

Щербинин Андрей РТ5-61Б РК1 Вариант 22 Задание: Для заданного набора данных (по Вашему варианту) постройте модели классификации или регрессии (в зависимости от конкретной задачи, рассматриваемой в наборе данных). Для построения моделей используйте методы 1 и 2 (по варианту для Вашей группы). Оцените качество моделей на основе подходящих метрик качества (не менее двух метрик).

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_absolute_error, r2_score,
mean_squared_error
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
import numpy as np
```

```
data = pd.read_csv('data.csv')
data.head()
```

	order	name	height_cm	height_in	weight_kg	weight_lb
\						
0	1	George Washington	188	74.0	79.4	175
1	2	John Adams	170	67.0	83.9	185
2	3	Thomas Jefferson	189	74.5	82.1	181
3	4	James Madison	163	64.0	55.3	122
4	5	James Monroe	183	72.0	85.7	189

	body_mass_index	body_mass_index_range	birth_day	birth_month	...
\					
0	22.5	Normal	22	2	...
1	29.0	Overweight	30	10	...
2	23.0	Normal	13	4	...
3	20.8	Normal	16	3	...
4	25.6	Overweight	28	4	...

	term_begin_year	term_begin_date	term_end_day	term_end_month
term_end_year				
\				
0	1789	30-04-1789	4.0	3.0

```

1797.0
1      1797      04-03-1797      4.0      3.0
1801.0
2      1801      04-03-1801      4.0      3.0
1809.0
3      1809      04-03-1809      4.0      3.0
1817.0
4      1817      04-03-1817      4.0      3.0
1825.0

```

```

      term_end_date  presidency_begin_age  presidency_end_age  \
0      04-03-1797          57          65.0
1      04-03-1801          61          65.0
2      04-03-1809          57          65.0
3      04-03-1817          57          65.0
4      04-03-1825          58          66.0

```

```

      political_party  corrected_iq
0      Unaffiliated      140.0
1      Federalist      155.0
2      Democratic-Republican      160.0
3      Democratic-Republican      160.0
4      Democratic-Republican      139.0

```

```
[5 rows x 32 columns]
```

```
data.isnull().sum()
```

```

order      0
name      0
height_cm  0
height_in  0
weight_kg  0
weight_lb  0
body_mass_index  0
body_mass_index_range  0
birth_day  0
birth_month  0
birth_year  0
birth_date  0
birthplace  0
birth_state  0
death_day  6
death_month  6
death_year  6
death_date  6
death_age  6
astrological_sign  0
term_begin_day  0
term_begin_month  0

```

```
term_begin_year      0
term_begin_date      0
term_end_day         1
term_end_month       1
term_end_year        1
term_end_date        1
presidency_begin_age 0
presidency_end_age   1
political_party       0
corrected_iq         3
dtype: int64
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 45 entries, 0 to 44
```

```
Data columns (total 32 columns):
```

#	Column	Non-Null Count	Dtype
0	order	45 non-null	object
1	name	45 non-null	object
2	height_cm	45 non-null	int64
3	height_in	45 non-null	float64
4	weight_kg	45 non-null	float64
5	weight_lb	45 non-null	int64
6	body_mass_index	45 non-null	float64
7	body_mass_index_range	45 non-null	object
8	birth_day	45 non-null	int64
9	birth_month	45 non-null	int64
10	birth_year	45 non-null	int64
11	birth_date	45 non-null	object
12	birthplace	45 non-null	object
13	birth_state	45 non-null	object
14	death_day	39 non-null	float64
15	death_month	39 non-null	float64
16	death_year	39 non-null	float64
17	death_date	39 non-null	object
18	death_age	39 non-null	float64
19	astrological_sign	45 non-null	object
20	term_begin_day	45 non-null	int64
21	term_begin_month	45 non-null	int64
22	term_begin_year	45 non-null	int64
23	term_begin_date	45 non-null	object
24	term_end_day	44 non-null	float64
25	term_end_month	44 non-null	float64
26	term_end_year	44 non-null	float64
27	term_end_date	44 non-null	object
28	presidency_begin_age	45 non-null	int64
29	presidency_end_age	44 non-null	float64
30	political_party	45 non-null	object

```
31 corrected_iq          42 non-null    float64
dtypes: float64(12), int64(9), object(11)
memory usage: 11.4+ KB
```

```
data.shape
```

```
(45, 32)
```

Перед нами задача регрессии, в предсказании IQ президента. Удалим строки, где нет corrected_iq которых всего 3

```
data = data.dropna(subset=['corrected_iq'])
data.isnull().sum()
```

```
order          0
name           0
height_cm      0
height_in      0
weight_kg      0
weight_lb      0
body_mass_index 0
body_mass_index_range 0
birth_day      0
birth_month    0
birth_year     0
birth_date     0
birthplace     0
birth_state    0
death_day      3
death_month    3
death_year     3
death_date     3
death_age      3
astrological_sign 0
term_begin_day 0
term_begin_month 0
term_begin_year 0
term_begin_date 0
term_end_day   0
term_end_month 0
term_end_year  0
term_end_date  0
presidency_begin_age 0
presidency_end_age 0
political_party 0
corrected_iq   0
dtype: int64
```

```
si = SimpleImputer(strategy='most_frequent')
data[['death_day', 'death_month',
```

```

    'death_year', 'death_age']] = si.fit_transform(data[['death_day',
'death_month', 'death_year', 'death_age']])
data.isnull().sum()

```

```

order                0
name                 0
height_cm            0
height_in            0
weight_kg            0
weight_lb            0
body_mass_index      0
body_mass_index_range 0
birth_day            0
birth_month          0
birth_year           0
birth_date           0
birthplace           0
birth_state          0
death_day            0
death_month          0
death_year           0
death_date           3
death_age            0
astrological_sign    0
term_begin_day       0
term_begin_month     0
term_begin_year      0
term_begin_date      0
term_end_day         0
term_end_month       0
term_end_year        0
term_end_date        0
presidency_begin_age 0
presidency_end_age   0
political_party       0
corrected_iq         0
dtype: int64

```

```

for col in data.columns:
    print(f"Уникальные значения в колонке '{col}':")
    print(data[col].unique())
    print()

```

Уникальные значения в колонке 'order':

```

['1' '2' '3' '4' '5' '6' '7' '8' '9' '10' '11' '12' '13' '14' '15'
'16'

```

```

'17' '18' '19' '20' '21' '22 & 24' '23' '25' '26' '27' '28' '29' '30'
'31' '32' '33' '34' '35' '36' '37' '38' '39' '40' '41' '42' '43']

```

Уникальные значения в колонке 'name':

```
['George Washington' 'John Adams' 'Thomas Jefferson' 'James Madison'
'James Monroe' 'John Quincy Adams' 'Andrew Jackson' 'Martin Van
Buren'
'William Henry Harrison' 'John Tyler' 'James Knox Polk' 'Zachary
Taylor'
'Millard Fillmore' 'Franklin Pierce' 'James Buchanan' 'Abraham
Lincoln'
'Andrew Johnson' 'Ulysses Simpson Grant' 'Rutherford Birchard Hayes'
'James Abram Garfield' 'Chester Alan Arthur' 'Stephen Grover
Cleveland'
'Benjamin Harrison' 'William McKinley' 'Theodore Roosevelt'
'William Howard Taft' 'Thomas Woodrow Wilson' 'Warren Gamal Harding'
'John Calvin Coolidge' 'Herbert Clark Hoover' 'Franklin Delano
Roosevelt'
'Harry S Truman' 'Dwight David Eisenhower' 'John Fitzgerald Kennedy'
'Lyndon Baines Johnson' 'Richard Milhous Nixon' 'Gerald Rudolph Ford'
'James Earl Carter' 'Ronald Wilson Reagan' 'George Herbert Walker
Bush'
'William Jefferson Clinton' 'George Walker Bush']
```

Уникальные значения в колонке 'height_cm':

```
[188 170 189 163 183 171 185 168 173 175 178 193 174 180 182 179 192
177]
```

Уникальные значения в колонке 'height_in':

```
[74. 67. 74.5 64. 72. 67.5 73. 66. 68. 69. 70. 76. 68.5 71.
71.5 70.5 75.5 69.5]
```

Уникальные значения в колонке 'weight_kg':

```
[ 79.4 83.9 82.1 55.3 85.7 92.1 63.5 74.4 73.5 72.6 78.9
104.3
65.3 98.4 69.9 70.8 90.7 101.6 117.9 90.3 107.5 154.2 77.1
78.5
84.8 86.2 75.7 78. 95.3 80.3 81.6 88.5 101.2 84.4]
```

Уникальные значения в колонке 'weight_lb':

```
[175 185 181 122 189 203 140 164 162 160 174 230 144 217 154 156 200
224
260 199 237 340 170 173 187 190 167 172 210 177 180 195 223 186]
```

Уникальные значения в колонке 'body_mass_index':

```
[22.5 29. 23. 20.8 25.6 31.5 18.6 26.4 24.6 21.7 34.8 21.3 23.2 29.4
22.1 23.7 26.2 27.1 28.7 36.4 25.7 31.2 33.9 46.6 23.8 23.4 22.3 24.4
24.7 24.3 25.9 24. 25. 28.6 25.5]
```

Уникальные значения в колонке 'body_mass_index_range':

```
['Normal' 'Overweight' 'Obese' 'Severely Obese' 'Morbidly Obese']
```

Уникальные значения в колонке 'birth_day':

```
[22 30 13 16 28 11 15 5 9 29 2 24 7 23 12 27 4 19 18 20 10 8 14
```

1

6]

Уникальные значения в колонке 'birth_month':

[2 10 4 3 7 12 11 1 8 9 5 6]

Уникальные значения в колонке 'birth_year':

[1732 1735 1743 1751 1758 1767 1782 1773 1790 1795 1784 1800 1804 1791
1809 1808 1822 1831 1829 1837 1833 1843 1858 1857 1856 1865 1872 1874
1882 1884 1890 1917 1908 1913 1924 1911 1946]

Уникальные значения в колонке 'birth_date':

['22-02-1732' '30-10-1735' '13-04-1743' '16-03-1751' '28-04-1758'
'11-07-1767' '15-03-1767' '05-12-1782' '09-02-1773' '29-03-1790'
'02-11-1795' '24-11-1784' '07-01-1800' '23-11-1804' '23-04-1791'
'12-02-1809' '29-12-1808' '27-04-1822' '04-10-1822' '19-11-1831'
'05-10-1829' '18-03-1837' '20-08-1833' '29-01-1843' '27-10-1858'
'15-09-1857' '28-12-1856' '02-11-1865' '04-07-1872' '10-08-1874'
'30-01-1882' '08-05-1884' '14-10-1890' '29-05-1917' '27-08-1908'
'09-01-1913' '14-07-1913' '01-10-1924' '06-02-1911' '12-06-1924'
'19-08-1946' '06-06-1946']

Уникальные значения в колонке 'birthplace':

['Westmoreland County' 'Braintree' 'Shadwell' 'Port Conway' 'Monroe
Hall'
'Waxhaws Region' 'Kinderhook' 'Charles City County' 'Pineville'
'Barboursville' 'Summerhill' 'Hillsborough' 'Cove Gap' 'Sinking
Spring'
'Raleigh' 'Point Pleasant' 'Delaware' 'Moreland Hills' 'Fairfield'
'Caldwell' 'North Bend' 'Niles' 'Manhattan' 'Cincinnati' 'Staunton'
'Blooming Grove' 'Plymouth' 'West Branch' 'Hyde Park' 'Lamar'
'Denison'
'Brookline' 'Stonewall' 'Yorba Linda' 'Omaha' 'Plains' 'Tampico'
'Milton'
'Hope' 'New Haven']

Уникальные значения в колонке 'birth_state':

['Virginia' 'Massachusetts' 'South Carolina' 'New York' 'North
Carolina'
'New Hampshire' 'Pennsylvania' 'Kentucky' 'Ohio' 'Vermont' 'New
Jersey'
'Iowa' 'Missouri' 'Texas' 'California' 'Nebraska' 'Georgia'
'Illinois'
'Arkansas' 'Connecticut']

Уникальные значения в колонке 'death_day':

[14. 4. 28. 23. 8. 24. 18. 15. 9. 1. 31. 17. 19. 13. 6. 3. 2.
5.
20. 12. 26. 22. 30.]

Уникальные значения в колонке 'death_month':

[12. 7. 6. 2. 4. 1. 5. 10. 9. 11. 3. 8.]

Уникальные значения в колонке 'death_year':

[1799. 1826. 1836. 1831. 1848. 1845. 1862. 1841. 1849. 1850. 1874.
1869.
1868. 1865. 1875. 1885. 1893. 1881. 1886. 1908. 1901. 1919. 1930.
1924.
1923. 1933. 1964. 1945. 1972. 1969. 1963. 1973. 1994. 2006. 2004.
2018.]

Уникальные значения в колонке 'death_date':

['14-12-1799' '04-07-1826' '28-06-1836' '04-07-1831' '23-02-1848'
'08-06-1845' '24-07-1862' '04-04-1841' '18-01-1862' '15-06-1849'
'09-07-1850' '08-05-1874' '08-10-1869' '01-06-1868' '15-04-1865'
'31-07-1875' '23-07-1885' '17-01-1893' '19-09-1881' '18-11-1886'
'24-06-1908' '13-03-1901' '14-09-1901' '06-01-1919' '08-03-1930'
'03-02-1924' '02-08-1923' '05-01-1933' '20-10-1964' '12-04-1945'
'26-12-1972' '28-03-1969' '22-11-1963' '22-01-1973' '22-04-1994'
'26-12-2006' nan '05-06-2004' '30-12-2018']

Уникальные значения в колонке 'death_age':

[67. 90. 83. 85. 73. 80. 78. 79. 68. 71. 53. 65. 74. 64. 77. 56. 66.
63.
70. 49. 57. 58. 60. 72. 88. 46. 81. 93. 94.]

Уникальные значения в колонке 'astrological_sign':

['Pisces' 'Scorpio' 'Aries' 'Taurus' 'Cancer' 'Sagittarius' 'Aquarius'
'Capricorn' 'Libran' 'Leo' 'Virgo' 'Gemini']

Уникальные значения в колонке 'term_begin_day':

[30 4 9 15 19 14 2 12 20 22]

Уникальные значения в колонке 'term_begin_month':

[4 3 7 9 8 1 11]

Уникальные значения в колонке 'term_begin_year':

[1789 1797 1801 1809 1817 1825 1829 1837 1841 1845 1849 1850 1853 1857
1861 1865 1869 1877 1881 1885 1889 1897 1901 1909 1913 1921 1923 1929
1933 1945 1953 1961 1963 1969 1974 1977 1981 1989 1993 2001]

Уникальные значения в колонке 'term_begin_date':

['30-04-1789' '04-03-1797' '04-03-1801' '04-03-1809' '04-03-1817'
'04-03-1825' '04-03-1829' '04-03-1837' '04-03-1841' '04-04-1841'
'04-03-1845' '04-03-1849' '09-07-1850' '04-03-1853' '04-03-1857'
'04-03-1861' '15-04-1865' '04-03-1869' '04-03-1877' '04-03-1881'
'19-09-1881' '04-03-1885' '04-03-1889' '04-03-1897' '14-09-1901'
'04-03-1909' '04-03-1913' '04-03-1921' '02-08-1923' '04-03-1929'
'04-03-1933' '12-04-1945' '20-01-1953' '20-01-1961' '22-11-1963'
'20-01-1969' '09-08-1974' '20-01-1977' '20-01-1981' '20-01-1989']


```
'20-01-1993' '20-01-2001']
```

Уникальные значения в колонке 'term_end_day':

```
[ 4.  9. 15. 19. 14.  2. 12. 20. 22.]
```

Уникальные значения в колонке 'term_end_month':

```
[ 3.  4.  7.  9.  8.  1. 11.]
```

Уникальные значения в колонке 'term_end_year':

```
[1797. 1801. 1809. 1817. 1825. 1829. 1837. 1841. 1845. 1849. 1850.  
1853.  
1857. 1861. 1865. 1869. 1877. 1881. 1885. 1897. 1893. 1901. 1909.  
1913.  
1921. 1923. 1929. 1933. 1945. 1953. 1961. 1963. 1969. 1974. 1977.  
1981.  
1989. 1993. 2001. 2009.]
```

Уникальные значения в колонке 'term_end_date':

```
['04-03-1797' '04-03-1801' '04-03-1809' '04-03-1817' '04-03-1825'  
'04-03-1829' '04-03-1837' '04-03-1841' '04-04-1841' '04-03-1845'  
'04-03-1849' '09-07-1850' '04-03-1853' '04-03-1857' '04-03-1861'  
'15-04-1865' '04-03-1869' '04-03-1877' '04-03-1881' '19-09-1881'  
'04-03-1885' '04-03-1897' '04-03-1893' '14-09-1901' '04-03-1909'  
'04-03-1913' '04-03-1921' '02-08-1923' '04-03-1929' '04-03-1933'  
'12-04-1945' '20-01-1953' '20-01-1961' '22-11-1963' '20-01-1969'  
'09-08-1974' '20-01-1977' '20-01-1981' '20-01-1989' '20-01-1993'  
'20-01-2001' '20-01-2009']
```

Уникальные значения в колонке 'presidency_begin_age':

```
[57 61 58 54 68 51 49 64 50 48 65 52 56 46 47 55 42 60 62 43 69]
```

Уникальные значения в колонке 'presidency_end_age':

```
[65. 66. 61. 69. 58. 68. 54. 53. 52. 56. 60. 49. 55. 59. 50. 64. 57.  
63.  
70. 46. 77.]
```

Уникальные значения в колонке 'political_party':

```
['Unaffiliated' 'Federalist' 'Democratic-Republican' 'Democrat' 'Whig'  
'Republican' 'National Union']
```

Уникальные значения в колонке 'corrected_iq':

```
[140. 155. 160. 139. 175. 145. 146. 148. 143. 149. 147. 150. 130. 152.  
144. 153. 142. 151. 141. 157. 159.]
```

Даты (дни месяцы годы) можно оставить как есть. Удалим ненужные столбцы для модели

```
data = data.drop(['name', 'birth_date', 'death_date',  
'term_begin_date', 'term_end_date'], axis=1)  
for col in data.columns:
```

```
print(f"Уникальные значения в колонке '{col}':")
print(data[col].unique())
print()
```

Уникальные значения в колонке 'order':

```
['1' '2' '3' '4' '5' '6' '7' '8' '9' '10' '11' '12' '13' '14' '15'
'16'
'17' '18' '19' '20' '21' '22 & 24' '23' '25' '26' '27' '28' '29' '30'
'31' '32' '33' '34' '35' '36' '37' '38' '39' '40' '41' '42' '43']
```

Уникальные значения в колонке 'height_cm':

```
[188 170 189 163 183 171 185 168 173 175 178 193 174 180 182 179 192
177]
```

Уникальные значения в колонке 'height_in':

```
[74.  67.  74.5 64.  72.  67.5 73.  66.  68.  69.  70.  76.  68.5 71.
71.5 70.5 75.5 69.5]
```

Уникальные значения в колонке 'weight_kg':

```
[ 79.4  83.9  82.1  55.3  85.7  92.1  63.5  74.4  73.5  72.6  78.9
104.3
 65.3  98.4  69.9  70.8  90.7 101.6 117.9  90.3 107.5 154.2  77.1
78.5
 84.8  86.2  75.7  78.   95.3  80.3  81.6  88.5 101.2  84.4]
```

Уникальные значения в колонке 'weight_lb':

```
[175 185 181 122 189 203 140 164 162 160 174 230 144 217 154 156 200
224
260 199 237 340 170 173 187 190 167 172 210 177 180 195 223 186]
```

Уникальные значения в колонке 'body_mass_index':

```
[22.5 29.  23.  20.8 25.6 31.5 18.6 26.4 24.6 21.7 34.8 21.3 23.2 29.4
22.1 23.7 26.2 27.1 28.7 36.4 25.7 31.2 33.9 46.6 23.8 23.4 22.3 24.4
24.7 24.3 25.9 24.  25.  28.6 25.5]
```

Уникальные значения в колонке 'body_mass_index_range':

```
['Normal' 'Overweight' 'Obese' 'Severely Obese' 'Morbidly Obese']
```

Уникальные значения в колонке 'birth_day':

```
[22 30 13 16 28 11 15  5  9 29  2 24  7 23 12 27  4 19 18 20 10  8 14
1
 6]
```

Уникальные значения в колонке 'birth_month':

```
[ 2 10  4  3  7 12 11  1  8  9  5  6]
```

Уникальные значения в колонке 'birth_year':

```
[1732 1735 1743 1751 1758 1767 1782 1773 1790 1795 1784 1800 1804 1791
1809 1808 1822 1831 1829 1837 1833 1843 1858 1857 1856 1865 1872 1874
1882 1884 1890 1917 1908 1913 1924 1911 1946]
```

Уникальные значения в колонке 'birthplace':

['Westmoreland County' 'Braintree' 'Shadwell' 'Port Conway' 'Monroe Hall'
'Waxhaws Region' 'Kinderhook' 'Charles City County' 'Pineville'
'Barboursville' 'Summerhill' 'Hillsborough' 'Cove Gap' 'Sinking Spring'
'Raleigh' 'Point Pleasant' 'Delaware' 'Moreland Hills' 'Fairfield'
'Caldwell' 'North Bend' 'Niles' 'Manhattan' 'Cincinnati' 'Staunton'
'Blooming Grove' 'Plymouth' 'West Branch' 'Hyde Park' 'Lamar'
'Denison'
'Brookline' 'Stonewall' 'Yorba Linda' 'Omaha' 'Plains' 'Tampico'
'Milton'
'Hope' 'New Haven']

Уникальные значения в колонке 'birth_state':

['Virginia' 'Massachusetts' 'South Carolina' 'New York' 'North Carolina'
'New Hampshire' 'Pennsylvania' 'Kentucky' 'Ohio' 'Vermont' 'New Jersey'
'Iowa' 'Missouri' 'Texas' 'California' 'Nebraska' 'Georgia'
'Illinois'
'Arkansas' 'Connecticut']

Уникальные значения в колонке 'death_day':

[14. 4. 28. 23. 8. 24. 18. 15. 9. 1. 31. 17. 19. 13. 6. 3. 2. 5.
20. 12. 26. 22. 30.]

Уникальные значения в колонке 'death_month':

[12. 7. 6. 2. 4. 1. 5. 10. 9. 11. 3. 8.]

Уникальные значения в колонке 'death_year':

[1799. 1826. 1836. 1831. 1848. 1845. 1862. 1841. 1849. 1850. 1874. 1869.
1868. 1865. 1875. 1885. 1893. 1881. 1886. 1908. 1901. 1919. 1930. 1924.
1923. 1933. 1964. 1945. 1972. 1969. 1963. 1973. 1994. 2006. 2004. 2018.]

Уникальные значения в колонке 'death_age':

[67. 90. 83. 85. 73. 80. 78. 79. 68. 71. 53. 65. 74. 64. 77. 56. 66. 63.
70. 49. 57. 58. 60. 72. 88. 46. 81. 93. 94.]

Уникальные значения в колонке 'astrological_sign':

['Pisces' 'Scorpio' 'Aries' 'Taurus' 'Cancer' 'Sagittarius' 'Aquarius'
'Capricorn' 'Libran' 'Leo' 'Virgo' 'Gemini']

Уникальные значения в колонке 'term_begin_day':

```
[30  4  9 15 19 14  2 12 20 22]
```

Уникальные значения в колонке 'term_begin_month':

```
[ 4  3  7  9  8  1 11]
```

Уникальные значения в колонке 'term_begin_year':

```
[1789 1797 1801 1809 1817 1825 1829 1837 1841 1845 1849 1850 1853 1857  
1861 1865 1869 1877 1881 1885 1889 1897 1901 1909 1913 1921 1923 1929  
1933 1945 1953 1961 1963 1969 1974 1977 1981 1989 1993 2001]
```

Уникальные значения в колонке 'term_end_day':

```
[ 4.  9. 15. 19. 14.  2. 12. 20. 22.]
```

Уникальные значения в колонке 'term_end_month':

```
[ 3.  4.  7.  9.  8.  1. 11.]
```

Уникальные значения в колонке 'term_end_year':

```
[1797. 1801. 1809. 1817. 1825. 1829. 1837. 1841. 1845. 1849. 1850.  
1853.  
1857. 1861. 1865. 1869. 1877. 1881. 1885. 1897. 1893. 1901. 1909.  
1913.  
1921. 1923. 1929. 1933. 1945. 1953. 1961. 1963. 1969. 1974. 1977.  
1981.  
1989. 1993. 2001. 2009.]
```

Уникальные значения в колонке 'presidency_begin_age':

```
[57 61 58 54 68 51 49 64 50 48 65 52 56 46 47 55 42 60 62 43 69]
```

Уникальные значения в колонке 'presidency_end_age':

```
[65. 66. 61. 69. 58. 68. 54. 53. 52. 56. 60. 49. 55. 59. 50. 64. 57.  
63.  
70. 46. 77.]
```

Уникальные значения в колонке 'political_party':

```
['Unaffiliated' 'Federalist' 'Democratic-Republican' 'Democrat' 'Whig'  
'Republican' 'National Union']
```

Уникальные значения в колонке 'corrected_iq':

```
[140. 155. 160. 139. 175. 145. 146. 148. 143. 149. 147. 150. 130. 152.  
144. 153. 142. 151. 141. 157. 159.]
```

Закодируем города и штаты рождения президентов, а также их знак зодиака и партию.

```
le = LabelEncoder()  
coded = data.copy()  
  
for col in ['birthplace', 'birth_state', 'astrological_sign',  
'political_party', 'order', 'body_mass_index_range']:  
    coded[col] = le.fit_transform(coded[col])
```

```
for col in coded.columns:  
    print(f"Уникальные значения в колонке '{col}':")  
    print(coded[col].unique())  
    print()
```

Уникальные значения в колонке 'order':

```
[ 0 11 21 32 37 38 39 40 41  1  2  3  4  5  6  7  8  9 10 12 13 14 15  
16  
17 18 19 20 22 23 24 25 26 27 28 29 30 31 33 34 35 36]
```

Уникальные значения в колонке 'height_cm':

```
[188 170 189 163 183 171 185 168 173 175 178 193 174 180 182 179 192  
177]
```

Уникальные значения в колонке 'height_in':

```
[74.  67.  74.5 64.  72.  67.5 73.  66.  68.  69.  70.  76.  68.5 71.  
71.5 70.5 75.5 69.5]
```

Уникальные значения в колонке 'weight_kg':

```
[ 79.4  83.9  82.1  55.3  85.7  92.1  63.5  74.4  73.5  72.6  78.9  
104.3  
 65.3  98.4  69.9  70.8  90.7 101.6 117.9  90.3 107.5 154.2  77.1  
78.5  
 84.8  86.2  75.7  78.   95.3  80.3  81.6  88.5 101.2  84.4]
```

Уникальные значения в колонке 'weight_lb':

```
[175 185 181 122 189 203 140 164 162 160 174 230 144 217 154 156 200  
224  
260 199 237 340 170 173 187 190 167 172 210 177 180 195 223 186]
```

Уникальные значения в колонке 'body_mass_index':

```
[22.5 29.  23.  20.8 25.6 31.5 18.6 26.4 24.6 21.7 34.8 21.3 23.2 29.4  
22.1 23.7 26.2 27.1 28.7 36.4 25.7 31.2 33.9 46.6 23.8 23.4 22.3 24.4  
24.7 24.3 25.9 24.  25.  28.6 25.5]
```

Уникальные значения в колонке 'body_mass_index_range':

```
[1 3 2 4 0]
```

Уникальные значения в колонке 'birth_day':

```
[22 30 13 16 28 11 15  5  9 29  2 24  7 23 12 27  4 19 18 20 10  8 14  
1  
 6]
```

Уникальные значения в колонке 'birth_month':

```
[ 2 10  4  3  7 12 11  1  8  9  5  6]
```

Уникальные значения в колонке 'birth_year':

```
[1732 1735 1743 1751 1758 1767 1782 1773 1790 1795 1784 1800 1804 1791  
1809 1808 1822 1831 1829 1837 1833 1843 1858 1857 1856 1865 1872 1874]
```

1882 1884 1890 1917 1908 1913 1924 1911 1946]

Уникальные значения в колонке 'birthplace':

[38 2 30 28 18 36 14 5 24 0 34 11 7 31 29 27 8 19 10 4 22 21 16
6
32 1 26 37 13 15 9 3 33 39 23 25 35 17 12 20]

Уникальные значения в колонке 'birth_state':

[19 7 16 12 13 10 15 6 14 18 11 5 8 17 1 9 3 4 0 2]

Уникальные значения в колонке 'death_day':

[14. 4. 28. 23. 8. 24. 18. 15. 9. 1. 31. 17. 19. 13. 6. 3. 2.
5.
20. 12. 26. 22. 30.]

Уникальные значения в колонке 'death_month':

[12. 7. 6. 2. 4. 1. 5. 10. 9. 11. 3. 8.]

Уникальные значения в колонке 'death_year':

[1799. 1826. 1836. 1831. 1848. 1845. 1862. 1841. 1849. 1850. 1874.
1869.
1868. 1865. 1875. 1885. 1893. 1881. 1886. 1908. 1901. 1919. 1930.
1924.
1923. 1933. 1964. 1945. 1972. 1969. 1963. 1973. 1994. 2006. 2004.
2018.]

Уникальные значения в колонке 'death_age':

[67. 90. 83. 85. 73. 80. 78. 79. 68. 71. 53. 65. 74. 64. 77. 56. 66.
63.
70. 49. 57. 58. 60. 72. 88. 46. 81. 93. 94.]

Уникальные значения в колонке 'astrological_sign':

[7 9 1 10 2 8 0 3 6 5 11 4]

Уникальные значения в колонке 'term_begin_day':

[30 4 9 15 19 14 2 12 20 22]

Уникальные значения в колонке 'term_begin_month':

[4 3 7 9 8 1 11]

Уникальные значения в колонке 'term_begin_year':

[1789 1797 1801 1809 1817 1825 1829 1837 1841 1845 1849 1850 1853 1857
1861 1865 1869 1877 1881 1885 1889 1897 1901 1909 1913 1921 1923 1929
1933 1945 1953 1961 1963 1969 1974 1977 1981 1989 1993 2001]

Уникальные значения в колонке 'term_end_day':

[4. 9. 15. 19. 14. 2. 12. 20. 22.]

Уникальные значения в колонке 'term_end_month':

[3. 4. 7. 9. 8. 1. 11.]

Уникальные значения в колонке 'term_end_year':
[1797. 1801. 1809. 1817. 1825. 1829. 1837. 1841. 1845. 1849. 1850.
1853.
1857. 1861. 1865. 1869. 1877. 1881. 1885. 1897. 1893. 1901. 1909.
1913.
1921. 1923. 1929. 1933. 1945. 1953. 1961. 1963. 1969. 1974. 1977.
1981.
1989. 1993. 2001. 2009.]

Уникальные значения в колонке 'presidency_begin_age':
[57 61 58 54 68 51 49 64 50 48 65 52 56 46 47 55 42 60 62 43 69]

Уникальные значения в колонке 'presidency_end_age':
[65. 66. 61. 69. 58. 68. 54. 53. 52. 56. 60. 49. 55. 59. 50. 64. 57.
63.
70. 46. 77.]

Уникальные значения в колонке 'political_party':
[5 2 1 0 6 4 3]

Уникальные значения в колонке 'corrected_iq':
[140. 155. 160. 139. 175. 145. 146. 148. 143. 149. 147. 150. 130. 152.
144. 153. 142. 151. 141. 157. 159.]

Займемся масштабированием с MinMaxScaler

coded.dtypes

order	int64
height_cm	int64
height_in	float64
weight_kg	float64
weight_lb	int64
body_mass_index	float64
body_mass_index_range	int64
birth_day	int64
birth_month	int64
birth_year	int64
birthplace	int64
birth_state	int64
death_day	float64
death_month	float64
death_year	float64
death_age	float64
astrological_sign	int64
term_begin_day	int64
term_begin_month	int64
term_begin_year	int64

```
term_end_day          float64
term_end_month        float64
term_end_year         float64
presidency_begin_age  int64
presidency_end_age    float64
political_party       int64
corrected_iq          float64
dtype: object
```

```
scaler = MinMaxScaler()
```

```
scaled_and_coded = coded.copy()
numeric_cols = scaled_and_coded.select_dtypes(include=['int64',
'float64']).columns
scaled_and_coded[numeric_cols] =
MinMaxScaler().fit_transform(scaled_and_coded[numeric_cols])
scaled_and_coded.head()
```

```
      order  height_cm  height_in  weight_kg  weight_lb
body_mass_index \
0  0.000000  0.833333  0.833333  0.243680  0.243119
0.139286
1  0.268293  0.233333  0.250000  0.289181  0.288991
0.371429
2  0.512195  0.866667  0.875000  0.270981  0.270642
0.157143
3  0.780488  0.000000  0.000000  0.000000  0.000000
0.078571
4  0.902439  0.666667  0.666667  0.307381  0.307339
0.250000
```

```
      body_mass_index_range  birth_day  birth_month  birth_year  ... \
0          0.25  0.724138  0.090909  0.000000  ...
1          0.75  1.000000  0.818182  0.014019  ...
2          0.25  0.413793  0.272727  0.051402  ...
3          0.25  0.517241  0.181818  0.088785  ...
4          0.75  0.931034  0.272727  0.121495  ...
```

```
      term_begin_day  term_begin_month  term_begin_year  term_end_day \
0          1.000000  0.3  0.000000  0.1
1          0.071429  0.2  0.037736  0.1
2          0.071429  0.2  0.056604  0.1
3          0.071429  0.2  0.094340  0.1
4          0.071429  0.2  0.132075  0.1
```

```
      term_end_month  term_end_year  presidency_begin_age
presidency_end_age \
0          0.2  0.000000  0.555556
0.612903
1          0.2  0.018868  0.703704
```


0.612903			
2	0.2	0.056604	0.555556
0.612903			
3	0.2	0.094340	0.555556
0.612903			
4	0.2	0.132075	0.592593
0.645161			

	political_party	corrected_iq
0	0.833333	0.222222
1	0.333333	0.555556
2	0.166667	0.666667
3	0.166667	0.666667
4	0.166667	0.200000

[5 rows x 27 columns]

Разделим данные

```
# Целевая переменная
target = 'corrected_iq'
X = coded.drop(columns=[target], axis=1)
y = coded[target]

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=1)

print("Размер обучающей выборки X:", X_train.shape)
print("Размер тестовой выборки X:", X_test.shape)
print("Размер обучающей выборки y:", y_train.shape)
print("Размер тестовой выборки y:", y_test.shape)

Размер обучающей выборки X: (33, 26)
Размер тестовой выборки X: (9, 26)
Размер обучающей выборки y: (33,)
Размер тестовой выборки y: (9,)
```

Обучаем модели. Используем дерево и градиентный бустинг.

```
#Решающее дерево
dt_model = DecisionTreeRegressor(random_state=1)
dt_model.fit(X_train, y_train)
y_pred_dt = dt_model.predict(X_test)

#Градиентный бустинг
gb_model = GradientBoostingRegressor(random_state=1)
gb_model.fit(X_train, y_train)
y_pred_gb = gb_model.predict(X_test)
```

Посмотрим на метрики

```
def print_metrics(y_true, y_pred, model_name):  
    print(f"\nМодель: {model_name}")  
    print("MAE: ", mean_absolute_error(y_true, y_pred))  
    print("RMSE:", np.sqrt(mean_squared_error(y_true, y_pred)))  
    print("R2 Score:", r2_score(y_true, y_pred))  
  
print_metrics(y_test, y_pred_dt, "Решающее дерево")  
print_metrics(y_test, y_pred_gb, "Градиентный бустинг")
```

```
Модель: Решающее дерево  
MAE: 9.444444444444445  
RMSE: 11.080513425729775  
R2 Score: -0.4484415962714827
```

```
Модель: Градиентный бустинг  
MAE: 8.000570033913798  
RMSE: 9.746098152723048  
R2 Score: -0.12057977940626863
```

Вывод Градиентный бустинг справился немного лучше решающего дерева