PUBG Finish Placement Prediction 캐글 대회

- 대회 : https://www.kaggle.com/competitions/pubg-finish-placement-prediction (https://www.kaggle.com/competitions/pubg-finish-placement-prediction)
- 평가지표 : Mean Absolute Error
- 데이터 :
  - train_V2.csv - 4446966, 29
  - test_V2.csv - 1934174, 28
  - sample_submission_V2.csv - 1934174, 2

학습 목표

- 최소한의 데이터를 활용하여 기본 모델을 만들고, 제출해 봅니다.
- 메모리를 최적화시키는 것도 확인해 봅니다.
- 선형회귀 모델을 기본모델로 하여 예측을 수행해 봅니다.

학습 내용

코드 환경

- 캐글 대회 노트북

01. 데이터 불러오기, 탐색

처음으로 이동하기

In [1]:

```
import os

import numpy as np # linear algebra
import pandas as pd

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/pubg-finish-placement-prediction/train_V2.csv
/kaggle/input/pubg-finish-placement-prediction/test_V2.csv
/kaggle/input/pubg-finish-placement-prediction/sample_submission_V2.csv
```

02. 메모리를 최적화 - 데이터 불러오기

처음으로 이동하기

In [2]:

```
%%time

# Any results you write to the current directory are saved as output.
path = "/kaggle/input/pubg-finish-placement-prediction/"
df_train = pd.read_csv(path + "train_V2.csv")
df_train.shape
```

```
CPU times: user 20.9 s, sys: 3.04 s, total: 24 s
Wall time: 32.4 s
```

Out[2]:

```
(4446966, 29)
```

In [3]:

```
df_train.memory_usage().sum() / 1024**2
```

Out[3]:

```
983.9022064208984
```

In [4]:

```
# iinfo() 는 ()안이 자료형이 표현가능한 한계를 반환한다.
ii16 = np.iinfo(np.int16)
print(ii16.min, ii16.max)

ii32 = np.iinfo(np.int32)
print(ii32.min, ii32.max)
```

```
-32768 32767
-2147483648 2147483647
```

In [5]:

```
def reduce_mem_usage(df):

    start_mem = df.memory_usage().sum() / 1024**2
    print('메모리 사용량 {:.2f} MB'.format(start_mem))

    for col in df.columns:
        col_type = df[col].dtype

        if col_type != object:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max < np.iinfo(np.int16).max:
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max < np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max < np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                if c_min > np.finfo(np.float16).min and c_max < np.finfo(np.float16).max:
                    df[col] = df[col].astype(np.float16)
                elif c_min > np.finfo(np.float32).min and c_max < np.finfo(np.float32).max:
                    df[col] = df[col].astype(np.float32)
                else:
                    df[col] = df[col].astype(np.float64)
        else:
            df[col] = df[col].astype('category')

    end_mem = df.memory_usage().sum() / 1024**2
    print('Memory usage after optimization is: {:.2f} MB'.format(end_mem))
    print('Decreased by {:.1f}%'.format(100 * (start_mem - end_mem) / start_mem))

    return df
```

In [6]:

```
def import_data(file):
    ### 메모리를 최적화 시킨다.
    df = pd.read_csv(file, parse_dates=True, keep_date_col=True)
    df = reduce_mem_usage(df)
    return df
```

In [7]:

```
%%time

path = "/kaggle/input/pubg-finish-placement-prediction/"

print('-' * 80)
print('train data loading...')
train = import_data(path + "train_V2.csv")

print('test data loading...')
test = import_data(path + "test_V2.csv")

print('submissiion csv loading...')
df_sub = import_data(path + "sample_submission_V2.csv")

train.shape, test.shape, df_sub.shape
```

```
--------------------------------------------------------------------------------
train data loading...
메모리 사용량 983.90 MB
Memory usage after optimization is: 452.07 MB
Decreased by 54.1%
test data loading...
메모리 사용량 413.18 MB
Memory usage after optimization is: 201.94 MB
Decreased by 51.1%
submissiion csv loading...
메모리 사용량 29.51 MB
Memory usage after optimization is: 88.48 MB
Decreased by -199.8%
CPU times: user 1min 23s, sys: 3.44 s, total: 1min 26s
Wall time: 1min 31s
```

Out[7]:

```
((4446966, 29), (1934174, 28), (1934174, 2))
```

In [8]:

```
train.info(null_counts=True)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: null_
counts is deprecated. Use show_counts instead
  """Entry point for launching an IPython kernel.

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4446966 entries, 0 to 4446965
Data columns (total 29 columns):
 #   Column           Non-Null Count    Dtype   
---  ------           --------------    -----   
 0   Id               4446966 non-null  category
 1   groupId          4446966 non-null  category
 2   matchId          4446966 non-null  category
 3   assists          4446966 non-null  int8    
 4   boosts           4446966 non-null  int8    
 5   damageDealt      4446966 non-null  float16 
 6   DBNOs            4446966 non-null  int8    
 7   headshotKills    4446966 non-null  int8    
 8   heals            4446966 non-null  int8    
 9   killPlace        4446966 non-null  int8    
 10  killPoints       4446966 non-null  int16   
 11  kills            4446966 non-null  int8    
 12  killStreaks      4446966 non-null  int8    
 13  longestKill      4446966 non-null  float16 
 14  matchDuration    4446966 non-null  int16   
 15  matchType        4446966 non-null  category
 16  maxPlace         4446966 non-null  int8    
 17  numGroups        4446966 non-null  int8    
 18  rankPoints       4446966 non-null  int16   
 19  revives          4446966 non-null  int8    
 20  rideDistance     4446966 non-null  float16 
 21  roadKills        4446966 non-null  int8    
 22  swimDistance     4446966 non-null  float16 
 23  teamKills        4446966 non-null  int8    
 24  vehicleDestroys  4446966 non-null  int8    
 25  walkDistance     4446966 non-null  float16 
 26  weaponsAcquired  4446966 non-null  int16   
 27  winPoints        4446966 non-null  int16   
 28  winPlacePerc     4446965 non-null  float16 
dtypes: category(4), float16(6), int16(5), int8(14)
memory usage: 452.1 MB
```

In [9]:

```
test.info(null_counts=True)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: null_
counts is deprecated. Use show_counts instead
  """Entry point for launching an IPython kernel.

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1934174 entries, 0 to 1934173
Data columns (total 28 columns):
 #   Column           Non-Null Count    Dtype
---  ------           --------------    -----
 0   Id               1934174 non-null  category
 1   groupId          1934174 non-null  category
 2   matchId          1934174 non-null  category
 3   assists          1934174 non-null  int8
 4   boosts           1934174 non-null  int8
 5   damageDealt      1934174 non-null  float16
 6   DBNOs            1934174 non-null  int8
 7   headshotKills    1934174 non-null  int8
 8   heals            1934174 non-null  int8
 9   killPlace        1934174 non-null  int8
 10  killPoints       1934174 non-null  int16
 11  kills            1934174 non-null  int8
 12  killStreaks      1934174 non-null  int8
 13  longestKill      1934174 non-null  float16
 14  matchDuration    1934174 non-null  int16
 15  matchType        1934174 non-null  category
 16  maxPlace         1934174 non-null  int8
 17  numGroups        1934174 non-null  int8
 18  rankPoints       1934174 non-null  int16
 19  revives          1934174 non-null  int8
 20  rideDistance     1934174 non-null  float16
 21  roadKills        1934174 non-null  int8
 22  swimDistance     1934174 non-null  float16
 23  teamKills        1934174 non-null  int8
 24  vehicleDestroys  1934174 non-null  int8
 25  walkDistance     1934174 non-null  float16
 26  weaponsAcquired  1934174 non-null  int16
 27  winPoints        1934174 non-null  int16
dtypes: category(4), float16(5), int16(5), int8(14)
memory usage: 201.9 MB
```

In [10]:

```
train.winPlacePerc.value_counts().head()
```

Out[10]:

```
0.000000    220505
1.000000    127573
0.500000     55065
0.333252     42508
0.666504     38112
Name: winPlacePerc, dtype: int64
```

In [11]:

```
train.nunique()
```

Out[11]:

```
Id                4446966
groupId           2026745
matchId             47965
assists                20
boosts                 27
damageDealt         12925
DBNOs                  39
headshotKills          34
heals                  63
killPlace             101
killPoints           1707
kills                  58
killStreaks            18
longestKill         11155
matchDuration        1267
matchType              16
maxPlace              100
numGroups             100
rankPoints           2262
revives                25
rideDistance        15734
roadKills              14
swimDistance        12589
teamKills              11
vehicleDestroys         6
walkDistance        15940
weaponsAcquired        97
winPoints            1447
winPlacePerc         2268
dtype: int64
```

03. 데이터 전처리

처음으로 이동하기

결측치는 1개, 없애기

In [12]:

```
train.dropna(axis=0,inplace=True)
train.info(null_counts=True)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:2: FutureWarning: null_
counts is deprecated. Use show_counts instead


<class 'pandas.core.frame.DataFrame'>
Int64Index: 4446965 entries, 0 to 4446965
Data columns (total 29 columns):
 #   Column           Non-Null Count    Dtype   
---  ------           --------------    -----   
 0   Id               4446965 non-null  category
 1   groupId          4446965 non-null  category
 2   matchId          4446965 non-null  category
 3   assists          4446965 non-null  int8    
 4   boosts           4446965 non-null  int8    
 5   damageDealt      4446965 non-null  float16 
 6   DBNOs            4446965 non-null  int8    
 7   headshotKills    4446965 non-null  int8    
 8   heals            4446965 non-null  int8    
 9   killPlace        4446965 non-null  int8    
 10  killPoints       4446965 non-null  int16   
 11  kills            4446965 non-null  int8    
 12  killStreaks      4446965 non-null  int8    
 13  longestKill      4446965 non-null  float16 
 14  matchDuration    4446965 non-null  int16   
 15  matchType        4446965 non-null  category
 16  maxPlace         4446965 non-null  int8    
 17  numGroups        4446965 non-null  int8    
 18  rankPoints       4446965 non-null  int16   
 19  revives          4446965 non-null  int8    
 20  rideDistance     4446965 non-null  float16 
 21  roadKills        4446965 non-null  int8    
 22  swimDistance     4446965 non-null  float16 
 23  teamKills        4446965 non-null  int8    
 24  vehicleDestroys  4446965 non-null  int8    
 25  walkDistance     4446965 non-null  float16 
 26  weaponsAcquired  4446965 non-null  int16   
 27  winPoints        4446965 non-null  int16   
 28  winPlacePerc     4446965 non-null  float16 
dtypes: category(4), float16(6), int16(5), int8(14)
memory usage: 486.0 MB
```

In [13]:

```
train.corr()
```

Out[13]:

	assists	boosts	damageDealt	DBNOs	headshotKills	heals	killP
assists	1.000000	0.307683	0.406726	0.301057	0.198289	0.228556	-0.29(
boosts	0.307683	1.000000	0.521947	0.358907	0.334661	0.535854	-0.55
damageDealt	0.406726	0.521947	1.000000	0.735762	0.613409	0.342987	-0.67
DBNOs	0.301057	0.358907	0.735762	1.000000	0.469923	0.265485	-0.55
headshotKills	0.198289	0.334661	0.613409	0.469923	1.000000	0.199917	-0.46
heals	0.228556	0.535854	0.342987	0.265485	0.199917	1.000000	-0.38(
killPlace	-0.290062	-0.554844	-0.677398	-0.555290	-0.469829	-0.386032	1.00(
killPoints	0.039066	0.008347	0.049904	0.042616	0.023988	-0.002585	-0.02
kills	0.319690	0.502024	0.888784	0.707848	0.674275	0.311781	-0.73(
killStreaks	0.243471	0.405327	0.703599	0.646872	0.512882	0.270412	-0.81(
longestKill	0.261426	0.423291	0.563838	0.451422	0.447261	0.263278	-0.54
matchDuration	-0.019450	0.072107	-0.006755	-0.014486	-0.017657	0.108901	-0.00
maxPlace	-0.147916	-0.013686	-0.040708	-0.267710	0.009211	-0.064759	0.01(
numGroups	-0.146805	-0.012929	-0.040078	-0.265789	0.009389	-0.064204	0.01
rankPoints	-0.016407	0.023202	-0.001459	-0.003451	0.005119	0.019142	-0.01
revives	0.198320	0.253125	0.256951	0.300999	0.150145	0.236680	-0.26
rideDistance	0.110644	0.328855	0.140883	0.102536	0.076494	0.297484	-0.23(
roadKills	0.011903	0.035124	0.052487	0.036093	0.013369	0.024619	-0.05
swimDistance	0.023372	0.107992	0.036729	0.017320	0.028531	0.079586	-0.08
teamKills	0.006081	0.013068	0.015468	0.069430	0.008658	0.035386	-0.03
vehicleDestroys	0.057921	0.087387	0.081182	0.060210	0.039200	0.062870	-0.07
walkDistance	0.290305	0.640150	0.398199	0.284660	0.250982	0.430266	-0.59(
weaponsAcquired	0.243882	0.406609	0.353249	0.218161	0.217552	0.309071	-0.49(
winPoints	0.024014	-0.007645	0.017762	0.011108	0.004982	-0.010990	-0.00
winPlacePerc	0.299439	0.634232	0.440506	0.279968	0.277722	0.427856	-0.71

25 rows × 25 columns

In [14]:

```
train.corr()['winPlacePerc'].sort_values()
```

Out[14]:

```
killPlace         -0.719069
matchDuration     -0.005171
winPoints          0.007062
killPoints         0.012909
rankPoints         0.013522
teamKills          0.015942
roadKills          0.034544
maxPlace           0.037381
numGroups          0.039625
vehicleDestroys    0.073436
swimDistance       0.149606
revives            0.240880
headshotKills      0.277722
DBNOs              0.279968
assists            0.299439
rideDistance       0.342914
killStreaks        0.377566
longestKill        0.410153
kills              0.419915
heals              0.427856
damageDealt        0.440506
weaponsAcquired    0.583806
boosts             0.634232
walkDistance       0.810886
winPlacePerc       1.000000
Name: winPlacePerc, dtype: float64
```

In [15]:

```
### 가장 상관관계 높은 2개의 변수를 선택
train[ ['killPlace', 'walkDistance']].head()
```

Out[15]:

	killPlace	walkDistance
0	60	244.75
1	57	1434.00
2	47	161.75
3	75	202.75
4	45	49.75

In [16]:

```
sel = ['killPlace', 'walkDistance']
X_tr = train[sel]
y_tr = train['winPlacePerc']

X_last_test = test[sel]
```

In [17]:

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [18]:

```
X_train, X_test, y_train, y_test = train_test_split(X_tr, y_tr, test_size=0.2, random_state=77)

X_train.shape, X_test.shape
```

Out[18]:

```
((3557572, 2), (889393, 2))
```

04. 모델 생성 및 학습, 예측

처음으로 이동하기

In [19]:

```
lr = LinearRegression()
lr.fit(X_train, y_train)

print("학습용 score() - 결정계수 : ", lr.score(X_train, y_train))
print("테스트용 score() - 결정계수 : ", lr.score(X_test, y_test))
```

```
학습용 score() - 결정계수 :  0.7464619534577828
테스트용 score() - 결정계수 :  0.7455769968673939
```

In [20]:

```
%%time

cvs_lr = cross_val_score(lr, X_test, y_test, cv=5, scoring="neg_median_absolute_error")
cvs_lr.mean(), cvs_lr.std()
```

```
CPU times: user 1.35 s, sys: 641 ms, total: 1.99 s
Wall time: 1.03 s
```

Out[20]:

```
(-0.09620575904846192, 0.00020372508052395657)
```

예측

In [21]:

```
df_sub.head()
```

Out[21]:

	Id	winPlacePerc
0	9329eb41e215eb	1
1	639bd0dcd7bda8	1
2	63d5c8ef8dfe91	1
3	cf5b81422591d1	1
4	ee6a295187ba21	1

In [22]:

```
test.head()
```

Out[22]:

	Id	groupId	matchId	assists	boosts	damageDealt	DBNOs	h
0	9329eb41e215eb	676b23c24e70d6	45b576ab7daa7f	0	0	51.46875	0	
1	639bd0dcd7bda8	430933124148dd	42a9a0b906c928	0	4	179.12500	0	
2	63d5c8ef8dfe91	0b45f5db20ba99	87e7e4477a048e	1	0	23.40625	0	
3	cf5b81422591d1	b7497dbdc77f4a	1b9a94f1af67f1	0	0	65.50000	0	
4	ee6a295187ba21	6604ce20a1d230	40754a93016066	0	4	330.25000	1	

5 rows × 28 columns

In [23]:

```
pred = lr.predict(X_last_test)
df_sub['winPlacePerc'] = pred
df_sub.to_csv("submission.csv", index=False)
```

- 참고 Notebook
  - https://www.kaggle.com/code/eryash15/pubg-simplest-model (https://www.kaggle.com/code/eryash15/pubg-simplest-model)
- 제출 결과
  - LeaderBoard 1393/1528