In [156]:

```
#Reading file
#Descriptive Statistics
# Part 1: presenting data set
import numpy as np
import pandas as pd
from subprocess import check_output
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.graphics.factorplots import interaction_plot
from statsmodels.stats.anova import anova_lm
sales=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWeek
ly_2.csv')
sales.describe()
```

Out[156]:

	W0	W1	W2	W3	W4	W5	
count	811.000000	811.000000	811.000000	811.000000	811.000000	811.000000	811.0
mean	8.902589	9.129470	9.389642	9.717633	9.574599	9.466091	9.720
std	12.067163	12.564766	13.045073	13.553294	13.095765	12.823195	13.34
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
50%	3.000000	3.000000	3.000000	4.000000	4.000000	3.000000	4.000
75%	12.000000	12.000000	12.000000	13.000000	13.000000	12.500000	13.00
max	54.000000	53.000000	56.000000	59.000000	61.000000	52.000000	56.00

8 rows × 52 columns

In [157]:

```
#Grouping the data by Product_Code and the mean
sales_mean=sales.groupby('Product_Code').mean()
print(sales_mean)
```

```
              W0  W1  W2  W3  W4  W5  W6  W7  W8  W9 ...   W42  W43 
W44  W45  \
Product_Code                                         ...
P1            11  12  10   8  13  12  14  21   6  14 ...     4    7
8   10
P10           22  19  19  29  20  16  26  20  24  20 ...    14   17
11   24
P100          13   6   7   9   9  15  13   6   7  10 ...    10   16
```

```
11    8
P101          26  51  45  47  40  36  38  28  38  45 ...    33   45
29   36
P102          34  41  22  32  36  27  44  33  22  38 ...    36   35
27   24
P103           7   5   7   7  14   9  10   6  12  13 ...    11    5
4    6
P104           4   4   5   5   3   7   5   6   5   7 ...     2    5
1    6
P105           2   3   3   2   5   4   5   7   5   3 ...     3    2
6    3
P106          11  10  11  16  16   7  11   6  16  15 ...     8    7
9    7
P107          19  19  15  24  16  23  27  20  19  17 ...    13   17
11   16
P108           0   2   2   0   3   1   2   1   6   3 ...     5    7
1    2
P109           9   8   8   5  13   9  13   7   9  14 ...     7    5
10    5
P11           15   7  15  14  17   7  10  16  11   8 ...    11   16
12   11
P110           9  12  12   6  15  10   9  10  10   9 ...    11    5
6    8
P111           5   4   7   4   4   5   4   4   2   6 ...     6    8
8    2
P112          31  36  32  51  39  33  35  46  57  34 ...    40   28
46   36
P113          22  26  31  37  32  25  28  33  40  27 ...    37   28
27   34
P114           7   6   7   8  10  12   7  10  11  14 ...     8    5
11   10
P115           9   9  12  13  14  13  10   6  10  11 ...    12    7
10   14
P116          13   7   8   8   7  12   9  11  11   9 ...    10    8
4    7
P117           6   2   3   3   3   2   5   2   4   4 ...     7    6
8    2
P118          16  13  10  10  16  15   9   5  13  13 ...     5    7
6    8
P119          30  40  46  33  28  41  34  33  44  49 ...    26   24
43   38
P12            3   4   1   6   4   3   7   3   5   3 ...     4    2
4    5
P120          37  35  27  30  31  42  30  37  37  39 ...    27   46
34   26
P121           7  10   8  10  11  10   9   9  11   7 ...     4   12
6    4
P122           7  10   7   8  11   2   9   7  12  17 ...    12    6
11   14
P123           5   5   3   6   4   6   4   7   4   2 ...     4    3
2    5
P124           4   6   0   7   4   3   5   4   2   7 ...     4   11
2    3
```

P125	6	9	5	10	6	9	10	8	15	14	...	9	6	7	10
...	..	..	..	..	..	..	..	..	..	..	...	...	...	...	...
P81	14	6	13	5	8	13	7	5	7	4	...	6	6	12	11
P810	0	0	1	0	0	0	0	0	0	0	...	1	1	1	0
P811	5	1	3	5	4	4	4	2	2	3	...	0	4	4	5
P812	3	2	0	0	2	1	0	2	2	2	...	0	1	3	1
P813	1	1	1	3	2	2	5	5	1	3	...	0	5	2	1
P814	4	2	2	6	0	4	7	8	3	2	...	6	7	3	4
P815	0	0	1	0	0	2	1	0	0	1	...	0	1	1	0
P816	0	1	0	0	1	2	2	6	0	1	...	3	3	4	2
P817	1	0	0	0	1	1	2	1	1	0	...	2	0	0	2
P818	0	0	0	1	0	0	0	0	1	0	...	0	0	0	1
P819	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0
P82	10	12	17	13	14	8	7	10	4	7	...	7	7	7	7
P83	46	40	34	38	36	45	48	37	63	37	...	31	32	42	36
P84	29	26	31	39	28	44	32	42	41	42	...	25	23	37	37
P85	39	34	28	38	33	37	52	28	31	27	...	14	29	20	25
P86	37	33	33	32	36	26	45	37	43	36	...	31	32	37	35
P87	30	38	35	39	37	42	40	35	40	41	...	31	25	23	27
P88	41	28	32	30	33	41	42	35	45	38	...	21	34	30	36
P89	24	26	44	31	37	37	38	30	35	49	...	30	32	25	38
P9	14	9	10	7	11	15	12	7	13	12	...	11	5	13	3
P90	42	34	40	39	47	31	43	40	47	35	...	33	29	25	30
P91	10	11	7	13	5	9	16	10	9	6	...	7	8	8	12
P92	26	36	56	40	31	44	37	40	38	43	...	42	26	32	40
P93	11	9	10	13	5	11	1	11	12	9	...	4	7	4	10
P94	14	9	10	9	5	11	6	6	6	9	...	10	10		

```
11    7
P95            12  15  15  13  17  12   9   9  11  12 ...     9   14
9   10
P96            31  35  36  44  36  38  36  39  47  39 ...    30   35
32   28
P97            29  37  36  40  33  29  27  49  38  43 ...    35   30
24   30
P98             4   2   4   4   7   2   3   4   4   4 ...     6    2
5    4
P99            16  11   9  21   9   5  12  15  11   9 ...     6    8
11    8
```

	W46	W47	W48	W49	W50	W51
Product_Code						
P1	12	3	7	6	5	10
P10	13	16	18	23	18	20
P100	17	16	8	10	23	21
P101	41	33	34	37	21	15
P102	22	19	32	26	31	29
P103	6	2	9	10	13	6
P104	3	3	3	5	5	4
P105	2	3	6	1	4	4
P106	8	9	8	13	14	21
P107	16	13	14	20	26	20
P108	3	3	3	4	1	4
P109	9	13	13	10	10	10
P11	13	8	17	10	10	21
P110	6	11	10	7	7	7
P111	1	6	2	3	3	9
P112	32	22	28	24	29	35
P113	41	39	28	32	40	16
P114	7	7	9	7	10	8
P115	12	12	15	7	6	10
P116	15	10	13	12	10	12
P117	3	5	3	4	5	2
P118	6	12	8	4	10	10
P119	35	35	24	29	33	19
P12	1	3	2	4	6	3
P120	27	38	37	27	29	32
P121	11	10	6	10	6	12
P122	13	6	9	1	1	12
P123	4	5	3	2	3	4
P124	8	3	2	2	9	4
P125	6	8	7	18	3	6
...	...	...	...	...	...	...
P81	6	3	3	13	12	12
P810	0	1	0	0	0	1
P811	6	3	4	5	3	5
P812	3	3	2	0	1	4
P813	1	1	3	1	4	4
P814	4	8	4	16	7	11
P815	0	1	0	0	2	0
P816	4	5	5	5	6	5

```
P817            2    0    0    0    4    3
P818            1    0    0    0    2    0
P819            0    0    0    0    0    1
P82            10   12   15    7    8    7
P83            34   31   31   26   26   33
P84            42   32   31   36   34   21
P85            36   46   34   36   31   16
P86            33   31   29   33   29   25
P87            36   24   40   23   20   24
P88            23   29   30   28   23   24
P89            25   29   28   28   12   28
P9              7    7   10   12    7   13
P90            40   37   36   29   27   18
P91             7    9    4    7    6    9
P92            40   29   34   30   30   19
P93             6    5    6    4   10   10
P94            10   11    9   10    6    9
P95             7   15   13    9   11   14
P96            30   41   40   39   38   21
P97            23   36   33   32   26   25
P98             7    2    5    4    5    5
P99             8    8   10    7    9   11

[811 rows x 52 columns]
```

In [158]:

```
# Making DataFrames
sales_df=pd.DataFrame(sales_mean)
print(sales_df)
```

```
              W0  W1  W2  W3  W4  W5  W6  W7  W8  W9 ...   W42  W43
W44  W45  \
Product_Code                                           ...
P1            11  12  10   8  13  12  14  21   6  14 ...     4    7
8   10
P10           22  19  19  29  20  16  26  20  24  20 ...    14   17
11   24
P100          13   6   7   9   9  15  13   6   7  10 ...    10   16
11    8
P101          26  51  45  47  40  36  38  28  38  45 ...    33   45
29   36
P102          34  41  22  32  36  27  44  33  22  38 ...    36   35
27   24
P103           7   5   7   7  14   9  10   6  12  13 ...    11    5
4    6
P104           4   4   5   5   3   7   5   6   5   7 ...     2    5
1    6
P105           2   3   3   2   5   4   5   7   5   3 ...     3    2
6    3
P106          11  10  11  16  16   7  11   6  16  15 ...     8    7
9    7
P107          19  19  15  24  16  23  27  20  19  17 ...    13   17
11   16
```

P108	0	2	2	0	3	1	2	1	6	3	...	5	7	1	2
P109	9	8	8	5	13	9	13	7	9	14	...	7	5	10	5
P11	15	7	15	14	17	7	10	16	11	8	...	11	16	12	11
P110	9	12	12	6	15	10	9	10	10	9	...	11	5	6	8
P111	5	4	7	4	4	5	4	4	2	6	...	6	8	8	2
P112	31	36	32	51	39	33	35	46	57	34	...	40	28	46	36
P113	22	26	31	37	32	25	28	33	40	27	...	37	28	27	34
P114	7	6	7	8	10	12	7	10	11	14	...	8	5	11	10
P115	9	9	12	13	14	13	10	6	10	11	...	12	7	10	14
P116	13	7	8	8	7	12	9	11	11	9	...	10	8	4	7
P117	6	2	3	3	3	2	5	2	4	4	...	7	6	8	2
P118	16	13	10	10	16	15	9	5	13	13	...	5	7	6	8
P119	30	40	46	33	28	41	34	33	44	49	...	26	24	43	38
P12	3	4	1	6	4	3	7	3	5	3	...	4	2	4	5
P120	37	35	27	30	31	42	30	37	37	39	...	27	46	34	26
P121	7	10	8	10	11	10	9	9	11	7	...	4	12	6	4
P122	7	10	7	8	11	2	9	7	12	17	...	12	6	11	14
P123	5	5	3	6	4	6	4	7	4	2	...	4	3	2	5
P124	4	6	0	7	4	3	5	4	2	7	...	4	11	2	3
P125	6	9	5	10	6	9	10	8	15	14	...	9	6	7	10
...	..	..	..	..	..	..	..	..	..	..	...	...	...	...	...
P81	14	6	13	5	8	13	7	5	7	4	...	6	6	12	11
P810	0	0	1	0	0	0	0	0	0	0	...	1	1	1	0
P811	5	1	3	5	4	4	4	2	2	3	...	0	4	4	5
P812	3	2	0	0	2	1	0	2	2	2	...	0	1	3	1
P813	1	1	1	3	2	2	5	5	1	3	...	0	5	2	1
P814	4	2	2	6	0	4	7	8	3	2	...	6	7		

```
3    4
P815            0   0   1   0   0   2   1   0   0   1 ...     0    1
1    0
P816            0   1   0   0   1   2   2   6   0   1 ...     3    3
4    2
P817            1   0   0   0   1   1   2   1   1   0 ...     2    0
0    2
P818            0   0   0   1   0   0   0   0   1   0 ...     0    0
0    1
P819            0   1   0   0   0   0   0   0   0   0 ...     0    0
0    0
P82            10  12  17  13  14   8   7  10   4   7 ...     7    7
7    7
P83            46  40  34  38  36  45  48  37  63  37 ...    31   32
42   36
P84            29  26  31  39  28  44  32  42  41  42 ...    25   23
37   37
P85            39  34  28  38  33  37  52  28  31  27 ...    14   29
20   25
P86            37  33  33  32  36  26  45  37  43  36 ...    31   32
37   35
P87            30  38  35  39  37  42  40  35  40  41 ...    31   25
23   27
P88            41  28  32  30  33  41  42  35  45  38 ...    21   34
30   36
P89            24  26  44  31  37  37  38  30  35  49 ...    30   32
25   38
P9             14   9  10   7  11  15  12   7  13  12 ...    11    5
13    3
P90            42  34  40  39  47  31  43  40  47  35 ...    33   29
25   30
P91            10  11   7  13   5   9  16  10   9   6 ...     7    8
8   12
P92            26  36  56  40  31  44  37  40  38  43 ...    42   26
32   40
P93            11   9  10  13   5  11   1  11  12   9 ...     4    7
4   10
P94            14   9  10   9   5  11   6   6   6   9 ...    10   10
11    7
P95            12  15  15  13  17  12   9   9  11  12 ...     9   14
9   10
P96            31  35  36  44  36  38  36  39  47  39 ...    30   35
32   28
P97            29  37  36  40  33  29  27  49  38  43 ...    35   30
24   30
P98             4   2   4   4   7   2   3   4   4   4 ...     6    2
5    4
P99            16  11   9  21   9   5  12  15  11   9 ...     6    8
11    8

              W46  W47  W48  W49  W50  W51
Product_Code
P1             12    3    7    6    5   10
```

P10	13	16	18	23	18	20
P100	17	16	8	10	23	21
P101	41	33	34	37	21	15
P102	22	19	32	26	31	29
P103	6	2	9	10	13	6
P104	3	3	3	5	5	4
P105	2	3	6	1	4	4
P106	8	9	8	13	14	21
P107	16	13	14	20	26	20
P108	3	3	3	4	1	4
P109	9	13	13	10	10	10
P11	13	8	17	10	10	21
P110	6	11	10	7	7	7
P111	1	6	2	3	3	9
P112	32	22	28	24	29	35
P113	41	39	28	32	40	16
P114	7	7	9	7	10	8
P115	12	12	15	7	6	10
P116	15	10	13	12	10	12
P117	3	5	3	4	5	2
P118	6	12	8	4	10	10
P119	35	35	24	29	33	19
P12	1	3	2	4	6	3
P120	27	38	37	27	29	32
P121	11	10	6	10	6	12
P122	13	6	9	1	1	12
P123	4	5	3	2	3	4
P124	8	3	2	2	9	4
P125	6	8	7	18	3	6
...	...	...	...	...	...	...
P81	6	3	3	13	12	12
P810	0	1	0	0	0	1
P811	6	3	4	5	3	5
P812	3	3	2	0	1	4
P813	1	1	3	1	4	4
P814	4	8	4	16	7	11
P815	0	1	0	0	2	0
P816	4	5	5	5	6	5
P817	2	0	0	0	4	3
P818	1	0	0	0	2	0
P819	0	0	0	0	0	1
P82	10	12	15	7	8	7
P83	34	31	31	26	26	33
P84	42	32	31	36	34	21
P85	36	46	34	36	31	16
P86	33	31	29	33	29	25
P87	36	24	40	23	20	24
P88	23	29	30	28	23	24
P89	25	29	28	28	12	28
P9	7	7	10	12	7	13
P90	40	37	36	29	27	18
P91	7	9	4	7	6	9
P92	40	29	34	30	30	19

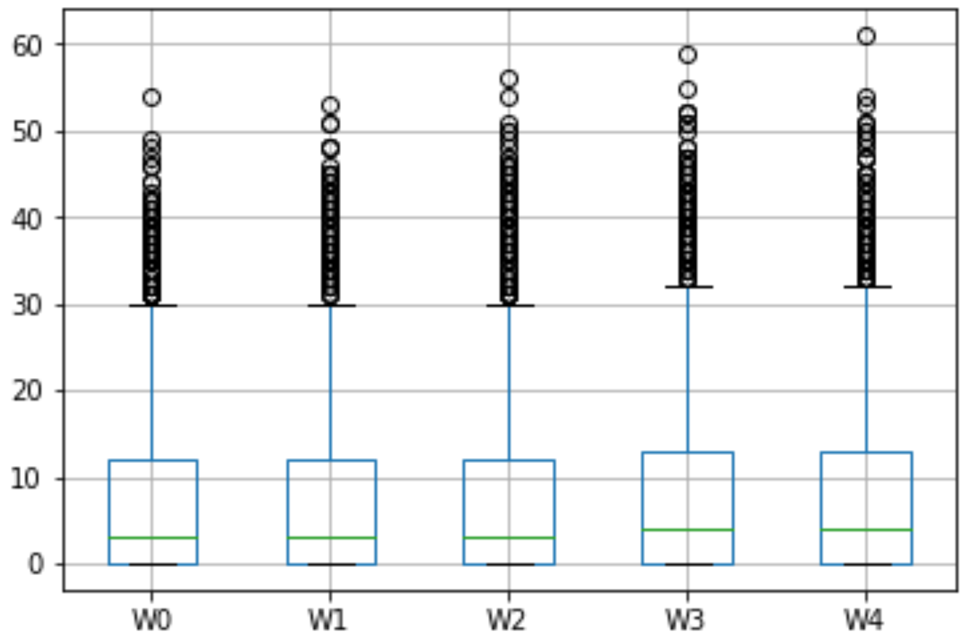
```
P93                6     5     6     4    10    10
P94               10    11     9    10     6     9
P95                7    15    13     9    11    14
P96               30    41    40    39    38    21
P97               23    36    33    32    26    25
P98                7     2     5     4     5     5
P99                8     8    10     7     9    11

[811 rows x 52 columns]
```

In [159]:

```
sales1=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWeekly_2_1.csv')
df = sales1.set_index('Product_Code')
df.boxplot()
```

Out[159]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c2af7f2e8>
```

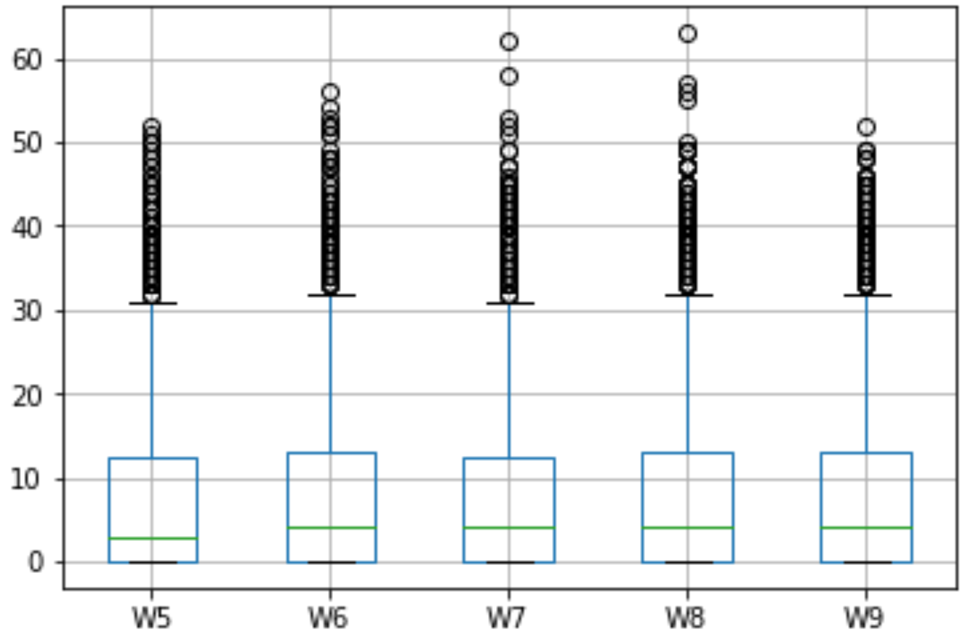


In [160]:

```
sales2=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2_2.csv')
df = sales2.set_index('Product_Code')
df.boxplot()
```

Out[160]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c28b5ac88>

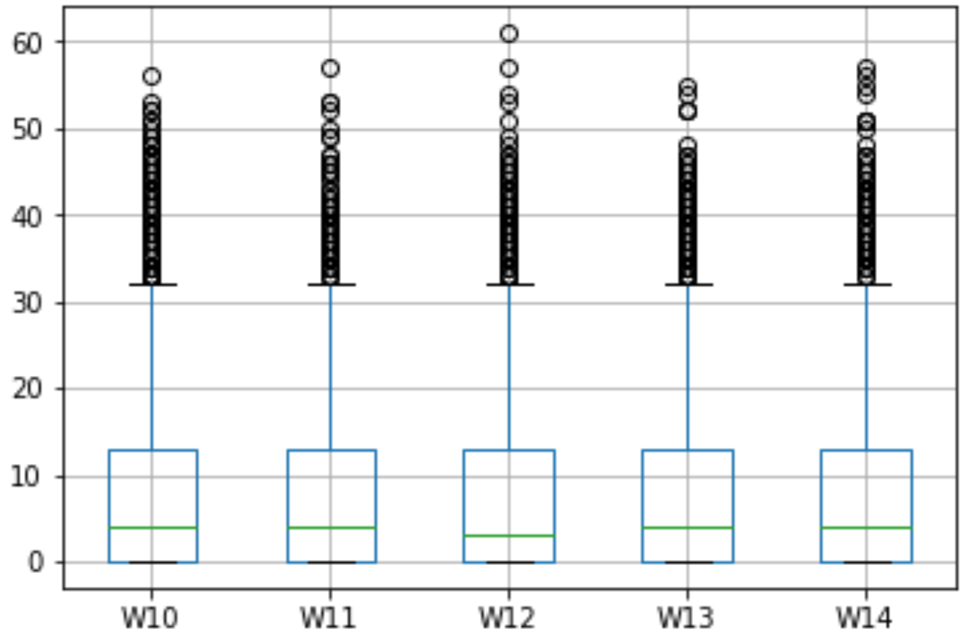


In [161]:

```
sales3=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2 _3.csv')
df = sales3.set_index('Product_Code')
df.boxplot()
```

Out[161]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c289c9668>

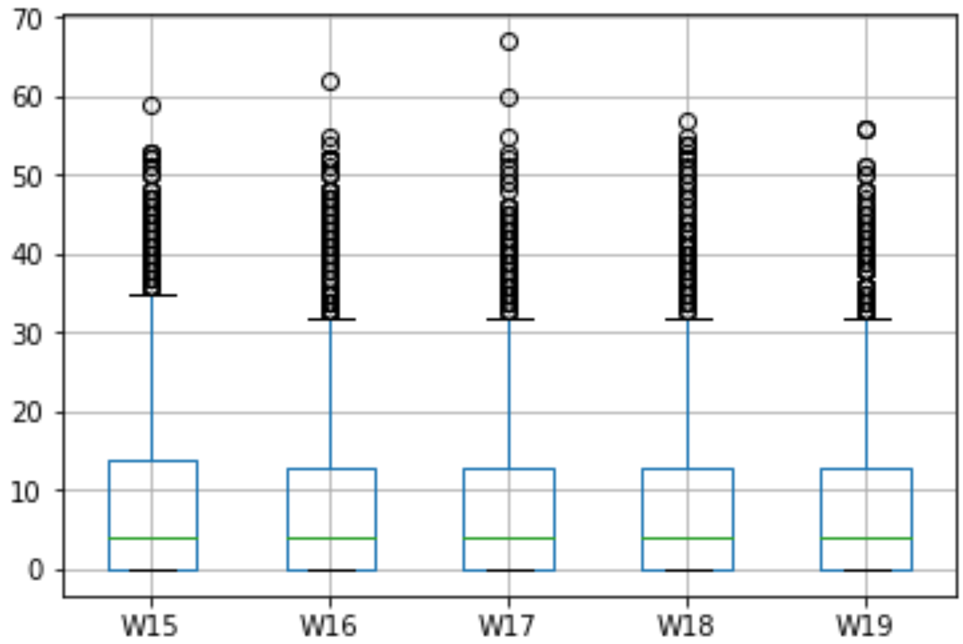


In [162]:

```
sales4=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2_4.csv')
df = sales4.set_index('Product_Code')
df.boxplot()
```

Out[162]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c2a07a748>

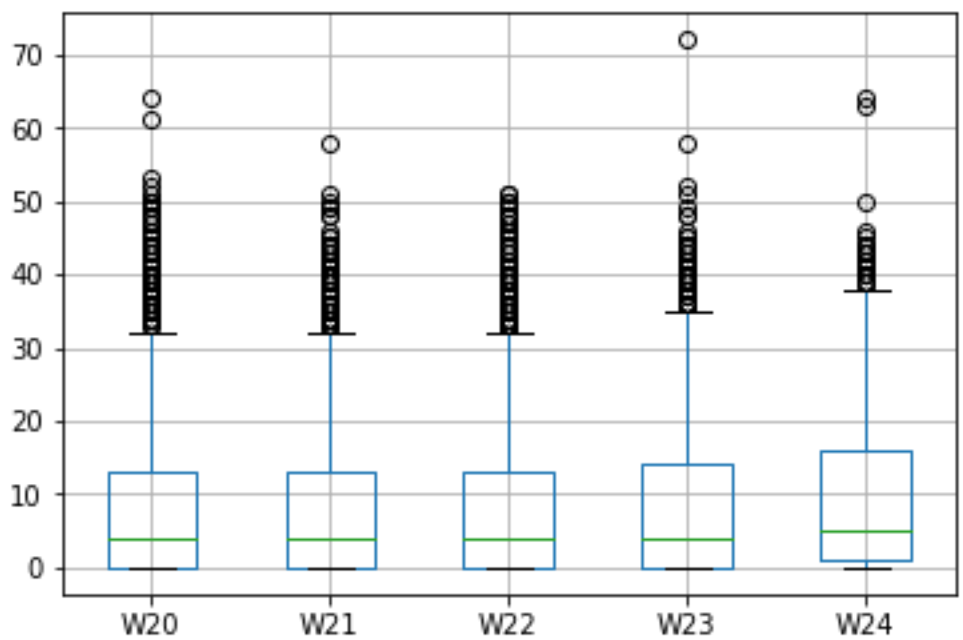


In [163]:

```
sales5=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2_5.csv')
df = sales5.set_index('Product_Code')
df.boxplot()
```

Out[163]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c2ce32438>

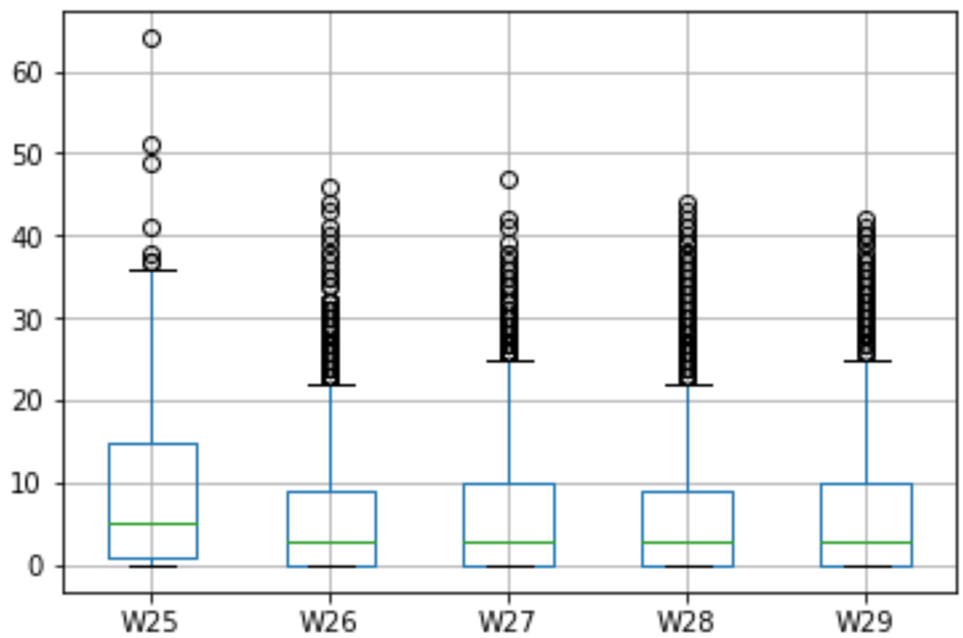


In [164]:

```
sales6=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2_6.csv')
df = sales6.set_index('Product_Code')
df.boxplot()
```

Out[164]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c2cd53b00>

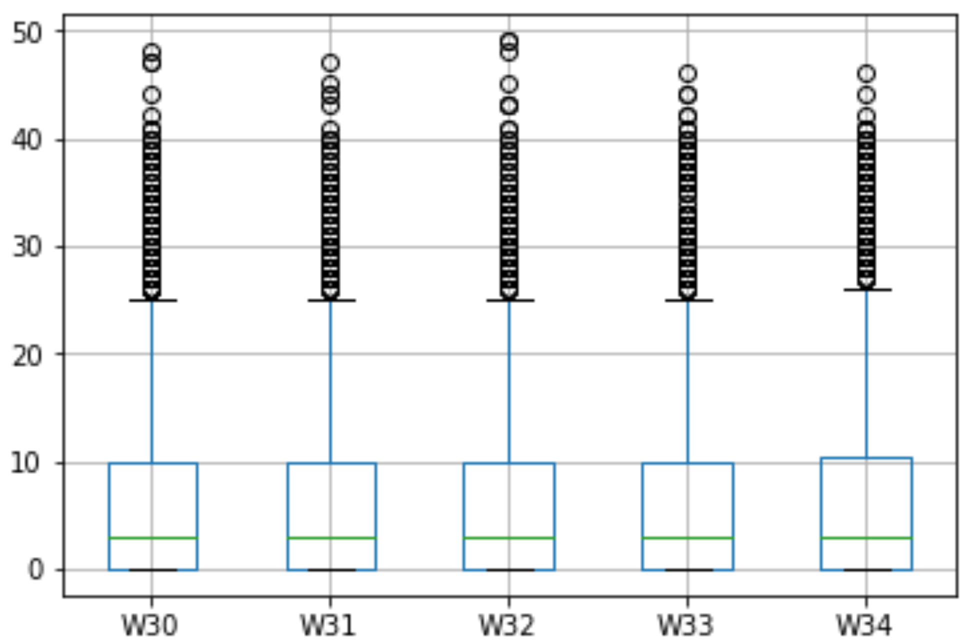


In [165]:

```
sales7=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2_7.csv')
df = sales7.set_index('Product_Code')
df.boxplot()
```

Out[165]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c295c1b70>

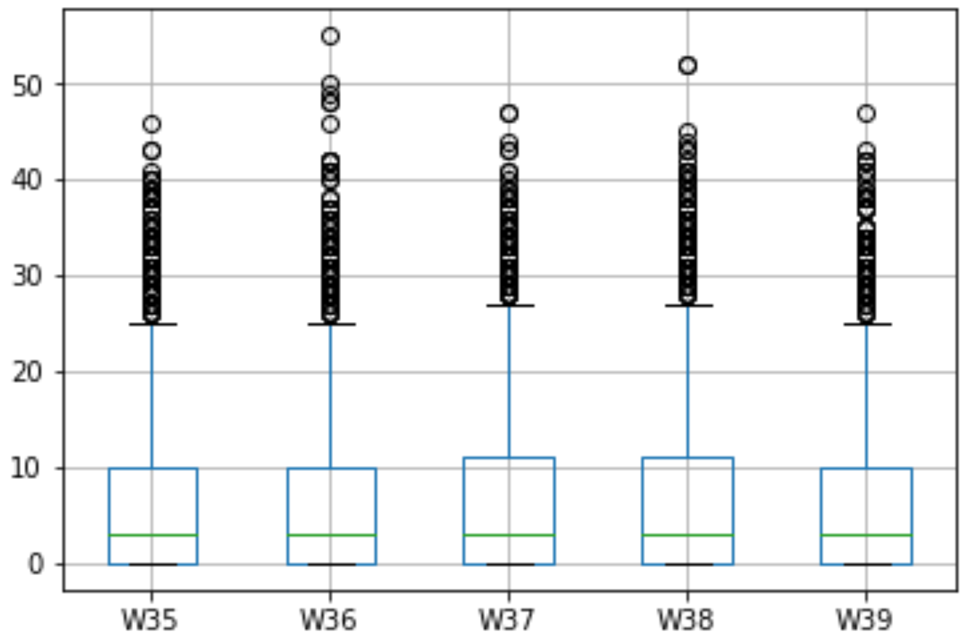


In [166]:

```
sales8=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2_8.csv')
df = sales8.set_index('Product_Code')
df.boxplot()
```

Out[166]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c29dc53c8>

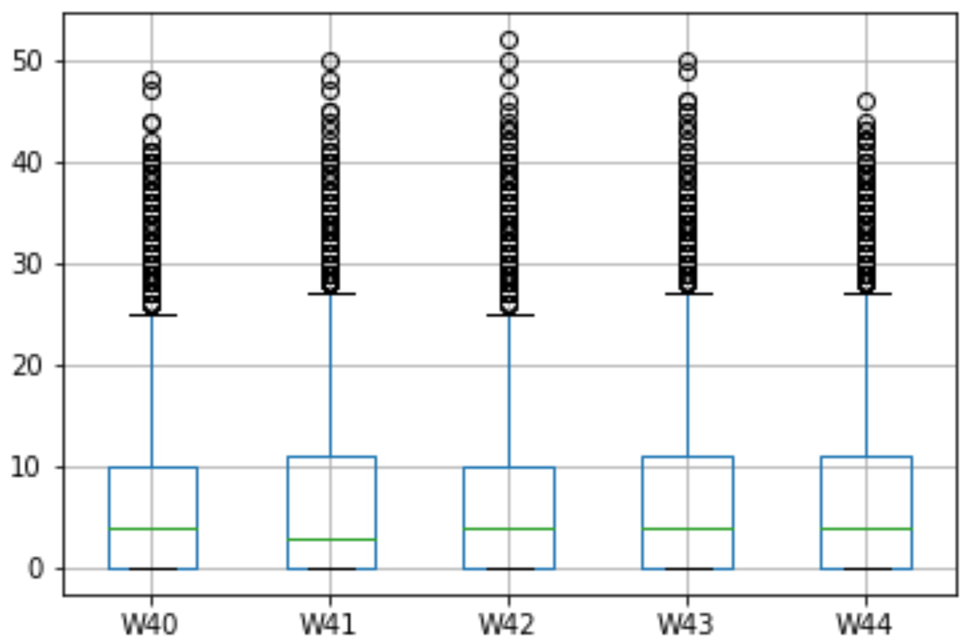


In [167]:

```
sales9=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2_9.csv')
df = sales9.set_index('Product_Code')
df.boxplot()
```

Out[167]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c2b892438>

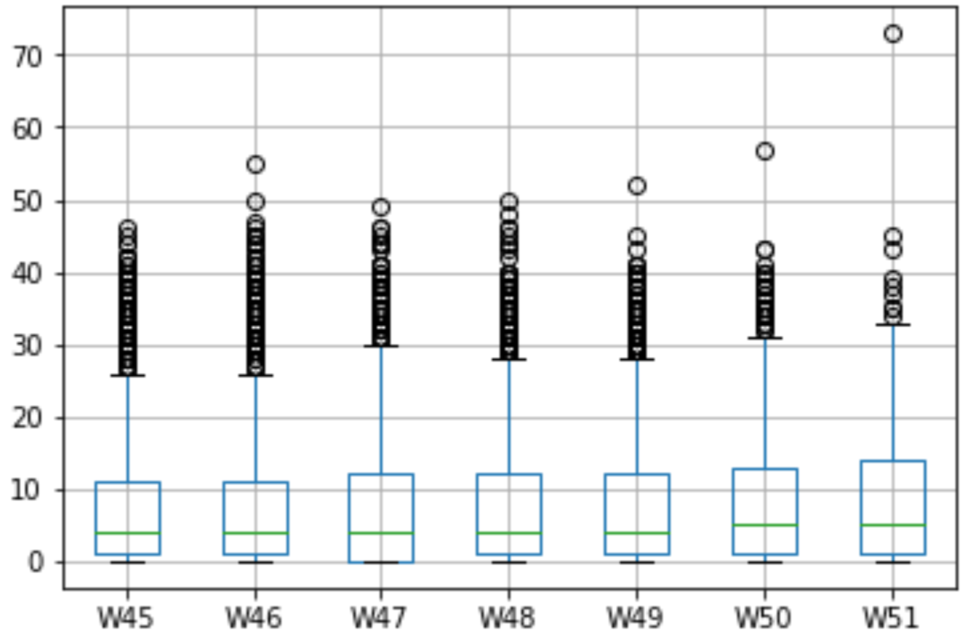


In [168]:

```
sales10=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWe
ekly_2_10.csv')
df = sales10.set_index('Product_Code')
df.boxplot()
```

Out[168]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c2cfc2128>

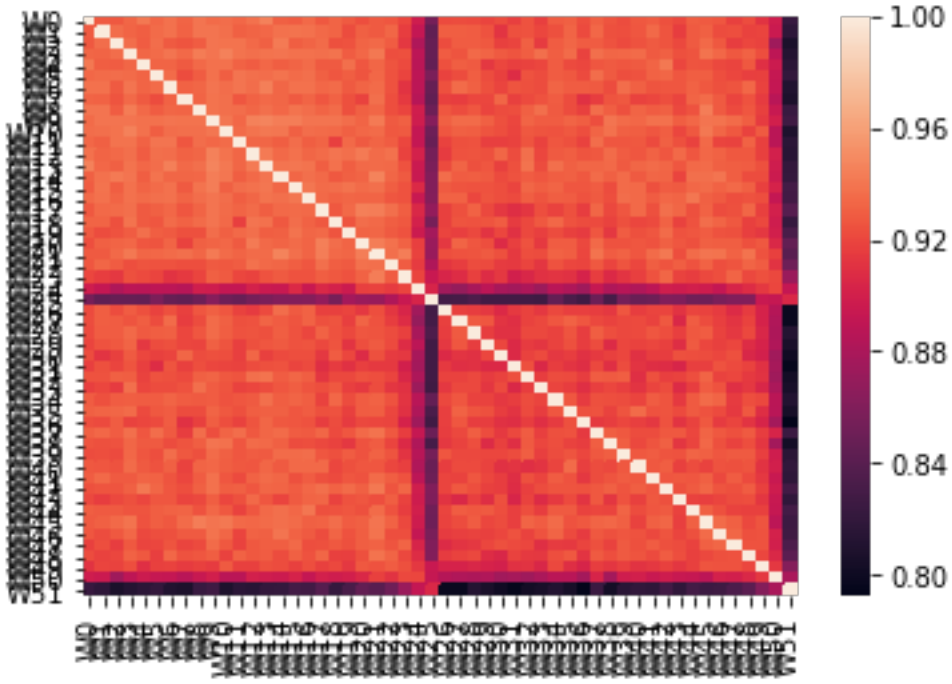


In [169]:

```
import seaborn as sns
%matplotlib inline
# calculate the correlation matrix
corr = sales.corr()

# plot the heatmap
sns.heatmap(corr,
        xticklabels=corr.columns,
        yticklabels=corr.columns)
```

Out[169]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c2cf9edd8>
```



In [170]:

```
# represent correlation matrix using a heatmap
#Ref: https://stackoverflow.com/questions/39409866/correlation-heatmap
cmap = cmap=sns.diverging_palette(5, 250, as_cmap=True)

def magnify():
    return [dict(selector="th",
                 props=[("font-size", "7pt")]),
            dict(selector="td",
                 props=[('padding', "0em 0em")]),
            dict(selector="th:hover",
                 props=[("font-size", "12pt")]),
            dict(selector="tr:hover td:hover",
                 props=[('max-width', '200px'),
                        ('font-size', '12pt')])
]

corr.style.background_gradient(cmap, axis=1)\
    .set_properties(**{'max-width': '80px', 'font-size': '10pt'})\
    .set_caption("Hover to magify")\
    .set_precision(2)\
    .set_table_styles(magnify())
```

Out[170]:

Hover to magify

	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W2
W0	1	0.93	0.93	0.93	0.94	0.93	0.94	0.93	0.94	0.94	0.94	0.93	0.94	0.95	0.93	0.93	0.93	0.93	0.93	0.93	0.9
W1	0.93	1	0.94	0.93	0.93	0.94	0.94	0.93	0.94	0.94	0.94	0.93	0.93	0.93	0.94	0.94	0.94	0.93	0.93	0.92	0.9
W2	0.93	0.94	1	0.94	0.93	0.94	0.93	0.93	0.93	0.94	0.94	0.94	0.93	0.93	0.94	0.93	0.93	0.94	0.92	0.92	0.9
W3	0.93	0.93	0.94	1	0.93	0.94	0.94	0.94	0.94	0.94	0.95	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.93	0.9
W4	0.94	0.93	0.93	0.93	1	0.93	0.93	0.93	0.93	0.94	0.94	0.93	0.93	0.93	0.93	0.93	0.94	0.93	0.93	0.94	0.9
W5	0.93	0.94	0.94	0.94	0.93	1	0.94	0.92	0.93	0.94	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.93	0.93	0.93	0.9
W6	0.94	0.94	0.93	0.94	0.93	0.94	1	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.93	0.94	0.94	0.93	0.94	0.93	0.9
W7	0.93	0.93	0.93	0.94	0.93	0.92	0.93	1	0.93	0.94	0.94	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.92	0.9
W8	0.94	0.94	0.93	0.94	0.93	0.93	0.93	0.93	1	0.94	0.94	0.94	0.93	0.94	0.93	0.94	0.94	0.93	0.94	0.92	0.9
W9	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	1	0.94	0.93	0.95	0.95	0.94	0.94	0.94	0.94	0.94	0.94	0.9
W10	0.94	0.94	0.94	0.95	0.94	0.93	0.94	0.94	0.94	0.94	1	0.94	0.94	0.94	0.94	0.93	0.94	0.94	0.93	0.93	0.9
W11	0.93	0.93	0.94	0.94	0.93	0.94	0.94	0.93	0.94	0.93	0.94	1	0.93	0.93	0.93	0.94	0.93	0.93	0.92	0.93	0.9
W12	0.94	0.93	0.93	0.94	0.93	0.94	0.94	0.93	0.93	0.95	0.94	0.93	1	0.94	0.93	0.94	0.93	0.94	0.94	0.94	0.9

W13	0.95	0.93	0.93	0.94	0.93	0.94	0.94	0.93	0.94	0.95	0.94	0.93	0.94	1	0.94	0.94	0.94	0.93	0.94	0.93	0.9
W14	0.93	0.94	0.94	0.94	0.93	0.94	0.93	0.93	0.93	0.94	0.94	0.93	0.93	0.94	1	0.93	0.94	0.93	0.93	0.93	0.9
W15	0.93	0.94	0.93	0.94	0.93	0.94	0.94	0.93	0.94	0.94	0.93	0.94	0.94	0.94	0.93	1	0.94	0.94	0.94	0.94	0.9
W16	0.93	0.94	0.93	0.94	0.94	0.94	0.94	0.93	0.94	0.94	0.94	0.93	0.93	0.94	0.94	0.94	1	0.93	0.93	0.93	0.9
W17	0.93	0.93	0.94	0.94	0.93	0.93	0.93	0.93	0.93	0.94	0.94	0.93	0.94	0.93	0.93	0.94	0.93	1	0.93	0.93	0.9
W18	0.93	0.93	0.92	0.94	0.93	0.93	0.94	0.93	0.94	0.94	0.93	0.92	0.94	0.94	0.93	0.94	0.93	0.93	1	0.93	0.9
W19	0.93	0.92	0.92	0.93	0.94	0.93	0.93	0.92	0.92	0.94	0.93	0.93	0.94	0.93	0.93	0.94	0.93	0.93	0.93	1	0.9
W20	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.92	0.93	0.93	0.93	0.93	0.94	0.93	0.93	0.94	0.93	0.94	0.94	0.93	1
W21	0.93	0.93	0.94	0.94	0.93	0.93	0.94	0.94	0.93	0.94	0.94	0.93	0.94	0.94	0.95	0.94	0.93	0.94	0.93	0.93	0.9
W22	0.93	0.93	0.93	0.92	0.93	0.93	0.93	0.92	0.92	0.93	0.93	0.93	0.94	0.93	0.93	0.93	0.93	0.94	0.93	0.93	0.9
W23	0.92	0.92	0.91	0.92	0.92	0.92	0.91	0.91	0.92	0.93	0.93	0.92	0.92	0.93	0.93	0.92	0.92	0.93	0.93	0.93	0.9
W24	0.9	0.89	0.88	0.89	0.89	0.89	0.89	0.89	0.9	0.9	0.89	0.89	0.9	0.9	0.9	0.91	0.89	0.91	0.9	0.9	0.9
W25	0.85	0.84	0.85	0.85	0.85	0.86	0.85	0.84	0.85	0.86	0.85	0.85	0.85	0.86	0.86	0.86	0.86	0.87	0.86	0.88	0.8
W26	0.92	0.93	0.92	0.93	0.92	0.92	0.92	0.92	0.93	0.93	0.93	0.92	0.92	0.92	0.94	0.93	0.93	0.92	0.92	0.92	0.9
W27	0.92	0.93	0.93	0.93	0.93	0.92	0.94	0.92	0.92	0.94	0.93	0.92	0.92	0.93	0.93	0.93	0.93	0.92	0.93	0.92	0.9
W28	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.92	0.93	0.94	0.93	0.92	0.93	0.92	0.93	0.92	0.92	0.93	0.93	0.92	0.9
W29	0.92	0.92	0.93	0.93	0.91	0.92	0.92	0.92	0.92	0.93	0.93	0.92	0.92	0.93	0.93	0.92	0.93	0.92	0.93	0.91	0.9
W30	0.92	0.91	0.92	0.93	0.92	0.92	0.92	0.93	0.92	0.92	0.92	0.92	0.93	0.92	0.92	0.92	0.92	0.91	0.93	0.92	0.9
W31	0.92	0.92	0.92	0.93	0.92	0.91	0.93	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.93	0.92	0.92	0.91	0.91	0.91	0.9
W32	0.92	0.93	0.93	0.93	0.92	0.93	0.92	0.92	0.93	0.93	0.93	0.93	0.94	0.94	0.93	0.93	0.93	0.91	0.92	0.92	0.9
W33	0.92	0.92	0.92	0.94	0.92	0.92	0.92	0.92	0.94	0.93	0.93	0.92	0.92	0.92	0.93	0.93	0.93	0.92	0.93	0.92	0.9
W34	0.93	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.92	0.93	0.93	0.94	0.93	0.93	0.93	0.92	0.93	0.92	0.9
W35	0.93	0.93	0.93	0.94	0.92	0.93	0.93	0.92	0.92	0.93	0.93	0.92	0.93	0.93	0.93	0.93	0.93	0.92	0.92	0.92	0.9
W36	0.92	0.92	0.93	0.92	0.92	0.92	0.93	0.91	0.92	0.93	0.92	0.92	0.93	0.93	0.93	0.93	0.93	0.92	0.92	0.92	0.9
W37	0.93	0.93	0.93	0.94	0.93	0.94	0.93	0.92	0.94	0.92	0.93	0.94	0.93	0.93	0.94	0.93	0.93	0.93	0.92	0.93	0.9
W38	0.93	0.92	0.92	0.93	0.93	0.93	0.92	0.93	0.93	0.94	0.94	0.93	0.94	0.93	0.93	0.92	0.92	0.92	0.92	0.91	0.9
W39	0.92	0.92	0.92	0.93	0.93	0.92	0.94	0.93	0.92	0.94	0.93	0.93	0.93	0.93	0.94	0.94	0.94	0.93	0.92	0.93	0.9
W40	0.93	0.93	0.92	0.94	0.93	0.93	0.93	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.92	0.92	0.9
W41	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.92	0.93	0.94	0.93	0.92	0.93	0.94	0.93	0.92	0.93	0.92	0.92	0.93	0.9
W42	0.92	0.93	0.93	0.93	0.94	0.93	0.93	0.92	0.93	0.94	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.94	0.93	0.94	0.9
W43	0.92	0.92	0.92	0.92	0.92	0.93	0.93	0.92	0.92	0.93	0.92	0.92	0.93	0.93	0.92	0.93	0.93	0.93	0.92	0.93	0.9

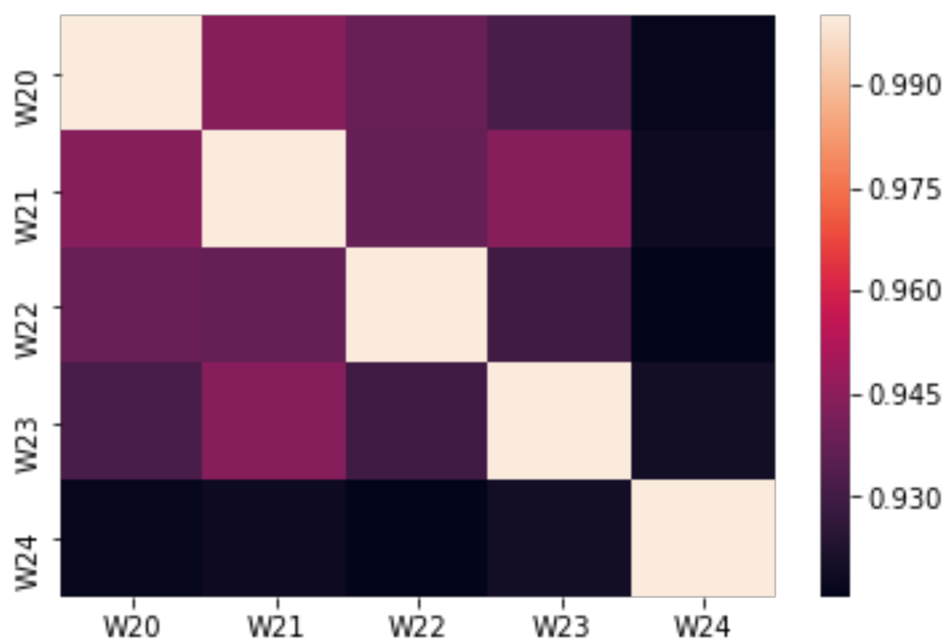
W44	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.92	0.93	0.93	0.92	0.93	0.93	0.94	0.93	0.94	0.92	0.92	0.93	0.9
W45	0.93	0.93	0.93	0.94	0.93	0.93	0.93	0.94	0.94	0.94	0.94	0.93	0.93	0.94	0.94	0.94	0.93	0.93	0.94	0.93	0.9
W46	0.93	0.92	0.93	0.93	0.92	0.92	0.92	0.93	0.93	0.94	0.93	0.92	0.93	0.93	0.94	0.93	0.93	0.92	0.93	0.92	0.9
W47	0.92	0.92	0.92	0.93	0.92	0.92	0.93	0.92	0.92	0.94	0.93	0.92	0.93	0.93	0.93	0.93	0.92	0.92	0.93	0.92	0.9
W48	0.92	0.92	0.92	0.93	0.93	0.92	0.93	0.93	0.93	0.93	0.92	0.93	0.93	0.93	0.92	0.93	0.92	0.93	0.92	0.92	0.9
W49	0.91	0.92	0.92	0.92	0.91	0.92	0.92	0.92	0.92	0.92	0.93	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.91	0.9
W50	0.9	0.9	0.87	0.89	0.9	0.89	0.89	0.89	0.89	0.9	0.89	0.89	0.9	0.9	0.9	0.9	0.89	0.9	0.9	0.9	0.9
W51	0.82	0.82	0.81	0.81	0.82	0.82	0.81	0.81	0.82	0.83	0.81	0.82	0.81	0.82	0.82	0.83	0.83	0.84	0.82	0.83	0.8

In [171]:

```
sales5=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2_5.csv')
import seaborn as sns
%matplotlib inline
# calculate the correlation matrix
corr = sales5.corr()

# plot the heatmap
sns.heatmap(corr,
        xticklabels=corr.columns,
        yticklabels=corr.columns)
```

Out[171]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c2f44bef0>

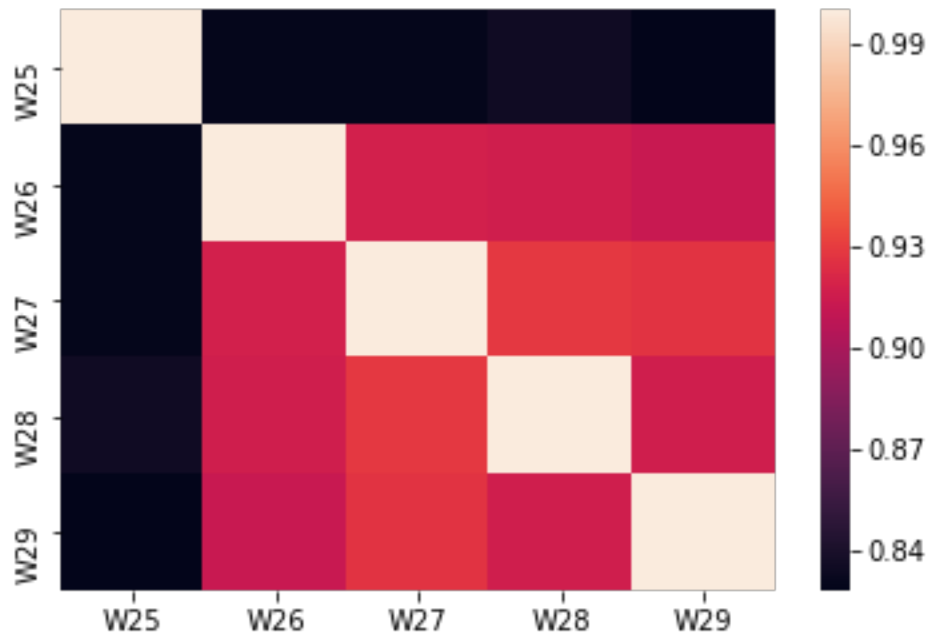


In [172]:

```
sales6=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWee
kly_2_6.csv')
import seaborn as sns
%matplotlib inline
# calculate the correlation matrix
corr = sales6.corr()

# plot the heatmap
sns.heatmap(corr,
        xticklabels=corr.columns,
        yticklabels=corr.columns)
```

Out[172]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c2fe69828>

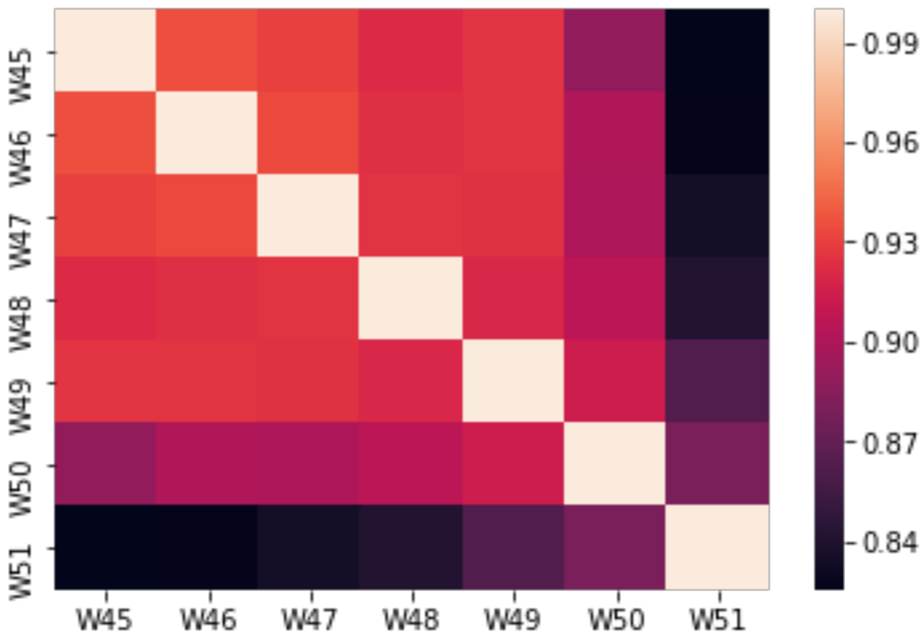


In [173]:

```
sales10=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_TransactionsWe
ekly_2_10.csv')
import seaborn as sns
%matplotlib inline
# calculate the correlation matrix
corr = sales10.corr()

# plot the heatmap
sns.heatmap(corr,
        xticklabels=corr.columns,
        yticklabels=corr.columns)
```

Out[173]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c2f5a3710>



In [174]:

```
EE=ols('W25~W0+W1+W2+W3+W4+W5+W6+W7+W8+W9+W10+W11+W12+W13+W14+W15+W16+W17+W18+W1
9+W20+W21+W22+W23+W24+W26+W27+W28+W29+W30+W31+W32+W33+W34+W35+W36+W37+W38+W39+W4
0+W41+W42+W43+W44+W45+W46+W47+W48+W49+W50+W51',data=sales).fit()
table_aov=sm.stats.anova_lm(EE,typ=2)
print (table_aov)
```

```
             sum_sq    df          F        PR(>F)
W0         6.310168   1.0   0.596113  4.403052e-01
W1        51.968980   1.0   4.909440  2.700629e-02
W2        46.547358   1.0   4.397266  3.632693e-02
W3         3.719886   1.0   0.351413  5.534911e-01
W4        38.689844   1.0   3.654978  5.627817e-02
W5        53.303479   1.0   5.035508  2.512029e-02
W6         1.710073   1.0   0.161548  6.878481e-01
W7        10.368295   1.0   0.979479  3.226429e-01
W8         6.793122   1.0   0.641737  4.233332e-01
W9        54.335617   1.0   5.133013  2.375589e-02
W10        3.217965   1.0   0.303997  5.815498e-01
```

W11	33.657738	1.0	3.179601	7.496245e-02
W12	0.600620	1.0	0.056740	8.117895e-01
W13	8.693415	1.0	0.821255	3.651010e-01
W14	43.047918	1.0	4.066679	4.408891e-02
W15	8.066831	1.0	0.762063	3.829599e-01
W16	19.859386	1.0	1.876090	1.711843e-01
W17	1.050779	1.0	0.099266	7.527993e-01
W18	0.008272	1.0	0.000781	9.777064e-01
W19	283.901062	1.0	26.819753	2.863754e-07
W20	0.034059	1.0	0.003217	9.547809e-01
W21	2.881287	1.0	0.272191	6.020178e-01
W22	94.293174	1.0	8.907750	2.930797e-03
W23	85.396887	1.0	8.067329	4.627692e-03
W24	211.256548	1.0	19.957123	9.125264e-06
W26	36.228480	1.0	3.422456	6.470377e-02
W27	4.279678	1.0	0.404295	5.250709e-01
W28	21.303810	1.0	2.012542	1.564142e-01
W29	117.773592	1.0	11.125913	8.926835e-04
W30	26.863776	1.0	2.537785	1.115670e-01
W31	2.477310	1.0	0.234028	6.286925e-01
W32	89.497664	1.0	8.454724	3.747158e-03
W33	72.937287	1.0	6.890288	8.840906e-03
W34	0.477124	1.0	0.045073	8.319264e-01
W35	18.954798	1.0	1.790634	1.812496e-01
W36	10.459932	1.0	0.988136	3.205152e-01
W37	1.128848	1.0	0.106641	7.440915e-01
W38	36.154371	1.0	3.415455	6.497745e-02
W39	74.309770	1.0	7.019944	8.228381e-03
W40	3.492769	1.0	0.329957	5.658545e-01
W41	0.000254	1.0	0.000024	9.960945e-01
W42	2.651954	1.0	0.250527	6.168496e-01
W43	14.930653	1.0	1.410479	2.353485e-01
W44	34.302194	1.0	3.240482	7.223645e-02
W45	23.720487	1.0	2.240843	1.348234e-01
W46	0.891595	1.0	0.084228	7.717271e-01
W47	11.130165	1.0	1.051452	3.054996e-01
W48	14.341029	1.0	1.354778	2.448103e-01
W49	298.612073	1.0	28.209482	1.431565e-07
W50	63.673808	1.0	6.015179	1.440740e-02
W51	1477.774651	1.0	139.603390	1.085528e-29
Residual	8034.410644	759.0	NaN	NaN

In [175]:

```
sales=pd.read_csv('/Users/sam/Desktop/MSBA 305/Term_paper/Sales_Transactions_Dat
aset_Weekly.csv')
sales.describe()
```

Out[175]:

	W0	W1	W2	W3	W4	W5	
count	811.000000	811.000000	811.000000	811.000000	811.000000	811.000000	811.0
mean	8.902589	9.129470	9.389642	9.717633	9.574599	9.466091	9.720
std	12.067163	12.564766	13.045073	13.553294	13.095765	12.823195	13.34
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
50%	3.000000	3.000000	3.000000	4.000000	4.000000	3.000000	4.000
75%	12.000000	12.000000	12.000000	13.000000	13.000000	12.500000	13.00
max	54.000000	53.000000	56.000000	59.000000	61.000000	52.000000	56.00

8 rows × 106 columns

In [176]:

```
# PCA: Principal component analysis (PCA) is a technique for redusing the data d
imension by use of linear mapping so that the
# data variance in the low-dimensional map would be maximized.
# FastICA: FastICA is an efficient and popular algorithm for independent compone
nt analysis. FastICA seeks an orthogonal
# rotation of prewhitened data,through a fixed-point iteration scheme, that maxi
mizes a measure of non-Gaussianity
# of the rotated components.
# sparse PCA: Sparse principal component analysis (sparse PCA) is a specialised
technique used in statistical analysis and,
# in particular, in the analysis of multivariate data sets. It extends the class
ic method of PCA for the reduction
# of dimensionality of data by adding sparsity constraint on the input variables
.
# NMF or NNMF: Non-negative matrix factorization (NMF or NNMF), also non-negativ
e matrix approximation is a group of algorithms
# in multivariate analysis and linear algebra where a matrix V is factorized int
o (usually) two matrices W and H, with the
# property that all three matrices have no negative elements.
# LatentDirichletAllocation: is a generative statistical model that allows sets
of observations to be explained by unobserved
# groups that explain why some parts of the data are similar.
# FactorAnalysis: aim to reduce the dimensionality of a set of data, but the app
roaches taken to do so are different for the
```

```
roaches taken to do so are different for the
# two techniques. Factor analysis is clearly designed with the objective to iden
tify certain unobservable factors from the
# observed variables, whereas PCA does not directly address this objectiveFactor
analysis is a statistical method used
# to describe variability among observed, correlated variables in terms of a pot
entially lower number of unobserved
# variables called factors. Factor analysis is related to principal component an
alysis (PCA), but the two are not identical.
# There has been significant controversy in the field over differences between t
he two techniques
# RandomProjection: In mathematics and statistics, random projection is a techni
que used to reduce the dimensionality
# of a set of points which lie in Euclidean space. Random projection methods are
powerful methods known for their simplicity
# and less erroneous output compared with other methods[citation needed]. Accord
ing to experimental results, random projection
# preserve distances well, but empirical results are sparse.
def dddraw(X_reduced,name):
    import matplotlib.pyplot as plt
    from mpl_toolkits.mplot3d import Axes3D
    # To getter a better understanding of interaction of the dimensions
    # plot the first three PCA dimensions
    fig = plt.figure(1, figsize=(8, 6))
    ax = Axes3D(fig, elev=-150, azim=110)
    ax.scatter(X_reduced[:, 0], X_reduced[:, 1], X_reduced[:, 2], c=Y,cmap=plt.c
m.Paired)
    titel="First three directions of "+name
    ax.set_title(titel)
    ax.set_xlabel("1st eigenvector")
    ax.w_xaxis.set_ticklabels([])
    ax.set_ylabel("2nd eigenvector")
    ax.w_yaxis.set_ticklabels([])
    ax.set_zlabel("3rd eigenvector")
    ax.w_zaxis.set_ticklabels([])

    plt.show()
```

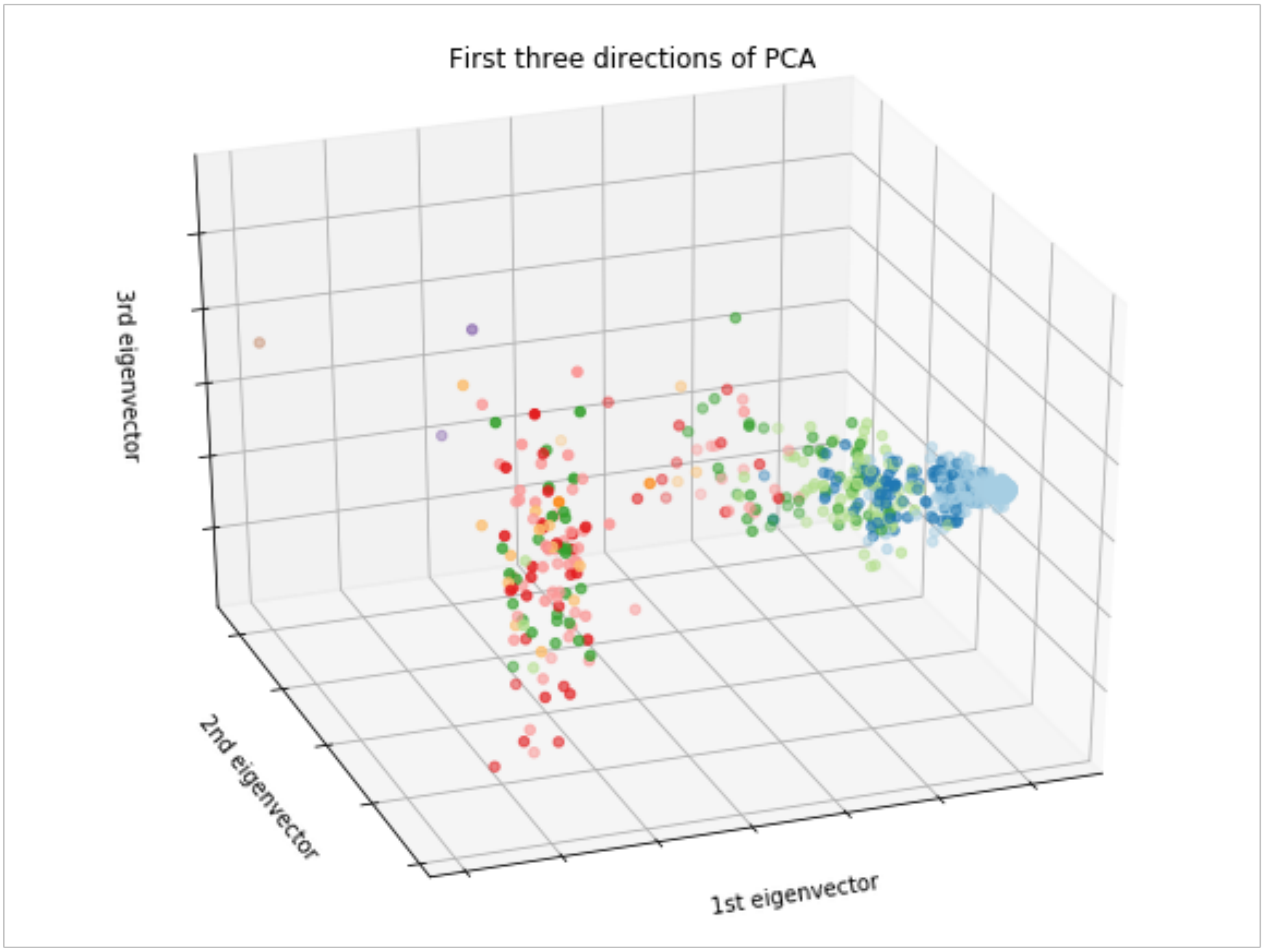
In [177]:

```
from sklearn.decomposition import PCA, FastICA,SparsePCA,NMF, LatentDirichletAll
ocation,FactorAnalysis
from sklearn.random_projection import GaussianRandomProjection,SparseRandomProje
ction
from sklearn.cluster import KMeans,Birch
import statsmodels.formula.api as sm
from scipy import linalg
from sklearn import preprocessing
from sklearn.preprocessing import MinMaxScaler,PolynomialFeatures
def rmsle(y_predicted, y_real):
    return np.sqrt(np.mean(np.power(np.log1p(y_predicted)-np.log1p(y_real), 2)))
def procenterror(y_predicted, y_real):
     return ( np.mean(np.abs(y_predicted-y_real) )/ np.mean(y_real) *100 ).round
()
```

```
n_col=50
X = sales.drop(['Product_Code','W25','Normalized 25'],axis=1)
def rmsle(y_predicted, y_real):
    return np.sqrt(np.mean(np.power(np.log1p(y_predicted)-np.log1p(y_real), 2)))
def procenterror(y_predicted, y_real):
     return np.round( np.mean(np.abs(y_predicted-y_real) )/ np.mean(y_real) *100
,1)


Y=sales['W25']
X=X.fillna(value=0)  #nasty NaN
#scaler = MinMaxScaler()
#scaler.fit(X)
#X=scaler.transform(X)
#poly = PolynomialFeatures(2)
#X=poly.fit_transform(X)


names = [
         'PCA',
         'FastICA',
         'Gauss',
         'KMeans',
         'SparsePCA',
         'SparseRP',
         'Birch',
         'NMF',
       #  'LatentDietrich',
        ]

classifiers = [

    PCA(n_components=n_col),
    FastICA(n_components=n_col),
    GaussianRandomProjection(n_components=3),
    KMeans(n_clusters=n_col),
    #SparsePCA(n_components=n_col),
    SparseRandomProjection(n_components=n_col, dense_output=True),
    Birch(branching_factor=10, n_clusters=7, threshold=0.5),
    NMF(n_components=n_col),
  #  LatentDirichletAllocation(n_topics=n_col),

]
correction= [1,1,0,0,0,0,0,0,0]

temp=zip(names,classifiers,correction)
print(temp)

for name, clf,correct in temp:
    Xr=clf.fit_transform(X,Y)
    dddraw(Xr,name)
    res = sm.OLS(Y,Xr).fit()
```

```
    #print(res.summary())  # show OLS regression

    #print(res.predict(Xr).round()+correct)  #show OLS prediction
    #print('Ypredict',res.predict(Xr).round()+correct)  #show OLS prediction

    print('Ypredict',res.predict(Xr).round()+correct*Y.mean())  #show OLS predic
tion
    print(name,'%error',procenterror(res.predict(Xr)+correct*Y.mean(),Y),'rmsle'
,rmsle(res.predict(Xr)+correct*Y.mean(),Y))
```

```
<zip object at 0x1c2d1254c8>
```




```
Ypredict [ 8.89272503  1.89272503  5.89272503  8.89272503 11.8927250
3  5.89272503
  3.89272503  7.89272503 12.89272503 22.89272503 13.89272503  1.8927
2503
  8.89272503 17.89272503 28.89272503 24.89272503 30.89272503 24.8927
2503
 21.89272503  8.89272503  6.89272503  4.89272503  2.89272503 21.8927
2503
 21.89272503 14.89272503 25.89272503 20.89272503 15.89272503 23.8927
2503
  4.89272503  4.89272503 10.89272503 25.89272503 24.89272503 26.8927
2503
 29.89272503 25.89272503 24.89272503 23.89272503 26.89272503 25.8927
2503
 31.89272503 21.89272503 21.89272503 23.89272503 26.89272503 27.8927
2503
```

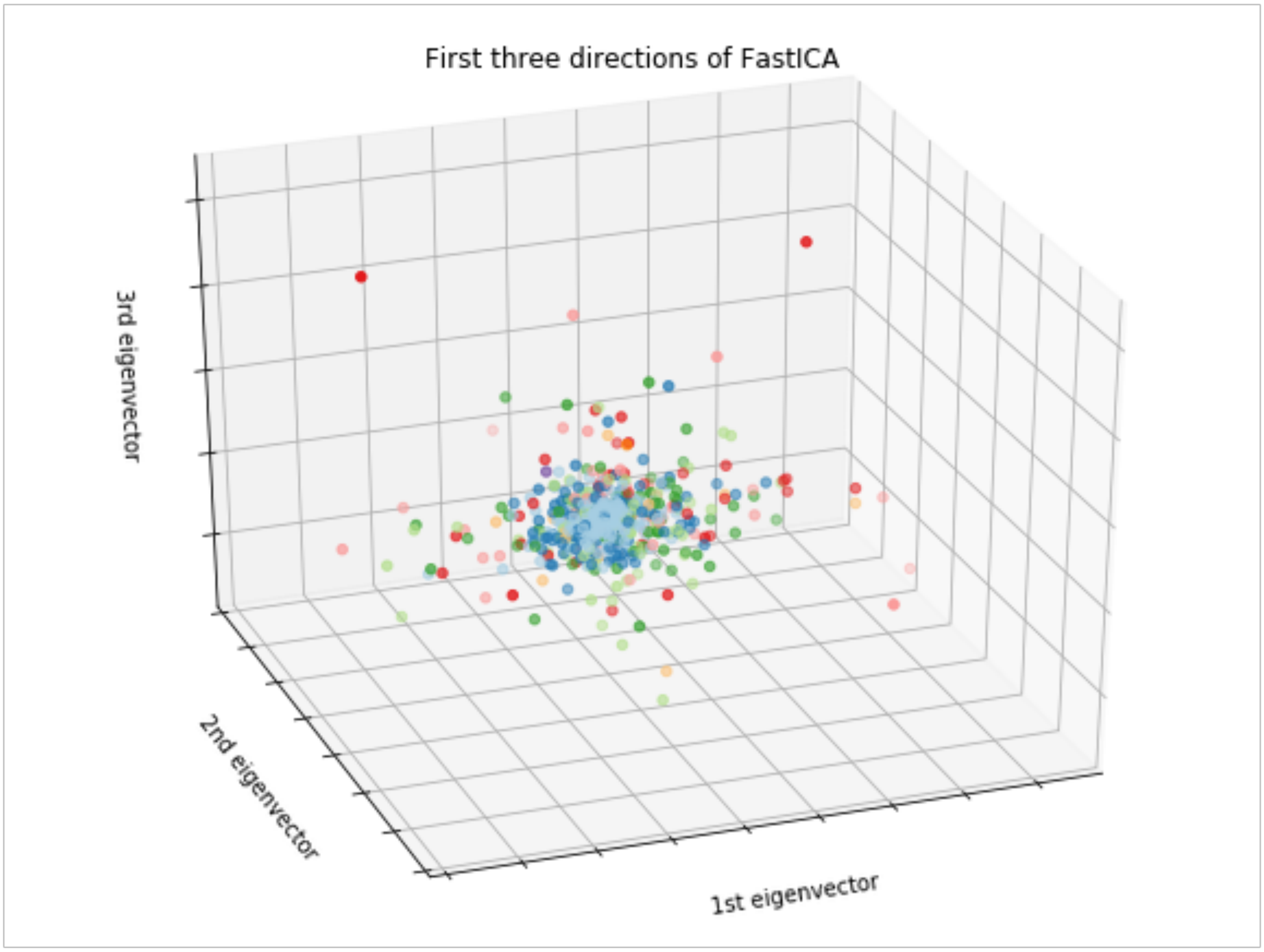
```
 26.89272503  8.89272503 10.89272503 30.89272503  5.89272503 26.8927
2503
 26.89272503 24.89272503 24.89272503 27.89272503  9.89272503 23.8927
2503
 22.89272503 11.89272503 27.89272503 23.89272503  7.89272503 24.8927
2503
 25.89272503  9.89272503 22.89272503 24.89272503  8.89272503 30.8927
2503
 25.89272503  9.89272503 27.89272503 23.89272503  5.89272503 24.8927
2503
 24.89272503 23.89272503 11.89272503  4.89272503 30.89272503 30.8927
2503
 22.89272503 23.89272503 21.89272503 22.89272503 25.89272503 18.8927
2503
  7.89272503 24.89272503  8.89272503  9.89272503 13.89272503 28.8927
2503
 27.89272503  5.89272503  9.89272503 19.89272503 25.89272503 28.8927
2503
  7.89272503  4.89272503  2.89272503 14.89272503 16.89272503  3.8927
2503
  7.89272503  7.89272503  5.89272503 23.89272503 21.89272503  5.8927
2503
  6.89272503 11.89272503  3.89272503  9.89272503 22.89272503 28.8927
2503
  8.89272503  5.89272503  2.89272503  5.89272503  8.89272503  5.8927
2503
  2.89272503 20.89272503 22.89272503 21.89272503 23.89272503 27.8927
2503
 24.89272503 26.89272503 22.89272503 22.89272503 29.89272503 20.8927
2503
 29.89272503 23.89272503 21.89272503 29.89272503 24.89272503 11.8927
2503
  7.89272503  7.89272503  3.89272503  5.89272503  4.89272503  4.8927
2503
  5.89272503 10.89272503  9.89272503  8.89272503  3.89272503  3.8927
2503
  6.89272503  4.89272503  5.89272503  7.89272503  4.89272503  9.8927
2503
  4.89272503  8.89272503  6.89272503 11.89272503 25.89272503 25.8927
2503
 25.89272503 24.89272503  7.89272503 27.89272503 27.89272503 27.8927
2503
 20.89272503 23.89272503 28.89272503 31.89272503 25.89272503 23.8927
2503
 20.89272503 26.89272503 21.89272503 28.89272503 26.89272503 23.8927
2503
 24.89272503 25.89272503 20.89272503 29.89272503 27.89272503 29.8927
2503
 30.89272503 28.89272503  3.89272503 23.89272503  7.89272503 14.8927
2503
  2.89272503 24.89272503  4.89272503 20.89272503  5.89272503  1.8927
2503
 16.89272503  4.89272503  9.89272503 42.89272503 13.89272503 15.8927
```

```
2503
 11.89272503  0.89272503  0.89272503  0.89272503  0.89272503  0.8927
2503
 -0.10727497  0.89272503  0.89272503 -0.10727497  0.89272503  0.8927
2503
  0.89272503  0.89272503  0.89272503  0.89272503  0.89272503 -0.1072
7497
  0.89272503 -0.10727497  0.89272503 -0.10727497 -0.10727497  0.8927
2503
  0.89272503 -0.10727497  0.89272503  0.89272503  0.89272503  1.8927
2503
  0.89272503  0.89272503 -0.10727497  0.89272503  0.89272503  0.8927
2503
  1.89272503  0.89272503 -0.10727497  0.89272503 -0.10727497  0.8927
2503
 -0.10727497  0.89272503  4.89272503  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503 23.89272503 50.89272503 26.89272503  9.8927
2503
  3.89272503 11.89272503  8.89272503 23.89272503 14.89272503 25.8927
2503
  4.89272503  0.89272503 -0.10727497  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503 -0.10727497  0.89272503  1.89272503  0.8927
2503
  0.89272503 19.89272503 11.89272503 27.89272503  1.89272503  0.8927
2503
  0.89272503  0.89272503  4.89272503  3.89272503  3.89272503 10.8927
2503
  5.89272503  4.89272503  4.89272503  5.89272503  9.89272503  3.8927
2503
  3.89272503  3.89272503  4.89272503  8.89272503  4.89272503  4.8927
2503
  4.89272503  2.89272503  8.89272503  3.89272503  4.89272503  5.8927
2503
  6.89272503 10.89272503  6.89272503  3.89272503  3.89272503  4.8927
2503
  8.89272503  5.89272503  2.89272503  3.89272503  2.89272503 10.8927
2503
  4.89272503  2.89272503  1.89272503  5.89272503  3.89272503  7.8927
2503
  6.89272503  8.89272503  7.89272503 13.89272503  3.89272503  4.8927
2503
  4.89272503  5.89272503  1.89272503  0.89272503  4.89272503  4.8927
2503
  4.89272503  1.89272503  4.89272503  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503  0.89272503  0.89272503 -0.10727497 -0.1072
7497
  5.89272503  3.89272503  2.89272503  6.89272503  1.89272503  3.8927
2503
  1.89272503  0.89272503 16.89272503  5.89272503  4.89272503  4.8927
2503
```

```
   8.89272503  6.89272503  5.89272503  3.89272503  3.89272503  1.8927
2503
   1.89272503  1.89272503  0.89272503 -0.10727497  0.89272503  0.8927
2503
   0.89272503 -0.10727497  0.89272503 -0.10727497  1.89272503 -0.1072
7497
   7.89272503  4.89272503  4.89272503  3.89272503  5.89272503  8.8927
2503
   2.89272503  2.89272503  3.89272503  3.89272503 12.89272503 15.8927
2503
  19.89272503 24.89272503  7.89272503  2.89272503 19.89272503 12.8927
2503
  26.89272503 20.89272503 16.89272503 17.89272503 59.89272503 25.8927
2503
  22.89272503  2.89272503 17.89272503  4.89272503  3.89272503  2.8927
2503
   0.89272503  0.89272503  0.89272503 -0.10727497 -0.10727497  2.8927
2503
   0.89272503  0.89272503 -0.10727497  0.89272503 -0.10727497  0.8927
2503
  13.89272503 15.89272503 10.89272503 14.89272503 10.89272503  5.8927
2503
  20.89272503 14.89272503  7.89272503  0.89272503  0.89272503  0.8927
2503
  -0.10727497 -0.10727497  0.89272503  0.89272503  0.89272503  0.8927
2503
   1.89272503  1.89272503  0.89272503  1.89272503  0.89272503  0.8927
2503
  -0.10727497  0.89272503  0.89272503  1.89272503  0.89272503  0.8927
2503
   0.89272503  0.89272503  0.89272503 -0.10727497  0.89272503  0.8927
2503
   0.89272503 -0.10727497 -0.10727497 -0.10727497 -0.10727497  0.8927
2503
  -0.10727497  0.89272503  0.89272503  0.89272503  2.89272503  1.8927
2503
   1.89272503  1.89272503  1.89272503  0.89272503  2.89272503  1.8927
2503
   0.89272503  3.89272503  9.89272503 15.89272503 13.89272503  8.8927
2503
   6.89272503  8.89272503 18.89272503 10.89272503 12.89272503 12.8927
2503
  21.89272503 12.89272503  4.89272503  3.89272503  7.89272503  5.8927
2503
   4.89272503 16.89272503 22.89272503 10.89272503 16.89272503 12.8927
2503
  25.89272503 11.89272503  6.89272503 13.89272503 25.89272503 27.8927
2503
  16.89272503 10.89272503 11.89272503 19.89272503 12.89272503 16.8927
2503
  17.89272503 16.89272503  8.89272503  8.89272503 12.89272503  9.8927
2503
  12.89272503 23.89272503 10.89272503 15.89272503 11.89272503 19.8927
```

```
2503
  4.89272503 11.89272503 29.89272503 11.89272503 27.89272503 17.8927
2503
 22.89272503 17.89272503 11.89272503 19.89272503 13.89272503 12.8927
2503
 11.89272503 11.89272503 21.89272503 13.89272503 13.89272503 26.8927
2503
 26.89272503  7.89272503  6.89272503  8.89272503  2.89272503 16.8927
2503
 10.89272503 24.89272503 26.89272503 13.89272503 12.89272503 10.8927
2503
  5.89272503  5.89272503 10.89272503  9.89272503  8.89272503 14.8927
2503
  1.89272503  4.89272503  3.89272503  4.89272503  8.89272503  3.8927
2503
  4.89272503 -0.10727497  0.89272503 -0.10727497  1.89272503  0.8927
2503
  0.89272503  5.89272503  4.89272503  1.89272503  3.89272503  4.8927
2503
  3.89272503  9.89272503  4.89272503  5.89272503  1.89272503  7.8927
2503
  5.89272503  3.89272503  2.89272503  5.89272503  4.89272503  8.8927
2503
  1.89272503 20.89272503  9.89272503  4.89272503  1.89272503  0.8927
2503
 -0.10727497  0.89272503  1.89272503  0.89272503  0.89272503  1.8927
2503
  7.89272503  3.89272503  6.89272503 14.89272503 29.89272503  7.8927
2503
 34.89272503  5.89272503 19.89272503 24.89272503 20.89272503 27.8927
2503
 30.89272503 26.89272503 25.89272503  6.89272503  9.89272503  9.8927
2503
  9.89272503  5.89272503  7.89272503  7.89272503  8.89272503 10.8927
2503
 11.89272503 11.89272503  7.89272503  8.89272503  4.89272503 20.8927
2503
  0.89272503 21.89272503  2.89272503  6.89272503  0.89272503 -0.1072
7497
  0.89272503  0.89272503 -0.10727497 -0.10727497  0.89272503 -0.1072
7497
  0.89272503 -0.10727497  0.89272503 -0.10727497  0.89272503  0.8927
2503
  0.89272503 -0.10727497 -0.10727497  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503  0.89272503  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503  4.89272503 11.89272503  5.89272503  6.8927
2503
 -0.10727497 -0.10727497  0.89272503 -0.10727497  0.89272503  0.8927
2503
  0.89272503  0.89272503  0.89272503  3.89272503  1.89272503  0.8927
2503
```

```
  1.89272503  0.89272503  0.89272503  4.89272503  4.89272503  0.8927
2503
  0.89272503  1.89272503  1.89272503  2.89272503  2.89272503  3.8927
2503
  5.89272503 10.89272503  4.89272503  0.89272503  3.89272503  2.8927
2503
  0.89272503 -0.10727497  0.89272503 -0.10727497  0.89272503 -0.1072
7497
 -0.10727497 -0.10727497  0.89272503  0.89272503 -0.10727497  0.8927
2503
 -0.10727497  0.89272503  0.89272503  1.89272503  4.89272503  1.8927
2503
  0.89272503 -0.10727497  0.89272503  3.89272503  1.89272503  0.8927
2503
  1.89272503  2.89272503  2.89272503  1.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503  1.89272503 -0.10727497  4.89272503  2.8927
2503
  3.89272503  2.89272503  0.89272503  0.89272503  0.89272503  3.8927
2503
  1.89272503  1.89272503  0.89272503  0.89272503  0.89272503  1.8927
2503
  0.89272503  0.89272503 -0.10727497  3.89272503 -0.10727497  8.8927
2503
  2.89272503  0.89272503  2.89272503  5.89272503  4.89272503  0.8927
2503
 -0.10727497  0.89272503  0.89272503 -0.10727497  0.89272503 -0.1072
7497
  1.89272503  0.89272503  1.89272503  0.89272503 22.89272503  3.8927
2503
 18.89272503  4.89272503  4.89272503  3.89272503  2.89272503  7.8927
2503
  6.89272503  3.89272503 11.89272503  1.89272503  6.89272503  1.8927
2503
  0.89272503  4.89272503  4.89272503  4.89272503  9.89272503  1.8927
2503
  5.89272503  1.89272503  6.89272503  4.89272503  2.89272503  7.8927
2503
  3.89272503  7.89272503  1.89272503  0.89272503  3.89272503  1.8927
2503
  4.89272503 10.89272503  1.89272503  3.89272503  1.89272503  0.8927
2503
  0.89272503]
PCA %error 24.5 rmsle 0.42374704359253124
```




```
Ypredict [ 8.89272503  1.89272503  5.89272503  8.89272503 11.8927250
3  5.89272503
  3.89272503  7.89272503 12.89272503 22.89272503 13.89272503  1.8927
2503
  8.89272503 17.89272503 28.89272503 24.89272503 30.89272503 24.8927
2503
 21.89272503  8.89272503  6.89272503  4.89272503  2.89272503 21.8927
2503
 21.89272503 14.89272503 25.89272503 20.89272503 15.89272503 23.8927
2503
  4.89272503  4.89272503 10.89272503 25.89272503 24.89272503 26.8927
2503
 29.89272503 25.89272503 24.89272503 23.89272503 26.89272503 25.8927
2503
 31.89272503 21.89272503 21.89272503 23.89272503 26.89272503 27.8927
2503
 26.89272503  8.89272503 10.89272503 30.89272503  5.89272503 26.8927
2503
 26.89272503 24.89272503 24.89272503 27.89272503  9.89272503 23.8927
2503
 22.89272503 11.89272503 27.89272503 23.89272503  7.89272503 24.8927
2503
 25.89272503  9.89272503 22.89272503 24.89272503  8.89272503 30.8927
2503
 25.89272503  9.89272503 27.89272503 23.89272503  5.89272503 24.8927
2503
 24.89272503 23.89272503 11.89272503  4.89272503 30.89272503 30.8927
```

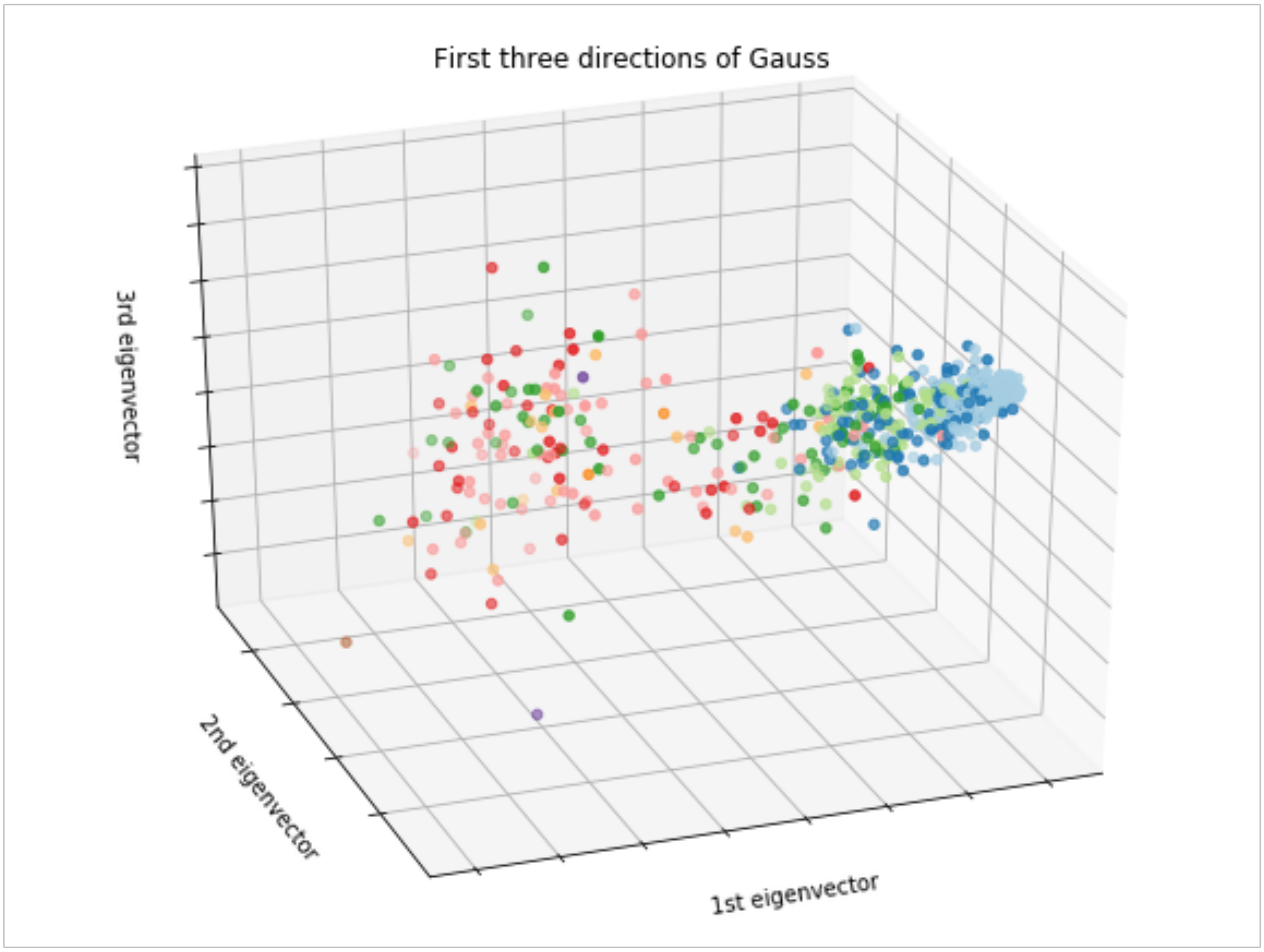
```
2503
 22.89272503 23.89272503 21.89272503 22.89272503 25.89272503 18.8927
2503
  7.89272503 24.89272503  8.89272503  9.89272503 13.89272503 28.8927
2503
 27.89272503  5.89272503  9.89272503 19.89272503 25.89272503 28.8927
2503
  7.89272503  4.89272503  2.89272503 14.89272503 16.89272503  3.8927
2503
  7.89272503  7.89272503  5.89272503 23.89272503 21.89272503  5.8927
2503
  6.89272503 11.89272503  3.89272503  9.89272503 22.89272503 28.8927
2503
  8.89272503  5.89272503  2.89272503  5.89272503  8.89272503  5.8927
2503
  2.89272503 20.89272503 22.89272503 21.89272503 23.89272503 27.8927
2503
 24.89272503 26.89272503 22.89272503 22.89272503 29.89272503 20.8927
2503
 29.89272503 23.89272503 21.89272503 29.89272503 24.89272503 11.8927
2503
  7.89272503  7.89272503  3.89272503  5.89272503  4.89272503  4.8927
2503
  5.89272503 10.89272503  9.89272503  8.89272503  3.89272503  3.8927
2503
  6.89272503  4.89272503  5.89272503  7.89272503  4.89272503  9.8927
2503
  4.89272503  8.89272503  6.89272503 11.89272503 25.89272503 25.8927
2503
 25.89272503 24.89272503  7.89272503 27.89272503 27.89272503 27.8927
2503
 20.89272503 23.89272503 28.89272503 31.89272503 25.89272503 23.8927
2503
 20.89272503 26.89272503 21.89272503 28.89272503 26.89272503 23.8927
2503
 24.89272503 25.89272503 20.89272503 29.89272503 27.89272503 29.8927
2503
 30.89272503 28.89272503  3.89272503 23.89272503  7.89272503 14.8927
2503
  2.89272503 24.89272503  4.89272503 20.89272503  5.89272503  1.8927
2503
 16.89272503  4.89272503  9.89272503 42.89272503 13.89272503 15.8927
2503
 11.89272503  0.89272503  0.89272503  0.89272503  0.89272503  0.8927
2503
 -0.10727497  0.89272503  0.89272503 -0.10727497  0.89272503  0.8927
2503
  0.89272503  0.89272503  0.89272503  0.89272503  0.89272503 -0.1072
7497
  0.89272503 -0.10727497  0.89272503 -0.10727497 -0.10727497  0.8927
2503
  0.89272503 -0.10727497  0.89272503  0.89272503  0.89272503  1.8927
2503
```

```
  0.89272503  0.89272503 -0.10727497  0.89272503  0.89272503  0.8927
2503
  1.89272503  0.89272503 -0.10727497  0.89272503 -0.10727497  0.8927
2503
 -0.10727497  0.89272503  4.89272503  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503 23.89272503 50.89272503 26.89272503  9.8927
2503
  3.89272503 11.89272503  8.89272503 23.89272503 14.89272503 25.8927
2503
  4.89272503  0.89272503 -0.10727497  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503 -0.10727497  0.89272503  1.89272503  0.8927
2503
  0.89272503 19.89272503 11.89272503 27.89272503  1.89272503  0.8927
2503
  0.89272503  0.89272503  4.89272503  3.89272503  3.89272503 10.8927
2503
  5.89272503  4.89272503  4.89272503  5.89272503  9.89272503  3.8927
2503
  3.89272503  3.89272503  4.89272503  8.89272503  4.89272503  4.8927
2503
  4.89272503  2.89272503  8.89272503  3.89272503  4.89272503  5.8927
2503
  6.89272503 10.89272503  6.89272503  3.89272503  3.89272503  4.8927
2503
  8.89272503  5.89272503  2.89272503  3.89272503  2.89272503 10.8927
2503
  4.89272503  2.89272503  1.89272503  5.89272503  3.89272503  7.8927
2503
  6.89272503  8.89272503  7.89272503 13.89272503  3.89272503  4.8927
2503
  4.89272503  5.89272503  1.89272503  0.89272503  4.89272503  4.8927
2503
  4.89272503  1.89272503  4.89272503  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503  0.89272503  0.89272503 -0.10727497 -0.1072
7497
  5.89272503  3.89272503  2.89272503  6.89272503  1.89272503  3.8927
2503
  1.89272503  0.89272503 16.89272503  5.89272503  4.89272503  4.8927
2503
  8.89272503  6.89272503  5.89272503  3.89272503  3.89272503  1.8927
2503
  1.89272503  1.89272503  0.89272503 -0.10727497  0.89272503  0.8927
2503
  0.89272503 -0.10727497  0.89272503 -0.10727497  1.89272503 -0.1072
7497
  7.89272503  4.89272503  4.89272503  3.89272503  5.89272503  8.8927
2503
  2.89272503  2.89272503  3.89272503  3.89272503 12.89272503 15.8927
2503
 19.89272503 24.89272503  7.89272503  2.89272503 19.89272503 12.8927
```

```
2503
 26.89272503 20.89272503 16.89272503 17.89272503 59.89272503 25.8927
2503
 22.89272503  2.89272503 17.89272503  4.89272503  3.89272503  2.8927
2503
  0.89272503  0.89272503  0.89272503 -0.10727497 -0.10727497  2.8927
2503
  0.89272503  0.89272503 -0.10727497  0.89272503 -0.10727497  0.8927
2503
 13.89272503 15.89272503 10.89272503 14.89272503 10.89272503  5.8927
2503
 20.89272503 14.89272503  7.89272503  0.89272503  0.89272503  0.8927
2503
 -0.10727497 -0.10727497  0.89272503  0.89272503  0.89272503  0.8927
2503
  1.89272503  1.89272503  0.89272503  1.89272503  0.89272503  0.8927
2503
 -0.10727497  0.89272503  0.89272503  1.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503  0.89272503 -0.10727497  0.89272503  0.8927
2503
  0.89272503 -0.10727497 -0.10727497 -0.10727497 -0.10727497  0.8927
2503
 -0.10727497  0.89272503  0.89272503  0.89272503  2.89272503  1.8927
2503
  1.89272503  1.89272503  1.89272503  0.89272503  2.89272503  1.8927
2503
  0.89272503  3.89272503  9.89272503 15.89272503 13.89272503  8.8927
2503
  6.89272503  8.89272503 18.89272503 10.89272503 12.89272503 12.8927
2503
 21.89272503 12.89272503  4.89272503  3.89272503  7.89272503  5.8927
2503
  4.89272503 16.89272503 22.89272503 10.89272503 16.89272503 12.8927
2503
 25.89272503 11.89272503  6.89272503 13.89272503 25.89272503 27.8927
2503
 16.89272503 10.89272503 11.89272503 19.89272503 12.89272503 16.8927
2503
 17.89272503 16.89272503  8.89272503  8.89272503 12.89272503  9.8927
2503
 12.89272503 23.89272503 10.89272503 15.89272503 11.89272503 19.8927
2503
  4.89272503 11.89272503 29.89272503 11.89272503 27.89272503 17.8927
2503
 22.89272503 17.89272503 11.89272503 19.89272503 13.89272503 12.8927
2503
 11.89272503 11.89272503 21.89272503 13.89272503 13.89272503 26.8927
2503
 26.89272503  7.89272503  6.89272503  8.89272503  2.89272503 16.8927
2503
 10.89272503 24.89272503 26.89272503 13.89272503 12.89272503 10.8927
2503
```

```
  5.89272503  5.89272503 10.89272503  9.89272503  8.89272503 14.8927
2503
  1.89272503  4.89272503  3.89272503  4.89272503  8.89272503  3.8927
2503
  4.89272503 -0.10727497  0.89272503 -0.10727497  1.89272503  0.8927
2503
  0.89272503  5.89272503  4.89272503  1.89272503  3.89272503  4.8927
2503
  3.89272503  9.89272503  4.89272503  5.89272503  1.89272503  7.8927
2503
  5.89272503  3.89272503  2.89272503  5.89272503  4.89272503  8.8927
2503
  1.89272503 20.89272503  9.89272503  4.89272503  1.89272503  0.8927
2503
 -0.10727497  0.89272503  1.89272503  0.89272503  0.89272503  1.8927
2503
  7.89272503  3.89272503  6.89272503 14.89272503 29.89272503  7.8927
2503
 34.89272503  5.89272503 19.89272503 24.89272503 20.89272503 27.8927
2503
 30.89272503 26.89272503 25.89272503  6.89272503  9.89272503  9.8927
2503
  9.89272503  5.89272503  7.89272503  7.89272503  8.89272503 10.8927
2503
 11.89272503 11.89272503  7.89272503  8.89272503  4.89272503 20.8927
2503
  0.89272503 21.89272503  2.89272503  6.89272503  0.89272503 -0.1072
7497
  0.89272503  0.89272503 -0.10727497 -0.10727497  0.89272503 -0.1072
7497
  0.89272503 -0.10727497  0.89272503 -0.10727497  0.89272503  0.8927
2503
  0.89272503 -0.10727497 -0.10727497  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503  0.89272503  0.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503  4.89272503 11.89272503  5.89272503  6.8927
2503
 -0.10727497 -0.10727497  0.89272503 -0.10727497  0.89272503  0.8927
2503
  0.89272503  0.89272503  0.89272503  3.89272503  1.89272503  0.8927
2503
  1.89272503  0.89272503  0.89272503  4.89272503  4.89272503  0.8927
2503
  0.89272503  1.89272503  1.89272503  2.89272503  2.89272503  3.8927
2503
  5.89272503 10.89272503  4.89272503  0.89272503  3.89272503  2.8927
2503
  0.89272503 -0.10727497  0.89272503 -0.10727497  0.89272503 -0.1072
7497
 -0.10727497 -0.10727497  0.89272503  0.89272503 -0.10727497  0.8927
2503
 -0.10727497  0.89272503  0.89272503  1.89272503  4.89272503  1.8927
```

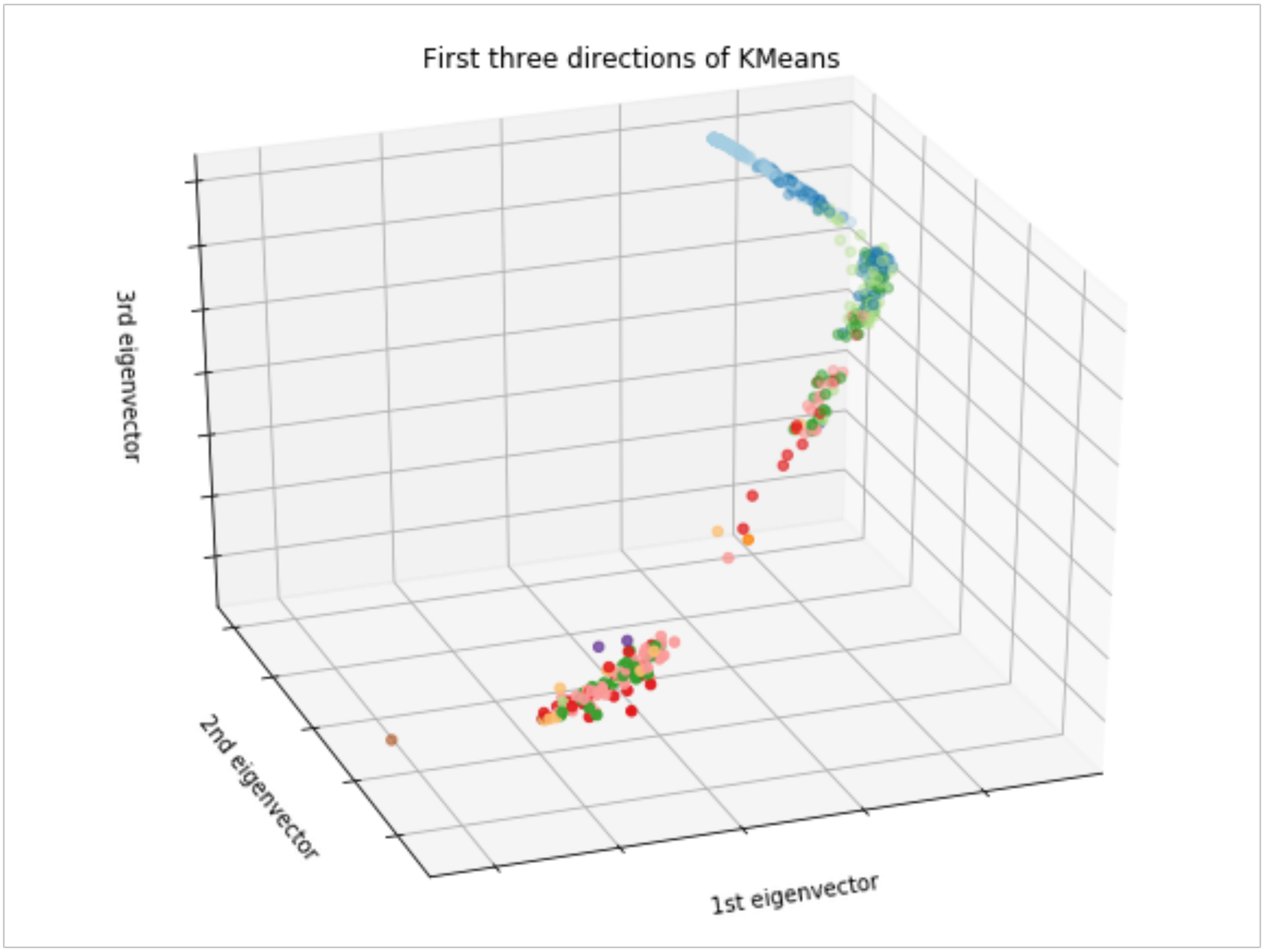
```
2503
  0.89272503 -0.10727497  0.89272503  3.89272503  1.89272503  0.8927
2503
  1.89272503  2.89272503  2.89272503  1.89272503  0.89272503  0.8927
2503
  0.89272503  0.89272503  1.89272503 -0.10727497  4.89272503  2.8927
2503
  3.89272503  2.89272503  0.89272503  0.89272503  0.89272503  3.8927
2503
  1.89272503  1.89272503  0.89272503  0.89272503  0.89272503  1.8927
2503
  0.89272503  0.89272503 -0.10727497  3.89272503 -0.10727497  8.8927
2503
  2.89272503  0.89272503  2.89272503  5.89272503  4.89272503  0.8927
2503
 -0.10727497  0.89272503  0.89272503 -0.10727497  0.89272503 -0.1072
7497
  1.89272503  0.89272503  1.89272503  0.89272503 22.89272503  3.8927
2503
 18.89272503  4.89272503  4.89272503  3.89272503  2.89272503  7.8927
2503
  6.89272503  3.89272503 11.89272503  1.89272503  6.89272503  1.8927
2503
  0.89272503  4.89272503  4.89272503  4.89272503  9.89272503  1.8927
2503
  5.89272503  1.89272503  6.89272503  4.89272503  2.89272503  7.8927
2503
  3.89272503  7.89272503  1.89272503  0.89272503  3.89272503  1.8927
2503
  4.89272503 10.89272503  1.89272503  3.89272503  1.89272503  0.8927
2503
  0.89272503]
FastICA %error 24.5 rmsle 0.4237470435928992
```




```
Ypredict [ 8.  4. 10.  5.  9.  2.  6.  8. 10. 15. 12.  2.  7.  8. 25
. 29. 27. 25.
 26.  9.  5.  9.  4. 25. 23.  8. 26. 30. 10. 37.  5.  6.  7. 36. 33.
28.
 30. 32. 24. 30. 32. 27. 36. 30. 15. 28. 27. 24. 30. 10. 17. 26.  4.
35.
 27. 26. 38. 27.  7. 32. 31. 17. 27. 24. 11. 38. 33.  5. 28. 28. 12.
29.
 27.  3. 24. 17.  3. 27. 24. 25.  9.  5. 28. 21. 25. 26. 29. 28. 27.
25.
  6. 24.  7. 10. 16. 26. 23.  3.  9. 11. 29. 31.  8.  3.  5. 11. 17.
0.
  6.  7.  5. 39. 22.  7.  9.  9.  5. 11. 21. 32.  7.  9.  4.  6.  5.
2.
  1. 30. 27. 28. 31. 31. 30. 22. 35. 31. 29. 24. 29. 28. 26. 23. 27.
9.
  9. 10. 10.  3.  7.  5.  4.  7.  8.  7.  4.  2.  5.  2.  2.  4.  5.
11.
  3. 10.  5.  8. 28. 27. 23. 21.  7. 32. 29. 28. 31. 29. 23. 36. 35.
30.
 24. 25. 22. 31. 34. 30. 30. 26. 22. 36. 26. 26. 30. 28.  3. 30.  9.
9.
  3. 14.  3. 13.  4.  2. 15.  3.  5. 37. 13. 10. 14.  0.  0.  0.  0.
0.
  0.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.
  1.  0.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
```

```
0.
  0.  0.  0.  0.  0.  0.  0.  0. 21. 24. 11. 10.  2.  7.  6. 13. 10.
20.
  2.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.  1. -1. 16. 10. 16.  0.
1.
  0.  0.  3.  3.  3.  5.  4.  5.  2.  4. 10.  5.  3.  2.  2.  7.  6.
6.
  4.  3.  8.  2.  5.  4.  4.  9.  3.  2.  6.  3. 11.  4.  0.  4.  3.
6.
  4.  5.  1.  2.  1.  3.  7.  8.  8.  9.  3.  1.  1.  3.  2.  1.  3.
3.
  1.  2.  5.  0.  0.  1.  0. -1.  0.  0.  0.  0.  6.  1.  0.  7.  1.
2.
  1.  0. 10.  2.  3.  0.  3.  5.  6.  1.  1.  1.  0.  0.  1.  1. -1.
0.
  0.  0.  0.  0.  0.  0.  5.  2.  3.  2.  4.  3.  1.  2.  3.  2.  8.
9.
 11. 12.  2.  1. 12. 11. 20.  8. 14. 12. 45. 14. 17.  4.  9.  4.  4.
1.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  6.  9.  5.  3.  3.
3.
 14. 11.  4.  0.  0.  1.  0.  0.  0.  0. -1.  0.  0.  1.  0.  1.  0.
0.
  0.  0. -1.  0.  0.  0.  0.  0. -1.  0.  1. -1.  1.  0. -1.  0.  0.
0.
  0.  0.  0.  0.  3.  0.  0.  1.  0.  0.  1.  0.  1.  3. 11. 19. 11.
12.
  4.  6. 19.  3.  7. 10. 11.  6.  0.  4. 11.  9.  7. 11. 10. 11. 12.
11.
 14.  8.  6.  6. 17. 18. 16. 11. 10. 17.  9. 15. 22. 10.  6. 11. 12.
6.
 11. 12.  9.  9. 14. 14.  1. 10. 21.  8. 17.  8. 15. 11. 10. 17. 13.
8.
 10.  8. 18.  9.  9. 23. 26.  9. 12.  9.  2. 16. 11. 13. 19. 11. 14.
11.
  2.  0.  7.  9. 10. 16.  2.  2.  1.  1.  4.  1.  4.  1.  0.  0.  0.
0.
  1.  2.  2.  2.  3.  4.  1.  7.  2.  1.  1.  4.  2.  2.  1.  2. -1.
3.
  0.  8.  3.  0.  1.  1.  0.  1.  2. -1.  0.  0.  5.  4.  4. 10. 22.
4.
 22.  3. 21. 31. 27. 25. 23. 24. 37.  7.  9.  6.  8.  8.  9.  8. 11.
7.
  8. 12.  9. 14.  4.  9.  0. 11.  4.  2.  0.  0.  0.  0.  0.  0. -1.
0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
1.
 -1.  0.  2.  3.  4.  5.  0.  0. -1.  0.  0.  0.  0.  0.  0.  3.  1.
0.
  0.  0.  0.  0.  2.  0.  0.  0.  0.  1.  1.  0.  4.  7.  2.  0.  2.
1.
  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  3.
0.
```

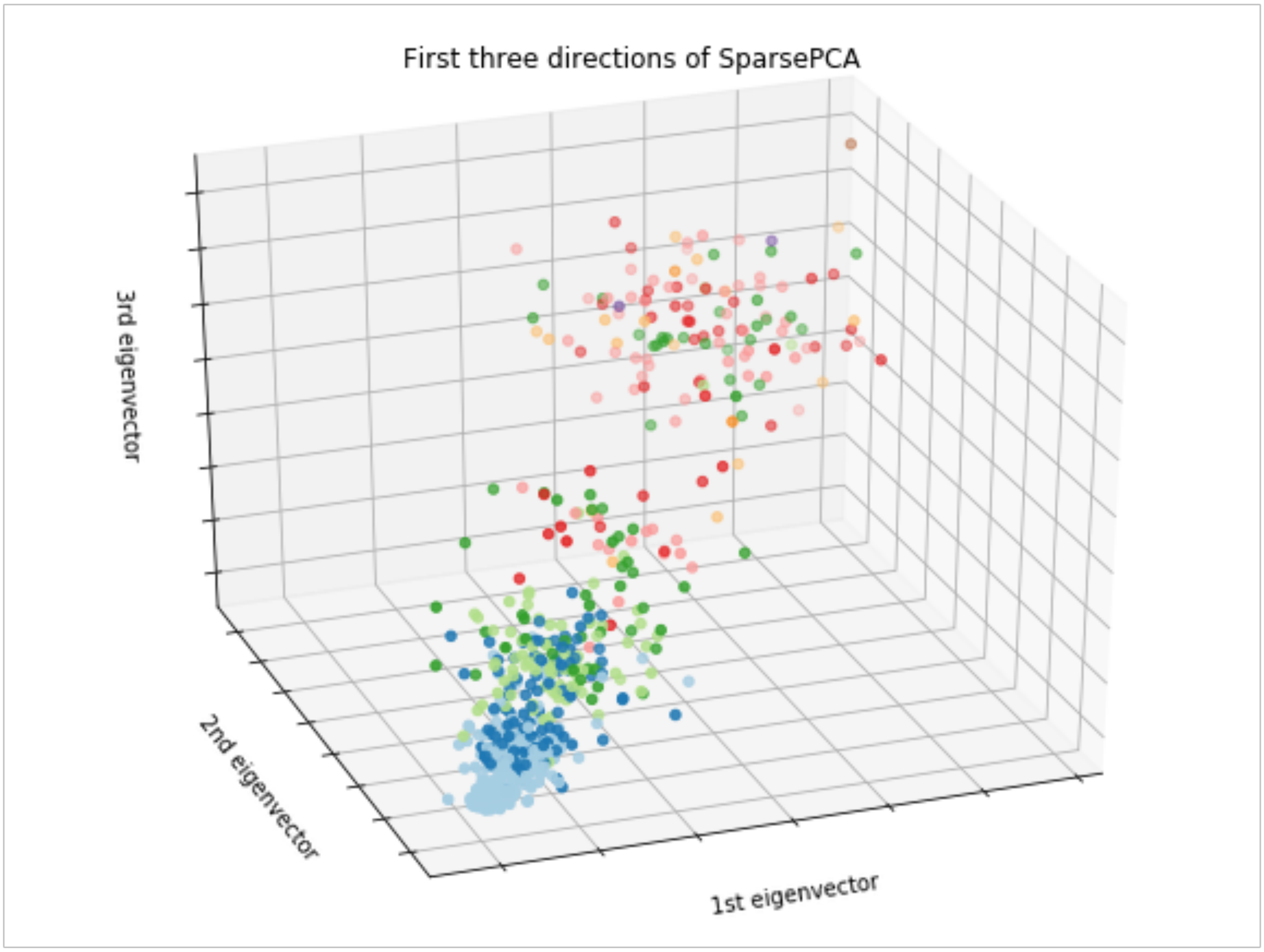
```
  0.   0.   0.   0.   0.   0.  -1.   0.   1.   0.   0.   1.   0.   1.   1.   0.   2.
0.
  2.   1.   1.   0.  -1.   3.   1.   1.   1.   0.   1.   0.   0.   0.   0.   1.   0.
2.
  0.   0.   0.   4.   2.   0.  -1.   0.   0.   0.   0.   0.   0.   0.  -1.   0.  15.
3.
 15.   3.   1.   1.   0.   3.   1.   2.   6.   2.   1.   0.   1.   2.   1.   1.   3.
0.
  2.   1.   4.   4.   3.   5.   0.   1.   1.   0.   3.   1.   4.   4.   0.   1.   1.
0.
  0.]
Gauss %error 36.5 rmsle 0.6227288725192254

/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:16: Run
timeWarning: invalid value encountered in log1p
  app.launch_new_instance()
```




```
Ypredict [ 9.   4.   8.  10.  11.   6.   5.   9.  13.  21.  13.   4.  12.  16.  25
.  26.  28.  24.
 20.   9.  11.   9.   4.  27.  22.  13.  21.  18.  13.  25.   7.   9.  12.  27.  26.
25.
 25.  29.  27.  25.  27.  26.  30.  25.  22.  24.  25.  24.  26.  10.  18.  27.   6.
20.
 23.  25.  25.  22.   9.  24.  25.  17.  29.  23.   9.  26.  26.  10.  25.  26.  10.
30.
 22.  12.  25.  22.   6.  23.  23.  24.  13.   8.  24.  25.  27.  22.  24.  21.  25.
18.
  9.  25.   9.  11.  14.  29.  26.   6.  10.  20.  26.  24.   9.   4.   4.  13.  19.
```

```
4.
 11.  9.  7. 26. 22. 10. 10. 12.  5. 11. 22. 29. 11.  9.  3.  6. 10.
5.
  3. 22. 24. 19. 26. 26. 23. 32. 25. 26. 28. 27. 27. 24. 24. 32. 25.
11.
 10.  9.  8.  5.  9.  5.  6. 12. 11.  8.  4.  4.  8.  5.  5. 10.  5.
10.
  5. 10.  9. 11. 29. 24. 20. 25.  9. 28. 28. 25. 26. 23. 31. 29. 26.
19.
 18. 25. 24. 21. 24. 25. 26. 27. 23. 29. 27. 29. 29. 26.  5. 25. 11.
14.
  4. 25.  5. 26.  7.  3. 19.  4.  9. 38. 14. 17. 13.  1.  0.  1.  0.
0.
  0.  0.  0.  0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.
  0.  0.  0.  0.  0.  1.  0.  1.  0.  0.  1.  0.  1.  0.  0.  0.  0.
1.
  0.  0.  3.  0.  1.  0.  0.  0. 28. 50. 27. 11.  3. 12.  9. 26. 17.
26.
  4.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  1. 22. 12. 27.  2.
1.
  1.  0.  5.  4.  5. 11.  6.  5.  5.  5. 11.  5.  5.  5.  5. 10.  6.
5.
  5.  4. 12.  5.  5.  6.  6. 11.  6.  4.  5.  4. 10.  6.  4.  4.  4.
12.
  5.  4.  4.  5.  5.  6.  6.  9. 12. 12.  4.  5.  5.  4.  2.  1.  5.
4.
  4.  2.  4.  1.  0.  0.  1.  0.  0.  0.  0.  0.  6.  3.  3.  7.  1.
4.
  2.  0. 17.  5.  5.  4.  9.  6.  5.  3.  4.  2.  2.  1.  1.  1.  0.
0.
  0.  0.  0.  0.  1.  0.  5.  4.  4.  3.  5.  8.  3.  4.  4.  3. 12.
16.
 21. 26.  5.  4. 24. 13. 26. 21. 22. 19. 63. 26. 24.  4. 16.  4.  4.
2.
  1.  0.  0.  0.  0.  2.  1.  1.  0.  0.  0.  0. 14. 18. 10. 14.  8.
4.
 20. 15.  9.  1.  1.  1.  0.  1.  1.  0.  0.  0.  1.  1.  1.  1.  0.
1.
  0.  0.  1.  1.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.
  0.  0.  0.  0.  3.  2.  1.  2.  2.  1.  2.  1.  1.  6. 10. 19. 13.
11.
  6. 12. 21. 13. 12. 15. 22. 12.  5.  4. 10.  8.  7. 15. 24. 14. 20.
15.
 24. 11.  8. 12. 29. 29. 20. 13. 10. 21. 13. 13. 23. 15. 11. 10. 12.
11.
 13. 23. 11. 15. 13. 25.  5. 12. 33. 13. 27. 16. 23. 19. 12. 23. 12.
13.
 13. 13. 24. 14. 13. 25. 23. 10. 10.  9.  4. 19. 13. 22. 28. 14. 13.
12.
  5.  6. 12. 12. 10. 17.  4.  5.  3.  4.  8.  3.  5.  1.  1.  0.  1.
0.
```

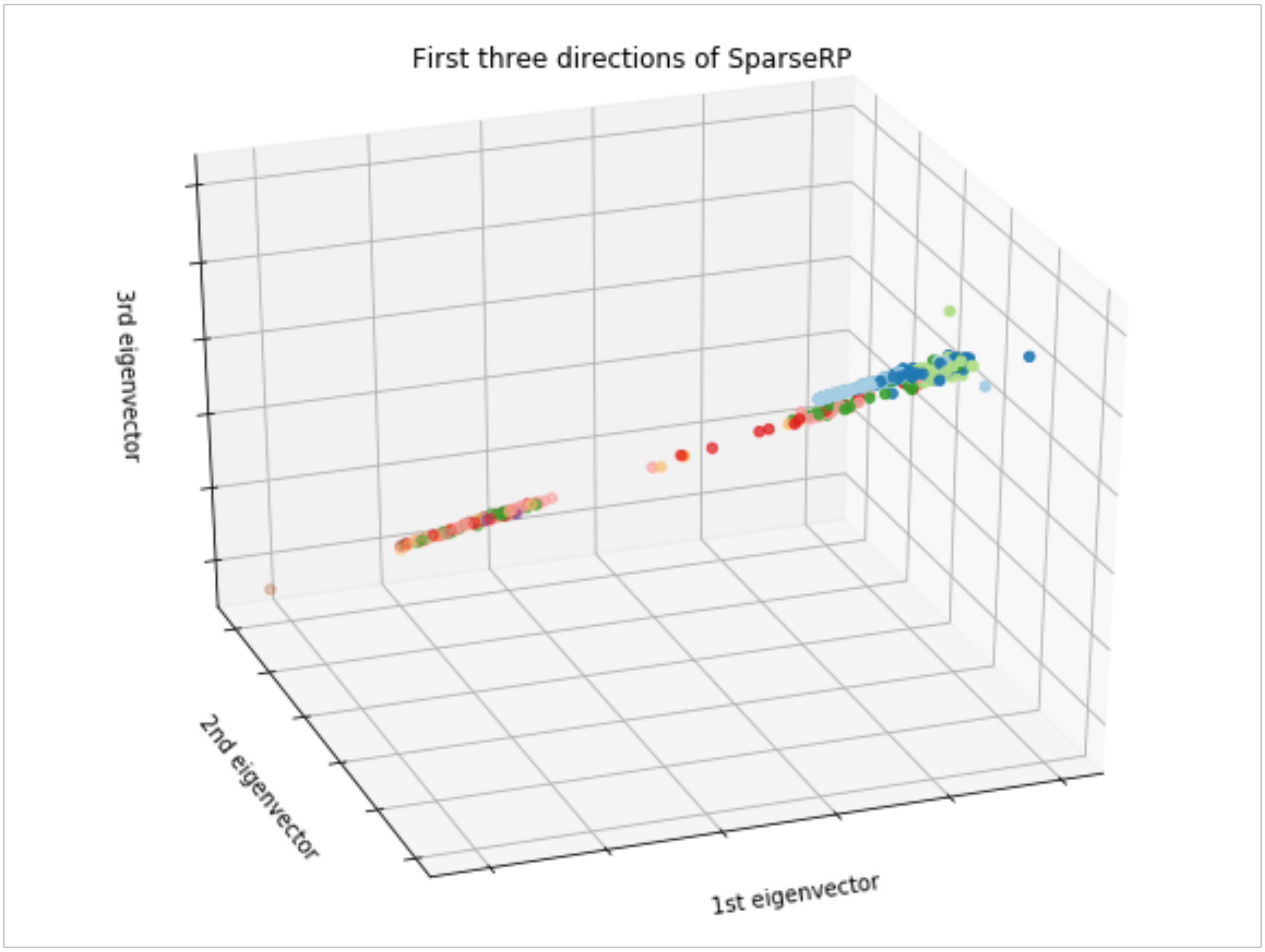
```
  0.  4.  4.  3.  4.  4.  2. 10.  5.  4.  2.  6.  4.  3.  4.  5.  5.
9.
  2. 20.  8.  5.  3.  1.  0.  0.  2.  1.  0.  1.  6.  5.  5. 18. 32.
8.
 39.  7. 20. 28. 22. 25. 29. 31. 25.  8. 10. 11. 12.  9. 11. 10. 10.
11.
 12. 12.  9. 12.  5. 21.  0. 22.  3.  5.  0.  0.  0.  0.  0.  0.  1.
0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  0.  0.  0.  0.  0.
1.
  1.  0.  5. 12.  7.  6.  0.  0.  0.  0.  0.  0.  0.  0.  0.  4.  1.
1.
  1.  1.  0.  4.  4.  1.  1.  1.  1.  3.  2.  2.  5. 10.  5.  1.  4.
2.
  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  5.
1.
  1.  0.  1.  3.  1.  0.  1.  2.  4.  1.  0.  0.  0.  1.  2.  0.  7.
2.
  5.  2.  1.  1.  1.  3.  1.  2.  1.  1.  1.  1.  1.  0.  0.  3.  0.
6.
  2.  1.  3.  7.  3.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1. 25.
4.
 20.  5.  3.  3.  2.  5.  6.  4. 10.  2.  5.  2.  1.  5.  4.  4.  8.
2.
  5.  1.  5.  4.  4.  8.  3.  6.  1.  0.  4.  2.  4.  9.  1.  5.  1.
0.
  0.]
KMeans %error 22.9 rmsle 0.3726247200598251
```




```
Ypredict [ 8.  1.  6.  9. 11.  6.  4.  8. 14. 23. 14.  3.  9. 17. 29
. 25. 31. 26.
 23.  9.  8.  4.  4. 21. 22. 15. 28. 22. 15. 24.  4.  5. 11. 25. 25.
26.
 30. 25. 24. 25. 27. 26. 32. 21. 22. 23. 26. 26. 28.  9. 11. 31.  7.
26.
 27. 25. 25. 28. 10. 24. 21. 12. 27. 23.  7. 26. 27. 10. 22. 26.  9.
30.
 25. 10. 27. 24.  6. 25. 25. 23. 12.  5. 31. 31. 22. 24. 23. 23. 26.
18.
  8. 25.  8. 10. 14. 30. 29.  6. 10. 19. 25. 30.  9.  5.  3. 15. 17.
5.
  7.  7.  6. 24. 22.  6.  8. 12.  4. 10. 23. 28.  8.  7.  4.  5.  9.
6.
  3. 20. 23. 21. 24. 28. 25. 28. 22. 21. 31. 21. 30. 24. 23. 30. 25.
11.
  8.  8.  4.  6.  7.  6.  6. 11. 10.  8.  5.  4.  8.  5.  5.  8.  5.
10.
  4.  9.  7. 11. 26. 26. 26. 25.  8. 26. 28. 28. 21. 22. 29. 31. 25.
23.
 20. 29. 23. 29. 27. 24. 24. 25. 20. 29. 28. 31. 30. 28.  4. 23.  9.
15.
  5. 24.  6. 21.  6.  4. 17.  6. 11. 42. 14. 16. 12.  1.  0.  0.  0.
0.
  0.  0.  1. -1.  1.  1.  1.  1.  0.  1.  1.  0.  0.  0.  0.  0.  1.
1.
  1.  2.  0.  2.  0.  1.  1.  2.  0.  1.  1.  1.  2.  2.  1.  1.  0.
```

```
1.
  0.  0.  4.  1.  1.  0.  0.  0. 23. 49. 27. 11.  4. 12.  8. 24. 15.
25.
  5.  1.  0.  1.  0.  0.  1.  1.  0.  1.  2.  1.  0. 21. 13. 27.  2.
2.
  1.  1.  5.  4.  4. 11.  6.  4.  5.  6.  9.  3.  4.  5.  5. 10.  6.
6.
  5.  3.  9.  4.  5.  6.  7. 11.  7.  5.  5.  6. 10.  6.  2.  5.  3.
11.
  5.  3.  2.  6.  4.  8.  7.  9.  8. 12.  4.  5.  5.  6.  2.  2.  5.
5.
  5.  4.  5.  1.  2.  2.  1.  0.  1.  0.  1.  0.  7.  5.  3.  8.  3.
5.
  2.  0. 17.  6.  5.  5. 10.  6.  6.  5.  4.  2.  2.  2.  1. -1.  0.
0.
  1.  1.  1.  0.  2.  0.  7.  5.  5.  4.  6.  9.  3.  4.  5.  4. 13.
16.
 20. 25.  8.  3. 19. 13. 26. 21. 17. 19. 62. 26. 22.  4. 17.  5.  4.
3.
  1.  1.  1.  0.  0.  2.  1.  1.  1.  1.  0.  0. 14. 16. 11. 15. 10.
5.
 21. 15.  8.  1.  1.  0.  2.  1.  1.  1.  0.  1.  2.  2.  1.  1.  1.
2.
  0.  1.  1.  1.  1.  0.  1.  2.  1. -1.  1.  0.  2.  0.  0.  0.  1.
1.
  0.  1.  0.  0.  3.  1.  2.  2.  2.  1.  2.  2.  2.  5. 10. 16. 15.
9.
  7.  9. 18. 11. 13. 12. 22. 13.  6.  5.  9.  6.  5. 18. 22. 11. 18.
12.
 26. 12.  7. 14. 27. 27. 17. 11. 12. 20. 13. 16. 18. 17.  8.  9. 13.
10.
 14. 24. 11. 15. 12. 20.  5. 13. 29. 12. 26. 19. 23. 17. 13. 21. 13.
12.
 12. 12. 22. 14. 14. 28. 27.  8.  8.  9.  4. 17. 10. 24. 27. 13. 12.
11.
  6.  5. 11. 10. 10. 14.  2.  6.  3.  5.  9.  3.  5.  1.  0.  1.  2.
2.
  1.  6.  5.  2.  4.  6.  4. 10.  5.  6.  2.  8.  5.  3.  3.  7.  5.
9.
  2. 22.  9.  5.  2.  1.  0.  0.  2.  1.  2.  2.  9.  5.  7. 15. 30.
8.
 35.  7. 20. 26. 22. 29. 29. 28. 26.  7.  9. 10.  9.  7.  8.  7.  9.
10.
 13. 12.  8. 10.  5. 21.  1. 22.  3.  7.  0.  0.  0.  1.  0.  0.  0.
0.
  1.  0.  1.  1.  1.  0.  1.  1. -1.  0.  1.  1.  0.  0.  2.  1.  1.
1.
  1.  1.  5. 12.  6.  7.  0.  1.  1.  0.  1.  1.  1.  0.  0.  4.  1.
0.
  2.  1.  1.  5.  5.  1.  1.  2.  1.  3.  3.  3.  6. 11.  6.  1.  5.
3.
  0.  0.  0.  0.  0.  1.  0.  1.  0.  1.  0.  1.  0.  0.  0.  2.  4.
2.
```

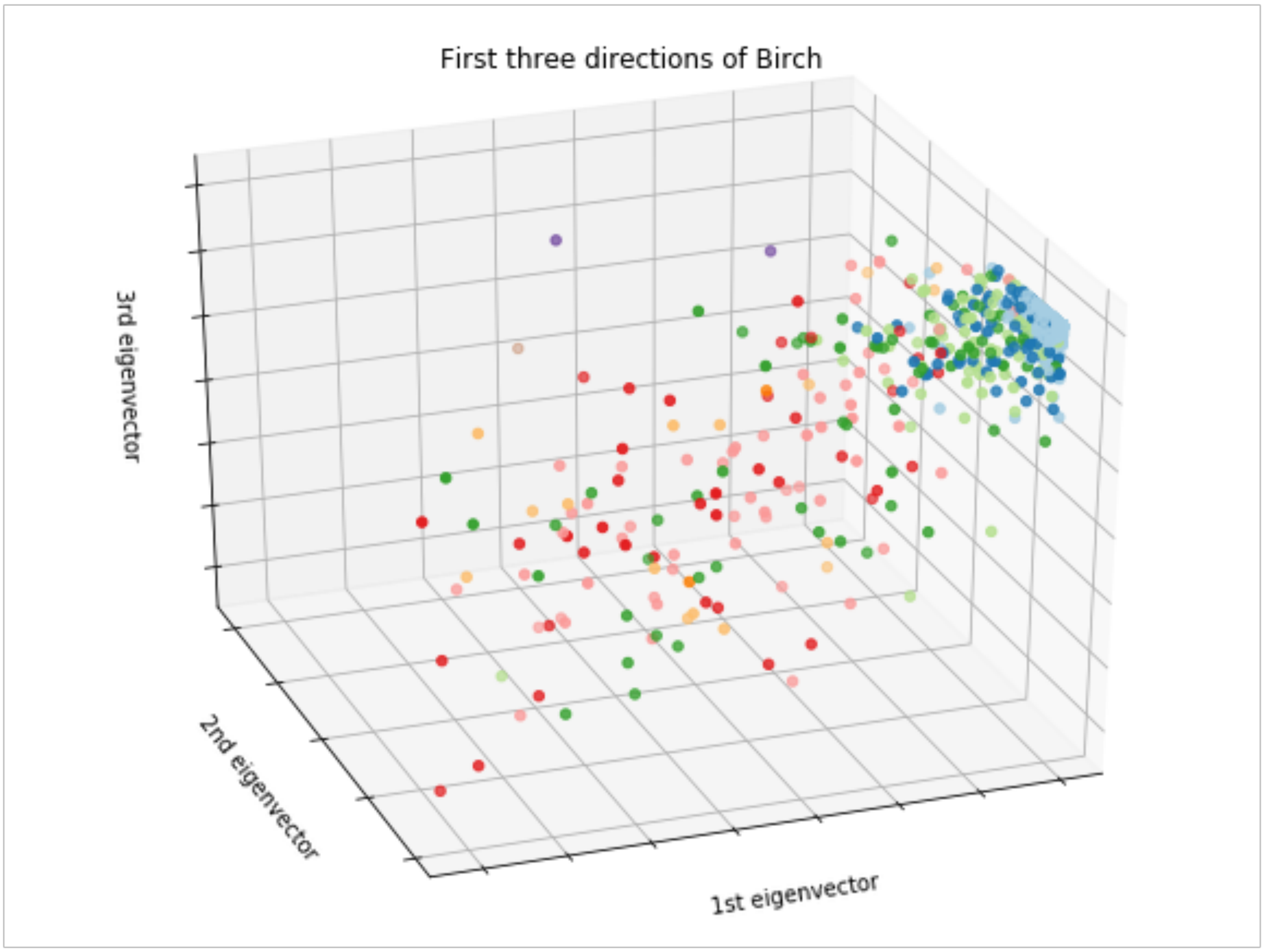
```
 1.  0.  1.  3.  2.  1.  2.  2.  3.  1.  0.  1.  1.  1.  3.  0.  5.
2.
 4.  3.  1.  1.  1.  5.  3.  2.  2.  1.  1.  1.  0.  1. -1.  4.  0.
8.
 2.  1.  3.  7.  5.  1.  0.  0.  1.  0.  0.  1.  1.  1.  1.  1. 24.
5.
19.  6.  5.  5.  2.  7.  7.  4. 12.  3.  6.  1.  0.  5.  5.  5. 10.
2.
 6.  2.  7.  6.  3.  9.  4.  8.  2.  0.  4.  2.  5. 10.  1.  4.  2.
1.
 0.]
SparsePCA %error 24.6 rmsle 0.47242543127653946
```




```
Ypredict [ 9.  3.  8.  6. 12.  2.  5. 12. 10. 27.  5.  6.  5. 17. 22
. 30. 30. 21.
 22.  8.  4.  1.  4. 34. 19.  9. 27. 22. 15. 20. 10. 10. 17. 30. 24.
24.
 22. 21. 28. 23. 28. 25. 32. 24. 24. 24. 20. 19. 35.  5.  7. 41.  5.
23.
 24. 28. 17. 27.  4. 18. 22. 13. 24. 23.  6. 31. 28. 14. 20. 18. 10.
25.
 13.  8. 31. 18.  4. 25. 30. 28.  9.  5. 21. 26. 16. 19. 26. 25. 27.
28.
 12. 27.  7. 10. 15. 32. 26.  4. 17. 13. 27. 20.  6.  2.  3.  7. 19.
5.
 11.  9.  5. 26. 23.  6. 11.  8.  2. 13. 28. 24. 11. 10.  7.  6. 13.
3.
```

```
  1. 18. 23. 20. 28. 21. 27. 36. 27. 25. 28. 23. 37. 26. 20. 31. 30.
14.
  5.  8.  6.  4.  5.  8.  3. 11. 11.  9.  3.  4. 15.  3.  2.  8.  2.
10.
  6. 15. 14.  8. 25. 23. 23. 14. 10. 27. 34. 19. 26. 26. 33. 32. 25.
12.
 16. 23. 22. 24. 29. 25. 23. 23. 20. 32. 30. 28. 21. 19.  7. 18. 12.
20.
  8. 24.  3. 25.  8.  4. 28.  1. 11. 49. 15. 14. 12.  3.  0.  1.  0.
0.
  0.  0.  0.  0.  1.  0.  0.  1.  0.  1.  0.  0.  0.  0.  0.  0.  1.
0.
  0.  0.  1.  0.  2.  1.  1.  0.  0.  0.  1.  2.  1.  1.  0.  0.  0.
0.
  0.  0.  1.  0.  3.  0.  0.  0. 30. 51. 32. 16.  2. 18. 13. 21. 25.
27.
  8.  1.  0.  0.  0.  0.  0.  0.  0.  0.  2.  2.  2. 15. 11. 37.  3.
2.
  0.  0.  4.  5.  5. 13.  4. 10.  3.  5. 14.  2.  3.  4.  4. 11.  3.
3.
  6.  4. 14.  4.  3.  4.  4. 15.  1.  7.  5.  5. 16.  4.  4.  6.  7.
9.
  2.  5.  4.  8.  6.  8.  7.  8.  8. 11.  2.  4.  5.  4.  1.  1.  6.
3.
  2.  2.  3.  2.  2.  0.  1.  0.  1.  0.  0.  0.  2.  7.  0.  6.  3.
3.
  2.  1. 18.  3.  0.  3. 12.  7.  6.  1.  3.  2.  3.  2.  0.  2.  1.
1.
  0.  0.  0.  1.  0.  0.  5.  7.  4.  3.  5.  5.  2.  2.  2.  6. 20.
21.
 20. 31.  3.  4. 18. 14. 35. 29. 16. 18. 64. 21. 24.  5. 13.  8.  3.
1.
  1.  3.  0.  0.  0.  2.  1.  1.  1.  0.  0.  1. 13. 14. 11. 11.  5.
6.
 22. 13. 12.  0.  1.  1.  1.  0.  1.  0.  0.  0.  1.  0.  2.  1.  1.
1.
  0.  0.  1.  1.  0.  1.  1.  0.  0.  0.  0.  2.  1.  0.  0.  0.  0.
0.
  1.  1.  1.  1.  4.  1.  0.  5.  3.  3.  3.  3.  1.  4. 12. 20. 18.
14.
  5. 17. 18. 10. 18. 14. 29. 15.  9.  2. 10.  4.  6. 18. 19. 10. 24.
18.
 23. 10.  4. 15. 38. 30. 12. 20. 13. 26. 13. 16. 25. 15. 11. 18. 15.
14.
 12. 25. 10. 16. 11. 21.  9. 12. 29. 17. 27. 15. 27.  9.  9. 21.  8.
21.
 15. 11. 24. 10. 16. 21. 34.  8.  6.  6.  6. 23. 11. 17. 24. 13. 11.
14.
  6.  2.  7.  5.  9. 19.  4. 11.  6.  2. 13.  3.  3.  0.  2.  0.  0.
0.
  1.  5.  4.  4.  4.  4.  4. 14.  5.  6.  4.  2.  6.  1.  8.  3.  2.
12.
  6. 25.  6.  8.  1.  0.  0.  0.  3.  2.  0.  1.  5.  5.  8. 20. 31.
```

```
14.
 33.   8.  22.  24.  21.  26.  29.  33.  25.  11.   7.   7.   8.   8.   3.   7.   7.
18.
 11.  11.  12.   7.   6.  20.   0.  26.   1.   7.   2.   0.   1.   0.   0.   0.   2.
2.
  0.   0.   0.   0.   0.   0.   0.   1.   0.   0.   2.   0.   1.   2.   0.   0.   0.
3.
  1.   0.   9.  15.  10.   6.   0.   0.   0.   1.   0.   0.   0.   0.   0.   3.   2.
0.
  0.   0.   1.   4.   7.   0.   0.   0.   0.   3.   1.   0.   4.  11.   2.   2.   2.
2.
  1.   0.   1.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   1.   3.
1.
  0.   0.   0.   3.   1.   0.   1.   1.   3.   1.   1.   0.   1.   4.   0.   0.  12.
1.
  8.   0.   1.   0.   1.   1.   0.   2.   0.   0.   1.   0.   0.   0.   0.   5.   0.
8.
  0.   0.   1.   9.   1.   0.   0.   0.   0.   0.   0.   0.   1.   0.   0.   0.  18.
3.
 19.   7.   4.   1.   1.   6.   5.   4.   9.   1.   6.   0.   0.   3.   9.   4.   9.
1.
  4.   1.   4.   5.   3.   7.   2.   7.   1.   0.   9.   2.   5.   7.   0.   3.   0.
1.
  0.]
SparseRP %error 0.0 rmsle 1.0038462317057075e-12
```




```
Ypredict [ 9.   2.   5.   9.  12.   6.   4.   8.  13.  22.  15.   2.   9.  17.  29
```

```
. 24. 31. 25.
 22.  9.  7.  5.  4. 23. 22. 15. 26. 20. 15. 24.  5.  5. 11. 26. 24.
27.
 30. 26. 24. 23. 26. 26. 31. 22. 22. 24. 27. 27. 27.  8. 12. 30.  6.
26.
 26. 25. 25. 27.  9. 23. 22. 12. 28. 23.  7. 25. 27.  9. 23. 25.  9.
31.
 26. 10. 28. 24.  6. 24. 25. 23. 12.  6. 31. 31. 23. 24. 23. 22. 26.
18.
  7. 25.  9. 10. 14. 29. 28.  6. 10. 20. 25. 29.  8.  5.  3. 15. 16.
4.
  8.  7.  7. 25. 21.  7.  7. 11.  4. 10. 23. 29.  8.  7.  4.  5.  9.
6.
  3. 20. 23. 22. 24. 28. 24. 27. 24. 23. 30. 20. 29. 23. 22. 30. 25.
12.
  8.  8.  4.  6.  6.  4.  6. 11. 10.  8.  4.  4.  8.  5.  6.  8.  5.
11.
  5.  9.  7. 12. 25. 26. 25. 24.  8. 28. 29. 28. 21. 24. 29. 32. 27.
24.
 21. 27. 22. 29. 27. 24. 25. 26. 21. 30. 28. 30. 31. 28.  4. 23.  8.
15.
  3. 24.  5. 21.  6.  2. 17.  5. 10. 43. 14. 16. 12.  1.  1.  1.  0.
1.
  0.  1.  0.  0.  1.  1.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.
  1.  0.  0.  1.  0.  1.  1.  1.  0.  1.  1.  0.  1.  1.  0.  0.  0.
1.
  0.  0.  5.  0.  1.  0.  0.  0. 24. 51. 27. 10.  4. 12.  9. 24. 15.
26.
  5.  1.  0.  1.  1.  0.  1.  0.  0.  1.  1.  1.  0. 20. 13. 28.  2.
1.
  1.  1.  5.  4.  4. 11.  5.  5.  4.  5. 10.  4.  3.  4.  5. 10.  5.
6.
  5.  3.  9.  4.  5.  6.  7. 11.  7.  4.  4.  5.  9.  6.  3.  3.  3.
11.
  5.  3.  2.  6.  4.  7.  7.  8.  8. 13.  4.  5.  4.  5.  2.  1.  5.
5.
  4.  2.  5.  1.  0.  1.  1.  0.  0.  1.  0.  0.  6.  4.  4.  7.  2.
4.
  2.  0. 16.  6.  5.  5.  9.  6.  6.  4.  4.  2.  2.  2.  1.  0.  1.
0.
  1.  0.  0.  0.  2.  0.  8.  5.  4.  4.  5.  9.  3.  4.  4.  3. 12.
16.
 21. 25.  8.  4. 19. 13. 27. 22. 17. 18. 60. 26. 23.  3. 17.  5.  3.
3.
  1.  1.  1.  0.  0.  2.  1.  1.  0.  1.  0.  0. 14. 16. 12. 15. 10.
6.
 21. 15.  8.  1.  1.  1.  0.  0.  1.  0.  0.  1.  2.  2.  1.  1.  1.
1.
  0.  1.  1.  1.  0.  1.  1.  1.  1.  0.  1.  0.  1.  0.  0.  0.  0.
0.
  0.  0.  0.  1.  2.  2.  2.  2.  2.  1.  3.  1.  1.  4. 10. 15. 14.
9.
```

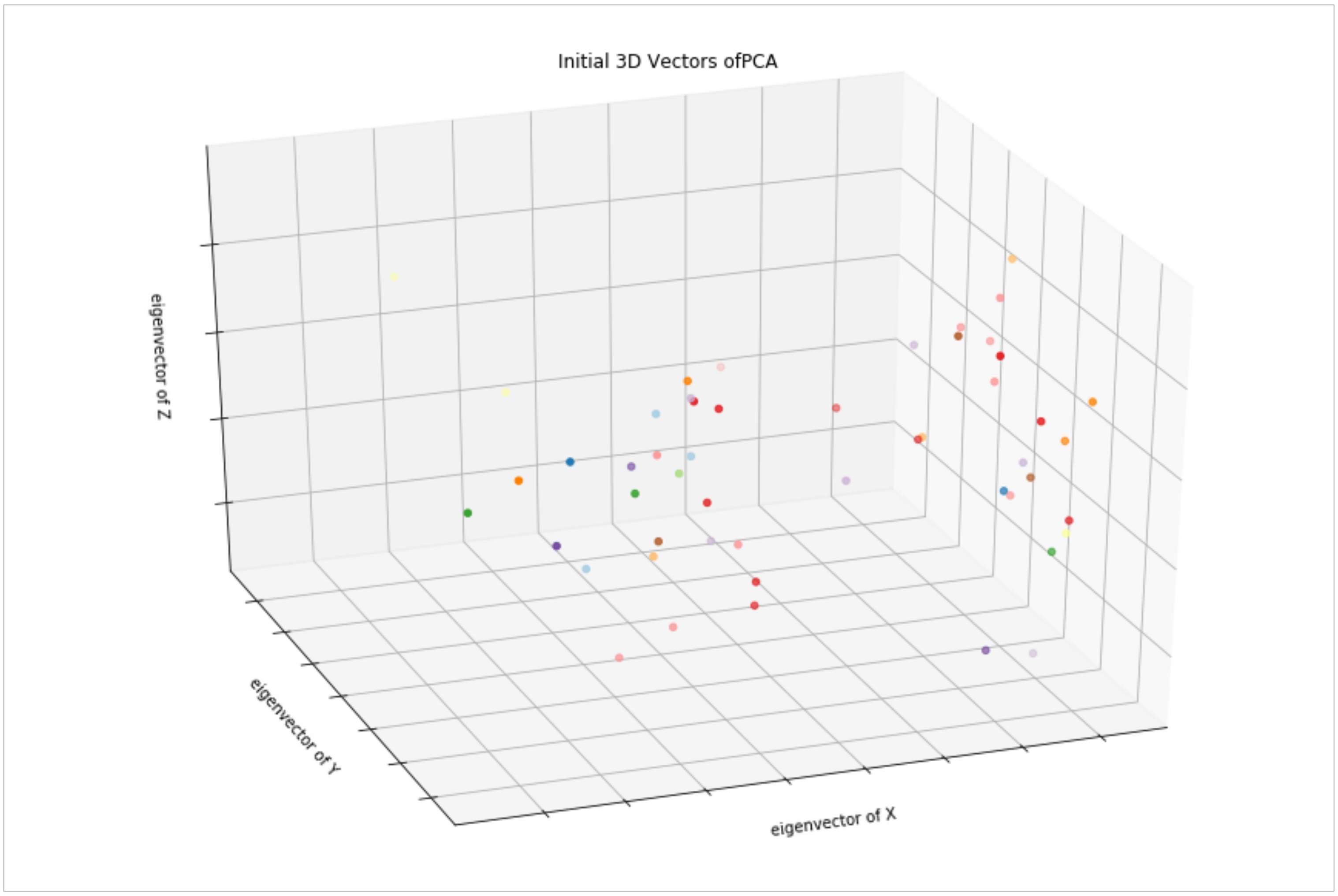
```
  6.   9. 19. 10. 13. 13. 22. 14.   5.   4.   9.   6.   5. 17. 22. 10. 18.
12.
 26. 12.   7. 14. 26. 28. 16. 11. 11. 20. 13. 17. 18. 17.   9.   9. 13.
11.
 14. 24. 11. 16. 12. 19.   5. 12. 29. 12. 28. 18. 23. 19. 13. 20. 14.
12.
 12. 12. 22. 13. 14. 28. 26.   8.   7.   9.   3. 17. 11. 25. 27. 14. 13.
11.
  6.   5. 11. 10.   9. 15.   2.   5.   4.   4.   9.   3.   5.   0.   1.   0.   1.
1.
  0.   6.   5.   2.   4.   5.   3. 10.   5.   5.   2.   8.   6.   3.   3.   5.   5.
8.
  2. 22. 11.   5.   2.   1.   0.   1.   2.   1.   0.   2.   8.   4.   7. 15. 31.
8.
 36.   6. 20. 25. 21. 29. 31. 27. 27.   7. 10. 10.   9.   6.   8.   8.   9.
11.
 12. 12.   8.   9.   5. 22.   1. 23.   3.   6.   0.   0.   0.   0.   0.   0.   1.
0.
  1.   0.   0.   0.   1.   1.   0.   0.   0.   0.   1.   1.   0.   1.   1.   0.   0.
1.
  1.   1.   5. 12.   6.   7.   0.   0.   1.   0.   1.   1.   0.   0.   1.   4.   1.
1.
  2.   1.   1.   5.   5.   1.   0.   1.   2.   3.   3.   3.   5. 11.   6.   1.   4.
3.
  1.   0.   0.   0.   0.   0.   0.   0.   0.   1.   0.   1.   0.   0.   0.   2.   5.
2.
  1.   0.   1.   4.   2.   1.   2.   3.   2.   2.   1.   1.   1.   1.   2.   0.   5.
3.
  4.   3.   1.   1.   1.   3.   2.   2.   1.   1.   1.   1.   1.   0.   0.   3.   0.
8.
  2.   1.   3.   6.   5.   1.   0.   1.   1.   0.   0.   0.   1.   1.   2.   1. 23.
4.
 19.   5.   4.   3.   3.   8.   7.   4. 12.   2.   7.   2.   1.   5.   5.   5. 10.
3.
  7.   2.   7.   4.   3.   8.   4.   8.   2.   1.   4.   2.   4. 12.   1.   5.   2.
0.
  1.]
Birch %error 24.0 rmsle 0.41079050551266977
```

In [178]:

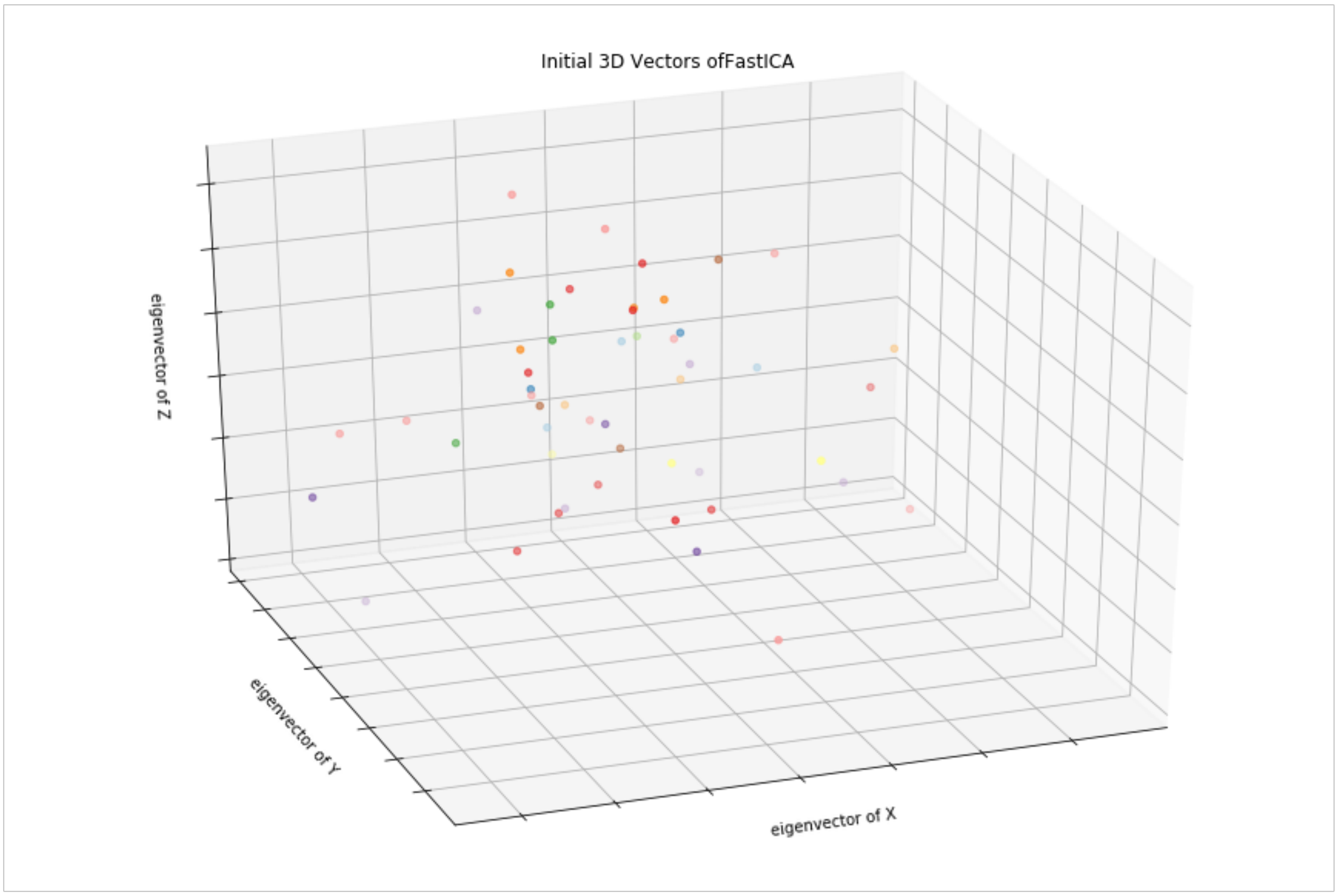
```
n_col=4
kolom=sales.Product_Code
X = (sales.drop(['Product_Code'],axis=1).T)[:51] #sales 51weeks
X.columns=kolom

def rmsle(y_predicted, y_real):
    return np.sqrt(np.mean(np.power(np.log1p(y_predicted)-np.log1p(y_real), 2)))
def procenterror(y_predicted, y_real):
     return np.round( np.mean(np.abs(y_predicted-y_real) )/ np.mean(y_real) *100
,1)


Y=X['P33']
```

```
X=X.fillna(value=0)  #nasty NaN


methods = [
         'PCA',
         'FastICA',
         'Gauss',
         'KMeans',
         'SparsePCA',
         'SparseRP',
         'Birch',
         'NMF',
       #  'LatentDietrich',
        ]

classifiers = [

    PCA(n_components=n_col),
    FastICA(n_components=n_col),
    GaussianRandomProjection(n_components=3),
    KMeans(n_clusters=n_col),
    #SparsePCA(n_components=n_col),
    SparseRandomProjection(n_components=n_col, dense_output=True),
    Birch(branching_factor=10, n_clusters=3, threshold=0.5),
    NMF(n_components=n_col),
  #  LatentDirichletAllocation(n_topics=n_col),

]
correction= [1,1,0,0,0,0,0,0,0]

temp=zip(methods,classifiers,correction)
print(temp)

for method, clf,correct in temp:
    Xr=clf.fit_transform(X,Y)
    V3(Xr,method)
    res = sm.OLS(Y,Xr).fit()
    #print(res.summary())  # show OLS regression
    #print(res.predict(Xr).round()+correct)  #show OLS prediction
    #print('Ypredict',res.predict(Xr).round()+correct)  #show OLS prediction

    print('Ypredict',res.predict(Xr).round()+correct*Y.mean())  #show OLS predic
tion
    print(method,'%error',procenterror(res.predict(Xr)+correct*Y.mean(),Y),'rmsl
e',rmsle(res.predict(Xr)+correct*Y.mean(),Y)) #
```

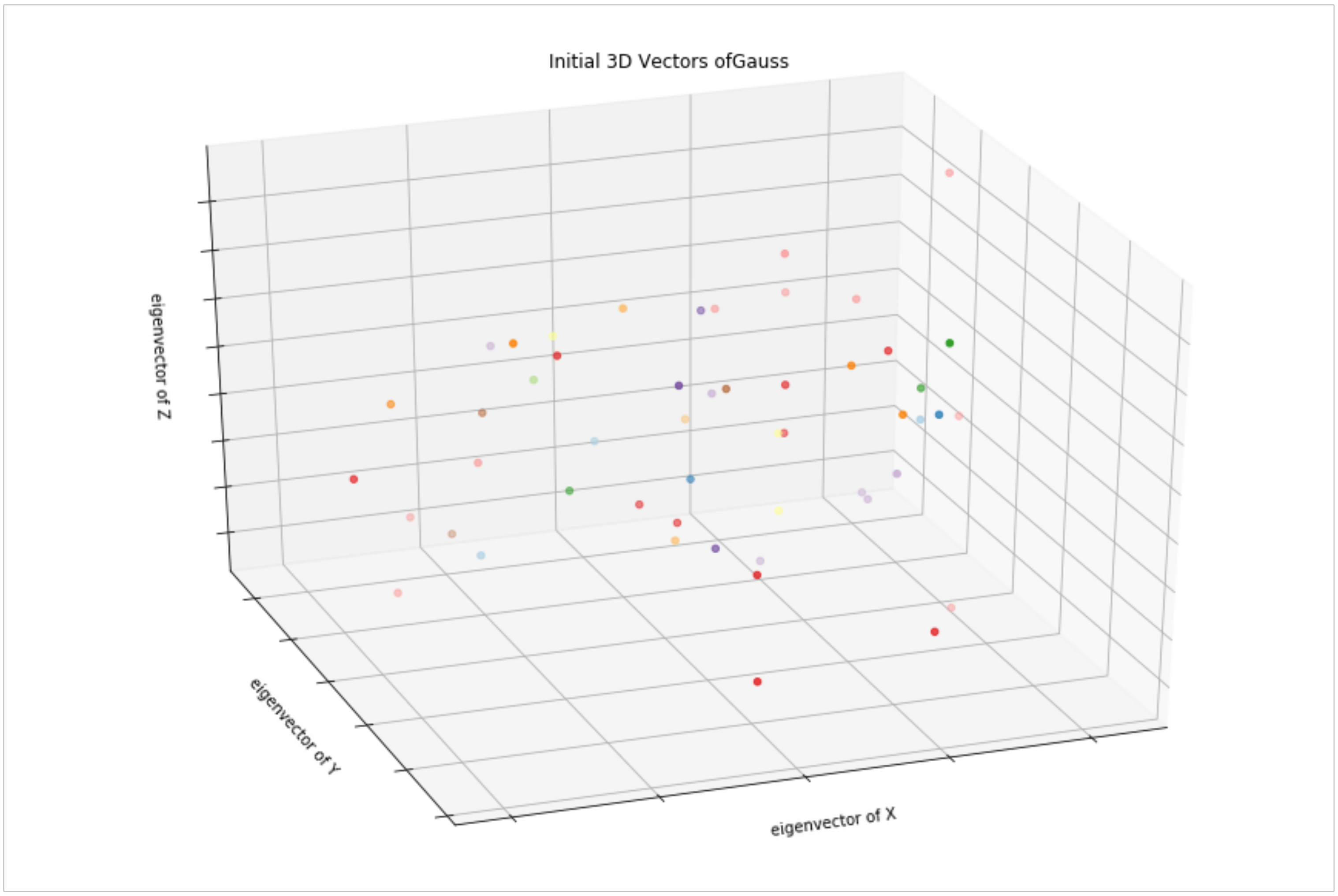
```
<zip object at 0x1c2f4edf88>
```



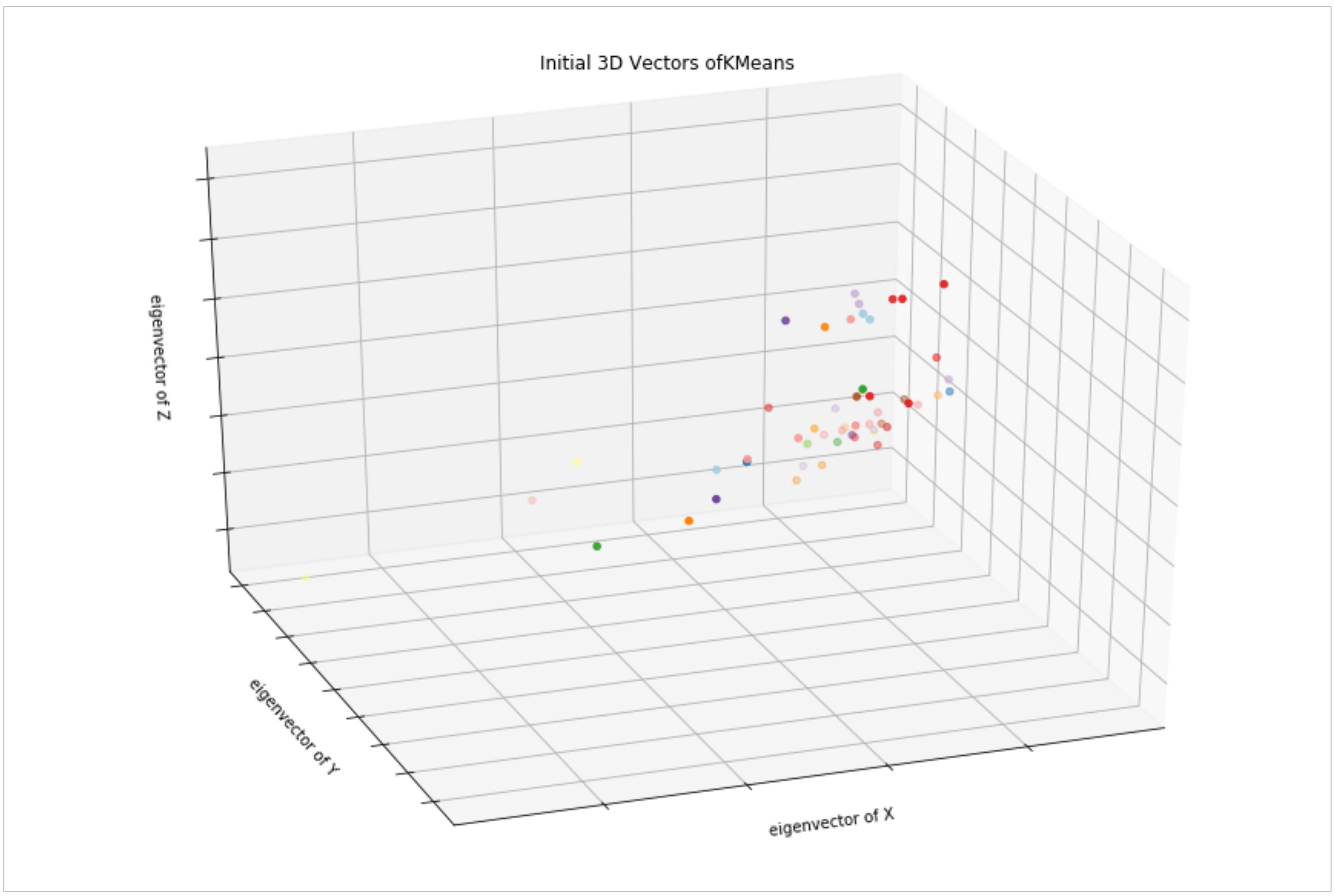

```
Ypredict [11.62745098 10.62745098 10.62745098 11.62745098 11.6274509
8 11.62745098
 11.62745098 12.62745098 12.62745098 11.62745098 11.62745098 10.6274
5098
 12.62745098 11.62745098 11.62745098 12.62745098 12.62745098 11.6274
5098
 13.62745098 12.62745098 12.62745098 12.62745098 12.62745098 13.6274
5098
 15.62745098 14.62745098  9.62745098  9.62745098  9.62745098 10.6274
5098
 10.62745098 10.62745098 10.62745098 10.62745098 10.62745098  9.6274
5098
 10.62745098 10.62745098 10.62745098 10.62745098 10.62745098 10.6274
5098
 11.62745098 11.62745098 11.62745098 11.62745098 11.62745098 12.6274
5098
 11.62745098 12.62745098 13.62745098]
PCA %error 22.7 rmsle 0.29086587715800943
```



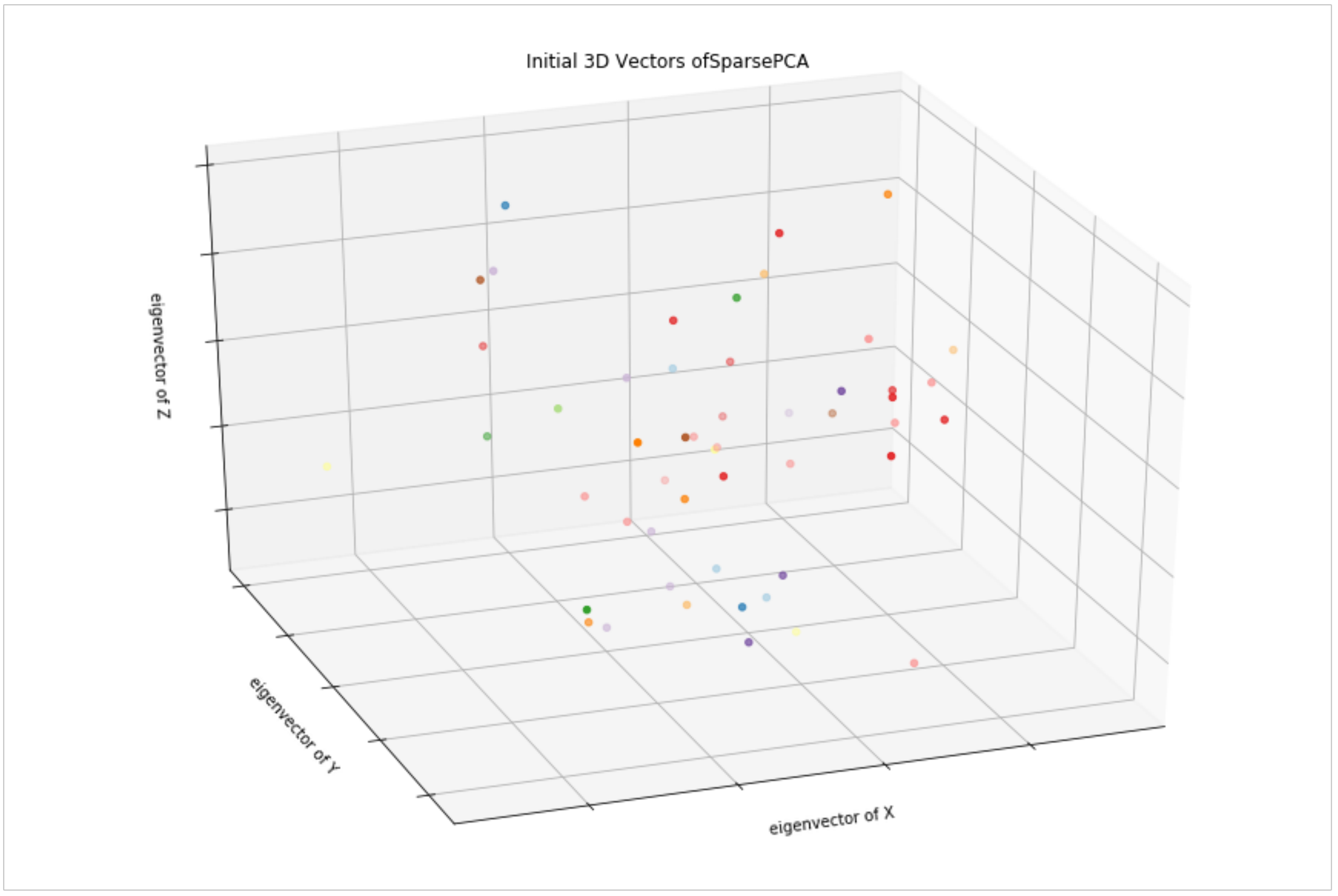

```
Ypredict [11.62745098 10.62745098 10.62745098 11.62745098 11.6274509
8 10.62745098
 11.62745098 12.62745098 12.62745098 11.62745098 11.62745098 10.6274
5098
 12.62745098 11.62745098 11.62745098 12.62745098 12.62745098 11.6274
5098
 13.62745098 12.62745098 12.62745098 12.62745098 12.62745098 13.6274
5098
 15.62745098 14.62745098  9.62745098  9.62745098  9.62745098 10.6274
5098
 10.62745098 10.62745098 10.62745098 10.62745098 10.62745098  9.6274
5098
  9.62745098 10.62745098 10.62745098 10.62745098 10.62745098 10.6274
5098
 11.62745098 11.62745098 11.62745098 11.62745098 11.62745098 12.6274
5098
 11.62745098 12.62745098 13.62745098]
FastICA %error 22.7 rmsle 0.2907674815926446
```



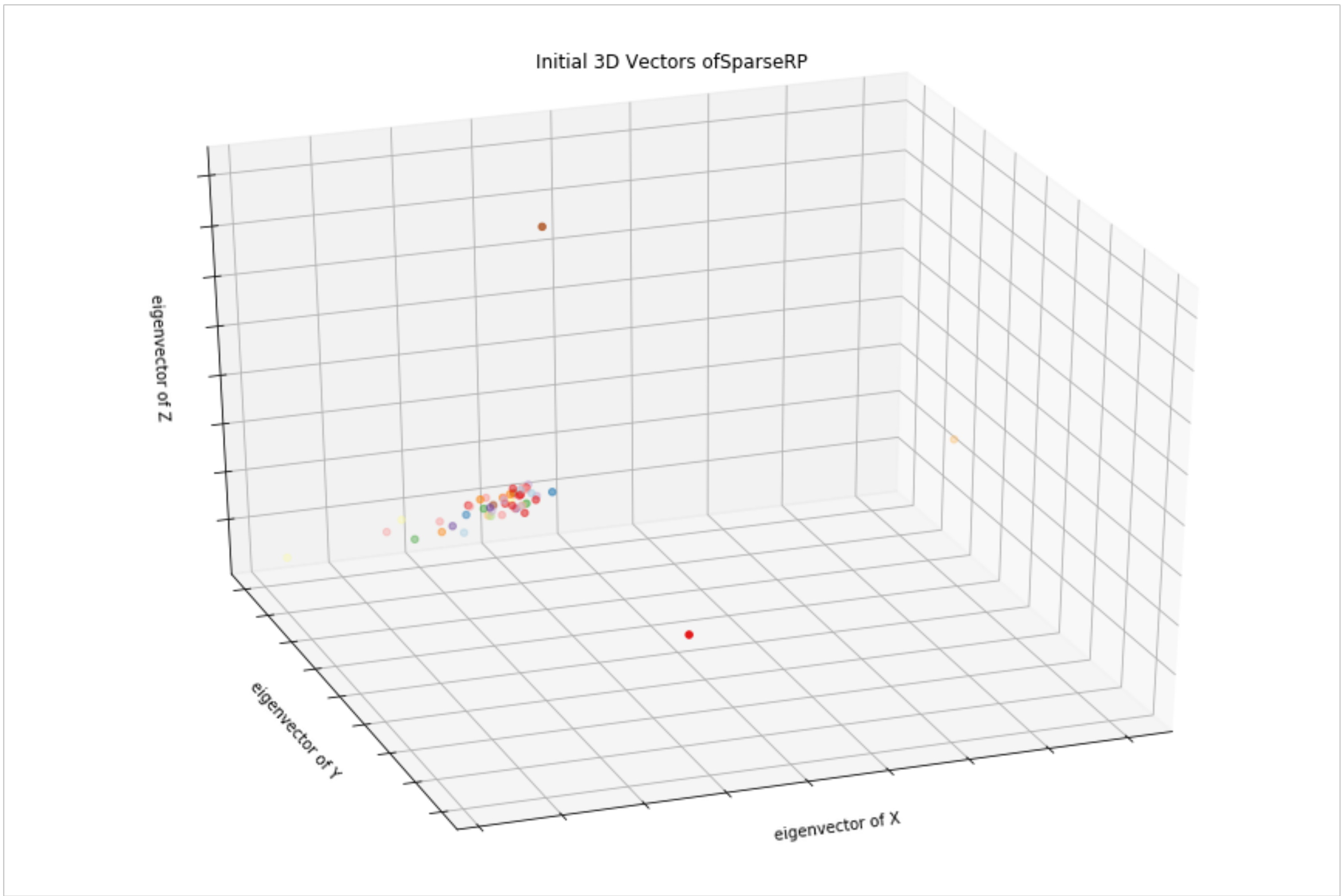

```
Ypredict [10. 14. 12. 11. 12. 14. 10. 13. 13. 13. 13. 12. 17. 11. 13
.  9. 13. 11.
 12. 15. 15. 14. 13. 13. 13. 11. 10.  9.  9. 10. 10. 13.  8.  8. 11.
9.
 11.  9. 12.  6. 11. 13. 10.  9. 10. 11. 10. 12.  7. 12. 11.]
Gauss %error 27.3 rmsle 0.34383107950426456
```



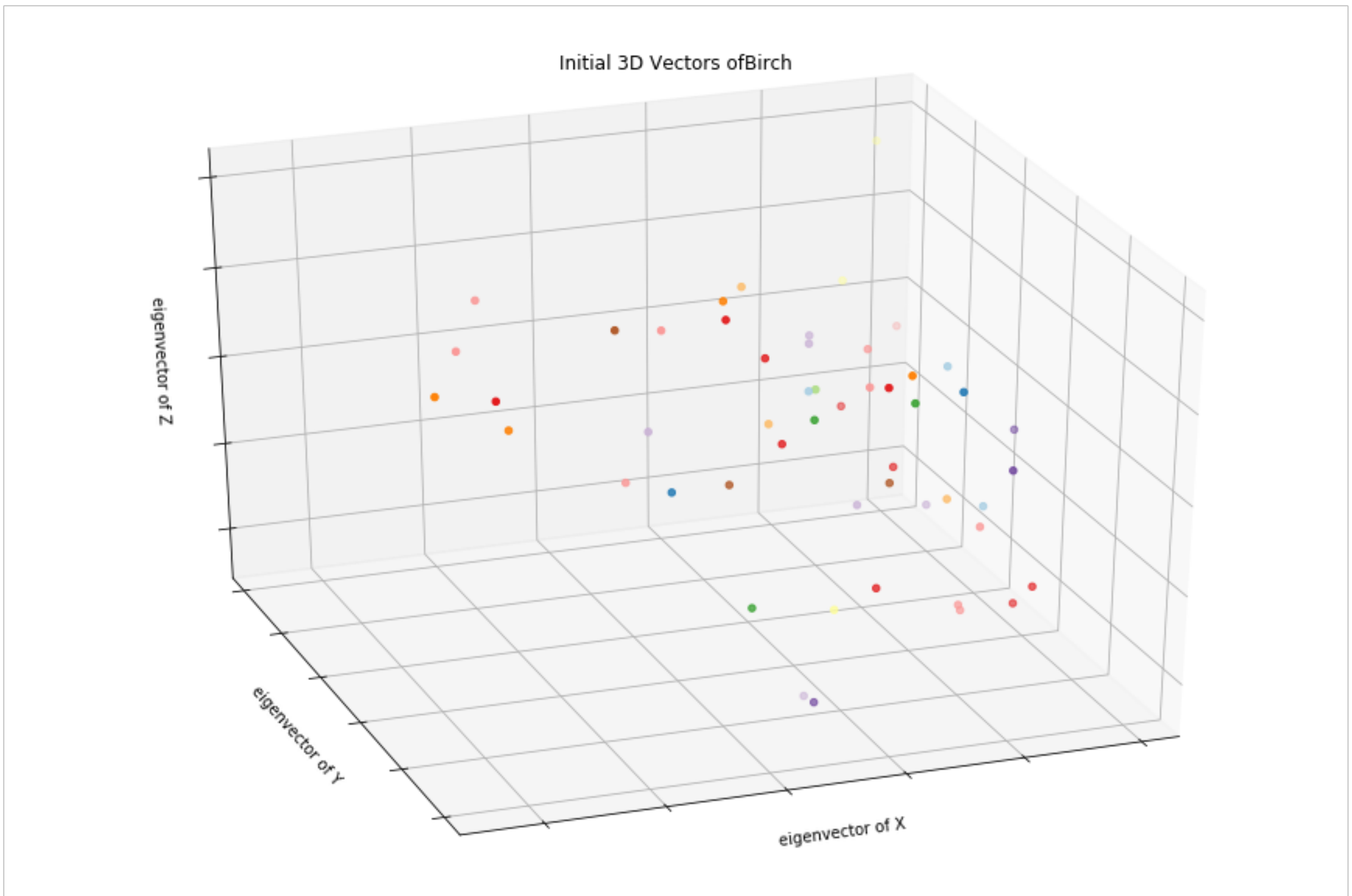
```
Ypredict [11. 11. 12. 12. 12. 12. 12. 12. 13. 11. 13. 12. 12. 12. 12
. 13. 13. 13.
 13. 12. 13. 12. 13. 13. 15. 16. 11. 10. 10. 10. 11. 11. 10. 10. 10.
10.
 10. 11. 11. 10. 10. 11. 11. 12. 10. 11. 11. 12. 11. 12. 13.]
KMeans %error 23.2 rmsle 0.2972499903151159
```




```
Ypredict [12. 14. 14. 12. 14. 13. 12. 13. 14. 14. 11. 11. 12. 12. 12
. 13. 10. 14.
 14. 12. 12. 12. 16. 12. 12.  8. 11.  8. 10. 10. 10.  9. 11. 11. 12.
13.
  9. 11. 10.  7. 10. 11. 12. 11. 11. 10. 13.  9. 10. 11. 11.]
SparsePCA %error 25.4 rmsle 0.316240501220898
```




```
Ypredict [15. 12. 11. 17. 10. 18. 11. 16.  8.  6. 11. 10. 18. 10. 14
. 13. 13. 12.
 14. 10. 10. 11. 14. 11. 10. 17.  8. 13.  6. 16.  9.  4. 12. 10. 19.
9.
  7.  5. 10.  8. 11.  5. 11. 13. 11. 11. 11. 14. 15. 16. 17.]
SparseRP %error 0.0 rmsle 1.6998793374513743e-15
```



```
Ypredict [12. 11. 10. 12. 12. 12. 12. 13. 13. 13. 12. 11. 13. 13. 12
. 13. 13. 12.
 14. 13. 13. 13. 13. 14. 16. 13.  9.  8.  9.  9. 10. 10. 10. 10. 10.
9.
 10. 10. 10. 10. 10. 11. 11. 11. 11. 11. 12. 12. 11. 12. 13.]
Birch %error 22.7 rmsle 0.28949425013355634
```

In [179]:

```
from sklearn.linear_model import OrthogonalMatchingPursuit,RANSACRegressor,Logis
ticRegression,ElasticNetCV,HuberRegressor, Ridge, Lasso,LassoCV,Lars,BayesianRid
ge,SGDClassifier,LogisticRegressionCV,RidgeClassifier
from sklearn.svm import SVC
from sklearn.preprocessing import MinMaxScaler,PolynomialFeatures
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

param_grid = {'C': [0.1,1, 10, 100, 1000], 'gamma': [1,0.1,0.01,0.001,0.0001], '
kernel': ['rbf']}


# import some data to play with
#X = df_new[df_new['split']==0]
X = X.drop(['P33'],axis=1)
def rmsle(y_predicted, y_real):
    return np.sqrt(np.mean(np.power(np.log1p(y_predicted)-np.log1p(y_real), 2)))
def procenterror(y_predicted, y_real):
     return np.round( np.mean(np.abs(y_predicted-y_real) )/ np.mean(y_real) *100
,1)

#Y=df_new[df_new['split']==0]


#X=X.replace([np.inf, -np.inf], np.nan).fillna(value=0)
#print(X) #nasty NaN
#scaler = MinMaxScaler()
#scaler.fit(X)
#X=scaler.transform(X)
#poly = PolynomialFeatures(2)
#X=poly.fit_transform(X)


methods = [
         #'ElasticNet',
         'SVC',
         'kSVC',
         'KNN',
         'DecisionTree',
         'RandomForestClassifier',
         #'GridSearchCV',
         'HuberRegressor',
```

```
         'Ridge',
         'Lasso',
         'LassoCV',
         'Lars',
         #'BayesianRidge',
         'SGDClassifier',
         'RidgeClassifier',
         'LogisticRegression',
         'OrthogonalMatchingPursuit',
         #'RANSACRegressor',
         ]

classifiers = [
    #ElasticNetCV(cv=10, random_state=0),
    SVC(),
    SVC(kernel = 'rbf', random_state = 0),
    KNeighborsClassifier(n_neighbors = 1),
    DecisionTreeClassifier(),
    RandomForestClassifier(n_estimators = 200),
    #GridSearchCV(SVC(),param_grid, refit = True, verbose = 1),
    HuberRegressor(fit_intercept=True, alpha=0.0, max_iter=100,epsilon=2.95),
    Ridge(fit_intercept=True, alpha=0.0, random_state=0, normalize=True),
    Lasso(alpha=0.05),
    LassoCV(),
    Lars(n_nonzero_coefs=10),
    #BayesianRidge(),
    SGDClassifier(),
    RidgeClassifier(),
    LogisticRegression(),
    OrthogonalMatchingPursuit(),
    #RANSACRegressor(),
]
correction= [0,0,0,0,0,0,0,0,0,0,0,0]

temp=zip(methods,classifiers,correction)
print(temp)

for method, clf,correct in temp:
    regr=clf.fit(X,Y)
    #print( method,'% errors', abs(regr.predict(X)+correct-Y).sum()/(Y.sum())*10
0)
    print(method,'%error',procenterror(regr.predict(X),Y),'rmsle',rmsle(regr.pre
dict(X),Y))
    from sklearn.metrics import classification_report, confusion_matrix, accurac
y_score,f1_score, precision_score, recall_score

    # Confusion Matrix
    print(method,'Confusion Matrix')
    print(confusion_matrix(Y, np.round(regr.predict(X) ) ) )
    print('--'*40)

    # Classification Report
    print('Classification Report')
    print(classification_report(Y,np.round( regr.predict(X) ) ))
```

```
    # Accuracy
    print('--'*40)
    logreg_accuracy = round(accuracy_score(Y, np.round( regr.predict(X) ) ) * 10
0,2)
    print('Accuracy', logreg_accuracy,'%')
```

```
<zip object at 0x1c2f576608>
SVC %error 0.0 rmsle 0.0
SVC Confusion Matrix
[[ 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       1.00      1.00      1.00         1
        5.0       1.00      1.00      1.00         2
        6.0       1.00      1.00      1.00         2
        7.0       1.00      1.00      1.00         1
        8.0       1.00      1.00      1.00         3
        9.0       1.00      1.00      1.00         2
       10.0       1.00      1.00      1.00         8
       11.0       1.00      1.00      1.00        10
       12.0       1.00      1.00      1.00         3
       13.0       1.00      1.00      1.00         4
       14.0       1.00      1.00      1.00         4
       15.0       1.00      1.00      1.00         2
       16.0       1.00      1.00      1.00         3
       17.0       1.00      1.00      1.00         3
       18.0       1.00      1.00      1.00         2
       19.0       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00        51

--------------------------------------------------------------------
------------
Accuracy 100.0 %
```

```
kSVC %error 0.0 rmsle 0.0
kSVC Confusion Matrix
[[ 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       1.00      1.00      1.00         1
        5.0       1.00      1.00      1.00         2
        6.0       1.00      1.00      1.00         2
        7.0       1.00      1.00      1.00         1
        8.0       1.00      1.00      1.00         3
        9.0       1.00      1.00      1.00         2
       10.0       1.00      1.00      1.00         8
       11.0       1.00      1.00      1.00        10
       12.0       1.00      1.00      1.00         3
       13.0       1.00      1.00      1.00         4
       14.0       1.00      1.00      1.00         4
       15.0       1.00      1.00      1.00         2
       16.0       1.00      1.00      1.00         3
       17.0       1.00      1.00      1.00         3
       18.0       1.00      1.00      1.00         2
       19.0       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00        51

--------------------------------------------------------------------
------------
Accuracy 100.0 %
KNN %error 0.0 rmsle 0.0
KNN Confusion Matrix
[[ 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
```

```
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       1.00      1.00      1.00         1
        5.0       1.00      1.00      1.00         2
        6.0       1.00      1.00      1.00         2
        7.0       1.00      1.00      1.00         1
        8.0       1.00      1.00      1.00         3
        9.0       1.00      1.00      1.00         2
       10.0       1.00      1.00      1.00         8
       11.0       1.00      1.00      1.00        10
       12.0       1.00      1.00      1.00         3
       13.0       1.00      1.00      1.00         4
       14.0       1.00      1.00      1.00         4
       15.0       1.00      1.00      1.00         2
       16.0       1.00      1.00      1.00         3
       17.0       1.00      1.00      1.00         3
       18.0       1.00      1.00      1.00         2
       19.0       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00        51

--------------------------------------------------------------------
------------
Accuracy 100.0 %
DecisionTree %error 0.0 rmsle 0.0
DecisionTree Confusion Matrix
[[ 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
```

```
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       1.00      1.00      1.00         1
        5.0       1.00      1.00      1.00         2
        6.0       1.00      1.00      1.00         2
        7.0       1.00      1.00      1.00         1
        8.0       1.00      1.00      1.00         3
        9.0       1.00      1.00      1.00         2
       10.0       1.00      1.00      1.00         8
       11.0       1.00      1.00      1.00        10
       12.0       1.00      1.00      1.00         3
       13.0       1.00      1.00      1.00         4
       14.0       1.00      1.00      1.00         4
       15.0       1.00      1.00      1.00         2
       16.0       1.00      1.00      1.00         3
       17.0       1.00      1.00      1.00         3
       18.0       1.00      1.00      1.00         2
       19.0       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00        51

--------------------------------------------------------------------
------------
Accuracy 100.0 %
RandomForestClassifier %error 0.0 rmsle 0.0
RandomForestClassifier Confusion Matrix
[[ 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       1.00      1.00      1.00         1
```

```
        5.0       1.00      1.00      1.00         2
        6.0       1.00      1.00      1.00         2
        7.0       1.00      1.00      1.00         1
        8.0       1.00      1.00      1.00         3
        9.0       1.00      1.00      1.00         2
       10.0       1.00      1.00      1.00         8
       11.0       1.00      1.00      1.00        10
       12.0       1.00      1.00      1.00         3
       13.0       1.00      1.00      1.00         4
       14.0       1.00      1.00      1.00         4
       15.0       1.00      1.00      1.00         2
       16.0       1.00      1.00      1.00         3
       17.0       1.00      1.00      1.00         3
       18.0       1.00      1.00      1.00         2
       19.0       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00        51

--------------------------------------------------------------------
------------
Accuracy 100.0 %
HuberRegressor %error 0.0 rmsle 5.300234359899972e-05
HuberRegressor Confusion Matrix
[[ 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       1.00      1.00      1.00         1
        5.0       1.00      1.00      1.00         2
        6.0       1.00      1.00      1.00         2
        7.0       1.00      1.00      1.00         1
        8.0       1.00      1.00      1.00         3
        9.0       1.00      1.00      1.00         2
       10.0       1.00      1.00      1.00         8
       11.0       1.00      1.00      1.00        10
       12.0       1.00      1.00      1.00         3
```

```
       13.0       1.00      1.00      1.00         4
       14.0       1.00      1.00      1.00         4
       15.0       1.00      1.00      1.00         2
       16.0       1.00      1.00      1.00         3
       17.0       1.00      1.00      1.00         3
       18.0       1.00      1.00      1.00         2
       19.0       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00        51

--------------------------------------------------------------------
------------
Accuracy 100.0 %
Ridge %error 0.0 rmsle 8.429441615446194e-16
Ridge Confusion Matrix
[[ 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       1.00      1.00      1.00         1
        5.0       1.00      1.00      1.00         2
        6.0       1.00      1.00      1.00         2
        7.0       1.00      1.00      1.00         1
        8.0       1.00      1.00      1.00         3
        9.0       1.00      1.00      1.00         2
       10.0       1.00      1.00      1.00         8
       11.0       1.00      1.00      1.00        10
       12.0       1.00      1.00      1.00         3
       13.0       1.00      1.00      1.00         4
       14.0       1.00      1.00      1.00         4
       15.0       1.00      1.00      1.00         2
       16.0       1.00      1.00      1.00         3
       17.0       1.00      1.00      1.00         3
       18.0       1.00      1.00      1.00         2
       19.0       1.00      1.00      1.00         1
```

```
avg / total       1.00      1.00      1.00        51

--------------------------------------------------------------------
------------
Accuracy 100.0 %
Lasso %error 0.4 rmsle 0.006010842419511365
Lasso Confusion Matrix
[[ 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       1.00      1.00      1.00         1
        5.0       1.00      1.00      1.00         2
        6.0       1.00      1.00      1.00         2
        7.0       1.00      1.00      1.00         1
        8.0       1.00      1.00      1.00         3
        9.0       1.00      1.00      1.00         2
       10.0       1.00      1.00      1.00         8
       11.0       1.00      1.00      1.00        10
       12.0       1.00      1.00      1.00         3
       13.0       1.00      1.00      1.00         4
       14.0       1.00      1.00      1.00         4
       15.0       1.00      1.00      1.00         2
       16.0       1.00      1.00      1.00         3
       17.0       1.00      1.00      1.00         3
       18.0       1.00      1.00      1.00         2
       19.0       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00        51

--------------------------------------------------------------------
------------
Accuracy 100.0 %
```

```
/anaconda3/lib/python3.6/site-packages/sklearn/linear_model/ridge.py
:154: UserWarning: Singular matrix in solving dual problem. Using le
ast-squares solution instead.
  warnings.warn("Singular matrix in solving dual problem. Using "
/anaconda3/lib/python3.6/site-packages/sklearn/linear_model/coordina
te_descent.py:491: ConvergenceWarning: Objective did not converge. Y
ou might want to increase the number of iterations. Fitting data wit
h very small alpha may cause precision problems.
  ConvergenceWarning)

LassoCV %error 23.9 rmsle 0.3057257200919362
LassoCV Confusion Matrix
[[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 3 5 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 9 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 3 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       0.00      0.00      0.00         1
        5.0       0.00      0.00      0.00         2
        6.0       0.00      0.00      0.00         2
        7.0       0.00      0.00      0.00         1
        8.0       0.00      0.00      0.00         3
        9.0       0.00      0.00      0.00         2
       10.0       0.00      0.00      0.00         8
       11.0       0.06      0.10      0.08        10
       12.0       0.03      0.33      0.05         3
       13.0       0.00      0.00      0.00         4
       14.0       0.00      0.00      0.00         4
       15.0       0.00      0.00      0.00         2
       16.0       0.00      0.00      0.00         3
       17.0       0.00      0.00      0.00         3
       18.0       0.00      0.00      0.00         2
       19.0       0.00      0.00      0.00         1

avg / total       0.01      0.04      0.02        51

--------------------------------------------------------------------
```

```
------------
Accuracy 3.92 %
Lars %error 19.1 rmsle 0.25835084881532167
Lars Confusion Matrix
[[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 4 4 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 3 6 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 2 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       0.00      0.00      0.00         1
        5.0       0.00      0.00      0.00         2
        6.0       0.00      0.00      0.00         2
        7.0       0.00      0.00      0.00         1
        8.0       0.00      0.00      0.00         3
        9.0       0.00      0.00      0.00         2
       10.0       0.00      0.00      0.00         8
       11.0       0.19      0.30      0.23        10
       12.0       0.06      0.33      0.10         3
       13.0       0.12      0.25      0.17         4
       14.0       0.00      0.00      0.00         4
       15.0       0.00      0.00      0.00         2
       16.0       0.00      0.00      0.00         3
       17.0       0.00      0.00      0.00         3
       18.0       0.00      0.00      0.00         2
       19.0       0.00      0.00      0.00         1

avg / total       0.05      0.10      0.06        51

--------------------------------------------------------------------
------------
Accuracy 9.8 %
SGDClassifier %error 25.8 rmsle 0.3227944909108929
SGDClassifier Confusion Matrix
[[ 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0]
```

```
 [ 0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 10  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       0.00      0.00      0.00         1
        5.0       0.00      0.00      0.00         2
        6.0       0.00      0.00      0.00         2
        7.0       0.00      0.00      0.00         1
        8.0       0.00      0.00      0.00         3
        9.0       0.00      0.00      0.00         2
       10.0       0.16      1.00      0.27         8
       11.0       0.00      0.00      0.00        10
       12.0       0.00      0.00      0.00         3
       13.0       0.00      0.00      0.00         4
       14.0       0.00      0.00      0.00         4
       15.0       0.00      0.00      0.00         2
       16.0       0.00      0.00      0.00         3
       17.0       0.00      0.00      0.00         3
       18.0       0.00      0.00      0.00         2
       19.0       0.00      0.00      0.00         1

avg / total       0.02      0.16      0.04        51

--------------------------------------------------------------------
------------
Accuracy 15.69 %
RidgeClassifier %error 0.0 rmsle 0.0
RidgeClassifier Confusion Matrix
[[ 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0]
```

```
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
--------------------------------------------------------------------
------------
Classification Report
             precision    recall  f1-score   support

        4.0       1.00      1.00      1.00         1
        5.0       1.00      1.00      1.00         2
        6.0       1.00      1.00      1.00         2
        7.0       1.00      1.00      1.00         1
        8.0       1.00      1.00      1.00         3
        9.0       1.00      1.00      1.00         2
       10.0       1.00      1.00      1.00         8
       11.0       1.00      1.00      1.00        10
       12.0       1.00      1.00      1.00         3
       13.0       1.00      1.00      1.00         4
       14.0       1.00      1.00      1.00         4
       15.0       1.00      1.00      1.00         2
       16.0       1.00      1.00      1.00         3
       17.0       1.00      1.00      1.00         3
       18.0       1.00      1.00      1.00         2
       19.0       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00        51

--------------------------------------------------------------------
------------
Accuracy 100.0 %

/anaconda3/lib/python3.6/site-packages/sklearn/metrics/classificatio
n.py:1135: UndefinedMetricWarning: Precision and F-score are ill-def
ined and being set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
/anaconda3/lib/python3.6/site-packages/sklearn/linear_model/stochast
ic_gradient.py:128: FutureWarning: max_iter and tol parameters have
been added in <class 'sklearn.linear_model.stochastic_gradient.SGDCl
assifier'> in 0.19. If both are left unset, they default to max_iter
=5 and tol=None. If tol is not None, max_iter defaults to max_iter=1
000. From 0.21, default max_iter will be 1000, and default tol will
be 1e-3.
  "and default tol will be 1e-3." % type(self), FutureWarning)
```