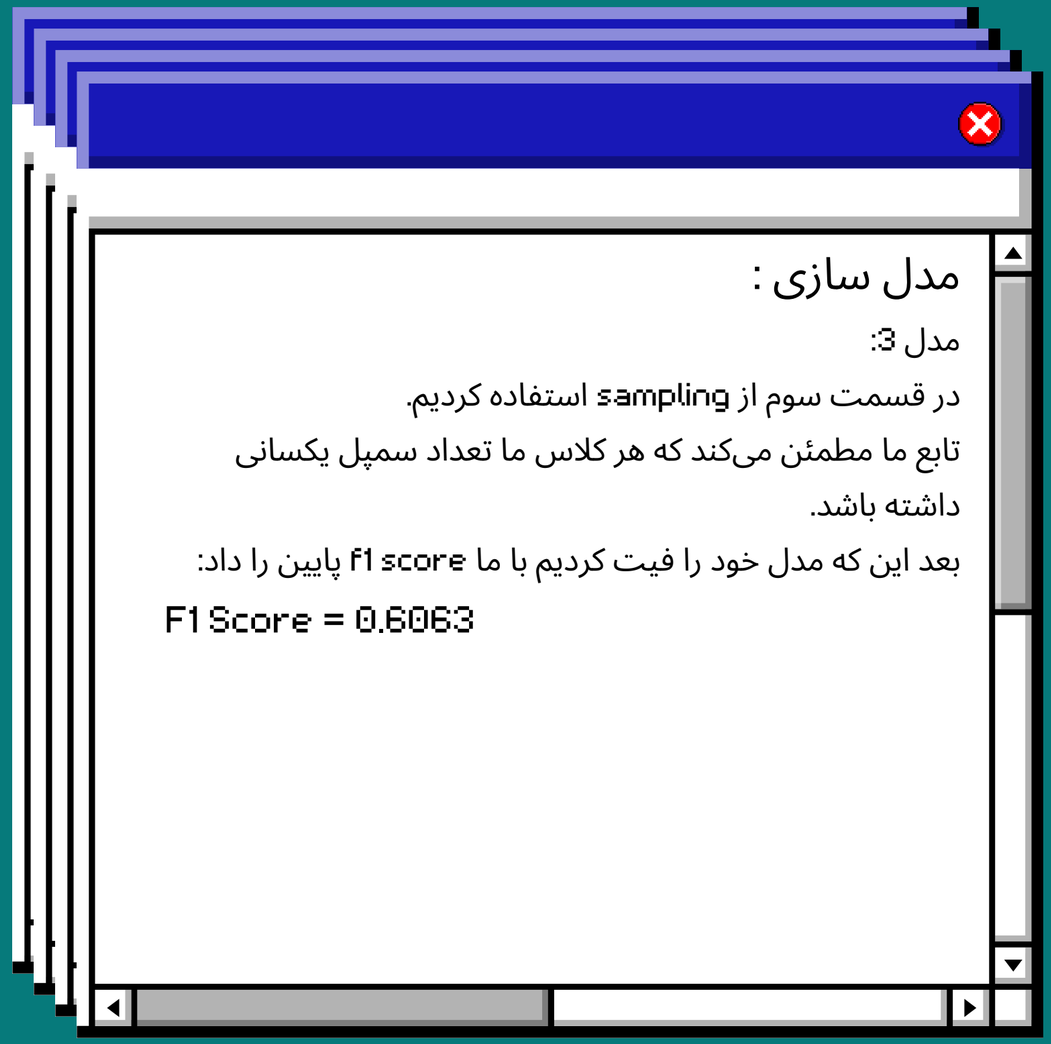
Back to Agenda Page

NLP

بخش سوم

مدل تحلیل احساس

```
def sampling(df, num_target,
each_class_size):
  min_class_size = min(each_class_size,
df[num_target].value_counts().min())
  balanced_df = pd.concat([
    df[df[num_target] ==
sentiment].sample(min_class_size,
replace=True)
    for sentiment in df[num_target].unique()
  ])
  return
balanced_df.sample(frac=1).reset_index(drop
=True)
```



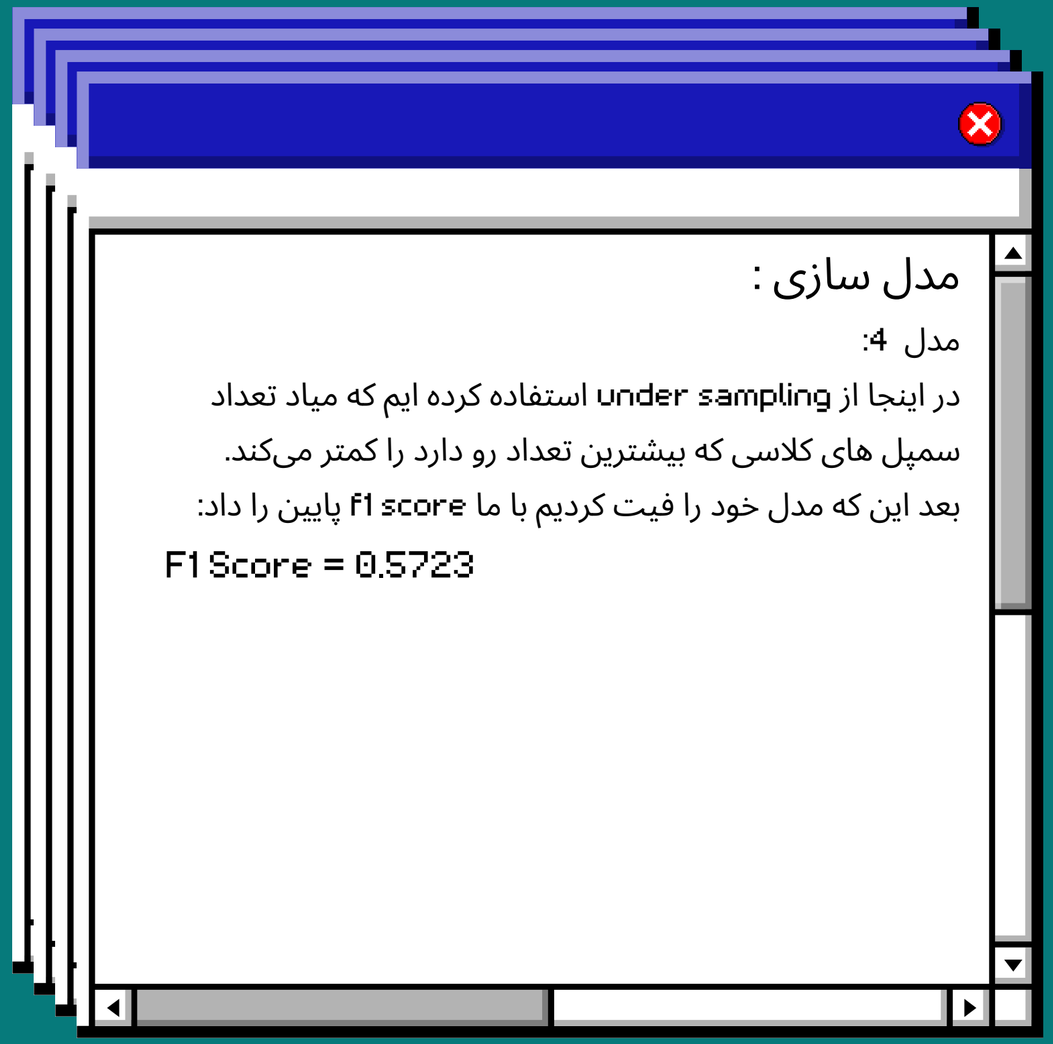
Back to Agenda Page

NLP

بخش سوم

مدل تحلیل احساس

```
undersampler =
RandomUnderSampler(random_state=42)
X_resampled, y_resampled =
undersampler.fit_resample(X.to_frame(), y)
```



Back to Agenda Page


Thank you!