گزارش پروژه سیستم‌های فازی

هوش محاسباتی

دکتر حسین کارشناس

آدرینا ابراهیمی ۹۹۳۶۲۳۰۰۲

کیان مجلسی ۹۹۳۶۱۳۰۵۱

ریپازیتوری گیت‌هاب پروژه

(ریپازیتوری گیت‌هاب پس از آخرین ارائه در دسترس همگان قرار می‌گیرد.)

خرداد ۱۴۰۲

فهرست

۱- مقدمه

این پروژه با هدف توسعه یک سیستم مبتنی بر سامانه‌های فازی برای تشخیص پیامک‌های جعلی از واقعی است. در ابتدا، با استفاده از داده‌های SMSSpamCollection سامانه را در یک فرایند بهینه‌سازی آموزش می‌دهیم و پایگاه قوانین را استخراج می‌کنیم. سپس، با استفاده از پایگاه قوانین به دست آمده که مبتنی بر منطق فازی است، به دسته‌بندی پیام‌های واقعی و جعلی می‌پردازیم.

۲- تشریح مسئله

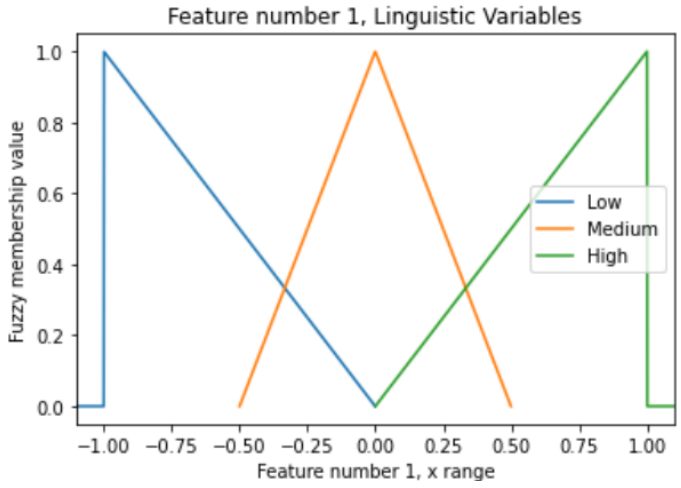
در این بخش به تشریح مسئله تعریف شده خواهیم پرداخت. هدف از این پروژه توسعه یک پایگاه قوانین فازی و استفاده از استدلال تقریبی برای دسته‌بندی پیامک‌ها است. هر قانون در این پایگاه برای نگاشت نمونه‌های منطبق با شرایط توصیف شده توسط یک مقداردهی برای متغیرهای زبانی به یک دسته به کار گرفته می‌شود. نمونه‌ای از قانون ذکر شده به شکل زیر است:

1) If X_1 is A_{1i} and X_2 is A_{2j} and ... and X_n is A_{nk} Then $Y = 0$

که در آن X_1 تا X_n متغیرهای زبانی مسئله بوده و هر کدام مرتبط با یکی از ویژگی‌هایی است که برای توصیف پیامک در نظر گرفته شده است. هر یک از این متغیرها دارای مجموعه‌ای مقادیر زبانی است که با استفاده از مجموعه های فازی تعریف می شوند:

$T(X_i) = \{A_{i1}, \ldots, A_{im(i)}\}$

متغیر Y نشان‌دهنده دسته بوده و دارای یکی از دو مقدار ۰ (واقعی) و ۱ (جعلی) است. شکل پایین مثالی از یک متغیر زبانی با مقادیر High, Medium, Low بوده که بر روی مجموعه جهانی اعداد حقیقی در بازه [-1,1] تعریف شده‌اند را نشان می‌دهد.



پس از استخراج ویژگی‌های هر پیامک به صورت ترد میزان تطابق مقدار هر ویژگی با مقادیر زبانی مختلف متغیر مربوط به آن ویژگی بدست می‌آید (از محاسبه درجه عضویت مقدار آن ویژگی در مجموعه‌های فازی هر یک از مقادیر). برای مثال اگر مقدار مشاهده شده برای ویژگی ۱ در شکل بالا، ۰.۲۵- باشد، میزان تطابق آن با مقادیر Low، Medium، High از متغیر زبانی مربوط به ویژگی ۱ به ترتیب برابر با ۰.۲۵، ۰.۵ و ۰ است. بر این اساس می‌توان با تجمیع میزان تطابق شرط‌های مختلف یک قانون میزان تطابق کلی آن قانون با یک ورودی را تعیین کرد. رابطه زیر میزان تطابق کلی قانون نشان داده شده در رابطه اول را با استفاده از عملگر ضرب جبری برای تجمیع نشان می‌دهد.

2)

$$g_R\left(\boldsymbol{x}^{(p)}\right) = \mu_{A_{1i}}\left(x_1^{(p)}\right) \times \mu_{A_{2j}}\left(x_2^{(p)}\right) \times \ldots \times \mu_{A_{nk}}\left(x_n^{(p)}\right)$$

این روند برای هر یک از قوانین موجود در پایگاه انجام شده و میزان تطابق کلی هر یک از آنها با ورودی محاسبه می شود. در این صورت می‌توان با تجمیع میزان تطابق قوانین مرتبط با هر یک از دسته ها (در این مسأله فقط دسته ۰ و ۱)، دسته ای که دارای تطابق بیشتری با ورودی است را مطابق روابط زیر برای آن ورودی در نظر گرفت.

3)

$$g_c\left(\boldsymbol{x}^{(p)}\right) = \sum_{R_j \in\, class(c)} g_{R_j}\left(\boldsymbol{x}^{(p)}\right)$$

4)

$$\hat{y}\left(\boldsymbol{x}^{(p)}\right) = \arg \max_{c \in \{0,1,\cdots\}} g_c\left(\boldsymbol{x}^{(p)}\right)$$

برای توسعه پایگاه قوانینی که به این شکل به کار گرفته می شود، در این پروژه از مجموعه داده SMS Spam Collection که از پایگاه UCI قابل دسترسی است استفاده می‌شود. این مجموعه داده دارای ٥٥٧٤ داده متنی است که به یکی از دو دسته خروجی متعلق هستند. پایگاه قوانین باید به شکلی طراحی شود که با داده‌های موجود در این مجموعه داده تطبیق پیدا کند.

با توجه به فضای پیچیده حاصل از مقادیر مختلف برای پارامترهای چنین پایگاه قوانینی، از الگوریتم‌های تکاملی برای بهینه‌سازی در روند آموزش سامانه استفاده می‌شود. در این رویکرد، به علت غیرقطعی بودن قوانین، کیفیت (برازندگی) هر قانون ایجاد شده در پایگاه قوانین را می‌توان با توجه به ضریب اطمینان (CF)

آن قانون در هنگام دسته‌بندی مشخص کرد که مرتبط با درجه تطابق کلی قانون برای نمونه‌های هر دسته است:

5)

$$f_c(R_j) = \sum_{x^{(p)}:y^{(p)}=c} g_{R_j}(x^{(p)})$$

6)

$$CF(R_j) = \frac{f_{y_j}(R_j) - f^{neg}(R_j)}{\sum_{c\in\{0,1,\cdots\}} f_c(R_j)}$$

7)

$$f^{neg}(R_j) = \frac{1}{r-1}\sum_{c\neq y_j} f_c(R_j)$$

در صورت کسر داده شده رابطه ۶، $f_{yj}(R_j)$ نشان‌دهنده درجه تطابق کلی نمونه‌های آموزشی دسته تعیین شده در خروجی R_j است و $f^{neg}(R_j)$ میانگین تطابق کلی قانون R_j با هر یک از کلاس‌های دیگر است. مقدار r در رابطه ۷ نشان‌دهنده تعداد کل دسته‌ها است.

در این پروژه ویژگی‌های TF-IDF از متن پیامک‌های موجود در مجموعه داده در روند پیش‌پردازش استخراج می‌شود، هر چند انواع دیگری از ویژگی‌ها به این منظور قابل استفاده است. با توجه به تعداد زیاد این ویژگی‌ها، در روند پیش‌پردازش تعداد آن‌ها کاهش می‌یابد تا ایجاد پایگاه قوانین ساده‌تر شود. برای کاهش ابعاد ویژگی‌ها دور رویکرد در نظر گرفته شده است: ۱) انتخاب ویژگی که به انتخاب زیرمجموعه‌ای از ویژگی‌های مهم‌تر می‌پردازد و در این پروژه از معیار اطلاعات متقابل برای شناسایی چنین ویژگی‌هایی استفاده شده است. ۲) استخراج ویژگی‌های جدیدی از روی ویژگی‌های اولیه ایجاد می‌کند و در این پروژه از روش تحلیل مولفه‌های اصلی به این منظور استفاده شده است.

۳- ملاحضاتی که در حل مسئله باید در نظر گرفته شوند

۳-۱- الف) مدل‌سازی زبانی مناسب برای ۵ ویژگی مهم‌تر به دست آمده از پیش‌پردازش متون

- هر متغیر زبانی (منتاظر با هر یک از ویژگی‌ها) می‌تواند بین ۳ تا ۵ مقدار زبانی داشته باشد.
- هر مقدار زبانی می‌تواند با یکی از چهار مجموعه فازی زیر نشان داده شود:
  - مجموعه فازی مثلثی متساوی الساقین (s>0)

$$\mu_{iso-tri}(x) = \max\left(\min\left(\frac{x-m+s}{s}, \frac{m-x+s}{s}\right), 0\right)$$

- مجموعه فازی ذوزنقه قائم الزاویه (|s|>0)

$$\mu_{rect-trap}(x) = \max\left(\min\left(\frac{x-m+s}{s}, 1\right), 0\right)$$

- مجموعه فازی گاوسی

$$\mu_{gaussian}(x) = e^{-\frac{1}{2}(\frac{x-m}{s})^2}$$

- مجموعه فازی سیگموئید

$$\mu_{sigmoid}(x) = \frac{1}{1+e^{-\frac{x-m}{s}}}$$

- هر مجموعه فازی دارای دو پارامتر m و s است.

۳-۲- ب) به دست آوردن قوانین فازی دسته‌بندی بر اساس مدل‌سازی زبانی

- برای هر قانون باید قسمت شرط آن با تعیین یکی از مقادیر برای هر متغیر زبانی تعیین شود.
  - هر یک از مقادیر به صورت مستقیم یا نفی شده (negated) می‌تواند در قانون به کار گرفته شود.
  - ممکن است برخی متغیرها در یک قانون بکار نروند.
- برای هر قانون یکی از دسته‌ها به عنوان خروجی تعیین شود.
- تعداد قوانین پایگاه محدود است (هدف انتخاب زیرمجموعه بهینه از قوانین نیست بلکه دستیابی به قوانینی است که بتواند عملیات دسته‌بندی را با عملکرد مناسب انجام دهد).

۳-۳- پ) شرح مؤلفه‌های الگوریتم تکاملی

۳-۳-۱ راه‌حل‌ها (کلاس Chromosome)

در این کلاس‌ها راه حل‌ها که هر کدام یک قانون را نمایش می‌دهند، پیاده‌سازی شده است. هر راه حل مجموعه‌ای از متغیرها، مقادیر مختلف برای هر متغیر، مجموعه فازی و پارامترهای آن مجموعه برای هر مقدار به همراه کلاسی که آن قانون به آن تعلق دارد و برازندگی را نمایش می‌دهد.

۳-۳-۱-۱ متد initialize

در این متد متغیرها به صورت رندوم به تعداد ۱ تا ۵ از ویژگی‌های استخراج شده انتخاب می‌شوند. پس از آن، برای هر متغیر از مقادیر زبانی Normal، Minor، Moderate، Serious و Severe یکی به تصادف انتخاب می‌شود و در نهایت ویژگی نفی بودن و کلاس کروموزوم به صورت رندوم مقداردهی می‌شود.

۳-۳-۱-۲ متد mutate

در این قسمت کد، هر متغیر قانون‌مان را به احتمال ۵۰ درصد مورد جهش قرار می‌دهیم. برای جهش هر متغیر، از لیستی از متغیرها که در قانون‌مان وجود ندارد استفاده می‌کنیم و یکی از آن‌ها را به تصادف جایگزین یکی از متغیرهای قانون می‌کنیم. برای اینکه کلیدهای دیکشنری‌های terms_per_variable و is_not نیز آپدیت باشد؛ قبل از جایگزینی مقدارهای آن‌ها را از دیکشنری pop کرده و برای کلید جدید (متغیر تغییر یافته) این مقدارها را قرار می‌دهیم. سپس مقادیر زبانی هر متغیر، نفی بودن متغیر و کلاس خروجی قانون را به صورت تصادفی تغییر می‌دهیم. در نهایت برای بروزرسانی مدل‌سازی زبانی، دیکشنری linguistic_model را پاک می‌کنیم و سپس به ازای هر متغیر مجددا یک تابع به صورت تصادفی انتخاب کرده و پارامترهای m و s را نیز بر اساس جهش گاوسی مقادیر قبلی تغییر می‌دهیم.

۳-۳-۱-۳ متد recombinate

این متد با دریافت دو والد عملیات بازترکیب را روی ژن‌ها انجام داده و دو فرزند تولید می‌کند. ابتدا متغیرهای زبانی، مقادیر زبانی و نفی بودن متغیرها برای دو والد را در لیست‌های جداگانه ذخیره می‌کنیم. سپس از لیست‌های تعریف شده به صورت رندوم متغیرهای زبانی، مقادیر متعلق به هر متغیر زبانی و نفی بودن متغیر زبانی را برای فرزندان تعیین می‌کنیم. در نهایت نیز، کلاس والد اول را به فرزند اول و کلاس والد دوم را به فرزند دوم نسبت می‌دهیم.

۳-۳-۱-۴ متد calculate_fitness

برای حساب کردن برازندگی یک قانون، از معیار ضریب اطمینان (CF) استفاده شده است. برای حساب کردن این ضریب از فرمول‌های ۵، ۶ و ۷ استفاده شده است.

۳-۳-۱-۵ متد compute_gR_

برای حساب کردن میزان تطابق یک قانون با داده ورودی، از فرمول ۲ استفاده می‌کنیم. به ازای هر متغیر در قانون، مقدار نظیر آن را از داده ورودی دریافت کرده و با توجه به مقدار زبانی و تابع نظیر آن، مقدار μ را به دست آورده و در نهایت این مقادیر را باهم ضرب/مین می‌کنیم.

۳-۳-۲ الگوریتم تکاملی (کلاس Evolutionary_Algorithm)

در این کلاس الگوریتم تکاملی پیاده‌سازی شده است. این کلاس با دریافت جمعیت اولیه، حداکثر تعداد نسل‌ها، احتمال جهش و احتمال بازترکیب و مقداردهی اولیه لیست والدین، فرزندان، تاریخچه برازندگی‌ها و متغیر جمعیتی با بهترین برازندگی کار خود را آغاز می‌کند. سپس عملیات لازم برای الگوریتم تکاملی مانند، بررسی مدل‌سازی زبانی، تولید جمعیت اولیه، جهش، بازترکیب، انتخاب، تولید نسل بعدی و انتخاب بهترین کروموزوم انجام می‌شود. متدهای این کلاس به شرح زیر است.

۳-۳-۲-۱ متد check_linguistic_model

در این تابع بررسی می‌کنیم تا تمامی جفت‌های متغیر و مقدار زبانی موجود در کرومزوم‌ها یک تابع به همراه پارامترهای m و s برای انجام مدل‌سازی داشته باشند.

۳-۳-۲-۲متد initialize

در این متد ابتدا لیست جمعیت را خالی کرده و پس از آن با استفاده از یک حلقه به طول اندازه جمعیت، کروموزم ساخته و بررسی می‌کنیم حتما به ازای جفت متغیر و مقدار زبانی یک تابع با پارامترهای s و m به متغیرهای کروموزوم تخصیص داده شود. پس از آن برازندگی کروموزوم را محاسبه کرده و آن را به لیست جمعیت اضافه می‌کنیم.

۳-۳-۲-۳ متد mutation

در این متد با احتمال جهشی که داریم، متد جهش از شئ کروموزوم صدا زده شده و عملیات جهش انجام شده و متد check_linguistic_model را برای کروموزوم فراخوانی می‌کنیم. سپس، به ازای هر جفت متغیر و مقدار زبانی با احتمالی تابع و پارامترهای s و m را جهش می‌دهیم.

۳-۳-۲-٤ متد recombinattion

این متد با دریافت دو والد با احتمال بازترکیبی که داریم، متد بازترکیب از شئ کروموزوم را صدا زده و عملیات بازترکیب انجام می‌شود. سپس، متد check_linguistic_model برای هر دو فرزند صدا زده می‌شود. در نهایت فرزندان تولید شده به لیست فرزندان اضافه می‌شوند. در صورتی که احتمال بازترکیب برآورده نشود والدین به لیست فرزندان اضافه می‌شوند.

۳-۳-۲-۵ متد tournament_selection

در این متد انتخاب با استفاده از روش tournament انجام شده و از نصف تعداد جمعیت کروموزوم انتخابی با جایگذاری به صورت رندوم، کروموزومی که بیشترین برازندگی را دارد به عنوان والد انتخاب می‌شود.

۳-۳-۲-٦ متد generate_next_generation

در این متد عملیات ساخت یک نسل انجام می‌شود. ابتدا لیست فرزندان خالی شده، سپس با استفاده از روش انتخاب دو والد انتخاب کرده و روی آن‌ها عملیات بازترکیب را انجام می‌دهیم. پس از آن، عملیات جهش را انجام داده و برازندگی را برای تمامی فرزندان محاسبه می‌کنیم. در نهایت جمعیت فرزندان را به جمعیتی که داشتیم اضافه می‌کنیم. حال برای انتخاب جمعیت نسل بعدی، جمعیت به دو گروه تقسیم می‌شود؛ یکی کلاس صفر و یکی کلاس یک؛ هر دو قسمت به صورت جداگانه برحسب برازندگی مرتب شده و در یک حلقه به ترتیب یک کروموزوم از گروه اول و یک کروموزوم از گروه دوم را در لیستی ذخیره کرده و در نهایت به تعداد population_size کروموزوم از ابتدای آن لیست را برای جمعیت نسل بعدی انتخاب می‌کنیم.

۳-۳-۲-۷ متد run

در این متد جمعیت اولیه تشکیل شده و به تعداد حداکثر تعداد نسل‌ها عملیات تولید جمعیت نسل بعد و اضافه کردن میانگین برازندگی هر نسل به لیست تاریچه برازندگی‌ها انجام می‌شود. در صورت متوالی بودن میانگین نسل‌ها برای ۵ نسل، عملیات تکامل متوقف می‌شود

۳-۳-۲-۸ متد mean_fitness

در این متد، میانگین برازندگی کروموزوم‌های هر نسل محاسبه و برگردانده می‌شود.

۳-٤- ت) دلیل انتخاب هر یک از مؤلفه‌های الگوریتم تکاملی

۳-٤-۱ روش نمایش راه حل‌ها

برای نمایش راه حل‌ها از روش میشیگان استفاده شده که هر قانون نشان‌دهنده یک کروموزوم می‌باشد و پس از پایان تکامل کل جمعیت باقی‌مانده به عنوان پایگاه‌قوانین به کار می‌روند. به دلیل سهولت در انجام عملیات جهش و بازترکیب و همچنین به کارگیری کم‌تر حافظه از این روش استفاده شده‌است.

۳-٤-۲ تابع هدف

توابع هدف کاملا مطابق با آنچه در صورت پروژه بیان شده بود پیاده‌سازی شده‌اند.

۳-٤-۳ روش انتخاب

استفاده از tournament selection به ما این امکان را می‌دهد که فشار انتخاب را متناسب با نیاز تغییر دهیم و امکان جایگذاری مجدد این امکان را به الگوریتم می‌دهد که با احتمالی بتواند راه حل‌های بدتر را نیز کاوش کند تا شاید به نقطه‌ای بهینه برسد.

۳-٤-٤ عملگرهای تغییر

استفاده از عملگرهای ریاضی و انتخاب‌های تصادفی برای جهش و بازترکیب به ما امکان کاوش و بهره‌برداری هم‌زمان از فضای جستجو را می‌دهد.

۳-٤-۵ روش مقداردهی اولیه جمعیت

مقداردهی اولیه جمعیت به صورت تصادفی به ما امکان پوشش و جستجو در نقاط مختلف از فضای حالت را می‌دهد.

۳-٤-٦ شرط توقف

با تغییر نکردن میانگین برازندگی نسل‌های متفاوت برای پنج بار پشت سر هم می‌توان الگوریتم را متوقف کرد تا در زمان و منابع صرفه‌جویی شود.

۳-۵- ث) عملکرد مجموعه قوانین

پس از استخراج پایگاه قوانین و به دست آوردن مدل‌سازی‌های زبانی، می‌توان دسته‌بندی پیام‌ها را با استفاده از این سامانه فازی آغاز کرد. لازم به ذکر است از آن‌جایی که برای ایجاد پایگاه قوانین از

الگوریتم تکاملی استفاده شده است، هر بار اجرای الگوریتم با تنظیمات‌های یکسان می‌تواند نتیجه مختلفی داشته باشد.

۳-۵-۱ توضیح اجتمالی کد قسمت پیش‌بینی‌کننده (تابع predict)

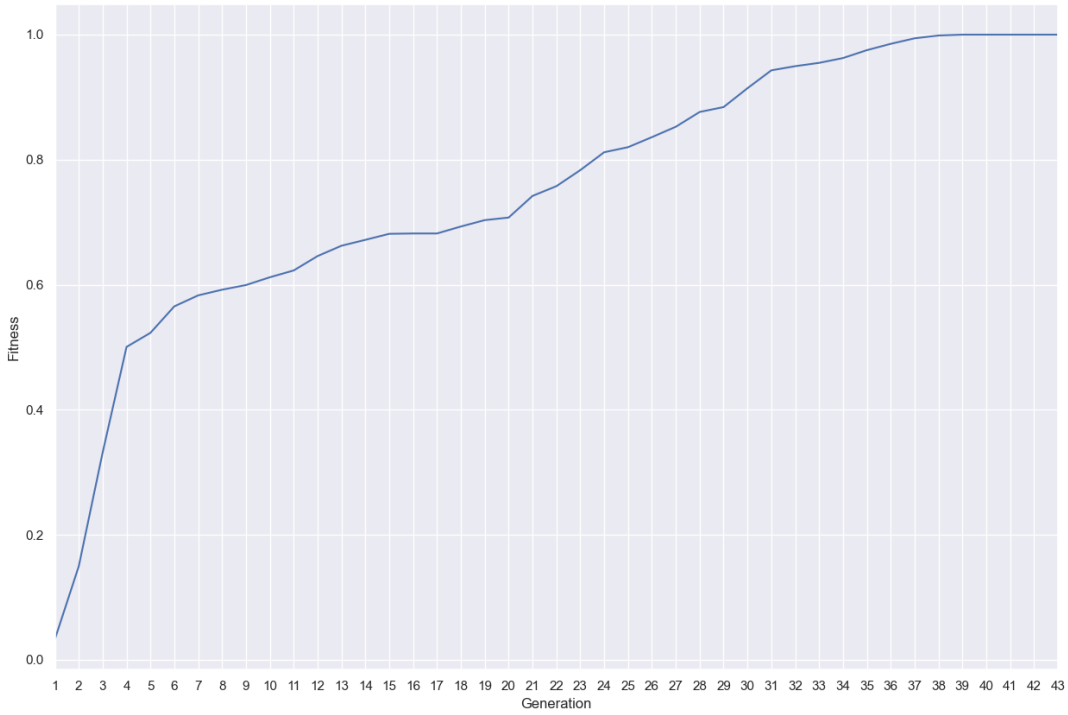
این تابع با دریافت جمعیت نهایی در الگوریتم تکاملی که پایگاه قوانین را تشکیل می‌دهد، با توجه به کلاس هر قانون میزان تطابق آن قانون را با داده ورودی طبق فرمول ۲ محاسبه کرده و کلاس مربوط به آن داده را برمی‌گرداند.

۳-۵-۲ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۹، جمعیت ۱۰۰، تعداد نسل ۲۰۰، انتخاب ویژگی و نمونه کاهی

در این روش با استفاده از نمونه‌کاهی تلاش بر این شده است تا توازن میان داده‌های کلاس یک و صفر برقرار شود و الگوریتم تکاملی به سمت یک کلاس متمایل نشود.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۴۰ با میانگین برازندگی ۱ به همگرایی رسیده و متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.9_0.9_100_200/undersample_featureSelection/

linguistic_model_0.9_0.9_100_200_undersample_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.58

f1_score: 0.28

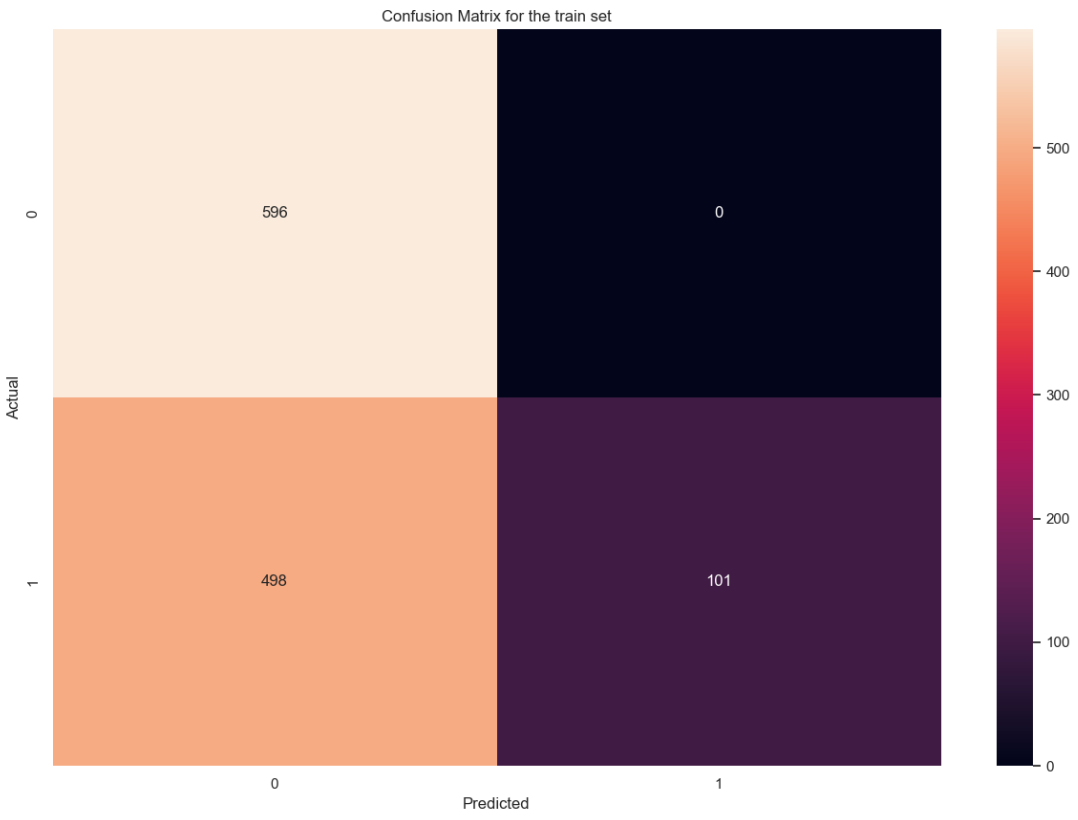
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.57

f1_score: 0.23

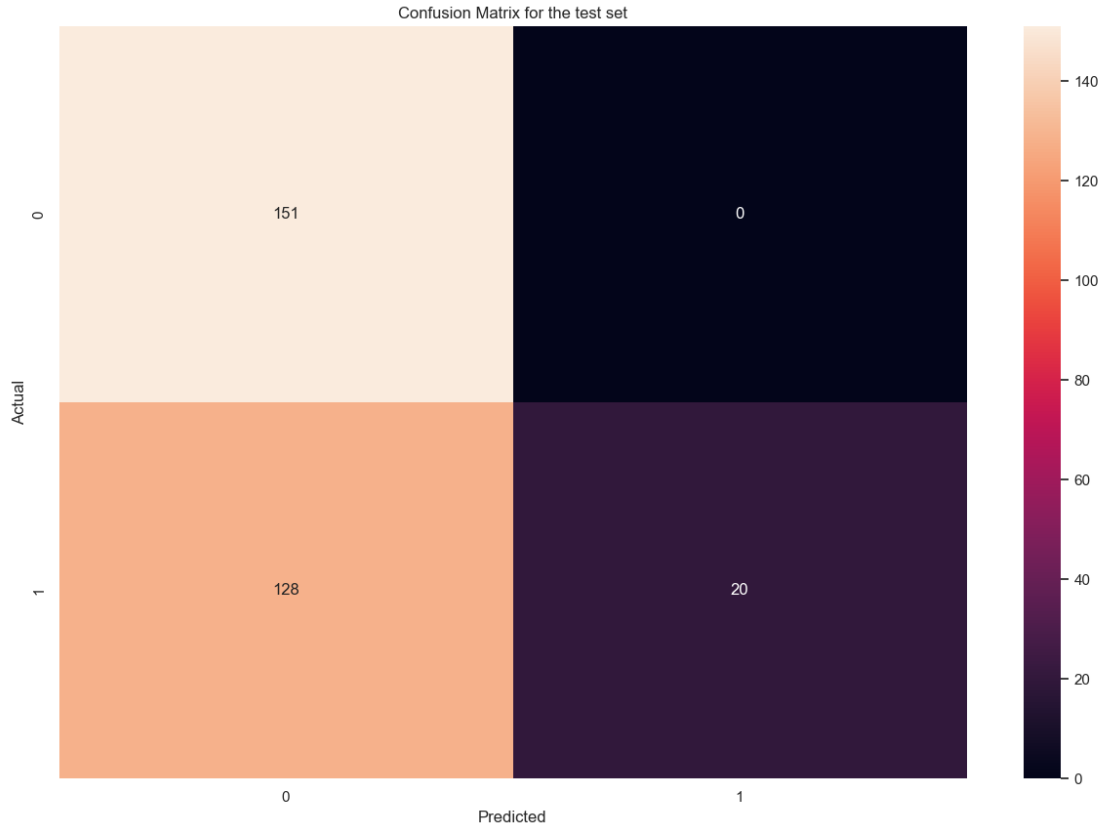
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیراسپم تشخیص داده است.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیراسپم تشخیص داده است.

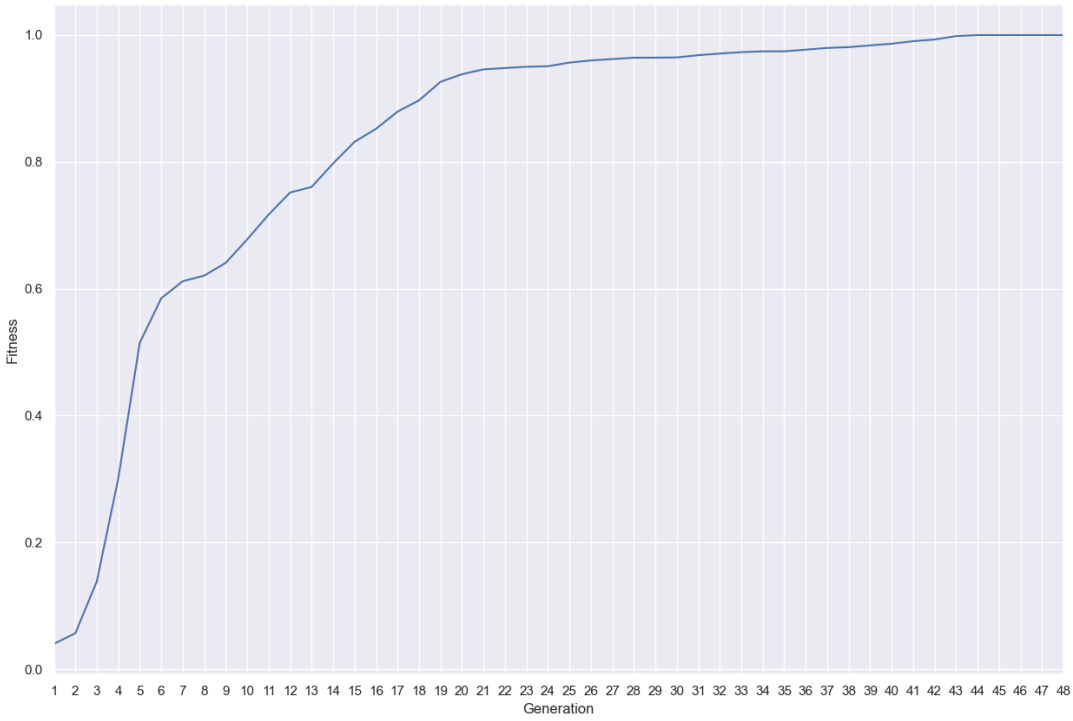


۳-۵-۳ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۹، جمعیت ۱۰۰، تعداد نسل ۲۰۰، انتخاب ویژگی و جداسازی دستی

در این روش ۵۰۰ داده از کلاس صفر و ۵۰۰ داده از کلاس یک به صورت تصادفی برای داده‌های آموزش انتخاب شده‌اند. همچنین ۲۰۰ داده از کلاس صفر و ۲۰۰ داده از کلاس یک به صورت تصادفی برای داده‌های تست انتخاب شده‌اند تا توازن میان داده‌های کلاس یک و صفر برقرار شود و الگوریتم تکاملی به سمت یک کلاس متمایل نشود. نتایج حاصل از اجرای الگوریتم با تنظیمات ذکر شده به شکل زیر است.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۴۰ با میانگین برازندگی ۱ به همگرایی رسیده و پس از پنج نسل ثابت بودن میانگین برازندگی متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.9_0.9_100_200/sample_featureSelection/

linguistic_model_0.9_0.9_100_200_sample_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.50

f1_score: 0.67

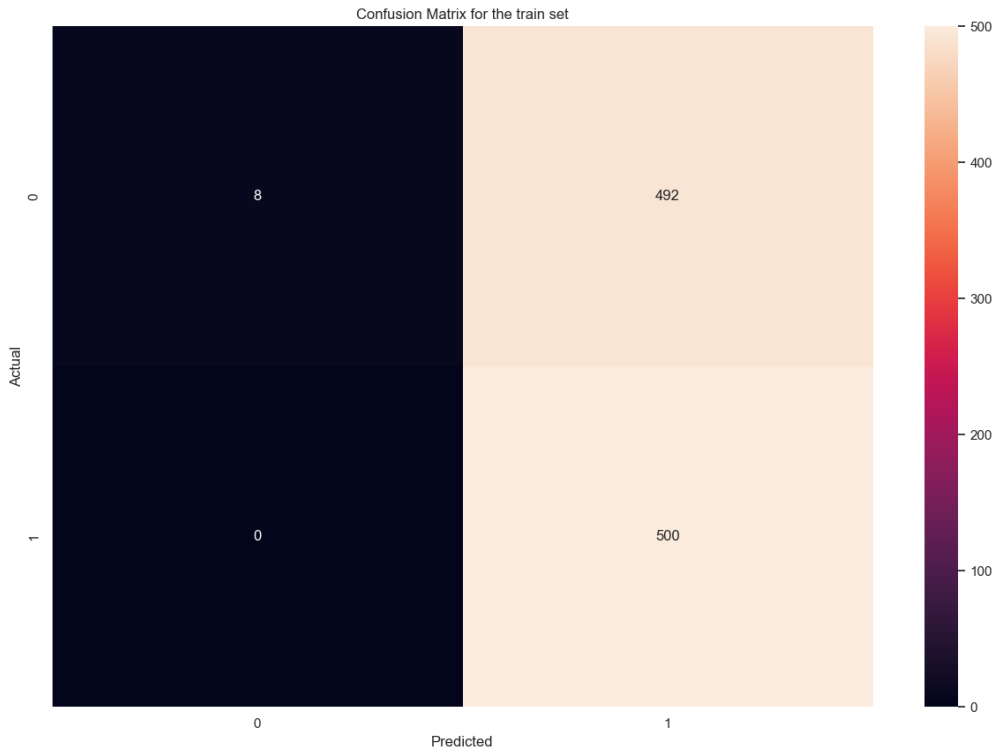
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.50

f1_score: 0.66

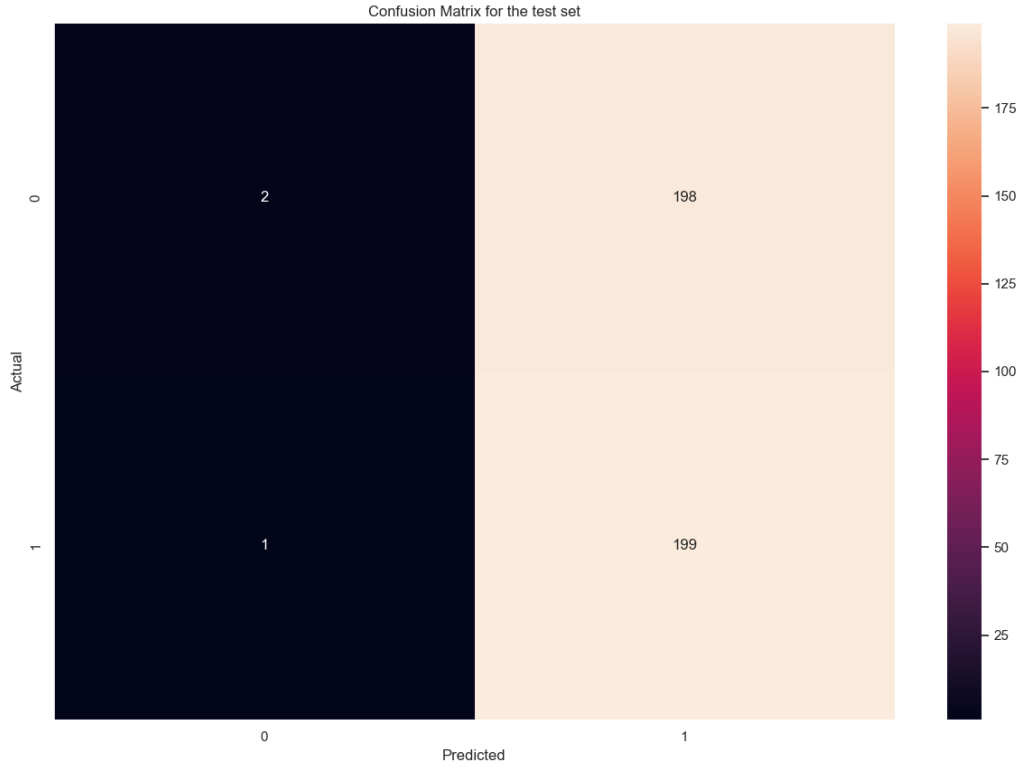
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های غیر اسپم را به اشتباه اسپم تشخیص داده است.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های غیر اسپم را به اشتباه اسپم تشخیص داده است.

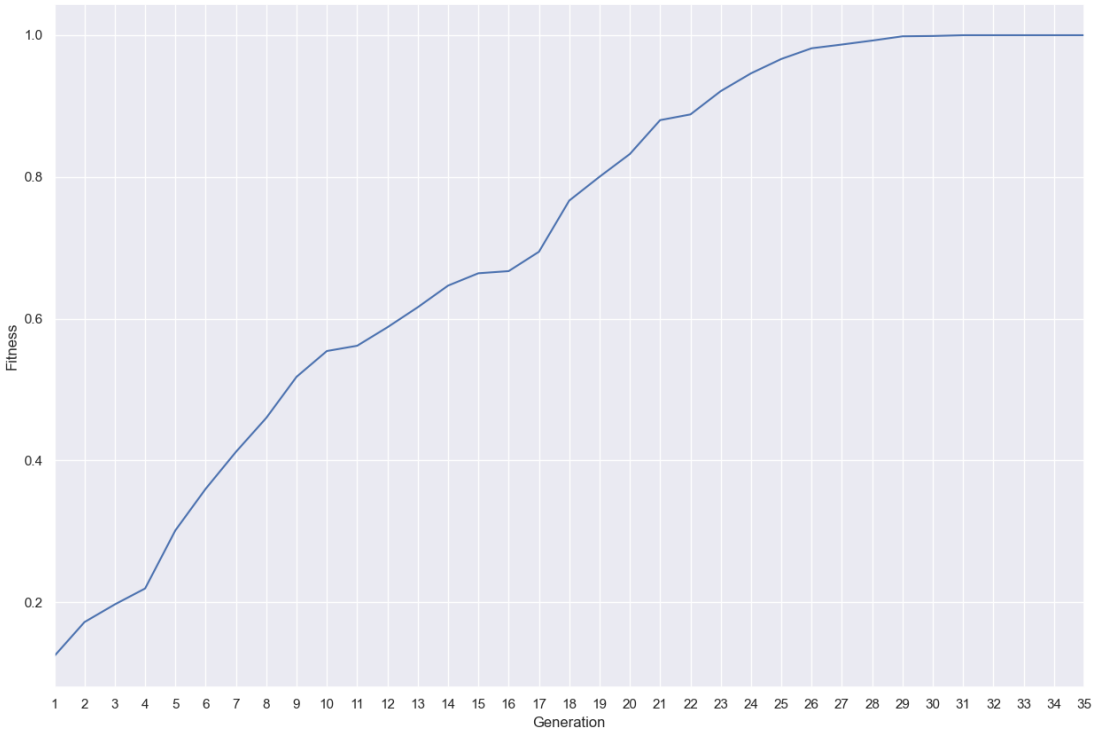


۳-۵-٤ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۹، جمعیت ۱۰۰، تعداد نسل ۲۰۰ و انتخاب ویژگی

در این روش هیچ متدی برای تنظیم تعادل داده‌ها استفاده نشده است. نتایج حاصل از اجرای الگوریتم با تنظیمات ذکر شده به شکل زیر است.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۳۰ با میانگین برازندگی ۱ به همگرایی رسیده و پس از ۵ نسل عدم تغییر میانگین برازندگی متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.9_0.9_100_200/featureSelection/

linguistic_model_0.9_0.9_100_200_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.89

f1_score: 0.37

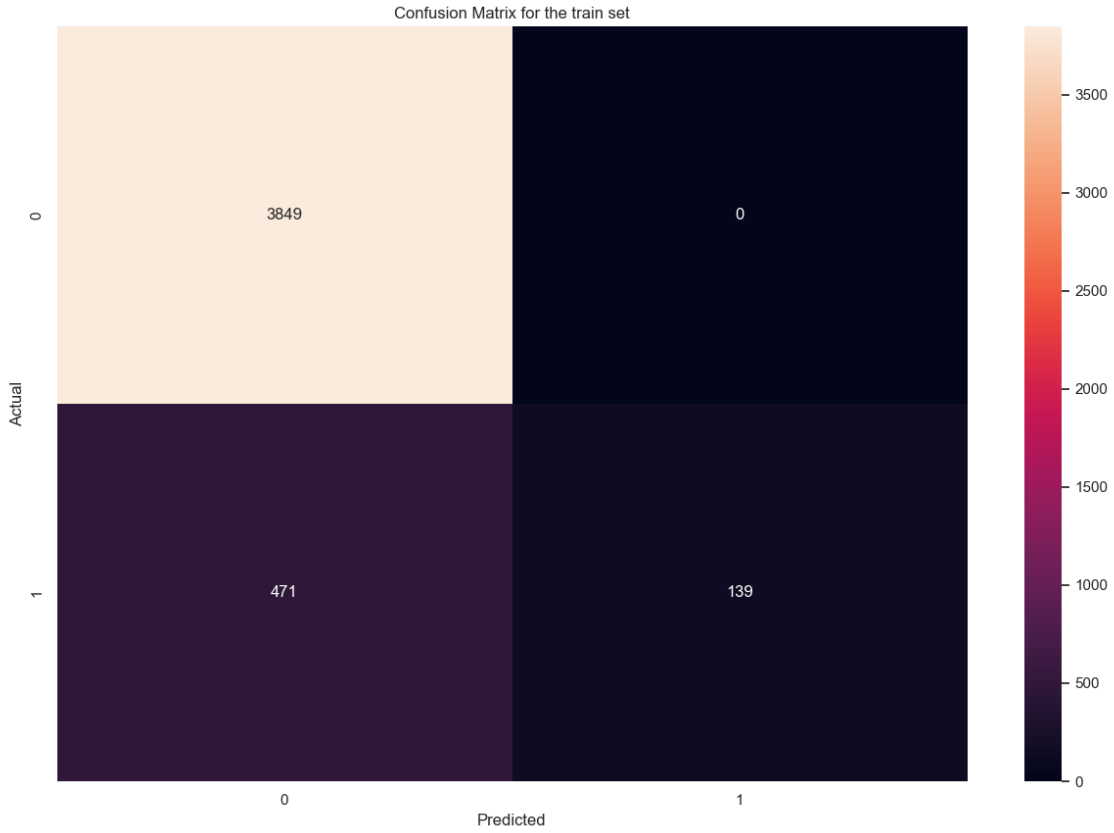
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.90

f1_score: 0.36

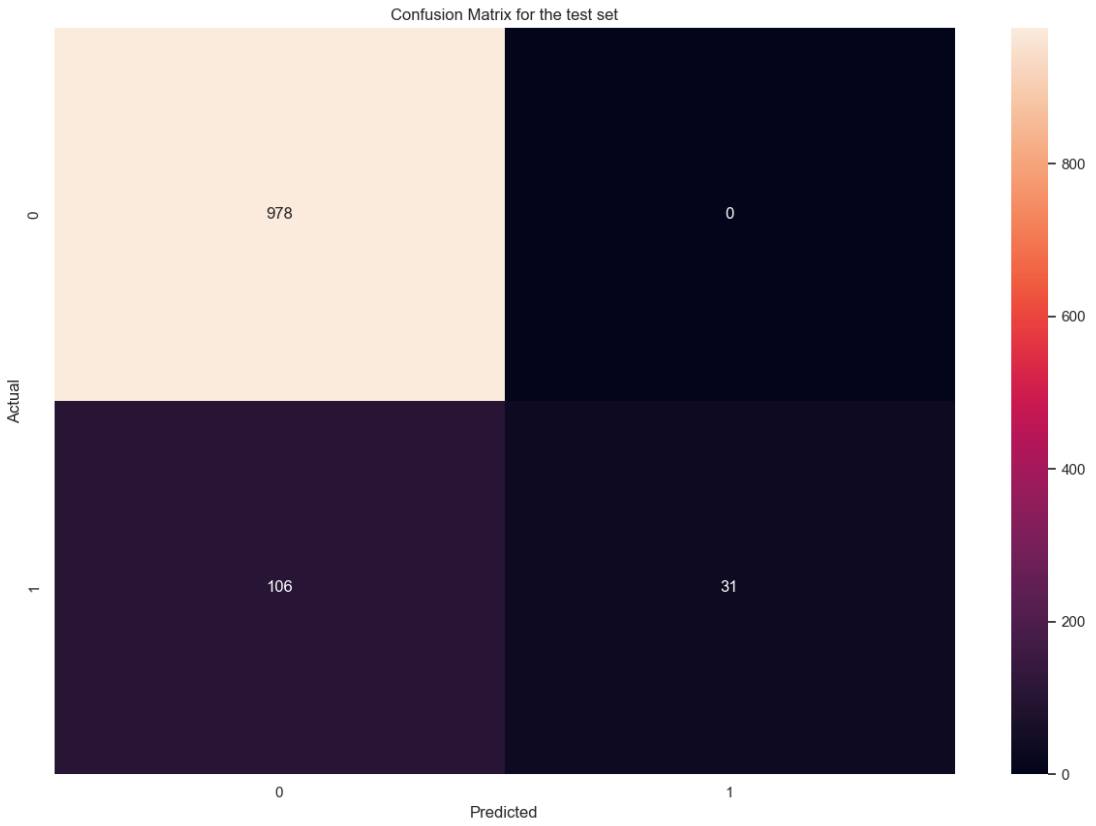
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است.

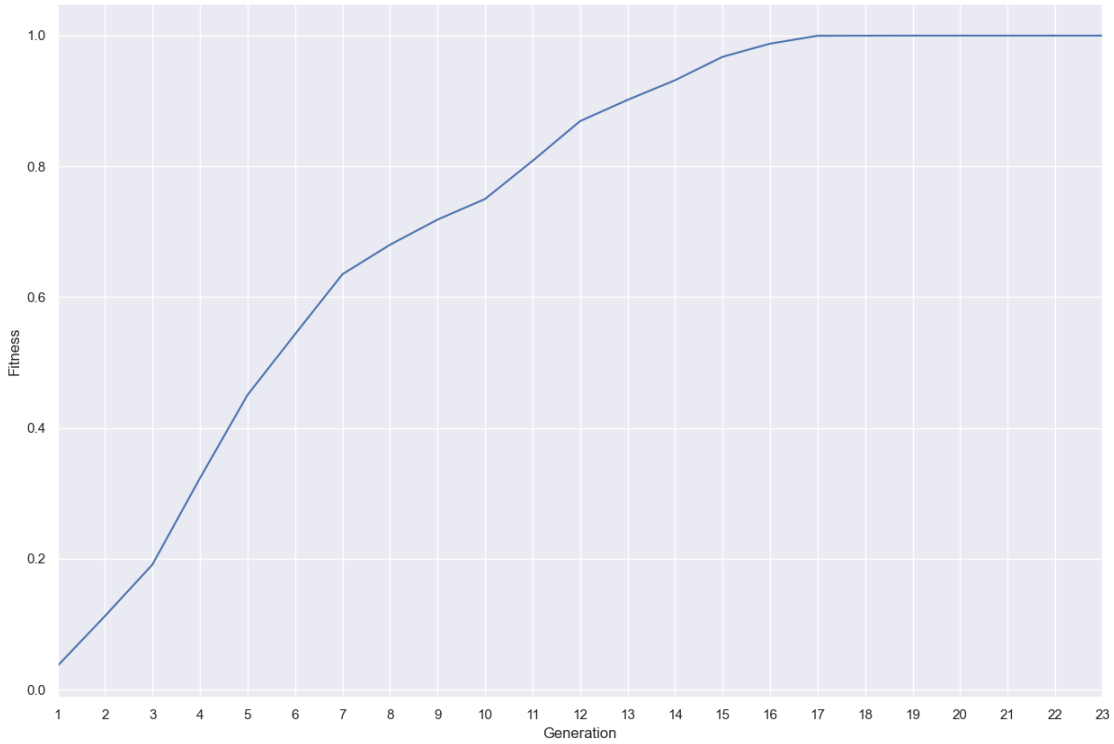


۳-۵-۵ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۱، جمعیت ۱۰۰، تعداد نسل ۲۰۰، انتخاب ویژگی و نمونه کاهی

در این روش با استفاده از نمونه‌کاهی تلاش بر این شده است تا توازن میان داده‌های کلاس یک و صفر برقرار شود و الگوریتم تکاملی به سمت یک کلاس متمایل نشود.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۴۰ با میانگین برازندگی ۱ به همگرایی رسیده و متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.9_0.1_100_200/undersample_featureSelection/

linguistic_model_0.9_0.1_100_200_undersample_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.50

f1_score: 0.66

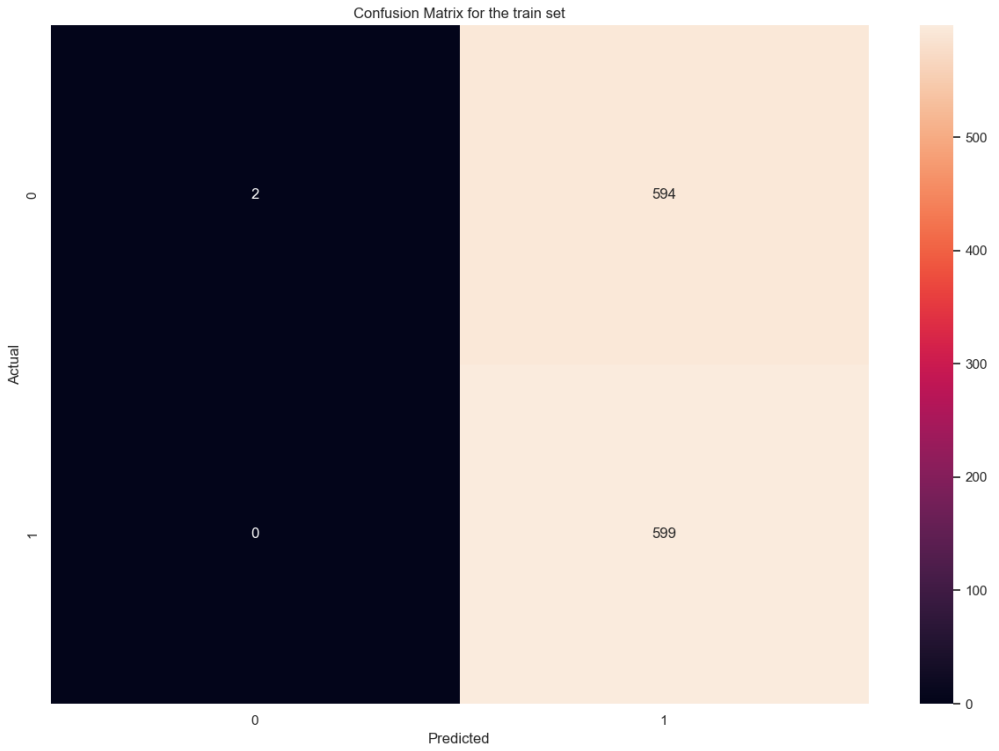
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.49

f1_score: 0.66

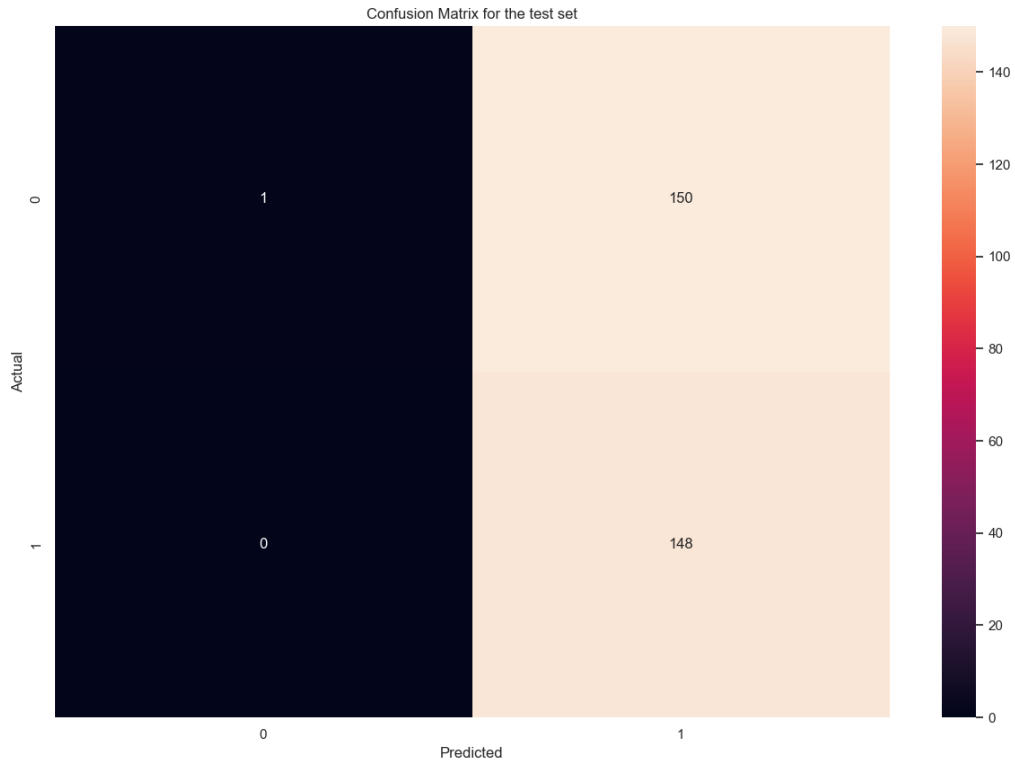
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های غیر اسپم را به اشتباه اسپم تشخیص داده است.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های غیر اسپم را به اشتباه اسپم تشخیص داده است.

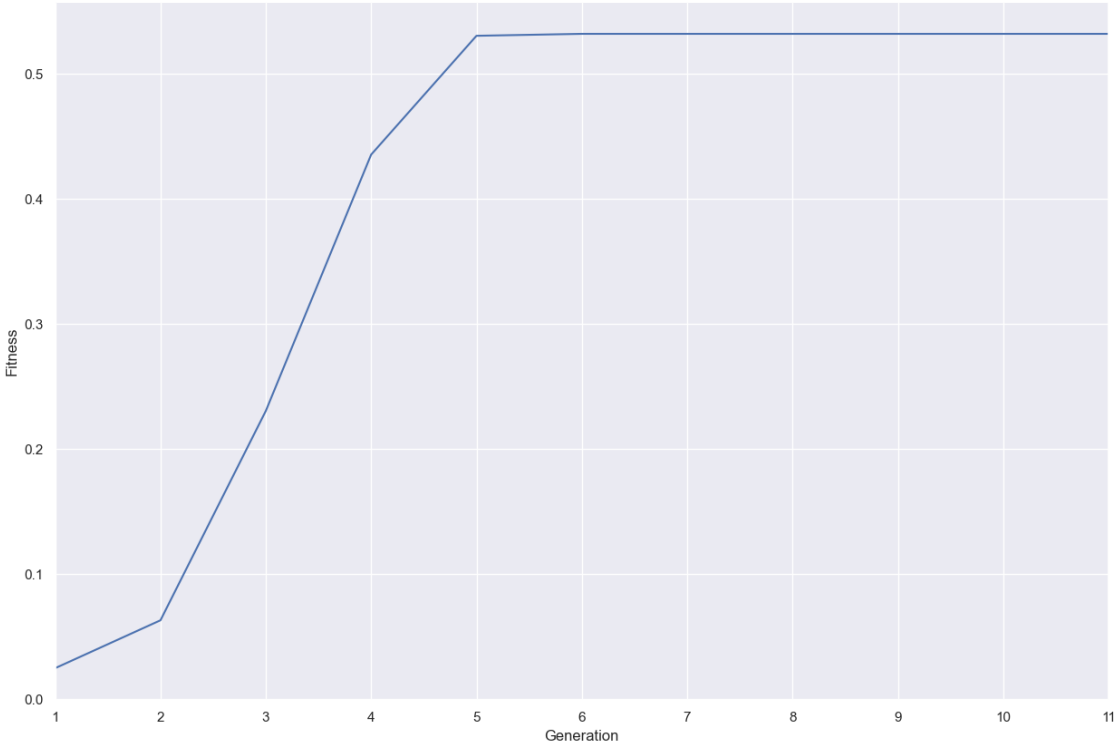


۳-۵-۶ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۱، جمعیت ۱۰۰، تعداد نسل ۲۰۰، انتخاب ویژگی و جداسازی دستی

در این روش ۵۰۰ داده از کلاس صفر و ۵۰۰ داده از کلاس یک به صورت تصادفی برای داده‌های آموزش انتخاب شده‌اند. همچنین ۲۰۰ داده از کلاس صفر و ۲۰۰ داده از کلاس یک به صورت تصادفی برای داده‌های تست انتخاب شده‌اند تا توازن میان داده‌های کلاس یک و صفر برقرار شود و الگوریتم تکاملی به سمت یک کلاس متمایل نشود. نتایج حاصل از اجرای الگوریتم با تنظیمات ذکر شده به شکل زیر است.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۵ با میانگین برازندگی ۱ به همگرایی رسیده و پس از پنج نسل ثابت بودن میانگین برازندگی متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.9_0.1_100_200/sample_featureSelection/

linguistic_model_0.9_0.1_100_200_sample_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.50

f1_score: 0.0

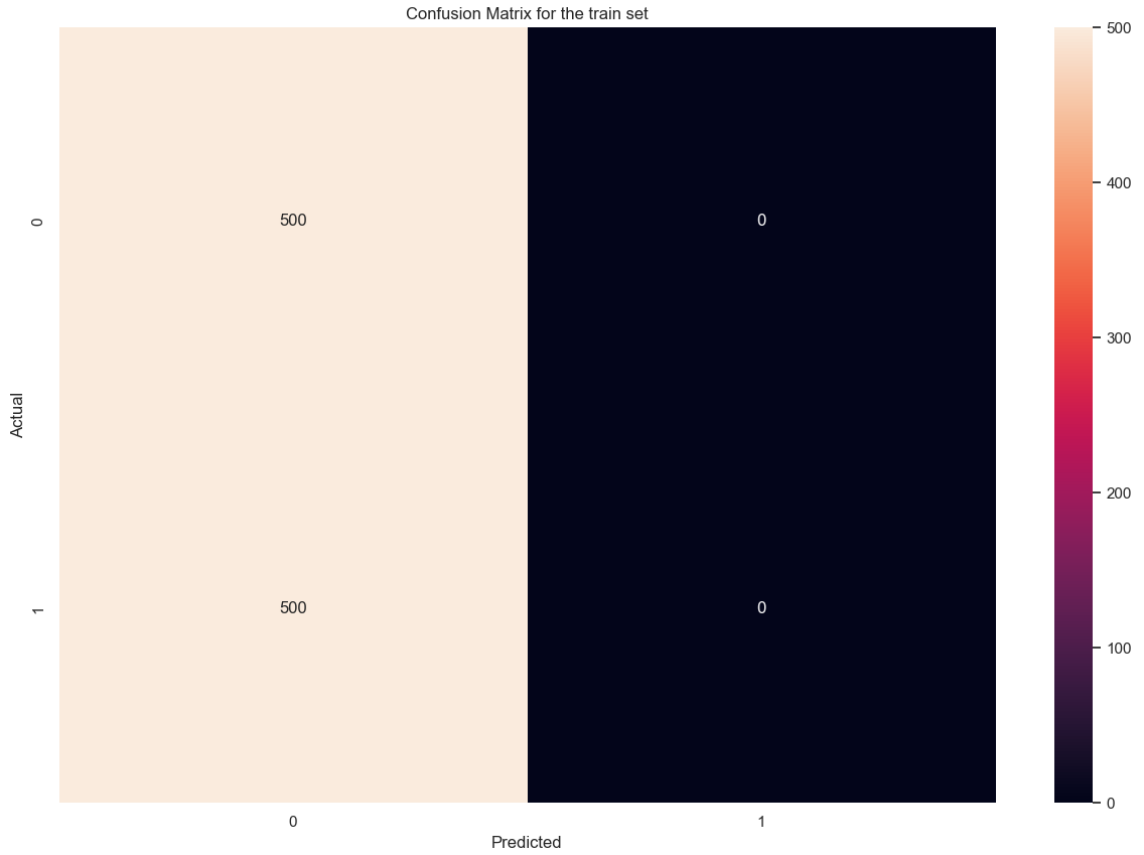
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.50

f1_score: 0.0

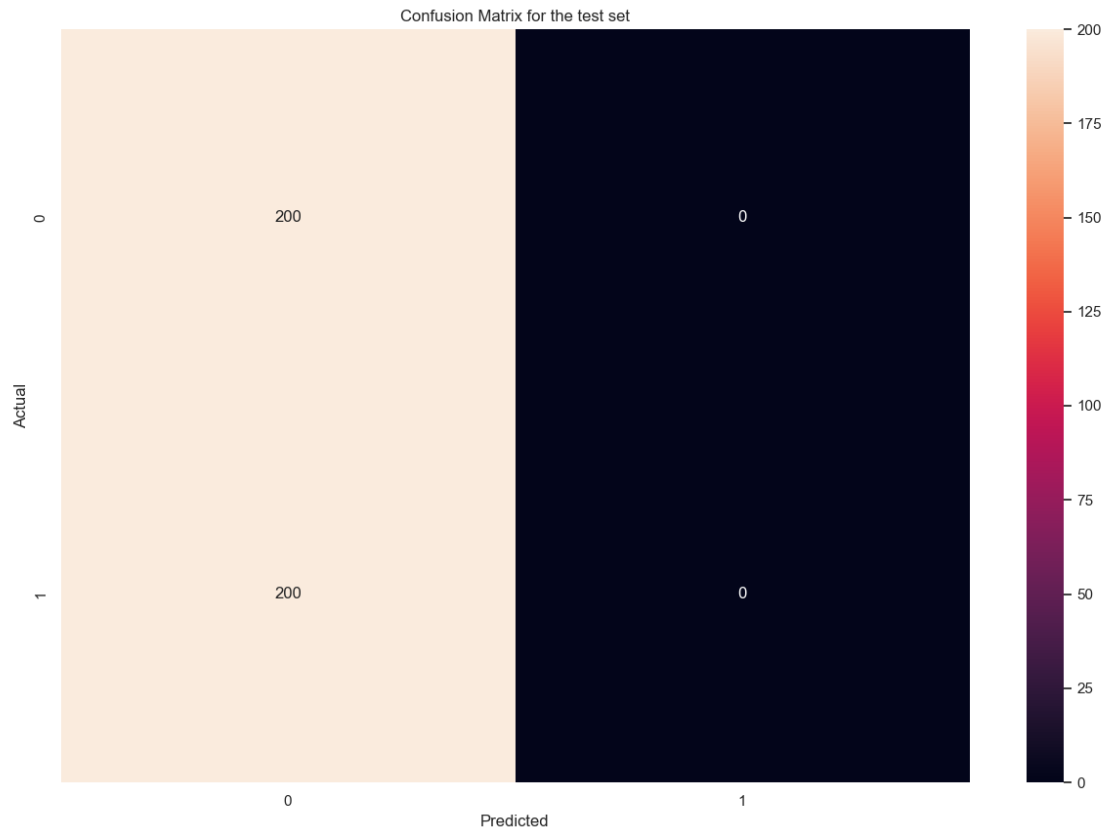
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل همه پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل همه پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است.

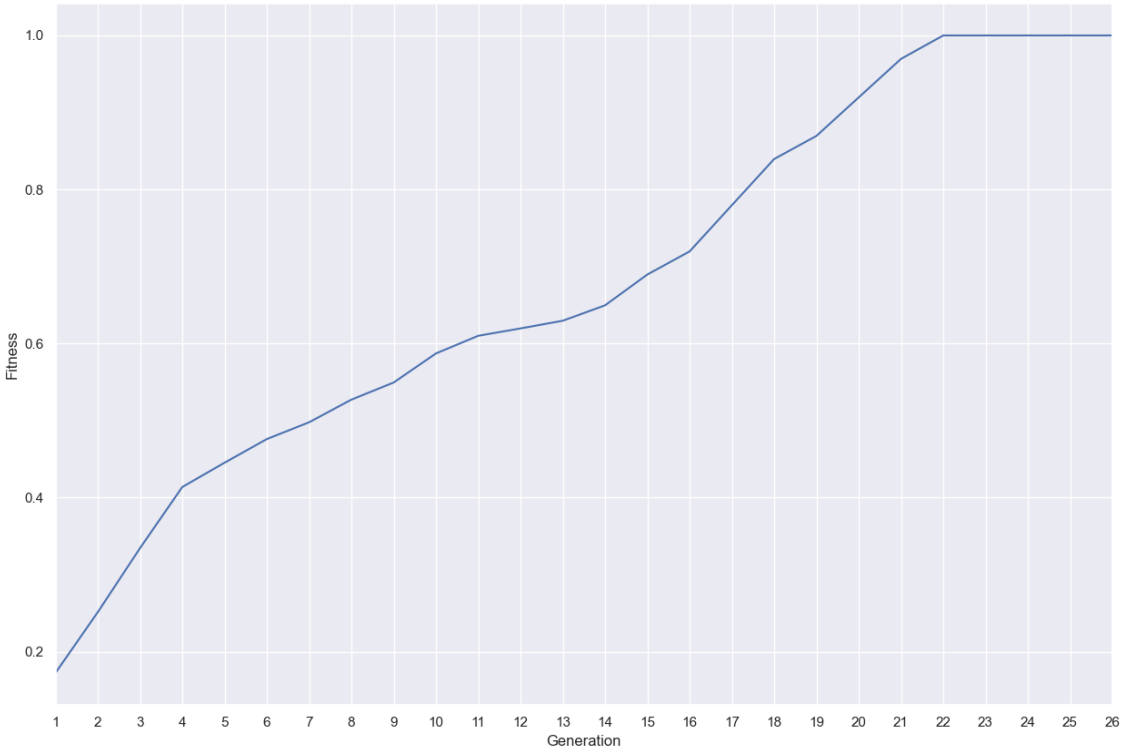


۳-۵-۷ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۱، جمعیت ۱۰۰، تعداد نسل ۲۰۰ و انتخاب ویژگی

در این روش هیچ متدی برای تنظیم تعادل داده‌ها استفاده نشده است. نتایج حاصل از اجرای الگوریتم با تنظیمات ذکر شده به شکل زیر است.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۳۰ با میانگین برازندگی ۱ به همگرایی رسیده و پس از ۵ نسل عدم تغییر میانگین برازندگی متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.9_0.1_100_200/featureSelection/

linguistic_model_0.9_0.9_100_200_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.86

f1_score: 0.03

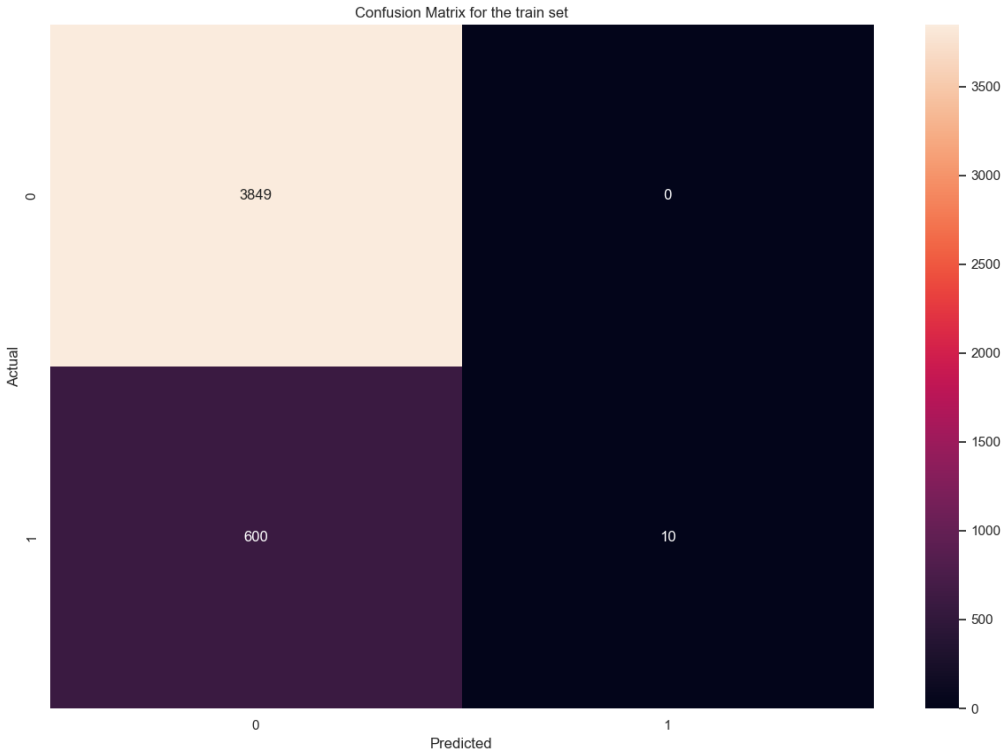
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.87

f1_score: 0.02

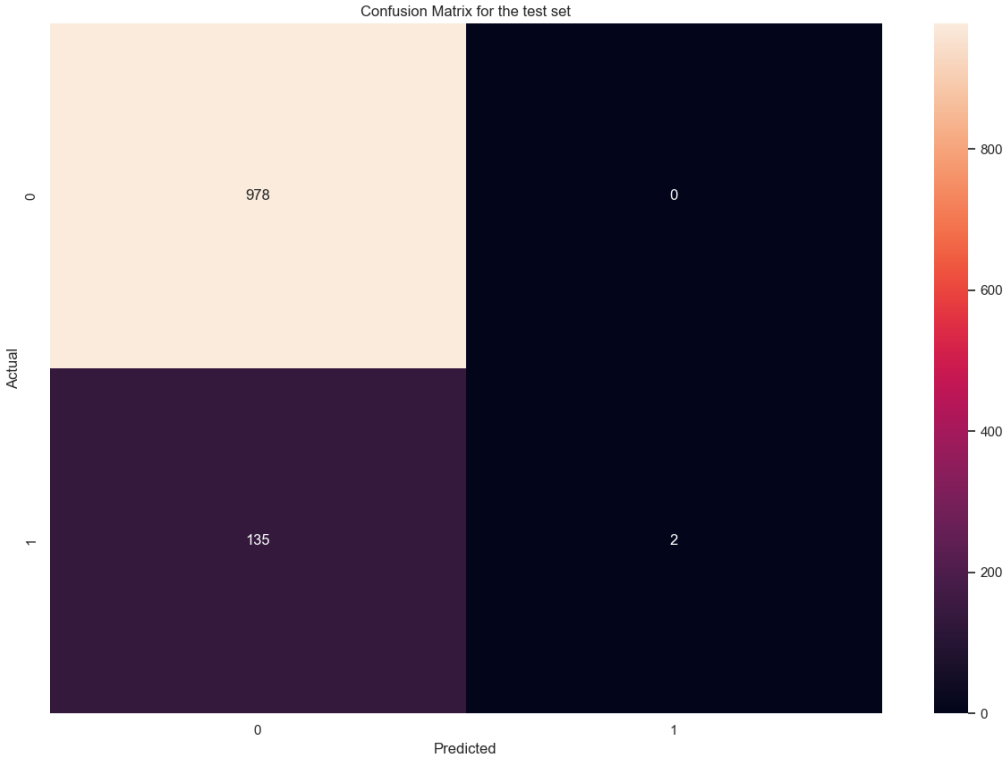
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است.

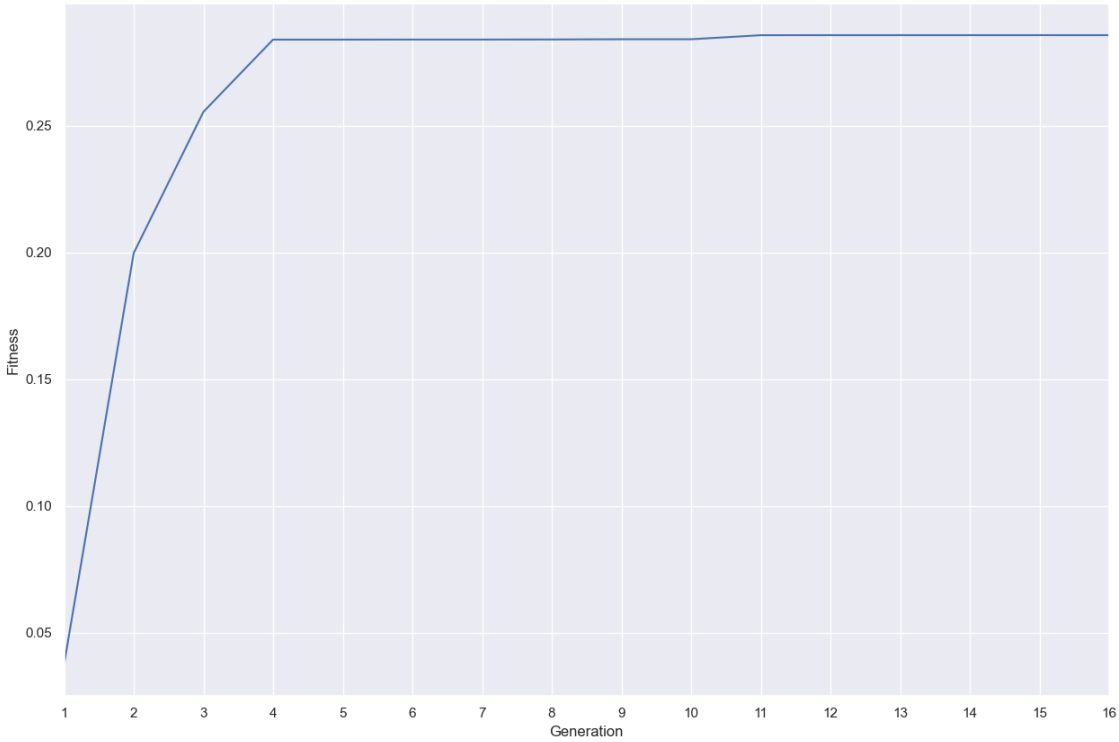


۳-۵-۸ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۱، بازترکیب ۰/۹، جمعیت ۱۰۰، تعداد نسل ۲۰۰، انتخاب ویژگی و نمونه کاهی

در این روش با استفاده از نمونه‌کاهی تلاش بر این شده است تا توازن میان داده‌های کلاس یک و صفر برقرار شود و الگوریتم تکاملی به سمت یک کلاس متمایل نشود.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۴۰ با میانگین برازندگی ۱ به همگرایی رسیده و متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.1_0.9_100_200/undersample_featureSelection/

linguistic_model_0.1_0.9_100_200_undersample_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.50

f1_score: 0.66

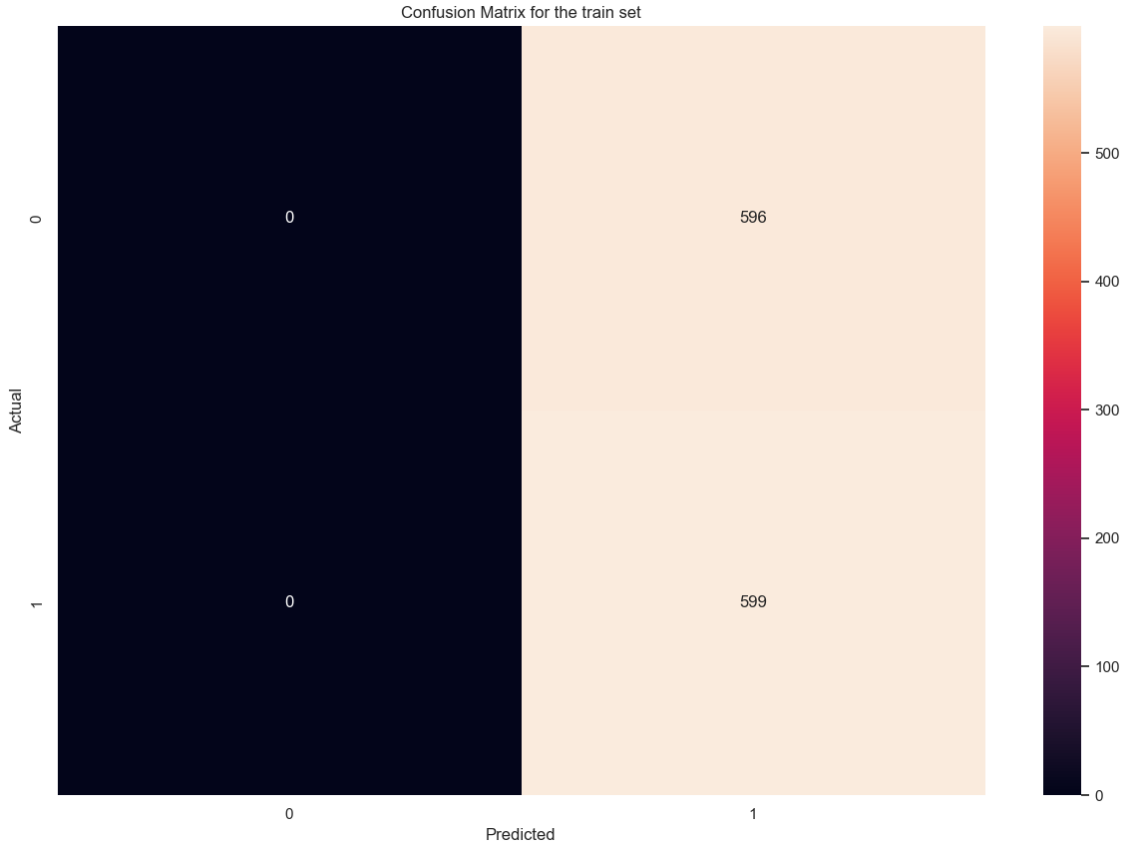
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.49

f1_score: 0.66

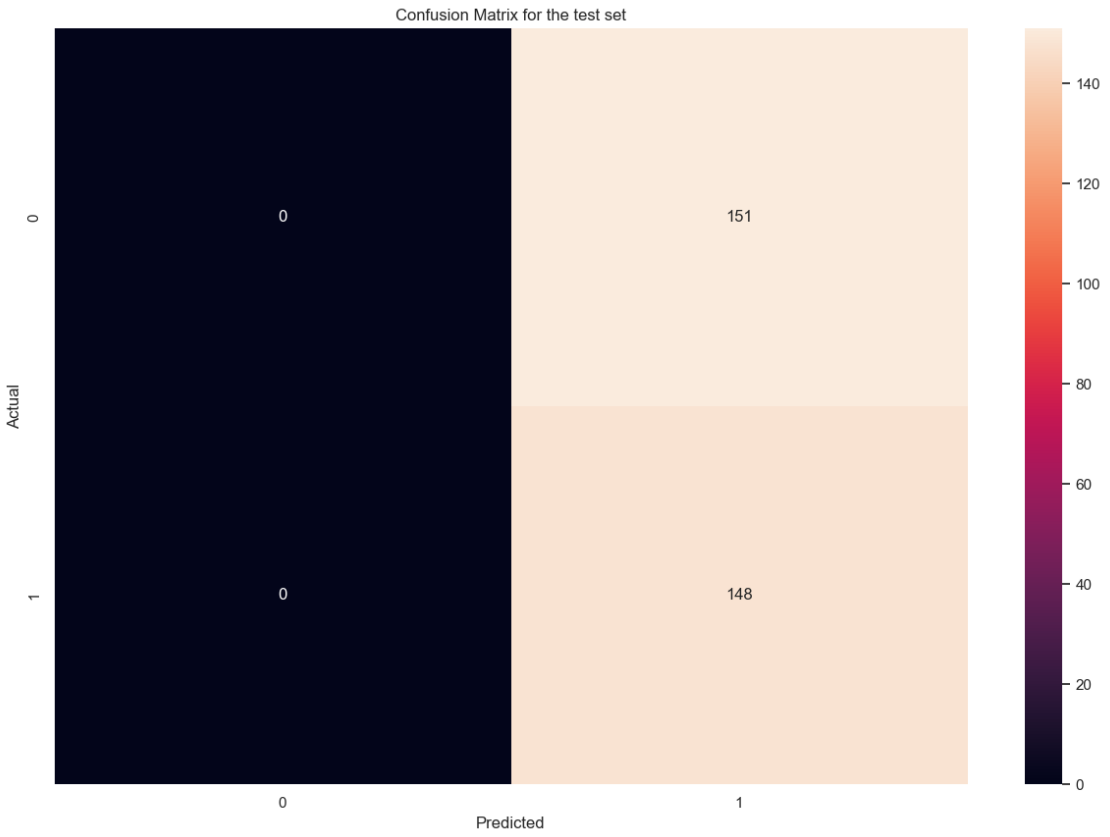
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل نتوانسته هیچ‌کدام از پیام‌های غیراسپم را شناسایی کند.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل نتوانسته هیچ‌کدام از پیام‌های غیراسپم را شناسایی کند.

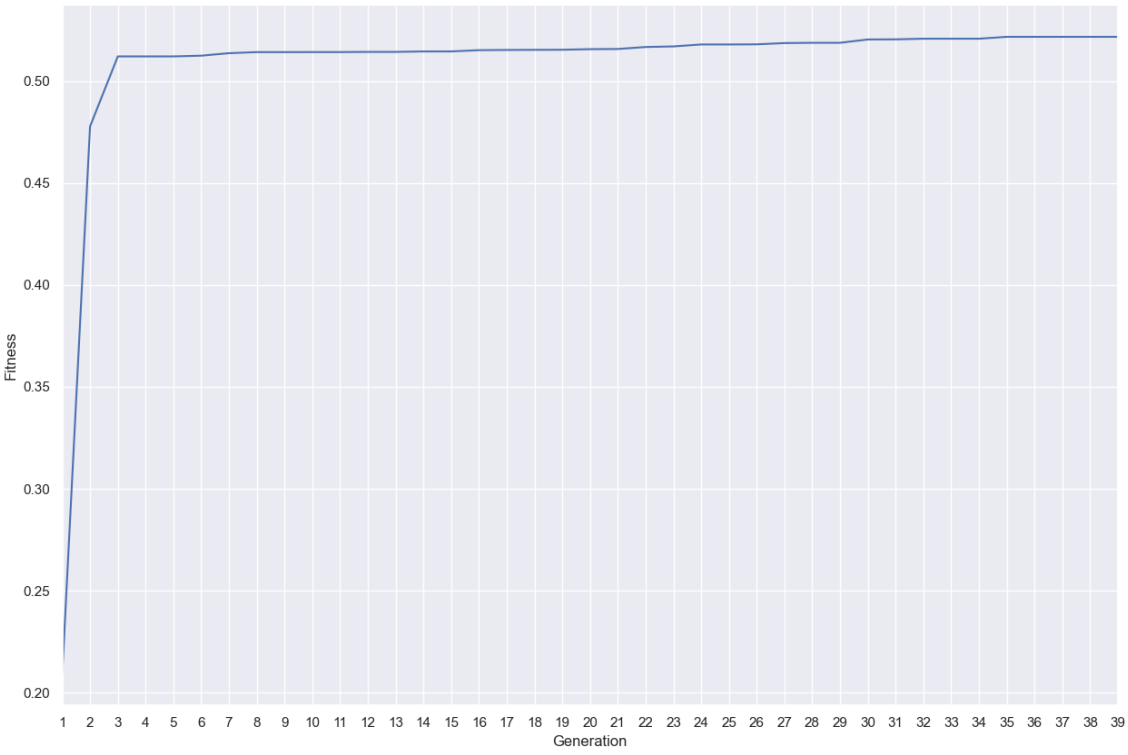


۳-۵-۹ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۱، بازترکیب ۰/۹، جمعیت ۱۰۰، تعداد نسل ۲۰۰، انتخاب ویژگی و جداسازی دستی

در این روش ۵۰۰ داده از کلاس صفر و ۵۰۰ داده از کلاس یک به صورت تصادفی برای داده‌های آموزش انتخاب شده‌اند. همچنین ۲۰۰ داده از کلاس صفر و ۲۰۰ داده از کلاس یک به صورت تصادفی برای داده‌های تست انتخاب شده‌اند تا توازن میان داده‌های کلاس یک و صفر برقرار شود و الگوریتم تکاملی به سمت یک کلاس متمایل نشود. نتایج حاصل از اجرای الگوریتم با تنظیمات ذکر شده به شکل زیر است.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۳۶ با میانگین برازندگی ۰/۵۲ به همگرایی رسیده و پس از پنج نسل ثابت بودن میانگین برازندگی متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.1_0.9_100_200/sample_featureSelection/

linguistic_model_0.1_0.9_100_200_sample_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.50

f1_score: 0.0

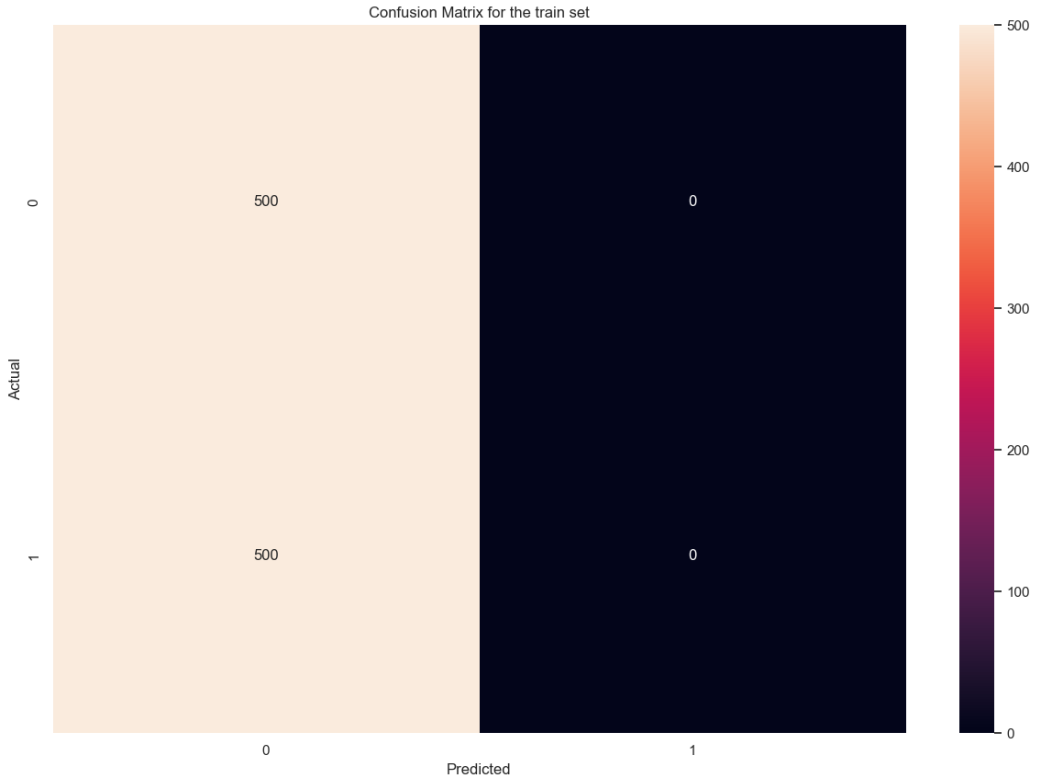
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.50

f1_score: 0.0

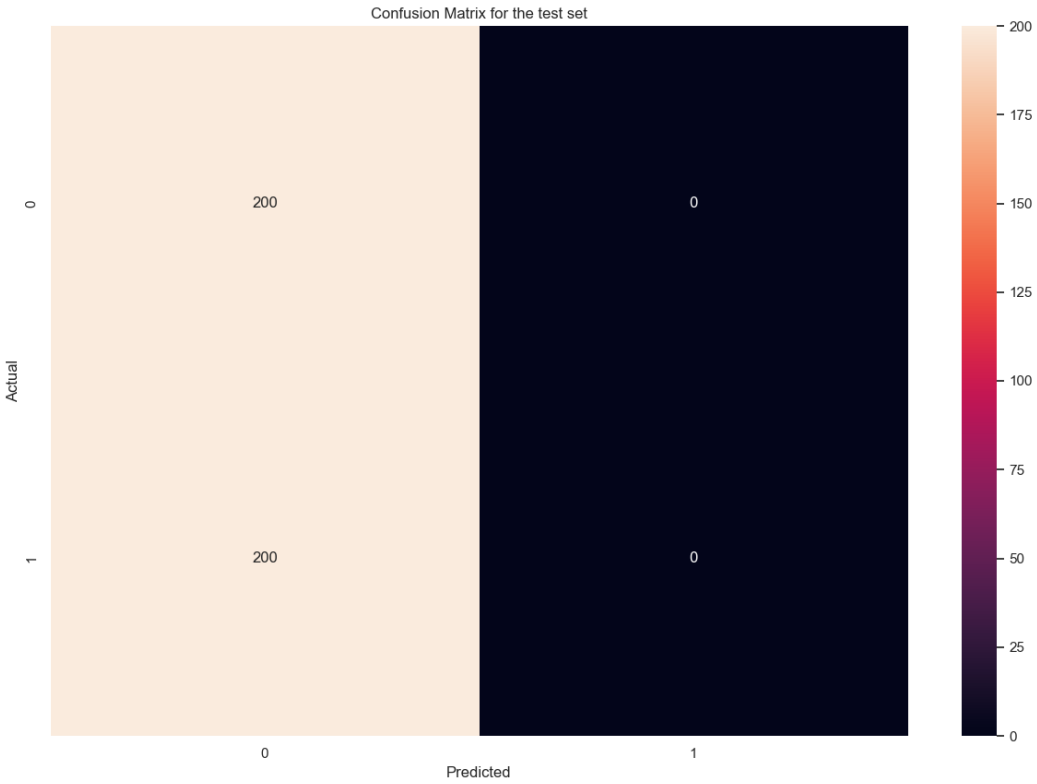
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل نتوانسته هیچ پیام اسمپی را تشخیص دهد.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل نتوانسته هیچ پیام اسمپی را تشخیص دهد.

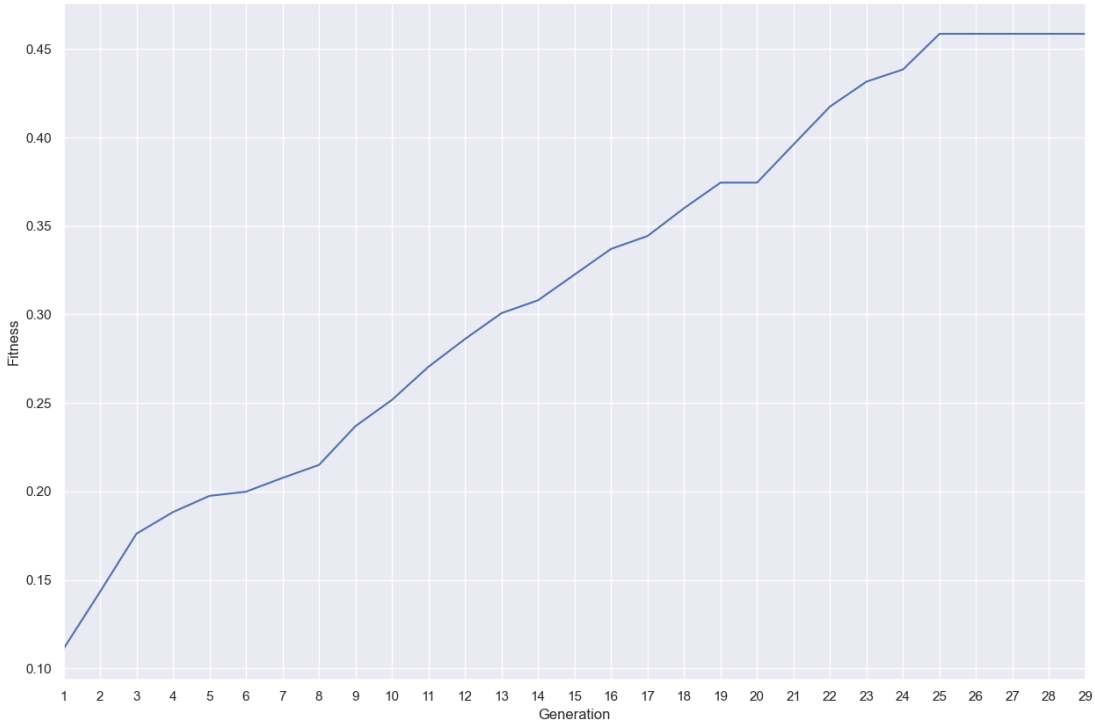


۳-۵-۱۰ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۱، بازترکیب ۰/۹، جمعیت ۱۰۰، تعداد نسل ۲۰۰ و انتخاب ویژگی

در این روش هیچ متدی برای تنظیم تعادل داده‌ها استفاده نشده است. نتایج حاصل از اجرای الگوریتم با تنظیمات ذکر شده به شکل زیر است.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۲۵ با میانگین برازندگی ۰/۴۵ به همگرایی رسیده و پس از ۵ نسل عدم تغییر میانگین برازندگی متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.1_0.9_100_200/featureSelection/

linguistic_model_0.1_0.9_100_200_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.86

f1_score: 0.0

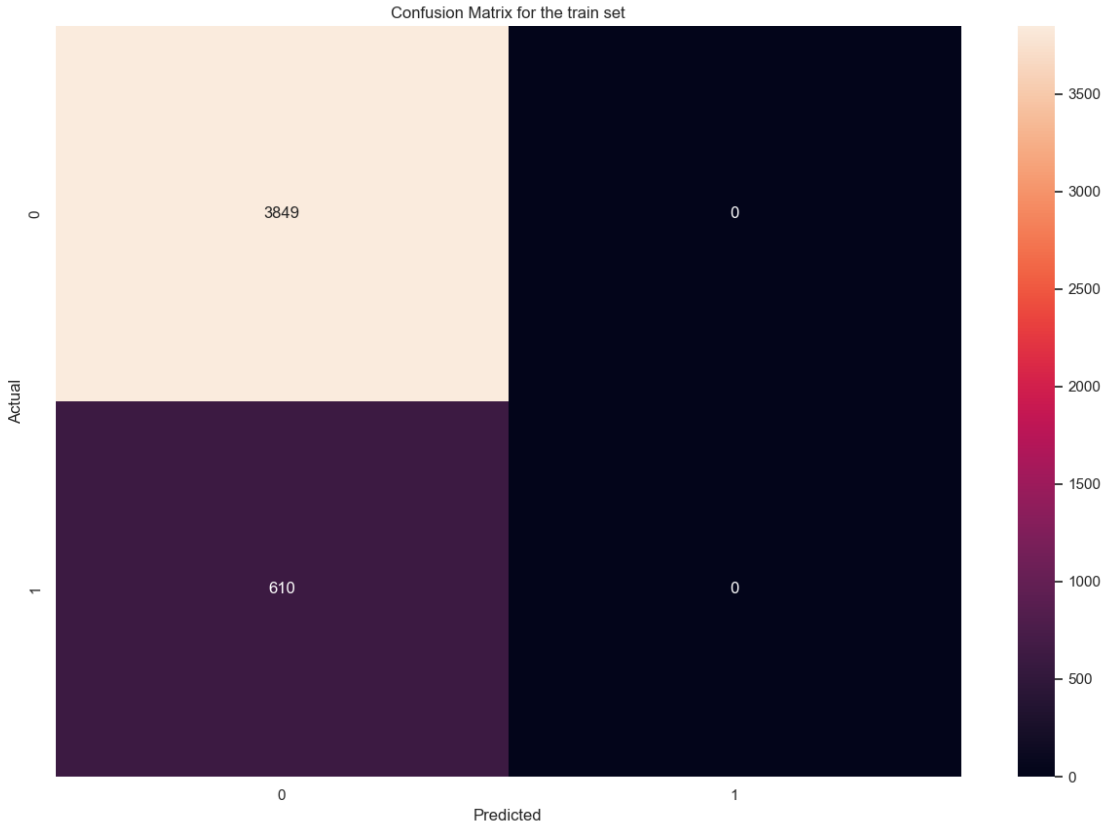
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.87

f1_score: 0.0

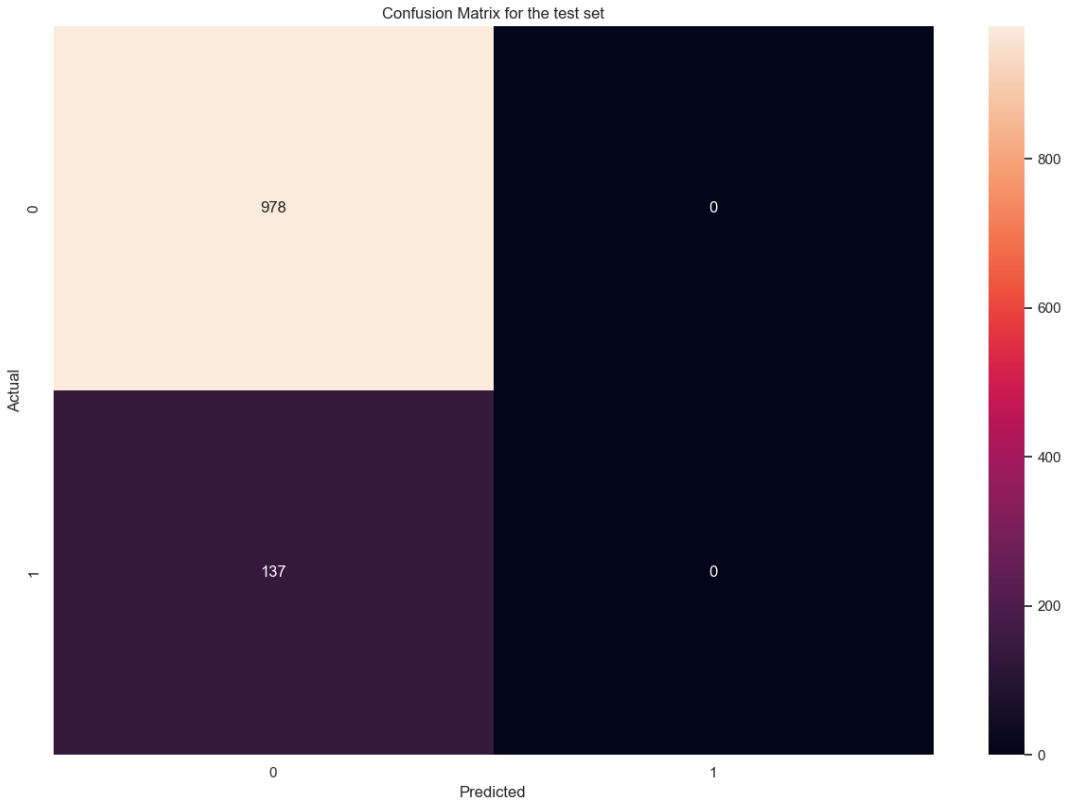
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است.

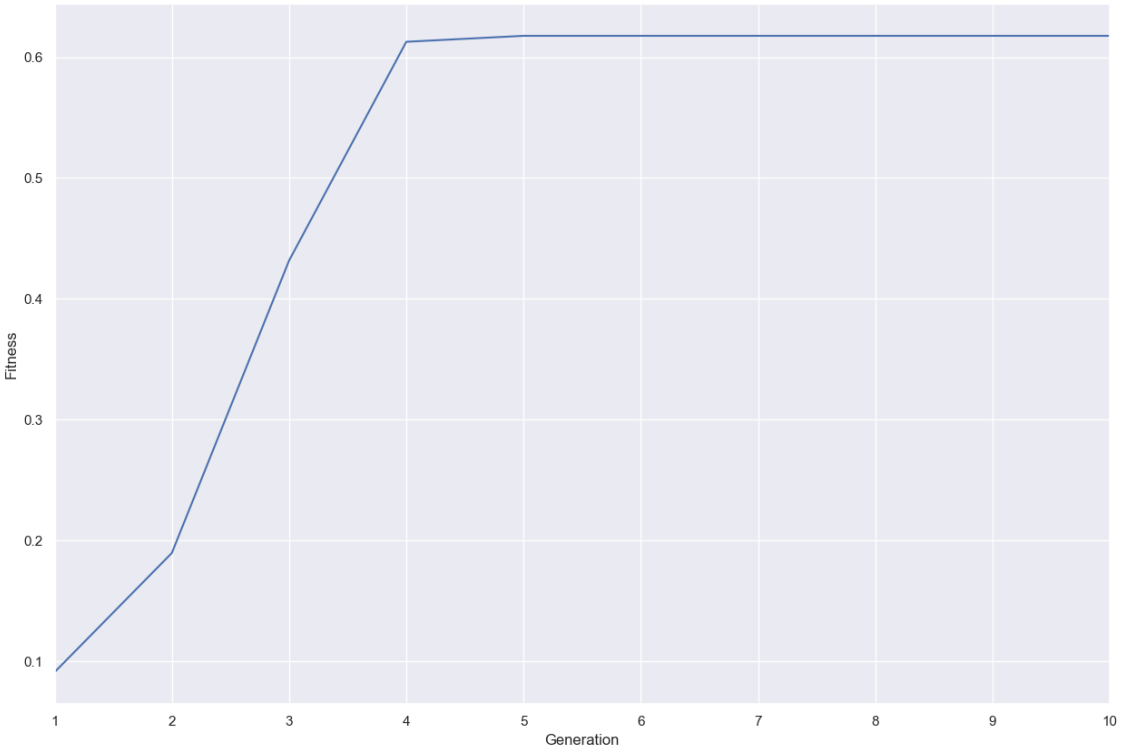


۳-۵-۱۱ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۵، بازترکیب ۰/۵، جمعیت ۱۰۰، تعداد نسل ۲۰۰، انتخاب ویژگی و نمونه کاهی

در این روش با استفاده از نمونه‌کاهی تلاش بر این شده است تا توازن میان داده‌های کلاس یک و صفر برقرار شود و الگوریتم تکاملی به سمت یک کلاس متمایل نشود.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۶ با میانگین برازندگی ۰/۶۳ به همگرایی رسیده و متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.5_0.5_100_200/undersample_featureSelection/

linguistic_model_0.5_0.5_100_200_undersample_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.49

f1_score: 0.0

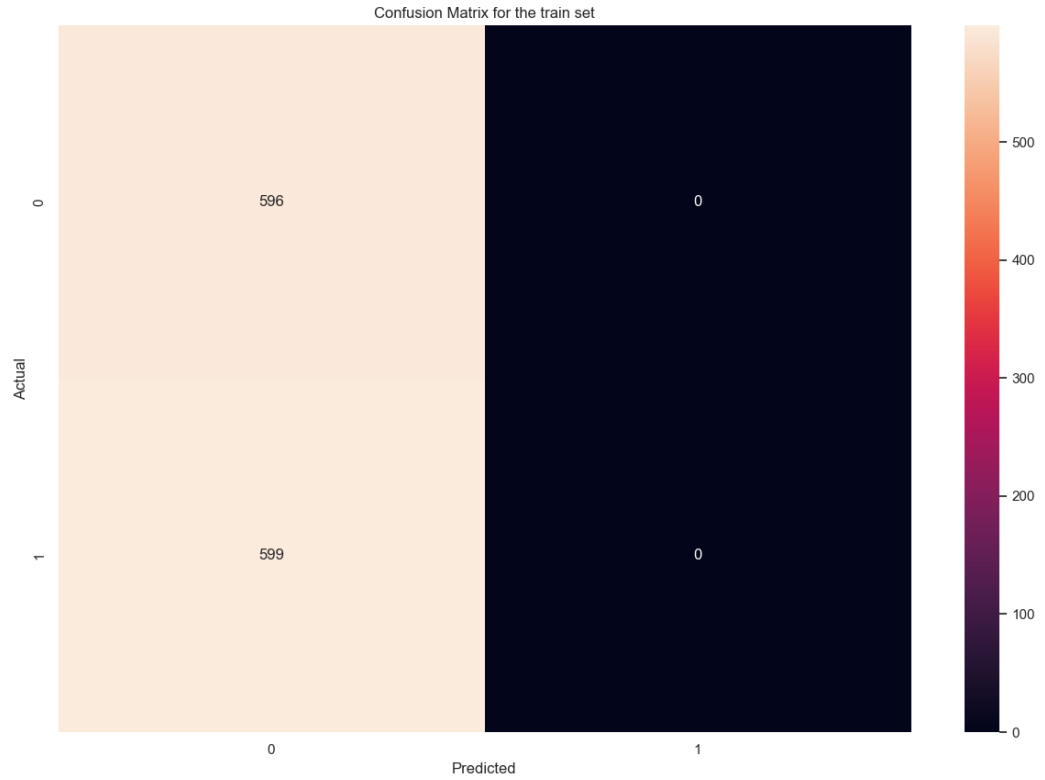
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.50

f1_score: 0.0

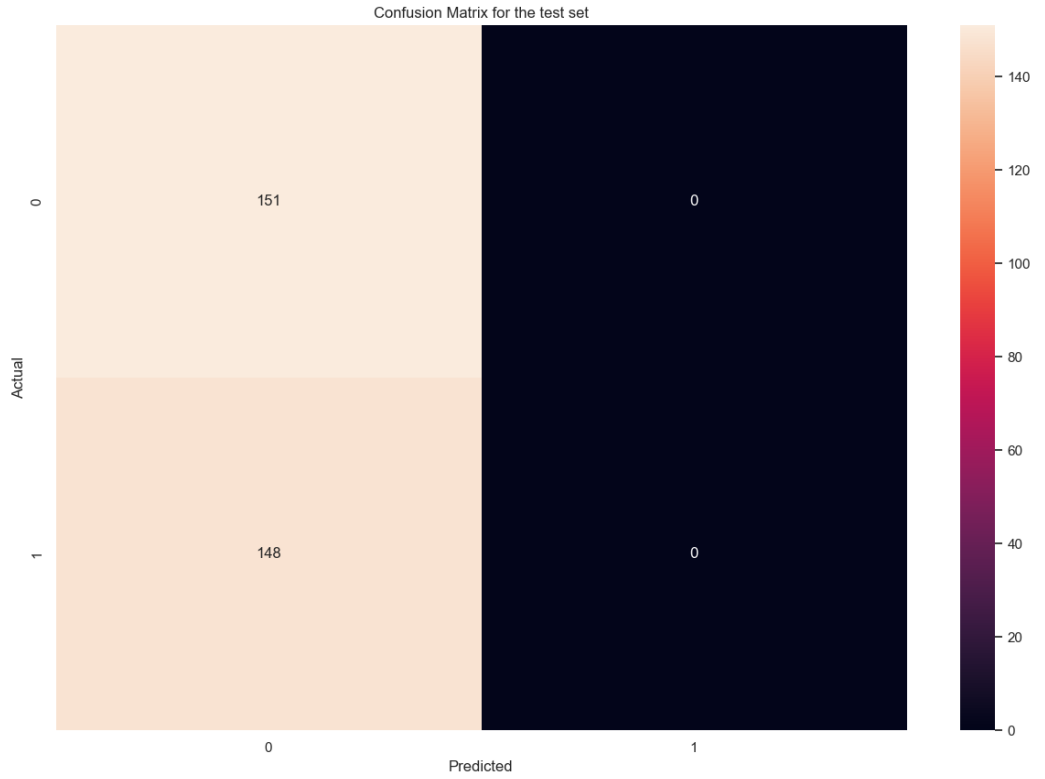
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل نتوانسته هیچ یک از پیام‌های اسپم را تشخیص دهد.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل نتوانسته هیچ یک از پیام‌های اسپم را تشخیص دهد.

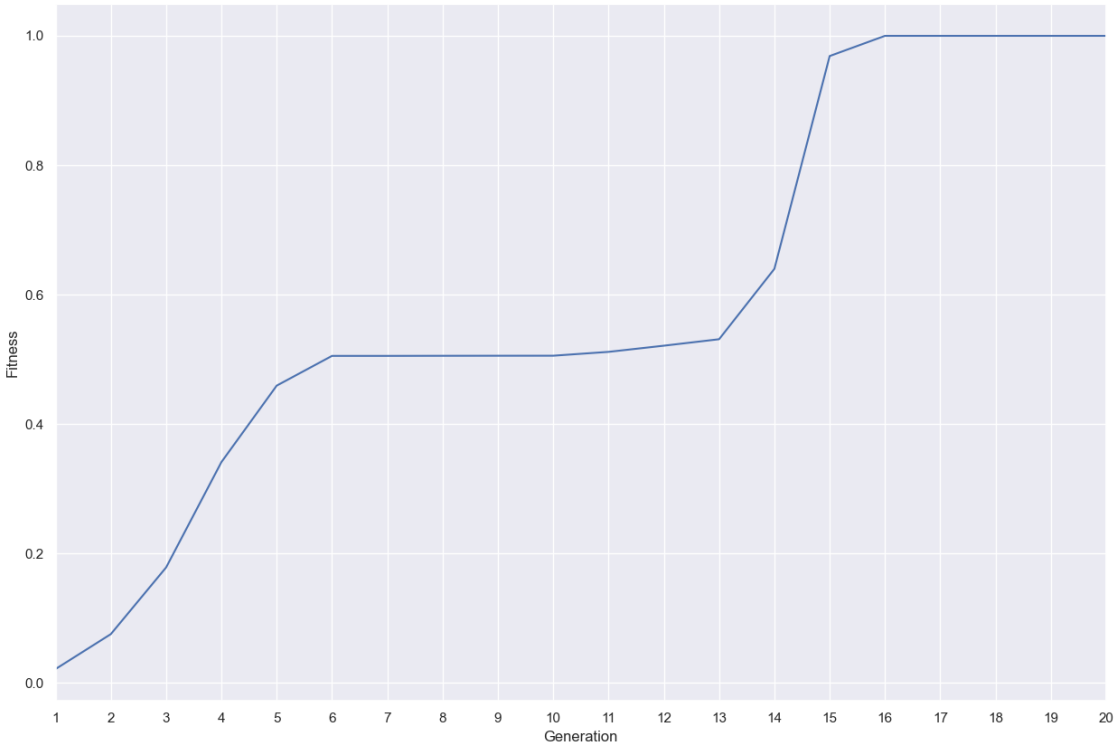


۳-۵-۱۲ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۵، بازترکیب ۰/۵، جمعیت ۱۰۰، تعداد نسل ۲۰۰، انتخاب ویژگی و جداسازی دستی

در این روش ۵۰۰ داده از کلاس صفر و ۵۰۰ داده از کلاس یک به صورت تصادفی برای داده‌های آموزش انتخاب شده‌اند. همچنین ۲۰۰ داده از کلاس صفر و ۲۰۰ داده از کلاس یک به صورت تصادفی برای داده‌های تست انتخاب شده‌اند تا توازن میان داده‌های کلاس یک و صفر برقرار شود و الگوریتم تکاملی به سمت یک کلاس متمایل نشود. نتایج حاصل از اجرای الگوریتم با تنظیمات ذکر شده به شکل زیر است.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۴۰ با میانگین برازندگی ۱ به همگرایی رسیده و پس از پنج نسل ثابت بودن میانگین برازندگی متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.5_0.5_100_200/sample_featureSelection/

linguistic_model_0.5_0.5_100_200_sample_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.64

f1_score: 0.45

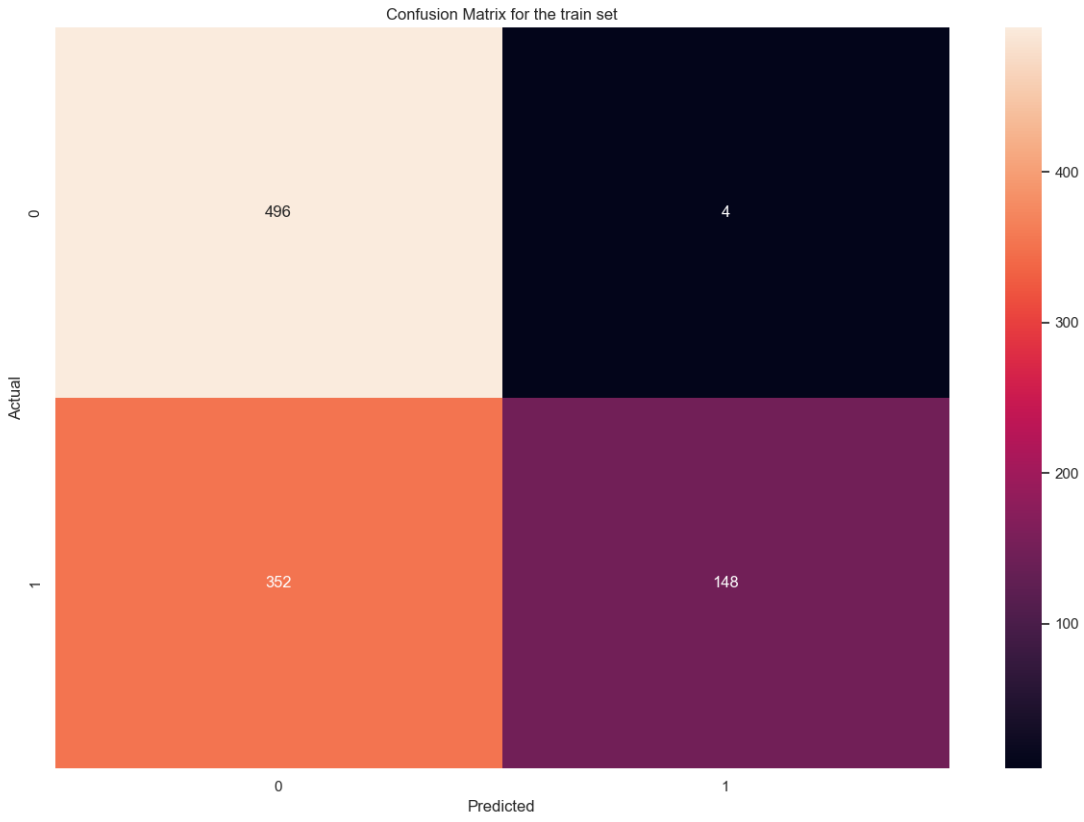
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.64

f1_score: 0.45

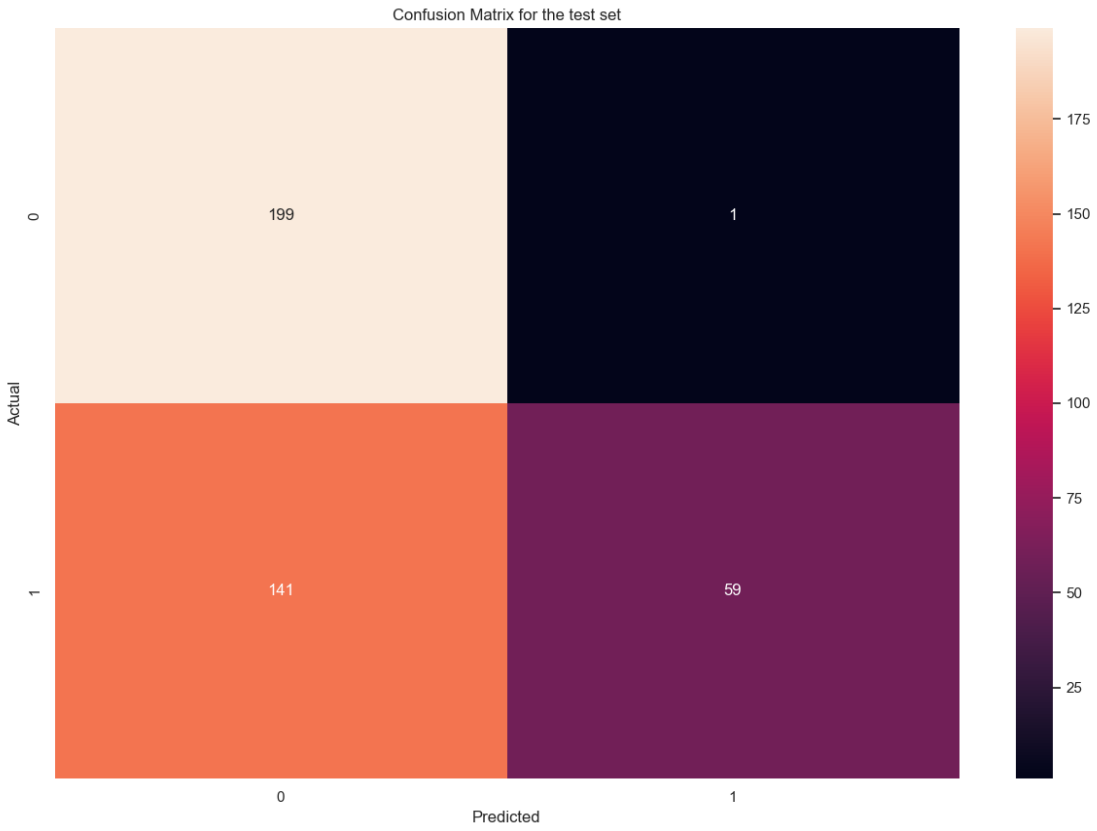
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است. اما قادر به تشخیص هر دو کلاس با احتمالی است.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است. اما قادر به تشخیص هر دو کلاس با احتمالی است.

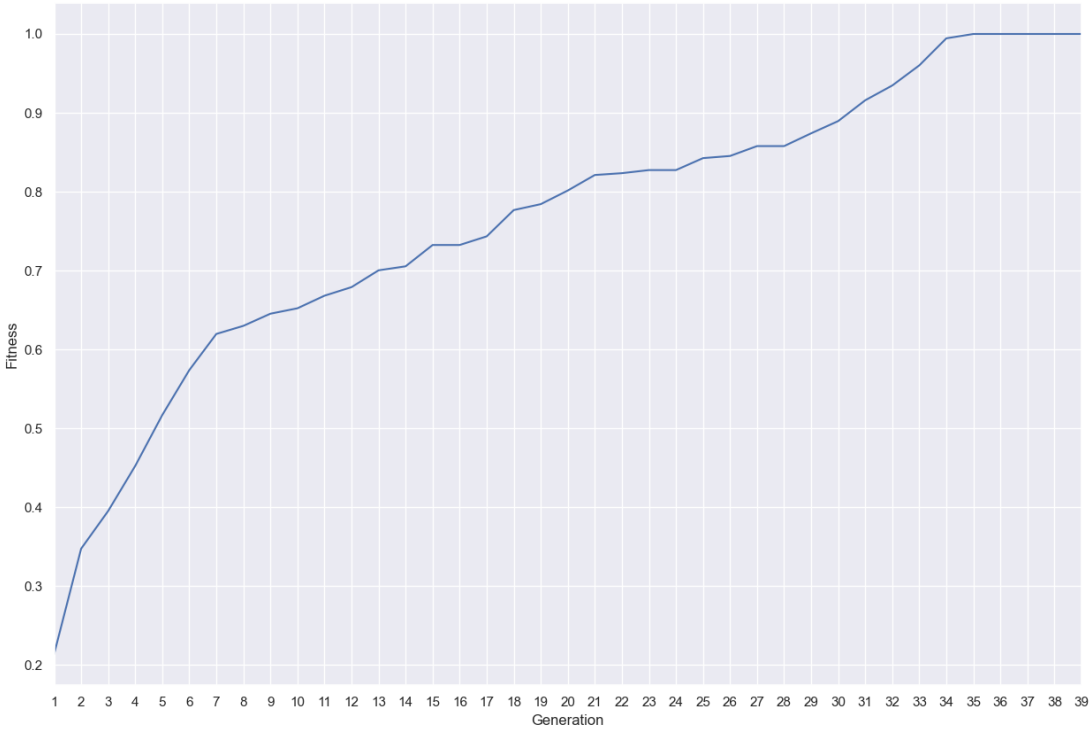


۳-۵-۱۳ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۵، بازترکیب ۰/۵، جمعیت ۱۰۰، تعداد نسل ۲۰۰ و انتخاب ویژگی

در این روش هیچ متدی برای تنظیم تعادل داده‌ها استفاده نشده است. نتایج حاصل از اجرای الگوریتم با تنظیمات ذکر شده به شکل زیر است.

- نمودار همگرایی الگوریتم

مشاهده می‌شود الگوریتم در نسل ۳۵ با میانگین برازندگی ۱ به همگرایی رسیده و پس از ۵ نسل عدم تغییر میانگین برازندگی متوقف می‌شود.



- نتیجه مدل‌سازی زبانی

نتیجه مدل‌سازی زبانی در آدرس زیر قابل مشاهده است.

0.5_0.5_100_200/featureSelection/

linguistic_model_0.5_0.5_100_200_featureSelection.txt

- معیار دقت و معیار f1 برای داده‌های آموزش

accuracy_score: 0.87

f1_score: 0.12

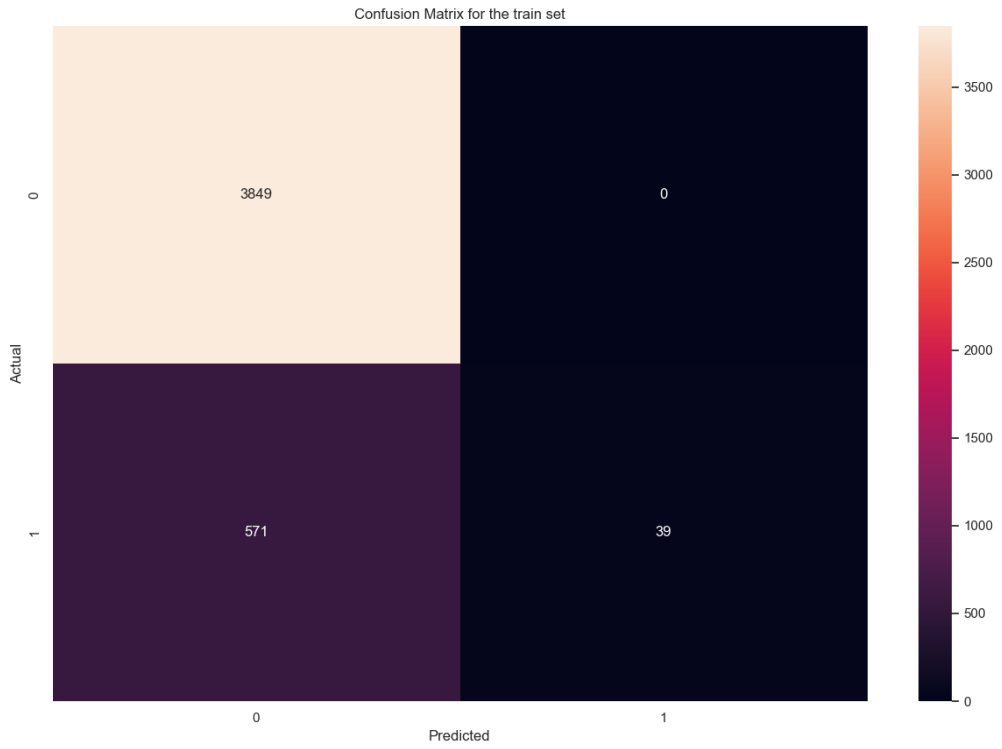
- معیار دقت و معیار f1 برای داده‌های تست

accuracy_score: 0.88

f1_score: 0.11

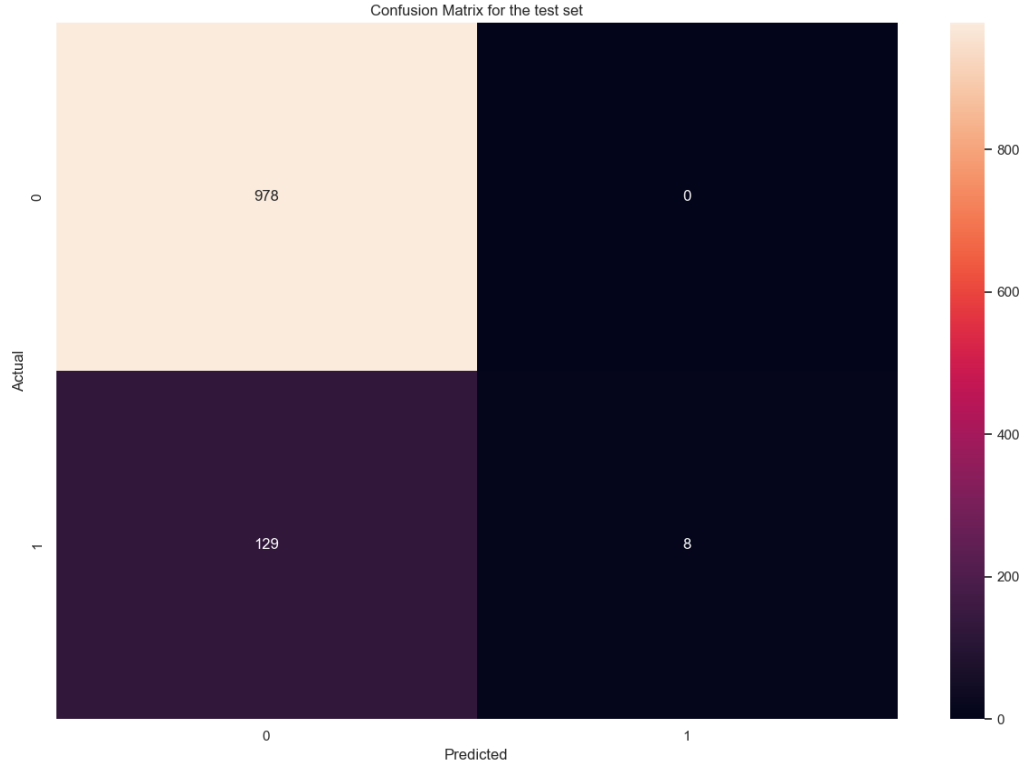
- ماتریس درهم‌ریختگی برای داده‌های آموزش

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است. اما قادر به تشخیص پیام‌های اسپم تا حدی است.



- ماتریس درهم‌ریختگی برای داده‌های تست

مشاهده می‌شود مدل تعداد بسیاری از پیام‌های اسپم را به اشتباه غیر اسپم تشخیص داده است. اما قادر به تشخیص پیام‌های اسپم تا حدی است.



۳-۶- ج) نتیجه نهایی مدل‌سازی زبانی و مجموعه قوانین فازی به دست آمده

پس از آزمایش‌های انجام شده در می‌یابیم مدل با کانفیگ‌های متفاوت و حتی در هر بار تست می‌تواند عملکرد مختلفی داشته باشد. برای بررسی نتیجه نهایی مدل‌سازی زبانی و مجموعه قوانین فازی به دست آمده برای هر تنظیم می‌توان فایلی با مشخصات زیر را در هر پوشه بررسی کرد.

{mutation}_{recombination}_{population_size}_{max_generation}_{prod(optional)}/{data_sampling_method}/linguistic_model{mutation}_{recombination}_{population_size}_{max_generation}.txt

۳-۷- چ) تحلیل دسته بندی یکی از داده‌ها

از آن‌جایی که تعداد قوانین بسیار زیاد بررسی تک‌تک آن‌ها برای یک داده عملا امکان‌ناپذیر می‌شود. بنابراین، به بررسی اجمالی دسته‌بندی داده‌ها می‌پردازیم. برای هر داده، میزان تطابقش را با قوانین موجود در پایگاه داده بررسی می‌کنیم. برای این کار، دو متغیر gC_0 و gC_1 را در نظر می‌گیریم. در صورتی که قانون مورد نظر کلاس ۰ را مشخص می‌کرد، میزان g_R آن داده با قانون را حساب کرده و به gC_0 اضافه می‌کنیم و در صورتی که کلاس ۱ را مشخص می‌کرد مقدار g_R را به متغیر gC_1 اضافه می‌کنیم و این روند را برای تک تک قوانین موجود در پایگاه داده‌مان تکرار می‌کنیم. در نهایت argmax([gC_0, gC_1]) را به عنوان خروجی دسته‌بند قرار می‌دهیم.

۳-۸- ح) تاثیر تعداد قوانین موجود در پایگاه

استفاده از تعداد قوانین مختلف در پایگاه قانون با تنظیمات مختلف هم می‌تواند نتایج مختلفی داشته باشد و به طور قطعی نمی‌توان گفت کدام یک عملکرد بهتری دارد. لازم به ذکر است از آن‌جایی که برای ایجاد پایگاه قوانین از الگوریتم تکاملی استفاده شده است، هر بار اجرای الگوریتم با تنظیمات‌های یکسان می‌تواند نتیجه مختلفی داشته باشد.

در اینجا دو مورد از تنظیماتی که عملکرد بهتری نسبت به بقیه داشتند را با جمعیت‌های مختلف بررسی می‌کنیم.

۳-۸-۱ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰٫۹، بازترکیب ۰٫۹ و تعداد نسل ۲۰۰

نتیجه عملکرد مدل با تنظیمات ذکر شده، بدون هیچ تغییری در تناسب داده‌ها و با استفاده از عملگر استاندارد min با جمعیت‌های ۵۰، ۲۵۰ و ۵۰۰ به ترتیب در آدرس‌های زیر قابل مشاهده است.

0.9_0.9_50_200/featureSelection

0.9_0.9_250_200/featureSelection

0.9_0.9_500_200/featureSelection

با توجه به نتایج دریافت شده در می‌یابیم تعداد قوانین بسیار کم (۵۰) و تعداد قوانین بسیار زیاد (۵۰۰) بر عملکرد الگوریتم تاثیر منفی دارند و بین این سه مقدار تعداد قوانین ۲۵۰ عملکرد بهتری دارد. قوانین باید به گونه‌ای انتخاب شوند که بهترین عملکرد را داشته باشند و تعداد آن‌ها نیز نباید منجر به افزایش زمان استنتاج شود.

۳-۸-۲ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۹ و تعداد نسل ۲۰۰

نتیجه عملکرد مدل با تنظیمات ذکر شده، پیش‌پردازش نمونه‌کاهی و با استفاده از عملگر ضرب جبری با جمعیت‌های ۵۰، ۲۵۰ و ۵۰۰ به ترتیب در آدرس‌های زیر قابل مشاهده است.

0.9_0.9_50_200_prod/undersample_featureSelection

0.9_0.9_250_200_prod/undersample_featureSelection

0.9_0.9_500_200_prod/undersample_featureSelection

با توجه به نتایج دریافت شده در می‌یابیم تعداد قوانین بسیار کم (۵۰) و تعداد قوانین بسیار زیاد (۵۰۰) بر عملکرد الگوریتم تاثیر منفی دارند و بین این سه مقدار تعداد قوانین ۲۵۰ عملکرد بهتری دارد. قوانین باید به گونه‌ای انتخاب شوند که بهترین عملکرد را داشته باشند و تعداد آن‌ها نیز نباید منجر به افزایش زمان استنتاج شود.

۳-۹- خ) تاثیر استفاده از عملگر ضرب جبری به جای عملگر استاندارد min

استفاده از عملگر ضرب جبری به جای عملگر استاندارد min با تنظیمات مختلف هم می‌تواند نتایج مختلفی داشته باشد و به طور قطعی نمی‌توان گفت کدام یک عملکرد بهتری دارد. اما استفاده از عملگر استاندارد min باعث کاهش زمان اجرا در الگوریتم شد. لازم به ذکر است از آن‌جایی که برای ایجاد پایگاه قوانین از الگوریتم تکاملی استفاده شده است، هر بار اجرای الگوریتم با تنظیمات‌های یکسان می‌تواند نتیجه مختلفی داشته باشد.

۳-۹-۱ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۹، جمعیت ۱۰۰، تعداد نسل ۲۰۰

نتیجه عملکرد مدل با تنظیمات ذکر شده و استفاده از نمونه‌کاهی، انتخاب نمونه‌ها به تعداد ۵۰۰ داده از هر کلاس و بدون هیچ تغییری در تناسب داده‌ها به ترتیب در آدرس‌های زیر قابل مشاهده است.

0.9_0.9_100_200_prod/undersample_featureSelection

0.9_0.9_100_200_prod/sample_featureSelection

0.9_0.9_100_200_prod/featureSelection

۳-۹-۲ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۱، جمعیت ۱۰۰، تعداد نسل ۲۰۰

نتیجه عملکرد مدل با تنظیمات ذکر شده و استفاده از نمونه‌کاهی، انتخاب نمونه‌ها به تعداد ۵۰۰ داده از هر کلاس و بدون هیچ تغییری در تناسب داده‌ها به ترتیب در آدرس‌های زیر قابل مشاهده است.

0.9_0.1_100_200_prod/undersample_featureSelection

0.9_0.1_100_200_prod/sample_featureSelection

0.9_0.1_100_200_prod/featureSelection

۳-۹-۳ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۱، بازترکیب ۰/۹، جمعیت ۱۰۰، تعداد نسل ۲۰۰

نتیجه عملکرد مدل با تنظیمات ذکر شده و استفاده از نمونه‌کاهی، انتخاب نمونه‌ها به تعداد ۵۰۰ داده از هر کلاس و بدون هیچ تغییری در تناسب داده‌ها به ترتیب در آدرس‌های زیر قابل مشاهده است.

0.1_0.9_100_200_prod/undersample_featureSelection

0.1_0.9_100_200_prod/sample_featureSelection

0.1_0.9_100_200_prod/featureSelection

۳-۹-۴ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۵، بازترکیب ۰/۵، جمعیت ۱۰۰، تعداد نسل ۲۰۰

نتیجه عملکرد مدل با تنظیمات ذکر شده و استفاده از نمونه‌کاهی، انتخاب نمونه‌ها به تعداد ۵۰۰ داده از هر کلاس و بدون هیچ تغییری در تناسب داده‌ها به ترتیب در آدرس‌های زیر قابل مشاهده است.

0.5_0.5_100_200_prod/undersample_featureSelection

0.5_0.5_100_200_prod/sample_featureSelection

0.5_0.5_100_200_prod/featureSelection

۳-۱۰- د) تاثیر استفاده از روش‌های کاهش بعد مختلف

استفاده از روش کاهش بعد استخراج ویژگی به جای انتخاب ویژگی با تنظیمات مختلف هم می‌تواند نتایج مختلفی داشته باشد و به طور قطعی نمی‌توان گفت کدام یک عملکرد بهتری دارد. لازم به ذکر است از آن‌جایی که برای ایجاد پایگاه قوانین از الگوریتم تکاملی استفاده شده است، هر بار اجرای الگوریتم با تنظیمات‌های یکسان می‌تواند نتیجه مختلفی داشته باشد.

در اینجا دو مورد از تنظیماتی که عملکرد مطلوب‌تری با استفاده از عملگر ضرب جبری و استاندارد min داشته‌اند را بررسی می‌کنیم.

۳-۱۰-۱ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۹، تعداد نسل ۲۰۰ و جمعیت ۱۵۰

نتیجه عملکرد مدل با تنظیمات ذکر شده، بدون هیچ تغییری در تناسب داده‌ها، با استفاده از عملگر استاندارد min و با روش کاهش بعد استخراج ویژگی در آدرس زیر قابل مشاهده است.

0.9_0.9_150_200/featureExtraction

با بررسی خروجی مدل در این روش و روش انتخاب ویژگی مشاهده می‌شود مدل در روش انتخاب ویژگی عملکرد بهتری داشته است که در صورت اجرای دوباره الگوریتم می‌تواند این نتیجه درست نباشد.

۳-۱۰-۲ عملکرد مدل روی داده‌های آموزش و تست با جهش ۰/۹، بازترکیب ۰/۹، تعداد نسل ۲۰۰ و جمعیت ۱۵۰

نتیجه عملکرد مدل با تنظیمات ذکر شده، پیش‌پردازش داده‌ها با نمونه‌کاهی، با استفاده از عملگر ضرب جبری و با روش کاهش بعد استخراج ویژگی در آدرس زیر قابل مشاهده است.

0.9_0.9_150_200_prod/undersample_featureExtraction

با بررسی خروجی مدل در این روش و روش انتخاب ویژگی مشاهده می‌شود مدل در روش انتخاب ویژگی عملکرد بهتری داشته است که در صورت اجرای دوباره الگوریتم می‌تواند این نتیجه درست نباشد.

۵- منابع

Computational Intelligence, A Methodological Introduction 3rd edition: Rudolf Kruse, Sanaz Mostaghim, Christian Borgelt, Christian Braune, Matthias Steinbrecher