Libraries

In [1]:

```
import pandas as pd # for datasets
import numpy as np
import nltk # for tokenization

"""
VIDEO LINKI
https://www.youtube.com/watch?v=egLfjYHu69M
"""
```

Datasets

In [48]:

```
# Load the data1
data1 = pd.read_csv('datasets/spam.csv')
data1 = data1.drop(columns=['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'])
data1['Sentiment'] = data1['Sentiment'].map({'ham': 0, 'spam': 1}) # ham 0, spam
data1 = data1.dropna()
data1 = pd.concat([
    data1[data1['Sentiment'] == 0].sample(250), # verisetinden rastgele 250 adet
    data1[data1['Sentiment'] == 1].sample(250)
])
data1
```

Out[48]:

	Sentiment	Sentence
4263	0	She just broke down a list of reasons why nobo...
3877	0	What you need. You have a person to give na.
5060	0	Sorry, I'll call you later. I am in meeting sir.
1849	0	You got job in wipro:)you will get every thing...
178	0	Text her. If she doesnt reply let me know so i...
...	...	...
716	1	+449071512431 URGENT! This is the 2nd attempt ...
1733	1	Hi, this is Mandy Sullivan calling from HOTMIX...
3764	1	Someone U know has asked our dating service 2 ...
432	1	Congrats! Nokia 3650 video camera phone is you...
2913	1	Sorry! U can not unsubscribe yet. THE MOB offe...

500 rows × 2 columns

In [49]:

```
# Load the data2
# sentiment 0: negative 1: positive
data2 = pd.read_csv('datasets/beyazperde.csv')
data2 = data2[['Sentence', 'Sentiment']]
data2 = data2.dropna()
data2 = pd.concat([
    data2[data2['Sentiment'] == 0].sample(250),  # verisetinden rastgele 250 adet
    data2[data2['Sentiment'] == 1].sample(250)
])
data2
```

Out[49]:

	Sentence	Sentiment
3027	filmi kurtaricak iki sahne adam sandler in cep...	0
7087	ben ilk filmi izlemedim ama kiyaslayamayacagim...	0
2724	arkadaslar filmden hi&#65533;bisi anlamdim.galiba o d...	0
5661	kesinlikle ve kesinlikle !!! k&#65533;t&#65533; &#65533;tesi puanin...	0
3854	o kadar &#65533;ok sa&#65533;ma bir film izledigimi hatirlam...	0
...	...	...
7898	&#65533;ogmuz gibi benimde sinemada gittigim ilk film...	1
566	yaw cok etkleyc saglam flm konusu cok y oyuncu...	1
6960	m&#65533;kemmel bir film . \r\n	1
2624	&#65533;ok samimi bir film.iki sevdigim oyuncu.replik...	1
1123	olduk&#65533;a kaliteli bir filmdi.sewerek izledim.....	1

500 rows × 2 columns

In [50]:

```
# Load the data3
data3 = pd.read_csv('datasets/financial_sentiment.csv')
data3['Sentiment'] = data3['Sentiment'].map({'negative': 0, 'positive': 1}) # neg
data3 = data3.dropna()
data3 = pd.concat([
    data3[data3['Sentiment'] == 0].sample(250),  # verisetinden rastgele 250 adet
    data3[data3['Sentiment'] == 1].sample(250)
])
data3
```

Out[50]:

	Sentence	Sentiment
2102	Philip Morris, BAT Sue Over Law Taking Brandin...	0.0
5804	The administrators have indicated a need for 9...	0.0
4753	Based on the first quarter result , existing o...	0.0
153	In January-June 2010 , diluted loss per share ...	0.0
5801	At this growth rate , paying off the national ...	0.0
...	...	...
2348	2 Turnaround Buys For 2016? BHP Billiton plc A...	1.0
276	Tesco closes in on new chairman with Dixons Ca...	1.0
341	Through the acquisition Solteq will expand its...	1.0
2606	Also, BMO had just initiated $incy with an out...	1.0
2074	Demand seems to have hit bottom now , and some...	1.0

500 rows × 2 columns

In [51]:

```
# Load the data4
data4 = pd.read_csv('datasets/magaza_yorumlari_duygu_analizi.csv', encoding='utf-
data4['Sentiment'] = data4['Sentiment'].map({'Olumsuz': 0, 'Olumlu': 1}) # Olumsu
data4 = data4.dropna()
data4 = pd.concat([
    data4[data4['Sentiment'] == 0].sample(250),  # verisetinden rastgele 250 adet
    data4[data4['Sentiment'] == 1].sample(250)
])
data4
```

Out[51]:

	Sentence	Sentiment
1593	Teşhir ürün gönderilmiş. Kutusu 10 yıllık kutu...	0.0
6057	malesef ürün sıfır değil gibi, altındaki çizik...	0.0
5428	Alacak olanlara tavsiye etmiyorum kesinlikle b...	0.0
6061	ürün arızalı çıktı ve samsung servisi arızalı ...	0.0
3397	Ne yazık ki 4 ay sonra parçalar da sıkıntı yaş...	0.0
...	...	...
3781	Ürünü kullanmadım ama gönderim hiç bekletilmed...	1.0
10048	kesinlikle tavsiye ederim 👍	1.0
10568	Açıklamalarda yazdığı özelliklerin tümü var v...	1.0
7012	Üründen çok memnunum iyi ki almışım fiyat-perf...	1.0
5162	Süper hız. Teşekkürler	1.0

500 rows × 2 columns

In [77]:

```
# Load the data5
# Sentiment 0: negative 1: positive
data5 = pd.read_csv('datasets/movie.csv')
data5 = data5.drop(columns=['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'])
data5 = data5.dropna()
data5 = pd.concat([
    data5[data5['Sentiment'] == '0'].sample(250),  # verisetinden rastgele 250 ad
    data5[data5['Sentiment'] == '1'].sample(250)
])
data5
```

Out[77]:

	Sentence	Sentiment
4997	I rented this thinking it would be pretty good...	0
3577	I read a lot of high hopes from readers of the...	0
745	I'm Irish and I've been living in Denmark for ...	0
5946	This one acts as a satire during the women's r...	0
3771	Firstly, this is NOT an adaptation of a Stephe...	0
...	...	...
3447	He is very good in this role as a disaffected ...	1
7348	 So, not being a poet myself, I hav...	1
7520	Moonwalker is probably not the film to watch i...	1
7432	Purple Rain... what else can i say, the title ...	1
4031	The effects of job related stress and the pres...	1

500 rows × 2 columns

Split Dataset

In [29]:

```
"""
Verisetinin sentence ve sentiment şeklinde ayrildi
"""
x =data3[['Sentence']]
y = data3[['Sentiment']]
x
```

Out[29]:

	Sentence
952	@saltwaternurse $INO can test 10 again in next...
1256	Standard Chartered Not Raising Capital Yet As ...
3468	$TWTR sad the only thing to move this is a tak...
4811	End Of Day Scan: Stochastic Overbought $JDST$...
3214	$QCOR a little pullback is fine but if this er...
...	...
4496	Finland 's national carrier Finnair PLC carrie...
4087	`` It allows the young child to move forward w...
3645	By combining its existing solutions into a sin...
430	Finnish fibers and plastic products maker Suom...
3740	At the same time I am delighted by the fact th...

500 rows × 1 columns

In [30]:

```
# x diziye çevrildi
x = np.array(x)
x
```

```
       ['Operating profit surged to EUR21m from EUR106 ,000 .'],
       ['Kalmar has been awarded a new 5-year contract to supply
its Rough Terrain Container Handler RTCH .'],
       ["London 's leading shares today jumped almost 100 points
, or 1.7 % , as the market opened ."],
       ["Buffett's Berkshire builds Deere stake, dumps Exxon"],
       ['$RCON some upside today. This thing is severe low float.
If there is catalyst for this guy. It can run.'],
       ['Loaded up on $bsx yesterday,looking good now. Still thin
k way more upside than downside.'],
       ['The company said that it has started to investigate stre
amlining its operations in order to meet the tightening competiti
on on the mobile phone charger market .'],
       ['Amazon to attack UK grocery market with Morrisons dea
l'],
       ['As a result , the distribution companies will start to d
istribute , in addition to their current product offering , Shima
no reels , rods and other Shimano fishing tackle products on an e
xclusive basis .'],
       ['Both operating profit and net sales for the nine-month p
```

Preprocessing-Tokenization

In [31]:

```
from nltk.corpus import stopwords
"""
Cümleler kelimelere ayrildiktan sonra stopwords word denilen gereksiz kelimeler v
Daha sonra ise her kelime tek bir listede toplanarak veri setinin kelime havuzu o
"""
stop_words = set(stopwords.words('english')) # türkçe veride turkish yazilmali
word_pool = x
word_pool = word_pool.ravel()

word_pool_tokens = []
for i in word_pool:
    token = nltk.word_tokenize(i)
    for word in token:
        if (word not in stop_words) and (word.isalnum()):
            word_pool_tokens.append(word)

# print(word_pool_tokens) # kelime havuzu
word_pool_tokens
```

```
 'recall',
 'Tesla',
 'recalls',
 'ModelX',
 'faulty',
 '3rd',
 'row',
 'seat',
 'https',
 'TSLA',
 'https',
 'BOBE',
 'premarket',
 'In',
 'Sweden',
 'oversupply',
 'pharmacies',
 'Agricultural',
 'newspaper',
 'Maaseudun',
```

In [32]:

```
import random
print(random.choice(word_pool_tokens))
```

```
price
```

Genetic Algorithm

In [33]:

```
def predict(best_individual, sentences):
    """
    best_individual: en iyi birey
    sentences: siniflandirilacak cümle
    """
    results = []
    pos_arr = best_individual[:len(best_individual) // 2]
    neg_arr = best_individual[len(best_individual) // 2:]

    for i in range(len(sentences)):
        count_pos = 0
        count_neg = 0

        for j in sentences[i]:
            tokens = nltk.word_tokenize(j)

            for token in tokens:
                if (token in pos_arr) and (token not in neg_arr): # eğer kelime b
                    count_pos += 1
                if (token in neg_arr) and (token not in pos_arr): # eğer kelime b
                    count_neg += 1

            if count_pos > count_neg:
                result = 1 # positive
            elif count_neg > count_pos:
                result = 0 # negative
            else:
                a = ["Negative", "Positive"] # olumlu ve olumsuz kelime sayıları
                random_a = random.choice(a)

                if random_a == a[0]:
                    result = 0
                if random_a == a[1]:
                    result = 1

            results.append(result)

    return results

def fitness_function(individual, x):
    """
    individual: siniflandirma için kullanilacak birey
    x: sadece cümlelerin olduğu veriseti

    İyilik fonksiyonu oluşturulurken öncelikle birey ikiye bölünmüştür.
    Daha sonra veri setindeki cümleler kelimelere ayrilmiştir.
    Ayrilan kelimeler sadece bireyin ilk yarisinda ise olumlu, sadece bireyin iki
    Daha sonra ise olumlu ve olumsuz sayilarindan fazla olani toplam değişkenine
    """
    toplam = 0
    pos_arr = individual[:len(individual) // 2]
    neg_arr = individual[len(individual) // 2:]

    for i in range(len(x)):
        count_pos = 0
        count_neg = 0
        for j in x[i]:
            tokens = nltk.word_tokenize(j)
```

```
            for token in tokens:
                if (token in pos_arr) and (token not in neg_arr):
                    count_pos += 1
                if (token in neg_arr) and (token not in pos_arr):
                    count_neg += 1

            if count_pos > count_neg:
                toplam += count_pos

            elif count_neg > count_pos:
                toplam += count_neg

    return toplam

def selection(population, scores, k=3):
    """
    population: bütün bireylerin toplami
    scores: fitness fonksiyonunun döndürdüğü değerlerin oluşturduğu dizi
    k: bireyin kaç parçaya ayrilacağinin sayisi. Örneğin k = 3 ise bireyin 2 parç
    """
    selection_ix = np.random.randint(len(population))
    for ix in np.random.randint(0, len(population), k-1):
        if scores[ix] > scores[selection_ix]:
            selection_ix = ix
    return population[selection_ix]

def crossover(parent1, parent2, r_cross):
    """
    parent1: selection fonksiyonu ile seçilen bireylerden birisi
    parent2: selection fonksiyonu ile seçilen bireylerden birisi
    r_cross: crossover olasiliği
    """
    child1, child2 = parent1.copy(), parent2.copy()
    if np.random.rand() < r_cross:
        pt = np.random.randint(1, len(parent1)-2)
        child1 = np.concatenate((parent1[:pt], parent2[pt:]), axis=None)
        child2 = np.concatenate((parent2[:pt], parent1[pt:]), axis=None)
    return [child1, child2]

def mutation(bitstring, r_mut):
    """
    bitstring: crossoverdan sonra bazi kisimlari değiştirilecek birey
    r_mut: mutasyon olasiliği
    """
    for i in range(len(bitstring)):
        if np.random.rand() < r_mut:
            bitstring[i] = random.choice(word_pool_tokens)
    return bitstring #sonradan eklendi

def genetic_algorithm(x, population, fitness, n_iter, n_pop, r_cross, r_mut):
    """
    x: sadece cümlelerin olduğu veriseti
    population: başta rastgele oluşturduğumuz populasyon
    fitness: fitness function
    n_iter: nesil sayisi
    n_pop: populasyondaki birey sayisi
    r_cross: crossover olasiliği
    r_mut: mutasyon olasiliği
    """
    best_eval_arr = []
    avg_arr = []
```

```
    best = 0
    best_eval = fitness(population[0], x)
    for gen in range(n_iter):
        scores = [] # sonradan eklendi
        for c in population:
            score = fitness(c, x)
            scores.append(score)

        for i in range(n_pop):
            if scores[i] > best_eval:
                print("En iyi bireyde Artiş")
                best, best_eval = population[i], scores[i]

        avg = sum(scores) / len(scores)
        print(">%d, new best f(%s) = %f" % (gen,  best, best_eval))
        print("Average fitness function is: ", avg)
        print("Best fitness function is: ", best_eval)

        best_eval_arr.append(best_eval)
        avg_arr.append(avg)
        selected = [selection(population, scores) for _ in range(n_pop)] # ilgili
        children = list()

        for i in range(0, n_pop, 2):
            parent1, parent2 = selected[i], selected[i+1]

            for c in crossover(parent1, parent2, r_cross): # crossover
                c = mutation(c, r_mut) # mutasyon
                children.append(c)

        population = children

    return [best_eval_arr, avg_arr, best, best_eval]
```

Training

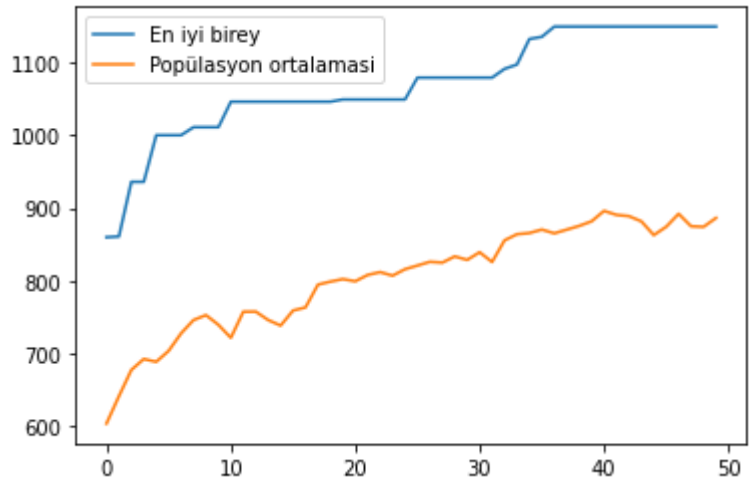
In [34]:

```
length = 100 # ödevdeki N değeri
n_pop = 100 # populasyon büyüklüğü
n_iter = 50  # nesil sayısı
r_cross = 0.7 # crossover rate
r_mut = 0.1 # mutation rate
population = []

for i in range(n_pop): # for döngüsü içerisinde rastgele kelimelerden oluşan popu
    individual = []
    for j in range(length): # rastgele kelime listesi
        element = random.choice(word_pool_tokens)
        individual.append(element)
    individual = [word for word in individual if not word in stop_words]
    population.append(individual)


best_eval_arr, avg_arr, best_individual, best_score = genetic_algorithm(x, popula

print('Done!')
print('f(%s) = %f' % (best_individual, best_score))
```

```
rlier'
 '423' 'Services' 'SEL' 'October' 'SysOpen' 'share' 'operating'
 'Wednesday' 'prices' 'superior' 'pct' 'industry' '90' 'Motors'
'60' 'mln'
 'Gabbana' 'Britvic' 'We' 'expected' 'upward' 'Finland' 'long' 'M
orrisons'
 'business' 'great']) = 1079.000000
Average fitness function is:  828.91
Best fitness function is:  1079
>30, new best f(['January' 'compared' 'pieces' 'period' 'BBH' 'st
rategy' 'million' 'TSLA'
 '2008' 'thing' 'http' 'profit' 'mn' 'Alus' 'net' 'https' 'shapin
g' 'euro'
 'short' 'first' 'Oyj' 'percent' 'The' 'million' '2009' 'EUR'
 'corresponding' 'http' 'In' 'Chairman' 'Tuesday' 'ago' 'Operatin
g'
 'Siemens' 'It' 'quality' 'fears' 'The' 'The' 'maximum' 'profit'
'world'
 'sales' 'The' 'items' 'slipped' 'April' '5' 'T' 'Helsinki' '200
```

In [35]:

```
import matplotlib.pyplot as plt

plt.plot(best_eval_arr, label='En iyi birey')
plt.plot(avg_arr, label='Popülasyon ortalamasi')
plt.legend()
plt.show()
```



In [37]:

```
# etiket verimiz diziye dönüştürüldü
y = y.to_numpy()
y = y.ravel()
```

In [38]:

```
pred = predict(best_individual, x) # tahmin fonksiyonumuzu çağırdık

print(y[0:10])
print(pred[0:10])
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0, 1, 1, 1, 0, 0, 0, 1, 1, 1]
```

In [39]:

```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y, pred)
cm
```

Out[39]:

```
array([[ 81, 169],
       [ 78, 172]])
```

In [40]:

```
import seaborn as sns

sns.heatmap(cm, annot=True)
plt.show()
```