Лабораторная работа 2
«Сравнение методов регрессионного анализа»

Цель работы

Освоить методы построения и сравнения различных типов регрессионных моделей: множественной линейной регрессии, регрессии с взаимодействиями и регуляризованной регрессии.

Задачи

1. Изучить основы регрессионного анализа и его применение для прогнозирования
2. Освоить построение множественной линейной регрессии
3. Изучить методы учета взаимодействий между переменными
4. Освоить методы регуляризации (Ridge, Lasso) и подбора гиперпараметров
5. Сравнить качество различных моделей регрессии

Исходные данные (пример, в работе — выбрать свои)

- Датасет: Concrete Compressive Strength с платформы Kaggle
- Описание: Данные о составе бетона и его прочности на сжатие
- Количество наблюдений: 1030
- Переменные:
  - `Cement` (кг/м^3) - количество цемента
  - `Blast Furnace Slag` (кг/м^3) - шлак
  - `Fly Ash` (кг/м^3) - зола-унос
  - `Water` (кг/м^3) - вода
  - `Superplasticizer` (кг/м^3) - суперпластификатор
  - `Coarse Aggregate` (кг/м^3) - крупный заполнитель
  - `Fine Aggregate` (кг/м^3) - мелкий заполнитель
  - `Age` (дни) - возраст бетона
  - `Compressive Strength` (МПа) - прочность на сжатие (целевая переменная)
- Источник: `https://www.kaggle.com/datasets/maajdl/yeh-concret-data`

Методика выполнения

Часть 1: Множественная линейная регрессия

1. Загрузка и предобработка данных
2. Разделение на обучающую и тестовую выборки (70/30)
3. Построение модели множественной линейной регрессии для предсказания прочности бетона на основе всех переменных
4. Оценка качества модели с помощью R^2, скорректированного R^2, RMSE
5. Проведение анализа значимости коэффициентов
6. Проверка выполнения предпосылок МНК (мультиколлинеарность, нормальность остатков)

Часть 2: Регрессия с взаимодействиями

1. Анализ корреляционной матрицы для выявления наиболее коррелированных переменных
2. Добавление в модель взаимодействий между 2-3 наиболее коррелированными переменными
3. Построение модели с взаимодействиями
4. Сравнение качества с базовой моделью
5. Интерпретация коэффициентов при взаимодействиях

Часть 3: Регрессия с регуляризацией

1. Построение Ridge регрессии с кросс-валидацией для подбора параметра α
2. Построение Lasso регрессии с кросс-валидацией для подбора параметра α
3. Сравнение коэффициентов регуляризованных моделей с обычной регрессией
4. Анализ переменных, исключенных в Lasso регрессии
5. Сравнение качества всех моделей

Необходимое программное обеспечение

Для Python

```
#
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.pipeline import Pipeline
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from scipy import stats
```

Для R

```r
#
library(tidyverse)
library(caret)
library(glmnet)
library(broom)
library(car)
library(MASS)
library(ggplot2)
library(corrplot)
```

Требования к отчету

Содержание отчета

1. Описание датасета и выбранных переменных
2. Методика исследования
3. Результаты по каждой части:
    - Код реализации
    - Таблицы с результатами
    - Графики и визуализации
    - Статистические выводы
4. Сравнительный анализ всех моделей
5. Выводы

Визуализация

- Матрица корреляций с тепловой картой
- Графики остатков для проверки предпосылок МНК
- Сравнение предсказанных и фактических значений
- Графики важности переменных
- Сравнение коэффициентов моделей

Аналитические таблицы

- Сравнение метрик качества (R^2, adj-R^2, RMSE, MAE) для всех моделей
- Таблица коэффициентов с указанием статистической значимости
- Результаты проверки мультиколлинеарности (VIF)
- Сравнение времени обучения моделей

Сравнение моделей

```python
#                                                    Python
models = {
    'Linear': linear_model,
    'Ridge': ridge_model,
    'Lasso': lasso_model,
    'Interaction': interaction_model
}

results = []
for name, model in models.items():
```

```
    y_pred = model.predict(X_test)
    r2 = r2_score(y_test, y_pred)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    results.append({'Model': name, 'R2': r2, 'RMSE': rmse})
```