

**درس پردازش زبان طبیعی**

**استاد ممتازی**

**نیما پری فرد**

**402131017**

فهرست تمرین اول پردازش زبان طبیعی

[بخش 1 3](#_Toc163759974)

[مشاهده داده و اعمال پیش پردازش های لازم 3](#_Toc163759975)

[بخش 2 4](#_Toc163759976)

[تعاریف 4](#_Toc163759977)

[پیاده سازی هموار سازی backoff و absolute 8](#_Toc163759978)

[تحلیل و گزارش نتایج 9](#_Toc163759979)

[بخش الف) 9](#_Toc163759980)

[بخش ب) 13](#_Toc163759982)

[بخش ج) 16](#_Toc163759983)

[بخش 3 17](#_Toc163759984)

[تعاریف 17](#_Toc163759985)

[پیاده سازی میانگین حسابی word2vec 18](#_Toc163759986)

[پیاده سازی وزنی word2vec با وزن های TF-IDF 21](#_Toc163759987)

[تحلیل نتایج 24](#_Toc163759988)

[بخش 4 26](#_Toc163759989)

[پیاده سازی perplexity 26](#_Toc163759990)

[پیاده سازی accuracy و f1 score در حالت چند کلاسه 28](#_Toc163759991)

# بخش 1

## مشاهده داده و اعمال پیش پردازش های لازم

A blurry image of a computer keyboard

Description automatically generated

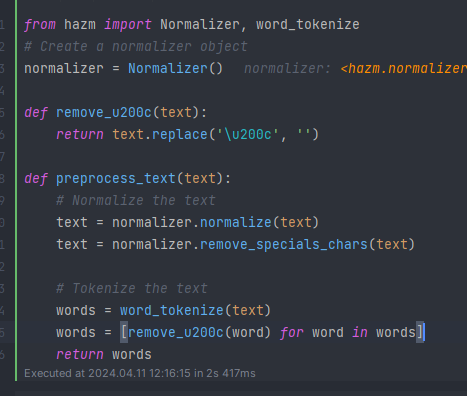
داده قرار داده شده از دو ستون تشکیل شده است که یکی شامل متن مورد نظر و دیگری دارای برچسب مورد نظر است.

ساختار سه داده به این صورت است:

A screen shot of a computer

Description automatically generated

به علت فارسی بودن متن از کتاب خانه Hazm برای پیش پردازش ممتون استفاده کردم.



# بخش 2

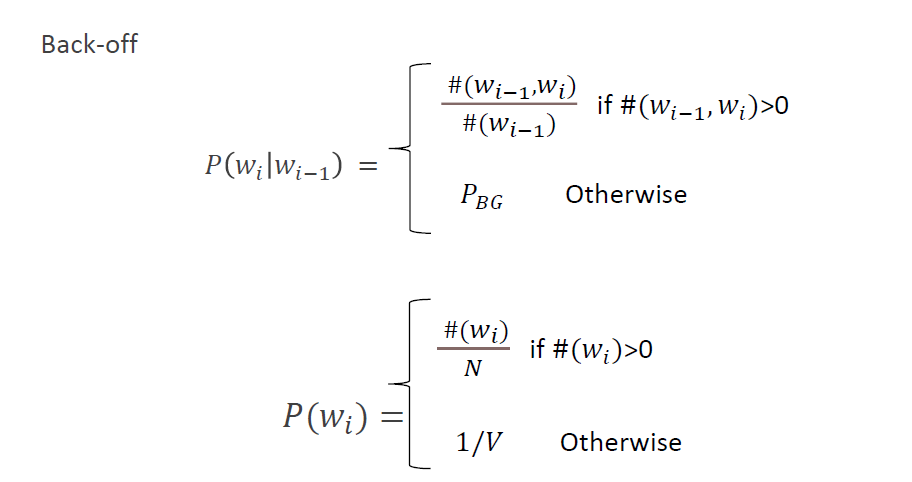
## تعاریف

مدل‌های زبانی n-gram روشی برای پیش‌بینی احتمال ظاهر شدن یک کلمه در متن بر اساس کلماتی که قبل از آن آمده‌اند هستند. تصور کنید که شما سعی دارید کلمه بعدی در یک جمله را حدس بزنید. یک مدل n-gram یک دنباله از n کلمه (که n هر عددی می‌تواند باشد) را در نظر می‌گیرد و تجزیه و تحلیل می‌کند که چه اندازه این دنباله در یک نمونه متن بزرگ ظاهر می‌شود. این به آن کمک می‌کند تا احتمالات مختلف کلمات بعدی را تخصیص دهد.

Uni-gram, Bi-gram, and Tri-gram Model


مدل‌های زبانی n-gram، با وجود کاربردی که دارند، ممکن است با مشکلاتی مواجه شوند هنگامی که با کلمات یا عباراتی غیر دیده شده سر و کار دارند. این به دلیل این است که آن‌ها بر اساس تعداد ظاهر شدن دنباله‌ها در داده‌های آموزشی حساب می‌کنند. اگر یک دنباله خاص قبلاً مشاهده نشده باشد، مدل احتمال آن را صفر می‌دهد، که هرگونه پیش‌بینی مربوط به آن دنباله را غیرممکن می‌کند.

تکنیک‌های هموارسازی این مشکل را با تنظیم احتمالات اختصاص داده شده توسط مدل‌های n-gram حل می‌کنند.



A screenshot of a computer

Description automatically generated

پیاده سازی مدل زبانی n-grams

در کلاس زیر مدل پایه n-grams را پیاده سازی کردم.

A computer screen shot of a program code

Description automatically generated



در متد بالا در کلاس n-grams دیکشنری تشکیل دادم و با توجه به شماره n رشته ها با طول n و n-1 به دست آوردم تا برای محاسبه احتمال از آن استفاده کنم.

این قسمت با داده های آموزش انجام شد.

A screen shot of a computer code

Description automatically generated

در این قسمت احتمال را جهت استفاده در مدل زبانی حساب کردم البته در کد های هموارسازی قسمت بعد هر دو بازنویسی شده اند.

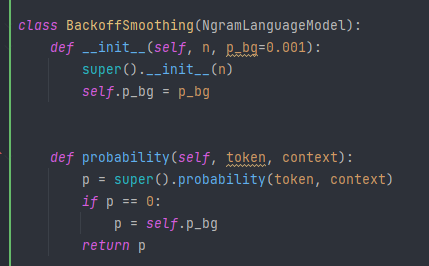
A computer screen shot of text

Description automatically generated

از این کد برای محاسبه perplexity استفاده کردم.

قبل از بررسی نتایج باید عرض کنم مدل های زبانی با داده آموزش آموزش دیده شده اند و کد آن در قسمت زیر است.

## پیاده سازی هموار سازی backoff و absolute



در این جا هموارسازی back off پیاده سازی کردم و از کلاس ngram قبلی ارث بری کردم.

A screen shot of a computer code

Description automatically generated

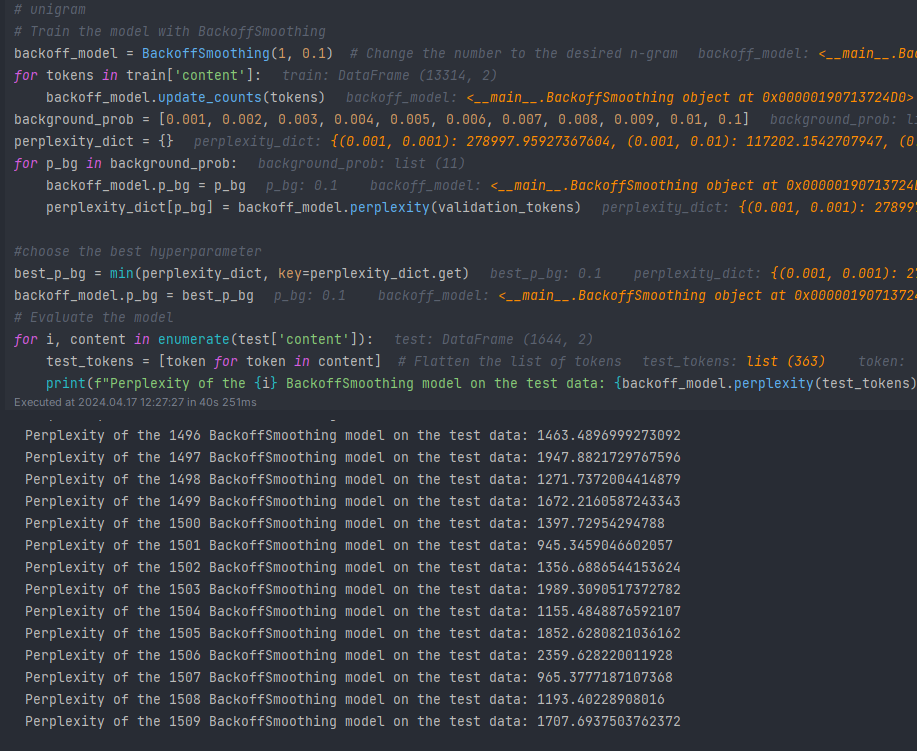
## تحلیل و گزارش نتایج

### بخش الف)

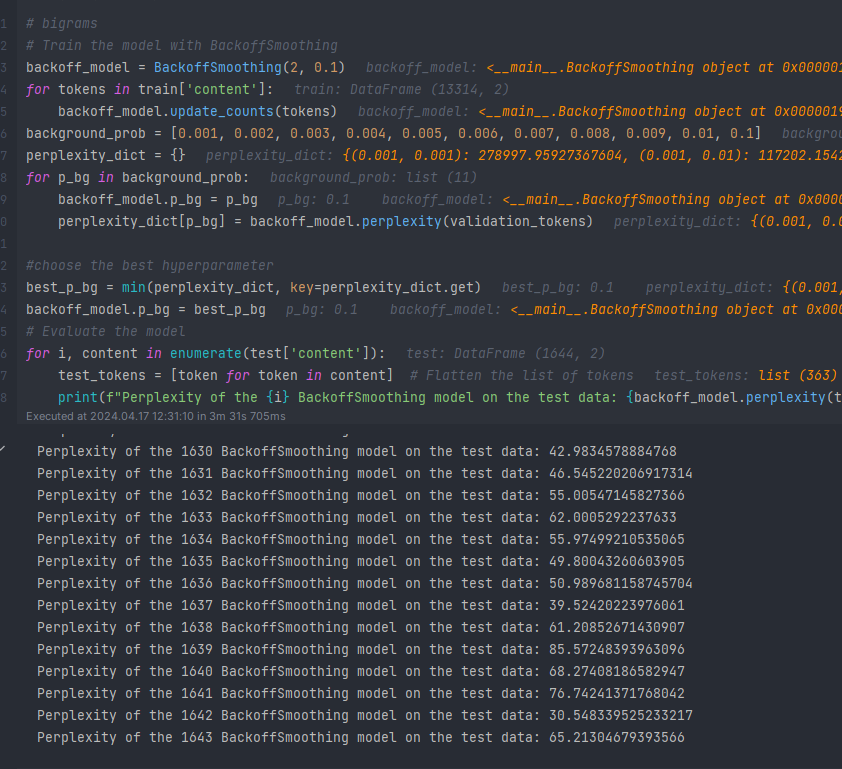
معیار ارزیابی برای تمام متون داده آزمون با مدل های unigram, bigram, trigram با هر دو هموارسازی پیاده سازی شده تست شده.

همین طور از داده ولیدیشن برای انتخاب بهترین ابرپارامتر استفاده کردم

یونیگرم با back off smoothing



بایگرم با back off smoothing



ترایگرم با back off smoothing

A screen shot of a computer

Description automatically generated

یونیگرم با absolute discounting smoothing

A screen shot of a computer program

Description automatically generated

بایگرم با absolute discounting smoothing

A screen shot of a computer program

Description automatically generated

ترایگرم با absolute discounting smoothing

### 

### بخش ب)

برای گزارش بر حسب برچسب کلاس اول داده های برچسب با هم اجماع کردم.

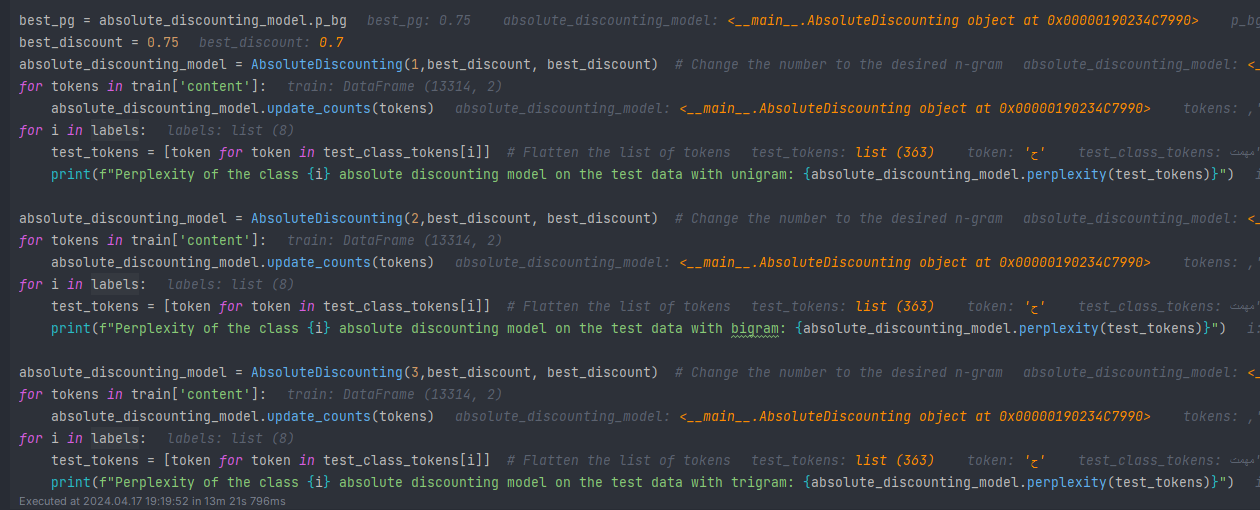


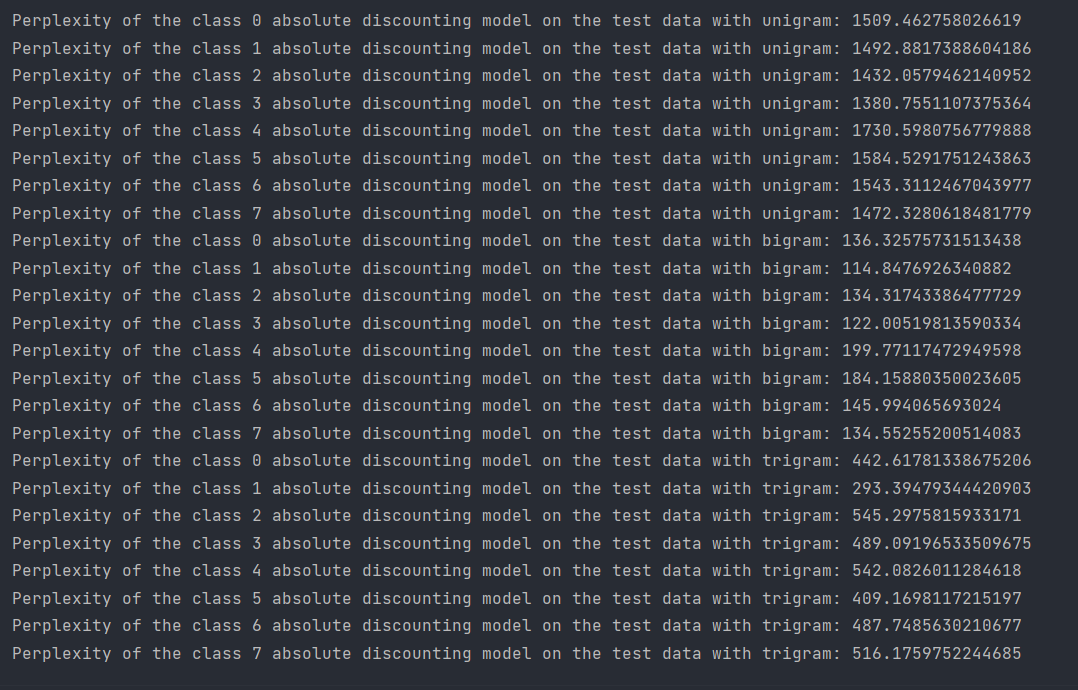
در نهایت با استفاده سه مدل زبانی perplexity را برای هر برچسب بر حسب داده آزمون گزارش کردم.

A screenshot of a computer program

Description automatically generated







### بخش ج)

مواردی که از نتایج به دست آمده می توان تحلیل کرد.

* با افزایش n در مدل زبانی perplexity کاهش پیدا می کند. دلیل ان هم وقتی تعداد (n) اضافه می‌کنیم، در واقع اطلاعات متون بیشتری به مدل زبان ارائه می‌دهیم. این متون اضافی به مدل کمک می‌کنند تا ساختار و جریان زبان را بهتر درک کند و در نتیجه، دقت پیش‌بینی کلمه بعدی در یک دنباله را بالا ببرد. به عبارت دیگر، با افزودن بیشترین میزان متون، مدل بیشتر اطمینان پیدا می‌کند و با این اطلاعات بیشتر، perplexity کاهش می‌یابد. البته در trigram برای absolute discounting این اتفاق نیافتاد دلیلش هم شاید به خاطر مدل هموار سازی است که به ما می دهد.
* توقع داشتم هموار سازی absolute discounting نتایج بهتری نسبت به backoff داشته باشد اما برای این مجموعه داده backoff برای من بهتر جواب داد.
* در داده تست داده برچسب های هر داده تقریبا با هر مدل perplexity مشابه دارند.
* مورد بعد این است که unigram اصلا نتایج جالبی به همراه ندارد.
* بعد همان طور که در بخش اخر گفته می شود که perplexity به تنهایی نمی تواند کافی باشد باید نظر انسانی هم در نظر گرفته شود.

# بخش 3

## تعاریف

Word2Vec یک تکنیک محبوب در پردازش زبان طبیعی (NLP) برای یادگیری تعبیه کلمات است که نمایش‌های برداری عمیق از کلمات را در یک فضای برداری پیوسته ارائه می‌دهد.

دو معماری اصلی برای آموزش مدل‌های Word2Vec وجود دارد: Continuous Bag of Words (CBOW) و Skip-gram.

۱. Continuous Bag of Words (CBOW): CBOW به توجه به متن پیشین کلمه هدف را پیش بینی می‌کند. این معماری یک متن از کلمات محیط را به عنوان ورودی می‌گیرد و سعی می‌کند کلمه هدف را پیش بینی کند. این معماری برای مجموعه داده‌های کوچکتر و کلمات متداول موثر است.

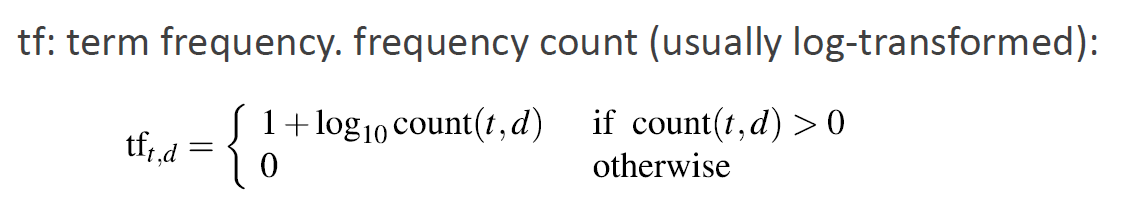
۲. Skip-gram: Skip-gram به عنوان یک معماری دیگر، کلمات محیطی را با توجه به یک کلمه هدف پیش‌بینی می‌کند. این معماری یک کلمه هدف را به عنوان ورودی می‌گیرد و سعی می‌کند کلمات محیط را پیش بینی کند. Skip-gram برای مجموعه داده‌های بزرگتر و گرفتن معنی کلمات کم‌تر موثر است.

هر دو CBOW و Skip-gram از یک شبکه عصبی با یک لایه پنهان برای یادگیری تعبیه کلمات استفاده می‌کنند. در طول آموزش، شبکه عصبی تعبیه کلمات را تنظیم می‌کند تا احتمال پیش‌بینی کلمات محیط (Skip-gram) یا کلمات هدف (CBOW) را به درستی بیشینه کند.

TF-IDF یک آمار عددی است که اهمیت یک کلمه در یک سند نسبت به یک مجموعه از اسناد یا متن مشخص می‌کند. این مفهوم به طور معمول در بازیابی اطلاعات و استخراج متن به عنوان یک روش برای تعیین ارتباط یک واژه با یک سند در یک مجموعه استفاده می‌شود.

در زیر تجزیه و تحلیل نحوه محاسبه TF-IDF آمده است:

1. TF این مقدار نشان می‌دهد که چقدر یک واژه در یک سند تکرار شده است. این به عنوان نسبت تعداد بارهایی که یک واژه در یک سند ظاهر می‌شود به تعداد کل واژگان در سند محاسبه می‌شود. در واقع، این مقدار اهمیت یک واژه درون سند را نشان می‌دهد.

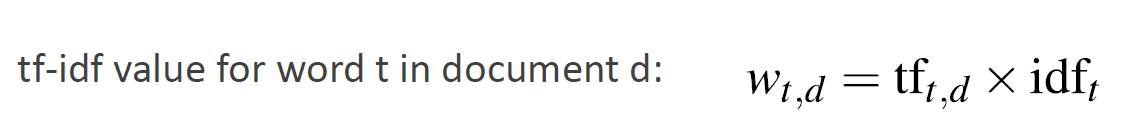


1. IDF: این مقدار نشان می‌دهد که یک واژه در سراسر کل مجموعه متن چقدر مهم است. این به عنوان لگاریتم نسبت تعداد کل اسناد در مجموعه به تعداد اسنادی که حاوی واژه هستند محاسبه می‌شود. واژگانی که در بسیاری از اسناد ظاهر می‌شوند ارزش IDF کمتری دارند، در حالی که واژگانی که در تعداد کمی از اسناد ظاهر می‌شوند، ارزش IDF بیشتری دارند.

A close-up of a computer code

Description automatically generated

1. TF-IDF: در نهایت، امتیاز TF-IDF برای یک واژه در یک سند با ضرب کردن مقادیر TF و IDF محاسبه می‌شود. این موجب می‌شود که واژگانی که درون سند متکرر ولی در کل مجموعه کمیاب هستند، وزن بیشتری داشته باشند و اهمیت آن‌ها در سند بیشتر در نظر گرفته شود

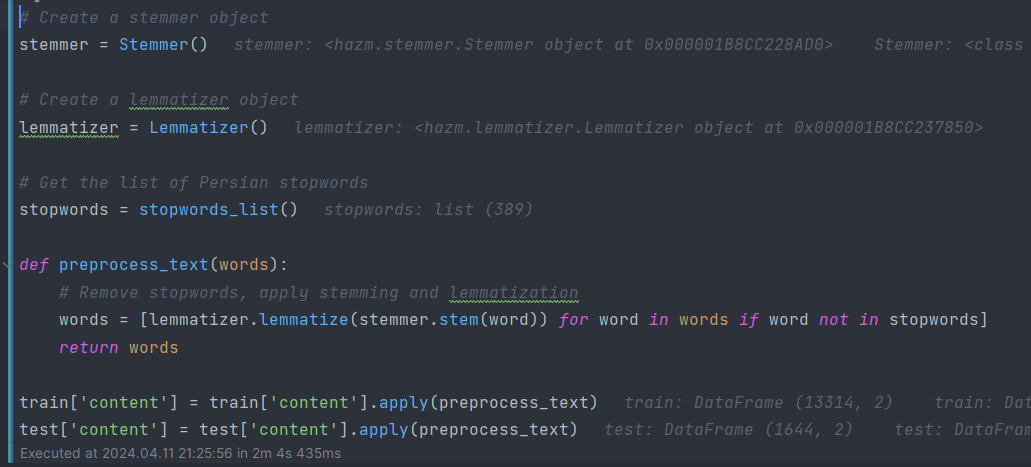


با افزایش امتیاز TF-IDF یک واژه در یک سند، اهمیت بیشتری به آن واژه درون سند نسبت داده می‌شود.

## پیاده سازی میانگین حسابی word2vec

طبق توصیه صورت سوال از کتابخانه genism برای پیاده سازی word2vec استفاده کردم.

پیش پردازش های که برای این سوال انجام دادم.





میانگین حسابی بردار های آموزش و تست استفاده کردم و از knn برای دسته استفاده کردم.

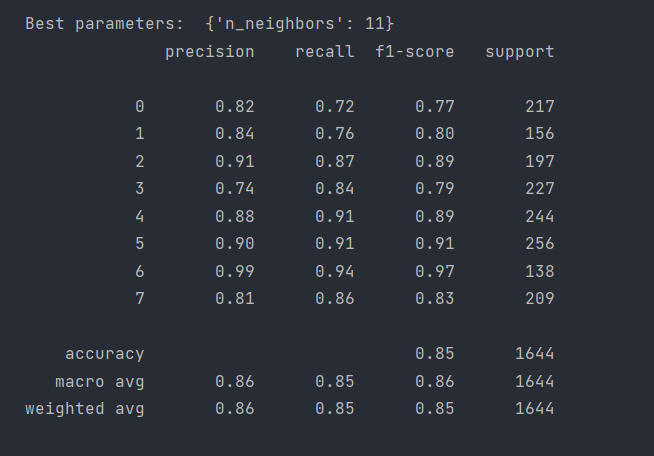


از روش gridsearch برای جستحو پارامتز بهینه استفاده کردم.

A computer screen shot of text

Description automatically generated

نتایج زیر به دست آمد.

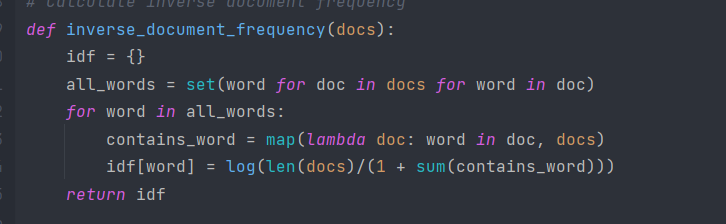


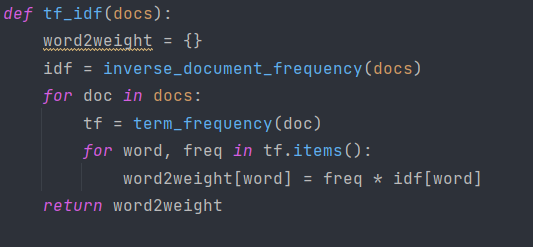
## پیاده سازی وزنی word2vec با وزن های TF-IDF

پیاده سازی tf idf با توجه به توضیح در قسمت توضیحات به این شرح است.

A black background with white text

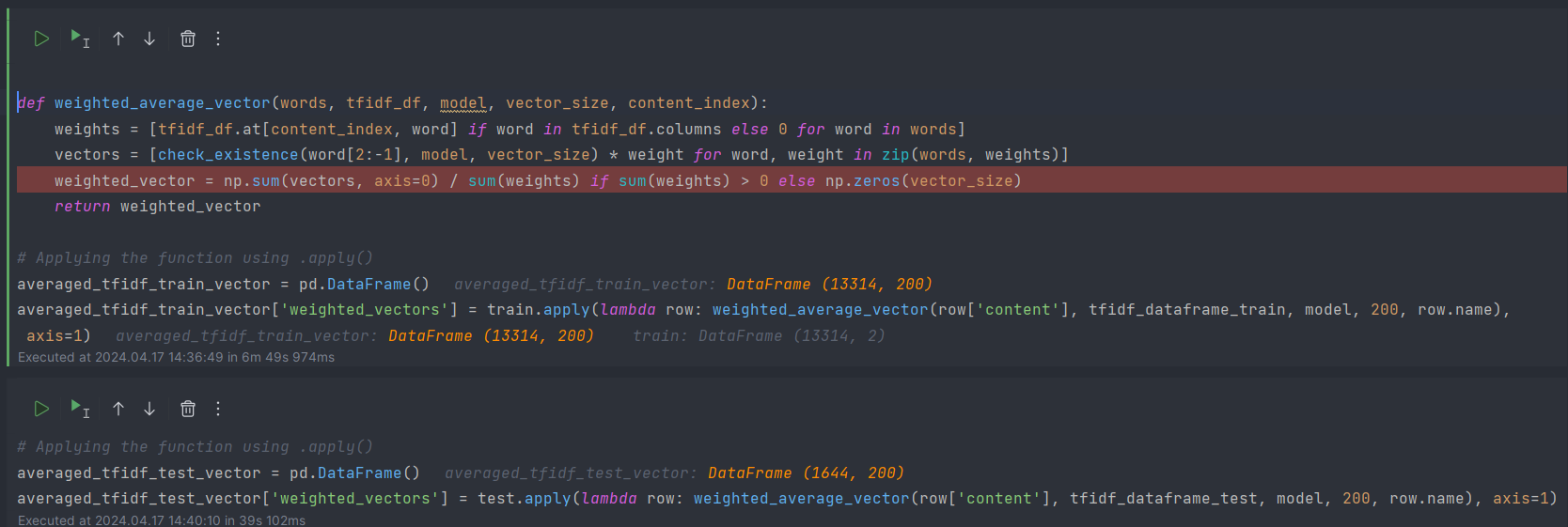
Description automatically generated

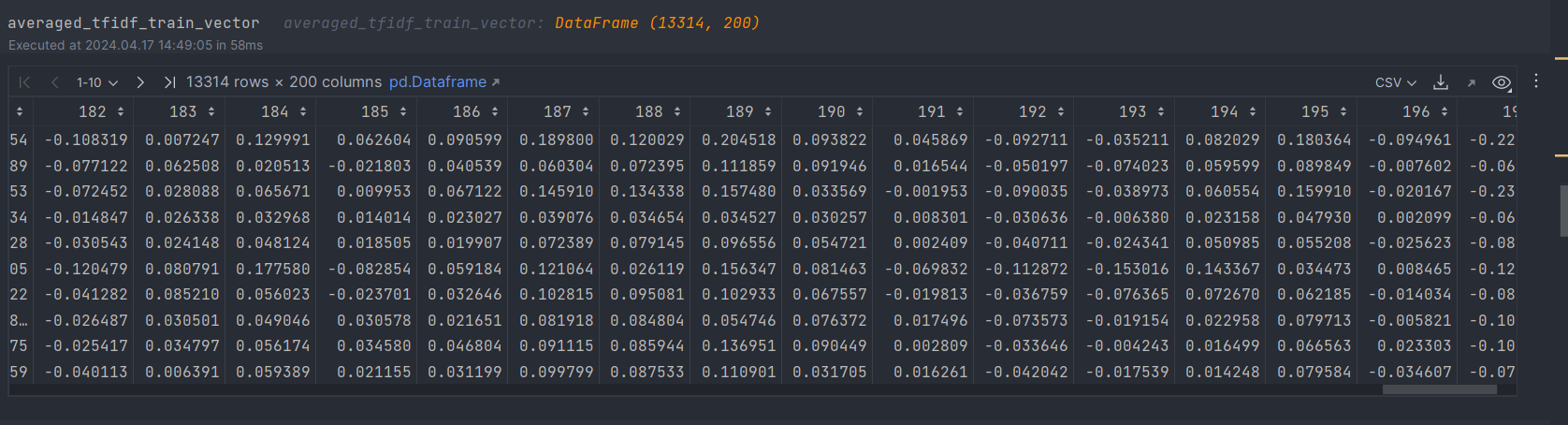




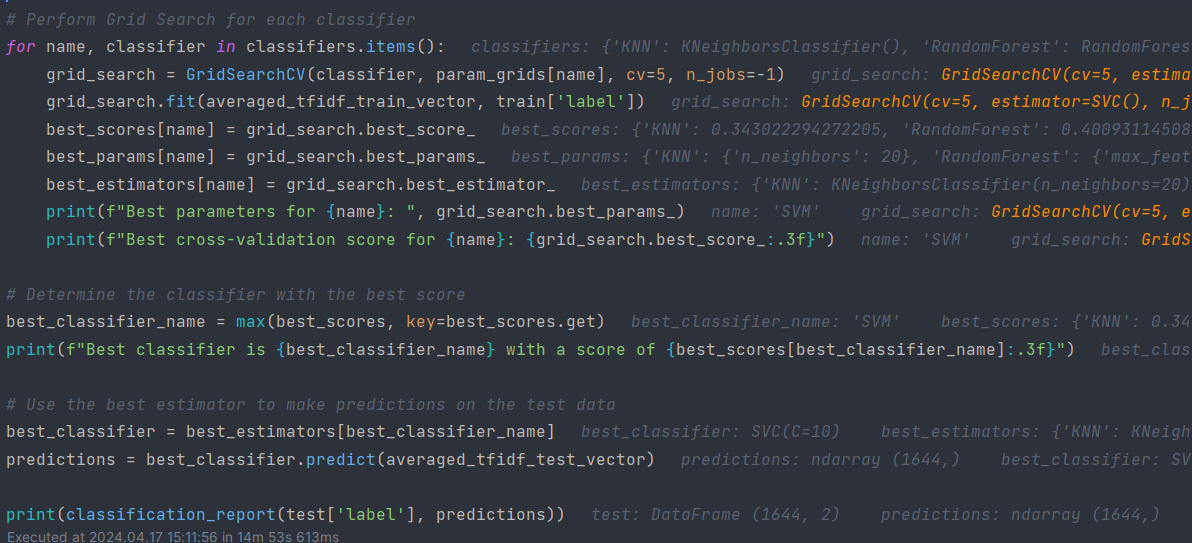
البته به علت کند بودن فرایند بسیار زیاد طول کشید برای همین از sklearn برای انجام ان استفاده کردم.

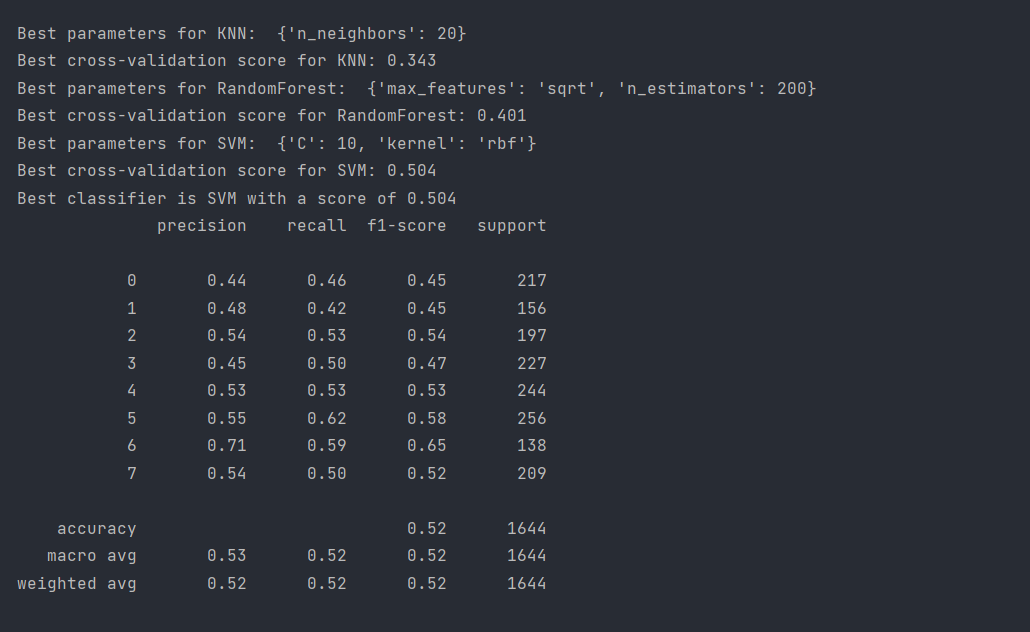








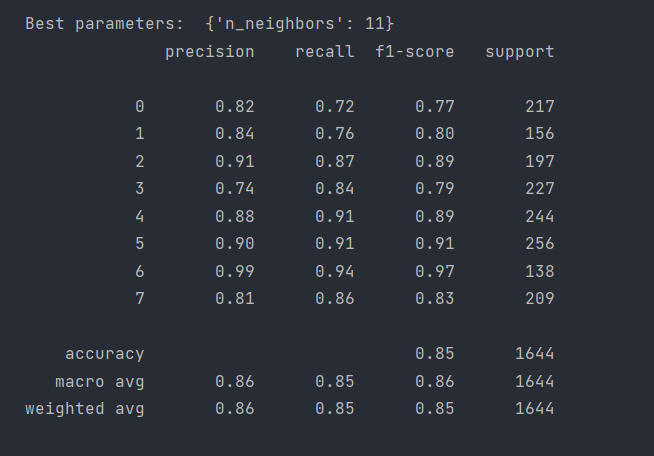




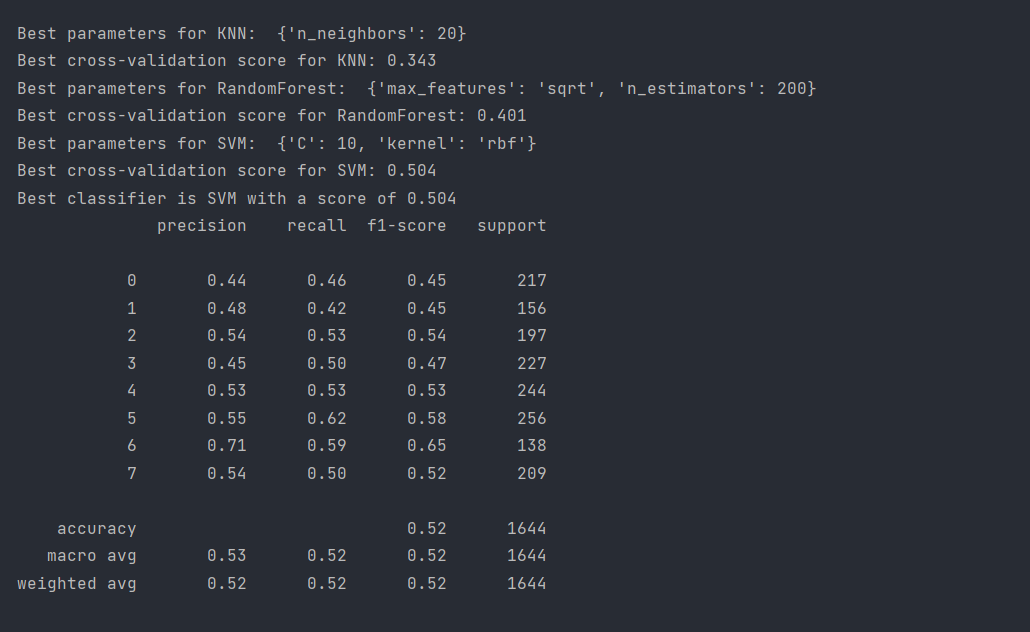
## 

## تحلیل نتایج

نتایج بخش الف



نتایج بخش ب



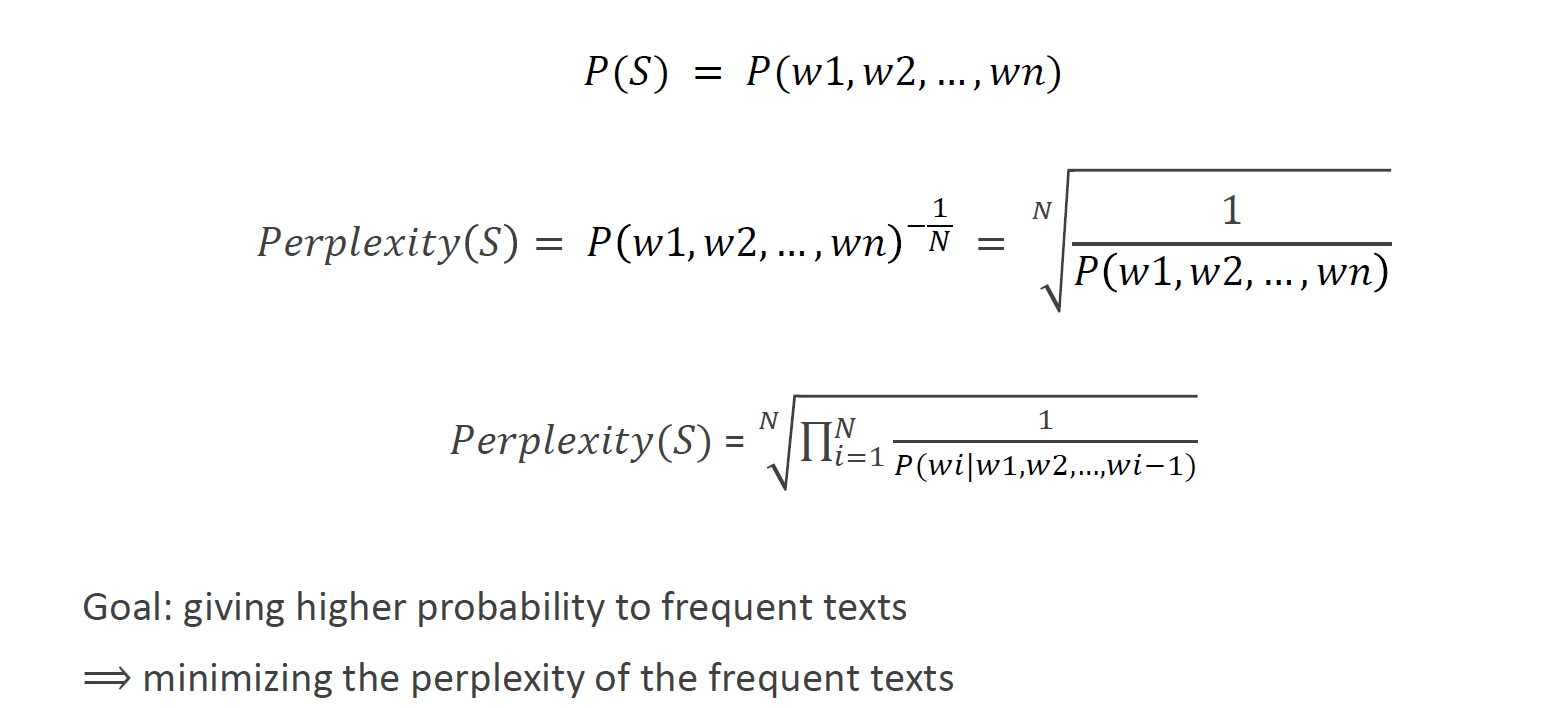
توقع داشتم که روش دوم نتایج بهتری را ارائه دهد چون در واقع بر اساس یک ضریب میانگین حسابی استفاده می کرد اما مشاهده کردیم که نتایج افت شدیدی پیدا کرد. شاید یکی از دلایلش این است word2vec یک مدل عمیق است و مدل های عمیق مشاهده شده که نتایج بهتری تحویل میدهند. و اینکه در word2vec یک فرایند اموزش را طی کرده تا اموزش ببیند بردار های مناسب را برگرداند و که منسب معنای کلمه باشد و نمی توان انتظار داشت با وزن دار کردن میانگین نتایج بهبود پیدا کند.

# بخش 4

## پیاده سازی perplexity

Perplexity یک معیار استفاده می‌شود برای ارزیابی عملکرد یک مدل زبان در پیش‌بینی یک دنباله کلمات. این معیار نشان می‌دهد که مدل زبان چقدر خوب در پیش‌بینی یک نمونه متن یا یک دنباله کلمات عمل می‌کند. به زبان ساده، perplexity نشان می‌دهد که مدل زبان چقدر شگفت‌زده یا گیج می‌شود وقتی که سعی در پیش‌بینی کلمه بعدی در یک دنباله دارد.

۱. محاسبه: Perplexity با استفاده از توزیع احتمالی که توسط مدل زبان برای یک دنباله کلمات داده شده پیش‌بینی می‌شود. این معیار برعکس احتمال مجموعه آزمایشی است، با توجه به تعداد کلمات معمولی‌سازی می‌شود. به طور ریاضی، perplexity (PP) به شکل زیر محاسبه می‌شود:

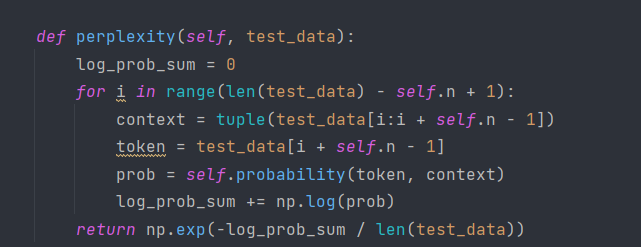


۲. تفسیر: یک perplexity کمتر نشان می‌دهد که مدل اطمینان بیشتری دارد و بهتر در پیش‌بینی دنباله کلمات عمل می‌کند. یک perplexity به اندازه یک به معنای این است که مدل دنباله کلمات را به‌طور کامل پیش‌بینی می‌کند که عملیاتی نیست. برعکس احتمال مرتبط است. مقدار کمتر نشان‌دهنده احتمال بالاتری است که مدل به درستی دنباله کلمات را پیش‌بینی کرده است. Perplexity را مثل انتخاب پاسخ در یک سوال چندگزینه تصور کنید. مقدار کم نشان می‌دهد که مدل دارای مجموعه کوچکی از کلمات احتمالی برای انتخاب است، در حالی که یکperplexity بالا نشان‌دهنده وجود یک انتخاب گسترده‌تر از امکانات است، که باعث می‌شود پیش‌بینی کلمه صحیح دشوارتر شود.

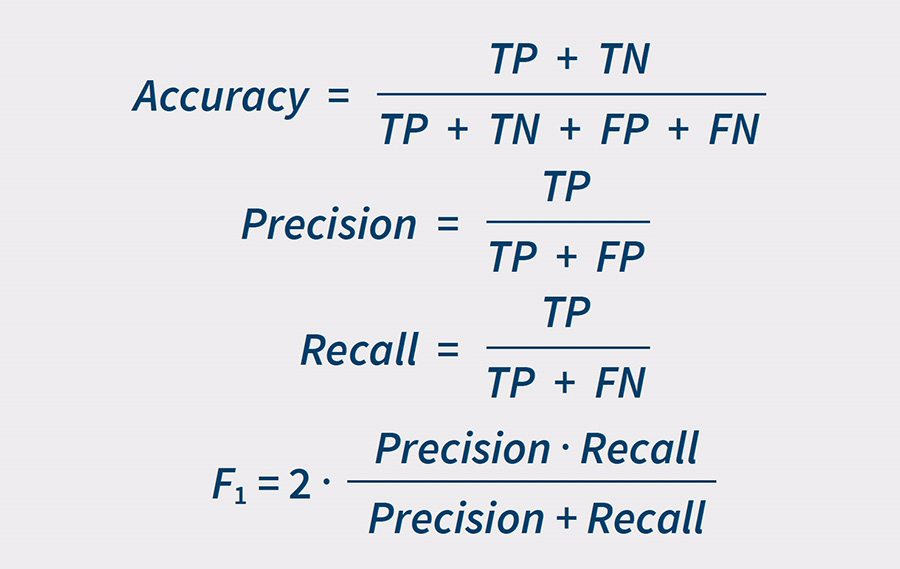
۳. ارزیابی: Perplexity معمولاً برای مقایسه مدل‌های زبانی مختلف یا تنظیمات مختلف همان مدل استفاده می‌شود. این معمولاً در طول فرایند آموزش برای نظارت بر عملکرد مدل و هدایت تنظیمات هایپرپارامتر استفاده می‌شود. علاوه بر این، perplexity می‌تواند برای مقایسه عملکرد مدل‌های زبانی در وظایف یا مجموعه داده‌های مختلف استفاده شود.

با این حال، مهم است بدانید که perplexity همیشه قابل فهم نیست و ممکن است با ارزیابی انسانی از پردازش زبان یا هماهنگی، هماهنگی نداشته باشد. بنابراین، اگرچه perplexity یک معیار مفید برای ارزیابی مدل‌های زبانی است، اما باید با تکنیک‌های ارزیابی دیگر مانند ارزیابی انسانی یا معیارهای وظیفه‌ای، ترکیب شود تا درک جامعی از عملکرد مدل بدست آید.

4. پیاده سازی: در قسمت 1 پیاده سازی انجام شده بود با این حال توضیح آن را نیز در این قسمت آوردیم. پیاده سازی از فرمول دیگری perplexity استفاده کردم که معادل همان عکس صفحه قبل است.

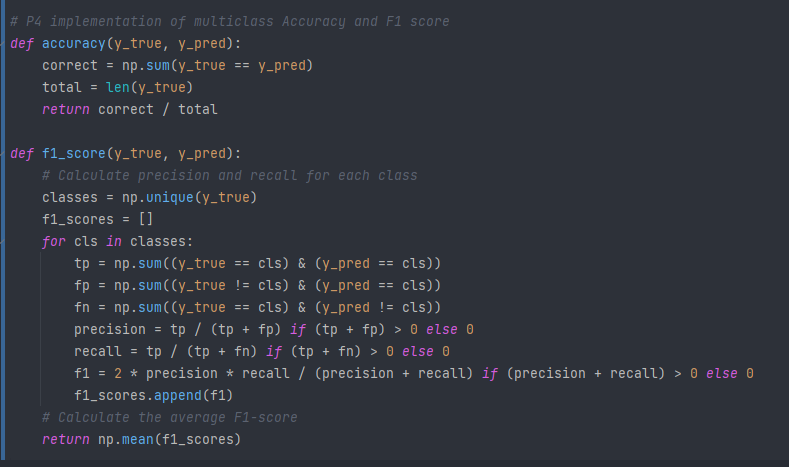


## پیاده سازی accuracy و f1 score در حالت چند کلاسه



صحت یا همان accuracy در حالت چند کلاسه تفاوتی ندارد.

اما f1 به اسین گونه است برای هر هر کلاس فرض می کنیم که بقیه کلاس ها منفی هستند و خودش مثبت این کار برای همه کلاس ها انجام می دهیم و در اخر میانگین می گیریم.



A screenshot of a computer

Description automatically generated