PlotEmAll

June 25, 2020

[4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[7]:
```
#Loading Dataset for Building the Model
data = pd.read_csv('datasets_2756_4568_pokemon.csv')
data.head(16)
```

[7]:
```
                                  abilities  against_bug  against_dark  \
0               ['Overgrow', 'Chlorophyll']         1.00           1.0
1               ['Overgrow', 'Chlorophyll']         1.00           1.0
2               ['Overgrow', 'Chlorophyll']         1.00           1.0
3                  ['Blaze', 'Solar Power']         0.50           1.0
4                  ['Blaze', 'Solar Power']         0.50           1.0
5                  ['Blaze', 'Solar Power']         0.25           1.0
6                  ['Torrent', 'Rain Dish']         1.00           1.0
7                  ['Torrent', 'Rain Dish']         1.00           1.0
8                  ['Torrent', 'Rain Dish']         1.00           1.0
9               ['Shield Dust', 'Run Away']         1.00           1.0
10                            ['Shed Skin']         1.00           1.0
11          ['Compoundeyes', 'Tinted Lens']         0.50           1.0
12              ['Shield Dust', 'Run Away']         0.50           1.0
13                            ['Shed Skin']         0.50           1.0
14                     ['Swarm', 'Sniper']         0.50           1.0
15  ['Keen Eye', 'Tangled Feet', 'Big Pecks']       0.50           1.0

    against_dragon  against_electric  against_fairy  against_fight  \
0              1.0               0.5            0.5           0.50
1              1.0               0.5            0.5           0.50
2              1.0               0.5            0.5           0.50
3              1.0               1.0            0.5           1.00
4              1.0               1.0            0.5           1.00
5              1.0               2.0            0.5           0.50
6              1.0               2.0            1.0           1.00
7              1.0               2.0            1.0           1.00
8              1.0               2.0            1.0           1.00
```

```
9            1.0             1.0            1.0            0.50
10           1.0             1.0            1.0            0.50
11           1.0             2.0            1.0            0.25
12           1.0             1.0            0.5            0.25
13           1.0             1.0            0.5            0.25
14           1.0             1.0            0.5            0.25
15           1.0             2.0            1.0            1.00

   against_fire  against_flying  against_ghost  …  percentage_male  \
0           2.0             2.0            1.0  …             88.1
1           2.0             2.0            1.0  …             88.1
2           2.0             2.0            1.0  …             88.1
3           0.5             1.0            1.0  …             88.1
4           0.5             1.0            1.0  …             88.1
5           0.5             1.0            1.0  …             88.1
6           0.5             1.0            1.0  …             88.1
7           0.5             1.0            1.0  …             88.1
8           0.5             1.0            1.0  …             88.1
9           2.0             2.0            1.0  …             50.0
10          2.0             2.0            1.0  …             50.0
11          2.0             2.0            1.0  …             50.0
12          2.0             2.0            1.0  …             50.0
13          2.0             2.0            1.0  …             50.0
14          2.0             2.0            1.0  …             50.0
15          1.0             1.0            0.0  …             50.0

   pokedex_number  sp_attack  sp_defense  speed   type1   type2  weight_kg  \
0               1         65          65     45   grass  poison        6.9
1               2         80          80     60   grass  poison       13.0
2               3        122         120     80   grass  poison      100.0
3               4         60          50     65    fire     NaN        8.5
4               5         80          65     80    fire     NaN       19.0
5               6        159         115    100    fire  flying       90.5
6               7         50          64     43   water     NaN        9.0
7               8         65          80     58   water     NaN       22.5
8               9        135         115     78   water     NaN       85.5
9              10         20          20     45     bug     NaN        2.9
10             11         25          25     30     bug     NaN        9.9
11             12         90          80     70     bug  flying       32.0
12             13         20          20     50     bug  poison        3.2
13             14         25          25     35     bug  poison       10.0
14             15         15          80    145     bug  poison       29.5
15             16         35          35     56  normal  flying        1.8

   generation  is_legendary
0           1             0
1           1             0
```

```
2            1            0
3            1            0
4            1            0
5            1            0
6            1            0
7            1            0
8            1            0
9            1            0
10           1            0
11           1            0
12           1            0
13           1            0
14           1            0
15           1            0

[16 rows x 41 columns]
```

[9]:
```
#Checking the shape of the dataframe
data.shape
```

[9]:
```
(801, 41)
```

[10]:
```
#displaying all the coloumn names
data.columns
```

[10]:
```
Index(['abilities', 'against_bug', 'against_dark', 'against_dragon',
       'against_electric', 'against_fairy', 'against_fight', 'against_fire',
       'against_flying', 'against_ghost', 'against_grass', 'against_ground',
       'against_ice', 'against_normal', 'against_poison', 'against_psychic',
       'against_rock', 'against_steel', 'against_water', 'attack',
       'base_egg_steps', 'base_happiness', 'base_total', 'capture_rate',
       'classfication', 'defense', 'experience_growth', 'height_m', 'hp',
       'japanese_name', 'name', 'percentage_male', 'pokedex_number',
       'sp_attack', 'sp_defense', 'speed', 'type1', 'type2', 'weight_kg',
       'generation', 'is_legendary'],
      dtype='object')
```

[11]:
```
#Descriptive statistics using describe
data.describe()
```

[11]:
```
       against_bug  against_dark  against_dragon  against_electric  \
count   801.000000    801.000000      801.000000        801.000000
mean      0.996255      1.057116        0.968789          1.073970
std       0.597248      0.438142        0.353058          0.654962
min       0.250000      0.250000        0.000000          0.000000
25%       0.500000      1.000000        1.000000          0.500000
50%       1.000000      1.000000        1.000000          1.000000
```

```
75%       1.000000      1.000000        1.000000          1.000000
max       4.000000      4.000000        2.000000          4.000000

       against_fairy  against_fight  against_fire  against_flying  \
count     801.000000     801.000000    801.000000      801.000000
mean        1.068976       1.065543      1.135456        1.192884
std         0.522167       0.717251      0.691853        0.604488
min         0.250000       0.000000      0.250000        0.250000
25%         1.000000       0.500000      0.500000        1.000000
50%         1.000000       1.000000      1.000000        1.000000
75%         1.000000       1.000000      2.000000        1.000000
max         4.000000       4.000000      4.000000        4.000000

       against_ghost  against_grass  …    height_m          hp  \
count     801.000000     801.000000  …  781.000000  801.000000
mean        0.985019       1.034020  …    1.163892   68.958801
std         0.558256       0.788896  …    1.080326   26.576015
min         0.000000       0.250000  …    0.100000    1.000000
25%         1.000000       0.500000  …    0.600000   50.000000
50%         1.000000       1.000000  …    1.000000   65.000000
75%         1.000000       1.000000  …    1.500000   80.000000
max         4.000000       4.000000  …   14.500000  255.000000

       percentage_male  pokedex_number   sp_attack  sp_defense       speed  \
count       703.000000      801.000000  801.000000  801.000000  801.000000
mean         55.155761      401.000000   71.305868   70.911361   66.334582
std          20.261623      231.373075   32.353826   27.942501   28.907662
min           0.000000        1.000000   10.000000   20.000000    5.000000
25%          50.000000      201.000000   45.000000   50.000000   45.000000
50%          50.000000      401.000000   65.000000   66.000000   65.000000
75%          50.000000      601.000000   91.000000   90.000000   85.000000
max         100.000000      801.000000  194.000000  230.000000  180.000000

        weight_kg  generation  is_legendary
count  781.000000  801.000000    801.000000
mean    61.378105    3.690387      0.087391
std    109.354766    1.930420      0.282583
min      0.100000    1.000000      0.000000
25%      9.000000    2.000000      0.000000
50%     27.300000    4.000000      0.000000
75%     64.800000    5.000000      0.000000
max    999.900000    7.000000      1.000000

[8 rows x 34 columns]
```

[12]:
```
data.dtypes
```

[12]:
```
abilities             object
against_bug          float64
against_dark         float64
against_dragon       float64
against_electric     float64
against_fairy        float64
against_fight        float64
against_fire         float64
against_flying       float64
against_ghost        float64
against_grass        float64
against_ground       float64
against_ice          float64
against_normal       float64
against_poison       float64
against_psychic      float64
against_rock         float64
against_steel        float64
against_water        float64
attack                 int64
base_egg_steps         int64
base_happiness         int64
base_total             int64
capture_rate          object
classfication         object
defense                int64
experience_growth      int64
height_m             float64
hp                     int64
japanese_name         object
name                  object
percentage_male      float64
pokedex_number         int64
sp_attack              int64
sp_defense             int64
speed                  int64
type1                 object
type2                 object
weight_kg            float64
generation             int64
is_legendary           int64
dtype: object
```

[14]:
```
missing_values_count = data.isnull().sum()
missing_values_count
```

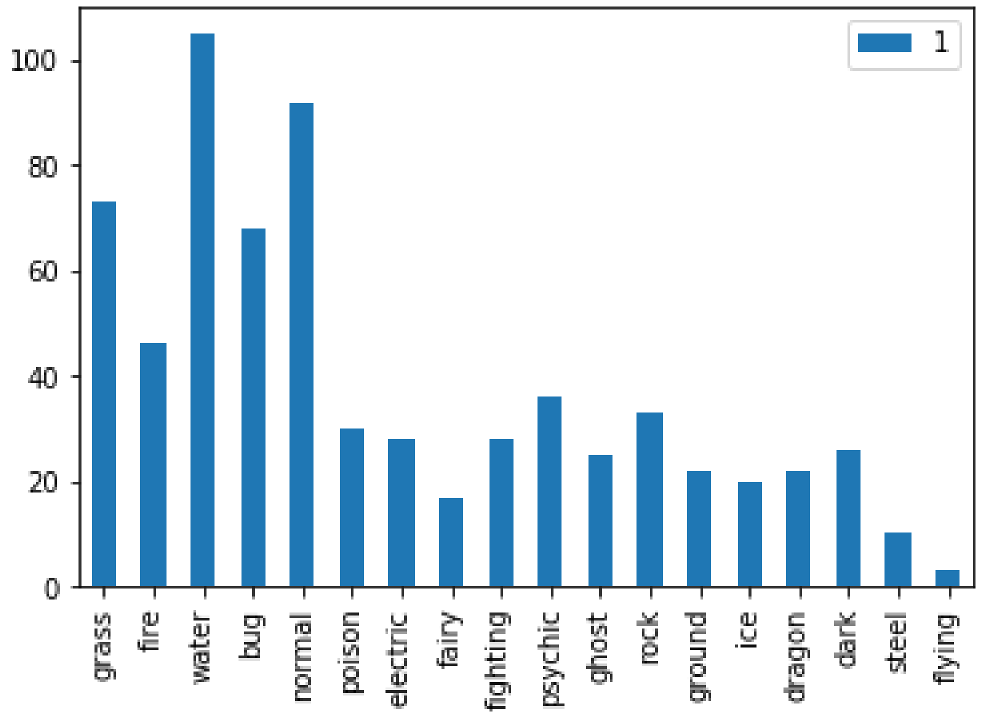
[14]:
```
abilities               0
against_bug             0
against_dark            0
against_dragon          0
against_electric        0
against_fairy           0
against_fight           0
against_fire            0
against_flying          0
against_ghost           0
against_grass           0
against_ground          0
against_ice             0
against_normal          0
against_poison          0
against_psychic         0
against_rock            0
against_steel           0
against_water           0
attack                  0
base_egg_steps          0
base_happiness          0
base_total              0
capture_rate            0
classfication           0
defense                 0
experience_growth       0
height_m               20
hp                      0
japanese_name           0
name                    0
percentage_male        98
pokedex_number          0
sp_attack               0
sp_defense              0
speed                   0
type1                   0
type2                 384
weight_kg              20
generation              0
is_legendary            0
dtype: int64
```

[32]:
```
#removing the data that has the null values.
data = data.dropna(subset=['percentage_male','weight_kg'])
data.shape
```

[32]: (684, 41)

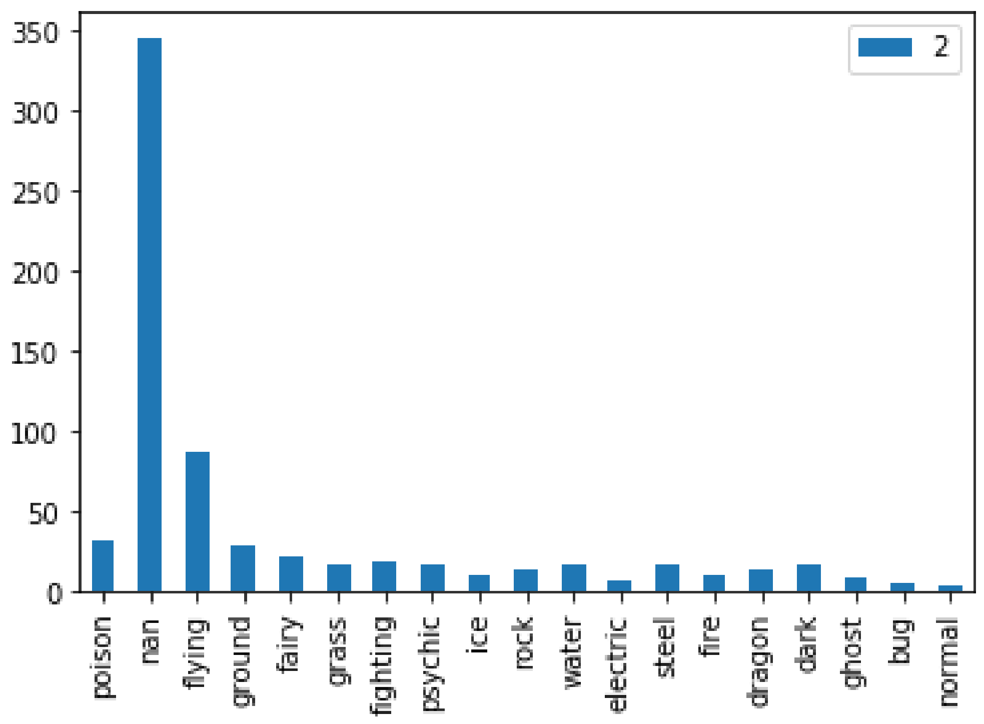
[33]:
```python
#Distribution of the types
from collections import Counter

type_counts = Counter(data['type1'])
type1 = pd.DataFrame.from_dict(type_counts, orient='index')
ax=type1.plot(kind='bar')
ax.legend('1')
```

[33]:
```
<matplotlib.legend.Legend at 0x2692a44d188>
```



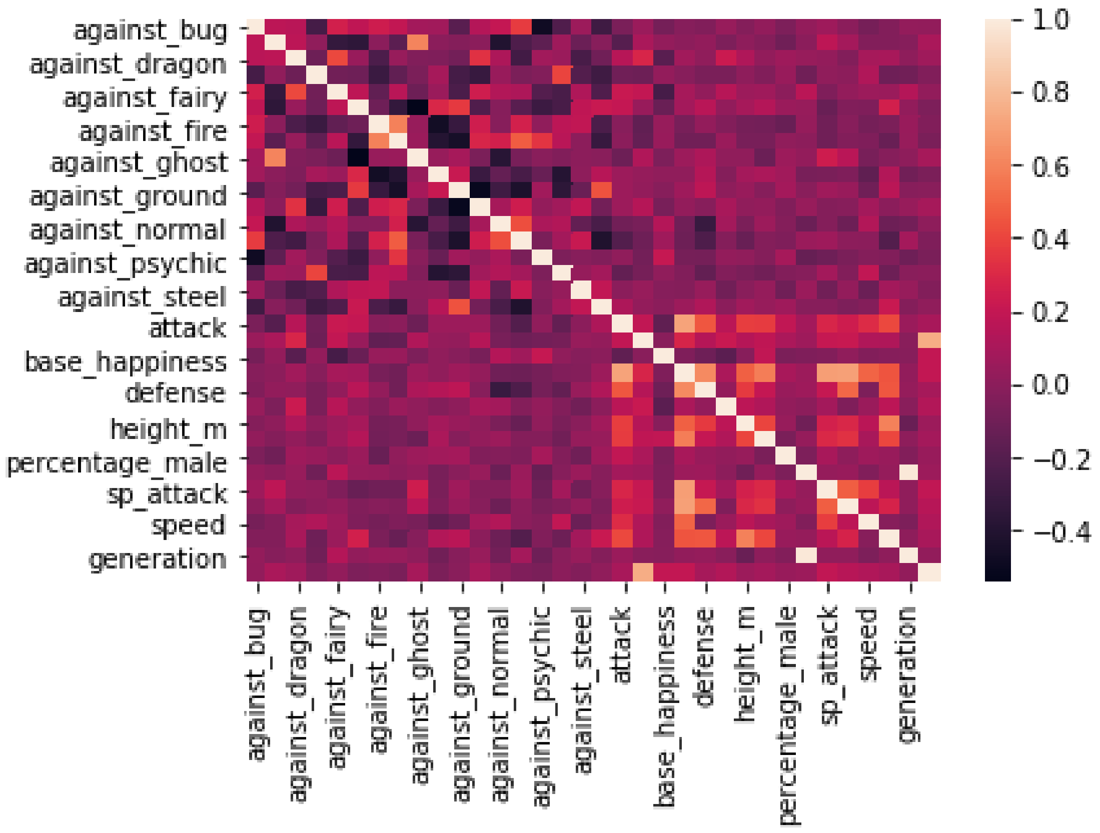
[34]:
```python
type_counts = Counter(data['type2'])
type1 = pd.DataFrame.from_dict(type_counts, orient='index')
ax=type1.plot(kind='bar')
ax.legend('2')
```

[34]:
```
<matplotlib.legend.Legend at 0x2692a51af48>
```

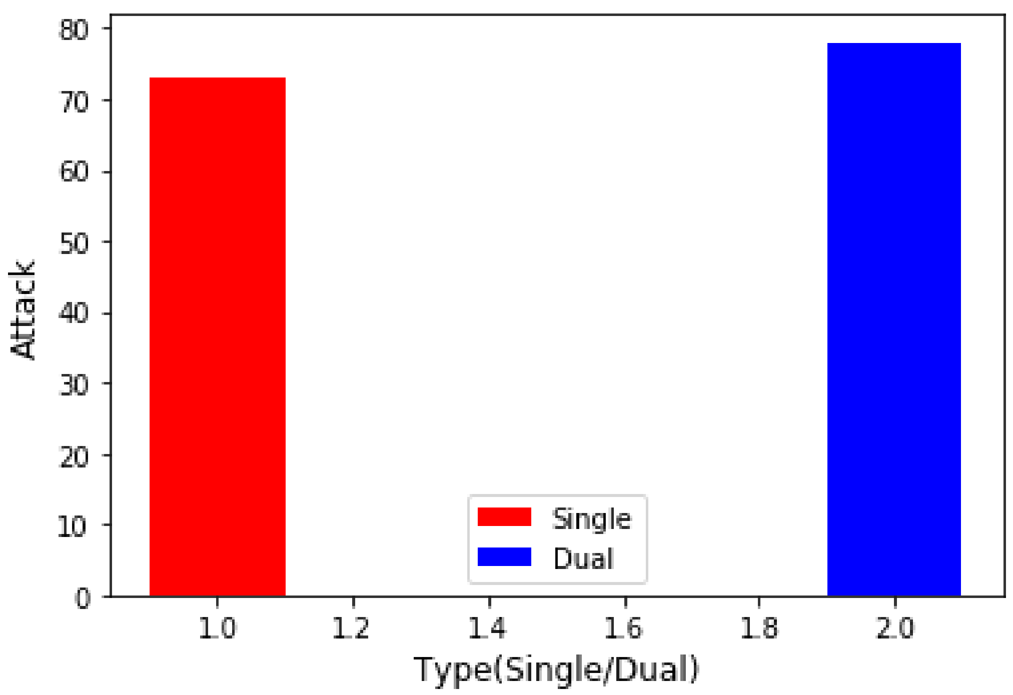


```
[35]: #correlation between the data by heat map
      corr = data.corr()
      sns.heatmap(corr)
```

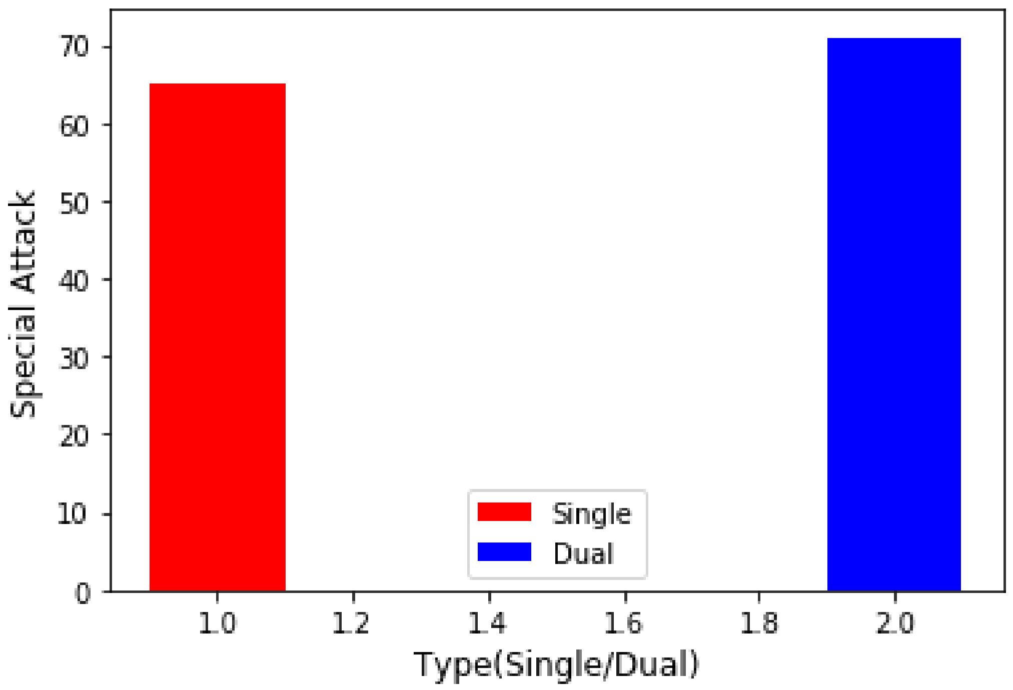
```
[35]: <matplotlib.axes._subplots.AxesSubplot at 0x2692a61d7c8>
```



[36]:
```
#Differentiating single vs dual types using attack and special attack␣
↪atttributes
df = data.rename(columns={'type1': 'Type_1', 'type2': 'Type_2'})
df.fillna(value='missing', axis=1, inplace=True)
single = df[df['Type_2'].str.contains('missing')]
dual = df[~df['Type_2'].str.contains('missing')]
atk_single = round(np.sum(single['attack'].values, axis = 0) / single.shape[0])
spatk_single = round(np.sum(single['sp_attack'].values, axis = 0) / single.
↪shape[0])
atk_dual = round(np.sum(dual['attack'].values, axis = 0) / dual.shape[0])
spatk_dual = round(np.sum(dual['sp_attack'].values, axis = 0) / dual.shape[0])
x = np.array([1,2])
y = np.array([atk_single,atk_dual])
plt.bar(x[0],y[0],color='r',label = 'Single',width = 0.2)
plt.bar(x[1],y[1],color='b', label = 'Dual',width = 0.2)
plt.xlabel("Type(Single/Dual)",fontsize = 12)
plt.ylabel("Attack",fontsize = 12)
plt.legend()
plt.show()
```

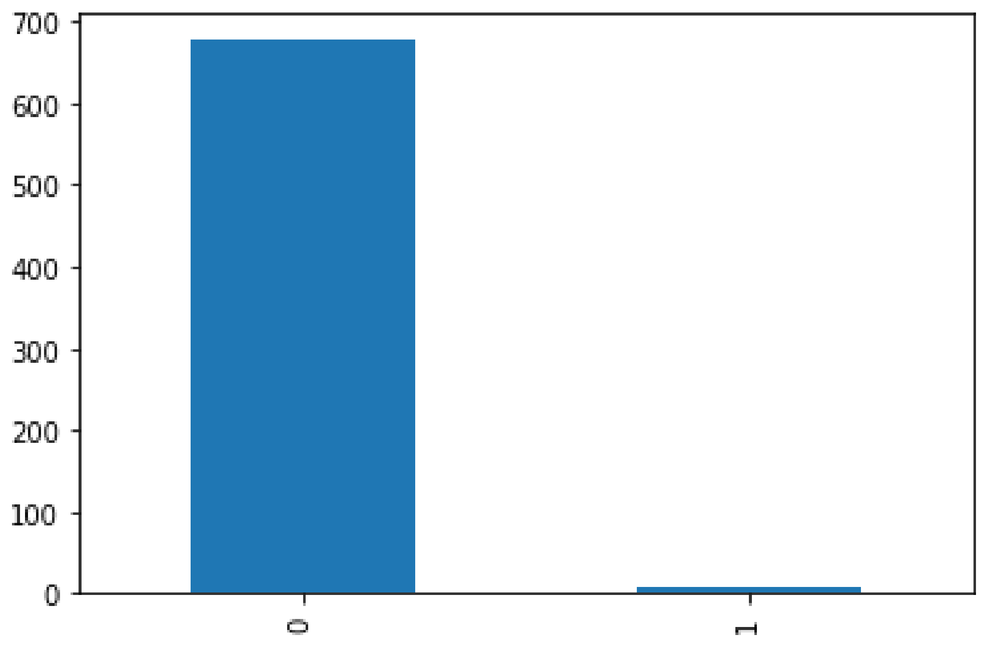


[37]:
```
x = np.array([1,2])
y = np.array([spatk_single,spatk_dual])
plt.bar(x[0],y[0],color='r',label = 'Single',width = 0.2)
plt.bar(x[1],y[1],color='b', label = 'Dual',width = 0.2)
plt.xlabel("Type(Single/Dual)",fontsize = 12)
plt.ylabel("Special Attack",fontsize = 12)
plt.legend()
plt.show()
```



```
[38]: #Finding out the count of legendary and non legendary pokemons
      data['is_legendary'].value_counts().plot.bar()
```

```
[38]: <matplotlib.axes._subplots.AxesSubplot at 0x2692a72d748>
```



```
[ ]:
```