**Московский государственный технический**

**университет им. Н. Э. Баумана**

**Факультет «Информатика и системы управления»**

Кафедра «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №1

Разведочный анализ данных. Исследование и визуализация данных

Группа: ИУ5-62Б

Студент: Селедкина А.С.

Преподаватель: Гапанюк Ю.Е.

Москва, 2020 г.

**Цель лабораторной работы:** изучение различных методов визуализация данных.

**Описание задания**

Выбрать набор данных (датасет).

Создать ноутбук, который содержит следующие разделы:

1. Текстовое описание выбранного набора данных.

2. Основные характеристики датасета.

3. Визуальное исследование датасета.

4. Информация о корреляции признаков.

**Текст программы и примеры выполнения**

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

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

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

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

%matplotlib inline

sns.set(style="ticks")

**from** sklearn.datasets **import** \*

sklearn\_data = load\_breast\_cancer()

# Возможные значения целевого признака

sklearn\_data['target\_names']



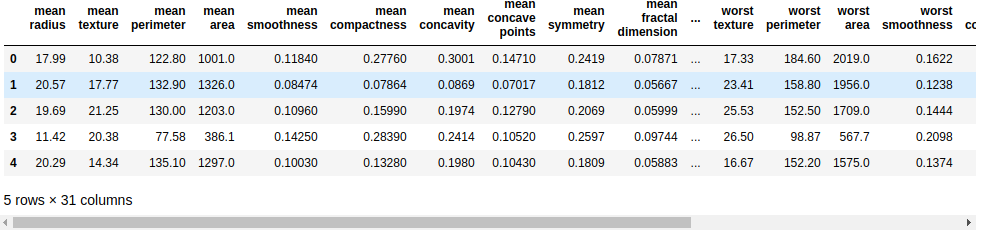
data = pd.DataFrame(data=np.c\_[sklearn\_data['data'],

sklearn\_data['target']],

columns=np.append(sklearn\_data['feature\_names'], ['target']))

# Первые 5 строк датасета

data.head()



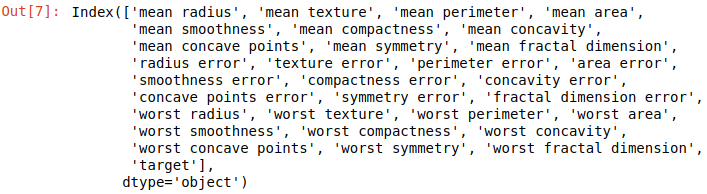
# Размер датасета - 569 строк, 31 колонка

data.shape



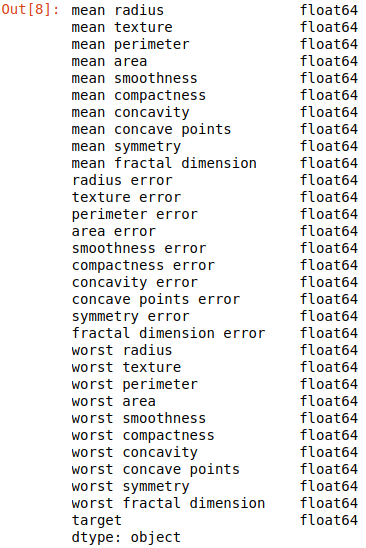
# Список колонок

data.columns



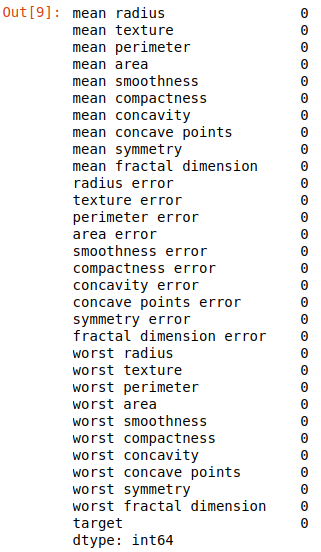
# Список колонок с типами данных

data.dtypes



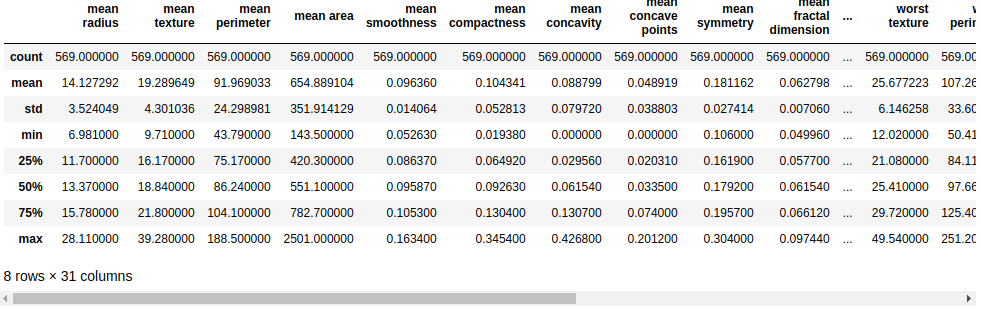
# Наличие пустых значений

data.isnull().sum()



# Основные статистические характеристики набора

data.describe()



# Уникальные значения для целевого признака

data['target'].unique()

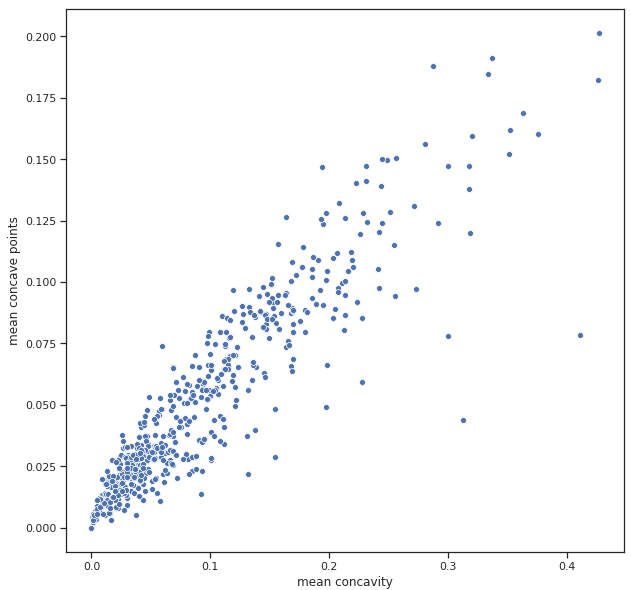


# Зависимость количества вогнутых частей контура от вогнутости

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

sns.scatterplot(ax=ax, x='mean concavity', y='mean concave points',

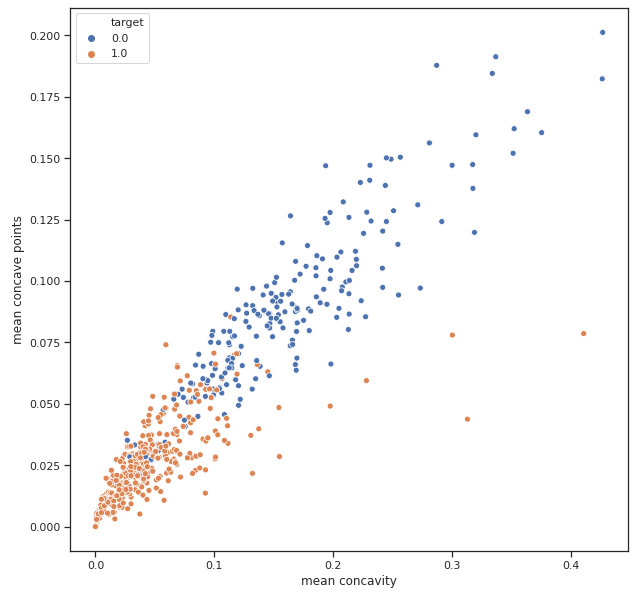
data=data)



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

sns.scatterplot(ax=ax, x='mean concavity', y='mean concave points',

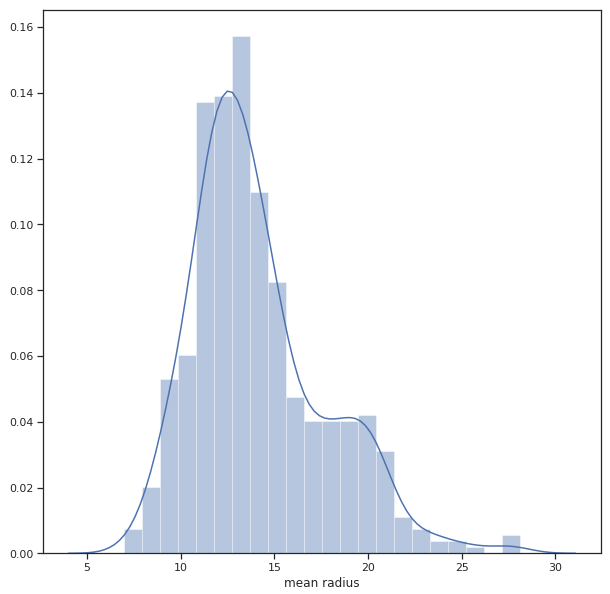
data=data, hue='target')



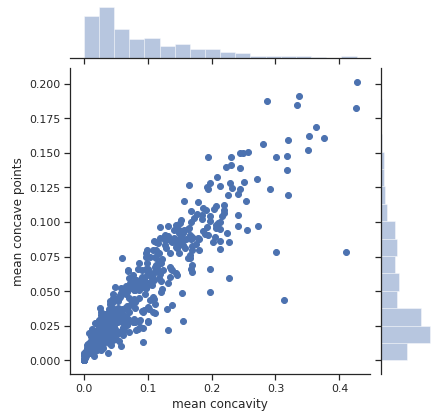
# Распределение среднего значения радиуса

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

sns.distplot(data['mean radius'])

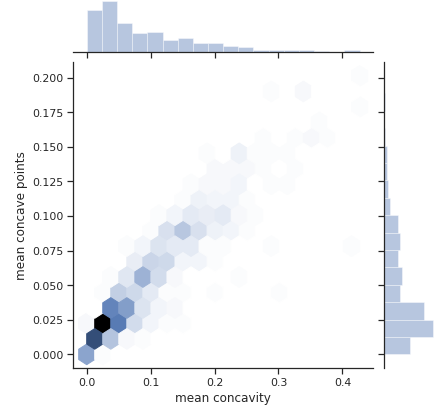


sns.jointplot(x='mean concavity', y='mean concave points', data=data)



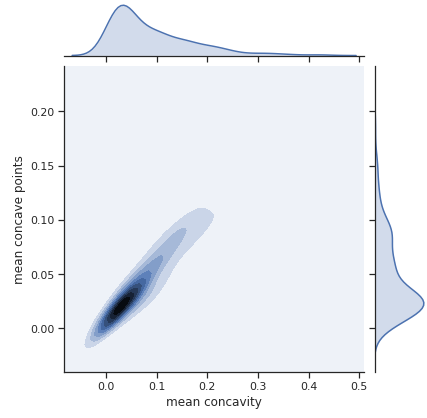
sns.jointplot(x='mean concavity', y='mean concave points', data=data,

kind='hex')



sns.jointplot(x='mean concavity', y='mean concave points', data=data,

kind='kde')



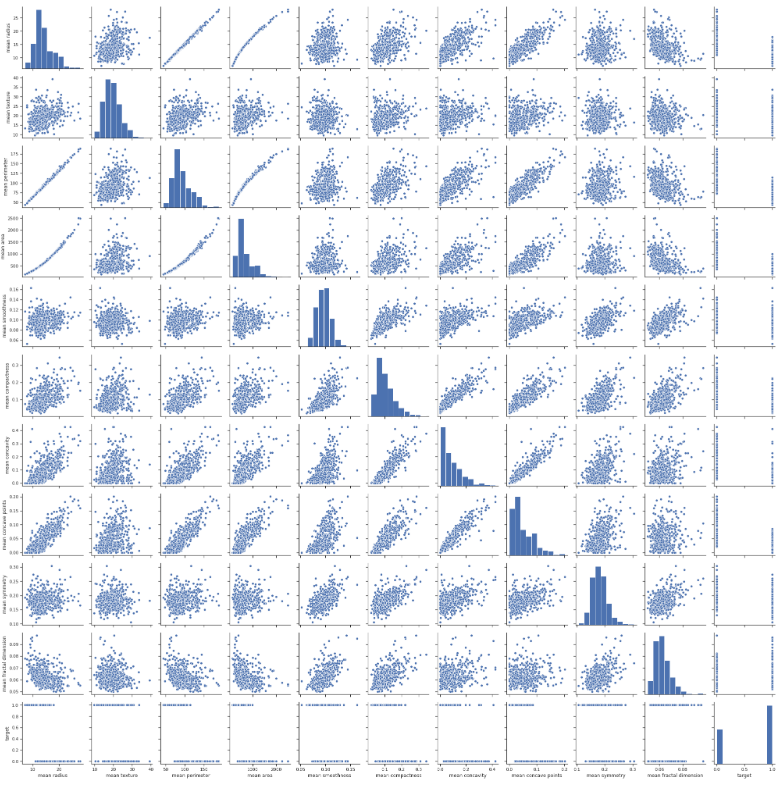
data1 = data[['mean radius', 'mean texture', 'mean perimeter', 'mean

area', 'mean smoothness', 'mean compactness', 'mean concavity',

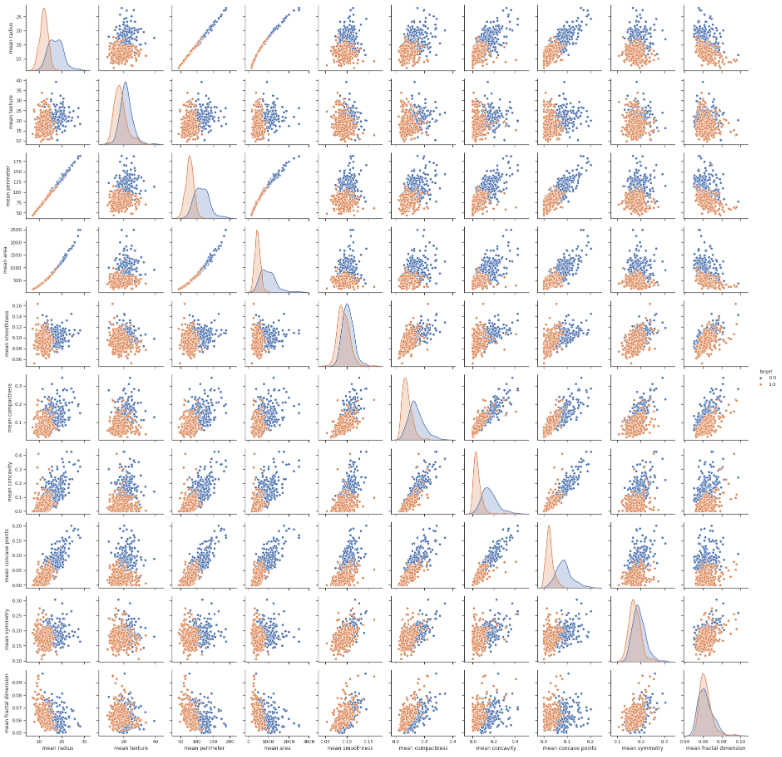
'mean concave points', 'mean symmetry', 'mean fractal dimension',

'target']]

sns.pairplot(data1)

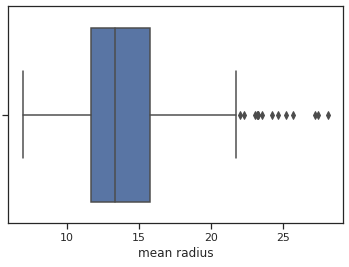


sns.pairplot(data1, hue='target')



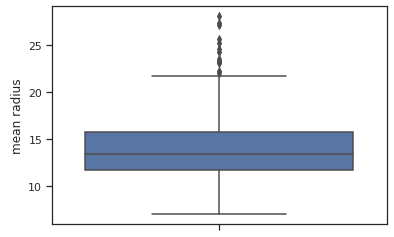
# По горизонтали

sns.boxplot(x=data['mean radius'])



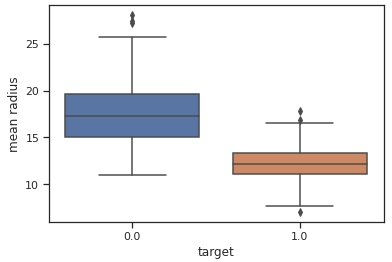
# По вертикали

sns.boxplot(y=data['mean radius'])

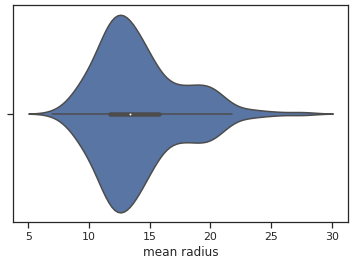


# Распределение параметра mean radius, сгруппированное по target

sns.boxplot(x=data['target'], y=data['mean radius'])

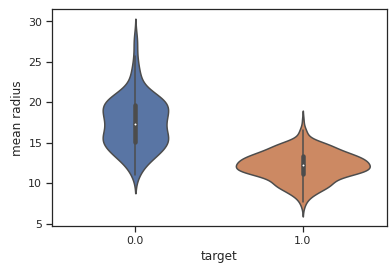


sns.violinplot(x=data['mean radius'])

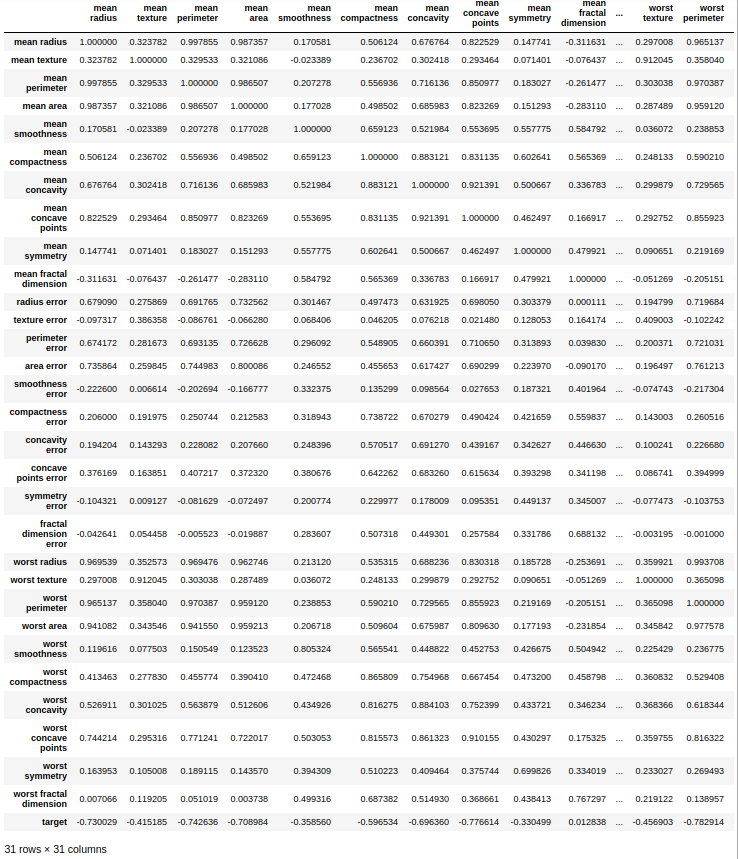


# Распределение параметра mean radius сгруппированные по target.

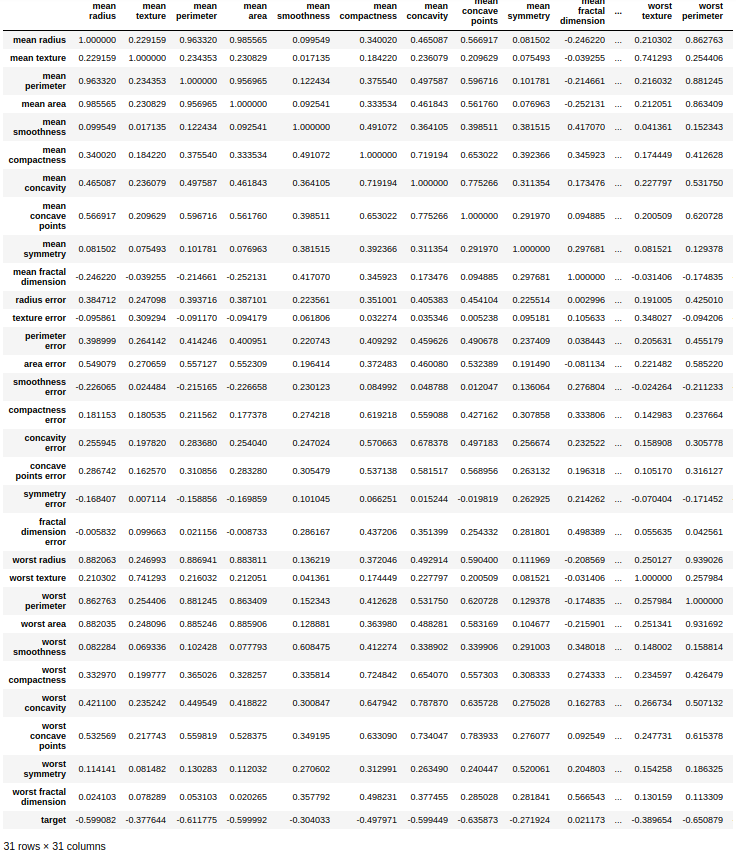
sns.violinplot(x='target', y='mean radius', data=data)



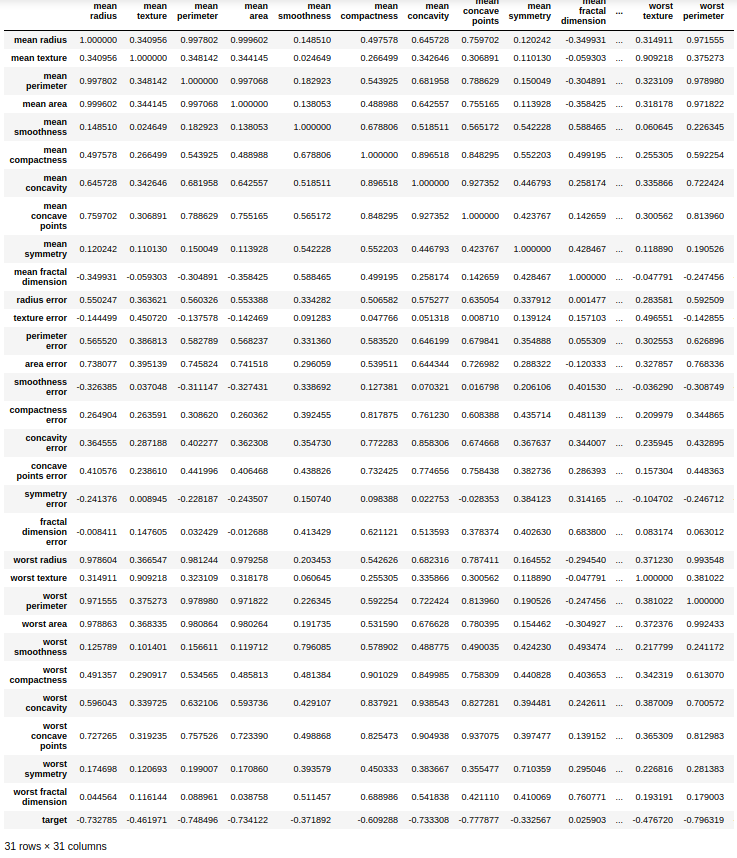
data.corr(method='pearson')



data.corr(method='kendall')



data.corr(method='spearman')



sns.heatmap(data.corr())

