

دانشکده مهندسی

گروه مهندسی کامپیوتر

گرایش هوش مصنوعی

الگویتم استراتژی تکاملی

استاد:

دکتر مجتبی روحانی

دانشجو:

علی گلی

آبان ماه

سال 1402

**1 الگویتم ES**

الگوریتم استراتژی های تکاملی با توابع بنچمارک؛ با یک جمعیت یک نفره انجام شده است. نکته قابل توجه میزان step یا sigma می باشد که در کد به صورت یک عدد کوچکتر از یک بیان شده است.

ابتدا یک جمعیت تک نفره به صورت تصادفی از روی یک رشته از اعداد (بازه اعداد قابل تعیین است) ساخته می‌شود. سپس جهش روی همین فرد انجام می‌شود. در صورتی که جهش باعث بهتر شدن ژن مربوطه بشود جهش نگاه داشته می شود. در غیر اینصورت همان نسل قبلی باقی می‌ماند.

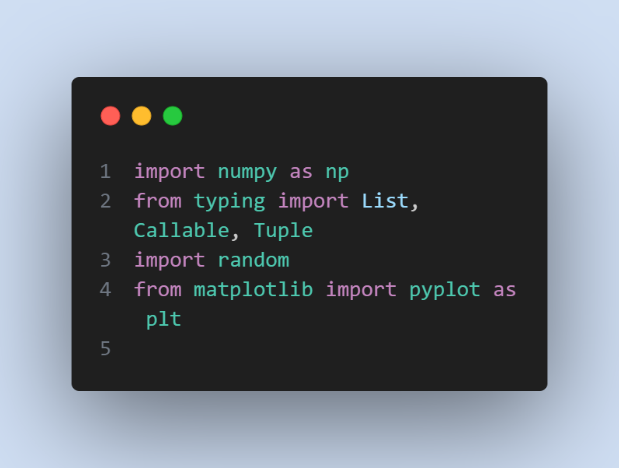
عملیات جهش به صورت اتفاقی یکی از اعداد بازه اعداد ژن تعریف شده را انتخاب می‌کند.

عملیات evaluation که برتری ژن را پیدا می‌کند؛ ابتدا باید یک تابع بنچمارک به عنوان ورودی بگیرد و سپس مقدار برتری ژن را محاسبه می‌کند. در صورت خارج شدن عدد از مقدار مشخص شده تابع بنچمارک؛ نرمال می‌‌شود.

**2 کد الگوریتم ES**

**1.2 کتابخانه های مورد نیاز**

ابتدا کتابخانه های مورد نیاز را به پروژه اضافه می‌کنیم:



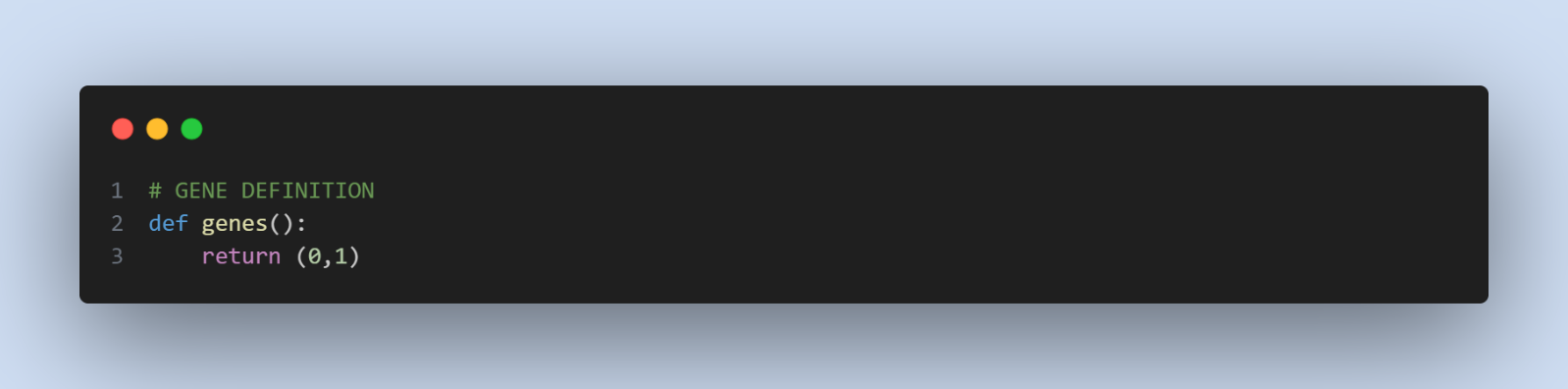
**2.2 توابع بنچمارک**

حال توابع بنچمارک که احتیاج داریم تعریف می‌کنیم. این توابع ژن را دریافت و یک اسکالر باز می‌گردانند.



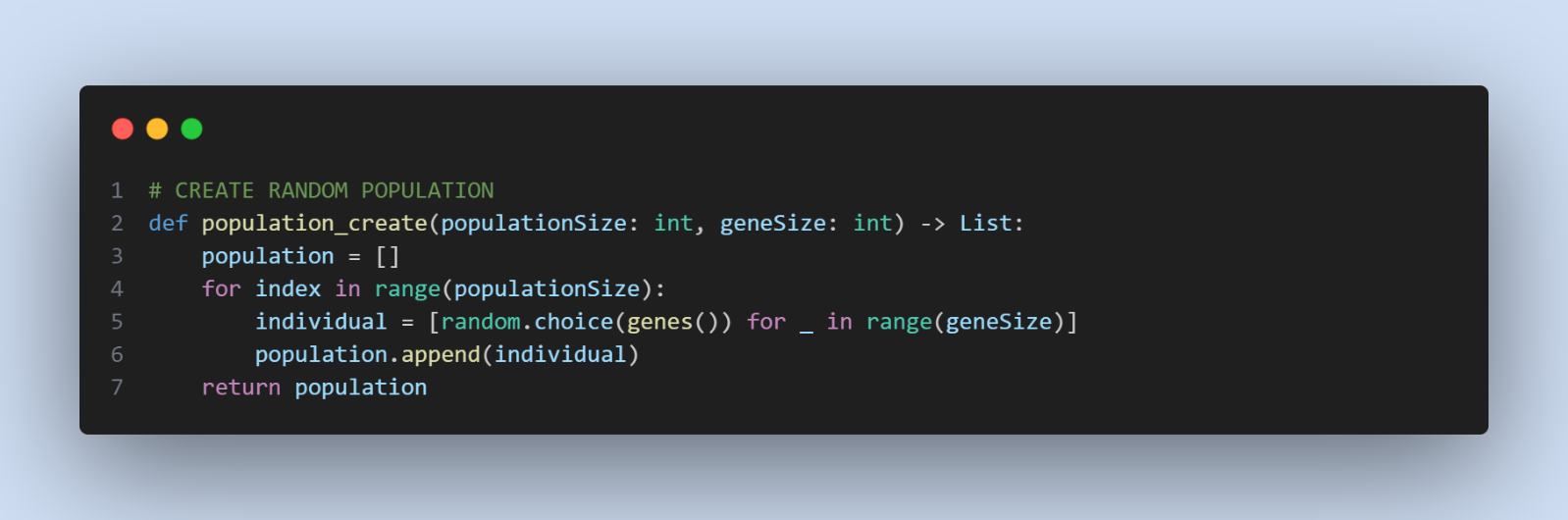
**3.2 تابع ژن**

حال تابع ژن را باید تعریف کنیم و اعداد مجازی که درون ژن می‌توانند قرار بگیرند را مشخص کنیم.



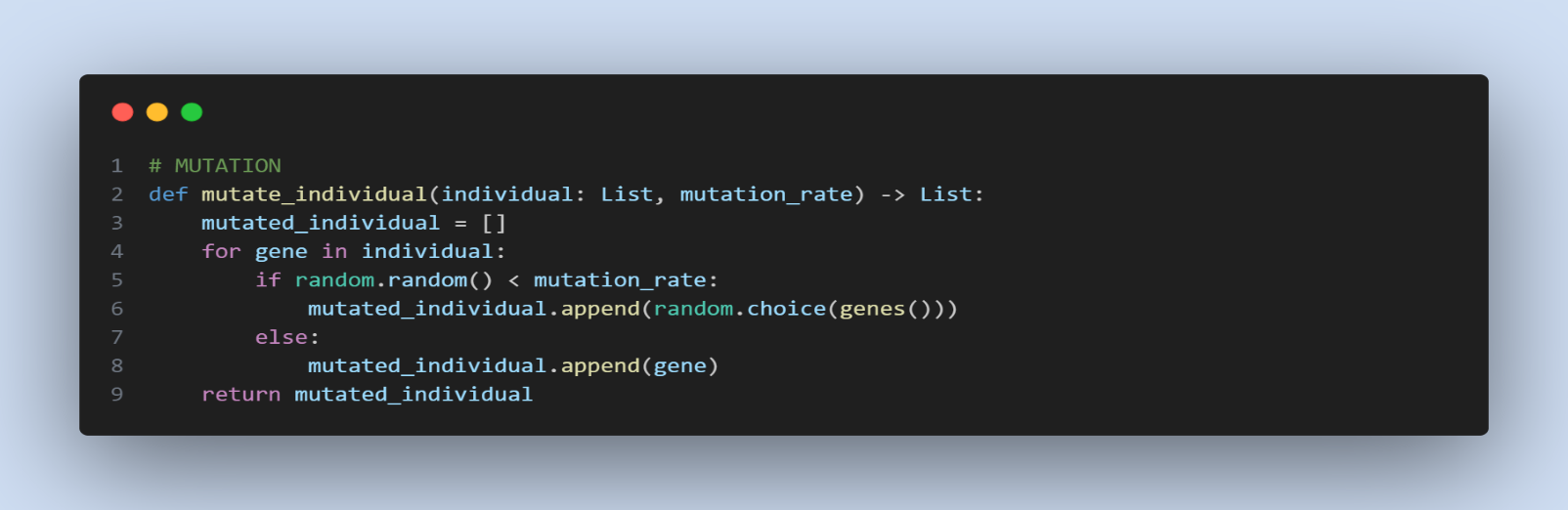
**4.2 ساخت جمعیت تک نفره**

یک جمعیت تک نفره که به صورت اتفاقی ژنی با سایز geneSize دارد می‌سازیم.



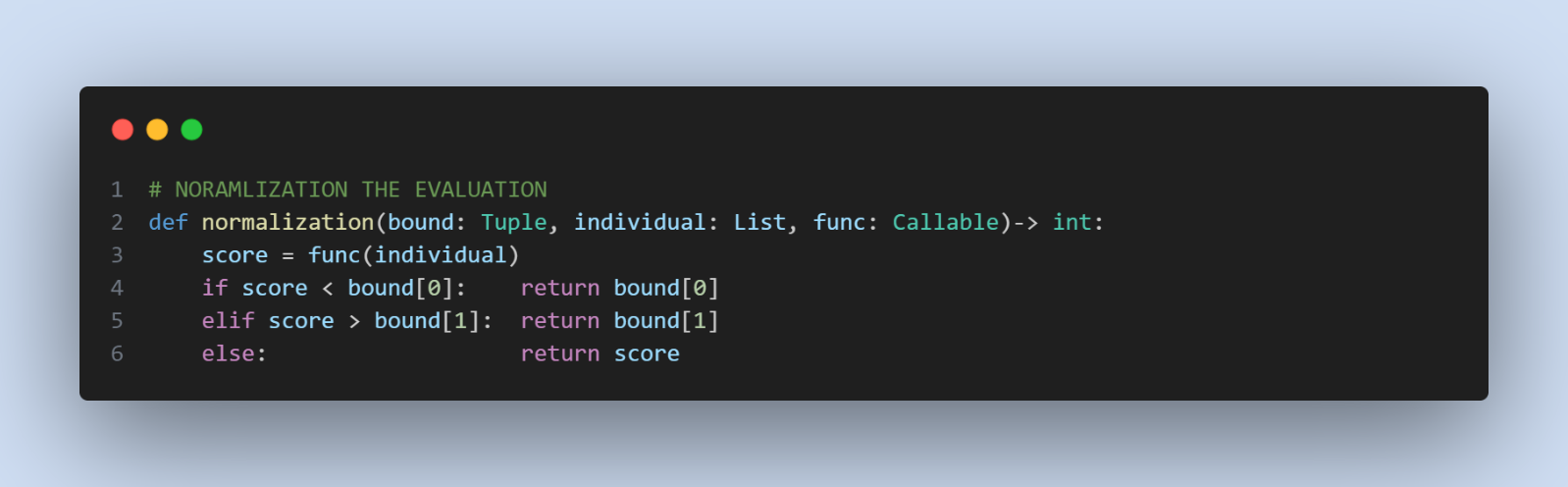
**5.2 تابع جهش**

تابع جهشی باید نوشته شود که بر اساس سیگما جهش دهد. این تابع ژن یک فرد را دریافت و نوکلئوتید های ژن را؛ اگر سیگما مجاز بداند؛ تغییر خواهد داد.



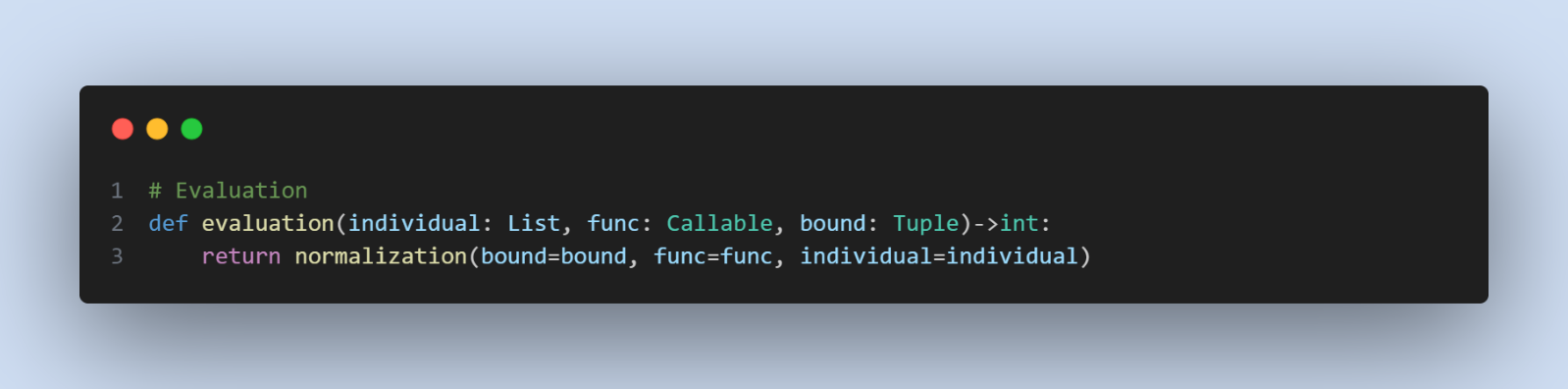
**6.2 تابع نرمال**

این تابع اگر میزان امتیاز ژن بیش از حد شود آن را به مقدار ماکسیمم؛ و اگر کمتر از حد شود آن را به مقدار مینیمم، تخمین می‌زند.



**7.2 تابع ارزیابی**

این تابع هر فردی از جامعه را دریافت و برای آن یک امتیاز برمی‌گرداند.



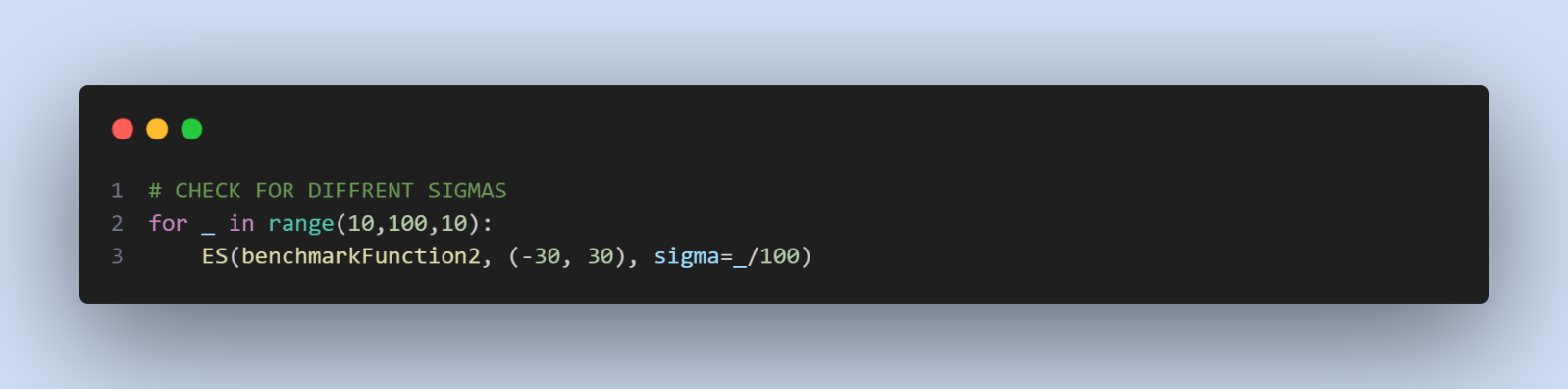
**8.2 توابع ES**

این توابع الگوریتم اصلی ES را اجرا می‌کنند. تابع اول این امکان را به شما می‌دهد که بتوانید به میزان geneSize های متفاوت کد خود را ارزیابی کنید. همچنین در نهایت نتایج را به شما نشان می‌‌دهد.



**9.2 مقدار سیگما**

قبل از اجرای الگوریتم اصلی می‌توانید یک حلقه بنویسید تا مقادیر مختلف سیگما را نیز به شما نشان دهد.



**10.2 کد نهایی**

در نهایت کد نهایی به شکل زیر خواهد بود.



**3 نتایج**

در نهایت می‌توانیم نتایج زیر را مشاهده کنیم:

