ejrzrvmcn

July 28, 2023

[20]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[21]:
```python
df=pd.read_csv("/content/15_Horse Racing Results.csv")
df
```

[21]:
```
             Dato    Track  Race Number  Distance Surface  Prize money  \
0      03.09.2017  Sha Tin           10      1400   Gress      1310000
1      16.09.2017  Sha Tin           10      1400   Gress      1310000
2      14.10.2017  Sha Tin           10      1400   Gress      1310000
3      11.11.2017  Sha Tin            9      1600   Gress      1310000
4      26.11.2017  Sha Tin            9      1600   Gress      1310000
…             …        …            …         …       …            …
27003  14.06.2020  Sha Tin           11      1200   Gress      1450000
27004  21.06.2020  Sha Tin            2      1200   Gress       967000
27005  21.06.2020  Sha Tin            4      1200   Gress       967000
27006  21.06.2020  Sha Tin            5      1200   Gress       967000
27007  21.06.2020  Sha Tin           11      1200   Gress      1450000

       Starting position         Jockey  Jockey weight      Country  …  \
0                      6      K C Leung             52      Sverige  …
1                     14         C Y Ho             52      Sverige  …
2                      8         C Y Ho             52      Sverige  …
3                     13  Brett Prebble             54      Sverige  …
4                      9         C Y Ho             52      Sverige  …
…                    …              …              …            …  …
27003                  6      A Hamelin             59    Australia  …
27004                  7      K C Leung             57    Australia  …
27005                  6    Blake Shinn             57    Australia  …
27006                 14   Joao Moreira             57  New Zealand  …
27007                  7    C Schofield             55  New Zealand  …

        TrainerName Race time Path  Final place  FGrating  Odds  RaceType  \
0            CH Yip     83,38    2            9       110    22  Handicap
1            CH Yip     81,56    3            4       124    48  Handicap
```

```
2            CH Yip     82,36    1            6       118     11  Handicap
3            CH Yip     96,53    0            8       107     11  Handicap
4            CH Yip     94,17    0            3       123     40  Handicap
…              …      …  …            …        …   …       …
27003         WY So     70,87    1            9       104     25  Handicap
27004        KL Man     69,91    2            5       110    124  Handicap
27005  P O'Sullivan     69,49    0            3       114     88  Handicap
27006       AS Cruz     70,08    2            7       109     22  Handicap
27007         WY So     69,51    2            9       118     55  Handicap

      HorseId  JockeyId  TrainerID
0        1736      8656       6687
1        1736      8659       6687
2        1736      8659       6687
3        1736      8453       6687
4        1736      8659       6687
…        …         …         …
27003   29038      9111       6683
27004   29056      8656       6693
27005   29057      8778       6691
27006   29058      8443       6684
27007   29059      8655       6683

[27008 rows x 21 columns]
```

[22]:
```
df.head()
```

[22]:
```
         Dato    Track  Race Number  Distance Surface  Prize money  \
0  03.09.2017  Sha Tin           10      1400   Gress      1310000
1  16.09.2017  Sha Tin           10      1400   Gress      1310000
2  14.10.2017  Sha Tin           10      1400   Gress      1310000
3  11.11.2017  Sha Tin            9      1600   Gress      1310000
4  26.11.2017  Sha Tin            9      1600   Gress      1310000

  Starting position        Jockey  Jockey weight  Country  …  TrainerName  \
0                 6      K C Leung             52  Sverige  …       CH Yip
1                14         C Y Ho             52  Sverige  …       CH Yip
2                 8         C Y Ho             52  Sverige  …       CH Yip
3                13  Brett Prebble             54  Sverige  …       CH Yip
4                 9         C Y Ho             52  Sverige  …       CH Yip

  Race time Path  Final place  FGrating  Odds  RaceType HorseId  JockeyId  \
0     83,38    2            9       110    22  Handicap    1736      8656
1     81,56    3            4       124    48  Handicap    1736      8659
2     82,36    1            6       118    11  Handicap    1736      8659
3     96,53    0            8       107    11  Handicap    1736      8453
4     94,17    0            3       123    40  Handicap    1736      8659
```

```
   TrainerID
0       6687
1       6687
2       6687
3       6687
4       6687

[5 rows x 21 columns]
```

1 DATA CLEANING AND DATA PREPROCESSING

[23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27008 entries, 0 to 27007
Data columns (total 21 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Dato               27008 non-null  object
 1   Track              27008 non-null  object
 2   Race Number        27008 non-null  int64
 3   Distance           27008 non-null  int64
 4   Surface            27008 non-null  object
 5   Prize money        27008 non-null  int64
 6   Starting position  27008 non-null  int64
 7   Jockey             27008 non-null  object
 8   Jockey weight      27008 non-null  int64
 9   Country            27008 non-null  object
 10  Horse age          27008 non-null  int64
 11  TrainerName        27008 non-null  object
 12  Race time          27008 non-null  object
 13  Path               27008 non-null  int64
 14  Final place        27008 non-null  int64
 15  FGrating           27008 non-null  int64
 16  Odds               27008 non-null  object
 17  RaceType           27008 non-null  object
 18  HorseId            27008 non-null  int64
 19  JockeyId           27008 non-null  int64
 20  TrainerID          27008 non-null  int64
dtypes: int64(12), object(9)
memory usage: 4.3+ MB
```

[24]:
```
df.describe()
```

[24]:
```
        Race Number      Distance   Prize money  Starting position  \
count  27008.000000  27008.000000  2.700800e+04       27008.000000
mean       5.268624   1401.666173  1.479445e+06           6.741447
std        2.780088    276.065045  2.162109e+06           3.691071
min        1.000000   1000.000000  6.600000e+05           1.000000
25%        3.000000   1200.000000  9.200000e+05           4.000000
50%        5.000000   1400.000000  9.670000e+05           7.000000
75%        8.000000   1650.000000  1.450000e+06          10.000000
max       11.000000   2400.000000  2.800000e+07          14.000000

       Jockey weight     Horse age          Path   Final place      FGrating  \
count   27008.000000  27008.000000  27008.000000  27008.000000  27008.000000
mean       55.867373      5.246408      1.678021      6.685834    113.428318
std         2.737006      1.519880      1.631784      3.664551     13.314949
min        47.000000      2.000000      0.000000      1.000000     -5.000000
25%        54.000000      4.000000      0.000000      4.000000    106.000000
50%        56.000000      5.000000      1.000000      7.000000    114.000000
75%        58.000000      6.000000      3.000000     10.000000    122.000000
max        63.000000     12.000000     11.000000     14.000000    157.000000

            HorseId      JockeyId     TrainerID
count  27008.000000  27008.000000  27008.000000
mean   23904.874889   8586.732042   6668.559205
std     2028.860311    569.616932     79.978067
min     1736.000000    227.000000   6431.000000
25%    22364.000000   8651.000000   6683.000000
50%    22868.500000   8658.000000   6687.000000
75%    25310.000000   8663.000000   6693.000000
max    29059.000000   9113.000000   7004.000000
```

[25]:
```python
df.columns
```

[25]:
```
Index(['Dato', 'Track', 'Race Number', 'Distance', 'Surface', 'Prize money',
       'Starting position', 'Jockey', 'Jockey weight', 'Country', 'Horse age',
       'TrainerName', 'Race time', 'Path', 'Final place', 'FGrating', 'Odds',
       'RaceType', 'HorseId', 'JockeyId', 'TrainerID'],
      dtype='object')
```

[26]:
```python
df1=df.dropna(axis=1)
df1
```

[26]:
```
             Dato    Track  Race Number  Distance Surface  Prize money  \
0      03.09.2017  Sha Tin           10      1400   Gress      1310000
1      16.09.2017  Sha Tin           10      1400   Gress      1310000
2      14.10.2017  Sha Tin           10      1400   Gress      1310000
3      11.11.2017  Sha Tin            9      1600   Gress      1310000
4      26.11.2017  Sha Tin            9      1600   Gress      1310000
```

```
…           …      …          …       …      …          …
27003  14.06.2020  Sha Tin          11      1200   Gress      1450000
27004  21.06.2020  Sha Tin           2      1200   Gress       967000
27005  21.06.2020  Sha Tin           4      1200   Gress       967000
27006  21.06.2020  Sha Tin           5      1200   Gress       967000
27007  21.06.2020  Sha Tin          11      1200   Gress      1450000

       Starting position        Jockey  Jockey weight      Country  …  \
0                      6      K C Leung             52      Sverige  …
1                     14         C Y Ho             52      Sverige  …
2                      8         C Y Ho             52      Sverige  …
3                     13  Brett Prebble             54      Sverige  …
4                      9         C Y Ho             52      Sverige  …
…                    …             …             …            …  …
27003                  6      A Hamelin             59    Australia  …
27004                  7      K C Leung             57    Australia  …
27005                  6    Blake Shinn             57    Australia  …
27006                 14   Joao Moreira             57  New Zealand  …
27007                  7    C Schofield             55  New Zealand  …

        TrainerName Race time Path  Final place  FGrating  Odds  RaceType  \
0            CH Yip     83,38    2            9       110    22  Handicap
1            CH Yip     81,56    3            4       124    48  Handicap
2            CH Yip     82,36    1            6       118    11  Handicap
3            CH Yip     96,53    0            8       107    11  Handicap
4            CH Yip     94,17    0            3       123    40  Handicap
…              …        …   …          …        …    …         …
27003         WY So     70,87    1            9       104    25  Handicap
27004        KL Man     69,91    2            5       110   124  Handicap
27005  P O'Sullivan     69,49    0            3       114    88  Handicap
27006       AS Cruz     70,08    2            7       109    22  Handicap
27007         WY So     69,51    2            9       118    55  Handicap

      HorseId  JockeyId  TrainerID
0        1736      8656       6687
1        1736      8659       6687
2        1736      8659       6687
3        1736      8453       6687
4        1736      8659       6687
…         …        …         …
27003   29038      9111       6683
27004   29056      8656       6693
27005   29057      8778       6691
27006   29058      8443       6684
27007   29059      8655       6683

[27008 rows x 21 columns]
```

```
[27]: df1.columns
```

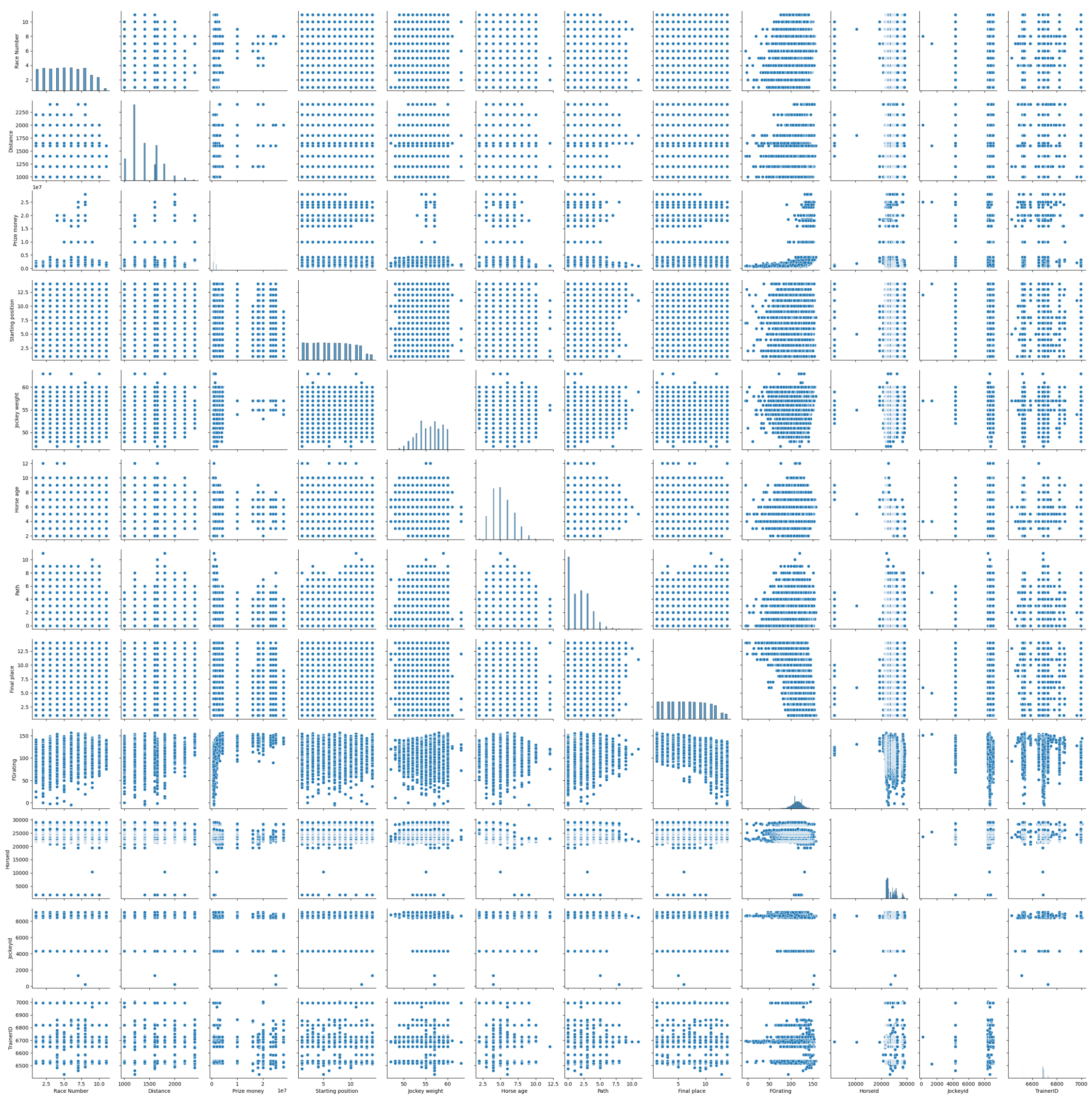
```
[27]: Index(['Dato', 'Track', 'Race Number', 'Distance', 'Surface', 'Prize money',
             'Starting position', 'Jockey', 'Jockey weight', 'Country', 'Horse age',
             'TrainerName', 'Race time', 'Path', 'Final place', 'FGrating', 'Odds',
             'RaceType', 'HorseId', 'JockeyId', 'TrainerID'],
            dtype='object')
```

```
[28]: df1=df1[['Dato', 'Track', 'Race Number', 'Distance', 'Surface', 'Prize money',
             'Starting position', 'Jockey', 'Jockey weight', 'Country', 'Horse age',
             'TrainerName', 'Race time', 'Path', 'Final place', 'FGrating', 'Odds',
             'RaceType', 'HorseId', 'JockeyId', 'TrainerID']]
```

2 EDA AND VISUALIZATION

```
[29]: sns.pairplot(df1)
```

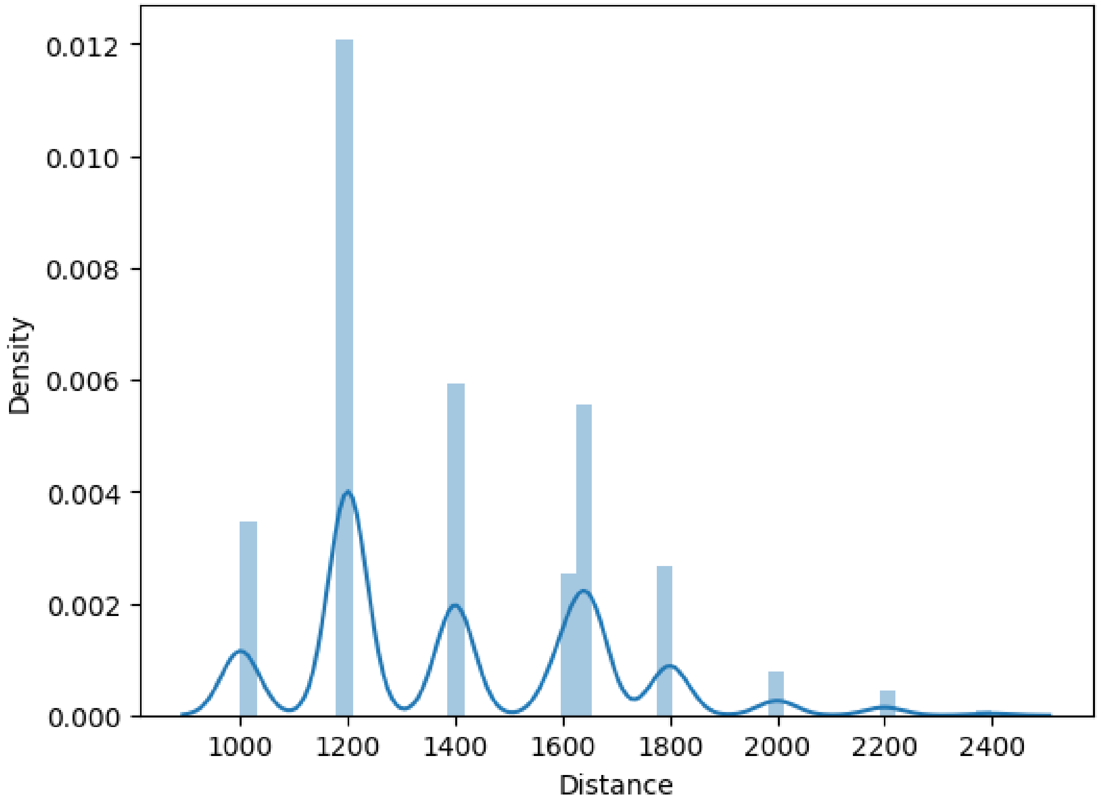
```
[29]: <seaborn.axisgrid.PairGrid at 0x7ea8e39bbb80>
```



```
[30]: sns.distplot(df1['Distance'])
```

```
<ipython-input-30-55baf5480dab>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df1['Distance'])
```

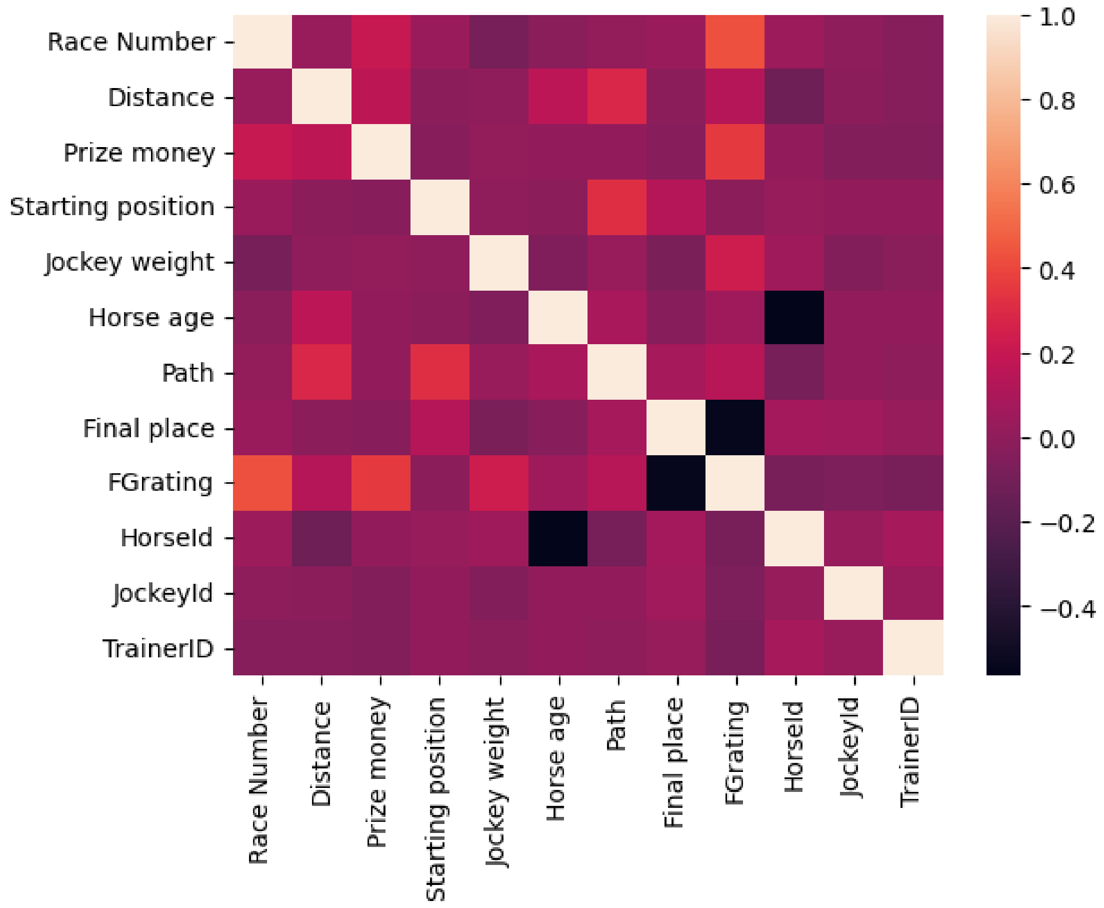
```
[30]: <Axes: xlabel='Distance', ylabel='Density'>
```



```
[31]: sns.heatmap(df1.corr())
```

```
<ipython-input-31-3ed1a1a51dc0>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df1.corr())
```

```
[31]: <Axes: >
```



3 TO TRAIN THE MODEL AND MODEL BULDING

[32]:
```
x=df[['Race Number', 'Prize money',
        'Starting position',  'Jockey weight',  'Horse age', 'Final place',␣
  ↪'FGrating',  'HorseId', 'JockeyId', 'TrainerID']]
y=df['Distance']
```

[33]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

[34]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

[34]:
```
LinearRegression()
```

[35]:
```
lr.intercept_
```

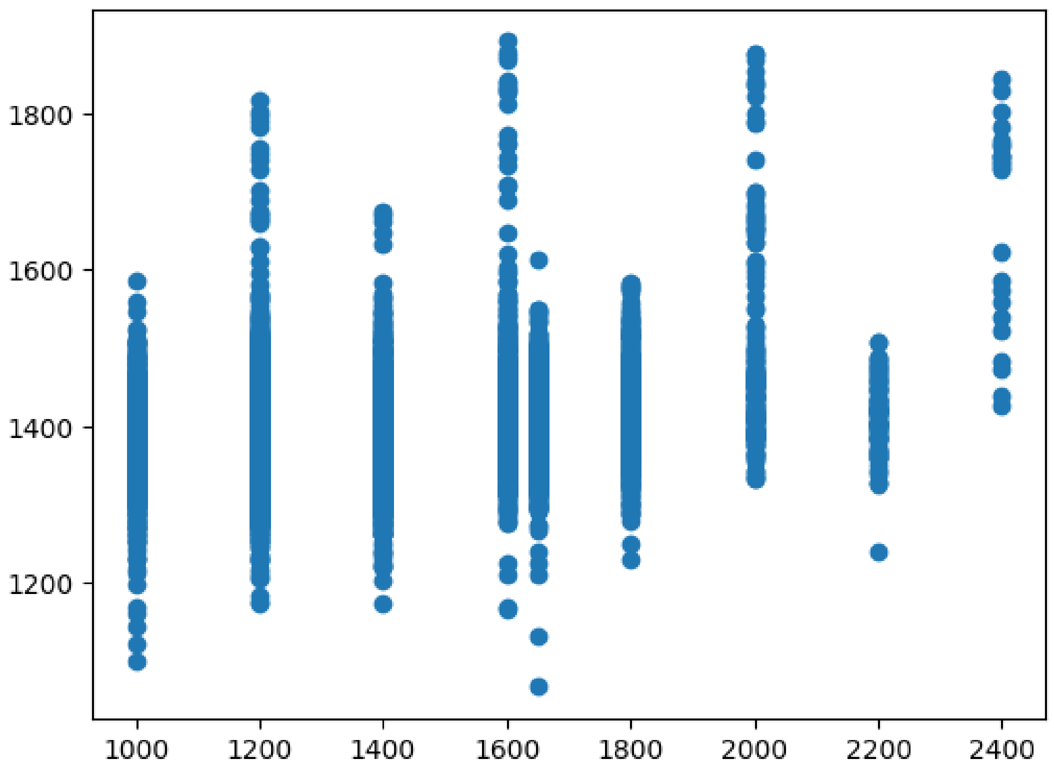
[35]: 1702.2246084652907

[36]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

[36]:
```
                   Co-efficient
Race Number           -6.066791
Prize money            0.000015
Starting position     -0.576600
Jockey weight         -1.939342
Horse age             22.775448
Final place            6.388438
FGrating               3.067410
HorseId               -0.006693
JockeyId              -0.006062
TrainerID             -0.071689
```

[37]:
```
prediction =lr.predict(x_test)
plt.scatter(y_test,prediction)
```

[37]:
```
<matplotlib.collections.PathCollection at 0x7ea8d2eb3ee0>
```



4 ACCURACY

```
[38]: lr.score(x_test,y_test)
```

```
[38]: 0.06099752116422741
```

```
[39]: lr.score(x_train,y_train)
```

```
[39]: 0.06411986650423529
```

```
[40]: from sklearn.linear_model import Ridge,Lasso
```

```
[41]: rr=Ridge(alpha=10)
      rr.fit(x_train,y_train)
```

```
[41]: Ridge(alpha=10)
```

```
[42]: rr.score(x_test,y_test)
```

```
[42]: 0.06099829608026752
```

```
[43]: rr.score(x_train,y_train)
```

```
[43]: 0.06411986523298252
```

```
[44]: la=Lasso(alpha=10)
      la.fit(x_train,y_train)
```

```
[44]: Lasso(alpha=10)
```

```
[45]: la.score(x_test,y_test)
```

```
[45]: 0.05982733108440974
```

```
[46]: la.score(x_train,y_train)
```

```
[46]: 0.06224170620823566
```