МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ

ТЕХНОЛОГИЙ И ТЕЛЕКОММУНИКАЦИЙ

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

**Лабораторная работа №7**

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.

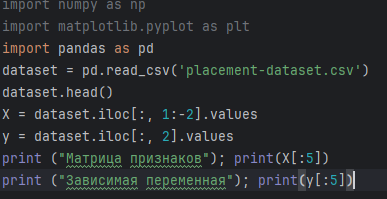
Тема: Построение пайплайна одномерной регрессии.

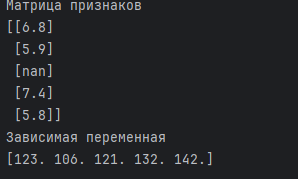
Цель работы: разработка единого пайплайна для решения задачи регрессии.

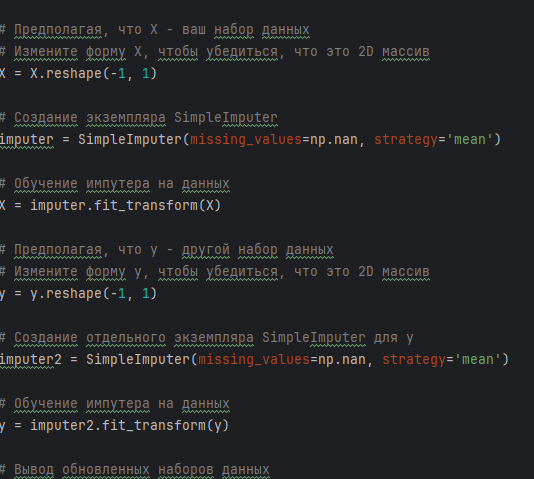
Выполнение работы:

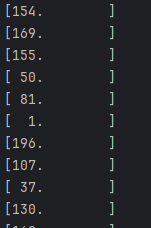
Датасет: college placement

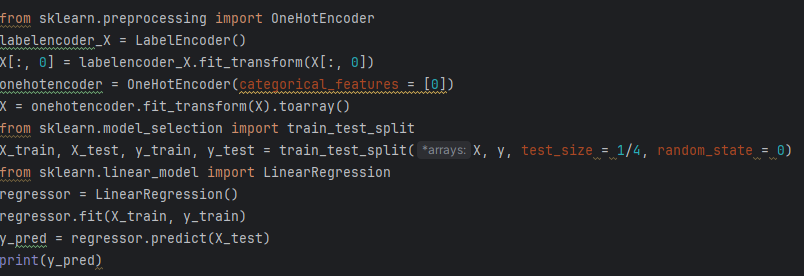
Описание датасета: Этот набор данных содержит записи об атрибутах учащихся и их соответствующем статусе зачисления. Он включает в себя такую информацию, как город проживания или местонахождение учебного заведения, совокупный средний балл успеваемости (CGPA), показатели коэффициента интеллекта (IQ) и двоичный показатель, указывающий, получил ли студент работу или стажировку. Набор данных представляет собой комбинацию числовых и категориальных данных, при этом в некоторых записях присутствуют экземпляры пропущенных значений.

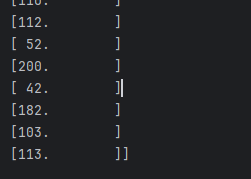
1. Разделение на матрицу признаков и зависимую переменную

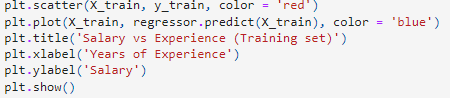


1. Обработка пропущенных значений (если требуется)

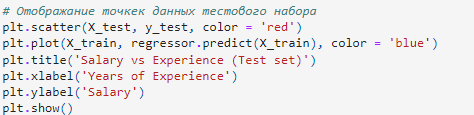


1. Предсказание, обработка и визуализация результатов











Листинг:

import numpy as np  
import matplotlib.pyplot as plt  
from sklearn.preprocessing import LabelEncoder  
from sklearn.preprocessing import OneHotEncoder  
import pandas as pd  
dataset = pd.read\_csv('placement-dataset.csv')  
dataset.head()  
X = dataset.iloc[:, 1:-2].values  
y = dataset.iloc[:, 2].values  
print ("Матрица признаков"); print(X[:5])  
print ("Зависимая переменная"); print(y[:5])  
from sklearn.impute import SimpleImputer  
import numpy as np  
  
# Предполагая, что X - ваш набор данных  
# Измените форму X, чтобы убедиться, что это 2D массив  
X = X.reshape(-1, 1)  
  
# Создание экземпляра SimpleImputer  
imputer = SimpleImputer(missing\_values=np.nan, strategy='mean')  
  
# Обучение импутера на данных  
X = imputer.fit\_transform(X)  
  
# Предполагая, что y - другой набор данных  
# Измените форму y, чтобы убедиться, что это 2D массив  
y = y.reshape(-1, 1)  
  
# Создание отдельного экземпляра SimpleImputer для y  
imputer2 = SimpleImputer(missing\_values=np.nan, strategy='mean')  
  
# Обучение импутера на данных  
y = imputer2.fit\_transform(y)  
  
# Вывод обновленных наборов данных  
print(X)  
print(y)  
# Сортировка массива X  
X = np.sort(X, axis=0)  
  
# Сортировка массива y  
y = np.sort(y, axis=0)  
  
labelencoder\_y = LabelEncoder()  
y = labelencoder\_y.fit\_transform(y)  
  
from sklearn.preprocessing import OneHotEncoder  
labelencoder\_X = LabelEncoder()  
X[:, 0] = labelencoder\_X.fit\_transform(X[:, 0])  
onehotencoder = OneHotEncoder(categorical\_features = [0])  
X = onehotencoder.fit\_transform(X).toarray()  
from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 1/4, 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)  
# Отображание точкек данных обучающего набора  
plt.scatter(X\_train, y\_train, color = 'red')  
plt.plot(X\_train, regressor.predict(X\_train), color = 'blue')  
plt.title('Salary vs Experience (Training set)')  
plt.xlabel('Years of Experience')  
plt.ylabel('Salary')  
plt.show()  
# Отображание точкек данных тестового набора  
plt.scatter(X\_test, y\_test, color = 'red')  
plt.plot(X\_train, regressor.predict(X\_train), color = 'blue')  
plt.title('Salary vs Experience (Test set)')  
plt.xlabel('Years of Experience')  
plt.ylabel('Salary')  
plt.show()

Вывод: Модель линейной регрессии - это статистическая модель, используемая для оценки отношения между зависимой переменной и одной или несколькими независимыми переменными. В основе линейной регрессии лежит предположение о линейной зависимости между переменными. Модель пытается найти линейную функцию, которая наилучшим образом соответствует данным путем минимизации суммы квадратов разностей между фактическими и предсказанными значениями.

Линейная регрессия широко используется для прогнозирования и моделирования в различных областях, таких как экономика, финансы, социология, биология и многие другие. В машинном обучении линейная регрессия также применяется для решения задачи предсказания значений на основе входных признаков.