|  |  |  |
| --- | --- | --- |
| Лабораторная работа № 3.  Регрессионный анализ | Студенты | Бокова О.Д. |
| Группа | ИВТ-363 |
| Преподаватель | Фокин Р.О. |
| Дата отчёта |  |

Решите задачу множественной линейной регрессии для выбранного

набора данных или произвольного набора данных из репозитория UC

Irvine Machine Learning Repository. Проанализируйте полученные результаты и сделайте выводы.

Код:

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

from sklearn.model\_selection import learning\_curve

from sklearn.metrics import make\_scorer

from sklearn import linear\_model, ensemble

# Загрузка данных

training\_data = pd.read\_csv("biofam.csv", delimiter = ';')

# Проверка размерности и типов данных

training\_data.shape

training\_data.info()

# Удаление пустых значений

training\_data = training\_data.dropna()

# Преобразование категориальных признаков в бинарные

from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder(handle\_unknown='ignore')

training\_points\_encoded = encoder.fit\_transform(training\_data[['nat\_1\_02']])

# Объединение закодированных признаков с числовыми

training\_points = np.hstack((training\_data[['a16', 'a29']].values, training\_points\_encoded.toarray()))

# Определение целевой переменной

training\_values = training\_data['birthyr'].values

# Создание моделей линейной регрессии и случайного леса

linear\_regression\_model = linear\_model.LinearRegression()

random\_forest\_model = ensemble.RandomForestRegressor(n\_estimators=100)

# Обучение моделей на тренировочных данных

linear\_regression\_model.fit(training\_points, training\_values)

random\_forest\_model.fit(training\_points, training\_values)

# Оценка качества моделей на кросс-валидации

from sklearn.model\_selection import cross\_val\_score

linear\_regression\_scores = cross\_val\_score(linear\_regression\_model, training\_points, training\_values, cv=10, scoring='r2')

random\_forest\_scores = cross\_val\_score(random\_forest\_model, training\_points, training\_values, cv=10, scoring='r2')

print('Linear Regression CV scores:', linear\_regression\_scores)

print('Random Forest CV scores:', random\_forest\_scores)

print('Average Linear Regression CV score:', np.mean(linear\_regression\_scores))

print('Average Random Forest CV score:', np.mean(random\_forest\_scores))

# Построение графика предсказаний линейной регрессии и истинных значений

predicted\_values = linear\_regression\_model.predict(training\_points)

plt.scatter(training\_values, predicted\_values)

plt.xlabel("True Values")

plt.ylabel("Predictions")

plt.show()

# Построение графика важности признаков в модели случайного леса

importances = random\_forest\_model.feature\_importances\_

std = np.std([tree.feature\_importances\_ for tree in random\_forest\_model.estimators\_], axis=0)

indices = np.argsort(importances)[::-1]

plt.figure()

plt.title("Feature importances")

plt.bar(range(training\_points.shape[1]), importances[indices], yerr=std[indices])

plt.xticks(range(training\_points.shape[1]), indices)

plt.xlim([-1, training\_points.shape[1]])

plt.show()

Результаты:

