This is a PUBG player Recommendation System Algorithm

Based on where you land, how you play, how often do you play and many more criteria, you will be given a list of other players who are like you.
The data set used is small as it was generated from scartch via a google form.

In [1]:

```
#Dataframe manipulation library
import pandas as pd
#Math functions, we'll only need the sqrt function so let's import only that
from math import sqrt
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Importing Data

In [2]:

```
df=pd.read_csv("C:/Users/Sanyam Srivastava/Desktop/PUBG Gameplay Survey.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	Timestamp	NameOrID	RateSkills	GameplayStyle	drop	Language	RankPusher
0	2020/04/30 8:05:17 PM GMT+5:30	MiuSan×	3	Moderate rush	Georgopol;Novo;Military Base;Pochinki;School/A...	English;Hindi	No
1	2020/04/30 8:32:39 PM GMT+5:30	AZĪlĪSPYDER	4	Mostly rushing	Georgopol;Novo;Military Base;Pochinki;School/A...	English;Hindi	Casual pusher
2	2020/05/07 11:02:04 AM GMT+5:30	AZĪlĪReApEr	4	Moderate rush	Military Base;Pochinki;School/Apartments;Rozho...	English;Hindi	Yes
3	2020/05/07 11:02:47 AM GMT+5:30	Sonic	4	Mostly rushing	Georgopol;Novo;Military Base	English	Casual pusher
4	2020/05/07 11:06:10 AM GMT+5:30	GodEater	3	Moderate rush	No preference	English;Others	No

Preprocessing

This dataset contains a few columns which needs to be dropped, like the last field and timestamp as well. They are not needed.

We will also drop the player type column, this is because we won't be needing a similar kind of player.eg: if you are a support player, you should be recommended an assault. This algo groups based on similarity and this

attribute will be not suitable.

In [4]:

```
df.drop('Timestamp', axis=1, inplace=True)
df.drop('PlayerType', axis=1, inplace=True)
df=df.iloc[:,0:12]
df.head()
```

Out[4]:

	NameOrID	RateSkills	GameplayStyle	drop	Language	RankPusher	PlayDays
0	MiuSan×	3	Moderate rush	Georgopol;Novo;Military Base;Pochinki;School/A...	English;Hindi	No	Most days
1	AZĪlĪSPYDER	4	Mostly rushing	Georgopol;Novo;Military Base;Pochinki;School/A...	English;Hindi	Casual pusher	Everyday
2	AZĪlĪReApEr	4	Moderate rush	Military Base;Pochinki;School/Apartments;Rozho...	English;Hindi	Yes	Everyday
3	Sonic	4	Mostly rushing	Georgopol;Novo;Military Base	English	Casual pusher	Most days
4	GodEater	3	Moderate rush	No preference	English;Others	No	Most days

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54 entries, 0 to 53
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   NameOrID        54 non-null     object
 1   RateSkills      54 non-null     int64 
 2   GameplayStyle   54 non-null     object
 3   drop            54 non-null     object
 4   Language        54 non-null     object
 5   RankPusher      54 non-null     object
 6   PlayDays        54 non-null     object
 7   MapPreference   54 non-null     object
 8   Driver          54 non-null     object
 9   camp            54 non-null     int64 
 10  snake           54 non-null     int64 
 11  bridgecamp      54 non-null     int64 
dtypes: int64(4), object(8)
memory usage: 5.2+ KB
```

We need to convert the columns with multiple inputs into a list of inputs.

In [6]:

```
#Every multi valued attribute is separated by a ; so we simply have to call the split fun
ction on ;
df['drop'] = df['drop'].str.split(';')
df['Language'] = df['Language'].str.split(';')
df['MapPreference'] = df['MapPreference'].str.split(';')
```

In [7]:

```
df.head()
```

Out[7]:

	NameOrID	RateSkills	GameplayStyle	drop	Language	RankPusher	PlayDays	MapPreference	Driver
				[Georgopol, Novo,	[English		Most		

	NameOrID	RateSkills	GameplayStyle	drop	Language	RankPusher	PlayDays	MapPreference	Driver
0	MiuSan×	3	Moderate rush	Military Base, Pochinki, Sch...	[English, Hindi]	No	Most days	[Erangel]	No
1	AZĪlĪSPYDER	4	Mostly rushing	[Georgopol, Novo, Military Base, Pochinki, Sch...	[English, Hindi]	Casual pusher	Everyday	[Erangel, Vikendi]	No prefernce
2	AZĪlĪReApEr	4	Moderate rush	[Military Base, Pochinki, School/Apartments, R...	[English, Hindi]	Yes	Everyday	[Erangel]	Yes
3	Sonic	4	Mostly rushing	[Georgopol, Novo, Military Base]	[English]	Casual pusher	Most days	[Erangel]	Yes
4	GodEater	3	Moderate rush	[No preference]	[English, Others]	No	Most days	[no preference]	No prefernce

Encoding Multi-input columns

Here, we split all the entries of columns with multiple values seperately. This is just like hot bar encoding. Other than this, we need to convert Categorical data to numberic and then perform hot bar encoding on them as well.

Here, I'm also alloting weightage values to the attributes. Language and map preference are the most important ones so they've value 2. Plane path, which is randomly decided every match has a huge impact on drop locations and this is why it has a weightage of 0.5

In [8]:

```
for index,row in df.iterrows():
    for language in row['Language']:
        df.at[index,language]=2
    for dr in row['drop']:
        df.at[index,dr]=0.5
    for map in row['MapPreference']:
        df.at[index,map]=1

df=df.fillna(0)
df.head()
```

Out[8]:

	NameOrID	RateSkills	GameplayStyle	drop	Language	RankPusher	PlayDays	MapPreference	Driver
0	MiuSan×	3	Moderate rush	[Georgopol, Novo, Military Base, Pochinki, Sch...	[English, Hindi]	No	Most days	[Erangel]	No
1	AZĪlĪSPYDER	4	Mostly rushing	[Georgopol, Novo, Military Base, Pochinki, Sch...	[English, Hindi]	Casual pusher	Everyday	[Erangel, Vikendi]	No prefernce
2	AZĪlĪReApEr	4	Moderate rush	[Military Base, Pochinki, School/Apartments, R...	[English, Hindi]	Yes	Everyday	[Erangel]	Yes
3	Sonic	4	Mostly rushing	[Georgopol, Novo, Military Base]	[English]	Casual pusher	Most days	[Erangel]	Yes
4	GodEater	3	Moderate rush	[No preference]	[English, Others]	No	Most days	[no preference]	No prefernce

5 rows × 66 columns

We delete the initial columns as they are not needed.

In [9]:

```
df.drop('MapPreference', axis=1, inplace=True)
df.drop('Language', axis=1, inplace=True)
df.drop('drop', axis=1, inplace=True)
df.head()
```

Out[9]:

	NameOrID	RateSkills	GameplayStyle	RankPusher	PlayDays	Driver	camp	snake	bridgecamp	English	...	zabaya
0	MiuSan×	3	Moderate rush	No	Most days	No	2	3	2	2.0	...	0.0
1	AZĪlĪSPYDER	4	Mostly rushing	Casual pusher	Everyday	No prefernce	2	1	3	2.0	...	0.0
2	AZĪlĪReApEr	4	Moderate rush	Yes	Everyday	Yes	3	2	4	2.0	...	0.0
3	Sonic	4	Mostly rushing	Casual pusher	Most days	Yes	1	2	4	2.0	...	0.0
4	GodEater	3	Moderate rush	No	Most days	No prefernce	2	2	1	2.0	...	0.0

5 rows × 63 columns

In [10]:

```
df.columns
```

Out[10]:

```
Index(['NameOrID', 'RateSkills', 'GameplayStyle', 'RankPusher', 'PlayDays',
       'Driver', 'camp', 'snake', 'bridgecamp', 'English', 'Hindi',
       'Georgopol', 'Novo', 'Military Base', 'Pochinki', 'School/Apartments',
       'Georgopol City', 'Mylta', 'Erangel', 'Rozhok', 'Severny', 'Vikendi',
       'Others', 'No preference', 'no preference', 'Mylta Power', 'Prison',
       'No mic', 'Paradise Resort', 'ha thin', 'san martin', 'el pozo',
       'goroka', 'dobro mesto', 'volnova', 'coal mine', 'cement factory',
       'Yasnaya', 'power grid', 'villa', 'port', 'others', 'Boot Camp(sanhok)',
       'pecado', 'hacienda del patron', 'Castle(vikendi)', 'cosmodrome',
       'Miramar', 'Sanhok', 'Korean', 'los leones', 'Quarry',
       'power grid(miramar)', 'zabaya', 'vihar', 'Factory', 'Mansion',
       'la bendita', 'krichas', 'movatra', 'dino park', 'Chinese', 'Urdu'],
      dtype='object')
```

Encoding Single Input columns

We perform hot bar encoding using get_dummies. First count all values of columns with categorical data

In [11]:

```
print(df['GameplayStyle'].value_counts())
print(df['RankPusher'].value_counts())
print(df['PlayDays'].value_counts())
print(df['Driver'].value_counts())
```

```
Moderate rush            32
Safe with little risk     8
Mostly rushing            8
Always rushing            5
Very Safe                 1
Name: GameplayStyle, dtype: int64
Casual pusher    29
Yes              16
No                9
Name: RankPusher, dtype: int64
```

```
Everyday     27
Most days    20
Weekends      5
Weekdays      2
Name: PlayDays, dtype: int64
Yes             31
No prefernce    12
No              11
Name: Driver, dtype: int64
```

This is going to create new attributes by spliting one into it's number of unique entries

In [12]:

```
df=pd.get_dummies(df, columns=["RankPusher","GameplayStyle","PlayDays","Driver"],prefix=["RankPusher","GameplayStyle","PlayDays","Driver"])
df=pd.get_dummies(df, columns=["RateSkills","camp","snake","bridgecamp"],prefix=["RateSkills","camp","snake","bridgecamp"])
```

In [13]:

```
df.head()
```

Out[13]:

	NameOrID	English	Hindi	Georgopol	Novo	Military Base	Pochinki	School/Apartments	Georgopol City	Mylta	...	snake_1	snak
0	MiuSan×	2.0	2.0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	...	0	
1	AZĪlĪSPYDER	2.0	2.0	0.5	0.5	0.5	0.5	0.5	0.0	0.5	...	1	
2	AZĪlĪReApEr	2.0	2.0	0.0	0.0	0.5	0.5	0.5	0.0	0.5	...	0	
3	Sonic	2.0	0.0	0.5	0.5	0.5	0.0	0.0	0.0	0.0	...	0	
4	GodEater	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	

5 rows × 89 columns

In [14]:

```
print(df.columns)
```

```
Index(['NameOrID', 'English', 'Hindi', 'Georgopol', 'Novo', 'Military Base',
       'Pochinki', 'School/Apartments', 'Georgopol City', 'Mylta', 'Erangel',
       'Rozhok', 'Severny', 'Vikendi', 'Others', 'No preference',
       'no preference', 'Mylta Power', 'Prison', 'No mic', 'Paradise Resort',
       'ha thin', 'san martin', 'el pozo', 'goroka', 'dobro mesto', 'volnova',
       'coal mine', 'cement factory', 'Yasnaya', 'power grid', 'villa', 'port',
       'others', 'Boot Camp(sanhok)', 'pecado', 'hacienda del patron',
       'Castle(vikendi)', 'cosmodrome', 'Miramar', 'Sanhok', 'Korean',
       'los leones', 'Quarry', 'power grid(miramar)', 'zabaya', 'vihar',
       'Factory', 'Mansion', 'la bendita', 'krichas', 'movatra', 'dino park',
       'Chinese', 'Urdu', 'RankPusher_Casual pusher', 'RankPusher_No',
       'RankPusher_Yes', 'GameplayStyle_Always rushing',
       'GameplayStyle_Moderate rush', 'GameplayStyle_Mostly rushing',
       'GameplayStyle_Safe with little risk', 'GameplayStyle_Very Safe',
       'PlayDays_Everyday', 'PlayDays_Most days', 'PlayDays_Weekdays',
       'PlayDays_Weekends', 'Driver_No', 'Driver_No prefernce', 'Driver_Yes',
       'RateSkills_2', 'RateSkills_3', 'RateSkills_4', 'RateSkills_5',
       'camp_1', 'camp_2', 'camp_3', 'camp_4', 'camp_5', 'snake_1', 'snake_2',
       'snake_3', 'snake_4', 'snake_5', 'bridgecamp_1', 'bridgecamp_2',
       'bridgecamp_3', 'bridgecamp_4', 'bridgecamp_5'],
      dtype='object')
```

Setting index of the dataset to NameOrID because they should be unique entries.

In [15]:

```
df=df.set_index(df['NameOrID'])
```

```
df.head()
```

Out[15]:

NameOrID	NameOrID	English	Hindi	Georgopol	Novo	Military Base	Pochinki	School/Apartments	Georgopol City	Mylta	...	sn
MiuSan×	MiuSan×	2.0	2.0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	...	
AZÏIÏSPYDER	AZÏIÏSPYDER	2.0	2.0	0.5	0.5	0.5	0.5	0.5	0.0	0.5	...	
AZÏIÏReApEr	AZÏIÏReApEr	2.0	2.0	0.0	0.0	0.5	0.5	0.5	0.0	0.5	...	
Sonic	Sonic	2.0	0.0	0.5	0.5	0.5	0.0	0.0	0.0	0.0	...	
GodEater	GodEater	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	

5 rows × 89 columns

View all columns along with attributes to confirm that all required entries are numeric.

In [16]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 54 entries, MiuSan× to 5165613933
Data columns (total 89 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   NameOrID                              54 non-null     object
 1   English                               54 non-null     float64
 2   Hindi                                 54 non-null     float64
 3   Georgopol                             54 non-null     float64
 4   Novo                                  54 non-null     float64
 5   Military Base                         54 non-null     float64
 6   Pochinki                              54 non-null     float64
 7   School/Apartments                     54 non-null     float64
 8   Georgopol City                        54 non-null     float64
 9   Mylta                                 54 non-null     float64
 10  Erangel                               54 non-null     float64
 11  Rozhok                                54 non-null     float64
 12  Severny                               54 non-null     float64
 13  Vikendi                               54 non-null     float64
 14  Others                                54 non-null     float64
 15  No preference                         54 non-null     float64
 16  no preference                         54 non-null     float64
 17  Mylta Power                           54 non-null     float64
 18  Prison                                54 non-null     float64
 19  No mic                                54 non-null     float64
 20  Paradise Resort                       54 non-null     float64
 21  ha thin                               54 non-null     float64
 22  san martin                            54 non-null     float64
 23  el pozo                               54 non-null     float64
 24  goroka                                54 non-null     float64
 25  dobro mesto                           54 non-null     float64
 26  volnova                               54 non-null     float64
 27  coal mine                             54 non-null     float64
 28  cement factory                        54 non-null     float64
 29  Yasnaya                               54 non-null     float64
 30  power grid                            54 non-null     float64
 31  villa                                 54 non-null     float64
 32  port                                  54 non-null     float64
 33  others                                54 non-null     float64
 34  Boot Camp(sanhok)                     54 non-null     float64
 35  pecado                                54 non-null     float64
 36  hacienda del patron                   54 non-null     float64
 37  Castle(vikendi)                       54 non-null     float64
 38  cosmodrome                            54 non-null     float64
 39  Miramar                               54 non-null     float64
```

```
 40  Sanhok                                  54 non-null     float64
 41  Korean                                  54 non-null     float64
 42  los leones                              54 non-null     float64
 43  Quarry                                  54 non-null     float64
 44  power grid(miramar)                     54 non-null     float64
 45  zabaya                                  54 non-null     float64
 46  vihar                                   54 non-null     float64
 47  Factory                                 54 non-null     float64
 48  Mansion                                 54 non-null     float64
 49  la bendita                              54 non-null     float64
 50  krichas                                 54 non-null     float64
 51  movatra                                 54 non-null     float64
 52  dino park                               54 non-null     float64
 53  Chinese                                 54 non-null     float64
 54  Urdu                                    54 non-null     float64
 55  RankPusher_Casual pusher                54 non-null     uint8
 56  RankPusher_No                           54 non-null     uint8
 57  RankPusher_Yes                          54 non-null     uint8
 58  GameplayStyle_Always rushing            54 non-null     uint8
 59  GameplayStyle_Moderate rush             54 non-null     uint8
 60  GameplayStyle_Mostly rushing            54 non-null     uint8
 61  GameplayStyle_Safe with little risk     54 non-null     uint8
 62  GameplayStyle_Very Safe                 54 non-null     uint8
 63  PlayDays_Everyday                       54 non-null     uint8
 64  PlayDays_Most days                      54 non-null     uint8
 65  PlayDays_Weekdays                       54 non-null     uint8
 66  PlayDays_Weekends                       54 non-null     uint8
 67  Driver_No                               54 non-null     uint8
 68  Driver_No prefernce                     54 non-null     uint8
 69  Driver_Yes                              54 non-null     uint8
 70  RateSkills_2                            54 non-null     uint8
 71  RateSkills_3                            54 non-null     uint8
 72  RateSkills_4                            54 non-null     uint8
 73  RateSkills_5                            54 non-null     uint8
 74  camp_1                                  54 non-null     uint8
 75  camp_2                                  54 non-null     uint8
 76  camp_3                                  54 non-null     uint8
 77  camp_4                                  54 non-null     uint8
 78  camp_5                                  54 non-null     uint8
 79  snake_1                                 54 non-null     uint8
 80  snake_2                                 54 non-null     uint8
 81  snake_3                                 54 non-null     uint8
 82  snake_4                                 54 non-null     uint8
 83  snake_5                                 54 non-null     uint8
 84  bridgecamp_1                            54 non-null     uint8
 85  bridgecamp_2                            54 non-null     uint8
 86  bridgecamp_3                            54 non-null     uint8
 87  bridgecamp_4                            54 non-null     uint8
 88  bridgecamp_5                            54 non-null     uint8
dtypes: float64(54), object(1), uint8(34)
memory usage: 25.4+ KB
```

Recommendation System Algorithm

Preprocessing and encoding is complete, we proceed with the methodology which is based on the concept of Context Based Recommendation System.
This technique attempts to figure out what a user's aspects of gameplay(attributes) is, and then recommends players that are most similar to those aspects

Select a person from the dataset on whom you want a recomendation made, you can also enter any new data entry, just make sure it's preprocessed like our df dataframe.

In [17]:

```
person=df.loc[['AZÎIÎReApEr']]
```

We will drop the nameOrID attribute from both the person as well as the df dataframe for computation. Add a column sum which is essentially sum o weightage attributes of the person for which recommendation is to be made,

In [18]:

```
person.drop('NameOrID', axis=1,inplace=True)
df.drop('NameOrID',axis=1,inplace=True)
```

In [19]:

```
person['sum']=person.sum(axis=1)
a=person.iloc[0,88]
person=person.iloc[0,0:]
person
```

Out[19]:

```
English           2.0
Hindi             2.0
Georgopol         0.0
Novo              0.0
Military Base     0.5
                 ...
bridgecamp_2      0.0
bridgecamp_3      0.0
bridgecamp_4      1.0
bridgecamp_5      0.0
sum              16.0
Name: AZĪIĪReApEr, Length: 89, dtype: float64
```

Make a copy of original dataframe. This will help in just changing the index of person and making recommendation again and again easy. Add a column sum to it, which is sum of all attributes of each row

In [20]:

```
df_copy=df
```

Now calculate the product of a column of the person with each entry of dataset.

The Concept

If attributes are similar, multiplication occurs and a number greater than 0 is the product. If the attribute do not match, the product of them is 0. We do this for all attributes and then compute sum of all. The more the value of sum, more was the similarity

In [21]:

```
df_copy=((person*df_copy).sum(axis=1))
```

Making Recommendation here. We take[1:10] because the 0th index will always be the person himself
Note: If you are using a person who is not in the dataset, change [1:10] with [0:10].

In [22]:

```
df_copy.sort_values(ascending=False)[1:10]
```

Out[22]:

```
NameOrID
MarcoPolo                 15.75
Sin6Rize                  15.50
AZĪIĪCYPHER / 510906864    15.00
SUNKEEc~HelBoy             14.50
chiyachu                  14.00
m24969                    13.75
69IVEDANT                 13.50
```

```
AZ | scarley                13.50
5650903128                 13.25
dtype: float64
```

In [ ]: