Predicting Mortgage Approvals (Classification)

By
Wenseslaus Raphael Mwita
Ahmed Meshref Ramadan
Goodluck Caiser Malata

Case scenario:
The world is facing a massive problem in the loan issuing process. Our goal from this project is to predict whether to accept a mortgage application or deny it according to the given dataset, which is adapted from the Federal Financial Institutions Examination Council's (FFIEC). Usually, it takes longer to decide whether a customer should get a loan or not. Customers get very frustrated when they have to wait for a very long period before they know whether they will get a loan or not even after submitting all the required documents. This is because most of the time, financial institutions don't have a system that can help them predict the outcome of the loan issued to the customer. So they try calculating all that manually, which takes longer and sometimes is not as accurate as a machine would do. With the dataset from FFIEC about previous loan status (accepted or denied), we will predict whether a person should get a loan or not.

Solution:
To solve this problem, we will use a classification model that will easily check whether or not to accept a loan request by feeding the needed data to the model. We have decided to use a classification model for this project because of the classification. We are dealing more with grouping/classes or categories, which for your case, is either a person is eligible for a loan or not. If our situation were to find out how much loan/money a person would be given in a bank, then we would use regression since, with regression, we are mostly predicting the quantitative aspect of the problem and not qualitative. The main difference between classification and regression is that with classification, we are anticipating a label, and in regression, we are predicting a quantity.

Classification:
Classification is just one model of different machine learning algorithms. There are three types of algorithms in machine learning supervised, unsupervised, and semi-supervised machine learning. The difference between all of those algorithms is the data labels (dependent variable).

Classification falls under supervised machine learning where a training dataset is labelled, and the labels (dependent variable) have categorical values. So, it is a prediction process of approximating a mapping function from input variables (x) to an output variable called category or labels. A classification model helps us to conclude the observed value given one or more inputs. Classification is different from other supervised machine learning models, such as linear regression when it comes to the values of the dependent variable. In linear regressions, the values of the dependent variable that we want to predict are continuous. But, for classification, the dependent variable is categorical (discrete), and we try to predict the category of each observation. That is why classification is the best model for our case scenario.

So, classification can be used in cases where we want to classify our observations based on their group. For example, if we're going to classify an email, either spam or not spam, or if we wanted to predict if a person has a specific disease or not based on the patients' age, symptoms, and nationality. Other cases that we can apply at ALU is predicting whether a student is going to pass or fail based on his past performance.

Classifier:
An algorithm that maps the input data (each observation) based on their independent features to a specific category from the dependent column. The algorithm learns from the training data, which are labelled and predict the labels of future data that are not labelled.

Classifiers:

- Binary classifier: Classification task with two possible outcomes.
  Examples:
  1. Gender classification (Male / Female)
  2. Classification of spam email and non-spam email

- Multi-class classifier: Classification with more than two distinct classes or two possible outcomes.
  Examples:
  1. Classification of animals.
  2. Classification of music types.

Classification Algorithms

Classification models include the following;

1. Random Forest
2. Gradient-Boosted tree
3. Naive Bayes
4. Decision Tree
5. Logistic Regression
6. Multilayer Perceptron
7. One-vs-rest.

We are not going to explain all of the above algorithms, but we will dig deep into one of them, which is logistic regression, which is one of the most popular algorithms for classification problems.

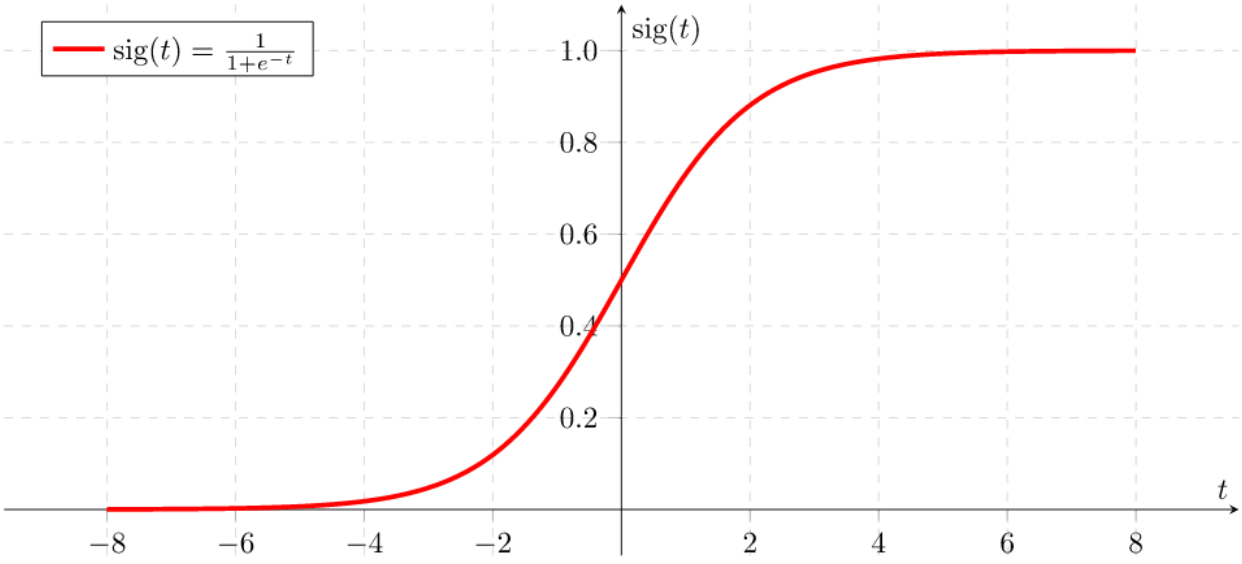
Logistic Regression

Despite the name logistic regression, it is not a regression algorithm but rather a classification. It merely predicts the probability of an observation belonging to each category from the set of categories in the dependent variable i.e., the total probability of an observation belonging to all of the categories is 1, but one probability can be higher than others, which means it belongs to that category. It takes certain inputs and determines the probability of a certain outcome. For example, if a child has a temperature of 104F (40C) and has a rash and nausea, the likelihood of having chickenpox could be around 80%. The rule in logistic

regression, if the likelihood is > 50%, then the decision is valid. So in this situation, the kid is likely to have chickenpox.

A mathematical explanation of logistic regression:

When we think about logistic regression, the first question is that how can we transform the model probability to an actual binary number, and for that we use a function called the sigmoid function which takes any real-valued number and maps it into a value between 0 and 1 but never exactly at those limits.



Sigmoid function

Logistic regression is similar to linear regression when it comes to the equation used which is the equation of the line that we use to predict values in the case of linear regression and separate data when it comes to logistic regression.

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2$$

x1, x2 are independent variables. βi, are the parameter s of the model such that β1, β2 are the coefficients of the independent variable, and β0 is the intercept or the bias factor that helps in not overfitting the data.

For example, if we are given some point that has two values for x1 and x2, (a,b), if we plugged it into our equation, the equation could output a positive result (for one class), negative result (for the other class), or 0 (the point lies right on the decision boundary).

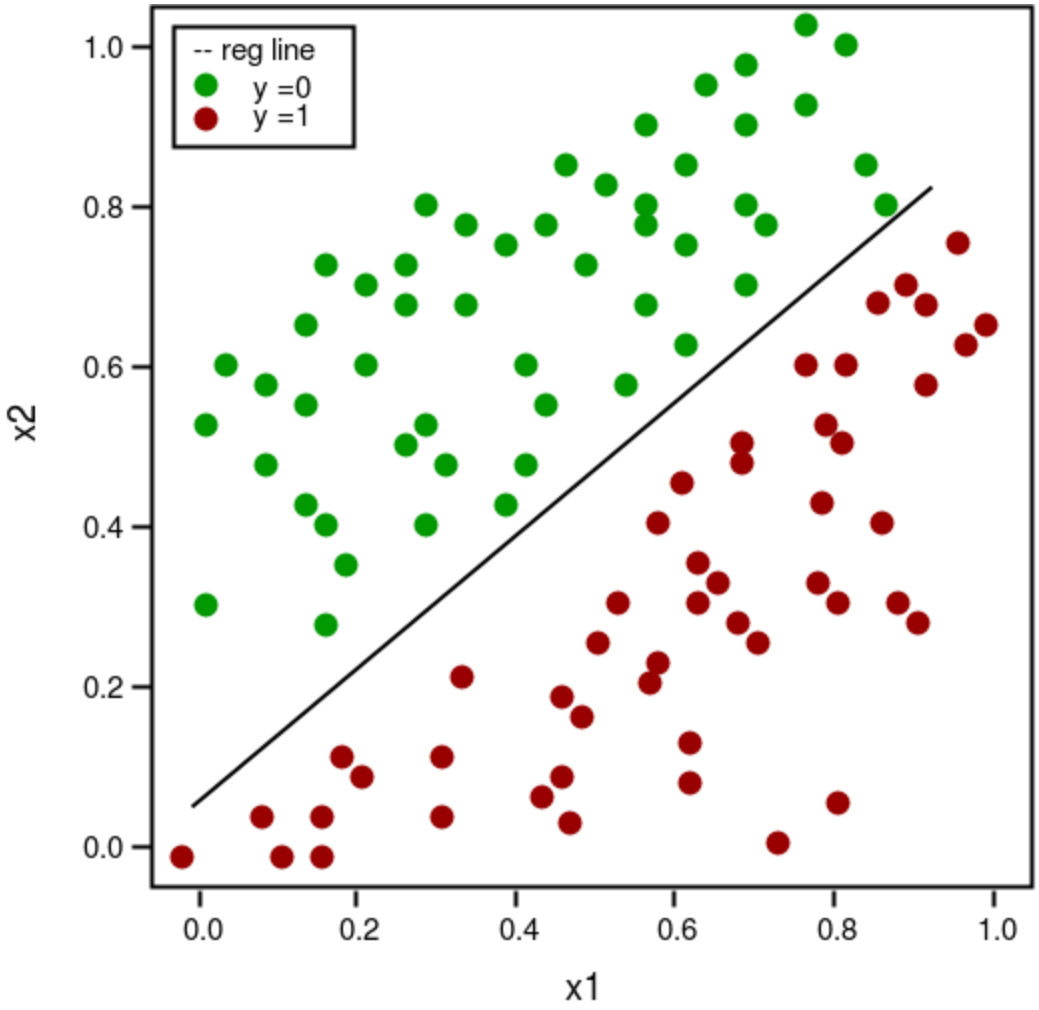
Know, the model needs to map the output of the function into the probability P, which goes from 0 to 1 and we will use the odds function for that.

So the final function becomes

$$Logit(p_{class1})) = log(\frac{p_{class1}}{1-p_{class1}}) = \beta_0 + x_1 * \beta_1 + x_2 * \beta_2$$

Limitations of logistic regression:

1. Before working with Logistic Regression, it is important to check that our data is linearly separable in n dimensions so that we can separate the data different classes with a linear line.



Example of a linearly separable data

2. Logistic regression works better when we remove unrelated or uncorrelated independent variables to the dependent variable (target).

3. The algorithm also works better when the independent variables are not correlated to each other. If two independent variables are correlated to

each other, we expect better results when we delete one of them and rely on the other to predict the independent variable.

4. Logistic regression can't solve non-linear problems (data) since its decision boundary is a linear line.

5. Logistic regression is simple and easy which makes it not most powerful algorithms out there and can be outperformed by other algorithms. In my case scenario, logistic regression gave me poor results and I have used CatBoostClassifier which is a more powerful algorithm.

Back to the case scenario (Summary) :
We analyzed the case scenario and used the CatBoostClassifier algorithm to classify the classes we have.

Please read the final report from the following Github repository **here**

Conclusion:
After using the logistic regression model which trained with 75% of the training data. Testing the model with the remaining 25% of the data yielded the following results:

- True Positives: 48935
- True Negatives: 42356
- False Positives: 20129
- False Negatives: 13580

With a total accuracy of 73%.

Many bank customers ask for loans, but they wait for a long time to get a response back from the bank after submitting all that's needed. With the classification model produced in this document, banks can easily check whether or not to accept a loan request by feeding the needed data to this model.