

دانشگاه صنعتي امیرکبیر  
(پلی­تکنیک تهران)

دانشكده ریاضی و علوم کامپیوتر

درس هوش مصنوعی و کارگاه

**پروژه ششم**

نگارش

فاطمه گل محمدی

40113426

استاد اول

دکتر مهدی قطعي

استاد دوم

بهنام یوسفي مهر

دی 1403

# چكيده

در این مقاله به بررسی الگوریتم ژنتیک میپردازیم . ما نشان می دهیم که چه اجزایی الگوریتم های ژنتیک را تشکیل می دهند و چگونه آنها را بنویسیم و کاربردهای فعلی و پیشرفت های آینده را مورد بحث قرار می دهیم.

الگوریتم‌های ژنتیک نشان‌دهنده یکی از شاخه‌های مطالعاتی به نام محاسبات تکاملی هستند که از فرآیندهای بیولوژیکی تولید مثل و انتخاب طبیعی برای حل بهینه ترین راه‌حل‌ها تقلید می‌کنند.

در ابتدا، داده‌ها بارگذاری و پیش‌پردازش می‌شوند. این مرحله شامل شناسایی و پر کردن مقادیر گمشده، تبدیل ویژگی‌های غیر عددی به عددی، و استانداردسازی داده‌ها است. سپس، بعد از تعریف و معرفی الگوریتم ژنتیک پیاده سازی انرا انجام میدهیم و مراحل اجرای الگوریتم را توضیح میدیم.

واژه های کلیدی:

الگوریتم ژنتیک ، بهینه ، تولید مثل ، پیاده سازی

|  |  |
| --- | --- |
| فهرست مطالب | صفحه |

[چكيده ‌أ](#_Toc148112500)

[۱. فصل اول مقدمه 3](#_Toc148112501)

[۲. فصل دوم معرفی الگوریتم ژنتیک 7](#_Toc148112504)

[۱-۲- بررسی الگوریتم 8](#_Toc148112505)

[۳. فصل سوم مراحل اجرای الگوریتم 12](#_Toc148112509)

[۱-۳- تعاریف عناصر 13](#_Toc148112510)

[۳-۲- مراحل اجرای الگوریتم 14](#_Toc148112511)

[۴. فصل چهارم کد 18](#_Toc148112513)

[۱-۴- شرح و بررسی پیاده سازی کد 19](#_Toc148112514)

[۵. فصل پنجم جمع‌بندی و نتیجه‌گیری 23](#_Toc148112517)

[منابع و مراجع 25](#_Toc148112518)

# ۱. فصل اول مقدمه

**مقدمه**

الگوریتم‌های ژنتیک یکی از مهم‌ترین روش‌ها برای حل مسائل بهینه‌سازی با استفاده از مفاهیم تکامل و تولید مثل در طبیعت هستند. این الگوریتم با مدل سازی جمعیت‌ها و فرزندان برای یافتن حل موردنظر به کار می‌رود. این متن شامل یک معرفی کلی از الگوریتم ژنتیک و کاربرد آن بر یک موجودیت داده‌ای می‌باشد.

بهینه‌سازی یک فرآیند مهم در مسائل متنوع از صنعت و مدیریت تا علوم داده و مشکلات محاسباتی است. بسیاری از مسائل بهینه‌سازی معمولا غیر خطی و دشوار هستند و روش‌های متعارف قادر به یافتن فضای پاسخ مناسب نیستند. الگوریتم‌های ژنتیک با به کارگیری ایده‌های تکاملی از جمله انتخاب مطلوب و تولیدمثل‌های جدید می‌توانند این مسائل را به صورت مؤثری حل نمایند.

در دنیای امروز، بسیاری از مسائل واقعی شامل فضای جستجوی بزرگ و پیچیده‌ای هستند که روش‌های سنتی در یافتن پاسخ‌های بهینه برای آن‌ها ناکارآمد هستند. الگوریتم ژنتیک با استفاده از جمعیت‌محوری و جستجوی تصادفی کنترل‌شده، قابلیت اجتناب از بهینه‌های محلی را فراهم کرده و به راه‌حل‌های نزدیک بهینه دست پیدا می‌کند. این ویژگی به‌ویژه در کاربردهای پیشرفته‌ای مانند طراحی مهندسی، یادگیری ماشین و مدل‌سازی زیستی بسیار مؤثر است.

# ۲. فصل دوم معرفی الگوریتم ژنتیک

## ۱-۲- بررسی الگوریتم

در حوزه [هوش](https://maktabkhooneh.org/mag/what-is-artificial-intelligence/) مصنوعی ، الگوریتم‌های مختلفی به حل مسائل پیچیده کمک می‌کنند. یکی از این الگوریتم‌ها که توجه زیادی را به خود جلب کرده است، الگوریتم ژنتیک (Genetic Algorithm) است. با ریشه در زیست‌شناسی تکاملی، GA ثابت کرده که ابزاری قدرتمند برای بهینه‌سازی و مشکلات جستجو در هوش مصنوعی است.  این الگوریتم تکاملی بر روی جمعیتی از راه‌حل‌های بالقوه عمل می‌کند که به صورت کروموزوم یا افراد نمایش داده می‌شوند. این کروموزوم‌ها تحت عملیات ژنتیکی مانند جهش (Mutation) و تقاطع (Crossover) قرار می‌گیرند که منجر به تکامل جمعیت در طول نسل‌های متوالی می‌شود.

ایده اساسی پشت این الگوریتم شبیه‌سازی انتقال ویژگی‌های ارثی از طریق ژن‌ها است، دقیقاً شبیه نحوه انتقال صفات انسانی از طریق کروموزوم‌ها. هر ژن در این کروموزوم‌ها نشان دهنده یک ویژگی خاص است. اما در عمل، انتقال کامل کروموزوم ها به نسل بعدی اتفاق نمی‌افتد. دو رویداد اولیه به طور هم‌زمان اتفاق می‌افتد. اولین رویداد به عنوان mutation شناخته می‌شود، که در آن ژن‌های خاصی دستخوش تغییرات تصادفی می‌شوند. اگرچه تعداد ژن‌های جهش‌یافته معمولاً کم است، اما این تغییرات تصادفی نقش مهمی دارند. به عنوان مثال، ژن مسئول رنگ چشم می‌تواند به طور تصادفی منجر به این شود که فردی در نسل بعدی چشمان سبز داشته باشد، درحالی‌که نسل قبلی عمدتاً دارای چشمان قهوه‌ای بود. جهش تنوع و امکان ظهور صفات جدید را معرفی می‌کند. رویداد دوم، که بیشتر از جهش رخ می‌دهد، crossover نامیده می‌شود. در طول تقاطع، دو کروموزوم با یکدیگر ترکیب می‌شوند و قسمت‌هایی را با یکدیگر مبادله می‌کنند. این فرآیند باعث می‌شود که فرزندان ترکیبات متفاوتی از ژن‌ها را در مقایسه با والدین خود به ارث ببرند. تقاطع ترکیب مجدد اطلاعات ژنتیکی را ترویج می‌کند و منجر به فرزندانی با ویژگی‌های منحصربه‌فرد می‌شود.

با ترکیب این مکانیسم‌های الهام گرفته از تکامل طبیعی، الگوریتم‌های ژنتیک می‌توانند به طور مؤثر فضاهای حل پیچیده را جستجو و بهینه کرده و امکان کشف راه‌حل‌های بهینه یا نزدیک به بهینه را برای طیف وسیعی از مسائل فراهم کنند. امروزه الگوریتم ژنتیک در هوش مصنوعی ترکیب شده است و توانایی‌های بسیار حیرت انگیزی دارد.

# ۳. فصل سوم مراحل اجرای الگوریتم ۱-۳- تعاریف عناصر

این الگوریتم شامل عنصر زیر است :

جمعیت: مجموعه ای از راه حل های بالقوه برای یک مشکل است.

کروموزوم ها: نمایشی از یک جواب که اغلب به صورت رشته کدگذاری می شود.

ژن ها: بخش هایی از یک کروموزوم که پارامترها یا متغیرهای خاصی را نشان می دهند.

تابع fitness: تابعی است که بر اساس نزدیکی هر کروموزوم به راه حل بهینه، آن را ارزیابی و مقداری به آن اختصاص می دهد و الگوریتم سعی در بهینه سازی آن دارد.

انتخاب: انتخاب فرآیندی است که تعیین می‌کند کدام راه‌حل‌ها باید حفظ شوند و اجازه تولید مجدد داده شود و کدام یک مستحق نابودی هستند. هدف اصلی عملگر انتخاب تاکید بر راه حل های خوب و حذف راه حل های بد در یک جمعیت است، در حالی که اندازه جمعیت را ثابت نگه می دارد. بهترین ها را انتخاب می کند، بقیه را کنار می گذارد. راه حل های خوب را در یک جمعیت شناسایی می‌کند. چندین کپی از راه حل های خوب تهیه می‌کند. راه حل های بد را از جمعیت حذف می‌کند تا بتوان چندین نسخه از راه حل های خوب را در جمعیت قرار داد.

ⅽrossover: ترکیب فرآیندی است که در آن دو کروموزوم مواد ژنتیکی خود را با هم ترکیب می‌کنند تا فرزندان جدیدی تولید کنند که هر دو ویژگی آنها را دارند.

ⅿutation: معرفی تغییرات تصادفی در یک کروموزوم برای حفظ تنوع در یک جمعیت را mutation میگویند.

با این تعاریف مراحل الگوریتم را توضیح میدهیم.

## ۳-۲- مراحل اجرای الگوریتم

این الگوریتم با تولید جمعیت اولیه‌ای از کروموزوم‌ها (راه‌حل‌ها) آغاز می‌شود. هر کروموزوم شامل یک مجموعه‌ از ژن‌ها است که نمایانگر مقادیر تصمیم در مسئله هستند. جمعیت اولیه معمولاً به‌صورت تصادفی مقداردهی می‌شود. سپس، برای هر کروموزوم، مقدار برازندگی (Fitness) محاسبه می‌گردد که نشان‌دهنده کیفیت آن راه‌حل است.

در هر نسل، کروموزوم‌های با برازندگی بالاتر انتخاب می‌شوند تا برای تولید نسل بعدی استفاده شوند. این فرآیند شامل دو مرحله اصلی است: ترکیب (Crossover) و جهش (Mutation). در ترکیب، دو کروموزوم والد برای تولید دو فرزند ترکیب می‌شوند و اطلاعات ژنتیکی آن‌ها ادغام می‌شود. در جهش، ژن‌های یک کروموزوم به‌صورت تصادفی تغییر می‌کنند تا تنوع جمعیت حفظ شود و امکان کشف راه‌حل‌های جدید فراهم گردد. این دو مرحله به الگوریتم کمک می‌کنند تا فضای جستجو را به‌صورت مؤثر پیمایش کند.

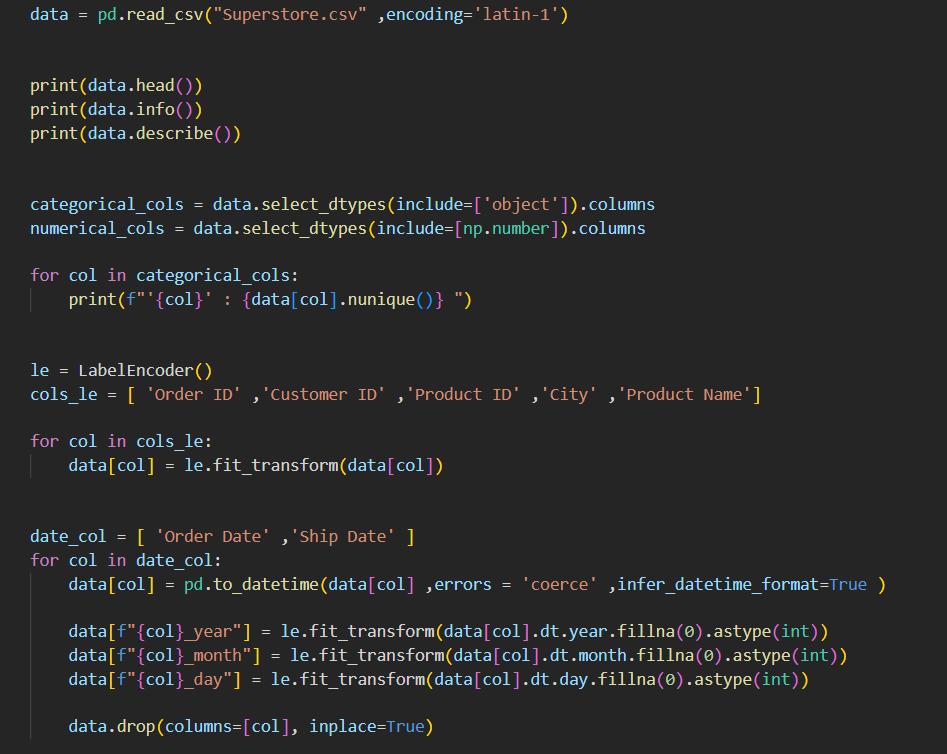
این فرآیند تا رسیدن به معیار توقف ادامه می‌یابد، مثلاً تعداد مشخصی از نسل‌ها یا یافتن کروموزومی با برازندگی مطلوب. در نهایت، بهترین کروموزوم از بین جمعیت انتخاب می‌شود که نمایانگر بهترین راه‌حل بهینه‌سازی است. الگوریتم ژنتیک به دلیل توانایی آن در جستجوی فضای راه‌حل‌های پیچیده و کاربرد گسترده در مسائل مختلف، یکی از روش‌های پرکاربرد در هوش مصنوعی و بهینه‌سازی محسوب می‌شود.

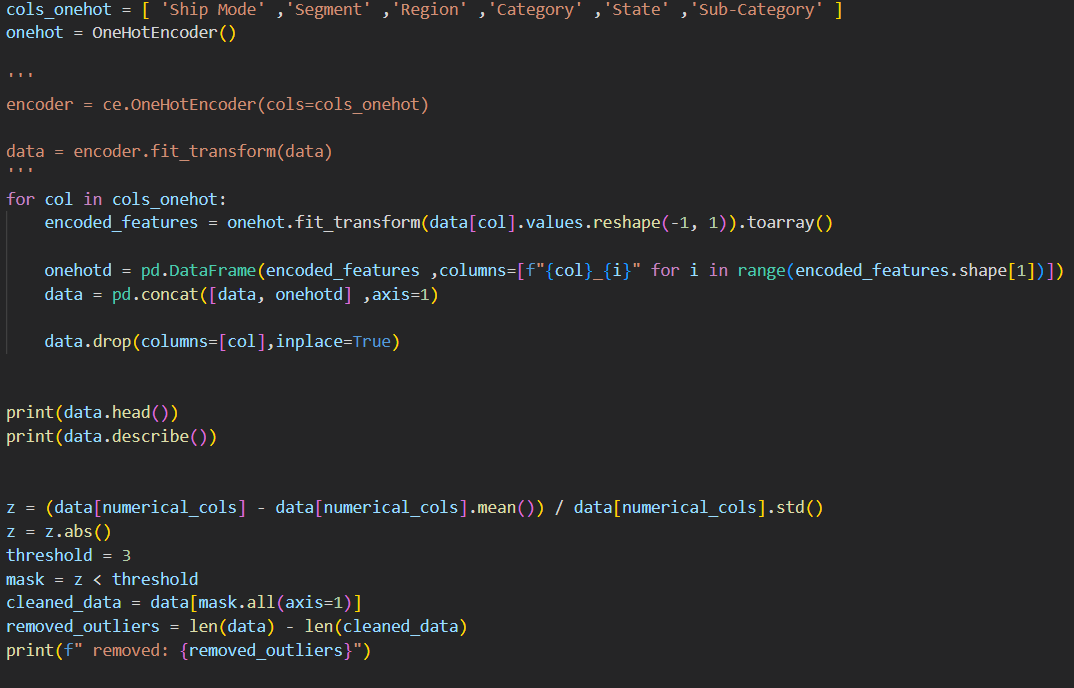
# ۴. فصل چهارم کد

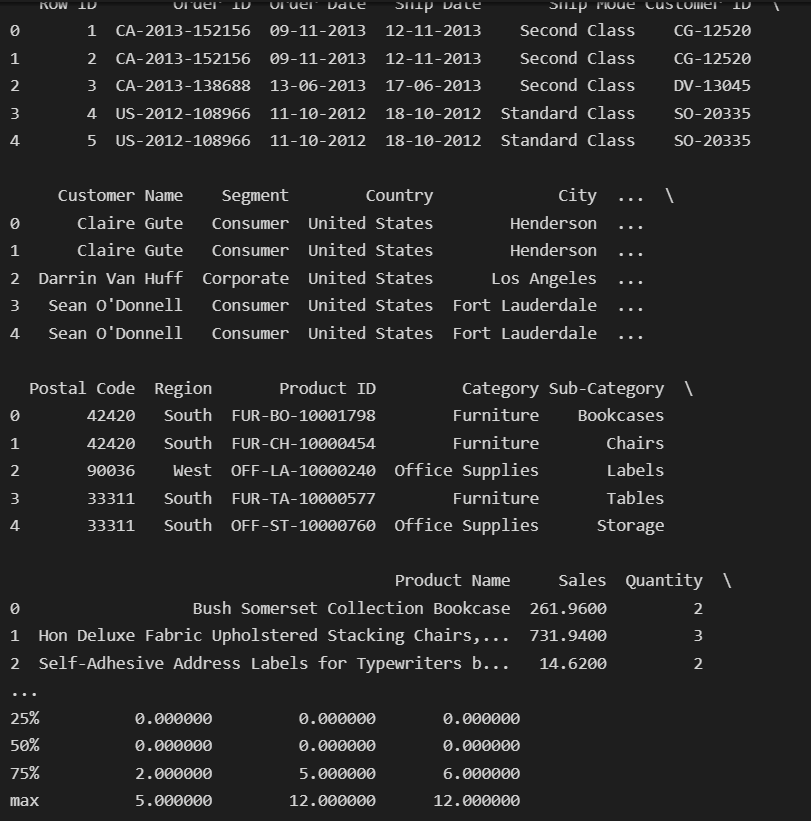
پیش‌پردازش داده‌ها مرحله‌ای کلیدی در تحلیل داده‌ها و یادگیری ماشین است که به آماده‌سازی داده‌ها برای مدل‌سازی و تحلیل کمک می‌کند. این مرحله شامل شناسایی و مدیریت داده‌های گمشده، که ممکن است به دلایل مختلفی ایجاد شوند، و تبدیل داده‌های غیر عددی به عددی است. برای کاهش تأثیر منفی داده‌های گمشده، معمولاً از روش‌هایی مانند پر کردن مقادیر گمشده با میانگین یا میانه استفاده می‌شود. همچنین، ویژگی‌های غیر عددی با استفاده از تکنیک‌هایی مانند One-Hot Encoding و Label Encoding به فرمت عددی تبدیل می‌شوند تا الگوریتم‌های یادگیری ماشین بتوانند به‌طور مؤثری با آن‌ها کار کنند.

در ابتدای این کد، داده‌های مربوط به فروشگاه بارگذاری شده و اطلاعات اولیه مانند تعداد مقادیر خالی، نوع داده‌ها، و توصیف آماری مورد بررسی قرار می‌گیرد. داده‌های دسته ای و عددی (Numerical) شناسایی شده و ستون‌های خاصی با استفاده از LabelEncoder و OneHotEncoder به داده‌های عددی تبدیل می‌شوند.

برای ستون‌های زمانی، مقادیر به تاریخ تبدیل شده و ویژگی‌های جدیدی مثل سال، ماه و روز استخراج می‌شود. علاوه بر این، شناسایی و حذف مقادیر پرت انجام شده تا کیفیت داده‌ها برای تحلیل بهبود یابد.







بخش دوم کد، مربوط به تعریف و اجرای الگوریتم ژنتیک است. ابتدا جمعیت اولیه‌ای از کروموزوم‌ها تولید می‌شود که هر کروموزوم نشان‌دهنده انتخاب یا عدم انتخاب ویژگی‌های خاصی از داده‌هاست. سپس، با محاسبه امتیاز برازندگی (Fitness)، کروموزوم‌ها بر اساس شانس موفقیت انتخاب می‌شوند. عملیات‌های متداول در الگوریتم ژنتیک شامل ترکیب (Crossover) و جهش (Mutation) اعمال شده و نسل جدیدی از کروموزوم‌ها ایجاد می‌شود. این فرآیند برای چندین نسل تکرار می‌گردد تا به یک کروموزوم بهینه دست پیدا کند که بهترین ترکیب ویژگی‌ها را نمایش دهد.

تابع fitness

این تابع به عنوان تابع برازندگی عمل می‌کند و تعیین می‌کند که یک کروموزوم چقدر مناسب است. کروموزوم یک آرایه باینری است که هر مقدار آن نشان‌دهنده انتخاب یا عدم انتخاب یک ویژگی است. تابع ابتدا تمام ستون‌های داده‌ی مرتبط با کروموزوم فعال (مقادیر 1) را انتخاب می‌کند، سپس مجموع مقادیر این ستون‌ها را برای هر ردیف محاسبه کرده و میانگین آن‌ها را برمی‌گرداند. این مقدار به عنوان معیاری از کیفیت آن کروموزوم استفاده می‌شود.

تابع population

این تابع جمعیتی اولیه برای الگوریتم ژنتیک تولید می‌کند. جمعیت یک ماتریس از کروموزوم‌ها است که هر کروموزوم به صورت تصادفی مقداردهی می‌شود. تابع به تعداد کروموزوم‌ و طول هر کروموزوم نیاز دارد. خروجی آن ماتریسی از اعداد صفر و یک است که نشان‌دهنده وضعیت اولیه کروموزوم‌ها برای شروع الگوریتم است.

تابع selection

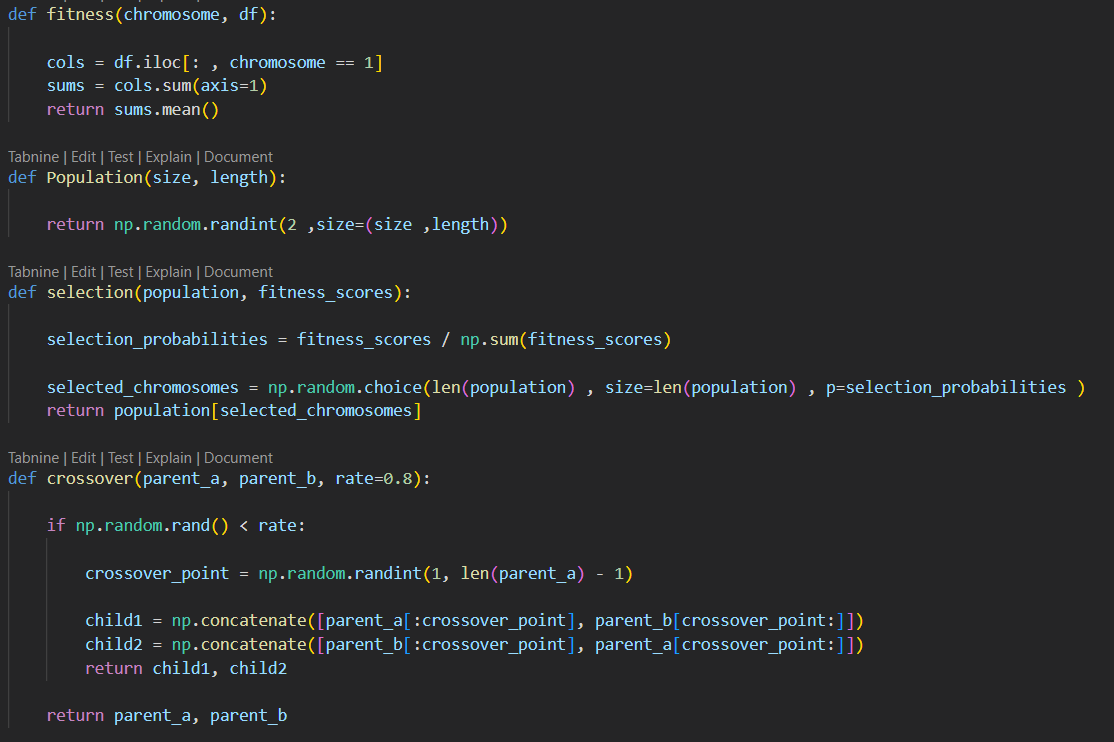
این تابع کروموزوم‌هایی را که احتمال بیشتری برای تولید نسل بعد دارند، انتخاب می‌کند. ابتدا، بر اساس مقدار برازندگی هر کروموزوم، احتمال انتخاب هر یک از آن‌ها محاسبه می‌شود. سپس، با استفاده از این احتمالات، کروموزوم‌هایی به تعداد جمعیت اولیه انتخاب می‌شوند. این فرایند به بقای کروموزوم‌های با برازندگی بالا و انتقال ویژگی‌های آن‌ها به نسل‌های بعد کمک می‌کند.

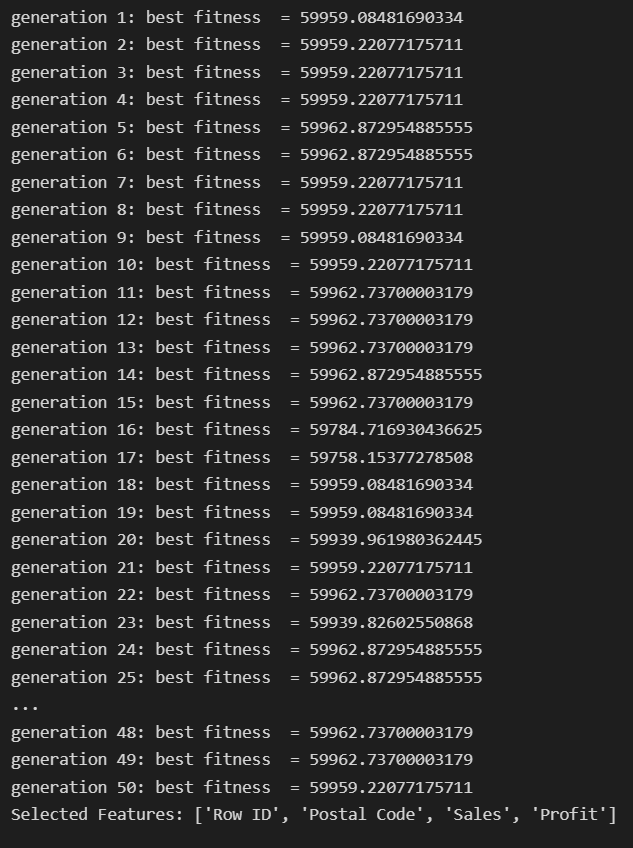
تابع crossover

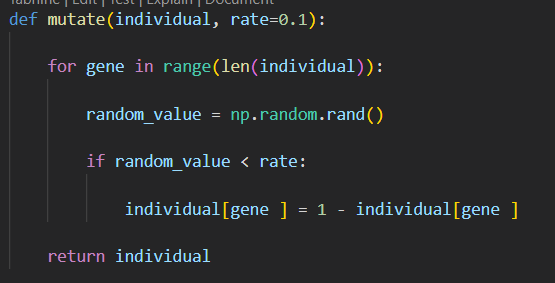
این تابع ترکیب دو کروموزوم والد را انجام می‌دهد تا دو کروموزوم جدید تولید کند. اگر مقدار تولیدشده کمتر از نرخ تلاقی (rate) باشد، یک نقطه تصادفی برای ترکیب انتخاب می‌شود. سپس بخش‌های قبل و بعد از این نقطه از والدین با یکدیگر تعویض می‌شوند. در غیر این صورت، والدین بدون تغییر به عنوان فرزندان بازمی‌گردند. این عمل باعث انتقال ژن‌های مفید به نسل‌های بعدی می‌شود.

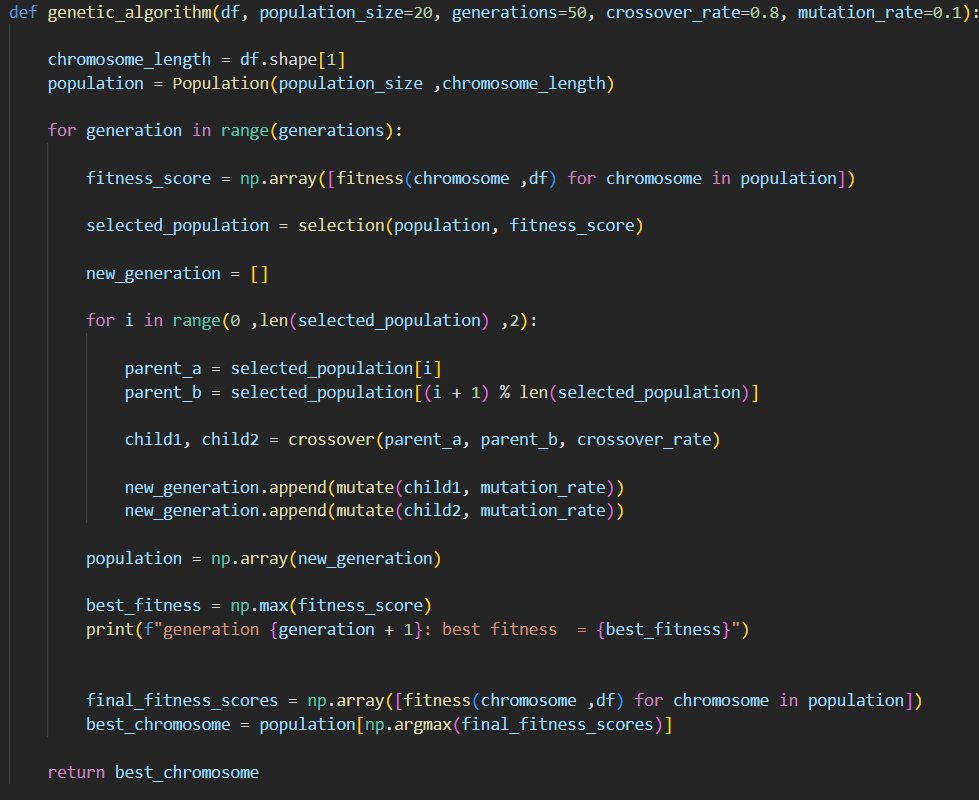
تابع mutate

این تابع مسئول ایجاد جهش در کروموزوم‌ها است. در هر موقعیت از کروموزوم، احتمال وقوع جهش بررسی می‌شود. اگر مقدار تصادفی تولیدشده کمتر از نرخ باشد، مقدار ژن از 1 به 0 یا از 0 به 1 تغییر می‌کند. این فرایند به حفظ تنوع در جمعیت و جلوگیری از گیر افتادن در بهینه‌های محلی کمک می‌کند.



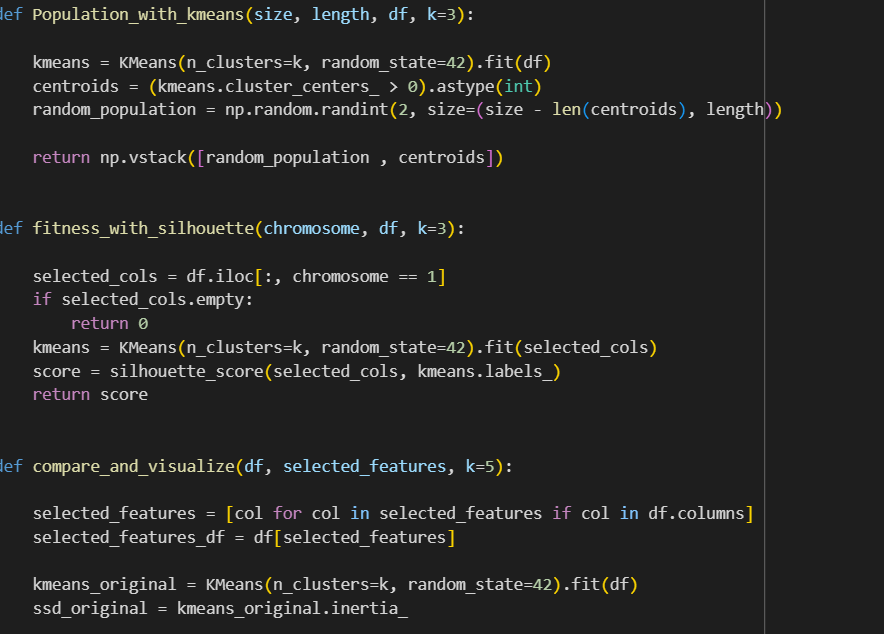


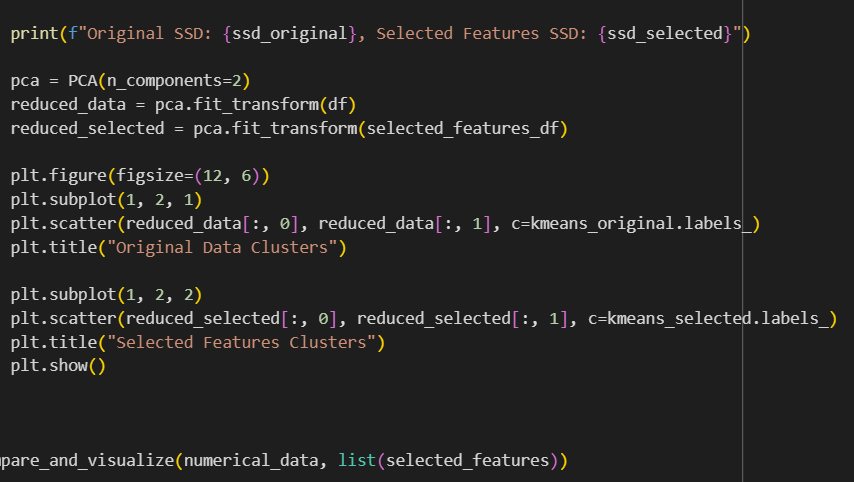


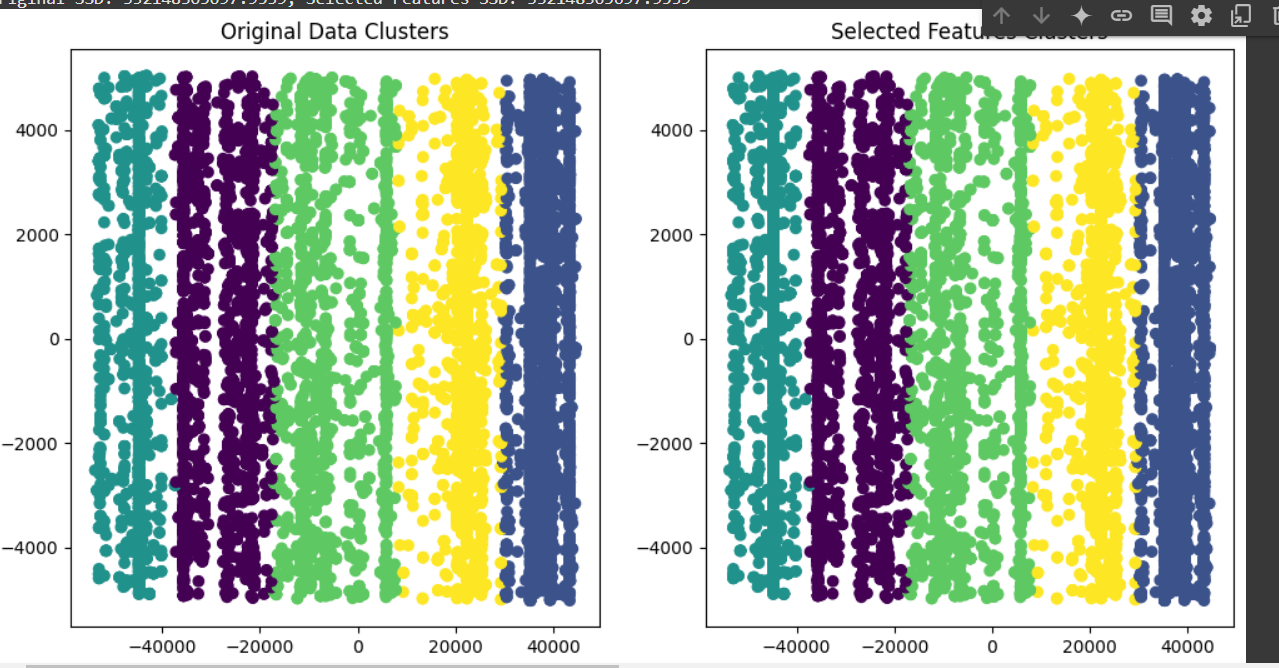


در این بخش از کد با استفاده از الگوریتم K-Means، اول خوشه‌بندی روی داده‌های اصلی را انجام می‌دهیم تا مراکز خوشه‌ها را پیدا کننیم و از آن‌ها برای ایجاد یک جمعیت اولیه شامل ویژگی‌های انتخابی استفاده کنیم. سپس کیفیت خوشه‌بندی داده‌های انتخاب‌شده توسط یک کروموزوم را ارزیابی می‌کنیم

در نهایت نتایج خوشه‌بندی برای داده‌های اصلی و ویژگی‌های انتخاب‌شده را در قالب نمودار به صورت بصری نمایش می‌دهیم تا تفاوت عملکرد آن‌ها مقایسه شود. این فرآیند به انتخاب بهترین زیرمجموعه از ویژگی‌ها کمک می‌کند.







# ۵. فصل پنجم جمع‌بندی و نتیجه‌گیری

**جمع‌بندی و نتیجه‌گیری**

در این مقاله، الگوریتم ژنتیک به‌عنوان یک روش کارآمد و الهام‌گرفته از طبیعت برای حل مسائل بهینه‌سازی معرفی شد. این الگوریتم با بهره‌گیری از مفاهیم تکاملی مانند انتخاب، ترکیب و جهش، امکان جستجوی مؤثر در فضای راه‌حل‌های پیچیده را فراهم می‌کند. در ادامه، با ارائه کد پیاده‌سازی الگوریتم ژنتیک، روند اجرای آن برای مسئله انتخاب ویژگی‌ها شرح داده شد. این کد مراحل مختلفی از قبیل مقداردهی اولیه، محاسبه برازندگی، انتخاب کروموزوم‌های برتر، و تولید نسل‌های جدید را شامل می‌شود

در نهایت، الگوریتم ژنتیک به‌عنوان ابزاری قدرتمند برای حل مسائل بهینه‌سازی در حوزه‌های مختلف معرفی شد. این الگوریتم می‌تواند به‌طور گسترده در کاربردهایی همچون یادگیری ماشین، تحلیل داده‌ها، و بهینه‌سازی سیستم‌ها مورد استفاده قرار گیرد.

**منابع و مراجع**

["faradars"](file:///C:\Users\Asus\AppData\Roaming\Microsoft\Word\AI_HW1.docx)

["geeksforgeeks"](file:///C:\Users\Asus\AppData\Roaming\Microsoft\Word\AI_HW1.docx)

["researchgate"](file:///C:\Users\Asus\AppData\Roaming\Microsoft\Word\AI_HW1.docx)

["logistic"](file:///C:\Users\Asus\AppData\Roaming\Microsoft\Word\AI_HW1.docx)

["gardner"](file:///C:\Users\Asus\AppData\Roaming\Microsoft\Word\AI_HW1.docx)

["wikipedia"](file:///C:\Users\Asus\AppData\Roaming\Microsoft\Word\AI_HW1.docx)

<https://maktabkhooneh.org/mag/genetic-algorithm-in-artificial-intelligence/>

<https://www.konkurcomputer.ir/%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%85-%DA%98%D9%86%D8%AA%DB%8C%DA%A9.html>

<https://blog.faradars.org/genetic-algorithm/>

<https://blog.faradars.org/%D9%BE%DB%8C%D8%A7%D8%AF%D9%87-%D8%B3%D8%A7%D8%B2%DB%8C-%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%85-%DA%98%D9%86%D8%AA%DB%8C%DA%A9-%D8%AF%D8%B1-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86/>

<https://www.datacamp.com/tutorial/genetic-algorithm-python?utm_source=google&utm_medium=paid_search&utm_campaignid=19589720821&utm_adgroupid=157098104615&utm_device=c&utm_keyword=&utm_matchtype=&utm_network=g&utm_adpostion=&utm_creative=726015683436&utm_targetid=dsa-2264919292029&utm_loc_interest_ms=&utm_loc_physical_ms=1006094&utm_content=&utm_campaign=230119_1-sea~dsa~tofu_2-b2c_3-row-p1_4-prc_5-na_6-na_7-le_8-pdsh-go_9-nb-e_10-na_11-na-jan25&gad_source=1&gclid=CjwKCAiAp4O8BhAkEiwAqv2UqLkCLKcCeOdHeCK-RS5afqmg3IwuITNQTO41VWZT4_QCUy6ZBszpVRoC-8EQAvD_BwE>