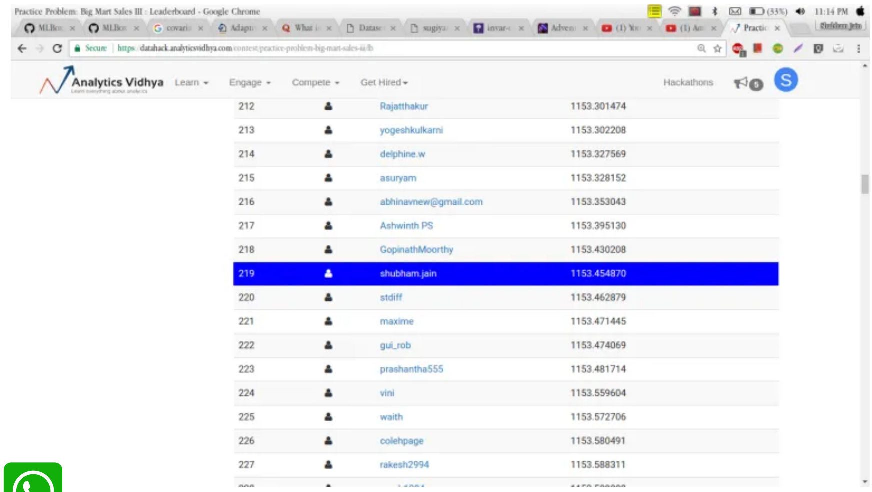
✔ if You Like the posts You can Follow Me here
Vipul Patel
LinkedIn https://www.linkedin.com/in/vipulppatel/

Introduction to Genetic Algorithm & their application in data science

SHUBHAM JAIN,

Introduction

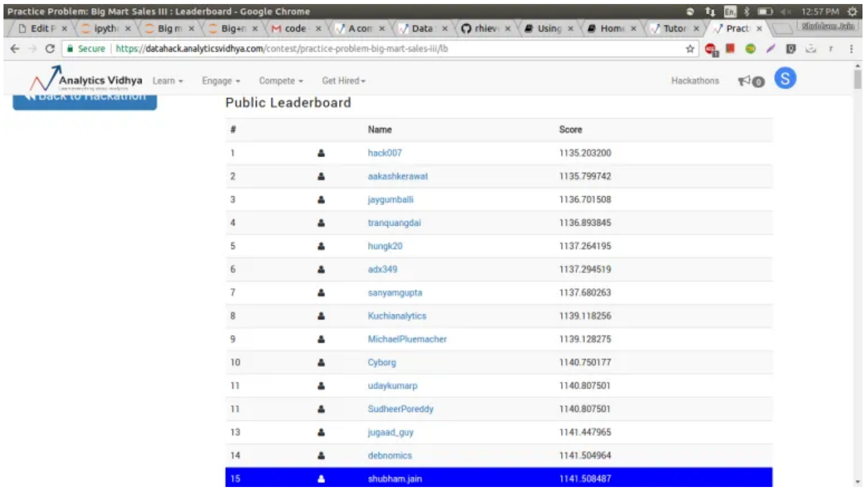
Few days back, I started working on a practice problem – Big Mart Sales. After applying some simple models and doing some feature engineering, I landed up on 219th position on the leader board.


Practice Problem: Big Mart Sales III : Leaderboard - Google Chrome
Secure | https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/lb
Analytics Vidhya
Learn
Engage
Compete
Get Hired
Hackathons
S
11:14 PM

212	Rajatthakur	1153.301474
213	yogeshkulkarni	1153.302208
214	delphine.w	1153.327569
215	asuryam	1153.328152
216	abhinavnew@gmail.com	1153.353043
217	Ashwinth PS	1153.395130
218	GopinathMoorthy	1153.430208
219	shubham.jain	1153.454870
220	stdiff	1153.462879
221	maxime	1153.471445
222	gui_rob	1153.474069
223	prashantha555	1153.481714
224	vini	1153.559604
225	waith	1153.572706
226	colehpage	1153.580491
227	rakesh2994	1153.588311

Not bad – but I needed something better.

So, I started searching for optimization techniques which could improve my score. It was during this search that I was introduced to genetic algorithms. After applying Genetric algorithm to the practice problem, I ended up taking a considerable leap on the leaderboard.



Practice Problem: Big Mart Sales III : Leaderboard - Google Chrome
https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/lb
Analytics Vidhya
Learn
Engage
Compete
Get Hired
Hackathons

Public Leaderboard

#		Name	Score
1	♟	hack007	1135.203200
2	♟	aakashkerawat	1135.799742
3	♟	jaygumballi	1136.701508
4	♟	tranquangdai	1136.893845
5	♟	hungk20	1137.264195
6	♟	adx349	1137.294519
7	♟	sanyamgupta	1137.680263
8	♟	Kuchianalytics	1139.118256
9	♟	MichaelPluemacher	1139.128275
10	♟	Cyborg	1140.750177
11	♟	udaykumarp	1140.807501
11	♟	SudheerPoreddy	1140.807501
13	♟	jugaad_guy	1141.447965
14	♟	debnomics	1141.504964
15	♟	shubham.jain	1141.508487

Yes, a jump from 219th to 15th position just on the basis on genetic algorithm. Isn't that great? By end of this article, you will be comfortable applying genetic algorithms and can expect similar benefit on the problems you are working on.

Table of Content

1. Intuition behind Genetic Algorithms
2. Biological Inspiration
3. What is Genetic Algorithm?
4. Steps Involved in Genetic Algorithm
    1. Initialisation
    2. Fitness Function
    3. Selection
    4. Crossover
    5. Mutation
5. Application of Genetic Algorithm
    1. Feature Selection
    2. Implementation using TPOT library
6. Application in Real World
7. End Notes

1. Intuition behind Genetic Algorithms

Let's start with the famous quote by Charles Darwin:

> *It is not the strongest of the species that survives, nor the most intelligent , but the one most responsive to change.*

You must be thinking what has this quote got to do with genetic algorithm? Actually, the entire concept of a genetic algorithm is based on the above line.

Let us understand with a basic example:

Let's take a hypothetical situation where, you are head of a country, and in order to keep your city safe from bad things, you implement a policy like this.

- You select all the good people, and ask them to extend their generation by having their children.
- This repeats for a few generations.
- You will notice that now you have an entire population of good people.

Now, that may not be entirely possible, but this example was just to help you understand the concept. So the basic idea was that we changed the input (i.e. population) such that we get better output (i.e. better country).

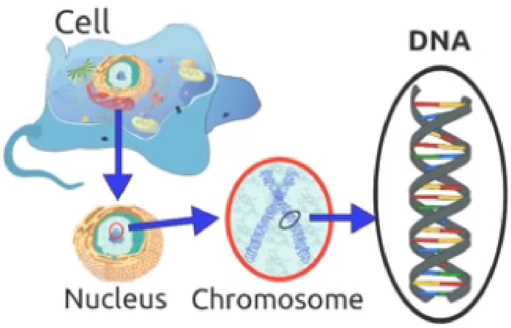
Now, I suppose you have got some intuition that the concept of a genetic algorithm is somewhat related to biology. So let's us quickly grasp some little concepts, so that we can draw a parallel line between them.

2. Biological Inspiration

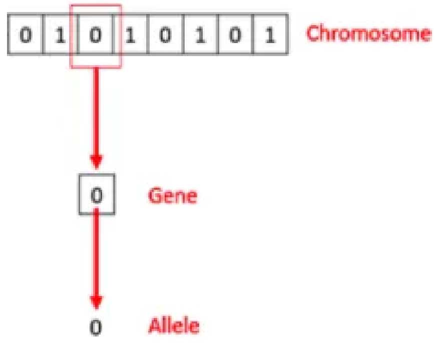
I am sure you would remember:

Cells are the basic building block of all living things.

Therefore in each cell, there is the same set of chromosomes. Chromosome are basically the strings of DNA.



Traditionally, these chromosomes are represented in binary as strings of 0's and 1's.



Source : link

A chromosome consists of genes, commonly referred as blocks of DNA, where each gene encodes a specific trait, for example hair color or eye color.

I wanted you to recall these basics concept of biology before going further. Let's get back and understand what actually is a genetic algorithm?

3. What is a Genetic Algorithm?

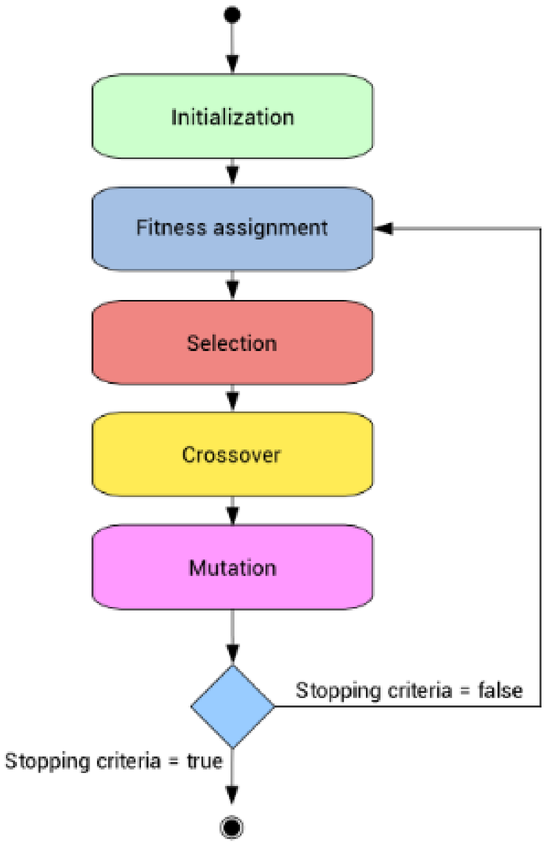
Let's get back to the example we discussed above and summarize what we did.

1. Firstly, we defined our initial population as our countrymen.
2. We defined a function to classify whether is a person is good or bad.
3. Then we selected good people for mating to produce their off-springs.
4. And finally, these off-springs replace the bad people from the population and this process repeats.

This is how genetic algorithm actually works, which basically tries to mimic the human evolution to some extent.

So to formalize a definition of a genetic algorithm, we can say that it is an optimization technique, which tries to find out such values of input so that we get the best output values or results.

The working of a genetic algorithm is also derived from biology, which is as shown in the image below.



Source: link

So, let us try to understand the steps one by one.

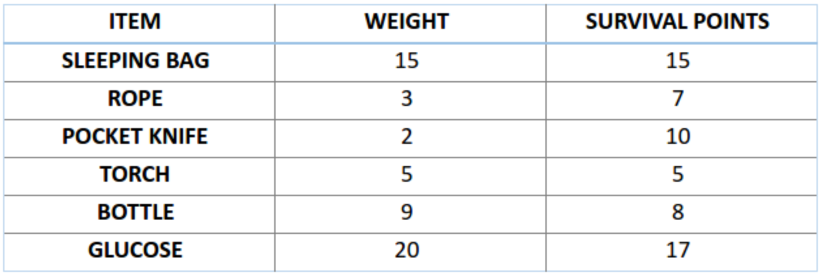
4. Steps Involved in Genetic Algorithm

Here, to make things easier, let us understand it by the famous Knapsack problem.

If you haven't come across this problem, let me introduce my version of this problem.

Let's say, you are going to spend a month in the wilderness. Only thing you are carrying is the backpack which can hold a maximum weight of **30 kg**. Now you have different survival items, each having its own "Survival Points" (which are given for each item in the table). So, your objective is maximise the survival points.

Here is the table giving details about each item.

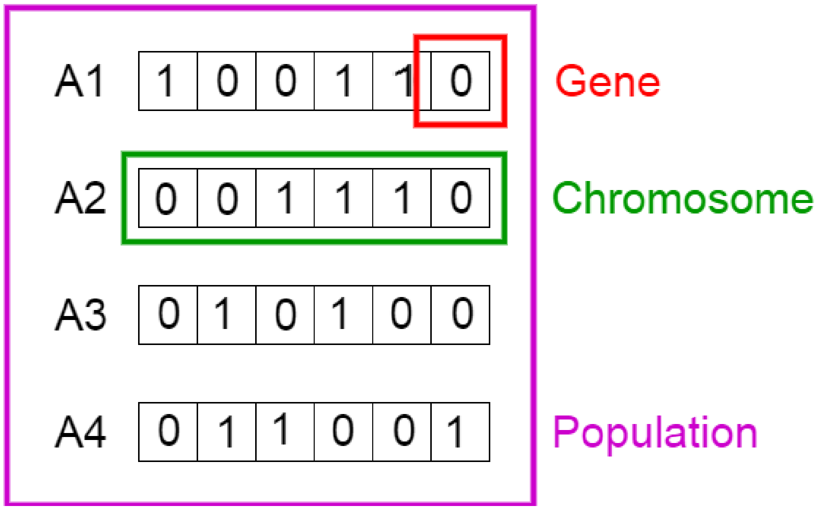


ITEM	WEIGHT	SURVIVAL POINTS
SLEEPING BAG	15	15
ROPE	3	7
POCKET KNIFE	2	10
TORCH	5	5
BOTTLE	9	8
GLUCOSE	20	17

4.1 Initialisation

To solve this problem using genetic algorithm, our first step would be defining our population. So our population will contain individuals, each having their own set of chromosomes.

We know that, chromosomes are binary strings, where for this problem 1 would mean that the following item is taken and 0 meaning that it is dropped.

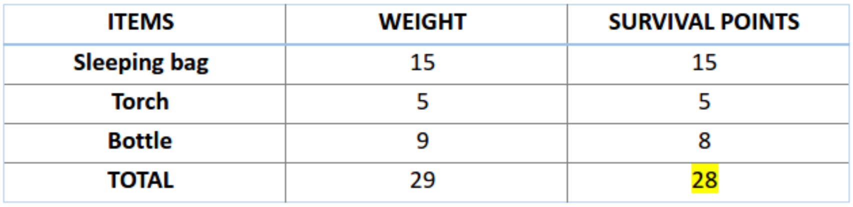


This set of chromosome is considered as our initial population.

4.2 Fitness Function

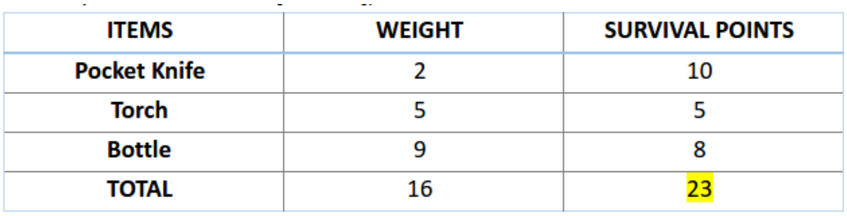
Let us calculate fitness points for our first two chromosomes.

For A1 chromosome [100110],



ITEMS	WEIGHT	SURVIVAL POINTS
Sleeping bag	15	15
Torch	5	5
Bottle	9	8
TOTAL	29	28

Similarly for A2 chromosome [001110],



ITEMS	WEIGHT	SURVIVAL POINTS
Pocket Knife	2	10
Torch	5	5
Bottle	9	8
TOTAL	16	23

So, for this problem, our chromosome will be considered as more fit when it contains more survival points.

Therefore chromosome 1 is more fit than chromosome 2.

4.3 Selection

Now, we can select fit chromosomes from our population which can mate and create their off-springs.

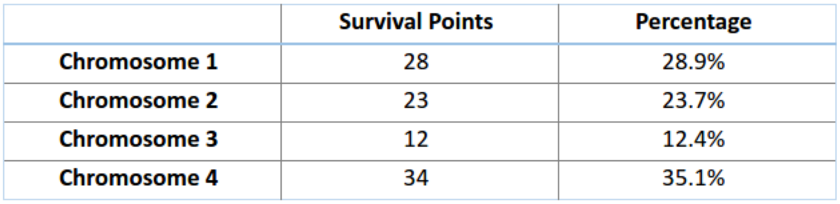
General thought is that we should select the fit chromosomes and allow them to produce off-springs. But that would lead to chromosomes that are more close to one another in a few next generation, and therefore less diversity.

Therefore, we generally use Roulette Wheel Selection method.

Don’t be afraid of name, just take a look at the image below.

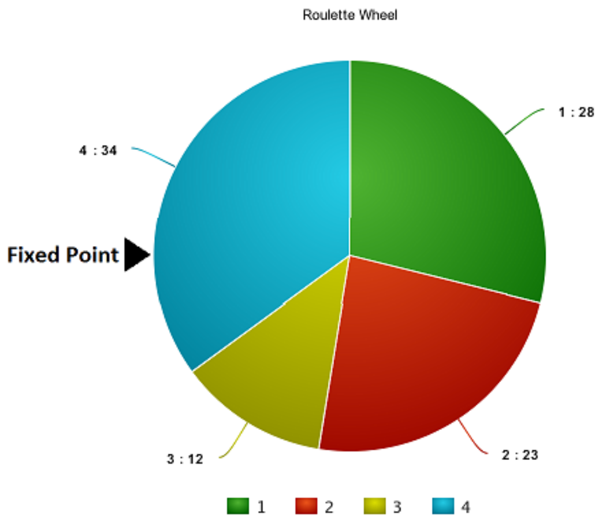
I suppose we all have seen this, either in real or in movies. So, let’s build our roulette wheel.

Consider a wheel, and let’s divide that into m divisions, where m is the number of chromosomes in our populations. The area occupied by each chromosome will be proportional to its fitness value.



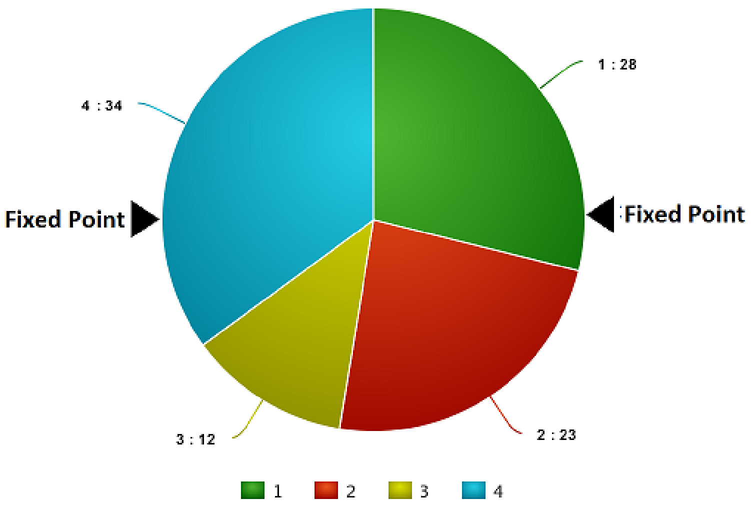
	Survival Points	Percentage
Chromosome 1	28	28.9%
Chromosome 2	23	23.7%
Chromosome 3	12	12.4%
Chromosome 4	34	35.1%

Based on these values, let us create our roulette wheel.



So, now this wheel is rotated and the region of wheel which comes in front of the fixed point is chosen as the parent. For the second parent, the same process is repeated.

Sometimes we mark two fixed point as shown in the figure below.

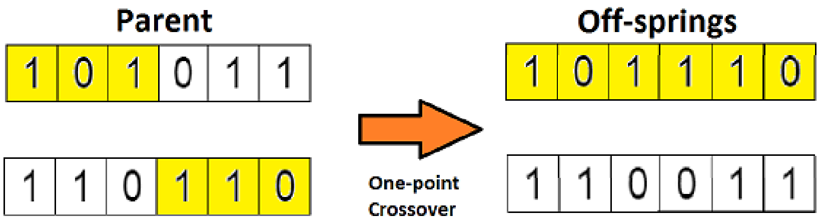


So, in this method we can get both our parents in one go. This method is known as Stochastic Universal Selection method.

4.4 Crossover

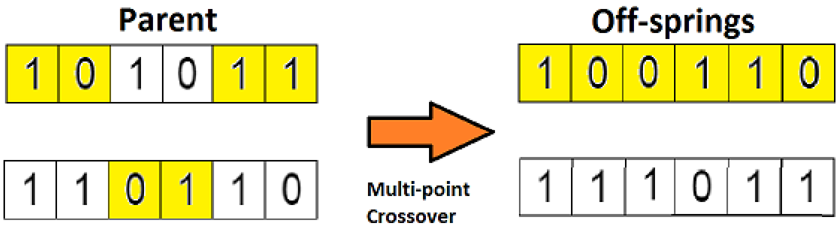
So in this previous step, we have selected parent chromosomes that will produce off-springs. So in biological terms, crossover is nothing but reproduction.

So let us find the crossover of chromosome 1 and 4, which were selected in the previous step. Take a look at the image below.



This is the most basic form of crossover, known as one point crossover. Here we select a random crossover point and the tails of both the chromosomes are swapped to produce a new off-springs.

If you take two crossover point, then it will called as multi point crossover which is as shown below.

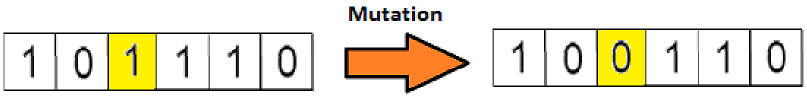


4.5 Mutation

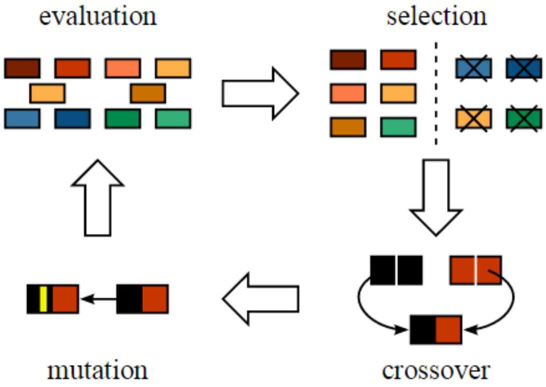
Now if you think in the biological sense, are the children produced have the same traits as their parents? The answer is NO. During their growth, there is some change in the genes of children which makes them different from its parents.

This process is known as mutation, which may be defined as a random tweak in the chromosome, which also promotes the idea of diversity in the population.

A simple method of mutation is shown in the image below.



So the entire process is summarise as shown in the figure.



Source : link

The off-springs thus produced are again validated using our fitness function, and if considered fit then will replace the less fit chromosomes from the population.

But the question is how we will get to know that we have reached our best possible solution?

So basically there are different termination conditions, which are listed below:

1. There is no improvement in the population for over x iterations.
2. We have already predefined an absolute number of generation for our algorithm.
3. When our fitness function has reached a predefined value.

Now, I suppose you have grasp the basic understanding of the genetic algorithm. So now let us look at some of the application of genetic algorithm in data science.

5. Application of Genetic Algorithm

5.1 Feature Selection

Every time you participate in a data science competition, how do you select features that are important in prediction of the target variable? You always look at the feature importance of some model, and then manually decide the threshold, and select the features which have importance above that threshold.

Is there any better way to deal with this kind of situations? Actually one of the most advanced algorithms for feature selection is genetic algorithm.

The method here is completely same as the one we did with the knapsack problem.

We will again start with the population of chromosome, where each chromosome will be binary string. 1 will denote “inclusion” of feature in model and 0 will denote “exclusion” of feature in the model.

And another difference would be that the fitness function would be changed. The fitness function here will be our accuracy metric of the competition. The more accurate our set of chromosome in predicting value, the more fit it will be.

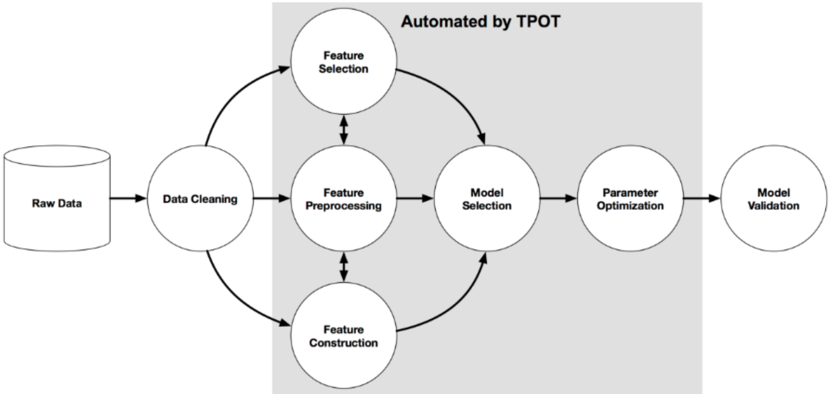
I suppose, you would now be thinking is there any use of such tough tasks. I will not answer this question now, rather let us look at the implementation of it using TPOT library and then you decide this.

5.2 Implementation using TPOT library

So finally, here the comes the part for which you have been waiting from the beginning of this article.

First, let's take a quick view on the TPOT (Tree-based Pipeline Optimisation Technique) which is build upon scikit-learn library.

A basic pipeline structure is shown in the image below.



An example Machine Learning pipeline

So the highlighted grey section in the image above is automated using TPOT. This automation is achieved using genetic algorithm.

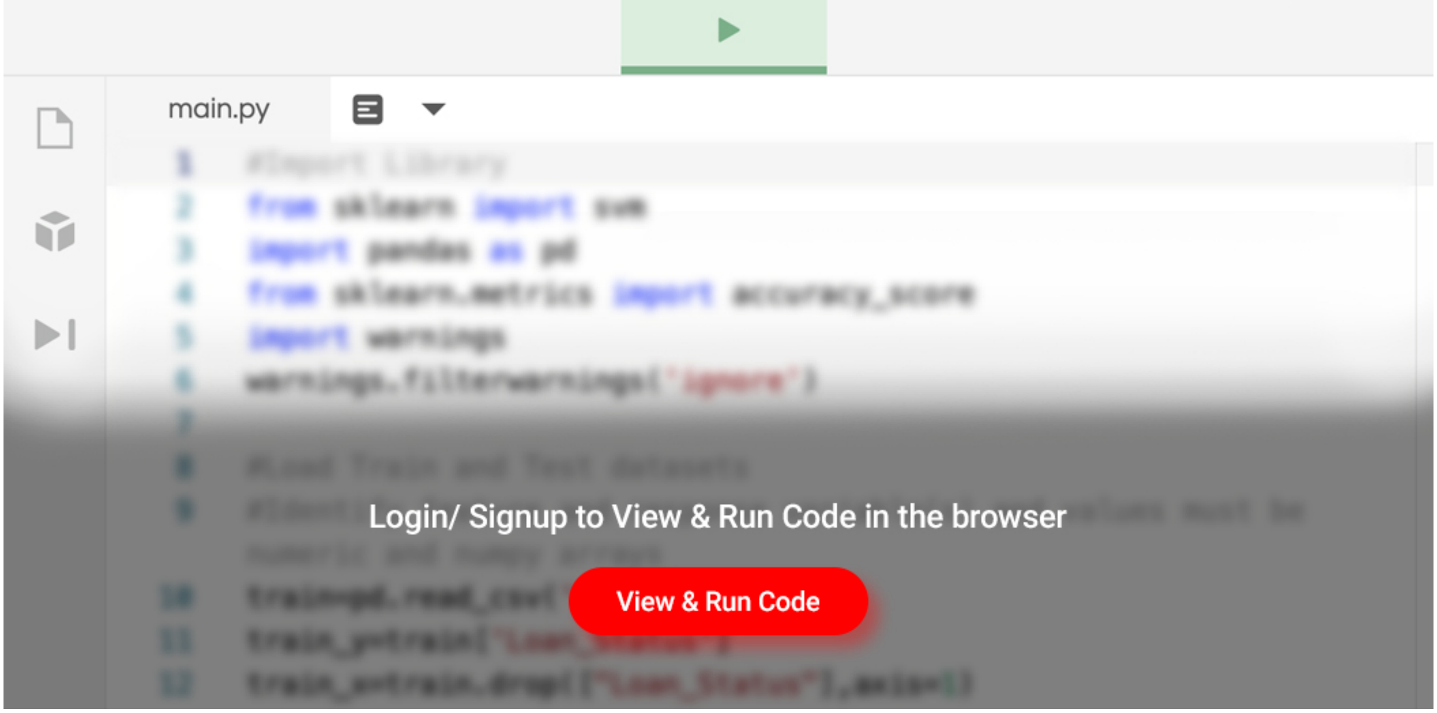
So, without going deep into this, let's directly try to implement it.

For using TPOT library, you first have to install some existing python libraries on which TPOT is build. So let us quickly install them.

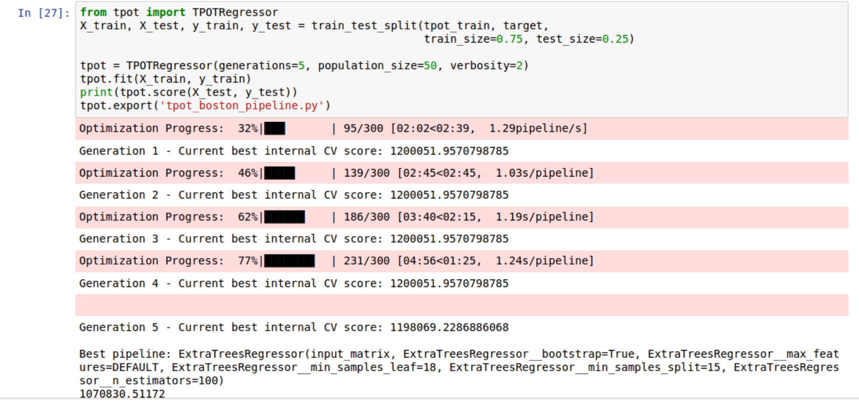
```
# installing DEAP, update_checker and tqdm

pip install deap update_checker tqdm

# installling TPOT

pip install tpot
```

For the implementation part, here I have used Big Mart Sales dataset. So quickly download the train and test file.

Now let's look at its python code.



```
#Import Library
from sklearn import svm
import pandas as pd
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings('ignore')

#Load Train and Test datasets
#Identify feature and response variable(s) and values must be
numeric and numpy arrays
train=pd.read_csv(
train_y=train['Loan_Status']
train_x=train.drop(["Loan_Status"],axis=1)
```



```
In [27]: from tpot import TPOTRegressor
         X_train, X_test, y_train, y_test = train_test_split(tpot_train, target,
                                                             train_size=0.75, test_size=0.25)

         tpot = TPOTRegressor(generations=5, population_size=50, verbosity=2)
         tpot.fit(X_train, y_train)
         print(tpot.score(X_test, y_test))
         tpot.export('tpot_boston_pipeline.py')
```

```
Optimization Progress:  32%|███        | 95/300 [02:02<02:39,  1.29pipeline/s]
Generation 1 - Current best internal CV score: 1200051.9570798785
Optimization Progress:  46%|████       | 139/300 [02:45<02:45,  1.03s/pipeline]
Generation 2 - Current best internal CV score: 1200051.9570798785
Optimization Progress:  62%|██████     | 186/300 [03:40<02:15,  1.19s/pipeline]
Generation 3 - Current best internal CV score: 1200051.9570798785
Optimization Progress:  77%|███████    | 231/300 [04:56<01:25,  1.24s/pipeline]
Generation 4 - Current best internal CV score: 1200051.9570798785

Generation 5 - Current best internal CV score: 1198069.2286886068

Best pipeline: ExtraTreesRegressor(input_matrix, ExtraTreesRegressor__bootstrap=True, ExtraTreesRegressor__max_feat
ures=DEFAULT, ExtraTreesRegressor__min_samples_leaf=18, ExtraTreesRegressor__min_samples_split=15, ExtraTreesRegres
sor__n_estimators=100)
1070830.51172
```

Once this code finishes running, `tpot_exported_pipeline.py` will contain the Python code for the optimised pipeline. We can see that ExtraTreeRegressor worked best for this problem.

```
## predicting using tpot optimised pipeline
```

```
tpot_pred = tpot.predict(tpot_test)
sub1 = pd.DataFrame(data=tpot_pred)
#sub1.index = np.arange(0, len(test)+1)
sub1 = sub1.rename(columns = {'0':'Item_Outlet_Sales'})
sub1['Item_Identifier'] = test['Item_Identifier']
sub1['Outlet_Identifier'] = test['Outlet_Identifier']
sub1.columns = ['Item_Outlet_Sales','Item_Identifier','Outlet_Identifier']
sub1 = sub1[['Item_Identifier','Outlet_Identifier','Item_Outlet_Sales']]
```

```
sub1.to_csv('tpot.csv',index=False)
```

If you submit this csv, you will notice that what I promised in the start has not been fulfilled. Was I lying to make you study all of these?

No, actually there is a simple rule of TPOT library, if you don't run TPOT for very long, then it may not find the best possible pipeline for your problem.

So, increase the number of generations, grab a cup of coffee and go out for a walk. TPOT will finish your work.

You can also do classification problems with this library. For more, I would suggest you to once check out its documentation.

Besides competitions, genetic algorithm also have many applications in the real world.

6. Applications in Real World

Genetic algorithm has many applications in real world. Here I have listed some of the interesting application, but explaining each one of them will require me an extra article.

6.1 Engineering Design

Engineering design has relied heavily on computer modeling and simulation to make design cycle process fast and economical. Genetic algorithm has been used to optimize and provide a robust solution.

Resources: link

6.2 Traffic and Shipment Routing (Travelling Salesman Problem)

This is a famous problem and has been efficiently adopted by many sales-based companies as it is time saving and economical. This is also achieved using genetic algorithm.

Winnipeg
Washington
Montana
North Dakota
Minnesota
Montreal
Maine
South Dakota
Wisconsin
Michigan
Toronto
Oregon
Idaho
Wyoming
New York
Chicago
Iowa
Nebraska
Illinois
Ohio
Pennsylvania
Nevada
Utah
United States
Colorado
Kansas
Missouri
West Virginia
Kentucky
Virginia
New York
Rhode
Connecticut
New Jersey
California
Tennessee
North Carolina
Delaware
Oklahoma
Arkansas
Maryland
Los Angeles
Arizona
New Mexico
Mississippi
South Carolina
District of Columbia
San Diego
Phoenix
Dallas
Alabama
Georgia
Texas
Louisiana
San Antonio
Houston
1
Florida
49 872
Gulf of California
Monterrey
Gulf of Mexico
Google
Mexico
Havana
Map data ©2014 Google, INEGI

Source: link

6.3 Robotics

The use of genetic algorithm in the field of robotics is quite big. Actually, genetic algorithm is being used to create learning robots which will behave as a human and will do tasks like cooking our meal, do our laundry etc.

Resources: link

Now after these I suppose, you must have developed enough curiosity to look out for some more other interesting applications of genetic algorithms. Also you can comment down if you want to share that with us.

7. End Notes

I hope that now you have gain enough understanding about what genetic algorithm is and also how to implement it using TPOT library. But this knowledge is not enough, if you don't apply it somewhere.

So try to implement it whether in any real world application or in a data science competition. If you face any difficulties, feel free to write on our discussion portal.

Did you find this article helpful? Please share your opinions / thoughts in the comments section below.