```python
import numpy as np
import pandas as pd

train=pd.read_csv("train.csv")
train
```

```
        timestamp  value  is_anomaly  predicted
0      1425008573     42       False  44.072500
1      1425008873     41       False  50.709390
2      1425009173     41       False  81.405120
3      1425009473     61       False  39.950367
4      1425009773     44       False  35.350160
...           ...    ...         ...        ...
15825  1429756073     44       False  53.624115
15826  1429756373     45       False  59.752296
15827  1429756673     48       False  52.147630
15828  1429756973     26       False  58.007545
15829  1429757273     38       False  59.144700

[15830 rows x 4 columns]
```

```python
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15830 entries, 0 to 15829
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   timestamp   15830 non-null  int64
 1   value       15830 non-null  int64
 2   is_anomaly  15830 non-null  bool
 3   predicted   15830 non-null  float64
dtypes: bool(1), float64(1), int64(2)
memory usage: 386.6 KB
```

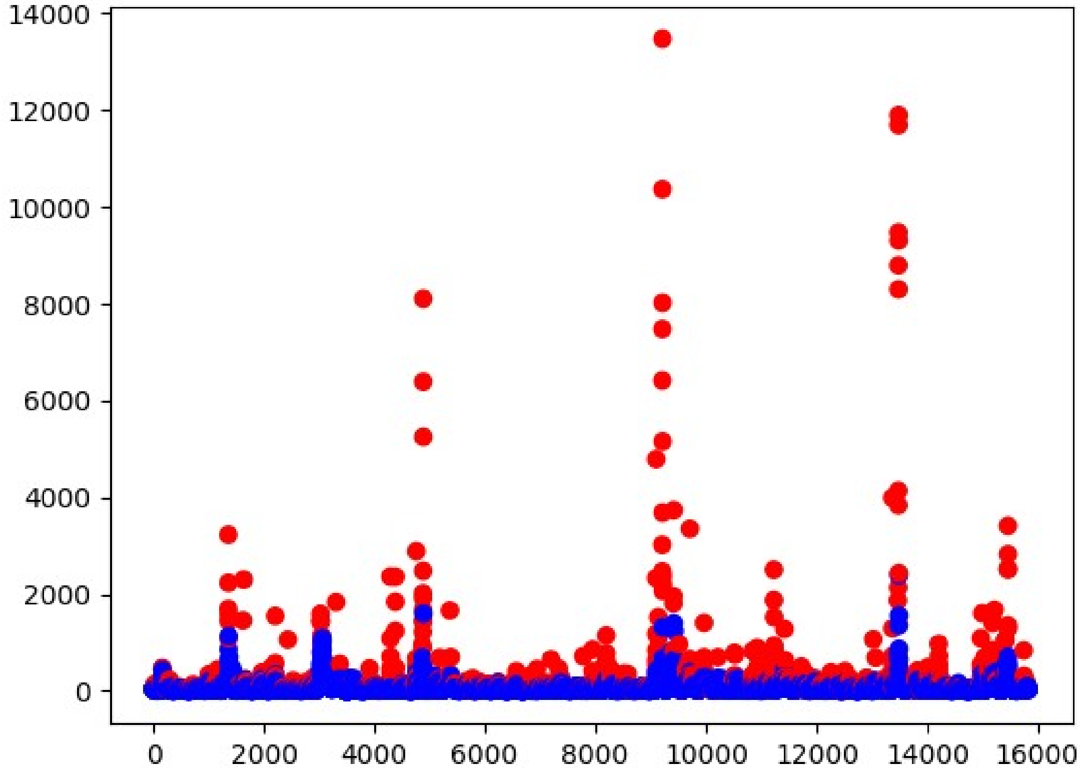
```python
train.head()
```

```
    timestamp  value  is_anomaly  predicted
0  1425008573     42       False  44.072500
1  1425008873     41       False  50.709390
2  1425009173     41       False  81.405120
3  1425009473     61       False  39.950367
4  1425009773     44       False  35.350160
```

VIZUALIZING THE DATA

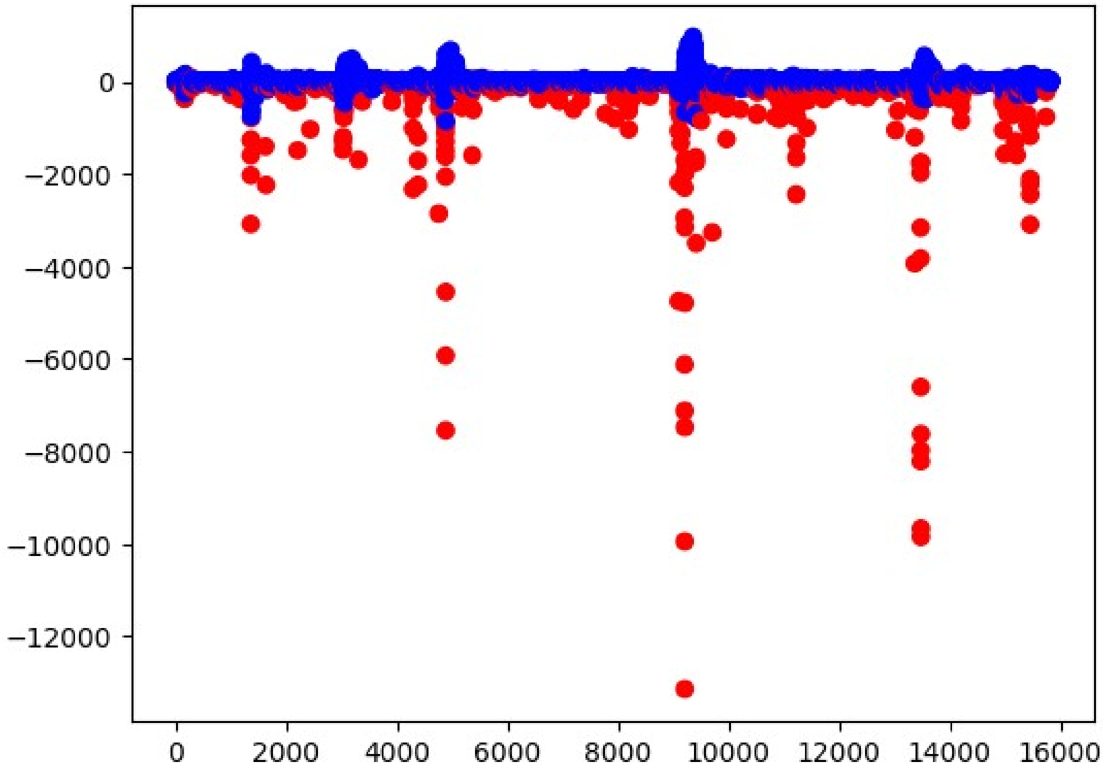
```python
import matplotlib.pyplot as plt

colors = np.where(train["is_anomaly"]==False,'b','r')
plt.scatter(range(15830),train['value'],c=colors)
```

```
<matplotlib.collections.PathCollection at 0x1591343ff20>
```




```
plt.scatter(range(15830),(train['predicted']-train['value']),c=colors)
```

```
<matplotlib.collections.PathCollection at 0x15913787020>
```




```
train.sort_values('timestamp').head()
```

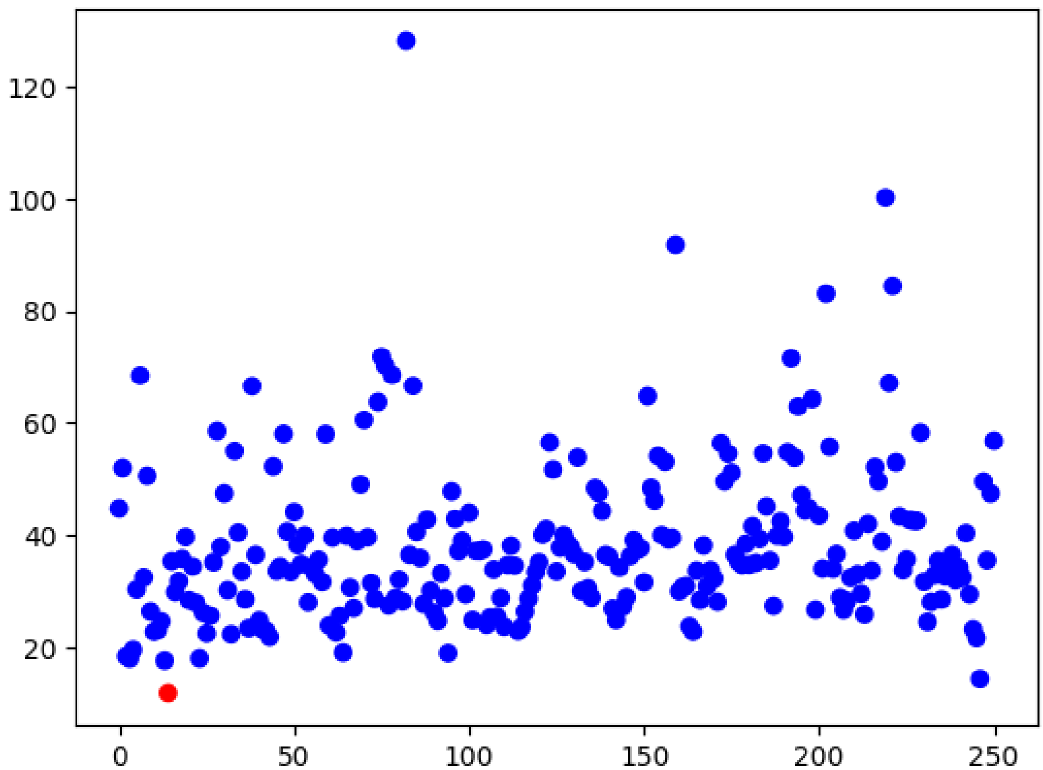
	timestamp	value	is_anomaly	predicted
0	1425008573	42	False	44.072500
1	1425008873	41	False	50.709390
2	1425009173	41	False	81.405120
3	1425009473	61	False	39.950367
4	1425009773	44	False	35.350160

```
train.value_counts('value')
```

```
value
29      259
21      252
30      251
33      251
26      245
       ...
3995      1
3838      1
3738      1
3689      1
3414      1
Name: count, Length: 631, dtype: int64
```

```
anomaly_column=[]
for i in range(15830):
    if train['value'][i]==30:
        if train['is_anomaly'][i]:
            anomaly_column.append('r')
        else:
            anomaly_column.append('b')
plt.scatter(range(251),train[train['value']==30]
['predicted'],c=anomaly_column)
```

```
<matplotlib.collections.PathCollection at 0x15913507f20>
```




```
train[train['value']==30].head(100)
```

```
        timestamp  value  is_anomaly  predicted
31    1425017873     30       False  44.719856
36    1425019373     30       False  51.963814
85    1425034073     30       False  18.330496
89    1425035273     30       False  17.930708
91    1425035873     30       False  19.451230
...          ...    ...         ...        ...
6170  1426859573     30       False  47.755207
6284  1426893773     30       False  42.847168
6311  1426901873     30       False  37.085125
6334  1426908773     30       False  38.982327
```

```
6339  1426910273     30       False  29.396100

[100 rows x 4 columns]
```

```
train.value_counts('is_anomaly')
```

```
is_anomaly
False    15054
True       776
Name: count, dtype: int64
```

```
train.head(5)
```

```
    timestamp  value  is_anomaly  predicted
0  1425008573     42       False  44.072500
1  1425008873     41       False  50.709390
2  1425009173     41       False  81.405120
3  1425009473     61       False  39.950367
4  1425009773     44       False  35.350160
```

```
train['value'].unique()
```

```
array([   42,    41,    61,    44,    27,    37,    36,    49,    32,
          50,    43,    47,    45,    56,    40,    57,    73,    59,
         140,    38,    28,    30,    31,    26,    20,    23,    15,
          25,    19,    18,    21,    17,    11,    13,    22,    14,
          16,    24,    29,    10,    39,    34,    35,    53,   222,
          46,   110,    65,    54,    66,   102,    92,    62,    86,
          67,    58,    52,    51,    68,   109,    80,   108,   172,
         271,   456,   440,   477,   426,   284,   159,   112,   118,
          70,    63,   105,    95,   111,   116,   155,   141,   115,
         137,   113,   136,   119,    97,   101,    96,    87,    93,
         103,    99,    90,   154,   131,   130,   100,   122,    89,
         106,   138,   123,   120,    88,   147,   129,    81,    85,
         114,    60,    91,    78,    74,    77,    83,    79,    76,
          98,   104,   117,   107,   204,   229,    82,     7,     9,
          12,     8,    55,    33,    48,    84,     6,    71,     4,
         134,    72,   346,   195,   149,   127,    64,    69,    75,
         235,   446,   226,   240,   264,   169,    94,   194,   193,
         162,   133,   132,   165,   379,  1698,  3228,  2234,  1452,
         865,  1088,  1585,  1132,   643,   536,   369,   294,   272,
         151,   143,   170,   694,   396,   203,   191,   146,   248,
         394,   416,   342,   219,   293,   316,   505,   292,   233,
         381,   343,   501,   312,   325,   257,   188,   153,   198,
         174,   201,   173,   453,   186,   181,  1458,   139,   183,
        2300,   125,   167,   243,   215,   156,   196,   158,   135,
         150,   232,   308,   388,   368,   253,   225,   192,   488,
         231,   254,   148,  1549,   571,   345,   205,   190,   171,
        1064,   211,   176,   275,   180,   197,   124,  1592,  1464,
        1422,   797,   567,   443,   399,   479,   529,   425,   424,
         600,   458,   551,   979,  1148,  1100,   981,  1087,   822,
```

```
         649,   561,   780,   591,   466,   523,   654,   846,   754,
         613,   996,  1018,   856,   685,   516,   430,   542,   496,
         537,   575,   603,   417,   344,   397,   374,   341,   323,
         418,   533,   461,   506,   473,   499,   423,   449,   380,
         433,   362,   340,   354,   616,   265,   245,   314,   349,
         334,   307,   309,   258,   287,   373,   317,   300,   352,
         297,   296,   286,   263,   283,   281,   290,   223,   302,
         375,   273,   269,   247,   278,   249,   260,   246,   252,
         242,   318,   500,   409,   370,   333,   299,   315,   288,
         274,   259,   319,   295,   175,   157,   267,   298,   255,
         241,   213,   214,   142,   187,   199,   224,   189,   179,
         145,   121,   303,   152,   285,  1835,   177,   168,   220,
         126,   207,   557,     0,   268,   185,   256,   210,     1,
           5,     2,   468,  2365,   710,   451,   489,   357,  1241,
        1843,  2361,   482,   161,   164,  2887,   674,   520,   261,
         216,   652,   957,   884,   913,   892,   384,   521,   476,
         494,   465,   434,   548,  1219,  1438,  2019,  1921,  1672,
        1443,  2481,  6393,  8107,  5249,  1596,   311,   322,   200,
         690,   184,   166,  1665,   160,   702,   178,     3,   227,
         324,   454,   217,   467,   212,   163,   270,   721,   858,
         415,   218,   144,   462,   403,   320,   638,  1147,   509,
         236,   422,   432,   363,   355,  4791,  2334,  1182,   844,
         648,   573,   412,   405,   387,   328,   310,   366,   301,
         238,   182,  1525,  1091,   450,   239,   202,  1206,   209,
        3024,  2471,  6418,  7479, 10372, 13479,  8025,  5157,  3689,
        2164,  2069,  2226,  2378,  1316,   679,   513,   244,   361,
         431,   128,   564,   206,   827,   753,   464,   208,   483,
        1959,  1811,   828,   514,   391,  3738,  1401,   554,   606,
         338,   350,   279,   647,  1238,   503,   313,   447,   389,
         326,   289,   967,   599,  3355,   662,   621,   277,   517,
         699,  1408,   438,   632,   701,   791,   330,   750,   234,
         742,   480,   889,   230,   769,   282,   586,  2505,  1877,
         751,   937,   670,   863,   452,   393,   237,   228,  1280,
         221,   377,   336,   351,   414,   266,  1068,   691,  3995,
        1294,   726,   592,   448,   615,   569,   623,  1871,   733,
         624,   568,   493,  2137,  4138,  3838,  8795,  9476,  8301,
       11694, 11899,  9310,  2393,  1353,  2437,  1561,   873,   653,
         455,   497,   419,   410,   339,   974,   457,   555,   725,
        1092,  1605,   693,   442,   812,  1403,   692,   460,  1441,
         470,   262,  1678,  1050,  1059,   543,   291,   402,   276,
        1360,  2526,  3414,  2826,  2510,  1299,   714,   576,   490,
         838])
```

```
train['value']-train['predicted']
```

```
0        -2.072500
1        -9.709390
2       -40.405120
3        21.049633
4         8.649840
```

```
             ...
15825    -9.624115
15826   -14.752296
15827    -4.147630
15828   -32.007545
15829   -21.144700
Length: 15830, dtype: float64
```

```
train.corr()
```

	timestamp	value	is_anomaly	predicted
timestamp	1.000000	0.032628	0.016457	0.030462
value	0.032628	1.000000	0.324859	0.445180
is_anomaly	0.016457	0.324859	1.000000	0.059719
predicted	0.030462	0.445180	0.059719	1.000000

MODEL CREATION AND TRAINING

```
from sklearn import svm

points=[]
y=[]
for i in range(15830):
    curr=[train['value'][i],abs(train['predicted'][i]-train['value']
[i])]
    points.append(curr)
    if train['is_anomaly'][i]==False:
        y.append(0)
    else:
        y.append(1)

clf=svm.SVC(class_weight={0:0.52577388,1:10.19974227})
clf.fit(points,y)
```

```
SVC(class_weight={0: 0.52577388, 1: 10.19974227})
```

```
from sklearn.utils.class_weight import compute_class_weight

class_weights = compute_class_weight(class_weight='balanced',
classes=np.unique(y), y=y)

print("Class weights:", class_weights)
```

```
Class weights: [ 0.52577388 10.19974227]
```

```
clf.predict([[20,100],[140,65]])
```

```
array([1, 1])
```