МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ

ТЕХНОЛОГИЙ И ТЕЛЕКОММУНИКАЦИЙ

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

**Лабораторная работа №9**

по дисциплине

«Искусственный интеллект и машинное обучение»

**Выполнил:**

Сердюков Никита Анатольевич

Студент 2 курса группы \_ПИН-б-о-22-1

Направления подготовки

09.03.03 Прикладная информатика

очной формы обучения

Ставрополь, 2023 г.

Тема: ИСПОЛЬЗОВАНИЕ РАЗРАБОТАННОГО ПАЙПЛАЙНА ДЛЯ МНОГОМЕРНОЙ РЕГРЕССИИ

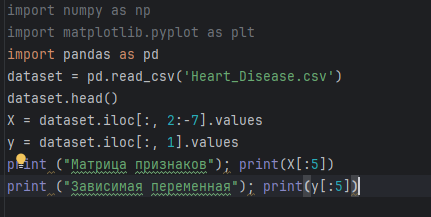
Цель работы: научиться применять разработанный пайплайн для тиражирования кода с целью решения широкого круга задач машинного обучения

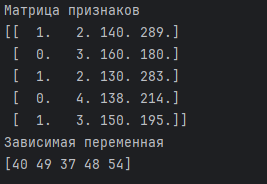
Выполнение работы:

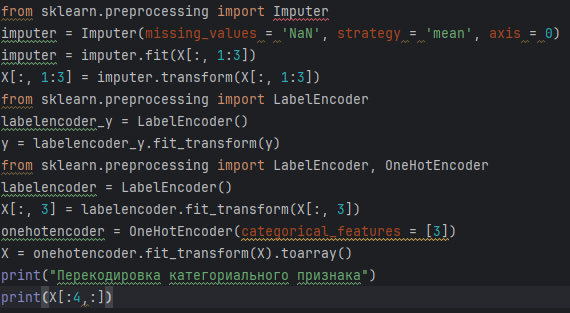
Датасет: Набор данных о сердечных заболеваниях

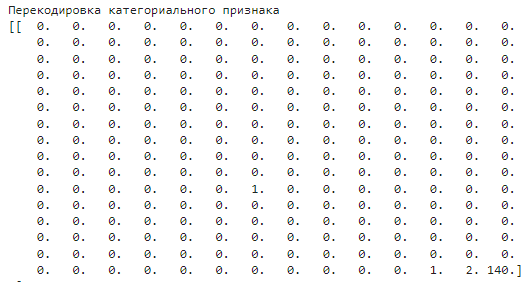
Описание датасета: Этот набор данных содержит медицинскую и поведенческую информацию, используемую для прогнозирования риска сердечно-сосудистых заболеваний. Он очень полезен для исследований и машинного обучения в области здравоохранения.

1. Загрузка данных





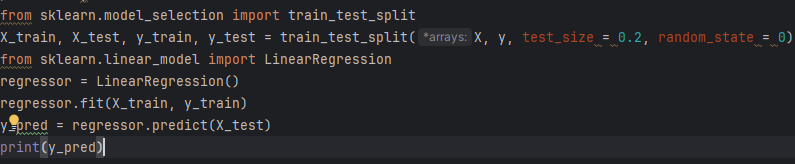
1. 

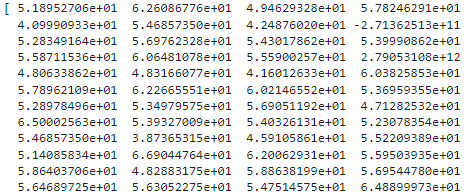


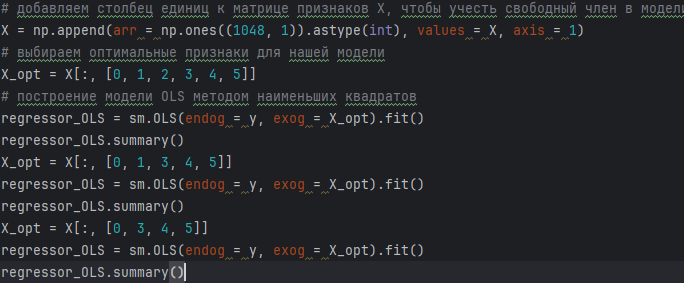
1. Разделение выборки на тестовую и тренировочную

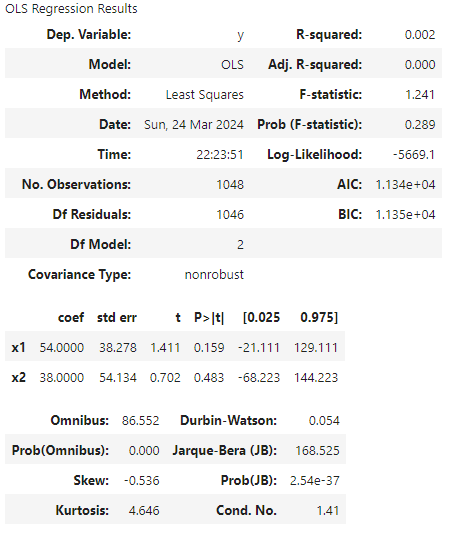


1. Предсказиние результатов.





1. 



Листинг:

import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
dataset = pd.read\_csv('Heart\_Disease.csv')  
dataset.head()  
X = dataset.iloc[:, 2:-7].values  
y = dataset.iloc[:, 1].values  
print ("Матрица признаков"); print(X[:5])  
print ("Зависимая переменная"); print(y[:5])  
from sklearn.preprocessing import Imputer  
imputer = Imputer(missing\_values = 'NaN', strategy = 'mean', axis = 0)  
imputer = imputer.fit(X[:, 1:3])  
X[:, 1:3] = imputer.transform(X[:, 1:3])  
from sklearn.preprocessing import LabelEncoder  
labelencoder\_y = LabelEncoder()  
y = labelencoder\_y.fit\_transform(y)  
from sklearn.preprocessing import LabelEncoder, OneHotEncoder  
labelencoder = LabelEncoder()  
X[:, 3] = labelencoder.fit\_transform(X[:, 3])  
onehotencoder = OneHotEncoder(categorical\_features = [3])  
X = onehotencoder.fit\_transform(X).toarray()  
print("Перекодировка категориального признака")  
print(X[:4,:])  
X = X[:, 1:]  
print(X[:4,:])  
from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)  
from sklearn.linear\_model import LinearRegression  
regressor = LinearRegression()  
regressor.fit(X\_train, y\_train)  
y\_pred = regressor.predict(X\_test)  
print(y\_pred)  
import statsmodels.formula.api as sm  
# добавляем столбец единиц к матрице признаков X, чтобы учесть свободный член в модели  
X = np.append(arr = np.ones((1048, 1)).astype(int), values = X, axis = 1)  
# выбираем оптимальные признаки для нашей модели  
X\_opt = X[:, [0, 1, 2, 3, 4, 5]]  
# построение модели OLS методом наименьших квадратов  
regressor\_OLS = sm.OLS(endog = y, exog = X\_opt).fit()  
regressor\_OLS.summary()  
X\_opt = X[:, [0, 1, 3, 4, 5]]  
regressor\_OLS = sm.OLS(endog = y, exog = X\_opt).fit()  
regressor\_OLS.summary()  
X\_opt = X[:, [0, 3, 4, 5]]  
regressor\_OLS = sm.OLS(endog = y, exog = X\_opt).fit()  
regressor\_OLS.summary()

Вывод: Многомерная регрессия, также известная как многомерная линейная регрессия, используется для моделирования влияния нескольких независимых переменных на зависимую переменную. Пайплайн для многомерной регрессии может включать в себя шаги по обработке данных, исключению мультиколлинеарности, выбору оптимальной модели и интерпретации результатов.

Пайплайн для многомерной регрессии помогает в подготовке данных, выборе оптимальных признаков, построении модели и оценке ее качества. Это позволяет исследователям и аналитикам более точно моделировать сложные взаимосвязи между несколькими независимыми переменными и зависимой переменной. Пайплайн также способствует систематизации процесса анализа данных и повышению воспроизводимости результатов исследования.