**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №2

«Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-62Б |  | преподаватель каф. ИУ5 |
| Щепетов Дмитрий |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

2024 г.

**Описание задания**

1. Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.)
2. Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:
   * обработку пропусков в данных;
   * кодирование категориальных признаков;
   * масштабирование данных.

**Текст программы**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

# from plotly.express import line

#import df

#import plotly.express as px

%matplotlib inline

sns.set(style="ticks")

plt.style.use('ggplot')

pd.set\_option('display.max\_columns',24)

pd.set\_option('display.max\_colwidth',None)

pd.set\_option('display.float\_format', lambda x: '%.4f' % x)

encoding = 'windows-1250'

data=pd.read\_csv(r'C:\Users\Dima\Desktop\Homework\3 курс\ТМО\ЛР2\archive\general\_government\_debt.csv', encoding = encoding)

data.shape

data.info()

input\_df = data

# input\_df.drop(columns=['indicator\_name'], inplace=True)

result\_df = data.T

result\_df.columns = result\_df.iloc[0]

result\_df = result\_df.iloc[1:]

result\_df = result\_df.tail(len(result\_df)-1)

# result\_df.columns = result\_df.columns.str.replace('ď»ż', '')

result\_df.head()

# проверим есть ли пропущенные значения

data.isnull().sum()

total\_count = data.shape[0]

print('Всего строк: {}'.format(total\_count))

# Заполнение всех пропущенных значений нулями

# В данном случае это некорректно, так как нулями заполняются в том числе категориальные колонки

data\_new\_3 = data.fillna(0)

data\_new\_3.head()

# Выберем числовые колонки с пропущенными значениями

# Цикл по колонкам датасета

num\_cols = []

for col in data.columns:

# Количество пустых значений

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

dt = str(data[col].dtype)

if temp\_null\_count>0 and (dt=='float64' or dt=='int64'):

num\_cols.append(col)

temp\_perc = round((temp\_null\_count / total\_count) \* 100.0, 2)

print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(col, dt, temp\_null\_count, temp\_perc))

# Фильтр по колонкам с пропущенными значениями

data\_num = data[num\_cols]

data\_num

# Гистограмма по признакам

for col in data\_num:

plt.hist(data[col], 50)

plt.xlabel(col)

plt.show()

result\_df.reset\_index(inplace=True)

result\_df.rename(columns={'index': 'Year'}, inplace=True)

# Создание графика

plt.figure(figsize=(10, 6))

for column in result\_df.columns[1:]:

plt.plot(result\_df['Year'], result\_df[column], label=column)

plt.xlabel('Year')

plt.ylabel('Value')

plt.title('Central Government Debt Over Time')

plt.legend()

plt.show()

mean\_values = result\_df.drop(columns=['Year']).mean(axis=1)

# Создание DataFrame из средних значений

mean\_df = mean\_values.to\_frame().reset\_index()

mean\_df.rename(columns={'index': 'Year', 0: 'Mean Value'}, inplace=True)

# Создание графика

plt.figure(figsize=(10, 6))

plt.plot(mean\_df['Year'], mean\_df['Mean Value'])

plt.xlabel('Year')

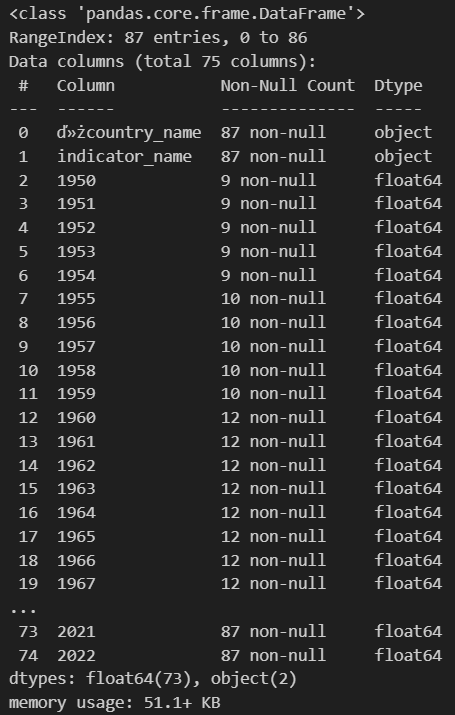
plt.ylabel('Mean Value')

plt.title('Mean Central Government Debt Over Time')

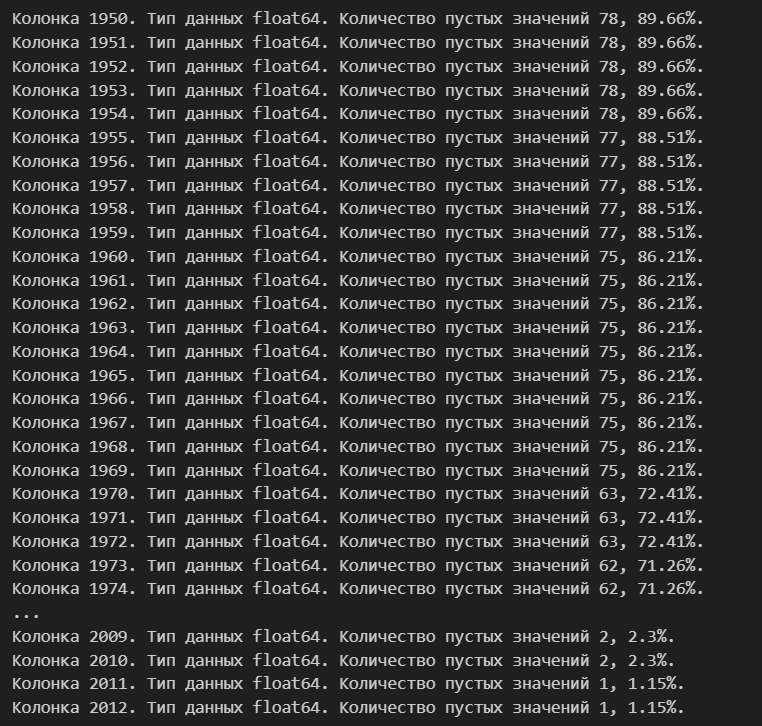
plt.show()

**Экранные формы с примерами выполнения программы**

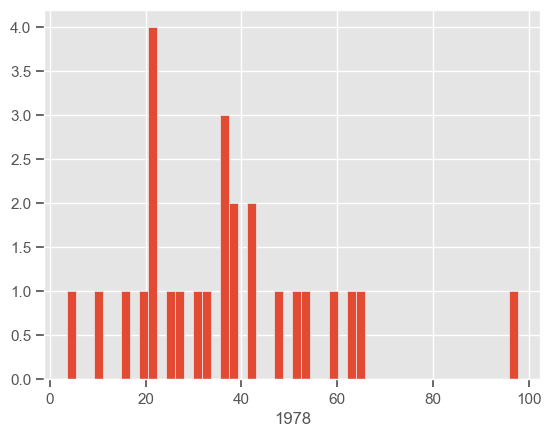
Просмотр информации о датасете



Анализ пустых значений



Гистограммы по признакам



Долг Центрального правительства с течением времени



Средний долг Центрального Правительства С течением времени

