**Phân loại khách hàng có tiềm năng phản hồi tích cực với các chiến dịch Marketing**

Nguyễn Hoàng Long - 19521788, Phan Phạm Quỳnh Hoa – 19521520,

Chung Khánh Gia - 19521442

**1. Abstract**

Phân tích hành vi khách hàng qua các chiến dịch Marketing là một bài toán thiết yếu giúp doanh nghiệp đưa ra được những quyết định đúng đắn, phân loại được tập khách hàng có khả năng phản hồi tích cực với các chiến dịch Marketing trong tương lai, từ đó giúp tối ưu hóa chi phí và lợi nhuận. Trong bài báo cáo này, nhóm thực hiện xử lý dữ liệu thô, đánh giá và so sánh kết quả trên ba mô hình máy học phổ biến Logistic Regression [1], Decision Trees [2] và một giải thuật được dựa trên gradient boosting (XGBoost [3]). Đồng thời kết hợp sử dụng GridSearchCV để chọn ra được bộ tham số cho kết quả tốt nhất.

**2. Introduction**

Bên cạnh tốc độ phát triển vượt bậc về công nghệ hiện nay, nhiều lĩnh vực kinh tế được thúc đẩy mạnh, thị trường dần xuất hiện đa dạng các doanh nghiệp, cá nhân với quy mô lớn nhỏ khác nhau. Cùng với đó, nhiều thử thách được đặt ra với các doanh nghiệp hoạt động trong lĩnh vực buôn bán và thương mại, đặc biệt nhất có thể kể tới là việc dẫn đầu thị trường tới từ việc đẩy mạnh Marketing trên nhiều nền tảng khác nhau. Từ đó, phân tích hành vi khách hàng qua các chiến dịch Marketing là bài toán đóng vai trò quan trọng trong việc thúc đẩy doanh nghiệp giảm thiểu được chi phí, thu hẹp phạm vi Marketing và lựa chọn được tập khách hàng tiềm năng nhằm tối ưu hóa lợi nhuận.

Trong báo cáo này, nhóm thực hiện phân tích và thử nghiệm trên bộ dữ liệu Marketing Analytics. Tập dữ liệu chứa 2240 samples, bao gồm những thuộc tính về thói quen khi mua

hàng, thông tin của mỗi khách hàng và phản hồi của khách hàng khi được tiếp cận các chiến dịch Marketing sản phẩm mà công ty đưa ra.

Thực hiện huấn luyện, đánh giá trên hai mô hình phổ biến cho các bài toán phân loại là Logistic Regression và Decision Trees. Đồng thời thử nghiệm XGBoost (Extreme Gradient Boosting), một giải thuật được dựa trên gradient boosting đã giành giải cao trong các cuộc thi trên Kaggle vì tính đơn giản và hiệu quả của nó cho nhóm bài toán về classification và cả regression.

Mở rộng hơn, (1) vì tập dữ liệu tìm được tương đối ít, nhóm thực hiện sử dụng k-fold cross validation để tránh việc overfitting có thể xảy ra khi phân chia kích thước tập validation quá nhỏ. (2) Fine-tune model sử dụng GridSearchCV nhằm tìm ra hyper-parameter tối ưu nhất

Nhóm thực hiện public toàn bộ source code cài đặt và thử nghiệm tại: [https://github.com/ nghoanglong/DS102-M11-CNCL](https://github.com/nghoanglong/DS102-M11-CNCL)

**3. Models**

**3.1 Logistic Regression:** là mô hình phân loại phổ biến được dùng để gán các đối tượng cho 1 tập hợp giá trị rời rạc.Logistic Regression được biến đổi một chút từ Linear Regression (hồi quy tuyến tính đa biến), bằng cách đặt kết quả vào một activation function phi tuyến, mà đầu ra được thể hiện dưới dạng xác suất mỗi lớp. Bài toán ta có thể xét một cách tổng quát với hàm sigmoid, công thức tính xác suất đầu ra lúc này có dạng:

P( = = σ(

với **a = x,** và nhiệm vụ ta cần tối thiểu hàm mất mát sau:

J(w) =

Công thức cập nhật tương tự bài toán Linear Regression sử dụng Gradient Descent:

**3.2 Decision Trees:** là một mô hình supervised learning, có thể được áp dụng vào cả hai bài toán classification và regression. Việc xây dựng một decision tree trên dữ liệu huấn luyện cho trước là việc đi xác định các câu hỏi và thứ tự của chúng. Một điểm đáng lưu ý của decision tree là nó có thể làm việc với các đặc trưng (trong các tài liệu về decision tree, các đặc trưng thường được gọi là thuộc tính – attribute) dạng categorical, thường là rời rạc và không có thứ tự. Decision tree cũng làm việc với dữ liệu có vector đặc trưng bao gồm cả thuộc tính dạng categorical và liên tục (numeric). Một điểm đáng lưu ý nữa là decision tree ít yêu cầu việc chuẩn hoá dữ liệu.

Điểm mạnh của thuật toán Decision tree:

* Có thể thẩm định một mô hình bằng các kiểm tra thống kê.
* Cây quyết định có thể xử lý tốt một lượng dữ liệu lớn trong thời gian ngắn.
* Cây quyết định giống cách tiếp cận ra quyết định của con người.
* Cây quyết định dễ giải thích. Kết quả của nó là tập các luật.
* Dễ trực quan hóa

Các độ đo:

* Gini impurity – Gini index: CART
* Information gain: ID3, C4.5, C5.0
* Variance reduction: CART

**3.3 XGBoost(Extreme Gradient Boosting):** là một giải thuật được base trên Gradient Boosting, thuộc nhóm Boosting – Ensemble Learning bao gồm: (1) chuỗi các model yếu (weak learners) học bổ sung cho nhau. (2) Thay vì đi tìm nghiệm tối ưu toàn cục, nhóm mô hình Boosting cố gắng đi tìm các nghiệm cục bộ của từng mô hình trong chuỗi với mong muốn dần đi đến nghiệm toàn cục với bài toán tối ưu như sau:

với

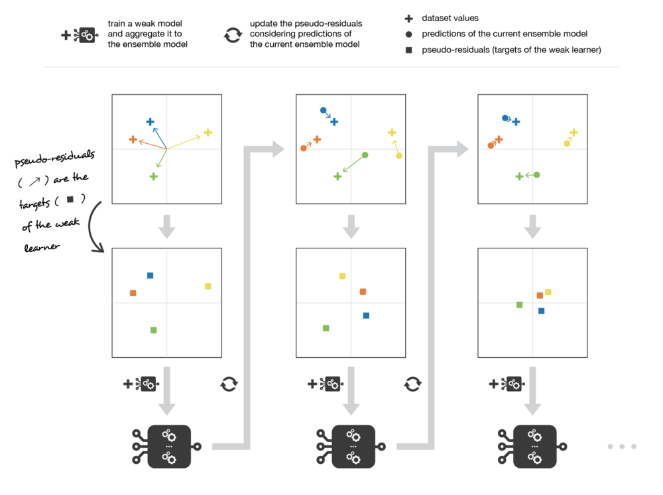
Trong đó:

* L: Loss function
* y: label
* : confidence score của weak learner thứ n (hay còn gọi là trọng số)
* : weak learner thứ n

Liên hệ với công thức cập nhật tham số mô hình theo Gradient Descent

nếu chúng ta coi chuỗi các weak learners là một hàm số *W,* thì mỗi learner có thể coi là một hàm số *w.* Đến đây, cực tiểu hóa hàm loss function **L(y, W)** áp dụng Gradient Descent

ta thấy được mối liên hệ như sau:

với là learner kế tiếp. Khi đó, model kế tiếp cần học để fit vào giá trị bên vế phải, giá trị bên vế phải còn có tên gọi khác là **pseudo-residuals**

*Hình 1: Minh họa quá trình học với Gradient Boosting*

**4. Experimental Setup and Evaluation Metrics**

Nhóm chúng tôi thực hiện bước xử lý dữ liệu đầu tiên cụ thể như sau: fetch file dataset và lưu vào đường dẫn cố định trên google drive, xem thông tin dữ liệu, đổi tên features phù hợp, chuyển đổi kiểu dữ liệu, sau đó kiểm tra và tiến hành drop những dữ liệu outliners, kiểm tra dữ liệu Null và thực hiện fill những giá trị Null bằng một giá trị trung bình, sử dụng correlation matrix với mức threshold >= 0.14 để chọn ra những features cần thiết, đưa những features dạng chuỗi về dạng phù hợp sử dụng OneHotEncoder và cuối cùng chuẩn hóa những features dạng số bằng StandardScaler

Sau khi hoàn thành bước xử lý dữ liệu, nhóm tiến hành training trên ba mô hình đã đề cập và sử dụng GridSearchCV để tìm ra bộ tham số tối ưu

Với mô hình Logistic Regression, nhóm cấu hình tham số C=0.01 để tăng khả năng tránh overfitting sử dụng regulization, solver LIBLINEAR [4], kết quả thử nghiệm được liệt kê ở bảng 1.

Với mô hình Decision Trees Classifier, nhóm cấu hình tham số độ đo “tính thuần khiết” sử dụng entropy thay vì gini bằng cách thiết lập criterion=‘entropy’, cấu hình chiều cao cây bằng tham số max\_depth=6, giảm mới max\_depth có thể giúp mô hình tránh overfitting, kết quả thử nghiệm được liệt kê ở bảng 1.

Với mô hình XGBoost, nhóm cấu hình số lượng estimators=200, độ sâu mỗi gradient boosting tree với tham số max\_depth=3, learning\_rate=0.1, kết quả thử nghiệm được liệt kê ở bảng 1.

Kết quả thử nghiệm được đánh giá qua 4 độ đo phổ biến cho nhóm bài toán classification: Accuracy, Precision, Recall, F1-Score

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approach** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| **Logistic Regression** | 92.2 | 89.2 | 54.8 | 67.7 |
| **Decision Trees** | 97.6 | 92.6 | 91.6 | 91.7 |
| **Decision Trees** | 99.4 | 96.6 | 99.4 | 98 |

*Bảng 1: Đánh giá mô hình qua các độ đo*

Bảng 1 cho ta thấy được khi sử dụng mô hình Logistic Regression thì accuracy ≈ 92.2% và nó được cải thiện lần lượt khi sử dụng Decision Trees ≈ 97.6% (92.2% → 97.6% ≈ 5.4%), XGBoost ≈ 99.4% (92.2% → 99.4% ≈ 7.2%). Các thông số khác như Precision, Recall và F1-Score cũng lần lượt được cải thiện, về mặt tổng quan ta có thể thấy sử dụng XGBoost cho chúng ta kết quả tốt nhất trên bộ dữ liệu này

**5. Conclusion**

Trong bài báo cáo này, nhóm thực hiện đã trình bày bài toán, những mô hình được sử dụng và phần mở rộng để cải thiện kết quả. Trong quá trình nghiên cứu, thử nghiệm nhóm thực hiện cũng gặp nhiều bất lợi như lượng dữ liệu tương đối hạn chế, mô hình và bài toán đơn giản. Hiệu suất trên các mô hình tốt, không tốn nhiều tài nguyên, bài toán tuy đơn giản nhưng sát với thực tế và là vấn đề thiết yếu ở nhiều công ty thương mại lớn hiện nay trên cả nước. Nhóm thực hiện mong rằng bài báo cáo này có thể phục vụ cho việc bắt đầu nghiên cứu sâu rộng và cải thiện hiệu suất bài toán trên các tập dữ liệu lớn.

**6. References**

[1] Hosmer, David W., and Stanley Lemeshow. Applied Logistic Regression. 1989.

[2] Quinlan, J. R. “Induction of Decision Trees.” Machine Learning, vol. 1, no. 1, 1986, pp. 81–106.

[3] Chen, Tianqi, and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[4] Fan, Rong-En, et al. “LIBLINEAR: A Library for Large Linear Classification.” Journal of Machine Learning Research, vol. 9, no. 61, 2008, pp. 1871–1874.