Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №7

з дисципліни

Аналіз даних з використанням мови Python

Виконав:

студент групи ІА-24:
Криворучек В.С.

Перевірила:

ст. викладач
Тимофєєва Ю.С.

Київ 2025

Тема: Кластеризація та регресія в scikit-learn

Мета роботи: Ознайомитись з побудовою моделей для вирішення задач регресії та кластеризації в scikit-learn, визначити основні оцінки цих моделей.

Хід роботи

Завдання:

diamonds.csv. 1) спрогнозувати ціну діаманту

Написати програму, яка здійснює попередню обробку даних та

1) обирає незалежні ознаки (не менше трьох), щоб спрогнозувати залежну ознаку (за варіантом), використовуючи дві різні моделі регресії. Оцінити моделі за відповідними метриками та спробувати їх покращити;

2) підбирає оптимальну кількість кластерів та виконує кластеризацію даних. Оцінити результати за відповідними метриками.

Код програми:

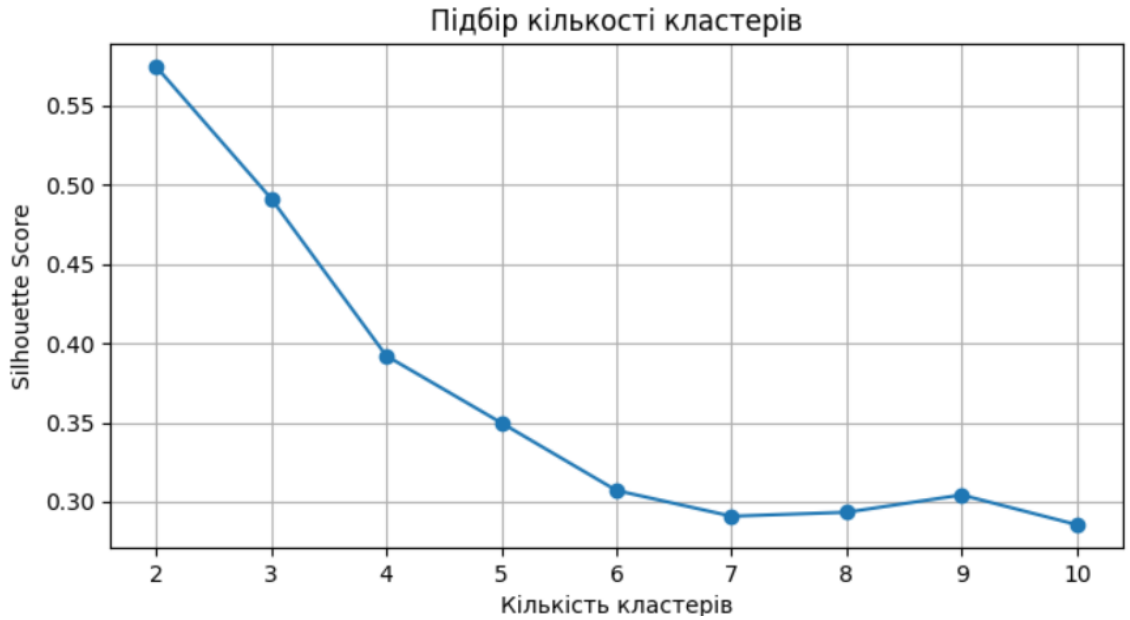
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Читання і очистка файлу
raw_df = pd.read_csv("diamonds.csv", header=None)
df = raw_df[0].str.split(",", expand=True)
df.columns = ['id', 'carat', 'cut', 'color', 'clarity', 'depth',
'table', 'price', 'x', 'y', 'z']
df = df.drop(index=0).reset_index(drop=True)

# Очистка: прибираємо лапки
for col in ['cut', 'color', 'clarity']:
    df[col] = df[col].str.replace('"', '')

# Конвертація числових значень
for col in ['carat', 'depth', 'table', 'price', 'x', 'y', 'z']:
    df[col] = df[col].astype(float)

# ----------------------------
# 1. РЕГРЕСІЯ
# ----------------------------
```

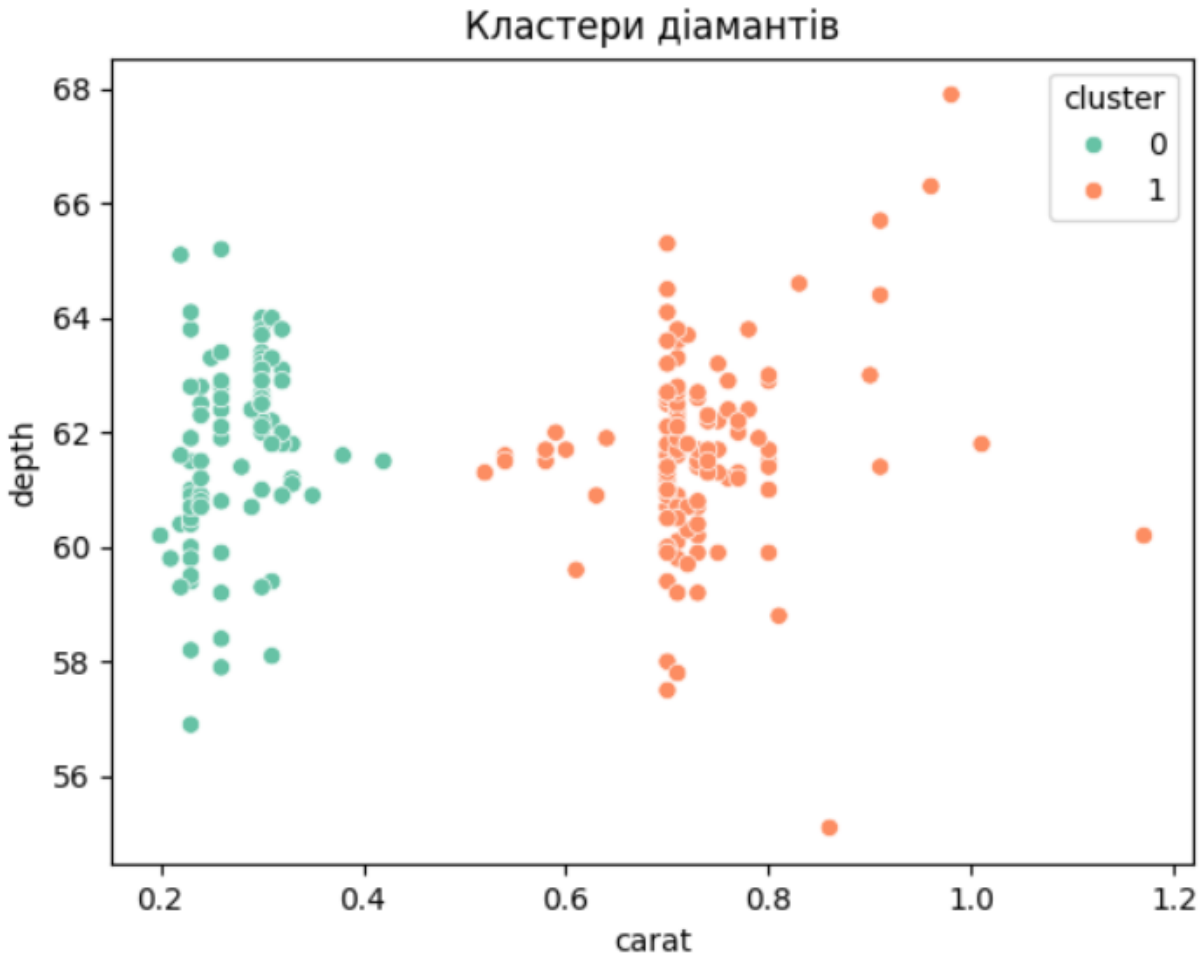
```
# Кодування категоріальних змінних
label_encoders = {}
for col in ['cut', 'color', 'clarity']:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

# Вибір ознак і цільової змінної
features = ['carat', 'depth', 'table', 'cut', 'color', 'clarity']
X = df[features]
y = df['price']

# Розділення на train/test
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Масштабування
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Модель 1: Лінійна регресія
model_lr = LinearRegression()
model_lr.fit(X_train_scaled, y_train)
y_pred_lr = model_lr.predict(X_test_scaled)

# Модель 2: Random Forest
model_rf = RandomForestRegressor(n_estimators=100, random_state=42)
model_rf.fit(X_train, y_train)
y_pred_rf = model_rf.predict(X_test)

# Оцінка моделей
def print_metrics(y_test, y_pred, name):
    print(f"\n{name}:\n" + "-"*30)
    print("MAE:", mean_absolute_error(y_test, y_pred))
    print("MSE:", mean_squared_error(y_test, y_pred))
    print("R²:", r2_score(y_test, y_pred))

print_metrics(y_test, y_pred_lr, "Лінійна регресія")
print_metrics(y_test, y_pred_rf, "Random Forest")

# ----------------------------
# 2. КЛАСТЕРИЗАЦІЯ
# ----------------------------

# Вибір ознак для кластеризації
X_clust = df[['carat', 'depth', 'table', 'x', 'y', 'z']]
X_clust_scaled = StandardScaler().fit_transform(X_clust)

# Підбір оптимальної кількості кластерів за метрикою Silhouette
silhouette_scores = []
K_range = range(2, 11)
for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = kmeans.fit_predict(X_clust_scaled)
    score = silhouette_score(X_clust_scaled, labels)
    silhouette_scores.append(score)

# Побудова графіку Silhouette
plt.figure(figsize=(8, 4))
```

```
plt.plot(K_range, silhouette_scores, marker='o')
plt.xlabel('Кількість кластерів')
plt.ylabel('Silhouette Score')
plt.title('Підбір кількості кластерів')
plt.grid(True)
plt.show()

# Найкраще k
best_k = K_range[np.argmax(silhouette_scores)]
print("Оптимальна кількість кластерів:", best_k)

# Кластеризація з оптимальним k
kmeans_final = KMeans(n_clusters=best_k, random_state=42, n_init=10)
df['cluster'] = kmeans_final.fit_predict(X_clust_scaled)

# Візуалізація кластерів (за двома ознаками)
sns.scatterplot(data=df, x='carat', y='depth', hue='cluster',
palette='Set2')
plt.title('Кластери діамантів')
plt.show()
```

Результати виконання коду:

```
Лінійна регресія:
------------------------------
MAE: 210.16139924308635
MSE: 100790.78089680569
R²: 0.9109533916941237

Random Forest:
------------------------------
MAE: 22.603181818181817
MSE: 2157.2462681818183
R²: 0.9980941167262234
Оптимальна кількість кластерів: 2

Process finished with exit code 0
```





Висновок: У ході виконання даної лабораторної роботи я ознайомився з побудовою моделей для вирішення задач регресії та кластеризації в scikit-learn, а також визначив основні оцінки цих моделей.