ASSIGNMENT REPORT

Jupyter Notebook

INTRODUCTION

ABC is a real-money online gaming company providing multiplayer games such as Ludo. A user can register as a player, deposit money in the platform and play games with other players on the platform. If he/she wins the game then they can withdraw the winning amount while the platform charges a nominal fee for the services. To retain players on the platform, the company ABC gives loyalty points to their players based on their activity on the platform. Loyalty points are calculated on the basis of the number of games played, deposits and withdrawal made on the platform by a particular player. The criteria to convert number of games played, deposits and withdrawal into points is given as below:

Type of Action	Weightage	Formulae	Ex.
Deposit of money on the platform	0.01	0.01 * Deposit Amount	0.01 * (1000 RS Deposit) = 10 Points
Withdrawal of money from the platform	0.005	0.005 * Withdrawal Amount	0.005 * (500 Rs Withdrawal) = 2.5 Points
How many more times did a player do deposit than withdrawal	0.001	0.001 * maximum of (#deposit - #windrawal) or 0	0.001 * max ((5-3, 0)) = 0.001 * 2 = 0.002 points where number of deposits = 5 and number of withdrawals = 3
Number of games played	0.2	0.2 *Number of Games Played	0.2 * (50 Total Games Played) = 10 Points

Final Loyalty Point Formula

Loyalty Point = (0.01 * deposit) + (0.005 * Withdrawal amount) + (0.001 * (maximum of (#deposit - #withdrawal) or 0)) + (0.2 * Number of games played)

DATA

The xlsx file had 3 sheets.

1. **Sheet Name: User Gameplay data**
   - **Column 1**: User ID
     - Data Type: Numeric
     - Description: Represents the unique identifier for each user.
   - **Column 2**: Games Played
     - Data Type: Numeric
     - Description: Represents the no. of games played by each user.
   - **Column 3**: Datetime
     - Data Type: String
     - Description: Represents timestamp when user played game.
2. **Sheet Name**: **Deposit Data**
   - **Column 1**: User Id
     - Data Type: Numeric
     - Description: Represents the unique identifier for each user.
   - **Column 2**: Datetime
     - Data Type: String
     - Description: Represents timestamp when user deposited money.
   - **Column 3**: Amount
     - Data Type: Numeric
     - Description: Represents amount which user deposited.
3. **Sheet Name**: **Withdrawal Data**
   - **Column 1**: User Id
     - Data Type: Numeric
     - Description: Represents the unique identifier for each user.
   - **Column 2**: Datetime
     - Data Type: String
     - Description: Represents timestamp when user withdrew money.
   - **Column 3**: Amount
     - Data Type: Numeric
     - Description: Represents amount which user withdrew.

APPROACH

1. Column name of Sheet 1 was changed from “User ID” to “User Id” to maintain consistency.
2. While loading, the datatype of Timestamp in all sheets were changed to Datetime for better analysis.
3. Started off with loading the data in Jupyter notebook using Pandas.
4. Segregated User Gameplay data according to Slots given.
5. Likewise, data for particular date was extracted and stored aside.
6. Applied needed functionalities and filtration to get desired insight.

RESULTS

Part A - Calculating loyalty points

1. Find Player wise Loyalty points earned by Players in the following slots :-
    - 2nd October Slot S1

```
In [14]: final_2oct
```

Out[14]:

	User Id	Result	sumDeposit	sumWithdrawl	Game_Count	loyalty_point
0	5.0	1.0	600.0	0.0	9.0	1.801
1	6.0	1.0	4000.0	0.0	2.0	0.401
2	9.0	2.0	3587.0	0.0	52.0	10.402
3	10.0	1.0	2000.0	0.0	13.0	2.601
4	11.0	1.0	2000.0	0.0	15.0	3.001
...	...	...	...	...	...	...
631	990.0	0.0	0.0	0.0	70.0	14.000
632	992.0	0.0	0.0	0.0	42.0	8.400
633	996.0	0.0	0.0	0.0	11.0	2.200
634	997.0	0.0	0.0	0.0	5.0	1.000
635	999.0	0.0	0.0	0.0	3.0	0.600

636 rows × 6 columns

    - 16th October Slot S2

In [16]:
```
final_16oct
```

Out[16]:

	User Id	Result	sumDeposit	sumWithdrawl	Game_Count	loyalty_point
0	5.0	2.0	1000.0	0.0	9.0	1.802
1	9.0	1.0	2000.0	0.0	51.0	10.201
2	11.0	1.0	4000.0	0.0	15.0	3.001
3	12.0	1.0	3200.0	0.0	3.0	0.601
4	16.0	0.0	5000.0	31305.0	25.0	474.575
...	...	...	...	...	...	...
589	989.0	0.0	0.0	0.0	65.0	13.000
590	990.0	0.0	0.0	0.0	62.0	12.400
591	992.0	0.0	0.0	0.0	44.0	8.800
592	996.0	0.0	0.0	0.0	9.0	1.800
593	999.0	0.0	0.0	0.0	1.0	0.200

594 rows × 6 columns

- 18th October Slot S1

In [18]:
```
final_18oct
```

Out[18]:

	User Id	Result	sumDeposit	sumWithdrawl	Game_Count	loyalty_point
0	5.0	1.0	500.0	0.0	7.0	1.401
1	9.0	4.0	7539.0	0.0	49.0	9.804
2	12.0	3.0	3190.0	0.0	0.0	0.003
3	14.0	1.0	100.0	0.0	0.0	0.001
4	16.0	2.0	8000.0	0.0	37.0	7.402
...	...	...	...	...	...	...
626	990.0	0.0	0.0	0.0	77.0	15.400
627	995.0	0.0	0.0	0.0	1.0	0.200
628	996.0	0.0	0.0	0.0	6.0	1.200
629	997.0	0.0	0.0	0.0	3.0	0.600
630	999.0	0.0	0.0	0.0	2.0	0.400

631 rows × 6 columns

- 26th October Slot S2

```
In [20]: final_26oct
```

Out[20]:

	User Id	Result	sumDeposit	sumWithdrawl	Game_Count	loyalty_point
0	2.0	2.0	90000.0	0.0	0.0	0.002
1	5.0	1.0	1000.0	0.0	4.0	0.801
2	9.0	0.0	1000.0	20076.0	44.0	309.940
3	10.0	1.0	1400.0	0.0	8.0	1.601
4	11.0	1.0	2000.0	0.0	11.0	2.201
...	...	...	...	...	...	...
623	991.0	0.0	0.0	0.0	1.0	0.200
624	995.0	0.0	0.0	0.0	1.0	0.200
625	996.0	0.0	0.0	0.0	4.0	0.800
626	997.0	0.0	0.0	0.0	2.0	0.400
627	999.0	0.0	0.0	0.0	1.0	0.200

628 rows × 6 columns

2. Calculate overall loyalty points earned and rank players on the basis of loyalty points in the month of October. In case of tie, number of games played should be taken as the next criteria for ranking.

```
In [28]: final_oct.sort_values(by = ['loyalty_point', 'Game_Count'], ascending = [False, False])
```

Out[28]:

	User Id	Result	sumDeposit	sumWithdrawl	Game_Count	loyalty_point
545	634.0	0.0	270000.0	11683352.0	22.0	175254.680
308	365.0	0.0	279000.0	1425235.0	2368.0	21852.125
795	920.0	0.0	211500.0	1333015.0	614.0	20118.025
312	369.0	1.0	450000.0	1326542.0	22.0	19902.531
132	162.0	0.0	56460.0	1270000.0	0.0	19050.000
...	...	...	...	...	...	...
608	703.0	2.0	8000.0	0.0	0.0	0.002
374	438.0	1.0	200000.0	0.0	0.0	0.001
653	756.0	1.0	5000.0	0.0	0.0	0.001
672	778.0	1.0	5000.0	0.0	0.0	0.001
674	780.0	1.0	300.0	0.0	0.0	0.001

996 rows × 6 columns

3. What is the average deposit amount?

Average deposit amount

```
In [29]: sum_deposit_amount = deposit_data.groupby('User Id')['Amount'].sum().reset_index(name='Total_Amount')

In [30]: sum_deposit_amount['Total_Amount'].mean()
Out[30]: 104669.64918032786
```

4. What is the average deposit amount per user in a month?

Average deposit amount per user in a month

```
In [31]: deposit_data['Month'] = pd.to_datetime(deposit_data['Datetime']).dt.month

In [32]: average_amount_per_user = deposit_data.groupby('Month')['Amount'].mean().reset_index(name='Average_Amount')

In [33]: average_amount_per_user
```

Out[33]:

	Month	Average_Amount
0	1	4896.937397
1	2	4883.574106
2	3	5154.971480
3	4	4990.715232
4	5	5179.601054
5	6	5584.013986
6	7	5881.150659
7	8	6300.759494
8	9	5074.035654
9	10	5604.514460
10	11	5074.368224
11	12	5309.488414

5. What is the average number of games played per user?

Average number of games played per user

```
In [34]: game_data_count=game_data.groupby('User Id').size().reset_index(name='Game_Count')

In [35]: game_data_count['Game_Count'].mean()
Out[35]: 355.266
```

Part B - How much bonus should be allocated to leaderboard players?

After calculating the loyalty points for the whole month find out which 50 players are at the top of the leaderboard.

```
In [36]: final_oct.sort_values(by = ['loyalty_point', 'Game_Count'], ascending = [False, False]).head(50)
```

Out[36]:

	User Id	Result	sumDeposit	sumWithdrawl	Game_Count	loyalty_point
545	534.0	0.0	270000.0	11563352.0	22.0	176284.680
308	305.0	0.0	279000.0	1428235.0	2388.0	21852.126
795	920.0	0.0	211600.0	1333015.0	614.0	20118.035
212	369.0	1.0	450000.0	1326542.0	22.0	19902.631
132	152.0	0.0	86460.0	1279000.0	0.0	19050.000
506	578.0	13.0	227000.0	1100000.0	9.0	16501.813
56	78.0	0.0	103600.0	1075400.0	379.0	16206.800
1	2.0	0.0	342000.0	898456.0	86.0	13490.190
354	415.0	1.0	200000.0	898784.0	19.0	13485.261
288	344.0	4.0	242150.0	864000.0	1.0	12960.204
79	99.0	23.0	817400.0	859325.0	4.0	12886.198
359	421.0	8.0	509100.0	825539.0	999.0	12580.443
882	988.0	0.0	0.0	814248.0	105.0	12234.720
856	969.0	0.0	200000.0	783000.0	2303.0	11755.500
62	82.0	0.0	66170.0	669910.0	3.0	10049.250
678	785.0	10.0	401600.0	648400.0	6.0	9727.210
199	245.0	7.0	78600.0	628200.0	2.0	9423.407
889	921.0	0.0	0.0	600000.0	10.0	9002.000
171	208.0	15.0	364174.0	572828.0	333.0	8660.508
164	200.0	2.0	264099.0	523000.0	2.0	7845.402
858	992.0	7.0	684988.0	466590.0	1501.0	7319.057
786	909.0	73.0	444468.0	408200.0	0.0	6123.073
887	785.0	0.0	0.0	395000.0	119.0	5963.800
396	464.0	3.0	171000.0	340660.0	292.0	5261.783
766	884.0	0.0	45000.0	349366.0	73.0	5255.075
464	540.0	0.0	255000.0	308569.0	1461.0	4824.735
403	471.0	0.0	85000.0	326486.0	87.0	4914.675
10	16.0	12.0	218200.0	307577.0	1346.0	4882.667
738	852.0	0.0	5000.0	320886.0	209.0	4855.270
174	212.0	17.0	1234986.0	319468.0	0.0	4792.037
723	836.0	0.0	54000.0	290000.0	1599.0	4667.800
536	625.0	0.0	104800.0	304745.0	17.0	4574.575
505	587.0	56.0	431600.0	295000.0	486.0	4522.256
487	566.0	24.0	117805.0	287949.0	842.0	4467.859
17	23.0	9.0	13950.0	297000.0	120.0	4479.009
792	915.0	55.0	574930.0	287609.0	7.0	4315.590
382	445.0	1.0	76942.0	277499.0	665.0	4295.466
796	922.0	0.0	150000.0	251000.0	1605.0	4126.000
166	202.0	12.0	206900.0	274434.0	1.0	4118.722
775	897.0	5.0	133700.0	260000.0	44.0	3908.505
515	601.0	15.0	509845.0	240000.0	22.0	3604.415
159	193.0	82.0	270632.0	238000.0	0.0	3570.062
886	786.0	0.0	0.0	30000.0	15558.0	3561.200
432	502.0	40.0	106250.0	190803.0	2889.0	3439.885
668	774.0	0.0	110000.0	215000.0	1043.0	3433.600
573	663.0	7.0	205000.0	184032.0	4171.0	3294.667
438	508.0	10.0	212000.0	216000.0	138.0	3257.610

The company has allocated a pool of Rs 50000 to be given away as bonus money to the loyal players. Now the company needs to determine how much bonus money should be given to the players. Should they base it on the amount of loyalty points? Should it be based on number of games? Or something else?

The player's win rate or performance in games can be taken into account for bonus allocation. Players who consistently perform well or have a high win rate can be rewarded with bonuses as an incentive to maintain their performance and encourage healthy competition.

Suggest a suitable way to divide the allocated money keeping in mind the following points:

1. Only top 50 ranked players are awarded bonus

    1. Sum the total loyalty points for employees ranked 1 to 50.
    2. Calculate the percentage share of each employee by dividing their loyalty points by the total sum of loyalty points and multiplying by 100.
    3. Calculate the bonus amount for each employee by multiplying their percentage share with the total bonus pool of 50000.

Part C

Would you say the loyalty point formula is fair or unfair? Can you suggest any way to make the loyalty point formula more robust?

The top-2 coefficient is of No. of Games played and Amount deposit respectively. This tell the loyalty points is awardee to those customers who play more no. of games and are depositing more money. Also, loyalty points are awarded when no of deposit is higher than no. of withdrawals. However, the subjective assignment of weights to these actions may raise questions about the fairness of the formula. It is important to ensure that these weights are based on justifiable and objective criteria.

Jupyter Notebook

----------End----------