|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Отчет по домашнему заданию №1

по дисциплине «Методы машинного обучения»

Выполнила:

студентка группы № ИУ5-21М

Саватюгина Ю.Б.

подпись, дата

Проверил:

Гапанюк Ю.Е.

подпись, дата

2021 г.

**Задание.**

1. Поиск и выбор набора данных для построения модели машинного обучения. На основе выбранного датасета строится модель машинного обучения для решения или задачи классификации, или задачи регрессии
2. Для выбранного датасета на основе материалов лекций, в целях улучшения выборки, решить следующие задачи:

* Устранение пропусков в данных
* Кодировка категориальных признаков
* Нормализацию числовых признаков
* Масштабирование признаков
* Обработку выбросов для числовых признаков
* Обработку нестандартных признаков (которые не являются категориальными или числовыми)
* Отбор признаков, наиболее подходящих для построения модели
* Устранение дисбаланса классов в случае решения задачи классификации на дисбалансированной выборке

1. Обучить модель и оценить метрика качества для двух выборок

* Исходная выборка, которая содержит только минимальную предобработку данных, необходимую для построения модели
* Улучшенная выборка, полученная в результате полной предобработки данных в пункте 2

1. Построить модель с использованием произвольной библиотеки AutoML
2. Сравнить метрики для трех полученных моделей

**Текст программы и экранные формы с примерами работы программы**

**from** **google.colab** **import** drive

drive.mount('/content/drive', force\_remount=**True**)

path = "/content/drive/"

Mounted at /content/drive

In [ ]:

!pip3 install tensorflow==2.2

Collecting tensorflow==2.2

Downloading https://files.pythonhosted.org/packages/4c/1a/0d79814736cfecc825ab8094b39648cc9c46af7af1bae839928acb73b4dd/tensorflow-2.2.0-cp37-cp37m-manylinux2010\_x86\_64.whl (516.2MB)

|████████████████████████████████| 516.2MB 32kB/s

Requirement already satisfied: wheel>=0.26; python\_version >= "3" in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (0.36.2)

Collecting gast==0.3.3

Downloading https://files.pythonhosted.org/packages/d6/84/759f5dd23fec8ba71952d97bcc7e2c9d7d63bdc582421f3cd4be845f0c98/gast-0.3.3-py2.py3-none-any.whl

Requirement already satisfied: absl-py>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (0.12.0)

Requirement already satisfied: protobuf>=3.8.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (3.12.4)

Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (1.34.1)

Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (1.1.0)

Requirement already satisfied: google-pasta>=0.1.8 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (0.2.0)

Requirement already satisfied: wrapt>=1.11.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (1.12.1)

Requirement already satisfied: numpy<2.0,>=1.16.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (1.19.5)

Requirement already satisfied: astunparse==1.6.3 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (1.6.3)

Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (3.3.0)

Collecting tensorflow-estimator<2.3.0,>=2.2.0

Downloading https://files.pythonhosted.org/packages/a4/f5/926ae53d6a226ec0fda5208e0e581cffed895ccc89e36ba76a8e60895b78/tensorflow\_estimator-2.2.0-py2.py3-none-any.whl (454kB)

|████████████████████████████████| 460kB 46.5MB/s

Requirement already satisfied: keras-preprocessing>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (1.1.2)

Requirement already satisfied: scipy==1.4.1; python\_version >= "3" in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (1.4.1)

Collecting h5py<2.11.0,>=2.10.0

Downloading https://files.pythonhosted.org/packages/3f/c0/abde58b837e066bca19a3f7332d9d0493521d7dd6b48248451a9e3fe2214/h5py-2.10.0-cp37-cp37m-manylinux1\_x86\_64.whl (2.9MB)

|████████████████████████████████| 2.9MB 34.2MB/s

Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.2) (1.15.0)

Collecting tensorboard<2.3.0,>=2.2.0

Downloading https://files.pythonhosted.org/packages/1d/74/0a6fcb206dcc72a6da9a62dd81784bfdbff5fedb099982861dc2219014fb/tensorboard-2.2.2-py3-none-any.whl (3.0MB)

|████████████████████████████████| 3.0MB 17.6MB/s

Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from protobuf>=3.8.0->tensorflow==2.2) (56.1.0)

Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (1.0.1)

Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (1.8.0)

Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (2.23.0)

Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (0.4.4)

Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (1.30.0)

Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (3.3.4)

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (1.24.3)

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (3.0.4)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (2020.12.5)

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (2.10)

Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (1.3.0)

Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from google-auth<2,>=1.6.3->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (0.2.8)

Requirement already satisfied: rsa<5,>=3.1.4; python\_version >= "3.6" in /usr/local/lib/python3.7/dist-packages (from google-auth<2,>=1.6.3->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (4.7.2)

Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from google-auth<2,>=1.6.3->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (4.2.2)

Requirement already satisfied: importlib-metadata; python\_version < "3.8" in /usr/local/lib/python3.7/dist-packages (from markdown>=2.6.8->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (4.0.1)

Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (3.1.0)

Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1->google-auth<2,>=1.6.3->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (0.4.8)

Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata; python\_version < "3.8"->markdown>=2.6.8->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (3.4.1)

Requirement already satisfied: typing-extensions>=3.6.4; python\_version < "3.8" in /usr/local/lib/python3.7/dist-packages (from importlib-metadata; python\_version < "3.8"->markdown>=2.6.8->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2) (3.7.4.3)

Installing collected packages: gast, tensorflow-estimator, h5py, tensorboard, tensorflow

Found existing installation: gast 0.4.0

Uninstalling gast-0.4.0:

Successfully uninstalled gast-0.4.0

Found existing installation: tensorflow-estimator 2.5.0

Uninstalling tensorflow-estimator-2.5.0:

Successfully uninstalled tensorflow-estimator-2.5.0

Found existing installation: h5py 3.1.0

Uninstalling h5py-3.1.0:

Successfully uninstalled h5py-3.1.0

Found existing installation: tensorboard 2.5.0

Uninstalling tensorboard-2.5.0:

Successfully uninstalled tensorboard-2.5.0

Found existing installation: tensorflow 2.5.0

Uninstalling tensorflow-2.5.0:

Successfully uninstalled tensorflow-2.5.0

Successfully installed gast-0.3.3 h5py-2.10.0 tensorboard-2.2.2 tensorflow-2.2.0 tensorflow-estimator-2.2.0

In [ ]:

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**import** **seaborn** **as** **sns**

**import** **matplotlib.pyplot** **as** **plt**

**from** **sklearn.svm** **import** SVR

**from** **sklearn.svm** **import** LinearSVC

**from** **sklearn.feature\_selection** **import** SelectFromModel

**from** **sklearn.linear\_model** **import** Lasso

**from** **sklearn.linear\_model** **import** LogisticRegression

**from** **sklearn.neighbors** **import** KNeighborsClassifier

**from** **sklearn.neighbors** **import** KNeighborsRegressor

**from** **sklearn.tree** **import** DecisionTreeClassifier

**from** **sklearn.ensemble** **import** RandomForestClassifier

**from** **sklearn.ensemble** **import** GradientBoostingClassifier

**from** **sklearn.tree** **import** DecisionTreeRegressor

**from** **sklearn.ensemble** **import** RandomForestRegressor

**from** **sklearn.ensemble** **import** GradientBoostingRegressor

**from** **sklearn.metrics** **import** mean\_squared\_error

**from** **sklearn.model\_selection** **import** train\_test\_split

**from** **sklearn.feature\_selection** **import** VarianceThreshold

**from** **sklearn.feature\_selection** **import** mutual\_info\_classif, mutual\_info\_regression

**from** **sklearn.feature\_selection** **import** SelectKBest, SelectPercentile

**from** **sklearn.impute** **import** SimpleImputer

**from** **sklearn.impute** **import** MissingIndicator

**from** **sklearn.impute** **import** KNNImputer

**from** **sklearn.preprocessing** **import** StandardScaler

**from** **sklearn.linear\_model** **import** Lasso

**from** **sklearn.pipeline** **import** Pipeline

**from** **sklearn.model\_selection** **import** GridSearchCV

**from** **sklearn.ensemble** **import** RandomForestRegressor

**from** **sklearn.experimental** **import** enable\_iterative\_imputer

**from** **sklearn.impute** **import** IterativeImputer

**from** **sklearn.model\_selection** **import** train\_test\_split

**from** **IPython.display** **import** Image

**import** **scipy.stats** **as** **stats**

**import** **warnings**

warnings.simplefilter("ignore", **UserWarning**)

%matplotlib inline

sns.set(style="ticks")

In [ ]:

data = pd.read\_csv(path+'/MyDrive/student-mat.csv')

data.head()

Out[ ]:

In [ ]:

data.shape

Out[ ]:

(395, 33)

**Пропуски в данных**

In [ ]:

data\_features = list(zip(

*# признаки*

[i **for** i **in** data.columns],

zip(

*# типы колонок*

[str(i) **for** i **in** data.dtypes],

*# проверим есть ли пропущенные значения*

[i **for** i **in** data.isnull().sum()],

[i **for** i **in** data.nunique()]

)))

*# Признаки с типом данных и количеством пропусков*

data\_features

Out[ ]:

[('school', ('object', 0, 2)),

('sex', ('object', 0, 2)),

('age', ('int64', 0, 8)),

('address', ('object', 0, 2)),

('famsize', ('object', 0, 2)),

('Pstatus', ('object', 0, 2)),

('Medu', ('int64', 0, 5)),

('Fedu', ('int64', 0, 5)),

('Mjob', ('object', 0, 5)),

('Fjob', ('object', 0, 5)),

('reason', ('object', 0, 4)),

('guardian', ('object', 0, 3)),

('traveltime', ('int64', 0, 4)),

('studytime', ('int64', 0, 4)),

('failures', ('int64', 0, 4)),

('schoolsup', ('object', 0, 2)),

('famsup', ('object', 0, 2)),

('paid', ('object', 0, 2)),

('activities', ('object', 0, 2)),

('nursery', ('object', 0, 2)),

('higher', ('object', 0, 2)),

('internet', ('object', 0, 2)),

('romantic', ('object', 0, 2)),

('famrel', ('int64', 0, 5)),

('freetime', ('int64', 0, 5)),

('goout', ('int64', 0, 5)),

('Dalc', ('int64', 0, 5)),

('Walc', ('int64', 0, 5)),

('health', ('int64', 0, 5)),

('absences', ('int64', 0, 34)),

('G1', ('int64', 0, 17)),

('G2', ('int64', 0, 17)),

('G3', ('int64', 0, 18))]

Пропусков в данных не обнаружено.

**Кодирование категориальных признаков**

In [ ]:

**from** **sklearn.preprocessing** **import** LabelEncoder

In [ ]:

data\_encoded = data.copy()

In [ ]:

columns = ['school','sex','address','famsize','Pstatus','Mjob','Fjob','reason','guardian','schoolsup','famsup','paid','activities','nursery','higher','internet','romantic']

**for** column **in** columns:

label\_encoder = LabelEncoder()

data\_encoded[column] = label\_encoder.fit\_transform(data\_encoded[column])

In [ ]:

data\_encoded.head()

Out[ ]:

**Нормализация числовых признаков**

In [ ]:

**def** diagnostic\_plots(df, variable):

plt.figure(figsize=(10,5))

*# гистограмма*

plt.subplot(1, 2, 1)

df[variable].hist(bins=30)

*## Q-Q plot*

plt.subplot(1, 2, 2)

stats.probplot(df[variable], dist="norm", plot=plt)

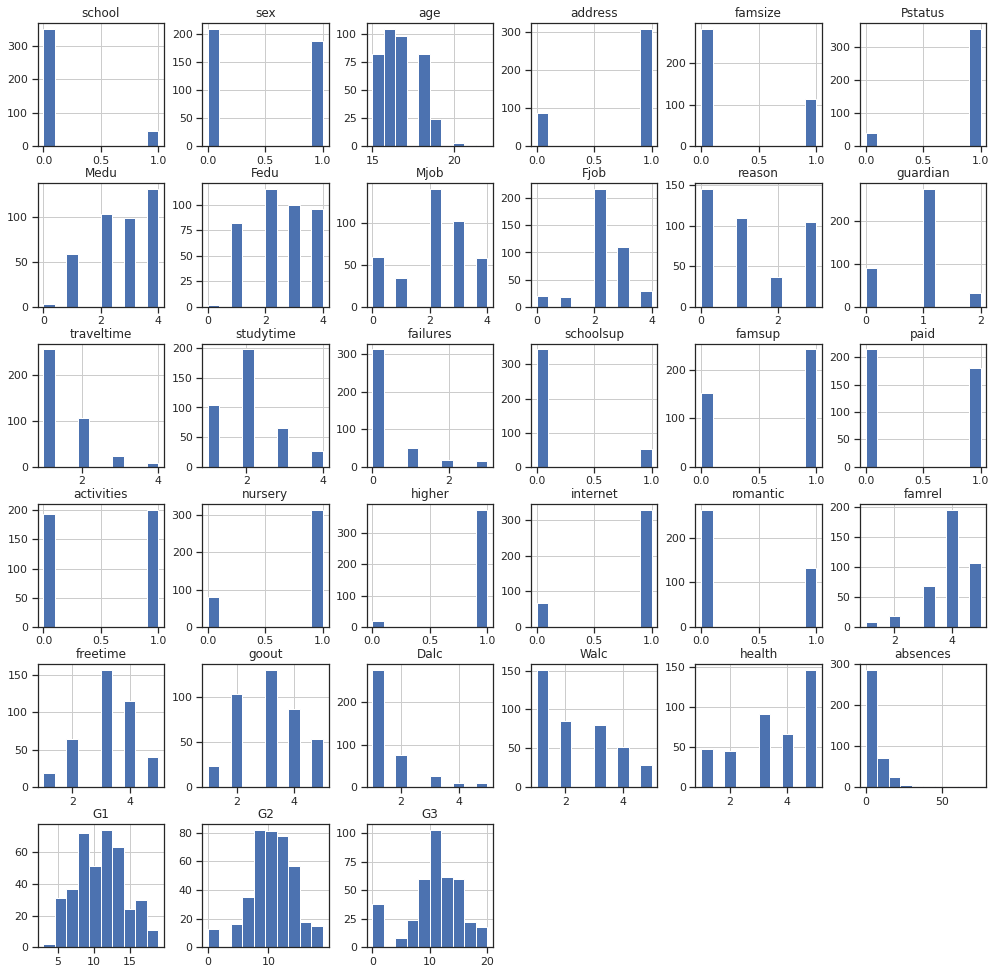
plt.show()

In [ ]:

data2 = data\_encoded.copy()

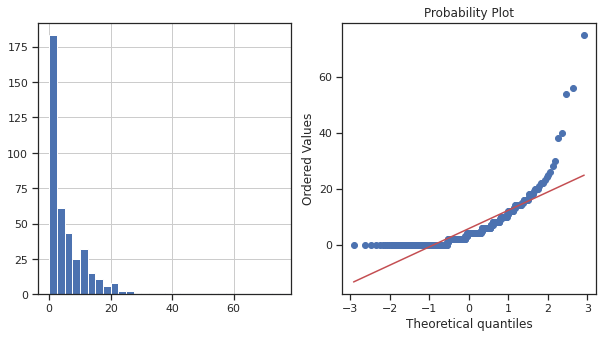
data2.hist(figsize=(17,17))

plt.show()

****

In [ ]:

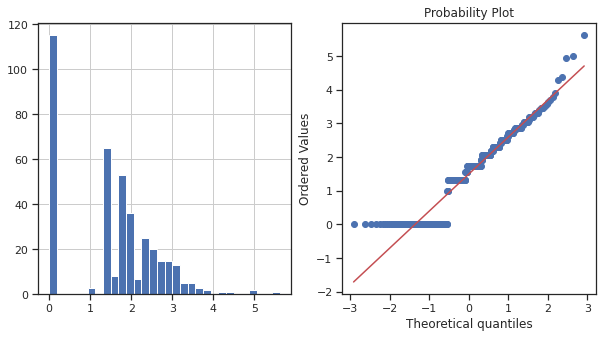
diagnostic\_plots(data2, 'absences')

****

In [ ]:

data2['absences\_exp1'] = data2['absences']\*\*(0.4)

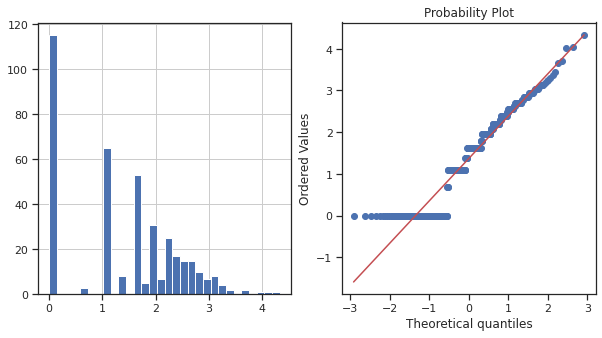
diagnostic\_plots(data2, 'absences\_exp1')

****

In [ ]:

data2['absences\_log'] = np.log(data2['absences']+1)

diagnostic\_plots(data2, 'absences\_log')

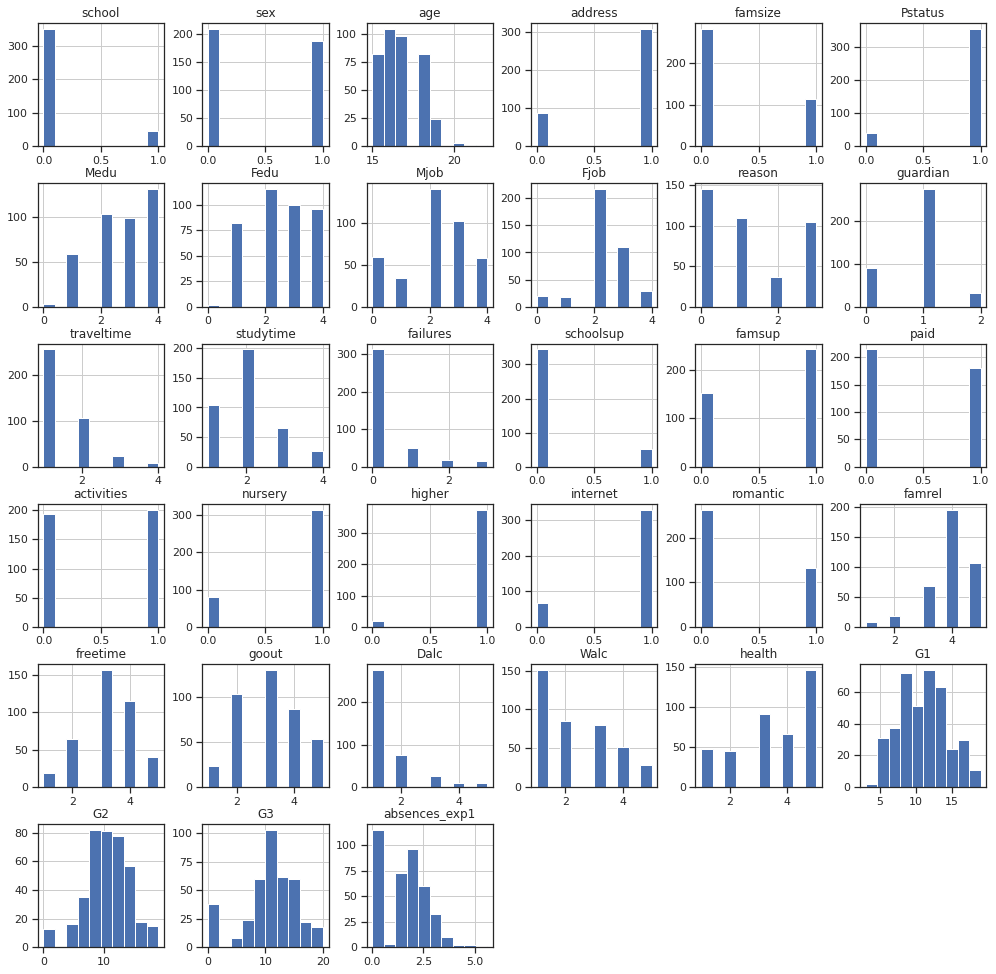
****

In [ ]:

data2 = data2.drop(columns=['absences'])

data2.hist(figsize=(17,17))

plt.show()

****

Пропустим пока нормализацию

**Масштабирование признаков**

In [ ]:

filtered\_columns = ['age','Medu','Fedu','traveltime','studytime','failures','famrel','freetime','goout','Dalc','Walc','health','absences\_exp1','G1','G2','G3']

data3 = data2[filtered\_columns]

*# Нужно ли масштабирование*

data3.describe()

Out[ ]:

In [ ]:

*# DataFrame не содержащий целевой признак*

X\_ALL = data3.drop('G3', axis=1)

In [ ]:

*# Функция для восстановления датафрейма*

*# на основе масштабированных данных*

**def** arr\_to\_df(arr\_scaled):

res = pd.DataFrame(arr\_scaled, columns=X\_ALL.columns)

**return** res

In [ ]:

*# Разделим выборку на обучающую и тестовую*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_ALL, data3['G3'],

test\_size=0.2,

random\_state=1)

*# Преобразуем массивы в DataFrame*

X\_train\_df = arr\_to\_df(X\_train)

X\_test\_df = arr\_to\_df(X\_test)

X\_train\_df.shape, X\_test\_df.shape

Out[ ]:

((316, 15), (79, 15))

In [ ]:

*# Обучаем StandardScaler на всей выборке и масштабируем*

cs11 = StandardScaler()

data\_cs11\_scaled\_temp = cs11.fit\_transform(X\_ALL)

*# формируем DataFrame на основе массива*

data\_cs11\_scaled = arr\_to\_df(data\_cs11\_scaled\_temp)

data\_cs11\_scaled

Out[ ]:

395 rows × 15 columns

In [ ]:

data\_cs11\_scaled.describe()

Out[ ]:

|  | **age** | **Medu** | **Fedu** | **traveltime** | **studytime** | **failures** | **famrel** | **freetime** | **goout** | **Dalc** | **Walc** | **health** | **absences\_exp1** | **G1** | **G2** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 | 3.950000e+02 |
| **mean** | 1.411529e-15 | 8.432074e-18 | -1.264811e-16 | -2.203582e-16 | -2.709506e-16 | -2.599889e-16 | -1.410967e-16 | 1.028713e-16 | -2.062345e-17 | 8.769357e-17 | -3.091760e-17 | 1.169248e-16 | -1.068063e-17 | -5.424634e-17 | -1.262000e-16 |
| **std** | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 | 1.001268e+00 |
| **min** | -1.330954e+00 | -2.514630e+00 | -2.320084e+00 | -6.432495e-01 | -1.235351e+00 | -4.499436e-01 | -3.287804e+00 | -2.240828e+00 | -1.896683e+00 | -5.406987e-01 | -1.003789e+00 | -1.839649e+00 | -1.307862e+00 | -2.385787e+00 | -2.851920e+00 |
| **25%** | -5.462869e-01 | -6.853872e-01 | -4.798568e-01 | -6.432495e-01 | -1.235351e+00 | -4.499436e-01 | 6.219406e-02 | -2.360102e-01 | -9.972953e-01 | -5.406987e-01 | -1.003789e+00 | -3.992895e-01 | -1.307862e+00 | -8.774869e-01 | -4.562264e-01 |

In [ ]:

*# Построение плотности распределения*

**def** draw\_kde(col\_list, df1, df2, label1, label2):

fig, (ax1, ax2) = plt.subplots(

ncols=2, figsize=(12, 5))

*# первый график*

ax1.set\_title(label1)

sns.kdeplot(data=df1[col\_list], ax=ax1)

*# второй график*

ax2.set\_title(label2)

sns.kdeplot(data=df2[col\_list], ax=ax2)

plt.show()

In [ ]:

draw\_kde(['absences\_exp1', 'health', 'traveltime'], data3, data\_cs11\_scaled, 'до масштабирования', 'после масштабирования')

**Работа с выбросами**

In [ ]:

data4 = data\_cs11\_scaled

In [ ]:

x\_col\_list = ['age','Medu','Fedu','traveltime','studytime','failures','famrel','freetime','goout','Dalc','Walc','health','absences\_exp1','G1','G2']

In [ ]:

**def** diagnostic\_plots(df, variable, title):

fig, ax = plt.subplots(figsize=(15,10))

*# гистограмма*

plt.subplot(2, 2, 1)

df[variable].hist(bins=30)

*## Q-Q plot*

plt.subplot(2, 2, 2)

stats.probplot(df[variable], dist="norm", plot=plt)

*# ящик с усами*

plt.subplot(2, 2, 3)

sns.violinplot(x=df[variable])

*# ящик с усами*

plt.subplot(2, 2, 4)

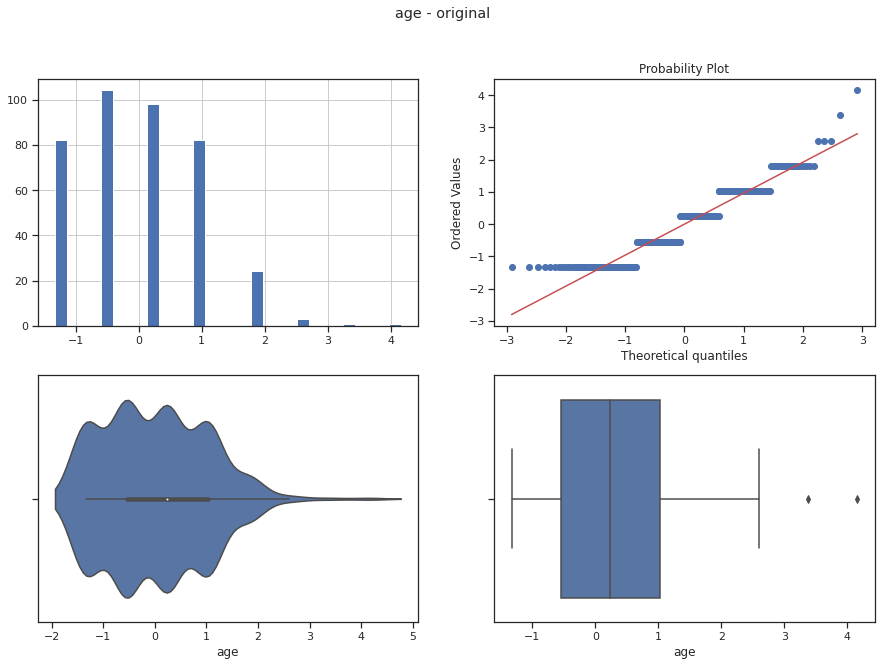
sns.boxplot(x=df[variable])

fig.suptitle(title)

plt.show()

In [ ]:

diagnostic\_plots(data4, 'age', 'age - original')

****

In [ ]:

*# Тип вычисления верхней и нижней границы выбросов*

**from** **enum** **import** Enum

**class** **OutlierBoundaryType**(Enum):

SIGMA = 1

QUANTILE = 2

IRQ = 3

In [ ]:

*# Функция вычисления верхней и нижней границы выбросов*

**def** get\_outlier\_boundaries(df, col, outlier\_boundary\_type: OutlierBoundaryType):

**if** outlier\_boundary\_type == OutlierBoundaryType.SIGMA:

K1 = 3

lower\_boundary = df[col].mean() - (K1 \* df[col].std())

upper\_boundary = df[col].mean() + (K1 \* df[col].std())

**elif** outlier\_boundary\_type == OutlierBoundaryType.QUANTILE:

lower\_boundary = df[col].quantile(0.05)

upper\_boundary = df[col].quantile(0.95)

**elif** outlier\_boundary\_type == OutlierBoundaryType.IRQ:

K2 = 1.5

IQR = df[col].quantile(0.75) - df[col].quantile(0.25)

lower\_boundary = df[col].quantile(0.25) - (K2 \* IQR)

upper\_boundary = df[col].quantile(0.75) + (K2 \* IQR)

**else**:

**raise** **NameError**('Unknown Outlier Boundary Type')

**return** lower\_boundary, upper\_boundary

In [ ]:

data4.shape

Out[ ]:

(395, 15)

In [ ]:

**for** col **in** x\_col\_list:

**for** obt **in** OutlierBoundaryType:

*# Вычисление верхней и нижней границы*

lower\_boundary, upper\_boundary = get\_outlier\_boundaries(data4, col, obt)

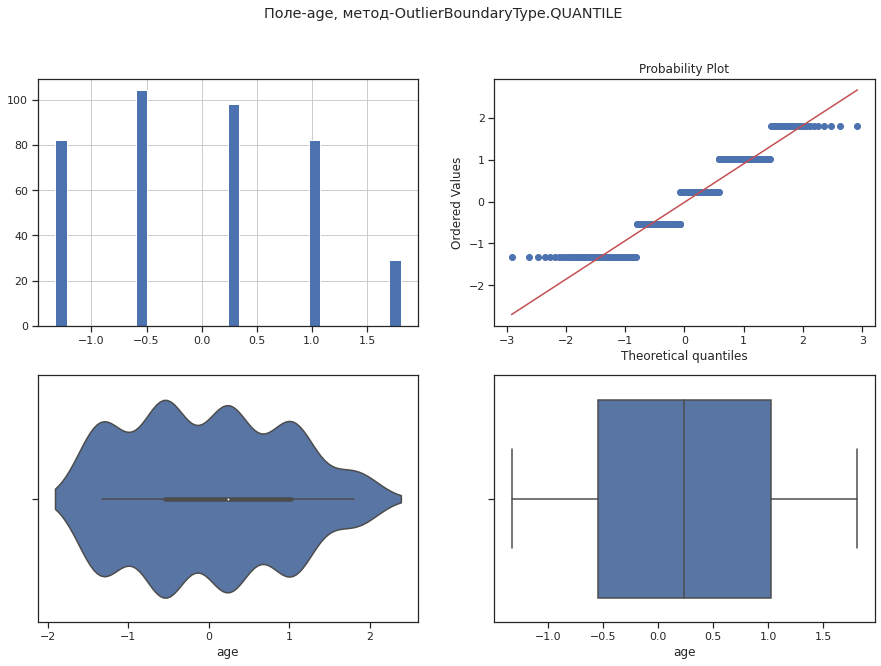
*# Изменение данных*

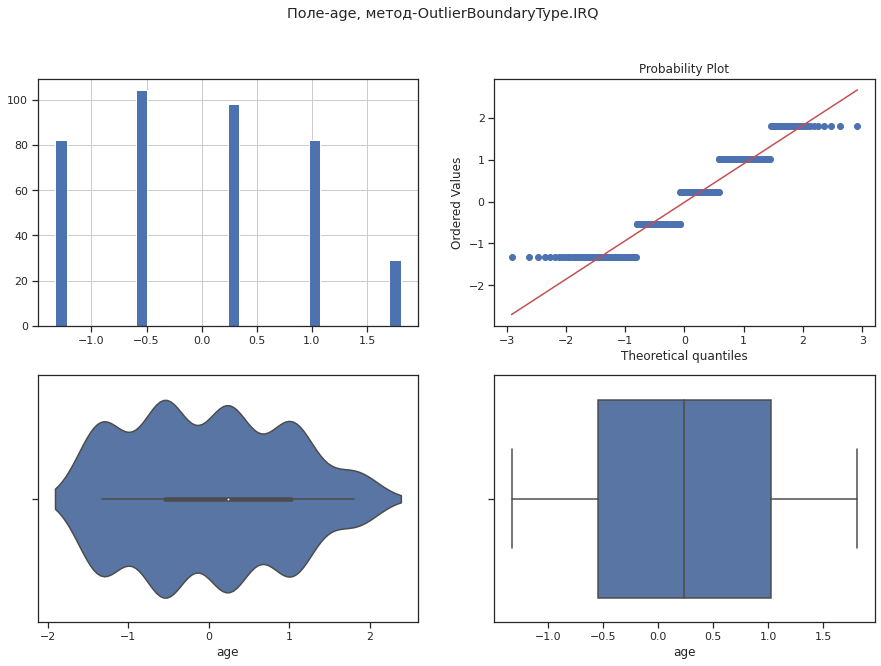
data4[col] = np.where(data4[col] > upper\_boundary, upper\_boundary,

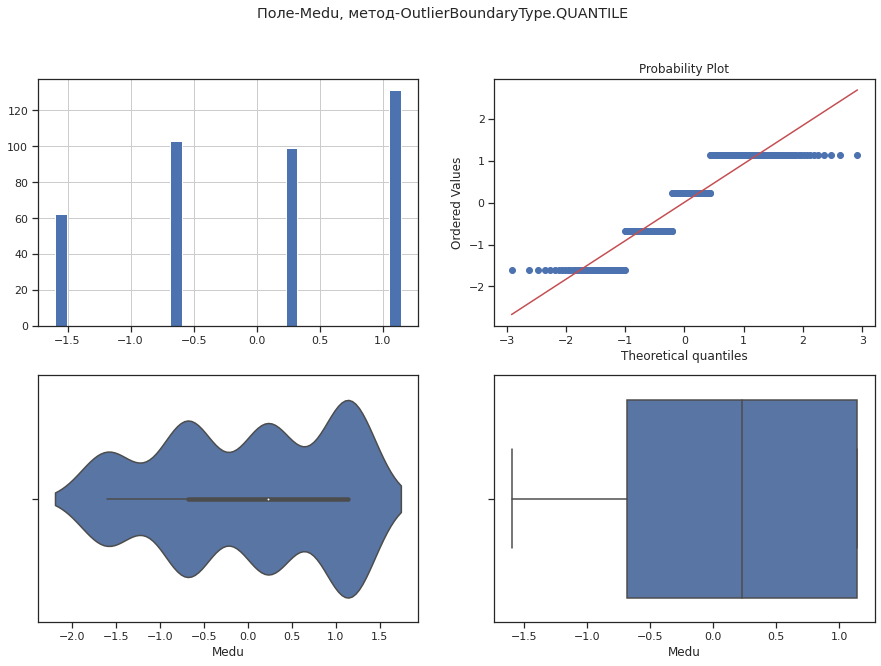
np.where(data4[col] < lower\_boundary, lower\_boundary, data4[col]))

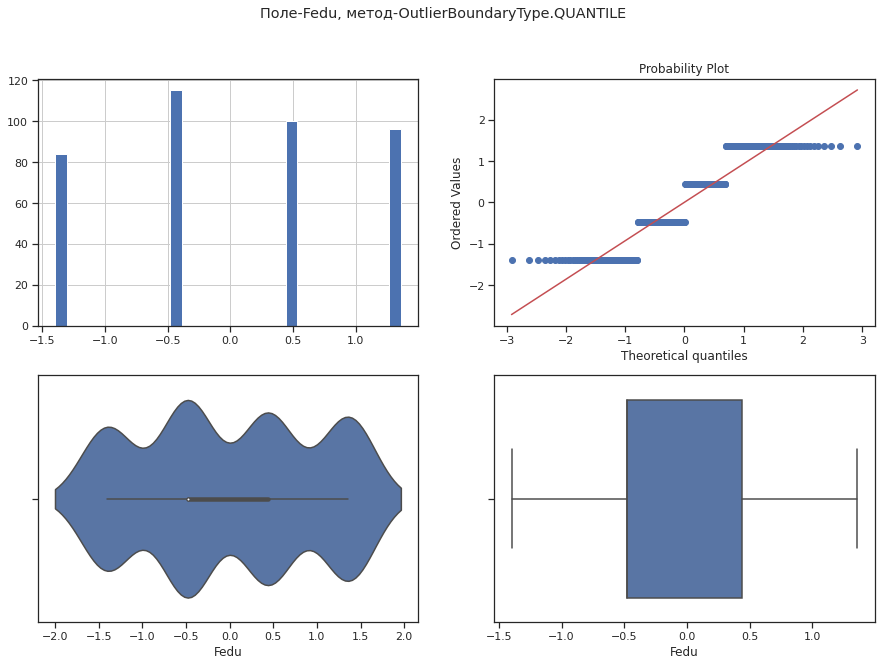
title = 'Поле-**{}**, метод-**{}**'.format(col, obt)

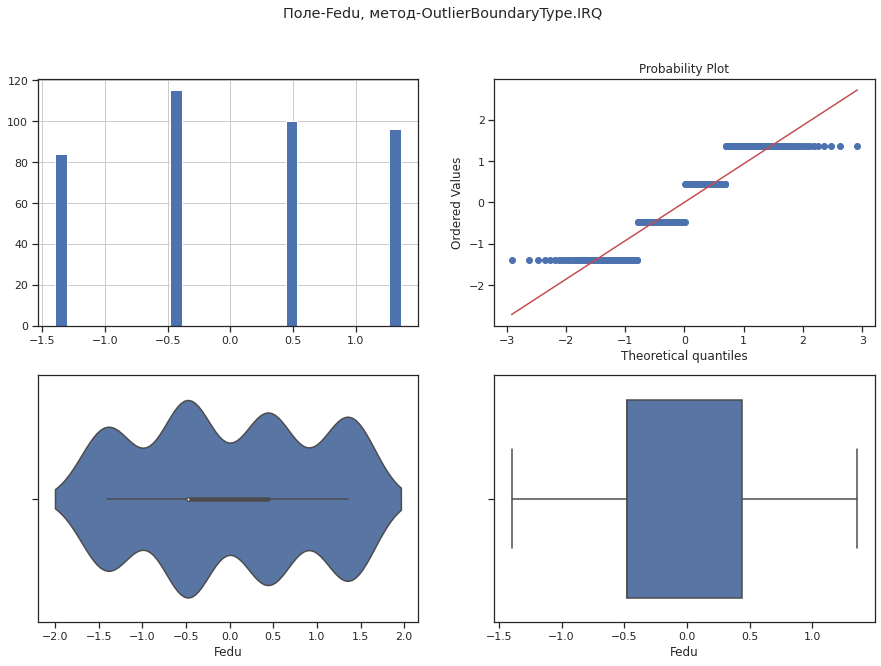
diagnostic\_plots(data4, col, title)

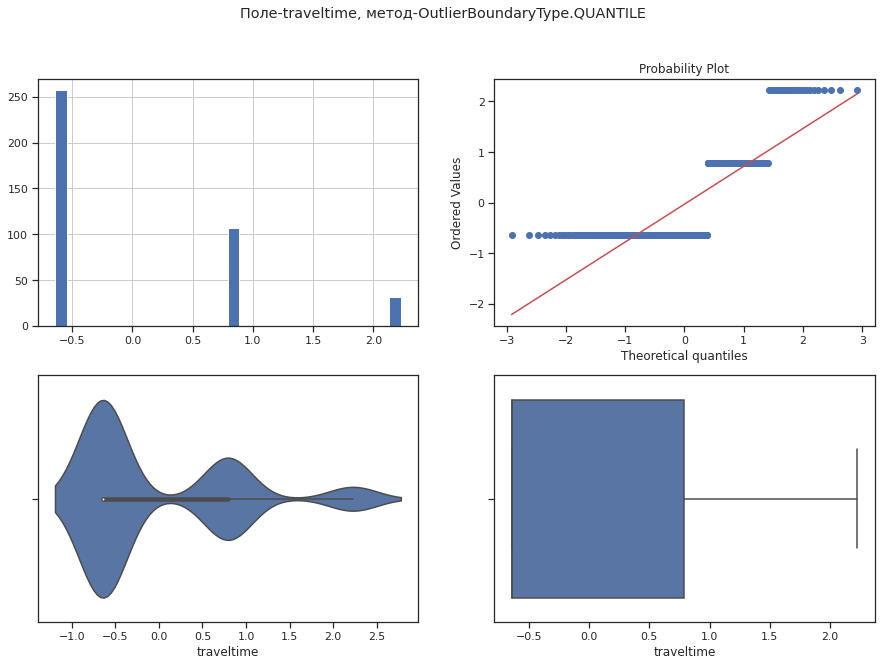
****

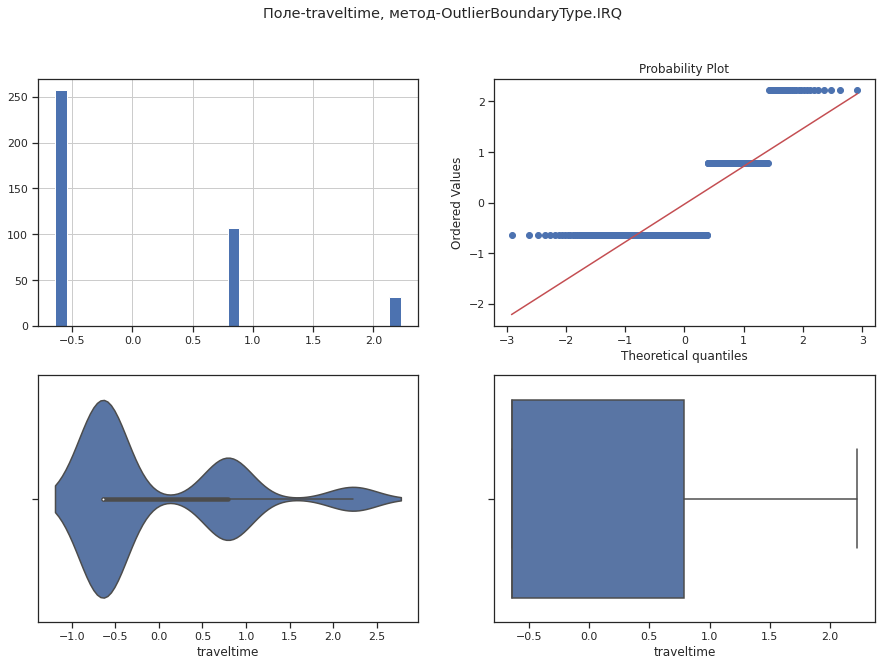
****

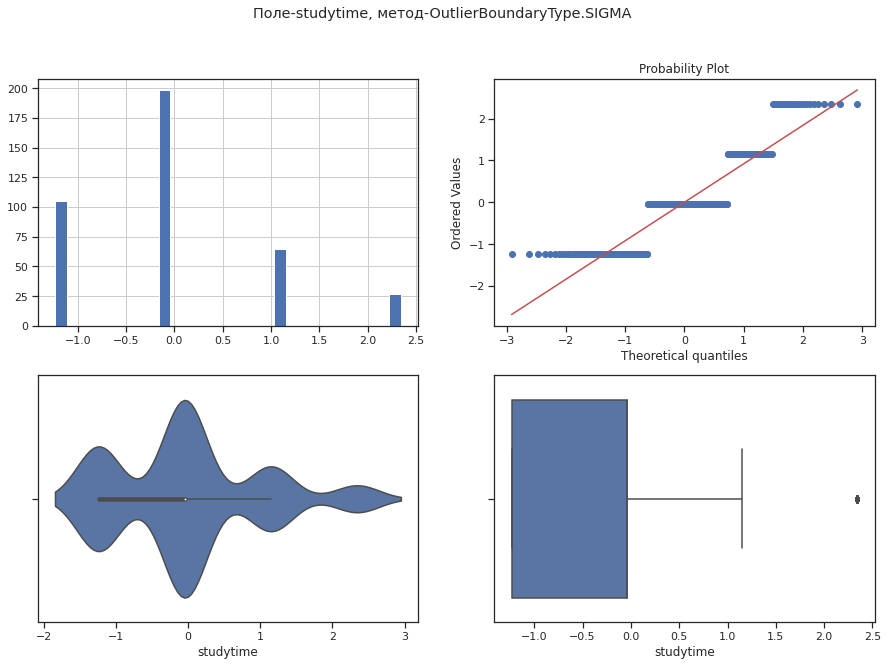
****

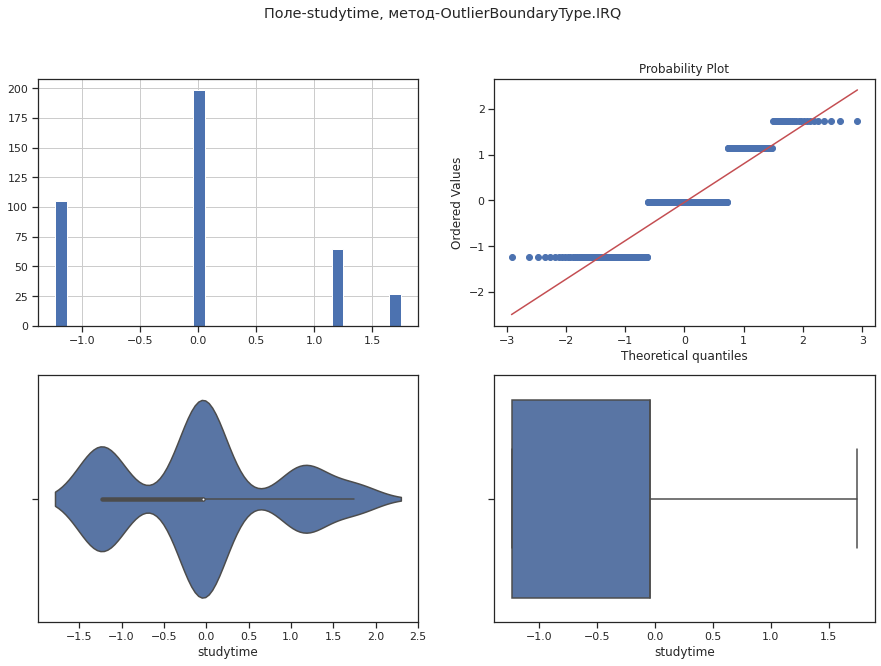
****

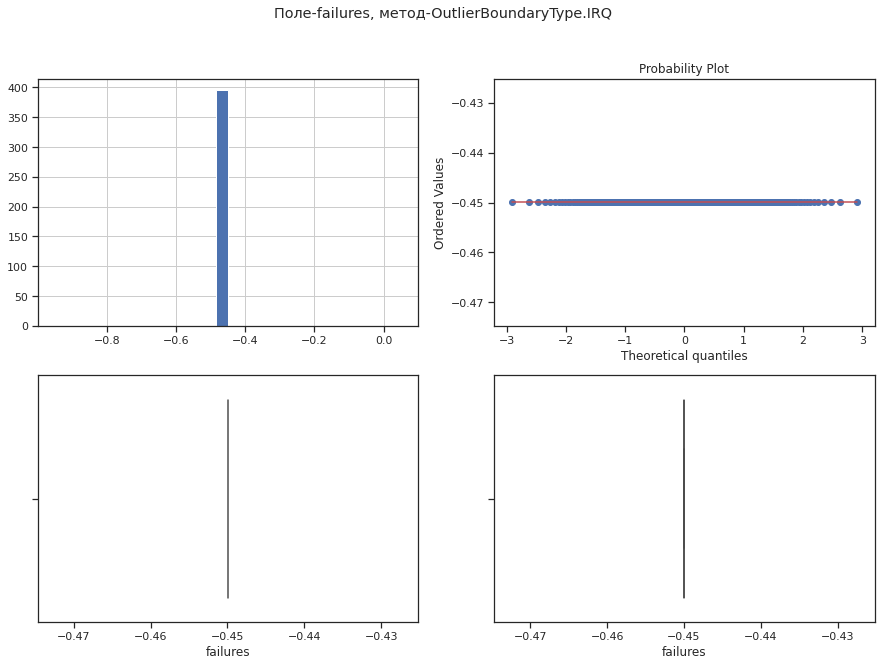
****

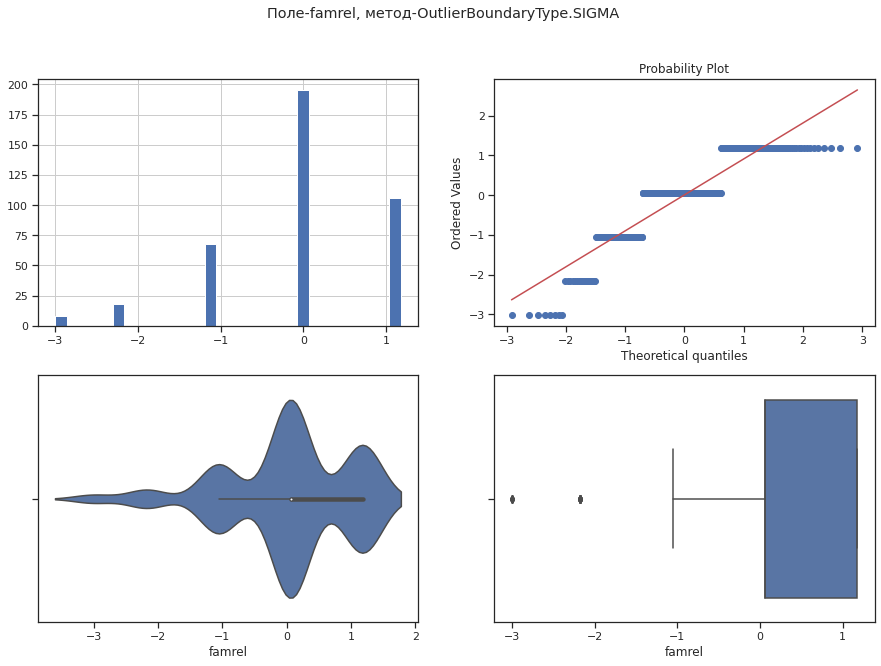
****

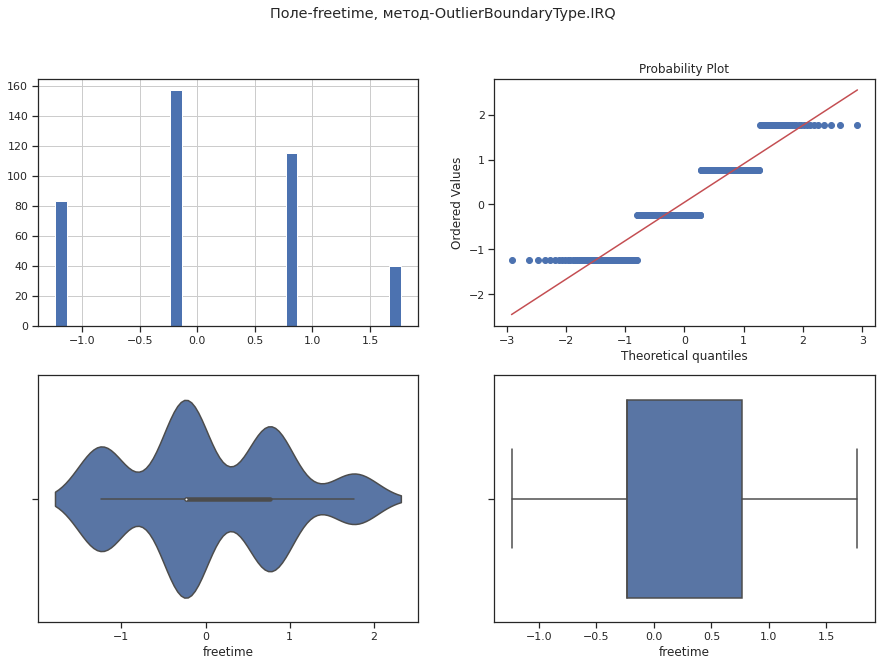
****

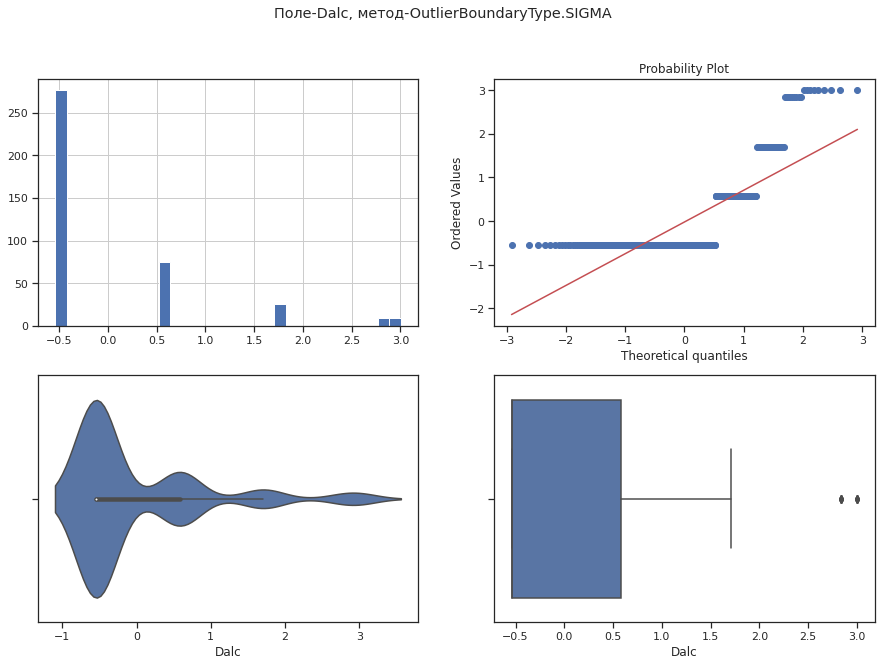
****

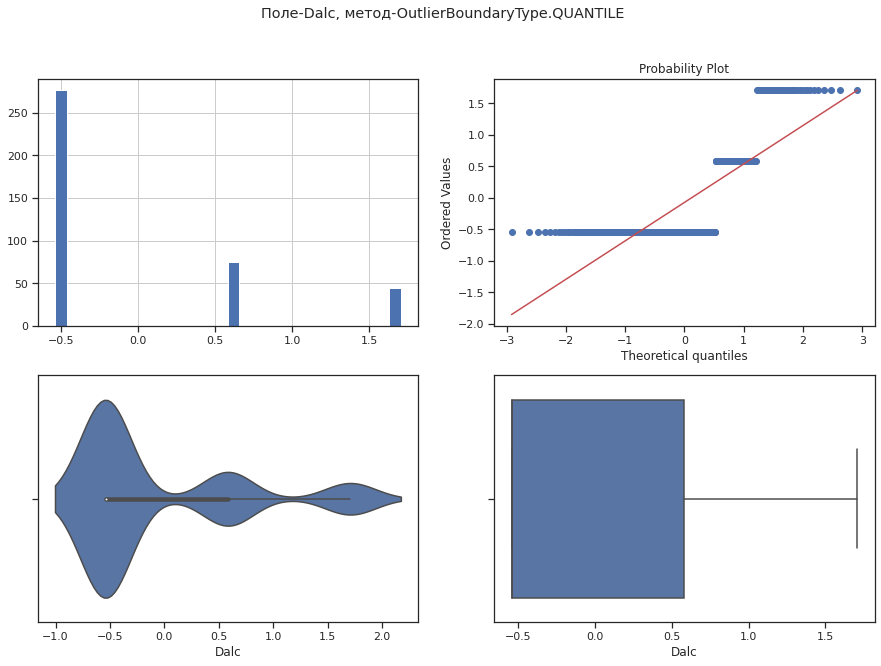
****

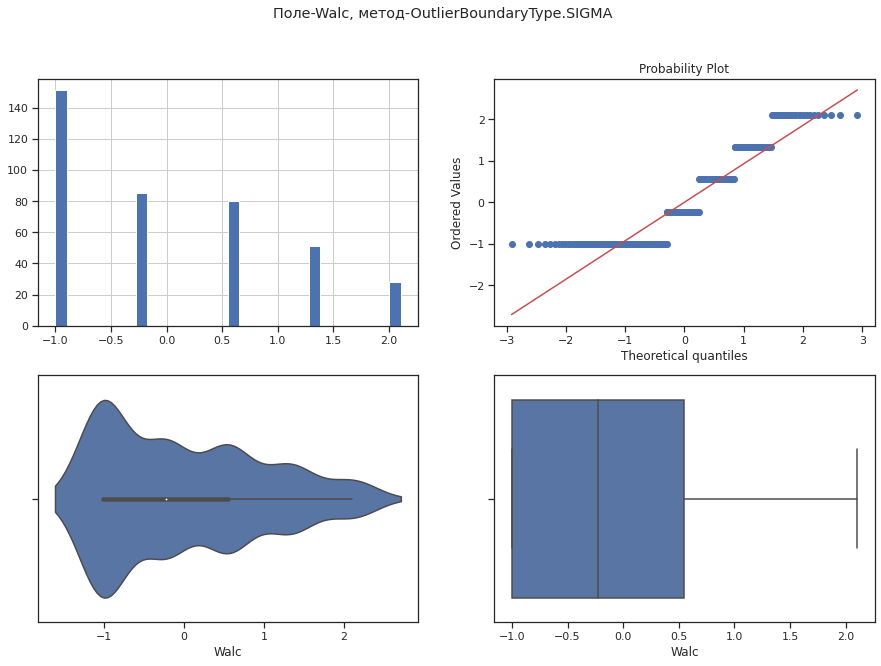
****

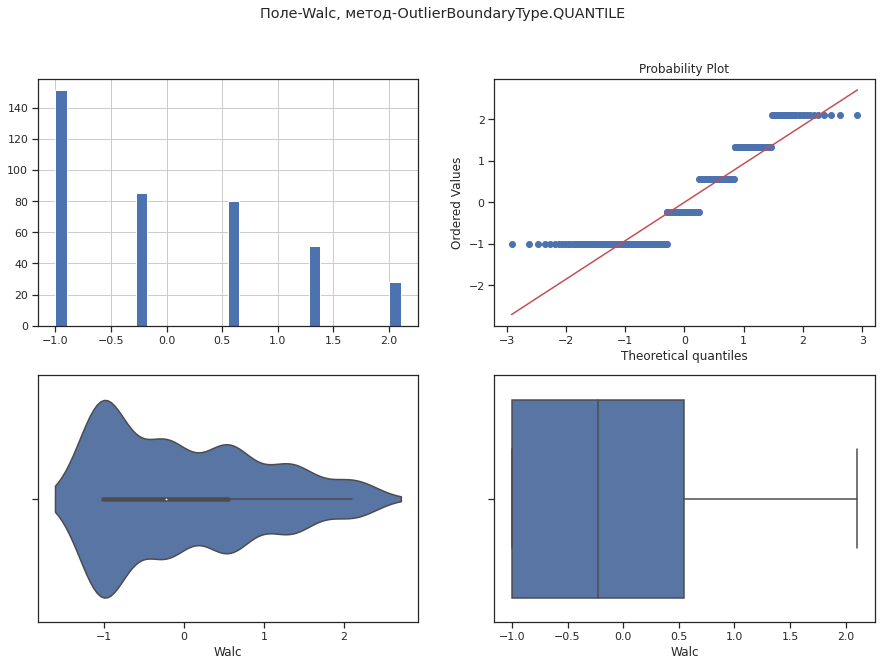
****

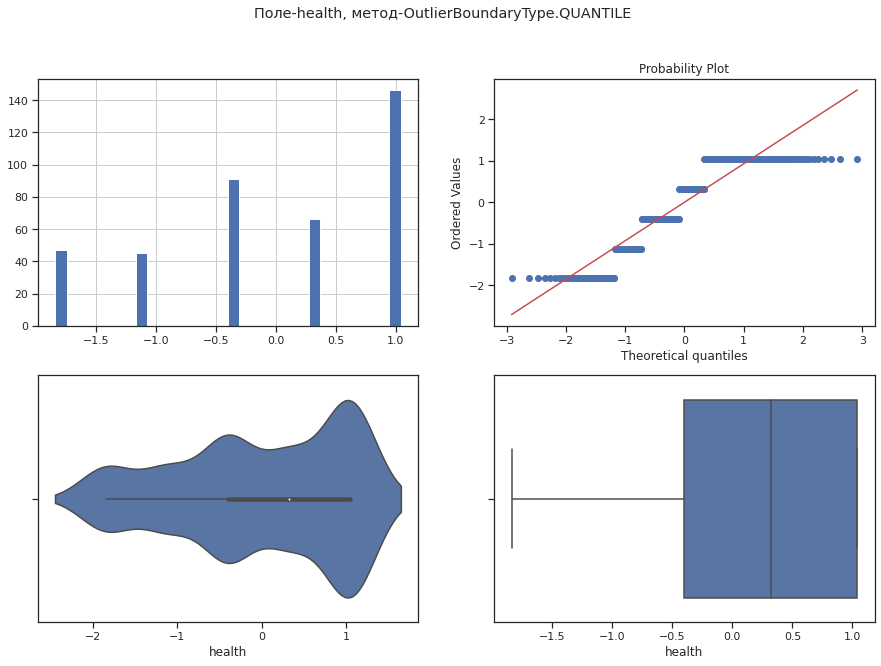
****

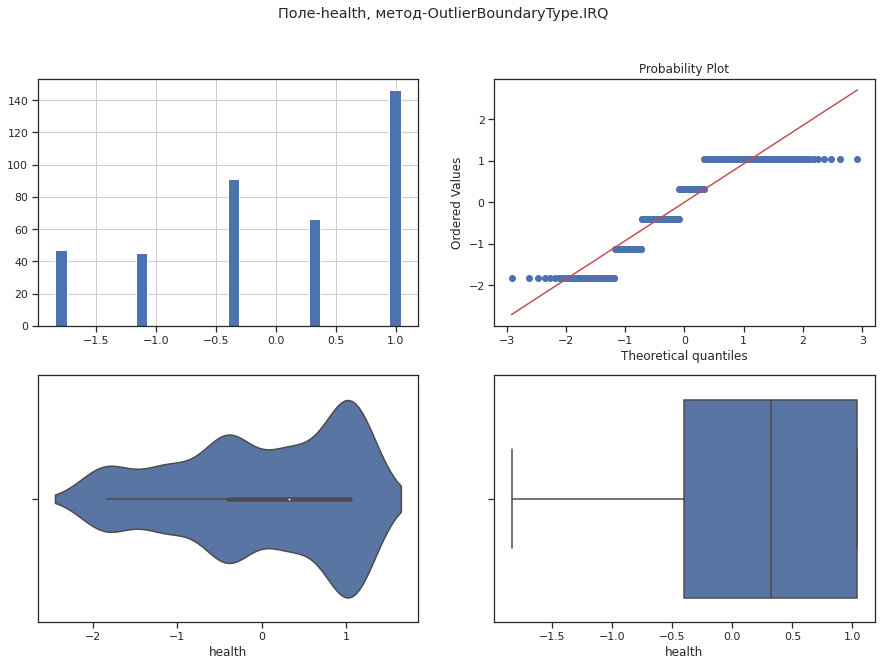
****

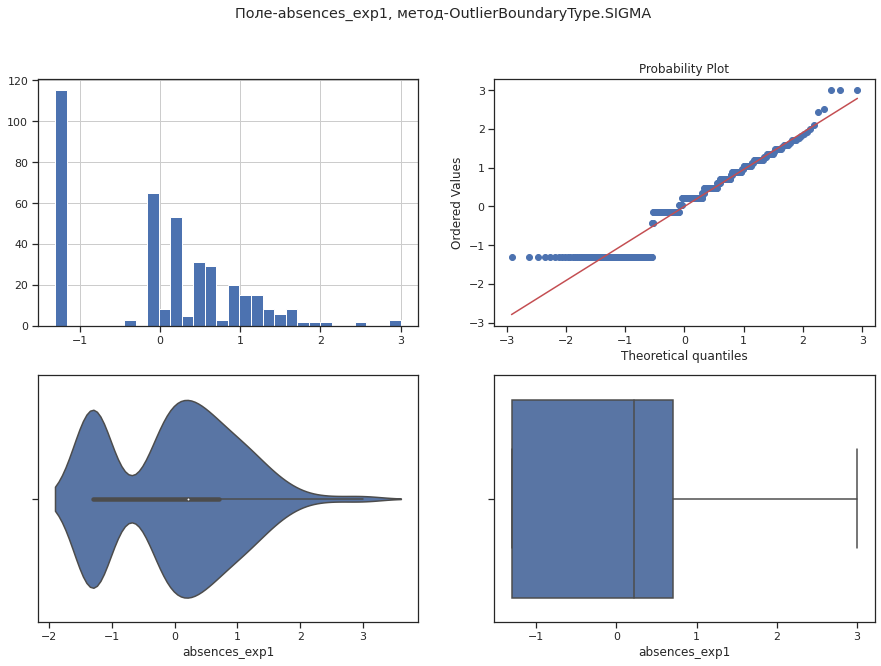
****

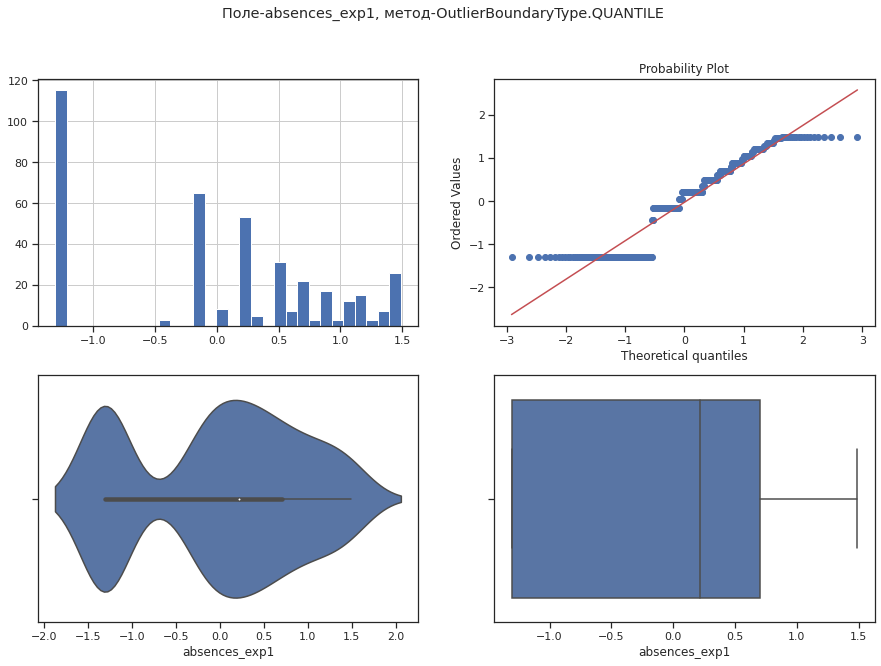
****

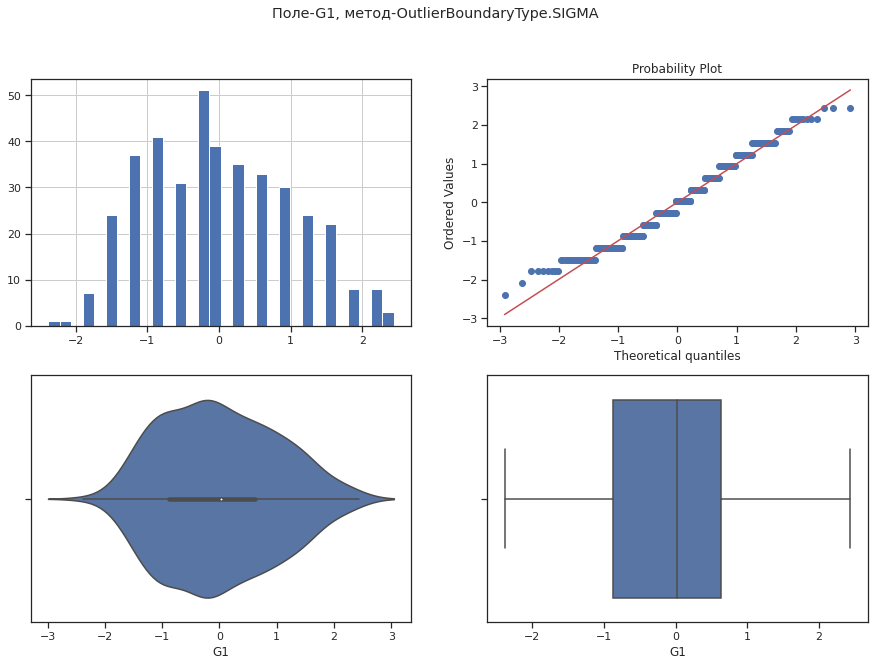
****

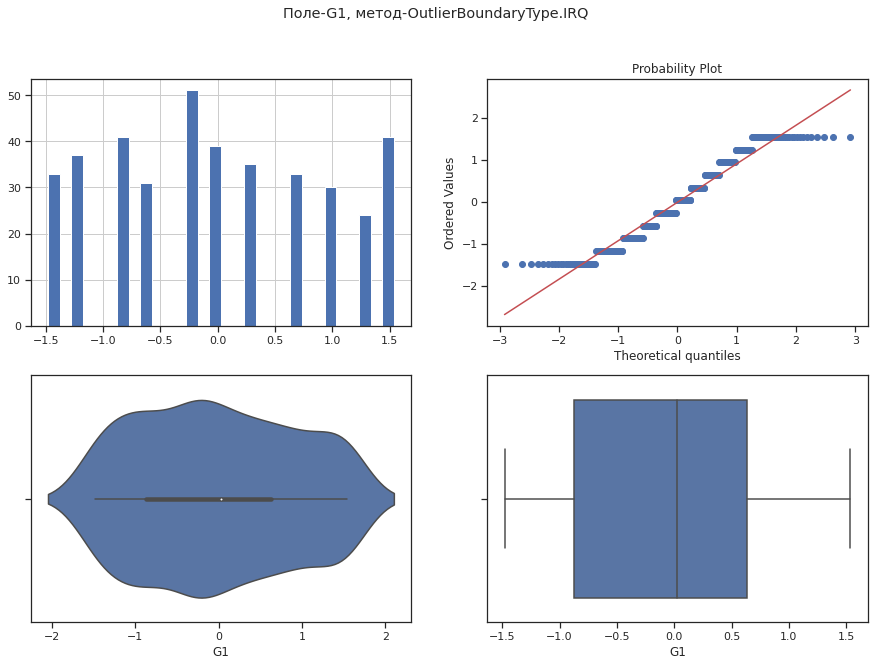
****

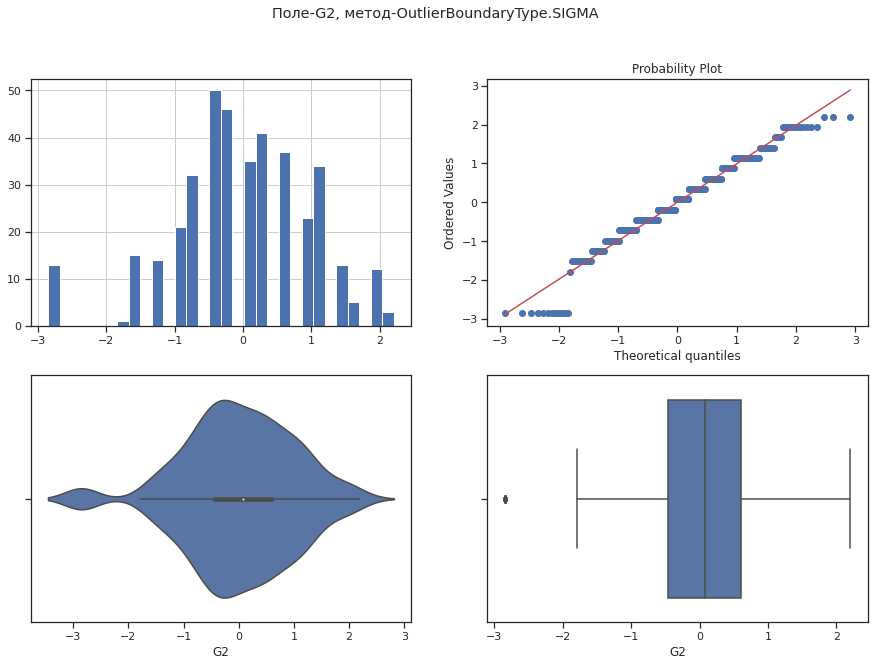
****

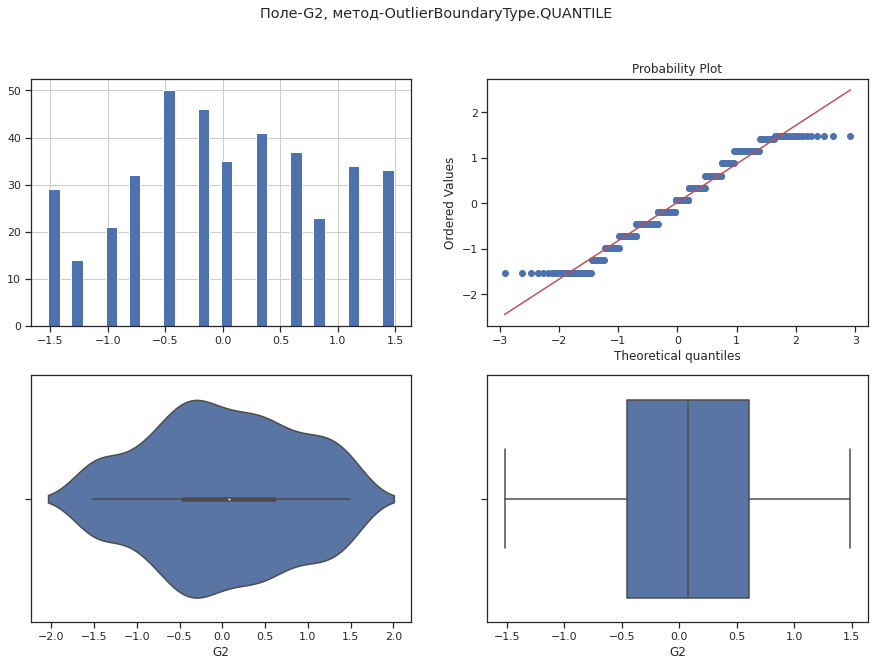
****

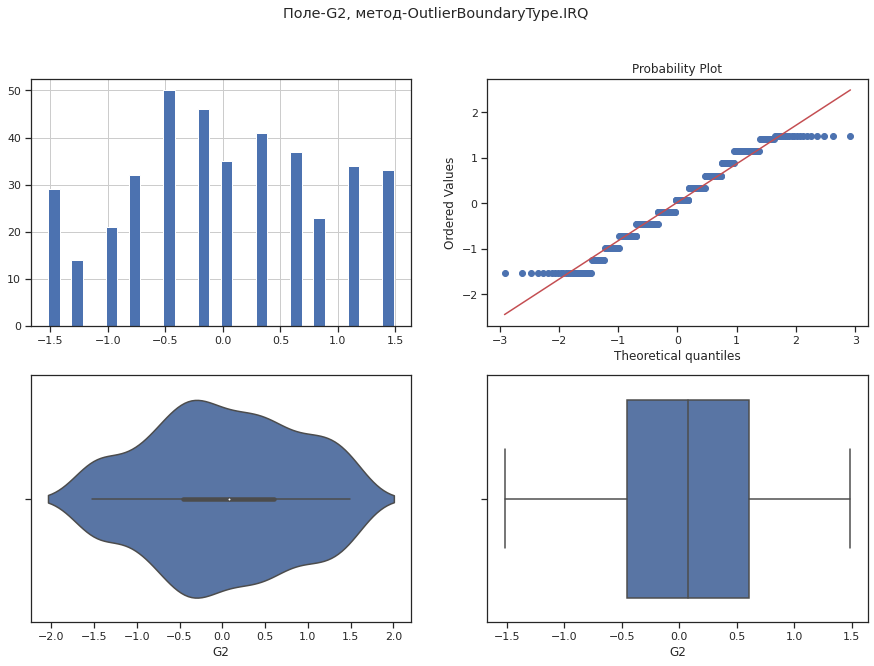
****

****

****

****

****

****

In [ ]:

diagnostic\_plots(data4, 'age', 'age - new')

**Соединяем**

In [ ]:

data5 = data4.copy()

data5.head()

Out[ ]:

In [ ]:

data2.head()

Out[ ]:

In [ ]:

replace\_columns = ['age','Medu','Fedu','traveltime','studytime','failures','famrel','freetime','goout','Dalc','Walc','health','absences\_exp1','G1','G2']

**for** i **in** replace\_columns:

data2[i]=data5[i]

In [ ]:

data2.head()

Out[ ]:

**Обработка нестандартных признаков: не требуется**

**Отбор признаков**

In [ ]:

data6 = data2.copy()

plt.figure(figsize=(30,30))

sns.heatmap(data6.corr(), cmap="Oranges", annot=**True**, linewidths=3)

Out[ ]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fbd413251d0>

In [ ]:

*# Формирование DataFrame с сильными корреляциями*

**def** make\_corr\_df(df):

cr = data6.corr()

cr = cr.abs().unstack()

cr = cr.sort\_values(ascending=**False**)

cr = cr[cr >= 0.8]

cr = cr[cr < 1]

cr = pd.DataFrame(cr).reset\_index()

cr.columns = ['f1', 'f2', 'corr']

**return** cr

In [ ]:

make\_corr\_df(data6)

Out[ ]:

|  | **f1** | **f2** | **corr** |
| --- | --- | --- | --- |
| **0** | G2 | G3 | 0.895755 |
| **1** | G3 | G2 | 0.895755 |
| **2** | G1 | G2 | 0.892269 |
| **3** | G2 | G1 | 0.892269 |

In [ ]:

*# Обнаружение групп коррелирующих признаков*

**def** corr\_groups(cr):

grouped\_feature\_list = []

correlated\_groups = []

**for** feature **in** cr['f1'].unique():

**if** feature **not** **in** grouped\_feature\_list:

*# находим коррелирующие признаки*

correlated\_block = cr[cr['f1'] == feature]

cur\_dups = list(correlated\_block['f2'].unique()) + [feature]

grouped\_feature\_list = grouped\_feature\_list + cur\_dups

correlated\_groups.append(cur\_dups)

**return** correlated\_groups

In [ ]:

*# Группы коррелирующих признаков*

corr\_groups(make\_corr\_df(data6))

Out[ ]:

[['G3', 'G1', 'G2']]

**Другой выбор признаков**

In [ ]:

data\_new = data2.copy()

data\_new.head()

Out[ ]:

In [ ]:

*# DataFrame не содержащий целевой признак*

X\_ALL = data\_new.drop('G3', axis=1)

In [ ]:

*# Функция для восстановления датафрейма*

*# на основе масштабированных данных*

**def** arr\_to\_df(arr\_scaled):

res = pd.DataFrame(arr\_scaled, columns=X\_ALL.columns)

**return** res

In [ ]:

*# Разделим выборку на обучающую и тестовую*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_ALL, data\_new['G3'],

test\_size=0.2,

random\_state=1)

*# Преобразуем массивы в DataFrame*

X\_train\_df = arr\_to\_df(X\_train)

X\_test\_df = arr\_to\_df(X\_test)

X\_train\_df.shape, X\_test\_df.shape

Out[ ]:

((316, 32), (79, 32))

In [ ]:

**from** **sklearn.neighbors** **import** KNeighborsClassifier

**from** **mlxtend.feature\_selection** **import** ExhaustiveFeatureSelector **as** EFS

knn = KNeighborsClassifier(n\_neighbors=3)

In [ ]:

efs1 = EFS(knn,

min\_features=2,

max\_features=4,

scoring='accuracy',

print\_progress=**True**,

cv=5)

efs1 = efs1.fit(X\_train, y\_train, custom\_feature\_names=X\_ALL.columns)

print('Best accuracy score: **%.2f**' % efs1.best\_score\_)

print('Best subset (indices):', efs1.best\_idx\_)

print('Best subset (corresponding names):', efs1.best\_feature\_names\_)

---------------------------------------------------------------------------

KeyboardInterrupt Traceback (most recent call last)

<ipython-input-191-1df39f4e8b39> in <module>()

6 cv=5)

7

----> 8 efs1 = efs1.fit(X\_train, y\_train, custom\_feature\_names=X\_ALL.columns)

9

10 print('Best accuracy score: %.2f' % efs1.best\_score\_)

/usr/local/lib/python3.7/dist-packages/mlxtend/feature\_selection/exhaustive\_feature\_selector.py in fit(self, X, y, custom\_feature\_names, \*\*fit\_params)

270 work = enumerate(parallel(delayed(\_calc\_score)

271 (self, X\_, y, c, \*\*fit\_params)

--> 272 for c in candidates))

273

274 try:

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in \_\_call\_\_(self, iterable)

1042 self.\_iterating = self.\_original\_iterator is not None

1043

-> 1044 while self.dispatch\_one\_batch(iterator):

1045 pass

1046

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in dispatch\_one\_batch(self, iterator)

857 return False

858 else:

--> 859 self.\_dispatch(tasks)

860 return True

861

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in \_dispatch(self, batch)

775 with self.\_lock:

776 job\_idx = len(self.\_jobs)

--> 777 job = self.\_backend.apply\_async(batch, callback=cb)

778 # A job can complete so quickly than its callback is

779 # called before we get here, causing self.\_jobs to

/usr/local/lib/python3.7/dist-packages/joblib/\_parallel\_backends.py in apply\_async(self, func, callback)

206 def apply\_async(self, func, callback=None):

207 """Schedule a func to be run"""

--> 208 result = ImmediateResult(func)

209 if callback:

210 callback(result)

/usr/local/lib/python3.7/dist-packages/joblib/\_parallel\_backends.py in \_\_init\_\_(self, batch)

570 # Don't delay the application, to avoid keeping the input

571 # arguments in memory

--> 572 self.results = batch()

573

574 def get(self):

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in \_\_call\_\_(self)

261 with parallel\_backend(self.\_backend, n\_jobs=self.\_n\_jobs):

262 return [func(\*args, \*\*kwargs)

--> 263 for func, args, kwargs in self.items]

264

265 def \_\_reduce\_\_(self):

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in <listcomp>(.0)

261 with parallel\_backend(self.\_backend, n\_jobs=self.\_n\_jobs):

262 return [func(\*args, \*\*kwargs)

--> 263 for func, args, kwargs in self.items]

264

265 def \_\_reduce\_\_(self):

/usr/local/lib/python3.7/dist-packages/mlxtend/feature\_selection/exhaustive\_feature\_selector.py in \_calc\_score(selector, X, y, indices, \*\*fit\_params)

34 n\_jobs=1,

35 pre\_dispatch=selector.pre\_dispatch,

---> 36 fit\_params=fit\_params)

37 else:

38 selector.est\_.fit(X[:, indices], y, \*\*fit\_params)

/usr/local/lib/python3.7/dist-packages/sklearn/model\_selection/\_validation.py in cross\_val\_score(estimator, X, y, groups, scoring, cv, n\_jobs, verbose, fit\_params, pre\_dispatch, error\_score)

388 fit\_params=fit\_params,

389 pre\_dispatch=pre\_dispatch,

--> 390 error\_score=error\_score)

391 return cv\_results['test\_score']

392

/usr/local/lib/python3.7/dist-packages/sklearn/model\_selection/\_validation.py in cross\_validate(estimator, X, y, groups, scoring, cv, n\_jobs, verbose, fit\_params, pre\_dispatch, return\_train\_score, return\_estimator, error\_score)

234 return\_times=True, return\_estimator=return\_estimator,

235 error\_score=error\_score)

--> 236 for train, test in cv.split(X, y, groups))

237

238 zipped\_scores = list(zip(\*scores))

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in \_\_call\_\_(self, iterable)

1042 self.\_iterating = self.\_original\_iterator is not None

1043

-> 1044 while self.dispatch\_one\_batch(iterator):

1045 pass

1046

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in dispatch\_one\_batch(self, iterator)

857 return False

858 else:

--> 859 self.\_dispatch(tasks)

860 return True

861

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in \_dispatch(self, batch)

775 with self.\_lock:

776 job\_idx = len(self.\_jobs)

--> 777 job = self.\_backend.apply\_async(batch, callback=cb)

778 # A job can complete so quickly than its callback is

779 # called before we get here, causing self.\_jobs to

/usr/local/lib/python3.7/dist-packages/joblib/\_parallel\_backends.py in apply\_async(self, func, callback)

206 def apply\_async(self, func, callback=None):

207 """Schedule a func to be run"""

--> 208 result = ImmediateResult(func)

209 if callback:

210 callback(result)

/usr/local/lib/python3.7/dist-packages/joblib/\_parallel\_backends.py in \_\_init\_\_(self, batch)

570 # Don't delay the application, to avoid keeping the input

571 # arguments in memory

--> 572 self.results = batch()

573

574 def get(self):

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in \_\_call\_\_(self)

261 with parallel\_backend(self.\_backend, n\_jobs=self.\_n\_jobs):

262 return [func(\*args, \*\*kwargs)

--> 263 for func, args, kwargs in self.items]

264

265 def \_\_reduce\_\_(self):

/usr/local/lib/python3.7/dist-packages/joblib/parallel.py in <listcomp>(.0)

261 with parallel\_backend(self.\_backend, n\_jobs=self.\_n\_jobs):

262 return [func(\*args, \*\*kwargs)

--> 263 for func, args, kwargs in self.items]

264

265 def \_\_reduce\_\_(self):

/usr/local/lib/python3.7/dist-packages/sklearn/model\_selection/\_validation.py in \_fit\_and\_score(estimator, X, y, scorer, train, test, verbose, parameters, fit\_params, return\_train\_score, return\_parameters, return\_n\_test\_samples, return\_times, return\_estimator, error\_score)

542 else:

543 fit\_time = time.time() - start\_time

--> 544 test\_scores = \_score(estimator, X\_test, y\_test, scorer)

545 score\_time = time.time() - start\_time - fit\_time

546 if return\_train\_score:

/usr/local/lib/python3.7/dist-packages/sklearn/model\_selection/\_validation.py in \_score(estimator, X\_test, y\_test, scorer)

589 scores = scorer(estimator, X\_test)

590 else:

--> 591 scores = scorer(estimator, X\_test, y\_test)

592

593 error\_msg = ("scoring must return a number, got %s (%s) "

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/\_scorer.py in \_\_call\_\_(self, estimator, \*args, \*\*kwargs)

85 if isinstance(scorer, \_BaseScorer):

86 score = scorer.\_score(cached\_call, estimator,

---> 87 \*args, \*\*kwargs)

88 else:

89 score = scorer(estimator, \*args, \*\*kwargs)

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/\_scorer.py in \_score(self, method\_caller, estimator, X, y\_true, sample\_weight)

203 """

204

--> 205 y\_pred = method\_caller(estimator, "predict", X)

206 if sample\_weight is not None:

207 return self.\_sign \* self.\_score\_func(y\_true, y\_pred,

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/\_scorer.py in \_cached\_call(cache, estimator, method, \*args, \*\*kwargs)

50 """Call estimator with method and args and kwargs."""

51 if cache is None:

---> 52 return getattr(estimator, method)(\*args, \*\*kwargs)

53

54 try:

/usr/local/lib/python3.7/dist-packages/sklearn/neighbors/\_classification.py in predict(self, X)

185 for k, classes\_k in enumerate(classes\_):

186 if weights is None:

--> 187 mode, \_ = stats.mode(\_y[neigh\_ind, k], axis=1)

188 else:

189 mode, \_ = weighted\_mode(\_y[neigh\_ind, k], weights, axis=1)

/usr/local/lib/python3.7/dist-packages/scipy/stats/stats.py in mode(a, axis, nan\_policy)

498 counts = np.zeros(a\_view.shape[:-1], dtype=np.int)

499 for ind in inds:

--> 500 modes[ind], counts[ind] = \_mode1D(a\_view[ind])

501 newshape = list(a.shape)

502 newshape[axis] = 1

/usr/local/lib/python3.7/dist-packages/scipy/stats/stats.py in \_mode1D(a)

485

486 def \_mode1D(a):

--> 487 vals, cnts = np.unique(a, return\_counts=True)

488 return vals[cnts.argmax()], cnts.max()

489

<\_\_array\_function\_\_ internals> in unique(\*args, \*\*kwargs)

/usr/local/lib/python3.7/dist-packages/numpy/lib/arraysetops.py in unique(ar, return\_index, return\_inverse, return\_counts, axis)

259 ar = np.asanyarray(ar)

260 if axis is None:

--> 261 ret = \_unique1d(ar, return\_index, return\_inverse, return\_counts)

262 return \_unpack\_tuple(ret)

263

/usr/local/lib/python3.7/dist-packages/numpy/lib/arraysetops.py in \_unique1d(ar, return\_index, return\_inverse, return\_counts)

335 ret += (inv\_idx,)

336 if return\_counts:

--> 337 idx = np.concatenate(np.nonzero(mask) + ([mask.size],))

338 ret += (np.diff(idx),)

339 return ret

<\_\_array\_function\_\_ internals> in nonzero(\*args, \*\*kwargs)

/usr/local/lib/python3.7/dist-packages/numpy/core/fromnumeric.py in nonzero(a)

1906

1907 """

-> 1908 return \_wrapfunc(a, 'nonzero')

1909

1910

/usr/local/lib/python3.7/dist-packages/numpy/core/fromnumeric.py in \_wrapfunc(obj, method, \*args, \*\*kwds)

56

57 try:

---> 58 return bound(\*args, \*\*kwds)

59 except TypeError:

60 # A TypeError occurs if the object does have such a method in its

KeyboardInterrupt:

**Еще одна попытка отбора признаков**

In [ ]:

datafo = data2.copy()

datafo.head()

Out[ ]:

In [ ]:

*# DataFrame не содержащий целевой признак*

X\_ALL = datafo.drop('G3', axis=1)

In [ ]:

*# Функция для восстановления датафрейма*

*# на основе масштабированных данных*

**def** arr\_to\_df(arr\_scaled):

res = pd.DataFrame(arr\_scaled, columns=X\_ALL.columns)

**return** res

In [ ]:

*# Разделим выборку на обучающую и тестовую*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_ALL, datafo['G3'],

test\_size=0.2,

random\_state=1)

*# Преобразуем массивы в DataFrame*

X\_train\_df = arr\_to\_df(X\_train)

X\_test\_df = arr\_to\_df(X\_test)

X\_train\_df.shape, X\_test\_df.shape

Out[ ]:

((316, 32), (79, 32))

In [ ]:

**from** **sklearn.linear\_model** **import** LogisticRegression

*# Используем L1-регуляризацию*

e\_lr1 = LogisticRegression(C=1000, solver='liblinear', penalty='l1', max\_iter=500, random\_state=1)

e\_lr1.fit(X\_train, y\_train)

*# Коэффициенты регрессии*

e\_lr1.coef\_

Out[ ]:

array([[-5.39523998e+00, 9.34031358e+00, 5.33611194e+00,

9.24360706e-01, 7.06166891e-01, -9.79451059e+00,

-1.53848774e+00, 6.23027389e-01, -1.30800055e+00,

-2.10485448e+00, 1.66984498e-01, -3.87929522e+00,

1.54457672e+00, 3.15366255e+00, 2.56369073e+01,

-3.30240778e+00, 4.00600960e+00, 3.49301273e+00,

2.42457322e+00, -3.51529005e+00, -1.42877606e+00,

2.20248431e-01, 3.44033783e+00, -3.38033712e+00,

-3.87678059e+00, 0.00000000e+00, 5.84769675e+00,

-4.58215577e+00, -2.73614560e+00, -8.86725563e-01,

-2.00545233e+01, -1.83465878e+01],

[ 0.00000000e+00, -7.79210611e-01, 6.88264413e-01,

0.00000000e+00, -7.68552642e-01, 0.00000000e+00,

3.50038832e+00, 0.00000000e+00, -1.17719126e+00,

-3.42006576e+00, 5.11276837e-01, 0.00000000e+00,

0.00000000e+00, 7.21955889e-02, 0.00000000e+00,

0.00000000e+00, -5.43803981e+00, 0.00000000e+00,

-2.52288896e+00, 0.00000000e+00, -2.05412334e+00,

0.00000000e+00, 0.00000000e+00, -1.00203154e+00,

0.00000000e+00, 3.93092225e-01, 0.00000000e+00,

1.26933772e+00, -2.64709302e+00, -1.66814848e+00,

-1.57541395e+00, 1.77379124e+00],

[ 0.00000000e+00, 0.00000000e+00, -3.12968652e+00,

-9.90325484e+00, 0.00000000e+00, 0.00000000e+00,

1.55953807e+00, 2.46463465e-01, -2.22389129e+00,

-1.36758873e+00, -2.61007803e+00, 0.00000000e+00,

-4.78523325e+00, 0.00000000e+00, 0.00000000e+00,

6.14986190e+00, 1.37416275e+00, -2.40194662e+00,

-4.67738912e-01, -1.27957693e+00, -6.93007304e-01,

0.00000000e+00, 0.00000000e+00, -6.50910805e-01,

3.25529879e+00, 1.15492185e+00, 0.00000000e+00,

2.95191041e+00, 2.69742337e+00, -6.57338872e+00,

-6.89955142e+00, 3.75461713e+00],

[ 0.00000000e+00, 0.00000000e+00, -2.58827755e+00,

1.41473186e-01, 1.26347490e+01, -4.70648909e+00,

2.17572899e+00, -2.72622114e+00, -5.67114459e+00,

-5.20660011e+00, 5.19518650e+00, -9.67484648e+00,

2.77143143e+00, -5.88948243e-01, 1.65695836e+01,

4.40884664e+00, 5.32042510e+00, 1.07479619e+00,

1.28807530e+01, 0.00000000e+00, -9.91449880e+00,

-1.61311187e+01, -7.13442299e+00, 5.22134841e+00,

5.78696782e-01, -2.16120033e-01, -7.28830327e+00,

-6.95034010e+00, 0.00000000e+00, -1.09324165e+01,

-2.58840982e+01, 9.41870296e+00],

[ 8.53361701e+00, -7.66434145e+00, -7.35323887e+00,

-2.20928867e+01, -6.34140081e+00, -6.42828248e+00,

1.65398420e+01, -2.03217717e+01, 9.95865057e+00,

-3.60799052e-01, -2.35803076e+00, -1.36740131e+01,

-5.57821529e+00, 7.76865886e+00, 0.00000000e+00,

0.00000000e+00, 6.86339632e+00, -1.42161453e+01,

8.97262212e+00, -1.28590923e+01, -1.28827314e+01,

-3.10440564e+01, 0.00000000e+00, 2.58424489e+00,

1.48908636e+00, 0.00000000e+00, -2.04470565e+00,

-4.48396824e-01, 5.51260822e+00, -4.09093219e+01,

-5.61812379e+00, 1.32817000e+01],

[-5.25905854e-01, 1.82881696e+00, 3.29974300e-01,

-4.77896416e-01, 2.35469779e-01, 1.18161142e+00,

-7.12219772e-01, 2.23121568e-01, 6.43424044e-02,

-2.70252359e-01, -4.18762596e-01, 1.34773767e+00,

2.07577237e-01, -8.34253012e-02, 7.80083059e+00,

8.27416358e-01, -3.35333313e-01, 2.63751299e+00,

-2.22608476e-01, 4.33861404e-01, -1.01532991e+00,

8.71033643e-01, -8.75332656e-01, -3.41260506e-02,

-1.35213802e-01, 1.23644497e-01, -6.34459600e-01,

-8.75719757e-02, -5.32695972e-02, -2.24622684e+00,

1.56611584e-01, 7.42013112e-01],

[ 7.41235012e-01, -4.02057672e-01, 6.22020774e-01,

1.26083757e+00, -1.64498125e-01, 5.28040176e-01,

4.08927388e-01, -4.58842858e-01, 3.17935574e-01,

3.36641508e-01, -1.73235909e-01, -5.05605220e-01,

-6.03519195e-01, 6.56459230e-01, 6.25654557e+00,

5.25511689e-01, 3.34148623e-01, -6.46702735e-01,

8.36608078e-01, 1.84106837e-01, -4.01613657e-01,

-6.49641686e-01, 5.67913979e-01, -1.71367425e-01,

-3.96331277e-02, -8.28383084e-04, 1.17122456e+00,

-5.88282709e-01, 4.44587411e-01, 4.68464589e-01,

-1.68500799e+00, 9.97825998e-01],

[ 5.10149355e-01, 3.54069647e-01, -3.45367795e-01,

-1.45682497e-01, 3.34304213e-02, 4.87150686e-03,

4.71216745e-01, -8.35595318e-01, -2.49576329e-01,

-1.07507028e-01, -4.94157328e-02, -5.91536646e-01,

7.78557499e-02, 3.32931274e-01, 1.76251461e+00,

-1.76436568e-01, -4.88919159e-01, 4.28904266e-01,

1.12891630e-01, 4.25015739e-01, 3.83295504e-01,

-1.95594597e-01, -9.69152440e-01, 4.96991222e-01,

1.18965125e-02, -4.31467276e-01, -3.46496369e-01,

9.49001957e-01, -4.32249419e-01, -1.17145474e+00,

2.09766763e-01, 6.08224177e-01],

[-6.77123060e-01, -7.75931613e-01, 1.01099208e-01,

4.99036769e-01, -4.11096554e-01, -6.83572281e-01,

-9.98253013e-01, 6.41252124e-01, 5.95721256e-01,

-9.55201958e-02, -1.89366614e-01, 4.95467768e-01,

7.21136091e-04, 1.33899123e-01, 7.93244277e+00,

8.87574184e-01, 9.02130538e-02, -3.38936806e-02,

-1.19580792e-01, -2.85463763e-01, 4.23897517e+00,

6.92784485e-02, 2.15308226e-01, -1.68764552e-01,

8.19507974e-02, -1.58758180e-01, 3.55640040e-04,

-9.84684529e-02, 1.07230714e-01, -6.60300635e-01,

7.82681786e-01, 4.47254886e-01],

[-4.81392361e-01, 1.04280079e+00, 4.70306180e-02,

-6.60179780e-01, 6.91882614e-01, -2.74289300e-01,

-1.70890100e-01, -7.13202819e-02, -1.72968541e-01,

5.49157660e-01, -1.68081922e-02, -5.16283753e-01,

-3.35227615e-01, 1.66225047e-01, 2.60377743e+00,

3.88154357e-01, -1.45242911e-01, -6.67127564e-03,

-4.39810127e-01, -8.98059672e-01, 1.10443706e-01,

-6.89725375e-01, 7.17955982e-01, -9.96421732e-03,

8.39805527e-02, -1.92984201e-01, -2.34873409e-02,

1.38766946e-01, 2.07671933e-01, -1.53838499e-01,

6.38848151e-01, -2.34775682e-01],

[ 1.58154634e+00, -3.22436654e-01, -4.29675097e-01,

-1.43788112e-02, -1.50706420e-01, 1.97004056e-01,

-5.52212863e-01, 7.10622099e-01, -1.38379822e-01,

-7.53285212e-01, -1.10993288e-01, 3.22212483e-01,

7.48482988e-01, -4.27715370e-01, 9.32419852e+00,

8.02850340e-02, -1.43653925e+00, 1.83377633e+00,

-1.12613714e+00, 1.38297177e+00, 4.40730232e+00,

3.71282021e-01, 6.26380757e-01, -1.31237281e-01,

-9.26604096e-02, -5.16227394e-01, 1.00630081e+00,

-2.05235782e-01, 4.78374442e-01, -5.50315209e-01,

2.57178229e+00, 1.33577664e+00],

[-2.72149545e-01, 8.03351606e-01, 1.70066147e-01,

1.31694047e+00, -1.23011316e+00, 1.44040105e+00,

-9.49303414e-02, 3.96570473e-02, -2.81923972e-01,

-9.25000325e-02, 4.78447901e-01, -4.83853854e-01,

-4.42642800e-01, 1.15599713e-02, 7.33020992e+00,

-5.50904537e-01, 5.38310146e-01, 7.22860611e-03,

-2.86401555e-01, -5.41778285e-01, 3.71647315e+00,

-1.59388691e+00, 2.85428556e-01, 2.28067582e-01,

-1.44235113e-01, 7.59928846e-01, 1.27573930e-02,

-5.51746043e-01, -4.83175852e-01, -3.94399117e-01,

1.80794286e+00, 2.71792303e-01],

[-1.14756595e+00, 6.48569154e-01, 1.33881280e-01,

1.23258774e+00, 2.16902325e-01, 2.82321679e-01,

-9.66531659e-02, 3.50207901e-01, -4.32198478e-01,

-1.28421760e-01, 5.48576317e-02, 5.73305287e-01,

3.38290483e-01, 2.77594414e-01, 6.96997979e+00,

-3.14491185e-02, -9.05208013e-01, 1.31437870e+00,

-2.02551746e-01, -1.76132755e+00, 1.49277472e+00,

9.18962455e-01, -4.36191680e-01, -7.81201570e-01,

1.64764743e-01, -1.03559587e-01, -1.06501303e+00,

-1.95626914e-01, 4.13573650e-01, 1.93817011e-01,

3.00001099e+00, -2.77609788e-01],

[-1.34164374e+02, -2.05670348e+01, 4.05665352e+01,

-2.75617542e+01, 5.12587769e+01, 2.61536464e+01,

6.04307456e+01, -6.96834298e+01, -5.48511269e+01,

2.91715201e+01, -2.23366552e+01, 2.02466928e+01,

-9.09327840e+01, -3.87089246e+01, 3.42174439e+02,

0.00000000e+00, 1.03853842e+02, -5.15463355e+01,

-8.73263081e+01, 4.04001982e+01, -2.31816620e+02,

2.28474592e+02, 6.99911894e+01, 5.90246878e+01,

3.12556509e+01, 9.01230891e+00, 3.78884446e+01,

2.94485746e+01, 5.46603246e+01, 1.24567941e+02,

1.40972020e+02, -1.12279477e+01],

[-2.63267321e+01, 0.00000000e+00, 3.09809344e+00,

-3.03357705e+00, 4.20566115e+00, 0.00000000e+00,

-8.88688949e-02, 8.82122775e+00, -4.46483986e-02,

0.00000000e+00, 1.38457100e+00, -9.58669344e-01,

2.22925480e+00, 3.60929075e+00, 0.00000000e+00,

0.00000000e+00, 8.77941640e+00, -9.28748300e+00,

0.00000000e+00, 0.00000000e+00, -1.75134376e+01,

-3.95461076e+00, 0.00000000e+00, 3.42959526e+00,

-5.43173937e-01, -2.78629339e+00, 3.62513889e+00,

0.00000000e+00, -2.48451592e+00, 7.96230542e+00,

3.96565809e+00, -4.71250915e+00],

[-7.40980029e+00, -4.56856808e+00, 3.39566600e+00,

0.00000000e+00, -3.76564626e+00, 1.11604637e+01,

4.34425450e+00, -4.64345876e-01, 2.27385874e+00,

-2.42599323e-01, -4.70057919e+00, -4.51576848e-01,

7.89718925e+00, 2.95463574e+00, 3.78821117e+01,

0.00000000e+00, 6.01890308e+00, 1.33808060e+01,

1.34909235e+01, -1.71392399e+00, -3.13417748e+01,

-1.37939308e+01, -1.20271365e+01, 1.24972231e+00,

2.09896710e+00, 0.00000000e+00, -4.89519248e+00,

-3.74730128e+00, -2.04908407e+00, 0.00000000e+00,

4.83401852e+01, 5.30217544e+00],

[ 0.00000000e+00, 0.00000000e+00, -1.43000497e+00,

-4.30785653e+00, 8.41814327e+00, -8.88966515e+00,

3.96067416e+00, 0.00000000e+00, 0.00000000e+00,

-5.56273923e-01, -1.51747545e+00, -4.11164544e-01,

0.00000000e+00, -2.93891095e-01, 0.00000000e+00,

0.00000000e+00, -7.01938593e+00, -9.93006950e-01,

-6.59178537e+00, -6.05757042e-01, -3.42924944e+00,

0.00000000e+00, -1.77932197e+00, 0.00000000e+00,

0.00000000e+00, 0.00000000e+00, -6.25154009e-01,

-3.93240729e+00, -1.79336996e+00, 3.44410582e+00,

3.16250112e+00, -4.32884298e-01],

[ 0.00000000e+00, 5.88724337e-01, 0.00000000e+00,

0.00000000e+00, 0.00000000e+00, 0.00000000e+00,

4.08094361e+00, 1.37313958e-01, -3.95147494e+00,

-1.02403445e-01, 0.00000000e+00, 0.00000000e+00,

-1.44550257e+00, 2.37481339e+00, 0.00000000e+00,

0.00000000e+00, -4.51422503e+00, -2.44994571e+00,

0.00000000e+00, 0.00000000e+00, -4.40187439e+00,

0.00000000e+00, -2.09622884e+00, 0.00000000e+00,

-2.89950723e-01, -1.72178931e+00, 0.00000000e+00,

-7.72628463e-01, -3.54917530e-01, 2.30444371e+00,

1.59642894e+00, 1.92682293e+00]])

In [ ]:

**from** **sklearn.feature\_selection** **import** SelectFromModel

sel\_e\_lr1 = SelectFromModel(e\_lr1)

sel\_e\_lr1.fit(X\_train, y\_train)

sel\_e\_lr1.get\_support()

Out[ ]:

array([ True, True, True, True, True, True, True, True, True,

True, True, True, True, True, True, True, True, True,

True, True, True, True, True, True, True, True, True,

True, True, True, True, True])

**Обучаем модель (минимальная обработка данных)**

In [ ]:

**from** **sklearn.model\_selection** **import** train\_test\_split

In [ ]:

x = data\_encoded.drop([data\_encoded.columns[-1]], axis=1)

x

Out[ ]:

395 rows × 32 columns

In [ ]:

**def** arr\_to\_df(arr\_scaled):

res = pd.DataFrame(arr\_scaled, columns=x.columns)

**return** res

In [ ]:

*# Разделим выборку на обучающую и тестовую*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, data\_encoded['G3'],

test\_size=0.2,

random\_state=1)

*# Преобразуем массивы в DataFrame*

X\_train\_df = arr\_to\_df(X\_train)

X\_test\_df = arr\_to\_df(X\_test)

X\_train\_df.shape, X\_test\_df.shape, y\_train.shape, y\_test.shape

Out[ ]:

((316, 32), (79, 32), (316,), (79,))

In [ ]:

**class** **MetricLogger**:

**def** \_\_init\_\_(self):

self.df = pd.DataFrame(

{'metric': pd.Series([], dtype='str'),

'alg': pd.Series([], dtype='str'),

'value': pd.Series([], dtype='float')})

**def** add(self, metric, alg, value):

*"""*

*Добавление значения*

*"""*

*# Удаление значения если оно уже было ранее добавлено*

self.df.drop(self.df[(self.df['metric']==metric)&(self.df['alg']==alg)].index, inplace = **True**)

*# Добавление нового значения*

temp = [{'metric':metric, 'alg':alg, 'value':value}]

self.df = self.df.append(temp, ignore\_index=**True**)

**def** get\_data\_for\_metric(self, metric, ascending=**True**):

*"""*

*Формирование данных с фильтром по метрике*

*"""*

temp\_data = self.df[self.df['metric']==metric]

temp\_data\_2 = temp\_data.sort\_values(by='value', ascending=ascending)

**return** temp\_data\_2['alg'].values, temp\_data\_2['value'].values

**def** plot(self, str\_header, metric, ascending=**True**, figsize=(5, 5)):

*"""*

*Вывод графика*

*"""*

array\_labels, array\_metric = self.get\_data\_for\_metric(metric, ascending)

fig, ax1 = plt.subplots(figsize=figsize)

pos = np.arange(len(array\_metric))

rects = ax1.barh(pos, array\_metric,

align='center',

height=0.5,

tick\_label=array\_labels)

ax1.set\_title(str\_header)

**for** a,b **in** zip(pos, array\_metric):

plt.text(0.5, a-0.05, str(round(b,3)), color='white')

plt.show()

In [ ]:

**from** **sklearn.linear\_model** **import** LinearRegression

**from** **sklearn.neighbors** **import** KNeighborsRegressor

**from** **sklearn.tree** **import** DecisionTreeRegressor

**from** **sklearn.ensemble** **import** RandomForestRegressor

**from** **sklearn.ensemble** **import** GradientBoostingRegressor

**from** **sklearn.svm** **import** SVR

**from** **sklearn.metrics** **import** mean\_squared\_error

In [ ]:

clas\_models\_dict = {'LinR': LinearRegression(),

'SVR': SVR(),

'KNN\_5':KNeighborsRegressor(n\_neighbors=5),

'Tree':DecisionTreeRegressor(random\_state=1),

'GB': GradientBoostingRegressor(random\_state=1),

'RF':RandomForestRegressor(n\_estimators=50, random\_state=1)}

In [ ]:

X\_data\_dict = {'Basic': (X\_train\_df, X\_test\_df)}

In [ ]:

**def** test\_models(clas\_models\_dict, X\_train, X\_test, y\_train, y\_test):

logger = MetricLogger()

**for** model\_name, model **in** clas\_models\_dict.items():

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

logger.add(model\_name, 'Basic', mse)

**return** logger

In [ ]:

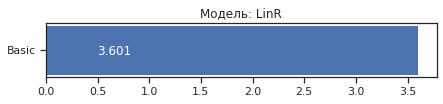
logger = test\_models(clas\_models\_dict, X\_train\_df, X\_test\_df, y\_train, y\_test)

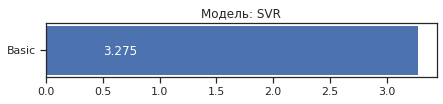
In [ ]:

*# Построим графики метрик качества модели*

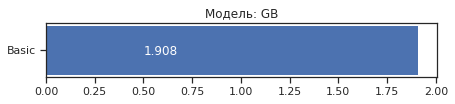
**for** model **in** clas\_models\_dict:

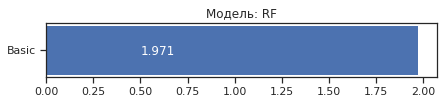
logger.plot('Модель: ' + model, model, figsize=(7, 1))

****

****

****

****

****

**Обучаем модель (полная обработка данных)**

In [ ]:

data\_res = datafo.copy()

*# data\_res = data\_res[['absences\_exp1','higher','Medu','goout','romantic','age','G2','G3']]*

data\_res.head()

Out[ ]:

In [ ]:

x = data\_res.drop([data\_res.columns[-1]], axis=1)

x

Out[ ]:

395 rows × 32 columns

In [ ]:

*# Разделим выборку на обучающую и тестовую*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, data\_res['G3'],

test\_size=0.2,

random\_state=1)

*# Преобразуем массивы в DataFrame*

X\_train\_df = arr\_to\_df(X\_train)

X\_test\_df = arr\_to\_df(X\_test)

X\_train\_df.shape, X\_test\_df.shape, y\_train.shape, y\_test.shape

Out[ ]:

((316, 32), (79, 32), (316,), (79,))

In [ ]:

X\_data\_dict = {'Basic': (X\_train\_df, X\_test\_df)}

In [ ]:

logger = test\_models(clas\_models\_dict, X\_train\_df, X\_test\_df, y\_train, y\_test)

In [ ]:

*# Построим графики метрик качества модели*

**for** model **in** clas\_models\_dict:

logger.plot('Модель: ' + model, model, figsize=(10, 10))

---------------------------------------------------------------------------

ValueError Traceback (most recent call last)

/usr/local/lib/python3.7/dist-packages/IPython/core/formatters.py in \_\_call\_\_(self, obj)

332 pass

333 else:

--> 334 return printer(obj)

335 # Finally look for special method names

336 method = get\_real\_method(obj, self.print\_method)

/usr/local/lib/python3.7/dist-packages/IPython/core/pylabtools.py in <lambda>(fig)

239

240 if 'png' in formats:

--> 241 png\_formatter.for\_type(Figure, lambda fig: print\_figure(fig, 'png', \*\*kwargs))

242 if 'retina' in formats or 'png2x' in formats:

243 png\_formatter.for\_type(Figure, lambda fig: retina\_figure(fig, \*\*kwargs))

/usr/local/lib/python3.7/dist-packages/IPython/core/pylabtools.py in print\_figure(fig, fmt, bbox\_inches, \*\*kwargs)

123

124 bytes\_io = BytesIO()

--> 125 fig.canvas.print\_figure(bytes\_io, \*\*kw)

126 data = bytes\_io.getvalue()

127 if fmt == 'svg':

/usr/local/lib/python3.7/dist-packages/matplotlib/backend\_bases.py in print\_figure(self, filename, dpi, facecolor, edgecolor, orientation, format, bbox\_inches, \*\*kwargs)

2124 orientation=orientation,

2125 bbox\_inches\_restore=\_bbox\_inches\_restore,

-> 2126 \*\*kwargs)

2127 finally:

2128 if bbox\_inches and restore\_bbox:

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in print\_png(self, filename\_or\_obj, metadata, pil\_kwargs, \*args, \*\*kwargs)

512 }

513

--> 514 FigureCanvasAgg.draw(self)

515 if pil\_kwargs is not None:

516 from PIL import Image

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in draw(self)

386 Draw the figure using the renderer.

387 """

--> 388 self.renderer = self.get\_renderer(cleared=True)

389 # Acquire a lock on the shared font cache.

390 with RendererAgg.lock, \

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in get\_renderer(self, cleared)

402 and getattr(self, "\_lastKey", None) == key)

403 if not reuse\_renderer:

--> 404 self.renderer = RendererAgg(w, h, self.figure.dpi)

405 self.\_lastKey = key

406 elif cleared:

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in \_\_init\_\_(self, width, height, dpi)

90 self.width = width

91 self.height = height

---> 92 self.\_renderer = \_RendererAgg(int(width), int(height), dpi)

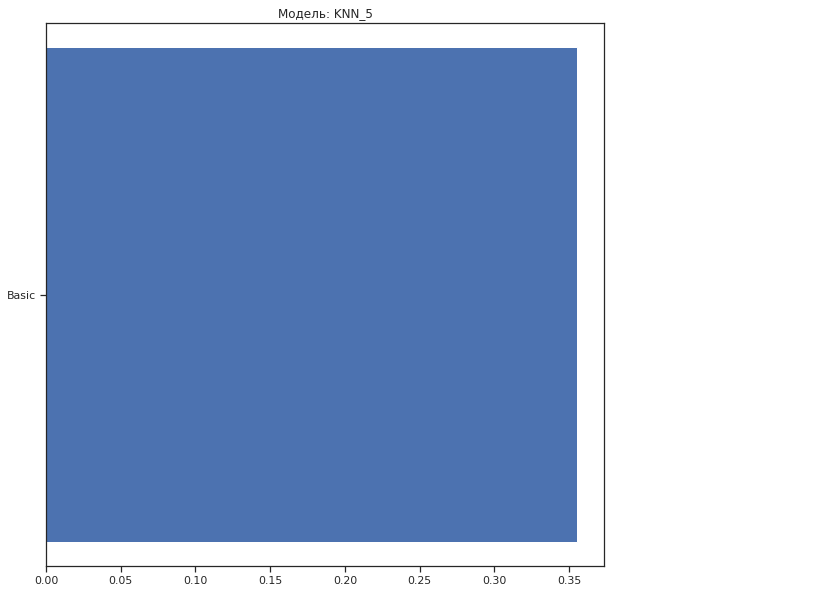
93 self.\_filter\_renderers = []

94

ValueError: Image size of -537773730x606 pixels is too large. It must be less than 2^16 in each direction.

<Figure size 720x720 with 1 Axes>

****

****

---------------------------------------------------------------------------

ValueError Traceback (most recent call last)

/usr/local/lib/python3.7/dist-packages/IPython/core/formatters.py in \_\_call\_\_(self, obj)

332 pass

333 else:

--> 334 return printer(obj)

335 # Finally look for special method names

336 method = get\_real\_method(obj, self.print\_method)

/usr/local/lib/python3.7/dist-packages/IPython/core/pylabtools.py in <lambda>(fig)

239

240 if 'png' in formats:

--> 241 png\_formatter.for\_type(Figure, lambda fig: print\_figure(fig, 'png', \*\*kwargs))

242 if 'retina' in formats or 'png2x' in formats:

243 png\_formatter.for\_type(Figure, lambda fig: retina\_figure(fig, \*\*kwargs))

/usr/local/lib/python3.7/dist-packages/IPython/core/pylabtools.py in print\_figure(fig, fmt, bbox\_inches, \*\*kwargs)

123

124 bytes\_io = BytesIO()

--> 125 fig.canvas.print\_figure(bytes\_io, \*\*kw)

126 data = bytes\_io.getvalue()

127 if fmt == 'svg':

/usr/local/lib/python3.7/dist-packages/matplotlib/backend\_bases.py in print\_figure(self, filename, dpi, facecolor, edgecolor, orientation, format, bbox\_inches, \*\*kwargs)

2124 orientation=orientation,

2125 bbox\_inches\_restore=\_bbox\_inches\_restore,

-> 2126 \*\*kwargs)

2127 finally:

2128 if bbox\_inches and restore\_bbox:

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in print\_png(self, filename\_or\_obj, metadata, pil\_kwargs, \*args, \*\*kwargs)

512 }

513

--> 514 FigureCanvasAgg.draw(self)

515 if pil\_kwargs is not None:

516 from PIL import Image

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in draw(self)

386 Draw the figure using the renderer.

387 """

--> 388 self.renderer = self.get\_renderer(cleared=True)

389 # Acquire a lock on the shared font cache.

390 with RendererAgg.lock, \

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in get\_renderer(self, cleared)

402 and getattr(self, "\_lastKey", None) == key)

403 if not reuse\_renderer:

--> 404 self.renderer = RendererAgg(w, h, self.figure.dpi)

405 self.\_lastKey = key

406 elif cleared:

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in \_\_init\_\_(self, width, height, dpi)

90 self.width = width

91 self.height = height

---> 92 self.\_renderer = \_RendererAgg(int(width), int(height), dpi)

93 self.\_filter\_renderers = []

94

ValueError: Image size of 35627669x606 pixels is too large. It must be less than 2^16 in each direction.

<Figure size 720x720 with 1 Axes>

---------------------------------------------------------------------------

ValueError Traceback (most recent call last)

/usr/local/lib/python3.7/dist-packages/IPython/core/formatters.py in \_\_call\_\_(self, obj)

332 pass

333 else:

--> 334 return printer(obj)

335 # Finally look for special method names

336 method = get\_real\_method(obj, self.print\_method)

/usr/local/lib/python3.7/dist-packages/IPython/core/pylabtools.py in <lambda>(fig)

239

240 if 'png' in formats:

--> 241 png\_formatter.for\_type(Figure, lambda fig: print\_figure(fig, 'png', \*\*kwargs))

242 if 'retina' in formats or 'png2x' in formats:

243 png\_formatter.for\_type(Figure, lambda fig: retina\_figure(fig, \*\*kwargs))

/usr/local/lib/python3.7/dist-packages/IPython/core/pylabtools.py in print\_figure(fig, fmt, bbox\_inches, \*\*kwargs)

123

124 bytes\_io = BytesIO()

--> 125 fig.canvas.print\_figure(bytes\_io, \*\*kw)

126 data = bytes\_io.getvalue()

127 if fmt == 'svg':

/usr/local/lib/python3.7/dist-packages/matplotlib/backend\_bases.py in print\_figure(self, filename, dpi, facecolor, edgecolor, orientation, format, bbox\_inches, \*\*kwargs)

2124 orientation=orientation,

2125 bbox\_inches\_restore=\_bbox\_inches\_restore,

-> 2126 \*\*kwargs)

2127 finally:

2128 if bbox\_inches and restore\_bbox:

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in print\_png(self, filename\_or\_obj, metadata, pil\_kwargs, \*args, \*\*kwargs)

512 }

513

--> 514 FigureCanvasAgg.draw(self)

515 if pil\_kwargs is not None:

516 from PIL import Image

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in draw(self)

386 Draw the figure using the renderer.

387 """

--> 388 self.renderer = self.get\_renderer(cleared=True)

389 # Acquire a lock on the shared font cache.

390 with RendererAgg.lock, \

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in get\_renderer(self, cleared)

402 and getattr(self, "\_lastKey", None) == key)

403 if not reuse\_renderer:

--> 404 self.renderer = RendererAgg(w, h, self.figure.dpi)

405 self.\_lastKey = key

406 elif cleared:

/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend\_agg.py in \_\_init\_\_(self, width, height, dpi)

90 self.width = width

91 self.height = height

---> 92 self.\_renderer = \_RendererAgg(int(width), int(height), dpi)

93 self.\_filter\_renderers = []

94

ValueError: Image size of 165634x593 pixels is too large. It must be less than 2^16 in each direction.

<Figure size 720x720 with 1 Axes>

**AUTOML**

In [ ]:

!pip3 install mljar-supervised

!pip3 install delayed

Collecting mljar-supervised

Downloading https://files.pythonhosted.org/packages/0b/45/878916fb350cbe08ce51365305c54ee56e253c7be7979e0a18d753ebbb84/mljar-supervised-0.10.4.tar.gz (109kB)

|████████████████████████████████| 112kB 3.9MB/s

Collecting numpy>=1.20.0

Downloading https://files.pythonhosted.org/packages/a5/42/560d269f604d3e186a57c21a363e77e199358d054884e61b73e405dd217c/numpy-1.20.3-cp37-cp37m-manylinux\_2\_12\_x86\_64.manylinux2010\_x86\_64.whl (15.3MB)

|████████████████████████████████| 15.3MB 278kB/s

Collecting pandas==1.2.0

Downloading https://files.pythonhosted.org/packages/ff/bd/fb376f9fbad92b9a6efdbb30ff32c80f3cba1368689309cbb5566364af5c/pandas-1.2.0-cp37-cp37m-manylinux1\_x86\_64.whl (9.9MB)

|████████████████████████████████| 9.9MB 42.1MB/s

Collecting scipy==1.6.1

Downloading https://files.pythonhosted.org/packages/b6/3a/9e0649ab2d5ade703baa70ef980aa08739226e5d6a642f084bb201a92fc2/scipy-1.6.1-cp37-cp37m-manylinux1\_x86\_64.whl (27.4MB)

|████████████████████████████████| 27.4MB 108kB/s

Collecting scikit-learn==0.24.2

Downloading https://files.pythonhosted.org/packages/a8/eb/a48f25c967526b66d5f1fa7a984594f0bf0a5afafa94a8c4dbc317744620/scikit\_learn-0.24.2-cp37-cp37m-manylinux2010\_x86\_64.whl (22.3MB)

|████████████████████████████████| 22.3MB 1.6MB/s

Collecting xgboost==1.3.3

Downloading https://files.pythonhosted.org/packages/2e/57/bf5026701c384decd2b995eb39d86587a103ba4eb26f8a9b1811db0896d3/xgboost-1.3.3-py3-none-manylinux2010\_x86\_64.whl (157.5MB)

|████████████████████████████████| 157.5MB 54kB/s

Collecting lightgbm==3.0.0

Downloading https://files.pythonhosted.org/packages/25/47/f8ef524e15ff86f5246cb4e1cee200b747ddb2536429fa021cc5f17ea40a/lightgbm-3.0.0-py2.py3-none-manylinux1\_x86\_64.whl (1.7MB)

|████████████████████████████████| 1.8MB 14.0MB/s

Collecting catboost==0.24.4

Downloading https://files.pythonhosted.org/packages/96/3b/bb419654adcf7efff42ed8a3f84e50c8f236424b7ed1cc8ccd290852e003/catboost-0.24.4-cp37-none-manylinux1\_x86\_64.whl (65.7MB)

|████████████████████████████████| 65.7MB 149kB/s

Requirement already satisfied: joblib==1.0.1 in /usr/local/lib/python3.7/dist-packages (from mljar-supervised) (1.0.1)

Requirement already satisfied: cloudpickle==1.3.0 in /usr/local/lib/python3.7/dist-packages (from mljar-supervised) (1.3.0)

Requirement already satisfied: pyarrow>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from mljar-supervised) (3.0.0)

Collecting tabulate==0.8.7

Downloading https://files.pythonhosted.org/packages/c4/f4/770ae9385990f5a19a91431163d262182d3203662ea2b5739d0fcfc080f1/tabulate-0.8.7-py3-none-any.whl

Requirement already satisfied: matplotlib>=3.2.2 in /usr/local/lib/python3.7/dist-packages (from mljar-supervised) (3.2.2)

Collecting dtreeviz==1.3

Downloading https://files.pythonhosted.org/packages/35/95/f54aa86548a549da2a6b4159e1d2801b74bf27e01b669e50f3bf26ce30b8/dtreeviz-1.3.tar.gz (60kB)

|████████████████████████████████| 61kB 5.4MB/s

Collecting shap==0.36.0

Downloading https://files.pythonhosted.org/packages/d2/17/37ee6c79cafbd9bb7423b54e55ea90beec66aa7638664d607bcc28de0bae/shap-0.36.0.tar.gz (319kB)

|████████████████████████████████| 327kB 10.1MB/s

Requirement already satisfied: seaborn==0.11.1 in /usr/local/lib/python3.7/dist-packages (from mljar-supervised) (0.11.1)

Collecting wordcloud==1.8.1

Downloading https://files.pythonhosted.org/packages/1b/06/0516bdba2ebdc0d5bd476aa66f94666dd0ad6b9abda723fdf28e451db919/wordcloud-1.8.1-cp37-cp37m-manylinux1\_x86\_64.whl (366kB)

|████████████████████████████████| 368kB 21.2MB/s

Collecting category\_encoders==2.2.2

Downloading https://files.pythonhosted.org/packages/44/57/fcef41c248701ee62e8325026b90c432adea35555cbc870aff9cfba23727/category\_encoders-2.2.2-py2.py3-none-any.whl (80kB)

|████████████████████████████████| 81kB 5.2MB/s

Collecting optuna==2.7.0

Downloading https://files.pythonhosted.org/packages/2b/21/d13081805e1e1afc71f5bb743ece324c8bd576237c51b899ecb38a717502/optuna-2.7.0-py3-none-any.whl (293kB)

|████████████████████████████████| 296kB 29.6MB/s

Collecting scikit-plot==0.3.7

Downloading https://files.pythonhosted.org/packages/7c/47/32520e259340c140a4ad27c1b97050dd3254fdc517b1d59974d47037510e/scikit\_plot-0.3.7-py3-none-any.whl

Requirement already satisfied: markdown in /usr/local/lib/python3.7/dist-packages (from mljar-supervised) (3.3.4)

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas==1.2.0->mljar-supervised) (2018.9)

Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas==1.2.0->mljar-supervised) (2.8.1)

Collecting threadpoolctl>=2.0.0

Downloading https://files.pythonhosted.org/packages/f7/12/ec3f2e203afa394a149911729357aa48affc59c20e2c1c8297a60f33f133/threadpoolctl-2.1.0-py3-none-any.whl

Requirement already satisfied: plotly in /usr/local/lib/python3.7/dist-packages (from catboost==0.24.4->mljar-supervised) (4.4.1)

Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from catboost==0.24.4->mljar-supervised) (1.15.0)

Requirement already satisfied: graphviz in /usr/local/lib/python3.7/dist-packages (from catboost==0.24.4->mljar-supervised) (0.10.1)

Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=3.2.2->mljar-supervised) (0.10.0)

Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=3.2.2->mljar-supervised) (2.4.7)

Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=3.2.2->mljar-supervised) (1.3.1)

Collecting colour

Downloading https://files.pythonhosted.org/packages/74/46/e81907704ab203206769dee1385dc77e1407576ff8f50a0681d0a6b541be/colour-0.1.5-py2.py3-none-any.whl

Requirement already satisfied: pytest in /usr/local/lib/python3.7/dist-packages (from dtreeviz==1.3->mljar-supervised) (3.6.4)

Requirement already satisfied: tqdm>4.25.0 in /usr/local/lib/python3.7/dist-packages (from shap==0.36.0->mljar-supervised) (4.41.1)

Collecting slicer

Downloading https://files.pythonhosted.org/packages/78/c2/b3f55dfdb8af9812fdb9baf70cacf3b9e82e505b2bd4324d588888b81202/slicer-0.0.7-py3-none-any.whl

Requirement already satisfied: numba in /usr/local/lib/python3.7/dist-packages (from shap==0.36.0->mljar-supervised) (0.51.2)

Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from wordcloud==1.8.1->mljar-supervised) (7.1.2)

Requirement already satisfied: patsy>=0.5.1 in /usr/local/lib/python3.7/dist-packages (from category\_encoders==2.2.2->mljar-supervised) (0.5.1)

Requirement already satisfied: statsmodels>=0.9.0 in /usr/local/lib/python3.7/dist-packages (from category\_encoders==2.2.2->mljar-supervised) (0.10.2)

Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packages (from optuna==2.7.0->mljar-supervised) (20.9)

Requirement already satisfied: sqlalchemy>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from optuna==2.7.0->mljar-supervised) (1.4.15)

Collecting cmaes>=0.8.2

Downloading https://files.pythonhosted.org/packages/01/1f/43b01223a0366171f474320c6e966c39a11587287f098a5f09809b45e05f/cmaes-0.8.2-py3-none-any.whl

Collecting colorlog

Downloading https://files.pythonhosted.org/packages/32/e6/e9ddc6fa1104fda718338b341e4b3dc31cd8039ab29e52fc73b508515361/colorlog-5.0.1-py2.py3-none-any.whl

Collecting cliff

Downloading https://files.pythonhosted.org/packages/a2/d6/7d9acb68a77acd140be7fececb7f2701b2a29d2da9c54184cb8f93509590/cliff-3.7.0-py3-none-any.whl (80kB)

|████████████████████████████████| 81kB 7.4MB/s

Collecting alembic

Downloading https://files.pythonhosted.org/packages/eb/bd/c3486fd57a3eec5162a2e32e8f05880c990f0d92b03d268342d2e8fe7032/alembic-1.6.4-py2.py3-none-any.whl (164kB)

|████████████████████████████████| 174kB 43.4MB/s

Requirement already satisfied: importlib-metadata; python\_version < "3.8" in /usr/local/lib/python3.7/dist-packages (from markdown->mljar-supervised) (4.0.1)

Requirement already satisfied: retrying>=1.3.3 in /usr/local/lib/python3.7/dist-packages (from plotly->catboost==0.24.4->mljar-supervised) (1.3.3)

Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.7/dist-packages (from pytest->dtreeviz==1.3->mljar-supervised) (21.2.0)

Requirement already satisfied: py>=1.5.0 in /usr/local/lib/python3.7/dist-packages (from pytest->dtreeviz==1.3->mljar-supervised) (1.10.0)

Requirement already satisfied: pluggy<0.8,>=0.5 in /usr/local/lib/python3.7/dist-packages (from pytest->dtreeviz==1.3->mljar-supervised) (0.7.1)

Requirement already satisfied: atomicwrites>=1.0 in /usr/local/lib/python3.7/dist-packages (from pytest->dtreeviz==1.3->mljar-supervised) (1.4.0)

Requirement already satisfied: more-itertools>=4.0.0 in /usr/local/lib/python3.7/dist-packages (from pytest->dtreeviz==1.3->mljar-supervised) (8.7.0)

Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from pytest->dtreeviz==1.3->mljar-supervised) (56.1.0)

Requirement already satisfied: llvmlite<0.35,>=0.34.0.dev0 in /usr/local/lib/python3.7/dist-packages (from numba->shap==0.36.0->mljar-supervised) (0.34.0)

Requirement already satisfied: greenlet!=0.4.17; python\_version >= "3" in /usr/local/lib/python3.7/dist-packages (from sqlalchemy>=1.1.0->optuna==2.7.0->mljar-supervised) (1.1.0)

Collecting stevedore>=2.0.1

Downloading https://files.pythonhosted.org/packages/d4/49/b602307aeac3df3384ff1fcd05da9c0376c622a6c48bb5325f28ab165b57/stevedore-3.3.0-py3-none-any.whl (49kB)

|████████████████████████████████| 51kB 5.2MB/s

Requirement already satisfied: PrettyTable>=0.7.2 in /usr/local/lib/python3.7/dist-packages (from cliff->optuna==2.7.0->mljar-supervised) (2.1.0)

Requirement already satisfied: PyYAML>=3.12 in /usr/local/lib/python3.7/dist-packages (from cliff->optuna==2.7.0->mljar-supervised) (3.13)

Collecting cmd2>=1.0.0

Downloading https://files.pythonhosted.org/packages/15/8b/15061b32332bb35ea2a2f6263d0f616779d576e82739ec8e7fcf3c94abf5/cmd2-1.5.0-py3-none-any.whl (133kB)

|████████████████████████████████| 143kB 42.8MB/s

Collecting pbr!=2.1.0,>=2.0.0

Downloading https://files.pythonhosted.org/packages/18/e0/1d4702dd81121d04a477c272d47ee5b6bc970d1a0990b11befa275c55cf2/pbr-5.6.0-py2.py3-none-any.whl (111kB)

|████████████████████████████████| 112kB 39.5MB/s

Collecting Mako

Downloading https://files.pythonhosted.org/packages/f3/54/dbc07fbb20865d3b78fdb7cf7fa713e2cba4f87f71100074ef2dc9f9d1f7/Mako-1.1.4-py2.py3-none-any.whl (75kB)

|████████████████████████████████| 81kB 7.8MB/s

Collecting python-editor>=0.3

Downloading https://files.pythonhosted.org/packages/c6/d3/201fc3abe391bbae6606e6f1d598c15d367033332bd54352b12f35513717/python\_editor-1.0.4-py3-none-any.whl

Requirement already satisfied: typing-extensions>=3.6.4; python\_version < "3.8" in /usr/local/lib/python3.7/dist-packages (from importlib-metadata; python\_version < "3.8"->markdown->mljar-supervised) (3.7.4.3)

Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata; python\_version < "3.8"->markdown->mljar-supervised) (3.4.1)

Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (from PrettyTable>=0.7.2->cliff->optuna==2.7.0->mljar-supervised) (0.2.5)

Collecting colorama>=0.3.7

Downloading https://files.pythonhosted.org/packages/44/98/5b86278fbbf250d239ae0ecb724f8572af1c91f4a11edf4d36a206189440/colorama-0.4.4-py2.py3-none-any.whl

Collecting pyperclip>=1.6

Downloading https://files.pythonhosted.org/packages/a7/2c/4c64579f847bd5d539803c8b909e54ba087a79d01bb3aba433a95879a6c5/pyperclip-1.8.2.tar.gz

Requirement already satisfied: MarkupSafe>=0.9.2 in /usr/local/lib/python3.7/dist-packages (from Mako->alembic->optuna==2.7.0->mljar-supervised) (2.0.1)

Building wheels for collected packages: mljar-supervised, dtreeviz, shap, pyperclip

Building wheel for mljar-supervised (setup.py) ... done

Created wheel for mljar-supervised: filename=mljar\_supervised-0.10.4-cp37-none-any.whl size=143914 sha256=e4b15880939fde3e02dafe8c17a29258fcc43d604c109a8c177ca8f68e8c2aa9

Stored in directory: /root/.cache/pip/wheels/a3/ea/35/583dcb9528d9a561e490f431abea5a7882cfcd83831abc7832

Building wheel for dtreeviz (setup.py) ... done

Created wheel for dtreeviz: filename=dtreeviz-1.3-cp37-none-any.whl size=66642 sha256=84f154f36aeb6e18e6548f82499240b0937d3b0d9a1856e1d56f7a321fa6310b

Stored in directory: /root/.cache/pip/wheels/60/36/b1/188ee35c677e48463f6482d580f81c19f5f82ae5adbe293fd8

Building wheel for shap (setup.py) ... done

Created wheel for shap: filename=shap-0.36.0-cp37-cp37m-linux\_x86\_64.whl size=457596 sha256=172ec3c8be255424c92b264040c1d57df5b277ee1dbbc3f7543bc1070686863e

Stored in directory: /root/.cache/pip/wheels/fb/15/e1/8f61106790da27e0765aaa6e664550ca2c50ea339099e799f4

Building wheel for pyperclip (setup.py) ... done

Created wheel for pyperclip: filename=pyperclip-1.8.2-cp37-none-any.whl size=11107 sha256=3f4c021ae2dd23890509cc4a83316de366f0a153e65b209f336dc2ae0915541c

Stored in directory: /root/.cache/pip/wheels/25/af/b8/3407109267803f4015e1ee2ff23be0c8c19ce4008665931ee1

Successfully built mljar-supervised dtreeviz shap pyperclip

ERROR: tensorflow 2.2.0 has requirement scipy==1.4.1; python\_version >= "3", but you'll have scipy 1.6.1 which is incompatible.

ERROR: google-colab 1.0.0 has requirement pandas~=1.1.0; python\_version >= "3.0", but you'll have pandas 1.2.0 which is incompatible.

ERROR: datascience 0.10.6 has requirement folium==0.2.1, but you'll have folium 0.8.3 which is incompatible.

ERROR: albumentations 0.1.12 has requirement imgaug<0.2.7,>=0.2.5, but you'll have imgaug 0.2.9 which is incompatible.

Installing collected packages: numpy, pandas, scipy, threadpoolctl, scikit-learn, xgboost, lightgbm, catboost, tabulate, colour, dtreeviz, slicer, shap, wordcloud, category-encoders, cmaes, colorlog, pbr, stevedore, colorama, pyperclip, cmd2, cliff, Mako, python-editor, alembic, optuna, scikit-plot, mljar-supervised

Found existing installation: numpy 1.19.5

Uninstalling numpy-1.19.5:

Successfully uninstalled numpy-1.19.5

Found existing installation: pandas 1.1.5

Uninstalling pandas-1.1.5:

Successfully uninstalled pandas-1.1.5

Found existing installation: scipy 1.4.1

Uninstalling scipy-1.4.1:

Successfully uninstalled scipy-1.4.1

Found existing installation: scikit-learn 0.22.2.post1

Uninstalling scikit-learn-0.22.2.post1:

Successfully uninstalled scikit-learn-0.22.2.post1

Found existing installation: xgboost 0.90

Uninstalling xgboost-0.90:

Successfully uninstalled xgboost-0.90

Found existing installation: lightgbm 2.2.3

Uninstalling lightgbm-2.2.3:

Successfully uninstalled lightgbm-2.2.3

Found existing installation: tabulate 0.8.9

Uninstalling tabulate-0.8.9:

Successfully uninstalled tabulate-0.8.9

Found existing installation: wordcloud 1.5.0

Uninstalling wordcloud-1.5.0:

Successfully uninstalled wordcloud-1.5.0

Successfully installed Mako-1.1.4 alembic-1.6.4 catboost-0.24.4 category-encoders-2.2.2 cliff-3.7.0 cmaes-0.8.2 cmd2-1.5.0 colorama-0.4.4 colorlog-5.0.1 colour-0.1.5 dtreeviz-1.3 lightgbm-3.0.0 mljar-supervised-0.10.4 numpy-1.20.3 optuna-2.7.0 pandas-1.2.0 pbr-5.6.0 pyperclip-1.8.2 python-editor-1.0.4 scikit-learn-0.24.2 scikit-plot-0.3.7 scipy-1.6.1 shap-0.36.0 slicer-0.0.7 stevedore-3.3.0 tabulate-0.8.7 threadpoolctl-2.1.0 wordcloud-1.8.1 xgboost-1.3.3

Collecting delayed

Downloading https://files.pythonhosted.org/packages/7b/80/96302b67fe8d324af597748d5eef9cfb98bb1e6590b5f25a5b58b5e6f93f/delayed-0.11.0b1-py2.py3-none-any.whl

Collecting redis

Downloading https://files.pythonhosted.org/packages/a7/7c/24fb0511df653cf1a5d938d8f5d19802a88cef255706fdda242ff97e91b7/redis-3.5.3-py2.py3-none-any.whl (72kB)

|████████████████████████████████| 81kB 2.7MB/s

Collecting hiredis

Downloading https://files.pythonhosted.org/packages/ed/33/290cea35b09c80b4634773ad5572a8030a87b5d39736719f698f521d2a13/hiredis-2.0.0-cp37-cp37m-manylinux2010\_x86\_64.whl (85kB)

|████████████████████████████████| 92kB 6.7MB/s

Installing collected packages: redis, hiredis, delayed

Successfully installed delayed-0.11.0b1 hiredis-2.0.0 redis-3.5.3

In [ ]:

**from** **supervised.automl** **import** AutoML

In [ ]:

train = data

train.head()

Out[ ]:

In [ ]:

automl = AutoML()

In [ ]:

automl.fit(train[train.columns[2:-3]], train['G3'])

AutoML directory: AutoML\_1

The task is multiclass\_classification with evaluation metric logloss

AutoML will use algorithms: ['Baseline', 'Linear', 'Decision Tree', 'Random Forest', 'Xgboost', 'Neural Network']

AutoML will ensemble availabe models

AutoML steps: ['simple\_algorithms', 'default\_algorithms', 'ensemble']

\* Step simple\_algorithms will try to check up to 3 models

1\_Baseline logloss 2.83078 trained in 0.67 seconds

2\_DecisionTree logloss 2.680544 trained in 39.07 seconds

3\_Linear logloss 2.496762 trained in 37.59 seconds

\* Step default\_algorithms will try to check up to 3 models

4\_Default\_Xgboost logloss 2.128942 trained in 53.48 seconds

5\_Default\_NeuralNetwork logloss 2.774731 trained in 3.33 seconds

6\_Default\_RandomForest logloss 2.319354 trained in 49.11 seconds

\* Step ensemble will try to check up to 1 model

Ensemble logloss 2.021816 trained in 0.66 seconds

/usr/local/lib/python3.7/dist-packages/scipy/stats/stats.py:4264: SpearmanRConstantInputWarning:

An input array is constant; the correlation coefficent is not defined.

AutoML fit time: 206.27 seconds

AutoML best model: Ensemble

Out[ ]:

AutoML()