**Лабораторная работа №1**

**Изучение библиотеки Pandas**

**Установка необходимых библиотек и инструментов**

Numpy - это библиотека Python для вычислительно эффективных операций с многомерными массивами, предназначенная в основном для научных вычислений.

*import numpy as np*

Pandas - это библиотека Python, предоставляющая широкие возможности для анализа данных. С ее помощью очень удобно загружать, обрабатывать и анализировать табличные данные с помощью SQL-подобных запросов.

*import pandas as pd*

Основными структурами данных в Pandas являются классы Series и DataFrame. Первый из них представляет собой одномерный индексированный массив данных некоторого фиксированного типа. Второй - это двумерная структура данных, представляющая собой таблицу, каждый столбец которой содержит данные одного типа. Можно представлять её как словарь объектов типа Series.

**Выполнение лабораторной работы**

1. Прочтите данные из файла data.csv

*Функции, которые могут пригодиться при решении: pd.read\_csv(..., delimiter=',')*

1. Выведите описание прочтенных данных.

*Функции, которые могут пригодиться при решении: .describe()*

1. Отобразите несколько первых и несколько последних записей.

*Функции, которые могут пригодиться при решении: .head(), .tail()*

*Какие параметры можно передать этим функциям?*

1. Прочтите в файле DataDictionary-ru.txt, что означают столбцы матрицы. Какому типу принадлежит каждый столбец (вещественный, целый, категориальный)?
2. Заметьте, что столбец DebtRatio содержит неправдоподобные данные. Только значения, соответствующие известному месячному доходу, являются отношениями. Остальные – абсолютные значения месячных выплат процентов. Исправьте данные, сделав все значения столбца DebtRatio абсолютными (умножьте их на MonthlyIncome). Чтобы ваша программа быстро работала на полных данных, постарайтесь не использовать цикл.

***Функции, которые могут пригодиться при решении:***

Обращение к элементам DataFrame:

* элемент: data.loc[i, 'названиеСтолбца']
* столбец: data['названиеСтолбца']
* подматрица: data.loc[a:b, списокНазванийСтолбцов]

Условная индексация:

* data.loc[data['столбец'] > 20, списокНазванийСтолбцов]

лучше писать так:

* i = data['столбец'] > 20 # вектор True и False
* data.loc[i, 'названиеСтолбца']

У подматриц номера строк наследуются от исходной.

* pandas.isnull(скаляр или массив) - проверка, является ли значение неопределенным (NaN)
* pandas.notnull(скаляр или массив) - проверка, является ли значение определенным (не NaN)

1. Поменяйте имя столбца на Debt. Функции, которые могут пригодиться при решении:

.rename(columns={'староеИмя':'новоеИмя'}, inplace=True)

1. Вычислите средний ежемесячный доход и присвойте всем клиентам с неизвестным доходом полученное число.

*Функции, которые могут пригодиться при решении: .mean()*

*Другие описательные статистики:* <https://pandas.pydata.org/pandas-docs/stable/reference/frame.html#computations-descriptive-stats>

1. Используя метод groupby, оцените вероятности невозврата кредита (SeriousDlqin2yrs=1) для различных значений количества иждивенцев (NumberOfDependents).

Проделайте аналогичную процедуру для различных значений столбца NumberRealEstateLoansOrLines

*Подсказка:* data['столбец1'].groupby(data['столбец2']).mean() *-- расчет средних значений столбца1 по группам из столбца2.*