**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 рубежному контролю №1

Технологии разведочного анализа и обработки данных

Группа: ИУ5-62Б

Студент: Селедкина А.С.

Преподаватель: Гапанюк Ю.Е.

Москва, 2020 г.

**Вариант 15**

**Описание задания**

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему? Для произвольной колонки данных построить гистограмму.

**Текст программы и примеры выполнения**

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

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

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

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

%matplotlib inline

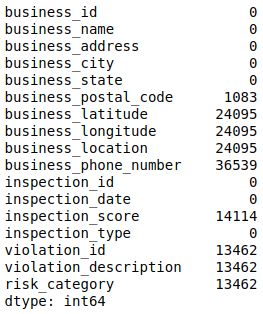
sns.set(style="ticks")

data = pd.read\_csv('data/restaurant-scores-lives-standard.csv', ',')

data.shape



data.isnull().sum()



data.head()



# Выберем числовые колонки с пропущенными значениями

rows\_num = data.shape[0]

for col in data.columns:

null\_count = data[data[col].isnull()].shape[0]

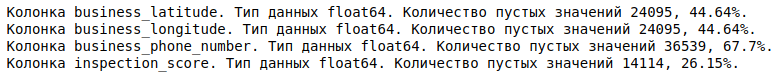
col\_type = str(data[col].dtype)

if null\_count > 0 and (col\_type == 'float64' or col\_type == 'int64'):

percent = round(null\_count / rows\_num \* 100.0, 2)

print('Колонка {}. Тип данных {}. Количество пустых значений {},

{}%.'.format(col, col\_type, null\_count, percent))

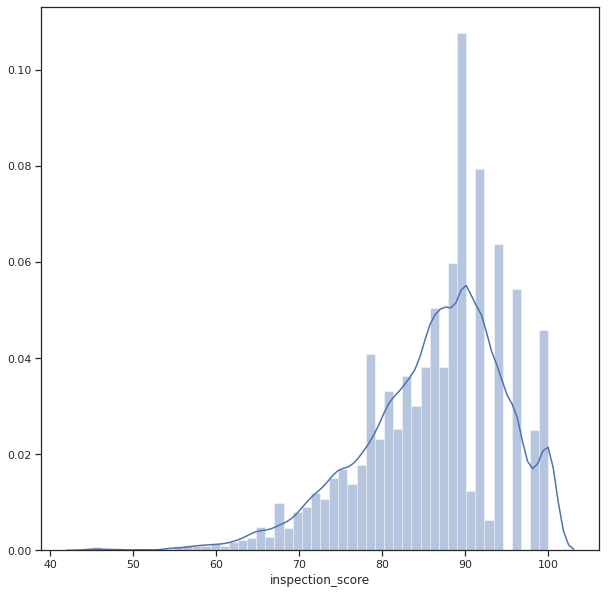


score\_col = 'inspection\_score'

data\_score\_col = data[[score\_col]]

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

sns.distplot(data[score\_col])



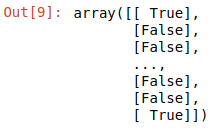
**from** sklearn.impute **import** SimpleImputer

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

indicator\_num = MissingIndicator()

mask\_missing\_values\_num = indicator\_num.fit\_transform(data\_score\_col)

mask\_missing\_values\_num



# Заполнение пропусков средним значением

imp\_num = SimpleImputer(strategy='mean')

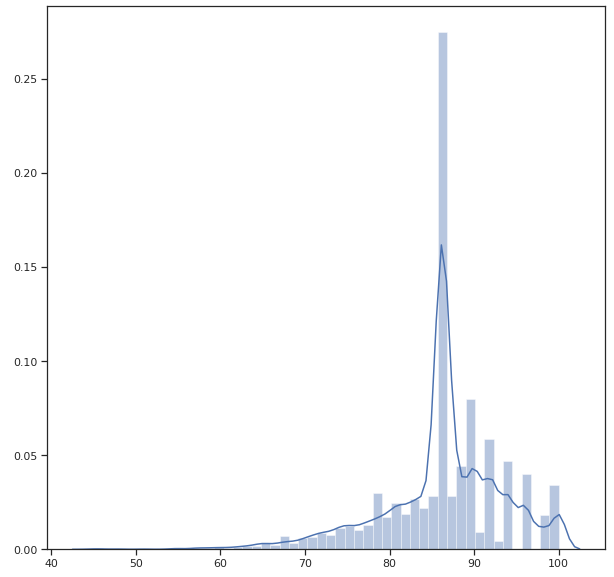
data\_num\_imp = imp\_num.fit\_transform(data\_score\_col)

data\_num\_imp[mask\_missing\_values\_num]



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

sns.distplot(data\_num\_imp)



Как видно из гистограммы, недостатком заполнения пропусков средним значением является искажение распределения.

# Выберем категориальные колонки с пропущенными значениями

for col in data.columns:

null\_count = data[data[col].isnull()].shape[0]

col\_type = str(data[col].dtype)

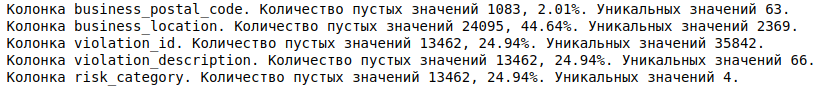
if null\_count > 0 and col\_type == 'object':

percent = round(null\_count / rows\_num \* 100.0, 2)

unique\_num = len(data[col].unique())

print('Колонка {}. Количество пустых значений {}, {}%. Уникальных значений

{}.'.format(col, null\_count, percent, unique\_num))



risk\_col = 'risk\_category'

data\_risk\_col = data[[risk\_col]]

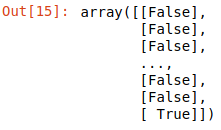
data[risk\_col].unique()



indicator\_cat = MissingIndicator()

mask\_missing\_values\_cat = indicator\_cat.fit\_transform(data\_risk\_col)

mask\_missing\_values\_cat



# Заполнение пропусков наиболее часто встречающимся значением

imp\_cat = SimpleImputer(strategy='most\_frequent')

data\_cat\_imp = imp\_cat.fit\_transform(data\_risk\_col)

data\_cat\_imp[mask\_missing\_values\_cat]



np.unique(data\_cat\_imp)



При заполнении пропусков категориального признака модой проявляются те же недостатки, что и при заполнении пропусков количественного признака средним значением. Если пропусков немного, данные методы не приведут к значительным искажениям результатов. Однако для более точных результатов целесообразно использовать более сложные методы, такие как линейная регрессия.