Trong hướng dẫn từng bước này, bạn sẽ:

-Tải xuống và cài đặt Python SciPy và nhận gói hữu ích nhất cho việc học máy bằng Python.

-Tải tập dữ liệu và hiểu cấu trúc của nó bằng cách sử dụng tóm tắt thống kê và hiển thị dữ liệu.

-Tạo 6 mô hình học máy, chọn tốt nhất và xây dựng sự tự tin rằng độ chính xác là đáng tin cậy.

**Bạn bắt đầu học máy bằng Python như thế nào?**

Cách tốt nhất để học máy học là thiết kế và hoàn thành các dự án nhỏ.

**Python có thể trở nên đáng sợ khi bắt đầu**

Python là một ngôn ngữ thông dịch phổ biến và mạnh mẽ. Không giống như R, Python là một ngôn ngữ và nền tảng hoàn chỉnh mà bạn có thể sử dụng cho cả nghiên cứu và phát triển và phát triển các hệ thống sản xuất.

Ngoài ra còn có rất nhiều mô-đun và thư viện để lựa chọn, cung cấp nhiều cách để thực hiện từng tác vụ. Nó có thể cảm thấy bị áp đảo( stund 3s).

Cách tốt nhất để bắt đầu sử dụng Python để học máy là hoàn thành một dự án.

Nó sẽ buộc bạn phải cài đặt và khởi động trình thông dịch Python (ít nhất).

Nó sẽ cho bạn cái nhìn toàn cảnh về cách bước qua một dự án nhỏ.

Nó sẽ cung cấp cho bạn sự tự tin, có thể để đi vào các dự án nhỏ của riêng bạn.

**Người mới bắt đầu cần một dự án từ đầu đến cuối**

Sách và các khóa học đang làm bạn bực bội. Họ cung cấp cho bạn rất nhiều công thức và đoạn trích, nhưng bạn không bao giờ có thể xem tất cả chúng phù hợp với nhau như thế nào.( QC vl)

Khi bạn đang áp dụng máy học tập cho bộ dữ liệu của riêng bạn, bạn đang làm việc trên một dự án.

Một dự án học theo một số bước cơ bản không đổi sau:

1.Xác định vấn đề.

2.Chuẩn bị dữ liệu.

3.Đánh giá các thuật toán.

4.Cải thiện kết quả.

5.Kết quả hiện tại.

Cách tốt nhất là đi từng bước một với bài toán. Cụ thể là, từ việc tải dữ liệu, tóm tắt dữ liệu, đánh giá các thuật toán và đưa ra một số dự đoán.

**Hello World trong Machine Learning**

Dự án nhỏ tốt nhất để bắt đầu với một công cụ mới là phân loại hoa iris (ví dụ: tập dữ liệu iris).

Đây là một dự án tốt vì nó được hiểu rất rõ.

Thuộc tính là số nên bạn phải tìm cách tải và xử lý dữ liệu.

Đây là một vấn đề phân loại, cho phép bạn thực hành với một loại thuật toán học có giám sát dễ dàng hơn.

Đây là một vấn đề phân loại nhiều lớp (đa danh nghĩa) có thể yêu cầu một số xử lý chuyên ngành.

Nó chỉ có 4 thuộc tính và 150 hàng, có nghĩa là nó nhỏ và dễ dàng phù hợp với bộ nhớ (và một màn hình hoặc trang A4).

Tất cả các thuộc tính số đều nằm trong cùng một đơn vị và cùng tỷ lệ, không yêu cầu bất kỳ tỷ lệ hoặc biến đổi đặc biệt nào để bắt đầu.

Chúng ta hãy bắt đầu với dự án học máy hello world của bạn bằng Python.

**Học máy bằng Python: Hướng dẫn từng bước**

Trong phần này, chúng ta sẽ làm việc thông qua một dự án học máy nhỏ từ đầu đến cuối.

Dưới đây là tổng quan về những gì chúng tôi sẽ giới thiệu:

Bước 1: Cài đặt nền tảng Python và SciPy.

Bước 2: Tải tập dữ liệu.

Bước 3: Tóm tắt tập dữ liệu.

Bước 4: Trực quan hóa tập dữ liệu.

Bước 5: Đánh giá một số thuật toán.

Bước 6: Đưa ra một số dự đoán.

Hãy dành thời gian của bạn. Làm việc qua từng bước.

Hãy tự mình gõ các lệnh hoặc sao chép và dán các lệnh để tăng tốc cho mọi thứ.

Bước 1: Load data

Chúng ta sẽ sử dụng bộ dữ liệu hoa iris. Tập dữ liệu này nổi tiếng vì nó được sử dụng làm bộ dữ liệu “hello world” trong học máy và thống kê của khá nhiều người.

Tập dữ liệu chứa 150 quan sát hoa iris. Có bốn cột của các phép đo hoa theo cm. Cột thứ năm là loài hoa được quan sát. Tất cả các loại hoa quan sát thuộc về một trong ba loài.

Tải dữ liệu iris từ URL tệp CSV.

2.2 Tải tập dữ liệu

Chúng tôi có thể tải dữ liệu trực tiếp từ kho lưu trữ Máy học UCI.

Chúng tôi đang sử dụng pandas để tải dữ liệu. Chúng tôi cũng sẽ sử dụng pandas để khám phá dữ liệu với cả số liệu thống kê mô tả và hiển thị dữ liệu.

Lưu ý rằng chúng tôi đang chỉ định tên của mỗi cột khi tải dữ liệu. Điều này sẽ giúp sau này khi chúng tôi khám phá dữ liệu.

Tập dữ liệu sẽ tải mà không có sự cố.

Nếu bạn có vấn đề về mạng, bạn có thể tải tệp iris.data vào thư mục làm việc của bạn và tải nó bằng cùng một phương thức, thay đổi URL thành tên tệp cục bộ.

3. Tóm tắt tập dữ liệu

Bây giờ là lúc để xem dữ liệu.

Trong bước này, chúng ta sẽ xem xét dữ liệu theo một vài cách khác nhau:

-Kích thước của tập dữ liệu.

-Peek tại các dữ liệu chính nó.

-Tóm tắt thống kê của tất cả các thuộc tính.

-Phân tích dữ liệu theo biến lớp.

Đừng lo lắng, mỗi cái nhìn vào dữ liệu là một lệnh. Đây là những lệnh hữu ích mà bạn có thể sử dụng một lần nữa và một lần nữa trên các dự án trong tương lai.

3.1 Kích thước của tập dữ liệu

Chúng ta có thể nhận được một ý tưởng nhanh về bao nhiêu trường hợp (hàng) và số lượng thuộc tính (cột) dữ liệu chứa với thuộc tính hình dạng.

3.2 Peek tại Data

Nó cũng luôn luôn là một ý tưởng tốt để thực sự nhãn cầu dữ liệu của bạn.

3.3 Tóm tắt thống kê

Bây giờ chúng ta có thể xem một bản tóm tắt của từng thuộc tính.

Điều này bao gồm số lượng, giá trị trung bình, giá trị nhỏ nhất và tối đa cũng như một số phần trăm.

3.4 Phân bố theo lớp

Bây giờ chúng ta hãy xem xét số lượng các cá thể (các hàng) thuộc về mỗi lớp. Chúng ta có thể xem đây là một số tuyệt đối.

**4. Hiển thị dữ liệu**

Bây giờ chúng ta có một ý tưởng cơ bản về dữ liệu. Chúng ta cần mở rộng điều đó với một số cách trực quan hóa.

Chúng ta sẽ xem xét hai loại hộp:

1. Các hộp đơn lẻ để hiểu rõ hơn về từng thuộc tính.

2. Hộp đa biến để hiểu rõ hơn mối quan hệ giữa các thuộc tính.

**4.1 Các hộp đơn lẻ**

Chúng tôi bắt đầu với một số hộp bất biến - là các hộp của mỗi biến riêng lẻ.

Cho rằng các biến đầu vào là số, chúng ta có thể tạo hộp và hộp của mỗi hộp.

**4.2 Hộp đa biến**

Bây giờ chúng ta có thể xem xét các tương tác giữa các biến.

Trước tiên, hãy xem xét các phân tán của tất cả các cặp thuộc tính. Điều này có thể hữu ích để phát hiện mối quan hệ có cấu trúc giữa các biến đầu vào.

Lưu ý nhóm đường chéo của một số cặp thuộc tính. Điều này cho thấy mối tương quan cao và mối quan hệ có thể dự đoán được.

**5. Đánh giá một số thuật toán**

Bây giờ là lúc tạo một số mô hình dữ liệu và ước tính độ chính xác của chúng trên dữ liệu không nhìn thấy.

Dưới đây là những gì chúng tôi sẽ giới thiệu trong bước này:

-Tách bộ dữ liệu xác thực.

-Thiết lập khai thác thử nghiệm để sử dụng xác thực chéo 10 lần.

-Xây dựng 5 mô hình khác nhau để dự đoán loài từ phép đo hoa

-Chọn mô hình tốt nhất.

**5.1 Tạo tập dữ liệu xác nhận**

Sử dụng các phương pháp thống kê để ước tính độ chính xác của các mô hình mà chúng tôi tạo ra trên dữ liệu không nhìn thấy được. Chúng tôi cũng muốn ước tính cụ thể hơn về tính chính xác của mô hình tốt nhất trên dữ liệu không nhìn thấy bằng cách đánh giá dữ liệu đó trên dữ liệu không nhìn thấy thực tế.

Đó là, sẽ giữ lại một số dữ liệu mà các thuật toán sẽ không thấy và chúng tôi sẽ sử dụng dữ liệu này để có ý tưởng thứ hai và độc lập về độ chính xác của mô hình tốt nhất.

Chia bộ dữ liệu đã tải thành hai, 80% trong số đó chúng tôi sẽ sử dụng để đào tạo mô hình của chúng tôi và 20% mà chúng tôi sẽ giữ lại dưới dạng tập dữ liệu xác thực.

**5.2 Kiểm tra khai thác**

Chúng tôi sẽ sử dụng xác thực chéo 10 lần để ước tính độ chính xác.

Điều này sẽ chia bộ dữ liệu của chúng tôi thành 10 phần, đào tạo trên 9 và kiểm tra trên 1 và lặp lại cho tất cả các kết hợp phân chia thử nghiệm đào tạo.

Chúng tôi đang sử dụng chỉ số ‘chính xác’ để đánh giá các mô hình. Đây là tỷ lệ số lượng các trường hợp được dự đoán chính xác chia cho tổng số phiên bản trong tập dữ liệu nhân với 100 để cung cấp phần trăm (ví dụ: 95% chính xác). Chúng tôi sẽ sử dụng biến số điểm khi chúng tôi chạy xây dựng và đánh giá từng mô hình tiếp theo.

**5.3Mô hình xây dựng**

Không biết thuật toán nào sẽ tốt về vấn đề này hoặc cấu hình nào sẽ sử dụng. Nhận được một ý tưởng từ các ô rằng một số các lớp được tách một phần tuyến tính trong một số kích thước, vì vậy chúng tôi đang mong đợi kết quả nói chung tốt.

Hãy đánh giá 6 thuật toán khác nhau:

-Hồi quy logistic (LR)

-Phân tích phân biệt tuyến tính (LDA)

-Hàng xóm K-gần nhất (KNN).

-Phân loại và cây hồi quy (CART).

-Gaussian Naive Bayes (NB).

-Hỗ trợ máy Vector (SVM).

Đây là một hỗn hợp tốt của các thuật toán đơn giản (LR và LDA), phi tuyến (KNN, CART, NB và SVM). Chúng tôi đặt lại hạt giống số ngẫu nhiên trước mỗi lần chạy để đảm bảo rằng việc đánh giá từng thuật toán được thực hiện bằng cách sử dụng chính xác cùng một phân tách dữ liệu. Nó đảm bảo kết quả có thể so sánh trực tiếp.

5.4 Chọn mô hình tốt nhất

Bây giờ chúng tôi có 6 mô hình và ước tính độ chính xác cho mỗi mô hình. Chúng ta cần phải so sánh các mô hình với nhau và chọn chính xác nhất.

Chạy ví dụ trên, chúng tôi nhận được các kết quả thô sau:

Lưu ý, kết quả của bạn có thể khác nhau. Để biết thêm về điều này, hãy xem bài đăng:

Chúng ta có thể thấy rằng có vẻ như KNN có điểm chính xác ước tính lớn nhất.

Chúng tôi cũng có thể tạo ra một âm mưu của các kết quả đánh giá mô hình và so sánh sự lây lan và độ chính xác trung bình của từng mô hình. Có một số lượng các phép đo chính xác cho mỗi thuật toán vì mỗi thuật toán được đánh giá 10 lần (xác nhận chéo 10 lần).

Bạn có thể thấy rằng các ô và ô trống được đè bẹp ở đầu phạm vi, với nhiều mẫu đạt được độ chính xác 100%.

**6. Đưa ra dự đoán**

Thuật toán KNN là mô hình chính xác nhất mà chúng tôi đã thử nghiệm. Bây giờ chúng tôi muốn có được một ý tưởng về tính chính xác của mô hình trên bộ xác thực của chúng tôi.

Điều này sẽ cung cấp cho chúng tôi kiểm tra cuối cùng độc lập về tính chính xác của mô hình tốt nhất. Nó là có giá trị để giữ một bộ xác nhận chỉ trong trường hợp bạn thực hiện một phiếu trong quá trình đào tạo, chẳng hạn như overfitting để tập huấn luyện hoặc rò rỉ dữ liệu. Cả hai sẽ dẫn đến kết quả quá lạc quan.

Chúng tôi có thể chạy mô hình KNN trực tiếp trên bộ xác nhận và tóm tắt kết quả dưới dạng điểm chính xác cuối cùng, ma trận nhầm lẫn và báo cáo phân loại.

Bạn có thể làm máy học trong Python

Làm việc thông qua hướng dẫn ở trên. Nó sẽ đưa bạn 5 đến 10 phút, tối đa!

Bạn không cần phải hiểu mọi thứ. (ít nhất là không phải bây giờ) Mục tiêu của bạn là chạy qua hướng dẫn từ đầu đến cuối và nhận được kết quả. Bạn không cần phải hiểu mọi thứ trong lần vượt qua đầu tiên. Liệt kê các câu hỏi của bạn khi bạn đi. Sử dụng nhiều cú pháp trợ giúp ("FunctionName") trong Python để tìm hiểu về tất cả các chức năng bạn đang sử dụng.

Bạn không cần biết các thuật toán hoạt động như thế nào. Điều quan trọng là phải biết về những hạn chế và cách cấu hình thuật toán học máy. Nhưng tìm hiểu về các thuật toán có thể đến sau. Bạn cần phải xây dựng kiến ​​thức thuật toán này chậm trong một thời gian dài. Hôm nay, bắt đầu bằng cách cảm thấy thoải mái với nền tảng này.

Bạn không cần phải là một lập trình viên Python. Cú pháp của ngôn ngữ Python có thể trực quan nếu bạn mới sử dụng nó. Cũng giống như các ngôn ngữ khác, hãy tập trung vào các cuộc gọi hàm (ví dụ: hàm ()) và các bài tập (ví dụ: a = “b”). Điều này sẽ giúp bạn tận dụng tối đa. Bạn là nhà phát triển, bạn biết cách tìm hiểu các khái niệm cơ bản về ngôn ngữ thật nhanh. Chỉ cần bắt đầu và đi sâu vào chi tiết sau.

Bạn không cần phải là một chuyên gia học máy. Bạn có thể tìm hiểu về các lợi ích và hạn chế của các thuật toán khác nhau sau này và có rất nhiều bài đăng mà bạn có thể đọc sau để đánh dấu các bước của dự án học máy và tầm quan trọng của việc đánh giá độ chính xác bằng cách sử dụng xác thực chéo.

Còn các bước khác trong dự án học máy. Chúng tôi đã không đề cập đến tất cả các bước trong một dự án học máy vì đây là dự án đầu tiên của bạn và chúng tôi cần tập trung vào các bước chính. Cụ thể là, tải dữ liệu, xem xét dữ liệu, đánh giá một số thuật toán và đưa ra một số dự đoán. Trong các hướng dẫn sau, chúng ta có thể xem xét các công việc chuẩn bị dữ liệu và cải tiến kết quả khác.

Tóm lược

Trong bài này, bạn đã khám phá từng bước cách hoàn thành dự án học máy đầu tiên của mình bằng Python.

Bạn đã khám phá ra rằng việc hoàn thành một dự án đầu cuối nhỏ để tải dữ liệu để đưa ra dự đoán là cách tốt nhất để làm quen với một nền tảng mới.

Bước tiếp theo của bạn

Bạn có làm việc thông qua hướng dẫn không?

Làm việc thông qua hướng dẫn ở trên.

Liệt kê bất kỳ câu hỏi nào bạn có.

Tìm kiếm hoặc nghiên cứu câu trả lời.

Hãy nhớ rằng, bạn có thể sử dụng trợ giúp (“FunctionName”) trong Python để được trợ giúp về bất kỳ chức năng nào.

=============================================================================================================================================================================================

Cách tạo mô hình ARIMA cho dự báo chuỗi thời gian bằng Python

Một phương pháp thống kê phổ biến và được sử dụng rộng rãi cho dự báo chuỗi thời gian là mô hình ARIMA.

ARIMA là một từ viết tắt viết tắt của AutoRegressive Integrated Moving Average. Nó là một lớp mô hình bắt giữ một bộ các cấu trúc thời gian chuẩn khác nhau trong dữ liệu chuỗi thời gian.

Trong hướng dẫn này, bạn sẽ khám phá cách phát triển một mô hình ARIMA cho dữ liệu chuỗi thời gian bằng Python.

Sau khi hoàn thành hướng dẫn này, bạn sẽ biết:

-Giới thiệu về mô hình ARIMA các tham số được sử dụng và các giả định của mô hình.

-Làm thế nào để phù hợp với một mô hình ARIMA để dữ liệu và sử dụng nó để thực hiện dự báo.

-Cách cấu hình mô hình ARIMA về vấn đề chuỗi thời gian của bạn.

Mô hình trung bình di chuyển tích hợp tự động

-Mô hình ARIMA là một lớp mô hình thống kê để phân tích và dự báo dữ liệu chuỗi thời gian.

ARIMA là một từ viết tắt viết tắt của AutoRegressive Integrated Moving Average. Đó là sự khái quát hóa của Trung bình Di chuyển Tự động Phân lớp đơn giản hơn và bổ sung khái niệm tích hợp.

Từ viết tắt này là mô tả, nắm bắt các khía cạnh chính của chính mô hình đó. Tóm lại, họ là:

AR: Autoregression. Một mô hình sử dụng mối quan hệ phụ thuộc giữa một quan sát và một số quan sát bị trễ.

I: Tích hợp Việc sử dụng sự khác biệt của các quan sát thô (ví dụ: trừ một quan sát từ một quan sát ở bước thời gian trước đó) để làm cho chuỗi thời gian dừng lại.

MA: Di chuyển trung bình. Một mô hình sử dụng sự phụ thuộc giữa một quan sát và một lỗi còn sót lại từ một mô hình trung bình động được áp dụng cho các quan sát bị trễ.

Mỗi thành phần này được xác định rõ ràng trong mô hình dưới dạng tham số. Một ký hiệu chuẩn được sử dụng của ARIMA (p, d, q), nơi các thông số được thay thế bằng các giá trị số nguyên để nhanh chóng chỉ ra mô hình ARIMA cụ thể đang được sử dụng.

Mỗi thành phần này được xác định rõ ràng trong mô hình dưới dạng tham số. Một ký hiệu chuẩn được sử dụng của ARIMA (p, d, q), nơi các thông số được thay thế bằng các giá trị số nguyên để nhanh chóng chỉ ra mô hình ARIMA cụ thể đang được sử dụng.

Các tham số của mô hình ARIMA được định nghĩa như sau:

p: Số quan sát trễ được bao gồm trong mô hình, còn được gọi là thứ tự trễ.

d: Số lần quan sát thô được phân biệt, cũng được gọi là mức độ khác biệt.

q: Kích thước của cửa sổ trung bình di chuyển, cũng được gọi là thứ tự trung bình động.

Mô hình hồi quy tuyến tính được xây dựng bao gồm số và loại thuật ngữ được chỉ định và dữ liệu được chuẩn bị theo mức độ khác biệt để làm cho nó tĩnh, tức là loại bỏ xu hướng và cấu trúc theo mùa ảnh hưởng tiêu cực đến mô hình hồi quy.

Giá trị 0 có thể được sử dụng cho tham số, cho biết không sử dụng phần tử đó của mô hình. Bằng cách này, mô hình ARIMA có thể được cấu hình để thực hiện chức năng của một mô hình ARMA, và thậm chí là một mô hình AR, I hoặc MA đơn giản.

Việc áp dụng mô hình ARIMA cho một chuỗi thời gian giả định rằng quá trình cơ bản tạo ra các quan sát là một quá trình ARIMA. Điều này có vẻ hiển nhiên, nhưng giúp thúc đẩy sự cần thiết phải xác nhận các giả định của mô hình trong các quan sát thô và trong các lỗi còn lại của các dự báo từ mô hình.

Tiếp theo, chúng ta hãy xem cách chúng ta có thể sử dụng mô hình ARIMA bằng Python. Chúng ta sẽ bắt đầu với việc tải một chuỗi thời gian đơn biến đơn giản.

**Bộ dữ liệu bán hàng dầu gội**

Tập dữ liệu này mô tả số lần bán dầu gội hàng tháng trong khoảng thời gian 3 năm.

Các đơn vị là một số lượng bán hàng và có 36 quan sát. Tập dữ liệu gốc được ghi nhận cho Makridakis, Wheelwright và Hyndman (1998).

Tải xuống tập dữ liệu và đặt nó vào thư mục làm việc hiện tại của bạn với tên tệp "shampoo-sales.csv".

Dưới đây là ví dụ về việc tải tập dữ liệu bán hàng dầu gội đầu bằng Pandas với chức năng tùy chỉnh để phân tích trường ngày giờ. Tập dữ liệu được baselined trong một năm tùy ý, trong trường hợp này là 1900.

ARIMA với Python

Thư viện mô hình thống kê cung cấp khả năng phù hợp với mô hình ARIMA.

Mô hình ARIMA có thể được tạo bằng cách sử dụng thư viện mô hình thống kê như sau:

-Xác định mô hình bằng cách gọi ARIMA () và truyền tham số p, d và q.

-Mô hình được chuẩn bị trên dữ liệu đào tạo bằng cách gọi hàm fit ()

-Dự đoán có thể được thực hiện bằng cách gọi hàm predict () và xác định chỉ mục thời gian hoặc thời gian được dự đoán.

Hãy bắt đầu với một cái gì đó đơn giản. Chúng tôi sẽ phù hợp với mô hình ARIMA cho toàn bộ tập dữ liệu bán hàng Dầu gội đầu và xem xét các lỗi còn lại.

Đầu tiên, chúng tôi fit với mô hình ARIMA (5,1,0). Điều này đặt giá trị độ trễ là 5 cho autoregression, sử dụng một thứ tự khác nhau của 1 để làm cho chuỗi thời gian tĩnh, và sử dụng một mô hình trung bình động là 0.

Khi lắp mô hình, rất nhiều thông tin gỡ lỗi được cung cấp về sự phù hợp của mô hình hồi quy tuyến tính. Chúng ta có thể tắt tính năng này bằng cách đặt đối số disp thành 0.

Chạy ví dụ in một bản tóm tắt của mô hình phù hợp. Điều này tóm tắt các giá trị hệ số được sử dụng cũng như kỹ năng phù hợp trên các quan sát trong mẫu.

Đầu tiên, nhận được một đường các chấm có lỗi, có thể một phần dữ liệu chưa đc đọc?

Tiếp đó là sơ đồ ko lấy số ấm.

**Mô hình dự báo ARIMA**

Mô hình ARIMA có thể được sử dụng để dự báo các bước thời gian trong tương lai.

Chúng ta có thể sử dụng hàm predict () trên đối tượng ARIMAResults để đưa ra các dự đoán. Nó chấp nhận chỉ mục của các bước thời gian để đưa ra các dự đoán làm đối số. Các chỉ mục này liên quan đến sự bắt đầu của tập dữ liệu đào tạo được sử dụng để đưa ra các dự đoán.

Nếu chúng ta sử dụng 100 quan sát trong tập dữ liệu huấn luyện để phù hợp với mô hình, thì chỉ số của bước thời gian tiếp theo để thực hiện dự đoán sẽ được chỉ định cho hàm dự đoán là start = 101, end = 101. Điều này sẽ trả về một mảng với một phần tử chứa dự đoán.

Chúng tôi cũng muốn các giá trị được dự báo ở quy mô ban đầu, trong trường hợp chúng tôi thực hiện bất kỳ sự khác biệt nào (d> 0 khi định cấu hình mô hình). Điều này có thể được chỉ định bằng cách đặt đối số nhập cho giá trị ‘levels’: typ = ’levels’.

Cách khác, chúng ta có thể tránh tất cả các đặc tả này bằng cách sử dụng hàm forecast (), thực hiện dự báo một bước bằng cách sử dụng mô hình.

Chúng tôi có thể chia tập dữ liệu đào tạo thành các tập huấn luyện và kiểm tra, sử dụng tập hợp đào tạo để phù hợp với mô hình và tạo dự đoán cho từng phần tử trên tập kiểm tra.

Một dự báo cán là cần thiết cho sự phụ thuộc vào các quan sát trong các bước thời gian trước cho sự khác biệt và mô hình AR. Một cách thô lỗ để thực hiện dự báo này là tái tạo mô hình ARIMA sau mỗi lần quan sát mới.

Chúng tôi theo dõi thủ công tất cả các quan sát trong danh sách được gọi là lịch sử được gieo giống với dữ liệu đào tạo và các quan sát mới được nối thêm vào mỗi lần lặp lại.

Đặt tất cả những thứ này lại với nhau, dưới đây là một ví dụ về dự báo cán với mô hình ARIMA bằng Python.

Chạy ví dụ in dự đoán và giá trị kỳ vọng mỗi lần lặp lại.

Chúng tôi cũng có thể tính toán điểm sai số bình phương cuối cùng (MSE) cho các dự đoán, cung cấp một điểm so sánh cho các cấu hình ARIMA khác.

Một biểu đồ đường thẳng được tạo hiển thị các giá trị dự kiến (màu xanh dương) so với các dự đoán dự báo cán (màu đỏ). Chúng ta có thể thấy các giá trị cho thấy một số xu hướng và đang ở quy mô chính xác.

Định cấu hình Mô hình ARIMA

Cách tiếp cận cổ điển để lắp một mô hình ARIMA là tuân theo phương pháp Box-Jenkins.

Đây là một quá trình sử dụng phân tích chuỗi thời gian và chẩn đoán để khám phá các thông số tốt cho mô hình ARIMA.

Tóm lại, các bước của quá trình này như sau:

- Nhận dạng mô hình. Sử dụng các lô và thống kê tóm tắt để xác định xu hướng, tính thời vụ và các yếu tố tự xác định để có được ý tưởng về số lượng sự khác biệt và kích thước của độ trễ sẽ được yêu cầu.

-Ước tính thông số. Sử dụng quy trình phù hợp để tìm các hệ số của mô hình hồi quy.

-Kiểm tra mô hình. Sử dụng lô và kiểm tra thống kê các lỗi còn lại để xác định số lượng và loại cấu trúc thời gian không được mô hình nắm bắt.

Quá trình này được lặp lại cho đến khi đạt được mức phù hợp mong muốn đối với các quan sát trong mẫu hoặc ngoài mẫu (ví dụ: tập dữ liệu đào tạo hoặc thử nghiệm).

Quá trình này được mô tả trong sách giáo khoa cổ điển năm 1970 về chủ đề có tựa đề Phân tích chuỗi thời gian: Dự báo và kiểm soát của George Box và Gwilym Jenkins. Một ấn bản thứ 5 được cập nhật hiện có sẵn nếu bạn quan tâm đến việc đi sâu vào loại mô hình và phương pháp này.

Do mô hình có thể phù hợp hiệu quả trên các tập dữ liệu chuỗi thời gian có kích thước khiêm tốn, các tham số tìm kiếm lưới của mô hình có thể là một cách tiếp cận có giá trị.

Tóm lược

Trong hướng dẫn này, bạn đã khám phá cách phát triển một mô hình ARIMA cho dự báo chuỗi thời gian bằng Python.

Cụ thể, bạn đã học:

Về mô hình ARIMA, làm thế nào nó có thể được cấu hình, và các giả định được thực hiện bởi mô hình.

Cách thực hiện phân tích chuỗi thời gian nhanh chóng bằng mô hình ARIMA.

Cách sử dụng mô hình ARIMA để dự báo các dự đoán mẫu.

Bạn có bất kỳ câu hỏi nào về ARIMA hoặc về hướng dẫn này không?

Đặt câu hỏi của bạn trong phần bình luận bên dưới và tôi sẽ cố hết sức để trả lời.