BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**BÁO CÁO**

**THỰC HÀNH NHẬP MÔN TIN HỌC**

Sinh viên trình bày: Họ tên sinh viên

Giảng viên hướng dẫn: Ths Nguyễn Thị Phương Trang

**TP. Hồ Chí Minh, tháng 12 năm 2020**

# GIỚI THIỆU

## Giới thiệu đề tài

## Bố cục bài báo cáo

Bài báo cáo được chia làm năm chương:

Ở chương 1 sẽ là phần giới thiệu đề tài khái quát về các nội dung, mục tiêu và phương pháp nghiên cứu.

Chương 2 là cơ sở lý thuyết để nghiên cứu. Ở chương này các khái niệm về các mô hình máy học và các mô hình làm lại mẫu sẽ được nêu rõ.

Chương 3 trình bày về giới thiệu mô hình.

Chương 4 trình bày về kết quả thực nghiệm

Chương cuối cùng là phần kết luận, ở đây sẽ là phần đánh giá về phương pháp, kết quả thực nghiệm, nêu lên những ưu, khuyết điềm cùng với hướng giải quyết cho tương lai.

# CƠ SỞ LÝ THUYẾT

## Linear Regression

Linear regression (hồi quy tuyến tính) là một thuật toán học có giám sát (supervised learning). Mục đích của thuật toán là từ một vector (đầu vào) sẽ dự đoán được một giá trị vô hướng (đầu ra). Trong hồi quy tuyến tính thì đầu ra đó sẽ là một hàm tuyến tính của đầu vào. Đặt là giá trị dự đoán của mô hình ta có:

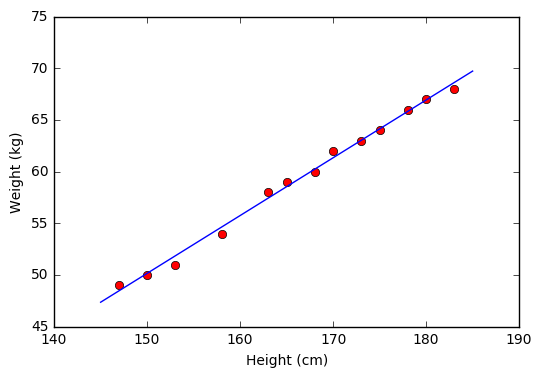
(1)

Trong đó:

là một vector tham số (trọng số)

là sự chênh lệch giữa giá trị đầu ra (y) và giá trị dự đoán ()

Như vậy, giá trị dự đoán là chính xác nhất thì sẽ đạt giá trị nhỏ nhất. Giá trị dự đoán nằm trong khoàng (

Hồi tuyến tính thường được sử dụng trong các bài toàn hồi quy như: dự doán giá nhà, dự đoán cân nặng, ….

*Biểu đồ quan hệ giữa cân nặng và chiều cao và đường dự đoán [1]*

1Hình 2.1-1

## Logistic Regression

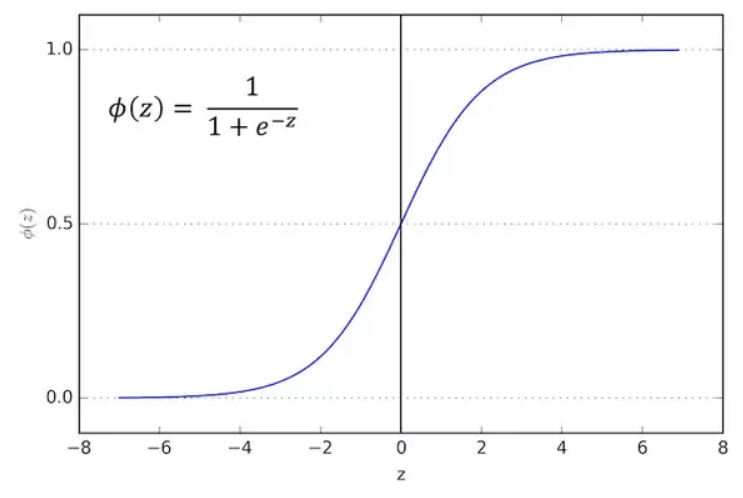
Như đã thấy mô hình hồi quy tuyến tính có giá trị đầu ra nằm trong khoàng (, thì Logistic Regression lại có giá trị đầu ra nằm trong khoảng [0,1]. Mục đích của việc này là để giá trị đầu ra của mô hình là một xác suất. Để thực hiện được việc này, giá trị dự đoán của mô hình hồi quy tuyến tính sẽ là đầu vào của hàm Sigmoid *(Hình 2.2):*

(2)

Trong dó:

là giá trị dự đoán của mô hình hồi quy tuyến tính

là hàm sigmoid và có khoảng giá trị từ (0,1)

*Đồ thì hàm sigmoid [2]*

2Hình 2.2-1

Vì có miền giá trị đầu ra nằm trong khoảng [0,1] nên Logistic regression thường được sử dụng trong các bài toài phân lớp như: dự đoán rủi ro tín dụng, dự đoán phân loại bệnh nhân.

# Giới thiệu mô hình

## Tổng quan

Như một xu hướng mới của ngành tài chính, cùng với sự phát triển của trí tuệ nhân tạo, số lượng nghiên cứu dự đoán rủ ro tín dụng trong mô hình peer to peer ngày càng tăng lên. Emekter, Tu, Jirasakuldech và Lu [3] đã áp dụng logistic regression dể dự đoán khả năng vỡ nợ và tìm ra một số mối quan hệ giữa khả năng trả nợ và điểm tín dụng (credit score), tỉ lệ nợ trên thu nhập (debt-to-income ratios), FICO scores, và hạn mức tín dụng xoay vòng (revolving credit lines). Malekipirbazari và Aksakalli [2] đã sử dụng các phương pháp máy học để để phân loại bad loans hay good loans như random forest, logistic regression, K- nearest neighbor, support vector machines. Họ thấy rằng sử dụng máy học là cách tiếp cận hiệu quả hơn so với phân tích những số liệu tài chính như FICO hay LC grades mà Lending club đã cung cấp.

## Vấn đề mất cân bằng lớp

Việc mất cân bằng lớp xảy ra thường xuyên trong các vấn đề phân loại. Mất cân bằng lớp thường được thấy trong các tập dữ liệu thực tế hay các tập dữ liệu về tài chính.

Mất cân bằng lớp xảy ra khi số lượng phần tử của một lớp chiếm ưu thế hơn so với các lớp còn lại. Việc này sẽ làm cho các dự đoán sẽ thiên về lớp chiếm đa số hơn, còn lớp thiểu số sẽ bị “ngó lơ”. Trong nhiều thập kỉ qua, nhiều thuật toán đã dược thiết kế để giải quyết vấn đề này. Trong đó, làm lại mẫu (Resampling) là một trong những thuật toán quan trọng nhất [4]. Resampling sẽ tạo ra sự cân bằng các lớp trong tập dữ liệu trước khi nó được đem đi huấn luyện.

Có 3 phương pháp chính để làm lại mẫu:

Oversampling là tăng số lượng mẫu ở lớp thiểu số bằng mới số lượng mấy của lớp đa số: SMOTE, ROS, ….

Undersampling là giảm số lượng mẫu ở lớp đa số bằng mới số lượng mấy của lớp thiểu số: NearMiss, RUS, ….

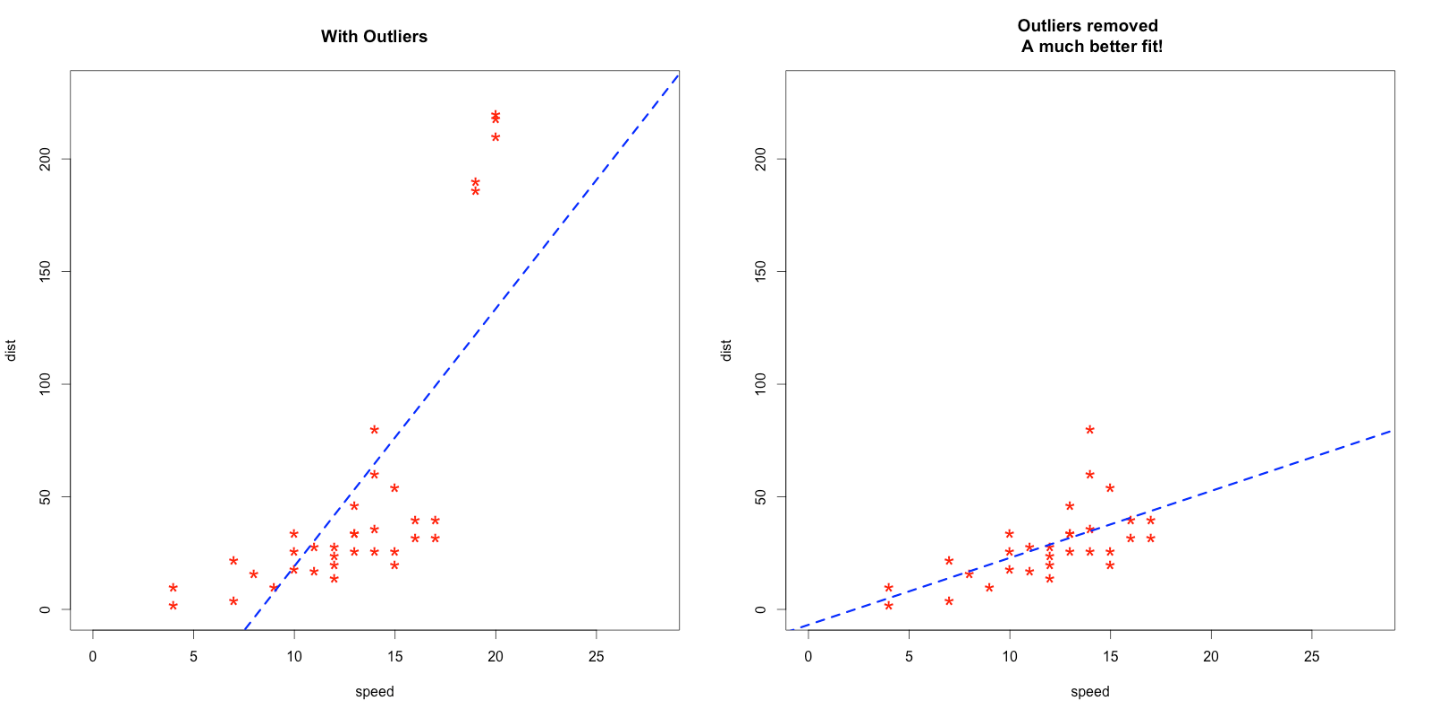
Kết hợp 2 phương pháp trên: SMOTE+ENN, SMOTE+Tomek, ….

## Feature Engineering

Feature Engineering là quá trình xử lý từ dữ liệu thô (raw datta) thành 1 tệp dữ liệu đã được làm sạch thông qua các kỹ thuật khai thác dữ liệu. Việc làm sạch dữ liệu dùng để cải thiện hiệu suất của các thuật toán máy học.

Các tập dữ liệu thực tế thường có những giá trị rỗng (Missing values). Tùy vào tính chất của từng tập dữ liệu sẽ có missing values khác nhau. Có những giá trị rỗng hiện hữu ta có thể quan sát trực tiếp được như các giá trị NULL, NaN (not a number). Hoặc cũng có những giá trị rỗng ẩn (hidden NaNs) thường là những giá trị quá khác biệt so với phân phối của dữ liệu. Việc xử lý Missing values có nhiều cách khác nhau phụ thuộc vào từng feature riêng biệt. Ta có thể xóa hẳn những dòng, côt chứa giá trị rỗng đó hoặc là lấp đầy nó bằng mean, mode, median, -1, ….

Không chỉ có Missing values, các giá trị ngoại lại (outliers) cũng ãnh hưởng tới mô hình phân loại. Các giá trị ngoại lai có thể làm lệch phân vùng dự đoán của mô hình.

*ảnh hưởng của giá trị ngoại lai tới mô hình dự đoán [11]*

*3*Hình 3.3-1

Có nhiều phương pháp để phát hiện giá trị ngoại lai và loại bỏ chúng như: Z-scores, tứ phân vị, DBSCAN, KNN, ….

Chuẩn hóa dữ liệu cũng là một trong những vấn đề quan trọng trong máy học. Để mô hình có thể học tối nhất với những dữ liệu đạng phân loại (categorical features) ta cẩn chuyển về kiểu dữ liệu số. Những cách chuyển thường dùng với categorical features là Label encoder, onehot encoder, …. Còn với những kiểu dữ liệu số (numeric features) thì lại khác có những mô hình phân loại như logistic regression, KNN hay regularization thì việc chuyển kề một chuẩn nhất định lại là ảnh hưởng lớn. Nhưng với những mô hình tree-based như decision tree hay random forest thì lại không. Có những thuật toán để chuyển dữ liệu dạng số này như: min max scale, standard scale, … Với mỗi mô hình thì ta sẽ dùng chung một loại chuyển đổi với số.

Min max scale:

## Mô hình đề xuất

Để giúp người cho vay đánh giá mức dộ tín nghiệm của ngưởi vay trong trong các nền tảng vay peer to peer, nhóm tác giả đề xuất mô hình này để hỗ trợ người cho vay quyết định đúng đắn hơn, từ đó giảm thiệu rủ ro tín dụng.

*A close up of a logo

Description automatically generatedHình 4.1. Các bước xử lý của mô hình đề xuất*

4Hình 3.4-1

Từ bộ dữ liệu thô, qua quá trình feature engineering, dữ liệu sẽ được sử lý missing values, xử lý giá trị ngoại lai và chuẩn hóa dữ liệu bằng các phương pháp đã nêu trên.

Sau đó dữ liệu sẽ được tách thành hai phần: test set, và training set.

Để chuẩn bị cho quá trình huấn luyện mô hình, tập training set sẽ được tách thành hai tập nữa một là training set, hai là tập validation set cho quá trình thẩm định. Training set sẽ được xử lý mất cân bằng dữ liệu bằng các thuật toán làm lại mẫu.

Sau quá trình làm lại mẫu, tập training set sẽ được đưa và mô hình huấn luyện bằng các thuật toán máy học. Mô hình sẽ được kiểm tra bằng cách bằng các phương pháp đánh giá qua tập validation test. Nếu mô hình tốt thì sẽ mô hình đó sẽ là mô hình được dùng để đự đoán. Nếu không ta sẽ trở lại feature engineering để cải thiện mô hình.

# Kết quả thực nghiệm

## Bộ dữ liệu

Bộ dữ liệu thực tế được lấy từ Lending Club từ năm 2007 đến quý 4 năm 2019.

Bộ dữ liệu có 151 features và 2260701 dòng dữ liệu

## Mô hình thực nghiệm

Dựa trên nguyên lý hoạt động, các ưu và khuyết điểm của các thuật toán huấn luyện dữ liệu (data train), nhóm quyết định chọn Logistic Regression và LightGBM cho bài toán này và thức hiện theo mô hình trên.

## Feature engineering

Làm sạch dữ liệu

Đối với tập dữ liệu thô ban đầu, nhóm tác giả đã lựa chọn ra các feature có số dòng dữ liệu rỗng không quá 20%.

Sau dó, nhóm tác giả đã kết hợp các kĩ thuật mean, mode, median để lập đầy những ô dữ liệu rỗng.

Chọn lọc features

Sau quá trình làm sạch dữ liệu, nhóm tác giả đã phân tích, nghiên cứu để chọn ra 25 features quan trọng sau:

|  |  |
| --- | --- |
| TÊN FEATURE | MÔ TẢ |
| loan\_anmt | Số tiền được liệt kê của khoản vay được áp dụng cho người vay. Nếu tại một thời điểm nào đó, bộ phận tín dụng giảm số tiền cho vay, thì nó sẽ được phản ánh trong giá trị này. |
| int\_rate | Lãi suất cho khoản vay |
| annual\_inc | Thu nhập hàng năm tự báo cáo được cung cấp bởi người vay trong quá trình đăng ký. |
| dti | Tỷ lệ được tính bằng cách sử dụng tổng số tiền thanh toán nợ hàng tháng của người vay trên tổng số nghĩa vụ nợ, không bao gồm thế chấp và khoản vay LC được yêu cầu, chia cho thu nhập hàng tháng của người vay. |
| delinq\_2yrs | Số lượng hơn 30 ngày quá hạn do các khoản nợ quá hạn trong hồ sơ tín dụng của người vay trong 2 năm qua |
| fico\_range\_low | Phạm vi ranh giới thấp hơn của người vay FICO tại nguồn gốc cho vay thuộc về. |
| inq\_last\_6mths | Số lượng câu hỏi trong 6 tháng qua (không bao gồm yêu cầu tự động và thế chấp) |
| verification\_status | Cho biết nếu thu nhập được xác minh bằng LC, không được xác minh hoặc nếu nguồn thu nhập được xác minh |

Bảng thống kê dữ liệu

1Bảng 4-1

Kết quả thực nghiệm

Với nhiều lần huấn luyện, thử lại với các thông số thuật toán khác nhau nhóm tác giả đã ra kết quả như sau:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classifier | Accuracy | AUC | Sensitivity | Specifity | FP - Rate | G - mean |
| LR - RUS | 0.6963 | 0.7358 | 0.72 | 0.6122 | 0.3877 | 0.6639 |
| LR - NM | 0.6045 | 0.6763 | 0.5917 | 0.6501 | 0.3498 | 0.6202 |
| LR - SMOTE | 0.6845 | 0.7344 | 0.6991 | 0.6324 | 0.3675 | 0.6649 |
| LR - ROS | 0.6958 | 0.7358 | 0.7188 | 0.6143 | 0.3856 | 0.6645 |
| LR - SMOTE\_TOMK | 0,6843 | 0,7339 | 0,6990 | 0,6319 | 0,3680 | 0,6646 |
| LR - SMOTE\_ENN | 0,5895 | 0,7340 | 0,5365 | 0,7778 | 0,2221 | 0.6460 |
|  | | | | | | |
| RF - RUS | 0.6732 | 0.7467 | 0.6716 | 0.6788 | 0.3211 | 0.6752 |
| RF - ROS | 0.6785 | 0.7483 | 0.6802 | 0.6723 | 0.3276 | 0.6763 |
| RF - NM | 0.5732 | 0.6712 | 0.5350 | 0.7093 | 0.2906 | 0.6160 |
| RF - SMOTE | 0.6978 | 0.7245 | 0.7285 | 0.6087 | 0.3912 | 0.6660 |
| RF - SMOTE\_TOMK | 0.7075 | 0.7178 | 0.7592 | 0.5239 | 0.4760 | 0.6307 |
| RF - SMOTE\_ENN | 0.6469 | 0.7279 | 0.6358 | 0.6862 | 0.3137 | 0.6605 |

(LR - logistic regression, RF – Random forest (LightGBM))

2Bảng 4-2

# KẾT LUẬN

## Kết quả đạt được của nghiên cứu:

Nhìn chung huấn luyện cho kết quả tương đối tốt, các chỉ số đánh giá mô hình cao, FPR tương đối thấp.

Thuật toán Random forest kết hợp với Gradient boosting cho tốc độ và kết quả dự đoán tốt hơn Logistic Regression

Ưu và khuyết điểm của phương pháp đề xuất

Kết quả của thực nghiệm tương đối tốt nhưng chưa tận dụng hết các feature liên quan đến thời gian và và các kiểu dữ liệu dạng chữ.

Thời gian huấn luyện tương đối nhanh, nhưng lại tốn nhiều thời gian trong việc tìm kiếm và tối ưu các tham số cho từng mô hình.

Ngoài RUS và ROS thì tất cả các thuật toán resampling kể trên đều có tốc độ chạy chậm trên tập dữ liệu lớn (>1,000,000 dòng).

## Đề xuất cho tương lai:

Tiếp tục quá trình feature engineering để tìm những feature tốt nhất.

Áp dụng mean encoder trong quá trình huấn luyện để tăng hiệu suất.

Tiếp tục huấn luyện dữ liệu bằng các thuật toán máy học khác.

Khi dữ liệu mới được cập nhập ưu tiên xét xem có feature mới hay không và làm sạch chúng.

TÀI LIỆU THAM KHẢO

[1] I. Tomek, “Two modifications of CNN,” In Systems, Man, and Cybernetics, IEEE Transactions on, vol. 6, pp 769-772, 2010

[2] Malekipirbazari, M., and Aksakalli, V.: ‘Risk assessment in social lending via random forests’, Expert Systems with Applications, 2015, 42, (10), pp. 4621- 4631

[3] Emekter, R., Tu, Y., Jirasakuldech, B., and Lu, M.: ‘Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending’, Applied Economics, 2015, 47, (1), pp. 54-70

[4] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G.: ‘Learning from classimbalanced data: review of methods and applications’, Expert Systems with Applications, 2017, 73, pp. 220- 239

[5] Anahita Namvar, Mohammad Siami, Fethi Rabhi, Mohsen Naderpour: “Credit risk prediction in an imbalanced social lending environment”

[6] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu, Microsoft Research, Peking University. Microsoft Redmond: “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”

**Mục lục**

Contents

[1 GIỚI THIỆU 2](#_Toc119675438)

[1.1 Giới thiệu đề tài 2](#_Toc119675439)

[1.2 Bố cục bài báo cáo 2](#_Toc119675440)

[2 CƠ SỞ LÝ THUYẾT 2](#_Toc119675441)

[2.1 Linear Regression 2](#_Toc119675442)

[2.2 Logistic Regression 3](#_Toc119675443)

[3 Giới thiệu mô hình 4](#_Toc119675444)

[3.1 Tổng quan 4](#_Toc119675445)

[3.2 Vấn đề mất cân bằng lớp 5](#_Toc119675446)

[3.3 Feature Engineering 5](#_Toc119675447)

[3.4 Mô hình đề xuất 7](#_Toc119675448)

[4 Kết quả thực nghiệm 8](#_Toc119675449)

[4.1 Bộ dữ liệu 8](#_Toc119675450)

[4.2 Mô hình thực nghiệm 8](#_Toc119675451)

[4.3 Feature engineering 8](#_Toc119675452)

[5 KẾT LUẬN 11](#_Toc119675453)

[5.1 Kết quả đạt được của nghiên cứu: 11](#_Toc119675454)

[5.2 Đề xuất cho tương lai: 11](#_Toc119675455)

**Mục lục hình**

[Hình 2.1-1 3](#_Toc119676182)

[Hình 2.2-1 4](#_Toc119676183)

[Hình 3.3-1 6](#_Toc119676184)

[Hình 3.4-1 7](#_Toc119676185)

**Mục lục bảng**

[Bảng 4-1 10](#_Toc119676211)

[Bảng 4-2 10](#_Toc119676212)