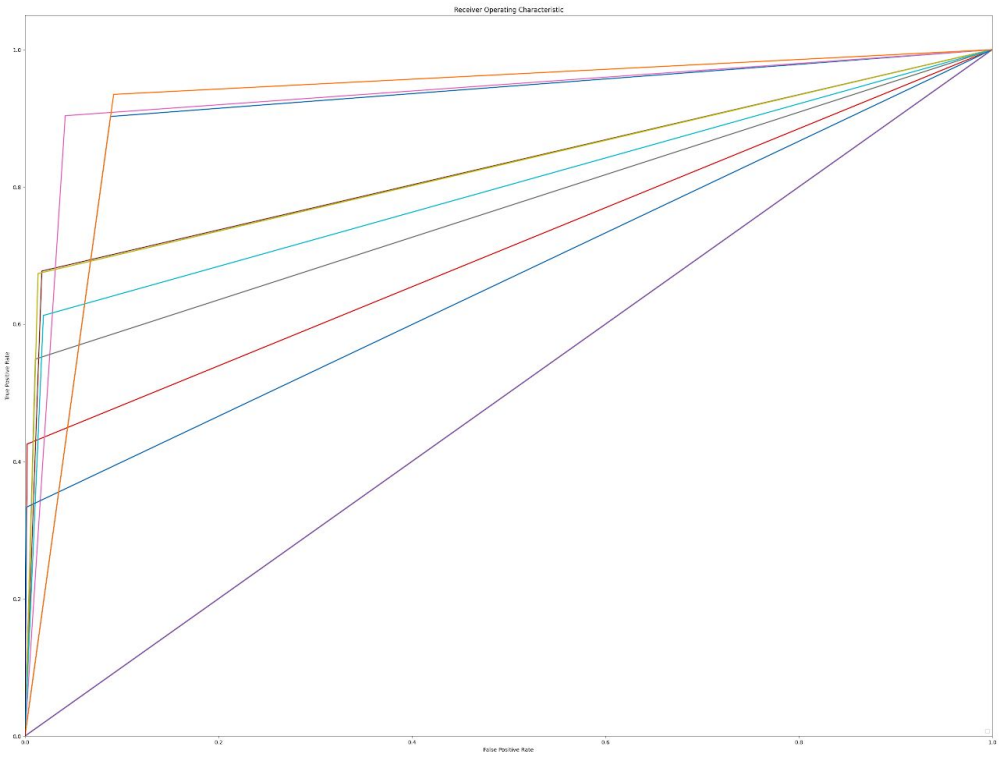
توضیحات داده:

پیکره شامل ۲بخش که یکی برای آموزش و validation که شامل ۱۱۷۱۹۲ خبر و دیگری برای ارزیابی که شامل ۲۱۱۰۴ است.

بعد از مراحل تمیزسازی داده‌های آموزشی این مجموعه داده شامل ۱۰۷ حرف و ۷۱۴۹۸ کلمه متفاوت است که بعد از محاسبه ۱۰۰۰۰ کلمه پرتکرار می‌توان گفت سهم این کلمات از کلیه توکن‌ها برابر ۹۶.۸۲ درصد است.

درنهایت بعد از آموزش یک مدل SVM روی این مجموعه داده نمودار ROC در نمایش سطح-کلمه را هم می‌توانید به شکل زیر مشاهده کنید.



توضیحات کد:

- فایل text_cleaner: در این فایل کلاس و توابع مختص به تمیزسازی و آماده‌سازی داده قرار دارد.
  - تابع init: این تابع یک متغیر به نام level به عنوان ورودی می‌گیرد که بر اساس این مقدار خروجی‌های مربوط به هر یک از سطح‌ها را ایجاد کند.
  - تابع remove_punctuations: این تابع یک متن را به عنوان ورودی می‌گیرد و بعد از حذف علائم نگارشی از متن آن را به عنوان خروجی برمی‌گرداند.
  - تابع remove_diacritics: این تابع یک متن را به عنوان ورودی می‌گیرد و بعد از حذف اعراب از متن آن را به عنوان خروجی برمی‌گرداند.
  - تابع remove_emojis: این تابع یک متن را به عنوان ورودی می‌گیرد و بعد از حذف ایموجی‌ها از متن آن را به عنوان خروجی برمی‌گرداند.
  - تابع remove_english_characters: این تابع یک متن را به عنوان ورودی می‌گیرد و بعد از حذف حروف انگلیسی از متن آن را به عنوان خروجی برمی‌گرداند.
  - تابع remove_half_spaces: این تابع یک متن را به عنوان ورودی می‌گیرد و بعد از حذف نیم‌فاصله‌ها از متن آن را به عنوان خروجی برمی‌گرداند که دلیل این کار برای یکسان شدن نمایش‌های متفاوت کلمات است.
  - تابع mask_numbers: این تابع یک متن را به عنوان ورودی می‌گیرد و بعد از جایگزینی عدد با یک رشته مشخص در متن آن را به عنوان خروجی برمی‌گرداند.
  - تابع clean_text: این تابع یک متن را به عنوان ورودی دریافت می‌کند و با استفاده از توابع بالا متن را تمیزسازی می‌کند و سپس آن را به عنوان خروجی برمی‌گرداند.
  - تابع stem: این تابع یک توکن را به عنوان ورودی دریافت می‌کند و بعد از stemming کردن این کلمه آن را به عنوان خروجی برمی‌گرداند.
  - تابع tokenize: این تابع یک متن را به عنوان ورودی دریافت می‌کند و در ابتدا با استفاده از تابع clean_text متن را تمیزسازی می‌کند و سپس توکن‌های اولیه را بدست می‌آورد و

سپس این توکن‌ها با اعمال تابع stem و حذف stopwordها توکن‌های نهایی را به عنوان خروجی برمی‌گرداند.

- تابع clean_training_set: این تابع با استفاده از توابع بالا مجموعه آموزش را به یک مجموعه از توکن‌های نهایی و مناسب برای روند بردار کردن و آموزش می‌کند.
- تابع clean_test_set: این تابع با استفاده از برخی نتایج ذخیره شده در مرحله تمیزسازی داده آموزشی و به کمک توابع بالا مجموعه test را آماده می‌کند.

- فایل classifier: در این فایل کلاس و توابع مختص به دسته‌بندی و بردار کردن متون قرار دارد.
  - تابع init: این تابع یک متغیر به نام level را به عنوان ورودی می‌گیرد که بر اساس این مقدار تصمیم گرفته می‌شود از کدام نمایش سطوح استفاده شود.
  - تابع clean: این تابع با از استفاده از کلاس TextCleaner مجموعه‌های آموزش و test را تمیزسازی می‌کند.
  - تابع vectorize: این تابع یک لیست را به عنوان ورودی می‌گیرد که هر عضو این لیست در واقع توکن‌های ایجاد شده برای یک سند هست و به عنوان خروجی یک لیست برمی‌گرداند که هر عضو این لیست نمایش برداری آن سند است.
  - تابع defining_model: در این تابع مدل‌هایی که در تابع train قصد استفاده از آن‌ها داریم را initialize می‌کنیم.
  - تابع train: در این تابع با استفاده مدل‌های تعریف شده در تابع defining_model و بردار کردن توکن‌های اسناد با تابع vectorize مدل‌ها را آموزش می‌دهیم.
  - تابع evaluate: در این تابع با استفاده از مدل‌های آموزش دیده شده در تابع train و مجموعه آماده شده test مدل را با معیارهای متفاوت ارزیابی می‌کنیم.