

روند کلی الگوریتم ژنتیک

- ۱- ایجاد جمعیت اولیه
- ۲- چک کردن شرط برآورده شدن یا رسیدن به انتهای حداکثر تعداد نسل
- ۲- ارزیابی و امتیازدهی به تک تک اعضای جمعیت
- ۳- انتخاب بهترین عضوهای جمعیت (۱۰ نفر)
- ۴- شافل کردن اعضای باقی مانده برای بالا بردن شانس تولید فرزندهای بهتر
- ۵- عملیات cross-over و تولید فرزند از یک جفت از جمعیت
- ۶- عملیات mutation و جهش برای فرزند تولید شده
- ۷- اضافه کردن فرزندان به جمعیت قبلی
- ۸- برو به ۲

[View Code on Colab](#) مسئله ماشین

انتخاب جمعیت اولیه:

به تعداد pop_number (۴۰) آرایه‌ی car_number عضوی (تعداد ماشین‌هایی که می‌توانیم بخریم) از ۰ و ۱ داریم که ۰ بیانگر عدم خرید ماشین نظیر به ایندکس آن و ۱ بیانگر خرید آن است.

عملیات cross-over: (احتمال ۹۵ درصد)

دو کروموزوم انتخاب شده به صورت رندوم از یک جا شکسته و به صورت ضربدری با هم ترکیب می‌شوند.

عملیات mutation: (احتمال ۲۰ درصد)

برخی از ژنوم‌های یک کروموزوم toggle می‌شوند. (تبدیل یک به صفر و بالعکس)

تابع ارزیابی:

جمع value ماشین‌های خریداری شده - در صورتی که قیمت ماشین‌های خریداری شده بیشتر از حداکثر پول باشد امتیاز صفر داده می‌شود. ← امتیاز بیشتر، کروموزوم بهتر

شرط پایان:

تعداد iteration مشخص شده (از ۵۰ تا ۱۰۰۰ به عنوان مثال)

[View Code on Colab](#) مسئله جمعیت

کلیه داده‌ها (محور x و محور y) نرمالایز می‌شوند.

انتخاب جمعیت اولیه:

به تعداد pop_number (۱۰۰) آرایه‌ی degree عضوی (درجه چند جمله) از اعداد حقیقی بین ۰ و ۱ (نشان گر ضرایب چندجمله‌ای) داریم.

عملیات cross-over: (احتمال ۹۵ درصد)

به جای استفاده از روش‌های معمول شکست و ترکیب از mutation بهتر و مناسب‌تری برای real-value ها استفاده می‌کنیم

$$\text{child1} = \text{parent1} \pm \text{abs}(\text{parent2} - \text{parent1}) * \alpha$$

$$\text{child2} = \text{parent2} \pm \text{abs}(\text{parent2} - \text{parent1}) * \alpha$$

که α عددی رندوم بین ۰ و ۱ است.

این عملیات روی تک تک ژنوم‌های نظیر به نظیر دو کروموزوم انجام می‌شود.

همچنین باید دقت شود که هیچ کدام از ژنوم‌های جدید نباید از بازه‌ی -۱ و ۱ خارج شوند.

عملیات mutation: (احتمال ۲۰ درصد)

به جای استفاده از روش‌های معمول شکست و ترکیب از mutation بهتر و مناسب‌تری برای real-value ها استفاده می‌کنیم و به تک تک ژنوم‌های یک کروموزوم یک مقدار رندوم ثابت را اضافه می‌کنیم.

$$\text{child1} = \text{child1} \pm \alpha$$

همچنین باید دقت شود که هیچ کدام از ژنوم‌های جدید نباید از بازه‌ی -۱ و ۱ خارج شوند.

تابع ارزیابی:

چندجمله‌ای را با توجه به ضرایب به دست آمده تشکیل داده و محور x (سال) را به عنوان ورودی به آن می‌دهیم. مقادیر جدید به عنوان پیش‌بینی بدست می‌آید که توسط تابع ارزیابی زیر امتیازدهی می‌شوند:

انتخاب بهترین افراد یک جمعیت:

$$\text{Score} = \frac{1}{m} \sum_{i=1}^m (y_{act} - y_{real})^2$$

که m برابر طول تعداد اعضای محور y است. \leftarrow امتیاز (loss) کمتر، کروموزوم بهتر

شرط پایان:

iteration مشخص شده (از ۱۰۰ تا ۱۰۰۰ به عنوان مثال)

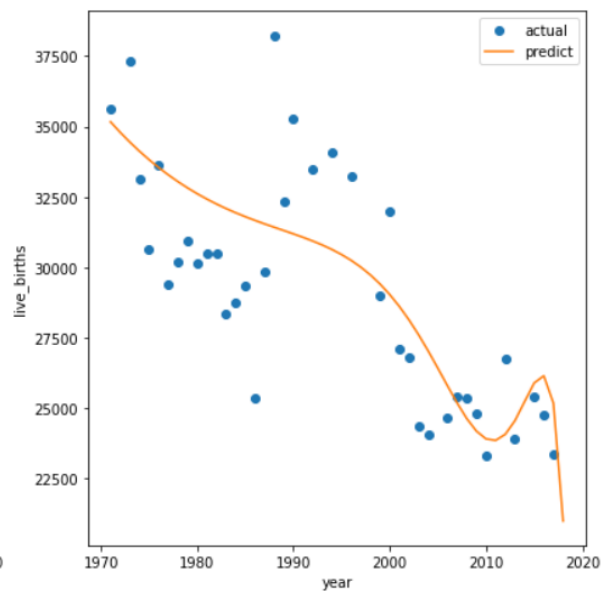
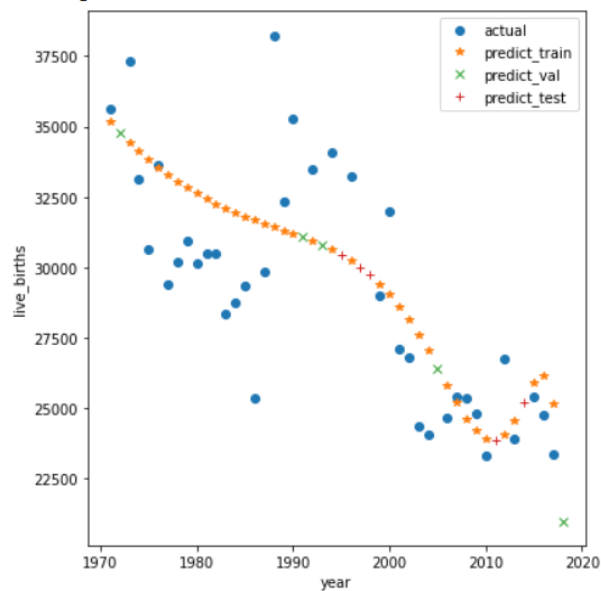
بهترین جواب (درجه):

داده به سه قسمت train, test, validation تقسیم شده است.

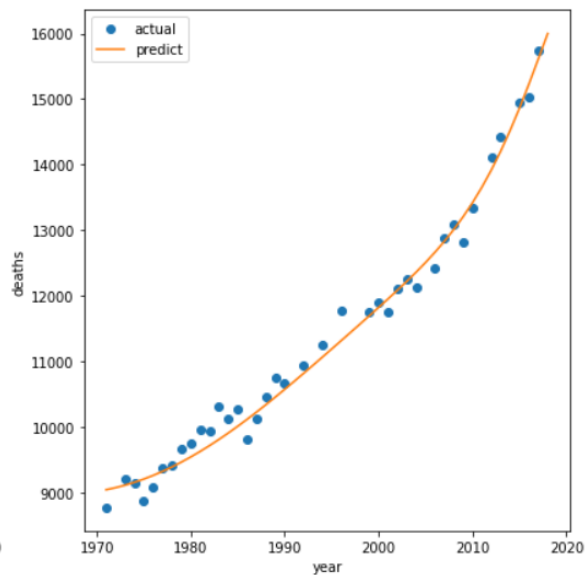
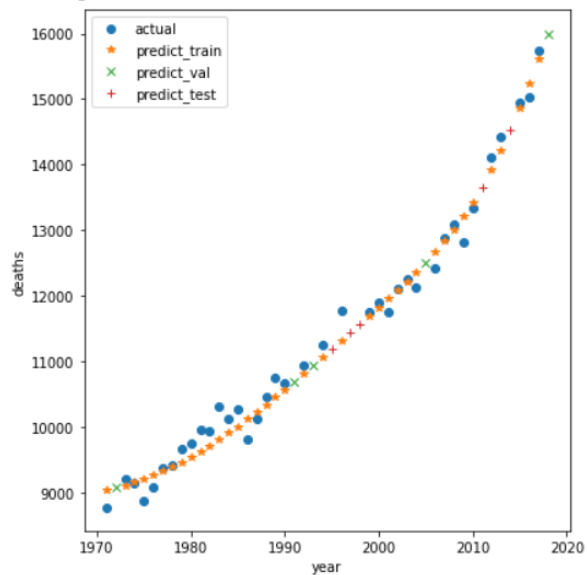
این الگوریتم روی درجه‌های مختلف توسط train آموزش می‌بیند سپس توسط val ارزیابی می‌شوند و همواره بهترین جواب (کمترین loss) و درجه مربوط به آن نگهداری می‌شود و در نهایت به کاربر نمایش داده می‌شود.

نتایج:

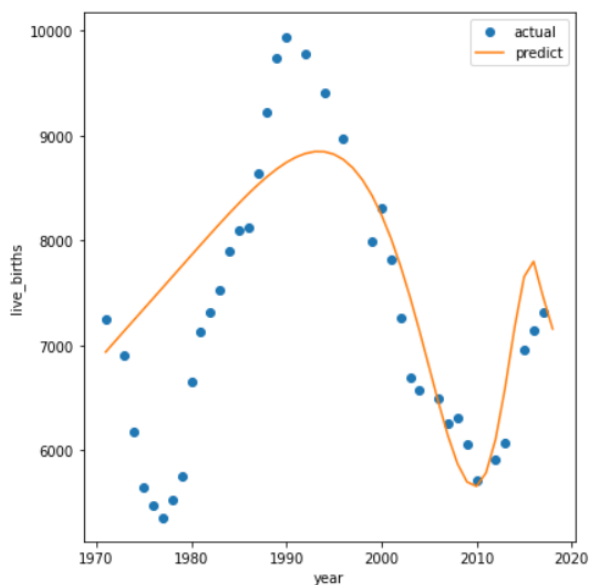
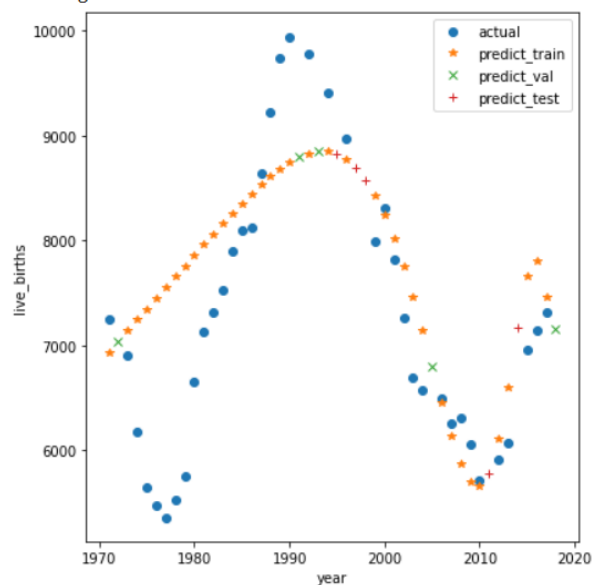
Chinese's live-births:
Best degree: 50



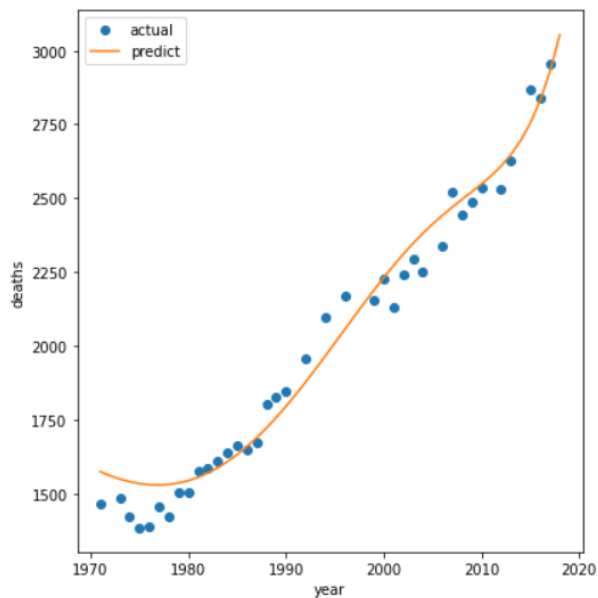
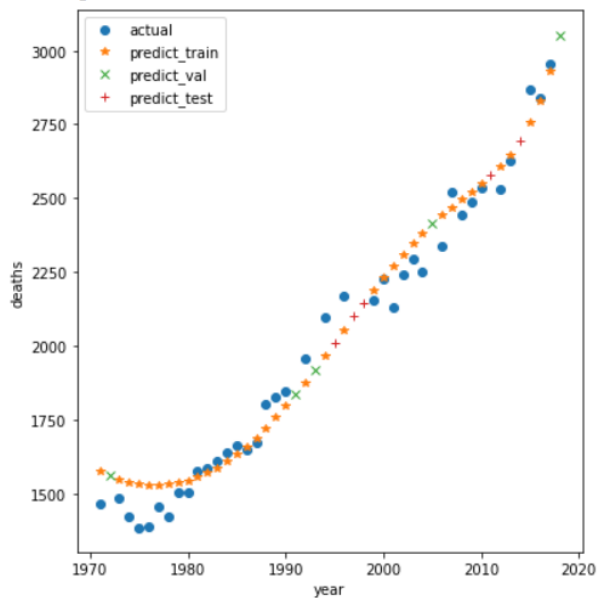
Chinese's deaths:
Best degree: 15



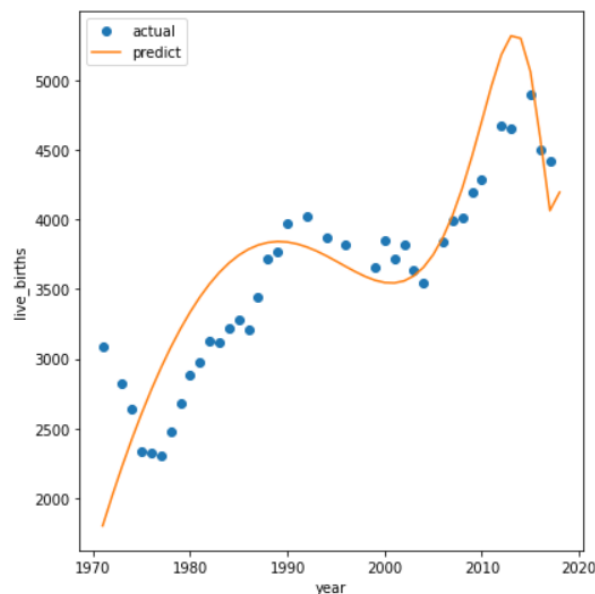
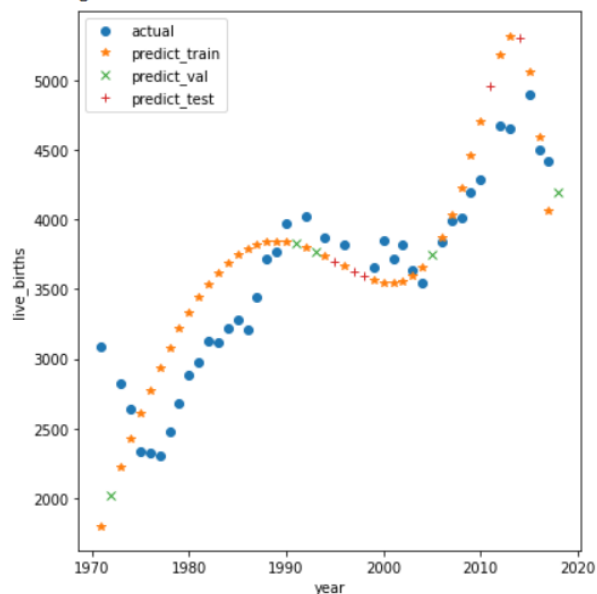
Malays' live-births:
Best degree: 40



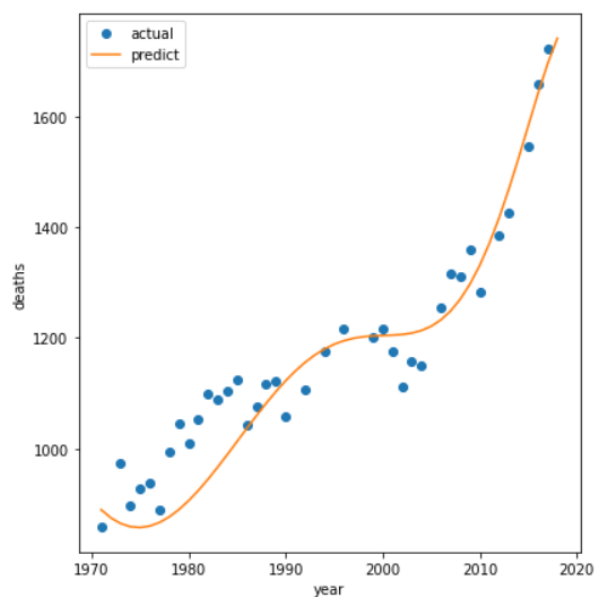
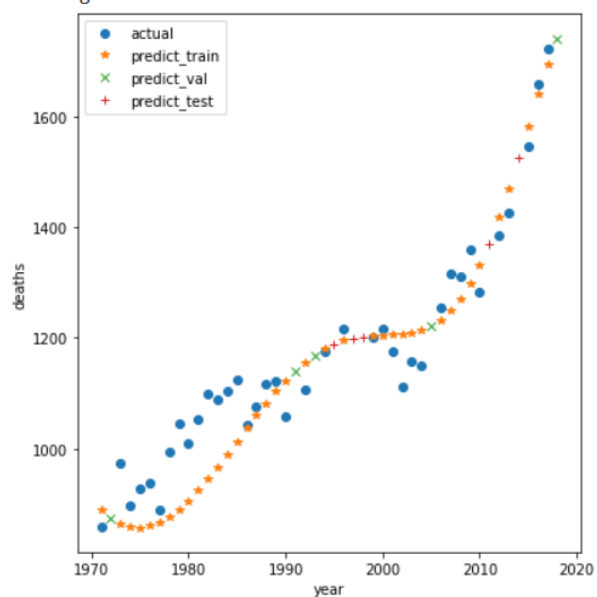
Malays' deaths:
Best degree: 9



Indians' live-births:
Best degree: 37



Indians' deaths:
Best degree: 10



[View Code on Colab](#) مسئله شطرنج

وزیر به طوری در صفحه شطرنج $n \times n$ قرار گیرد بطوریکه دو به دو همدیگر را تهدید نکنند.

انتخاب جمعیت اولیه:

به تعداد pop_number (۴۰) آرایه‌ی $queen_number$ (تعداد وزیرها یا سایز صفحه شطرنج) از ۱ تا

queen_number داریم که ایندکس هر کدام از آن‌ها برابر شماره سطر و خود مقدار آن برابر شماره ستون می‌باشد.

عملیات cross-over:

مثل مسئله اول

عملیات mutation:

مثل مسئله اول با این تفاوت که دو بار عددی رندوم بین ۱ تا queen_number جایگزین مقدار قبلی می‌شود.

تابع ارزیابی:

$$\text{conflicts} = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} g(i, j),$$

$$g(i, j) = \begin{cases} 1, & \begin{cases} \text{if } NQ_i == NQ_j \\ \text{or } |NQ_j - NQ_i| == j - i, \end{cases} \\ 0, & \text{otherwise.} \end{cases}$$

امتیاز هر کروموزوم برابر تعداد conflict هاست. ← امتیاز کمتر، کروموزوم بهتر
روند محاسبه conflict ها به این صورت است که برای هر ردیف وضعیت تهدید مهره در آن با همه‌ی مهره‌های ردیف بعدی بررسی می‌شود.

تهدید عمودی: مطمئن هستیم که نداریم چون در هر ستون با ستون‌های بعد از خود مقایسه می‌شود.
تهدید افقی: به دلیل جهش ممکن است حالتی پیش آید که در یک ردیف ممکن است دو مهره قرار بگیرند.
(بخش اول if)

تهدید ضربدری: اگر اختلاف سطرهای و ستون‌های دو مهره با هم برابر شوند به صورت ضربدری همدیگر را تهدید می‌کنند. (بخش دوم if)

شرط پایان:

امتیاز کروموزوم برابر ۰ شود یا تعداد iteration به حدنصاب برسد

نتیجه: (برای ۱۲ وزیر)

