DỰ ĐOÁN CHẤT LƯỢNG KHÔNG KHÍ THÔNG QUA CÁC MÔ HÌNH HỒI QUY

1st Phạm Quốc Đăng
University of Information Technology
Ho Chi Minh, VietNam
19520036@gm.uit.edu.vn

2nd Nguyễn Thành Đạt
University of Information Technology
Ho Chi Minh, VietNam
19520040@gm.uit.edu.vn

3rd Nguyễn Hoàng Nam
University of Information Technology
Ho Chi Minh, VietNam
19520171@gm.uit.edu.vn

***Tóm tắt nội dung*—Kiểm tra và bảo vệ chất lượng không khí trên thế giới đã trở thành một trong những hoạt động thiết yếu đối với mỗi con người tại các khu công nghiệp và đô thị hiện nay. Với tình trạng ô nhiễm không khí ngày càng tăng, chúng ta cần triển khai các mô hình ghi lại thông tin về nồng độ các chất gây ô nhiễm không khí. Sự lắng đọng của các khí độc hại này trong không khí đang ảnh hưởng đến chất lượng cuộc sống của con người, làm thay đổi sức khỏe của họ, đặc biệt là ở các khu vực đô thị. Trong bài báo cáo này, kỹ thuật hồi quy được sử dụng để dự đoán nồng độ Benzene trong môi trường. Benzene là một hợp chất có thể gây ngộ độc cấp tính, trong thời gian ngắn, ngoài niêm mạc và phổi bị kích ứng, trung khu thần kinh cũng bị ức chế, xuất hiện hiện tượng đau đầu, buồn nôn...**

I. GIỚI THIỆU

Ô nhiễm không khí sẽ gây nguy hiểm đến sức khỏe và tính mạng con người ở các thành phố lớn, đặc biệt là người già và trẻ em. Đây không phải là vấn đề của riêng một người mà là vấn đề toàn cầu. Vì vậy, nhiều nước trên thế giới đã thực hiện trạm giám sát chất lượng không khí ở nhiều thành phố để quan sát các chất ô nhiễm như NO2, CO, SO2, PM2.5 và PM10 và cảnh báo người dân về chỉ số ô nhiễm vượt quá mức cho phép ngưỡng chất lượng.

Tập dữ liệu được sử dụng cho dự án được thu thập từ các thiết bị được đặt tại các khu vực bị ô nhiễm, ở tuyến đường đông đúc, trong một thành phố của Ý. Dữ liệu được ghi lại từ tháng 3 năm 2004 đến tháng 2 năm 2005 (gồm 9358 mẫu dữ liệu) [1], nó bao gồm phản ứng trung bình hàng giờ của các chất gây ô nhiễm không khí chính trong gần một năm. Bộ dữ liệu này được sử dụng để dự đoán nồng độ Benzene dựa trên các thông số khác như CO, NO, NO2, AH, T bằng cách sử dụng các mô hình phân tích hồi quy. Nó tạo ra nhận thức cho mọi người về sự xuống cấp của chất lượng không khí và ảnh hưởng của nó đối với sức khỏe. Hỗ trợ các nhà môi trường và chính phủ xây dựng các tiêu chuẩn và quy định về chất lượng không khí dựa trên các vấn đề về phơi nhiễm không khí độc hại và gây bệnh cũng như các nguy cơ liên quan đến sức khỏe đối với phúc lợi của con người.

II. TỔNG QUÁT

A. Dữ liệu đầu vào

Các dữ liệu như nồng độ trung bình hàng giờ của NOx, CO2, CO v.v. được đưa ra làm đầu vào cho hệ thống hỗ trợ ra quyết định.

Bảng I
CÁC THUỘC TÍNH CỦA BỘ DỮ LIỆU

Table	Table Column Head
0	Date (DD/MM/YYYY)
1	Time (HH.MM.SS)
2	True hourly average concentration CO in mg/m^3
3	PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)
4	True hourly averaged overall Non-Metanic Hydro Carbons concentration in $microgram/m^3$ (reference analyzer)
5	True hourly averaged Benzene concentration in $microg/m^3$ (reference analyzer)
6	PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)
7	True hourly averaged NOx concentration in ppb (reference analyzer)
8	PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)
9	True hourly averaged NO2 concentration in $microg/m^3$ (reference analyzer)
10	PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)
11	PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)
12	Temperature in Â°C
13	Relative Humidity (%)
14	AH Absolute Humidity

B. Tiền xử lý dữ liệu

Dữ liệu được chuyển đổi và xử lý dữ liệu thô sang định dạng dễ hiểu bằng các quy trình sau:

- Các cột như Data, time không đóng góp vào việc dự đoán nồng độ C6H6 trong không khí và cột NMHC_GT có nhiều giá trị bị mất. Do đó các cột này bị loại bỏ.
- Sau đó, sử dụng fillna để thay đổi các giá trị bị mất với giá trị trung bình của mỗi cột trong dataframe.
- Chuẩn hóa dữ liệu bằng StandardScaler của sklearn.

C. Các mô hình được sử dụng

Linear regression, Support Vector Regression, K-Nearest Neighbors

D. Dữ liệu đầu ra

Thông qua tập dữ liệu đầu vào tạo ra các mô hình máy học và từ các mô hình máy học này đưa ra dự đoán về nồng độ C6H6 với các tập dữ liệu mới. Sau đó ta sử dụng MSE và MAE để đánh giá kết quả dự đoán của các mô hình.

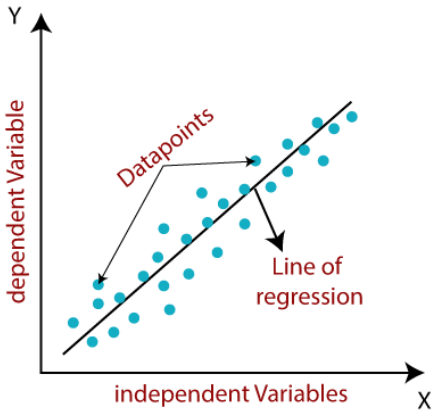
III. PHƯƠNG PHÁP

A. Linear regression

a) Giới thiệu Linear Regression

Linear regression là một trong những thuật toán Machine Learning dễ dàng và phổ biến nhất. Nó là một phương pháp thống kê được sử dụng để phân tích dự đoán. Linear Regression đưa ra dự đoán cho các biến liên tục hoặc số như doanh số, tiền lương, tuổi, giá sản phẩm, v.v.

Thuật toán Linear Regression cho thấy mối quan hệ tuyến tính giữa biến phụ thuộc (y) và một hoặc nhiều biến độc lập (x), do đó được gọi là hồi quy tuyến tính. Mô hình Linear Regression cung cấp một đường thẳng dốc thể hiện mối quan hệ giữa các biến. Hãy xem xét hình ảnh dưới đây:



Hình 1. Đồ thị miêu tả Linear Regression

Về mặt toán học, chúng ta có thể biểu diễn thuật toán Linear Regression như sau:

$$y = wx + b \tag{1}$$

Trong đó:

- y: Biến phụ thuộc (Biến mục tiêu)
- x: Biến độc lập (Biến dự báo)
- w và b là các tham số huấn luyện sẽ được tối ưu hóa trong quá trình huấn luyện.

b) Điều chỉnh siêu tham số cho thuật toán Linear Regression

Các tham số được điều chỉnh trong thuật toán Linear Regression: [2]

- fit_intercept: xác định có nên tính toán hệ số chặn cho mô hình này hay không.
- copy_X: xác định X sẽ được sao chép hay nó có thể bị ghi đè.

```
parameters = {"fit_intercept": [True, False],
              "copy_X": [True, False],
              "positive": [True, False]
             }
```

Hình 2. Chuẩn bị parameters cho GridSearchCV (Linear Regression)

Sau đó, sử dụng phương pháp Grid Search [5] và thu được các siêu tham số tương ứng cho Linear Regression và tập dữ liệu huấn luyện:

'copy_X': True, 'fit_intercept': True, 'positive': False

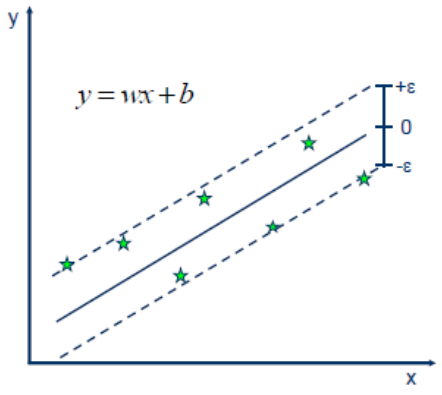
B. Support Vector Regression

a) Giới thiệu Support Vector Regression

Support Vector Machine (SVM) là một thuật toán Machine Learning rất phổ biến được sử dụng trong cả Hồi quy và Phân loại. Support Vector Regression (SVR) là một thuật toán học có giám sát được sử dụng để dự đoán các giá trị rời rạc. SVR tương tự như Linear Regression ở chỗ phương trình của đường thẳng là y = wx + b. Trong SVR, đường thẳng này được gọi là hyperplane (siêu phẳng). Các điểm dữ liệu ở hai bên của hyperplane gần hyperplane nhất được gọi là các vectơ hỗ trợ được sử dụng để vẽ boundary line (đường biên).

Không giống như các mô hình Hồi quy khác cố gắng giảm thiểu sai số giữa giá trị thực và giá trị dự đoán, SVR cố gắng khớp đường tốt nhất trong giá trị ngưỡng (khoảng cách giữa hyperplane và boundary lines). Như vậy, có thể nói rằng mô hình SVR cố gắng thỏa mãn điều kiện:

$$-\varepsilon < y - wx + b < \varepsilon \tag{2}$$



Hình 3. Đồ thị miêu tả Support Vector Regression

b) Điều chỉnh siêu tham số cho thuật toán Support Vector Regression

Các tham số được điều chỉnh trong thuật toán Support Vector Regression: [3]

- C: Tham số chính quy hóa. Độ mạnh của chính quy hóa tỷ lệ nghịch với C (tham số này kiểm soát sự đánh đổi việc phân loại chính xác với độ suôn sẻ của decision boundary)
- gamma: hệ số của các kernel 'rbf', 'poly' và 'sigmoid'
- kernel: Chỉ định loại kernel sẽ được sử dụng trong thuật toán.

```
parameters = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']
             }
```

Hình 4. Chuẩn bị parameters cho GridSearchCV (SVR)

Sau đó, sử dụng phương pháp Grid Search và thu được các siêu tham số tương ứng cho Support Vector Regression và tập dữ liệu huấn luyện:

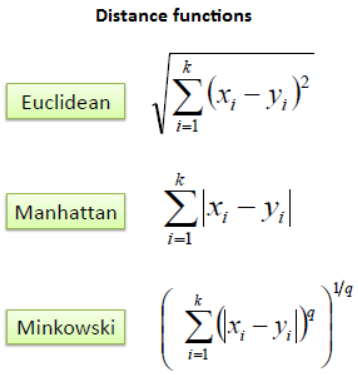
'kernel': rbf, 'gamma': 0.01, 'C': 1000

C. K-Nearest Neighbors

a) Giới thiệu K-Nearest Neighbors

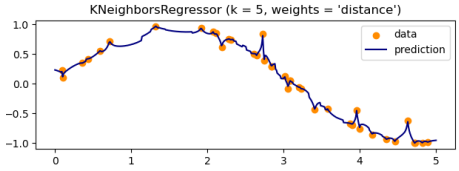
Thuật toán K-Nearest Neighbors, còn được gọi là KNN hoặc k-NN, là một thuật toán phân loại học có giám sát, phi tham số, sử dụng khoảng cách gần để phân loại hoặc dự đoán về việc nhóm một điểm dữ liệu riêng lẻ. Mặc dù nó có thể được sử dụng cho các vấn đề hồi quy hoặc phân loại, nhưng nó thường được sử dụng như một thuật toán phân loại, dựa trên giả định rằng các điểm tương tự có thể được tìm thấy gần nhau.

Thuật toán KNN cho rằng những dữ liệu tương tự nhau sẽ tồn tại gần nhau trong một không gian, từ đó công việc của chúng ta là sẽ tìm k điểm gần với dữ liệu cần kiểm tra nhất. Việc tìm khoảng cách giữa 2 điểm cũng có nhiều công thức có thể sử dụng, tùy trường hợp mà chúng ta lựa chọn cho phù hợp.



Hình 5. Ba cách tính khoảng cách giữa hai điểm x, y có k thuộc tính

Trong bài toán Regresssion, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K=1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó. Vì vậy, thuật toán KNeighborsRegressor() từ thư viện sklearn sẽ thực hiện là tính toán hồi quy cho tập dữ liệu, sau đó kiểm tra n_neighbors kết quả của các lân cận và lấy trung bình cộng các kết quả đó, cho ra kết quả ước tính.



Hình 6. Đồ thị miêu tả K-Nearest Neighbors Regression

b) Điều chỉnh siêu tham số cho thuật toán K-Nearest Neighbors

Các tham số được điều chỉnh trong thuật toán K-Nearest Neighbors: [4]

- n_neighbors: Số lượng neighbors mặc định cho các truy vấn lân cận (kneighbors)
- weights: chức năng weights được sử dụng trong dự đoán. Những lựa chọn của weights:
  - uniform: tất cả các điểm trong mỗi vùng lân cận đều có trọng số như nhau.
  - distance: các neighbors gần điểm truy vấn hơn sẽ có ảnh hưởng lớn hơn các neighbors ở xa hơn.
  - callable: một hàm do người dùng xác định chấp nhận một mảng khoảng cách và trả về một mảng có cùng hình dạng chứa các trọng số.
- algorithm: Thuật toán được sử dụng để tính toán các neighbors gần nhất: auto, ball_tree, kd_tree, brute

```
parameters = {'n_neighbors' : range(1, 31),
              'weights' : ['uniform', 'distance'],
              'algorithm' : ['auto', 'ball_tree', 'kd_tree', 'brute']
             }
```

Hình 7. Chuẩn bị parameters cho GridSearchCV (KNeighborsRegressor)

Sau đó, sử dụng phương pháp Grid Search và thu được các siêu tham số tương ứng cho KNeighborsRegressor và tập dữ liệu huấn luyện:

'algorithm': auto, 'n_neighbors': 5, 'weights': distance

IV. THỰC NGHIỆM

Tiến hành áp dụng các siêu tham số đã tìm được vào các mô hình hồi quy trên và đưa ra các dự đoán về nồng độ C6H6 trong không khí. Sau đó sử dụng phương pháp Mean Squared Error (MSE), Mean Absolute Error (MAE), R-Square (R2) để đánh giá hiệu suất của các mô hình

Mean Squared Error (MSE) của một phép ước lượng là trung bình của bình phương các sai số, tức là chênh lệch bình

phương trung bình giữa các giá trị ước tính và giá trị thực. Giá trị này luôn luôn không âm và có giá trị lý tưởng là 0. Chúng ta có độ đo MSE được tính theo công thức sau:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (3)$$

Các giá trị MSE ứng với các mô hình sau khi sử dụng các siêu tham số được ghi nhận trong bảng dưới đây:

Bảng II
GIÁ TRỊ MSE CỦA CÁC MÔ HÌNH

Regression type	MSE value
Linear Regression	1.2763351670716676
Support Vector Regression	0.01874239729528239
K-Neighbors Regression	0.9712837031401753

Mean Absolute Error (MAE) là một phương pháp đo lường sự khác biệt giữa hai biến liên tục. Giả sử rằng X và Y là hai biến liên tục thể hiện kết quả dự đoán của mô hình và kết quả thực tế. Chúng ta có độ đo MAE được tính theo công thức sau:

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n} \quad (4)$$

Các giá trị MAE ứng với các mô hình sau khi sử dụng các siêu tham số được ghi nhận trong bảng dưới đây:

Bảng III
GIÁ TRỊ MAE CỦA CÁC MÔ HÌNH

Regression type	MAE value
Linear Regression	0.8115887191669415
Support Vector Regression	0.07120944693054165
K-Neighbors Regression	0.633825593435906

R-Squared là thước đo thống kê về mức độ phù hợp cho biết mức độ thay đổi của một biến phụ thuộc được giải thích bởi (các) biến độc lập trong mô hình hồi quy. R-Squared cho biết mô hình đó hợp với dữ liệu ở mức bao nhiêu %.

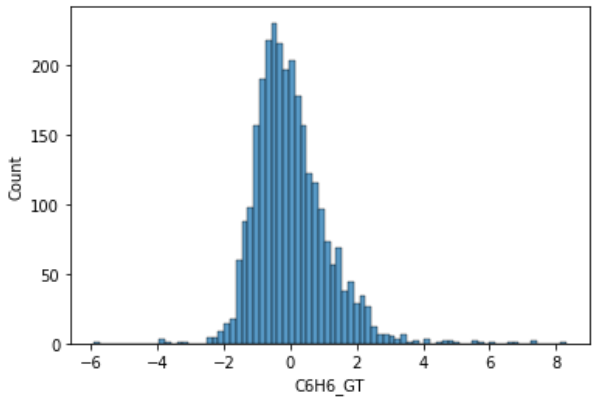
$$R^2 = 1 - \frac{ESS}{TSS} \quad (5)$$

Các giá trị R-Squared ứng với các mô hình sau khi sử dụng các siêu tham số được ghi nhận trong bảng dưới đây:

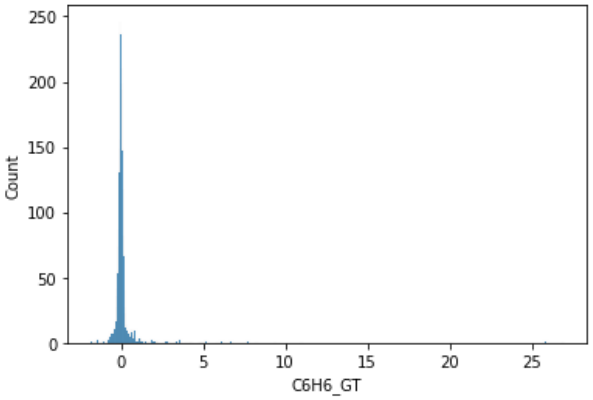
Bảng IV
GIÁ TRỊ R-SQUARED CỦA CÁC MÔ HÌNH

Regression type	R-SQUARED value
Linear Regression	0.9764916033677273
Support Vector Regression	0.9996547899636206
K-Neighbors Regression	0.9821102457058604

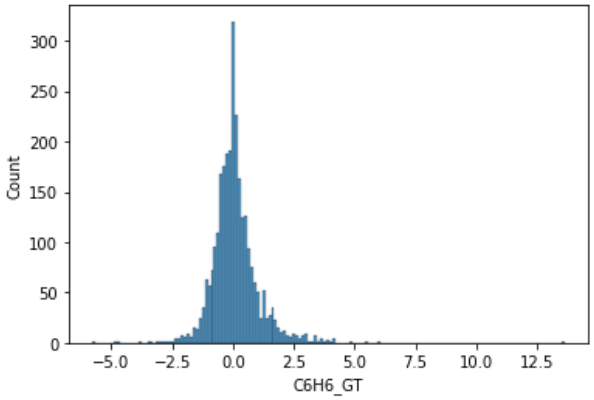
Sau đây là các biểu đồ biểu thị số lượng và độ sai lệch giữa các giá trị thực tế và giá trị dự đoán [sns.histplot(y_true-y_pred)] của nồng độ C6H6:



Hình 8. Số lượng và độ sai lệch giữa y_true và y_pred (Linear Regression)



Hình 9. Số lượng và độ sai lệch giữa y_true và y_pred (Support Vector Regression)



Hình 10. Số lượng và độ sai lệch giữa y_true và y_pred (K-Neighbors Regression)

Như đã thấy ở trên, Support Vector Regression có các giá trị đánh giá MAE, MSE, R-Squared và số lượng cũng như độ sai lệch giữa y_true và y_pred là tốt nhất cho thấy mô hình đã thực hiện tốt trên tập dữ liệu được xem xét. Bên cạnh đó hai mô hình Linear Regression và K-Neighbors Regression cũng có kết quả đánh giá khá tốt cho thấy chúng có thể được cân nhắc sử dụng trên tập dữ liêu xem xét phục vụ cho một vài mục đích khác.

V. PHẦN KẾT LUẬN

Nhóm đã đề xuất thực hiện phân tích và so sánh các mô hình hồi quy Linear regression, Support Vector Regression, K-Nearest Neighbors cho tập dữ liệu về chất lượng không khí (Air Quality UCI) và có các kết luận như sau:

- Support Vector Regression cho kết quả rất khả quan với tập dữ liệu. Vì Support Vector Regression có thể xử lý trên không gian có số chiều lớn (với tập dữ liệu trên có 11 input), có khả năng áp dụng Kernel cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.
- K-Nearest Neighbors cho kết quả cũng khá tốt với tập dữ liệu. Vì K-Nearest Neighbors có khả năng xử lý tốt với tập dữ liệu nhiễu tuy nhiên lại rất nhạy cảm với nhiễu khi K (không phù hợp). Với K càng lớn thì độ phức tạp và thời gian thực thi cũng sẽ tăng lên.
- Tuy có kết quả không tốt so với hai mô hình trên nhưng kết quả đánh giá cũng khá khả quan. Hạn chế đầu tiên của Linear Regression là nó rất nhạy cảm với nhiễu, đối với dữ liệu phi tuyến tính, hồi quy đa thức có thể khá khó khăn để triển khai.
- Kết luận, đối với tập dữ liệu Air Quality có nhiều thuộc tính đánh giá và được thu thập thực tế thì việc nhiễu dữ liệu là không thể tránh khỏi. Trong ba mô hình đã thực hiện thì Support Vector Regression có thể đáp ứng và thực hiện tốt đối với tập dữ liệu trên.

Các mô hình hồi quy trên được sử dụng để dự đoán nồng độ Benzene trong môi trường. Việc tiếp xúc với Benzene có thể gây ngộ độc cấp tính, phổi bị kích ứng, bị ức chế, xuất hiện hiện tượng đau đầu, buồn nôn. Vì vậy, dự án này rất hữu ích để biết về nồng độ Benzene trong không khí. Nó tạo ra nhận thức cho mọi người về sự xuống cấp chất lượng không khí và những ảnh hưởng đối với sức khỏe.

NHÌN NHẬN

Chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Vinh Tiệp đã quan tâm, hướng dẫn, truyền đạt những kiến thức và kinh nghiệm cho chúng em trong suốt thời gian học tập môn Lập trình Python cho Máy học.

Sau đây là bảng phân công công việc của các thành viên trong nhóm:

Bảng V
THÀNH VIÊN VÀ CÔNG VIỆC THỰC HIỆN

Họ và Tên - MSSV	Nhiệm vụ
Phạm Quốc Đăng 19520036	Thực hiện phần Clean the dataset, mô hình Support Vector Regression, viết báo cáo
Nguyễn Thành Đạt 19520040	Thực hiện phần Data Correlation, mô hình Linear Regression, viết báo cáo
Nguyễn Hoàng Nam 19520171	Thực hiện phần Data Preparation, mô hình K-Neighbors Regression, viết báo cáo

TÀI LIỆU

[1] "UCI ML Air QualityDataset", kaggle.com. https://www.kaggle.com/datasets/nishantbhadauria/datasetucimlairquality (accessed Jan. 3, 2023)

[2] "sklearn.linear_model.LinearRegression", scikit-learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html (accessed Jan. 3, 2023)

[3] "sklearn.svm.SVR", scikit-learn.org. https://scikitlearn.org/stable/modules/generated/sklearn.svm.SVR.html (accessed Jan. 3, 2023)

[4] "sklearn.neighbors.KNeighborsRegressor", scikit-learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html (accessed Jan. 3, 2023)

[5] "sklearn.model_selection.GridSearchCV", scikit-learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (accessed Jan. 3, 2023)