In [1]:

```python
import numpy as np
import pandas as pd
pd.options.display.max_columns=100

from IPython.core.display import display, HTML
display(HTML("<style>.container { width:90% !important; }</style>"))
```

1 Wrangling

1.1 import data

In [2]:

```python
alldata = pd.read_csv(r"StarCraftData.csv",low_memory=False)
```

In [3]:

```python
datalength = alldata.count(axis=0)[0]
print(datalength)
alldata.head()
```

```
3338
```

Out[3]:

	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	AssignToHo
0	52	5	27	10	3000	143.7180	0.003515	0.0
1	55	5	23	10	5000	129.2322	0.003304	0.0
2	56	4	30	10	200	69.9612	0.001101	0.0
3	57	3	19	20	400	107.6016	0.001034	0.0
4	58	3	32	10	500	122.8908	0.001136	0.0

1.2 Choose desired data and add onehot encoding for league

In [4]:

```python
alldata = alldata.drop('GameID',1) #remove GameID
alldata.head()
```

Out[4]:

	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkeys	AssignToHotkeys	Un
0	5	27	10	3000	143.7180	0.003515	0.000220	
1	5	23	10	5000	129.2322	0.003304	0.000259	
2	4	30	10	200	69.9612	0.001101	0.000336	
3	3	19	20	400	107.6016	0.001034	0.000213	
4	3	32	10	500	122.8908	0.001136	0.000327	

In [5]:

```python
hotleague = pd.get_dummies(alldata["LeagueIndex"])
leagues = ["Bronze","Silver","Gold","Platinum","Diamond","Master","GrandMast
for i in hotleague.columns:
    hotleague = hotleague.rename(columns={i:leagues[i-1]})
hotdata = pd.concat([hotleague,alldata],axis=1)
hotdata = hotdata.astype(float)
hotdata.head(10)
```

Out[5]:

	Bronze	Silver	Gold	Platinum	Diamond	Master	GrandMaster	LeagueIndex	Age	HoursPerWe
0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	5.0	27.0	1
1	0.0	0.0	0.0	0.0	1.0	0.0	0.0	5.0	23.0	1
2	0.0	0.0	0.0	1.0	0.0	0.0	0.0	4.0	30.0	1
3	0.0	0.0	1.0	0.0	0.0	0.0	0.0	3.0	19.0	2
4	0.0	0.0	1.0	0.0	0.0	0.0	0.0	3.0	32.0	1
5	0.0	1.0	0.0	0.0	0.0	0.0	0.0	2.0	27.0	
6	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	21.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	1.0	7.0	17.0	4
8	0.0	0.0	0.0	1.0	0.0	0.0	0.0	4.0	20.0	1
9	0.0	0.0	0.0	1.0	0.0	0.0	0.0	4.0	18.0	2

1.3 examine data

In [6]:

```python
hotdata = hotdata.astype(np.float32, copy=True, errors='raise')
hotdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3338 entries, 0 to 3337
Data columns (total 26 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Bronze                3338 non-null   float32
 1   Silver                3338 non-null   float32
 2   Gold                  3338 non-null   float32
 3   Platinum              3338 non-null   float32
 4   Diamond               3338 non-null   float32
 5   Master                3338 non-null   float32
 6   GrandMaster           3338 non-null   float32
 7   LeagueIndex           3338 non-null   float32
 8   Age                   3338 non-null   float32
 9   HoursPerWeek          3338 non-null   float32
 10  TotalHours            3338 non-null   float32
 11  APM                   3338 non-null   float32
 12  SelectByHotkeys       3338 non-null   float32
 13  AssignToHotkeys       3338 non-null   float32
 14  UniqueHotkeys         3338 non-null   float32
 15  MinimapAttacks        3338 non-null   float32
 16  MinimapRightClicks    3338 non-null   float32
 17  NumberOfPACs          3338 non-null   float32
 18  GapBetweenPACs        3338 non-null   float32
 19  ActionLatency         3338 non-null   float32
 20  ActionsInPAC          3338 non-null   float32
 21  TotalMapExplored      3338 non-null   float32
 22  WorkersMade           3338 non-null   float32
 23  UniqueUnitsMade       3338 non-null   float32
 24  ComplexUnitsMade      3338 non-null   float32
 25  ComplexAbilitiesUsed  3338 non-null   float32
dtypes: float32(26)
memory usage: 339.1 KB
```

In [7]:

```python
#check for empty
print(np.where(pd.isnull(hotdata)))
for i,j in pd.isna(hotdata).iteritems():
    if j.any() == True:
        print(j)
```

```
(array([], dtype=int64), array([], dtype=int64))
```

In [8]:

```
cols = []
for i in hotdata:
    cols.append(i)
cols
```

Out[8]:

```
['Bronze',
 'Silver',
 'Gold',
 'Platinum',
 'Diamond',
 'Master',
 'GrandMaster',
 'LeagueIndex',
 'Age',
 'HoursPerWeek',
 'TotalHours',
 'APM',
 'SelectByHotkeys',
 'AssignToHotkeys',
 'UniqueHotkeys',
 'MinimapAttacks',
 'MinimapRightClicks',
 'NumberOfPACs',
 'GapBetweenPACs',
 'ActionLatency',
 'ActionsInPAC',
 'TotalMapExplored',
 'WorkersMade',
 'UniqueUnitsMade',
 'ComplexUnitsMade',
 'ComplexAbilitiesUsed']
```

In [9]:

```
n=len(hotdata[cols[0]])
print(f"{n}")
print(hotdata[cols[0]][n-1])
print()
```

```
3338
0.0
```

In [10]:

```python
hotdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3338 entries, 0 to 3337
Data columns (total 26 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Bronze                3338 non-null   float32
 1   Silver                3338 non-null   float32
 2   Gold                  3338 non-null   float32
 3   Platinum              3338 non-null   float32
 4   Diamond               3338 non-null   float32
 5   Master                3338 non-null   float32
 6   GrandMaster           3338 non-null   float32
 7   LeagueIndex           3338 non-null   float32
 8   Age                   3338 non-null   float32
 9   HoursPerWeek          3338 non-null   float32
 10  TotalHours            3338 non-null   float32
 11  APM                   3338 non-null   float32
 12  SelectByHotkeys       3338 non-null   float32
 13  AssignToHotkeys       3338 non-null   float32
 14  UniqueHotkeys         3338 non-null   float32
 15  MinimapAttacks        3338 non-null   float32
 16  MinimapRightClicks    3338 non-null   float32
 17  NumberOfPACs          3338 non-null   float32
 18  GapBetweenPACs        3338 non-null   float32
 19  ActionLatency         3338 non-null   float32
 20  ActionsInPAC          3338 non-null   float32
 21  TotalMapExplored      3338 non-null   float32
 22  WorkersMade           3338 non-null   float32
 23  UniqueUnitsMade       3338 non-null   float32
 24  ComplexUnitsMade      3338 non-null   float32
 25  ComplexAbilitiesUsed  3338 non-null   float32
dtypes: float32(26)
memory usage: 339.1 KB
```

In [11]:

```python
hotdata.describe()
```

Out[11]:

	Bronze	Silver	Gold	Platinum	Diamond	Master	GrandMaste
count	3338.000000	3338.000000	3338.000000	3338.000000	3338.000000	3338.000000	3338.00000
mean	0.050030	0.103954	0.165668	0.242960	0.240863	0.186040	0.01048
std	0.218039	0.305247	0.371838	0.428935	0.427671	0.389197	0.10187
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.00000

In [12]:

```python
hotdata.to_csv("Data-Prepped.csv")
```

Make datasets in which Leagues are grouped for easier differentiation

groupings chosen based on data analysis in part 1 - Data Comprehension

In [37]:

```
Elos = pd.DataFrame(np.zeros((len(alldata),4)))
for i, val in enumerate(Elos.values):
    if (alldata["LeagueIndex"][i] < 4):
        Elos[0][i] = 1.0
        Elos[3][i] = 0.0
    elif(3 < alldata["LeagueIndex"][i] and alldata["LeagueIndex"][i] < 6):
        Elos[1][i] = 1.0
        Elos[3][i] = 1.0
    elif(5 < alldata["LeagueIndex"][i] and alldata["LeagueIndex"][i] < 8):
        Elos[2][i] = 1.0
        Elos[3][i] = 2.0
Elos = Elos.rename(columns={0:"Low_ELO",1:"Med_ELO",2:"High_ELO",3:"GroupedL
Elos.head()
hotdata2 = pd.concat([Elos,alldata],axis=1)
hotdata2 = hotdata2.astype(float)
hotdata2.head(10)
```

Out[37]:

	Low_ELO	Med_ELO	High_ELO	GroupedLeagueIndex	LeagueIndex	Age	HoursPerWeek	Totall
0	0.0	1.0	0.0	1.0	5.0	27.0	10.0	3
1	0.0	1.0	0.0	1.0	5.0	23.0	10.0	5
2	0.0	1.0	0.0	1.0	4.0	30.0	10.0	
3	1.0	0.0	0.0	0.0	3.0	19.0	20.0	
4	1.0	0.0	0.0	0.0	3.0	32.0	10.0	
5	1.0	0.0	0.0	0.0	2.0	27.0	6.0	
6	1.0	0.0	0.0	0.0	1.0	21.0	8.0	
7	0.0	0.0	1.0	2.0	7.0	17.0	42.0	10
8	0.0	1.0	0.0	1.0	4.0	20.0	14.0	2
9	0.0	1.0	0.0	1.0	4.0	18.0	24.0	

In [38]:

```
hotdata2.to_csv("Data-Prepped-ELO-3-Groups.csv")
```

In [43]:

```python
Elos = pd.DataFrame(np.zeros((len(alldata),3)))
for i, val in enumerate(Elos.values):
    if (alldata["LeagueIndex"][i] < 4):
        Elos[0][i] = 1.0
        Elos[2][i] = 0.0
    elif(4 < alldata["LeagueIndex"][i]):
        Elos[1][i] = 1.0
        Elos[2][i] = 1.0
Elos = Elos.rename(columns={0:"Low_ELO",1:"High_ELO",2:"GroupedLeagueIndex"}
Elos.head()
hotdata3 = pd.concat([Elos,alldata],axis=1)
hotdata3 = hotdata3.astype(float)
hotdata3.head(10)
```

Out[43]:

	Low_ELO	High_ELO	GroupedLeagueIndex	LeagueIndex	Age	HoursPerWeek	TotalHours	
0	0.0	1.0	1.0	5.0	27.0	10.0	3000.0	143.
1	0.0	1.0	1.0	5.0	23.0	10.0	5000.0	129.
2	0.0	0.0	0.0	4.0	30.0	10.0	200.0	69.
3	1.0	0.0	0.0	3.0	19.0	20.0	400.0	107.
4	1.0	0.0	0.0	3.0	32.0	10.0	500.0	122.
5	1.0	0.0	0.0	2.0	27.0	6.0	70.0	44.
6	1.0	0.0	0.0	1.0	21.0	8.0	240.0	46.
7	0.0	1.0	1.0	7.0	17.0	42.0	10000.0	212.
8	0.0	0.0	0.0	4.0	20.0	14.0	2708.0	117.
9	0.0	0.0	0.0	4.0	18.0	24.0	800.0	155.

In [44]:

```python
hotdata3.to_csv("Data-Prepped-ELO-2-Groups.csv")
```

In [ ]: