

## Population Rate

In this problem, we have to predict a polynomial produced by genetic algorithm which predicts **Birth** and **Deaths** of **China**, **India** and **Malaysia**.

### Initial population

For this, there is a function named *initial\_population()* in the code which has 2 arguments which are the **degree** of the polynomial and the number of **Coefficients**. It produces random coefficients between **0** and **1**.

### Crossover

For this part, Because the representation of the problem is **real-valued**, if we suppose that the chromosomes are  $\langle x_1, x_2, \dots, x_n \rangle$ , it is divided into 2 parts. The first half is father and the other half of another chromosome is mother and by this, new child is generated.

### Mutation

I perform **mutation** on **20%** of the chromosomes and it is by adding or subtracting a random number between **0** and **1** to the genes of the selected chromosome.

### Fitness function

In order to work with small numbers instead of huge ones, I normalized all datas as in this problem the error values may be so large. The fitness function is the sum of all errors for one polynomial. For example, if we have **40** years, the error value for one polynomial is the sum of the errors of the specified polynomial exerted on each year. The error function is the absolute difference of the **real value** and the **computed value**.

### Survival Selection

For this part, I save each **polynomial** and its **error** in a list and sort it ascending, which means that by preceding the list, the suitability decreases. So I remove the last **20%** of the elements and the others will go for the next crossover.

## Plots

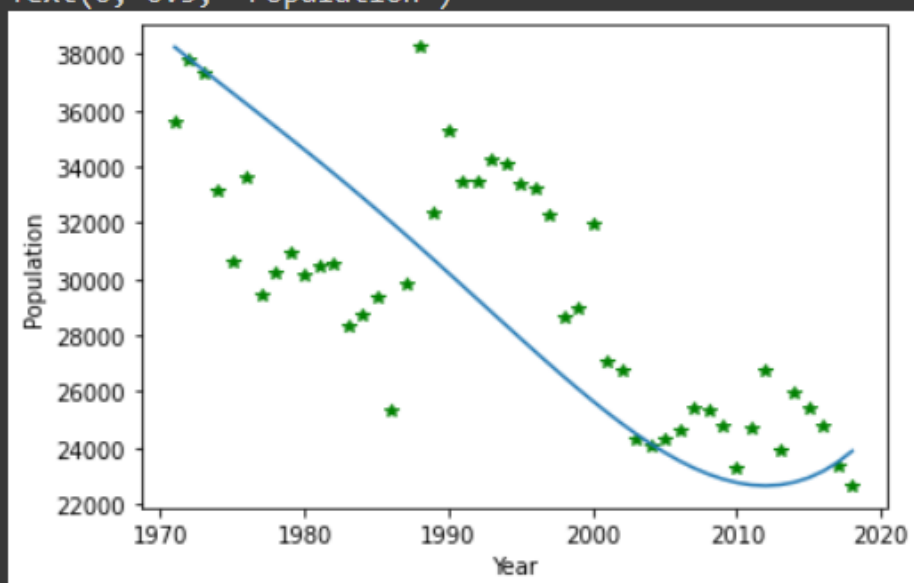
### Chinese Birth

degree 4

## Chinese Birth Plot

```
[18] 1 plt.plot(ch['Year'], chinese_df['live_births'], '*', color='g')  
    2 plt.plot(ch['Year'], chinese_birth_results)  
    3 plt.xlabel('Year')  
    4 plt.ylabel('Population')
```

```
Text(0, 0.5, 'Population')
```



degree 11

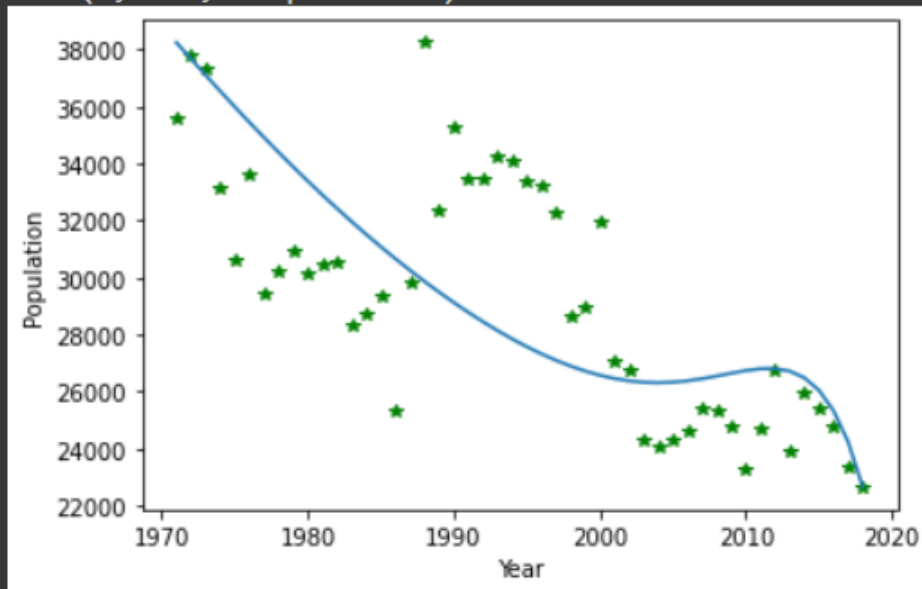
## Chinese Birth Plot



```
1 plt.plot(ch['Year'], chinese_df['live_births'], '*', color='g')
2 plt.plot(ch['Year'], chinese_birth_results)
3 plt.xlabel('Year')
4 plt.ylabel('Population')
```



```
Text(0, 0.5, 'Population')
```

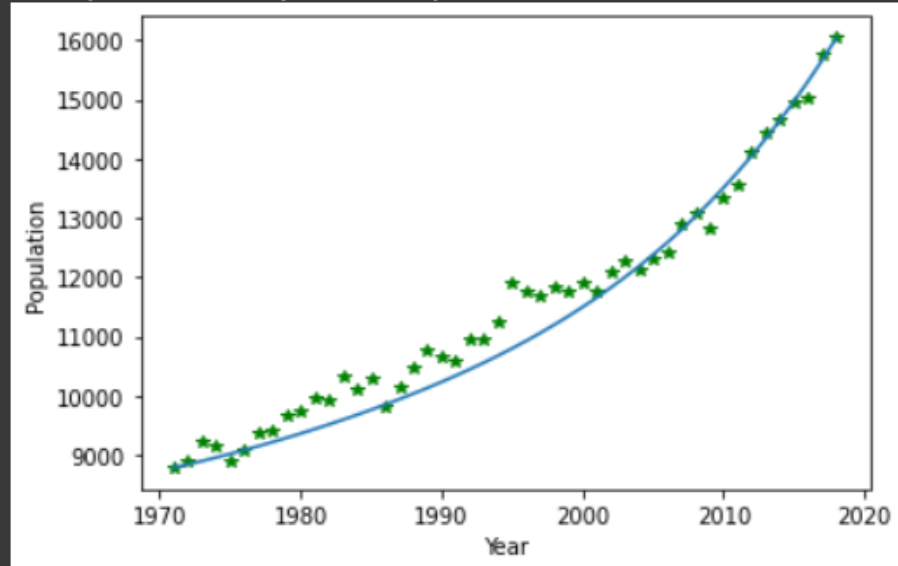


Chinese Deaths



```
1 plt.plot(ch['Year'], chinese_df['deaths'], '*', color='g')
2 plt.plot(ch['Year'], chinese_death_results)
3 plt.xlabel('Year')
4 plt.ylabel('Population')
```

Text(0, 0.5, 'Population')



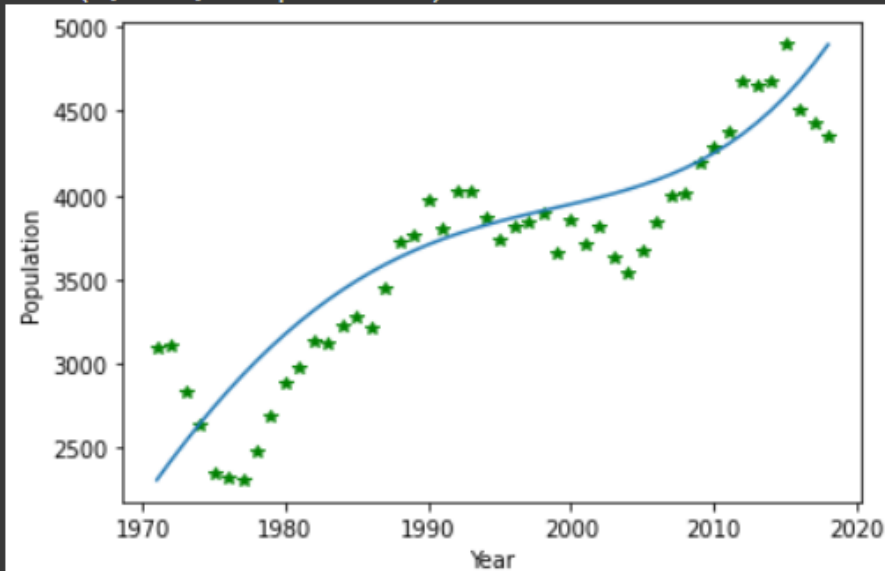
## Indians Birth

### Indians Birth Plot



```
1 plt.plot(indians['Year'], indians_df['live_births'], '*', color='g')
2 plt.plot(indians['Year'], indians_birth_results)
3 plt.xlabel('Year')
4 plt.ylabel('Population')
```

Text(0, 0.5, 'Population')



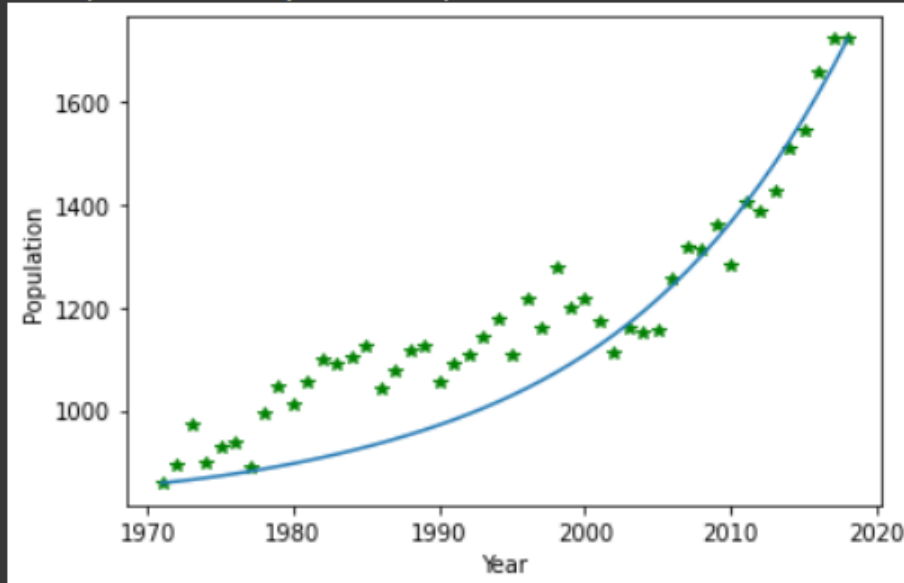
## Indians Death

## Indians Death Plot



```
1 plt.plot(indians_df['year'], indians_df['deaths'], '*', color='g')  
2 plt.plot(indians_df['year'], indians_death_results)  
3 plt.xlabel('Year')  
4 plt.ylabel('Population')
```

```
Text(0, 0.5, 'Population')
```



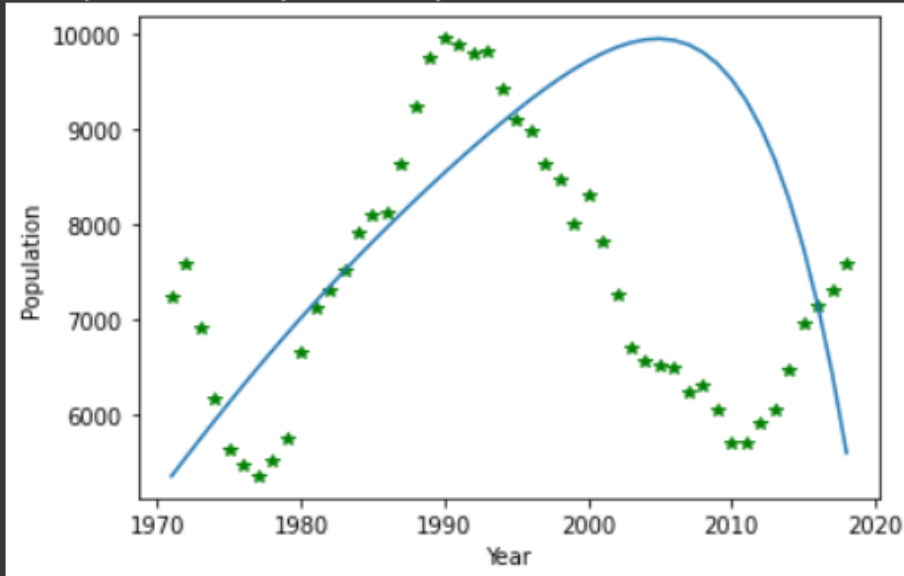
## Malays Birth

degree 4

## Malays Birth Plot

```
1 plt.plot(malays_df['year'], malays_df['live_births'], '*', color='g')  
2 plt.plot(malays_df['year'], malays_birth_results)  
3 plt.xlabel('Year')  
4 plt.ylabel('Population')
```

```
Text(0, 0.5, 'Population')
```



degree 7

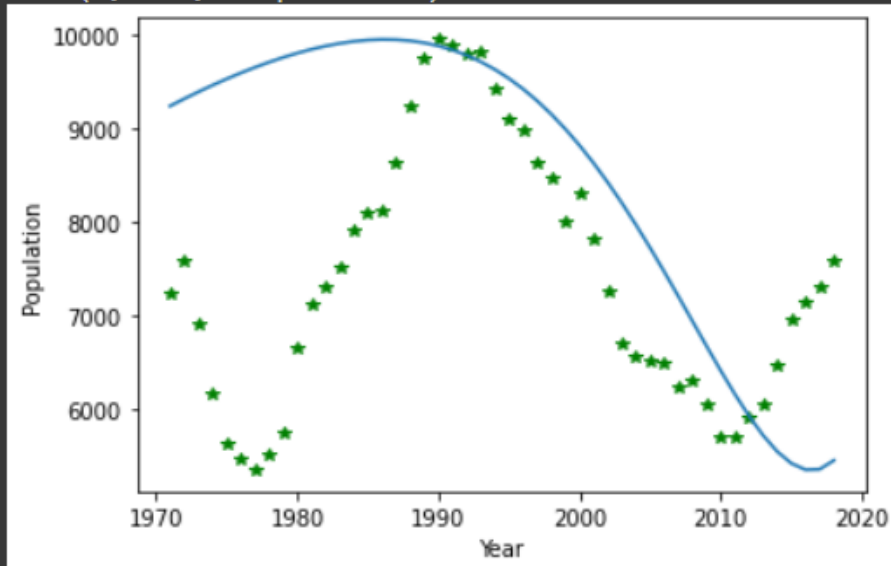
## Malays Birth Plot



```
1 plt.plot(malays_df['year'], malays_df['live_births'], '*', color='g')
2 plt.plot(malays_df['year'], malays_birth_results)
3 plt.xlabel('Year')
4 plt.ylabel('Population')
```



```
Text(0, 0.5, 'Population')
```

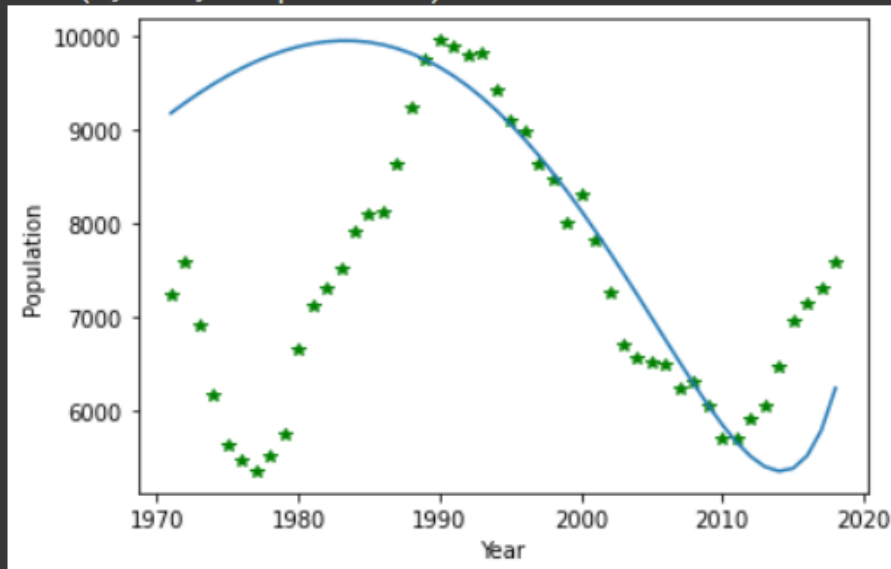


degree 13

## Malays Birth Plot

```
1 plt.plot(malays_df['year'], malays_df['live_births'], '*', color='g')  
2 plt.plot(malays_df['year'], malays_birth_results)  
3 plt.xlabel('Year')  
4 plt.ylabel('Population')
```

```
Text(0, 0.5, 'Population')
```



## Malays Death



## Malays Death Plot

```
1 plt.plot(malays_df['year'], malays_df['deaths'], '*',color='g')  
2 plt.plot(malays_df['year'], malays_death_results)  
3 plt.xlabel('Year')  
4 plt.ylabel('Population')
```

```
Text(0, 0.5, 'Population')
```

