Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

Федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет ИТР

Кафедра ПИн

ЛАБОРАТОРНАЯ

РАБОТА №6

По Анализ данных

Тема Линейный регрессионный анализ на Python. Построение

моделей для задачи регрессии. Линейная регрессия. Регуляризация

Руководитель

Белякова А.C.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИН - 121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

Лабораторная работа №6

Тема: линейный регрессионный анализ на Python. Построение моделей для задачи регрессии. Линейная регрессия. Регуляризация.

Цели и задачи: изучение способов построения регрессионных моделей.

Ход работы: провести анализ набора данных о заработной плате в области науки о данных, получить следующие данные:

Задание 1) Обучите модель гребневой регрессии Ridge. Выведите коэффициенты. Узнать зануляются ли какие-то? Вычислить MSE.

Листинг кода 1 – подключение библиотек и чтение данных из файла:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings("ignore")

plt.style.use("fivethirtyeight")

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.linear\_model import LinearRegression, Lasso, Ridge

from sklearn.ensemble import RandomForestRegressor

from sklearn.preprocessing import StandardScaler,MinMaxScaler

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

from sklearn.linear\_model import LassoCV, RidgeCV

df = pd.read\_csv("D:/ucheba/python/sales.csv")

Листинг кода 2 – Обучение модели и вывод коэффициентов:

linreg = LinearRegression(fit\_intercept = True)

X = df.drop(["ADV"],axis=1)

y = df["ADV"]

df.info()

X.shape, y.shape

X\_scal = StandardScaler().fit\_transform(X)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X\_scal, y,

test\_size = 0.3, random\_state=42)

linreg.fit(x\_train, y\_train)

y\_pred = linreg.predict(X\_scal)

y\_pred.shape

df["ADV"][33]

y\_pred[33]

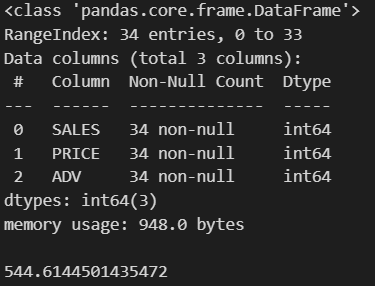


Рисунок 1 – Вывод коэффициентов

Оценка качества модели при помощи MSE:

Листинг кода 3 – Вычисление MSE:

np.sqrt(mean\_squared\_error(y\_test, linreg.predict(x\_test)))

X.columns, linreg.coef\_

linreg.intercept\_

pd.DataFrame(linreg.coef\_, X.columns, columns=["coef"]).sort\_values(

by="coef", ascending=False)

linreg.predict(X\_scal)[0], df["ADV"][0]



Рисунок 2 –Результат вычисления MSE

Задание 2) Найти оптимальное значение коэффициента для гребневой регрессии (L2-регуляризация).

Листинг кода 4 – Нахождение оптимального значения коэффициента для гребневой регрессии:

rf = RandomForestRegressor(n\_estimators=100, max\_depth = 5,

min\_samples\_leaf= 5, random\_state=42)

rf.fit(x\_train,y\_train)

rf\_pred = rf.predict(x\_test)

np.sqrt(mean\_squared\_error(y\_test, rf\_pred))

pd.DataFrame(rf.feature\_importances\_, X.columns,

columns=["rf\_coef"]).sort\_values(by="rf\_coef", ascending=False)



Рисунок 3 – Оптимальное значение коэффициента для гребневой регрессии

Задание 3) Загрузите набор данных sales.csv. Рассчитайте коэффициенты множественной регрессии для расчета объема продаж (SALES) по объему рекламы (ADV) и цены батончика (PRICE). При помощи рассчитанных коэффициентов найдите объем продаж в магазине с рекламой 400 долларов в месяц и ценой батончика 79 центов. Деление на тренировочную и тестовую выборку делать не нужно. Масштабирование делать не нужно.

Листинг кода 5 – Нахождение объёма продаж в магазине:

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.linear\_model import LinearRegression, Lasso, Ridge

from sklearn.ensemble import RandomForestRegressor

from sklearn.preprocessing import StandardScaler,MinMaxScaler

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

from sklearn.linear\_model import LassoCV, RidgeCV

df = pd.read\_csv('sales.csv')

X = df[['ADV', 'PRICE']]

y = df['SALES']

model = LinearRegression()

model.fit(X, y)

adv = 400

price = 79

sales\_prediction = model.predict([[adv, price]])

sales\_prediction



Рисунок 4 – Объём продаж в магазине

Задание 4) Загрузить набор данных sandler.csv - набор данных для предсказания дохода от показа фильмов. Применить простую линейную регрессию для предсказания дохода от показа фильмов от количества кинотеатров. Предскажите, чему будет равен доход для 1700 кинотеатров.

Листинг кода 6 – Нахождение предсказанного дохода с помощью простой линейной и множественной линейной регрессий:

from sklearn.linear\_model import LinearRegression

# Загрузка данных

df = pd.read\_csv('sandler.csv')

df['Opening Theaters']=df['Opening Theaters'].apply(lambda x: x.replace(' ','',10).replace("$",'',10))

df['Theaters']=df['Theaters'].apply(lambda x: x.replace(' ','',10).replace("$",'',10))

df['Gross']=df['Gross'].apply(lambda x: x.replace(' ','',10).replace("$",'',10))

df['Opening Gross']=df['Opening Gross'].apply(lambda x: x.replace(' ','',10).replace("$",'',10))

df.drop(['Date','Title', 'Genre', 'Studio'], axis=1, inplace=True)

df['Opening Theaters'] = pd.to\_numeric( df['Opening Theaters'])

df['Theaters'] = pd.to\_numeric( df['Theaters'])

df['Gross'] = pd.to\_numeric( df['Gross'])

df['Opening Gross'] = pd.to\_numeric( df['Opening Gross'])

X1, y = df.drop(["Gross"], axis = 1), df['Gross']

# Простая линейная регрессия

X\_simple = df[['Theaters']]

y\_simple = df['Gross']

model\_simple = LinearRegression()

model\_simple.fit(X\_simple, y\_simple)

theaters = 1700

gross\_prediction\_simple = model\_simple.predict([[theaters]])

print("Предсказанный доход для 1700 кинотеатров (простая линейная регрессия):", gross\_prediction\_simple)

# Множественная линейная регрессия

X\_multiple = df[['Theaters', 'Opening Theaters', 'Opening Gross']]

y\_multiple = df['Gross']

model\_multiple = LinearRegression()

model\_multiple.fit(X\_multiple, y\_multiple)

theaters = 1700

opening\_theaters = 1700

opening\_gross = 5000000

gross\_prediction\_multiple = model\_multiple.predict([[theaters, opening\_theaters, opening\_gross]])

print("Предсказанный доход для 1700 кинотеатров, 1700 открывающих кинотеатров и 5 миллионов дохода от премьеры (множественная линейная регрессия):", gross\_prediction\_multiple)



Рисунок 5 – Предсказанный доход

Задание 5) Открыть набор House3.csv. Предсказать стоимость дома по его площади и наличию камина. Сделать предсказание цены дома, используя модель линейной регрессии для размера дома равного 2, с камином.

Листинг кода 7 – Нахождение предсказанной цены дома:

from sklearn.linear\_model import LinearRegression

# Загружаем набор данных

data = pd.read\_csv('house.csv')

# Закодируем наличие камина как 0/1

data['Kamin'] = data['Kamin'].astype('category').cat.codes

# Создаем модель линейной регрессии

model = LinearRegression()

# Обучаем модель на данных о площади и наличии камина

model.fit(data[['Area', 'Kamin']], data['Price'])

# Делаем предсказание цены дома для площади 2 и наличия камина

predicted\_price = model.predict([[2, 1]])[0]

# Выводим предсказанную цену

print('Предсказанная цена дома:', predicted\_price)



Рисунок 6 – Предсказанная цена дома

Вывод: в ходе работы изучили способы построения регрессионных моделей.