Báo Cáo Dự Án: Dự Đoán Khách Hàng Rời Mạng Viễn Thông

* 7. Mô hình phân lớp
* 7.1. Các mô hình đã thử

Trong phần này, nhiều mô hình phân lớp đã được thử nghiệm:

* Decision Tree
* là một trong những thuật toán phân loại đơn giản và dễ hiểu nhất, thường được sử dụng để giải quyết các bài toán phân loại và hồi quy. Trong dự án này, Decision Tree được sử dụng để phân loại xem khách hàng có rời bỏ dịch vụ hay không dựa trên các đặc điểm của họ
* Decision Tree rất trực quan và dễ dàng diễn giải. Các quy tắc phân lớp rõ ràng và có thể được biểu diễn dưới dạng biểu đồ cây, giúp các nhà quản lý dễ dàng hiểu được lý do đằng sau dự đoán của mô hình, Decision Tree có thể xử lý dữ liệu thiếu giá trị mà không cần chuẩn bị nhiều, không yêu cầu dữ liệu phải chuẩn hóa hoặc chuyển đổi như một số mô hình khác, ví dụ như mô hình tuyến tính, tốt trong việc nắm bắt các mối quan hệ phi tuyến tính giữa các biến.
* Tuy nhiên, Decision Tree có xu hướng phức tạp quá mức và ghi nhớ các chi tiết nhỏ không cần thiết từ dữ liệu huấn luyện, đặc biệt là khi độ sâu của cây không được kiểm soát, một thay đổi nhỏ trong dữ liệu có thể dẫn đến một cấu trúc cây hoàn toàn khác nhau. Khi làm việc với tập dữ liệu lớn, Decision Tree có thể trở nên cồng kềnh và khó xử lý, mô hình có thể bị thiên vị đối với lớp chiếm đa số trừ khi có các kỹ thuật cân bằng hoặc đánh trọng số.
* Kết quả của mô hình Decision Tree Classifier cho thấy một hiệu suất tổng thể khá tốt với độ chính xác 81.53%. Tuy nhiên, khi xem xét kỹ hơn, ta thấy có sự chênh lệch đáng kể giữa hiệu suất của hai lớp. Lớp đa số (lớp 0) có hiệu suất tốt với precision, recall và f1-score đều trên 0.86, trong khi lớp thiểu số (lớp 1) có hiệu suất thấp hơn đáng kể, đặc biệt là về precision (0.62). Điều này phản ánh một trong những hạn chế của Decision Tree khi đối mặt với dữ liệu không cân bằng. Cây quyết định có xu hướng ưu tiên các đặc trưng phổ biến, dẫn đến việc dự đoán tốt hơn cho lớp đa số nhưng kém chính xác hơn cho lớp thiểu số. Tuy nhiên, điểm đáng chú ý là recall của lớp thiểu số (0.70) vẫn ở mức khá, cho thấy mô hình có khả năng nhận diện được phần lớn các mẫu thuộc lớp này. Sự chênh lệch giữa precision và recall của lớp thiểu số gợi ý rằng mô hình có xu hướng phân loại nhiều mẫu vào lớp thiểu số, dẫn đến tỷ lệ dương tính giả cao. Nguyên nhân có thể là do overfitting, thiếu tinh chỉnh hyperparameter.
* KNN
* là một thuật toán học máy đơn giản và hiệu quả cho các bài toán phân loại và hồi quy. KNN hoạt động dựa trên nguyên lý tìm kiếm các điểm dữ liệu lân cận gần nhất trong không gian đặc điểm để dự đoán đầu ra cho một điểm dữ liệu mới
* KNN rất dễ cài đặt và trực quan, vì nó chỉ đơn giản phân loại dựa trên điểm lân cận gần nhất, không có quá trình huấn luyện tốn thời gian, do đó KNN có thể dễ dàng mở rộng cho các dữ liệu mới. Có thể được sử dụng cho cả phân loại và hồi quy mà không cần điều chỉnh mô hình, KNN có thể nắm bắt tốt các mối quan hệ phi tuyến tính giữa các biến.
* Tuy nhiên, KNN yêu cầu tính toán khoảng cách cho mỗi điểm dữ liệu, điều này có thể rất tốn thời gian với các tập dữ liệu lớn, Dễ bị ảnh hưởng bởi nhiễu và outliers trong dữ liệu, do mỗi điểm dữ liệu đều có thể ảnh hưởng đến dự đoán, vì tất cả dữ liệu cần được lưu trữ để tính toán khoảng cách. KNN cần nhiều tài nguyên bộ nhớ, khi số lượng đặc điểm tăng cao, KNN có thể trở nên kém hiệu quả do hiện tượng "lời nguyền của kích thước"
* Để cải thiện các nhược điểm của KNN thì chúng ta nên sử dụng thêm các lựa chọn tối ưu như: chọn số lánh giềng tối ưu(k), luôn chuẩn hóa dữ liệu để tránh ảnh hưởng của các đặc điểm có thang đo lớn, giảm kích thước dữ liệu, **sử dụng các biến thể của KNN**: Ví dụ, **Weighted KNN** có thể cải thiện hiệu suất bằng cách gán trọng số cho các láng giềng dựa trên khoảng cách của chúng
* Kết quả của mô hình KNN với tập dữ liệu đã cho cho thấy một độ chính xác tổng thể ở mức trung bình với 74.66%. Điều đáng chú ý là sự khác biệt rõ rệt giữa hiệu suất của hai lớp. Lớp đa số (lớp 0) có precision rất cao (0.95) nhưng recall thấp hơn (0.70), trong khi lớp thiểu số (lớp 1) có xu hướng ngược lại với recall cao (0.89) nhưng precision thấp (0.50). Điều này cho thấy mô hình KNN có xu hướng phân loại nhiều mẫu vào lớp thiểu số, dẫn đến tỷ lệ dương tính giả cao cho lớp này. Đặc điểm này của KNN có thể là do việc chọn k = 45, một giá trị khá lớn so với kích thước của lớp thiểu số (387 mẫu). Với k lớn, mô hình có xu hướng ưu tiên lớp đa số trong không gian đặc trưng, nhưng đồng thời cũng "bắt" được nhiều mẫu thuộc lớp thiểu số. Điều này giải thích cho recall cao của lớp thiểu số nhưng precision thấp. Ngược lại, lớp đa số có precision cao nhưng recall thấp hơn, cho thấy khi mô hình dự đoán một mẫu thuộc lớp đa số, nó thường chính xác, nhưng có nhiều mẫu thực sự thuộc lớp đa số bị phân loại nhầm sang lớp thiểu số.
* SVM
* là một thuật toán học máy mạnh mẽ được sử dụng cho các bài toán phân loại và hồi quy. SVM hoạt động bằng cách tìm kiếm một siêu phẳng tối ưu trong không gian đặc điểm để phân tách các lớp dữ liệu một cách rõ ràng nhất
* SVM thường cho hiệu suất cao với các bài toán phân loại phức tạp, đặc biệt khi dữ liệu có biên tách rõ ràng giữa các lớp, với việc sử dụng kernel, SVM có thể mô hình hóa các mối quan hệ phi tuyến phức tạp giữa các đặc điểm. SVM ít bị ảnh hưởng bởi nhiễu hơn so với các mô hình khác, đặc biệt với giá trị tham số ‘C’ được lựa chọn phù hợp, Khả năng tổng quát hóa tốt hơn trong không gian đặc điểm cao, làm giảm nguy cơ overfitting so với các mô hình khác.
* Với tập dữ liệu khách hàng đã cho và sử dụng thuật toán SVN thì ta thấy SVM có thể mất nhiều thời gian huấn luyện, đặc biệt với các tập dữ liệu lớn và không cân bằng. Mô hình SVM không dễ diễn giải như các mô hình cây quyết định hay hồi quy tuyến tính. Yêu cầu dữ liệu phải được chuẩn hóa để hoạt động hiệu quả.
* Để tối ưu hóa SVN với lượng dữ liệu lớn thì ta cần chọn kernel phù hợp, sử dụng kỹ thuật giảm kích thước như PCA để giảm số lượng đặc điểm và tăng hiệu quả của SVM, sử dụng các kỹ thuật như SMOTE để tạo dữ liệu mẫu cho các lớp chiếm thiểu số.
* Kết quả của mô hình SVC cho thấy một hiệu suất tổng thể khá tốt với độ chính xác 82.76%. Đây là một cải thiện đáng kể so với mô hình KNN trước đó và cũng nhỉnh hơn một chút so với Decision Tree. Khi xem xét chi tiết, ta thấy có sự chênh lệch giữa hiệu suất của hai lớp, nhưng không quá lớn như trong trường hợp của KNN. Lớp đa số (lớp 0) có precision rất cao (0.92) và recall khá tốt (0.84), cho thấy mô hình rất hiệu quả trong việc nhận diện và phân loại chính xác các mẫu thuộc lớp này. Đối với lớp thiểu số (lớp 1), mô hình đạt được recall khá cao (0.78), cao hơn so với Decision Tree, nhưng precision vẫn ở mức thấp (0.62), tương tự như Decision Tree. Điều này cho thấy SVC có khả năng phát hiện tốt các mẫu thuộc lớp thiểu số, nhưng vẫn có xu hướng phân loại nhầm một số mẫu của lớp đa số vào lớp thiểu số. Sự cân bằng tốt hơn giữa precision và recall của cả hai lớp đã dẫn đến f1-score cao hơn so với các mô hình trước đó, đặc biệt là đối với lớp thiểu số (0.69). Điều này phản ánh một trong những ưu điểm của SVC trong việc xử lý dữ liệu không cân bằng, có thể là nhờ khả năng tạo ra một đường biên quyết định phức tạp và linh hoạt trong không gian đặc trưng.
* Logistic Regression
* là một thuật toán học máy phổ biến dùng cho các bài toán phân loại nhị phân. Mặc dù tên gọi có chứa "hồi quy", đây thực sự là một kỹ thuật phân loại. Logistic Regression dựa trên hàm logistic (hay hàm sigmoid) để dự đoán xác suất một mẫu thuộc về một lớp cụ thể.
* Logistic Regression rất dễ hiểu và giải thích, với trọng số đặc điểm biểu diễn tầm quan trọng và mối quan hệ với kết quả dự đoán. Đối với các tập dữ liệu lớn, Logistic Regression có thể được huấn luyện nhanh chóng và hiệu quả. Hoạt động tốt khi dữ liệu có mối quan hệ tuyến tính giữa các đặc điểm và biến mục tiêu. Ít bị ảnh hưởng bởi vấn đề overfitting nếu dữ liệu được chuẩn bị và kiểm soát tốt.
* Nhược điểm của Logistic Regression là cần một lượng dữ liệu đủ lớn để mô hình hóa hiệu quả các đặc điểm và đưa ra dự đoán chính xác. Logistic Regression có thể bị ảnh hưởng bởi outliers, làm thay đổi mô hình đáng kể. Không phù hợp cho dữ liệu có mối quan hệ phi tuyến mạnh giữa các đặc điểm
* Khi quyết định sử dụng Logistic Regression thì chúng ta cần lưu ý chọn các đặc điểm phù hợp, sử dụng regularization (L1 hoặc L2) để giảm overfitting, với L1 giúp lựa chọn đặc điểm và L2 giảm thiểu đa cộng tuyến. Tạo ra các đặc điểm phi tuyến hoặc tương tác để cải thiện khả năng phân loại khi cần thiết.
* Kết quả của mô hình Hồi quy Logistic cho thấy một hiệu suất tổng thể khá tốt với độ chính xác 80.23%. Mặc dù không cao bằng SVC, nhưng vẫn tốt hơn KNN và gần với Decision Tree. Khi xem xét chi tiết, ta thấy có sự khác biệt đáng kể giữa hiệu suất của hai lớp. Lớp đa số (lớp 0) có precision rất cao (0.94), cao nhất trong số các mô hình đã xem xét, nhưng recall thấp hơn (0.78). Điều này cho thấy khi mô hình dự đoán một mẫu thuộc lớp 0, nó rất có khả năng chính xác, nhưng mô hình cũng bỏ sót một số mẫu thực sự thuộc lớp này. Đối với lớp thiểu số (lớp 1), ta thấy một xu hướng ngược lại: recall cao (0.86), thậm chí cao nhất trong số các mô hình đã xem xét, nhưng precision thấp (0.57). Điều này ngụ ý rằng mô hình có khả năng phát hiện tốt các mẫu thuộc lớp thiểu số, nhưng cũng có xu hướng phân loại nhầm nhiều mẫu của lớp đa số vào lớp thiểu số. Sự chênh lệch lớn giữa precision và recall của cả hai lớp cho thấy mô hình Hồi quy Logistic đang gặp khó khăn trong việc cân bằng giữa hai lớp không cân bằng. Điều này có thể là do bản chất tuyến tính của mô hình Hồi quy Logistic, khiến nó khó tạo ra một đường biên quyết định phức tạp để phân tách tốt hai lớp trong không gian đặc trưng. Tuy nhiên, f1-score của lớp thiểu số (0.68) vẫn khá tốt, chỉ thấp hơn một chút so với SVC.
* Random Forest Classifier
* là một thuật toán học máy mạnh mẽ và linh hoạt dựa trên việc kết hợp nhiều cây quyết định (Decision Trees) để cải thiện độ chính xác và tránh overfitting. Random Forest hoạt động bằng cách xây dựng một "rừng" ngẫu nhiên từ nhiều cây quyết định, mỗi cây được huấn luyện trên một mẫu ngẫu nhiên của tập dữ liệu.
* Random Forest giảm thiểu vấn đề overfitting thường gặp ở cây quyết định đơn lẻ bằng cách trung bình hóa dự đoán từ nhiều cây. Thường cho kết quả tốt trên nhiều loại dữ liệu và bài toán phân loại khác nhau. Random Forest có thể xử lý tốt cả dữ liệu phân loại và hồi quy, đồng thời không cần chuẩn hóa dữ liệu. Mô hình cung cấp thông tin về tầm quan trọng của các đặc điểm trong việc dự đoán kết quả, giúp cho việc phân tích và lựa chọn đặc điểm tốt hơn.
* Với các tập dữ liệu lớn, thời gian huấn luyện của Random Forest có thể rất lâu. Kết quả từ một rừng cây khó diễn giải và trực quan hơn so với một cây quyết định đơn lẻ, yêu cầu tài nguyên tính toán và bộ nhớ lớn hơn so với các mô hình đơn giản như cây quyết định hoặc hồi quy logistic.
* Và để tối ưu hóa thì trong quá trình sử dụng chúng ta nên sử dụng thêm các kỹ thuật như SMOTE hoặc điều chỉnh trọng số lớp để đảm bảo rằng dữ liệu không bị thiên lệch. Sử dụng phương pháp bagging (bootstrap aggregating) để cải thiện độ chính xác và độ ổn định của mô hình. Nếu dữ liệu quá lớn, có thể giảm kích thước dữ liệu bằng cách loại bỏ các đặc điểm không quan trọng hoặc sử dụng các kỹ thuật giảm chiều như PCA.
* Kết quả của mô hình Random Forest cho thấy hiệu suất tổng thể rất tốt với độ chính xác 85.55%, cao nhất trong số các phương pháp đã xem xét. Điều này phản ánh một trong những ưu điểm chính của Random Forest - khả năng xử lý hiệu quả các bộ dữ liệu phức tạp và không cân bằng. Khi phân tích chi tiết, ta thấy có sự cân bằng tốt giữa hiệu suất của hai lớp, điều mà các phương pháp khác chưa đạt được. Đối với lớp đa số (lớp 0), mô hình đạt được cả precision (0.90) và recall (0.91) cao, dẫn đến f1-score rất tốt (0.90). Điều này cho thấy Random Forest rất hiệu quả trong việc nhận diện và phân loại chính xác các mẫu thuộc lớp đa số, với rất ít trường hợp bỏ sót hoặc phân loại sai. Đối với lớp thiểu số (lớp 1), mặc dù hiệu suất thấp hơn so với lớp đa số, nhưng vẫn đạt được sự cân bằng tốt giữa precision (0.72) và recall (0.69), dẫn đến f1-score khá cao (0.71). Đây là f1-score cao nhất cho lớp thiểu số trong số các phương pháp đã xem xét, cho thấy Random Forest xử lý tốt vấn đề không cân bằng dữ liệu. Sự cân bằng tốt giữa precision và recall của cả hai lớp phản ánh một trong những ưu điểm chính của Random Forest: khả năng tạo ra một tập hợp các cây quyết định đa dạng, giúp mô hình có thể nắm bắt được các mẫu phức tạp trong dữ liệu. Điều này đặc biệt hữu ích trong trường hợp dữ liệu không cân bằng, nơi một số cây có thể chuyên biệt hóa trong việc nhận diện các mẫu thuộc lớp thiểu số.
* 7.2. Mô hình được chọn

Dựa trên các chỉ số đánh giá mô hình như độ chính xác, độ nhạy, độ chính xác, và điểm F1, mô hình **Random Forest** được chọn vì:

* **Độ chính xác cao**: Random Forest đạt độ chính xác cao nhất trong số các mô hình thử nghiệm.
* **Tính ổn định**: Mô hình ít bị ảnh hưởng bởi outliers và noise trong dữ liệu.
* **Khả năng diễn giải**: Cung cấp các chỉ số tầm quan trọng của đặc điểm, giúp hiểu rõ yếu tố nào ảnh hưởng đến quyết định rời bỏ của khách hàng.
* 7.3. Đánh giá tổng quan

Mô hình Random Forest cho thấy hiệu suất tốt với các chỉ số chính:

* **Độ chính xác**: Xấp xỉ 85,5%, cho thấy mô hình phân loại tốt.
* **Ma trận nhầm lẫn**: Cho thấy tỷ lệ dự đoán đúng cao giữa các lớp.
* **Điểm F1**: Cân bằng giữa độ chính xác và độ nhạy, đặc biệt quan trọng trong bối cảnh dữ liệu không cân bằng.
* 8. Học sâu với mô hình mạng nơ-ron nhân tạo ANN(Artificial Neural Network)
* 8.1. Giới thiệu về mô hình ANN

Mạng nơ-ron nhân tạo (ANN) là một công cụ mạnh mẽ trong lĩnh vực học máy, đặc biệt phù hợp cho các bài toán phân loại và dự đoán. ANN mô phỏng hoạt động của não người thông qua các lớp nơ-ron, giúp nhận diện và xử lý các mẫu dữ liệu phức tạp.

* 8.2. Cấu trúc của mô hình

Mô hình ANN được xây dựng bao gồm các thành phần sau:

* Lớp đầu vào(input layer): Xác định kích thước đầu vào dựa trên số lượng đặc trưng (features) của dữ liệu. Trong mô hình này, số lượng nơ-ron đầu vào tương ứng với số lượng đặc trưng trong tập dữ liệu sau khi được chuẩn hóa.
* **Các lớp ẩn (Hidden Layers):**
* Lớp ẩn đầu tiên chứa 64 nơ-ron với hàm kích hoạt ReLU (Rectified Linear Unit), giúp mô hình học các đặc trưng phi tuyến tính.
* Lớp ẩn thứ hai chứa 32 nơ-ron, cũng sử dụng hàm kích hoạt ReLU. Việc giảm số lượng nơ-ron giúp giảm thiểu tình trạng quá khớp (overfitting).
* Mỗi lớp ẩn đều có thêm một lớp Dropout với tỷ lệ 0.5, nhằm ngăn chặn quá khớp bằng cách loại bỏ ngẫu nhiên một số nơ-ron trong quá trình huấn luyện.
* **Lớp đầu ra(output layer):** sử dụng một nơ-ron duy nhất với hàm kích hoạt sigmoid, do bài toán cần giải quyết là phân loại nhị phân (binary classification).
* Biên dịch mô hình
* Sử dụng hàm mất mát binary\_crossentropy cho bài toán phân loại nhị phân.
* Bộ tối ưu hóa adam được sử dụng để cập nhật trọng số của mô hình nhằm giảm thiểu hàm mất mát.
* Đánh giá mô hình bằng độ chính xác (accuracy).
* 8.3. Đánh giá và kết luận
* Quá trình huấn luyện: Mô hình được huấn luyện qua 50 epoch với batch size là 32.
* Độ chính xác (Accuracy): Độ chính xác trên tập huấn luyện tăng dần từ 60% đến hơn 85%, trong khi độ chính xác trên tập kiểm tra đạt khoảng 82-83%.
* Hàm mất mát (Loss): Giá trị loss giảm dần trong quá trình huấn luyện, cho thấy mô hình đang học và cải thiện. Tuy nhiên, loss trên tập kiểm tra có xu hướng ổn định và có thể tăng nhẹ, chỉ ra dấu hiệu của overfitting.
* Đánh giá chi tiết: Độ chính xác của mô hình trên tập kiểm tra là khoảng 81%, với precision cao hơn ở lớp 0 so với lớp 1. Điều này cho thấy mô hình hoạt động tốt hơn trong việc dự đoán các mẫu thuộc lớp 0​(12)​.
* Accuracy và Loss trong tập huấn luyện và tập kiểm tra: Trong quá trình huấn luyện, độ chính xác (accuracy) trên tập huấn luyện tăng dần từ khoảng 60% lên hơn 85%. Độ chính xác trên tập kiểm tra cũng tăng dần, đạt đến khoảng 82-83%. Loss (mất mát) giảm dần, cho thấy mô hình đang học và cải thiện qua từng epoch. Validation Accuracy và Loss: Độ chính xác trên tập kiểm tra dao động quanh mức 82-83%. Loss trên tập kiểm tra giảm dần, nhưng sau đó có xu hướng ổn định và thậm chí có tăng nhẹ.
* Accuracy trên tập kiểm tra là 83.71%. Loss trên tập kiểm tra là 0.3748. Validation Accuracy là: 83.09%
* Độ chính xác của mô hình trên tập huấn luyện và tập kiểm tra đều cao, cho thấy mô hình học tốt từ dữ liệu và có khả năng tổng quát tốt. Độ chính xác trên tập kiểm tra khá ổn định qua các epoch cuối cùng, điều này cho thấy mô hình không bị overfitting quá nhiều. Loss trên tập kiểm tra có xu hướng ổn định sau một thời gian giảm, điều này cho thấy mô hình đã đạt đến mức tối ưu nhất định.