Министерство науки и высшего образованияРоссийской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф.Уткина»

Кафедра «Электронные вычислительные машины»

Отчет по практическому занятию №3

на тему

«Линейная регрессия»

по дисциплине

«**Машинное обучение**»

Выполнили:

Студенты группы №140

Бригада 7

Сафаров Д. А.

Тимохин Е. М.

Проверила:

ас. Панина И.С.

**Цель работы**

Изучить модель линейной регрессии

**Задание**

Выполнить пример, представленный в работе. Изменить значения аргументов функции datasets.make\_regression, сделать выводы по выполненной работе.

**Практическая часть**

from tkinter import \*  
import matplotlib.pyplot as plt  
from sklearn import datasets, linear\_model  
  
window = Tk()  
window.title('Построение модели линейной регрессии')  
window.geometry('400x400')  
frame = Frame(window, padx=15, pady=15)  
frame.pack(expand=True)  
lb1 = Label(frame, text='Введите количество точек:')  
lb1.grid(row=3, column=1)  
tf1 = Entry(frame)  
tf1.grid(row=3, column=2)  
lb2 = Label(frame, text='Введите разброс точек:')  
lb2.grid(row=4, column=1)  
tf2 = Entry(frame)  
tf2.grid(row=4, column=2)  
n\_features = 1

# Функция построения модели

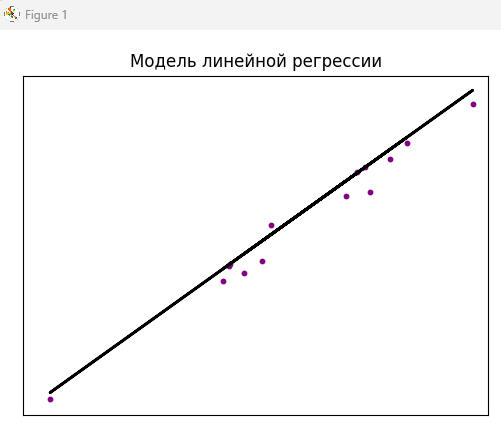
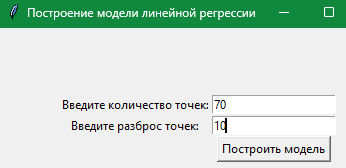
def calc():  
 k\_noise = int(tf2.get())  
 k\_samples = int(tf1.get())  
 X, y = datasets.make\_regression(n\_samples=k\_samples, n\_features=1, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_noise=k\_noise)  
 # Обучающие и тестовые данные  
 train\_size = int(k\_samples \* 0.8)  
 test\_size = int(k\_samples \* 0.2)  
 # Нарезаем входные данные на обучающие и тестовые  
 X\_train = X[:-train\_size]  
 X\_test = X[-test\_size:]  
 # Указываем, сколько будет выходных обучающих и тестовых данных  
 y\_train = y[:-train\_size]  
 y\_test = y[-test\_size:]  
 # Создали регрессионную модель  
 regr = linear\_model.LinearRegression()  
 # Обучение модели  
 regr.fit(X\_train, y\_train)  
 # Получение значений y  
 y\_pred = regr.predict(X\_test)  
 # Вывод картинки  
 plt.scatter(X\_test, y\_test, color='purple', s=10)  
 plt.plot(X\_test, y\_pred, color='black', linewidth=2)  
 plt.title('Модель линейной регрессии')  
 plt.xticks(())  
 plt.yticks(())  
 plt.savefig('model.png')  
 plt.show()  
  
btn = Button(frame, text="Построить модель", command=calc)  
btn.grid(row=5, column=2)  
window.mainloop()

Были изменены значения аргументов функцииdatasets.make и выявлены следующие закономерности:

\_samples: Этот параметр контролирует количество образцов в наборе данных. Увеличение или уменьшение этого значения изменит размер набора данных.

n\_features: Этот параметр контролирует количество признаков (независимых переменных) в каждом образце. Если увеличить или уменьшить это значение, изменится размерность входных данных.

noise: Параметр noise контролирует количество добавляемого шума к зависимой переменной. Увеличение шума сделает отношение между независимыми и зависимыми переменными менее явным.

На рисунках 1 и 2 представлены результаты работы программы при разных параметрах.

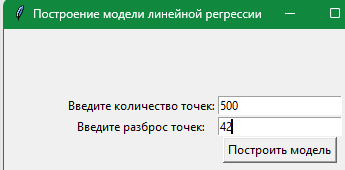
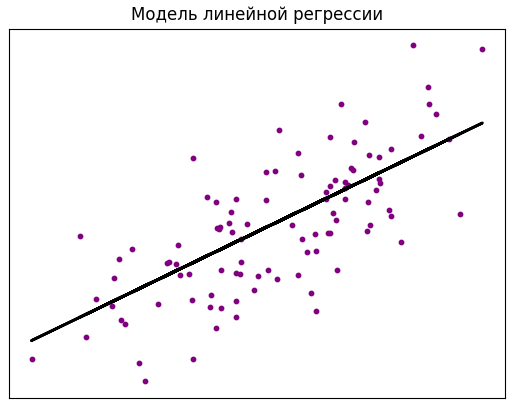
Рисунок 1 – Первый пример работы программы

Рисунок 2 – Второй пример работы программы